CONSENSUS PROBLEM :—

DISTRIBUTED SYSTEMS
- Reliable Multicast
- Membership / Failure Detection
- Leader Election
- Mutual Exclusion

* COMMON : Coordinate and agree on a value of some parameter

CONSENSUS :—
- Validity
- Integrity
- Non-Triviality

parameter:
- Leader
- Status of failed process
- Ordering of messages
- Access to critical resource

SYSTEM MODELS

More general

SYNCHRONOUS
- Time bound on messages
- Drift bound
- Each process is bounded

Consensus Solution — Possible

ASYNCHRONOUS
- No bound on Process execution
- No bound on drift
- No bound on message transmission delays

Consensus Solution — Impossible

Popular Solutions: PAXOS / SIMPLY

* Use time slicing or analyse different periods of rounds based on above bounds
⇒ Reverse induction on faulty nodes

PAXOS :

- **SAFETY** → No 2 processes decide different values
- **EVENTUAL LIVENESS** → Consensus reached eventually

* Move to next round → if timeout or message from next round

PHASE-1 : Leader Election

⇒ Potential leader chooses unique ballot id and send to all processes
  ↳ Higher than anything seen

⇒ Processes wait and respond to highest ballot id

⇒ Process includes v in the response

⇒ If Quorum respond OK, then you are the leader
  ↳ If failed quorum, start next round

* Potential leader withdraws if it sees higher value

PHASE-2 : Proposal (Bill)

→ Leader sends proposed value v to all
  - use v=v' if some process already decided in a previous round

→ Recipients log on disk and respond OK

PHASE-3 : Decision (LAW)

→ If leader has majority of OK, it sends decision to all recipients

To ensure safety → Previous and new quorum intersect and hence same value used by new quorum

Eventual liveness

- Leader fails
- Messages dropped
- Process fails