

**231901021**

**H.Kavya Varshini**

**Date-19/8/2024**

**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:**

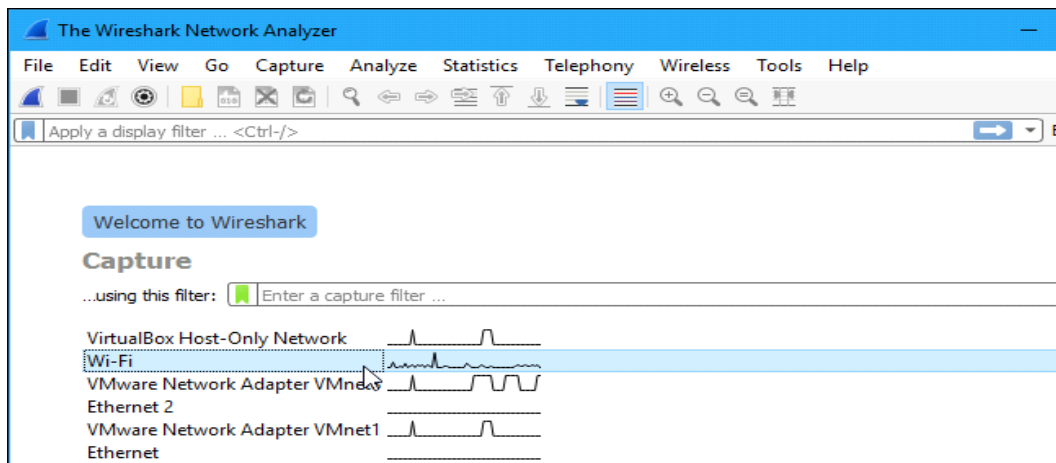
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

**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.

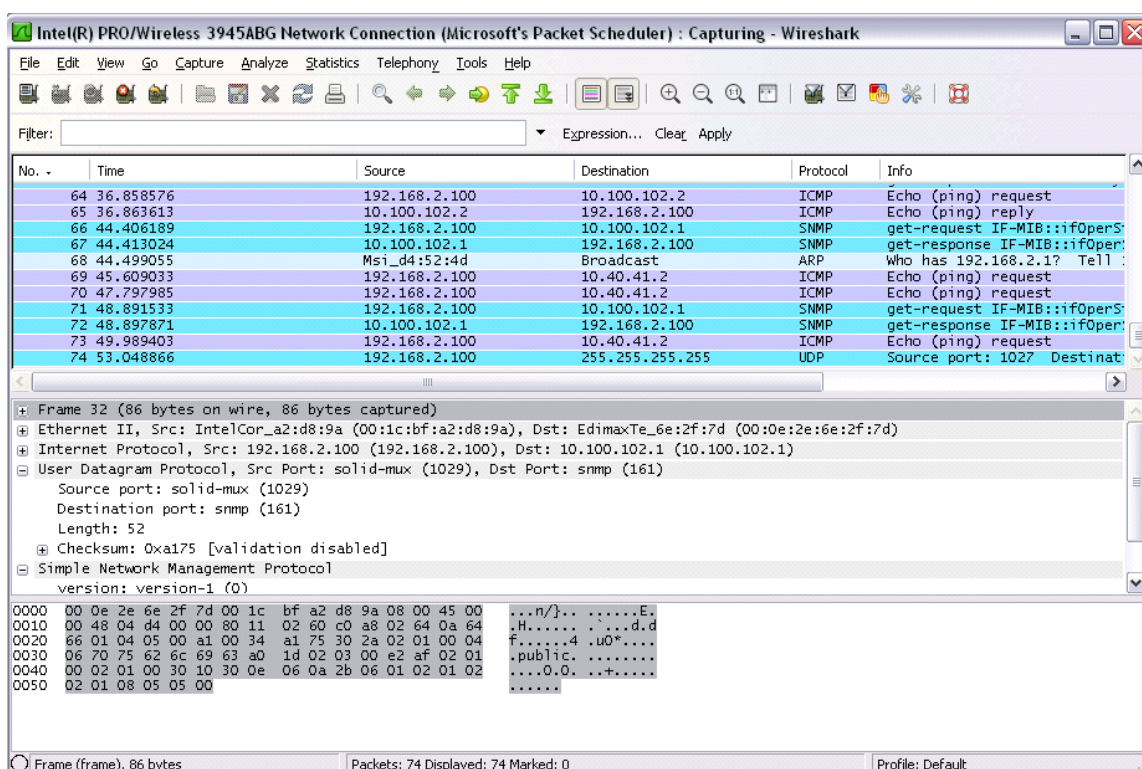
**Capturing Packets**

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.

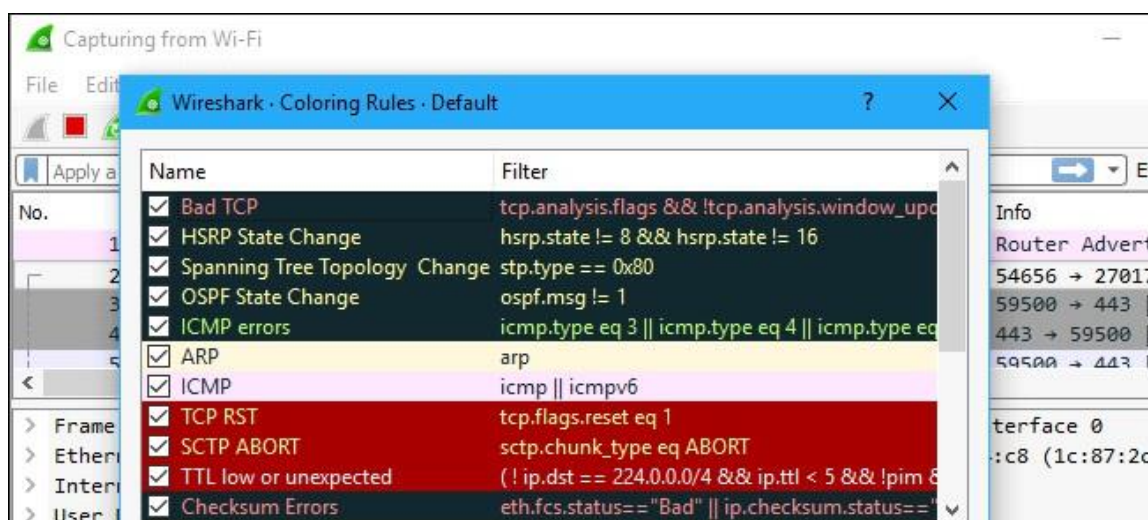
## The “Packet Bytes” Pane

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.

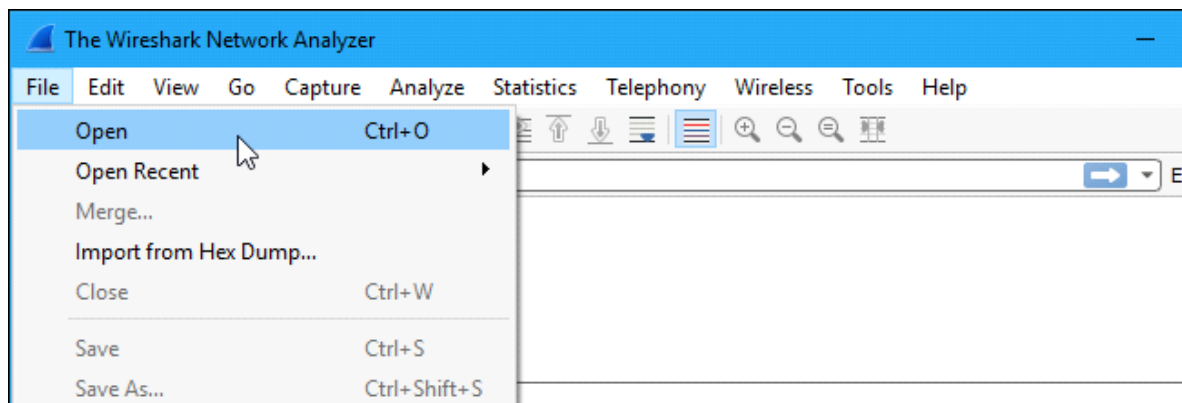
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



## 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.

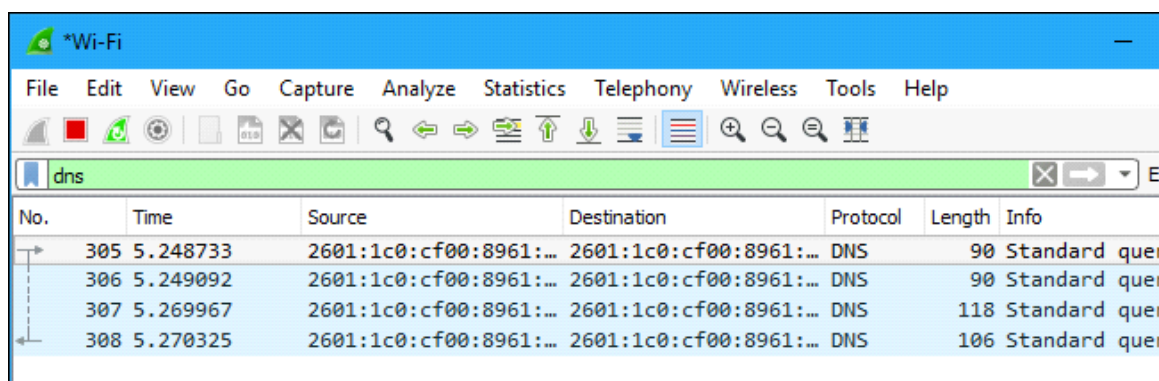
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



## Filtering Packets

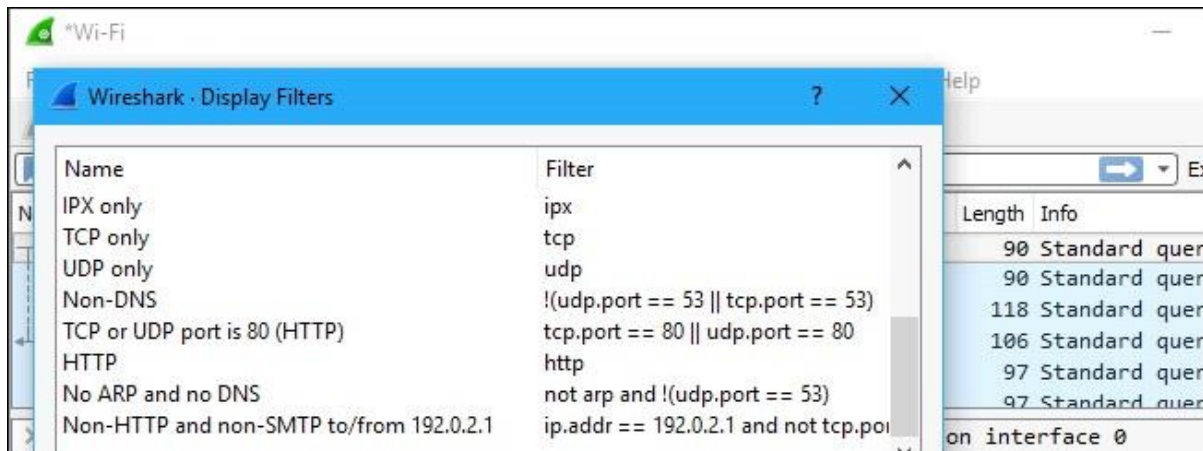
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters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



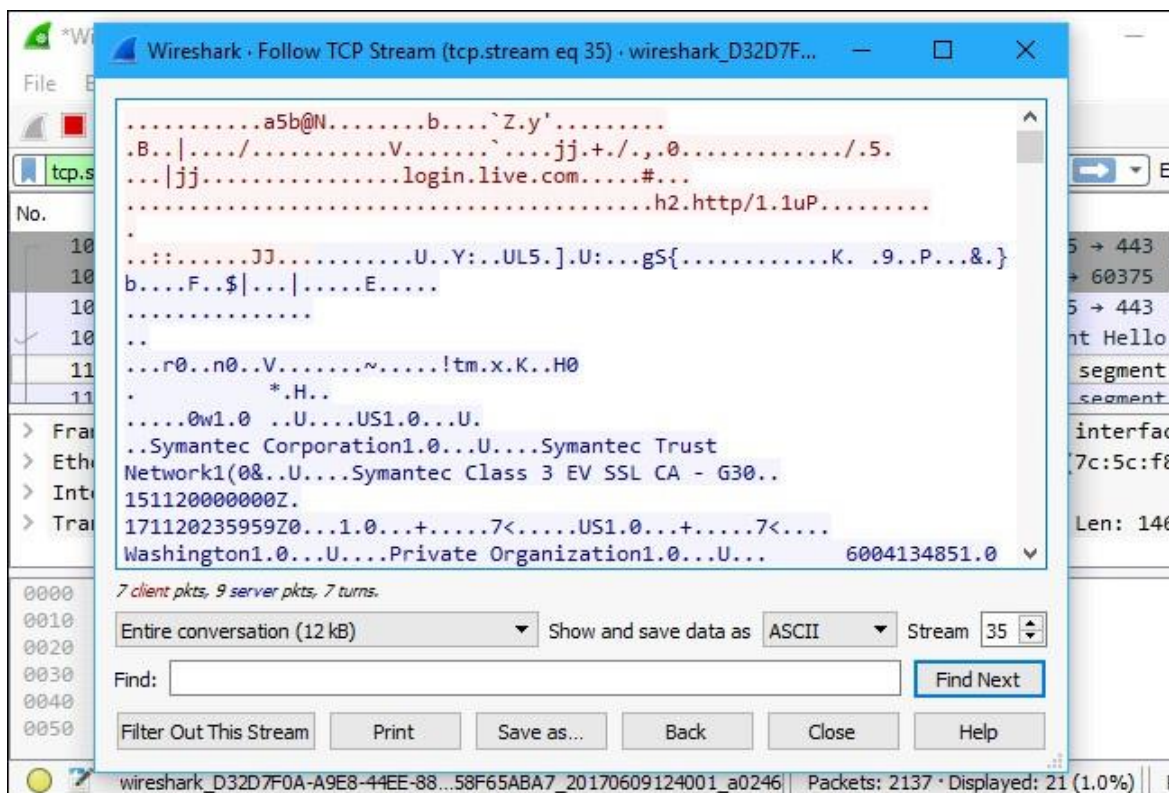
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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.



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.

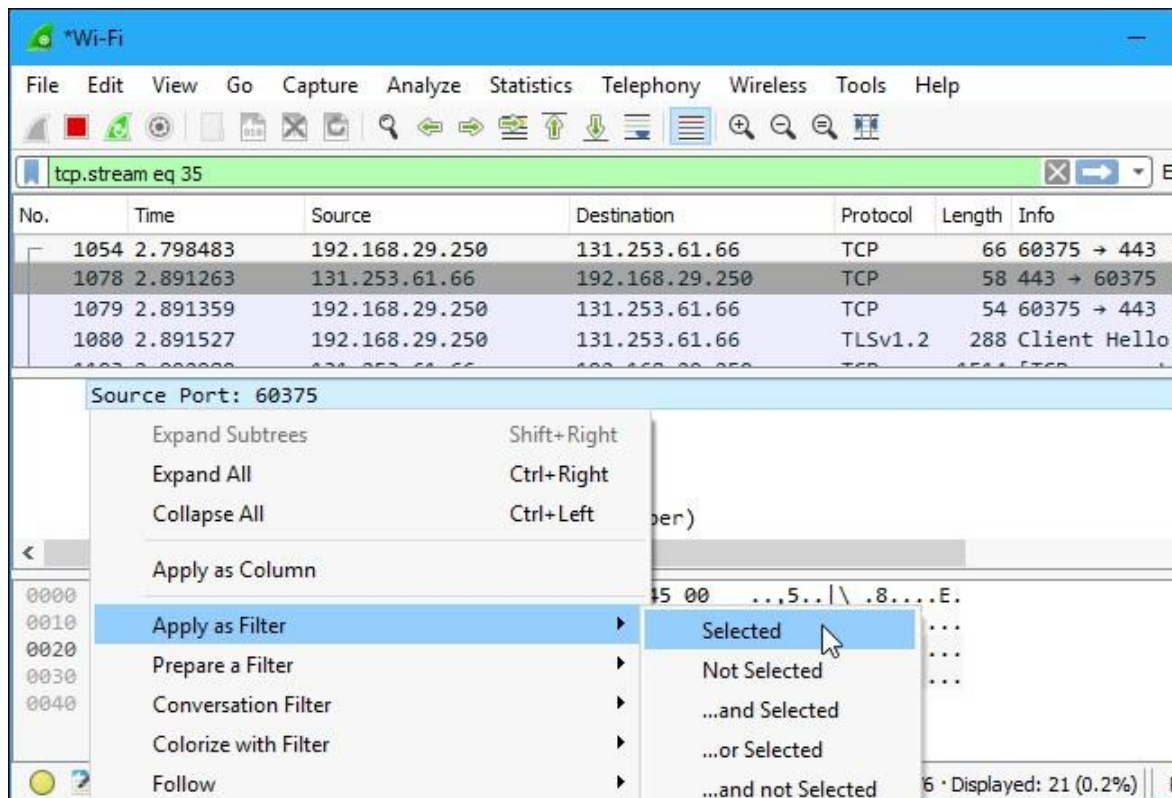
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

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.]@... 0.....
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	..

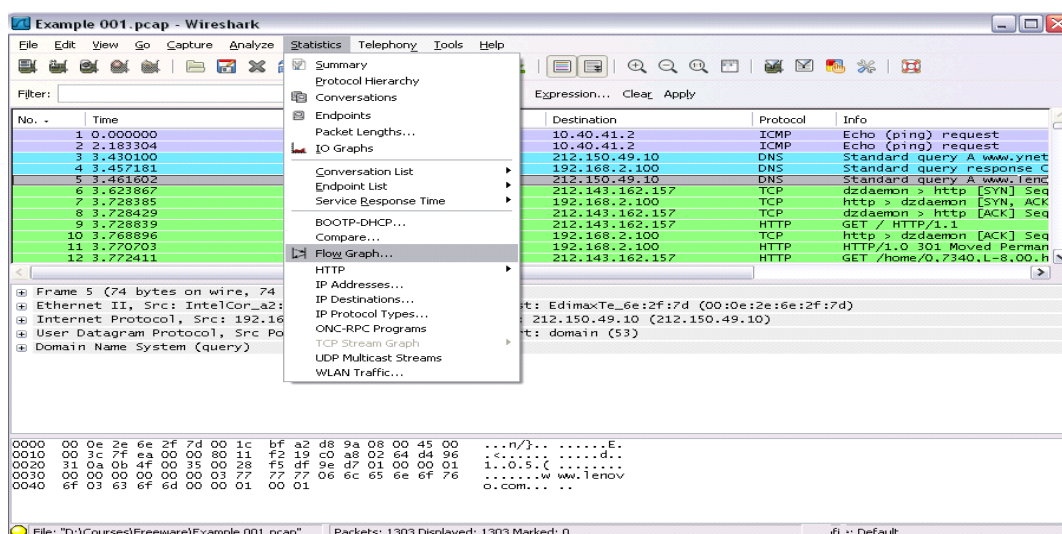
Encapsulation type (frame.encap\_type) | Packets: 8136 · Displayed: 21 (0.3%)

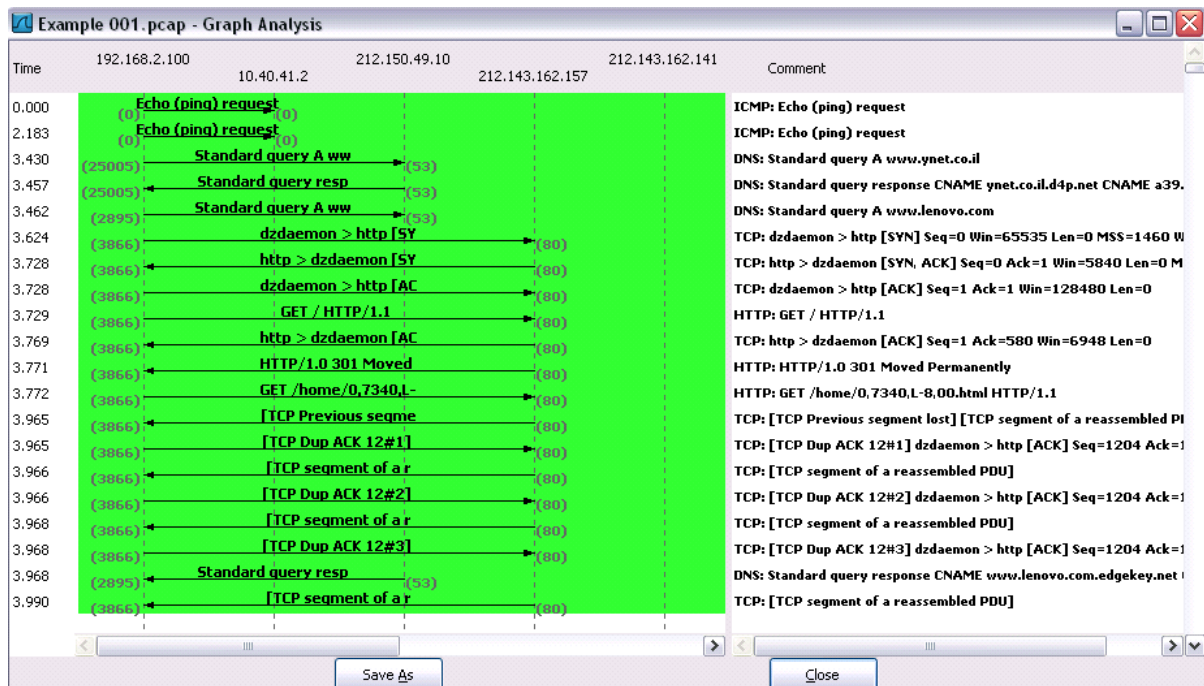
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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: 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

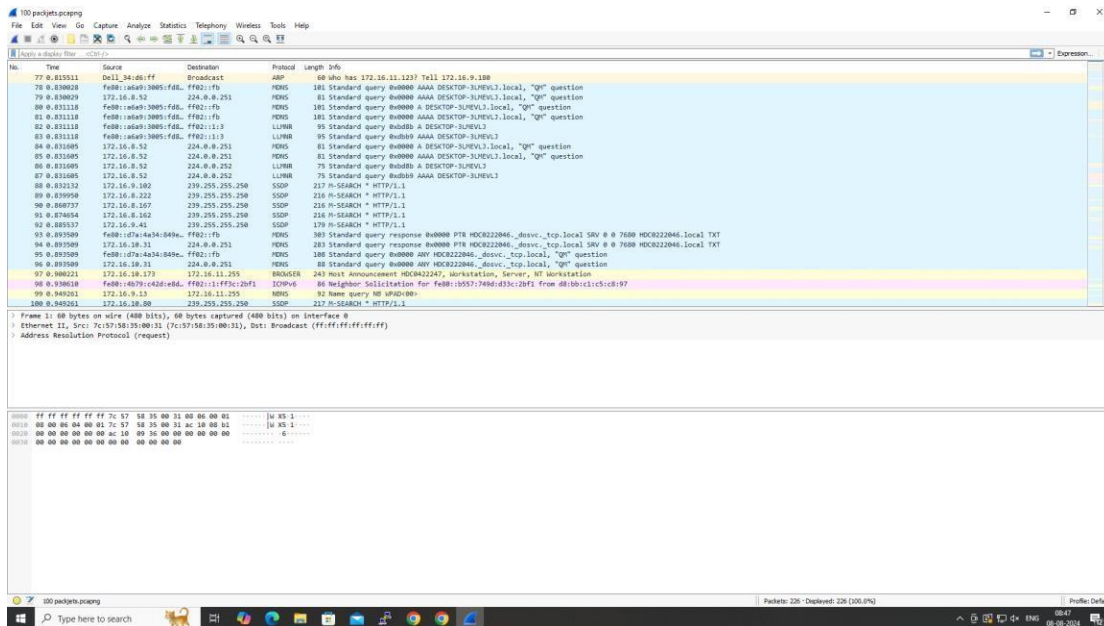
1. Capture 100 packets from the Ethernet: IEEE 802.3 LAN Interface and save it.

### 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



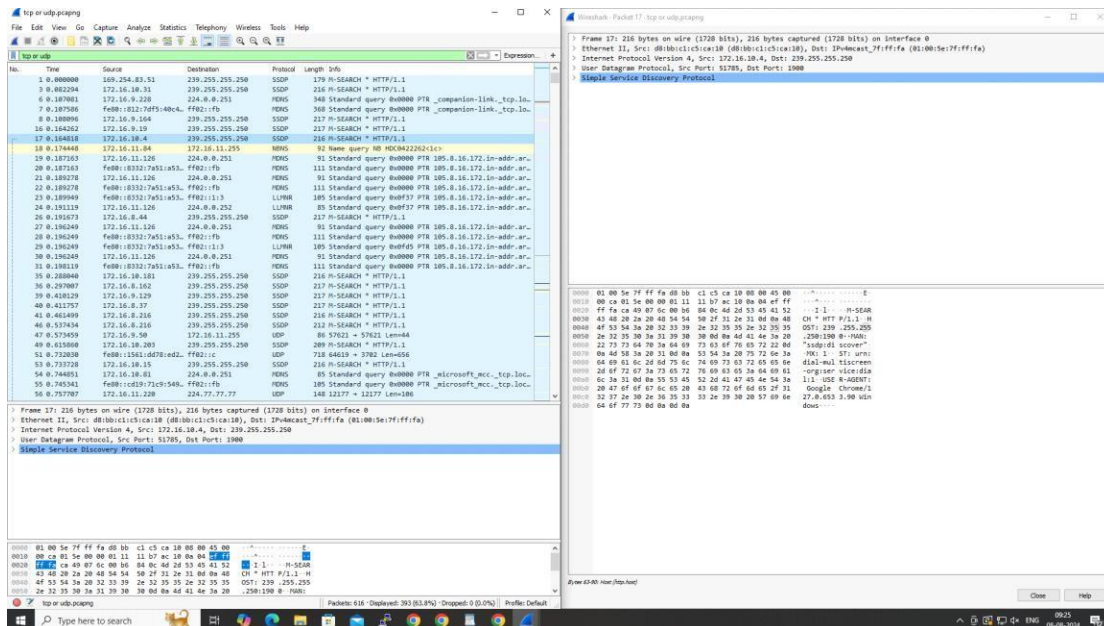


**2. Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph.**

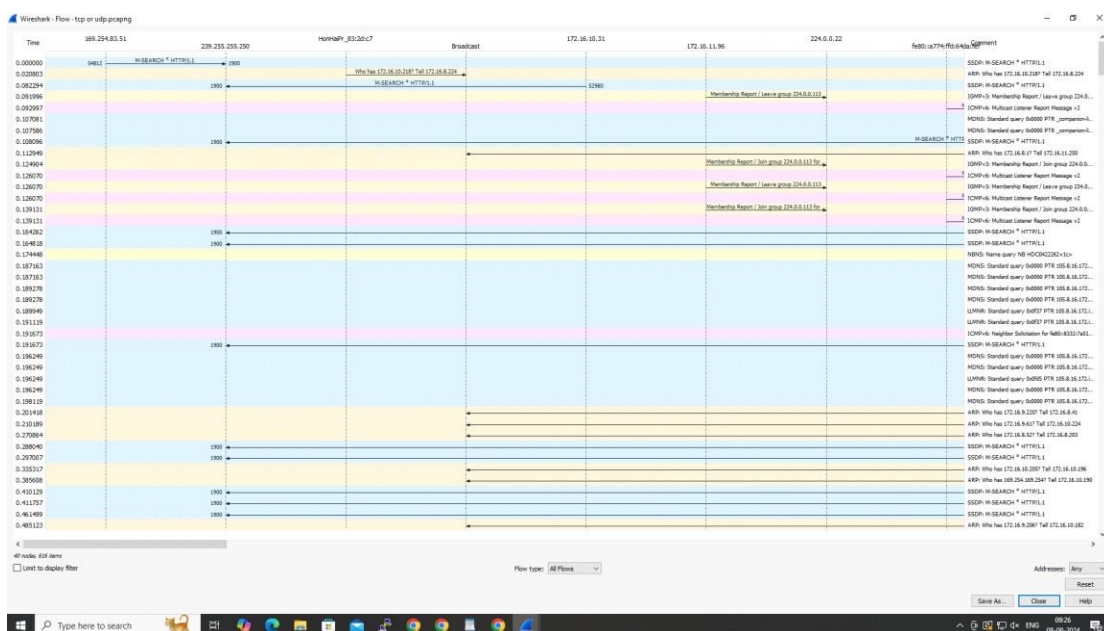
## Procedure

- 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:



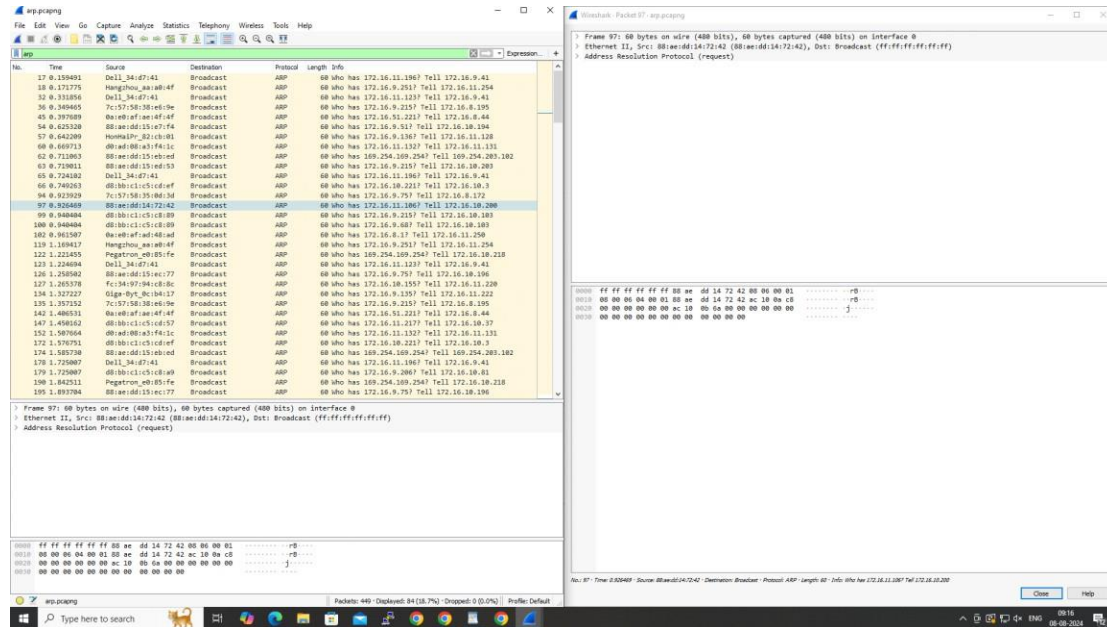
## 3. Create a Filter to display only ARP packets and inspect the packets.

### 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

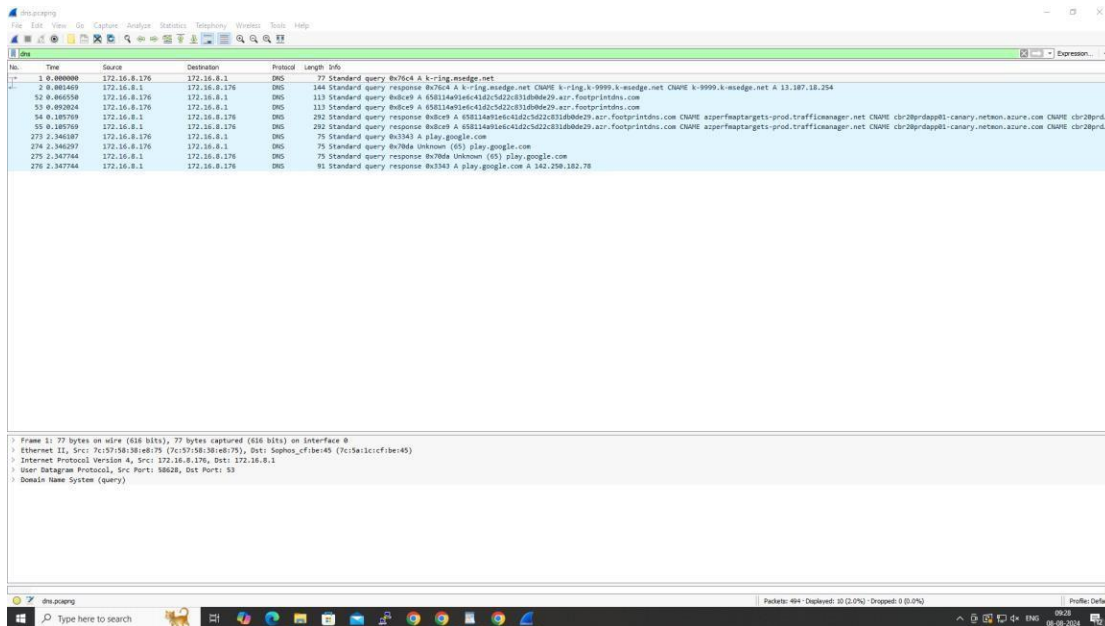


## 4. Create a Filter to display only DNS packets and provide the flow graph.

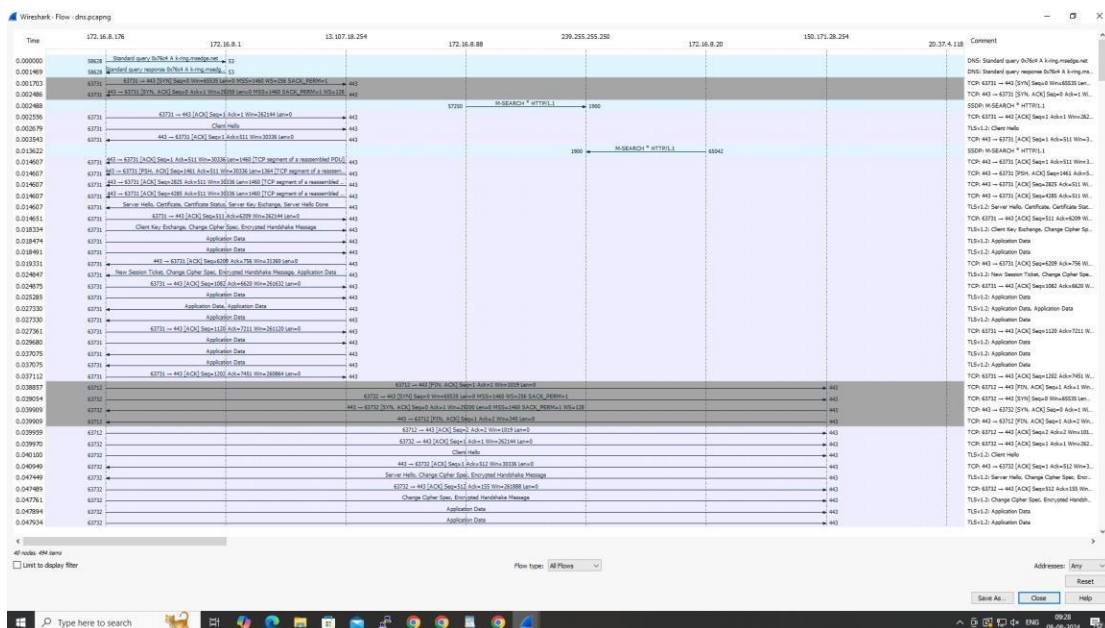
### Procedure

- 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:


The image displays a Wireshark packet capture window. The top pane shows a list of captured packets, with packet 2317 selected. The middle pane shows the details of the selected packet, which is an HTTP GET request for '/edgedl/release2/chrome\_component/ac41h5au7f123174h...'. The bottom pane shows the raw packet data in hexadecimal and ASCII. The status bar at the bottom indicates that the packet is a GET request for a resource that is not satisfiable.

# Flow Graph output:

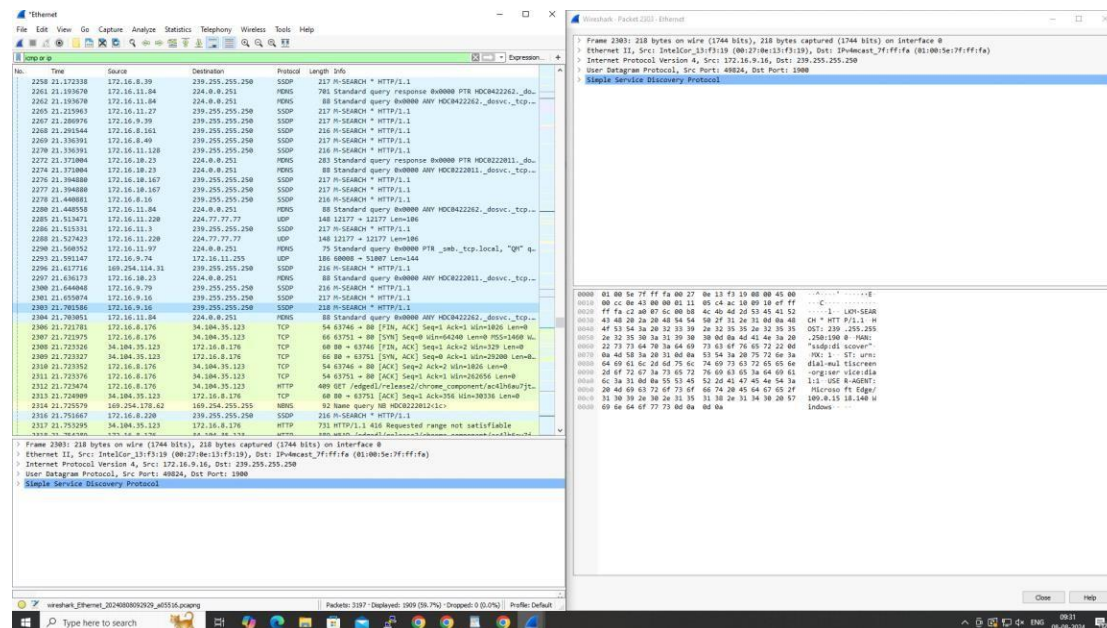
The image displays a Wireshark Flow Graph window. The top pane shows a list of captured packets, with packet 2317 selected. The middle pane shows the details of the selected packet, which is an HTTP GET request for '/edgedl/release2/chrome\_component/ac41h5au7f123174h...'. The bottom pane shows the raw packet data in hexadecimal and ASCII. The status bar at the bottom indicates that the packet is a GET request for a resource that is not satisfiable.

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

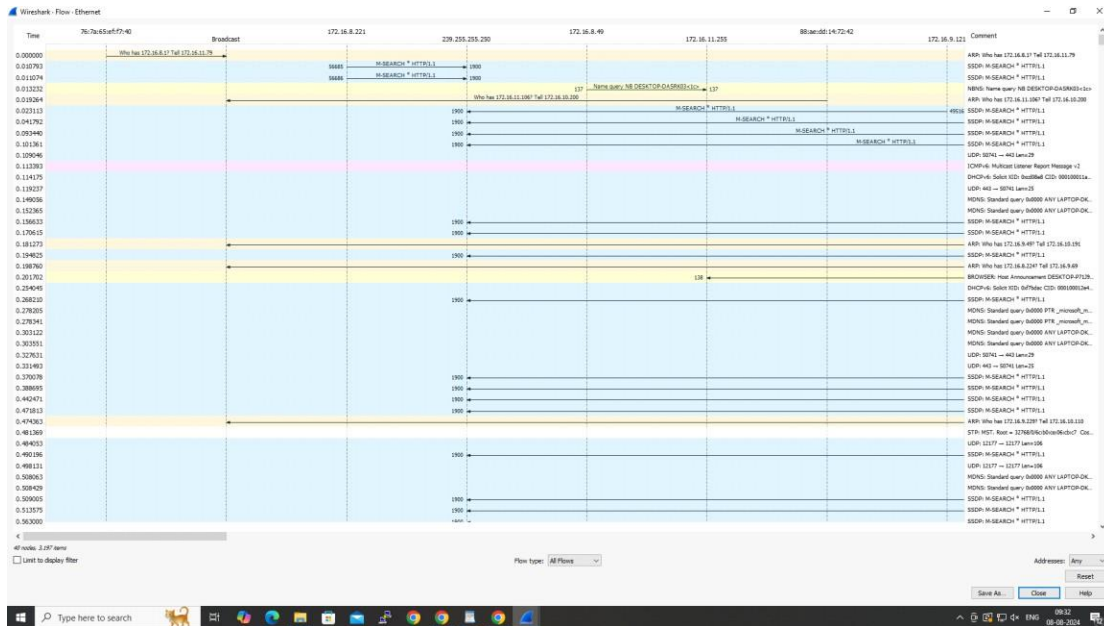
### Procedure

- 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:

The screenshot displays the Wireshark network protocol analyzer. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Windows, Tools, and Help. Below the menu is a toolbar with icons for common actions like opening files, saving, zooming, and filtering.

The main window is divided into three panes:

- Packets List Pane (Left):** Shows a list of captured packets. Packet 173 is selected, which is an Internet Protocol Version 4 packet from 172.16.0.1 to 255.255.255.255.
- Packet Details Pane (Middle):** Provides a hierarchical view of the selected packet's structure. It shows Ethernet II, Internet Protocol Version 4, and Dynamic Host Configuration Protocol (DHCP) fields.
- Packet Bytes Pane (Right):** Displays the raw hexadecimal and ASCII data of the selected packet. The hex data starts with 0000 ff ff ff ff ff ff 20 c1 90 8e 3f e1 00 00 45 00, representing the Ethernet frame header.

At the bottom status bar, it indicates the current file is "wireshark\_ethernet\_202408080920\_405556.pcapng", with statistics showing 1387 packets displayed, 0 dropped, and 0% profile default.