

ECE428 Homework 1

Due: 11:59 p.m. on Monday March 6th, 2023

This assignment has 5 questions, and with 60 points in total. The solutions must be typed, and submitted via Gitlab. However, the diagrams can be hand-drawn. You must acknowledge any sources used to arrive at your solutions, other than the course materials and textbook. All homework assignments are expected to be an individual work, so no collaborations are allowed.

Question 1: Ring Around the Rosy [12 points]

Consider a ring-based failure detection scheme with processes p_1, \dots, p_n . The process p_i sends a heartbeat every T seconds to process p_{i+1} , and the process p_n sends a heartbeat to p_1 .

- (6) (a) This failure detection mechanism is not complete, since it does not tolerate failure of two neighboring processes. How would you modify it so that a failure of any two processes is always detected? Be explicit in your answer and compare the bandwidth cost of your proposed scheme to the original ring-based failure detection scheme. It is required that the proposed scheme should maintain a ring structure for the processes.
- (4) (b) Assume now that p_i only sends a heartbeat to p_{i+1} after receiving a heartbeat from p_{i-1} , and then waits T seconds. Specifically:
1. p_1 sends a heartbeat to p_2 ;
 2. p_2 receives the heartbeat, waits T seconds, and then sends a heartbeat to p_3 ;
 3. p_3 receives the heartbeat, waits T seconds, and sends a heartbeat to p_4 ;
 4. ... and so on.

Provided that there is exactly one heartbeat (or token) circulating in the ring at any time, what should the failure timeout be set to, given a maximum one-way delay of Δ , to ensure the accurate failure detection? What would be the maximum detection time?

- (2) (c) The token scheme above ensures that a failure of one process is detected by every other process, but it does not directly reveal which process has failed. Sketch a modification of the failure detection algorithm, so that identity of the failed process can be discovered. The modification should preserve the token-based design.

Question 2: Xtreme Drift [6 points]

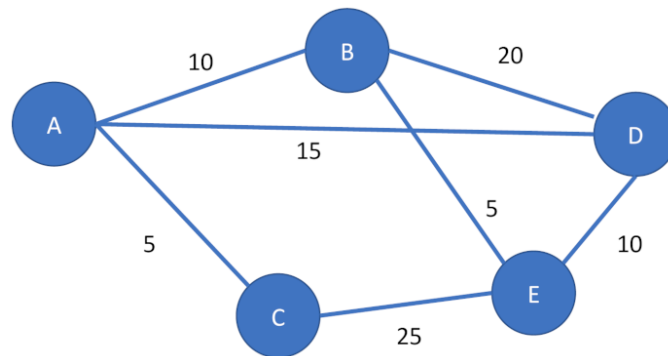
Usually the clock drift is small enough, so it only occasionally needs to be corrected for. This question considers the problem with larger drifts.

- (2) (a) Consider a heartbeat protocol in a system with the maximum one-way delay Δ and a period of T . Assume that the clock drift is bounded by 1% (i.e., the clocks lose or gain at most 1s every 100s relative to each other). What should be the value of a timeout to ensure the accurate failure detection?
- (2) (b) What would be the maximum detection time of a failure?

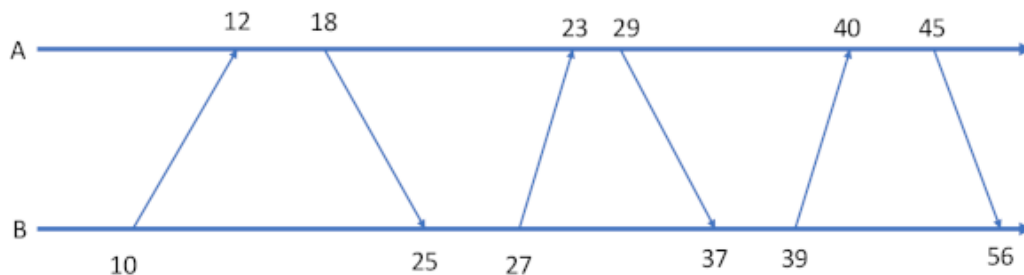
- (2) (c) Client A uses Cristian's algorithm to synchronize with server B. If the RTT between A and B is 20 ms, how often should Cristian's algorithm be re-run to keep the clocks synchronized within 100 ms, if the maximum drift rate is again 1%?

Question 3: You Say Go Slow, I Fall Behind [18 points]

- (6) (a) Consider the following diagram that lists five servers (A, B, C, D, E), and the RTT for each link between them. Explain which servers should synchronize with which other servers in minimize the worst-case skew between any two servers. What will be the worst-case skew in this case?



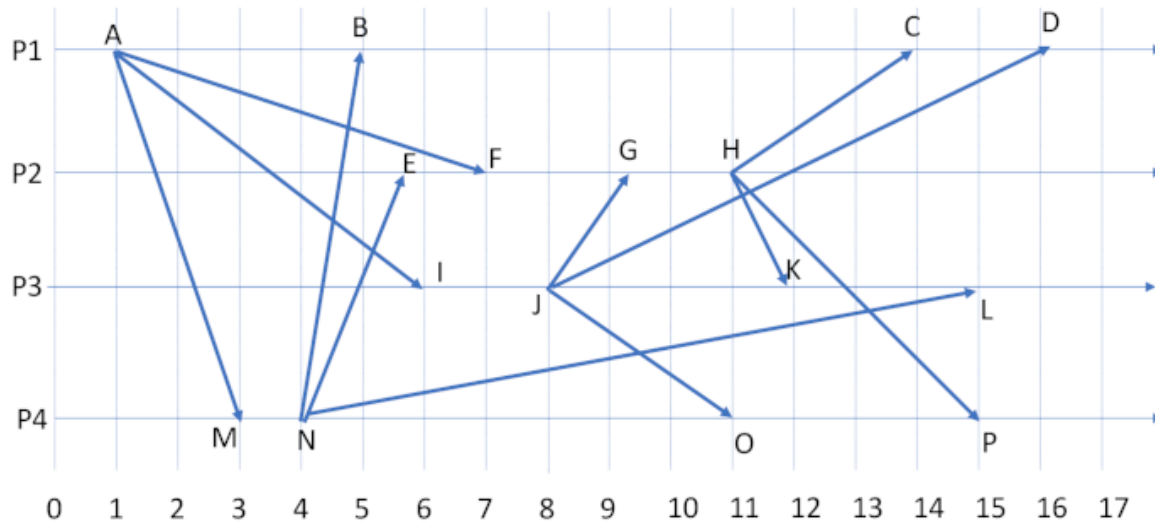
- (6) (b) Consider the following diagram that lists transmission and reception time of NTP messages between servers A and B in each server's clock. Assuming there is no clocks drift, what is an upper bound on the one-way transmission time of each of the six messages shown? *Hint: consider an entire diagram, not just adjacent message pairs*



- (6) (c) Based on your answer in part (a), what is the maximum skew of server B with respect to server A? What is the minimum skew? By how much the time of server B should be corrected and in which direction?

Question 4: Teach Me How To Be Sensible and Keep The Timestamps Logical [18 points]

The following figure shows timeline of 16 events (A to P) for four processes. The numbers along the bottom horizontal axis indicate the real time instances.



Answer the following questions.

- (6) (a) Write down the Lamport timestamp of each event.
- (6) (b) Write down the vector timestamp of each event.
- (6) (c) List all events concurrent with the event (i) C, (ii) I, and (iii) O, respectively.

Question 5: Sna-sna-snapshots everybody! [6 points]

Consider again the timeline of events shown in the figure above in Question 4. Assume that P3 initiates the Chandy-Lamport snapshot algorithm at time instance 9. Assuming the FIFO channels, write down all possible consistent cuts that the resulting snapshot can capture. The cuts can be described by their corresponding frontier events.