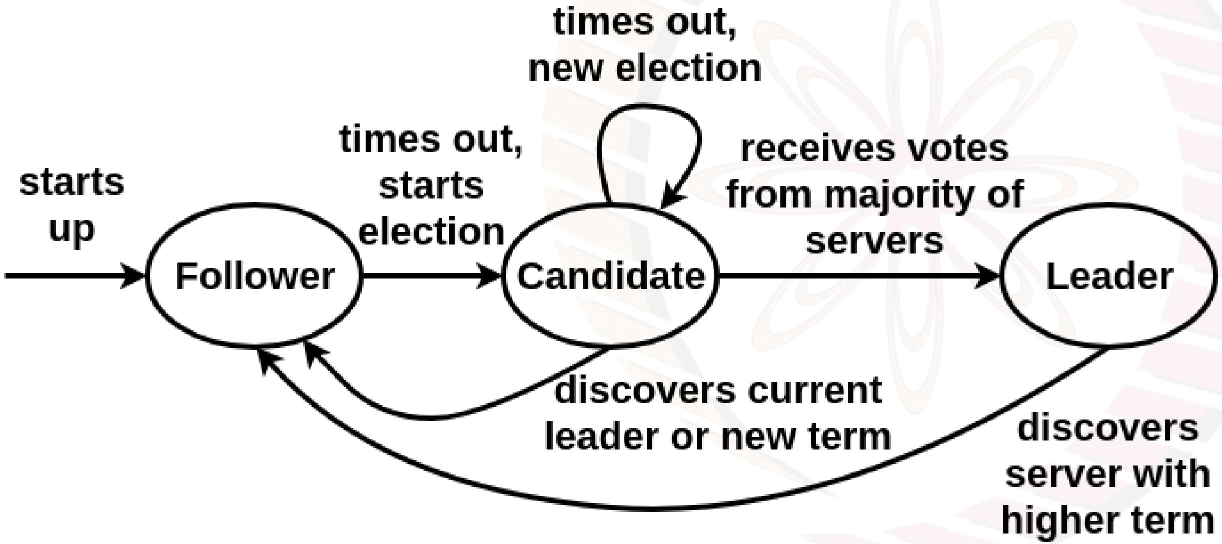
RAFT Consensus

- Basic idea -
  - The nodes collectively selects a *leader*; others become *followers*
  - The leader is responsible for state transition log replication across the followers

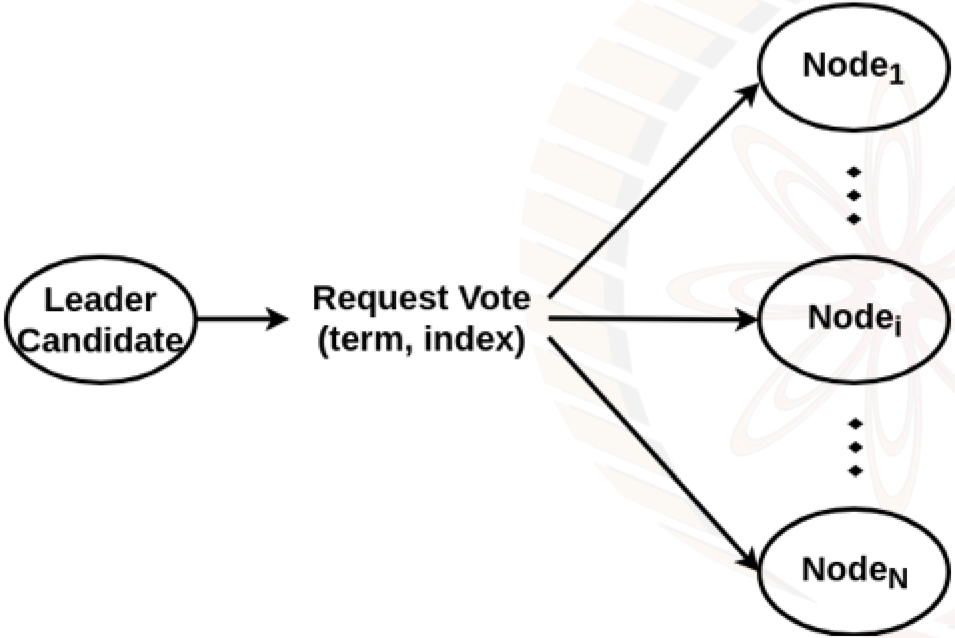
RAFT


starts up
Follower
times out, starts election
Candidate
times out, new election
receives votes from majority of servers
Leader
discovers current leader or new term
discovers server with higher term

- (re)electing a leader
- committing multiple values to the transaction log
- dealing with replicas failing

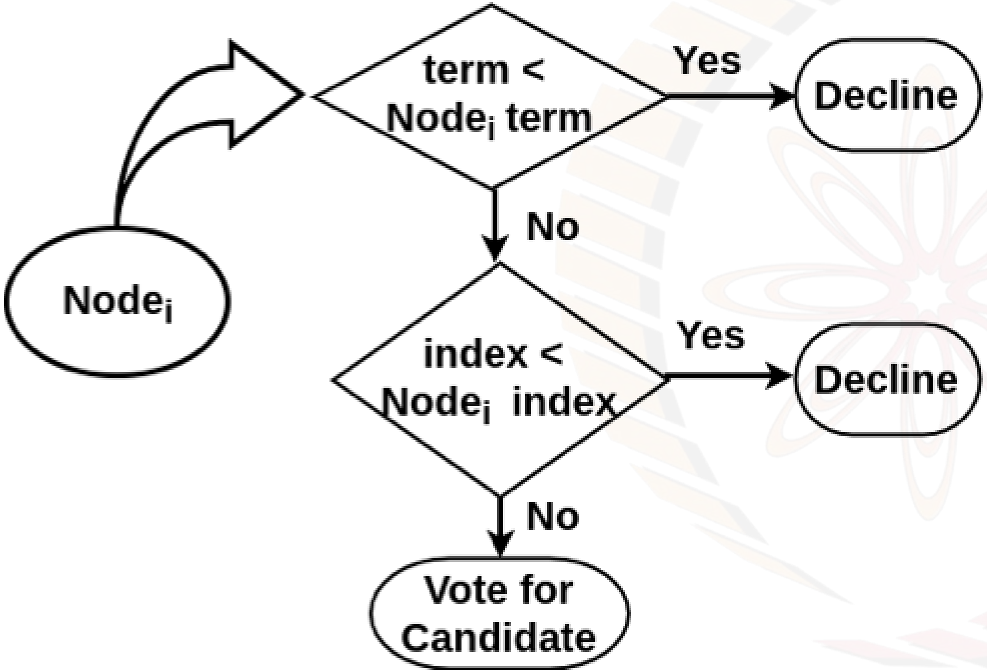
Electing the Leader: Voting Request

Some body just becomes leader candidate



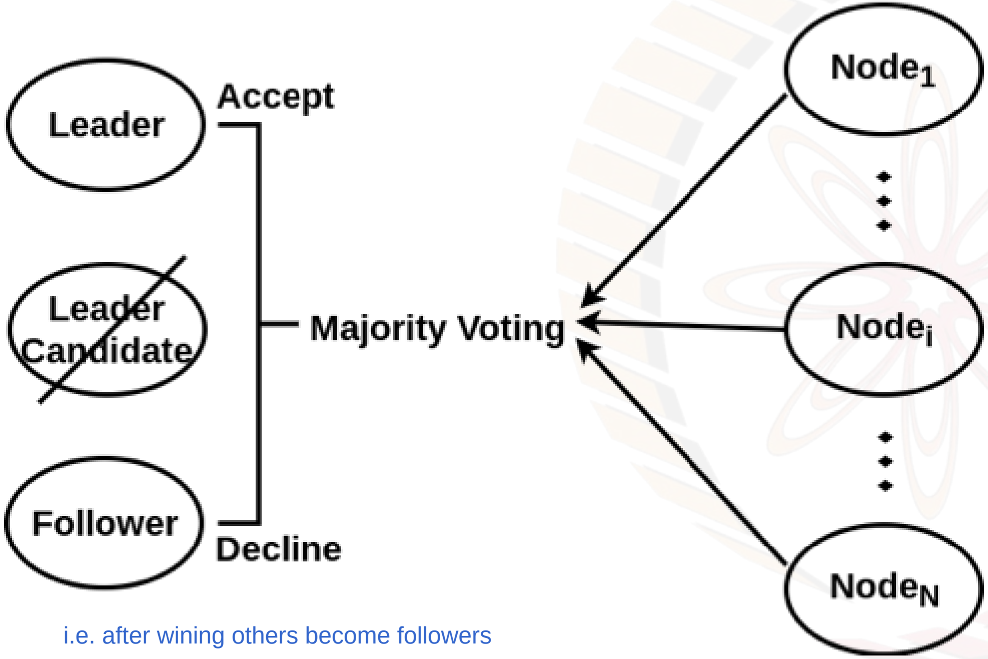
- **term:** last calculated # known to candidate + 1
- **index:** committed transaction available to the candidate

Electing the leader: Follower Node’s Decision Making



- Each node compares received term and index with corresponding current known values

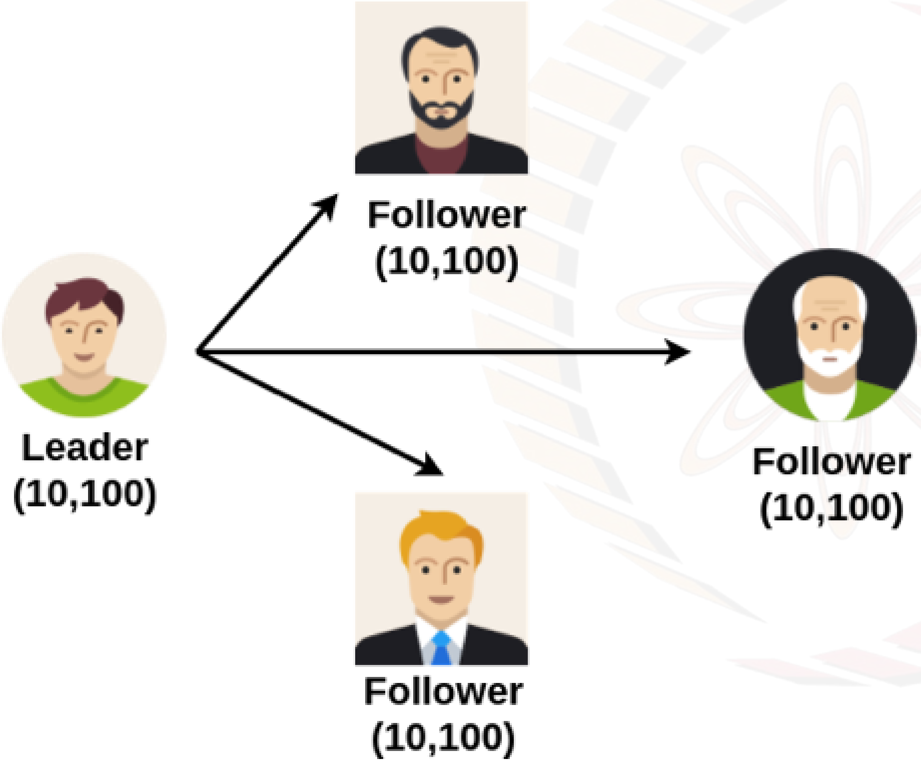
Electing the leader: Majority Voting



i.e. after wining others become followers

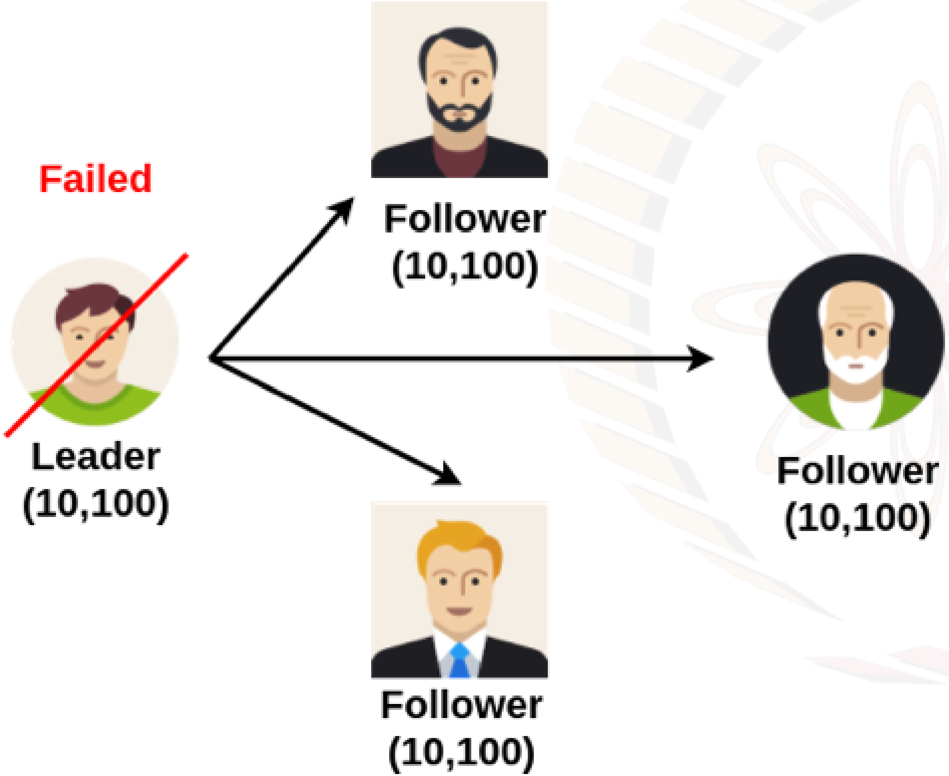
- Use of Majority voting
  - leader selection
  - commit the log entry

Multiple Leader Candidates: Current Leader Failure



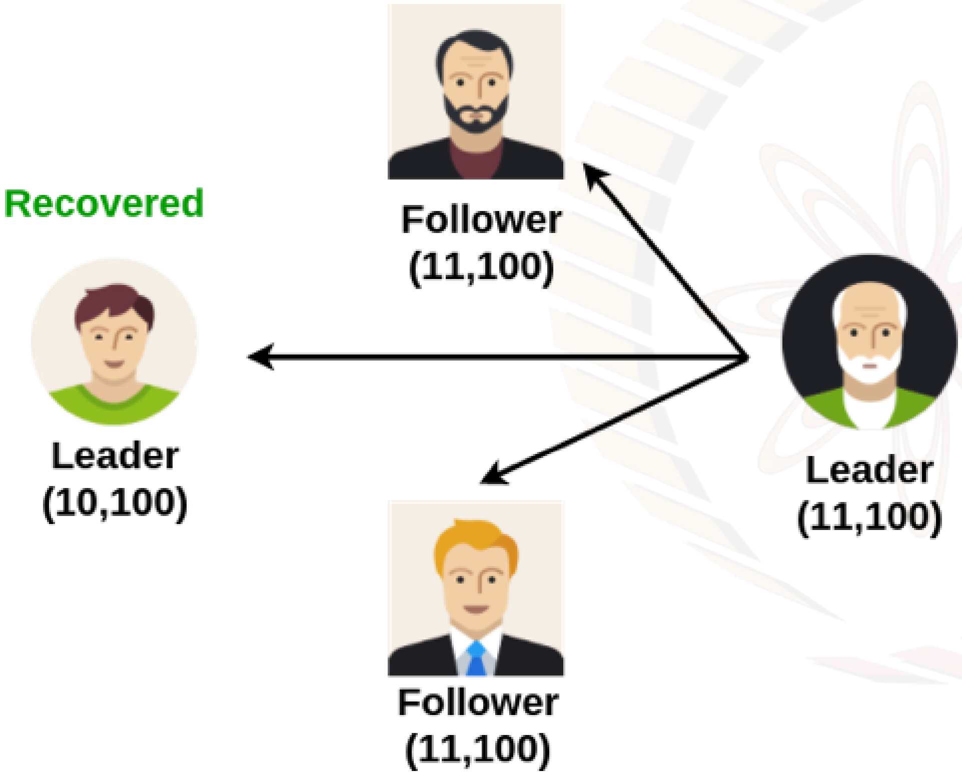
- A leader with three followers
- term: 10
- commit index: 100

Multiple Leader Candidates: Current Leader Failure



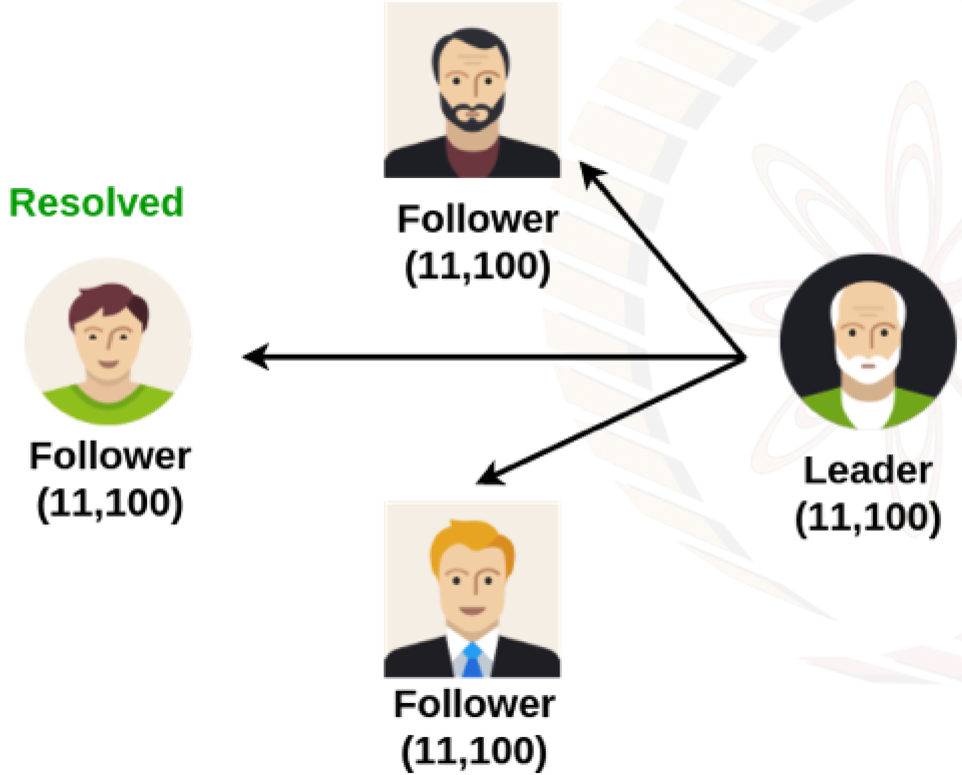
- The leader node failed

Multiple Leader Candidates: Current Leader Failure



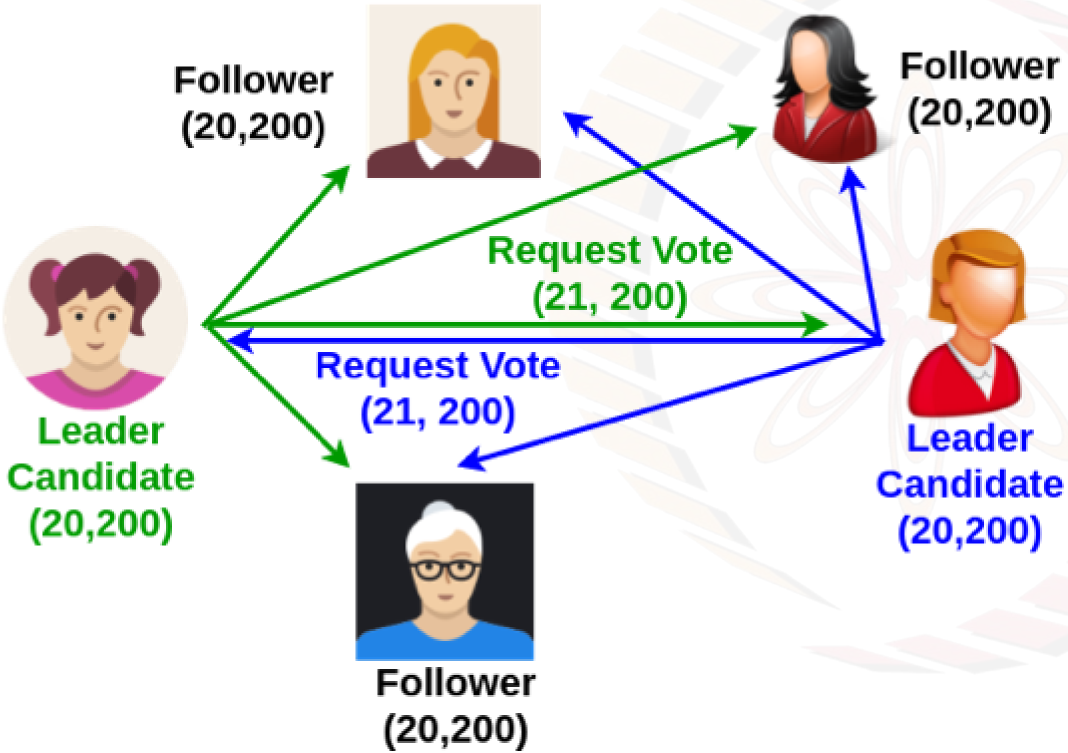
- New leader elected with term 11
- Old leader recovered

Multiple Leader Candidates: Current Leader Failure



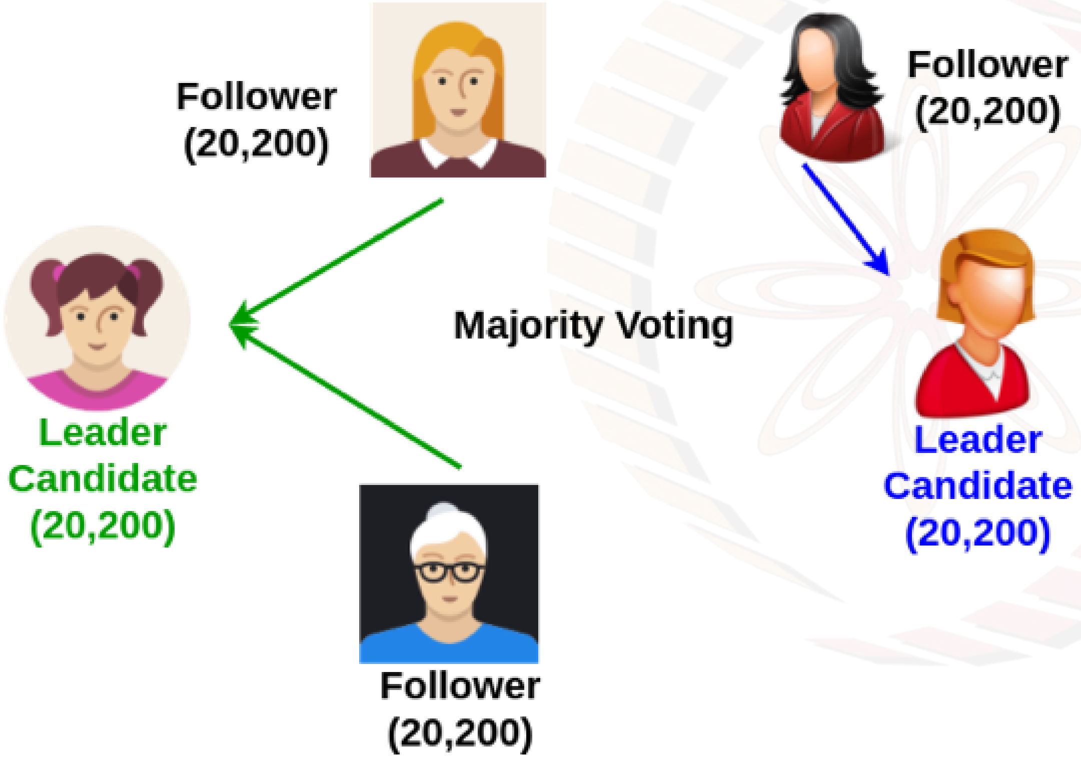
- Old leader receive heartbeat message from new leader with greater term
- Old leader drops to follower state

Multiple Leader Candidates: Simultaneous Request Vote



- Two nodes send Request vote message with term 21 at the same time

Multiple Leader Candidates: Simultaneous Request Vote



Follower (20,200)

Follower (20,200)

Leader Candidate (20,200)

Majority Voting

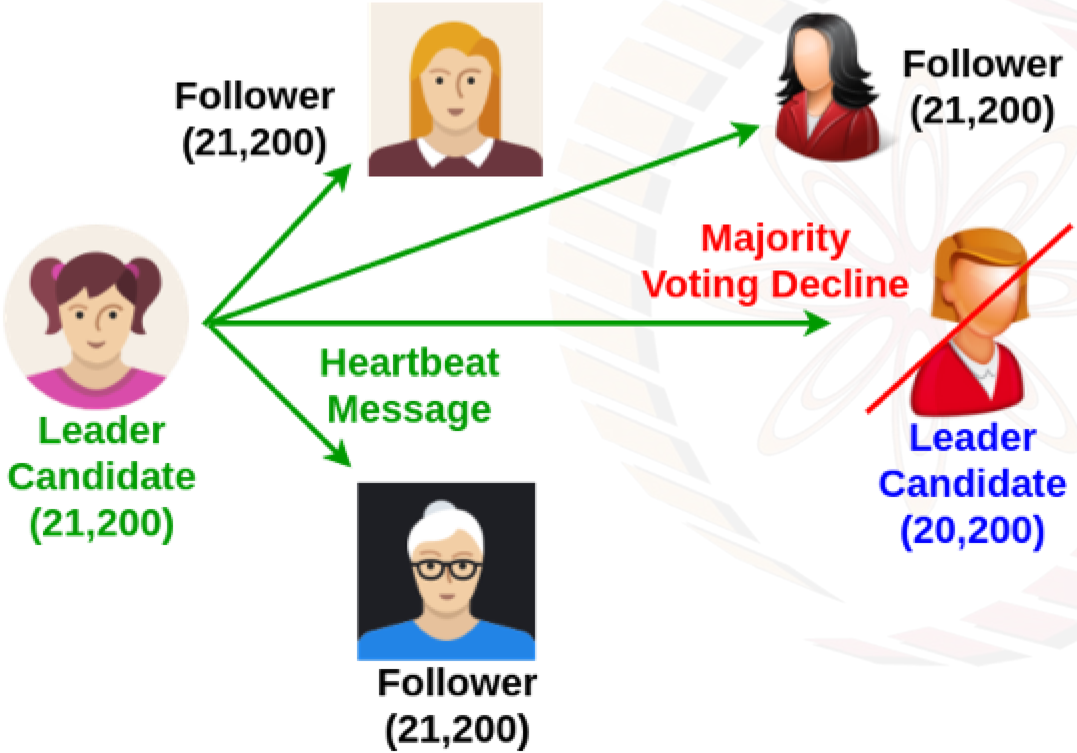
Follower (20,200)

Leader Candidate (20,200)

Follower (20,200)

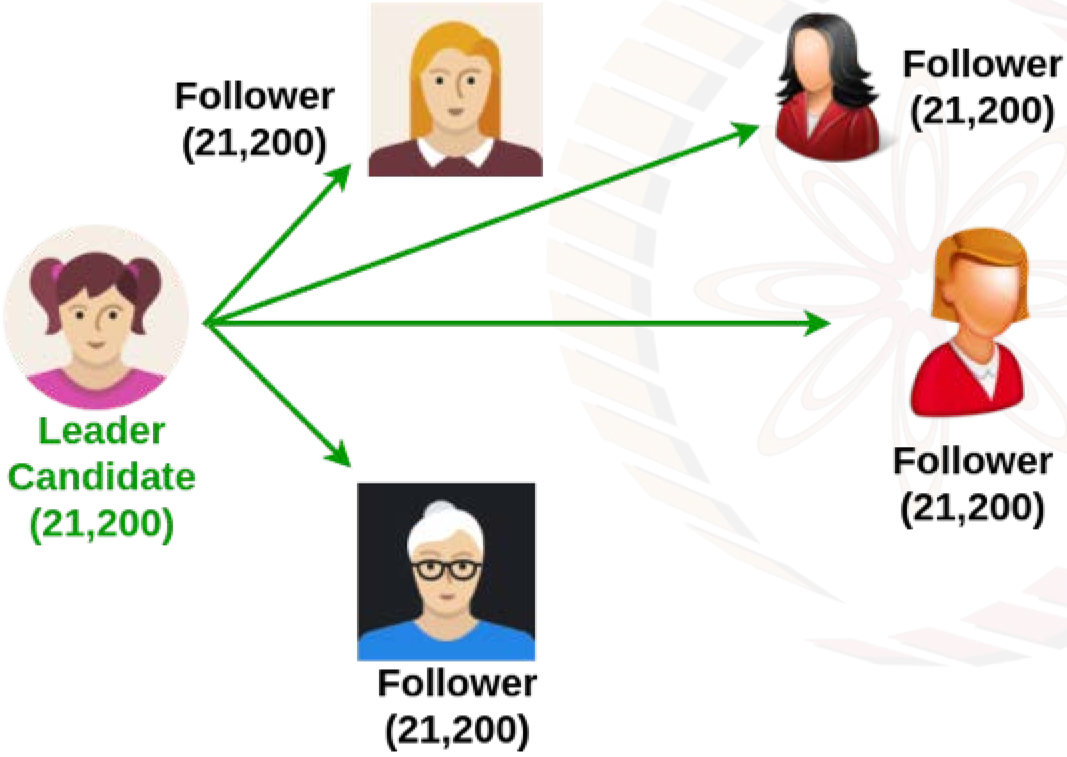
- One of them gets majority voting

Multiple Leader Candidates: Simultaneous Request Vote



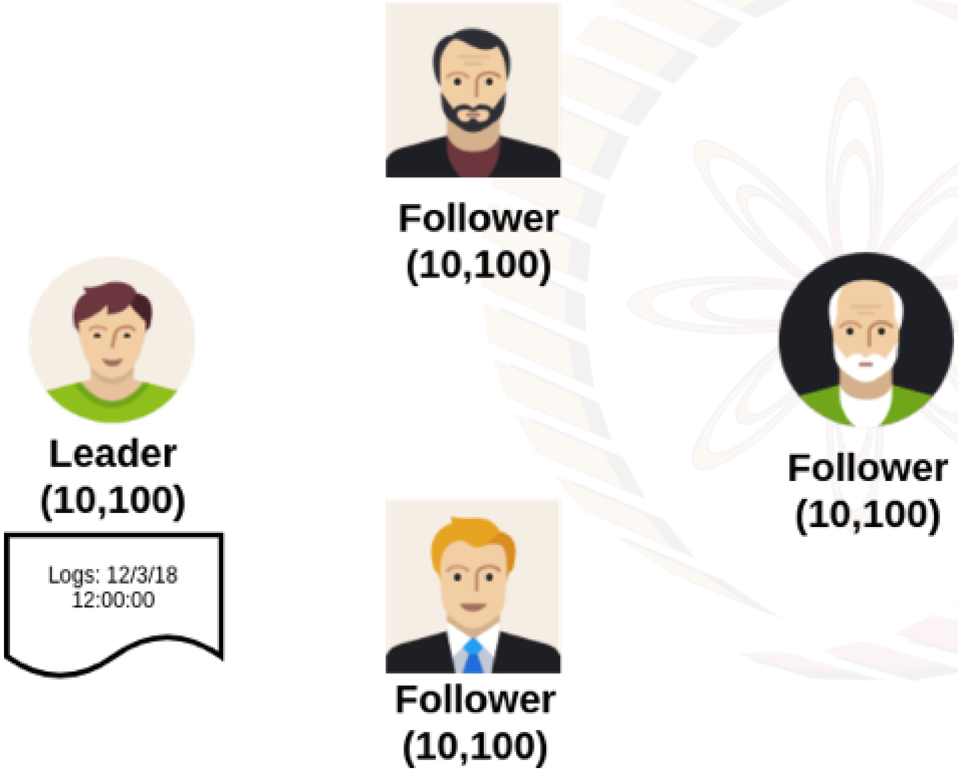
- Winner sends heartbeat message

Multiple Leader Candidates: Simultaneous Request Vote



- Other leader candidate switches to follower state

Commiting Entry Log



Follower
(10,100)

Leader
(10,100)

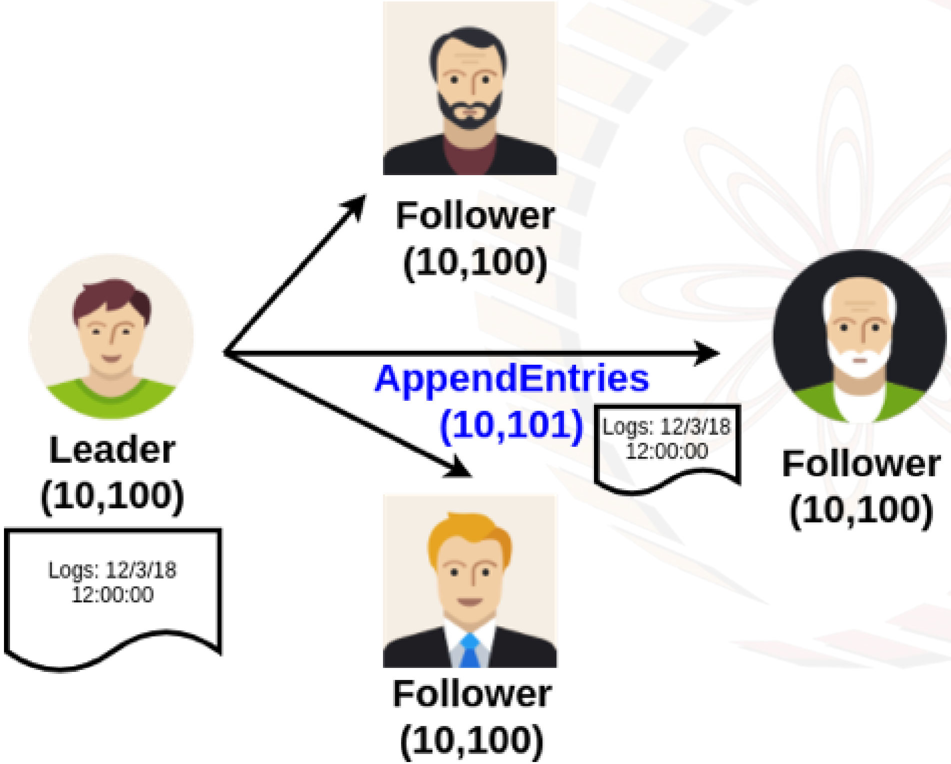
Logs: 12/3/18
12:00:00

Follower
(10,100)

Follower
(10,100)

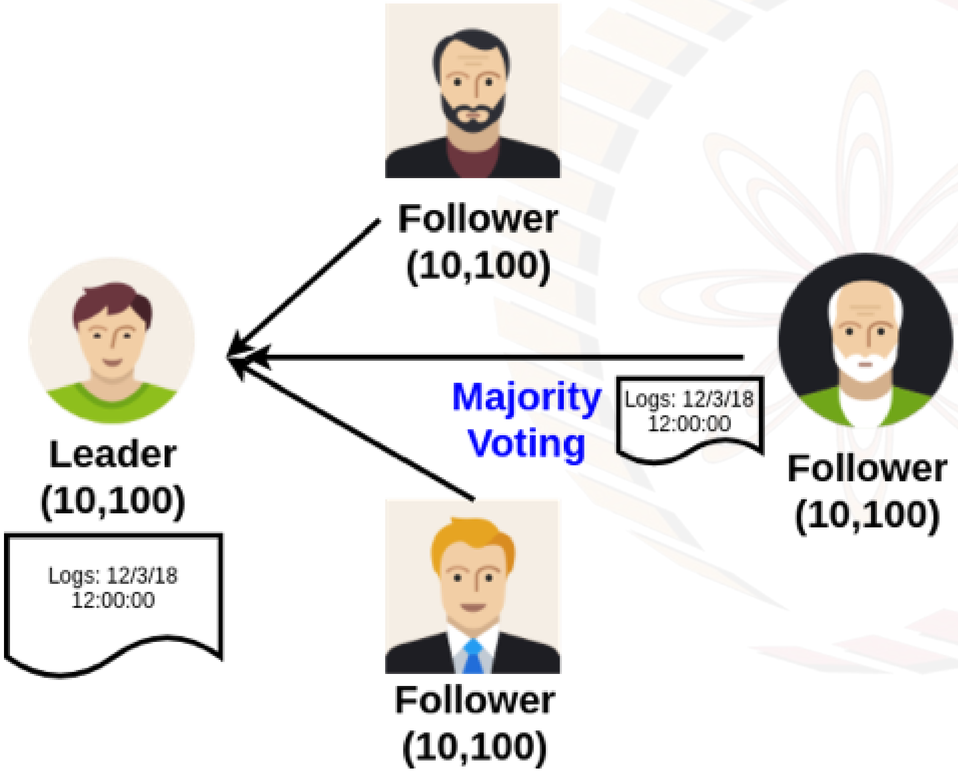
- Leader adds entry to log with term 10 and index 101

Commiting Entry Log



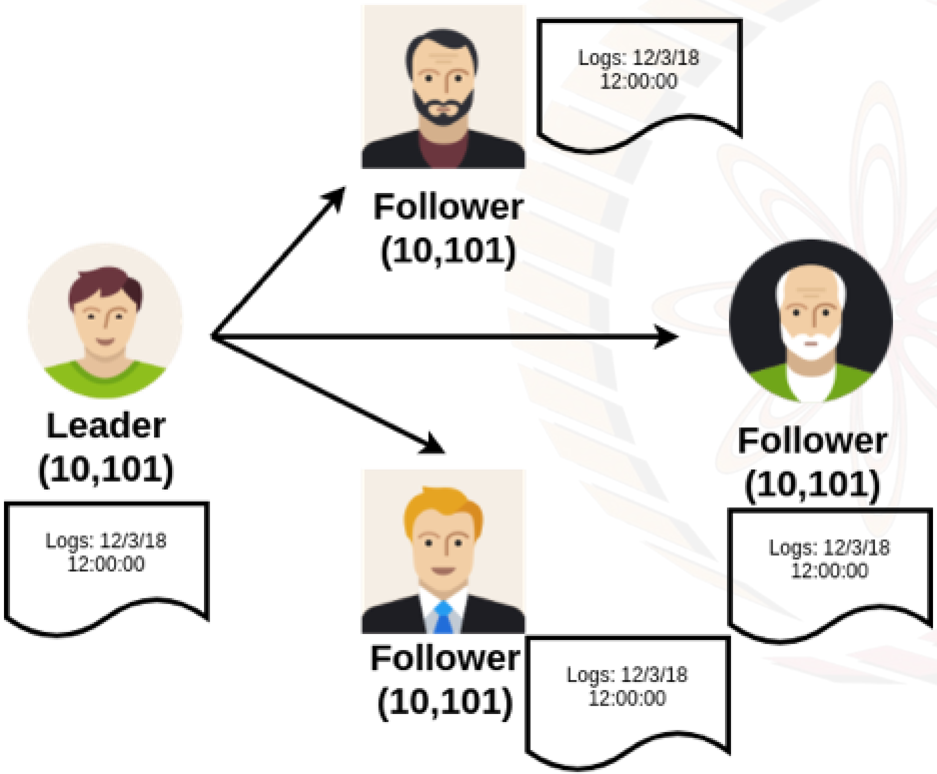
- Leader sends *AppendEntries* message to followers with index 101

Commiting Entry Log



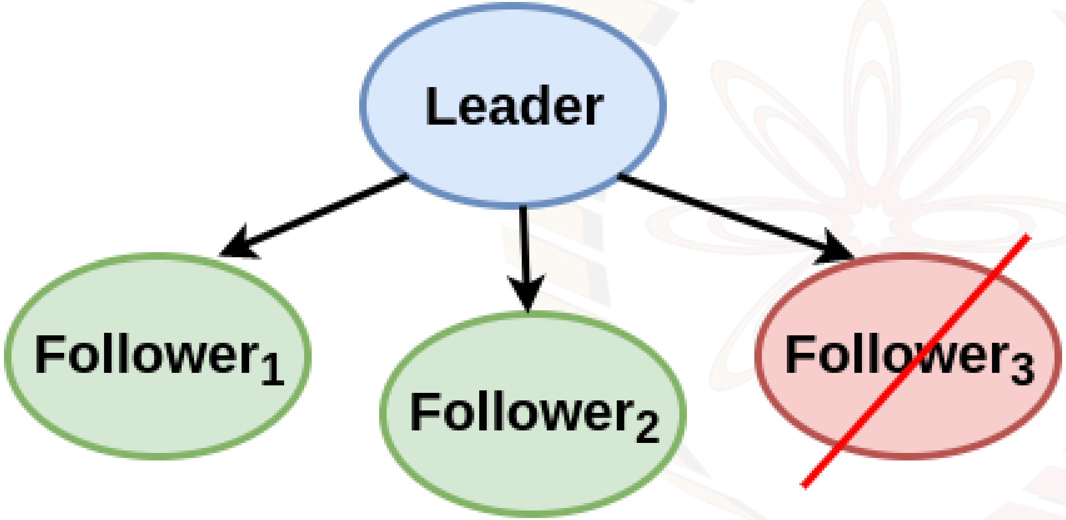
- Majority voting decides to accept or reject the entry log

Commiting Entry Log



- Successfully accept entry log
  - All leader and followers update committed index to 101

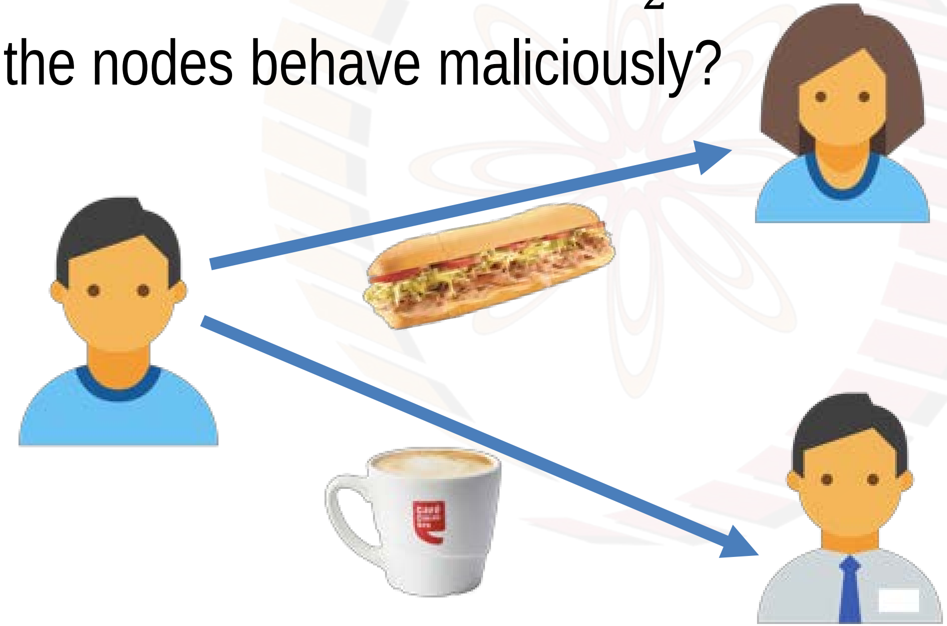
Handling Failure



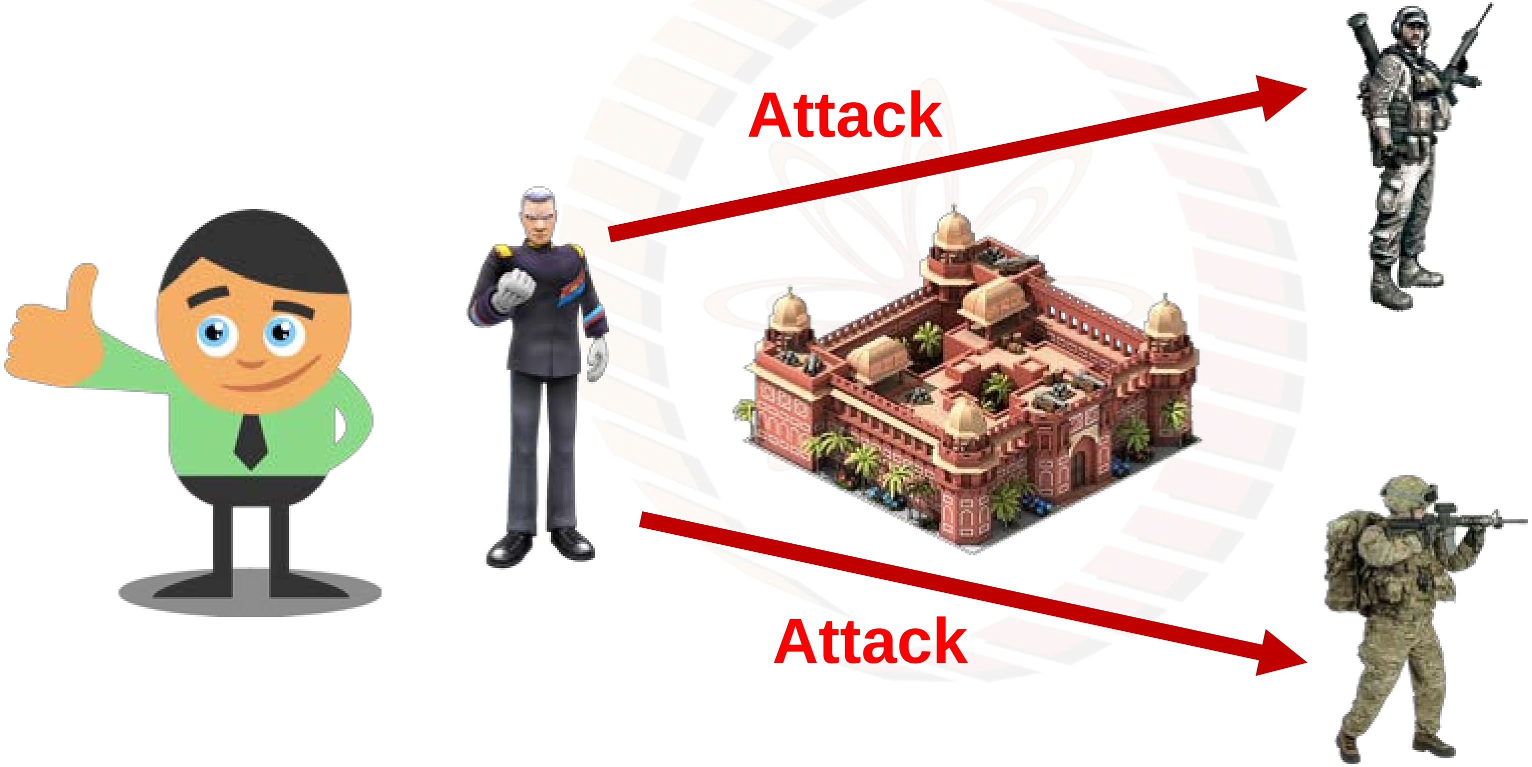
- Failure of up to N/2 - 1 nodes does not affect the system due to majority voting

Byzantine Generals Problem

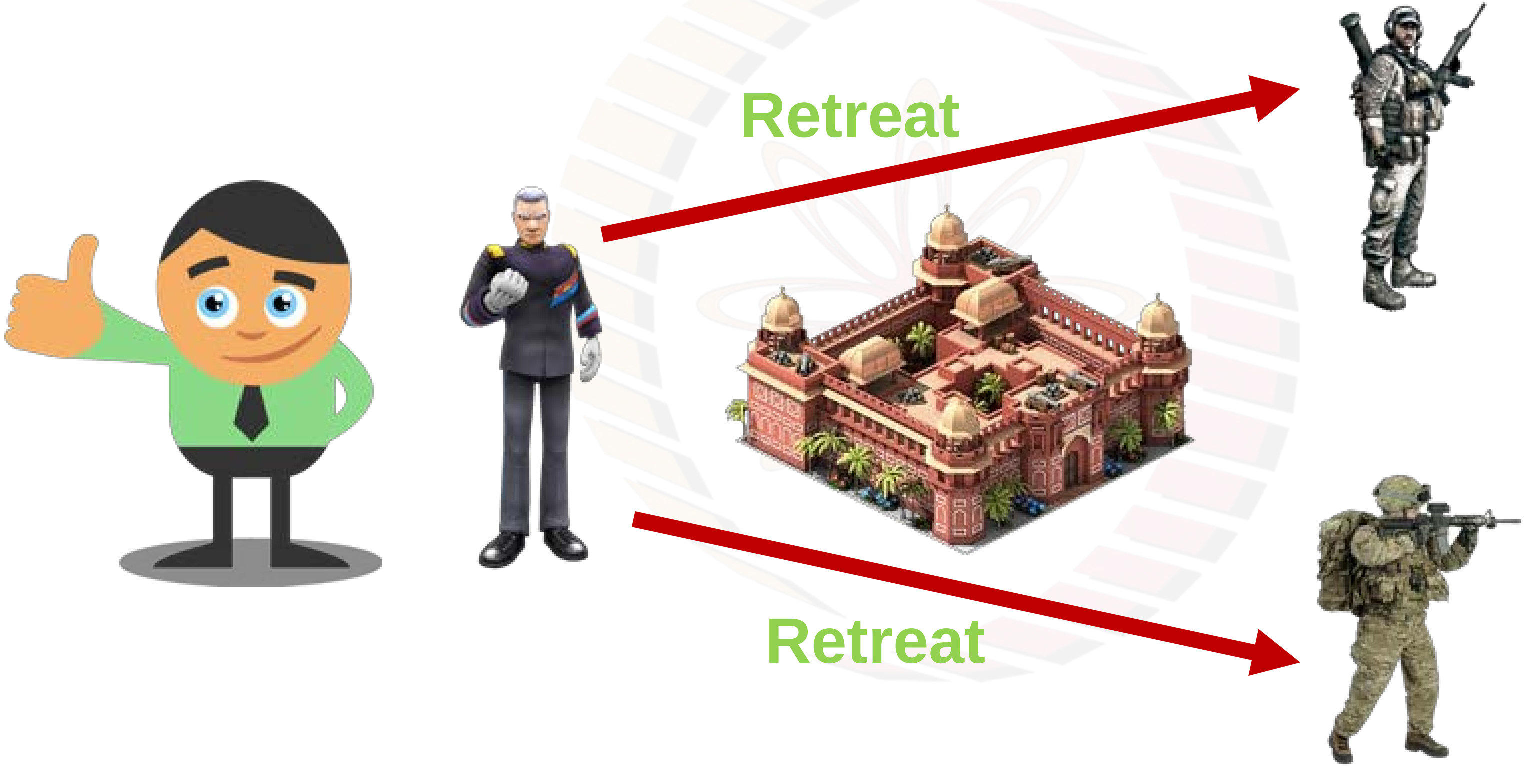
- Paxos and Raft can tolerate up to $\frac{N}{2} - 1$ number of crash faults
- What if the nodes behave maliciously?



Byzantine Generals Problem



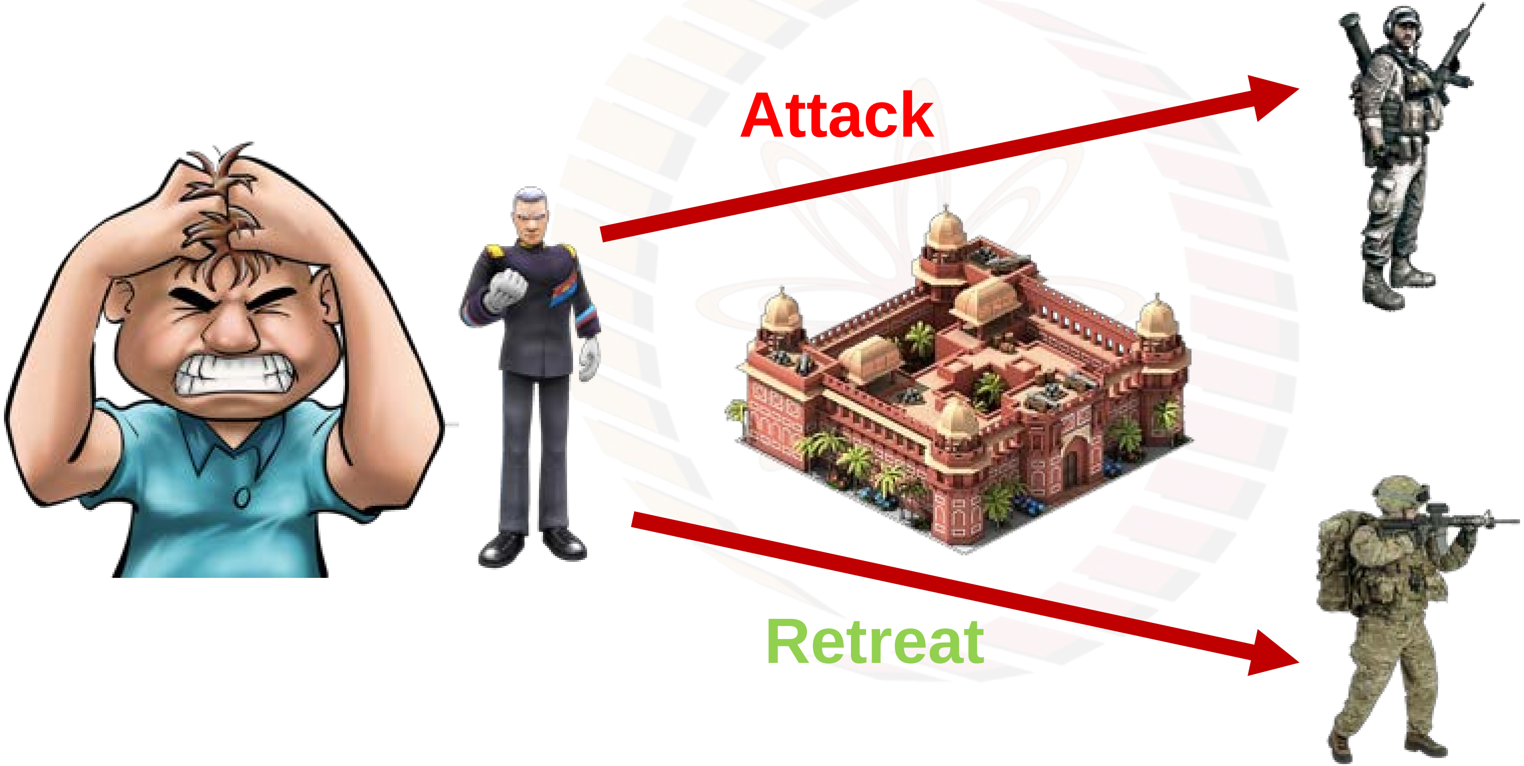
Byzantine Generals Problem



Retreat

Retreat

Byzantine Generals Problem



Attack

Retreat