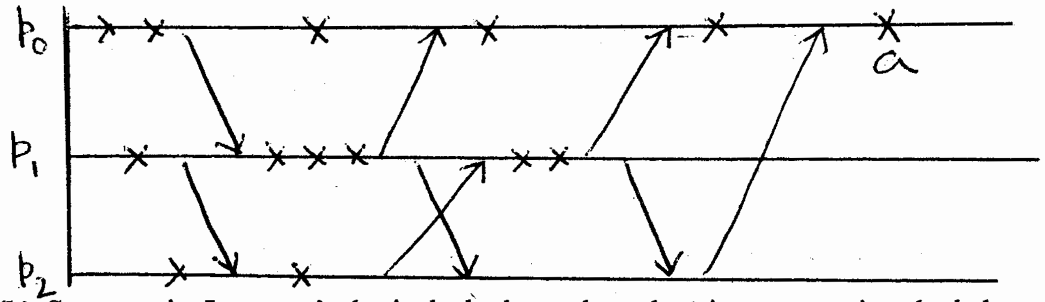
CS60002: Distributed Systema
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur
Midsem Examination, Spring 2011

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

INSTRUCTIONS: Please read carefully before starting

1. *Answer ALL Questions. Answers should be brief and to the point.*
2. *All communications can be assumed to be reliable unless otherwise mentioned.*
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 will incur a penalty.*
4. *To describe any algorithm, first list the state at each node and all types of messages that you use and their content, then list the initialization actions, and finally the send-receive rules for each message type.*

1. **(a)** With respect to the space-time diagram given below (x's indicate local computation events), what would be the Lamport's logical timestamp and vector clock timestamp of the event *a*? (6)



(b) Suppose in Lamport's logical clock, each node *i* increments its clock by a fixed positive constant $d_i > 0$ on an event (instead of incrementing by 1)? Will the clocks work correctly if the d_i's are different for different *i*? Justify your answer. (6)

(c) Argue that in case of Raymond's algorithm, the number of messages per critical section invocation under heavy load is approximately 4. (6)

(d) Consider two clocks, one with drift rate δ_1 and the other with drift rate δ_2. If the clocks are resynchronized every ρ seconds, and they can be synchronized to within θ of each other (given the errors in the synchronization algorithm), find an expression for the maximum skew of the clocks at the beginning of each resynchronization interval. (6)

(e) Explain with an example the occurrence of phantom deadlocks in a distributed system. (6)

2. **(a)** 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 5×5 = 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. (7)

P.T.O

(b) Suppose you are given a connected, undirected graph with a special node i and an upper bound T on the message delay over any link. Design a global state collection algorithm that does not rely on FIFO channels and does not use piggybacking. At the end of your algorithm, the global state should be available at node i. Analyze its time complexity (in terms of T, assuming processing time can be ignored)? Clocks in the nodes are not synchronized, but you can assume that they have no drift. Assume that every process knows an upper bound n on the no. of processes in the system. (15)

3. **(a)** An arbitrary synchronous network with unique node ids is broken up into k disjoint zones with unique zone ids. Each node belongs to exactly one zone, and knows its own zone id in addition to knowing its own id. Give a protocol that elects exactly k leaders in the network such that there is one leader from each zone. When your protocol ends, exactly k nodes, one from each zone, will declare themselves as leaders, and all other nodes will know that they are not leaders. Analyze the time and message complexity of your algorithm. (12)

(b) Consider implementing Raymond's algorithm in a system where the token message and the request message each consist of 32 bits. The links are fast; but the nodes are slow. So it is more important to reduce the no. of messages rather than the size of a message, as long as the message size is constant (i.e., independent of the network size). Suggest a modification to reduce the number of messages sent in Raymond's algorithm under heavy load in this system (just write down the change, no need to write the whole algorithm again). Is the strategy also helpful under light load? Justify in 1-2 sentences. (5)

(c) Consider a weighted, connected (but not necessarily completely connected) network with n nodes, with weights indicating average delays on the link. We want to implement a Maekawa-like algorithm (the exact details are irrelevant here) on this network, but the request sets of all nodes need not be of the same size. Given two such possible sets of request sets (each with n request sets, one for each node), suggest a measure to evaluate which set is better? Note that you do NOT have to say how to form the set of request sets, just how to evaluate it. (6)

4. **(a)** Design an asynchronous distributed algorithm to find the shortest path from a designated node X to all other nodes in an undirected, connected graph with positive weights. X should be the only node to initiate the algorithm, and no flooding should be used. At the end of the algorithm, each node should have a local variable value set to its shortest path length from X. The exact path need not be maintained. What is the message complexity of the algorithm? Do not worry about how X gets to know when all the values are correctly computed at the nodes (i.e., about termination of the algorithm). (15)

(b) Suppose you are given a distributed algorithm to find an unrooted spanning tree (so no parent or child pointers, every node only knows which of its edges belong to the spanning tree) of a connected network. How can you use this algorithm to elect a single leader node in the network? Assume that processes have distinct ids. 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? (10)