

Ex No: 14a
DATE:13.8.24

NAME:KAVIYA J J
ROLL NO:231901019

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:

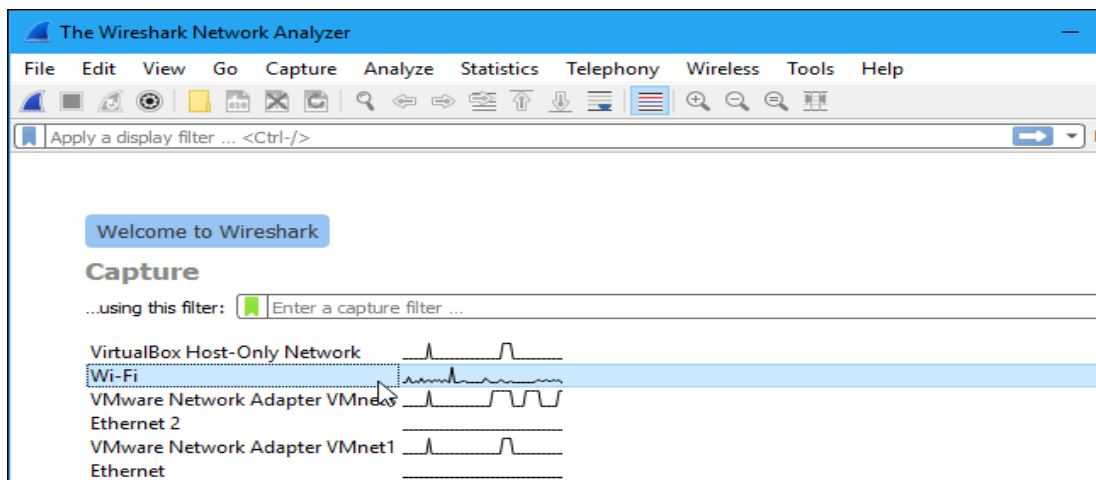
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

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.

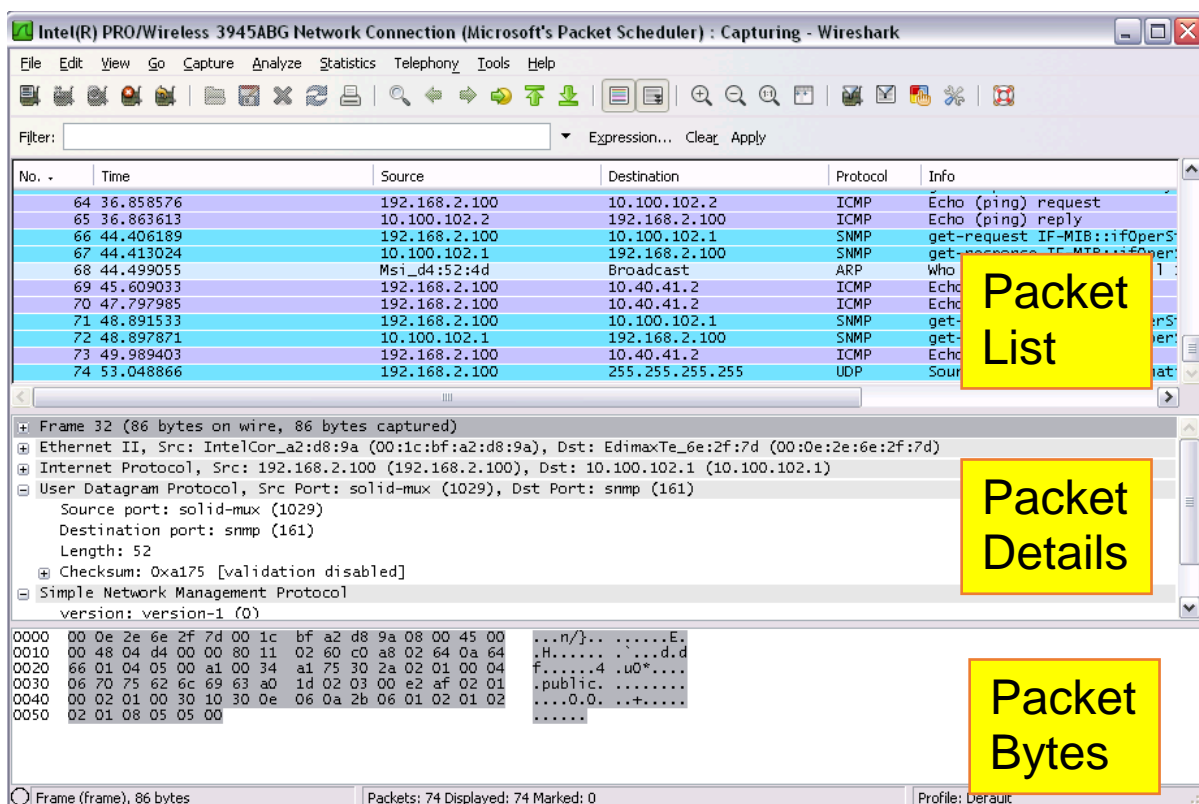
Capturing Packets

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.

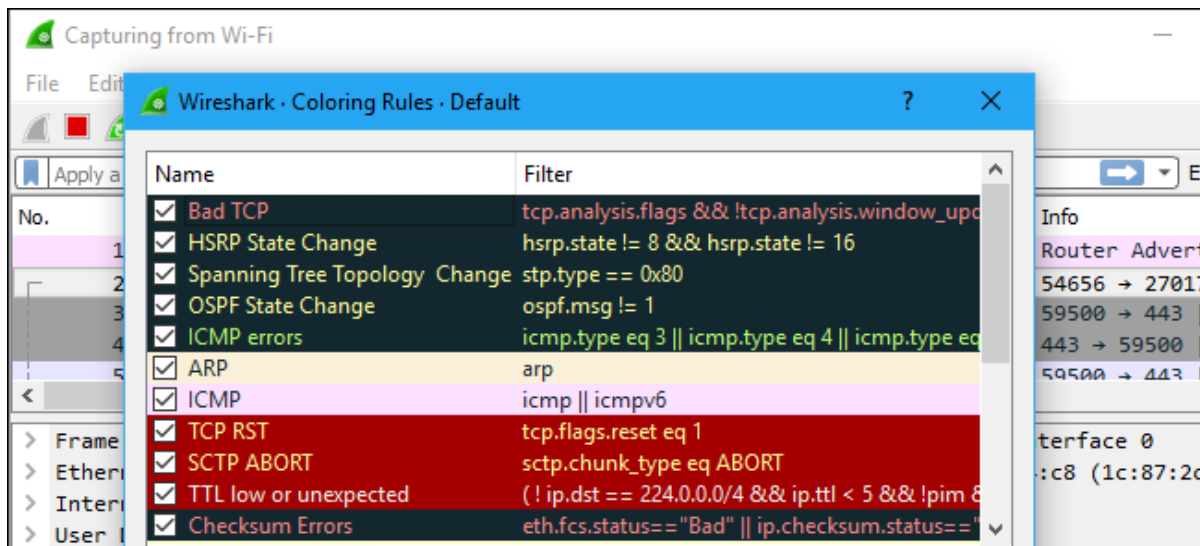
The “Packet Bytes” Pane

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.

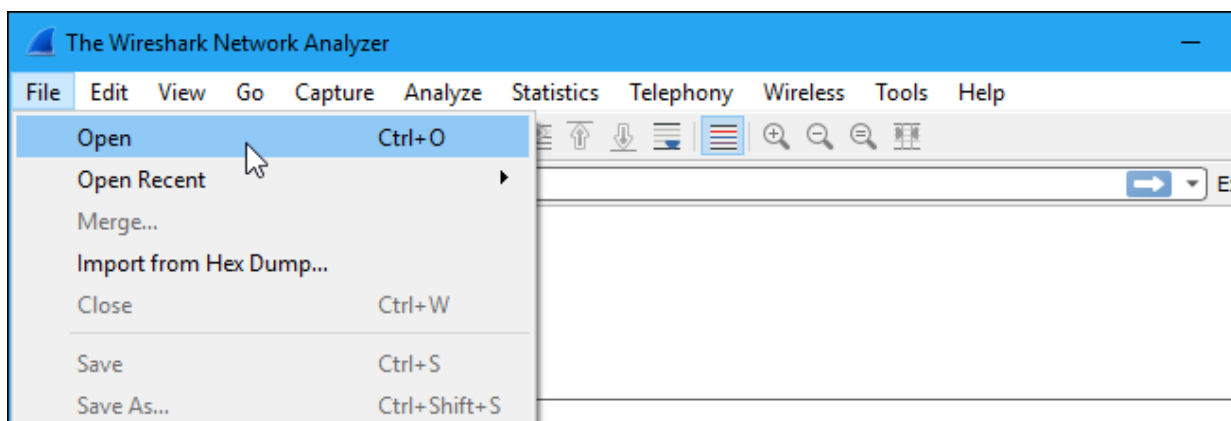
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



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.

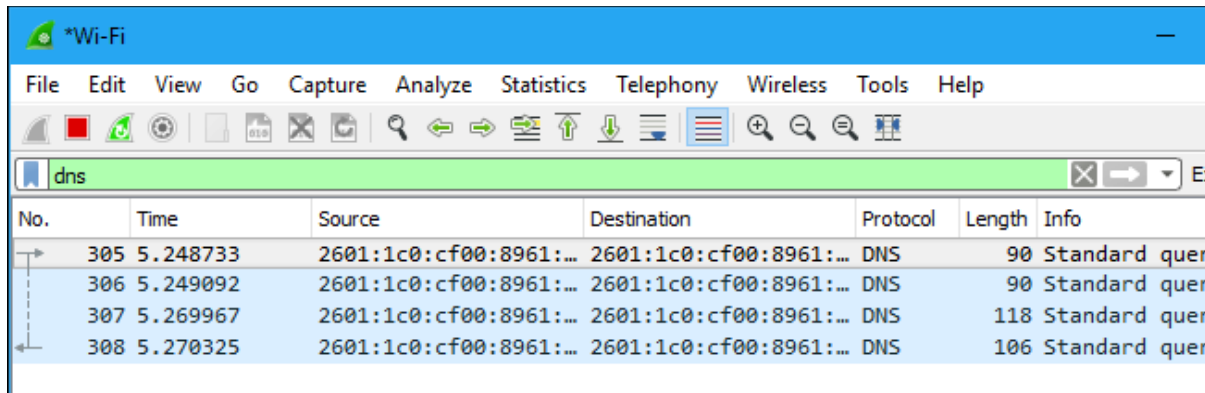


Filtering Packets

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

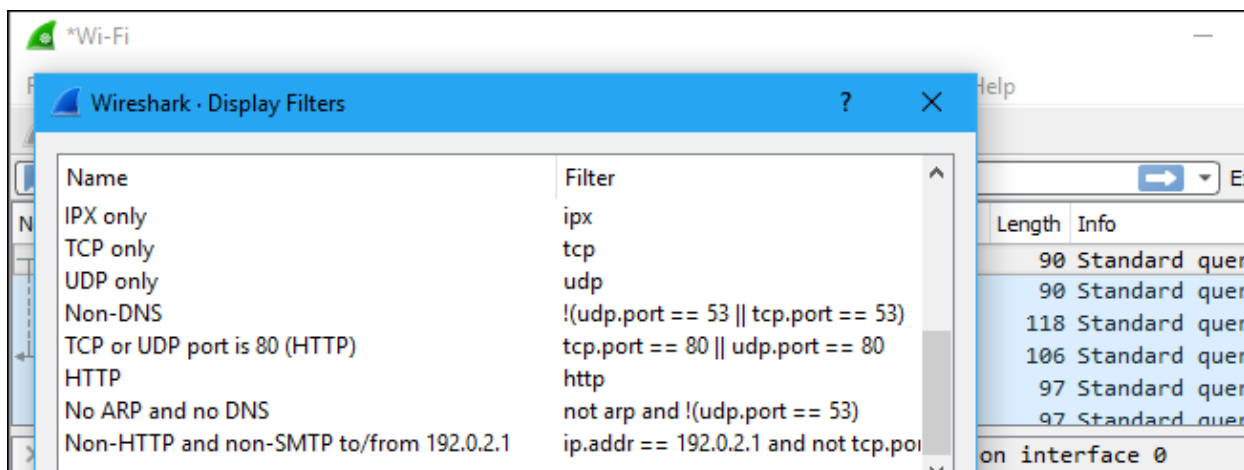
the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

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.



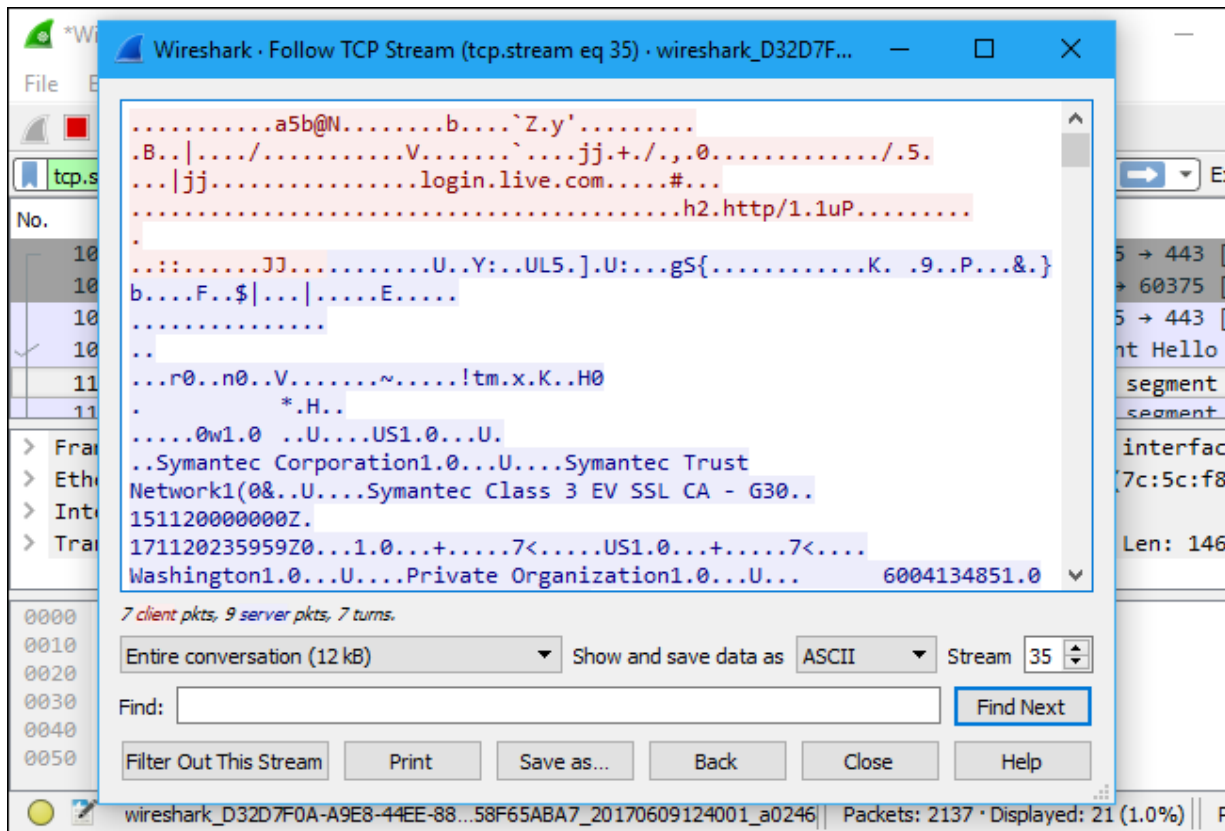
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

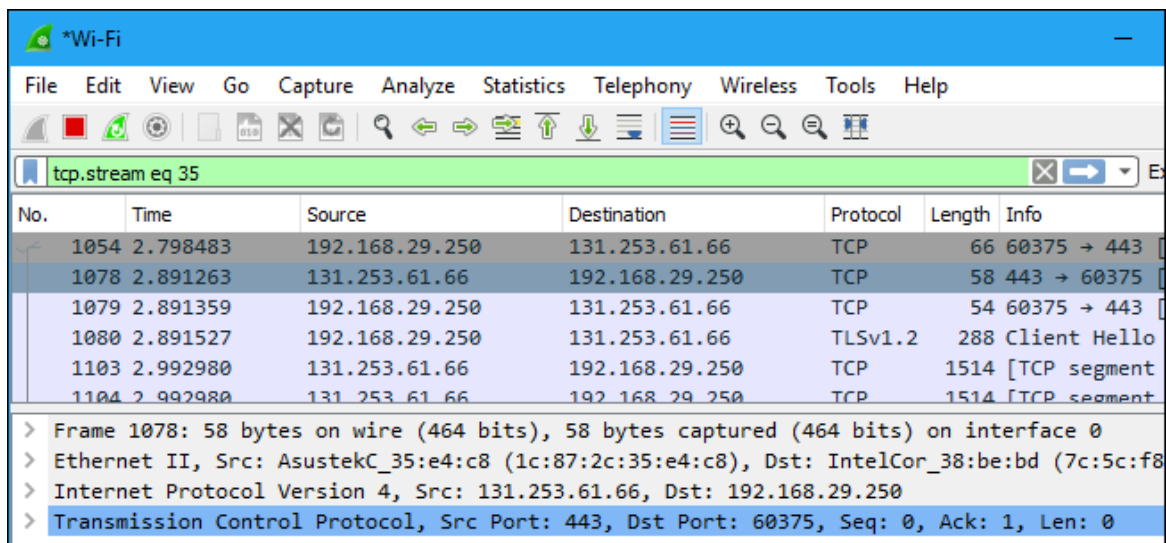


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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

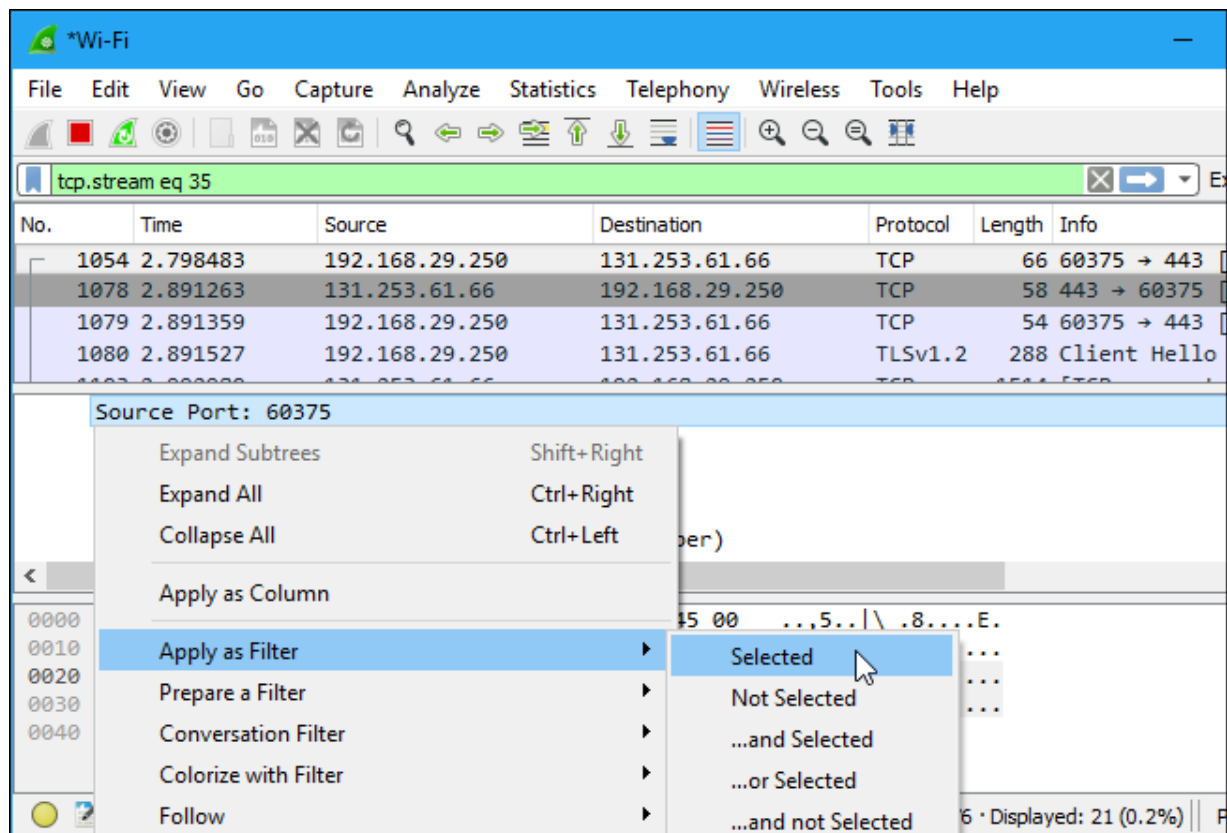
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

▼ 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	..

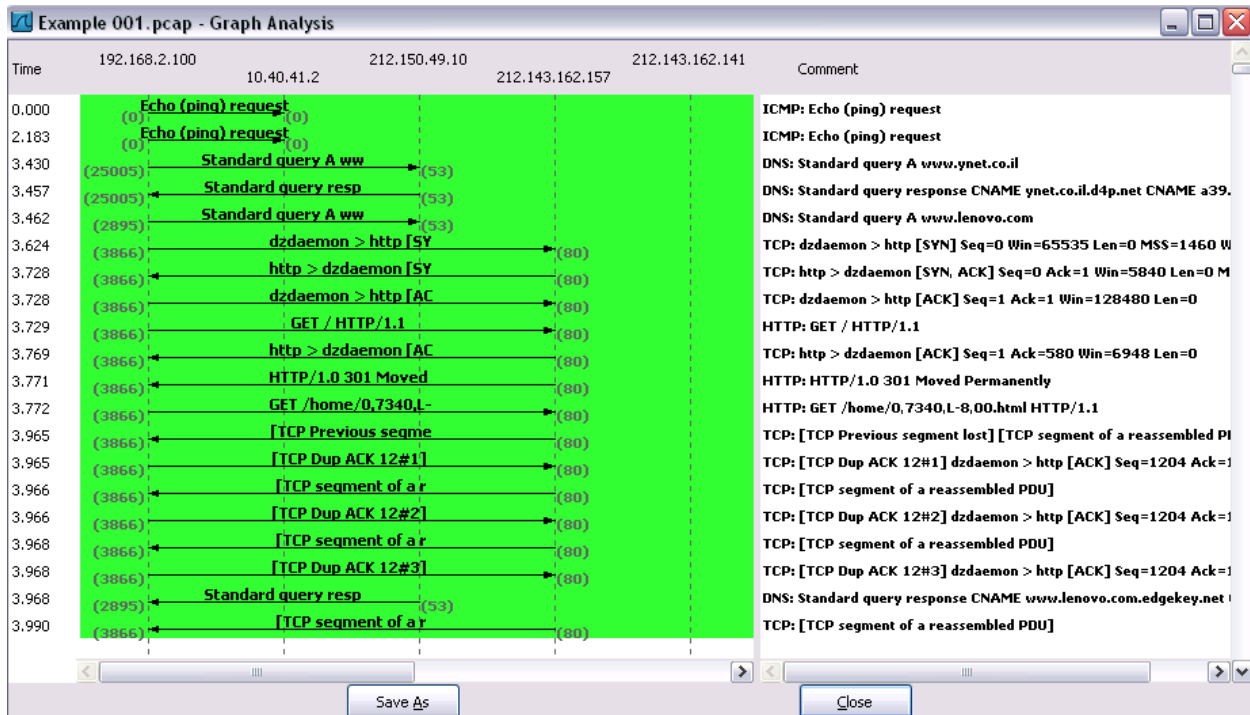
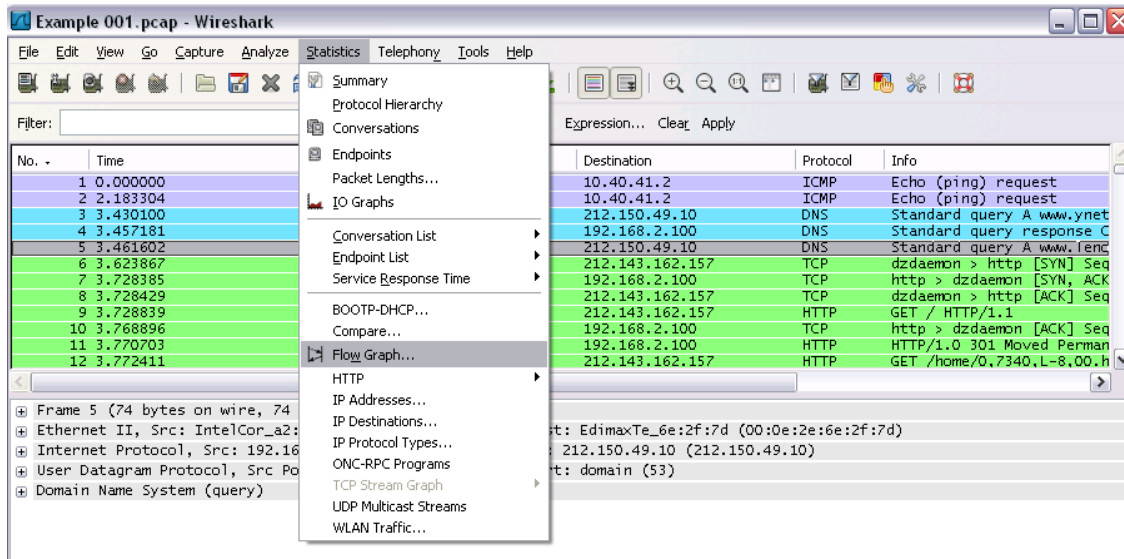
Encapsulation type (frame.encap_type) | Packets: 8136 · Displayed: 21 (0.3%)

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: 4 b
DATE:19.8.24

NAME:KAVIYA J J
ROLL NO:231901019

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

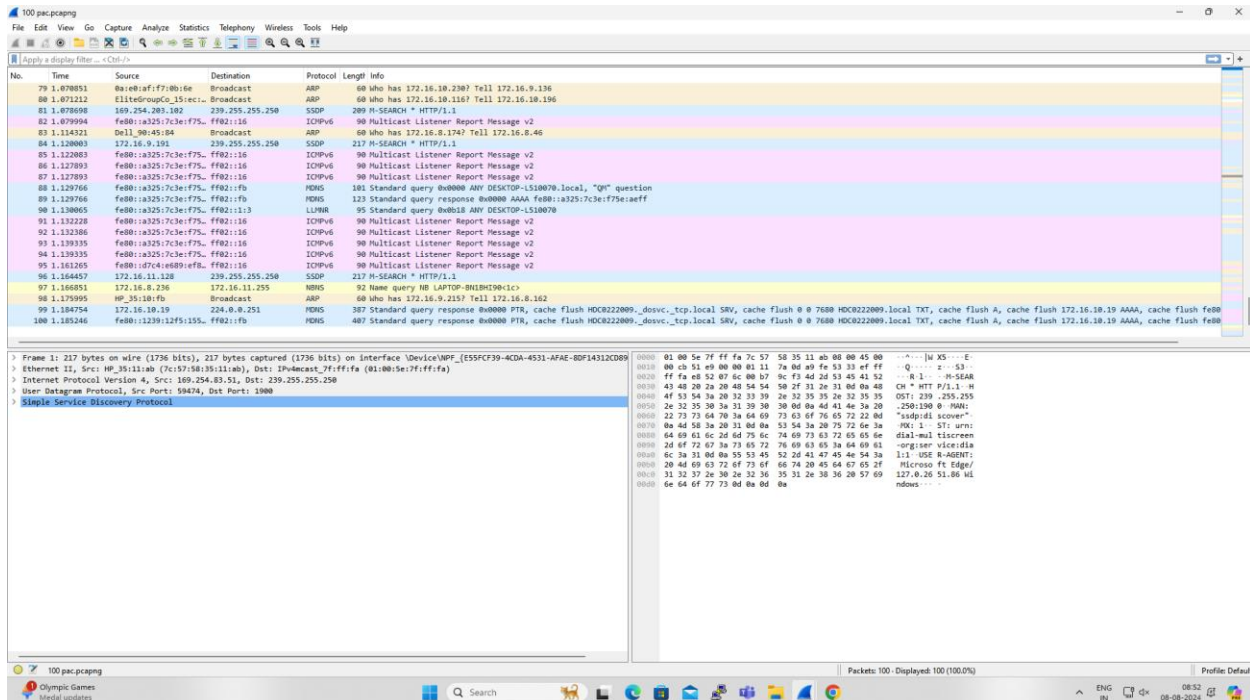
Exercises

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

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

Output:

IC	Time	Source	Destination	Protocol	Length	Info
75	0.004313	172.16.10.30	224.0.0.251	NMS	387	Standard query response 0x0000 PTR HCC0222051._dovsc._tcp.local SRV 0 7680 HCC0222051.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
76	0.004313	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	387	Standard query response 0x0000 PTR HCC0222051._dovsc._tcp.local SRV 0 7680 HCC0222051.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
77	0.004377	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	405	Standard query response 0x0000 PTR DESKTOP-05A939Q._dovsc._tcp.local SRV 0 7680 DESKTOP-05A939Q.local TXT A 172.16.10.62 AAAA fe8b:18b3:b4d5:320b:bdb0
78	0.004383	172.16.10.7	224.0.0.251	NMS	277	Standard query response 0x0000 PTR DESKTOP-05A939Q._dovsc._tcp.local SRV 0 7680 DESKTOP-05A939Q.local TXT A 172.16.10.7 AAAA fe8b:18b3:b4d5:320b:bdb0
79	0.004442	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	387	Standard query response 0x0000 PTR HCC0222008._dovsc._tcp.local SRV 0 7680 HCC0222008.local TXT A 172.16.10.17 AAAA fe8b:18b3:b4d5:320b:bdb0
80	0.004445	172.16.11.81	224.0.0.251	NMS	321	Standard query response 0x0000 PTR LAPTOP-VHLMF7G._dovsc._tcp.local SRV 0 7680 LAPTOP-VHLMF7G.local TXT A 172.16.11.81 AAAA fe8b:18b3:b4d5:320b:bdb0
81	0.004449	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	407	Standard query response 0x0000 PTR HCC0222096._dovsc._tcp.local SRV 0 7680 HCC0222096.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
82	0.004458	172.16.11.33	224.0.0.251	NMS	259	Standard query response 0x0000 PTR fail._dovsc._tcp.local SRV 0 7680 fail.local TXT A 172.16.11.33 AAAA fe8b:18b3:b4d5:320b:bdb0
83	0.004537	172.16.10.7	224.0.0.251	NMS	363	Standard query response 0x0000 PTR DESKTOP-05C1137._dovsc._tcp.local SRV 0 7680 DESKTOP-05C1137.local TXT A 172.16.10.7 AAAA fe8b:18b3:b4d5:320b:bdb0
84	0.004540	172.16.10.7	224.0.0.251	NMS	355	Standard query response 0x0000 PTR HCC0210239._dovsc._tcp.local SRV 0 7680 HCC0210239.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
85	0.004586	172.16.10.111	224.0.0.251	NMS	363	Standard query response 0x0000 PTR DESKTOP-556F18B._dovsc._tcp.local SRV 0 7680 DESKTOP-556F18B.local TXT A 172.16.10.111 AAAA fe8b:18b3:b4d5:320b:bdb0
86	0.004586	172.16.10.30	224.0.0.251	NMS	321	Standard query response 0x0000 PTR HCC0222028._dovsc._tcp.local SRV 0 7680 HCC0222028.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
87	0.004586	172.16.10.30	224.0.0.251	NMS	321	Standard query response 0x0000 PTR HCC0222028._dovsc._tcp.local SRV 0 7680 HCC0222028.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
88	0.004752	172.16.9.217	224.0.0.251	NMS	273	Standard query response 0x0000 PTR HCC0222025._dovsc._tcp.local SRV 0 7680 HCC0222025.local TXT A 172.16.9.217 AAAA fe8b:18b3:b4d5:320b:bdb0
89	0.004753	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	333	Standard query response 0x0000 PTR R1n1k131._dovsc._tcp.local SRV 0 7680 R1n1k131.local TXT A 172.16.10.201 AAAA fe8b:18b3:b4d5:320b:bdb0
90	0.004753	172.16.10.30	224.0.0.251	NMS	407	Standard query response 0x0000 PTR HCC0222096._dovsc._tcp.local SRV 0 7680 HCC0222096.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
91	0.004802	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	581	Standard query response 0x0000 PTR DESKTOP-P4114D0._dovsc._tcp.local SRV 0 7680 DESKTOP-P4114D0.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
92	0.004905	172.16.10.34	224.0.0.251	NMS	297	Standard query response 0x0000 PTR IEA399Q._dovsc._tcp.local SRV 0 7680 DESKTOP-IEA399Q.local TXT A 172.16.10.34 AAAA fe8b:18b3:b4d5:320b:bdb0
93	0.004991	172.16.1.177	224.0.0.251	NMS	429	Standard query response 0x0000 PTR DESKTOP-05A939Q._dovsc._tcp.local SRV 0 7680 DESKTOP-05A939Q.local TXT A 172.16.1.177 AAAA fe8b:18b3:b4d5:320b:bdb0
94	0.005025	172.16.10.30	224.0.0.251	NMS	299	Standard query response 0x0000 PTR DESKTOP-IEA399Q._dovsc._tcp.local SRV 0 7680 DESKTOP-IEA399Q.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0
95	0.005362	172.16.10.43	224.0.0.251	NMS	343	Standard query response 0x0000 PTR HCC0222018._dovsc._tcp.local SRV 0 7680 HCC0222018.local TXT A 172.16.10.43 AAAA fe8b:18b3:b4d5:320b:bdb0
100	0.005439	fe8b:18b3:b4d5:320b:bdb0	FQID::fbb0	NMS	297	Standard query response 0x0000 PTR HCC0222034._dovsc._tcp.local SRV 0 7680 HCC0222034.local TXT A 172.16.10.30 AAAA fe8b:18b3:b4d5:320b:bdb0

Inspecting packets

```

> Frame 279: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{E55FCF39-4CDA-4531-AFAE-8DF14312CD89}, id 0
> Ethernet II, Src: HP_35:05:d6 (7c:57:58:35:05:d6), Dst: HP_35:10:c5 (7c:57:58:35:10:c5)
> Internet Protocol Version 4, Src: 172.16.8.179, Dst: 172.16.8.182
> Transmission Control Protocol, Src Port: 7680, Dst Port: 50644, Seq: 94, Ack: 94, Len: 0

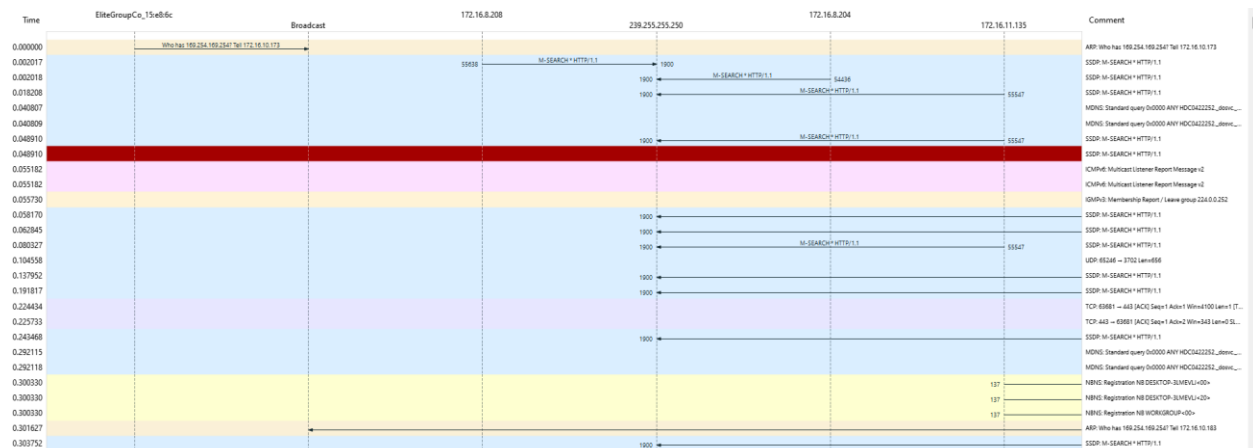
```

```

0000  7c 57 58 35 10 c5 7c 57 58 35 05 d6 00 00 45 00 |NXS...|W X5...E|
0010  00 28 1e 3e 40 00 00 06 00 00 ac 10 08 b3 ac 10 |(->@...|.....|
0020  00 b6 1e 00 c5 d4 b9 34 b6 6a fd 1e 1b 97 50 10 |.....4 .j....P.|
0030  10 04 69 a4 00 00 |...1....|


```

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

No.	Time	Source	Destination	Protocol	Length	Info
12	0.117472	MicroStarINT_c5icb0	Broadcast	ARP	68	Who has 172.16.11.98? Tell 172.16.10.43
14	0.141179	HP_35f05106	MicroStarINT_c5icac	ARP	42	Who has 169.254.178.62? Tell 172.16.8.179
15	0.141875	MicroStarINT_c5icac	HP_35f05106	ARP	68	169.254.178.62 is at d8bb1c1c3c13
17	0.157840	EliteGroupCo_151edc	Broadcast	ARP	68	Who has 172.16.9.200? Tell 172.16.10.203
27	0.371513	EliteGroupCo_151edc	Broadcast	ARP	68	Who has 169.254.142.18? Tell 172.16.10.196
32	0.434239	Waehafafad40ad	Broadcast	ARP	68	Who has 172.16.8.17? Tell 172.16.11.250
34	0.438790	RealtekSemi_421be1	Broadcast	ARP	68	Who has 172.16.11.103? Tell 172.16.11.126
35	0.442304	MicroStarINT_c5icac	Broadcast	ARP	68	Who has 172.16.9.200? Tell 172.16.10.110
48	0.594219	HikvisionDg_aa0eb	Broadcast	ARP	68	Who has 172.16.9.251? Tell 172.16.11.254
55	0.708130	Dell_xe091b	Broadcast	ARP	68	Who has 172.16.9.61? Tell 172.16.9.212
56	0.745332	MicroStarINT_c5icb1	Broadcast	ARP	68	Who has 172.16.10.48? Tell 172.16.10.44
61	0.770631	EliteGroupCo_151edc	Broadcast	ARP	68	Who has 169.254.169.254? Tell 172.16.10.173
62	0.881837	EliteGroupCo_151edc	Broadcast	ARP	68	Who has 172.16.8.233? Tell 172.16.10.187
71	0.802084	Dell_x7fa1c	Broadcast	ARP	68	Who has 172.16.11.48? Tell 172.16.9.200
85	1.134439	Waehafafad40ad	Broadcast	ARP	68	Who has 172.16.11.18? Tell 172.16.11.4
90	1.197695	EliteGroupCo_151edc	Broadcast	ARP	68	Who has 172.16.9.200? Tell 172.16.10.203
96	1.279587	MicroStarINT_c5icb1	Broadcast	ARP	68	Who has 172.16.10.48? Tell 172.16.10.44
106	1.401017	MicroStarINT_c5icb1	Broadcast	ARP	68	Who has 172.16.11.98? Tell 172.16.10.43
119	1.524072	Waehafafad40ad	Broadcast	ARP	68	Who has 172.16.8.17? Tell 172.16.11.250
120	1.551663	Dell_xe091b	Broadcast	ARP	68	Who has 172.16.9.61? Tell 172.16.9.212
122	1.594223	HikvisionDg_aa0eb	Broadcast	ARP	68	Who has 172.16.9.251? Tell 172.16.11.254
123	1.602135	MicroStarINT_c5icd1	Broadcast	ARP	68	Who has 169.254.192.79? Tell 172.16.10.23
130	1.690105	Dell_x07b1e	Broadcast	ARP	68	Who has 172.16.11.103? Tell 172.16.8.68

Frame 12: 68 bytes on wire (400 bits), 60 bytes captured (400 bits) on interface DeviceVPP (E55FCF39-4CDA-4531-AFAE-B0F1A312C0B9), Ethernet II, Src: MicroStarINT_c5icb0 (d8bb1c1c3c1b0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Address Resolution Protocol [request]

```

0000  ff ff ff ff ff ff 00 00 01 c5 c5 00 00 00 00 01
0010  00 00 00 04 00 01 00 00 c1 c5 c5 00 ac 18 0a 20
0020  00 00 00 00 00 00 ac 18 00 61 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Inspecting packets

```
> Frame 106: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{E55FCF39-4CDA-4531-AFAE-8DF14312CD89}, id 0
> Ethernet II, Src: MicroStarINT_c5:cb:d0 (d8:bb:c1:c5:cb:d0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff d8 bb c1 c5 cb d0 08 06 00 01 .....
0010  08 00 06 04 00 01 d8 bb c1 c5 cb d0 ac 10 0a 2b .....+
0020  00 00 00 00 00 00 ac 10 0b 62 00 00 00 00 00 .....b.....
0030  00 00 00 00 00 00 00 00 00 00 00 00 .....

```

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

- 2 Select stop capture automatically after 100 packets.
- 2 Then click Start capture.
- 2 Search DNS packets in search bar.
- 2 To see flow graph click Statistics 3 Flow graph.
- 2 Save the packets.

Output

Id	Time	Source	Destination	Protocol	Length	Info
9	0.009518	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd194 A t-ring-fallback2.smedge.net
162	0.008949	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd194 A t-ring-fallback2.smedge.net
307	0.040971	172.16.8.179	172.16.8.1	DNS	81	Standard query 0bd44a A static-ec2t-licdn.com
388	0.009551	172.16.8.179	172.16.8.179	DNS	136	Standard query response 0bd44a A static-ec2t-licdn.com CNAME sc1404.usp.epillicons.net A 152.190.43.62
517	0.016178	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd194 A t-ring-fallback2.smedge.net
744	10.118313	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd2ce A t-ring-fallback2.smedge.net
860	10.118475	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd2ce A t-ring-fallback2.smedge.net
861	11.155008	172.16.8.179	172.16.8.1	DNS	88	Standard query 0bd2ce A t-ring-fallback2.smedge.net

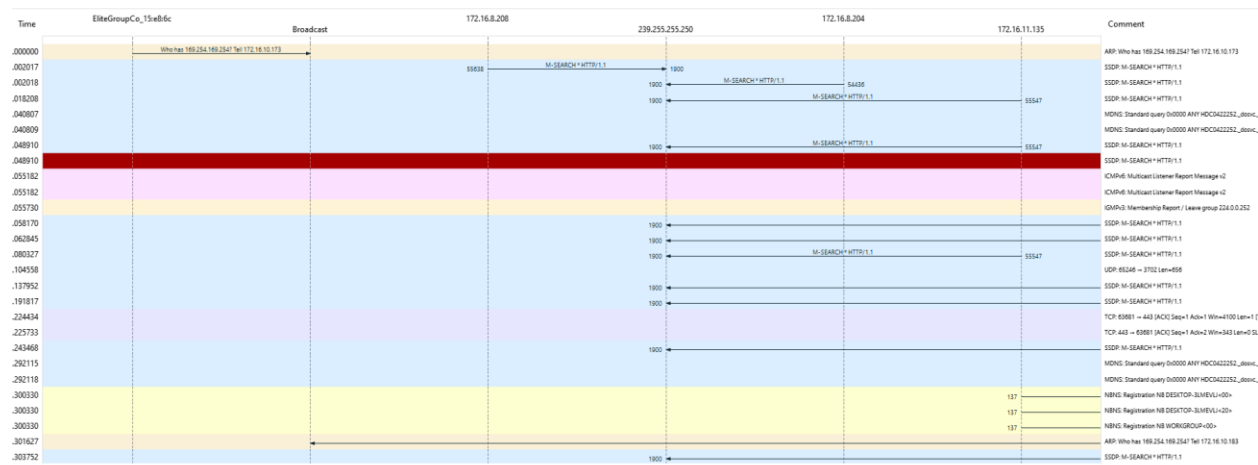
```

Frame 9: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface \Device\NPF{E55CF39-4CDA-4531-AFAE-BDF14312C0B9}, 1
Ethernet II, Src: HP_35:05:d6 (7c15758:35:05:d6), Dst: Sophos_cf1be145 (7c15a1c:cf1be145)
Internet Protocol Version 4, Src: 172.16.0.179, Dst: 172.16.8.1
User Datagram Protocol, Src Port: 54150, Dst Port: 53
Domain Name System (query)

0000  7c 5a 1c cf be 45 7c 57 58 35 05 d6 00 00 00 00 00 00  [2] ...E|W X5....
0010  00 4a f9 82 00 00 80 11 00 00 ac 10 00 b3 ac 10 00 00  [2] .....
0020  08 01 d3 8e 00 35 00 36 09 1c b1 94 01 00 00 01 00 00  [2] .....5 4 i
0030  00 00 00 00 00 00 11 74 25 72 69 6e 07 26 66 61 00 00  [2] .....t r!ng-fa
0040  6c 6c 62 61 63 60 73 32 06 6d 73 65 64 67 65 03 00 00  [2] libacks2
0050  6e 65 74 00 00 01 00 01 00 00 00 00 00 00 00 00 00  [2] net ....


```

Flow Graph output

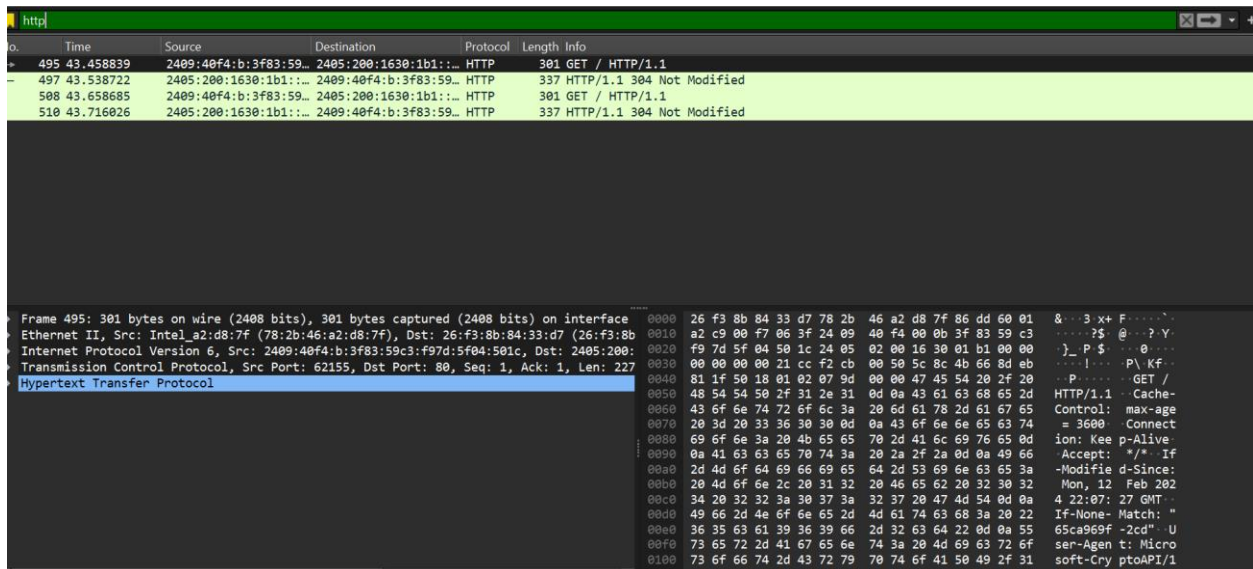


5. Create a Filter to display only HTTP packets and inspect the packets

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



Inspecting packets

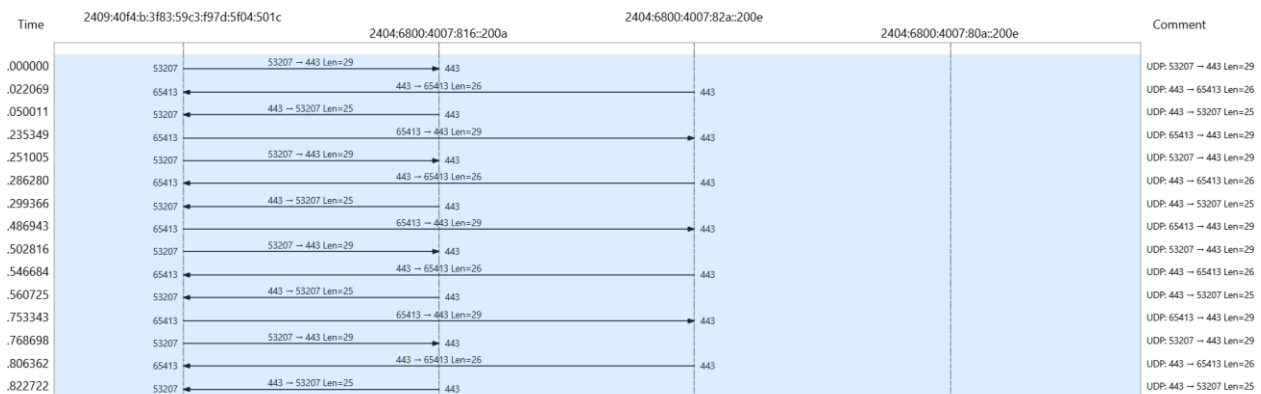
```

Frame 510: 337 bytes on wire (2696 bits), 337 bytes captured (2696 bits) on interface \Device\NPF_{C95C89FD-7053-473E-B986-5228ADF37491}
Ethernet II, Src: 26:f3:8b:84:33:d7 (26:f3:8b:84:33:d7), Dst: Intel_a2:d8:7f (78:2b:46:a2:d8:7f)
Internet Protocol Version 6, Src: 2405:200:1630:1b1::21cc, Dst: 2409:40f4:b:3f83:59c3:f97d:5f04:501c
Transmission Control Protocol, Src Port: 80, Dst Port: 62156, Seq: 1, Ack: 228, Len: 263
Hypertext Transfer Protocol

0000 78 2b 46 a2 d8 7f 26 f3 8b 84 33 d7 86 dd 60 06 x+F...&...3...
0010 ba 7a 01 1b 06 38 24 05 02 00 16 30 01 b1 00 00 z...8$...0...
0020 00 00 00 00 21 cc 24 09 40 f4 00 0b 3f 83 59 c3 ....!$.@...?Y-
0030 f9 7d 5f 04 50 1c 00 50 f2 cc 80 73 69 2a 21 73 }_P..P...si*!s
0040 0c cf 50 18 01 f9 0d 48 00 00 48 54 50 2f 31 ..P...H...HTTP/1
0050 2e 31 20 33 30 34 20 4e 6f 74 20 4d 6f 64 69 66 .1 304 N ot Modif
0060 69 65 64 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 ied Content-Typ
0070 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 70 e: appli cation/p
0080 6b 69 78 2d 63 72 6c 0d 0a 4c 61 73 74 2d 4d 6f kix-crl Last-Mo
0090 64 69 66 69 65 64 3a 20 4d 6f 6e 2c 20 31 32 20 dified: Mon, 12
00a0 46 65 62 20 32 30 32 34 20 32 32 3a 30 37 3a 32 Feb 2024 22:07:2
00b0 37 20 47 4d 54 0d 0a 45 54 61 67 3a 20 22 36 35 7 GMT-E Tag: "65
00c0 63 61 39 36 39 66 2d 31 32 62 22 0d 0a 43 61 63 ca969f-1 2b"...Cac
00d0 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 2d he-Contr ol: max-

```

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

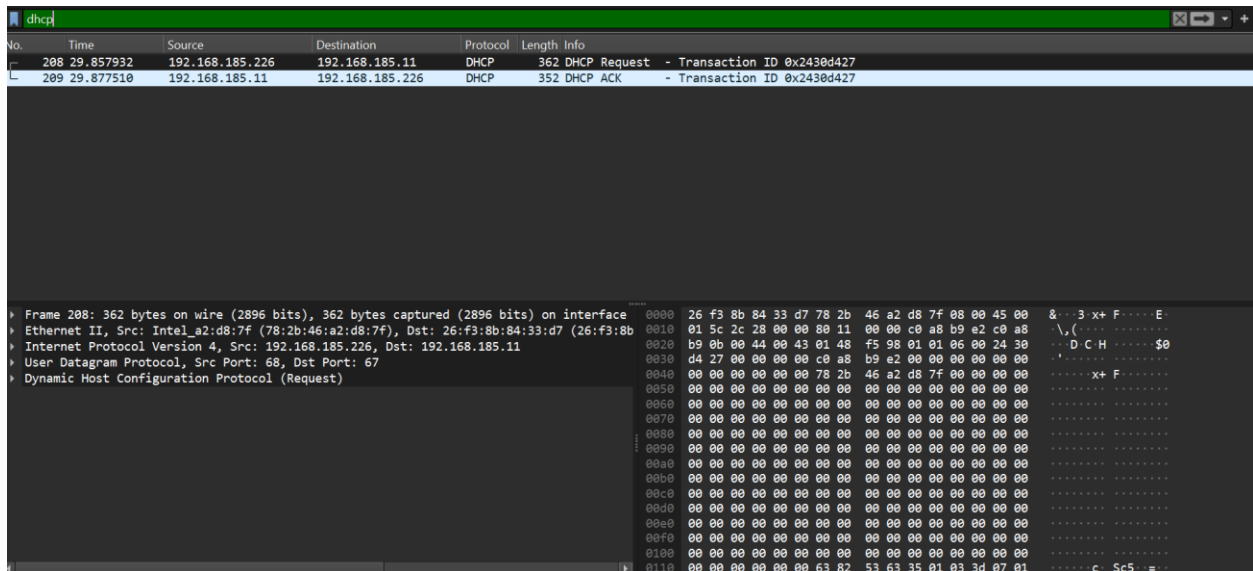
Output

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 packets

```

> Frame 209: 352 bytes on wire (2816 bits), 352 bytes captured (2816 bits) on interface \Device\NPF_{C95C89FD-7053-473E-B986-5228ADF37491}
> Ethernet II, Src: 26:f3:8b:84:33:d7 (26:f3:8b:84:33:d7), Dst: Intel_a2:d8:7f (78:2b:46:a2:d8:7f)
> Internet Protocol Version 4, Src: 192.168.185.11, Dst: 192.168.185.226
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Dynamic Host Configuration Protocol (ACK)

```

```

0000  78 2b 46 a2 d8 7f 26 f3 8b 84 33 d7 08 00 45 00  x+F...&...3...E
0010  01 52 e1 ee 40 00 40 11 63 6d c0 a8 b9 0b c0 a8  R...@.@ cm...
0020  b9 e2 00 43 00 44 01 3e 9e c9 02 01 06 00 24 30  ...C.D>...$0
0030  d4 27 00 00 00 00 c0 a8 b9 e2 c0 a8 b9 e2 c0 a8  ...
0040  b9 0b 00 00 00 00 78 2b 46 a2 d8 7f 00 00 00 00  ...x+ F...
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  ...

```