Infrastructure Technologies for Large-Scale Service-Oriented Systems

Kostas Magoutis

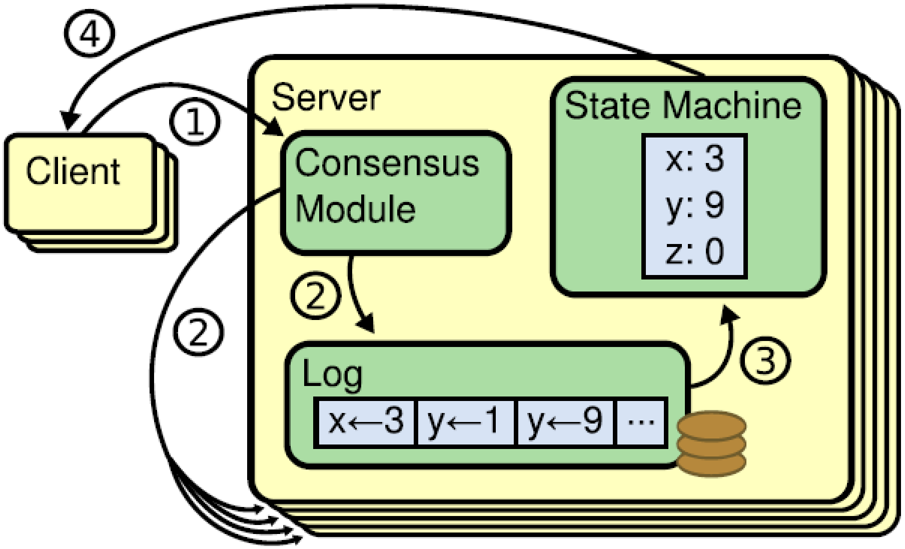
magoutis@cse.uoi.gr

http://www.cse.uoi.gr/~magoutis

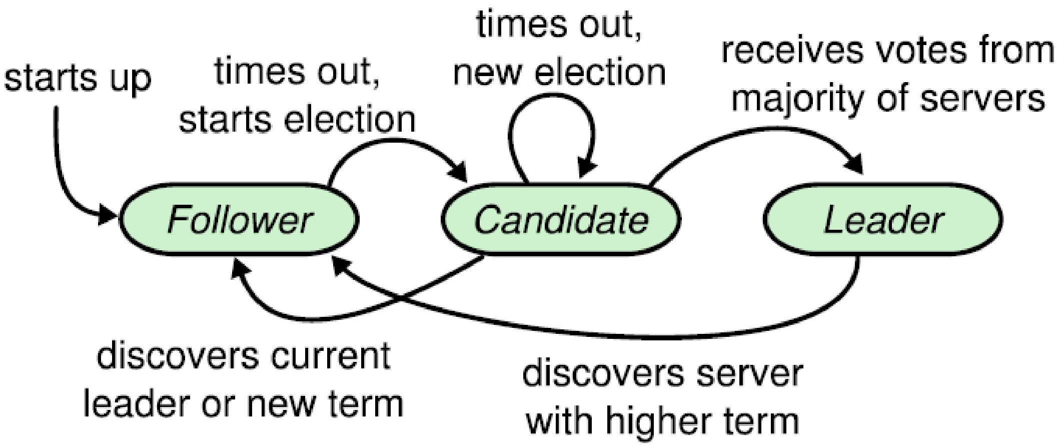
Raft

- Consensus algorithm for log replication
- Easier to understand compared to Multi-Paxos

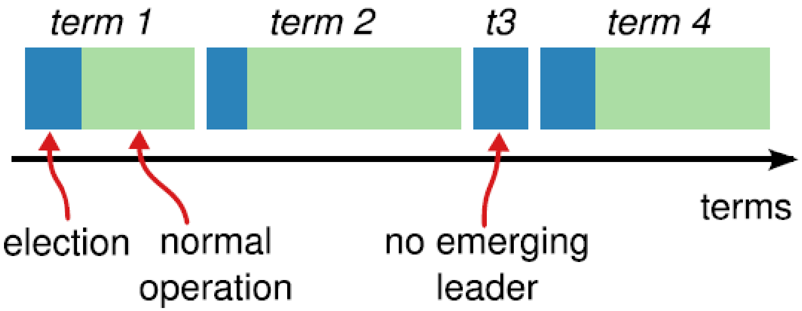
Replicated state machine architecture



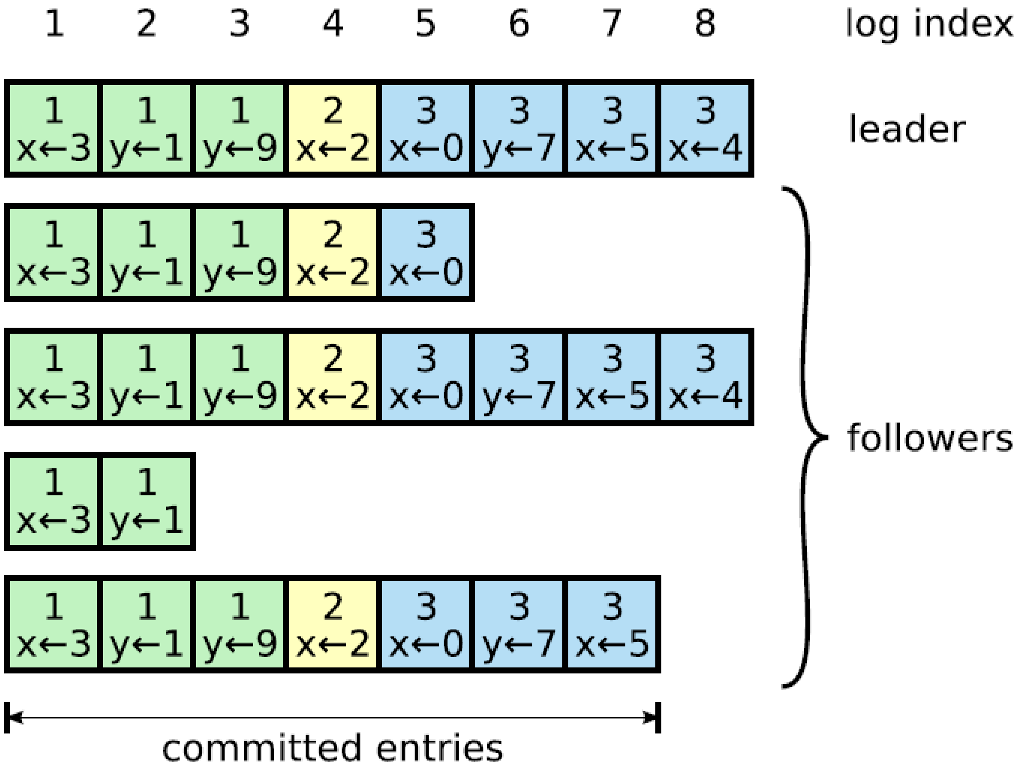
Server states


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

Terms (epochs)



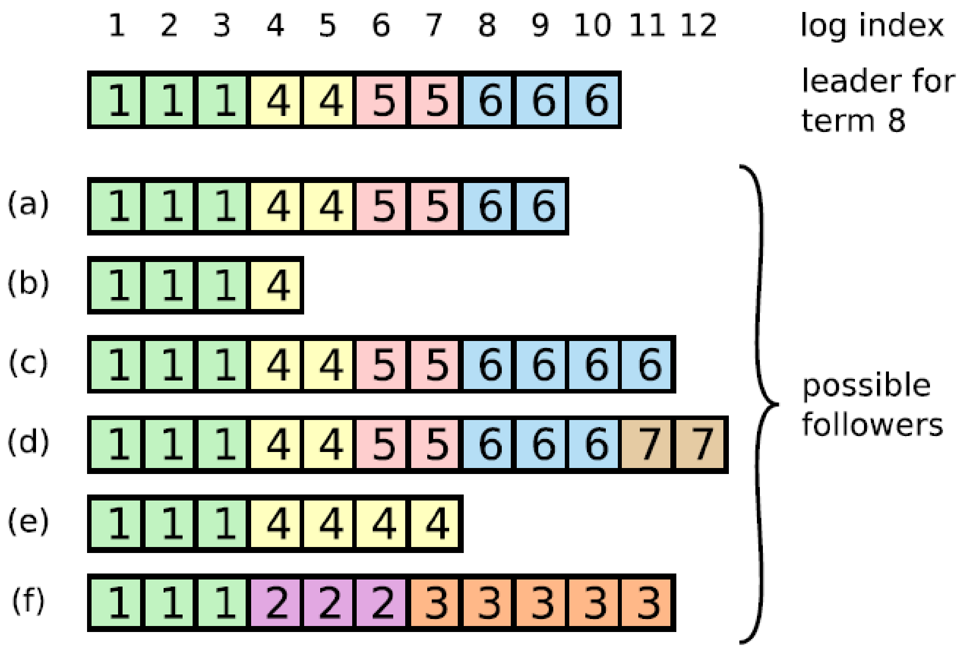
Log entries



log index	1	2	3	4	5	6	7	8
leader	1 x←3	1 y←1	1 y←9	2 x←2	3 x←0	3 y←7	3 x←5	3 x←4
followers	1 x←3	1 y←1	1 y←9	2 x←2	3 x←0			
	1 x←3	1 y←1	1 y←9	2 x←2	3 x←0	3 y←7	3 x←5	3 x←4
	1 x←3	1 y←1						
	1 x←3	1 y←1	1 y←9	2 x←2	3 x←0	3 y←7	3 x←5	

committed entries

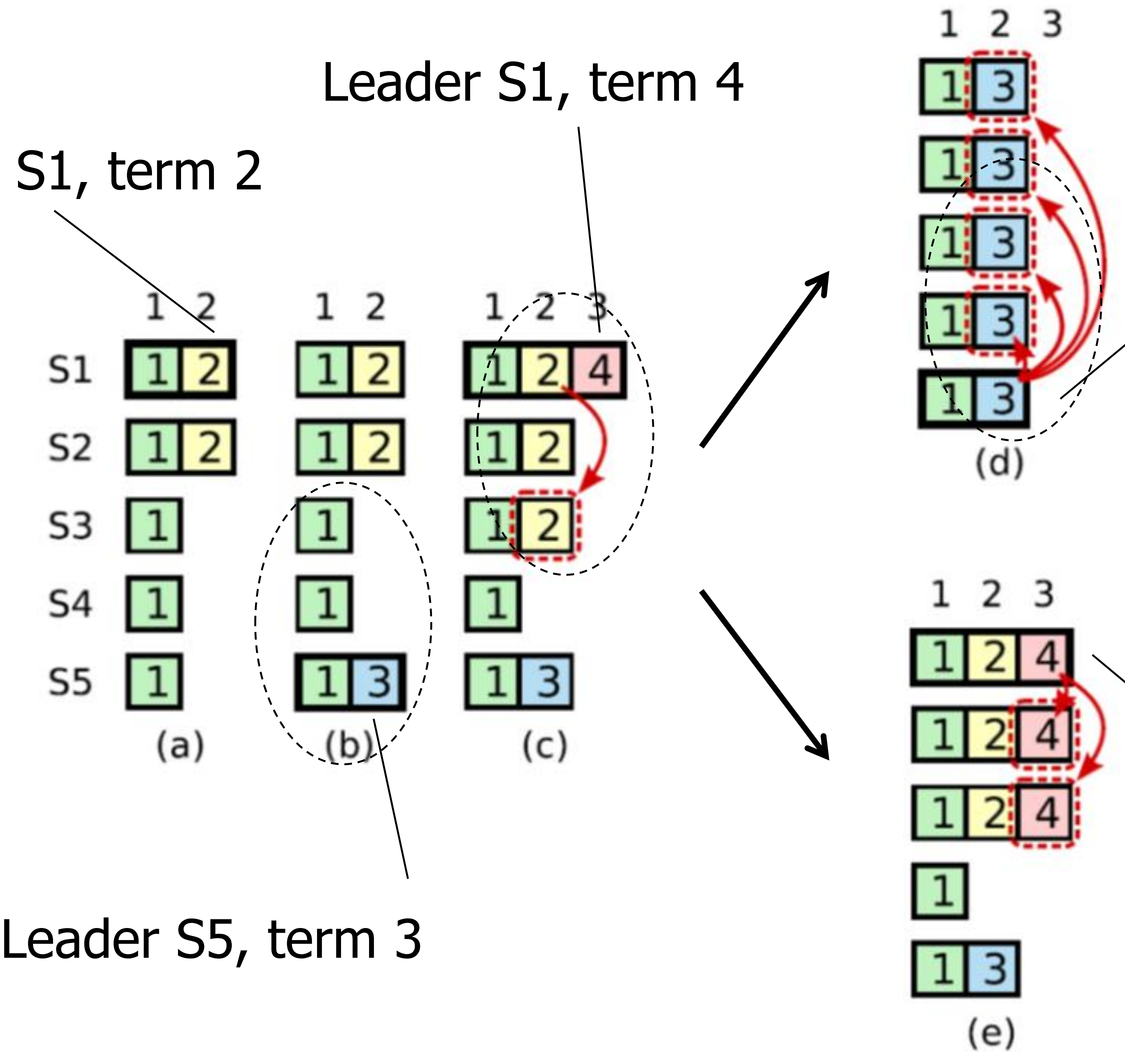
Possible states of followers



log index	1	2	3	4	5	6	7	8	9	10	11	12
leader for term 8	1	1	1	4	4	5	5	6	6	6		
(a)	1	1	1	4	4	5	5	6	6			
(b)	1	1	1	4								
(c)	1	1	1	4	4	5	5	6	6	6	6	
(d)	1	1	1	4	4	5	5	6	6	6	7	7
(e)	1	1	1	4	4	4	4					
(f)	1	1	1	2	2	2	3	3	3	3	3	

possible followers

When is an entry committed?



Leader S1, term 2

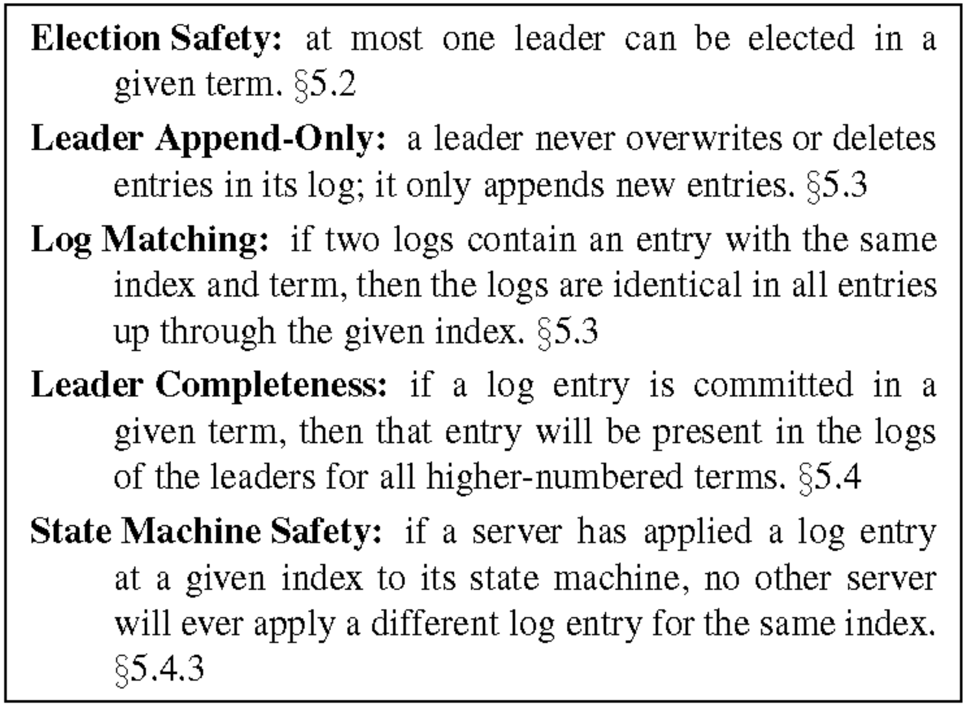
Leader S1, term 4

Leader S5, term 3

Leader S5, term 5

Leader S1, term 4

Properties



Election Safety: at most one leader can be elected in a given term. §5.2

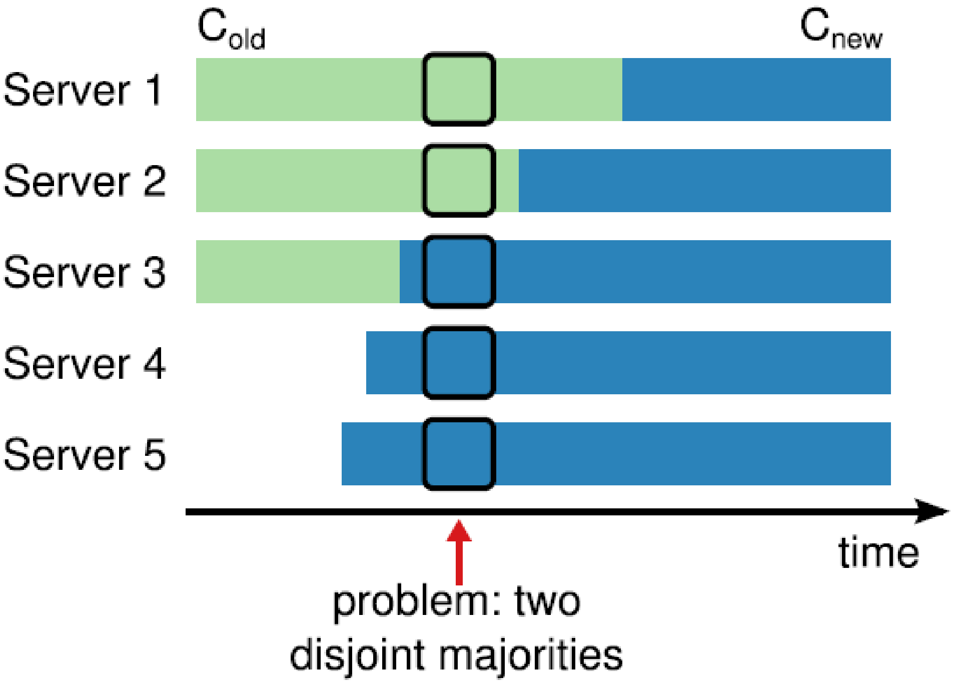
Leader Append-Only: a leader never overwrites or deletes entries in its log; it only appends new entries. §5.3

Log Matching: if two logs contain an entry with the same index and term, then the logs are identical in all entries up through the given index. §5.3

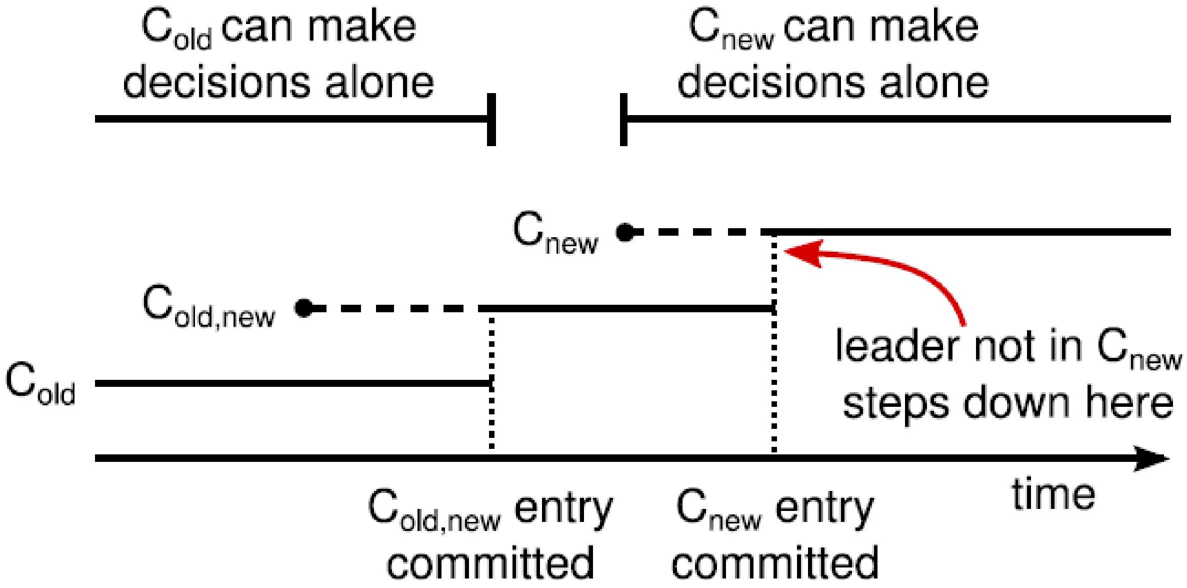
Leader Completeness: if a log entry is committed in a given term, then that entry will be present in the logs of the leaders for all higher-numbered terms. §5.4

State Machine Safety: if a server has applied a log entry at a given index to its state machine, no other server will ever apply a different log entry for the same index. §5.4.3

Reconfiguration

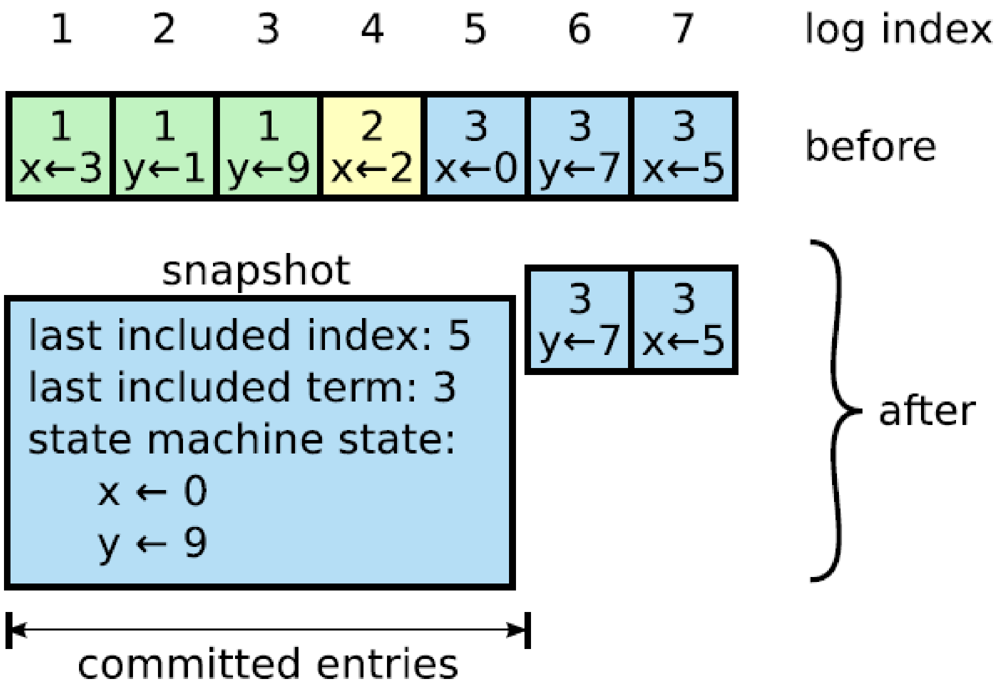


Joint consensus

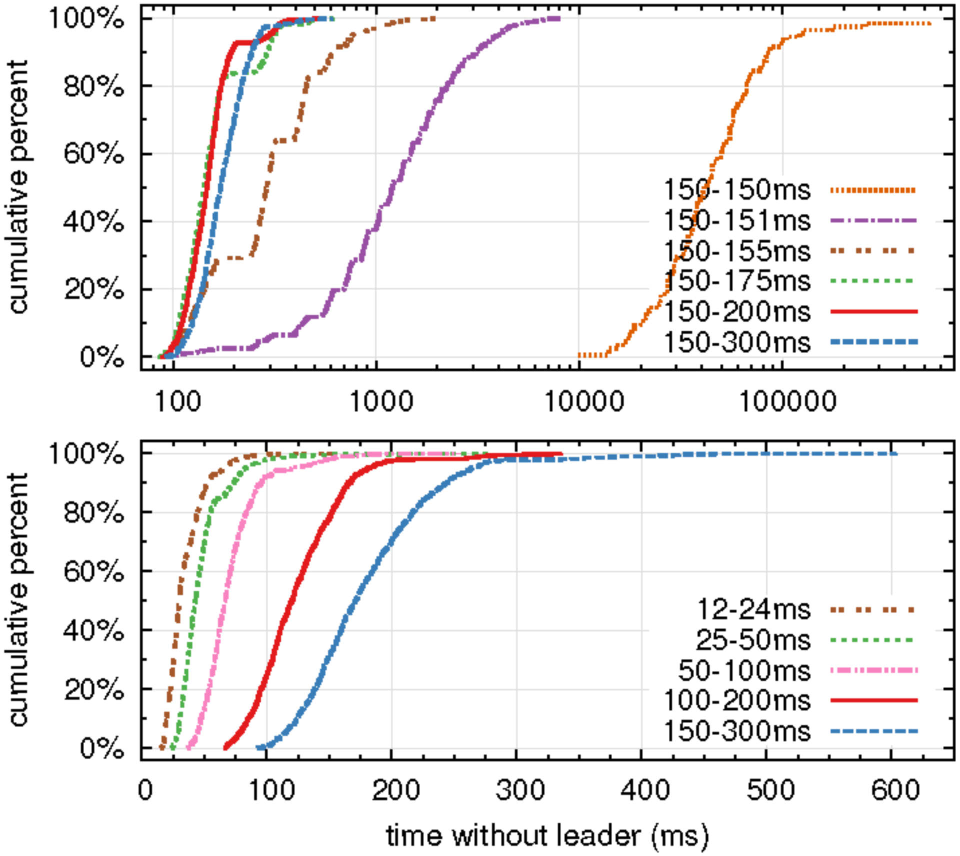


C_{old} can make decisions alone
C_{new} can make decisions alone
C_{new}
$C_{old,new}$
C_{old}
leader not in C_{new} steps down here
time
$C_{old,new}$ entry committed
C_{new} entry committed

Log compaction - snapshots



Time to detect and replace crashed leader



Timing requirement

$$broadcastTime \ll electionTimeout \ll MTBF$$

Raft