

EXPERIMENT : 4
4A : STUDY OF WIRESHARK TOOL FOR PACKET SNIFFING
Date : 19.08.2024

AIM:

To study packet sniffing concepts using Wireshark Tool.

DESCRIPTION:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets. You can use Wireshark to inspect a suspicious program's network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

What we can do with Wireshark:

- Capture network traffic
- Decode packet protocols using dissectors
- Define filters – capture and display
- Watch smart statistics
- Analyze problems
- Interactively browse that traffic

Wireshark used for:

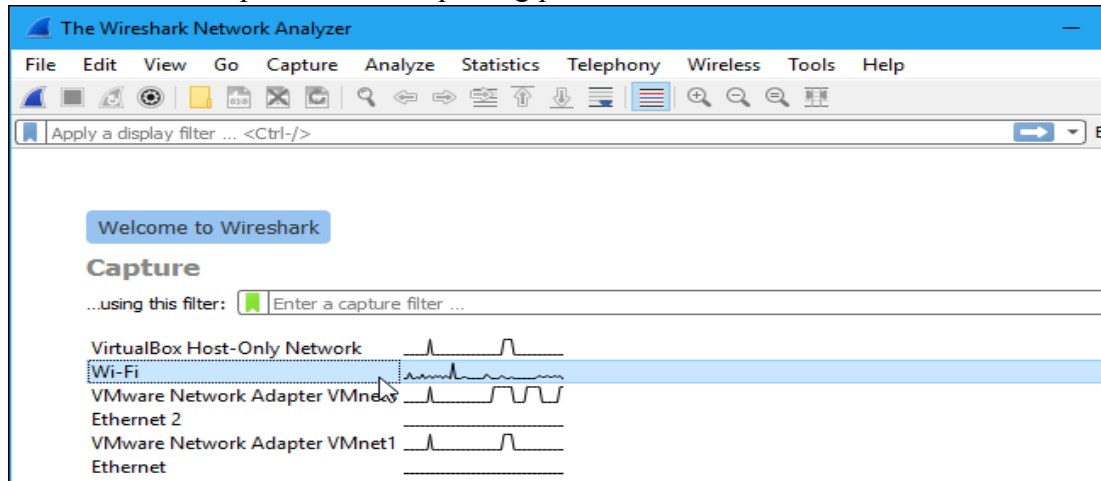
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

Getting Wireshark

Wireshark can be downloaded for Windows or macOS from [its official website](#). For Linux or another UNIX-like system, Wireshark will be found in its package repositories. For Ubuntu, Wireshark will be found in the Ubuntu Software Center.

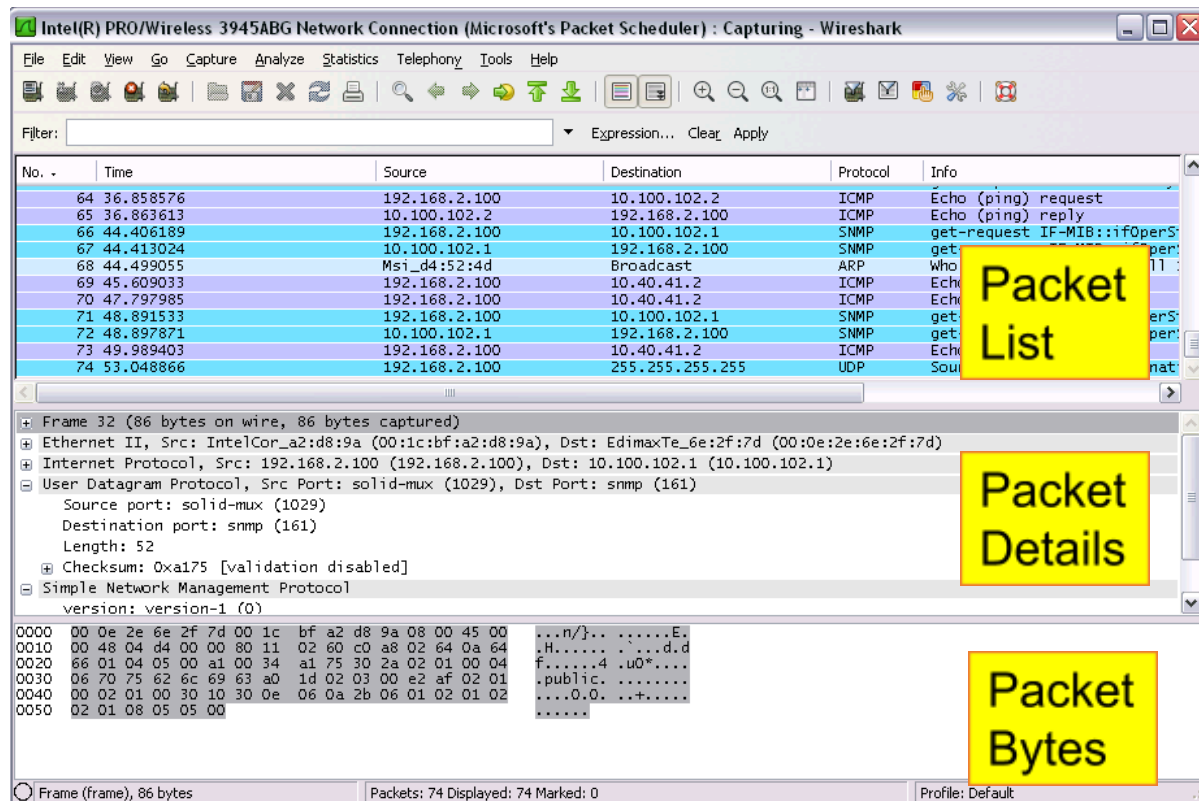
Capturing Packets

After downloading and installing Wireshark, launch it and double-click the name of a network interface under Capture to start capturing packets on that interface



As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

If you have promiscuous mode enabled—it's enabled by default—you'll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture > Options and verify the "Enable promiscuous mode on all interfaces" checkbox is activated at the bottom of this window.



Click the red “Stop” button near the top left corner of the window when you want to stop capturing traffic.

The “Packet List” Pane

The packet list pane displays all the packets in the current capture file. The “Packet List” pane Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the “Packet Details” and “Packet Bytes” panes.

The “Packet Details” Pane

The packet details pane shows the current packet (selected in the “Packet List” pane) in a more detailed form. This pane shows the protocols and protocol fields of the packet selected in the “Packet List” pane. The protocols and fields of the packet shown in a tree which can be expanded and collapsed.

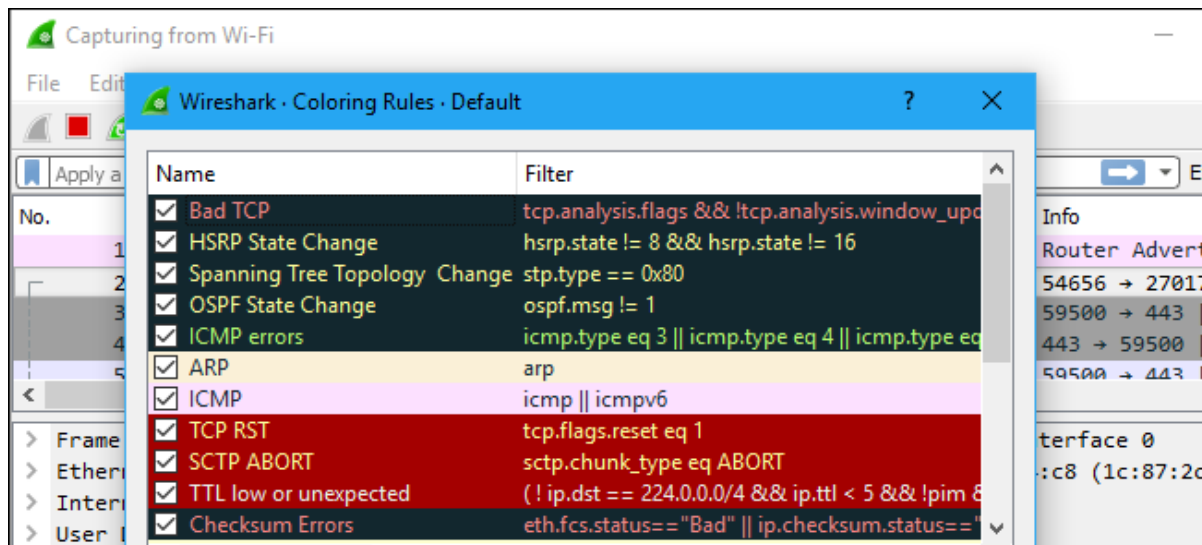
The “Packet Bytes” Pane

The packet bytes pane shows the data of the current packet (selected in the “Packet List” pane) in a hexdump style.

Color Coding

You’ll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order.

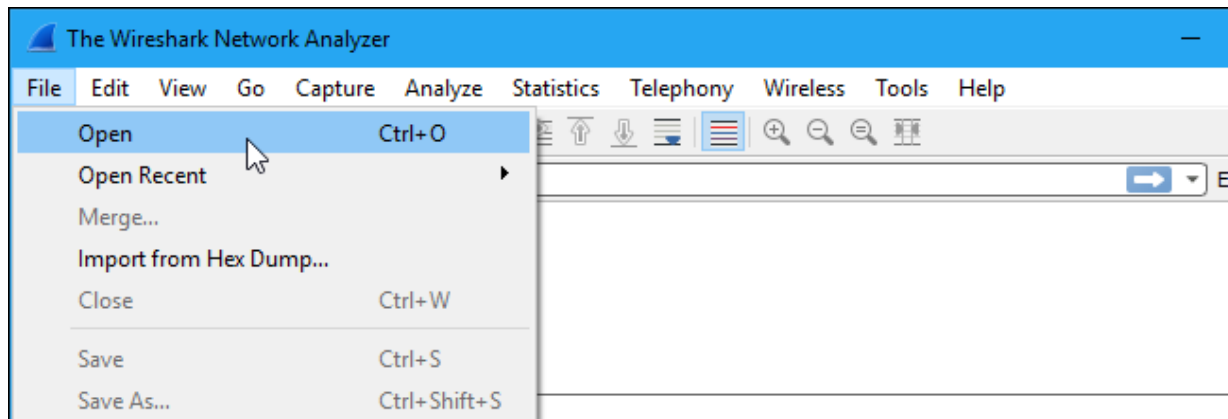
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



Sample Captures

If there’s nothing interesting on your own network to inspect, Wireshark’s wiki has you covered. The wiki contains a [page of sample capture files](#) that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.

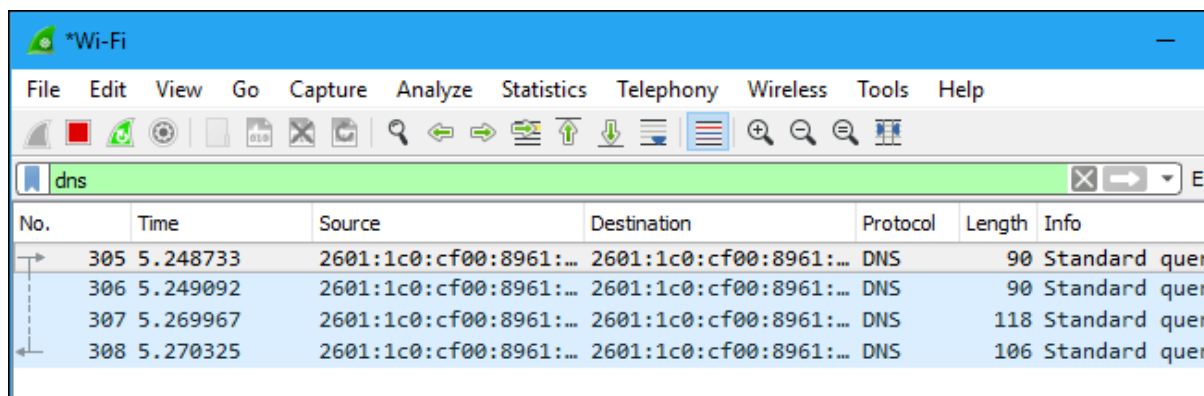
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Filtering Packets

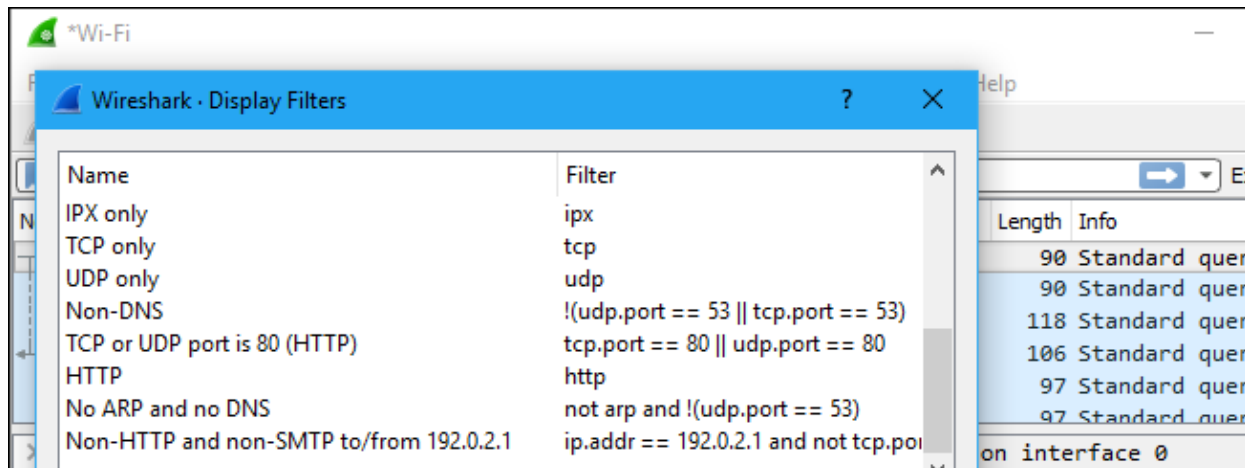
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



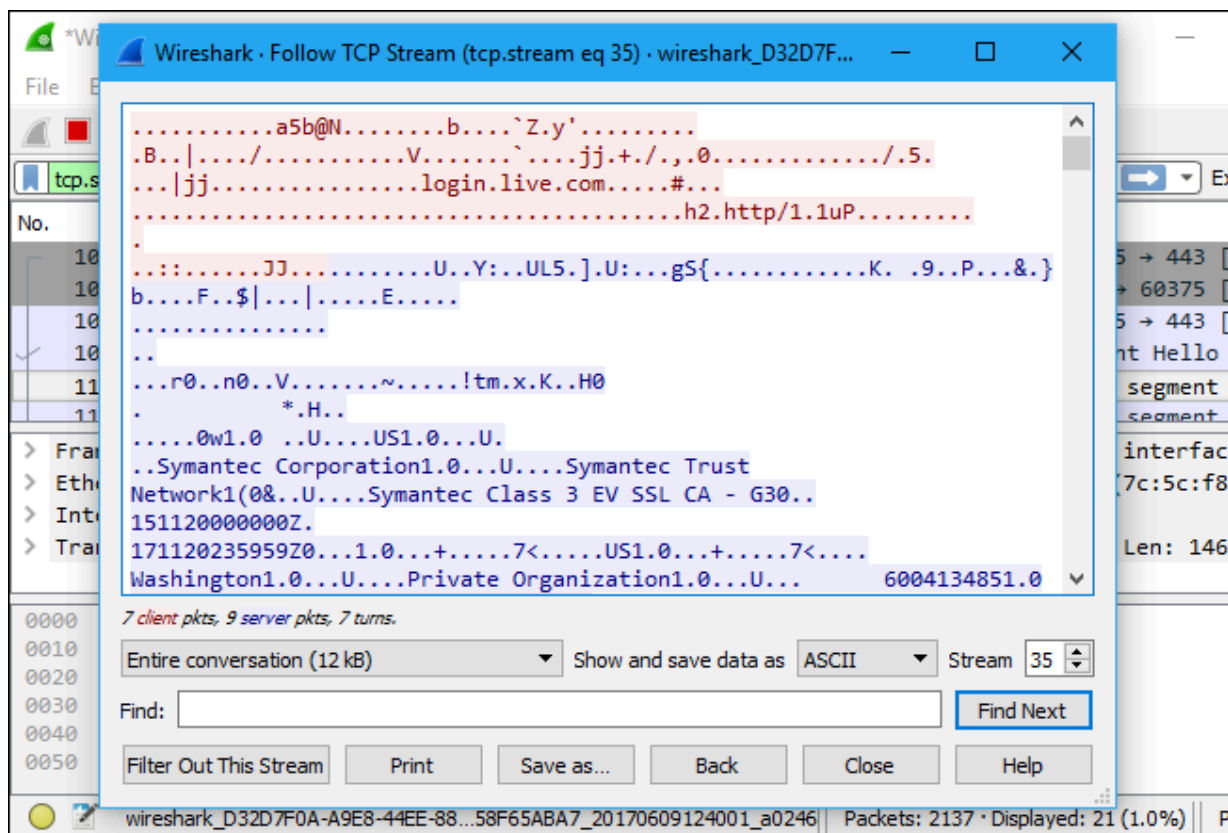
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

For more information on Wireshark's display filtering language, read the [Building display filter expressions](#) page in the official Wireshark documentation.

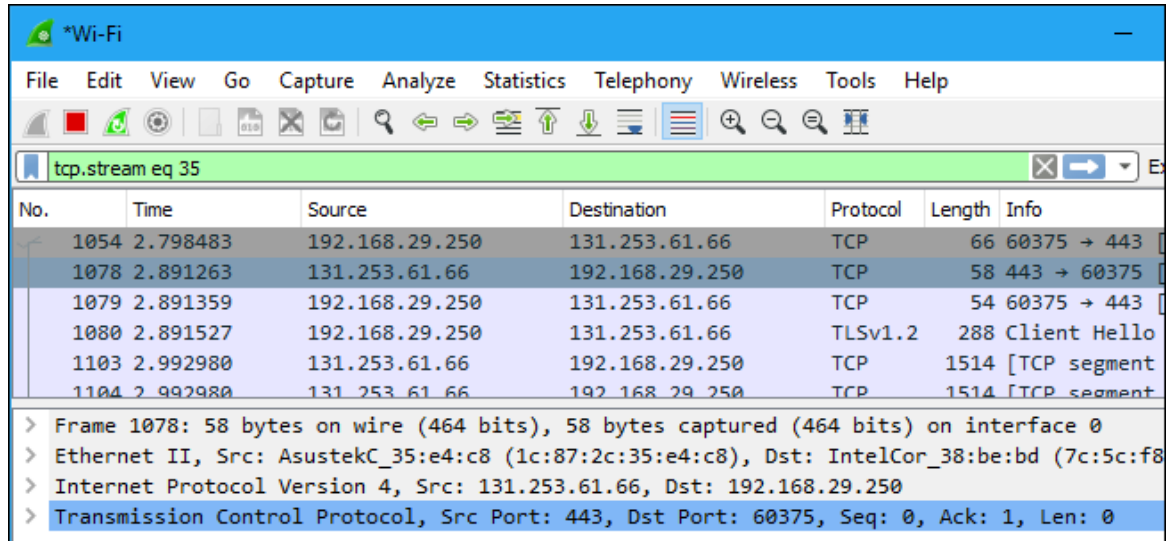


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.



Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

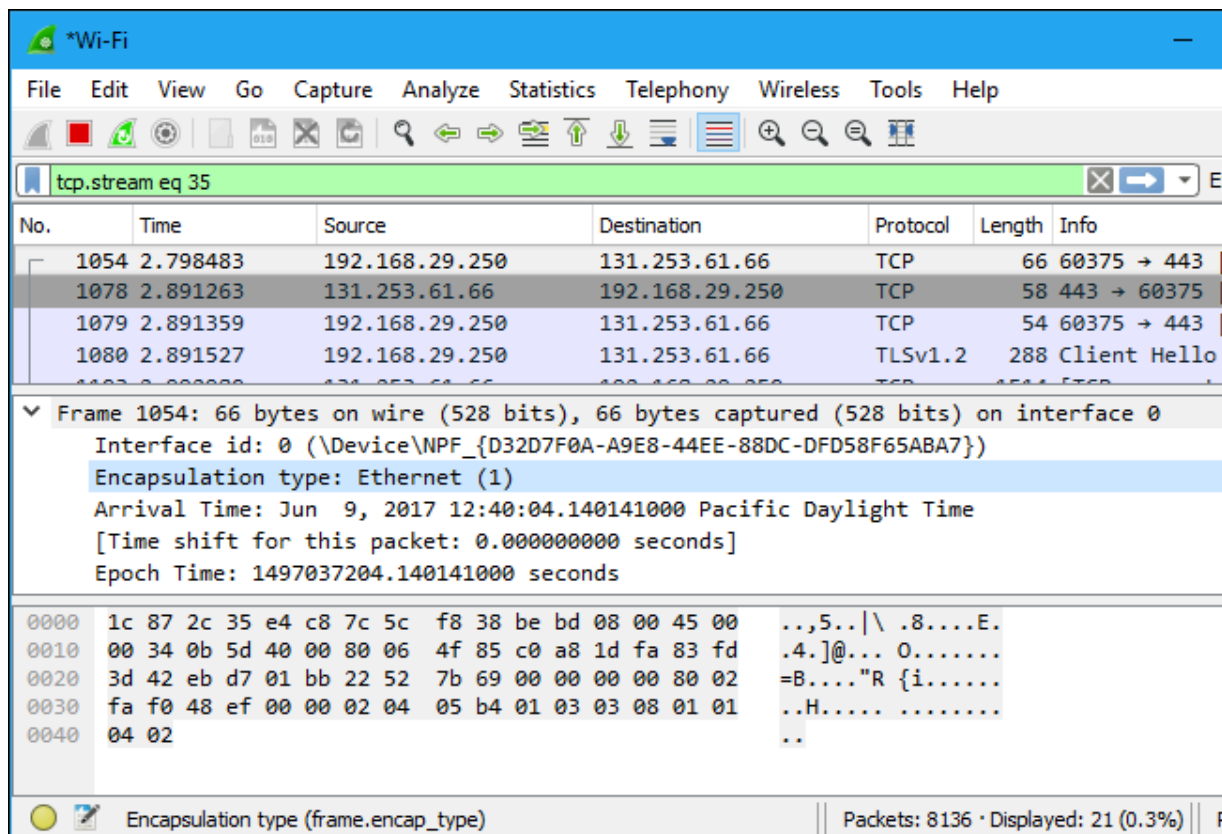


No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello
1103	2.992980	131.253.61.66	192.168.29.250	TCP	1514	[TCP segment
1104	2.992980	131.253.61.66	192.168.29.250	TCP	1514	[TCP segment

> Frame 1078: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
 > Ethernet II, Src: AsustekC_35:e4:c8 (1c:87:2c:35:e4:c8), Dst: IntelCor_38:be:bd (7c:5c:f8
 > Internet Protocol Version 4, Src: 131.253.61.66, Dst: 192.168.29.250
 > Transmission Control Protocol, Src Port: 443, Dst Port: 60375, Seq: 0, Ack: 1, Len: 0

Inspecting Packets

Click a packet to select it and you can dig down to view its details.



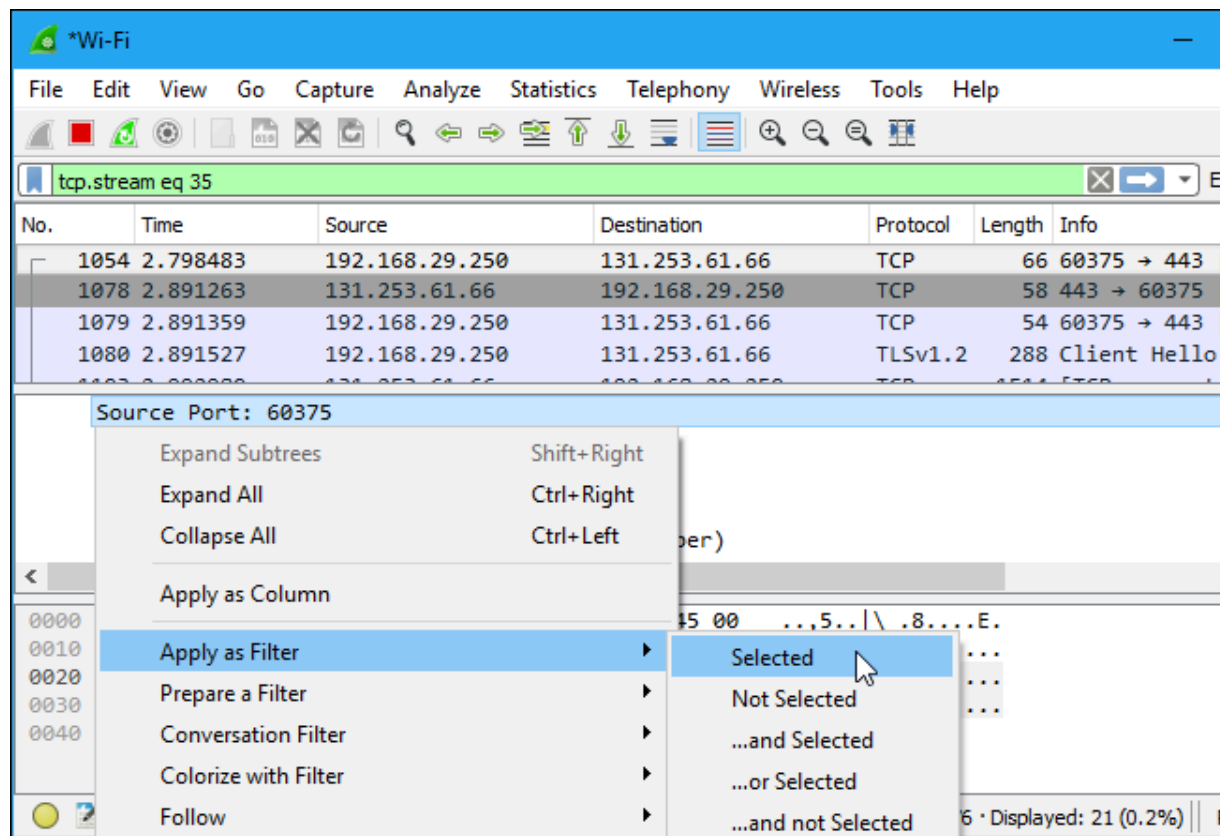
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

Frame 1054: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 Interface id: 0 (\Device\NPF_{D32D7F0A-A9E8-44EE-88DC-DFD58F65ABA7})
 Encapsulation type: Ethernet (1)
 Arrival Time: Jun 9, 2017 12:40:04.140141000 Pacific Daylight Time
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1497037204.140141000 seconds

0000	1c 87 2c 35 e4 c8 7c 5c f8 38 be bd 08 00 45 00	..,5.. \ .8....E.
0010	00 34 0b 5d 40 00 80 06 4f 85 c0 a8 1d fa 83 fd	.4.]@... O.....
0020	3d 42 eb d7 01 bb 22 52 7b 69 00 00 00 00 80 02	=B...."R {i.....
0030	fa f0 48 ef 00 00 02 04 05 b4 01 03 03 08 01 01	..H.....
0040	04 02	..

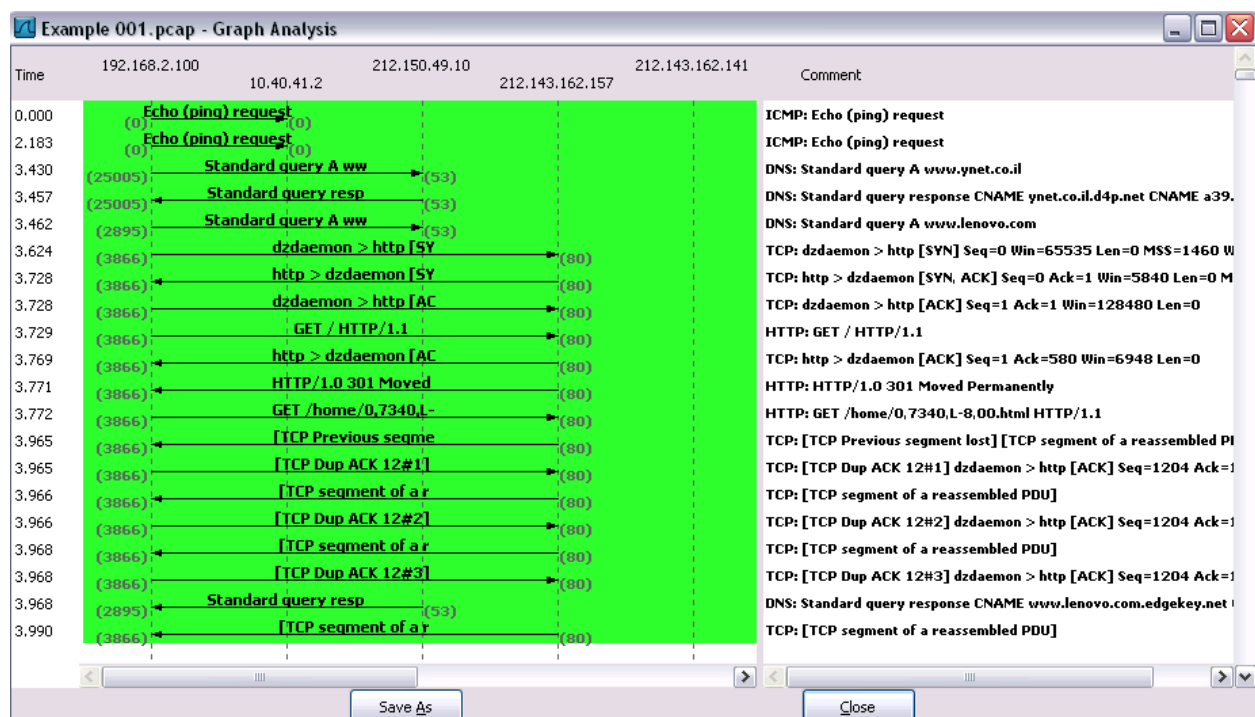
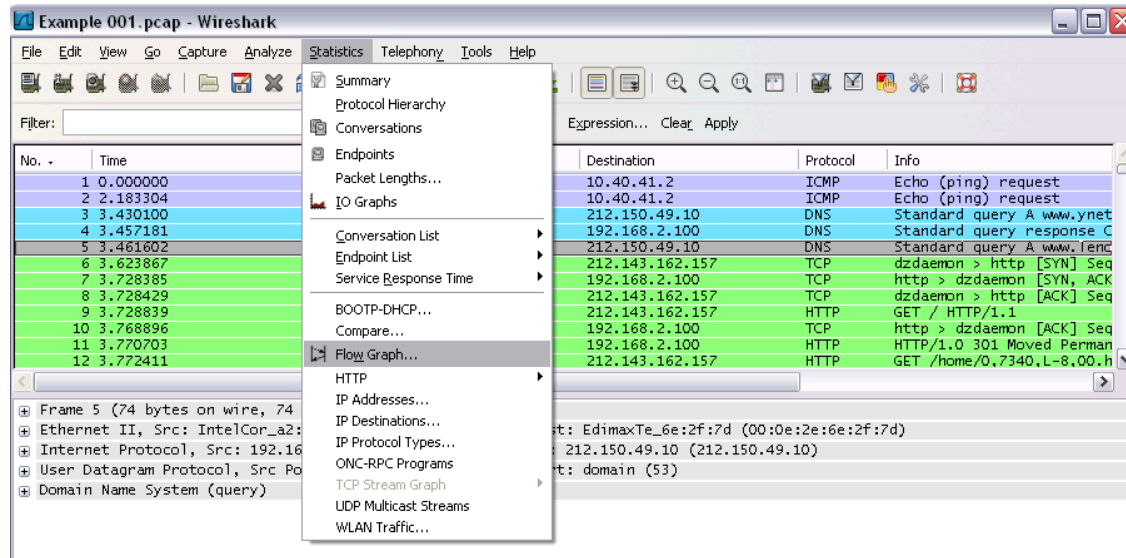
Encapsulation type (frame.encap_type) | Packets: 8136 · Displayed: 21 (0.3%) |

You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



Wireshark is an extremely powerful tool, and this tutorial is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

Flow Graph: Gives a better understanding of what we see.



Ex No: 4 B
Date : 20.08.2024

PACKET SNIFFING USING WIRESHARK

AIM:

To capture, save, filter and analyze network traffic on TCP / UDP / IP / HTTP / ARP /DHCP /ICMP /DNS using Wireshark Tool

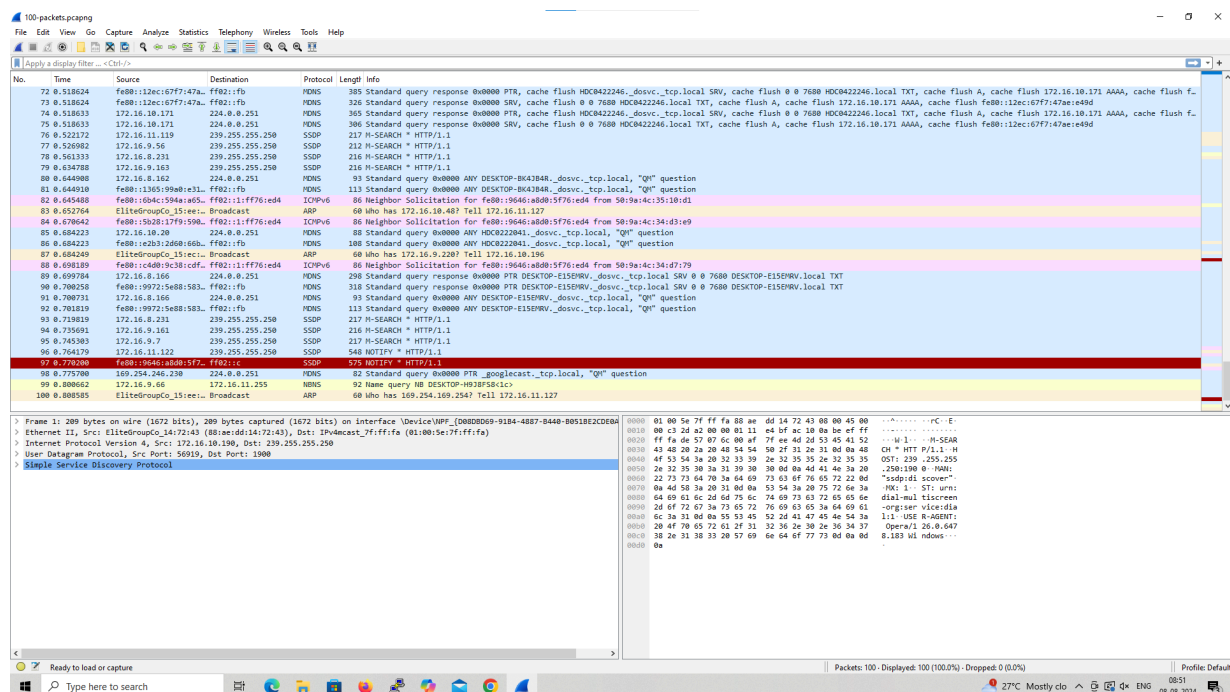
Exercises

1. Capture 100 packets from the Ethernet: IEEE 802.3 LAN Interface and save it.

Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Save the packets.

Output



2.Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph.

Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search TCP packets in search bar.
- ☐ To see flow graph click Statistics ☐ Flow graph.
- ☐ Save the packets.

Output:

The screenshot displays the Wireshark interface with a network capture. The packet list shows several TCP segments and application data. The packet details pane shows the structure of a TCP segment and its application data. The packet bytes pane shows the raw hex and ASCII data.

No.	Time	Source	Destination	Protocol	Length	Info
264	1.261938	172.16.8.190	142.250.205.234	TCP	55	64621 → 443 [ACK] Seq=1 Ack=1 Win=8287 Len=1 [TCP segment of a reassembled PDU]
265	1.263312	142.250.205.234	172.16.8.190	TCP	66	443 → 64621 [ACK] Seq=1 Ack=2 Win=586 Len=0 SLE=1 SRE=2
266	1.265839	172.16.8.190	142.250.193.110	TLSv1.2	3015	Application Data
267	1.265904	172.16.8.190	142.250.193.110	TLSv1.2	93	Application Data
268	1.265921	172.16.8.190	142.250.193.110	TLSv1.2	117	Application Data
269	1.267005	142.250.193.110	172.16.8.190	TCP	60	443 → 58033 [ACK] Seq=1 Ack=2062 Win=3628 Len=0
270	1.267005	142.250.193.110	172.16.8.190	TCP	60	443 → 58033 [ACK] Seq=1 Ack=3081 Win=3628 Len=0
271	1.267005	142.250.193.110	172.16.8.190	TCP	60	443 → 58033 [ACK] Seq=1 Ack=3084 Win=3628 Len=0
273	1.290701	142.250.193.110	172.16.8.190	TLSv1.2	93	Application Data
275	1.339267	172.16.8.190	142.250.193.110	TCP	54	58033 → 443 [ACK] Seq=3064 Ack=40 Win=8212 Len=0
286	1.505153	142.250.193.110	172.16.8.190	TLSv1.2	1321	Application Data, Application Data
287	1.506979	172.16.8.190	142.250.193.110	TLSv1.2	89	Application Data
288	1.507042	142.250.193.110	172.16.8.190	TLSv1.2	148	Application Data
289	1.507042	142.250.193.110	172.16.8.190	TLSv1.2	179	Application Data
290	1.507042	142.250.193.110	172.16.8.190	TLSv1.2	93	Application Data
291	1.507076	172.16.8.190	142.250.193.110	TCP	54	58033 → 443 [ACK] Seq=3099 Ack=1565 Win=8212 Len=0
292	1.507683	172.16.8.190	142.250.193.110	TLSv1.2	93	Application Data
293	1.508664	142.250.193.110	172.16.8.190	TCP	60	443 → 58033 [ACK] Seq=1565 Ack=3118 Win=3645 Len=0
312	1.865046	172.16.8.190	142.250.196.165	TCP	55	64627 → 443 [ACK] Seq=1 Ack=1 Win=8212 Len=1 [TCP segment of a reassembled PDU]
313	1.867269	142.250.196.165	172.16.8.190	TCP	66	443 → 64627 [ACK] Seq=1 Ack=2 Win=343 Len=0 SLE=1 SRE=2
321	1.944454	172.16.8.190	172.16.50.112	TCP	66	64654 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
355	6.257327	172.16.8.190	172.16.50.112	TCP	66	[TCP Retransmission] 64654 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
485	4.952642	172.16.8.190	172.16.50.112	TCP	66	[TCP Retransmission] 64654 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
555	6.257917	172.16.8.190	142.250.67.67	TLSv1.2	149	Application Data
556	6.257965	172.16.8.190	142.250.67.67	TLSv1.2	93	Application Data
557	6.259410	142.250.67.67	172.16.8.190	TCP	60	443 → 58041 [ACK] Seq=1 Ack=135 Win=383 Len=0
559	6.263521	142.250.67.67	172.16.8.190	TLSv1.2	93	Application Data
566	6.297878	142.250.67.67	172.16.8.190	TLSv1.2	231	Application Data, Application Data
567	6.297878	142.250.67.67	172.16.8.190	TLSv1.2	93	Application Data

Frame 264: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF{08B0B069-9184-4887-B440-80518E2CDE0A},
 > Ethernet II, Src: HP_3504b1 (7c:57:58:35:04:b1), Dst: Sophos_cf1be45 (7c:5a:1c:1f:be:45)
 > Internet Protocol Version 4, Src: 172.16.8.190, Dst: 142.250.205.234
 > Transmission Control Protocol, Src Port: 64621, Dst Port: 443, Seq: 1, Ack: 1, Len: 1

0000 7c 5a 1c 1f be 45 7c 57 58 35 04 b1 08 00 45 00 [Z...E]u X5....E:
 0010 00 29 15 90 40 00 00 06 00 00 ac 10 00 be 8e fa -> @ :
 0020 cd ea fc 6d 61 b0 86 36 83 e4 58 6e e7 70 50 18-m-@-Xn gp-
 0030 20 ef 11 cf 00 00 00 00
 0040

Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ws.cap protocol == UDP

No.	Time	Source	Destination	Protocol	Length	Info
17	8.238972	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
21	8.259992	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
37	8.437512	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
96	1.813759	172.16.10.175	239.255.255.250	UDP	698	53853 → 3782 Len=656
129	1.906667	fe80::1557:e72d:910...	ff02::c	UDP	718	53854 → 3782 Len=656
130	1.948246	172.16.9.74	172.16.11.255	UDP	186	60008 → 51007 Len=144
202	3.817074	172.16.10.175	239.255.255.250	UDP	698	53853 → 3782 Len=656
221	3.248613	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
223	3.254383	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
269	3.630859	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
289	3.908057	fe80::1557:e72d:910...	ff02::c	UDP	718	53854 → 3782 Len=656
1196	5.955879	172.16.9.74	172.16.11.255	UDP	186	60008 → 51007 Len=144
1215	6.246778	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1218	6.276472	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1231	6.642740	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
1641	9.252637	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1642	9.268289	fe80::5313:1f1c:90b...	ff02::c	UDP	718	56352 → 3782 Len=656
1643	9.268289	172.16.10.211	239.255.255.250	UDP	698	56351 → 3782 Len=656
1649	9.281294	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1654	9.415787	172.16.10.211	239.255.255.250	UDP	698	56351 → 3782 Len=656
1655	9.501892	fe80::5313:1f1c:90b...	ff02::c	UDP	718	56352 → 3782 Len=656
1664	9.641430	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
1666	9.723266	172.16.10.211	239.255.255.250	UDP	698	56351 → 3782 Len=656
1685	9.963382	172.16.9.74	172.16.11.255	UDP	186	60008 → 51007 Len=144
1686	9.974373	fe80::5313:1f1c:90b...	ff02::c	UDP	718	56352 → 3782 Len=656
1691	10.004004	172.16.9.50	172.16.11.255	UDP	86	57621 → 57621 Len=44
1706	10.315798	172.16.10.211	239.255.255.250	UDP	698	56351 → 3782 Len=656
1749	10.904999	fe80::5313:1f1c:90b...	ff02::c	UDP	718	56352 → 3782 Len=656
1800	11.514242	172.16.10.211	239.255.255.250	UDP	698	56351 → 3782 Len=656
1861	12.243224	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1864	12.256156	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1892	12.648812	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
1911	12.762779	fe80::5313:1f1c:90b...	ff02::c	UDP	718	56352 → 3782 Len=656

> Frame 1902: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface \Device\NPF_{0808D069-91B4-4287-B440-B051B2CDE0A} id 0

> Ethernet II, Src: ASUSTekComPU3 (81:9f:b8:c3:65:05), Dst: IPMulticast_4d:4d:4d (01:00:5e:4d:4d:4d)

> Internet Protocol Version 4, Src: 172.16.9.205, Dst: 224.77.77.77

> User Datagram Protocol, Src Port: 12177, Dst Port: 12177

> Data (106 bytes)

```
0000  01 00 5e 4d 4d 4d 00 00 00 00 00 00 00 00 00 00  --> Ethernet II, Src: ASUSTekComPU3 (81:9f:b8:c3:65:05), Dst: IPMulticast_4d:4d:4d (01:00:5e:4d:4d:4d)
0010  00 06 9b 69 00 00 00 11 3a 86 ac 10 09 cd e0 4d  --> Internet Protocol Version 4, Src: 172.16.9.205, Dst: 224.77.77.77
0020  4d 4d 2f 91 2f 91 00 00 72 33 28 3c 41 53 55 53 5f  --> User Datagram Protocol, Src Port: 12177, Dst Port: 12177
0030  41 52 6d 4f 53 52 59 5f 43 52 41 54 43 3e 3c 4c  --> Data (106 bytes)
0040  41 4e 20 50 6f 72 74 3d 22 31 32 31 37 37 22 20  -->
0050  43 75 73 09 4a 3d 22 31 39 46 44 41 45 33 38 2d  -->
0060  30 37 31 39 2d 34 46 35 46 2d 41 41 45 32 20 37  -->
0070  35 41 42 45 34 41 38 36 38 45 31 22 28 2f 3e 3c  -->
0080  2f 41 53 55 53 5f 41 52 4d 4f 53 52 59 5f 43 52  -->
0090  41 54 45 3e  -->
```

Wireshark - Packet 1196 - Ethernet

> Frame 1196: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface \Device\NPF_{0808D069-91B4-4287-B440-B051B2CDE0A}, id 0

> Ethernet II, Src: ASIXElectron_e2ee:ab (20:7b:d2:e2:ee:ab), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4, Src: 172.16.9.74, Dst: 172.16.11.255

> User Datagram Protocol, Src Port: 60008, Dst Port: 51007

> Data (144 bytes)

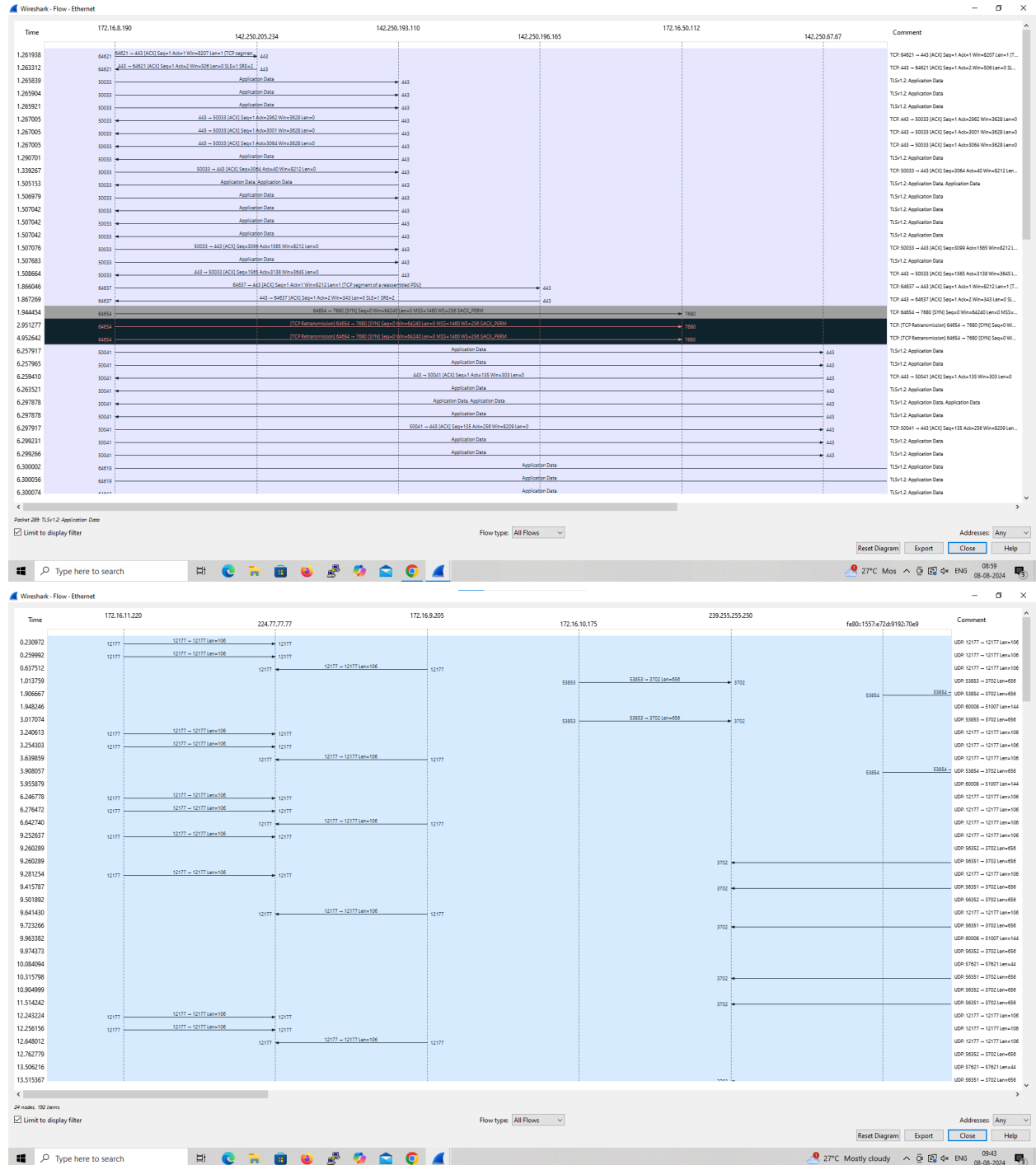
```
0000  ff ff ff ff ff ff 20 7b d2 e2 ee ab 08 00 45 00  --> Ethernet II, Src: ASIXElectron_e2ee:ab (20:7b:d2:e2:ee:ab), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
0010  00 ac 58 2c 00 00 00 74 ab ac 10 09 4a ac 10  --> Internet Protocol Version 4, Src: 172.16.9.74, Dst: 172.16.11.255
0020  0b ff ea 68 c7 3f 00 98 f3 cc 4a b5 7f e5 21 43  --> User Datagram Protocol, Src Port: 60008, Dst Port: 51007
0030  bd f3 2d 9e da ff f9 b6 b3 37 0a 1b c3 30 e2 91  --> Data (144 bytes)
0040  86 80 24 65 99 2c 29 6c 00 4b 1b 19 83 16 c2 2b  -->
0050  22 c2 a1 a6 f6 32 42 e0 1a cf 22 da 07 cc 60 75  -->
0060  53 7d 25 a5 3e aa 85 7f db 31 d7 a3 2e cd 60 9b  -->
0070  3d 1f 12 0c 73 87 30 25 0c ca f9 0d 6c 84 92 fc  -->
0080  60 19 e2 ba be aa 37 78 7f d2 02 c9 01 4e c3 80  -->
0090  9e 01 96 f2 ff a1 85 9b 9c 67 17 de c2 28 ed 95  -->
00a0  29 15 41 cb 99 3e 1a 4f 68 12 70 0f 43 d5 b0 fb  -->
00b0  46 02 78 94 8e 9c d6 bc ba 66  -->
```

No: 1196 Time: 5.955879 Source: 172.16.9.74 Destination: 172.16.11.255 Protocol: UDP Length: 106 Info: 60008 → 51007 Len=144

☒ Show packet bytes

Close Help

Flow Graph output



3.Create a Filter to display only ARP packets and inspect the packets.

Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search ARP packets in search bar.
- ☐ Save the packets.

Output

The screenshot shows the Wireshark interface with a packet capture filter set to 'arp'. The packet list displays several ARP packets. The selected packet (No. 6) is an ARP Announcement from HP_39:53:a6 to Broadcast. The packet details pane shows the Ethernet II header and the Address Resolution Protocol (ARP) section, which includes the Sender Hardware Address (7c:57:58:39:53:a6) and the Destination Broadcast address (ff:ff:ff:ff:ff:ff).

No.	Time	Source	Destination	Protocol	Length	Info
5	0.148237	Dell_34:d4:98	Broadcast	ARP	60	who has 172.16.9.163? Tell 172.16.11.47
6	0.165888	HP_39:53:a6	Broadcast	ARP	60	ARP Announcement for 172.16.9.216
11	0.272167	MicroStarINT_c5:cb::	Broadcast	ARP	60	who has 172.16.11.121? Tell 172.16.10.2
20	0.351318	EliteGroupCo_15:e7::	Broadcast	ARP	60	who has 169.254.169.254? Tell 172.16.10.194
25	0.485853	Dell_34:d7:f5	Broadcast	ARP	60	who has 172.16.10.69? Tell 172.16.11.16
39	0.758628	Ba0b:af:ad:48:ad	Broadcast	ARP	60	who has 172.16.8.1? Tell 172.16.11.258
40	0.776793	AzureWaveTec_9f:0c::	Broadcast	ARP	60	who has 172.16.11.114? (ARP Probe)
46	0.797990	Dell_34:d6:cb	Broadcast	ARP	60	who has 172.16.8.227? Tell 172.16.9.79
68	1.103962	MicroStarInt_ad:3c::	Broadcast	ARP	60	who has 172.16.10.123? Tell 172.16.8.13
71	1.139723	EliteGroupCo_15:ed::	Broadcast	ARP	60	who has 169.254.169.254? Tell 172.16.9.202
82	1.274244	MicroStarINT_c5:cb::	Broadcast	ARP	60	who has 172.16.11.121? Tell 172.16.10.2
83	1.274244	MicroStarInt_ad:3c::	Broadcast	ARP	60	who has 172.16.9.69? Tell 172.16.8.23
88	1.323668	AGOSTekCOMP_84:c8::	Broadcast	ARP	60	who has 172.16.9.173? Tell 172.16.11.220
91	1.354368	EliteGroupCo_15:e7::	Broadcast	ARP	60	who has 169.254.169.254? Tell 172.16.10.194
92	1.385753	EliteGroupCo_15:eb::	Broadcast	ARP	60	who has 169.254.169.254? Tell 172.16.10.191

Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{08080B69-9184-4887-B440-B051B2CDE0A}, 1
 Ethernet II, Src: HP_39:53:a6 (7c:57:58:39:53:a6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (ARP Announcement)

Packet details (Frame 6):

```

0000  ff ff ff ff ff ff 7c 57 58 39 53 a6 06 00 01 .....[M] XPS:.....
0010  00 00 00 04 00 01 7c 57 58 39 53 a6 ac 10 00 00 .....[M] XPS:.....
0020  00 00 00 00 00 ac 10 00 00 00 00 00 00 00 00 .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
  
```

Summary: Packets: 100 - Displayed: 15 (15.0%) - Dropped: 0 (0.0%)

4.Create a Filter to display only DNS packets and provide the flow graph.

Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search DNS packets in search bar.
- ☐ To see flow graph click Statistics ☐ Flow graph.
- ☐ Save the packets.

Output

The screenshot shows the Wireshark network protocol analyzer interface. The top pane displays a list of captured packets, with a filter 'dns' applied. The middle pane shows the details of the selected packet (No. 1521), including the Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (query) fields. The bottom pane shows the packet bytes and their hexadecimal representation.

No.	Time	Source	Destination	Protocol	Length	Info
1521	15.175484	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x1acd A edge.microsoft.com
1522	15.175597	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x60b0 HTTPS edge.microsoft.com
1524	15.177324	172.16.8.1	172.16.8.190	DNS	181	Standard query response 0x1acd A edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net A 13.107.21.239 A 204.79.197.239
1525	15.177324	172.16.8.1	172.16.8.190	DNS	149	Standard query response 0x60b0 HTTPS edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net
1590	15.170966	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x32e9 A edge.microsoft.com
1591	15.171175	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x985d HTTPS edge.microsoft.com
1592	15.171948	172.16.8.1	172.16.8.190	DNS	181	Standard query response 0x32e9 A edge-microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net A 13.107.21.239 A 204.79.197.239
1593	15.171948	172.16.8.1	172.16.8.190	DNS	149	Standard query response 0x985d HTTPS edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net
1686	15.729996	172.16.8.190	172.16.8.1	DNS	92	Standard query 0x02ef A edge-mobile-static.azureedge.net
1697	15.730154	172.16.8.190	172.16.8.1	DNS	92	Standard query 0x0453 HTTPS edge-mobile-static.azureedge.net
1688	15.731830	172.16.8.1	172.16.8.190	DNS	245	Standard query response 0x0453 HTTPS edge-mobile-static.azureedge.net CNAME edge-mobile-static.afd.azureedge.net CNAME azureedge-t-prod.trafficmanager.net CNAME shed.dual-low-s-part-0030.
1689	15.731830	172.16.8.1	172.16.8.190	DNS	261	Standard query response 0x02ef A edge-mobile-static.azureedge.net CNAME edge-mobile-static.afd.azureedge.net CNAME azureedge-t-prod.trafficmanager.net CNAME shed.dual-low-s-part-0030.
1629	15.730530	172.16.8.190	172.16.8.1	DNS	72	Standard query 0x09d4 A www.bing.com
1630	15.730699	172.16.8.190	172.16.8.1	DNS	72	Standard query 0x3ae7 HTTPS www.bing.com
1631	15.781570	172.16.8.1	172.16.8.190	DNS	193	Standard query response 0x3ae7 HTTPS www.bing.com CNAME www.bing.com.trafficmanager.net CNAME www.bing.com.edgekey.net CNAME e86303.dscx.akamaiedge.net
1632	15.781570	172.16.8.1	172.16.8.190	DNS	337	Standard query response 0x09d4 A www.bing.com CNAME www-bing.com.trafficmanager.net CNAME www.bing.com.edgekey.net CNAME e86303.dscx.akamaiedge.net A 23.223.244.123 A 23.223.244.131 A
1698	16.046668	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x05ab A edge.microsoft.com
1699	16.046740	172.16.8.190	172.16.8.1	DNS	78	Standard query 0x0c68 HTTPS edge.microsoft.com
1700	16.047677	172.16.8.1	172.16.8.190	DNS	149	Standard query response 0x0c68 HTTPS edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net
1701	16.047677	172.16.8.1	172.16.8.190	DNS	181	Standard query response 0x05ab A edge-microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net A 13.107.21.239 A 204.79.197.239
1932	17.726760	172.16.8.190	172.16.8.1	DNS	71	Standard query 0x6ef1 A ntp.msn.com
1933	17.726909	172.16.8.190	172.16.8.1	DNS	71	Standard query 0x0512 HTTPS ntp.msn.com
1935	17.727890	172.16.8.1	172.16.8.190	DNS	130	Standard query response 0x0512 HTTPS ntp.msn.com CNAME www-msn-com-a-0003.a-msedge.net CNAME a-0003.a-msedge.net
1936	17.727890	172.16.8.1	172.16.8.190	DNS	146	Standard query response 0x6ef1 A ntp.msn.com CNAME www-msn-com-a-0003.a-msedge.net CNAME a-0003.a-msedge.net A 204.79.197.203
1952	17.758437	172.16.8.190	172.16.8.1	DNS	71	Standard query 0x6f1b A ntp.msn.com
1953	17.759553	172.16.8.1	172.16.8.190	DNS	146	Standard query response 0x6f1b A ntp.msn.com CNAME www-msn-com-a-0003.a-msedge.net CNAME a-0003.a-msedge.net A 204.79.197.203
2045	17.975432	172.16.8.190	172.16.8.1	DNS	87	Standard query 0x050f A img-s-msn-com.akamaized.net
2046	17.975632	172.16.8.190	172.16.8.1	DNS	87	Standard query 0x087f HTTPS img-s-msn-com.akamaized.net
2047	17.975862	172.16.8.190	172.16.8.1	DNS	84	Standard query 0x7e08 A sb.scorecardresearch.com
2048	17.975922	172.16.8.190	172.16.8.1	DNS	84	Standard query 0x0ced HTTPS sb.scorecardresearch.com
2049	17.976081	172.16.8.190	172.16.8.1	DNS	71	Standard query 0x4d70 A th.bing.com
2050	17.976140	172.16.8.190	172.16.8.1	DNS	71	Standard query 0x7bde HTTPS th.bing.com
2051	17.976471	172.16.8.1	172.16.8.190	DNS	130	Standard query response 0x087f HTTPS img-s-msn-com.akamaized.net CNAME a1834.dscg2.akamai.net
2052	17.976471	172.16.8.1	172.16.8.190	DNS	148	Standard query response 0x7e08 A sb.scorecardresearch.com A 18.161.216.181 A 18.161.216.83 A 18.161.216.23 A 18.161.216.37
2053	17.976471	172.16.8.1	172.16.8.190	DNS	152	Standard query response 0x050f A img-s-msn-com.akamaized.net CNAME a1834.dscg2.akamai.net A 23.215.215.104 A 23.215.215.187

Frame 1521: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{0080BD09-9184-4887-B440-8B51B2CDE0A}

Ethernet II, Src: HP_25:54:00:12:7c:57:58:30:04:01, Dst: Sophos_cf8ae45 (7c:5a:1c:f1:be:45)

Internet Protocol Version 4, Src: 172.16.8.190, Dst: 172.16.8.1

User Datagram Protocol, Src Port: 51966, Dst Port: 53

Domain Name System (query)

0000 7c 5a 1c f1 be 45 7c 57 58 30 04 01 00 00 45 00 [Z...E]u X5...E.
 0010 00 40 a3 63 00 00 00 11 00 00 ac 10 00 be ac 10 @.....
 0020 00 01 ca fe 00 35 00 2c 69 1d 1a cd 01 00 00 01S,i.....
 0030 00 00 00 00 00 00 04 65 64 67 65 09 dd 69 63 72e dge micr
 0040 6f 73 6f 66 74 03 63 6f 6d 00 01 00 01o soft co m.....

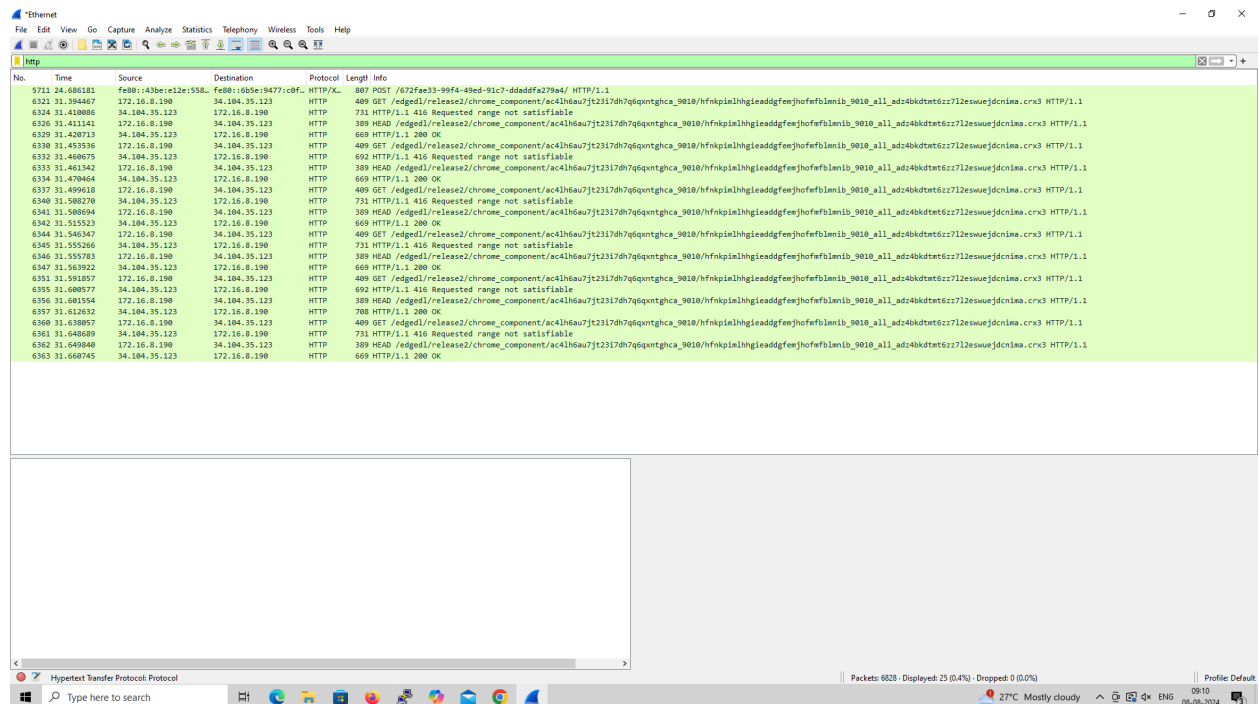
[illegible]

5.Create a Filter to display only HTTP packets and inspect the packets

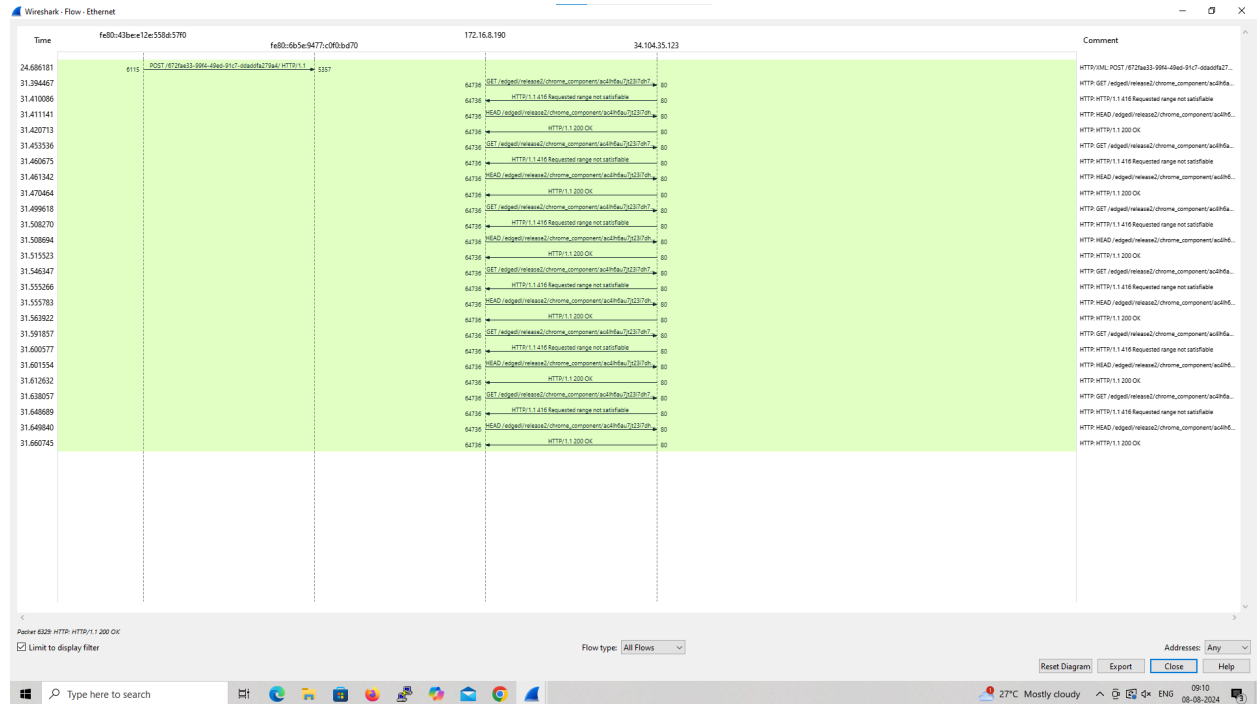
Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search HTTP packets in the search bar.
- ☐ Save the packets.

Output



Flow Graph output

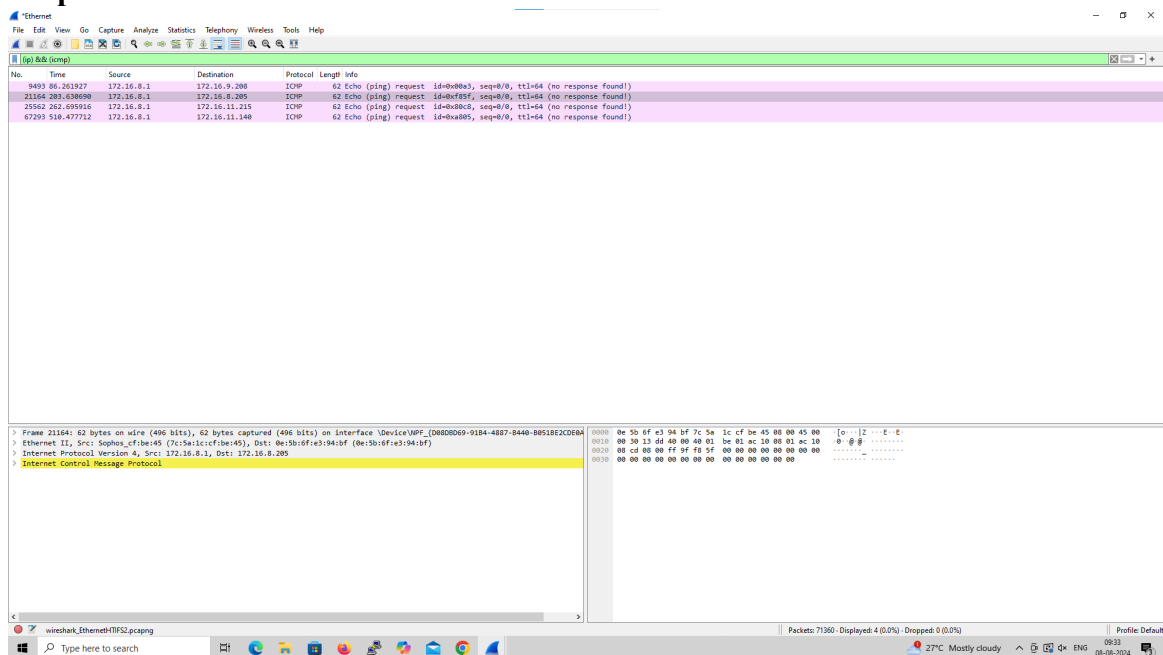


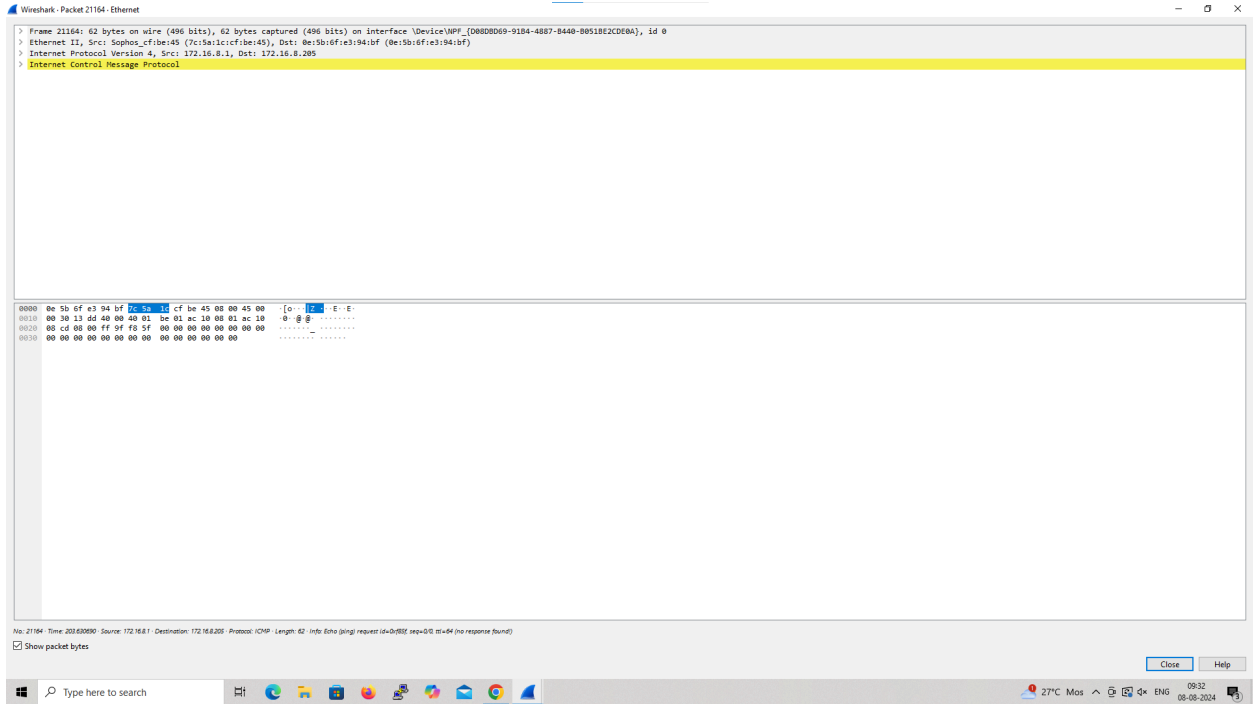
6.Create a Filter to display only IP/ICMP packets and inspect the packets.

Procedure

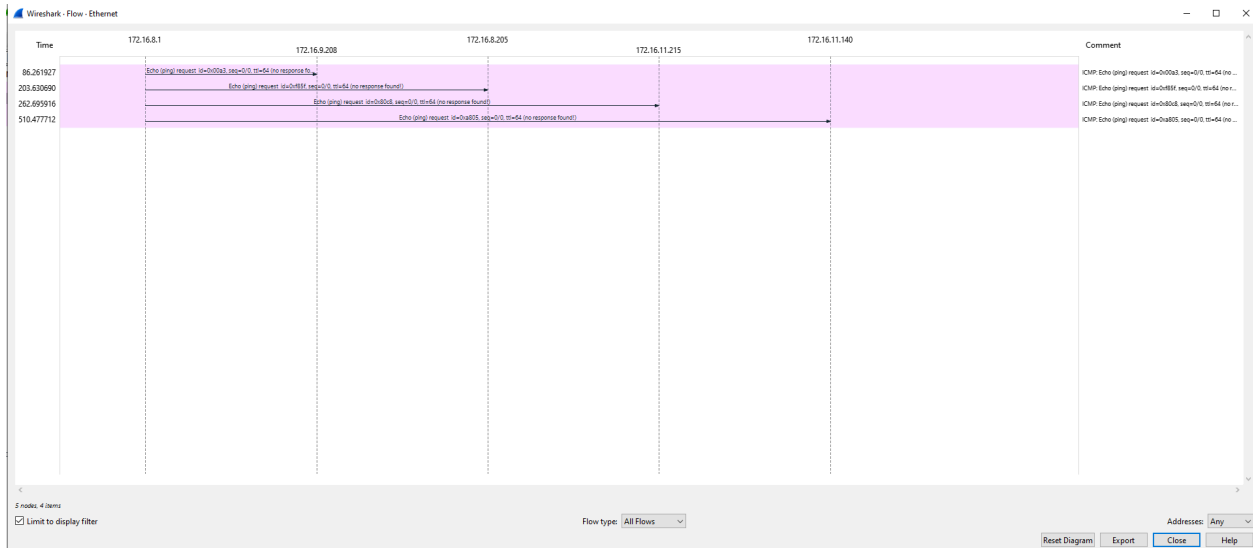
- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search ICMP/IP packets in search bar.
- ☐ Save the packets

Output





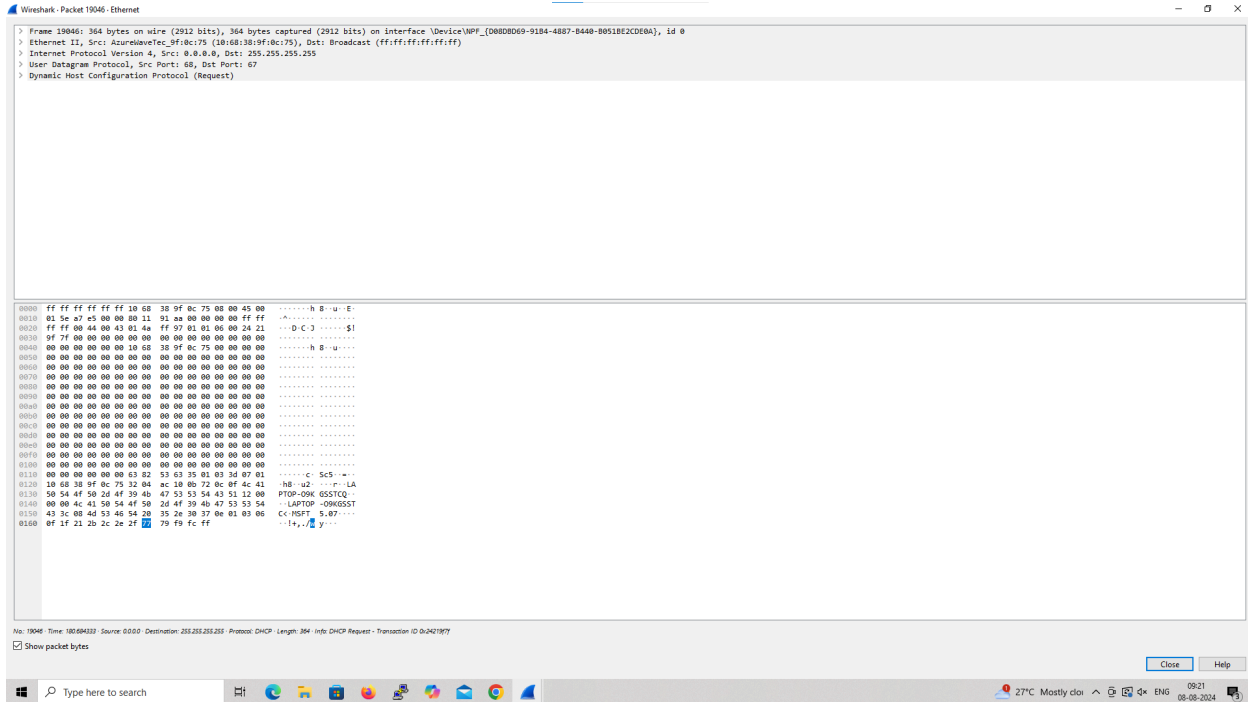
Flow Graph output



Procedure

- ☐ Select Local Area Connection in Wireshark.
- ☐ Go to capture ☐ option
- ☐ Select stop capture automatically after 100 packets.
- ☐ Then click Start capture.
- ☐ Search DHCP packets in search bar.
- ☐ Save the packets

[illegible]



RESULT :

Thus , we analyzed the network traffic using Wireshark