THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Name: Shreya Chawale

Roll No: 22

Date: 08/03/24

Lab Assignment No.08

Aim: Installation of Wire shark and Analysis of Packet headers

Theory :

What is wireshark?

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems. Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

Wireshark installation in ubuntu/ Linux :

-sudo add-apt-repository ppa:wireshark-dev/stable

Update the repository:

-sudo apt-get update

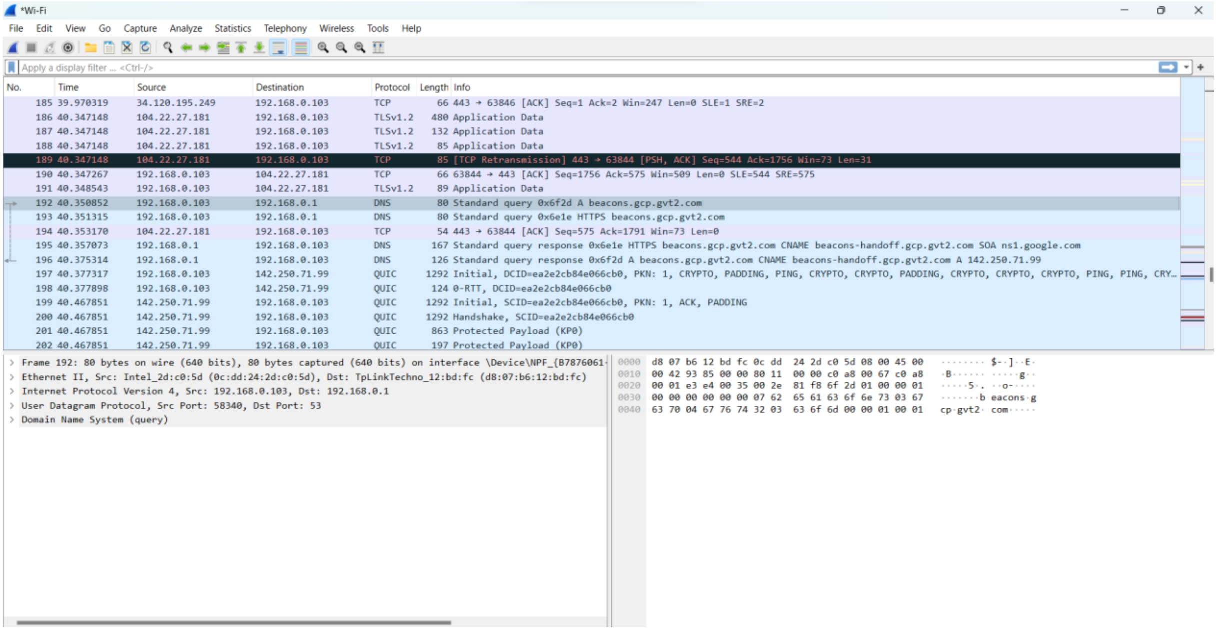
Install wire shark using the below command:

-sudo apt-get install wireshark

To run the wire shark use the below command

-sudo wireshark

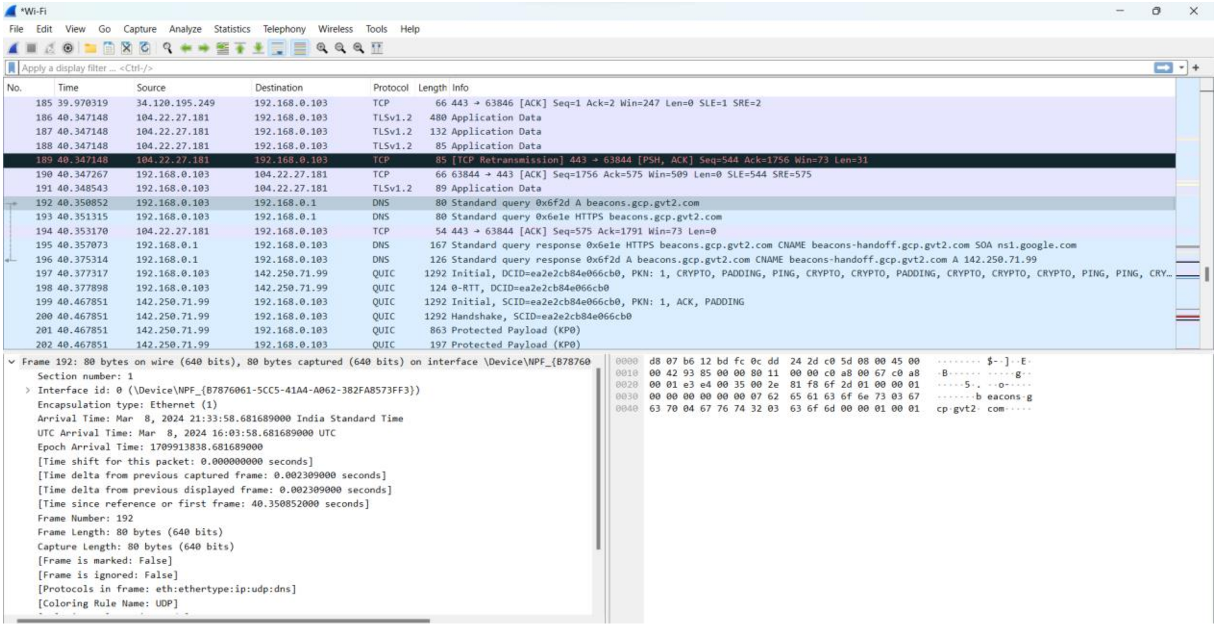
Screenshots :


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time       Source          Destination     Protocol Length Info
 185 39.970319  34.120.195.249  192.168.0.103   TCP        66 443 → 63846 [ACK] Seq=1 Ack=2 Win=247 Len=0 SLE=1 SRE=2
 186 40.347148  104.22.27.181   192.168.0.103   TLSv1.2   480 Application Data
 187 40.347148  104.22.27.181   192.168.0.103   TLSv1.2   132 Application Data
 188 40.347148  104.22.27.181   192.168.0.103   TLSv1.2    85 Application Data
 189 40.347148  104.22.27.181   192.168.0.103   TCP        85 [TCP Retransmission] 443 → 63844 [PSH, ACK] Seq=544 Ack=1756 Win=73 Len=31
 190 40.347267  192.168.0.103   104.22.27.181   TCP        66 63844 → 443 [ACK] Seq=1756 Ack=575 Win=509 Len=0 SLE=544 SRE=575
 191 40.348543  192.168.0.103   104.22.27.181   TLSv1.2    89 Application Data
 192 40.350852  192.168.0.103   192.168.0.1     DNS        80 Standard query 0x6f2d A beacons.gcp.gvt2.com
 193 40.351315  192.168.0.103   192.168.0.1     DNS        80 Standard query 0x6e1e HTTPS beacons.gcp.gvt2.com
 194 40.353170  104.22.27.181   192.168.0.103   TCP        54 443 → 63844 [ACK] Seq=575 Ack=1791 Win=73 Len=0
 195 40.357073  192.168.0.1     192.168.0.103   DNS       167 Standard query response 0x6e1e HTTPS beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com SOA ns1.google.com
 196 40.375314  192.168.0.1     192.168.0.103   DNS       126 Standard query response 0x6f2d A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com A 142.250.71.99
 197 40.377317  192.168.0.103   142.250.71.99   QUIC     1292 Initial, DCID=ea2e2cb84e066cb0, PKN: 1, CRYPTO, PADDING, PING, CRYPTO, CRYPTO, PADDING, CRYPTO, CRYPTO, CRYPTO, PING, PING, CRY…
 198 40.377898  192.168.0.103   142.250.71.99   QUIC      124 0-RTT, DCID=ea2e2cb84e066cb0
 199 40.467851  142.250.71.99   192.168.0.103   QUIC     1292 Initial, SCID=ea2e2cb84e066cb0, PKN: 1, ACK, PADDING
 200 40.467851  142.250.71.99   192.168.0.103   QUIC     1292 Handshake, SCID=ea2e2cb84e066cb0
 201 40.467851  142.250.71.99   192.168.0.103   QUIC      863 Protected Payload (KP0)
 202 40.467851  142.250.71.99   192.168.0.103   QUIC      197 Protected Payload (KP0)
```

```
> Frame 192: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface \Device\NPF_{B7876061-
> Ethernet II, Src: Intel_2d:c0:5d (0c:dd:24:2d:c0:5d), Dst: TpLinkTechno_12:bd:fc (d8:07:b6:12:bd:fc)
> Internet Protocol Version 4, Src: 192.168.0.103, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 58340, Dst Port: 53
> Domain Name System (query)
```

```
0000  d8 07 b6 12 bd fc 0c dd  24 2d c0 5d 08 00 45 00   ········ $-·]··E·
0010  00 42 93 85 00 00 80 11  00 00 c0 a8 00 67 c0 a8   ·B······ ·····g··
0020  00 01 e3 e4 00 35 00 2e  81 f8 6f 2d 01 00 00 01   ·····5·. ··o-····
0030  00 00 00 00 00 00 07 62  65 61 63 6f 6e 73 03 67   ·······b eacons·g
0040  63 70 04 67 76 74 32 03  63 6f 6d 00 00 01 00 01   cp·gvt2· com·····
```


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time       Source          Destination     Protocol Length Info
 185 39.970319  34.120.195.249  192.168.0.103   TCP        66 443 → 63846 [ACK] Seq=1 Ack=2 Win=247 Len=0 SLE=1 SRE=2
 186 40.347148  104.22.27.181   192.168.0.103   TLSv1.2   480 Application Data
 187 40.347148  104.22.27.181   192.168.0.103   TLSv1.2   132 Application Data
 188 40.347148  104.22.27.181   192.168.0.103   TLSv1.2    85 Application Data
 189 40.347148  104.22.27.181   192.168.0.103   TCP        85 [TCP Retransmission] 443 → 63844 [PSH, ACK] Seq=544 Ack=1756 Win=73 Len=31
 190 40.347267  192.168.0.103   104.22.27.181   TCP        66 63844 → 443 [ACK] Seq=1756 Ack=575 Win=509 Len=0 SLE=544 SRE=575
 191 40.348543  192.168.0.103   104.22.27.181   TLSv1.2    89 Application Data
 192 40.350852  192.168.0.103   192.168.0.1     DNS        80 Standard query 0x6f2d A beacons.gcp.gvt2.com
 193 40.351315  192.168.0.103   192.168.0.1     DNS        80 Standard query 0x6e1e HTTPS beacons.gcp.gvt2.com
 194 40.353170  104.22.27.181   192.168.0.103   TCP        54 443 → 63844 [ACK] Seq=575 Ack=1791 Win=73 Len=0
 195 40.357073  192.168.0.1     192.168.0.103   DNS       167 Standard query response 0x6e1e HTTPS beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com SOA ns1.google.com
 196 40.375314  192.168.0.1     192.168.0.103   DNS       126 Standard query response 0x6f2d A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com A 142.250.71.99
 197 40.377317  192.168.0.103   142.250.71.99   QUIC     1292 Initial, DCID=ea2e2cb84e066cb0, PKN: 1, CRYPTO, PADDING, PING, CRYPTO, CRYPTO, PADDING, CRYPTO, CRYPTO, CRYPTO, PING, PING, CRY…
 198 40.377898  192.168.0.103   142.250.71.99   QUIC      124 0-RTT, DCID=ea2e2cb84e066cb0
 199 40.467851  142.250.71.99   192.168.0.103   QUIC     1292 Initial, SCID=ea2e2cb84e066cb0, PKN: 1, ACK, PADDING
 200 40.467851  142.250.71.99   192.168.0.103   QUIC     1292 Handshake, SCID=ea2e2cb84e066cb0
 201 40.467851  142.250.71.99   192.168.0.103   QUIC      863 Protected Payload (KP0)
 202 40.467851  142.250.71.99   192.168.0.103   QUIC      197 Protected Payload (KP0)
```

```
v Frame 192: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface \Device\NPF_{B78760
    Section number: 1
  > Interface id: 0 (\Device\NPF_{B7876061-5CC5-41A4-A062-382FA8573FF3})
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar  8, 2024 21:33:58.681689000 India Standard Time
    UTC Arrival Time: Mar  8, 2024 16:03:58.681689000 UTC
    Epoch Arrival Time: 1709913838.681689000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.002309000 seconds]
    [Time delta from previous displayed frame: 0.002309000 seconds]
    [Time since reference or first frame: 40.350852000 seconds]
    Frame Number: 192
    Frame Length: 80 bytes (640 bits)
    Capture Length: 80 bytes (640 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
```

```
0000  d8 07 b6 12 bd fc 0c dd  24 2d c0 5d 08 00 45 00   ········ $-·]··E·
0010  00 42 93 85 00 00 80 11  00 00 c0 a8 00 67 c0 a8   ·B······ ·····g··
0020  00 01 e3 e4 00 35 00 2e  81 f8 6f 2d 01 00 00 01   ·····5·. ··o-····
0030  00 00 00 00 00 00 07 62  65 61 63 6f 6e 73 03 67   ·······b eacons·g
0040  63 70 04 67 76 74 32 03  63 6f 6d 00 00 01 00 01   cp·gvt2· com·····
```

Report for LAB 3-2: TCP

Name: Shreya Chawale	**Student ID: 22**	**Date: 08/03/24**

Part I	
1	Socket addresses: - Source Socket Address: 10.0.2.15:41140 - Destination Socket Address: 185.125.190.48:80
2	Set flags: - No flags are set in this packet.
3	Sequence number and acknowledgement number: - Sequence Number: 0 - Acknowledgement Number: Not applicable (0 because no data was being acknowledged)
4	Window size: - Window Size: Not provided in this packet.

Part II	
1	Set flag in HTTP GET message:- The HTTP GET message flag is not specified in the provided details.
2	Number of bytes transmitted by the HTTP GET message: - The number of bytes transmitted by the HTTP GET message is not provided.
3	Acknowledgement frequency: - Acknowledgement frequency and corresponding rule are not specified.
4	Number of bytes transmitted by each packet: - The number of bytes transmitted by each packet is not provided. Relation to sequence and acknowledgement Number: - Relation to sequence and acknowledgement number: Not applicable without specific packet details.
5	Original window sizes: - Original window sizes are not provided.

	Are these numbers expected? - It's unclear if the window sizes are expected without further information. How window sizes change? - How window sizes change is not specified.
6	How the window size is used in flow control? - The purpose and usage of the window size in flow control are not explained.
7	Purpose of the HTTP OK message:

	Part III
1	Number of TCP segments exchanged for connection termination: - The number of TCP segments exchanged for connection termination is not specified.
1	Which end point started the connection termination phase? - The initiating endpoint for connection termination is not mentioned.
2	Flags sets in each of the segments used for connection termination: - Flags set during connection termination are not provided.

	Part IV	
1	a. Source port number: 41140	b. Destination port number: 80
	c. Sequence number: 0	d. Acknowledgment number: Not Applicable
	e. Header length: 74 bytes	f. Set flags: None
	g. Window size:Not provided	h. Urgent pointer:Not Provided
2	Are answer in the question number 1 verified by the information in the detail pane lane?	

	- Information provided in question 1 can be verified using the details from the packet.
3	Does any of the TCP packet headers carry options? Explain: - The presence of options in TCP packet headers is not specified in the provided details.
4	Size of a TCP packet with no option: - Size of a TCP packet with no options is 74 bytes. Size of a TCP packet with options: - Size of a TCP packet with options is not specified.
5	Is window size in any of the TCP packet zero? Explain: - Zero window size in TCP packets is not mentioned.

Report for Lab 3-1: UDP

Name: Shreya Chawale	**Student ID: 22**	**Date: 08/03/24**

1	a. Source port number: 58340	b. Destination port number: 53
	c. Total length of user diagram: 80 bytes	d. Length of data: 80 bytes
	e. Is the packet from client or server? IP (192.168.0.103) is a private IP, it's likely from a client.	f. Application layer protocol: Domain Name System(DNS)
	g. Is checksum calculated?Yes, UDP checksum is calculated.	
2	Are answer in number 1 are verified by the information in the detail pane lane?	
3	Source and destination IP addresses in the query message: Source: 192.168.0.103 Destination: 192.168.0.1 Source and destination IP addresses in the response) message: Source: 192.168.0.1 Destination: 192.168.0.103 Relation between IP addresses: The source IP of the query becomes the destination IP in the response, and vice versa.	
4	Source and destination port number in the query message: Source port: 58340 Destination port: 53 Source and destination port number in the response message: Source port: 53 Destination port: 58340 Relation between port numbers: Ports are switched between the query and response. Which port number is well-known? Port 53 is well-known for DNS	
5	The length of the first UDP packet: 80 bytes How many bytes of payload are carried by the first UDP packet? Since the length of the UDP packet is 80 bytes, and there's no additional encapsulation mentioned, the payload length would be 80 bytes.	

6	Number of bytes in the DNS message: Since the UDP payload is the DNS message, and it's 80 bytes long, the DNS message is also 80 bytes long. Does the count agree with the answer to question 5? Yes, both are 80 bytes.
7	Is the checksum calculated for the first UDP packet? Yes Value of the checksum: The value of the checksum is not provided in the information given. If needed, you would have to inspect the packet further to obtain this value.