Data Communication and Computer Network Laboratory

ASSIGNMENT

MCA 1st year 2nd Sem

Name	Roll-no
Dwip Shekhar Mondal	002210503037

Assignment - III

Problem statement →

The objective of this laboratory exercise is to look at the details of the User Datagram Protocol (UDP). UDP is a transport layer protocol. It is used by many application protocols like DNS, DHCP, SNMP etc., where reliability is not a concern.
To do this exercise you need to install the Wireshark tool, which is widely used to capture and examine a packet trace. Wireshark can be downloaded from www.wireshark.org.

Step 1: Capture a Trace

(i) Launch Wireshark
(ii) From Capture→Options select Loopback interface
(iii) Start a capture with a filter of "ip.addr==127.0.0.1 and udp.port==xxxx", where xxxx is the port number used by the UDP server.
(iv) Run the UDP server program on a terminal.
(v) Run multiple instances of the UDP client program on separate terminals and send requests to the server.
(vi) Stop Wireshark capture

Step 2: Inspect the Trace

Select different packets in the trace and browse the expanded UDP header and record the following fields:

- Source Port: the port from which the udp segment is sent.
- Destination Port: the port to which the udp segment is sent.
- Length: the length of the UDP segment.

Step 1: Capture a Trace →

For this task, the client/server program of question 3 from assignment-1 is used, in which the UDP client program sends requests to the server with a student name to get the requested student data (name, address).
The server returns the student info if the student exists in the local student database (a csv file stored in the server) otherwise it sends a 'student does not exist message'.
Using wireshark the packets sent/received by the client/server will be observed.

Note: there will be two instances of clients, both will request the server for student info, one will request for a student name that exists in the database the other client will request for a student that does not exist in the database.

- ➔ Wireshark capture with Loopback interface as input and with filter of "ip.addr==127.0.0.1 and udp.port==9999" (Port 9999 is used by the UDP server)
- ➔ Two instances of the client connect to the server.

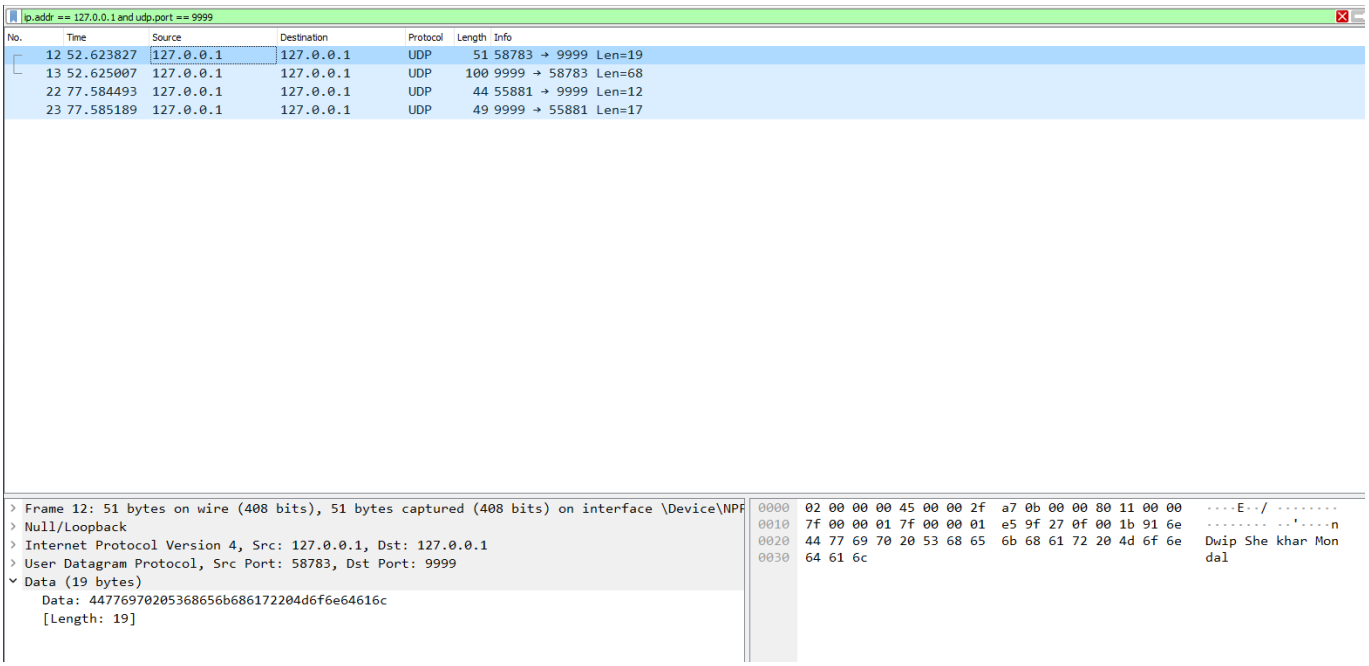

Figure 1: Capture of the packets from the client/server (UDP)

From the above capture it it's evident that →

- ➔ Two clients connects to the UDP server
- ➔ Server listens at port 9999
- ➔ 2 instances of client connects to the server
  - ◆ Client 1 connects from 58789 → 9999
  - ◆ Client 2 connects from 55881 → 9999
- ➔ Client 1 sends a request to get a student info "Dwip Shekhar Mondal" that exists in the student database stored at the server, and gets response "Name: Dwip Shekhar Mondal, Address: Greenpark Duttapukur pin-743248"
- ➔ Client 2 sends a request to get a student info "Rohit Sharma" that does not exist in the student database stored in the server, and gets a generic error message "Student does not exist"

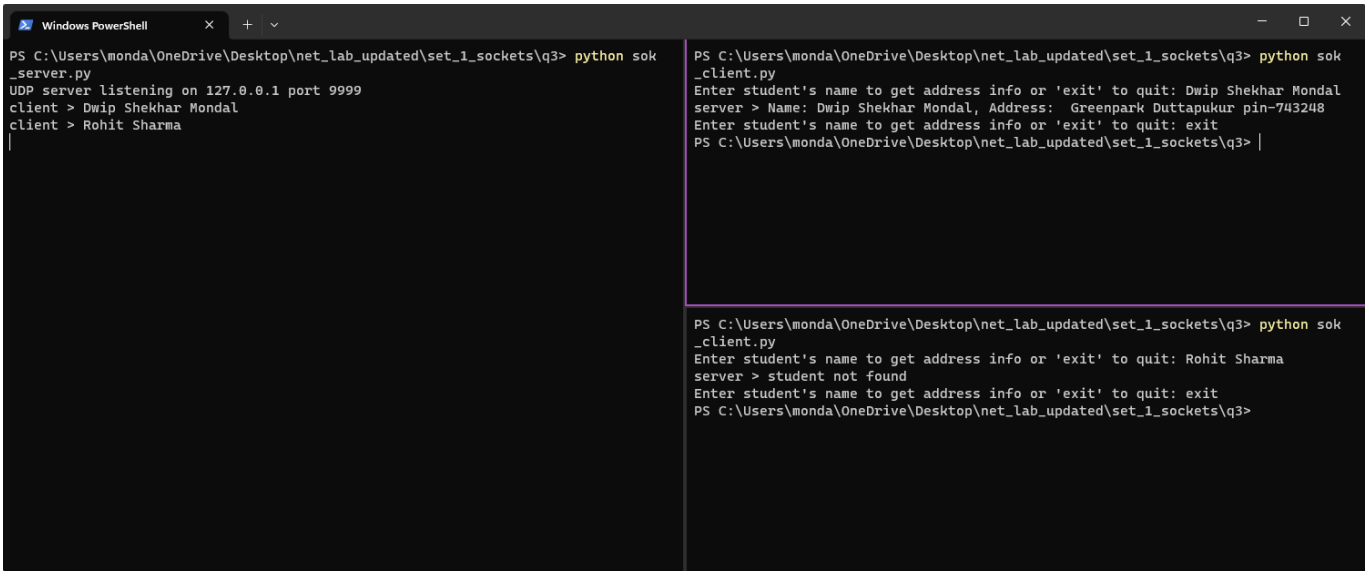

Figure 2: Request/response from client/server console (UDP)

Step 2: Inspect the Trace

Client 1 - server →

- ➔ Client 1 connects from port 58783 → 9999
  - ◆ It sends the request with source port: 58783 and destination port: 9999
  - ◆ It sends data with length 19 bytes i.e the UDP payload size is 19 bytes. Data = "Dwip Sekhar Mondal"
  - ◆ Server Responds with appropriate message.
    - Source port → 9999
    - Destination port → 58783
    - Payload size → 68 Bytes
    - Data = "Name: Dwip Shekhar Mondal, Address: Greenpark Duttapukur pin-743248"

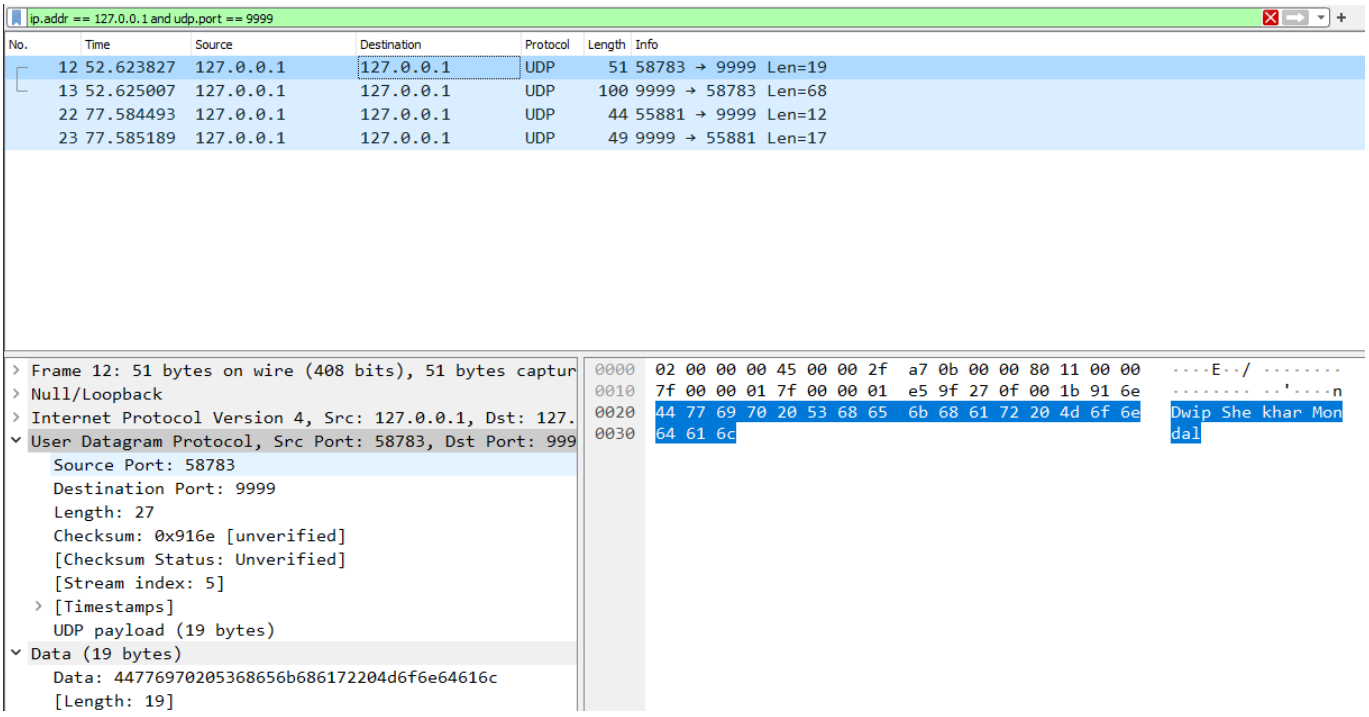

Figure 3: Request from client 1 to server (UDP)

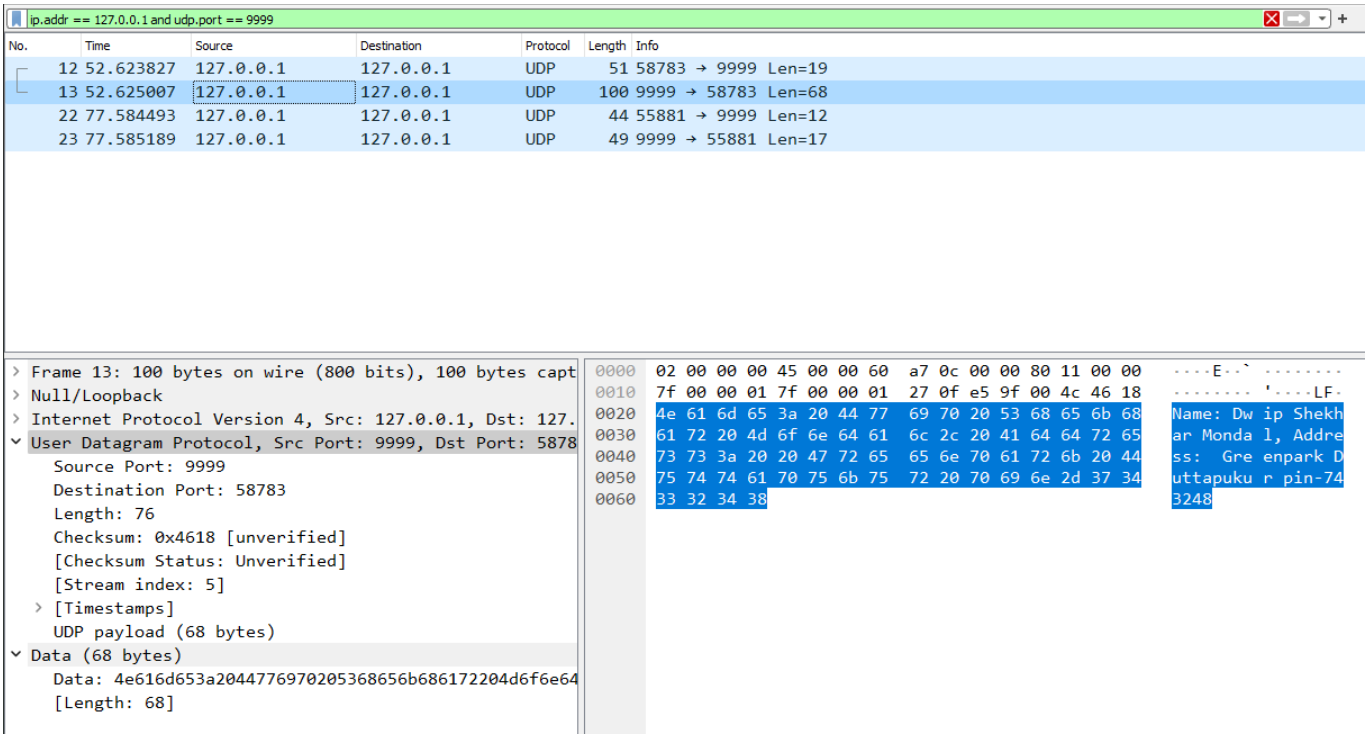

Figure 4: Response from server to client 1 (UDP)

Client 2 - server →

- ➔ Client 2 connects from port 55881 → 9999
  - ◆ It sends the request with source port: 55881 and destination port: 9999
  - ◆ It sends data with length 12 bytes i.e the UDP payload size is 12 bytes. Data = “Rohit Sharma”
  - ◆ Server Responds with appropriate message.
    - Source port → 9999
    - Destination port → 55881
    - Payload size → 17 Bytes
    - Data = “student not found”

ip.addr == 127.0.0.1 and udp.port == 9999

No.	Time	Source	Destination	Protocol	Length	Info
12	52.623827	127.0.0.1	127.0.0.1	UDP	51	58783 → 9999 Len=19
13	52.625007	127.0.0.1	127.0.0.1	UDP	100	9999 → 58783 Len=68
22	77.584493	127.0.0.1	127.0.0.1	UDP	44	55881 → 9999 Len=12
23	77.585189	127.0.0.1	127.0.0.1	UDP	49	9999 → 55881 Len=17

```
> Frame 22: 44 bytes on wire (352 bits), 44 bytes captur
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.
v User Datagram Protocol, Src Port: 55881, Dst Port: 999
    Source Port: 55881
    Destination Port: 9999
    Length: 20
    Checksum: 0xaf35 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 7]
  > [Timestamps]
    UDP payload (12 bytes)
v Data (12 bytes)
    Data: 526f68697420536861726d61
    [Length: 12]

0000  02 00 00 00 45 00 00 28  a7 0d 00 00 80 11 00 00   ····E··(  ********
0010  7f 00 00 01 7f 00 00 01  da 49 27 0f 00 14 af 35   ********  ·I'····5
0020  52 6f 68 69 74 20 53 68  61 72 6d 61               Rohit Sh  arma
```

Figure 5: Request from client 2 to server (UDP)

ip.addr == 127.0.0.1 and udp.port == 9999

No.	Time	Source	Destination	Protocol	Length	Info
12	52.623827	127.0.0.1	127.0.0.1	UDP	51	58783 → 9999 Len=19
13	52.625007	127.0.0.1	127.0.0.1	UDP	100	9999 → 58783 Len=68
22	77.584493	127.0.0.1	127.0.0.1	UDP	44	55881 → 9999 Len=12
23	77.585189	127.0.0.1	127.0.0.1	UDP	49	9999 → 55881 Len=17

```
> Frame 23: 49 bytes on wire (392 bits), 49 bytes captur
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.
v User Datagram Protocol, Src Port: 9999, Dst Port: 5588
    Source Port: 9999
    Destination Port: 55881
    Length: 25
    Checksum: 0x1b8b [unverified]
    [Checksum Status: Unverified]
    [Stream index: 7]
  > [Timestamps]
    UDP payload (17 bytes)
v Data (17 bytes)
    Data: 73747564656e74206e6f7420666f756e64
    [Length: 17]

0000  02 00 00 00 45 00 00 2d  a7 0e 00 00 80 11 00 00   ····E··-  ********
0010  7f 00 00 01 7f 00 00 01  27 0f da 49 00 19 1b 8b   ********  '··I····
0020  73 74 75 64 65 6e 74 20  6e 6f 74 20 66 6f 75 6e   student  not foun
0030  64                                                 d
```

Figure 6: Response from server to client 2 (UDP)