Paxos

14-736 (Distributed Systems)

This lecture based heavily upon :

- https://www.quora.com/In-distributed-systems-what-is-a-simple-explanation-of-the-Paxos-algorithm
- Lamport, Leslie, "Paxos Made Simple", 01 Nov 2001.

Consensus

- A collection of process can propose values. A consensus algorithm ensures
  - That a single proposal is chosen
  - The processes can learn the proposed value
  - No value is chosen if there are no proposals.

Consensus Safety Requirements

- Only a value that has been proposed may be chosen
- Only a single value is chosen, and
- A process never learns that a value has been chosen unless it has been

Goal, Simply Put

- The goal of the Paxos algorithm is for some number of peers to reach agreement on a value.
- Paxos guarantees that if one peer believes some value has been agreed upon by a majority, the majority will ***never*** agree on a different value.

Liveness, By Intuition

- A proposed value is eventually chosen
- Once a value is chosen, the processes eventually learn it

Communication

- Agents operate at arbitrary speed
- Agents may fail by stopping and then be restarted
  - Unless some information can be remembered across restart, consensus isn't possible

Mechanism

- The protocol is designed so that any agreement *must* go through a majority of nodes.
- Any future attempts at agreement, if successful must also go through at least one of those nodes.
- Thus: **Any node that proposes after a decision has been reached must communicate with a node in the majority. The protocol guarantees that it will learn the previously agreed upon value from that majority.**

Three Phases

- Prepare
- Accept
- Decided

Prepare Phase:
Prepare and Promise

- First, we have the prepare phase. **A** sends a **prepare request** to A, B and C.
  - Paxos relies on sequence numbers to achieve its guarantees.
  - The prepare request asks a node to promise: "I will never accept any proposal with a sequence number less than that in the prepare request."
  - **The nodes reply with any value they have previously agreed to (if any).**
  - **Node A must propose the value it receives with the highest sequence number.** This action provides the guarantee the previously agreed upon values will be preserved.

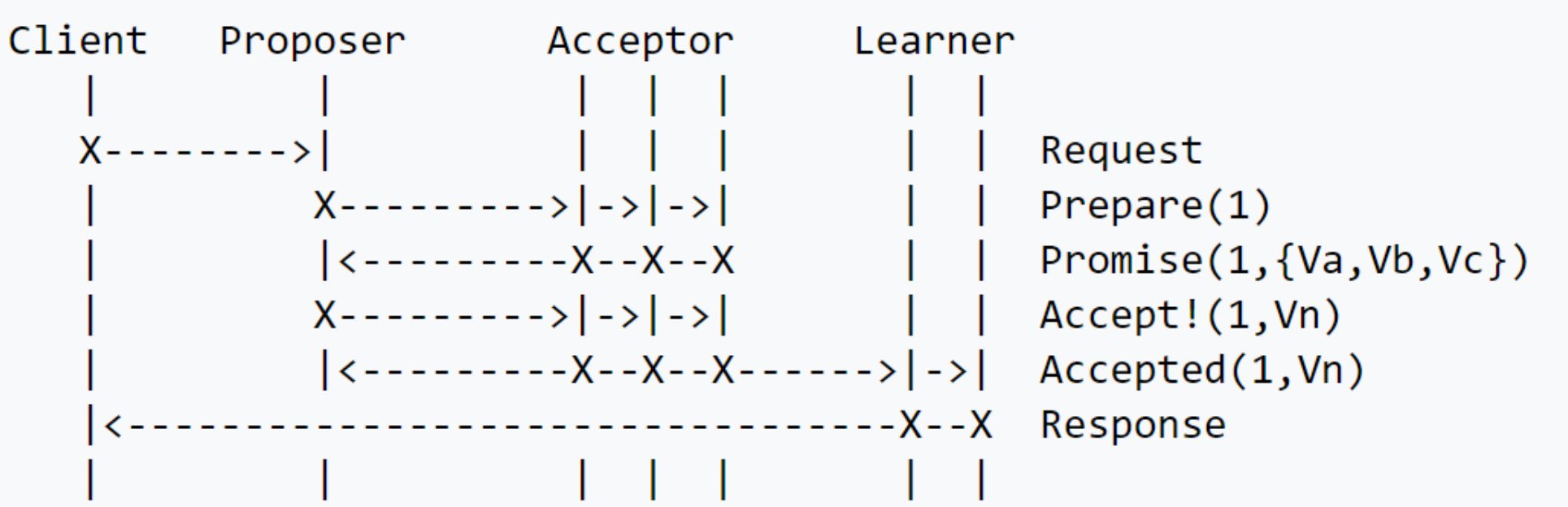
Accept Phase

- **A** sends an **accept request** to A, B and C.
  - The accept request states: **"Do you accept foo?"**
  - If the accompanying sequence number is not below the what the node had previously promised or request the node has previously accepted, it will accept the new value and sequence number.
- If node A receives accepts from a majority of nodes, **the value is decided.** This round of Paxos will **never** agree to another value

Decided/Accepted Phase

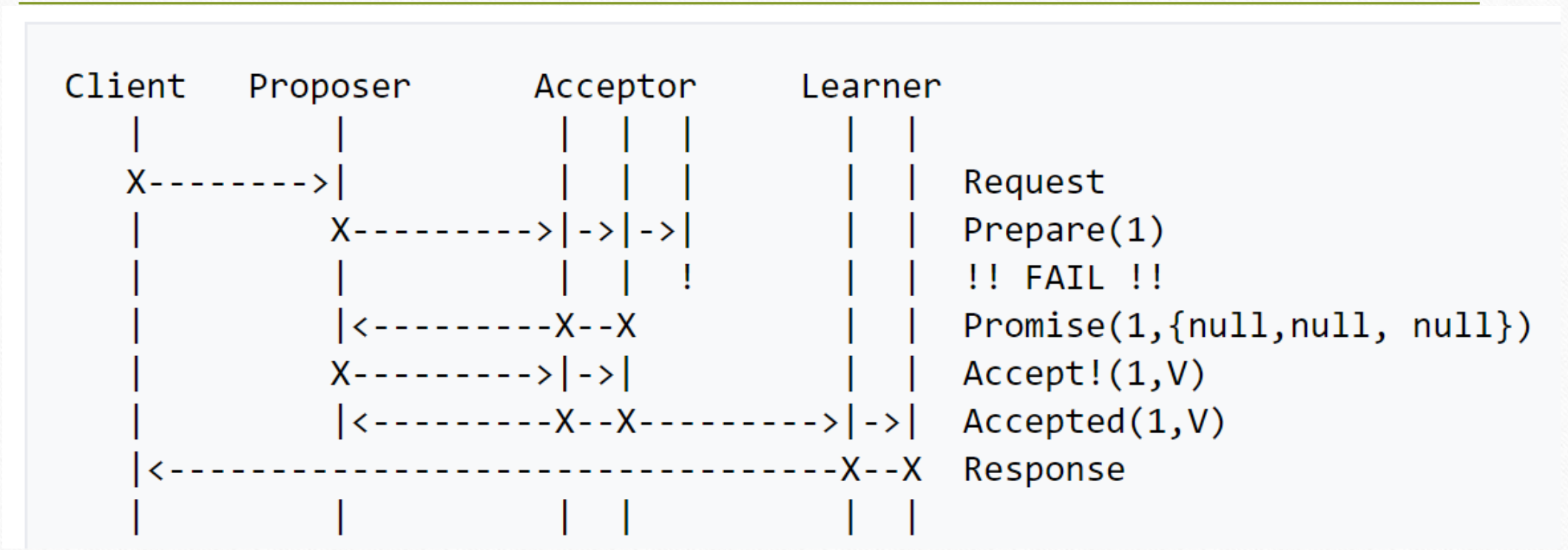
- The third phase is not strictly necessary, but is a crucial optimization in any productionized Paxos implementation.
- After **A** receives a majority of accepts, it sends **decided** messages to **A, B and C.**
- These messages let all the peers know that a value has been chosen, and accelerate the end of the decision process.
- Without this message, the other peers would have to attempt to propose a value to learn of the agreement.
  - In the prepare phase, they'd learn of the previously agreed upon value. Once that agreement was driven to conclusion, the node would recognize the agreement.

Paxos Message Flow



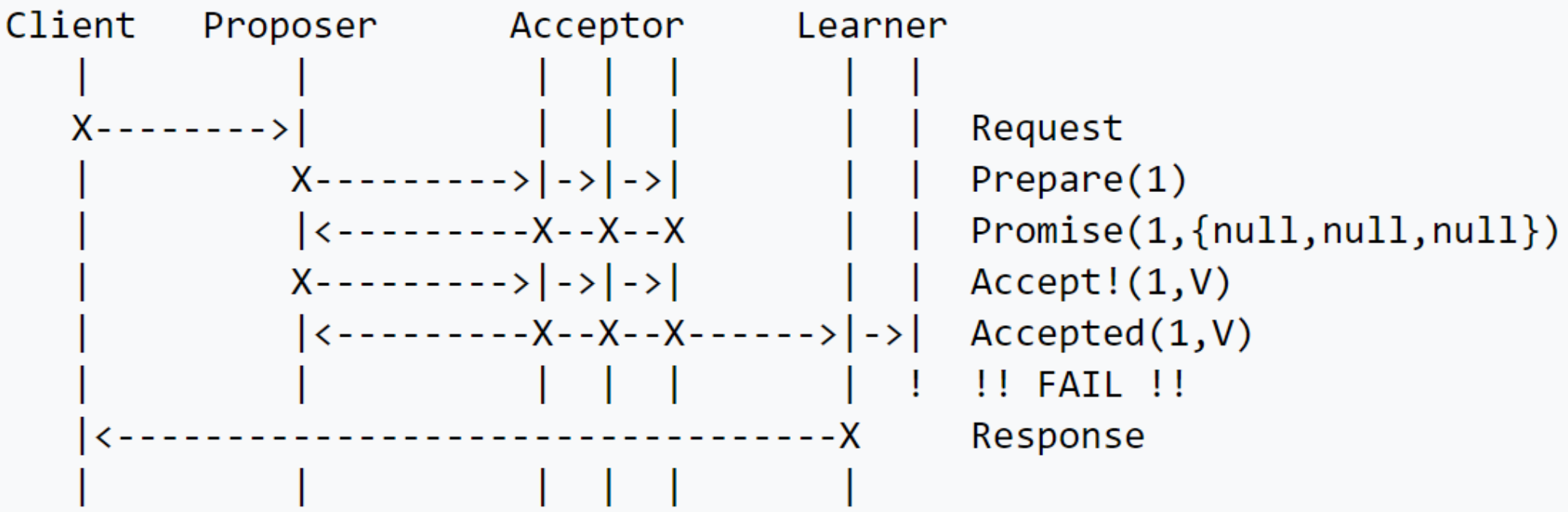
https://en.wikipedia.org/wiki/Paxos_(computer_science)

Message Flow: Failure of Acceptor



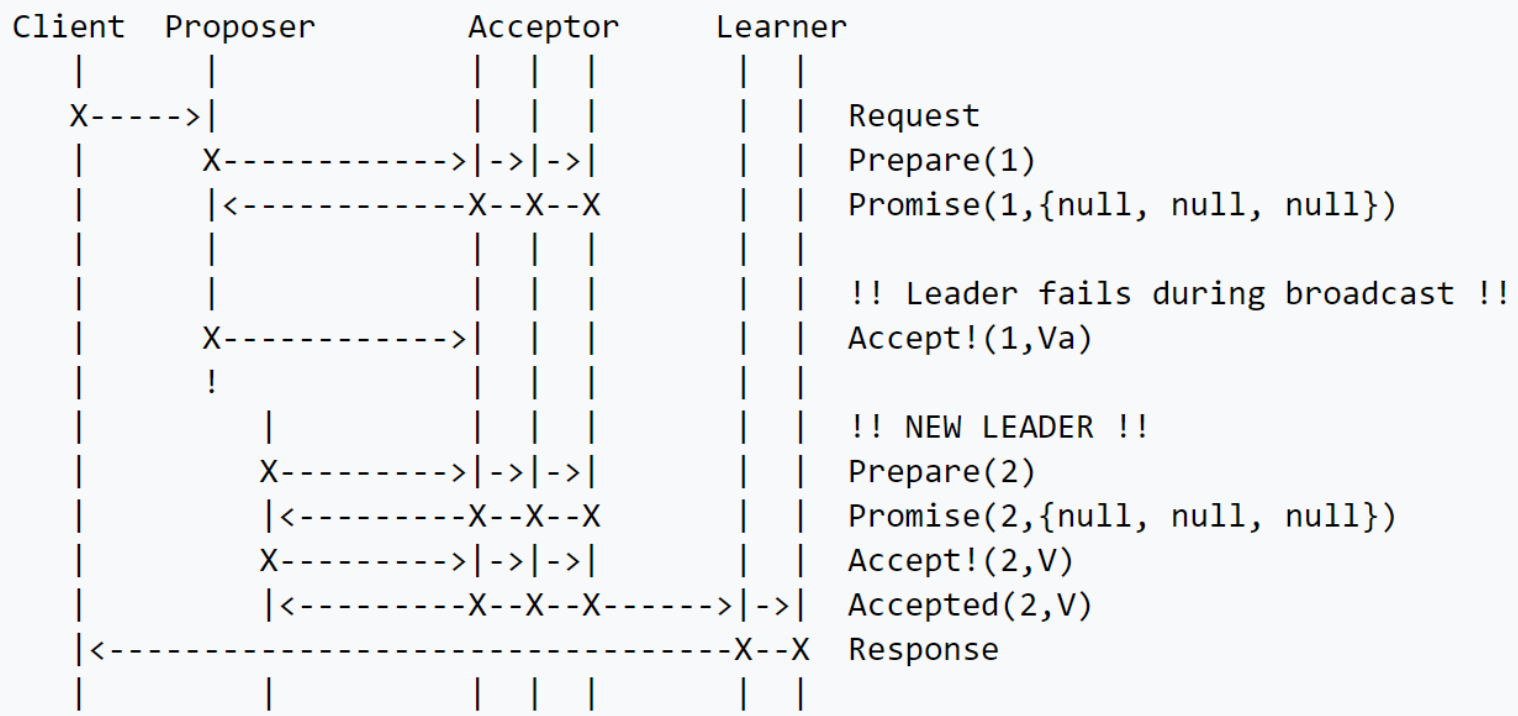
https://en.wikipedia.org/wiki/Paxos_(computer_science)

Message Flow:
Failure of Redundant Learner



https://en.wikipedia.org/wiki/Paxos_(computer_science)

Message Flow: Failure of Proposer



https://en.wikipedia.org/wiki/Paxos_(computer_science)

Multi-Paxos

- If leader is stable, no need for Prepare phase
- Include round number included in proposal.
  - Incremented with each proposal from same leader.

https://en.wikipedia.org/wiki/Paxos_(computer_science)