Deep Dive into NTP Pool Popularity and Mapping

Moura, Giovane C. M.; Davids, Marco; Schutijser, Caspar; Hesselman, Cristian; Heidemann, John; Smaragdakis, G.

Publication date
2023

Document Version
Final published version

Citation (APA)
Moura, G. C. M., Davids, M., Schutijser, C., Hesselman, C., Heidemann, J., & Smaragdakis, G. (2023). *Deep Dive into NTP Pool Popularity and Mapping*. (SIDN Labs Technical Report). SIDN.

Deep Dive into NTP Pool Popularity and Mapping

SIDN Labs Technical Report – 2023-10-12

GIOVANE C. M. MOURA, SIDN Labs and TU Delft, The Netherlands
MARCO DAVIDS, SIDN Labs, The Netherlands
CASPAR SCHUTIJSER, SIDN Labs, The Netherlands
CRISTIAN HESSELMAN, SIDN Labs and University of Twente, The Netherlands
JOHN HEIDEMANN, USC/ISI and CS Dept., USA
GEORGIOS SMARAGDAKIS, TU Delft, The Netherlands

Time synchronization is crucial on the Internet, and the Network Time Protocol (NTP) serves as the primary synchronization protocol. The NTP Pool, a volunteer-driven project introduced 20 years ago, connects clients with NTP servers. Our analysis of Root DNS queries reveals the NTP Pool's widespread use as the most popular time service. Despite its popularity, there has been limited scrutiny of how NTP servers are assigned to clients. In this paper, we investigate the NTP Pool's DNS component (`GeoDNS`), which maps clients to servers, and find that the current algorithm is overly strict, creating unnecessary risks. We have shared our findings with the NTP Pool operators, who acknowledge them and plan to revise their algorithm to enhance security.

1 INTRODUCTION

Global time synchronization underpins modern life. It is crucial to the Internet and to critical systems such as financial markets, power grids, and telecommunications networks [30]. In businesses, precise clock information is also vital: distributed systems and applications such as backup systems are entirely dependent on precise clock information [39, 70]. Operational failures can occur whenever clocks are out of sync – ultimately leading to data loss [27].

On the Internet, a series of applications, services, and protocols that can be compromised or impaired simply by tampering with their hosting system's clocks. TLS [19], DNSSEC signatures [2], DNS caches [54], RPKI [12], Kerberos [59], and even Bitcoin transactions are among the many that fundamentally depend on clock synchronization to prove (cryptographic) freshness [18, 42, 49, 80]. In November 2012, the US Navy Naval Observatory (USNO) NTP servers [71] started to provide wrong time information (roughly 12 years wrong) [9] and caused outages in multiple places, including Active Directory and even routers [42, 45].

The Network Time Protocol (NTP) [49] is the Internet's default protocol for clock synchronization[1]. It is designed to mitigate the effects of changing network latency (jitter) between client and server. NTP *servers* are synchronized out-of-band with reference clocks, which can be atomic clocks, radio signals (*e.g.,* DC77 [11]), and satellites such as GPS and Galileo. Clients and other secondary NTP servers, in turn, synchronize their clocks with NTP servers over the Internet. Clients either use servers they have been pre-configured with or servers provided by their networks with DHCP [21, 24].

There are many publicly available NTP servers on the Internet. NIST [60] and the USNO [71] have been providing NTP services for decades. Later, several vendors such as Apple [1], Google [25], Cloudflare [15], Meta [70], Microsoft [47] and Ubuntu [86] started their own services.

Unlike the previous providers, the NTP Pool [65] is a time provider that *does not* run NTP servers itself. Instead, it relies on third-party NTP servers run by volunteers, and the NTP Pool's main task is to map NTP clients to these volunteer NTP servers – which they do by using DNS [50]. These

[1]The Precision Time Protocol (PTP) [31, 32] provides higher precision than NTP, and it is used in local area networks and wide area networks, and usually not over the Internet. It is typically deployed on layer 2 and it is often used in financial transactions, mobile phone towers and other industrial networks.

volunteer servers vary significantly: from DSL home users to large cloud operators. The NTP Pool currently lists 4,403 volunteer NTP servers – 3,056 IPv4 and 1,671 IPv6 (2023-10-09) [61], and it has been providing time services for two decades, being popular among vendors [66, 80], including various Linux distributions and Android devices.

With today's free time services run by large operators, one may wonder whether people still use the NTP Pool. Our *first contribution* is to show that, based on DNS traffic at the Root DNS servers [79], the NTP Pool is the *most popular* time service on the Internet (§3), surpassing NIST and USNO as time provider. We show how a large variety of clients uses it and how a single server on the NTP Pool receives 7.2 billion daily queries, from 158M clients and 52k Autonomous Systems (ASes) globally (for comparison, all twenty NIST NTP servers received 16 billion queries/day in 2016 [82]).

Given the NTP Pool's popularity, it is paramount to understand *how* it decides which clients are served by what NTP servers, and *why*. There is currently little scrutiny on *how* the NTP Pool performs this mapping: the project's documentation is incomplete (a fact acknowledged by its operators, who are volunteers primarily focused on *running* the service).

Thus, our *second contribution* (§4) is to provide a detailed exploration of the client/server mapping process, and, more importantly, its implications for clients. We demonstrate that it is a composite of two factors: (i) the geographical locations of both NTP servers and clients, and (ii) the performance of NTP servers, as assessed by the NTP Pool monitoring services. Our analysis reveals that this process can be overly strict and inequitable for clients in underrepresented locations, leading to their reliance on only one or two time providers (§5), despite the NTP Pool's extensive network of volunteer NTP servers. For instance, we observed that *all* clients from 21 countries, including Israel, Egypt, and Nigeria, are mapped mapped to just two NTP servers (both operated by Cloudflare), which introduces unnecessary risk and centralizes too much control within the hands of a single operator.

Our *third contribution* highlights that the `GeoDNS` mappings can be relaxed (§6). Discussions with NTP Pool operators reveal that these strict mappings aim to prevent asymmetric routing and mitigate packet loss concerns. However, our experiments show that these fears of significant packet loss from distant servers are unfounded: our measurements show that far away NTP servers can also provide good timing services with low packet loss ratios. Therefore, we recommend that NTP Pool operators consider relaxing their mapping criteria to mitigate well-known vulnerabilities [73, 80] that could lead to complete or partial time synchronization takeover for entire countries.

2 THE NTP POOL PROJECT

The NTP Pool project is a dynamic collection of thousands of NTP servers that provide accurate time via the NTP protocol [49] to clients worldwide. These NTP servers are assigned to clients using DNS, under the pool.ntp.org zone. It was proposed as a solution to reduce the abuse of publicly available NTP servers [88]. Instead of having a long list of public NTP servers (which individually could more easily become overloaded), the NTP Pool project proposed to "load balance" NTP traffic using DNS. The project has been active for roughly 20 years.

The NTP Pool does not run NTP servers; volunteers run their own NTP servers which they add to the NTP Pool using a web interface [67], where they can also choose how much capacity they want to donate (values ranging from 512Kbps to 1Gbps). We show this process on the left side of Figure 1, where IP addresses are added with their respective *BW* capacity parameter.

To use the NTP Pool for clock synchronization, *clients'* time software is set with domain names from the NTP Pool DNS zone, as shown in the right side of Figure 1. In this figure, we see a client that uses the domain names `[0-3].debian.pool.ntp.org` in their config files (*e.g.*, `/etc/ntp.conf`). Whenever needed, the client will ask its local DNS resolver to fetch the IP addresses for these domain names (B in Figure 1), and the NTP Pool *authoritative* DNS servers (`[a-i].ntpns.org`)

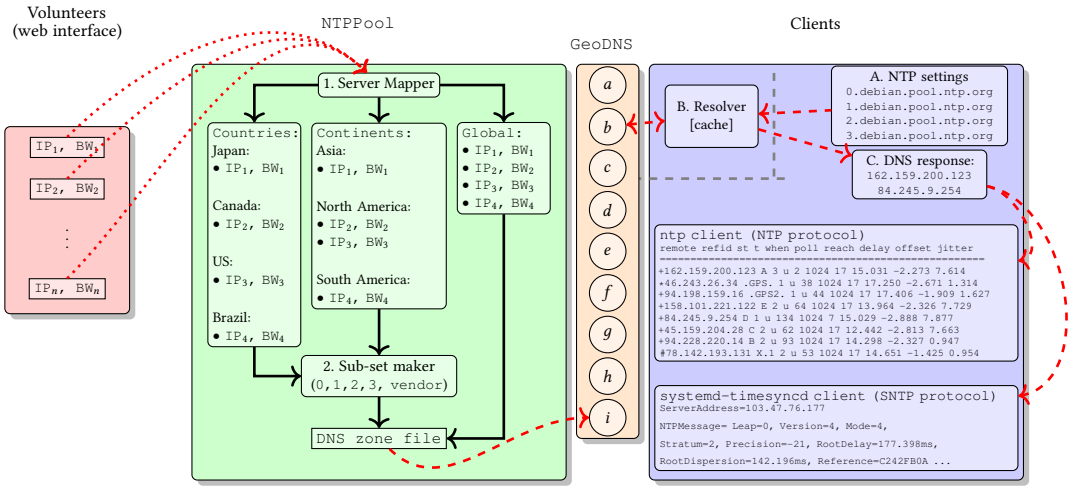


Fig. 1. NTP Pool operations: from volunteers to clients

will answer with a list of IP addresses (C) to the DNS resolvers, which will then forward them to the clients. The NTP Pool runs `GeoDNS`, their customized authoritative DNS server software [6]. Upon receiving NTP servers addresses from the NTP Pool authoritative servers, the client's time synchronization software will contact these NTP servers for accurate time information.

Do clients blindly trust NTP servers? Consider a malicious NTP server that provides wrong time data – say, ten years ahead of the current time. To prevent attackers from tampering with the client's clock, the NTP protocol has built-in algorithms (§10 and §11 in [49]) that continuously evaluate the timing samples from various NTP servers, disregarding discrepant ones. Moreover, it can combine time information from multiple NTP servers to update the system's clock.

Figure 1 shows an example of a `NTP` reference implementation client status [58]. It shows 8 NTP servers under evaluation – all retrieved from the NTP Pool by the client (NTP clients use multiple NTP servers if they are available to select the best ones). Each server has several metrics (offset, delay, jitter), which are all part of the NTP protocol filtering specifications. This particular client combines data from all servers with marked with '*' and '+' symbols on the left side of the IP address to synchronize its clock. In this way, NTP implementations prevent the harmful effects of individual malicious NTP servers. The exception occurs when a server reboots and may trust whatever time information is provided with – or when ntpd is ran with the `-s` option.

SNTP [48] clients, however, will blindly trust the time information provided by time servers. SNTP is a simplified version of NTP which is designed to provide basic time synchronization functionality with minimal overhead. Even if an SNTP client receives multiple NTP servers from the NTP Pool, it will use only a single server. It is especially suitable for resource-constrained devices such as IoT devices. In Figure 1, we show the status of `systemd-timesyncd`, a SNTP implementation running on Ubuntu, where we see a single NTP server.

How does the NTP Pool prevent malicious volunteers? Anyone can add an NTP server to the NTP Pool. To prevent malicious or bogus NTP servers from being assigned to clients, the NTP Pool operators constantly monitor every volunteer NTP server. Bogus servers are removed from the zone and not served to the clients. (We demonstrate in §4.2 how this system works).

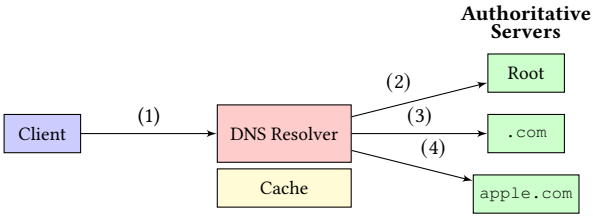


Fig. 2. Time servers domain name resolution.

Changing system configurations: clients are typically configured with *hardcoded* NTP servers, which they can manually change or, if available, use the NTP servers provided by the DHCP [21] protocol, which also enables dynamically setting NTP servers. For example, one of the authors institution provides NTP servers via DHCP, which causes the NTP client on Linux boxes to *not* use the NTP Pool while connected to the institution network.

3 IS THE NTP POOL RELEVANT NOWADAYS?

Given that several large cloud and content providers now have their own NTP services (Microsoft, Google, Facebook, Cloudflare, and Apple), one may wonder how relevant is NTP Pool still for keeping time on the Internet. For DNS, we have seen that when large cloud and content providers entered the DNS resolver market, they quickly amassed the market [51]. Has the same happened for NTP, reducing the NTP Pool relevance?

The direct way to answer this question would be to compare NTP traffic across different time providers. That, however, would require access to vantage points inaccessible to us. Thus, we address this question by comparing the NTP Pool popularity with other NTP services by analyzing DNS traffic collected at the Root DNS servers [79].

3.1 Root DNS and time keeping

Before synchronizing clocks with NTP servers, clients must first resolve the domain names associated with the time service. Consider Figure 2 as an example, where a client first (step 1) sends a DNS query for `time.apple.com` to its DNS resolver (*e.g.,* its local ISP's resolver). The DNS resolver's job is to perform the domain name *resolution*, i.e., find the IP addresses corresponding to this domain.

In this process, the resolver must first contact one of the 13 Root DNS servers, asking for the authoritative server of `.com` zone authoritative servers (step 2 in Figure 2). Recall that authoritative DNS servers are another type of DNS server that knows zone contents from memory [28]. Once the resolver knows this answer, it can contact the `.com` authoritative DNS servers for the authoritative servers of `apple.com` (step 3). Next, the `apple.com` authoritative servers can tell the resolver which IP addresses are associated with `time.apple.com` (step 4) and finally can answer its client. The client then uses the IP addresses to synchronize its clock.

While we do not have access to DNS traffic from each time provider, we have access to traffic from the Root DNS servers – the DITL datasets [20], a two-day-a-year traffic capture of the Root DNS servers. The Root DNS traffic provides a view of the top of global DNS traffic [14, 40]. As such, this data set can be seen as a lower bound of estimative of the number of time services users.

3.2 Limitations

The Root DNS data has several limitations regarding the question we seek to answer:

They do not see real clients: The root servers only see client's resolvers (Figure 2), not actual clients. As such, one resolver may either cache results for one client or a thousand – our vantage point does not allow us to tell how many are behind the resolver.

They only see cache misses: As shown in Figure 2, the DNS resolvers deploy caches for retrieved records, which are used to improve response times [54, 55] by eliminating the need to send repetitive queries. If records are cached, resolvers can *directly* answer clients, eliminating the need of queries to (some) authoritative servers. As such, the root servers only see cache *misses*.

Query name (qname) minimization reduces coverage: qname minimization [10] improves user privacy by avoiding domain name leakage. For example, instead of querying the Roots for `time.apple.com` (which the roots cannot directly answer, they can only point to where the `.com` authoritative servers), an RFC9156-compliant would query only the roots for `.com`, and avoid leaking `time.apple.com` domain name. A previous study has shown that qname minimization is still not widespread [17], but it is on the rise [41]. Our method only leverages resolvers that do not deploy qname minimization.

Junk: The Root DNS server is known to receive large amounts of bogus queries, i.e., spoofed addresses, wrong query names, wrong query types, and so forth [14]. While we cannot verify if UDP queries are spoofed, but we disregard queries whose source addresses are from private or unrouted address space.

3.3 Datasets

There are thirteen root DNS servers on the Internet. Each one is referred to by the first letter in its name (`[a-m].root-servers.net`). We analyze traffic collected at twelve of the thirteen root servers (DITL 2022 dataset [20]); I-ROOT has anonymized IP addresses, which prevents us from doing further analysis. For a historical comparison, we compare against data at ten servers from DITL 2017 dataset. The two datasets collect all incoming queries to the Root servers between April 12–14, 2022 (E-ROOT seems to provide only partial data) and from April 11–13, 2017, respectively.

For each query q, we extract its query name and match it against a list of server names used by time providers we compiled using multiple sources (Table 1). We then compute the number of unique queries, clients, and autonomous systems (ASes) for each time provider (we use CAIDA's Routeviews Prefix2AS datasets to map IP addresses to ASes [13]).

Table 2 shows the DITL datasets after processing. In 2022, we identified 126 million queries from 491 thousand resolvers and 22 thousand ASes that queried for names matching the domain names in Table 1. We notice that the distribution varies per root letter as resolvers employ their own criteria to choose which root server to contact [57].

3.4 Comparing time services

Figure 3a shows the query distribution per time provider from the DITL datasets. We see that NTP Pool receives more than 90 million out of 126 millions queries in total, being far more popular than the all other time providers combined.

This dominance can be due to several causes. First, the NTP Pool has multiple subzones – the default one is `pool.ntp.org`, but there are geographical ones (`europe.pool.ntp.org`) and vendor zones (`android.pool.ntp.org`). As such, this large number of subzones increases the query volume in compared to other providers that use one or few domain names.

To rule out the effect of multiple zones, we compute the number of queries to `pool.ntp.org`, which counts only queries to the main zone of the NTP Pool. In total, 25 million queries from 138 thousand resolvers and 10.2 thousand Ases have queried for it, which is more queries than NIST in 2022 (6.8 million), but fewer resolvers and ASes (145 thousand resolvers and 14 thousand ASes).

DNS time-to-live (TTLs) may also influence the query counts. Indeed, for a new query to reach the Root, the TLD TTL must expire and, at the same time, the resolver must ask for the record of

Provider	Server Name	TTL	TLD TTL
Apple	{time,time[1–7],time.euro, time.asia}.apple.com	2h	2 days
Cloudflare	time.cloudflare.com	5min	2 days
Facebook	{time,time[1–5],}.facebook.com}	1h	2 days
Google	{time,time[1–4],}.google.com},time.android.com	4h	2 days
Microsoft	time.windows.com	1h	2 days
NIST	{time,time-[a,b,c,d,e]-[g,wwv,b]} .nist.gov,{utcnist,utcnist2.colorado.edu}	30min	2 days
NTP Pool	*.pool.ntp.org	2.5min	1h
Ubuntu	ntp.ubuntu.com	1min	2 days
USNO	{u,tock,ntp2}.usno.navy.mil	(<5min)	6h
VNIIFTRI	ntp[1–4].vniiftri.ru,ntp[1-2].niiftri. irkutsk.ru,vniiftri[,2].khv.ru	1 day	4 days
Rest	137 NTP servers – see §A	–	–

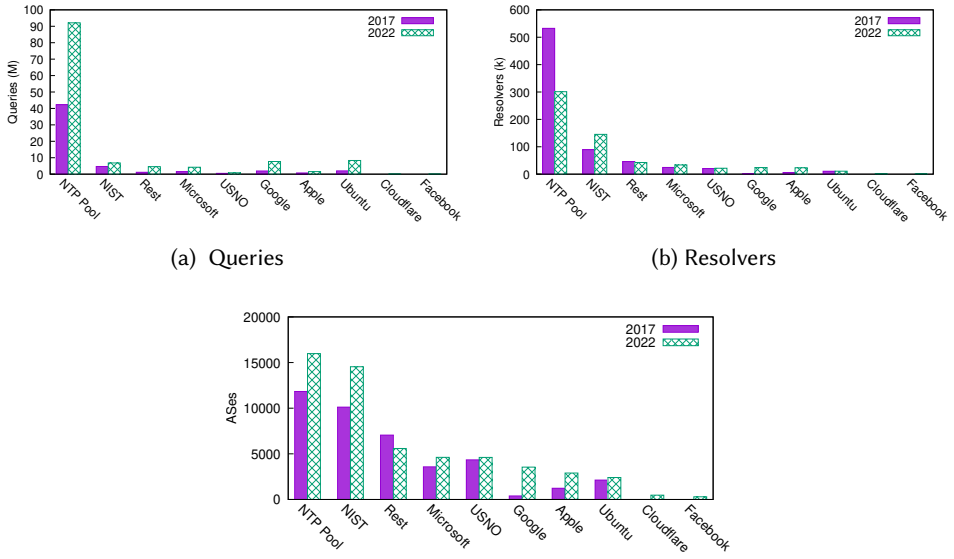
Table 1. Evaluated Time Providers and their records TTL, and their TLD's own TTL

Srv.	Queries		Resolvers		ASes	
	2017	2022	2017	2022	2017	2022
A	29,178,992	30,088,926	197,721	117,175	913	10,747
B	7,449,043	357,4484	131,362	45,932	7,022	6,107
C	8,359,883	13,153,018	15,6942	89,692	8,517	9,506
D	410,6686	7,498,890	127,895	68,408	6,499	8,204
E*	5,693,446	144,861	152,961	1,065	8,108	219
F	2,361,662	3,692,906	44,032	17,083	4,366	2,817
G	NA	3,862,762	NA	48,307	NA	6,353
H	834,493	4,545,561	76,836	50,538	4,389	6,740
J	6,692,983	13,311,972	157,677	95,582	8,086	10,021
K	6,402,332	15,835,168	146,007	92,450	8,154	9,234
L	5,882,535	16,294,733	134,487	74,854	7,199	6,055
M	NA	14,200,343	NA	98,377	NA	9,187
Total	76,962,055	126,203,624	873,543	491,764	17,047	22,167

Table 2. DITL datasets: Matching queries per Root Servers (IPv4 and IPv6). *E-root 2022 datasets are incomplete. April 11–13, 2017 and April 12–14, 2022.

one of the time providers. As show in Table 1, the NTP Pool TLD TTL is one hour – the smallest of all we evaluate – which increases the chances of cache misses and increasing query volumes. (Even though IP anycast [72] can be used in NTP servers [74], their deployment does not intefere with DNS TTLs and how the NTP Pool maps clients to servers).

Mitigating TTL influence: We can reduce the influence of TTLs by looking into the number of unique resolvers and ASes each time provider receives (Figure 3b, Figure 3c). In this way, we could only once each resolver or AS and reduce the impact of expired TTL records. We see that the NTP Pool is still the most popular in terms of resolver and ASes – but NIST, the second most



(a) Queries (b) Resolvers

(c) ASes

Fig. 3. DITL results and top 10 time providers. Dates: April 11–13, 2017 and April 12–14, 2022.

popular provider in terms of both metrics – both known to have been providing time services for an extended period.

Changes over time: when we compare the results from 2022 with 2017, we see that the NTP Pool was also the most popular service by any metric. We also note that newcomers did not have a significant impact in 2022 (Facebook and Cloudflare did not offer time services in 2017).

Despite the large cloud providers entering the market later, we observe that the NTP Pool is still the most queried service in terms of DNS traffic (at least a the Root level), followed by NIST. Even though Microsoft, Ubuntu, Google, and Apple set their devices to use their own time services, the NTP Pool attracts more resolvers and ASes. We attribute this, as we shall show in the next section, due to a large set of devices using it.

NTP Traffic from one server: we run an NTP server as volunteers within the NTP Pool. Our NTP server serves multiple regions and uses IP anycast. We have collected 24 hours traffic during Jun 22–23 and observed 7.2 billion queries from 158 million clients from 52,014 ASes globally[8]. Bear in mind that this is a single server out of the more than 4000 listed at the NTP Pool. For comparison, in 2016, NIST reported 16 billion daily queries [83].

There is also an interesting relationship between Root DNS traffic and *actual* NTP traffic: a single anycast NTP service sees far more ASes (52 thousands) than the Root DNS sees for NTP Pool names (16 thousands). This example provides evidence of the popularity of NTP Pool in terms of NTP traffic, even if is from a single NTP server.

4 CLIENT-TO-SERVER MAPPING

The NTP Pool utilizes `GeoDNS` for the mapping of clients to volunteer NTP servers (middle box in Figure 1). It could be argued that examining the `GeoDNS` source code alone should be adequate for comprehending the mapping criteria. While this is a valid point, it is important to note that a code

analysis alone cannot be applied to ascertain, in practical terms, the specific servers assigned to real-world clients. This is because such analysis does not encompass the dynamic state of the NTP Pool, which is defined by the list of NTP servers and their performance metrics. These are used to derive the input files of `GeoDNS`, which are frequently changing.

Hence, it is crucial to consider the state of the NTP Pool, encompassing the list of volunteer servers, their configuration parameters, and their status. This comprehensive understanding can only be attained through active Internet measurements. Bearing this in mind, we perform two types of measurements: (a) in a real-world scenario, employing 9.2k vantage points (VPs) (§4.1), and (b) in a controlled environment (§4.2).

4.1 View from the wild

The NTP Pool operators list that 4.7k NTP Servers (2023-10-10). We seek to understand the logic between client/server mapping *and* its implications for real-world clients, given the population of NTP servers. A previous work [80] observed the client for a single vantage point in Germany was mapped to NTP servers located in Germany by the NTP Pool. However, it did not explore the reasons why and how.

To understand the NTP Pool mappings in practice, we set up two measurements (for IPv4 and IPv6) using 9.2 thousand VPs using RIPE Atlas probes [77, 78] (RIPE Atlas probes are are hardware devices or virtual machines (VMs) that can be remotely instructed to carry out active measurements). In total, our VPs cover 3,082 ASes in 166 countries.

We configure these 9 thousand Atlas probes to send DNS queries to one of the NTP Pool authoritative servers (b.ntpns.org over IPv4 – 185.120.22.23), so we bypass DNS resolvers (Figure 1) and avoid hitting the resolver's cache. By passing resolvers, we can retrieve new NTP Pool addresses for every new query. The probes are configured to send queries every 5 min – a safe limit that does not overload RIPE Atlas and the NTP Pool authoritative DNS servers[2].

Table 3 shows the experiments' details. In the first experiment (EnumV4), we configure Atlas probes to retrieve IPv4 NTP servers, whereas in the second (EnumV6) we retrieve IPv6 NTP servers. For both experiments, we see ~9.2 thousand active VPs, having 9.1 thousand received valid responses (some VPs are blocked or contain bogus responses – a problem reported in Atlas probes in other works [53], which we disregard). These 9.1 thousand VPs provide us with a view from ~3 thousand ASes, totaling ~2.5 million DNS queries/responses per experiment.

For each experiment, each Atlas VP sends roughly 275 queries, receiving up to 4 NTP server addresses per response (Table 3). Theoretically, this would allow each probe to retrieve up to 1,100 unique NTP server addresses from the NTP Pool, if the process were completely random and if each client would not receive repeated NTP servers (we refrain from running an experiment that could span over all servers to avoid overloading RIPE Atlas).

Figure 4 shows the cumulative distribution function (CDF) of the number of unique IPs retrieved by each probe. We see a very different distribution of NTP servers per probe for both IPv4 and IPv6. Roughly 10% of the clients see up to 12 NTP servers (EnumV4) and 5 NTP servers (EnumV6). Considering that the NTP Pool has thousands of NTP servers, it is quite remarkable such limited number of assigned servers. Next, we explain the reasons behind these differences.

[2] The effects of caching in DNS have been previously studied [54, 56], and most resolvers seem to to respect the TTL of DNS records. In the case of the NTP Pool, the DNS records have a 150 sec TTL, which means that once a resolver obtains a response, all subsequent client queries within this TTL period are responded from cache instead of triggering new queries. As such, for a given resolver, all its clients will see the same NTP Pool view for 150 sec. If a `ntpd` client is receive multiple NTP servers, it will evaluate them, whereas a SNTP client chooses only one – see §2.

Measurement	EnumV4	EnumV6
Target	185.120.22.23	
QNAME	`2.pool.ntp.org`	
QType	A	AAAA
Date	2021-08-2[6-7]	2021-08-3[0-1]
Interval	5min	5min
Duration	24h	24h
VPs	9,260	9,272
valid resp.	9,113	9,127
no resp.	147	145
ASes	3,116	3,133
valid resp.	3,082	3,095
no resp.	156	148
Countries	166	168
Responses	2,534,199	2,583,318
Valid Responses	2,469,211	2,535,981
invalid/empty	64,988	47,337
NTP servers	3,056	1,479
Queries/VP	275	275

Table 3. RIPE Atlas experiments. Datasets: [75].

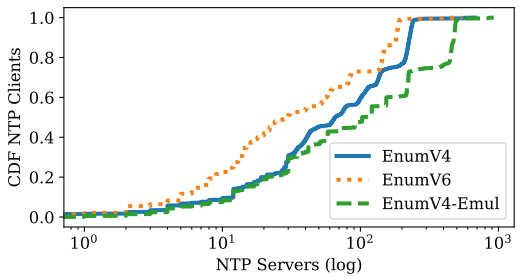


Fig. 4. CDF of NTP servers seen per Atlas VP.

4.2 Controlled experiment

We first *replicate* the NTP Pool authoritative DNS server setup and then *replay* the DNS queries from our experiments in the wild. By doing that, we can obtain the `GeoDNS` logs from our own instance and use this information to understand how it maps clients to servers.

`GeoDNS` (v. 3.0.2) takes as input a DNS zone file, that lists all zones in the pool (geographical and vendors) and their respective NTP servers. The `GeoDNS` source code does not have the actual zone file used by the NTP Pool, and we were not able to obtain them from the NTP Pool operators. It contains a demo zone file, which we use as starting point. We refrain from doing source code analysis given it does not include the zone files, which are a product of the NTP Pool monitoring systems and the networking conditions of each server. Therefore, we need to an empirical measurement to determine the status of the servers and understanding how the mapping works in practice.

4.2.1 Reversing the NTP Pool DNS zone. We resort to reverse engineering the NTP Pool zone files (sample in Appendix §B). We start by using the demo zone file available with the `GeoDNS` source

code and populate it with servers that we have found with EnumV4 and EnumV6 experiments, in the following way:

(1) Generate a list of all NTP servers from EnumV4 and EnumV6 measurements
(2) Retrieve metadata (DNS zones) from each NTP server from the NTP Pool website
(3) Populate the demo zone file using the retrieved metadata

In the first step, we obtain 3,056 NTP server addresses. We then crawl each of them from the NTP Pool website using each their IP address. Each NTP server in the pool has a dedicated page (in the form of https://www.ntppool.org/scores/IP), which lists the zones the NTP server is associated. For example, the NTP Pool page for 95.217.188.206 shows that this NTP server is allocated to the global (`@`), `europe`, and Finland's `fi` zones [64]. Then, we assigned this particular IP address to these subzones in our reverse-engineered zone file. We repeat this process for all 3,056 IPv4 and 1,479 IPv6 addresses we found from EnumV4 and EnumV6.

In the `GeoDNS` zone files, each NTP server has a *weight* associated with it, which is derived from how much service capacity the volunteer wants to donate to the pool (*BW* in Figure 1). In practice, servers with higher weight values are picked more often. For example, a server with a 100 weight will be seen 100 times more often than a server with one weight. We demonstrate the weights influence in §C.

Our reverse-engineered zone file has 126 non-empty zones in total – all country and continent zones (we found no vendor zones using this method). We found 125 zones for IPv4 and 112 for IPv6. We also found many countries (101 for IPv4, 145 for IPv6) that have zero servers in their zones.

4.2.2 Validation. The next step consists in replaying the DNS queries from the EnumV4 experiment on our controlled environment. We use the reverse-engineered zone file on `GeoDNS` and use Maxmind's GeoLite2 country IP2location database [46] from 2021-08-24 – a required input by `GeoDNS` to operate.

Client setup: To replay the queries from EnumV4, we send spoofed IP packets (forged source IP addresses [22]), using a customized `Python` script, and run our experiment on our server disconnected from the Internet – so our spoofed packets cause no harm.

Collected datasets: we collect two datasets, namely, network traces (`pcap` files), and `GeoDNS` log files (Listing 1, which lists the metadata associated with each DNS query and response), both from the same Linux server. We refer to this experiment as EnumV4-emul.

By analyzing `GeoDNS` log files, we see how the mapping occurs: first, the client's geographical information is retrieved from MaxMind's database (country and continent). These are used to populate a list of *candidate* zones that can be used to answer this client, which is shown by the `Targets` tag (Listing 1). Then, the tag `LabelName` shows which zone the client has been mapped to. For this particular client, we see it could have gone to Israel (`il`), Asia, or the Global (`@`) zone, and it was ultimately mapped to Israel's zone. The logs do not show, however, which NTP servers were included in the DNS response. We analyzed the pcap files and confirmed they belong to the Israel's zone.

Results: For each VP (IP address from the Atlas in EnumV4), we compute two sets: S_{EnumV4} and $S_{EnumV4_{emul}}$, in which we list all NTP servers the VP has seen on each measurement – the experiment in the wild and our emulation. The latter we obtained from the pcap files. We then compare the sets for each VP. Table 4 shows the results. We see three main categories: Equal shows that zones and VPs matched perfectly in the wild and our controlled experiments. These comprise 2.2 thousand VPs from 93 zones. The second category is More, in which the VPs in our controlled experiment saw more NTP servers than those in the wild. These comprise most probes (7.2 thousand, 66 zones). We speculate this can be due to the use of uniform weights (1,000) in our emulation experiment, in which each server gets the same odds of being included in the response.

```
{ "Time": 1626941639825507800,
  "Origin": "2.pool.ntp.org.",
  "Name": "2.pool.ntp.org.",
  "Qtype": 1,
  "Rcode": 0,
  "Answers": 2,
  "Targets": ["il", "asia", "@"],
  "LabelName": "il",
  "RemoteAddr": "132.64.6.1",
  "ClientAddr": "132.64.6.1/32",
  "HasECS": false}

```

Listing 1. `GeoDNS` server log sample

Cat.		#Zones	#VPs
Equal	$S_{EnumV4_{emul}} = S_{EnumV4}$	93	2,265
More	$S_{EnumV4_{emul}} > S_{EnumV4}$	66	7,282
Fewer	$S_{EnumV4_{emul}} < S_{EnumV4}$	12	47

Table 4. Validation results per zone.

In the NTP Pool, however, these weights vary by a factor of 2,000. As such, our Emulation retrieves most if not all servers in the zone, while in the wild, the distribution would have been shifted to servers with higher weights (see Appendix §C).

The last category is the more concerning one: 47 VPs in our controlled experiment saw *fewer* NTP servers than in our emulation experiments. We believe this may be due to two reasons: their DNS traffic being intercepted and ultimately to send to resolvers elsewhere, and dynamic changes in the NTP Pool NTP server population along our measurements. Next, we cover the second reason.

4.3 NTP Pool monitoring system

The second reason is that the NTP Pool continuously monitors the volunteer's NTP servers. Poorly performing servers (unreachable, providing incorrect time data) have points deducted up to a threshold and are evicted from the NTP Pool zone file if they cross this threshold (10 points). While evicting servers from zones should not change much our results, they change in a specific case: if a country zone has a single server and the server is evicted. If this happens, then the client will, from that point on, be mapped to its respective continent zone, which is not empty (we found 101 country empty zones for IPv4 – §4.2),

This case covers 34 VPs that see only one NTP server in our experiment hosted in Cameroon, Guernsey, and Reunion – the latter two islands belonging to the United Kingdom and France, respectively. These VPs are mapped to a single NTP server in their zone that was eventually evicted from the NTP Pool zone due to poor performance. This caused these VPs to fallback to its continent zone (Europe), which has many servers.

We demonstrate that with VP 17580 located in Guernsey. The EnumV4 experiment (in the wild) shows that this probe sees, in total, 21 unique NTP servers – even though its associated territory zone (`gg`) zone has only one NTP server. We plot the responses seen by this VP in the wild in Listing 2. This VP initially receives a single NTP server in the DNS responses – 51.255.142.175 – an NTP server from Guernsey until 20:56. From 21:01 to 21:21, this VP receives 20 different NTP servers in 4 subsequent queries. These 20 servers belong to the `europe` zone, suggesting t this VP was mapped during this period to `europe` zone and not `gg` zone, which seem to have been empty.

While this shows that the probe sees more servers from Europe's zone, it does not show its country zone was empty at the same time. To show that, we analyze the scores associated with

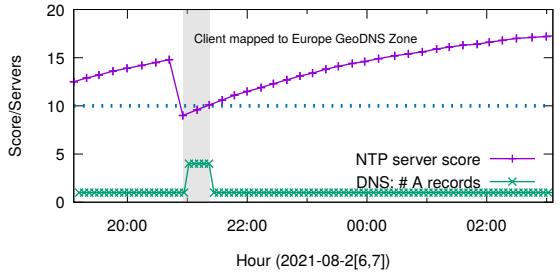


Fig. 5. NTP servers per DNS response from VP 17580 and server score from NTP Pool for server NTP server 51.255.142.175: low scores lead to NTP Pool eviction and fallback to the continent zone.

```
#time (UTC), DNS responses (A records)
20:54:41,51.255.142.175
20:56:45,51.255.142.175
21:01:36,37.221.193.210|149.156.70.60|185.57.191.229|94.16.114.254
21:06:49,213.239.234.28|194.58.204.148|95.215.175.2|54.36.152.158
21:06:49,78.36.18.184|138.201.16.22562.116.130.3|212.83.158.83
21:16:38,49.12.125.53|85.199.214.100|85.236.36.4|178.62.250.107
21:21:43,217.114.59.3|217.114.59.66|213.239.234.28|130.208.87.151
21:26:47,51.255.142.175
21:31:38,51.255.142.175
```

Listing 2. Atlas VP 17580 responses (2021-08-26).

this NTP server from the NTP Pool web page – file [63]) – a measurement carried independently from ours. We correlate our measurement results with the NTP Pool's server score logs, as seen in Figure 5. We see that this particular server score dropped below 10 – the minimum value for it to be used in the NTP Pool zones, otherwise a server is evicted – between 20:55:32 and 21:22:04 (2021-08-26, UTC). We show this in the gray area.

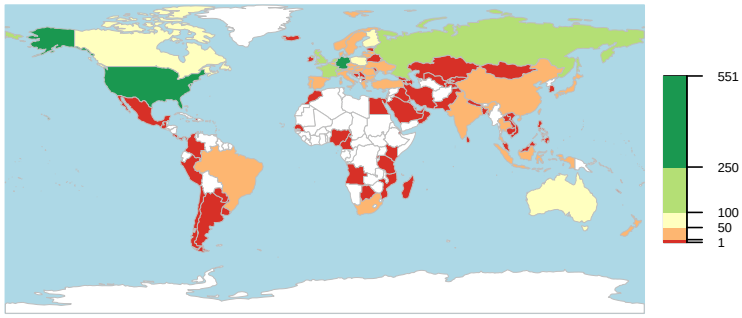
This period of scores lower than 10 coincides precisely when this VP receives 4 NTP servers per response (Listing 2). Given these low scores, we can infer that this NTP server was likely evicted from the `gg` zone in this period. Since this was the *only* server in the zone, `GeoDNS` mapped this VP to its respective Continent zone (`europe`). Once the NTP server's score surpasses 10 again, after 21:22, we see the client again receiving 1 NTP server in the DNS response, likely from the NTP server joining the `gg` zone again.

New monitoring system: On March 2023, the NTP Pool operators released a new monitoring system, which consists of multiple measurement servers across the globe instead of a single one in California [69]. (Its beta version has been evaluated in [38]). Each NTP server server is now evaluated by five monitoring servers instead of one, and the scores of the five are combined [68]. This prevents that failures on a single monitoring server wrongfully evicts well-performing servers. Whereas the scoring logic has change, the eviction process and reinsert has not.

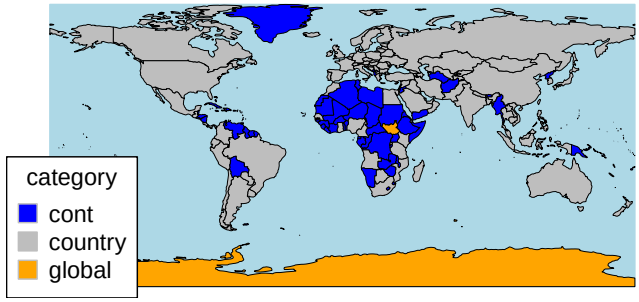
Vendor zones: (such as `debian.pool.ntp.org`) behave like the default zone: a client will be first mapped to its country of origin, and if its zone it is empty, to its continent. We show it in Appendix §D

5 PREDICTING MAPPINGS FOR ANY CLIENT IN THE WORLD

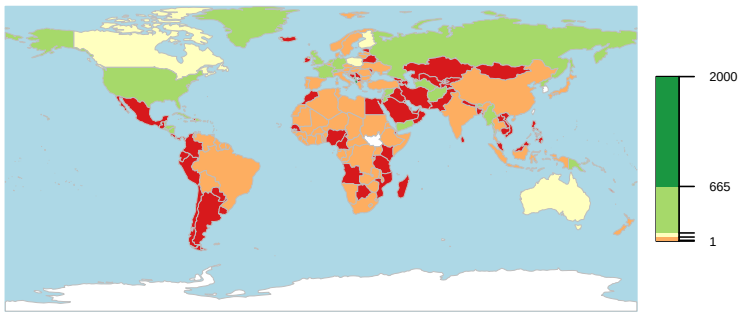
Now that we understand how the NTP Pool maps clients to NTP servers, we can determine, for any client in the world, the number and which NTP servers will be made available if it chooses to use the NTP Pool. This applies only for clients that either use the default pool zone (`pool.ntp.org`) or their vendors subzone (*e.g.,* `android.pool.ntp.org`).



(a) GeoDNS subzone sizes. Countries in white have zero servers.



(b) Client country and and GeoDNS subzone mapping.



(c) Client Visibility: number of NTP servers that a client sees from NTP Pool (IPv4)

Fig. 6. GeoDNS mappings.

5.1 Methodology

We first start with the analysis of all the GeoDNS DNS subzones we identified. For each country/territory subzone, we show in Figure 6a the number of NTP serves that its respective GeoDNS subzone has. For example, the United States' `us.pool.ntp.org` has 551 IPv4 NTP servers listed in it. In the same figure, all countries listed in white have zero servers in their zone.

As we have seen in §4.2, a client from a country which has subzone with more than one server will be then mapped to its country subzone. If its country zone is empty, then it will either fall back to the continent or global zone. To determine who serves those countries with no NTP servers,

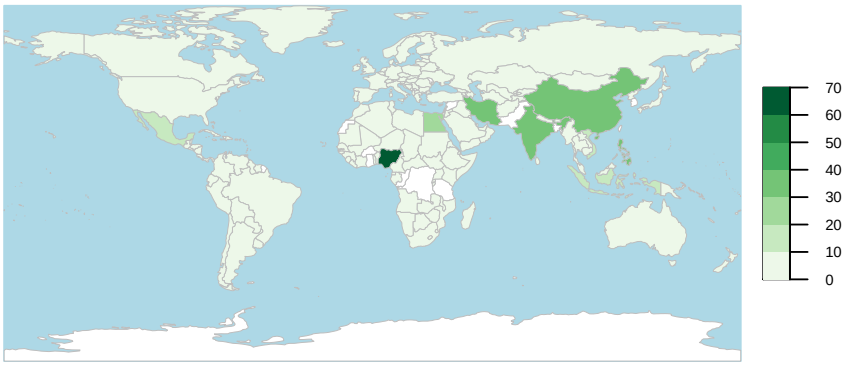


Fig. 7. Ratio of million Internet users per NTP server.

we obtained the mappings for each client in country C to which `GeoDNS` zone M it is mapped ($C \rightarrow$M). We start by first enumerating all countries and territories listed on Maxmind's database – the same database used by `GeoDNS`. We obtain 246 countries/territories and their first occurring IP address by querying Maxmind database for every /24 on the IPv4 space. We find that clients from countries/territories with no servers (white in Figure 6a) are mapped to their respective continent zone Figure 6b, except for 8 countries/territories are mapped to the global zone (in orange), including South Sudan and Antarctica.

Then, the next step consists in determining the *client visibility*, *i.e.,* the effective number of NTP servers each client will see from the NTP Pool. We query the NTP Pool authoritative serves 1000 times for each subozone – we choose a random authoritative server before each query, so we divide the load among the 29 available authoritative servers in order to minimize stress, and do frequent pauses between measurements. In total, we send 451k IPv4 queries and 337k IPv6 queries, which were carried on 2022-11-24. Then, we use the mappings we have obtained from all countries/territories and plot the clients's visiblity, *i.e.,*, the *effective* number of NTP servers from the 4k from the NTP Pool that get to server all clients within a country.

Figure 6c shows the client visibility results. In this figure, consider Algeria and Morocco have zero servers in their subzones (Figure 6a), so they are mapped to their continent zone (Figure 6b), and ultimately are served by up 40 NTP servers – which is the number of NTP serves in the Africa zone. US clients, which are mapped to the US zone, are served to up 551 servers. It's interesting to observe that countries that are mapped to the continent zone end up being server by more NTP servers than their neighbors that have servers in their own country – for example, Bolivia is mapped to the South America zone, which has 50 servers. A client in Argentina, in turn, despite being mapped to its own country servers, it is served by up to 10 servers.

We see also that Greenland was served by 665 servers from the North America zone in our measurements. Those who see more serves are countries mapped to the European continent zone (Monaco, Kosovo, and others): they are served by 2k NTP servers. (We leave countries mapped to the global zone in white, given there is no direct way to query the NTP Pool for their global zone).

Users per NTP server: next, we compute the number Internet users per NTP server for each country. This metric complements the previous one by showing an estimative of client population per each NTP server. To compute the number of users per NTP server, we divide a country's Internet population (obtained from the ITU entry [33]) by the number of servers that serve that zone. Figure 7 shows the results: a single NTP server provides service for Nigeria. China and India both have roughly 48 million Internet users per NTP Server.

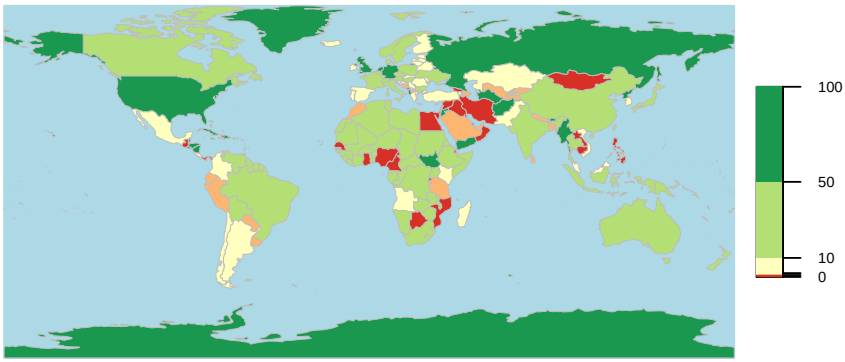


Fig. 8. Number of ASes serving clients from each country (IPv4). Red means 1 AS serving the entire country, orange is two countries. Green is 100 and more.

Bahrain	**Gibraltar**	Kuwait	Oman
Botswana	Guatemala	**Laos**	Panama
Cameroon	Haiti	Lebanon	Philippines
Curaçao	**Iran**	Macao	Qatar
Djibouti	Iraq	Mongolia	Rwanda
Egypt	Israel	Mozambique	Senegal
Georgia	Cambodia	Nigeria	

Table 5. Countries served by a single time provider: Clouflare and other AS (bold)

5.1.1 Time Providers per country. Next we compute the number of time service providers for each client country C. Given that a single time provider may have multiple servers, we want to know the consequences of a time provider failure.

To map NTP servers to providers, we use their associated AS number, retrieved from CAIDA's IP to AS maps [13]. Then, for each client country, we compute the number of unique ASes serving it from all NTP servers available.

Figure 8 shows a world map of the number of ASes serving each country. We see that countries in red are served by only a single AS – these include all countries shown in Table 5 for IPv4. These countries are only served by Cloudflare's time service – except for those in bold.

Countries in orange, in Figure 8 have all their NTP Pool clients served by only two time providers. These include Uruguay, Saudi Arabia, and Tanzania. We see that most countries are served by 10 to 50 ASes, and a few are served by more than 100 (*e.g.,* US, Russia, and Germany).

Ultimately, this single-provider dependency is an example of centralization and consolidation on the Internet [3–5, 35, 36, 52, 81, 84]. In this case, the main drive is not large companies dominating a market – it is rather combined with strict client/server mappings by `GeoDNS`, and the lack of volunteers to provide free service in a country.

Takeaway: despite thousands of servers being available to the NTP Pool, users from many countries can only receive time information from a limited set of NTP servers and providers, given the number of volunteer servers available in the country and how strict `GeoDNS` is.

6 GEODNS MAPPINGS: ARE THEY SOUND AND THEIR IMPLICATIONS

In §5, we demonstrated that `GeoDNS` employs a stringent client-to-NTP server mapping approach, which imposes limitations on the number of NTP servers available for serving clients. We contacted

the NTP Pool operators and they argued the idea behind this mappings is twofold: reduce the risk of asymmetric routing and minimize packet loss [7] .

Asymmetric routing occurs when incoming and outgoing network traffic for a given connection follows different paths, can have a significant impact NTP and potentially lead to synchronization issues and time inaccuracies [26] (NTP assumes symmetric paths). It is questionable where keep clients within a country can reduce route assymetry. Recent work has shown that most Internet paths are assymetrical [87], so it is not a NTP Pool only problem. Binding client to countries does not take into account the large diversity in country sizes – a client in Belgium may be geographically and topologically closer to NTP servers located in neighboring Germany, while a client in Honolulu being served by a NTP server in Boston (both in the US but 8.2k km apart).

Packet loss can also impact clock synchronization, given NTP responses may simply not arrive. To determine whether these packet loss concerns are sound and to determine if clients can consistently access accurate time information from remote servers, we carry out experiments using RIPE Atlas probes(§6.1). Then, we discuss the security implications of current `GeoDNS` mappings (§6.2). Lastly, we put forth a set of recommendations for NTP Pool regarding the implementation of a less restrictive allocation system (expounded upon in §6.3).

6.1 Can far away NTP services provide good time information?

Is it possible for clients to receive accurate time service from servers located in distant regions, rather than being limited to strict in-country mappings? To examine this hypothesis, we undertake an experiment employing 132 RIPE Atlas probes as vantage points. These probes are drawn from 21 countries that are presently solely served by Cloudflare (highlighted in bold within Table 5). It's worth noting that the majority of these countries are situated in Africa, the Middle East, and Southeast Asia, as opposed to regions like the United States or Europe, where more favorable outcomes might be anticipated.

Our objective is to ascertain whether clients in these countries experience no significant packet loss among all servers. To establish a baseline, we compare the service offered by their current sole time provider, Cloudflare, with five additional NTP servers from the NTP Pool. We choose one server per continent, with our choice based on the NTP server that exhibits the highest frequency per zone, as shown in §5. We configure these Atlas VPs to conduct queries every 30 minutes over the course of one week (Dec. 16–23, 2022). This extended observation period enhances our chances of identifying potential failures.

The details of our experiments are consolidated in Table 6. It provides an overview of the the specific NTP servers for which we configured Atlas probes to sent NTP queries. Over the span of one week, we received a total of 33,000 to 36,000 queries per NTP server, with one notable exception being the NTP server located in South America. This server received 21,000 valid responses but from a reduced pool of only 90 probes.

Lack of NTP Responses: We compute, for each Atlas Probe (VP), the ratio of NTP queries that receive no response. It's important to note that RIPE Atlas does not provide specific reasons for these lack of responses; it could be due to timeouts, filtering, or other factors [37].

In Figure 9, we show a CDF of the Atlas VPs and their respective rate of unanswered queries. We see that *90% of our VPs have no queries loss* for the Cloudflare, Europe and North America servers, despite many of the VPs being located in Africa, Asia, and Middle East. For the Asia NTP server, we see that 80% of the VPs have up to 10% unanswered queries. Only the South American server has not very good results: 40% of VPs have more than 50% of unanswered queries.

Even though we cannot determine the reasons why this South American server performed worse than the others for the same VPs, we see in Figure 10 that the same VPs that failed to retrieve responses from the South American server could retrieve responses from the other servers. In this

Provider	Cloudflare	Africa	Asia	Europe	North Am.	South Am.
NTP Server	162.159.200.123	41.220.128.73	144.24.146.96	94.198.159.11	45.33.65.68	186.155.28.147
# Atlas Probes	132	132	132	132	132	132
Valid	131	130	130	130	130	90
Countries	21	21	21	21	21	21
Valid	21	21	21	21	21	16
Valid Queries	36,501	34,835	33,145	35,763	35,918	21,540
Avg. Offset (s)	1.96	1.97	1.78	1.97	2.03	1.66
Med. Offset (s)	0	0	0	0	0	0

Table 6. Evaluating NTP servers from clients located in clients only served by Cloudflare. Datasets: [76]

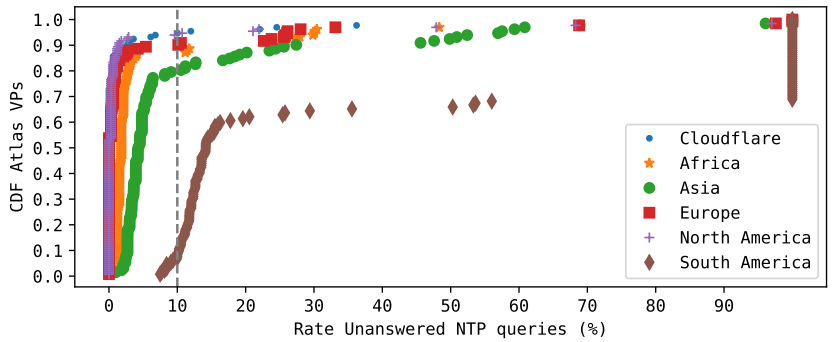


Fig. 9. Unanswered queries (%), per NTP server

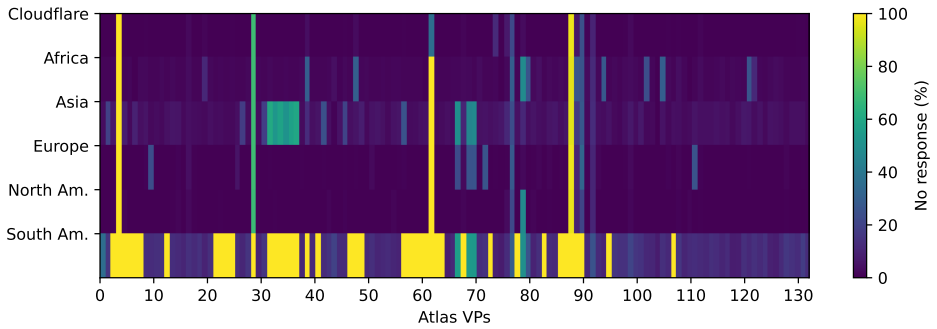


Fig. 10. Unanswered queries (%), per NTP server, for each Atlas VP.

figure, we show the ratio of non responses per probe (each entry in x axis is a Atlas probe) and per time server (y axis). As such, we can disregard issues with specific Atlas probes, as they could retrieve responses from other servers.

In practice, these client, if they ran a NTP client served by these NTP servers, would likely disregard the South American server (§2), so it would cause no harm.

We next set out to compute the quality of timing data provided by each server. For every NTP response, we extract its offset value, indicating the time difference in seconds between the local probe clock the time provided by the NTP server. To mitigate the effects of clock drifting, we exclusively consider results from probes whose clocks were synchronized within a maximum

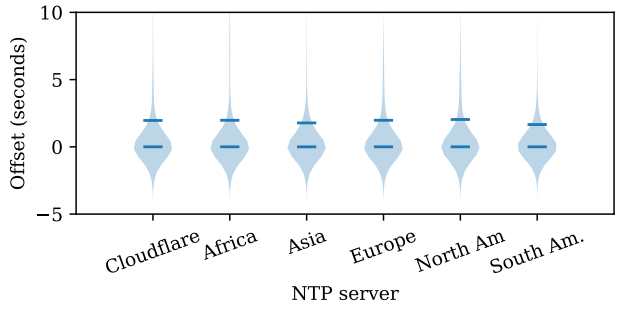


Fig. 11. Offset distribution, with bars showing average and median. We see that servers in other regions can provide similar quality service for these probes – all have similar offset values (<2 seconds).

deviation of five minutes by the time our NTP queries were initiated. (Atlas probes are designed to synchronize their clocks approximately every three minutes [29]).

Subsequently, we consolidate all offset data points for each NTP server and compute their aggregate statistics. Our analysis reveals that the average offset for all NTP servers remains under 2.1 seconds (Table 6), which is considered a favorable result. This finding is visually represented in Figure 11, where the majority of offsets fall within the range of ± 2 seconds for all probed servers, signifying reliable performance.

Despite the scale of this experiment, we see that that NTP servers located in diverse geographical regions and continents can provide dependable time services for clients located elsewhere. Hence, this experiment serves to underscore that the assumption of tethering clients exclusively to servers within their own countries is overly restrictive and opens up opportunities for potential attacks.

6.2 Security Implications

The stringent mappings enforced by `GeoDNS` introduce unwarranted risks for NTP Pool clients. It has first been proposed to hijack traffic from an entire country [80, §6.2]. It consists of hosting a single server on countries where there are currently no NTP servers listed in the NTP Pool. Whereas the authors do not confirm the monopoly of traffic as our experiments (and do not point to a particular example, as we did with our analysis in §4.3 with Guernsey), they were the first to identify this vulnerability.

Notably, all countries shaded in blue in Figure 6b, including but not limited to Albania, Bolivia, Tunisia, Namibia, Venezuela, Guatemala meet this criteria. This encompasses a total of 101 countries and territories for IPv4 and 145 for IPv6, collectively representing approximately 260 million Internet users.

These mappings can also be exploited to also exploit countries with multiple NTP servers listed in NTP Pool [73]. The attack consists in introducing numerous NTP servers into densely populated zones. This tactic aims to create a race condition, thereby increasing the likelihood of clients being served by their designated NTP servers (while maximizing the *BW* value, as shown in Figure 1). Particularly, in countries with fewer servers, the task of diverting a significant portion of the traffic becomes notably more manageable.

Both of these methods can potentially facilitate time-shifting attacks on clients. In this scenario, an attacker can distinguish between real clients and NTP Pool monitoring servers by initially configuring their NTP server to operate in "monitor only" mode. In a manner akin to previous studies such as [73] and [38], the attacker can furnish accurate time readings to the NTP Pool's

monitoring servers while deliberately providing incorrect time to other clients. This manipulation effectively subverts the NTP Pool's eviction system [73].

In conclusion, the stringent mappings enforced by `GeoDNS` introduce unwarranted risks for NTP Pool clients. If clients had access to a broader selection of servers, potentially numbering in the thousands rather than being limited to a few servers within their respective countries, it could substantially decrease the susceptibility to attacks as identified in prior research. To enable such a scenario, it becomes crucial for distant NTP servers to consistently offer reliable timing information.

6.3 Recommendations

Our results reveal that the overly strict mapping by `GeoDNS` introduces more risks than benefits, primarily due to potential packet loss issues (we leave asymmetric routing for future work). Discussions with the NTP Pool operators [7] have been productive. They acknowledge the need for changes to enhance system security, which they plan to implement. Additionally, the use of DNS for load balancing across all volunteer servers must be considered when designing the new system. One potential solution could involve eliminating country zones in favor of larger continent zones.

7 ETHICS, PRIVACY, AND DISCLOSURE

Our paper has three ethical concerns: avoiding negative consequences of our measurements in both clients (VPs) and servers, respecting the privacy of these VPs, and disclosing our findings to the NTP Pool operators.

Responsible experimentation: We design our experiments to minimize the impact on clients, measurement platform (RIPE Atlas), and measured DNS and Web servers. Whenever we use RIPE Atlas VPs, we use safe query rates (1 DNS query per 5, 10, or 30 minutes, depending on the experiment). We also crawl web pages related to each NTP server on the NTP Pool website – fewer than 5 thousand pages. To minimize impact, we rate-limit our crawler to 1 webpage/second. Part of our experiments was done in an isolated network (§5), so no traffic was sent to the NTP Pool servers.

Privacy: We found cases of RIPE VPs that seem to use overseas DNS servers (which may be due to trying to bypass government censorship or DNS hijack). DNS hijack in RIPE VPs has been known for years [53, 85]. While we cannot determine which is the reason, we do not disclose details about these cases to protect these VPs and their owners, who volunteer to host them.

Disclosure to NTP Pool operators: We shared multiple versions of this manuscript with the NTP Pool operators, who provided valuable feedback. We also have exchanged e-mails and discussion on the public NTP Pool forum [7] . While we did not disclose any new attack models (they have been previously presented), we provide data showing how current mappings mechanisms are too strict. We expect `GeoDNS` can be improved to introuduce more diversity in the number of servers clients get to see.

8 RELATED WORK

NTP Pool measurement studies: our study is the most comprehensive evaluating the inner works and popularity of the NTP Pool. A previous study has also crawled the NTP Pool authoritative servers to enumerate them [80]. They used a single VP in Germany to query the NTP Pool authoritative servers. We scrutinize the inner works of `GeoDNS` and by unveiling how the NTP Pool monitors, evicts, and cleans its zone (§4), and show how clients all over the world see the NTP Pool (§5).

NTP Pool vulnerabilities: Previous studies have shown how the NTP Pool can be exploited to hijack traffic from countries with empty zones [80] – they run a brief experiment on IPv6 – but they not confirm the traffic monopoly. Our measurements from §4.3 confirms it is feasible and

demonstrate traffic monopoly, and we provide open datasets (by RIPE Atlas). Another study has shown that an attacker can also control traffic by introducing multiple NTP servers into densely populated zones. The latter aims to create a race condition, thereby increasing the likelihood of clients being served by their malicious NTP servers [73] to perform time-shift attacks.

Another work has identified vulnerabilities with the NTP Pool monitoring system [38]. The authors present multiple attack methods against the monitoring servers – which include BGP hijack and delay attacks. Another attack model they cover assumes an attacker controls one of the pool monitoring servers.

While not directly related to ours, there are several other studies that focused on NTP security. They either covered the NTP protocol vulnerabilities [42–44], or show how NTP clients can be vulnerable to malicious time servers [18], or how NTP servers can be used in distributed denial-of-service (DDoS) amplification attacks [16] (in which spoofed queries are sent with the source address of the target, which then receives unsolicited traffic), or study off-path attacks using DNS cache poisoning [34]. While related to ours, they do not focus on the NTP Pool itself, as we do.

With regards to NTP traffic characterization, a previous studies have characterized traffic at the NIST's NTP servers [82] or running many NTP servers that are part of the NTP Pool [80]. We analyze Root DNS traffic to determine how popular time providers are and briefly cover 24 of traffic of a NTP server listed in the NTP Pool.

While NTP traffic is transmitted in clear (and thus prone to tampering), Network Time Security (NTS) [23] protocol provides client-server encryption and therefore eliminates the possibility of tampering between client and server. NTS, however, is currently not supported by the NTP Pool.

9 CONCLUSION

The NTP Pool has played a vital role in ensuring accurate timekeeping on the Internet. Similar to Wikipedia, it operates as a community-driven initiative. Our research has demonstrated that it is the most widely used time service on the Internet. We extend our gratitude to the volunteers who have dedicated their time and resources to support this endeavor over the past two decades.

In our investigation, we have delved into the intricate workings of the NTP Pool and highlighted an aspect of concern. While the strict client/NTP server mapping was implemented with good intentions (avoid packet loss and traffic assymetry), we have shown that these fears may be unfonded, and that these mappings can be relaxed to prevent attackers from carrying out current vulnerabilities.

In light of these findings, we raise awareness among NTP Pool operators regarding these shortcomings. Furthermore, we view our work as a call to action, urging the community to increase the number of volunteers, allocate additional resources, and enhance server diversity in each country and region. These measures are essential to effectively handle the billions of requests handled daily by the NTP Pool infrastructure.

Through our research, disclosures, and recommendations, we hope to foster a greater understanding of the challenges faced by the NTP Pool and encourage necessary improvements for the continued reliability and security of public and free time synchronization services on the Internet.

REFERENCES

[1] Apple. 2021. Apple NTPService. time.apple.com.

[2] 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

[3] J. Arkko. 2019. Centralised Architectures in Internet Infrastructure. Internet Draft. https://tools.ietf.org/html/draft-arkko-arch-infrastructure-centralisation-00

[4] Jari Arkko. 2020. The influence of Internet architecture on centralised versus distributed Internet services. *Journal of Cyber Policy* 5, 1 (2020), 30–45. https://doi.org/10.1080/23738871.2020.1740753

[5] Arkko, Jari and Tramme, B. and Nottingham, M and Huitema, C and Thomson, M. and Tantsura, J. and ten Oever, N. 2019. Considerations on Internet Consolidation and the Internet Architecture. Internet Draft. https://tools.ietf.org/html/draft-arkko-iab-internet-consolidation-02

[6] Ask Bjørn Hansen. 2021. GeoDNS servers. https://github.com/abh/geodns/.

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

[8] Rushvanth Bhaskar. 2022. *A Day in the Life of NTP: Analysis of NTPPool Traffic.* Master's thesis. University of Twente and SIDN Labs, Enschede and Arnhem, The Netherlands. Master's thesis.

[9] Leo Bicknell. 2012. NTP Issues Today. https://mailman.nanog.org/pipermail/nanog/2012-November/053449.html.

[10] S. Bortzmeyer, R. Dolmans, and P. Hoffman. 2021. *DNS Query Name Minimisation to Improve Privacy.* RFC 9156. IETF. http://tools.ietf.org/rfc/rfc9156.txt

[11] Physikalisch Technische Bundesanstalt. 2022. FDCF77 - PTB.de. (Nov. 5 2022). https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/ag-442/dissemination-of-legal-time/dcf77.html

[12] 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

[13] CAIDA. 2022. Index of /datasets/routing/routeviews-prefix2as. https://publicdata.caida.org/datasets/routing/routeviews-prefix2as.

[14] Sebastian Castro, Duane Wessels, Marina Fomenkov, and Kimberly Claffy. 2008. A Day at the Root of the Internet. *ACM Computer Communication Review* 38, 5 (April 2008), 41–46.

[15] Cloudflare. 2021. Cloudflare Time Service. https://www.cloudflare.com/time/.

[16] 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

[17] Wouter B de Vries, Quirin Scheitle, Moritz Müller, Willem Toorop, Ralph Dolmans, and Roland van Rijswijk-Deij. 2019. A First Look at QNAME Minimization in the Domain Name System. In *International Conference on Passive and Active Network Measurement*. Springer, 147–160.

[18] Omer Deutsch, Neta Rozen Schiff, Danny Dolev, and Michael Schapira. 2018. Preventing (Network) Time Travel with Chronos.. In *NDSS*.

[19] T. Dierks and E. Rescorla. 2008. *The Transport Layer Security (TLS) Protocol Version 1.2.* RFC 5246. IETF. http://tools.ietf.org/rfc/rfc5246.txt

[20] DNS OARC. 2022. DITL Traces and Analysis. https://www.dns-oarc.net/index.php/oarc/data/ditl/.

[21] R. Droms. 1997. *Dynamic Host Configuration Protocol.* RFC 2131. IETF. http://tools.ietf.org/rfc/rfc2131.txt

[22] Toby Ehrenkranz and Jun Li. 2009. On the state of IP spoofing defense. *ACM Transactions on Internet Technology (TOIT)* 9, 2 (2009), 1–29.

[23] 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

[24] R. Gayraud and B. Lourdelet. 2010. *Network Time Protocol (NTP) Server Option for DHCPv6.* RFC 5908. IETF. http://tools.ietf.org/rfc/rfc5908.txt

[25] Google. 2021. Google Public NTP. https://developers.google.com/time.

[26] Mohammad Javad Hajikhani, Thomas Kunz, and Howard Schwartz. 2016. A Recursive Method for Clock Synchronization in Asymmetric Packet-Based Networks. *IEEE/ACM Transactions on Networking* 24, 4 (2016), 2332–2342. https://doi.org/10.1109/TNET.2015.2462772

[27] Stewart Hampton. 2018. Five Dangers of Poor Network Timekeeping + Easy and Cost Effective Solutions (Part 2 of 10). (Sept. 5 2018). https://www.microsemi.com/blog/2018/09/05/five-dangers-of-poor-network-timekeeping-easy-and-cost-effective-solutions-to-avoid-networks-fall-out-of-sync-part-2-of-10/

[28] P. Hoffman, A. Sullivan, and K. Fujiwara. 2019. *DNS Terminology.* RFC 8499. IETF. http://tools.ietf.org/rfc/rfc8499.txt

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

[30] 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

[31] IEEE. 2002. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std. 1588-2002* (2002). https://standards.ieee.org/ieee/1588/3140/

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

[33] ITU. 2023. Statistics. https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

[34] 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
[35] Cecilia Kang and David McCabe. 2020. Lawmakers, United in Their Ire, Lash Out at Big Tech's Leaders. *New York Times* (July. 29 2020). https://www.nytimes.com/2020/07/29/technology/big-tech-hearing-apple-amazon-facebook-google.html
[36] Aqsa Kashaf, Vyas Sekar, and Yuvraj Agarwal. 2020. Analyzing Third Party Service Dependencies in Modern Web Services: Have We Learned from the Mirai-Dyn Incident?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 634–647.
[37] Robert Kisteleki. 2023. NTP empty results ('result': ['x': '*']). https://www.ripe.net/ripe/mail/archives/ripe-atlas/2023-October/005607.html.
[38] 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 *30th USENIX Security Symposium*. https://www.usenix.org/conference/usenixsecurity23/presentation/kwon
[39] Leslie Lamport. 2019. *Time, Clocks, and the Ordering of Events in a Distributed System*. Association for Computing Machinery, New York, NY, USA, 179–196. https://doi.org/10.1145/3335772.3335934
[40] Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, and Kimberly Claffy. 2007. Two Days in the Life of the DNS Anycast Root Servers. In *Proceedings of the International conference on Passive and Active Measurements (PAM)*. 125–134.
[41] Jonathan Magnusson, Moritz Müller, Anna Brunstrom, and Tobias Pulls. 2023. A Second Look at DNS QNAME Minimization. In *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*. Springer, 496–521.
[42] 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).
[43] Aanchal Malhotra and Sharon Goldberg. 2016. Attacking NTP's Authenticated Broadcast Mode. *SIGCOMM Comput. Commun. Rev.* 46, 2 (may 2016), 12–17.
[44] 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.
[45] 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.
[46] Maxmind. 2021. Maxmind. http://www.maxmind.com/
[47] Microsoft. 2021. Microsoft NTP Service. http://time.windows.com.
[48] D. Mills. 2006. *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*. RFC 4330. IETF. http://tools.ietf.org/rfc/rfc4330.txt
[49] 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
[50] P.V. Mockapetris. 1987. *Domain names - concepts and facilities*. RFC 1034. IETF. http://tools.ietf.org/rfc/rfc1034.txt
[51] Giovane C. M. Moura, Sebastian Castro, Wes Hardaker, Maarten Wullink, and Cristian Hesselman. 2020. Clouding up the Internet: How Centralized is DNS Traffic Becoming?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 42–49.
[52] Giovane C. M. Moura, Sebastian Castro, Wes Hardaker, Maarten Wullink, and Cristian Hesselman. 2020. Clouding up the Internet: How Centralized is DNS Traffic Becoming?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 42–49. https://doi.org/10.1145/3419394.3423625
[53] 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
[54] Giovane C. M. Moura, John Heidemann, Ricardo de O. Schmidt, and Wes Hardaker. 2019. Cache Me If You Can: Effects of DNS Time-to-Live. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Amsterdam, the Netherlands, 101–115. https://doi.org/10.1145/3355369.3355568
[55] Giovane C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt, and Marco Davids. 2018. When the Dike Breaks: Dissecting DNS Defenses During DDoS. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Boston, MA, USA, 8–21. https://doi.org/10.1145/3278532.3278534
[56] Giovane C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt, and Marco Davids. 2018. *When the Dike Breaks: Dissecting DNS Defenses During DDoS (extended)*. Technical Report ISI-TR-725. USC/Information Sciences Institute. https://www.isi.edu/%7ejohnh/PAPERS/Moura18a.html

[57] Moritz Müller, Giovane C. M. Moura, Ricardo de O. Schmidt, and John Heidemann. 2017. Recursives in the Wild: Engineering Authoritative DNS Servers. In *Proceedings of the ACM Internet Measurement Conference*. ACM, London, UK, 489–495. https://doi.org/10.1145/3131365.3131366

[58] Network Time Foundation. 2022. Download NTP . https://doc.ntp.org/downloads/.

[59] C. Neuman, T. Yu, S. Hartman, and K. Raeburn. 2005. *The Kerberos Network Authentication Service (V5).* RFC 4120. IETF. http://tools.ietf.org/rfc/rfc4120.txt

[60] NIST. 2022. NIST Internet Time Service (ITS). (Nov. 5 2022). https://www.nist.gov/pml/time-and-frequency-division/time-distribution/internet-time-service-its

[61] NTP Pool. 2021. All Pool Servers. https://www.ntppool.org/zone.

[62] NTP Pool. 2021. Argentina — ar.pool.ntp.org. https://www.ntppool.org/zone/ar.

[63] NTP Pool. 2021. pool.ntp.org: statistics for 51.255.142.175 . https://www.ntppool.org/scores/51.255.142.175/.

[64] NTP Pool. 2021. pool.ntp.org: Statistics for 95.217.188.206. https://www.ntppool.org/scores/95.217.188.206.

[65] NTP Pool. 2021. pool.ntp.org: the internet cluster of ntp servers. https://www.ntppool.org/en/.

[66] NTP Pool. 2021. The NTP Pool for vendors. https://www.ntppool.org/en/vendors.html.

[67] NTP Pool. 2022. How do I join pool.ntp.org? https://www.ntppool.org/en/join.html.

[68] NTP Pool. 2023. Monitoring System - Technical details. https://news.ntppool.org/docs/monitoring/.

[69] NTP Pool. 2023. NTP Pool Monitoring v2. https://news.ntppool.org/2023/03/ntp-pool-monitoring-v2/.

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

[71] United States Naval Observatory. 2022. Information about NTP, the time backbone of the Internet. (Nov. 5 2022). https://www.cnmoc.usff.navy.mil/Our-Commands/United-States-Naval-Observatory/Precise-Time-Department/Network-Time-Protocol-NTP/

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

[73] 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).

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

[75] RIPE NCC. 2021. RIPE Atlas Measurement IDS. https://atlas.ripe.net/measurements/ID. , where ID is the experiment ID: EnumV4: 32025718, EnumV6: 32058440, ArgV4: 31789516, ArgV4-Emul:31830680, ArgV4-Android: 31992051, DE-Android:31970486, ArgV6:32001506.

[76] RIPE NCC. 2023. RIPE Atlas Measurement IDS. https://atlas.ripe.net/measurements/ID. , where ID is the experiment ID: Cloudflare: 47865355, Africa: 47867480, Asia:47867358, Europe: 47867632, North America:47867336, South America:47867316:.

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

[78] RIPE Network Coordination Centre. 2020. RIPE Atlas. https://atlas.ripe.net.

[79] Root Server Operators. 2021. Root DNS. http://root-servers.org/.

[80] 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

[81] Bruce Schneier. 2018. Censorship in the Age of Large Cloud Providers. https://www.schneier.com/essays/archives/2018/06/censorship_in_the_ag.html

[82] Jeff A Sherman and Judah Levine. 2016. Usage analysis of the NIST internet time service. https://tf.nist.gov/general/pdf/2818.pdf. *Journal of Research of the National Institute of Standards and Technology* 121 (2016), 33. https://tf.nist.gov/general/pdf/2818.pdf

[83] Jeff A. Sherman and Judah Levine. 2016. Usage Analysis of the NIST Internet Time Service. *Journal of Research of the National Institute of Standards and Technology* 121 (March 2016), 33. https://doi.org/10.6028/jres.121.003

[84] Internet Society. 2019. Consolidation in the Internet Economy. https://future.internetsociety.org/2019/

[85] Stéphane Bortzmeyer. 2015. DNS Censorship (DNS Lies) As Seen By RIPE Atlas. https://labs.ripe.net/author/stephane_bortzmeyer/dns-censorship-dns-lies-as-seen-by-ripe-atlas/.

[86] Ubuntu. 2023. Ubuntu NTP Service. https://ubuntu.com/server/docs/network-ntp.

[87] 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

[88] Adrian von Bidder. 2003. ntp DNS round robin experiment. https://groups.google.com/g/comp.protocols.time.ntp/c/cShrN7imCJ0.

A LIST OF TIME PROVIDERS AND SERVERS

Listing 3 shows the list of time providers and their respective time services domain names used in Table 1. We built this list based on a public repository on Github[3].

```
{
    "Apple": [
        "time.apple.com",
        "time.asia.apple.com",
        "time.euro.apple.com",
        "time1.apple.com",
        "time2.apple.com",
        "time3.apple.com",
        "time4.apple.com",
        "time5.apple.com",
        "time6.apple.com",
        "time7.apple.com"
    ],
    "Cloudflare": [
        "time.cloudflare.com"
    ],
    "Facebook": [
        "time.facebook.com",
        "time1.facebook.com",
        "time2.facebook.com",
        "time3.facebook.com",
        "time4.facebook.com",
        "time5.facebook.com"
    ],
    "Google": [
        "time.android.com",
        "time.google.com",
        "time1.google.com",
        "time2.google.com",
        "time3.google.com",
        "time4.google.com"
    ],
    "Microsoft": [
        "time.windows.com"
    ],
    "NIST": [
        "time-a-b.nist.gov",
        "time-a-g.nist.gov",
        "time-a-wwv.nist.gov",
        "time-b-b.nist.gov",
        "time-b-g.nist.gov",
        "time-b-wwv.nist.gov",
        "time-c-b.nist.gov",
        "time-c-g.nist.gov",
        "time-c-wwv.nist.gov",
        "time-d-b.nist.gov",
        "time-d-g.nist.gov",
        "time-d-wwv.nist.gov",
        "time-e-b.nist.gov",
        "time.nist.gov",
        "utcnist.colorado.edu.",
        "utcnist2.colorado.edu"
    ],
    "NTP Pool": [
        "pool.ntp.org",
        "*.pool.ntp.org"
    ],
    "Rest": [
        "asynchronos.iiss.at",
        "chime1.surfnet.nl",
        "clepsydra.dec.com",
        "clepsydra.hpl.hp.com",
        "clepsydra.labs.hp.com",
        "clock.isc.org",
        "clock.nyc.he.net",
        "clock.sjc.he.net",
        "clock.uregina.ca",
        "cronos.cenam.mx",
```

[3] https://gist.github.com/mutin-sa/eea1c396b1e610a2da1e5550d94b0453

```
        "gbg1.ntp.se",
        "gbg2.ntp.se",
        "gnomon.cc.columbia.edu",
        "gps.layer42.net",
        "hora.roa.es",
        "minuto.roa.es",
        "mizbeaver.udel.edu",
        "mmo1.ntp.se",
        "mmo2.ntp.se",
        "navobs1.gatech.edu",
        "navobs1.oar.net",
        "navobs1.wustl.edu",
        "nist1.symmetricom.com",
        "now.okstate.edu",
        "ntp-ca.stygium.net",
        "ntp-galway.hea.net",
        "ntp-s1.cise.ufl.edu",
        "ntp.atomki.mta.hu",
        "ntp.colby.edu",
        "ntp.dianacht.de",
        "ntp.fiord.ru",
        "ntp.fizyka.umk.pl",
        "ntp.gsu.edu",
        "ntp.i2t.ehu.eus",
        "ntp.ix.ru",
        "ntp.lcf.mx",
        "ntp.mobatime.ru",
        "ntp.nat.ms",
        "ntp.neel.ch",
        "ntp.neu.edu.cn",
        "ntp.nic.cz",
        "ntp.nict.jp",
        "ntp.nsu.ru",
        "ntp.ntsc.ac.cn",
        "ntp.qix.ca",
        "ntp.ripe.net",
        "ntp.rsu.edu.ru",
        "ntp.se",
        "ntp.shoa.cl",
        "ntp.time.in.ua",
        "ntp.time.nl",
        "ntp.vsl.nl",
        "ntp.yycix.ca",
        "ntp0.as34288.net",
        "ntp0.nl.uu.net",
        "ntp1.as34288.net",
        "ntp1.fau.de",
        "ntp1.hetzner.de",
        "ntp1.inrim.it",
        "ntp1.jst.mfeed.ad.jp",
        "ntp1.net.berkeley.edu",
        "ntp1.niiftri.irkutsk.ru",
        "ntp1.nl.uu.net",
        "ntp1.oma.be",
        "ntp1.ona.org",
        "ntp1.qix.ca",
        "ntp1.stratum1.ru",
        "ntp1.time.nl",
        "ntp1.usv.ro",
        "ntp1.vniiftri.ru",
        "ntp2.fau.de",
        "ntp2.hetzner.de",
        "ntp2.inrim.it",
        "ntp2.jst.mfeed.ad.jp",
        "ntp2.net.berkeley.edu",
        "ntp2.niiftri.irkutsk.ru",
        "ntp2.oma.be",
        "ntp2.qix.ca",
        "ntp2.stratum1.ru",
        "ntp2.stratum2.ru",
        "ntp2.time.in.ua",
        "ntp2.time.nl",
        "ntp2.vniiftri.ru",
        "ntp21.vniiftri.ru",
        "ntp3.hetzner.de",
        "ntp3.jst.mfeed.ad.jp",
        "ntp3.stratum1.ru",
        "ntp3.stratum2.ru",
        "ntp3.time.in.ua",
        "ntp3.usv.ro",
        "ntp3.vniiftri.ru",
```

```
        "ntp4.stratum1.ru",
        "ntp4.stratum2.ru",
        "ntp4.vniiftri.ru",
        "ntp5.stratum1.ru",
        "ntp5.stratum2.ru",
        "ntps1-0.cs.tu-berlin.de",
        "ntps1-0.uni-erlangen.de",
        "ntps1-1.cs.tu-berlin.de",
        "ntps1-1.uni-erlangen.de",
        "ntps1.pads.ufrj.br",
        "ntpstm.netbone-digital.com",
        "otc1.psu.edu",
        "ptbtime1.ptb.de",
        "ptbtime2.ptb.de",
        "rackety.udel.edu",
        "rustime01.rus.uni-stuttgart.de",
        "rustime02.rus.uni-stuttgart.de",
        "sesku.planeacion.net",
        "sth1.ntp.se",
        "sth2.ntp.se",
        "stratum1.net",
        "svl1.ntp.se",
        "svl2.ntp.se",
        "t2.timegps.net",
        "tempus1.gum.gov.pl",
        "tempus2.gum.gov.pl",
        "tick.usask.ca",
        "time-a.as43289.net",
        "time-b.as43289.net",
        "time-c.as43289.net",
        "time.esa.int",
        "time.fu-berlin.de",
        "time.nrc.ca",
        "time.ufe.cz",
        "time1.esa.int",
        "time1.stupi.se",
        "timehost.lysator.liu.se",
        "timekeeper.isi.edu",
        "tock.usask.ca",
        "ts1.aco.net",
        "ts2.aco.net",
        "vniiftri.khv.ru",
        "vniiftri2.khv.ru",
        "x.ns.gin.ntt.net",
        "y.ns.gin.ntt.net",
        "zeit.fu-berlin.de"
    ],
    "US Navy": [
        "ntp2.usno.navy.mil",
        "tick.usno.navy.mil",
        "tock.usno.navy.mil"
    ],
    "Ubuntu": [
        "ntp.ubuntu.com"
    ]
}
```

Listing 3. List of Time Providers and their respective server names.

B SAMPLE GEODNS ZONE FILE

Listing 4 shows a GeoDNS sample DNS zone file. The GeoDNS zone file has multiple DNS subzones, like Turkey's tr (tr.pool.ntp.org). Each subzone has a list of IPv4 and IPv6 addresses (A and AAAA records), which lists NTP servers available to that particular country – and clients from these countries will see the A/AAAA records showed in this subzones[4]. Each A/AAAA records is followed by a *weight*, which is a non-standard DNS feature used by GeoDNS to sort the frequency in which records should be returned to clients, a method that allow NTP Pool volunteers to set indirectly the amount of traffic they want to receive at their NTP servers.

[4]Traditional authoritative DNS server use standardized text zone file formats [50], but GeoDNS uses JSON zone files instead [6].

In Listing 4 example, the server 203.17.251.1 is likely to appear 100x more often in responses than 149.255.99.71 (calculated by the ratio between their weights).

```
{
   "ttl": 390,                % DNS TTL
   "serial": 1345449135,      % DNS Zone file serial number
   "data": {
      "": {                   % Empty string indicates the "global" pool
         "ns": [              % Authoritative DNS servers
            "a.ntpns.org",
            "b.ntpns.org",
            "x.example.com"
         ],
         "a": [                % IPv4 addresses of all NTP servers in the global zone
            [
               "203.17.251.1", % IPv4
               "1000"          % Weight
            ],
            % Additional IPv4 entries...
         ]
      },
      "tr": {                 % Subzone: tr.pool.ntp.org
         "a": [               % IPv4 addresses for the subzone
            [
               "77.243.184.65",
               "1000000"
            ],
            [
               "212.175.18.126",
               "100000"
            ],
            % Additional IPv4 entries...
         ]
      }
   }
}
```

Listing 4. GeoDNS demo zone file for pool.ntp.org

C WEIGHTS VALIDATION

Next, we evaluate how each NTP server weight determines the distribution of NTP servers among clients when weight is considered. To do that, we carry out two experiments: a baseline experiment measured in the wild, which we compared against a controlled emulation in our setup.

In the baseline experiment, we query the authoritative server of one of its country subzones – Argentina's `ar`. We choose it because it has only eight active IPv4 NTP servers (on 2021-08-02 [62]), reducing the number of necessary queries to evaluate the weight's influence. By directly querying Argentina's 3.ar.pool.ntp.org, we *bypass* `GeoDNS`'s geolocation steps, obtaining records only listed in the `ar` subzone. (We confirm this behavior experimentally by running a test locally).

As shown in Table 7 (experiment ArgV4), we send 107k queries from 9.2k Atlas VPs. Each valid response received only *two* A records, and in total, we see eight distinct A records associated with NTP servers under Argentina's zone, as also reported in [62].

Table 8 shows the results. We see that each server receives from 7.2% to 18.3% of all queries – so, in the case of Argentina's subzone, the popular NTP service may appear at least twice as often than the less popular server. We use these results as a *baseline*.

For the emulation experiment, we create a test zone using the A records from Table 8 and, as weights, we use the counts value. We configure `GeoDNS` with this zone file on an AWS EC2 Frankfurt Ubuntu VM and use ~ 9k Atlas probes to query this zone, as shown in Table 3 (dataset ArgV4-Enum), use the same parameters (frequency, duration) in the ArgV4 experiment.

Measurement	ArgV4	ArgV4-Emul
Target	185.20.22.23	54.93.163.251
QNAME	`3.ar.pool.ntp.org`	`wilson.ants`
QType	A	A
Date	2021-08-02	2021-08-06
Interval	10min	10min
Duration	2h	2h
VPs	9219	9229
valid resp.	9068	9052
no resp.	783	382
ASes	3127	3128
valid resp.	3080	3067
no resp.	474	262
Countries	1	1
Responses	107031	110292
Valid Responses	104331	107793
invalid/empty	2700	2499
NTP servers	8	8
per response (median)	2	2
per response (q1)	2	2
per response (q3)	2	2
Queries/VP	11.6	11.9

Table 7. NTP Pool RIPE Atlas experiments with weights validation. Datasets: [75].

		ArgV4		**ArgV4-Emul**	
IP	**ASN**	**Counts**	**Ratio**	**Counts**	**Ratio**
162.159.200.1	13335-Cloudflare	37580	18.3%	37504	17.7%
168.96.251.227	3597-InnovaRed	31142	15.2%	31763	15.1%
170.210.222.10	4270-Red de Inter.	28599	13.9%	29288	13.9%
168.96.251.226	3597-InnovaRed	25707	12.5%	26737	12.7%
181.93.10.58	7303-TelecomArg	24878	12.1%	25836	12.2%
168.96.251.195	3597-InnovaRed	24731	12.1%	25812	12.2%
168.96.251.197	3597-InnovaRed	17223	8.4%	18288	8.7%
162.159.200.123	13335-Cloudflare	14838	7.2%	15832	7.5%

Table 8. NTP Servers occurrence for ArgV4 and ArvgV4-Emul experiments. Datasets: [75].

Similarly to EnumV4-Emul experiment, we reproduce the ArvgV4 experiment as ArgV4-Emul. We generate 110k responses from 3128 ASes, as shown in Table 8. We then compute the occurrence of each IP address from our demo zone in the Atlas responses and find that the query distribution per IP is *very similar* to the *original* experiment using the production servers of the NTP Pool. We can conclude that frequency counts can be used to infer weights in the NTP Pool zones.

D VENDOR ZONES AND IPV6 CLIENTS

The NTP Pool operators encourage vendors to ask for their own DNS subzones [66]. However, we did not find any vendor zones while reverse engineering in NTP Pool zone files (they are not publicly disclosed, and server's report pages do not list them). Are the vendors zones kept apart from the geographical zones? If so, how `GeoDNS` handles them?

NTP server	ArgV4-Android	ArgV6
162.159.200.1	748	182
162.159.200.123	748	182
168.96.251.195	747	181
168.96.251.227	379	52
168.96.251.197	195	61
168.96.251.226	121	63
170.210.222.10	54	7

Table 9. Query distribution for ArvgV4-Android and ArgV6 experiments. Datasets: [75].

We found out experimentally that they are a *replica* of the geographical zones. Their job is to allow the NTP Pool operators with a easy way to *remove* problematic vendors from service without affecting other users.

To determine that, we carry out experiments with RIPE Atlas, asking 32 probes located in Argentina to query for the A record of the Android vendor zone (2.android.pool.ntp.org). We analyzed the A records returned to these responses (dataset ArgV4-Android in [75]), and found only 7 distinct IP addresses,as shown in Table 9, all of them belonging *also* to the `ar` geographical zone. On the same day (2021-08-23), there were only 7 servers active in the `ar` zone [62] .

Therefore, we can conclude that the vendor zones seem to be a *replica* of the geographical zones – only that they give the ability to the NTP Pool operators to remove them in case of a vendor specific errors that can lead to DDoS attacks (CNAME records in DNS can be used to link both zones). As such, clients using vendor subzones are still bound by the geographical zones.

D.1 IPv6 clients

Clients can send queries over IPv4 and IPv6 to the NTP Pool authoritative servers, and they can be used to retrieve both A or AAAA records. To determine if IPv6 clients have a different view from the NTP Pool, we configure 12 RIPE Atlas probes to send queries over IPv6 from Argentina to the NTP Pool authoritative servers. Our goal is to determine if they would be also mapped to the Argentina's `ar` subzone, or if they would use other criteria.

Table 9 shows the results (Argv6 column and dataset).We see that IPv6 clients geolocated in Argentina are also mapped to the ar subzone when asking for A records 2.pool.ntp.org, are also mapped to the `ar` subzone. We confirm that by manually checking the IP address against Maxmind's geolocation database. Therefore, we can conclude that `GeoDNS` uses the same mapping process for IPv4 and IPv6 clients.