Basic

Server
- follower
- candidate
- leader

Raft divides tme into terms of arbitrary length

term1   term2

terms

election   nomral operation

Each server stores a current term number → exchange term number when communicating

start → follower

follower — times out, start election → candidate

candidate — times out, new election → candidate

candidate — receives votes from majority of servers → Leader

candidate — discovers current leader or new term → follower

Leader — discovers server with higher term → follower

Raft Outline

Raft
- Leader election → A new leader must be chosen when an existing leader fails
- Log replication → the leader must accept log entries from clients and replicate them acorrs the cluster
- Safety
  - Election safety
  - Leader Append-Only
  - Log Matching
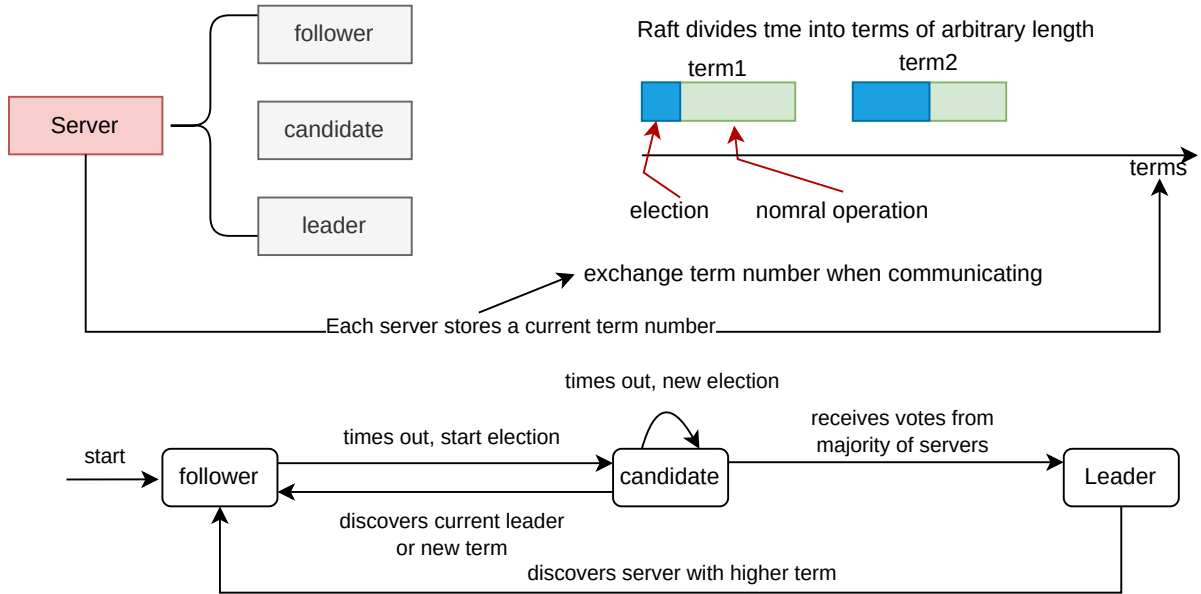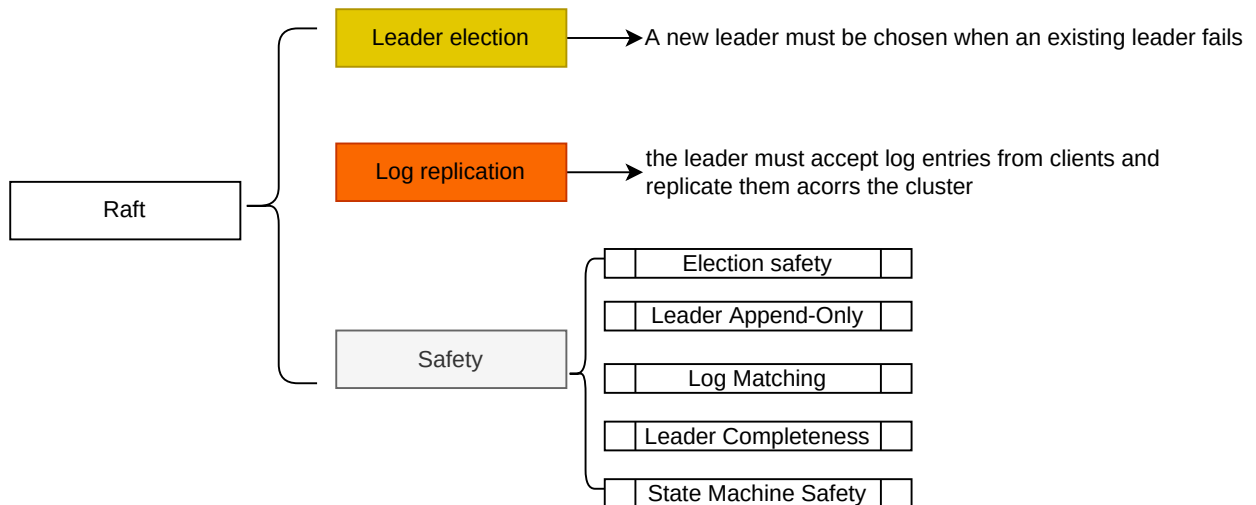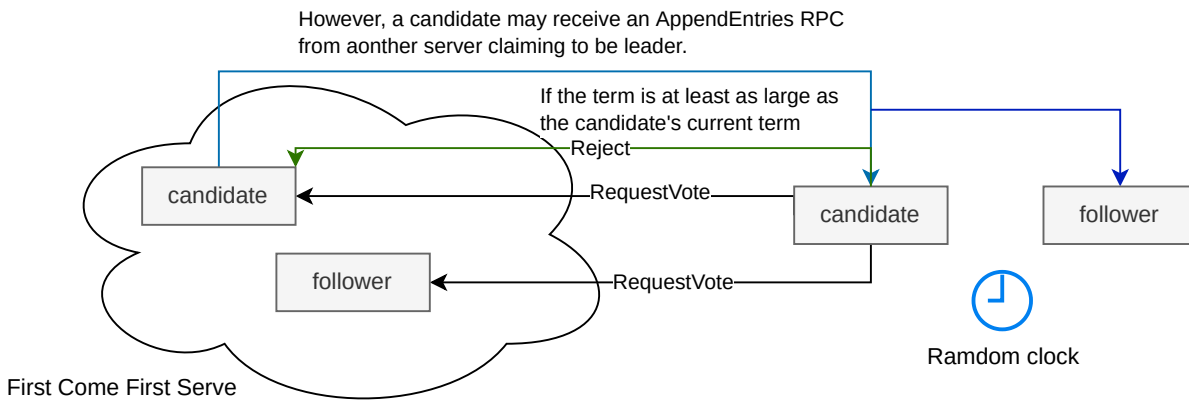 - Leader Completeness
  - State Machine Safety

Leader Election

The process
A follower increments its current term and transitions to candidate state.
Votes for iteself and issue RequestVote RPCs in parallel to other servers
Until (a) it wins (b) timeout, no winner

However, a candidate may receive an AppendEntries RPC
from aonther server claiming to be leader.

If the term is at least as large as
the candidate's current term

Reject

candidate ← RequestVote ← candidate   follower

follower ← RequestVote

First Come First Serve

Ramdom clock

A candidate must contact a majority of the cluster in order to
be elected, which means thatevery committed entry must be
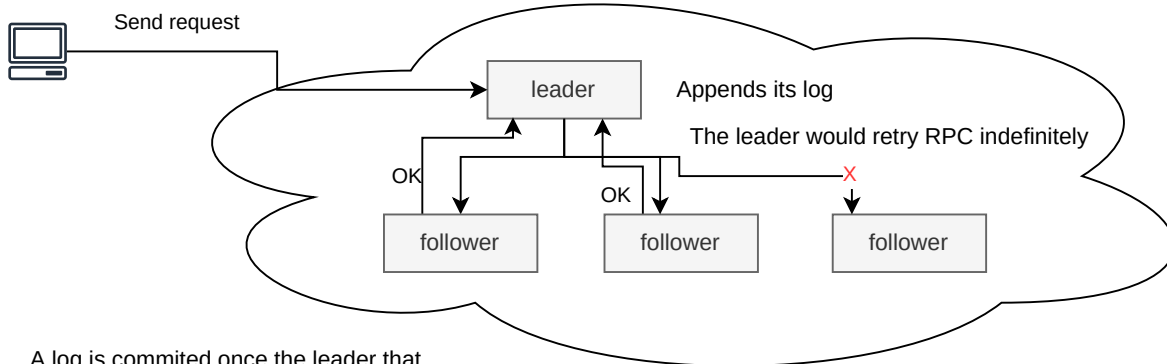present in at least one of those servers.
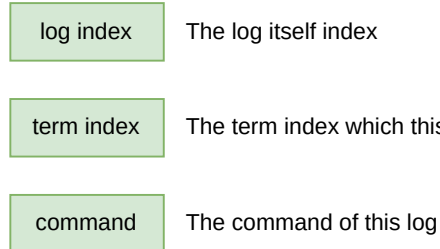
The criterion of the more up-to-date

Compare the terms, greater means more up-to-date ⟶ **Election Safety**

If the trems are equal, longer log means more up-to-date

Send request

leader   Appends its log

The leader would retry RPC indefinitely

follower   follower   follower

OK   OK   X

A log is commited once the leader that created the entry has replicated it on a majority of the servers.

Log

log index   The log itself index

term index   The term index which this log belongs to

command   The command of this log
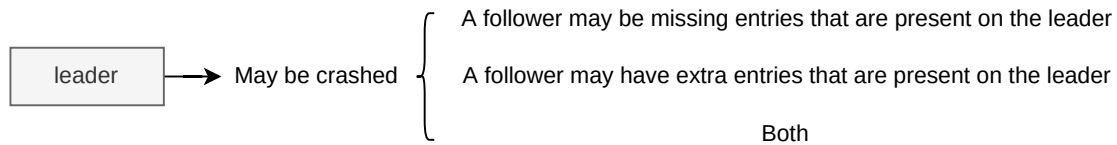

Log Macthing Property

If two entries in different logs have the same index and term, then they store the same command

If two entries in different logs have the same index and term, then the logs are identical in all preceding entries.

Induction Proof

Initial, the leader send the first log, which would must match log matching property

Next, when sending the next log, leader sends the previous log's log index and term index, followers insepct it, if it finds an entry, accept it. Otherwise, refuse it

leader → May be crashed

- A follower may be missing entries that are present on the leader
- A follower may have extra entries that are present on the leader
- Both

How to make the followers' log consistent with the leader

↓

Forcing the followers' logs to duplicate the log of the leader

leader

Maintain a nextIndex for each follower

1 2 3 4 5 6 7 8 9 10

1 1 1 4 4 5 5 6 6 6   Set a nextIndex = 11 for each follower

1 1 1 4 4 5 5 6 6   nextIndex 11 failed, decrement 1, success

1 1 1 4   nextIndex 11 failed, decrement 1, ..., nextIndex = 5, success

Leader Append Only → 1 1 1 2 2 2   nextIndex = 4, success, removes three 2's

1 1 1 4 4 5 5 6 6 6 6 6   success, removes extra two 6's