

COMPUTER NETWORKS

NAME: KEERTHAN P.V

SRN:PES2UG23CS272

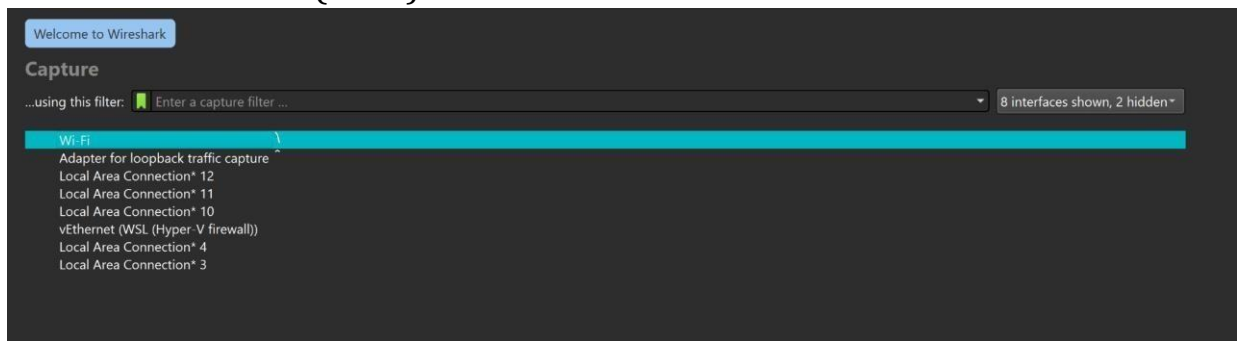
SEC : E

LAB 3

Objective

This lab introduces students to the UDP (User Datagram Protocol) and its role in DNS (Domain Name System) queries. Using Wireshark, students will capture and analyse DNS traffic, observe UDP behaviour, and understand how domain names are resolved into IP addresses.

OPEN WIRESHARK (WIFI) :



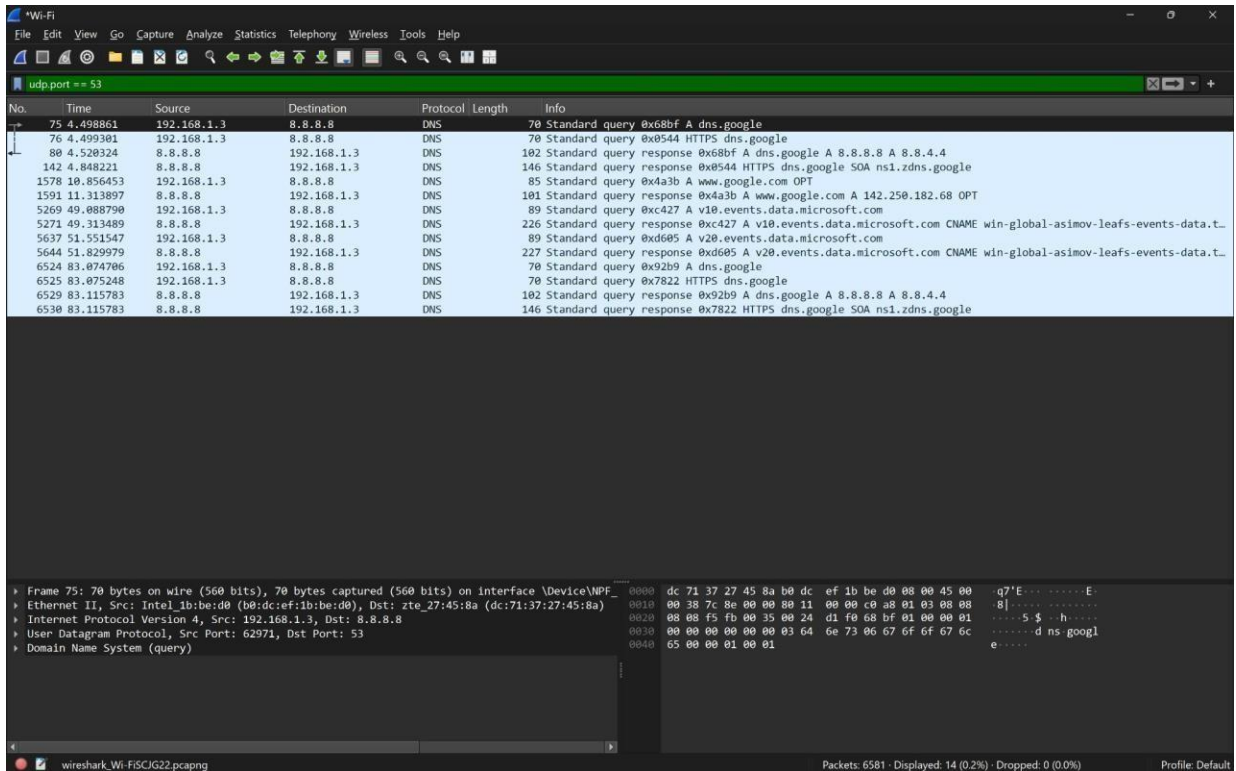
Start Capturing Packets

```
;; <<>> DiG 9.18.30-Ubuntu0.24.04.2-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4916
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

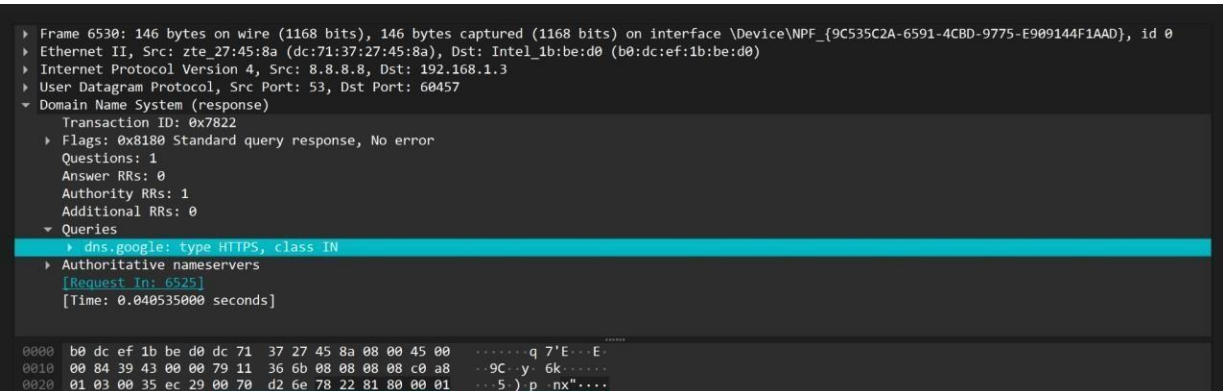
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;; www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.      261     IN      A      142.250.182.68

;; Query time: 465 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Sun Feb 16 17:21:12 UTC 2025
;; MSG SIZE rcvd: 59
```



Analyse DNS Packets



Compare UDP and TCP DNS Queries

```

> Frame 12: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface eth0, id 0
> Ethernet II, Src: VMWare_ed:82:37 (00:50:56:ed:82:37), Dst: VMWare_fc:c8:08 (00:0c:29:fc:c8:08)
  > Destination: VMWare_fc:c8:08 (00:0c:29:fc:c8:08)
  > Source: VMWare_ed:82:37 (00:50:56:ed:82:37)
  Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.219.129
> Transmission Control Protocol, Src Port: 53, Dst Port: 45677, Seq: 1, Ack: 54, Len: 57
  Source Port: 53
  Destination Port: 45677
  [Stream index: 0]
  > [Conversation completeness: Complete. WITH DATA (31)]
0000  00 0c 29 fc c8 08 00 50 56 ed 82 37 08 00 45 00  ...P V..7..E.
0010  00 61 fe fb 00 00 80 06 8f 61 08 08 08 08 c0 a8  ..a.....a.....
0020  db 81 00 35 b2 6d 7a 37 48 e3 6b ff ff bc 50 18  ...5mz7 H k...P.
0030  fa f0 b3 b8 00 00 00 37 a2 66 81 80 00 01 00 01  ....7..f.....
0040  00 00 00 01 06 67 6f 6f 67 6c 65 03 63 6f 6d 00  ....goo gle.com.
0050  00 01 00 01 c0 0c 00 01 00 01 00 00 00 7f 00 04  .....
0060  8e fa c3 ce 00 00 29 02 00 00 00 00 00 00 00 00  .....).

```

Homework

Run dig www.google.com twice and observe the difference.

```

; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41592
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                295     IN      A      142.250.195.132

;; Query time: 29 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Sun Feb 16 17:25:38 UTC 2025
;; MSG SIZE rcvd: 59

```

```

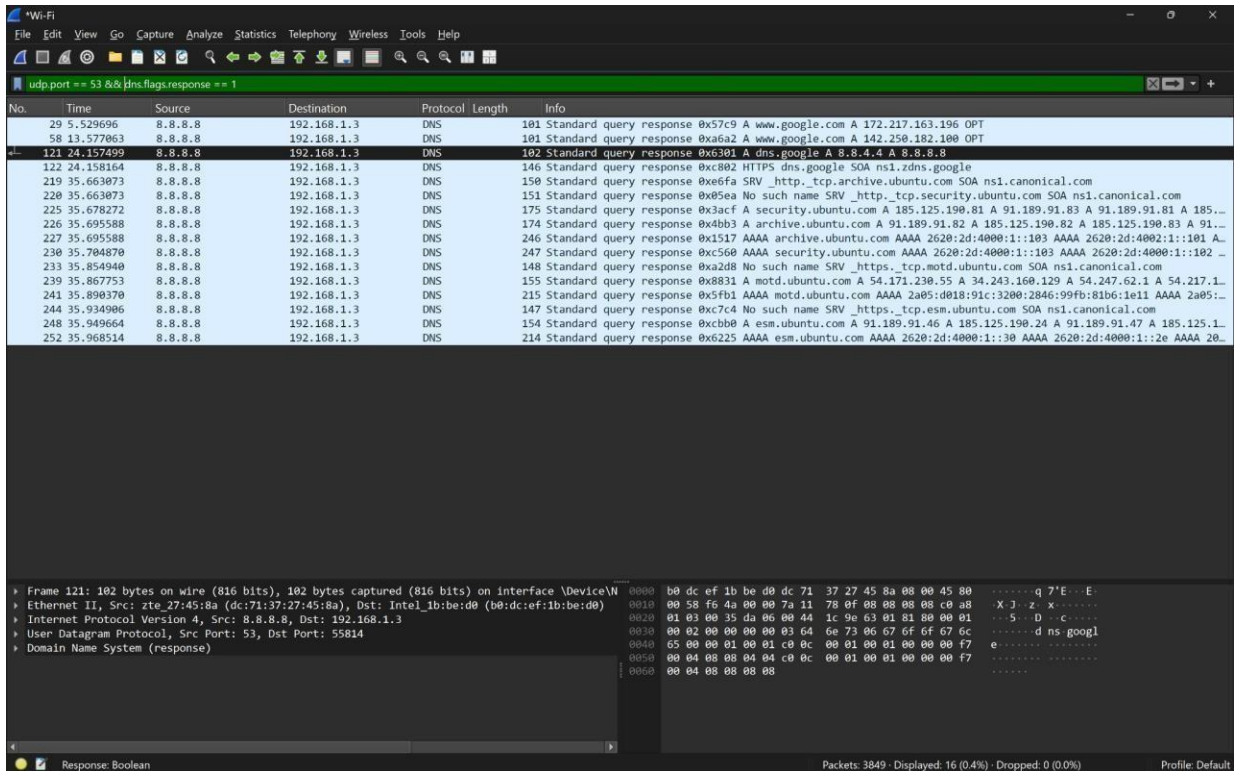
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4916
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                261     IN      A      142.250.182.68

;; Query time: 465 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Sun Feb 16 17:21:12 UTC 2025
;; MSG SIZE rcvd: 59

```



Dig www.google.com AAAA

```

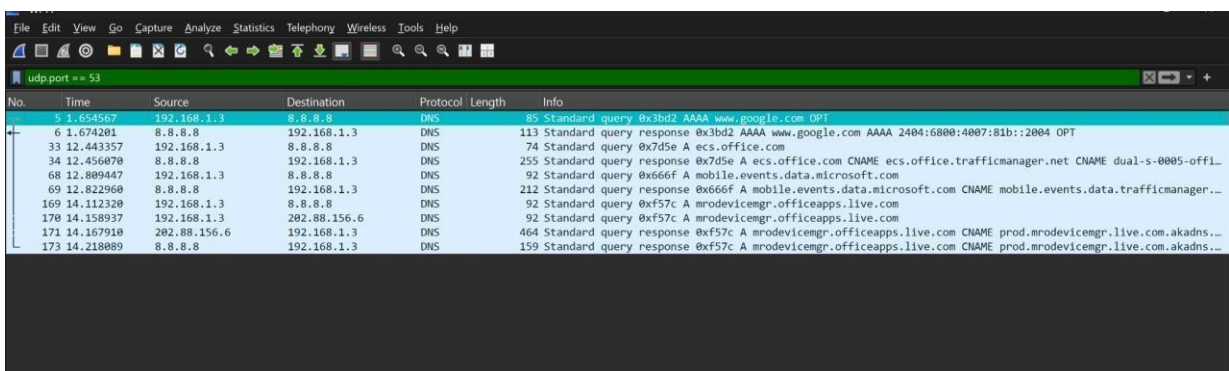
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> www.google.com AAAA
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 55496
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      AAAA

;; ANSWER SECTION:
www.google.com.                 99      IN      AAAA    2404:6800:4007:81f::2004

;; Query time: 9 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
;; WHEN: Sun Feb 16 17:28:24 UTC 2025
;; MSG SIZE rcvd: 71

```



dig @8.8.8.8 www.google.com

```
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> @8.8.8.8 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10253
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                19      IN      A      142.250.195.196

;; Query time: 9 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Sun Feb 16 17:30:16 UTC 2025
;; MSG SIZE rcvd: 59
```

dig @1.1.1.1 www.google.com

```
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> @1.1.1.1 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56176
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
) ;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                300     IN      A      142.250.196.36

;; Query time: 9 msec
;; SERVER: 1.1.1.1#53(1.1.1.1) (UDP)
;; WHEN: Sun Feb 16 17:30:32 UTC 2025
;; MSG SIZE rcvd: 59
```

Capture **DNS over TCP** using dig +tcp

```

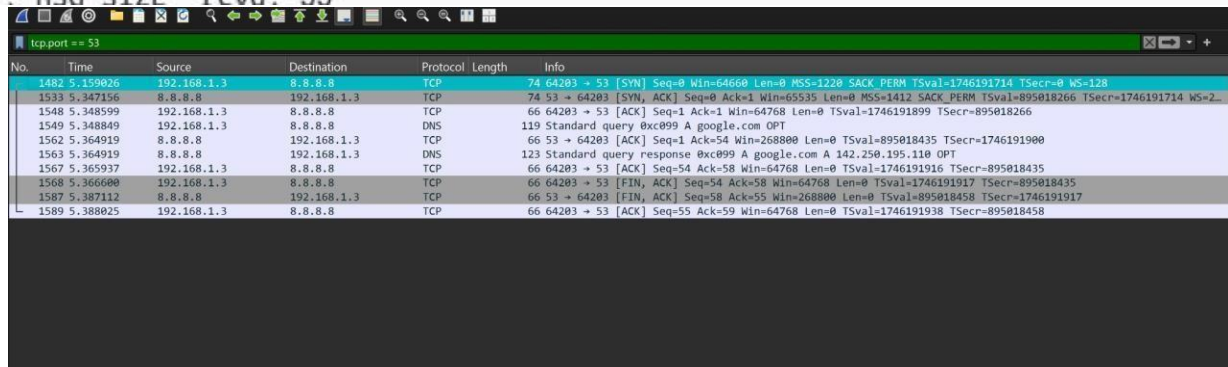
; <<>> DiG 9.18.30-0ubuntu0.24.04.2-Ubuntu <<>> +tcp google.com @8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49305
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                 285     IN      A      142.250.195.110

;; Query time: 9 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (TCP)
;; WHEN: Sun Feb 16 17:32:16 UTC 2025
; MSG SIZE rcvd: 55

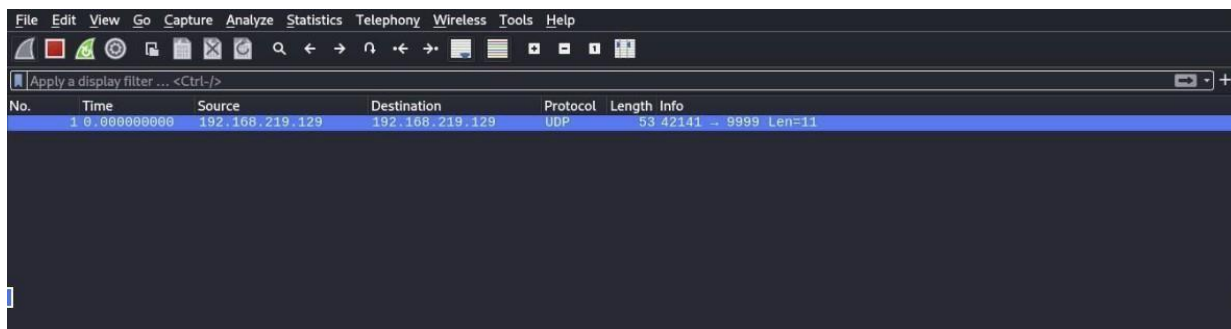
```



tcp.port == 53

No.	Time	Source	Destination	Protocol	Length	Info
1482	5.159026	192.168.1.3	8.8.8.8	TCP	74	64203 → 53 [SYN] Seq=0 Win=64660 Len=0 MSS=1220 SACK_PERM TSval=1746191714 TSecr=0 WS=128
1533	5.347156	8.8.8.8	192.168.1.3	TCP	74	53 → 64203 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=895018266 TSecr=1746191714 WS=2..
1548	5.348599	192.168.1.3	8.8.8.8	TCP	66	64203 → 53 [ACK] Seq=1 Ack=1 Win=64768 Len=0 TSval=1746191899 TSecr=895018266
1549	5.348849	192.168.1.3	8.8.8.8	DNS	119	Standard query 0xc099 A google.com OPT
1562	5.364919	8.8.8.8	192.168.1.3	TCP	66	53 → 64203 [ACK] Seq=1 Ack=54 Win=268800 Len=0 TSval=895018435 TSecr=1746191900
1563	5.364919	8.8.8.8	192.168.1.3	DNS	123	Standard query response 0xc099 A google.com A 142.250.195.110 OPT
1567	5.365937	192.168.1.3	8.8.8.8	TCP	66	64203 → 53 [ACK] Seq=54 Ack=58 Win=64768 Len=0 TSval=1746191916 TSecr=895018435
1568	5.366600	192.168.1.3	8.8.8.8	TCP	66	64203 → 53 [FIN, ACK] Seq=54 Ack=58 Win=64768 Len=0 TSval=1746191917 TSecr=895018435
1587	5.387112	8.8.8.8	192.168.1.3	TCP	66	53 → 64203 [FIN, ACK] Seq=58 Ack=55 Win=268800 Len=0 TSval=895018458 TSecr=1746191917
1589	5.388025	192.168.1.3	8.8.8.8	TCP	66	64203 → 53 [ACK] Seq=55 Ack=59 Win=64768 Len=0 TSval=1746191938 TSecr=895018458

UDP Communication



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.219.129	192.168.219.129	UDP	53	42141 → 9999 Len=11

TCP

Source	Destination	Protocol	Length	Info
192.168.219.129	192.168.219.129	UDP	53	38276 → 9999 Len=11
192.168.219.129	192.168.219.129	ICMP	81	Destination unreachable (Port unreachable)
192.168.219.129	192.168.219.129	TCP	74	48298 → 9999 [SYN, Seq=0 Win=33280 Len=0 MSS=65495 SACK_PERM TSval=31...
192.168.219.129	192.168.219.129	TCP	74	9999 → 48298 [SYN, ACK] Seq=0 Ack=1 Win=33280 Len=0 MSS=65495 SACK_PE...
192.168.219.129	192.168.219.129	TCP	66	48298 → 9999 [ACK] Seq=1 Ack=1 Win=33280 Len=0 TSval=3135279950 TSecr...
192.168.219.129	192.168.219.129	TCP	77	48298 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=33280 Len=11 TSval=3135279950...
192.168.219.129	192.168.219.129	TCP	66	9999 → 48298 [ACK] Seq=1 Ack=12 Win=33280 Len=0 TSval=3135279950 TSec...
192.168.219.129	192.168.219.129	TCP	66	9999 → 48298 [FIN, ACK] Seq=1 Ack=12 Win=33280 Len=0 TSval=3135285524...
192.168.219.129	192.168.219.129	TCP	66	48298 → 9999 [FIN, ACK] Seq=12 Ack=2 Win=33280 Len=0 TSval=3135285525...
192.168.219.129	192.168.219.129	TCP	66	9999 → 48298 [ACK] Seq=2 Ack=13 Win=33280 Len=0 TSval=3135285525 TSec...
192.168.219.129	192.168.219.129	TCP	74	42496 → 9999 [SYN] Seq=0 Win=33280 Len=0 MSS=65495 SACK_PERM TSval=31...
192.168.219.129	192.168.219.129	TCP	74	9999 → 42496 [SYN, ACK] Seq=0 Ack=1 Win=33280 Len=0 MSS=65495 SACK_PE...
192.168.219.129	192.168.219.129	TCP	66	42496 → 9999 [ACK] Seq=1 Ack=1 Win=33280 Len=0 TSval=3135310061 TSecr...
192.168.219.129	192.168.219.129	TCP	77	42496 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=33280 Len=11 TSval=3135310061...
192.168.219.129	192.168.219.129	TCP	66	9999 → 42496 [ACK] Seq=1 Ack=12 Win=33280 Len=0 TSval=3135310061 TSec...

How does UDP differ from TCP?

UDP is **faster** because it just sends data without setting up a connection first. But there's **no guarantee** the data will arrive or be in order.

TCP, on the other hand, **ensures reliability** by establishing a connection first (three-way handshake) and making sure data is received correctly, but it's **slower** due to this extra processing.

Why is UDP used for DNS?

DNS needs to be **fast**, and UDP helps by sending requests without waiting for a connection. Since DNS queries are **small**, UDP works perfectly.

TCP is used only when the response is too big or extra security is needed.

What did I see in Wireshark?

DNS requests used **UDP on port 53**, with random high-numbered source ports. There was **no handshake**, making it **faster** but with **no packet recovery** if something gets lost.