

Computer networks laboratory week 5

Name: Pranav R. Hegde

SRN: PES1UG19CS343

Section: D

TASK 1: Socket programming(client server app)

Simple client/server application using TCP

The screenshot displays a Linux desktop with a purple background. On the left is a dock with various application icons. The desktop contains several files and folders: 'Week 1 OOAD.docx', 'CN_Lab_5', '1_PES1201800124_OOAD&SE.docx', '1_PES1201800124_OOAD&SE.pdf', 'obs5.png', 'ArmSim', and 'MPCA LAB Week5 Problem Statements_upd...'. Two terminal windows are open. The top terminal, titled 'student@CSELAB: ~', shows the execution of a Python TCP server script, which has successfully started and is ready to receive connections. The bottom terminal, titled 'student@CSELAB: ~/PES1UG19CS343_PranavRHegde', shows the execution of a Python TCP client script. The client has connected to the server and sent the message 'pranav r hegde cn lab week 5 Q#\$@!\$'. The server has responded with 'PRANAV R HEGDE CN LAB WEEK 5 Q#\$@!\$'.

Terminal 1 (Top):

```
student@CSELAB: ~
student@CSELAB:~$ python TCPServer.py
The server is ready to receive
█
```

Terminal 2 (Bottom):

```
student@CSELAB: ~/PES1UG19CS343_PranavRHegde
student@CSELAB:~/PES1UG19CS343_PranavRHegde$ python TCPClient.py
Input lowercase sentence:pranav r hegde cn lab week 5 Q#$@!$
From Server: PRANAV R HEGDE CN LAB WEEK 5 Q#$@!$
student@CSELAB:~/PES1UG19CS343_PranavRHegde$ █
```

Simple client/server application using UDP

The screenshot displays an Ubuntu Desktop environment. On the left, a vertical dock contains icons for the Dash, Home Folder, Firefox, LibreOffice Writer, LibreOffice Calc, LibreOffice Impress, a file manager, a terminal, a web browser, a PDF viewer, and a trash bin. The desktop background is a solid blue color. In the center, a file manager window is open, showing a list of files and folders. The files include 'Week 1 OOAD.docx', '1_PES1201800124_OOAD&SE.docx', '1_PES1201800124_OOAD&SE.pdf', 'obs5.png', 'ArmSim', and 'MPCA LAB Week5 Problem Statements_upd...'. To the right of the file manager, a terminal window is open, displaying the command 'python UDPClient.py' and its output 'The server is ready to receive'.

Ubuntu Desktop

Week 1 OOAD.docx

CN_Lab_5

1_PES1201800124_OOAD&SE.docx

1_PES1201800124_OOAD&SE.pdf

obs5.png

ArmSim

MPCA LAB Week5 Problem Statements_upd...

```
student@CSELAB: ~
student@CSELAB:~$ python UDPClient.py
The server is ready to receive

```

1. Suppose you run TCP Client before you run TCP Server. What happens? Why?

-> If the TCP client is run before the TCP server, the TCP client program throws an error. This is because a server is needed in order to establish a communication between the server and the client. If the client requests for a connection even before the server is active, the client program ends up throwing an error.

2. Suppose you run UDP Client before you run UDP Server. What happens? Why?

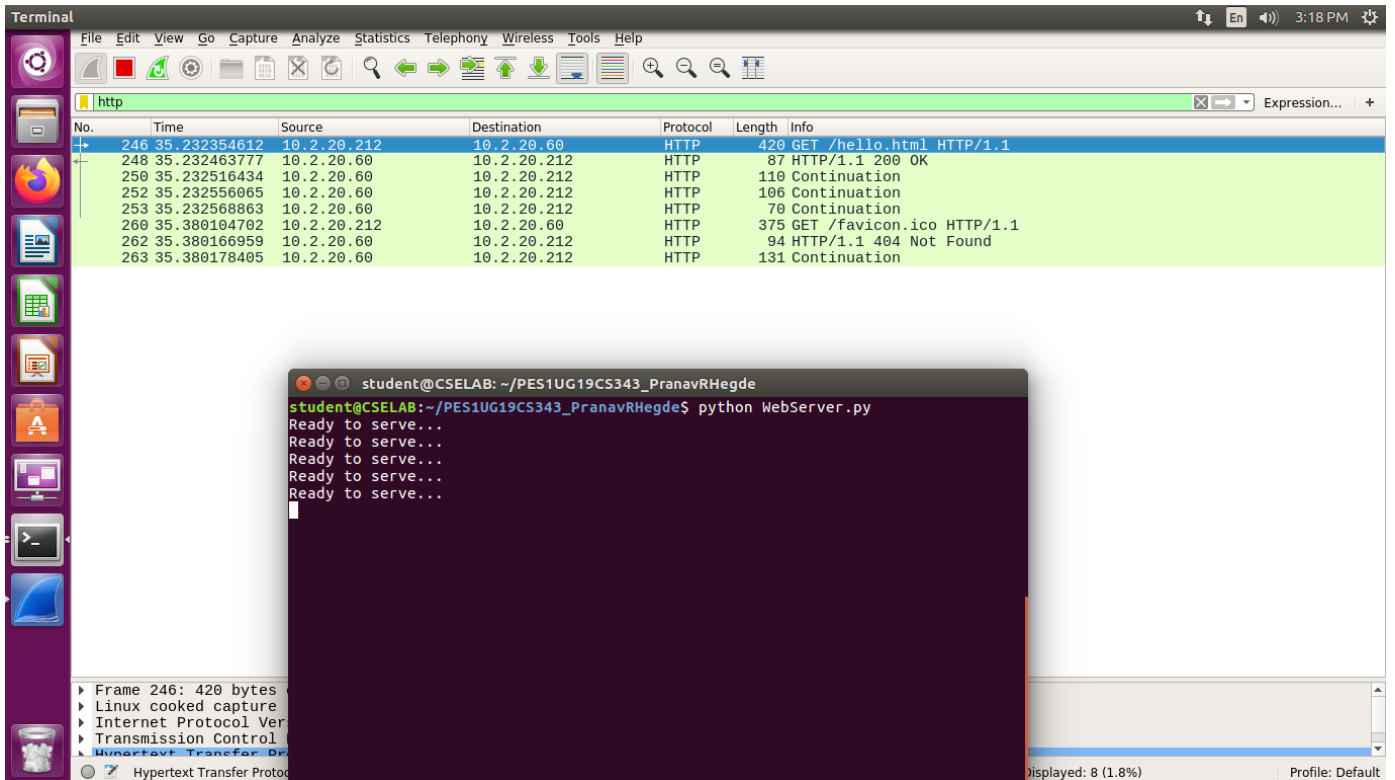
-> If the UDP client is run before the UDP server, the UDP client program throws an error. This is because a server is needed in order to establish a communication between the server and the client. If the client requests for a connection even before the server is active, the client program ends up throwing an error.

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

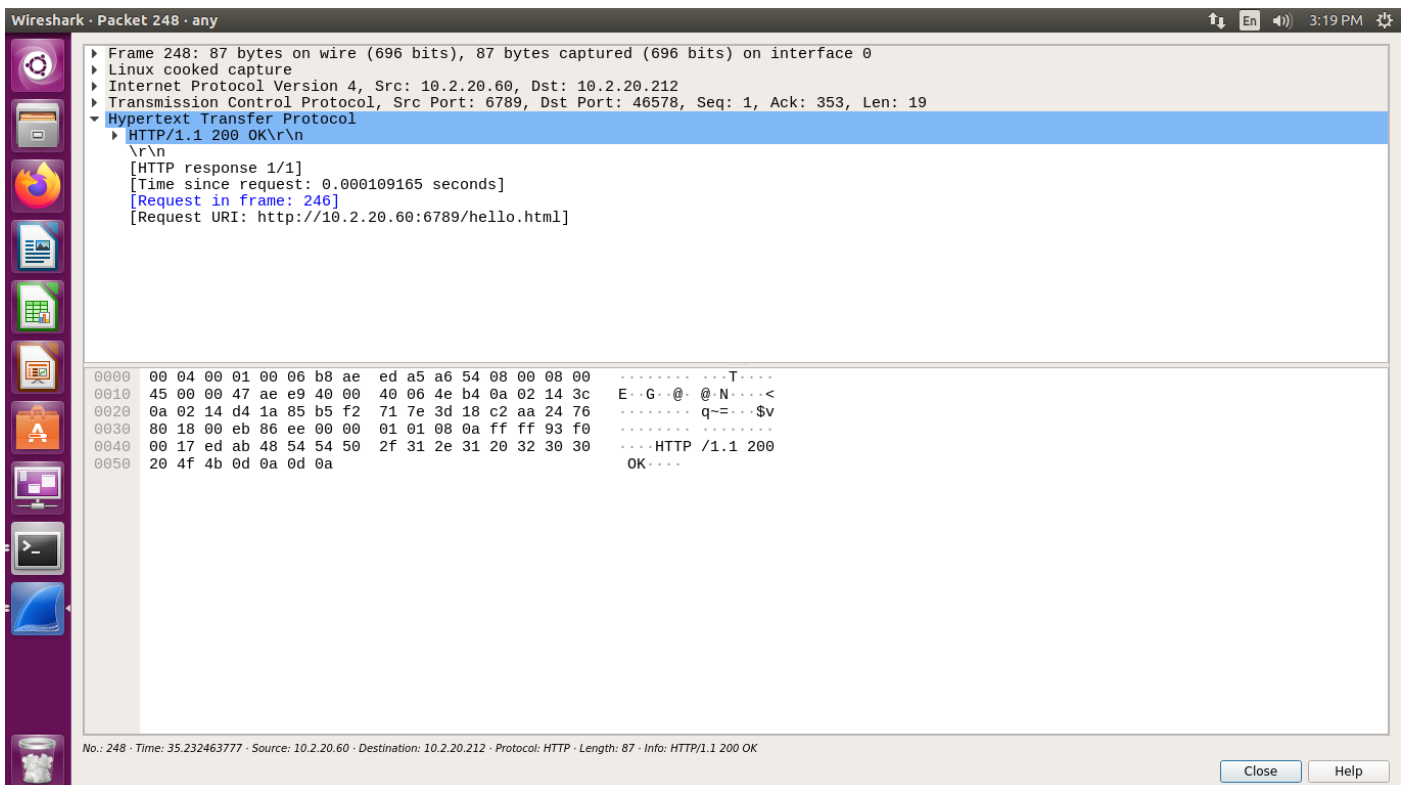
-> If a different port number is used on both the client and the server ends, communication is unsuccessful as port number is associated with each and every process. If two different port numbers are used, the communication is unsuccessful as they're considered to be a part of two different processes.

TASK2: Creating and running a Web server

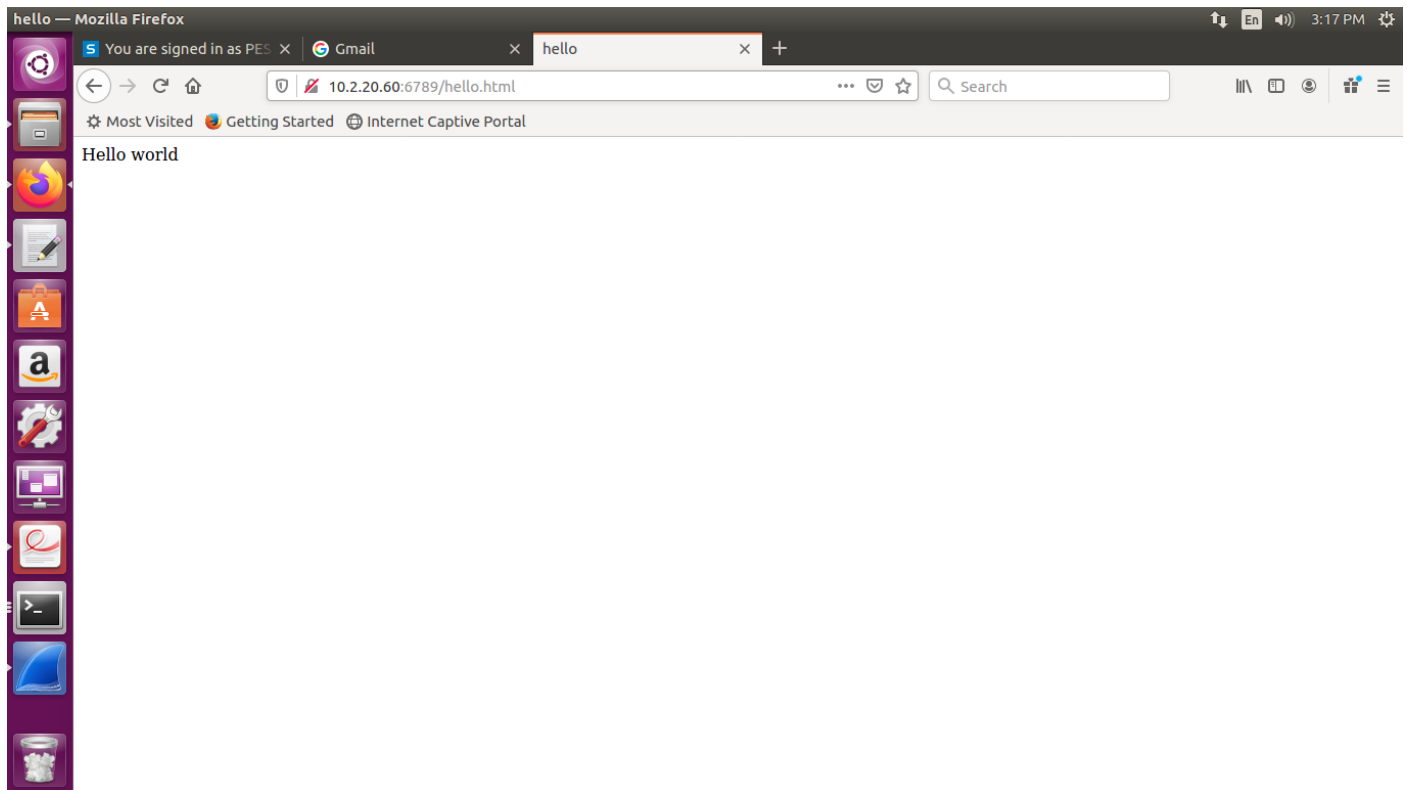
Wireshark capture on the server PC. Server ready to serve.



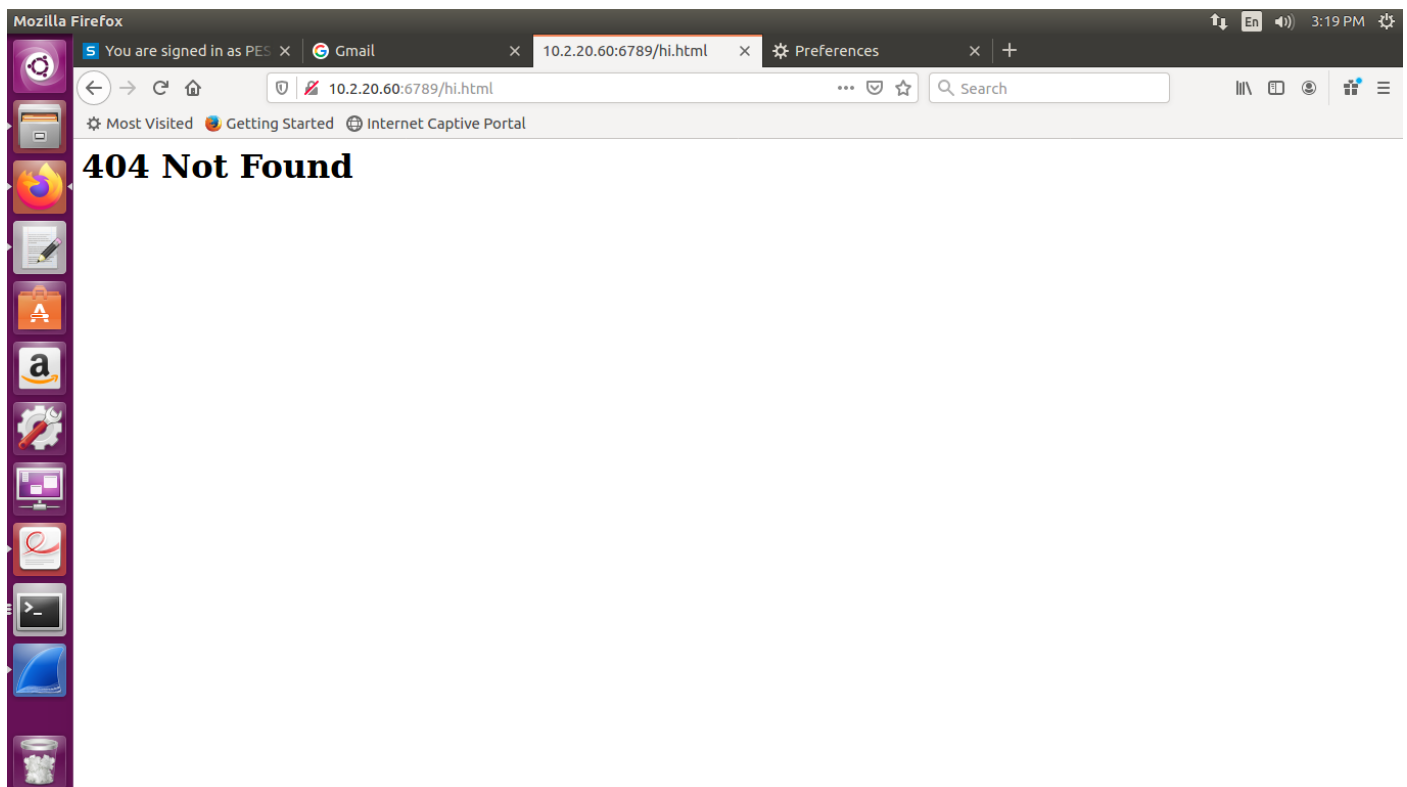
Wireshark packet capture of the first packet sent to the client.



The webpage(hosted by the server machine) being accessed from the client machine.



Accessing the webpage not present on the server machine. The client machine browser throws a 404 not found error.



Wireshark packet capture data for the webpage not found on the server machine.

Wireshark · Packet 104 · any

Frame 104: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits) on interface 0

- Linux cooked capture
- Internet Protocol Version 4, Src: 10.2.20.60, Dst: 10.2.20.212
- Transmission Control Protocol, Src Port: 6789, Dst Port: 46592, Seq: 27, Ack: 376, Len: 63**
- Hypertext Transfer Protocol
 - File Data: 63 bytes
 - Data (63 bytes)
 - Data: 3c68746d6c3e3c686561643e3c2f686561643e3c626f6479...
 - [Length: 63]

0000	00 04 00 01 00 06 b8 ae	ed a5 a6 54 00 00 08 00T....
0010	45 00 00 73 cd f1 40 00	40 06 2f 80 0a 02 14 3c	E..s..@. @ /...<
0020	0a 02 14 d4 1a 85 b6 00	77 9b d9 26 86 3d 2b fbw..&.=+.
0030	80 19 00 eb bb aa 00 00	01 01 08 0a 00 00 14 e3
0040	00 18 6e 9e 3c 68 74 6d	6c 3e 3c 68 65 61 64 3e	..n.<html><head>
0050	3c 2f 68 65 61 64 3e 3c	62 6f 64 79 3e 3c 68 31	</head>< body><h1
0060	3e 34 30 34 20 4e 6f 74	20 46 6f 75 6e 64 3c 2f	>404 Not Found</
0070	68 31 3e 3c 2f 62 6f 64	79 3e 3c 2f 68 74 6d 6c	h1></body></html
0080	3e 0d 0a		>..

No.: 104 · Time: 7.631459040 · Source: 10.2.20.60 · Destination: 10.2.20.212 · Protocol: HTTP · Length: 131 · Info: Continuation

Close Help