Homework 3

CS425/ECE428 Spring 2023

Due: Monday, March 27 at 11:59 p.m.

1. **RAFT leader election**

   Consider a system of 5 processes $\{P1, P2, P3, P4, P5\}$ using Raft's algorithm for leader election. Suppose P1, the leader for term 1, fails and its four followers receive its last heartbeat at exactly the same time. Answer the following questions assuming that the election timeout is chosen uniformly at random from the range [100,500] ms (unless otherwise specified), no processing delay exists, and the one-way delay for all messages between two processes are as shown in Figure 1. The processes communicate with one-another only through their direct channels (not via other processes).

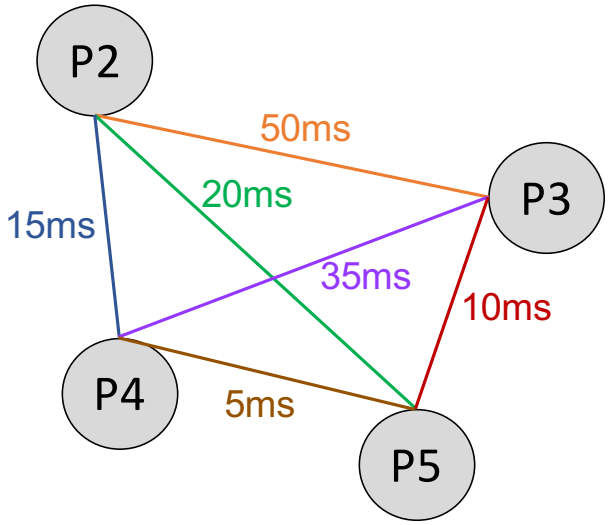


Figure 1: Figure for question 2

Suppose P2 sets its election timeout to 170 ms and calls for an election for term 2. Assume P4 and P5 have their timeout values set to more than 400 ms. What range of timeout values for P3 (within [100,500] ms) will certainly result in:

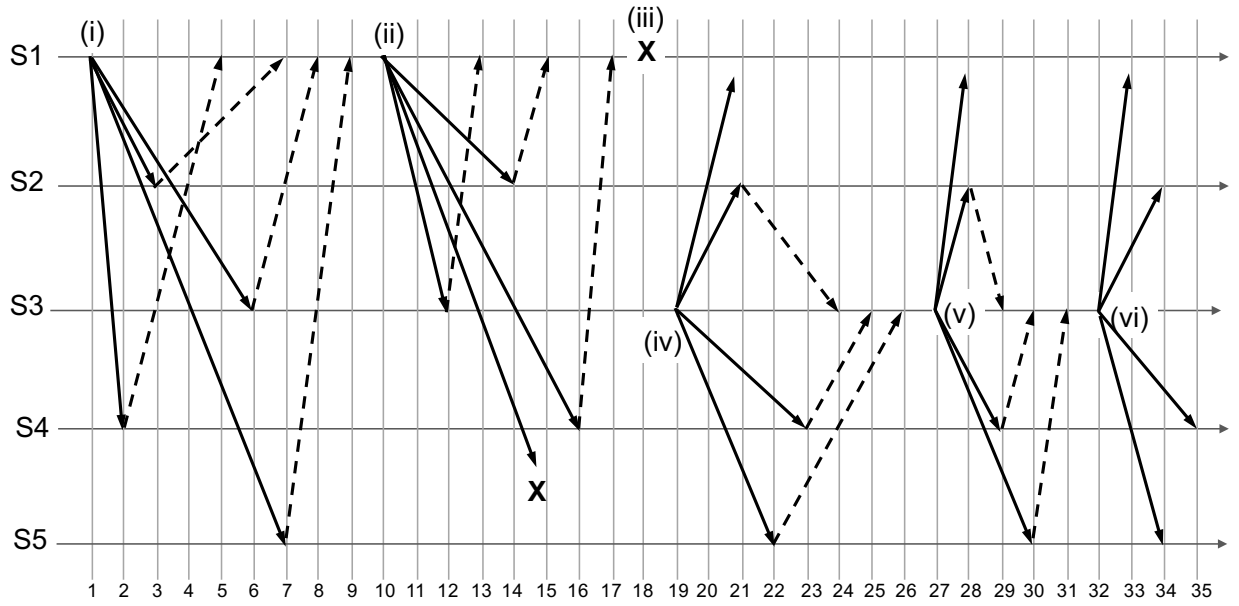
(a) (2 points) P2 winning the election?

(b) (2 points) P3 winning the election?

(c) (2 points) split vote?

2. **RAFT timeline**

Consider the timeline below.



It demonstrates a series of events / message exchanges in a Raft cluster. The numbers at the bottom show the real time. Initially all servers start in follower state in term 0. The following key events occur as indicated in the figure:

(i) Server 1's election timeout expires, it sends RequestVote messages for term 1.

(ii) Server 1 receives log entry P from a client and sends AppendEntries messages, including P. Its message to Server 5 is dropped (i.e., never received).

(iii) Server 1 crashes; it remains crashed for the remainder of the execution.

(iv) Server 3's election timeout expires, it sends RequestVote messages for term 2.

(v) Server 3 receives log entry Q from a client and sends AppendEntries messages to nodes, including Q.

(vi) Server 3 sends the next AppendEntries messages to nodes.

List the time that each of the following happen, using the timeline on the diagram (i.e., 1–35), or write "never" if the event never occurs.

(a) (1 point) Server 1 transitions to Leader state.

(b) (1 point) Server 4 updates its current term to 1.

(c) (1 point) Server 1 *commits* event P in its log.

(d) (1 point) Server 2 *appends* event P to its log.

(e) (1 point) Server 4 updates its current term to 2.

(f) (1 point) Server 3 transitions to Leader state.

(g) (1 point) Server 4 *commits* event Q in its log.

(h) (1 point) Server 4 *commits* event P in its log.

(i) (1 point) Server 5 updates its current term to 2.

For the next parts, state the *earliest* time after which the following *must* be true, even if execution after this time proceeded differently than what is shown in the diagram. Assume that once server 1 has crashed, it can never recover.

(j) (1 point) Server 5 *cannot* be elected as leader in term 1.

(k) (1 point) Server 2 *cannot* be elected as leader in term 2.

(l) (1 point) Event P will eventually be committed by at least one server.

(m) (1 point) Event Q will eventually be committed by at least one server.

3. **RAFT Log Consensus**

Consider a system of three servers $\{S_1, S_2, S_3\}$ wanting to achieve log consensus using the Raft algorithm. For each sub-part below, state whether the shown snapshot of log entries at each server could arise from a valid run of the Raft algorithm. If yes, construct a scenario that would lead to these log entries in Raft's execution. If not, explain what makes the entries invalid.

Each number in the shown log entries represents the Raft term that the corresponding event is associated with.

For the valid log entries, the scenario you construct should include, for each term: which server gets elected as the leader, which servers vote for it, and which log entries does it append / replicate at each server.

(a) (3 points)

S_1: 1, 1, 1
S_2: 1, 1, 2, 2, 2
S_3: 1, 1

(b) (3 points)

S_1: 1, 1, 1
S_2: 1, 1, 2, 2, 2
S_3: 1, 1, 3

(c) (3 points)

S_1: 1, 1, 1
S_2: 1, 2, 2, 2
S_3: 1, 1, 1, 3

(d) (3 points)

S_1: 1, 1, 1, 3
S_2: 1, 2, 2, 2
S_3: 1, 1, 1, 3

(e) (3 points)

S_1: 1, 1, 1, 1
S_2: 1, 1, 2, 2
S_3: 1, 1, 2, 3

4. **Blockchains**

In a system using a blockchain for distributed consensus, in order to add a block to a chain, a participating node must solve the following puzzle: it must find a value x such that its hash, $H(x||seed)$, is less than T. The hash function is such that a given value of x can uniformly map to any integer in $[0, 2^{256} - 1]$. Assume T is set to 2^{220}.

(a) (2 points) What is the probability that a given value of x, randomly chosen by the participating node, is a winning solution to the puzzle (i.e. $H(x||seed) < T$)?

(b) (2 points) Assume a participating node adopts the standard strategy for solving the puzzle: it randomly picks a value x and checks if it is the winning solution. It keeps repeating this step, until a winning solution is found. Further assume that, for simplicity, the strategy is memoryless (unoptimized), in the sense that a value of x that has already been checked can get re-checked if it is randomly selected again. If the node can hash and check 2^{10} values per second, what is the probability of finding a winning solution within 5 hours? (You may round your answer to five decimal places.)

(c) (2 points) Assume there are 10000 participating nodes in the system, and that each node starts solving the puzzle at exactly the same time. Assuming the same rate of computing hashes at each node (i.e. 2^{10} values per second), what is the probability that at least one node in the system finds a winning solution in 5 hours? (You may round your answer to four decimal places.)