In Search of an Understandable Consensus Algorithm

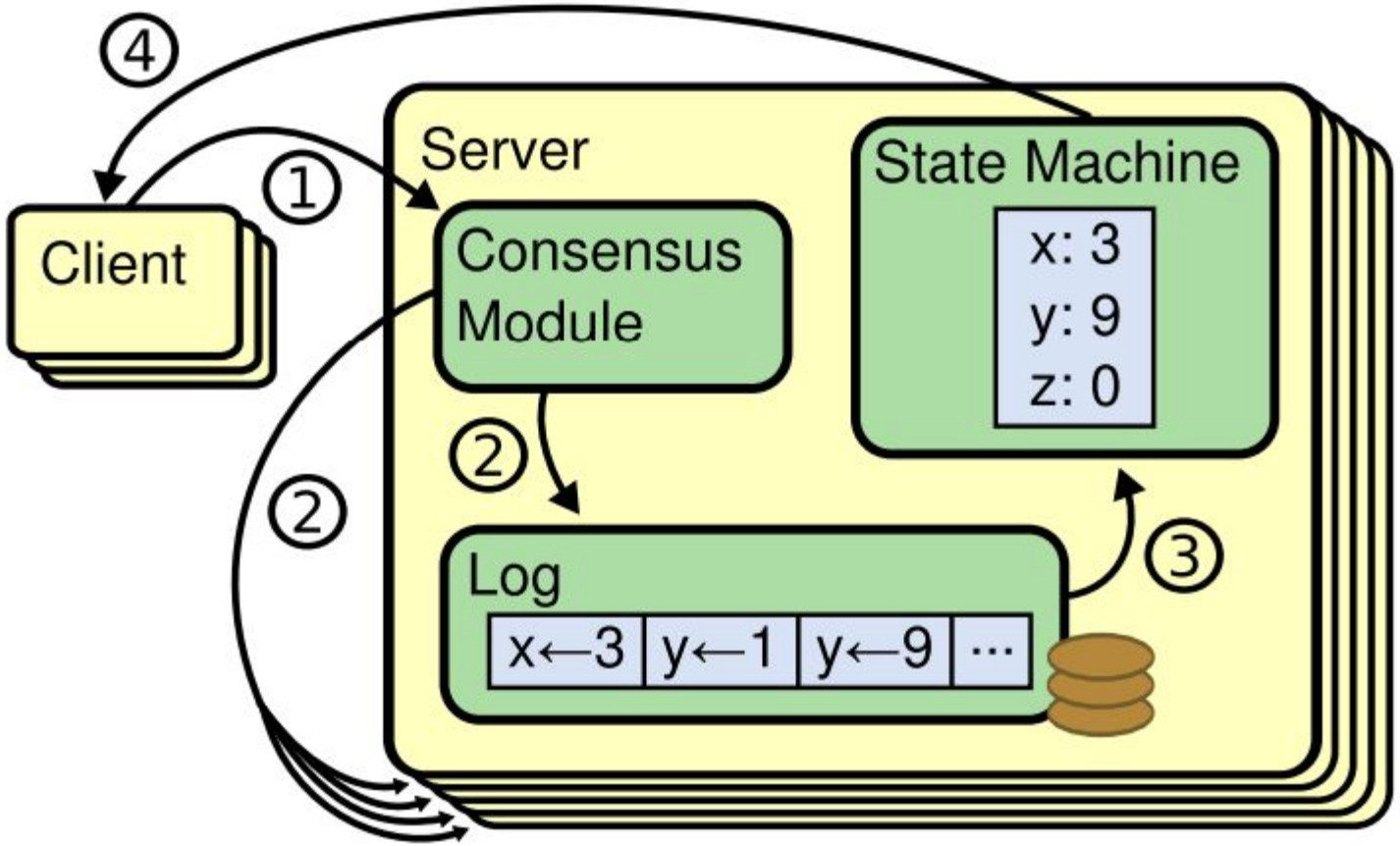
Wei Chen, Yuxuan Zhu

Universiteit Leiden
The Netherlands

Index

Introduction — Replicated state machine

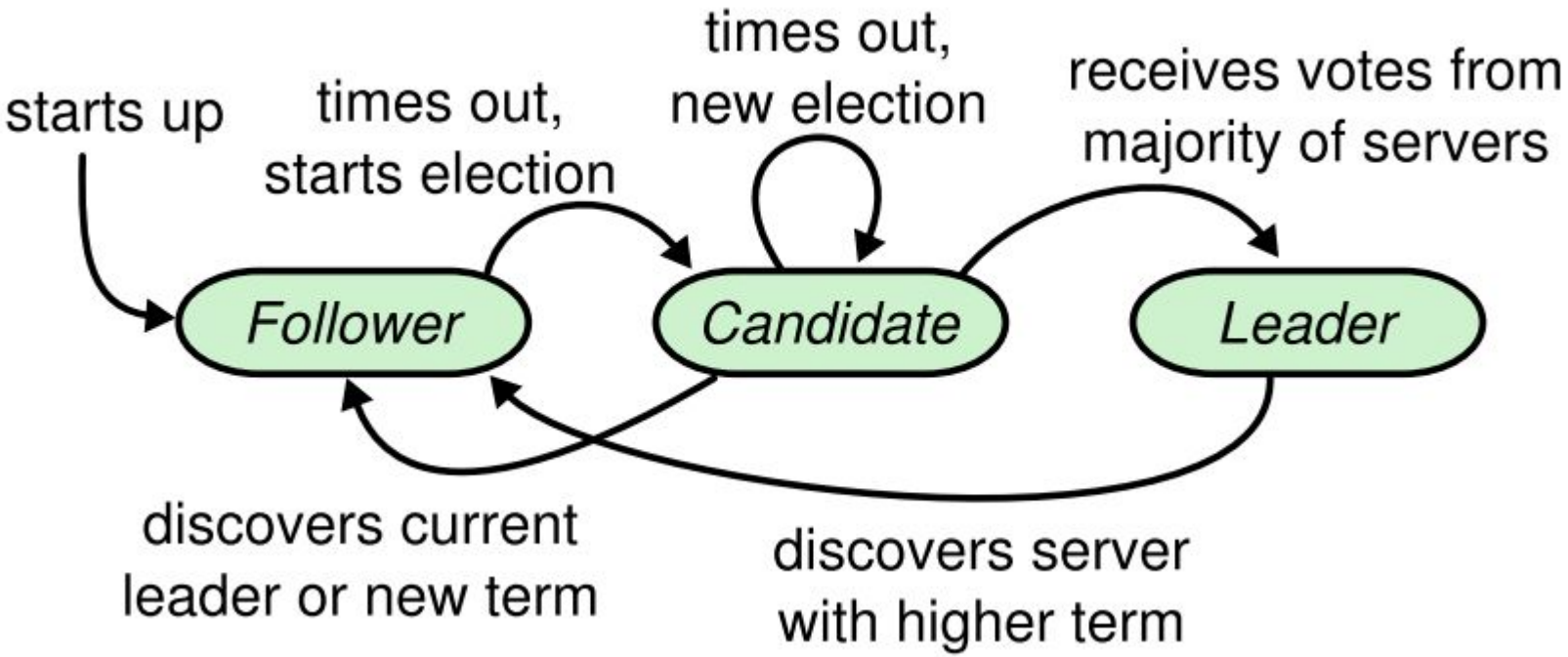
- **Client**
- **Server**
  - **Consensus module**
  - **Log**
  - **State Machine**



Introduction — Why not Paxos?

- **Unintelligible**
- **Hard to build practical implementations.**

Introduction — Three states of Raft

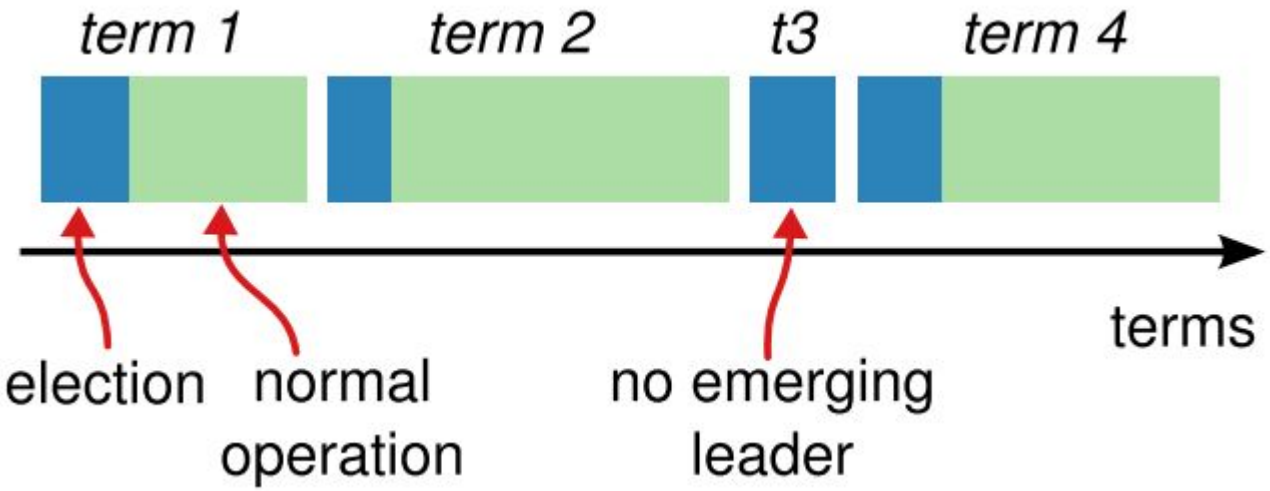


Introduction — Communication

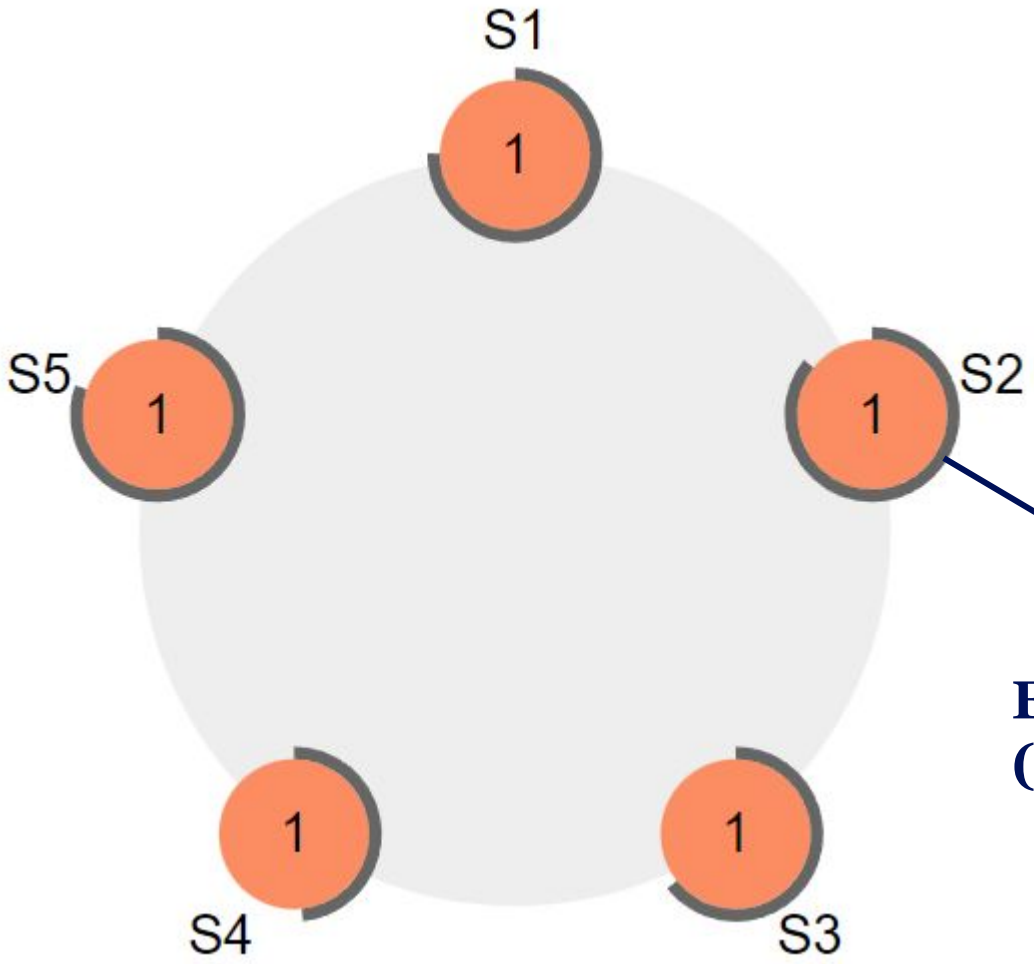
- **AppendEntries RPC**
  - heartbeat
  - add log entry
- **RequestVote RPC**
- **InstallSnapshot RPC**

Remote procedure calls (RPCs)

Introduction — Term



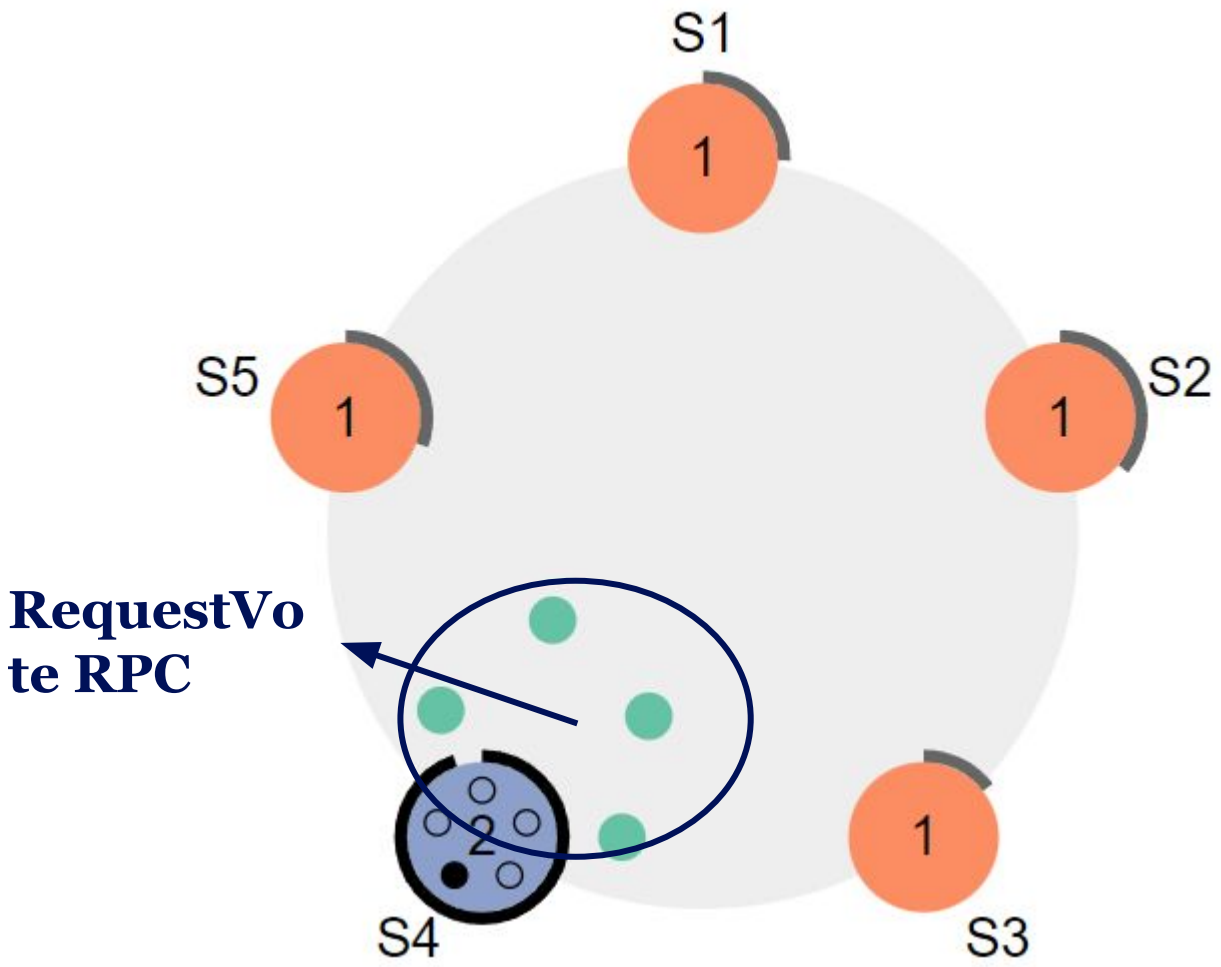
Leader election — Initial state

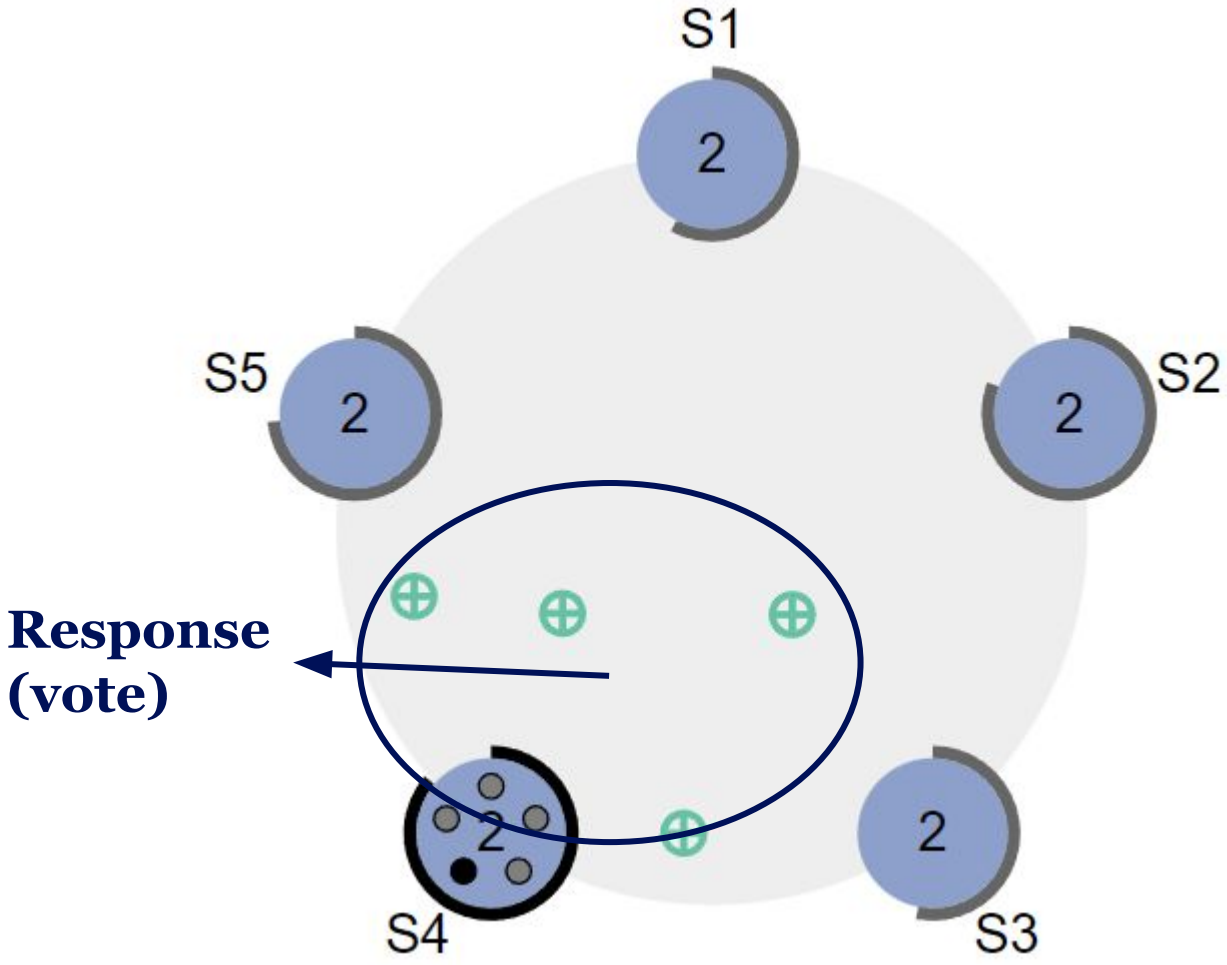


Election timeout (150 ms to 300 ms)

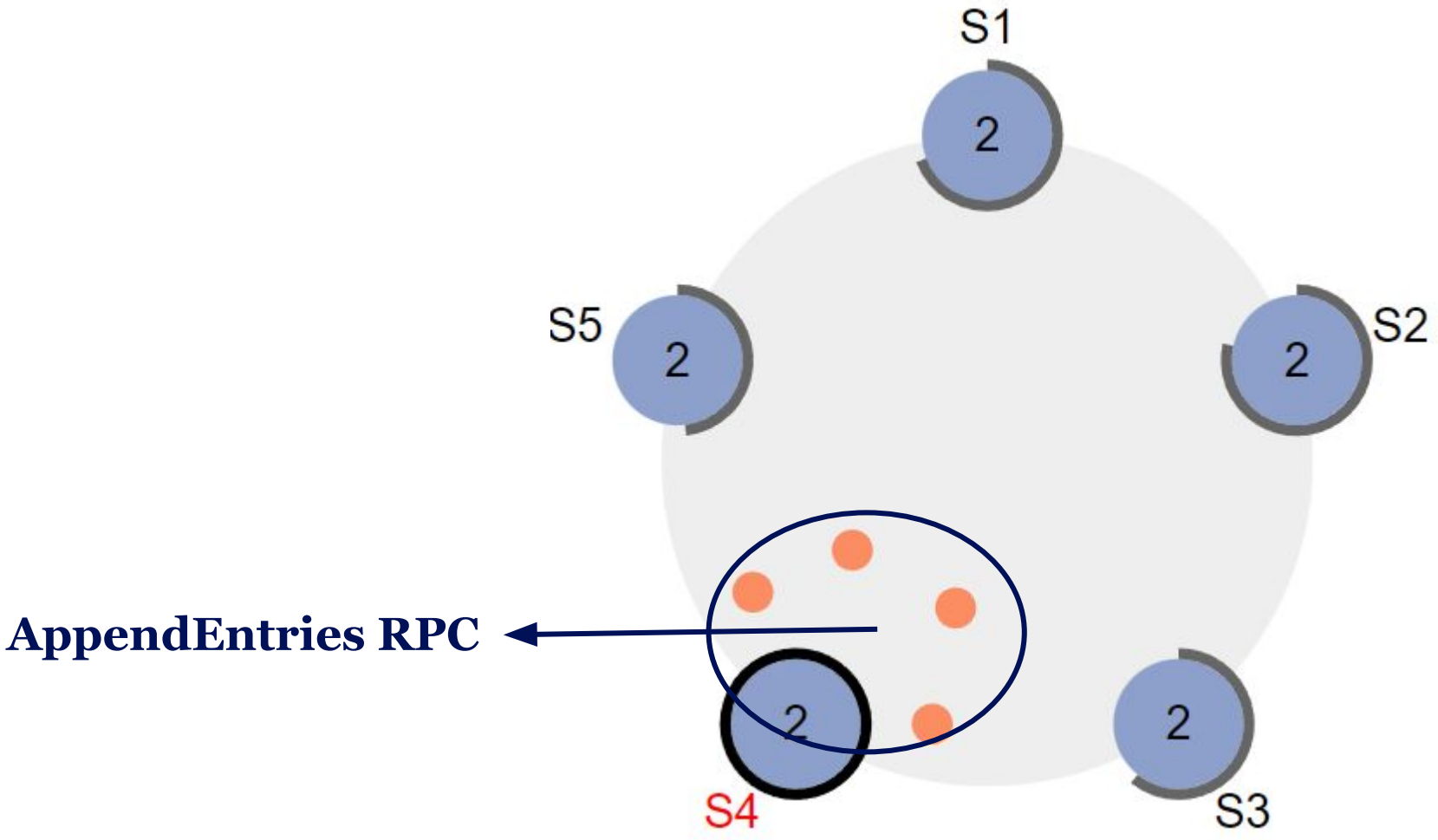
Server 4 is the first to become a candidate

Leader election — Voting Process

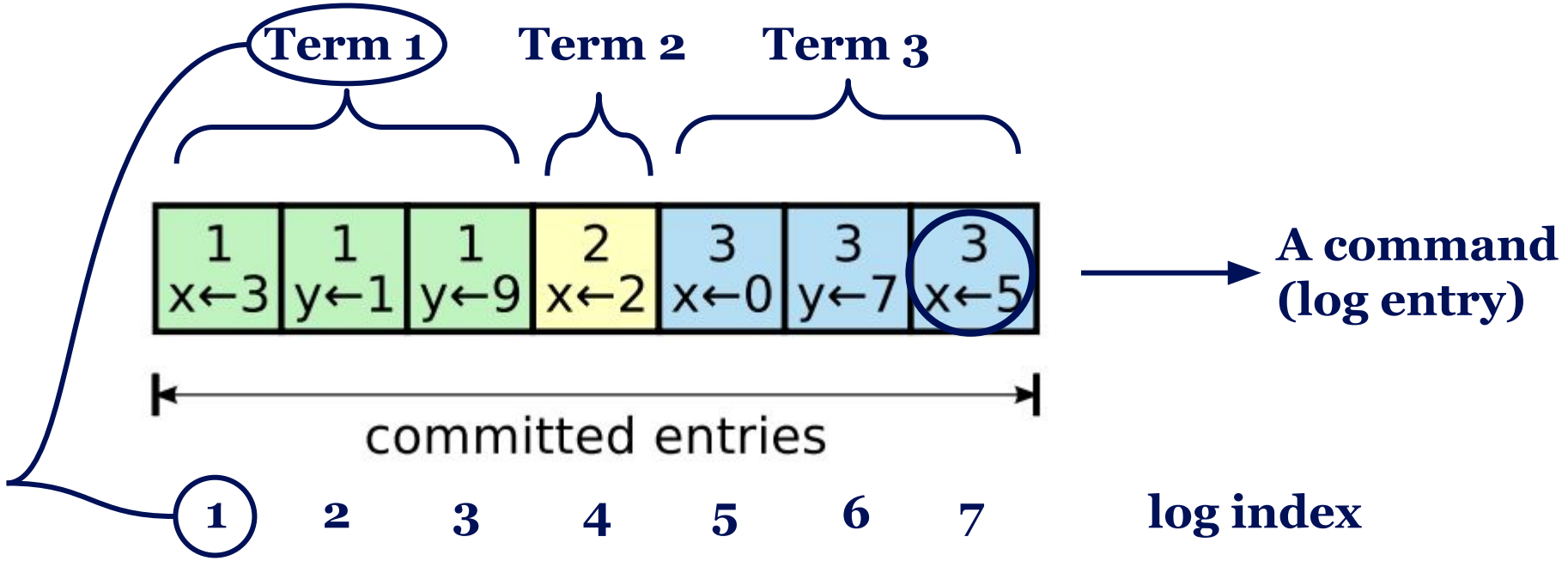




Leader election — Normal operations



Log replication — Log



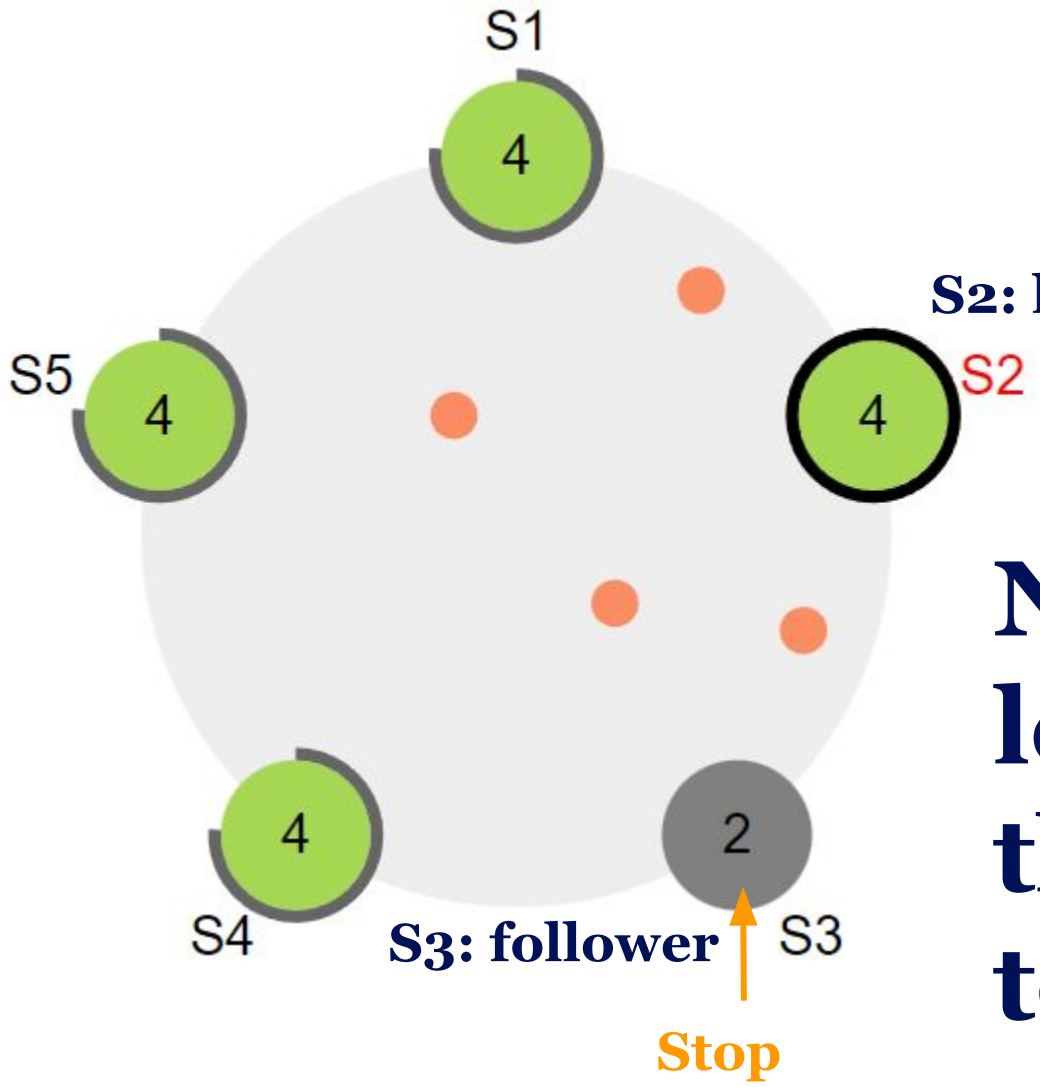
Stored!

Log replication — Replication

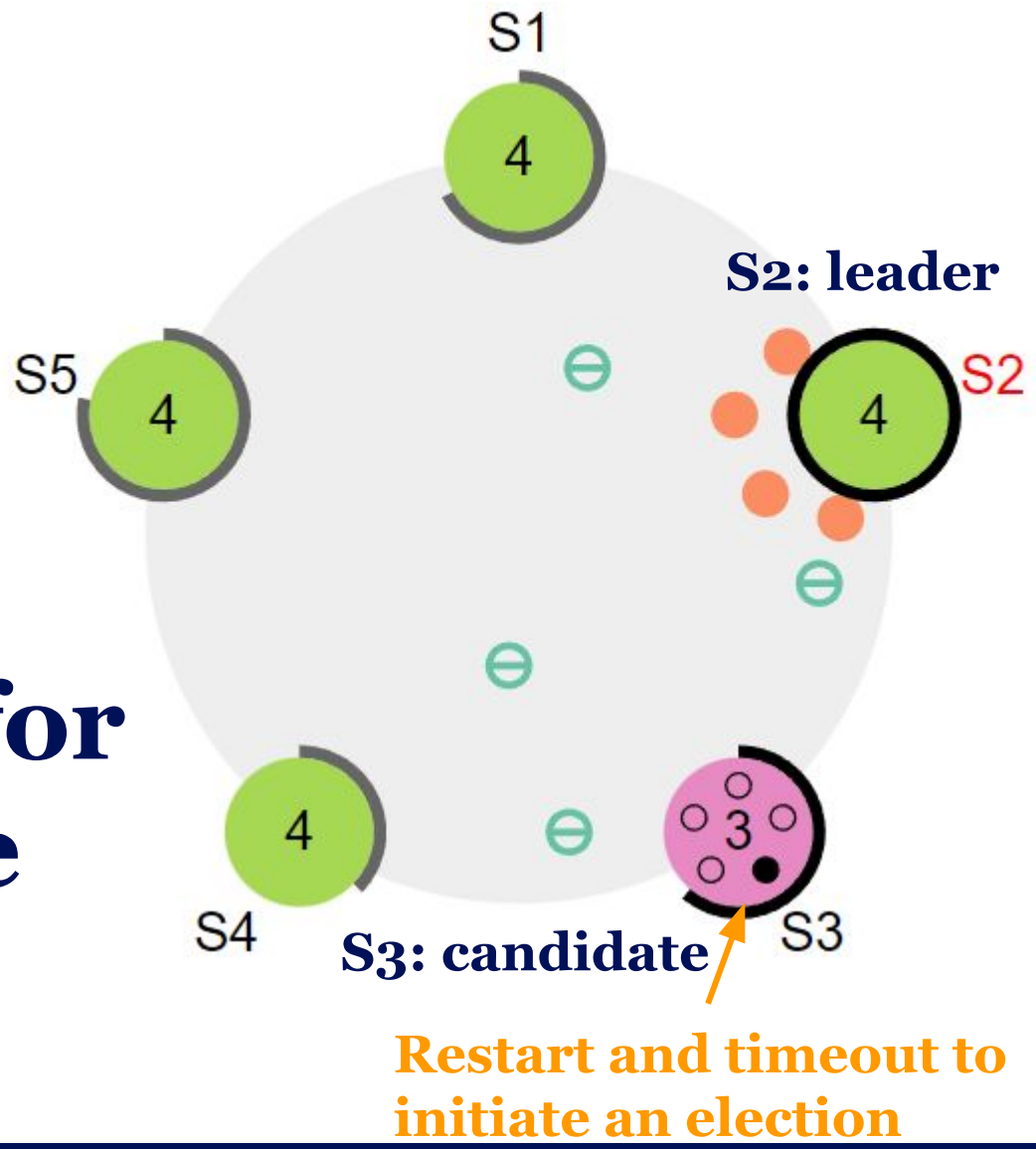
Process of replication

1. The leader (or follower) receives a request from a client
2. The leader adds it to its log as a new entry
3. The leader replicate it to its followers by AppendEntries RPCs
4. The leader commits this entry (no way to be changed) and bring it to its state machine to be processed if the leader receives a majority of acknowledges
5. The leader return the result processed by state machine to the client
6. Each follower applies this entry to its state machine when finding it has not yet to be processed.

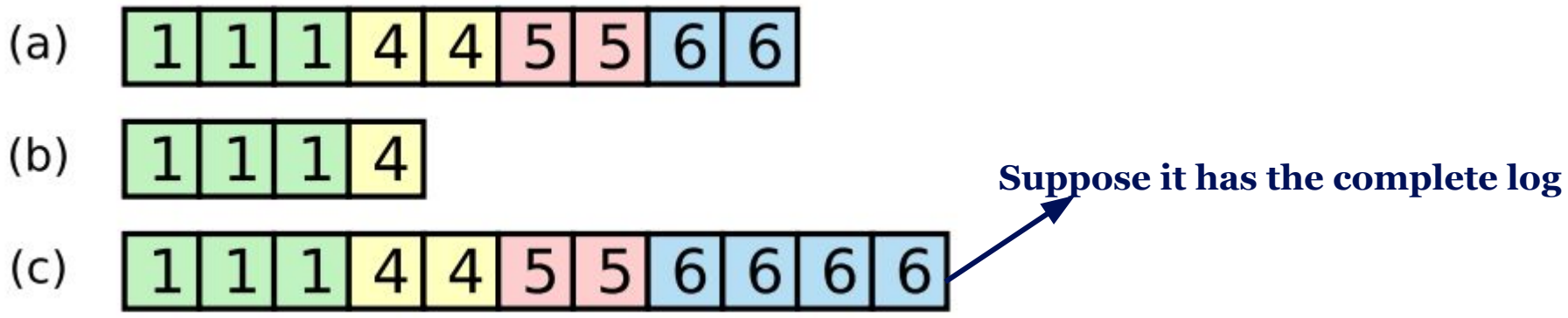
Safety — Leader election (Multi-Leaders)



No two leaders for the same term !!!

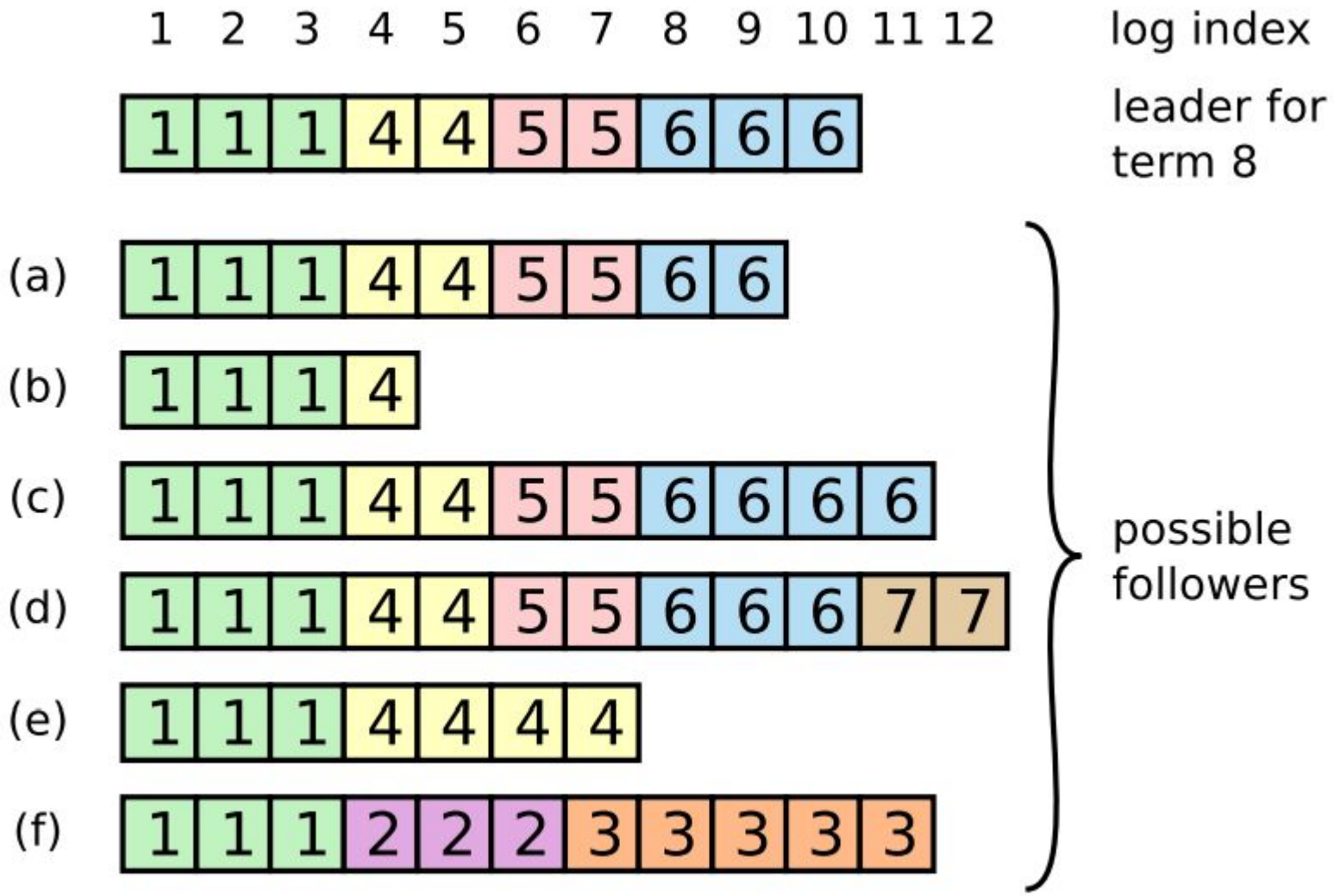


Safety — Leader election (Incomplete logs)

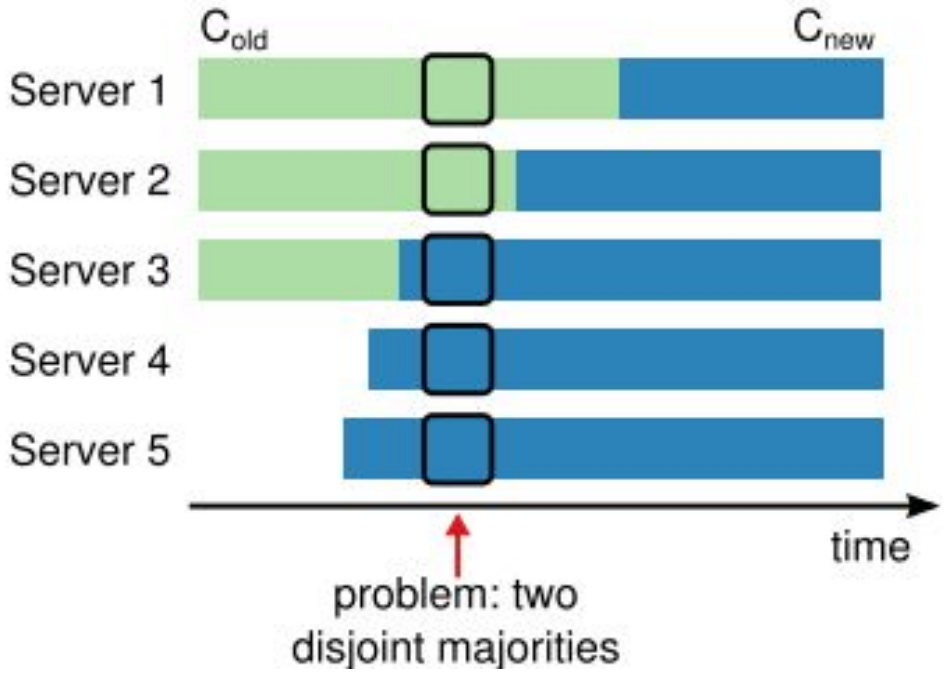


(c) will definitely be elected as a leader

Safety — Log replication



Cluster membership changes

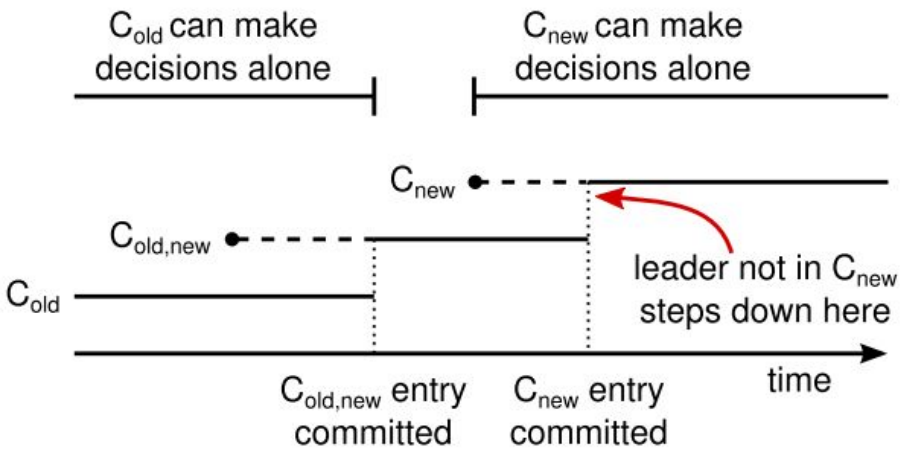


- **One Learder from Server 1/2**
- **One Learder from Server 3/4/5**

Split Brain

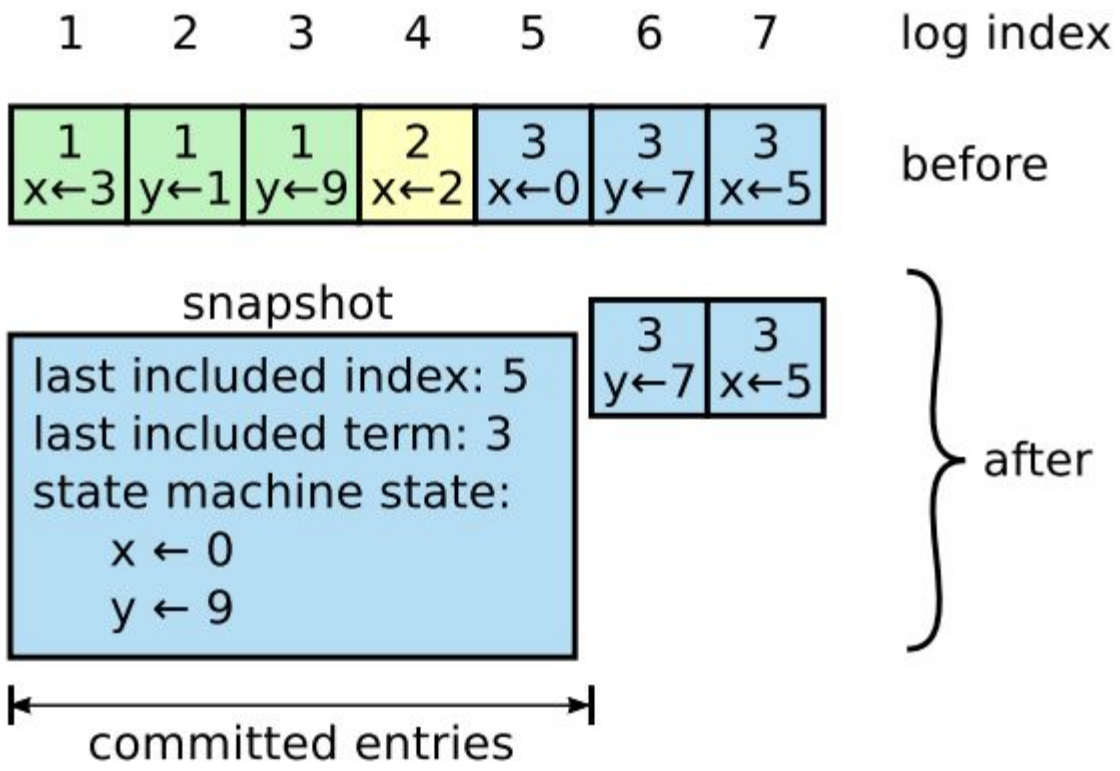
Cluster membership changes

In order to ensure safety, configuration changes must use a two-phase approach.



Joint Consensus: a transitional configuration

Log compaction



Client interaction

How clients interact with Raft?

Clients of Raft send all of their requests to the leader.

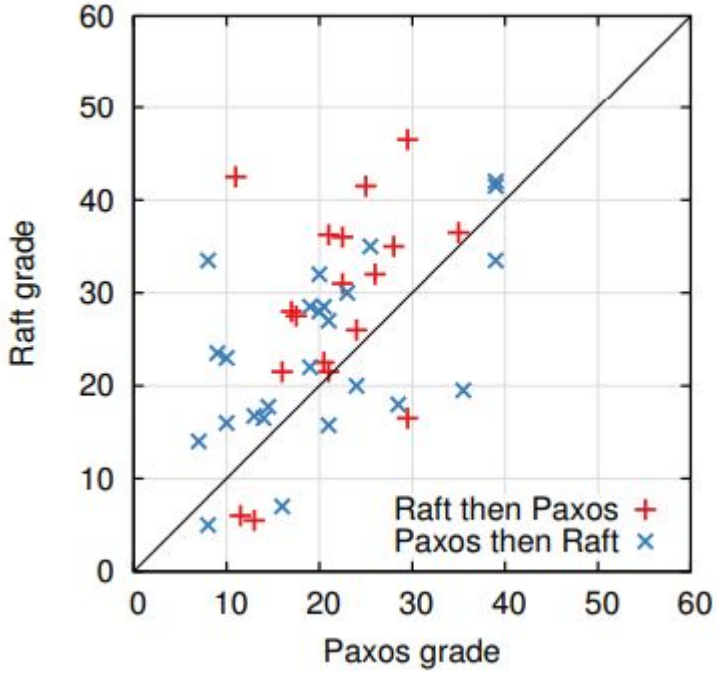
- Connects to a randomly chosen server when start up.
  - Connect to leader
  - Connect to follower, and follower supply the information of leader
  - Leader crashes: requests time out, try again

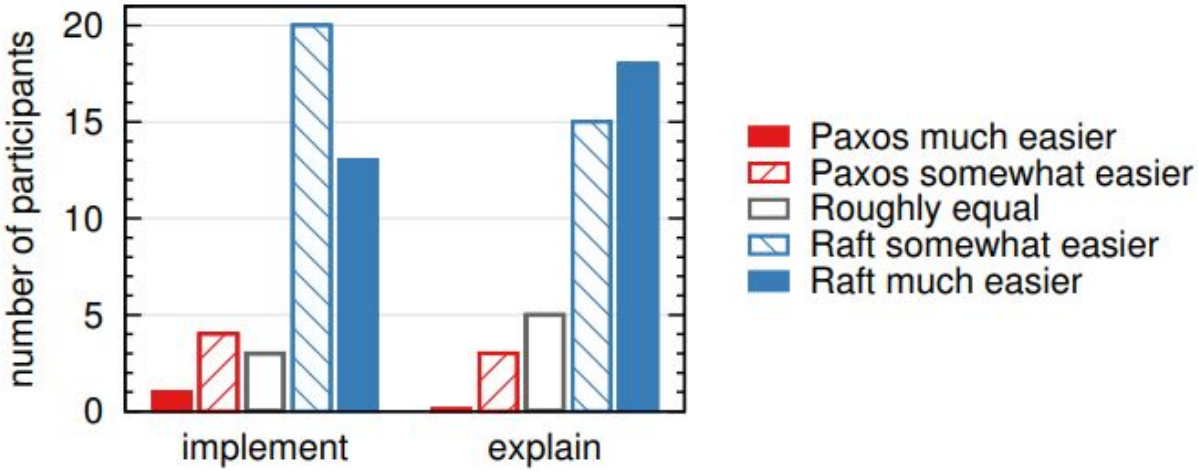
Goal for Raft: implement **linearizable semantics**

- The result of the read request is the result that was committed when the request was initiated
- Make the outcome of the read request a consensus at most nodes
  - Linearizable reads must be sent to leader
  - The Leader must have been submitted one log during its term, before it can respond to a client's read request

Evaluation

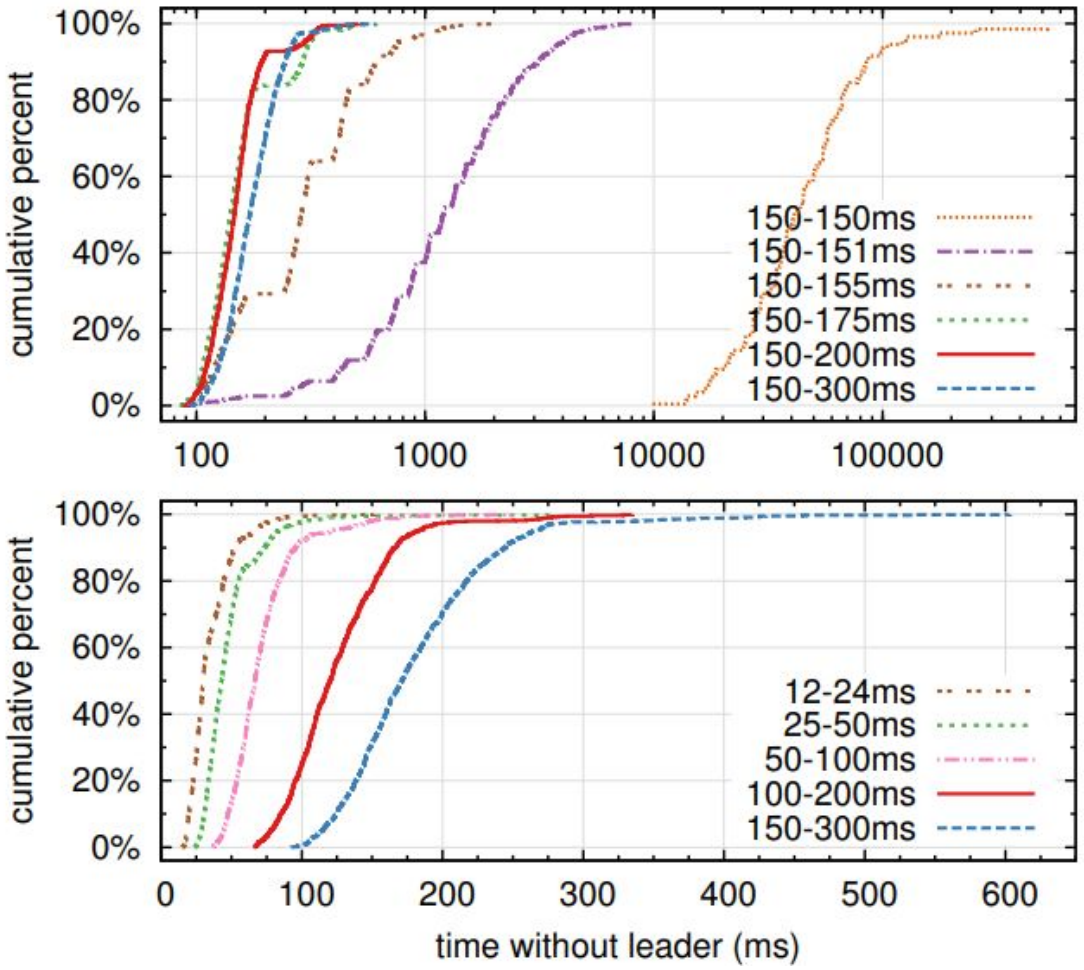
1. **Understandability**





Evaluation

2. Performance



Questions?

Universiteit Leiden
The Netherlands