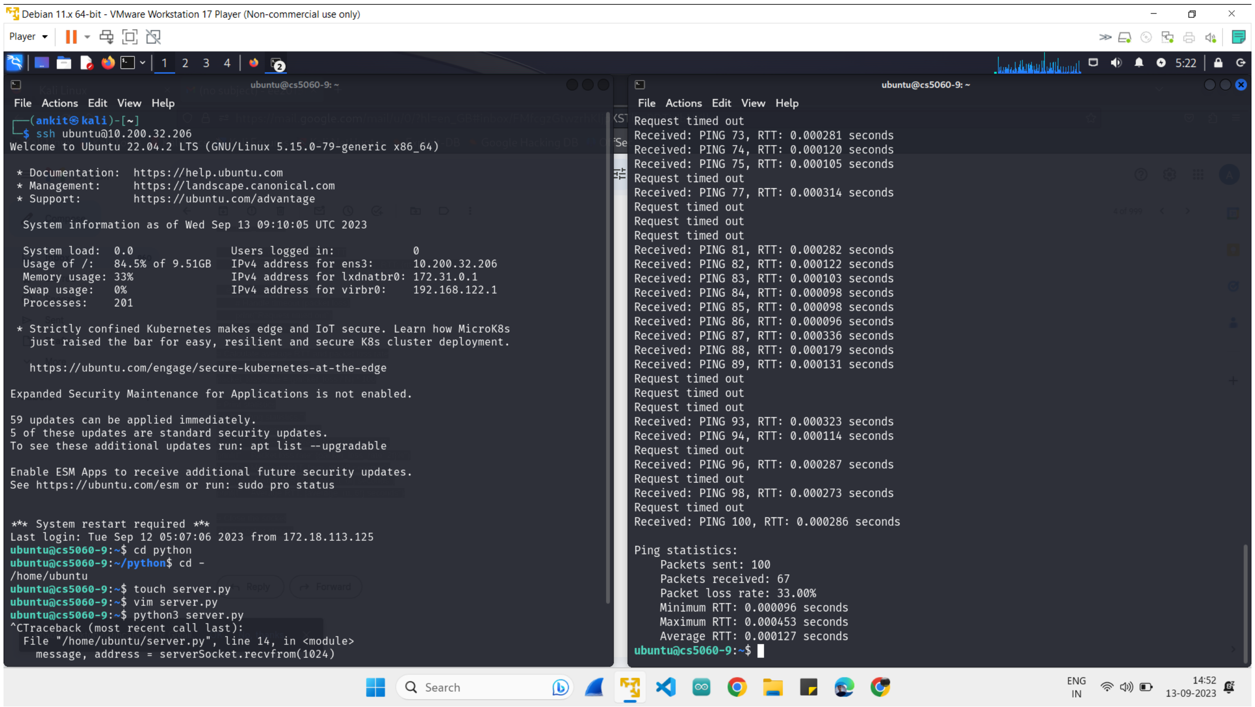
UDP CODE : UDPPingerClient.py

```
import socket
import time

# Server address and port
server_address = (172.31.0.2, 12000)
server_timeout = 1

# Number of pings to send
N = int(input("Enter the number of pings: "))

# Initialize variables for RTT statistics
min_rtt = float('inf')
max_rtt = 0
total_rtt = 0
packet_loss = 0

# Create a UDP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Loop for sending pings
for sequence_number in range(1, N + 1):
    try:
        # Prepare the message to send
        message = f"Ping {sequence_number}".encode()

        # Record the start time
        start_time = time.time()

        # Send the ping message to the server
        client_socket.sendto(message, server_address)

        # Set a timeout for receiving a response
        client_socket.settimeout(server_timeout)

        # Wait for a response
        response, server_address = client_socket.recvfrom(1024)

        # Calculate round-trip time (RTT)
        rtt = time.time() - start_time

        # Update RTT statistics
        min_rtt = min(min_rtt, rtt)
        max_rtt = max(max_rtt, rtt)
        total_rtt += rtt

        # Print the response and RTT
        print(f"Received: {response.decode()}, RTT: {rtt:.6f} seconds")

    except socket.timeout:
        # Handle timeout (packet loss)
        print("Request timed out")
        packet_loss += 1

# Calculate average RTT and packet loss rate
average_rtt = total_rtt / N
packet_loss_rate = (packet_loss / N) * 100

# Print statistics
print("\nPing statistics:")
print(f"    Packets sent: {N}")
print(f"    Packets received: {N - packet_loss}")
print(f"    Packet loss rate: {packet_loss_rate:.2f}%")
print(f"    Minimum RTT: {min_rtt:.6f} seconds")
print(f"    Maximum RTT: {max_rtt:.6f} seconds")
print(f"    Average RTT: {average_rtt:.6f} seconds")

# Close the socket
client_socket.close()
```

SCREENSHORT



Debian 11.x 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
1 2 3 4
5:22
ubuntu@cs5060-9: ~
File Actions Edit View Help
ubuntu@cs5060-9: ~
File Actions Edit View Help
Search
ENG
IN
14:52
13-09-2023

```
┌──(ankit㉿kali)-[~]
└─$ ssh ubuntu@10.200.32.206
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Sep 13 09:10:05 UTC 2023

  System load:  0.0               Users logged in:          0
  Usage of /:   84.5% of 9.51GB   IPv4 address for ens3:    10.200.32.206
  Memory usage: 33%               IPv4 address for lxdnatbr0: 172.31.0.1
  Swap usage:   0%                IPv4 address for virbr0:    192.168.122.1
  Processes:    201

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Tue Sep 12 05:07:06 2023 from 172.18.113.125
ubuntu@cs5060-9:~$ cd python
ubuntu@cs5060-9:~/python$ cd -
/home/ubuntu
ubuntu@cs5060-9:~$ touch server.py
ubuntu@cs5060-9:~$ vim server.py
ubuntu@cs5060-9:~$ python3 server.py
^CTraceback (most recent call last):
  File "/home/ubuntu/server.py", line 14, in <module>
    message, address = serverSocket.recvfrom(1024)
```

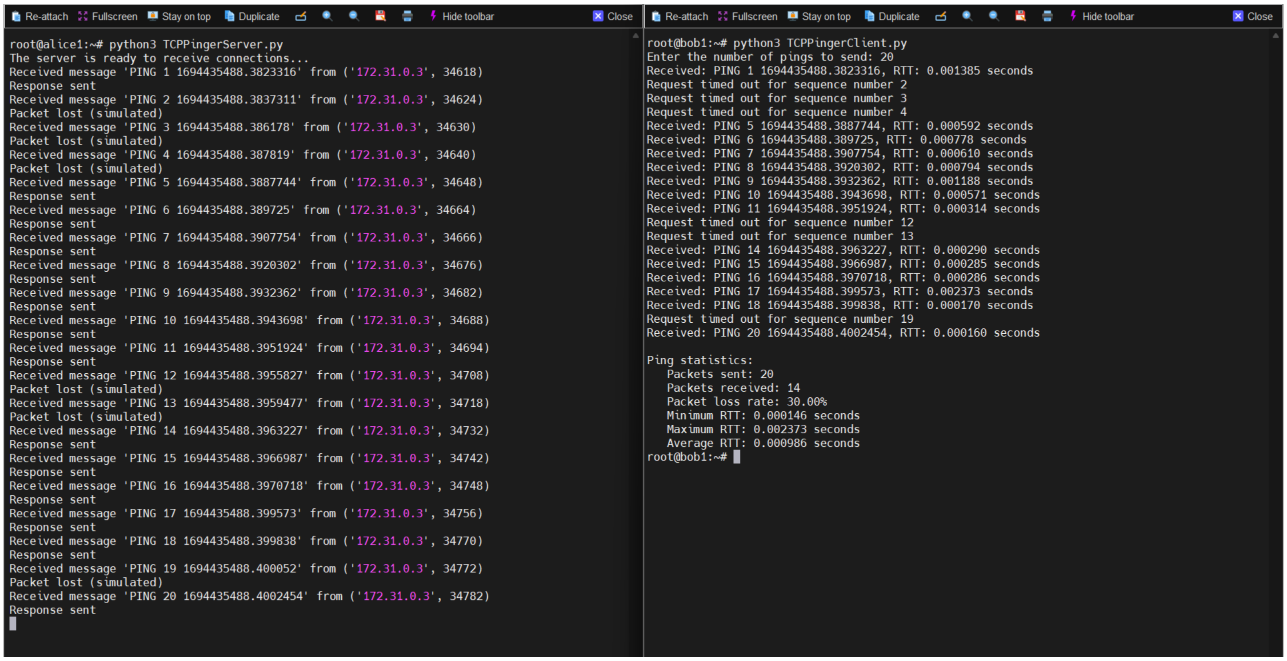
```
Request timed out
Received: PING 73, RTT: 0.000281 seconds
Received: PING 74, RTT: 0.000120 seconds
Received: PING 75, RTT: 0.000105 seconds
Request timed out
Received: PING 77, RTT: 0.000314 seconds
Request timed out
Request timed out
Request timed out
Received: PING 81, RTT: 0.000282 seconds
Received: PING 82, RTT: 0.000122 seconds
Received: PING 83, RTT: 0.000103 seconds
Received: PING 84, RTT: 0.000098 seconds
Received: PING 85, RTT: 0.000098 seconds
Received: PING 86, RTT: 0.000096 seconds
Received: PING 87, RTT: 0.000336 seconds
Received: PING 88, RTT: 0.000179 seconds
Received: PING 89, RTT: 0.000131 seconds
Request timed out
Request timed out
Request timed out
Received: PING 93, RTT: 0.000323 seconds
Received: PING 94, RTT: 0.000114 seconds
Request timed out
Received: PING 96, RTT: 0.000287 seconds
Request timed out
Received: PING 98, RTT: 0.000273 seconds
Request timed out
Received: PING 100, RTT: 0.000286 seconds

Ping statistics:
    Packets sent: 100
    Packets received: 67
    Packet loss rate: 33.00%
    Minimum RTT: 0.000096 seconds
    Maximum RTT: 0.000453 seconds
    Average RTT: 0.000127 seconds
ubuntu@cs5060-9:~$
```

TCP CODE : TCPPingerClient.py

```
import socket
import time

# Server address and port
server_address = ('server_ip_address', 12000)  # Replace 'server_ip_address' with the actual server IP address

# Number of pings to send
N = int(input("Enter the number of pings to send: "))

# Create a TCP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Set a timeout for socket operations (1 second)
client_socket.settimeout(1.0)

# Initialize variables for statistics
total_rtt = 0.0
min_rtt = float('inf')
max_rtt = 0.0
```

```
lost_count = 0

try:
    # Connect to the server
    client_socket.connect(server_address)

    # Sending pings and measuring RTT
    for i in range(N):
        # Get the current time before sending the ping
        start_time = time.time()

        # Send a ping message to the server
        message = f"Ping {i + 1}".encode()
        client_socket.send(message)

        try:
            # Receive the response from the server
            response = client_socket.recv(1024)

            # Get the current time after receiving the response
            end_time = time.time()

            # Calculate the RTT
            rtt = end_time - start_time

            # Update statistics
            total_rtt += rtt
            if rtt < min_rtt:
                min_rtt = rtt
            if rtt > max_rtt:
                max_rtt = rtt

            # Print the response and RTT
            print(f"Response: {response.decode()}, RTT: {rtt:.6f} seconds")
        except socket.timeout:
            # Handle timeout (packet loss)
            print("Request timed out")
            lost_count += 1

    # Calculate average RTT and packet loss rate
    average_rtt = total_rtt / N
    packet_loss_rate = (lost_count / N) * 100
    # Print statistics
    print("\nPing statistics:")
    print(f"  Packets sent: {N}")
    print(f"  Packets received: {N - lost_count}")
    print(f"  Packets lost: {lost_count} ({packet_loss_rate:.2f}%)")
    print(f"  Min RTT: {min_rtt:.6f} seconds")
    print(f"  Max RTT: {max_rtt:.6f} seconds")
    print(f"  Average RTT: {average_rtt:.6f} seconds")

except Exception as e:
    print(f"Error: {e}")

finally:
    # Close the socket
    client_socket.close()
```

SCREENSHORT



```
root@alice1:~# python3 TCPPingerServer.py
The server is ready to receive connections...
Received message 'PING 1 1694435488.3823316' from ('172.31.0.3', 34618)
Response sent
Received message 'PING 2 1694435488.3837311' from ('172.31.0.3', 34624)
Packet lost (simulated)
Received message 'PING 3 1694435488.386178' from ('172.31.0.3', 34630)
Packet lost (simulated)
Received message 'PING 4 1694435488.387819' from ('172.31.0.3', 34640)
Packet lost (simulated)
Received message 'PING 5 1694435488.3887744' from ('172.31.0.3', 34648)
Response sent
Received message 'PING 6 1694435488.389725' from ('172.31.0.3', 34664)
Response sent
Received message 'PING 7 1694435488.3907754' from ('172.31.0.3', 34666)
Response sent
Received message 'PING 8 1694435488.3920302' from ('172.31.0.3', 34676)
Response sent
Received message 'PING 9 1694435488.3932362' from ('172.31.0.3', 34682)
Response sent
Received message 'PING 10 1694435488.3943698' from ('172.31.0.3', 34688)
Response sent
Received message 'PING 11 1694435488.3951924' from ('172.31.0.3', 34694)
Response sent
Received message 'PING 12 1694435488.3955827' from ('172.31.0.3', 34708)
Packet lost (simulated)
Received message 'PING 13 1694435488.3959477' from ('172.31.0.3', 34718)
Packet lost (simulated)
Received message 'PING 14 1694435488.3963227' from ('172.31.0.3', 34732)
Response sent
Received message 'PING 15 1694435488.3966987' from ('172.31.0.3', 34742)
Response sent
Received message 'PING 16 1694435488.3970718' from ('172.31.0.3', 34748)
Response sent
Received message 'PING 17 1694435488.399573' from ('172.31.0.3', 34756)
Response sent
Received message 'PING 18 1694435488.399838' from ('172.31.0.3', 34770)
Response sent
Received message 'PING 19 1694435488.400052' from ('172.31.0.3', 34772)
Packet lost (simulated)
Received message 'PING 20 1694435488.4002454' from ('172.31.0.3', 34782)
Response sent
```

```
root@bob1:~# python3 TCPPingerClient.py
Enter the number of pings to send: 20
Received: PING 1 1694435488.3823316, RTT: 0.001385 seconds
Request timed out for sequence number 2
Request timed out for sequence number 3
Request timed out for sequence number 4
Received: PING 5 1694435488.3887744, RTT: 0.000592 seconds
Received: PING 6 1694435488.389725, RTT: 0.000778 seconds
Received: PING 7 1694435488.3907754, RTT: 0.000610 seconds
Received: PING 8 1694435488.3920302, RTT: 0.000794 seconds
Received: PING 9 1694435488.3932362, RTT: 0.001188 seconds
Received: PING 10 1694435488.3943698, RTT: 0.000571 seconds
Received: PING 11 1694435488.3951924, RTT: 0.000314 seconds
Request timed out for sequence number 12
Request timed out for sequence number 13
Received: PING 14 1694435488.3963227, RTT: 0.000290 seconds
Received: PING 15 1694435488.3966987, RTT: 0.000285 seconds
Received: PING 16 1694435488.3970718, RTT: 0.000286 seconds
Received: PING 17 1694435488.399573, RTT: 0.002373 seconds
Received: PING 18 1694435488.399838, RTT: 0.000170 seconds
Request timed out for sequence number 19
Received: PING 20 1694435488.4002454, RTT: 0.000160 seconds

Ping statistics:
   Packets sent: 20
   Packets received: 14
   Packet loss rate: 30.00%
   Minimum RTT: 0.000146 seconds
   Maximum RTT: 0.002373 seconds
   Average RTT: 0.000986 seconds
root@bob1:~#
```

References:
1. http://docs.python.org/howto/sockets.html
2. https://man7.org/linux/man-pages/man8/tc-netem.8.html
3. https://srtlab.github.io/srt-cookbook/how-to-articles/using-netem-to-emulate-networks.html
4. https://www.cs.unm.edu/~crandall/netsfall13/TCtutorial.pdf
5. https://docs.python.org/3/library/concurrency.html

ANTI-PLAGIARISM Statement

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. Additionally, I acknowledge that I may have used AI tools, such as language models (e.g., ChatGPT, Bard), for assistance in generating and refining my assignment, and I have made all reasonable efforts to ensure that such usage complies with the academic integrity policies set for the course. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name Ankit Kumar
Roll No SM23MTECH14001
Date: 17/09/23
Signature: Ankit kumar