The Network Time Protocol

Presenter: Mark Barbone

About the Author: David Mills

- Died in January
- Mostly known for NTP
- Also made the first internet routers
  - “Fuzzball routers”
  - Used to prototype TCP/IP

Image source: David Mills’s website

Logical vs. actual clocks

Logical clocks (Lamport, 1978)

- Maintain causal ordering
- Unrelated to what time it is

Actual clocks (c. 1200 BC)

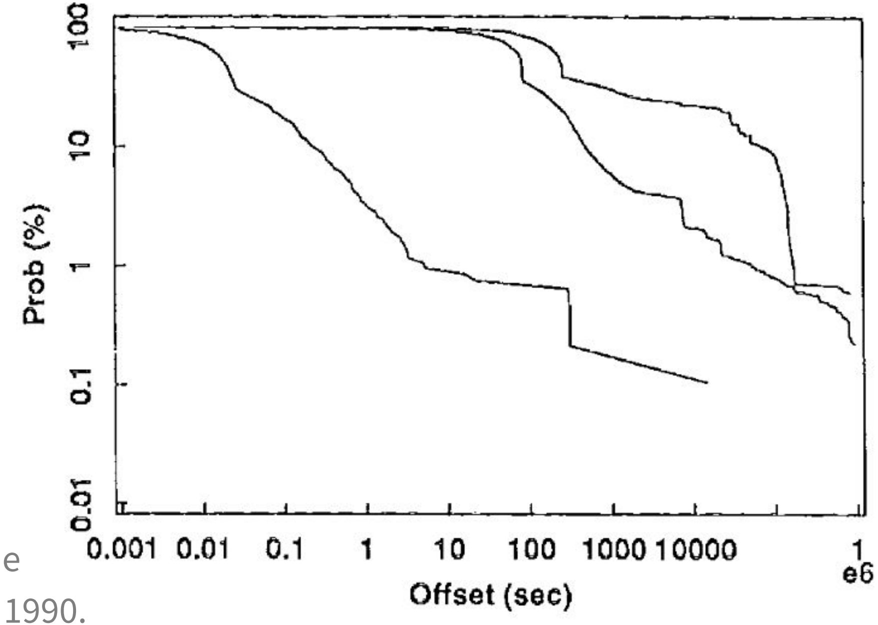
- Tell you what time it is
- Rarely* guarantee causal ordering

Problem

People want their computer clocks set to the right time!

Most common mechanism at the time: look at your watch and set the computer clock yourself

- *Offset:* difference in wall clock times
- *Skew:* difference in frequencies



On the Accuracy and Stability of Clocks Synchronized by the Network Time Protocol in the Internet System. David Mills, 1990.

Prior art

- “Time of day” phone line (POPCORN) – 1930s
- NIST Automated Computer Time Service – 1988

- IP suite Daytime protocol (RFC 867) – 1983
- IP suite Time protocol (RFC 868) – 1983

- **Network Time Protocol** (RFC 1129) – 1989

Lamport's clock synchronization

- Strongly connected graph of nodes
- Regularly send your neighbors your time
- When you receive a time bigger than yours, update your time to the **max**

$$\text{difference} < d(2\kappa\tau + \xi)$$

- d is number of hops
- κ is clock frequency error
- τ is synchronization period
- ξ is communication time

Discussion

Thoughts?

- Why do we even care about clock synchronization?
- What's wrong with the existing protocols?

NTP, simplified

Nodes are all part of the "NTP subnet", and we assume they know of some peers.

1. Pick a server
2. Send a NTP packet to the server
3. Get a response back
4. Compute offset
5. Fix your clock

2-3. Send + receive packets

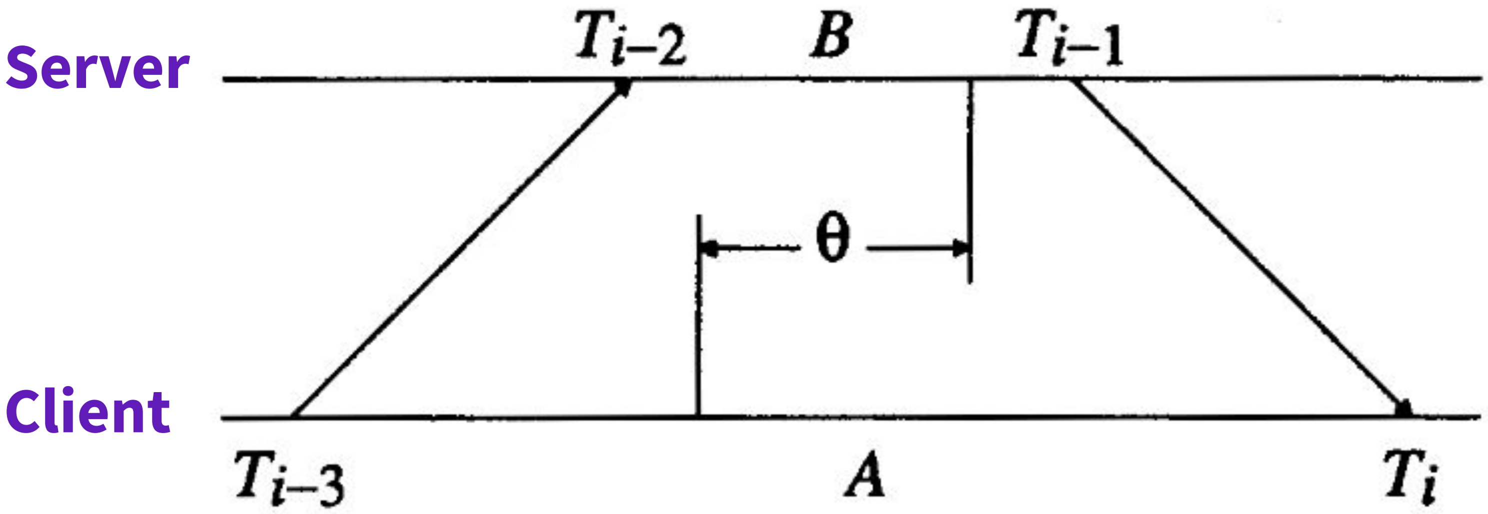


Fig. 3. Measuring delay and offset.

4. Compute offset

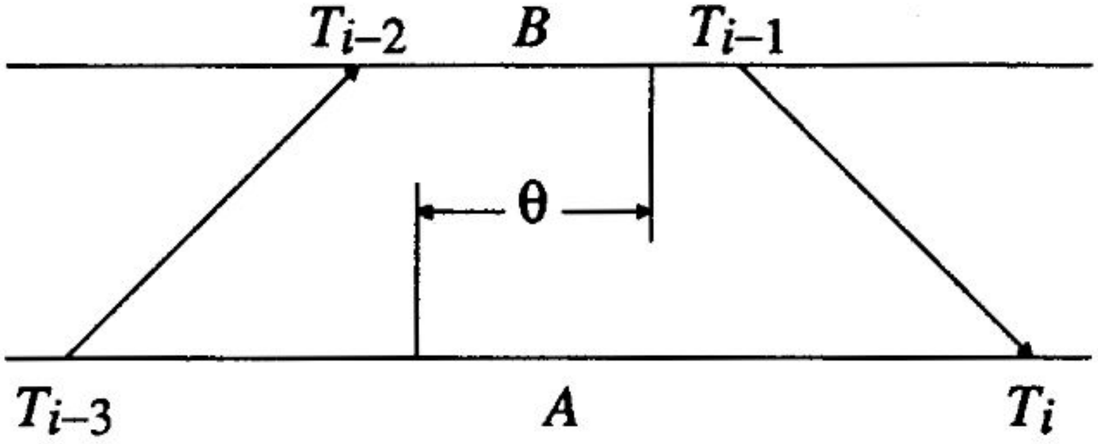
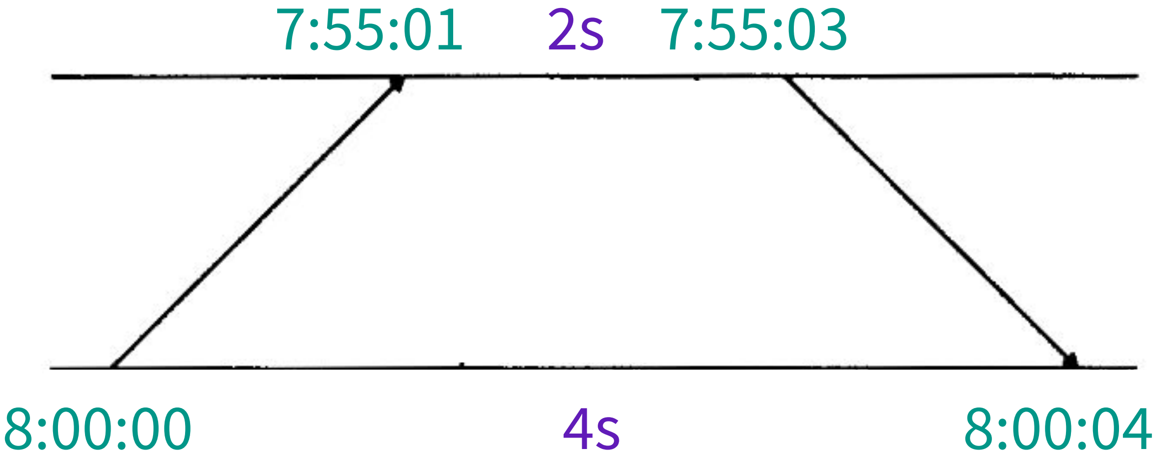


Fig. 3. Measuring delay and offset.

one-way delay ≈ (A - B) / 2

offset = T(i) – T(i-1) – one-way delay

4. Compute offset



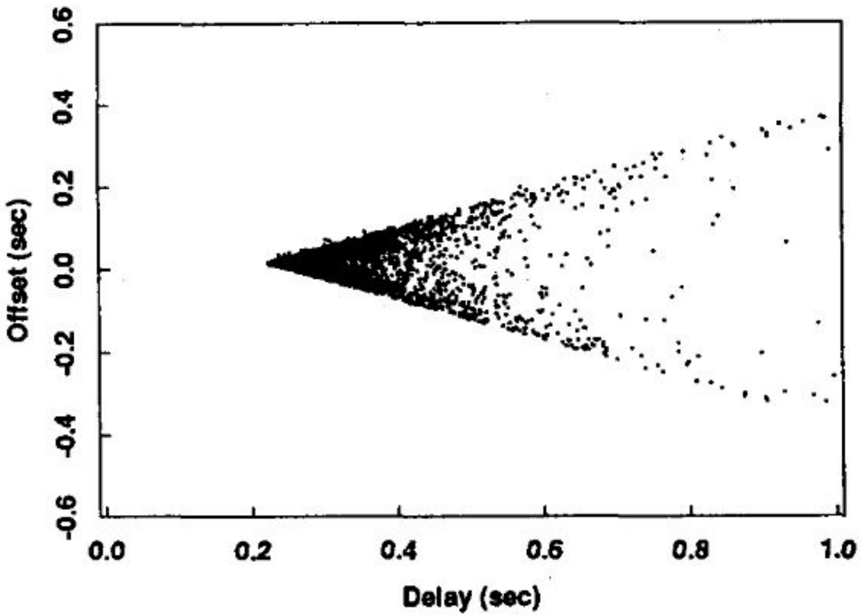
one-way delay ≈ 1 second

offset = 8:00:04 – 7:55:03 – 1 second

= 5 minutes

4. Compute offset

- Measure time difference several times (steps **2**, **3**)
- Pick the one with **fastest round-trip time**



1. Pick a server

- Talk to several peers (steps **2**, **3**, **4**)
- Sort them by number of hops to a reference clock
- Remove ones with too high estimated error
    - “[I]n practice it is not possible to distinguish the **truechimer** clocks, which maintain timekeeping accuracy to a previously published (and trusted) standard, from the **falseticker** clocks, which do not, on other than a **statistical basis**.”
- Pick the best one

Key words: Marzullo’s algorithm; intersection algorithm

1. Pick a server

Original NTP: servers self-organize into a **spanning tree**

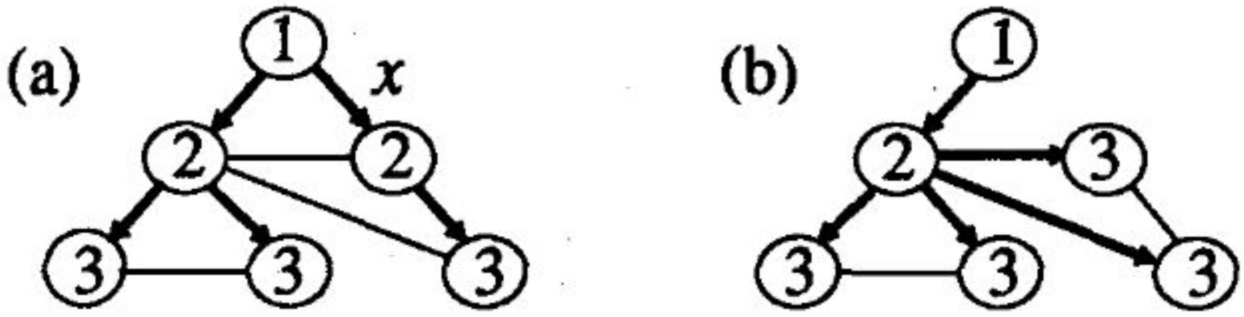
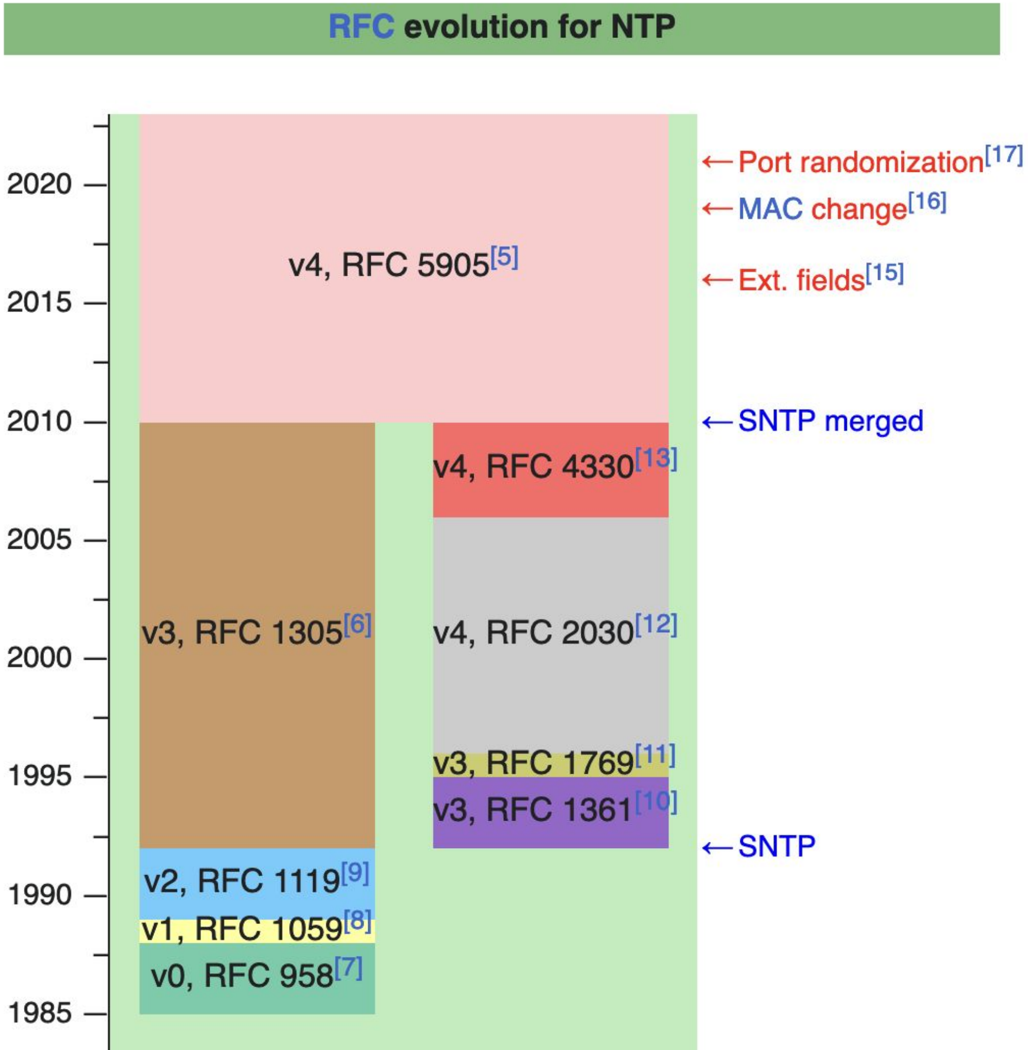


Fig. 1. Subnet synchronization.

Modern users (Linux, Windows) don’t seem to self-organize like this

NTP today


RFC evolution for NTP
2020
2015
2010
2005
2000
1995
1990
1985
v4, RFC 5905[5]
v3, RFC 1305[6]
v2, RFC 1119[9]
v1, RFC 1059[8]
v0, RFC 958[7]
v4, RFC 4330[13]
v4, RFC 2030[12]
v3, RFC 1769[11]
v3, RFC 1361[10]
←Port randomization[17]
←MAC change[16]
←Ext. fields[15]
←SNTP merged
←SNTP

5. Update your clock

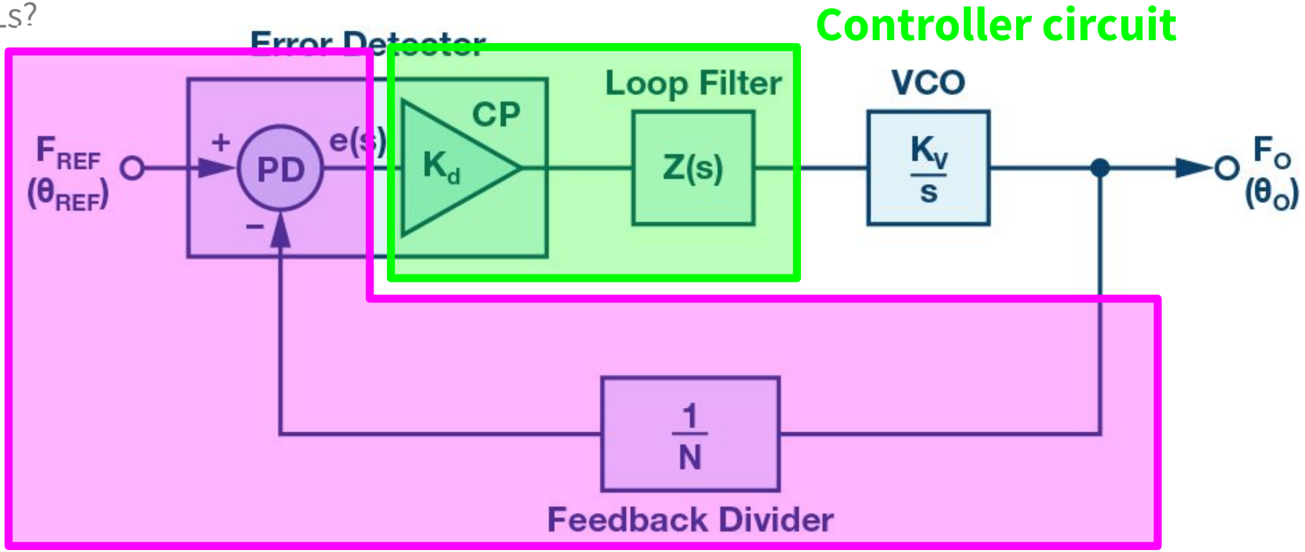
$$\text{clock} \leftarrow \text{clock} - \text{offset}$$

not good enough?

5. Update the clock

- The most confusing part of the paper (for me)
- They **don’t update the clock** directly
- Instead: NTP gives feedback to a controller for oscillator frequency
    - So what’s up with PLLs?

Comparator circuit
(gives the time offset)



Takeaway: Their controller’s update formulas are inspired by PLL’s control circuit

https://www.analog.com/en/resources/analog-dialogue/articles/phase-locked-loop-pll-fundamentals.html

Discussion

Thoughts?

- Benefits of self-organization?
- Drawbacks of self-organization?

- Why not just set the clock?

Modern datacenter time synchronization research

- Google TrueTime (OSDI 2012)
  - NTP-like
- DTP (SIGCOMM 2016)
  - Physical-layer hackery
- Huygens (NSDI 2018)
  - Handles asymmetric transit times
- Sundial (OSDI 2020)
  - Rapid network fault recovery

Uses

- Databases: Google Spanner (OSDI 2012)
- Network congestion control: On-Ramp (NSDI 2021)
  - Improves on Timely (SIGCOMM 2015)