CS60002: Distributed Systema

Department of Computer Science & Engineering

Indian Institute of Technology, Kharagpur

Midsem Examination, Spring 2016

Total Marks: 100 **Time: 2 Hours**

INSTRUCTIONS: Please read carefully before starting

1. *Answer ALL Questions.*
2. *Unless otherwise mentioned, assume that systems are asynchronous with unique node ids and reliable communication.*
3. *Write clearly at the beginning of your answer any additional assumptions that you make which are not given in the question, though making unnecessary assumptions that simplifies a problem may incur a penalty.*

---

1. **(a)** Define Lamport's Happened Before relation. Explain how Lamport's logical clock works. (6)

(b) In an implementation of Birman-Schiper-Stephenson's protocol for causal broadcast, checking for the second condition got left out by mistake. Show clearly with an example why the protocol will not work correctly in this system. (6)

(c) Define a consistent and a strongly consistent global state. (6)

(d) Suppose that two processes i and j take five snapshots each of their local states independently at arbitrary times, with no communication for snapshot capture between them (i.e., no explicit snapshot capturing algorithm is run, processes just record their states whenever they want). Channel states are implicitly captured in the process states using message logs. Is it possible to have a scenario in which no pair of snapshots, one taken from i and one taken from j (so there are $5 \times 5 = 25$ such pairs) is consistent? Justify your answer clearly with a space-time diagram (no marks will be given without a space-time diagram). Note that the application whose snapshot is being taken sends messages as needed, it is only the snapshot capture that does not send any messages. (5)

(e) Consider a system of 2 nodes running Berkeley's algorithm with a timeserver (a separate third node). The nodes start at the same time with a perfectly synchronized clock. The clocks in the 2 nodes have drift 10^{-5}, the maximum message transmission delay of each link is 100 milliseconds, and the clocks are resynchronized every 10 seconds. The timeserver collects the clock values from each node, computes the average of the values, and sends it back to the two nodes. The timeserver estimates each-way delay as half of the round trip delay. The nodes, on getting the time from the timeserver, blindly set their clocks to the received value. What is the maximum possible skew between the two clocks at the beginning of an arbitrary resynchronization period? Show all calculations. (10)

(f) State the conditions for forwarding a probe (i, j, k) by a controller k in Chandy-Mishra-Haas algorithm for detecting a resource deadlock. Argue why these conditions prevent the detection of false deadlocks. (3 + 4)

2. (a) Consider Ricart-Agarawala's mutual exclusion algorithm being run on a synchronous system with maximum message delay T for any link. Processing time at each node is negligible. The clocks are not synchronized but you can assume that they have zero drift. Is it possible to improve the message complexity (no. of messages per CS execution) of the algorithm by modifying it in some way? Justify your answer clearly. (10)

(c) Consider a leader election algorithm for a synchronous unidirectional ring in which nodes send their ids around the ring similar to Lelann-Chang-Roberts algorithm. However, for a node with id u, the message initiated by u travels 1 hop every 2^u rounds. Write the complete algorithm so that it elects a leader at the end. What is its message and time complexity (give arguments/calculations to show how you got the values, no marks will be given without that)? (6 + 6 + 3)

3. (a) Consider an arbitrarily connected network of machines that are expected to service requests from outside users continuously. Requests for service from users can come at any machine and each machine usually receives requests very regularly, and almost at the same rate as any other machine. All of these requests require access to a shared database. Design a token based algorithm to enforce starvation-free mutual exclusion to the shared resource in this system. You should try to make your algorithm efficient. (15)

(b) Suppose you are given a distributed algorithm to find an unrooted spanning tree (so there is no parent or child pointers, every node only knows which of its edges belong to the spanning tree) of an arbitrary connected network. How can you use this algorithm to elect a single leader node in an arbitrary connected network? At the end of your algorithm, only one process will declare itself the leader (other processes may or may not know who the final leader is). What is the message complexity of your algorithm? Note that you do NOT have to design the algorithm to find the unrooted spanning tree, it is given to you as a black box (so you cannot change it either). You can directly use it to design your leader election algorithm. Write the algorithm clearly. You should try to make your algorithm efficient. (20)