CSC 258/CSC 458
Assignment 2
April 6th, 2020
Due April 13th, 2020

Please be sure to explain your answers in sufficient detail and write down any assumptions you make.

1. Consider the use of synchronized clocks for at-most-once delivery, as described in [Liskov et al., Efficient at-most-once messages based on synchronized clocks, May 1991] (the algorithm is also outlined in slide 6 of the class lecture slides on the topic). Suppose your disk crashes in addition to your machine, so that on recovery, you are unable to read the time (for the largest timestamp possibly accepted/delivered) from disk. How could you continue to ensure at-most-once delivery while minimizing the number of messages rejected (what value would you use to initialize G, the minimum time of a message you will accept, on recovery)?

2. Why would one use vector timestamps over lamport timestamps, i.e., what does one provide that the other does not and why might this be useful? Describe one instance each where you think vector or lamport timestamps are more appropriate. Explain your rationale.

3. TreadMarks is a software distributed shared memory system built for high-latency local area networks, while Cashmere is a software distributed shared memory system targeted at low-latency system area networks. Describe the key differences between the consistency models provided by the two systems (i.e., lazy release consistency and moderately lazy release consistency). How do they differ from the release consistency model specified in the Adve-Gharachorloo paper? Illustrate with a common programming practice that would work under the release consistency model and not under either the TreadMarks or Cashmere model.

4. Chandy and Lamport introduced the notion of a distributed snapshot, which reflects a state in which the distributed system might have been. An important property is that such a snapshot reflect a consistent global state. Describe (in one sentence) what the consistency requirement is. Describe two applications that demonstrate the utility of being able to capture a distributed snapshot.