CLOUD COMPUTING CONCEPTS

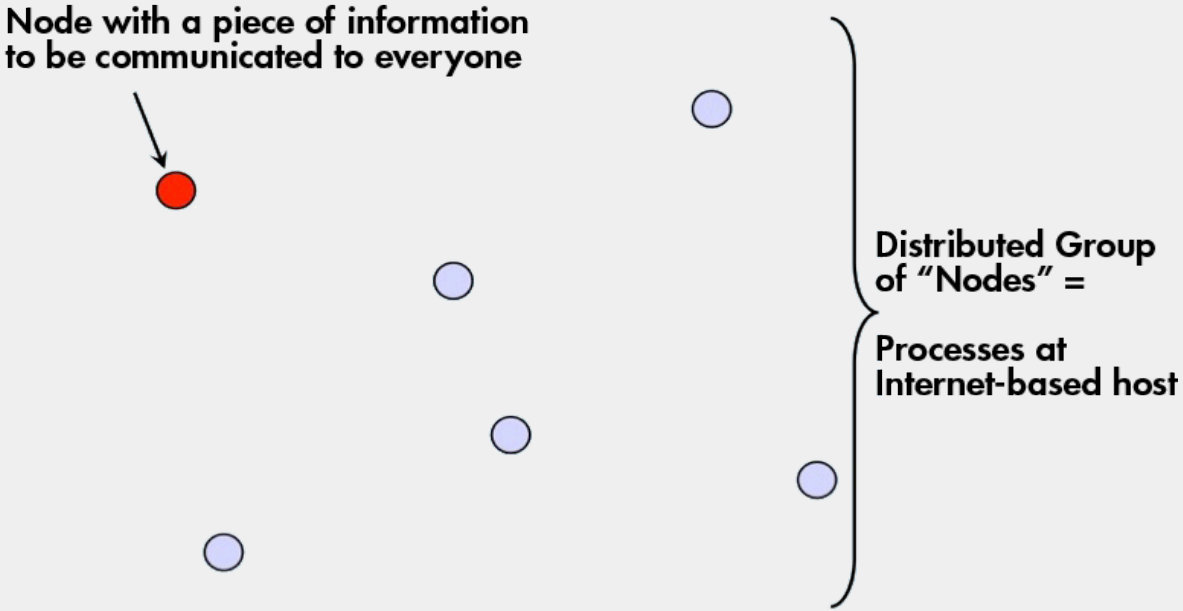
with Indranil Gupta (Indy)

MULTICAST

Lecture A

MULTICAST ORDERING

Multicast Problem



Other Communication Forms

- **Multicast** → message sent to a group of processes
- **Broadcast** → message sent to all processes (anywhere)
- **Unicast** → message sent from one sender process to one receiver process

Who Uses Multicast?

- A widely-used abstraction by almost all cloud systems
- Storage systems like Cassandra or a database
  - Replica servers for a key: Writes/reads to the key are multicast within the replica group
  - All servers: membership information (e.g., heartbeats) is multicast across all servers in cluster
- Online scoreboards (ESPN, French Open, FIFA World Cup)
  - Multicast to group of clients interested in the scores
- Stock exchanges
  - Group is the set of broker computers
  - Groups of computers for high-frequency trading
- Air traffic control system
  - All controllers need to receive the same updates in the same order

Multicast Ordering

- Determines the meaning of “same order” of multicast delivery at different processes in the group
- Three popular flavors implemented

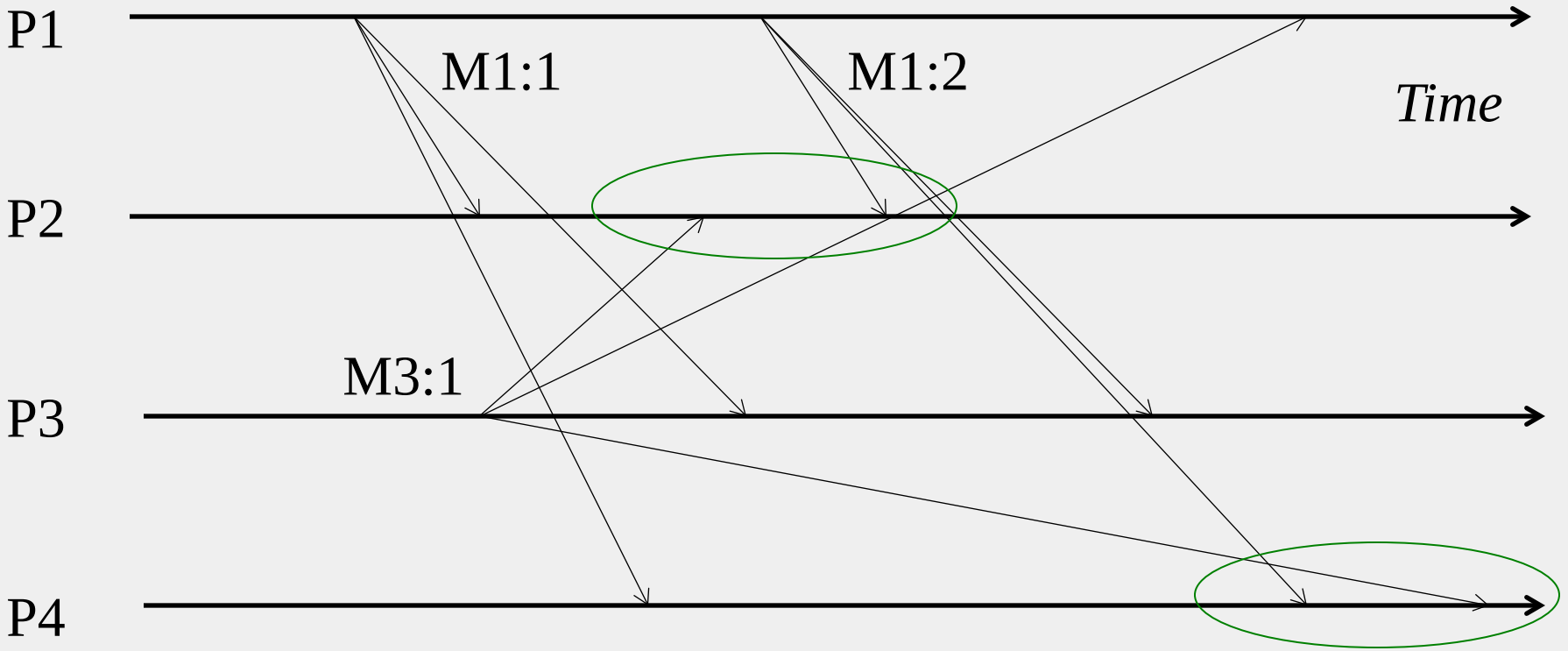
  by several multicast protocols

  1. FIFO ordering
  2. Causal ordering
  3. Total ordering

1. FIFO ordering

- Multicasts from each sender are received in the order they are sent, at all receivers
- Don’t worry about multicasts from different senders
- More formally
  - *If a correct process issues (sends) multicast(g,m) to group g and then multicast(g,m’), then every correct process that delivers m’ would already have delivered m.*

FIFO Ordering: Example



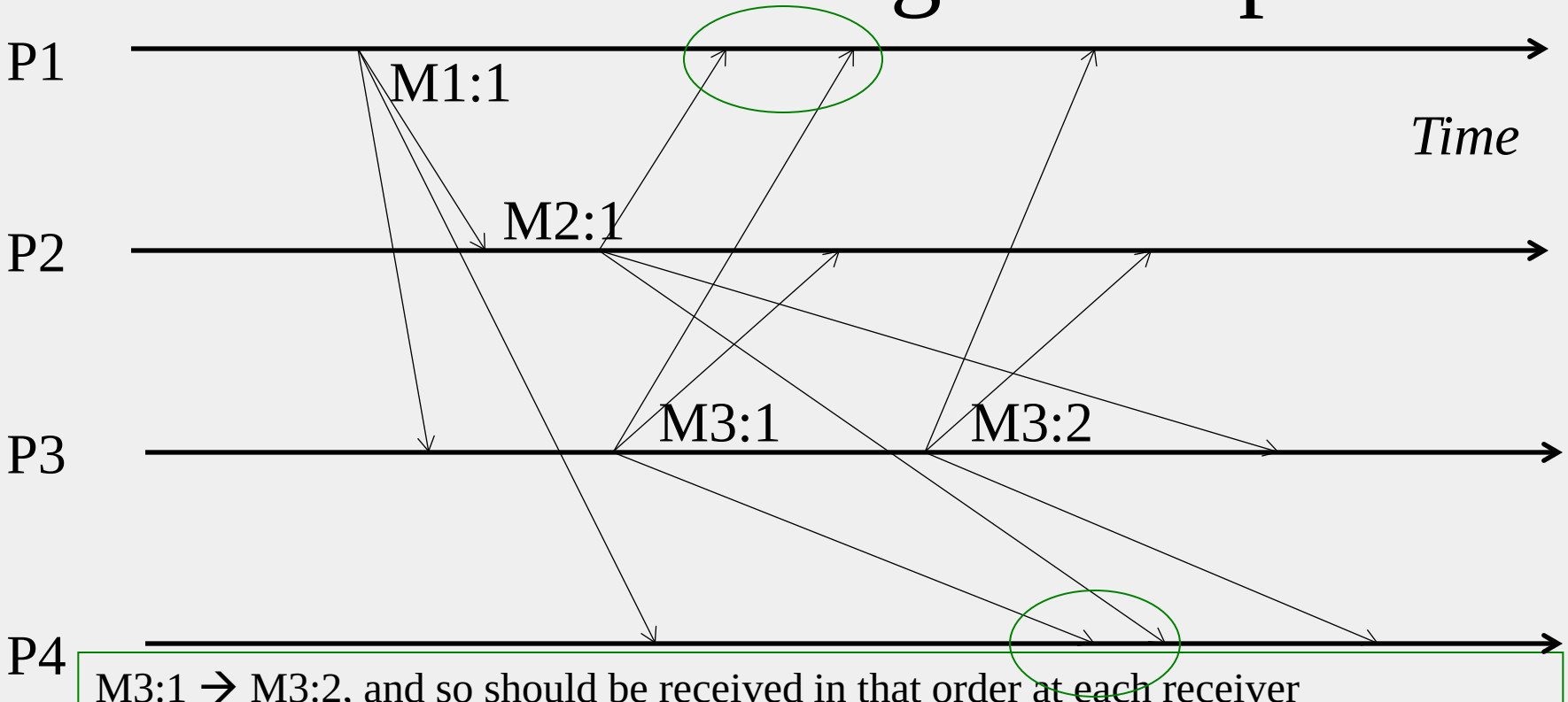
M1:1 and M1:2 should be received in that order at each receiver

Order of delivery of M3:1 and M1:2 could be different at different receivers

2. Causal Ordering

- Multicasts whose send events are causally related, must be received in the same causality-obeying order at all receivers
- Formally
  - *If multicast(g,m) → multicast(g,m′) then any correct process that delivers m′ would already have delivered m.*
  - *(→ is Lamport's happens-before)*

Causal Ordering: Example



M3:1 → M3:2, and so should be received in that order at each receiver

M1:1 → M3:1, and so should be received in that order at each receiver

M3:1 and M2:1 are concurrent and thus ok to be received in different orders at different receivers

Causal vs. FIFO

- Causal Ordering => FIFO Ordering
- Why?
  - If two multicasts M and M' are sent by the same process P, and M was sent before M', then M → M'
  - Then a multicast protocol that implements causal ordering will obey FIFO ordering since M → M'
- Reverse is not true! FIFO ordering does not imply causal ordering.

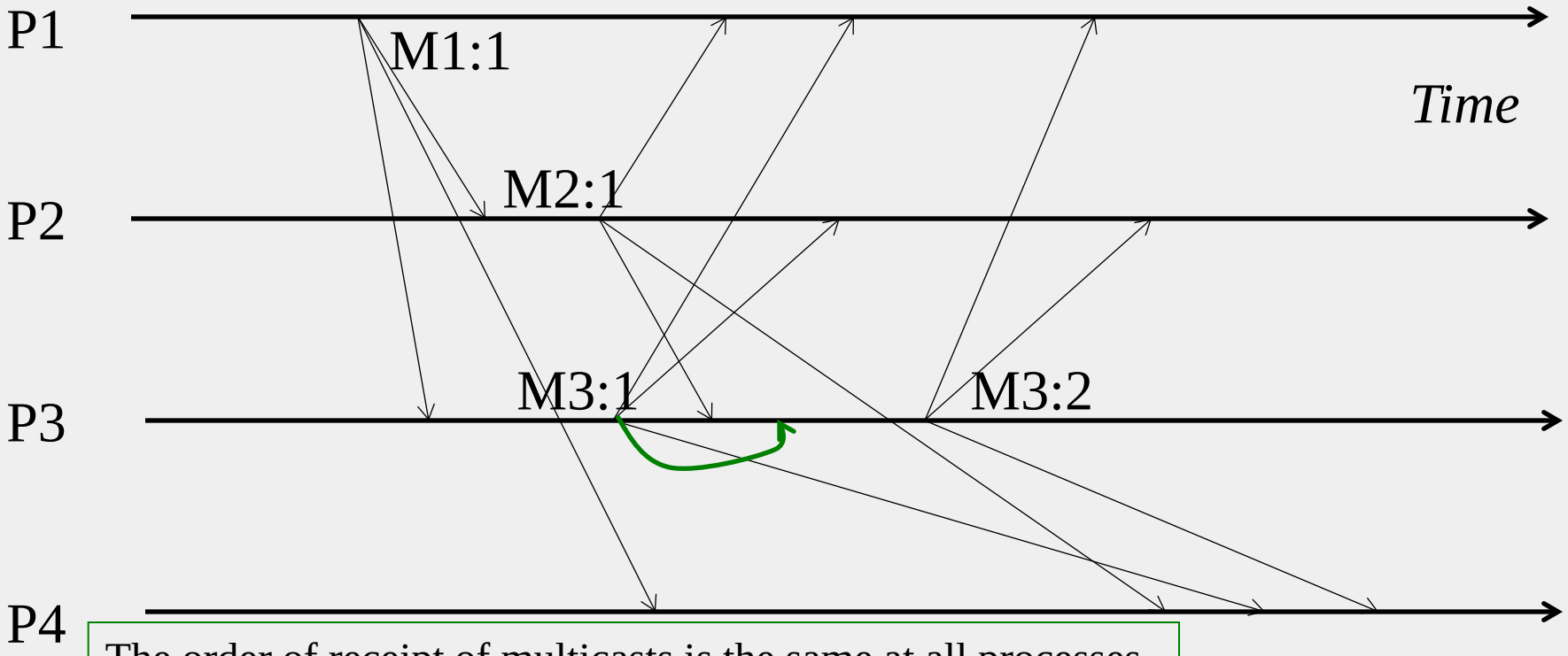
Why Causal at All?

- Group = set of your friends on a social network
- A friend sees your message m, and she posts a response (comment) m’ to it
  - If friends receive m’ before m, it wouldn’t make sense
  - But if two friends post messages m” and n” concurrently, then they can be seen in any order at receivers
- A variety of systems implement causal ordering: Social networks, bulletin boards, comments on websites, etc.

3. Total Ordering

- Also known as “Atomic Broadcast”
- Unlike FIFO and causal, this does not pay attention to order of multicast sending
- Ensures all receivers receive all multicasts in the same order
- Formally
  - *If a correct process P delivers message m before m′ (independent of the senders), then any other correct process P’ that delivers m′ would already have delivered m.*

Total Ordering: Example



The order of receipt of multicasts is the same at all processes.
M1:1, then M2:1, then M3:1, then M3:2
May need to delay delivery of some messages

Hybrid Variants

- Since FIFO/Causal are orthogonal to Total, can have hybrid ordering protocols too
  - FIFO-total hybrid protocol satisfies both FIFO and total orders
  - Causal-total hybrid protocol satisfies both Causal and total orders

Implementation?

- That was *what* ordering is
- But *how* do we implement each of these orderings?
- Next lecture