

Date: - 01/02/2023

Kanwar raj Singh (1903122) &

PART A

1. We got following Output after running given python files

The image shows two terminal windows side-by-side. The left window shows the execution of a UDP client program. The right window shows the execution of a UDP server program.

```

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Client.py
Client received MESSAGE= Ok from ADDR= ('127.0.0.1', 43387)
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 --version
Python 3.10.6
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Server.py
Server started...waiting for a connection from the client
Server received MESSAGE= I could tell you a UDP joke but I'm not sure you'll
get it. from ADDR= ('127.0.0.1', 58866)
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$
  
```

2. I run UDP_Client.py in my computer and got following output:

```

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Client.py
Client received MESSAGE= Ok from ADDR= ('10.196.12.239', 43387)
  
```

Output of another computer:

```

PS C:\Users\Tushar Jain\Downloads> python .\UDP_Server.py
Server started...waiting for a connection from the client
Server received MESSAGE= I could tell you a UDP joke but I'm not sure you'll get it. from ADDR= ('10.196.10.43', 56648)
  
```

3. I used following code for sending date to client in Server file.

```

from datetime import date
dateAsString = str(date.today())

client.send(dateAsString.encode())
  
```

```

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Client3.py
Client received MESSAGE= 2023-02-01 from ADDR= ('127.0.0.1', 43387)
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Server.py
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 UDP_Server.py
Server started...waiting for a connection from the client
Server received MESSAGE= Send Date, from ADDR= ('127.0.0.1', 57658)
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$

```

PART B

4. Following output, I got in terminal

```

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 TCP_client.py
CLIENT RECEIVED: Whose there?
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$

rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$ python3 TCP_server.py
Server started...waiting for a connection from the client
Connection initiated from ('127.0.0.1', 49274)
SERVER RECEIVED: Knock knock..
rajkanwar@Kanwarraj:/mnt/c/Users/Student/Desktop/CS212$

```

5. In the Wireshark, we started capturing the packets and applied the filter **tcp.port==43389**. Then we ran the TCP server and TCP client files in different windows of terminal. After the execution of the program, we investigated the contents of the packets captured. As we have not used HTTPS protocol, the messages were visible to us in the packet descriptions (the images of which are attached below).

Packet captured: -

No.	Time	Source	Destination	Protocol	Length	Info
314	149.443264119	127.0.0.1	127.0.0.1	TCP	76	49642 → 43389 [SYN] Seq=0 Win=65495 Len=0 M
315	149.443274176	127.0.0.1	127.0.0.1	TCP	76	43389 → 49642 [SYN, ACK] Seq=0 Ack=1 Win=65
316	149.443280873	127.0.0.1	127.0.0.1	TCP	68	49642 → 43389 [ACK] Seq=1 Ack=1 Win=65536 L
317	149.443323271	127.0.0.1	127.0.0.1	TCP	81	49642 → 43389 [PSH, ACK] Seq=1 Ack=1 Win=65
318	149.443347944	127.0.0.1	127.0.0.1	TCP	68	43389 → 49642 [ACK] Seq=1 Ack=14 Win=65536
319	149.443427782	127.0.0.1	127.0.0.1	TCP	80	43389 → 49642 [PSH, ACK] Seq=1 Ack=14 Win=6
320	149.443430810	127.0.0.1	127.0.0.1	TCP	68	49642 → 43389 [ACK] Seq=14 Ack=13 Win=65536
321	149.443463612	127.0.0.1	127.0.0.1	TCP	68	43389 → 49642 [FIN, ACK] Seq=13 Ack=14 Win=
322	149.443486533	127.0.0.1	127.0.0.1	TCP	68	49642 → 43389 [FIN, ACK] Seq=14 Ack=14 Win=
323	149.443490092	127.0.0.1	127.0.0.1	TCP	68	43389 → 49642 [ACK] Seq=14 Ack=15 Win=65536

Message received by server: -

```
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 06 08 00
0010 45 00 00 41 ff b3 40 00 40 06 3d 01 7f 00 00 01
0020 7f 00 00 01 c1 ea a9 7d e0 09 6d 45 d7 b4 2d 8a
0030 80 18 02 00 fe 35 00 00 01 01 08 0a 03 68 f7 cb
0040 03 68 f7 cb 4b 6e 6f 63 6b 20 6b 6e 6f 63 6b 2e
0050 2e
```

```
.....
E..A..@..@..=.....
.....}.....mE...
.....5.....h...
-h..Knoc k knock..
.
```

Message received by client: -

```
0000 00 00 03 04 00 06 00 00 00 00 00 00 64 00 08 00
0010 45 00 00 40 3f 16 40 00 40 06 fd 9f 7f 00 00 01
0020 7f 00 00 01 a9 7d c1 ea d7 b4 2d 8a e0 09 6d 52
0030 80 18 02 00 fe 34 00 00 01 01 08 0a 03 68 f7 cb
0040 03 68 f7 cb 57 68 6f 73 65 20 74 68 65 72 65 3f
```

```
.....d...
E..@?..@..@.....
.....}.....mR
.....4.....h...
-h..Whos e there?
```

6. Output:

```
rajkanwar@Kanwarraj: /mnt/c/
y
Server started...waiting for a connection from the client
Connection initiated and adress is: ('127.0.0.1', 43948)
Current user is : Kanwar
SERVER RECEIVED: 5 + 4
Answer Sent!
SERVER RECEIVED: 7 / -10
Answer Sent!
SERVER RECEIVED: q
Kanwar left!
rajkanwar@Kanwarraj: /mnt/c/Users/Student/Desktop/CS212$
rajkanwar@Kanwarraj: /mnt/c/Users/Student/Desktop/CS212$

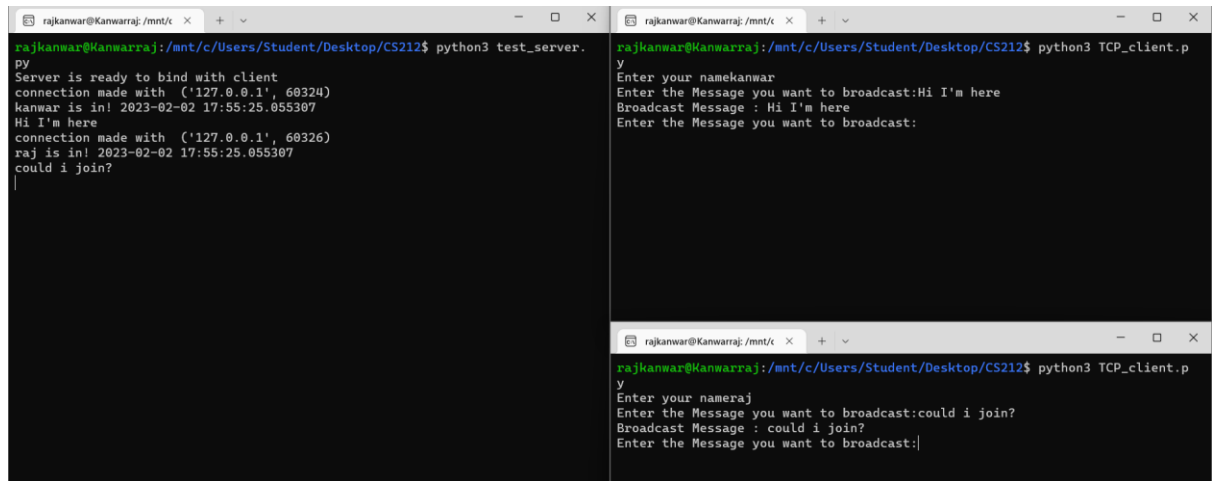
rajkanwar@Kanwarraj: /mnt/c/Users/Student/Desktop/CS212$ python3 TCP_client.p
y
Enter your nameKanwar
Enter the string5 + 4
Warning! Input is not correct
RECEIVED Answer from Server is : 9.0
Great This is Correct!
Enter the string7 / -10
Warning! Input is not correct
RECEIVED Answer from Server is : -0.7
Great This is Correct!
Enter the stringq
BYE
rajkanwar@Kanwarraj: /mnt/c/Users/Student/Desktop/CS212$
```

7. A **stateless protocol** is a communication protocol in which the receiver must not retain session state from previous requests. A protocol is stateful if any information about the connection is remembered. Whereas in a stateless communication, no information related to the connection is remembered.

- Yes, the application-layer protocol designed by us for question 3 is Stateless as no information is remembered about any client. Whenever a client sends a request, server sends the message without any evaluation of prior information.

- No, the application-layer protocol designed by us for question 6 is not Stateless as the Server remembers the name sent by the client for the entire session.
- No, TCP is a Stateful protocol as a particular socket is assigned to every client and this socket is remembered throughout the session with the client.
- Yes, UDP is a stateless protocol as messages can be sent between any 2 sockets just by using the opposite one's Ip and port no. and as no remembering is happening.

8. Output which I got,



The image shows two terminal windows side-by-side. The left window shows the output of running a Python server script, and the right window shows the output of running a Python client script.

```

rajkanwar@Kanwarraj:/mnt/c$ python3 test_server.py
Server is ready to bind with client
connection made with ('127.0.0.1', 60324)
kanwar is in! 2023-02-02 17:55:25.055307
Hi I'm here
connection made with ('127.0.0.1', 60326)
raj is in! 2023-02-02 17:55:25.055307
could i join?

```

```

rajkanwar@Kanwarraj:/mnt/c$ python3 TCP_client.py
Enter your namekanwar
Enter the Message you want to broadcast:Hi I'm here
Broadcast Message : Hi I'm here
Enter the Message you want to broadcast:

rajkanwar@Kanwarraj:/mnt/c$ python3 TCP_client.py
Enter your nameraj
Enter the Message you want to broadcast:could i join?
Broadcast Message : could i join?
Enter the Message you want to broadcast:

```