# Week #5

## Simple Client-Server Application using Network Socket Programming

**NAME:T.LOHITH SRINIVAS**

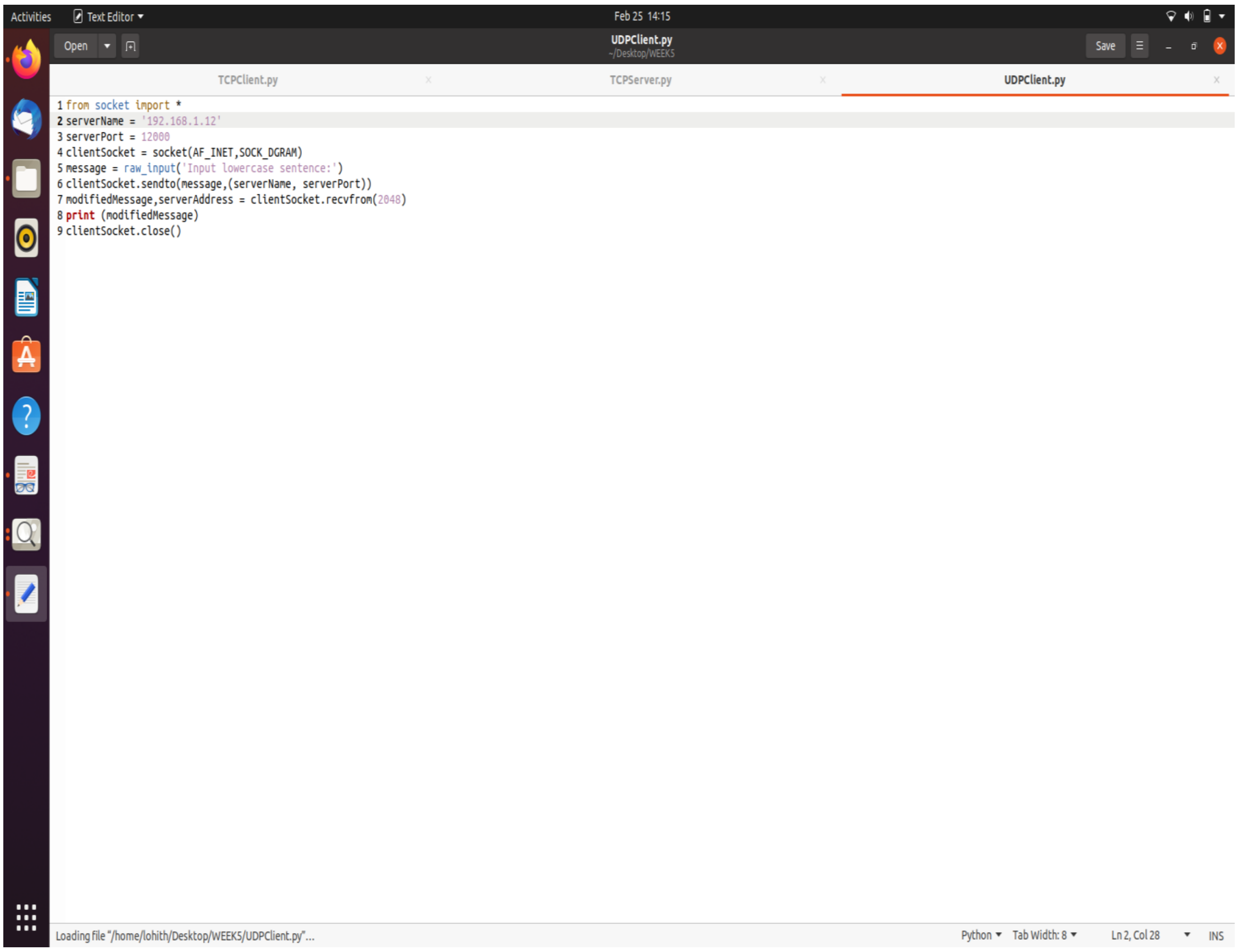
**SECTION:D**

**SRN-PES2UG19CS203**

Task 1

### Socket Programming with UDP

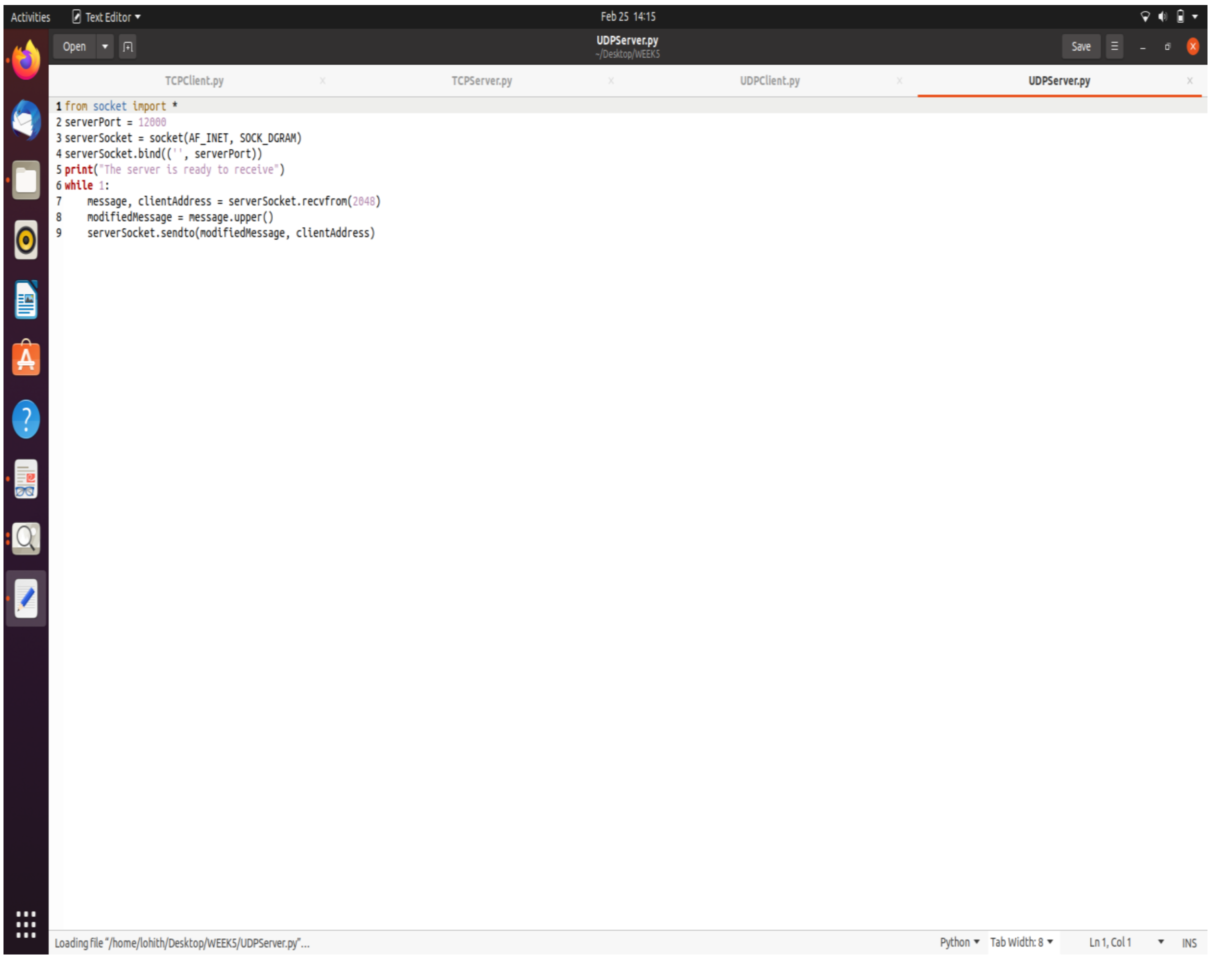
### UDPClient.py



```
from socket import *
serverName = '192.168.1.12'
serverPort = 12000
clientSocket = socket(AF_INET,SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message,(serverName, serverPort))
modifiedMessage,serverAddress = clientSocket.recvfrom(2048)
print (modifiedMessage)
clientSocket.close()
```

```
Loading file "/home/lohith/Desktop/WEEK5/UDPClient.py"...
```

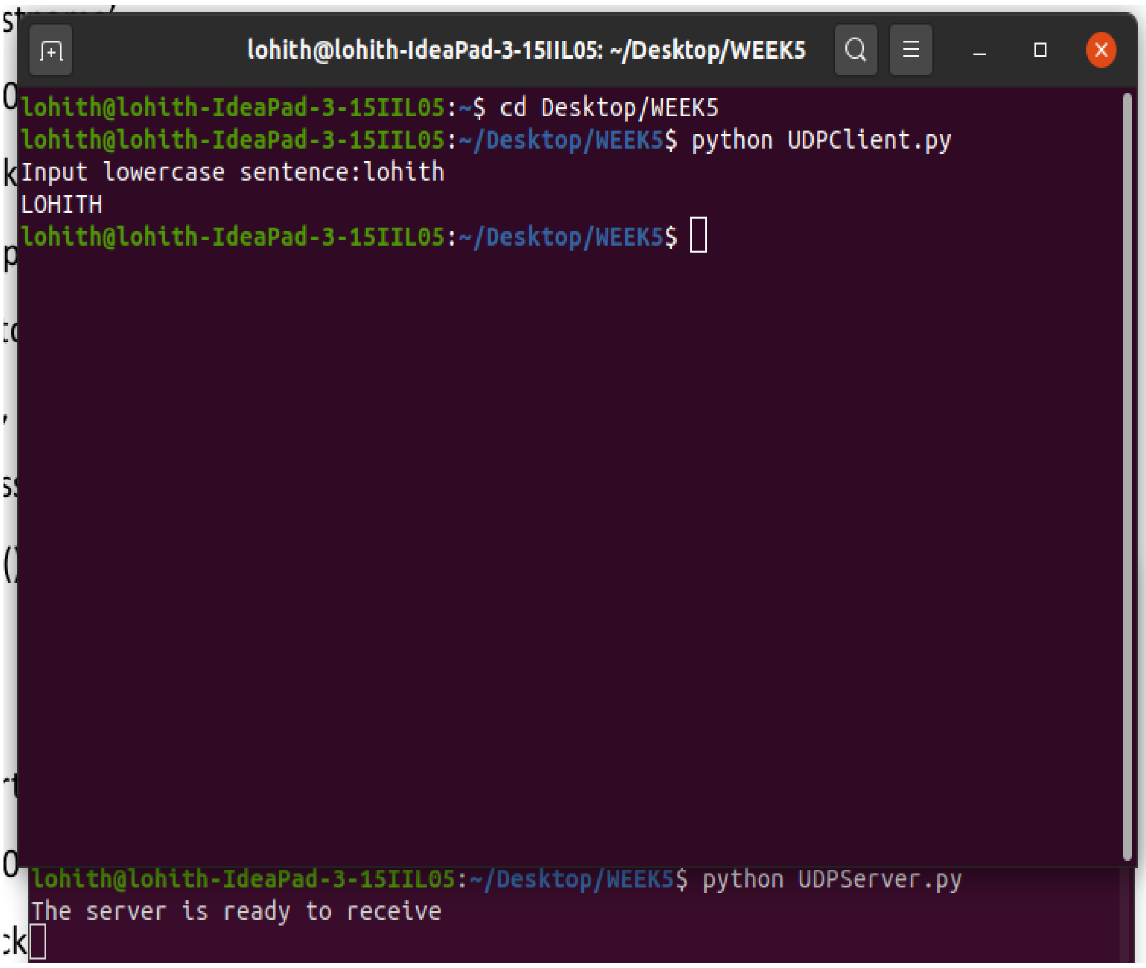
# UDPServer.py



```python
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print("The server is ready to receive")
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```

Loading file "/home/lohith/Desktop/WEEK5/UDPServer.py"...

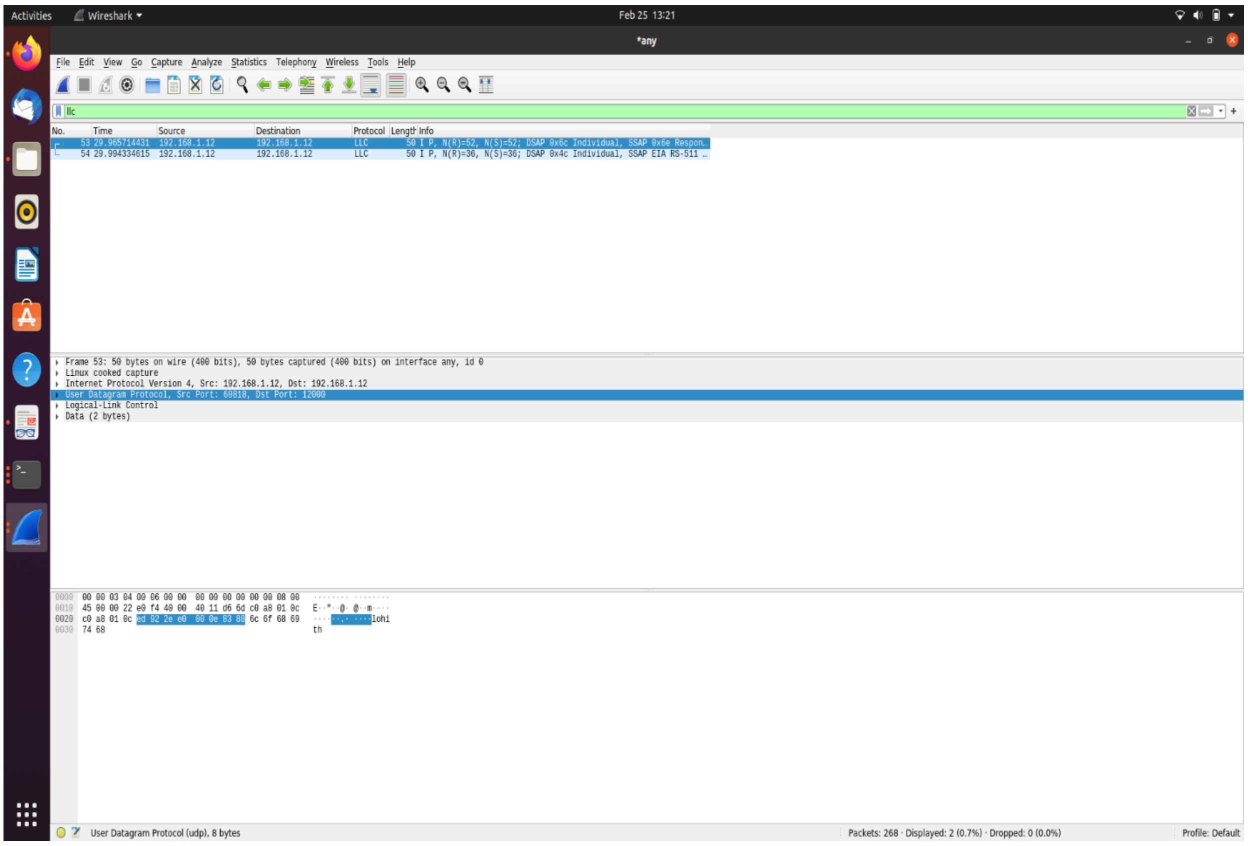
# TERMINAL OUTPUT



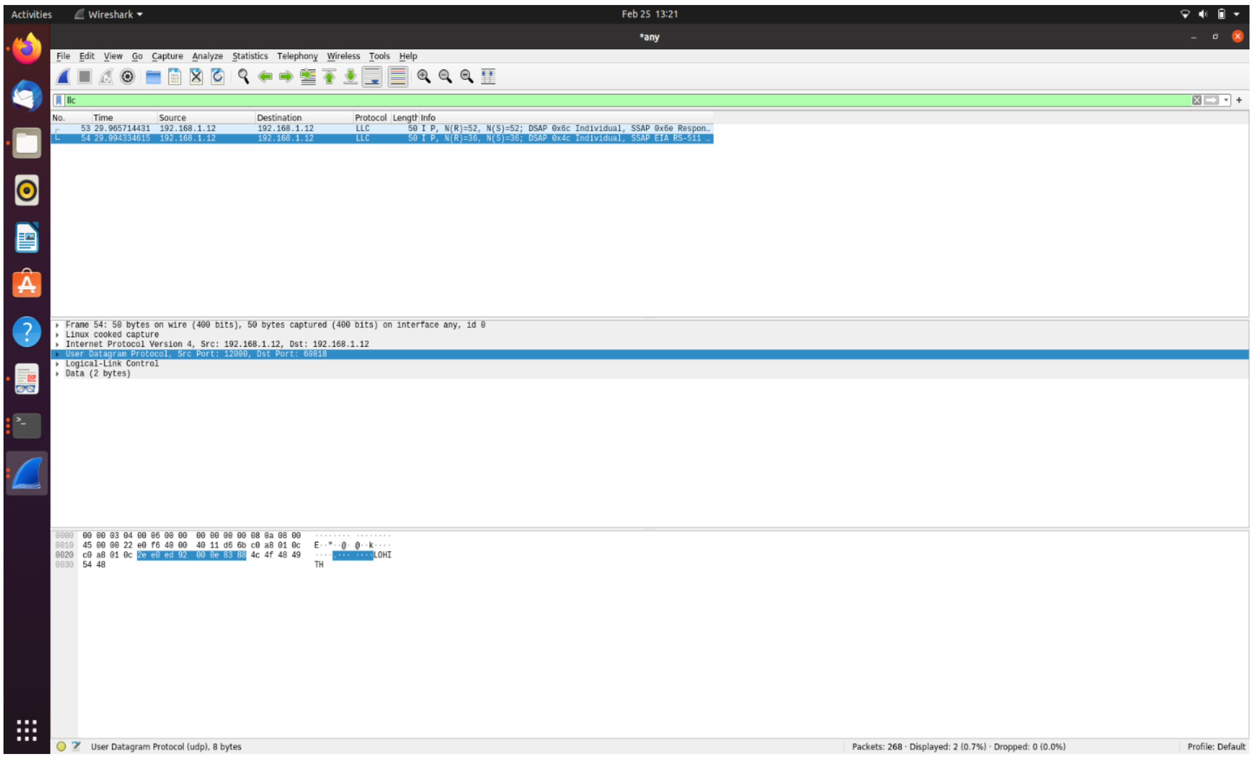
```
lohith@lohith-IdeaPad-3-15IIL05:~$ cd Desktop/WEEK5
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$ python UDPClient.py
Input lowercase sentence:lohith
LOHITH
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$ 
```

```
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$ python UDPServer.py
The server is ready to receive
```

# WIRESHARK CAPTURE SCREENSHOTS


Activities
Wireshark
Feb 25 13:21
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
llc

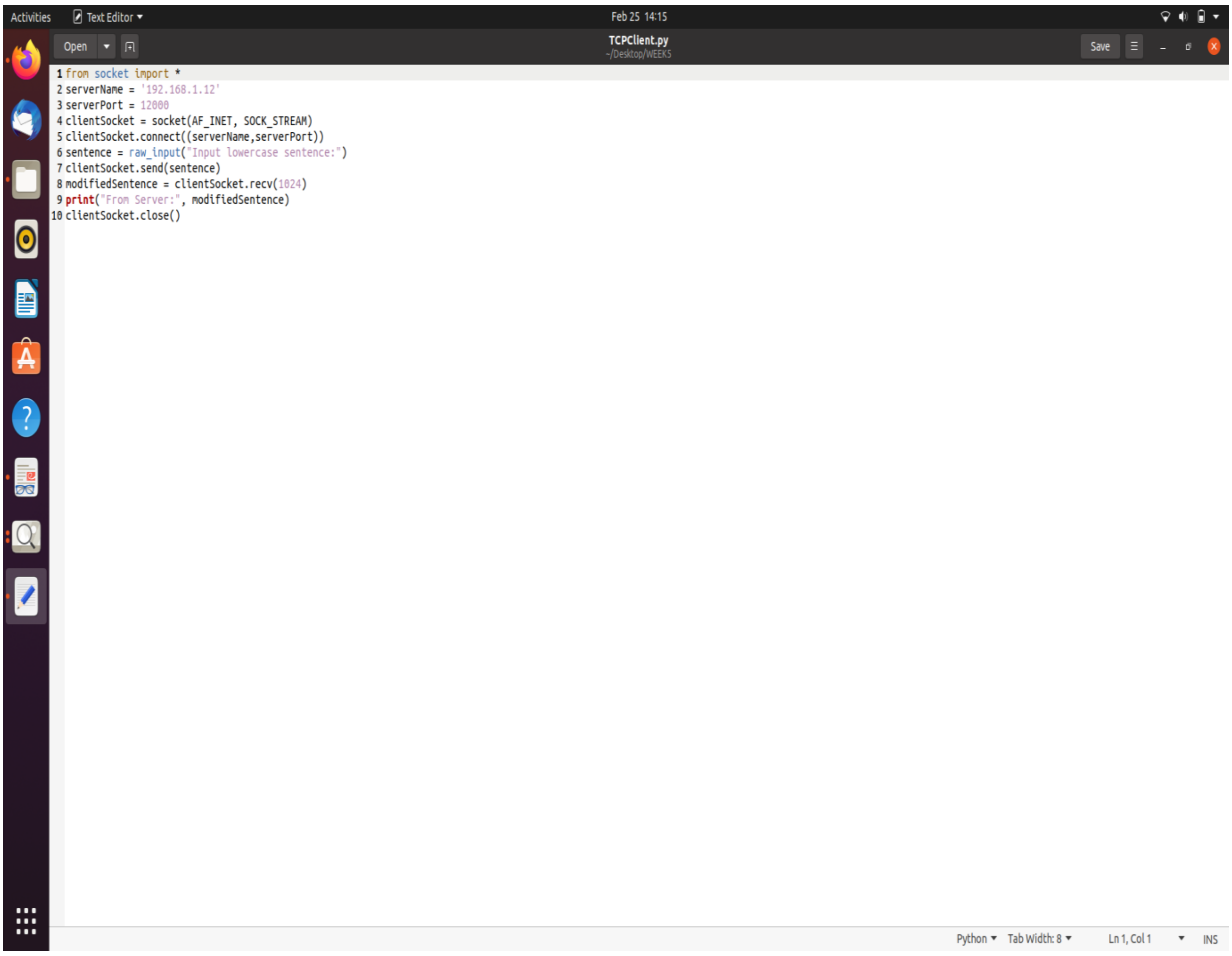
```
No.  Time          Source        Destination   Protocol Length Info
  53 29.965714431  192.168.1.12  192.168.1.12  LLC      50 I P, N(R)=52, N(S)=52; DSAP 0x6c Individual, SSAP 0x6e Respon…
  54 29.994334615  192.168.1.12  192.168.1.12  LLC      50 I P, N(R)=36, N(S)=36; DSAP 0x4c Individual, SSAP EIA RS-511 …

Frame 53: 50 bytes on wire (400 bits), 50 bytes captured (400 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
User Datagram Protocol, Src Port: 60818, Dst Port: 12000
Logical-Link Control
Data (2 bytes)

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 22 e0 f4 40 00  40 11 d6 6d c0 a8 01 0c   E··"··@· @··m····
0020  c0 a8 01 0c ed 92 2e e0  00 0e 83 88 6c 6f 68 69   ······.· ····lohi
0030  74 68                                              th

User Datagram Protocol (udp), 8 bytes        Packets: 268 · Displayed: 2 (0.7%) · Dropped: 0 (0.0%)        Profile: Default
```


Activities
Wireshark
Feb 25 13:21
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
llc

```
No.  Time          Source        Destination   Protocol Length Info
  53 29.965714431  192.168.1.12  192.168.1.12  LLC      50 I P, N(R)=52, N(S)=52; DSAP 0x6c Individual, SSAP 0x6e Respon…
  54 29.994334615  192.168.1.12  192.168.1.12  LLC      50 I P, N(R)=36, N(S)=36; DSAP 0x4c Individual, SSAP EIA RS-511 …

Frame 54: 50 bytes on wire (400 bits), 50 bytes captured (400 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
User Datagram Protocol, Src Port: 12000, Dst Port: 60818
Logical-Link Control
Data (2 bytes)

0000  00 00 03 04 00 06 00 00  00 00 00 00 08 0a 08 00   ········ ········
0010  45 00 00 22 e0 f6 40 00  40 11 d6 6b c0 a8 01 0c   E··"··@· @··k····
0020  c0 a8 01 0c 2e e0 ed 92  00 0e 83 88 4c 4f 48 49   ····.··· ····LOHI
0030  54 48                                              TH

User Datagram Protocol (udp), 8 bytes        Packets: 268 · Displayed: 2 (0.7%) · Dropped: 0 (0.0%)        Profile: Default
```

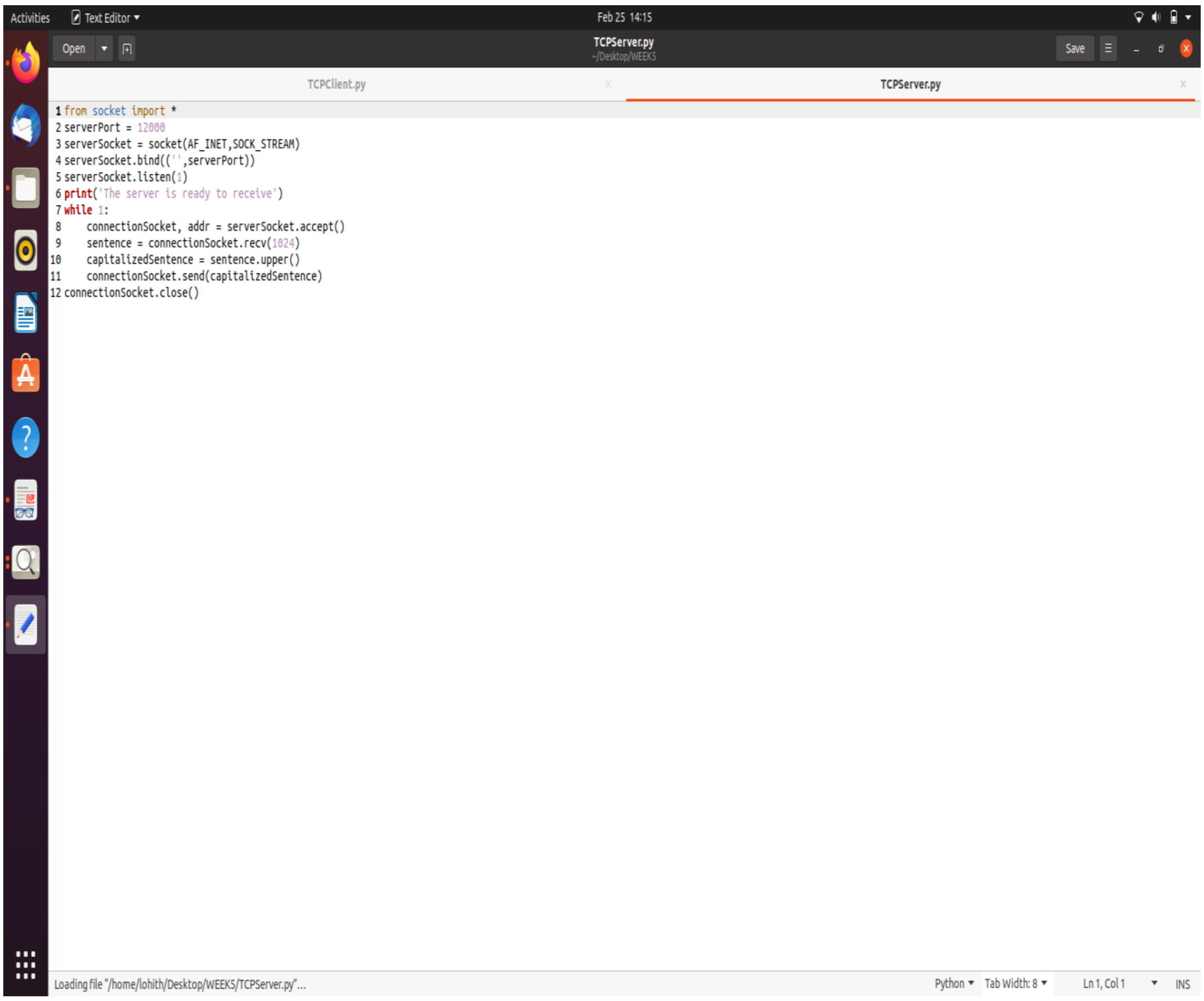
# Socket Programming with TCP

## TCPClient.py



```python
from socket import *
serverName = '192.168.1.12'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = raw_input("Input lowercase sentence:")
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print("From Server:", modifiedSentence)
clientSocket.close()
```

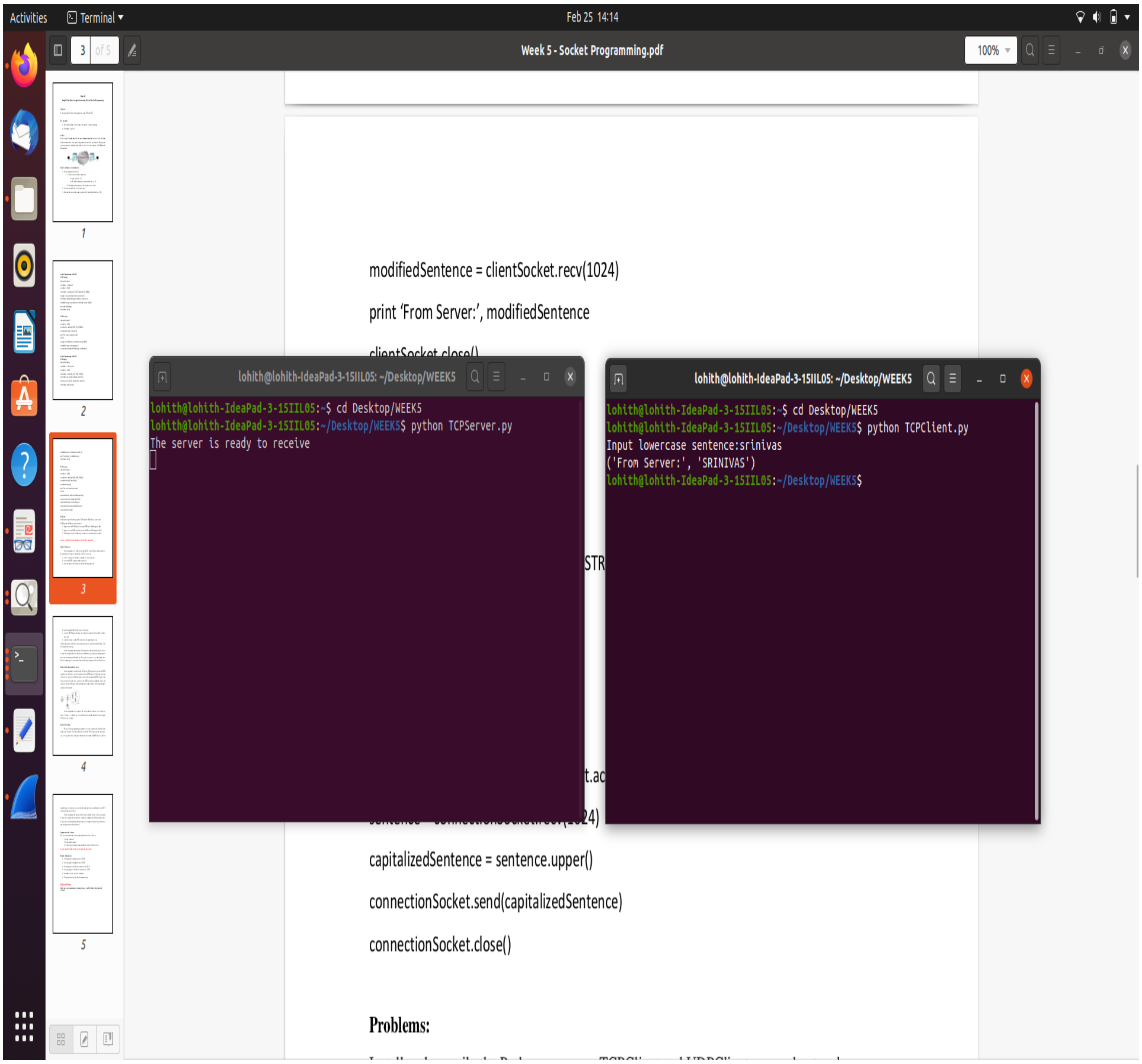
# TCPServer.py



```python
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print('The server is ready to receive')
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
connectionSocket.close()
```

Loading file "/home/lohith/Desktop/WEEK5/TCPServer.py"...

# TERMINAL OUTPUT


Activities
Terminal
Feb 25 14:14
Week 5 - Socket Programming.pdf
100%
lohith@lohith-IdeaPad-3-15IIL05: ~/Desktop/WEEK5

```
modifiedSentence = clientSocket.recv(1024)

print 'From Server:', modifiedSentence
```

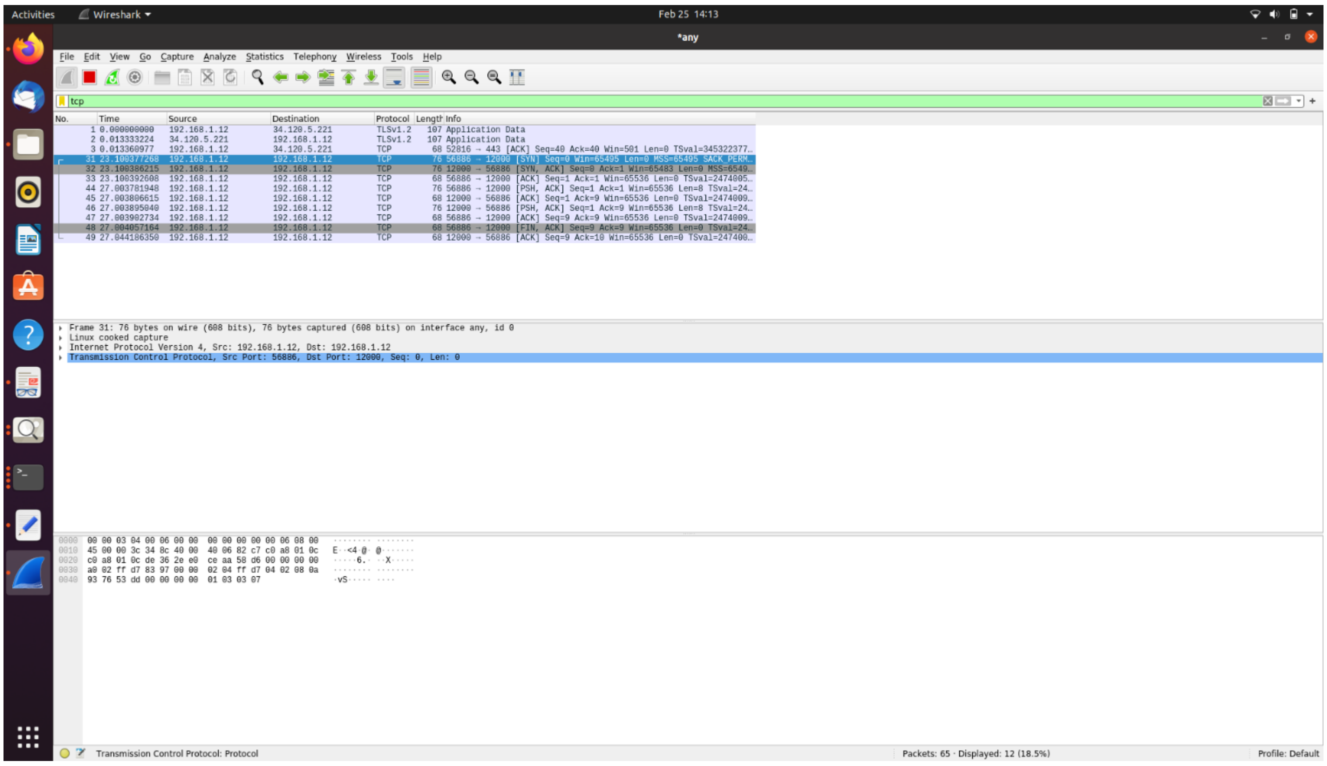
```
lohith@lohith-IdeaPad-3-15IIL05:~$ cd Desktop/WEEK5
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$ python TCPServer.py
The server is ready to receive
```

```
lohith@lohith-IdeaPad-3-15IIL05:~$ cd Desktop/WEEK5
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$ python TCPClient.py
Input lowercase sentence:srinivas
('From Server:', 'SRINIVAS')
lohith@lohith-IdeaPad-3-15IIL05:~/Desktop/WEEK5$
```

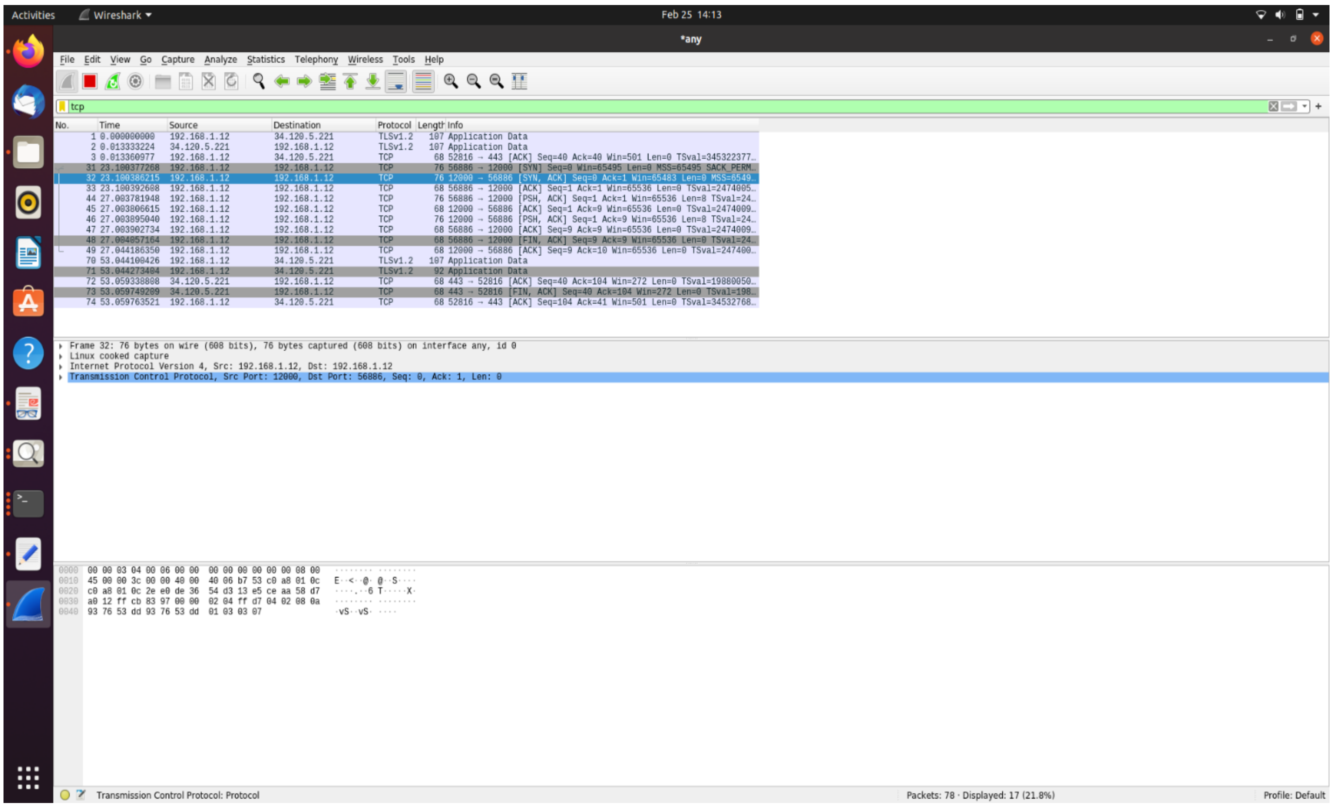
```
capitalizedSentence = sentence.upper()

connectionSocket.send(capitalizedSentence)

connectionSocket.close()
```

**Problems:**

# WIRESHARK CAPTURE



```
No.  Time          Source         Destination    Protocol Length Info
   1 0.000000000   192.168.1.12   34.120.5.221   TLSv1.2  107 Application Data
   2 0.013333224   34.120.5.221   192.168.1.12   TLSv1.2  107 Application Data
   3 0.013360977   192.168.1.12   34.120.5.221   TCP       68 52816 → 443 [ACK] Seq=40 Ack=40 Win=501 Len=0 TSval=345322377…
  31 23.100377268  192.168.1.12   192.168.1.12   TCP       76 56886 → 12000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM…
  32 23.100386215  192.168.1.12   192.168.1.12   TCP       76 12000 → 56886 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=6549…
  33 23.100392608  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2474005…
  44 27.003781948  192.168.1.12   192.168.1.12   TCP       76 56886 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=24…
  45 27.003806615  192.168.1.12   192.168.1.12   TCP       68 12000 → 56886 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=2474009…
  46 27.003895040  192.168.1.12   192.168.1.12   TCP       76 12000 → 56886 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=8 TSval=24…
  47 27.003902734  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [ACK] Seq=9 Ack=9 Win=65536 Len=0 TSval=2474009…
  48 27.004057164  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [FIN, ACK] Seq=9 Ack=9 Win=65536 Len=0 TSval=24…
  49 27.044186350  192.168.1.12   192.168.1.12   TCP       68 12000 → 56886 [ACK] Seq=9 Ack=10 Win=65536 Len=0 TSval=247400…

▸ Frame 31: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
▸ Transmission Control Protocol, Src Port: 56886, Dst Port: 12000, Seq: 0, Len: 0

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 06 08 00   ········ ········
0010  45 00 00 3c 34 8c 40 00  40 06 82 c7 c0 a8 01 0c   E··<4·@· @·······
0020  c0 a8 01 0c de 36 2e e0  ce aa 58 d6 00 00 00 00   ·····6.· ··X·····
0030  a0 02 ff d7 83 97 00 00  02 04 ff d7 04 02 08 0a   ········ ········
0040  93 76 53 dd 00 00 00 00  01 03 03 07               ·vS····· ····

Transmission Control Protocol: Protocol        Packets: 65 · Displayed: 12 (18.5%)        Profile: Default
```



```
No.  Time          Source         Destination    Protocol Length Info
   1 0.000000000   192.168.1.12   34.120.5.221   TLSv1.2  107 Application Data
   2 0.013333224   34.120.5.221   192.168.1.12   TLSv1.2  107 Application Data
   3 0.013360977   192.168.1.12   34.120.5.221   TCP       68 52816 → 443 [ACK] Seq=40 Ack=40 Win=501 Len=0 TSval=345322377…
  31 23.100377268  192.168.1.12   192.168.1.12   TCP       76 56886 → 12000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM…
  32 23.100386215  192.168.1.12   192.168.1.12   TCP       76 12000 → 56886 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=6549…
  33 23.100392608  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2474005…
  44 27.003781948  192.168.1.12   192.168.1.12   TCP       76 56886 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8 TSval=24…
  45 27.003806615  192.168.1.12   192.168.1.12   TCP       68 12000 → 56886 [ACK] Seq=1 Ack=9 Win=65536 Len=0 TSval=2474009…
  46 27.003895040  192.168.1.12   192.168.1.12   TCP       76 12000 → 56886 [PSH, ACK] Seq=1 Ack=9 Win=65536 Len=8 TSval=24…
  47 27.003902734  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [ACK] Seq=9 Ack=9 Win=65536 Len=0 TSval=2474009…
  48 27.004057164  192.168.1.12   192.168.1.12   TCP       68 56886 → 12000 [FIN, ACK] Seq=9 Ack=9 Win=65536 Len=0 TSval=24…
  49 27.044186350  192.168.1.12   192.168.1.12   TCP       68 12000 → 56886 [ACK] Seq=9 Ack=10 Win=65536 Len=0 TSval=247400…
  70 53.044100426  192.168.1.12   34.120.5.221   TLSv1.2  107 Application Data
  71 53.044273404  192.168.1.12   34.120.5.221   TLSv1.2   92 Application Data
  72 53.059338808  34.120.5.221   192.168.1.12   TCP       68 443 → 52816 [ACK] Seq=40 Ack=104 Win=272 Len=0 TSval=19880050…
  73 53.059749209  34.120.5.221   192.168.1.12   TCP       68 443 → 52816 [FIN, ACK] Seq=40 Ack=104 Win=272 Len=0 TSval=198…
  74 53.059763521  192.168.1.12   34.120.5.221   TCP       68 52816 → 443 [ACK] Seq=104 Ack=41 Win=501 Len=0 TSval=34532768…

▸ Frame 32: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
▸ Transmission Control Protocol, Src Port: 12000, Dst Port: 56886, Seq: 0, Ack: 1, Len: 0

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 00 3c 00 00 40 00  40 06 b7 53 c0 a8 01 0c   E··<··@· @··S····
0020  c0 a8 01 0c 2e e0 de 36  54 d3 13 e5 ce aa 58 d7   ····.··6 T·····X·
0030  a0 12 ff cb 83 97 00 00  02 04 ff d7 04 02 08 0a   ········ ········
0040  93 76 53 dd 93 76 53 dd  01 03 03 07               ·vS··vS· ····

Transmission Control Protocol: Protocol        Packets: 78 · Displayed: 17 (21.8%)        Profile: Default
```

1. Suppose you run TCPClient before you run TCPServer. What happens? Why?

This will lead to a ConnectionRefusedError, since the server socket application we are trying to connect to has not been initiated and is not listening for connections on the given port number. Hence, any connection requests sent by a client machine at that IP and port number immediately fail since the connection gets refused.

A TCP connection can be established between two socket interfaces only when a host machine listens to requests on a given IP address and port number and accepts connections made by another machine at the same address and port.

2. Suppose you run UDPClient before you run UDPServer. What happens? Why?

No error will be obtained since UDP does not require a prior connection to be set up between the host machines for data transfer to begin.

It is a connectionless protocol which transfers packets of data to a destination IP and port number without verifying the existence of the connection

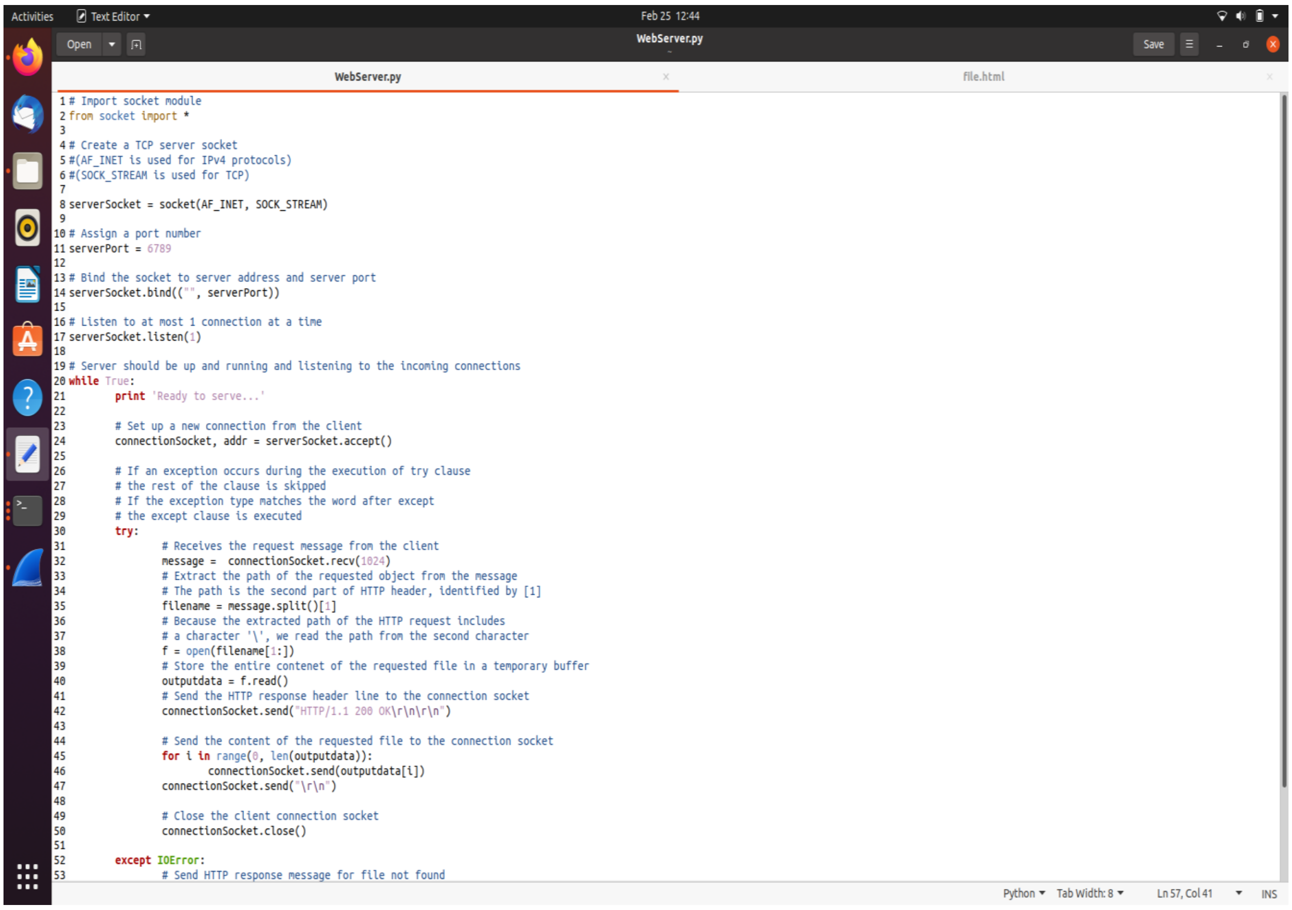
3. What happens if you use different port numbers for the client and server sides?

**This will lead to a ConnectionRefusedError for a TCP connection**

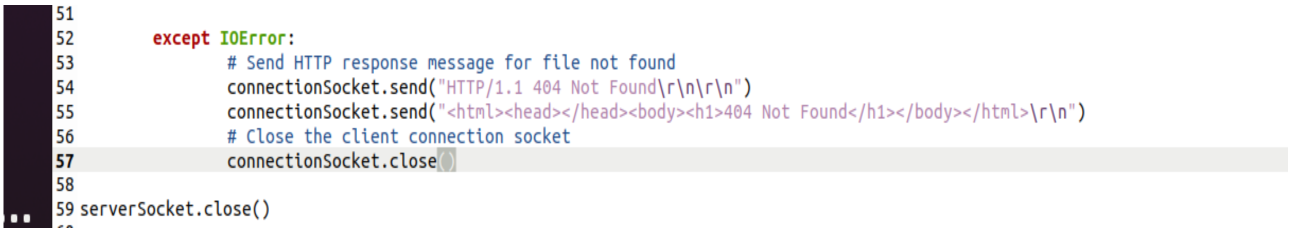
**On a UDP connection, since no prior connection is required to be established between the host machines for data transfer to take place, no error as such is obtained,but any messages sent by the client are lost.**

## Task 2: Web Server

## WebServer.py

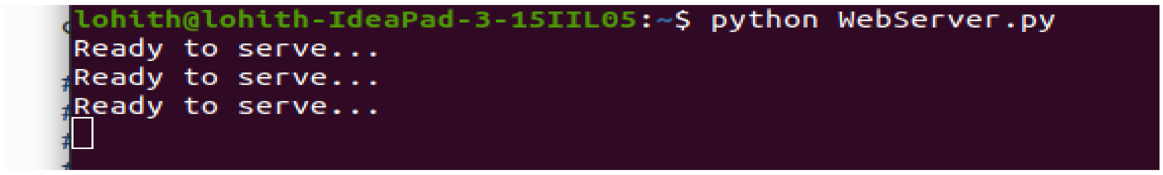


```python
# Import socket module
from socket import *

# Create a TCP server socket
#(AF_INET is used for IPv4 protocols)
#(SOCK_STREAM is used for TCP)

serverSocket = socket(AF_INET, SOCK_STREAM)

# Assign a port number
serverPort = 6789

# Bind the socket to server address and server port
serverSocket.bind(("", serverPort))

# Listen to at most 1 connection at a time
serverSocket.listen(1)

# Server should be up and running and listening to the incoming connections
while True:
        print 'Ready to serve...'

        # Set up a new connection from the client
        connectionSocket, addr = serverSocket.accept()

        # If an exception occurs during the execution of try clause
        # the rest of the clause is skipped
        # If the exception type matches the word after except
        # the except clause is executed
        try:
                # Receives the request message from the client
                message =  connectionSocket.recv(1024)
                # Extract the path of the requested object from the message
                # The path is the second part of HTTP header, identified by [1]
                filename = message.split()[1]
                # Because the extracted path of the HTTP request includes
                # a character '\', we read the path from the second character
                f = open(filename[1:])
                # Store the entire contenet of the requested file in a temporary buffer
                outputdata = f.read()
                # Send the HTTP response header line to the connection socket
                connectionSocket.send("HTTP/1.1 200 OK\r\n\r\n")

                # Send the content of the requested file to the connection socket
                for i in range(0, len(outputdata)):
                        connectionSocket.send(outputdata[i])
                connectionSocket.send("\r\n")

                # Close the client connection socket
                connectionSocket.close()

        except IOError:
                # Send HTTP response message for file not found
```



```python

        except IOError:
                # Send HTTP response message for file not found
                connectionSocket.send("HTTP/1.1 404 Not Found\r\n\r\n")
                connectionSocket.send("<html><head></head><body><h1>404 Not Found</h1></body></html>\r\n")
                # Close the client connection socket
                connectionSocket.close()

serverSocket.close()
```

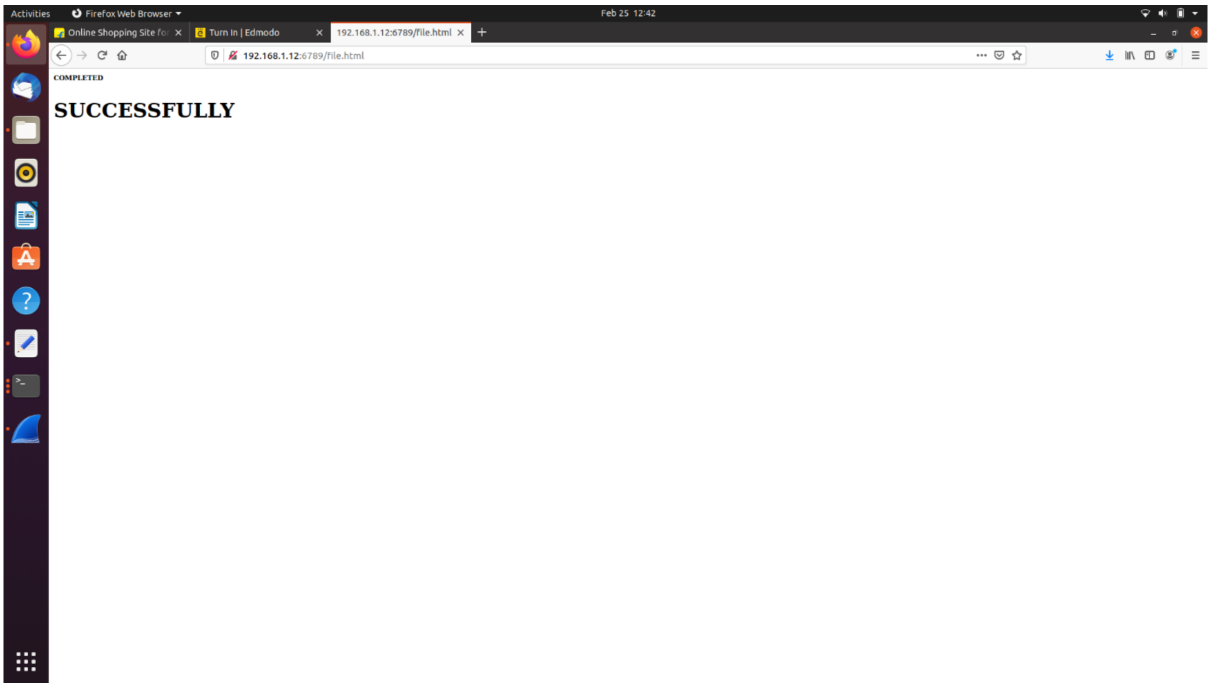
## TERMINAL OUTPUT



```
lohith@lohith-IdeaPad-3-15IIL05:~$ python WebServer.py
Ready to serve...
Ready to serve...
Ready to serve...
```

## 1.VALID HTML FILE

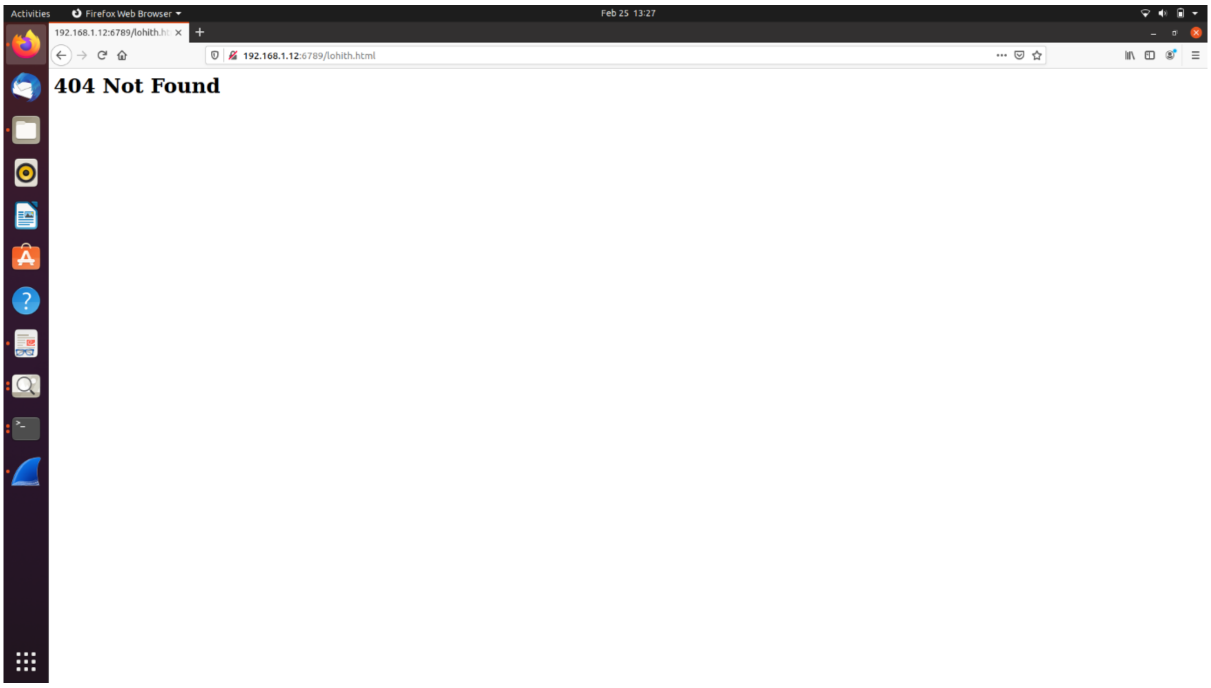
## 2.UNAVAILABLE HTML FILE



```
192.168.1.12:6789/file.html
```

COMPLETED

SUCCESSFULLY



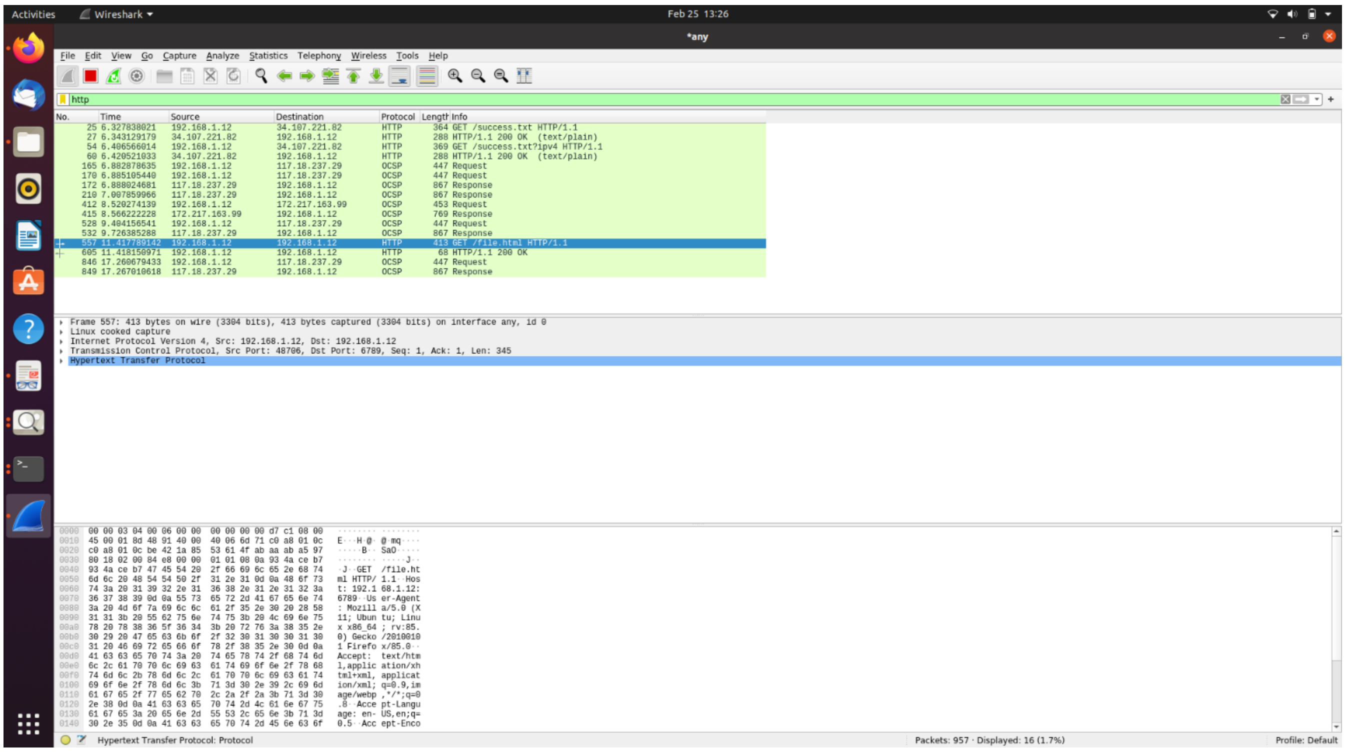
```
192.168.1.12:6789/lohith.html
```

404 Not Found

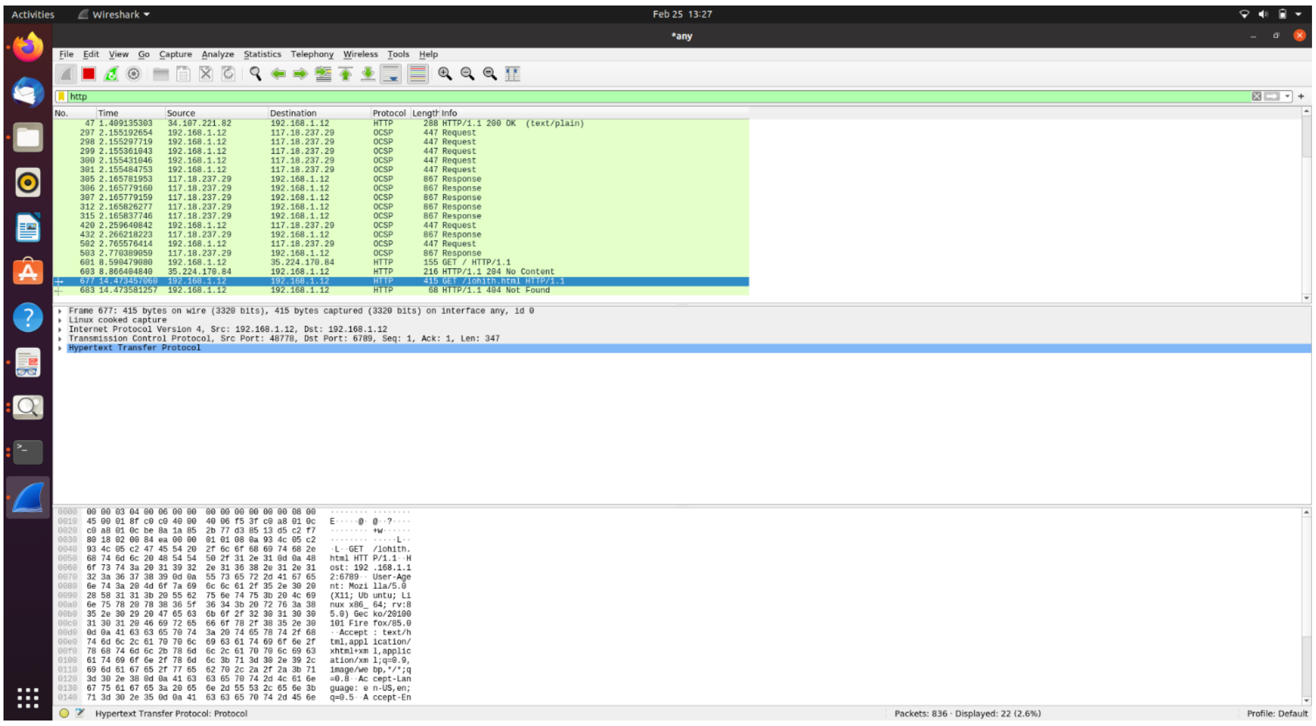
# WIRESHARK CAPTURE

## 1.VALID HTML FILE

## 2.UNAVAILABLE HTML FILE


Activities
Wireshark
Feb 25 13:26
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time          Source          Destination     Protocol Length Info
   25 6.327838021   192.168.1.12    34.107.221.82   HTTP     364 GET /success.txt HTTP/1.1
   27 6.343129179   34.107.221.82   192.168.1.12    HTTP     288 HTTP/1.1 200 OK  (text/plain)
   54 6.406566014   192.168.1.12    34.107.221.82   HTTP     369 GET /success.txt?ipv4 HTTP/1.1
   60 6.420521033   34.107.221.82   192.168.1.12    HTTP     288 HTTP/1.1 200 OK  (text/plain)
  165 6.882878635   192.168.1.12    117.18.237.29   OCSP     447 Request
  170 6.885105440   192.168.1.12    117.18.237.29   OCSP     447 Request
  172 6.888024681   117.18.237.29   192.168.1.12    OCSP     867 Response
  210 7.007859966   117.18.237.29   192.168.1.12    OCSP     867 Response
  412 8.520274139   192.168.1.12    172.217.163.99  OCSP     453 Request
  415 8.566222228   172.217.163.99  192.168.1.12    OCSP     769 Response
  528 9.404156541   192.168.1.12    117.18.237.29   OCSP     447 Request
  532 9.726385288   117.18.237.29   192.168.1.12    OCSP     867 Response
  557 11.417789142  192.168.1.12    192.168.1.12    HTTP     413 GET /file.html HTTP/1.1
  605 11.418150971  192.168.1.12    192.168.1.12    HTTP      68 HTTP/1.1 200 OK
  846 17.260679433  192.168.1.12    117.18.237.29   OCSP     447 Request
  849 17.267010618  117.18.237.29   192.168.1.12    OCSP     867 Response

Frame 557: 413 bytes on wire (3304 bits), 413 bytes captured (3304 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
Transmission Control Protocol, Src Port: 48706, Dst Port: 6789, Seq: 1, Ack: 1, Len: 345
Hypertext Transfer Protocol

0000  00 00 03 04 00 06 00 00  00 00 00 00 d7 c1 08 00   ········ ········
0010  45 00 01 8d 48 91 40 00  40 06 6d 71 c0 a8 01 0c   E···H·@· @·mq····
0020  c0 a8 01 0c be 42 1a 85  53 61 4f ab aa ab a5 97   ·····B·· SaO·····
0030  80 18 02 00 84 e8 00 00  01 01 08 0a 93 4a ce b7   ········ ·····J··
0040  93 4a ce b7 47 45 54 20  2f 66 69 6c 65 2e 68 74   ·J··GET  /file.ht
0050  6d 6c 20 48 54 54 50 2f  31 2e 31 0d 0a 48 6f 73   ml HTTP/ 1.1··Hos
0060  74 3a 20 31 39 32 2e 31  36 38 2e 31 2e 31 32 3a   t: 192.1 68.1.12:
0070  36 37 38 39 0d 0a 55 73  65 72 2d 41 67 65 6e 74   6789··Us er-Agent
0080  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 58   : Mozill a/5.0 (X
0090  31 31 3b 20 55 62 75 6e  74 75 3b 20 4c 69 6e 75   11; Ubun tu; Linu
00a0  78 20 78 38 36 5f 36 34  3b 20 72 76 3a 38 35 2e   x x86_64 ; rv:85.
00b0  30 29 20 47 65 63 6b 6f  2f 32 30 31 30 30 31 30   0) Gecko /2010010
00c0  31 20 46 69 72 65 66 6f  78 2f 38 35 2e 30 0d 0a   1 Firefo x/85.0··
00d0  41 63 63 65 70 74 3a 20  74 65 78 74 2f 68 74 6d   Accept:  text/htm
00e0  6c 2c 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 68   l,applic ation/xh
00f0  74 6d 6c 2b 78 6d 6c 2c  61 70 70 6c 69 63 61 74   tml+xml, applicat
0100  69 6f 6e 2f 78 6d 6c 3b  71 3d 30 2e 39 2c 69 6d   ion/xml; q=0.9,im
0110  61 67 65 2f 77 65 62 70  2c 2a 2f 2a 3b 71 3d 30   age/webp ,*/*;q=0
0120  2e 38 0d 0a 41 63 63 65  70 74 2d 4c 61 6e 67 75   .8··Acce pt-Langu
0130  61 67 65 3a 20 65 6e 2d  55 53 2c 65 6e 3b 71 3d   age: en- US,en;q=
0140  30 2e 35 0d 0a 41 63 63  65 70 74 2d 45 6e 63 6f   0.5··Acc ept-Enco

Hypertext Transfer Protocol: Protocol        Packets: 957 · Displayed: 16 (1.7%)        Profile: Default
```


Activities
Wireshark
Feb 25 13:27
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time          Source          Destination     Protocol Length Info
   47 1.409135303   34.107.221.82   192.168.1.12    HTTP     288 HTTP/1.1 200 OK  (text/plain)
  297 2.155192654   192.168.1.12    117.18.237.29   OCSP     447 Request
  298 2.155297719   192.168.1.12    117.18.237.29   OCSP     447 Request
  299 2.155361043   192.168.1.12    117.18.237.29   OCSP     447 Request
  300 2.155431046   192.168.1.12    117.18.237.29   OCSP     447 Request
  301 2.155484753   192.168.1.12    117.18.237.29   OCSP     447 Request
  305 2.165781953   117.18.237.29   192.168.1.12    OCSP     867 Response
  306 2.165779160   117.18.237.29   192.168.1.12    OCSP     867 Response
  307 2.165779159   117.18.237.29   192.168.1.12    OCSP     867 Response
  312 2.165826277   117.18.237.29   192.168.1.12    OCSP     867 Response
  315 2.165837746   117.18.237.29   192.168.1.12    OCSP     867 Response
  420 2.259640842   192.168.1.12    117.18.237.29   OCSP     447 Request
  432 2.266218223   117.18.237.29   192.168.1.12    OCSP     867 Response
  502 2.765576414   192.168.1.12    117.18.237.29   OCSP     447 Request
  503 2.770389059   117.18.237.29   192.168.1.12    OCSP     867 Response
  601 8.590479080   192.168.1.12    35.224.170.84   HTTP     155 GET / HTTP/1.1
  603 8.866404840   35.224.170.84   192.168.1.12    HTTP     216 HTTP/1.1 204 No Content
  677 14.473457060  192.168.1.12    192.168.1.12    HTTP     415 GET /lohith.html HTTP/1.1
  683 14.473581257  192.168.1.12    192.168.1.12    HTTP      68 HTTP/1.1 404 Not Found

Frame 677: 415 bytes on wire (3320 bits), 415 bytes captured (3320 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.1.12, Dst: 192.168.1.12
Transmission Control Protocol, Src Port: 48778, Dst Port: 6789, Seq: 1, Ack: 1, Len: 347
Hypertext Transfer Protocol

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ········ ········
0010  45 00 01 8f c0 c0 40 00  40 06 f5 3f c0 a8 01 0c   E·····@· @··?····
0020  c0 a8 01 0c be 8a 1a 85  2b 77 d3 85 13 d5 c2 f7   ········ +w······
0030  80 18 02 00 84 ea 00 00  01 01 08 0a 93 4c 05 c2   ········ ·····L··
0040  93 4c 05 c2 47 45 54 20  2f 6c 6f 68 69 74 68 2e   ·L··GET  /lohith.
0050  68 74 6d 6c 20 48 54 54  50 2f 31 2e 31 0d 0a 48   html HTT P/1.1··H
0060  6f 73 74 3a 20 31 39 32  2e 31 36 38 2e 31 2e 31   ost: 192 .168.1.1
0070  32 3a 36 37 38 39 0d 0a  55 73 65 72 2d 41 67 65   2:6789·· User-Age
0080  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 35 2e 30 20   nt: Mozi lla/5.0
0090  28 58 31 31 3b 20 55 62  75 6e 74 75 3b 20 4c 69   (X11; Ub untu; Li
00a0  6e 75 78 20 78 38 36 5f  36 34 3b 20 72 76 3a 38   nux x86_ 64; rv:8
00b0  35 2e 30 29 20 47 65 63  6b 6f 2f 32 30 31 30 30   5.0) Gec ko/20100
00c0  31 30 31 20 46 69 72 65  66 6f 78 2f 38 35 2e 30   101 Fire fox/85.0
00d0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
00e0  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
00f0  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
0100  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0110  69 6d 61 67 65 2f 77 65  62 70 2c 2a 2f 2a 3b 71   image/we bp,*/*;q
0120  3d 30 2e 38 0d 0a 41 63  63 65 70 74 2d 4c 61 6e   =0.8··Ac cept-Lan
0130  67 75 61 67 65 3a 20 65  6e 2d 55 53 2c 65 6e 3b   guage: e n-US,en;
0140  71 3d 30 2e 35 0d 0a 41  63 63 65 70 74 2d 45 6e   q=0.5··A ccept-En

Hypertext Transfer Protocol: Protocol        Packets: 836 · Displayed: 22 (2.6%)        Profile: Default
```