$A \rightarrow B \Rightarrow LC(A) < LC(B)$

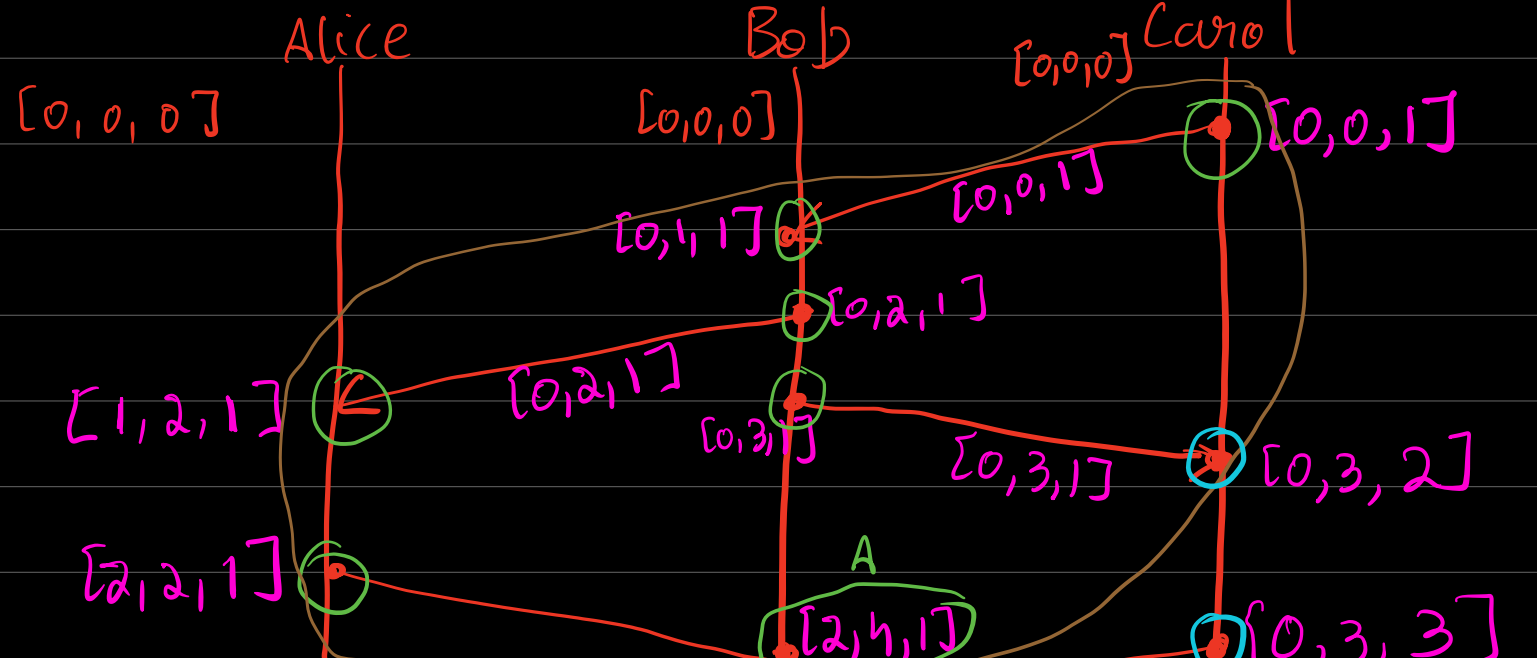
$LC(A) < LC(B) \Rightarrow A \not\rightarrow B$

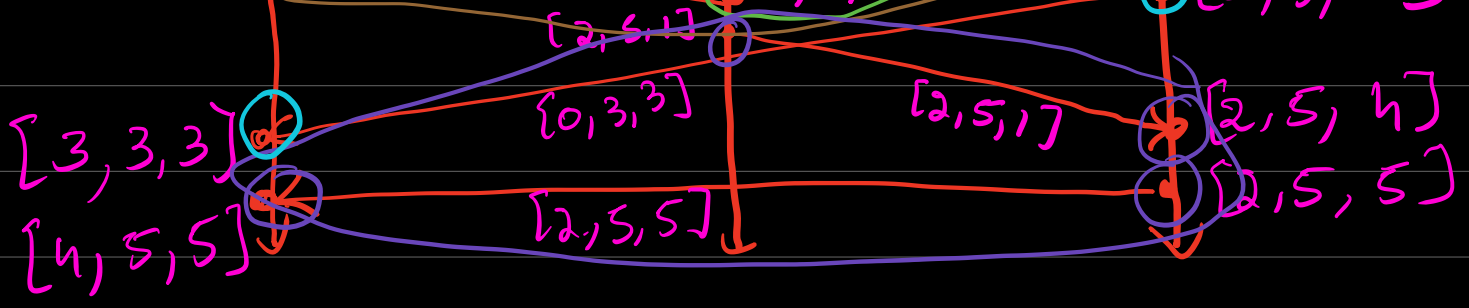
Pink not related by $\rightarrow$ relationship

Solution : VECTOR CLOCKS

$A \rightarrow B \iff VC(A) < VC(B)$

Example





Events befor A ? Everything in green

Area enclosing all events: causal history of A!

All elements are smaller than VC of A IN EVERY POSITION

Events in purple are causally later (or causal future) than A.

All elements are >= the VC of A IN EVERY POSITION

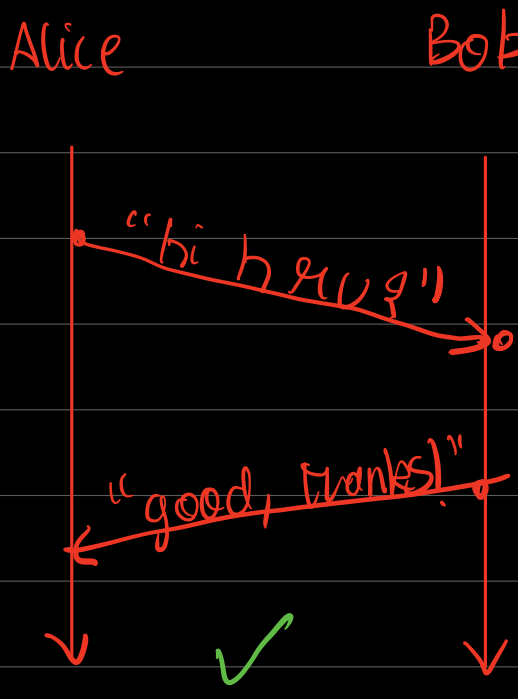
Events in cyan, some elements bigger or smaller than VC(A).

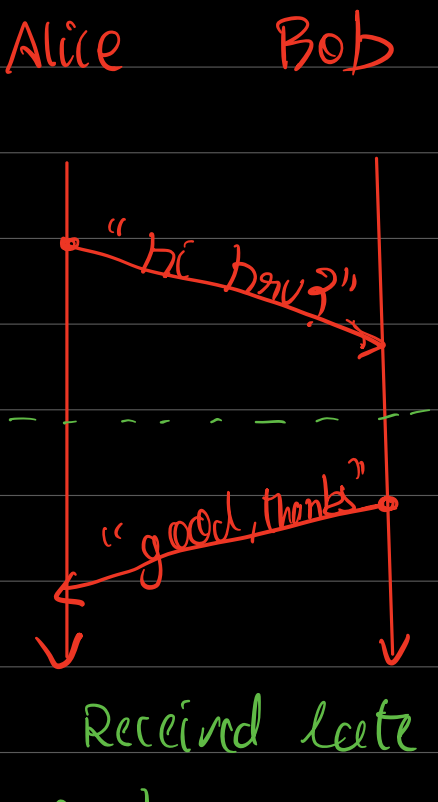
Events in cyan are concurrent or causally independent

Computers are great at comparing vectors!

Protocol:

Set of rules that processes use to communicate with each other.





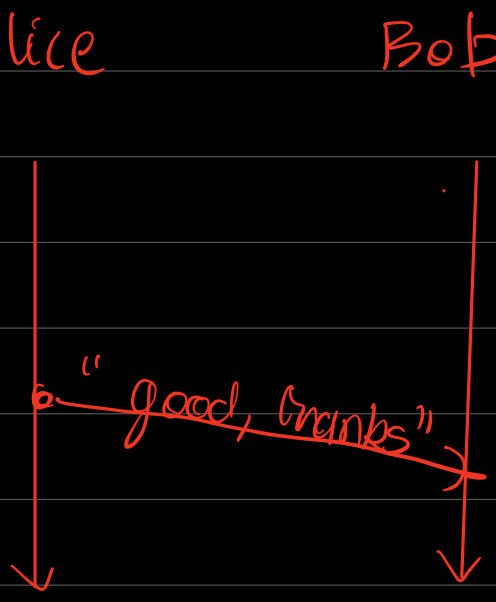
Received late response

For every "hi how r u" message, the process MUST RESPOND with "good, thanks!"

Comparing Lamport diagrams, is there a diff between the above 2 images?

NO. Diff doesn't mean anything here!

Diagram to show violation.



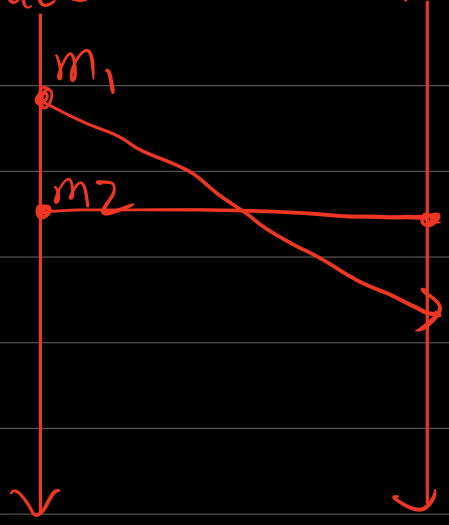
Correctness property of executions.

FIFO Delivery: If a process sends message M_2 after M_1, any process delivering both, delivers M_1 first.

Sending → something you do
Receiving → something that happens to you

Delivering → something you can do with a message you receive
(you can queue up received msgs & wait to deliver them!)

Alice Bob



VIOLATION OF FIFO DELIVERY (if Bob delivers m_2 before m_1)

FIFO delivery already part of TCP. Need to implement if using UDP.

How to enforce FIFO?

SEQUENCE NUMBERS!!

Sender
- messages get tagged with sender ID & sender seq number
- senders increment seq number after sending message

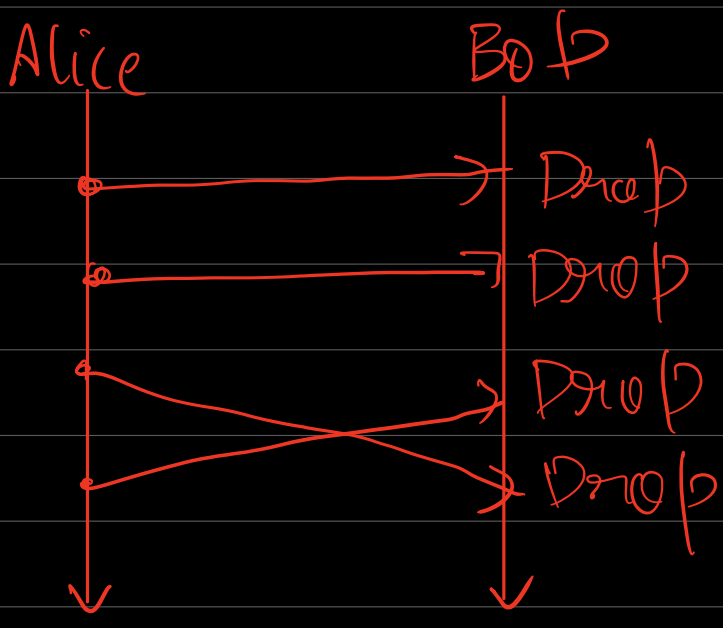
Receiver
- if received msg seq number is the SN of the prev. messages from sender +1, deliver it!

what if message gets lost?

later message gets buffered forever. Increasing memory)

FIFO only works well when you have reliable delivery.

TCP has FIFO & reliable delivery!

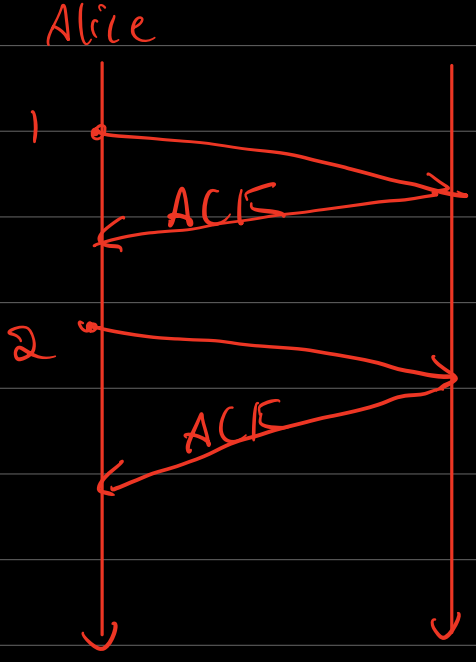


If all msgs dropped, this vacuously satisfies FIFO delivery

other ways to implement FIFO?

Timeouts!

Alice waits for Bob to ACK before sending next msg!

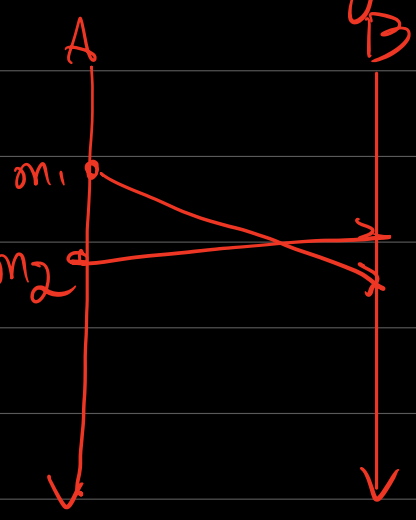


PROBLEM:

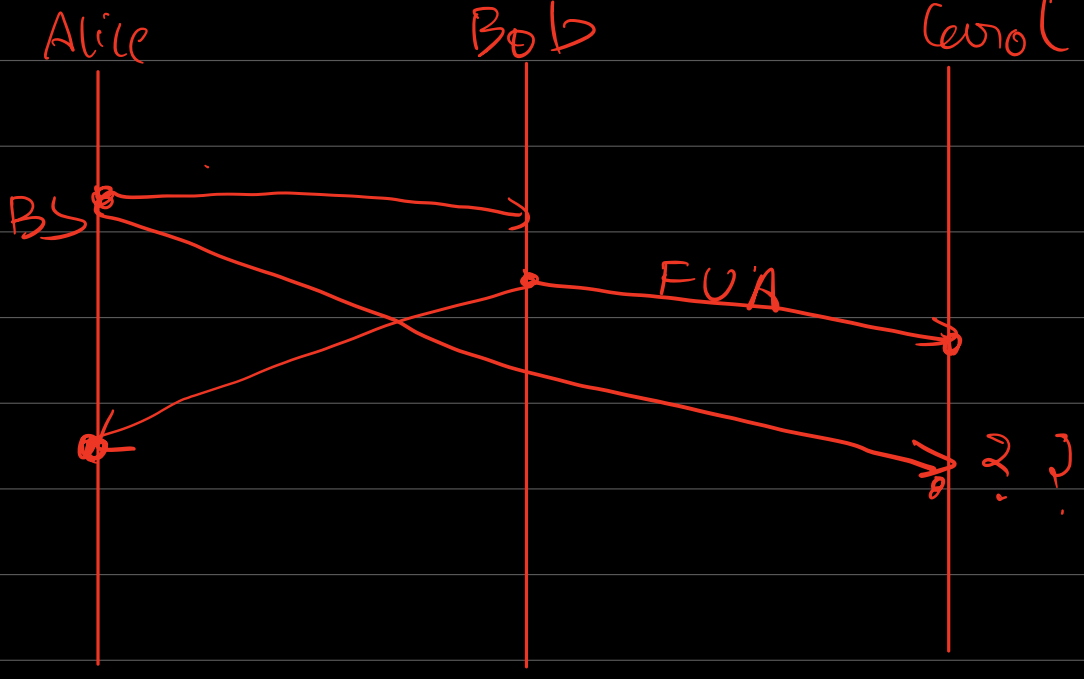
Slow AF!
ACK gets lost

Causal Delivery

If m_1's send happened before m_2's send, then m_1's delivery must happen before m_2's delivery



Violation of FIFO delivery
Violation of causal delivery



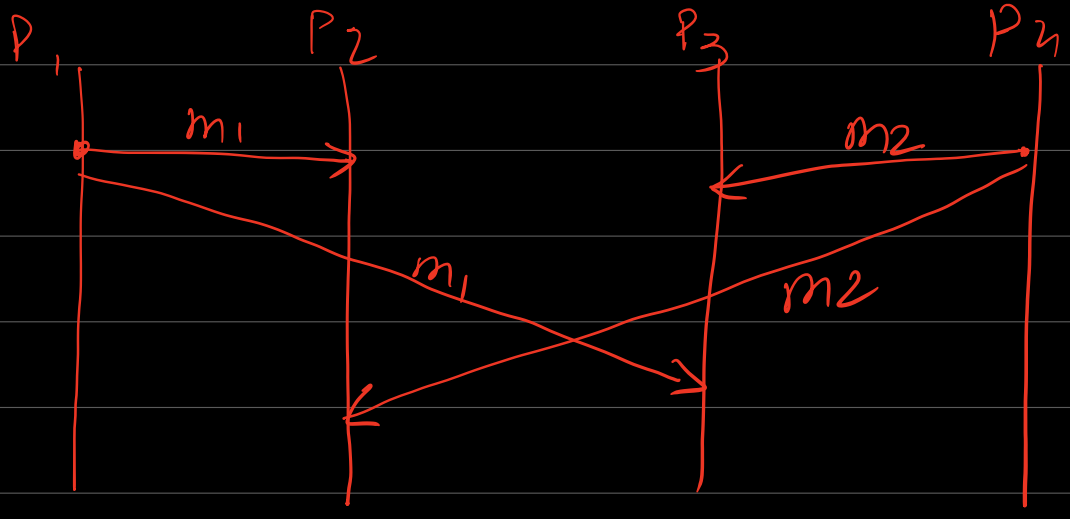
NO Violation of FIFO delivery
But, Violation of causal delivery

FIFO $\subset$ Causal delivery

↓ subset

Totally ordered delivery

If a process delivers m_1 & m_2, then all processes delivering m_1 & m_2 deliver m_1 first



Violation of totally ordered delivery