

13/8/24

Ex No: 4a STUDY OF WIRESHARK TOOL FOR PACKET SNIFFING 231901001

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:

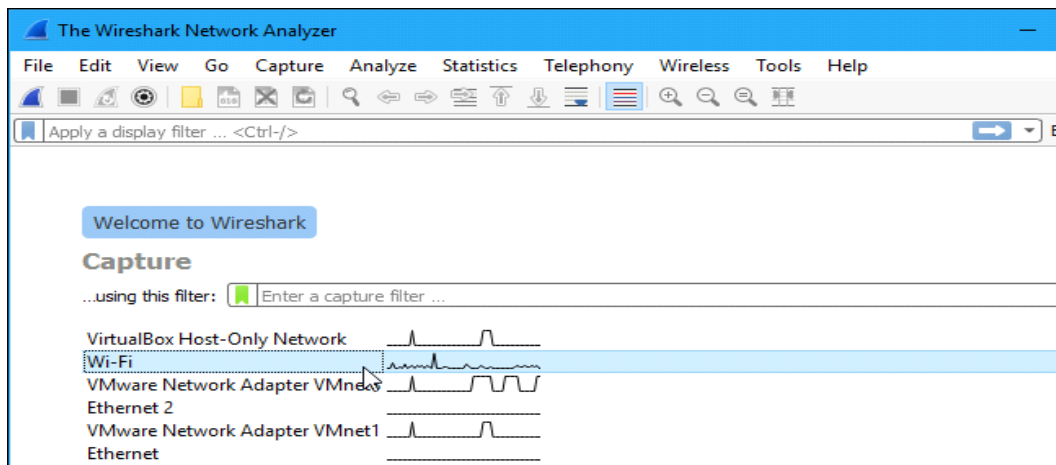
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

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.

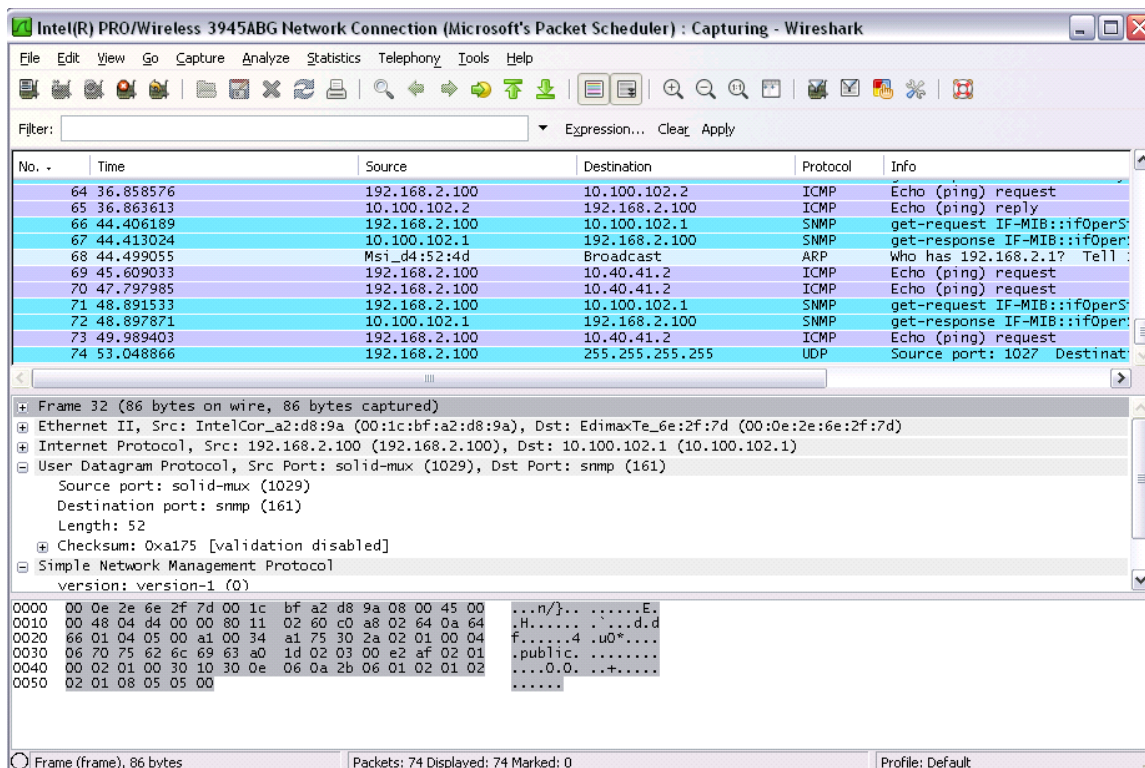
Capturing Packets

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.

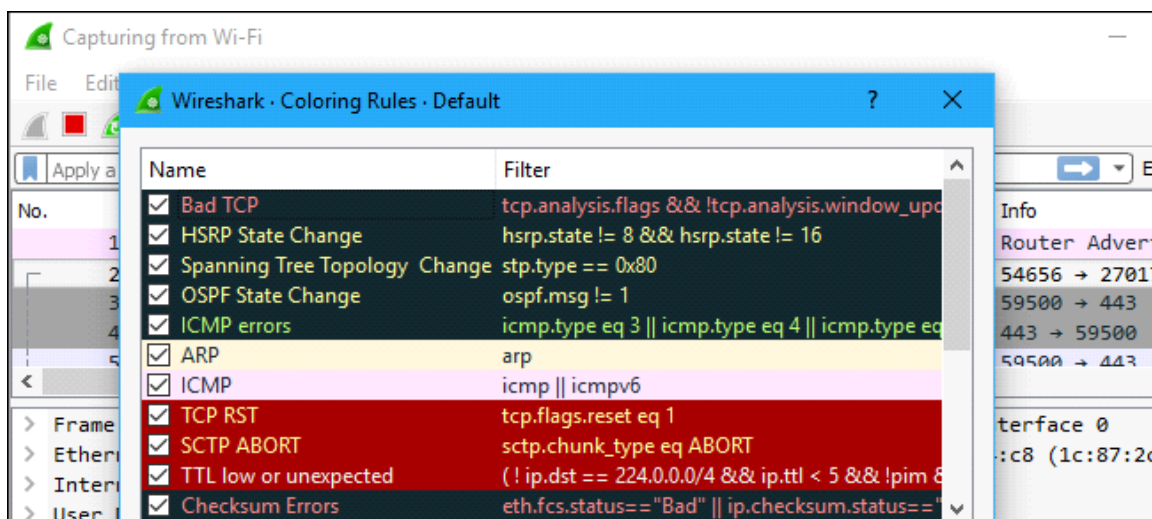
The “Packet Bytes” Pane

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.

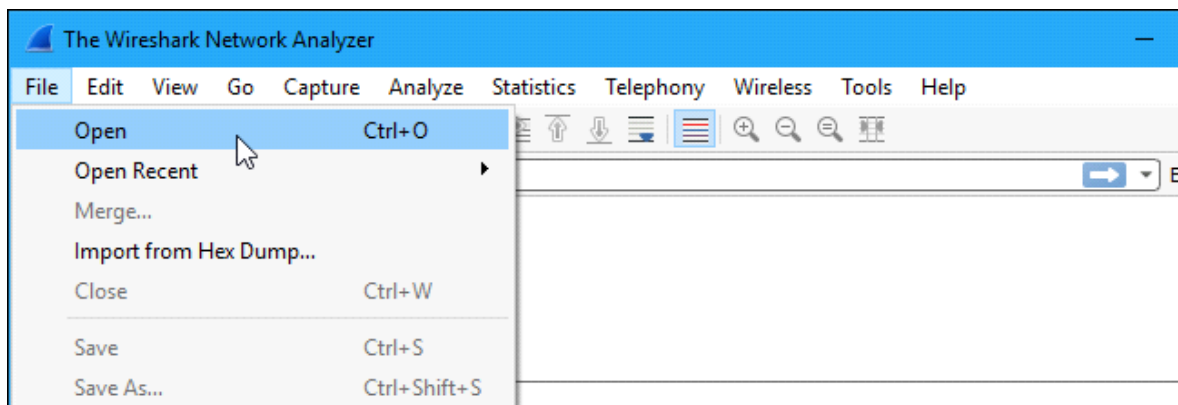
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



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.

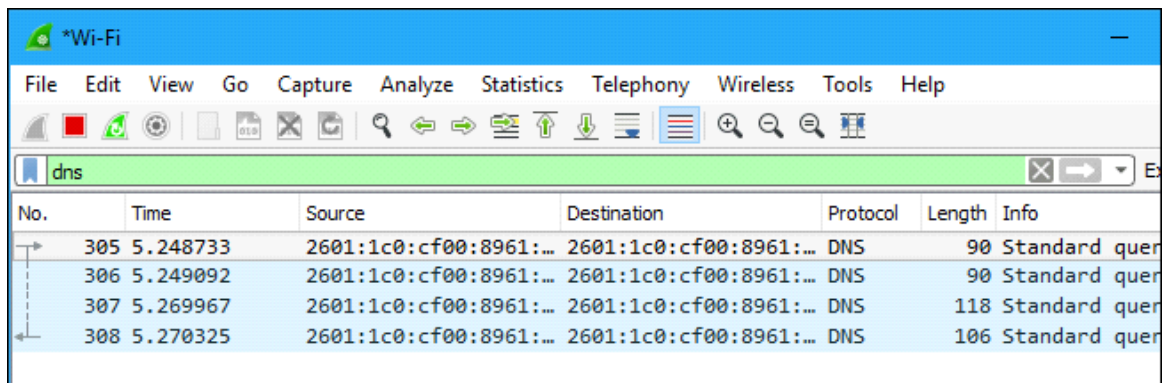
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Filtering Packets

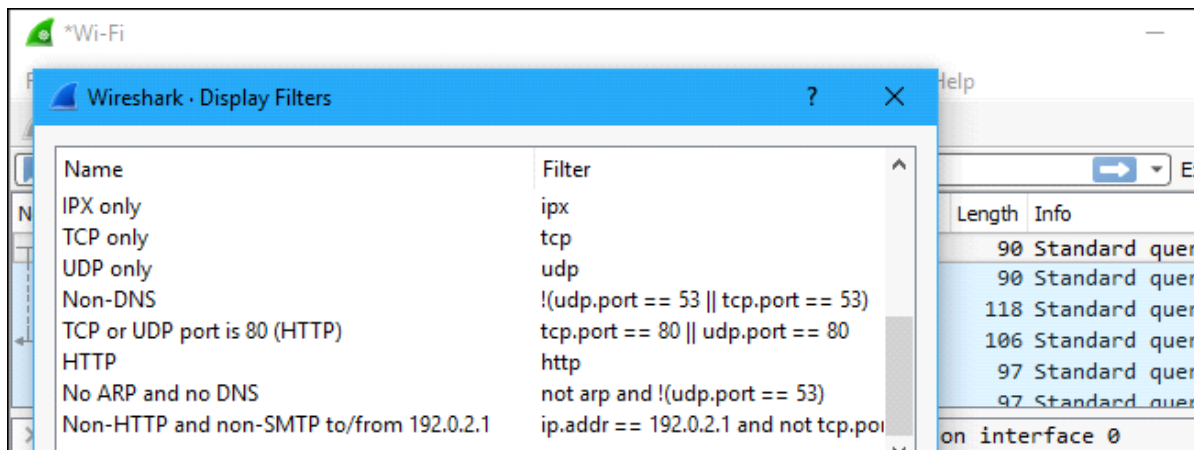
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters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



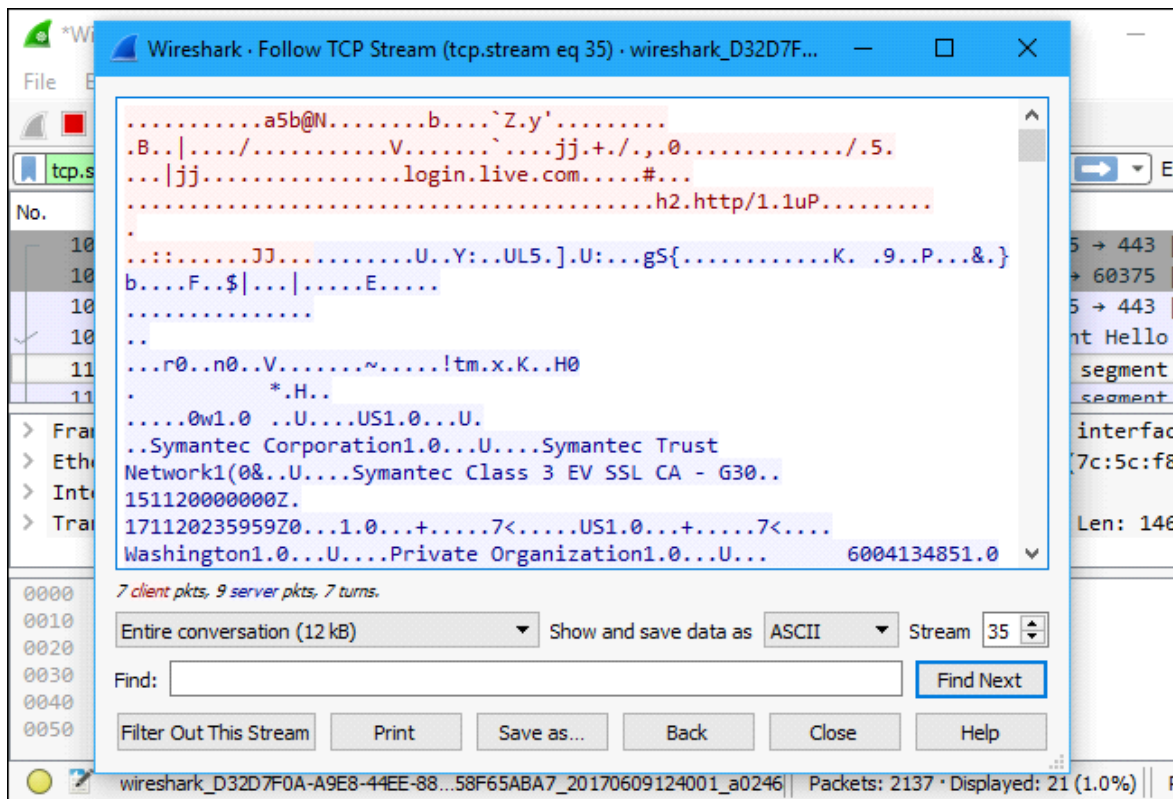
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

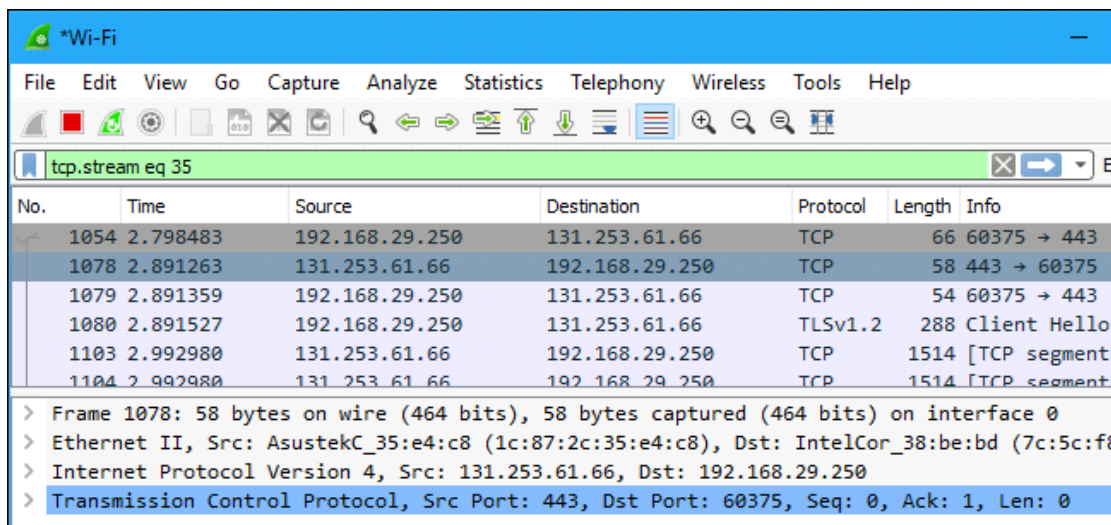


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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.

Wireshark interface showing a packet capture filter `tcp.stream eq 35`. The packet list shows several TCP packets. Packet 1054 is selected, showing details for Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes are displayed in hexadecimal and ASCII.

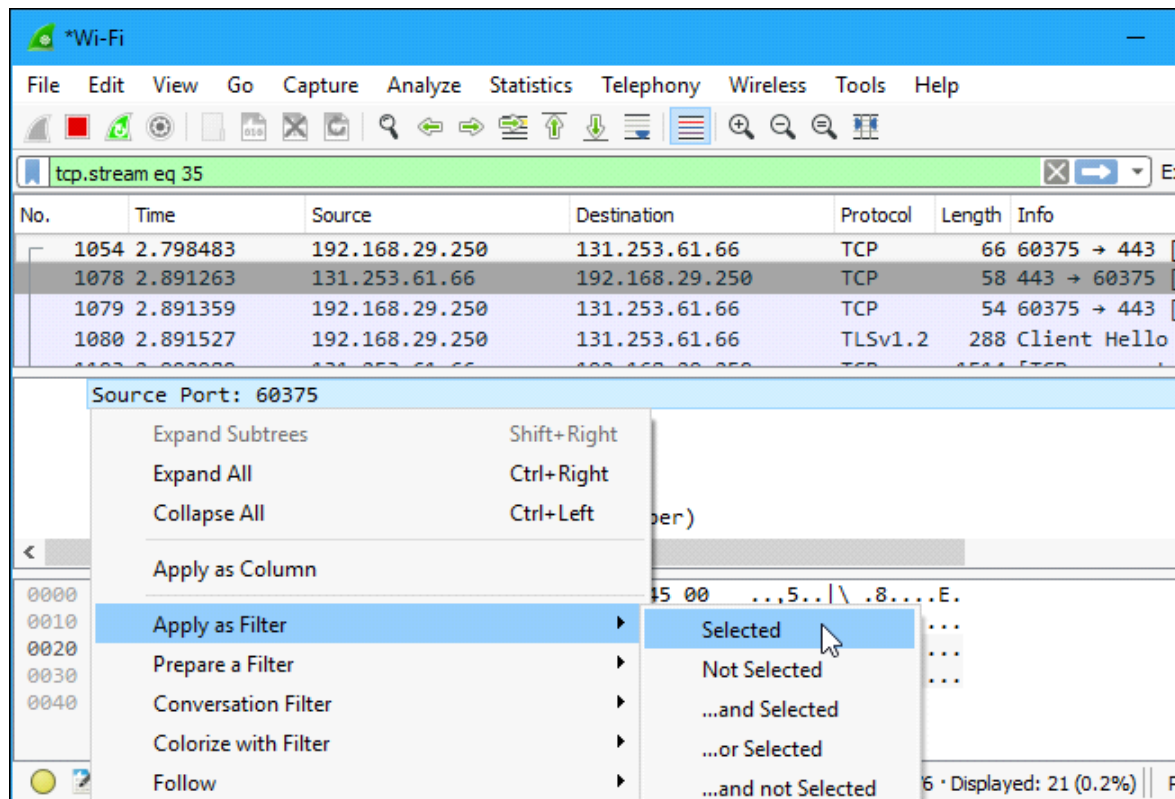
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443 [
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375 [
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443 [
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client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

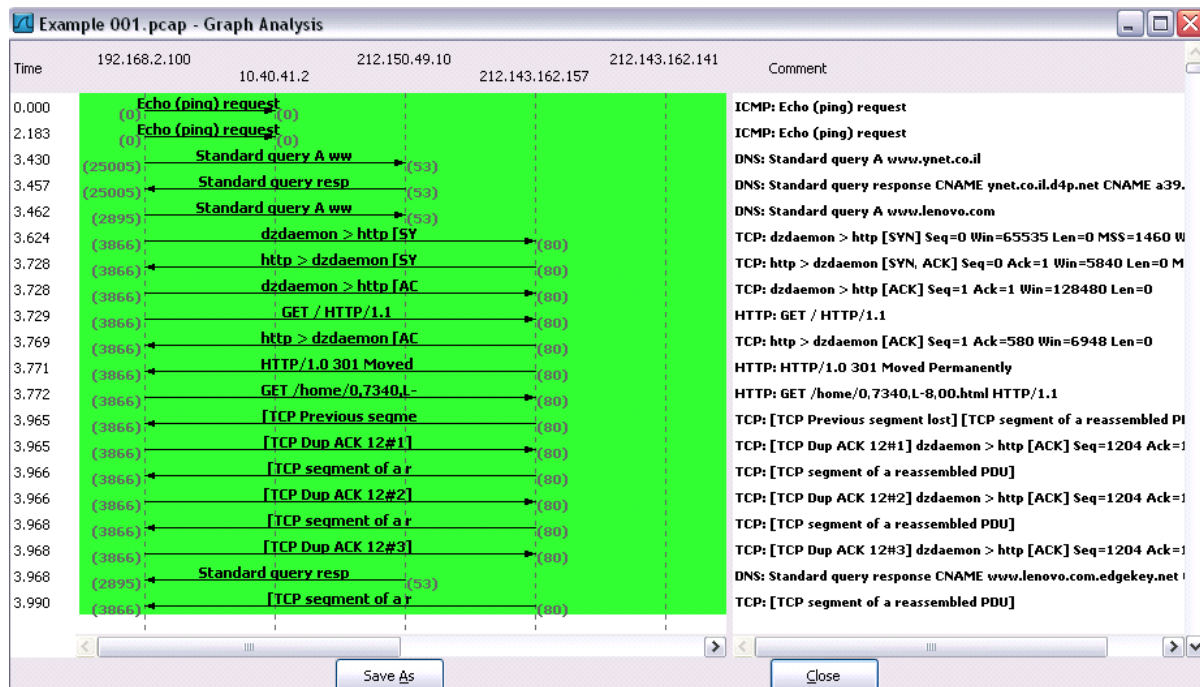
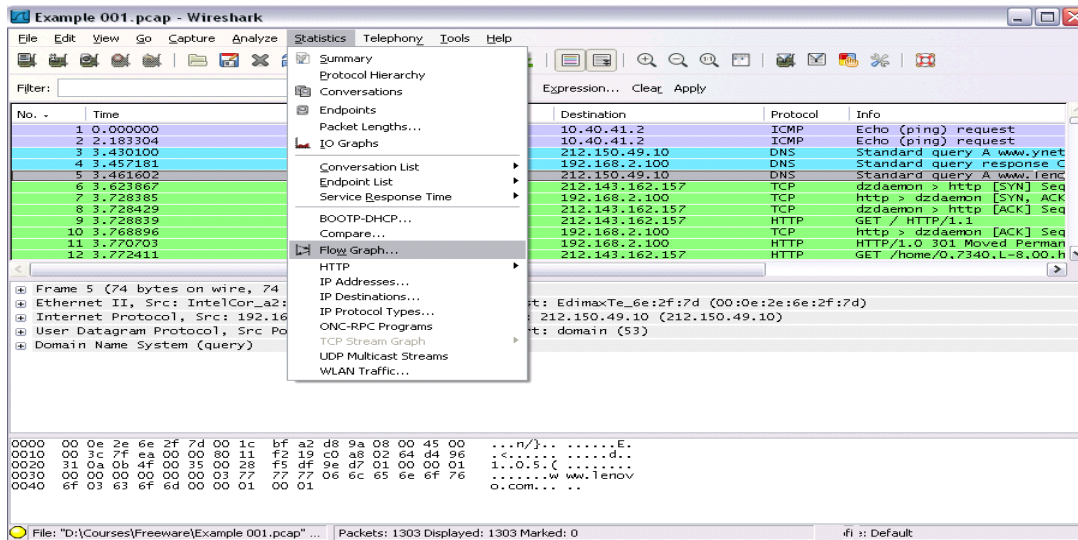
Encapsulation type (frame.encap_type) | Packets: 8136 · Displayed: 21 (0.3%)

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.



231901001

Ex No: 14 b

PACKET SNIFFING USING WIRESHARK

AIM:

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

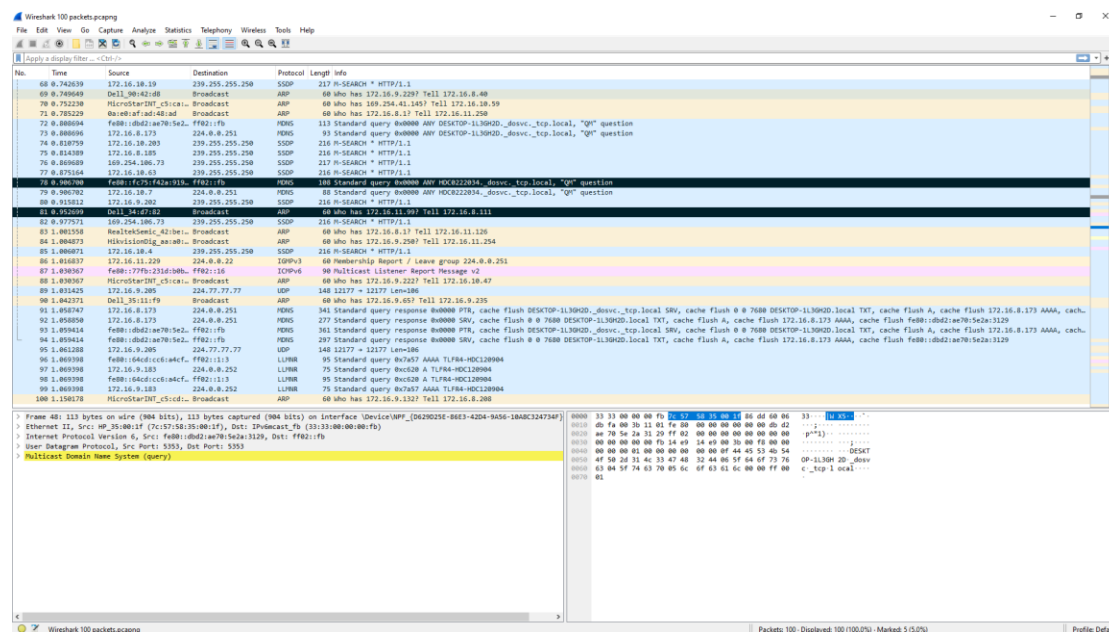
Exercises

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

Procedure



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

Output

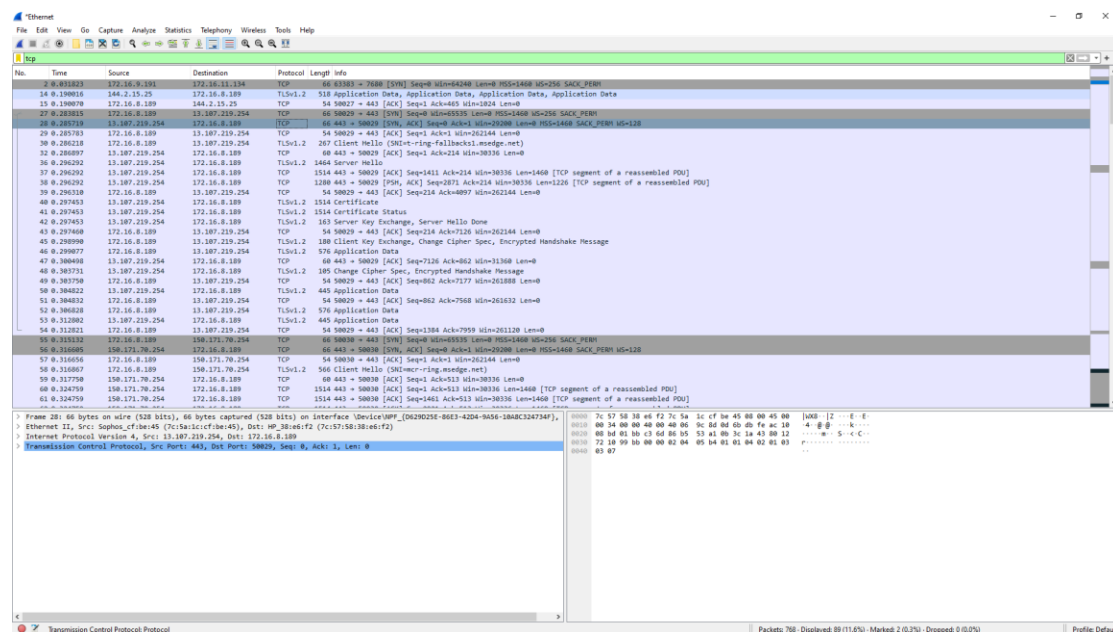


2. Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph.

Procedure

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

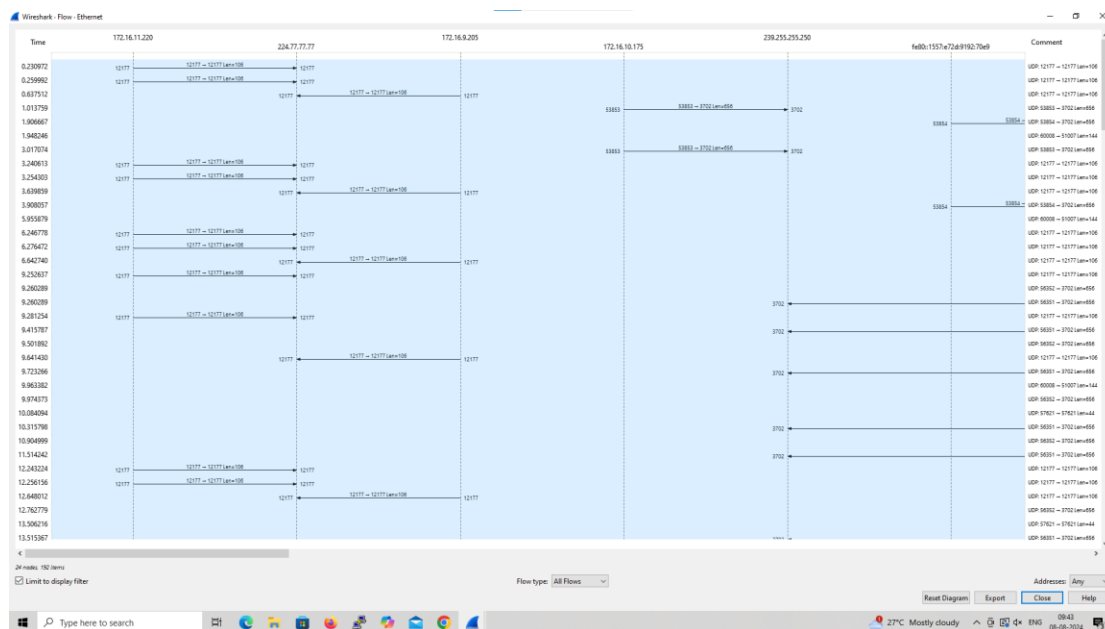
Output:



Wireshark interface showing a packet capture on the 'ethernet' interface. The packet list on the left shows multiple UDP packets from 172.16.11.220 to 224.77.77.77. The packet details pane on the right shows the structure of a User Datagram Protocol (UDP) packet, including Ethernet II, Internet Protocol Version 4, and User Datagram Protocol fields. The packet bytes pane at the bottom shows the raw hex and ASCII data of the selected packet.

Wireshark - Packet 1196: Ethernet. This pane provides a detailed view of the selected packet (Frame 1902). It shows the packet structure: Ethernet II (Broadcast), Internet Protocol Version 4 (Destination: 224.77.77.77), User Datagram Protocol (Destination Port: 51007), and Data (144 bytes). The packet bytes pane displays the raw hex and ASCII data of the packet, including the Ethernet header, IP header, UDP header, and the application data payload.

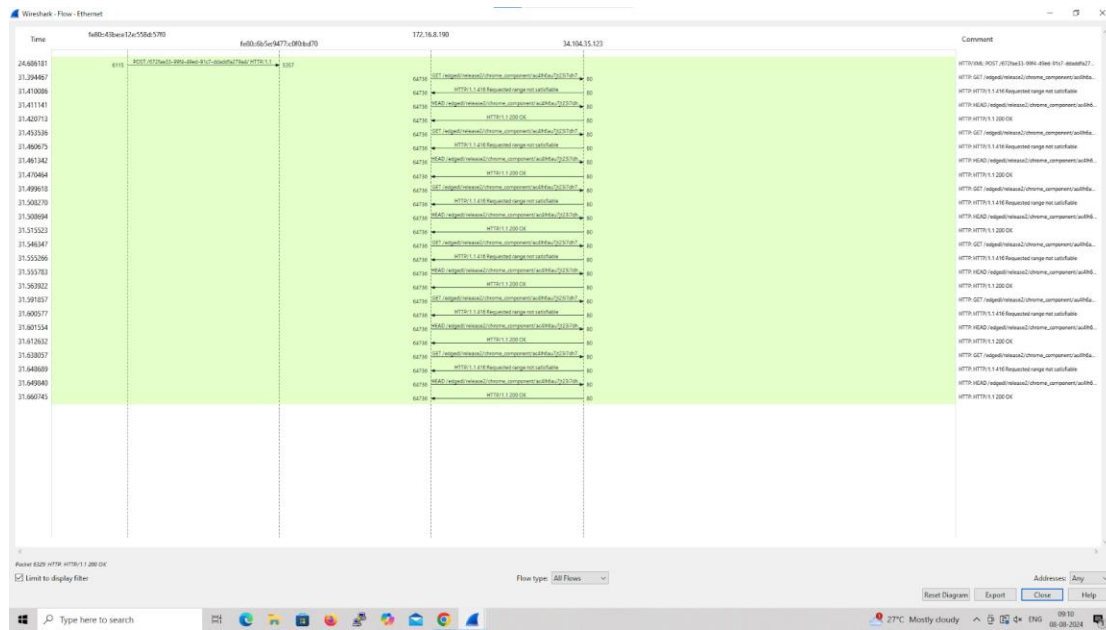
Flow Graph output



- Select Local Area Connection in Wireshark.
- Go to capture □ option
- Select stop capture automatically after 100 packets.


Output



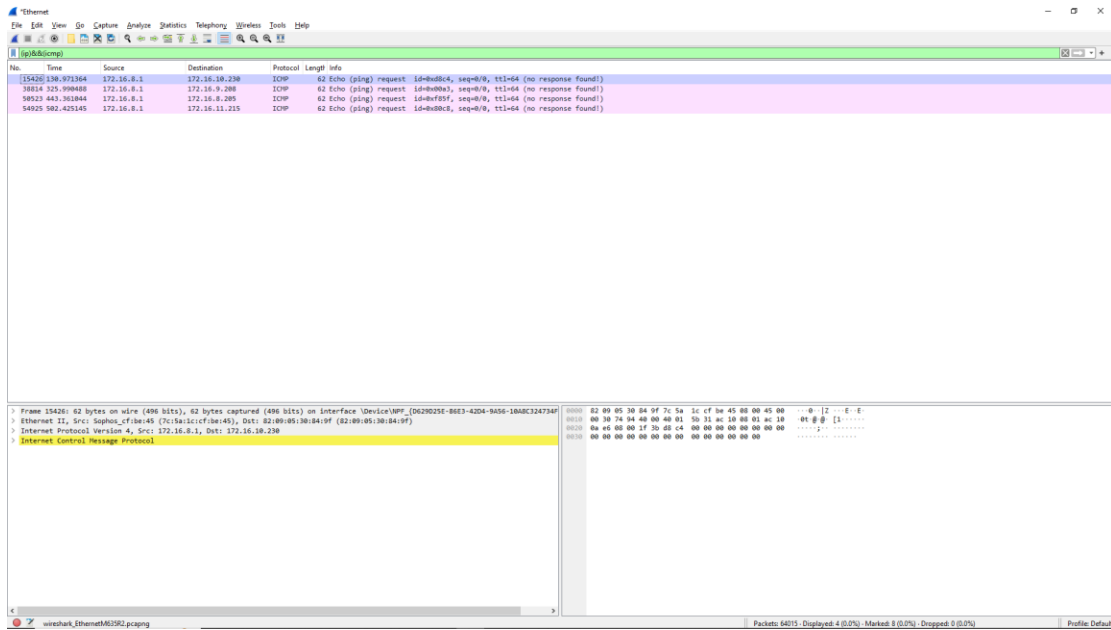


6.Create a Filter to display only IP/ICMP packets and inspect the packets.

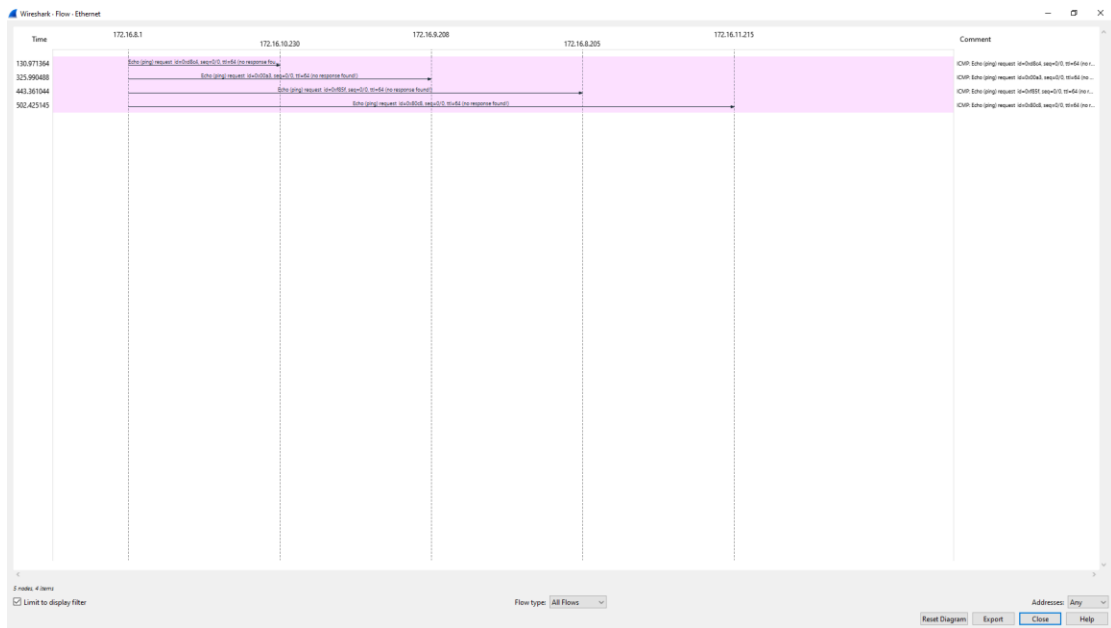
Procedure

- 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

The image displays a Wireshark packet capture of a DHCP transaction. The packet list shows a series of DHCP messages: Discover, Offer, Request, and ACK. The packet details pane shows the structure of a DHCP packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Dynamic Host Configuration Protocol. The packet bytes pane shows the raw hex and ASCII data.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
2567	20.858876	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 8bd61899b0
2572	20.869740	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd61899b1
2782	22.498424	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd61899b1
3430	30.883870	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bd637033d
3431	30.884556	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bd637033d
3907	35.488891	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8ba2746da9f
4554	40.763564	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8ba2746da9f
4877	41.688287	0.0.0.0	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 8ba2746da9f
8888	61.618106	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8ba2746da9f
8748	69.139428	0.0.0.0	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 8bd3b4144f
8749	69.139431	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bd3b4144f
8950	80.897575	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8ba2746da2
11081	88.362479	0.0.0.0	255.255.255.255	DHCP	358	DHCP Request - Transaction ID 8bd6c79989
11082	88.362512	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bd6c79989
12799	118.666969	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8b734ae87d
12800	118.666969	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8b734ae87d
14330	120.752894	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bfc758826
14331	120.752893	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bfc758826
14363	120.365412	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd955fcb8
14376	120.365150	0.0.0.0	255.255.255.255	DHCP	342	DHCP Decline - Transaction ID 8b0
15268	120.366298	0.0.0.0	255.255.255.255	DHCP	348	DHCP Request - Transaction ID 8b38ff72a1
15384	120.559820	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8b73978a7b
15385	120.559831	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8b73978a7b
15427	130.875437	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 8bd955fcb8
15572	133.111376	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd955fcb8
16888	144.484848	172.16.8.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 8bd6c79989
16889	144.484848	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 8bd6c79989
16814	145.699551	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd61899b2
17807	149.193889	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 8bd955fcb8
17639	155.444831	0.0.0.0	255.255.255.255	DHCP	354	DHCP Request - Transaction ID 8bd6388ffe
18054	166.312441	0.0.0.0	255.255.255.255	DHCP	354	DHCP Request - Transaction ID 8bd6c79989
28621	176.401517	0.0.0.0	255.255.255.255	DHCP	324	DHCP Discover - Transaction ID 8bd617277b
28736	171.427682	0.0.0.0	255.255.255.255	DHCP	330	DHCP Request - Transaction ID 8bd617277b

Packet Details:

- Frame 8888: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{062025E-86E3-4204-9456-1848C324734F}
- Ethernet II, Src: Sophos_cf8e45 (7c15a1c1cf8e45), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Internet Protocol Version 4, Src: 172.16.8.1, Dst: 255.255.255.255
- User Datagram Protocol, Src Port: 67, Dst Port: 68
- Dynamic Host Configuration Protocol (ACK)

Packet Bytes:

```

0000  ff ff ff ff ff ff 7c 15 1c cf be 45 00 00 45 18  ....[2]...E:
0010  01 40 00 00 00 00 11 25 84 ac 18 00 01 ff ff  ..H.....
0020  ff ff 00 43 00 44 81 34 bf 32 02 01 00 07 e4  ....C D 4 2...
0030  0f 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .........
0040  00 00 00 00 00 28 c0 c4 a0 63 13 00 00 00 00  ....[C]...B...
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  .........
0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .........
0110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .........
0120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .........
0130  18 00 01 33 04 00 0c 43 01 84 ff ff 00 00 03  ....g 5c5 6...
0140  84 ac 18 00 01 05 04 ac 18 00 01 ff ff 00 00  ....3.....
0150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .........

```

