Application 5 : Relay based Peer-to-Peer System using Client-Server socket programming

Group Number- 15

~Sagar Kesarwani 214101044
~Sagar Kumar 214101045
~Sama Rohith Reddy 214101046

Phase 1:

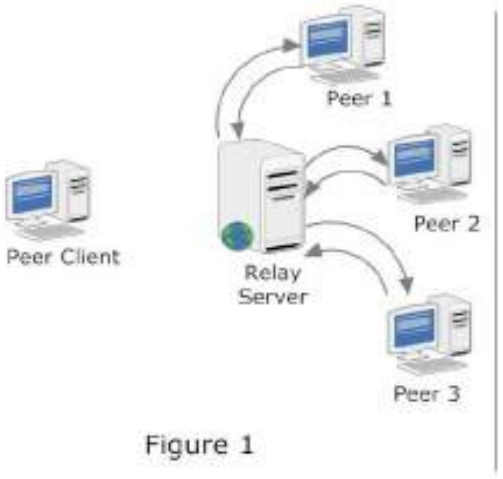


Figure 1

Usage of server.c :

Compilation: gcc server.c -o server
Running: ./server <Server port number>
(Port number should be > 1023, port numbers 0 to 1023 are reserved)

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15$ ./server 1700
Server established with server port number = 1700
```

**This denotes that the server is now open with a given port number.

Usage of nodes.c :

Compilation: gcc nodes.c -o nodes
Running: ./nodes <Server IP Address> <Server port number> <Peer port number>
(Port number should be > 1023, port numbers 0 to 1023 are reserved)
IP address can used by using command "ifconfig"

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode1$ ./node 127.0.0.1 1600 1801
Server says: Hi there! This is the server.
Port number of peer node: 1801
```

**Peer Node 1 connected to the relay server

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode2$ ./node 127.0.0.1 1600 1802
Server says: Hi there! This is the server.
Port number of peer node: 1802
```

**Peer Node 2 connected to the relay server

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode3$ ./node 127.0.0.1 1600 1803
Server says: Hi there! This is the server.
Port number of peer node: 1803
```

**Peer Node 3 connected to the relay server

The Relay_Server actively maintains all the received information with it.

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15$ ./server 1600
Server established with server port number = 1600

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1801
Peer node IP: 127.0.0.1

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1802
Peer node IP: 127.0.0.1

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1803
Peer node IP: 127.0.0.1
```

Phase 2:

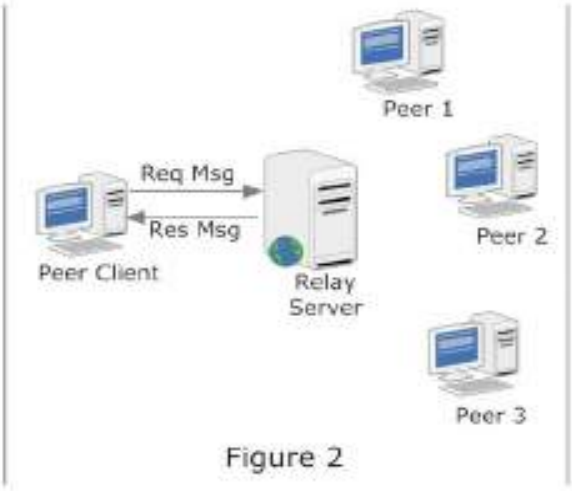


Figure 2

Usage of client.c :

Compilation: gcc client.c -o client

Running: ./client <Server IP address> <Server port number>

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15$ ./server 1600
Server established with server port number = 1600

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1801
Peer node IP: 127.0.0.1

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1802
Peer node IP: 127.0.0.1

Connection accepted
Message: Hi there! This is peer node.
Peer node port: 1803
Peer node IP: 127.0.0.1

Connection accepted
Message: Hi there! This is the client.
Peer client port: 54964
Peer client IP: 127.0.0.1
Number of peer nodes = 3
```

Peer_Client connects with Relay_Server and requests it for active Peer_Nodes information.
The Relay_Server responses to the Peer_Client with the active Peer_Nodes information
currently having it. On receiving the response message from the Relay_Server, the Peer_Client closes the connection gracefully.

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15$ ./client 127.0.0.1 1600
Server says: Hi there! This is the server.

PeerNode Count : 3

PeerNode IP: 127.0.0.1
PeerNode port: 1801
PeerNode IP: 127.0.0.1
PeerNode port: 1802
PeerNode IP: 127.0.0.1
PeerNode port: 1803
```

Phase 3:

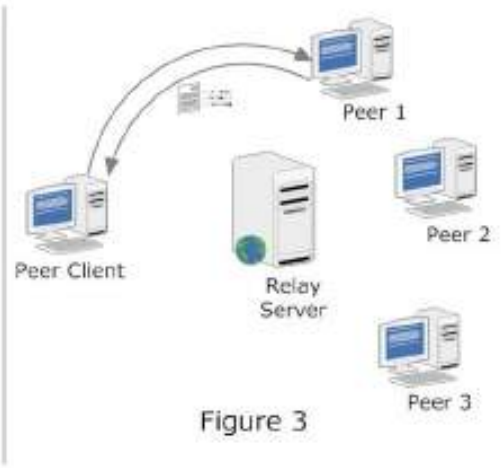
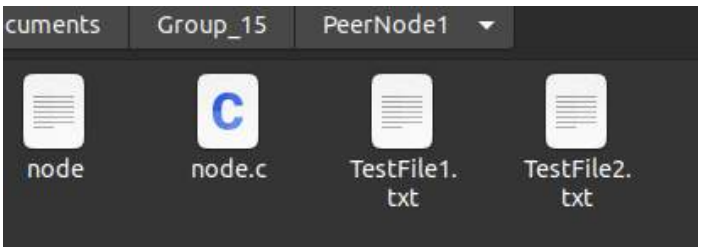


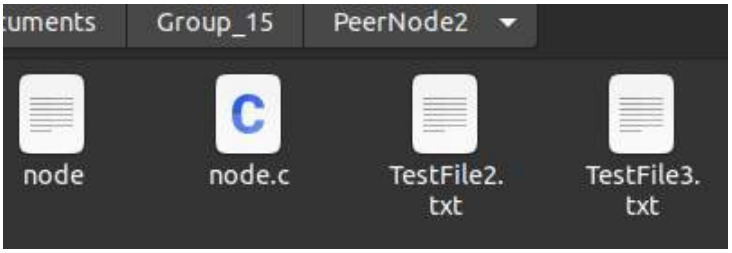
Figure 3

The distribution of test files is as follows:

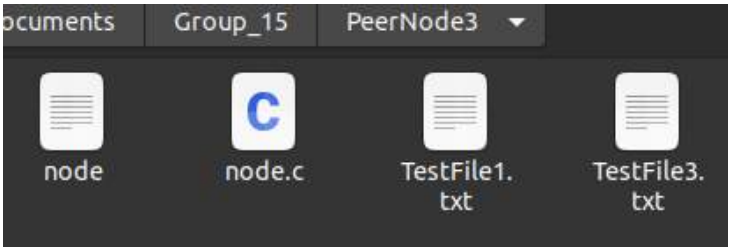
Peer node 1 contains TestFile1.txt and TextFile2.txt



Peer node 2 contains TestFile2.txt and TextFile3.txt



Peer node 3 contains TestFile1.txt and TextFile3.txt

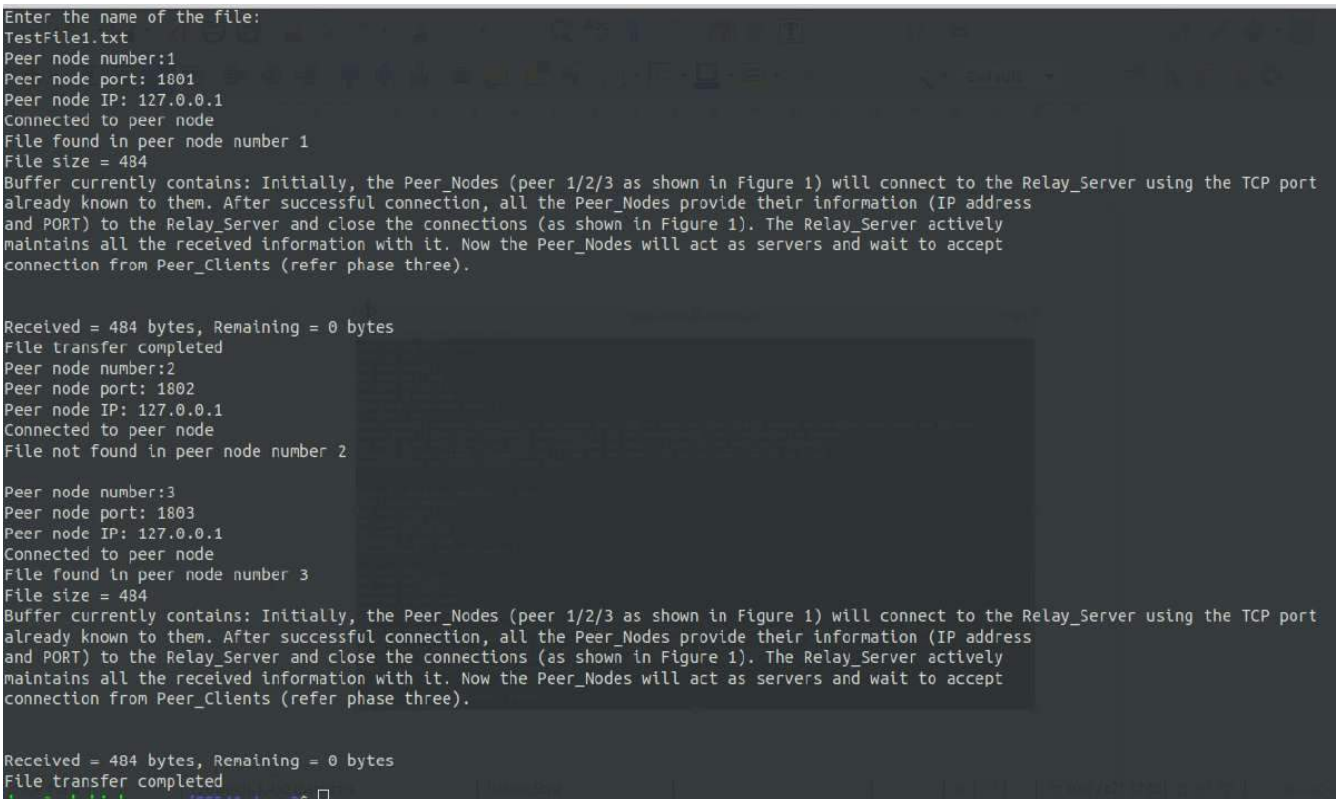


On requesting TestFile1.txt, it is found in peer nodes 1 and 3 and they transfer the data back as shown:

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode1$ ./node 127.0.0.1 1600 1801
Server says: Hi there! This is the server.
Port number of peer node: 1801
client server is requesting for file with filename = TestFile1.txt
File size: 484 bytes
1. Server sent 484 bytes from file's data, offset is now : 484 and remaining data = 0
File transfer completed
```

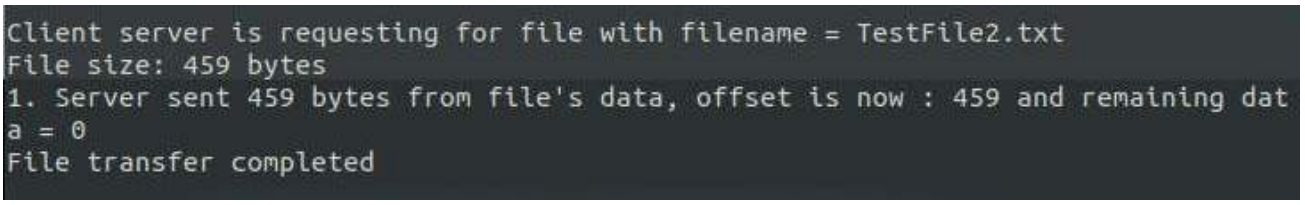
```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode2$ ./node 127.0.0.1 1600 1802
Server says: Hi there! This is the server.
Port number of peer node: 1802
client server is requesting for file with filename = TestFile1.txt
File not found
```

```
sagar@sagar-HP-Pavilion-15-Notebook-PC:~/Documents/Group_15/PeerNode3$ ./node 127.0.0.1 1600 1803
Server says: Hi there! This is the server.
Port number of peer node: 1803
client server is requesting for file with filename = TestFile1.txt
File size: 484 bytes
1. Server sent 484 bytes from file's data, offset is now : 484 and remaining data = 0
File transfer completed
```

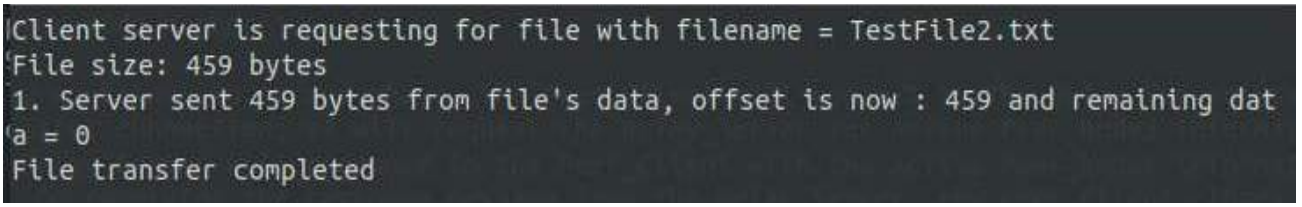


On requesting TestFile2.txt, it is found in peer nodes 1 and 2 and they transfer the data back as shown:

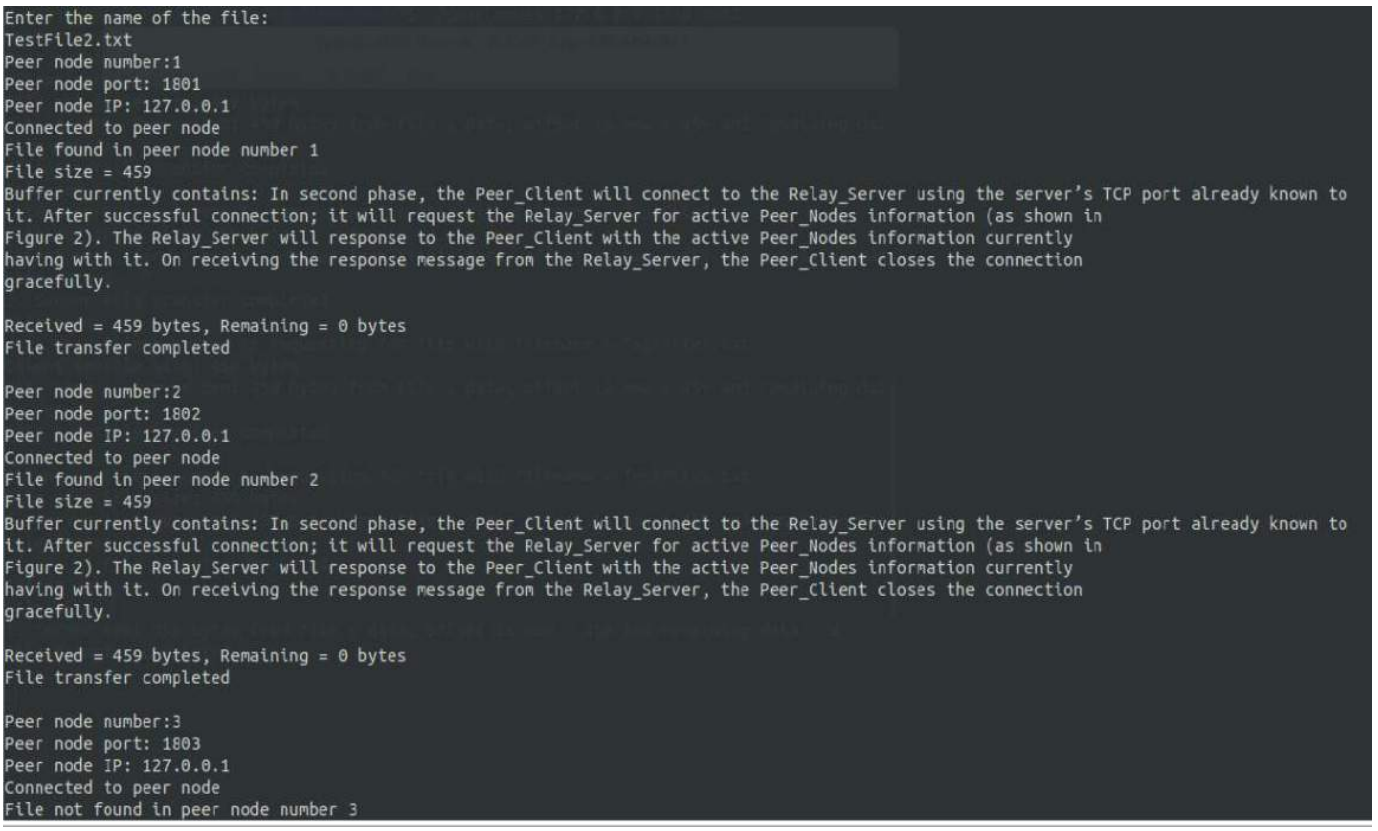
Peernode-1:



Peernode-2:



Peernode-3:



If requested file is not present in any peer node: