Brian Lucero

Part 1 – UDP Pinger with No Delay and No Loss.

---

(1) Describe the operation of your UDP Pinger, for example how it works.

This project utilizes a UDP pinger, a tool designed to send a UDP packet to a target on an unallocated port and waits for a specific error answer. The most basic and correct way to retrieve data is targeting as many devices on the target network as we can.The Python Socket library has very straightforward functions in order to set up a socket, which acts as an endpoint in 2 way communication over a network: the socket() function returns a socket object whose methods implement the various socket system calls. So, basically:

UDP sender: Send the host name and system time as a message to a destination host designated by its IP address and port number. Repeat the transmission when enabled to do so.Read and display the returned messages, and indicate the IP address and port number of the sender. Also, will keep track of the RTT values of each ping and give a stats summary at the end, also indicating if there's packet loss.

UDP receiver: Listens for messages on a user-defined port. Echo any received messages back to the sender. Display all IP addresses known to the host.

(2) Explain how to specify the timeout value for a datagram socket. Provide an example.

After the Socket has already been made and the client attempts to send a message out to server then a timeout of the socket can be made with the following code line:

```
timeout = 1

clientSocket.settimeout(timeout)
```

We can see that if a response is not sent back in 1 second, then it will timeout.

(3) Explain how to run your code, i.e., command line and any applicable parameter(s)

Brian Lucero

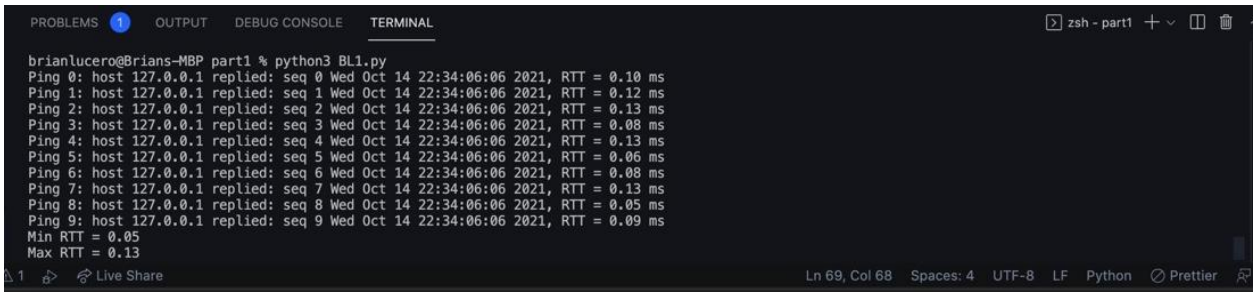
So, in order to run my code, you must have python 3.9.7 installed. Then navigate to the directory containing the udppingserver_no_loss.py and BL1.py file. Type the following command:

```
python3 udppingserver_no_loss.py
```

Then open another terminal window, and navigate to the same directory. Then type the following command:

```
python3 BL1.py
```

a. Include run-time screen captures for a sequence consists of 10 pings



(4) IncludeyourPythoncodelisting:

a. Include as text the listing of your Python code.

```
# Brian Lucero
## CPSC 471 Computer Communications
## Project2 - Part 1
## clientcode.py

import socket
from time import *
import sys

######## socket variables
host = 'localhost' # set to server ip or hostname
port = 12000
serverAddress = (host, port)

######## ping variables
number_of_pings = 10
timeout = 1 # 1 second = max amount of time to make client wait, any longer and it times out
sleep_time = 0

######## message size
message_bytes = 256
#byteTracker = bytearray([1] * message_bytes)
```

Brian Lucero

```
######## ping stats variables
min_ping = 999999
max_ping = 0
ping_count = 0
ping_received = 0
avg_ping = 0

######## RTT variables
minRTT = 0
maxRTT = 0
avgRTT = 0

######## send out msg to server
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
######## set timeout limit (1 sec)
clientSocket.settimeout(timeout)

####### stats summary
def summary():
   ######## get end time of ping loop when summary function is called (after 10 pings OR num of pings after timeout)
   ######## subtract start time before ping loop from the end time after ping
   # total_time = (time.time() - time_start) * 1000

   ######## calculate stats
   packet_loss = (ping_count - ping_received) / ping_count * 100
   avgRTT = avg_ping / ping_count
   minRTT = min_ping
   maxRTT = max_ping

   ######## print out summary stats
   print("Min RTT = " + str('%.2f'%minRTT))
   print("Max RTT = " + str('%.2f'%maxRTT))
   print("Avg RTT = " + str('%.2f'%avgRTT))
   print("Packet Lost = " + str('%0.2f%%'%packet_loss))
   sys.exit()

####### get time before the seq of pings starts
time_start = time()

####### ping loop
for seq in range(number_of_pings):

   ######## set the message
   t = strftime("%H:%M:%S:%S", localtime())
   dt = "Wed Oct 14 " + t + " 2021"
   message = "seq " + str(seq) + " " + str(dt)# + str(byteTracker)

   ######## try to send & receive a message in <= 1 second
   try:
       ####### send message to server
       clientSocket.sendto(message.encode('utf-8'), serverAddress)
       ####### get start time
```

```
        start = time()
        ####### record response from server and address
        response, address = clientSocket.recvfrom(1024)
        ####### get end time
        end = time()

        ####### calculate stats
        ping = (end - start) * 1000
        if ping < min_ping: min_ping = ping
        if ping > max_ping: max_ping = ping

        ####### update ping counter & stats variables
        ping_count += 1
        ping_received += 1
        avg_ping += ping
        time_delay = ping - min_ping # keeps track of the jitter

        ####### remove byte tracker from response
        # bt = len(str(byteTracker))
        # r = response[:-bt]

        a = str(address)[2:]
        sa = a[:-9]
        b = str(response)[2:]
        res = b[:-1]

        ####### print out response
        print("Ping " + str(seq) + ":" + " host " + sa + " replied: " + res + ", RTT = " + str('%.2f'%ping) + " ms")
        # time.sleep(sleep_time) # can be adjusted to see the pings come in more slowly

    ####### if socket timesout, no stats will be calculated and ping is not lossless
    except socket.timeout as e:
        ####### time out message to inform client
        print('udp_seq=%d REQUEST TIMED OUT' % (seq))

####### display stats
summary()
```

Part 2: UDP Pinger with Delays

Delays

Brian Lucero

Our experiment so far has been on a local host running both server and client programs, and therefore we saw zero delays. In this portion of the project, you are asked to modify the server code to simulate random RTT delays ranging from 10ms to 40ms.

Hint: Create a variable which holds a randomized integer to determine the delay amount.

What to Hand in

PDF file report

Create a section called **Part 2 – UDP Pinger with Delays**. Include the followings:

1. (1) Describe the operation of your UDP Ping Server and explain how it simulates 10ms to 40ms RTT delays.

So basically, all I did was add the delay variable into the server file, which generates a random number between 10 and 40. Then I converted the number to fit the milliseconds unit format. Made the program wait the amount of milliseconds before sending back a response back to the client. Just like shown below:

```
####### simulate random rtt delay

delay = random.randint(10, 40)

ms = delay / 1000

time.sleep(ms)
```

2. (2) Explain how to run your code, i.e., command line and any applicable parameter(s)

So, in order to run my code, you must have python 3.9.7 installed. Then navigate to the directory containing the BL2.py and BL1.py file (original client file from part 1). Type the following command:

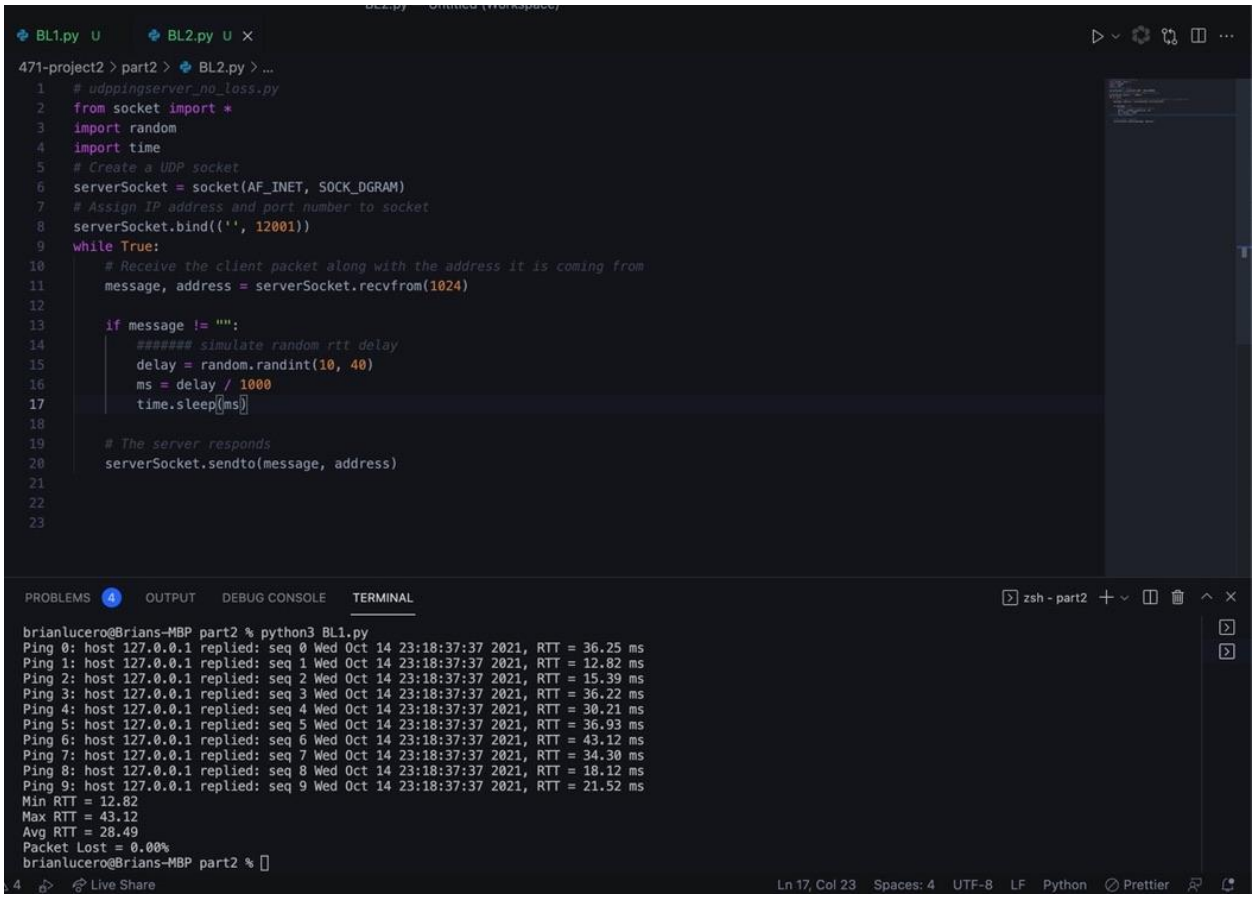
```
python3 BL2.py
```

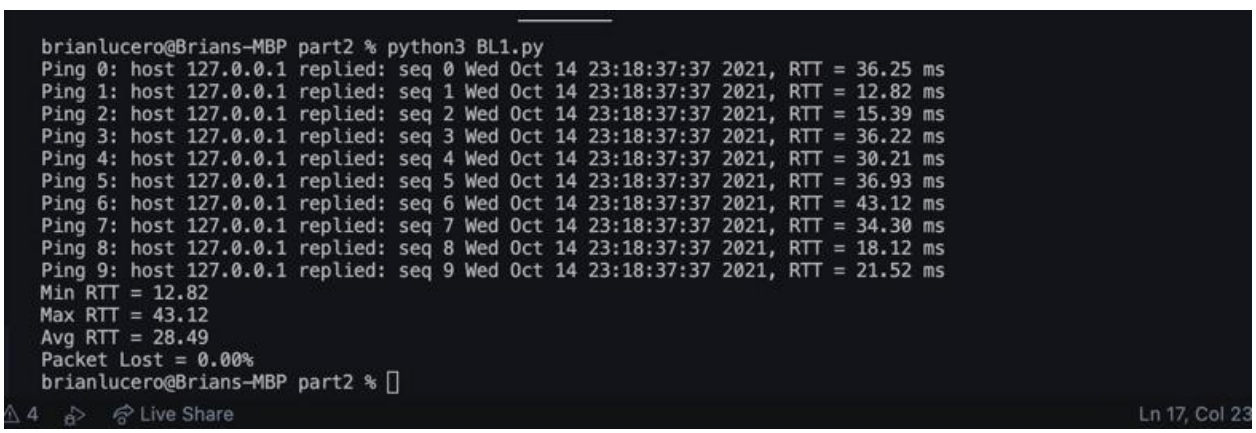
Then open another terminal window, and navigate to the same directory. Then, to run the client program again, type the following command:

```
python3 BL1.py
```

Brian Lucero

a. Include run-time screen captures for a sequence consists of 10 pings





(3) Include your Python code listing of your UDP Ping Server with 10ms to 40ms RTT delays: a. Include as text the listing of your Python code.

```
# udppingserver_no_loss.py

from socket import *
```

Brian Lucero

```
import random

import time

# Create a UDP socket

serverSocket = socket(AF_INET, SOCK_DGRAM)

# Assign IP address and port number to socket

serverSocket.bind(('', 12001))

while True:

   # Receive the client packet along with the address it is coming from

   message, address = serverSocket.recvfrom(1024)



   if message != "":

       ####### simulate random rtt delay

       delay = random.randint(10, 40)

       ms = delay / 1000

       time.sleep(ms)



   # The server responds

   serverSocket.sendto(message, address)
```

Brian Lucero

Part 3: UDP Pinger with Delays and Packet Losses

(1) Describe the operation of your UDP Ping Server and explain how it simulates delays be- tween 10ms and 40ms, with 20% packet losses.

(2) Explain how to run your code, i.e., command line and any applicable parameter(s)

a. Include run-time screen captures for a sequence consists of 100 pings

Brian Lucero

```
brianlucero@Brians-MBP part3 % python3 BL1.py
Ping 0: host 127.0.0.1 replied: seq 0 Wed Oct 14 23:50:50:50 2021, RTT = 36.78 ms
Ping 1: timed out, message was lost
Ping 2: host 127.0.0.1 replied: seq 1 Wed Oct 14 23:50:50:50 2021, RTT = 0.05 ms
Ping 3: host 127.0.0.1 replied: seq 2 Wed Oct 14 23:50:51:51 2021, RTT = 26.21 ms
Ping 4: timed out, message was lost
Ping 5: host 127.0.0.1 replied: seq 3 Wed Oct 14 23:50:51:51 2021, RTT = 0.04 ms
Ping 6: host 127.0.0.1 replied: seq 4 Wed Oct 14 23:50:51:51 2021, RTT = 23.02 ms
Ping 7: host 127.0.0.1 replied: seq 5 Wed Oct 14 23:50:52:52 2021, RTT = 12.71 ms
Ping 8: host 127.0.0.1 replied: seq 6 Wed Oct 14 23:50:52:52 2021, RTT = 17.35 ms
Ping 9: host 127.0.0.1 replied: seq 7 Wed Oct 14 23:50:52:52 2021, RTT = 32.42 ms
Ping 10: host 127.0.0.1 replied: seq 8 Wed Oct 14 23:50:52:52 2021, RTT = 29.45 ms
Ping 11: host 127.0.0.1 replied: seq 9 Wed Oct 14 23:50:52:52 2021, RTT = 35.56 ms
Ping 12: host 127.0.0.1 replied: seq 10 Wed Oct 14 23:50:52:52 2021, RTT = 999.69 ms
Ping 13: host 127.0.0.1 replied: seq 11 Wed Oct 14 23:50:52:52 2021, RTT = 15.46 ms
Ping 14: host 127.0.0.1 replied: seq 12 Wed Oct 14 23:50:52:52 2021, RTT = 999.36 ms
Ping 15: host 127.0.0.1 replied: seq 13 Wed Oct 14 23:50:53:53 2021, RTT = 27.15 ms
Ping 16: host 127.0.0.1 replied: seq 14 Wed Oct 14 23:50:53:53 2021, RTT = 34.24 ms
Ping 17: host 127.0.0.1 replied: seq 15 Wed Oct 14 23:50:54:54 2021, RTT = 31.42 ms
Ping 18: timed out, message was lost
Ping 19: host 127.0.0.1 replied: seq 16 Wed Oct 14 23:50:54:54 2021, RTT = 0.12 ms
Ping 20: host 127.0.0.1 replied: seq 17 Wed Oct 14 23:50:54:54 2021, RTT = 997.71 ms
Ping 21: timed out, message was lost
Ping 22: host 127.0.0.1 replied: seq 18 Wed Oct 14 23:50:54:54 2021, RTT = 0.03 ms
Ping 23: host 127.0.0.1 replied: seq 19 Wed Oct 14 23:50:55:55 2021, RTT = 28.88 ms
Ping 24: host 127.0.0.1 replied: seq 20 Wed Oct 14 23:50:55:55 2021, RTT = 24.04 ms
Ping 25: host 127.0.0.1 replied: seq 21 Wed Oct 14 23:50:56:56 2021, RTT = 15.39 ms
Ping 26: host 127.0.0.1 replied: seq 22 Wed Oct 14 23:50:57:57 2021, RTT = 28.73 ms
Ping 27: host 127.0.0.1 replied: seq 23 Wed Oct 14 23:50:57:57 2021, RTT = 23.65 ms
Ping 28: host 127.0.0.1 replied: seq 24 Wed Oct 14 23:50:57:57 2021, RTT = 11.63 ms
Ping 29: host 127.0.0.1 replied: seq 25 Wed Oct 14 23:50:57:57 2021, RTT = 31.52 ms
Ping 30: host 127.0.0.1 replied: seq 26 Wed Oct 14 23:50:57:57 2021, RTT = 11.74 ms
Ping 31: timed out, message was lost
Ping 32: host 127.0.0.1 replied: seq 27 Wed Oct 14 23:50:57:57 2021, RTT = 0.08 ms
Ping 33: host 127.0.0.1 replied: seq 28 Wed Oct 14 23:50:57:57 2021, RTT = 30.39 ms
Ping 34: host 127.0.0.1 replied: seq 29 Wed Oct 14 23:50:57:57 2021, RTT = 15.59 ms
Ping 35: host 127.0.0.1 replied: seq 30 Wed Oct 14 23:50:57:57 2021, RTT = 18.88 ms
Ping 36: host 127.0.0.1 replied: seq 31 Wed Oct 14 23:50:57:57 2021, RTT = 21.15 ms
Ping 37: host 127.0.0.1 replied: seq 32 Wed Oct 14 23:50:58:58 2021, RTT = 28.28 ms
Ping 38: host 127.0.0.1 replied: seq 33 Wed Oct 14 23:50:58:58 2021, RTT = 26.41 ms
Ping 39: host 127.0.0.1 replied: seq 34 Wed Oct 14 23:50:58:58 2021, RTT = 14.68 ms
Ping 40: host 127.0.0.1 replied: seq 35 Wed Oct 14 23:50:58:58 2021, RTT = 12.46 ms
Ping 41: host 127.0.0.1 replied: seq 36 Wed Oct 14 23:50:58:58 2021, RTT = 31.27 ms
Ping 42: host 127.0.0.1 replied: seq 37 Wed Oct 14 23:50:58:58 2021, RTT = 27.84 ms
Ping 43: host 127.0.0.1 replied: seq 38 Wed Oct 14 23:50:58:58 2021, RTT = 13.44 ms
Ping 44: host 127.0.0.1 replied: seq 39 Wed Oct 14 23:50:58:58 2021, RTT = 999.79 ms
Ping 45: host 127.0.0.1 replied: seq 40 Wed Oct 14 23:50:58:58 2021, RTT = 23.93 ms
Ping 46: host 127.0.0.1 replied: seq 41 Wed Oct 14 23:50:58:58 2021, RTT = 22.98 ms
Ping 47: host 127.0.0.1 replied: seq 42 Wed Oct 14 23:50:58:58 2021, RTT = 17.37 ms
Ping 48: host 127.0.0.1 replied: seq 43 Wed Oct 14 23:50:58:58 2021, RTT = 33.18 ms
Ping 49: host 127.0.0.1 replied: seq 44 Wed Oct 14 23:50:58:58 2021, RTT = 33.82 ms
Ping 50: host 127.0.0.1 replied: seq 45 Wed Oct 14 23:50:59:59 2021, RTT = 25.07 ms
Ping 51: host 127.0.0.1 replied: seq 46 Wed Oct 14 23:50:59:59 2021, RTT = 16.95 ms
```

Brian Lucero

```
Ping 51: host 127.0.0.1 replied: seq 46 Wed Oct 14 23:50:59:59 2021, RTT = 16.95 ms
Ping 52: host 127.0.0.1 replied: seq 47 Wed Oct 14 23:50:59:59 2021, RTT = 17.54 ms
Ping 53: host 127.0.0.1 replied: seq 48 Wed Oct 14 23:50:59:59 2021, RTT = 13.79 ms
Ping 54: host 127.0.0.1 replied: seq 49 Wed Oct 14 23:50:59:59 2021, RTT = 999.79 ms
Ping 55: host 127.0.0.1 replied: seq 50 Wed Oct 14 23:51:00:00 2021, RTT = 30.22 ms
Ping 56: host 127.0.0.1 replied: seq 51 Wed Oct 14 23:51:00:00 2021, RTT = 21.41 ms
Ping 57: host 127.0.0.1 replied: seq 52 Wed Oct 14 23:51:00:00 2021, RTT = 13.87 ms
Ping 58: host 127.0.0.1 replied: seq 53 Wed Oct 14 23:51:00:00 2021, RTT = 13.14 ms
Ping 59: host 127.0.0.1 replied: seq 54 Wed Oct 14 23:51:00:00 2021, RTT = 16.12 ms
Ping 60: host 127.0.0.1 replied: seq 55 Wed Oct 14 23:51:01:01 2021, RTT = 33.25 ms
Ping 61: host 127.0.0.1 replied: seq 56 Wed Oct 14 23:51:01:01 2021, RTT = 23.47 ms
Ping 62: timed out, message was lost
Ping 63: host 127.0.0.1 replied: seq 57 Wed Oct 14 23:51:01:01 2021, RTT = 0.06 ms
Ping 64: host 127.0.0.1 replied: seq 58 Wed Oct 14 23:51:01:01 2021, RTT = 9.53 ms
Ping 65: host 127.0.0.1 replied: seq 59 Wed Oct 14 23:51:01:01 2021, RTT = 12.38 ms
Ping 66: host 127.0.0.1 replied: seq 60 Wed Oct 14 23:51:01:01 2021, RTT = 23.24 ms
Ping 67: host 127.0.0.1 replied: seq 61 Wed Oct 14 23:51:01:01 2021, RTT = 26.80 ms
Ping 68: host 127.0.0.1 replied: seq 62 Wed Oct 14 23:51:01:01 2021, RTT = 25.13 ms
Ping 69: host 127.0.0.1 replied: seq 63 Wed Oct 14 23:51:02:02 2021, RTT = 23.81 ms
Ping 70: host 127.0.0.1 replied: seq 64 Wed Oct 14 23:51:02:02 2021, RTT = 30.01 ms
Ping 71: host 127.0.0.1 replied: seq 65 Wed Oct 14 23:51:02:02 2021, RTT = 14.10 ms
Ping 72: host 127.0.0.1 replied: seq 66 Wed Oct 14 23:51:02:02 2021, RTT = 22.15 ms
Ping 73: host 127.0.0.1 replied: seq 67 Wed Oct 14 23:51:02:02 2021, RTT = 24.75 ms
Ping 74: host 127.0.0.1 replied: seq 68 Wed Oct 14 23:51:02:02 2021, RTT = 13.60 ms
Ping 75: host 127.0.0.1 replied: seq 69 Wed Oct 14 23:51:02:02 2021, RTT = 22.75 ms
Ping 76: host 127.0.0.1 replied: seq 70 Wed Oct 14 23:51:02:02 2021, RTT = 11.04 ms
Ping 77: host 127.0.0.1 replied: seq 71 Wed Oct 14 23:51:02:02 2021, RTT = 16.17 ms
Ping 78: timed out, message was lost
Ping 79: host 127.0.0.1 replied: seq 72 Wed Oct 14 23:51:02:02 2021, RTT = 0.03 ms
Ping 80: host 127.0.0.1 replied: seq 73 Wed Oct 14 23:51:02:02 2021, RTT = 14.98 ms
Ping 81: host 127.0.0.1 replied: seq 74 Wed Oct 14 23:51:02:02 2021, RTT = 20.43 ms
Ping 82: host 127.0.0.1 replied: seq 75 Wed Oct 14 23:51:02:02 2021, RTT = 27.55 ms
Ping 83: host 127.0.0.1 replied: seq 76 Wed Oct 14 23:51:02:02 2021, RTT = 999.60 ms
Ping 84: host 127.0.0.1 replied: seq 77 Wed Oct 14 23:51:02:02 2021, RTT = 29.31 ms
Ping 85: host 127.0.0.1 replied: seq 78 Wed Oct 14 23:51:02:02 2021, RTT = 999.40 ms
Ping 86: host 127.0.0.1 replied: seq 79 Wed Oct 14 23:51:03:03 2021, RTT = 31.56 ms
Ping 87: host 127.0.0.1 replied: seq 80 Wed Oct 14 23:51:03:03 2021, RTT = 11.07 ms
Ping 88: host 127.0.0.1 replied: seq 81 Wed Oct 14 23:51:03:03 2021, RTT = 24.81 ms
Ping 89: host 127.0.0.1 replied: seq 82 Wed Oct 14 23:51:03:03 2021, RTT = 22.25 ms
Ping 90: host 127.0.0.1 replied: seq 83 Wed Oct 14 23:51:03:03 2021, RTT = 9.47 ms
Ping 91: host 127.0.0.1 replied: seq 84 Wed Oct 14 23:51:04:04 2021, RTT = 11.42 ms
Ping 92: host 127.0.0.1 replied: seq 85 Wed Oct 14 23:51:04:04 2021, RTT = 30.00 ms
Ping 93: host 127.0.0.1 replied: seq 86 Wed Oct 14 23:51:05:05 2021, RTT = 33.26 ms
Ping 94: host 127.0.0.1 replied: seq 87 Wed Oct 14 23:51:05:05 2021, RTT = 24.74 ms
Ping 95: host 127.0.0.1 replied: seq 88 Wed Oct 14 23:51:05:05 2021, RTT = 13.02 ms
Ping 96: timed out, message was lost
Ping 97: host 127.0.0.1 replied: seq 89 Wed Oct 14 23:51:05:05 2021, RTT = 0.14 ms
Ping 98: host 127.0.0.1 replied: seq 90 Wed Oct 14 23:51:05:05 2021, RTT = 18.48 ms
Ping 99: timed out, message was lost
91
100
Min RTT = 0.03
Max RTT = 999.79
```

(3) Include your Python code listing of your UDP Ping Server with delays between 10ms and 40ms, and 20% packet losses:

a. Include as text the listing of your Python code.

```
# udppingserver_no_loss.py

from socket import *

import random

import time

# Create a UDP socket

serverSocket = socket(AF_INET, SOCK_DGRAM)

# Assign IP address and port number to socket

serverSocket.bind(('', 12002))

while True:
```

```
# Receive the client packet along with the address it is coming from

message, address = serverSocket.recvfrom(1024)



if message != "":

    ####### simulate random rtt delay

    delay = random.randint(10, 40)

    ####### 20% of (40-10) = 6

    if delay >= 34:

        ms = 1

    else:

        ms = delay / 1000

    time.sleep(ms)



# The server responds

serverSocket.sendto(message, address)
```

Part 4: HeartBeat Monitor Using Python

Another similar application to the UDP Ping would be the UDP Heartbeat. The Heartbeat can be used to check if an application is up and running on the client side and to report one-way packet loss. The client continuously sends a message acting as a heartbeat in the UDP packet to the server, which is monitoring

the heartbeat (i.e., the UDP packets) of the client. Upon receiving the packets, the server calculates the time difference. If the heartbeat packets are missing for some specified time interval, the server can assume that the client application has stopped working.

Implement the UDP Heartbeat (both client and server). You are asked to create both the server and client programs.

Use the following file naming convention:

- **xx4c.py** for client
- **xx4s.py** for server

where xx = initials (the two characters representing the first character of your first and last name).

The client program sends a ping message to the server using UDP every 5 seconds.

The server program monitors if a ping is received from the client. If the ping from the client was absent for more than 10 seconds, it prints the message “No pulse after 10 seconds. Server quits”.