**In Search of an Understandable Consensus Algorithm**

**Summary:**

Raft is a distributed consensus algorithm with an understandable design used for managing a replicated log. Collection of machines work in cluster or coherent group and can overcome failure of some machines.

Raft solve the problem by defining roles.  Every node in a raft cluster have one of three roles, leader, follower, or candidate. In general there is a single leader, and every other node is a follower. A new leader will be elected on old leader failure for management of replications. Leader keep all servers logs sync with itself using replications.

The leader handles all the requests from users and forwards them to followers. Followers receive messages from leaders and follow instructions. Candidates perform leader elections in an attempt to become a leader. The newly elected leader will have all of the already committed log entries.

Randomized timers based leader election are used to choose new leader for a specific timer calculated term. Leader use heartbeats mechanism to communicate with followers to resolve conflicts. Election safety is assured by Raft.

Duplicate log entry at a particular index cannot be possible however If two entries in different logs have the same index and term, then they store the same command and logs are identical in all next entries. Leader replicate the entry to all the follower nodes in order to keep the logs consistent.

The leader executes the entry once it is committed and returns the result to the client. Leader maintains the index number. Leader make followers overwrite the conflicting log entries with leader’s log entries. The leader maintains a next Index for each follower. A leader never overwrites or deletes entries in its own log.

Two logs up-to-date status is defines on basis of index and term of last log entries. For different terms case, log with later term is more up-to-date however for same terms, the longer log is more up-to-date.

Two leaders cannot be elected for the same term thus keeping configuration change membership safe. Raft’s new mechanism for changing the cluster membership using overlapping majorities guarantee safety.The joint consensus allows individual servers to transition between configurations at different times without compromising safety.

**Question:**

**1 -** In which case, information must flow from followers to followers or from followers to leader?

**2 -** Is this paper applicable in future with restriction of information flow?

**3 -** Isn’t it costly to duplicate logs on all follower servers?

**4 -** How follower server failure is handled?