Output

Buffer size = 4 out of Bucket size = 10
Buffer size = 7 out of Bucket size = 10
Buffer size = 10 out of Bucket size = 10
Packet Loss = 4
Buffer size = 9 out of bucket size = 10

Socket Programming : using TCP sockets, write client server program make client send file name & server send back the contents of requested file if present.

[client.py]

```
From socket import *
serverName = "DESKTOP-HMPOOFC"
serverport = 12550
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket = connect((serverName, serverPort))
Sentance = input("Enter Filename");
client.socket.send(sentence.encode())
file contents = clientsocket.recv(1024).decode()
print('From Server:', file content)
client socket.close().
```

```
From socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INFP, SOCK_DGRAM)
Sentance = input("Enter file name")
print("From server:", file contents)
client Socket.close().
```

[servudp.py]

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    file = open(sentence, "r")
    l = file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"), clientAddress)
print("sent back to client", l)
file.close()
```

```
from socket import *
serverName = "DESKTOP-HMPODEC"
serverport = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(("", serverport))
serverSocket.listen(1)
print("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(2048).decode()
    file = open(sentence, "r")
    l = file.read(2048)
    connectionSocket.send(l.encode())
file.close()
connectionSocket.close()
```

Output:

Server started
waiting for a client
connected
Server started
Waiting for a client
Client Accepted.

1/3/2023