Network Systems: NSS370S Dr Angus Brandt
Department of Electrical, Electronic and Computer Engineering
Cape Peninsula University of Technology

Lab 1

WireShark – Five-Layer Model

Name, Surname	Mnqobi Jeza
Student Number:	230878369
Date:	10 February 2025

Wireshark- Examining the Five-Layer Model.

The software that will be used is **WireShark**

Introduction to packet tracer.

Wireshark is a powerful open-source network protocol analyser used to capture and inspect network traffic in real-time.

It allows users to analyse packets at a granular level, making it an essential tool for network troubleshooting, security analysis, and protocol development. With its intuitive graphical interface, Wireshark enables filtering, sorting, and deep data inspection across protocols like TCP, UDP, HTTP, and more. It is widely used by network engineers, cybersecurity professionals, and researchers to diagnose network issues, detect anomalies, and understand network behaviour.

Objective: Students will use Wireshark to capture and analyse network traffic generated when visiting a website from a web browser. This lab will help students understand HTTP, HTTPS, and DNS interactions in real time and relate them to the five-layer model

Step 1: Start Wireshark and Select the Network Interface

1. Open Wireshark.
2. Identify the correct network interface:
   - If using a wired connection, select the Ethernet interface.
   - If using Wi-Fi, select the wireless network interface.
3. Click on the interface and then click **Start Capture**.

Step 2: Filter Traffic to Focus on HTTP/HTTPS and DNS

1. In the **Capture Filter** bar, type:
   - Port 80 or Port 443 or Port 53
   - This ensures only HTTP (port 80), HTTPS (port 443), and DNS (port 53) traffic is captured.
2. Click **Apply**.

Step 3: Visit a Website and Capture Traffic

1. Open a web browser.
2. In the address bar, enter http://example.com or https://www.example.com and press **Enter**. (enter your website choice here)
3. Wait for the page to load completely.
4. Return to Wireshark and click **Stop Capture**.

Step 4: Analyse the Captured Packets

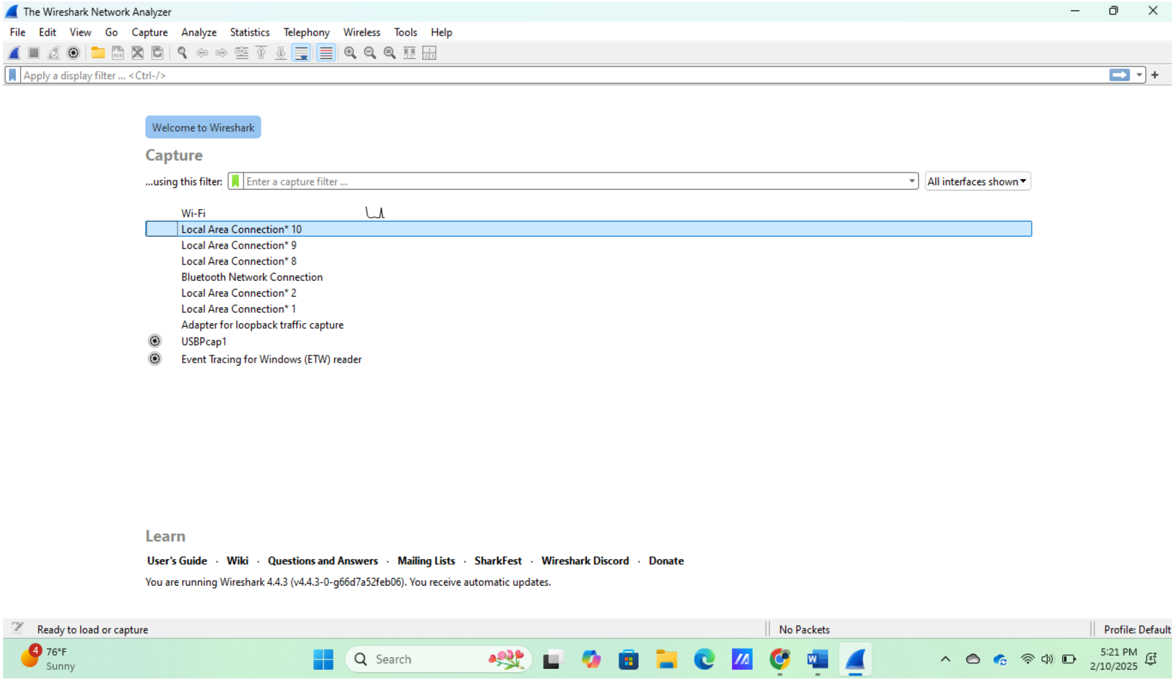
1. **Inspect DNS Queries:**
   - In the filter bar, type DNS and press **Enter**.
   - Look for DNS queries that resolve the domain name to an IP address.
   - Identify the request and response.

1. Submit the following

Provide evidence of your work by pasting the results as requested below. Increase the area of the square as needed.

1.1. Paste a screenshot here showing that Wireshark is operational. (Adjust this block size as needed)

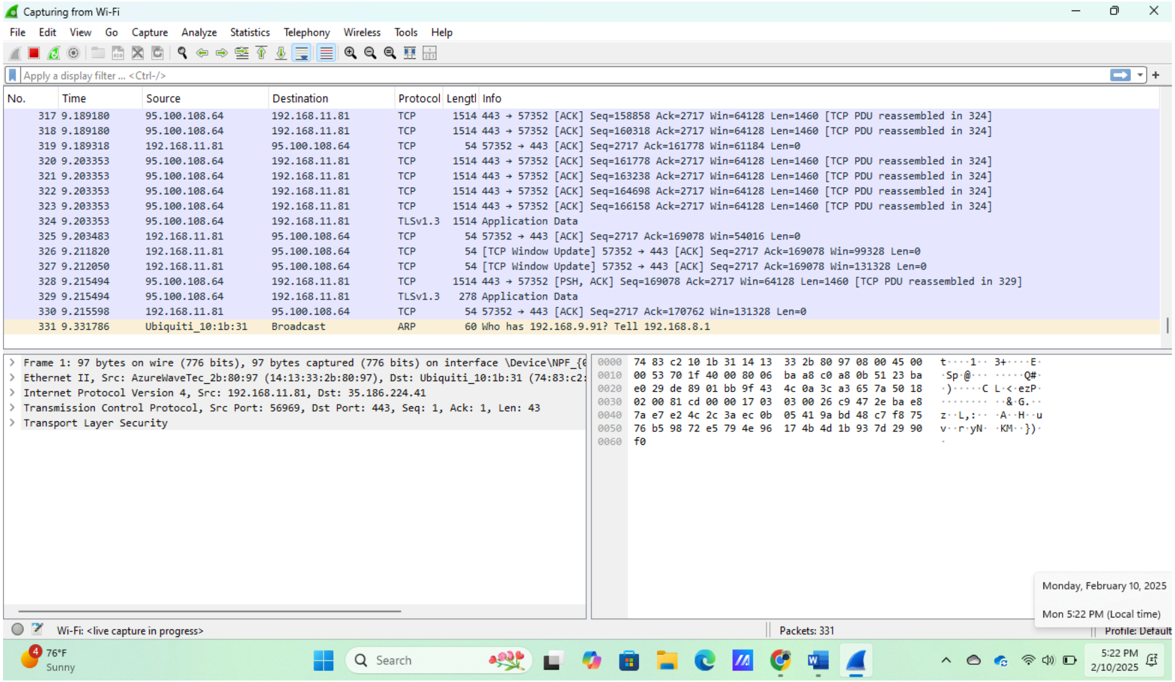
Home Page of Wireshark.



Wi-Fi
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
Bluetooth Network Connection
Local Area Connection* 2
Local Area Connection* 1
Adapter for loopback traffic capture
USBPcap1
Event Tracing for Windows (ETW) reader

You are running Wireshark 4.4.3 (v4.4.3-0-g66d7a52feb06). You receive automatic updates.

Capturing Page for functionality.



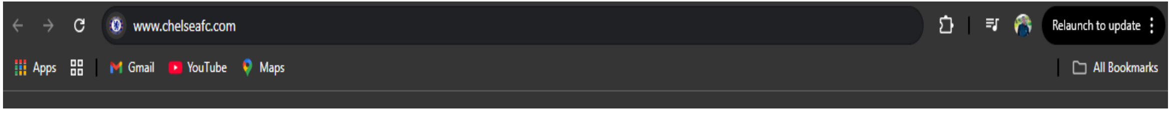
```
No.   Time        Source            Destination      Protocol Lengtl Info
  317 9.189180    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=158858 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  318 9.189180    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=160318 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  319 9.189318    192.168.11.81     95.100.108.64    TCP        54 57352 → 443 [ACK] Seq=2717 Ack=161778 Win=61184 Len=0
  320 9.203353    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=161778 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  321 9.203353    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=163238 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  322 9.203353    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=164698 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  323 9.203353    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [ACK] Seq=166158 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 324]
  324 9.203353    95.100.108.64     192.168.11.81    TLSv1.3  1514 Application Data
  325 9.203483    192.168.11.81     95.100.108.64    TCP        54 57352 → 443 [ACK] Seq=2717 Ack=169078 Win=54016 Len=0
  326 9.211820    192.168.11.81     95.100.108.64    TCP        54 [TCP Window Update] 57352 → 443 [ACK] Seq=2717 Ack=169078 Win=99328 Len=0
  327 9.212050    192.168.11.81     95.100.108.64    TCP        54 [TCP Window Update] 57352 → 443 [ACK] Seq=2717 Ack=169078 Win=131328 Len=0
  328 9.215494    95.100.108.64     192.168.11.81    TCP      1514 443 → 57352 [PSH, ACK] Seq=169078 Ack=2717 Win=64128 Len=1460 [TCP PDU reassembled in 329]
  329 9.215494    95.100.108.64     192.168.11.81    TLSv1.3   278 Application Data
  330 9.215598    192.168.11.81     95.100.108.64    TCP        54 57352 → 443 [ACK] Seq=2717 Ack=170762 Win=131328 Len=0
  331 9.331786    Ubiquiti_10:1b:31 Broadcast        ARP        60 Who has 192.168.9.91? Tell 192.168.8.1
```

```
> Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF_{6
> Ethernet II, Src: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97), Dst: Ubiquiti_10:1b:31 (74:83:c2:
> Internet Protocol Version 4, Src: 192.168.11.81, Dst: 35.186.224.41
> Transmission Control Protocol, Src Port: 56969, Dst Port: 443, Seq: 1, Ack: 1, Len: 43
> Transport Layer Security
```

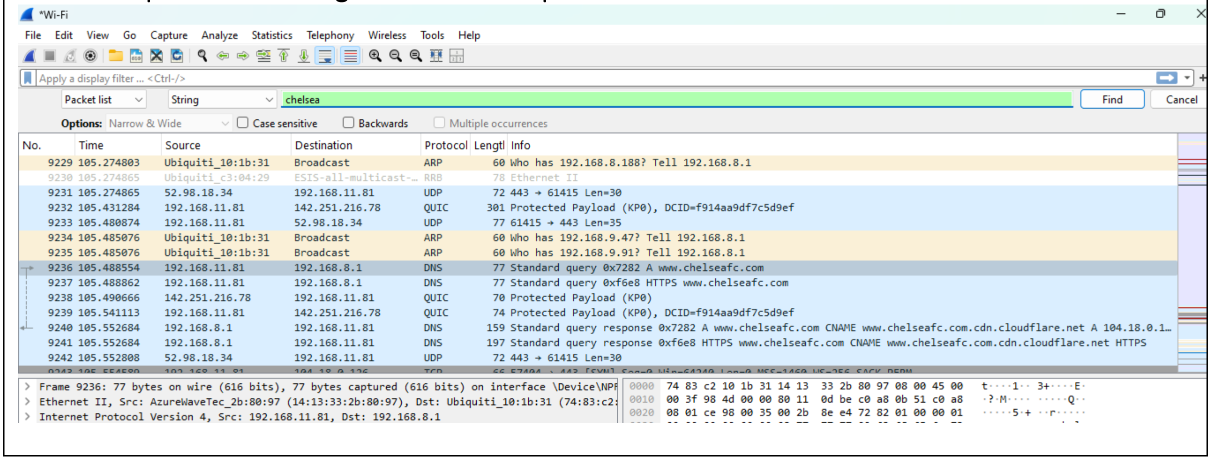
```
0000  74 83 c2 10 1b 31 14 13  33 2b 80 97 08 00 45 00   t····1·· 3+····E·
0010  00 53 70 1f 40 00 80 06  ba a8 c0 a8 0b 51 23 ba   ·Sp·@··· ·····Q#·
0020  e0 29 de 89 01 bb 9f 43  4c 0a 3c a3 65 7a 50 18   ·)·····C L·<·ezP·
0030  02 00 81 cd 00 00 17 03  03 00 26 c9 47 2e ba e8   ········ ··&·G.··
0040  7a e7 e2 4c 2c 3a ec 0b  05 41 9a bd 48 c7 f8 75   z··L,:·· ·A··H··u
0050  76 b5 98 72 e5 79 4e 96  17 4b 4d 1b 93 7d 29 90   v··r·yN· ·KM··})·
0060  f0                                                 ·
```

1.2. Past the browser screenshot here showing which website request you will be monitoring with Wireshark. (Adjust this block size as needed)

Google picture of the website.



Wireshark picture searching for the website packet..



Packet list | String | chelsea

Options: Narrow & Wide ☐ Case sensitive ☐ Backwards ☐ Multiple occurrences

```
No.   Time        Source              Destination           Protocol Lengtl Info
 9229 105.274803  Ubiquiti_10:1b:31   Broadcast             ARP       60 Who has 192.168.8.188? Tell 192.168.8.1
 9230 105.274865  Ubiquiti_c3:04:29   ESIS-all-multicast-…  RRB       78 Ethernet II
 9231 105.274865  52.98.18.34         192.168.11.81         UDP       72 443 → 61415 Len=30
 9232 105.431284  192.168.11.81       142.251.216.78        QUIC     301 Protected Payload (KP0), DCID=f914aa9df7c5d9ef
 9233 105.480874  192.168.11.81       52.98.18.34           UDP       77 61415 → 443 Len=35
 9234 105.485076  Ubiquiti_10:1b:31   Broadcast             ARP       60 Who has 192.168.9.47? Tell 192.168.8.1
 9235 105.485076  Ubiquiti_10:1b:31   Broadcast             ARP       60 Who has 192.168.9.91? Tell 192.168.8.1
 9236 105.488554  192.168.11.81       192.168.8.1           DNS       77 Standard query 0x7282 A www.chelseafc.com
 9237 105.488862  192.168.11.81       192.168.8.1           DNS       77 Standard query 0xf6e8 HTTPS www.chelseafc.com
 9238 105.490666  142.251.216.78      192.168.11.81         QUIC      70 Protected Payload (KP0)
 9239 105.541113  192.168.11.81       142.251.216.78        QUIC      74 Protected Payload (KP0), DCID=f914aa9df7c5d9ef
 9240 105.552684  192.168.8.1         192.168.11.81         DNS      159 Standard query response 0x7282 A www.chelseafc.com CNAME www.chelseafc.com.cdn.cloudflare.net A 104.18.0.1…
 9241 105.552684  192.168.8.1         192.168.11.81         DNS      197 Standard query response 0xf6e8 HTTPS www.chelseafc.com CNAME www.chelseafc.com.cdn.cloudflare.net HTTPS
 9242 105.552808  52.98.18.34         192.168.11.81         UDP       72 443 → 61415 Len=30
```

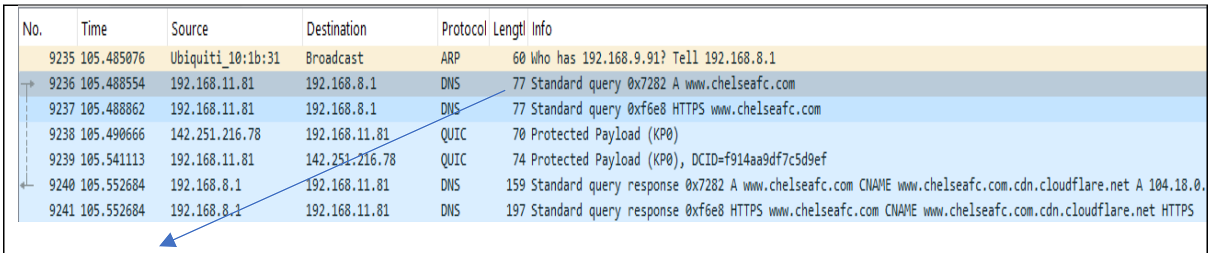
```
> Frame 9236: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF
> Ethernet II, Src: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97), Dst: Ubiquiti_10:1b:31 (74:83:c2:
> Internet Protocol Version 4, Src: 192.168.11.81, Dst: 192.168.8.1
```

```
0000  74 83 c2 10 1b 31 14 13  33 2b 80 97 08 00 45 00   t····1·· 3+····E·
0010  00 3f 98 4d 00 00 80 11  0d be c0 a8 0b 51 c0 a8   ·?·M···· ·····Q··
0020  08 01 ce 98 00 35 00 2b  8e e4 72 82 01 00 00 01   ·····5·+ ··r·····
```

1.3. After you've searched for the packet of interest on Wireshark, inspect the five-layer model on Wireshark. Take a screenshot and past it here. (Adjust this block size as needed)

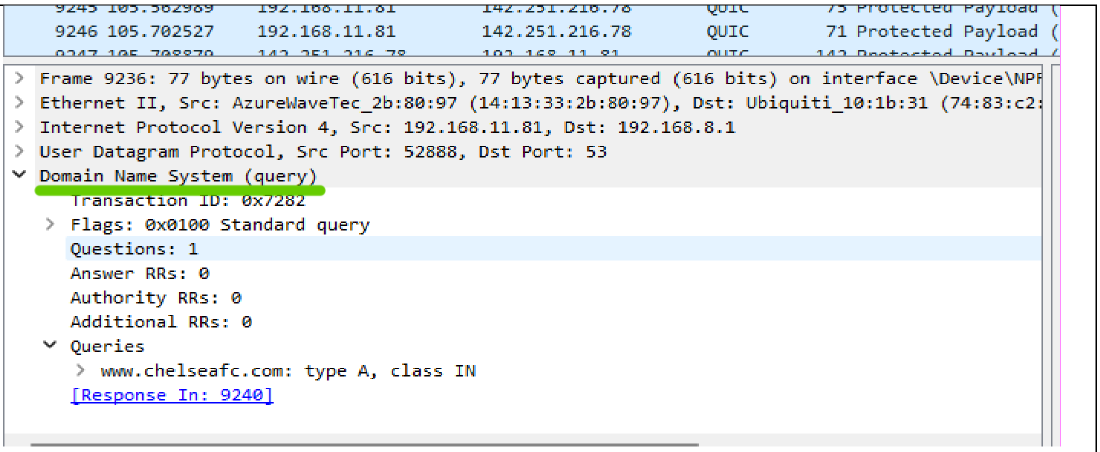
1.3.1. On the same screenshot, show how you relate the information you see on Wireshark with the five layer model. This can be done by making a drawing of the five layer model, and drawing lines linking the model to the Wireshark information. (This was shown in class)

1.3.2. Include some detail in the five layer model that you have learned after inspecting the various layers.



```
No.   Time        Source             Destination     Protocol Lengtl Info
 9235 105.485076  Ubiquiti_10:1b:31  Broadcast       ARP       60 Who has 192.168.9.91? Tell 192.168.8.1
 9236 105.488554  192.168.11.81      192.168.8.1     DNS       77 Standard query 0x7282 A www.chelseafc.com
 9237 105.488862  192.168.11.81      192.168.8.1     DNS       77 Standard query 0xf6e8 HTTPS www.chelseafc.com
 9238 105.490666  142.251.216.78     192.168.11.81   QUIC      70 Protected Payload (KP0)
 9239 105.541113  192.168.11.81      142.251.216.78  QUIC      74 Protected Payload (KP0), DCID=f914aa9df7c5d9ef
 9240 105.552684  192.168.8.1        192.168.11.81   DNS      159 Standard query response 0x7282 A www.chelseafc.com CNAME www.chelseafc.com.cdn.cloudflare.net A 104.18.0.
 9241 105.552684  192.168.8.1        192.168.11.81   DNS      197 Standard query response 0xf6e8 HTTPS www.chelseafc.com CNAME www.chelseafc.com.cdn.cloudflare.net HTTPS
```

- **DNS**: www.chelseafc.com (a HTTPS and record query) which is an **Application** category in the 5 layer model.

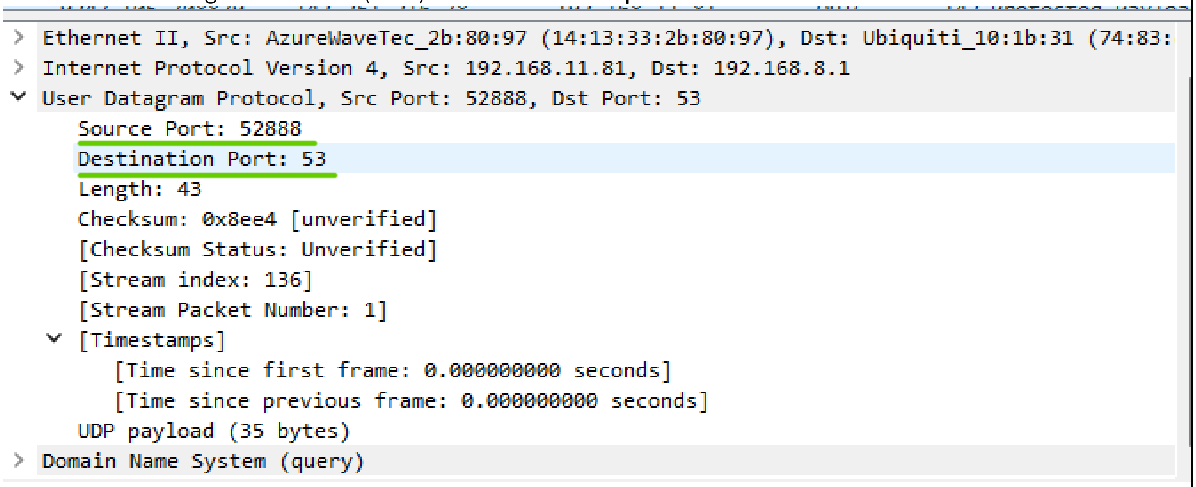


```
9245 105.582989   192.168.11.81      142.251.216.78     QUIC       75 Protected Payload (
9246 105.702527   192.168.11.81      142.251.216.78     QUIC       71 Protected Payload (
9247 105.708870   142.251.216.78     192.168.11.81      QUIC      142 Protected Payload (
```

```
> Frame 9236: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF
> Ethernet II, Src: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97), Dst: Ubiquiti_10:1b:31 (74:83:c2:
> Internet Protocol Version 4, Src: 192.168.11.81, Dst: 192.168.8.1
> User Datagram Protocol, Src Port: 52888, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x7282
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > www.chelseafc.com: type A, class IN
    [Response In: 9240]
```

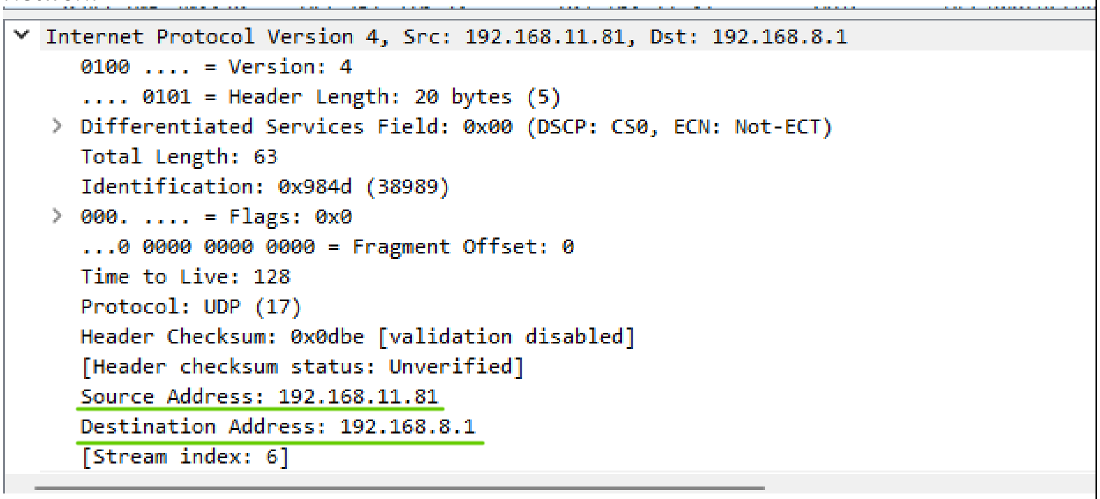
This is a more detailed DNS screenshot.

- User Datagram Protocol (**UDP**) which is the **Transport.**



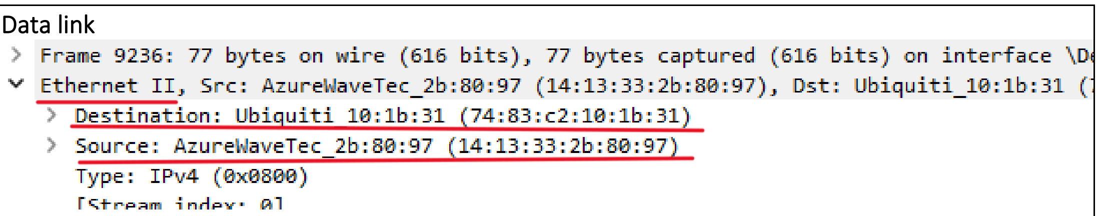
```
> Ethernet II, Src: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97), Dst: Ubiquiti_10:1b:31 (74:83:
> Internet Protocol Version 4, Src: 192.168.11.81, Dst: 192.168.8.1
v User Datagram Protocol, Src Port: 52888, Dst Port: 53
    Source Port: 52888
    Destination Port: 53
    Length: 43
    Checksum: 0x8ee4 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 136]
    [Stream Packet Number: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (35 bytes)
> Domain Name System (query)
```

- Network



```
v Internet Protocol Version 4, Src: 192.168.11.81, Dst: 192.168.8.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 63
    Identification: 0x984d (38989)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x0dbe [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.11.81
    Destination Address: 192.168.8.1
    [Stream index: 6]
```

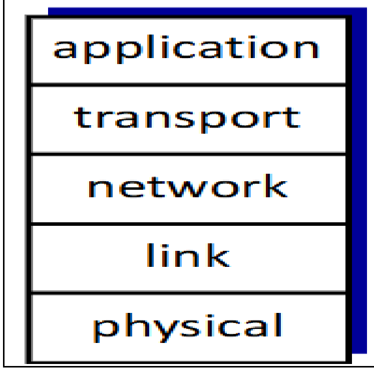
- Data link



```
> Frame 9236: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \De
v Ethernet II, Src: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97), Dst: Ubiquiti_10:1b:31 (7
  > Destination: Ubiquiti_10:1b:31 (74:83:c2:10:1b:31)
  > Source: AzureWaveTec_2b:80:97 (14:13:33:2b:80:97)
    Type: IPv4 (0x0800)
    [Stream index: 0]
```

- Physical
  Wireless WiFi

5 Layer Models



application
transport
network
link
physical

END