**Assignment 1**

**(20 points)** Consider some client-server architecture as follows. A number of clients are registered to the server. Periodically, each client sends message to the server. Once the server receives a message, it distributes the message to all other registered clients. Additionally, the server randomly delays each received message while broadcasting to a registered client.

To solve this question, you will do the following:

1. Simulate the behavior of both the server and the registered clients via GO routines.
2. Use Lamport’s logical clock to determine a total order of all the messages received at all the registered clients. Subsequently, present (i.e. print) this order for all registered clients to know the order in which the messages should be read.
3. Use Vector clock to redo the assignment. Once the vector clocks are assigned for all the events, check for any causality violation and present the causality violation to the programmer (if any).

**(20 points)** Use GO language to implement the original Bully algorithm. You can assume a fixed timeout to simulate the behavior of detecting a fault. You can also randomly select a GO routine to be the victim (i.e. faulty node). The objective is to have a consensus across all GO nodes in terms of the newly elected coordinator.

1. While implementing your solution, try to simulate both the worst-case and the best-case situation.
2. Consider the case where a GO routine fails during the election, yet the routine was alive when the election started.
3. Multiple GO routines start the election process simultaneously.