DOP: / /2023 **DOS: / /2023**

Experiment No: 08

Aim: Study of packet sniffer tools wireshark: -

a. Observer performance in promiscuous as well as non-promiscuous mode.
b. Show the packets can be traced based on different filters.

Theory:

Wireshark:

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

Uses of Wireshark:

Wireshark can be used in the following ways:

- It is used by network security engineers to examine security problems.
- It allows the users to watch all the traffic being passed over the network.
- It is used by network engineers to troubleshoot network issues.
- It also helps to troubleshoot latency issues and malicious activities on your network.
- It can also analyze dropped packets.

features:

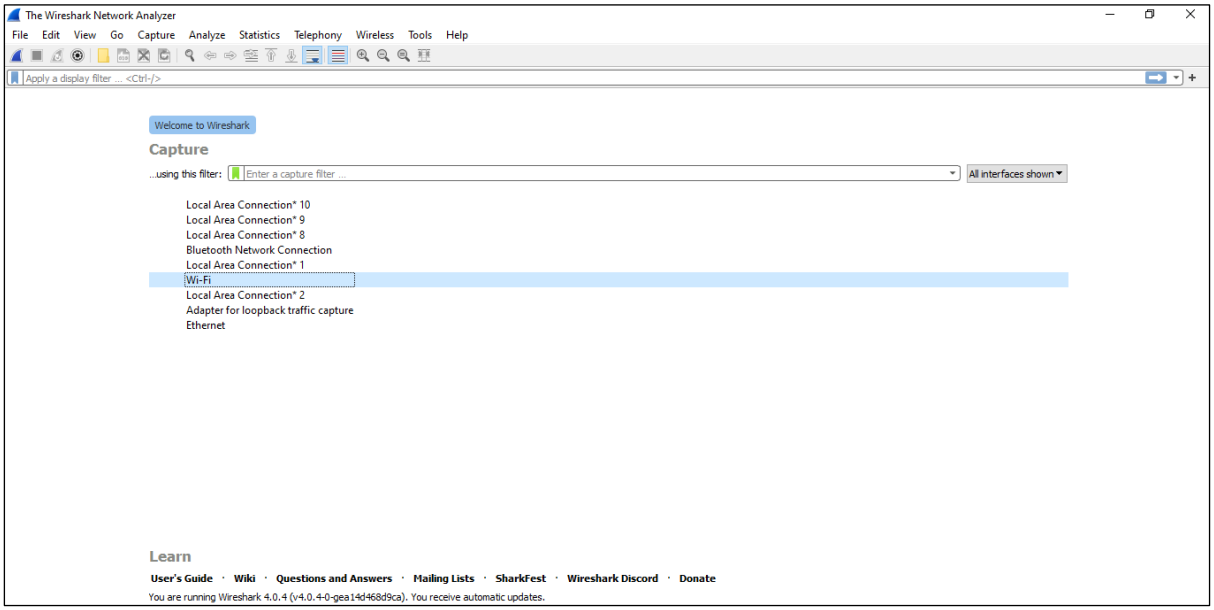
The following are some of the many features wireshark provides:

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark,
- and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.

Wireshark can also monitor the unicast traffic which is not sent to the network's MAC address interface. But the switch does not pass all the traffic to the port. Hence, the **promiscuous mode** is not sufficient to see all the traffic. The various network taps or port mirroring is used to extend capture at any point.

🔷Capturing Packets:

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

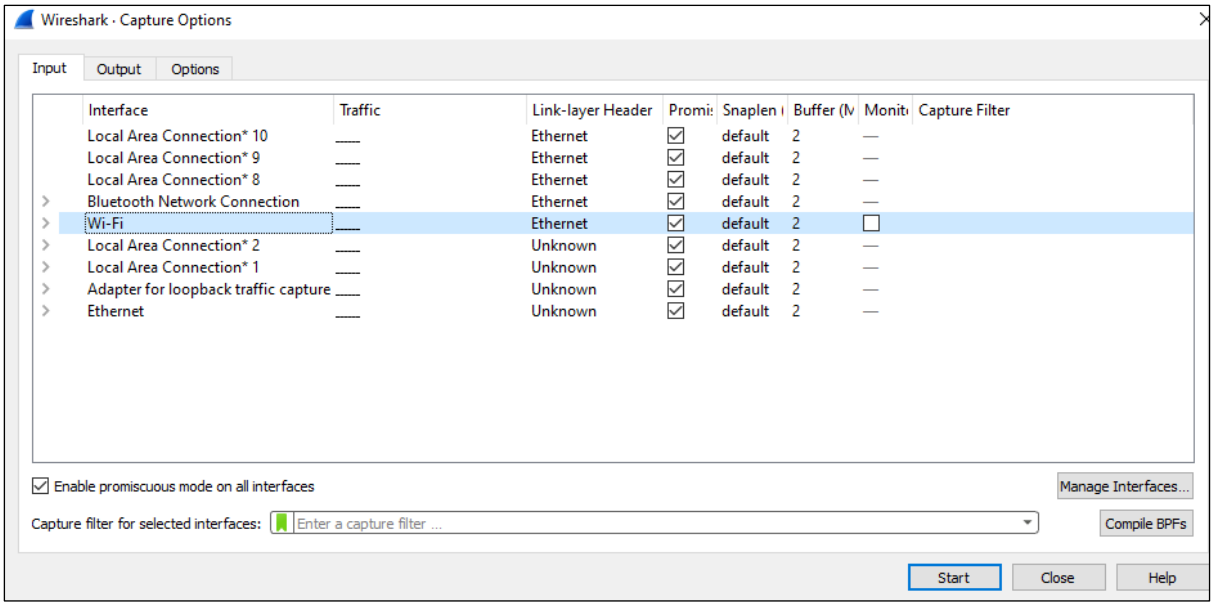


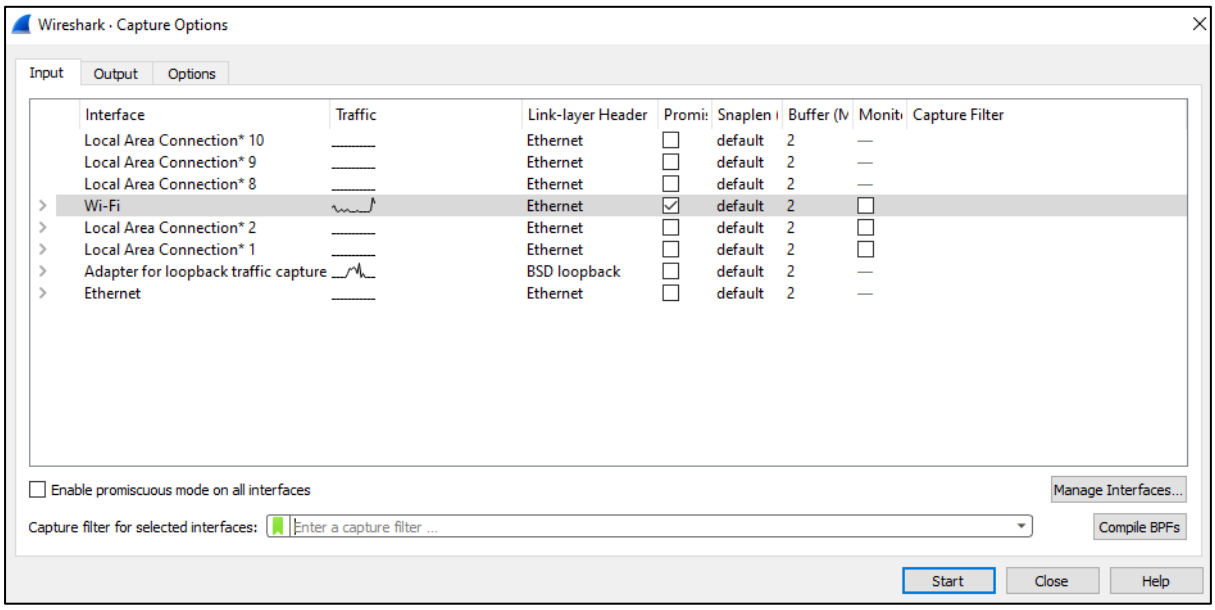
As soon as you click the interface‘s name, you‘ll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network.

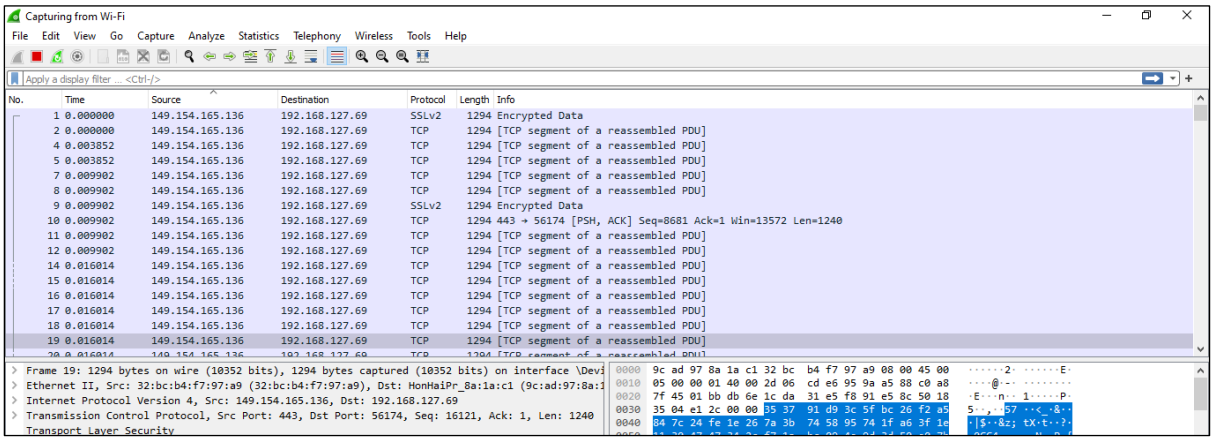
🔷performance in promiscuous as well as non-promiscuous mode:

To turn on promiscuous mode, click on the CAPTURE OPTIONS dialog box and select it from the options. If everything goes according to plan, you'll now see all the network traffic in your network. However, many network interfaces aren't receptive to promiscuous mode, so don't be alarmed if it doesn't work for you.

Click on the network and make sure the promiscuous mode settings are set to ALLOW ALL. Promiscuous mode enables lots of Wireshark's functions, so you should do all you can to make sure your interface can use it, if possible.


Wireshark · Capture Options
Input
Output
Options
Interface
Traffic
Link-layer Header
Promi
Snaplen
Buffer (M
Monit
Capture Filter
Local Area Connection* 10 Ethernet default 2
Local Area Connection* 9 Ethernet default 2
Local Area Connection* 8 Ethernet default 2
Bluetooth Network Connection Ethernet default 2
Wi-Fi Ethernet default 2
Local Area Connection* 2 Unknown default 2
Local Area Connection* 1 Unknown default 2
Adapter for loopback traffic capture Unknown default 2
Ethernet Unknown default 2
Enable promiscuous mode on all interfaces
Manage Interfaces...
Capture filter for selected interfaces:
Enter a capture filter ...
Compile BPFs
Start
Close
Help


Wireshark · Capture Options
Input
Output
Options
Interface
Traffic
Link-layer Header
Promi
Snaplen
Buffer (M
Monit
Capture Filter
Local Area Connection* 10 Ethernet default 2
Local Area Connection* 9 Ethernet default 2
Local Area Connection* 8 Ethernet default 2
Wi-Fi Ethernet default 2
Local Area Connection* 2 Ethernet default 2
Local Area Connection* 1 Ethernet default 2
Adapter for loopback traffic capture BSD loopback default 2
Ethernet Ethernet default 2
Enable promiscuous mode on all interfaces
Manage Interfaces...
Capture filter for selected interfaces:
Enter a capture filter ...
Compile BPFs
Start
Close
Help

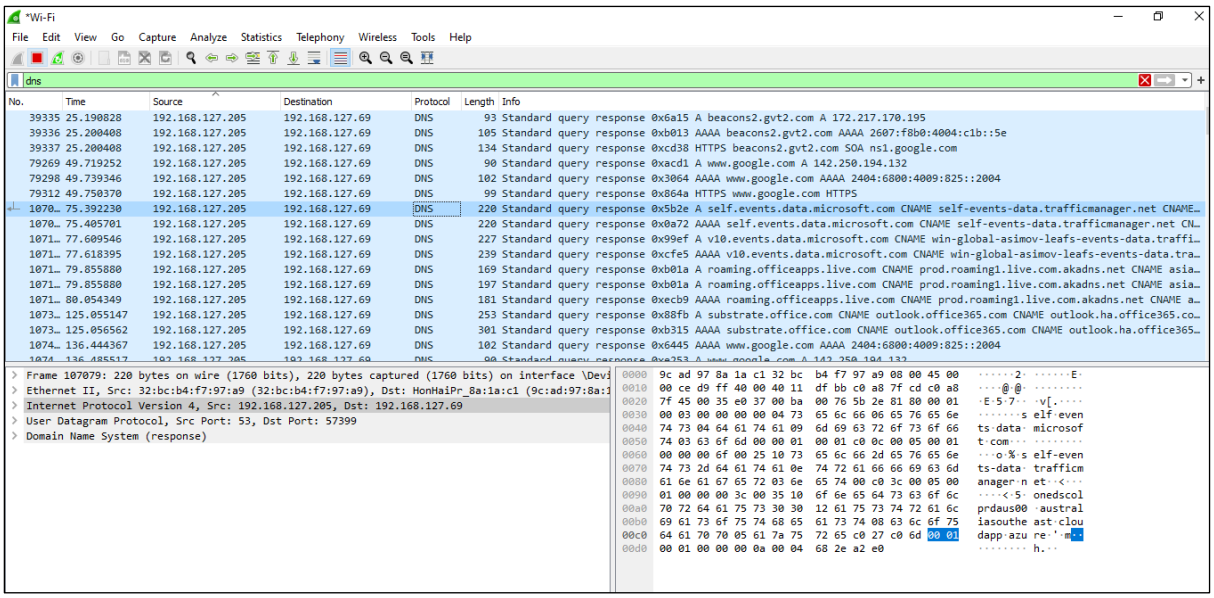

Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
No. Time Source Destination Protocol Length Info
1 0.000000 149.154.165.136 192.168.127.69 SSLv2 1294 Encrypted Data
2 0.000000 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
4 0.003852 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
5 0.003852 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
7 0.009902 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
8 0.009902 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
9 0.009902 149.154.165.136 192.168.127.69 SSLv2 1294 Encrypted Data
10 0.009902 149.154.165.136 192.168.127.69 TCP 1294 443 → 56174 [PSH, ACK] Seq=8681 Ack=1 Win=13572 Len=1240
11 0.009902 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
12 0.009902 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
14 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
15 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
16 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
17 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
18 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
19 0.016014 149.154.165.136 192.168.127.69 TCP 1294 [TCP segment of a reassembled PDU]
Frame 19: 1294 bytes on wire (10352 bits), 1294 bytes captured (10352 bits) on interface \Devi
Ethernet II, Src: 32:bc:b4:f7:97:a9 (32:bc:b4:f7:97:a9), Dst: HonHaiPr_8a:1a:c1 (9c:ad:97:8a:1
Internet Protocol Version 4, Src: 149.154.165.136, Dst: 192.168.127.69
Transmission Control Protocol, Src Port: 443, Dst Port: 56174, Seq: 16121, Ack: 1, Len: 1240
Transport Layer Security
0000 9c ad 97 8a 1a c1 32 bc b4 f7 97 a9 08 00 45 00
0010 05 00 00 01 40 00 2d 06 cd e6 95 9a a5 88 c0 a8
0020 7f 45 01 bb db 6e 1c da 31 e5 f8 91 e5 8c 50 18
0030 35 04 e1 2c 00 00 35 37 91 d9 3c 5f bc 26 f2 a5
0040 84 7c 24 fe 1e 26 7a 3b 74 58 95 74 1f a6 3f 1e

Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

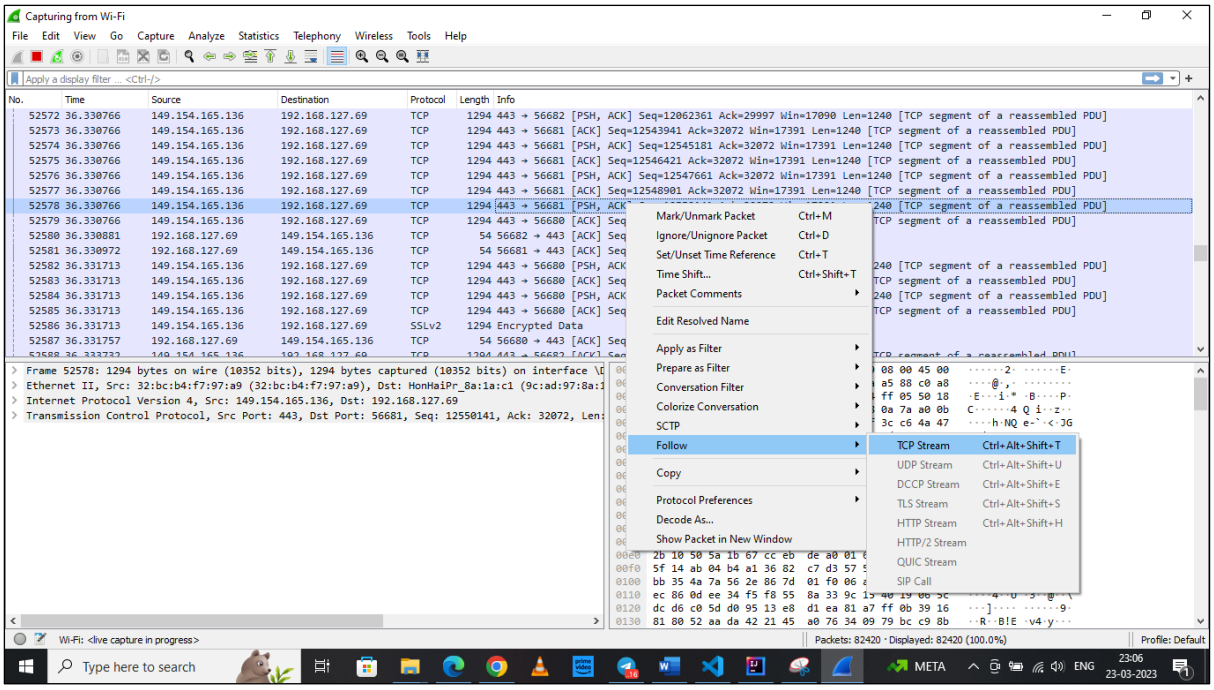
🔷Filtering Packets:

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.

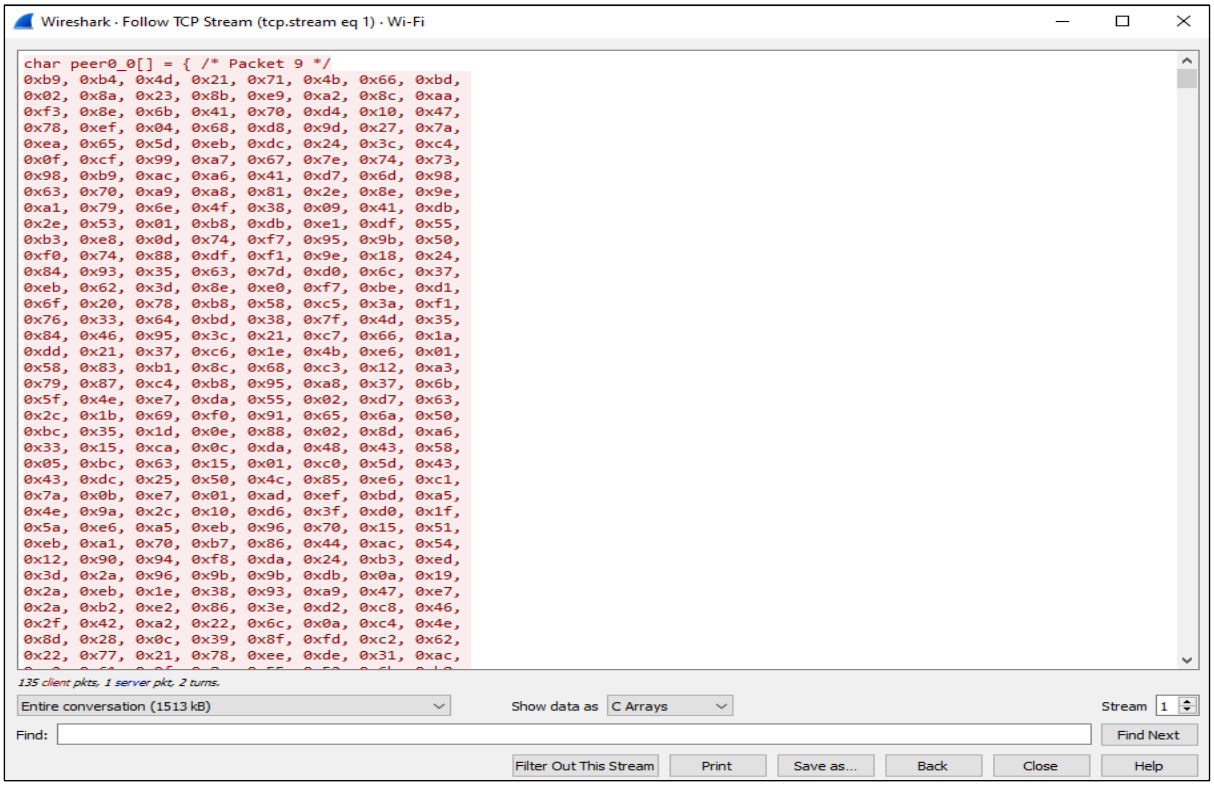
The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type ―dns‖ and you‘ll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



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



You'll see the full conversation between the client and the server.

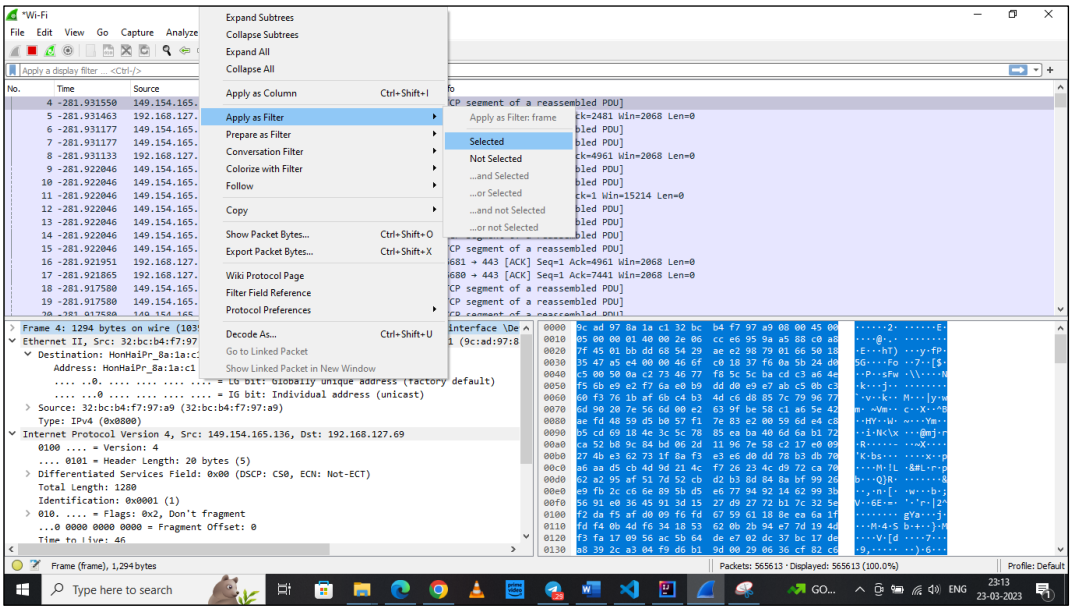


Close the window and you'll find a filter has been applied automatically — Wireshark is showing you the packets that make up the conversation.

🔷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.

Conclusion: -

In this experiment we analyze various packet sniffing tools that monitor network traffic transmitted between legitimate users or in the network. The packet sniffer is network monitoring tool. It is opted for network monitoring, traffic analysis, troubleshooting, Packet grapping, message, protocol analysis, penetration testing and many other purposes