

Lamport Timestamps

Lamport timestamps provide a method to **order events in a distributed system** without relying on physical clocks. They assign a **logical clock** to each event, ensuring that **causal relationships are respected**:

1. **Send event:** Sender increments its logical clock before sending a message.
2. **Receive event:** Receiver updates its clock using $C_{receiver} = \max(C_{receiver}, C_{sender}) + 1$.
3. **Local events:** Each process increments its clock for internal events.

This method guarantees the **happens-before relation** (\rightarrow):

- If event **a** happens before **b** ($a \rightarrow b$), then $Lamport(a) < Lamport(b)$.
- Concurrent events (no causal relationship) may have the same or unordered timestamps.

Applications:

- Ordering messages in distributed systems
- Detecting causal dependencies
- Synchronizing distributed logs

Python Implementation Notes:

- **clocks** array stores each process's current logical time
- **timestamps** array records send and receive times for all messages
- Output shows both **event timestamps** and **final clock values per process**

Conclusion: Lamport timestamps provide a **lightweight and effective way to impose partial ordering** in distributed systems without needing synchronized physical clocks.

