Ex No: 14a STUDY OF WIRESHARK TOOL FOR PACKET SNIFFING

AIM:

To study packet sniffing concepts using Wireshark Tool.

DESCRIPTION:

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

What we can do with Wireshark:

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

Wireshark used for:

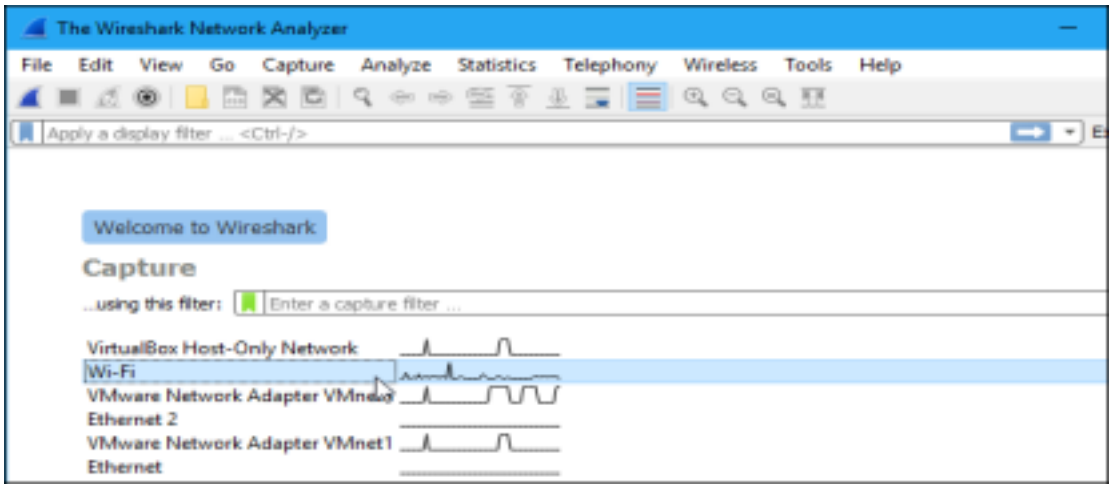
- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: learn **network protocol internals**

Getting Wireshark

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

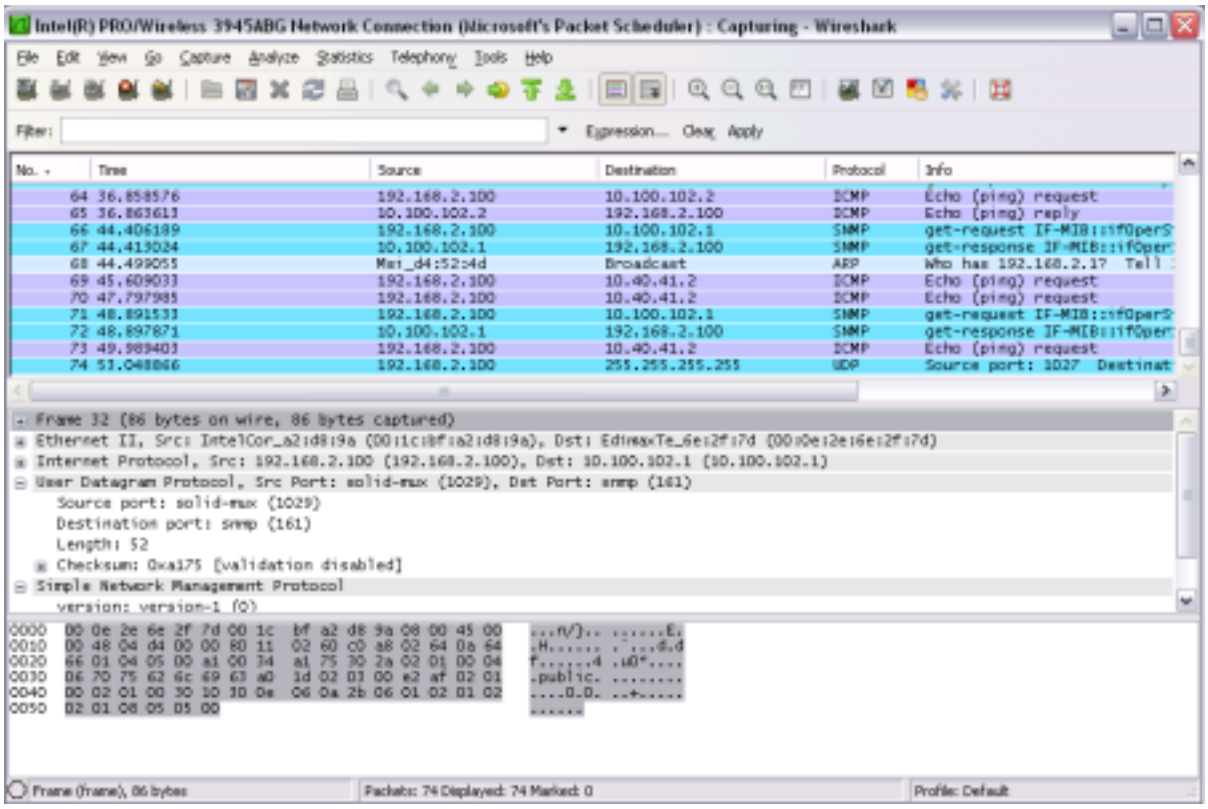
Capturing Packets

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



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

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



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

The "Packet List" Pane

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

The “Packet Details” Pane

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

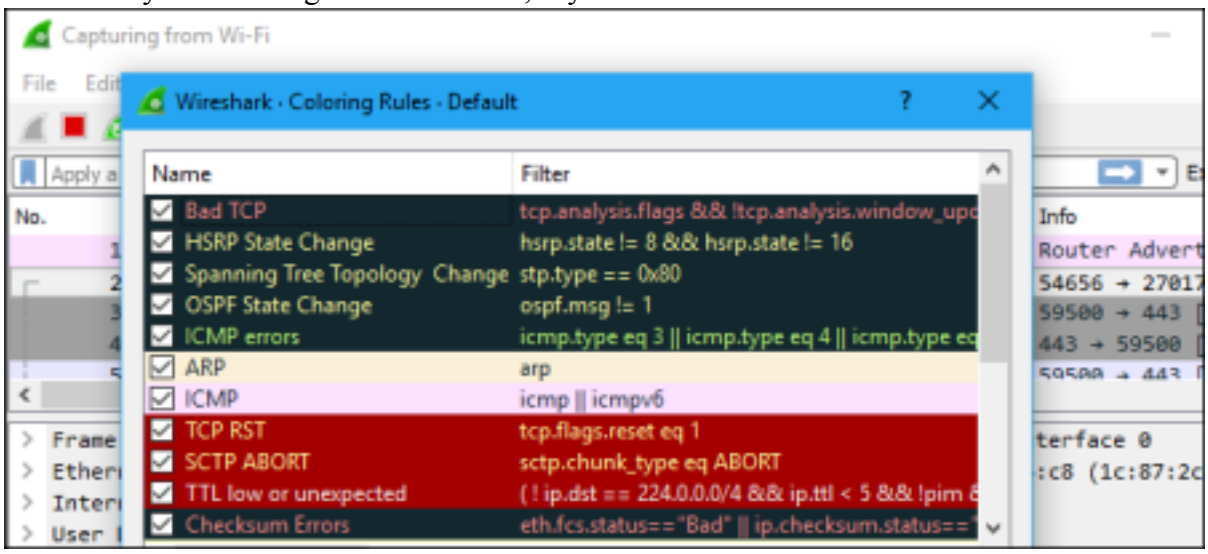
The “Packet Bytes” Pane

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

Color Coding

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

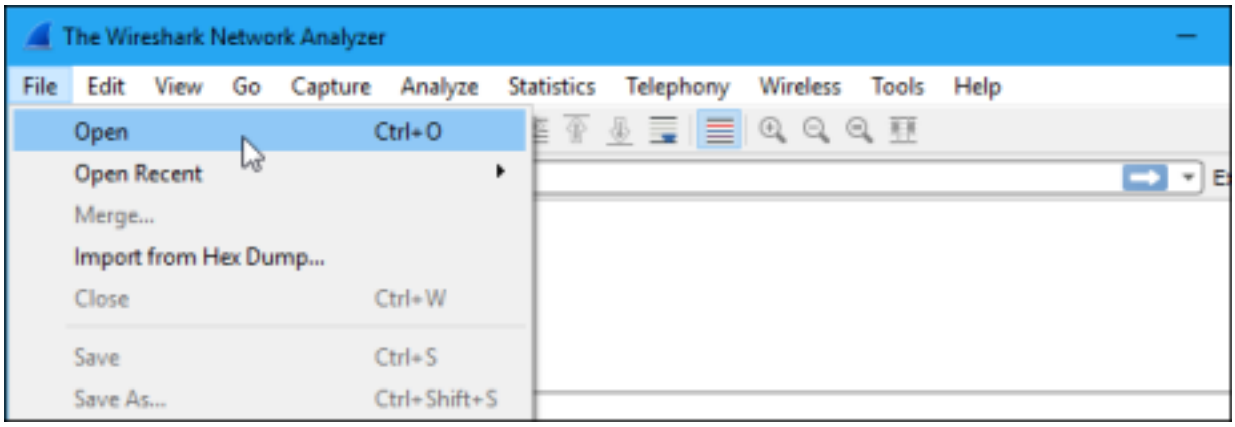
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



Sample Captures

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

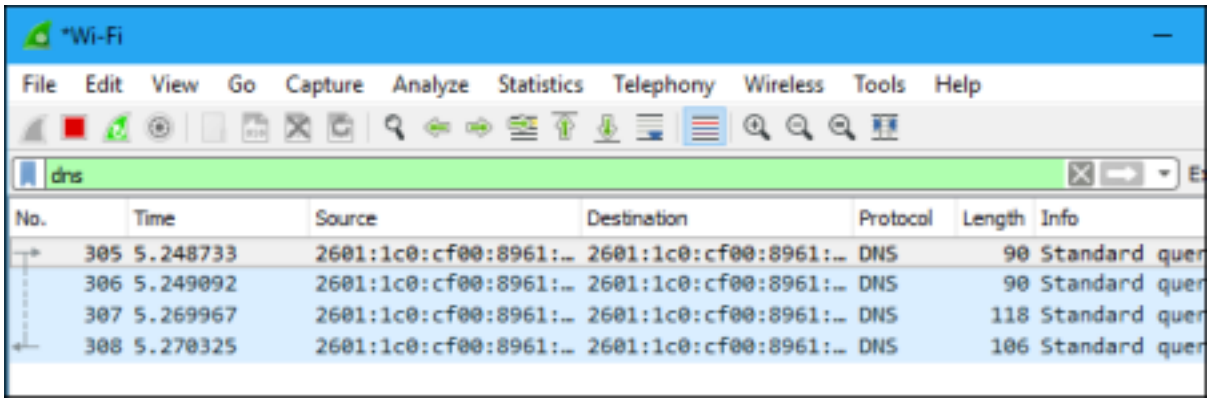
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Filtering Packets

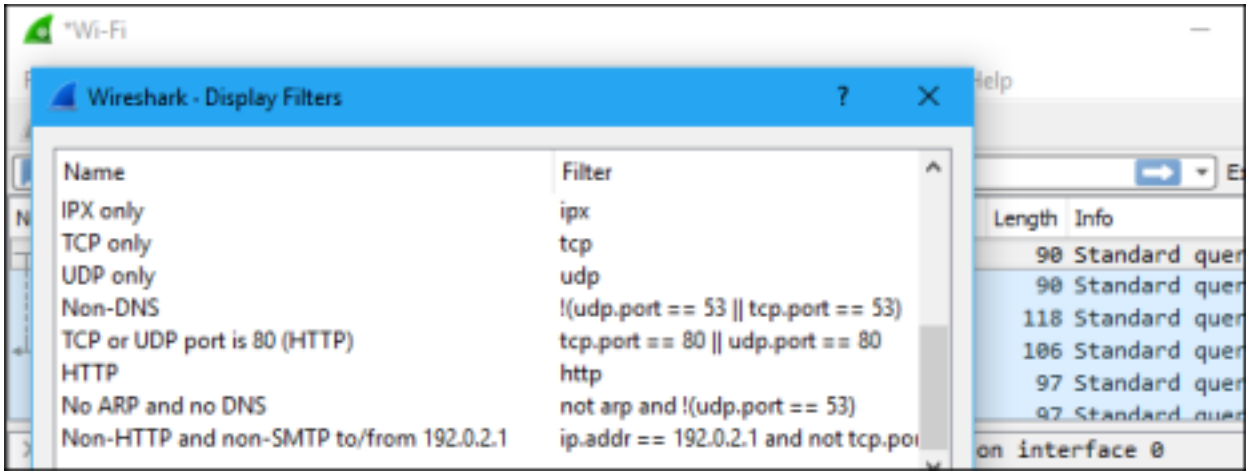
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters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

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



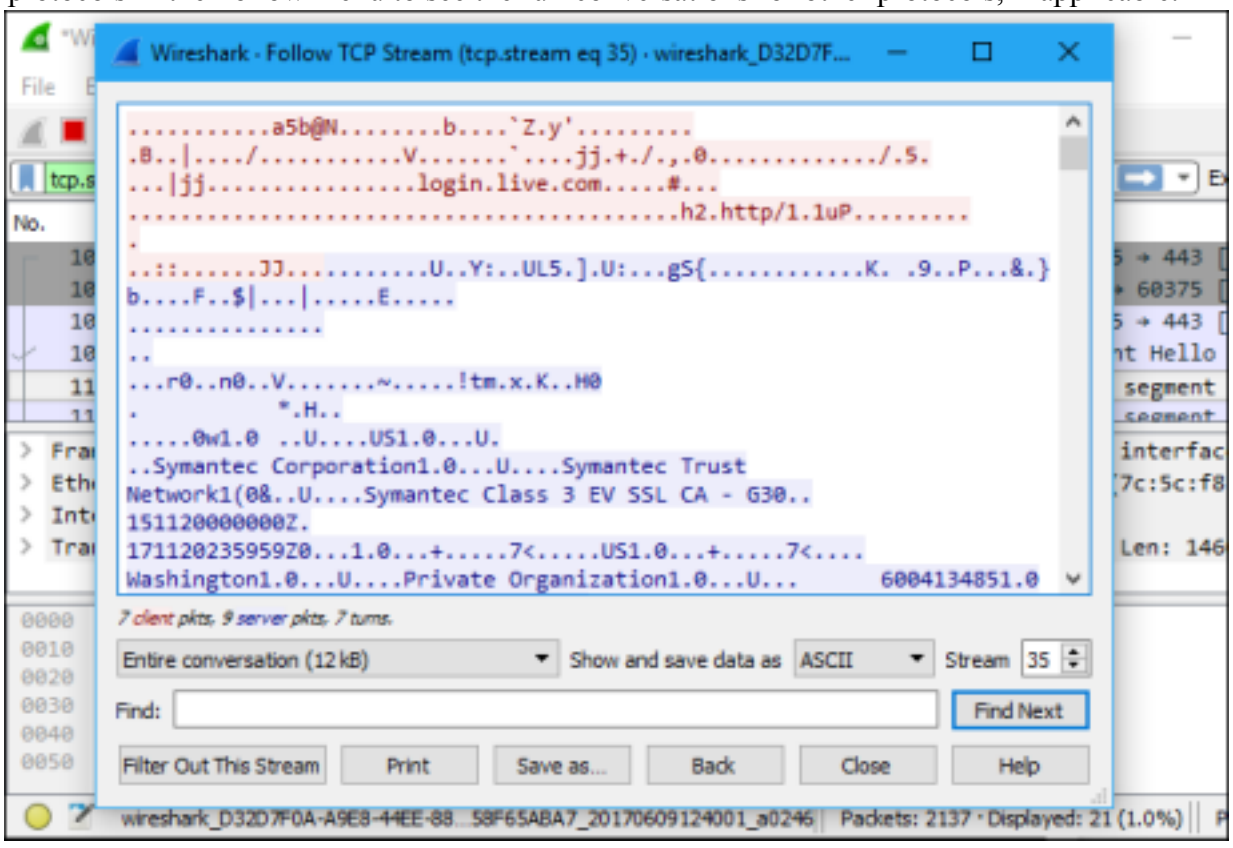
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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

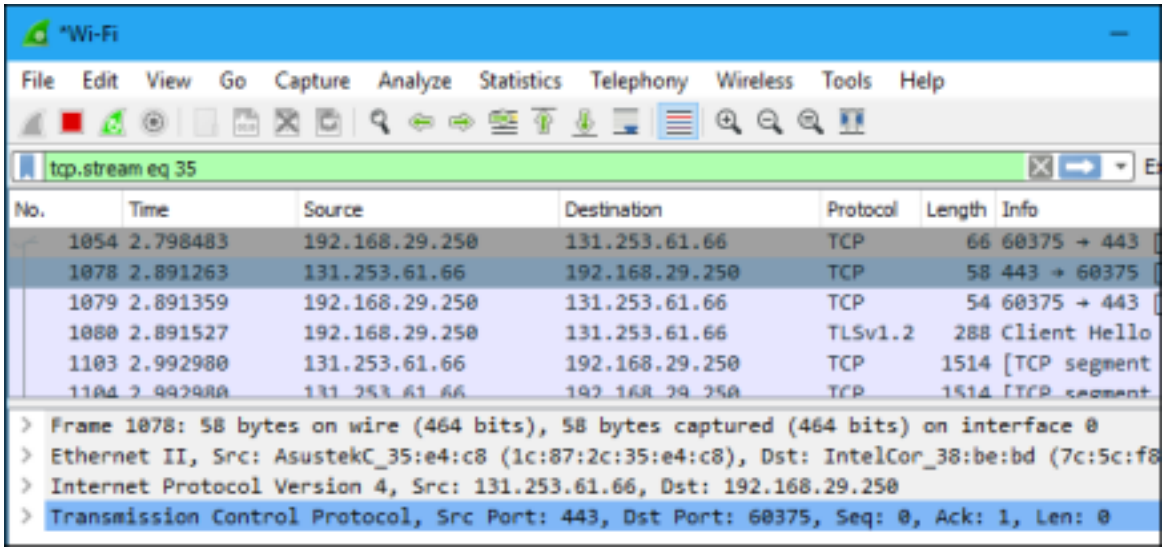


Another interesting thing you can do is right-click a packet and select Follow > TCP Stream.

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

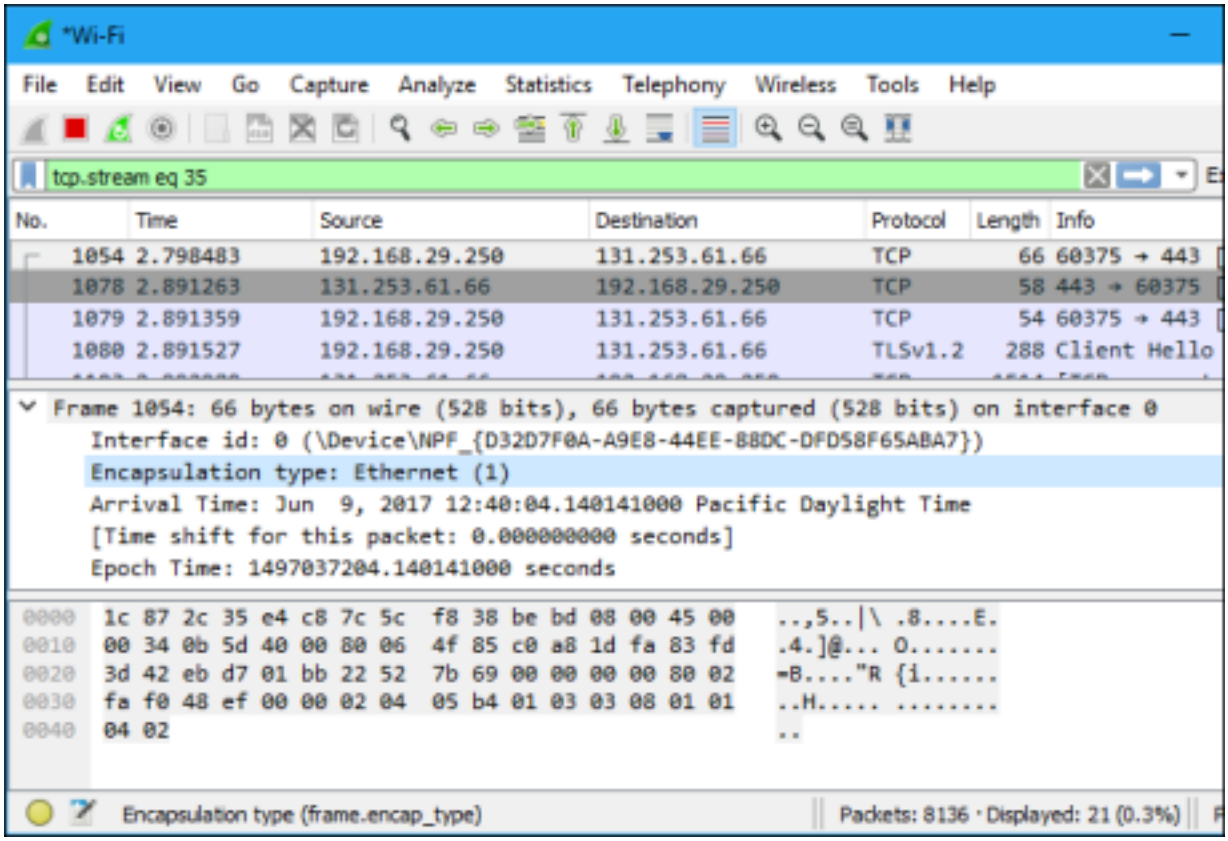


Close the window and you'll find a filter has been applied automatically. Wireshark is showing you the packets that make up the conversation.

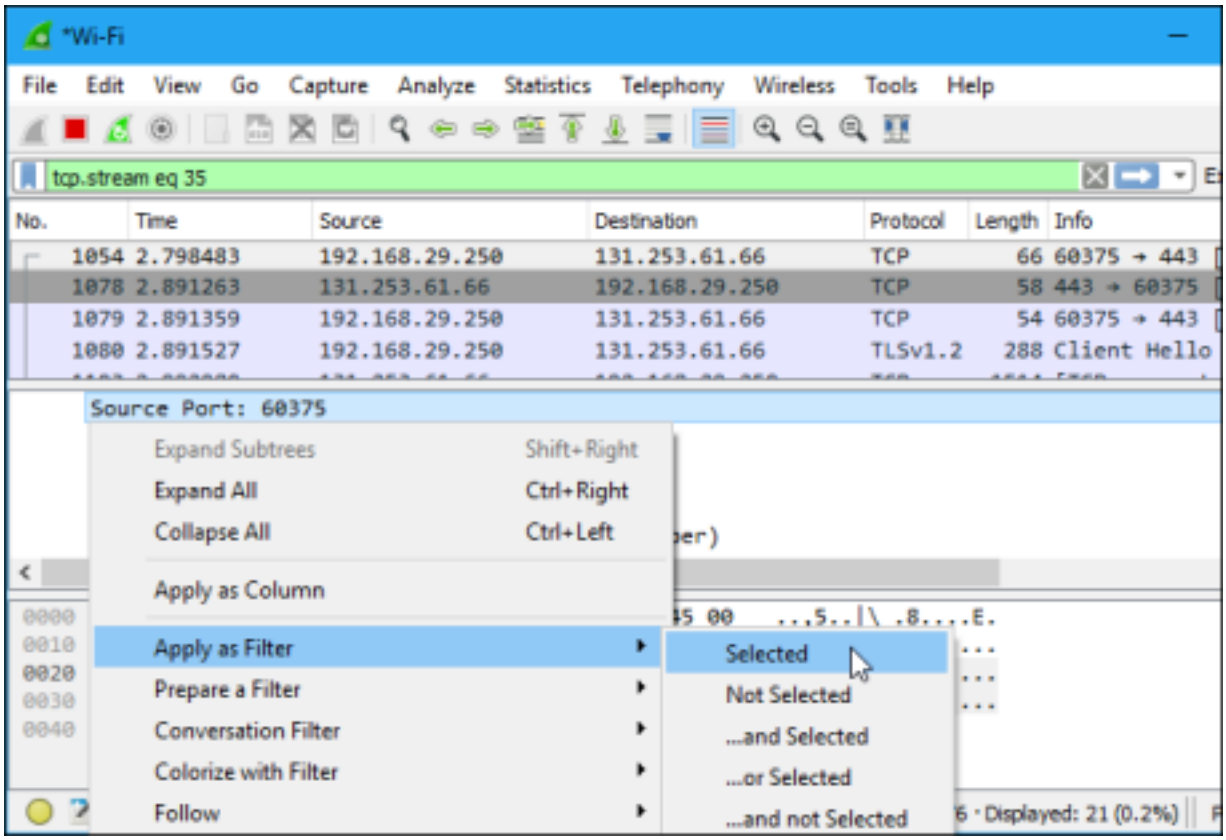


Inspecting Packets

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

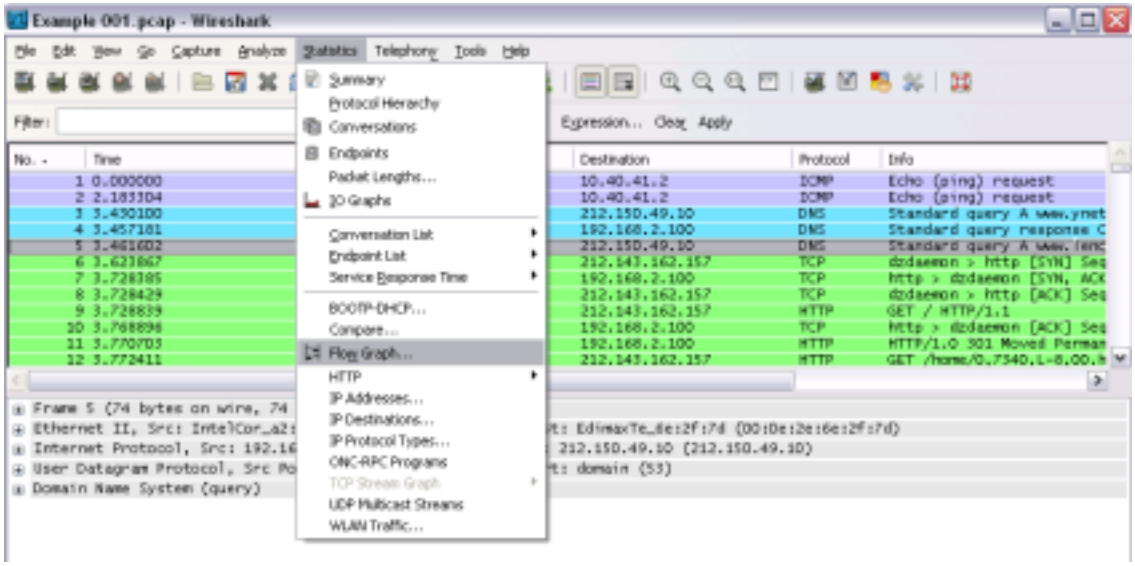


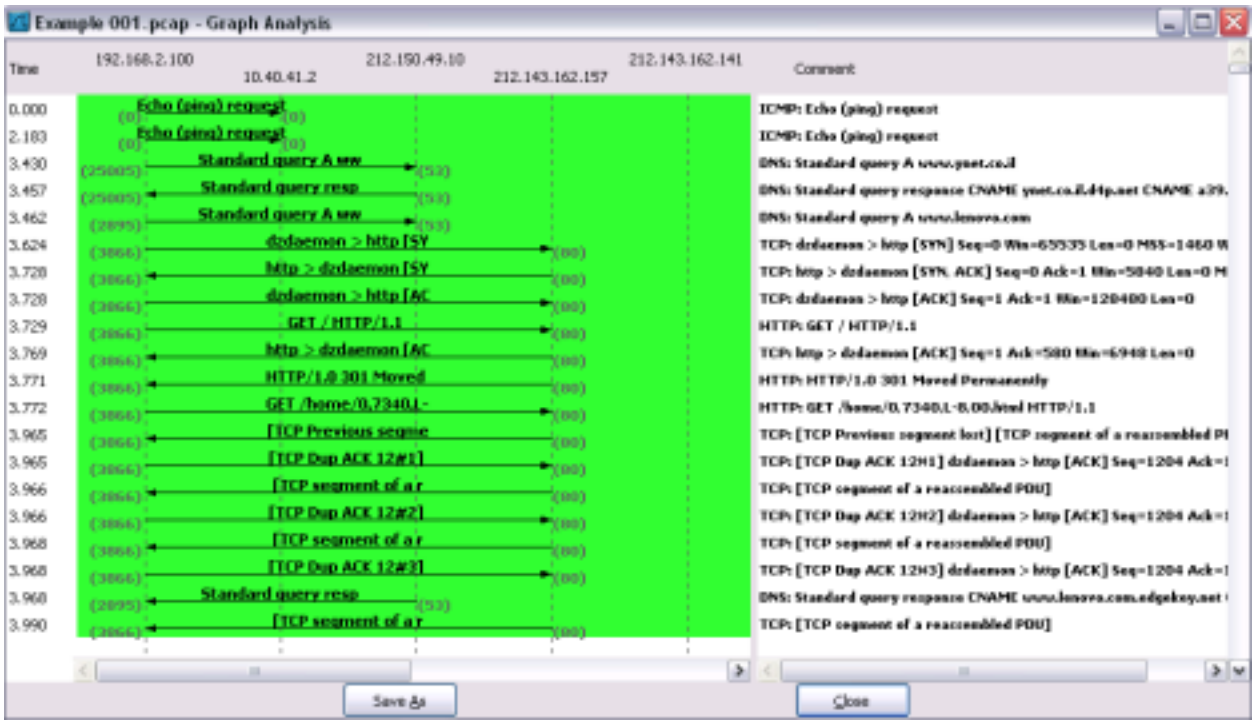
You can also create filters from here — just right-click one of the details and use the Apply as Filter submenu to create a filter based on it.



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

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





Ex No: 14 b PACKET SNIFFING USING WIRESHARK

AIM:

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

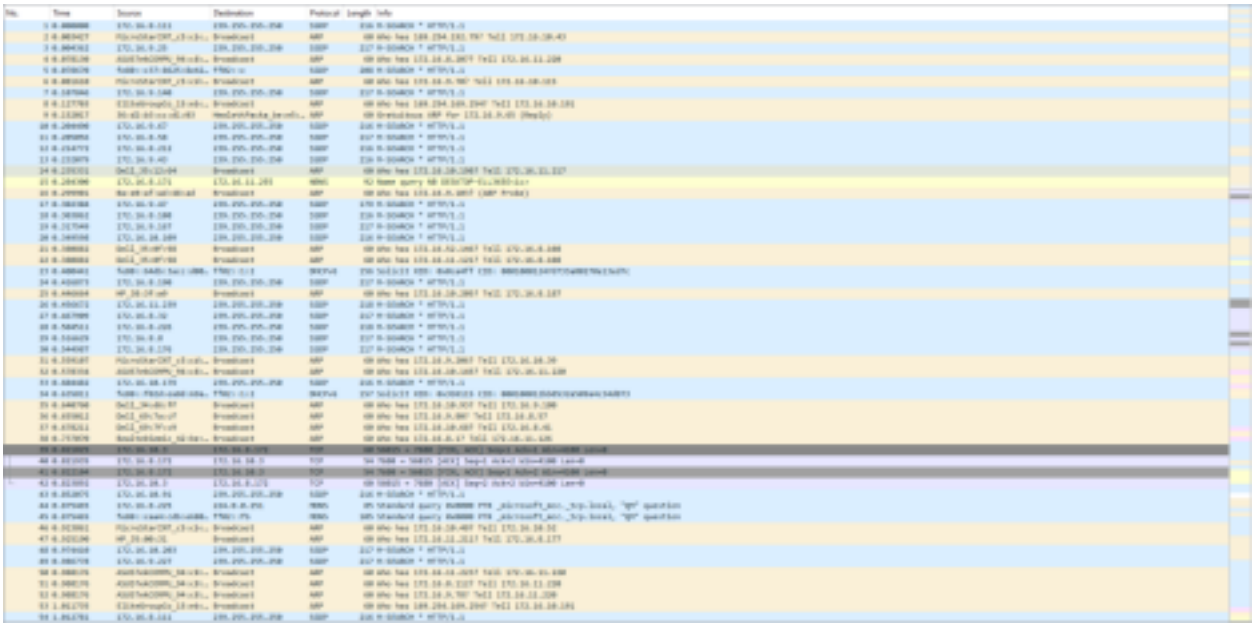
Exercises

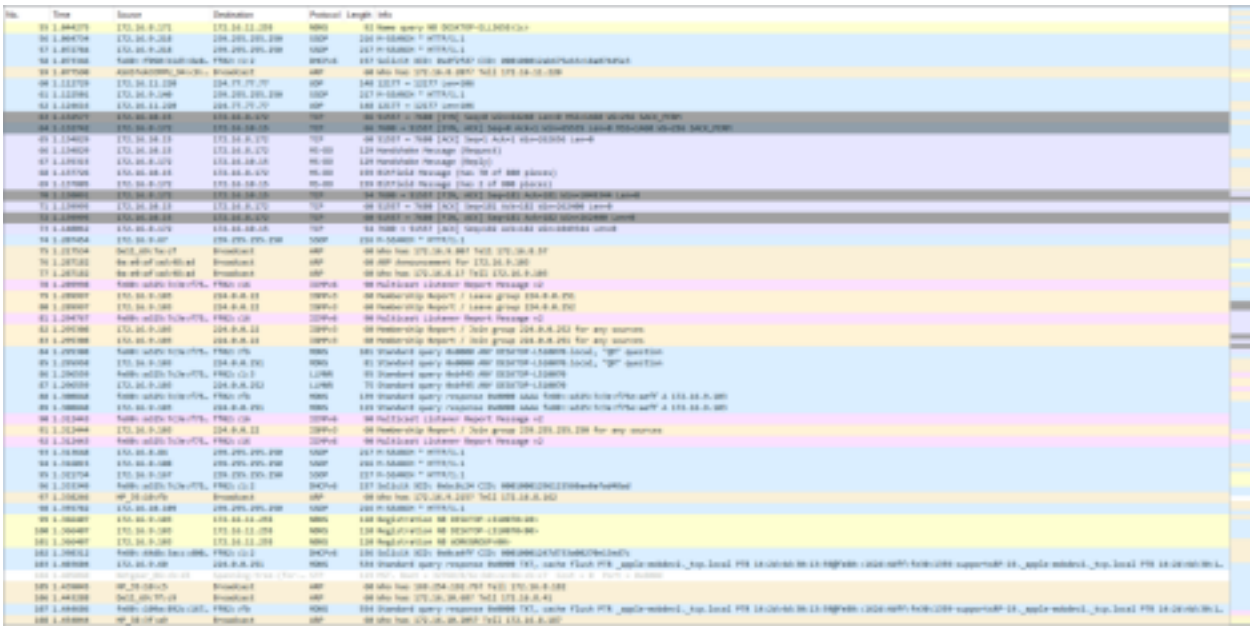
1. Capture 100 packets from the Ethernet: IEEE 802.3 LAN Interface and save

it. Procedure

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

Output





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

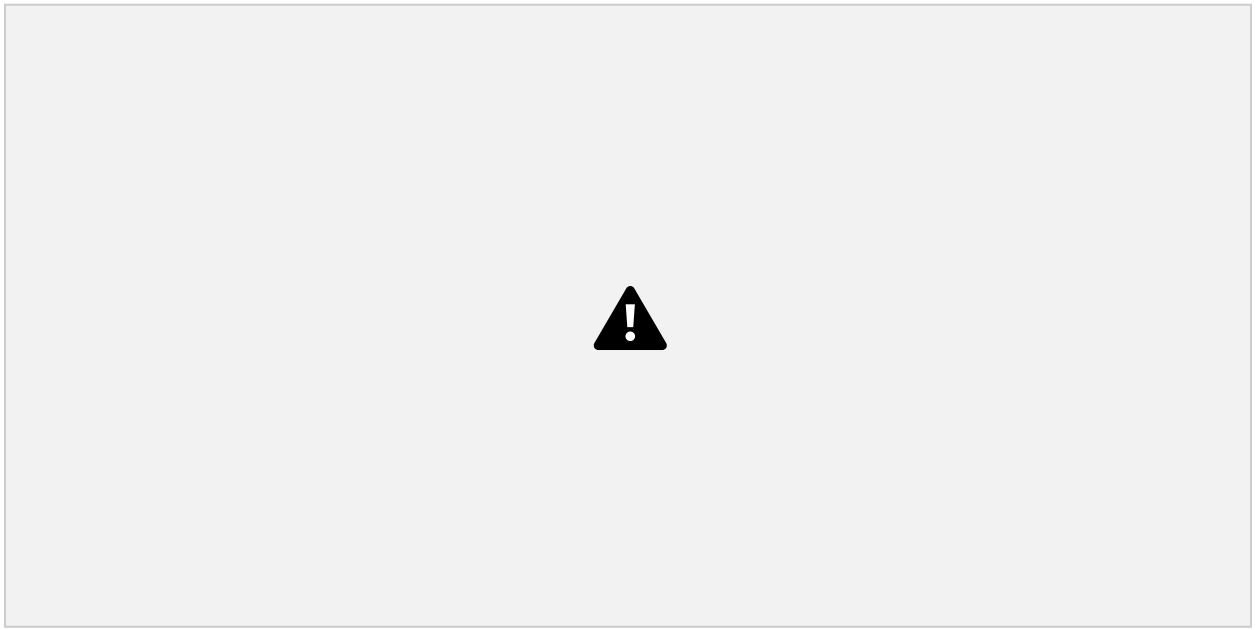
Procedure

- 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:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.8.172	172.16.8.165	MS-DO	58	KeepAlive Message
14	0.053403	172.16.8.165	172.16.8.172	TCP	60	50150 → 7680 [ACK] Seq=1 Ack=5 Win=1026 Len=0
16	0.060472	172.16.10.49	172.16.8.172	TCP	66	60679 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
17	0.060625	172.16.8.172	172.16.10.49	TCP	66	7680 → 60679 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
19	0.061601	172.16.10.49	172.16.8.172	TCP	60	60679 → 7680 [ACK] Seq=1 Ack=1 Win=131328 Len=0
20	0.061859	172.16.10.49	172.16.8.172	MS-DO	129	Handshake Message (Request)
21	0.062029	172.16.8.172	172.16.10.49	MS-DO	129	Handshake Message (Reply)
23	0.063515	172.16.10.49	172.16.8.172	MS-DO	91	BitField Message (has 44 of 256 pieces)
24	0.063569	172.16.8.172	172.16.10.49	MS-DO	91	BitField Message (has 2 of 256 pieces)
25	0.063699	172.16.8.172	172.16.10.49	TCP	54	7680 → 60679 [FIN, ACK] Seq=113 Ack=113 Win=1049600 Len=0
26	0.064508	172.16.10.49	172.16.8.172	TCP	60	60679 → 7680 [ACK] Seq=113 Ack=114 Win=131072 Len=0
28	0.064978	172.16.10.49	172.16.8.172	TCP	60	60679 → 7680 [FIN, ACK] Seq=113 Ack=114 Win=131072 Len=0
29	0.065009	172.16.8.172	172.16.10.49	TCP	54	7680 → 60679 [ACK] Seq=114 Ack=114 Win=1049600 Len=0
56	0.609533	172.16.10.200	172.16.8.172	MS-DO	60	KeepAlive Message
62	0.651125	172.16.8.172	172.16.10.200	TCP	54	7680 → 50408 [ACK] Seq=1 Ack=5 Win=4099 Len=0
80	0.924162	172.16.10.190	172.16.8.172	TCP	66	51020 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
81	0.924326	172.16.8.172	172.16.10.190	TCP	66	7680 → 51020 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
82	0.924481	172.16.10.190	172.16.8.172	TCP	60	51020 → 7680 [ACK] Seq=1 Ack=1 Win=262656 Len=0
83	0.924481	172.16.10.190	172.16.8.172	MS-DO	129	Handshake Message (Request)
84	0.925334	172.16.8.172	172.16.10.190	MS-DO	129	Handshake Message (Reply)
85	0.925798	172.16.10.190	172.16.8.172	MS-DO	68	BitField Message (has 18 of 72 pieces)
86	0.925859	172.16.8.172	172.16.10.190	MS-DO	68	BitField Message (has 4 of 72 pieces)
87	0.925985	172.16.8.172	172.16.10.190	TCP	54	7680 → 51020 [FIN, ACK] Seq=90 Ack=90 Win=1049600 Len=0
88	0.926651	172.16.10.190	172.16.8.172	TCP	60	51020 → 7680 [ACK] Seq=90 Ack=91 Win=262656 Len=0
89	0.926651	172.16.10.190	172.16.8.172	TCP	60	51020 → 7680 [FIN, ACK] Seq=90 Ack=91 Win=262656 Len=0
90	0.926695	172.16.8.172	172.16.10.190	TCP	54	7680 → 51020 [ACK] Seq=91 Ack=91 Win=1049600 Len=0
91	0.967290	172.16.10.62	172.16.8.172	MS-DO	60	KeepAlive Message
97	1.010286	172.16.8.172	172.16.10.62	TCP	54	51865 → 7680 [ACK] Seq=1 Ack=5 Win=4100 Len=0
192	2.111659	172.16.8.172	172.16.10.200	TCP	66	52059 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
193	2.113222	172.16.10.200	172.16.8.172	TCP	66	7680 → 52059 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
194	2.113384	172.16.8.172	172.16.10.200	TCP	54	52059 → 7680 [ACK] Seq=1 Ack=1 Win=131328 Len=0
195	2.113477	172.16.8.172	172.16.10.200	MS-DO	129	Handshake Message (Request)
196	2.115086	172.16.10.200	172.16.8.172	MS-DO	129	Handshake Message (Reply)
197	2.115086	172.16.10.200	172.16.8.172	TCP	60	7680 → 52059 [FIN, ACK] Seq=76 Ack=76 Win=2097920 Len=0
198	2.115074	172.16.8.172	172.16.10.200	TCP	54	52059 → 7680 [ACK] Seq=76 Ack=77 Win=131072 Len=0
199	2.115178	172.16.8.172	172.16.10.200	TCP	54	52059 → 7680 [FIN, ACK] Seq=76 Ack=77 Win=131072 Len=0
200	2.115950	172.16.10.200	172.16.8.172	TCP	60	7680 → 52059 [ACK] Seq=77 Ack=77 Win=2097920 Len=0

Flow Graph output



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

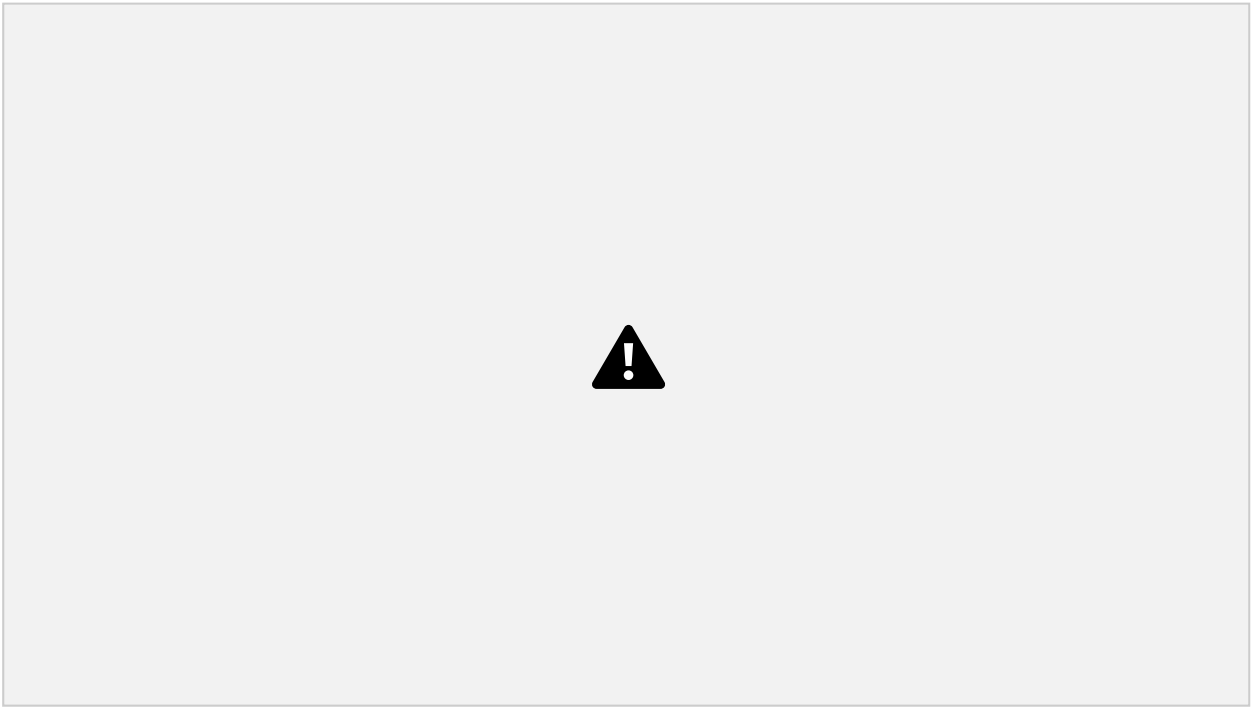
- Select Local Area Connection in Wireshark.
- Go to capture ✇option
- Select stop capture automatically after 100 packets.

 Then click Start capture.

 Search ARP packets in search bar.

 Save the packets.

Output



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.

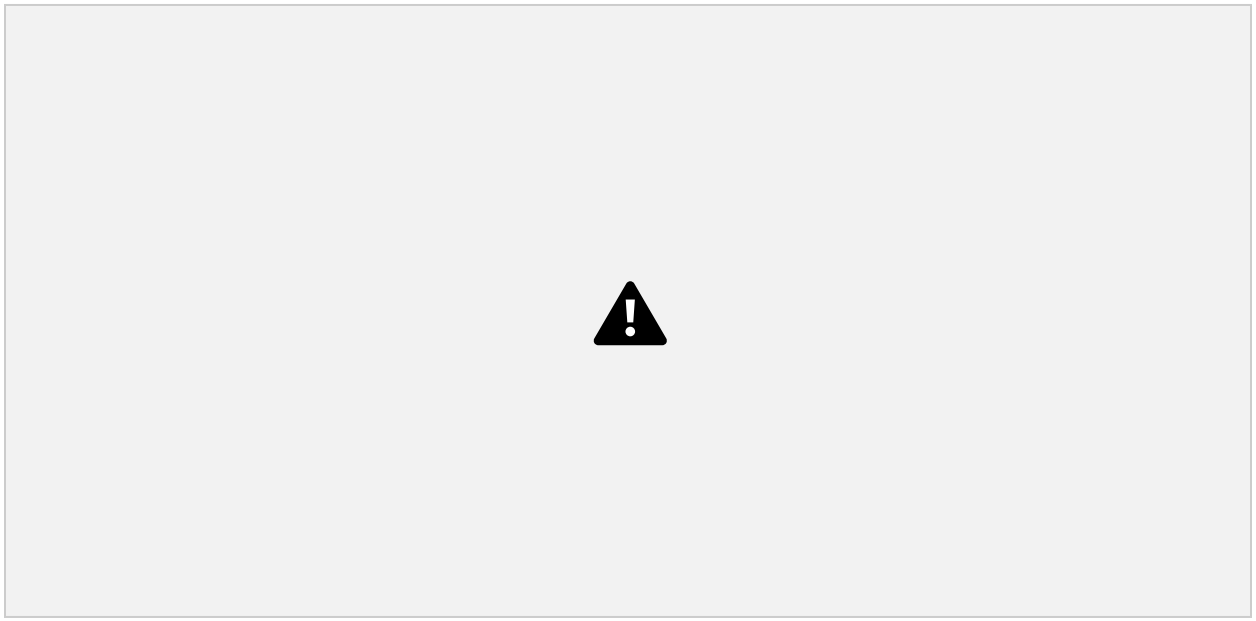
 Then click Start capture.

 Search DNS packets in search bar.

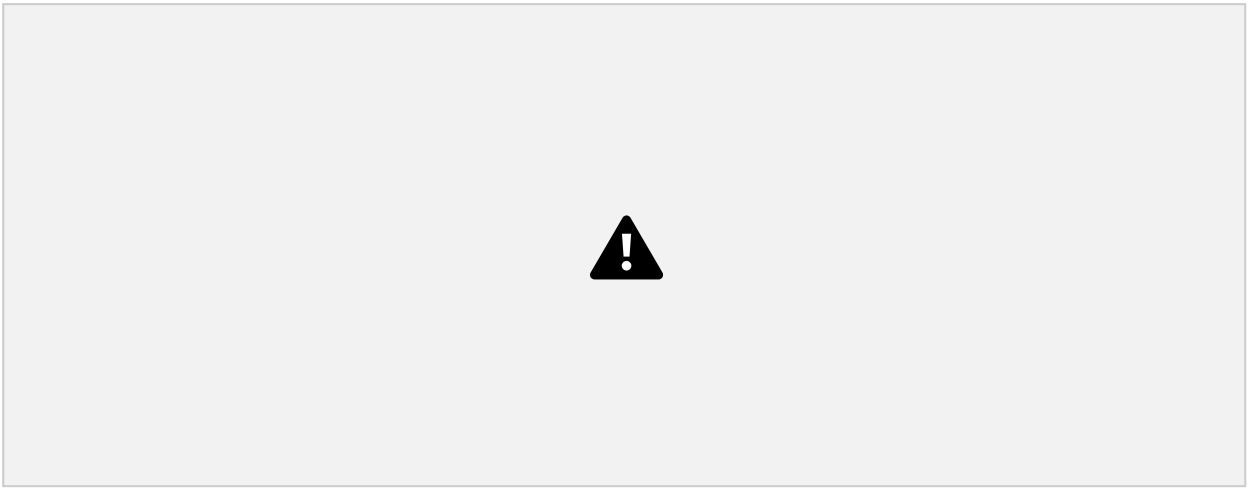
 To see flow graph click Statistics✇Flow graph.

 Save the packets.

Output



Flow Graph output

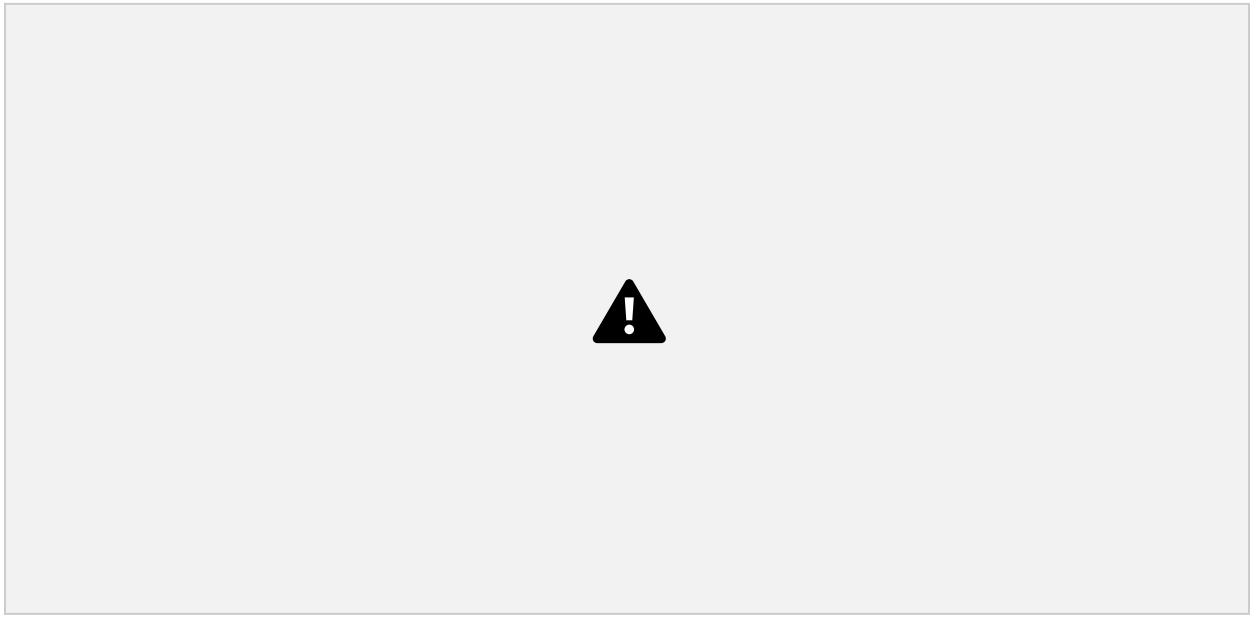


5.Create a Filter to display only HTTP packets and inspect the

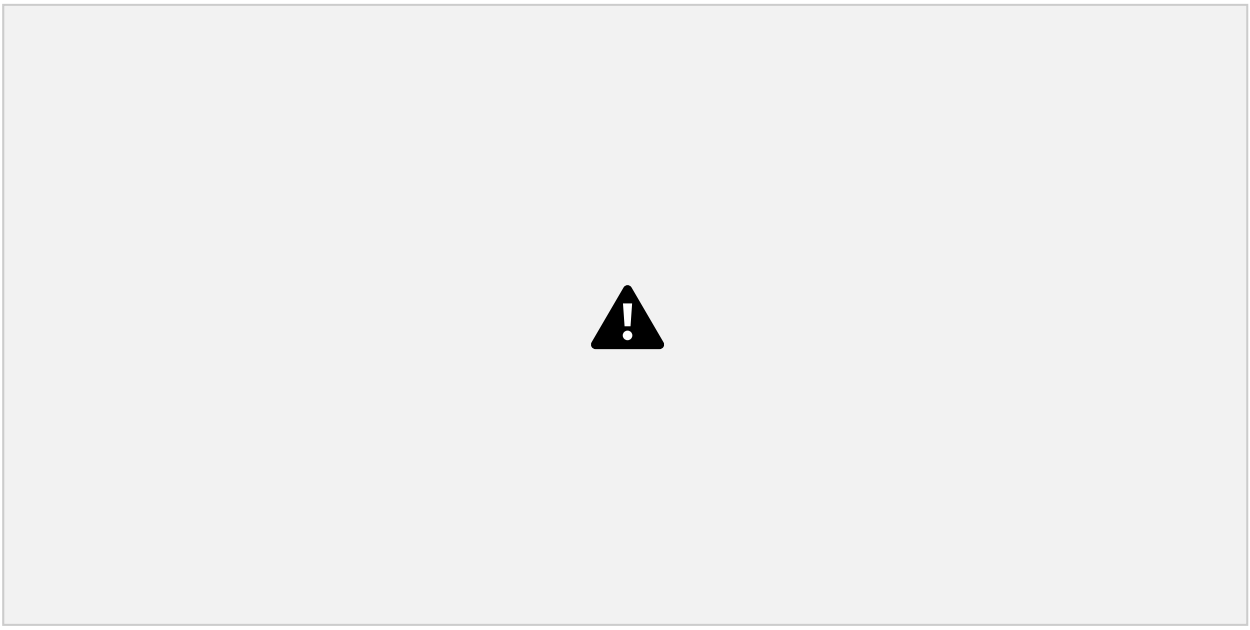
packets Procedure

- 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



Flow Graph output



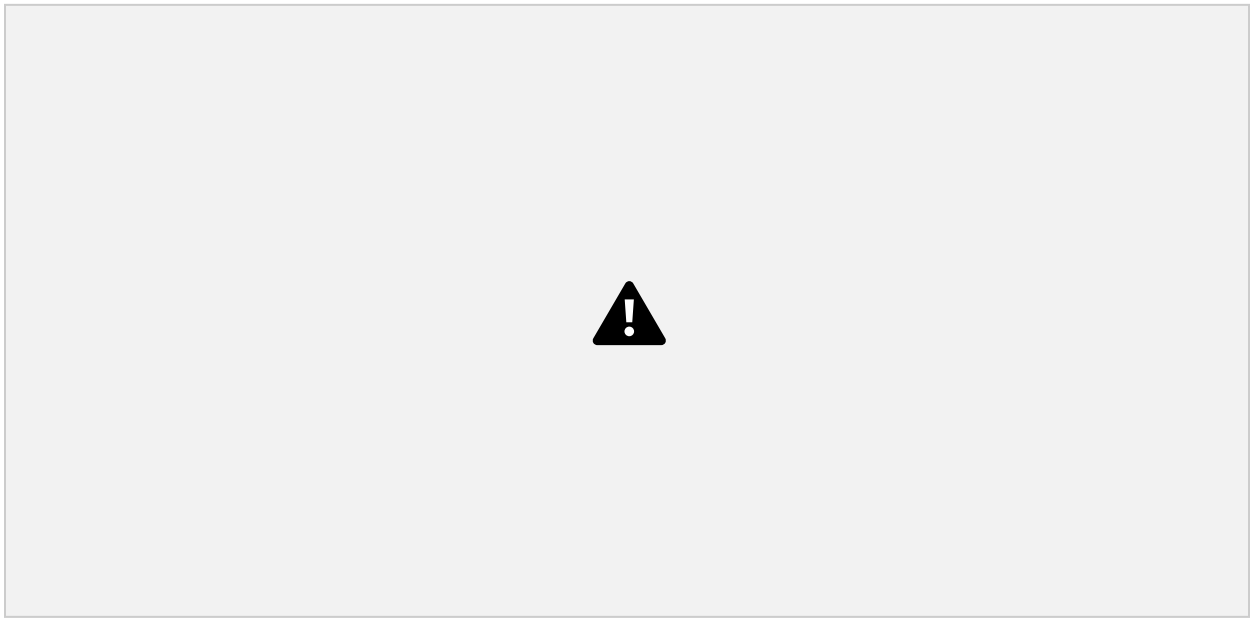
6.Create a Filter to display only IP/ICMP packets and inspect the packets.

Procedure

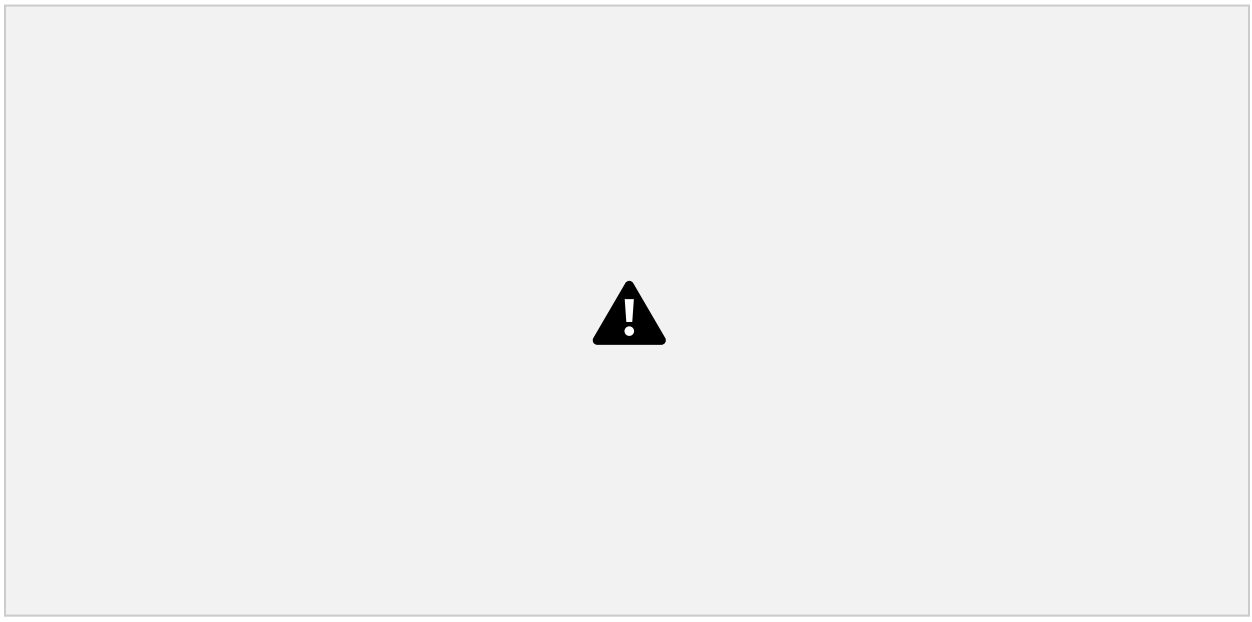
- Select Local Area Connection in Wireshark.
- Go to capture ✇option
- Select stop capture automatically after 100 packets.
- Then click Start capture.
- Search ICMP/IP packets in search bar.
- Save the packets

Output

ICMP:

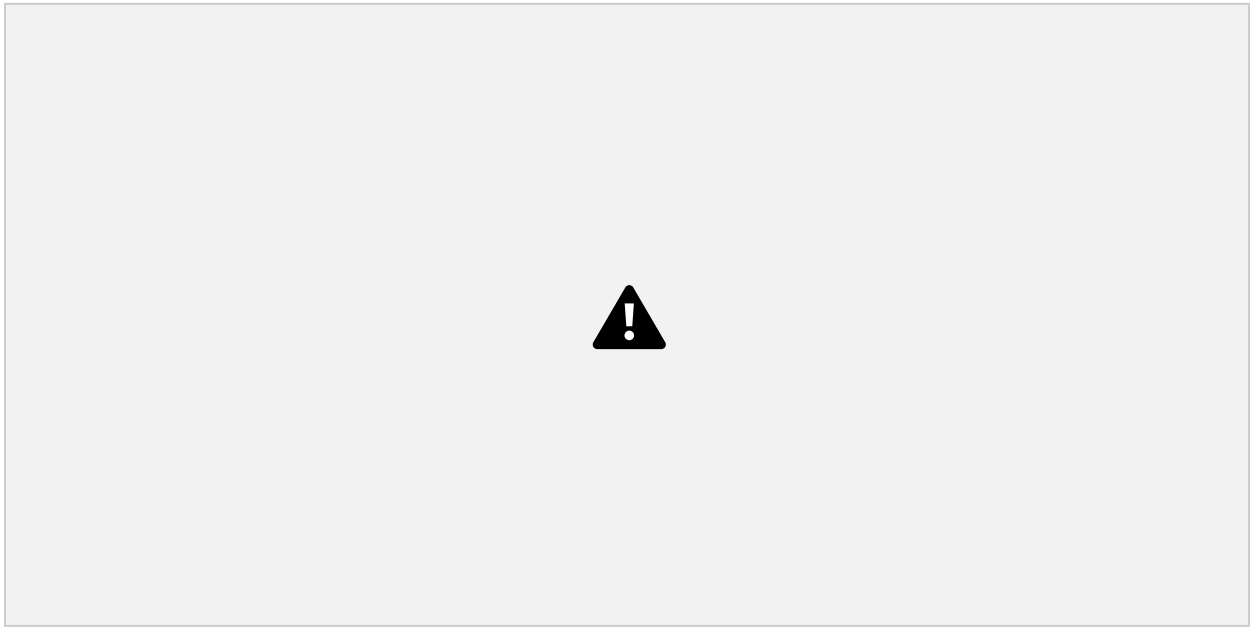


IP:

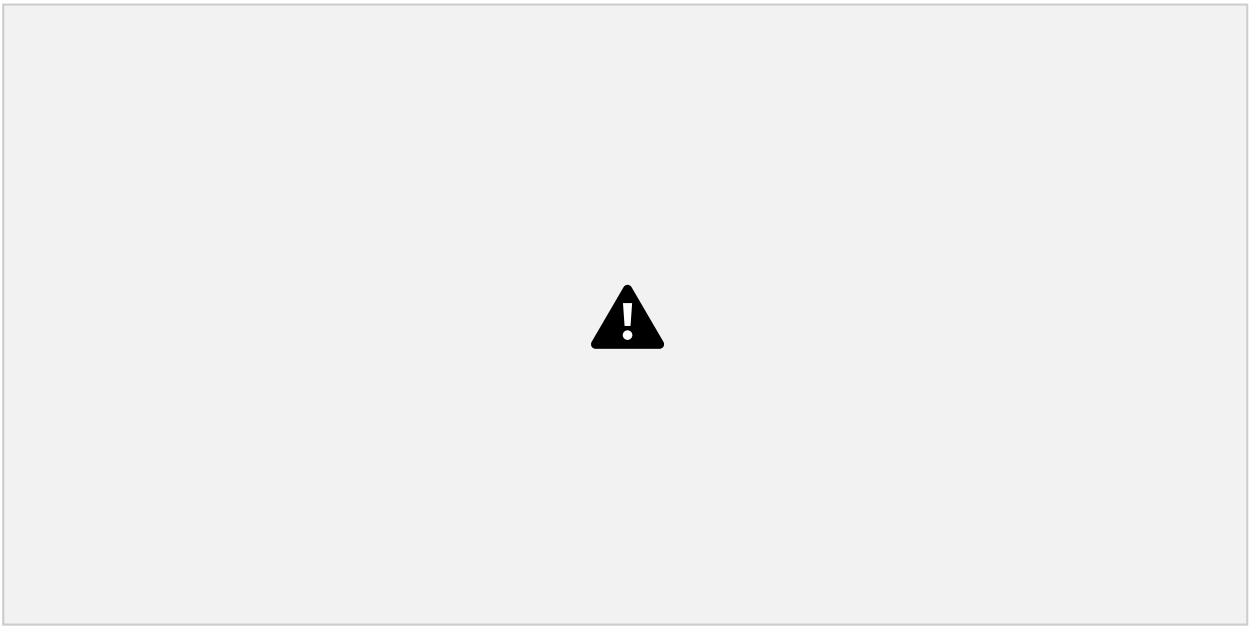


Flow Graph output:

ICMP:



IP:



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