Computer Networks

Assignment 2

Overview of the Problem:

To mimic remote program execution, use the OMNeT++ Discrete Event Simulator. Each job is split up into n subtasks and delivered to n/2 + 1 servers. The majority's response will be regarded as legitimate. Server is requested to carry out a basic operation, such as determining the number of vowels in the string. The client splits string into about n equal portions, where $x/n \geq 2$.

Implementation:

First we had created a network architecture made up of clients and servers that are connected to one another to enable communication. Two simple modules Client and Server are initiated. Each module is configured with parameters like ‘id’, ‘numServers’ and ‘numClients’.

Code Explanation:

1) **Server.cc :** This file contains declaration of Server module. Server will perform task of “Counting Vowels” in the input string sent by the Client. Server may or may not respond honestly. Server will act maliciously randomly with some probability such that at most numServers / 4 servers will act maliciously to identify the correct result.

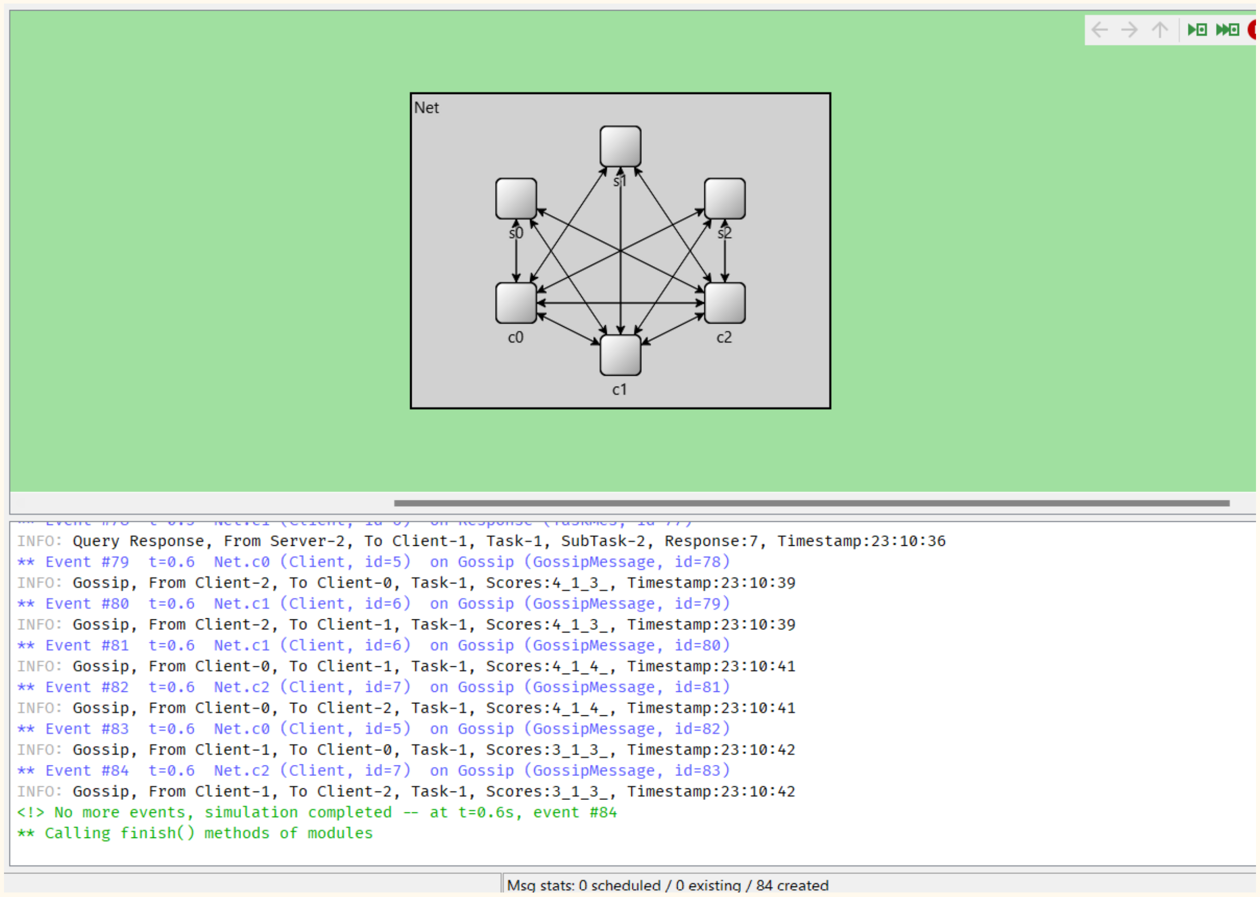
2) **Client.cc :** This file contains declaration of Client module. Firstly client will divide task into numServers subtasks and then it will send numServers subtasks to numServers / 2 + 1 selected servers. For Task 1, these servers will be selected randomly. For later tasks servers will be selected on the basis of scores assigned by clients till previous tasks. Clients will also communicate these scores with each other. We can set value of the variable ‘numOfRounds’ and task query string in initialisation of the Client module to perform those many tasks dynamically. Method ‘performTask1()’ will execute for the first round and for subsequent rounds method ‘performTask2()’ will be executed. Variable ‘serverScores’ will store server scores based on server behaviour.

3) "TaskMes" and "GossipMessage" are message classes used for message transmission. Network.ned file will be generated dynamically using 'topo.txt' configuration file.

Steps to execute the program:

1) Create topo.txt configuration file and then generate Network.ned file using python script file script.py.
2) Set Values of variables numServers and numClients in omnet.ini file.
3) Set Values of variables numOfRounds and set task query strings of length greater than 2* numServers.
4) Build and Execute the simulation in Omnetpp environment.

Output Screenshots: Network Simulation



```
INFO: Query Response, From Server-2, To Client-1, Task-1, SubTask-2, Response:7, Timestamp:23:10:36
** Event #79  t=0.6  Net.c0 (Client, id=5)  on Gossip (GossipMessage, id=78)
INFO: Gossip, From Client-2, To Client-0, Task-1, Scores:4_1_3_, Timestamp:23:10:39
** Event #80  t=0.6  Net.c1 (Client, id=6)  on Gossip (GossipMessage, id=79)
INFO: Gossip, From Client-2, To Client-1, Task-1, Scores:4_1_3_, Timestamp:23:10:39
** Event #81  t=0.6  Net.c1 (Client, id=6)  on Gossip (GossipMessage, id=80)
INFO: Gossip, From Client-0, To Client-1, Task-1, Scores:4_1_4_, Timestamp:23:10:41
** Event #82  t=0.6  Net.c2 (Client, id=7)  on Gossip (GossipMessage, id=81)
INFO: Gossip, From Client-0, To Client-2, Task-1, Scores:4_1_4_, Timestamp:23:10:41
** Event #83  t=0.6  Net.c0 (Client, id=5)  on Gossip (GossipMessage, id=82)
INFO: Gossip, From Client-1, To Client-0, Task-1, Scores:3_1_3_, Timestamp:23:10:42
** Event #84  t=0.6  Net.c2 (Client, id=7)  on Gossip (GossipMessage, id=83)
INFO: Gossip, From Client-1, To Client-2, Task-1, Scores:3_1_3_, Timestamp:23:10:42
<!> No more events, simulation completed -- at t=0.6s, event #84
** Calling finish() methods of modules
```

Msg stats: 0 scheduled / 0 existing / 84 created

<u>Made by:-</u>

1)Anurag Kumar Bharti [B21CS012]

2)Ankush Deshmukh [B21CS022]