National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Sir Hassan Khaliq **Dated: 4/19/2023**

Semester: 6th **Section: C**

EE-357 Computer and Communication Networks

Experiment - 9

Wireshark – UDP (User datagram protocol)

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance** **5 Marks**	**Analysis of data in Lab Report** **5 Marks**	**Modern Tool Usage** **5 Marks**	**Ethics and Safety** **5 Marks**	**Individual and Team Work** **5 Marks**
Muhammad Ahmed Mohsin	**333060**					
Imran Haider	**332569**					
Amina Bashir	**343489**					

1 TABLE OF CONTENTS

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Wireshark – UDP (User datagram protocol)

2 OBJECTIVE OF THIS LAB:

In this lab, we'll explore several aspects of the User datagram protocol.

3 INSTRUCTIONS:

- Read carefully before starting the lab.
- These exercises are to be done individually.
- You are supposed to provide the answers to the in-line questions in this document and upload the completed document to your course's LMS site.
- **For all questions, you must not only answer the question, but also supply all necessary information regarding how you arrived at the answer (e.g., use screenshots/ accompanying text, etc.) Use red font color to distinguish your replies from the rest of the text.**
- Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.

4 BACKGROUND:

In this lab, we will take a quick look at the UDP transport protocol. UDP is a streamlined, no-frills protocol. You may want to read relevant text before doing this lab. Because UDP is simple, you will be able to cover it fairly quickly in this lab.

At this stage, you should be a Wireshark expert. Thus, the steps to acquire UDP packets will be not be provided as explicitly as in the earlier labs. In particular, no example screenshots will be given for all of the steps.

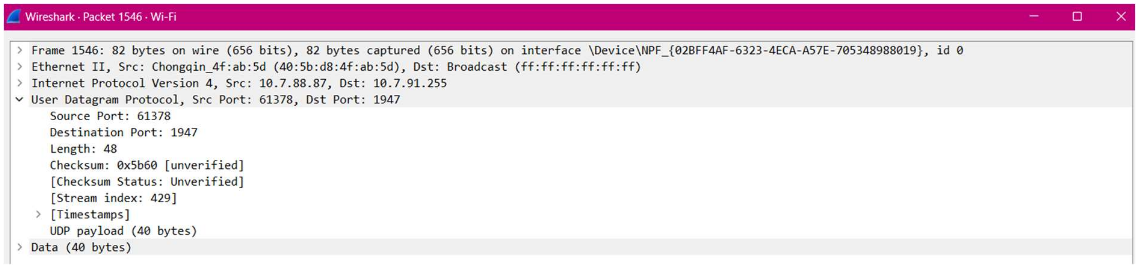
5 THE ASSIGNMENT

Start capturing packets in Wireshark and then do something that will cause your host to send and receive several UDP packets. It's also likely that just by doing nothing (except capturing packets via Wireshark) that some UDP packets sent by others will appear in your trace. In particular, the Simple Network Management Protocol (SNMP – see section 5.7 in the text) sends SNMP messages inside of UDP, so it's likely that you'll find some SNMP messages (and therefore UDP packets) in your trace.

After stopping packet capture, set your packet filter so that Wireshark only displays the UDP packets sent and received at your host. Pick one of these UDP packets and expand the UDP fields in the details window. If you are unable to find UDP packets or are unable to run Wireshark on a live network connection, you can download a packet trace containing some UDP packets.

Whenever possible, when answering a question below, you should provide any screenshots that are relevant to the work you are doing. Annotate the screenshots to explain your answer.

- **Select *one* UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldn't look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.**



```
> Frame 1546: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{02BFF4AF-6323-4ECA-A57E-705348988019}, id 0
> Ethernet II, Src: Chongqin_4f:ab:5d (40:5b:d8:4f:ab:5d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.7.88.87, Dst: 10.7.91.255
v User Datagram Protocol, Src Port: 61378, Dst Port: 1947
    Source Port: 61378
    Destination Port: 1947
    Length: 48
    Checksum: 0x5b60 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 429]
  > [Timestamps]
    UDP payload (40 bytes)
> Data (40 bytes)
```

UDP header contains 4 fields:

1. source port.

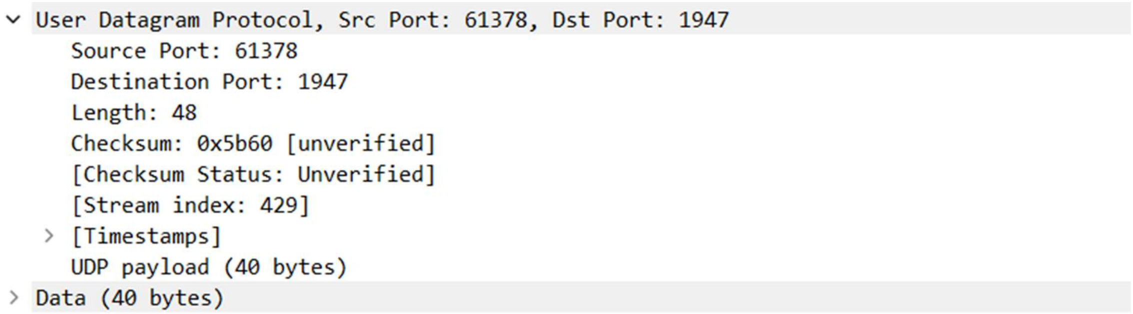
2. destination port.
3. length.
4. checksum

- **By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.**

The UDP header has a fixed length of 8 bytes. Each of these 4 header fields is 2 bytes long.

- **The value in the *Length* field is the length of what? Verify your claim with your captured UDP packet.**

The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The length of UDP payload for selected packet is 40 bytes. 48 bytes - 8 bytes = 40 bytes.



```
⌄ User Datagram Protocol, Src Port: 61378, Dst Port: 1947
    Source Port: 61378
    Destination Port: 1947
    Length: 48
    Checksum: 0x5b60 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 429]
  › [Timestamps]
    UDP payload (40 bytes)
› Data (40 bytes)
```

- **What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)**

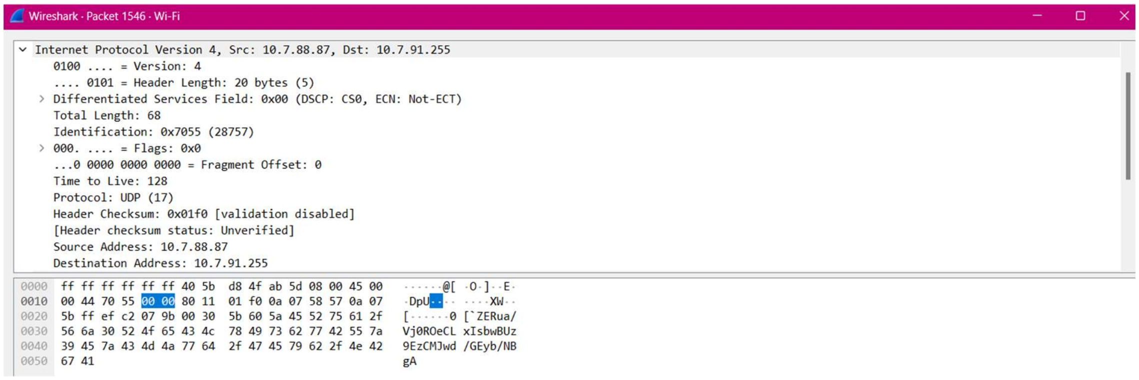
The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.

- **What is the largest possible source port number? (Hint: see the hint in 4.)**

The largest possible source port number is (2^16 – 1) = 65535

- **What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notations. To answer this question, you will need to look thoroughly in the headers.**

The IP protocol number for UDP is 0x11 hex, which is 17 in decimal value



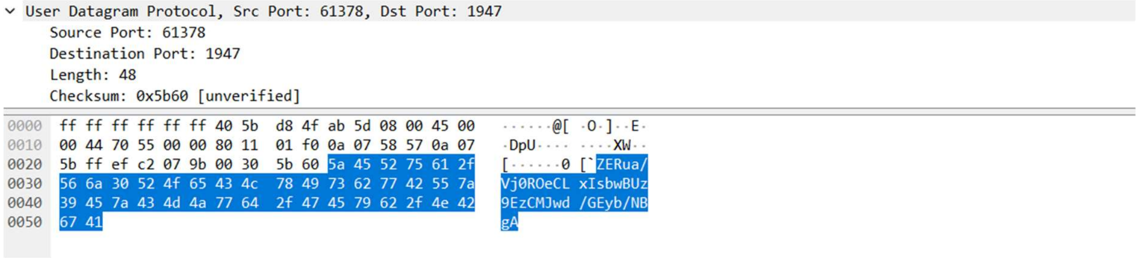
Wireshark · Packet 1546 · Wi-Fi

```
∨ Internet Protocol Version 4, Src: 10.7.88.87, Dst: 10.7.91.255
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 68
     Identification: 0x7055 (28757)
   > 000. .... = Flags: 0x0
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 128
     Protocol: UDP (17)
     Header Checksum: 0x01f0 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 10.7.88.87
     Destination Address: 10.7.91.255
```

```
0000  ff ff ff ff ff ff 40 5b  d8 4f ab 5d 08 00 45 00   ······@[ ·O·]··E·
0010  00 44 70 55 00 00 80 11  01 f0 0a 07 58 57 0a 07   ·DpU···· ····XW··
0020  5b ff ef c2 07 9b 00 30  5b 60 5a 45 52 75 61 2f   [······0 [`ZERua/
0030  56 6a 30 52 4f 65 43 4c  78 49 73 62 77 42 55 7a   Vj0ROeCL xIsbwBUz
0040  39 45 7a 43 4d 4a 77 64  2f 47 45 79 62 2f 4e 42   9EzCMJwd /GEyb/NB
0050  67 41                                              gA
```

- **Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.**

The source port of the UDP packet sent by the host is the same as the destination port of the reply packet, and conversely the destination port of the UDP packet sent by the host is the same as the source port of the reply packet.



```
˅ User Datagram Protocol, Src Port: 61378, Dst Port: 1947
     Source Port: 61378
     Destination Port: 1947
     Length: 48
     Checksum: 0x5b60 [unverified]

0000  ff ff ff ff ff ff 40 5b  d8 4f ab 5d 08 00 45 00   ······@[ ·O·]··E·
0010  00 44 70 55 00 00 80 11  01 f0 0a 07 58 57 0a 07   ·DpU···· ····XW··
0020  5b ff ef c2 07 9b 00 30  5b 60 5a 45 52 75 61 2f   [······0 [`ZERua/
0030  56 6a 30 52 4f 65 43 4c  78 49 73 62 77 42 55 7a   Vj0ROeCL xIsbwBUz
0040  39 45 7a 43 4d 4a 77 64  2f 47 45 79 62 2f 4e 42   9EzCMJwd /GEyb/NB
0050  67 41                                              gA
```

6 CONCLUSION

In conclusion, analysing the captured UDP packets using Wireshark revealed that UDP is suitable for applications that require low latency and high throughput. However, it lacks reliability and security features, making it unsuitable for applications that require reliable data transfer and data security. As a network administrator or software developer, it is essential to consider the requirements of the application when choosing the appropriate protocol. We answered questions regarding UDP and examined the UDP header and datagram protocols deeply.