CMPS 4760/6760 Distributed Systems – Spring 21

Lab 2 (20 points)

Due 04/01/21 before class

This lab may be performed by a group of two students, or individually. You may use either Java or Python to implement the lab.

In this lab, you will implement a simple bulletin board that allows multiple users to post messages using a command line interface. You should allow at least **four** processes to post and receive messages using **ordered multicast**. Your multicast implementation should support **total ordering** and **causal ordering** (the latter implies FIFO ordering as we discussed in class).

For each process, you need to implement a multicast interface that includes a *multicast* methohd and a *receive* method. You should also implement the application layer methods for reading the messages to be posted from the command line and displaying the received messages. In addition, to support total ordering using a sequencer (which can be one of the processes), you should also implement the interface for the sequencer.

You may assume all the processes are in the same group and all the group memebers are fixed and known to each process. Therefore, no group id is needed and you don't need to worry about group membership management. You can use UDP sockets to implement message passing and basic multicast. Since all the processes are running on the same machine, they should be given different port numbers.

Since all the processes are running on the same machine, you may assume that the channels are reliable. However, you need to simulate channel delays to make the problem more interesting. In particular, a random transmission delay should be added to each message so that messages on the same channel may be recevied out of order. This can be implemented at either the sender side or the receiver side.

You need to turn in:

1. the source code;
2. a README file that describes:
   (a) how to compile and run your code. **It is your responsibility to make sure that I can compile and run your code;**
   (b) how you address the above requirements in your implementation;
   (c) how the workload is distributed among the group members;
3. screenshots of the command windows for posting and displaying messages, which should demonstrate how total ordering and causal ordering are supported.