TEST DOCUMENT

CS 677 - Lab 1

1. Introduction

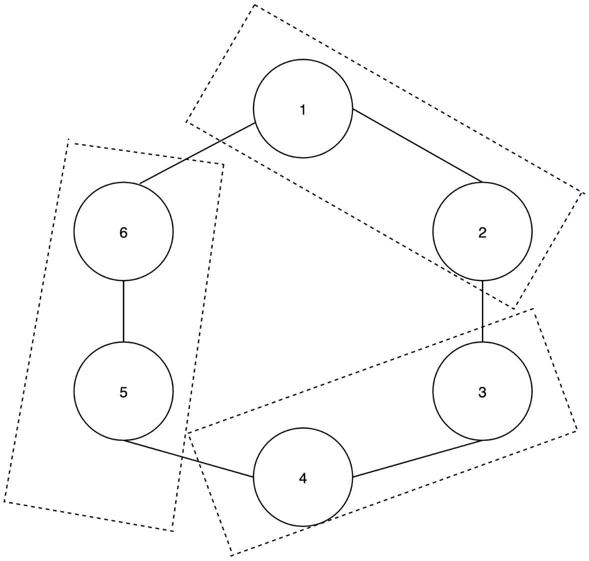
This document provides a detailed description of all the test cases, how to run them and how to verify the correctness of the system. Since this is a concurrent system, it was not possible to write unit tests, so we have aimed to write test cases which cover 100% of our code, in addition to extensively 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

Dotted boxes represent which nodes are running on the same machines
Each test case represents a different network topology which is described as follows:

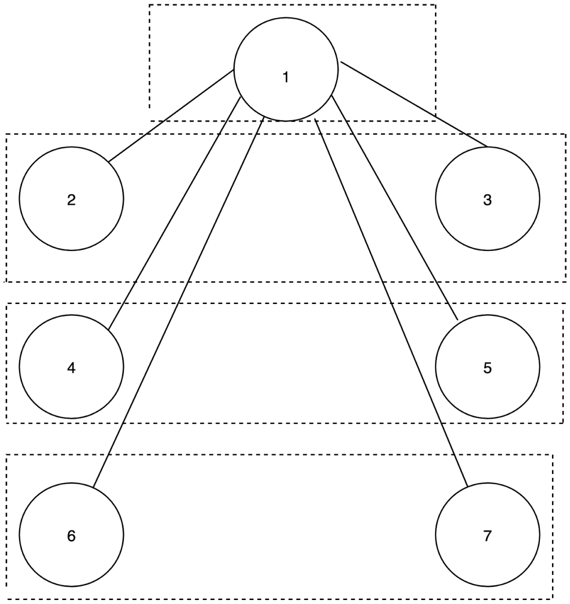
a. Test Case 1 - Six Nodes (Ring)

This test case represents the following topology:



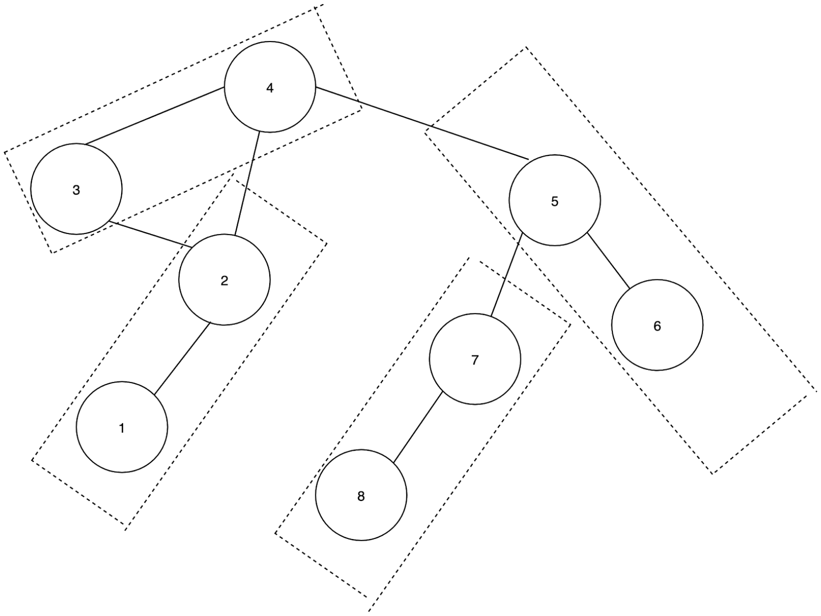
b. Test Case 2 - Seven Nodes

This test case represents the following topology:



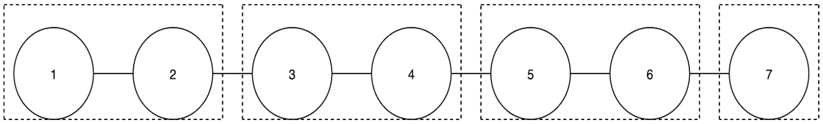
c. Test Case 3 - Eight Nodes

This test case represents the following topology:



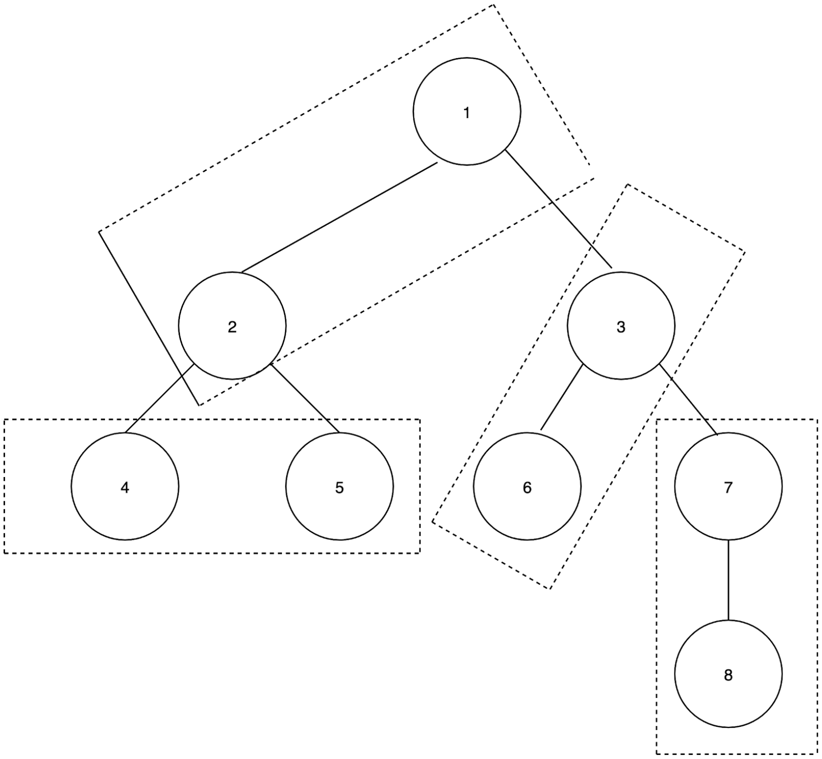
d. Test Case 4 - Seven Nodes

This test case represents the following topology:



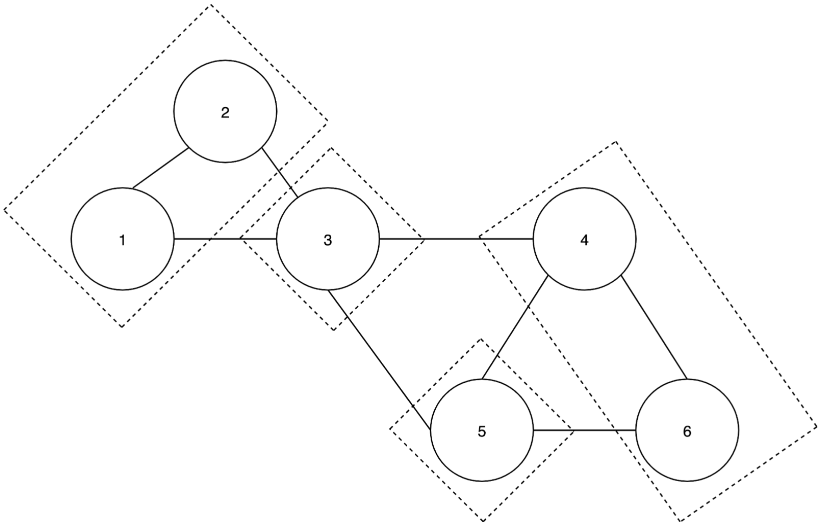
e. Test Case 5 - Eight Nodes

This test case represents the following topology:



f. Test Case 6 - Six Nodes

This test case represents the following topology:



3. Running the Test Cases

Each test case is placed in its own sub-directory. Each sub-directory contains a number of shell scripts, one for each machine that will be involved in the test (since each script dictates which nodes will be spun on that machine). The scripts are titled as : “Test_<Test case no.>_EDLAB_<EDLAB machine no.>.sh”. As such, each script will need to be executed on the appropriate EDLAB machine no. mentioned in the name of the script

Each test case is a collection of shell scripts, which can be run as follows:

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

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

All involved scripts should be run in quick succession

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 while running more than 2 peers on a single node, due to resource constraints on edlab. As such, the tests have been designed to exhibit the system’s distributed nature while keeping this constraint in mind, and should be run in an isolated environment in order to ensure availability of specified ports on all machines, with ample resources available.