EXP NO: 4A STUDY OF WIRESHARK TOOL ENVIRONMENT

DATE:13/08/2024

AIM:

To study packet sniffing concepts using Wireshark Tool.

DESCRIPTION:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets. You can use Wireshark to inspect a suspicious program's network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

What we can do with Wireshark:

- Capture network traffic
- Decode packet protocols using dissectors
- Define filters – capture and display
- Watch smart statistics
- Analyze problems
- Interactively browse that traffic

Wireshark used for:

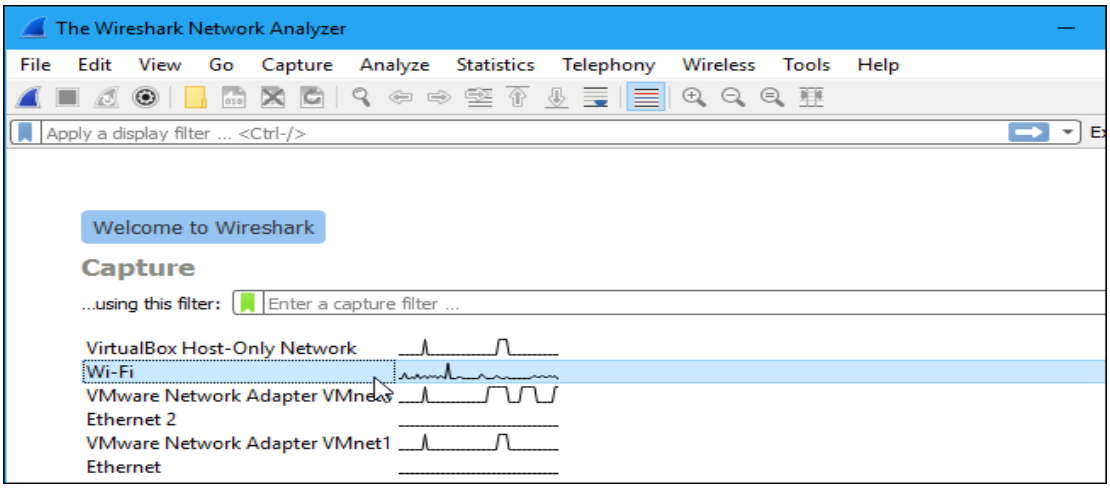
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

Getting Wireshark

Wireshark can be downloaded for Windows or macOS from its official website. For Linux or another UNIX-like system, Wireshark will be found in its package repositories. For Ubuntu, Wireshark will be found in the Ubuntu Software Center.

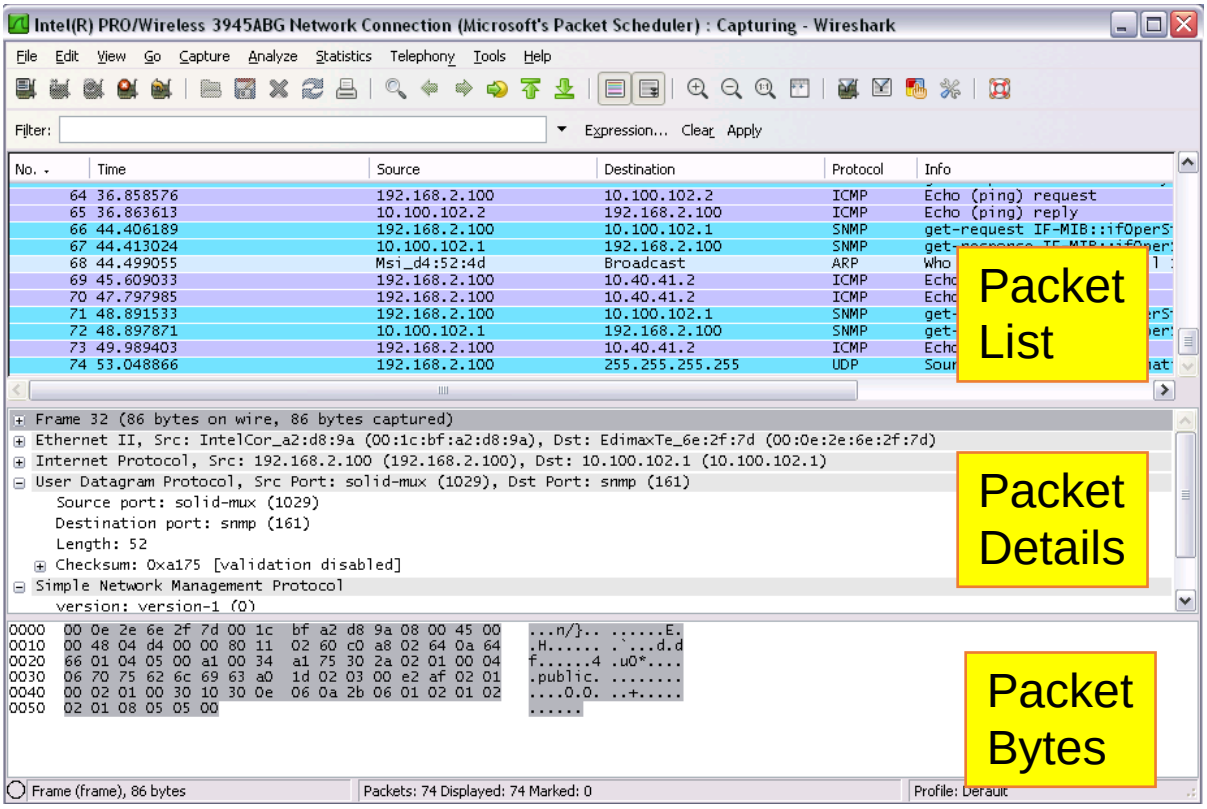
Capturing Packets

After downloading and installing Wireshark, launch it and double-click the name of a network interface under Capture to start capturing packets on that interface



As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

If you have promiscuous mode enabled—it's enabled by default—you'll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture > Options and verify the "Enable promiscuous mode on all interfaces" checkbox is activated at the bottom of this window.



Click the red "Stop" button near the top left corner of the window when you want to stop capturing traffic.

The "Packet List" Pane

The packet list pane displays all the packets in the current capture file. The "Packet List" pane Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes.

The "Packet Details" Pane

The packet details pane shows the current packet (selected in the "Packet List" pane) in a more detailed form. This pane shows the protocols and protocol fields of the packet selected in the "Packet List" pane. The protocols and fields of the packet shown in a tree which can be expanded and collapsed.

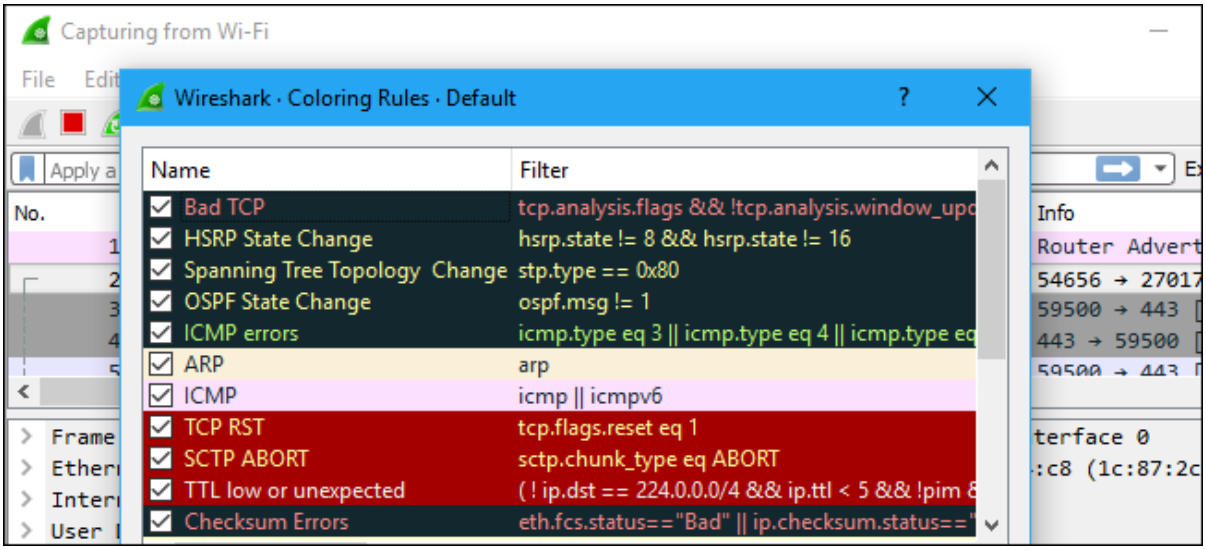
The "Packet Bytes" Pane

The packet bytes pane shows the data of the current packet (selected in the "Packet List" pane) in a hexdump style.

Color Coding

You'll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order.

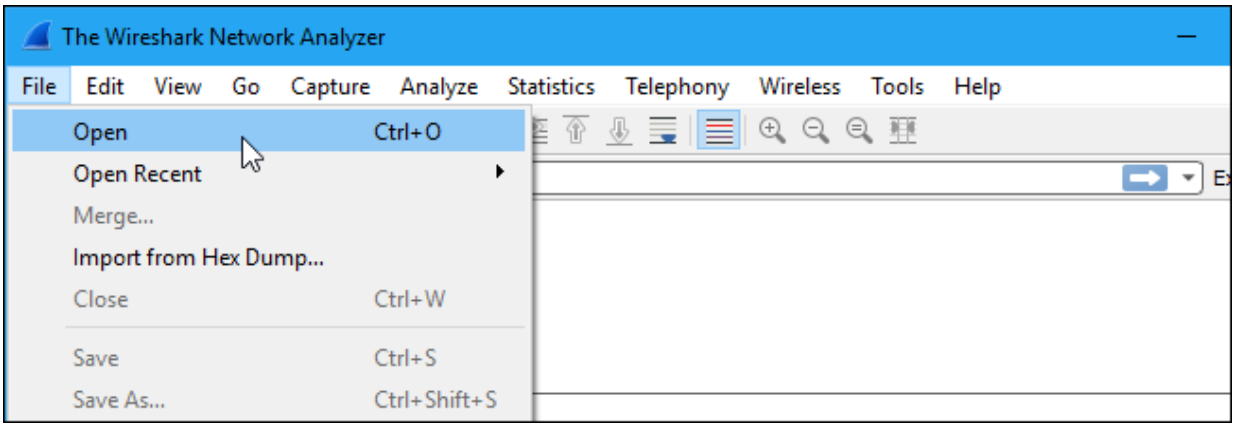
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



Sample Captures

If there's nothing interesting on your own network to inspect, Wireshark's wiki has you covered. The wiki contains a page of sample capture files that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.

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

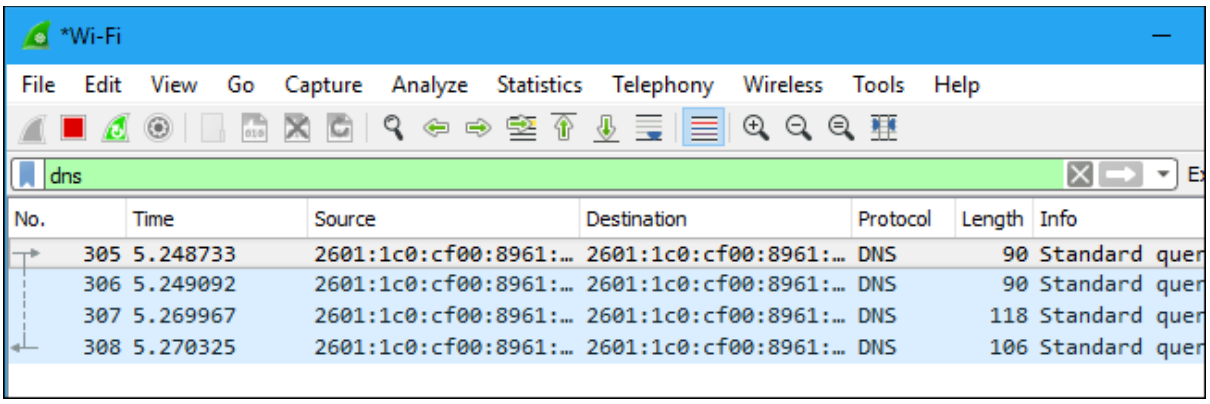


Filtering Packets

If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the

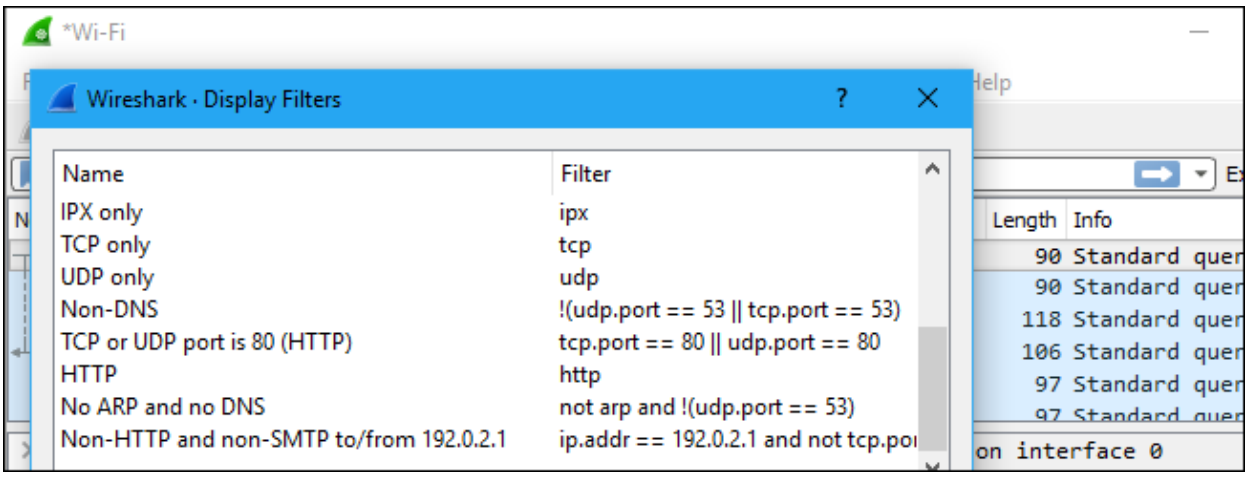
traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



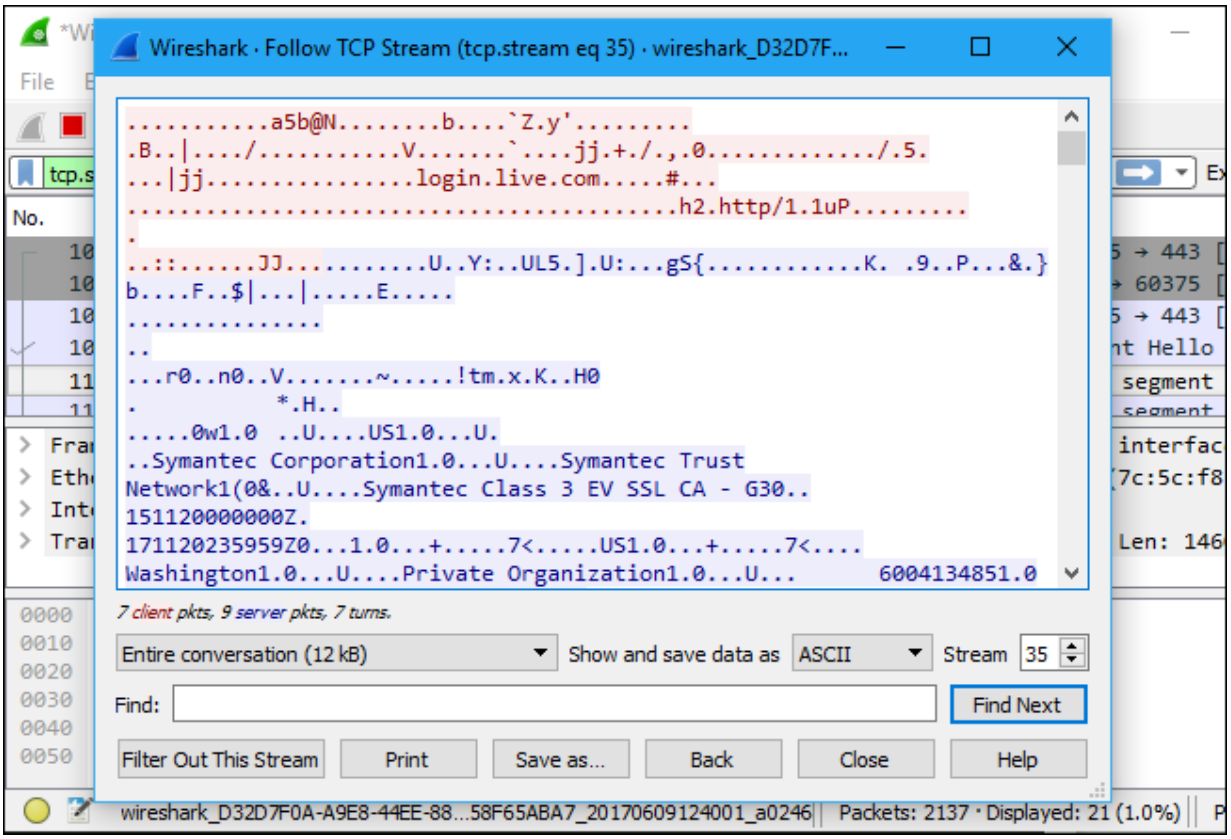
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

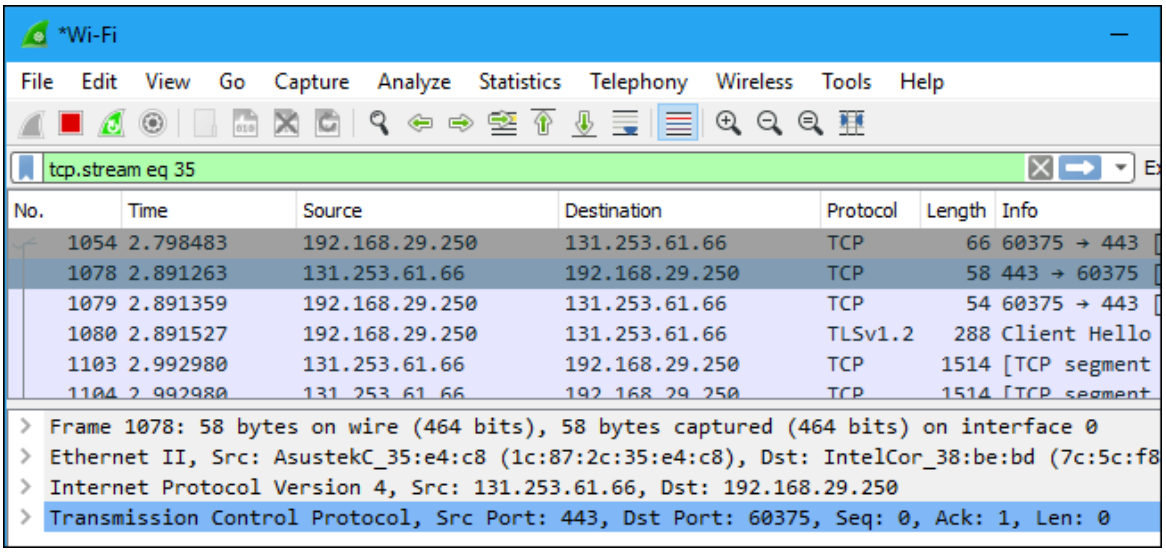


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

You'll see the full TCP conversation between the client and the server. You can also click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.

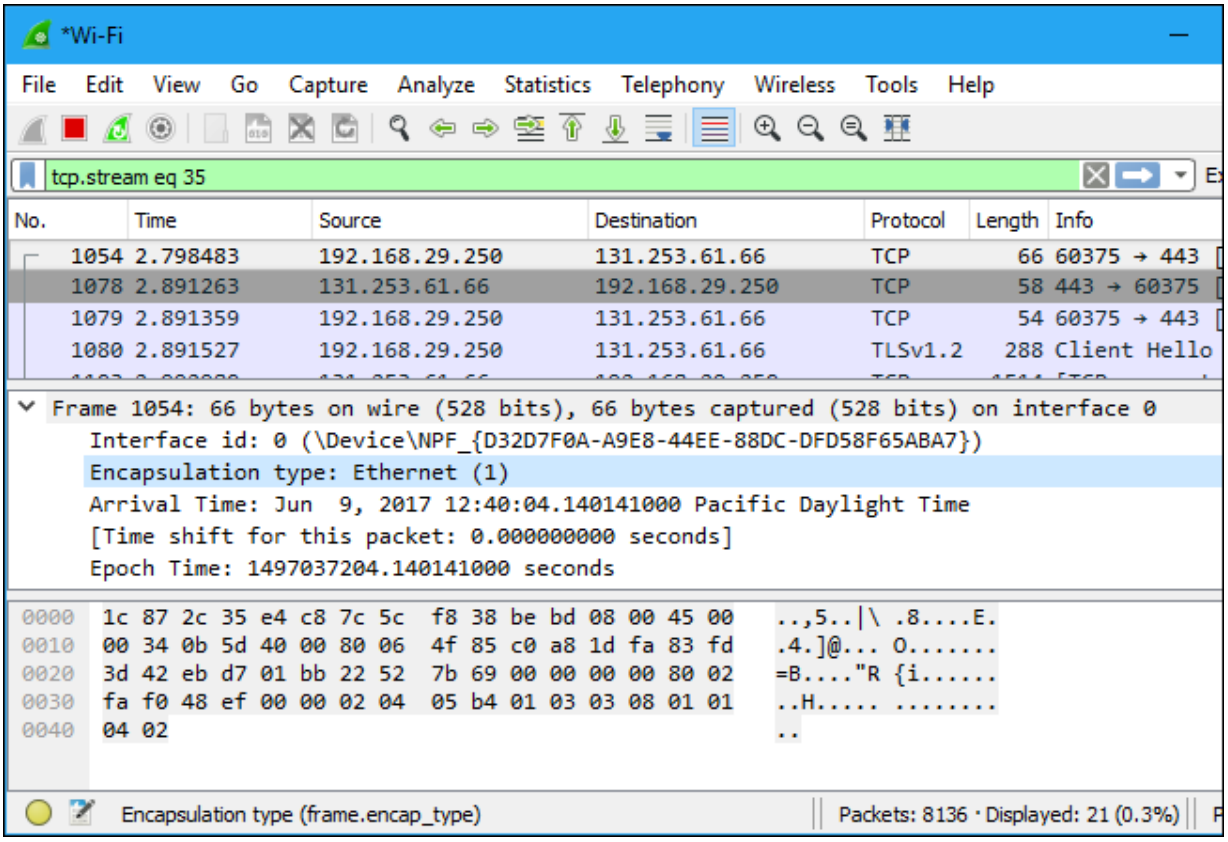


Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

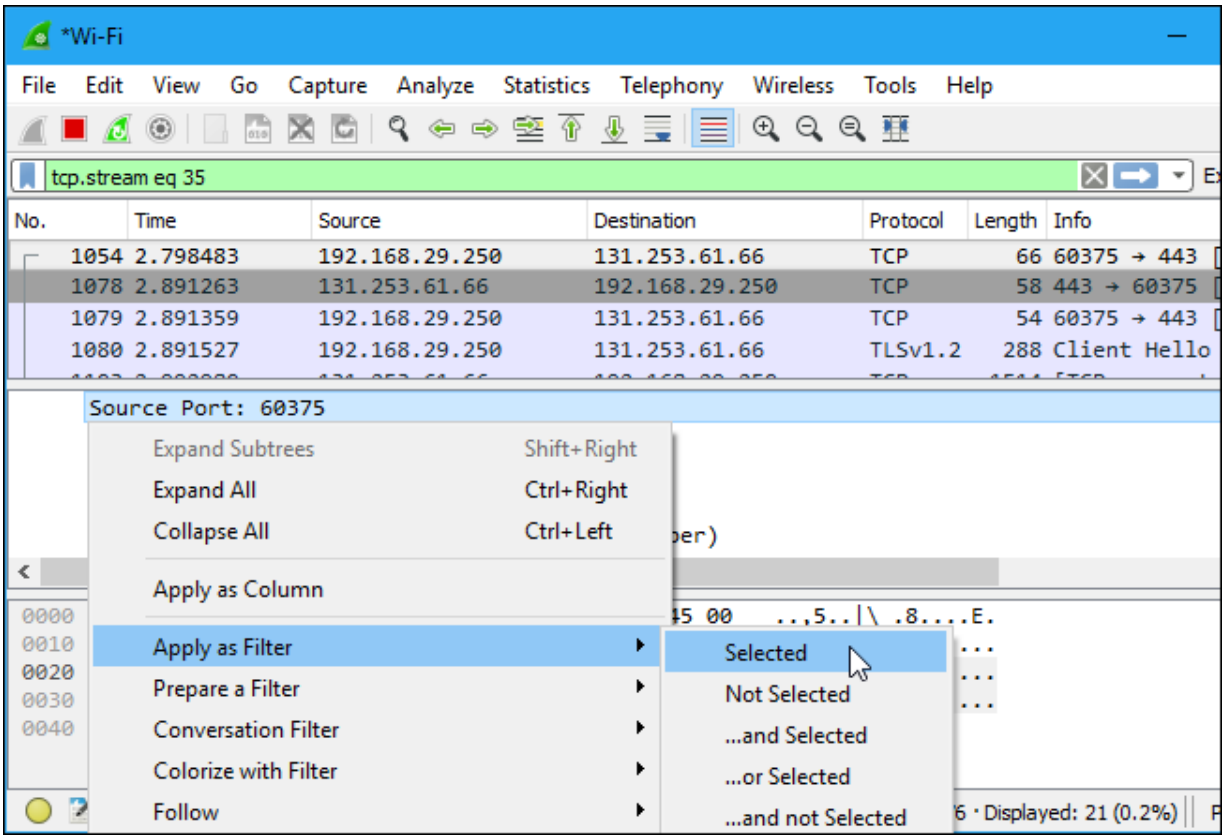


Inspecting Packets

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

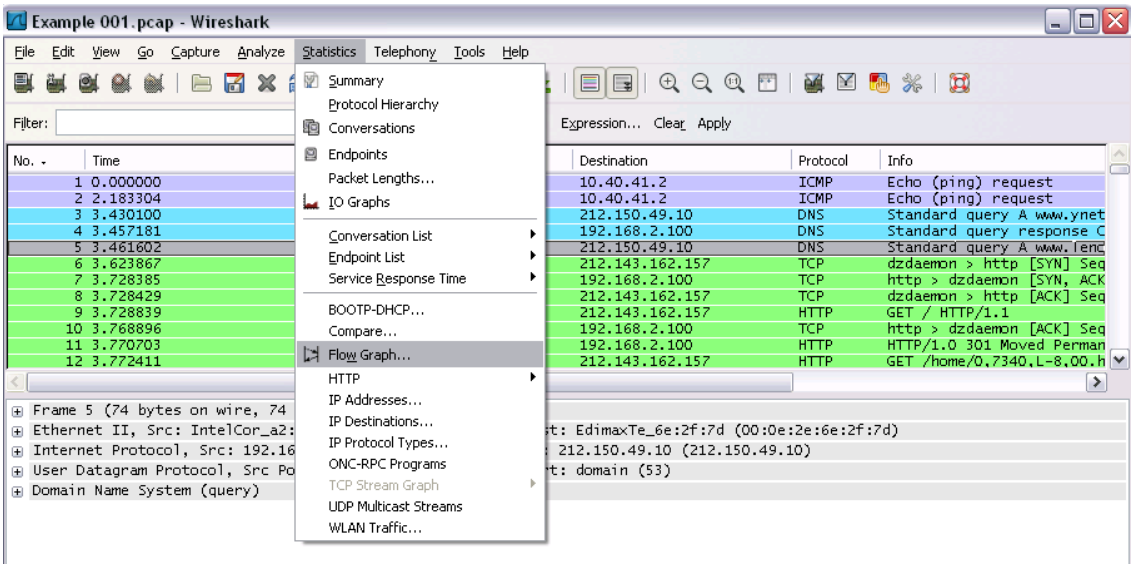


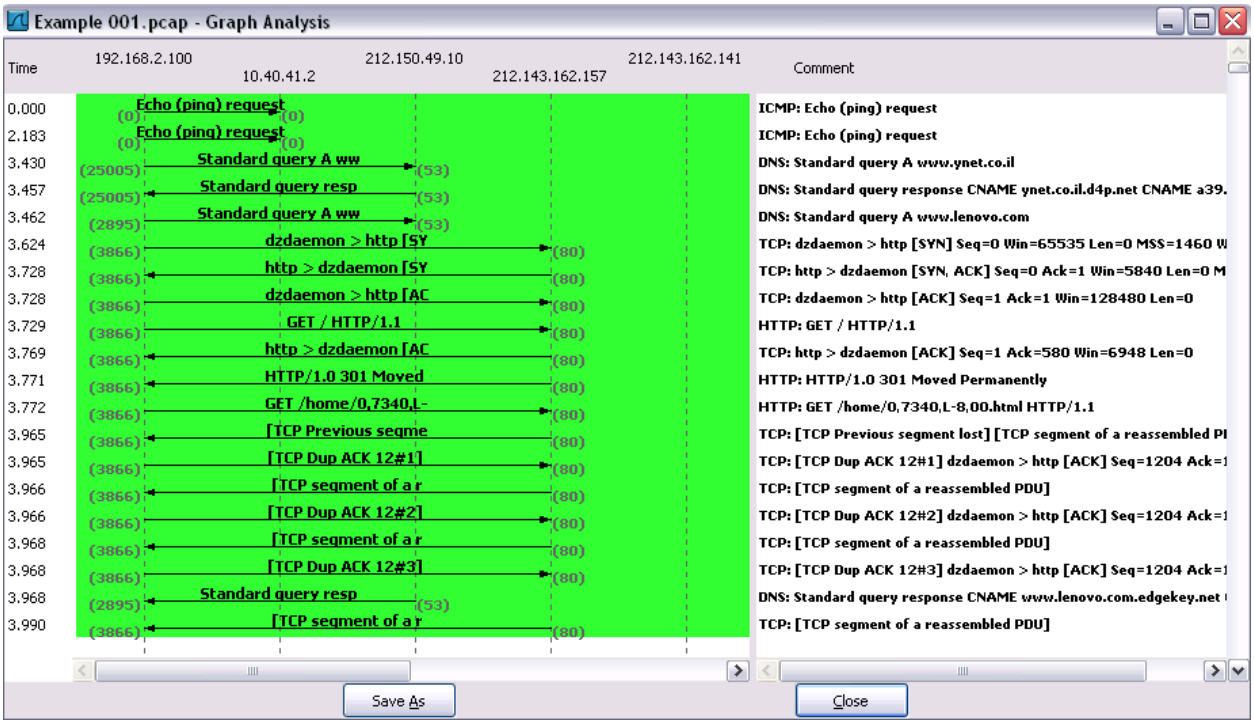
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



Wireshark is an extremely powerful tool, and this tutorial is just scratching the surface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

Flow Graph: Gives a better understanding of what we see.





Result:

packet sniffing concepts using Wireshark Tool is studied.

EXP NO:4B ANALYZE NETWORK TRAFFIC USING WIRESHARK TOOL

DATE:19/08/2024

AIM:

To capture, save, filter and analyze network traffic on TCP / UDP / IP / HTTP / ARP /DHCP /ICMP /DNS using Wireshark Tool

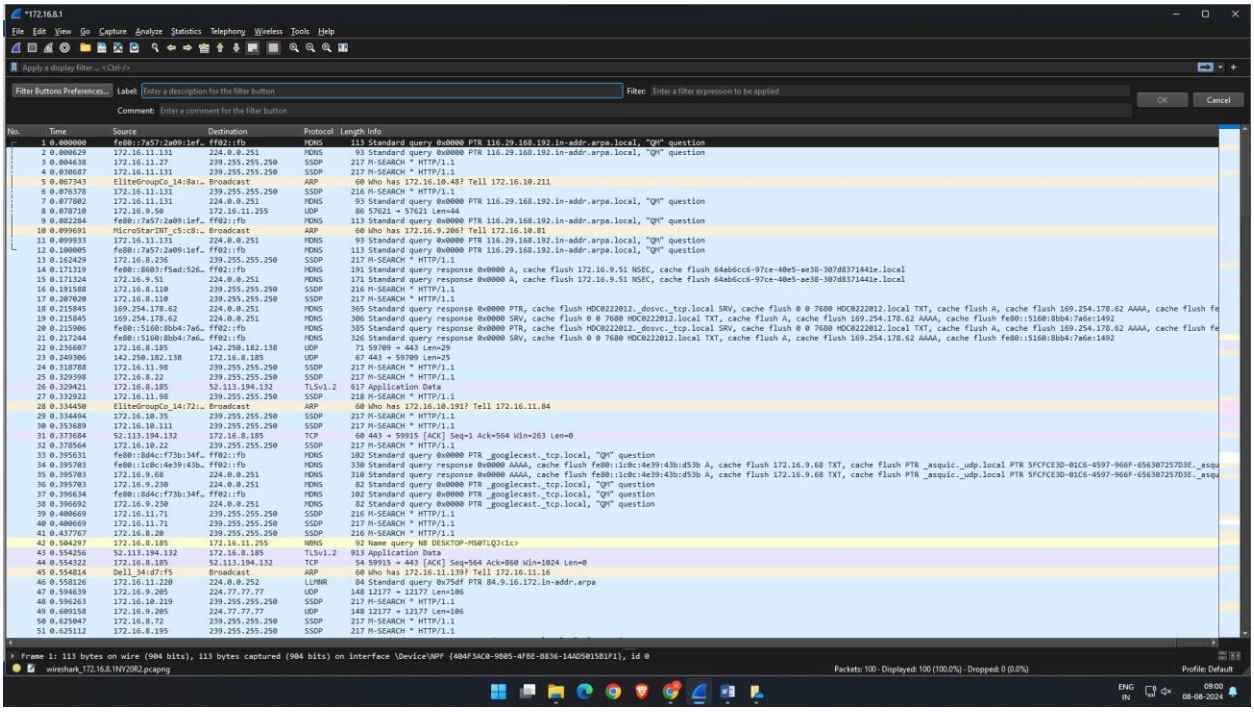
Exercises

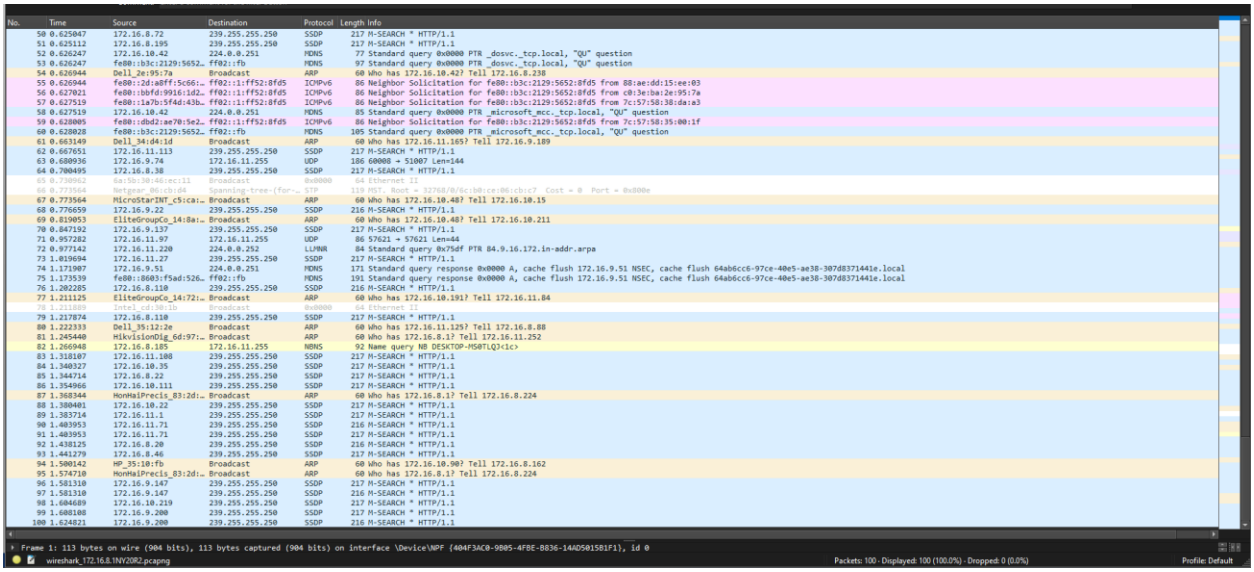
1. Capture 100 packets from the Ethernet: IEEE 802.3 LAN Interface and save it.

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



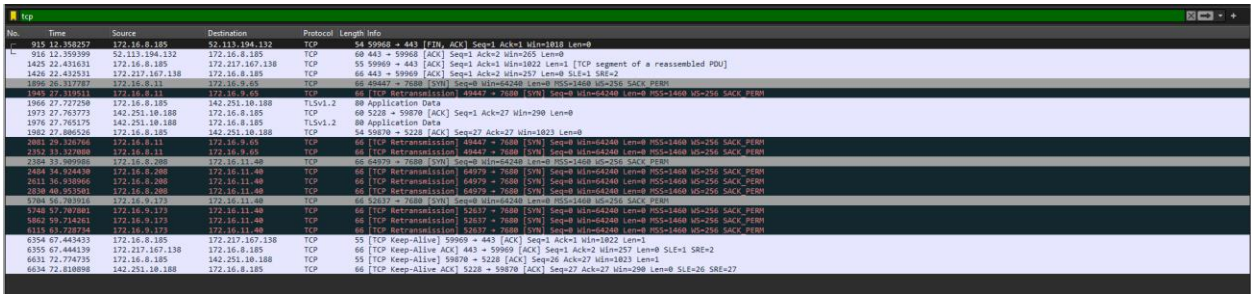


2.Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph.

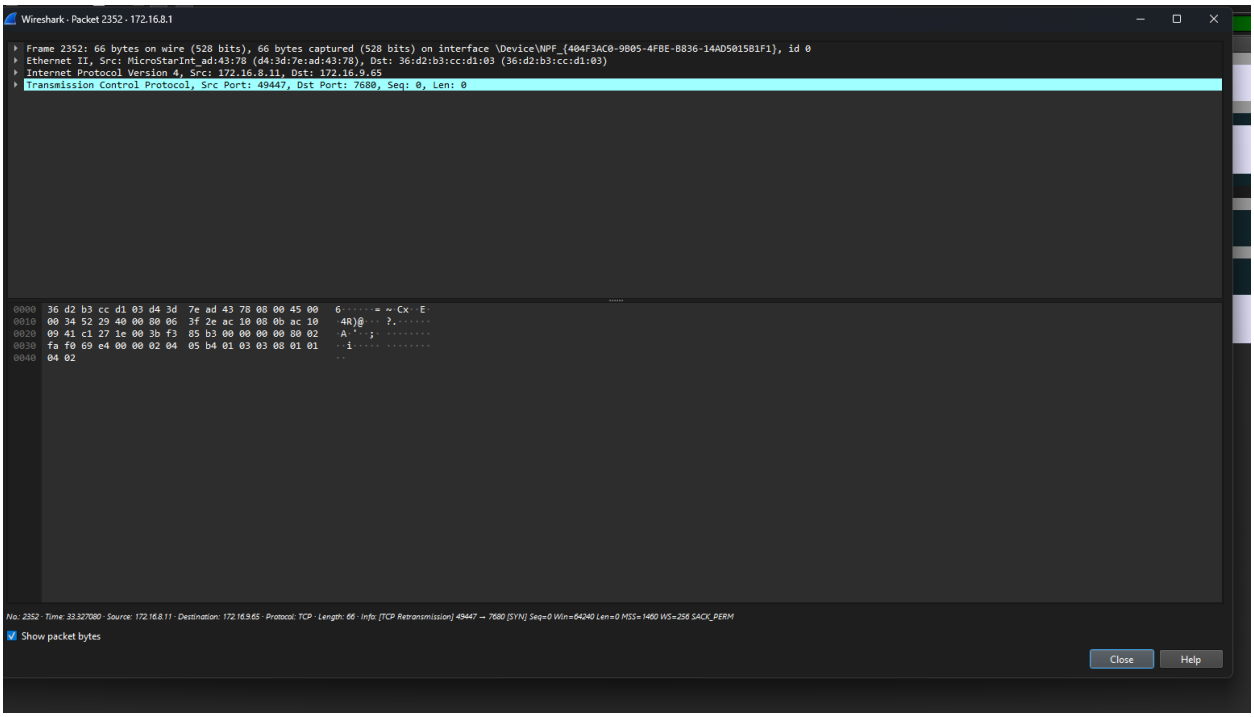
Procedure

- 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:udp

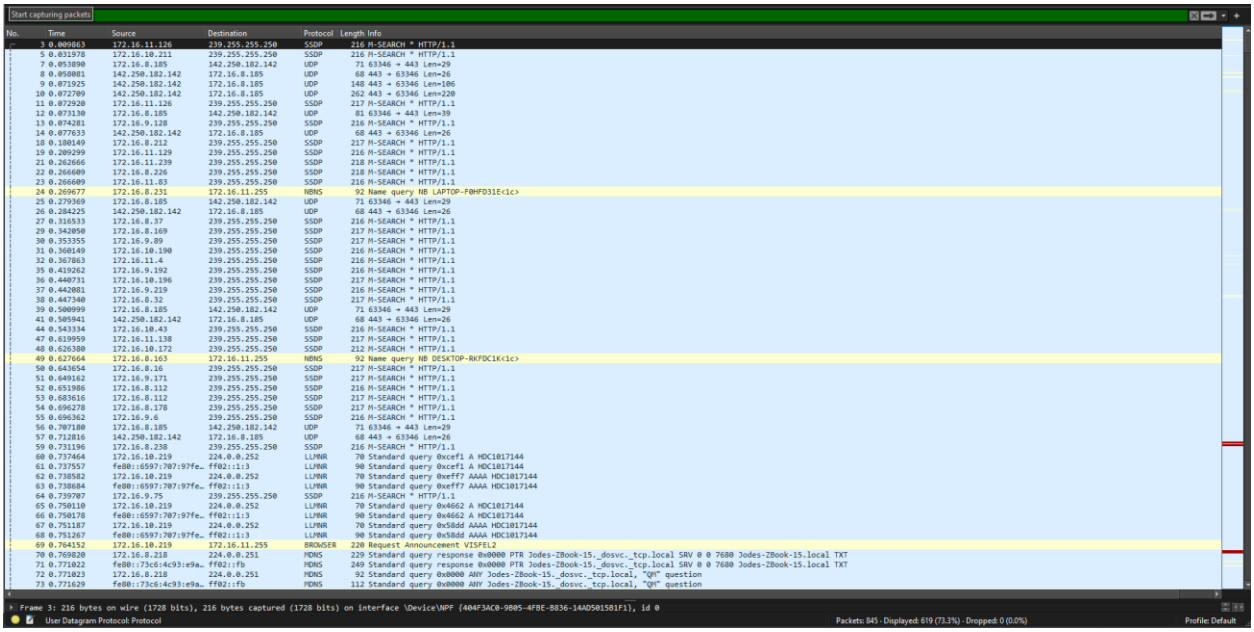


Inspecting the packets

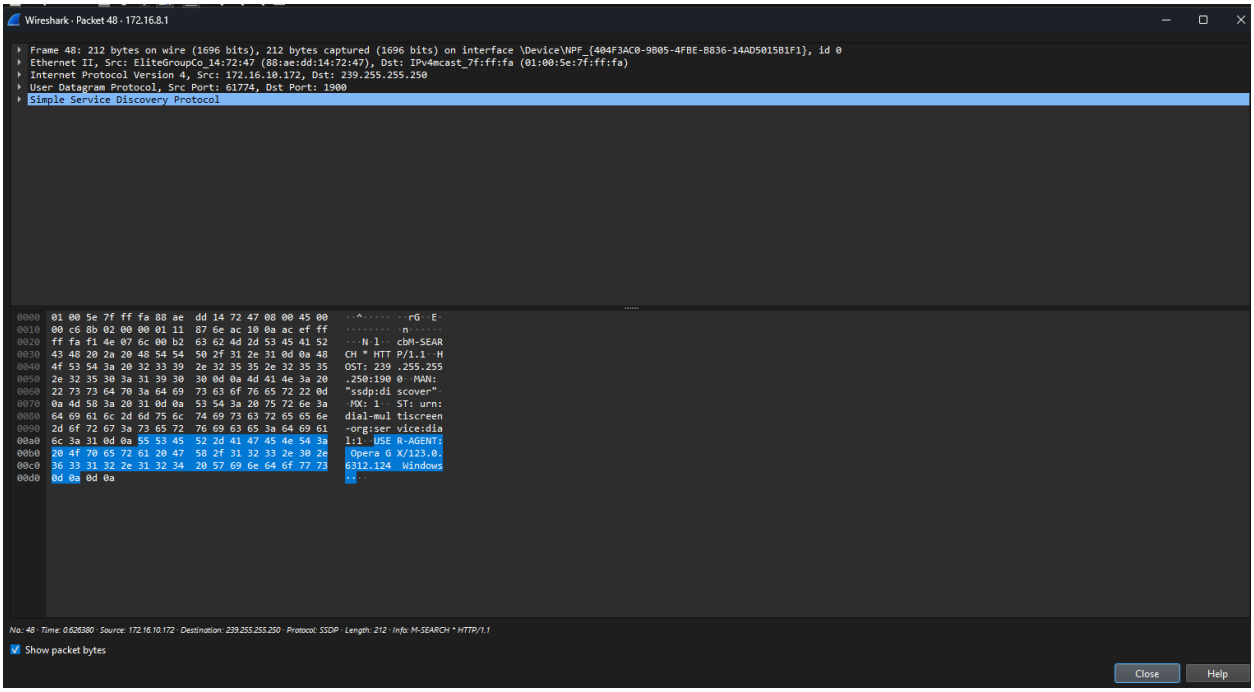


Flow Graph output

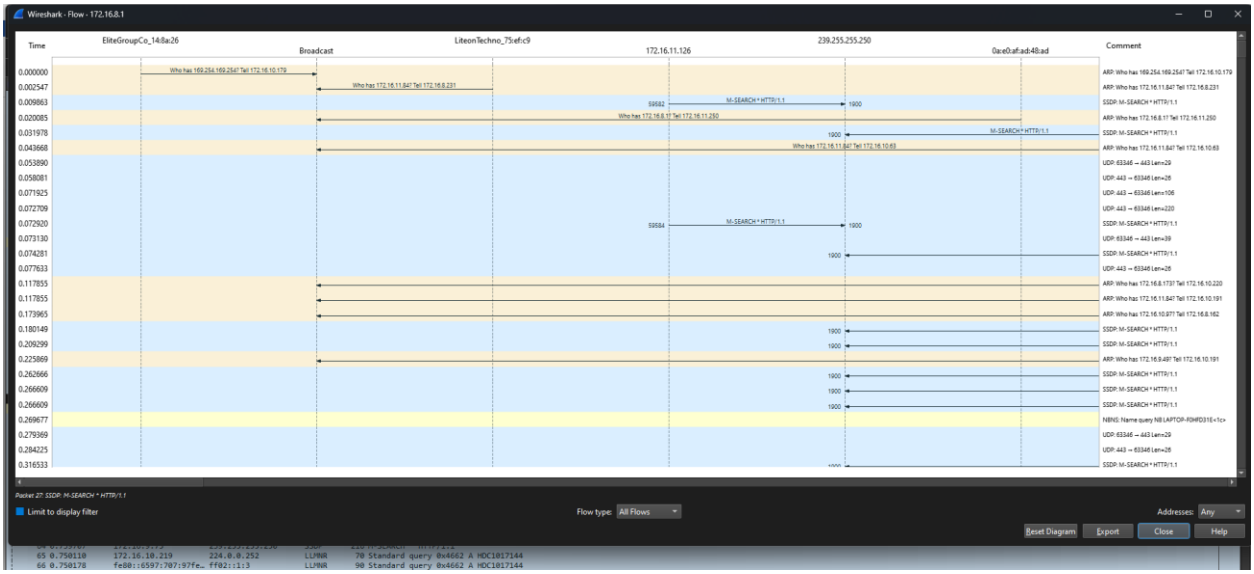
Output:tcp



Inspecting the packets



Flow chart output

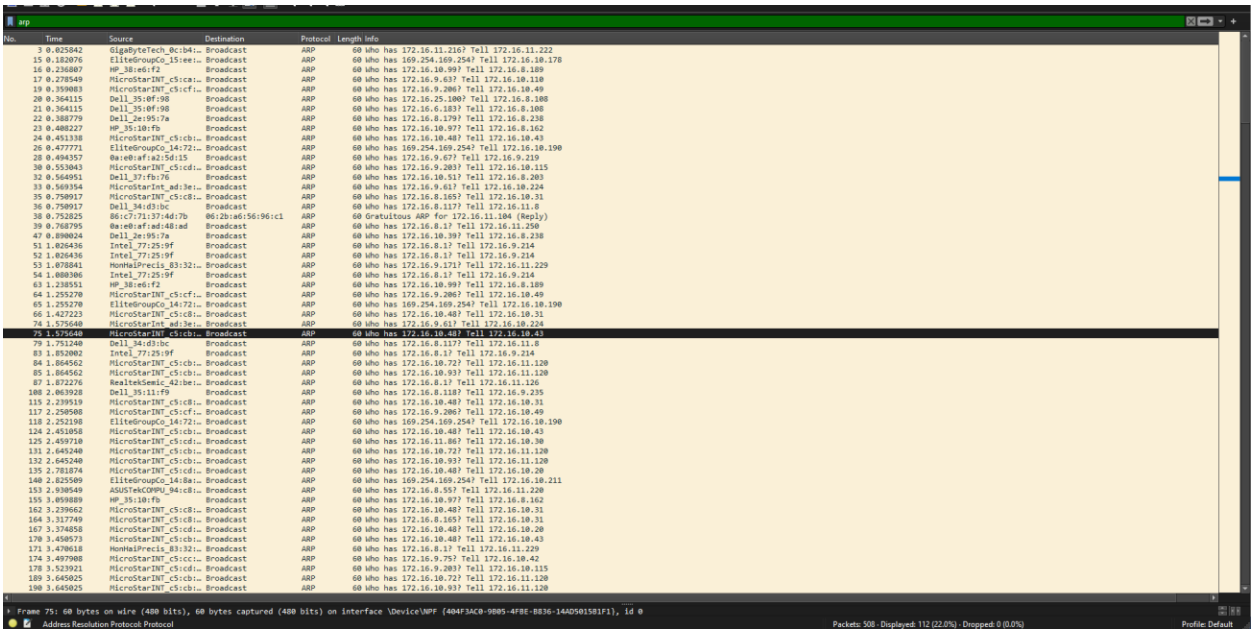


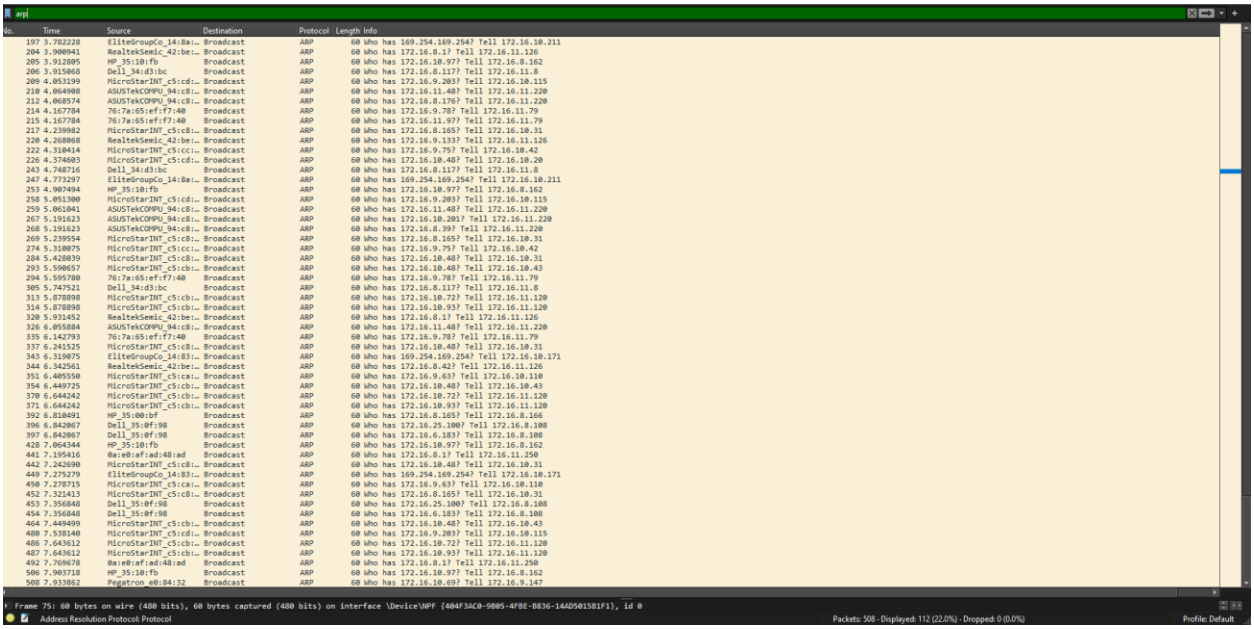
3.Create a Filter to display only ARP packets and inspect the packets.

Procedure

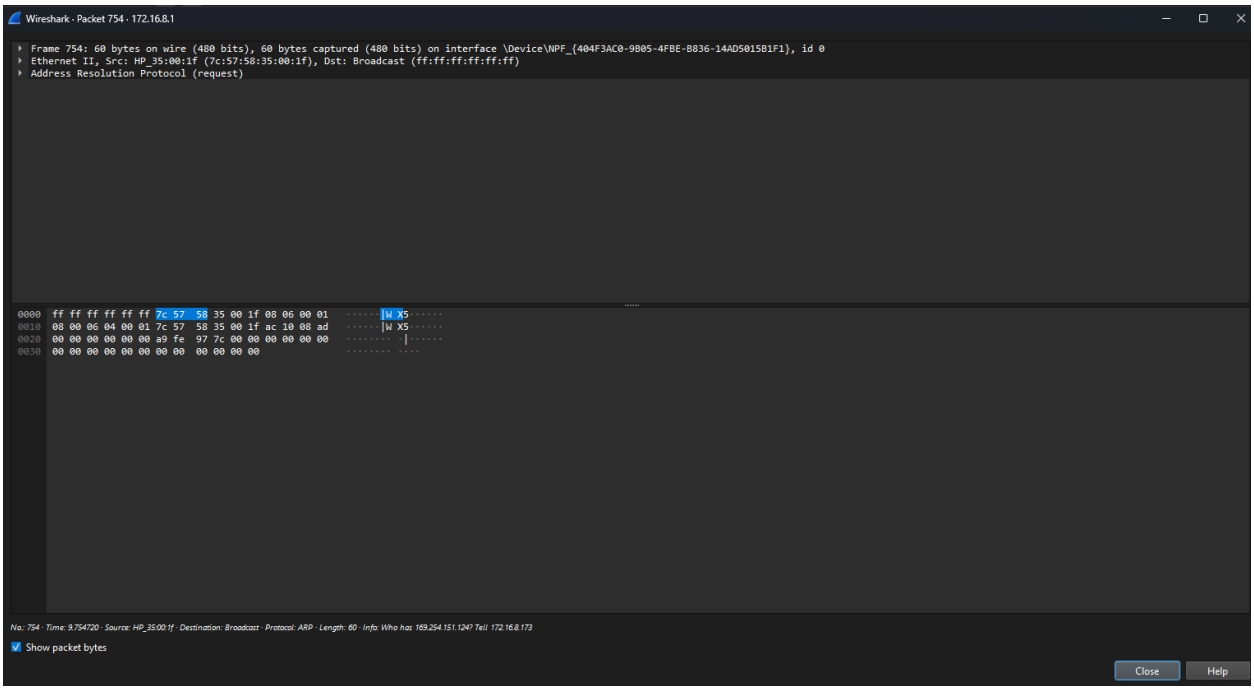
- Select Local Area Connection in Wireshark.
- Go to capture option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ARP packets in search bar.
- Save the packets.

Output





Inspecting the packets

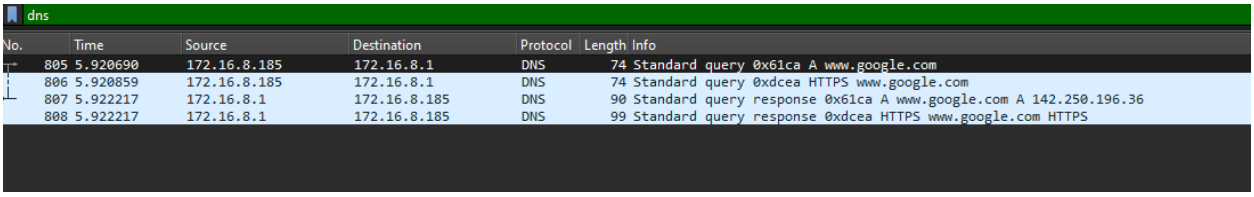


4.Create a Filter to display only DNS packets and provide the flow graph.

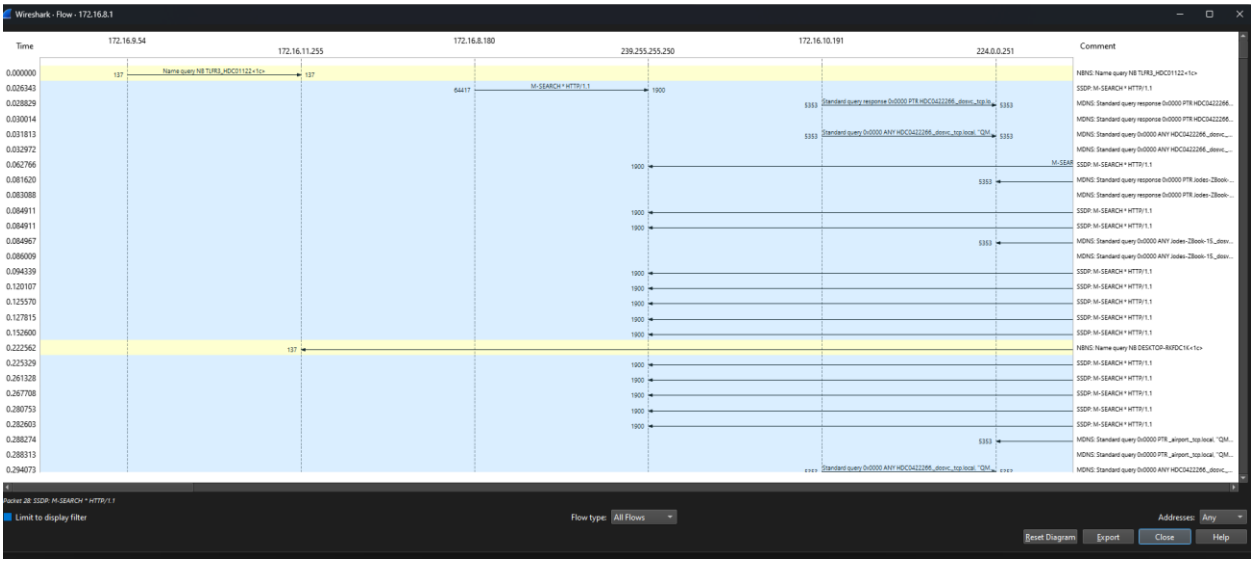
Procedure

- 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

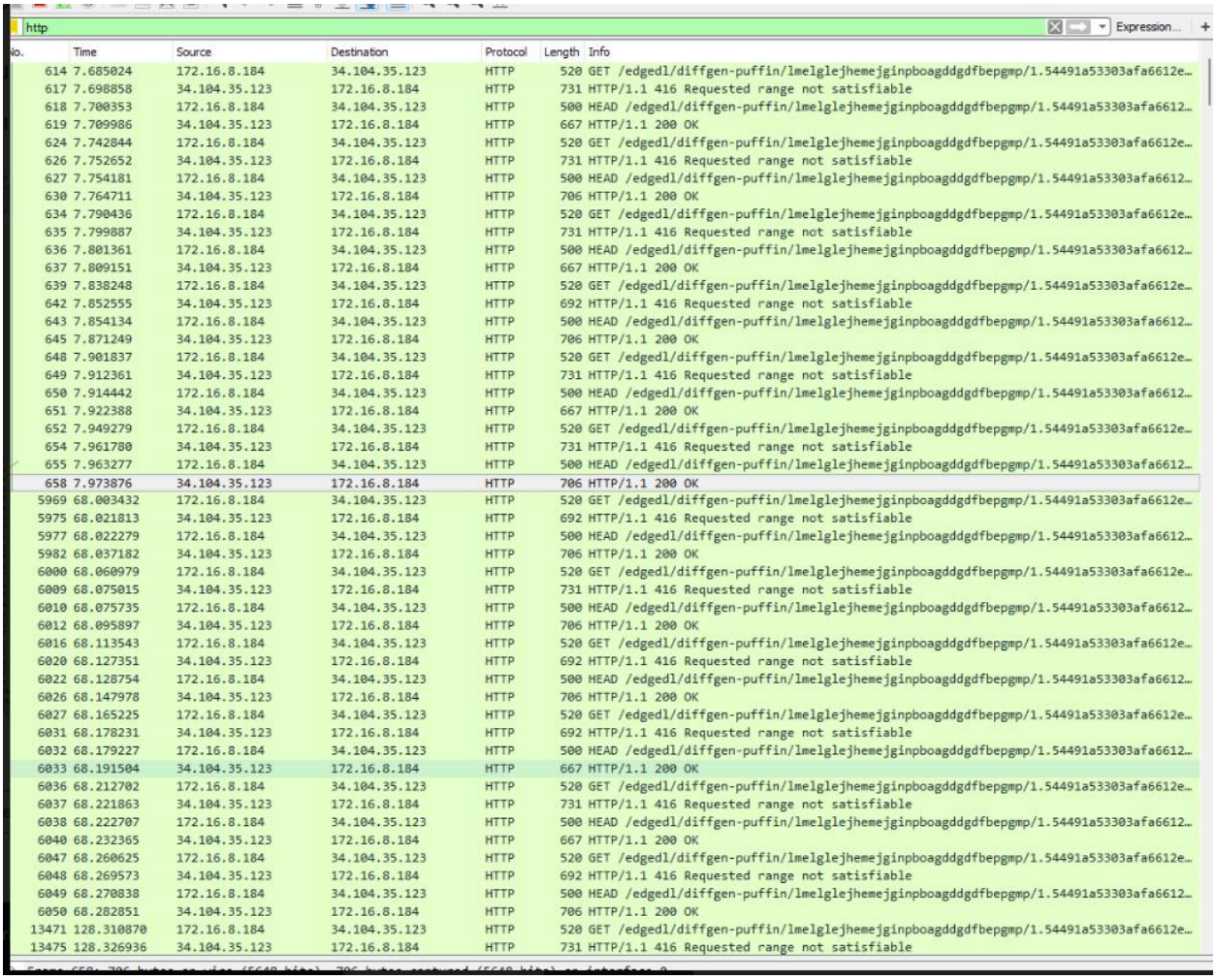


5.Create a Filter to display only HTTP packets and inspect the packets

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

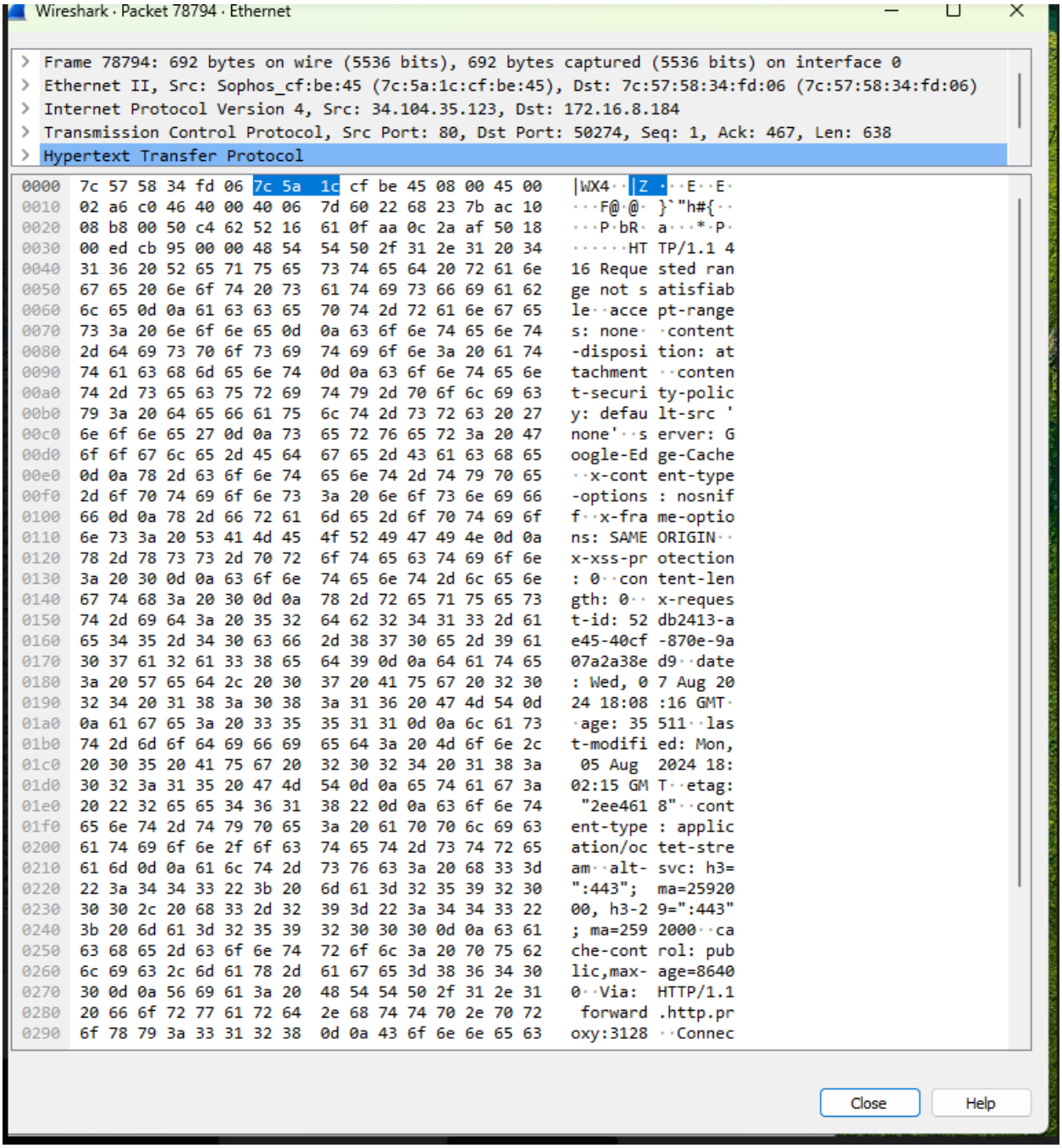


http

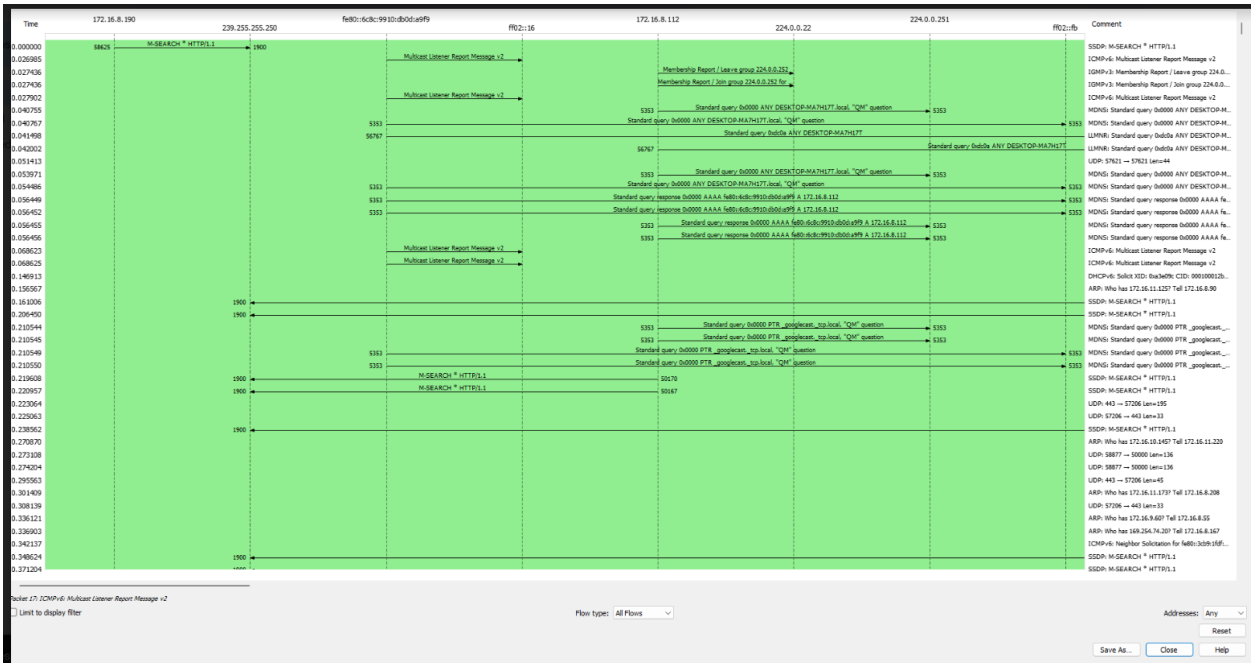
Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
614	7.685024	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
617	7.698858	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
618	7.700353	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
619	7.709986	34.104.35.123	172.16.8.184	HTTP	667	HTTP/1.1 200 OK
624	7.742844	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
626	7.752652	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
627	7.754181	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
630	7.764711	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
634	7.790436	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
635	7.799887	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
636	7.801361	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
637	7.809151	34.104.35.123	172.16.8.184	HTTP	667	HTTP/1.1 200 OK
639	7.838248	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
642	7.852555	34.104.35.123	172.16.8.184	HTTP	692	HTTP/1.1 416 Requested range not satisfiable
643	7.854134	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
645	7.871249	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
648	7.901837	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
649	7.912361	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
650	7.914442	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
651	7.922388	34.104.35.123	172.16.8.184	HTTP	667	HTTP/1.1 200 OK
652	7.949279	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
654	7.961780	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
655	7.963277	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
658	7.973876	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
5969	68.003432	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
5975	68.021813	34.104.35.123	172.16.8.184	HTTP	692	HTTP/1.1 416 Requested range not satisfiable
5977	68.022279	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
5982	68.037182	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
6000	68.060979	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
6009	68.075015	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
6010	68.075735	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
6012	68.095897	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
6016	68.113543	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
6020	68.127351	34.104.35.123	172.16.8.184	HTTP	692	HTTP/1.1 416 Requested range not satisfiable
6022	68.128754	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
6026	68.147978	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
6027	68.165225	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
6031	68.178231	34.104.35.123	172.16.8.184	HTTP	692	HTTP/1.1 416 Requested range not satisfiable
6032	68.179227	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
6033	68.191504	34.104.35.123	172.16.8.184	HTTP	667	HTTP/1.1 200 OK
6036	68.212702	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
6037	68.221863	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable
6038	68.222707	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
6040	68.232365	34.104.35.123	172.16.8.184	HTTP	667	HTTP/1.1 200 OK
6047	68.260625	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
6048	68.269573	34.104.35.123	172.16.8.184	HTTP	692	HTTP/1.1 416 Requested range not satisfiable
6049	68.270838	172.16.8.184	34.104.35.123	HTTP	500	HEAD /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612…
6050	68.282851	34.104.35.123	172.16.8.184	HTTP	706	HTTP/1.1 200 OK
13471	128.310870	172.16.8.184	34.104.35.123	HTTP	520	GET /edgedl/diffgen-puffin/lmelglejhemejginpboagddgdfbepgmp/1.54491a53303afa6612e…
13475	128.326936	34.104.35.123	172.16.8.184	HTTP	731	HTTP/1.1 416 Requested range not satisfiable

Inspecting the packets



Flow Graph output

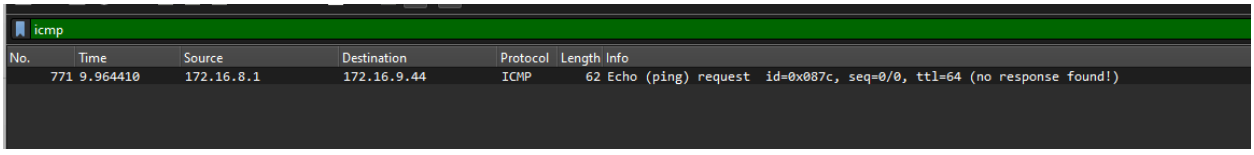


6.Create a Filter to display only IP/ICMP packets and inspect the packets.

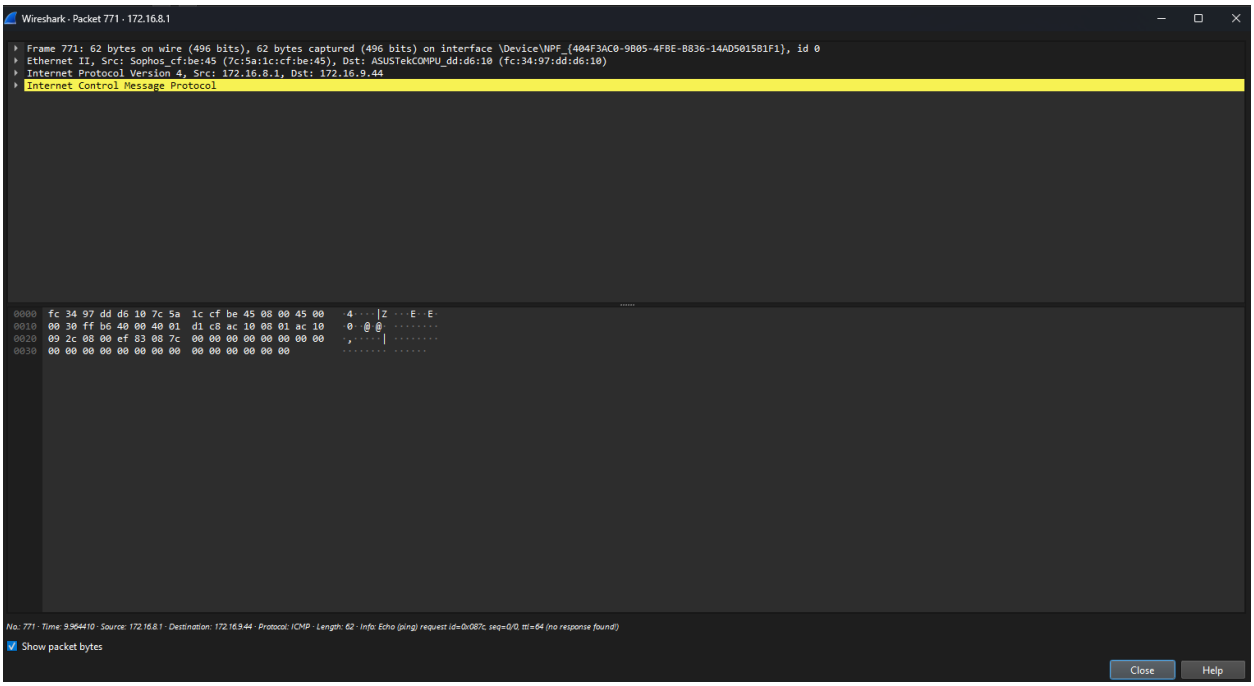
Procedure

- 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:icmp

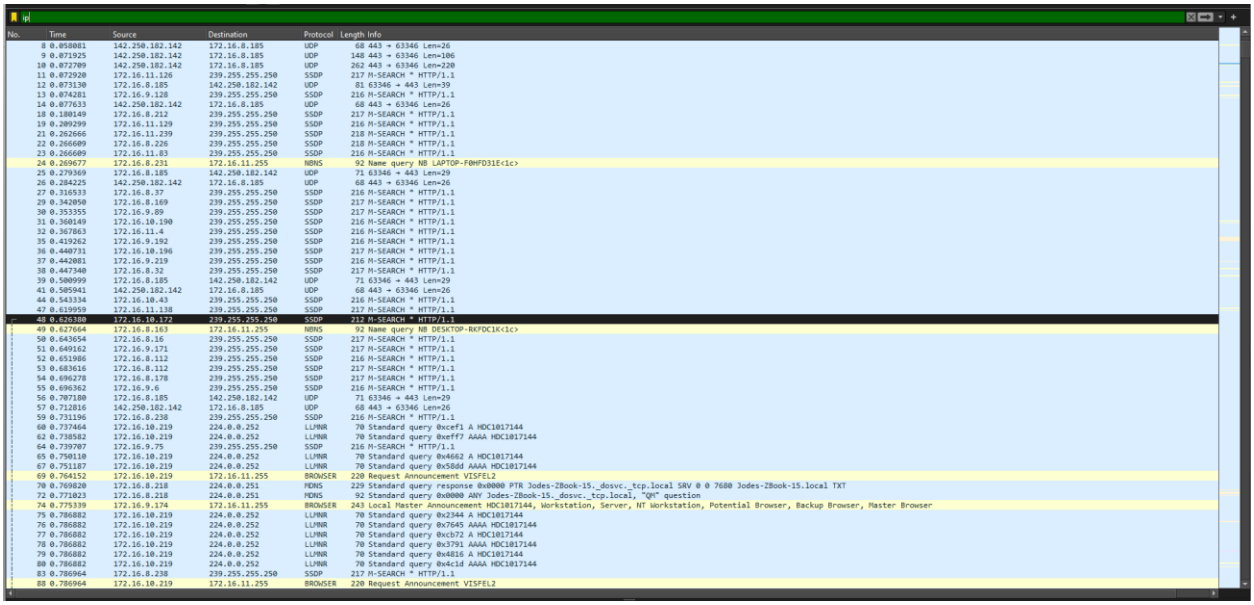


Inspecting the packets



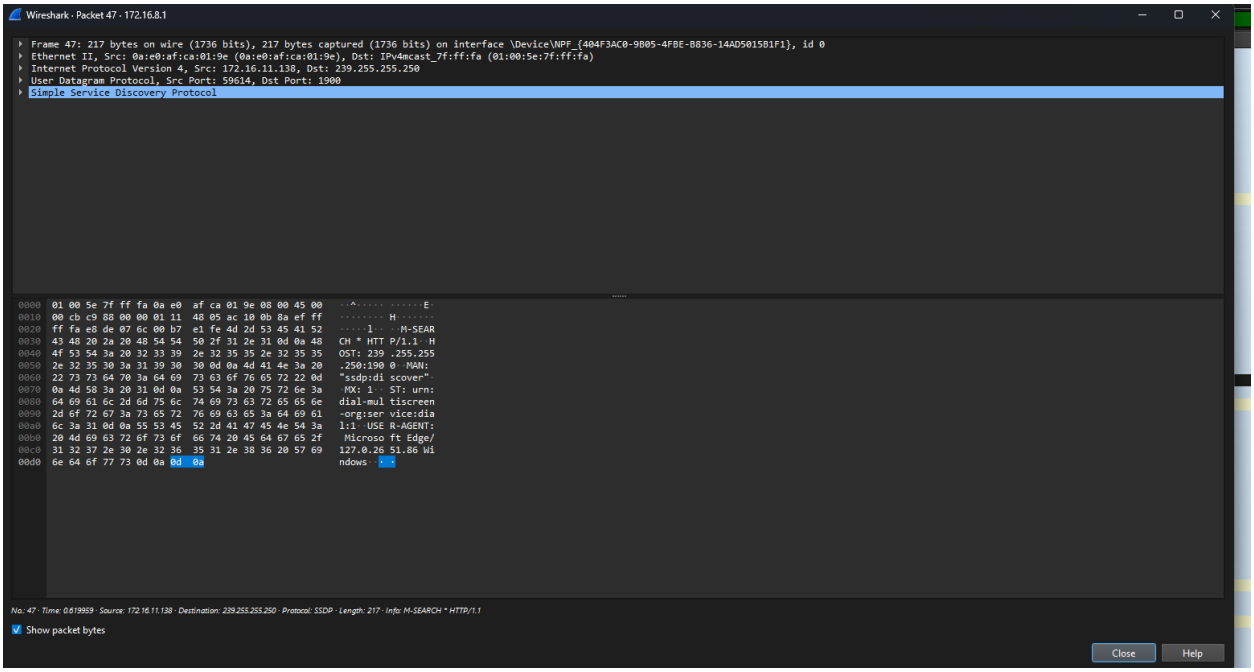
Flow Graph output

Output:ip



No.	Time	Source	Destination	Protocol	Length	Info
8	0.050081	142.250.182.142	172.16.8.185	UDP	68	443 → 63346 Len=26
9	0.071925	142.250.182.142	172.16.8.185	UDP	148	443 → 63346 Len=106
10	0.072709	142.250.182.142	172.16.8.185	UDP	262	443 → 63346 Len=220
11	0.072920	172.16.11.126	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
12	0.073130	172.16.8.185	142.250.182.142	UDP	81	63346 → 443 Len=39
13	0.074281	172.16.9.128	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
14	0.077633	142.250.182.142	172.16.8.185	UDP	68	443 → 63346 Len=26
18	0.180149	172.16.8.212	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
19	0.209299	172.16.11.129	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
21	0.262666	172.16.11.239	239.255.255.250	SSDP	218	M-SEARCH * HTTP/1.1
22	0.266609	172.16.8.226	239.255.255.250	SSDP	218	M-SEARCH * HTTP/1.1
23	0.266609	172.16.11.83	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
24	0.269677	172.16.8.231	172.16.11.255	NBNS	92	Name query NB LAPTOP-F0HFD31E<1c>
25	0.279369	172.16.8.185	142.250.182.142	UDP	71	63346 → 443 Len=29
26	0.284225	142.250.182.142	172.16.8.185	UDP	68	443 → 63346 Len=26
27	0.316533	172.16.8.37	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
29	0.342050	172.16.8.169	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
30	0.353355	172.16.9.89	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
31	0.360149	172.16.10.190	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
32	0.367863	172.16.11.4	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
35	0.419262	172.16.9.192	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
36	0.440731	172.16.10.196	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
37	0.442081	172.16.9.219	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
38	0.447340	172.16.8.32	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
39	0.500999	172.16.8.185	142.250.182.142	UDP	71	63346 → 443 Len=29
41	0.505941	142.250.182.142	172.16.8.185	UDP	68	443 → 63346 Len=26
44	0.543334	172.16.10.43	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
47	0.619959	172.16.11.138	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
48	0.626380	172.16.10.172	239.255.255.250	SSDP	212	M-SEARCH * HTTP/1.1
49	0.627664	172.16.8.163	172.16.11.255	NBNS	92	Name query NB DESKTOP-RKFDC1K<1c>
50	0.643654	172.16.8.16	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
51	0.649162	172.16.9.171	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
52	0.651986	172.16.8.112	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
53	0.683616	172.16.8.112	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
54	0.696278	172.16.8.178	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
55	0.696362	172.16.9.6	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
56	0.707180	172.16.8.185	142.250.182.142	UDP	71	63346 → 443 Len=29
57	0.712816	142.250.182.142	172.16.8.185	UDP	68	443 → 63346 Len=26
59	0.731196	172.16.8.238	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
60	0.737464	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0xcef1 A HDC1017144
62	0.738582	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0xeff7 AAAA HDC1017144
64	0.739707	172.16.9.75	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
65	0.750110	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x4662 A HDC1017144
67	0.751187	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x58dd AAAA HDC1017144
69	0.764152	172.16.10.219	172.16.11.255	BROWSER	220	Request Announcement VISFEL2
70	0.769820	172.16.8.218	224.0.0.251	MDNS	229	Standard query response 0x0000 PTR Jodes-ZBook-15._dosvc._tcp.local SRV 0 0 7680 Jodes-ZBook-15.local TXT
72	0.771023	172.16.8.218	224.0.0.251	MDNS	92	Standard query 0x0000 ANY Jodes-ZBook-15._dosvc._tcp.local, "QM" question
74	0.775339	172.16.9.174	172.16.11.255	BROWSER	243	Local Master Announcement HDC1017144, Workstation, Server, NT Workstation, Potential Browser, Backup Browser, Master Browser
75	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x2344 A HDC1017144
76	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x7645 AAAA HDC1017144
77	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0xcb72 A HDC1017144
78	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x3791 AAAA HDC1017144
79	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x4816 A HDC1017144
80	0.786882	172.16.10.219	224.0.0.252	LLMNR	70	Standard query 0x4c1d AAAA HDC1017144
83	0.786964	172.16.8.238	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
88	0.786964	172.16.10.219	172.16.11.255	BROWSER	220	Request Announcement VISFEL2

Inspecting the packets



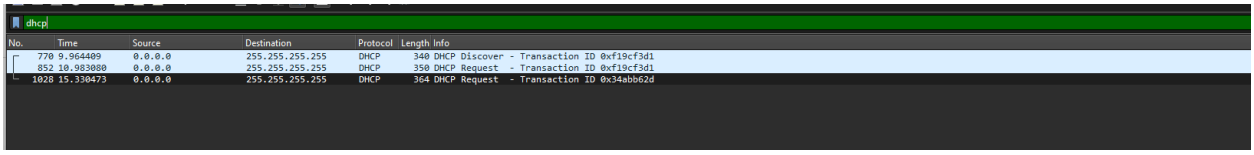
Flow chart output

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

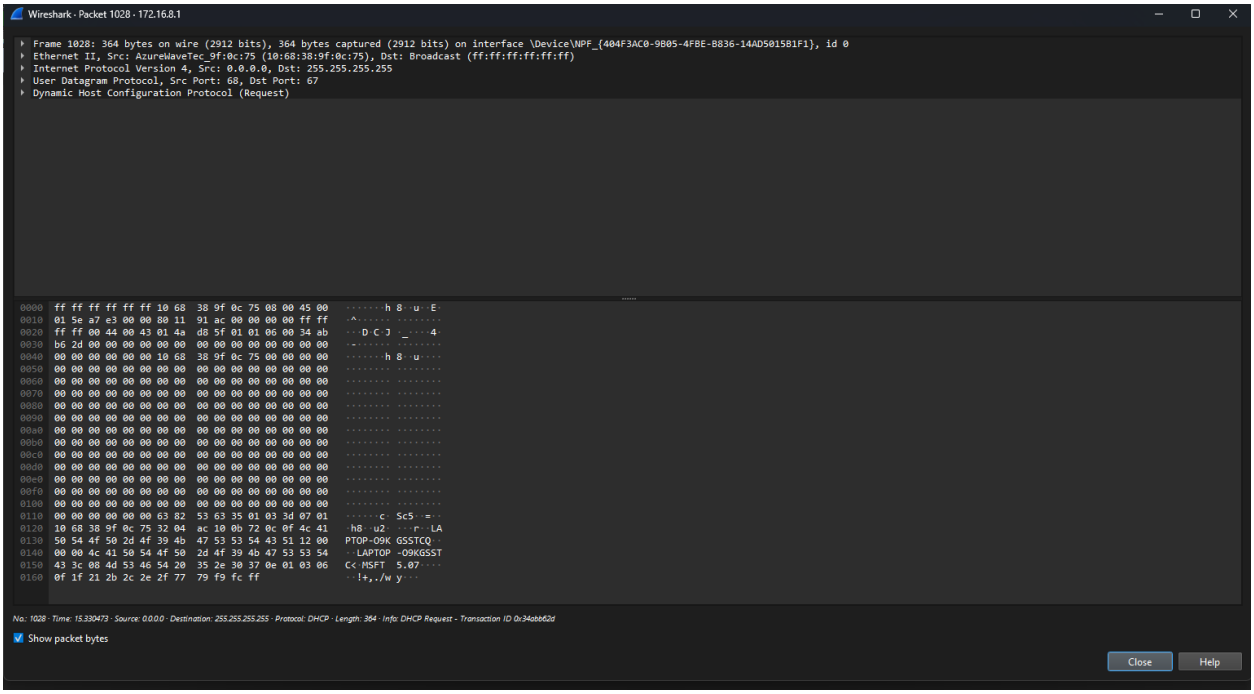
Procedure

- 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 the packets



Result:

Network traffic on TCP / UDP / IP / HTTP / ARP /DHCP /ICMP /DNS using Wireshark Tool is analyzed.