# Homework 2

CS425/ECE428 Spring 2019

**Due:** Wednesday, Feb 20 at 11:55 p.m.

1. For this question, consider a distributed system where each pair of processes is connected by a first-in, first-out (FIFO) reliable channel. Assume that there are no process failures.

   (a) (4 points) In this system, if the basic multicast scheme (`B-multicast`) is used, what ordering guarantees would be satisfied? FIFO, causal, and/or total? Justify your answer: either argue why a guarantee *would* be satisfied, or present a counterexample.

   Assume that a process immediately delivers a message to itself before sending it out to any other processes.

   (b) (4 points) What if the reliable multicast scheme (`R-multicast`) were used instead? Assume again that the process immediately delivers a message to itself before sending it out.

   (c) (2 points) Do your answers for the above two parts change in a system with only two processes?

2. (a) (3 points) Consider a synchronous system with $N = 100$ processes running the synchronous consensus protocol. When sending a message, each value $v_i = z$ is encoded using two bytes. Assuming there are no failures, how many total bytes are sent by the system over five rounds of the protocol, starting from round 1?

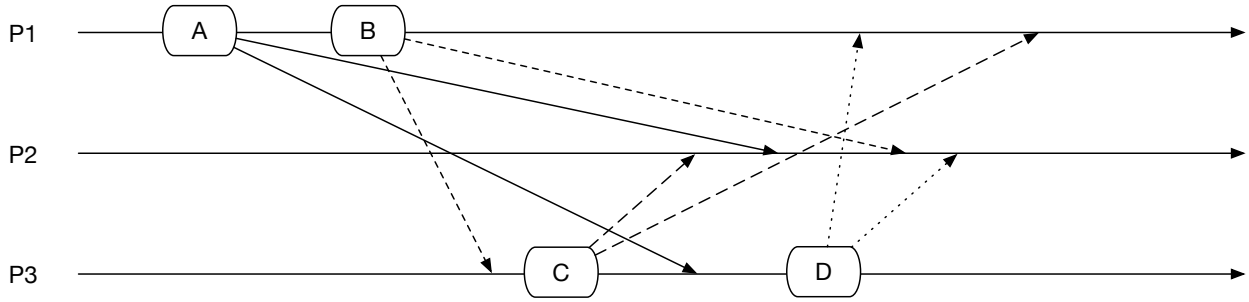
   (b) (3 points) Consider a synchronous system where processes can only fail in pairs, so that in any round, an even number of proceses fail (e.g., 0, 2, 4, etc.) How many rounds would you need to ensure consensus among $N = 100$ processes if there is no bound on the number of failures?

3. (3 points) Consider the following sequence of events, using the model from the Fisher-Lynch-Patterson proof:

| | |
|---|---|
| $e_1 @ p_1$ | $\mathsf{receive}(m_1), \mathsf{send}(p_2, m_2), \mathsf{send}(p_3, m_3)$ |
| $e_2 @ p_2$ | $\mathsf{receive}(m_2), \mathsf{send}(p_1, m_4)$ |
| $e_3 @ p_3$ | $\mathsf{receive}(m_3), \mathsf{send}(p_2, m_5), \mathsf{send}(p_1, m_6)$ |
| $e_4 @ p_1$ | $\mathsf{receive}(m_4), \mathsf{send}(p_3, m_7)$ |
| $e_5 @ p_1$ | $\mathsf{receive}(m_6), \mathsf{send}(p_2, m_8)$ |
| $e_6 @ p_2$ | $\mathsf{receive}(m_8)$ |

   Which events can commute with $e_2$?

4. For this question, consider the diagram below. It shows four messages, $A, B, C, D$, being multicast by processes $P1$ and $P3$.




(a) (6 points) Consider a FIFO multicast implementation using sequence numbers, as discussed in class. Write down the sequence number vector for each send and receive event at each process, and the sequence number sent with each message. Also indicate when each message will be delivered at each process. (Do not worry about processes delivering their own message, e.g., $P1$ delivering $A$.)

(b) (5 points) Repeat the exercise with a causal multicast implementation. (Instead of a sequence number for each message you will include its sequence vector).