Chapter 3
Global Time

External Synchronization and Internal Synchronization

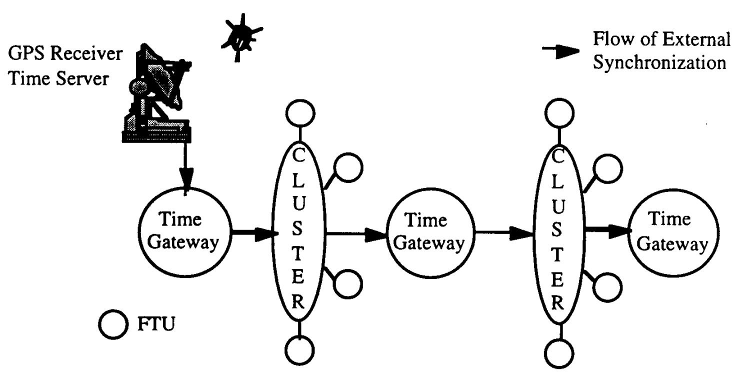


Figure 3.13: Flow of external synchronization.

Global Time -- Basic Notions & Terms

- Temporal order
- Causal order
  - More than the temporal order
- Delivery order
  - Not necessarily related to the temporal order or the causal order

Global Time -- Basic Notions & Terms

- **Local physical clock**
- **Reference clock**
- **Microtick**,
  Granularity := **Duration between 2 consecutive microticks**
- **Time-stamp: clock(event) := TS of the event made by use of the clock**
- **Drift**

$$drift_i^k = \frac{z(microtick_{i+1}^k) - z(microtick_i^k)}{n^k}$$

- **Drift rate**

$$\rho_i^k = \left| \frac{z(microtick_{i+1}^k) - z(microtick_i^k)}{n^k} - 1 \right|.$$

Global Time -- Basic Notions & Terms

- **Failure mode of a clock**

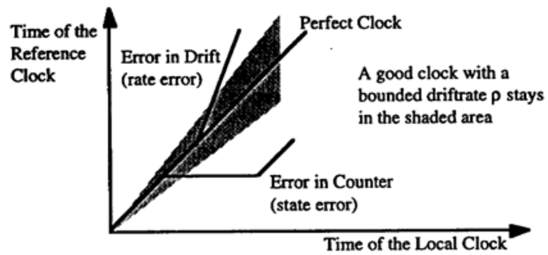


Figure 3.1: Failure modes of a physical clock.

- **Offset: at microtick i between clock j and clock k**

$$offset_i^{jk} = |z(microtick_i^j) - z(microtick_i^k)|$$

UCI DREAM Lab

Global Time -- Basic Notions & Terms

- **Precision**: at microtick i

$$\Pi_i = \max_{\forall 1 \le j,k \le n} \{offset_i^{jk}\}$$

=> **Internal synchronization**

- **Accuracy**: at microtick i
  := Offset of clock k with respect to the reference clock z

=> **External synchronization**

* If all clocks have **accuracy** of A, then their **precision** is at most 2A.

* **Time standards**
  - **Int'l Atomic Time: TAI**
  - **Universal Time Coordinated: UTC**

UCI DREAM Lab

Global Time -- Basic Notions & Terms

- **Global time**
  - Abstraction or representation by selected microticks of synchronized local physical clocks
  - Generates macroticks or ticks
  - **Reasonableness condition**:

    granularity g > precision Π

    => $| t^j(e) - t^k(e) | \leq 1$

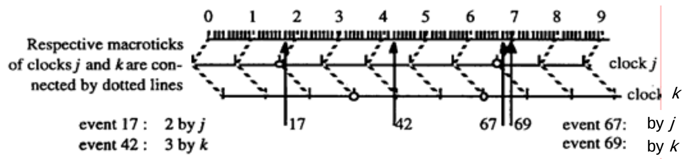


Figure 3.3: Temporal order of two events with a difference of one tick.

UCI DREAM Lab

Interval measurement

- Limited precision of interval measurement

$$(d_{obs} - 2g) < d_{true} < (d_{obs} + 2g)$$

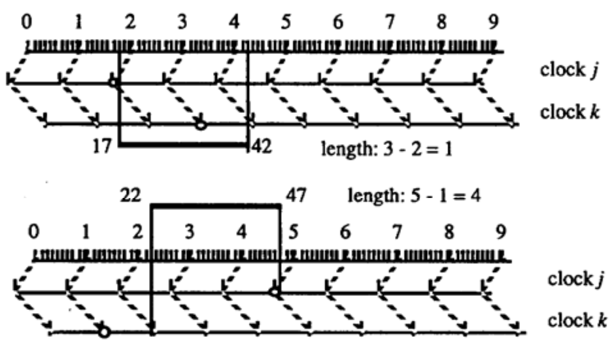


Figure 3.4: Errors in interval measurement.

UCI DREAM Lab

Dense Time-Base v.s. Sparse Time-Base

- If events are allowed to occur at any instant of the timeline, then we call the time base **dense**.
- If the occurrence of events is restricted to some active intervals of duration ε, with an interval of silence of duration Δ between any two active intervals, then we call the time base **ε/Δ-sparse**, or simply **sparse** for short.

Dense Time Base - Problem

- Event e1 is observed by node j at time 2 and by node m at time 1.
- Event e2 is only observed by node k that reports its observation "e2 occurred at time 3" to node j and node m.
- Node j calculates a timestamp difference of one tick and concludes that the two events cannot be ordered.
  Node m calculates a timestamp difference of 2 ticks and concludes that e1 has definitely occurred before e2. ==> Inconsistent view

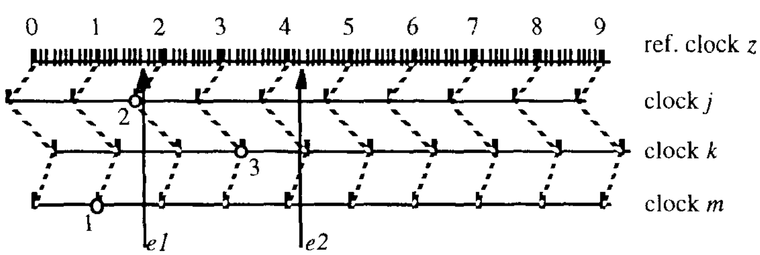
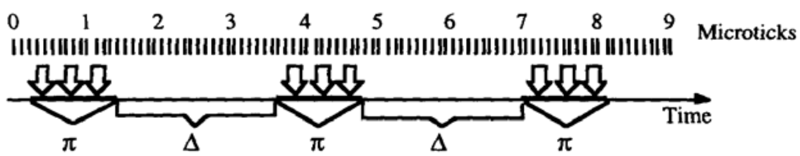


Figure 3.7: Different observed order of two events $e1$ and $e2$.

Sparse Time Base

- Events are generated in clustered manners such that each cluster is within the time-window of Pi and clusters are separated by at least Delta.
- Computing nodes based on a sparse time base will create events only within the time-windows of Pi .



Events ⇩ are only allowed to occur within the intervals π.

(=> External events, which can naturally occur at any points on a dense time line, can still be observed / mapped into points on a sparse time base inconsistently by multiple observing nodes.

- This problem can be avoided only if each external event is observed by at most one node.)

Irrelevant ?

π / Δ-Precedence

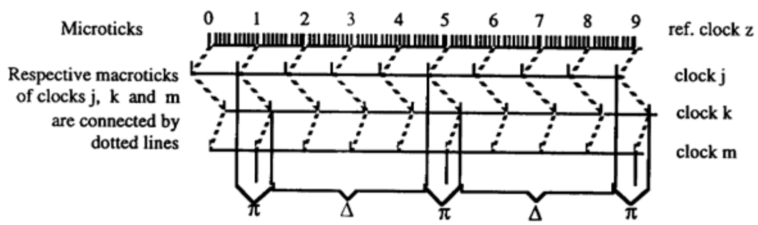
- **Consider a distributed system that consists of three node, j, k, and m.**
  - **All events that are generated locally at the same global clock tick will occur within a small interval π, where π ≤Π, the precision of the ensemble.**
  - **Events that occur at different ticks will be at least Δ apart.**
- **Given a set of events {E} and two durations π and Δ where π << Δ, such that for any two elements e_i and e_j of this set, the following condition holds:**

$$[|z(e_i) - z(e_j)| \le \pi] \vee [|z(e_i) - z(e_j)| > \Delta]$$

where *z* is the reference clock.

→ Such an event set is called π / Δ-precedent.

π / Δ-Precedence



- **How many granules of silence must exist between the event subsets so that an outside observer or another cluster will always recover the temporal order intended by the sending cluster ?**

Sparse Time-Base

- **Consider a distributed system that consists of two clusters: cluster A generates events, and cluster B observes these generated events.**
- **Assumption 1: Each one of the clusters has its own cluster-wide synchronized time with a granularity g, but these two cluster-wide time bases are not synchronized with each other.**

Bad env't considered here "?

- **Under what circumstances is it possible for the nodes in the observing cluster to reestablish the intended temporal order of the generated events?**

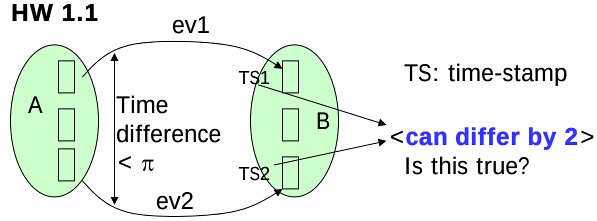
→ **Is it sufficient if cluster A always generates a 1g/3g precedent event set in order for cluster B to reestablish the temporal order of the events ?**

Sparse Time Base

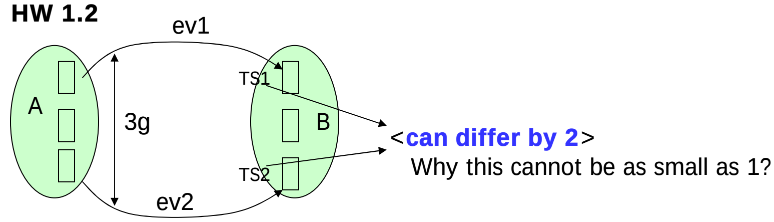
* **Assumption 2: A does not attach TSs.**
- **Given Cluster A that generates events to be observed by Cluster B, 1/3g-precedent event set generated by A cannot be properly ordered by B.**
  - **Two events generated at the same cluster-wide tick can lead to TSs by the nodes in B that differ by 2 ticks.**
  - **Two events generated 3g-apart (i.e., at different cluster-wide ticks) in A can also lead to TSs by B that differ by 2 ticks.**
  - **The cluster B cannot decide whether or not to order events with a timestamp difference of 2 ticks.**

Homework: If A attaches TSs, can 1/2g-precedent event set generated by A be properly ordered by B ?
What about 1/1g-precedent set ?

HW 1.1



HW 1.2



HW 1.3: Change Assumption 1 only, then could cluster B order the 1/3g-precedent events?

HW 1.4: Change Assumption 2 only, then could cluster B order the 1/3g-precedent events?

Sparse Time Base

- **1/4g-precedent event set generated by A can be properly ordered by B.**
  - **The cluster B will not order two events if their timestamps differ by less than or equal to 2g, but will order two events if their timestamps differ by larger than or equal to 3g.**
- **However, this argument cannot be an adequate basis for suggesting a sparse time base in Figure 3.8.**
  - **If Cluster A attaches timestamps to the events that it generates, then even 1/0g-precedent event set generated by A can be properly ordered by B.**
  - **If Cluster A does not use a global time base, then it cannot create 1/4-g precedent event set !**

Sparse Time Base

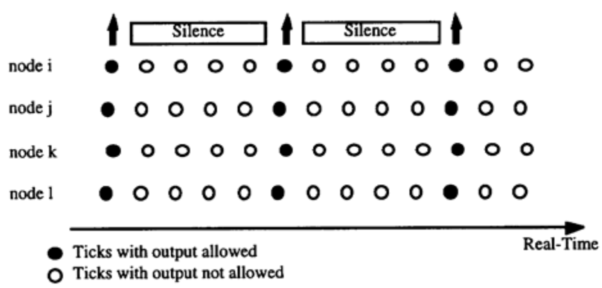


Figure 3.8: *1/4g* precedent event set.

UCI
DREAM Lab

π/Δ-Precedence

- Assume a global time base with granularity *g* and two event *e1* and *e2*, that are observed by two different nodes of the distributed system.
- The following table gives the minimum difference of the observed timestamps for differing 0/Δ-precedence.

Event Set	Observed timestamps of two non-simultaneous events are always greater or equal to	Temporal order of the events can always be reestablished
0/1g precedent	$\|t^j(e1) - t^k(e2)\| \geq 0$	no
0/2g precedent	$\|t^j(e1) - t^k(e2)\| \geq 1$	no
0/3g precedent	$\|t^j(e1) - t^k(e2)\| \geq 2$	yes
0/4g precedent	$\|t^j(e1) - t^k(e2)\| \geq 3$	yes

Internal Clock Synchronization

- **Synchronization condition**: $\Phi + \Gamma \leq \Pi$

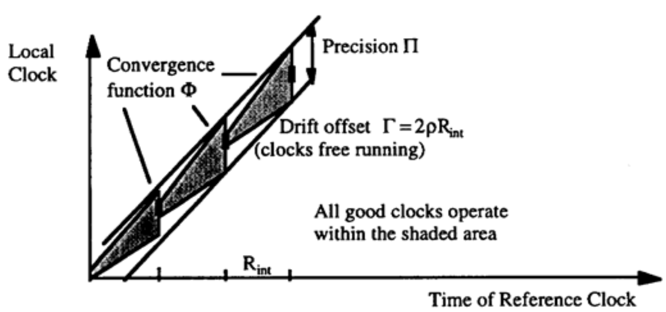


Figure 3.9: Synchronization condition.

Internal Clock Synchronization

- **Central master synchronization**
  - Latency jitter: ε

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

synchronization message assembled and interpreted	approximate range of jitter
at the application software level	500 μsec to 5 msec
in the kernel of the operating system	10 μsec to 100 μsec
in the hardware of the communication controller	less than 10 μsec

Table 3.2: Approximate jitter of the synchronization message.

- **Distributed synchronization - Fault-tolerant averaging**

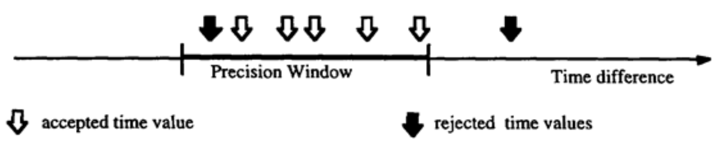
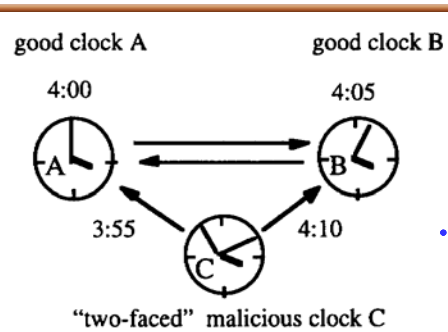


Figure 3.11: Accepted and rejected time values.

Byzantine Error - Poor example

View at A
B 4:05
A 4:00
C 3:55

- **Avg: 4:00**



View at B
A 4:00
B 4:05
C 4:10

- **Avg: 4:05**

1. **Probability of this occurring just once per any reasonable resynchronization interval may be non-negligible.**

   However, Prob of " " more than once " ", is negligible.

2. **If ECC is used in msg comm among the nodes, then even Prof of " " just once "" is negligible.**

Byzantine Error - Poor example

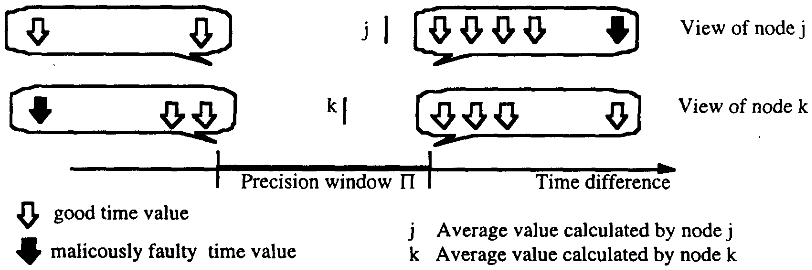


Figure 3.12: Worst possible behavior of a malicious (Byzantine) clock.

- The author recognizes that Byzantine errors are very rare and their consequences are not serious.

Rate correction is better than state correction

- Under rate correction, a clock never needs to go back !

External clock synchronization

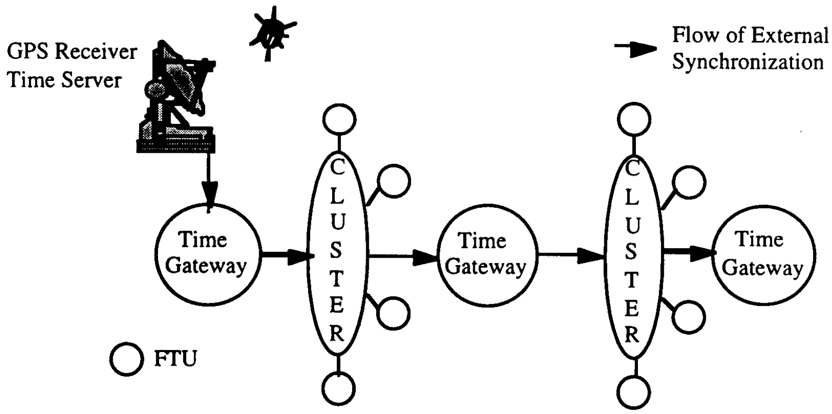


Figure 3.13: Flow of external synchronization.

UCI
DREAM Lab