Technische Universität Dresden

barkhausen institut

Time and Order

Real-Time Systems

Michael Roitzsch

Real-Time Systems Technology Stack

- Application Concept
- Timing Requirements
- Task Model
- Executable Code
- Timing Analysis
- Schedulability Analysis
- Operating System
- Resource Access
- Temporal Isolation
- Hardware Platform

Overview

following Tanenbaum/Coulouris for Logical Time and Kopetz for Physical Time

- Events, computer generated and environmental
- The order of events, temporal and causal
- Logical clocks
- Physical clocks and their properties
- Global time in distributed systems

Questions to Answer

Can clocks (logical or physical) be used …

- to derive the order of events
- to identify events
- to generate events at certain points in time?

Which precision can be achieved …

- to measure time?
- to measure durations?

How and how often should clocks be synchronized?

Basics of Time and Clocks

Events in the Real World

Environmental Events:

- mechanical processes
- emergency situations
- human interaction

Sequence of States Determined by:

- laws of physics
- physical time in seconds
- continuous

Events in Computers

Computer Generated Events:

- execution of statement
- sending / receiving a message
- start and end of a compilation
- creation / modification of a file

Sequence of States Determined by:

- instructions, memory accesses
- discrete steps

Time in Distributed Systems

Actions / Events in Distributed Real-Time Systems

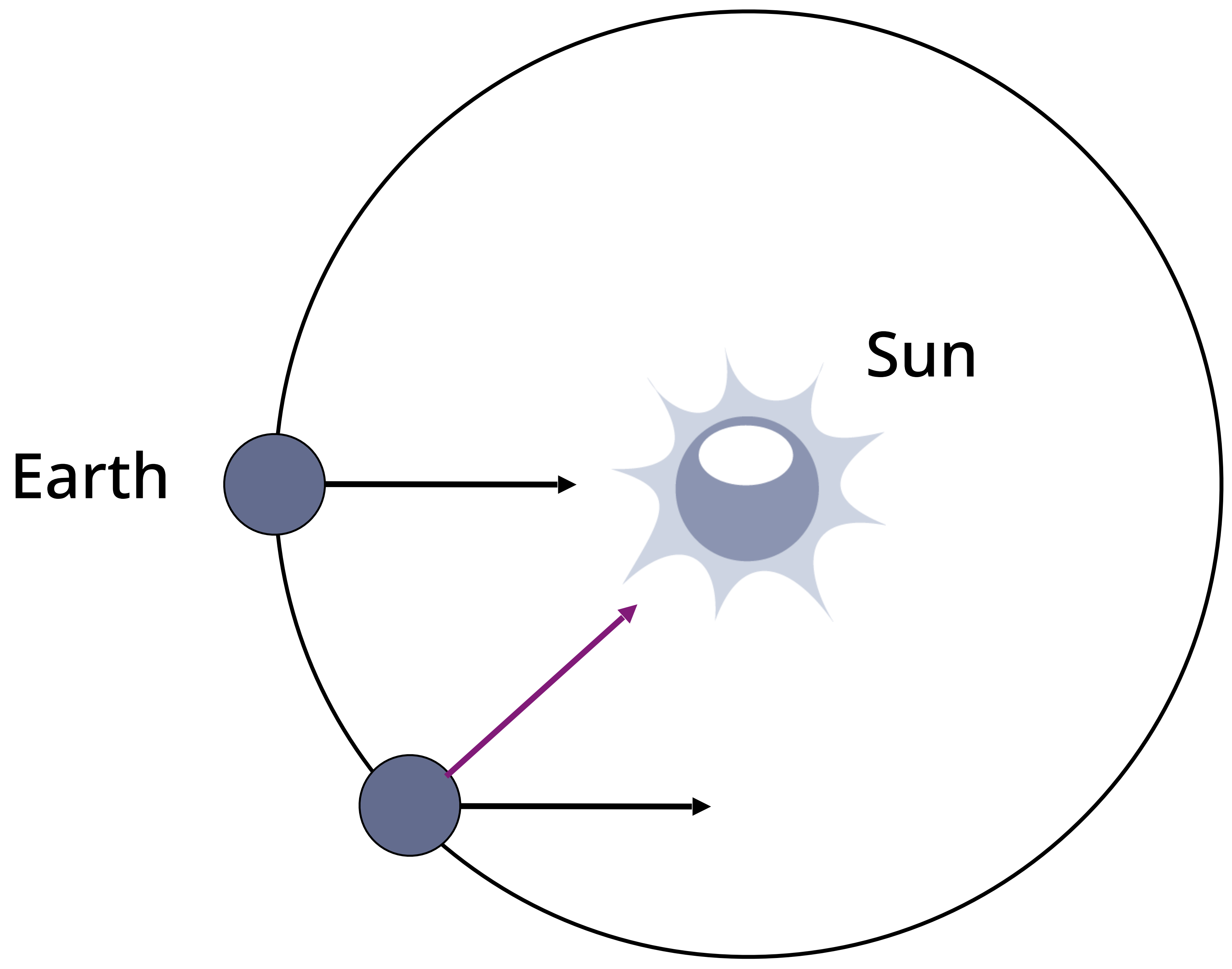
- concurrent
- on different nodes
- global time base
- must have a consistent behavior / order

Consistent Order

- temporal order
- causal order

Solar Time

- Solar Day: from noon to noon
- Solar Second: Solar Day / (24 * 60 * 60)
- Sidereal Day: 360° rotation, distant stars in same position



Time Standards

TAI: International Atomic Time

- 1 second = duration of 9192631770 (9 Gigahertz) periods of the radiation of a specified transition of the caesium atom 133
- TAI is a weighted average of 450 atomic clocks worldwide
- GPS Time is based on onboard atomic clock in the satellites
- continuous, without leap seconds

UTC: Coordinated Universal Time

- agreed-upon leap seconds to match human perception of years

Timestamps, Durations, Clocks

Timestamp

Events occur at an instant of the timeline

Duration

Duration is a section of the timeline

Clock

Clocks measure time imperfectly, creating imperfect timestamps

Temporal vs. Causal Order of Events

Temporal Order

induced by (perfect) timestamp

Causal Order

induced by some causal dependency between events

Example

- event 1: somebody enters a room
  event 2: the telephone rings
- e1 occurs after e2: causal dependency possible
  e2 occurs after e1: causal dependency not possible
- temporal order is necessary but not sufficient to establish causal order

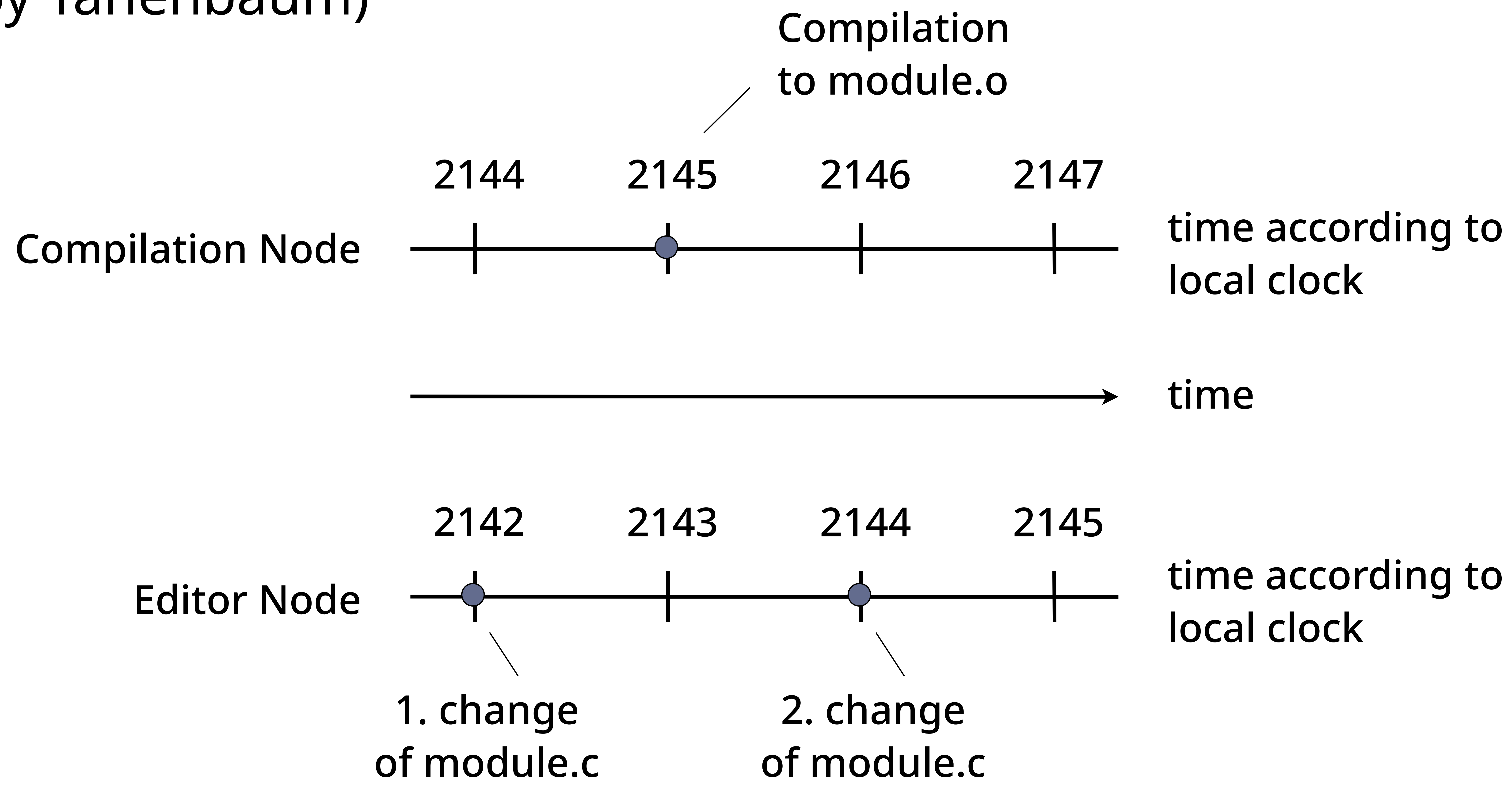
Temporal Order

Modeling the Continuum of Time:

- infinite set of instants {t}
- {t} is totally ordered:
  if p, q are any 2 instants, then either
  - p, q are simultaneous (the same instant)
  - or p precedes q
  - or q precedes p
- {t} is dense:
  iff p and q are not simultaneous → instant r between p and q exists

Imperfect Timestamps

imperfect Timestamps can be misleading in establishing causal dependency (example by Tanenbaum)



Causal Order

Partial Order for Computer Generated Events

a → b 'a causes b'
(happened before, causally dependent)

- if a, b are events within a sequential process, then:
  a → b if a is executed before b
- if a is 'sending of a message' event by a process and b is 'reception of that message' event by another process, then:
  a → b
- → is transitive

Logical Clocks and Global Time

Physical Clocks

- devices to measure time
- necessarily imperfect

Questions:

- How to create knowledge about causal dependency of computer events without relying on physical clocks? → Logical Clocks
- How to establish a ‘x certainly occurred after y relation’ (temporal order) for environmental events? → Global Time

Logical Clocks

Logical Clocks

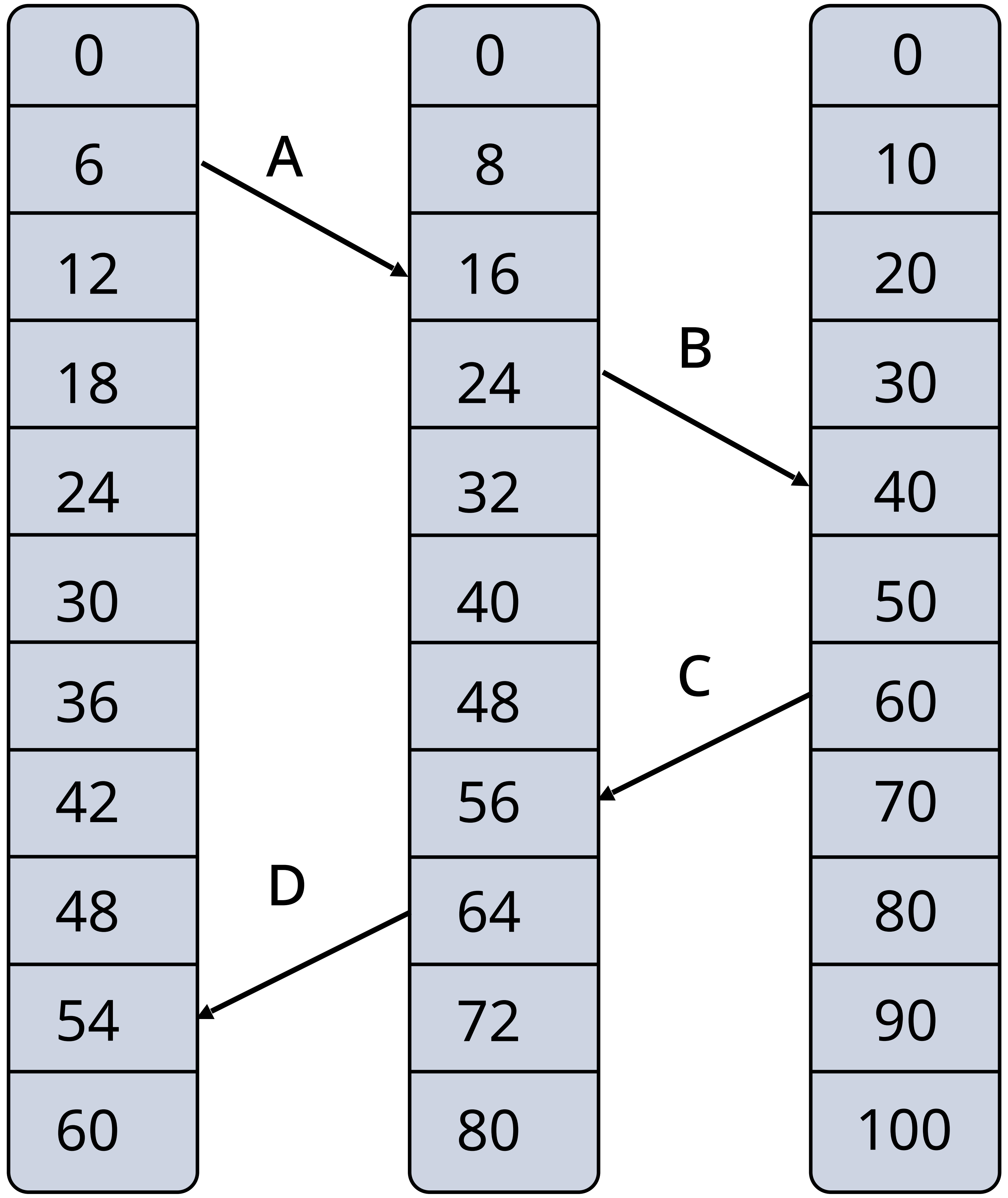
Definitions:

- monotonically increasing software counters (Coulouris)
- clocks on different computers that are consistent (Lamport)
- events a, b: $a \rightarrow b$: a causes b (causally dependent)
- timestamps: $C(a)$, $C(b)$
- potential requirements for logical clocks:

  $a \rightarrow b \;\Rightarrow\; C(a) < C(b)$

  $a \rightarrow b \;\Leftrightarrow\; C(a) < C(b)$

Logical Clocks Example



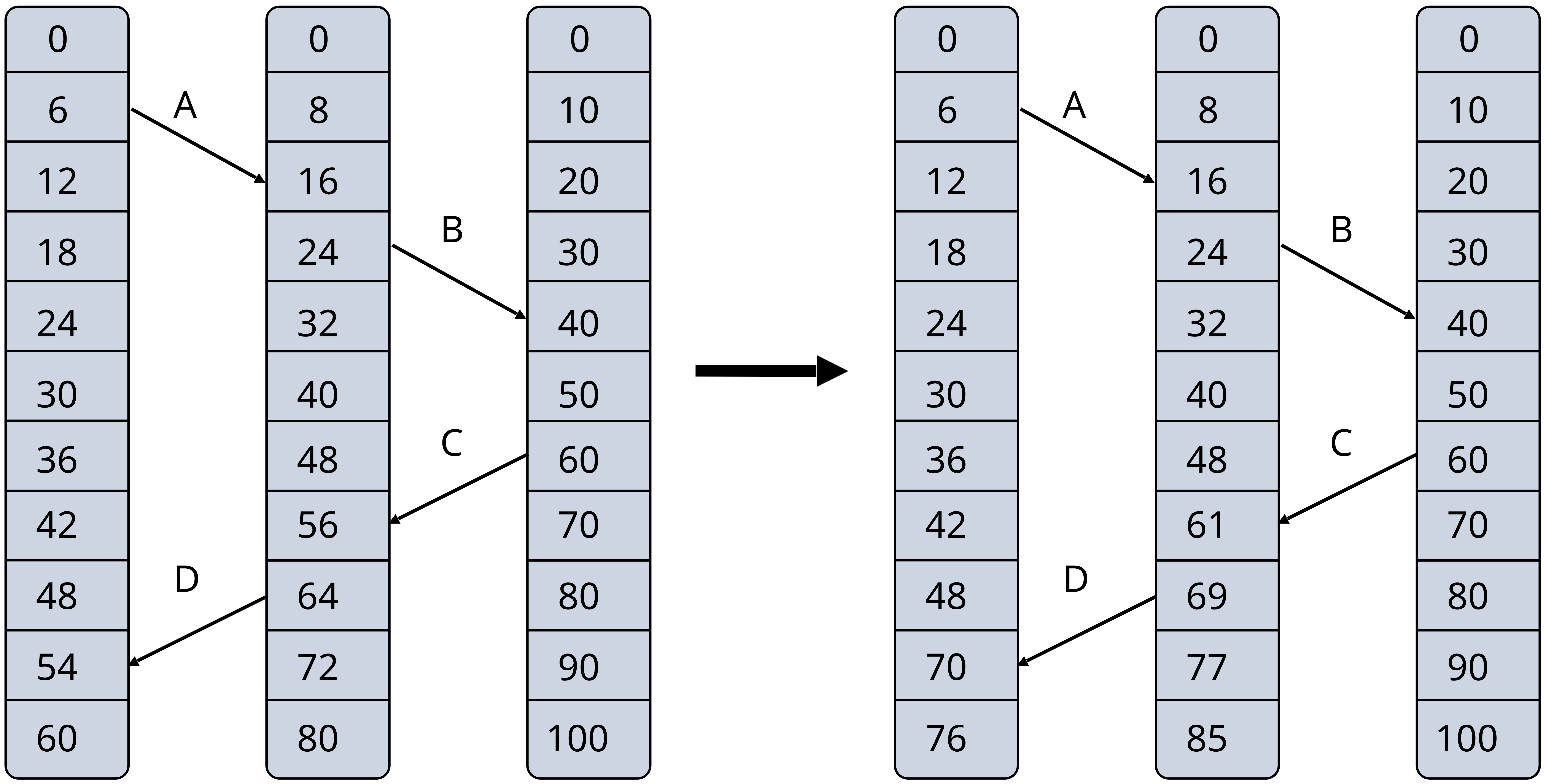
Lamport Clocks

each Process has local clock LC_i

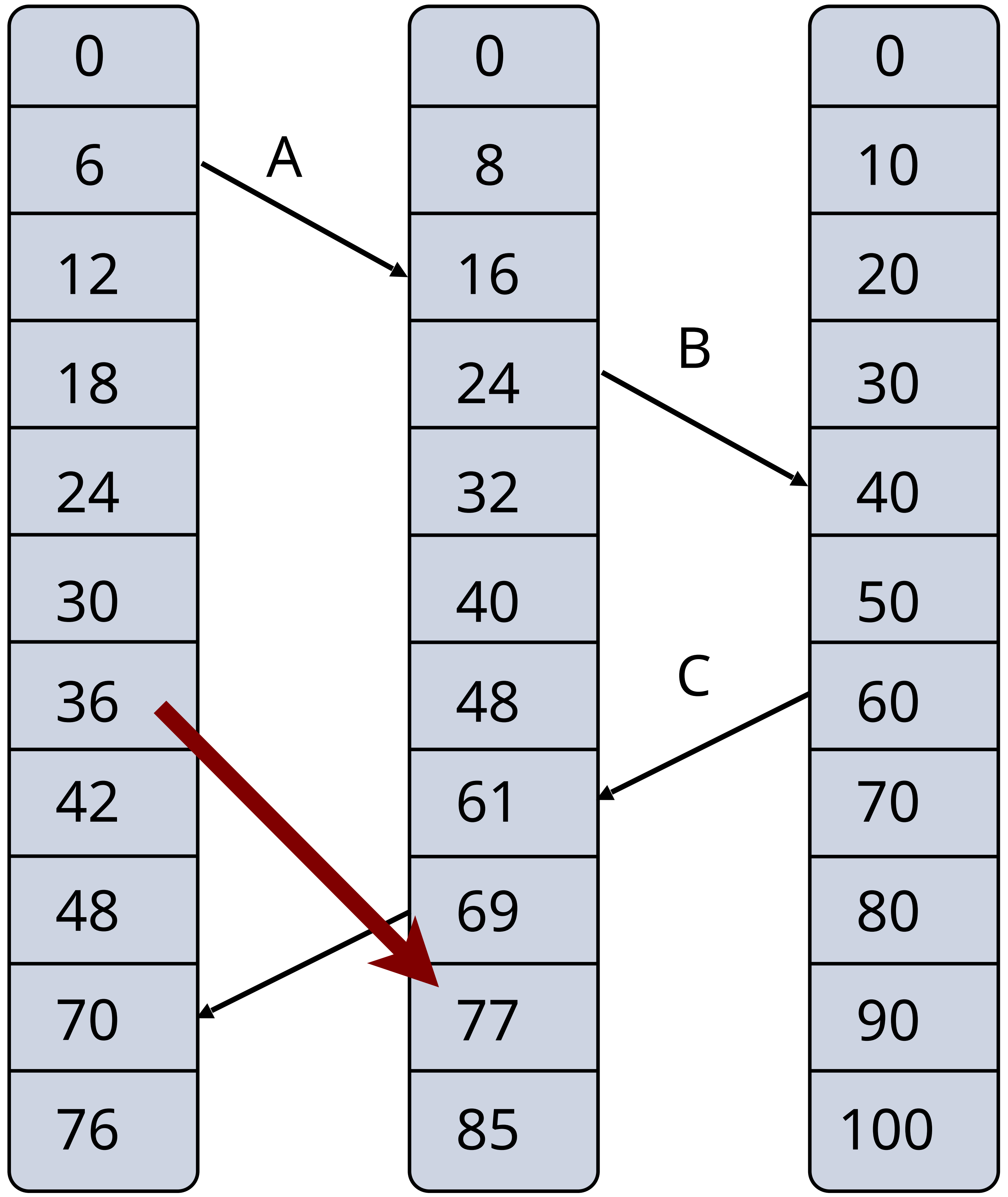
Ticks

- with each local event e:
  $LC_i := LC_i + 1$; e
- with each sending of a message by process P_i:
  $LC_i := LC_i + 1$; send(m, LC_i)
- with each reception of a message (m, LC_m) by P_j:
  $LC_j := \max(LC_m, LC_j)$; $LC_j := LC_j + 1$

Lamport Clocks Example



Partial Timestamp Order



Lamport Clocks

Properties

- $a \rightarrow b \Rightarrow C(a) < C(b)$,
- but not:
  $C(a) < C(b) \Rightarrow a \rightarrow b$
- Lamport Clocks establish a partial order relation

Vector Clocks — *Mattern 1989*

- Each process P_i has its own vector clock C_i
- C_i: n-dimensional vector
- n: number of processes
- Intuition: $C_i[j]$ is the timestamp of the last event in P_j by which P_i has potentially been effected

Vector Clocks

Ticks

- Initial:

  $C_i := (0, \ldots, 0)$ for all i

- Local event in P_i:

  $C_i[i] := C_i[i] + 1;$

- Sending message m in P_i:

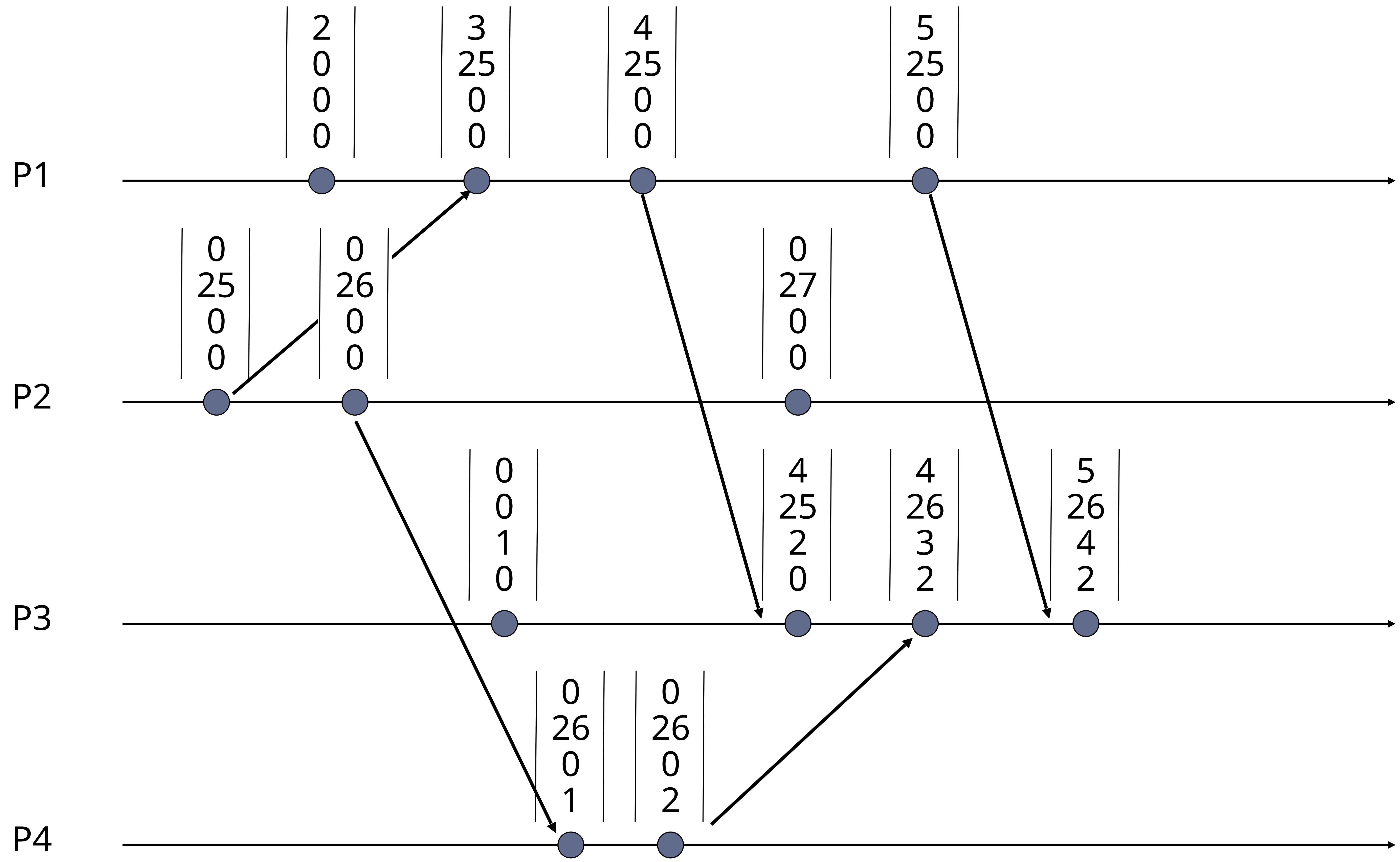
  $C_i[i] := C_i[i] + 1;$ send(m, C_i)

- Receiving a message (m, C_m) in P_j:

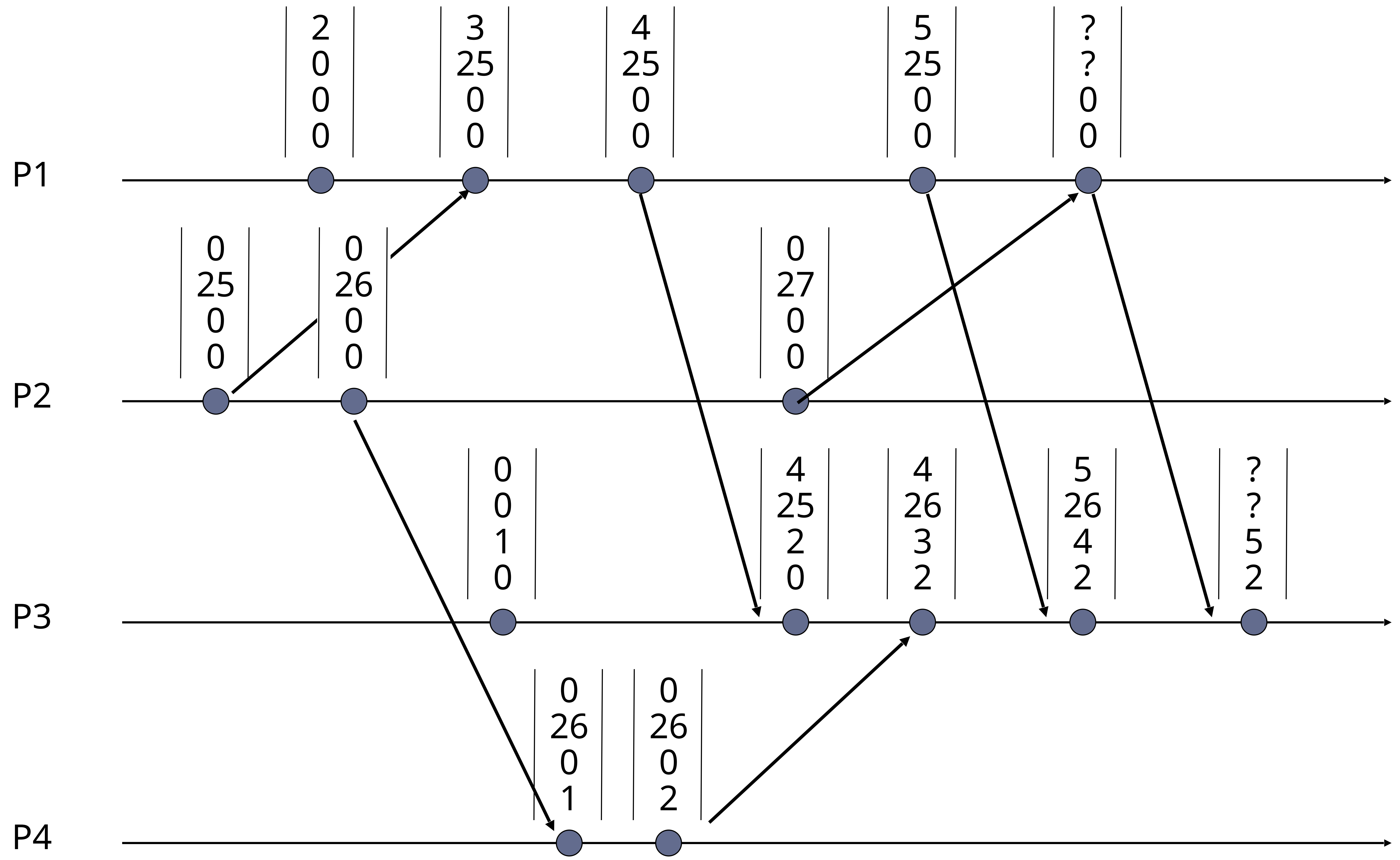
  $C_j[j] := C_j[j] + 1;$

  $C_j[k] := \max(C_m[k], C_j[k])$, for all k

Vector Clocks Example



Vector Clocks Example



Properties of Vector Time

Definitions

- $C_a \leq C_b :\Leftrightarrow \forall k: C_a[k] \leq C_b[k]$
- $C_a < C_b :\Leftrightarrow C_a \leq C_b \wedge C_a \neq C_b$
- $C_a \parallel C_b :\Leftrightarrow C_a \nless C_b \wedge C_b \nless C_a$

Property

- $C_a < C_b \Leftrightarrow a \rightarrow b$

Physical Clocks

Properties of Physical Clocks

Physical Clock

- device for measuring time
- counter + oscillator → microtick
- interval between microticks has the same time reading:
  granularity leads to digitalization error

Notation

- $\text{microtick}^{\text{clock}}_{\text{number}}$
- to discuss properties of physical clocks, we invent a perfect reference clock
  as a purely theoretical construct

Reference Clock Notation — *Daum*

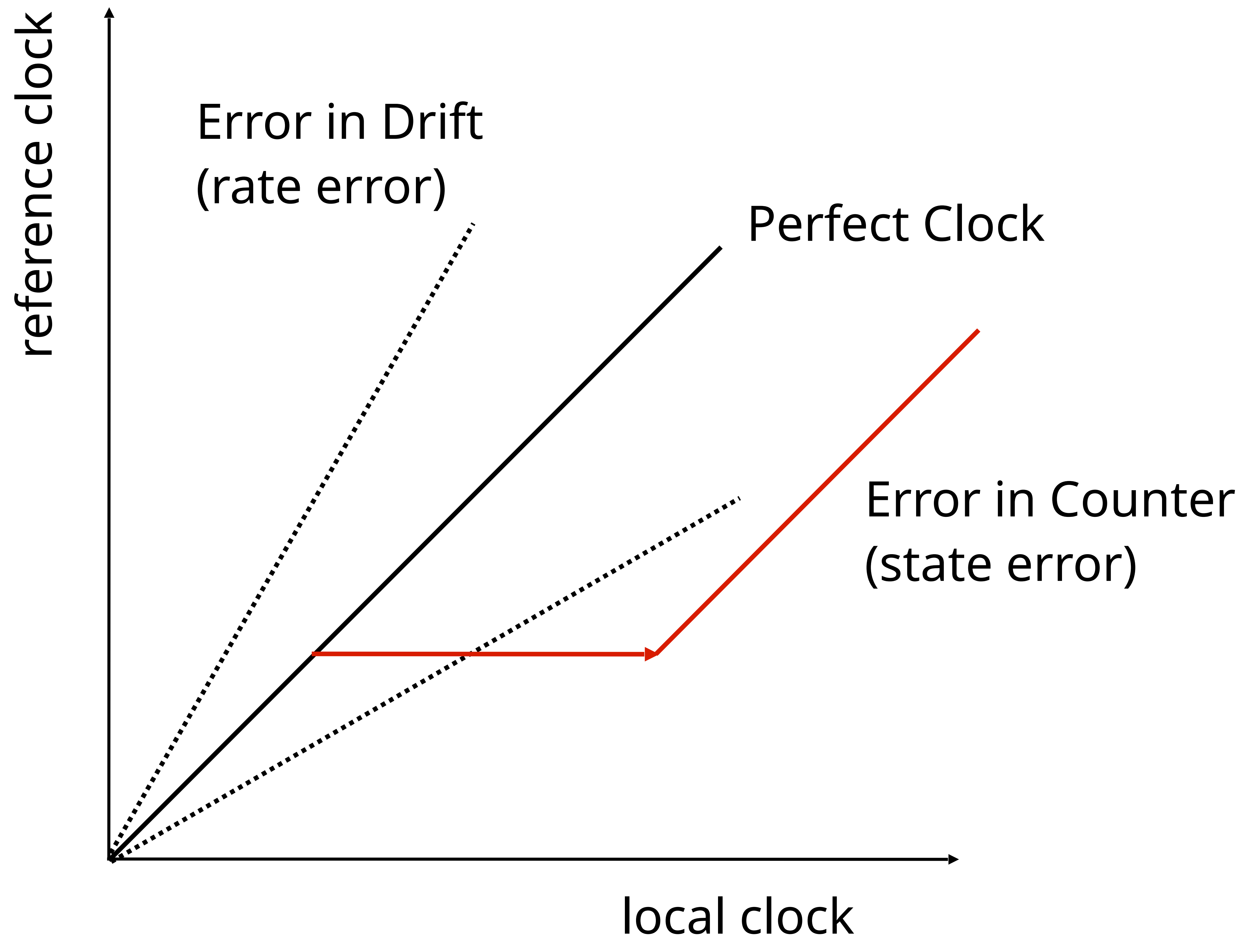
Reference Clock z

- perfect with regard to atomic time, continuous
- dense: no ticks, to avoid digitalisation error
- z(event): absolute timestamp from reference clock establishes temporal order
- g^k granularity of a clock k in terms of (intended) z-duration

Clock Terminology

- microticks — ticks generated by the physical oscillator of a clock
- macroticks — multiple microticks chosen by designer of clock
- t^k(event) — timestamp in number of microticks of clock k

Failure Modes: Drift and Counter Errors



Single Clock Property: Drift

Drift Rate

- unit-less measure
- varying, influenced by environmental conditions: temperature, ...

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

- clocks specify maximum drift rate (10^{-2} ... 10^{-7}):

Ensemble of Clocks Property: Precision

Offset

- between two clocks j,k of same granularity at microtick i:

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

- in the period of interest: $$offset^{jk} = \max_i(offset_i^{jk})$$

Precision

- of an ensemble of clocks {1,2, … ,n} in the period of interest:

$$\Pi = \max_{1 \le j,k \le n}(offset^{jk})$$

- maximum offset for any two clocks within an ensemble

Single Clock Property: Accuracy

Accuracy

- of a given clock in the period of interest:

$$accuracy^k = \max_i \left| z(microtick_i^k) - i \cdot g^k \right|$$

- maximum offset to reference clock
- If all clocks of an ensemble have accuracy A, what is the precision of the ensemble?

Clock Synchronisation

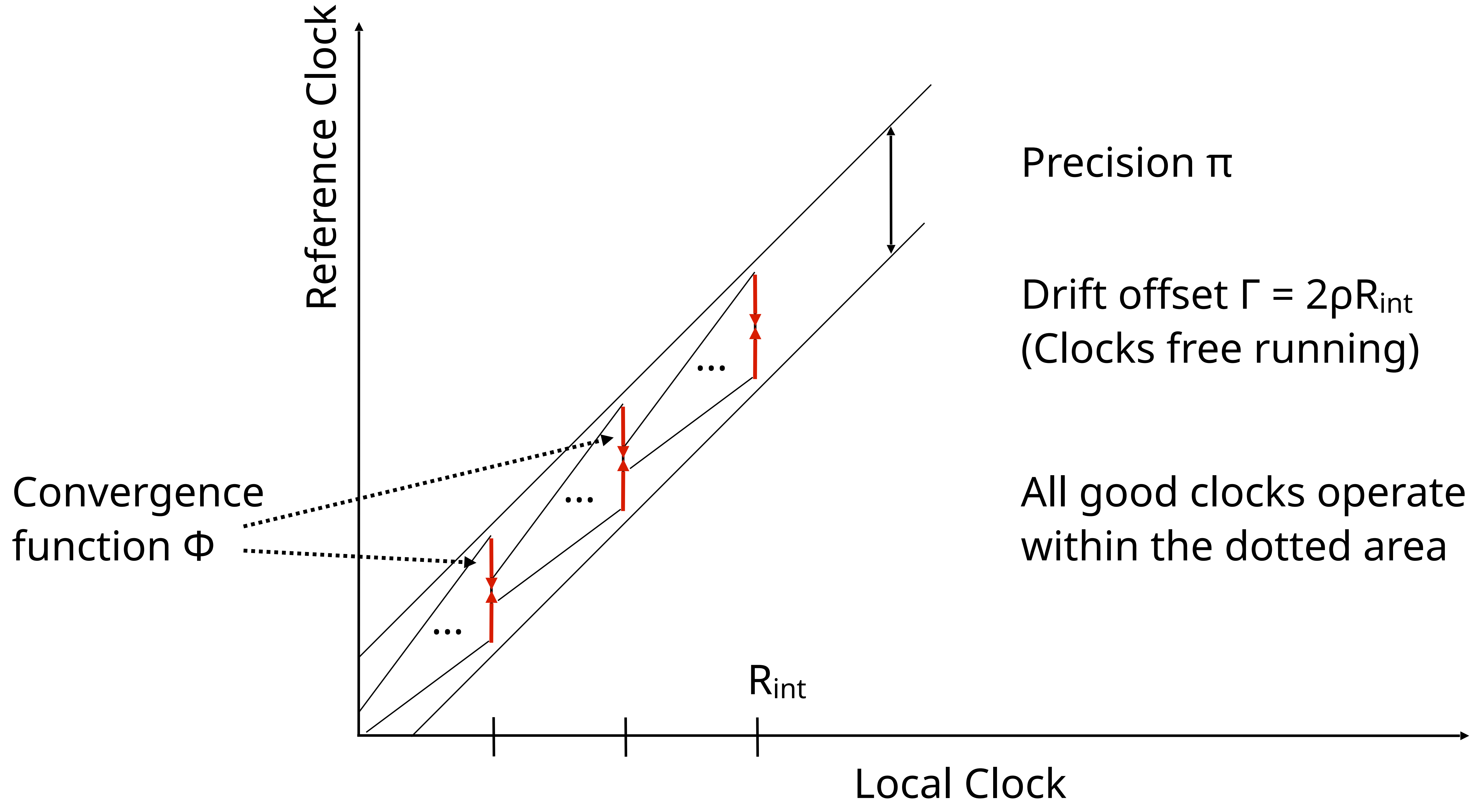
External Synchronization

- synchronization with reference clock
- to maintain bounded accuracy

Internal Synchronization

- mutual synchronization of an ensemble
- to maintain bounded precision

Clock Synchronisation



Precision π

Drift offset $\Gamma = 2\rho R_{int}$
(Clocks free running)

All good clocks operate
within the dotted area

$$\Phi + \Gamma = \Pi$$

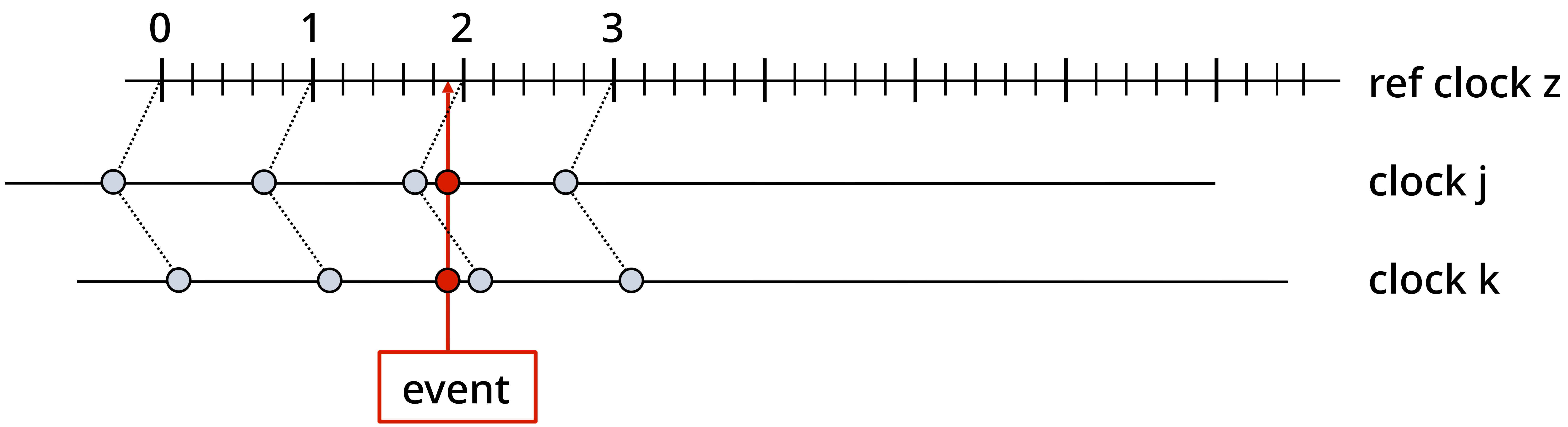
Synchronisation Condition

- (re)synchronization interval: R_{int}
- convergence function : Φ offset after (imperfect) synchronisation
- drift offset: $\Gamma = 2\rho R_{int}$
- Required: $\Phi + \Gamma = \Pi$

Global Time

Global Time

- given an ensemble of clocks (internally) synchronized with precision π
- for each clock: select macrotick as local implementation of a global notion of time with granularity g^{global}
- we also note reference clock time in units of g^{global}

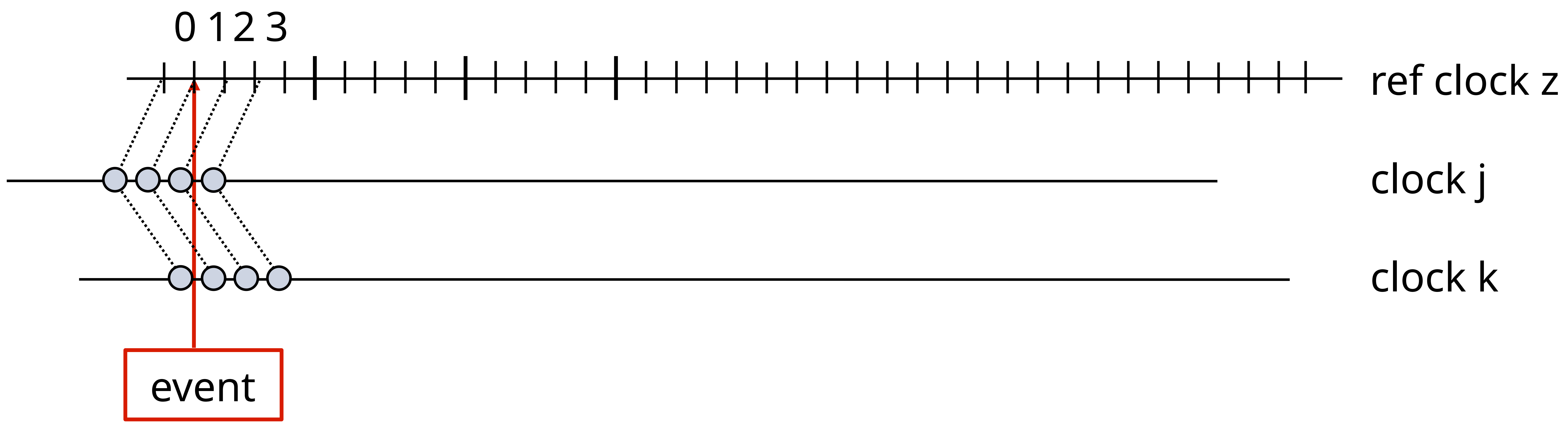


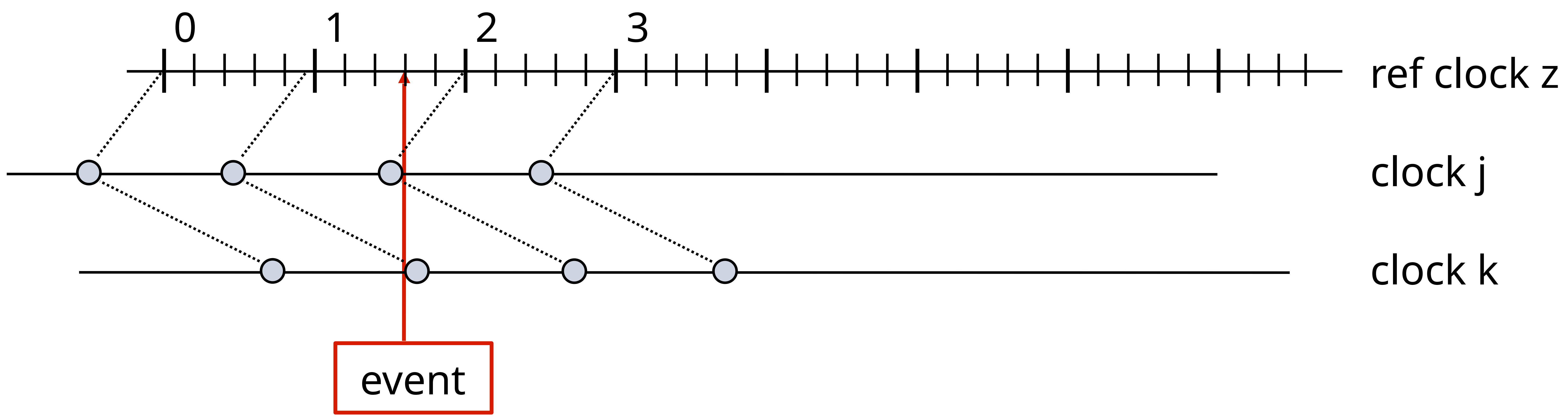
One Tick Difference

Reasoning Condition:

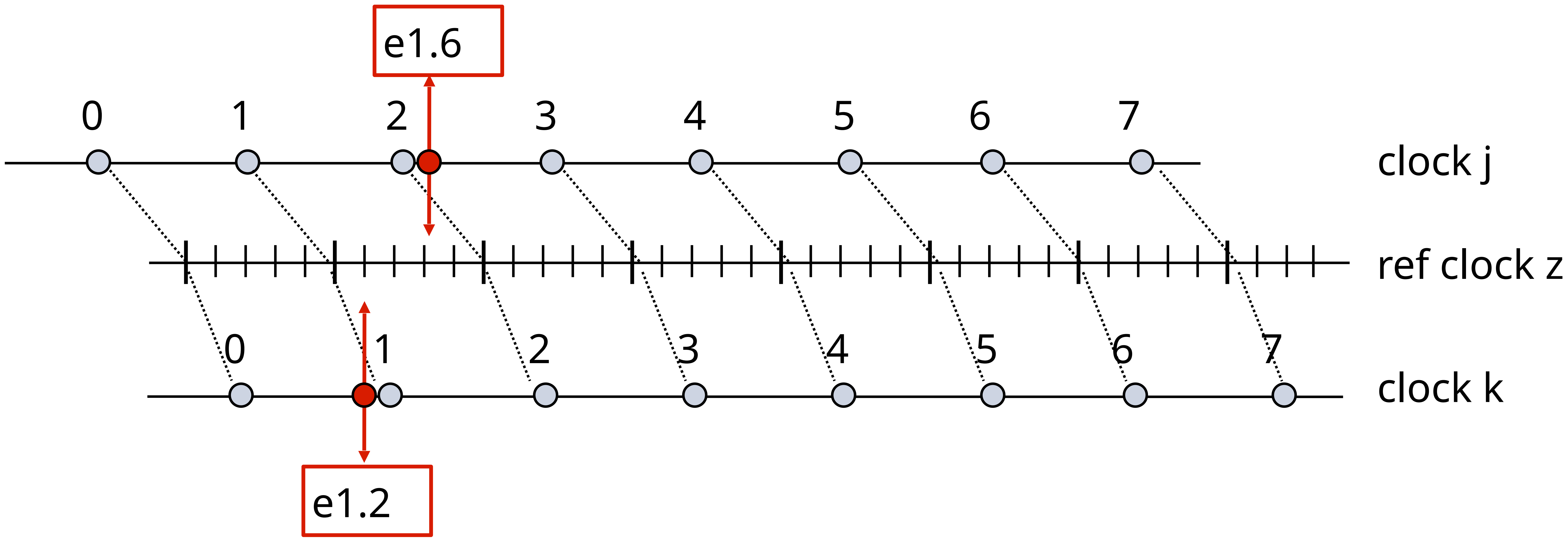
- global time t is reasonable if
  $\pi < g^{global}$ holds for all local implementations
- t^j(event): denotes global timestamp for event at clock j
- then, for any single event e: $\left| t^j(e) - t^k(e) \right| \leq 1$
- global timestamps of the same event differ at most by one macrotick
- This is the best we can achieve!

Bad Choice for Global Time





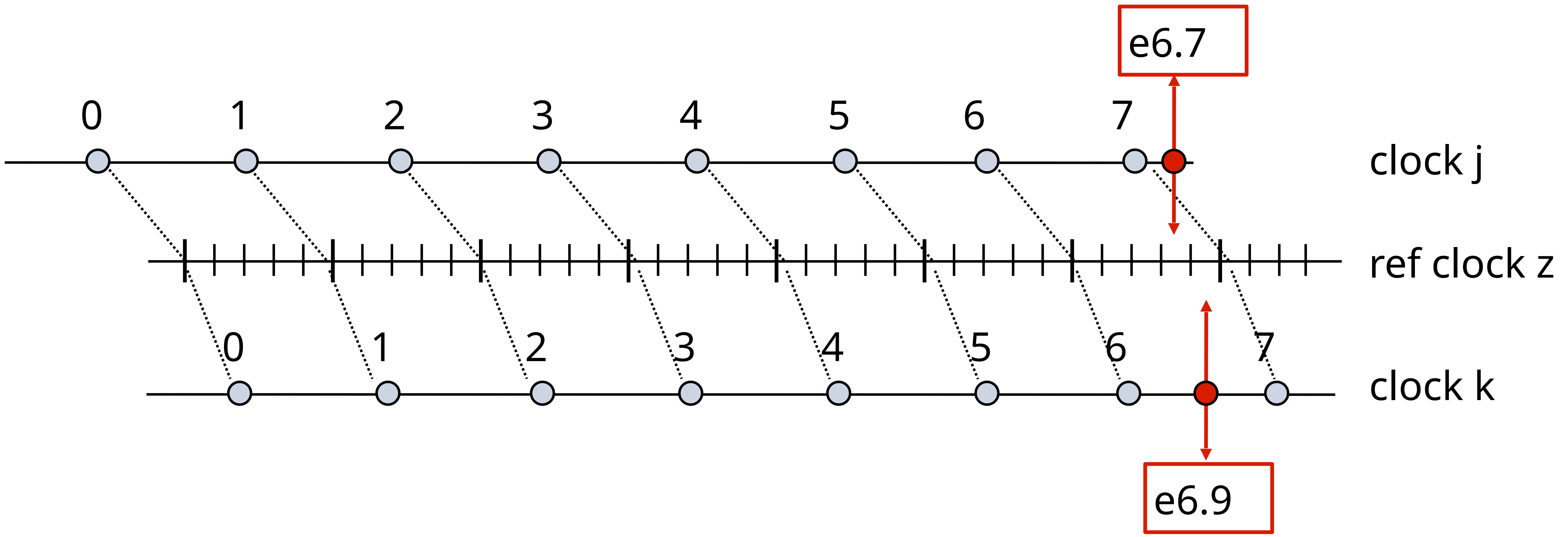
Necessary Condition to Establish Order



- z(e1.6) – z(e1.2): 0.4 reference clock
- t^j(e1.6) – t^k(e1.2): 2 global time ticks
- temporal order can be established because
  tick seen by k must be before tick seen by j (due to $\pi < g^{global}$)

If timestamps differ by two ticks, temporal order can be established.

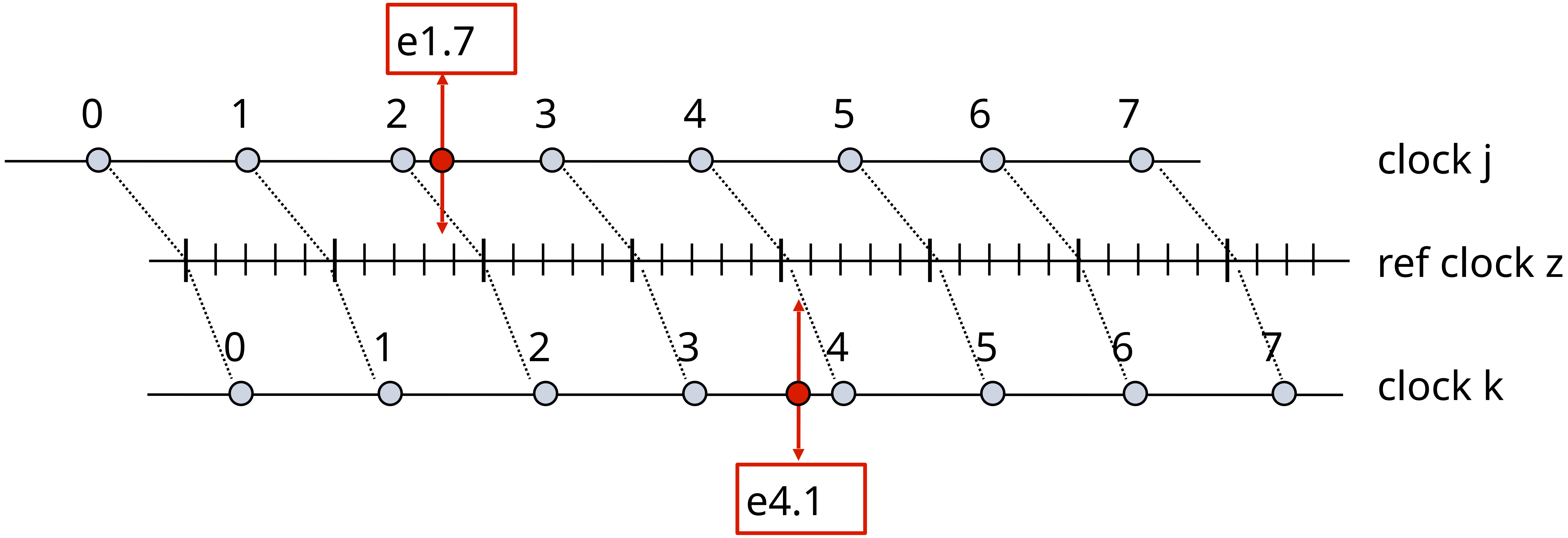
Counter Example for Close Events



$z(e6.9) > z(e6.7)$

$t^k(e6.9) < t^j(e6.7)$

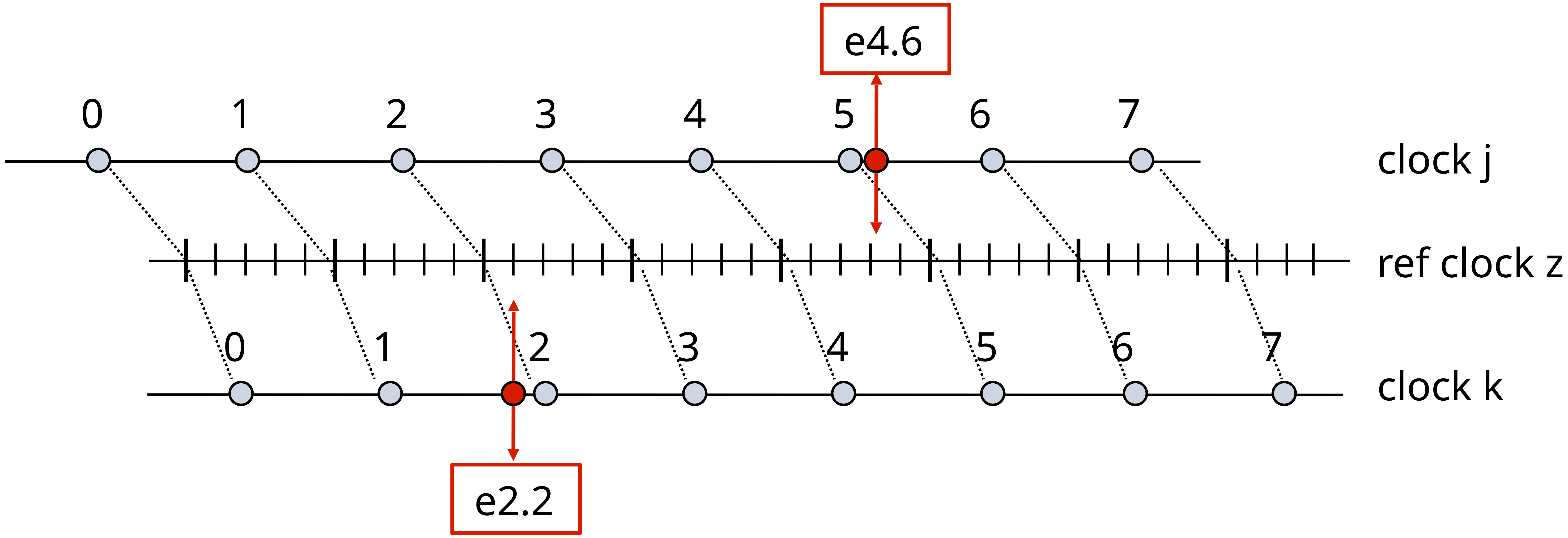
Sufficient Condition to Establish Order



- $z(e4.1) - z(e1.7)$: 2.4 reference clock
- $t^k(e4.1) - t^j(e1.7)$: 1 global time ticks

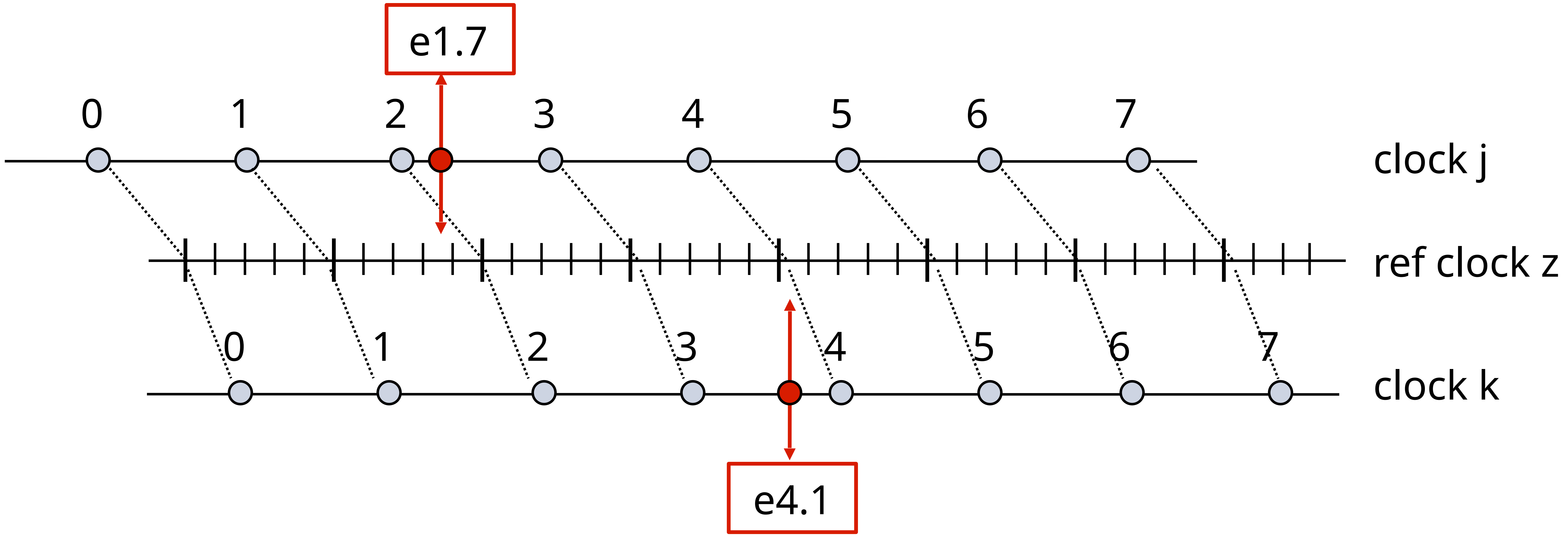
A distance of $2 \times g^{global}$ between two events is not sufficient to reliably establish temporal order. A distance of $3 \times g^{global}$ is sufficient.

Interpretation for Durations



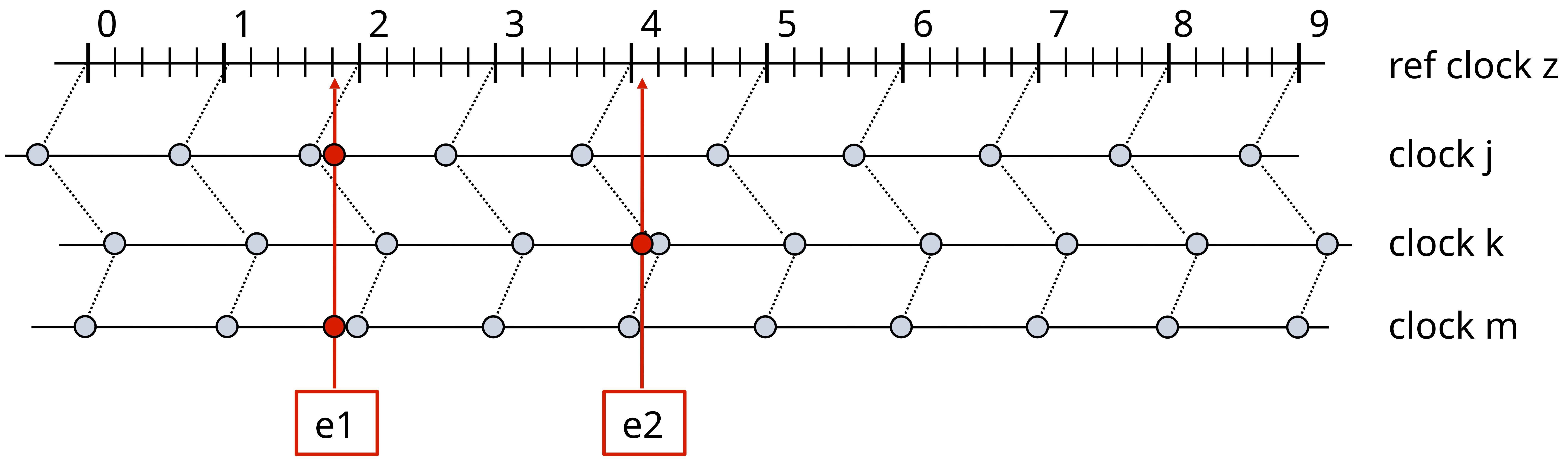
- true duration: 2.4
- observed duration d: $t^j(e4.6) - t^k(e2.2) = 5 - 1 = 4$
- extreme case: true duration can become 2 + ε (for small ε), while observed duration remains 4
- $d^{obs} - 2 \times g^{global} < d^z$

Interpretation for Durations



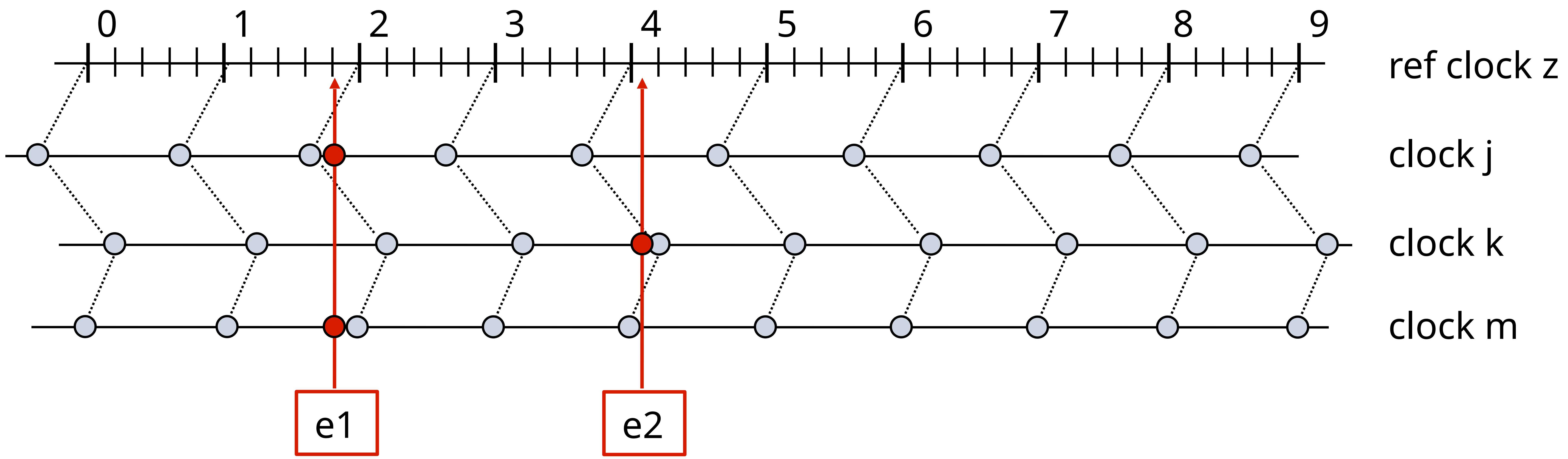
- true duration: 2.4
- observed duration d: $t^k(e4.1) - t^j(e1.7) = 3 - 2 = 1$
- extreme case: true duration can become 3 – ε (for small ε), while observed duration remains 1
- $d^{obs} - 2 \times g^{global} < d^z < d^{obs} + 2 \times g^{global}$

Cooperation and Clocks



- only nodes j and m can observe e1
- only node k can observe e2
- node k tells nodes j and m about e2
- nodes j and m draw their conclusions …

Dense Time Requires Agreement



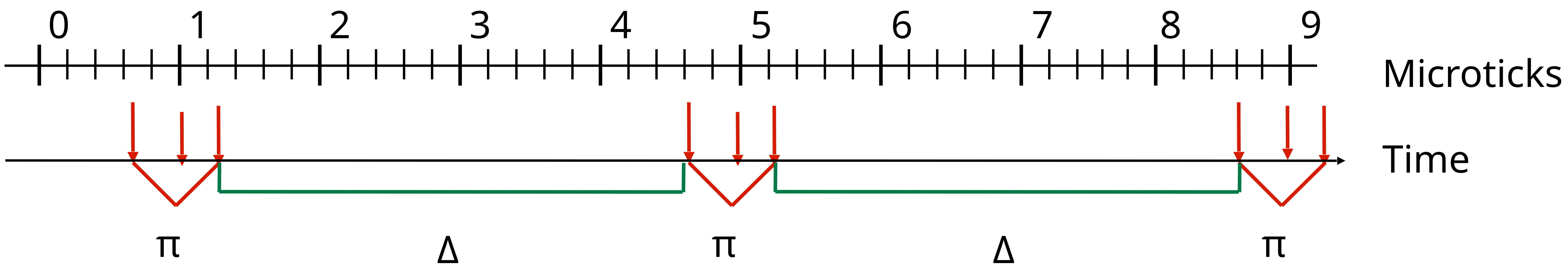
- j observes e1 at t=2, m observes e1 at t=1
- k observes e2 and reports to j and m: e2 occurred at t=3
- j calculates a time difference of 1: events cannot be ordered
- m calculates a time difference of 2: events can be ordered

Agreement Protocols

- information interchange:
  each node acquires local views from all other nodes
- deterministic consensus algorithm that leads to same result on all nodes
- expensive

Sparse Time

Dense Time vs. Sparse Time

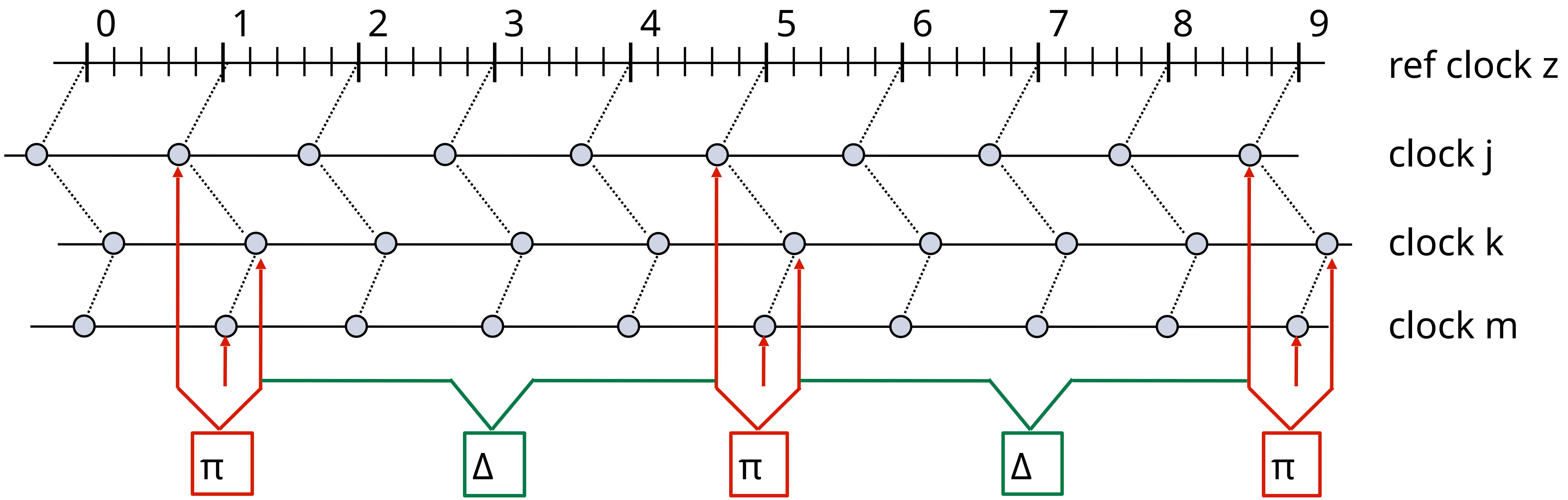


- dense time: events are allowed at any time
- sparse time: events are only allowed within active time intervals π
- sparse time only possible for computer controlled events
- goal: all nodes agree on the same outcome of ordered/not ordered

π/Δ-Precedence of Sets of Events

Properties of sets of events:

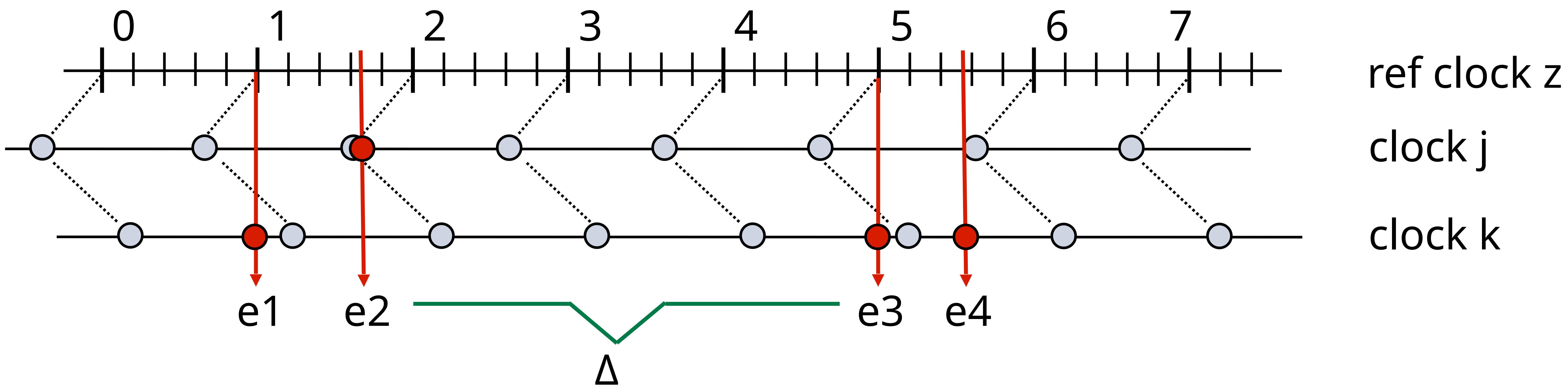
- How far apart must events be to enable reconstruction of order?
- a set of events is called π/Δ-precedent, if: $\left(\left|z(e_i) - z(e_j)\right| \leq \Pi\right) \vee \left(\left|z(e_i) - z(e_j)\right| > \Delta\right)$



Temporal Order

Event Set	Observed timestamps of two nonsimultaneous events are always greater or	Temporal order of the events can always be reestablished
0/1g precendent	$\left\| t^j(e_1) - t^k(e_2) \right\| \geq 0$	no
0/2g precendent	$\left\| t^j(e_1) - t^k(e_2) \right\| \geq 1$	no
0/3g precendent	$\left\| t^j(e_1) - t^k(e_2) \right\| \geq 2$	yes
0/4g precendent	$\left\| t^j(e_1) - t^k(e_2) \right\| \geq 3$	yes

Example for 1g/3g



- $t^j(e2) - t^k(e1) = 2$:
  should not derive order because events are intended for the same time
- $t^k(e4) - t^j(e2) > 2$ but: $t^k(e3) - t^j(e2) = 2$:
  temporal order is intended ($\Delta > 3g$), but we cannot distinguish this case from the case above
- 1g/3g precedence not sufficient → 1g/4g required
- further results in literature

Fundamental Results in Time Measurement

- a single event observed at different nodes may have timestamps differing by one → not sufficient to establish temporal order
- temporal order can be recovered from their timestamps if they differ by 2 global time ticks
- temporal order of events can always be recovered from timestamps if the event set is 0/3g precedent (distance between events is $\geq 3g^{global}$)
- duration: $d^{obs} - 2 \times g^{global} < d^{z} < d^{obs} + 2 \times g^{global}$