

ALGORITHMIC APPROACH TO D.S.

2020/2021



20/09/21

LEADER ELECTION

COMPLETE
GRAPH LB

WORST CASE

- Process i receive IDs in increasing order
 - it receive $m-1$ msgs
 - it broadcast ($m-1$ msgs) for each one he receive
 - there are m processors
- $$\Rightarrow \# \text{MSGS} = (m-1)(m-1)m = O(m^3)$$

BEST CASE:

- Process i receives IDs in mon-increasing order
 - it receive $m-1$ msgs
 - there are m processors
- $$\Rightarrow \# \text{MSGS} = (m-1)m = O(m^2)$$

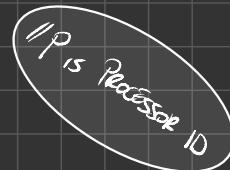
CHANG / ROBERTS

- ASSUME that exists a UNIDIRECTIONAL RING of non faulty processors
- PROCLAMATION is needed (<End> msg)

ALGO

I_p

$$\{M = \emptyset\}$$



$$\{ \text{RECEIVING } J \}$$

IF ($J > M$)

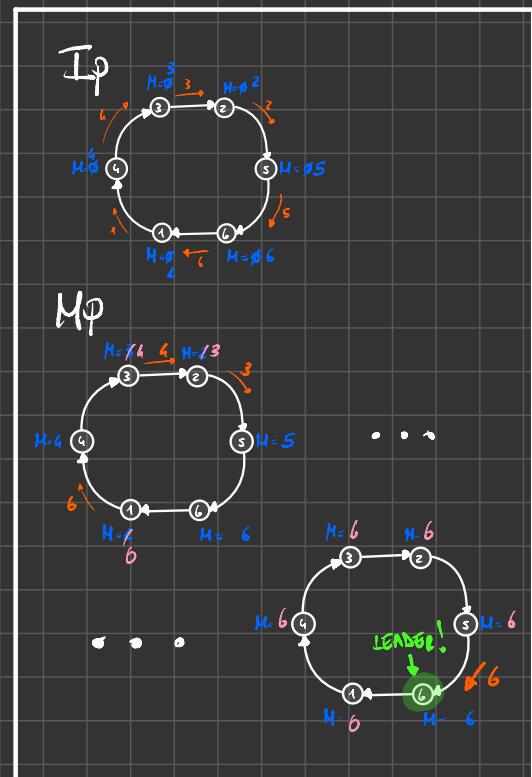
$$M = J$$

SEND M

IF ($M = p$)

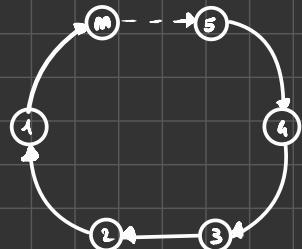
PROCLAMATION

STOP



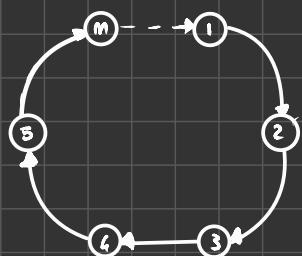
NOTE IGNORE PROCLAMATION

WORST CASE



#MSUs = $m + (m-1) + (m-2) + \dots + 1 = \sum_{i=1}^m i = \frac{m(m+1)}{2} = \Theta(m^2)$

BEST CASE



#MSUs = $\underbrace{m + 1 + \dots + 1}_m = 2m = \Theta(m)$

AVG CASE

- The Prob. for a node to make one step is $\frac{1}{m}$, so to complete the circle is $\sum_{i=1}^m i \cdot P(\text{MAKING ONE STEP}) \leq \Theta(m \log m)$
 $\Rightarrow \Theta(m \log m)$

ROUND BASED LEADER ELECTION

- A mode become PASSIVE if its value P is less than the one it receive
- A passive mode only forwards messages

FRANKLIN ALGO (ON BIDIRECTED G)

- Processes send msgs in 2 directions
- At each round at least half Proc's become PASSIVE
 (Since every 2 near nodes one should win) $\Rightarrow \# \text{Rounds } O(\log m)$
- In each round ACTIVE send 2 msgs, PASSIVE forward 2 msgs $\Rightarrow \# \text{MSUs} = 2m \cdot O(\log m) = O(m \log m)$

27/09/21

ELECTION ON FAULTY SYSTEMS

- We assume that links are faulty free.
- We assume a TIMEOUT. (BOUNDED SYNCHRONICITY)
 - ⇒ It follows that we can DETECT FAULTY PROCESS
- Assume a fully-connected topology.
- If the leader fail, every process will try to replace it.

IDEA

BULLY ALGO [GARCIA-MOLINA]

- Detection Ph.
- TIMEOUT reached while waiting LEADER's ping reply.
- Preparation Ph.
- P_i sends "ELECTION" msg to every $P_j | j > i$
- Election Ph.
- If no P_j reply in TIME then I'M THE LEADER ⇒ Proclamation Ph
 - If $P_j | j > i$ receive P_i msg ⇒ P_j reply "YOU CANNOT BE THE LEADER" ⇒ P_j (if not yet) goes to Preparation Ph. and P_i goes to Proclamation Ph. (will wait a LEADER)
- Proclamation Ph.
- LEADER P_i LEADER send its "ID" to all $P_j | j < i$
 - Other P_j waits for LEADER PROCLAMATION MSG, If TIMEOUT expires ⇒ goto Proclamation Ph.
LEADER FAILED DURING
IT'S OWN PROCLAMATION

COMPLEXITY

- When P_i detects LEADER FAILURE it broadcast to his $(m-1)$ neighbourhood, getting back at most $(m-1)$ replies
- WORST CASE:

$$\# \text{MSGS}_{\text{worst}} = O(m^3)$$

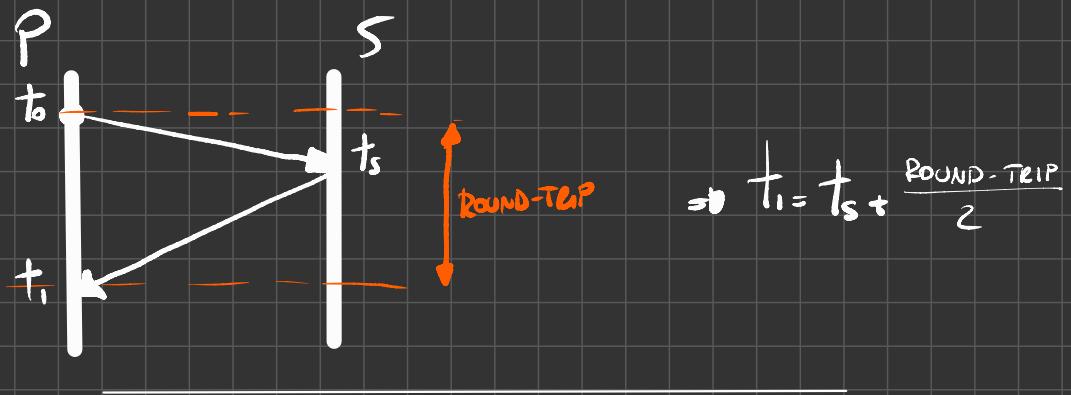
NOTE

- If a failed processor is restarted it try to elect himself, so if it's the greatest ID proc., we will have 2 LEADERS, we should so consider EPOCHS to keep the latest elected one.

27/09/21

PHYSICAL CLOCKS

CHRISTIAN'S
PHYSICAL CLOCK



COMPUTER
CLOCK

- Compute the average of the times of every processor; then each one set the average as it's TIME
- Same as COMPUTER CLOCK, but it consider ROUND-TRIP DELAYS too

BERKEK'S
PHYSICAL CLOCK

LOGICAL CLOCKS

LAMPORT'S
LOGICAL
CLOCK

- Each p_i starts with a var. $L = \emptyset$
- BEFORE a LOCAL EVENT $L = L + 1$
- A msg m sent have L as TIMESTAMP RECEIVING m IS AN EVENT
- When a msg m is received $L = \max(L, \text{TIMESTAMP}(m)) + 1$ \sim

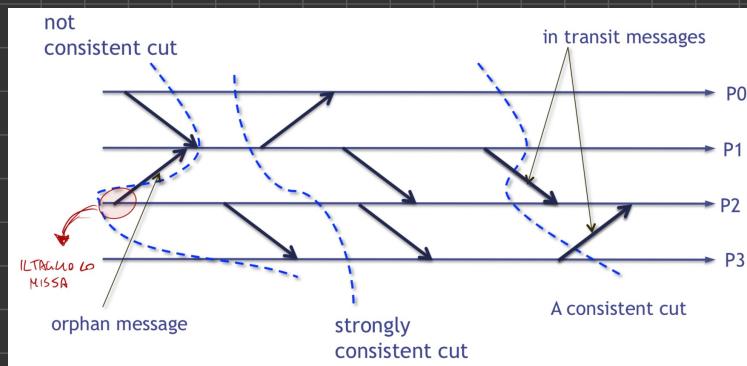
VECTOR
CLOCKS

- P_i start with a vector $V = [V_1, \dots, V_n]$ initialized to \emptyset
- $V[i]$ contains #events timestamped by P_i
- $V[j] \neq i$ contains #events of P_j which have interacted with P_i
- BEFORE P_i LOCAL event: $V[i] = V[i] + 1$
- A msg sent has the whole vector V
- When a msg m is received from P_j $V[j] = \max(V[j], V_{\text{recv}}[j]) + 1$

6/10/21

FAULTS AND RECOVERY

CUT

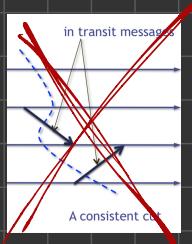


CHECKPOINT

- Restart from a **CONSISTENT CUT**; So in the cut we save the state of the system.

NB

- In case of RELIABLE system we must use only **STRONGLY CONNECTED CUTS**



- In this case is not guaranteed that after a restart will be sent the very same msgs.

ORPHAN MSHS

- If it restart the msg will be RECEIVED z times from the top processor

