

# Time, Clocks, and the Ordering of Events in a Distributed System

Distributed system transmission delay is not negligible compared to time between events in a node/process. Partial ordering, "happens before" " $\rightarrow$ " if two events occur after another on the same process/nodes or are related by message transmission i.e. a is transmission and b is receiving event. Partial order is transitive. Happens before can mean a causally affects b. Logical clocks, counters. Lamport clocks: clock condition if  $a \rightarrow b$  then  $c(a) < c(b)$ , Converse not true. Concurrent if  $a \not\rightarrow b$  and  $b \not\rightarrow a$ . Take max when message exchange occurs. Total ordering can add some conditions to break ties, this will not be unique and will depend on the system. Also cannot be dependent on to imply causal relationship. (Note vector clocks help with the converse condition

[https://www.youtube.com/watch?v=x-D8iFU1d-o&list=PLeKd45zvjcDFUEv\\_ohr\\_HdUFe97RitdiB&index=11](https://www.youtube.com/watch?v=x-D8iFU1d-o&list=PLeKd45zvjcDFUEv_ohr_HdUFe97RitdiB&index=11) . Distributed algorithm example for resource consumption/mutual exclusion.

Need to introduce physical time to prevent anomalies than can come when considering external system events.

Ordering reference: [https://www.isical.ac.in/~ansuman/dist\\_sys/Lecture1.pdf](https://www.isical.ac.in/~ansuman/dist_sys/Lecture1.pdf).