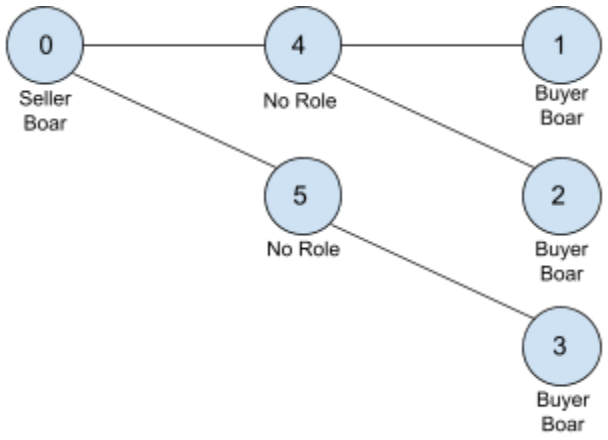
TEST CASE DOCUMENTATION

For this milestone, we have 2 test cases against which we test our peer to peer network.

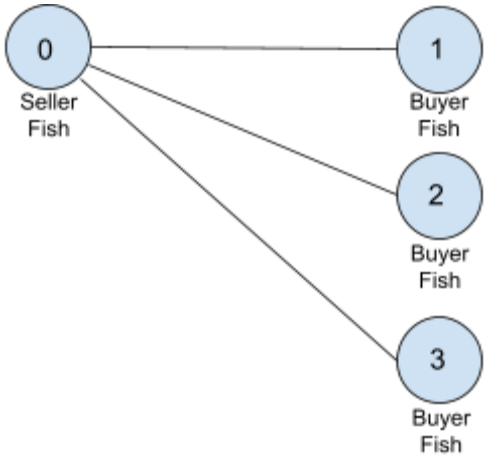
TEST CASE 1 :

One seller of boar, 3 buyers of boars, the remaining peers have no role. Fix the neighborhood structure so that buyers and sellers are 2-hop away in the peer-to-peer overlay network. Ensure that all items are sold and restocked and that all buyers can buy forever.
For this test case, we hard code a network as shown below to ensure all the conditions mentioned in the test case are followed accordingly. The network can be seen below with the respective roles and items. The seller is initialized with 5 units of item boar. After each successful transaction, the buyer waits for a specific amount of time before looking to buy another unit of boar. When the seller runs out of items to sell it will restock back to 5 units. All transactions will continue for 40 seconds following which the node processes will be terminated.



TEST CASE 2 :

Simulate a race condition in buy() wherein a seller has stock of 1 for an item but then replies to multiple buyers. For this test case, we hard code a network as shown below to ensure all the conditions mentioned in the test case are followed accordingly. The network can be seen below with the respective roles and items. After sending the lookup request, the buyers will wait for a small unit of time before attempting to complete the transaction. Once the race condition is simulated, only a single buyer will be able to complete the transaction. The other nodes will not be able to complete the transaction in spite of being able to find a seller. Thus, race condition is simulated. This is further proved when the seller sends back the message “Sorry I have sold my item to someone else”. This will be sent to the two buyers who will not be able to buy. This can be checked in the log files. After 5 seconds, all node processes will be terminated.



DEFAULT :

If no test case is passed, a completely random network is formed with N nodes and k peers for each node where k<N. These values of N and k are read from a config.txt file. After the network is initialized, the buyers will randomly pick items to buy and attempt to buy them from its peers. Each seller will sell 5 units of an item until it runs out at which point it will randomly select an item to restock back to 5 units. Transactions will continue for 40 seconds following which all node processes are terminated.

We log all output of the test cases in separate test case log files.