

EXPERIMENT NO.5

AIM: Use Wireshark to understand the operation of TCP/IP layers:

- Ethernet Layer: Frame header, Frame size etc.
- Data Link Layer: MAC address, ARP (IP and MAC address binding)
- Network Layer: IP Packet (header, fragmentation), ICMP (Query and Echo)
- Transport Layer: TCP Ports, TCP handshake segments etc.

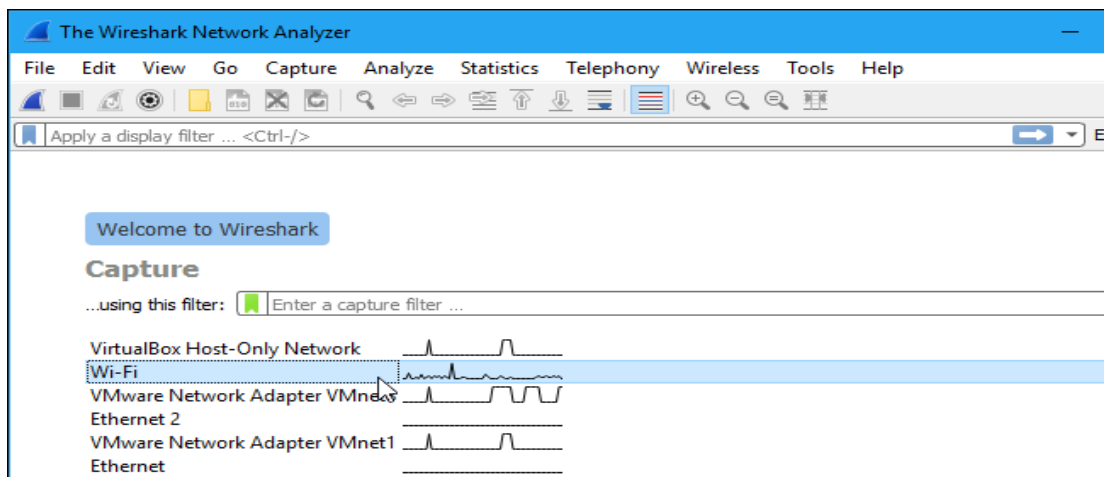
Application Layer: DHCP, FTP, HTTP header formats

THEORY:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and displays 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.

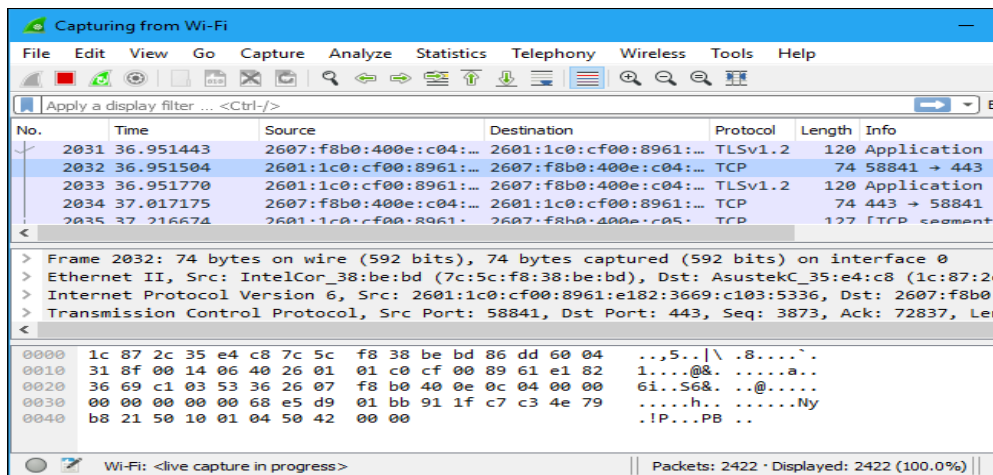
Capturing Packets

After downloading and installing Wireshark, you can launch it and double-click the name of a network interface under Capture to start capturing packets on that interface. For example, if you want to capture traffic on your wireless network, click your wireless interface. You can configure advanced features by clicking Capture > Options, but this isn't necessary for now.

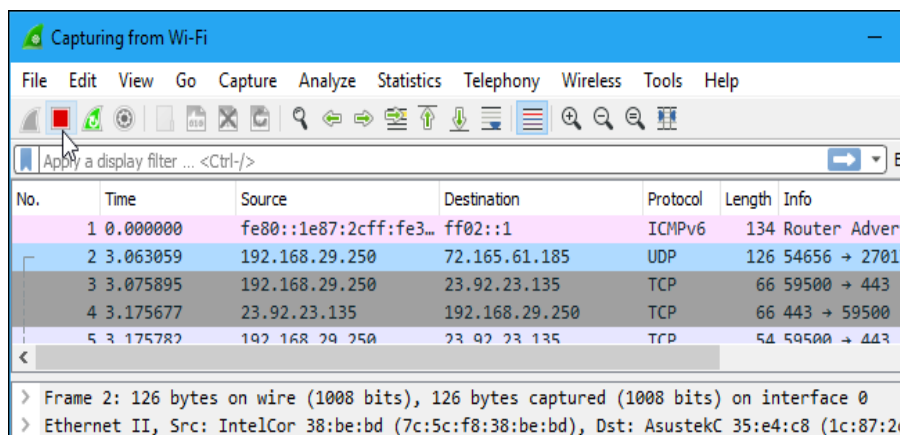


As soon as you click the interface's name, you'll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system.

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.



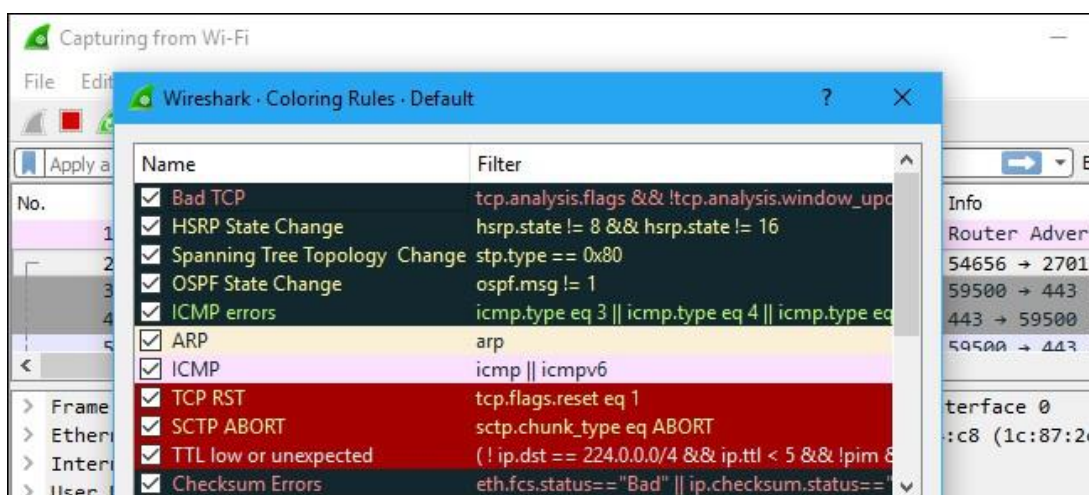
Click the red “Stop” button near the top left corner of the window when you want to stop capturing traffic.



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.

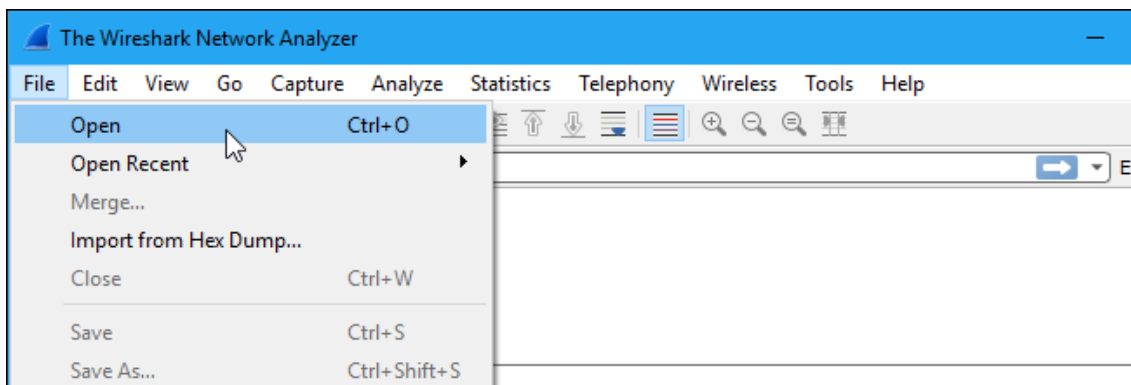
To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.



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.

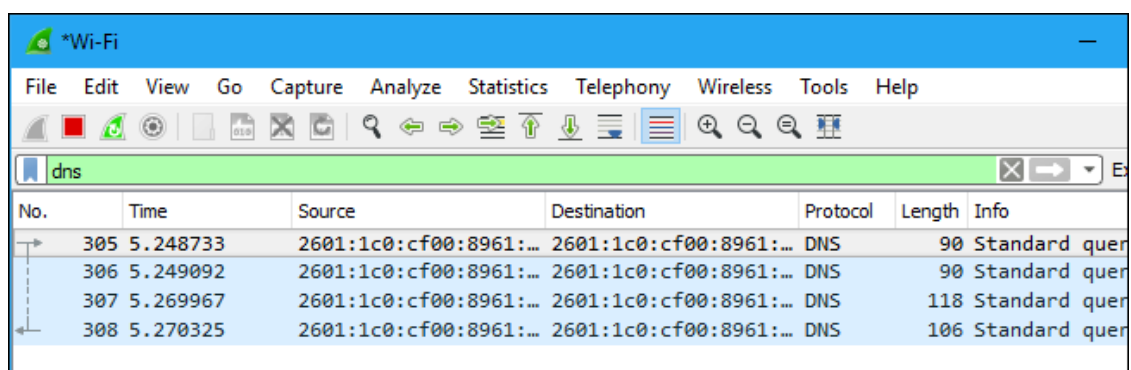
You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.



Filtering Packets

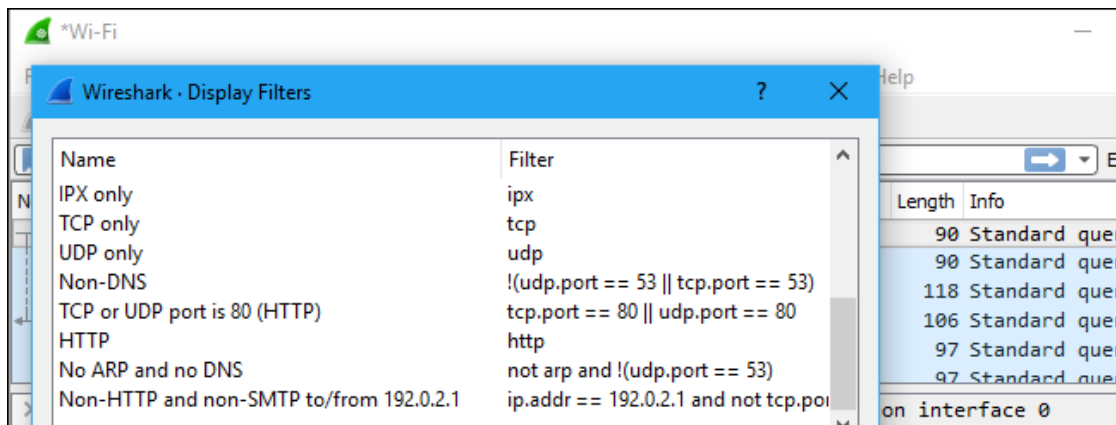
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a large amount of packets to sift through. That's where Wireshark's filters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



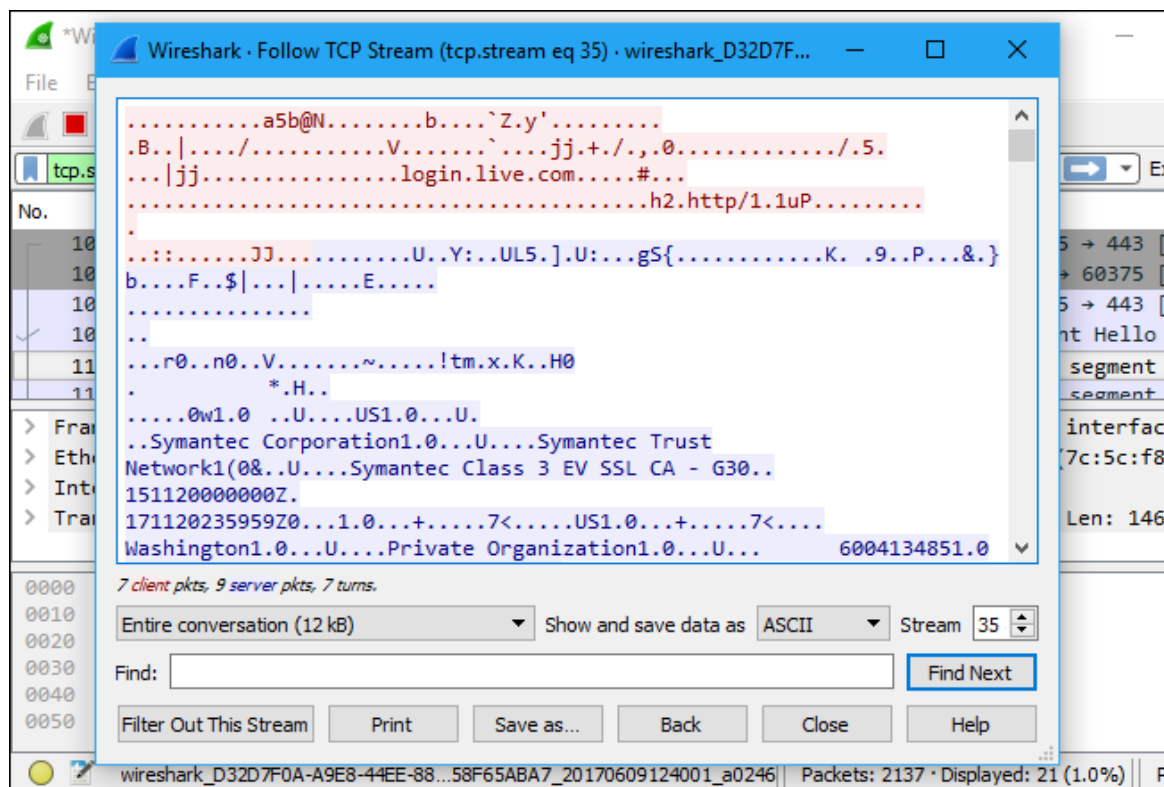
You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

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.

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
1103	2.992980	131.253.61.66	192.168.29.250	TCP	1514	[TCP segment
1104	2.992980	131.253.61.66	192.168.29.250	TCP	1514	[TCP segment

> Frame 1078: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
 > Ethernet II, Src: AsustekC_35:e4:c8 (1c:87:2c:35:e4:c8), Dst: IntelCor_38:be:bd (7c:5c:f8
 > Internet Protocol Version 4, Src: 131.253.61.66, Dst: 192.168.29.250
 > Transmission Control Protocol, Src Port: 443, Dst Port: 60375, Seq: 0, Ack: 1, Len: 0

Inspecting Packets

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

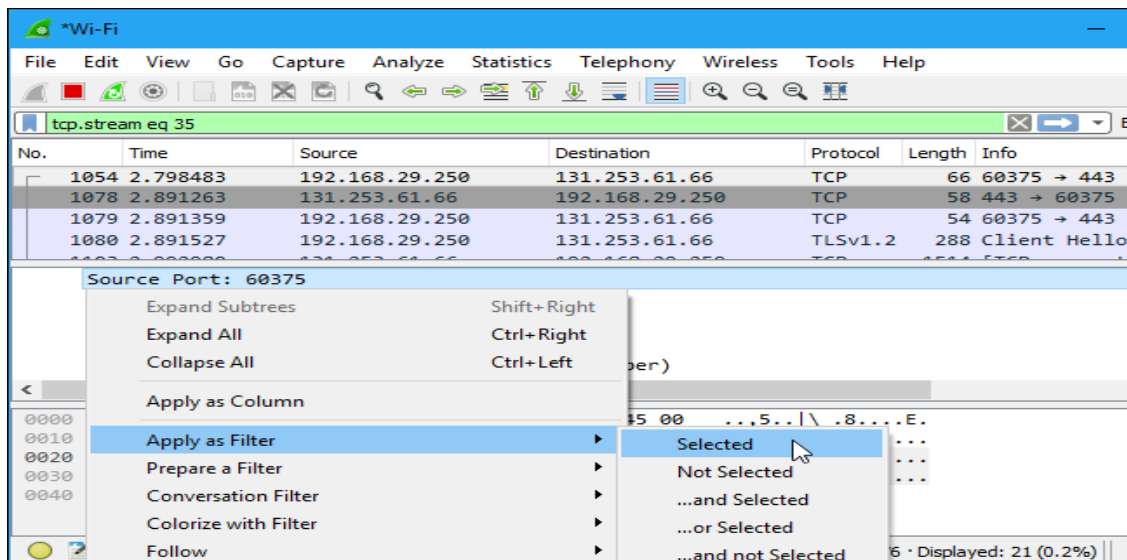
No.	Time	Source	Destination	Protocol	Length	Info
1054	2.798483	192.168.29.250	131.253.61.66	TCP	66	60375 → 443
1078	2.891263	131.253.61.66	192.168.29.250	TCP	58	443 → 60375
1079	2.891359	192.168.29.250	131.253.61.66	TCP	54	60375 → 443
1080	2.891527	192.168.29.250	131.253.61.66	TLSv1.2	288	Client Hello

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.

CONCLUSION: Thus, we have studied the working of Wire Shark.

Post Lab Assignments:

1. What is Ethernet?
2. Explain the significance of ARP and RARP.

CN Lab 5

Name: Mark Lopes
Branch: TE Comps A Batch C
Roll No: 9913

What is ethernet?

=> Ethernet is a widely used networking technology that connects devices in a local area network (LAN). It enables data transmission using cables, typically with speeds ranging from 10 Mbps to 100 Gbps. It's essential for wired networking in homes and businesses. Key features include:

1. High-speed data transfer.
2. Reliable and stable communication.
3. Uses twisted-pair or fiber optic cables.
4. Supports various topologies like star and bus.
5. Full-duplex and half-duplex modes for data flow.

2. Explain the significance of ARP and RARP.

=>

ARP	RARP
A protocol used to map an IP address to a physical address	A protocol used to map a physical address to an IP address
To obtain the MAC address of a network device when only its IP address is known	To obtain the IP address of a network device when only its MAC address is known
ARP stands for Address Resolution Protocol.	RARP stands for Reverse Address Resolution Protocol.
In ARP, broadcast MAC address is used.	RARP, broadcast IP address is used.
In ARP, ARP table is managed or maintained by local host.	RARP, RARP table is managed or maintained by RARP server.
In Address Resolution Protocol, Receiver's MAC address is fetched.	In RARP, IP address is fetched.
ARP is used in sender's side to map the receiver's MAC address.	RARP is used in receiver's side to map the sender's IP.

References:

https://www.geeksforgeeks.org/what-is-ethernet/?utm_source=perplexity

<https://www.geeksforgeeks.org/difference-between-arp-and-rarp/>