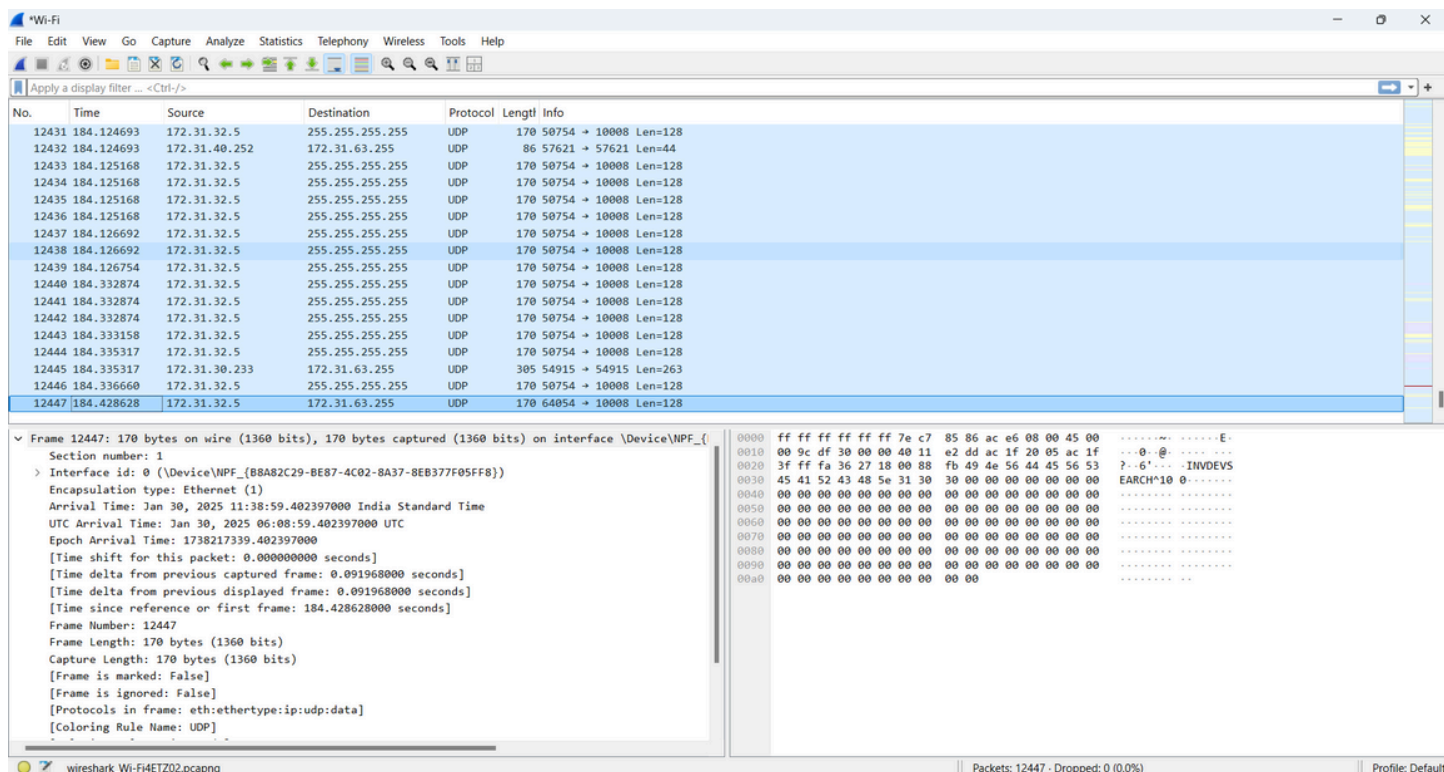


# Cyber Security Lab 3

Name : Anubhav Tandon

Roll Number : B22CS013

1. When I started packet capture in Wireshark on my wireless interface, I observed a constant stream of network packets flowing through my network adapter, which is shown in the screenshot below:



I analyzed the frame 12447, and my observations are as follows:

```
Frame 12447: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface \Device\NPF_{B8A82C29-BE87-4C02-8A37-8EB377F05FF8}, id 0
  Section number: 1
    > Interface id: 0 (\Device\NPF_{B8A82C29-BE87-4C02-8A37-8EB377F05FF8})
      Encapsulation type: Ethernet (1)
      Arrival Time: Jan 30, 2025 11:38:59.402397000 India Standard Time
      UTC Arrival Time: Jan 30, 2025 06:08:59.402397000 UTC
      Epoch Arrival Time: 1738217339.402397000
      [Time shift for this packet: 0.000000000 seconds]
      [Time delta from previous captured frame: 0.091968000 seconds]
      [Time delta from previous displayed frame: 0.091968000 seconds]
      [Time since reference or first frame: 184.428628000 seconds]
      Frame Number: 12447
      Frame Length: 170 bytes (1360 bits)
      Capture Length: 170 bytes (1360 bits)
      [Frame is marked: False]
      [Frame is ignored: False]
      [Protocols in frame: eth:ethertype:ip:udp:data]
      [Coloring Rule Name: UDP]
      [Coloring Rule String: udp]
    > Ethernet II, Src: 7e:c7:85:86:ac:e6 (7e:c7:85:86:ac:e6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Internet Protocol Version 4, Src: 172.31.32.5, Dst: 172.31.63.255
    > User Datagram Protocol, Src Port: 64054, Dst Port: 10008
    > Data (128 bytes)
```

- There's a surprising amount of background network traffic even when I'm not actively browsing or using the internet. This includes:
- Various broadcast messages
- DNS queries
- ARP requests and responses
- The packets are color-coded in Wireshark:
- Light purple packets for TCP
- Light blue for UDP
- Black for TCP packets with problems
- Each packet shows important information in the columns
- Time
- Source IP address
- Destination IP address
- Protocol (TCP, UDP, DNS, etc.)
- Length
- Info about what the packet contains
- The traffic is continuous and updates in real-time, showing how my computer is constantly communicating with other devices on the network and the internet.

## 2.

The image shows a Wireshark packet capture. The packet list on the left shows a DNS query (1262) and a response (1263). The packet details pane on the right shows the structure of the DNS query, including the question section for www.iitj.ac.in.

Packet 1262: 376 bytes on wire (3008 bits), 376 bytes captured (3008 bits) on interface \Device\NPF\_{B8A82C29-BE87-4C02-8A37-8EB377F05FF8}

Section number: 1

Interface id: 0 (\Device\NPF\_{B8A82C29-BE87-4C02-8A37-8EB377F05FF8})

Encapsulation type: Ethernet (1)

Arrival Time: Jan 30, 2025 12:20:39.624928000 India Standard Time

UTC Arrival Time: Jan 30, 2025 06:50:39.624928000 UTC

Epoch Arrival Time: 1738219839.624928000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.000000000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 8.441006000 seconds]

Frame Number: 1262

Frame Length: 376 bytes (3008 bits)

Capture Length: 376 bytes (3008 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:tls]

[Coloring Rule Name: Bad TCP]

0000 50 c2 e8 d3 4c 57 cc db 93 19 96 6f 08 00 45 80 P...LW...o...E...  
 0010 01 6a d7 2f 00 00 fb 06 30 3f 8e fa 4d ce ac 1f .j.-/...0?...M...  
 0020 2d b7 01 bb c5 eb 35 15 0c ed 42 60 2c 08 50 18 .....S...B'.P...  
 0030 03 8e 98 56 00 00 17 03 03 01 3d 05 77 61 84 a9 ...V.....m-wa...  
 0040 44 57 b0 09 9d c7 9f d3 08 4e e0 b3 bd ff db 49 Dm.....N.....I...  
 0050 a7 c1 c4 d7 47 bf af 5c 9c 1b 15 8d 5c a8 94 5f ....G...\.....V...  
 0060 3a 8d 8b f4 c6 cd 79 f1 a2 18 6d 1b bc df 99 26 .....y.....&...  
 0070 6c 3f 00 76 a2 3a d0 dc 53 42 e7 c5 0e aa 0e 92 1p-v...1u?...RCrF...  
 0080 aa 68 fb bf f6 31 cb 75 3f b8 e4 df 52 43 72 46 .h...:u?...AVut...  
 0090 62 da a7 41 46 6b 87 d3 20 de 41 56 75 74 0b a0 b-AFk...AVut...  
 00a0 cd 7f b4 3f 86 e0 af 28 65 96 7a 99 38 c6 0f 1e ...?...(e-z:8...  
 00b0 d9 c8 61 6d 5b 0f b3 59 99 a5 d9 07 9f 2c 03 60 ram[...Y...t...  
 00c0 57 7b fe a2 5a 2c 1b 83 d5 59 40 d1 e2 74 13 9b W(-Z...Y@...t...  
 00d0 a3 83 80 8c 1f 8c 40 39 7f 6f 3e b7 99 8a f5 64 .....@9...o>...d...  
 00e0 ff bb ad 91 bf f8 1d 6f b7 02 36 39 26 2e 63 0f .....o...69&.c...  
 00f0 75 70 88 ed a1 9c 3a df 52 8f 28 c1 3f a3 e1 b1 up.....R:(?...  
 0100 88 86 11 52 49 14 23 a0 98 e6 32 f0 4d ba a6 80 ...RI-#...-2-M...  
 0110 1a 7a e4 44 51 97 58 ed 59 ef b4 3c 0b 9a 64 52 .z-DQ-X...Y...<-d...  
 0120 03 fa 62 89 c7 d7 e0 bf d3 3c cd e3 25 f7 d7 2a ...b.....<-X...\*...  
 0130 1c 49 b3 ec 67 d7 cf 84 9f 54 81 a1 4f 3d e4 2f .I-g...-T...0a-/...  
 0140 0c 4a fd 02 07 f8 f7 e9 0a e7 58 05 de fe cc e6 .J.....X.....  
 0150 1a 2e d1 3d 82 d8 8b 95 cb f2 fd 6a 93 9a 89 c9 .u.....j.....  
 0160 f1 5e fa 01 14 1f 4b 44 0b b8 7a 6c 2d 44 21 7e .~.....KD...zl-DI~

- Yes, I could see the DNS request! A DNS query request was sent to the DNS server for www.iitj.ac.in.
- A DNS query packet sent to my DNS server
- The response containing the IP address for [www.iitj.ac.in](http://www.iitj.ac.in)
- The DNS protocol appears in light blue in Wireshark
- I noticed it used UDP port 53, which is standard for DNS

Encapsulation type: Ethernet (1)

Arrival Time: Jan 30, 2025 12:20:34.986328000 India Standard Time

UTC Arrival Time: Jan 30, 2025 06:50:34.986328000 UTC

Epoch Arrival Time: 1738219834.986328000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.000287000 seconds]

[Time delta from previous displayed frame: 0.000287000 seconds]

[Time since reference or first frame: 3.802406000 seconds]

Frame Number: 127

Frame Length: 74 bytes (592 bits)

Capture Length: 74 bytes (592 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:udp:dns]

[Coloring Rule Name: UDP]

[Coloring Rule String: udp]

&gt; Ethernet II, Src: CloudNetwork\_d3:4c:57 (50:c2:e8:d3:4c:57), Dst: IETF-VRRP-VRID\_0a (00:00:5e:00:00:0a)

&gt; Internet Protocol Version 4, Src: 172.31.45.183, Dst: 172.16.100.205

&gt; User Datagram Protocol, Src Port: 55319, Dst Port: 53

Domain Name System (query)

&gt; Transaction ID: 0xa091

&gt; Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

&gt; Queries

```

0000  00 00 5e 00 01 0a 50 c2 e8 d3 4c 57 08 00 45 00  --^...P...LW..E.
0010  00 3c f6 a4 00 00 80 11 59 58 ac 1f 2d b7 ac 10  -<.....YX.....
0020  64 cd d8 17 00 35 00 28 0d 24 a0 91 01 00 00 01  d....5.(.$. ....
0030  00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c  ....w ww..googl
0040  65 03 63 6f 6d 00 00 41 00 01                    e.com..A ..

```

5717	25.622693066	172.16.100.5	172.31.43.56	TLSv1.2	1514 Ignored Unknown Record
5718	25.622693578	172.16.100.5	172.31.43.56	TCP	30474 443 → 59994 [ACK] Seq=2970801 Ack=7648 Win=56192 Len=
5719	25.622841875	172.31.43.56	172.16.100.5	TCP	66 59994 → 443 [ACK] Seq=7648 Ack=3001209 Win=1962880 Le
5720	25.627041725	172.31.43.56	172.16.100.3	DNS	98 Standard query 0x42bc A googleads.g.doubleclick.net
5721	25.627303068	172.31.43.56	172.16.100.3	DNS	98 Standard query 0xc4dd HTTPS googleads.g.doubleclick.r
5722	25.638003101	172.31.43.56	172.16.100.3	DNS	93 Standard query 0xd847 A static.doubleclick.net OPT
5723	25.638249786	172.31.43.56	172.16.100.3	DNS	93 Standard query 0x3cbc HTTPS static.doubleclick.net OF
5724	25.646351158	172.16.100.5	172.31.43.56	TLSv1.2	14546 Application Data
5725	25.646388625	172.16.100.5	172.31.43.56	TLSv1.2	5858 Ignored Unknown Record
5726	25.646389033	172.16.100.5	172.31.43.56	TLSv1.2	11650 Ignored Unknown Record
5727	25.646442938	172.31.43.56	172.16.100.5	TCP	66 60008 → 443 [ACK] Seq=6141 Ack=1042068 Win=326656 Ler
5728	25.646491780	172.31.43.56	172.16.100.5	TCP	66 59984 → 443 [ACK] Seq=7531 Ack=581309 Win=435968 Len=
5729	25.646547371	172.16.100.5	172.31.43.56	TCP	4410 443 → 60008 [ACK] Seq=1042068 Ack=6141 Win=50432 Len=
5730	25.646547510	172.16.100.5	172.31.43.56	TLSv1.2	1259 [TCP Previous segment not captured] , Ignored Unknown
5731	25.646547614	172.16.100.5	172.31.43.56	TCP	7306 [TCP Out-Of-Order] 443 → 60008 [ACK] Seq=1046412 Ack=
5732	25.646584258	172.16.100.5	172.31.43.56	TCP	8754 443 → 59994 [ACK] Seq=3001209 Ack=7648 Win=56192 Len=
5733	25.646587704	172.31.43.56	172.16.100.5	TCP	78 60008 → 443 [ACK] Seq=6141 Ack=1046412 Win=322432 Ler
5734	25.646584380	172.16.100.5	172.31.43.56	TLSv1.2	65226 Encrypted Alert, Ignored Unknown Record

- I observed TCP 3-way handshake: SYN packet from my computer
- SYN-ACK from the server



- ACK from my computer

This established the TCP connection needed for HTTP

Yes, I could see multiple HTTP requests and responses:

No.	Time	Source	Destination	Protocol	Length	Info
1344	23.191219182	172.31.43.56	16.182.98.237	HTTP	342	GET /Entrust-0VTLs-I-R1.cer HTTP/1.1
1391	23.458981590	16.182.98.237	172.31.43.56	HTTP	248	HTTP/1.1 200 OK (application/pkix-cert)

The IP Address of the IITJ server is **172.16.100.5** As we can see in the screenshot below.

702	6.080102	172.31.45.183	142.250.206.162	TLSv1.2	128	Change Cipher Spec, Application Data
716	6.106543	172.31.45.183	142.250.194.234	TLSv1.2	128	Change Cipher Spec, Application Data
257	5.064312	172.31.45.183	172.16.100.5	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
261	5.064846	172.31.45.183	172.16.100.5	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
263	5.065763	172.31.45.183	172.16.100.5	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
633	5.916865	172.31.45.183	172.16.100.5	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
554	5.455641	172.31.45.183	172.16.100.115	TLSv1.3	670	Client Hello (SNI=aide.iitj.ac.in)
415	5.146203	172.31.45.183	151.101.66.137	TLSv1.3	542	Client Hello (SNI=code.jquery.com)
335	5.092669	172.31.45.183	142.250.193.10	TLSv1.2	832	Client Hello (SNI=fonts.googleapis.com)
683	5.995392	172.31.45.183	142.250.206.162	TLSv1.2	878	Client Hello (SNI=googleads.g.doubleclick.net)
251	5.061913	172.31.45.183	172.16.100.5	TLSv1.2	618	Client Hello (SNI=iitj.ac.in)
629	5.908999	172.31.45.183	172.16.100.5	TLSv1.2	522	Client Hello (SNI=iitj.ac.in)
693	6.017968	172.31.45.183	142.250.194.234	TLSv1.2	898	Client Hello (SNI=jnn-pa.googleapis.com)
158	4.321035	172.31.45.183	142.250.194.4	TLSv1.3	430	Client Hello (SNI=www.google.com)
339	5.095413	172.31.45.183	142.250.182.168	TLSv1.2	872	Client Hello (SNI=www.googletagmanager.com)
247	5.060722	172.31.45.183	172.16.100.5	TLSv1.2	526	Client Hello (SNI=www.iitj.ac.in)
249	5.061165	172.31.45.183	172.16.100.5	TLSv1.2	558	Client Hello (SNI=www.iitj.ac.in)

### 3.

No.	Time	Source	Destination	Protocol	Length	Info
1016	7.265958	142.250.77.206	172.31.45.183	TLSv1.2	737	[TCP Previous segment not captured], Application Data
1037	7.268881	142.250.77.206	172.31.45.183	TLSv1.2	1350	[TCP Previous segment not captured], Application Data
1041	7.269065	142.250.77.206	172.31.45.183	TLSv1.2	617	[TCP Previous segment not captured], Application Data
1262	8.441006	142.250.77.206	172.31.45.183	TLSv1.2	376	[TCP Previous segment not captured], Application Data
496	5.175943	142.250.193.10	172.31.45.183	TLSv1.2	111	[TCP Previous segment not captured], Ignored Unknown Record
507	5.183886	142.250.182.168	172.31.45.183	TLSv1.2	115	[TCP Previous segment not captured], Ignored Unknown Record
698	6.079242	142.250.206.162	172.31.45.183	TLSv1.2	118	[TCP Previous segment not captured], Ignored Unknown Record
712	6.105906	142.250.194.234	172.31.45.183	TLSv1.2	112	[TCP Previous segment not captured], Ignored Unknown Record
764	6.267528	142.250.194.234	172.31.45.183	TLSv1.2	265	[TCP Previous segment not captured], Ignored Unknown Record
1081	7.294793	142.250.77.206	172.31.45.183	TLSv1.2	439	[TCP Previous segment not captured], Ignored Unknown Record
1140	7.777513	142.250.77.206	172.31.45.183	TLSv1.2	99	[TCP Previous segment not captured], Ignored Unknown Record
1185	7.795959	142.250.77.206	172.31.45.183	TLSv1.2	778	[TCP Previous segment not captured], Ignored Unknown Record
163	4.402101	142.250.194.4	172.31.45.183	TCP	1196	[TCP Previous segment not captured] 443 → 50716 [PSH, ACK] Seq=4237 Ack=1789 Win=268032 Len=1142 [TCP PDU reassembled in 164]
490	5.173887	172.16.100.5	172.31.45.183	TCP	1363	[TCP Previous segment not captured] 443 → 50717 [PSH, ACK] Seq=299813 Ack=3151 Win=36096 Len=1309 [TCP PDU reassembled in 494], 4
270	5.071096	172.16.100.5	172.31.45.183	TCP	298	[TCP Previous segment not captured] 443 → 50717 [PSH, ACK] Seq=4518 Ack=3151 Win=36096 Len=244 [TCP PDU reassembled in 274]
327	5.091325	172.16.100.5	172.31.45.183	TCP	891	[TCP Previous segment not captured] 443 → 50717 [PSH, ACK] Seq=63144 Ack=3151 Win=36096 Len=837 [TCP PDU reassembled in 328]
362	5.128928	172.16.100.5	172.31.45.183	TCP	473	[TCP Previous segment not captured] 443 → 50717 [PSH, ACK] Seq=80041 Ack=3151 Win=36096 Len=419 [TCP PDU reassembled in 364]
370	5.129411	172.16.100.5	172.31.45.183	TCP	407	[TCP Previous segment not captured] 443 → 50717 [PSH, ACK] Seq=96520 Ack=3151 Win=36096 Len=353 [TCP PDU reassembled in 376]
782	6.280046	142.250.194.234	172.31.45.183	TCP	470	[TCP Previous segment not captured] 443 → 50726 [PSH, ACK] Seq=46808 Ack=3113 Win=267008 Len=416 [TCP PDU reassembled in 989]
1196	7.966257	142.250.77.206	172.31.45.183	TCP	93	[TCP Retransmission] 443 → 50667 [PSH, ACK] Seq=252483 Ack=21266 Win=911 Len=39
1265	8.441006	142.250.77.206	172.31.45.183	TCP	124	[TCP Retransmission] 443 → 50667 [PSH, ACK] Seq=340458 Ack=21672 Win=910 Len=70
1106	7.504361	142.250.194.46	172.31.45.183	TCP	85	[TCP Retransmission] 443 → 50680 [PSH, ACK] Seq=2953 Ack=5029 Win=1023 Len=31
1097	7.504216	142.250.194.234	172.31.45.183	TCP	93	[TCP Retransmission] 443 → 50726 [PSH, ACK] Seq=92778 Ack=5867 Win=264960 Len=39
390	5.135934	172.16.100.5	172.31.45.183	TCP	2974	[TCP Spurious Retransmission] 443 → 50717 [ACK] Seq=69821 Ack=3151 Win=36096 Len=2920 [TCP PDU reassembled in 364]

**Packets highlighted in black color signify TCP segments with problems or errors.** Based on the packet capture analysis, these highlighted packets indicate several types of TCP issues:

#### 1. "TCP Previous segment not captured":

- Seen in multiple packets from IPs 142.250.77.206 and 142.250.194.234 to 172.31.45.183
- Example lengths include 737, 1350, and 617 bytes
- These show missing segments in the TCP conversation

#### 2. "TCP Retransmission":

- Found in frames 1196, 1265, and 1106
- All with small packet sizes (93, 124, and 85 bytes, respectively)
- Shows packets that needed to be sent again

### 3. "TCP Spurious Retransmission":

- Seen in frame 398 (5.135934)
- From 172.16.100.5 to 172.31.45.183
- Length of 2974 bytes
- Indicates an unnecessary retransmission occurred

## 4. Here are 5 different filters and their observations:

### 1. **tcp.flags.syn == 1**

- Displays packets that initiate TCP connections (SYN packets)
- Helped visualize new connection establishments between my computer and servers

### 2. **ip.addr == 172.31.45.183**

- Shows all traffic involving my computer's IP address
- Useful for isolating my network communication from other traffic

### 3. **dns**

- Revealed DNS queries and responses for domain name resolution
- Allowed me to track which domain names were being looked up during browsing

### 4. **http.request**

- Filtered to show only HTTP request packets
- It made it easy to see what resources websites were requesting

### 5. **tcp.window\_size == 0**

- Shows packets where TCP flow control window is zero
- Helped identify potential network congestion points

## 5. The filter command to list all outgoing traffic would be:

**ip.src == 172.31.45.183**

This displays:

- All packets where my IP (172.31.45.183) is the source
- Only outbound traffic, filtering out incoming packets

6. Start a new packet capture to now visit an external website, say [www.cricinfo.com](http://www.cricinfo.com). Can you show the 3-way TCP handshake happening? Can you see your IITJ proxy in between? What is its IP address?

**IP address of cricinfo.com : 18.138.38.184 Source IP Address of my machine : 172.31.93.71**  
**IITJ Proxy : 172.16.100.206**

136	2.894904417	172.31.93.71	18.138.38.184	TCP	66 37554 → 443 [FIN, ACK] Seq=1805 Ack=4379 Win=60032 Len=0 TSval=1765194207 TSecr=36140
137	2.912981352	172.16.100.206	172.31.93.71	DNS	143 Standard query response 0xf75f HTTPS www.gstatic.com SOA ns1.google.com OPT
138	2.990316177	18.138.38.184	172.31.93.71	TCP	66 443 → 37554 [FIN, ACK] Seq=4379 Ack=1805 Win=47872 Len=0 TSval=3614079196 TSecr=17651
139	2.990382846	172.31.93.71	18.138.38.184	TCP	66 37554 → 443 [ACK] Seq=1806 Ack=4380 Win=60032 Len=0 TSval=1765194302 TSecr=3614079196

**7. DNS uses UDP** because it is a **lightweight, connectionless protocol that allows for quick, simple, and efficient communication**. DNS queries are usually small, and the protocol does not require the overhead of establishing a connection or ensuring reliability through retransmissions, as failed queries are typically retried.

On the other hand, **HTTP uses TCP because it requires reliable, ordered delivery of data**. TCP's connection-oriented nature ensures that large files, such as webpages and their associated resources, are transferred correctly, with error correction, flow control, and retransmission capabilities. This makes TCP ideal for the more complex, multi-step communication needed for web browsing, where data integrity is crucial.

## 8.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	60029 → 65432 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
2	0.000044	127.0.0.1	127.0.0.1	TCP	56	65432 → 60029 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3	0.000078	127.0.0.1	127.0.0.1	TCP	44	60029 → 65432 [ACK] Seq=1 Ack=1 Win=327424 Len=0
4	0.000128	127.0.0.1	127.0.0.1	TCP	62	60029 → 65432 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=18
5	0.000142	127.0.0.1	127.0.0.1	TCP	44	65432 → 60029 [ACK] Seq=1 Ack=19 Win=2161152 Len=0
6	0.000785	127.0.0.1	127.0.0.1	TCP	62	65432 → 60029 [PSH, ACK] Seq=1 Ack=19 Win=2161152 Len=18
7	0.000811	127.0.0.1	127.0.0.1	TCP	44	60029 → 65432 [ACK] Seq=19 Ack=19 Win=327424 Len=0
8	0.000835	127.0.0.1	127.0.0.1	TCP	44	65432 → 60029 [FIN, ACK] Seq=19 Ack=19 Win=2161152 Len=0
9	0.000844	127.0.0.1	127.0.0.1	TCP	44	60029 → 65432 [ACK] Seq=19 Ack=20 Win=327424 Len=0
10	0.001069	127.0.0.1	127.0.0.1	TCP	44	60029 → 65432 [FIN, ACK] Seq=19 Ack=20 Win=327424 Len=0
11	0.001113	127.0.0.1	127.0.0.1	TCP	44	65432 → 60029 [ACK] Seq=20 Ack=20 Win=2161152 Len=0

```
Server.py > ...
1  # server.py
2  import socket
3
4  # Set up server details
5  host = '127.0.0.1' # Localhost
6  port = 65432 # Port to listen on
7
8  # Create a TCP/IP socket
9  server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10
11 # Bind the socket to the port
12 server_socket.bind((host, port))
13
14 # Listen for incoming connections
15 server_socket.listen(1)
16 print(f"Server listening on {host}:{port}...")
17
18 # Accept a connection from a client
19 conn, addr = server_socket.accept()
20 print(f"Connected by {addr}")
21
22 # Receive and send data
23 data = conn.recv(1024)
24 print(f"Received from client: {data.decode()}")
25 conn.sendall(b"Hello from server!")
26
27 # Close the connection
28 conn.close()
29
```

```

# client.py
import socket

# Set up server details
host = '127.0.0.1' # Localhost
port = 65432 # Port to connect to

# Create a TCP/IP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connect to the server
client_socket.connect((host, port))

# Send data to the server
client_socket.sendall(b"Hello from client!")

# Receive data from the server
data = client_socket.recv(1024)
print(f"Received from server: {data.decode()}")

# Close the socket
client_socket.close()

```

- The communication is between a client and a server both running on the **localhost (127.0.0.1)**.
- The client and server are using different **ports (60029 and 65432)** for communication.
- The packets show a typical TCP data exchange where **data is sent and acknowledged between the client and server**.
- The use of **PSH flags** indicates that the data is being pushed to the receiving application immediately.
- The sequence and acknowledgment numbers are correctly incremented, showing a reliable data transfer process.

