Clock Synchronization

slide credits: H. Kopetz, P. Puschner

Overview

- The Problem with Clocks
- Internal Clock Synchronization
    - Synchronization Condition
    - Central and Distributed Clock Synchronization
- External Clock Synchronization
    - Algorithms
    - Time Standards

Clock Drift

Real clocks deviate from the reference clock

Clock drift

$$drift^k_i = \frac{z(microtick^k_{i+1}) - z(microtick^k_i)}{g^k}$$

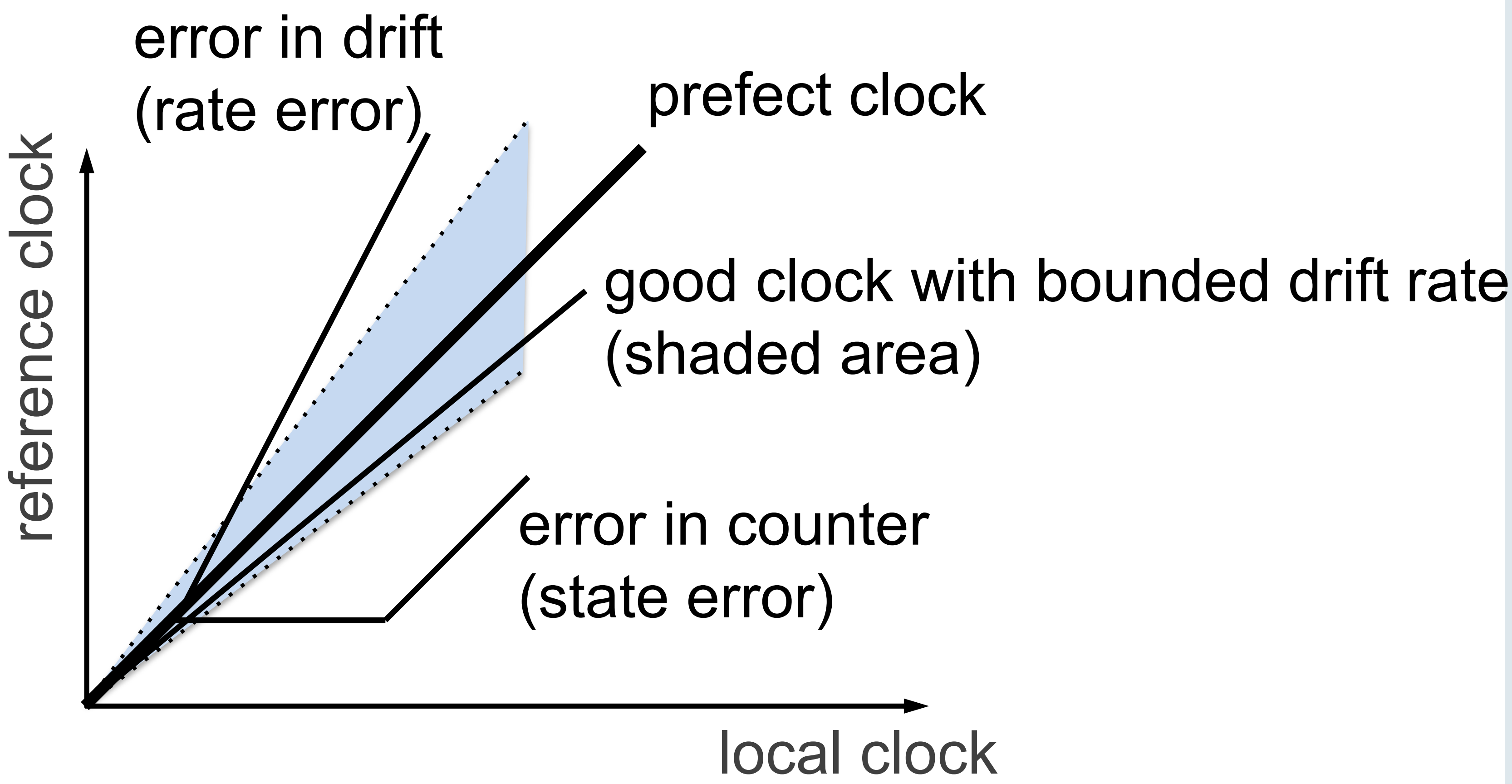
Drift rate

$$\rho^k_i = \left| \frac{z(microtick^k_{i+1}) - z(microtick^k_i)}{g^k} - 1 \right|$$

Drift rate of perfect clock: 0

Drift rate of real clocks: $10^{-8} \ldots 10^{-2}$

Failure Modes of Clocks

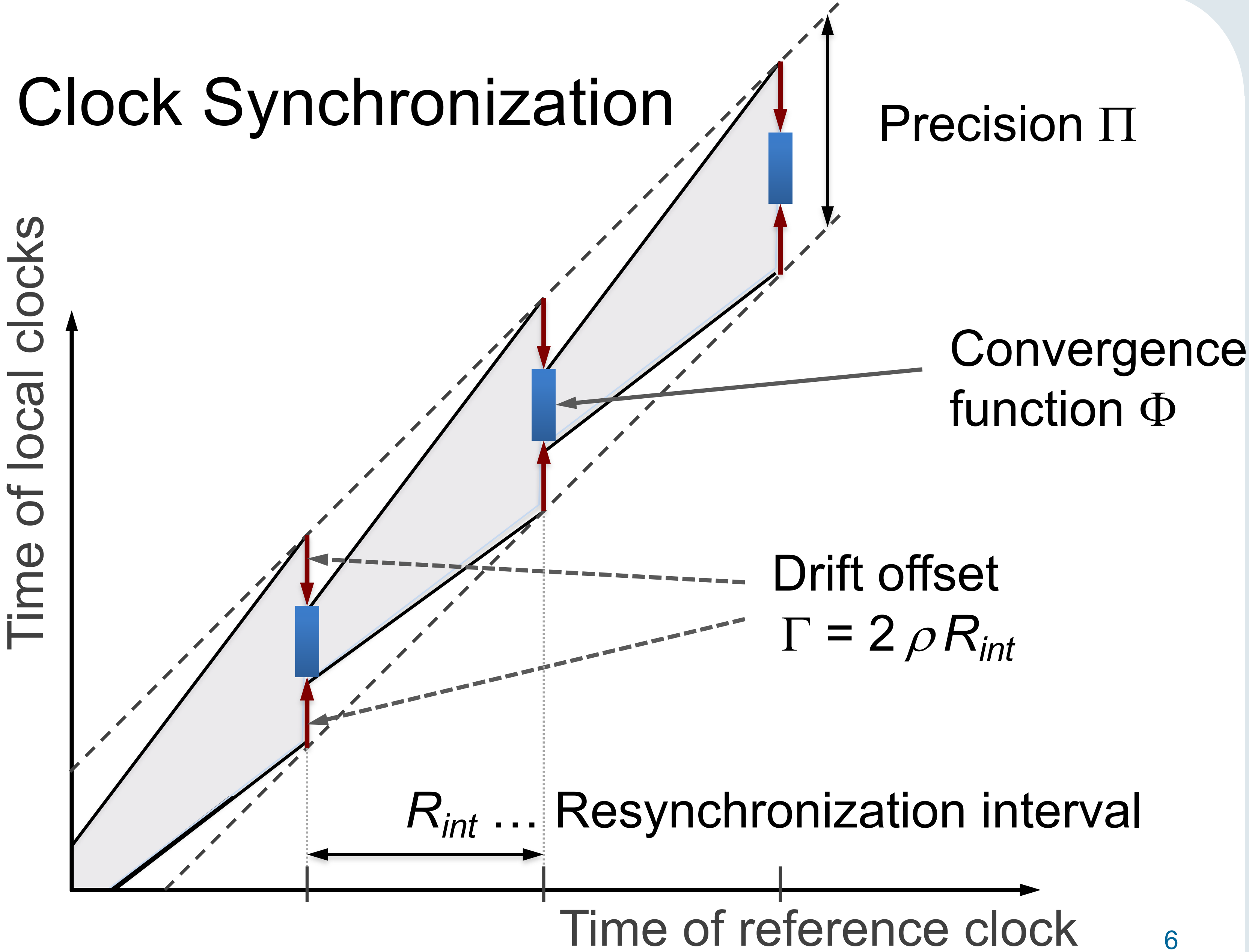


Internal and External Clock Synchronization

Internal clock synchronization: mutual resynchronization of an ensemble of clocks in order to maintain a bounded precision.

External clock synchronization: resynchronization of a clock with the reference clock.

Internal Clock Synchronization



Synchronization Condition

To keep the clocks internally synchronized with precision Π, the synchronization condition must hold:

$$\Phi + \Gamma \leq \Pi$$

Φ … convergence function: max. offset after synchronization;
depends on synchronization algorithm and
message latency jitter ε (= transmission-time difference
between fastest and slowest message, $\varepsilon = d_{max} - d_{min}$)

Γ … drift offset: divergence of free-running clocks; $\Gamma = 2\,\rho\,R_{int}$

R_{int} … resynchronization interval

Central Master Algorithm

- Master node sends periodic synchronization messages, containing its local time
- Slaves adjust local clocks
  - Record local arrival time of sync. message
  - Compute difference *master clock – local clock*
  - Adjust this difference by latency (known, local parameter)
  - Adjust local clock
- Precision of Central Master Algorithm

$$\Pi_{central} = \varepsilon + \Gamma$$

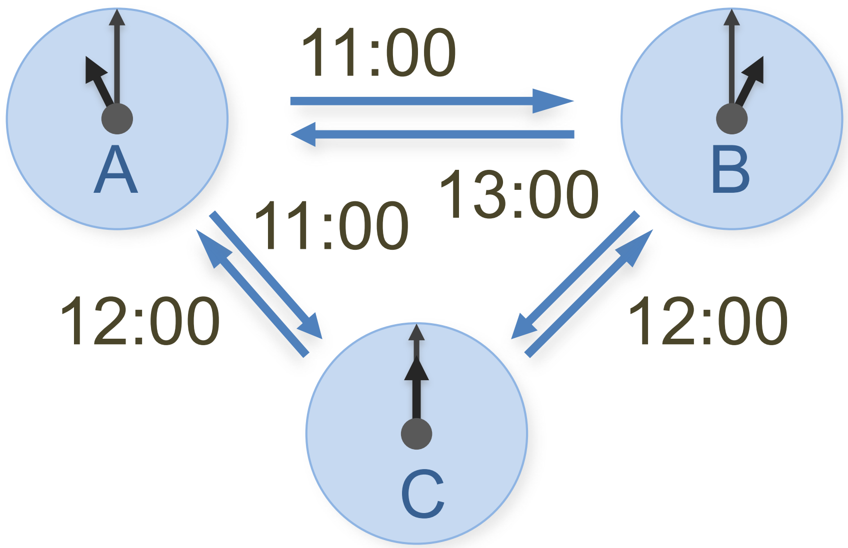
Distributed Clock Synchronization

Use of distributed algorithms to provide fault tolerance;

Typically three phases:

- Nodes exchange messages and acquire information about global-time counters at other nodes.
- Every node analyzes collected information (error detection) and executes the convergence function to compute a correction term for its local global-time counter
- Every node adjusts its local time counter by its correction term

Clock Synchronization – Example



A's view

11:00 (A)
12:00 (C)
13:00 (B)

➪ 12:00

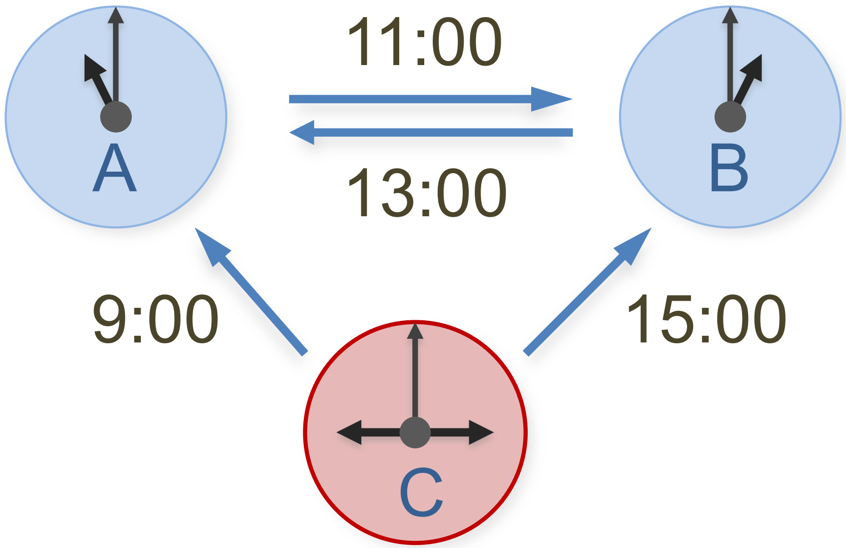
B's, C's view

11:00 (A)
12:00 (C)
13:00 (B)

➪ 12:00

Averaging Algorithm

Malicious (Byzantine) Clocks



A, B: good clocks

two-faced malicious clock

A's view

9:00 (C)
11:00 (A)
13:00 (B)

➪ 11:00

B's view

11:00 (A)
13:00 (B)
15:00 (C)

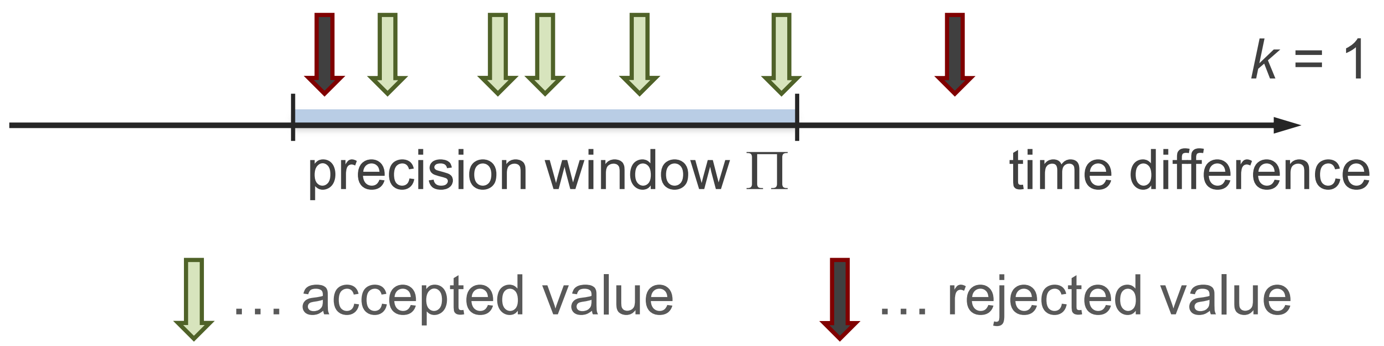
➪ 13:00

Clock synchronization: in the presence of *k* Byzantine clocks the number of clocks, *N*, must be: $N \geq 3k + 1$

Fault-Tolerant Average (FTA) Algorithm

Computation of correction term:

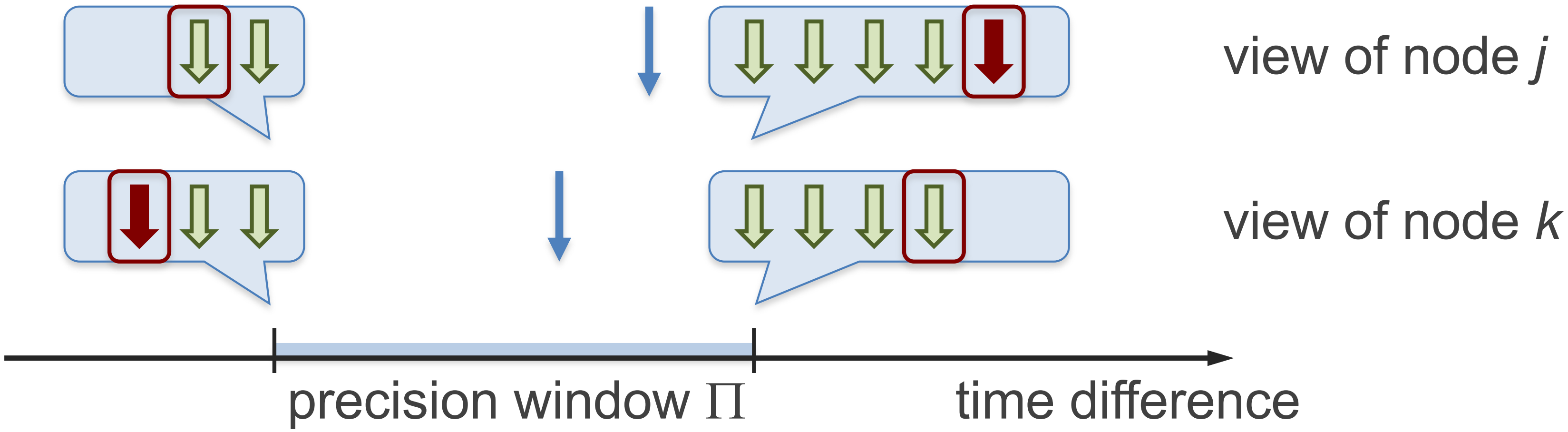
- Calculate differences between local clock and all other clocks
- Sort clock-difference values
- Eliminate k smallest and k largest values
  (k ... max number of erroneous clocks)
- Correction term = average of remaining $N - 2k$ time differences
  (state correction vs. rate correction)

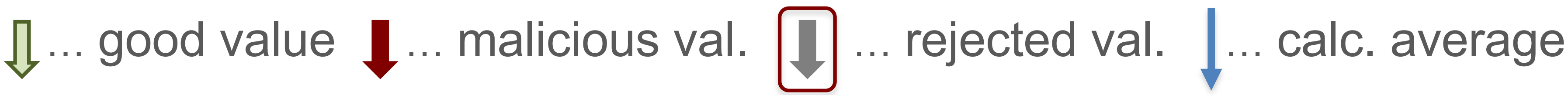


FTA Algorithm – Effect of Byzantine Clock on Φ

Worst-case effect of a Byzantine node:

- Byzantine time values at different ends of precision window
- Error term of a Byzantine error: $E_{byz} = \Pi / (N - 2k)$





Precision of the FTA Algorithm

Convergence Function

$$\Phi(N, k, \varepsilon) = k\,\Pi / (N - 2k) + \varepsilon$$

Precision

$$\Pi(N, k, \varepsilon, \Gamma) = (\varepsilon + \Gamma)\,\frac{N - 2k}{N - 3k} = (\varepsilon + \Gamma)\,\mu(N, k)$$

$\mu(N, k)$ is called the Byzantine error term

number of nodes N

k \ $\mu(N, k)$	4	5	6	7	10	15	20
1	2	1.5	1.33	1.25	1.14	1.08	1.06
2				3	1.5	1.22	1.14
3					4	1.5	1.27

Interactive Consistency Algorithm

Eliminates Byzantine error term

- After collecting the time values of all other clocks, every node sends its view of the clock ensemble to all other clocks
  ➪ extra communication round!
- Nodes have global view; can identify Byzantine nodes
- Correction based on matrix of time vectors of all views
- $\mu(N, k) = 1$

Limit to Internal Clock Synchronization

Lundelius and Lynch show limits of clock synchronization:

The best achievable precision even with perfect clocks is

$$\Pi_{opt} = \varepsilon\,(1 - 1/N)$$

Clock-Synchronization Quality Parameters

- Drift offset $\Gamma = 2 \rho R_{int}$
- Delay jitter $\varepsilon = d_{max} - d_{min}$
- Byzantine failures: rare events
- Clock synchronization algorithms: effect on sync. quality is small compared to delay jitter

Keeping the Drift Offset Small

Minimize relative drift rates of clocks

- Use rate master with precise clock in each cluster
- Adjust rates of local clocks to rate of the master
- Use state correction in FTA
  ➪ mask errors in rate correction of local clocks

Jitter of Synchronization Messages

Message jitter ε depends on where message timestamps are inserted and interpreted

Message assembly/interpretation	appr. range of jitter
Application software level	500 μs … 5 ms
Operating system kernel	10 μs … 100 μs
Hardware: communication controller	< 10 μs

Quality Attributes of a Global Time Base

- Precision
- Accuracy
- Fault tolerance: number and types of faults the system of clocks can tolerate
- Blackout survivability: blackout duration that can be tolerated without losing synchronism

External Clock Synchronization

Synchronize clock ensemble to an external time reference

Example: GPS, achievable accuracy below 1μs

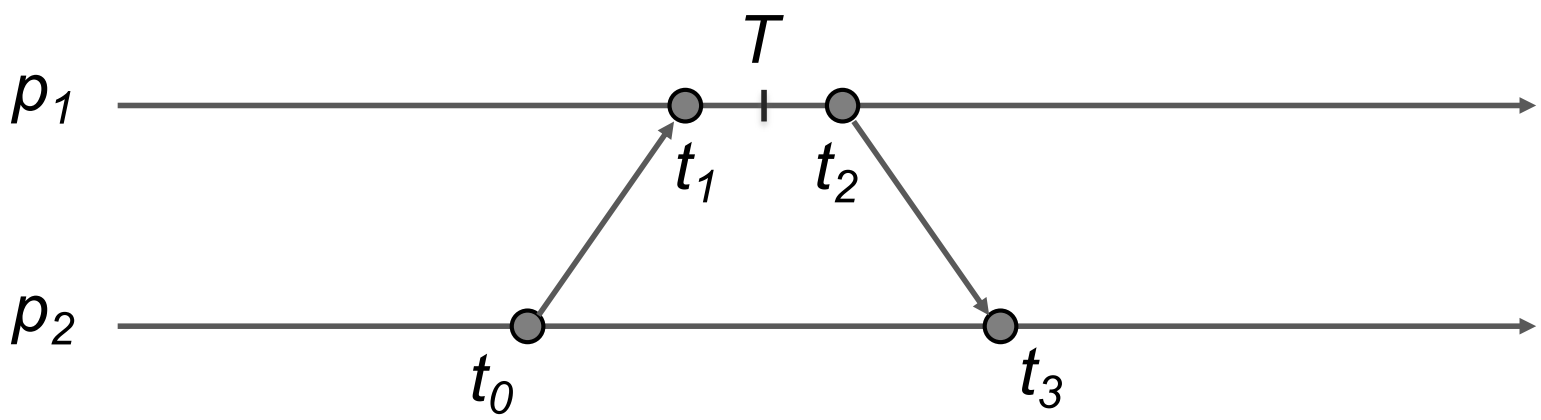
Complementary properties of internal/external synchronization:

- Internal clock synchronization:
  high availability, good short-time stability
- External clock synchronization:
  long-term stability, possibly lower availability

Promising combination:
gateway to external time reference = rate master for internal synchronization

Cristian‘s Algorithm

Request time and evaluate reply



Time-request from p_2 to p_1 at t_0

Reply from p_1 arrives at t_3: contains T, round-trip time $d = t_3 - t_0$

Clock sync: p_2 sets local time to $T + d / 2$

Clock sync. error $\leq d / 2$

Network Time Protocol (NTP)

- Built on idea of Christian's algorithm
- Hierarchy of time servers
  - Class 1: connected to atomic clocks, GPS clocks
  - Class 2: receive time from Class 1 servers, synchronize with other Class 2 devices
  - Class 3: receive time from Class 2 servers, …
- Clock correction based on statistical analysis of t_0 … t_3 of multiple clock readings

Precision Time Protocol (PTP) builds on NTP, uses hardware support for message timestamping to keep ε small

Time Standards

International Atomic Time (TAI)

- physical time standard
- defines the second as the duration of 9 192 631 770 periods of the radiation of a specified transition of the Cesium 133 atom.
- chronoscopic timescale, i.e., a timescale without discontinuities.
- defines the epoch, the origin of time measurement, as Jan. 1, 1958 at 00:00:00 hours

Time Standards (2)

Universal Time Coordinated (UTC)

- astronomical time standard, basis for the time on the "wall clock”.
- duration of the second conforms to the TAI standard
- number of seconds in an hour occasionally modified by inserting a leap second into UTC to maintain synchrony between the wall-clock time and the astronomical phenomena, like day and night.

Adjusting Time can be Tricky ...

Insertion of a leap second at midnight, New Year's Eve 1995, caused a glitch that affected the time signal for the AP radio broadcast network for hours.

Sequence of events:

1. The day increments to January 1, 1996, 00:00:00.
2. The clock is set back one second, to 23:59:59.
3. The clock continues running.
4. The day changes again. Suddenly it is January 2, 00:00:00.

Lessons Learned

- Internal clock synchronization
  - Synchronization Condition
  - Central Master
  - Fault-Tolerant Clock Synchronization
  - Quality Criteria
- External clock synchronization
  - NTP, PTP
  - Standards