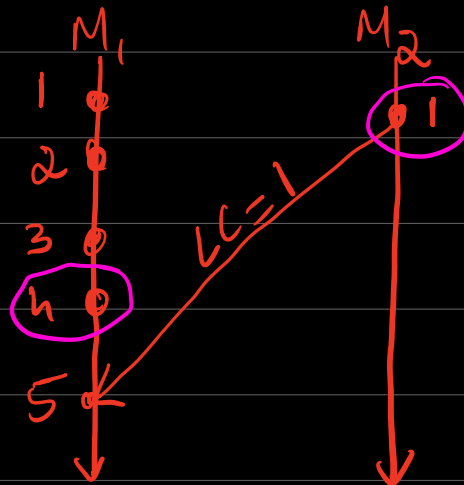


$$A \rightarrow B \Rightarrow LC(A) < LC(B)$$

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

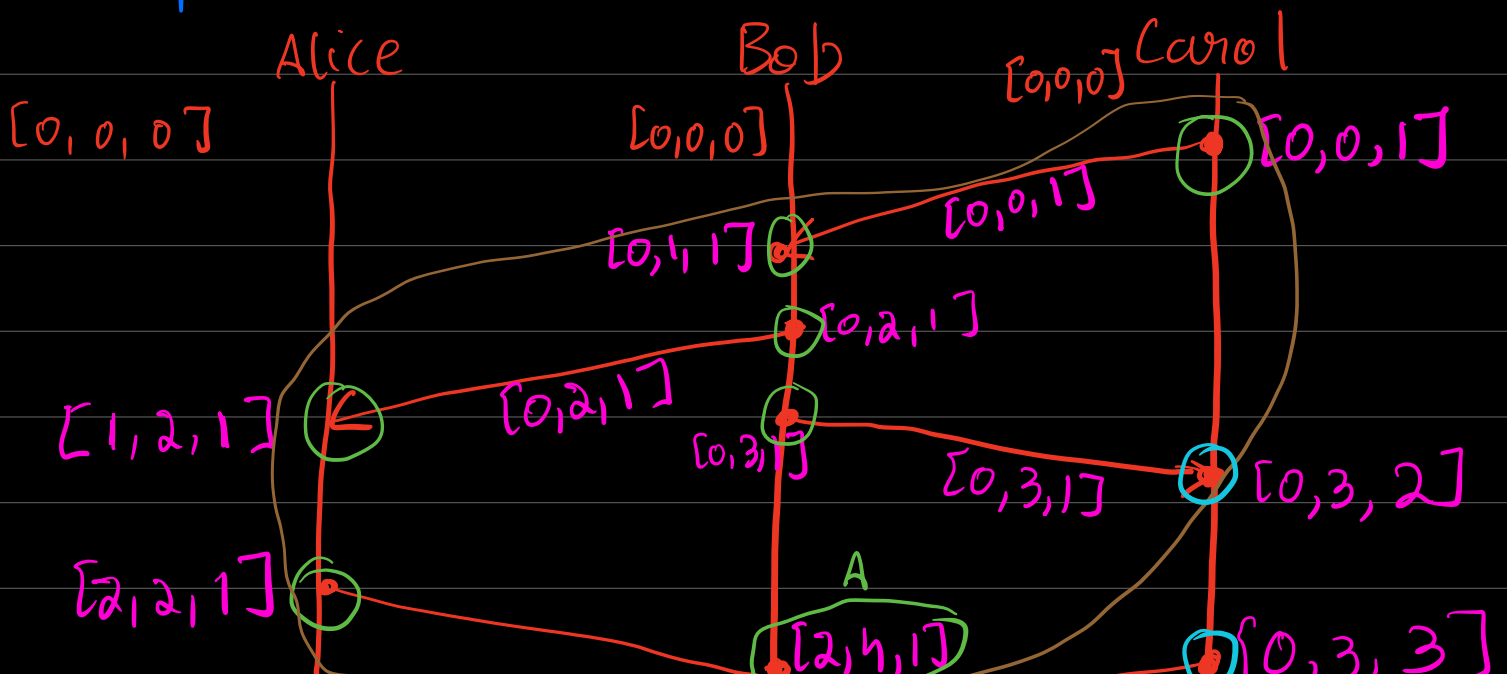


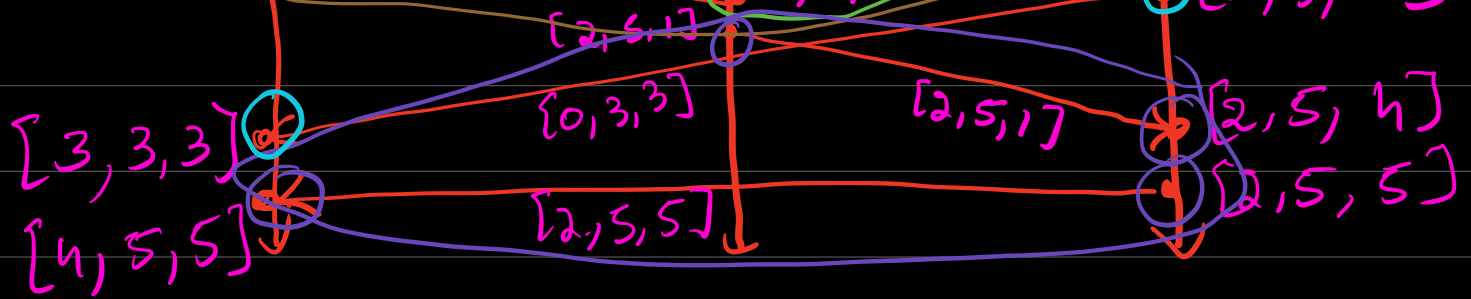
pink not related
by \rightarrow relationship

Solution: VECTOR CLOCKS

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

Example





Events before 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 ^(or causal future) causally later than A.

All elements are \geq the VC of A
IN EVERY POSITION

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

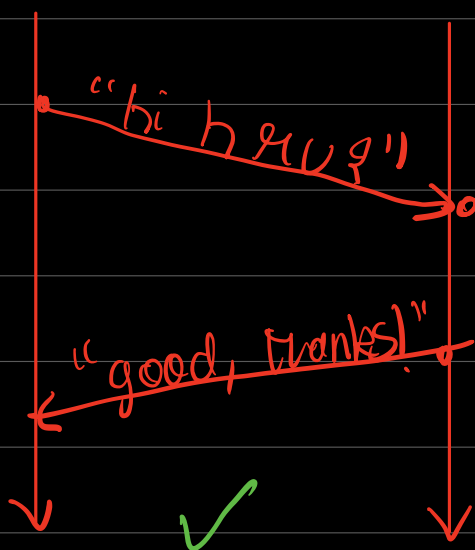
Events in cyan are concurrent or causally independent

Computers are great at comparing vectors!

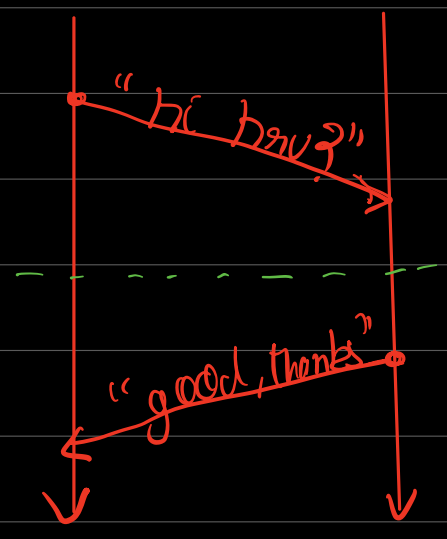
Protocol:

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

Alice Bob



Alice Bob



For every "hi hru?" message, the process MUST RESPOND with "good, thanks!"

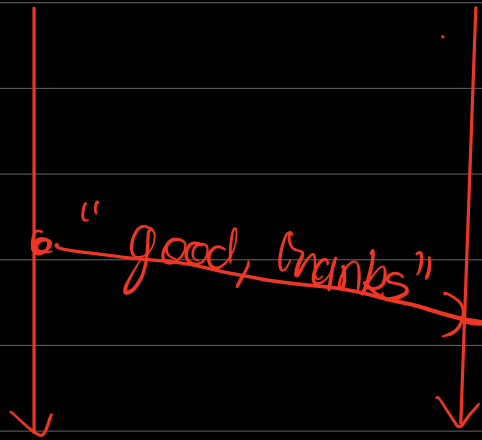
Comparing Lamport diagrams, is there a diff between the above 2 images?

NO. Diff doesn't mean anything here!

Diagram to show violation.

Alice

Bob



VIOLATION !
Initial message
not received

Correctness property of executions.

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

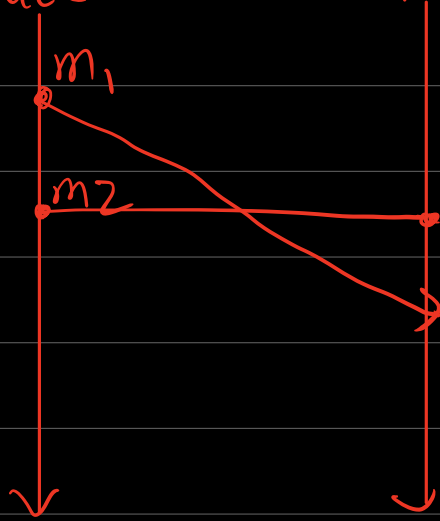
Sending \rightarrow something you do

Receiving \rightarrow something that happens to you

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

Alice

Bob



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

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

How to enforce FIFO?

SEQUENCE NUMBERS !!

Sender

- messages get tagged with sender ID & sender seq number
- sender increments seq number after sending message

Receiver

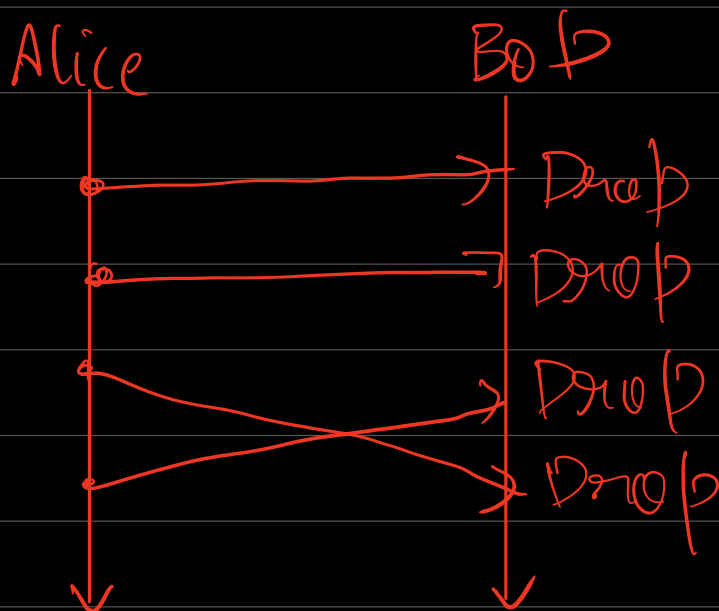
- if received msg seq number is the SN of the prev. message from sender + 1, deliver it!

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!

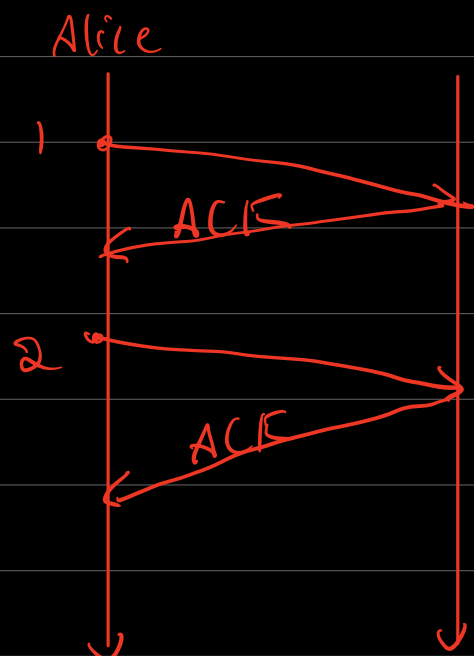


If all msgs dropped,
this vacuously
satisfies FIFO
delivery

Other ways to implement FIFO?

Timeouts!

Alice waits for Bob
to ACK before sending
next msg!



PROBLEM:

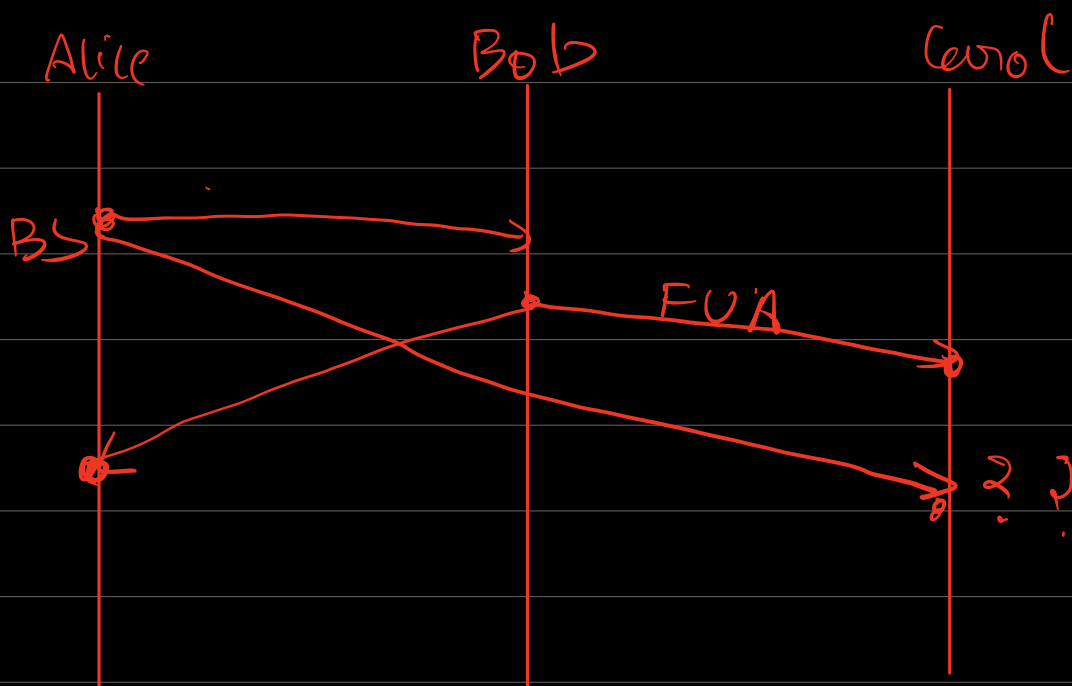
SLOW AF!
ACK gets lost

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 \subseteq Causal delivery
↓
subset

