

**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:**

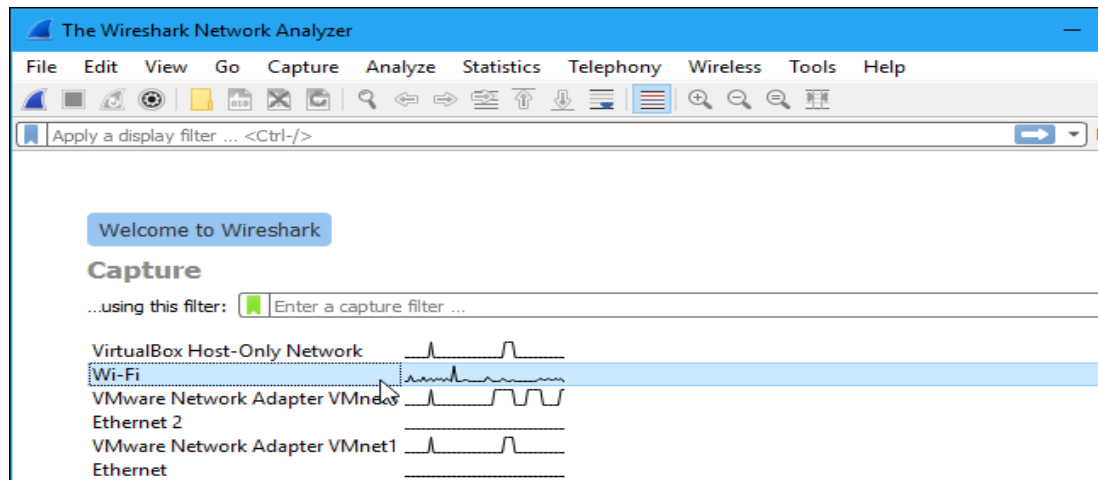
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

**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.

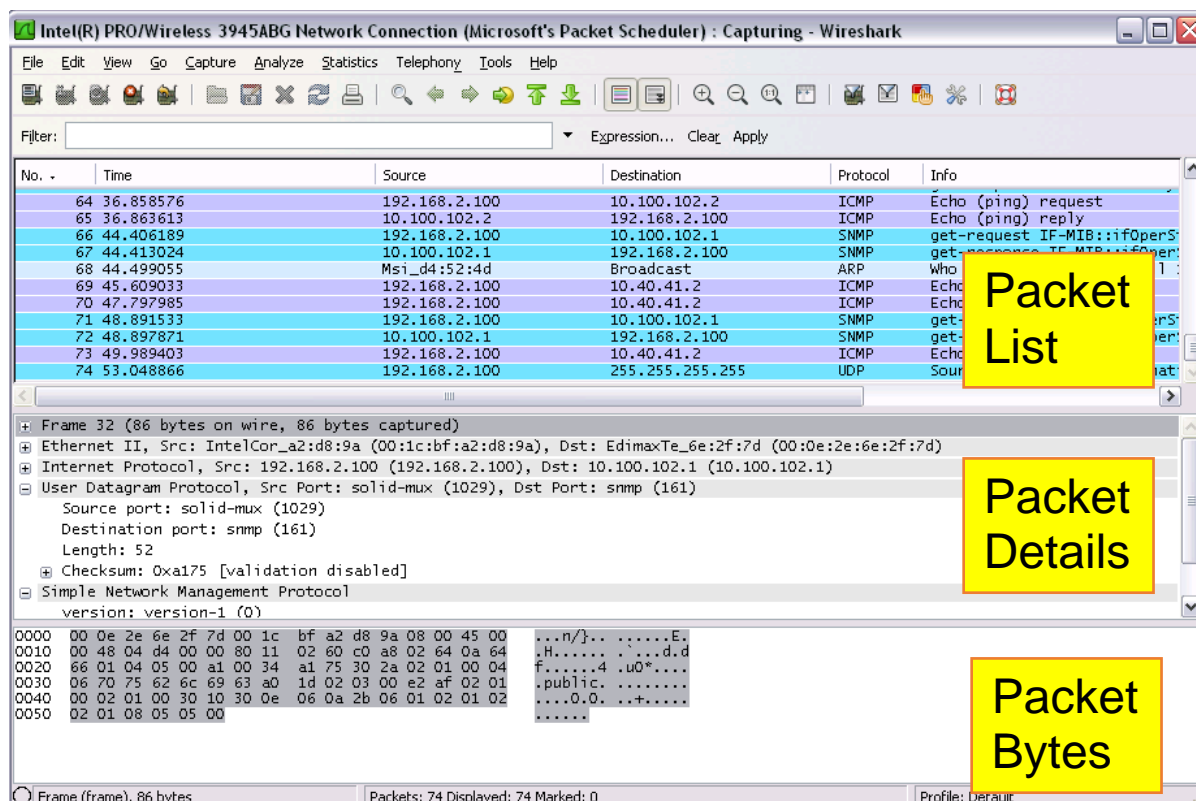
**Capturing Packets**

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.

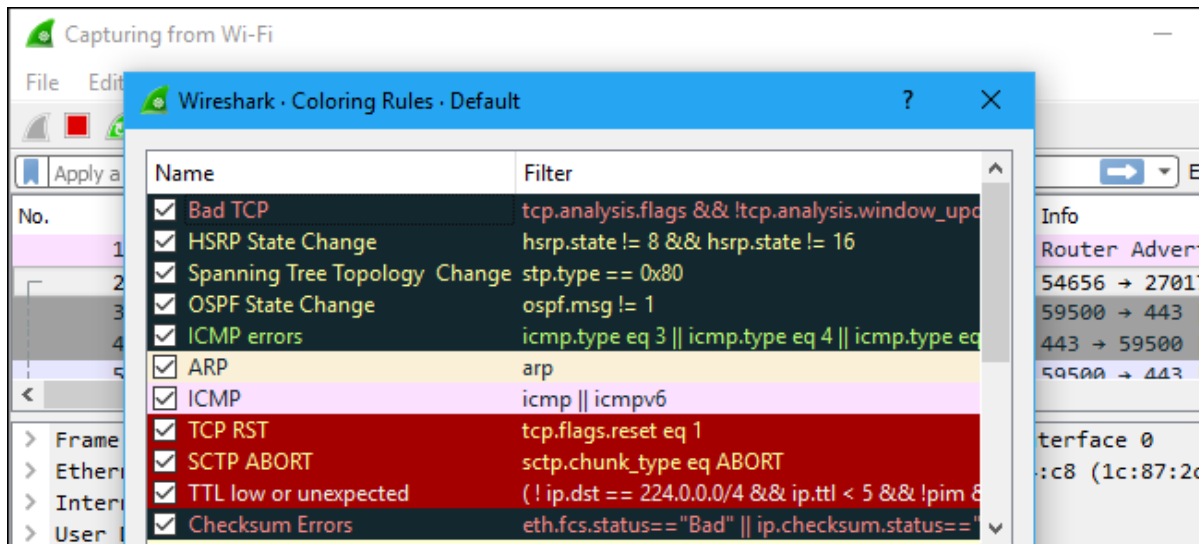
### **The “Packet Bytes” Pane**

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.

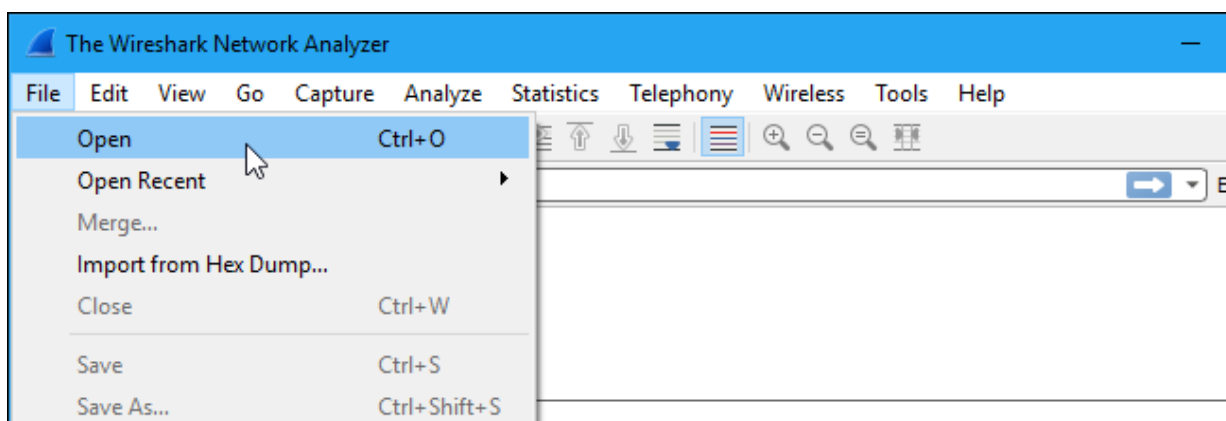
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



## 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.

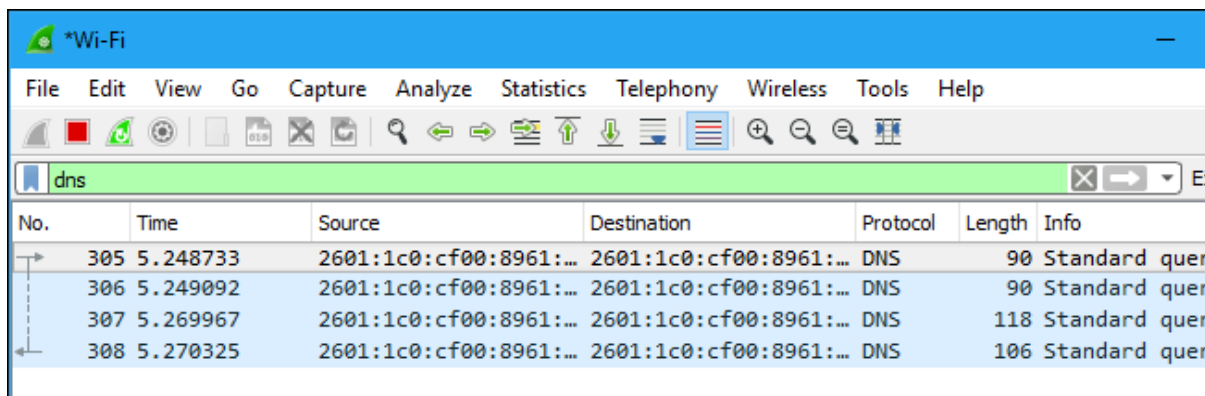


## Filtering Packets

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

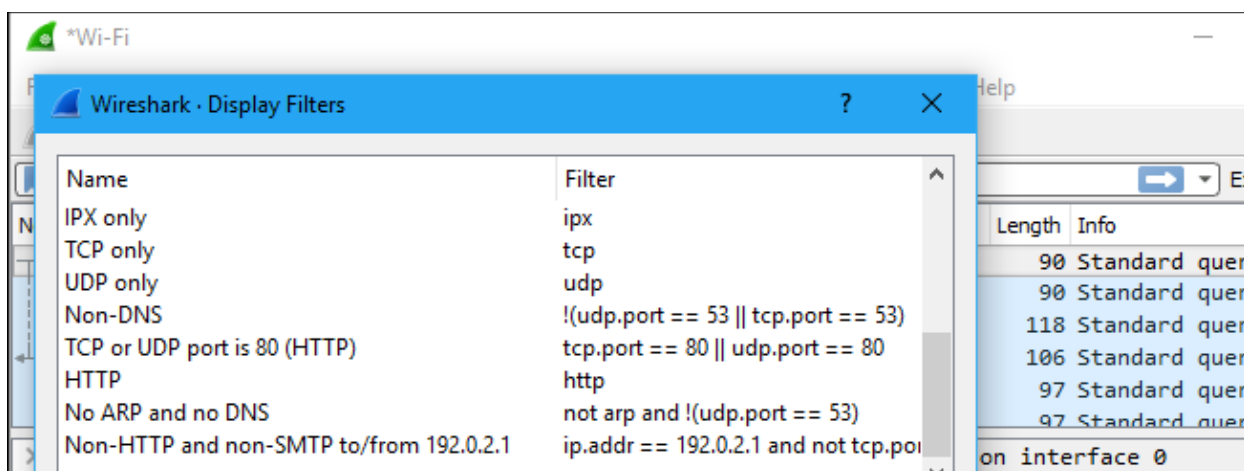
traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

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.



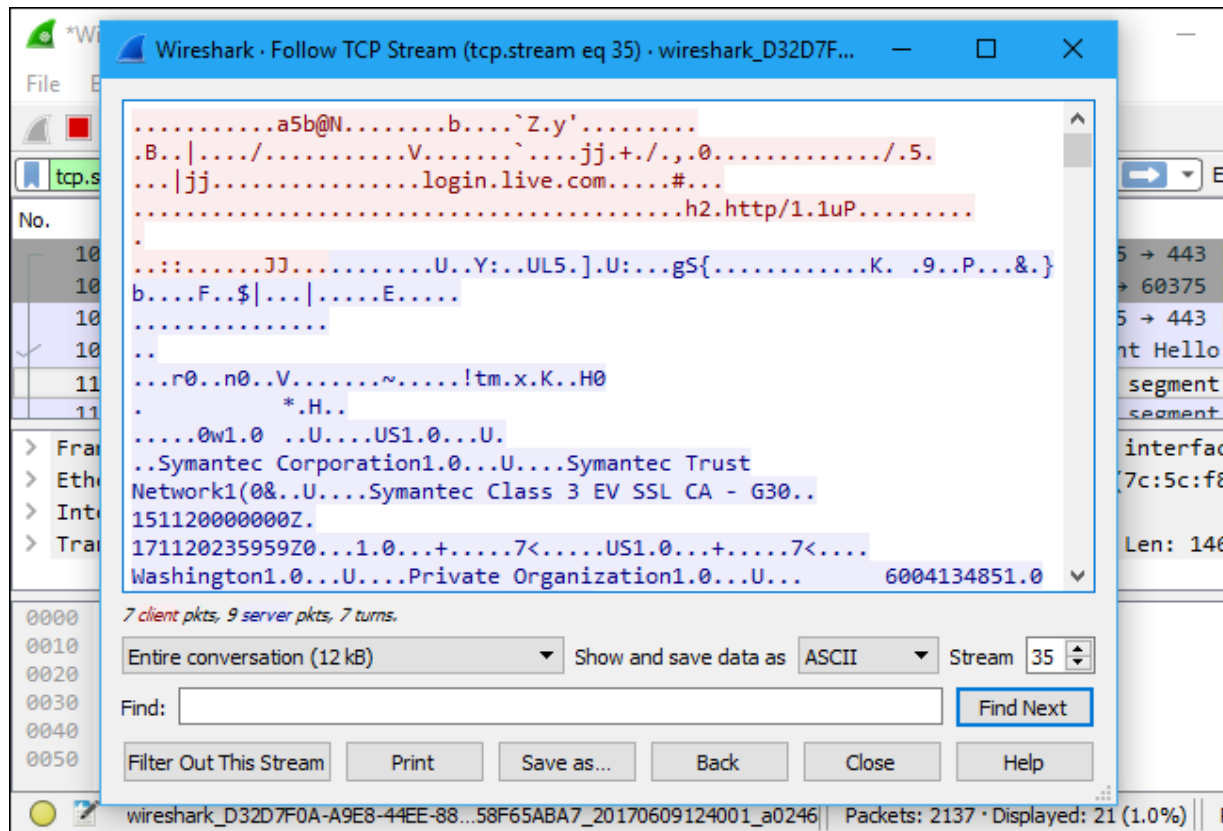
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

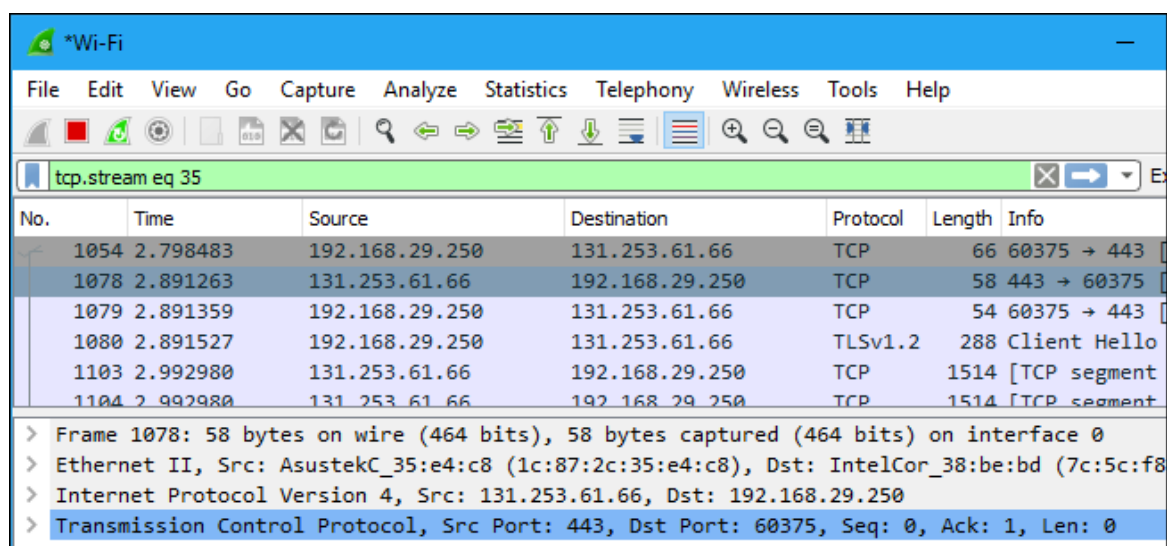


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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.

\*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 35

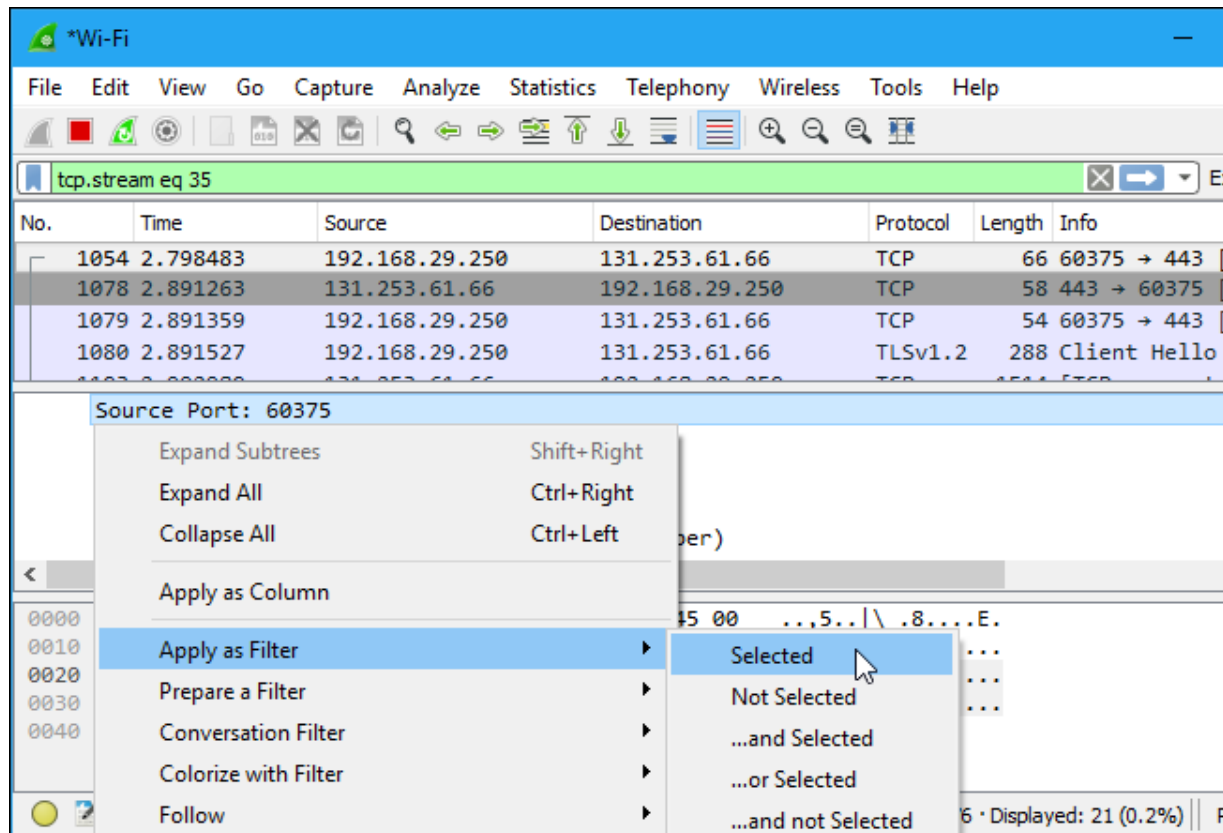
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

▼ 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

Offset	Hex	ASCII
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%) |

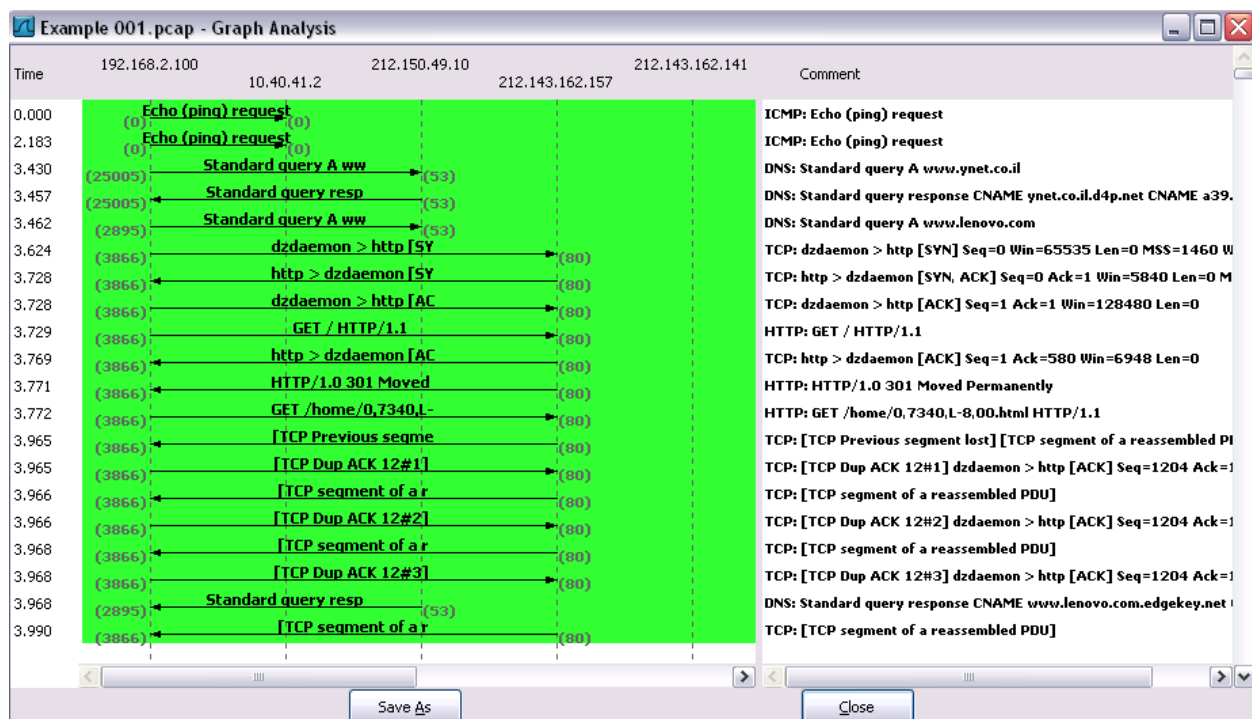
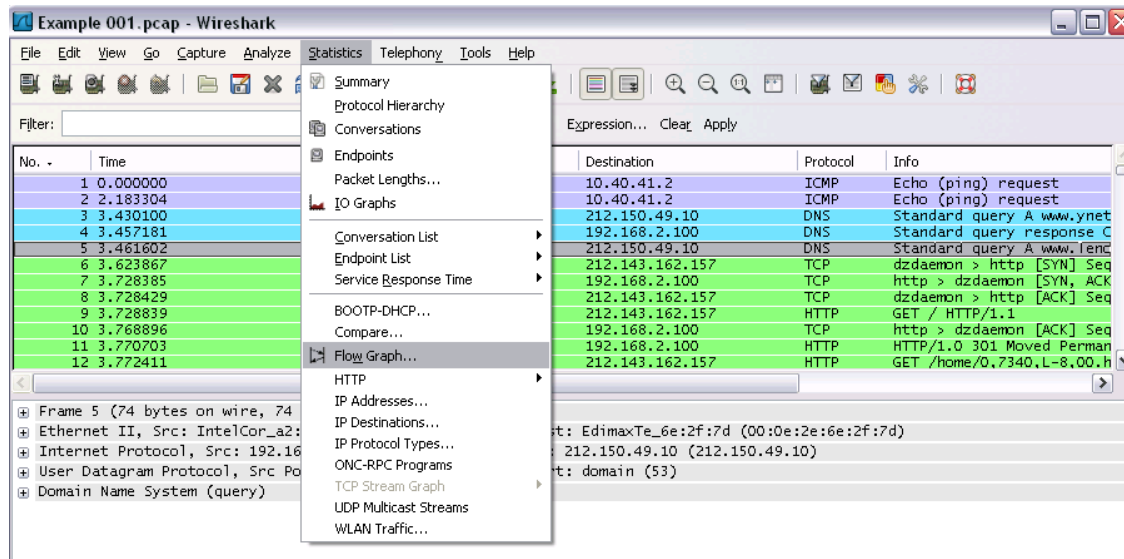
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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

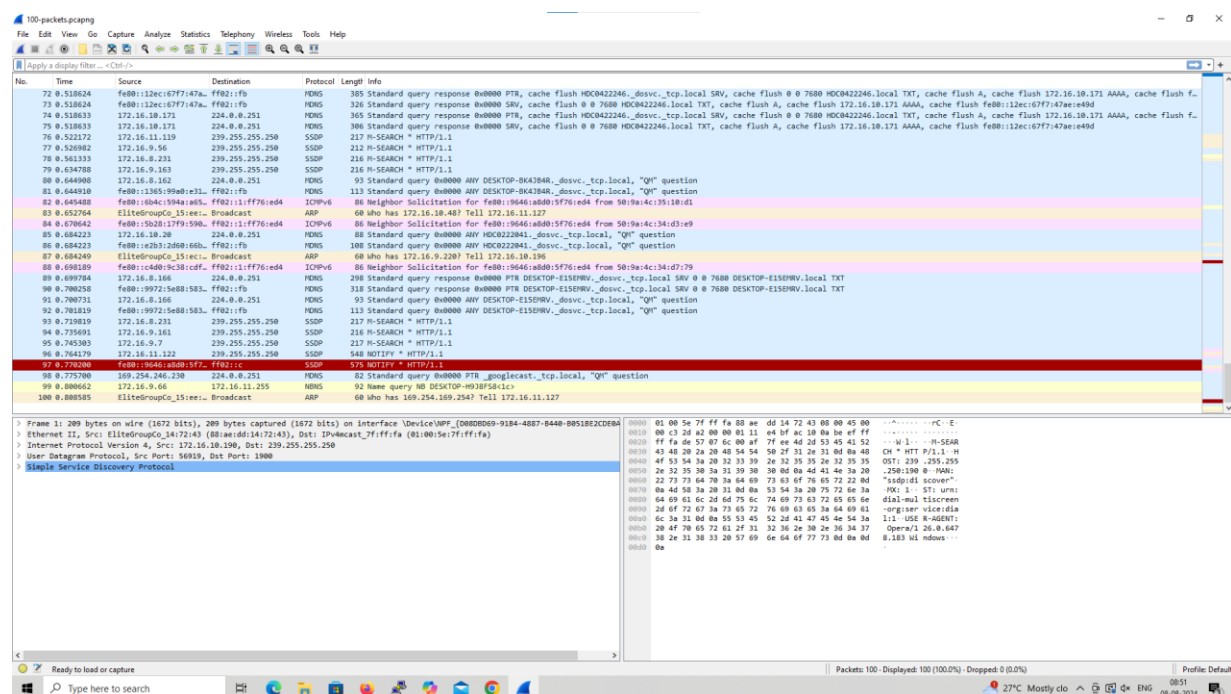
## Exercises

## 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



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 window is divided into three panes: Packet List, Packet Details, and Packet Bytes.

**Packet List:** Shows 100 captured packets. The first few packets are DNS queries and responses. The 100th packet is a DHCP Discover message.



**Packet Details:** The selected packet (No. 100) is a DHCP Discover message. The details pane shows the following structure:

- Frame 1: 289 bytes on wire (1672 bits), 289 bytes captured (1672 bits) on interface \\Device\\NPF\_{D08D0D9-9184-4887-B440-B0518E2CDE84}
- Ethernet II, Src: EliteGroupC\_1472:43 (88:e6:dd:14:72:43), Dst: IPmulticast\_ffff:fa (01:00:5e:7f:ff:fa)
- Internet Protocol Version 4, Src: 172.16.10.190, Dst: 239.255.255.250
- User Datagram Protocol, Src Port: 56919, Dst Port: 1900
- Simple Service Discovery Protocol

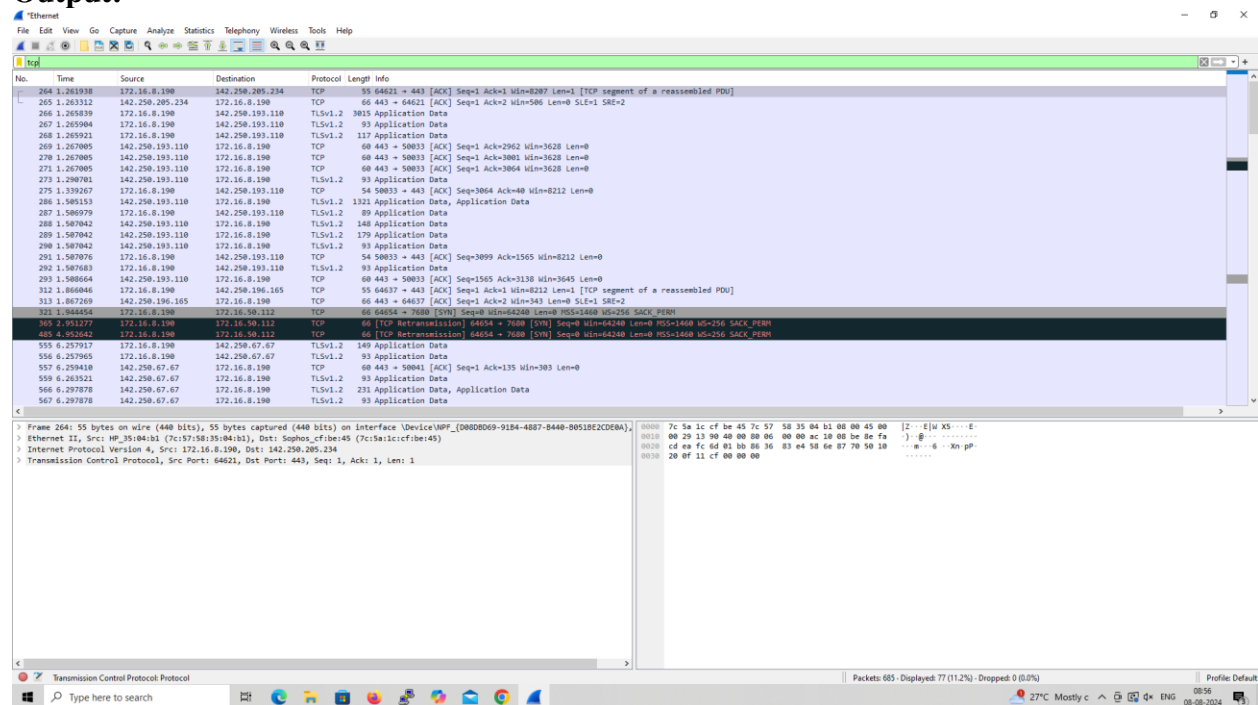
**Packet Bytes:** Shows the raw data of the selected packet in hexadecimal and ASCII. The data starts with 0000 01 00 5e 7f ff fa 88 ae dd 14 72 43 00 00 45 80, which corresponds to the Ethernet II header.

## 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 packet list shows several packets, including TCP and TLSv1.2. The packet details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw hex and ASCII data.

Packet 264: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on Interface UbcvclWPF\_{0808D09-91B4-48B7-8448-80518E2CDEA},  
 Ethernet II, Src: HP\_35:84:b1 (7c:57:56:35:84:b1), Dst: Sophos\_f8be:45 (7c:5a:1c:cf:f8be:45)  
 Internet Protocol Version 4, Src: 172.16.8.196, Dst: 142.250.205.234  
 Transmission Control Protocol, Src Port: 64621, Dst Port: 443, Seq: 1, Ack: 1, Len: 1

Wireshark - Packet 1196 - Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

\_ws.col.protocol == "UDP"

No.	Time	Source	Destination	Protocol	Length	Info
17	0.238992	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
21	0.259992	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
27	0.637532	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
56	1.013759	172.16.10.175	239.255.255.250	UDP	698	53853 → 3702 Len=656
129	1.986667	fe80::1557:e72d:91b... ffe02::c	UDP	718	53854 → 3702 Len=656	
130	1.946246	172.16.9.74	172.16.11.255	UDP	186	60000 → 51007 Len=144
202	3.017074	172.16.10.175	239.255.255.250	UDP	698	53853 → 3702 Len=656
221	3.248613	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
223	3.254303	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
269	3.639859	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
289	3.988057	fe80::1557:e72d:91b... ffe02::c	UDP	718	53854 → 3702 Len=656	
1195	9.955879	172.16.9.74	172.16.11.255	UDP	186	60000 → 51007 Len=144
1215	9.246778	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1218	9.276472	172.16.11.220	224.77.77.77	UDP	148	12177 → 12177 Len=106
1231	9.642748	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::1513:1f1c:90b... ffe02::c	UDP	718	56352 → 3702 Len=656	
1643	9.268289	172.16.10.211	239.255.255.250	UDP	698	56351 → 3702 Len=656
1645	9.281254	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 → 3702 Len=656
1655	9.581892	fe80::1513:1f1c:90b... ffe02::c	UDP	718	56352 → 3702 Len=656	
1654	9.641438	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 → 3702 Len=656
1685	9.963382	172.16.9.74	172.16.11.255	UDP	186	60000 → 51007 Len=144
1686	9.970373	fe80::1513:1f1c:90b... ffe02::c	UDP	718	56352 → 3702 Len=656	
1691	10.084094	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 → 3702 Len=656
1749	10.949999	fe80::1513:1f1c:90b... ffe02::c	UDP	718	56352 → 3702 Len=656	
1800	11.514262	172.16.10.211	239.255.255.250	UDP	698	56351 → 3702 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
1902	13.469812	172.16.9.205	224.77.77.77	UDP	148	12177 → 12177 Len=106
1911	12.762779	fe80::1513:1f1c:90b... ffe02::c	UDP	718	56352 → 3702 Len=656	

> Frame 1902: 148 bytes on wire (1184 bits), 146 bytes captured (1184 bits) on interface \Device\NPF\_{D080D69-9184-4887-8440-B051BE2CDE} 0000 01 00 5e 4d 4d 4d 00 bf b8 c3 65 65 08 00 45 00 ...FFFF...ne:E-  
0010 00 86 90 09 00 00 01 11 3a 06 ac 18 00 cd e0 4d ...-1...:.....H  
0020 4d 4d 2f 91 2f 91 00 72 33 20 3c 41 53 55 53 5f PW//...p 3(CASUS  
0030 41 52 4d 4f 55 52 59 5f 43 52 41 54 45 3e 3c 4c ARMOURY\_CRATE<L  
0040 41 4e 20 50 6f 72 74 50 22 31 32 31 37 37 22 20 All Port= "12177"  
0050 43 75 73 09 46 5d 22 31 39 46 44 41 45 33 38 2d CuiID="1 HP&C89-  
0060 30 37 31 39 20 34 46 35 46 2d 41 41 45 32 2d 37 0719-4F5 F-A&E2-7  
0070 35 41 42 45 34 41 38 38 45 31 22 20 2f 3e 3c SAB&A&B BE2"/+  
0080 2f 41 53 55 5f 41 52 4d 4f 55 52 59 5f 43 52 /ASUS\_ARM MOURY\_CR  
0090 41 54 45 3e ATE>

WireShark\_Ethernet0R3K2.pcapng Packets: 3154 - Displayed: 69 (2.2%) Profile: Default 09:41 08-08-2024

Wireshark - Packet 1196 - Ethernet

> Frame 1196: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface \Device\NPF\_{D080D69-9184-4887-8440-B051BE2CDE6A}, id 0  
> Ethernet II, Src: ASUSTekCompu\_c3:65:65 (08:bf:b0:c3:65:65), Dst: TP4decast\_4d:4d:4d (01:00:5e:4d:4d:4d)  
> 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 .....E-  
0010 00 ac 58 2c 00 00 80 74 ab ac 10 09 4a ac 10 ...X...t...J-  
0020 00 ff ea 68 c7 3f 00 98 f3 cc 4a b5 7f e5 21 43 ...h?...J...:C  
0030 bd f3 2d 9e da ff f9 b6 b3 37 0a 18 c3 30 c2 91 .....7...0-  
0040 86 24 65 99 2c 29 6c 00 4b 1b 19 83 16 c2 2b ...Se...J...+  
0050 22 c2 a1 a6 f6 32 42 e8 1a cf 22 da 07 cc 60 75 ..."...2B..."-u  
0060 53 7d 25 a5 3e aa 85 7f db 31 d7 a3 2e cd 60 9b 5}%>...-1...-  
0070 3d f1 12 0c 73 87 30 25 0c ca f9 0d 6c 84 92 fc ...s:0%.....1-  
0080 6b 19 e2 ba be aa 37 78 7f d2 02 c9 01 4e c3 00 k...7x...>N-  
0090 9e 01 96 f2 ff a1 05 9b 9c 67 17 de c2 28 ed 95 .....g...(-  
00a0 29 15 41 cb 99 3e 1a 4f 68 12 70 0f 43 d5 b0 fb ).A->O h p C...  
00b0 46 02 78 94 8e 9c d6 bc ba 66 .....f  
F:x.....f

No: 1196 Time: 3.658879 Source: 172.16.9.74 Destination: 172.16.11.255 Protocol: UDP Length: 186 Info: 60008 → 51007 Len=144

☒ Show packet bytes

Close Help

The top screenshot displays a network traffic analysis in Wireshark. The packet list shows a TCP Reset (RST) packet (Seq=6057, Win=0) sent from 172.16.8.190 to 172.16.9.112. The packet details pane shows the TCP segment with Seq=6057, Win=0, and a RST flag set. The packet bytes pane shows the raw data of the TCP segment. The packet list pane shows the packet details.

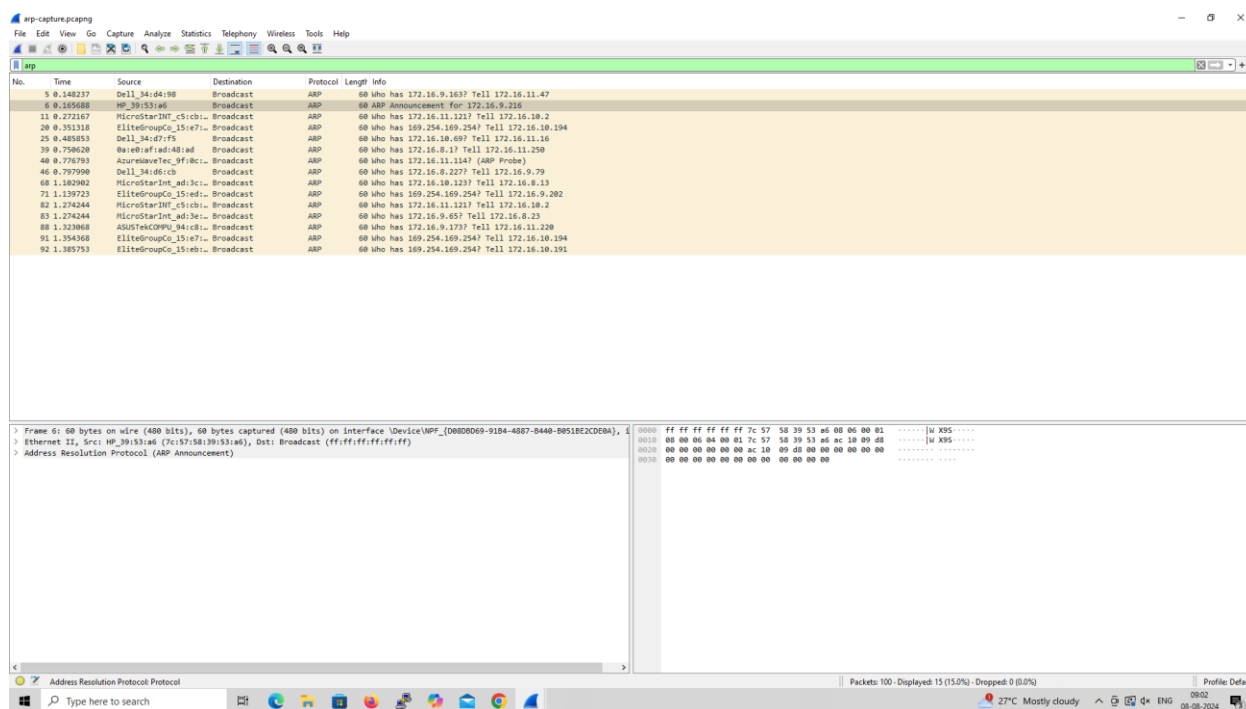
The bottom screenshot displays a network traffic analysis in Wireshark. The packet list shows a TCP Reset (RST) packet (Seq=12177, Win=0) sent from 172.16.11.220 to 172.16.9.205. The packet details pane shows the TCP segment with Seq=12177, Win=0, and a RST flag set. The packet bytes pane shows the raw data of the TCP segment. The packet list pane shows the packet details.

### 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





Wrench - Flow Ethernet

Time	172.16.8.190	172.16.8.1	Comment
15.175404	Standard query 0x1ea2 A edge.microsoft.com	DNS Standard query 0x1ea2 A edge.microsoft.com	
15.175597	Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	DNS Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	
15.177234	Standard query response 0x1ea2 A edge.microsoft.com CHAVE	DNS Standard query response 0x1ea2 A edge.microsoft.com	
15.177234	Standard query response 0x0b0a HTTP/1.5 edge.microsoft.com CH	DNS Standard query response 0x0b0a HTTP/1.5 edge.micr...	
15.1710966	Standard query 0x02d4 A edge.microsoft.com	DNS Standard query 0x02d4 A edge.microsoft.com	
15.1711175	Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	DNS Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	
15.1711948	Standard query response 0x02d4 A edge.microsoft.com CHAVE	DNS Standard query response 0x02d4 A edge.microsoft.com	
15.1711948	Standard query response 0x0b0a HTTP/1.5 edge.microsoft.com CH	DNS Standard query response 0x0b0a HTTP/1.5 edge.micr...	
15.729996	Standard query 0x02d4 A edge-mobile-static.akamai.net	DNS Standard query 0x02d4 A edge-mobile-static.akamai...	
15.730154	Standard query 0x0b0a HTTP/1.5 edge-mobile-static.akamai.net	DNS Standard query 0x0b0a HTTP/1.5 edge-mobile-static.akamai...	
15.731030	Standard query response 0x02d4 A edge-mobile-static.akamai.net	DNS Standard query response 0x02d4 A edge-mobile-static.akamai...	
15.731030	Standard query response 0x0b0a HTTP/1.5 edge-mobile-static.akamai.net	DNS Standard query response 0x0b0a HTTP/1.5 edge-mobile-static.akamai...	
15.780530	Standard query 0x02d4 A www.bing.com	DNS Standard query 0x02d4 A www.bing.com	
15.780699	Standard query 0x02d4 HTTP/1.5 www.bing.com	DNS Standard query 0x02d4 HTTP/1.5 www.bing.com	
15.781570	Standard query response 0x02d4 HTTP/1.5 www.bing.com CHAVE	DNS Standard query response 0x02d4 HTTP/1.5 www.bing.com	
15.781570	Standard query response 0x02d4 A www.bing.com CHAVE	DNS Standard query response 0x02d4 A www.bing.com	
16.046688	Standard query 0x02d4 A edge.microsoft.com	DNS Standard query 0x02d4 A edge.microsoft.com	
16.046740	Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	DNS Standard query 0x0b0a HTTP/1.5 edge.microsoft.com	
16.047677	Standard query response 0x02d4 HTTP/1.5 edge.microsoft.com CH	DNS Standard query response 0x02d4 HTTP/1.5 edge.micr...	
16.047677	Standard query response 0x02d4 A edge.microsoft.com CHAVE	DNS Standard query response 0x02d4 A edge.microsoft.com	
17.726760	Standard query 0x02d4 A msp.msn.com	DNS Standard query 0x02d4 A msp.msn.com	
17.726909	Standard query 0x02d4 HTTP/1.5 msp.msn.com	DNS Standard query 0x02d4 HTTP/1.5 msp.msn.com	
17.727980	Standard query response 0x02d4 HTTP/1.5 msp.msn.com CHAVE	DNS Standard query response 0x02d4 HTTP/1.5 msp.msn.com	
17.727980	Standard query response 0x02d4 A msp.msn.com CHAVE	DNS Standard query response 0x02d4 A msp.msn.com	
17.758437	Standard query 0x02d4 A msp.msn.com	DNS Standard query 0x02d4 A msp.msn.com	
17.759553	Standard query response 0x02d4 A msp.msn.com CHAVE	DNS Standard query response 0x02d4 A msp.msn.com	
17.975412	Standard query 0x02d4 A msp.msn.com alarmdata.net	DNS Standard query 0x02d4 A msp.msn.com alarmdata.net	
17.975632	Standard query 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	DNS Standard query 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	
17.975632	Standard query 0x02d4 A msp.msn.com alarmdata.net	DNS Standard query 0x02d4 A msp.msn.com alarmdata.net	
17.975632	Standard query 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	DNS Standard query 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	
17.976081	Standard query 0x02d4 A msp.msn.com	DNS Standard query 0x02d4 A msp.msn.com	
17.976140	Standard query 0x02d4 HTTP/1.5 msp.msn.com	DNS Standard query 0x02d4 HTTP/1.5 msp.msn.com	
17.976471	Standard query response 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	DNS Standard query response 0x02d4 HTTP/1.5 msp.msn.com alarmdata.net	
17.976471	Standard query response 0x02d4 A msp.msn.com alarmdata.net	DNS Standard query response 0x02d4 A msp.msn.com alarmdata.net	

Active 1652 DNS Standard query response 0x02d4 A www.bing.com CHAVE www.bing.com.edgekey.net CHAVE www.bing.com.edgekey.net CHAVE www.bing.com.edgekey.net A 23.223.244.123 A 23.223.244.131 A 23.223.244.133 A 23.223.244.147 A 23.223.244.158 A 23.223.244.170 A 23.223.244.171 A 23.223.244.181 A 23.223.244.182 A 23.223.244.184

☒ Limit to display filter

Flow type: All Flows


Address: Any

Reset Diagram Export Close Help

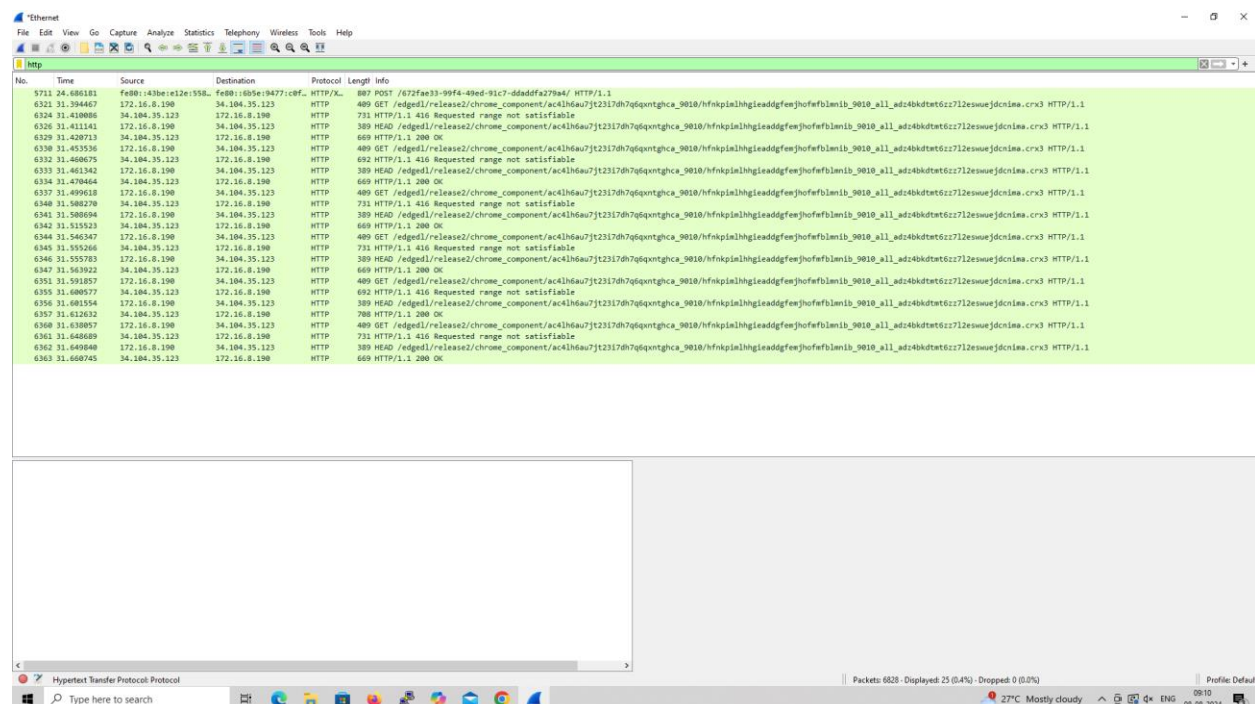


## 5. Create a Filter to display only HTTP packets and inspect the packets

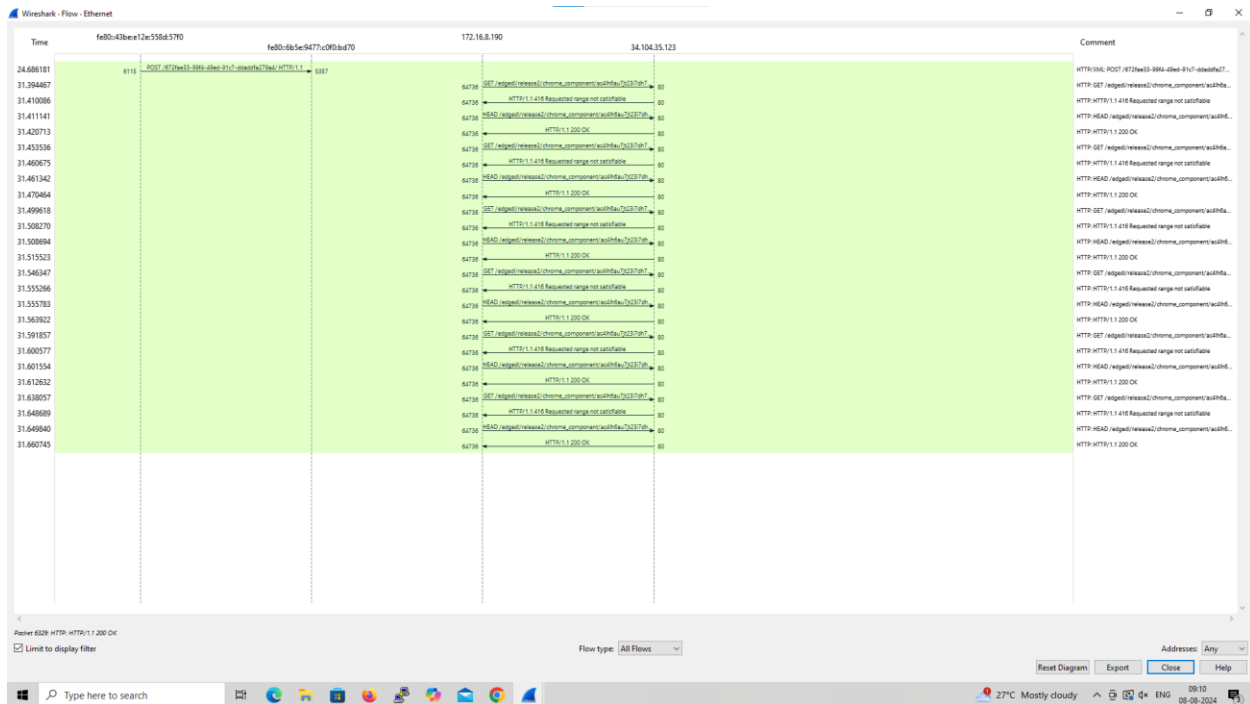
### Procedure

- 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

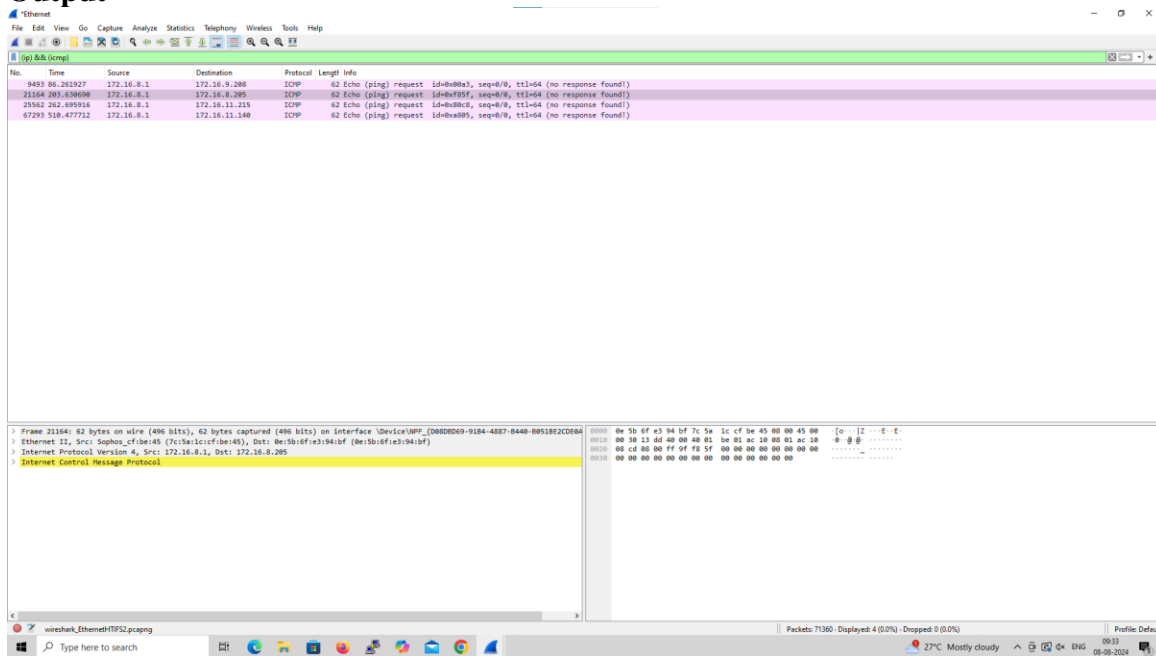


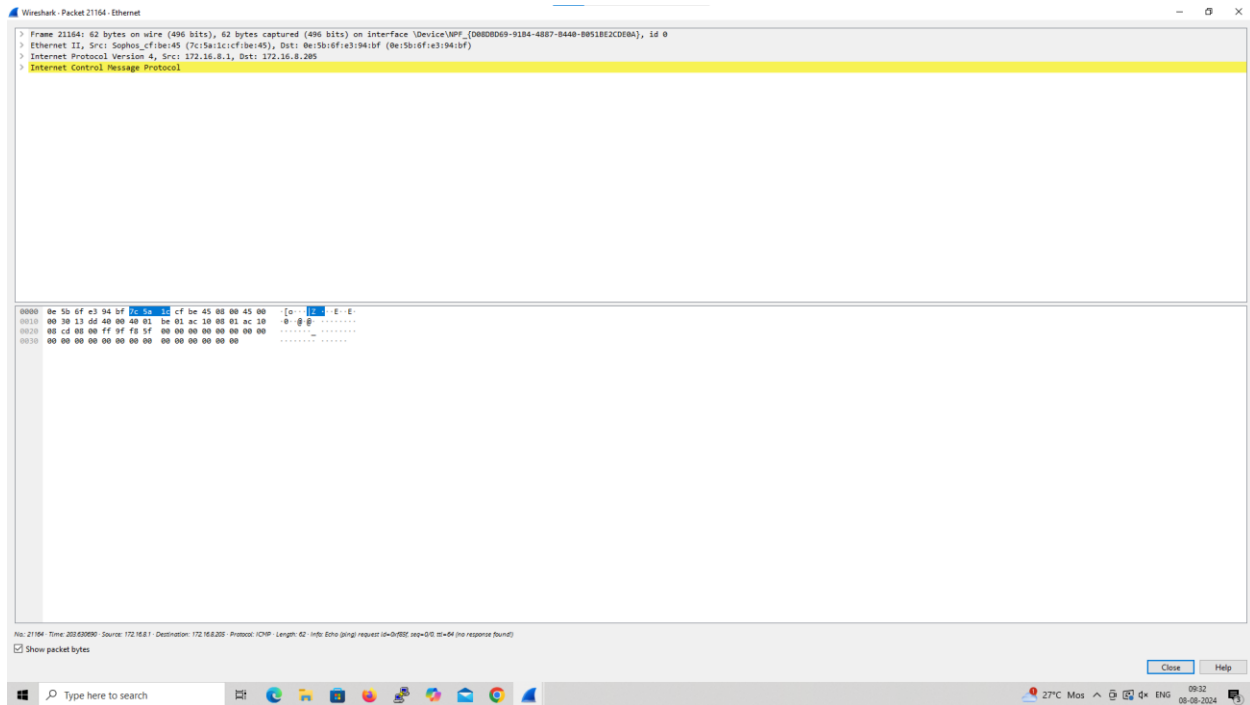
## 6.Create a Filter to display only IP/ICMP packets and inspect the packets.

### Procedure

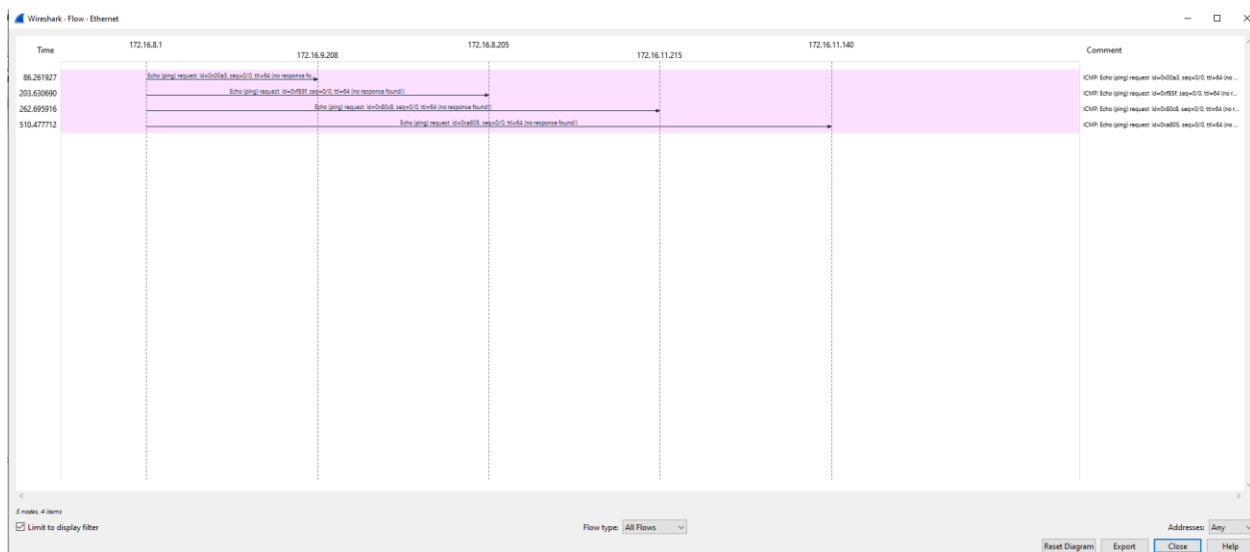
- Select Local Area Connection in Wireshark.
- Go to capture □ option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ICMP/IP packets in search bar.
- Save the packets

### 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

