COMPUTER COMMUNICATION NETWORKS

LAB EXPERIMENT 13

NAME: RAHIL SHARMA
PRN: 18070123062
BATCH: 2018-2022
DIVISION: G2; EA 3

AIM: Packet Capture using Wireshark.
Using Wireshark capture – Address Resolution Protocol (Network Layer), TCP Packet, UDP and Ethernet frames on Data Link layer. Check the values of each field and attach a screenshot output obtained.

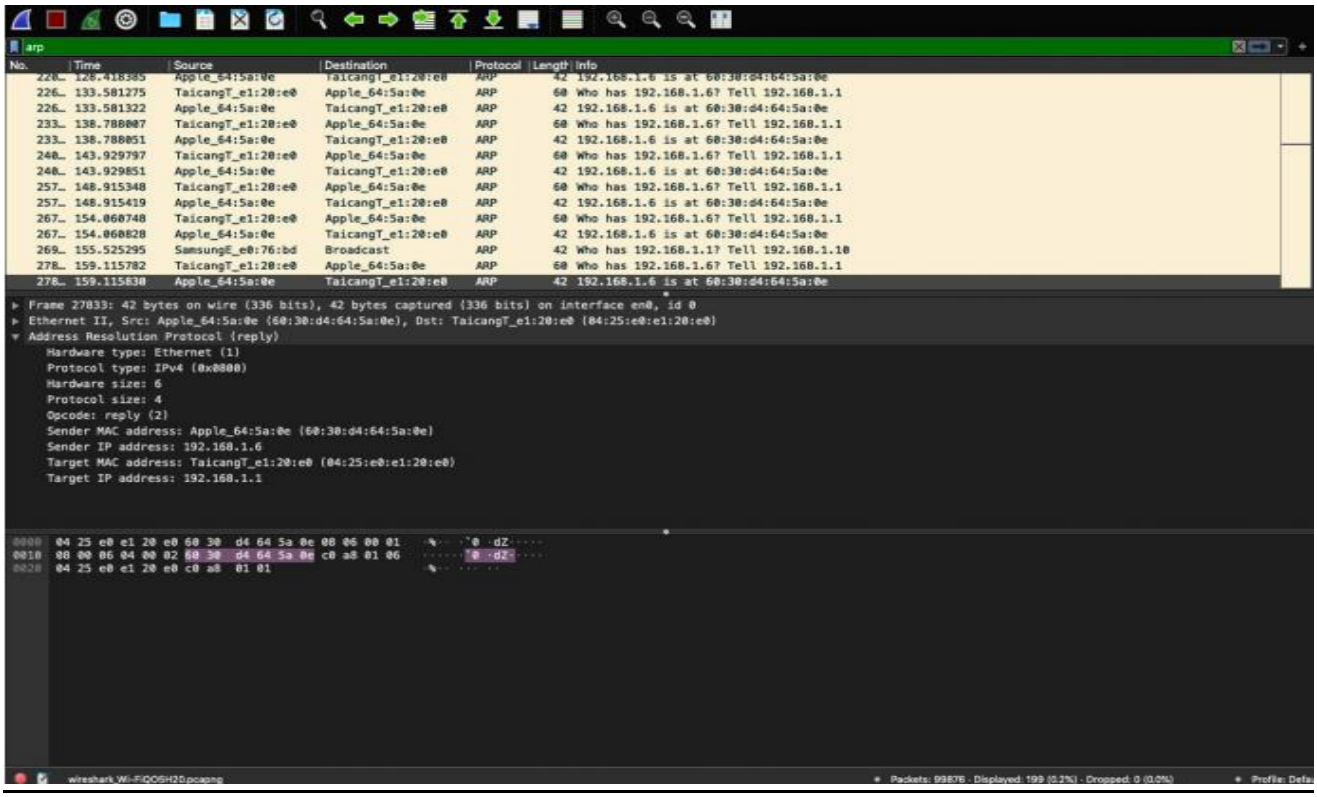
THEORY:

1. **HTTP:** The Hypertext Transfer Protocol (HTTP) is an application layer protocol for distributed, collaborative, hypermedia information systems. HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client.

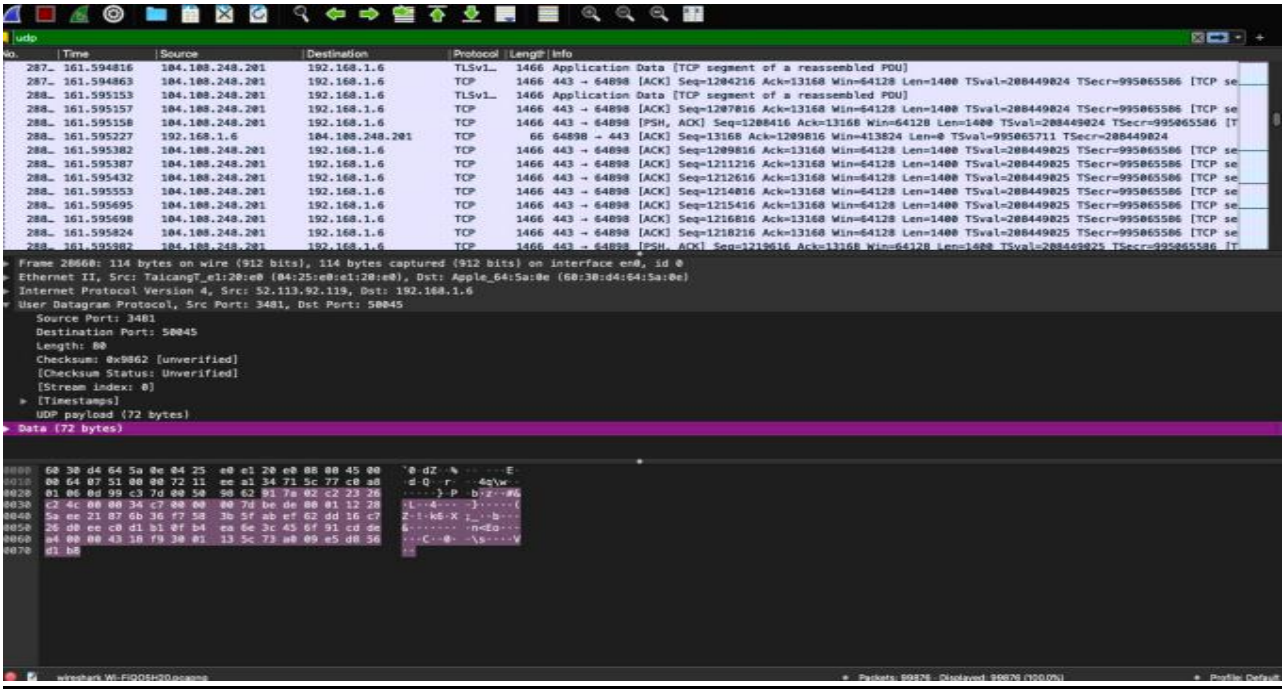
2. **TCP:** The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. TCP is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. Three-way handshake (active open), retransmission, and error-detection adds to reliability but lengthens latency.
3. **ARP:** The acronym ARP stands for Address Resolution Protocol which is one of the most important protocols of the Network layer in the OSI model. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. The devices of the network peel the header of the data link layer from the protocol data unit (PDU) called frame and transfers the packet to the network layer (layer 3 of OSI) where the network ID of the packet is validated with the destination IP's network ID of the packet and if it's equal then it responds to the source with the MAC address of the destination, else the packet reaches the gateway of the network and broadcasts packet to the devices it is connected with and validates their network ID
4. **UDP:** The UDP stands for user datagram protocol. this layer provides datagram based connectionless transport layer (layer 4) functionality in the Internet protocol family. UDP is only a thin layer, and provides not much more than the described UDP port multiplexing

Outputs:

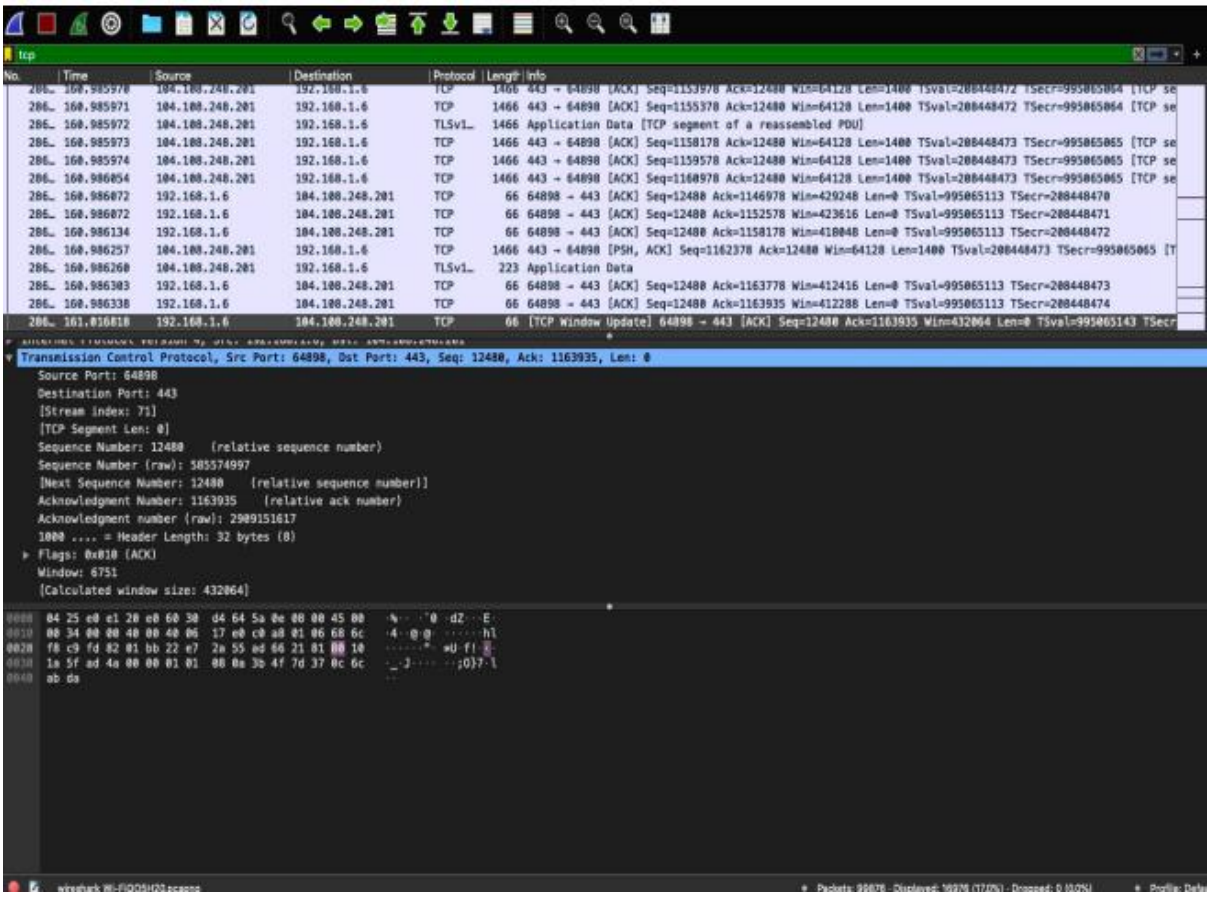
- **ARP:**



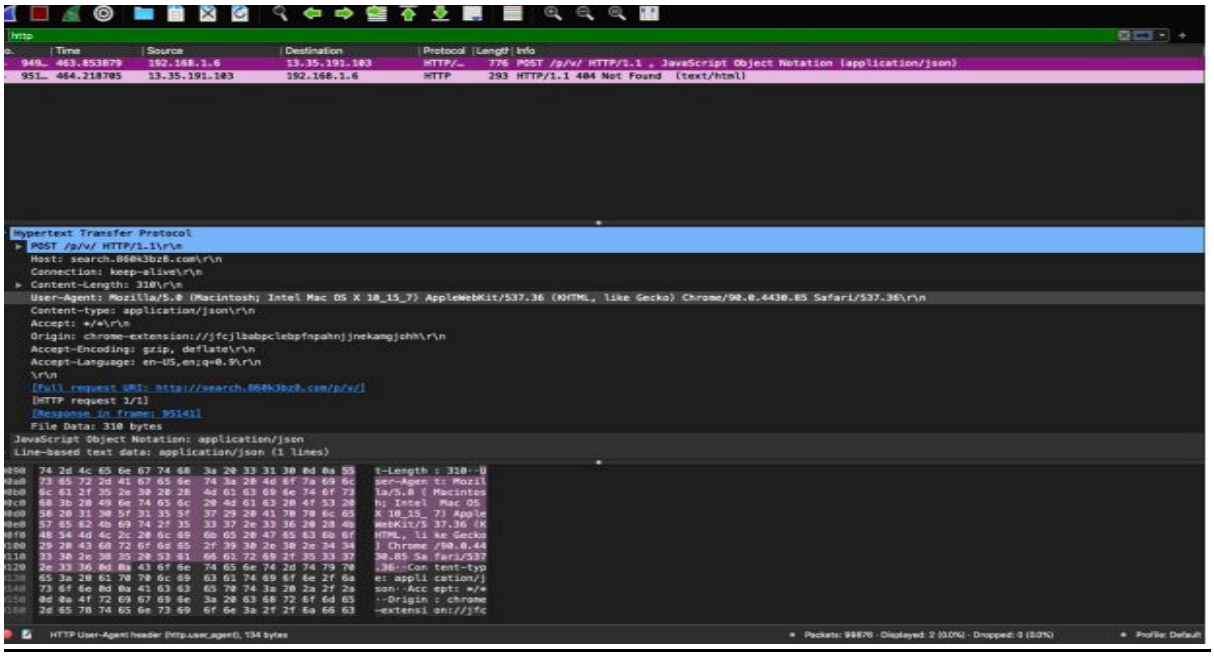
- **UDP:**



- **TCP:**


tcp
No. | Time | Source | Destination | Protocol | Length | Info
286… 160.985971 104.108.248.201 192.168.1.6 TCP 1466 443 → 64898 [ACK] Seq=1155370 Ack=12480 Win=64128 Len=1400 TSval=208448472 TSecr=995065064 [TCP se
286… 160.985972 104.108.248.201 192.168.1.6 TLSv1… 1466 Application Data [TCP segment of a reassembled PDU]
286… 160.985973 104.108.248.201 192.168.1.6 TCP 1466 443 → 64898 [ACK] Seq=1158178 Ack=12480 Win=64128 Len=1400 TSval=208448473 TSecr=995065065 [TCP se
286… 160.985974 104.108.248.201 192.168.1.6 TCP 1466 443 → 64898 [ACK] Seq=1159578 Ack=12480 Win=64128 Len=1400 TSval=208448473 TSecr=995065065 [TCP se
286… 160.986054 104.108.248.201 192.168.1.6 TCP 1466 443 → 64898 [ACK] Seq=1160978 Ack=12480 Win=64128 Len=1400 TSval=208448473 TSecr=995065065 [TCP se
286… 160.986072 192.168.1.6 104.108.248.201 TCP 66 64898 → 443 [ACK] Seq=12480 Ack=1146978 Win=429248 Len=0 TSval=995065113 TSecr=208448470
286… 160.986072 192.168.1.6 104.108.248.201 TCP 66 64898 → 443 [ACK] Seq=12480 Ack=1152578 Win=423616 Len=0 TSval=995065113 TSecr=208448471
286… 160.986134 192.168.1.6 104.108.248.201 TCP 66 64898 → 443 [ACK] Seq=12480 Ack=1158178 Win=418048 Len=0 TSval=995065113 TSecr=208448472
286… 160.986257 104.108.248.201 192.168.1.6 TCP 1466 443 → 64898 [PSH, ACK] Seq=1162378 Ack=12480 Win=64128 Len=1400 TSval=208448473 TSecr=995065065 [T
286… 160.986260 104.108.248.201 192.168.1.6 TLSv1… 223 Application Data
286… 160.986303 192.168.1.6 104.108.248.201 TCP 66 64898 → 443 [ACK] Seq=12480 Ack=1163778 Win=412416 Len=0 TSval=995065113 TSecr=208448473
286… 160.986338 192.168.1.6 104.108.248.201 TCP 66 64898 → 443 [ACK] Seq=12480 Ack=1163935 Win=412288 Len=0 TSval=995065113 TSecr=208448474
286… 161.016818 192.168.1.6 104.108.248.201 TCP 66 [TCP Window Update] 64898 → 443 [ACK] Seq=12480 Ack=1163935 Win=432064 Len=0 TSval=995065143 TSecr
Transmission Control Protocol, Src Port: 64898, Dst Port: 443, Seq: 12480, Ack: 1163935, Len: 0
Source Port: 64898
Destination Port: 443
[Stream index: 71]
[TCP Segment Len: 0]
Sequence Number: 12480 (relative sequence number)
Sequence Number (raw): 585574997
[Next Sequence Number: 12480 (relative sequence number)]
Acknowledgment Number: 1163935 (relative ack number)
Acknowledgment number (raw): 2909151617
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window: 6751
[Calculated window size: 432064]
0000 84 25 e0 e1 20 e0 60 30 d4 64 5a 0e 08 00 45 00
0010 00 34 00 00 40 00 40 06 17 e0 c0 a8 01 06 68 6c
0020 f8 c9 fd 82 01 bb 22 e7 2e 55 ad 66 21 01 80 10
0030 1a 5f ad 4a 00 00 01 01 08 0a 3b 4f 7d 37 0c 6c
0040 ab da

- **HTTP:**


http
No. | Time | Source | Destination | Protocol | Length | Info
948… 463.853879 192.168.1.6 13.35.191.103 HTTP/… 776 POST /p/v/ HTTP/1.1 , JavaScript Object Notation (application/json)
951… 464.218705 13.35.191.103 192.168.1.6 HTTP 293 HTTP/1.1 404 Not Found (text/html)
Hypertext Transfer Protocol
POST /p/v/ HTTP/1.1\r\n
Host: search.860k3bz8.com\r\n
Connection: keep-alive\r\n
Content-Length: 310\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36\r\n
Content-type: application/json\r\n
Accept: */*\r\n
Origin: chrome-extension://jfcjlbabpclebpfnpahnjjnekamgjohh\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://search.860k3bz8.com/p/v/]
[HTTP request 1/1]
[Response in frame: 95141]
File Data: 310 bytes
JavaScript Object Notation: application/json
Line-based text data: application/json (1 lines)
0090 74 2d 4c 65 6e 67 74 68 3a 20 33 31 30 0d 0a 55 t-Length : 310··U
00a0 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c ser-Agen t: Mozil
00b0 6c 61 2f 35 2e 30 20 28 4d 61 63 69 6e 74 6f 73 la/5.0 (Macintos
00c0 68 3b 20 49 6e 74 65 6c 20 4d 61 63 20 4f 53 20 h; Intel Mac OS
00d0 58 20 31 30 5f 31 35 5f 37 29 20 41 70 70 6c 65 X 10_15_ 7) Apple
00e0 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b WebKit/5 37.36 (K
00f0 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b 6f HTML, li ke Gecko
0100 29 20 43 68 72 6f 6d 65 2f 39 30 2e 30 2e 34 34) Chrome /90.0.44
0110 33 30 2e 38 35 20 53 61 66 61 72 69 2f 35 33 37 30.85 Sa fari/537
0120 2e 33 36 0d 0a 43 6f 6e 74 65 6e 74 2d 74 79 70 .36··Con tent-typ
0130 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 6a e: appli cation/j
0140 73 6f 6e 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a son··Acc ept: */*
0150 0d 0a 4f 72 69 67 69 6e 3a 20 63 68 72 6f 6d 65 ··Origin : chrome
0160 2d 65 78 74 65 6e 73 69 6f 6e 3a 2f 2f 6a 66 63 -extensi on://jfc
HTTP User-Agent header (http.user_agent), 134 bytes
Packets: 99876 · Displayed: 2 (0.0%) · Dropped: 0 (0.0%)
Profile: Default

CONCLUSION: From this experiment we have studied how to use the software Wireshark and use it to monitor, implement and check our UDP, TCP and HTTP and ARP.