

1. (15) With respect to broadcast
  - (a) Explain the relationship between reliable, atomic and causal broadcast.
  - (b) Discuss the complexity of each one and rank them in order of cost w.r.t. message complexity.
  - (c) A distributed database system implementing a transaction system that manages checking accounts at a bank is to be implemented. Which of the following broadcast algorithms do you suggest should be used. Justify your choice.
2. (10) Agreement research was started when it was discovered how clocks could be fooled in a way that made them drift apart when using a midpoint algorithm. In such an algorithm in each round all processors send their values to all other processors and then use the median value as the voted value for that round. Assume you have three clocks,  $A$ ,  $B$ , and  $C$ , explain how a malicious processor (the one with clock  $A$ ) can cause clocks  $B$  and  $C$  to slowly drift apart.
3. (25) The agreement algorithm Thambidurai and Park was shown to have a problem. Take the role of an investigator and find out and explain what the problem was.
4. (25) The paper by Thambidurai and Park [Tha88] considers a 3-fault-model. The fault modes *asymmetric*, *symmetric* and *benign* are labeled  $a$ ,  $s$ , and  $b$  respectively. The respective fail rates are  $\lambda_a$ ,  $\lambda_s$  and  $\lambda_b$ .

Now consider a real-time Hybrid NMR system architecture. Recall that such a system uses voting for fault masking, in addition to using diagnosis and exclusion for recovery. According to [Tha88] this system can function properly only if  $N > 2a + 2s + b + r$ , with  $r \geq a$ .

Draw a GSPN model for system unreliability  $F(t)$  assuming you start out with  $N$  processors. Show all timed transition rates and assume that  $\rho$  is the recover rate. Recall that the recovery rate is **not** a repair rate. It simply is the rate at which components are diagnosed, masked and excluded. Note that, at any point in time, you need to have enough processors to deal with the current fault situation based on the formula above. However, note that, once you recover from a fault, e.g. a symmetric, the burden on the system is reduced, e.g. from 2 to 1 in the case of a symmetric fault.

5. (25) Assume the Byzantine Agreement algorithm by Lamport et. al. for a system consisting of processors  $P_0, P_1, \dots, P_{n-1}$ , with  $P_0$  being the general.
  - (a) How many processors does one need in order to deal with  $m$  faults?
  - (b) Draw the value tree as shown in class for  $m = 1$ , assuming that  $P_0$  is the traitor, sending out  $v_0 = 1$  to  $P_1$  and  $v_0 = 0$  to all other processors. I am interested to see the tree *before* voting.
  - (c) Now consider the scenario for  $m = 2$ . Assume that  $P_0$  is a traitor who tries to “split the vote” by sending  $v_0 = 0$  to half of the processors, i.e.  $P_1, \dots, P_3$ , and  $v_0 = 1$  to the other half, i.e.  $P_4, \dots, P_6$ .  $P_1$  is also a traitor who flips every values sent to  $P_4, \dots, P_6$  at each round, e.g.  $P_1$  receives  $v_0 = 0$  and forwards  $v_{0,1} = 1$  to the selected group of processors.
    - i. What is the final value tree **before** voting for processor  $P_2$ ?
    - ii. What is the final voted value?