Types of messages:

1. In-transit message:
   messages that have been sent but not yet received.

2. Lost message:
   msg whose 'send' is done but 'receive' is undone due to rollback.

3. Delayed msg:
   msg whose 'receive' is not recorded because the receiving process was either down or the message arrived after rollback.

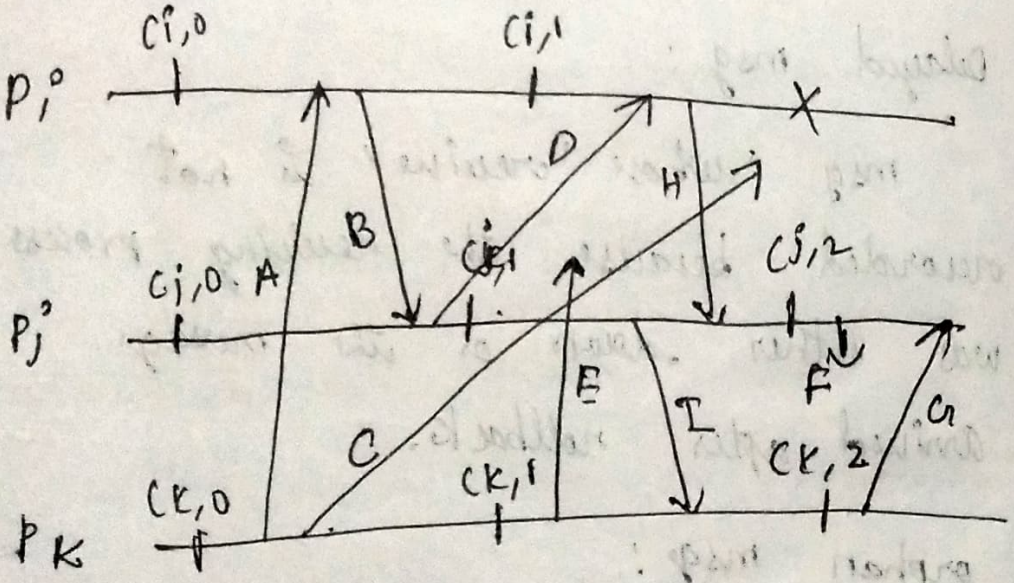
4. orphan msg:
   msg with "receive" recorded but message 'send' not recorded.
   DO not arise if processes roll back to a consistent global state.

5. Duplicate msg: arise due the message logging and replaying during process recovery.

Issues in failure recovery:

In a failure recovery, we must not only restore the system to a consistent state, but also handle message in failure recovery.

Diagram:



The 3 process Pi, Pj, Pk are connected through a communication network;

The processes communicate solely by exchanging msg over fault-free, FIFO comm. channels

> checkpoints : $\{C_{i,0}, C_{i,1}\}$, $\{C_{j,0}, C_{j,1}, C_{j,2}\}$ and $\{C_{k,0}, C_{k,1}, C_{k,2}\}$.

> Messages : A – J.

> The restored global consistent state :

$\{C_{i,1}, C_{j,1}, C_{k,1}\}$.

> - Message G : delayed msg.

- Message D : A lost message since the sent event for D is recorded in Restored state

- Msg E, F : delayed orphan msg : After resuming execution from their checkpoints, processes will generate both of these msg.

- Lost msg can be handled by having process keep a message log of all the sent message.

- Message H & I : Created due to the rollback of process C to checkpoint C_1. (orphan Msg).

Handling Issues :

* Delayed Msg : Mechanisms must ensure that delayed msg do not cause inconsistencies.

* Lost Msg : Implementing msg log helps recover lost msg.

* orphan Msg : processes should handle orphan msg by discarding or reintegrating them based on the current state.

strategies :

* consistent state Restoration
* Message logging
* checkpoint coordination.