Homework 1

CS425/ECE428 Spring 2024

Due: Monday, Feb 19 at 11:59 p.m.

1. Process p uses a ping-ack failure detector to detect process q's failure, sending pings every 25s. The system is synchronous with minimum and maximum message communication delays as shown in the figure. If q fails 12s after sending an ack, how long after that will p detect the failure? [2 points]

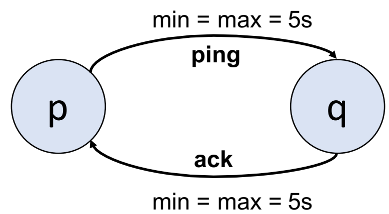

Figure 1: Figure for question 1.

2. Consider Figure 2. The client has an option of using any of the three authoritative sources of real time (s1, s2, or s3) for external synchronization via Cristian algorithm. The round-trip times (RTT) between the client and the three servers are shown in the figure. Assume that the observed RTT to each server remains constant across all synchronization attempts.

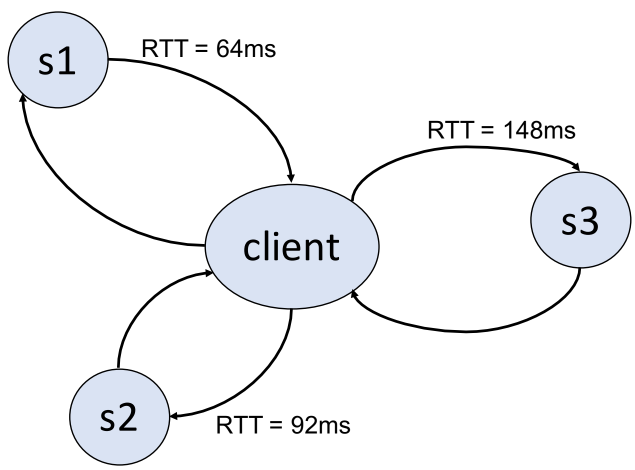

Figure 2: Figure for question 2.

   (a) (1 point) Which server should the client choose to achieve the lowest accuracy bound right after synchronization? What is the value of this bound, as estimated by the client right after synchronization?

   (b) (3 points) If the client's local clock drifts at the rate of 25μs every second, what is the longest time period (in seconds) at which the client must initiate synchronization with the server it chose in part (i), so as to maintain an accuracy bound within 200ms at all times.

3. The timeline in Figure 3 shows 16 events (A to P) across four processes. The numbers below indicate real time.

   (a) (3 points) Write down the Lamport timestamp of each event.

   (b) (5 points) Write down the vector timestamp of each event.

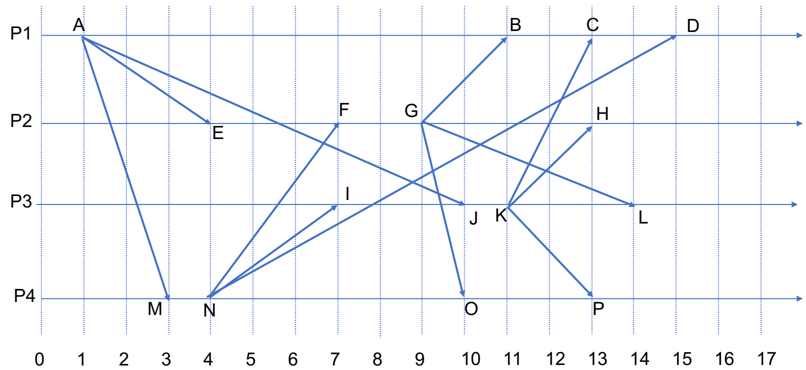

Figure 3: Timeline for questions 3 and 4.

(c) (3 points) List all events considered concurrent with (i) B, and (ii) J.

4. (a) (5 points) Consider the timeline and events in Figure 3 again. Suppose that P2 initiates the Chandy-Lamport snapshot algorithm at (real) time 8. Assuming FIFO channels, write down *all* possible consistent cuts that the resulting snapshot could capture. You can describe each cut by its frontier events.

   (b) (3 points) Write *all* possible states of the incoming channels at P1 that the above snapshot could record. You can denote each message by its send and receive event ids.

5. Consider a modification to the Chandy-Lamport algorithm where it is desirable to find a consistent cut, and only record the local state at each process for the cut, and not the channel state. In this modification, the initiator process records its state and sends a marker to all other processes. Each process records its state upon receiving the marker from the initiator process. The algorithm terminates when all processes have received a marker from the initiator process. Will this modification result in a consistent cut? If yes, prove why. If not, present a counter-example. Assume all requirements for Chandy-Lamport algorithm hold (i.e. each channel follows FIFO order, no messages are dropped, and no process fails). [2 points]

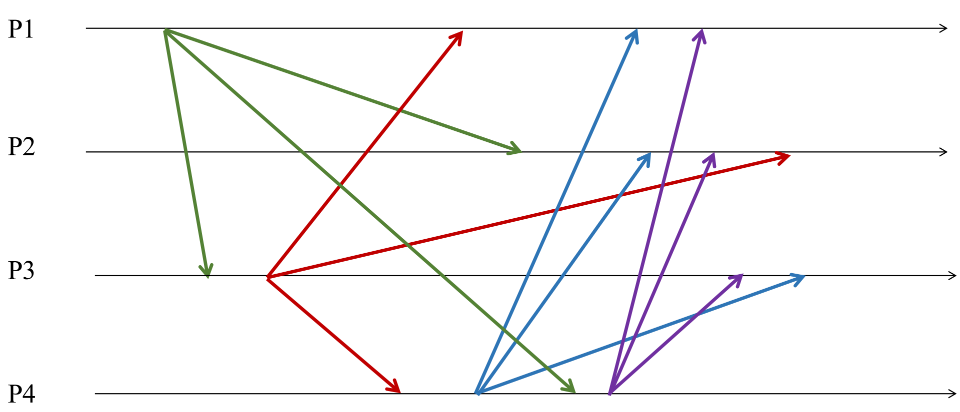

Figure 4: Figure for question 6

6. In the execution in Figure 4, processes send messages to each other to implement *causal multicast*. To simplify the picture, messages sent by each process to itself are not shown but assume that such messages

are received and delivered instantaneously. For the questions below, you may use printed or hand-drawn figure with hand-drawn responses, or digitally edit the figure from the homework PDF.

(6 points) Identify the messages that are buffered at the processes to ensure causally-ordered multicast delivery (Circle the receive event for the buffered messages to identify those messages.) For each message buffered as above, determine the earliest instant of time at which the message may be delivered, while ensuring causally-ordered multicast. (To identify the instant of time draw an arrow that begins at the time when the message is received to the time at which the message may be delivered.)

7. For each of the statements below, identify whether it is *true* or *false*. If it is false, present a counter-example. If it is true, prove why.
   (a) (2 points) A total ordered multicast is also causal ordered.
   (b) (2 points) A causal ordered multicast is also FIFO ordered.
   (c) (3 points) We can implement the ISIS algorithm for total ordering on top of (or using) causal-ordered multicast, to achieve a total causal multicast.