BLOCKCHAINS

ARCHITECTURE, DESIGN AND USE CASES

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

PRAVEEN JAYACHANDRAN
IBM RESEARCH,
INDIA

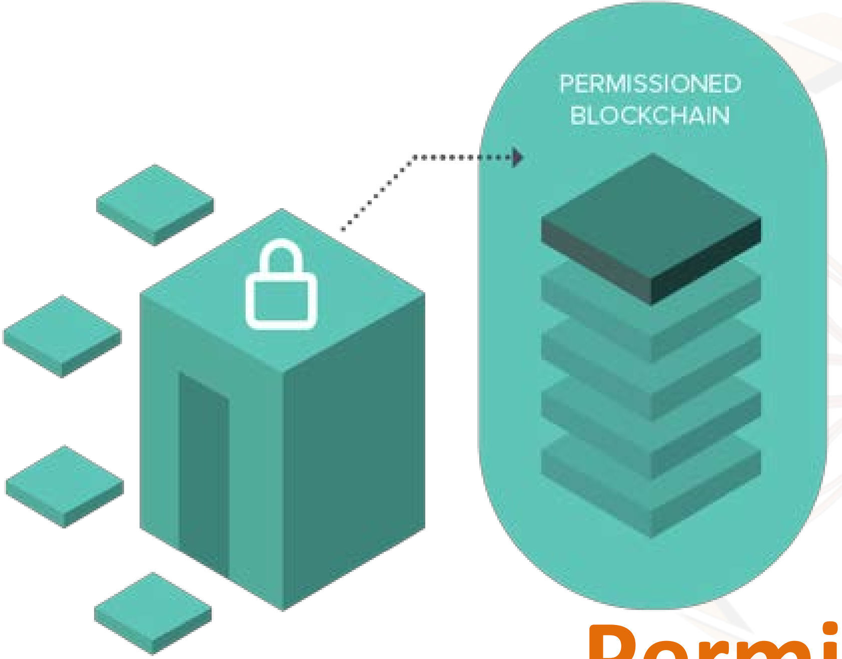
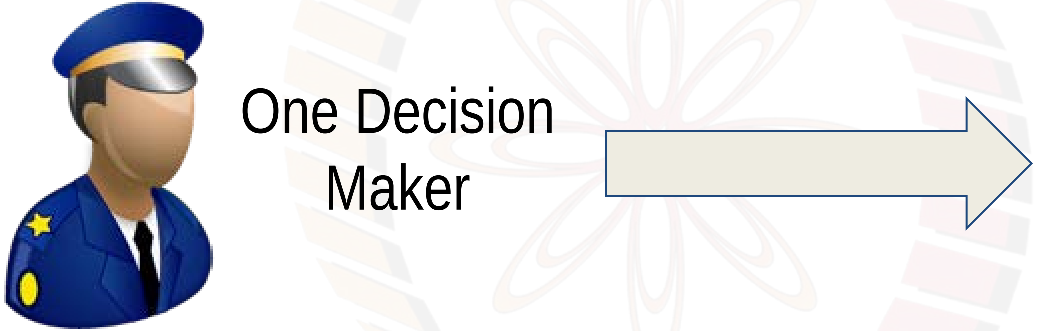


Image Source: https://nem.io/enterprise/

Permissioned Blockchain - II
Consensus Algorithms

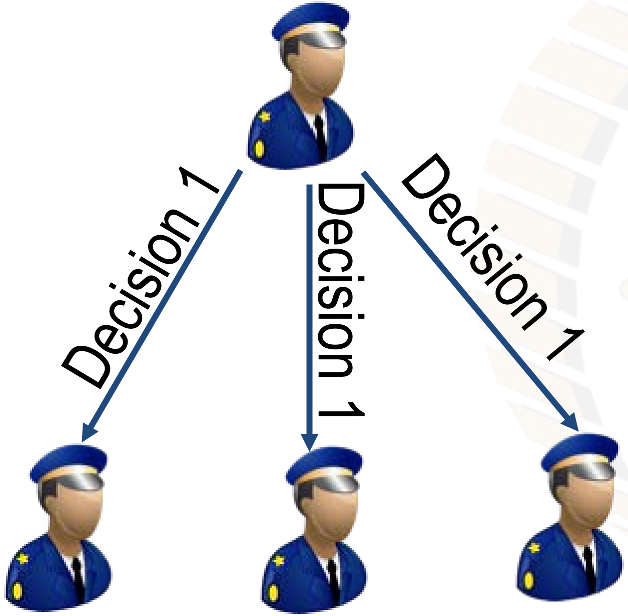
Why Distributed Consensus



One Decision Maker

No Consensus

Why Distributed Consensus



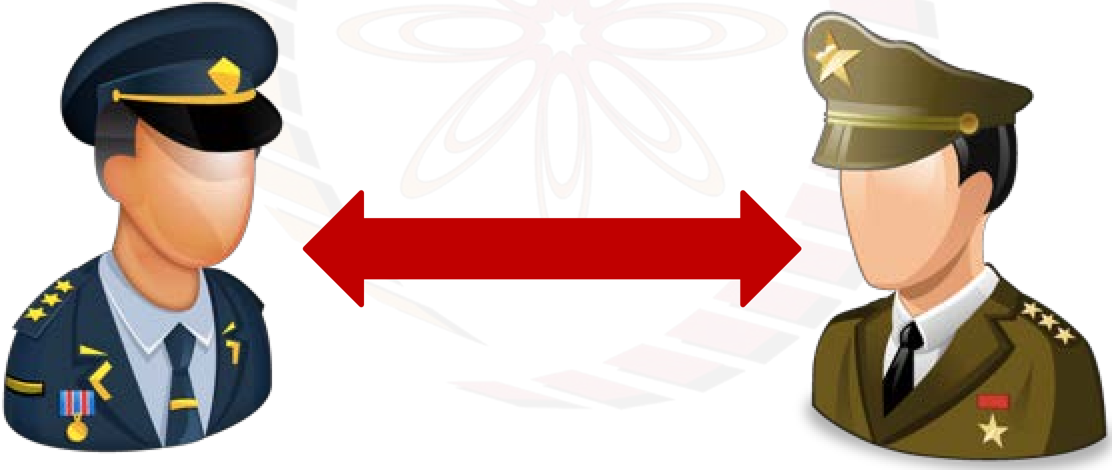
Multiple Decision Maker → Consensus Required

Why Distributed Consensus

- Reaching agreement in distributed computing
- Replication of common state so that all processes have same view
- Applications:
  - Flight control system: E.g. Boeing 777 and 787
  - Fund transferring system: Bitcoin and cryptocurrencies
  - Leader election/Mutual Exclusion

Why Distributed Consensus

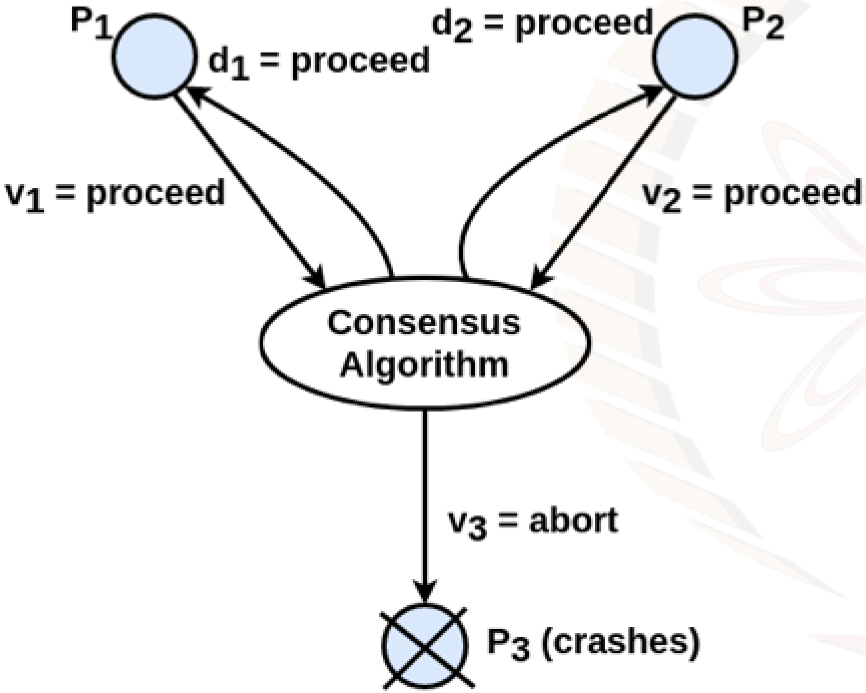
- So, no need of consensus in a single node process.
- What about when there are two nodes?
  - Network or partitioned fault, consensus cannot be reached



Faults in Distributed Consensus

- Crash Fault
- Network or Partitioned Faults
- Byzantine Faults
  - malicious behaviour in nodes
  - hardware fault
  - software error

Consensus for three processes



- Each process P_i (i=1,2,...N):
  - **Undecided state:** proposed value v_i from set D
  - **Communication state:** exchange values
  - **Decided state:** set decision variable d_i

Requirements of a Consensus Algorithm

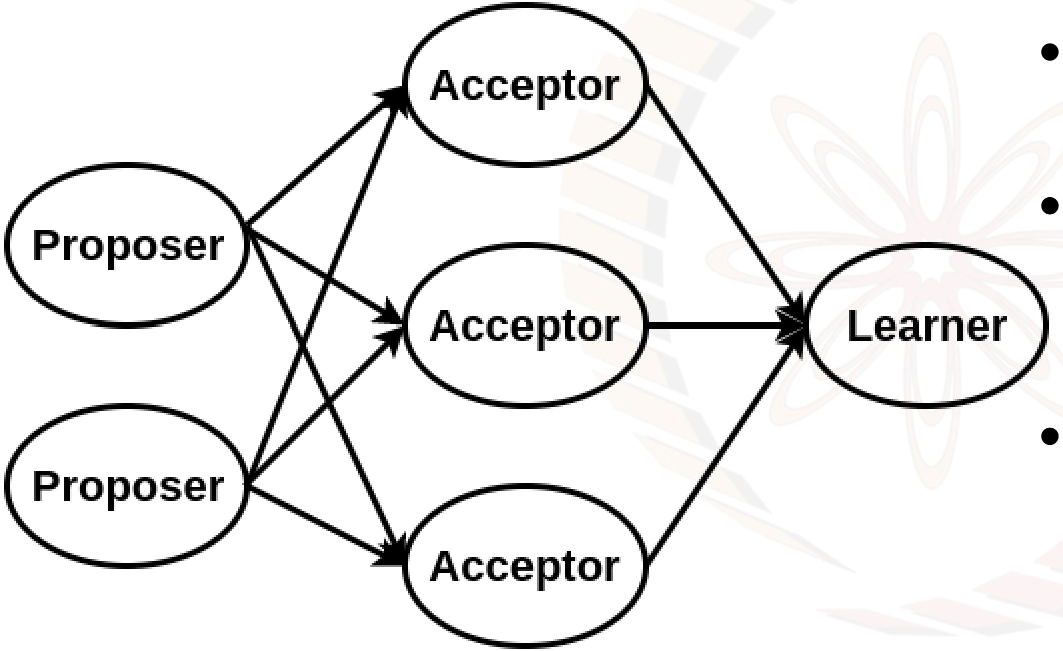
- Termination:
  - Eventually each correct process sets its decision variable
- Agreement:
  - The decision value of all correct processes is the same
- Integrity:
  - If the correct processes all proposed the same value, then any correct process in the decided state has chosen that value

Different Algorithms

- Crash or Network Faults:
  - PAXOS
  - RAFT

- Byzantine Faults (including Crash or Network Failures):
  - Byzantine fault tolerance (BFT)
  - Practical Byzantine Fault Tolerance (PBFT)

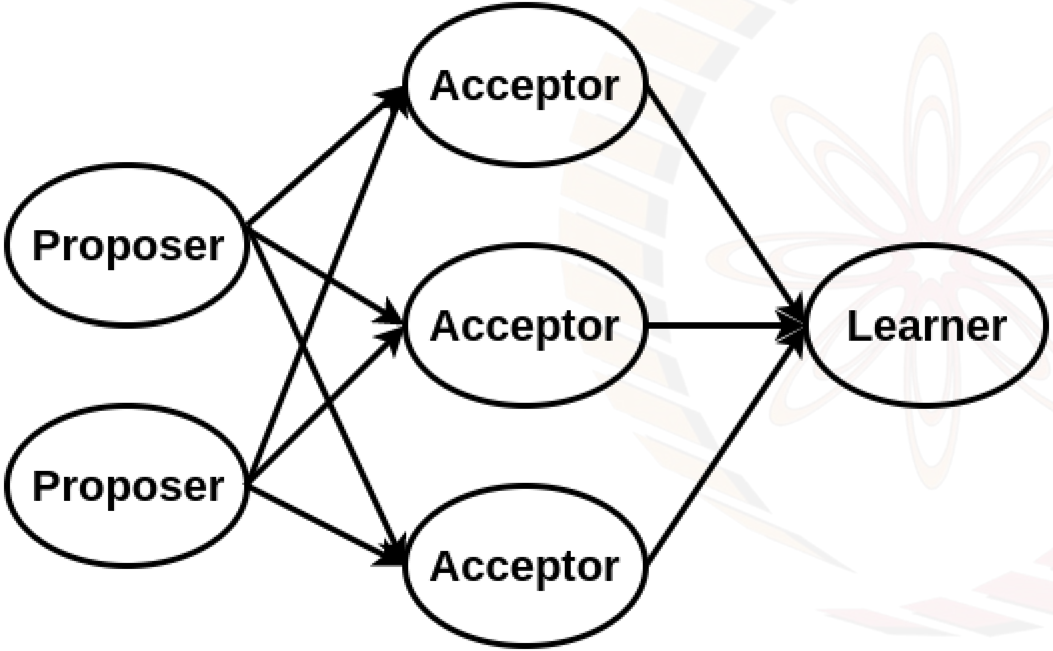
PAXOS



Source: Lamport, Leslie. "Paxos made simple." ACM Sigact News 32.4 (2001): 18-25.

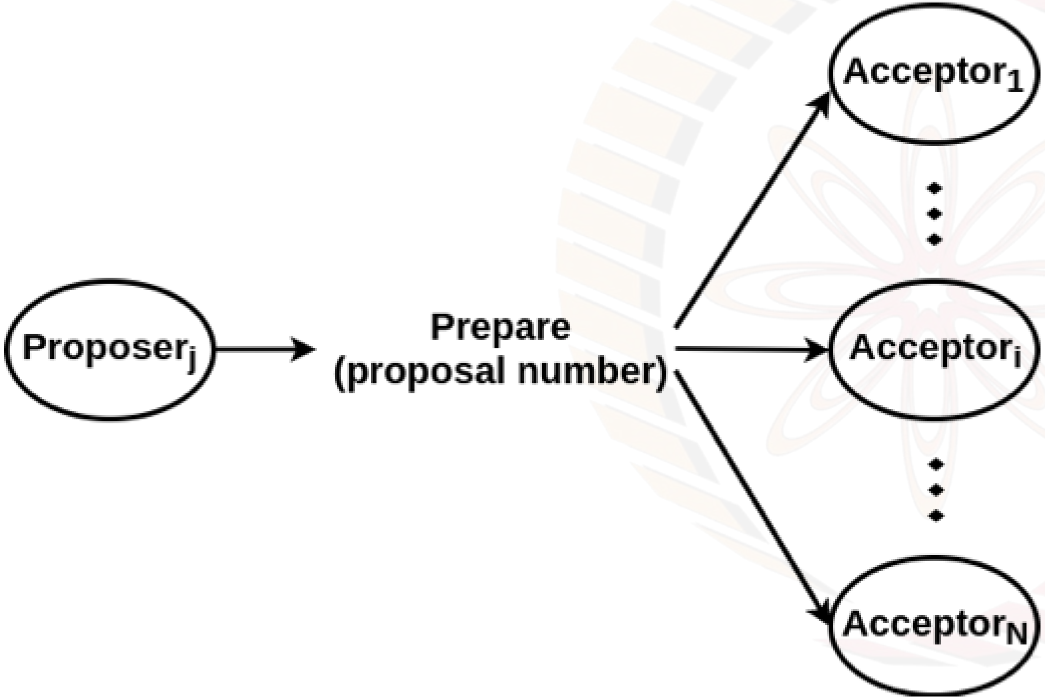
- First Consensus Algorithm proposed by L. Lamport in 1989
- Objective: choosing a single value under crash or network fault
- System process
  - Making a proposal
  - Accepting a value
  - Handling Failures

PAXOS: Types of Nodes



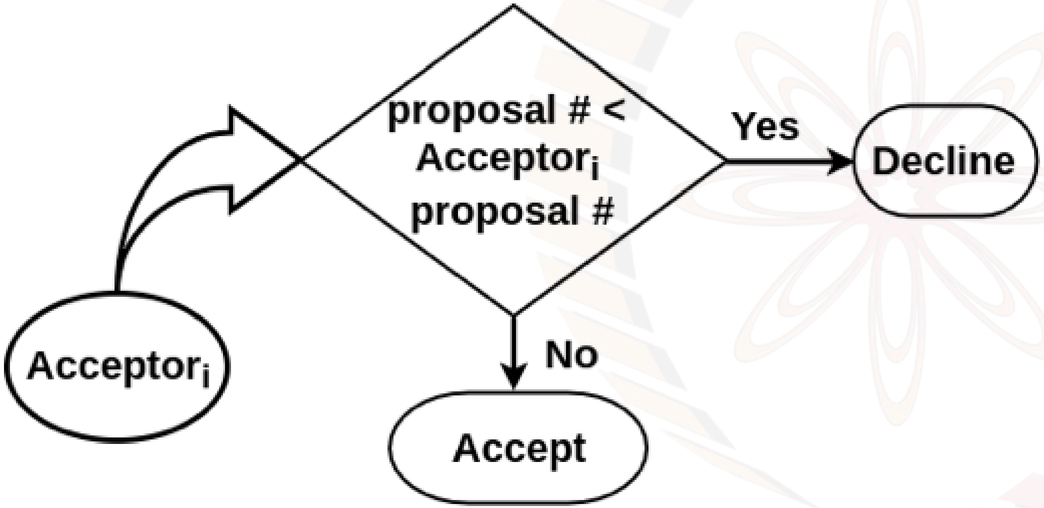
- **Proposer:** propose values that should be chosen by the consensus
- **Acceptor:** form the consensus and accept values
- **Learner:** learn which value was chosen by each acceptor

Making a Proposal: Proposer Process



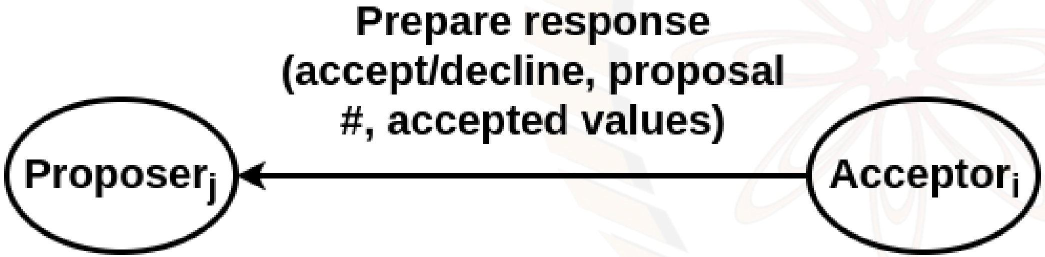
- **proposal number:** form a timeline, biggest number considered up-to-date

Making a Proposal: Acceptor's Decision Making



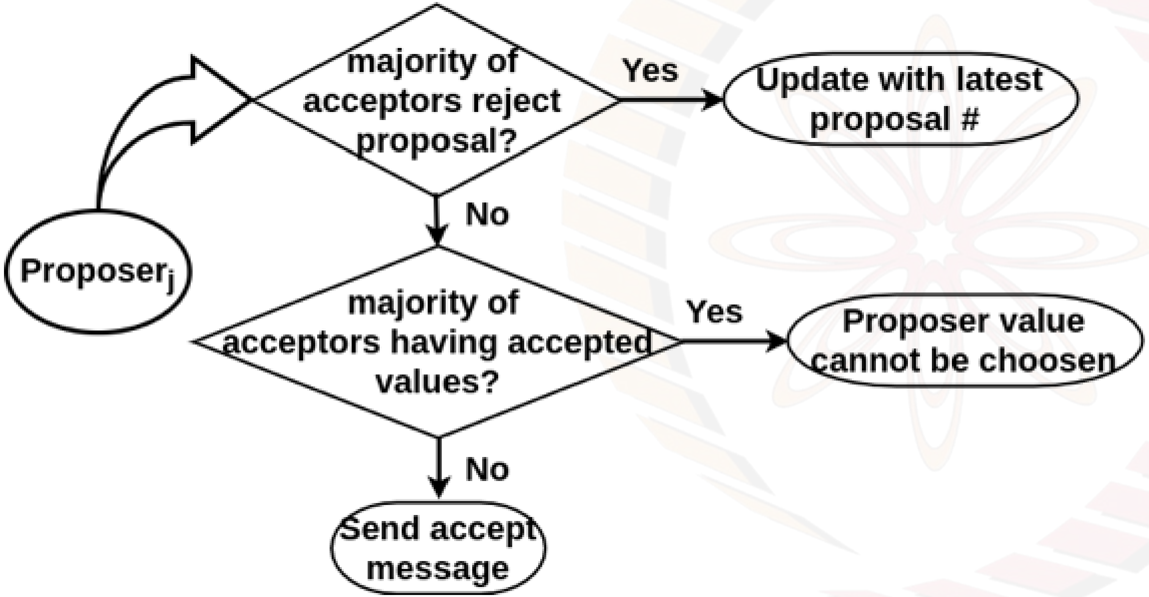
- Each acceptor compares received proposal number with the current known values for all proposer's prepare message

Making a Proposal: Acceptor’s Message


Prepare response (accept/decline, proposal #, accepted values)
Proposer_j
Acceptor_i

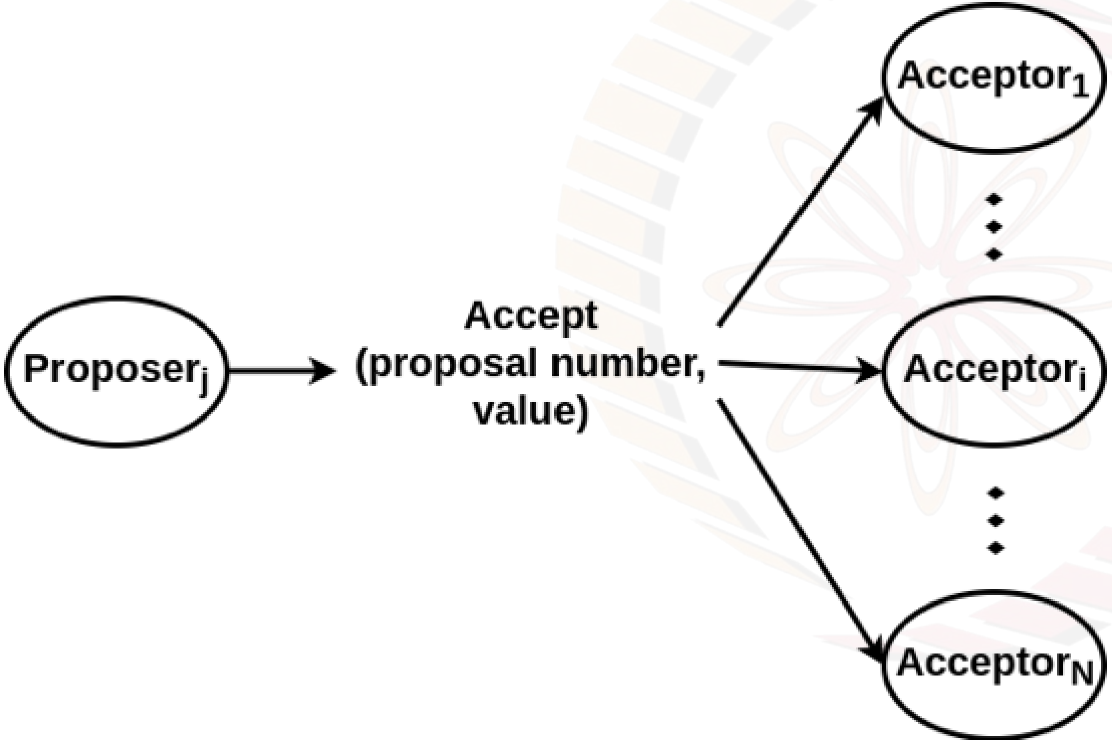
- **accept/decline:** whether prepare accepted or not
- **proposal number:** biggest number the acceptor has seen
- **accepted values:** already accepted values from other proposer

Accepting a Value: Proposer's Decision Making


Proposer$_j$
majority of acceptors reject proposal?
Yes
Update with latest proposal #
No
majority of acceptors having accepted values?
Yes
Proposer value cannot be choosen
No
Send accept message

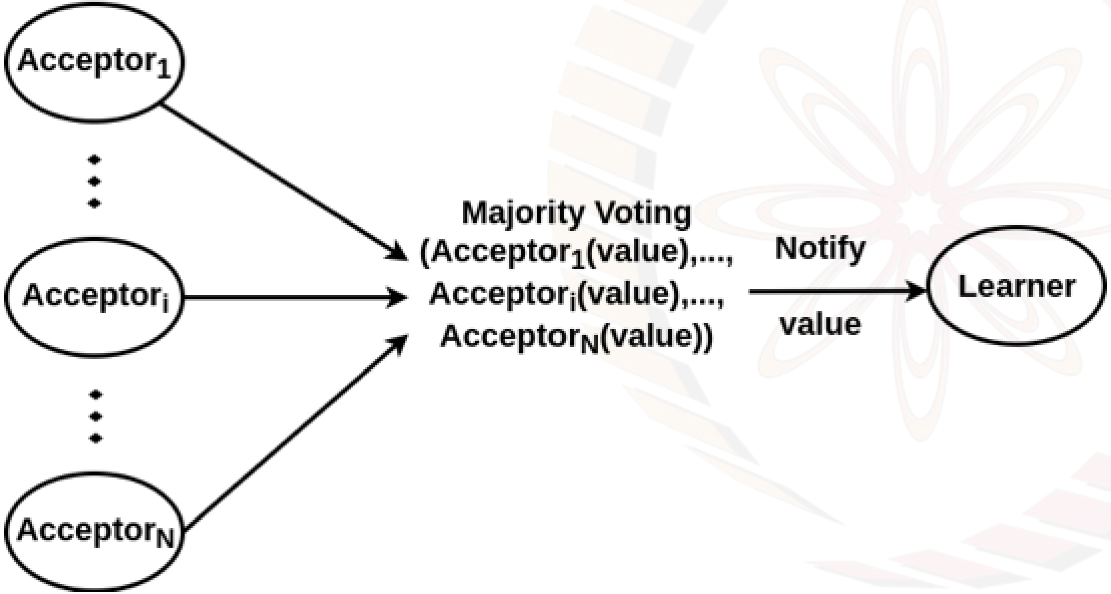
- Proposer receive a response from majority of acceptors before proceeding

Accepting a Value: Accept Message



- **proposal number:** same as prepare phase value
- **value:** single value proposed by proposer

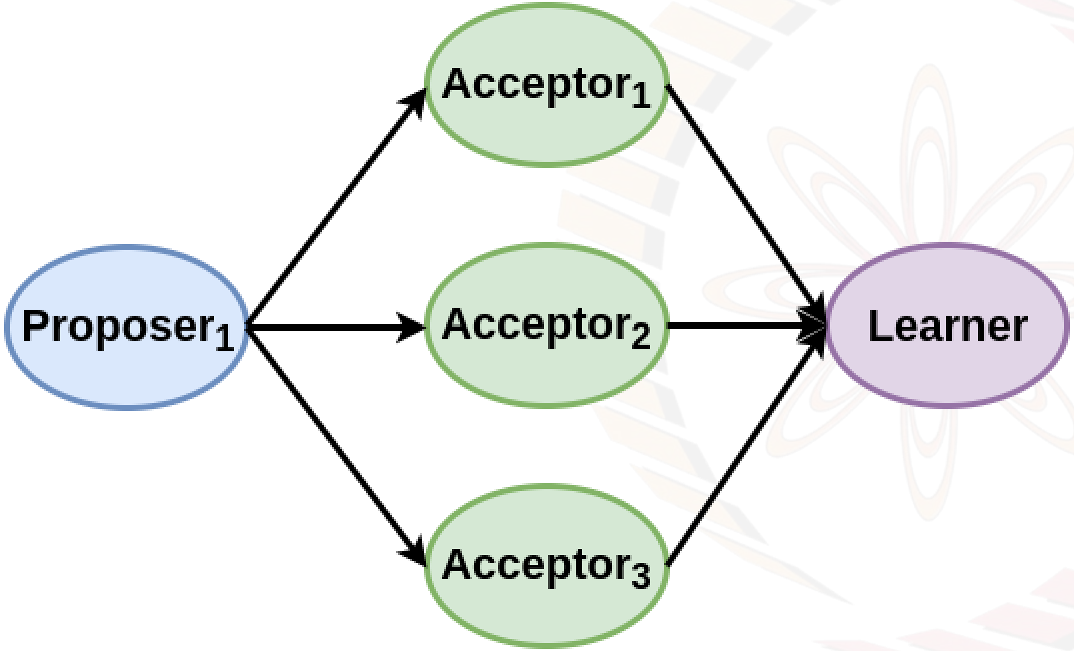
Accepting a Value: Notifying Learner



Acceptor_1
Acceptor_i
Acceptor_N
Majority Voting
$(\text{Acceptor}_1(\text{value}),\ldots,$
$\text{Acceptor}_i(\text{value}),\ldots,$
$\text{Acceptor}_N(\text{value}))$
Notify
value
Learner

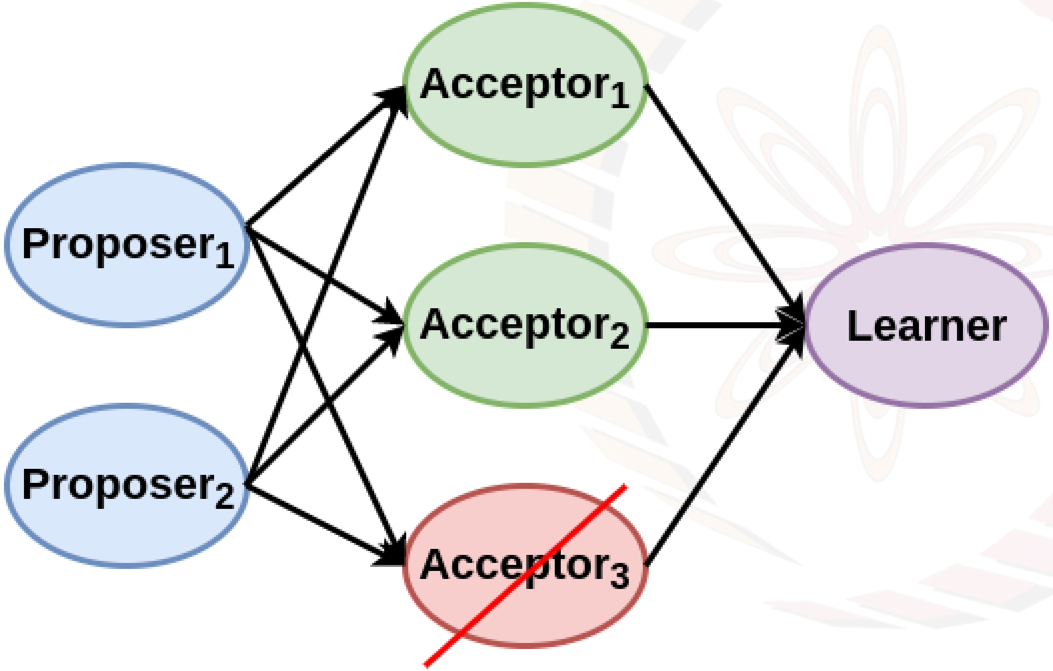
- Each acceptor accept value from any of the proposer
- Notify learner the majority voted value

Single Proposer: No Rejection



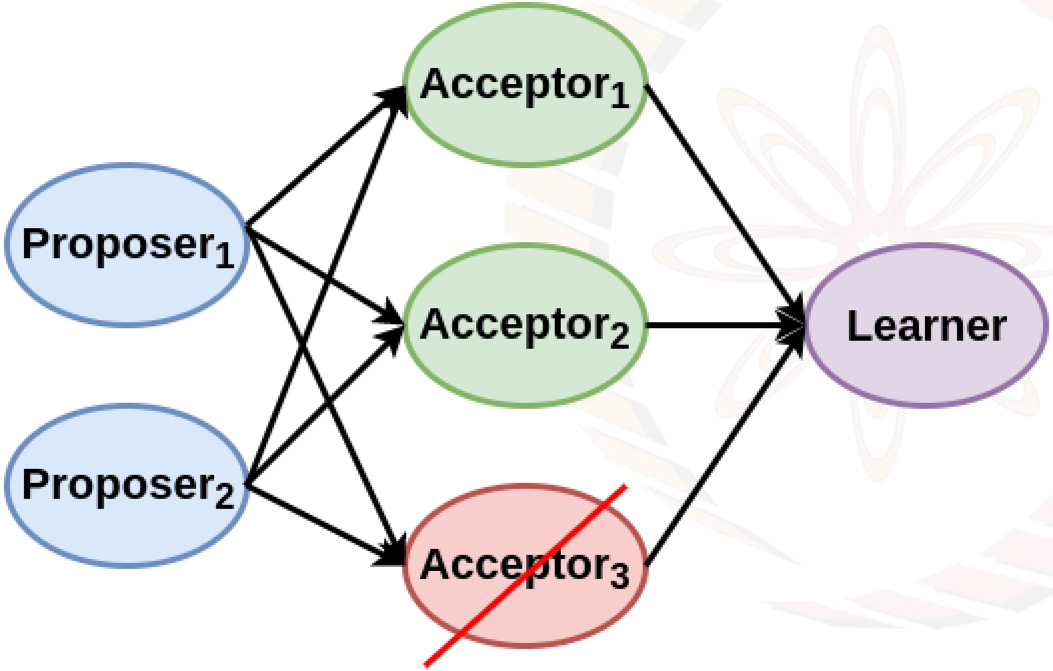
- Proposer always have proposal with biggest number
- No proposal rejected

Handling Failure: Acceptor Failure



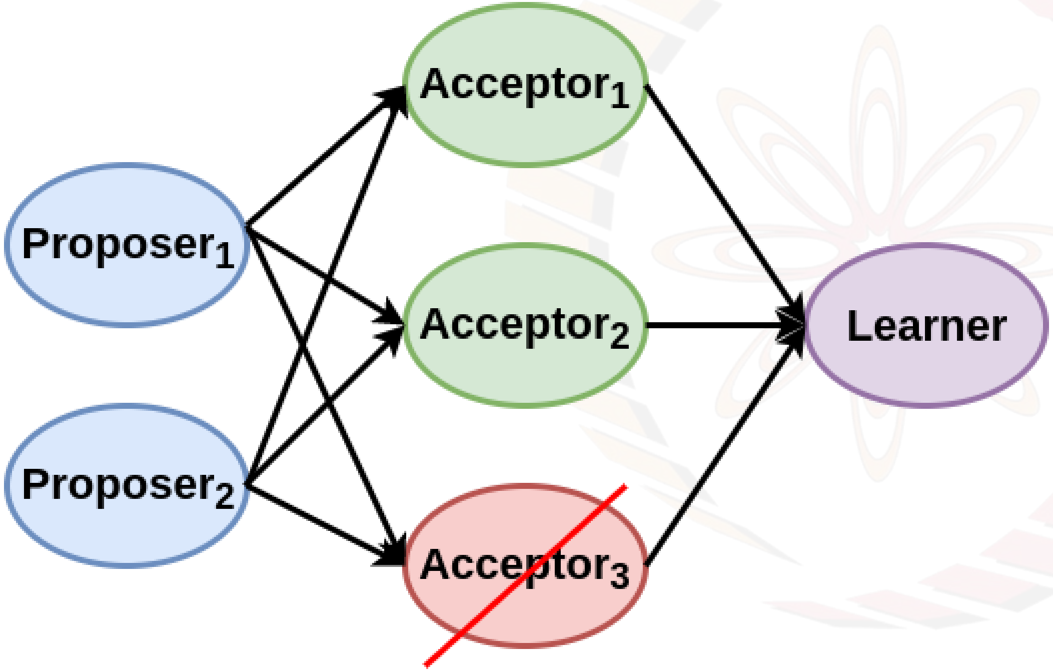
- Acceptor fails during prepare
    - No issues, other acceptor can hear the proposal and vote

Handling Failure: Acceptor Failure



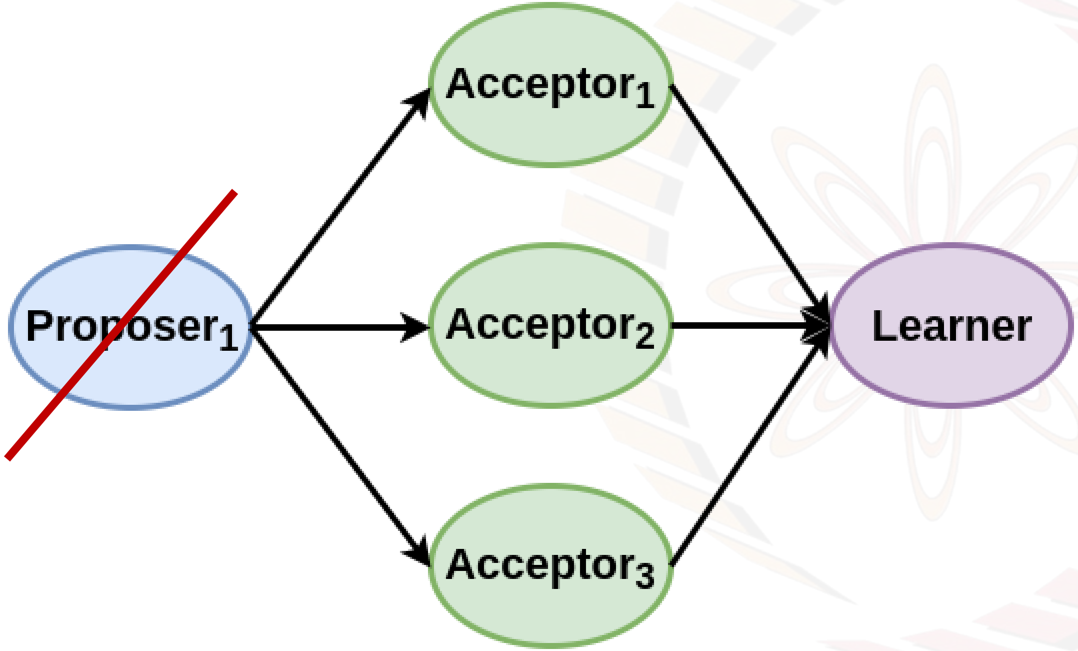
- Acceptor fails during accept
  - Again, no issues, other acceptor can vote for the proposal

Handling Failure: Acceptor Failure



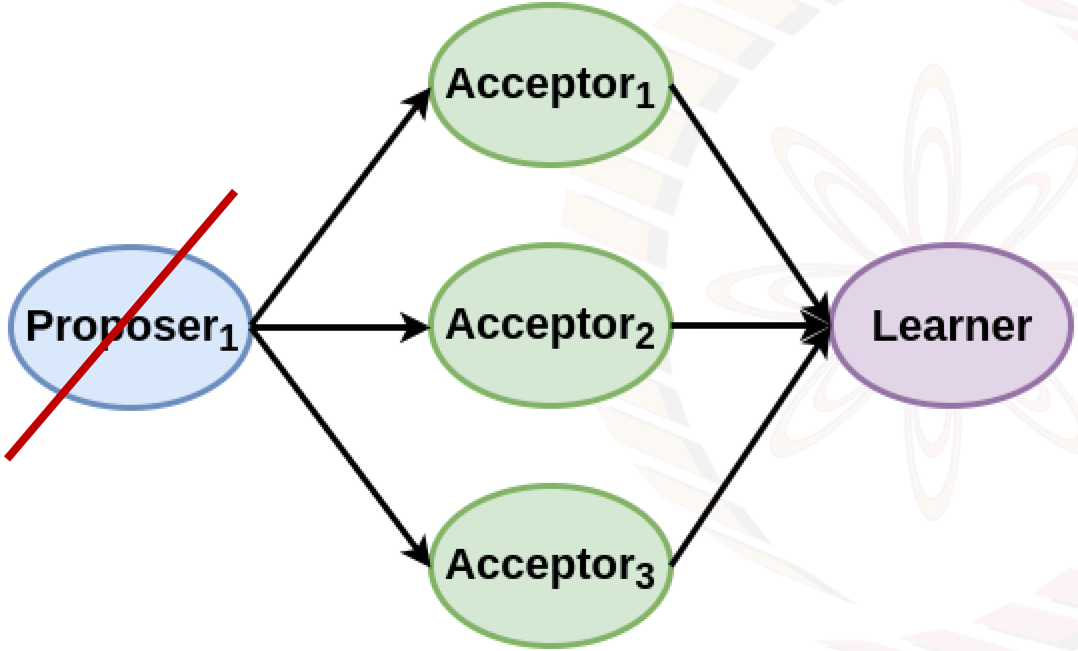
- More than N/2 - 1 acceptors fail
  - no proposer get a reply
  - no values can be accepted

Handling Failure: Proposer Failure



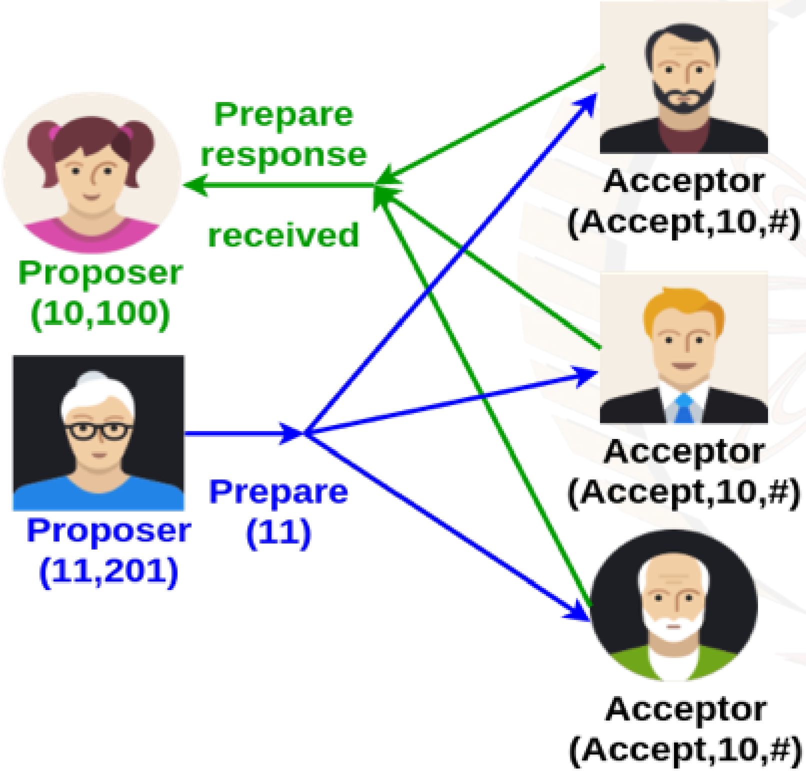
- Proposer fails during prepare phase
  - Acceptors wait, wait, wait, and then someone else become the proposer

Handling Failure: Proposer Failure



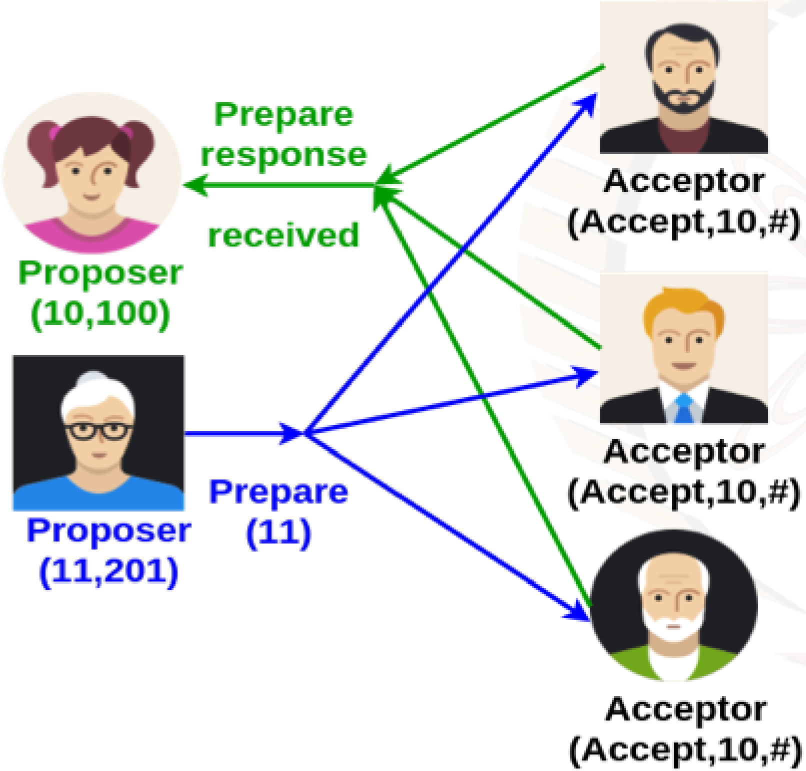
- Proposer fails during accept phase
  - Acceptors have already agreed upon whether to choose or not to choose the proposal

Handling Failure: Dueling Proposers



- Proposer received confirmations to her prepare message from majority
  - yet to send accept messages
- Another proposer sends prepare message with higher proposal number
- Block the first proposer’s proposal from being accepted

Handling Failure: Dueling Proposers



- Use **leader election** - select one of the proposer as leader
- Paxos can be used for leader election !!

thank you!