Distributed Systems
Mid Semester Exam

1st March,2025

Please attempt All questions. Make assumptions if required. State them clearly.

Question 1 : Clocks and Global Snapshot (4+6)

A. Will Acharya Badrinath Algorithm for taking global snapshot for causal ordering work for FIFO Channels.. Explain your answer

B. Consider the following simple method to collect a global snapshot (it may not always collect a consistent global snapshot): an initiator process takes its snapshot and broadcasts a request to take snapshot. When some other process receives this request, it takes a snapshot. Channels are not FIFO. Prove that such a collected distributed snapshot will be consistent iff the following holds (assume there are n processes in the system and Vt_i denotes the vector timestamp of the snapshot taken process p_i): $(Vt_1[1], Vt_2[2], \ldots\ldots\ldots, Vt_n[n]) = \max(Vt_1, Vt_2, \ldots, Vt_n)$

Don't worry about whether transit messages are recorded in the channel states or not.
Note: $\max(Vt_1, Vt_2, \ldots, Vt_n)$ produces vector $(\max(Vt_1[1], Vt_2[1], \ldots, Vt_n[1]), \max(Vt_1[2], Vt_2[2], \ldots, Vt_n[2])), \ldots\ldots\ldots, \max(Vt_1[n], Vt_2[n], \ldots, Vt_n[n]))$

Question 2: Mutual Exclusion (4+3+8)

A. Consider the token based Raymonds (tree based) algorithm for mutual exclusion algorithm. In the version discussed in class, a process P in the tree can enter critical section if it has the token and the process is also on top of its request queue.

   We propose a modified approach to reduce the number of messages passed in which a request linked list is to be used instead of a request queue for every node. A process P can enter critical section if it has the token and the process P is also present anywhere in its request linked list.
   If the process is not present in its request linked list then the token is sent to the first entry in the request list. The node to which the token is sent is removed from the list as in the previous version. Everything else remains same as the original algorithm.
   Discuss reasons why you would prefer this algorithm. Also mention reasons you would not prefer this algorithm.

B. What concerns do you see if the original Raymond Algorithm is run on a graph that can have cycles instead of a tree.

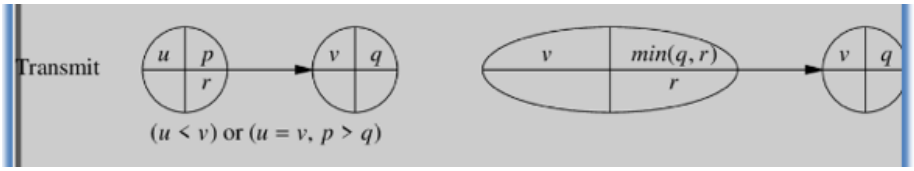
C. Consider the following four algorithms (i) token based Raymond algorithm(on trees), (ii) token based Suzuki Kasami algorithm (iii) the permission based Ricart Agarwala (which is a modification of the Chandy Lamport ME algorithm) (iv) Maekawas Algorithms that takes permission from $\sqrt{N}$ of N processes. Compare the four algorithms in a table with each other based on the following two performance metrics:

a. Synchronisation Delay - After a site leaves the CS, the number of messages required before the next site enters CS
b. Response Time – The number of messages required by a process from sending out its request messages(included) till completion of its CS execution

Question 3: Deadlock Detection (6 +4 + 5)

A. Consider the Mitchell Meritt Algorithm algorithm to detect deadlocks in single resource model.

(i) For the priority based version the transmit operation is shown below. Explain why transmit does not consider using min(p,q) instead of min(q,r). Give an example to explain what goes wrong if min(p, q) is used.



(ii). Is it necessary to generate unique public numbers in the priority and non priority versions of Merritt Michell algorithm while doing an inc(u, v) on a block. Provide an example to illustrate for either of the versions, what can go wrong or prove that we do not require unique ids.

B. Consider the Chandy Misra Haas probe based algorithm for AND model. Do you notice any concerns if multiple processes initiate the algorithm at the same time. How do you propose to fix the concern noticed ?