

ClientRequest RPC

Invoked by clients to modify the replicated state.

Arguments:

clientId	client invoking request (§6.3)
sequenceNum	to eliminate duplicates (§6.4)
command	request for state machine, may affect state

Results:

status	OK if state machine applied command
response	state machine output, if successful
leaderHint	address of recent leader, if known (§6.2)

Receiver implementation:

1. Reply NOT_LEADER if not leader, providing hint when available (§6.2)
2. Append command to log, replicate and commit it
3. Reply SESSION_EXPIRED if no record of clientId or if response for client's sequenceNum already discarded (§6.3)
4. If sequenceNum already processed from client, reply OK with stored response (§6.3)
5. Apply command in log order
6. Save state machine output with sequenceNum for client, discard any prior response for client (§6.3)
7. Reply OK with state machine output

Rules for Leaders

- Upon becoming leader, append *no-op* entry to log (§6.4)
- If election timeout elapses without successful round of heartbeats to majority of servers, convert to follower (§6.2)

RegisterClient RPC

Invoked by new clients to open new session, used to eliminate duplicate requests. §6.3

No arguments

Results:

status	OK if state machine registered client
clientId	unique identifier for client session
leaderHint	address of recent leader, if known

Receiver implementation:

1. Reply NOT_LEADER if not leader, providing hint when available (§6.2)
2. Append register command to log, replicate and commit it
3. Apply command in log order, allocating session for new client
4. Reply OK with unique client identifier (the log index of this register command can be used)

ClientQuery RPC

Invoked by clients to query the replicated state (read-only commands). §6.4

Arguments:

query	request for state machine, read-only
--------------	--------------------------------------

Results:

status	OK if state machine processed query
response	state machine output, if successful
leaderHint	address of recent leader, if known

Receiver implementation:

1. Reply NOT_LEADER if not leader, providing hint when available (§6.2)
2. Wait until last committed entry is from this leader's term
3. Save commitIndex as local variable readIndex (used below)
4. Send new round of heartbeats, and wait for reply from majority of servers
5. Wait for state machine to advance at least to the readIndex log entry
6. Process query
7. Reply OK with state machine output