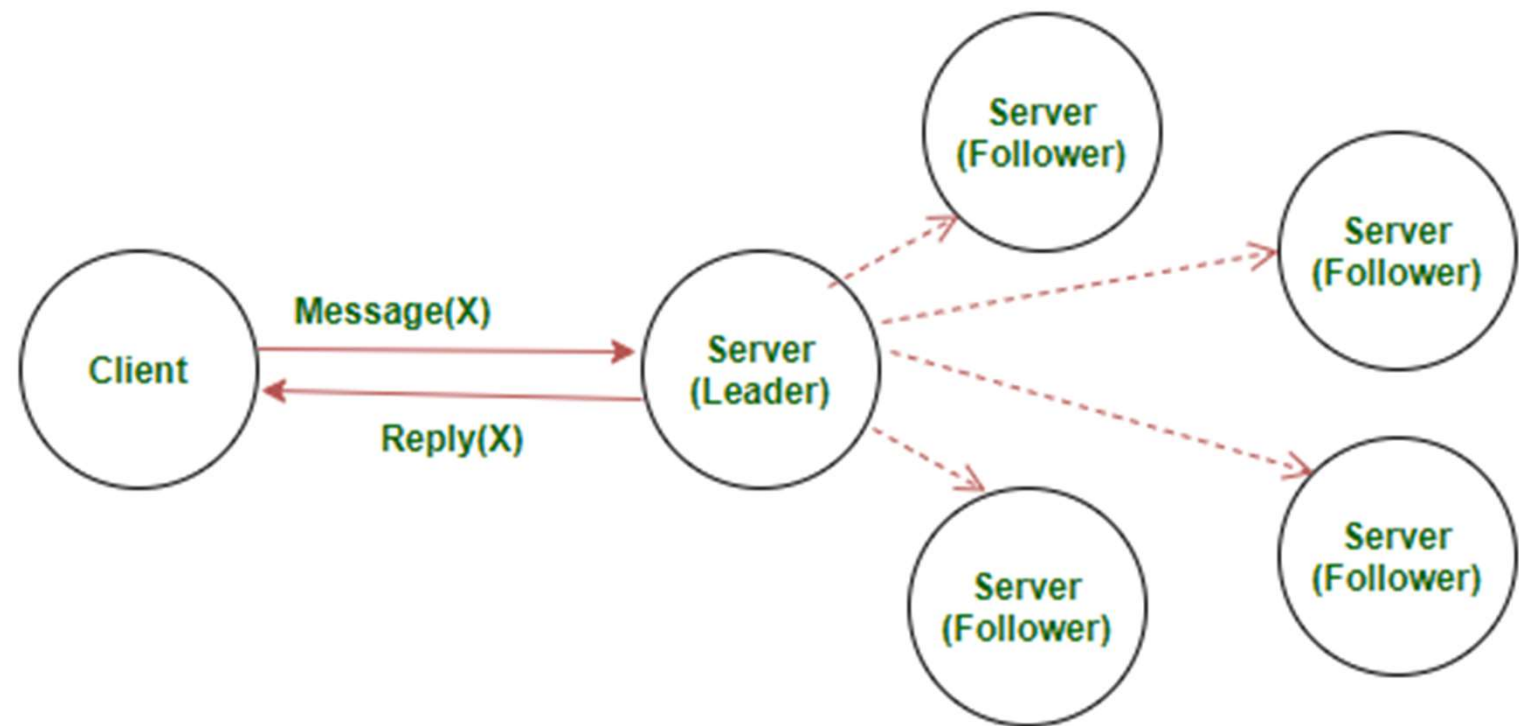


The Raft Consensus Algorithm

Interactive Guide

- <http://thesecretlivesofdata.com/raft/>
 - - Let's play around with this interactive guide to understand.

What is RAFT?



What is Consensus?

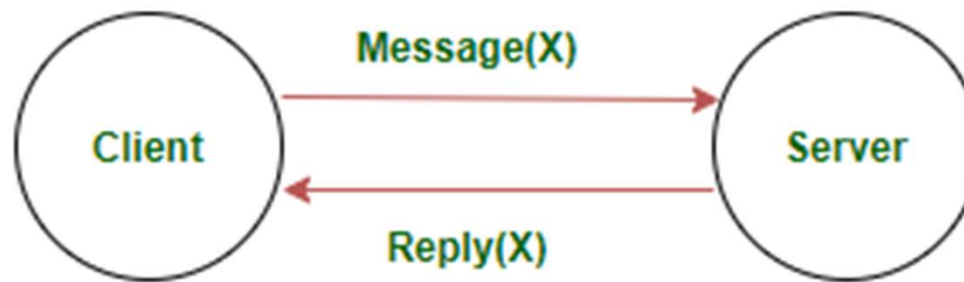
- Which problem Raft protocol tries to solve?
 - That is achieving Consensus
- Consensus means multiple servers agreeing on same information

Client server interaction Process

- The client sends a message to the server and the server responds back with a reply.
- A consensus protocol tolerating failures must have the following features :
 - Validity : If a process decides(read/write) a value, then it must have been proposed by some other correct process
 - Agreement : Every correct process must agree on the same value
 - Termination : Every correct process must terminate after a finite number of steps.
 - Integrity : If all correct processes decide on the same value, then any process has the said value

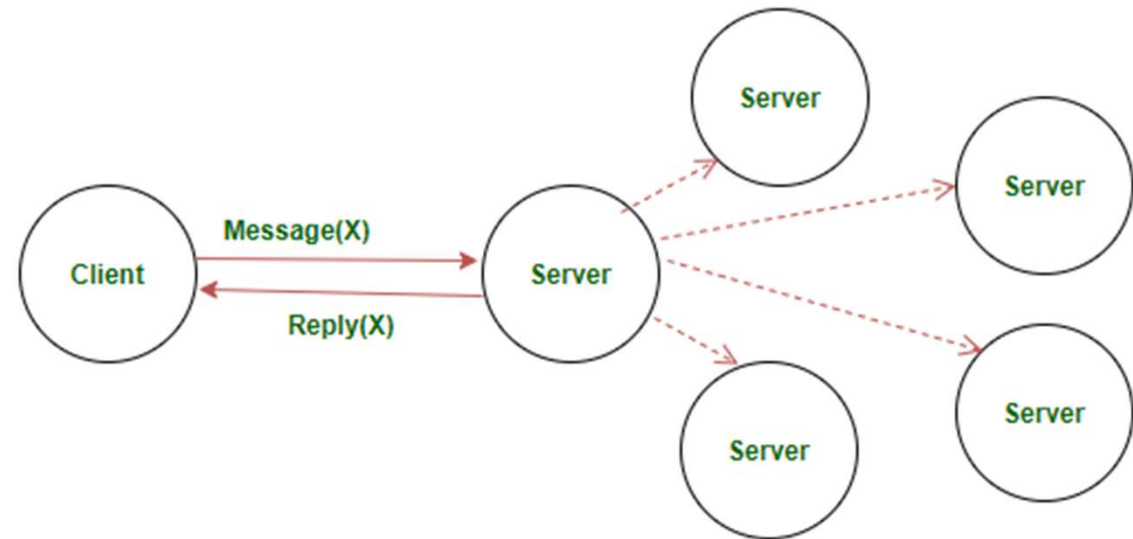
Single Server system

- The client interacts with a system having only one server with no backup
- There is no problem in achieving consensus in such a system.



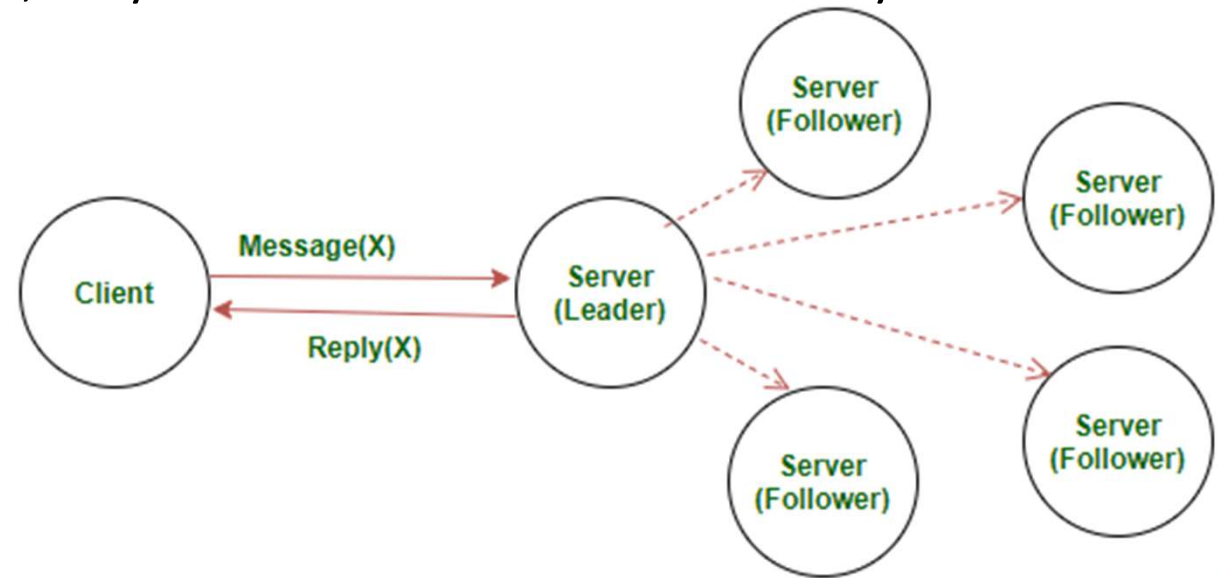
Multiple Server system

- The client interacts with a system having multiple servers
- Such systems can be of two types :
 - Symmetric :- Any of the multiple servers can respond to the client and all the other servers are supposed to sync up with the server that responded to the client's request, and
 - Asymmetric :- Only the elected leader server can respond to the client. All other servers then sync up with the leader server.
- Given below is an example of an asymmetric multiple server system



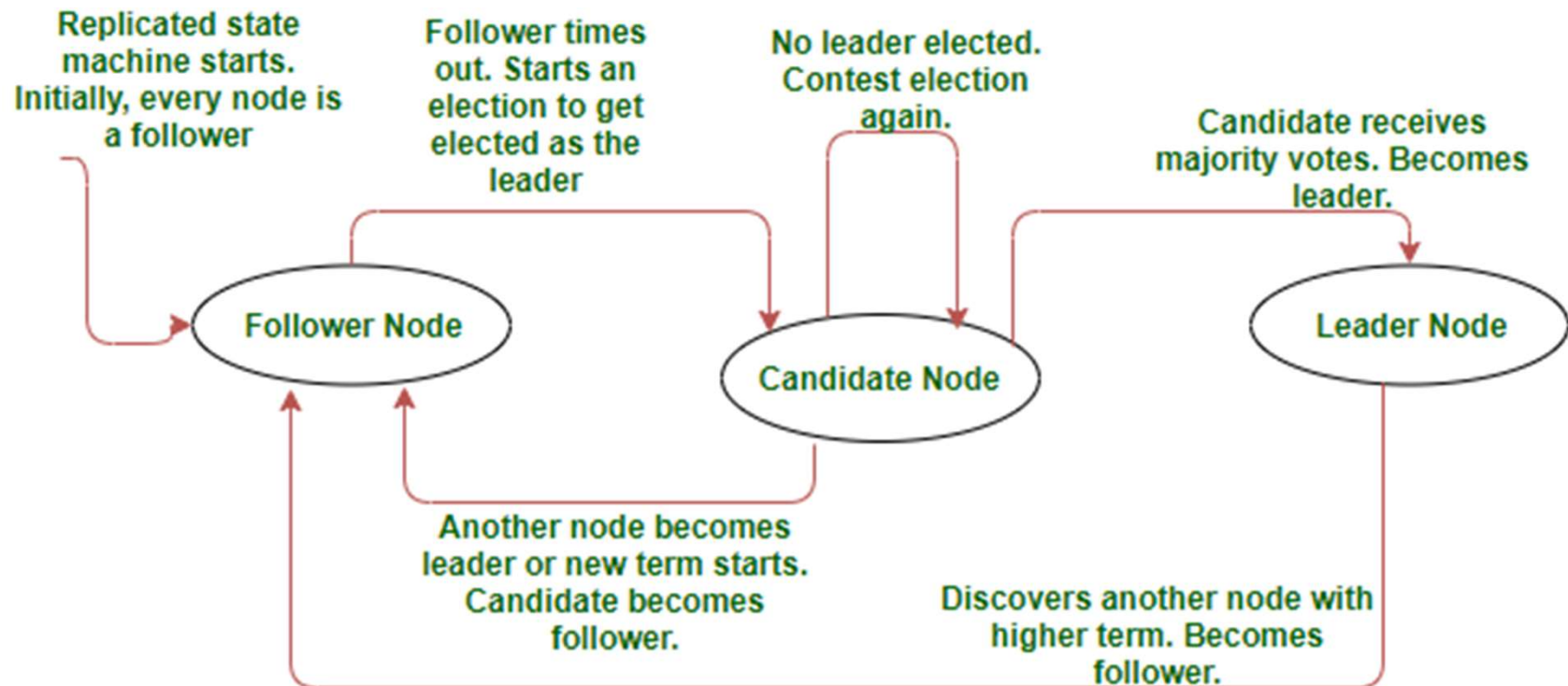
Replicated state machine

- System in which all the servers maintain similar data across time
- Terms used to refer individual servers in a distributed system
 - Leader – Only the server elected as leader can interact with the client
 - Follower – Follower servers sync up their copy of data with that of the leader's
 - Candidate – At the time of contesting an election to choose the leader server, the servers can ask other servers for votes. Hence, they are called candidates when they have requested votes.



What is the Raft protocol

- A consensus algorithm
- Raft states that each node in cluster can stay in any of the three states, namely, leader, candidate, follower



What is the Raft protocol

- To maintain these server status(es), the Raft algorithm divides time into small terms of arbitrary length
- Each term is identified by a monotonically increasing number, called term number.

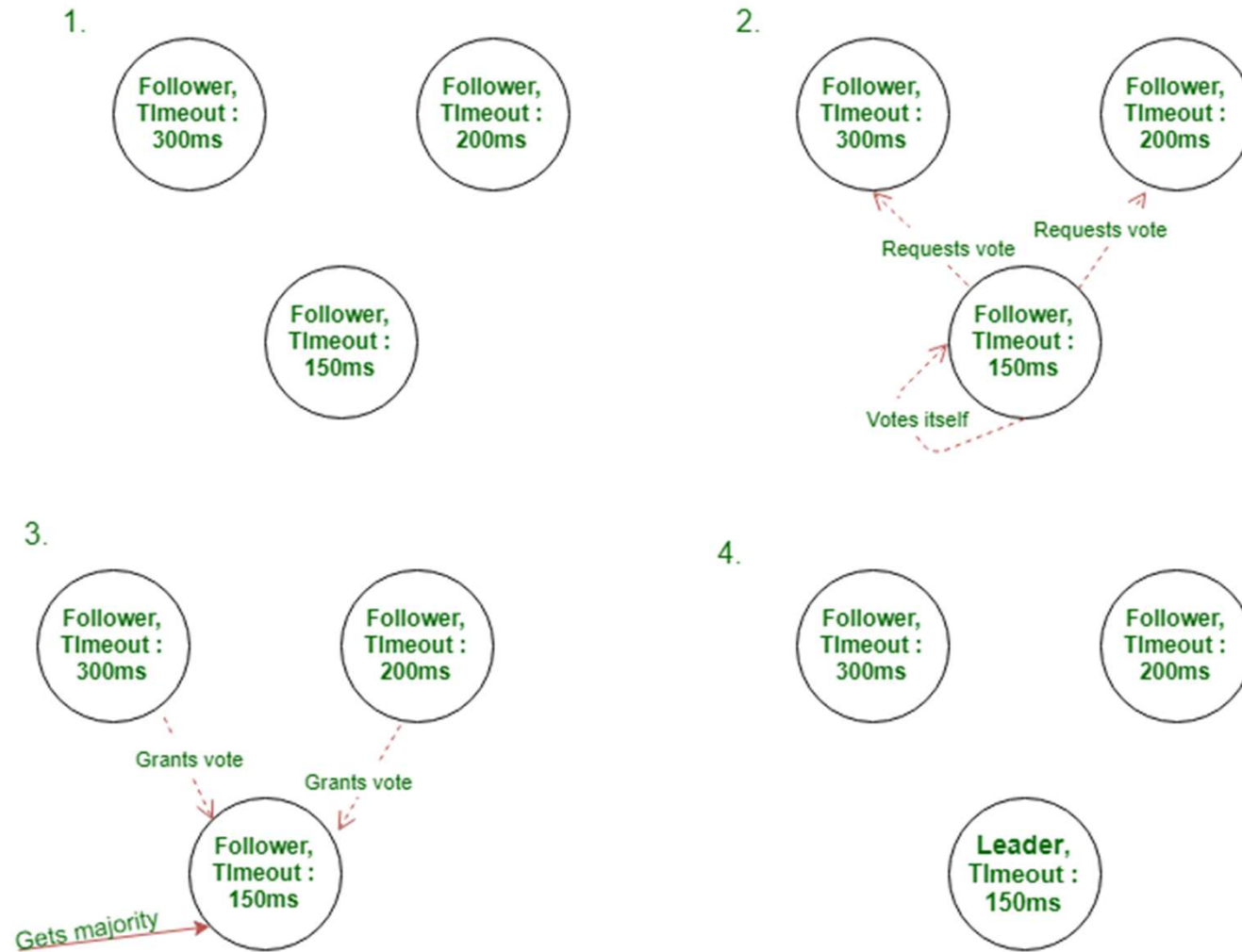
Term number

- Is maintained by every node
- Is passed while communications between nodes
- Every term starts with an election to determine the new leader
- The candidates ask for votes from other server nodes(followers) to gather majority
- If the majority is gathered, the candidate becomes the leader for the current term
- If no majority is established, the situation is called a split vote and the term ends with no leader
- Hence, a term can have at most one leader

Leader election

- Leader node sends heartbeat to express dominion to other Follower nodes
- A leader election takes place when a Follower node times out while waiting for a heartbeat from the Leader node
- At this point of time, the timed out node
 - Changes its state to Candidate state,
 - Votes for itself and
 - Issues RequestVotes RPC to establish majority and attempt to become the Leader

Leader election



Log Replication

- Each request made by the client is stored in the Logs of the Leader
- This log is then replicated to other nodes(Followers)
- Typically, a log entry contains the following three information :
 - Command/Data specified by the client to execute
 - Index to identify the position of entry in the log of the node
 - Term Number to ascertain the time of entry of the command.
- The Leader node fires AppendEntries RPCs to all other servers(Followers) to sync/match up their logs with the current Leader
- The Leader keeps sending the RPCs until all the Followers safely replicate the new entry in their logs.

Log Replication

- When the majority of the servers in the cluster successfully copy the new entries in their logs, it is considered committed
- At this point, the Leader also commits the entry in its log to show that it has been successfully replicated.
- After the entry is committed, the leader executes the entry and responds back with the result to the client.