Paxos by Example

Umar Javed

Paxos

- Consensus Protocol in a distributed system
    - unreliable machines, network
    - multiple machines proposing different values
- Quorum-based
    - only a simple majority needs to agree
    - at least one overlapping node in successive proposals
        - e.g., f failures in a $2f + 1$ node system after agreement. Value is remembered during the next Paxos round
    - guaranteed progress if #failures <= f

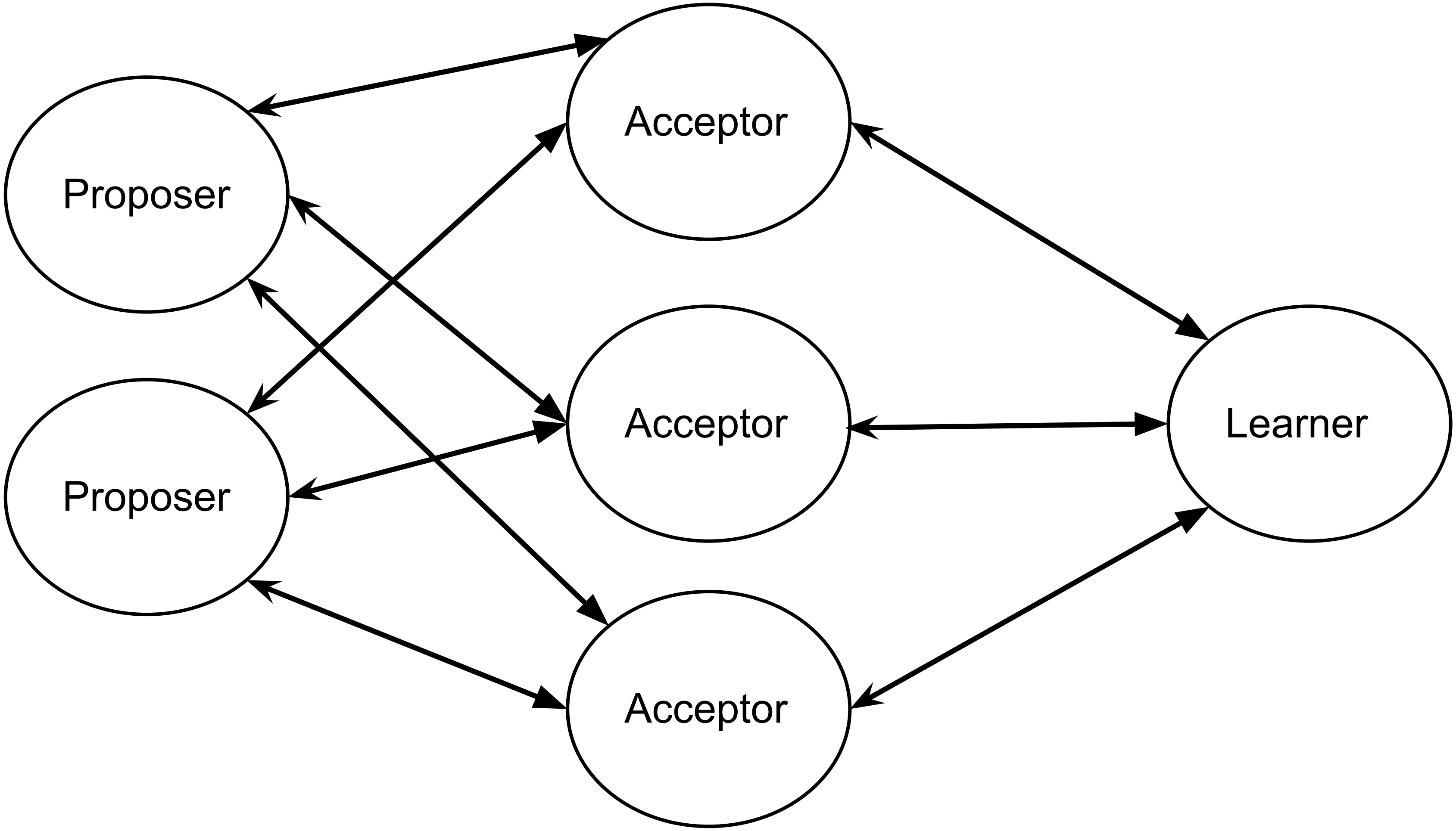
Paxos Applications

- leader election
- fault-tolerant replicated state machine
    - all replicas in same state given the same input sequence
    - distributed transactions

        (all replicas execute/log the same op)
    - distributed file server

        (agree on the same session id)

Paxos Components



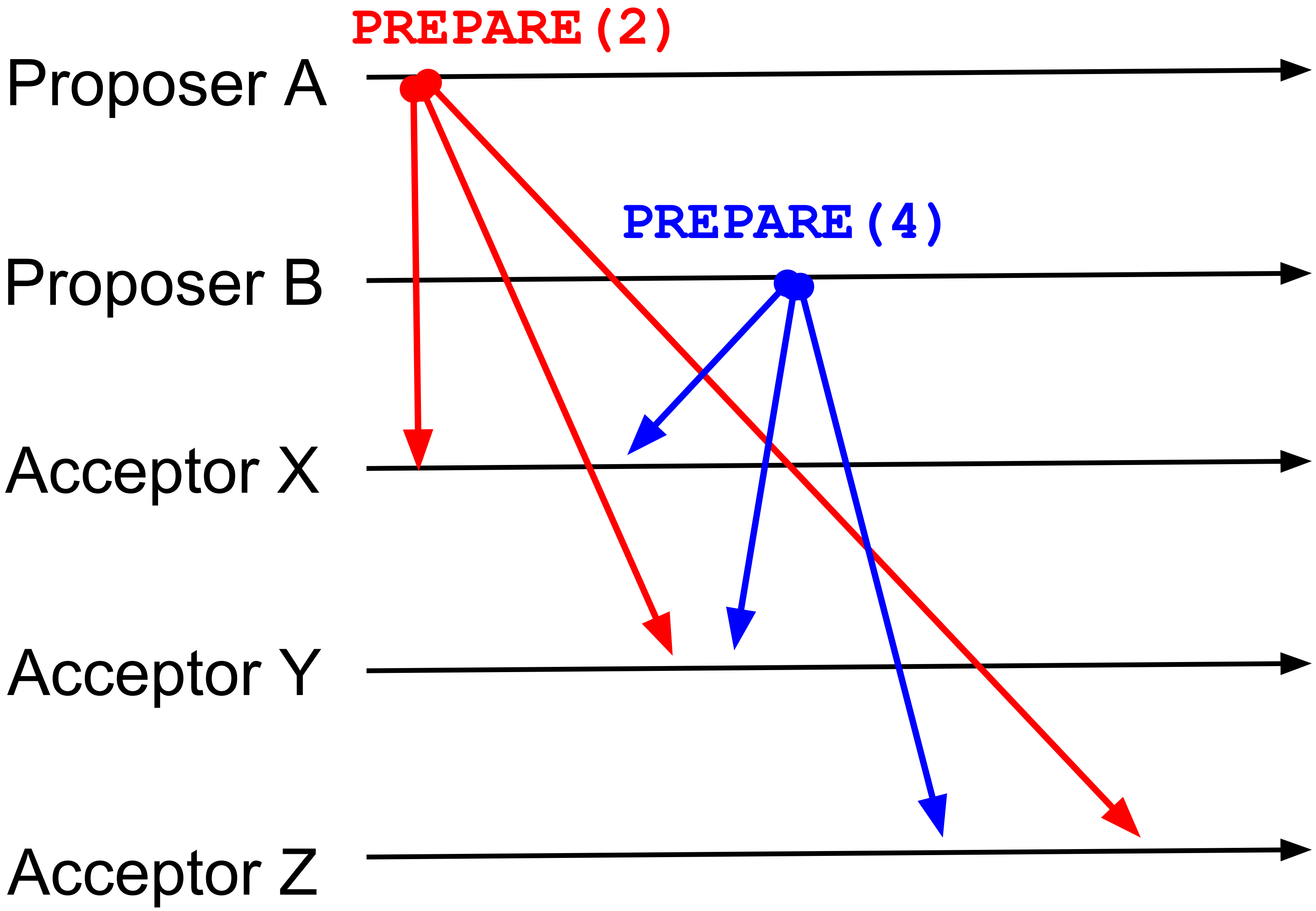
Paxos Basic Mechanisms

1) Assign ordering on proposals
   - goal: since an acceptor can accept multiple proposals, prevent invalid/stale proposals from interfering
   - lower numbered proposals rejected
   - allows laggers to catch up
2) Restrict the proposer's choice of value
   - goal: avoid conflicting proposed values
   - actual value doesn't matter as long as same
   - learn previously agreed value, if any

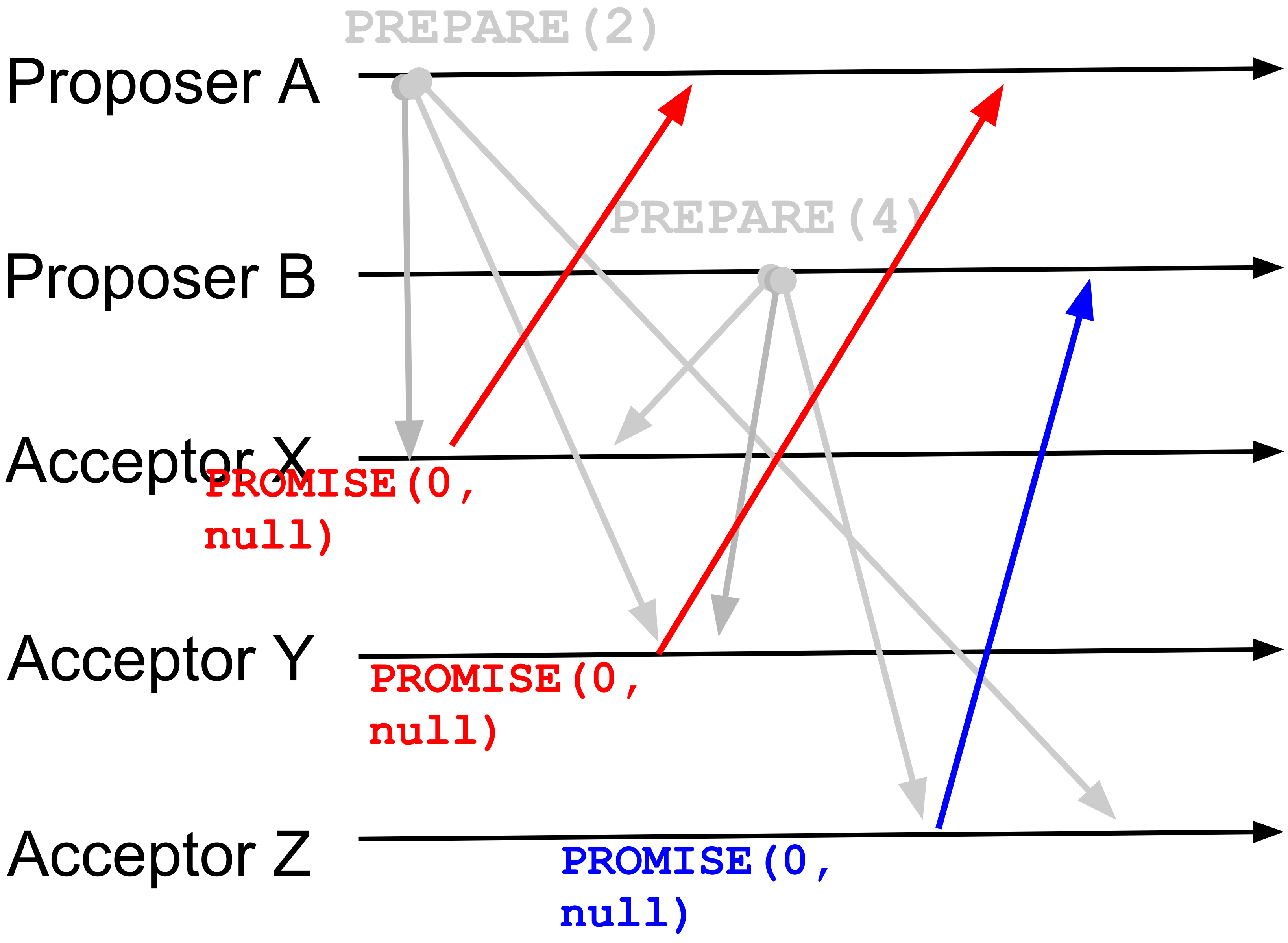
Paxos Phase 1: `PREPARE/PROMISE`

- Proposer wants consensus on a value
- Tell all acceptors that it wants consensus
  - make sure acceptors don't accept invalid values
  - `sqn n = max_sqn + 1`
  - broadcast `PREPARE(n)`
- Acceptors issue `PROMISE(n',v)` if `n > n'` where `n' = max_proposal_num`
  - reject/ignore otherwise
  - promises to reject future `m < n`
  - `v =` value of `n'`
  - `n' = 0, v= null` if no proposal seen
- Proposer succeeds if majority of replicas reply

Paxos Phase 1: PREPARE/PROMISE



Paxos Phase 1: `PREPARE/PROMISE`



Proposer A

Proposer B

Acceptor X

Acceptor Y

Acceptor Z

PREPARE(2)

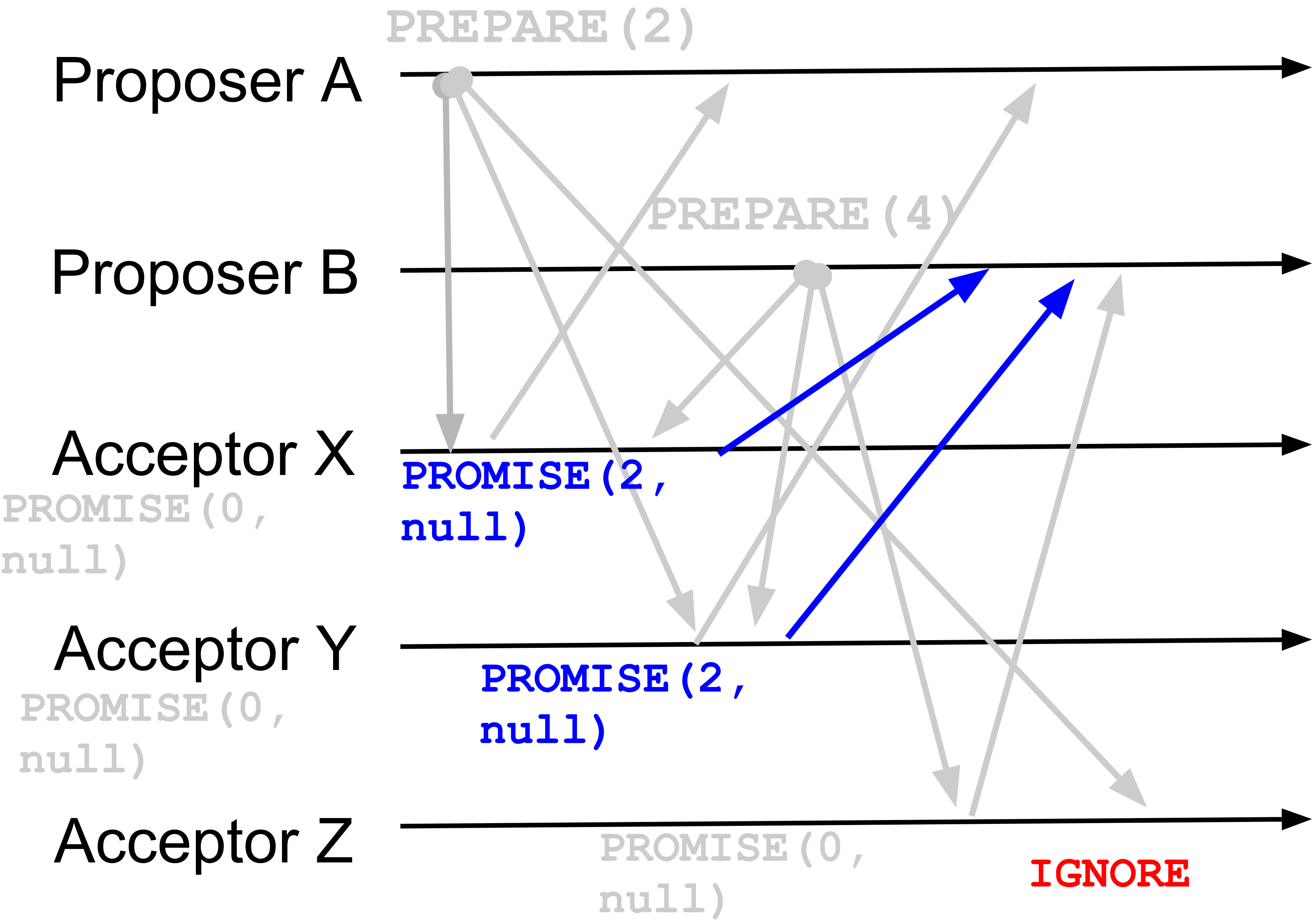
PREPARE(4)

PROMISE(0, null)

PROMISE(0, null)

PROMISE(0, null)

Paxos Phase 1: PREPARE/PROMISE



PREPARE(2)

Proposer A

PREPARE(4)

Proposer B

Acceptor X

PROMISE(0, null)

PROMISE(2, null)

Acceptor Y

PROMISE(0, null)

PROMISE(2, null)

Acceptor Z

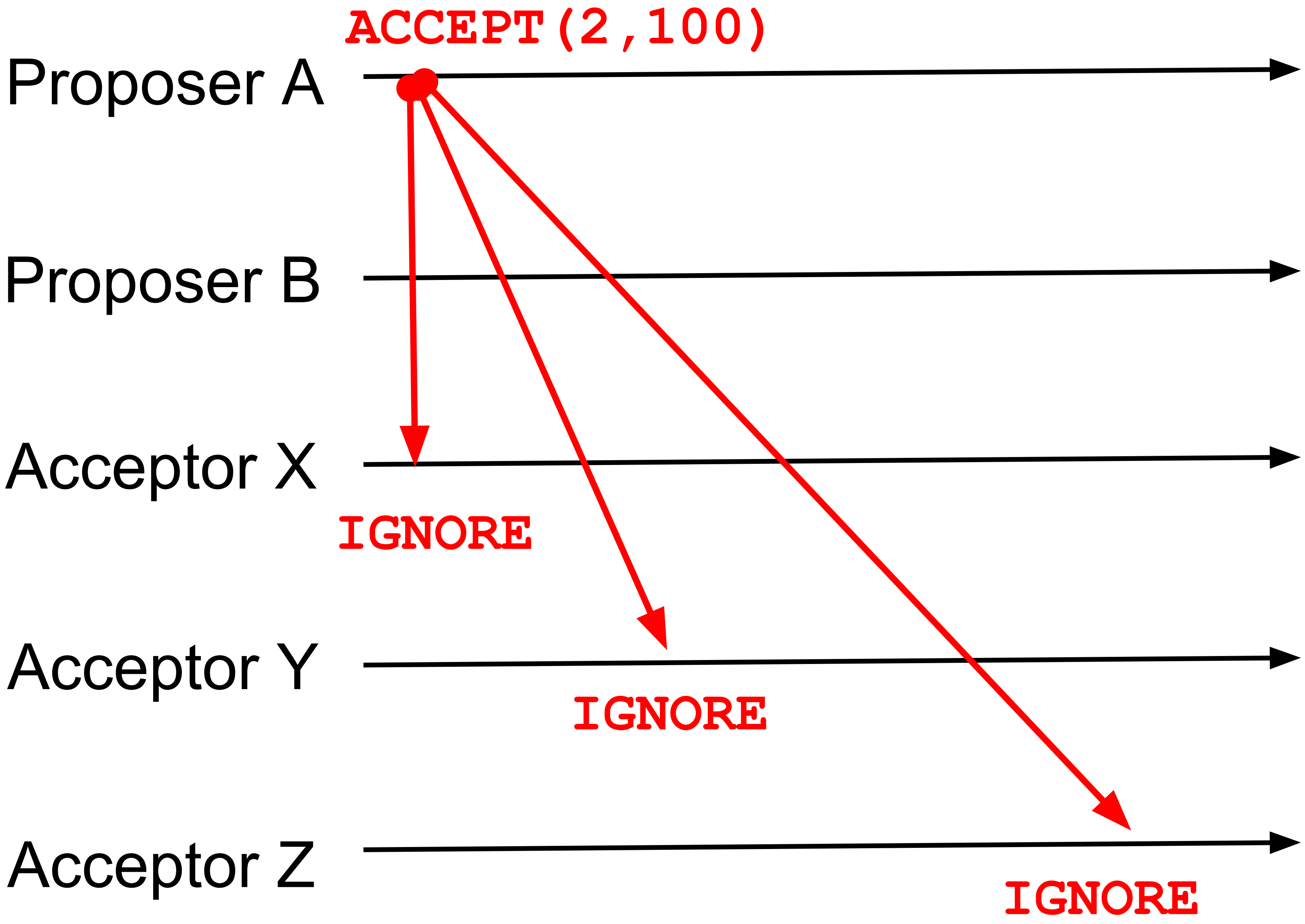
PROMISE(0, null)

IGNORE

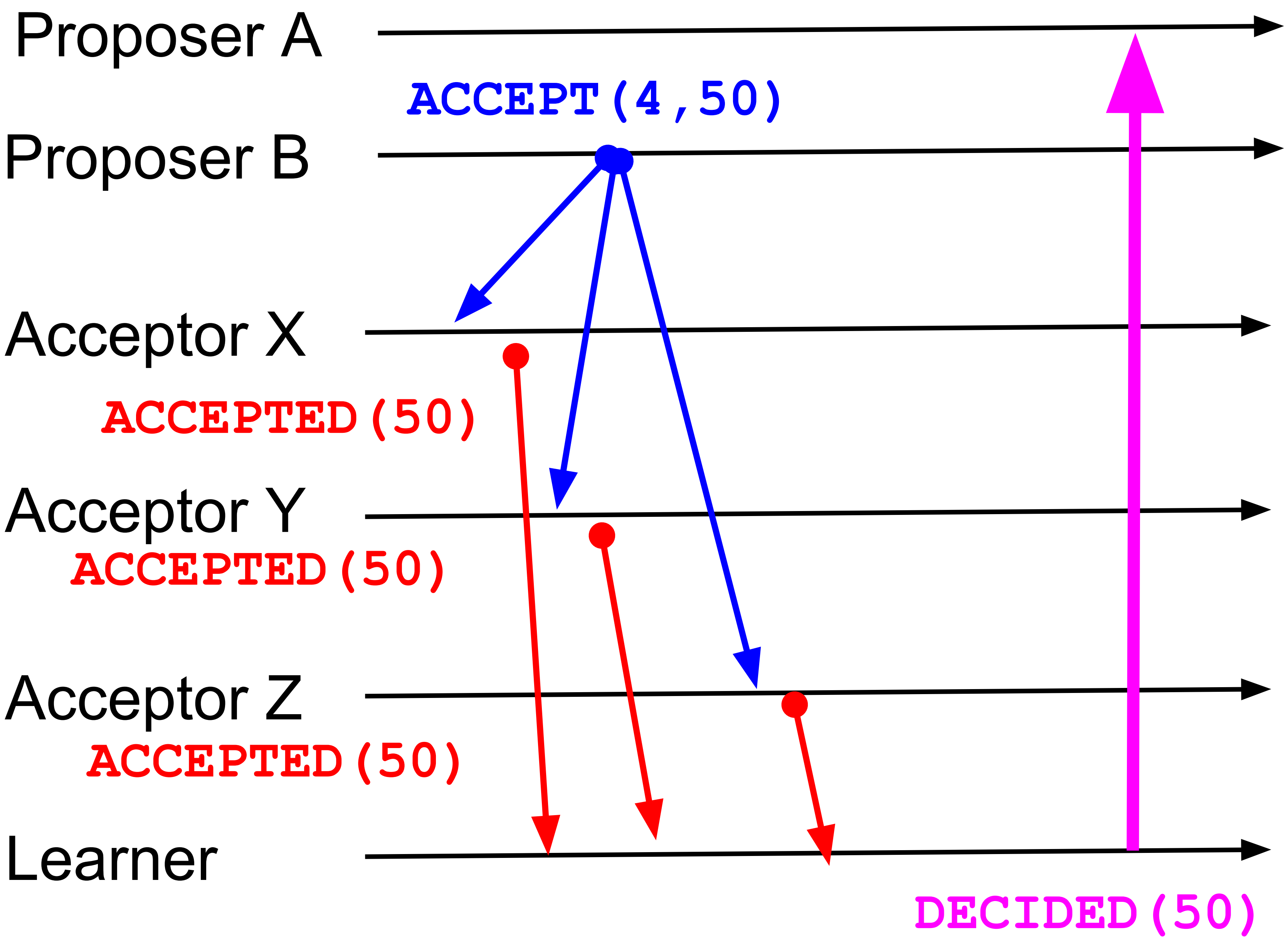
Paxos Phase 2: ACCEPT/ACCEPTED

- Proposer A thinks its the boss
  - received 2/3 PROMISE messages
- Proposer B also thinks its the boss!
  - received 3/3 PROMISE messages
- Both will generate their own value
  - not guaranteed that there will be a 'master' proposer in phase 2
  - Phase 2 filters any such contention
  - All acceptors have seen the max proposal number: 4 => ignore messages from proposer A

Paxos Phase 2: ACCEPT/ACCEPTED



Paxos Phase 2: ACCEPT/ACCEPTED



Supplementary Readings

- Paxos Made Live - An Engineering Perspective
  - Chubby distributed locking by Google
  - Implementation for a real large-scale fault-tolerant system
- Paxos Made Practical
  - Implementation of an RPC-based distributed filesystem
  - Election of a master server