Physical clocks synchronisation

https://www.cenam.mx/hora_oficial/

Clock Synchronization

In centralized systems, there is only single clock. A process gets the time by simply issuing a system call to the kernel.

In distributed systems, there is no global clock or common memory. Each processor has its own internal clock and its own notion of time.

Clock Synchronization

For most applications and algorithms that run in a distributed system, we need to know:

- The relative ordering of events that happened on different machines in the network.
- The time interval between two events that happened on different machines in the network.
- The time of the day at which an event happened on a specific machine in the network.

Definitions and Terminology

The **time** of a clock in a machine A is given by the function $C_A(t)$, where $C_A(t) = t$ for a perfect clock.

Clock **offset** (skew) is the difference between the time reported by a clock and the real time.

- The offset of the clock C_A is given by $C_A(t) - t$.
- The offset of clock C_A relative to C_B is given by $\theta_{AB} = C_A(t) - C_B(t)$.

Definitions and Terminology

Frequency is the rate at which a clock progresses. The frequency at time t of clock C_A is $C'_A(t)$.

The **drift** of a clock is the difference in the frequencies of the clock and the perfect clock.

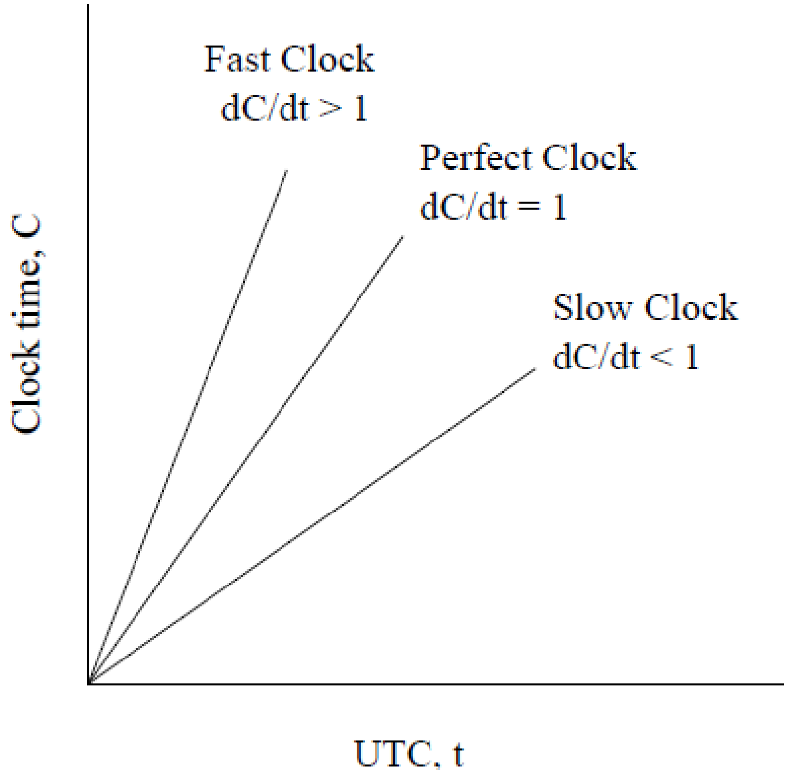
- The drift of a clock C_A at time t is $C'_A(t)-1$.
- The drift of a clock C_A relative to clock C_B is $C'_A(t)-C'_B(t)$.

Definitions and Terminology

A clock is said to be working within its specification if

$$1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$$

where constant ρ is the maximum drift rate specified by the manufacturer.



Definitions and Terminology

The absolute value of the relative drift rate of any two clocks is at most 2ρ

If two clocks are synchronized at time t_0, at any later time t_1 their values can differ at most by $\pm 2\rho(t_1 - t_0)$.

Clock Synchronization

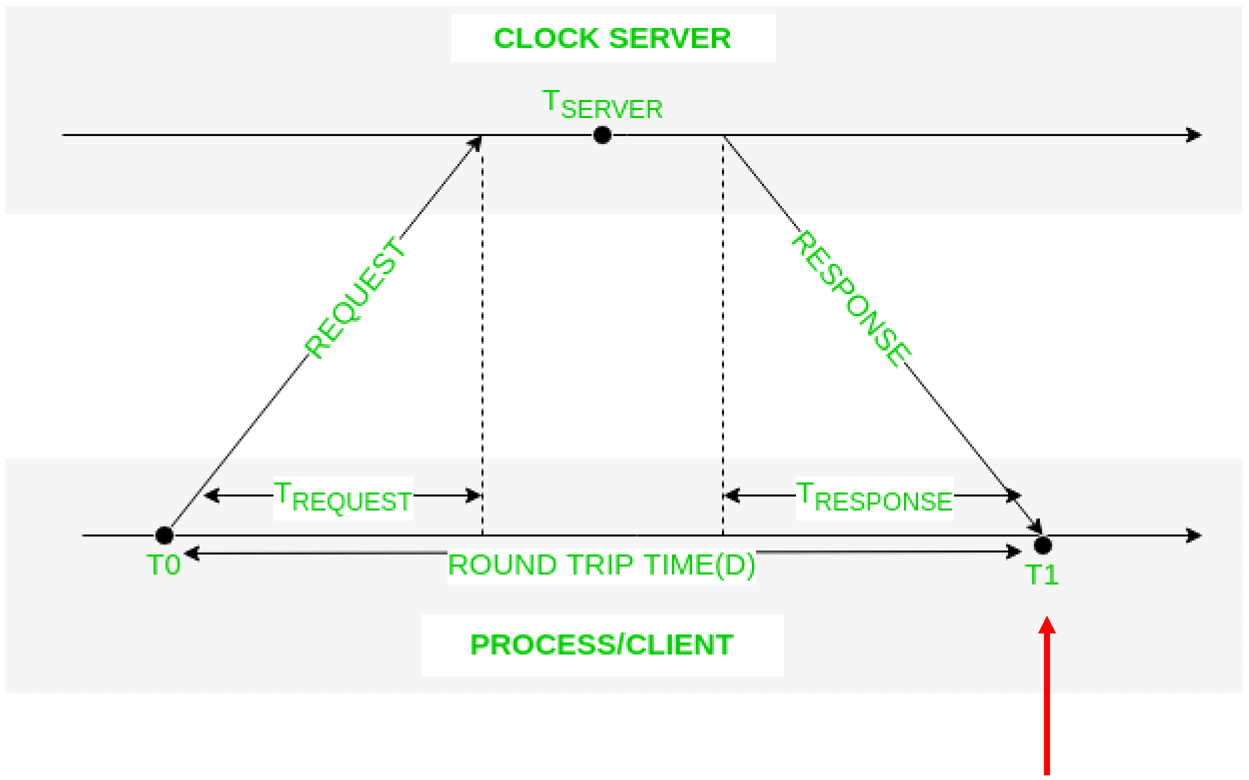
Since two clocks will never have the same frequency, they will tend to drift further and further apart.

Clock synchronization is the process of ensuring that physically distributed processors have a common notion of time.

Probabilistic clock synchronization

Cristian, F. (1989)
Distributed Computing,
Vol. 3, pages 146–158.

Clock Server

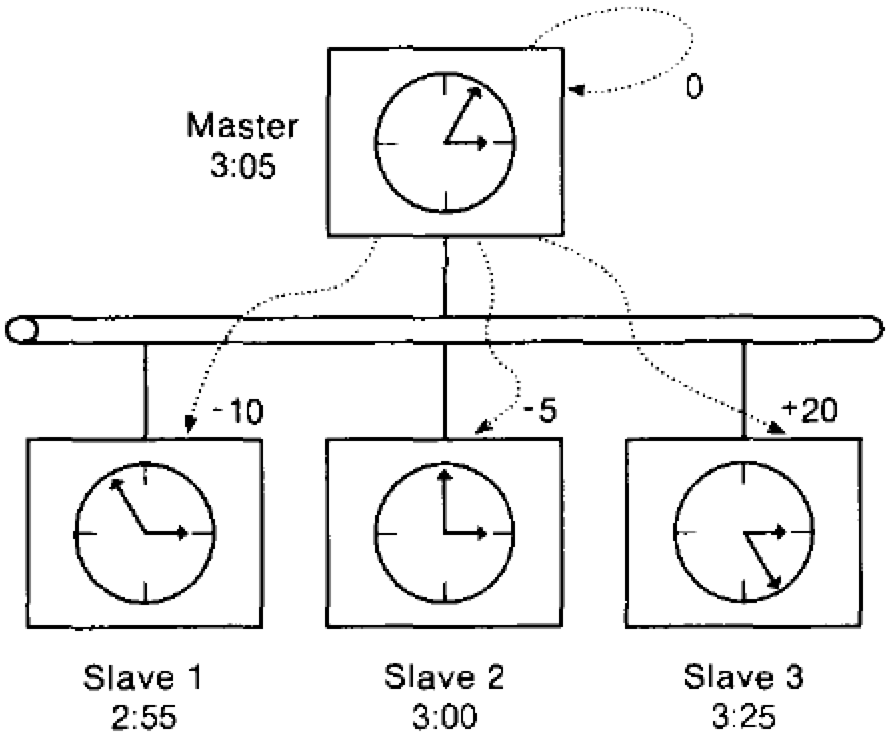


Client set it’s time to be T_{SERVER} + D/2

The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD

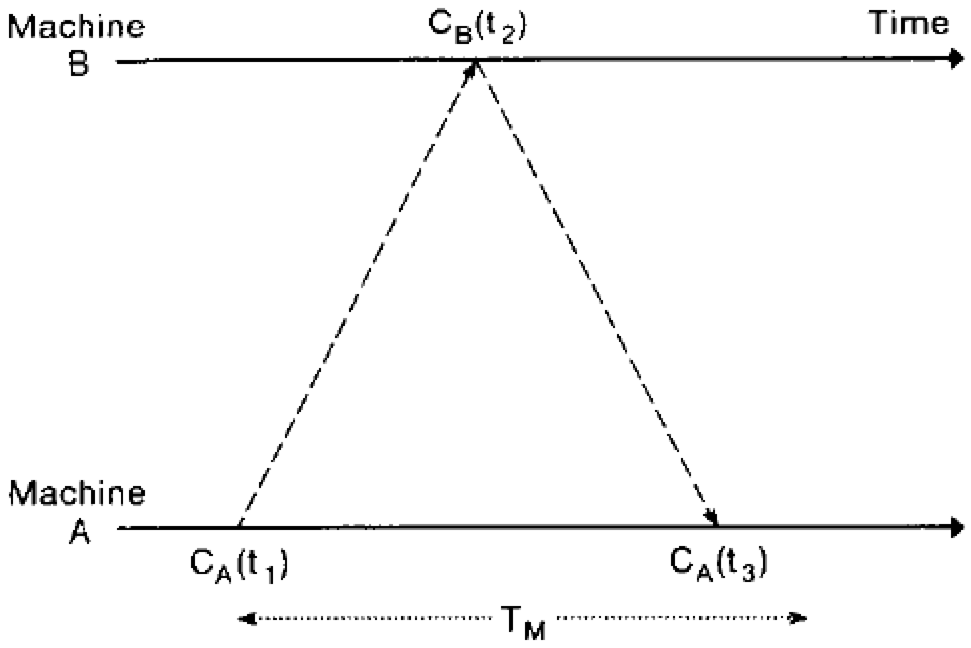
Gusella, R. and Zatti, S. (1989). *IEEE Transactions on Software Engineering*, Vol. 15, No. 7, pp. 847-853

Measurements



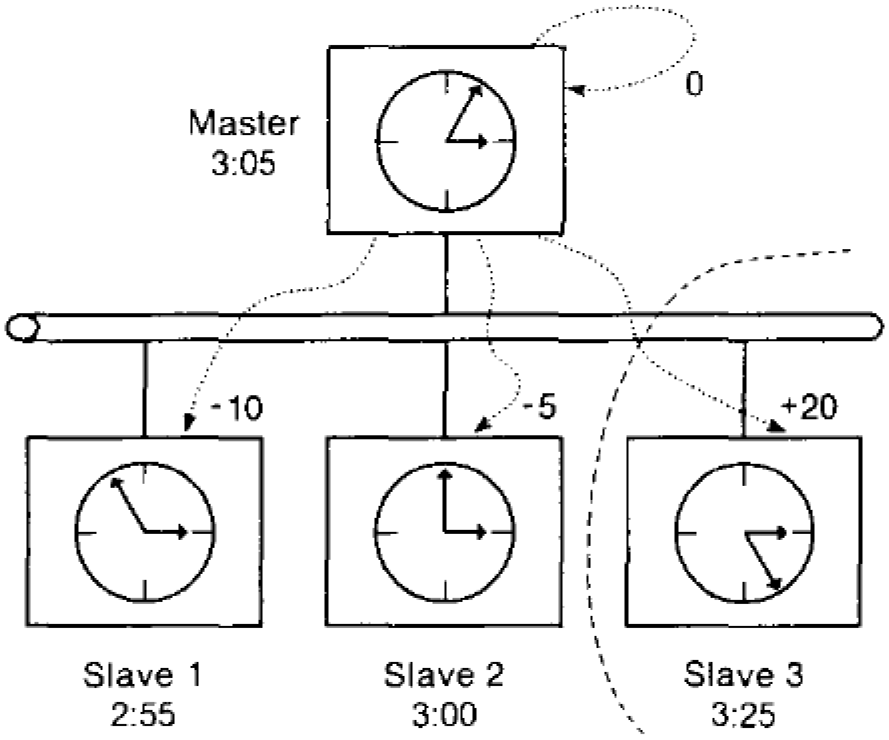
What time is there?

Estimation of the Offset



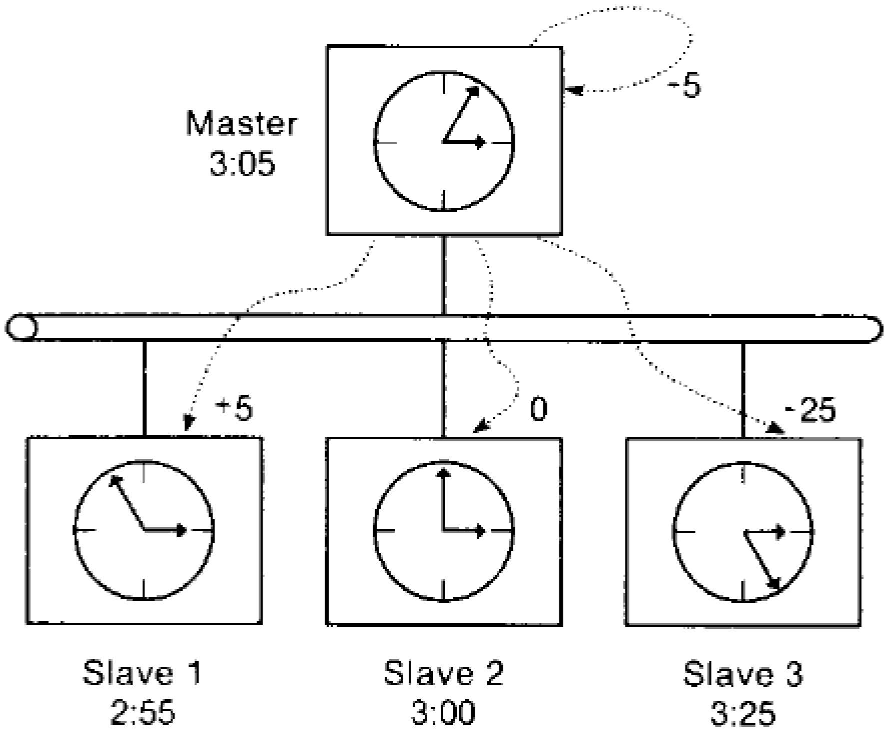
$$\theta_{AB} \approx \frac{C_A(t_1) + C_A(t_3)}{2} - C_B(t_2)$$

Computation of the Average



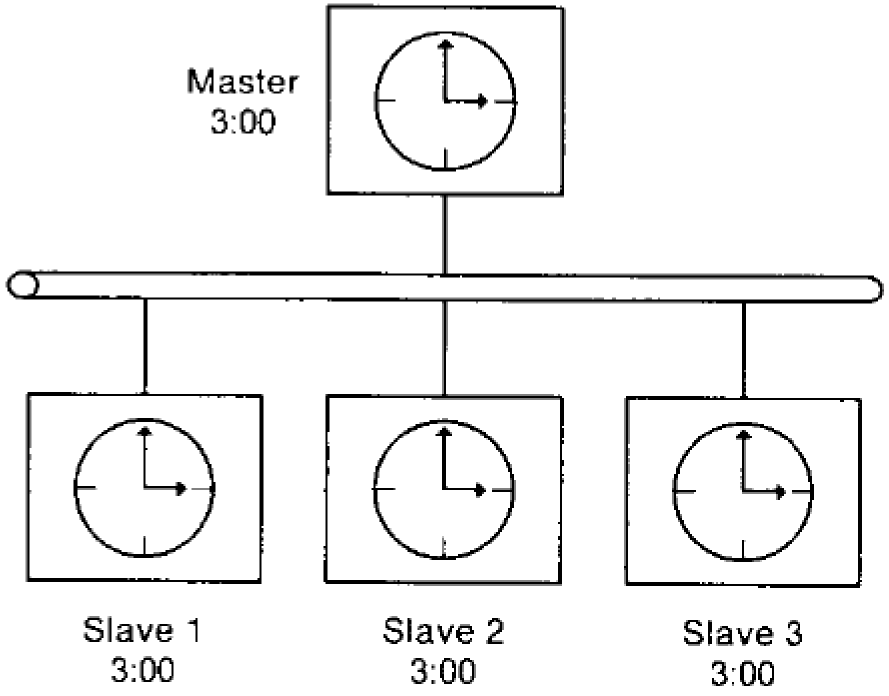
$$Av = \frac{0 - 10 - 5}{3} = -5$$

Correction of the Clocks



Amount of adjustment

Synchronized Clocks



Improved algorithms for synchronizing computer network clocks

Mills, D. (1995). *IEEE Transactions on Networking*, Vol. 3, No. 3, pp. 245–254.

Physical Clocks

Physical clocks are synchronized to an accurate real-time standard like UTC.

Time

UT (Universal Time) is defined by Earth's rotation.

TAI (Temps Atomique International) is a high-precision atomic coordinate time standard.

UTC (Coordinated Universal Time) is based on TAI seconds but stays in phase with UT (introducing leap seconds).

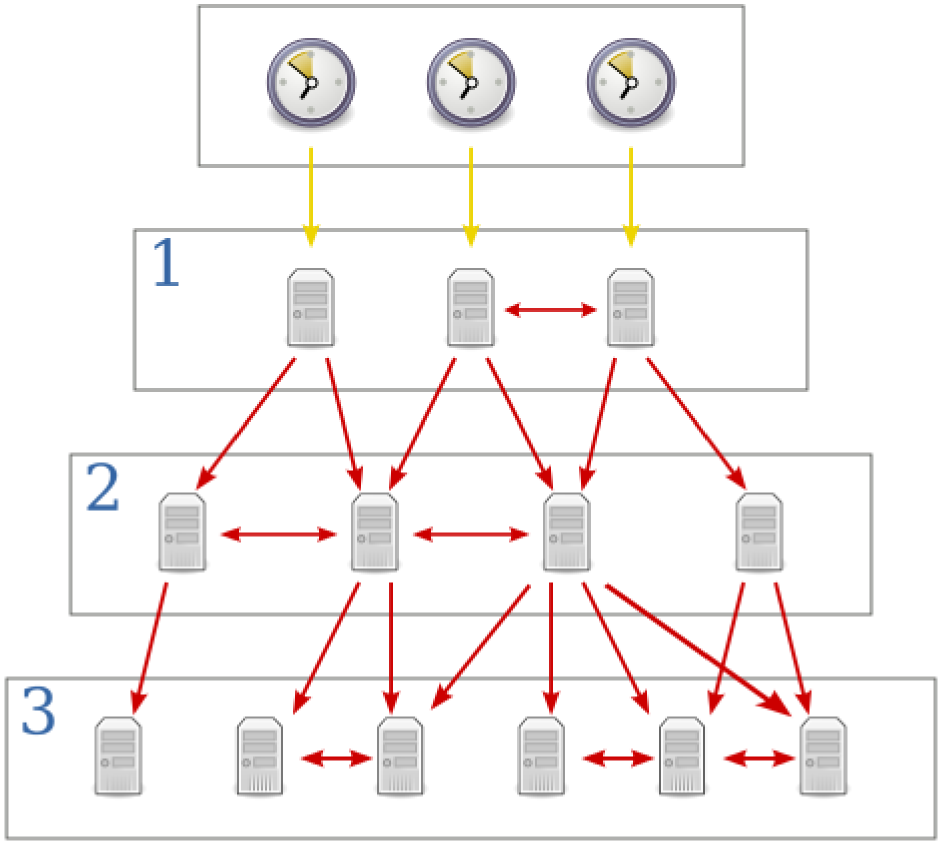
Network Time Protocol

NTP is widely used for clock synchronization on the Internet.

The design of NTP involves a hierarchical tree of time servers.

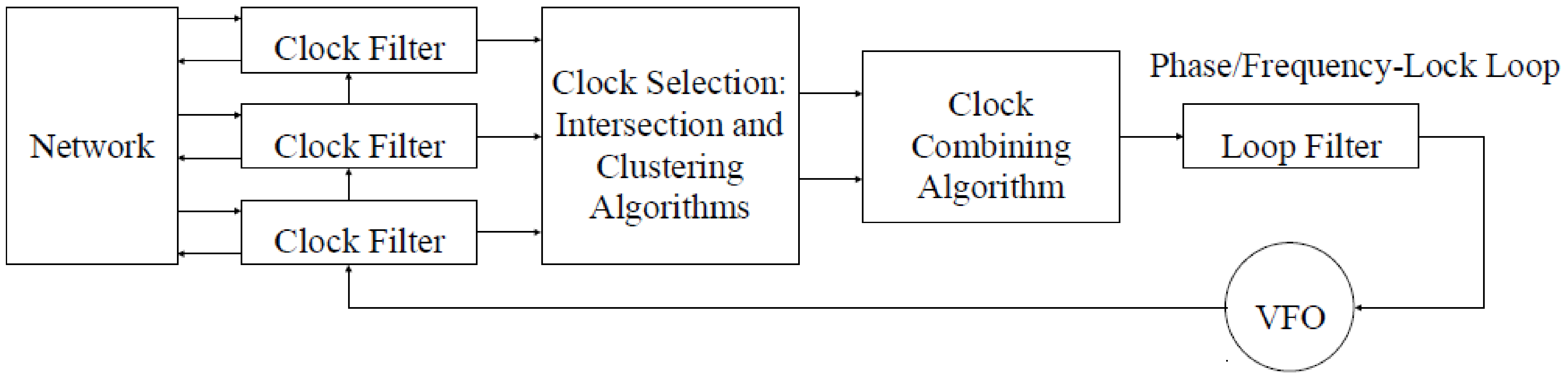
- The primary server at the root synchronizes with the UTC.
- The next level contains secondary servers, which act as a backup to the primary server.
- At the lowest level is the synchronization subnet which has the clients.

Sratum 0



https://www.pool.ntp.org/es/

Overall Organization


Network
Clock Filter
Clock Filter
Clock Filter
Clock Selection: Intersection and Clustering Algorithms
Clock Combining Algorithm
Phase/Frequency-Lock Loop
Loop Filter
VFO

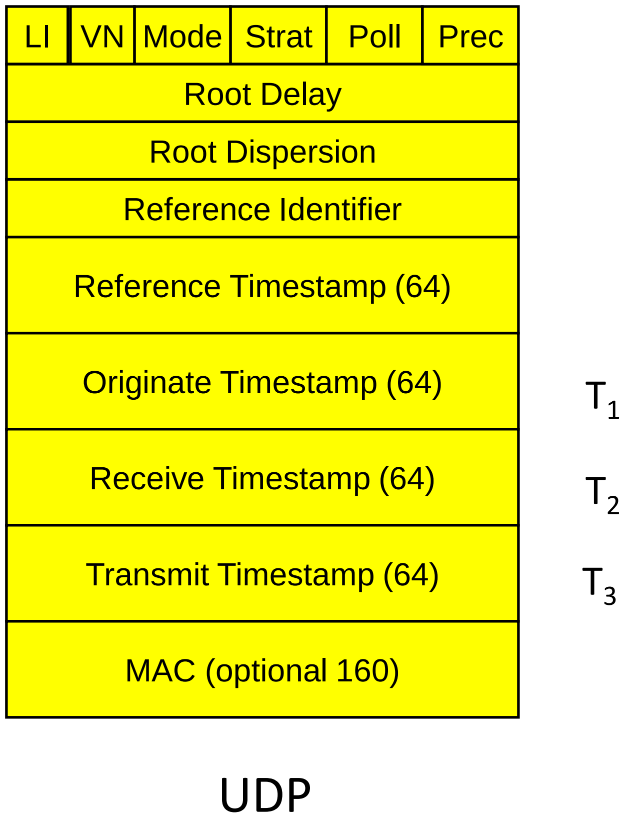
Clock Filter

A pair of servers in symmetric mode exchange pairs of timing messages.

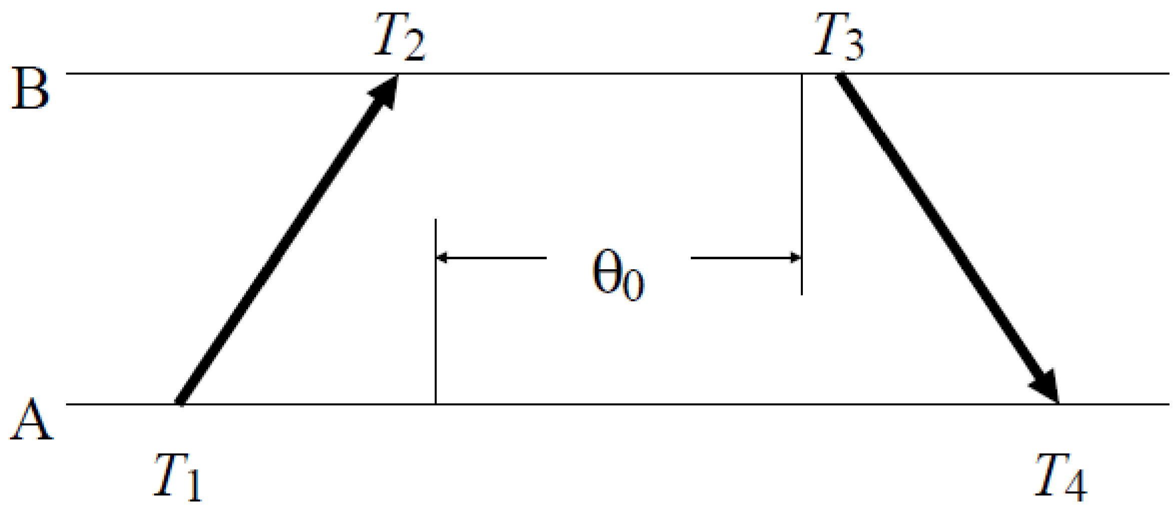
Clock Filter

Each NTP message includes the latest three timestamps T_1, T_2 and T_3, while T_4 is determined upon arrival.

Thus, both peers can independently calculate delay and offset using a single bidirectional message stream.



Clock Filter



Let $a = T_2 - T_1$ and $b = T_3 - T_4$

$\theta = (a + b) / 2$ offset

$\delta = a - b$ roundtrip delay

θ is an estimate of the offset of B relative to A at T_4 and δ is a measure of the accuracy of this estimate.

Clock Filter

In practice, a source node cannot accurately estimate the local time on the target node due to varying message or network delays between the nodes.

NTP employs a common practice of performing several trials.

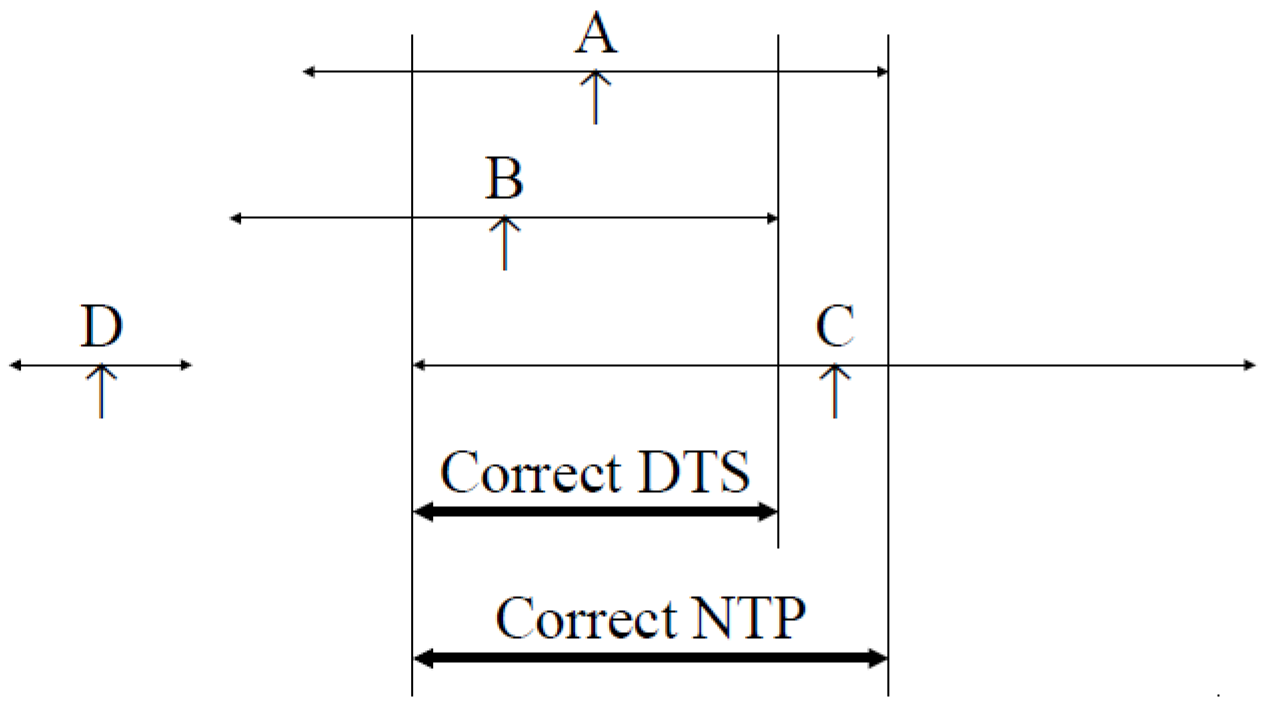
Clock Filter

A store of data is then built up about the relationship between the two servers (pairs of offset and delay).

Each peer maintains the last 8 pairs (θ_i , δ_i)

The offset corresponding to the minimum delay is chosen.

Clock Selection

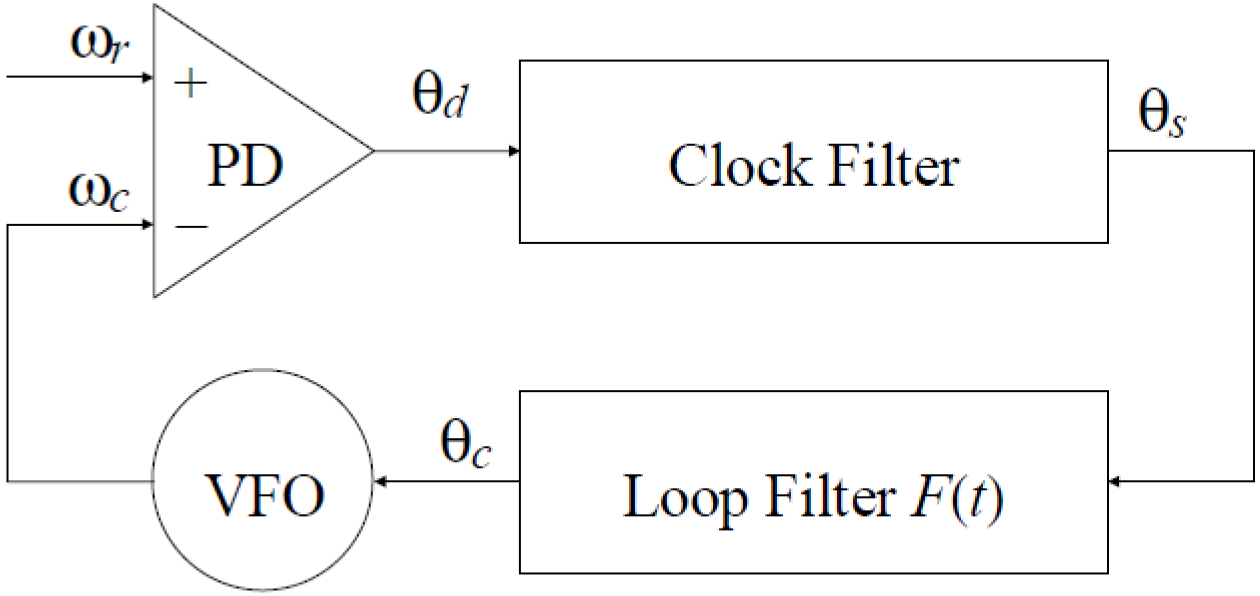


Clock Combining Algorithm

The resulting selected offsets are first combined on a weighted-average basis and then used to drive the clock-discipline algorithm.

Peers with lower stratum numbers are more favored than those in higher strata because they are “closer” to the primary time sources; also, those with the lowest synchronization dispersion are relatively favored.

Disciplined Oscillatior Model



Time, clocks, and the ordering of events in a distributed system

Lamport, L. (1978). *Communications of the ACM*, Vol. 21, No. 7, pp. 558-65.

Anomalous Behavior

Consider a nationwide system of interconnected computers. Suppose a person issues a request A on a computer A, and then telephones a friend in another city to have him issue a request B on a different computer B. It is quite possible for request B to receive a lower timestamp and be ordered before request A.

This can happen because the system has no way of knowing that A actually preceded B, since that precedence information is based on messages external to the system.

Anomalous Behavior

One way to avoid an anomalous behavior is to construct a system of clocks which satisfies the following *condition* for any relevant physical events *a* and *b*

if $a \rightarrow \mathrm{b}$ then $C(a) < C(b)$

Physical Clocks

PC1: $| \, dC_i(t)/dt - 1 \, | < \rho$ $\rho \ll 1$

PC2: *For all i,j :* $| \, C_i(t) - C_j(t) \, | < \varepsilon$

Since two clocks will never have the same frequency, they will tend to drift further and further apart

We must therefore have an algorithm to ensure that *PC2* always holds

Physical Clocks

Let μ be a number such that if event a occurs at physical time t and event b in another process satisfies $a \rightarrow b$, then b occurs later than physical time t+μ

In other words, μ is less than the shortest transmission time for interprocess messages

To avoid anomalous behavior, we must have

$$C_j(t) < C_i(t+\mu).$$

This inequality is true if $\varepsilon/(1-\rho) \leq \mu$

Synchronization Algorithm

μ_m is the minimum delay of a message between two processes (known by the receiving process)

IR1':

For each i, C_i is differentiable at t and $dC_i(t)/dt > 0$

IR2':

(a) P_i sends a message m at t with timestamp $T_m = C_i(t)$

(b) Upon receipt of m at t', P_j sets $C_j(t')$ to $\max(C_j(t'), T_m + \mu_m)$