CSE 476 Mobile Communication

Fall 2021 Term Project Report

ALI BAHAR - 171044066

1 - Web Server

- ```python
  # import socket module
  from socket import *

  serverSocket = socket(AF_INET, SOCK_STREAM)

  ```

- ```python
  # Prepare a sever socket
  port = 80
  ip = "192.168.43.116"

  serverSocket.bind((ip,port))
  print(f"Server socket binded to {ip}:{port}")

  serverSocket.listen(1)

  ```

- ```python
  while True:

    try:
      # Establish the connection
      print("Ready to serve...")
      connectionSocket, addr = serverSocket.accept()
      print(f"connectionSocket: {connectionSocket}, addr: {addr}")

  ```

- ```python
      message = connectionSocket.recv(1024).decode()

      s_msg = message.split()
      filename = ""
      if(len(s_msg) > 1):
        filename = s_msg[1]
      else:
        print("Empty file name")
        raise IOError

      f = open(filename[1:])
      outputdata = f.read()

  ```

- ```python
      # Send one HTTP header line into socket
      connectionSocket.send(
        "HTTP/1.1 200 OK\r\n"                \
        "Content-Type: text/html; charset-utf-8\r\n\r\n"  \
        .encode()
      )

  ```
  ```

```
    # Send the content of the requested file to the client
    for i in range(0, len(outputdata)):
      connectionSocket.send(outputdata[i].encode())
    connectionSocket.close()

    f.close()

```

```
  except IOError:
    #Send response message for file not found
    response =  "HTTP/1.1 404 Not Found\r\n"           \
          "Content-Type: text/html; charset-utf-8\r\n"  \
          "\r\n"                        \
          "<html>"                      \
          "\t<head>"                      \
          "\t<title>"                     \
          "\t\t404 Not Found"                 \
          "\t</title>"                    \
          "\t</head>"                     \
          "\t<body>"                      \
          "\t\t<h1>"                      \
          "\t\t\t404 Not Found "                \
          "\t\t</h1>"                     \
          "\t</body>"                     \
          "</html>".encode()

    connectionSocket.sendall(response)

    #Close client socket
    connectionSocket.close()

```

```
  except KeyboardInterrupt:
    print("\nShutting Down...\n")
    break

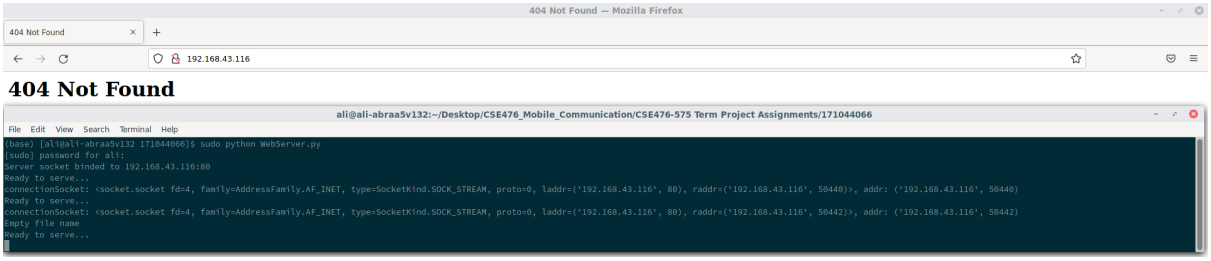
```

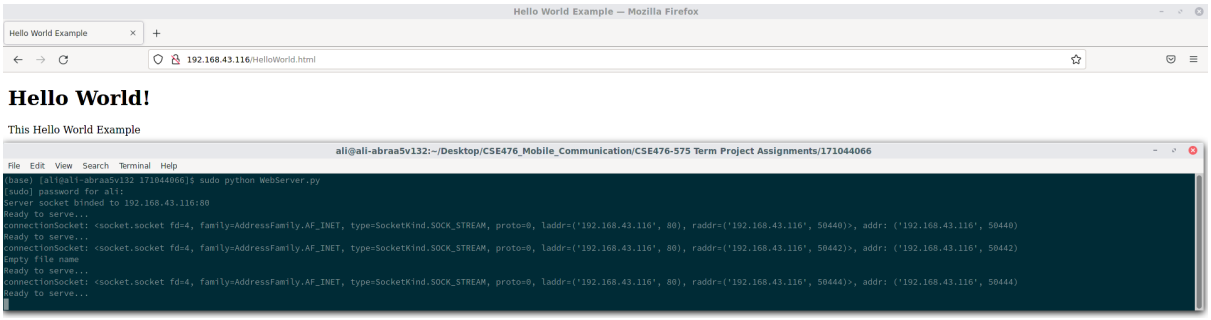
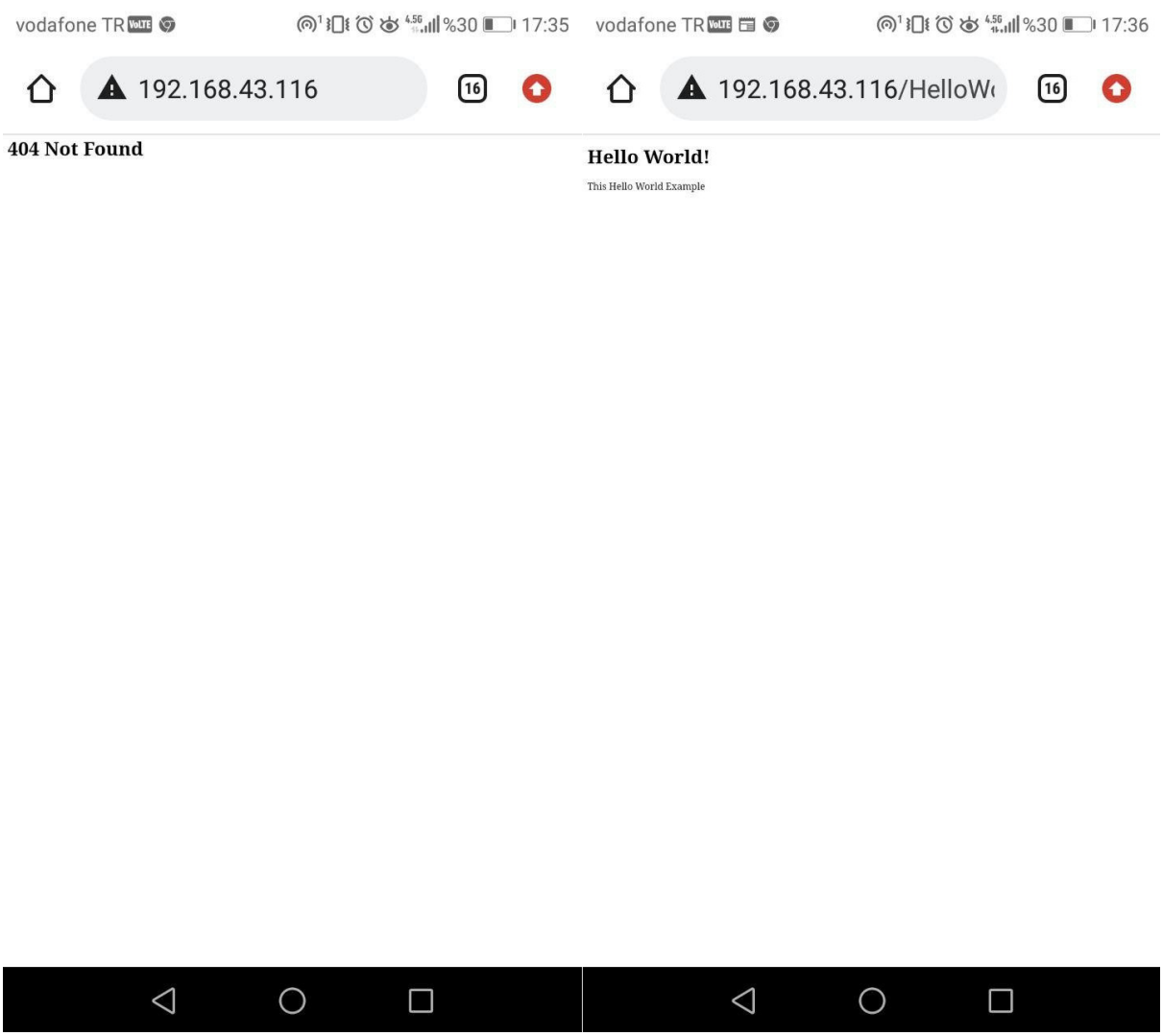
```
serverSocket.close()

```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World Example</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This Hello World Example</p>
  </body>
</html>

```



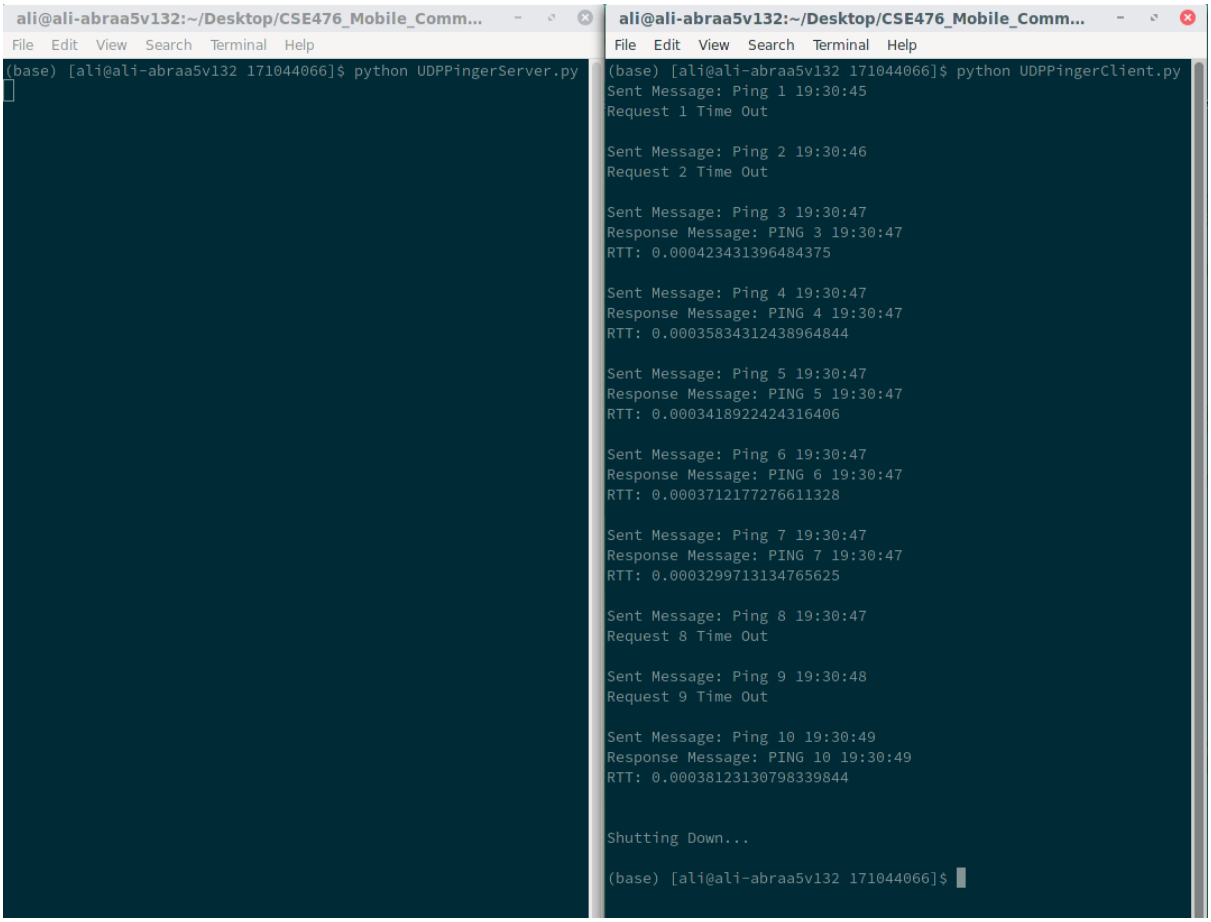
- 
Hello World Example — Mozilla Firefox
Hello World Example
192.168.43.116/HelloWorld.html
Hello World!
This Hello World Example
ali@ali-abraa5v132:~/Desktop/CSE476_Mobile_Communication/CSE476-575 Term Project Assignments/171044066
File Edit View Search Terminal Help
(base) [ali@ali-abraa5v132 171044066]$ sudo python WebServer.py
[sudo] password for ali:
Server socket binded to 192.168.43.116:80
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50440)>, addr: ('192.168.43.116', 50440)
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50442)>, addr: ('192.168.43.116', 50442)
Empty file name
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50444)>, addr: ('192.168.43.116', 50444)
Ready to serve...
- 
vodafone TR
17:35
192.168.43.116
404 Not Found
vodafone TR
17:36
192.168.43.116/HelloW
Hello World!
This Hello World Example
- 
ali@ali-abraa5v132:~/Desktop/CSE476_Mobile_Communication/CSE476-575 Term Project Assignments/171044066
File Edit View Search Terminal Help
(base) [ali@ali-abraa5v132 171044066]$ sudo python WebServer.py
[sudo] password for ali:
Server socket binded to 192.168.43.116:80
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50440)>, addr: ('192.168.43.116', 50440)
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50442)>, addr: ('192.168.43.116', 50442)
Empty file name
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.116', 50444)>, addr: ('192.168.43.116', 50444)
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.1', 39736)>, addr: ('192.168.43.1', 39736)
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.1', 39734)>, addr: ('192.168.43.1', 39734)
Ready to serve...
connectionSocket: <socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('192.168.43.116', 80), raddr=('192.168.43.1', 39744)>, addr: ('192.168.43.1', 39744)
^C
Shutting Down...
(base) [ali@ali-abraa5v132 171044066]$

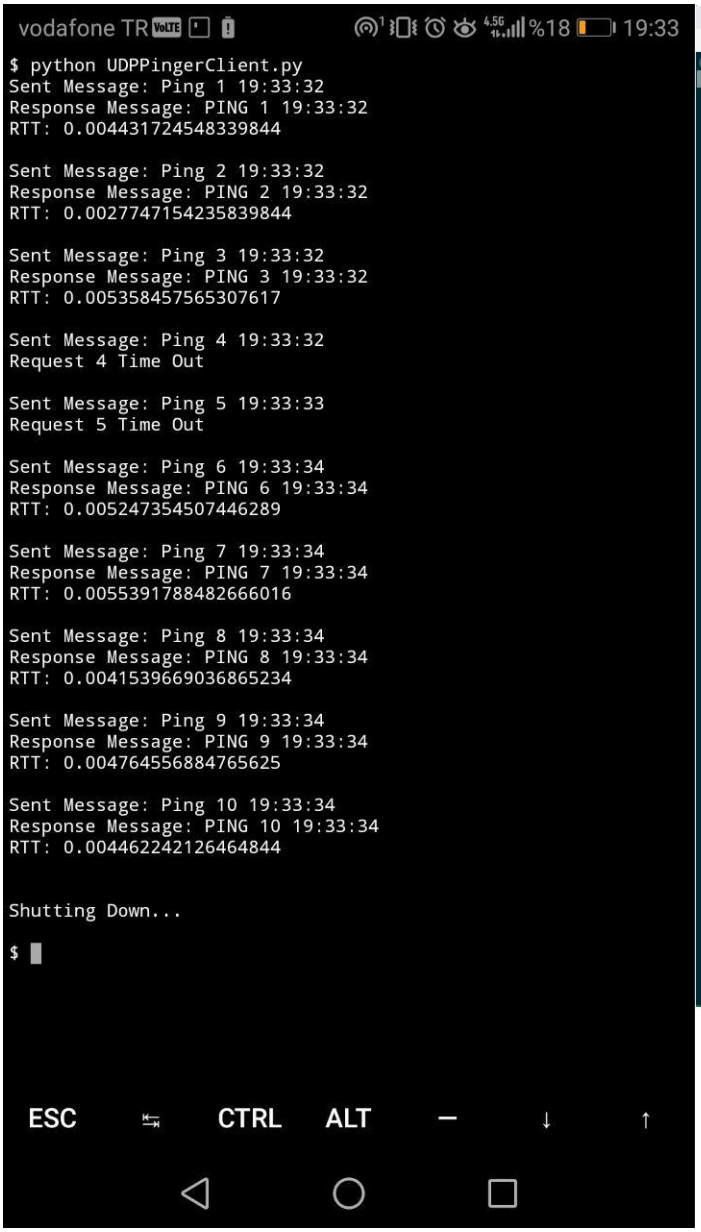
2 - UDP Pinger

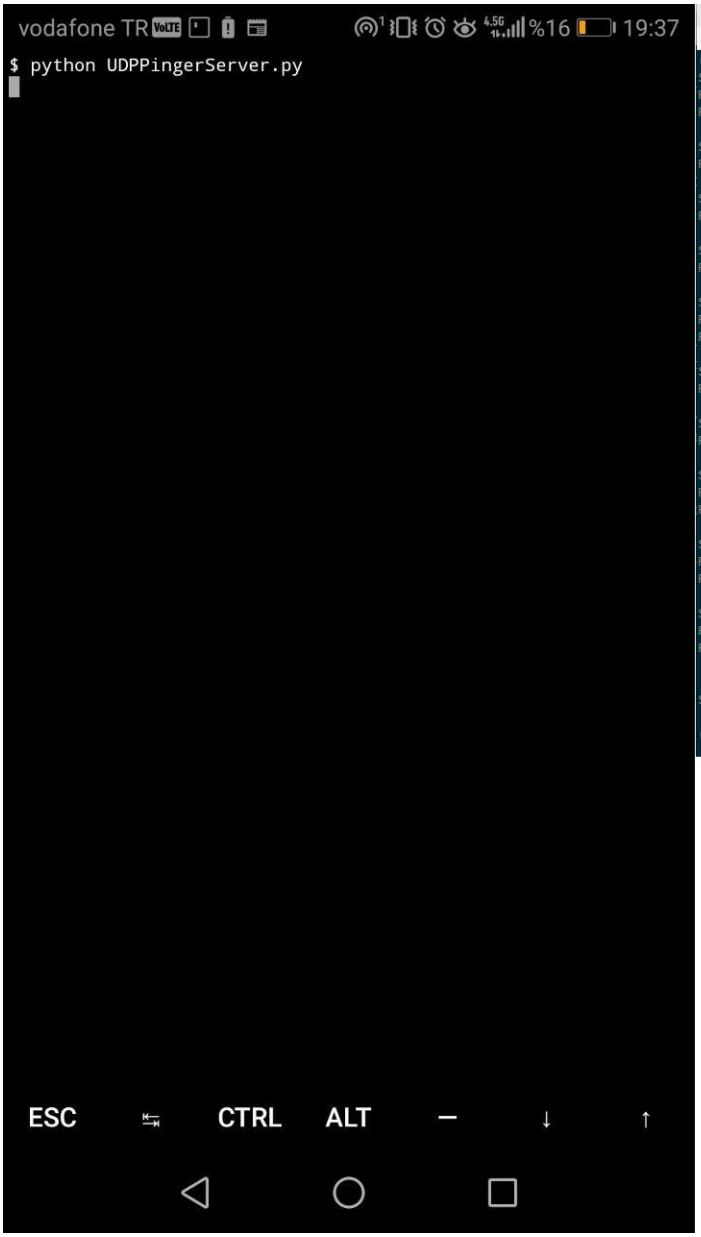
-

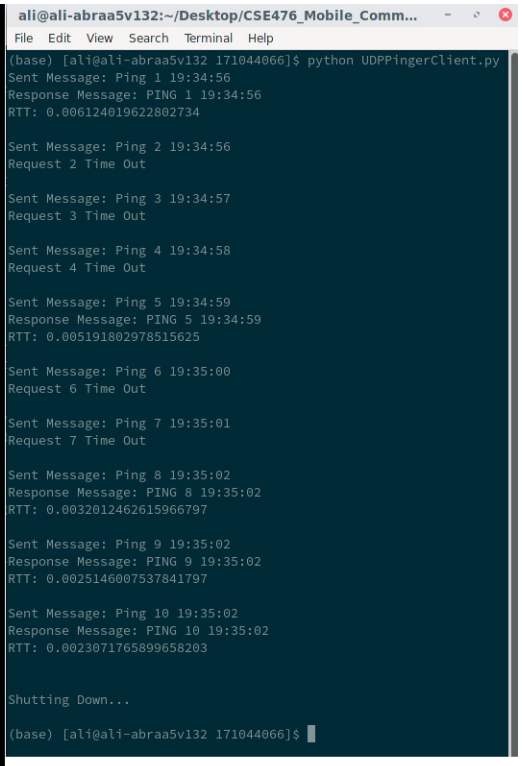
```python
from socket import *
from time import time
from time import strftime

client_socket = socket(AF_INET, SOCK_DGRAM)
client_socket.settimeout(1)

for i in range(1, 11):

  try:

    start_time = time()

    msg = f"Ping {i} {strftime('%H:%M:%S')}"
    client_socket.sendto(msg.encode(), ('', 12000))
    print(f"Sent Message: {msg}")

    data, server = client_socket.recvfrom(1024)
    print(f"Response Message: {data.decode()}", )

    end_time = time()

    print(f"RTT: {end_time - start_time}\n")

  except timeout:
    print(f"Request {i} Time Out\n")

print("\nShutting Down...\n")
client_socket.close()

```

-



- 
vodafone TR
19:33
$ python UDPPingerClient.py
Sent Message: Ping 1 19:33:32
Response Message: PING 1 19:33:32
RTT: 0.004431724548339844
Sent Message: Ping 2 19:33:32
Response Message: PING 2 19:33:32
RTT: 0.0027747154235839844
Sent Message: Ping 3 19:33:32
Response Message: PING 3 19:33:32
RTT: 0.005358457565307617
Sent Message: Ping 4 19:33:32
Request 4 Time Out
Sent Message: Ping 5 19:33:33
Request 5 Time Out
Sent Message: Ping 6 19:33:34
Response Message: PING 6 19:33:34
RTT: 0.005247354507446289
Sent Message: Ping 7 19:33:34
Response Message: PING 7 19:33:34
RTT: 0.0055391788482666016
Sent Message: Ping 8 19:33:34
Response Message: PING 8 19:33:34
RTT: 0.0041539669036865234
Sent Message: Ping 9 19:33:34
Response Message: PING 9 19:33:34
RTT: 0.004764556884765625
Sent Message: Ping 10 19:33:34
Response Message: PING 10 19:33:34
RTT: 0.004462242126464844
Shutting Down...
$
ESC
CTRL
ALT
ali@ali-abraa5v132:~/Desktop/CSE476_Mobile_Comm...
File Edit View Search Terminal Help
(base) [ali@ali-abraa5v132 171044066]$ python UDPPingerServer.py
-





3 - Mail Client

- ```python
  from socket import *
  import ssl
  from base64 import b64encode

  sourceAddr  = "mobilecommunication476@gmail.com"
  password  = "passwordispassword"
  destAddr  = "ali.bahar12389@gmail.com"

  msg = "\r\n I love computer networks!"
  subject = "Test Mail"
  endmsg = "\r\n.\r\n"

  ```

- ```python
  # Choose a mail server (e.g. Google mail server) and call it mailserver
  mailserver = ("smtp.gmail.com", 587)

  ```

- ```python
  # Create socket called clientSocket and establish a TCP connection with mailserver
  clientSocket = socket(AF_INET, SOCK_STREAM)
  ```
  ```

```
clientSocket.connect(mailserver)

recv = clientSocket.recv(1024).decode()
print(recv)
if recv[:3] != "220":
  print("220 reply not received from server.")

```

```
# Send HELO command and print server response.
heloCommand = "HELO Alice\r\n".encode()
clientSocket.send(heloCommand)
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != "250":
  print("250 reply not received from server.")

```

```
# This part is required for connect gmail smtp server because of security reasons.
clientSocket.send("STARTTLS\r\n".encode())
recv2 = clientSocket.recv(1024).decode()
print(recv2)
if recv2[:3] != "220":
  print("220 reply not received from server.")
clientSocket = ssl.wrap_socket(clientSocket)

```

```
# Authentication
clientSocket.send("AUTH LOGIN ".encode() + b64encode(sourceAddr.encode()) + "\r\n".
    encode())
recv3 = clientSocket.recv(1024).decode()
print(recv3)
if recv3[:3] != "334":
  print("334 reply not received from server.")

clientSocket.send(b64encode(password.encode()) + "\r\n".encode())
recv4 = clientSocket.recv(1024).decode()
print(recv4)
if recv4[:3] != "235":
  print("235 reply not received from server.")

```

```
# Send MAIL FROM command and print server response.
# This part carries the sender information.
clientSocket.send(f"MAIL FROM: <{sourceAddr}>\r\n".encode())
recv5 = clientSocket.recv(1024).decode()
print(recv5)
if recv5[:3] != "250":
  print("250 reply not received from server.")

```

```
# Send RCPT TO command and print server response.
# This part carries the receiver information.
clientSocket.send(f"RCPT TO: <{destAddr}>\r\n".encode())
recv6 = clientSocket.recv(1024).decode()
print(recv6)
if recv6[:3] != "250":
  print("250 reply not received from server.")

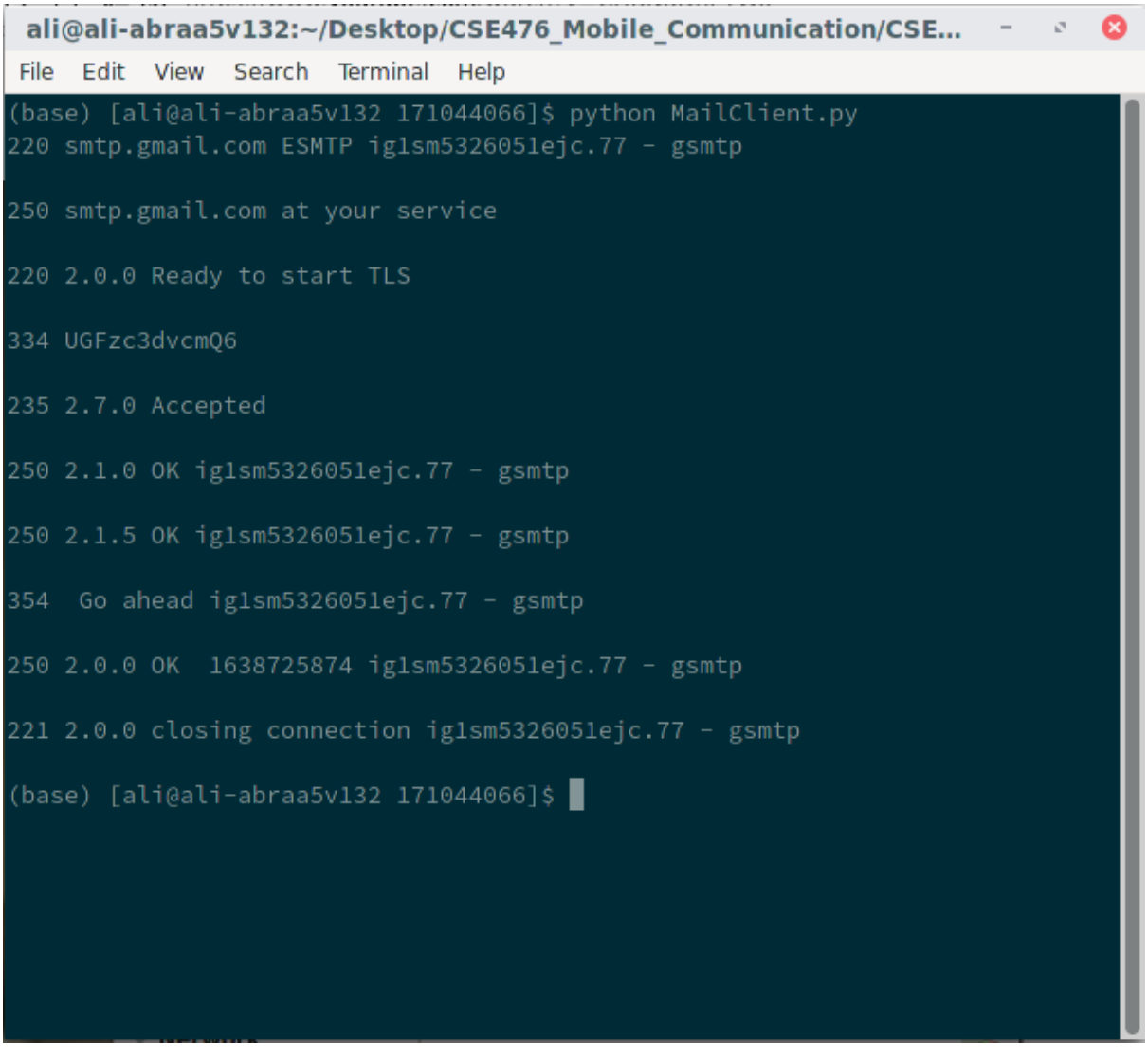
```

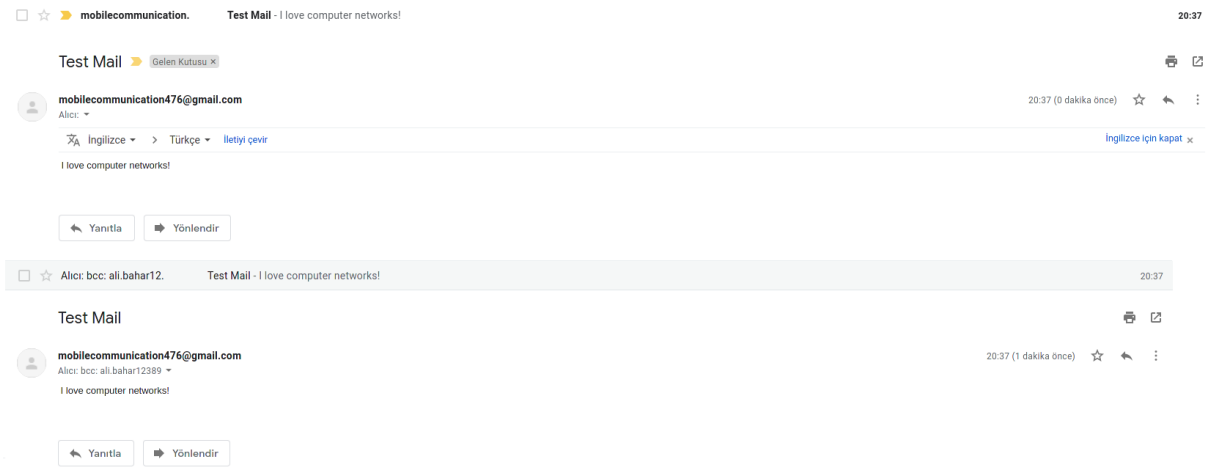
```
# Send DATA command and print server response.
# Mail content information will come after the data command.
clientSocket.send("Data\r\n".encode())
recv7 = clientSocket.recv(1024).decode()
print(recv7)
if recv7[:3] != "354":
  print("354 reply not received from server.")

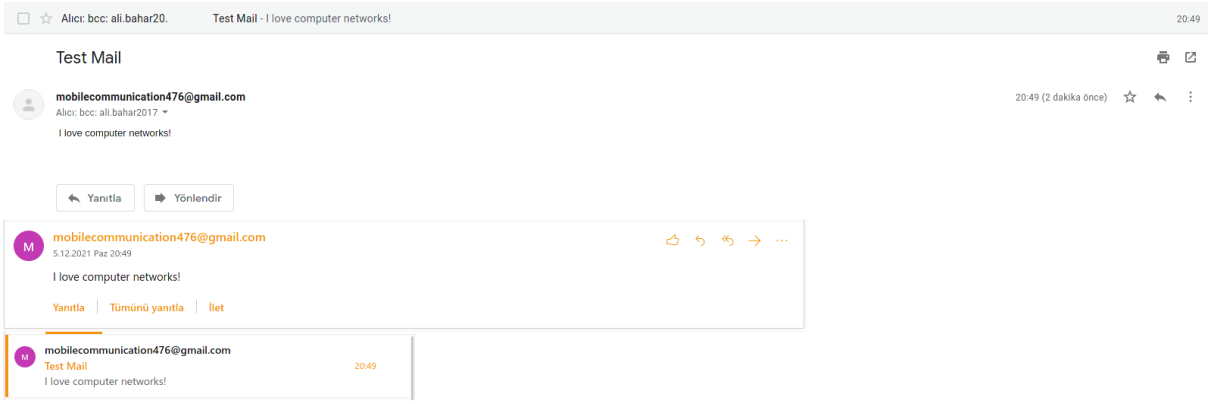
```

```python
# Send message data.
# This part carries the content information of the mail.
# Message ends with a single period.
clientSocket.send(f"SUBJECT: {subject}\n{msg + endmsg}".encode())
recv8 = clientSocket.recv(1024).decode()
print(recv8)
if recv8[:3] != "250":
  print("250 reply not received from server.")

```

```python
# Send QUIT command and get server response.
clientSocket.send("Quit\r\n".encode())
recv9 = clientSocket.recv(1024).decode()
print(recv9)
if recv9[:3] != "221":
  print('221 reply not received from server.')

clientSocket.close()

```




mobilecommunication. Test Mail - I love computer networks! 20:37
Test Mail Gelen Kutusu ×
mobilecommunication476@gmail.com
Alıcı:
20:37 (0 dakika önce)
İngilizce Türkçe İletiyi çevir
İngilizce için kapat
I love computer networks!
Yanıtla Yönlendir
Alıcı: bcc: ali.bahar12. Test Mail - I love computer networks! 20:37
Test Mail
mobilecommunication476@gmail.com
Alıcı: bcc: ali.bahar12389
20:37 (1 dakika önce)
I love computer networks!
Yanıtla Yönlendir

- 
Alıcı: bcc: ali.bahar20. Test Mail - I love computer networks! 20:49
Test Mail
mobilecommunication476@gmail.com
Alıcı: bcc: ali.bahar2017
20:49 (2 dakika önce)
I love computer networks!
Yanıtla Yönlendir
M
mobilecommunication476@gmail.com
5.12.2021 Paz 20:49
I love computer networks!
Yanıtla | Tümünü yanıtla | İlet
M
mobilecommunication476@gmail.com
Test Mail 20:49
I love computer networks!