Assignment 2: Communication in Distributed Systems

This assignment is a continuation of assignment 1 “Coordination and Leader Election Simulating and Evaluating Distributed Protocols in Java”. The aim of this coursework is to build a Server-Client network. In other words, a server will be created using Sockets and the Client will be able to send information to the server and the server will then process the data and send back the results. The process that the server will effectuate is the same algorithm as in assignment 1, LCR and HS implementations. Therefore, the results remain the same and there is no need to put them again in this report.

1. Description of the new functionalities

This new implementation is divided into two parts: the client, and the server. The client part contains one class named (“Client”). The purpose of this class is to send information and receive information to print out depending on the Server. The server part contains most of the data processing. Therefore, the previous classes remain in the server part, with the addition of class Server which calls the needed functions.

The Client is only a graphical front-end which means it only processes what is necessary to present the surface. The Server operates all the data received and does all the leader election simulation.

2. Connection to the server and exchange of information

To connect the two servers, I used a local host since I was running the simulation on one laptop only. Only one client can connect to the server. The system follows a request-reply behavior following figure 1.

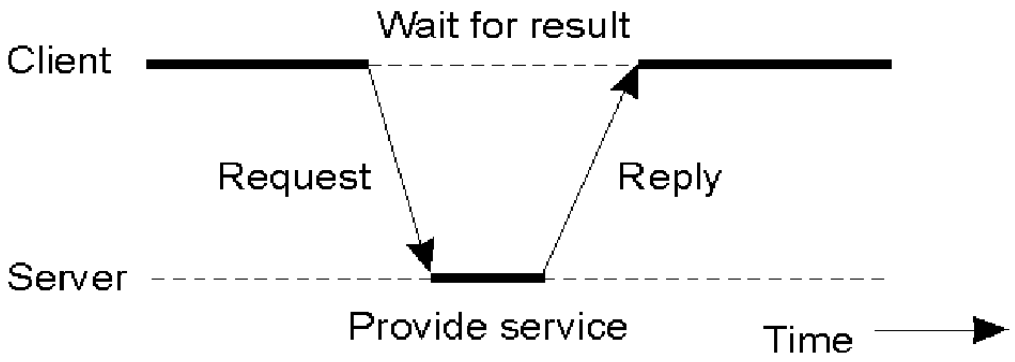

Figure 1: Request-reply system

As depicted in figure 1, the Client will stop all action and wait for the result from the server.

```
Welcome to my COMP212 Assignment
Please choose which algorithm you want: A for LCR algorithm and B for HS algorithm
a
A: Random || B: Ascending || C: Descending
a
Insert the number of processors you want. Please choose a number between 0 - 1000
5
The number of processor is 5
5
Now: 7
Previous: 6
Next: 4
---------------------------------------
Now: 4
Previous: 7
Next: 2
---------------------------------------
Now: 2
Previous: 4
Next: 12
---------------------------------------
Now: 12
Previous: 2
Next: 6
---------------------------------------
Now: 6
Previous: 12
```

```
My ID: 7| ID to Send: 12| Status: UNKNOWN
My ID: 2| ID to Send: 7| Status: UNKNOWN
My ID: 6| ID to Send: null| Status: UNKNOWN
My ID: 4| ID received: 12| Status: UNKNOWN
My ID: 12| ID received: 7| Status: UNKNOWN
My ID: 7| ID to Send: null| Status: UNKNOWN
My ID: 4| ID to Send: 12| Status: UNKNOWN
My ID: 2| ID to Send: null| Status: UNKNOWN
My ID: 12| ID to Send: null| Status: UNKNOWN
My ID: 6| ID to Send: null| Status: UNKNOWN
My ID: 7| ID received: null| Status: UNKNOWN
My ID: 2| ID received: 12| Status: UNKNOWN
My ID: 6| ID received: null| Status: UNKNOWN
My ID: 4| ID to Send: null| Status: UNKNOWN
My ID: 12| ID to Send: null| Status: UNKNOWN
My ID: 7| ID received: null| Status: UNKNOWN
My ID: 4| ID received: null| Status: UNKNOWN
My ID: 2| ID received: null| Status: UNKNOWN
My ID: 12| ID received: 12| Status: UNKNOWN
My ID: 6| ID received: null| Status: UNKNOWN
My ID: 7| ID to Send: null| Status: UNKNOWN
My ID: 2| ID to Send: null| Status: UNKNOWN
leader elected
The id is 12
The number of messages sent is 12
java.net.SocketException: Connection reset
```

Figure 2: Results from the Client

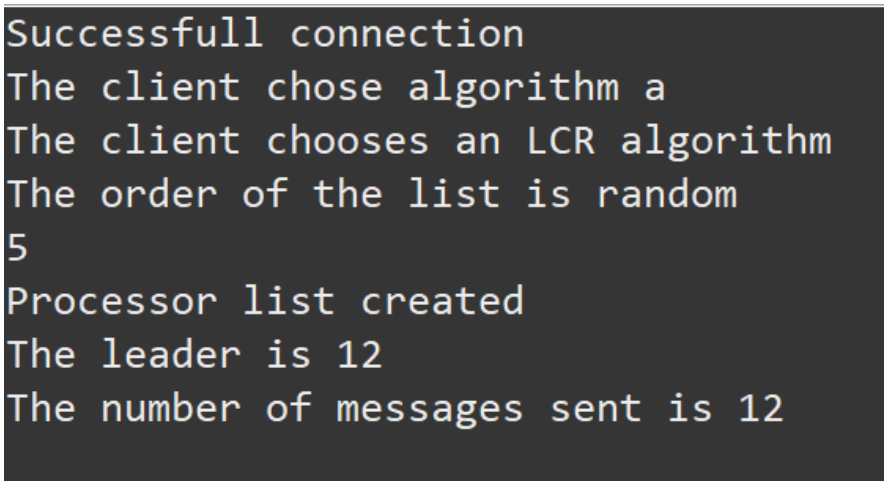

Figure 3: Results from the Server

The server only prints out the most important information and not all the results and process as in the client side.

Conclusion:

To conclude, the connection between the server and the client is successful on a local port. It has not been tested on different computers therefore, no data has been collected for that part. The client can implement the leader election and see all the results and the process behind. The leader election part is correct since it follows the same algorithm as in coursework 1 and has the same number of rounds.