231901020 **Kaviya V(CSE(CS)**

Ex No: 14a 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:

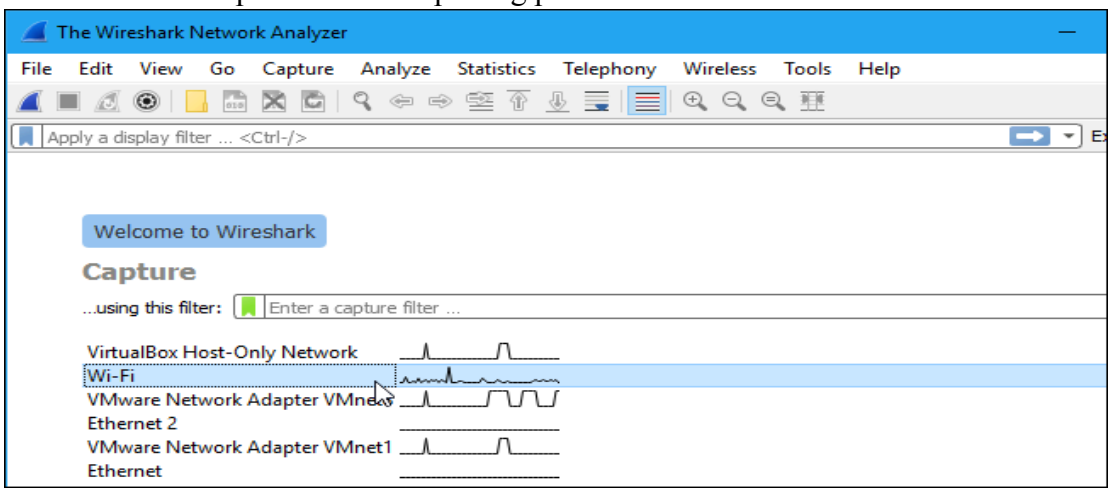
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

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.

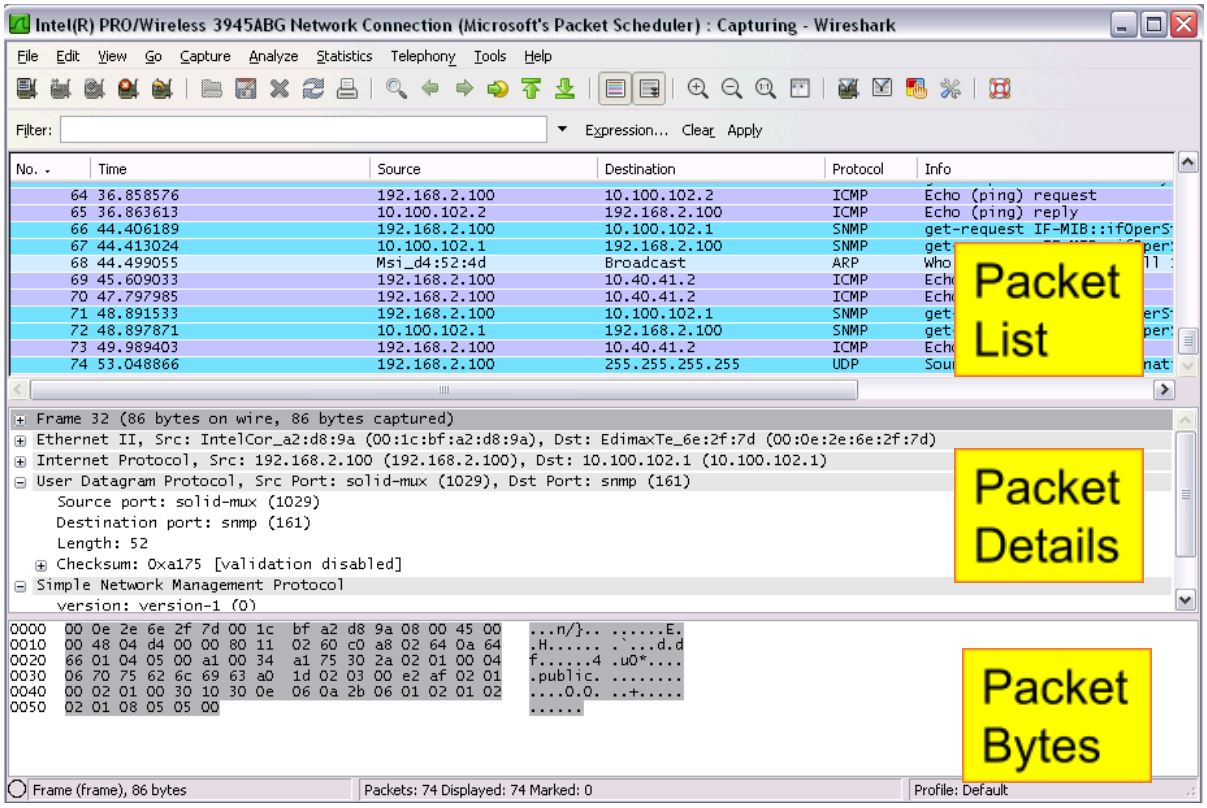
Capturing Packets

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.

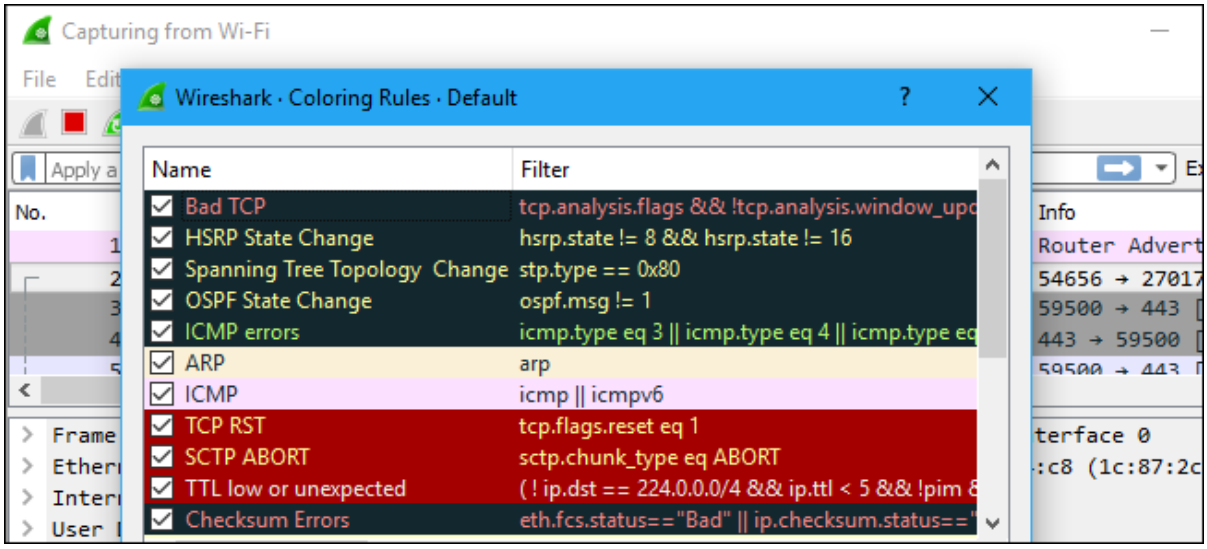
The “Packet Bytes” Pane

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.

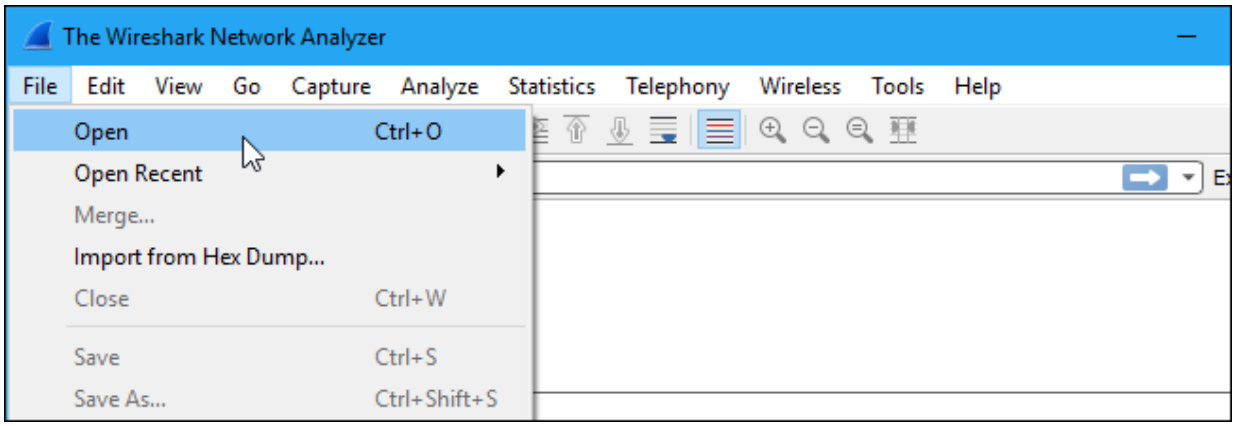
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



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.

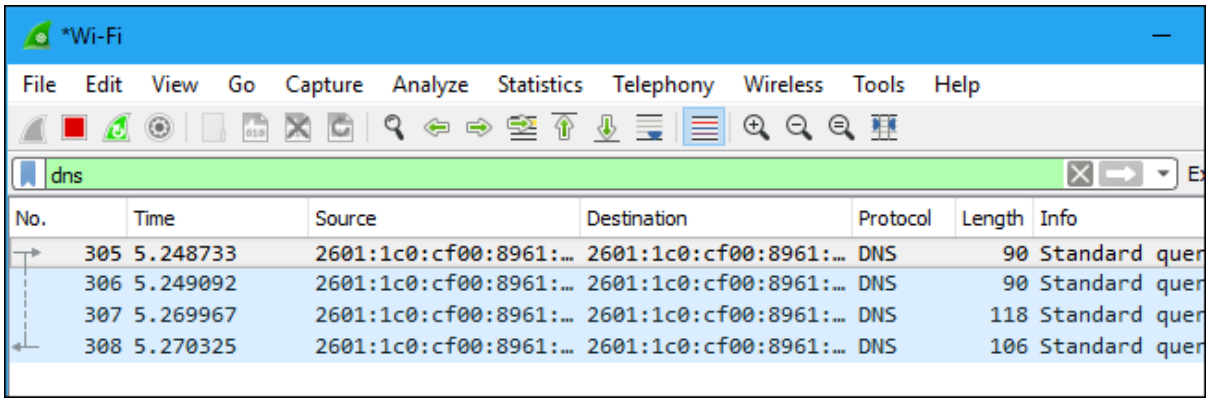


Filtering Packets

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

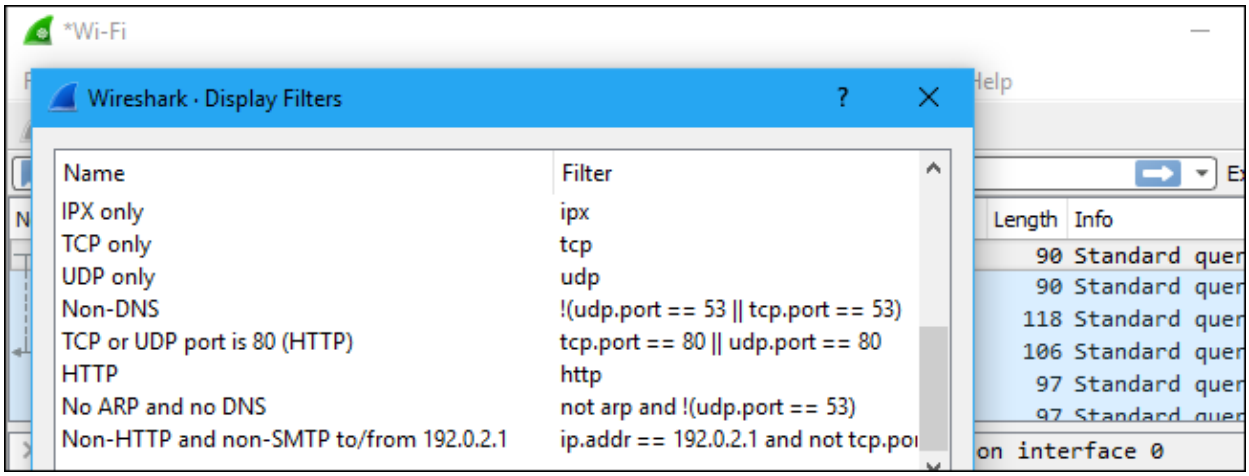
the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

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.



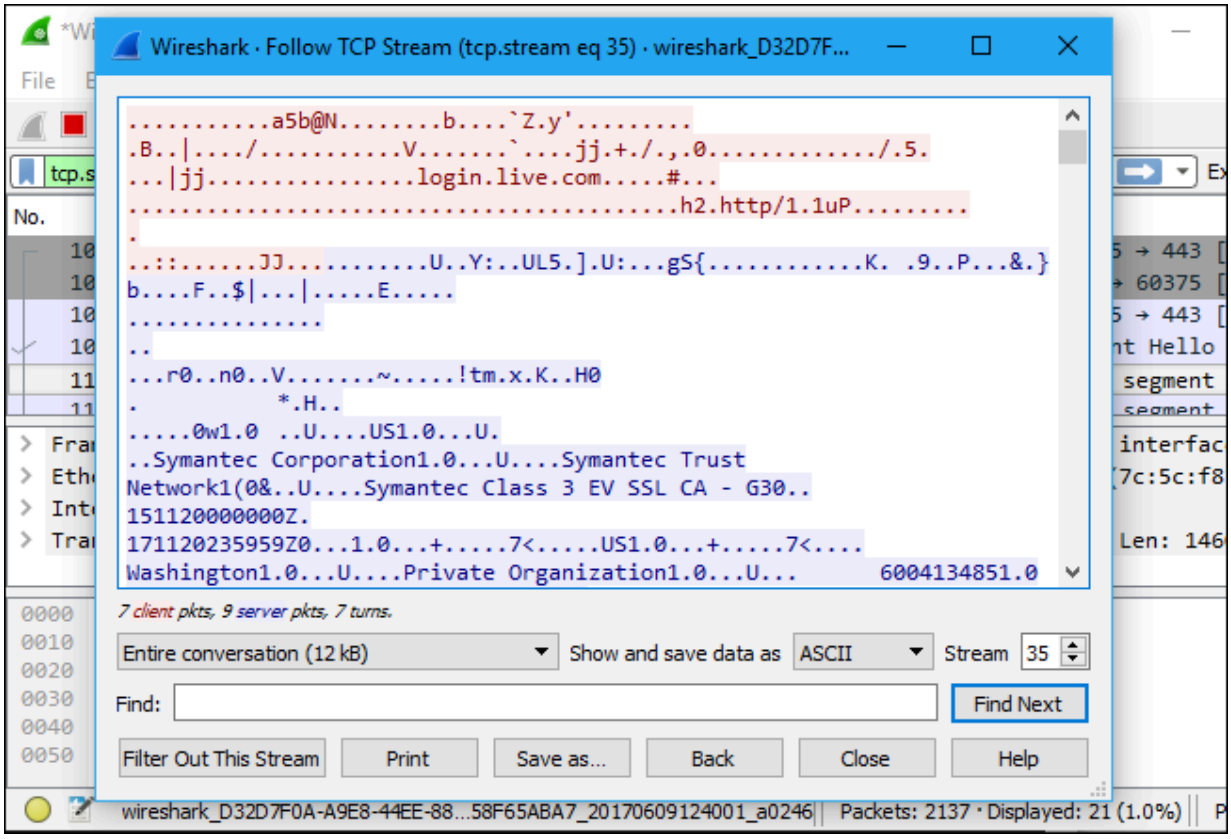
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

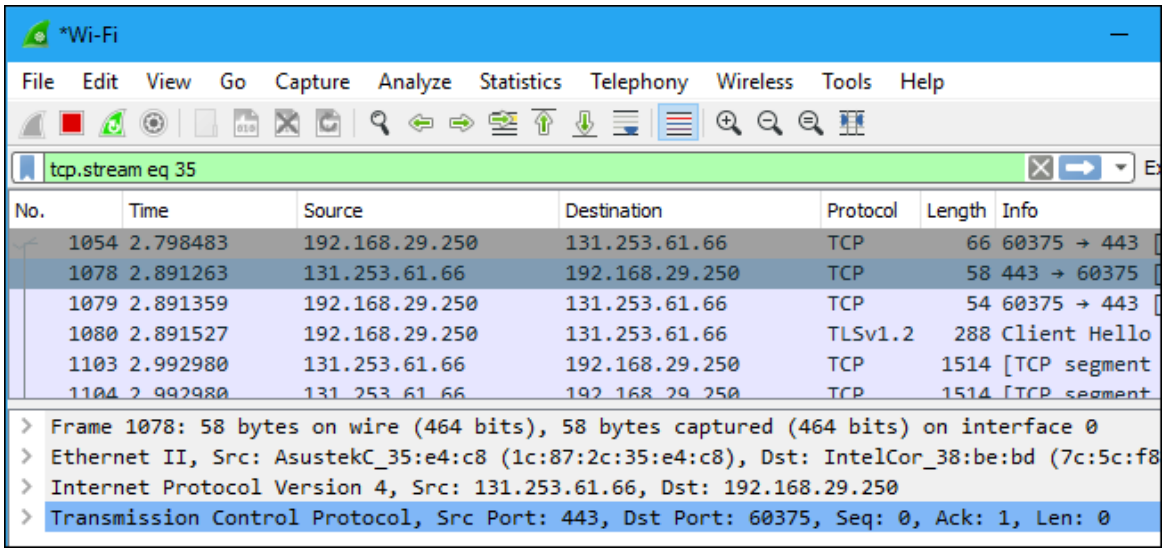


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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.

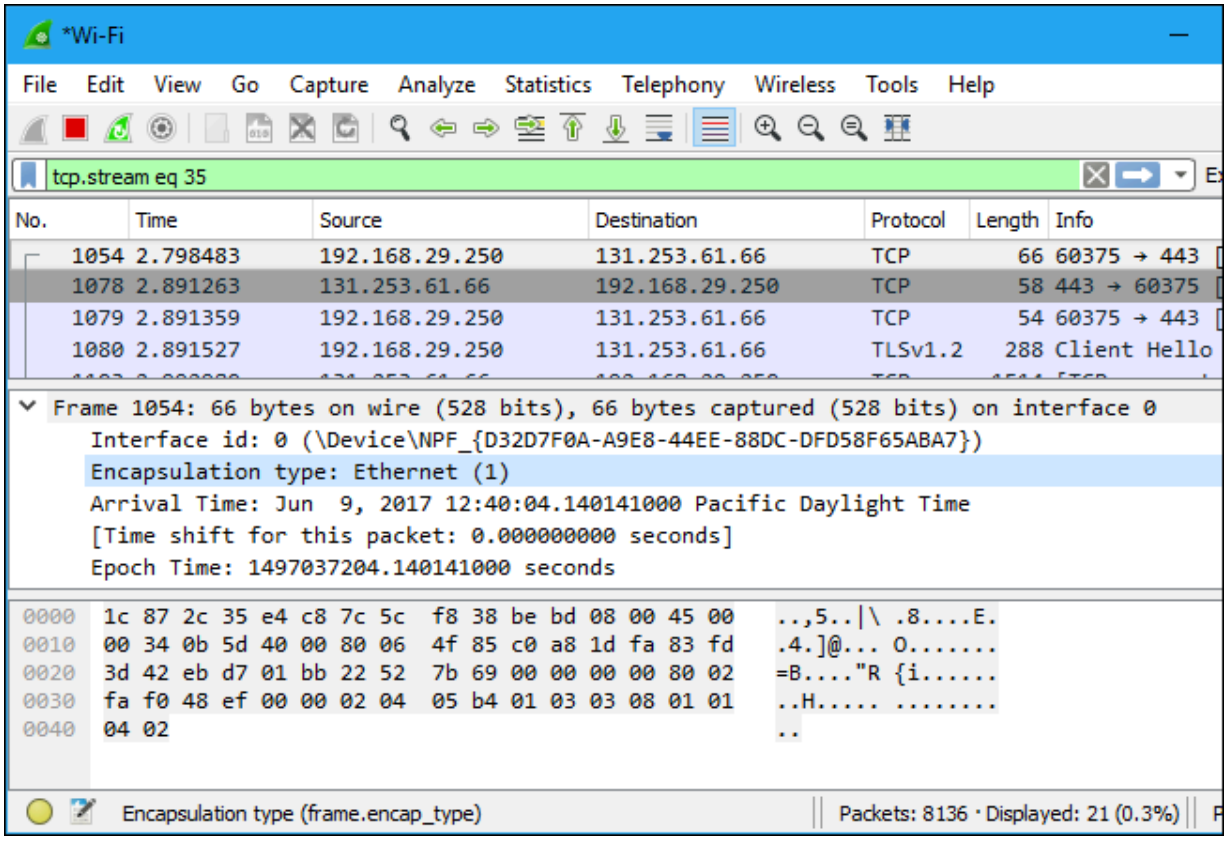


Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

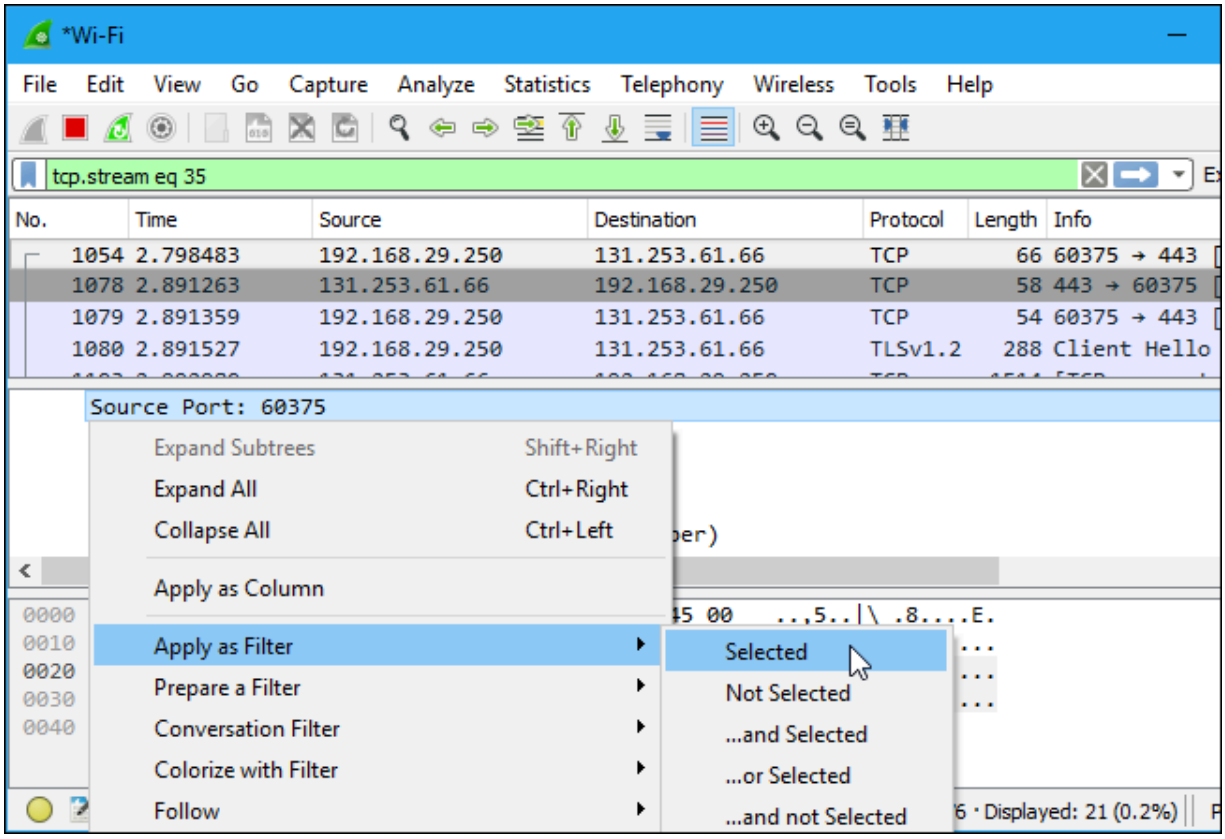


Inspecting Packets

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

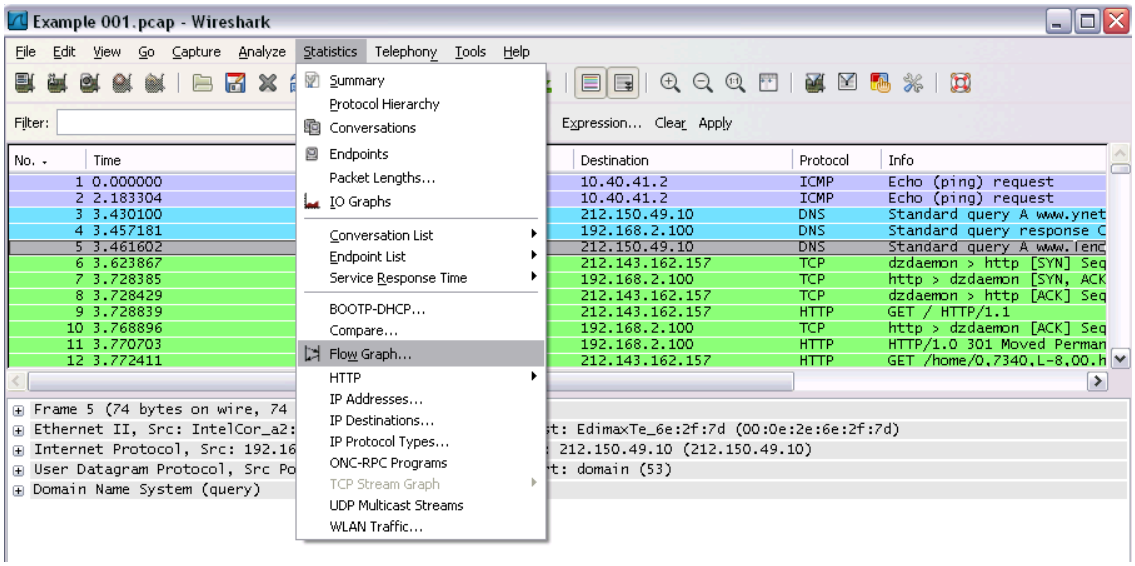


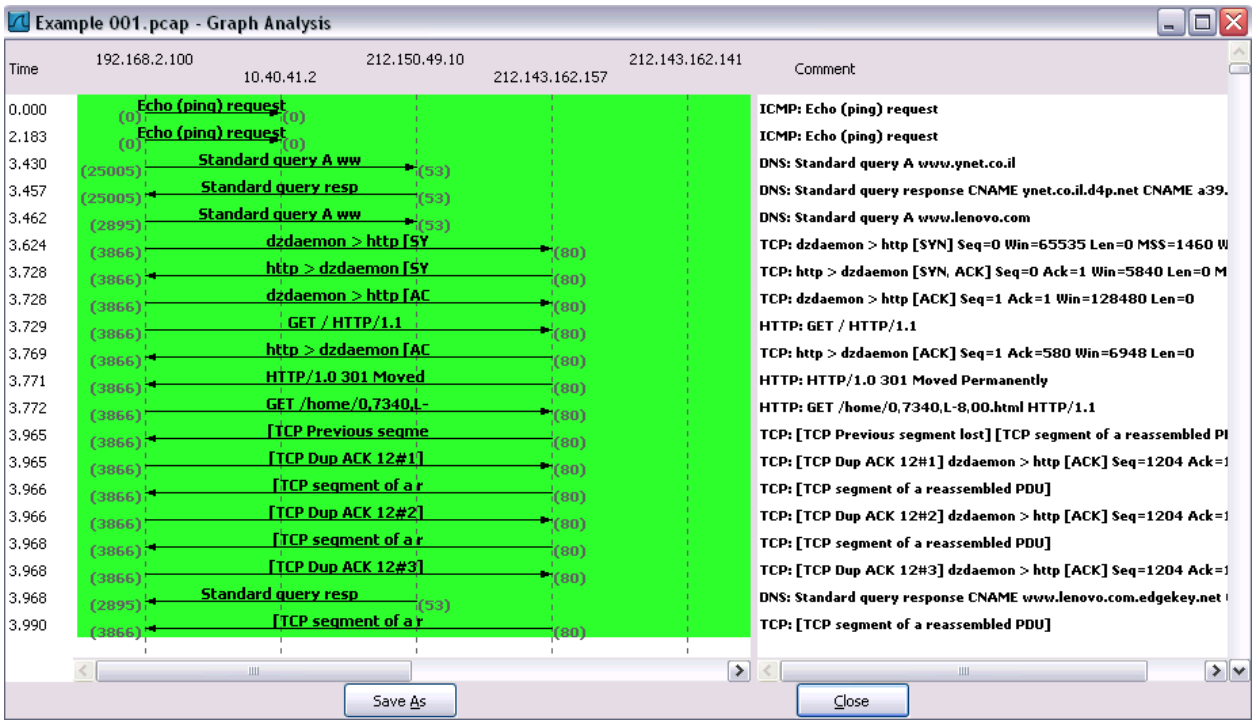
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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.


Example 001.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Summary
Protocol Hierarchy
Conversations
Endpoints
Packet Lengths...
IO Graphs
Conversation List
Endpoint List
Service Response Time
BOOTP-DHCP...
Compare...
Flow Graph...
HTTP
IP Addresses...
IP Destinations...
IP Protocol Types...
ONC-RPC Programs
TCP Stream Graph
UDP Multicast Streams
WLAN Traffic...
Filter:
Expression... Clear Apply
No. Time Destination Protocol Info
1 0.000000 10.40.41.2 ICMP Echo (ping) request
2 2.183304 10.40.41.2 ICMP Echo (ping) request
3 3.430100 212.150.49.10 DNS Standard query A www.ynet
4 3.457181 192.168.2.100 DNS Standard query response C
5 3.461602 212.150.49.10 DNS Standard query A www.lenc
6 3.623867 212.143.162.157 TCP dzdaemon > http [SYN] Seq
7 3.728385 192.168.2.100 TCP http > dzdaemon [SYN, ACK
8 3.728429 212.143.162.157 TCP dzdaemon > http [ACK] Seq
9 3.728839 212.143.162.157 HTTP GET / HTTP/1.1
10 3.768896 192.168.2.100 TCP http > dzdaemon [ACK] Seq
11 3.770703 192.168.2.100 HTTP HTTP/1.0 301 Moved Perman
12 3.772411 212.143.162.157 HTTP GET /home/0,7340,L-8,00.h
Frame 5 (74 bytes on wire, 74
Ethernet II, Src: IntelCor_a2:
Internet Protocol, Src: 192.16
User Datagram Protocol, Src Po
Domain Name System (query)
t: EdimaxTe_6e:2f:7d (00:0e:2e:6e:2f:7d)
212.150.49.10 (212.150.49.10)
t: domain (53)


Example 001.pcap - Graph Analysis
Time
192.168.2.100
10.40.41.2
212.150.49.10
212.143.162.157
212.143.162.141
Comment
0.000 Echo (ping) request (0) (0) ICMP: Echo (ping) request
2.183 Echo (ping) request (0) (0) ICMP: Echo (ping) request
3.430 Standard query A ww (25005) (53) DNS: Standard query A www.ynet.co.il
3.457 Standard query resp (25005) (53) DNS: Standard query response CNAME ynet.co.il.d4p.net CNAME a39.
3.462 Standard query A ww (2895) (53) DNS: Standard query A www.lenovo.com
3.624 dzdaemon > http [SY (3866) (80) TCP: dzdaemon > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 W
3.728 http > dzdaemon [SY (3866) (80) TCP: http > dzdaemon [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M
3.728 dzdaemon > http [AC (3866) (80) TCP: dzdaemon > http [ACK] Seq=1 Ack=1 Win=128480 Len=0
3.729 GET / HTTP/1.1 (3866) (80) HTTP: GET / HTTP/1.1
3.769 http > dzdaemon [AC (3866) (80) TCP: http > dzdaemon [ACK] Seq=1 Ack=580 Win=6948 Len=0
3.771 HTTP/1.0 301 Moved (3866) (80) HTTP: HTTP/1.0 301 Moved Permanently
3.772 GET /home/0,7340,L- (3866) (80) HTTP: GET /home/0,7340,L-8,00.html HTTP/1.1
3.965 [TCP Previous segme (3866) (80) TCP: [TCP Previous segment lost] [TCP segment of a reassembled PI
3.965 [TCP Dup ACK 12#1] (3866) (80) TCP: [TCP Dup ACK 12#1] dzdaemon > http [ACK] Seq=1204 Ack=1
3.966 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
3.966 [TCP Dup ACK 12#2] (3866) (80) TCP: [TCP Dup ACK 12#2] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
3.968 [TCP Dup ACK 12#3] (3866) (80) TCP: [TCP Dup ACK 12#3] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968 Standard query resp (2895) (53) DNS: Standard query response CNAME www.lenovo.com.edgekey.net
3.990 [TCP segment of a r (3866) (80) TCP: [TCP segment of a reassembled PDU]
Save As
Close

Ex No: 14 b **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

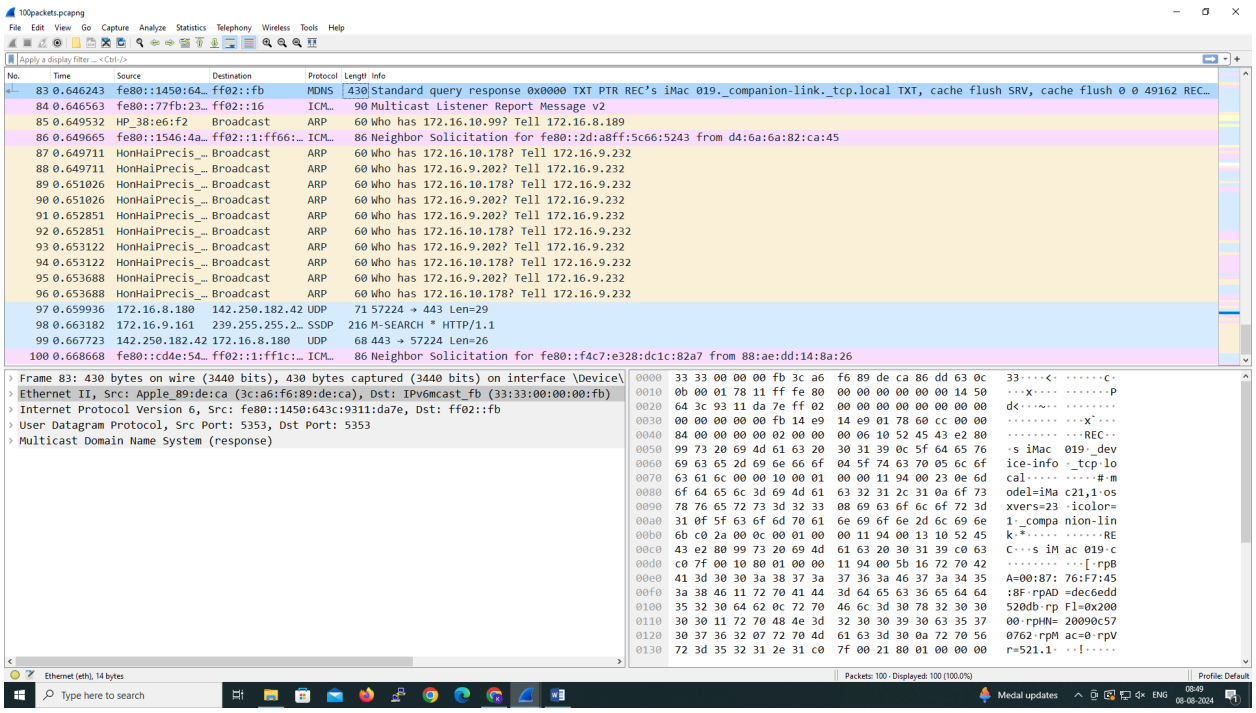
Exercises

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

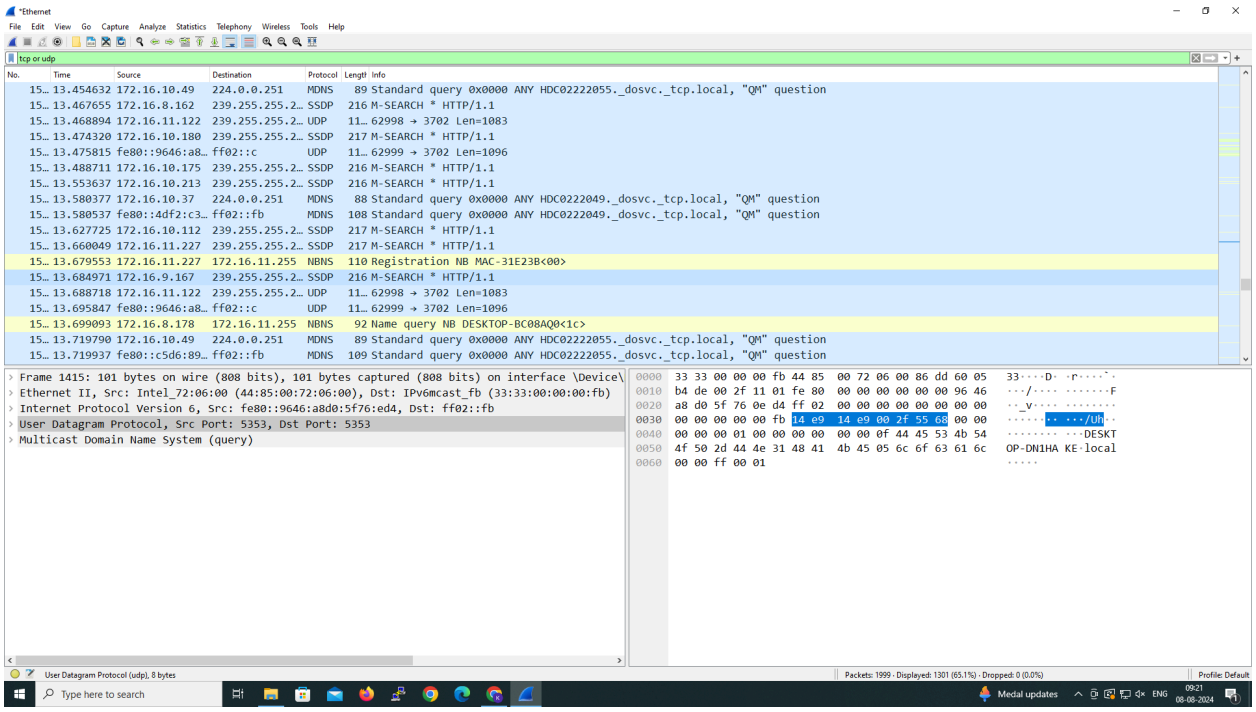


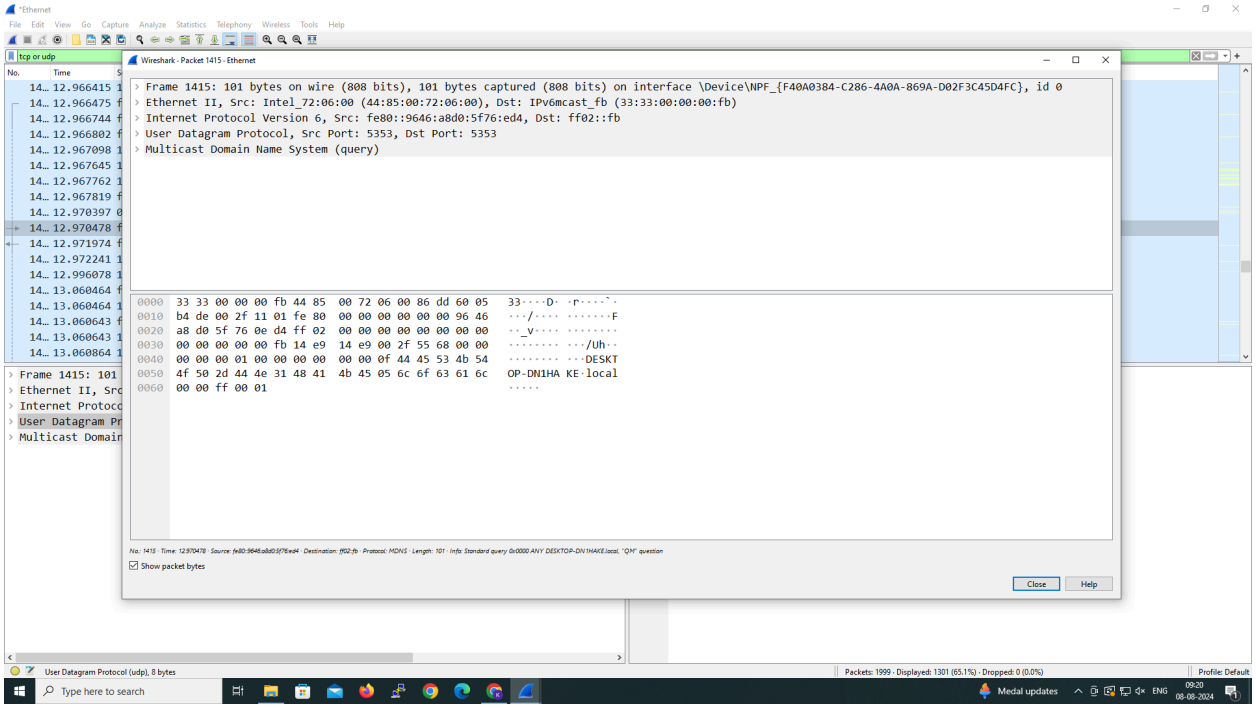
2.Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph.

Procedure

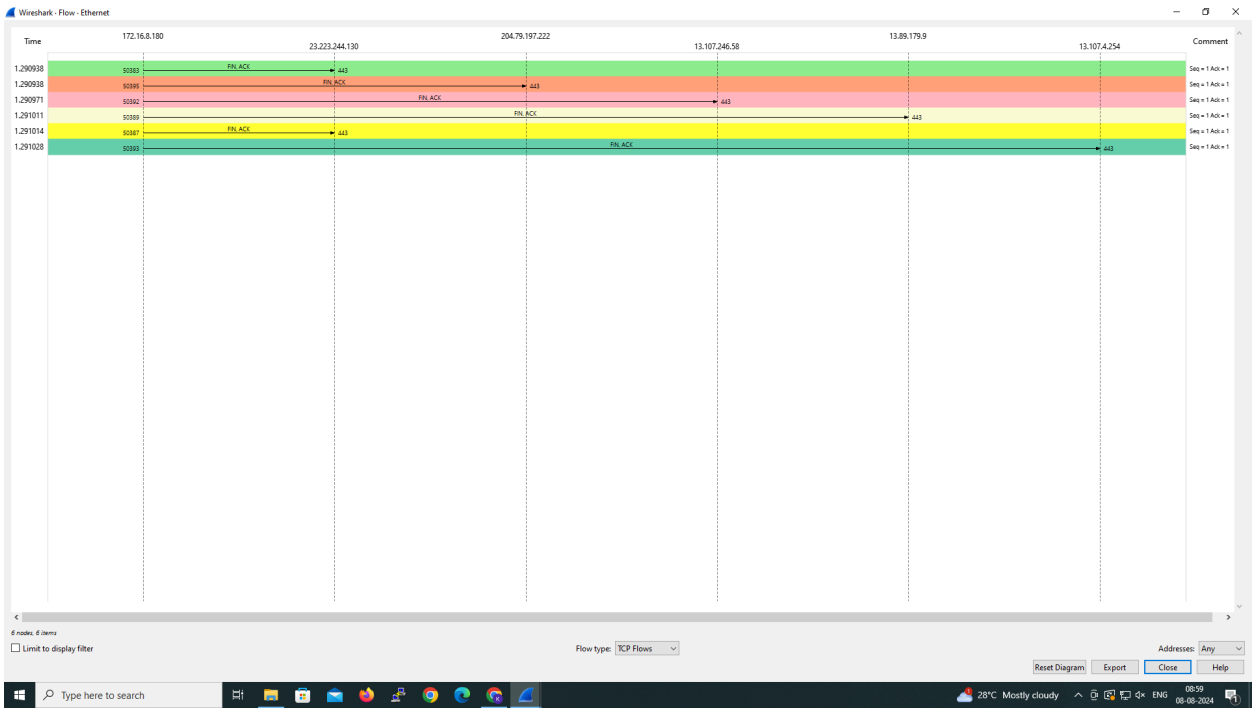
- Select Local Area Connection in Wireshark.
- Go to capture option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search TCP packets in search bar.
- To see flow graph click Statistics Flow graph.
- Save the packets.

Output:





Flow Graph output

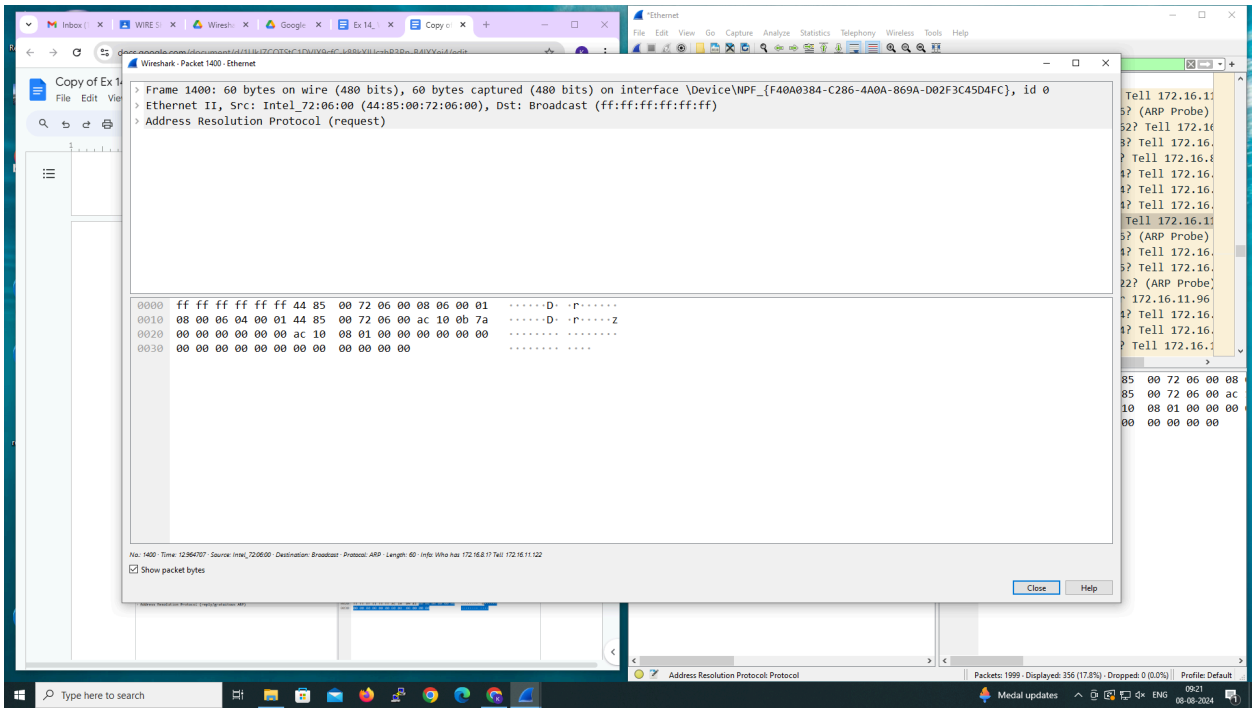


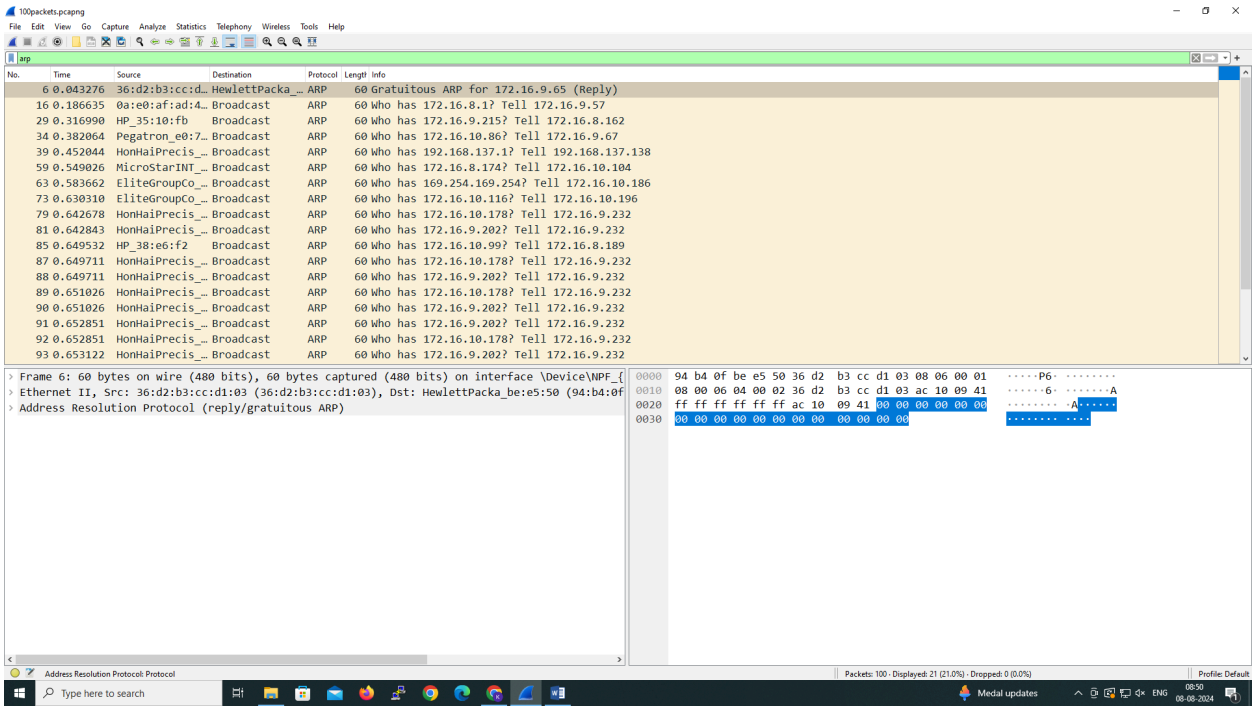
3.Create a Filter to display only ARP packets and inspect the packets.

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



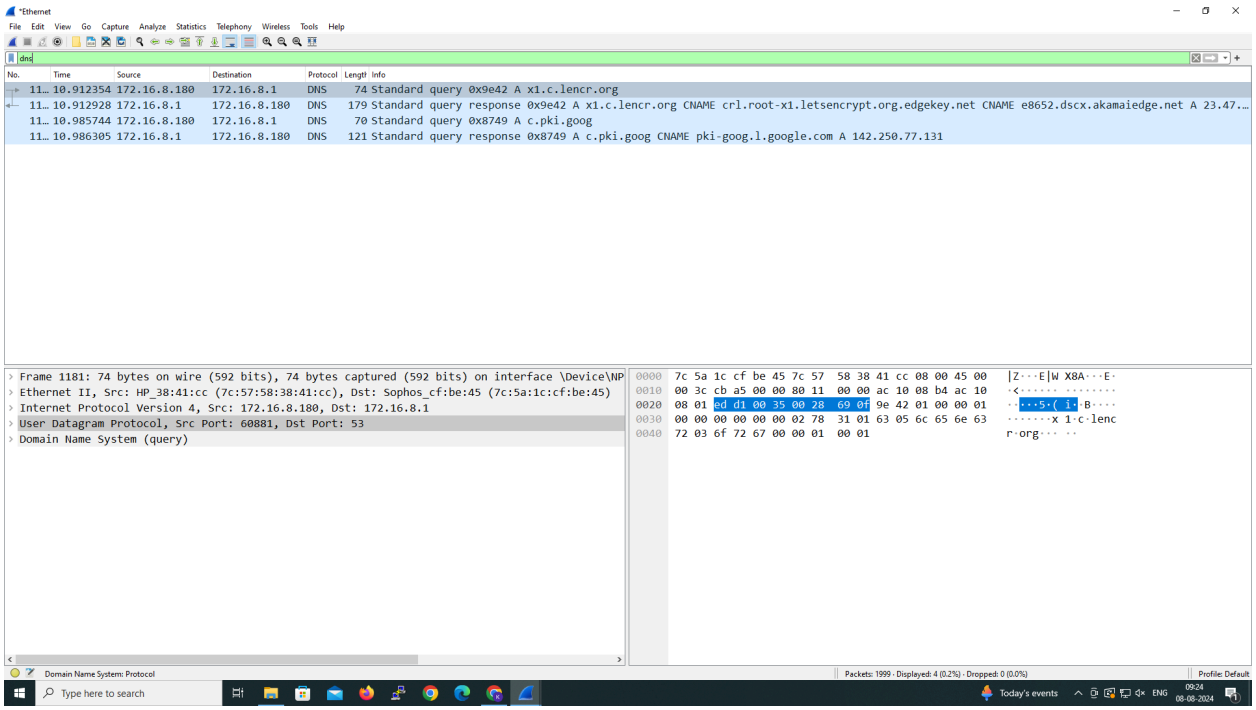


4.Create a Filter to display only DNS packets and provide the flow graph.

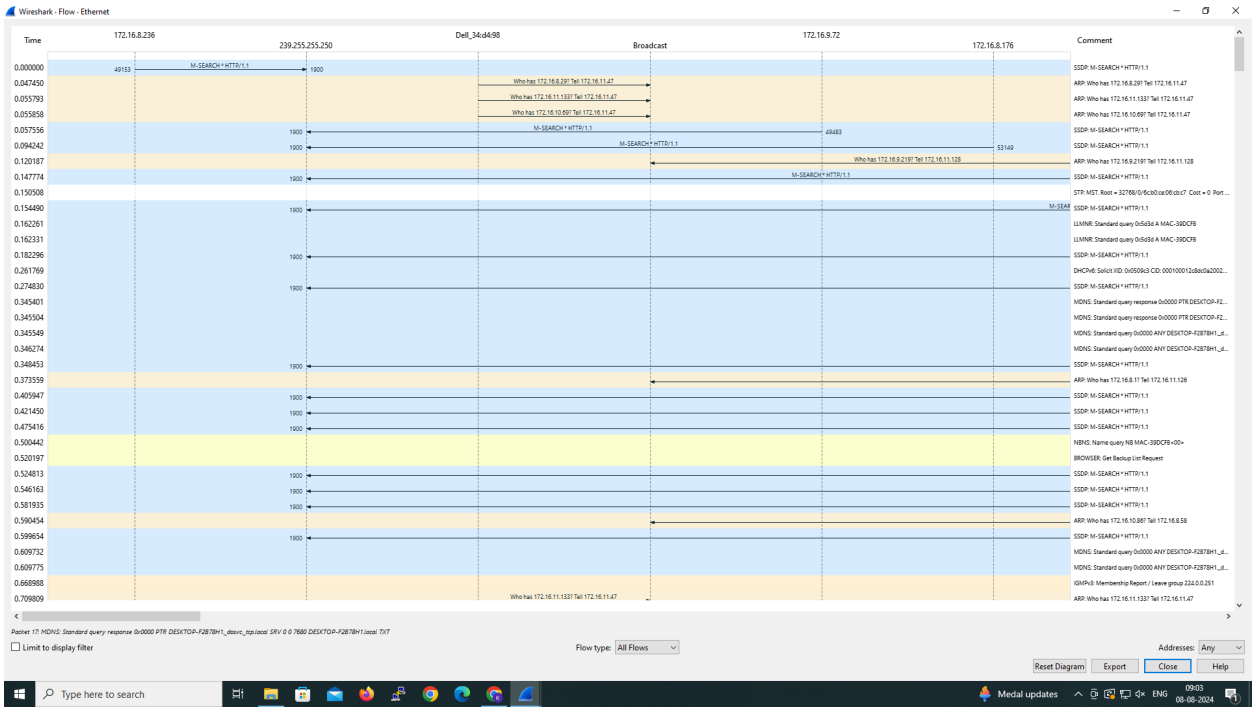
Procedure

- 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



Flow Graph output

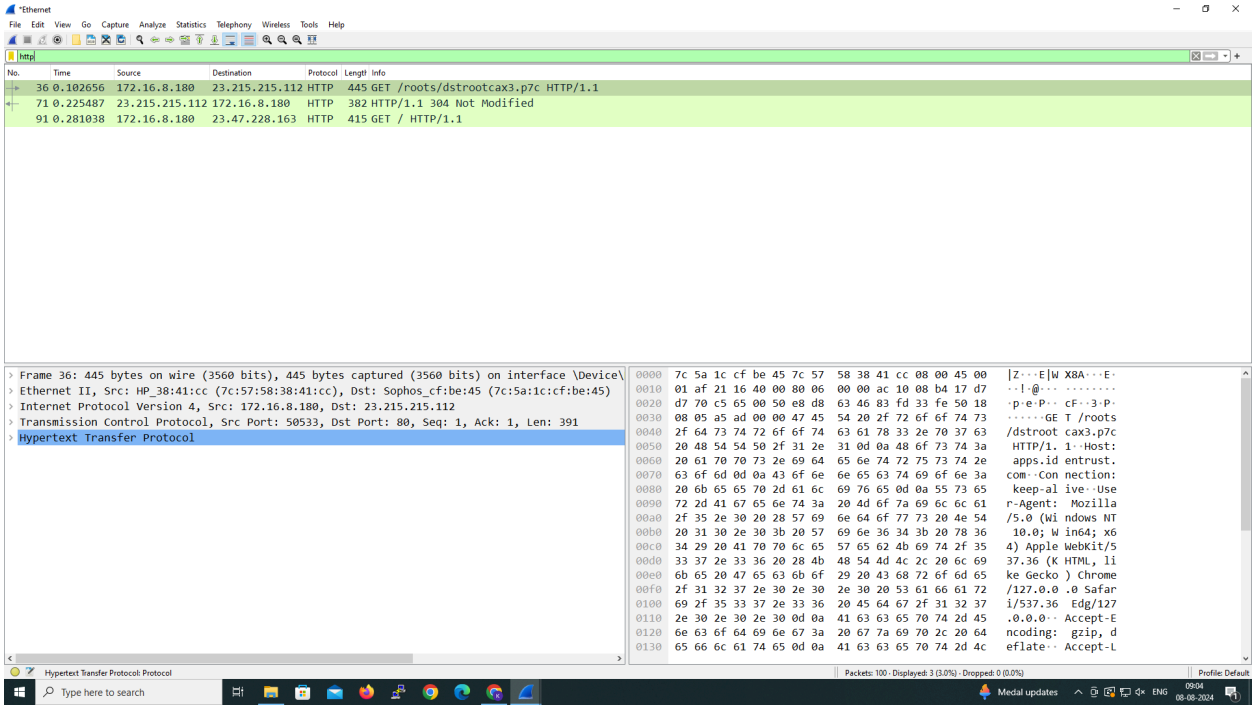


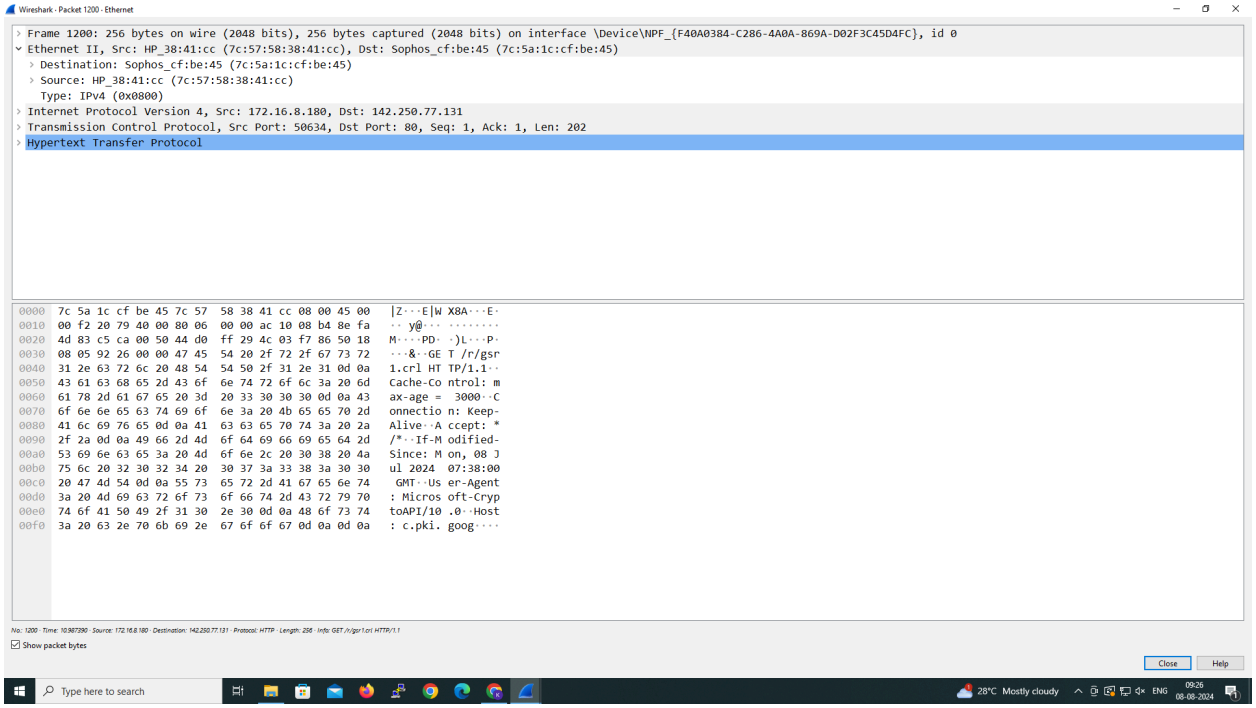
5.Create a Filter to display only HTTP packets and inspect the packets

Procedure

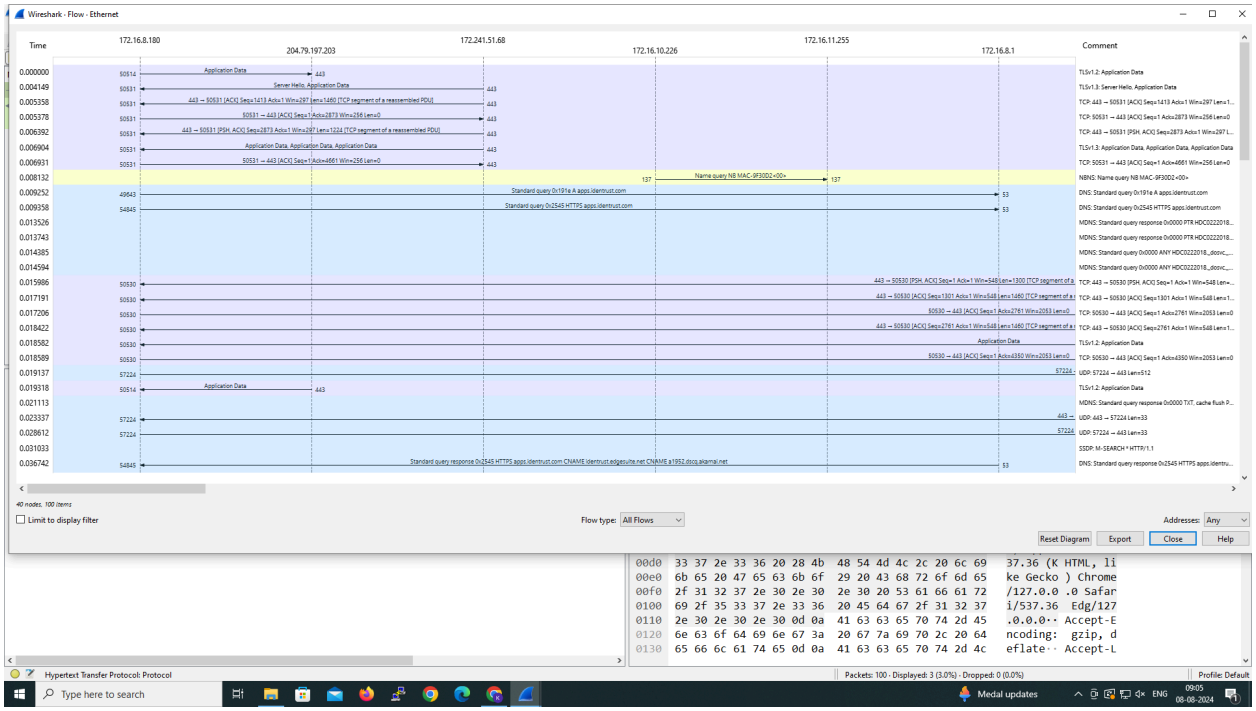
- Select Local Area Connection in Wireshark.
- Go to capture option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search HTTP packets in the search bar.
- Save the packets.

Output





Flow Graph output

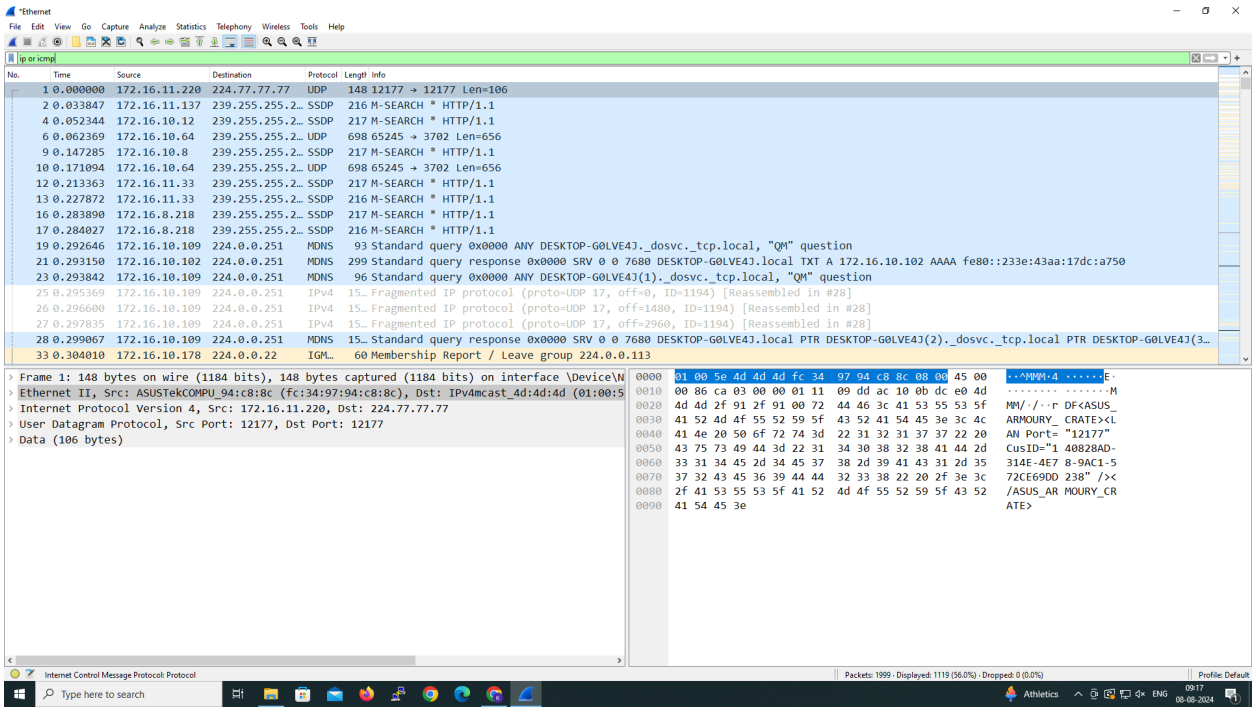


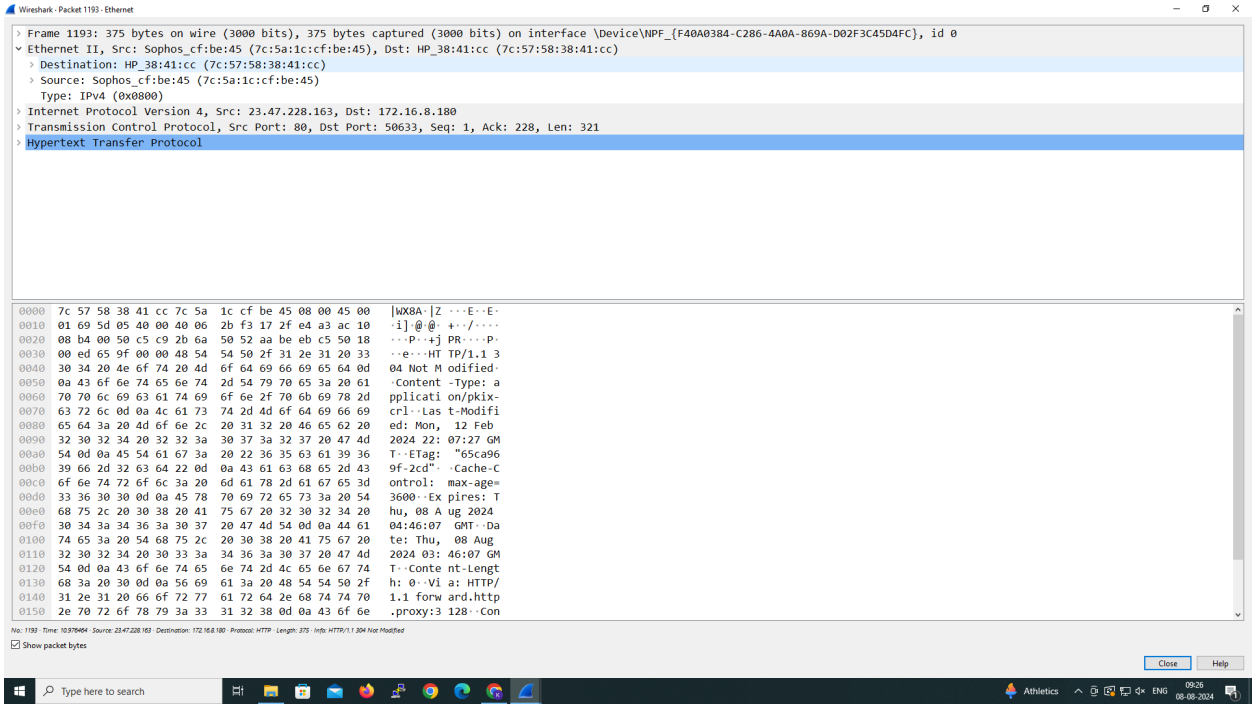
6.Create a Filter to display only IP/ICMP packets and inspect the packets.

Procedure

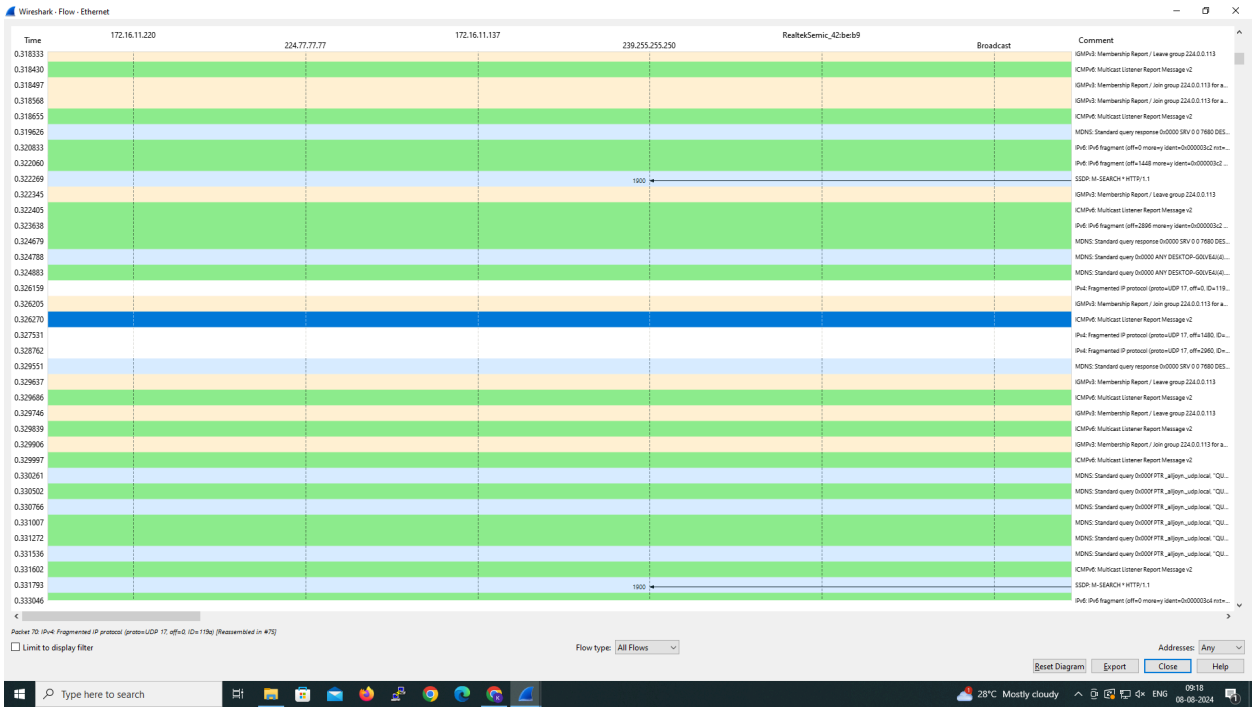
- Select Local Area Connection in Wireshark.
- Go to capture option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ICMP/IP packets in search bar.
- Save the packets

Output





Flow Graph output

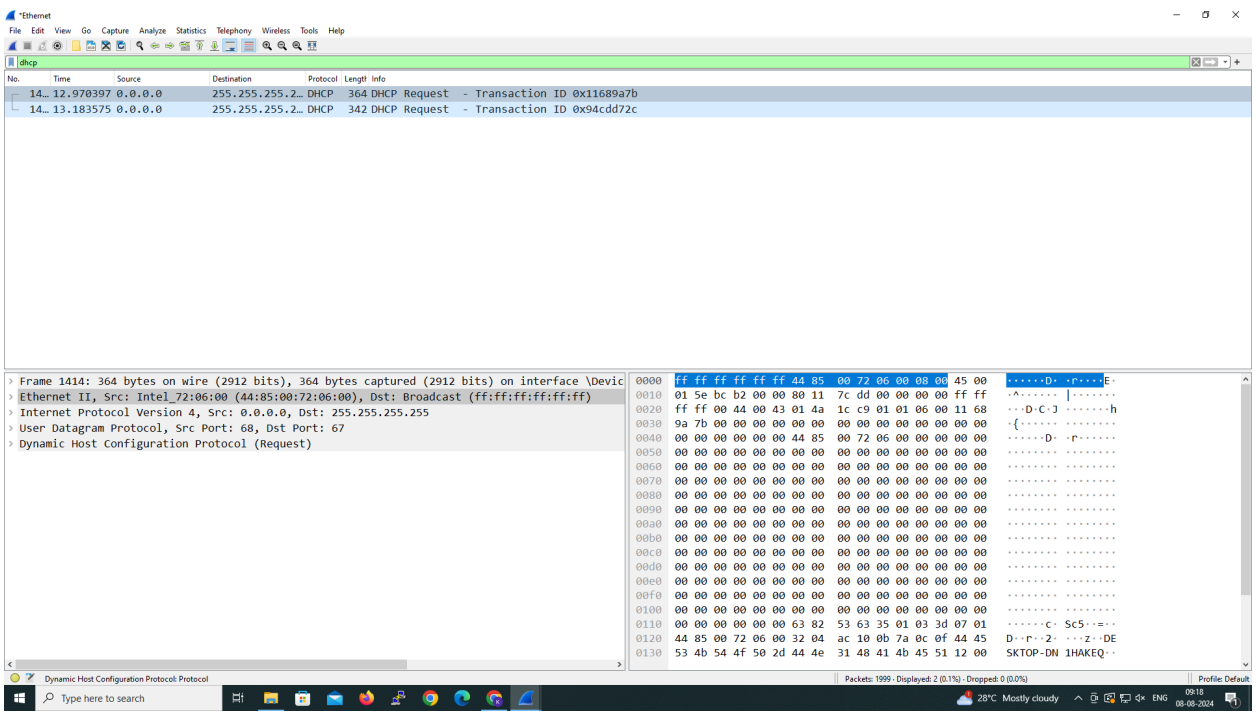


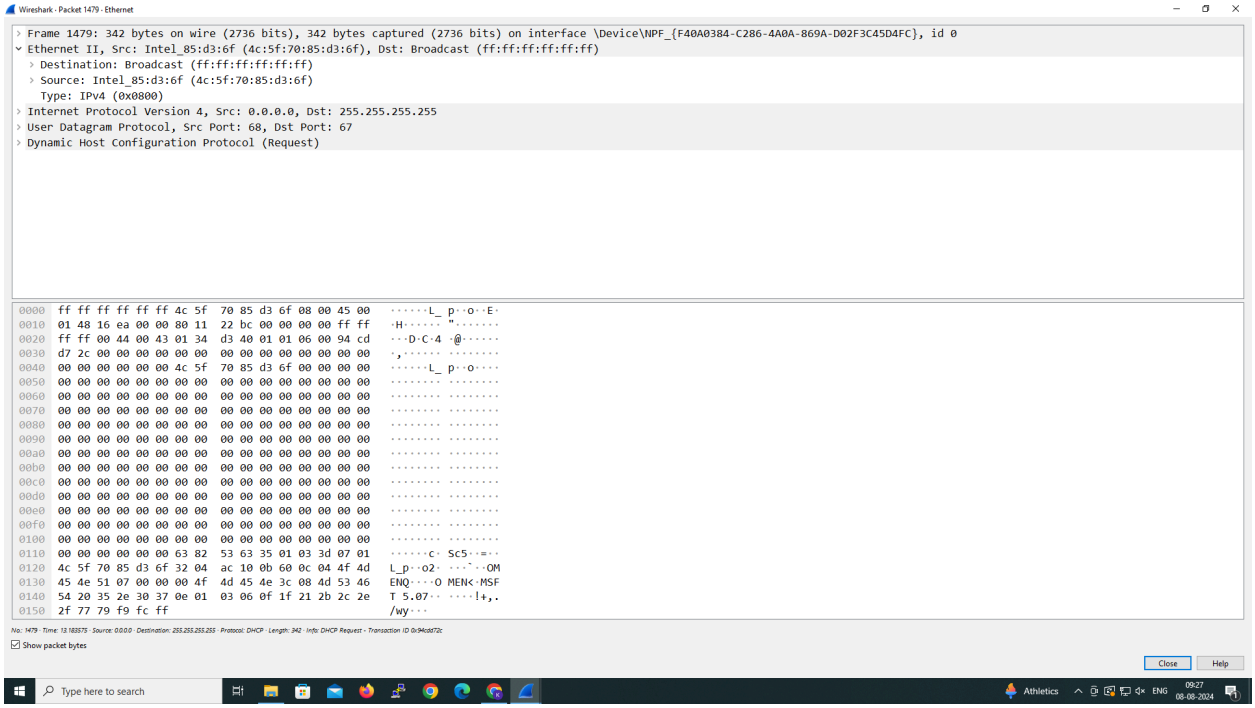
7.Create a Filter to display only DHCP packets and inspect the packets.

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




Wireshark · Packet 1479 · Ethernet
Frame 1479: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{F40A0384-C286-4A0A-869A-D02F3C45D4FC}, id 0
Ethernet II, Src: Intel_85:d3:6f (4c:5f:70:85:d3:6f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: Intel_85:d3:6f (4c:5f:70:85:d3:6f)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Request)
0000 ff ff ff ff ff ff 4c 5f 70 85 d3 6f 08 00 45 00
0010 01 48 16 ea 00 00 80 11 22 bc 00 00 00 00 ff ff
0020 ff ff 00 44 00 43 01 34 d3 40 01 01 06 00 94 cd
0030 d7 2c 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 4c 5f 70 85 d3 6f 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110 00 00 00 00 00 00 63 82 53 63 35 01 03 3d 07 01
0120 4c 5f 70 85 d3 6f 32 04 ac 10 0b 60 0c 04 4f 4d
0130 45 4e 51 07 00 00 00 4f 4d 45 4e 3c 08 4d 53 46
0140 54 20 35 2e 30 37 0e 01 03 06 0f 1f 21 2b 2c 2e
0150 2f 77 79 f9 fc ff
Show packet bytes
Close
Help
Type here to search
Athletics
ENG
09:27
08-08-2024