Overlooked and Neglected

How NTP services and clients leave billions of devices exposed to unnecessary risk

---

Giovane C. M. Moura

SIDN Labs and TU Delft

SURFnetworking 2025
Utrecht, The Netherlands
2025-12-10

Summary of two tech reports

Are NTP clients always right?

Evaluating NTP clients under normal and attack scenarios

Technical Report SIDN Labs 2025-10-16

Update: 2025-10-30

Shreyas Konjerla
TU Delft
Delft, The Netherlands
s.konjerla@student.tudelft.nl

Giovane C. M. Moura
SIDN Labs and TU Delft
Arnhem and Delft, The Netherlands
giovane.moura@tudelft.nl

Georgios Smaragdakis
TU Delft
Delft, The Netherlands
G.Smaragdakis@tudelft.nl

Tamme Dittrich
Tweede Golf
Nijmegen, The Netherlands
tamme@tweedegolf.com

Credit goes to all authors involved in both reports

Summary of two tech reports

BigTime: Characterizing Large Time Service Providers

Technical Report SIDN Labs 2025-12-01

Pascal Huppert
University of Twente and
University of Münster
The Netherlands / Germany
pascal.huppert@utwente.nl

Giovane C. M. Moura
SIDN Labs and TU Delft
The Netherlands
giovane.moura@sidn.nl

Marco Davids
SIDN Labs
The Netherlands
marco.davids@sidn.nl

Pieter-Tjerk de Boer
University of Twente
The Netherlands
p.t.deboer@utwente.nl

Ralph Holz
University of Münster and
University of Twente
Germany / The Netherlands
ralph.holz@uni-muenster.de

Rein Fernhout
University of Twente
The Netherlands
r.p.j.fernhout@student.utwente.nl

Georgios Smaragdakis
TU Delft
The Netherlands
g.smaragdakis@tudelft.nl

Cristian Hesselman
SIDN Labs and University of
Twente
The Netherlands
cristian.hesselman@sidn.nl

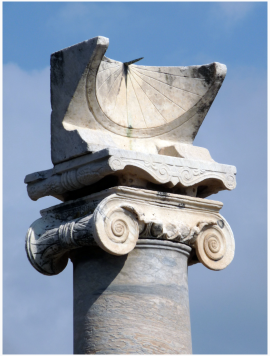
Credit goes to all authors involved in both reports

Outline

- Introduction
- Clients
  - Attacking Clients
- NTP Providers
  - Replication
  - Client/Server mapping
  - Accuracy

Timekeeping over time

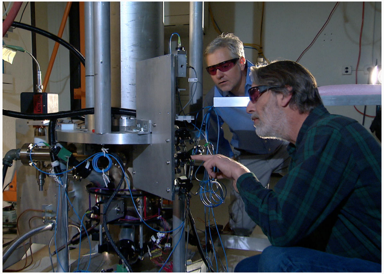
Ancient Roman Sundial
Pompeii, 70 AD



Churches with pendulum clocks
Middle ages to now

Timekeeping nowadays

Atomic “clocks” : oscillators



US NIST-F2

Precision: 1s in 300M years

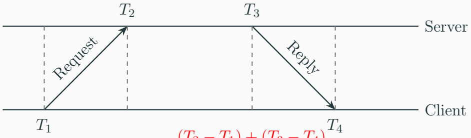
- Reference sources (GNSS, Atomic, Radio (DC77) etc) produce time info
- Distributed over the Internet
  - **Using the Network Time Protocol (NTP)**
- (SIDN provides free time service at `https://time.nl`)

Why clock synchronization matters?

- It underpins modern life:
  - Phone sync
  - Computers sync
  - Utility bills
  - Trains on time
- On the Internet:
  - TLS
  - DNSSEC
  - DNS caches
  - RPKI
- USNO 2012 time incident caused outages in multiple places

The NTP protocol

- NTP is the default protocol for clock sync
- Clients queries NTP servers
  - which responde with correct time
- Without NTP, client's clock would drift
- Standards: NTPv4, SNTP, NTS
  - New: NTPv5 (in dev)

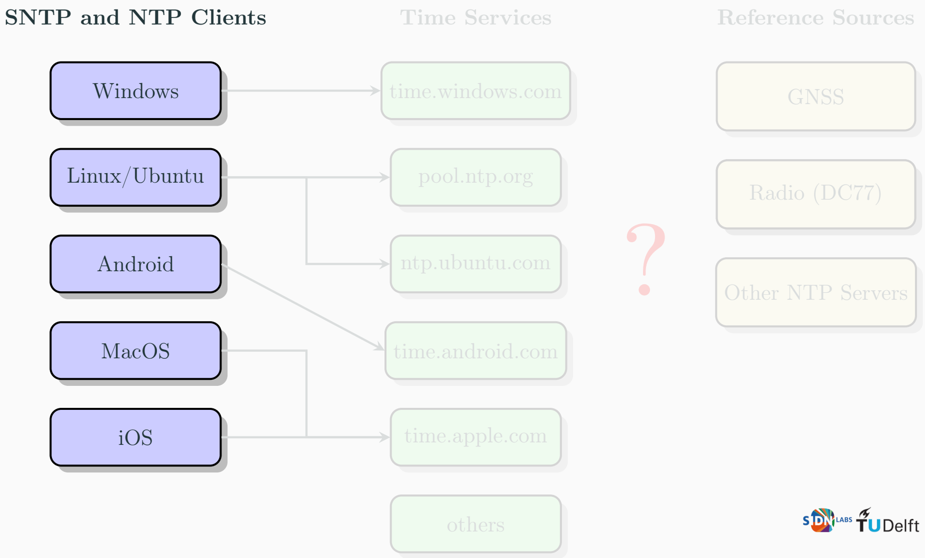


Offset: $\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$

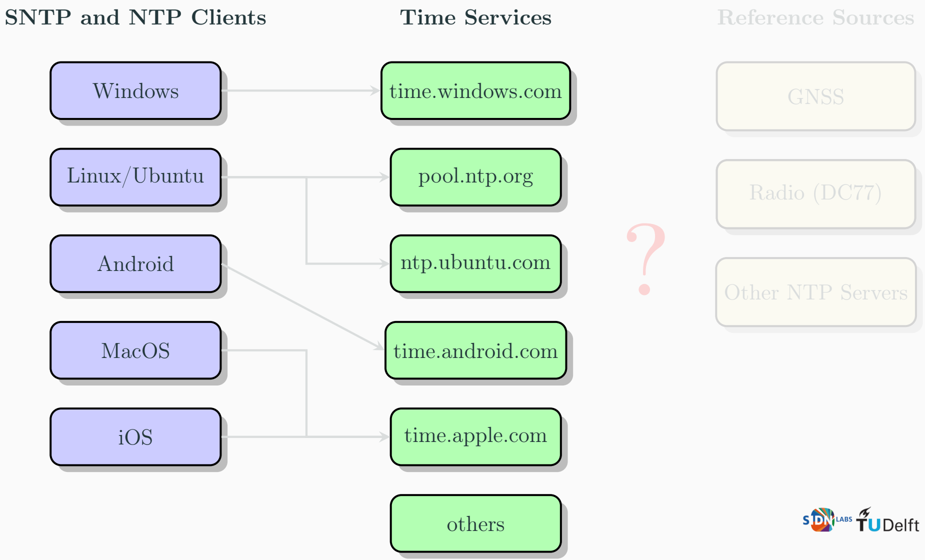
RTT: $\delta = (T_4 - T_1) - (T_3 - T_2)$

Figure 1: Timestamps used in NTP offset calculations

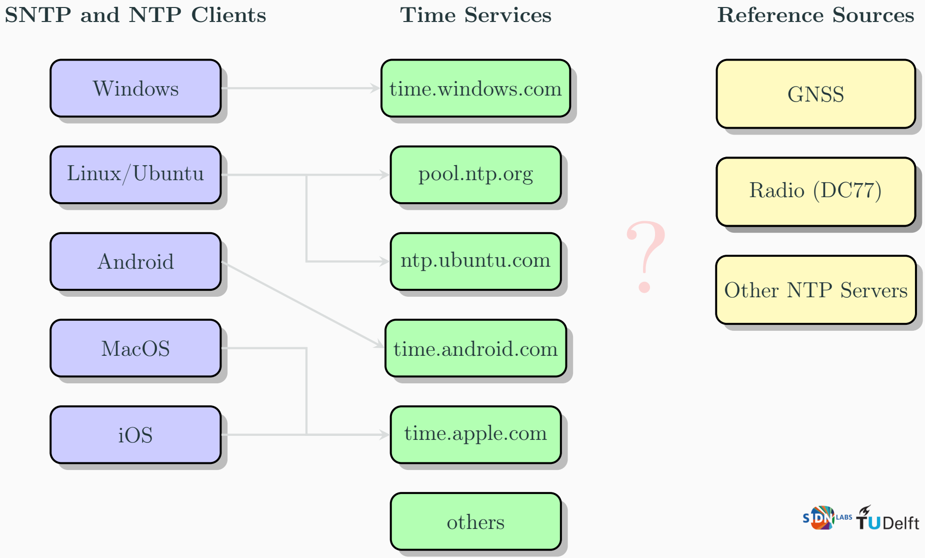
NTP Ecosystem



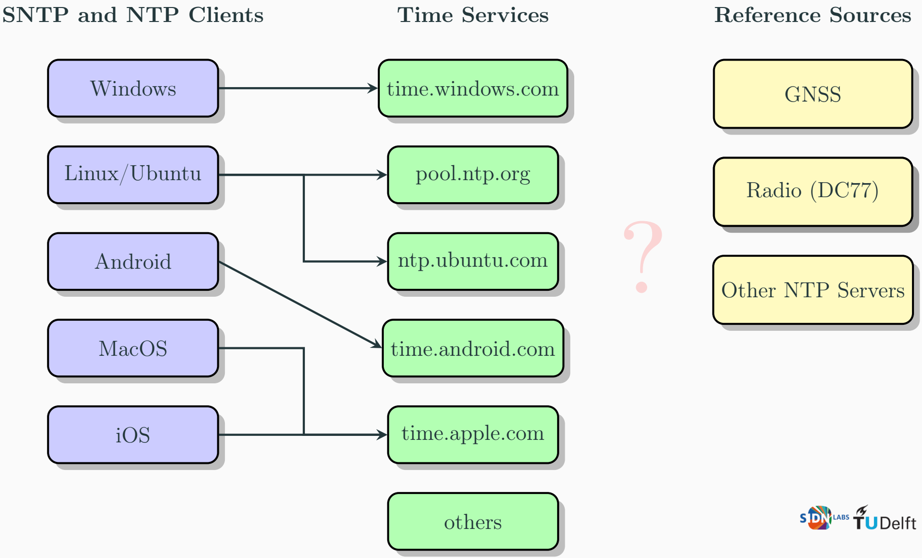
NTP Ecosystem



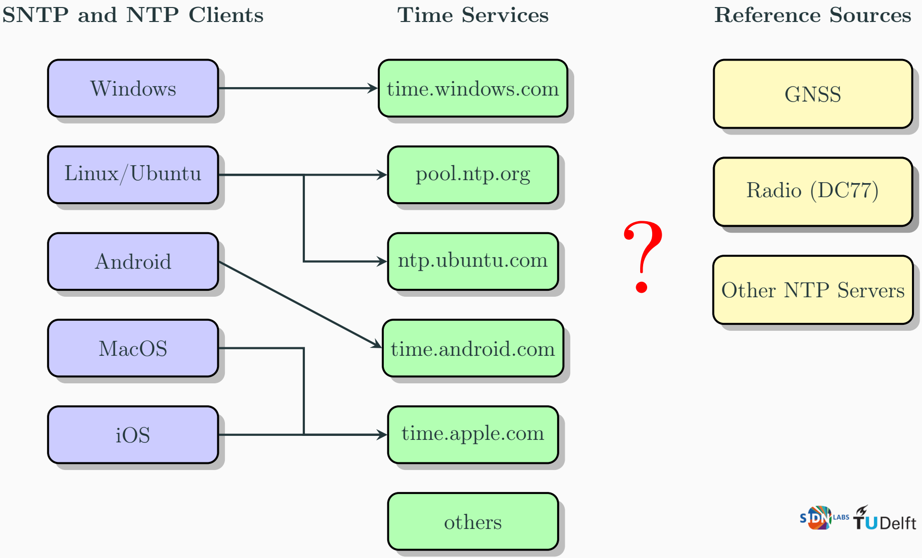
NTP Ecosystem



NTP Ecosystem



NTP Ecosystem



Outline

Introduction

Clients

- Attacking Clients

NTP Providers

- Replication
- Client/Server mapping
- Accuracy

NTP Clients

Are NTP clients always right?

Evaluating NTP clients under normal and attack scenarios

Technical Report SIDN Labs 2025-10-16

Update: 2025-10-30

Shreyas Konjerla
TU Delft
Delft, The Netherlands
s.konjerla@student.tudelft.nl

Giovane C. M. Moura
SIDN Labs and TU Delft
Arnhem and Delft, The Netherlands
giovane.moura@tudelft.nl

Georgios Smaragdakis
TU Delft
Delft, The Netherlands
G.Smaragdakis@tudelft.nl

Tamme Dittrich
Tweede Golf
Nijmegen, The Netherlands
tamme@tweedegolf.com

NTP Clients

Client (version)	User base	OS	Release
macOS (15.4.1)	2.2B*	macOS	Sept 2024
W32Time	1.4B	Windows	Apr 2025
timesyncd (255.4)	–	Ubuntu	Sept 2024
NTPSec (1.2.2)	–	Ubuntu	Nov 2024
NTPD-RS (1.1.2)	–	Ubuntu	Jan 2025
ntpd (4.2.8p18)	–	Ubuntu	Jun 2023
OpenNTPd (1:6.2p3-4.2)	–	Ubuntu	Oct 2022
Chrony (4.5)	–	Ubuntu	Oct 2024

Table 1: Clients highlighted are OS defaults and SNTP-based. *Includes macOS and iOS devices.

Default clients matter more! Few change NTP settings.

NTP Clients

Client (version)	User base	OS	Release
`macOS` (15.4.1)	2.2B*	macOS	Sept 2024
`W32Time`	1.4B	Windows	Apr 2025
`timesyncd` (255.4)	–	Ubuntu	Sept 2024
`NTPSec` (1.2.2)	–	Ubuntu	Nov 2024
`NTPD-RS` (1.1.2)	–	Ubuntu	Jan 2025
`ntpd` (4.2.8p18)	–	Ubuntu	Jun 2023
`OpenNTPd` (1:6.2p3-4.2)	–	Ubuntu	Oct 2022
`Chrony` (4.5)	–	Ubuntu	Oct 2024

Table 1: Clients highlighted are OS defaults and SNTP-based. *Includes macOS and iOS devices.

Default clients matter more! Few change NTP settings.

What did we do

1. Evaluated clients under normal operations
2. And under attack (time-shift attacks)

Setup: we configure the clients to query 3 servers we run ourselves

Clients under normal operations

		Queries Per Server		
Client	**Avg/h**	**S1**	**S2**	**S3**
`Chrony`	80.37	1,643	1,498	1,681
`macOS`	5.15	102	101	106
`ntpd`	12.32	246	246	246
`NTPD-RS`	250.8	5,012	5,012	5,026
`OpenNTPd`	22.68	425	473	463
`NTPSec`	16.82	336	336	337
`W32Time`	3.55	80	77	56
`timesyncd`	1.77	108	0	0

Table 2: Queries sent per hour per client, broken down by server. Highlighted are SNTP clients.

Clients under normal operations

Findings:

1. Default clients are all SNTP
   - not supposed to be used like this
2. Large variation in traffic
3. `timesyncd`, used in Ubuntu, Flatcar, and many Linux Server distros, use a single time source
   - violates RFC8633

		Queries Per Server		
Client	**Avg/h**	**S1**	**S2**	**S3**
`Chrony`	80.37	1,643	1,498	1,681
`macOS`	5.15	102	101	106
`ntpd`	12.32	246	246	246
`NTPD-RS`	250.8	5,012	5,012	5,026
`OpenNTPd`	22.68	425	473	463
`NTPSec`	16.82	336	336	337
`W32Time`	3.55	80	77	56
`timesyncd`	1.77	108	0	0

Outline

Introduction

Clients

Attacking Clients

NTP Providers

Replication

Client/Server mapping

Accuracy

Attacking Clients

- We now do new experiments where we do **time-shift attacks**
    - we lie about the time
    - goal: mess up with client's clock
    - most dangerous type of attack

Attacking Clients: results

	Offset					
Client	**900s**	**1M**	**2M**	**3M**	**1Y**	**2Y**
macOS	✓	✓	✓	✓	✓	✓
W32Time	✓					
timesyncd	✓	✓	✓	✓	✓	✓
NTPSec						
NTPD-RS						
ntpd						
OpenNTPd						
Chrony						

Table 3: Client behavior to time shift attacks. (✓) shows vulnerable clients. (M = month, Y = Year).

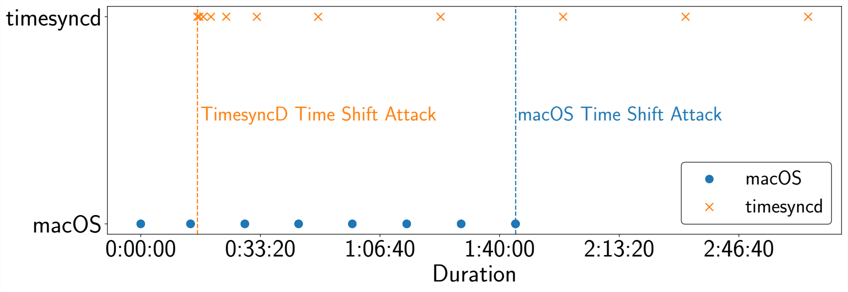
Attacking Clients: results

- All OS Default clients are vulnerable!
  - MS caps time shifts at two weeks
- None of the NTP clients are vulnerable
  - only SNTP clients

	Offset					
Client	**900s**	**1M**	**2M**	**3M**	**1Y**	**2Y**
`macOS`	✓	✓	✓	✓	✓	✓
`W32Time`	✓					
`timesyncd`	✓	✓	✓	✓	✓	✓
`NTPSec`						
`NTPD-RS`						
`ntpd`						
`OpenNTPd`						
`Chrony`						

Vulnerability to time-shift attacks (✓ = vulnerable).

How long does it take for the atack to succeed



(a) timesyncd and macOS (2 years offset)

Figure 2: Time-series of NTP queries for systemd-timesyncd and macOS. Dashed lines show when the attacks succeed.

Recommendations

1. For operators: do NOT use default clients (no `timesyncd`)
    - For SURF folks here, check your Linux VMS: `timesyncd` has got to go
2. For vendors: do what Ubuntu did, stop using SNTP clients for these OSes
3. Use NTS servers to prevent man-in-the-middle attacks

Extra: we found 10 bugs in software and notified vendors. See report

Outline

Introduction

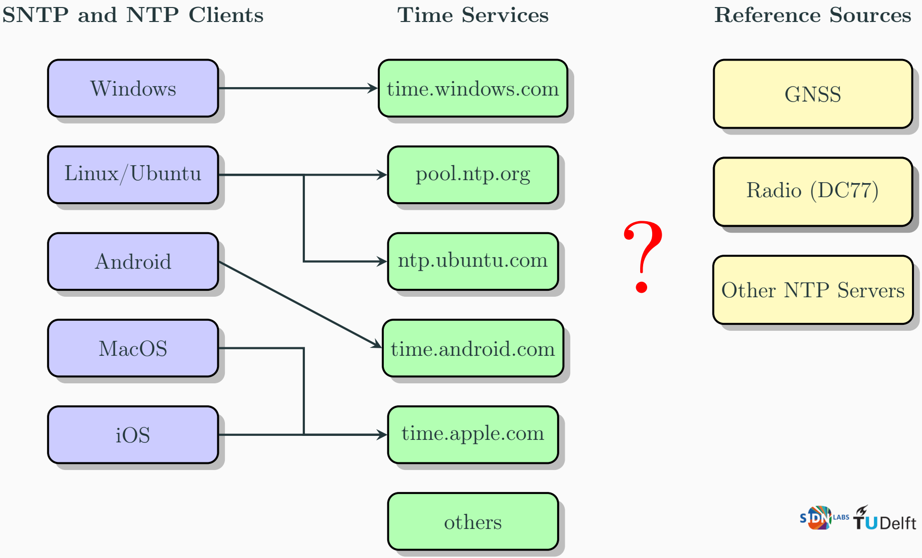
Clients

- Attacking Clients

NTP Providers

- Replication
- Client/Server mapping
- Accuracy

NTP Ecosystem



Time services providers

BigTime: Characterizing Large Time Service Providers

Technical Report SIDN Labs 2025-12-01

Pascal Huppert
University of Twente and
University of Münster
The Netherlands / Germany
pascal.huppert@utwente.nl

Giovane C. M. Moura
SIDN Labs and TU Delft
The Netherlands
giovane.moura@sidn.nl

Marco Davids
SIDN Labs
The Netherlands
marco.davids@sidn.nl

Pieter-Tjerk de Boer
University of Twente
The Netherlands
p.t.deboer@utwente.nl

Ralph Holz
University of Münster and
University of Twente
Germany / The Netherlands
ralph.holz@uni-muenster.de

Rein Fernhout
University of Twente
The Netherlands
r.p.j.fernhout@student.utwente.nl

Georgios Smaragdakis
TU Delft
The Netherlands
g.smaragdakis@tudelft.nl

Cristian Hesselman
SIDN Labs and University of
Twente
The Netherlands
cristian.hesselman@sidn.nl

Time services providers

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

Table 4: Evaluated time service providers. Providers highlighted are enabled by default in their respective operating systems (OSes).

SIDN LABS TU Delft UNIVERSITY OF TWENTE. tweede golf

What do we evaluate?

1. Their replication architecture
   - They have to serve BILLIONS of daily clients
2. Their client/server mapping
3. Their accuracy

Outline

Introduction

Clients

- Attacking Clients

NTP Providers

- Replication
- Client/Server mapping
- Accuracy

Replication: the Root DNS system

Example of DNS replication

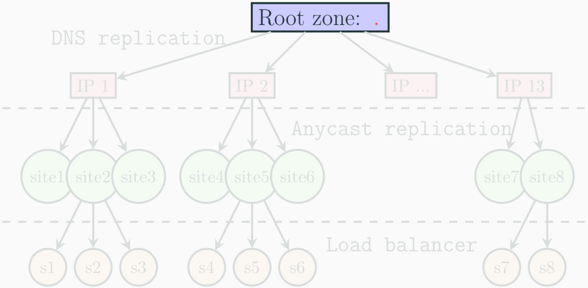


Figure 3: Service replication in the Root DNS system.

Replication: the Root DNS system

Example of DNS replication

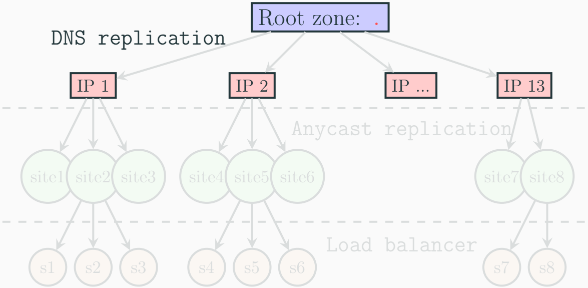


Figure 3: Service replication in the Root DNS system.

Replication: the Root DNS system

Example of DNS replication

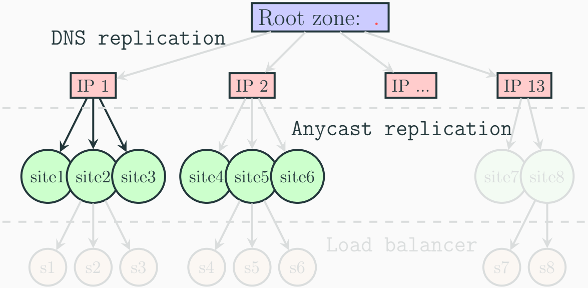


Figure 3: Service replication in the Root DNS system.

Replication: the Root DNS system

Example of DNS replication

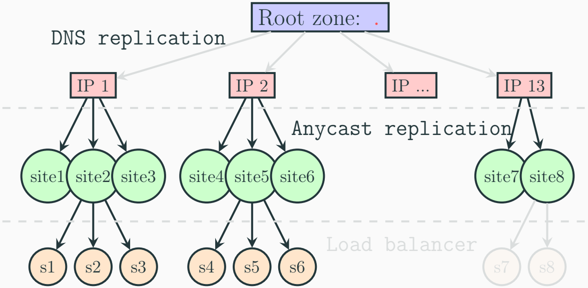


Figure 3: Service replication in the Root DNS system.

Replication: the Root DNS system

Example of DNS replication

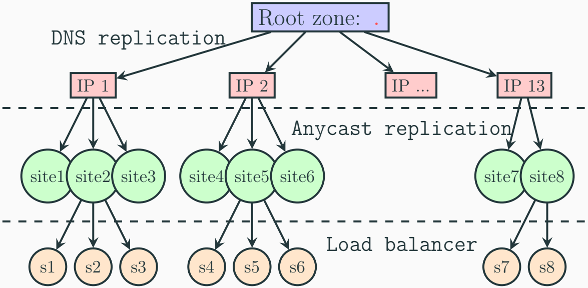


Figure 3: Service replication in the Root DNS system.

Time providers: DNS replication

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

Table 5: DNS-level replication of time service providers.

Time providers: Anycast replication

Provider	Prefixes	Anycast	Sites (IPv4)	Sites (IPv6)
Microsoft	11/0	No	–	–
Apple	53/48	No	–	–
Google	1/1	Yes	41	No data
Ubuntu	4/3	No	–	–
AWS	46/9	No	–	–
Cloudflare	1/1	Yes	63	47
Meta	5/5	Yes	8	11

Table 6: Anycast replication of time service providers.

We found no evidence of local load balancers

Time providers: Anycast replication

Provider	Prefixes	Anycast	Sites (IPv4)	Sites (IPv6)
Microsoft	11/0	No	–	–
Apple	53/48	No	–	–
Google	1/1	Yes	41	No data
Ubuntu	4/3	No	–	–
AWS	46/9	No	–	–
Cloudflare	1/1	Yes	63	47
Meta	5/5	Yes	8	11

Table 6: Anycast replication of time service providers.

We found no evidence of local load balancers

Server replication for time providers

Provider	DNS Replication?	Anycast?	Load balancer?
Microsoft	Yes	No	–
Apple	Yes	No	No*
Google	Yes	Yes	–
Ubuntu	Yes	No	No*
Cloudflare	Yes	Yes	–
Meta	Yes	Yes	–
AWS	Yes	No	–

Table 7: Server replication for time providers

Unicast NTP servers geolocation

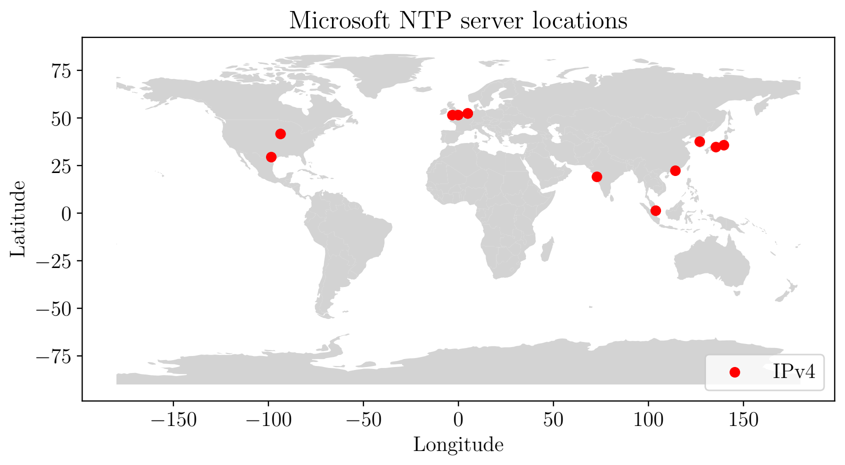


Figure 4: Microsoft.

Unicast NTP servers geolocation

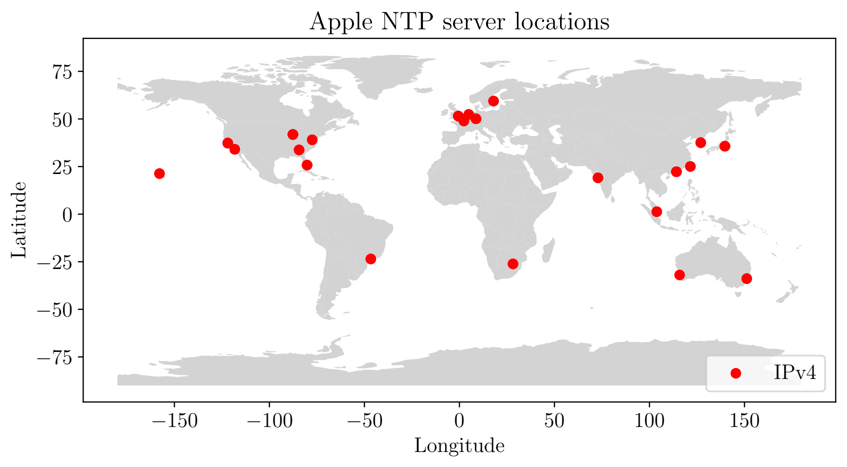


Figure 5: Apple.

Unicast NTP servers geolocation

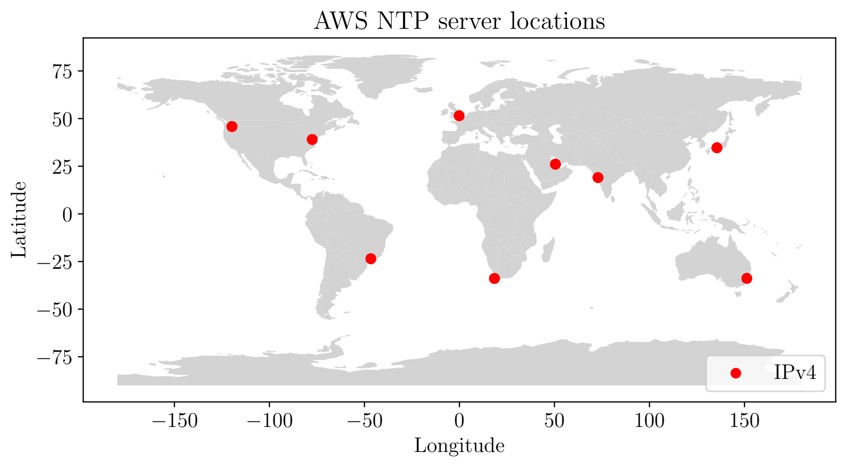


Figure 6: AWS.

Outline

Introduction

Clients

- Attacking Clients

NTP Providers

- Replication
- Client/Server mapping
- Accuracy

Server mapping mechanisms

- **Anycast:** BGP maps clients to nearest site
- **Unicast:** All providers use geolocation for mapping

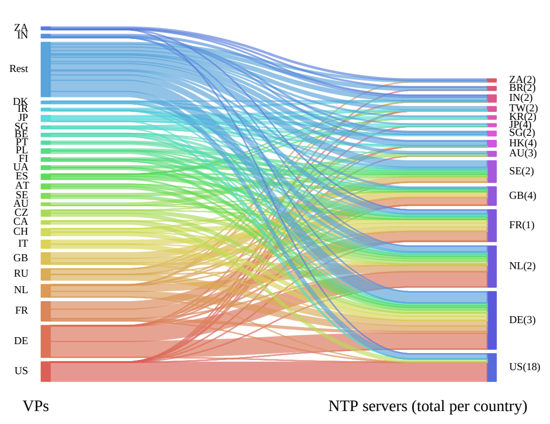
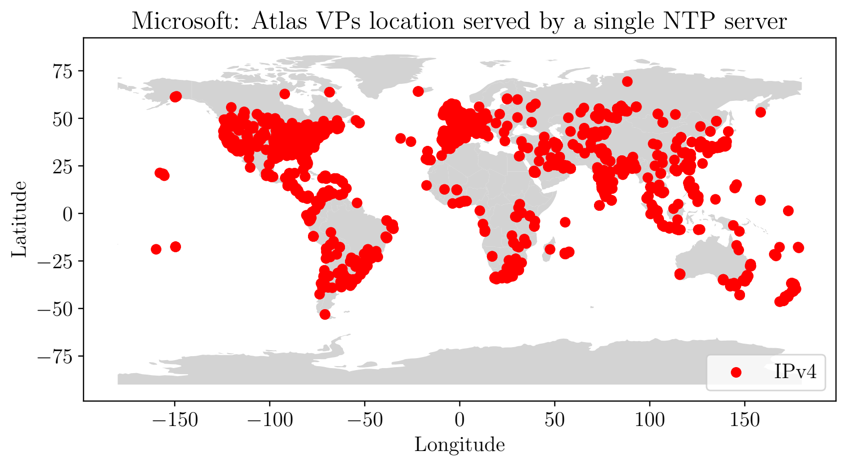


Figure 7: Apple geo-based mappings

Problem: Microsoft: 50% of VPs receive 1 IP address



That's a violation of RFC8633 – more than 1 time source

Outline

Introduction

Clients

- Attacking Clients

NTP Providers

- Replication
- Client/Server mapping
- Accuracy

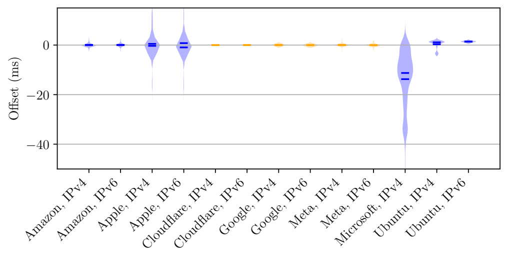
Accuracy

- NTP services MUST be accurate
- We carry out experiments from two VPs with “ground truth”

VP	ASN	Time Source	Method
SE-AWS	16509	GNSS, atomic	PHC
NL-SIDN	1140	GNSS, radio, atomic	Linux PTP

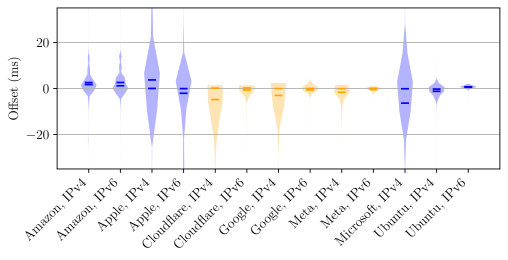
Table 8: Accuracy Experiment Vantage Points

Offset distributions (violin plots)



SE-AWS

Positive offset means server clock is ahead of reference.



NL–SIDN

Positive offset means server clock is ahead of reference.

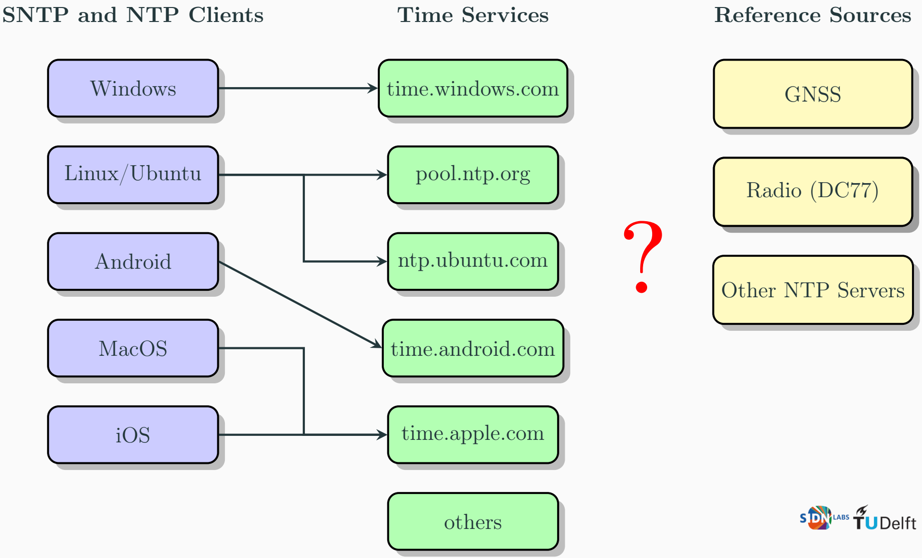
Time Services Measurements

Provider	IPv	#IP Addr.	RPKI	DNSSEC	RTT mean	NTP version	NTS	Stratum
Amazon	v4	89	Yes	No	162.05 ms	v4	No	4
	v6	89	Yes	No	162.08 ms	v4	No	4
Apple	v4	51	No	No	166.54 ms	v4	No	1/2 (78%/22%)
	v6	46	No	No	168.75 ms	v4	No	1/2 (77%/23%)
Cloudflare	v4	2	Yes	Yes	3.97 ms	v4	Yes	3
	v6	2	Yes	Yes	3.95 ms	v4	Yes	3
Google	v4	4	Yes	No	11.26 ms	v4	No	1
	v6	4	Yes	No	11.12 ms	v4	No	1
Meta	v4	5	Yes	No	23.02 ms	v4	No	1
	v6	5	Yes	No	36.24 ms	v4	No	1
Microsoft	v4	12	Yes	No	145.65 ms	v3	No	3
	v6	0	–	–	–	–	–	–
Ubuntu	v4	4	No	No	47.45 ms	v4	Yes	2
	v6	3	No	No	31.01 ms	v4	Yes	2

Recommendations

- Providers should support NTS services and clients
  - no reasons to expose clients this way
  - Ubuntu already did it (kudos!)
- Extra features: RPKI, DNSSEC
- Microsoft has lot of work to do:
  - fix their clocks: they clocks are sometimes out-of-sync
  - adhere to RFC8633 and servers clients with more than 1 server
  - Deploy NTPv4 and NTS
- For ops: do not use Microsoft NTP client or time service
- Why many Linux distros still use `timesyncd`?

NTP Ecosystem



Summary

- We've looked both clients and services, and found issues at both sides
- We show that time services despite billions of clients, can have not ideal offsets
- We hope other vendors follow Ubuntu and enable NTS for all its clients

Clients report:

Time services report: