

Programming Assignment 2: Stale Example for Read Only Commands in Paxos

Harsh Pareek(hhp266), Nazneen Rajani(nnr259)

November 11, 2013

In this document, we provide an example where the read only implementation in the van Renesse “Paxos Made Complex” paper gives stale results assuming that the system is asynchronous and the channels are not FIFO.

Consider the following execution with 2 replicas R1 and R2, 1 leader L and 3 acceptors A1–A3, in the bank application setting: The client sends 2 commands, a deposit (D) for \$10 and an inquiry (I) to both replicas. I is a read only command. The client’s balance is \$0 initially.

- R1 and R2 propose D and I to leader L. Both propose D in slot 1. I is read only and is not assigned a slot.
- L is adopted by all acceptors. It has an appropriately long lease time which will not expire during this execution.
- L proposes D for slot 1, and is accepted. L sends this final decision for slot 1 to R1 and R2 through messages m11 and m12 respectively.
- Since L is active, has a valid lease and all proposals have been decided, it instructs R1 to respond to I. This message is m21.
- m12 arrives at R2. R2 updates its state to \$10.
- Then, m21 arrives at R1 out of order. R1 responds to the client giving a balance as \$0. This value is stale as R2 has already overwritten this value in its state.
- Finally, m11 arrives at R1 and R1 updates its balance to \$10.

Thus, R1 returns stale state to the client. Van Renesse notes that the leader should wait until all proposals are decided before sending, but if channels are not FIFO, this is not enough, as shown above. If instead the *replica* waits until all proposals are decided, stale data will not be returned, even if channels are not FIFO, though this would be inefficient.