

# Time

CSE 486/586: Distributed Systems

Ethan Blanton

Department of Computer Science and Engineering  
University at Buffalo



# Time

Time is very important to distributed systems.

As we saw, it can be critical for identifying failures.

It can also be used to determine ordering of events.

For most purposes, there must be agreement on timings.

What *is* time, anyway?

# Physical Time

We will leave **defining spacetime** to the physicists!

Instead, we will agree that:

- Time proceeds **forward** monotonically
- The **passage of time** is measurable
- The **relative passage** of time is computable at different locations
- There is an ideal **true time**

# Ideal Time

The **ideal, true physical time** is standardized.

We call it **International Atomic Time (TAI)** [1].

We use the related **Coordinated Universal Time (UTC)** [8].

UTC is a standardized, global reference for “real time.”

Several **government services** distribute UTC via radio:

- NIST (WWV, WWVB), CHU, and DCF77
- GPS, GLONASS, Galileo, and BeiDou satellites

# Celestial Time

Early humans measured time by **celestial motions**:

- The diurnal cycle
- The seasons caused by the Earth's orbit
- Relationships of the sun/planets/stars

The **sundial** is an early celestial timekeeping device.



# Mechanical Oscillators

Later, **mechanical oscillators** based on **physical moment** were developed:

- Tuning forks
- Pendulums
- Spring escapements

These allowed timekeeping to within **seconds per week**.

# Electromechanical Oscillators

In the early 20th century, **crystal oscillators** appeared.

Quartz crystals exhibit **mechanical resonance** when excited by **electrical fields**.

The first precision computer oscillators<sup>1</sup> were crystals.

Modern watches and computers use crystals and related technologies.

Crystal oscillators keep time to within **seconds per year**.

---

<sup>1</sup>“Line clocks,” driven from the electrical power line, have also been used.

# Atomic Clocks and The Second

*The second, symbol  $s$ , is the SI unit of time. It is defined by taking the fixed numerical value of the caesium frequency,  $\Delta\nu_{\text{CS}}$ , the unperturbed ground-state hyperfine transition frequency of the caesium 133 atom, to be 9,192,631,770 when expressed in the unit Hz, which is equal to  $s^{-1}$  [5]*

Atomic clocks [4] measure sub-nuclear particle spin to achieve considerable accuracy.

Cesium and Rubidium standards are reasonably available.

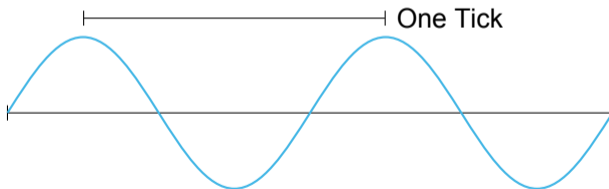
Atomic oscillators keep time to within seconds per millenia.



# Phase and Frequency

Two clocks may disagree on:

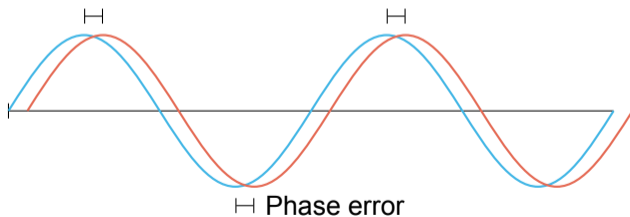
- What time it is: **phase error**
- How fast time is proceeding: **frequency error**



# Phase and Frequency

Two clocks may disagree on:

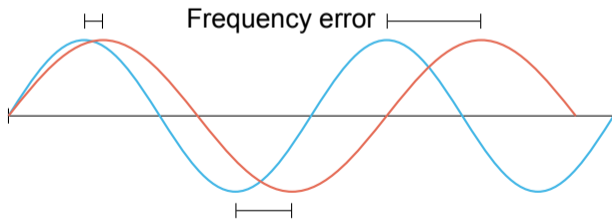
- What time it is: **phase error**
- How fast time is proceeding: **frequency error**



# Phase and Frequency

Two clocks may disagree on:

- What time it is: **phase error**
- How fast time is proceeding: **frequency error**



# Jitter and Discontinuities

Clocks may **change speed slightly** over time.

If this change is **short-term** and **about a point**, we call it **jitter**.

Clocks may **jump forward or backward** abruptly.

We call these jumps **discontinuities**.

Neither one is desirable!

# Clock Discipline

A clock may be **corrected** by a process called **disciplining**.

Disciplining is **tweaking the clock** using external information.

For example:

- A rubidium standard has excellent **short-term stability**.
- GPS satellite signals have excellent **long-term stability**.
- The opposites are less true.
- The **short-term stability** of a **GPS signal** can be improved by combining the two into a **GPS-disciplined oscillator**.

# Thought Experiment

I give you an **extremely stable** oscillator.

It pulses **exactly** once per second, with zero error.

What time is it?

# Synchronization

Agreeing on the **phase** and **frequency** of time is hard.

It is **much easier now** than it was just a few years ago!

Computer clocks can be readily synchronized to about  $\pm 10$  ms.

External time sources can reduce this to about  $\pm 10$   $\mu$ s.

Higher precision is attainable with **more effort**.

# Network Time Services

It is **impractical** to attach a GPS to every computer.

(If nothing else they **require a view of the sky!**)

We use **network time protocols** to synchronize clocks.

They can be **very precise** over fast local networks.

They can be **accurate to tens of ms** over the global Internet [7].

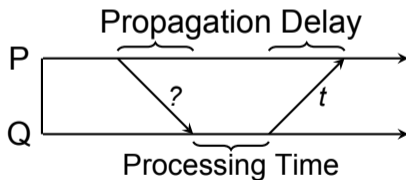


# A Trivial Approach

A trivial approach to synchronization:

- $P$  sends “What time is it?” to  $Q$
- $Q$  replies with “It is time  $t$ ”

The problem is **delay**:



$P$ 's clock is **behind**  $Q$ .

# Delay

Communication delay leads to phase error.

Communication delay cannot be eliminated.

(Why not? The speed of light.)

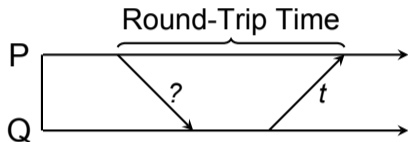
It is also hard to measure.

Think about it:

If clocks aren't already synchronized, how do you measure it?

# Cristian's Algorithm

Cristian's Algorithm [2] tries to **estimate and remove** delay.



It does this by removing **1/2 round-trip time** from Q's response.

Assuming that:

- Propagation delays are equal
- Processing time is small

...this **approximates the error** in the delay.

# NTP

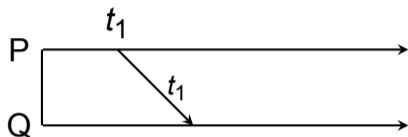
The **NTP protocol** [6] tries to tighten this bound.

It improves performance over **the global Internet**.

It does this by:

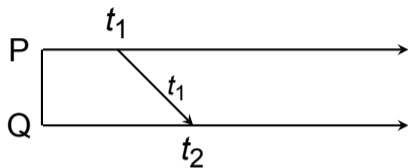
- Including some **extra timestamps**
- Querying **multiple time servers**

# NTP Exchange



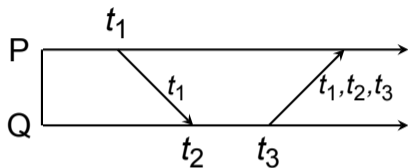
$P$  sends  $t_1$ , the **local** time at which it sends its query.

# NTP Exchange



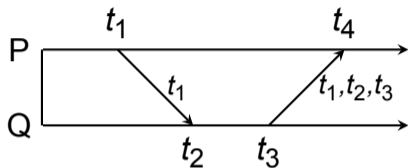
Q records the **local** time  $t_2$  when it receives the query.

# NTP Exchange



Q eventually sends its reply at time  $t_3$ , and includes all three timestamps.

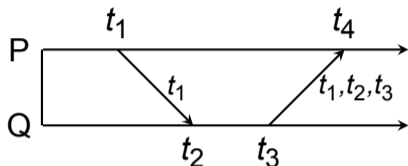
# NTP Exchange



$P$  receives the reply at time  $t_4$  and records all four timestamps.



# NTP Exchange



$P$  can calculate:

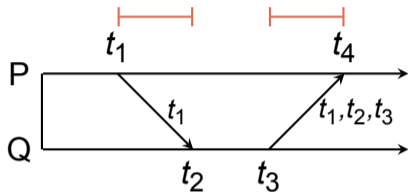
- The **phase difference** between  $P$  and  $Q$ 's clocks:

$$\Theta = \frac{1}{2} [(t_2 - t_1) + (t_3 - t_4)]$$

- The round-trip delay:

$$\delta = (t_4 - t_1) - (t_3 - t_2)$$

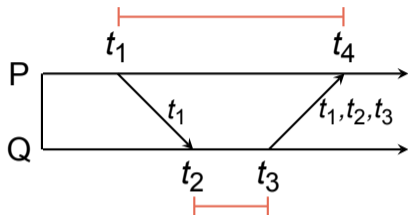
# NTP Exchange



$$\Theta = \frac{1}{2} [(t_2 - t_1) + (t_3 - t_4)]$$

These intervals are **differences between the  $P$  and  $Q$  clocks.**

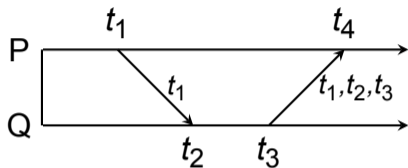
# NTP Exchange



$$\delta = (t_4 - t_1) - (t_3 - t_2)$$

These intervals are **local measurements at P and Q.**

# NTP Exchange



$\Theta$  is the adjustment for  $P$ 's clock.

$\delta$  bounds the error between the clocks at  $P$  and  $Q$ .

$\delta$  is calculated into the goodness of  $Q$  as a time source.

# Additional Complexities

NTP tracks **multiple servers** and tries to build a picture of:

- The **absolute accuracy** of server clocks.
- **Network conditions** affecting time signals
- The **relative frequency error** of system clocks
- The **relative phase error** of the local clock to UTC

It continuously selects the **best estimate of UTC** and **disciplines the local clock**.

By **tracking local frequency error** it can survive **network outages**.

# Summary

- Time is **important** to distributed systems
- There are standards for **measuring time**
- Different **clock technologies** have strengths and weaknesses
- Clocks experience relative **phase** and **frequency** errors
- Synchronization protocols must deal with **network delays**
- NTP provides **robust synchronization** over **Internet paths**

# Next Time ...

- Logical Clocks

# References I

## Required Readings

- [3] Ajay D. Kshemkalyani and Mukesh Singhal. *Distributed Computing: Principles, Algorithms, and Systems*. Chapter 3: 3.9. Cambridge University Press, 2008. ISBN: 978-0-521-18984-2.

## Optional Readings

- [1] *BIPM — International Atomic Time*. URL: <https://www.bipm.org/en/bipm/tai/tai>.



## References II

- [2] Flaviu Cristian. “A Probabilistic Approach to Distributed Clock Synchronization”. In: *Proceedings of the International Conference on Distributed Computing Systems*. June 1989, pp. 288–296.  
URL: [https://search.lib.buffalo.edu/permalink/01SUNY\\_BUF/12pkqkt/cdi\\_ieee\\_primary\\_37958](https://search.lib.buffalo.edu/permalink/01SUNY_BUF/12pkqkt/cdi_ieee_primary_37958).
- [4] Arthur O. McCoubrey. “History of Atomic Frequency Standards: A Trip through 20th Century Physics”. In: 1996, pp. 1225–1241.  
URL: <https://ieee-uffc.org/about-us/history/uffc-s-history/history-of-atomic-frequency-standards-a-trip-through-20th-century-physics/>.

## References III

- [5] Bureau International des Poids et Mesures. *The International System of Units (SI) (Ninth Edition)*. 2019. URL: <https://www.bipm.org/utils/common/pdf/si-brochure/SI-Brochure-9.pdf>.
- [6] David L. Mills. *Network Time Protocol Version 4 Reference and Implementation Guide*. Tech. rep. 06-6-1. NTP Working Group, June 2006. URL: <https://www.eecis.udel.edu/~mills/database/reports/ntp4/ntp4.pdf>.
- [7] David L. Mills. *NTP Performance Analysis*. Aug. 2004. URL: <https://www.eecis.udel.edu/~mills/database/brief/perf/perf.pdf>.

## References IV

- [8] *Standard-frequency and time-signal emissions.*  
Recommendation ITU-R TF.460-6. 2002. URL:  
[https://www.itu.int/dms\\_pubrec/itu-r/rec/tf/R-REC-TF.460-6-200202-I!!PDF-E.pdf](https://www.itu.int/dms_pubrec/itu-r/rec/tf/R-REC-TF.460-6-200202-I!!PDF-E.pdf).

Copyright 2023 Ethan Blanton, All Rights Reserved.

Reproduction of this material without written consent of the author is prohibited.

To retrieve a copy of this material, or related materials, see <https://www.cse.buffalo.edu/~eblanton/>.