TUM I 13
Chair of Application and Middleware Systems
Prof. Dr. H.- A. Jacobsen

Tutorial
**Distributed Systems (IN2259)**
WS 2020/21

# EXERCISES ON PAXOS & REPLICATED STATE MACHINES

## EXERCISE 1 Basic Paxos

(a) Consider the following system assumptions for running the Paxos algorithm: (1) Each message takes **exactly** 0.5 sec to reach its destination and (2) the processing time at the processes can be ignored (i.e, is zero). In this system, we assume that there are three acceptors $(A, B, C)$, two proposers $(D, E)$ and one learner $(F)$. Proposer $D$ wishes to propose value `"write"` with proposal number 1 as its proposal, and it sends its prepare message at time 0. 0.9 sec later, proposer $E$, who wants to make its proposal with value `"read"` and proposal number 2, sends its prepare message. During the whole execution, no message gets lost. Draw all messages exchanged during the execution of the Paxos algorithm for this exercise in Figure 1.1, and also specify which value will finally be chosen.

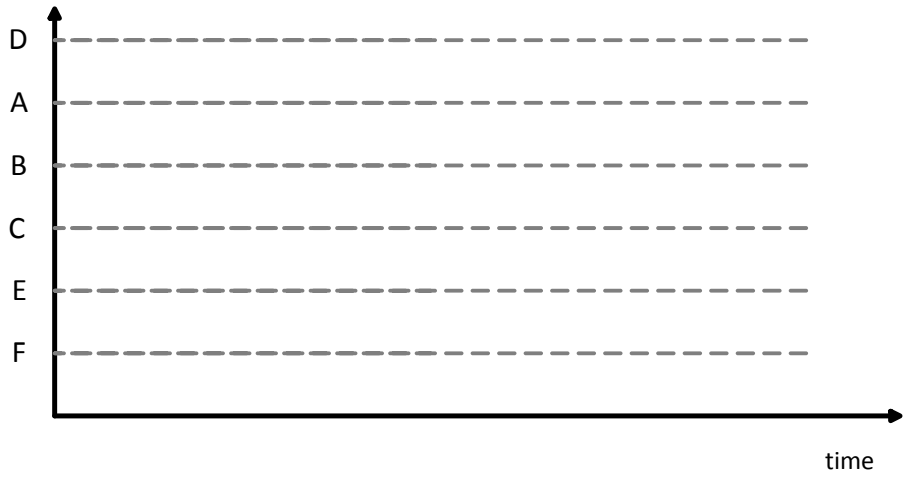


Figure 1.1: Message diagram

(b) Construct a scenario in which the Paxos algorithm does not terminate.

## EXERCISE 2 RSM Using Paxos

(a) Write a program (in pseudocode) for implementing an RSM using the Paxos algorithm.

(b) Assume that at fictitious time $t$, two processes think that they are the distinguished process (i.e., proposer & learner). Discuss what happens to the system in terms of the safety and liveness requirements.

(c) Explain what could be done if there are gaps in command execution (cf. Figure 2.2). I.e., a particular instance among the Paxos instances that are realizing the RSM failed to decide for a value (command).

## EXERCISE 3 Byzantine Paxos

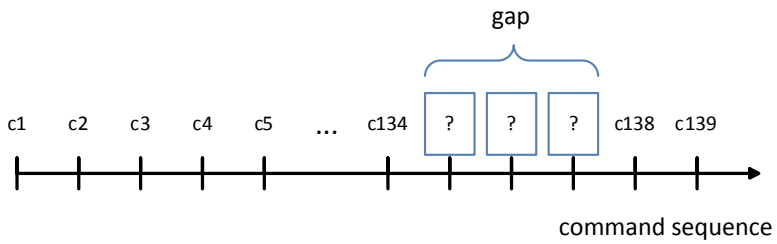


Figure 2.2: Gaps in command sequence.

Suppose that we have three proposers, five acceptors and one learner that run the Paxos algorithm. We assume that proposers and the learner never fail, but two out of five acceptors are suffering from arbitrary (byzantine) failure (i.e., they can lie and send wrong messages to others, and generally speaking, they do not follow their own specification). Using a fictitious scenario, show that this Paxos algorithm is no longer satisfying the non-triviality safety requirements of the Paxos specification.