**COIT20257: Distributed Systems: Principles and Development (T1, 2020)**

**Peer-to-Peer Content Distribution and Distributed Query**

**Assessment – 2**

**Lecturer / Tutor**

**Partha Gangavalli**

**Prepared By**

**Name: Aman Kumar Maharjan**

**SID: 12102452**

**Campus: Sydney**

**Due Date: Friday, 22 May 2020, 11:55 PM**

**Contents**

[Program Compilation and Installation: 3](#_Toc41048114)

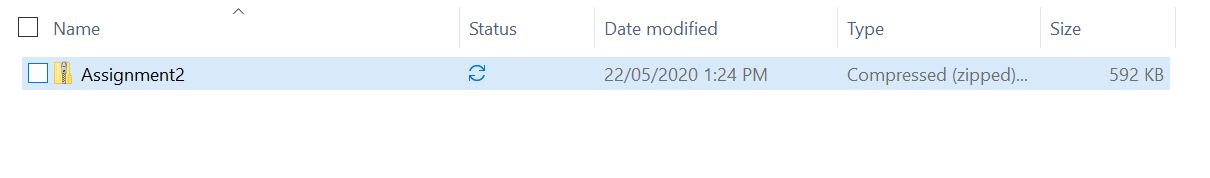
[Running the application 4](#_Toc41048115)

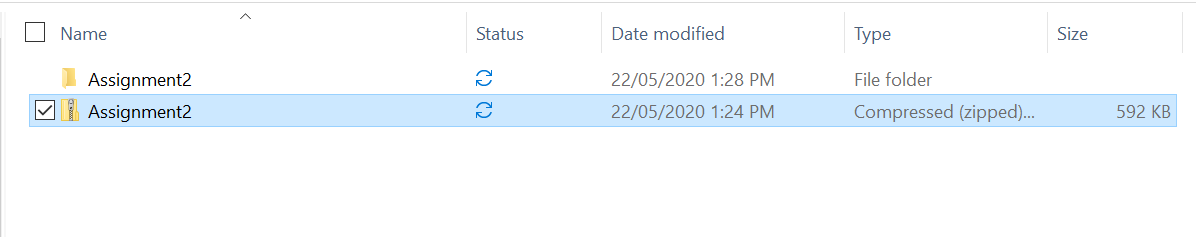
[Test instruction 6](#_Toc41048116)

[key potential problems with IP multicast 14](#_Toc41048117)

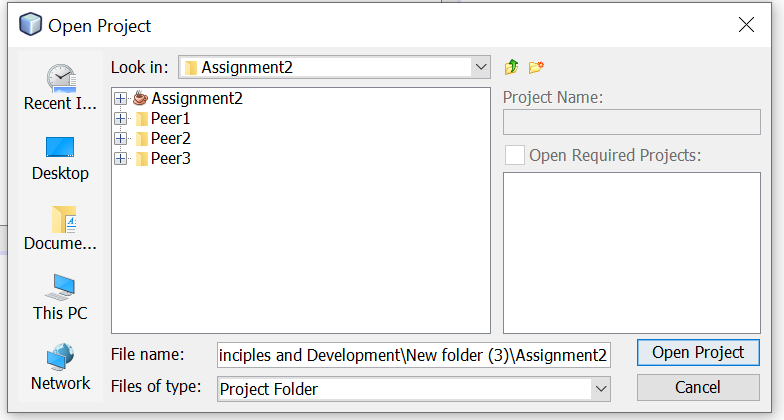
# Program Compilation and Installation:

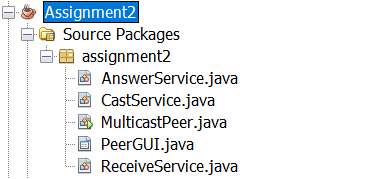
1. Unzip the Assignment2.zip file





1. Then, got to Assignment2 folder and open the assignment2 project through NetBeans

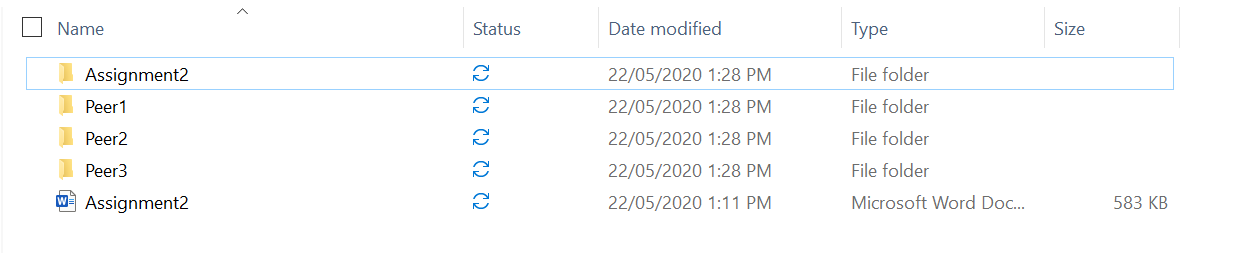




1. Run the MulticastPeer.java main class.

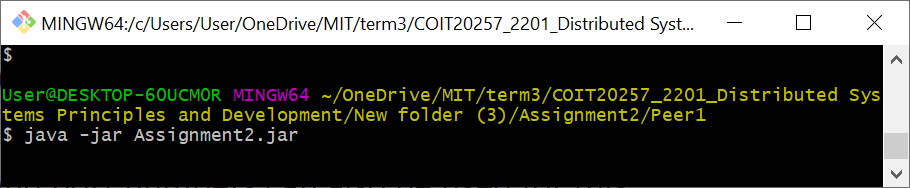
# Running the application

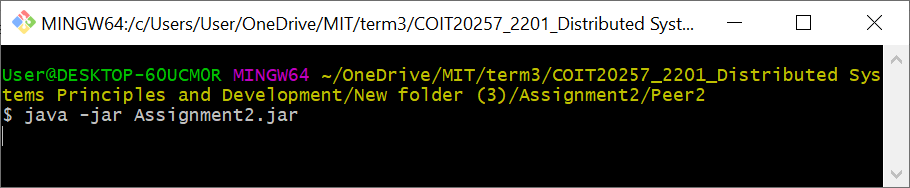
For running the different peers, inside Assignment folder, there are 3 folder peer1, peer2 and peer3 respectively, of which contains Assignment2.jar file.

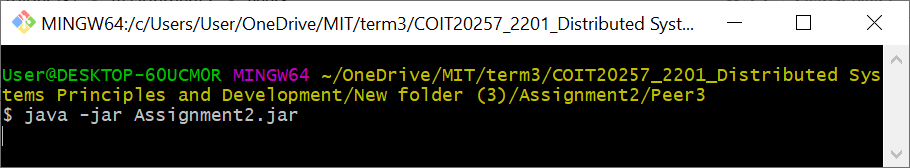


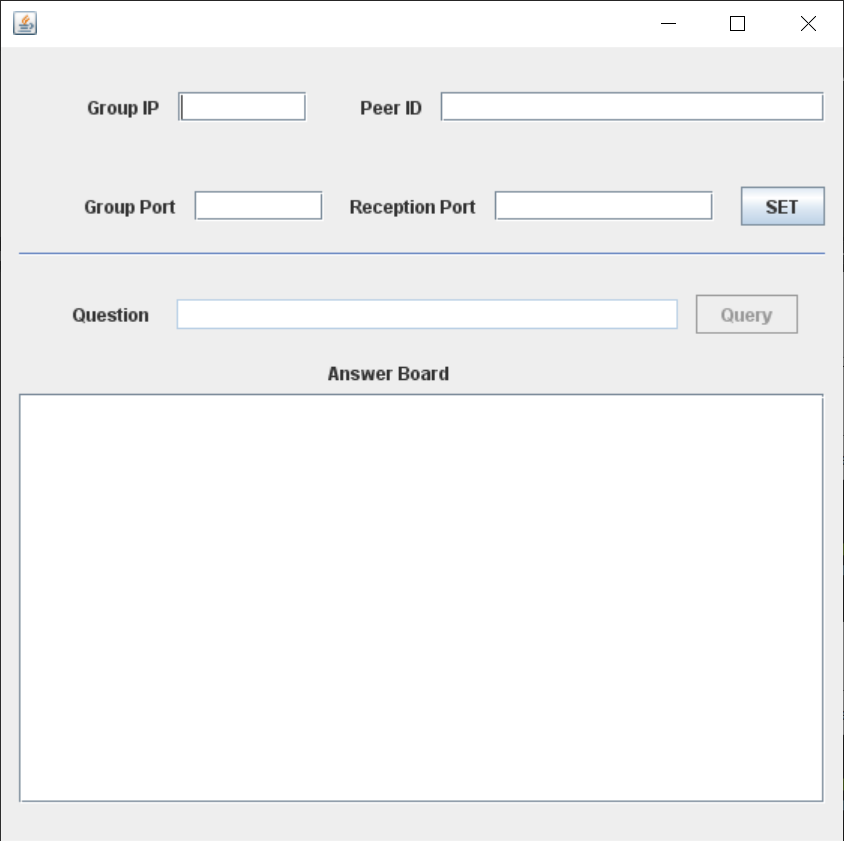
In each folder, from command prompt, run the following command

java -jar "Assignment2.jar"









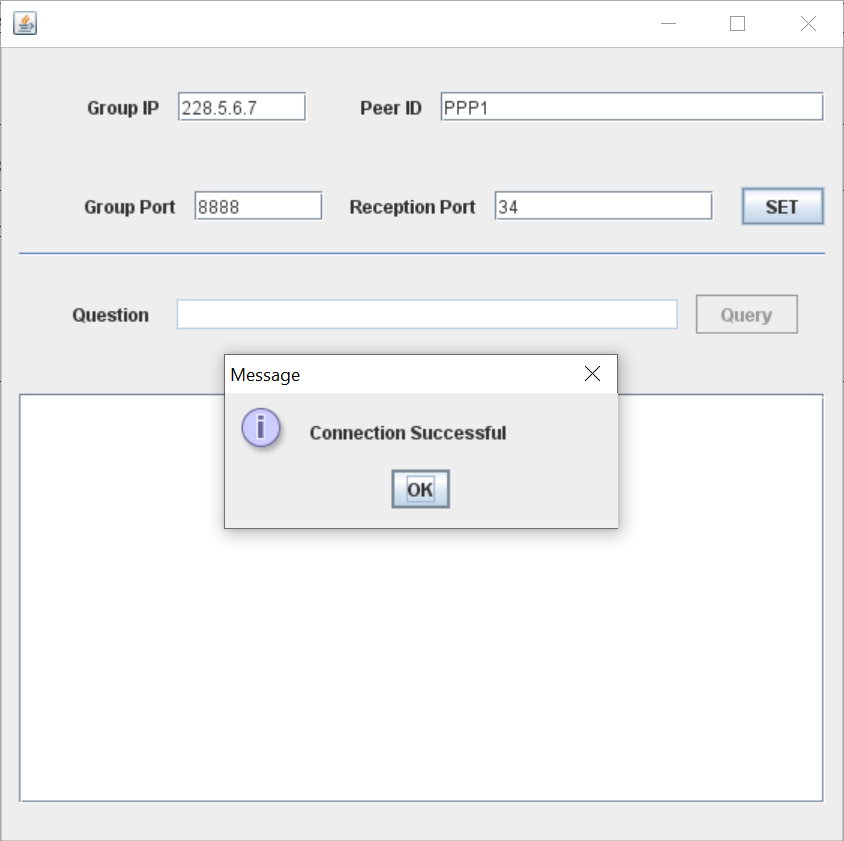
# Test instruction

The following test cases have been taken into consideration.

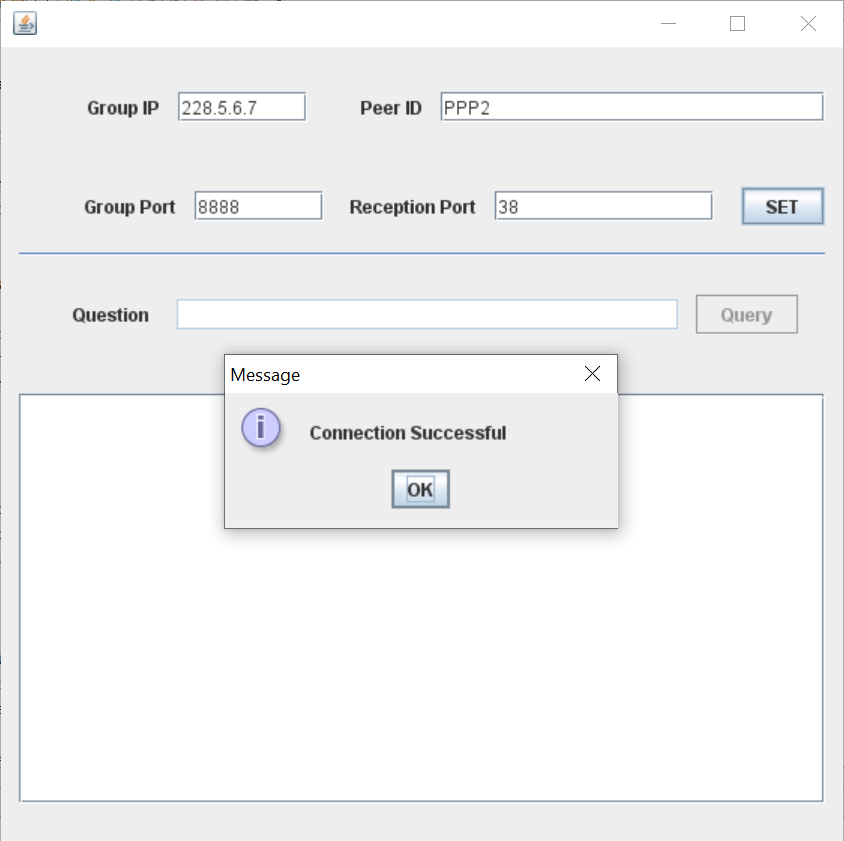
|  |  |
| --- | --- |
| **Test Plans** | |
| 1. | Set button: Run services |
| 2. | Query button: Submit query and display the answer in text field |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input Data** | **Expected Result** | **Actual Result** |
| Set button: | Set button pressed | Run answer service, cast service, receive service and enable the query text field and query button with connection successful message | Ran answer service, cast service, receive service and enabled the query text field and query button with connection successful message |

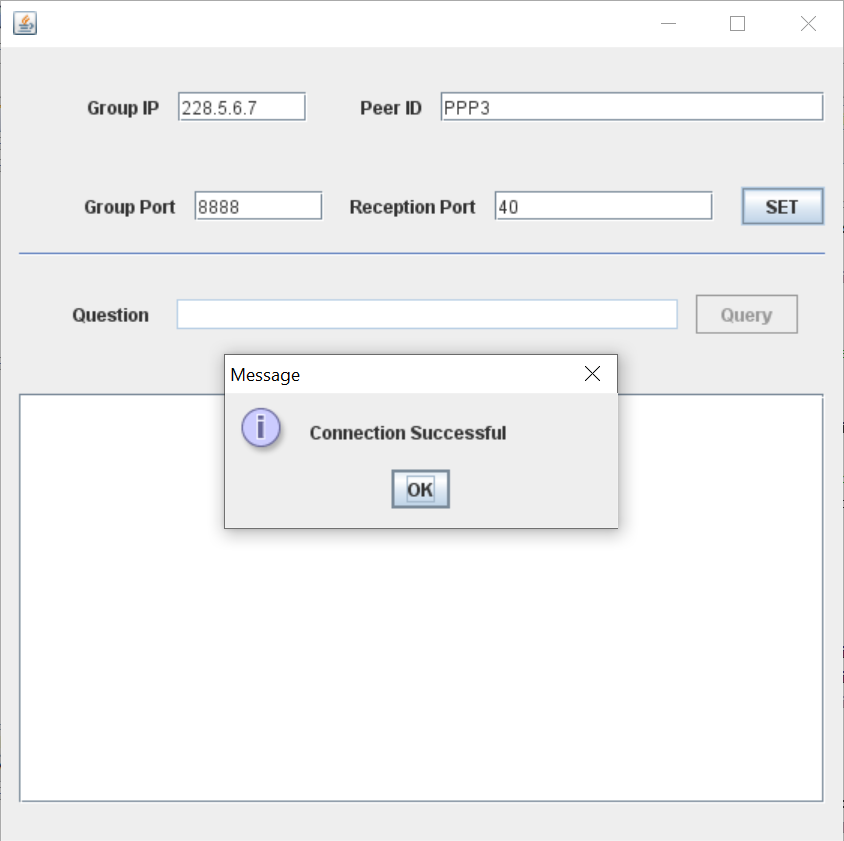
**Peer 1**



**Peer2**



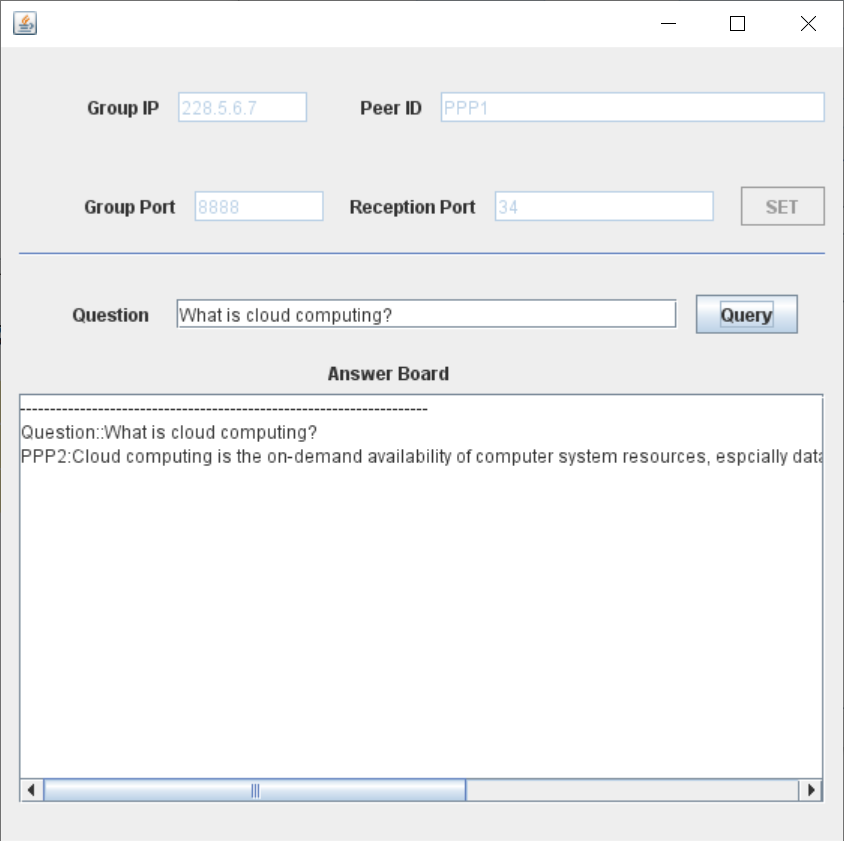
**Peer3**

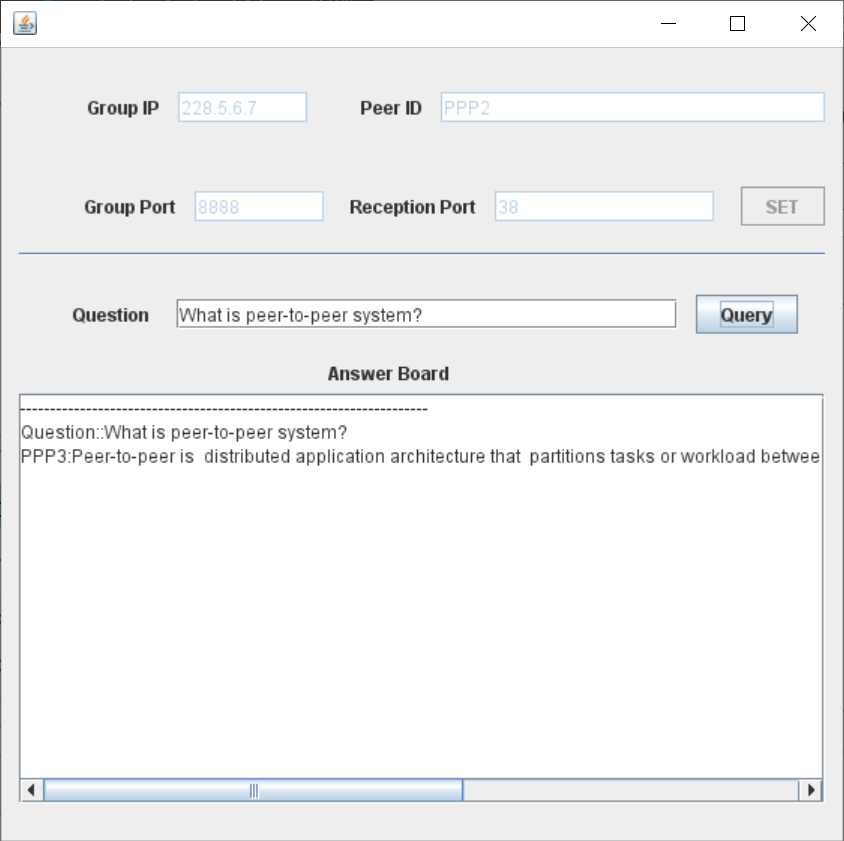


|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input Data** | **Expected Result** | **Actual Result** |
| Query button: Submit query and display the answer in text field. | Question in the Text field and Query button pressed | Display the question and answer to the questions in the text field | Displayed the question and answer to the questions in the text field |

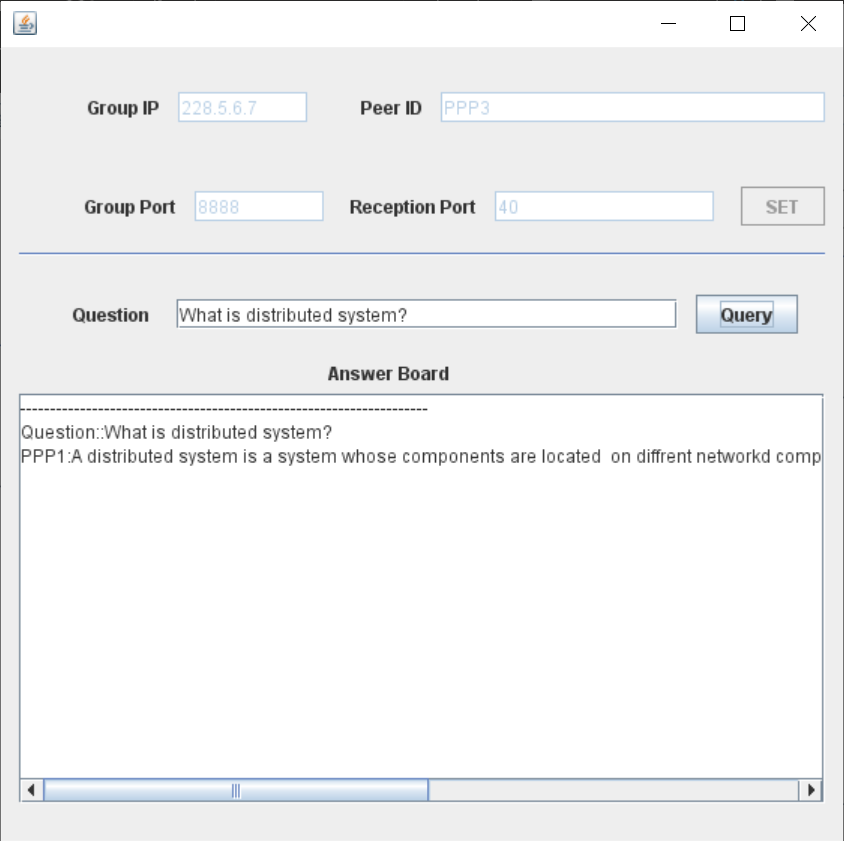
**Single Question Single answer testing**

PPP1 queries ‘What is cloud computing?’ Only PPP2 has the answer and replies.



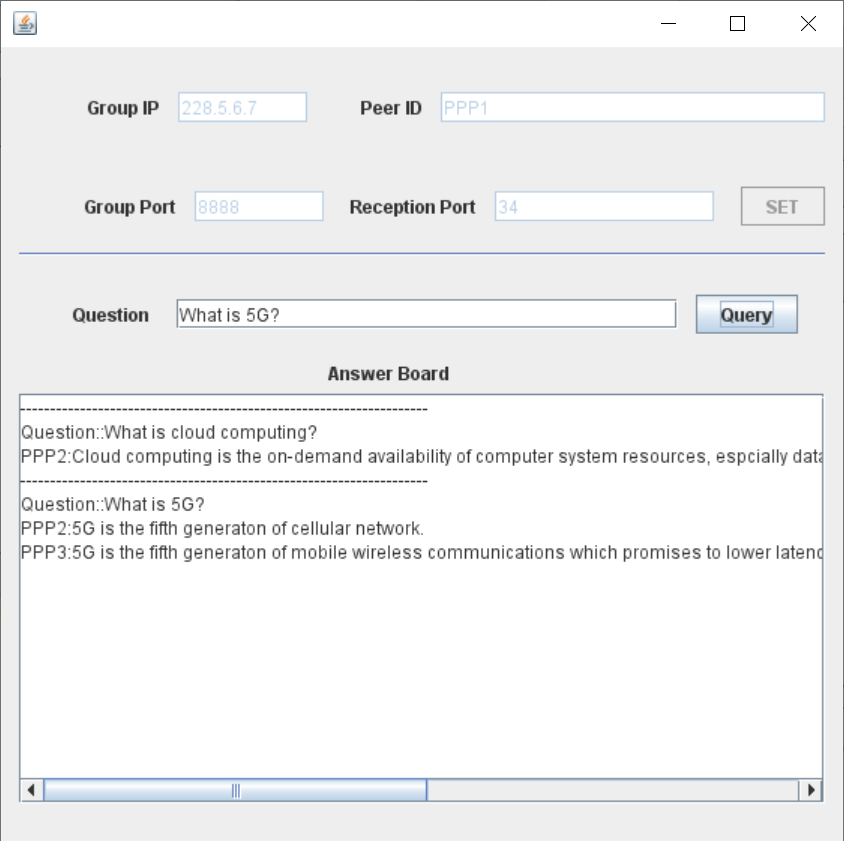
*PPP2* queries “What is peer-to-peer system?’ Only *PPP3* has the answer and replies. 

PPP3 queries “What is distributed system?’, Only PPP1 has the answer and replies.

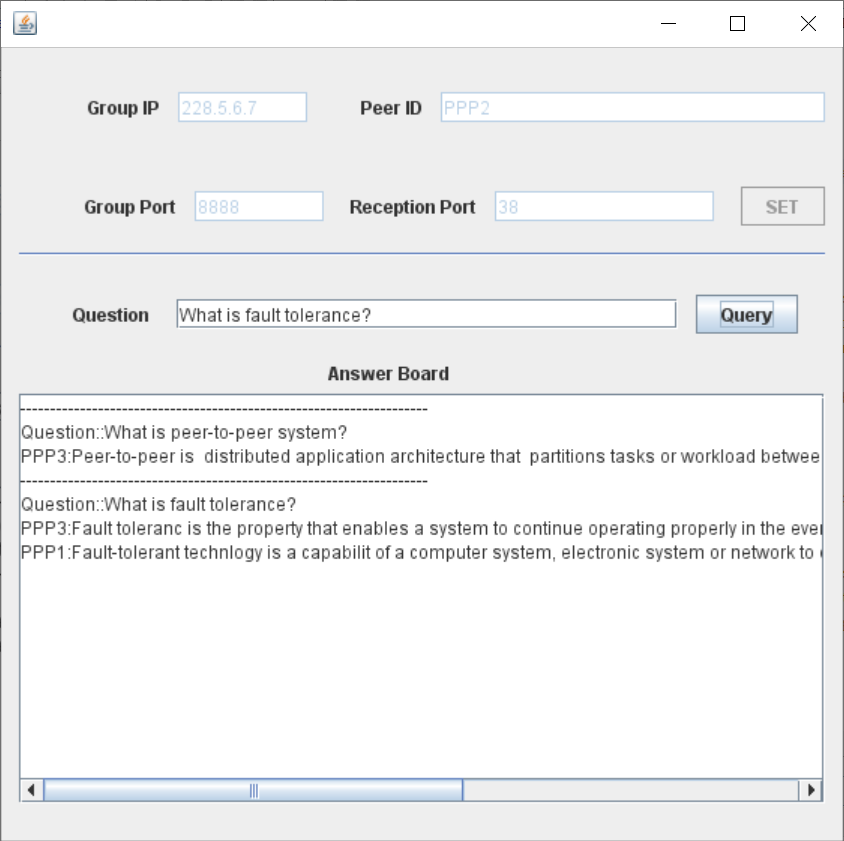


**Single question Multiple answers**

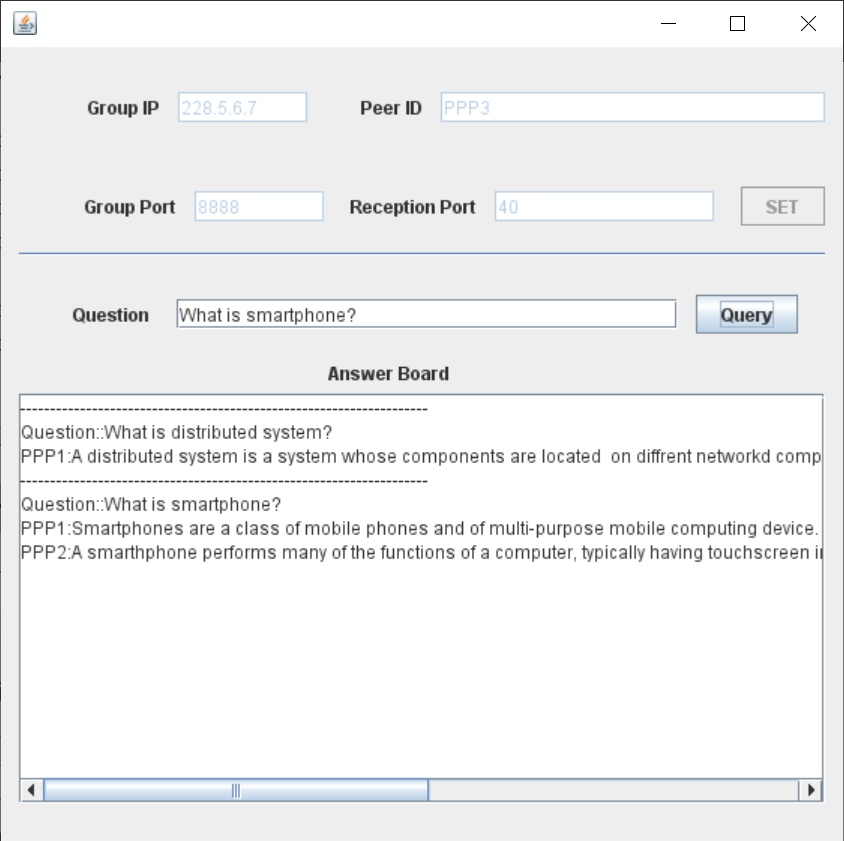
*PPP1* queries ‘What is 5G?’, *PPP2* and *PPP3* have answers and reply.



PPP2 queries ‘What is fault tolerance?’, PPP1 and PPP3 have answers and reply.



PPP3 queries ‘What is smartphone?’, PPP1 and PPP2 have answers and reply



# key potential problems with IP multicast

**Mutual Exclusion:** By applying the multiples thread like answer service, cast service and the receive service in the program, the problem of mutual exclusion has been resolved for synchronized operations.

**Ignore-self Message:** Since, each peer broadcast the question to all the peer in overlay multicast address, message is received by sending peer as well. So, the message has been parsed to check the peered to ignore the message.