CS 677 Lab1 Test Document

1. Message formats:

- [INIT] Peer 3 plans to buy Boar
  => peer ID 3 generates a buy request

- [INIT] Peer 2 sells 8 units of Salt
  => peer ID 2 generate a commodity to sell

- [LOOKUP propagate] Peer 1: (next) 2 <- 1 (curr) - 0-3 (path)
  => peer ID 1 propagates a lookup request to peer ID 2. The lookup request has traveled from peer ID 3 -> peer ID 0 -> current peer.

- [REPLY propagate] Peer 3: (curr) 3 -> 0 (next) --> (path)
  => peer ID 3 propagates a reply request to peer ID 0. Peer ID 0 is a buyer.

- [REPLY propagate] Peer 2: (curr) 2 -> 1 (next) --> 0 (path)
  => peer ID 2 propagates a reply request to peer ID 1. This reply request will stop at peer ID 0.

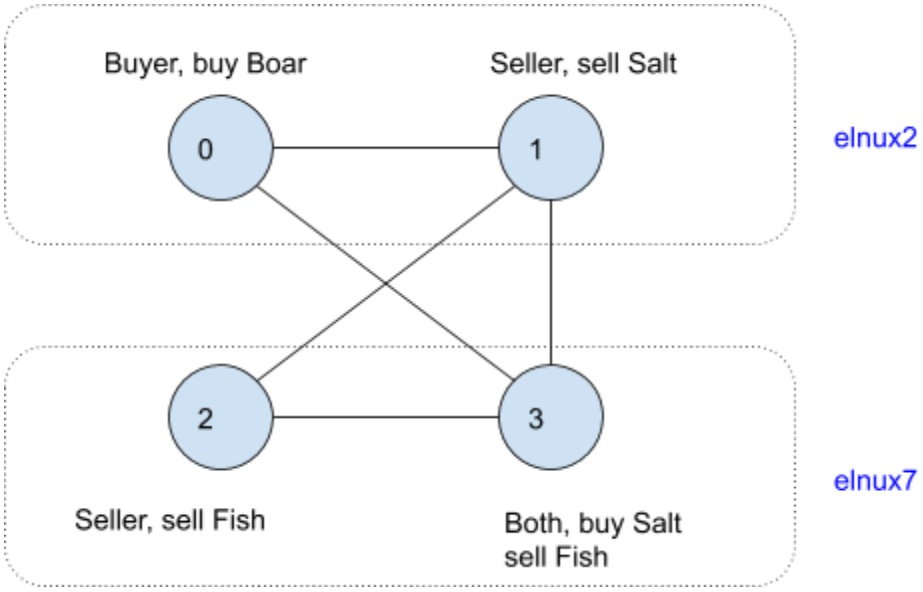
- Peer 2 has Salt!! Reply to peer 3
  => peer ID 2 has salt to sell. It will reply to neighbor peer ID 3

- [SUCCESS] Peer 3 buys Salt from peer 1
  => peer ID 3 successfully buys salt from peer ID 1

- [FAILURE] The Fish of peer 3 has been sold out.
  => peer ID 3 sold out its Fish

2. Output:

Below is the output between two machines with total 4 peers. The output results worked as we had expected. The peers are connected as following:



Output from elnux2:

elnux2 lab1) > python3 node.py
Node is running on: 128.119.243.164

\\\\\ *NEIGHBOR MAP* /////
[False, True, False, True]
[True, False, True, True]
[False, True, False, True]
[True, True, True, False]

[Neighbor] Neighbors of peer 0: 1,3
[Neighbor] Neighbors of peer 1: 0,2,3
[INIT] Peer 1 sells 3 units of Salt
[LOOKUP propagate] Peer 0: (next) 1 <- 0 (curr) - 3 (path)
Peer 1 has Salt!! Reply to peer 3
Peer 1 has Salt!! Reply to peer 2
Peer 1 has Salt!! Reply to peer 0
[REPLY propagate] Peer 0: (curr) 0 -> 3 (next) --> (path)
[INIT] Peer 0 plans to buy Boar
[LOOKUP propagate] Peer 1: (next) 2 <- 1 (curr) - 0 (path)

[LOOKUP propagate] Peer 1: (next) 3 <- 1 (curr) - 0 (path)
[LOOKUP propagate] Peer 1: (next) 2 <- 1 (curr) - 3-0 (path)
[UPDATE] Peer 1 has 2 units of Salt to sell
[LOOKUP propagate] Peer 0: (next) 1 <- 0 (curr) - 3 (path)
Peer 1 has Salt!! Reply to peer 3
Peer 1 has Salt!! Reply to peer 0
[REPLY propagate] Peer 0: (curr) 0 -> 3 (next) --> (path)
Peer 1 has Salt!! Reply to peer 2
[INIT] Peer 0 plans to buy Boar
[LOOKUP propagate] Peer 1: (next) 2 <- 1 (curr) - 0 (path)
[LOOKUP propagate] Peer 1: (next) 3 <- 1 (curr) - 0 (path)
[LOOKUP propagate] Peer 1: (next) 2 <- 1 (curr) - 3-0 (path)
[UPDATE] Peer 1 has 1 units of Salt to sell
[LOOKUP propagate] Peer 0: (next) 1 <- 0 (curr) - 3 (path)
Peer 1 has Salt!! Reply to peer 3
Peer 1 has Salt!! Reply to peer 0
[REPLY propagate] Peer 0: (curr) 0 -> 3 (next) --> (path)
Peer 1 has Salt!! Reply to peer 2
[INIT] Peer 0 plans to buy Salt

Output from elnux7:
elnux7 lab1) > python3 node.py
Node is running on: 128.119.243.175

\\\\\ *NEIGHBOR MAP* /////
[False, True, False, True]
[True, False, True, True]
[False, True, False, True]
[True, True, True, False]

[Neighbor] Neighbors of peer 2: 1,3
[Neighbor] Neighbors of peer 3: 0,1,2
[INIT] Peer 2 sells 5 units of Fish
[INIT] Peer 3 sells 7 units of Fish
[INIT] Peer 3 plans to buy Salt
[LOOKUP propagate] Peer 2: (next) 1 <- 2 (curr) - 3 (path)
[RECEIVE] Peer 3 receives a reply from peer 1
[REPLY propagate] Peer 2: (curr) 2 -> 3 (next) --> (path)
[RECEIVE] Peer 3 receives a reply from peer 1
[RECEIVE] Peer 3 receives a reply from peer 1
[LOOKUP propagate] Peer 3: (next) 1 <- 3 (curr) - 0 (path)

[LOOKUP propagate] Peer 3: (next) 2 <- 3 (curr) - 0 (path)
[LOOKUP propagate] Peer 2: (next) 3 <- 2 (curr) - 1-0 (path)
[LOOKUP propagate] Peer 3: (next) 2 <- 3 (curr) - 1-0 (path)
[LOOKUP propagate] Peer 2: (next) 1 <- 2 (curr) - 3-0 (path)
[SUCCESS] Peer 3 buys Salt from peer 1; avg. response time: 0.023833 (sec/req)
[INIT] Peer 3 plans to buy Salt
[LOOKUP propagate] Peer 2: (next) 1 <- 2 (curr) - 3 (path)
[RECEIVE] Peer 3 receives a reply from peer 1
[REPLY propagate] Peer 2: (curr) 2 -> 3 (next) --> (path)
[RECEIVE] Peer 3 receives a reply from peer 1
[RECEIVE] Peer 3 receives a reply from peer 1
[LOOKUP propagate] Peer 3: (next) 1 <- 3 (curr) - 0 (path)
[LOOKUP propagate] Peer 3: (next) 2 <- 3 (curr) - 0 (path)
[LOOKUP propagate] Peer 2: (next) 3 <- 2 (curr) - 1-0 (path)
[LOOKUP propagate] Peer 3: (next) 2 <- 3 (curr) - 1-0 (path)
[LOOKUP propagate] Peer 2: (next) 1 <- 2 (curr) - 3-0 (path)
[SUCCESS] Peer 3 buys Salt from peer 1; avg. response time: 0.023097 (sec/req)
[INIT] Peer 3 plans to buy Salt

3. Known Issues:

- A RPC server (peer) may shut down unexpectedly or refuse to connect especially when all peers are both sellers and buyers in a fully connected network. If this happens, please restart the program.
- The RPC server cannot be initialized if the specified port has been occupied by another service. Please change the value of **`PORT_START_NUM`** in define.py