



CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

MULTICAST

Lecture C

IMPLEMENTING MULTICAST
ORDERING 2

CAUSAL ORDERING

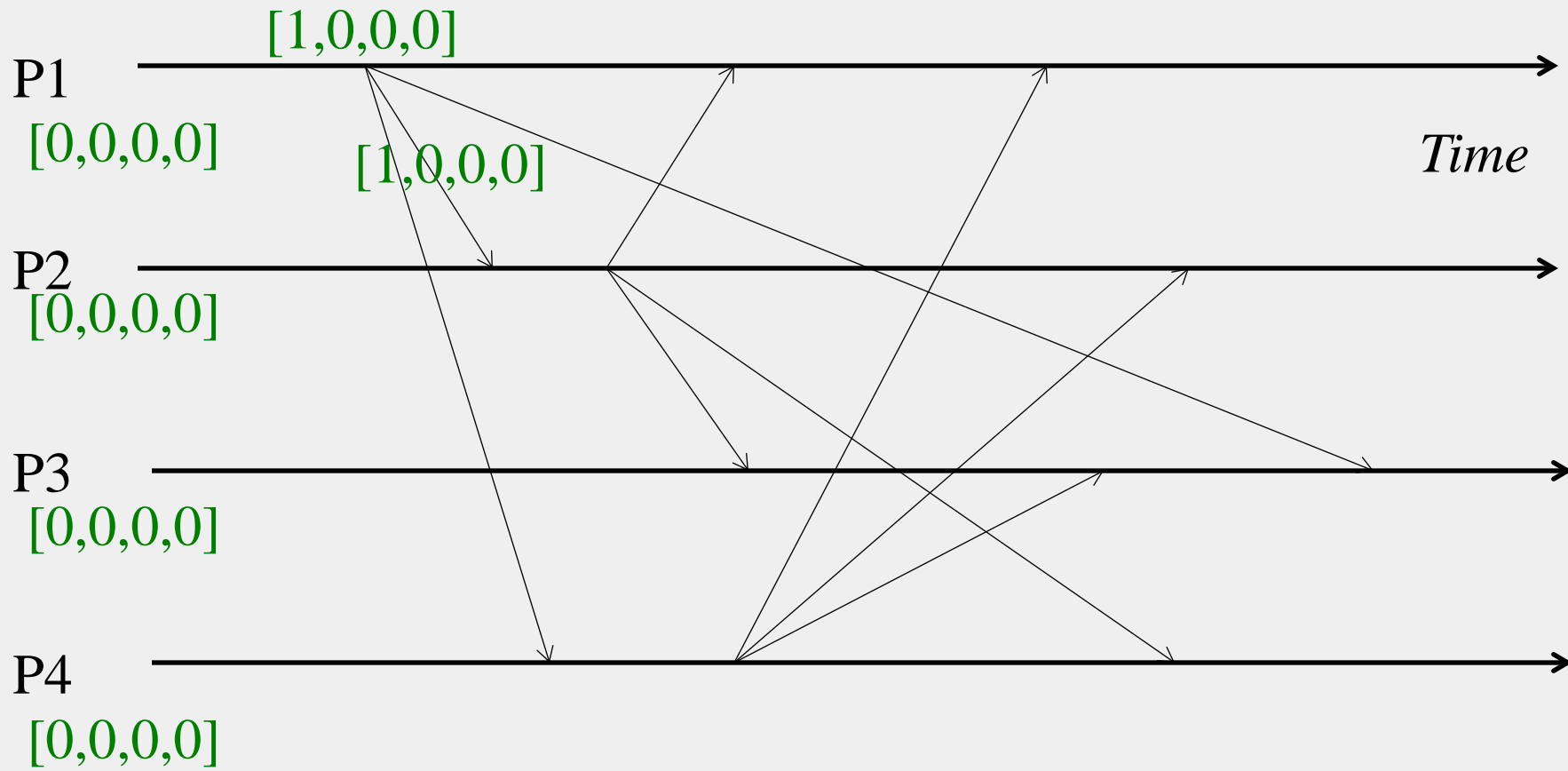
- Multicasts whose send events are causally related, must be received in the same causality-obeying order at all receivers
- Formally
 - *If $\text{multicast}(g, m) \rightarrow \text{multicast}(g, m')$ then any correct process that delivers m' would already have delivered m .*
 - *(\rightarrow is Lamport's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

CAUSAL MULTICAST: DATASTRUCTURES

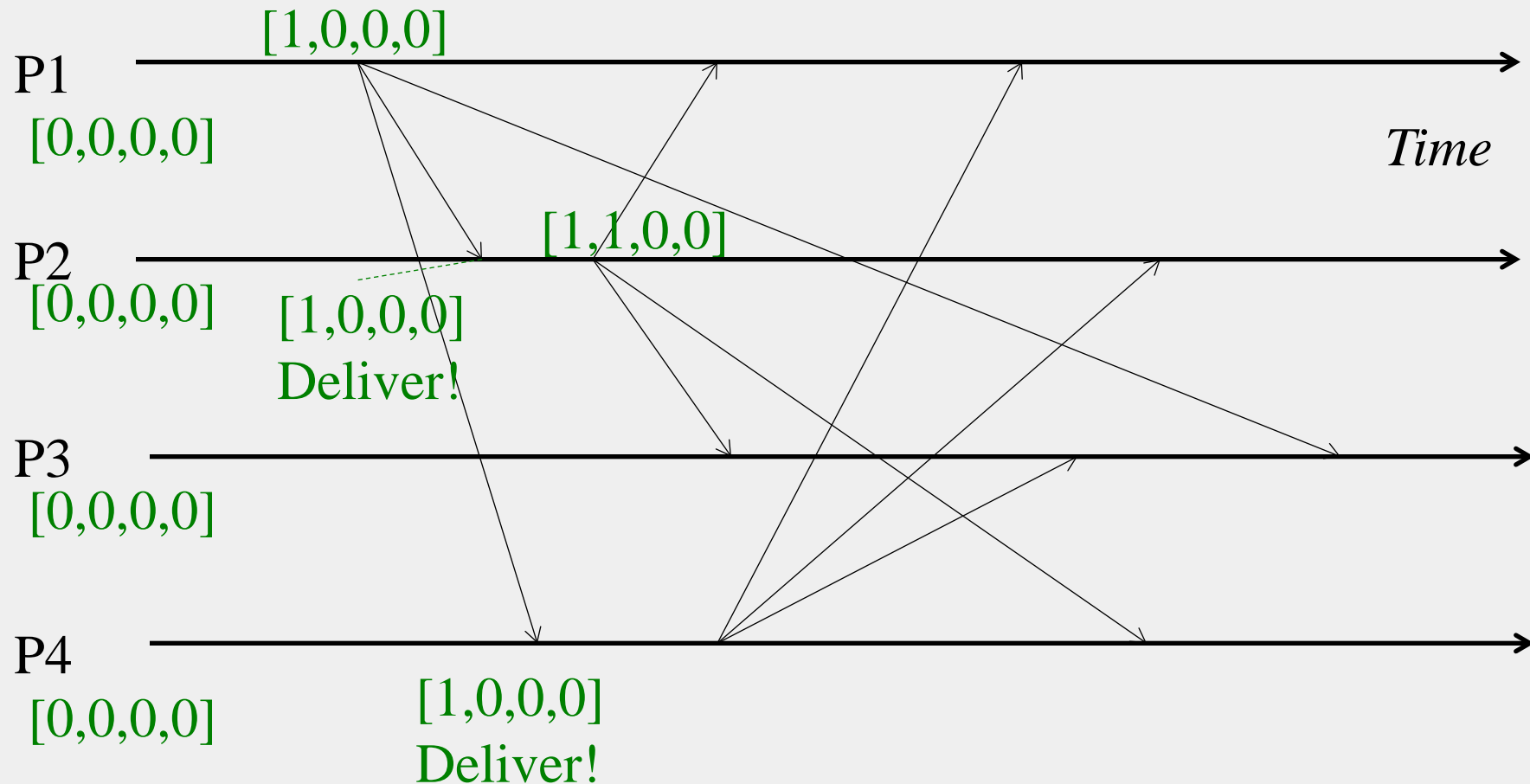
- Each receiver maintains a vector of per-sender sequence numbers (integers)
 - Similar to FIFO Multicast, but updating rules are different
 - Processes P_1 through P_N
 - P_i maintains a vector $P_i[1 \dots N]$ (initially all zeroes)
 - $P_i[j]$ is the latest sequence number P_i has received from P_j

CAUSAL MULTICAST: UPDATING RULES

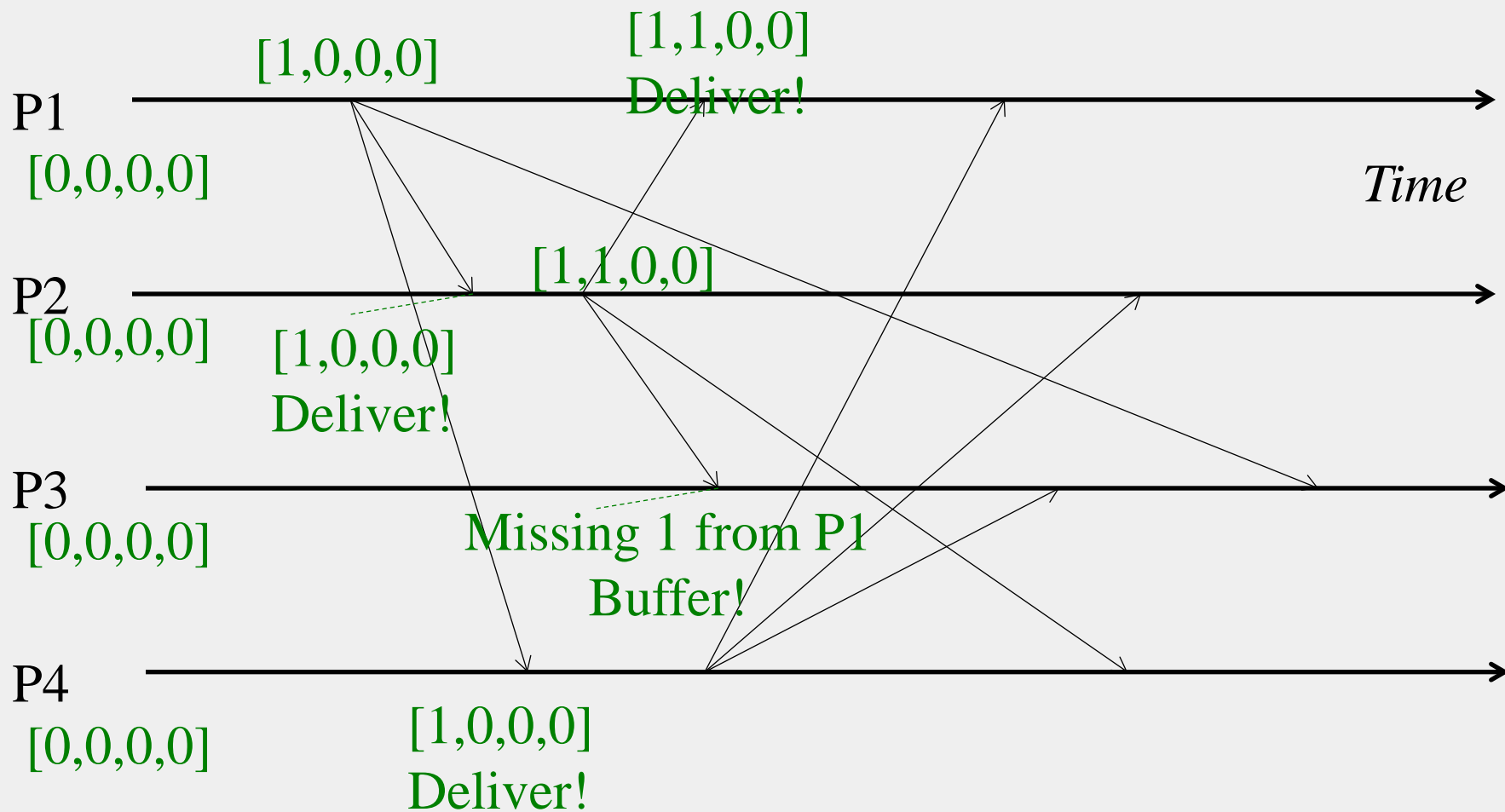
- Send multicast at process P_j :
 - Set $P_j[j] = P_j[j] + 1$
 - Include new entire vector $P_j[1 \dots N]$ in multicast message as its sequence number
- Receive multicast: If P_i receives a multicast from P_j with vector $M[1 \dots N]$ ($= P_j[1 \dots N]$) in message, buffer it until both:
 1. This message is the next one P_i is expecting from P_j , i.e.,
 - $M[j] = P_i[j] + 1$
 2. All multicasts, anywhere in the group, which happened-before M have been received at P_i , i.e.,
 - For all $k \neq j$: $M[k] \leq P_i[k]$
 - i.e., *Receiver satisfies causality*
 3. When above two conditions satisfied, deliver M to application and set $P_i[j] = M[j]$



Causal Ordering: Example

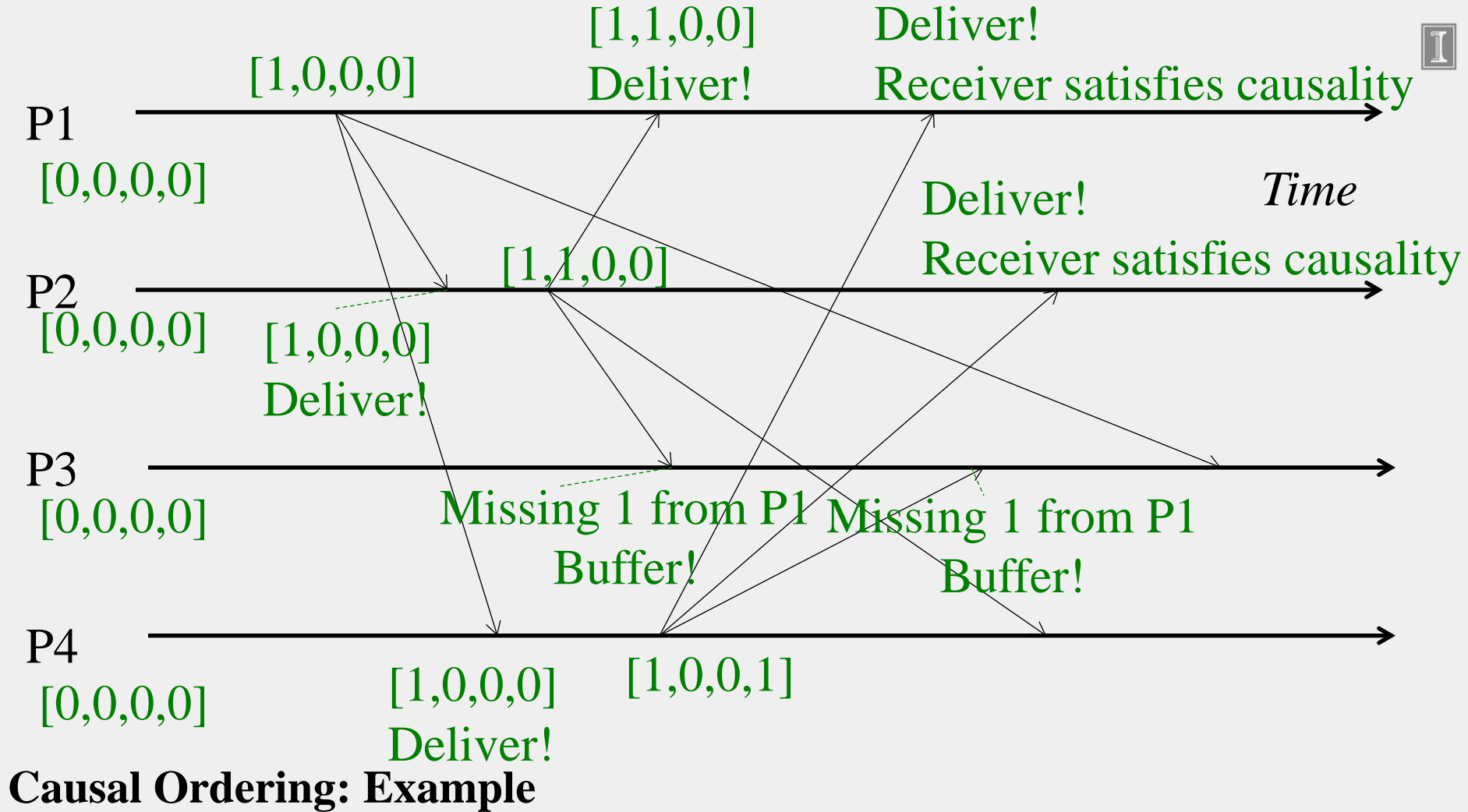


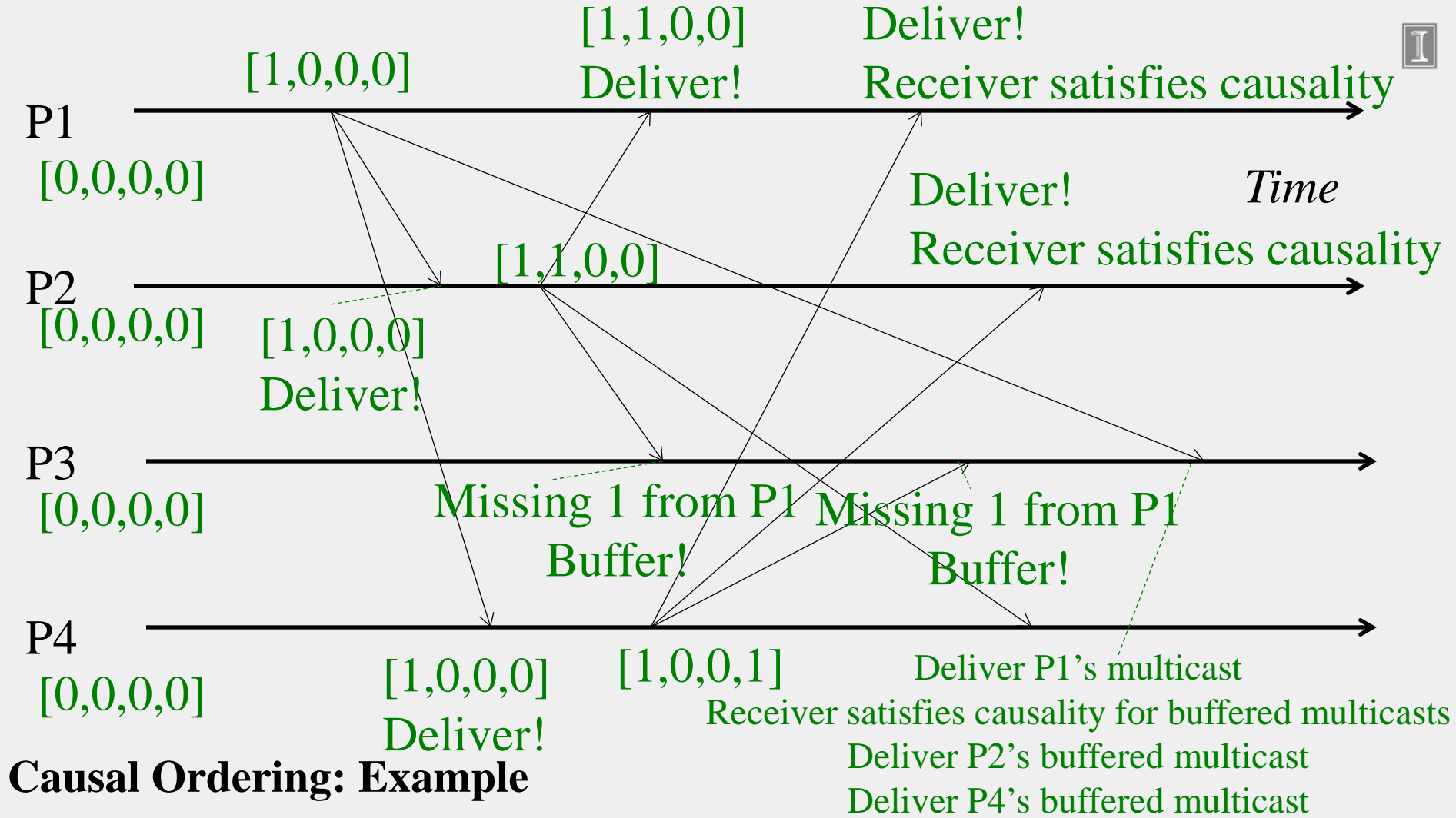
Causal Ordering: Example

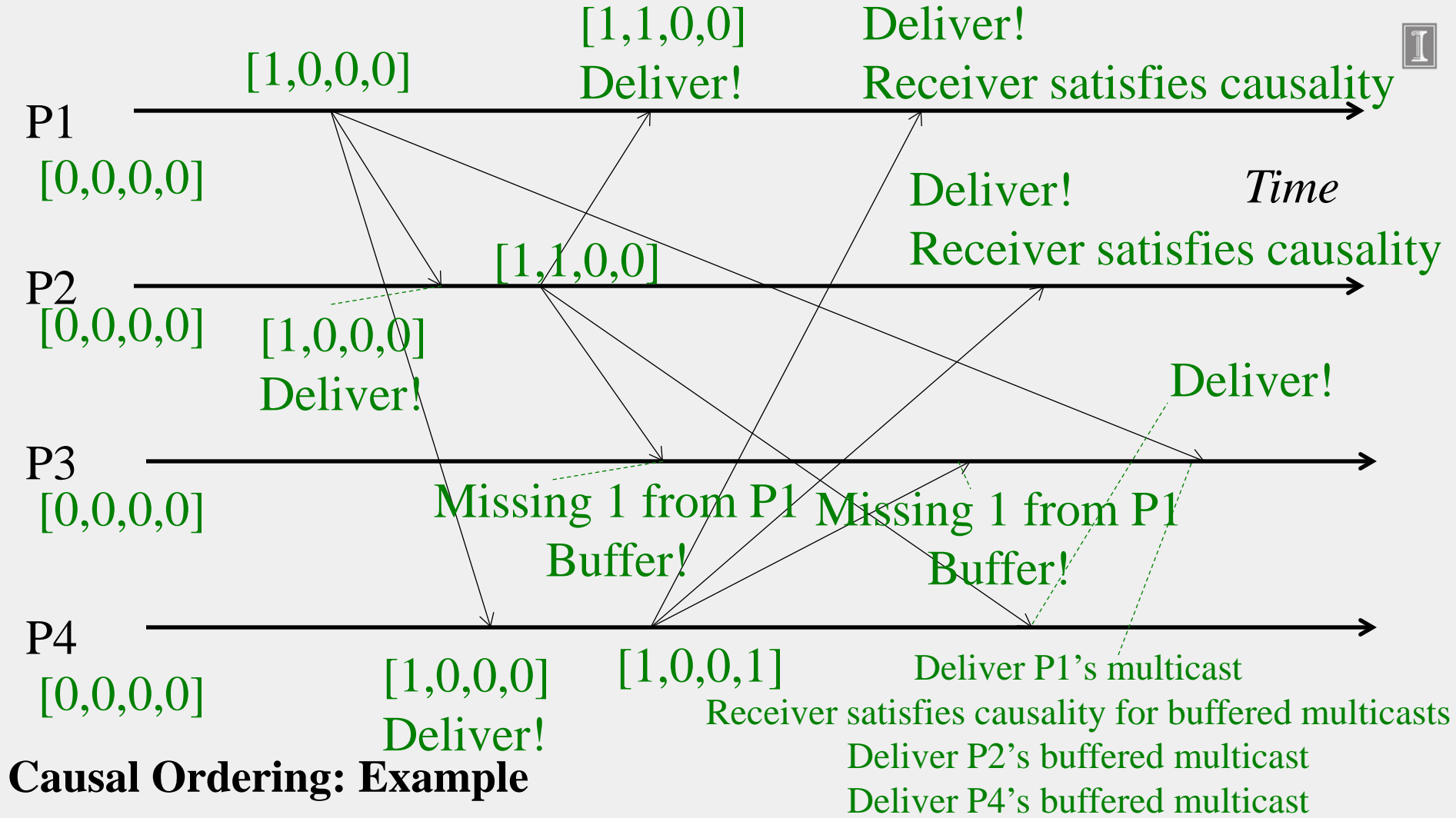


Causal Ordering: Example









SUMMARY: MULTICAST ORDERING

- Ordering of multicasts affects correctness of distributed systems using multicasts
- Three popular ways of implementing ordering
 - FIFO, Causal, Total
- And their implementations
- What about reliability of multicasts?
- What about failures?