A Day in the Life of NTP: Analysis of NTPPool Traffic

Rushvanth Bhaskar
University of Twente & SIDN Labs
R.Bhaskar@student.utwente.nl

***Index Terms*—NTP, NTPPool, DNS, GeoDNS, Anycast, Traffic characterization, NTP Server Analysis**

***Abstract*—Accurate timekeeping is crucial for the functioning of applications and protocols in distributed networks - especially the Internet. The default protocol used for synchronizing time among servers and peers in the Internet is the Network Time Protocol. NTP is usually unauthenticated and is therefore prone to attacks though there have been multiple extensions and additions to the protocol to make it more secure. There are multiple public time providers that provide NTP servers that clients can use to synchronize time. NTPPool is one such volunteer run project that uses DNS to map clients to NTP servers that are closest to them. This is done by using an open source software named GeoDNS in the authoritative DNS servers of NTPPool. SIDN Labs contributes multiple NTP servers to the NTPPool project. One of these servers is deployed in 30 sites through Anycast and serves millions of clients. There has been little research into the characteristics of traffic that is received at a public NTP server. This research aims at analyzing the traffic received at the anycast NTP server that SIDN contributes to NTPPool in order to analyze the characteristics of the traffic that it receives. This includes information such as type of clients that use the NTP service, the catchment of the anycast sites, presence of anomalies in the NTP traffic, etc. This research will provide valuable insight into the the current state of the NTP ecosystem.**

I. Introduction

The Network Time Protocol (NTP) is used to "synchronize timekeeping among a set of distributed time servers and clients" [1, 2] and is the default time synchronization protocol that is used in the Internet. It works on top of the Internet Protocol (IP) [3] and User Datagram Protocol (UDP) [4] and is essential to the distributed working of machines in networks. Accurate timekeeping is crucial for applications like financial and legal transactions, and core protocols like TLS [5], DNSSEC signatures [6], DNS caches [7], RPKI [8], Kerberos [9], etc. Even cryptocurrencies like Bitcoin fundamentally rely on accurate timekeeping for its functionality [10].

The NTP architecture is organized as a tree where each level is a stratum (numbered from 1) consisting of servers. The accuracy of a server is related to how high it is in the tree with a server at a higher level (thus having a low stratum value) having more accuracy than a server at a lower level due to network paths and local clock stabilities. Servers that are present in Stratum 1 (primary servers) synchronize their time directly from an authoritative source like atomic clocks or GPS, Stratum 2 servers (secondary servers) synchronize time with Stratum 1 servers and so on [11].

The other side of the NTP ecosystem consists of clients that make use of NTP servers to synchronize their clocks. Clients that use NTP are extremely varied and range from servers and computers to IoT devices and mobile phones. In a system that is similar to DNS, through the DHCP protocol [12], clients can be configured to use NTP servers dynamically using DHCP options [13]. If NTP servers are not configured for clients through DHCP, then they fall back to the servers hard-coded in their Operating Systems. The synchronization of a client to a NTP server happens over several packet exchanges. This is done in order to accommodate for transmission delays, server delays, network congestion, etc that are present in especially large networks like the Internet. Only after these several packet exchanges, each satisfying the protocol requirements, does the client "believe" the time that it receives and is considered synchronized.

There are multiple public time providers that clients can use. Major vendors like Google [14], Microsoft [15], Apple [16], and Cloudflare [17] provide their own time services. `NTPPool` is one such time provider [18] that provides time for an estimated 5-15 million clients [19]. `NTPPool` lists more than 4500 servers around the world and is the default time provider for many vendors including Android, Linux, Asus, etc [20]. Contrary to the other public time providers mentioned before, `NTPPool` does not host its own servers, but rather lists servers run by volunteers and aims to simplify access to these servers through DNS. The maintainers of `NTPPool` mention that users will be mapped to NTP servers that are in or close to a users country [21]. This is done by using `GeoDNS` to map users to NTP servers.

SIDN Labs contributes to the `NTPPool` project by providing servers that are extremely popular in the pool and serve millions of clients worldwide [22]. One of these servers, `any.time.nl`, serves both IPv4 and IPv6 clients and is deployed in 30 sites through anycast [23].

It is evident that `NTPPool` is one of the core services on the internet but research into characterizing the traffic that a server which is listed on `NTPPool` receives is lacking. Previous research in this area includes work done by Rytilahti et.al [24] and Sherman et.al [25] where authors analyzed traffic from public NTP servers. These studies were performed quite a time back and critically, both of the studies do not focus on Anycast NTP servers and the role that anycast plays in NTP traffic.

Thus, the goal of this research project is to characterize NTP

traffic received at one anycast server that is listed in NTPPool. This is done by collecting data for 24 hours from the NTP server from all 30 locations. This data is then centralized, pre-processed to extract relevant fields, enriched with geo-location information, and analysed to give valuable insights into the NTP ecosystem. This characterization would give valuable insights into the kind of clients that use the NTPPool service, their geographical locations, the catchment of anycast sites, etc. There is also particular interest in the kind of *anomalies* that are observable in the traffic. Analysis of the traffic reveals extremely interesting patterns in client traffic while also exposing much of the RFC non compliant behavior that is particularly found in IPv4 NTP clients. Overall, this research gives a overlook of how clients typically use NTPPool and shines a light on the NTP ecosystem of the modern internet.

The primary research question for this study is thus defined as:

> *What are the traffic characteristics observed at an anycast NTP server listed in NTPPool?*

Additionally, the following sub research questions are also proposed:

1) Which anycast sites receive the most traffic (or) What is the Utilization of servers at each anycast site?
2) What is the catchment of each anycast site?
3) What are the characteristics of the client population?
4) Are there any traces of malicious NTP traffic sent by clients? If yes, what kind of attacks are performed?
5) What kind of traffic anomalies are observed?

This thesis is divided as follows - §II provides an introduction to the Network Time Protocol and explains how packets are exchanged between servers and clients along with possible methods to attack NTP. §III describes how NTPPool works and provides an in-depth look at GeoDNS and its configuration. §V describes the collected dataset that was used in analysis for this research and provides some background information on the data. §VI and §VII present the main results that were obtained in this thesis characterized by the server and client characteristics.

II. NTP - A Primer

A. The NTP Ecosystem

The Network Time Protocol (NTP) is designed to distribute information about time in large networks. As mentioned in §I, NTP (like DNS) is a core protocol of the internet and a wide variety of applications, protocols, and services depend on accurate time information for their working. NTPv4 [2] is the currently used NTP version on the internet and replaces NTPv3. The current version improves upon NTPv3 by addressing the bugs present in the previous version of the protocol and allows for the use of extended timestamps which results in time resolution at the scale of one nanosecond.

The NTP ecosystem consists of *primary* NTP servers, *secondary* NTP servers, and NTP clients. Primary NTP servers get time information by synchronizing directly with reference clocks such as GPS data, atomic clocks, etc. Secondary NTP servers in turn synchronize with primary servers to provide time information for other clients. NTP clients synchronize with primary or secondary NTP servers to receive time information but do not provide synchronization information to other NTP clients. This architecture transforms to a network tree model which is represented in Figure 1 with each level having a *Stratum* value.

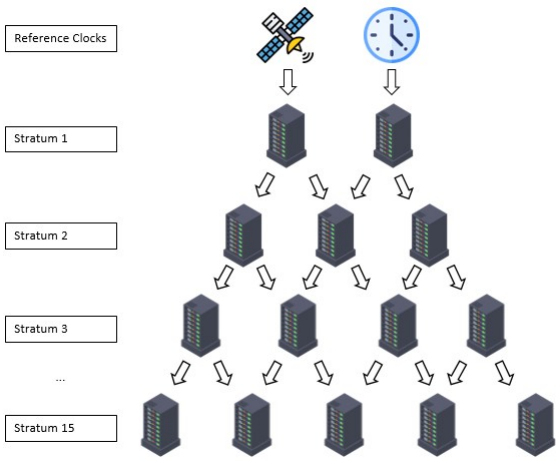

Fig. 1: NTP Architecure

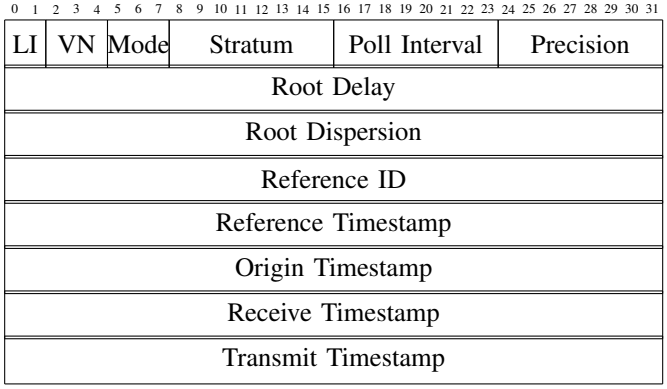

Fig. 2: NTP Packet Header Format

Time synchronization in a typical client-server implementation of NTP happens with the repeated exchange of messages between two systems periodically. The client sends an NTP request to a server following the packet format illustrated in Figure 2 after setting the "Origin Timestamp" field. The server responds back to the client with the same packet format by setting all fields with relevant information except the ones which are set by the client. The "Version"(VN) and "Mode" fields are set by both the server and client. The Mode field is used to indicate the mode of operation of the machine in the NTP ecosystem. A value of 3 indicates that the source of the packet is a client and a value of 4 indicates that the source of the packet is an NTP server. In total, NTP defines 8 modes of operation in its RFC [2]. The Leap Indicator (LI) field is set to warn the client of an impending leap second that is to

be added or subtracted from the last minute of the month. The Version field indicates the NTP version number that is used and when using NTPv4, this field is set to the number 4. The Stratum field is used to indicate the Stratum in which the machine is present in the NTP architecture. As indicated in Figure 1, values can range from 1-15 with 0 and 16 indicating that the server is not in sync, and values greater than 16 being reserved for future use. The poll interval represents the interval between successive messages and the precision indicates the precision of the system clock.

The Root Delay and Dispersion fields are set by the NTP server and they indicate the total round trip delay and dispersion to the physical reference clock. The Reference ID is an important field that is crucial to avoid timing loops. Timing loops occur when 2 servers or a set of servers use each other as upstream servers which could lead to both servers not having the correct time information. For Stratum 1 servers, this field contains an ASCII string that represents the physical clock that is used for time synchronization. For secondary servers and clients (stratum 2 and above), this field contains the IPv4 IP address (or) the MD5 hash of first 4 octets of the IPv6 IP address of the upstream server.

The main information in the NTP header is present in four fields that contain timestamp information. Three of these fields - origin, receive, and transmit timestamp are used to calculate time offsets and the reference timestamp field is the time when the system clock was last set. The origin timestamp is the time at the client when the NTP request was dispatched to the server. The receive timestamp indicates the time at which the server received the request from the client and consequently the transmit timestamp is time at the server when the response was sent back to the client. There is a separate field called "Destination Timestamp" that is not included in the header format which is the time at the client when it receives the response from the server.

In normal operation, consider a peer *A* which wishes to synchronize time with a server *B*. At time t_1, *A* transmits a packet to *B* after setting the origin timestamp. *B* receives the packet with the origin time t_1 and calculates the receive timestamp t_2. *B* then sends a response to *A* containing timestamp information t_1 and t_2 after setting transmit timestamp t_3. Finally, *A* receives the response and calculates the destination timestamp at time t_4. *A* then uses the 4 timestamps to calculate the offset (θ) and delay(Δ) of *B* relative to *A* which is then used to set the system clock.

$$\theta = T(B) - T(A) = 1/2 * [(t_2 - t_1) + (t_3 - t_4)]$$

$$\Delta = T(ABA) = (t_4 - t_1) - (t_3 - t_2)$$

Depending on the poll interval, these messages are exchanged periodically which allows the client to calibrate its clock and account for network delays, congestion, etc. Other fields present in the NTP header include Extension fields, Key identifier, and digest which are not discussed here as they are not critical to the focus of this paper.

B. Security of the Network Time Protocol

DDoS attacks using NTP amplification have also become a major threat in the past few years [26]. The effects of DDoS can be increased using *reflectors* and *amplifications*. In reflection and amplification attacks, attackers send a spoofed packet with the source IP of the victim to a server. The server in turn sends the amplified response to the actual client thereby inundating the victim with unwanted traffic [27]. An amplifier in this case would be a public server that is running a suitable protocol where the response packet sizes are much larger than the request packet size. UDP based protocols such as DNS and NTP allow for huge amplifications which is used by attackers to exhaust the bandwidth of victims. NTP was identified to have one of the most potent DDoS amplification vectors [28].

NTPv4 does allow for both symmetric and asymmetric cryptographic authentication so that clients can make sure that they are communicating with an authentic NTP server but these options are rarely used [29]. The usage of symmetric encryption is cumbersome as this requires the exchange of keys before time synchronization and this would be hard to do for public NTP servers which need to receive queries from random clients. Symmetric encryption also works by appending an MD5 hash of the packet contents to the NTP packet. The usage of MD5 has been deprecated [30] as it has been proven to be insecure. Asymmetric authentication in NTP is based on the Autokey protocol [31] but this protocol is not widely supported in NTP servers and is also insecure which resulted in minimal usage. Therefore, it is safe to assume that usage of NTPv4 has widely been unauthenticated.

To improve the security of the Network Time Protocol, a security extension called Network Time Security (NTS) [32] was added to NTP that allowed the use of Transport Layer Security and authenticated encryption for the client-server mode of NTP. The NTS Key Establishment (NTS-KE) protocol is used to exchange keys that allows the client to authenticate the NTP server. When using NTS, clients initially connect to a NTS-KE server and perform a TLS handshake. After the TLS channel is established, the server sends the client an IP address of an NTP server along with some cookies for that particular NTP server. The server and client also use TLS key export [33] to obtain key material. After this exchange, the TLS connection is closed and the client now initiates a connection with the NTP server and includes the previously received cookies and an authentication tag that is derived using the key in its request. The NTP server then uses the cookies that it receives to derive the same key material and then sends an authenticated response. In a report published by SIDN [34], NTS is described as a "relatively complex protocol" and also gives rise to a bootstrapping problem. When a client wants to synchronize its clock for the first time, it has to assume that the first TLS connection initiated to an NTS server is legitimate. However NTS provides feasible cryptographic authentication to the most widely used modes (client and server) of NTP thus helping secure the protocol.

C. Simple Network Time Protocol

The Simple Network Time Protocol(SNTP) [35, 2] is a subset of Network Time Protocol. SNTP clients can synchronize with any NTP server as the protocols are completely interoperable and the packet information is exactly the same. The main difference is the time setting itself: NTP uses algorithms that are intended to maintain a highly accurate time. For this reason, multiple time servers are consulted and checked for accuracy. NTP adjusts the system clock using small, skewed adjustments in order to ensure seamless time correction; in order to make time changes, the clock is sped up or slowed down slightly. In contrast, SNTP usually uses a simpler approach: e. g., some implementations utilize time jumps to adjust their clocks. SNTP uses only a single upstream server in contrast to NTP which can synchronize time information from multiple servers. SNTP is primarily used by clients that may not have high processing power like IP cameras, DVRs, and other IoT devices. SNTP is also useful for applications that might require one-off time information and do not need to be constantly updated with the latest time. In essence, SNTP offers time synchronization that is at a lower quality than NTP but is ideal for scenarios where requirements of the highest accuracy and security are not a priority.

III. How does the Pool Work?

`NTPPool` is a community project which helps clients gain access to NTP servers that are run by volunteers through the use of DNS. The `NTPPool` project is one of the biggest NTP providers in the internet serving hundreds of millions of clients [18]. This number is likely lower than the actual number of clients using `NTPPool` due to a combination of clients using resolvers that cache DNS responses and the widespread use of Network Address Translation to map multiple IPv4 addresses to a single address. `NTPPool` was initially introduced to combat the misuse of a static list of public NTP servers [36]. Instead of relying on a small set of public servers, `NTPPool` maps clients to NTP servers using DNS to provide users with NTP servers that are geographically closer to them. `NTPPool` also performs DNS load balancing in order to distribute queries among servers and not overload one or some of them. `NTPPool` uses the domain `pool.ntp.org` and its subdomains to map clients to NTP servers. The DNS zone for the main domain and subdomains contain IP addresses of NTP servers that are returned to clients.

For all the servers that are listed in `NTPPool`, the pool system also performs health checks on the servers to verify whether the server is reachable and provides accurate time information. `NTPPool` has one monitoring station in San Jose, California, United States from where the monitoring checks are performed. Based on the monitoring, a maximum score of 20 is assigned to the server and IP addresses of the servers are only returned in DNS responses if they have a score above 10. The score of a server is reduced if the NTP server is not reachable or if the time information that it provides is inaccurate by more than 100ms.

A client that wishes to use `NTPPool` has to configure the NTP servers {`0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org, 3.pool.ntp.org`} in its operating system [21]. Once this is done, when an NTP update is required, the client will first perform a DNS resolution on the above mentioned domain names to get the IP of that particular NTP server. Upon receiving this DNS query, the authoritative servers of `NTPPool` will return a list of 4 IP addresses to the client. The NTP client then reaches out any of the received IPs to get time information as described in §II.

`NTPPool` is unique in the way that its authoritative DNS servers return a list of IPs to a DNS query. `NTPPool` uses `GeoDNS` [37] at its authoritative DNS servers, which is an open source authoritative DNS server that was developed by the maintainers of `NTPPool`. `GeoDNS` supports Geolocation based routing that allows `NTPPool` to map users to the geographically closest NTP server [38]. For example, clients using DNS resolvers in Netherlands will receive a list of NTP servers that are located in Netherlands. `GeoDNS` translates IPs to geolocation using the Maxmind IP2Location Database [39]. DNS servers use zone files [40] to store DNS records that have domain to IP data. These zone files are used by DNS authoritative servers to respond to DNS queries. `GeoDNS` does not use the standard DNS zone file, but instead uses a JSON configuration file that contains information about the global zone and sub zones. Significant research has gone into reverse engineering and validating the zone file that `NTPPool` uses by Moura et.al [41]. This is because the authors found that there was little to no documentation available for `GeoDNS` and `NTPPool` does not publish its zone files. Therefore, the method in which `NTPPool` maps clients to NTP servers and the way in which clients use `NTPPool` was relatively unknown. In [41], the authors found that `NTPPool` mainly uses two types of zones - geographical (country, continent) and vendor specific zones. The authors also found that the distribution of NTP servers around the world was not uniform and some countries (like France) can have hundreds of available NTP servers while other countries have very few to none.

Therefore, the main zone `pool.ntp.org` has subzones for countries (`fr.pool.ntp.org` for France), continents (`europe.pool.ntp.org` for Europe), and vendors (`debian.pool.ntp.org` for Debian). `NTPPool` uses separate zone files for vendors as it allows for easy separation of traffic and unintentional traffic can be identified and migrated easily [19].

The administrators of `NTPPool` mention that clients should get the closest servers if they configure the main domains ({`0-3`}`.pool.ntp.org`) but clients can also use any of the geographical sub domains. When a client makes a request to DNS authoritative servers running `GeoDNS`, it first fetches the geolocation (country, continent) of the client from the Maxmind location database. It then attempts to match the country of the client to a sub zone that is specified for that country. If there is a match, then 4 IPs are selected from that sub-zone depending on the weights assigned to the IPs and

returned to the client. If there are no available NTP servers in the sub zone defined for that country, GeoDNS will attempt to match the client's continent to a sub zone. In cases where there is no location data for a client IP or the country/continent have no sub zones defined, GeoDNS will match the client to global '@' zone. In this way, GeoDNS attempts to iteratively match the client to closest geographical location that has valid IP addresses of NTP servers available.

GeoDNS supports load balancing at the DNS level through the specification of weights for IP addresses associated with a domain. Since NTPPool is a volunteer run project where users contribute server bandwidth, all users might not be able to contribute the same amount of resources. Therefore, users can specify the bandwidth that they contribute and GeoDNS transforms this into a weight for that particular IP. The weight associated with an IP address affects how often that particular IP address is included in DNS responses. For example, an IP address with a weight of 1000 will be included in DNS responses twice as much as an IP address with a weight of 500. This in turn becomes a load balancing solution to how much traffic that a particular IP address receives. A sample GeoDNS configuration file is displayed in Listing 1.

One of the volunteers contributing to NTPPool is SIDN Labs [42] - the research wing of SIDN which is the registry for the *.nl* Top Level Domain. A report published by SIDN Labs [43], describes the physical setup of the stratum 1 NTP servers and the reference clocks that are used. The motivation for running the NTP servers was the lack of transparency in the NTP ecosystem with regard to the primary reference clocks that servers use and details about the privacy of clients. The Stratum 1 NTP servers that are managed by SIDN Labs rely on two physical reference clocks. The primary clock receives time information from the Global Navigation Satellite System (GNSS) with DCF77, which is a German long wave time signal and standard frequency radio station, acting as backup. The GNSS data relies on the European Galileo system [44] instead of the American Global Positioning System (GPS). The secondary core clock is a Rubidium atomic clock that acts as both a reference clock for the primary core clock and as a backup should the GNSS based clock go down.

From these physical reference clocks, front-end clocks receive time information through the Precision Time Protocol (PTP) [45]. These front-end clocks are Stratum 1 NTP servers which in turn provide time information to Stratum 2 servers that are listed in NTPPool. NTPPool lists all the servers that are managed by SIDN Labs on their website [22] and this research particularly focuses on the NTP server any.time.nl which is deployed in 30 sites through anycast. These 30 nodes are spread across 21 distinct physical locations around the globe and handle an estimated 100,000 - 175,000 queries per second. A list of sites that the NTP server is deployed in is published by SIDN Labs [46] and also available in Appendix A. NTPPool also lists the DNS zones that this NTP server is included in which implies that clients in all the countries that is an anycast site of the NTP server any.time.nl make use of it to synchronize time information depending on how often NTPPool returns this server in its DNS responses. The any.time.nl NTP server serves both IPv4 [47] and IPv6 [48] clients and is extremely popular in the pool. any.time.nl is configured with a weight of 500 for IPv4 and 1000 for IPv6. The server consistently receives the maximum score of 20 from the monitoring service of NTPPool.

```
{
  "serial": 3,
  "ttl": 600, # TTL for DNS records in this zone
  "data": {
    "": {
      "ns": {
        "ns1.pool.ntp.org",
        "ns2.pool.ntp.org"
      },
      "a": [ # IP addresses of all NTP servers in
    main global (@) zone
        [
          "192.168.1.1", # IPv4 address
          "1000" # weight
        ]
        ...
      ]
    },
    "europe": { # Geographical sub zone for
    continent Europe
      "a": [
        [
          "10.0.1.4",
          "500"
        ]
      ]
    },
    "fr": { # Geographical sub zone for France
      "a": [
        [
          "192.168.1.2",
          10
        ],
        [
          "192.168.1.3",
          10
        ]
      ]
    },
    "android": { # Vendor specific sub zone for
    Android
  "a": [
   [
     "10.1.1.1",
     "100"
   ]
      ]
    }
  }
}
```

Listing 1: Sample GeoDNS Configuration File

IV. Related Work

A. NTP Traffic Characterization

Previous research in this area includes an extensive longitudinal study conducted by Rytilahti et.al [24] on NTPPool and the NTP ecosystem as a whole. The authors deploy NTP servers to various geographical locations to categorize

incoming traffic and found that they were able to reliably categorize incoming NTP traffic as NTP or SNTP. An interesting observation is made regarding the setup of NTPPool whereby it would be possible to "poison" NTPPool by deploying a malicious NTP server in countries where are there no listed NTP servers since GeoDNS always attempts to match the client to the most specific location. This would mean that this malicious server would be the only server that is returned to queries from that particular country. Notably, there is only a minimal characterization of the NTP traffic received above characterizing between NTP and SNTP. They conclude by pointing out that the NTP infrastructure is becoming centralized with a few popular service providers and volunteer efforts like NTPPool are crucial to ensure that the Internet stays distributed.

Sherman et.al [25] also conducted a study on NTP Traffic characterization on the NIST time servers. NIST operates one of the largest collection of NTP servers which are linked to atomic clocks and provide reference to the GPS, Satellite Navigation Systems, etc. Traffic to these servers has grown at an exponential rate with billions of queries being received each day. The authors in their research look at the traffic received at NIST servers and attempt to analyze the characteristics of this traffic. They mainly look at the request rate, time when the requests were received, etc. Interesting temporal variations are observed with respect to receiving NTP traffic where NIST servers seem to get bursts of NTP queries every one and a half hours. The authors also observe that roughly 316 million unique IP addresses requested time information from their NTP servers in a 2 month time period. The majority of the NTP requests used the obsolete NTP version 3 instead of the current NTP version 4. They also saw that the use of SNTP was widespread which was in line with Rytilahti et.al's findings.

The current study differs from these related works on traffic characterization by focusing on an anycast NTP server. Both of the previous studies collected results from individual NTP servers that were sometimes located in different locations. Traditionally, anycast deployments have been growing in the Internet as it offers various advantages like using the same IP for different physical locations, fail over in case of a single site going down, etc. This study also gives a well rounded look at the NTP ecosystem as data is collected from NTP servers that are deployed all over the world.

B. Attacks against NTP

The usage of unauthenticated NTP can leave servers and clients open to attacks. Malhotra et.al [49] found that NTP can be attacked in various ways through *on-path* and *off-path* attacks. Attacking NTP could have dire consequences as many other applications and protocols depend on accurate time information. The authors of the paper describe an event in 2012 when the time on two Stratum 1 servers went back by 12 years causing widespread outages in Active Directories, routers, and PBXs. *Time Skimming* is a possible *on-path* attack against NTP. This attack works by altering the time on a clients machine.

NTP can also be attacked through *off-path* attacks such as Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks. One way of performing a DoS attack is by exploiting the Kiss-o'-Death (KoD) packet that is defined in the NTP protocol. The KoD packet is defined in the NTPv4 RFC [2] as a rate limiting feature for NTP servers. A server sends the KoD packet to a client that queries it too many times within a short time period. The client after receiving the KoD packet does not query the server for a time period that is at least equal to the time in the poll interval field of the NTP header. When using unauthenticated NTP, it is possible for attackers to send spoofed KoD packets to NTP clients that appear to come from their NTP server in order to restrict them from contacting their NTP servers for extended periods of time.

Another study [26] describes a rapid rise in DDoS attacks that use NTP for amplification surpassing even DNS based amplification attacks. They find that misconfigurations in public NTP servers allow for these servers to be used as reflectors and amplifiers. An example of amplification in NTP is the use of the *monlist* query. A client sends a mode 7 message to the server, setting the request code to the *monlist* query. On receiving this query, a server returns a list of the previous 600 clients that contacted the server. The resulting size of the response packet is many times larger when compared to the request packet. Ideally, this command, which is only intended for diagnostic purposes should be disabled for public clients. But wrongly configured servers on the internet still contribute to such DDoS attacks. The authors measured that in January 2014, there were approximately 1.4 million potential amplifiers in the internet and this figure gradually declined to 110K by April, 2014 as operators started patching their systems. Other mode 6 and 7 NTP queries like *version* and *showpeers* can also be used to achieve amplification. The current study measures whether wrongly configured NTP servers are still a threat vector for reflection and amplification based DDoS attacks by analysing the collected NTP traffic for traces of mode 6 and 7 queries. The presence and quantity of these queries will give an indication of whether NTP is still used for these attacks.

V. Dataset

The dataset that was used for this study consists of packet captures from all the 30 anycast NTP servers that are listed in NTPPool. Appendix A contains the physical locations of the 30 NTP servers that are deployed in anycast and listed in NTPPool under the IPv4 address 194.0.5.123 and IPv6 address 2001:678:8::123 [47, 48]. These servers receive traffic from clients that use NTPPool as their NTP service provider. The process of receiving a particular NTP server is explained in §III.

NTP Traffic arriving at these servers was captured using tcpdump [50] for a period of 24 hours between June 22 to June 23, 2022. Relevant IP, UDP, and NTP data was extracted

Total queries	7,283,163,487
IPv4	7,086,571,522
IPv6	196,591,965
Responses	6,394,872,496
IPv4	6,199,968,273
IPv6	194,904,223
Ratio Responses	0.87
IPv4	0.87
IPv6	0.99
Total clients	158,711,167
IPv4	132,123,933
IPv6	26,587,234

TABLE I: NTP Client Queries

Continent	Clients	Queries	% Total Queries	% Total Clients
Asia	80,418,081	4,757,704,427	65.32	49
N. America	40,693,609	1,555,527,952	21.35	24.83
Europe	34,644,160	660,212,853	9.06	21.14
S. America	2,996,825	108,501,122	1.49	1.82
Oceania	3,539,260	132,772,676	1.82	2.15
Africa	1,566,678	68,444,457	0.93	0.95

TABLE II: Breakdown of Queries Received Per Continent

from the captured `pcap` files and then enriched with client geolocation data including City, Country, Autonomous System Numbers (ASN) [51], and physical coordinates of client IPs. This information was gathered using the latest available Maxmind [39] geolocation databases at the time of collection of data. Subsequently, the IP addresses in the collected files were anonymized using `cryptopANT` [52] which performs IP anonymization using the crypto-Pan algorithm, first introduced by Fan et.al [53]. This was in accordance with the privacy considerations of SIDN and was approved by the privacy board.

In total, about 4.7 TB of data was collected which contained 13.67 billion NTP queries and responses across the 30 servers. Of these, 7.28 billion were client queries from ≈158.7 million unique clients. The majority was IPv4 traffic (97.3%) and the rest was IPv6. A breakdown of the exact number of client queries and responses is specified in Table I. This dataset was further analyzed to study NTP server and client characteristics and to answer the Research Questions outlined in §**??**.

VI. NTP Server Characteristics

A. NTP Server Utilisation

In this section, the characteristics of the anycast NTP servers are analyzed in order to answer Research Questions 1 and 2. First, the utilisation of the NTP servers was analyzed to gather insights into which server/anycast site was particularly popular in the NTP ecosystem. This analysis gives an idea about the underlying client distribution in the internet. To determine the usage of NTP servers, the aggregated data was partitioned into queries received at each individual server. Determining the server with the most usage was done by ranking servers in order of the amount of client queries received. Figure 3 compares the number of queries received at each Anycast site.

From Figure 3, it is apparent that the anycast sites in Asia are particularly popular on `NTPPool`. The servers `bom1-1`, `bom1-2` combine to serve more than 1.59 Billion client queries from ≈28.4 million clients. This constitutes for about 22% of the total traffic received on `any.time.nl`. Other servers in Asia also receive considerably more than traffic than their counterparts in North America and Europe as evidenced by the number of queries received at `nrt1-{1-8}`, `icn1-1`, `sin1-{1-2}`. Table II gives a breakdown of the distribution of queries per continent of the anycast NTP servers. In total, the NTP servers in Asia account for ≈65% of the total utilisation across the 30 servers.

Among the 7 servers in North America, there is an outlier in terms of queries received and utilisation. The NTP server `sea1-1` stands out as one of the most utilised servers in North America with similar traffic received as `bom1-{1-2}`. Out of all the 7 North American sites, `sea1-1` constitutes 49.3% of the traffic and 10.5% of total traffic among all 30 sites. A deeper look at the geographic distribution of the clients that use this server reveals that the majority of traffic that this server receives is from China. In the observed period, `sea1-1` received more than 5 times the amount of traffic from China than from United States and Canada. The clients from US and Canada would be the geographically closest locations to this server. Therefore, it is evident that looking at distribution of queries received at each server alone is not enough to get a complete picture of the NTP ecosystem. It is also imperative to look at the distribution of clients and the routing behaviour of `NTPPool` to understand how clients are actually mapped to NTP servers and contrast the observed results with the intended behavior described in §III.

Possible reasons for the high utilisation in the Asian servers could be caused by the relatively lower number of volunteer NTP servers in the Asian zone of `NTPPool` and relatively higher number of NTP clients in the continent. In essence, there are lower number of servers handling a larger number of clients when compared to the other continents. Since `NTPPool` publishes data on the number of NTP servers included in each country and continent zone [54, 55], it is possible to derive how many servers were available in Asia versus other continents during data collection. Scraping the

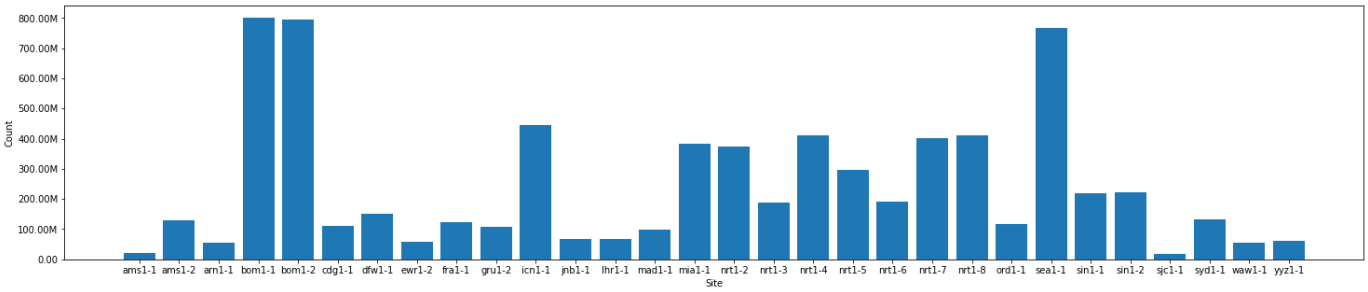

Fig. 3: Anycast Site Utilisation

data from the website shows that NTPPool had a total of 243 IPv4 servers and 102 IPv6 servers in the Asia zone while the Europe zone had 2088 IPv4 servers and 1018 IPv6 servers. There is a stark contrast in number of available servers where Europe alone has close to 10 times more IPv4 and IPv6 servers. With GeoDNS trying to route clients to the geographically closest available NTP server, it is apparent that there will be a greater load on the NTP servers that are in the Asian DNS zones. There is an even greater divide between the number of servers available in the European and North American continents and the African and Oceanic continents. During data collection, the entire African continent had a total of 38 IPv4 servers and 20 IPv6 servers. The majority of the servers (>50%) are concentrated in South Africa. Surprisingly, the utilisation on the anycast site located in South Africa is among one of the lowest. Queries from the African continent as a whole only account for less than 1% of the total queries that were collected across the 30 servers. Therefore, at an initial look, it looks like the number of NTP servers currently available in Africa are capable of handling the demand. There is also a need to look at the actual servers that clients in Africa reach out to as there could be cases where African clients are routed to NTP servers outside the continent.

As discussed in §III and by Moura et.al [41], when clients want to synchronize time, they reach out to any of the 4 servers {1-4}.pool.ntp.org and receive 4 IP addresses determined by GeoDNS. any.time.nl is included in 23 zones in NTPPool [47]. This includes all the country zones that have an anycast site, all the continent zones, and the global zone. Therefore, any.time.nl will receive traffic when it's IP is returned to clients under the following conditions: -

1) A client which belongs to the zone of any country that any.time.nl is a part of and thus GeoDNS selects this IP as one of the 4 IPs to return.
2) A client from a country who's country zone in NTPPool does not have any NTP servers available and thus GeoDNS matches the client to its continent zone.
3) A client with no valid geolocation information or is matched to the global ('@') zone in some other way.

The frequency of this IP being returned to different clients depends on the weight assigned to the NTP server and as mentioned in §III, any.time.nl has a weight of 500 for its IPv4 address and a weight of 1000 for its IPv6 address. One factor that affects IPv6 traffic is the fact that only the name server 2.pool.ntp.org supports IPv6 and returns AAAA records [56]. This could result in situations where IPv6 capable clients fall back to IPv4 when they contact a name server other than 2.pool.ntp.org.

There are of course other edge conditions in which clients can send NTP queries to any.time.nl such as directly configuring its IP in their NTP client/OS, etc. Since DNS responses are cached on intermediate resolvers, clients might always get the same 4 IP addresses when making a request. NTP client software/daemons could also cache the IP address of an NTP server and continue to use the same server to synchronize time. Hence, it is also important to characterize NTP client behavior which is done in §VII.

Among the 3 scenarios listed above for in regards of routing of NTPPool and GeoDNS, the first scenario is simple in terms of routing as GeoDNS matches the client to the zone of its country and returns 4 IP addresses from that zone. Scenario 2 is interesting as it implies that when GeoDNS geolocates a client, there were no NTP servers in the zone of the client's country and thus GeoDNS picked 4 IP addresses from the zone of the continent that the client belongs to. Taking a deeper look at the queries from clients in countries that did not have any NTP servers in their country zone gives us a better idea of this behavior. In the Asian zone, there are 12 country zones that did not have any NTP servers during data collection [54]. The NTP server any.time.nl received queries from 9 out of the 12 countries. These clients would have followed the scenario described above to get any.time.nl as 1 of the 4 NTP servers that GeoDNS selected from the Asian continent zone. In total, there were 8.03 million queries from Asian countries that had zero servers in NTPPool. The majority of these queries were from the countries of Jordan (31.4%) and Myanmar (42.8%). Overall, queries from countries that did not have an NTP server only constituted for 0.14% of the total queries sent by clients from countries in Asia. In Africa, there were a total of 41 country zones with no NTP servers. any.time.nl received queries from 39 of these countries which totalled to 48.2 million queries. In contrast to Asia, this traffic constituted to 39% of the total queries sent by clients from African countries. It is apparent that a large portion of the NTP queries in the African continent are from countries that do not have any NTP servers and thus get IPs from the African

Continent	Queries	% Total Traffic
Asia	8032973	0.147
Africa	48266484	39.30
Europe	258826	0.053
North America	3063388	0.511
Oceania	473167	0.35
South America	13472294	2.7

TABLE III: Total traffic from countries with 0 NTP servers per continent

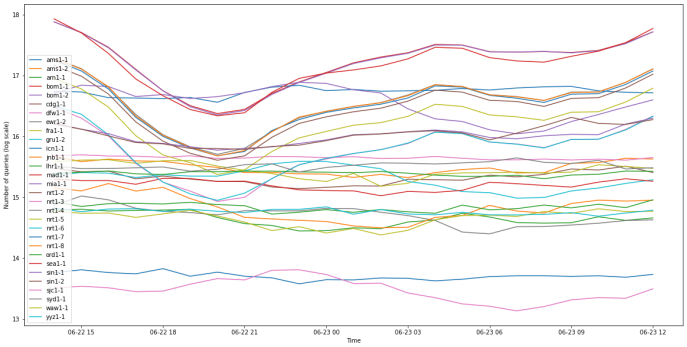

Fig. 4: Number of Queries per Hour

continent zone. In contrast to Africa and Asia, Europe has 9 countries with no NTP servers and `any.time.nl` received queries from 6 of these countries. But, in total, these queries only totalled to 258.8k which constituted for only 0.05% of the total traffic from clients in Europe. In North America, queries were received from 21 countries which had 0 NTP servers and this constituted for 0.5% of total queries. Thus, even though the number of NTP servers between Europe and North America are comparable, the distribution of these NTP servers are better in Europe as more countries have an NTP server. South America is also a standout because countries like Venezuela and Bolivia that send a lot of queries (1.08 and 2.03 million respectively) do not have any NTP servers in their country zones. Therefore, around 2.7% of total traffic come from all 6 countries that do not have an NTP server. A detailed breakdown of queries from countries that do not have an NTP server in their corresponding zone in `NTPPool` is shown in Table III along with the percentage of total traffic received from the clients in these continents. Thus, from the previous results on utilisation of anycast sites and a look into the routing behavior of `GeoDNS` shows that there is a need for more NTP servers in the Asian continent in particular. Europe and North America seem to have a robust NTP infrastructure in `NTPPool` with enough servers to handle client traffic. It is concerning that many countries in Africa do not have NTP servers available and a major portion of NTP traffic seems to be answered from servers outside of a client's country.

Server utilisation was also analysed by looking at the number of queries each anycast server receives per hour for the duration of the packet capture. Figure 4 shows the number of queries per hour at each NTP server. The Y-axis is on a logarithmic scale for better representation of the difference in queries between the various servers.

At the outset, there are some patterns visible especially with respect to the Asian NTP servers. A clear dip in the number of queries is observed during the hours of night. This implies that the number of clients sending NTP queries reduces in the night when compared to the daytime hours. On the other hand, traffic at the European and American sites stay mostly constant throughout the day. This observation is in line with the research of Quan et.al [57] where authors show that diurnal patterns in internet traffic is prevalent in Asia while it is mostly always-on in US and Western Europe. This study with respect to NTP traffic on the internet conforms to this observation by Quan et.al in terms of traffic patterns across various continents.

Analysing the traffic received in the night versus the traffic received the day, the Asian NTP servers receive 30% less traffic in the night. Night time in this case was taken from the hours of 14:00 UTC to 01:00 UTC. This traffic pattern also implies that Asian sites could have more people-operated devices as compared to IoT devices or devices that send traffic in an automated manner.

B. Anycast Catchments

Another important aspect of Anycast deployments is the catchment of anycast sites. The Operation of Anycast Services RFC4786 [58] defines anycast catchments as "the topological region of a network within which packets directed at an Anycast Address are routed to one particular node". Therefore, studying the catchments of a deployment gives an idea of the underlying routing behavior in the Internet and is helpful to fine tune the anycast deployment in order to serve clients in an efficient manner. Figure 5 shows the catchment of each NTP server (anycast site) as a heat map. Further, Appendix B contains separate heatmaps for all the sites for IPv4 and IPv6 traffic. The heat maps were plotted by getting all the latitude, longitude positions of clients for each site. The heat maps shown in Figure 5 was plotted with the data collected but the current heatmaps for each site is provided by SIDN [46] and can be referred to for an up-to-date version. It is important to note that `NTPPool` itself does not have influence on the anycast site that a client uses. `GeoDNS` and `NTPPool` only make decisions in the IPs that are returned to a client when it is synchronizing time. The actual site that a client reaches is determined by the underlying routing protocol on the internet - Border Gateway Protocol(BGP) [59]. Therefore, in a sense, anycast catchments help to visualize BGP routing behavior.

As described above, the catchments of each site simply depict the various locations from which NTP servers receive traffic. The darker an area, the greater number of clients from that location. From the catchment visualizations, it is immediately apparent that some sites have very good catchments while other sites do not. The quality of a catchment here is the specificity of locations from where a server receives traffic. For example, the site at `dfw1-1` which is located in Texas, United

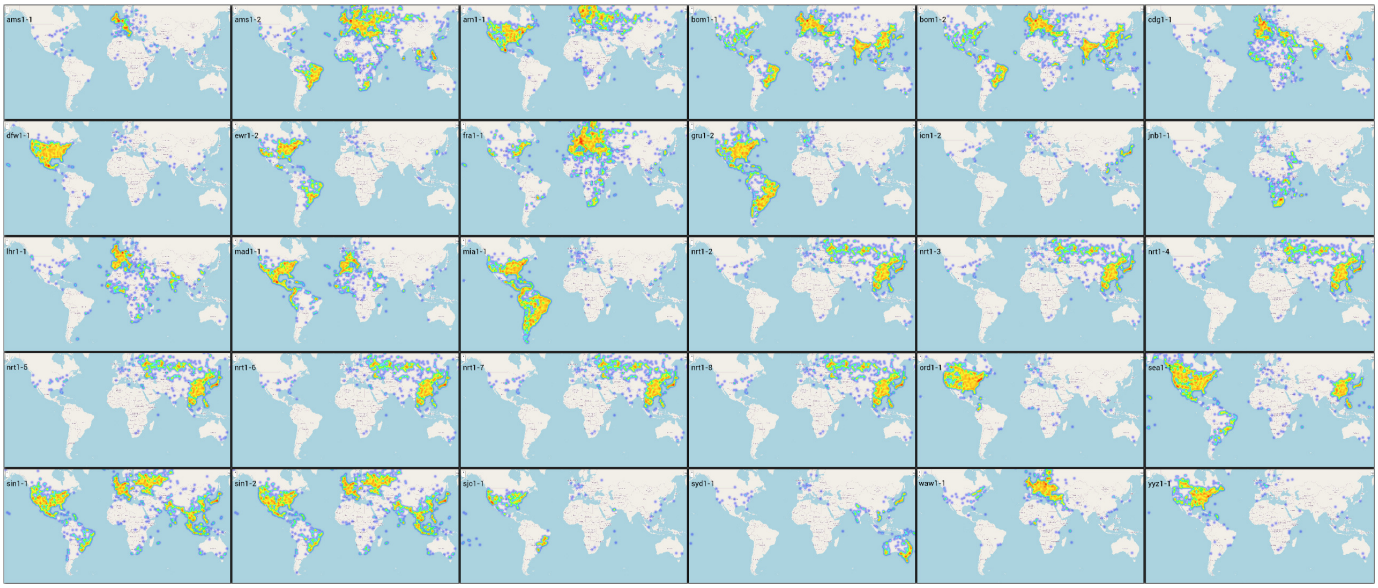

Fig. 5: Catchment Maps of Anycast Sites

States (refer Appendix A) has a good catchment because the majority of the traffic that reaches this site is localised to the United States. This in turn implies that the BGP routing for clients is done in an efficient way as to ensure that the client reaches a site that is close to it. Some other sites that have good anycast catchments are `icn1-1`, `jnb1-1`, `ord1-1`, `syd1-1`, etc. In particular, the site at `icn1-1` is able to capture the majority of its queries only from its country. This is also true for the site in Sydney (`syd1-1`) which receives most of its queries from Australia.

On the other hand, there are multiple sites that have a very broad catchment. A broad catchment refers to catchments that receive traffic from a large topological area. In an anycast deployment, this is not desired as the servers are geographically separated in order to handle client traffic from that particular regions. A client which is routed to a server that is farther away experiences more delay and latency which is counter intuitive to the goals of Anycast. `bom1-1, bom1-2` for example receive traffic from all over the world even though clients could potentially reach servers that are closer to them. This is also partially the case for locations like `ams1-2` and `sea1-1`. Earlier, the `sea1-1` stood out in terms of utilisation and the majority of the clients that were reaching this server seemed to come from China. That observation is backed up by the catchment data where this particular site has a large footprint of clients from China. It is also observed that different sites that are located in the same country generally have identical catchments. An example of this would be the `nrt{2-8}` sites. All 7 of these sites are located in Japan and their catchments are identical. `ams1-2` stands out in this regard because the catchment of this server is widely different than the other server located in the same country (`ams1-1`). While `ams1-1` seems to have a very good catchment, `ams1-2` is much more broad in the sense that it receives traffic from far away locations. The variations in BGP routing in this case between servers in the same country could be because of the fact that the different servers in The Netherlands are hosted on different cloud providers thus displaying varying behaviors in terms of BGP routing.

Appendix B contains catchment data that is broken down in terms of IPv4 and IPv6 traffic. IPv4 catchments are very similar to Figure 5 as majority of the traffic received at the NTP servers are on IPv4. IPv6 catchments are extremely interesting to look at because when compared to IPv4, the catchment of NTP servers in terms of IPv6 traffic is much better. The spread of IPv6 clients for each server is much narrower than IPv4 clients. It looks like IPv6 clients in general are routed to servers that are much closer to them. This difference is also evident when comparing the IPv4 catchment and IPv6 catchment of the same server. Servers `ams1-2` and `arn1-1`, for example, have a very broad IPv4 catchment. But the IPv6 catchments are restricted mostly to the same country of the server. Therefore, it looks like BGP routing performs better for IPv6 clients than IPv4 clients. The effect of this probably lesser than expected because only one out of the 4 `NTPPool` servers support IPv6. Overall, the catchment data is eye-opening in terms of revealing the underlying routing behaviour of BGP in the internet. Once `GeoDNS` picks `any.time.nl` as one of the 4 NTP servers to return to a client, BGP then takes care of routing the NTP query to the appropriate NTP server.

Taking a deeper look at clients from countries where there are anycast sites reveals more details about BGP routing. Table IV gives an indication of the queries that reached an NTP server that was located in the same country as the client. The third column represents the total number of queries sent by clients belonging to a specific country and the second column represents the number of those queries that reached an anycast

Country	Traffic remaining inside the Country	Total Queries	Percentage
US	395,029,279	483,860,277	81.64
NL	38,616,392	72,460,506	53.29
CA	29,522,098	99,079,832	29.79
ZA	55,782,678	66,903,972	83.37
IN	588,292,770	601,554,845	97.79
KR	434,309,459	440,593,438	98.57
JP	287,502,786	333,860,147	86.11
BR	69,622,891	456,232,685	15.26
AU	129,581,489	131,034,811	98.89
DE	52,117,516	60,009,979	86.84
ES	72,371,497	113,935,040	63.51
FR	35,122,726	65,299,909	53.78
PL	42,774,989	44,252,430	96.66
SE	24,988,620	26,930,556	92.78
GB	23,840,012	64,751,937	36.81
SG	211,945,802	213,348,477	99.34

TABLE IV: NTP queries that are served by an anycast site inside the country

Country	% IPv4 Queries remaining inside the Country	% IPv6 Queries remaining inside the Country
US	81.22	94.08
NL	53.89	14.22
CA	28.87	49.81
ZA	83.24	98.98
IN	97.78	98.07
KR	99.92	66.32
JP	85.12	98.3
BR	17.17	3.11
AU	98.82	99.9
DE	88.27	79.59
ES	63.39	93.53
FR	54.92	36.42
PL	96.65	97.09
SE	92.89	73.99
GB	37.8	17.07
SG	99.38	98.38

TABLE V: Table IV data grouped by IPv4 and IPv6 traffic

site in the same country. For example, in Row 1, there were a total of 483.8 million queries sent by clients that located in the US. Of these, 395 million queries were served by NTP servers in anycast sites located inside the US - `dfw1-1`, `ewr1-2`, `sea1-1`, `mia1-1`, `sjc1-1`, `ord1-1`. Conversely, 395 million clients were routed to an NTP server in the same country. An important distinction here is that this does not represent the total queries received at any of the servers. Table IV only represents a subset of the traffic that generated from countries where there are anycast sites.

From Table IV, some countries like India, South Korea, Australia, Singapore perform really well in terms of BGP routing of clients to the nearest NTP server. Clients in these countries are extremely likely to contact an NTP server that is in their country. This is backed up by the fact that all these countries have more than 96% of traffic generated by clients located inside them reaching an anycast site that is also located in the same country. This is the desired behavior in an anycast deployment in terms of network routing. This is also desirable in terms of security as user traffic stays within the same country. This has particularly been a shortcoming of IoT devices which have been observed to send traffic to countries outside which they have been deployed in [60]. On the other hand, countries like Brazil and United Kingdom do not perform well in terms of routing. In Brazil especially, only 15% of the traffic is served by the NTP server located in Sao Paulo (`gru1-2`). These could be due to various reasons including clients sticking to a particular NTP server that they used and not making a DNS query for new servers, BGP routing clients to servers outside the country that might be closer in terms of network hops, etc.

With regard to IPv4 and IPv6 traffic, the catchment maps already displayed that IPv6 clients have much higher chances to get routed to the closest NTP server. To confirm this observation, the data represented in Table IV was segregated with respect to IP version. Table V show the distribution of IPv4 and IPv6 queries. Overall, the data confirms that the IPv6 clients are much more likelier to reach an NTP server in their own country. In this US, for example, 94% of IPv6 clients are reach an anycast site inside the US when compared to only 81% of IPv4 clients. In Spain, 93% of IPv6 clients reach `mad1-1` as compared to 63% to IPv4 clients. That being said, this is not the case for all sites however, as in Brazil for example only 3% of IPv6 clients reach `gru1-1` as compared to 17% of IPv6 clients. Therefore, while the catchments of IPv6 traffic might look better for most sites, it is not the case for all sites.

In conclusion, this section looked at the server characteristics of the 30 NTP servers in terms of utilisation and anycast catchments. The most utilised and least utilised servers were identified, day-night patterns were analysed, and anycast catchments were plotted. The next section dives deep into NTP client characteristics and both the sections combine to represent the entirety of the NTP ecosystem.

Country	No. of Clients	No.of Queries	Average Queries/Client
CN	28,365,331	3,574,929,867	126.03
BR	20,407,323	456,232,685	22.35
IN	17,862,422	601,554,845	33.67
US	14,117,645	483,860,277	34.27
KR	11,946,428	440,593,438	36.88
JP	6,689,320	333,860,147	49.90
ES	4,638,306	113,935,040	24.56
GB	3,925,534	64,751,937	16.49
FR	3,713,258	65,299,909	17.58
VN	3,358,550	19,983,189	5.94

TABLE VI: Number of Clients and Queries per Country

VII. NTP Client Ecosystem

A. NTP Client Characteristics

The other side of the NTP ecosystem is made up of the clients that send NTP queries to the NTP servers. This section will dive deep into client characteristics such as NTP versions used, type of NTP queries that clients send, number of queries sent by clients, etc. in order to answer Research questions 3,4,5 outlined in §**??**. Analysis of NTP traffic sent by clients provides an understanding of how NTP is used in the internet and whether this behaviour confirms to the specifications listed in the RFC [2]. The NTP packet header and the fields contained in it was outlined in §II. The collected data was analysed to check the values that clients were using for each of these fields and compared with RFC specifications in order to determine non-compliant/misbehaving clients. Initially, clients were geolocated and the distribution of client locations was analysed to determine countries and continents with the most number of clients. A breakdown of client population by continent was already analysed in Table II which also contains data about the number of queries received per continent. A more granular breakdown by country is available in Figure 6, which depicts the distribution of clients across countries. China has the most number of clients (>28 million) with Brazil, India, US, and South Korea close behind. Table VI shows the 10 countries with the most clients and the average number of queries per client. Clients in China generate a staggering 3.5 billion queries which amounts to an average of 126 queries per client. While China has only 8 million clients more than Brazil, these clients generate 7.8 times more queries than clients in Brazil. On the other hand, Brazil has 2.2 million clients more than India but clients in India still generate 1.3 times more queries than clients in Brazil. In general, Brazil has a low average queries per client when compared to the other top countries. This implies that while Brazil has the second most clients, these clients do not generate as many queries as clients from other countries. Japan also has a high density of queries generated by a relatively low number of clients. Of the top 10 countries. Japan has the second highest average queries per client close to 50 which is only behind China.

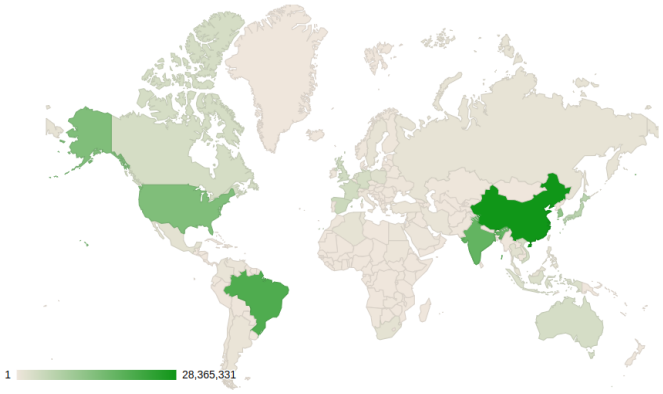

Fig. 6: Number of Queries per Country

As mentioned in §V, in the collected data there were a total of 158.7 million unique clients with the majority of them querying on IPv4. In total, clients sent an excess of 7 billion queries across the 30 servers in 1 day. Table VII shows the distribution of the NTP version across clients including the percentage of IPv4 and IPv6 queries and number of clients for each NTP version. As expected, NTP version 4 is the most widely used NTP version by clients as evidenced by 81% of total queries having this version. The NTPv4 RFC was proposed in 2010 and was adopted in 2011 [61]. The newest version of NTP also addressed a lot of security flaws that were present in previous versions of the protocol as described in §II. The NTP RFC also recommends the usage of NTPv4 for this reason. From the results obtained, it is observable that the adoption of NTPv4 has risen through the years. As mentioned in §IV-A, Rytilahti et.al [24] conducted a similar study in NTP traffic characterization and they observed that clients in Asia predominantly used NTPv3 (68%). But this number has fallen, and in this collected dataset, only 17% of clients still used NTPv3. 82% of the clients in Asia now use NTPv4 and this percentage is comparable in other continents. Therefore, the overall trend in NTPv4 adoption has gone up and this is desirable as it ensures that clients use the latest NTP version.

That being said, a significant portion of the NTP clien-

tele still seems to use outdated versions of the protocol to synchronize time. More than 17% of the total queries still use NTP version 3 which was outdated more than 10 years ago. Moreover, a minute percentage of clients are still sending queries on NTP version 1 and 2. This could indicate the usage of extremely old firmware or devices that have very limited computing power like low powered cameras, or other IoT devices. This is concerning as this could represent devices that are vulnerable to old security flaws that have since been patched.

A deep dive on the number of clients per NTP version shows that there are more clients using NTPv3 than NTPv4. Even more surprising is the fact that the number of clients using NTPv4 is 2.8 times lesser than the number of clients on NTPv3. This lower number of clients are responsible for sending the majority of NTP queries. An important point to consider is that there could be clients sending queries with different NTP versions. For example, a NAT device could be forwarding traffic from lots of clients that are located behind it. These clients could be sending queries on different NTP versions but these would be registered as a single IP sending queries with different NTP versions at the NTP servers. This could skew the number of clients per NTP version and is likely noticeable for clients with NTP version 3 and 4. That being said, it is still evident that more ≈6.9 million clients are still using NTP version 1 to synchronize time. This could, of course, be the intended behavior for some clients. Scanners, for example, send queries with multiple NTP versions in order to discover and document servers that still support/operate on these old NTP versions. But it is unlikely that these types of scanners constitute a significant portion of the client populace and the majority would still be genuine clients that still use outdated software.

Further, more than 7000 clients send more than 100k queries with invalid NTP version numbers. These include NTP queries with versions 0,5,6, and 7. The majority of clients using these invalid NTP version numbers seem to be located in the US and Canada with more than 48k queries from US and 16k queries from Canada. The usage of these NTP version numbers is against the protocol specification. There are also close to 40k clients that send NTP packets with no NTP version specified or a malformed packet from which the NTP version could not be extracted. These are represented by '`NULL`' NTP version and there are more than 1 million queries that were received across all NTP servers. The major traffic with NTP version as `NULL` was from Brazil, India, US, Argentina, and Spain. These 5 countries combined to produce more than 900k of the 1 million queries. Thus, from looking at the NTP version, while the majority of clients conform to protocol specifications, there is still widespread use of older NTP versions (especially NTPv3). This could potentially be a huge security issue as these older versions of NTP are vulnerable to various attacks. Further, there is also not an insignificant portion of the clientele that do not conform to the NTP RFC specifications and send invalid queries. Ideally, this behavior would be addressed by the operators of the client devices or software libraries as these queries still consume network bandwidth which is not desirable.

Passive fingerprinting using the IP Time-To-Live (TTL) values in client NTP packets was performed to gain a better understanding of the client OS and NTP software stack that clients use. While this might not be accurate for all cases, it still gives a rudimentary understanding of client characteristics. Table VIII has the distribution of IPv4 TTL values observed in clients. As expected, the majority of the clientele (89%) are Linux based where the OS has a default TTL value of 64. This is because devices like routers, firewalls, Android phones, IoT devices, etc. are all based on the Linux kernel and likely form the majority of NTP clients. This trend is similar in IPv6 clients where the IPv6 `hlim` field was analysed. Further, trends across continents were also in line with the total observations with the majority of both IPv4 and IPv6 clients in Asia, Europe, and America having a TTL value of 64 followed by 255. The next popular was devices with TTL values of 128 which typically indicate Windows devices including Windows server installations.

Figures 7, 8, and 9 show the distribution of the Leap Indicator (LI), Stratum, and Mode NTP fields. The Y-axis of the graphs have been converted to log scale for better visualization. The distribution of values for the Leap Indicator field is as expected with the majority of clients setting it to 0 followed by using a value of 3 and leaving the field empty. The Leap Indicator field is set in order to inform clients about impending corrections to the time which is caused due to the rotation of the earth. A value of 3 can be set for this field under a few conditions. The most common one is when a client sets the LI field to 3 and the Transmit Timestamp field to 0. This indicates that the client has never synchronized time or not synchronized to an NTP server in the previous 24 hours. In total, more than 2.8 million queries were received where clients indicated that they had not synchronized time in the past 24 hours or it was their first time synchronization. A Leap Indicator of 3 is also set for KoD packets that servers send to clients. In most cases, servers send clients a KoD packets as a form of rate control. Upon receiving a KoD packet, an RFC compliant client must increase its polling interval to the server so that it sends NTP queries much less frequently. Even though, a KoD packet is generated by servers, there were more than 20k queries received that contained a Leap Indicator of 3 and a Kiss code of 'RATE'. There are also clients that send queries with the Leap Indicator field set to 1 or 2. While these are valid values for the field, they must not be set by clients. Servers set the LI field to 1 to indicate that clients must add 1 second to their synchronized time and 2 to indicate that clients must subtract 1 second from their synchronized time. Overall, there are a significant number of queries received where clients set the LI value to 1, 2, or do not set is altogether.

The NTP stratum field is set to a number usually between 0-15 to indicate where a particular machine belongs in the NTP architecture. Servers usually have a stratum value of 1 or 2 where a value of 1 indicates that the server is directly connected to a physical reference clock. The NTP servers de-

NTP Version	No. Of Queries	% Total Queries	% IPv4 Queries	% IPv6 Queries	No. Of Clients
4	5,942,781,255	81.59	98.07	1.92	45,037,414
3	1,292,326,962	17.74	93.63	6.36	127,218,626
1	43,996,867	0.6	99.99	0.005	6,967,648
2	2,535,364	0.03	99.7	0.29	37,055
NULL	1,420,042	0.02	99.99	≈0	39,905
0	86,725	0.001	99.99	0.002	4,723
6	8,737	0.0001	98.24	1.75	1,180
7	4,397	≈0	99.97	0.02	926
5	3,138	≈0	98.94	1.05	810

TABLE VII: Distribution of NTP versions across Clients

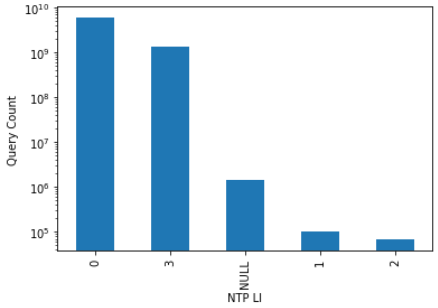

Fig. 7: NTP Leap Indicator

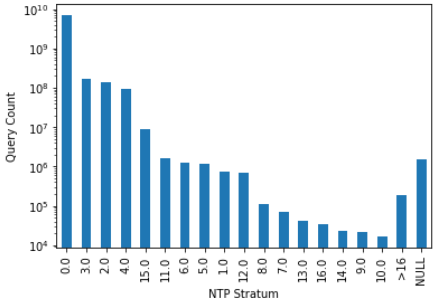

Fig. 8: NTP Stratum

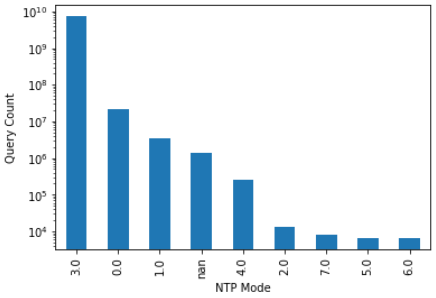

Fig. 9: NTP Mode

TTL	No. of Queries	% Total Queries
64	6,341,304,476	89.48
255	691,301,924	9.75
128	52,525,409	0.74
32	1,439,713	0.02

TABLE VIII: Distribution of IPv4 TTL values

ployed at the anycast sites have a stratum of value of 2 and they directly synchronize time from upstream Stratum 1 servers that are deployed at SIDN Labs. As expected, the majority of client queries have a stratum value of 0. According to the RFC, a value of 0 means unspecified or invalid and is usually used by clients. The next most popular NTP stratum value is 3 which would indicate direct downstream NTP servers/clients that are synchronizing time. Other stratum 2 servers also form a significant portion of the queries which indicates that other servers also use `any.time.nl` to synchronize time. After that, there are various NTP stratum values from 2-15 which indicate that the clients are secondary servers in the respective stratum. There are more than 34k queries with a stratum value of 16. This value indicates that the client is unsynchronized and serves a similar function to the LI value of 3. Finally, there is also a significant portion of clients that do not confirm to the protocol specification. More than 1.5 million queries did not have a stratum value that was set and more than 193k queries had stratum values of 17 and above when the RFC specifies that stratum values in the range 17-255 are reserved and should not be used.

The next field that is of interest is the NTP mode field. The value of this field is used to indicate the mode in which a particular machine wants to use NTP. The most commonly used modes of NTP in the internet is 3 and 4 which stands for client and server modes. This is reflected in the client queries where more than 99.6% of the received queries had an NTP mode value of 3. For a public NTP server, this is the expected behavior as these servers typically do not support other modes of usage. There are also ≈3 million queries that have an NTP mode of 1. This indicates that a client wants to use NTP in the symmetric active mode. There are about 260k packets that are received with the NTP mode set to a value of 4. Typically, a value of 4 is set by a server that sends a response to an NTP query sent by a client in mode 3. These could belong to traffic that is part of `any.time.nl` reaching out to Stratum 1 servers to periodically synchronize time. But, a deeper look into the traffic reveals that there are also other sources that send traffic with NTP mode set to 4 which is not the intended behavior as this mode is only set for replies from servers. Similar to the LI and stratum fields, there are also queries where the NTP mode is not set or set to NULL. In total, there are a bit more than 1 million queries in this category. Also similar to the LI and stratum fields is the usage of reserved values. The NTP RFC specifies that the NTP mode value of 0 is reserved but there were 21.3 million queries that were received with 0 as the NTP mode which is

non-compliant behavior. Lastly, there were about 8000 mode 7 queries and 6500 mode 6 queries received at the NTP servers. As mentioned in §IV-B, mode 6 and 7 NTP queries are usually used in DoS/DDoS amplification attacks as the size of the response returned for these queries are much bigger than the requests. These modes are used to send and receive control messages to NTP servers for diagnostic purposes and should not generally be enabled for public servers. `any.time.nl` does not support mode 6 and 7 queries so clients sending these queries do not get an answer. But the presence of these queries is interesting as they are evidence of attempts at using the servers to perform malicious activities or the presence of malicious or non-malicious scanners that test NTP servers to see if they are vulnerable. The results of a deeper look at NTP Mode 6 and 7 queries are presented later in the paper. Overall, analysis of the NTP Leap Indicator, Stratum, Mode fields show that the vast majority of the clients set expected values for these fields but there is a significant number of clients sending queries that are not RFC compliant due to the usage of unsupported or reserved values for some or all of these fields.

B. NTP vs. SNTP Queries

§II-C gave a brief introduction to the Simple Network Time Protocol and its uses. From previous related work (§IV-A), the parameters that clients set for SNTP queries was determined and queries with these parameters were analysed to get an idea of the ratio of SNTP to NTP queries along with the geographical and temporal distribution of SNTP queries. SNTP was designed to be inter-operable with NTP and all NTP servers are thus capable of responding to both NTP and SNTP queries. Since SNTP is a simplified version of the Network Time Protocol it uses the same NTP header format but clients do not set all the fields as they would for an NTP query. Most commonly, in an SNTP query, the client usually sets the Leap Indicator, Stratum, Poll, and Reference ID fields to 0. Further, the NTP timestamp fields including the NTP reference time, origin time, and receive time are also set to 0. Using these parameters, received queries can be filtered to approximate the number of SNTP clients vs NTP clients. Out of the total 7.28 billion queries, 1.55 billion queries were filtered using the above parameters which gives the number of SNTP queries. Therefore, SNTP queries make up about 21% of the total traffic that was captured. This is also a reduce from the 45% of SNTP traffic observed by Rytilahti et.al [24] in their study but the results in this case are not comparable as the authors of the other study analysed traffic that was captured for a longer duration from NTP servers deployed in 3 locations.

Clients from India sent the most number of SNTP queries at 354.68 million followed by China with 294.5 million, Brazil with 179.2 million, South Korea with 112.9 million queries, and the US with 103.8 million queries. While the top 5 countries with the most SNTP queries are the same as the top 5 countries with the most NTP queries (Figure 6), there are a greater number of SNTP queries from India. In general, Asia has the majority of SNTP queries followed by

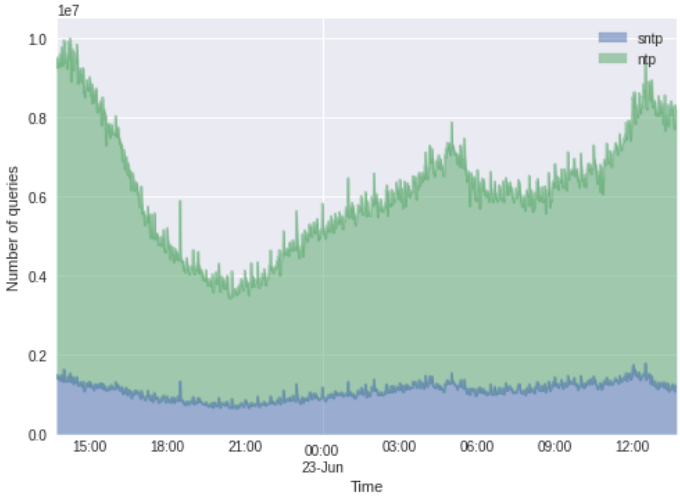

Fig. 10: Distribution of SNTP vs NTP queries

South American and North American continents. Apart from the India standing out in the number of SNTP queries, the geographical spread of queries and clients are in-line with the geographical distribution of NTP queries. Figure 10 shows the temporal distribution of SNTP and NTP queries across the 30 sites. Since Asia has the most number of queries, the diurnal trends analysed in §VI-A is prominent in the NTP queries across time even when queries from all 30 sites are combined. Surprisingly, the same diurnal pattern is not observed in the SNTP queries. While there is a dip in the SNTP queries in the same time period as the reduce in NTP queries, it is not as pronounced, which implies that SNTP queries are not subjected to same the day-night patterns of NTP queries. The reason for this deviation in pattern can be found by looking at the types of clients that typically use SNTP. Since SNTP is a simpler version of NTP, it is most commonly used by automated programs that require not too precise time information such as micro-controllers, and small computers. Since this is not the case for most devices used by humans such as smartphones, computers, etc., the same trend of reduced queries during the night is not visible for SNTP queries.

C. The Great NTP Flood

Previous analysis of NTP client characteristics focused on client population distribution, NTP versions used, and the usage of NTP vs SNTP. In this section, deeper analysis is done into the frequency of client requests in terms of number of requests per client IP to identify anomalous NTP clients. Before looking into how often clients send NTP requests, it is important to understand when NTP requests are sent by clients. As explained in §II-A, the NTP packet header contains the Poll Interval field that servers set in order to inform clients on how often they can send requests to that particular NTP server. Clients use the value set in this field as the exponent to 2 and calculate the minimum time they need to wait before sending an NTP request to that server.

Client IP	Queries	Time frame	Queries/Second	Responses	% Responses / Queries	Country
192.33.151.159	8,713,100	87830	99.20	5974	0.068	KR
25.193.210.30	7,571,623	87551	86.48	84	0.001	PH
5.52.96.86	6,313,197	87838	71.87	227	0.0035	KR
178.20.203.52	5,501,880	85854	64.08	1067	0.019	JP
95.8.85.197	5,490,735	87838	62.50	0	0	ZA
217.18.48.60	4,843,415	87838	55.14	2299	0.047	KR
16.230.152.164	4,183,282	61968	67.50	4579	0.1	IN
192.115.161.111	3,048,685	87838	34.70	0	0	KR
10.104.238.155	2,614,373	85183	30.69	1377	0.05	SG
10.224.182.21	2,572,870	87838	29.2	344	0.013	KR

TABLE IX: Number of Queries sent by the Top 10 IPs (Anonymized)

For example, if a server sets this value to 10, clients need to wait a minimum of 2^{10} (1024) seconds before sending the next NTP request. The Poll interval field is present so that clients do not send a flood of NTP requests to servers and exhaust server and network resources. Another form of rate limiting available to NTP servers is the KoD packet. If a server detects that a client is ignoring the Poll Interval and sending an excessive amount of queries, it can reply with a KoD packet containing the 'RATE' code and clients must necessarily stop sending requests to the server for the duration of the value set in the Poll Interval. Clients also usually have different configuration options that are dynamically adjusted that govern how often they send requests. These *minpoll* and *maxpoll* settings govern the minimum and maximum duration between requests. The default values used in various client implementations like `xntpd`, etc are 6 for *minpoll* and 10 for *maxpoll*. These values would translate to a duration of 16s and 4.5 hours for the 2 options. Usually, 4 is lowest possible option for *minpoll* while *maxpoll* can be set to higher values like 17. During time synchronization, clients usually dynamically adjust these parameters depending on the value that the server sets in the Poll Interval field of the NTP response. In the dataset, the most commonly used values for the Poll Interval field by the NTP servers is 4. For more than 3.5 billion queries, poll interval was set to 4 and for 2 billion queries, it was set to 3.

It is usually observed by NTP server operators that NTP clients send more queries than they should to servers. This would mean that clients do not respect the value in the poll interval or KoD packets. To dig deeper into this phenomenon and identify rogue clients, analysis was done into how many queries each client IP sends and the time duration in which a client IP is seen. Table IX shows the top 10 IPs that send the most number of queries and the time frame (in seconds) during which the IP was seen. The IP addresses are all anonymized using the method described in §V. The top IP sent a total of 8.7 million queries throughout the entire day of the capture with a an average queries per second close to 100. This is a lot more than the lowest *minpoll* setting of 4 which translates to 1 query per 16 seconds. The actual number of queries received from the top IP translates to 1600x the number of queries that the client can actually send. This clearly constitutes anomalous behavior by clients. One possible explanation for the excessive number of queries could be that the client IPs could belong to ISPs that perform Carrier Grade NATting (CGNAT). This is especially common in continents like Asia where the high number of users force ISPs to use CGNAT to avoid running out of public IPv4 addresses. CGNAT is a technique where a large number of users are fronted by a single public IP. In this case, there could potentially by thousands of clients behind a single IP that are sending NTP requests. Looking at the Poll Interval that the servers set for responses to these IPs, 4 is the most commonly used value with the NTP servers even setting a poll interval of 17 for some IPs. This means that when an IP gets a response with a poll interval of 17, it must not send any further NTP queries for a minimum of 1.5 days. But, it is clear that client IPs do not respect the value set in the poll interval as evidenced by the number of queries sent to NTP servers. In case of presence of CGNAT or other NAT techniques, it might not be possible for the NAT device to honour the poll interval as it only performs the role of forwarding requests from clients that located behind it. Regardless, it is evident that clients send an excessive number of queries to NTP servers even after the maximum poll interval was sent in response back to the clients. This would imply that the clients do not comply with RFC in terms of sending the appropriate number of queries.

Table IX also shows the number of responses that the top 10 client IPs got in response from the NTP servers. The IP with the most number of queries only received 5974 responses for more than 8.7 million queries sent. Since `any.time.nl` implements rate limiting to preserve network bandwidth and server resources, if clients send more queries than they are supposed to, they will not receive responses to these queries as is shown by the extremely low ratio of responses to queries. In all cases, clients only get responses for less than 0.1% of the queries sent and in most cases clients only get responses

to about 0.01% of the queries it sends. Looking at the AS that the IPs belong to gives a better understanding of the client. It is important to note that the IPs themselves are anonymized but geolocation and ASN information was added to the dataset before anonymization was performed. The top IP belongs to the LG Dacom corporation in South Korea. Looking at the `whois` info of this ASN shows that it belongs to one of the largest telecom network operators in Korea. Hence, it is reasonable to conclude that this IP is probably a public NAT IP that is used by a huge number of mobile clients to synchronize time. Similarly, the second IP belongs to the ASN of an ISP that is located in Philippines. Therefore, this IP also is the NAT IP that users of the ISP use. Of the top 10 IPs that sent the most number of queries, 9 of them are located in Asia, with 5 of them located in South Korea. The outlier is an IP that is located in South Africa. The ASN that this IP belongs to was identified and subsequently the company's website was identified using `whois` information. This IP from South Africa belongs to a popular mobile network operator similar to the first IP. The third and tenth IP that is located in South Korea belongs to another network operator, Korean Telecommunications Authority (KT) while the fourth IP from Japan belongs to NTT. In fact, all the IPs in the top 10 most talkative list belong to an ISP or a mobile network operator. Therefore, it is safe to assume that all these IPs are not actual clients, but rather NAT IPs that have many thousands of clients behind them. This is also further validated by looking at the distribution of queries sent by the top IP `192.33.151.159` over time. Figure 11 shows the distribution of queries received from this IP throughout the day. The number of queries are broken down per minute and visualized. It is evident that the traffic pattern has a sudden burst in the number of queries and then there are little to no queries sent after that. That being said, it appears that there is no discernable pattern to the time between successive bursts of queries. The traffic distribution for all the top 10 IPs looks similar with dips and troughs in terms of queries sent. This shows that the possible devices that are NATted behind this IP do not keep sending requests. Rather, most of the devices send a burst of NTP queries at a single time. This is not desirable behavior in terms of client time synchronization. The reason is because of the number of responses that is returned to this IP. Not all users that are located behind this IP will get an answer to their NTP query. A better implementation by ISPs and mobile network operators would be to localise the NTP service possibly through DHCP. Currently, time synchronization is done by clients depending on the NTP server that is configured in their operating system. Disseminating NTP server IPs through DHCP will allow for successful time synchronization behavior and avoid traffic like the pattern in Figure 11. This will also lead to conservation of network bandwidth in the form of excessive and unwanted NTP queries.

Since the top 10 IPs that send the most queries overall mostly to Asia, further analysis was done to get the top talkers in other continents and see if the situation was similar to Asia. Appendix C contains the detailed breakdown of the top IPs that

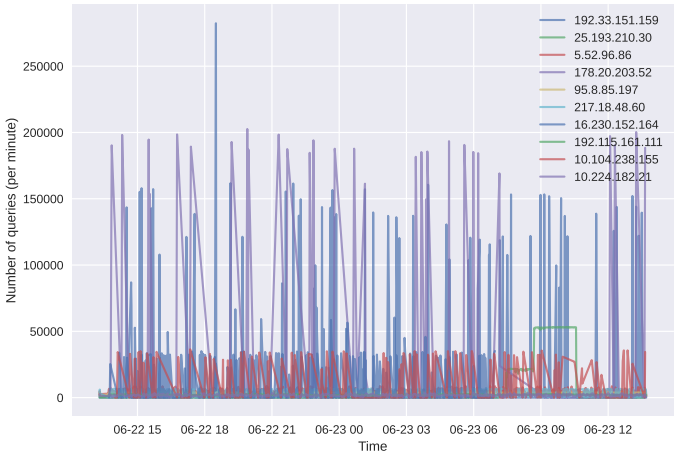

Fig. 11: Distribution of Queries sent by Top 10 IPv4 Clients

send the most traffic from the North American and European continents. Table XIII shows the top 10 IPs (anonymized) that send the most traffic from the North American Continent. 7 out of the 10 IPs are from the US, while the rest are from Canada. The top IP sends close to 1.2 million queries through the day with an average of 13 queries/second. The other IPs send between 3-9 NTP queries per second on average. While this is much better than the clients in Asia most of which more than 50 queries per second, it is still more queries than allowed and is not compliant to the NTP RFC. It is interesting to note that none of the top 10 IPs in the North American continent received any replies from the NTP servers that they contacted. The ASN `whois` info of the top IP from Canada reveals that the IP belongs to Rogers communication which is one of the largest mobile networks and home internet providers in the country. Similarly, the second IP belongs to Cogeco which is another ISP and mobile network provider in Canada and the third IP belongs to Pilot Fiber which is an ISP in the US. The rest of the IPs also belong to various other ISPs, mobile network providers, or independent fiber bandwidth companies except 2 of the IPs which belonged to large Financial services firms. Therefore, even the top IPs that send traffic in North America mostly fall under the same category as Asia and most likely have some form of NATting implemented in front of clients.

In Europe, a similar observation is made with the data available in Table XIV. While the number of queries and the average queries per second sent by the top 10 clients in Europe is extremely similar to the top 10 clients in North America, there is much diverse spread of countries. The top 10 IPs belong to Spain, Netherlands, Germany, Sweden, and France. The IP with most queries sent 2.5 million queries at 18.5 queries per second. But other than particular IP, the other IPs sent about 4-12 queries per second on average. Contrary to the clients in the US, all the top 10 IPs from Europe received responses. But in this case, the number of responses are in the range of the Asian clients with most clients in Europe getting responses for less than 0.1% of the queries except for

Client IP	Queries	Time frame	Queries/Second	Responses	% Responses / Queries	Country
6a63::7581	40248	21872	1.84	732	1.81	BR
6a63::8aa9	25844	76553	0.33	616	2.38	BR
63ab::5492	17302	87833	0.19	8642	49.9	JP
6032::a004	12940	81000	0.15	12902	99.7	US
6a60::b175	12850	85260	0.15	12818	99.75	MX
6763::da11	12054	83721	0.14	11604	96.26	NL
6763::5c8c	10105	63470	0.15	10011	99.06	CA
6031::8d95	9264	67502	0.13	9258	99.93	US
63a2::a3ea	8638	87829	0.098	8618	99.76	JP
63ac::29cc	8386	52409	0.16	8378	99.9	PK

TABLE X: Number of Queries sent by the Top 10 IPv6 IPs (Anonymized)

the last IP which stands at 1%. The ASN `whois` paints a familiar picture with the top IP belonging to Telefonica which is a mobile service provider in Spain. The last IP also belongs to Telefonica so it seems like there is a lot of traffic from this network. The second IP from the Netherlands belongs to SURF, which is a research collaboration between educational institutions. The third IP belongs to the EUNetwork which is located in Germany and provides bandwidth services. All the other IPs belong to various ISPs or backbone operators in the respective countries with the exception of `39.189.129.73` which belongs to a digital services and software firm. While the behavior in terms of the number of queries sent by clients in Asia, North America, and Europe might be different, the possible cause behind these large number of queries remain constant among clients from all around the world. While CGNAT is quite common in Asian countries due to the large number of clients, it is surprising to see some ISPs in Europe and North America also implement this. The large presence of mobile network operators in clients with the most number of queries makes sense because the users in this case are mobile phones and the sheer number might cause operators to implement some form of NAT solution.

To gain a better understanding of client behavior across IP versions, the top clients on IPv6 were also analysed to see how many queries these clients send. Table X has the number of queries sent by the top 10 most talkative IPv6 clients. Only the first and last octet of the anonymized IPv6 IP is presented in the table. The data is a stark contrast to IPv4 client behavior. While the number of queries sent by IPv6 clients is obviously much lower than the number of queries sent by IPv4 clients, the main difference is in the queries per second and percentage of traffic that received a response. Except the top 2 IPs, all IPv6 IPs got responses to almost 100% of the queries they sent. Further, IPv6 clients in the majority of cases only send about 0.1 queries per second which roughly equates to about 1 query every 10 seconds. The most common value for Poll Interval set by NTP servers for IPv6 clients is 3. This means that clients can send queries every 8 seconds. Looking at the data, except the top 3 IPv6 clients, every IPv6 client follows the polling interval while sending NTP queries. This is also further strengthened by the fact that almost no IPv6 client gets rate limited.

The overall trend in client queries is quite troubling as IPv4 clients tend to waste a lot of bandwidth sending NTP queries for which they get no response. Further, due to widespread NAT implementations in Asia and other parts of the world, a vast majority of actual users might end up not getting responses for their queries. Further, since most NAT implementations simply forward queries and responses, they do not respect the values for fields set in the NTP header. This leads to widespread non compliance of the NTP RFC and also leads to wasted network bandwidth. Looking at the top 10 IPv4 clients, more than 99% of the queries they send are unanswered. The average packet length in a small subset of the captured data was found to 90 bytes. Therefore, traffic sent only by the top IP would amount 0.78 GB out of which 99% is wasted. Extrapolating this to only the top IPv4 IPs leads to a bandwidth wastage of more than 4.5 GB for the 10 clients. Considering that there are 132 million IPv4 clients, the actual network resources would be a lot higher. Therefore, it is important to implement solutions that use network resources frugally and are compliant with the NTP RFC.

D. Malicious NTP Clients

Recently, NTP has been used in multiple amplification and reflection DDoS attacks as described in §IV-B and §II-B. This is because NTP, like DNS, is mainly a UDP based protocol which allows attackers to spoof queries. Also, NTP has certain queries like *monlist*, *peerlist*, etc. that return a response that has a much larger packet size than the request. This allows attackers to craft spoofed requests with the IP address of a victim that are small in size and send these requests to a public NTP servers. These public servers in turn send a large response to the victim thus inundating it with large amounts of unsolicited traffic, leading to a DDoS or a DoS attack.

The *monlist* and *peerlist* queries are administrative queries built-in to the protocol in order for server administrators to

Client IP	Queries	Anycast Site	Country
88.162.105.2	999	mia1-1	US
195.158.75.168	507	sjc1-1	US
212.252.246.84	258	lhr1-1	FR
195.158.90.138	257	sjc1-1	US
84.67.213.125	144	cdg1-1	GB
194.111.154.223	142	ord1-1	US
217.57.175.119	124	sin1-1	LA
217.57.175.119	106	sin1-2	LA
189.184.250.159	67	mia1-1	US
88.53.68.117	65	mia1-1	BR

TABLE XI: Top 10 IPs (Anonymized) that sent NTP Mode 7 Queries

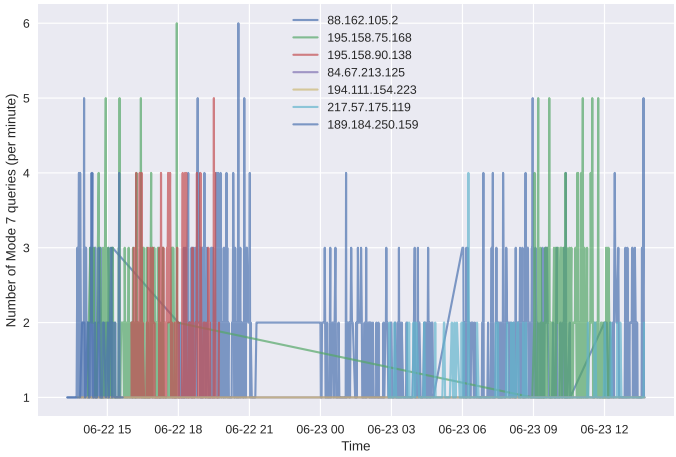

Fig. 12: Distribution of Mode 7 Queries

troubleshoot or gather diagnostic information from their NTP servers. Contrary to client-server model where a client sends a packet with NTP mode 3, these queries use NTP mode 7 which the RFC describes as "reserved for private use". Hence, it is clear that NTP in mode 7 is not really intended for clients to use with a public server. During the height of NTP based DDoS attacks, there were hundreds of thousands of public NTP servers in which mode 7 operation was available for public use. As time went, server administrators began to disable these modes for the public clients so that their servers were not vulnerable. That being said, there are still vulnerable NTP servers and malicious actors that look to use these servers in various attacks.

Since queries of these kind use a separate NTP mode of operation (7), it is quite easy to identify clients that send these type of queries to `any.time.nl`. In the collected dataset, there were a total of 8136 queries sent by 1021 clients across the 30 servers. Table XI shows the top 10 IPs that sent the most amount of Mode 7 queries along with the anycast site that these IPs reached and the country that the IP belongs to. While the IP `88.162.105.2` sent almost 1000 queries to the `mia1-1` site, further analysis into these queries show that the IP also sent 3-5 queries to all the other 29 sites. This is not normal behavior as once clients are routed to a particular site, further queries will be sent to that particular site. There are of course some deviations where clients can send queries to different sites in the same country like `sin1-1` and `sin1-2`. At the outset, this behavior looks like suspicious and typical of scanners. From Figure 12, it is apparent the client IPs send queries in bursts. Some IPs like `195.158.90.138` are only active once throughout the day while others like `195.158.75.168` has sent 2 bursts of queries in the 24 hour duration. Further, contrary to normal traffic sent by clients in Figure 11, there are periods where these IPs do not send any requests.

To get a better understanding of the exact queries that are sent, some timestamps at which the top IP `88.162.105.2` sent these mode 7 requests was gathered and the exact query details was studied. This revealed that these IPs send a combination of *monlist* and *peerlist* commands. Since `any.time.nl` does not support Mode 6 or 7 operation, none of these clients got a response to their queries. Further, when looking at the queries sent, it was observed that the destination IP that this client sent the mode 7 requests to did not match the IPv4 IP of `any.time.nl` which is `194.0.5.123`. Rather, the destination IP in this case across multiple sites belonged the unicast IPv4 address that was assigned to the server. This is common in anycast deployments where each server advertises an anycast IP that is common to all the anycast sites but also has a separate IP address for management purposes. The fact that the mode 7 queries were sent to the unicast IP of servers further points towards these clients being scanners as scanners usually send these queries to the entire IPv4 range to identify and document vulnerable NTP servers. It was also confirmed that all the top 10 IPs did not send requests to `any.time.nl` but rather sent requests to the unicast IP addresses of the different servers. The ASN info for the IP `88.162.105.2` reveals that it belongs to Frantech solutions which is a VPN and hosting provider that is no longer providing services under this name. Since the company also provides VPN services, combined with the fact that the IP address sent requests to all 30 sites makes this indicative of a malicious scanner. The reputation scores for this ASN and indicate that this ASN has a spam rate of close to 65% from this AS [62] which further points to the fact that this IP could belong to a malicious scanner looking for vulnerable NTP servers to exploit. Out of the top 10 IPs in Table XI, the first, second, and fourth IPs - `88.162.105.2`, `195.158.75.168`, `195.158.90.138` all belong to the same AS owned by Frantech solutions. The third IP `212.252.246.84` belongs to a hosting provider in France and the fifth IP `84.67.213.125` belongs to another hosting provider with reports showing it having more than 700 active spam IPs [63]. Similarly, all the top 10 IPs seem to have some amount of spam activity giving more credence to the argument that these are malicious scanners. Most of these malicious clients seem to originate from the United States with

5 of the 9 distinct IPs originating from there.

Compared to IPv4 clients, only 2 IPv6 clients sent mode 7 queries to the NTP server. Both of the IPv6 clients combined to send just 19 mode 7 queries with one IP sending 11 and the other sending 8. Both these IPv6 clients sent queries to the `bom1-2` site and originated from India. The AS info for the IPs revealed that the IPs belonged to Reliance Jio Info Communication which is a public hosting service and offers a paid VPN. Reputation statistics show that there is 2.55% spam rate in this AS with over 10k IPs being tagged with spam activity. Thus, there are high chances that these IPs are also scanners that look for vulnerable NTP servers. Overall, the number of malicious (mode 7) NTP queries compared to general NTP traffic is quite less. Therefore, while there were no traces of large scale DDoS attacks for the duration of data collection, the presence of numerous clients that look like scanners is suspicious and proves that NTP as an attack vector for DDoS and other malicious activity is still viable. The results show that it is extremely important to implement the correct safeguards on NTP servers in order to protect them from being misused by malicious actors. While the total number of mode 7 queries are miniscule compared to the traffic captured, it still proves that scanners must send traffic responsibly. None of the top 10 IPs that sent mode 7 requests had valid reverse DNS records. If these IPs were genuine scanners, it is important to perform data collection in an ethical manner so as to avoid being tagged as malicious.

VIII. Conclusion and Future Work

In this thesis, traffic from an anycast NTP server listed in `NTPPool` was collected for a day and analyzed. Since NTP is a core protocol on the internet similar to DNS and it is critical to the functioning of devices, this data was instrumental in providing an in-depth look at the state of the current NTP ecosystem. Extensive analysis of the data reveals that `NTPPool` provides a crucial service to users but direly needs more servers to handle NTP traffic outside of North America and Europe, especially in Asia and Africa. The anycast server provided by SIDN takes crucial steps to address this scarcity of servers and similar initiatives are the need of the hour. The anycast deployment proves to be extremely capable of handling the demand of clients in the current NTP ecosystem especially since it is deployed in locations that do not have many other options for NTP servers. Results also show that while the adoption of the latest NTP version has grown compared to previous years, a significant portion of clients still use outdated NTP versions. There is a lot of RFC non compliant behavior among clients with respect to setting reserved or incorrect values in NTP packets. Further, architectural designs of ISPs and network providers actively hamper NTP usage by flooding servers with excessive NTP queries. With the current architecture, in-built server rate limiting is also not effective and only serves to denigrate the client's experience when using the protocol. Lastly, traces of malicious traffic was found in the dataset and analysed which leads to the conclusion that NTP is still a viable attack vector and servers administrators must take steps to protect their deployments.

Specifically, the thesis answers the research questions laid out in §I. §VI discusses the results for the analysis regarding anycast site utilisation and anycast catchments. It was seen that the NTP servers deployed in Asia saw a lot of usage when compared to the servers in Europe and other continents. Interestingly, Asia has fewer number of servers listed in `NTPPool` to handle traffic that many multitudes more than the other continents. Anycast site heatmaps were also plotted with the data available and these maps showed that BGP routing in the internet is not always exact and traffic can often go to unexpected places. Since this research, the administrators of the NTP server at SIDN Labs have already improved the catchments of various sites by tweaking various BGP configuration settings. Therefore, it is important to keep in mind that these heatmaps can easily influenced by various factors.

§VII provides a detailed analysis and discussion on the clients that use the NTP service and answers research questions 3,4,5 regarding client characteristics, anomalous clients, and traces of malicious traffic. The client population was overwhelmingly Linux based devices and this makes sense as Windows has its own NTP service that Windows devices use by default and this is the case for other manufacturers like Apple too. As mentioned before, there is a lot of non compliant RFC behavior that is observed particularly in IPv4 clients. This behavior ranges from setting non supported values in the NTP header to sending much more than the allowed number of queries to an NTP server. Further analysis on IPs that send a lot of requests showed that this was because of widespread usage of NAT by ISPs and telecom network operators. This directly leads to the majority of NTP bandwidth being unwanted. Thus, a main motivation of ISPs and network operators could be to push to wider IPv6 adoption. IPv6 would alleviate a lot of the problems that were observed with respect to the flood of NTP queries from NAT IPs.

In conclusion, this thesis answers the main research question regarding traffic characteristics of a public NTP server that is deployed using anycast. Data was collected from all 30 sites of the anycast deployment and extensively analysed with respect to various perspectives. This thesis provides an up to date overview of the NTP ecosystem and highlights various issues faced by operators of public NTP servers.

In the future, more work can be done on the following topics: -

Expansive Data Collection and Analysis - In this study, data was collected and analysed for 1 day. While this gives an accurate impression of the NTP ecosystem, future research could involve collecting data multiple times for a longer time period from an anycast NTP server deployment. This will also have the advantages of addressing some phenomenon like DHCP churn, etc.

NTP Client Daemon Behavior - While NTP clients and their software was studied, more emphasis was given to the traffic clients generate. Future research regarding client soft-

ware and their behavior in terms of server selection, fallback, etc. would provide valuable insights into client traffic patterns.

IX. Acknowledgements

I would like to extend my deep gratitude to SIDN Labs for providing the resources that made this research possible and especially thank Giovane for his review and feedback and Marco for providing the dataset and valuable insights into NTP. I would also like to thank Anna from the University of Twente for the continued support and feedback throughout the research.

References

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

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

[3] J. Postel, "Internet Protocol," IETF, RFC 791, Sep. 1981. [Online]. Available: http://tools.ietf.org/rfc/rfc0791.txt

[4] ——, "User Datagram Protocol," IETF, RFC 768, Aug. 1980. [Online]. Available: http://tools.ietf.org/rfc/rfc0768.txt

[5] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF, RFC 5246, Aug. 2008. [Online]. Available: http://tools.ietf.org/rfc/rfc5246.txt

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

[7] G. C. M. Moura, J. Heidemann, R. de O. Schmidt, and W. Hardaker, "Cache me if you can: Effects of DNS Time-to-Live," in *Proceedings of the ACM Internet Measurement Conference*. Amsterdam, the Netherlands: ACM, Oct. 2019, pp. 101–115. [Online]. Available: https://www.isi.edu/~johnh/PAPERS/Moura19b.html

[8] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol," IETF, RFC 6810, Jan. 2013. [Online]. Available: http://tools.ietf.org/rfc/rfc6810.txt

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

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[11] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[12] R. Droms, "Dynamic Host Configuration Protocol," IETF, RFC 2131, Mar. 1997. [Online]. Available: http://tools.ietf.org/rfc/rfc2131.txt

[13] R. Gayraud and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6," IETF, RFC 5908, Jun. 2010. [Online]. Available: http://tools.ietf.org/rfc/rfc5908.txt

[14] Google, "Google Public NTP," https://developers.google.com/time, May 2022.

[15] Microsoft, "Microsoft NTP Service," http://time.windows.com, May 2022.

[16] Apple, "Apple NTPService," time.apple.com, May 2022.

[17] Cloudflare, "Cloudflare Time Service," https://www.cloudflare.com/time/, May 2022.

[18] NTP Pool, "pool.ntp.org: the internet cluster of ntp servers," https://www.ntppool.org/en/, May 2022.

[19] ——, "The NTP Pool for vendors," https://www.ntppool.org/en/vendors.html, May 2022.

[20] ——, "All Pool Servers," https://www.ntppool.org/zone, May 2022.

[21] ——, "How do I use pool.ntp.org?" https://www.ntppool.org/en/use.html, May 2022.

[22] ——, "SIDN's Pool Servers," https://www.ntppool.org/a/TimeNL, May 2022.

[23] D. McPherson, D. Oran, D. Thaler, and E. Osterweil, "Architectural Considerations of IP Anycast," IETF, RFC 7094, Jan. 2014. [Online]. Available: http://tools.ietf.org/rfc/rfc7094.txt

[24] T. Rytilahti, D. Tatang, J. Köpper, and T. Holz, "Masters of time: An overview of the ntp ecosystem," in *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, 2018, pp. 122–136.

[25] J. A. Sherman and J. Levine, "Usage analysis of the NIST internet time service," *Journal of Research of the National Institute of Standards and Technology*, vol. 121, p. 33, Mar. 2016. [Online]. Available: https://doi.org/10.6028/jres.121.003

[26] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 435–448. [Online]. Available: https://doi.org/10.1145/2663716.2663717

[27] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 3, p. 38–47, jul 2001. [Online]. Available: https://doi.org/10.1145/505659.505664

[28] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, February 2014.

[29] SIDN Labs, "Security flaws in Network Time Protocol make other (security) protocols vulnerable," https://www.sidn.nl/en/news-and-blogs/security-

flaws-in-network-time-protocol-make-other-security-protocols-vulnerable, Dec. 2020.
[30] A. Malhotra and S. Goldberg, "Message Authentication Code for the Network Time Protocol," RFC 8573, Jun. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8573
[31] B. Haberman and D. Mills, "Network Time Protocol Version 4: Autokey Specification," IETF, RFC 5906, Jun. 2010. [Online]. Available: http://tools.ietf.org/rfc/rfc5906.txt
[32] D. F. Franke, D. Sibold, K. Teichel, M. Dansarie, and R. Sundblad, "Network Time Security for the Network Time Protocol," RFC 8915, Sep. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8915
[33] E. Rescorla, "Keying Material Exporters for Transport Layer Security (TLS)," IETF, RFC 5705, Mar. 2010. [Online]. Available: http://tools.ietf.org/rfc/rfc5705.txt
[34] SIDN Labs, "Security for Network Time Protocol standardised in RFC 8915," https://www.sidn.nl/en/news-and-blogs/security-for-network-time-protocol-standardised-in-rfc-8915, Dec. 2020.
[35] D. Mills, "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI," IETF, RFC 4330, Jan. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4330.txt
[36] A. von Bidder, "ntp DNS round robin experiment," https://groups.google.com/g/comp.protocols.time.ntp/c/cShrN7imCJ0, Jan. 2003.
[37] Ask Bjørn Hansen, "GeoDNS servers," https://github.com/abh/geodns/, May 2022.
[38] G. C. M. Moura, M. Davids, C. Schutijser, and C. Hesselman, "New (and Old) Kid on the block: GeoDNS Authoritative Server," 2021. [Online]. Available: https://www.sidnlabs.nl/downloads/4mNvwNvIShXDba7uq3ygIY/6d3a51c4920cc435335be8a28228faad/New__and_Old__Kid_on_the_block_GeoDNS_Authoritative_Server.pdf
[39] Maxmind, "Maxmind," 2021. [Online]. Available: http://www.maxmind.com/
[40] P. Mockapetris, "Domain names - concepts and facilities," IETF, RFC 1034, Nov. 1987. [Online]. Available: http://tools.ietf.org/rfc/rfc1034.txt
[41] G. C. M. Moura, M. Davids, C. Schutijser, and C. Hesselman, "Diving into the NTP Pool," 2022. [Online]. Available: https://www.sidnlabs.nl/downloads/5aPx86UtFmvKs6WE3LHwbU/acfcf45571436eca4eb924f62bdddc10/Diving_into_the_NTP_Pool.pdf
[42] SIDN Labs, "SIDN Labs Netherlands," https://www.sidnlabs.nl/en.
[43] ——, "TimeNL Comes of Age," https://www.sidnlabs.nl/en/news-and-blogs/timenl-comes-of-age, Apr. 2022.
[44] EUSPA, "Galileo Global Satellite Navigation System," https://www.usegalileo.eu/EN/about.html.
[45] V. Shankarkumar, L. Montini, T. Frost, and G. Dowd, "Precision Time Protocol Version 2 (PTPv2) Management Information Base," IETF, RFC 8173, Jun. 2017. [Online]. Available: http://tools.ietf.org/rfc/rfc8173.txt
[46] SIDN Labs, "Anycast2020," http://dnstest.nl/anycast2020/heatmaps/.
[47] NTP Pool, "194.0.5.123 / any.time.nl," https://www.ntppool.org/scores/194.0.5.123.
[48] ——, "2001:678:8::123 / any.time.nl ," https://www.ntppool.org/scores/2001:678:8::123.
[49] A. Malhotra, I. E. Cohen, E. Brakke, and S. Goldberg, "Attacking the network time protocol," in *Proceedings 2016 Network and Distributed System Security Symposium*. Internet Society, 2016. [Online]. Available: https://doi.org/10.14722/ndss.2016.23090
[50] libpcap, "TCPDUMP/LIBPCAP public repository," 2012. [Online]. Available: http://www.tcpdump.org/
[51] ICANN, "Global Policy for Allocation of ASN Blocks to Regional Internet Registries," 2012. [Online]. Available: http://www.icann.org/en/news/in-focus/global-addressing/global-policy-asn-blocks-31jul08-en.htm
[52] The ANT Lab, "cryptopANT IP Address Anonymization Library," https://ant.isi.edu/software/cryptopANT/index.html.
[53] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128604001197
[54] NTP Pool, "Asia — asia.pool.ntp.org," https://www.ntppool.org/zone/asia.
[55] ——, "Europe — europe.pool.ntp.org," https://www.ntppool.org/zone/europe.
[56] NTP Pool Community Forums, "The time has come: we must enable IPv6 entirely," https://community.ntppool.org/t/the-time-has-come-we-must-enable-ipv6-entirely/1968.
[57] L. Quan, J. Heidemann, and Y. Pradkin, "When the Internet sleeps: Correlating diurnal networks with external factors," in *Proceedings of the ACM Internet Measurement Conference*. Vancouver, BC, Canada: ACM, Nov. 2014, pp. 87–100. [Online]. Available: http://www.isi.edu/%7ejohnh/PAPERS/Quan14b.html
[58] J. Abley and K. Lindqvist, "Operation of Anycast Services," IETF, RFC 4786, Dec. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4786.txt
[59] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF, RFC 4271, Jan. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4271.txt
[60] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information Exposure for Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach," in *Proc. of the Internet Measurement Conference (IMC)*, 2019.
[61] EECIS, University of Delaware, "NTP Version 4 Release Notes," https://www.eecis.udel.edu/~mills/ntp/

html/release.html.
[62] CleanTalk, "Spam Stats for AS53667," https://cleantalk.org/blacklists/as53667.
[63] ——, "Spam Stats for AS15830," https://cleantalk.org/blacklists/as15830.

APPENDIX A
LIST OF NTP SERVERS

Table XII: NTP Servers

NTP Server	Location
ams1-1	Amsterdam/Netherlands
ams1-2	Amsterdam/Netherlands
arn1-1	Stockholm/Sweden
bom1-1	Bombay/India
bom1-2	Bombay/India
cdg1-1	Paris/France
dfw1-1	Dallas/United States
ewr1-2	Newark/United States
fra1-1	Frankfurt/Germany
gru1-2	Sao Paulo/Brazil
icn1-1	Seoul/South Korea
jnb1-1	Johannesburg/South Africa
lhr1-1	London/United Kingdom
mad1-1	Madrid/Spain
mia1-1	Miami/United States
nrt1-2	Tokyo/Japan
nrt1-3	Tokyo/Japan
nrt1-4	Tokyo/Japan
nrt1-5	Tokyo/Japan
nrt1-6	Tokyo/Japan
nrt1-7	Tokyo/Japan
nrt1-8	Tokyo/Japan
ord1-1	Chicago/United States
sea1-1	Seattle/United States
sin1-1	Singapore/Singapore
sin1-2	Singapore/Singapore
sjc1-1	San Jose/United States
syd1-1	Sydney/Australia
waw1-1	Warsaw/Poland
yyz1-1	Toronto/Canada

APPENDIX B
ANYCAST CATCHMENT MAPS FOR IPV4 & IPV6

APPENDIX C
TOP TALKERS - NORTH AMERICA & EUROPE

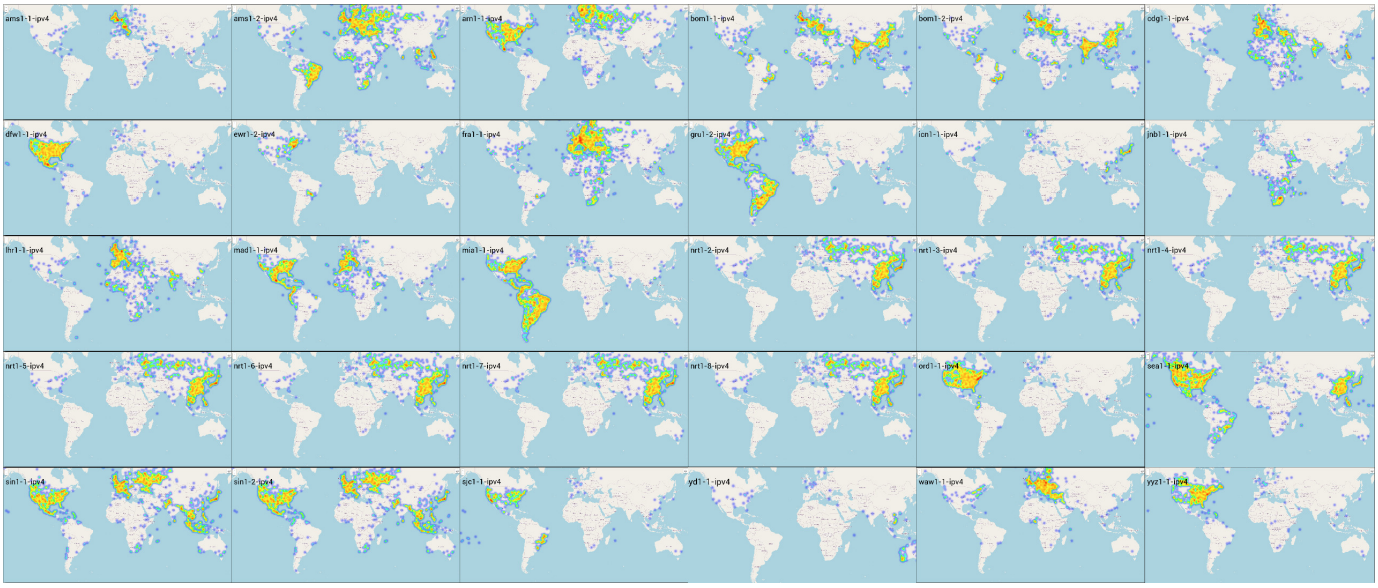

Fig. 13: Catchment Maps of Anycast Sites (IPv4)

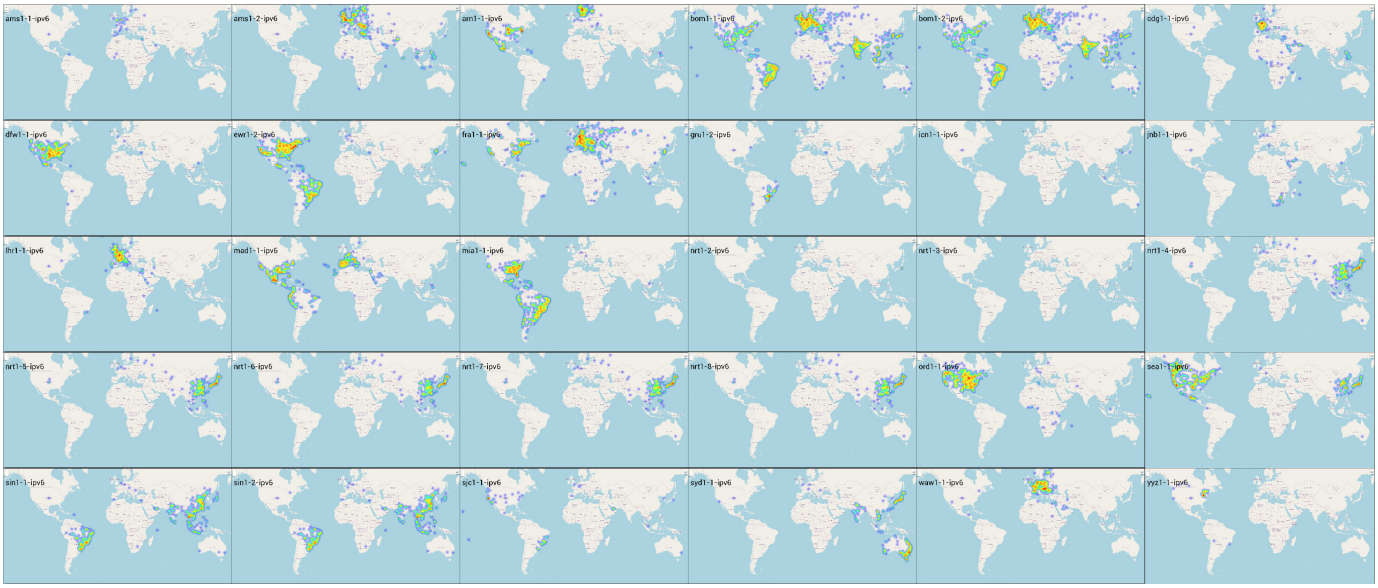

Fig. 14: Catchment Maps of Anycast Sites (IPv6)

Client IP	Queries	Time frame	Queries/Second	Responses	% Responses / Queries	Country
55.14.123.0	1,197,811	87791	13.64	0	0	CA
55.247.135.171	841,593	87634	9.6	0	0	CA
207.109.20.23	732,750	61277	11.95	0	0	US
57.224.52.182	525,796	87773	5.99	0	0	US
220.174.213.224	474,156	86104	5.5	0	0	US
207.174.139.246	450,914	87801	5.13	0	0	US
222.176.245.78	342,739	87634	3.91	0	0	US
22.181.254.85	334,351	87662	3.81	0	0	US
63.94.254.66	330,833	87627	3.77	0	0	US
50.86.7.46	322,532	83927	3.84	0	0	CA

TABLE XIII: Number of Queries sent by the Top 10 IPs in North America (Anonymized)

Client IP	Queries	Time frame	Queries/Second	Responses	% Responses / Queries	Country
212.149.201.123	2,506,744	87730	28.57	6569	0.26	ES
190.88.34.241	1,158,432	86150	13.44	888	0.076	NL
37.152.232.162	939,874	76362	12.3	230	0.024	DE
37.206.76.222	699,672	87795	7.96	983	0.14	NL
107.101.144.27	648,687	85449	7.59	146	0.022	DE
36.70.118.251	530,172	87836	6.03	4035	0.76	NL
45.56.178.242	452,885	83440	5.42	503	0.11	FR
39.189.129.73	442,865	87816	5.04	3510	0.79	SE
73.144.37.25	415,040	85193	4.87	1545	0.37	SE
36.28.72.143	381,689	87837	4.34	4999	1.3	ES

TABLE XIV: Number of Queries sent by the Top 10 IPs in Europe (Anonymized)