Distributed System 1 - synch 3

Francesco Penasa

March 11, 2020

```
2020 03 03
```

1 Vector clocks

Example: bulletin boards Msg and reply should arrive in order. Just look at the rules to understand if we can accept.

1. condition one
2. condition two

2 Global States

2.1 Cut

The union of the partial histories of all the processes.

Consistent cut send should happen before than receive if receive happen before send it is **not** consistent

Distributed Snapshots A picture of system how it might have been (In a consistent way). How to do it:

1. Assume FIFO
2. Any process initiate a snapshot: record your state, send it to all outgoing links, monitor all incoming links.
3. Upon receiving a token: start recording if you were not recording, stop recording the link in which we have received.
4. Recording messages: when all incoming have arrived stop monitoring.

3 TODO

Exercises! 45: write the list of the nodes table style.

4 Exam questions

- Exercise **synch 31:** $P_1 8, 3, 4, 5$; $P_2 1, 9, 4, 5$; $P_3 1, 3, 5, 5$; $P_4 1, 3, 4, 9$;
- Exercise **synch 32:**
  1. it's not truly multicast because all 3 starting clocks are different, while in multicast there could be only one different from the others.
  2. $P_2 receive(m4) \neq (6, 5, 5)$
  3. $P_3 send(m2) \neq (6, 4, 4)$
  4. links are FIFO but $P_1 receive(m4) < P_1 receive(m2)$ while $P_3 send(m2) > P_3 send(m4)$