

## TASK 1.

### UDPPingerServer:

```
# UDPPingerServer.py
# We will need the following module to generate randomized
# lost packets
import random
import socket

# Create a UDP socket
# Notice the use of SOCK_DGRAM for UDP packets
SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)
serverSocket = socket.socket(socket.AF_INET,
                              socket.SOCK_DGRAM)

# Assign IP address and port number to socket

serverSocket.bind(ADDR)
print("UDP server up and listening")

while True:
    # Generate random number in the range of 0 to 10
    rand = random.randint(0, 10)

    # Receive the client packet along with the address it is
    # coming from
    message, address = serverSocket.recvfrom(HEADER)
    print(f"Message from Client :
    {message}          rand_value : {rand}")

    # Capitalize the message from the client
    message = message.upper()

    # If rand is less is than 4, we consider the packet lost
    # and do not respond
```

```
if rand < 4:
    continue

# Otherwise, the server responds
serverSocket.sendto(message, address)
```

#### UDPclient:

```
import socket
import time

SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)

# Create a UDP socket at client side
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client.settimeout(1)

# Send to server using created UDP socket
rtts = []
packet_loss_rate = 0

for i in range(1, 11):

    send_time = time.time()
    msgFromClient = f"PING sequence:{i} {send_time}"
    bytesToSend = str.encode(msgFromClient)
    client.sendto(bytesToSend, ADDR)

    try:
        msgFromServer = client.recvfrom(HEADER)
        recv_time = time.time()
        rtts.append(recv_time-send_time)
```

```

        msg = f"Message from Server
{msgFromServer[0]}          Round-trip time : {recv_time-
send_time}"
        print(msg)

    except TimeoutError:
        packet_loss_rate = packet_loss_rate + 1
        print("Request timed out!")

print("")
print(f"Maximum Round trip time : {max(rtts)}")
print(f"Minimum Round trip time : {min(rtts)}")
print(f"Average Round trip time : {sum(rtts)/len(rtts)}")
print(f"Packet loss rate percentage :
{(100*packet_loss_rate)/10}%")

```

The screenshot shows a code editor with several tabs: UDPpingServer.py, server.py, client.py (active), heartbeatServer.py, and heartbeatClient.py. The active tab, client.py, contains the following code:

```

30
31 except TimeoutError:
32     packet_loss_rate = packet_loss_rate + 1
33     print("Request timed out!")
34

```

Below the code editor is a terminal window titled "powershell". It shows the command to run the script and its output:

```

PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming> python -u "c:\Users\Beyond Adil\OneDrive
\Documents\Computer Networking- Sem IV\Socket Programming\UDP\Lab3\client.py"
Request timed out!
Message from Server b'PING SEQUENCE:2 1675421958.6595666'      Round-trip time : 0.0010085105895996094
Message from Server b'PING SEQUENCE:3 1675421958.6605752'      Round-trip time : 0.0
Request timed out!
Message from Server b'PING SEQUENCE:5 1675421959.6728196'      Round-trip time : 0.0010232925415039062
Message from Server b'PING SEQUENCE:6 1675421959.673843'        Round-trip time : 0.0
Request timed out!
Request timed out!
Request timed out!
Message from Server b'PING SEQUENCE:10 1675421962.7080264'      Round-trip time : 0.0010442733764648438

Maximum Round trip time : 0.0010442733764648438
Minimum Round trip time : 0.0
Average Round trip time : 0.0006152153015136719
Packet loss rate percentage : 50.0%
PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming>

```

The screenshot shows a code editor with several files open: UDPPingerServer.py, server.py, client.py (active), heartbeatServer.py, and heartbeatClient.py. The active file, client.py, contains the following code:

```
30
31 except TimeoutError:
32     packet_loss_rate = packet_loss_rate + 1
33     print("Request timed out!")
34
```

The terminal window at the bottom shows the command to run the script and its output:

```
PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming> python -u "c:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming\UDP\Lab3\UDPPingerServer.py"
UDP server up and listening
Message from Client : b'PING sequence:1 1675421957.6343713'      rand_value : 1
Message from Client : b'PING sequence:2 1675421958.6595666'      rand_value : 7
Message from Client : b'PING sequence:3 1675421958.6605752'      rand_value : 8
Message from Client : b'PING sequence:4 1675421958.6615748'      rand_value : 0
Message from Client : b'PING sequence:5 1675421959.6728196'      rand_value : 7
Message from Client : b'PING sequence:6 1675421959.673843'       rand_value : 6
Message from Client : b'PING sequence:7 1675421959.6748447'      rand_value : 1
Message from Client : b'PING sequence:8 1675421960.686841'       rand_value : 3
Message from Client : b'PING sequence:9 1675421961.69946'        rand_value : 1
Message from Client : b'PING sequence:10 1675421962.7080264'     rand_value : 4
```

## TASK 2:

Server:

```
import socket
SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)

msgFromServer = "Hello UDP Client"
bytesToSend = str.encode(msgFromServer)

# Create a datagram socket
server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# Bind to address and ip
server.bind(ADDR)
print("UDP server up and listening")

# Listen for incoming datagrams
while (True):

    bytesAddressPair = server.recvfrom(HEADER)
    message = bytesAddressPair[0]
```

```

address = bytesAddressPair[1]

clientMsg = "Message from Client:{}".format(message)
clientIP = "Client IP Address:{}".format(address)
print(clientMsg)
print(clientIP)

# Sending a reply to client
server.sendto(bytesToSend, address)

```

Client:

```

import socket
import time

SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)

# Create a UDP socket at client side
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client.settimeout(1)

# Send to server using created UDP socket
rtts = []
packet_loss_rate = 0

for i in range(1, 11):

    send_time = time.time()
    msgFromClient = f"PING sequence:{i} {send_time}"
    bytesToSend = str.encode(msgFromClient)
    client.sendto(bytesToSend, ADDR)

    try:

```

```

msgFromServer = client.recvfrom(HEADER)
recv_time = time.time()
rtts.append(recv_time-send_time)
msg = f"Message from Server
{msgFromServer[0]}      Round-trip time : {recv_time-
send_time}"
print(msg)

except TimeoutError:
    packet_loss_rate = packet_loss_rate + 1
    print("Request timed out!")

print("")
print(f"Maximum Round trip time : {max(rtts)}")
print(f"Minimum Round trip time : {min(rtts)}")
print(f"Average Round trip time : {sum(rtts)/len(rtts)}")
print(f"Packet loss rate percentage :
{(100*packet_loss_rate)/10}%")

```

The screenshot shows a VS Code editor with a Python project. The file explorer on the left shows a folder named 'Sem IV\Socket Programming' containing files 'server.py', 'client.py', 'heartbeatServer.py', and 'heartbeatClient.py'. The 'server.py' file is open in the editor, showing the following code:

```

11 server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
12 # Bind to address and ip
13 server.bind(ADDR)
14 print("UDP server up and listening")
15

```

The terminal window at the bottom shows the output of running the server and client. The server output is as follows:

```

Sem IV\Socket Programming> python -u "c:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming\UDP\Lab3\server.py"
UDP server up and listening
Message from Client:b'PING sequence:1 1675422282.5068533'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:2 1675422282.5088415'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:3 1675422282.5098445'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:4 1675422282.510843'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:5 1675422282.510843'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:6 1675422282.512212'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:7 1675422282.512212'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:8 1675422282.512212'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:9 1675422282.5132308'
Client IP Address:('172.27.37.21', 64757)
Message from Client:b'PING sequence:10 1675422282.5132308'
Client IP Address:('172.27.37.21', 64757)

```

The client output is as follows:

```

Message from Server b'Hello UDP Client'      Round-trip time : 0.0010030269622802734
Message from Server b'Hello UDP Client'      Round-trip time : 0.0009984970092773438
Message from Server b'Hello UDP Client'      Round-trip time : 0.0
Message from Server b'Hello UDP Client'      Round-trip time : 0.0009987354278564453
Message from Server b'Hello UDP Client'      Round-trip time : 0.0
Message from Server b'Hello UDP Client'      Round-trip time : 0.0
Message from Server b'Hello UDP Client'      Round-trip time : 0.0010187625885009766
Message from Server b'Hello UDP Client'      Round-trip time : 0.0
Message from Server b'Hello UDP Client'      Round-trip time : 0.0009987354278564453
Maximum Round trip time : 0.0019881725311279297
Minimum Round trip time : 0.0
Average Round trip time : 0.0007005929946899414
Packet loss rate percentage : 0.0%
PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming>

```

### TASK 3:

heartbeatServer:

```
import random
import time
import socket

# Create a UDP socket
SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)
serverSocket = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
serverSocket.bind(ADDR)
print("UDP server up and listening")

while True:
    rand = random.randint(0, 2)
    recv_time = time.time()
    message, address = serverSocket.recvfrom(HEADER)
    print(f"Message from Client : {message}")

    send_time = float(message)
    time_diff = (recv_time - send_time) + rand

    if time_diff > 1:
        print(f"Message from Client : {message}      RTT :
{time_diff-1}      Client stopped running!")
        break

    serverSocket.sendto(message, address)
```

heartbeatClient:

```
import socket
import time

SERVER = socket.gethostbyname(socket.gethostname())
PORT = 12000
HEADER = 1024
ADDR = (SERVER, PORT)

# Create a UDP socket at client side
client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
client.settimeout(1)

# Send to server using created UDP socket
rtts = []
packet_loss_rate = 0

for i in range(1, 11):

    send_time = time.time()
    msgFromClient = f"{send_time}"
    bytesToSend = str.encode(msgFromClient)
    client.sendto(bytesToSend, ADDR)

    try:
        msgFromServer = client.recvfrom(HEADER)
        recv_time = time.time()
        rtts.append(recv_time-send_time)
        msg = f"Message from Server {msgFromServer[0]} Round trip time : {recv_time-send_time}"
        print(msg)

    except TimeoutError:
        packet_loss_rate = packet_loss_rate + 1
        print("Application Stopped Working!")
        break
```



UDPPingerServer.pyserver.pyclient.pyheartbeatServer.pyheartbeatClient.py X

UDP > Lab3 > heartbeatClient.py

```
15 packet_loss_rate = 0
16
17 for i in range(1, 11):
18
19     send_time = time.time()
```

PROBLEMSOUTPUTDEBUG CONSOLETERMINALCOMMENTS

PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming> python -u "c:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming\UDP\Lab3\heartbeatServer.py"

UDP server up and listening  
Message from Client : b'1675422518.3219357'  
Message from Client : b'1675422518.3239357'  
Message from Client : b'1675422518.3239357'  
Message from Client : b'1675422518.3249435'  
Message from Client : b'1675422518.3249435' RTT : 0.9989922046661377 Client stopped running!

PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming>

PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming> python -u "c:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming\UDP\Lab3\heartbeatClient.py"

Message from Server b'1675422518.3219357' Round trip time : 0.0010013580322265625  
Message from Server b'1675422518.3239357' Round trip time : 0.0  
Message from Server b'1675422518.3239357' Round trip time : 0.0  
Application Stopped Working!

PS C:\Users\Beyond Adil\OneDrive\Documents\Computer Networking- Sem IV\Socket Programming>