TEST DOCUMENT

CS 677 - Lab 1

1. Introduction

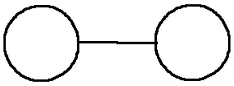
This document provides a detailed description of all the test cases, how to run them and how to verify the correctness of the system. Since this is a concurrent system, it was not possible to write unit tests, so we have aimed to write test cases which cover 100% of our code, in addition to extensively testing the robustness of the system. To this end, each case represents a different topology of our peer to peer system. The following sections describe the said topologies.

2. Test Cases

Each test case represents a different network topology which is described as follows:

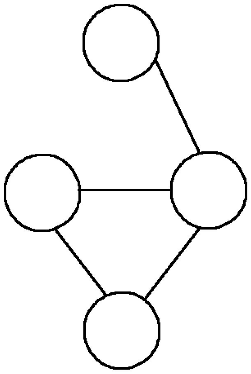
a. Test Case 1 - Two Nodes

This test case is a simple one having two nodes connected to each other, just to check if the system is working or not.



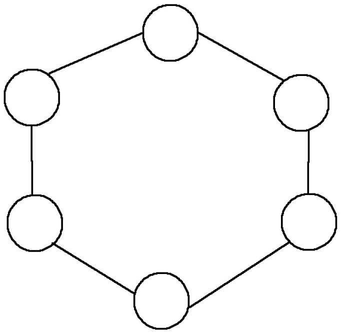
b. Test Case 2 - Four Nodes

This test case represents the following topology:



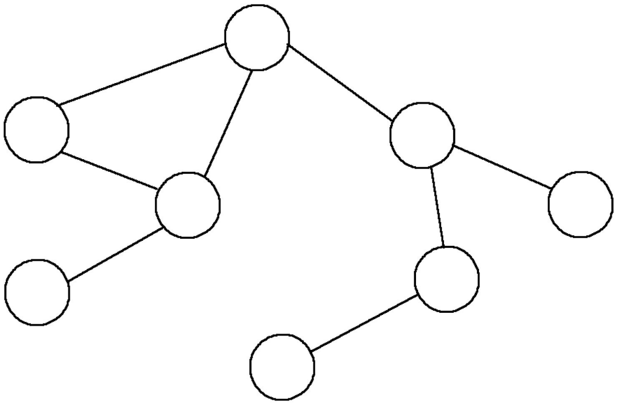
c. Test Case 3 - Six Nodes

This test case represents the following topology:



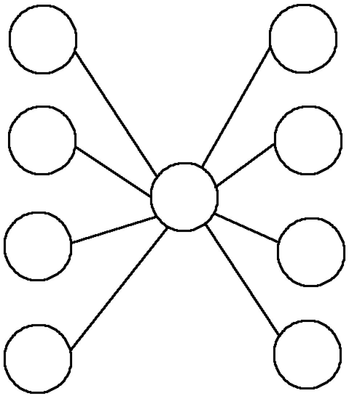
d. Test Case 4 - Eight Nodes

This test case represents the following topology:



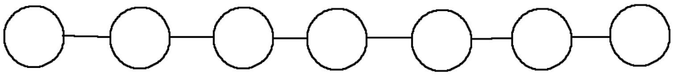
e. Test Case 5 - Nine Nodes

This test case represents the following topology:



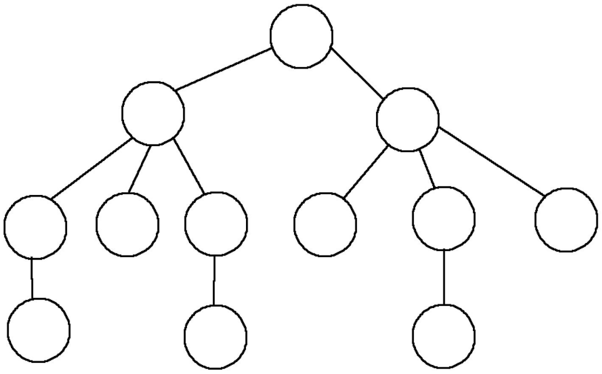
f. Test Case 6 - Seven Nodes

This test case represents the following topology:



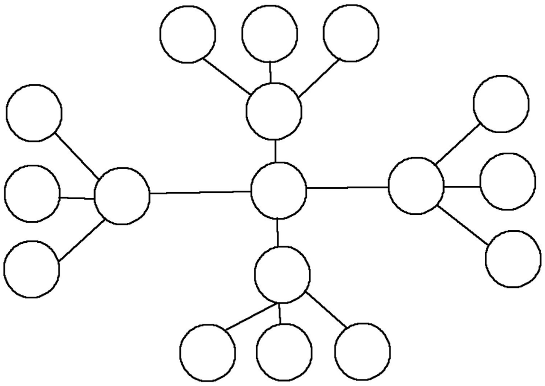
g. Test Case 7 - Twelve Nodes

This test case represents the following topology:



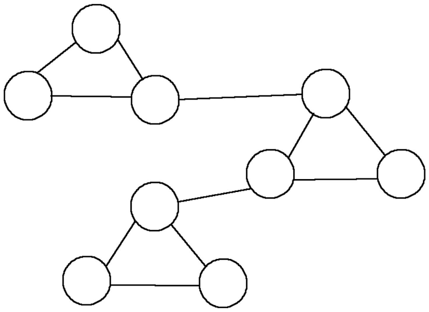
h. Test Case 8 - Seventeen Nodes

This test case represents the following topology:



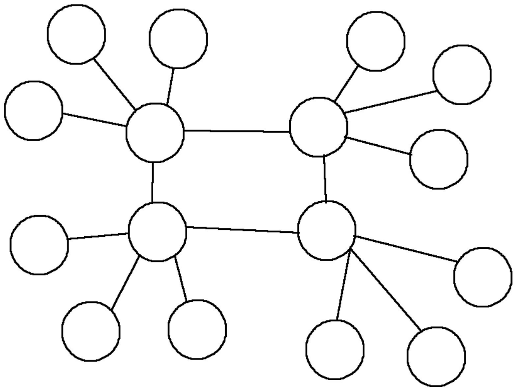
i. Test Case 9 - Nine Nodes

This test case represents the following topology:



j. Test Case 10 - Sixteen Nodes

This test case represents the following topology:



3. Running the Test Cases

Each testcase is a shell script, which can be run as follows:

```
> chmod +x test1.sh
> ./test1.sh
```

to produce output files of the format *"PeerID_Testcase_output.txt"*

4. Verifying the correctness

Each output file will contain details about the peer -
1. The role of the peer (buyer/seller/both)
2. The product which the peer is buying/selling
3. Transactions

It's very easy to verify the correctness of the system. For each buy message in a buyer node, there should be a corresponding sell message in the seller node. We can easily find all matching pairs, and if there are no odd ones left (only buy no sell or vice versa) then the system works correctly.

5. Final Thoughts

We have extensively tested our system on a variety of test cases apart from what are included in this document. Our system has worked perfectly for all of them on our local system. However, there is an issue while running the system on Edlab, especially for more peers, due to resource constraints. As such, the system should be run in an isolated environment, with ample resources available.