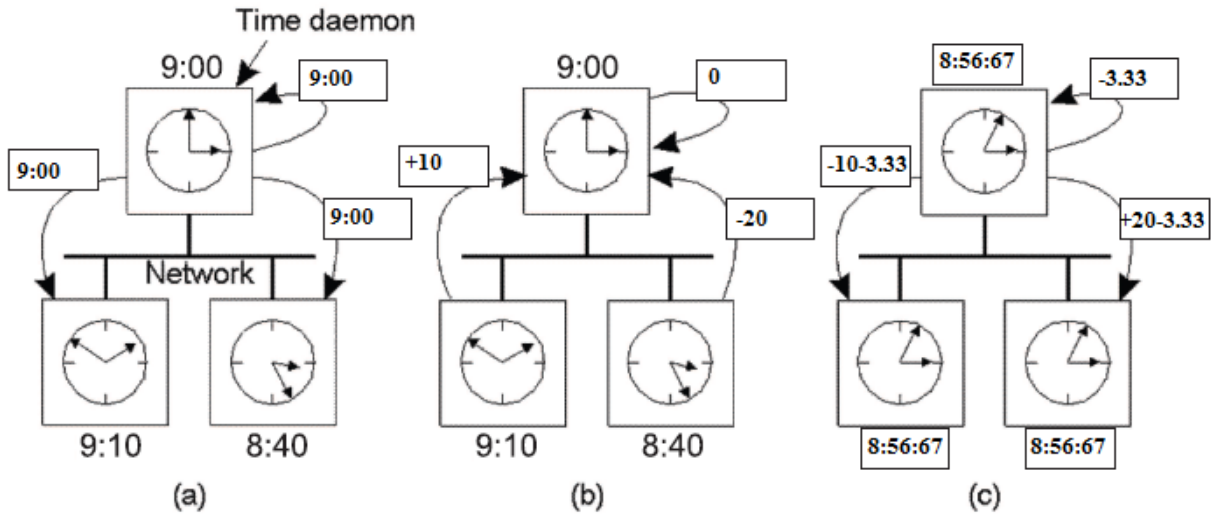
Homework Assignment 4 Solutions
ECE 151 / CMPSC 171 - Spring 2013

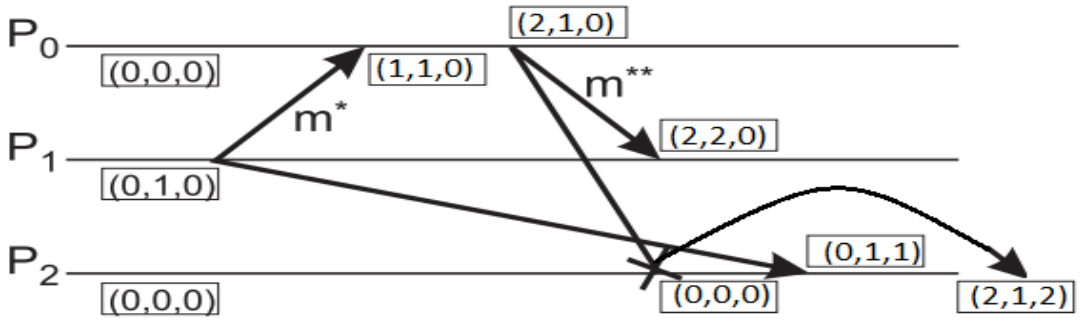
MARKING SCHEME:
Problems 1,2,3,4,5 - 10 points
Problems 6 - 50 points

1. The Berkeley Algorithm for clock synchronization consists of three steps: (a) The time daemon asks all the other machines for their clock values, (b) the machines answer, (c) the time daemon tells every machine how to adjust its clock. In the example given below, fill in the boxes to show the times or offsets that are communicated in (a), (b), (c).



The Berkeley Algorithm for clock synchronization finds the **average** of the times of the participating nodes.

2. In the following diagram, fill in the values of the vector clocks for causal ordering and show the point at which process P2 delivers the message m.**



3. (a) Explain how the distributed mutual exclusion algorithm of Ricart and Agrawala works.
(b) Explain why the distributed mutual exclusion algorithm of Ricart and Agrawala requires: (i) Total ordering of events, and (ii) Reliable delivery of messages.

(a) To access a shared resource, a process sends a request message to all processes (including itself), containing: the resource name, its process id, the current timestamp.
After sending the request message, the process waits until it has received permission from all processes and then proceeds to access the shared resource. When it has finished accessing the shared resource, the process sends OK messages to all processes in its queue and then removes them from its queue.

When a process receives a request message from another process, its action depends on its current state:
(i) If it is not accessing the resource and does not want to access the resource, the process sends an OK message to the sending process. (ii) If it already has access to the resource, the process does not reply. Instead, it queues the request message. (iii) If it wants to access the resource but has not yet done so, the process compares the timestamp of the request message it received with the timestamp of the request message it sent. If the request message it received has a lower timestamp, it sends an OK message to the sending process. If the request message it sent has a lower timestamp, it queues the request message it received and sends nothing.

(b) Requirement for total ordering: Since the algorithm compares timestamps of requests to grant access, it is necessary to have total ordering. Requests may come from different nodes which have different local clocks (timestamps have local time, not global time). Total ordering lets us compare these different local times. Without total ordering, a request which has later timestamp may be given access as compared to request with earlier timestamp. Also, if the timestamps are not compared then in case of simultaneous multiple requests, multiple nodes may be granted access to the shared resource at the same time leading to race condition.
Requirement for reliable delivery: Reliable delivery has to be ensured because the nodes wait for replies from all other nodes. Without reliable delivery, the requesting node may not get permission from all remaining nodes, leading to deadlock.

4. Is a fully distributed algorithm (*i.e.*, one without a coordinator) more robust than a centralized/coordinated algorithm?

The centralized/coordinated algorithm is more robust than the fully distributed algorithm due to the following reason.

- There is only a single point of failure in the centralized algorithm since the implementation of the algorithm gets affected only when the coordinator crashes. A node wanting to access a shared resource sends the request to the centralized server and waits for the response only from it. In the case of fully distributed algorithm there exist multiple points of failure. The implementation of the algorithm gets affected even if at least one among the nodes in the network crashes since a node wanting to access a shared resource waits for the responses from all other nodes in the system.

5. Give three examples of distributed applications that use GPS receivers.

The three examples of distributed applications that use GPS receivers are,

- Vehicle tracking system
- Disaster relief system
- Clock synchronization