

## Agenda:

- Chandy- Lamport snapshot algorithm
- Limitations | assumptions | properties
- Centralised vs decentralized algorithms

Given events A and B,  $A \rightarrow B$ , if B is in snapshot, A should be too!

## Recording a snapshot

The initiator process: (one or more)

- Records its own state
- sends a marker message out on all its outgoing channels.
- start recording the msgs it receives on all its incoming channels.

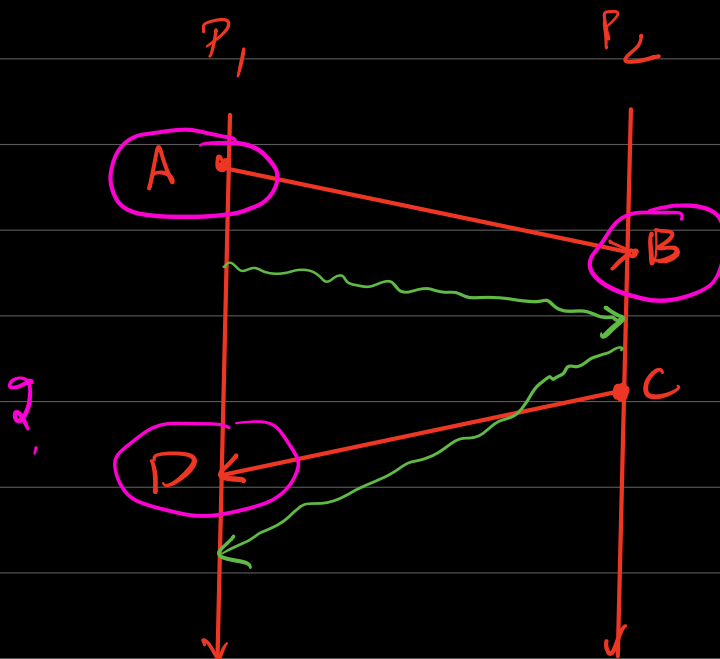
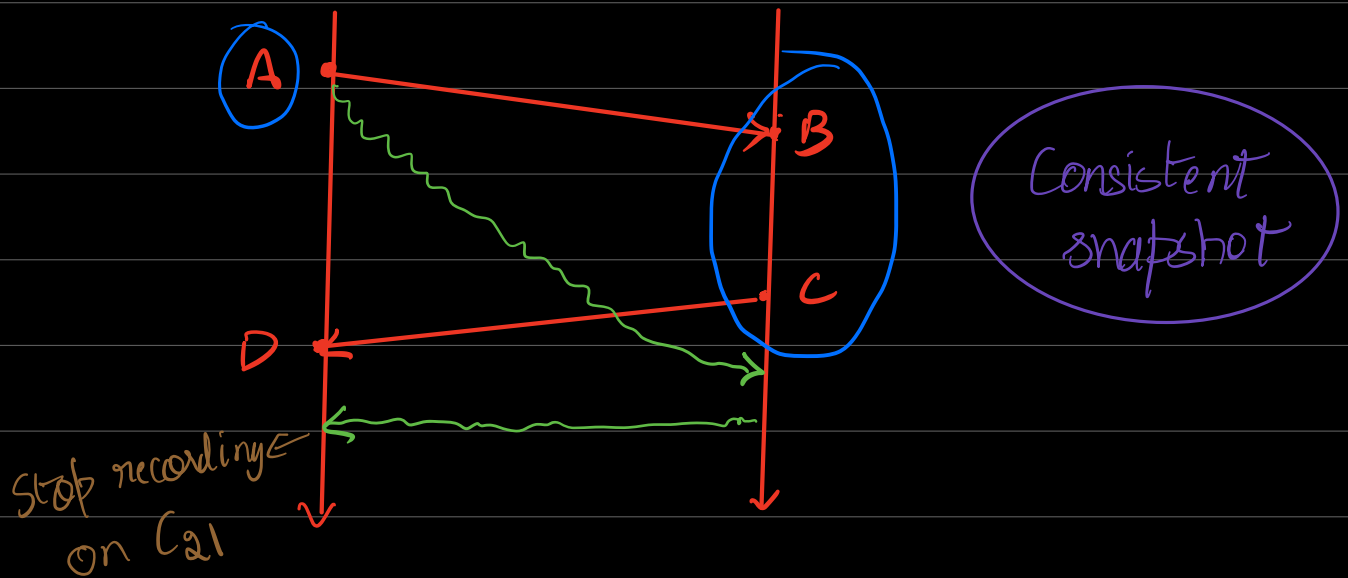
When process  $P_i$  receives a marker message on  $C_{ki}$ :

- if it is the first marker  $P_i$  has seen
  - $P_i$  records its state
  - $P_i$  mark channel  $C_{ki}$  as empty
  - $P_i$  sends a marker out on all its outgoing channels
  - $P_i$  starts recording on all incoming

channels except  $C_{ki}$

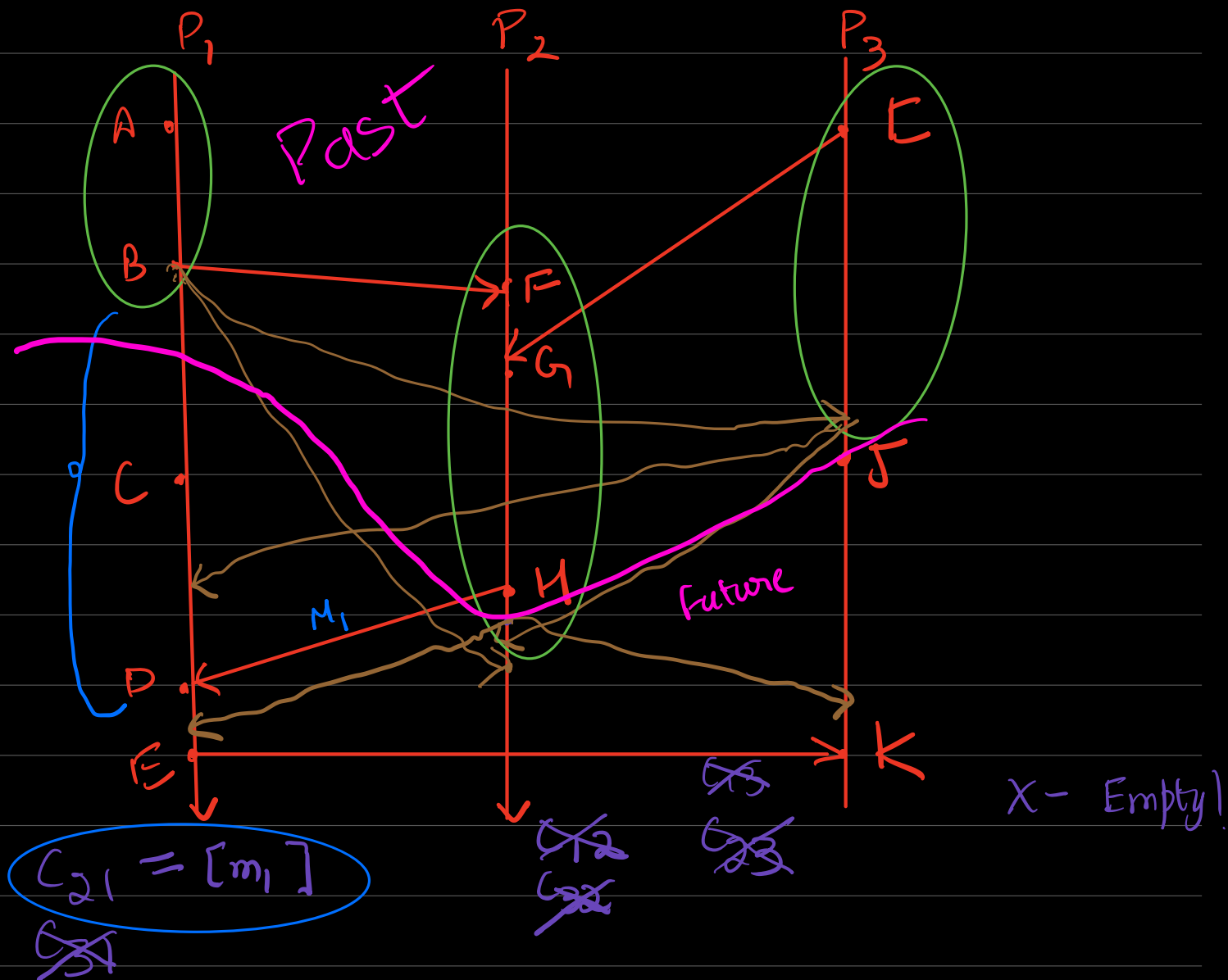
- Otherwise :

-  $P_i$  stop recording on  $C_{ki}$ .



why can't this NOT happen in Chandy-Lamport?  
(channels are FIFO) Marker message from  $P_2$  to  $P_1$  cannot happen before msg from  $C$  to  $D$ .

# More interesting CL algorithm



Total number of marker msgs.  
 $N(N-1)$


$N$  - No. of processes

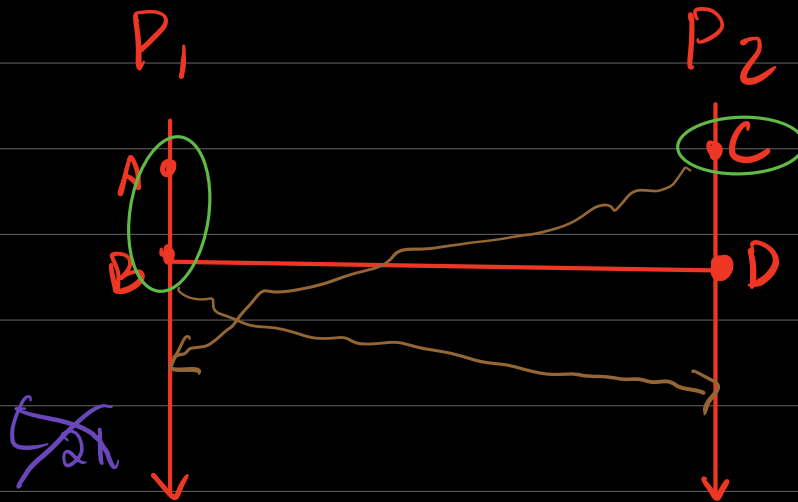
- Even with multiple initiators, the final snapshot is a consistent snapshot.
- It can run even when app messages are being sent / happening.

# CL snapshots: BIG PICTURE

- **channels**:
  - How msgs travel
  - FIFO behaviour
- This FIFO assumption is a requirement for CL algorithm!
- No pausing application messages in CL.
- C-L assumes messages are NOT lost, corrupted or duplicated! Also, processes don't crash!

## C-L: Good things

- Snapshots are consistent ( $\nexists A, B$   
if  $A \rightarrow B$   
& B is in snapshot,  
so is A)
- Guaranteed to terminate (given assumptions about msgs not being lost, etc.)
-  - Works fine with more than one initiator!



This property makes C-L a decentralized algorithm.

Examples : Chandy-Lamport  
Paxos

A centralized algorithm must be initiated by exactly one process!

What are snapshots for?

- Checkpointing!
- Deadlock detection!
- Detection of any stable property

once something becomes true,  
it always stays true.

