PAST AND FUTURE CONES OF AN EVENT

In a distributed computation, an event e_j could have been affected only by all events e_i, such that $e_i \rightarrow e_j$ and all the information available at e_i could be made accessible at e_j. In other word e_i and e_j should have a causal relationship. Let Past(e_j) denote all events in the past of e_j in any computation.

$$Past(e_j) = \{e_i | \forall e_i \in H,\ e_i \rightarrow e_j\}$$

The term max(past(e_i)) denotes the latest event of process p_i that has affected e_j. This will always be a message sent event.

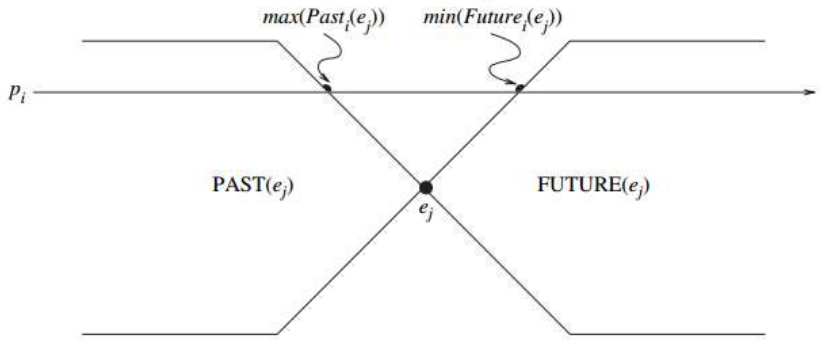

Fig : Past and future cones of event

A cut in a space-time diagram is a line joining an arbitrary point on each process line that slices the space-time diagram into a PAST and a FUTURE. A consistent global state corresponds to a cut in which every message received in the PAST of the cut was sent in the PAST of that cut.

The future of an event $e_j$denoted by Future(e_j) contains all the events e_i that are casually affected by e_j.

$$Future(e_j) = \{e_i | \forall e_i \in H,\ e_j \rightarrow e_i\}.$$

Future$_i$(e_i) is the set of those events of Future (e_j) are the process p_i and min(Future$_i$(e_j)) as the first event on process p_i that is affected by e_j. All events at a process p_i that occurred afterMax(Past(e_j)) but before min(Future$_i$(e_j)) are concurrent with e_j.

MODELS OF PROCESS COMMUNICATIONS

There are two basic models of process communications

Synchronous: The sender process blocks until the message has been received by the receiver process. The sender process resumes executiononly after it learns that the receiver process has accepted the message. The sender and the receiver processes must synchronize to exchange a message.