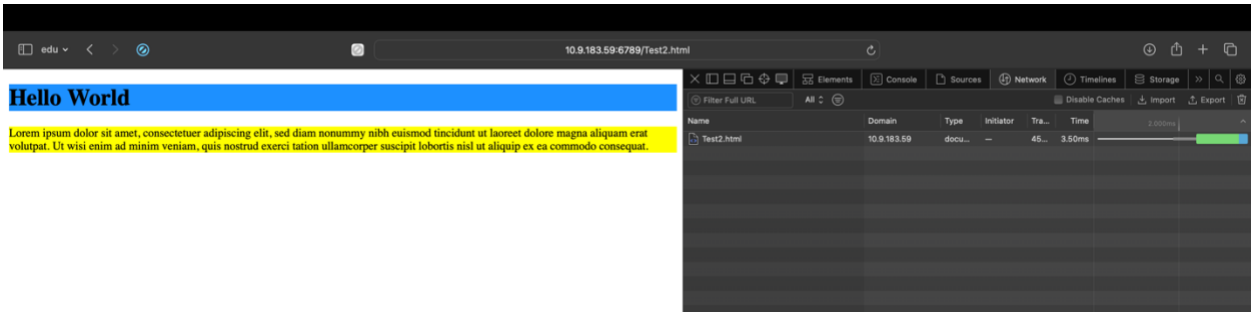
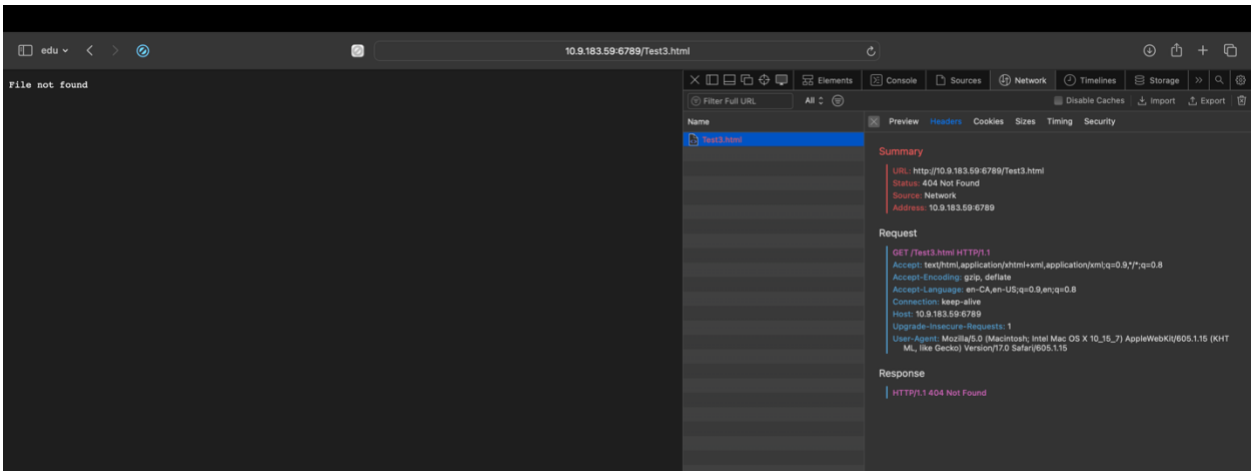
Name : Nimna Wijedasa

UCID : 30146042

```python
from socket import *
import threading

def handle_client(connectionSocket):
    try:
        # Receive the HTTP request message
        message = connectionSocket.recv(1024).decode()

        # Extract the filename from the request
        filename = message.split()[1][1:]
        print('Filename:', filename)

        # Open the requested file
        with open(filename, "rb") as f:
            outputdata = f.read()

        # Send HTTP response headers
        response_headers = "HTTP/1.1 200 OK\r\n\r\n"
        connectionSocket.send(response_headers.encode())

        # Send the content of the requested file to the client
        connectionSocket.sendall(outputdata)

    except IOError:
        # Send response message for file not found (404 Not Found)
        not_found_response = "HTTP/1.1 404 Not Found\r\n\r\nFile not found"
        connectionSocket.send(not_found_response.encode())

    # Close the client socket
    connectionSocket.close()

serverSocket = socket(AF_INET, SOCK_STREAM)

# Prepare a server socket
serverPort = 6789
serverSocket.bind(('0.0.0.0', serverPort))
serverSocket.listen(5)
```

```python
print('The server is ready to receive')

while True:
    # Establish the connection
    print('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()
    print('Connected by', addr)

    # Create a new thread to handle the client connection
    client_thread = threading.Thread(target=handle_client, args=(connectionSocket,))
    client_thread.start()
```

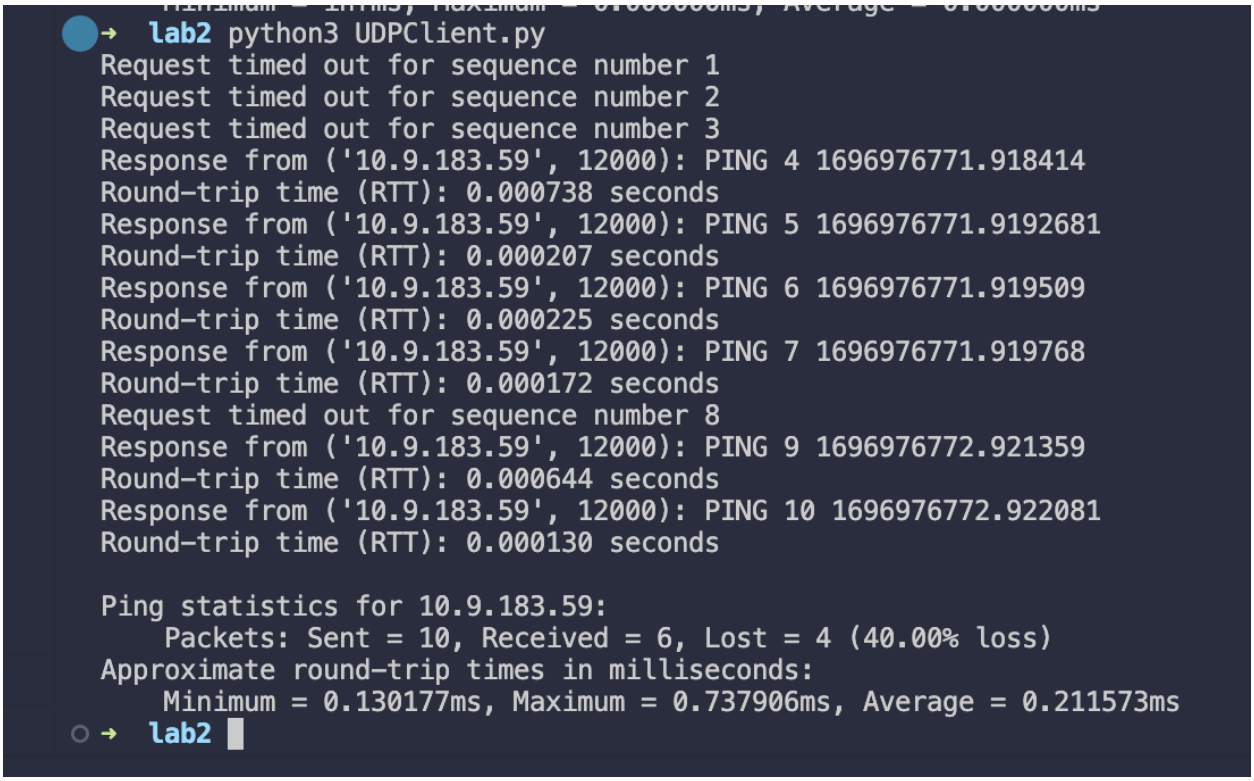





```python
import socket
import time

# Server address (replace with the actual server's IP address and port)
server_address = ('10.9.183.59', 12000)

# Number of pings to send
num_pings = 10

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

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

```
for sequence_number in range(1, num_pings + 1):
    # Get the current time
    send_time = time.time()

    # Prepare the ping message
    ping_message = f'Ping {sequence_number} {send_time}'

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

    # Set a timeout for receiving the response (1 second)
    client_socket.settimeout(1)

    try:
        # Receive the response from the server
        response, server_address = client_socket.recvfrom(1024)

        # Calculate the round-trip time (RTT)
        receive_time = time.time()
        rtt = receive_time - send_time

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

        # Print the response message and RTT
        print(f'Response from {server_address}: {response.decode()}')
        print(f'Round-trip time (RTT): {rtt:.6f} seconds')

    except socket.timeout:
        # Handle a timeout (packet loss)
        print(f'Request timed out for sequence number {sequence_number}')
        packets_lost += 1

# Calculate the average RTT
```

```
average_rtt = total_rtt / num_pings

# Calculate the packet loss rate
packet_loss_rate = (packets_lost / num_pings) * 100

# Print statistics
print(f'\nPing statistics for {server_address[0]}:')

print(f'Packets: Sent = {num_pings}, Received = {num_pings - packets_lost}, Lost = {packets_lost}
({packet_loss_rate:.2f}% loss)')

print('Approximate round-trip times in milliseconds:')

print(f'Minimum = {min_rtt * 1000:.6f}ms, Maximum = {max_rtt * 1000:.6f}ms, Average = {average_rtt * 1000:.6f}ms')

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


lab2 python3 UDPClient.py
Request timed out for sequence number 1
Request timed out for sequence number 2
Request timed out for sequence number 3
Response from ('10.9.183.59', 12000): PING 4 1696976771.918414
Round-trip time (RTT): 0.000738 seconds
Response from ('10.9.183.59', 12000): PING 5 1696976771.9192681
Round-trip time (RTT): 0.000207 seconds
Response from ('10.9.183.59', 12000): PING 6 1696976771.919509
Round-trip time (RTT): 0.000225 seconds
Response from ('10.9.183.59', 12000): PING 7 1696976771.919768
Round-trip time (RTT): 0.000172 seconds
Request timed out for sequence number 8
Response from ('10.9.183.59', 12000): PING 9 1696976772.921359
Round-trip time (RTT): 0.000644 seconds
Response from ('10.9.183.59', 12000): PING 10 1696976772.922081
Round-trip time (RTT): 0.000130 seconds
Ping statistics for 10.9.183.59:
Packets: Sent = 10, Received = 6, Lost = 4 (40.00% loss)
Approximate round-trip times in milliseconds:
Minimum = 0.130177ms, Maximum = 0.737906ms, Average = 0.211573ms
lab2