SIDN Labs

https://sidnlabs.nl

October 20, 2026

Peer-reviewed Publication

Title: BigTime – Characterizing Large Time Service Providers

Authors: Pascal Huppert, Giovane C. M. Moura, Marco Davids, Pieter-Tjerk de Boer, Ralph Holz, Rein Fernhout, Georgios Smaragdakis, and Cristian Hesselman

Venue: Proc. ACM Netw. 4, ACM CONEXT 2026

DOI: https://doi.org/10.1145/3856960
Article: 49, 26 pages

Citation:

- Pascal Huppert, Giovane C. M. Moura, Marco Davids, Pieter-Tjerk de Boer, Ralph Holz, Rein Fernhout, Georgios Smaragdakis and Cristian Hesselman. 2026. BigTime – Characterizing Large Time Service Providers. Proc. ACM Netw. 4, CoNEXT4, Article 49 (December 2026), 26 pages. https://doi.org/10.1145/3856960
- Bibtex:

```
@article{Huppert26,
  author    = {Huppert, Pascal and Moura, Giovane C. M. and
               Davids, Marco and de Boer, Pieter-Tjerk and
               Holz, Ralph and Fernhout, Rein and
               Smaragdakis, Georgios and Hesselman, Cristian},
  title     = {BigTime -- Characterizing Large Time Service Providers},
  journal   = {Proc. ACM Netw.},
  volume    = {4},
  number    = {CoNEXT4},
  articleno = {49},
  numpages  = {26},
  month     = dec,
  year      = {2026},
  doi       = {10.1145/3856960},
}
```

BigTime – Characterizing Large Time Service Providers

PASCAL HUPPERT, University of Twente, The Netherlands and University of Münster, Germany
GIOVANE C. M. MOURA, SIDN Labs and Delft University of Technology, The Netherlands
MARCO DAVIDS, SIDN Labs, The Netherlands
PIETER-TJERK DE BOER, University of Twente, The Netherlands
RALPH HOLZ, University of Münster, Germany and University of Twente, The Netherlands
REIN FERNHOUT, University of Twente, The Netherlands
GEORGIOS SMARAGDAKIS, Delft University of Technology, The Netherlands
CRISTIAN HESSELMAN, SURF and University of Twente, The Netherlands

The Network Time Protocol (NTP) keeps Internet-connected clocks synchronized, which core applications and protocols depend on for proper functioning, including DNS caches, RPKI, and TLS. Next to free time servers (NIST, NTP Pool), large cloud providers and OS vendors such as Microsoft, Apple, and Google have begun operating their own time services and setting their billions of systems/devices to use their NTP servers by default. This creates a level of service centralization around these providers, and yet there has been little scrutiny of these services and their characteristics. In this paper, we characterize seven large time service providers in terms of replication architecture, client mapping, accuracy and service provided. We found a large variation in service replication (DNS and anycast), how they map clients to servers, and in their accuracy and (security) feature support. This includes violations of best practices, servers clocks being incorrect by up to 50ms on one occasion, the use of outdated NTP versions, and a lack of support for NTS, RPKI, and DNSSEC.

CCS Concepts: • **Networks** → **Application layer protocols**; **Network measurement**; **Naming and addressing**; **Location based services**; **Network monitoring**.

Additional Key Words and Phrases: NTP; Accuracy; Time Services; NTS; Client mapping

ACM Reference Format:
Pascal Huppert, Giovane C. M. Moura, Marco Davids, Pieter-Tjerk de Boer, Ralph Holz, Rein Fernhout, Georgios Smaragdakis, and Cristian Hesselman. 2026. BigTime – Characterizing Large Time Service Providers. *Proc. ACM Netw.* 4, CoNEXT4, Article 49 (December 2026), 26 pages. https://doi.org/10.1145/3856960

1 Introduction

Synchronized clocks are essential for modern societies, being required for financial transactions, power grid operations, and telecommunication networks [36], among others. On the Internet, protocols that check validity periods depend on accurate clocks [27, 51, 58, 92]: TLS certificates [85], DNSSEC signatures [4], RPKI objects [14], and DNS caches.

Authors' Contact Information: Pascal Huppert, University of Twente, Enschede, The Netherlands and University of Münster, Münster, Germany, pascal.huppert@utwente.nl; Giovane C. M. Moura, SIDN Labs and Delft University of Technology, Arnhem and Delft, The Netherlands, giovane.moura@sidn.nl; Marco Davids, SIDN Labs, Arnhem, The Netherlands, marco.davids@sidn.nl; Pieter-Tjerk de Boer, University of Twente, Enschede, The Netherlands, p.t.deboer@utwente.nl; Ralph Holz, University of Münster, Münster, Germany and University of Twente, Enschede, The Netherlands, ralph.holz@uni-muenster.de; Rein Fernhout, University of Twente, Enschede, The Netherlands, r.p.j.fernhout@student.utwente.nl; Georgios Smaragdakis, Delft University of Technology, Delft, The Netherlands, g.smaragdakis@tudelft.nl; Cristian Hesselman, SURF and University of Twente, Utrecht and Enschede, The Netherlands, cristian.hesselman@surf.nl.

This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2834-5509/2026/12-ART49
https://doi.org/10.1145/3856960

Provider	Domain Name	User Base
Microsoft	`time.windows.com`	1.4 billion [55]
Apple	`time,time-[macos,euro,ios].apple.com`	2.2 billion [3]
Google	`time.android.com`, `time.google.com`	3.0 billion [24]
Ubuntu	`ntp.ubuntu.com`	Unclear
AWS	`time.aws.com`	-
Cloudflare	`time.cloudflare.com`	-
Meta	`time,time[1-5].facebook.com`	-

Table 1. Evaluated time service providers. Providers highlighted are enabled by default in their clients' OSes.

The Network Time Protocol (NTP) [58] is the Internet's default protocol for clock synchronization. NTP clients obtain time from NTP servers, which form a hierarchy: servers at the top (stratum 1) synchronize directly to a time reference, such as a Global Navigation Satellite System (GNSS) receiver (e.g., GPS or Galileo) or a radio time-signal receiver (e.g., DCF77 or WWVB) [58]. Public NTP services have operated for decades. Notable providers include metrological institutes such as NIST [71] (US), PTB [12] (DE), and VSL [102] (NL), as well as the volunteer-run NTP Pool project [76].

When an NTP server serves incorrect time, dependent services fail, and the resulting outages can extend beyond the Internet. In November 2012, the US Naval Observatory (USNO) NTP servers served timestamps 12 years in the past, causing outages in network routers and Active Directory servers [9, 64]. More recently, in July 2026, a single NTP server in the network of Telstra, Australia's largest telecommunications operator, set its clock back 1,024 weeks (to 2006) and propagated that time to downstream hosts. These hosts then rejected certificates as not yet valid, causing a nationwide mobile outage of about 12 hours that also blocked emergency (000) calls [98, 99]. In both cases, a single faulty time source disrupted every service that depended on it.

Besides metrological institutes and the NTP Pool, OS vendors recently started to run their own time services and set them as the default in their OSes [65]: Microsoft, Apple, Google, and Canonical (Ubuntu). Some cannot be changed: iOS, for instance, offers no option to set a different time server [47]. Cloud/Content providers also started public time services, including Cloudflare [21], AWS [94], and Meta [78].

OS vendors *control both ends* of synchronization: they operate the NTP servers and configure the clients to use them. As a result, just a few vendors are the time source for billions of devices, wherever those devices connect; Microsoft, Apple, and Google alone report 6.6 billion active devices (Table 1). Unlike the DNS, in which clients typically use the resolver of the network they join, NTP clients generally keep their configuration across networks[1].

The NTP Pool, default in many Linux distributions, has been examined by multiple studies [7, 46, 65, 92]. The time services of OS vendors and cloud and content providers have not, although the OS vendors alone serve billions of devices. This paper closes this gap. We measure the time services of the seven time providers in Table 1 ("providers" hereafter) along the following three dimensions:

- *Deployment*: how each provider replicates its service, compared with a well-documented replicated service, the Root DNS [91] (§3).
- *Client mapping*: which criteria each provider uses to assign clients to servers (§4).
- *Accuracy*: the offset of each provider's servers, measured from GPS-synchronized vantage points, along with their stratum, NTP version, and support for IPv6 and NTS (§5).

We make the following contributions:

[1]DHCP servers can announce NTP servers [1], but clients are not required to use them.

- We show a large diversity in terms of replication: three out of seven providers use IP anycast [80], whereas the other four generally use more IP addresses (up to 90). For unicast-based providers, we find that most use client geolocation to map clients to server, just like the NTP Pool.
- We show how Microsoft's time service served half of our 9k vantage points (VPs) with only one NTP server IP address, providing no redundancy and violating NTP best practices [84].
- We measure the accuracy of the providers on four GPS-synced VPs and show they deliver accurate time. However, three of Microsoft's servers were out-of-sync for several hours during one measurement, by up to 50ms.
- We show the diversity of NTP services provided in terms of stratum-level of the servers, NTP version, and IPv6 support. Only Cloudflare deploys the latest standards: IPv6, Network Time Security (NTS) [30], RPKI, and DNSSEC. Microsoft is the only provider not supporting IPv6, and provides NTP version 3 only (all others provide NTPv4).

2 Background

2.1 The Network Time Protocol

NTP is designed to synchronize the clocks of computer systems over variable-latency networks [58]. Over the Internet, it provides an accuracy of milliseconds from Coordinated Universal Time (UTC) by using a hierarchical system of time sources and servers organized into levels by *stratum*, which indicate the distance from the reference clock. Stratum 1 servers are directly connected to a time source (e.g., GNSS, or radio). Stratum 2 servers are synchronized with stratum 1 servers.

Client-Server synchronization: NTP clients synchronize their clocks by exchanging timestamped requests and responses with NTP servers. Each exchange yields four timestamps: the origin timestamp T_1 (client time when the request is sent), the receive timestamp T_2 (server time when the request arrives), the transmit timestamp T_3 (server time when the response is sent), and the destination timestamp T_4 (client time when the response arrives), as can be seen in Figure 1. The client computes the offset θ between its clock and the server's assuming symmetric network delay and *may* adjust its clock by θ to align with the server. Often, further algorithms are used to filter and average multiple responses from multiple servers.

Protocols: SNTP: The Simple Network Time Protocol (SNTP) is a subset of NTP mechanisms for clients that do not need full NTP accuracy [58, §14]. SNTP uses the NTP packet format, but its clients typically query only one server and lack NTP's filtering and selection algorithms. A recent non-peer-reviewed study [45] found that the default clients of Windows Server, macOS, and Ubuntu (as of October 2025) are SNTP-based and vulnerable to time-shift attacks.

NTS: Network Time Security (NTS) [30], standardized in 2020, authenticates NTP. Client and server first establish keys over TLS (NTS Key Establishment); they then exchange NTP packets whose integrity and authenticity are protected, which prevents spoofing, replay, and on-path modification of time. Six years after standardization, NTS is still rarely deployed (§5.4).

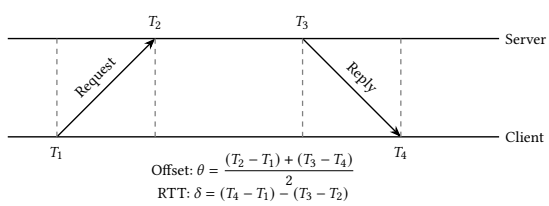


Fig. 1. Timestamps used in NTP offset calculations

NTPv5, PTP, and White Rabbit: NTPv5 is currently being standardized [49]; it only has client-server mode, removing uncommon modes of operation present in NTPv4. Applications that require sub-microsecond accuracy, such as mobile fronthaul and power grid operations, use the Precision Time Protocol (PTP) [39] instead. White Rabbit extends PTP with Synchronous Ethernet

and hardware phase measurement, reaching sub-nanosecond accuracy over fiber [19, 50]. Both rely on hardware timestamping and on switches that support them, so they run within a single administrative network (e.g., SURF's time and frequency research network in The Netherlands [97]), not across the Internet; we do not consider them in this paper.

GNSS dependence: Many stratum 1 servers use GNSS receivers as their reference clock, which makes NTP, and the services that depend on it, vulnerable to GNSS jamming and spoofing. Norway's telecom regulator reports that GNSS signals have been jammed or degraded with increasing frequency in recent years, in particular in eastern Finnmark, along the Russian border [40]. In response, the Norwegian government proposed NOK 94 million (≈US$9.3 million) in its 2027 budget for a ground-based national time service, independent of satellites, to supply mobile networks, financial transactions, power distribution, and data networks [40, 72].

2.2 RIPE Atlas

RIPE Atlas [89, 90] is an Internet measurement platform operated by the RIPE NCC, the Regional Internet Registry (RIR) for Europe, the Middle East, and parts of Central Asia [88]. Volunteers host its probes in homes, enterprises, and data centers. Probes run DNS, ping, traceroute, and NTP measurements, and results of measurements are openly available. Researchers can use existing data or schedule own measurements, making it widely used in measurement studies (e.g., [6, 65, 66, 74]). We use RIPE Atlas probes as vantage points (VPs) for DNS measurements.

VP distribution: RIPE Atlas probes are concentrated in Europe and North America (Table 2), a bias documented since the platform's early years [6]. It has a vast number of vantage points, supports DNS and NTP measurements, and results are public. For our study, country coverage matters more than probe density: providers map clients by location (§4), so each country with at least one VP reveals its mapping. Atlas covers 184 countries. We used Atlas for our measurements and make the data available: [86, 87]

			ASes	
Continent	**VPs**	**Countries**	**IPv4**	**IPv6**
Europe	9,235 (60.4%)	46	2,615	1,463
North America	2,947 (19.3%)	22	958	415
Asia	1,808 (11.8%)	47	735	374
Africa	363 (2.4%)	40	164	85
South America	492 (3.2%)	12	276	154
Oceania	445 (2.9%)	16	130	83
Unspecified	4 (0.0%)	-	4	1
Total	15,294	183	4,686	2,429

Table 2. Active RIPE Atlas VPs, per continent (2026-10-05). Some ASes span multiple continents.

3 Time Services Architecture

The OS vendors in Table 1 serve billions of devices; a service at this scale must be replicated. In this section, we measure the mechanism and extent of this replication. For reference, we first describe how two well-documented services are replicated: the NTP Pool [76] and the Root DNS [91].

3.1 Replication in the NTP Pool and the Root DNS

Services can be replicated at three levels (Figure 2): *DNS-level*, mapping a domain name to multiple IP addresses; *IP-level*, with IP anycast [80], i.e., setting up multiple locations announcing the same IP prefix (*sites*); and *site-level*, distributing queries among multiple servers within a site using a load balancer. We use these levels to classify each provider in the remainder of this section.

NTP Pool: The NTP Pool is replicated only at the DNS level. On 2026-06-06, it had 3,800 IPv4 NTP servers, all reachable via `pool.ntp.org`. The Pool's authoritative DNS servers answer each query with a subset of these servers, chosen based on the client's location: servers from the same

country if any exist, otherwise from the same continent [65]. Individual volunteers within the NTP Pool, such as Cloudflare, may additionally use IP anycast.

Root DNS: The DNS root servers are replicated at all three levels. At the DNS level, the root zone lists 13 name servers (`[a–m].root-servers.net`), each with one IPv4 and one IPv6 address. Each address uses anycast: on 2026-06-06, the 13 root servers together operated 2,015 sites, 366 of them for F-Root alone [22]. Within a site, operators may run multiple servers behind a load balancer; K-Root, for instance, runs three servers at its Amsterdam site [17].

Replication and NTP: NTP is more delicate when it comes to replication than the DNS. A DNS resolver only sends individual queries, whereas an NTP client keeps per-server state (e.g., offset and delay samples) and assumes that consecutive responses from an IP address come from the same server. Anycast complicates this [84, §7]: the client does not know which server it reaches. When routing changes, it silently switches to another server, which can have a slightly different time or communication delay. Moreover, a client configured with one anycast address still only has a single time source and cannot identify falsetickers, a disadvantage over using individual IP addresses for each site; RFC 8633 [84] therefore recommends multiple anycast addresses, and that each server withdraw from anycast when it fails. Site-level load balancing has similar downsides to anycast: consecutive queries from one client may reach different servers.

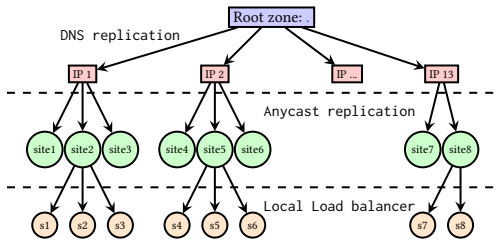


Fig. 2. Service replication in the Root DNS system.

Limitation: site-level replication. We do not quantify NTP site-level load balancing. A load balancer hides its servers behind a single IP address, and the hidden NTP servers cannot generally be distinguished, unlike in the Root DNS, whose servers can be identified via NSID [5] or `hostname.bind`. NTP has no such field. We attempted to distinguish servers behind the same NTP address by combining the reference ID and reference timestamp [58] of their responses, but we could not identify multiple servers. However, this does not prove the absence of local load balancing. Our counts – the number of IP addresses and anycast sites per provider – are therefore lower bounds on the number of servers each provider operates.

3.2 DNS and Anycast Replication

3.2.1 DNS-level replication. We first evaluate if the providers deploy DNS-level replication. To determine whether they rely on the client's geographical location when mapping users to servers as the NTP Pool does, we configure 500 RIPE Atlas probes (§2.2) as vantage points to send one DNS query per 10 minutes, for 8h (datasets: [86]). We configure each vantage point to query one of the authoritative DNS servers of the time service directly[2]. This way, we bypass the recursive resolvers' caches and retrieve fresh responses with each new query.

Why bypassing resolvers: We query the authoritative servers directly, as in a previous study of the NTP Pool [65], to measure the provider's mapping policy itself. Through a recursive resolver, a client's answer also depends on factors outside the provider's control: the resolver may return a cached answer until its DNS time-to-live (TTL) expires, it may be far from the client, and the

[2]For Microsoft and Apple, we do not configure the VPs to query the domains shown in Table 1 but the *actual* domains used. Microsoft's `time.windows.com` domain name redirects to `twc.trafficmanager.net` using a CNAME record. Apple's domains redirect to two domains: `[time,time-osx].g.aaplimg.com`. However, we find that both domain names point to the same NTP servers. Hence, we just report on `time.g.aaplimg.com` and include `time-osx.g.aaplimg.com` in Appendix B and Appendix C

authoritative server then sees the resolver's address unless the resolver forwards the client's subnet via EDNS Client Subnet (ECS) [23]. Direct queries remove these factors: each query returns a fresh answer, and the authoritative server sees the VP's own address. Our results therefore show how each provider maps clients by location; the mapping a client actually uses may differ when its resolver is distant and does not use ECS.

For each provider, we then compute the number of unique IP addresses they return over the course of the measurement. Table 3 summarizes the results. We notice clear differences: AWS has the most servers at 90 IPv4 addresses, followed by Apple (53) and Microsoft (12). Google, despite likely having the largest client population, has only 4 addresses.

3.2.2 Anycast replication. Next, we determine if the IP addresses from our DNS measurements (§3.2) use anycast. Anycast is not visible from a single vantage point: BGP routes each client to a single site, so unicast and anycast addresses appear the same.

Detecting anycast requires measuring an address from multiple geographically distributed vantage points and either finding latencies that violate speed-of-light constraints [20] or observing responses arriving at multiple sites of an (in turn) anycast-enabled network [96]. We rely on three existing, regular measurements: IP-Info [41], the BGP.Tools anycast prefix list [8], and the LACeS anycast census [34]. Each method can miss anycast deployments, e.g., those with few or geographically close sites; we therefore classify an address as anycast if at least one classifies it as such.

	Domains	IPv4	IPv6	ASes v4	ASes v6
Microsoft	1	12	0	1	0
Apple	4	53	48	2	2
Google	2	4	4	1	1
Ubuntu	1	4	3	1	1
Amazon	1	90	90	2	2
Cloudflare	1	2	2	1	1
Meta	1	5	5	1	1

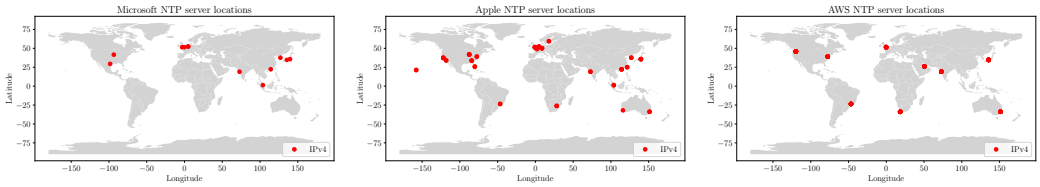
Table 3. DNS-level replication of time service providers.

Provider	Prefixes		Anycast	Sites	
	IPv4	IPv6		IPv4	IPv6
Microsoft	11	0	No	–	–
Apple	23	16	No	–	–
Google	1	1	Yes	62	35
Ubuntu	2	1	No	–	–
AWS	41	9	No	–	–
Cloudflare	1	1	Yes	75	51
Meta	5	5	Yes	10 to 18	7 to 16

Table 4. Anycast replication of time service providers.

Table 4 summarizes the results. We see that Google, Cloudflare, and Meta deploy IP anycast for their time services. For each network prefix, we report the maximum number of anycast sites as reported in October 2026 by the LACeS data [34]. For instance, Google has at least 62 sites, far more than IP addresses. Note that these are lower bounds, as measuring anycast deployments requires a large number of vantage points and depends on BGP routing. For Meta, the small number of sites is likely close to correct, but for larger numbers, the anycast census might report significantly smaller numbers, meaning that Google and Cloudflare could have multiple hundreds of sites [34]: Statistics about Google's network [32] suggest the actual number may be up to 200 sites, while Cloudflare claims to have 330 locations [21].

3.2.3 Unicast services. The remaining four providers (Microsoft, Apple, AWS, and Ubuntu) use unicast addresses. Counting their servers requires a different measurement than detecting DNS-level replication. These providers map clients by location (§4), so each VP may only receive the servers assigned to its region, and the number of servers we observe grows with the number of locations we measure from. Our first measurement (500 VPs, §3.2) suffices to detect DNS-level replication, but not to enumerate servers.



(a) Dataset: Microsoft-map [86] (b) Dataset: Apple-map1 [86] (c) Dataset: AWS-map1 [86]

Fig. 3. Unicast NTP servers geolocation.

	Microsoft	**Apple**	**AWS**
Domain	twc.trafficmanager.net	time.g.aaplimg.com	time.aws.com
VPs	9,010	8,994	8,939
Countries	170	170	171
ASes	3,094	3,094	3,081
Valid DNS Responses	642,617	642,042	640,253
NTP Servers	12	53	90
ASes	1 (AS8075)	2 (AS6185 & AS714)	2 (AS14618 & AS16509)
Dataset	Microsoft-map	Apple-map-1	AWS-map
Date	2025-02-08	2025-02-09	2025-02-18
Auth server	13.107.222.240	17.253.206.8	205.251.193.84

Table 5. Atlas measurements. Datasets: [86]

We therefore repeat the measurement for Microsoft, Apple, and AWS. We exclude Ubuntu as it consistently returned the same 4 IP addresses to each VP. We use 9,000 VPs in 170 countries and 3,000 ASes (Table 5), i.e., nearly all available probes. Each VP queries every 10 minutes for 12 hours. We discard addresses outside the provider's own ASes, which VPs likely receive because their DNS queries are intercepted [66, 73]. Microsoft, Apple, and AWS return 12, 53, and 90 IPv4 addresses, from 1, 2, and 2 ASes, respectively. Appendix C lists how often each address is returned.

Atlas DNS interception has been measured in previous studies [66, 73], which found it in a small fraction of VPs, similar to our results. Due to the small size of this phenomenon, we leave its effects on time synchronization to future work.

Measurement duration: Our 8- and 12-hour campaigns capture how DNS answers change over short timescales (48-72 queries per VP), but not how deployments change over weeks, as providers add or remove servers. To determine whether 8 hours are sufficient, we ran a second campaign from April to May 2026 (60 days) with the same 8.9k VPs, at three queries per day (datasets: [87]). Over two months, we observed 12, 52, and 86 addresses for Microsoft, Apple, and AWS (Table 13 in Appendix C); *all* of them were also observed in the previous measurement. We therefore conclude that our server counts do not underestimate the deployments.

Server locations: Figure 3 shows the geolocation of the servers of Microsoft, Apple, and AWS (Ubuntu in Appendix B). None of Microsoft's 12 addresses map to the southern hemisphere, so its clients in South America, Africa, and Oceania reach servers in the northern hemisphere.

Takeaway: Replication does not follow client population: Google serves 3 billion devices from one anycast prefix, and Microsoft serves 1.4 billion from 12 unicast addresses. Yet for NTP, providing multiple independent time sources to each client is even more relevant than the number of servers theoretically available. RFC 8633 recommends at least four [84, §3.2]. We measure this next.

4 Client/Server Mapping

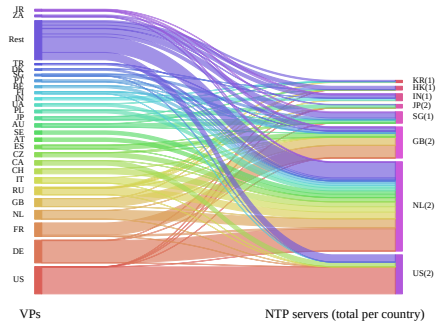
Next, we determine how each provider maps clients to its servers, and to how many. As a reference, the NTP Pool maps clients by geolocation: to servers in the client's country, or in its continent if the country has none [65]. Its operators chose this to limit packet loss and reduce time inaccuracies that asymmetric routes cause [33].

4.1 Mapping Criteria

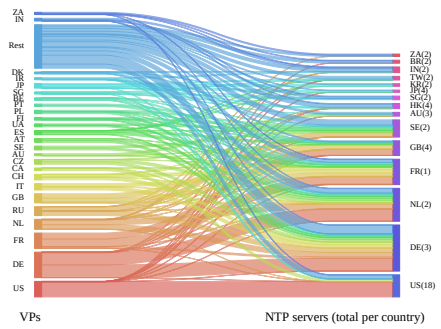
We analyze the mapping of Microsoft, Apple, and AWS, using the DNS measurements in Table 5. For each VP, we compare its location (from the RIPE Atlas probe metadata [18]) with the geolocation of the servers it receives (Figure 3).

For anycast-enabled providers, clients are mapped to sites by BGP, not the DNS. Because NTP responses generally do not identify the answering server (§3.1), determining which site serves each VP requires inferring it from traceroutes or latency [20]. We leave this analysis, and how many of each provider's sites our VPs reach, to future work. We also exclude Ubuntu as it simply returns all four addresses to every VP.

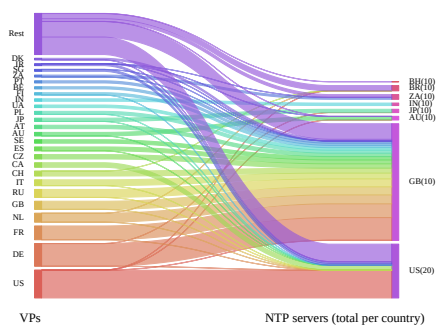
Figure 4 shows the DNS mappings we observe: the left side of each diagram shows the countries of the VPs, and the right side shows the countries of the servers returned to them. All three providers map clients by location. Microsoft's domain, for instance, resolves through Azure Traffic Manager (`twc.trafficmanager.net`), which supports geolocation-based mapping [56]. As discussed in §3.2, these mappings are based on the VP's own address. Clients whose resolver is in another country and does not use ECS [23] may be mapped based on the resolver's location instead.



(a) Microsoft. Dataset: Microsoft-map [86]



(b) Apple. Dataset: Apple-map-1 [86]



(c) AWS. Dataset: AWS-map-1 [86]

Fig. 4. Client-server mapping: sankey diagrams of Atlas Probe VPs and NTP Servers. Countries with less than 70 VPs are grouped as *Rest*

In-country mappings: Clients in countries where a provider operates servers mostly receive servers in their own country, as in the NTP Pool. Microsoft maps 1,297 of 1,313 US VPs (98.7%) to US servers, and AWS 1,305 of 1,311 (99.5%). Apple does the same in the US, but treats Western Europe as one region: its 1,036 German VPs receive servers in both Germany (DE) and the Netherlands (NL) (Figure 4b). Western Europe covers about one eighth of the contiguous US area, which Apple serves in-country, so the distances are comparable.

Out-of-country mappings: Providers handle clients from countries without servers differently.

- *Microsoft* falls back to the client's continent as the NTP Pool does: German VPs receive mostly Dutch servers, and Russian VPs, whose country spans two continents, receive servers in both

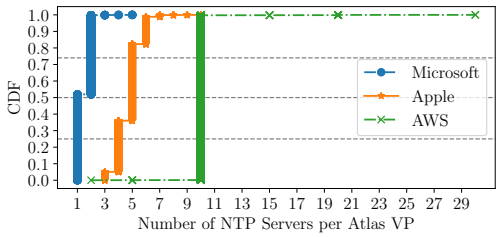


Fig. 5. Distinct NTP server addresses per VP for Microsoft, Apple, and AWS (Table 5).

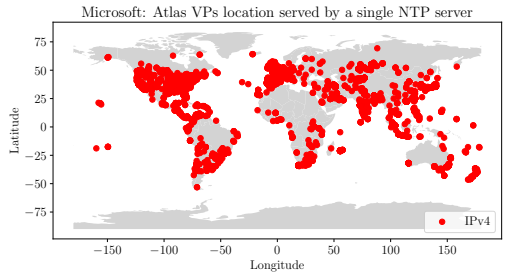


Fig. 6. Locations of VPs that Microsoft serves a single NTP server address.

Europe and Asia. Without servers in South America or Africa (Figure 3a), it maps most South American VPs to the US and most African VPs to Europe.

- *Apple* maps clients across continents. South African VPs receive servers from Brazil (53) and Asia (38) as often as South African ones (91), even though Apple operates servers in-country; and Iranian VPs (77) receive servers in India and Sweden. Spanish VPs receive servers in France, the United Kingdom, and Germany.
- *AWS* serves all of Europe from the United Kingdom, its only European NTP location (Figure 3c), and South American and African VPs from Brazil and South Africa, respectively. In Asia, it maps VPs from Singapore, Indonesia, Iran, and China to North America, although it operates NTP servers in Bahrain, India, and Japan.

Takeaway: Like the NTP Pool, all three providers use the client's location as the primary mapping criterion, and map clients in-country where they have servers. They differ in the fallback: only Microsoft follows the Pool's continent rule, while Apple and AWS map some clients to other continents.

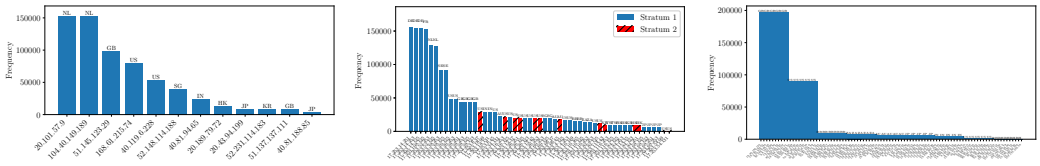
4.2 Number of servers per client

RFC 8633 (BCP 223) recommends that clients use at least four independent time sources, so that they are able to detect and discard a false-ticking one [84, §3.2]. We measure how many sources each provider makes available to each client.

The providers fall into three groups. (1) Google (anycast) and Ubuntu (unicast) return the same four addresses to every client (Table 3), providing four sources as RFC 8633 recommends. (2) Meta and Cloudflare return a single and two anycast addresses, respectively: BGP selects the site, and the client has only one or two sources [84, §7]. Since neither group's answers depend on the client, one query suffices to count its sources. (3) Microsoft, Apple, and AWS return a location-dependent subset of their 12 to 90 addresses (§4.1), so the number can vary by client.

We analyze this group: for each VP, we count the distinct addresses it receives across 72 queries (one every 10 minutes for 12 hours, sent directly to the authoritative servers; Table 5). Figure 5 shows the results for these providers. AWS serves each client with 10 addresses, whereas Apple serves most with five. Microsoft, however, serves half of our VPs with a single time source, violating RFC 8633. This reduces reliability, given those clients have no redundancy in their time source. Considering that Windows has 1.4 billion active installations, the possibility of so many installations depending on a single source is quite alarming.

To determine if these VPs are bound to specific regions, we show the geolocation of VPs served by a single IP address in Figure 6. We see that they are not constrained to any specific regions. We notified Microsoft about the issue on 2025-05-02 and received no response.



(a) Microsoft (b) Apple (`time.g.aaplimg.com`) (c) AWS

Fig. 7. Number of responses containing each NTP server address for each time provider.

4.3 Load balancing clients

Unicast providers can also use DNS to balance load among the servers in a region, by returning some addresses more often than others. (For anycast providers, BGP determines the load per site.) For Microsoft, Apple, and AWS, we count how often each address appears in the DNS responses of Table 5 (Figure 7). Because our VPs are concentrated in Europe and North America (§2.2), absolute counts are biased by where our VPs are; we therefore compare only servers that serve the same VPs, i.e., within a region.

Microsoft: Figure 7a shows the client distribution for Microsoft's servers. We show the NTP server's geolocation above each bar. We see that the first two servers are returned in the same frequency – both located in the Netherlands. Even if Microsoft has two other servers in Europe (in the United Kingdom), the Netherlands-based ones are returned more often. With regard to the US, we see that one server (`168.61.215.74`) is returned more often than the other. This demonstrates that Microsoft prioritizes some servers over others, i.e., using DNS for load balancing, at least for our set of VPs.

Apple weights servers within a region unevenly. Among servers returned to US VPs, the most frequent one appears 4× as often as the least frequent (48k vs. 12k responses); among European servers, 3.6× (155k vs. 43k; Appendix B). We see that stratum 1 servers are served more often than stratum 2 (Figure 7b).

AWS returns all 10 servers of a location equally often (Figure 7c): DNS selects the location, and load is spread evenly within it. Of the two US locations, Virginia appears 90× as often as Oregon.

4.4 Discussion

Except Ubuntu, all unicast providers adopt the NTP Pool's model: DNS maps clients by location, to in-country servers where available. They diverge in what the mapping gives each client. Ubuntu and AWS give every client at least four sources, as RFC 8633 recommends; Apple does so for most; Microsoft, the default for 1.4B Windows devices, gives more than 50% of our VPs a single one. Unlike the number of servers a provider operates, this choice is made per client, in the DNS. Because a Windows device cannot use more servers than Microsoft provides (no other time servers are configured by default), it cannot detect when that server serves wrong time (as in the outage of Telstra [98, 99]). Microsoft should return at least four addresses per client; it already operates 12 servers and maps clients through DNS, so this requires no new servers. Meta's single anycast address and Cloudflare's two addresses have the same limitation, but they are not an OS default: they should always be configured alongside other time services, not alone.

5 Accuracy and NTP features

Now that we understand the architecture of these time services, we turn to the question of the quality of their service: *How accurate are they*?

Whereas the previous sections relied on DNS measurements, in this section we carry out NTP measurements from multiple VPs synchronized with reference sources (§5.1). We compare the responses to our reference clock to determine their accuracy (§5.2), detect cases of out-of-sync servers (§5.3) and explore the characteristics of services they provide (§5.4). Our measurement data for this section is also openly available [38].

5.1 Experiment setup

To evaluate the accuracy of time services providers, we need to use VPs which are directly connected to time references, having "ground truth" time information. We cannot rely on Atlas probes as VPs, given they have varying methods of updating their clocks [35] but no precise source of time.

Therefore, we set up our own VPs, as shown in Table 6, which are connected to accurate references, and run Ubuntu Linux and `chrony` [25] as a time client on all four VPs, which is a state-of-the-art NTP implementation [45] supporting PTP sources.

VP locations: Our VPs rely on sources that employ both GNSS and atomic clocks. We use three VPs located at AWS in Sweden (SE-AWS), the US (US-AWS), and Japan (JP-AWS), which leverage the AWS time service [2]. To prevent bias from the AWS-based location (given AWS is also a time provider), we set up a fourth, non-AWS VP in the Netherlands (NL-SIDN).

VP	ASN	Time Source	Method	Date
SE-AWS	16509	GNSS, atomic [2]	PHC [2, 83]	2025-05-07
US-AWS	14618			2026-06-03
JP-AWS	16509			
NL-SIDN	1140	GNSS, radio, atomic [95]	Linux PTP [83]	2025-10-29

Table 6. Accuracy Experiment Vantage Points

Time sources: We connect NL-SIDN to a Multi Reference Source (MRS [43]), which takes time input from multiple time sources, in our case GNSS (GPS and Galileo), Radio (DCF77 [13]), and a rubidium atomic oscillator. All VPs are synchronized to the local source using the Precision Time Protocol (PTP), which is typically at least three orders of magnitude more accurate than NTP [39]. For the AWS VPs, we configure `chrony` to use the PTP Hardware Clock (PHC) [83], whereas for NL-SIDN we configure it with Linux PTP [28].

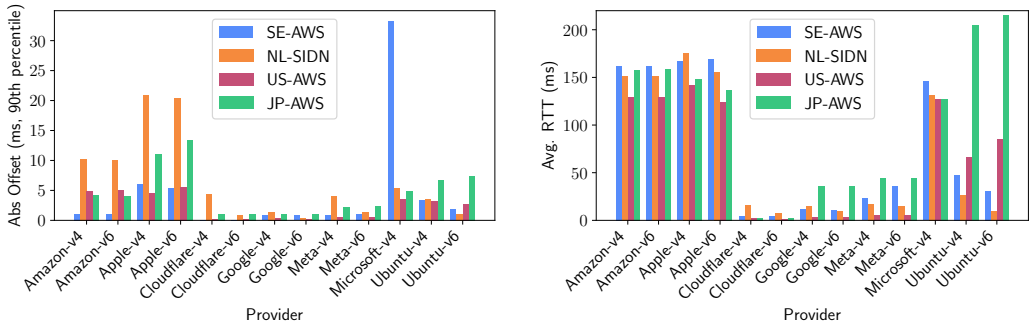
Measurements: We send one NTPv4 query (mode 3, client) to each IP address associated with the time services (Table 3) every 5 minutes for 24h (from midnight to midnight UTC on the dates in Table 6). In total, each VP sends 288 NTP queries to each of the 322 IP addresses over the 24h period. We capture the traffic using `tcpdump` and analyze the traces to compute the offset between our VP clocks and the provider's clocks. The measurements run between May 2025 and June 2026. Because some IP addresses have become inactive (not responding to any NTP queries), NL-SIDN, US-AWS and JP-AWS have reduced numbers of IP addresses for Amazon and Apple (see Appendix C), which seem to routinely change their server addresses.

Limitations: Our experiment has two limitations. First, every VP can only reach a single site (location) of each anycast provider (Google, Cloudflare, and Meta), assuming routing does not change during the measurements [103]. Therefore, we can only partially evaluate these services. (For non-anycast services, we measure all locations by measuring all IP addresses.)

Second, while not an inherent limitation of our experiments, NTP assumes symmetric network paths [58]. However, most Internet paths are *asymmetric* [101], often exhibiting different one-way delays ($T_2 - T_1 \neq T_4 - T_3$ in Figure 1). Since the NTP clock offset computation (Figure 1) relies on an equal-delay assumption by averaging the timestamps, persistent delay asymmetries result in systematic estimation errors in the reported offsets [63, 75]. We reduce the impact of this by using multiple vantage points and assuming the maximal error in §5.3.

Provider	IP ver.	#IP Addr.	#Req.	#Resp.	No Resp. #	No Resp. %	Offset mean	Abs. Offset 90th %ile	RTT mean	NTP version	Stratum
Amazon	v4	89	25 632	25 608	24	0.09%	0.05 ms	1.03 ms	162.05 ms	v4	4
	v6	89	25 632	25 612	20	0.08%	0.04 ms	1.04 ms	162.08 ms	v4	4
Apple	v4	51	14 688	14 058	630	4.29%	0.50 ms	6.12 ms	166.54 ms	v4	1/2 (78%/22%)
	v6	46	13 248	12 253	995	7.51%	0.80 ms	5.44 ms	168.75 ms	v4	1/2 (77%/23%)
Cloudflare	v4	2	576	576	0	0.00%	-0.01 ms	0.06 ms	3.97 ms	v4	3
	v6	2	576	576	0	0.00%	0.01 ms	0.05 ms	3.95 ms	v4	3
Google	v4	4	1 152	1 152	0	0.00%	0.02 ms	0.79 ms	11.26 ms	v4	1
	v6	4	1 152	1 152	0	0.00%	-0.02 ms	0.79 ms	11.12 ms	v4	1
Meta	v4	5	1 440	1 440	0	0.00%	0.04 ms	0.90 ms	23.02 ms	v4	1
	v6	5	1 440	1 440	0	0.00%	-0.04 ms	1.07 ms	36.24 ms	v4	1
Microsoft	v4	12	3 456	3 450	6	0.17%	-13.75 ms	33.25 ms	145.65 ms	v3	3
Ubuntu	v4	4	1 152	1 151	1	0.09%	0.34 ms	3.30 ms	47.45 ms	v4	2
	v6	3	864	864	0	0.00%	1.38 ms	1.85 ms	31.01 ms	v4	2

Table 7. SE-AWS measurement details. Anycasted services are highlighted in orange.



(a) 90th-percentile absolute offset

(b) Average RTT

Fig. 8. Accuracy and RTT results per provider.

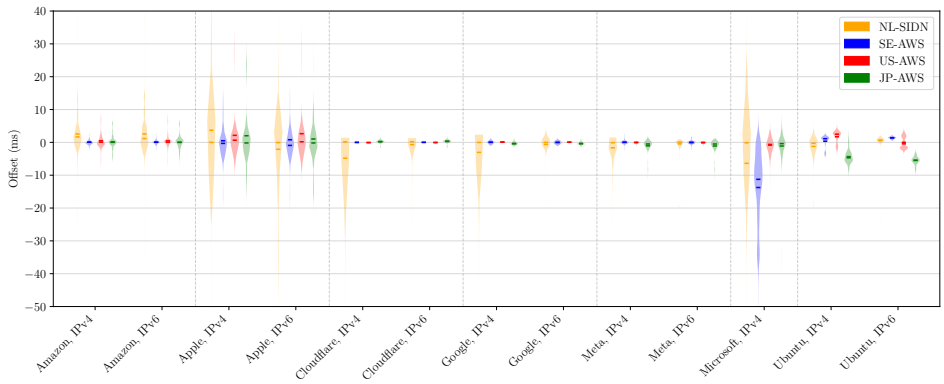
5.2 Accuracy results

Table 7 summarizes our results for the vantage point SE-AWS. We include the same details for the other VPs in Appendix C and discuss any differences in this section.

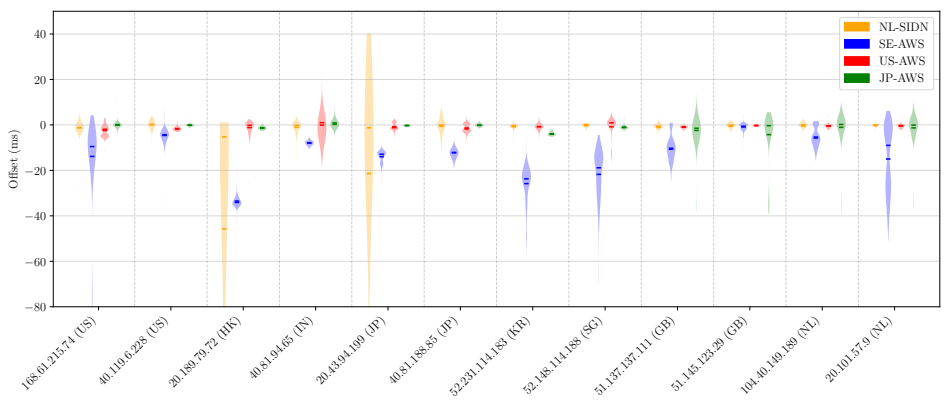
We evaluate *accuracy* by measuring the clock offset between our VPs and the time service providers (Figure 1). An offset of zero indicates perfect synchronization between the providers' clocks and our own ("ground truth", to within the uncertainty introduced by propagation delays).

In Figure 8a, we show 90th-percentile absolute offset for each provider and both VPs (using absolute values prevents positive and negative values from evening each other out). We see that from each VP, all offsets' 90%iles are under 21ms, which can be considered good for most applications, except for Microsoft during the measurement from SE-AWS (33.25ms). (We include a figure of the mean absolute offsets in Appendix B, which shows the same pattern).

Accuracy per VP: Comparing VPs, we see that the AWS vantage points consistently have offsets closer to 0 than NL-SIDN (except for Microsoft from SE-AWS, and Ubuntu), despite having mostly larger average RTTs for the non-anycast providers (Figure 8b). This might be due to better network conditions at the AWS vantage points, especially for the anycasted providers and AWS time servers. Google and Meta offer nearby anycast sites in the US (Figure 8b) (<5ms avg. RTT), but not in Japan (>30ms), which similarly has no nearby Ubuntu servers (≈200ms).



(a) Per organization



(b) Microsoft servers

Fig. 9. Violin plots of offset distributions. Positive offset indicates that the polled server's clock is ahead of our synchronized reference clock.

We see significant packet loss for Apple (>4% each measurement) and Microsoft (>2%, except during the first measurement). Amazon does not suffer from this even measured from NL-SIDN (Table 15), despite also using DNS-based replication (§3).

Anycast: For all VPs, we see how anycast providers (Google, Cloudflare and Meta) have a lower RTT, which is due to us only reaching a single site, whereas we measured all locations for the others, which are globally distributed (§3.2).

Offsets distribution per provider: Figure 9 shows the offset distribution. We see a clear difference between the same providers for the VPs – NL-SIDN has a more dispersed offset for the same providers – including anycast providers, even though it has a lower RTT to many providers.

Microsoft: We zoom in on Microsoft's servers (Figure 9b). We see that for most servers, the later measurements (NL-SIDN, US-AWS, and JP-AWS) generally have a distribution that is more compact and close to 0 than SE-AWS. The exceptions are: two servers from NL-SIDN (one in JP and other in HK) as well as JP-AWS combined with European time servers. For our first measurement (SE-AWS), Microsoft's time servers had predominantly negative offsets, meaning that the server's clock is

lagging behind our reference clock. This will impact the accuracy for clients because Microsoft offers only one server to most of them (§4.2). The significant differences for Microsoft are likely due to the different measurement times, while the large spread in offsets for two of the servers as measured from NL-SIDN might be due to large variances in network delays to those specific servers in Asia. During the last measurement (from US-AWS and JP-AWS), some of Microsoft's servers report to run at stratum 4 (Table 15), and we see significantly improved accuracy (Figure 9b).

RTT and accuracy: Mathematically, lower RTT improves worst-case accuracy of the offset (Appendix D). This is also part of the reason for geo-based mappings in the NTP Pool. Our results, however, show that it depends on the VP. SE-AWS had a weak Pearson correlation between absolute offset and RTT (0.38), whereas NL-SIDN had a strong one (0.75). Therefore, we conclude that higher RTT is not always the most relevant factor for accuracy, and the VP location also plays a role. This makes intuitive sense: inaccuracies are not caused by RTT, but by asymmetric network latency. We see that despite us measuring all sites, 90% of offsets for AWS are under 2ms for SE-AWS and 5ms for US- and JP-AWS (possibly due to bias given both the VP and provider are the same company), but not for NL-SIDN: here Apple and AWS delivered significantly larger absolute offsets (Appendix C).

Stratum and accuracy: Cloudflare's stratum 3 servers consistently provide similar or more accurate results than Google's stratum 1 (Figure 9). This visualizes that a lower stratum is not the most relevant factor influencing accuracy, either.

5.3 Detecting out-of-sync servers

Given that our VPs maintain ground-truth time, any offsets we observe are errors. Such errors can arise from two sources: variable network delay or clock errors on the provider's side. Because the measured offset combines the server's true offset with an error introduced by network delay, an offset $\neq 0$ does not necessarily mean that the server's clock is incorrect. Moreover, without prior clock synchronization, it is generally not possible to distinguish between delay asymmetry and an actual clock error.

However, the contribution of network delay to the offset is limited by the RTT. For two synchronized clocks, the offset must fall within $\pm RTT/2$ (Appendix D, [59]). If our measured offset falls within this delay-dependent bound, then network latency is sufficient to explain the deviation, and the server's clock may be correct. However, offsets that exceed this bound cannot be explained by latency alone and therefore indicate a clock error.

In our measurements, almost all offsets fall within the delay-derived bound, indicating that network latency dominates over clock errors in the measurement. However, 199 responses from Microsoft servers to SE-AWS violate this, meaning the server's clocks are incorrect. Figure 10 shows the offsets received during our measurement. The colored area marks the maximum offset that network latency can cause. Square points are offsets that fall outside the bounds, indicating that the server's clock is incorrect (the offset cannot be explained by latency alone). The shape of the curve suggests that these servers have synchronized to an incorrect time and slowly approach 0 again. We provide further statistics about the three affected servers in Appendix C (all of which are in Western Europe, with low latency to SE-AWS).

During the measurements from the other three VPs, we see no further errors from Microsoft. While we also measure some of Cloudflare's and Meta's servers to be in error briefly (from JP-AWS and US-AWS, respectively), this never exceeds 2ms (10μs for Meta). Because inaccuracy increases with RTT, we may miss other clock errors that are overshadowed by the RTT.

5.4 Time services profiling

Next we determine other characteristics of the NTP services providers based on the responses we collected.

NTP version and IPv6 support: As shown in Table 7, all services deploy NTPv4, which is the current specification, standardized in 2010 [58]. Only Microsoft still deploys NTPv3, standardized in 1992 [57]. Moreover, Microsoft is the only provider which does not support IPv6.

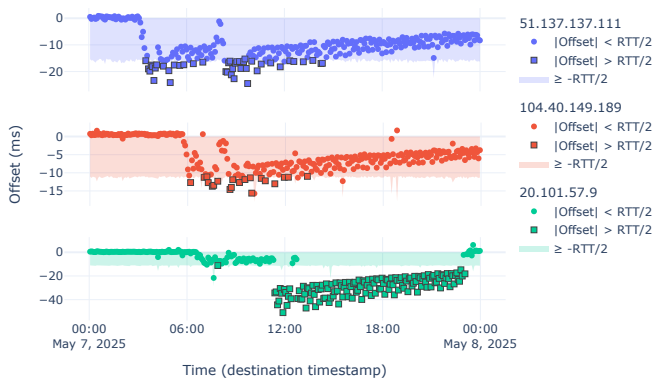


Fig. 10. Offsets of out-of-sync Microsoft servers from SE-AWS. Points with |Offset| > RTT/2 indicate the server must be out-of-sync.

DNSSEC and RPKI support: The Domain Name System Security Extensions (DNSSEC [4]) add cryptographic signatures to DNS data to ensure its authenticity and integrity, protecting against spoofing and cache poisoning, which can protect NTP clients and others from man-in-the-middle attacks. It is used by DNS resolvers. Resource Public Key Infrastructure (RPKI [48]) provides cryptographic signatures that prove whether an IP prefix may be announced by any given AS, which helps to prevent BGP hijacks from false route announcements. Table 8 provides an overview: All providers except for Apple and Ubuntu use RPKI, while DNSSEC is only used by Cloudflare.

NTS support: As of the writing of this paper, only Cloudflare [21] and Ubuntu support NTS. Ubuntu started support in 2025 [54] and made it the default in Ubuntu 25.10 [15]. The other 5 providers do not support NTS. Neither does NIST [79] nor the NTP Pool (NTS pools would require sharing certificates [70]).

Provider	RPKI	DNSSEC	NTS
Amazon	Yes	No	No
Apple	No	No	No
Cloudflare	Yes	Yes	Yes
Google	Yes	No	No
Meta	Yes	No	No
Microsoft	Yes	No	No
Ubuntu	No	No	Yes

Table 8. Security standards implemented by each time provider

Stratum: The time service providers use a variety of strata (Table 7), from 1 to 4. Only Google, Meta and (80% of) Apple's servers are stratum 1, although we could not measure all anycast sites of the first two. The remaining are all stratum 2 to 4 (Table 7).

Reference IDs: NTP responses include a reference ID, describing the time source. We classify the reference IDs obtained and show them in Figure 16 (Appendix B).

We find that 5 of the 7 providers return reference ID values that do not communicate the upstream time source. Google, for instance, returns "GOOG", Meta returns Airport codes (e.g., LHR6), while Cloudflare, AWS, Microsoft return private IP addresses. Only Apple and Ubuntu return references that provide information about their source. However, Meta has published information online about their time services, including the time source: GNSS [78]. Apple's stratum 1 servers all return descriptive strings, the majority returning `GPSs` (for: GPS via Shared Memory), and few returning `SHM` or `MRS` (for Shared Memory and likely Multi-reference Source), the two of which are generic technologies and do not describe a specific time source [43]. Apple's stratum 2 servers return IP addresses of Apple stratum 1 servers. Ubuntu, in turn, provides the IP addresses of their upstream sources. One of Ubuntu's servers is synchronized with one of Apple's stratum 1 servers, thus

creating a dependency between both services. Another Ubuntu server synchronized with NIST's server at the time.

Server software fingerprinting: We attempt to fingerprint the software used by each provider by analyzing the NTP version field returned in the responses. First, we run the NTP server software shown in Table 9, varying the version field in the NTP request, and we observe the responses in order to obtain fingerprints of common server software.

		NTP Version								
Daemon	Version	v0	v1	v2	v3	v4	v5	v6	v7	Q.
chronyd [25]	4.7	-	v1	v2	v3	v4	-	-	-	Response
ntpd [69]	4.2.8	-	v1	v2	v3	v4	v5	v6	v7	
ntpd-rs [82]	1.6.0	-	-	-	v3	v4	-	-	-	
ntpsec [77]	1.2.4	-	v1	v2	v3	v4	-	-	-	
openntpd [11]	6.8p1	-	v1	v2	v3	v4	v5	v6	v7	

Table 9. NTP version of queries and responses.

Then, we send NTP queries to each service provider varying the NTP version field. Table 10 shows the results. Our results show that Apple (in part) and Google may be using ntpd or OpenNTPD. Amazon, Cloudflare, Meta and Ubuntu use software that behaves similar to Chrony or NTPsec. There are public reports of Meta using Chrony [78]. Ubuntu uses Chrony by default [54], although Canonical has announced plans to replace it with ntpd-rs [82, 93]. We could not find matches for Microsoft and part of Apple's servers.

	NTP Version								
Provider	v0	v1	v2	v3	v4	v5	v6	v7	Query
Apple	v0	v1	v2	v3	v4	v4	v4	v4	Response
	-	v1	v2	v3	v4	v5	v6	v7	
Amazon	-	v1	v2	v3	v4	-	-	-	
Cloudflare	-	v1	v2	v3	v4	-	-	-	
Google	-	v1	v2	v3	v4	v5	v6	v7	
Meta	-	v1	v2	v3	v4	-	-	-	
Microsoft	-	v3	v3	v3	v3	-	-	-	
Ubuntu	-	v1	v2	v3	v4	-	-	-	

Table 10. NTP version responses per provider

6 Discussion and Recommendations

A handful of OS vendors are the default time source for billions of devices, yet, unlike the NTP Pool, their services had not been measured. Our measurements yield three lessons, each with an action for the party that controls it.

Lesson 1: Sources per client, not servers per provider, determine resilience. Providers differ by an order of magnitude in size, from 4 to 90 addresses, but this says little about what a client receives. Microsoft operates 12 servers, yet restricts more than 50% of our VPs to using only one (§4.2). RFC 8633 recommends at least four sources [84, §3.2], and a client with one cannot detect a faulty server.

Lesson 2: Providers differ in faults, not in typical accuracy, and the one with faults is the one whose clients cannot detect them. For all providers except Microsoft, the 90th-percentile offset was below 21 ms, and almost all offsets fell within the error that network delay alone can cause (§5.2). Microsoft was the exception: three of its servers served time significantly off (one by 40-50 ms) for several hours, beyond what delay can explain (§5.3). The same provider lets 50% of our VPs synchronize to a single source (Lesson 1), so those clients had no means to detect the error. Telstra's outage [98, 99] shows the same, but with a larger error.

What accuracy do applications need? Table 11 compares the errors we measured with tolerances of common example applications. Security protocols tolerate both typical offsets and the maximum ones we have measured for Microsoft; only errors of longer periods (≥30 s up to years), as in the USNO and Telstra incidents, break them. Clients configured with a single source cannot reject such errors (Lesson 1). For applications that need accuracy of a millisecond or less, NTP over the Internet is insufficient. They must rely on local reference sources and/or more precise protocols such as PTP (§2.1).

Tolerance	Applications	Error		
		≤21 ms (typical)	50 ms (Microsoft)	Years (USNO, Telstra)
Hours+	TLS, DNSSEC validity periods	ok	ok	fails
Minutes	Kerberos (5 min) [62]	ok	ok	fails
Seconds	TOTP (30 s) [68]	ok	ok	fails
1 ms–100 µs	EU algorithmic trading [29]	fails	fails	fails
1.5 µs	TDD mobile (LTE, NR) [42]	fails	fails	fails

Table 11. Clock-error tolerance of applications, against the errors we measured and the USNO and Telstra incidents.

Lesson 3: Security standards lag years behind. Six years after its standardization, only Cloudflare and Ubuntu support NTS; only Cloudflare signs its DNS zone with DNSSEC; Apple and Ubuntu do not use RPKI; and Microsoft serves NTPv3 (1992), not supporting IPv6 (§5.4). We make the following recommendations:

- *Microsoft* should return at least four addresses per client, which requires no new servers, and support NTPv4 and IPv6. *Cloudflare and Meta* should likewise provide at least four servers for clients to synchronize with. While Meta provides 5 individual domain names and IP addresses, using `time.facebook.com` (or any single name) will mean a client synchronizes to only one IP address.
- *OS vendors* should ship clients that combine sources from more than one provider, and enable NTS by default – like Ubuntu already does.
- *All providers* should deploy NTS, DNSSEC, and RPKI; derive time from diverse references, including non-GNSS sources, given the jamming that motivates national efforts such as Norway's [40]; and publish their deployments and upstream sources (which includes using reference IDs in their NTP responses).

7 Ethics and Disclosure

Our paper has two ethical concerns: minimizing the impact of our measurements and notifying vendors of inaccuracies and issues we have found. Most of our measurements use RIPE Atlas and query at a low frequency (every 10 min), which cannot stress the authoritative DNS servers or NTP servers that we measure – they are provisioned to answer to billions of clients.

Secondly, we found inconsistencies in some vendors, and reported them following coordinated vulnerability disclosure guidelines [67]. We have reported to Apple that one of its time servers was sending stale reference timestamps (but precise timestamps, meaning likely no impact on clock synchronization), and they have fixed the issue. We also reported to Microsoft that their DNS servers return only one NTP server to most clients, which violates RFC 8633 [84]. We have later informed them that we will make this information public 45 days from the original notification. We also notified Meta that their domain names return 1 IP address only – and it would be beneficial to return multiple ones. We have not obtained a response. We then notified them again about the public disclosure of our findings.

8 Related Work

The closest study to ours investigates the NTP Pool architecture, client/server mapping [65], and accuracy from a subset of RIPE Atlas VPs. Different than this, we analyze seven providers and carry out accuracy experiments from four VPs synced with reference clocks. [92] analyzed some of the

same vendors (Apple, Canonical, Google and Microsoft next to Alibaba and NIST) on the side, but did not analyze their underlying replication or features. Our results show that the infrastructure has changed significantly: We find more IP addresses for Apple (101 vs. 8, counting IPv4 and IPv6) and Microsoft (12 vs. 9), with Apple using a geographical DNS mapping. Google's servers now run at stratum 1 (from 2), while Microsoft's changed to 3 and 4 (from 2). Since Microsoft's upstreams are private IP addresses, they might still rely on NIST servers. The Ubuntu OS is now using Chrony instead of timesyncd [54]. Many aspects remain the same: Apple still has stratum 1 and 2 servers relying mostly on GPS, Ubuntu has the same number of IP addresses as before, Google's DNS configuration did not change, and Microsoft has no reverse pointers for their NTP servers. [92] describes an outage of Microsoft's servers, while we only found them to have subpar accuracy.

NTP Pool: The NTP Pool has been studied extensively [46, 65, 81, 92], but so far the same analysis has not been done for other large time service providers, despite them being the default providers for billions of devices. We cover them in this paper.

IP-wide scans: Multiple studies have focused on mapping NTP servers in the entire IP space [37, 60, 61, 92]. Considering the user base of the largest time providers (estimated in [65]), we argue that it is more important to focus on scrutinizing these – which include those in this paper and the NTP Pool.

Accuracy: A previous study also used a GPS-synced VP with an atomic holdover clock to measure clock errors of stratum 1 NTP servers [16]. They perform in-depth analysis of the accuracy of a different server population than ours.

NTP security: Protocol vulnerabilities have also been investigated [10, 51–53, 100], demonstrating how NTP clients can be susceptible to malicious time servers [27, 31] or to attacks by third parties [51], often focusing on broadcast-mode availability [51, 100]. Moreover, usage of NTP for amplification of distributed denial-of-service (DDoS) attacks [26] has been covered, as well as off-path attacks using DNS cache poisoning or BGP hijacking [44, 51].

9 Conclusion

NTP is a fundamental but often overlooked service on the Internet. Compared to the DNS, NTP services are significantly more centralized, as most devices use default (and sometimes hardcoded) configuration settings in their OSes.

We measured the time services of seven providers, including the defaults of Windows, macOS/iOS, Android, and Ubuntu. The unicast providers map clients by location, but all providers differ in how many sources each client receives. Microsoft gives more than half of our VPs a single server, and three of its servers served inaccurate time for hours: a fault its clients had no way to detect.

These problems are easy to fix. Microsoft already operates 12 servers; returning four of them per client is a DNS configuration change. NTS, DNSSEC, and RPKI are standardized, and Cloudflare already deploys all three. The providers have the infrastructure; they need only use it.

We hope our findings prompt these vendors to make these changes, and thereby improve the resilience of the services that depend on their time.

Acknowledgements

The work of Pascal Huppert and Cristian Hesselman was partially funded by the INTERSECT project, supported by the Dutch Research Council (NWO) under grant NWA.1160.18.301. Giovane C. M. Moura and Marco Davids were supported by SIDN through the Time.nl project. Georgios Smaragdakis was supported by the European Union under the Horizon Europe Programme as part of the project SafeHorizon (Grant Agreement #101168562). We are grateful to the reviewers for their valuable feedback and the time invested in evaluating our manuscript.

References

[1] S. Alexander and R. Droms. 1997. *DHCP Options and BOOTP Vendor Extensions.* RFC 2132. IETF. http://tools.ietf.org/rfc/rfc2132.txt

[2] Amazon Web Services. 2024. *Precision clock and time synchronization on your EC2 instance.* https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/set-time.html Accessed: 2025-10-27.

[3] Apple Inc. 2024. Apple Reports First Quarter Results. https://www.apple.com/newsroom/2024/02/apple-reports-first-quarter-results/. Accessed: 2025-02-18.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. *DNS Security Introduction and Requirements.* RFC 4033. IETF. http://tools.ietf.org/rfc/rfc4033.txt

[5] R. Austein. 2007. *DNS Name Server Identifier (NSID) Option.* RFC 5001. IETF. http://tools.ietf.org/rfc/rfc5001.txt

[6] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. 2015. Lessons Learned from using the RIPE Atlas Platform for Measurement Research. *SIGCOMM Comput. Commun. Rev.* 45, 3 (July 2015), 35–42. http://www.sigcomm.org/sites/default/files/ccr/papers/2015/July/0000000-0000005.pdf

[7] Robert Beverly and Erik Rye. 2026. On Borrowed Time: Measurement-Informed Understanding of the NTP Pool's Robustness to Monopoly Attacks. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium 2026*. Internet Society, San Diego, CA, USA. https://doi.org/10.14722/ndss.2026.240541

[8] bgptools. 2025. anycast-prefixes. https://github.com/bgptools/anycast-prefixes Accessed: 2025-02-20.

[9] Leo Bicknell. 2012. NTP Issues Today. https://lists.nanog.org/archives/list/nanog@lists.nanog.org/message/HIROUHZQ3HGIHMJU65GCN7T36A6OOF2O/.

[10] M. Bishop. 1990. A security analysis of the NTP protocol version 2. In *[1990] Proceedings of the Sixth Annual Computer Security Applications Conference*. 20–29. https://doi.org/10.1109/CSAC.1990.143746

[11] Henning Brauer. 2022. *OpenNTPd.* https://www.openntpd.org Accessed: 2025-05-13.

[12] Physikalisch-Technische Bundesanstalt. 2025. *Questions about Time.* https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/fragenzurzeit/fragenzurzeit07.html Accessed: 2025-05-01.

[13] Physikalisch-Technische Bundesanstalt. n.d.. DCF77. https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/ag-442/dissemination-of-legal-time/dcf77.html. Accessed: 2025-11-06.

[14] R. Bush and R. Austein. 2013. *The Resource Public Key Infrastructure (RPKI) to Router Protocol.* RFC 6810. IETF. http://tools.ietf.org/rfc/rfc6810.txt

[15] Canonical. 2025. Ubuntu 25.10 release notes. https://documentation.ubuntu.com/release-notes/25.10/. Accessed: 2026-10-10.

[16] Yi Cao and Darryl Veitch. 2020. Toward Trusted Time: Remote Server Vetting and the Misfiring Heart of Internet Timing. *IEEE/ACM Transactions on Networking* 28, 2 (2020), 944–956. https://doi.org/10.1109/TNET.2020.2977024

[17] RIPE Network Coordination Centre. 2025. *K-Root.* https://www.ripe.net/analyse/dns/k-root/ Accessed: 2025-03-13.

[18] RIPE Network Coordination Centre. 2025. *RIPE Atlas Probe Archive.* https://ftp.ripe.net/ripe/atlas/probes/archive/ Accessed: 2025-02-12.

[19] CERN. 2024. *White Rabbit.* CERN Knowledge Transfer. https://kt.cern/white-rabbit/ Accessed: 2026-10-08.

[20] Danilo Cicalese, Diana Joumblatt, Dario Rossi, Marc-Olivier Buob, Jordan Augé, and Timur Friedman. 2015. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2776–2784. https://doi.org/10.1109/INFOCOM.2015.7218670

[21] Cloudflare. 2024. Cloudflare Time Service. https://www.cloudflare.com/time/. Accessed: 2026-10-08.

[22] Internet Systems Consortium. 2025. *F-Root.* https://www.isc.org/f-root/ Accessed: 2025-03-13.

[23] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari. 2016. *Client Subnet in DNS Queries.* RFC 7871. IETF. http://tools.ietf.org/rfc/rfc7871.txt

[24] Alex Cranz. 2021. *Google announces that Android has over 3 billion active devices.* https://www.theverge.com/2021/5/18/22440813/android-devices-active-number-smartphones-google-2021 Accessed: 2025-02-18.

[25] Richard Curnow and Miroslav Lichvar. 2024. *chrony - Network Time Protocol (NTP) Implementation.* https://chrony-project.org/ Accessed: 2025-03-04.

[26] Jakub Czyz, Michael Kallitsis, Manaf Gharaibeh, Christos Papadopoulos, Michael Bailey, and Manish Karir. 2014. Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks. In *Proceedings of the 2014 ACM Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC)*. ACM, 435–448. https://doi.org/10.1145/2663716.2663717

[27] Omer Deutsch, Neta Rozen Schiff, Danny Dolev, and Michael Schapira. 2018. Preventing (Network) Time Travel with Chronos.. In *Network and Distributed Systems Security (NDSS) Symposium 2018*. https://doi.org/10.14722/ndss.2018.23231

[28] Linux PTP development team. 2025. *LinuxPTP – Precision Time Protocol (PTP) implementation for Linux.* https://linuxptp.sourceforge.net/ Accessed: 2025-11-05.

[29] European Commission. 2017. Commission Delegated Regulation (EU) 2017/574 of 7 June 2016 supplementing Directive 2014/65/EU with regard to regulatory technical standards for the level of accuracy of business clocks. Official Journal of the European Union, L 87, 31.3.2017, pp. 148–151. https://eur-lex.europa.eu/eli/reg_del/2017/574/oj/eng Accessed: 2026-10-08.

[30] D. Franke, D. Sibold, K. Teichel, M. Dansarie, and R. Sundblad. 2020. *Network Time Security for the Network Time Protocol.* RFC 8915. IETF. http://tools.ietf.org/rfc/rfc8915.txt

[31] Tweede Golf. 2024. Manipulating time through (S)NTP. https://tweedegolf.nl/en/blog/142/manipulating-time-through-sntp Accessed: 2025-11-18.

[32] Google. 2025. Network edge locations. https://cloud.google.com/about/locations/. Accessed 15-05-2025.

[33] Ask Bjørn Hansen. 2023. Minor New Features on the website. https://community.ntppool.org/t/minor-new-features-on-the-website/2947/8.

[34] Remi Hendriks, Matthew Luckie, Mattijs Jonker, Raffaele Sommese, and Roland van Rijswijk-Deij. 2025. LACeS: An Open, Fast, Responsible and Efficient Longitudinal Anycast Census System. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25).* 445–461. https://doi.org/10.1145/3730567.3764484

[35] Philip Homburg. 2015. NTP Measurements with RIPE Atlas. https://labs.ripe.net/author/philip_homburg/ntp-measurements-with-ripe-atlas/.

[36] Nate Hopper. 2022. The Thorny Problem of Keeping the Internet's Time. *The New Yorker* (Sept. 30 2022). https://www.newyorker.com/tech/annals-of-technology/the-thorny-problem-of-keeping-the-internets-time

[37] Zhentian Huang, Shuai Wang, Li Chen, Dan Li, and Yilun Liu. 2025. Measuring the Time Source Vulnerabilities in the NTP Ecosystem. In *Proceedings of the 2025 ACM Internet Measurement Conference* (Madison, WI, USA) *(IMC '25).* Association for Computing Machinery, New York, NY, USA, 493–510. https://doi.org/10.1145/3730567.3764478

[38] Pascal Huppert, Giovane C.M. Moura, Marco Davids, Pieter-Tjerk de Boer, Ralph Holz, Rein Fernhout, Georgios Smaragdakis, and Cristian Hesselman. 2026. BigTime NTP Server Feature and Accuracy Measurement Data. https://doi.org/10.4121/9cd3c5d2-0703-4370-a5bf-d519cb9de516

[39] IEEE. 2020. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)* (2020), 1–499. https://doi.org/10.1109/IEEESTD.2020.9120376

[40] Inside GNSS. 2026. *Norway Proposes NOK 94 Million for National Time Service to Cut GPS Dependence.* https://insidegnss.com/norway-proposes-nok-94-million-for-national-time-service-to-cut-gps-dependence/ Accessed: 2026-10-05.

[41] IPinfo.io. 2025. *IPinfo - Trusted IP Data Provider.* https://ipinfo.io Accessed: 2025-02-12.

[42] ITU-T. 2020. *Time and phase synchronization aspects of telecommunication networks.* Recommendation G.8271/Y.1366. International Telecommunication Union.

[43] Andreja Jarc. 2017. *Multi-Reference Clock.* https://blog.meinbergglobal.com/2017/11/16/multi-reference-clock/ Accessed: 2025-11-10.

[44] Philipp Jeitner, Haya Shulman, and Michael Waidner. 2020. The Impact of DNS Insecurity on Time. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).* 266–277. https://doi.org/10.1109/DSN48063.2020.00043

[45] S. Konjerla, G. C. M. Moura, G. Smaragdakis, and T. Dittrich. 2025. *Are NTP Clients Always Right? Evaluating NTP Clients under Normal and Attack Scenarios.* Technical Report SIDN Labs Technical Report 2025-10-16. SIDN Labs and TU Delft, Arnhem, The Netherlands. https://www.sidnlabs.nl/downloads/7M1bz1otGu0D7hqr0hRT2c/3bc09536c3bc87f63b80d66944abc1d9/ntp-clients-tech_report-20251016.pdf Updated October 30, 2025.

[46] Jonghoon Kwon, Jeonggyu Song, Junbeom Hur, and Adrian Perrig. 2023. Did the Shark Eat the Watchdog in the NTP Pool? Deceiving the NTP Pool's Monitoring System. In *32nd USENIX Security Symposium (USENIX Security 23).* USENIX Association, Anaheim, CA, 6151–6166. https://www.usenix.org/conference/usenixsecurity23/presentation/kwon

[47] leeand00. 2019. *Is it true that iOS devices use a hard coded* `time.apple.com` *DNS for an NTP server?* https://apple.stackexchange.com/questions/364765/is-it-true-that-ios-devices-use-a-hard-coded-time-apple-com-dns-for-an-ntp-serve Question on Ask Different (Stack Exchange).

[48] M. Lepinski and S. Kent. 2012. *An Infrastructure to Support Secure Internet Routing.* RFC 6480. IETF. http://tools.ietf.org/rfc/rfc6480.txt

[49] Miroslav Lichvar and Tal Mizrahi. 2026. *Network Time Protocol Version 5.* Internet-Draft draft-ietf-ntp-ntpv5-09. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-ntp-ntpv5/09/ Work in Progress.

[50] Maciej Lipinski, Tomasz Wlostowski, Javier Serrano, and Pablo Alvarez. 2011. White Rabbit: A PTP Application for Robust Sub-Nanosecond Synchronization. In *Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS).* 25–30. https://doi.org/10.1109/ISPCS.2011.6070148

[51] Aanchal Malhotra, Isaac E Cohen, Erik Brakke, and Sharon Goldberg. 2016. Attacking the Network Time Protocol. In *Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS 2016)* (San Diego, California).

[52] Aanchal Malhotra and Sharon Goldberg. 2016. Attacking NTP's Authenticated Broadcast Mode. *SIGCOMM Comput. Commun. Rev.* 46, 2 (may 2016), 12–17.

[53] Aanchal Malhotra, Matthew Van Gundy, Mayank Varia, Haydn Kennedy, Jonathan Gardner, and Sharon Goldberg. 2017. The Security of NTP's Datagram Protocol. In *Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21*. Springer, 405–423.

[54] Lukas Märdian. 2025. PSA: Installing chrony by default, to enable Network Time Security (NTS). Ubuntu-devel mailing list. https://lists.ubuntu.com/archives/ubuntu-devel/2025-May/043355.html Message posted on May 22, 2025, Accessed: 2026-05-02, Archived version: https://web.archive.org/web/20260502175853/https://lists.ubuntu.com/archives/ubuntu-devel/2025-May/043355.html.

[55] Microsoft. 2022. Windows Devices by the Numbers. https://news.microsoft.com/bythenumbers/en/windowsdevices. Accessed: 2025-02-18, Archived version: https://web.archive.org/web/20220419050729/https://news.microsoft.com/bythenumbers/en/windowsdevices.

[56] Microsoft. 2025. *Traffic Manager Geographic Traffic Routing Method.* https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-faqs#traffic-manager-geographic-traffic-routing-method Accessed: 2025-05-02.

[57] D. Mills. 1992. *Network Time Protocol (Version 3) Specification, Implementation and Analysis.* RFC 1305. IETF. http://tools.ietf.org/rfc/rfc1305.txt

[58] D. Mills, J. Martin, J. Burbank, and W. Kasch. 2010. *Network Time Protocol Version 4: Protocol and Algorithms Specification.* RFC 5905. IETF. http://tools.ietf.org/rfc/rfc5905.txt

[59] David L Mills. 1991. Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions on Communications* 39, 10 (1991), 1482–1493. https://doi.org/10.1109/26.103043

[60] David L. Mills, Ajit Thyagarjan, and Brian C. Huffman. 1997. Internet Timekeeping Around the Globe. https://www.eecis.udel.edu/~mills/database/papers/survey5.pdf

[61] Nelson Minar. 1999. A Survey of the NTP Network. https://www.eecis.udel.edu/~mills/database/reports/ntp-survey99-minar.pdf Accessed: 2026-10-08.

[62] MIT Kerberos Consortium. [n. d.]. krb5.conf: Kerberos configuration file (`clockskew` relation). https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html. Accessed: 2026-10-08.

[63] Faten Mkacher and Andrzej Duda. 2019. Calibrating NTP. In *2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. 1–6. https://doi.org/10.1109/ISPCS.2019.8886646

[64] Mark Morowczynski. 2012. Did Your Active Directory Domain Time Just Jump To The Year 2000? https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/did-your-active-directory-domain-time-just-jump-to-the-year-2000/ba-p/255873. Accessed: 2026-10-08.

[65] Giovane C. M. Moura, Marco Davids, Caspar Schutijser, Cristian Hesselman, John Heidemann, and Georgios Smaragdakis. 2024. Deep Dive into NTP Pool's Popularity and Mapping. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS 2024)* 8, 1, Article 15 (February 2024), 30 pages. https://doi.org/10.1145/3639041

[66] Giovane C. M. Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Müller, Lan Wei, and Christian Hesselman. 2016. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Santa Monica, California, USA, 255–270. https://doi.org/10.1145/2987443.2987446

[67] Giovane C. M. Moura and John Heidemann. 2023. Vulnerability Disclosure Considered Stressful. *SIGCOMM Comput. Commun. Rev.* 53, 2 (July 2023), 2–10. https://doi.org/10.1145/3610381.3610383

[68] D. M'Raihi, S. Machani, M. Pei, and J. Rydell. 2011. *TOTP: Time-Based One-Time Password Algorithm.* RFC 6238. IETF. http://tools.ietf.org/rfc/rfc6238.txt

[69] Network Time Foundation. 2025. *NTPD 4.2.8p-series.* https://www.ntp.org/documentation/4.2.8-series/#building-and-installing-ntp Accessed: 2025-05-13.

[70] Ruben Nijveld. 2024. Enabling Pools in NTS. https://tweedegolf.nl/en/blog/139/enabling-pools-in-nts/. Accessed 02-10-2026.

[71] NIST. 2025. NIST Internet Time Service (ITS). (May. 1 2025). https://www.nist.gov/pml/time-and-frequency-division/time-distribution/internet-time-service-its

[72] Nkom. 2026. *Nytt system vil gjøre Norge mindre avhengig av tidssignaler fra GPS – viktig for samfunnssikkerheten.* https://nkom.no/aktuelt/tema-frekvensforum-2026 Accessed: 2026-10-05.

[73] Yevheniya Nosyk, Qasim Lone, Yury Zhauniarovich, Carlos H. Gañán, Emile Aben, Giovane C. M. Moura, Samaneh Tajalizadehkhoob, Andrzej Duda, and Maciej Korczyński. 2023. Intercept and Inject: DNS Response Manipulation in the Wild. In *Passive and Active Measurement*, Anna Brunstrom, Marcel Flores, and Marco Fiore (Eds.). Springer Nature Switzerland, Cham, 461–478.

[74] Yevheniya Nosyk, Malte Tashiro, Qasim Lone, Robert Kisteleki, Andrzej Duda, and Maciej Korczyński. 2025. Day in the Life of RIPE Atlas: Operational Insights and Applications in Network Measurements. *arXiv preprint arXiv:2511.22474* (2025). https://doi.org/10.48550/arXiv.2511.22474

[75] Andrew N. Novick and Michael A. Lombardi. 2015. Practical limitations of NTP time transfer. In *2015 Joint Conference of the IEEE International Frequency Control Symposium & the European Frequency and Time Forum*. 570–574. https://doi.org/10.1109/FCS.2015.7138909

[76] NTP Pool. 2024. pool.ntp.org: the Internet cluster of NTP servers. https://www.ntppool.org/en/.

[77] NTPsec project. 2022. *NTPSec*. https://www.ntpsec.org Accessed: 2025-05-13.

[78] Oleg Obleukhov. 2020. Building a more accurate time service at Facebook scale. https://engineering.fb.com/2020/03/18/production-engineering/ntp-service/.

[79] National Institute of Standards and Technology. 2026. NIST Authenticated NTP Service. https://www.nist.gov/pml/time-and-frequency-division/time-services/nist-authenticated-ntp-service. Accessed 02-10-2026.

[80] C. Partridge, T. Mendez, and W. Milliken. 1993. *Host Anycasting Service*. RFC 1546. IETF. http://tools.ietf.org/rfc/rfc1546.txt

[81] Yarin Perry, Neta Rozen-Schiff, and Michael Schapira. 2021. A Devil of a Time: How Vulnerable is NTP to Malicious Timeservers? In *Proceedings of the 28th Network and Distributed System Security Symposium (NDSS 2021)* (Virtual Conference).

[82] Pendulum Project. [n. d.]. *ntpd-rs: A full-featured implementation of the Network Time Protocol, including NTS support*. https://github.com/pendulum-project/ntpd-rs GitHub repository.

[83] The Linux Kernel Documentation Project. 2024. PTP hardware clock infrastructure for Linux. https://docs.kernel.org/driver-api/ptp.html. Accessed: 2025-11-05.

[84] D. Reilly, H. Stenn, and D. Sibold. 2019. *Network Time Protocol Best Current Practices*. RFC 8633. IETF. http://tools.ietf.org/rfc/rfc8633.txt

[85] E. Rescorla. 2018. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. IETF. http://tools.ietf.org/rfc/rfc8446.txt

[86] RIPE NCC. 2025. RIPE Atlas Measurements. https://atlas.ripe.net/measurements/ID. ID is the measurement ID: Google-android: 100787434, Meta: 100787995, Meta-v6: 100930063, Ubuntu: 100788409, Apple-ZA-map: 87282270, Microsoft-map: 86743242, Apple-map1: 86743255, Apple-map2: 86743251, Ubuntu-map: 87076806, AWS-map: 87077187, AWS-map-v6: 90364265, ms-server1-detection: 87119082, ubuntu-local-server: 90982800, apple-wrong-refts: 89211992, apple-local-servers: [89183871-89183874,89183992].

[87] RIPE NCC. 2026. RIPE Atlas Measurement IDs. https://atlas.ripe.net/measurements/ID. ID is the experiment ID: Apple-long: 160722842, Microsoft-long: 160723781, AWS-long: 160723821.

[88] RIPE NCC. 2026. RIPE Network Coordination Centre. https://www.ripe.net/. Accessed: 2026-10-08.

[89] RIPE NCC Staff. 2015. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal (IPJ)* 18, 3 (Sep 2015), 2–26.

[90] RIPE Network Coordination Centre. 2025. RIPE Atlas. https://atlas.ripe.net.

[91] Root Server Operators. 2025. Root DNS. http://root-servers.org/.

[92] Teemu Rytilahti, Dennis Tatang, Janosch Köpper, and Thorsten Holz. 2018. Masters of Time: An Overview of the NTP Ecosystem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. 122–136. https://doi.org/10.1109/EuroSP.2018.00017

[93] Jon Seager. 2026. Ntpd-rs: it's about time! Ubuntu Community Discourse. https://discourse.ubuntu.com/t/ntpd-rs-its-about-time/79154

[94] Amazon Web Services. 2022. *Amazon Time Sync is now available over the internet as a public NTP service*. https://aws.amazon.com/about-aws/whats-new/2022/11/amazon-time-sync-internet-public-ntp-service/ Accessed: 2025-03-11, Archived version: https://web.archive.org/web/20260701060007/https://aws.amazon.com/about-aws/whats-new/2022/11/amazon-time-sync-internet-public-ntp-service/.

[95] SIDN Labs. 2025. TimeNL. https://time.nl/index_en.html.

[96] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. 2020. MAnycast2: Using Anycast to Measure Anycast. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 456–463. https://doi.org/10.1145/3419394.3423646

[97] SURF. 2026. *Available to all scientists: the most accurate clock in the Netherlands*. https://www.surf.nl/en/news/available-to-all-scientists-the-most-accurate-clock-in-the-netherlands Accessed: 2026-10-08.

[98] Josh Taylor. 2026. *Neglected software update caused widespread Telstra network outage, CEO tells Senate inquiry*. https://www.theguardian.com/business/2026/jul/17/telstra-missing-software-update-undocumented-design-change-outage Accessed: 2026-10-05.

[99] Telstra. 2026. *What we've learned from the external investigation of our July outage.* https://www.telstra.com.au/exchange/what-we-ve-learned-from-the-external-investigation-of-our-july-o Accessed: 2026-10-05.
[100] Nikhil Tripathi and Neminath Hubballi. 2021. Preventing time synchronization in NTP broadcast mode. *Computers & Security* 102 (2021), 102135. https://doi.org/10.1016/j.cose.2020.102135
[101] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, David Choffnes, and Ethan Katz-Bassett. 2022. Internet Scale Reverse Traceroute. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 694–715. https://doi.org/10.1145/3517745.3561422
[102] VSL. [n. d.]. *VSL | National Metrological Institute of the Netherlands.* https://www.vsl.nl/en/ Accessed: 2026-10-07.
[103] Lan Wei and John Heidemann. 2017. Does Anycast Hang Up On You? In *IEEE Network Traffic Monitoring and Analysis Conference*. IEEE, Dublin, Ireland, 9. https://doi.org/10.23919/TMA.2017.8002905

A Use of Generative AI

We used Claude Opus 5.5 medium (Anthropic) to revise the language of author-written text. The authors take full responsibility for the content.

B Extra Graphs

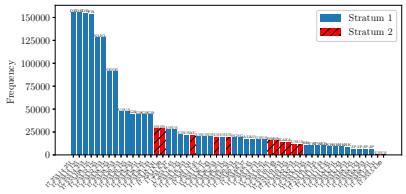


Fig. 11. Number of responses per NTP server for `time-osx.g.aaplimg.com`, Apple's second domain name.

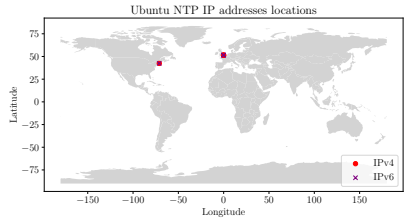


Fig. 12. Location of Ubuntu NTP server IP addresses. The servers are in only two distinct locations.

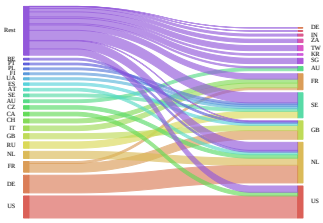


Fig. 13. Countries of RIPE Atlas probes (left) and IPv4 NTP servers (right) for `time-osx.g.aaplimg.com`. Dataset: Apple-map-2 [86].

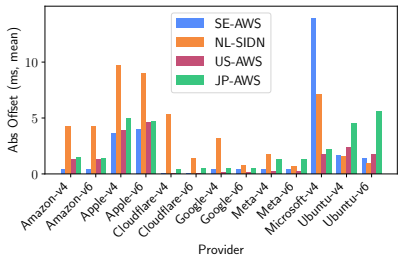
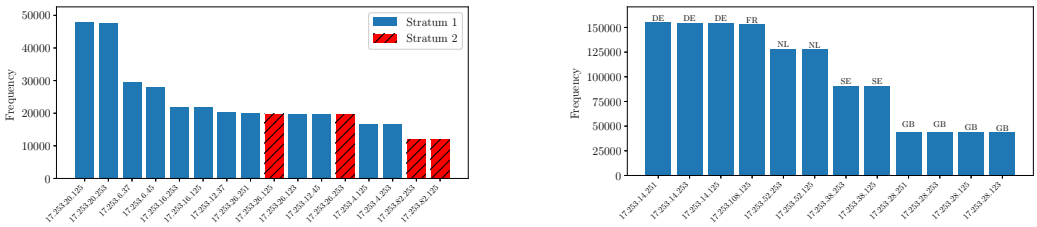


Fig. 14. Average absolute time offset per provider, as measured with NTP.

Figure 11 shows the results from §4.3 for `time-osx.g.aaplimg.com`, Apple's second domain name. Figure 12 shows the geolocation of Ubuntu's IP addresses: they are in only 2 different locations. Figure 13 shows the mappings of RIPE Atlas probes' countries to Apple NTP servers for `time-osx.g.aaplimg.com`. It follows the same mapping (and the same servers) as Figure 4b. Figure 14 shows the measured average absolute time offsets per provider. Figure 15 shows the number of responses for Apple's US and EU-based NTP servers. Figure 16 shows the reference IDs reported by each provider's time servers.



(a) Apple (`time.g.aaplimg.com` – US Servers) (b) Apple (`time.g.aaplimg.com` – EU Servers)

Fig. 15. Number of responses for each NTP server for Apple US and EU Servers

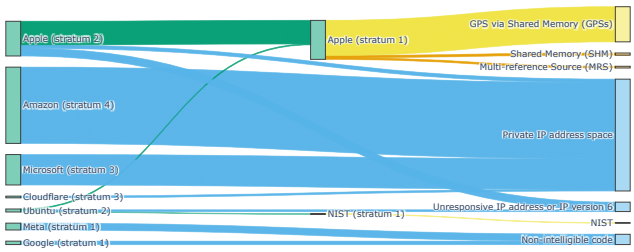


Fig. 16. Sankey plot of references (time sources) used by each provider's servers (SE-AWS measurement).

C Extra Tables

Table 12 gives a detailed overview of the DNS responses for the Microsoft, Apple, and AWS time servers. Table 13 presents the results of the extended 60-day measurement from §3.2.3. Table 14 gives statistics on the out-of-sync servers we found; cf. Figure 10 and §5.3. Table 15 shows the accuracy experiment results for NL-SIDN, US-AWS and JP-AWS.

D Offset Bounds and the Effect of Path Asymmetry

Figure 1 illustrates the standard NTP four-timestamp exchange, where T_1 and T_4 are recorded by the client, and T_2 and T_3 are recorded by the server. Let θ denote the true offset of the server clock relative to the client, and δ be the *measured* RTT, i.e., $\delta = (T_4 - T_1) - (T_3 - T_2)$.

Symmetric paths. If communication delays are known and constant (d_1, d_2), the offset can be calculated from either one-way delay:

$$\theta = T_2 - T_1 - d_1 = -(T_4 - T_3 - d_2)$$

If d_1 and d_2 are unknown, but assumed equal, then we can calculate them using the round-trip time $d_1 = d_2 = \frac{\delta}{2} = \frac{d_1+d_2}{2} = \frac{(T_4-T_1)-(T_3-T_2)}{2} = \frac{(T_4-T_3)+(T_2-T_1)}{2}$, yielding

$$\theta_{\text{NTP}} = T_2 - T_1 - d_1 = -(T_4 - T_3 - d_2) = T_2 - T_1 - \frac{(T_4 - T_3) + (T_2 - T_1)}{2} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

which is the normal NTP offset calculation, giving correct/exact offsets for equal delays.

Asymmetric paths. In practice, delays are neither equal nor constant, but depend on Internet routing. Asymmetry is common, i.e., $d_1 \neq d_2$, introducing a bias in the measured offset, i.e., a

	Microsoft	Apple1	Apple2	AWS
Query name	twc.trafficmanager.net A	time.g.aaplimg.com A	time-osx.g.aaplimg.com A	time.aws.com
Dataset	Microsoft-map	Apple-map-1	Apple-map-2	AWS-map
		Ripe Atlas		
Auth server	13.107.222.240	17.253.206.8	17.253.201.8	205.251.193.84
Total Probes	9,231	9,205	9,211	9,068
without valid response	221	211	223	129
with valid responses	9,010	8,994	8,988	8,939
only valid resp.	8,270	8,066	8,248	8,284
both valid and invalid	740	928	740	655
Total countries	171	171	172	172
without valid response	52	52	53	41
with valid responses	170	170	171	171
only valid resp.	167	163	164	165
both valid and invalid	170	170	171	171
Total ASes	3,143	3,142	3,143	3,115
without valid response	158	152	158	104
with valid responses	3,094	3,094	3,093	3,081
only with valid resp.	2,865	2,874	2,921	2,905
both valid and invalid	487	489	394	424
DNS queries	663,637	661,834	662,165	652,105
valid response (RCODE 0)	642,617	642,042	641,436	640,253
Timeout/ network error	11,945	12,896	12,080	11,754
SERVFAIL (RCODE 2)	4,378	2,670	4337	20
REFUSED (RCODE 5)	4,697	4,210	4,312	78
malformed response	0	16	0	0
		Time Servers		
Unique NTP servers	15	57	55	93
Valid	12	53	53	90
Invalid	3	4	2	3
ASes NTP Servers	1 (AS8075)	2 (AS6185 and AS714)	2 (AS6185 and AS714)	2(AS14618 and AS16509)

Table 12. Mapping Time Service Networks. Datasets: [86]

	Apple	**Microsoft**	**AWS**
Probes	8,929	8,930	8,930
Queries	2,835,666	2,793,836	2,836,777
Responses	2,714,279	2,675,008	2,714,153
NTP servers	52	12	86
Dataset ID	160722842	160723781	160723821

Table 13. Measurement summary across the three target services. Datasets: [87]

Server	104.40.149.189	20.101.57.9	51.137.137.111
# Mismatches	20	137	42
Largest Mismatch	-4.20ms	-39.08ms	-9.16ms
Mean RTT	22.31ms	22.93ms	31.57ms
Abs. Offset Q1	11.35ms	22.92ms	16.38ms
Abs. Offset Q4	13.14ms	33.74ms	18.98ms
Largest Offset	-15.61ms	-50.97ms	-24.51ms

Table 14. Statistics about out-of-sync servers (SE-AWS): mismatch $> 0 \Leftrightarrow |\text{offset}| > \text{RTT}/2$; Mismatch $= T_2 - T_1$ if $T_1 > T_2$ else $T_3 - T_4$ if $T_3 > T_4$ else 0; Q1 and Q4: first and last quartile

difference between θ and θ_{NTP}:

$$\theta = \frac{(\theta + \theta)}{2} = \frac{T_2 - T_1 - d_1 - (T_4 - T_3 - d_2)}{2} = \frac{(T_2 - T_1) - (T_4 - T_3) + (d_2 - d_1)}{2}$$
$$= \frac{(T_2 - T_1) - (T_4 - T_3) + d_2 - d_1}{2} = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} + \frac{d_2 - d_1}{2} = \theta_{\text{NTP}} + \frac{d_2 - d_1}{2}$$

NL-SIDN

Provider	IP ver.	#IP Addr.	#Req.	#Resp.	No Resp. #	No Resp. %	Offset mean	Abs. Offset 90th %ile	RTT mean	NTP version	Stratum
Amazon	v4	88	25 344	25 329	15	0.06%	2.55 ms	10.16 ms	151.36 ms	v4	4
	v6	88	25 344	25 327	17	0.07%	2.59 ms	9.92 ms	151.02 ms	v4	4
Apple	v4	46	13 248	12 366	882	6.66%	3.71 ms	20.90 ms	174.68 ms	v4	1/2 (77%/23%)
	v6	45	12 960	11 630	1330	10.26%	-2.09 ms	20.33 ms	155.06 ms	v4	1/2 (79%/21%)
Cloudflare	v4	2	576	576	0	0.00%	-4.82 ms	4.31 ms	15.18 ms	v4	3
	v6	2	576	576	0	0.00%	-0.68 ms	0.81 ms	6.73 ms	v4	3
Google	v4	4	1 152	1 152	0	0.00%	-3.05 ms	1.33 ms	14.37 ms	v4	1
	v6	4	1 152	1 152	0	0.00%	-0.62 ms	0.30 ms	9.39 ms	v4	1
Meta	v4	5	1 440	1 311	129	8.96%	-1.65 ms	3.92 ms	16.43 ms	v4	1
	v6	5	1 440	1 440	0	0.00%	-0.46 ms	1.38 ms	14.83 ms	v4	1
Microsoft	v4	12	3 456	3 303	153	4.43%	-6.37 ms	5.39 ms	130.98 ms	v3	3
Ubuntu	v4	4	1 152	1 146	6	0.52%	-1.24 ms	3.52 ms	26.13 ms	v4	2
	v6	3	864	863	1	0.12%	0.52 ms	0.98 ms	9.73 ms	v4	2

US-AWS

Provider	IP ver.	#IP Addr.	#Req.	#Resp.	No Resp. #	No Resp. %	Offset mean	Abs. Offset 90th %ile	RTT mean	NTP version	Stratum
Amazon	v4	75	21 599	21 585	14	0.06%	0.48 ms	4.86 ms	128.65 ms	v4	4
	v6	75	21 600	21 592	8	0.04%	0.46 ms	4.93 ms	128.66 ms	v4	4
Apple	v4	27	7 776	7 404	372	4.78%	2.14 ms	4.54 ms	141.00 ms	v4	1/2 (72%/28%)
	v6	27	7 776	7 151	625	8.04%	2.64 ms	5.41 ms	123.54 ms	v4	1/2 (75%/25%)
Cloudflare	v4	2	576	576	0	0.00%	-0.07 ms	0.18 ms	1.42 ms	v4	3
	v6	2	576	576	0	0.00%	-0.05 ms	0.17 ms	1.38 ms	v4	3
Google	v4	4	1 152	1 152	0	0.00%	0.10 ms	0.23 ms	2.99 ms	v4	1
	v6	4	1 152	1 152	0	0.00%	0.05 ms	0.21 ms	2.91 ms	v4	1
Meta	v4	5	1 440	1 440	0	0.00%	-0.07 ms	0.47 ms	4.97 ms	v4	1
	v6	5	1 440	1 440	0	0.00%	-0.10 ms	0.52 ms	4.94 ms	v4	1
Microsoft	v4	12	3 456	3 372	84	2.43%	-0.85 ms	3.52 ms	126.76 ms	v3	4/3 (58%/42%)
Ubuntu	v4	4	1 152	1 102	50	4.34%	1.71 ms	3.17 ms	66.28 ms	v4	2
	v6	3	864	864	0	0.00%	0.02 ms	2.58 ms	85.15 ms	v4	2

JP-AWS

Provider	IP ver.	#IP Addr.	#Req.	#Resp.	No Resp. #	No Resp. %	Offset mean	Abs. Offset 90th %ile	RTT mean	NTP version	Stratum
Amazon	v4	75	21 600	21 581	19	0.09%	0.16 ms	4.17 ms	157.65 ms	v4	4
	v6	75	21 600	21 583	17	0.08%	0.00 ms	3.92 ms	157.94 ms	v4	4
Apple	v4	27	7 776	7 425	351	4.51%	2.02 ms	11.02 ms	147.99 ms	v4	1/2 (72%/28%)
	v6	27	7 776	7 169	607	7.81%	1.04 ms	13.37 ms	136.43 ms	v4	1/2 (75%/25%)
Cloudflare	v4	2	576	576	0	0.00%	0.20 ms	1.03 ms	1.92 ms	v4	3
	v6	2	576	576	0	0.00%	0.27 ms	1.03 ms	1.72 ms	v4	3
Google	v4	4	1 152	1 152	0	0.00%	-0.32 ms	0.98 ms	35.28 ms	v4	1
	v6	4	1 152	1 152	0	0.00%	-0.31 ms	0.95 ms	35.01 ms	v4	1
Meta	v4	5	1 440	1 440	0	0.00%	-1.03 ms	2.15 ms	43.51 ms	v4	1
	v6	5	1 440	1 440	0	0.00%	-1.10 ms	2.24 ms	43.49 ms	v4	1
Microsoft	v4	12	3 456	3 312	144	4.17%	-1.11 ms	4.74 ms	127.26 ms	v3	4/3 (59%/41%)
Ubuntu	v4	4	1 152	1 142	10	0.87%	-4.33 ms	6.67 ms	204.01 ms	v4	2
	v6	3	864	864	0	0.00%	-5.52 ms	7.37 ms	215.30 ms	v4	2

Table 15. From top to bottom: Statistics from the NTP measurement results from the vantage points NL-SIDN, US-AWS and JP-AWS. Anycasted services are highlighted in orange.

Because $d_1, d_2 \geq 0$, it follows that $-(d_2 + d_1) \leq d_2 - d_1 \leq d_2 + d_1$, with those bounds occurring when one delay is 0. Therefore, the corresponding bounds expand to $-\frac{d_1+d_2}{2} \leq \theta - \theta_{\text{NTP}} \leq \frac{d_1+d_2}{2}$, i.e. $|\theta - \theta_{\text{NTP}}| \leq \frac{\delta}{2}$ – the measurement error in the offset is bounded by half of the RTT.

Received 5 June 2026; accepted 10 September 2026