

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:

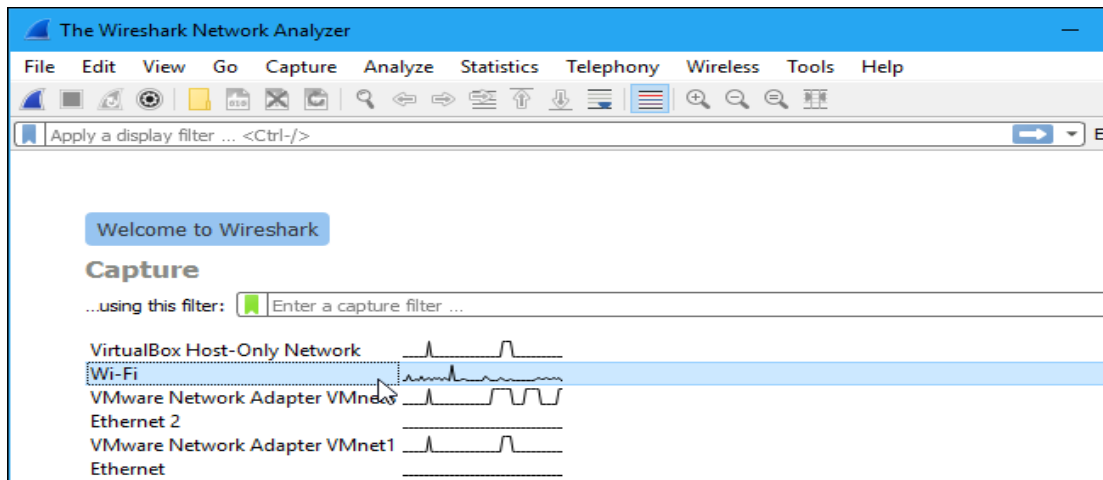
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

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.

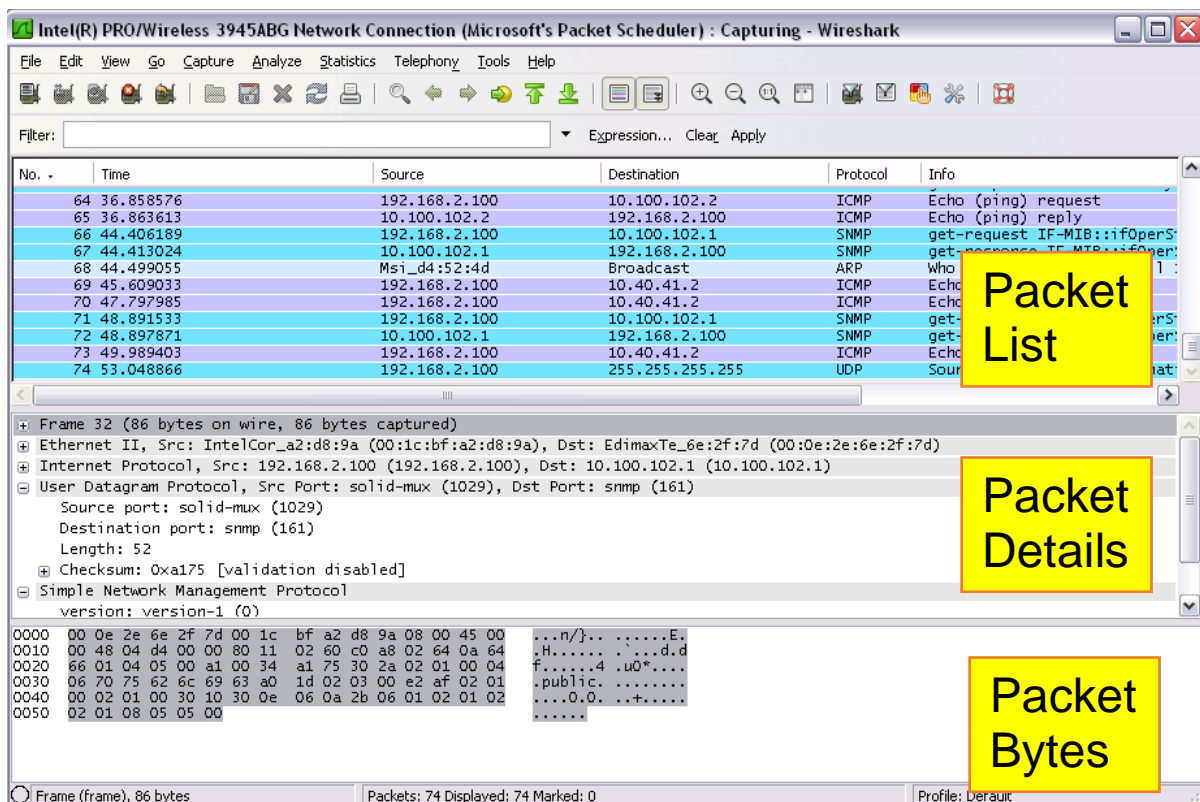
Capturing Packets

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.

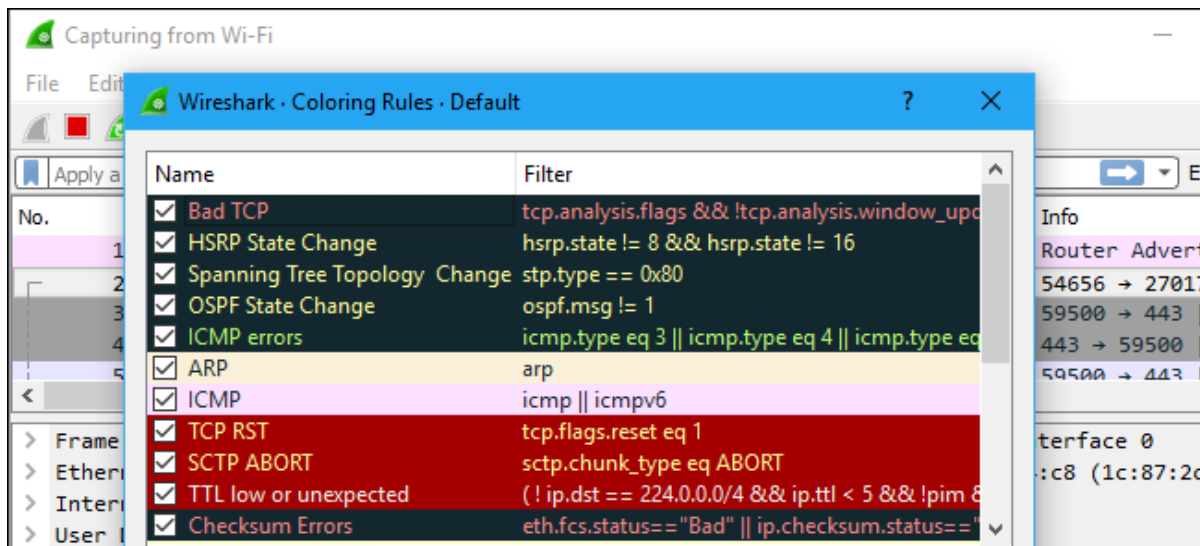
The “Packet Bytes” Pane

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.

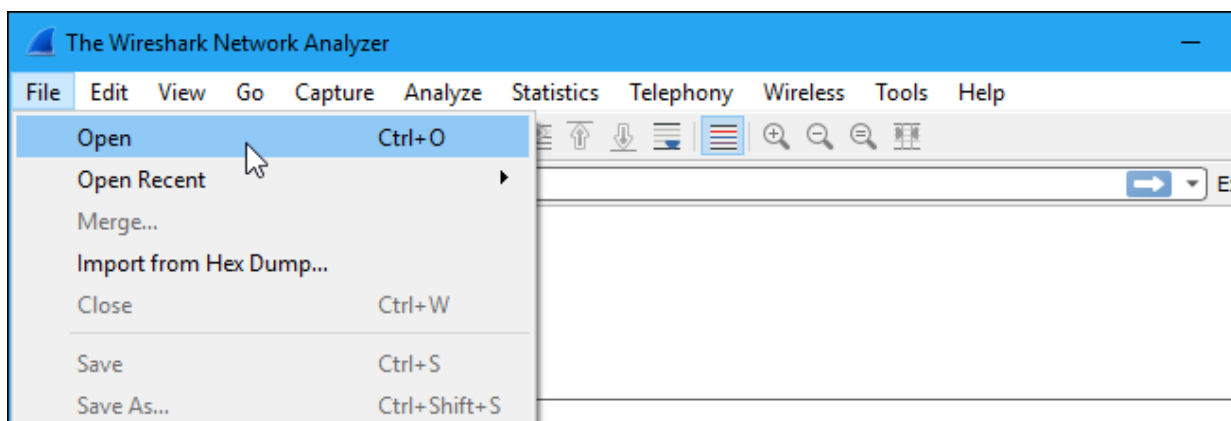
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



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.

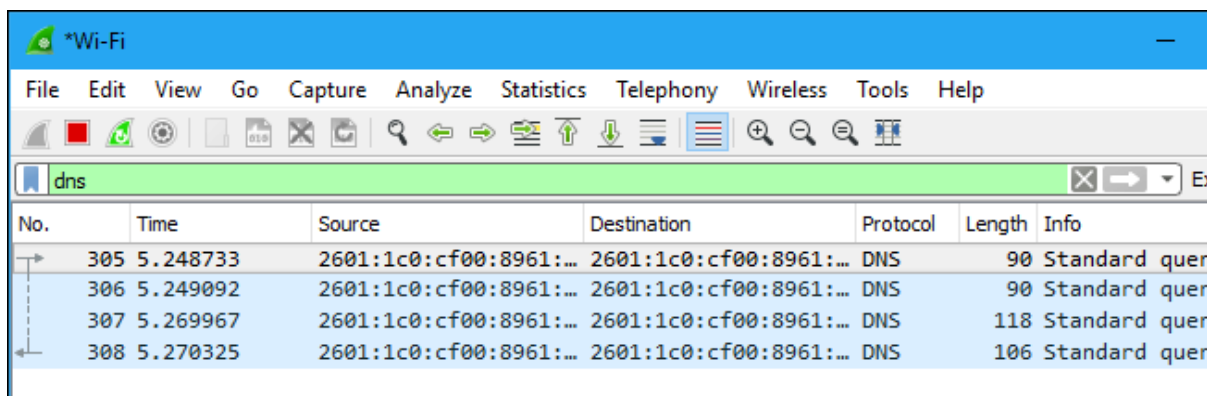


Filtering Packets

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

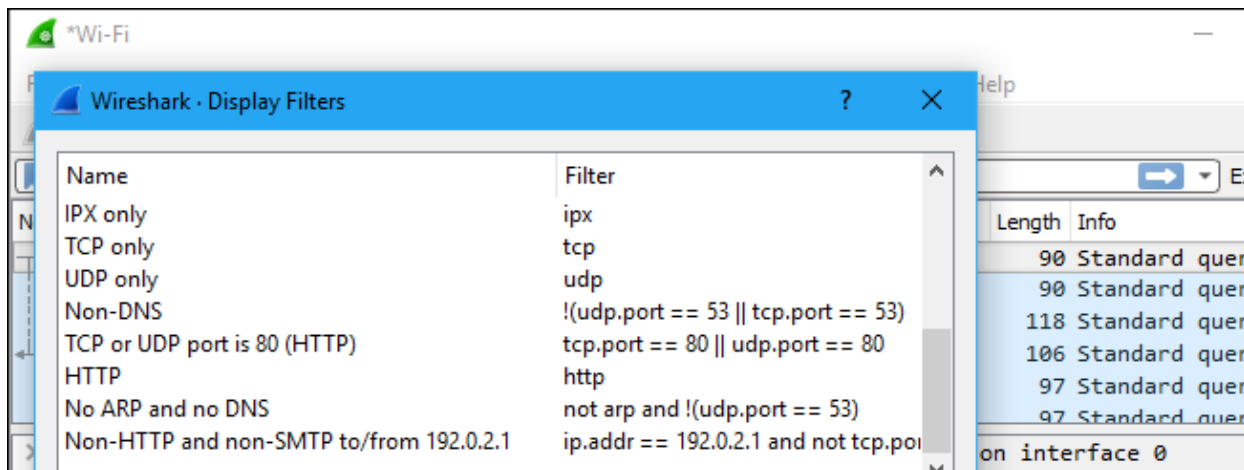
the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

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.



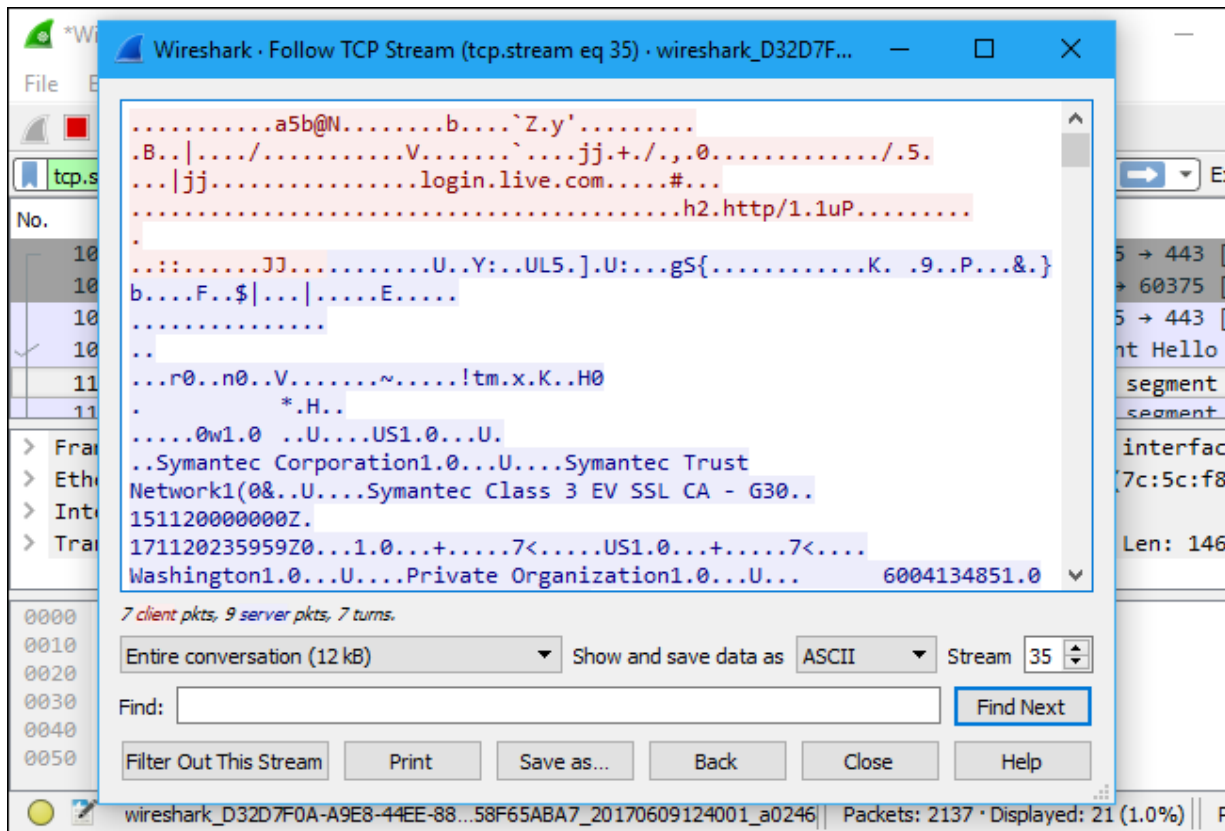
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

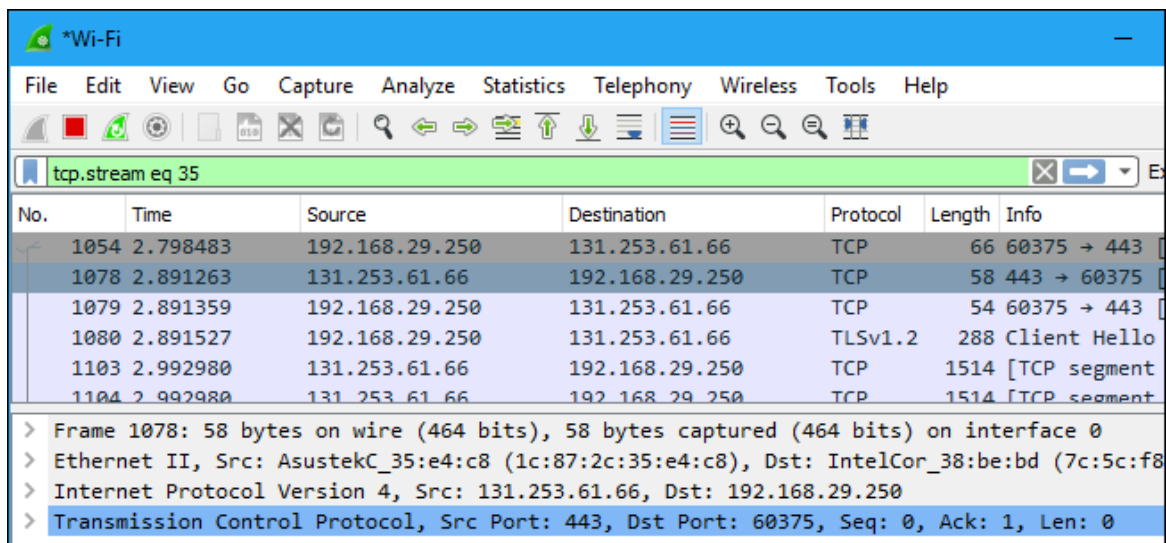


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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

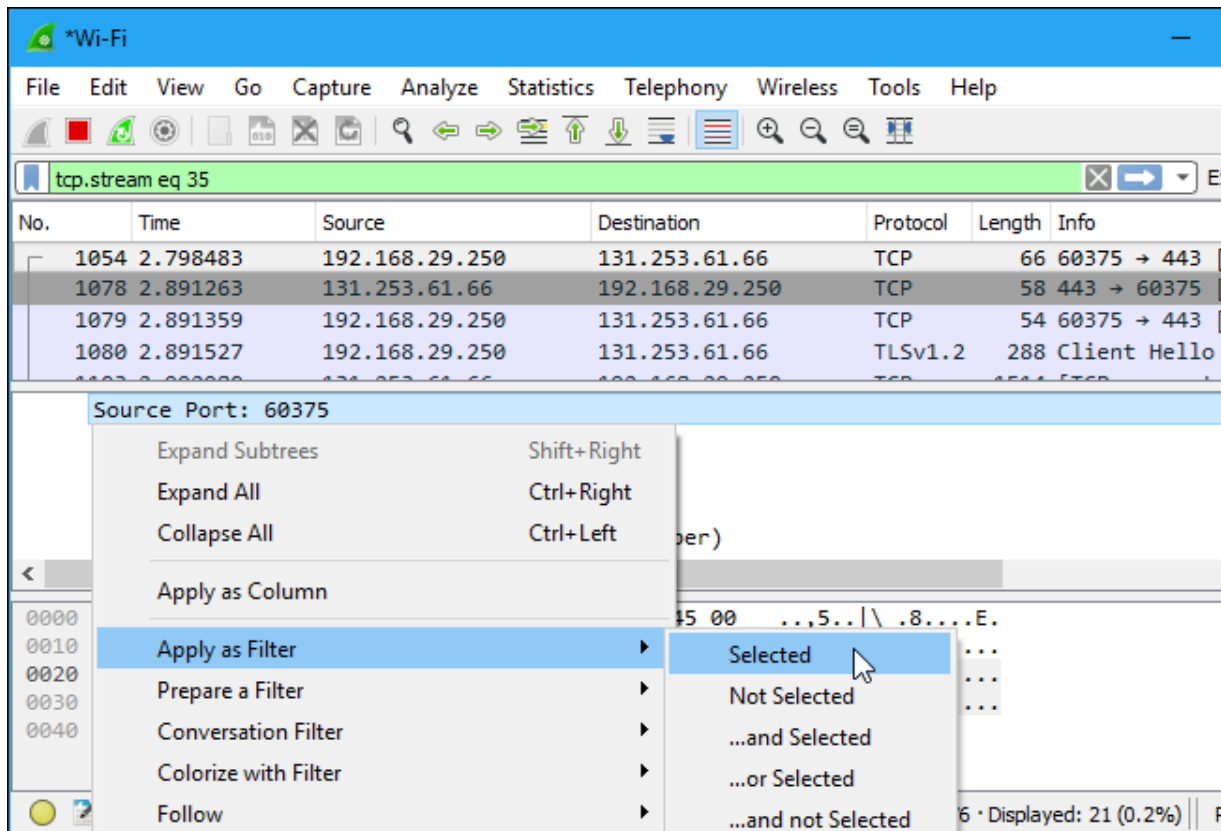
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

▼ 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	..

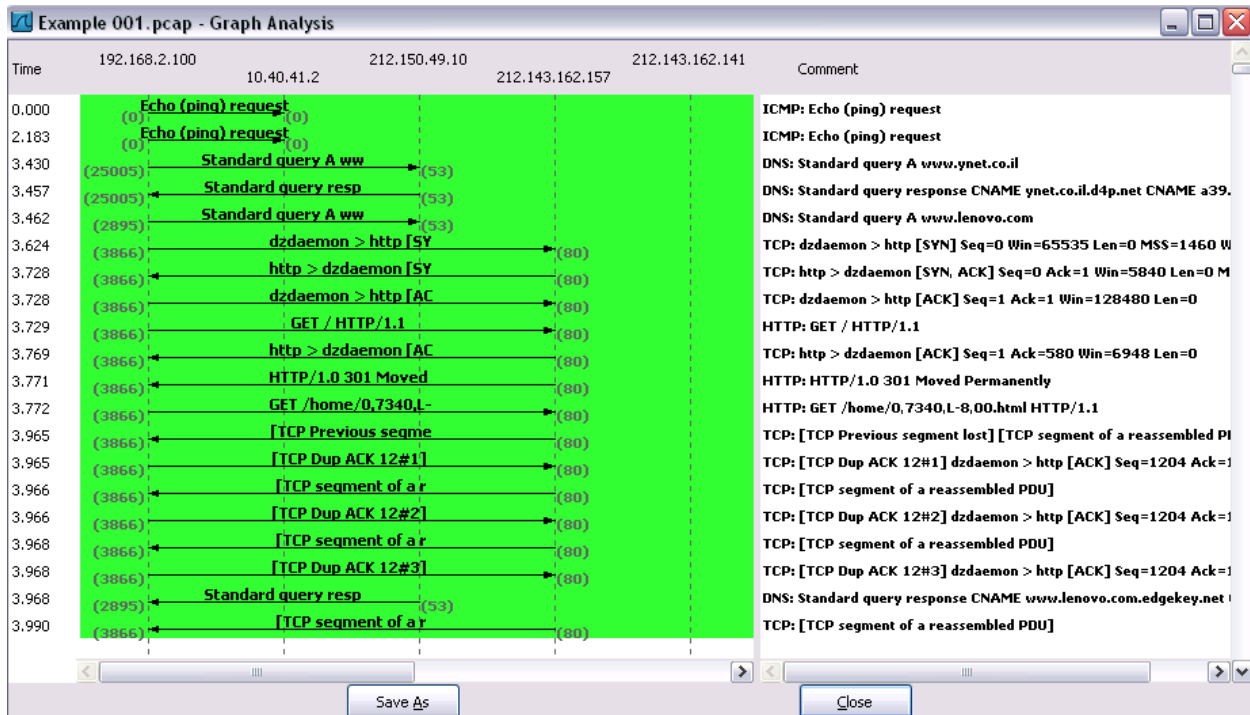
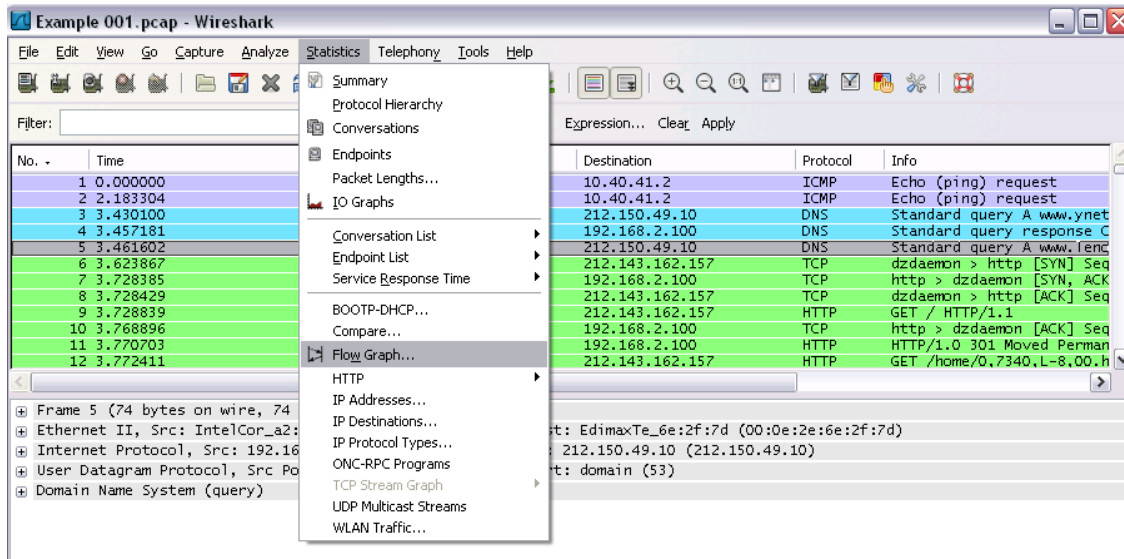
Encapsulation type (frame.encap_type) | Packets: 8136 · Displayed: 21 (0.3%)

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: 4 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

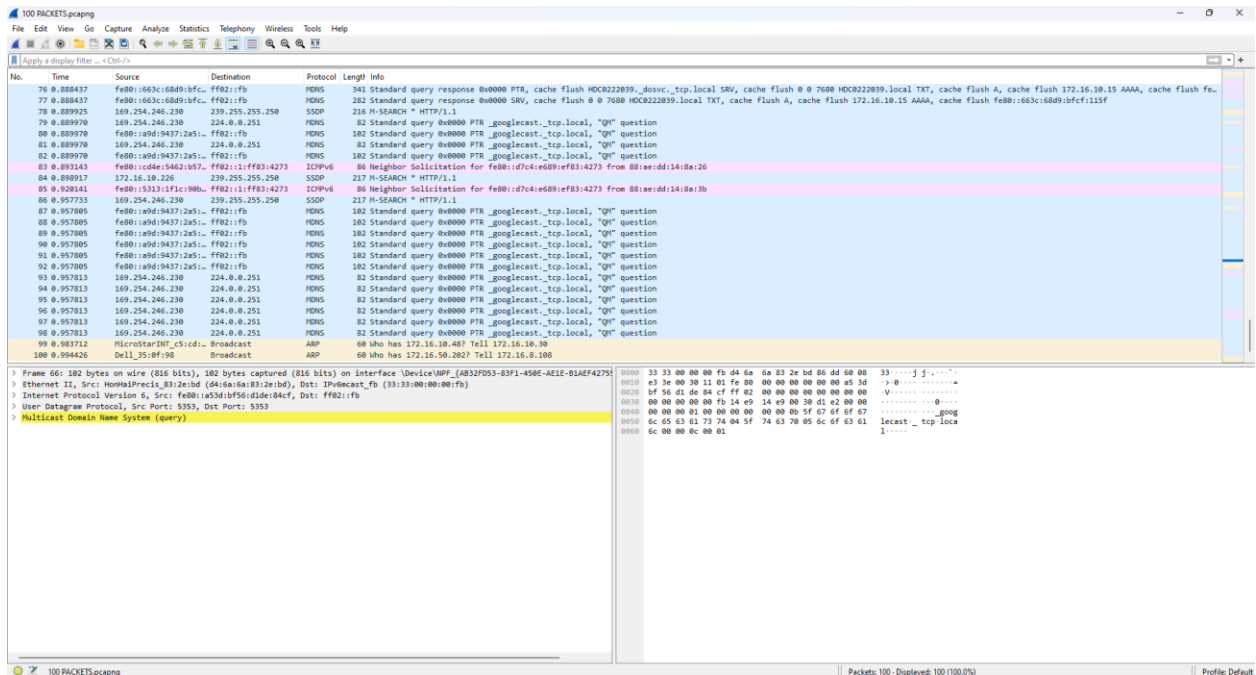
Exercises

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:

The screenshot shows the Wireshark interface with a packet capture list on the left and a packet details pane on the right. The packet list shows various protocols including DNS, HTTP, and TCP. The packet details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, and User Datagram Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.10.64	172.16.11.255	NDNS	92	Name query 000000 PTR microsoft_mcc_tcp.local, "QI" question
2	0.001449	172.16.10.237	239.255.255.250	SSDP	217	H-SEARCH * HTTP/1.1
4	0.004775	172.16.11.222	239.255.255.250	SSDP	216	H-SEARCH * HTTP/1.1
5	0.007740	172.16.11.220	224.77.77.77	UDP	140	12177 → 12177 Len=106
6	0.110299	172.16.10.64	239.255.255.250	SSDP	216	H-SEARCH * HTTP/1.1
7	0.118432	172.16.10.189	239.255.255.250	SSDP	216	H-SEARCH * HTTP/1.1
8	0.123932	172.16.10.189	239.255.255.250	SSDP	217	H-SEARCH * HTTP/1.1
9	0.142184	172.16.9.75	224.0.0.251	PDNS	85	Standard query 000000 PTR microsoft_mcc_tcp.local, "QI" question
10	0.142476	fe80::11f6:c576:eb7...	ff02::fb	PDNS	105	Standard query 000000 PTR microsoft_mcc_tcp.local, "QI" question
11	0.153326	172.16.10.182	224.0.0.251	PDNS	363	Standard query response 000000 PTR, cache flush DESKTOP-08LVE43.local TXT, cache flush A, cache flush 172.16.10.182 AAAA, cac...
12	0.153326	fe80::233e:43aa:17d...	ff02::fb	PDNS	363	Standard query response 000000 PTR, cache flush DESKTOP-08LVE43.local TXT, cache flush A, cache flush 172.16.10.182 AAAA, cac...
13	0.153326	172.16.10.182	224.0.0.251	PDNS	290	Standard query response 000000 SRV, cache flush 0 0 7680 DESKTOP-08LVE43.local TXT, cache flush A, cache flush 172.16.10.182 AAAA, cache flush fe80::233e:43aa:17d:c7a750
14	0.153326	fe80::233e:43aa:17d...	ff02::fb	PDNS	319	Standard query response 000000 SRV, cache flush 0 0 7680 DESKTOP-08LVE43.local TXT, cache flush A, cache flush 172.16.10.182 AAAA, cache flush fe80::233e:43aa:17d:c7a750
15	0.175583	172.16.10.25	224.0.0.251	PDNS	88	Standard query 000000 ANY HCR022043_dosvc_tcp.local, "QI" question
16	0.175583	fe80::ada5:b680:afa...	ff02::fb	PDNS	108	Standard query 000000 ANY HCR022043_dosvc_tcp.local, "QI" question
17	0.182225	172.16.9.144	239.255.255.250	SSDP	217	H-SEARCH * HTTP/1.1
18	0.196705	172.16.8.171	224.0.0.251	PDNS	82	Standard query 000000 PTR _googlecast_tcp.local, "QI" question
19	0.197940	fe80::7425:e621:c64...	ff02::fb	PDNS	102	Standard query 000000 PTR _googlecast_tcp.local, "QI" question
20	0.198580	172.16.9.75	224.0.0.251	PDNS	234	Standard query response 000000 PTR LAPTOP-DK9BASTC_dosvc_tcp.local SRV 0 0 7680 LAPTOP-DK9BASTC.local TXT
21	0.198675	fe80::11f6:c576:eb7...	ff02::fb	PDNS	234	Standard query response 000000 PTR LAPTOP-DK9BASTC_dosvc_tcp.local SRV 0 0 7680 LAPTOP-DK9BASTC.local TXT
22	0.199185	172.16.9.75	224.0.0.251	PDNS	93	Standard query 000000 ANY LAPTOP-DK9BASTC_dosvc_tcp.local, "QI" question
23	0.199188	fe80::11f6:c576:eb7...	ff02::fb	PDNS	113	Standard query 000000 ANY LAPTOP-DK9BASTC_dosvc_tcp.local, "QI" question
24	0.218600	172.16.9.214	239.255.255.250	SSDP	217	H-SEARCH * HTTP/1.1
27	0.262958	172.16.10.35	172.16.11.255	NDNS	92	Name query 000000 PTR 14:2d:4d:30:13:59@fe80::162d:4dff:fe30:1359-supportsRP-19_apple-mobdev2_tcp.local PTR 14:2d:4d:30:13:59@fe80::162d:4dff:fe30:1359-supportsRP-19_app...
28	0.284626	172.16.9.10	224.0.0.251	PDNS	200	Standard query response 000000 PTR 14:2d:4d:30:13:59@fe80::162d:4dff:fe30:1359-supportsRP-19_apple-mobdev2_tcp.local PTR 14:2d:4d:30:13:59@fe80::162d:4dff:fe30:1359-supportsRP-19_app...

Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on Interface 0:usbnet\{4032f053-83f1-408e-ae1e-81aef427556c}

Ethernet II, Src: MicrosoftINT_c5cc1b1 (d8:bb:1c:1c:5c:1b1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Internet Protocol Version 4, Src: 172.16.10.64, Dst: 172.16.11.255

User Datagram Protocol, Src Port: 137, Dst Port: 137

NetBIOS Name Service

Packets: 396 - Displayed: 309 (78.0%)

Profile: Default

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 image shows a Wireshark packet capture of network traffic. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for various functions like opening files, saving, and zooming. The packet list pane on the left shows 100 packets, with the first 100 packets being ARP requests. The packet details pane on the right shows the details of the first packet, which is an ARP request from 192.168.1.100 to 192.168.1.1.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000535	192.168.1.100	Broadcast	ARP	60	Who has 172.16.58.202? Tell 172.16.8.100
3	0.000535	192.168.1.100	Broadcast	ARP	60	Who has 172.16.58.202? Tell 172.16.8.100
15	0.150406	192.168.1.100	Broadcast	ARP	60	Who has 172.16.9.204? Tell 172.16.11.254
16	0.173999	192.168.1.100	Broadcast	ARP	60	Who has 172.16.8.104? Tell 172.16.8.208
19	0.203406	192.168.1.100	Broadcast	ARP	60	Who has 169.254.77.71? Tell 172.16.11.228
34	0.380712	192.168.1.100	Broadcast	ARP	60	Who has 172.16.8.202? Tell 172.16.11.217
43	0.413280	192.168.1.100	Broadcast	ARP	60	Who has 172.16.9.206? Tell 172.16.10.211
44	0.452738	192.168.1.100	Broadcast	ARP	60	Who has 172.16.10.48? Tell 172.16.10.30
48	0.512560	192.168.1.100	Broadcast	ARP	60	Who has 172.16.9.206? Tell 172.16.10.220
67	0.762755	192.168.1.100	Broadcast	ARP	60	Who has 172.16.10.208? Tell 172.16.8.57
68	0.795871	192.168.1.100	Broadcast	ARP	60	Who has 169.254.74.20? Tell 172.16.10.196
99	0.983712	192.168.1.100	Broadcast	ARP	60	Who has 172.16.10.48? Tell 172.16.10.30
100	0.984426	192.168.1.100	Broadcast	ARP	60	Who has 172.16.58.202? Tell 172.16.8.100

Packet 1 details:

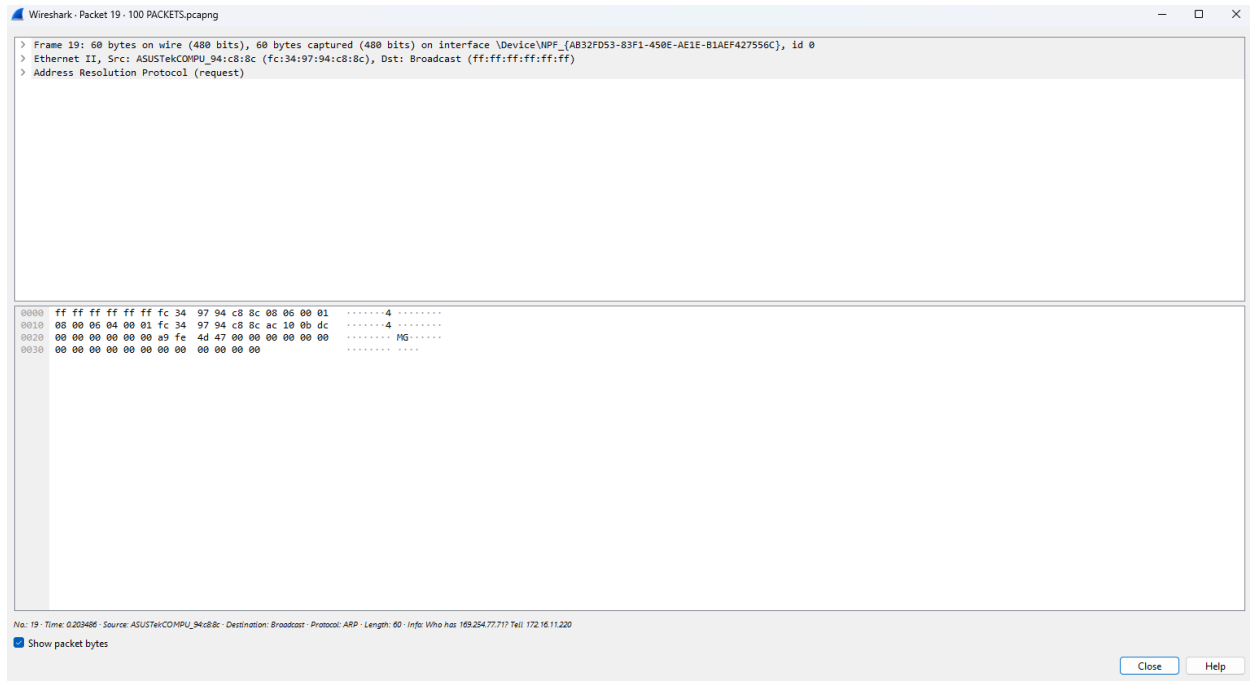
```

Frame 2: 60 bytes on wire (4800 bits), 60 bytes captured (4800 bits) on interface \Device\NPF_{A8327053-83F1-4508-AE8E-81AEF427556C}
  Ethernet II, Src: Dell_15:0f:98 (58:94:c3:15:0f:98), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    Address Resolution Protocol (request)
  
```

Packet 1 hex dump:

```

0000  ff ff ff ff ff ff 58 9a 4c 35 0f 98 00 00 00 01 .....P.LS.....
0010  00 00 00 04 00 01 58 9a 4c 35 0f 98 ac 10 08 6c .....P.LS.....1
0020  00 00 00 00 00 00 ac 10 32 ca 00 00 00 00 00 00 .....2.....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
  
```

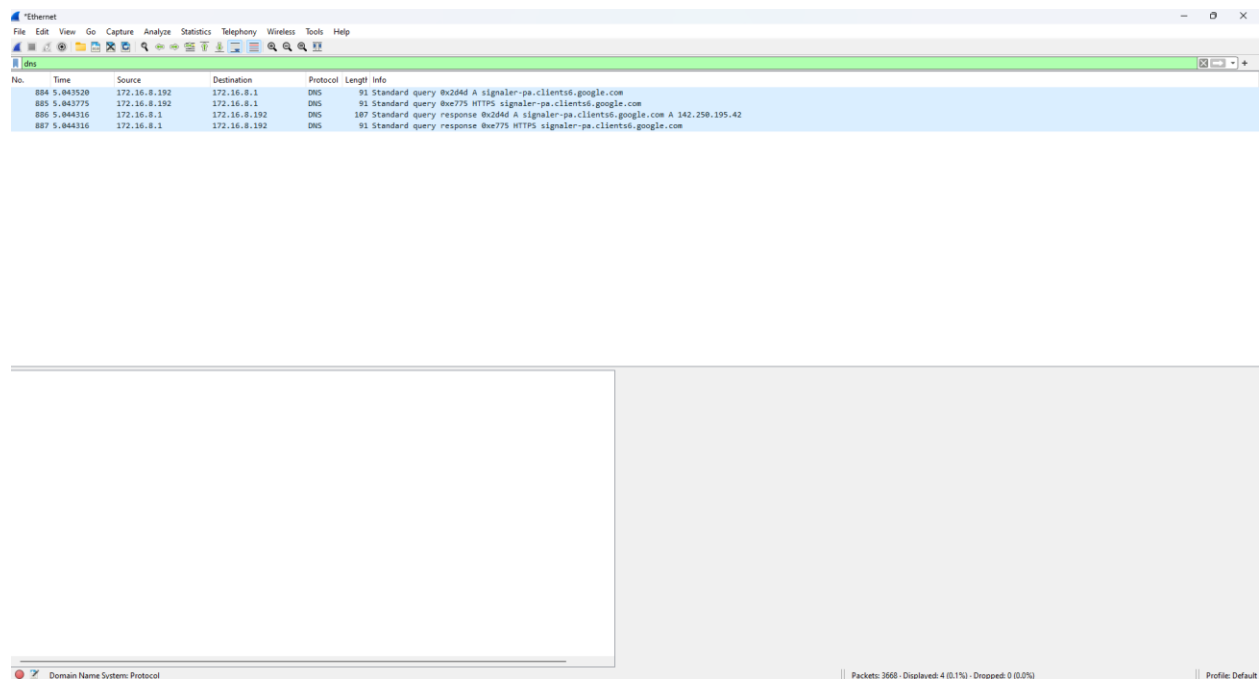


4.Create a Filter to display only DNS packets and provide the flow graph.

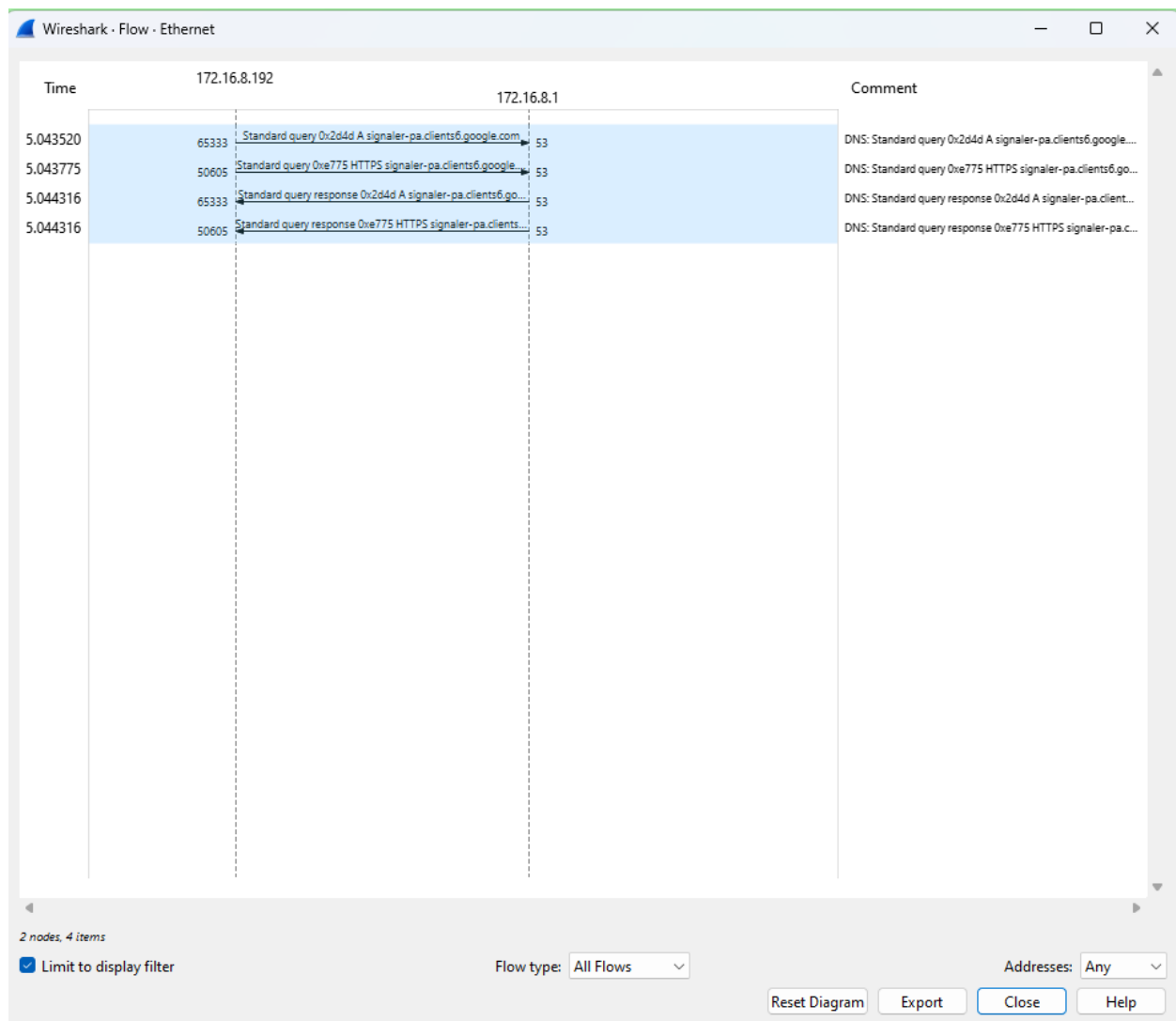
Procedure

- Select Local Area Connection in Wireshark.
- Go to capture ☐ option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search DNS packets in search bar.
- To see flow graph click Statistics<input type="checkbox"/>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.

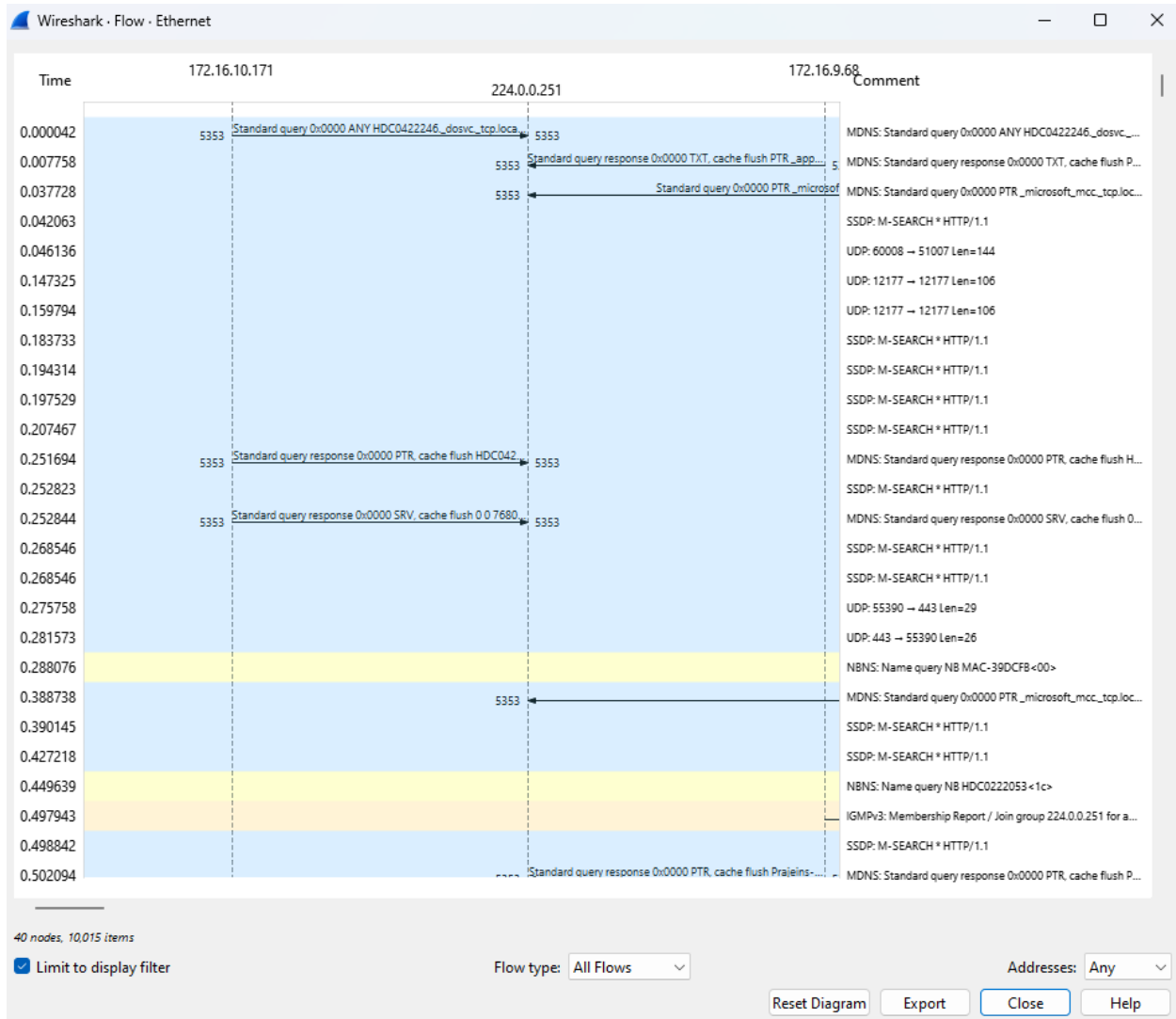
Procedure

- 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


Ethernet					
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help					
ipfilter					
No.	Time	Source	Destination	Protocol	Length Info
2	0.000042	172.16.10.171	224.0.0.251	PMNS	88 Standard query 0x0000 ANY HDC8422246._dosvc._tcp.local, "QI" question
3	0.000758	172.16.9.68	224.0.0.251	PMNS	358 Standard query response 0x0000 TXT, cache flush PTR _apple-mobdev2._tcp.local PTR 0:1fc7:d3:4f:7d:fe00:1f21fc7fff:fed3:4f7e-supports8P-19_apple-mobdev2._tcp.local PTR 0:1fc7:d3:4f:7d:fe00:1f21fc7fff:fed3:4f7e-supports8P-19_apple-mobdev2._tcp.local
5	0.037728	172.16.10.203	224.0.0.251	PMNS	85 Standard query 0x0000 PTR _microsoft._tcp.local, "QI" question
7	0.042063	172.16.8.161	239.255.255.250	SSDP	217 N-SEARCH * HTTP/1.1
8	0.046136	172.16.9.74	172.16.11.255	UDP	186 60000 → 51007 Len=144
13	0.147325	172.16.11.220	224.77.77.77	UDP	148 12177 → 12177 Len=186
14	0.159794	172.16.11.220	224.77.77.77	UDP	148 12177 → 12177 Len=186
15	0.163733	172.16.11.93	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
16	0.194314	172.16.9.45	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
17	0.197529	172.16.10.51	239.255.255.250	SSDP	217 N-SEARCH * HTTP/1.1
18	0.207467	172.16.8.30	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
19	0.251694	172.16.10.171	224.0.0.251	PMNS	365 Standard query response 0x0000 PTR, cache flush HDC8422246._dosvc._tcp.local SRV, cache flush 0 0 7680 HDC8422246.local TXT, cache flush A, cache flush 172.16.10.171 AAAA, cache flush f...
21	0.252823	172.16.10.35	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
23	0.252844	172.16.10.171	224.0.0.251	PMNS	386 Standard query response 0x0000 SRV, cache flush 0 0 7680 HDC8422246.local TXT, cache flush A, cache flush 172.16.10.171 AAAA, cache flush fe00:12ec:6777:47ae:e49d
24	0.265546	172.16.8.105	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
25	0.268546	172.16.8.65	239.255.255.250	SSDP	217 N-SEARCH * HTTP/1.1
27	0.275758	172.16.8.192	142.250.102.142	UDP	71 55390 → 443 Len=29
29	0.281571	142.208.182.142	172.16.8.192	UDP	68 443 → 55390 Len=26
30	0.288076	172.16.9.174	172.16.11.255	PMNS	92 Name query NB PAC-390CF8:000
32	0.388738	172.16.10.105	224.0.0.251	PMNS	85 Standard query 0x0000 PTR _microsoft._tcp.local, "QI" question
34	0.390145	172.16.8.7	239.255.255.250	SSDP	217 N-SEARCH * HTTP/1.1
37	0.427218	172.16.8.10	239.255.255.250	SSDP	216 N-SEARCH * HTTP/1.1
38	0.449639	172.16.10.44	172.16.11.255	PMNS	92 Name query NB HDC8222053:1c3
42	0.497943	172.16.9.68	224.0.0.22	IGMPv3	68 Membership Report / Join group 224.0.0.251 for any sources
43	0.498042	172.16.10.79	239.255.255.250	SSDP	217 N-SEARCH * HTTP/1.1
> Frame 2: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface [Device\NPF...A832F053-83F1-458E-AE1E-81AEF427556C]					
> Ethernet II, Src: ElitedropCo_14:83:9c (81ae:dd14:83:9c), Dst: IPv4cast_fb (01:00:5e:00:00:fb)					
> Internet Protocol Version 4, Src: 172.16.10.171, 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 08 ae dd 14 83 9c 06 00 45 00E: 0010 00 4a 9a 8e 00 00 01 11 87 5e ac 10 0a ab e0 003:..... 0020 00 f0 14 e9 14 e9 00 26 09 f7 00 00 00 00 016..... 0030 00 00 00 00 00 00 0a 68 44 43 10 34 32 32 34M DC842224 0040 36 06 5f 64 6f 73 76 63 04 5f 74 63 70 05 6c 6f 6 _dosvc _tcp lo 0050 63 61 6c 00 00 ff 00 01 cal					

Flow Graph outp



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

The image shows a Wireshark packet capture of DHCP traffic. The packet list on the left shows several DHCP messages. The packet details pane on the right shows the structure of a DHCP Request packet, including the Transaction ID field.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.1	255.255.255.255	DHCP	348	DHCP Request - Transaction ID #da5986d1
2	0.000000	192.168.1.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID #b0b349f6
3	0.000000	192.168.1.1	255.255.255.255	DHCP	362	DHCP Request - Transaction ID #b0b349f6
4	0.000000	192.168.1.1	255.255.255.255	DHCP	358	DHCP Request - Transaction ID #b0b349f6
5	0.000000	192.168.1.1	255.255.255.255	DHCP	364	DHCP Request - Transaction ID #b0b349f6
6	0.000000	192.168.1.1	255.255.255.255	DHCP	364	DHCP Request - Transaction ID #b0b349f6
7	0.000000	192.168.1.1	255.255.255.255	DHCP	364	DHCP Request - Transaction ID #b0b349f6
8	0.000000	192.168.1.1	255.255.255.255	DHCP	354	DHCP Discover - Transaction ID #b0b349f6
9	0.000000	192.168.1.1	255.255.255.255	DHCP	360	DHCP Request - Transaction ID #b0b349f6
10	0.000000	192.168.1.1	255.255.255.255	DHCP	364	DHCP Request - Transaction ID #b0b349f6

No.	dhcp	Source	Destination	Protocol	Length	Info
1	dhcpcd	0.0.0.0	255.255.255.255	DHCP	548	DHCP Request - Transaction ID 8bda5066d1
2	dhcpcd	24.8.1.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bda51af96
3	dhcpcd	255.255.255.255	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 8bda51af96
4	dhcpcd	255.255.255.255	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 8bda51af96
5	dhcpcd	255.255.255.255	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bda52067c
11168	128.92.2155	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bda52067c
11894	131.289574	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bda52067c
13395	147.161633	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bda52067c
14243	147.338071	0.0.0.0	255.255.255.255	DHCP	354	DHCP Discover - Transaction ID 8bda52067c
15612	148.550346	0.0.0.0	255.255.255.255	DHCP	366	DHCP Request - Transaction ID 8bda52067c
16920	168.181384	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bda52067c

```

> Frame 1887: 348 bytes on wire (2784 bits), 348 bytes captured (2784 bits) on interface Device\NPF_{AB32F053-83F1-4506-AE1E-B3AEF}
> Ethernet II, Src: Intel_ba:91:28:a0:b8:69:be:91:28, Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> 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 a0 69 be 91 28 00 00 45 00  .....I(-E-
0010  01 4e e2 00 00 00 00 11 56 ef 00 00 00 00 ff  ff  -N-----V-----
0020  00 ff 00 48 00 43 01 3a ef 23 01 00 00 00 4a 59  .....D-C-I-.....Y
0030  8e 21 00 00 00 00 00 00 00 00 00 00 00 00 00  .....S(-C-----
0040  00 00 00 00 00 a0 00 00 69 be 91 28 00 00 00 00  .....S(-C-----
0050  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  .....
0070  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  .....
0090  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  .....
00b0  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  .....
00d0  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  .....
00f0  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  .....
0110  00 00 00 00 00 00 63 82 53 63 35 01 03 3d 07 01  .....-C-ScS-...
0120  a0 00 69 be 91 28 32 04 ac 18 00 59 0c 07 49 49  .....I(-2-----Y-II
0130  49 43 52 45 43 51 0a 00 00 49 49 49 43 52 45  ICRQQ-...IIICR
0140  43 3c 08 46 53 46 20 34 25 2e 38 37 0e 01 03  C<PQPT 5.07...
0150  0f 1f 21 2b 2c 2e 2f 79 7b fc ff                -1p,r,y

```

Wireshark - Packet 11168 - Ethernet

> Frame 11168: 364 bytes on wire (2912 bits), 364 bytes captured (2912 bits) on interface \Device\NPF_{AB32FD53-83F1-450E-AE1E-B1AEF427556C}, id 0

> Ethernet II, Src: ASIXElectron_e2:ee:ab (20:7b:d2:e2:ee:ab), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> 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 00 7b d2 e2 ee ab 00 00 45 00  ....E-
0010  01 5e 02 7b 00 00 00 11 37 15 00 00 00 00 ff ff  -A- { 7 .....
0020  ff ff 00 44 00 43 01 4a 52 00 01 01 00 00 86 26  --D-C- R .....&
0030  b7 c7 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0040  00 00 00 00 00 20 7b d2 e2 ee ab 00 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 63 82 53 63 35 01 03 3d 07 01  ....c-Sc5-...
0120  20 7b d2 e2 ee ab 32 04 ac 10 09 4a 0c ff 44 45  {-2-...-DE
0130  53 4b 54 4f 50 2d 46 53 33 50 55 4c 33 51 12 00  SKTOP-FS 3PUL30-
0140  00 00 44 45 53 4b 54 4f 50 2d 46 53 33 50 55 4c  DESKTOP-FS3PUL
0150  33 3c 08 4d 53 46 20 3e 20 30 37 0e 01 03 06  3c-MSFT 5.07-...
0160  0f 1f 21 2b 2c 2e 2f 77 79 f9 fc ff  --!+./W y-...

```

