

## **Ex No:4a      STUDY OF WIRESHARK TOOL FOR PACKET SNIFFING**

### **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:**

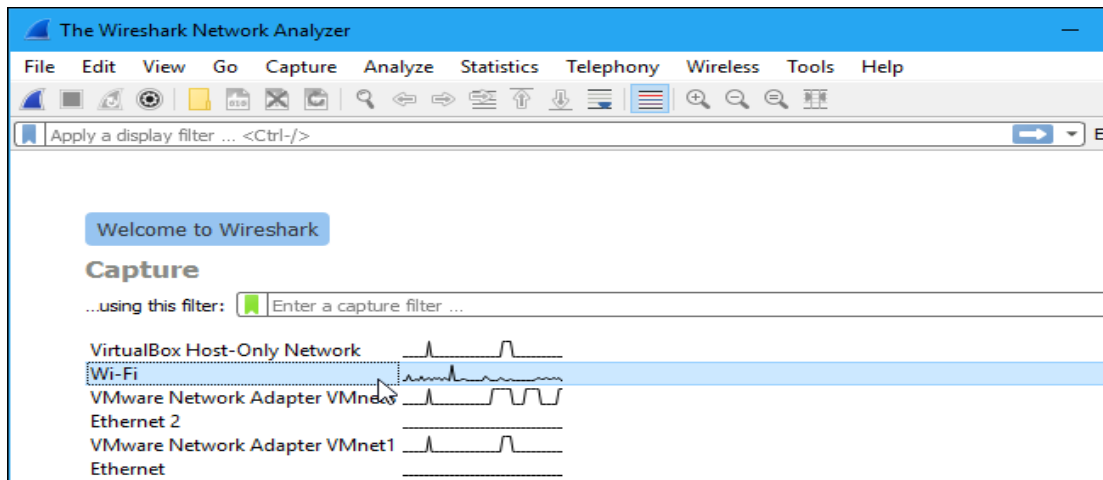
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

### **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.

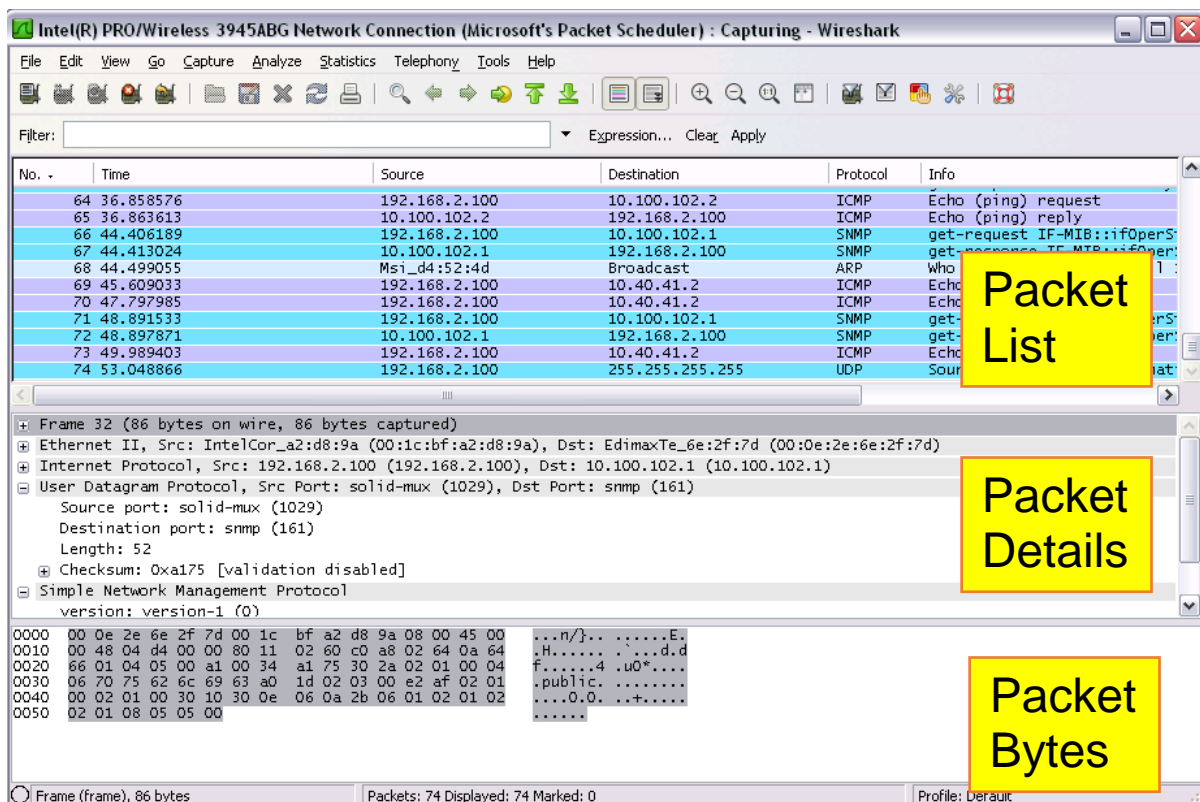
### **Capturing Packets**

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.

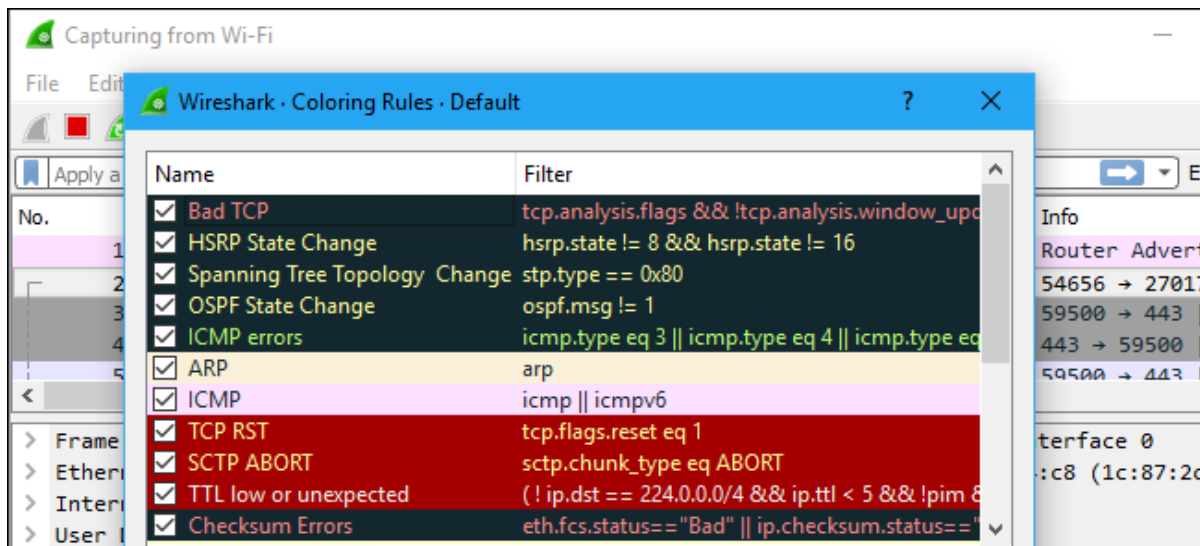
### **The “Packet Bytes” Pane**

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.

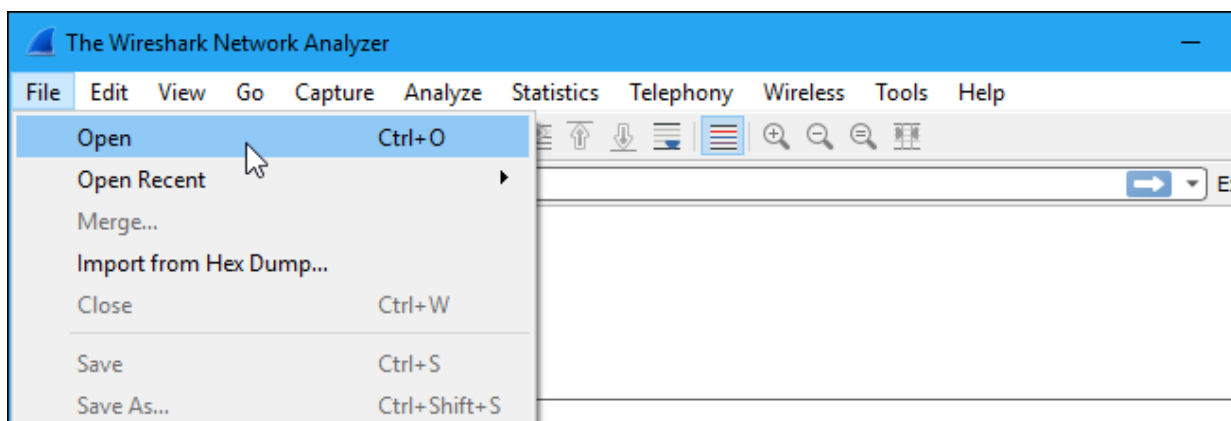
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



## 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.

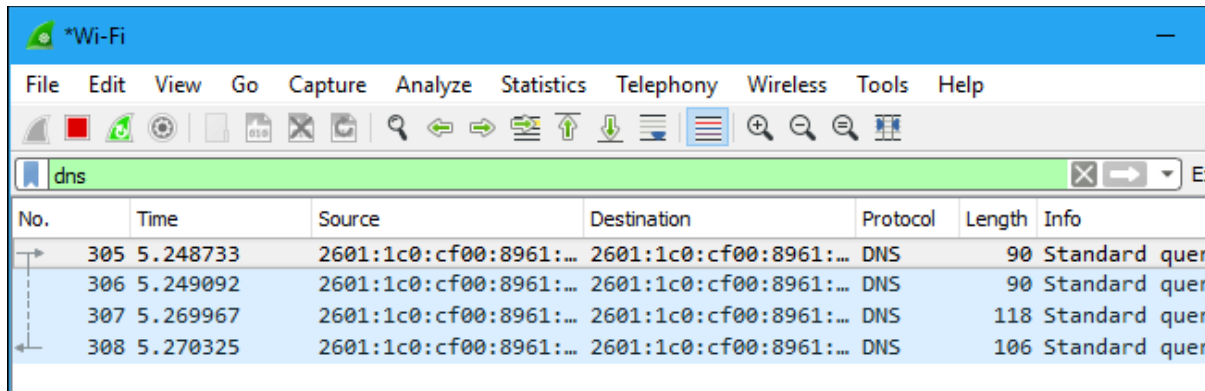


## Filtering Packets

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

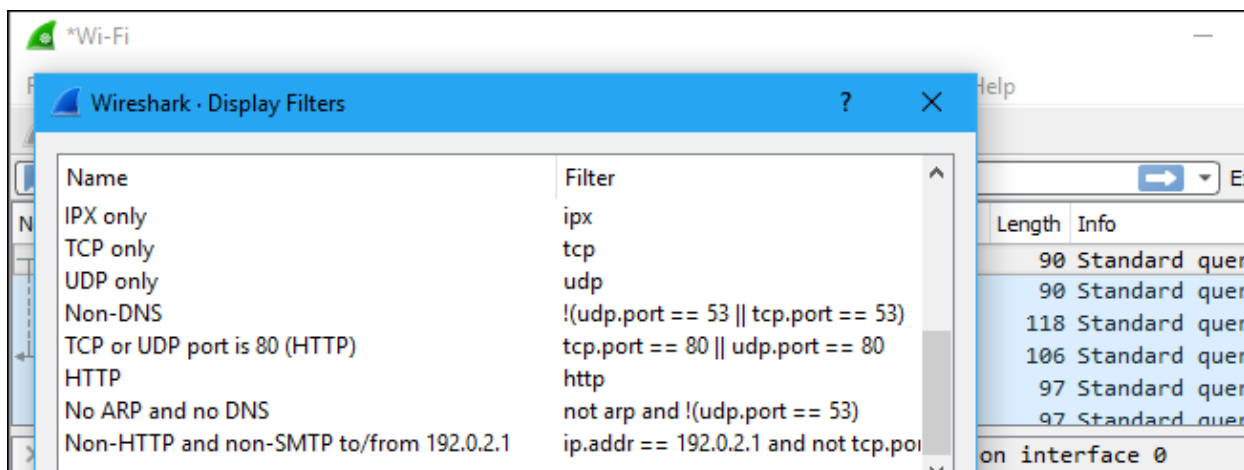
the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

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.



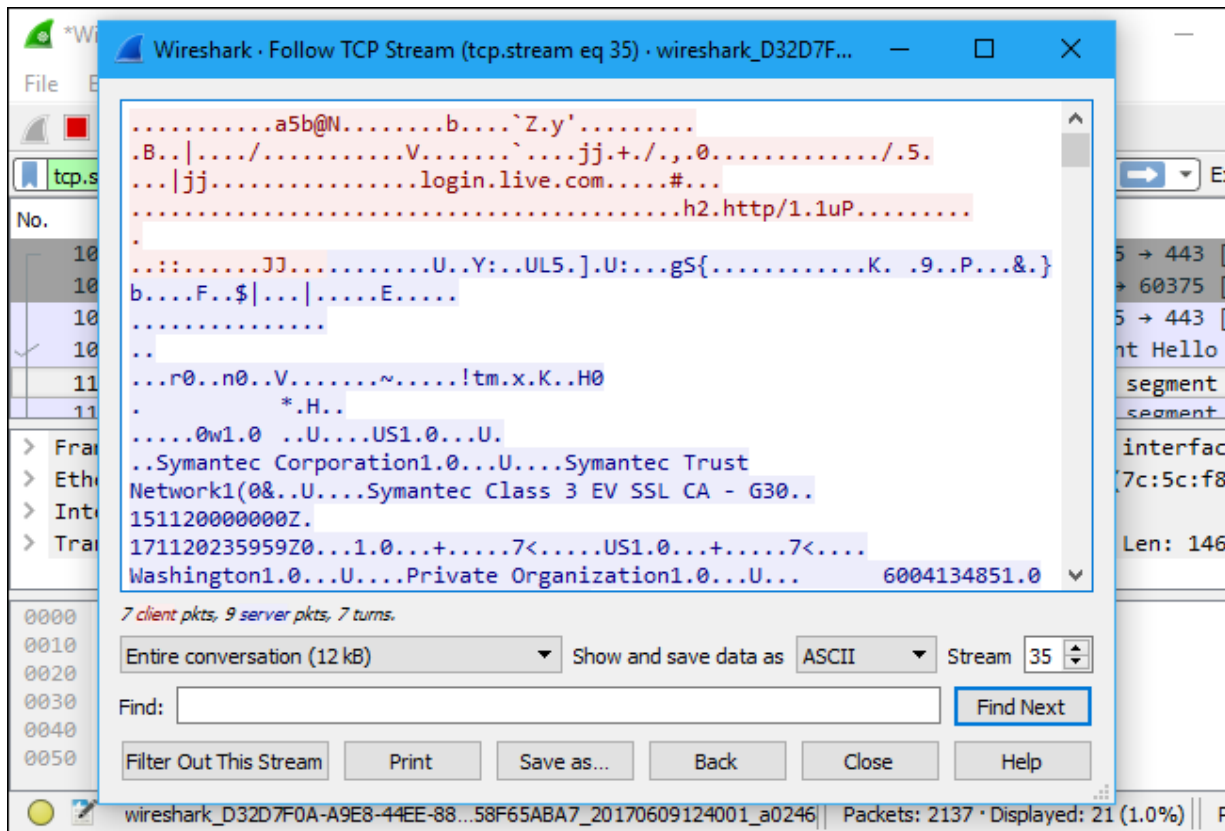
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

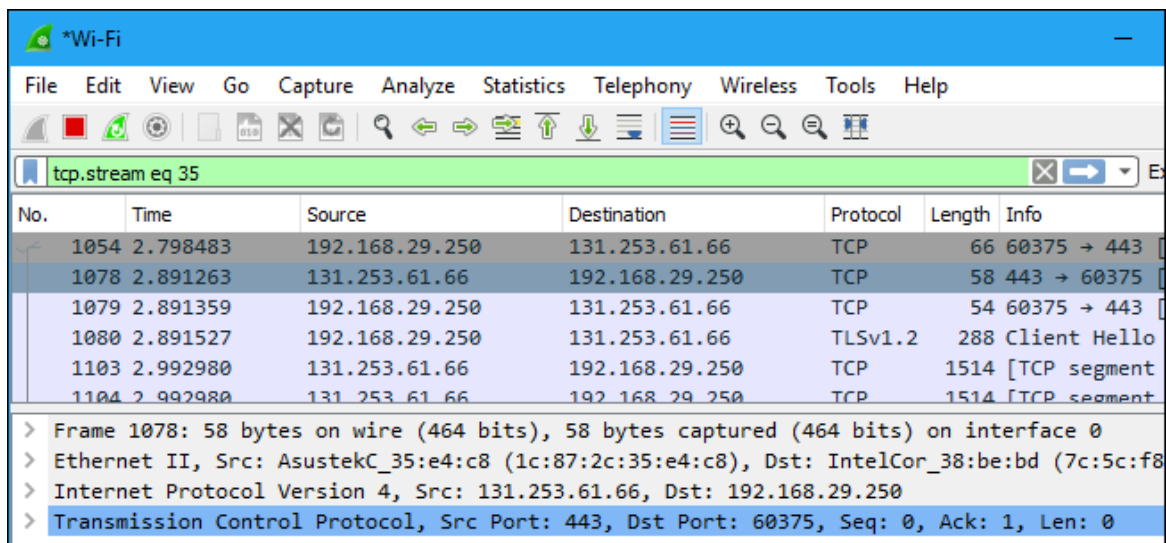


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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.



## Inspecting Packets

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

\*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 35

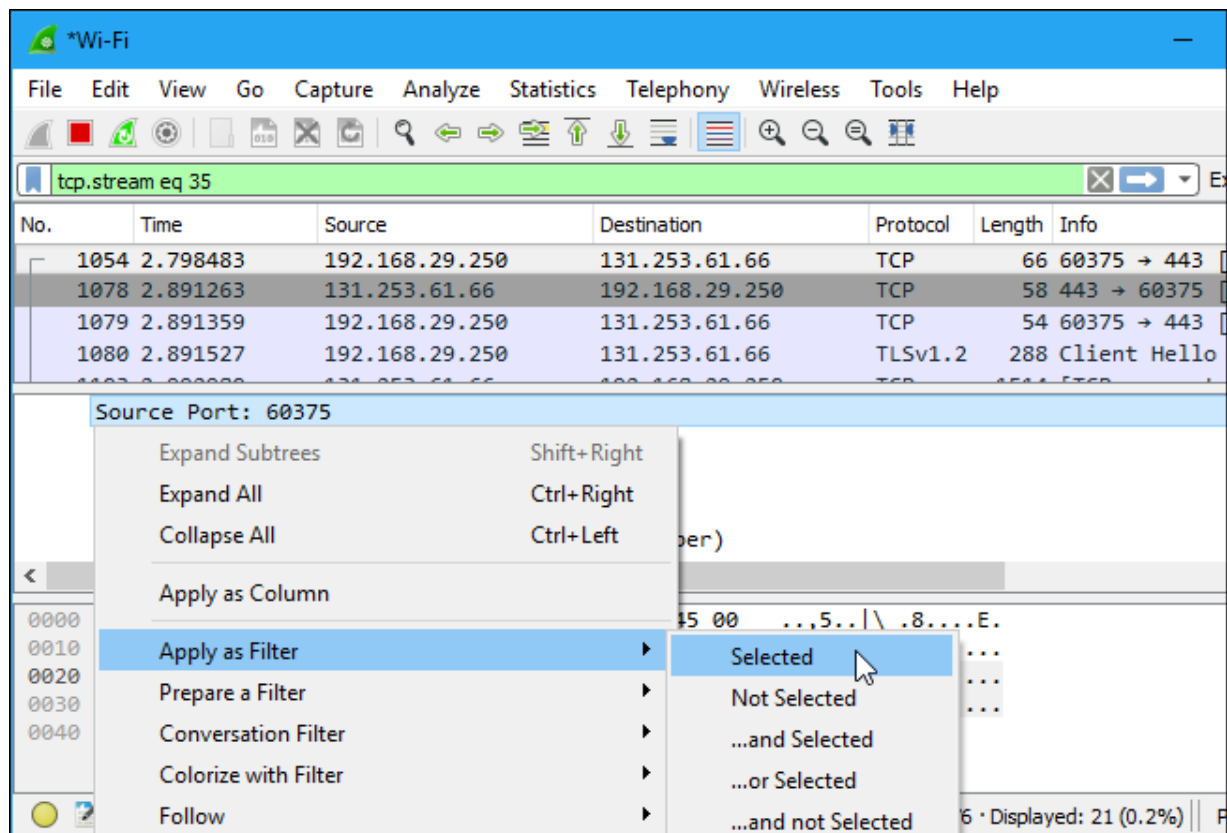
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

▼ 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%)

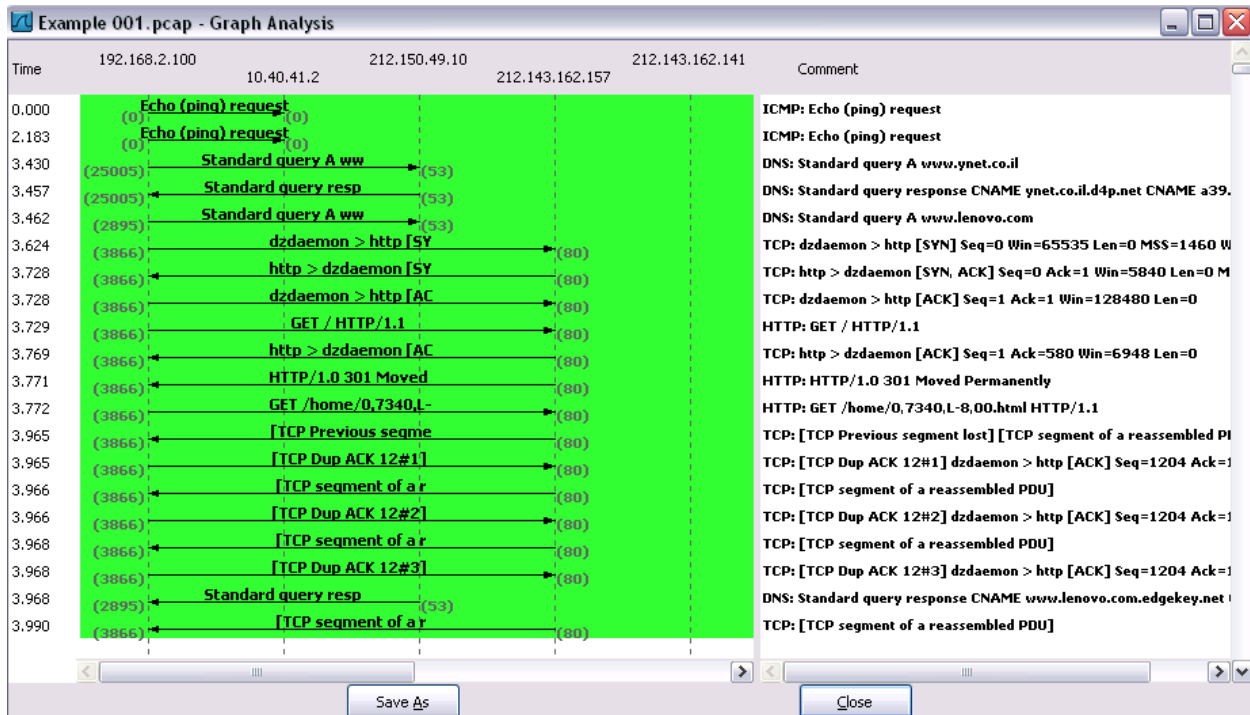
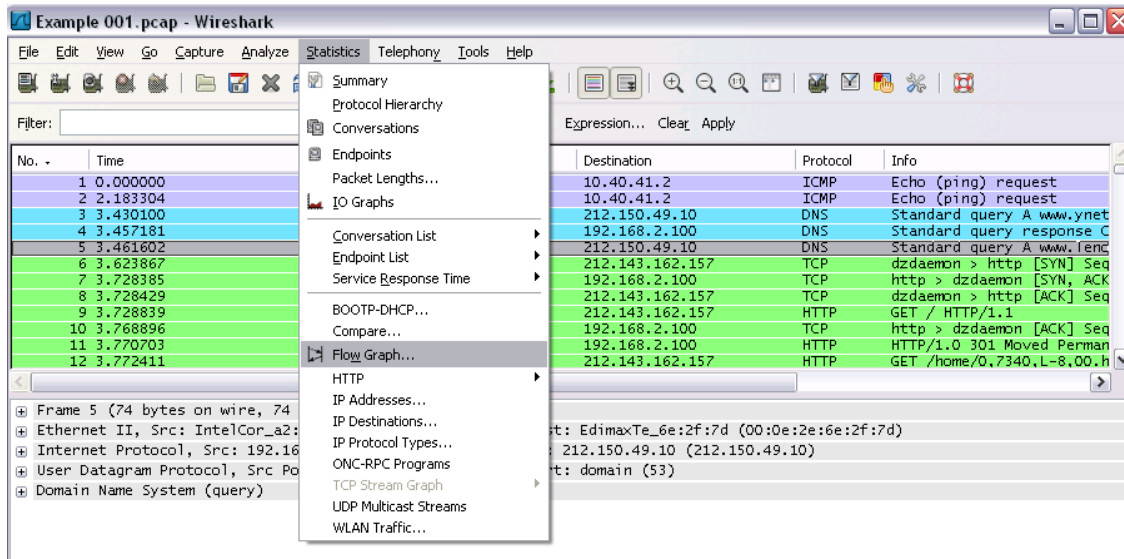
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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:4b

## 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

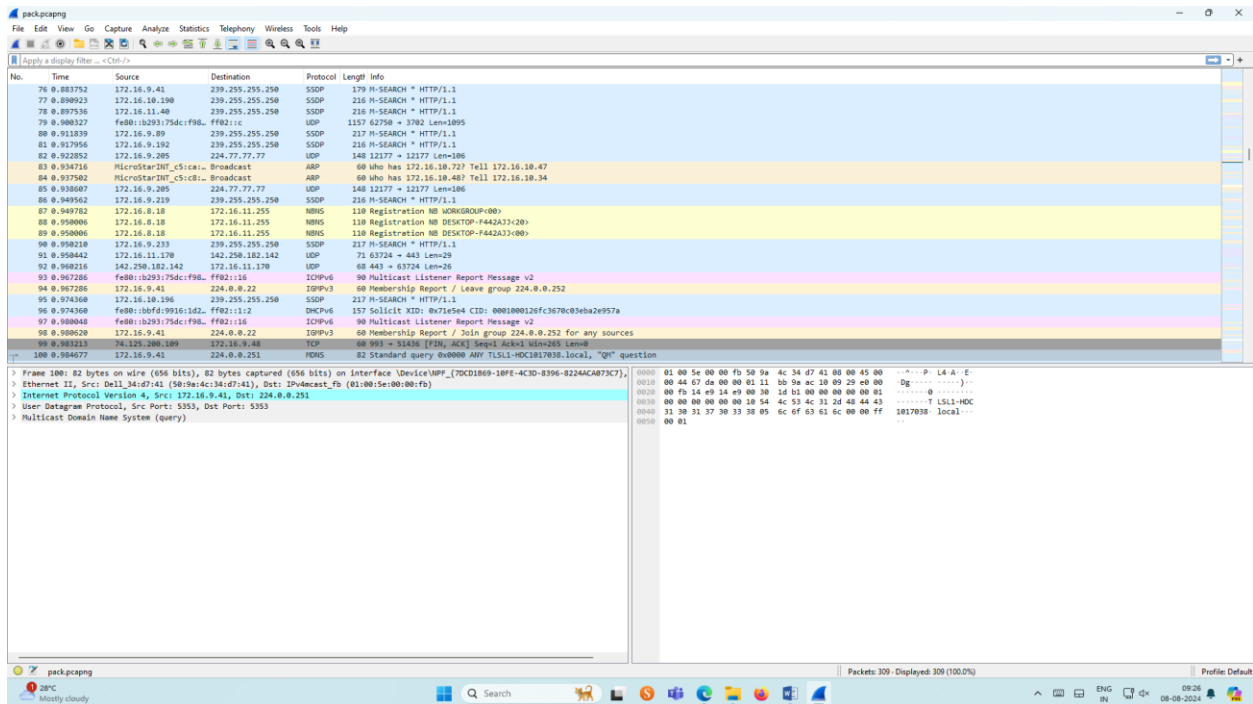
### Exercises

## 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:

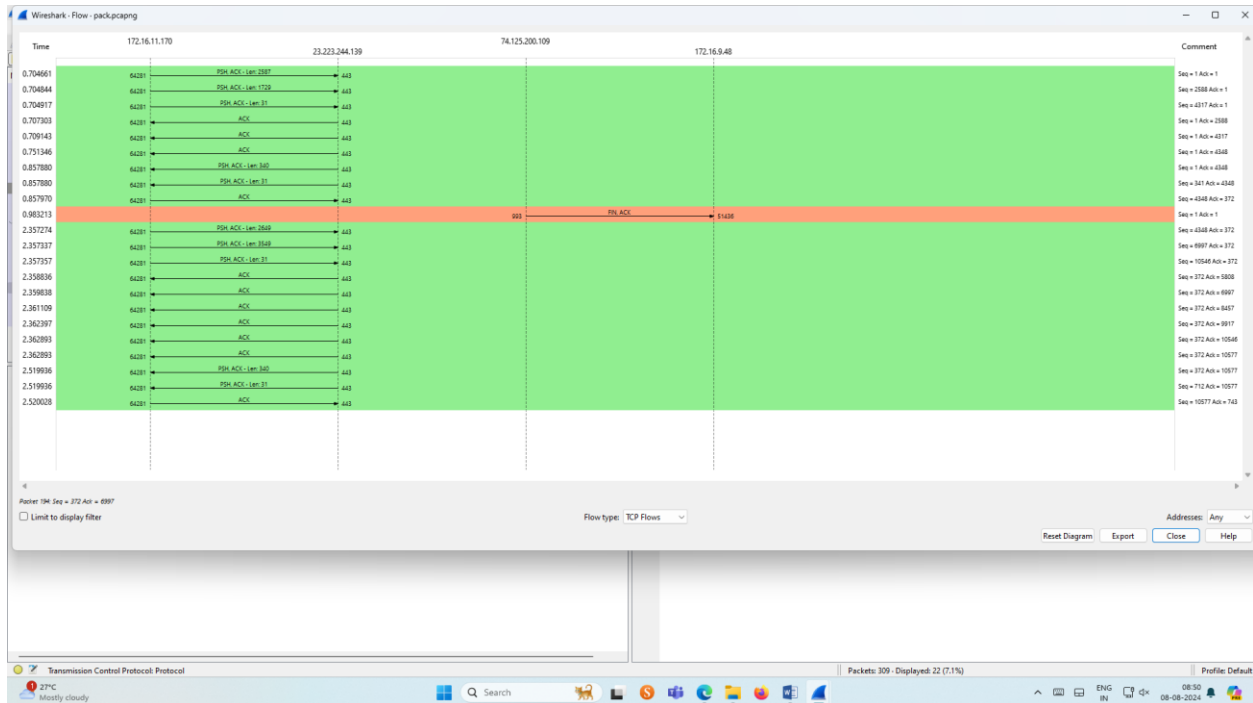
The screenshot displays the Wireshark interface with a network traffic capture. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for file operations, capture control, and analysis. The main pane shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packet list is filtered to show 'tcp or udp' packets. The packet details pane on the right shows the selected packet's structure, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Multicast Domain Name System (query). The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
25	0.114005	142.250.196.163	172.16.11.170	QUIC	979	Protected Payload (XPN)
26	0.114119	142.250.196.163	172.16.11.170	QUIC	72	Protected Payload (XPN)
27	0.114332	142.250.196.163	172.16.11.170	QUIC	66	Protected Payload (XPN)
28	0.114805	142.250.196.163	172.16.11.170	QUIC	64	Protected Payload (XPN)
29	0.114818	142.250.196.163	172.16.11.170	QUIC	73	Protected Payload (XPN), DCID=f42c93207f91c06a
30	0.114832	142.250.196.163	172.16.11.170	QUIC	73	Protected Payload (XPN), DCID=f42c93207f91c06a
31	0.117046	142.250.196.163	172.16.11.170	QUIC	162	Protected Payload (XPN)
32	0.140756	172.16.11.170	142.250.196.163	QUIC	74	Protected Payload (XPN), DCID=f42c93207f91c06a
33	0.148181	142.250.196.163	172.16.11.170	QUIC	559	Protected Payload (XPN)
34	0.148181	142.250.196.163	172.16.11.170	QUIC	64	Protected Payload (XPN)
35	0.148642	172.16.11.170	142.250.196.163	QUIC	77	Protected Payload (XPN), DCID=f42c93207f91c06a
36	0.152186	142.250.196.163	172.16.11.170	QUIC	67	Protected Payload (XPN)
37	0.156315	172.16.18.52	172.16.11.255	NBNS	92	Name query NB WPAD(00)
38	0.157718	172.16.18.167	172.16.11.255	BROWSER	255	Host Announcement APPLE-IPUC, Workstation, Server, NT Workstation
40	0.254287	fe80::e1d3:71ab:3c4...	ff02::1:3	LLMNR	84	Standard query INADDR.Aaaa upad
41	0.254287	fe80::e1d3:71ab:3c4...	ff02::1:3	LLMNR	84	Standard query Bx55f3 A upad
42	0.254287	172.16.18.227	224.0.0.252	LLMNR	64	Standard query INADDR.Aaaa upad
43	0.254287	172.16.18.227	224.0.0.252	LLMNR	64	Standard query INADDR.Aaaa upad
44	0.272335	fe80::c37:0625:0642...	ff02::1:2	DHCPv6	150	Solicit XID: 0x0f4f9e CID: 0001000121c3700001a4d7c0a2
46	0.347979	172.16.18.179	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
47	0.368496	172.16.18.179	239.255.255.250	SSDP	46	M-SEARCH * HTTP/1.1 Seq=8 hlen=64268 len=0 jts=1468 vs=256 SACK_PENI
48	0.370175	172.16.18.179	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
49	0.401971	172.16.18.179	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
50	0.402240	172.16.18.179	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
52	0.408407	172.16.18.179	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1

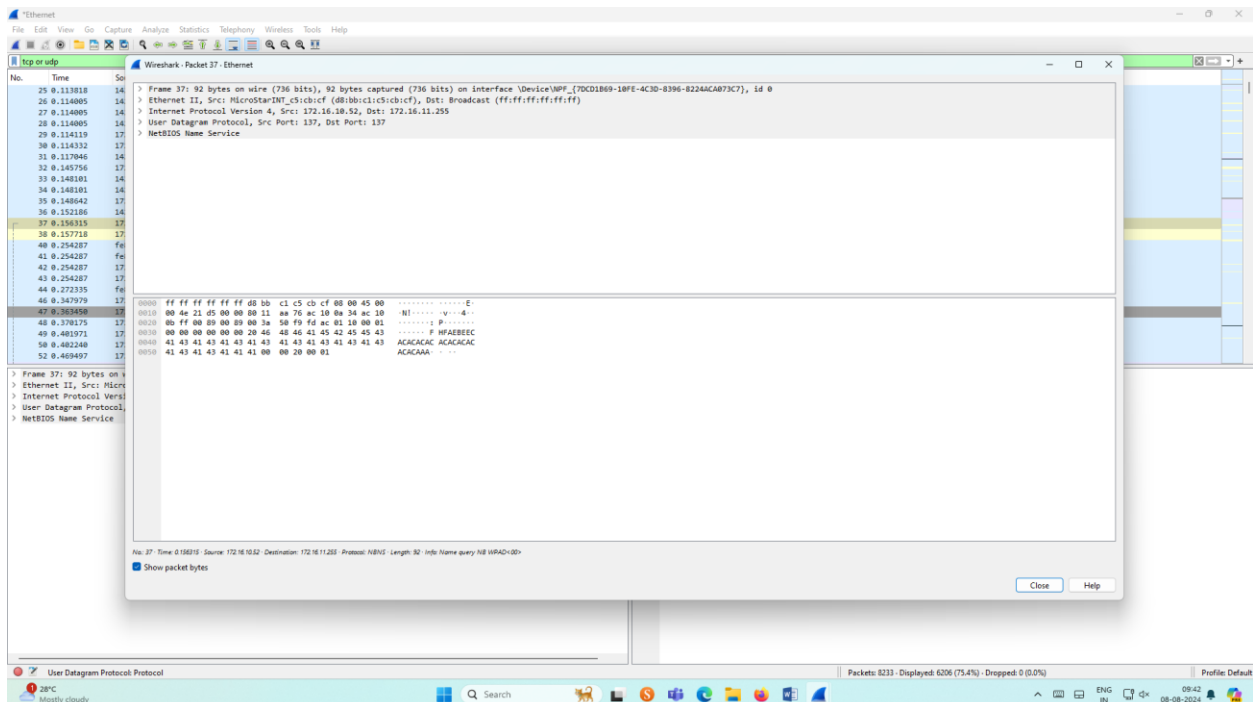
Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on Interface {Device\NPF\_{70C1B69-18FE-4C3D-8396-8224AC873C7}, 1  
> Ethernet II, Src: MicroStarINT\_05:cb:bf (d8:bb:c1:c5:cb:bf), Dst: IPMulticast\_Fb (01:00:5e:00:00:fb)  
> Internet Protocol Version 4, Src: 172.16.18.52, Dst: 224.0.0.251  
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353  
> Multicast Domain Name System (query)

0000 01 00 5e 00 00 fb d8 bb c1 c5 cb cf 00 00 45 00 --Aaaaaa-----E-  
0010 00 38 e5 68 00 00 01 11 3d 0d ac 10 0a 34 e0 00 --B:h-----a---4--  
0020 00 f5 14 e9 14 e9 00 24 29 e2 00 00 00 00 01 -----}-----  
0030 00 00 00 00 00 00 64 77 70 c1 64 05 sc ff 63 61 -----w pad loca  
0040 6c 00 00 01 00 01 1-----

## Flow Graph output




## Inspecting the packets

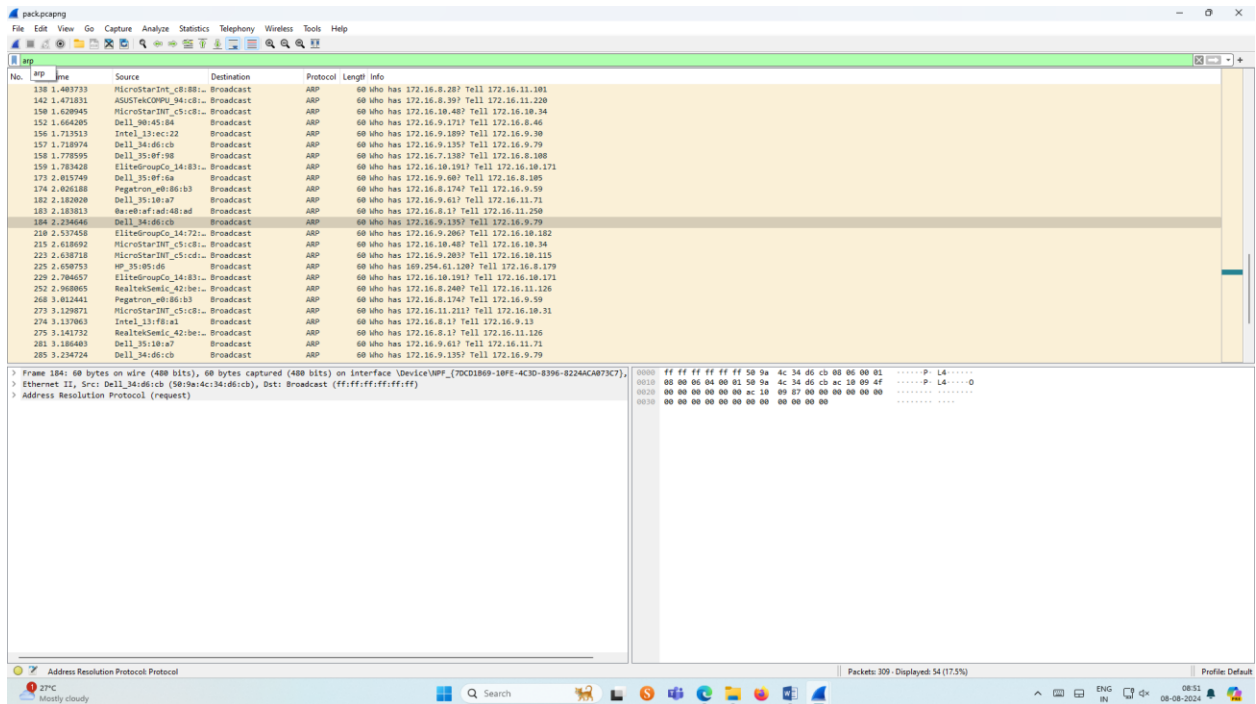


### 3.Create a Filter to display only ARP packets and inspect the packets.

#### 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 network protocol analyzer interface. The top pane displays a list of captured packets, all of which are ARP requests. The middle pane shows the details of the selected packet (No. 184), including the Ethernet II header and the Address Resolution Protocol (ARP) section. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
138	1.481733	MicroStarInt_c8:88:1	Broadcast	ARP	60	Who has 172.16.8.28? Tell 172.16.11.181
142	1.471831	ASUSTekCOMPU_94:c8:1	Broadcast	ARP	60	Who has 172.16.8.39? Tell 172.16.11.220
150	1.628945	MicroStarInt_c5:c8:1	Broadcast	ARP	60	Who has 172.16.10.48? Tell 172.16.10.34
151	1.664295	Dell_34:d6:cb	Broadcast	ARP	60	Who has 172.16.9.13? Tell 172.16.8.46
156	1.713513	Intel_13:ec:22	Broadcast	ARP	60	Who has 172.16.9.189? Tell 172.16.9.30
157	1.718974	Dell_34:d6:cb	Broadcast	ARP	60	Who has 172.16.9.135? Tell 172.16.9.79
158	1.778595	Dell_35:0f:58	Broadcast	ARP	60	Who has 172.16.7.118? Tell 172.16.8.380
159	1.783428	EliteGroupCo_14:83:1	Broadcast	ARP	60	Who has 172.16.10.191? Tell 172.16.10.171
173	2.015749	Dell_35:0f:5a	Broadcast	ARP	60	Who has 172.16.9.60? Tell 172.16.8.185
174	2.026188	Pegatron_eb:86:b3	Broadcast	ARP	60	Who has 172.16.8.174? Tell 172.16.9.59
182	2.182020	Dell_35:18:a7	Broadcast	ARP	60	Who has 172.16.9.61? Tell 172.16.11.71
183	2.183813	Waioeaf:ad:48:ad	Broadcast	ARP	60	Who has 172.16.8.1? Tell 172.16.11.250
184	2.236446	Dell_34:d6:cb	Broadcast	ARP	60	Who has 172.16.9.135? Tell 172.16.9.79
218	2.531458	EliteGroupCo_14:72:1	Broadcast	ARP	60	Who has 172.16.9.200? Tell 172.16.10.182
215	2.618692	MicroStarInt_c5:c8:1	Broadcast	ARP	60	Who has 172.16.10.48? Tell 172.16.10.34
223	2.638718	MicroStarInt_c5:c8:1	Broadcast	ARP	60	Who has 172.16.9.203? Tell 172.16.10.115
225	2.658753	HP_3b:95:d6	Broadcast	ARP	60	Who has 169.254.61.120? Tell 172.16.8.179
229	2.784657	EliteGroupCo_14:83:1	Broadcast	ARP	60	Who has 172.16.10.191? Tell 172.16.10.171
252	2.968005	RealtekSem_42:b6:1	Broadcast	ARP	60	Who has 172.16.8.240? Tell 172.16.11.126
268	3.012441	Pegatron_eb:86:b3	Broadcast	ARP	60	Who has 172.16.8.174? Tell 172.16.9.59
273	3.128871	MicroStarInt_c5:c8:1	Broadcast	ARP	60	Who has 172.16.11.211? Tell 172.16.10.31
274	3.137063	Intel_13:fb:a1	Broadcast	ARP	60	Who has 172.16.8.1? Tell 172.16.9.13
275	3.141732	RealtekSem_42:b6:1	Broadcast	ARP	60	Who has 172.16.8.1? Tell 172.16.11.126
281	3.166480	Dell_35:18:a7	Broadcast	ARP	60	Who has 172.16.9.61? Tell 172.16.11.71
285	3.234724	Dell_34:d6:cb	Broadcast	ARP	60	Who has 172.16.9.135? Tell 172.16.9.79

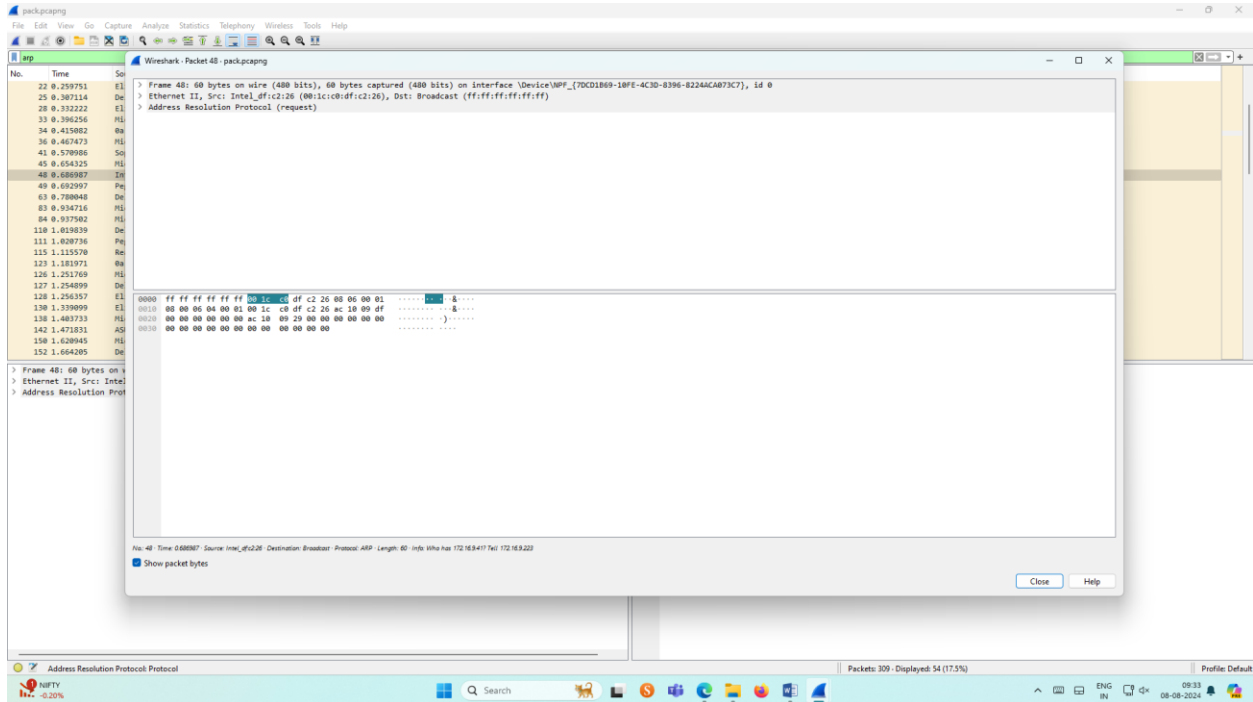
Packet 184 details:

- Ethernet II, Src: Dell\_34:d6:cb (58:9a:4c:34:d6:cb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)

Raw packet data (hex):



```
0000 ff ff ff ff ff ff 58 9a 4c 34 d6 cb 00 00 01 .....P: L4-----
0010 00 00 06 04 00 01 58 9a 4c 34 d6 cb ac 18 09 4f .....P: L4-----0
0020 00 00 00 00 00 00 ac 18 09 4f 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

## Inspecting the packets



## 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 displays the Wireshark interface with a DNS traffic capture. The packet list on the left shows four packets: a standard query from 172.16.11.170 to 172.16.8.1, and three standard query responses from 172.16.8.1 back to 172.16.11.170. The packet details pane on the right shows the structure of the first packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (query). The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

# Graph output


The screenshot displays the Wireshark interface with a packet capture graph. The graph shows a series of green bars representing network traffic over time. The x-axis represents time, and the y-axis represents the number of bytes. The graph shows a series of green bars representing network traffic over time. The x-axis represents time, and the y-axis represents the number of bytes. The graph shows a series of green bars representing network traffic over time. The x-axis represents time, and the y-axis represents the number of bytes.

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

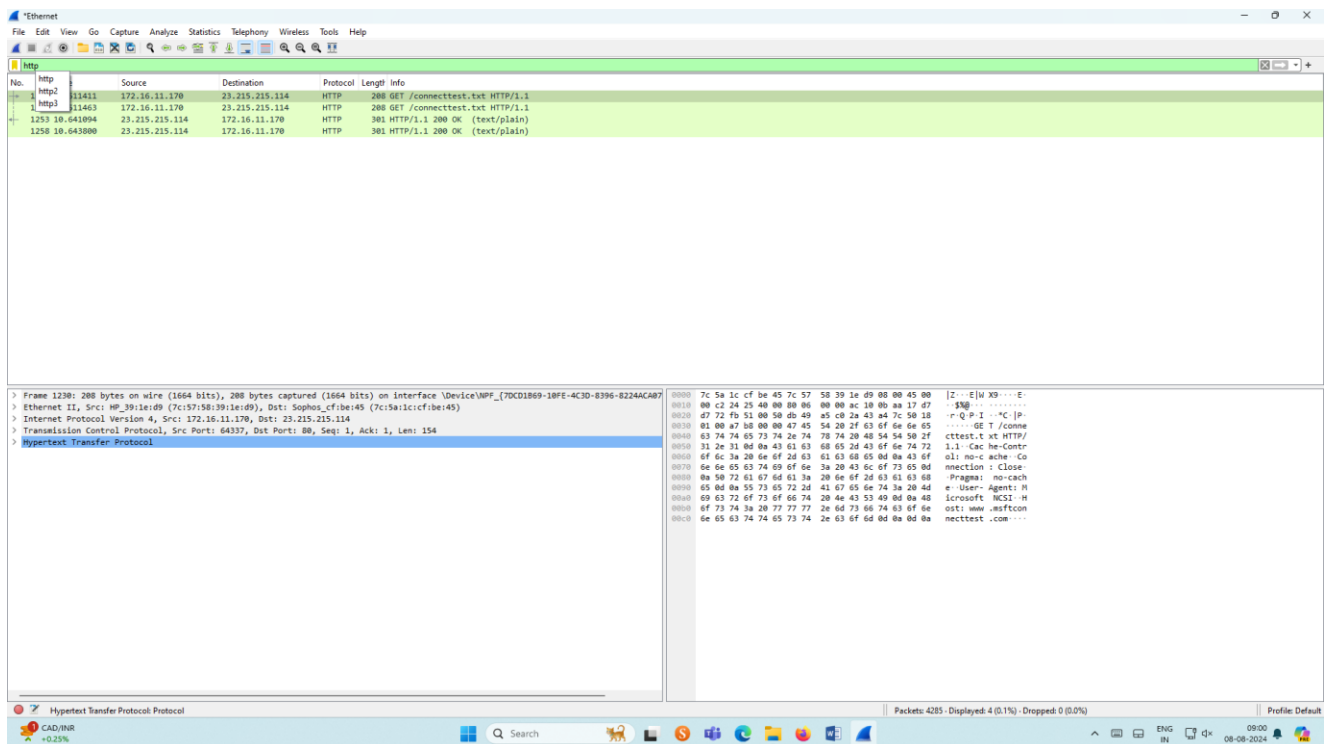
Procedure

## 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



The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for file operations, capture control, and analysis. The main packet list pane shows a filtered view of HTTP packets. The details pane on the right provides a hierarchical view of the selected packet's structure, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet list pane shows the following packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.11.170	23.215.215.114	HTTP	200	GET /connecttest.txt HTTP/1.1
2	0.000000	23.215.215.114	172.16.11.170	HTTP	200	GET /connecttest.txt HTTP/1.1
3	0.000000	23.215.215.114	172.16.11.170	HTTP	301	HTTP/1.1 200 OK (text/plain)

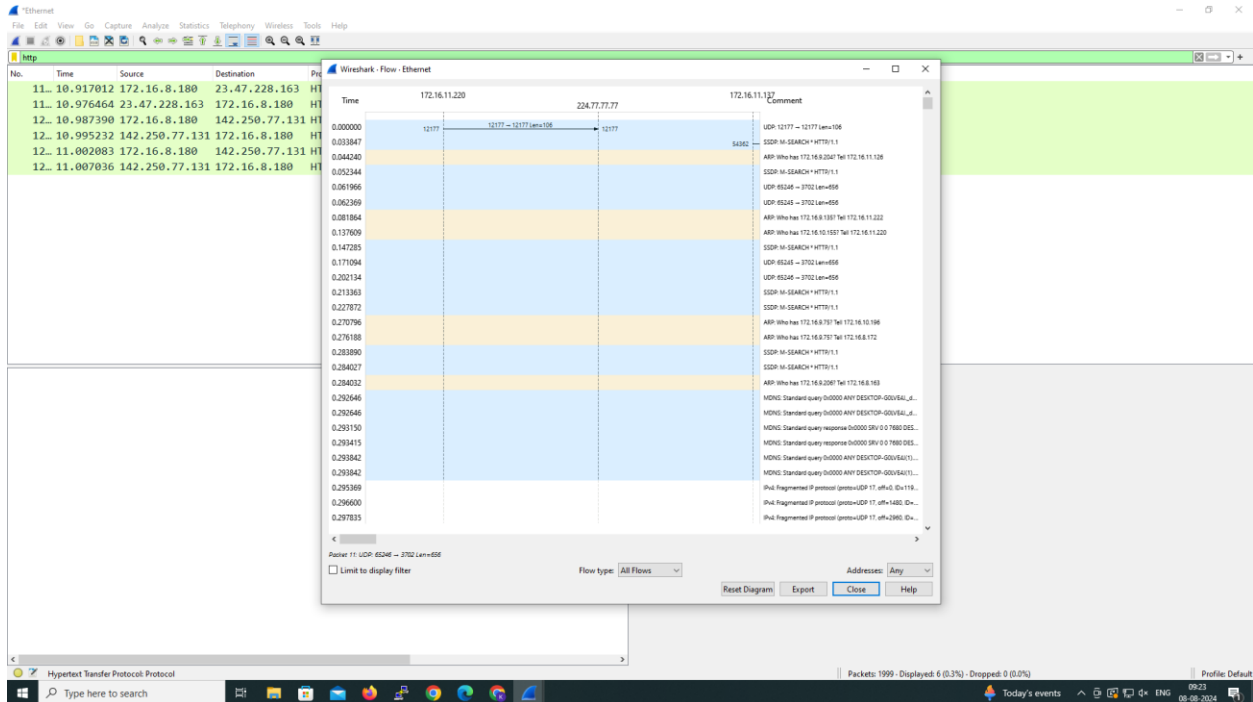
The details pane for the selected packet (No. 1) shows the following structure:

- Ethernet II, Src: HP\_39:1e:d9 (7c:97:9b:39:1e:d9), Dst: Suplus\_cf:be:45 (7c:9a:1c:cf:be:45)
- Internet Protocol Version 4, Src: 172.16.11.170, Dst: 23.215.215.114
- Transmission Control Protocol, Src Port: 64337, Dst Port: 80, Seq: 1, Ack: 1, Len: 154
- Hypertext Transfer Protocol

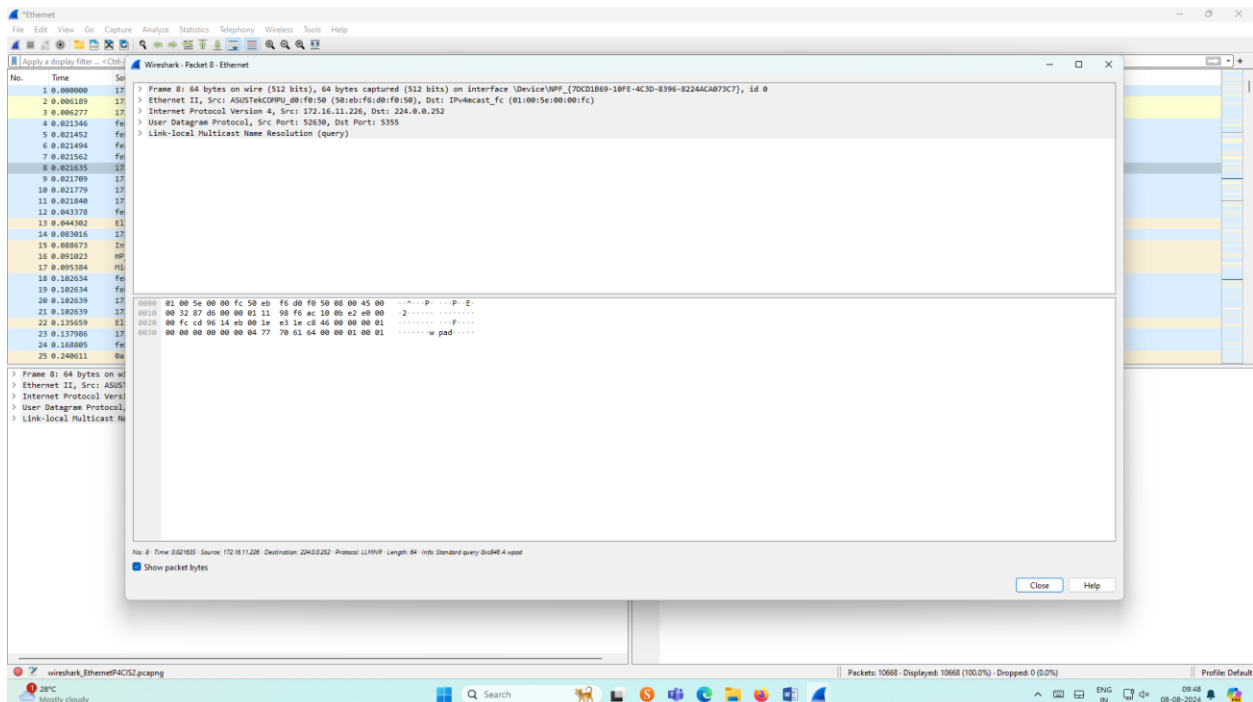
The packet bytes pane at the bottom shows the raw data of the selected packet, including the Ethernet II header, IP header, TCP header, and HTTP request body.



## Flow Graph output




## Inspecting the packets

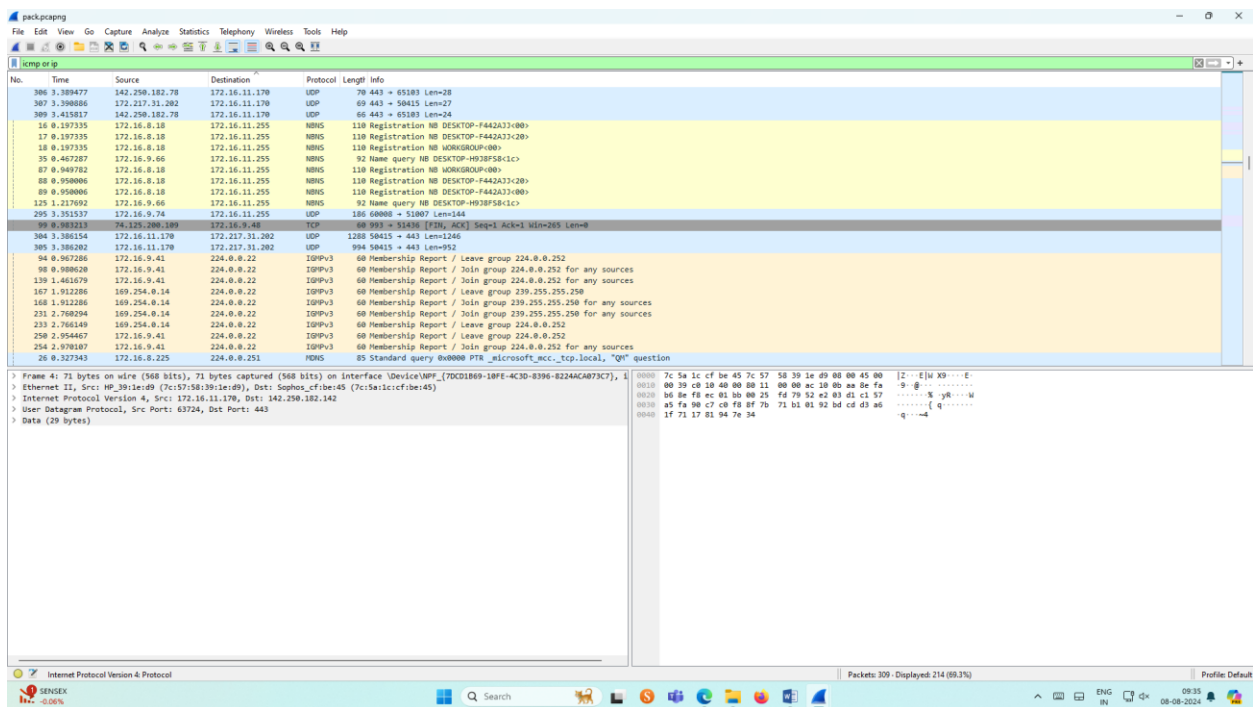


## 6.Create a Filter to display only IP/ICMP packets and inspect the packets.

### 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

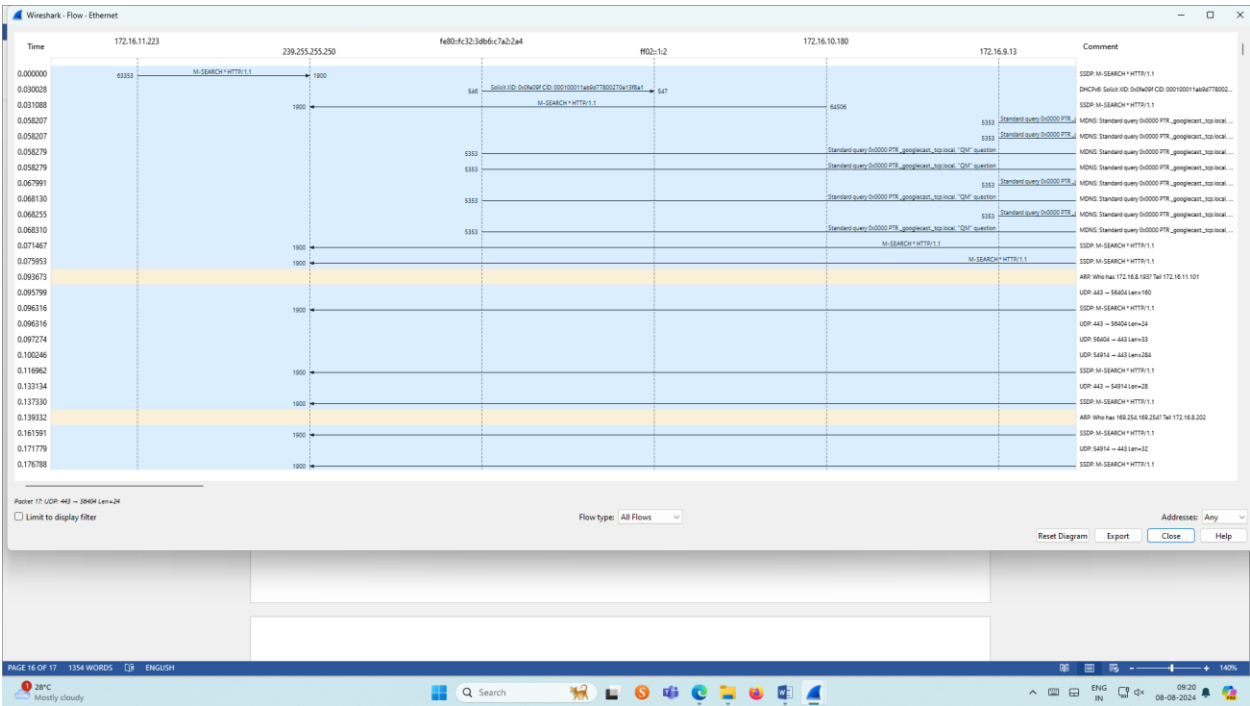


The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for opening files, saving, capturing, and analyzing. The main window is divided into three panes:

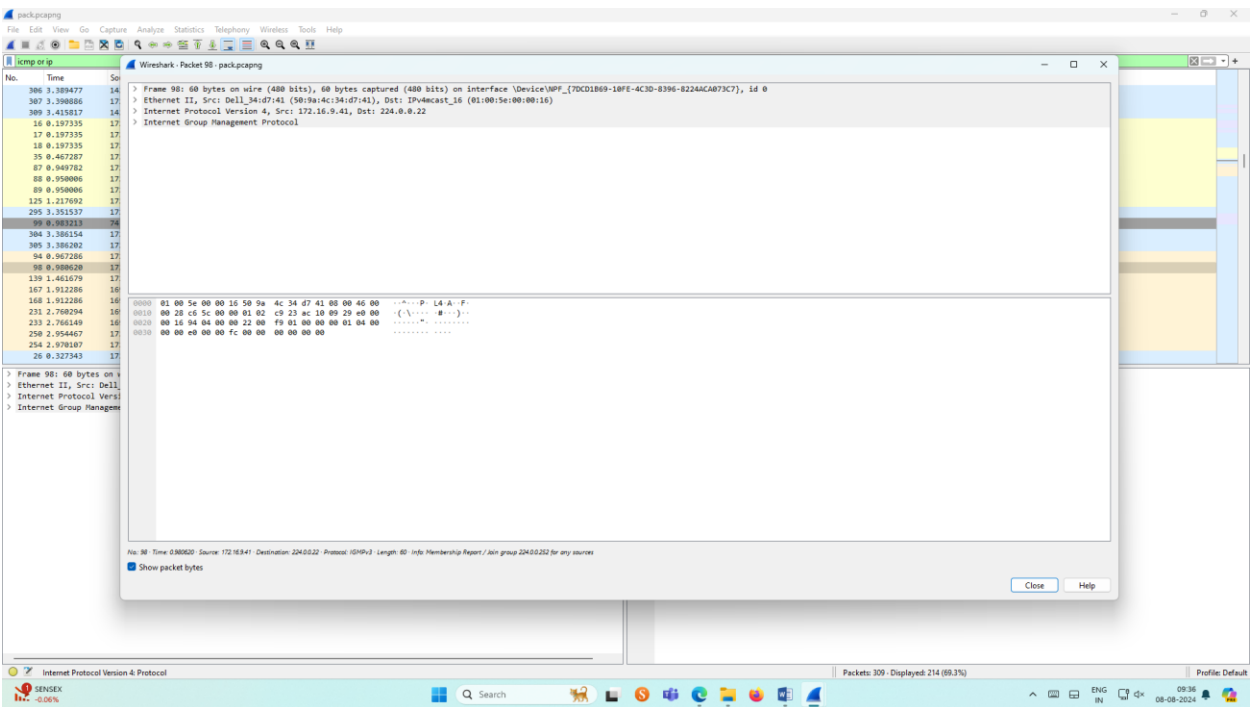
- Packet List Pane:** Displays a list of captured packets. The first few packets are UDP and ICMP. Packet 295 is highlighted, showing it's an ICMP Echo (ping) request from 172.16.11.170 to 172.16.11.255.
- Packet Details Pane:** Shows the hierarchical structure of the selected packet (Frame 4). It includes Ethernet II, Internet Protocol Version 4, and User Datagram Protocol.
- Packet Bytes Pane:** Displays the raw data of the selected packet in hexadecimal and ASCII.

The bottom status bar indicates that 300 packets are displayed, representing 214 (89.3%) of the total capture.

# Flow Graph output




# Inspecting the packets

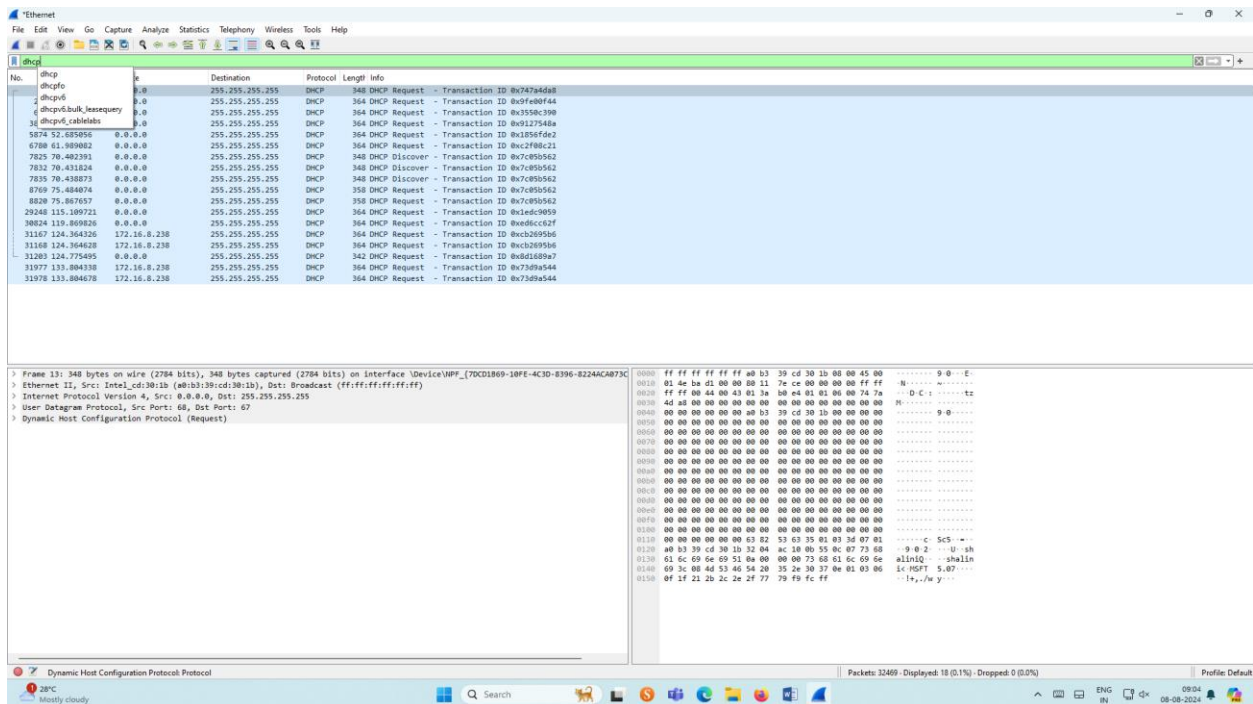


## 7. Create a Filter to display only DHCP packets and inspect the packets.

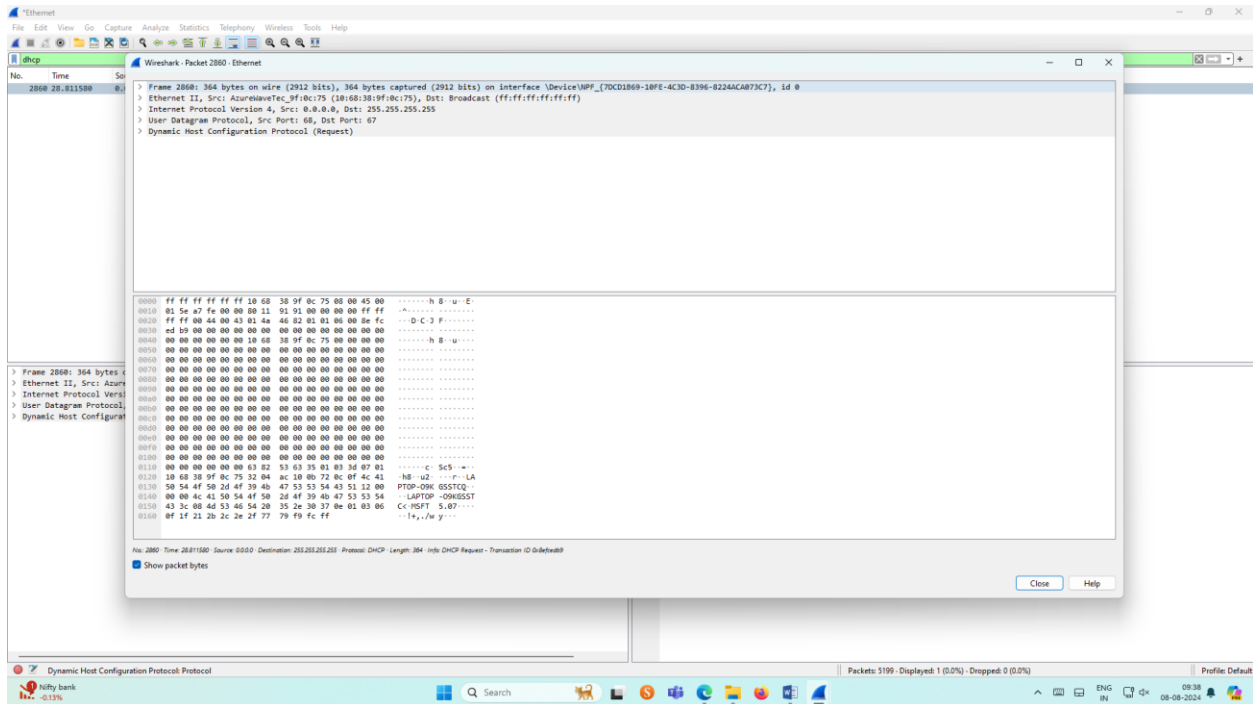
## Procedure

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

## Output



## Inspecting the packets



Result:

The filtering, searching and inspecting of packets using wireshark tool has been done successfully .