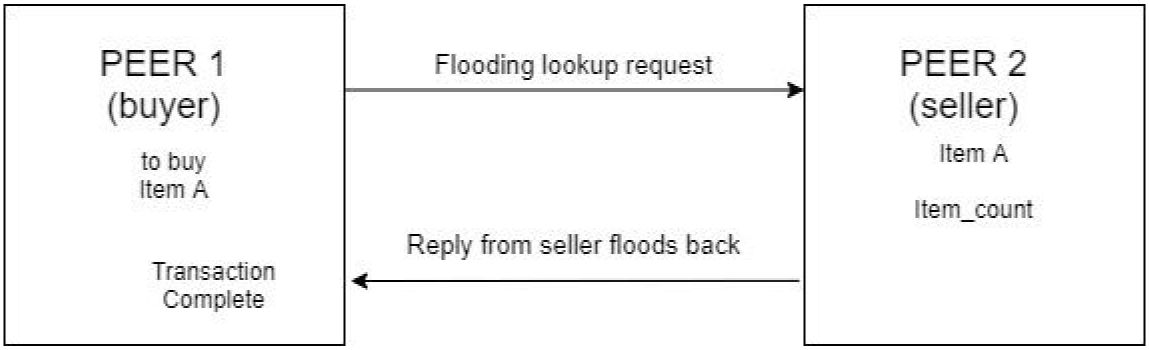
GENERAL DESCRIPTION

This project implements a peer to peer network with two peers. Out of these, one peer is assigned the role of a buyer while the other is assigned the role of a seller. Each peer knows the Peer IDs of its neighbours. The buyer and seller are initialized with items based on each test case. The seller generates m=5 quantities of an item, and once it runs out of items to sell, it restocks an item dependending on the testcase.



Currently, the P2P network is initialized with two Peers, one of which is buyer and the other is seller.
At the code level, we instantiate and deal with peers through the 'Node' class.

Each peer is instantiated with a peer ID, role, port it listens on, it's list of neighbours, item it sells or buys and the count of it.

The buyer loops through all it's available peers and sends lookup requests for the item it wants to buy(*flooding*). The lookup_helper is the Java RMI function that the peer exposes to the network. If the current peer is the seller of the item which the buyer is looking for, the seller traces back to the buyer using the reply_helper interface. If the current peer is not the seller of the item which the buyer is looking for, it will forward the lookup request for the particular item to its next neighbours. Upon receiving the ID of the seller, the buyer can initiate a transaction directly with the seller using the buy interface that the seller exposes. The buyer buys the item, waits for a certain amount of time and then picks another item to buy. The seller keeps decreasing the count of the item it is selling. Once the item stock finishes, the seller restocks with another randomly picked item with the same count as before(m=5).