21 Feb 2022

Termination Detection (TD)

Using [Distributed Snapshots]

Process P_i → active

→ idle

Logical clock

→ No physical clock

($clock_p$)

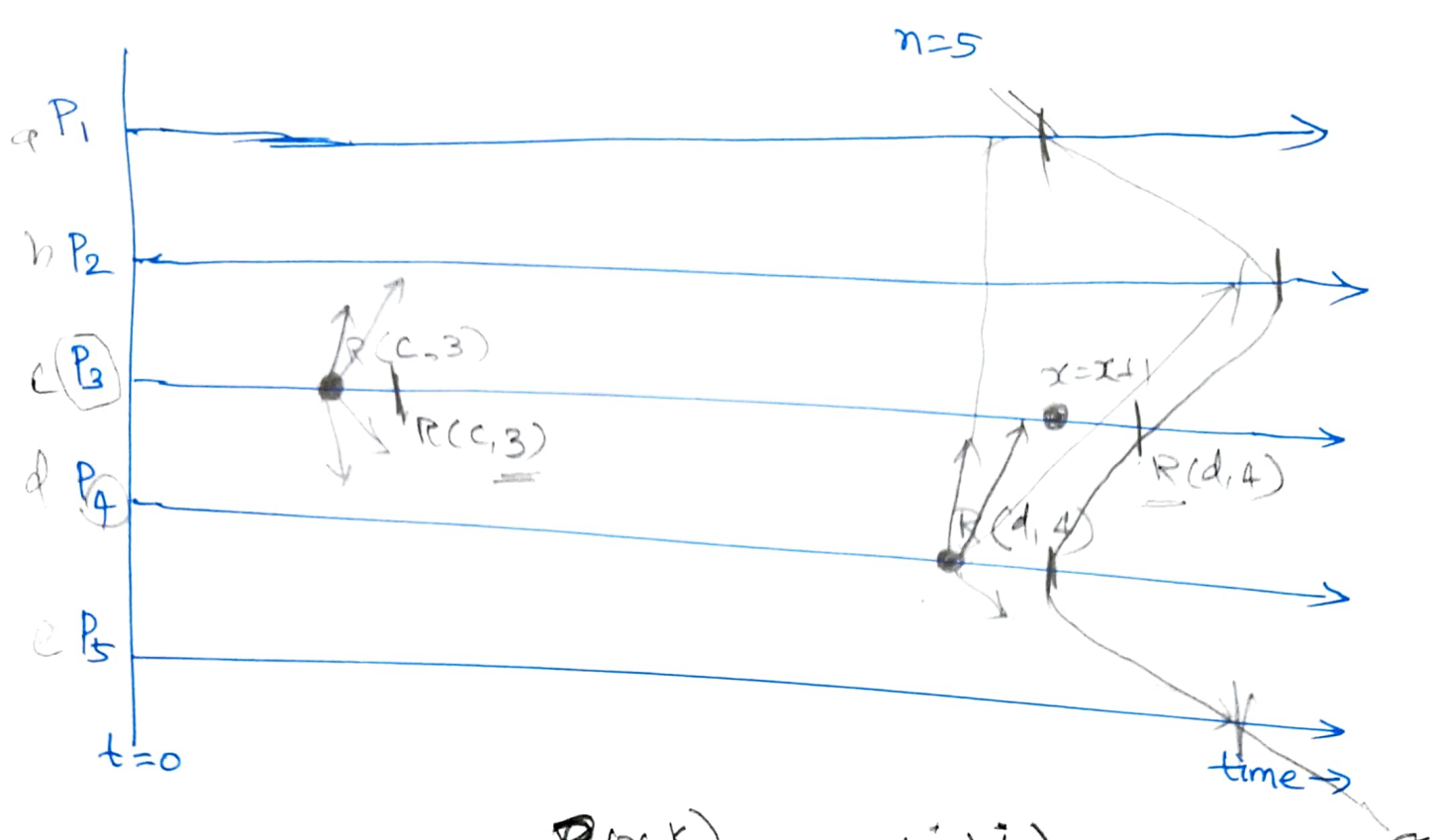
4 Rules

(R1) P_i is active P_i may send an application msg to P_j (C_{ij})

(R2) when P_i receives $B(x)$:

(R3) P_i goes idle

- send $R(x, m)$ to all P_j ($j \neq i$)
- take snapshot (local) for the app to $R(x, P_i)$



(R4) when P_i receives $R(x', k')$ from P_j $(j \neq i)$:

$P_i \rightarrow$

* $[[(x', k') > (x, k)] \wedge (i \text{ is idle})]$

$\rightarrow (x, k) = (x', k') + 1$

Take a local snapshot for $R(x', k')$

* $(x', k') \leq (x, k) \wedge (i \text{ is idle})$

$\rightarrow$ do nothing.

$(i \text{ is active}) \Rightarrow x = \max(x, x')$

P_i $(i = 3)$ (c) sends $(n-1)$ replies:

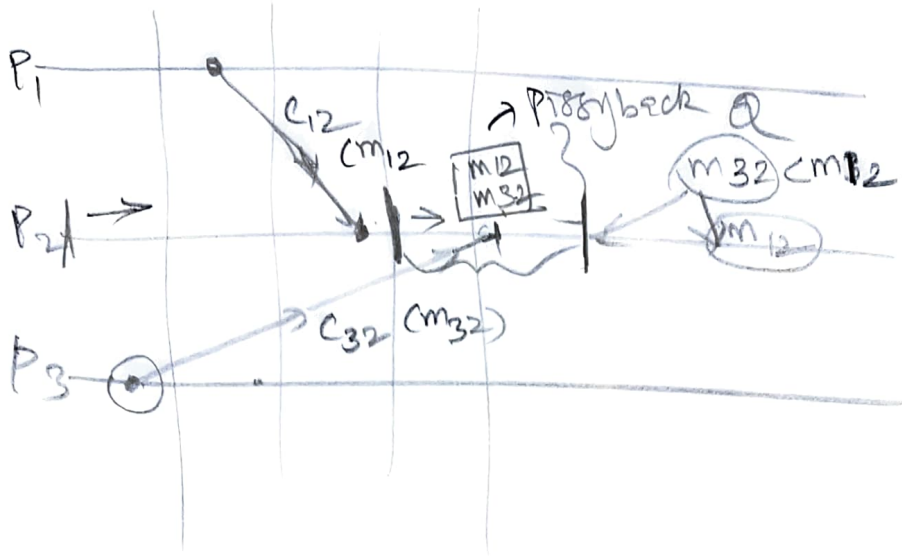
- P_1 (a): $a' = a + 1$;
- P_2 (b)
- P_4 (d)
- P_5 (e): $e' = e + 2$

$c = \max(a', c)$
$c = \max(c, b)$
$c = \max(c, d)$
$c = \max(c, e')$

$n = 5$, $(n-1) = 4$

$P_i(x) \leq (n-1)$ PE's.

$\Rightarrow$ P_i will always have the largest clock value at the end of the execution


P1
P2
P3
C12 (m12
m12
m32
piggyback
m32 (m12
m12
C32 (m32)