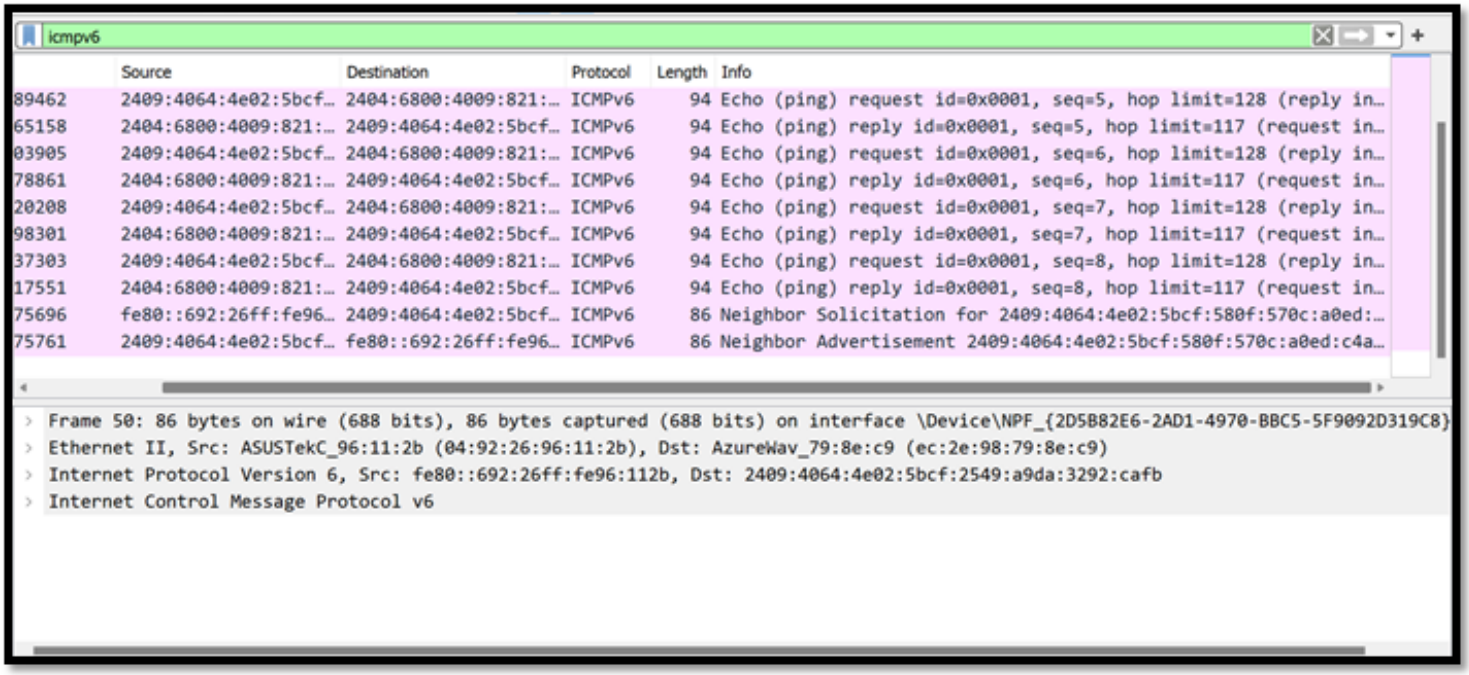
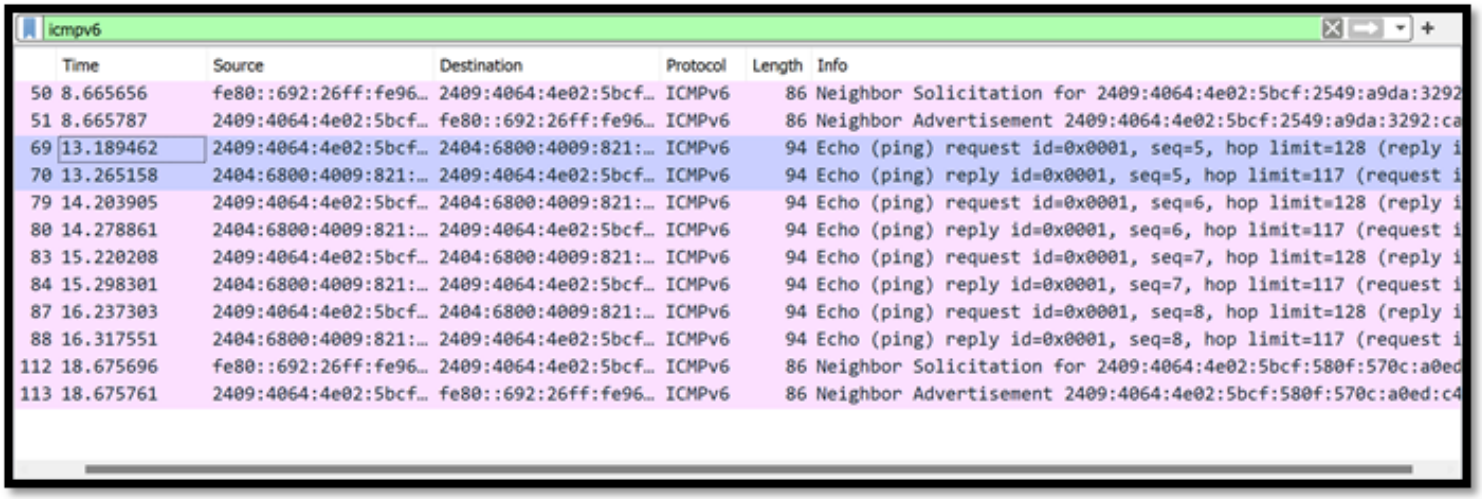
# ASSIGNMENT 5

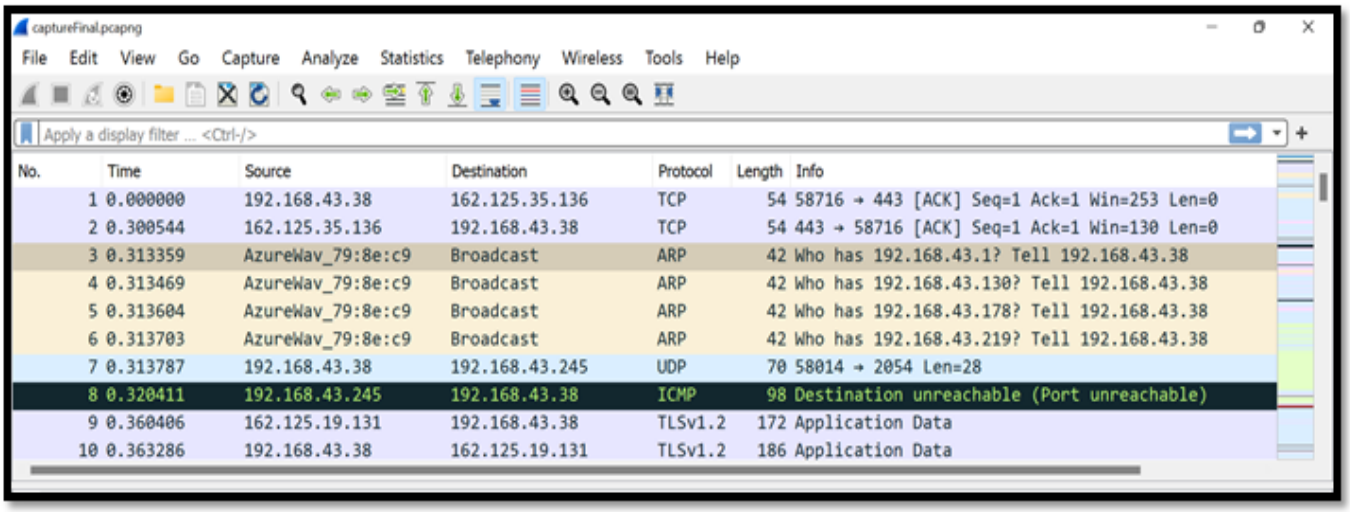
Wireshark is an open source cross-platform packet capture and analysis tool, with versions for Windows and Linux. The GUI window gives a detailed breakdown of the network protocol stack for each packet, colorizing packet details based on protocol, as well as having functionality to filter and search the traffic, and pick out TCP streams. Wireshark can also save packet data to files for offline analysis and export/import packet captures to/from other tools. Statistics can also be generated for packet capture files.

## QUESTIONS

**1. Generate some ICMP traffic by using the Ping command line tool to check the connectivity of a neighbouring machine (or router). Note the results in Wireshark. The initial ARP request broadcast from your PC determines the physical MAC address of the network IP Address, and the ARP reply from the neighbouring system. After the ARP request, the pings (ICMP echo request and replies) can be seen.**



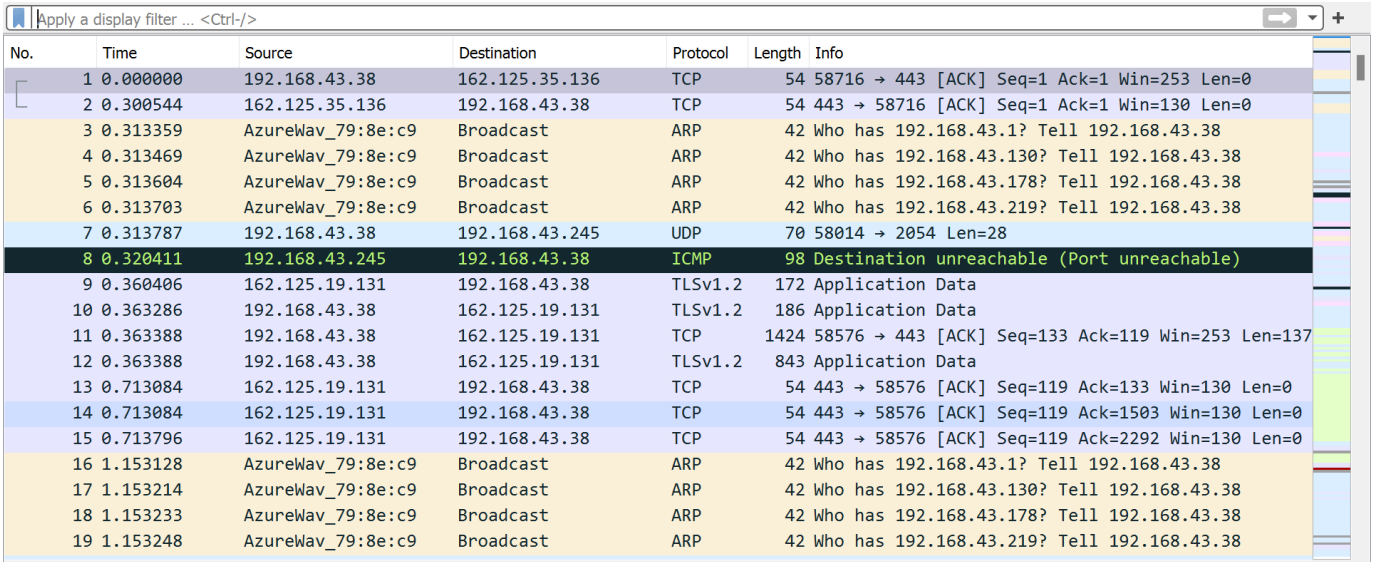



**The ICMPv6 protocol can be seen for the transfer of the packets and for the initial connection the ARP protocol can also be observed above.**

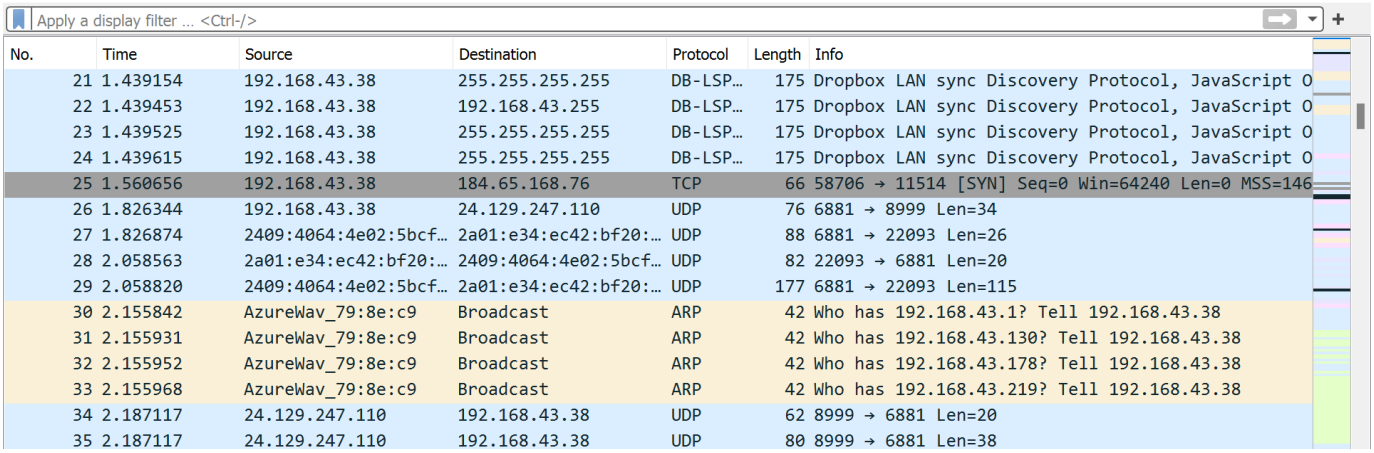
## 2. Generate some web traffic and

## a. find the list the different protocols that appear in the protocol column in the unfiltered packet-listing window of Wireshark.



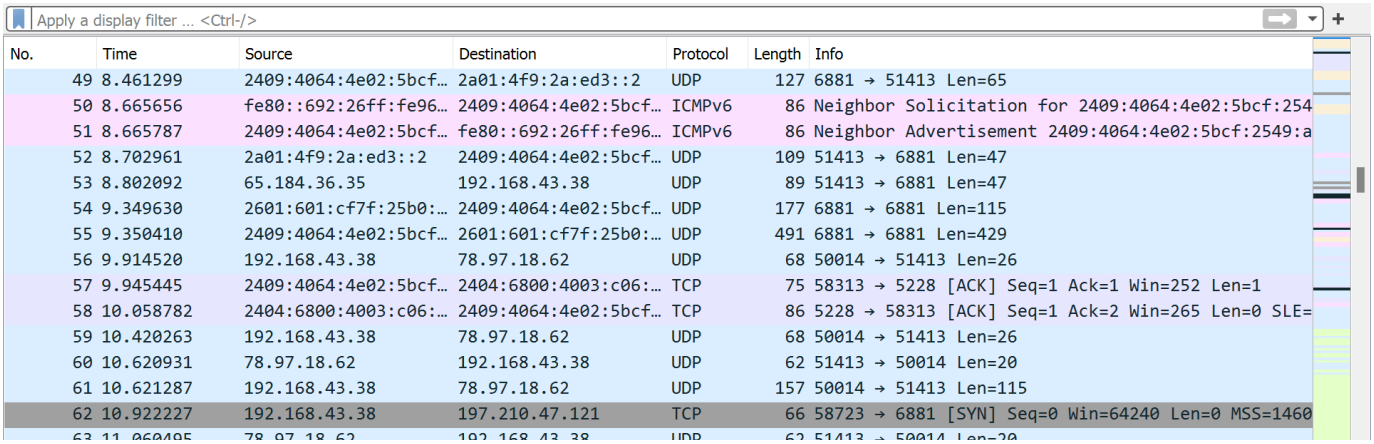
Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.43.38 | 162.125.35.136 | TCP | 54 | 58716 → 443 [ACK] Seq=1 Ack=1 Win=253 Len=0 |
| 2 | 0.300544 | 162.125.35.136 | 192.168.43.38 | TCP | 54 | 443 → 58716 [ACK] Seq=1 Ack=1 Win=130 Len=0 |
| 3 | 0.313359 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.1? Tell 192.168.43.38 |
| 4 | 0.313469 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.130? Tell 192.168.43.38 |
| 5 | 0.313604 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.178? Tell 192.168.43.38 |
| 6 | 0.313703 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.219? Tell 192.168.43.38 |
| 7 | 0.313787 | 192.168.43.38 | 192.168.43.245 | UDP | 70 | 58014 → 2054 Len=28 |
| 8 | 0.320411 | 192.168.43.245 | 192.168.43.38 | ICMP | 98 | Destination unreachable (Port unreachable) |
| 9 | 0.360406 | 162.125.19.131 | 192.168.43.38 | TLSv1.2 | 172 | Application Data |
| 10 | 0.363286 | 192.168.43.38 | 162.125.19.131 | TLSv1.2 | 186 | Application Data |
| 11 | 0.363388 | 192.168.43.38 | 162.125.19.131 | TCP | 1424 | 58576 → 443 [ACK] Seq=133 Ack=119 Win=253 Len=137 |
| 12 | 0.363388 | 192.168.43.38 | 162.125.19.131 | TLSv1.2 | 843 | Application Data |
| 13 | 0.713084 | 162.125.19.131 | 192.168.43.38 | TCP | 54 | 443 → 58576 [ACK] Seq=119 Ack=133 Win=130 Len=0 |
| 14 | 0.713084 | 162.125.19.131 | 192.168.43.38 | TCP | 54 | 443 → 58576 [ACK] Seq=119 Ack=1503 Win=130 Len=0 |
| 15 | 0.713796 | 162.125.19.131 | 192.168.43.38 | TCP | 54 | 443 → 58576 [ACK] Seq=119 Ack=2292 Win=130 Len=0 |
| 16 | 1.153128 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.1? Tell 192.168.43.38 |
| 17 | 1.153214 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.130? Tell 192.168.43.38 |
| 18 | 1.153233 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.178? Tell 192.168.43.38 |
| 19 | 1.153248 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.219? Tell 192.168.43.38 |



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

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 21 | 1.439154 | 192.168.43.38 | 255.255.255.255 | DB-LSP… | 175 | Dropbox LAN sync Discovery Protocol, JavaScript O |
| 22 | 1.439453 | 192.168.43.38 | 192.168.43.255 | DB-LSP… | 175 | Dropbox LAN sync Discovery Protocol, JavaScript O |
| 23 | 1.439525 | 192.168.43.38 | 255.255.255.255 | DB-LSP… | 175 | Dropbox LAN sync Discovery Protocol, JavaScript O |
| 24 | 1.439615 | 192.168.43.38 | 255.255.255.255 | DB-LSP… | 175 | Dropbox LAN sync Discovery Protocol, JavaScript O |
| 25 | 1.560656 | 192.168.43.38 | 184.65.168.76 | TCP | 66 | 58706 → 11514 [SYN] Seq=0 Win=64240 Len=0 MSS=146 |
| 26 | 1.826344 | 192.168.43.38 | 24.129.247.110 | UDP | 76 | 6881 → 8999 Len=34 |
| 27 | 1.826874 | 2409:4064:4e02:5bcf… | 2a01:e34:ec42:bf20:… | UDP | 88 | 6881 → 22093 Len=26 |
| 28 | 2.058563 | 2a01:e34:ec42:bf20:… | 2409:4064:4e02:5bcf… | UDP | 82 | 22093 → 6881 Len=20 |
| 29 | 2.058820 | 2409:4064:4e02:5bcf… | 2a01:e34:ec42:bf20:… | UDP | 177 | 6881 → 22093 Len=115 |
| 30 | 2.155842 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.1? Tell 192.168.43.38 |
| 31 | 2.155931 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.130? Tell 192.168.43.38 |
| 32 | 2.155952 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.178? Tell 192.168.43.38 |
| 33 | 2.155968 | AzureWav_79:8e:c9 | Broadcast | ARP | 42 | Who has 192.168.43.219? Tell 192.168.43.38 |
| 34 | 2.187117 | 24.129.247.110 | 192.168.43.38 | UDP | 62 | 8999 → 6881 Len=20 |
| 35 | 2.187117 | 24.129.247.110 | 192.168.43.38 | UDP | 80 | 8999 → 6881 Len=38 |



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

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 49 | 8.461299 | 2409:4064:4e02:5bcf… | 2a01:4f9:2a:ed3::2 | UDP | 127 | 6881 → 51413 Len=65 |
| 50 | 8.665656 | fe80::692:26ff:fe96… | 2409:4064:4e02:5bcf… | ICMPv6 | 86 | Neighbor Solicitation for 2409:4064:4e02:5bcf:254 |
| 51 | 8.665787 | 2409:4064:4e02:5bcf… | fe80::692:26ff:fe96… | ICMPv6 | 86 | Neighbor Advertisement 2409:4064:4e02:5bcf:2549:a |
| 52 | 8.702961 | 2a01:4f9:2a:ed3::2 | 2409:4064:4e02:5bcf… | UDP | 109 | 51413 → 6881 Len=47 |
| 53 | 8.802092 | 65.184.36.35 | 192.168.43.38 | UDP | 89 | 51413 → 6881 Len=47 |
| 54 | 9.349630 | 2601:601:cf7f:25b0:… | 2409:4064:4e02:5bcf… | UDP | 177 | 6881 → 6881 Len=115 |
| 55 | 9.350410 | 2409:4064:4e02:5bcf… | 2601:601:cf7f:25b0:… | UDP | 491 | 6881 → 6881 Len=429 |
| 56 | 9.914520 | 192.168.43.38 | 78.97.18.62 | UDP | 68 | 50014 → 51413 Len=26 |
| 57 | 9.945445 | 2409:4064:4e02:5bcf… | 2404:6800:4003:c06:… | TCP | 75 | 58313 → 5228 [ACK] Seq=1 Ack=1 Win=252 Len=1 |
| 58 | 10.058782 | 2404:6800:4003:c06:… | 2409:4064:4e02:5bcf… | TCP | 86 | 5228 → 58313 [ACK] Seq=1 Ack=2 Win=265 Len=0 SLE= |
| 59 | 10.420263 | 192.168.43.38 | 78.97.18.62 | UDP | 68 | 50014 → 51413 Len=26 |
| 60 | 10.620931 | 78.97.18.62 | 192.168.43.38 | UDP | 62 | 51413 → 50014 Len=20 |
| 61 | 10.621287 | 192.168.43.38 | 78.97.18.62 | UDP | 157 | 50014 → 51413 Len=115 |
| 62 | 10.922227 | 192.168.43.38 | 197.210.47.121 | TCP | 66 | 58723 → 6881 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 |
| 63 | 11.060495 | 78.97.18.62 | 192.168.43.38 | UDP | 62 | 51413 → 50014 Len=20 |

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 96 | 17.944044 | 2409:4064:4e02:5bcf… | 2001:818:df18:8b00:… | UDP | 1014 | 6881 → 49001 Len=952 |
| 97 | 17.985497 | 2409:4064:4e02:5bcf… | 2606:4700:8de4:a907… | TCP | 75 | 58461 → 443 [ACK] Seq=1 Ack=1 Win=2081 Len=1 [TCP |
| 98 | 18.025330 | 78.196.43.242 | 192.168.43.38 | UDP | 68 | 22093 → 50014 Len=26 |
| 99 | 18.025694 | 192.168.43.38 | 78.196.43.242 | UDP | 62 | 50014 → 22093 Len=20 |
| 100 | 18.089536 | 2606:4700:8de4:a907… | 2409:4064:4e02:5bcf… | TCP | 86 | 443 → 58461 [ACK] Seq=1 Ack=2 Win=74 Len=0 SLE=1 |
| 101 | 18.260705 | 78.196.43.242 | 192.168.43.38 | UDP | 142 | 22093 → 50014 Len=100 |
| 102 | 18.260964 | 192.168.43.38 | 78.196.43.242 | UDP | 62 | 50014 → 22093 Len=20 |
| 103 | 18.358665 | 2001:fb1:df:695f:81… | 2409:4064:4e02:5bcf… | UDP | 124 | 20426 → 6881 Len=62 |
| 104 | 18.359481 | 2409:4064:4e02:5bcf… | 2001:fb1:df:695f:81… | UDP | 159 | 6881 → 20426 Len=97 |
| 105 | 18.424136 | 192.168.43.38 | 20.197.71.89 | TLSv1.2 | 155 | Application Data |
| 106 | 18.470389 | 192.168.43.38 | 197.210.47.121 | TCP | 55 | [TCP ZeroWindowProbe] 58723 → 6881 [ACK] Seq=1 Ac |
| 107 | 18.486548 | 192.168.43.38 | 173.212.227.13 | UDP | 107 | 6881 → 56988 Len=65 |
| 108 | 18.486852 | 2409:4064:4e02:5bcf… | 2804:d4b:79b1:3e00:… | UDP | 166 | 6881 → 6881 Len=104 |
| 109 | 18.487004 | 2409:4064:4e02:5bcf… | 240d:1a:4b0:200:6d0… | UDP | 127 | 6881 → 18586 Len=65 |
| 110 | 18.566417 | 20.197.71.89 | 192.168.43.38 | TLSv1.2 | 225 | Application Data |
| 111 | 18.611071 | 192.168.43.38 | 20.197.71.89 | TCP | 54 | 49431 → 443 [ACK] Seq=102 Ack=172 Win=256 Len=0 |

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 199 | 23.391798 | 41.62.102.96 | 192.168.43.38 | UDP | 62 | 12522 → 6881 Len=20 |
| 200 | 23.392161 | 192.168.43.38 | 222.254.172.132 | UDP | 146 | 6881 → 6881 Len=104 |
| 201 | 23.499208 | 192.168.43.38 | 191.189.210.42 | UDP | 107 | 6881 → 50321 Len=65 |
| 202 | 23.499568 | 2409:4064:4e02:5bcf… | 2a01:e34:eec8:7dd0:… | UDP | 166 | 6881 → 30295 Len=104 |
| 203 | 23.499729 | 2409:4064:4e02:5bcf… | 2a01:e0a:3b2:ede0:2… | UDP | 127 | 6881 → 6881 Len=65 |
| 204 | 23.719259 | 2a01:e0a:3b2:ede0:2… | 2409:4064:4e02:5bcf… | UDP | 152 | 6881 → 6881 Len=90 |
| 205 | 23.721046 | 41.62.102.96 | 192.168.43.38 | UDP | 62 | 12522 → 6881 Len=20 |
| 206 | 23.726961 | 41.62.102.96 | 192.168.43.38 | UDP | 62 | 12522 → 6881 Len=20 |
| 207 | 23.733256 | 192.168.43.38 | 239.255.255.250 | SSDP | 215 | M-SEARCH * HTTP/1.1 |
| 208 | 23.982086 | 191.189.210.42 | 192.168.43.38 | UDP | 119 | 50321 → 6881 Len=77 |
| 209 | 24.108543 | 192.168.43.38 | 41.62.102.96 | TCP | 66 | 58732 → 12522 [SYN] Seq=0 Win=64240 Len=0 MSS=146 |
| 210 | 24.418384 | 2a01:e0a:511:8e50:4… | 2409:4064:4e02:5bcf… | UDP | 177 | 42084 → 6881 Len=115 |
| 211 | 24.419127 | 2409:4064:4e02:5bcf… | 2a01:e0a:511:8e50:4… | UDP | 492 | 6881 → 42084 Len=430 |
| 212 | 24.457442 | 41.62.102.96 | 192.168.43.38 | TCP | 66 | 12522 → 58732 [SYN, ACK] Seq=0 Ack=1 Win=65535 Le |
| 213 | 24.457654 | 192.168.43.38 | 41.62.102.96 | TCP | 54 | 58732 → 12522 [ACK] Seq=1 Ack=1 Win=65536 Len=0 |
| 214 | 24.458063 | 192.168.43.38 | 41.62.102.96 | BitTor… | 122 | Handshake |

**The list of Protocols appearing are :**

**1.TCP**

**2.ARP**

**3.ICMP**

**4.TLSv1.2**

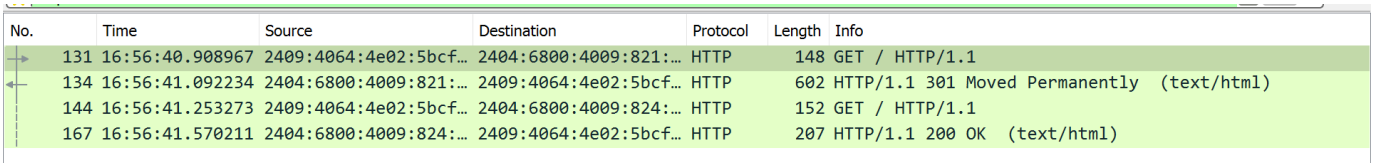
**5.UDP**

**6.SSDP**

**7.ICMPv6**

**b. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing**

**began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)**

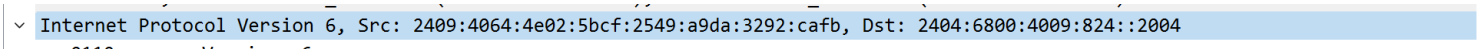


| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 131 | 16:56:40.908967 | 2409:4064:4e02:5bcf… | 2404:6800:4009:821:… | HTTP | 148 | GET / HTTP/1.1 |
| 134 | 16:56:41.092234 | 2404:6800:4009:821:… | 2409:4064:4e02:5bcf… | HTTP | 602 | HTTP/1.1 301 Moved Permanently (text/html) |
| 144 | 16:56:41.253273 | 2409:4064:4e02:5bcf… | 2404:6800:4009:824:… | HTTP | 152 | GET / HTTP/1.1 |
| 167 | 16:56:41.570211 | 2404:6800:4009:824:… | 2409:4064:4e02:5bcf… | HTTP | 207 | HTTP/1.1 200 OK (text/html) |

**Time difference between the GET message and the OK reply :**

**41.570211-41.253273=0.31316938 seconds**

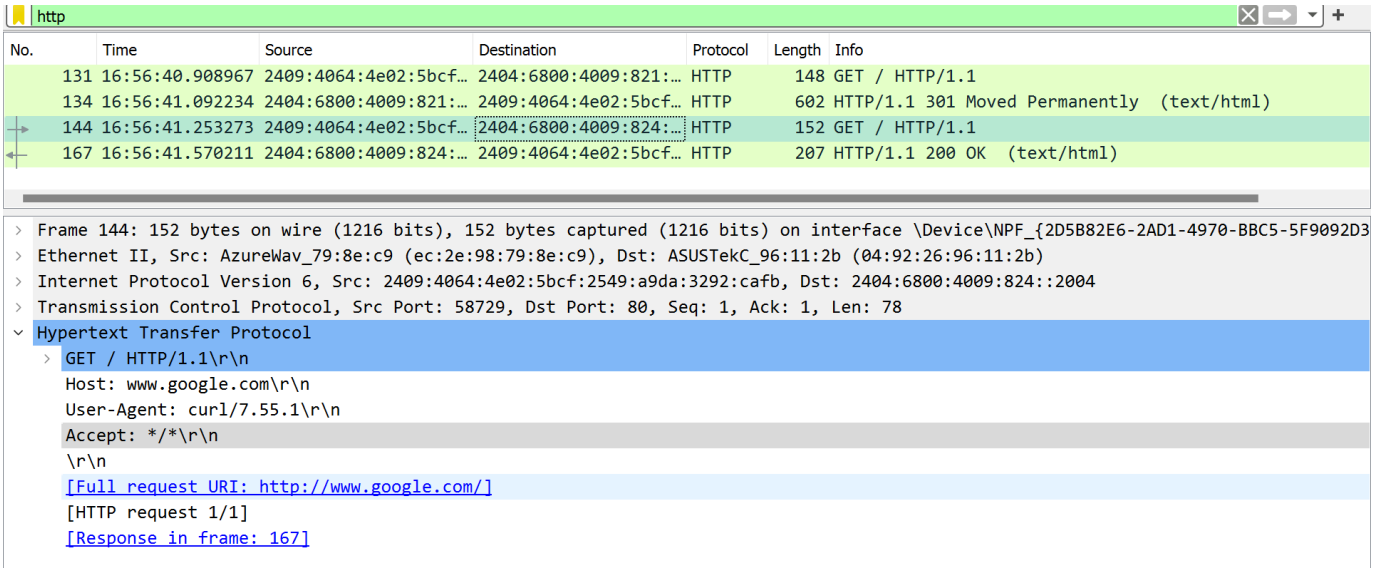
## c. What is the Internet address of the website? What is the Internet address of your computer?



```
Internet Protocol Version 6, Src: 2409:4064:4e02:5bcf:2549:a9da:3292:cafb, Dst: 2404:6800:4009:824::2004
```

**The Src denotes the IPv6 128 bit address of my computer and the Dst gives the destination 128 bit IPv6 address.**

## d. Search back through your capture, and find an HTTP packet containing a GET command. Click on the packet in the Packet List Panel. Then expand the HTTP layer in the Packet Details Panel, from the packet.

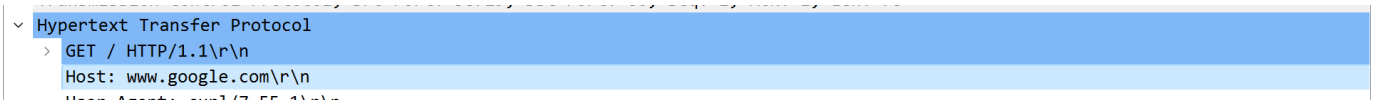


http

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 131 | 16:56:40.908967 | 2409:4064:4e02:5bcf… | 2404:6800:4009:821:… | HTTP | 148 | GET / HTTP/1.1 |
| 134 | 16:56:41.092234 | 2404:6800:4009:821:… | 2409:4064:4e02:5bcf… | HTTP | 602 | HTTP/1.1 301 Moved Permanently (text/html) |
| 144 | 16:56:41.253273 | 2409:4064:4e02:5bcf… | 2404:6800:4009:824:… | HTTP | 152 | GET / HTTP/1.1 |
| 167 | 16:56:41.570211 | 2404:6800:4009:824:… | 2409:4064:4e02:5bcf… | HTTP | 207 | HTTP/1.1 200 OK (text/html) |

```
Frame 144: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface \Device\NPF_{2D5B82E6-2AD1-4970-BBC5-5F9092D3
Ethernet II, Src: AzureWav_79:8e:c9 (ec:2e:98:79:8e:c9), Dst: ASUSTekC_96:11:2b (04:92:26:96:11:2b)
Internet Protocol Version 6, Src: 2409:4064:4e02:5bcf:2549:a9da:3292:cafb, Dst: 2404:6800:4009:824::2004
Transmission Control Protocol, Src Port: 58729, Dst Port: 80, Seq: 1, Ack: 1, Len: 78
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: www.google.com\r\n
  User-Agent: curl/7.55.1\r\n
  Accept: */*\r\n
  \r\n
  [Full request URI: http://www.google.com/]
  [HTTP request 1/1]
  [Response in frame: 167]
```

**e. Find out the value of the Host from the Packet Details Panel, within the GET command.**




**3. Highlight the Hex and ASCII representations of the packet in the Packet Bytes Panel.**

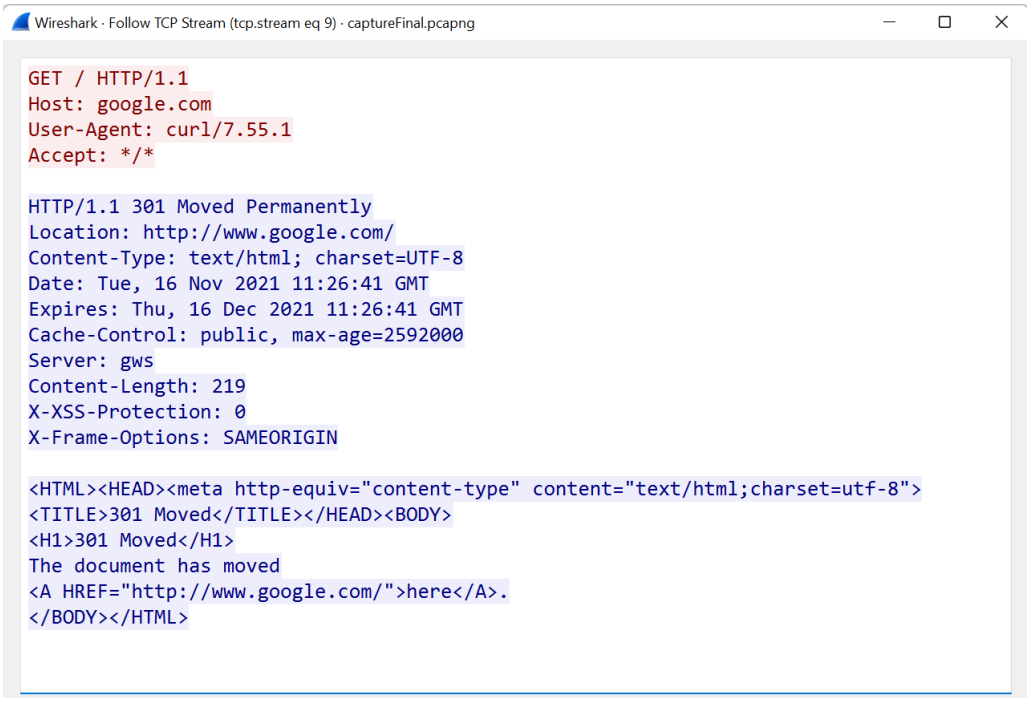

**The left two column represents the hex and the right column represents the Ascii for the Packets.**

**4. Find out the first 4 bytes of the Hex valueof the Host parameter from the Packet Bytes Panel.**

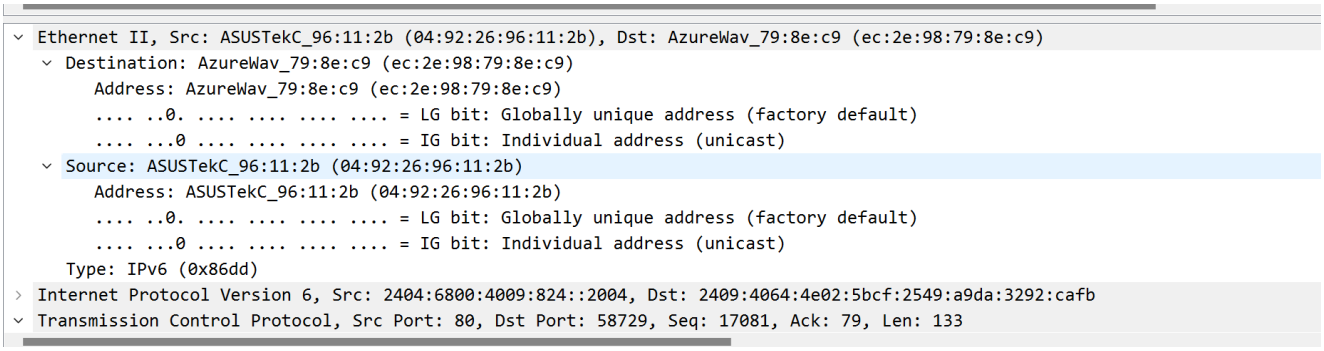

**The first four bytes are : 48 6f 73 74**

**5. Filter packets with http, TCP, DNS and otherprotocols.**

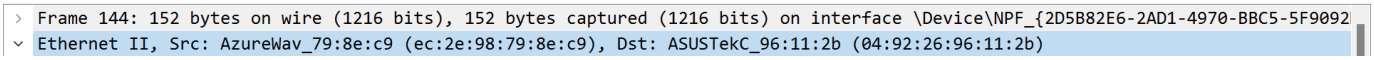
**a. Find out what are those packets contain by following one of the conversations (also called network flows), select one of the packets and press the right mouse button..click on follow.**




**6. Search through your capture, and find an HTTP packet coming back from the server (TCP Source Port == 80). Expand the Ethernet layer in the Packet Details Panel.**



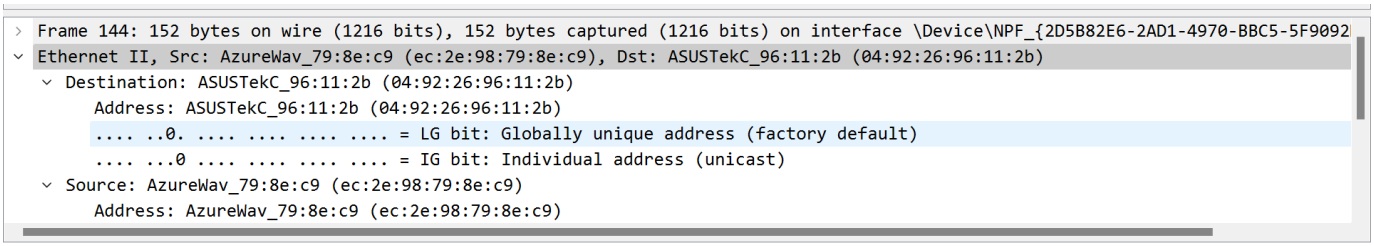

**7. What are the manufacturers of your PC's Network Interface Card (NIC), and the servers NIC?**




**PC NIC MANUFACTURER:AZUREWAVE TECHNOLOGIES INC.**

**SERVERS NIC:ASUSTEK COMPUTER INC.**

## 8. What are the Hex values (shown the raw bytes panel) of the two NICS Manufacturers OUIs?

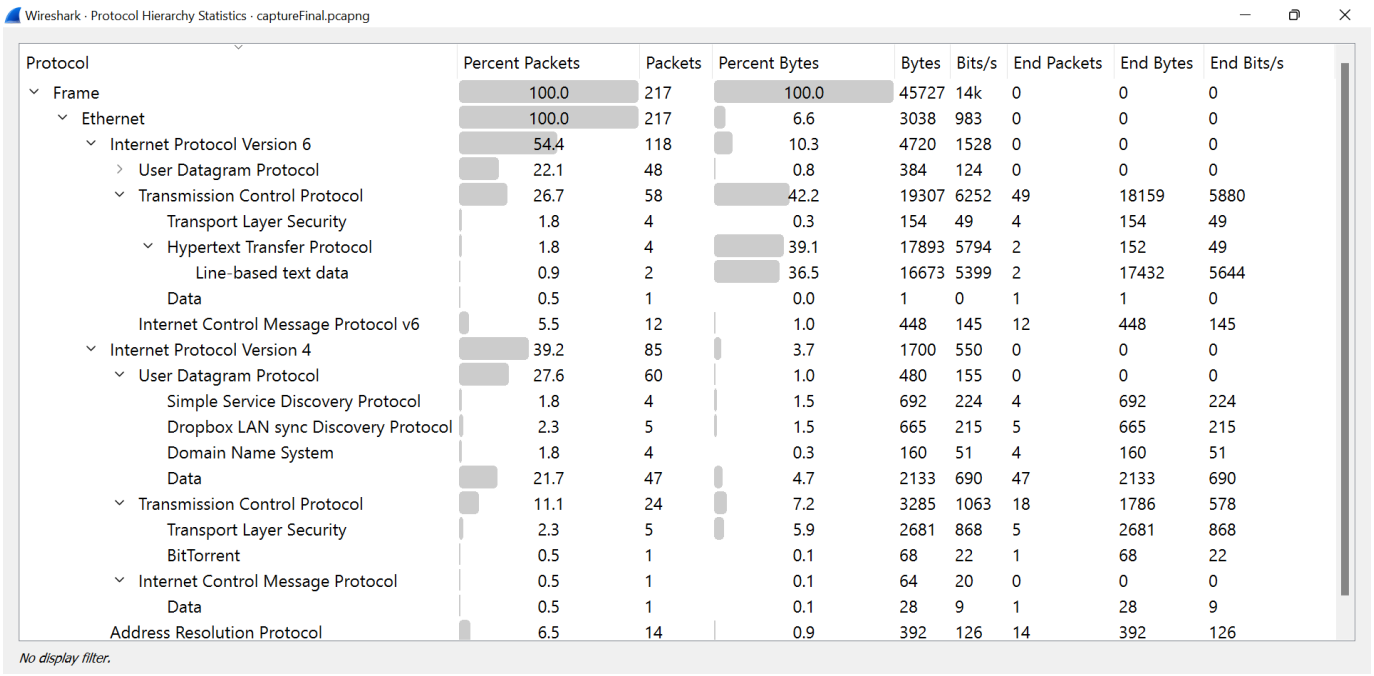



**Pc manufacturers OUI (hex): EC:2E:98**

**SERVERS NIC MANUFACTURER'S OUI(hex):04:92:26**

## 9. Find the following statistics:

## a. What percentage of packets in your capture are TCP, and give an example of the higher level protocol which uses TCP?




| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 217 | 100.0 | 45727 | 14k | 0 | 0 | 0 |
| Ethernet | 100.0 | 217 | 6.6 | 3038 | 983 | 0 | 0 | 0 |
| Internet Protocol Version 6 | 54.4 | 118 | 10.3 | 4720 | 1528 | 0 | 0 | 0 |
| User Datagram Protocol | 22.1 | 48 | 0.8 | 384 | 124 | 0 | 0 | 0 |
| Transmission Control Protocol | 26.7 | 58 | 42.2 | 19307 | 6252 | 49 | 18159 | 5880 |
| Transport Layer Security | 1.8 | 4 | 0.3 | 154 | 49 | 4 | 154 | 49 |
| Hypertext Transfer Protocol | 1.8 | 4 | 39.1 | 17893 | 5794 | 2 | 152 | 49 |
| Line-based text data | 0.9 | 2 | 36.5 | 16673 | 5399 | 2 | 17432 | 5644 |
| Data | 0.5 | 1 | 0.0 | 1 | 0 | 1 | 1 | 0 |
| Internet Control Message Protocol v6 | 5.5 | 12 | 1.0 | 448 | 145 | 12 | 448 | 145 |
| Internet Protocol Version 4 | 39.2 | 85 | 3.7 | 1700 | 550 | 0 | 0 | 0 |
| User Datagram Protocol | 27.6 | 60 | 1.0 | 480 | 155 | 0 | 0 | 0 |
| Simple Service Discovery Protocol | 1.8 | 4 | 1.5 | 692 | 224 | 4 | 692 | 224 |
| Dropbox LAN sync Discovery Protocol | 2.3 | 5 | 1.5 | 665 | 215 | 5 | 665 | 215 |
| Domain Name System | 1.8 | 4 | 0.3 | 160 | 51 | 4 | 160 | 51 |
| Data | 21.7 | 47 | 4.7 | 2133 | 690 | 47 | 2133 | 690 |
| Transmission Control Protocol | 11.1 | 24 | 7.2 | 3285 | 1063 | 18 | 1786 | 578 |
| Transport Layer Security | 2.3 | 5 | 5.9 | 2681 | 868 | 5 | 2681 | 868 |
| BitTorrent | 0.5 | 1 | 0.1 | 68 | 22 | 1 | 68 | 22 |
| Internet Control Message Protocol | 0.5 | 1 | 0.1 | 64 | 20 | 0 | 0 | 0 |
| Data | 0.5 | 1 | 0.1 | 28 | 9 | 1 | 28 | 9 |
| Address Resolution Protocol | 6.5 | 14 | 0.9 | 392 | 126 | 14 | 392 | 126 |

**TCP PACKETS: 11.1 %**

## b. What percentage of packets in your capture are UDP, and give an example of the higher level protocol which uses UDP?

Wireshark · Protocol Hierarchy Statistics · captureFinal.pcapng

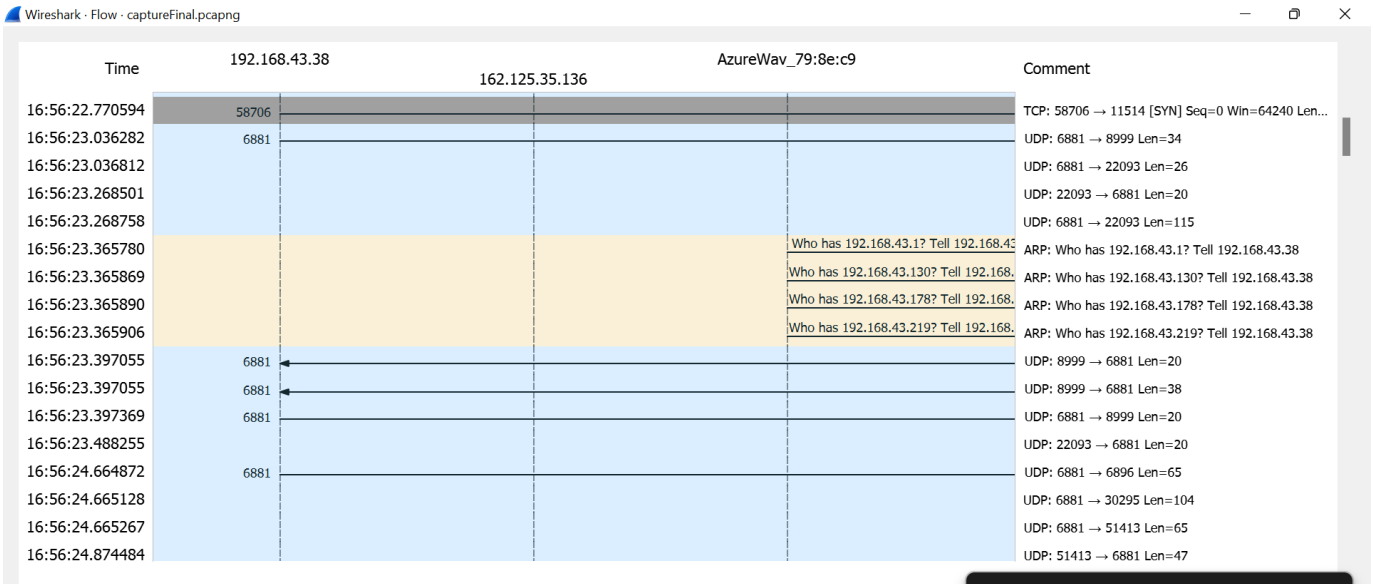
| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 217 | 100.0 | 45727 | 14k | 0 | 0 | 0 |
| Ethernet | 100.0 | 217 | 6.6 | 3038 | 983 | 0 | 0 | 0 |
| Internet Protocol Version 6 | 54.4 | 118 | 10.3 | 4720 | 1528 | 0 | 0 | 0 |
| User Datagram Protocol | 22.1 | 48 | 0.8 | 384 | 124 | 0 | 0 | 0 |
| Transmission Control Protocol | 26.7 | 58 | 42.2 | 19307 | 6252 | 49 | 18159 | 5880 |
| Transport Layer Security | 1.8 | 4 | 0.3 | 154 | 49 | 4 | 154 | 49 |
| Hypertext Transfer Protocol | 1.8 | 4 | 39.1 | 17893 | 5794 | 2 | 152 | 49 |
| Line-based text data | 0.9 | 2 | 36.5 | 16673 | 5399 | 2 | 17432 | 5644 |
| Data | 0.5 | 1 | 0.0 | 1 | 0 | 1 | 1 | 0 |
| Internet Control Message Protocol v6 | 5.5 | 12 | 1.0 | 448 | 145 | 12 | 448 | 145 |
| Internet Protocol Version 4 | 39.2 | 85 | 3.7 | 1700 | 550 | 0 | 0 | 0 |
| User Datagram Protocol | 27.6 | 60 | 1.0 | 480 | 155 | 0 | 0 | 0 |
| Simple Service Discovery Protocol | 1.8 | 4 | 1.5 | 692 | 224 | 4 | 692 | 224 |
| Dropbox LAN sync Discovery Protocol | 2.3 | 5 | 1.5 | 665 | 215 | 5 | 665 | 215 |
| Domain Name System | 1.8 | 4 | 0.3 | 160 | 51 | 4 | 160 | 51 |
| Data | 21.7 | 47 | 4.7 | 2133 | 690 | 47 | 2133 | 690 |
| Transmission Control Protocol | 11.1 | 24 | 7.2 | 3285 | 1063 | 18 | 1786 | 578 |
| Transport Layer Security | 2.3 | 5 | 5.9 | 2681 | 868 | 5 | 2681 | 868 |
| BitTorrent | 0.5 | 1 | 0.1 | 68 | 22 | 1 | 68 | 22 |
| Internet Control Message Protocol | 0.5 | 1 | 0.1 | 64 | 20 | 0 | 0 | 0 |
| Data | 0.5 | 1 | 0.1 | 28 | 9 | 1 | 28 | 9 |
| Address Resolution Protocol | 6.5 | 14 | 0.9 | 392 | 126 | 14 | 392 | 126 |

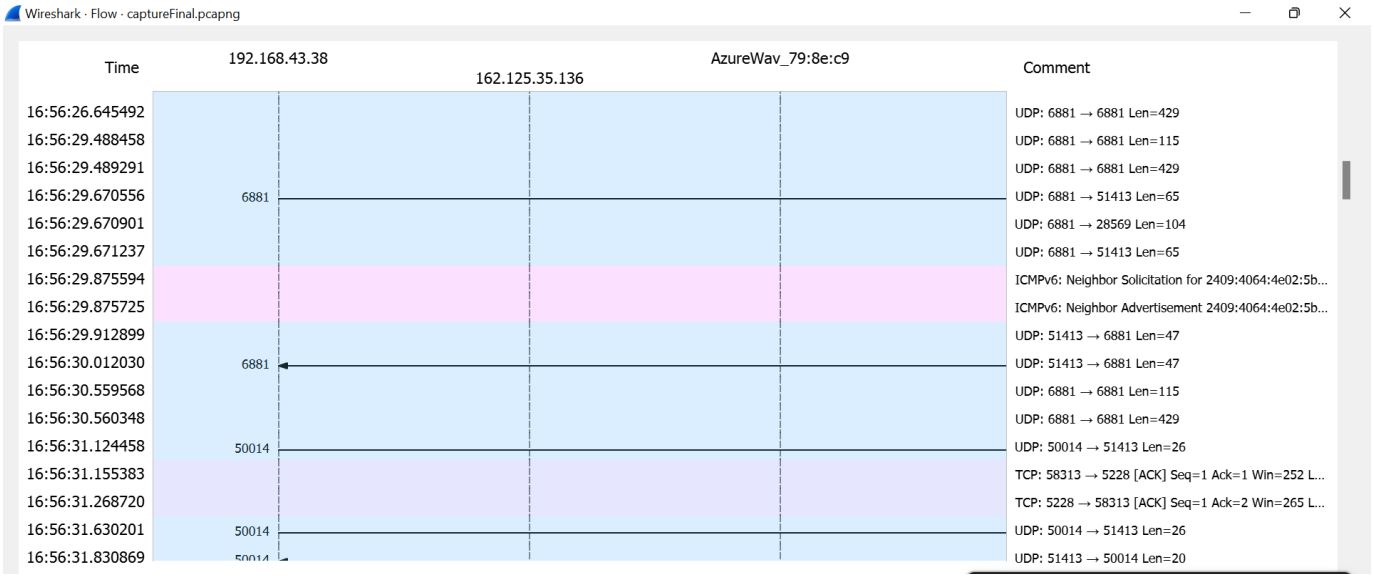
**UDP PACKETS:22.1 %**

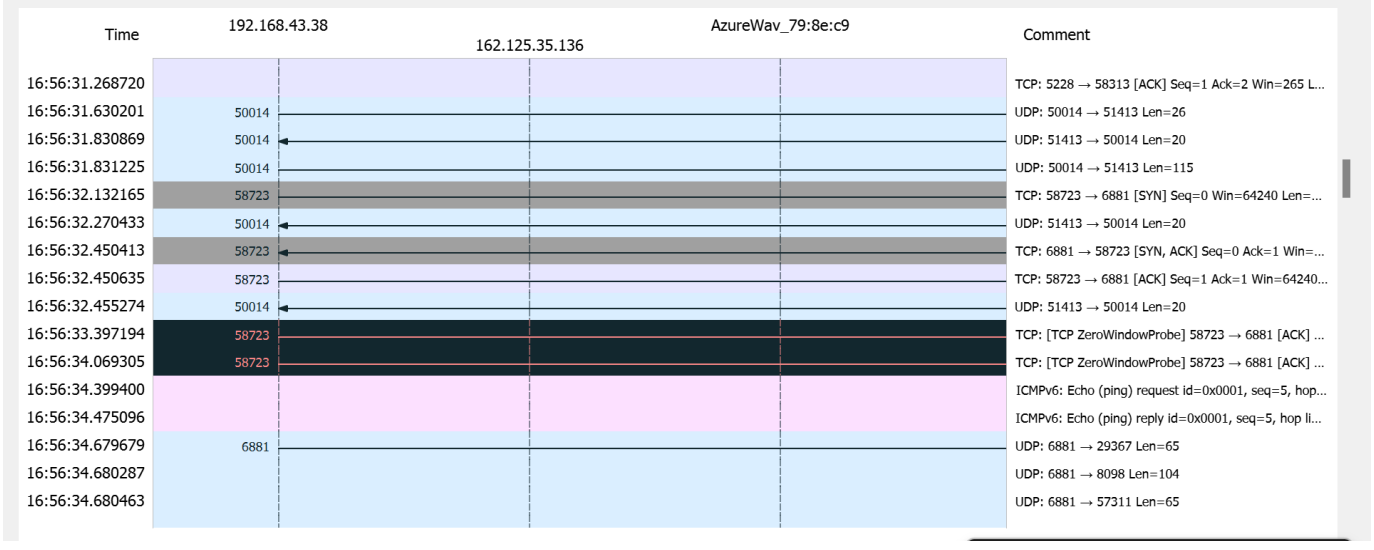
**10. Find the traffic flow Select the Statistics->FlowGraph menu option. Choose General Flow and Network Source options, and click the OK button.**

Wireshark · Flow · captureFinal.pcapng

| Time | 192.168.43.38 | 162.125.35.136 | AzureWav_79:8e:c9 | Comment |
|---|---|---|---|---|
| 16:56:21.570344 | 58576 | | Application Data | TLSv1.2: Application Data |
| 16:56:21.573224 | 58576 | | Application Data | TLSv1.2: Application Data |
| 16:56:21.573326 | 58576 | 58576 → 443 [ACK] Seq=133 Ack=119 Win=253 Len=1370 [TCP | | TCP: 58576 → 443 [ACK] Seq=133 Ack=119 Win=25... |
| 16:56:21.573326 | 58576 | | Application Data | TLSv1.2: Application Data |
| 16:56:21.923022 | 58576 | 443 → 58576 [ACK] Seq=119 Ack=133 Win= | | TCP: 443 → 58576 [ACK] Seq=119 Ack=133 Win=13... |
| 16:56:21.923022 | 58576 | 443 → 58576 [ACK] Seq=119 Ack=1503 Win= | | TCP: 443 → 58576 [ACK] Seq=119 Ack=1503 Win=1... |
| 16:56:21.923734 | 58576 | 443 → 58576 [ACK] Seq=119 Ack=2292 Win= | | TCP: 443 → 58576 [ACK] Seq=119 Ack=2292 Win=1... |
| 16:56:22.363066 | | | Who has 192.168.43.1? Tell 192.168.43 | ARP: Who has 192.168.43.1? Tell 192.168.43.38 |
| 16:56:22.363152 | | | Who has 192.168.43.130? Tell 192.168. | ARP: Who has 192.168.43.130? Tell 192.168.43.38 |
| 16:56:22.363171 | | | Who has 192.168.43.178? Tell 192.168. | ARP: Who has 192.168.43.178? Tell 192.168.43.38 |
| 16:56:22.363186 | | | Who has 192.168.43.219? Tell 192.168. | ARP: Who has 192.168.43.219? Tell 192.168.43.38 |
| 16:56:22.646488 | 17500 | | Dropbox LAN sync Discovery | DB-LSP-DISC/JSON: Dropbox LAN sync Discovery Pro... |
| 16:56:22.649092 | 17500 | | Dropbox LAN sync Discovery | DB-LSP-DISC/JSON: Dropbox LAN sync Discovery Pro... |
| 16:56:22.649391 | 17500 | | Dropb | DB-LSP-DISC/JSON: Dropbox LAN sync Discovery Pro... |
| 16:56:22.649463 | 17500 | | Dropbox LAN sync Discovery | DB-LSP-DISC/JSON: Dropbox LAN sync Discovery Pro... |
| 16:56:22.649553 | 17500 | | Dropbox LAN sync Discovery | DB-LSP-DISC/JSON: Dropbox LAN sync Discovery Pro... |
| 16:56:22.770594 | 58706 | | | TCP: 58706 → 11514 [SYN] Seq=0 Win=64240 Len... |


Wireshark · Flow · captureFinal.pcapng
Time
192.168.43.38
162.125.35.136
AzureWav_79:8e:c9
Comment
16:56:22.770594 58706 TCP: 58706 → 11514 [SYN] Seq=0 Win=64240 Len...
16:56:23.036282 6881 UDP: 6881 → 8999 Len=34
16:56:23.036812 UDP: 6881 → 22093 Len=26
16:56:23.268501 UDP: 22093 → 6881 Len=20
16:56:23.268758 UDP: 6881 → 22093 Len=115
16:56:23.365780 Who has 192.168.43.1? Tell 192.168.43 ARP: Who has 192.168.43.1? Tell 192.168.43.38
16:56:23.365869 Who has 192.168.43.130? Tell 192.168. ARP: Who has 192.168.43.130? Tell 192.168.43.38
16:56:23.365890 Who has 192.168.43.178? Tell 192.168. ARP: Who has 192.168.43.178? Tell 192.168.43.38
16:56:23.365906 Who has 192.168.43.219? Tell 192.168. ARP: Who has 192.168.43.219? Tell 192.168.43.38
16:56:23.397055 6881 UDP: 8999 → 6881 Len=20
16:56:23.397055 6881 UDP: 8999 → 6881 Len=38
16:56:23.397369 6881 UDP: 6881 → 8999 Len=20
16:56:23.488255 UDP: 22093 → 6881 Len=20
16:56:24.664872 6881 UDP: 6881 → 6896 Len=65
16:56:24.665128 UDP: 6881 → 30295 Len=104
16:56:24.665267 UDP: 6881 → 51413 Len=65
16:56:24.874484 UDP: 51413 → 6881 Len=47


Wireshark · Flow · captureFinal.pcapng
Time
192.168.43.38
162.125.35.136
AzureWav_79:8e:c9
Comment
16:56:26.645492 UDP: 6881 → 6881 Len=429
16:56:29.488458 UDP: 6881 → 6881 Len=115
16:56:29.489291 UDP: 6881 → 6881 Len=429
16:56:29.670556 6881 UDP: 6881 → 51413 Len=65
16:56:29.670901 UDP: 6881 → 28569 Len=104
16:56:29.671237 UDP: 6881 → 51413 Len=65
16:56:29.875594 ICMPv6: Neighbor Solicitation for 2409:4064:4e02:5b...
16:56:29.875725 ICMPv6: Neighbor Advertisement 2409:4064:4e02:5b...
16:56:29.912899 UDP: 51413 → 6881 Len=47
16:56:30.012030 6881 UDP: 51413 → 6881 Len=47
16:56:30.559568 UDP: 6881 → 6881 Len=115
16:56:30.560348 UDP: 6881 → 6881 Len=429
16:56:31.124458 50014 UDP: 50014 → 51413 Len=26
16:56:31.155383 TCP: 58313 → 5228 [ACK] Seq=1 Ack=1 Win=252 L...
16:56:31.268720 TCP: 5228 → 58313 [ACK] Seq=1 Ack=2 Win=265 L...
16:56:31.630201 50014 UDP: 50014 → 51413 Len=26
16:56:31.830869 UDP: 51413 → 50014 Len=20


Time
192.168.43.38
162.125.35.136
AzureWav_79:8e:c9
Comment
16:56:31.268720 TCP: 5228 → 58313 [ACK] Seq=1 Ack=2 Win=265 L...
16:56:31.630201 50014 UDP: 50014 → 51413 Len=26
16:56:31.830869 50014 UDP: 51413 → 50014 Len=20
16:56:31.831225 50014 UDP: 50014 → 51413 Len=115
16:56:32.132165 58723 TCP: 58723 → 6881 [SYN] Seq=0 Win=64240 Len=...
16:56:32.270433 50014 UDP: 51413 → 50014 Len=20
16:56:32.450413 58723 TCP: 6881 → 58723 [SYN, ACK] Seq=0 Ack=1 Win=...
16:56:32.450635 58723 TCP: 58723 → 6881 [ACK] Seq=1 Ack=1 Win=64240...
16:56:32.455274 50014 UDP: 51413 → 50014 Len=20
16:56:33.397194 58723 TCP: [TCP ZeroWindowProbe] 58723 → 6881 [ACK] ...
16:56:34.069305 58723 TCP: [TCP ZeroWindowProbe] 58723 → 6881 [ACK] ...
16:56:34.399400 ICMPv6: Echo (ping) request id=0x0001, seq=5, hop...
16:56:34.475096 ICMPv6: Echo (ping) reply id=0x0001, seq=5, hop li...
16:56:34.679679 6881 UDP: 6881 → 29367 Len=65
16:56:34.680287 UDP: 6881 → 8098 Len=104
16:56:34.680463 UDP: 6881 → 57311 Len=65

**SAURABH MUKHERJEE**

**001910501006**

**BCSE III**