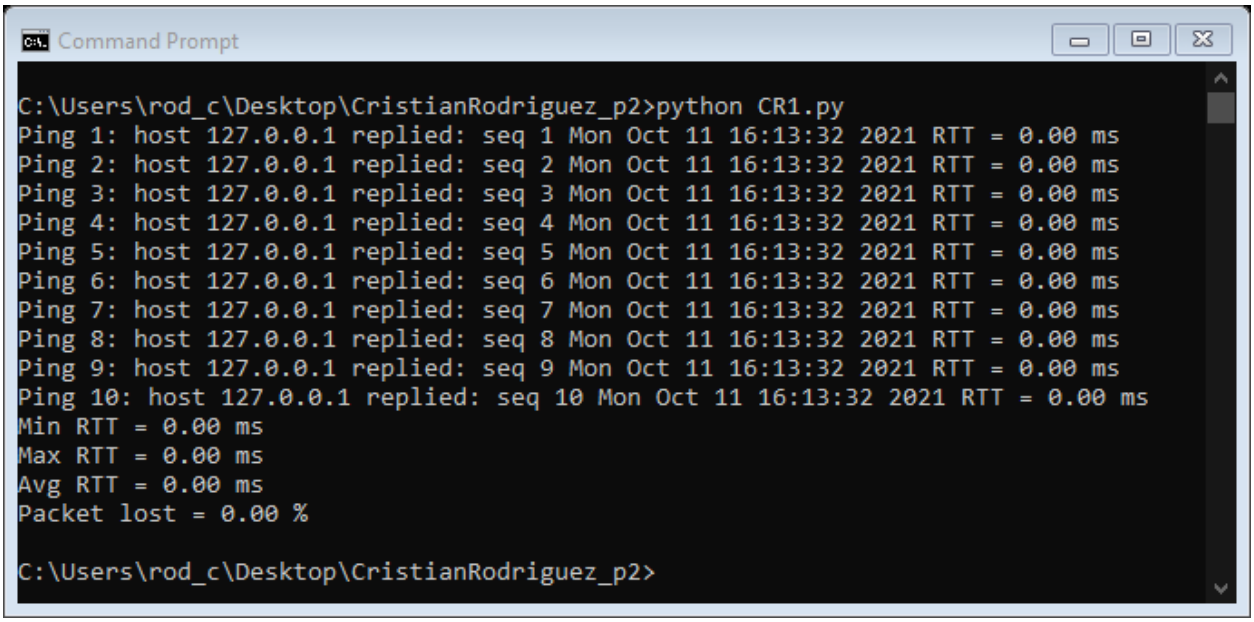
Part 1 - UDP Pinger with No Delay and No Loss

1) I connect to the server address after I create a loop which creates how many pings I would like to send to the server before the socket closes. Before I send anything, I create a timer to see how long it takes to send the message to the server then I create an end timer once I receive the message and subtract the two to retrieve how much time it took for the whole process, but if I don't retrieve anything by 1 second then it times out, but if does receive something then it prints out the message that was received from the server. Then it calculates the min, max, average round trip time and the percentage of packets lost.

2) I used a variable which lets me set up a certain amount of time the client should wait to receive a message from the server before knowing it timed out. For example, if my timeout value is set to two seconds then if the client doesn't get any data from the server before those two seconds, then the client just declares that the server has timeout out.

3) Open two terminals and go into the "CristianRodriguez_p2" directory for both. In one terminal we will start the server by typing "python server.py" and in the other terminal we will start the client by typing "python CR1.py".



4) Python Code:

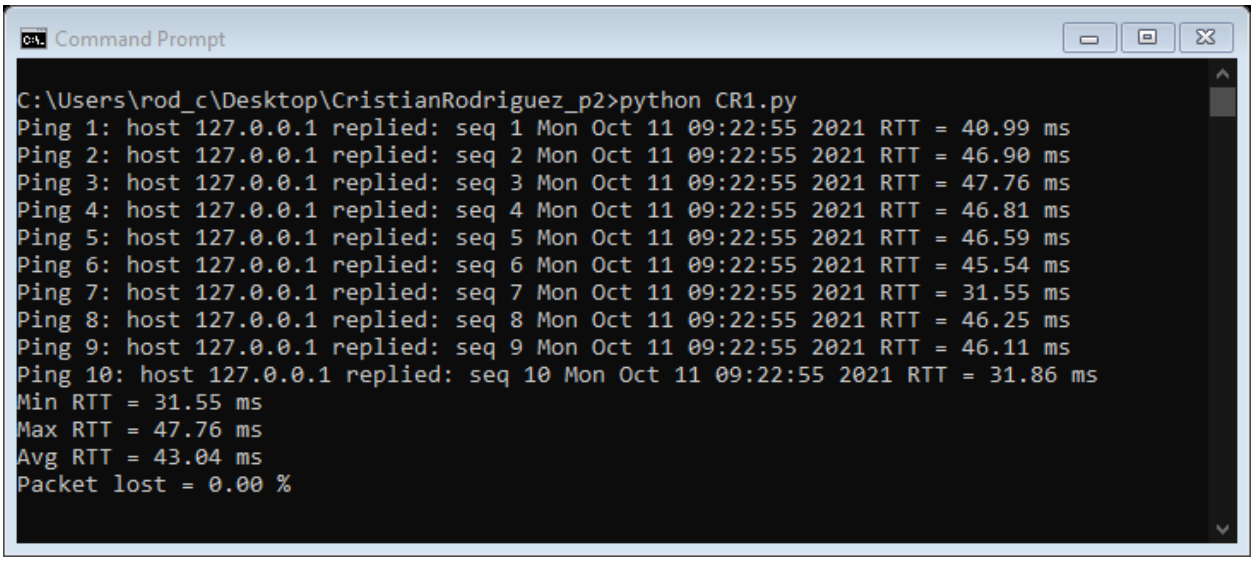
```
from socket import *

import time


serverAddress = ('127.0.0.1', 12000)

clientSocket = socket(AF_INET, SOCK_DGRAM)

packetLoss = 0

clientSocket.settimeout(1)
```

```
RTT = []


try:
    for i in range(1,11):
        start = time.time()
        message = "seq " + str(i) + " " + time.ctime(start)
        try:
            sent = clientSocket.sendto(message.encode('utf-8'),serverAddress)
            data, address = clientSocket.recvfrom(4096)
            elapsed = format((time.time()-start)*1000, '.2f')
            RTT.append(elapsed)
            print('Ping ' + str(i) + ": host 127.0.0.1 replied: " + data.decode('utf-8') + ' RTT = ' + elapsed + (' ms'))
        except timeout:
            print('Ping ' + str(i) + ': timed out, message was lost')
            packetLoss += 1

finally:
    mean = sum(map(float,RTT))/len(RTT)-packetLoss
    print('Min RTT = ' + min(RTT) + ' ms')
    print('Max RTT = ' + max(RTT) + ' ms')
    print('Avg RTT = ' +  format(mean, '.2f') + ' ms')
    print('Packet lost = ' + format(packetLoss, '.2f') + ' %')
    clientSocket.close()
```

Part 2 – UDP Pinger with Delays

1) The server first receives the message from the client and then sleeps for a random amount of time between 10 ms up to 40ms which is created from a random variable which chooses a random value between 0.01 thru 0.04. Then once its awake it sends the message back to the client.

2) Open two terminals and go into the "CristianRodriguez_p2" directory for both. In one terminal we will start the server by typing "python CR2.py" and in the other terminal we will start the client by typing "python CR1.py".



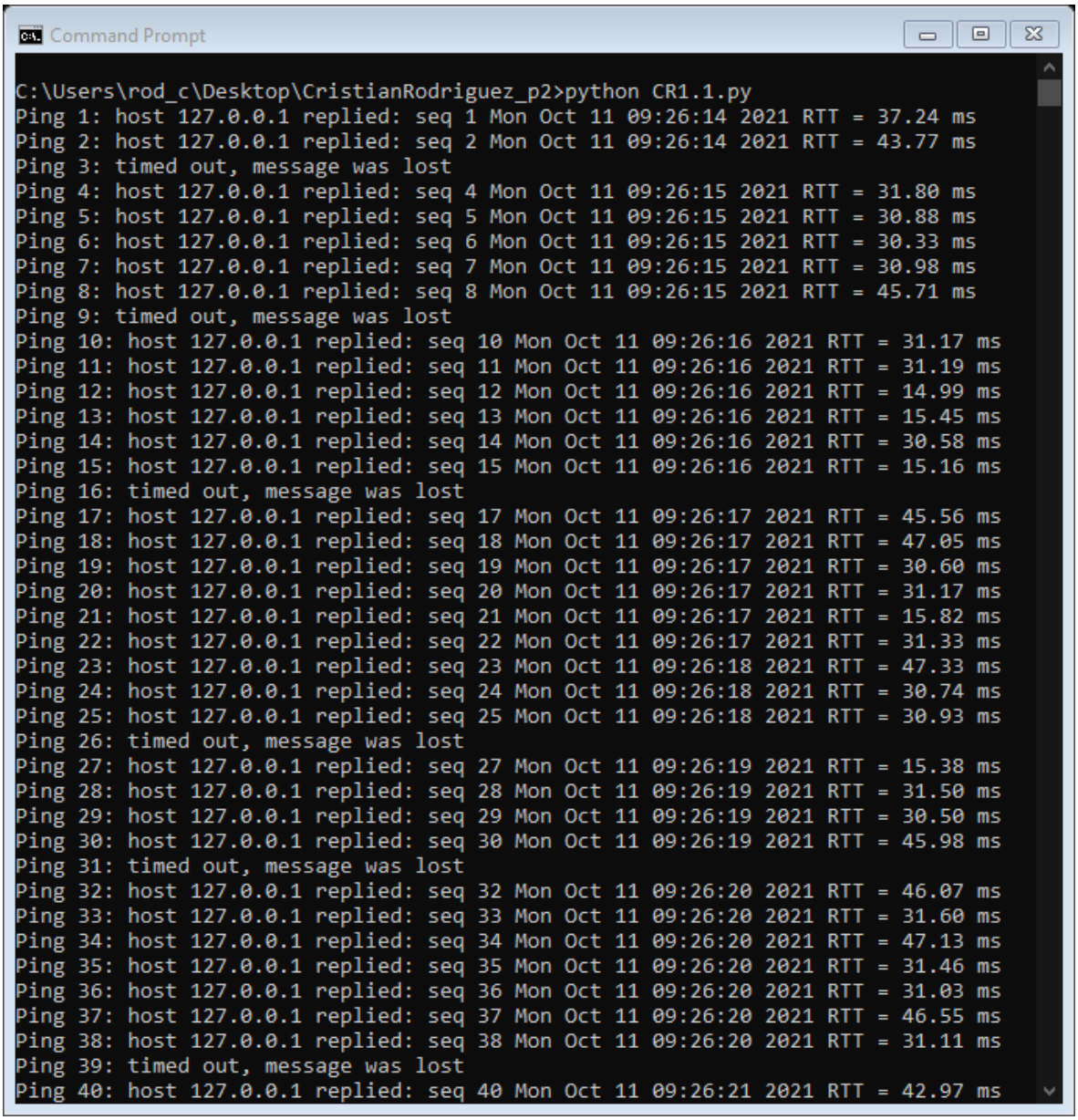
3) Python Code:

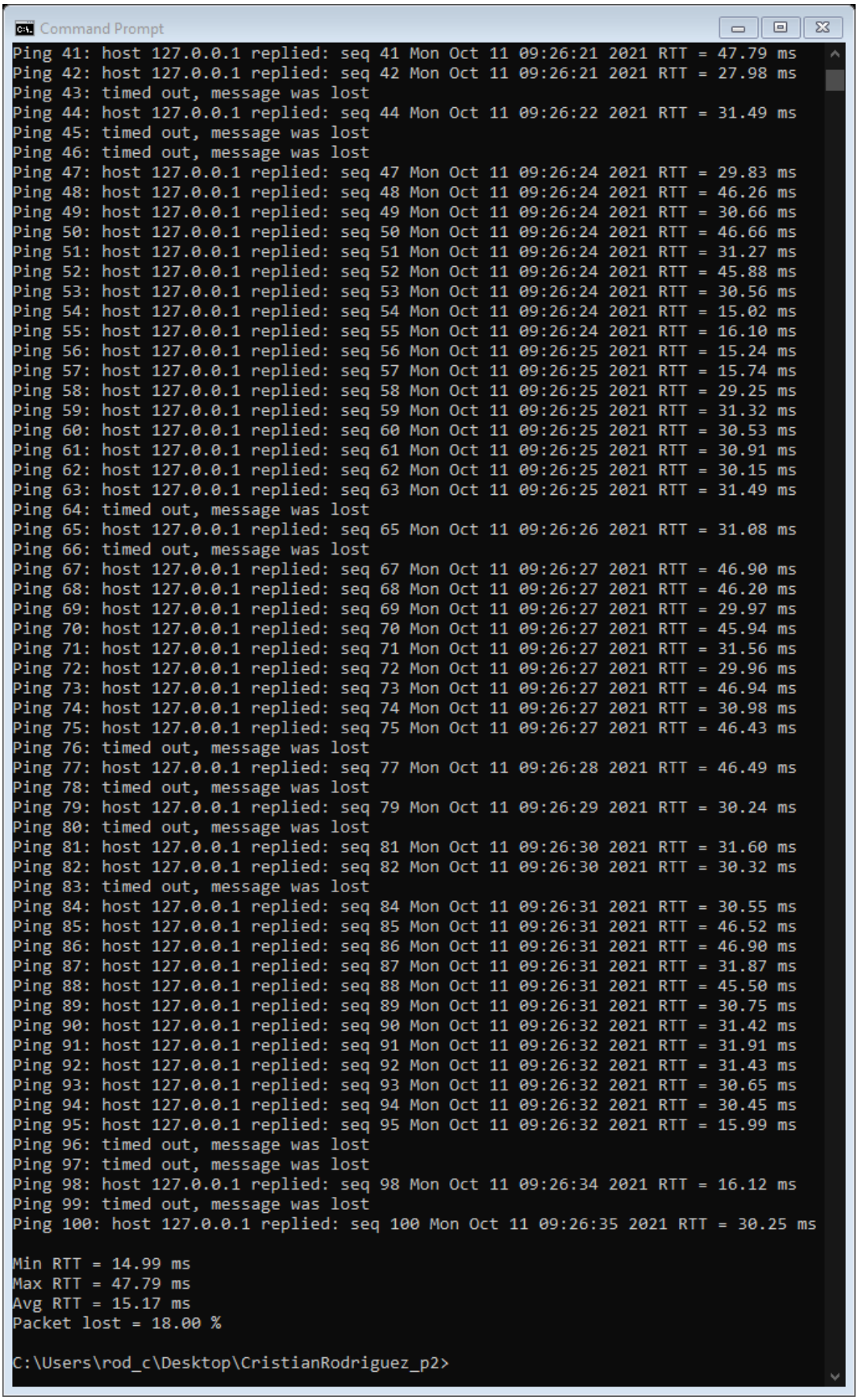
```
from socket import *
import random
import time

serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', 12000))
while True:
    rand = random.uniform(0.01, 0.04)

    message, address = serverSocket.recvfrom(1024)
    time.sleep(rand)

    serverSocket.sendto(message, address)
```

Part 3 – UDP Pinger with Delays and Packet Losses

1) The server first receives the message from the client and then sleeps for a random amount of time between 10 ms up to 40ms which is created from a random variable which chooses a random value between 0.01 thru 0.04. And to create the random losses I created a variable which generates a random value between 0 thru 1 and if the loss value is equal to 0.2 and less than 0.2 then it doesn't send anything back to the client, but if it is greater than 0.2 then is sends the message back to the client. The server could create losses from 0 to 20 percent or could sometimes be a little over.

2) Open two terminals and go into the "CristianRodriguez_p2" directory for both. In one terminal we will start the server by typing "python CR3.py" and in the other terminal we will start the client by typing "python CR1.1.py".




Command Prompt
Ping 41: host 127.0.0.1 replied: seq 41 Mon Oct 11 09:26:21 2021 RTT = 47.79 ms
Ping 42: host 127.0.0.1 replied: seq 42 Mon Oct 11 09:26:21 2021 RTT = 27.98 ms
Ping 43: timed out, message was lost
Ping 44: host 127.0.0.1 replied: seq 44 Mon Oct 11 09:26:22 2021 RTT = 31.49 ms
Ping 45: timed out, message was lost
Ping 46: timed out, message was lost
Ping 47: host 127.0.0.1 replied: seq 47 Mon Oct 11 09:26:24 2021 RTT = 29.83 ms
Ping 48: host 127.0.0.1 replied: seq 48 Mon Oct 11 09:26:24 2021 RTT = 46.26 ms
Ping 49: host 127.0.0.1 replied: seq 49 Mon Oct 11 09:26:24 2021 RTT = 30.66 ms
Ping 50: host 127.0.0.1 replied: seq 50 Mon Oct 11 09:26:24 2021 RTT = 46.66 ms
Ping 51: host 127.0.0.1 replied: seq 51 Mon Oct 11 09:26:24 2021 RTT = 31.27 ms
Ping 52: host 127.0.0.1 replied: seq 52 Mon Oct 11 09:26:24 2021 RTT = 45.88 ms
Ping 53: host 127.0.0.1 replied: seq 53 Mon Oct 11 09:26:24 2021 RTT = 30.56 ms
Ping 54: host 127.0.0.1 replied: seq 54 Mon Oct 11 09:26:24 2021 RTT = 15.02 ms
Ping 55: host 127.0.0.1 replied: seq 55 Mon Oct 11 09:26:24 2021 RTT = 16.10 ms
Ping 56: host 127.0.0.1 replied: seq 56 Mon Oct 11 09:26:25 2021 RTT = 15.24 ms
Ping 57: host 127.0.0.1 replied: seq 57 Mon Oct 11 09:26:25 2021 RTT = 15.74 ms
Ping 58: host 127.0.0.1 replied: seq 58 Mon Oct 11 09:26:25 2021 RTT = 29.25 ms
Ping 59: host 127.0.0.1 replied: seq 59 Mon Oct 11 09:26:25 2021 RTT = 31.32 ms
Ping 60: host 127.0.0.1 replied: seq 60 Mon Oct 11 09:26:25 2021 RTT = 30.53 ms
Ping 61: host 127.0.0.1 replied: seq 61 Mon Oct 11 09:26:25 2021 RTT = 30.91 ms
Ping 62: host 127.0.0.1 replied: seq 62 Mon Oct 11 09:26:25 2021 RTT = 30.15 ms
Ping 63: host 127.0.0.1 replied: seq 63 Mon Oct 11 09:26:25 2021 RTT = 31.49 ms
Ping 64: timed out, message was lost
Ping 65: host 127.0.0.1 replied: seq 65 Mon Oct 11 09:26:26 2021 RTT = 31.08 ms
Ping 66: timed out, message was lost
Ping 67: host 127.0.0.1 replied: seq 67 Mon Oct 11 09:26:27 2021 RTT = 46.90 ms
Ping 68: host 127.0.0.1 replied: seq 68 Mon Oct 11 09:26:27 2021 RTT = 46.20 ms
Ping 69: host 127.0.0.1 replied: seq 69 Mon Oct 11 09:26:27 2021 RTT = 29.97 ms
Ping 70: host 127.0.0.1 replied: seq 70 Mon Oct 11 09:26:27 2021 RTT = 45.94 ms
Ping 71: host 127.0.0.1 replied: seq 71 Mon Oct 11 09:26:27 2021 RTT = 31.56 ms
Ping 72: host 127.0.0.1 replied: seq 72 Mon Oct 11 09:26:27 2021 RTT = 29.96 ms
Ping 73: host 127.0.0.1 replied: seq 73 Mon Oct 11 09:26:27 2021 RTT = 46.94 ms
Ping 74: host 127.0.0.1 replied: seq 74 Mon Oct 11 09:26:27 2021 RTT = 30.98 ms
Ping 75: host 127.0.0.1 replied: seq 75 Mon Oct 11 09:26:27 2021 RTT = 46.43 ms
Ping 76: timed out, message was lost
Ping 77: host 127.0.0.1 replied: seq 77 Mon Oct 11 09:26:28 2021 RTT = 46.49 ms
Ping 78: timed out, message was lost
Ping 79: host 127.0.0.1 replied: seq 79 Mon Oct 11 09:26:29 2021 RTT = 30.24 ms
Ping 80: timed out, message was lost
Ping 81: host 127.0.0.1 replied: seq 81 Mon Oct 11 09:26:30 2021 RTT = 31.60 ms
Ping 82: host 127.0.0.1 replied: seq 82 Mon Oct 11 09:26:30 2021 RTT = 30.32 ms
Ping 83: timed out, message was lost
Ping 84: host 127.0.0.1 replied: seq 84 Mon Oct 11 09:26:31 2021 RTT = 30.55 ms
Ping 85: host 127.0.0.1 replied: seq 85 Mon Oct 11 09:26:31 2021 RTT = 46.52 ms
Ping 86: host 127.0.0.1 replied: seq 86 Mon Oct 11 09:26:31 2021 RTT = 46.90 ms
Ping 87: host 127.0.0.1 replied: seq 87 Mon Oct 11 09:26:31 2021 RTT = 31.87 ms
Ping 88: host 127.0.0.1 replied: seq 88 Mon Oct 11 09:26:31 2021 RTT = 45.50 ms
Ping 89: host 127.0.0.1 replied: seq 89 Mon Oct 11 09:26:31 2021 RTT = 30.75 ms
Ping 90: host 127.0.0.1 replied: seq 90 Mon Oct 11 09:26:32 2021 RTT = 31.42 ms
Ping 91: host 127.0.0.1 replied: seq 91 Mon Oct 11 09:26:32 2021 RTT = 31.91 ms
Ping 92: host 127.0.0.1 replied: seq 92 Mon Oct 11 09:26:32 2021 RTT = 31.43 ms
Ping 93: host 127.0.0.1 replied: seq 93 Mon Oct 11 09:26:32 2021 RTT = 30.65 ms
Ping 94: host 127.0.0.1 replied: seq 94 Mon Oct 11 09:26:32 2021 RTT = 30.45 ms
Ping 95: host 127.0.0.1 replied: seq 95 Mon Oct 11 09:26:32 2021 RTT = 15.99 ms
Ping 96: timed out, message was lost
Ping 97: timed out, message was lost
Ping 98: host 127.0.0.1 replied: seq 98 Mon Oct 11 09:26:34 2021 RTT = 16.12 ms
Ping 99: timed out, message was lost
Ping 100: host 127.0.0.1 replied: seq 100 Mon Oct 11 09:26:35 2021 RTT = 30.25 ms
Min RTT = 14.99 ms
Max RTT = 47.79 ms
Avg RTT = 15.17 ms
Packet lost = 18.00 %
C:\Users\rod_c\Desktop\CristianRodriguez_p2>

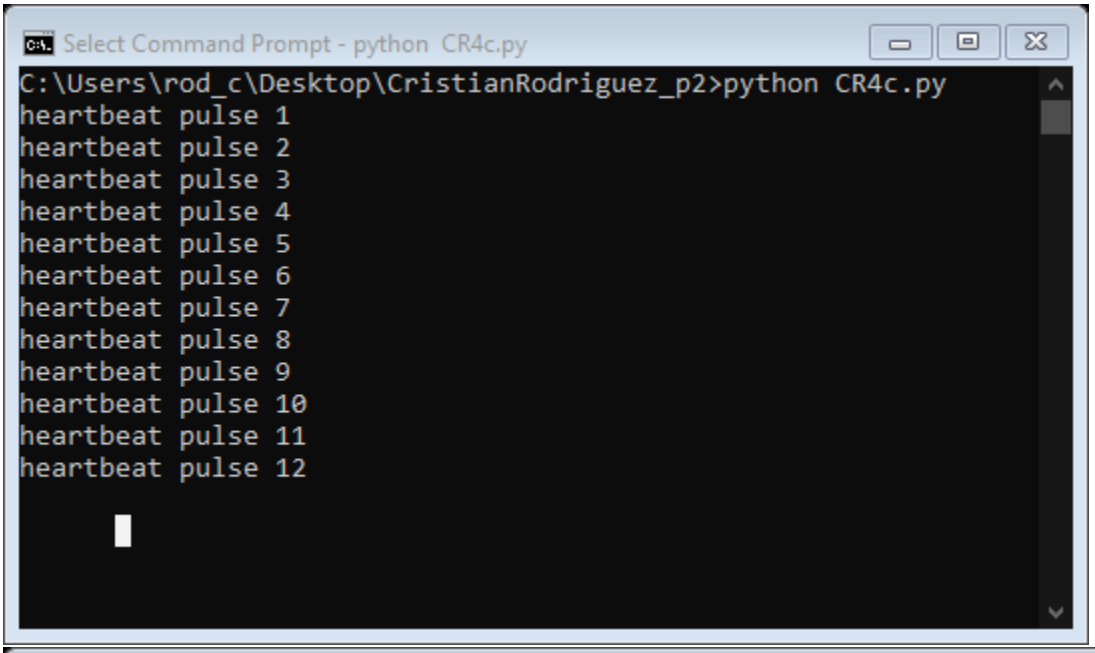
3) Python Code:

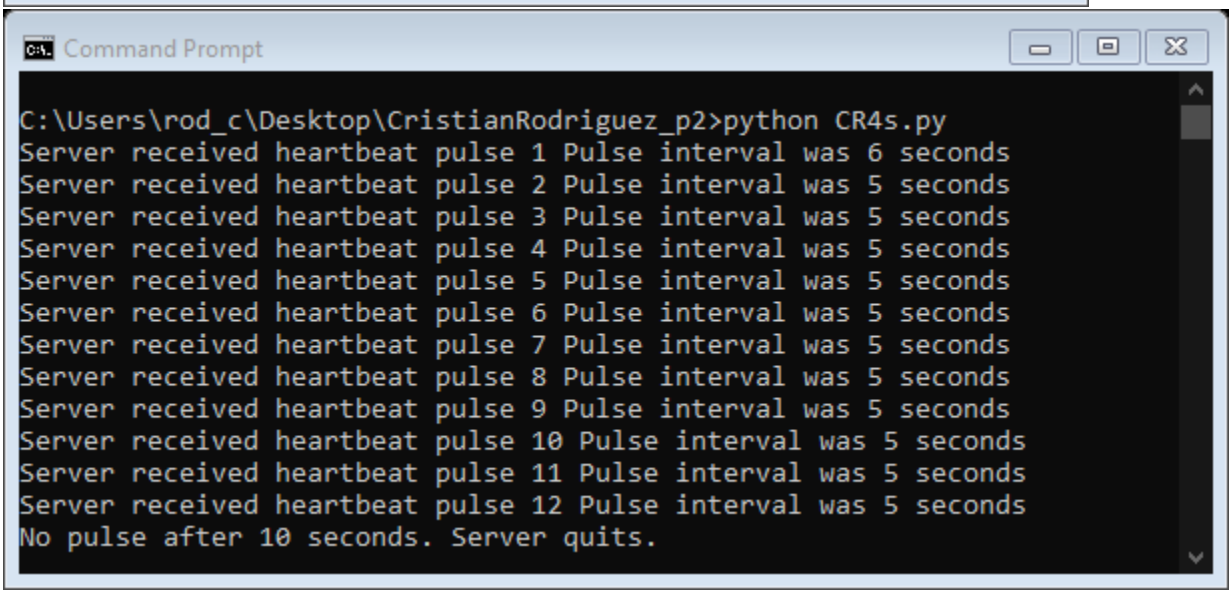
```
from socket import *
import random
import time

serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', 12000))
while True:
   rand = random.uniform(0.01, 0.04)
   loss = random.uniform(0, 1)

   message, address = serverSocket.recvfrom(1024)
   time.sleep(rand)

   if loss <= 0.20:
      continue
   serverSocket.sendto(message, address)
```

Part 4 – UDP Heartbeat Monitor

1) Open two terminals and go into the “CristianRodriguez_p2” directory for both. In one terminal we will start the server by typing “python CR4s.py” and in the other terminal we will start the client by typing “python CR4c.py”.

2)





3) Client Python Code:

```
import time

from socket import *
```

```
clientSocket = socket(AF_INET, SOCK_DGRAM)
serverAddress = ('127.0.0.1', 12000)

i = 1
while True:
   msg = 'heartbeat pulse ' + str(i)
   print(msg)
   clientSocket.sendto(msg.encode('utf-8'), serverAddress)
   i += 1
   time.sleep(5)
```

Server Python Code:

```
from socket import *
import time

serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', 12000))

serverSocket.settimeout(10)

while True:
   try:
      start = time.time()

      message, address = serverSocket.recvfrom(1024)
      end = time.time()

      elapsed = end - start
      print('Server received ' + message.decode('utf-8') + ' Pulse interval was ' + format(round(elapsed)) + '
seconds')
   except timeout:
```

```
        print('No pulse after 10 seconds. Server quits.')
        break
print('Server stops.')
```