Homework 4

CS425/ECE428 Spring 2022

Due: Wednesday, March 30th at 11:59 p.m.

1. 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). Each question below is independent of others.

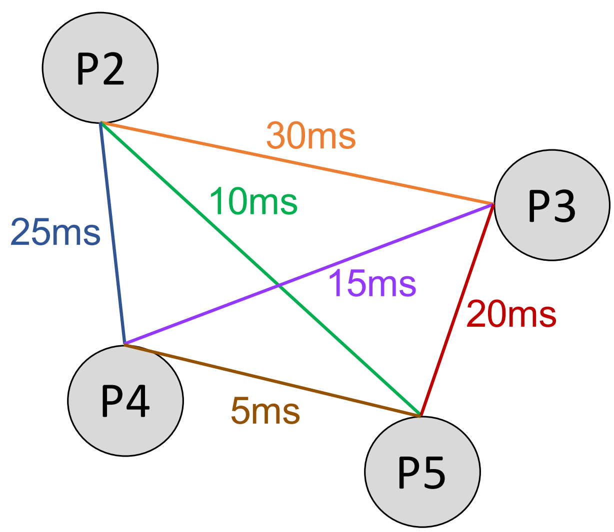

Figure 1: Figure for question 1

(a) (2 points) Suppose P2 and P3 both set their election timeout to 150 ms and call an election for term 2. Assume P4 and P5 have their timeout values set to more than 400 ms. Which candidate (P2 or P3) will each of the four alive processes vote for? Will a leader be elected for term 2? If yes, which process?

(b) (2 points) Suppose P2 sets its election timeout to 150 ms and calls an election for term 2, and P3 sets its timeout to 170 ms and also calls an election for term 2. Assume P4 and P5 have their timeout values set to more than 400 ms. Which candidate (P2 or P3) will each of the four alive processes vote for? Will a leader be elected for term 2? If yes, which process?

(c) (6 points) Suppose P2 sets its election timeout to 150 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:

(i) P2 winning the election?

(ii) P3 winning the election?

(iii) split vote?

(d) (5 points) Suppose P2 sets its election timeout to 105 ms and calls for an election for term 2. What is the probability that another process (among P3, P4 and P5) also calls an election for term 2? Round your response upto 4 decimal places. [*Hint:* this probability can be computed as (1 - *(probability that neither P3 nor P4 nor P5 call for an election for term 2)*)]

2. 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, 2, 2
   S_3: 1, 1

   (b) (3 points)
   S_1: 1, 1, 1
   S_2: 1, 1, 2
   S_3: 1, 1

   (c) (3 points)
   S_1: 1, 1, 1
   S_2: 1, 1, 2
   S_3: 1, 1, 3

   (d) (3 points)
   S_1: 1, 1, 1
   S_2: 1, 2, 2
   S_3: 1, 2, 3

   (e) (3 points)
   S_1: 1, 1, 3, 3
   S_2: 1, 2, 2
   S_3: 1, 1, 3

3. 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^{226}.

   (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) (4 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 2^5 values per second, what is the probability of finding a winning solution within 10 hours? (You may round your answer to five decimal places.)

   (c) (4 points) Assume there are 5000 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. 2^5 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.)