

Physical Clock Synchronization

Question 1. Why is physical clock synchronization so important?

Question 2. With atomic clocks becoming affordable, should we care about physical clock synchronization?

Types of Synchronization

External Synchronization

Internal Synchronization

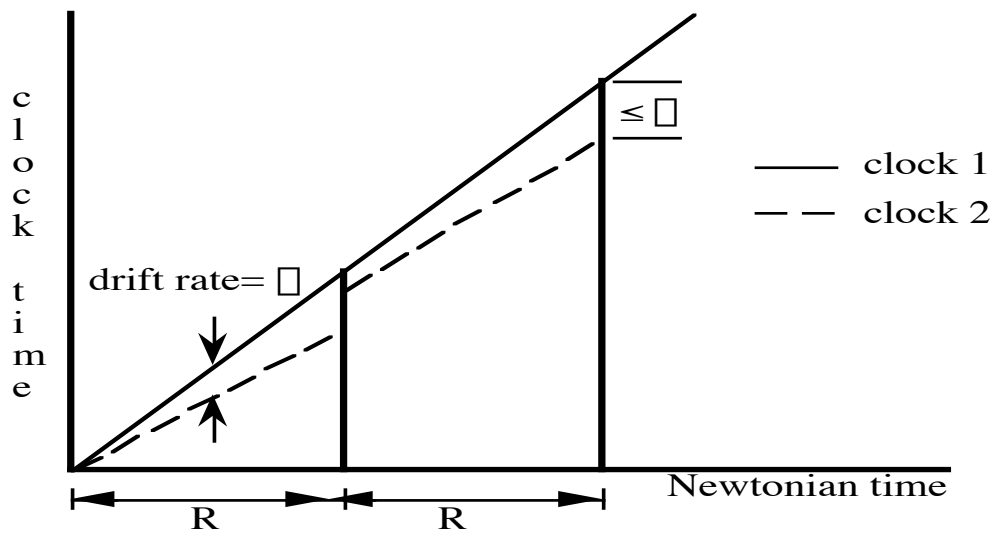
Phase Synchronization

Types of clocks

Unbounded 0, 1, 2, 3, . . .

Bounded 0, 1, 2, . . . M-1, 0, 1, . . .

Unbounded clocks are not realistic, but sometimes they are easier to deal with in algorithms.



What are these?

Drift rate α

Clock skew α

Resynchronization interval R

Challenges

Drift is unavoidable

Accounting for propagation delay

Processing delay

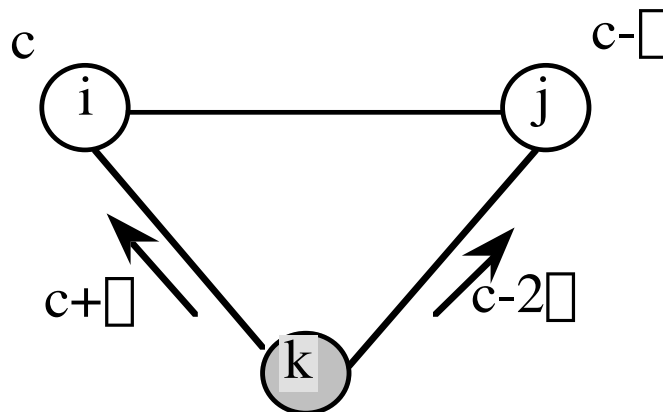
Faulty clocks

Requirements

$$(1-\alpha)(t'-t) \leq C(t')-C(t) \leq (1+\alpha)(t'-t)$$

Internal Synchronization

A simple averaging algorithm



Step 1. Read every clock in the system.

Step 2. Discard "bad" clock values and substitute them by the value of the local clock.

Step 3. Update the clock using the average of these values.

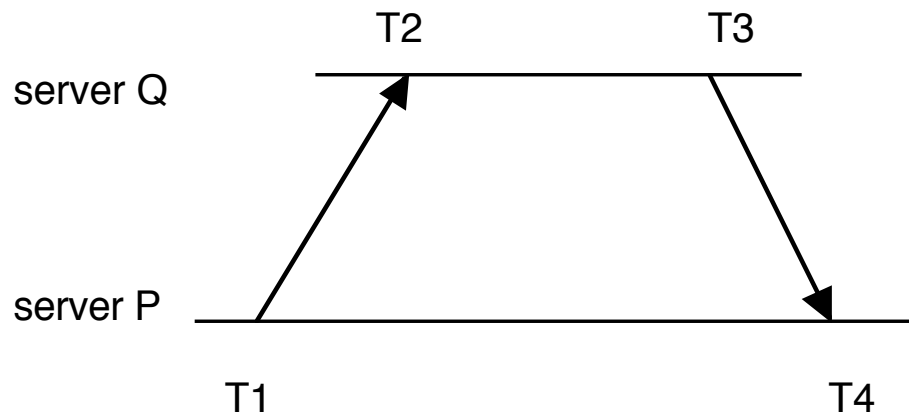
Theorem. Synchronization is maintained if $n > 3t + 1$ where t is the number of faulty (may be 2-faced) clocks.

$$R = \frac{\delta}{\delta(3t+1)} . \text{ Why?}$$

External Synchronization

The Role of Time Server

1. Timeserver *broadcasts* time, computers listen.
2. Computers ask for time (*procedure call*).
(Propagation delay has to be compensated)
3. *Peer-to-peer communication* improves accuracy



Let Q be ahead of P by Δ .

$$T_2 = T_1 + T_{PQ} + \Delta$$

$$T_4 = T_3 + T_{QP} - \Delta$$

(Adding) $T_2 + T_4 = T_1 + T_3 + (T_{PQ} + T_{QP})$

The round trip delay $y = T_{PQ} + T_{QP} = T_2 - T_1 + T_4 - T_3$

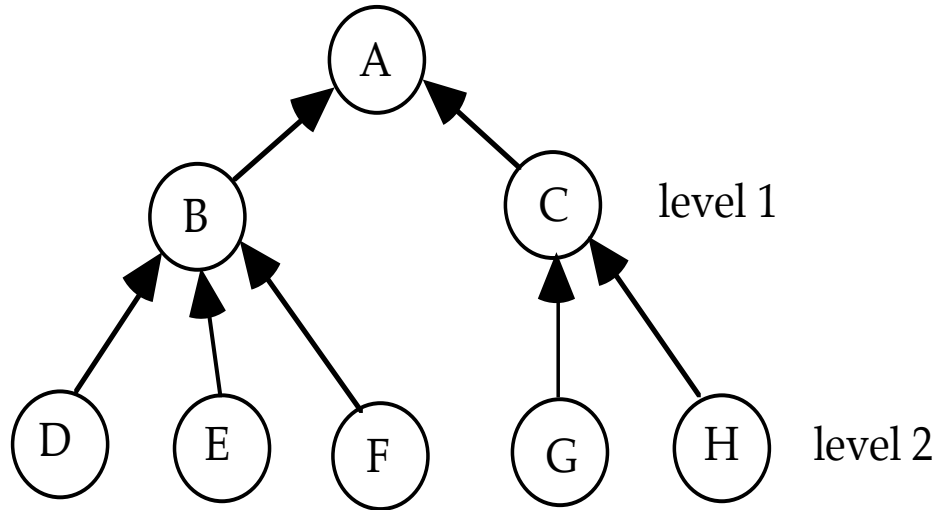
(Subtracting) $2x = (T_2 - T_4 - T_1 + T_3) - (T_{PQ} - T_{QP})$

The offset $x = (T_2 - T_4 - T_1 + T_3) / 2 - (T_{PQ} - T_{QP}) / 2$
 $= x - (T_{PQ} - T_{QP}) / 2$

The value of x will range between $x+y/2$ and $x-y/2$

Record multiple samples of (x,y) . Discard bad values. To estimate x pick those pairs for which y is the *minimum*.

Network Time Protocol (NTP)



Synchronization subnet

Goals

Accurately synchronizes clock on the Internet

Survives loss of connectivity via self-reconfiguration

Provides authenticated time service

The clocks at the *lower levels* are more accurate

Monotonicity criterion $t' > t \Rightarrow C(t') - C(t)$

(This is violated by the Y2K bug)

Clock adjustment

Problems with incrementing clock - missed deadlines

Problems with decrementing clock - the same event
may be scheduled more than once.