Grading Policy for the second programming assignment
Total points = 50

You should print the activities of each computer (coordinator and participants). In particular we need to see:
1) If a participant is waiting for the coordinator's "VOTE_REQUEST"
2) If a participant is waiting for the coordinator's decision (either "GLOBAL_COMMIT" or "GLOBAL_ABORT")
3) If a participant receives a message from the coordinator or other participant
4) If a participant locally aborts or commits
5) If a participant times out and locally aborts
6) If a participant sends a message (and which message) to the coordinator or other participants
7) If the coordinator multicasts messages (and which message)
8) If the coordinator is waiting for replies from the participants
9) If the coordinator receives messages from the participants
10) Any other information you may find relevant to show that your program implement the cases we will be evaluating.

We recommend having two programs, one acting as the coordinator and the other as the participants.

5pts: Demo your implementation on 3 different computers.

5pts: Show the normal case in which:
1) The coordinator multicasts a "VOTE_REQUEST"
2) All participants reply with a "VOTE_COMMIT"
3) The coordinator receives all the replies and multicasts a "GLOBAL_COMMIT"
4) All participants locally **commit**

5pts : Show the normal case in which:
1) The coordinator multicasts a "VOTE_REQUEST"
2) At least one participant replies with "VOTE_ABORT" and others with "VOTE_COMMIT"
3) The coordinator receives all the replies and multicasts a "GLOBAL_ABORT"
4) All participants locally **abort**

10pts: Show the special case in which:
1) The coordinator multicasts a "VOTE_REQUEST"
2) At least one participant takes too much time to reply
3) The coordinator times out and multicasts a "GLOBAL_ABORT"
4) All participants locally **abort**

10pts: Show the special case in which:
 1) The coordinator takes too much to send a "VOTE_REQUEST"
 2) Participants time out and (all of them) send a "VOTE_ABORT" to the coordinator
 3) The coordinator multicasts a "GLOBAL_ABORT"
 4) All participants locally **abort**

15pts: Show the special case in which:
 1) The coordinator multicasts a "VOTE_REQUEST"
 2) All participants reply with a "VOTE_COMMIT"
 3) The coordinator sends a "GLOBAL_COMMIT" to one of the participants.
 4) The participant that did not receive the "GLOBAL_COMMIT" times out and asks the other participant by sending "DECISION_REQUEST"
 5) Participant receiving the "DECISION_REQUEST" replies with a "GLOBAL_COMMIT" to the requester
 6) The requesting participant, after receiving "GLOBAL_COMMIT" locally **commits**

NOTE: Any other case will not be evaluated and as such, you are not required to implement it.