AMRITA VISHWA VIDYAPEETHAM

AHEAD Online

Distributed Systems

Dr. Swaminathan J
Department of Computer Science
Amrita Vishwa Vidyapeetham

Objectives

- To discuss the different message ordering paradigms.
- To discuss Raynal-Schiper-Toueg algorithm for causal ordering.
- To discuss 3-phase distributed algorithms for total ordering.

Place your
Webcam Video here
Size 100%

Message ordering paradigms

The order of delivery of messages in a distributed system is an important aspect of system executions.

- Because it determines the messaging behavior that can be expected of the distributed program.

1. Async / Non-FIFO
2. FIFO
3. Causal Order
4. Synchronous order

Group Communication

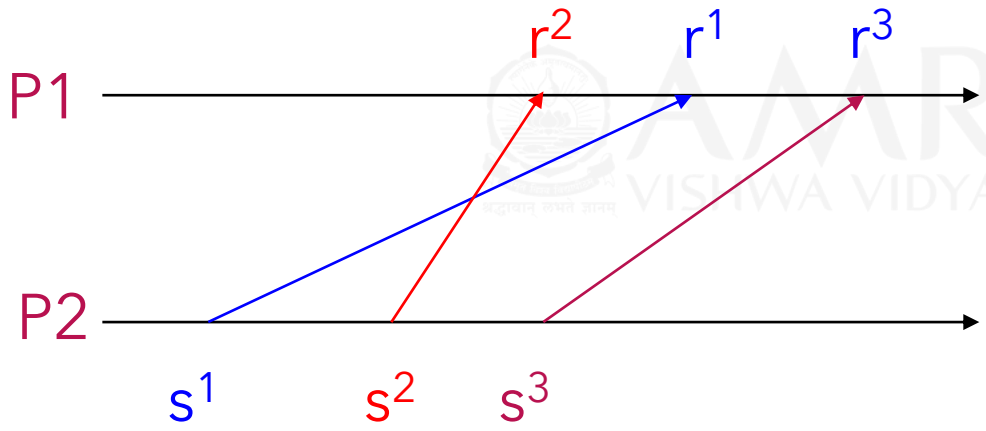
1. Causal Order
2. Total Order

$$\text{Sync} \subset \text{CO} \subset \text{FIFO} \subset \text{Async}$$

1. Asynchronous

Place your
Webcam Video here
Size 38%

- Async → Causality relation is of partial order.



Send will be before receive
(for each message)

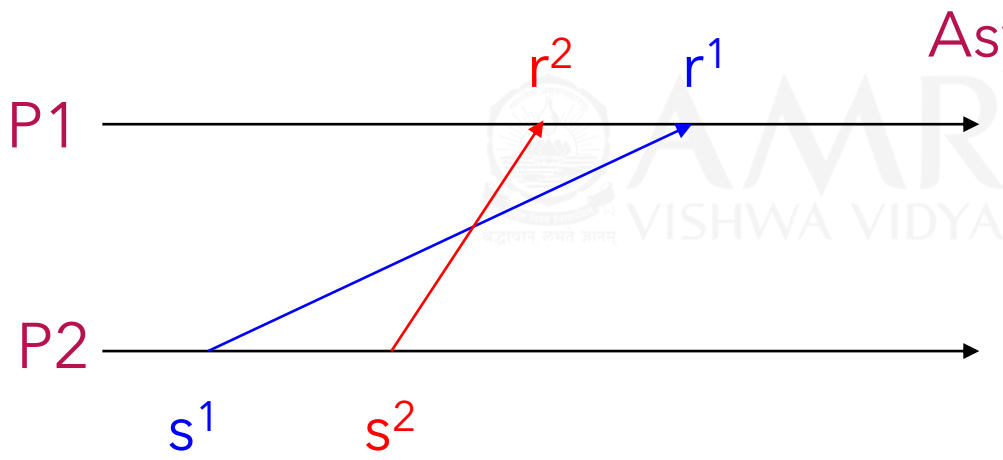
Implementing Async order

- Use Lamport's scalar clock

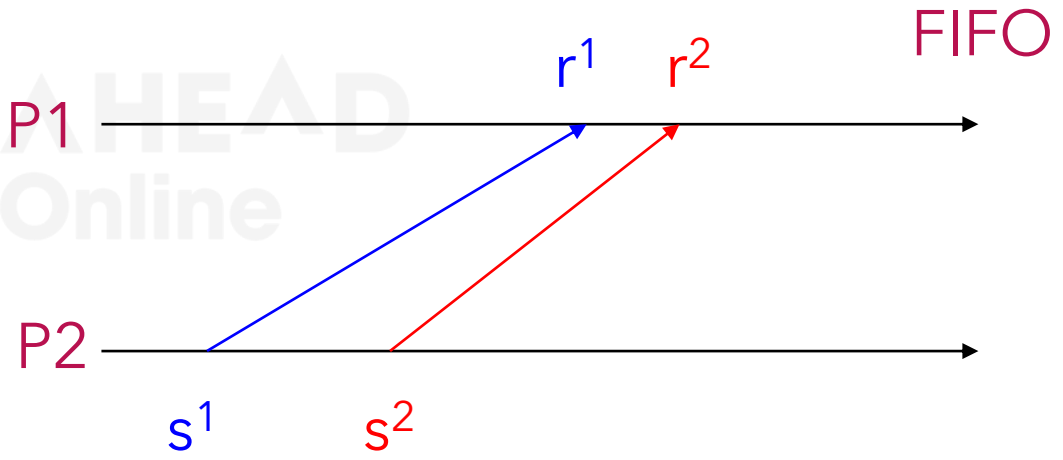
2. FIFO (vs. Async)

Place your
Webcam Video here
Size 38%

- Async → Causality relation is of partial order.
- FIFO → for all (s,r) (s′,r′) on a link, if s → s′ then r → r′



Send will be before receive
(for each message)

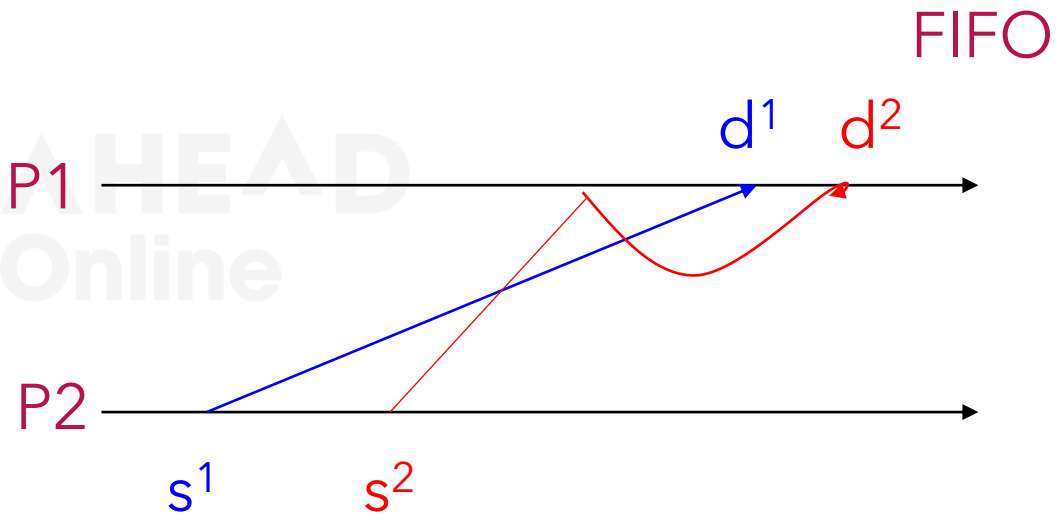


On each link, messages are
delivered in the order sent.

Implementing FIFO

- Using sequence number for each message along each communication link. Just like how TCP does.

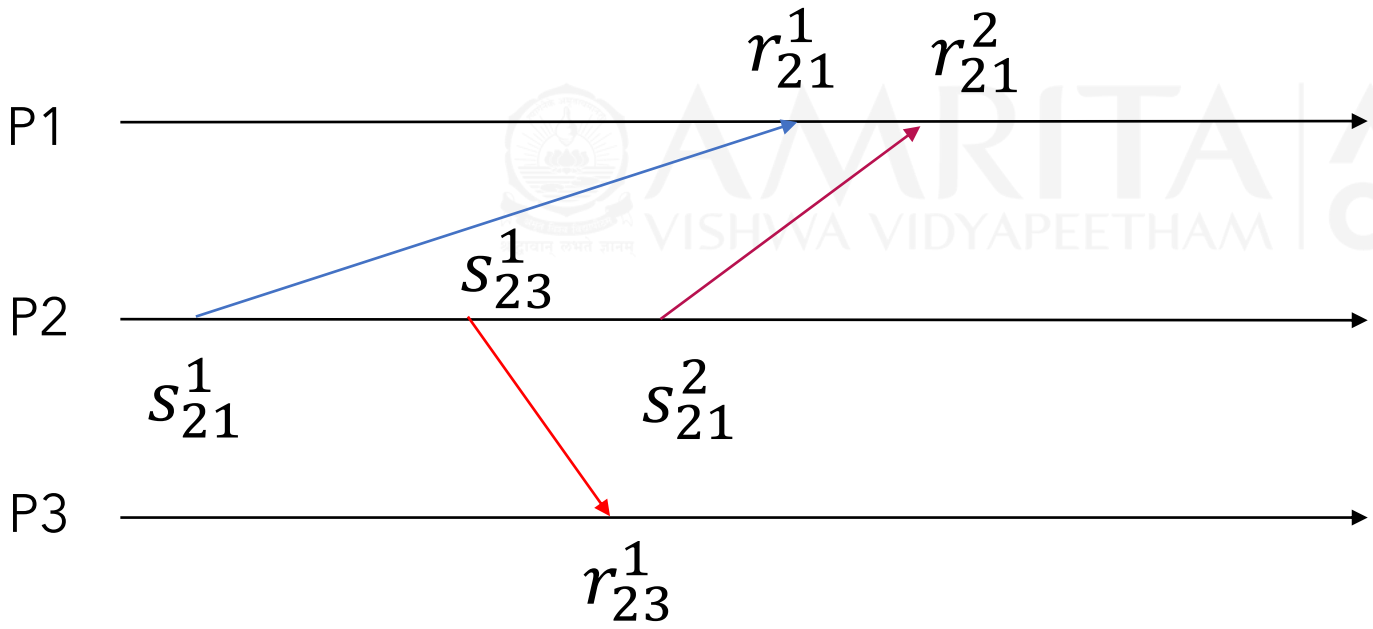
The middleware parks the later message, waits until the earlier message arrives and delivers to the above application in the correct order.



On each link, messages are delivered in the order sent.

What FIFO does not guarantee?

- FIFO does not worry about ordering of messages along two different channels.

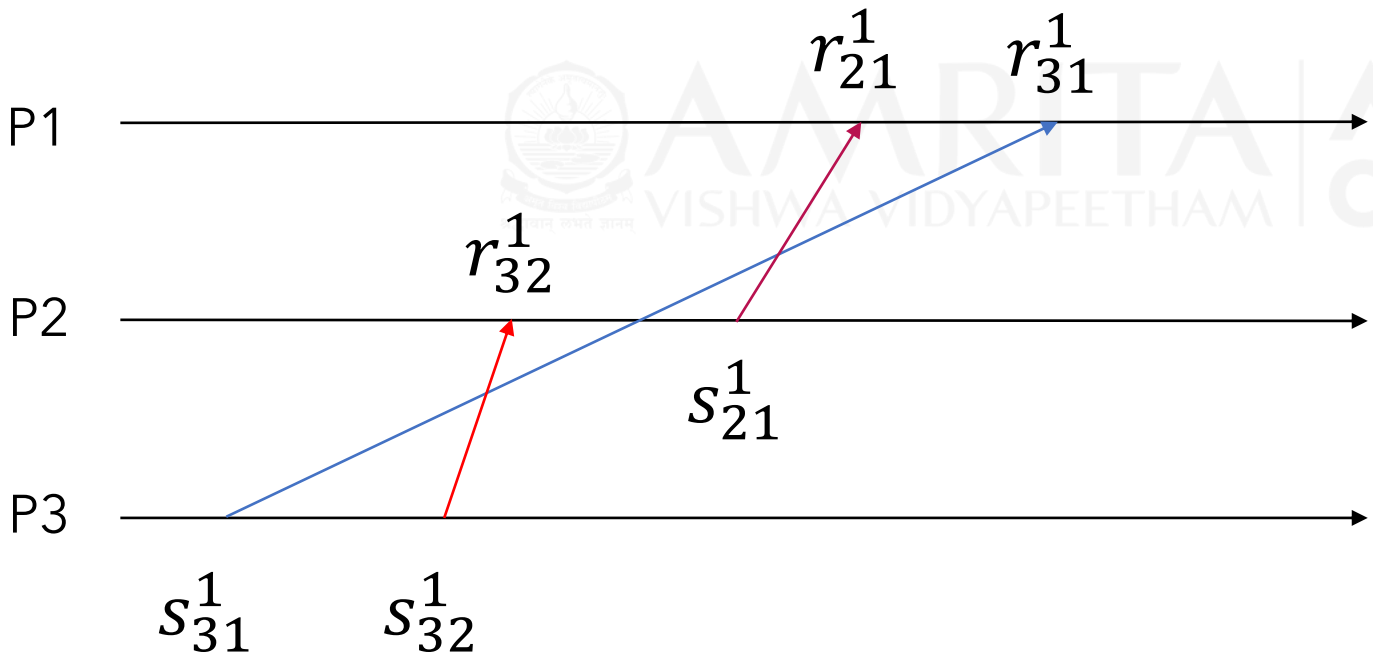


This is trivially FIFO since all 3 messages are along 3 different channels.

3A. Causal Order

Place your
Webcam Video here
Size 38%

- Two causally related messages, arriving at the same destination, although along different channels, are received in the correct order.



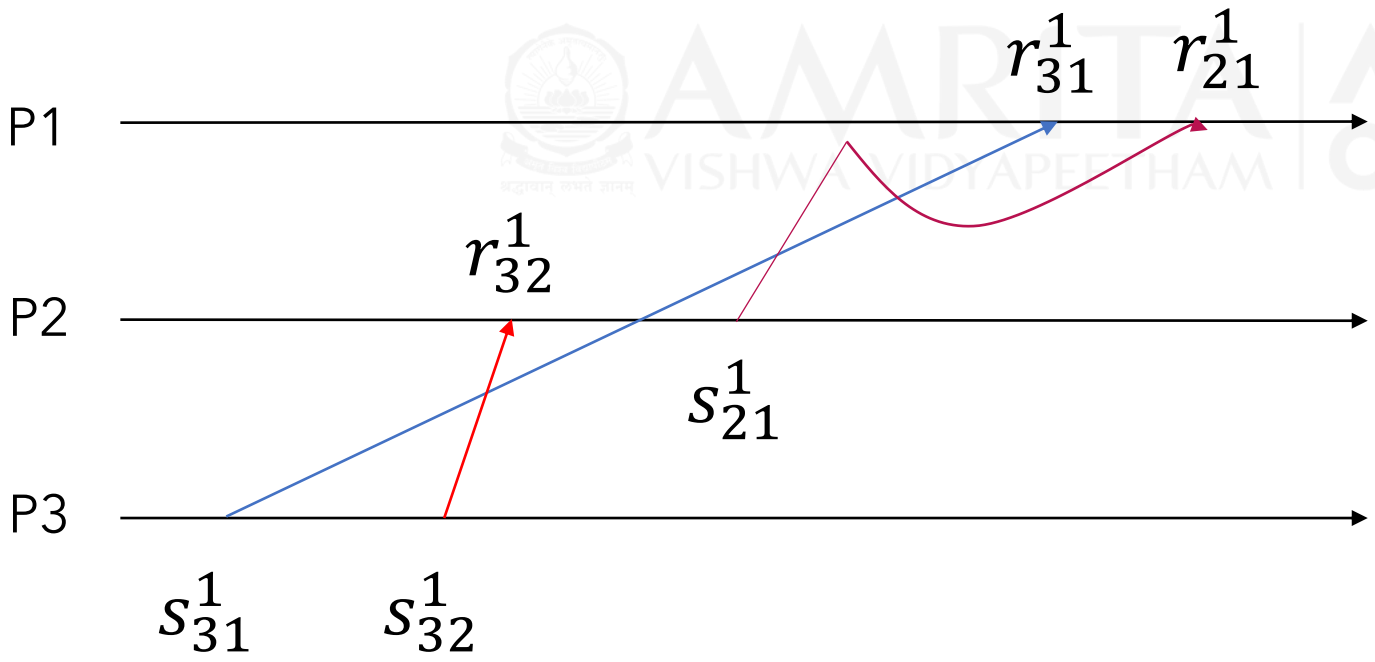
This is not causally ordered

Although, this is FIFO trivially!

Implementing Causal Order

Place your
Webcam Video here
Size 38%

- The middleware recognizes the arrival of later message, parks it until the earlier message arrives and delivers in correct order to application.



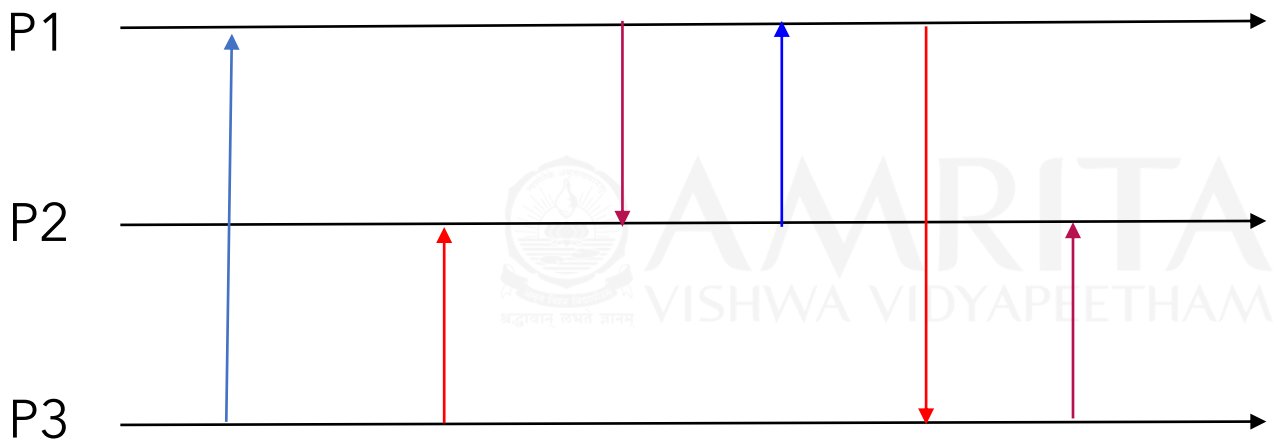
This is now causally ordered

The actual algorithm will be discussed after introducing Causal Order in multicast communication.

4. Synchronous Order

Place your
Webcam Video here
Size 38%

- Every send-receive communication is atomic.



Implementing sync order

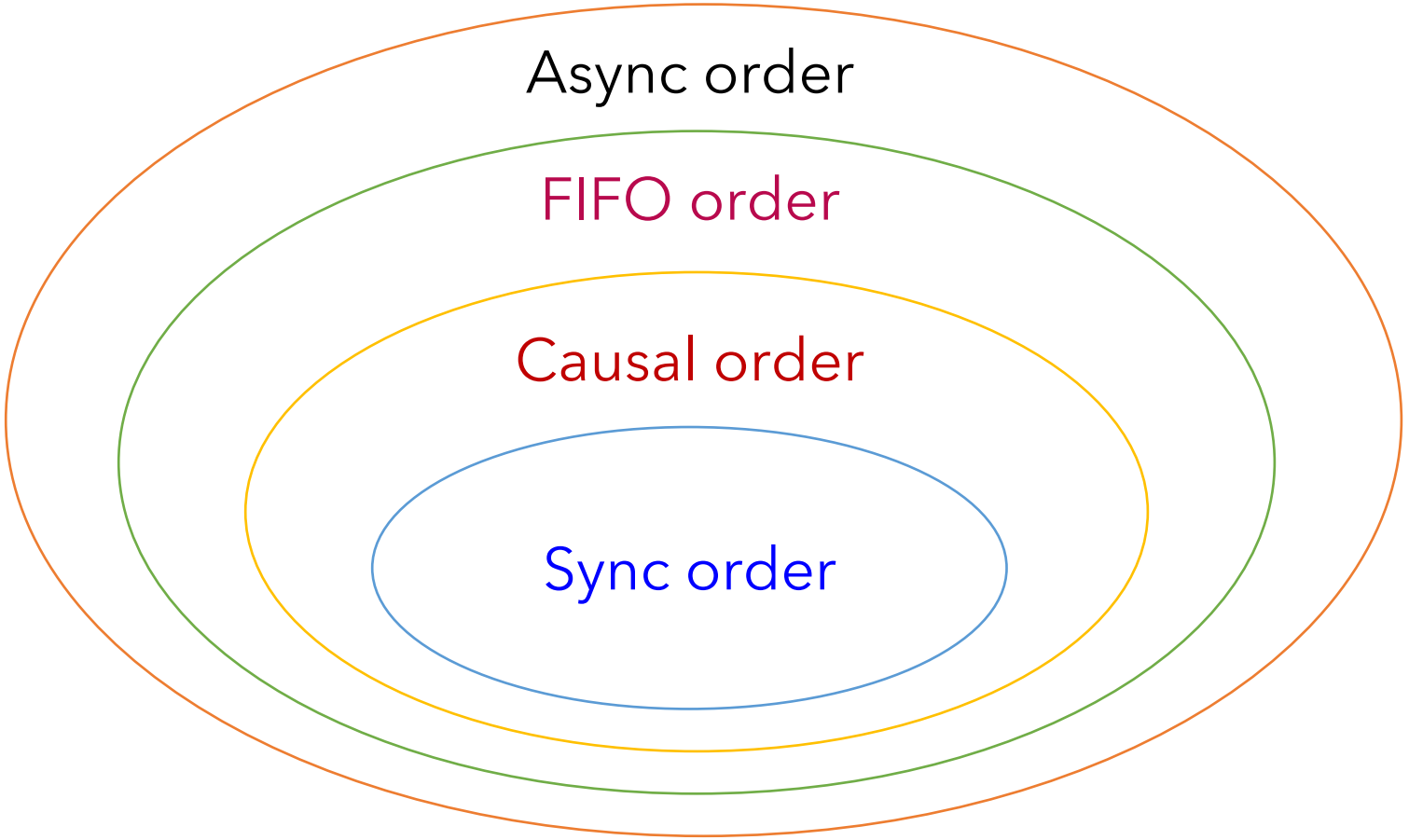
- Centralized coordination (Leader grants permission)
- Token based method (Token owner gets to send)
- Decentralized approach (Voting based methods)

Achieving synchronous order requires a whole lot of work and slows down the performance of the distributed system.

In summary

Place your
Webcam Video here
Size 38%

- $\text{Sync} \subset \text{CO} \subset \text{FIFO} \subset \text{Async}$



Group Communication

Place your
Webcam Video here
Size 38%

- In multi-cast communication, two message ordering paradigms come into picture.

3B. Causal Order

Two causally related messages, arriving at the same destination, although along different links, are received in the correct order.

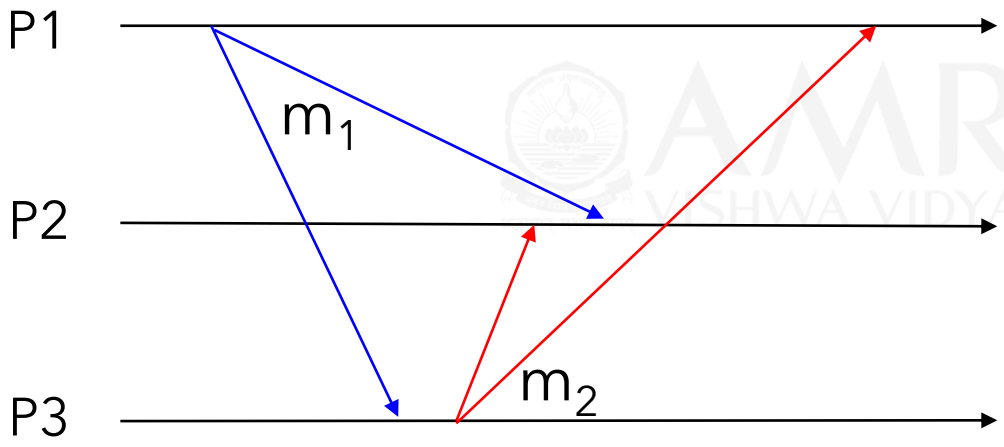
5. Total Order

Messages are delivered in the same order to all recipients of the multi-cast communication.

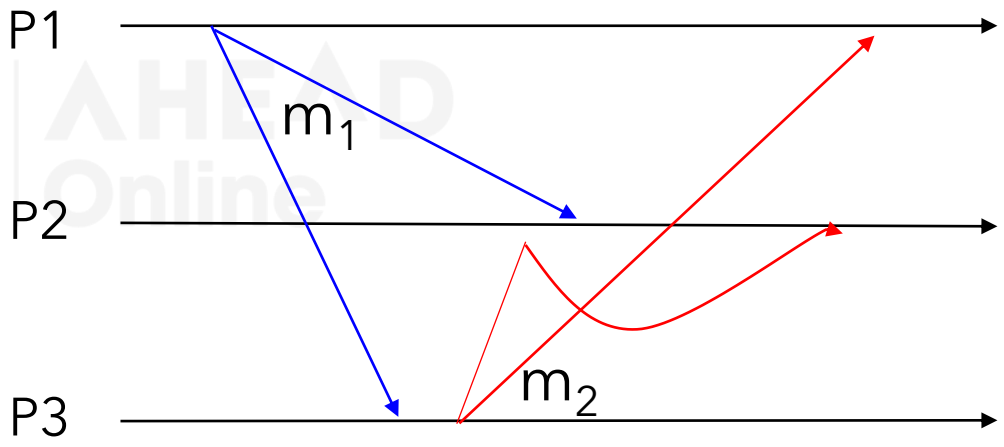
3B. Causal Order in Multicast

Place your
Webcam Video here
Size 38%

- Causal ordering takes a different form in multicast communication.



Not causally ordered
(P2 receives m_2 before m_1)

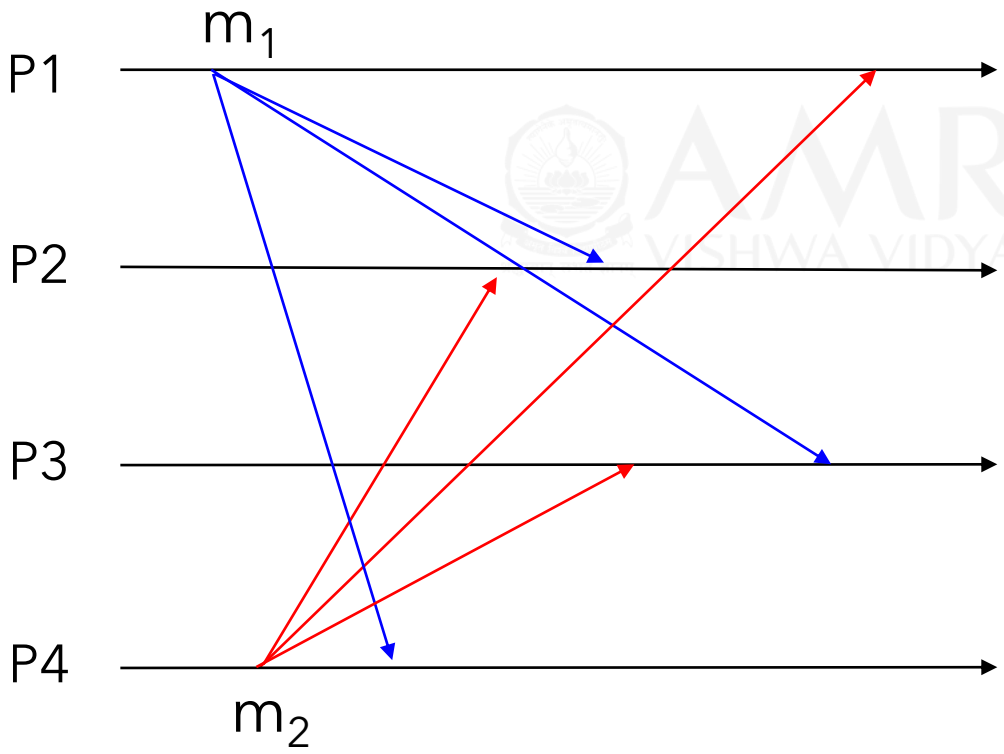


Causally ordered
(P2 parks m_2 until m_1 arrives)

5. Total order

Place your
Webcam Video here
Size 38%

- The order of delivery to all processes must be same.
- Total order is applicable in multicast communication.



- P2 and P3 receive messages in same order. First m2, then m1.
- Hence, total order.

Note: m1 and m2 are concurrent. Neither caused the other.

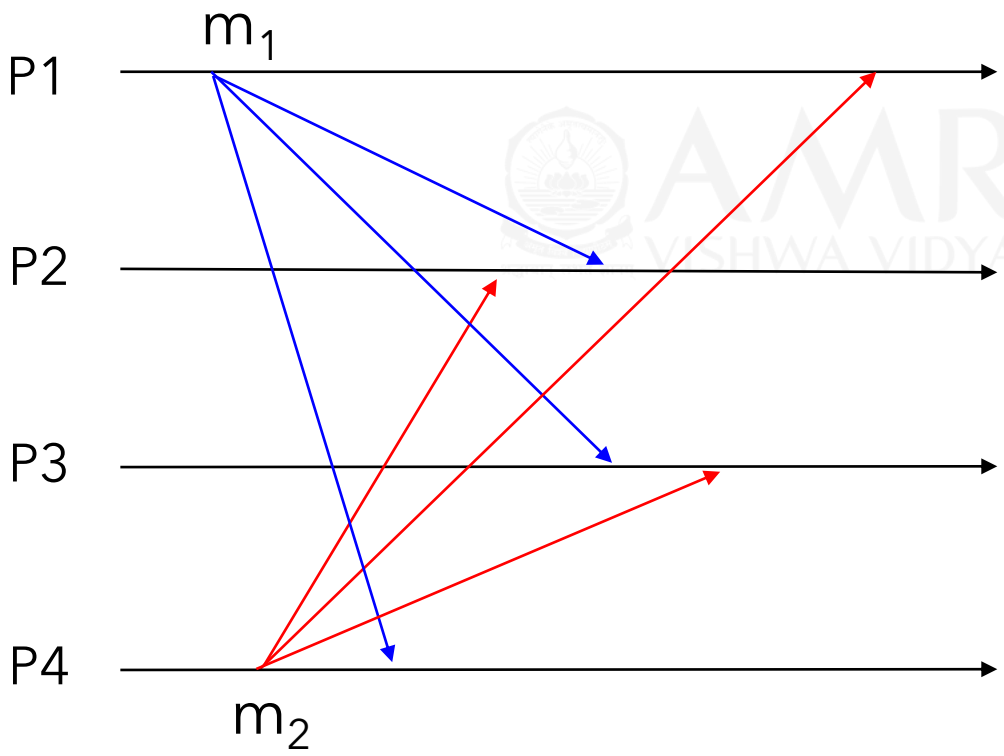
So, this is causally ordered trivially.

P1 and P4 receive only one message and hence is not a concern.

No Total order, but Causal order

Place your
Webcam Video here
Size 38%

- When the order of delivery to different processes are different, then total order does not exist.



- P2 and P3 receive the messages in different orders.
- P2: m2, m1 | P3: m1, m2
- Hence, no total order exists.

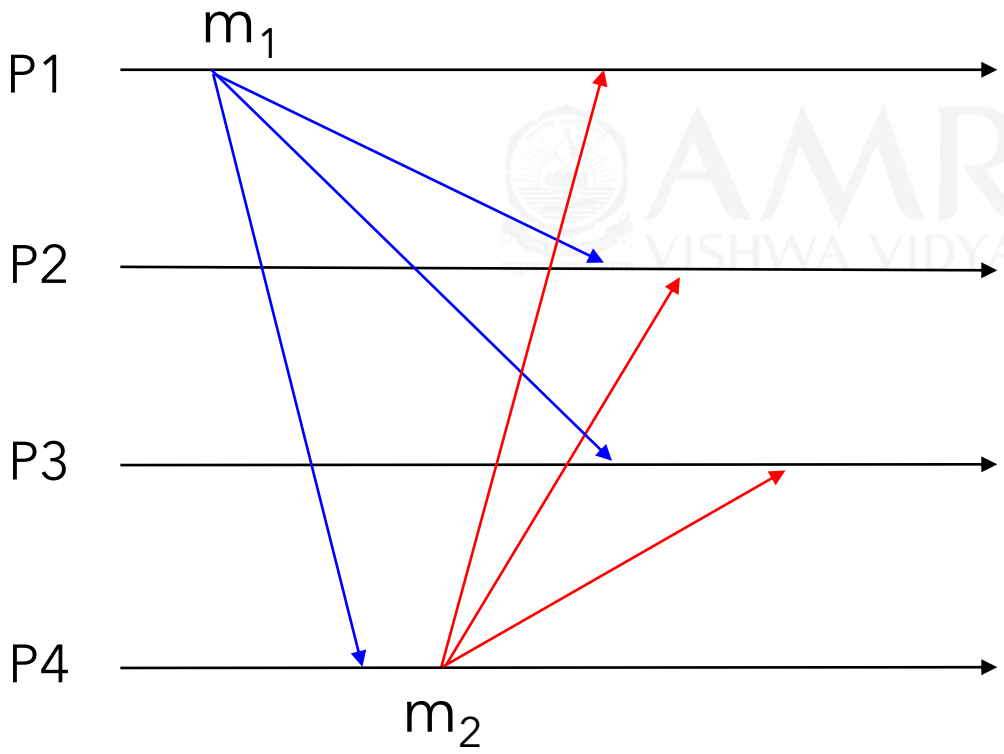
Note: m1 and m2 are concurrent messages. Didn't cause each other.

So, causal order exists trivially.

Both Total order & Causal order

Place your
Webcam Video here
Size 38%

- Let's consider a case when messages are causally related.



- Both P2 and P3 receive messages in the same order. First m1, then m2. Hence, total order exists.

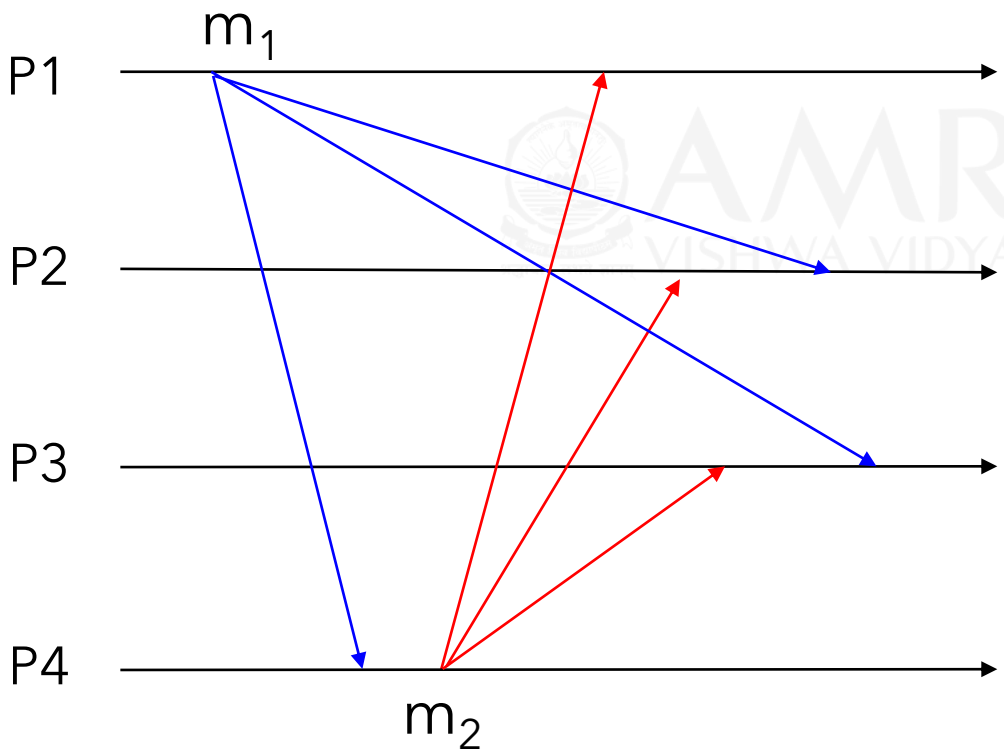
Note: m1 caused m2. Both P2 and P3 receive m1 and then m2. No causal break by either P1 or P2.

So, causal order exists.

Total order, but No Causal order

Place your
Webcam Video here
Size 38%

- Let's again consider a case when messages are causally related.



- Both P2 and P3 receive messages in the same order. First m2, then m1. Hence, total order.

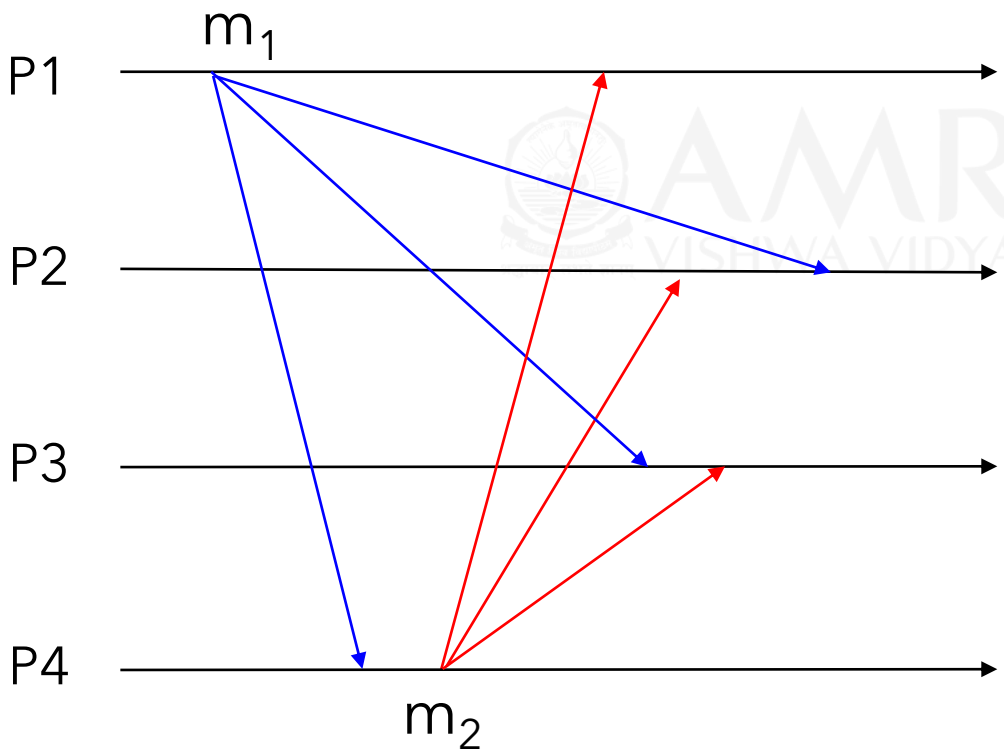
Note: m1 caused m2. But both P2 and P3 receive m2 and then m1.

So, this is not causally ordered.

Neither Total order Nor Causal order

Place your
Webcam Video here
Size 38%

- Let's again consider a case when messages are causally related.



- P2 and P3 receive messages in different order.
- P2: m2, m1 | P3: m1, m2
- Hence, no total order.

Note: m1 caused m2. But P2 receives m2 first and then m1.

So, this is not causally ordered.

Summary

- The table below gives summary of TO and CO in a multicast communication.

	All processes receive m1, then m2	All processes receive m2, then m1	At least one process receives in different order
m1 → m2	• Total Order • Causal Order	• Total Order • No Causal Order	• No Total Order • No Causal Order
m1 || m2	• Total Order • Causal Order	• Total Order • Causal Order	• No Total Order • Causal Order

Implementing message ordering

Place your
Webcam Video here
Size 38%

- Summary of approaches to implement different message ordering paradigms.

Ordering Paradigm	Implementation approach
Async order	Lamport's Scalar clock
FIFO order	Sequence numbering along each channel
Causal order	Raynal-Schiper-Toueg algorithm*
Sync order	Mutual exclusion, agreement algorithms
Total order	Three Phase Distributed algorithm*

* Will be dealt next.

Conclusion

- We discussed different message ordering paradigms.
  - Async order
  - FIFO order
  - Causal order
  - Sync order
- Multicast communication
  - Causal order
  - Total order

Place your
Webcam Video here
Size 100%