

Question 1:

Issues of Basic Paxos

- A process learns the the value thats proposed and chosen by majority of acceptors for maintaining liveness
- If there are mutiple proposals with higher proposal number, acceptance might fail as might cause liveness violation.
- Consider the case when P1 sends 'prepare 1' to P,Q
- Both P,Q respond P1 with a promise to not to accept any request numbered smaller than 1.
- P2 sends 'prepare 200' to P,Q.
- Both P,Q respond P1 with a promise to not to accept any request numbered smaller than 1.
- Now P1 sends acceptor request to P, Q. It gets rejected as they already promised P2 that they don't accept requests greater than 200.
- Hence P1 now adjusts its proposal value to a number higher than 200, such that P,Q will accept P1 proposer request, discarding P2's old one.
- If the channel is lossy, there might be message loss, and a value cant be learnt.

Solution for above Issues:

- If the channel is lossy, and the message have been lost, the learn then get the accepted value from the acceptors.
- However if the acceptor fails then a new set of proposal requests are to be generated with the current acceptors.
- We can follow the Election algorithm, to have one process as the proposer that will generate the proposals higher than the current accepted value.
- In such cases, we can avoid liveness issues, as its the only process that does proposals with higher numbers.

Question 2

Multipaxos:

- The number of accepted messages increases with execution.
- This highly effects the performance of the algorithm

Solutions:

The solution is to not keep all triples received in 2a messages, or those that have the maximum ballot when the triple was received in a 2a message, as in the original pseudocode but keep only triples with the maximum ballot for each slot, so there is at most one triple for each slot.

(Referred from Moderately Complex Paxos Made Simple: High-Level Specification of Distributed Algorithms)

Question 3:

Preemption: `python3 -m da origPreempt.da`

- Whenever a proposer sends a proposal with a lower proposal number than the previously accepted one, then the acceptor sends a reject message to the proposer indicating that it has accepted the request of higher number.
- The proposer when receives the preempt message then tries adjusts its proposal number to the highest value.

Files: `origPreempt.da`

Correctness Verification: `python3 -m da correctness.da`

Files used: `controller.da` , `origPerformance.da`,`correctness.da`

- A test process `correctness.da` is created that will check Agreement violation, Validity violation etc.

Lossy Channel:

- A lossy channel is created by using random value generated. Features like message delay, wait time are added .
- **Files used: `controller.da` , `origPerformance.da`,`correctness.da`**

