Termination Detection
(TD)

Assumption: $\boxed{n}$ – is known ie prior.

$l=0$ $l=1$ $l=2$ $r=2$ $r=1$ $r=0$

P_1 $r=4$ $r=3$ $l=3$ $l=4$ P_6

     P_2 P_3 P_4 P_5

(P_2) C_{12} → (P_8) (P_{11}) (P_{14}) (P_{15}) (P_{19})

P_a ← C_{21} P_b P_c P_e P_d P_f

(6) **Messages:** (6) ← (6) (6) → (6) (6)

          → Apply $\boxed{\text{Count}} < n-1$

          → Marker.

                           ID15

          ID14   −1   +1   I 19
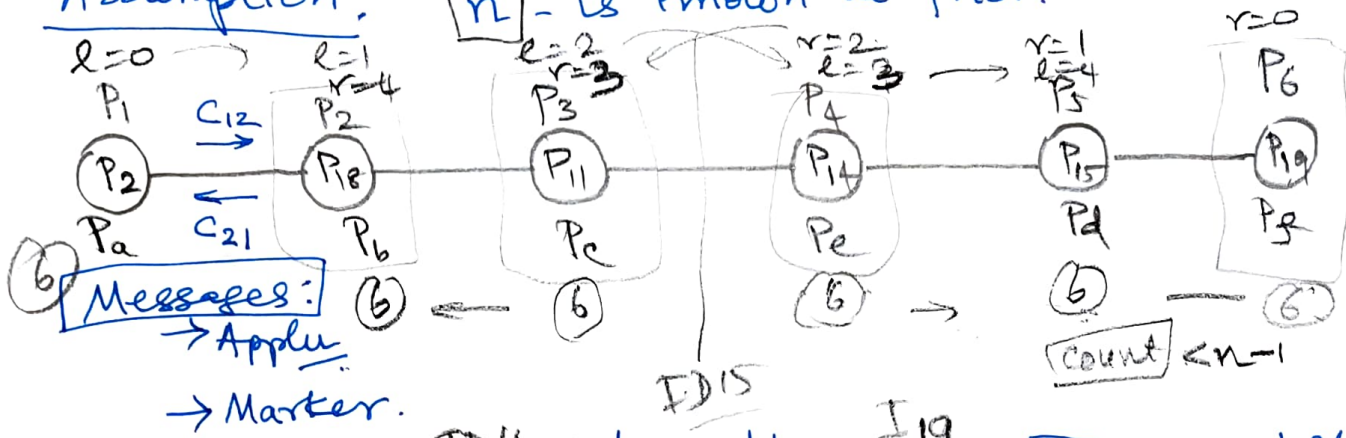
Defn:

     Each $(P_{i-1}) \longleftrightarrow \boxed{P_i} \longleftrightarrow (P_{i+1})$ $\boxed{P_1}$ has no left $\boxed{P_n}$ has no right

$$\forall\, P_i \,, \quad 1 \le i \le n$$

$r=5$ $r=4$ $r=3$ $l=3$ $l=4$ $l=5$

$l=0$ $l=1$ $l=2$ $r=2$ $r=1$ $r=0$

$\boxed{\text{DSLN}}$ → ○ —— ○ —— ○ —— l ○ r —— ○ —— ○ ←

   6 6 $\boxed{6}$ $\boxed{6}$ 6 6

                                     line (n)

         ~~$n = 6$~~
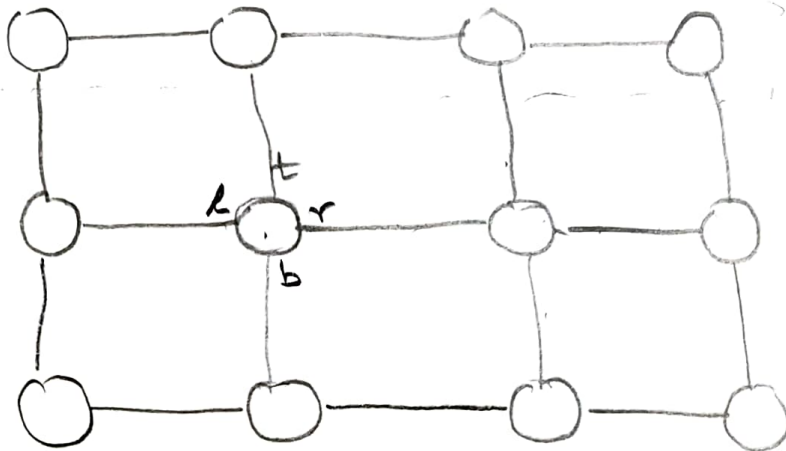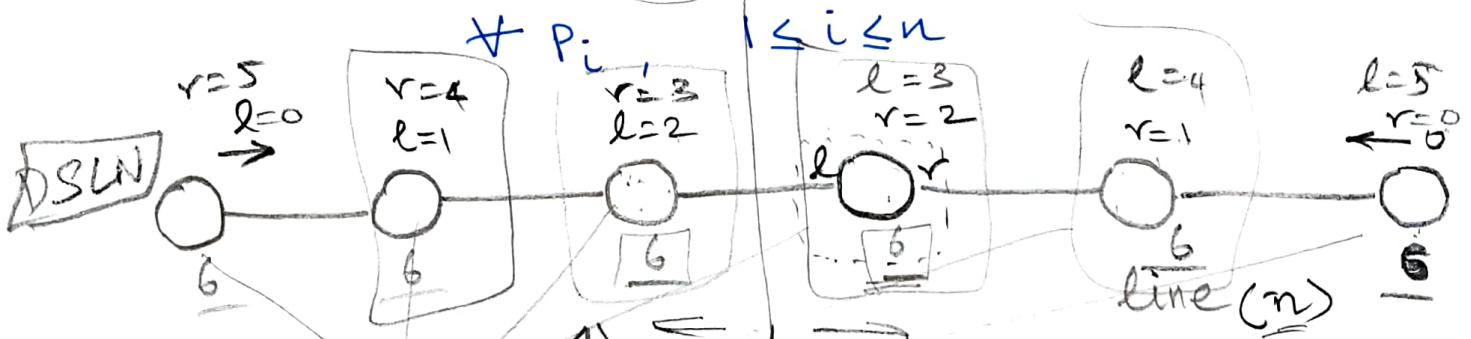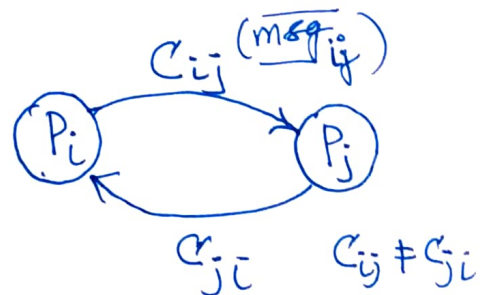

mesh (3×4)
$= 12$

A Distributed computation is globally terminated iff * every message is locally terminated
* there is no message in transit between any pair of processes.

Channel between any P_i and P_j is empty

→ Basic (Apply)
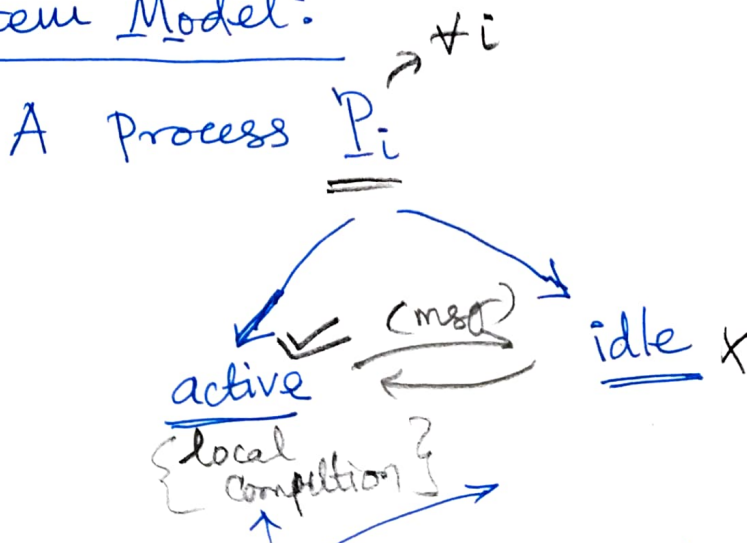 → Data exchange

→ Control (Marker)
  → Termination Detection.

$C_{ij} \; (\overline{msg}_{ij})$

C_{ji} $C_{ij} \neq C_{ji}$

TD Ensures:
  * Do not indefinitely delay the underlying computation

  * Do not require additional channels.

System Model:

A Process P_i → ∀i

active   (msg)   idle ✗

{local completion}

$P_i(t)$ → state of a process P_i @ time t.

$C_{ij}(t)$ → state of the channel connecting P_i & P_j @ time instance t.