Massive Improvements in the Accuracy of Computer Clocks

Kurt Rosenfeld
kurt.harlem@gmail.com

January 22nd, 2024

In the past 30 years, computer clock accuracy commonly seen in datacenters improved from 1ms to 100ns because:

1. GPS-based time servers
2. IEEE 1588 Precision Time Protocol (PTP)
3. more stable low-cost oscillators

Who needs an accurate clock?

1. Scientific data acquisition, e.g., physics
2. Distributed data storage systems
3. High frequency trading

Computer clocks: What do they do for us?

- get
- alarm
- sleep
- is-before
- is-after
- when did the packet arrive
- when was the packet transmitted

Computer clocks: How we access them

- I/O
- CPU internal
- RPC
- network receive and transmit

The UNIX system clock (on x86)

$$time = a \cdot TSC + b \tag{1}$$

time is the number of seconds since the beginning of the year 1970.

What about SMP?

What is clock accuracy?

1. Phase agreement between two peer computers.
2. Rate agreement between two peer computers.
3. Phase agreement between a computer and a global reference.
4. Rate agreement between a computer and a global reference.

Why did clock accuracy improve?

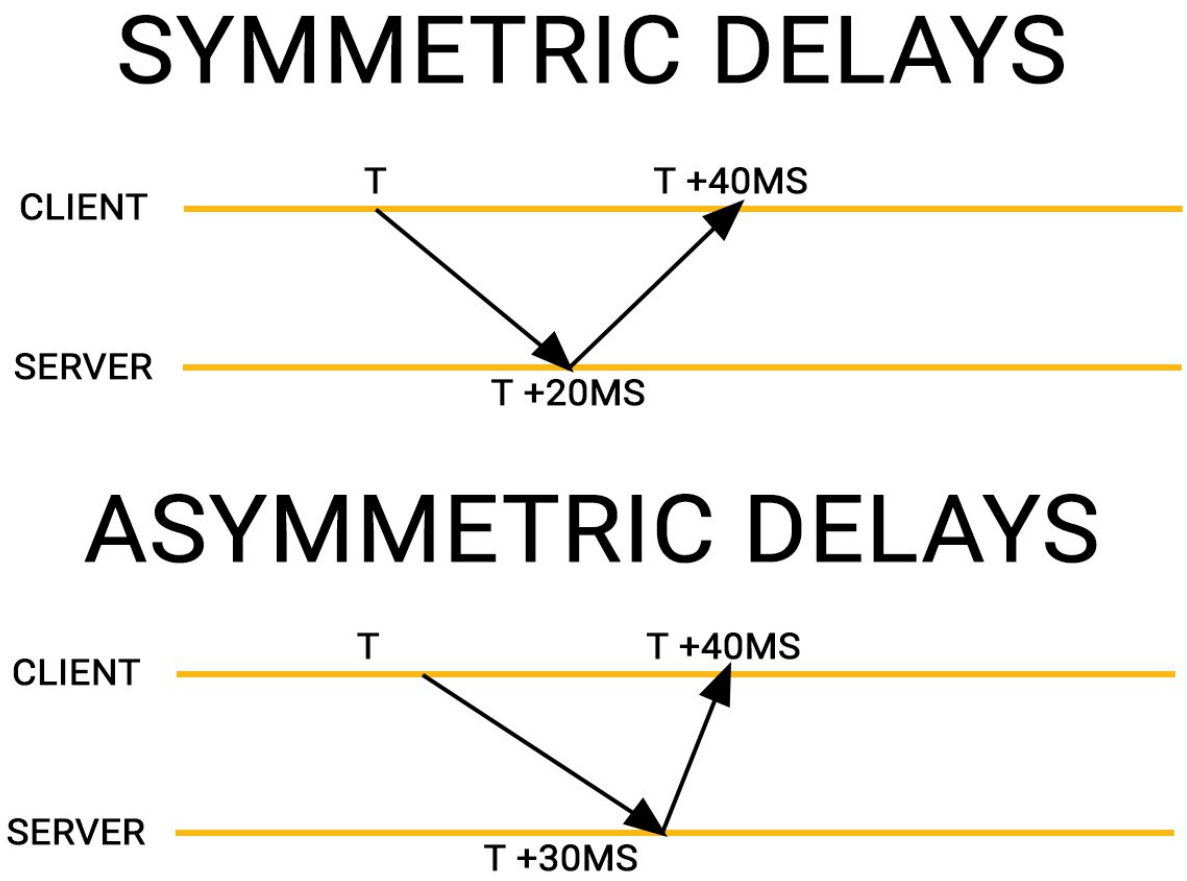
1. Use-cases demanded more accurate clocks.
2. Time engineers developed better clock sync technology.
3. Hardware vendors added support for the new technology.
4. New datacenters deployed the new technology.

Thirty years ago

Computers used NTP to sync their clocks with NIST via the Internet. From the official NTP webpage:

> *If left running continuously, an NTP client on a fast LAN in a home or office environment can maintain synchronization nominally within one millisecond.*

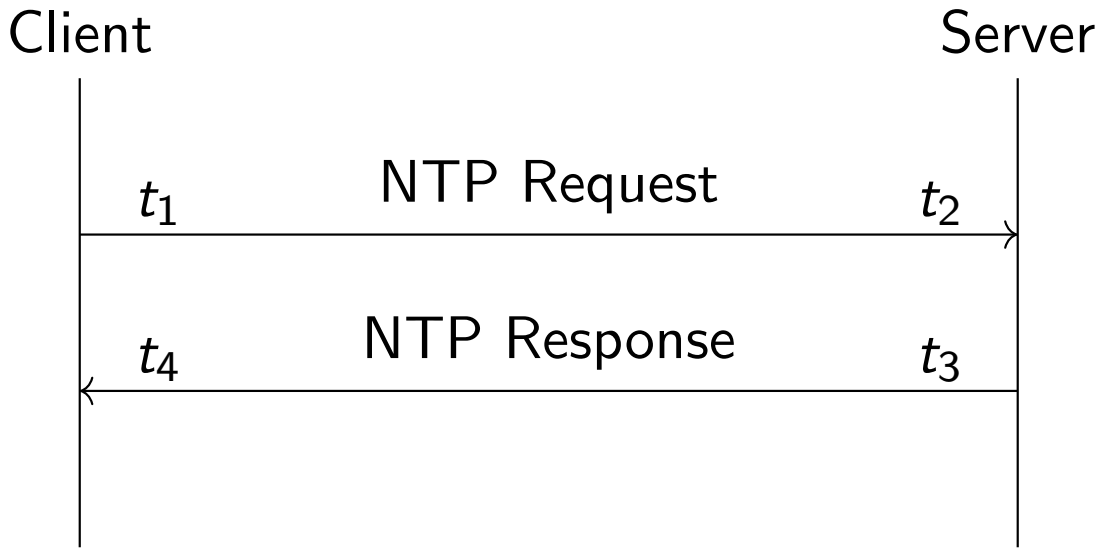
How NTP adds uncertainty during time transfer



$$\texttt{client\_uncertainty} = 40ms + \texttt{server\_uncertainty}$$

figure from RIPE Labs

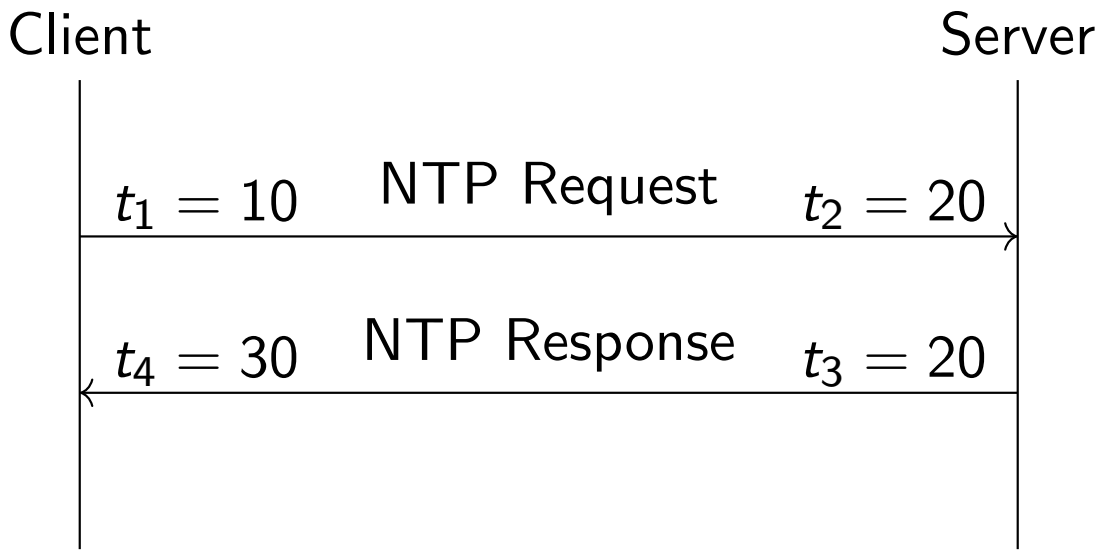
NTP internal timestamps



$$t_2 - t_1 = delay_{CS} + offset \quad (2)$$

$$t_4 - t_3 = delay_{SC} - offset \quad (3)$$

NTP internal timestamps



$$t_2 - t_1 = delay_{CS} + offset \quad (4)$$

$$t_4 - t_3 = delay_{SC} - offset \quad (5)$$

Solve for $offset$, $delay_{CS}$, and $delay_{SC}$.

- $offset = 0$, $delay_{CS} = 10$, and $delay_{SC} = 10$
- $offset = 10$, $delay_{CS} = 0$, and $delay_{SC} = 20$

Latency asymmetry on the Internet

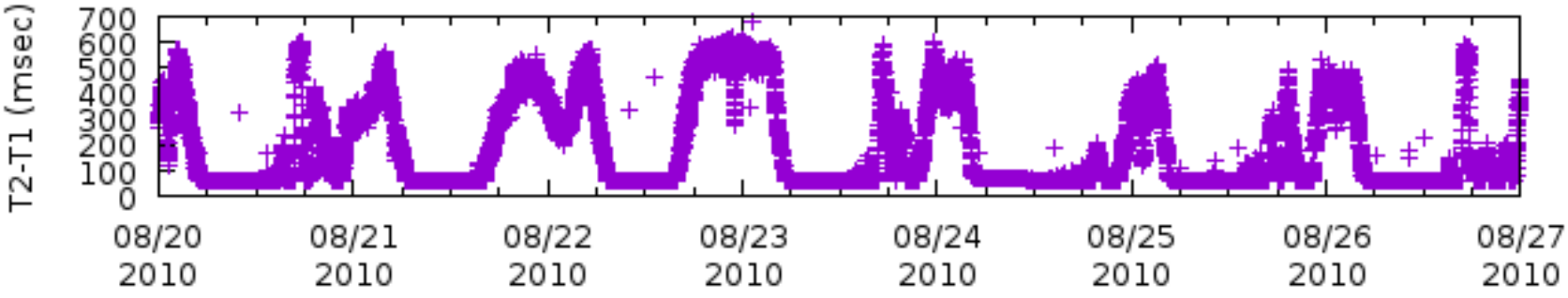


Figure 6 T2-T1 delay between Chicago client and Hawaii stratum 1. (MJD 55428-55434)

From *Challenges in Time Transfer Using the Network Time Protocol (NTP)*, Steven E. Sommars, Nokia, PTTI 2017

Typical Internet NTP service: pool.ntp.org

server address	RTT	one-way hop count
96.233.62.59	38 ms	14
45.63.54.13	87 ms	14
108.61.73.243	29 ms	11
104.167.241.197	52 ms	19

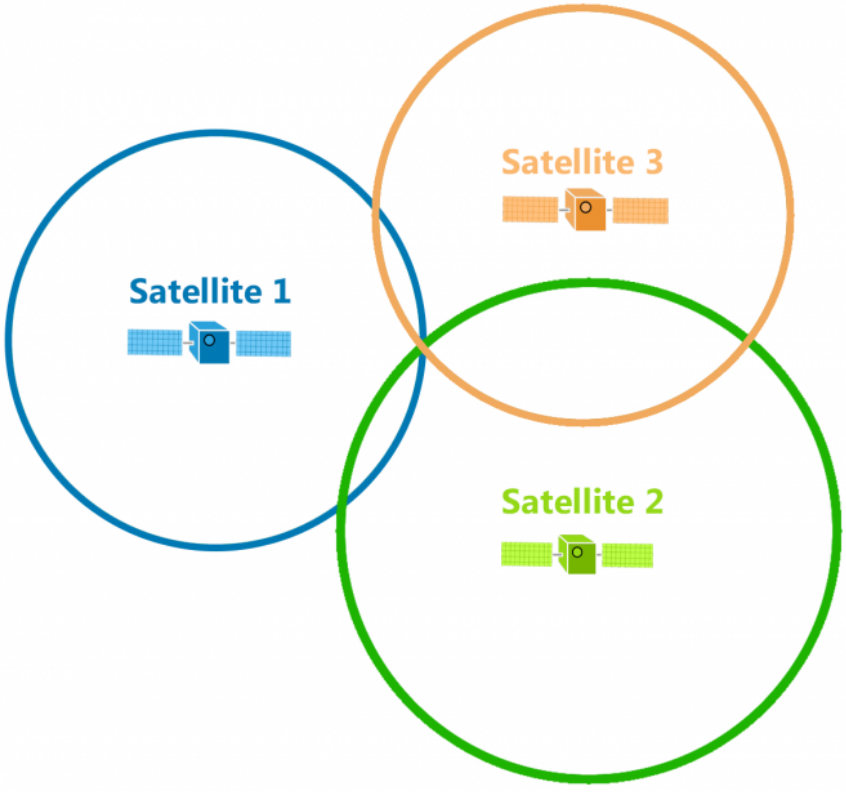
GPS as a time reference

A GPS fix gives us { x, y, z, t }.

From gps.gov:

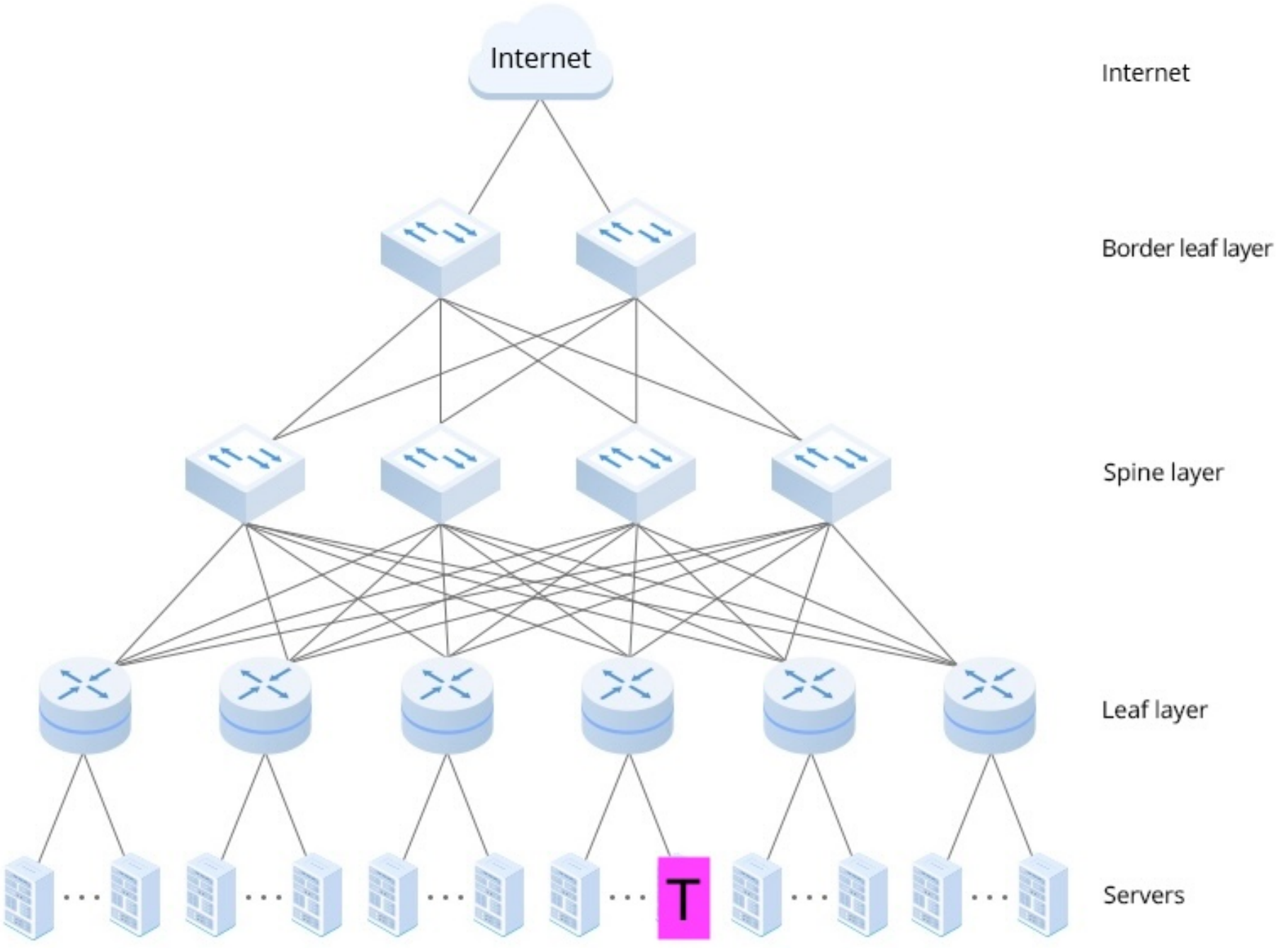
> *The government distributes UTC as maintained by the U.S. Naval Observatory (USNO) via the GPS signal in space with a time transfer accuracy relative to UTC(USNO) of $\leq$ 30 nanoseconds (billionths of a second), 95% of the time.*

Where are we and what time is it?



$$
\begin{aligned}
(x - A_1)^2 + (y - B_1)^2 + (z - C_1)^2 - (c(t_1 - d))^2 &= 0 \\
(x - A_2)^2 + (y - B_2)^2 + (z - C_2)^2 - (c(t_2 - d))^2 &= 0 \\
(x - A_3)^2 + (y - B_3)^2 + (z - C_3)^2 - (c(t_3 - d))^2 &= 0 \\
(x - A_4)^2 + (y - B_4)^2 + (z - C_4)^2 - (c(t_4 - d))^2 &= 0
\end{aligned}
$$

Typical data center network



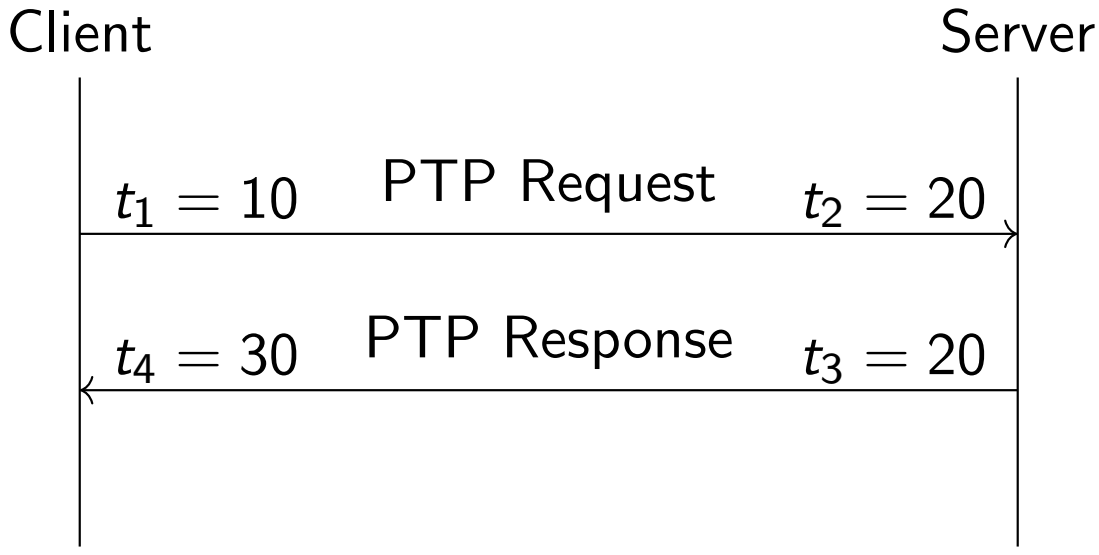
IEEE 1588: Precision Time Protocol (PTP)

Goal: Transfer time with minimal added uncertainty.

Approach taken:

- Account for link-level propagation delay.
- Account for queuing delay in store-and-forward network nodes.
- Stipulate symmetric latency at each hop.
- Use hardware timestamping as much as possible to avoid random latency in kernel and userspace.

PTP as idealized NTP



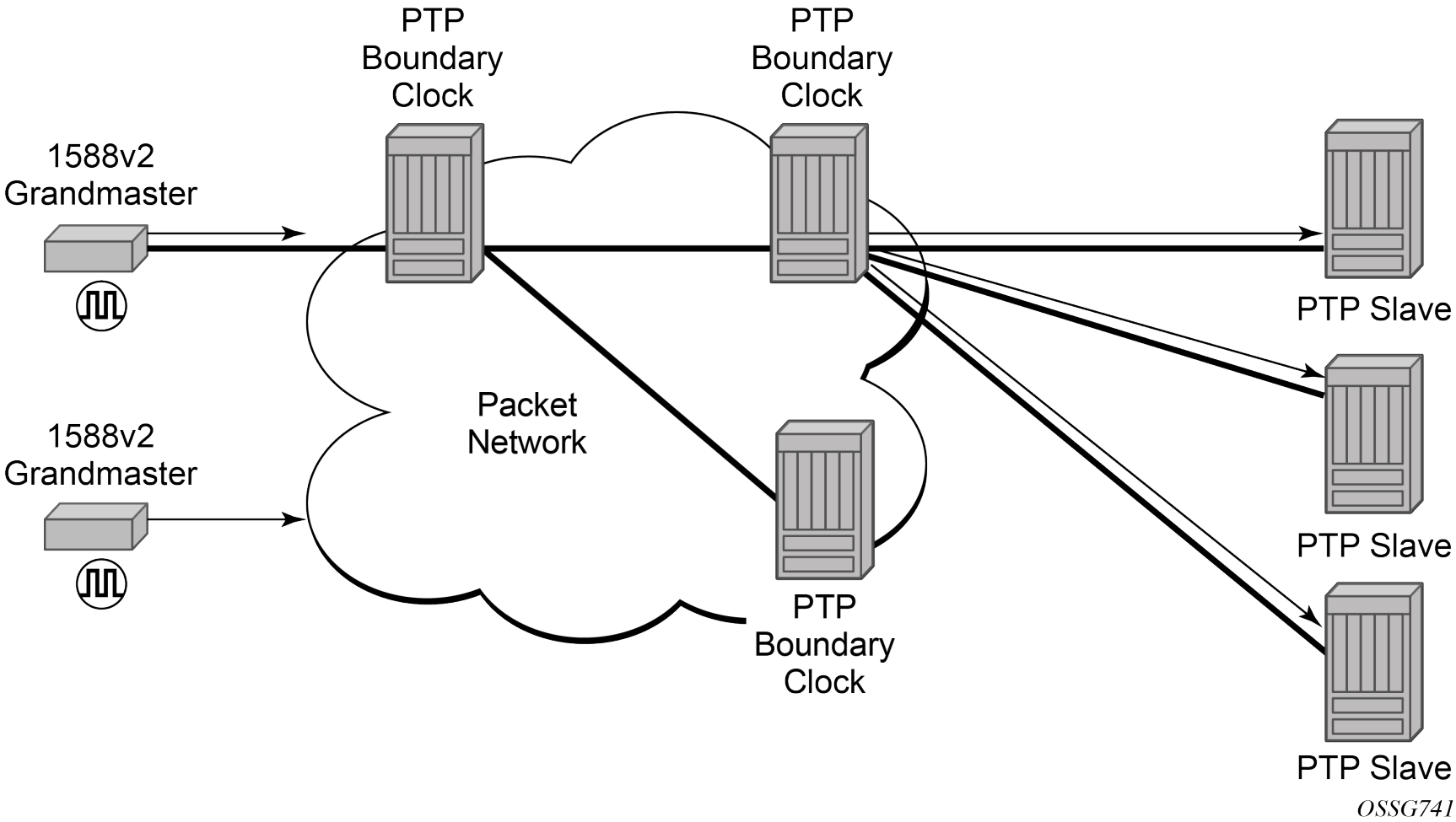
$$t_2 - t_1 = delay + offset \quad (6)$$

$$t_4 - t_3 = delay - offset \quad (7)$$

Solve for *offset*, *delay*.

No ambiguity, no added uncertainty, in theory.

PTP Boundary Clock Architecture



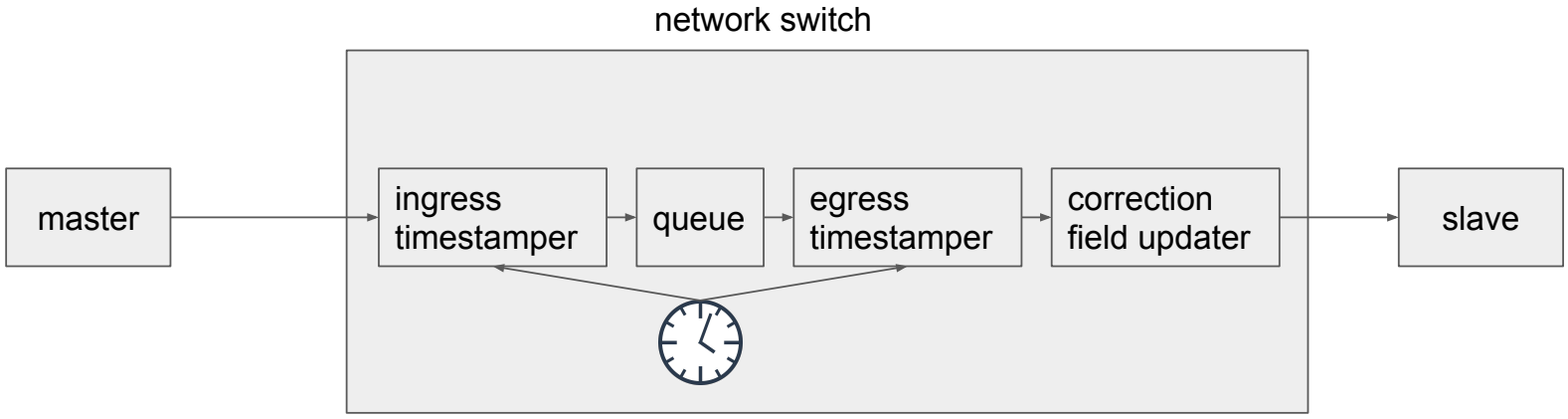
Boundary clock functionality is built into network switches and routers.

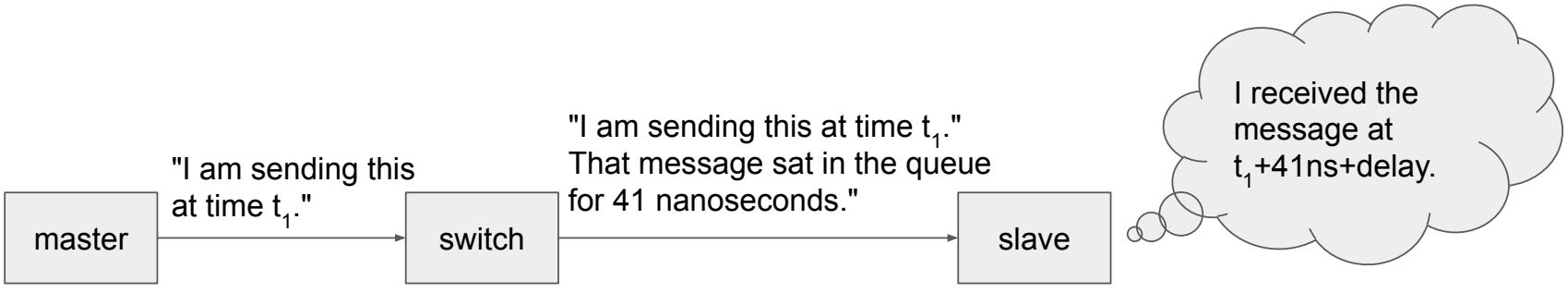
---

figure from Nokia

PTP Transparent Clock Architecture

Removing Queuing Delay from Offset Measurements





PTP clients

Two step process:

1. Sync the NIC's clock with the PTP server.
2. Sync the system clock to the NIC clock.

The latency-critical stuff happens in the NIC. The protocol can be driven by a userspace daemon.

Optional byproduct from NIC: A one pulse-per-second electrical signal that is closely aligned with top-of-second.

Mainstream 2024 Practice

- GPS-based PTP server on the campus network.
- Antennas on the roof.
- Client machines sync their clocks with PTP.

For more ambitious sites, add:

- Redundant GPS-based PTP servers with redundant antennas.
- Atomic oscillators feeding a frequency reference to the PTP servers to maintain tight sync with UTC during periods of GPS unavailability.
- High stability oscillators in client machines so they can maintain tight sync during brief PTP outages.
- Instrument some machines to enable independent end-to-end validation of time sync accuracy.