Semester: V (2023-24) **Subject:** Communication Protocols

Name: Shreerang Mhatre **Class: TY**

Roll No: 52 **Batch: A3**

Experiment No: 05

Name of the Experiment: Study of Network Protocol Analyser Tool/Software.

Performed on: 25/10/2023

Submitted on: 30/10/2023

Aim: Study of Network Protocol Analyser Tool/Software.

Prerequisite:

- Basic knowledge of Data communication and Networking protocol.

Objectives:

- Download Wireshark on Windows, Capture data packets and analyse them.

Components and equipment required/studied:

- Computer with Operating System installed and Internet connection
- Wireshark software.

Theory:

Wireshark is a free application that allows you to capture and view the data traveling back and forth on your network, providing the ability to drill down and read the contents of each packet – filtered to meet your specific needs. It is commonly utilized to troubleshoot network problems as well as to develop and test software. It is used for network troubleshooting, analysis, software and communications protocol development, and education.

This open-source protocol analyser is widely accepted as the industry standard, winning its fair share of awards over the years. There is also a terminal-based (non-GUI) version called Shark.

Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License. Originally known as Ethereal, Wireshark features a user-friendly interface that can display data from hundreds of different protocols on all major network types. These data packets can be viewed in Real-time or analysed offline, with dozens of capture/trace file formats supported including CAP and ERF. Integrated decryption tools allow you to view encrypted packets for several popular protocols such as WEP and WPA/WPA2.

Wireshark lets the user put network interface controllers into promiscuous mode (if supported by the network interface controller), so they can see all the traffic visible on that interface including unicast traffic not sent to that network interface controller's MAC address. However, when capturing with a packet analyser in promiscuous mode on a port on a network switch, not all traffic through the switch is necessarily sent to the port where the capture is done, so capturing in promiscuous mode is not necessarily sufficient to see all network traffic.

Port mirroring or various network taps extend capture to any point on the network. If a remote machine captures packets and sends the captured packets to a machine running Wireshark using the TZSP protocol or the protocol used by OmniPeek, Wireshark dissects those packets, so it can analyse packets captured on a remote machine at the time that they are captured.

Features:

Wireshark is a data capturing program that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark uses pcap to capture packets, so it can only capture packets on the types of networks that pcap supports. Data can be captured "from the wire" from a live network connection or read from a file of already-captured packets. Live data can be read from different types of networks, including Ethernet, IEEE 802.11, PPP, and loopback.

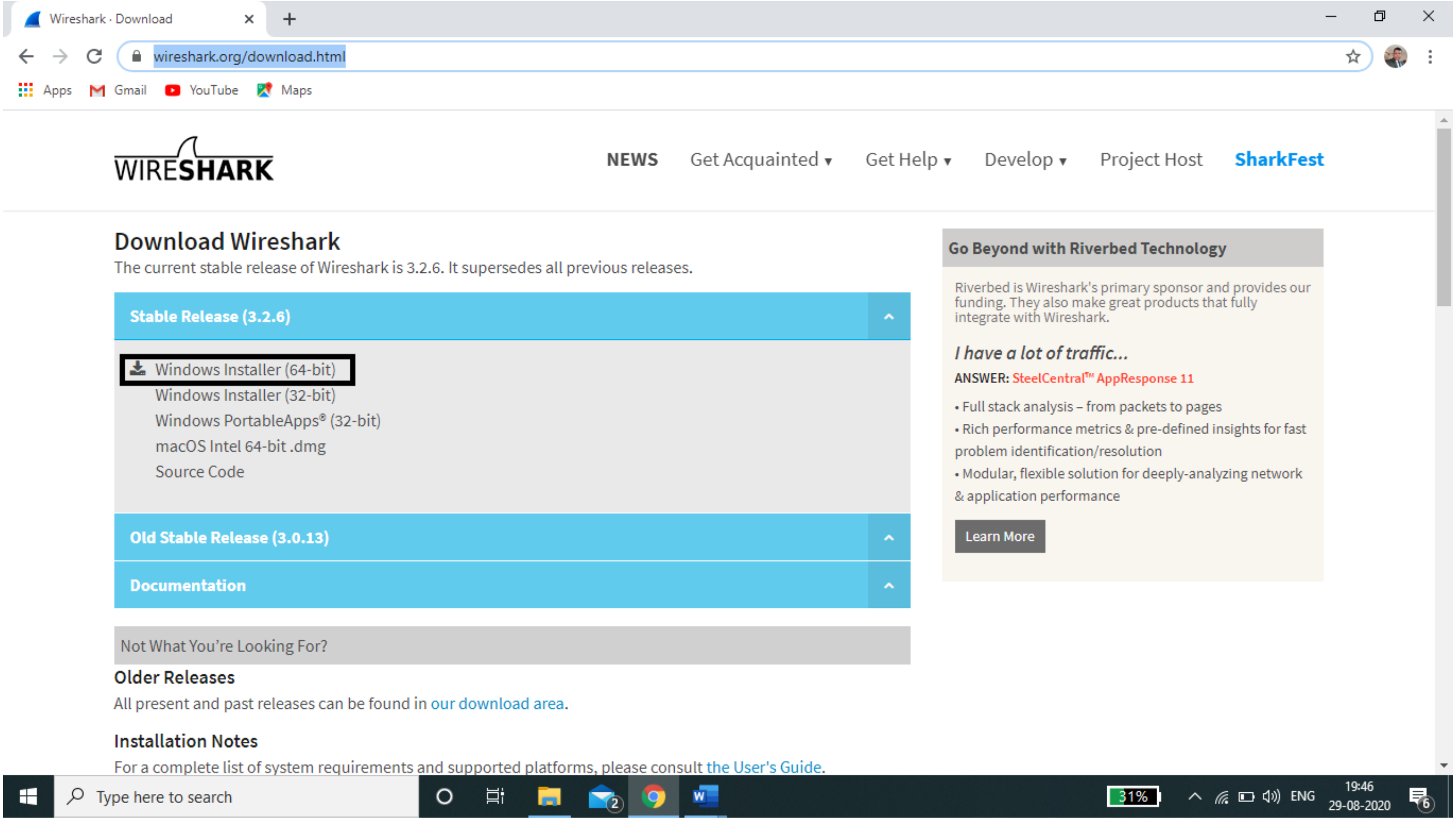
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
- Captured files can be programmatically edited or converted via command-line switches to the "edit-cap" program.
- Data display can be refined using a display filter.
- Plug-ins can be created for dissecting new protocols.

- VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
- Raw USB traffic can be captured.
- Wireless connections can also be filtered as long as they traverse the monitored Ethernet.
- Various settings, timers, and filters can be set to provide the facility of filtering the output of the captured traffic.

Procedure:

Downloading and Installing Wireshark: https://www.wireshark.org/download.html

Wireshark can be downloaded at no cost from the Wireshark Foundation website for both macOS and Windows operating systems via above link. It is recommended that to download the latest stable release.



Download Wireshark

The current stable release of Wireshark is 3.2.6. It supersedes all previous releases.

Stable Release (3.2.6)

- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (32-bit)
- macOS Intel 64-bit .dmg
- Source Code

Old Stable Release (3.0.13)

Documentation

Not What You're Looking For?

Older Releases

All present and past releases can be found in our download area.

Installation Notes

For a complete list of system requirements and supported platforms, please consult the User's Guide.

Go Beyond with Riverbed Technology

Riverbed is Wireshark's primary sponsor and provides our funding. They also make great products that fully integrate with Wireshark.

I have a lot of traffic...

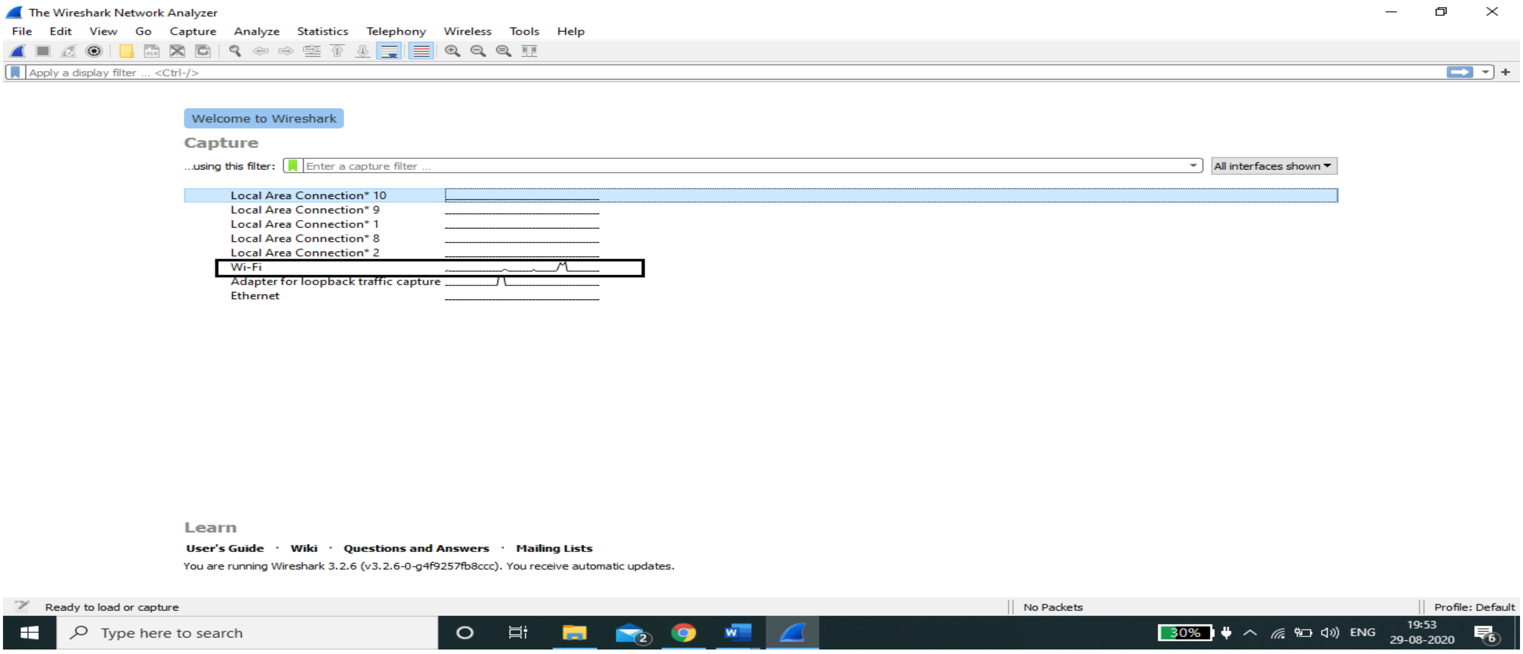
ANSWER: SteelCentral™ AppResponse 11

- Full stack analysis – from packets to pages
- Rich performance metrics & pre-defined insights for fast problem identification/resolution
- Modular, flexible solution for deeply-analyzing network & application performance

Learn More

Fig. 5.1

After downloading and installing the app, run the application and it will open a dialog box in which you can see all networks. There you click on Wi-Fi option; you can also take other option if available like Ethernet connection. (Fig. 5.2).



Apply a display filter ... <Ctrl-/>

...using this filter: Enter a capture filter ...   All interfaces shown

Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 1
Local Area Connection* 8
Local Area Connection* 2
Wi-Fi
Adapter for loopback traffic capture
Ethernet

You are running Wireshark 3.2.6 (v3.2.6-0-g4f9257fb8ccc). You receive automatic updates.

Ready to load or capture   No Packets   Profile: Default

19:53
29-08-2020

Fig. 5.2

For capturing packets, first select one or more of these networks by clicking on your choice(s) and using the *Shift* or *Ctrl* keys if you'd like to record data from multiple networks simultaneously. Once a connection type is selected for capturing purposes, its background will be shaded in either blue or grey. Click on *Capture* from the main menu, located towards the top of the Wireshark interface. When the dropdown menu appears, select the *Start* option.

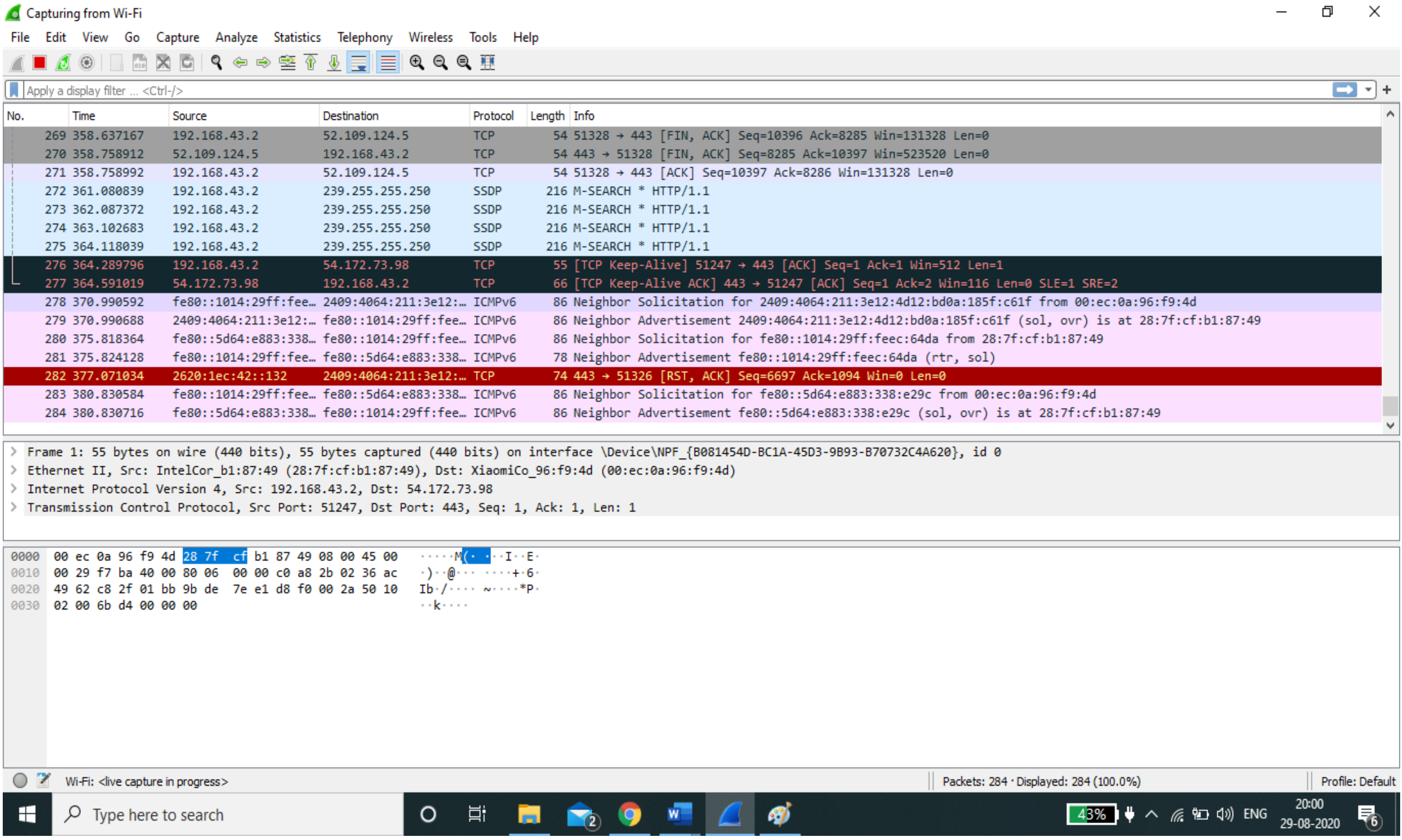
You can also initiate packet capturing via one of the following shortcuts.

- **Keyboard:** Press *Ctrl* + *E*
- **Mouse:** To begin capturing packets from one particular network, simply double-click on its name
- **Toolbar:** Click on the blue shark fin button, located on the far left-hand side of the Wireshark toolbar

The live capture process will now begin, with packet details displayed in the Wireshark window as they are recorded. Perform one of the actions below to stop capturing.

- **Keyboard:** Press *Ctrl* + *E*

- **Toolbar:** Click on the red stop button, located next to the shark fin on the Wireshark toolbar.



```
No.     Time        Source                 Destination            Protocol  Length Info
    269 358.637167  192.168.43.2           52.109.124.5           TCP        54 51328 → 443 [FIN, ACK] Seq=10396 Ack=8285 Win=131328 Len=0
    270 358.758912  52.109.124.5           192.168.43.2           TCP        54 443 → 51328 [FIN, ACK] Seq=8285 Ack=10397 Win=523520 Len=0
    271 358.758992  192.168.43.2           52.109.124.5           TCP        54 51328 → 443 [ACK] Seq=10397 Ack=8286 Win=131328 Len=0
    272 361.080839  192.168.43.2           239.255.255.250        SSDP      216 M-SEARCH * HTTP/1.1
    273 362.087372  192.168.43.2           239.255.255.250        SSDP      216 M-SEARCH * HTTP/1.1
    274 363.102683  192.168.43.2           239.255.255.250        SSDP      216 M-SEARCH * HTTP/1.1
    275 364.118039  192.168.43.2           239.255.255.250        SSDP      216 M-SEARCH * HTTP/1.1
    276 364.289796  192.168.43.2           54.172.73.98           TCP        55 [TCP Keep-Alive] 51247 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1
    277 364.591019  54.172.73.98           192.168.43.2           TCP        66 [TCP Keep-Alive ACK] 443 → 51247 [ACK] Seq=1 Ack=2 Win=116 Len=0 SLE=1 SRE=2
    278 370.990592  fe80::1014:29ff:fee…   2409:4064:211:3e12:…   ICMPv6     86 Neighbor Solicitation for 2409:4064:211:3e12:4d12:bd0a:185f:c61f from 00:ec:0a:96:f9:4d
    279 370.990688  2409:4064:211:3e12:…   fe80::1014:29ff:fee…   ICMPv6     86 Neighbor Advertisement 2409:4064:211:3e12:4d12:bd0a:185f:c61f (sol, ovr) is at 28:7f:cf:b1:87:49
    280 375.818364  fe80::5d64:e883:338…   fe80::1014:29ff:fee…   ICMPv6     86 Neighbor Solicitation for fe80::1014:29ff:feec:64da from 28:7f:cf:b1:87:49
    281 375.824128  fe80::1014:29ff:fee…   fe80::5d64:e883:338…   ICMPv6     78 Neighbor Advertisement fe80::1014:29ff:feec:64da (rtr, sol)
    282 377.071034  2620:1ec:42::132       2409:4064:211:3e12:…   TCP        74 443 → 51326 [RST, ACK] Seq=6697 Ack=1094 Win=0 Len=0
    283 380.830584  fe80::1014:29ff:fee…   fe80::5d64:e883:338…   ICMPv6     86 Neighbor Solicitation for fe80::5d64:e883:338:e29c from 00:ec:0a:96:f9:4d
    284 380.830716  fe80::5d64:e883:338…   fe80::1014:29ff:fee…   ICMPv6     86 Neighbor Advertisement fe80::5d64:e883:338:e29c (sol, ovr) is at 28:7f:cf:b1:87:49

> Frame 1: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{B081454D-BC1A-45D3-9B93-B70732C4A620}, id 0
> Ethernet II, Src: IntelCor_b1:87:49 (28:7f:cf:b1:87:49), Dst: XiaomiCo_96:f9:4d (00:ec:0a:96:f9:4d)
> Internet Protocol Version 4, Src: 192.168.43.2, Dst: 54.172.73.98
> Transmission Control Protocol, Src Port: 51247, Dst Port: 443, Seq: 1, Ack: 1, Len: 1

0000  00 ec 0a 96 f9 4d 28 7f  cf b1 87 49 08 00 45 00   ·····M(· ···I··E·
0010  00 29 f7 ba 40 00 80 06  00 00 c0 a8 2b 02 36 ac   ·)··@··· ····+·6·
0020  49 62 c8 2f 01 bb 9b de  7e e1 d8 f0 00 2a 50 10   Ib·/···· ~····*P·
0030  02 00 6b d4 00 00 00                               ··k····

Wi-Fi: <live capture in progress>        Packets: 284 · Displayed: 284 (100.0%)        Profile: Default
```

Fig:5.3

Now that you've recorded some network data it's time to take a look at the captured packets. As shown in the screenshot above, the captured data interface contains three main sections: The packet list pane, the packet details pane, and the packet bytes pane. Packet List

The packet list pane, located at the top of the window, shows all packets found in the active capture file. Each packet has its own row and corresponding number assigned to it, along with each of these data points.

- **Time:** The timestamp of when the packet was captured is displayed in this column, with the default format being the number of seconds (or partial seconds) since this specific capture file was first created. To modify this format to something that may be a bit more useful, such as the actual time of day, select the *Time Display Format* option from Wireshark's *View* menu - located at the top of the main interface.

- **Source:** This column contains the address (IP or other) where the packet originated.
- **Destination:** This column contains the address that the packet is being sent to.
- **Protocol:** The packet's protocol name (i.e., TCP) can be found in this column.
- **Length:** The packet length, in bytes, is displayed in this column.
- **Info:** Additional details about the packet are presented here. The contents of this column can vary greatly depending on packet contents.

When a packet is selected in the top pane, you may notice one or more symbols appear in the first column. Open and/or closed brackets, as well as a straight horizontal line, can indicate whether or not a packet or group of packets are all part of the same back-and-forth conversation on the network. A broken horizontal line signifies that a packet is not part of said conversation.

Packets Details:

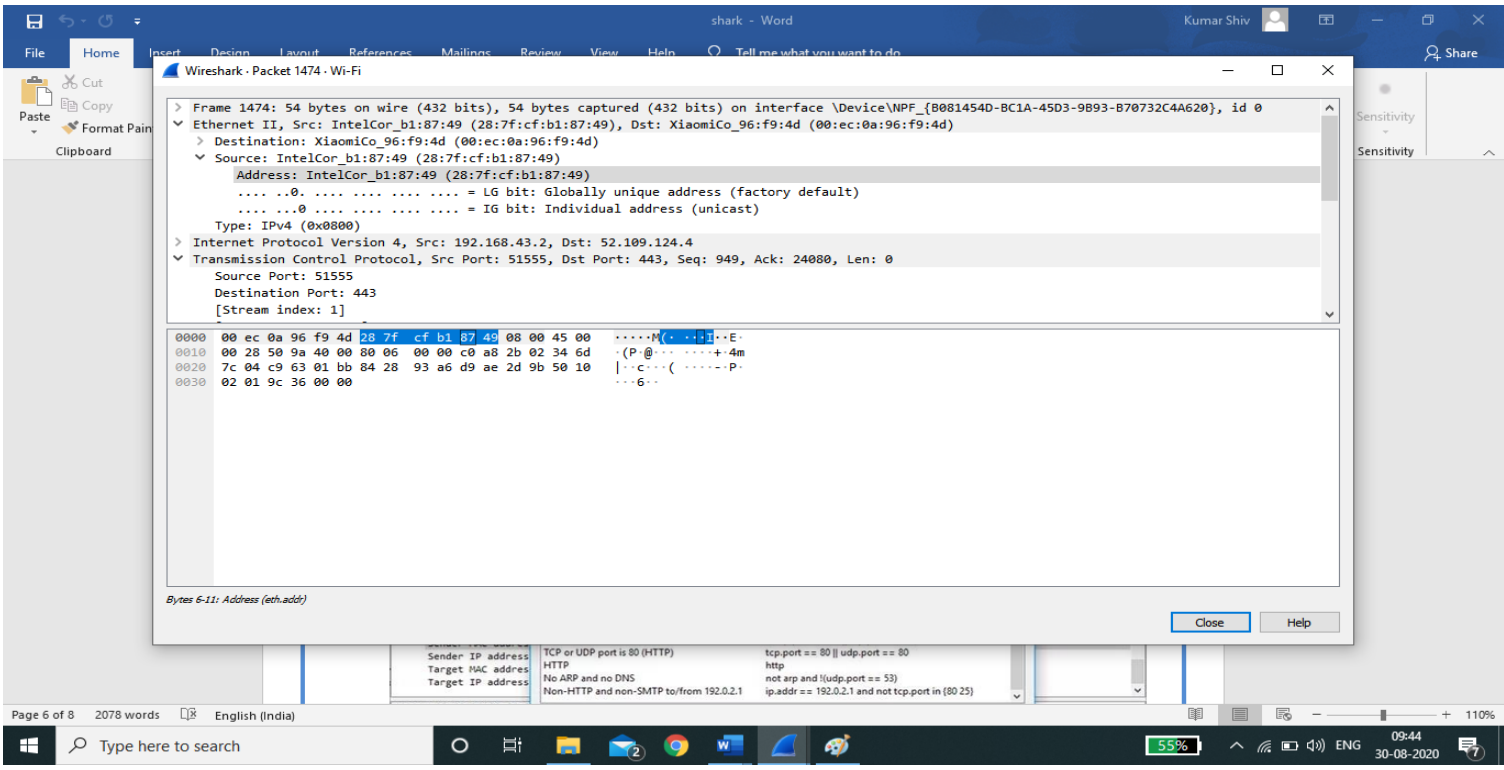
The details pane, found in the middle, presents the protocols and protocol fields of the selected packet in a collapsible format. In addition to expanding each selection, you can also apply individual Wireshark filters based on specific details as well as follow streams of data based on protocol type via the detail’s context menu – accessible by right-clicking your mouse on the desired item within this pane.

Packet Byte: At the bottom is the packet bytes pane, which displays the raw data of the selected packet in a hexadecimal view. This hex dump contains 16 hexadecimal bytes and 16 ASCII bytes alongside the data offset.

Selecting a specific portion of this data automatically highlights its corresponding section in the packet details pane and vice versa. Any bytes that cannot be printed are instead represented by a period. You can choose to show this data in bit format as opposed to hexadecimal by right-clicking anywhere within the pane and selecting the appropriate option from the context menu.

Fig:5.4

Fig:5.4



Wireshark · Packet 1474 · Wi-Fi

```
> Frame 1474: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B081454D-BC1A-45D3-9B93-B70732C4A620}, id 0
v Ethernet II, Src: IntelCor_b1:87:49 (28:7f:cf:b1:87:49), Dst: XiaomiCo_96:f9:4d (00:ec:0a:96:f9:4d)
  > Destination: XiaomiCo_96:f9:4d (00:ec:0a:96:f9:4d)
  v Source: IntelCor_b1:87:49 (28:7f:cf:b1:87:49)
      Address: IntelCor_b1:87:49 (28:7f:cf:b1:87:49)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.43.2, Dst: 52.109.124.4
v Transmission Control Protocol, Src Port: 51555, Dst Port: 443, Seq: 949, Ack: 24080, Len: 0
    Source Port: 51555
    Destination Port: 443
    [Stream index: 1]
```

```
0000  00 ec 0a 96 f9 4d 28 7f  cf b1 87 49 08 00 45 00   ·····M(· ···I··E·
0010  00 28 50 9a 40 00 80 06  00 00 c0 a8 2b 02 34 6d   ·(P·@··· ····+·4m
0020  7c 04 c9 63 01 bb 84 28  93 a6 d9 ae 2d 9b 50 10   |··c···( ····-·P·
0030  02 01 9c 36 00 00                                   ···6··
```

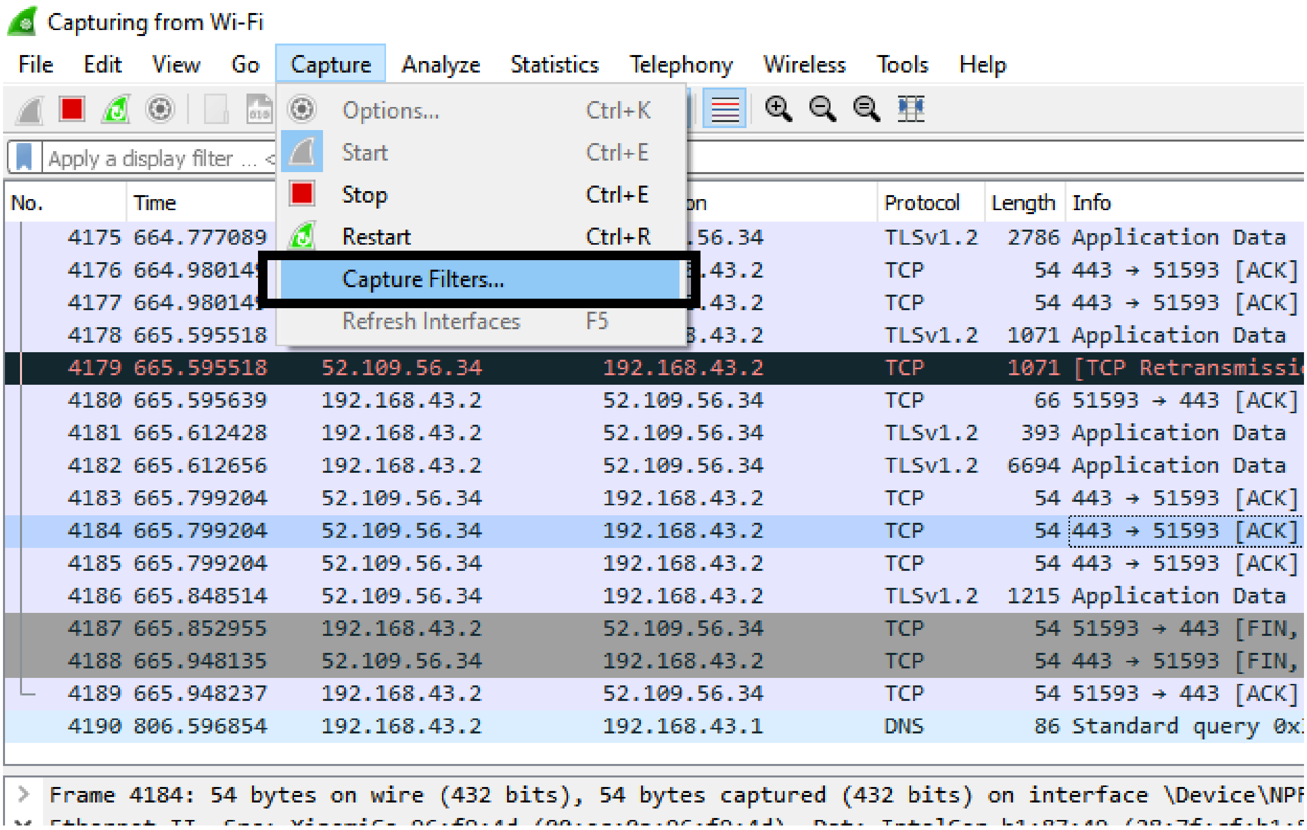
Bytes 6-11: Address (eth.addr)

Fig:5.4

Using Wireshark Filters

One of the most important feature sets in Wireshark is its filter capabilities, especially when you're dealing with files that are significant in size. Capture filters can be set before the fact, instructing Wireshark to only record those packets that meet your specified criteria.

Filters can also be applied to a capture file that has already been created so that only certain packets are shown. These are referred to as display filters.



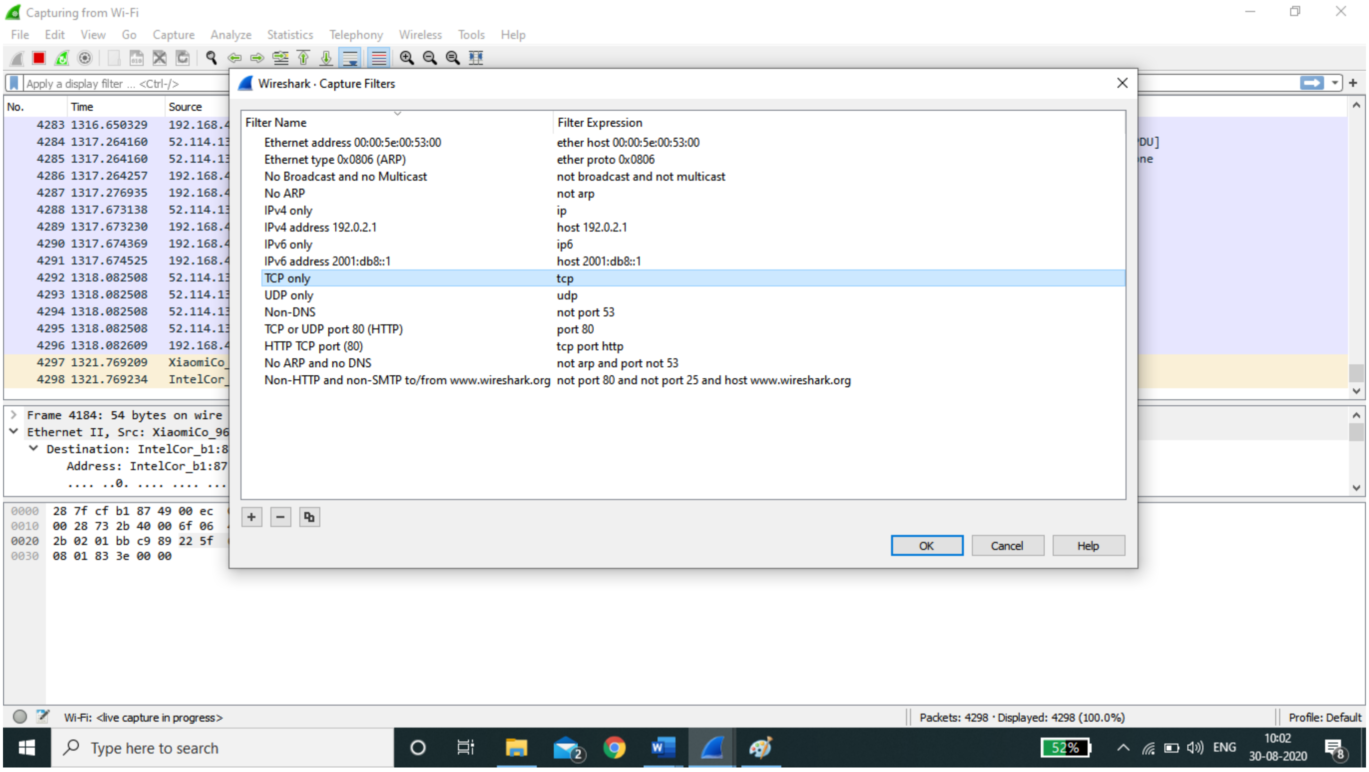
```
No.    Time                                     on             Protocol  Length  Info
  4175 664.777089                               .56.34         TLSv1.2     2786  Application Data
  4176 664.98014                                .43.2          TCP           54  443 → 51593 [ACK]
  4177 664.98014                                .43.2          TCP           54  443 → 51593 [ACK]
  4178 665.595518                               3.43.2         TLSv1.2     1071  Application Data
  4179 665.595518   52.109.56.34      192.168.43.2     TCP         1071  [TCP Retransmissi
  4180 665.595639   192.168.43.2      52.109.56.34     TCP           66  51593 → 443 [ACK]
  4181 665.612428   192.168.43.2      52.109.56.34     TLSv1.2      393  Application Data
  4182 665.612656   192.168.43.2      52.109.56.34     TLSv1.2     6694  Application Data
  4183 665.799204   52.109.56.34      192.168.43.2     TCP           54  443 → 51593 [ACK]
  4184 665.799204   52.109.56.34      192.168.43.2     TCP           54  443 → 51593 [ACK]
  4185 665.799204   52.109.56.34      192.168.43.2     TCP           54  443 → 51593 [ACK]
  4186 665.848514   52.109.56.34      192.168.43.2     TLSv1.2     1215  Application Data
  4187 665.852955   192.168.43.2      52.109.56.34     TCP           54  51593 → 443 [FIN,
  4188 665.948135   52.109.56.34      192.168.43.2     TCP           54  443 → 51593 [FIN,
  4189 665.948237   192.168.43.2      52.109.56.34     TCP           54  51593 → 443 [ACK]
  4190 806.596854   192.168.43.2      192.168.43.1     DNS           86  Standard query 0x

> Frame 4184: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF
```

Fig. 5.6

Wireshark provides a large number of predefined filters by default, letting you narrow down the number of visible packets with just a few keystrokes or mouse clicks. To use one of these existing filters, place its name in the *Apply a display filter* entry field (located directly below the Wireshark toolbar) or in the *Enter a capture filter* entry field (located in the centre of the welcome screen).

There are multiple ways to achieve this. If you already know the name of your filter, simply type it into the appropriate field. For example, if you only wanted to display TCP packets you would type *tcp*. Wireshark's autocompleting feature will show suggested names as you begin typing, making it easier to find the correct moniker for the filter you're seeking.

Another way to choose a filter is to click on the bookmark-like icon positioned on the left-hand side of the entry field. This will present a menu containing some of the most commonly-used filters as well as an option to *Manage Capture Filters* or *Manage Display Filters*. If you choose to manage either type an interface will appear allowing you to add, remove or edit filters. One can also access previously-used filters by selecting the down arrow, located on the right-hand side of the entry field, which displays a history drop-down list. Once set, capture filters will be applied as soon as you begin recording network traffic. To apply a display filter, however, you'll need to click on the right arrow button found on the far-right hand side of the entry field.


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wireshark · Capture Filters
+ − ⧉
OK Cancel Help
Wi-Fi: <live capture in progress>
Packets: 4298 · Displayed: 4298 (100.0%)
Profile: Default
Type here to search

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	not port 53
TCP or UDP port 80 (HTTP)	port 80
HTTP TCP port (80)	tcp port http
No ARP and no DNS	not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshark.org

Fig.5.7

Colouring Rules:

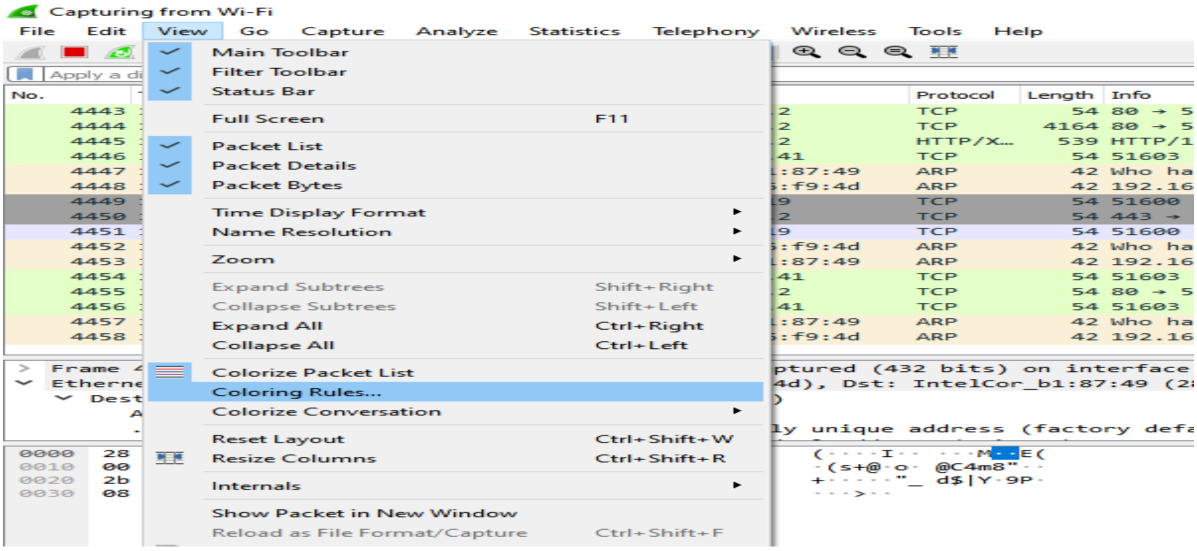
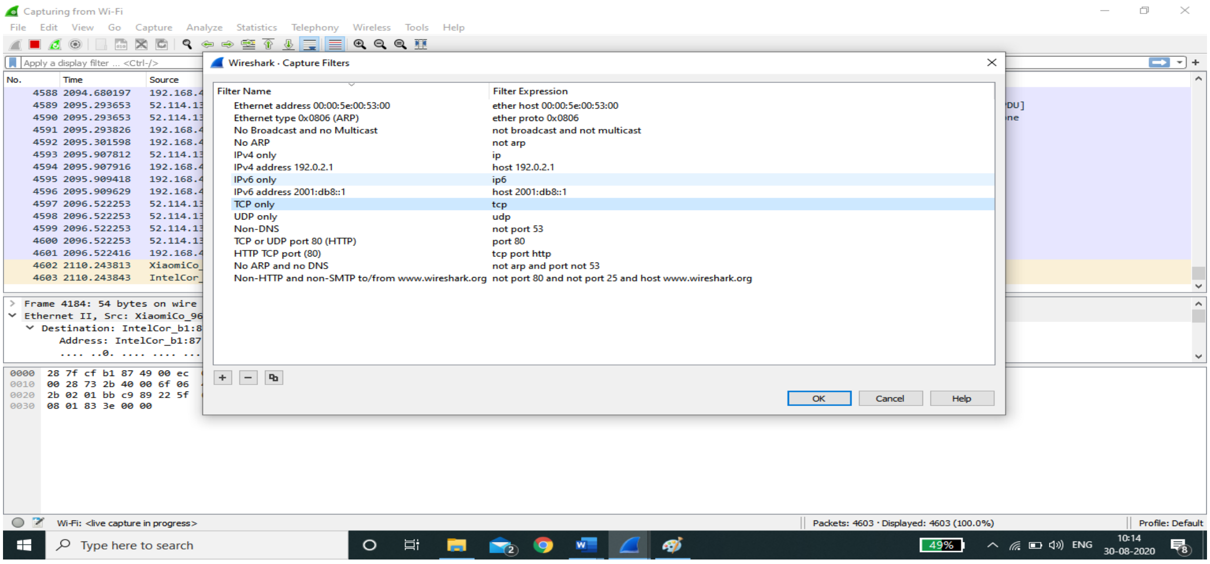


Fig. 5.8

While Wireshark's capture and display filters allow you to limit which packets are recorded or shown on the screen, its colorization functionality takes things a step further by making it easy to distinguish between different packet types based on their individual hue.

This handy feature lets you quickly locate certain packets within a saved set by their row's colour scheme in the packet list pane. Wireshark can colour packets based on rules that match particular fields in packets, to help the user identify the types of traffic at a glance.

Wireshark comes with about 20 default colouring rules built in; each which can be edited, disabled, or deleted if you wish. You can also add new shade-based filters through the colouring rules interface, accessible from the *View* menu. In addition to defining a name and filter criteria for each rule, you are also asked to associate both a background colour and a text colour. Packet colorization can be toggled off and on via the *Colorize Packet List* option, also found within the *View* menu.


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Capture Filters
+ −
OK
Cancel
Help
Wi-Fi: <live capture in progress>
Packets: 4603 · Displayed: 4603 (100.0%)
Profile: Default
Type here to search
49%
ENG
10:14
30-08-2020

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	not port 53
TCP or UDP port 80 (HTTP)	port 80
HTTP TCP port (80)	tcp port http
No ARP and no DNS	not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshark.org

Fig. 5.9

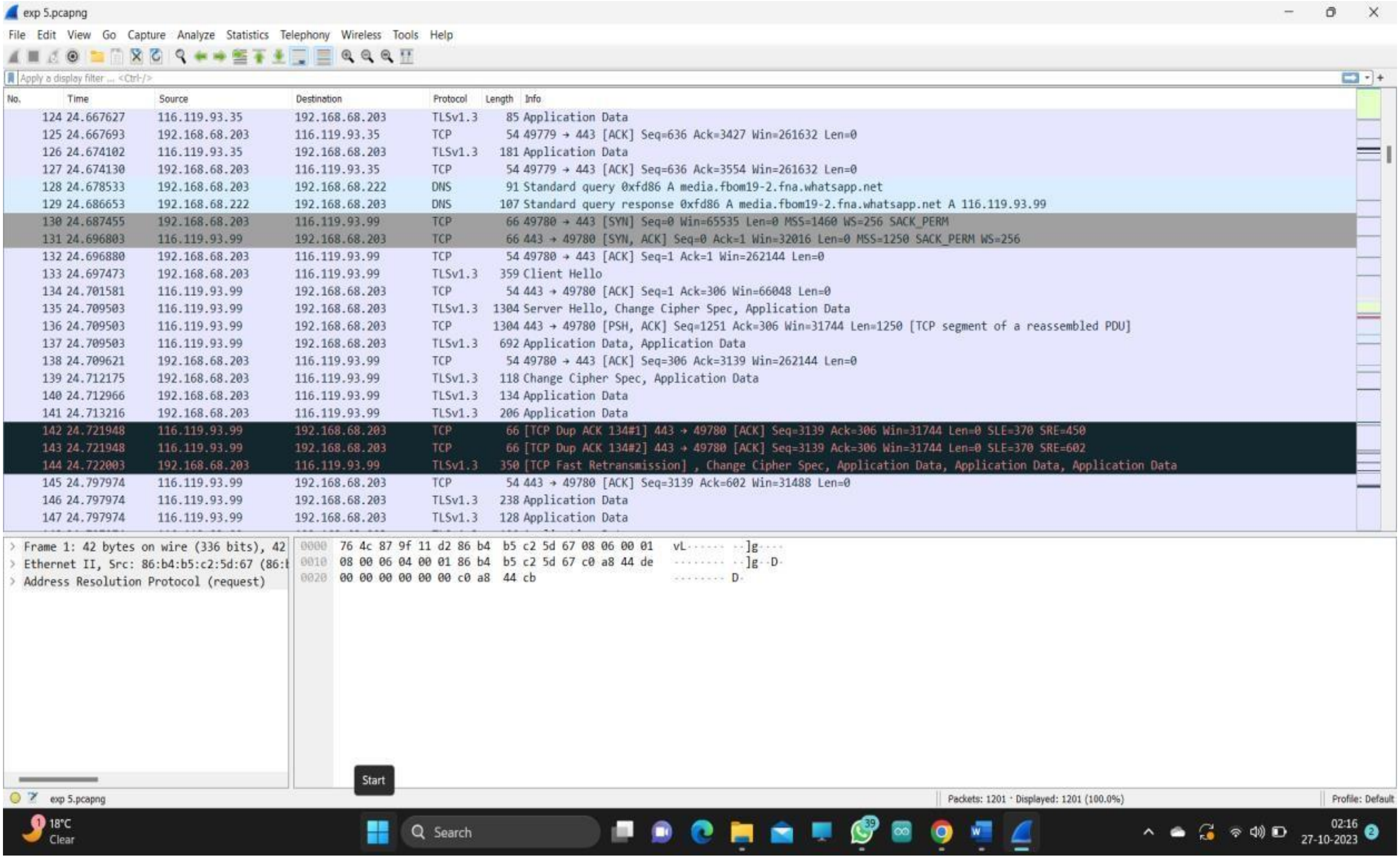
Simulation Packet Capture:

Wireshark can also be used to capture packets from most network simulation tools such as ns, OPNET Modeler, GNS3 and NetSim.

Conclusion:

Post Lab Questions:

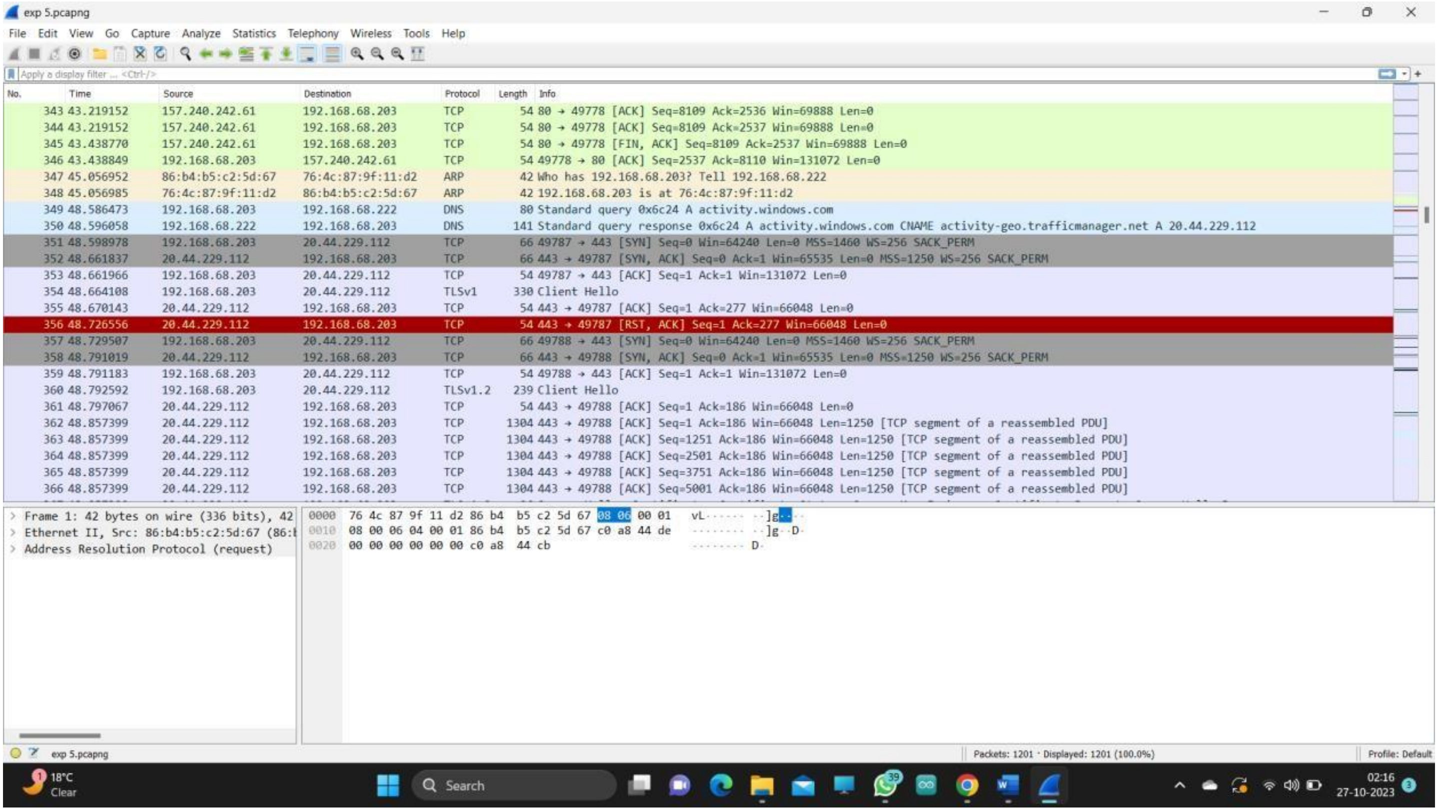
1. Enlist the other packet software tools you know.
2. What do we need to do packet capturing and its analysis?
3. If you capture TCP and UDP packets, what change you will observer in these packets.
4. What is the role of Filters in Wireshark?
5. How can one can use Wireshark software in day to day life and in Industry?


exp 5.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Start
exp 5.pcapng
Packets: 1201 · Displayed: 1201 (100.0%)
Profile: Default
18°C
Clear
Search
02:16
27-10-2023

```
No.   Time       Source          Destination     Protocol Length Info
  124 24.667627  116.119.93.35   192.168.68.203  TLSv1.3    85 Application Data
  125 24.667693  192.168.68.203  116.119.93.35   TCP        54 49779 → 443 [ACK] Seq=636 Ack=3427 Win=261632 Len=0
  126 24.674102  116.119.93.35   192.168.68.203  TLSv1.3   181 Application Data
  127 24.674130  192.168.68.203  116.119.93.35   TCP        54 49779 → 443 [ACK] Seq=636 Ack=3554 Win=261632 Len=0
  128 24.678533  192.168.68.203  192.168.68.222  DNS        91 Standard query 0xfd86 A media.fbom19-2.fna.whatsapp.net
  129 24.686653  192.168.68.222  192.168.68.203  DNS       107 Standard query response 0xfd86 A media.fbom19-2.fna.whatsapp.net A 116.119.93.99
  130 24.687455  192.168.68.203  116.119.93.99   TCP        66 49780 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
  131 24.696803  116.119.93.99   192.168.68.203  TCP        66 443 → 49780 [SYN, ACK] Seq=0 Ack=1 Win=32016 Len=0 MSS=1250 SACK_PERM WS=256
  132 24.696880  192.168.68.203  116.119.93.99   TCP        54 49780 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
  133 24.697473  192.168.68.203  116.119.93.99   TLSv1.3   359 Client Hello
  134 24.701581  116.119.93.99   192.168.68.203  TCP        54 443 → 49780 [ACK] Seq=1 Ack=306 Win=66048 Len=0
  135 24.709503  116.119.93.99   192.168.68.203  TLSv1.3  1304 Server Hello, Change Cipher Spec, Application Data
  136 24.709503  116.119.93.99   192.168.68.203  TCP      1304 443 → 49780 [PSH, ACK] Seq=1251 Ack=306 Win=31744 Len=1250 [TCP segment of a reassembled PDU]
  137 24.709503  116.119.93.99   192.168.68.203  TLSv1.3   692 Application Data, Application Data
  138 24.709621  192.168.68.203  116.119.93.99   TCP        54 49780 → 443 [ACK] Seq=306 Ack=3139 Win=262144 Len=0
  139 24.712175  192.168.68.203  116.119.93.99   TLSv1.3   118 Change Cipher Spec, Application Data
  140 24.712966  192.168.68.203  116.119.93.99   TLSv1.3   134 Application Data
  141 24.713216  192.168.68.203  116.119.93.99   TLSv1.3   206 Application Data
  142 24.721948  116.119.93.99   192.168.68.203  TCP        66 [TCP Dup ACK 134#1] 443 → 49780 [ACK] Seq=3139 Ack=306 Win=31744 Len=0 SLE=370 SRE=450
  143 24.721948  116.119.93.99   192.168.68.203  TCP        66 [TCP Dup ACK 134#2] 443 → 49780 [ACK] Seq=3139 Ack=306 Win=31744 Len=0 SLE=370 SRE=602
  144 24.722003  192.168.68.203  116.119.93.99   TLSv1.3   350 [TCP Fast Retransmission] , Change Cipher Spec, Application Data, Application Data, Application Data
  145 24.797974  116.119.93.99   192.168.68.203  TCP        54 443 → 49780 [ACK] Seq=3139 Ack=602 Win=31488 Len=0
  146 24.797974  116.119.93.99   192.168.68.203  TLSv1.3   238 Application Data
  147 24.797974  116.119.93.99   192.168.68.203  TLSv1.3   128 Application Data
```

```
> Frame 1: 42 bytes on wire (336 bits), 42
> Ethernet II, Src: 86:b4:b5:c2:5d:67 (86:t
> Address Resolution Protocol (request)
```

```
0000  76 4c 87 9f 11 d2 86 b4  b5 c2 5d 67 08 06 00 01   vL······ ··]g····
0010  08 00 06 04 00 01 86 b4  b5 c2 5d 67 c0 a8 44 de   ········ ··]g··D·
0020  00 00 00 00 00 00 c0 a8  44 cb                     ········ D·
```


exp 5.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
exp 5.pcapng
Packets: 1201 · Displayed: 1201 (100.0%)
Profile: Default
18°C
Clear
Search
02:16
27-10-2023

```
No.   Time       Source             Destination        Protocol Length Info
  343 43.219152  157.240.242.61     192.168.68.203     TCP        54 80 → 49778 [ACK] Seq=8109 Ack=2536 Win=69888 Len=0
  344 43.219152  157.240.242.61     192.168.68.203     TCP        54 80 → 49778 [ACK] Seq=8109 Ack=2537 Win=69888 Len=0
  345 43.438770  157.240.242.61     192.168.68.203     TCP        54 80 → 49778 [FIN, ACK] Seq=8109 Ack=2537 Win=69888 Len=0
  346 43.438849  192.168.68.203     157.240.242.61     TCP        54 49778 → 80 [ACK] Seq=2537 Ack=8110 Win=131072 Len=0
  347 45.056952  86:b4:b5:c2:5d:67  76:4c:87:9f:11:d2  ARP        42 Who has 192.168.68.203? Tell 192.168.68.222
  348 45.056985  76:4c:87:9f:11:d2  86:b4:b5:c2:5d:67  ARP        42 192.168.68.203 is at 76:4c:87:9f:11:d2
  349 48.586473  192.168.68.203     192.168.68.222     DNS        80 Standard query 0x6c24 A activity.windows.com
  350 48.596058  192.168.68.222     192.168.68.203     DNS       141 Standard query response 0x6c24 A activity.windows.com CNAME activity-geo.trafficmanager.net A 20.44.229.112
  351 48.598978  192.168.68.203     20.44.229.112      TCP        66 49787 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  352 48.661837  20.44.229.112      192.168.68.203     TCP        66 443 → 49787 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 WS=256 SACK_PERM
  353 48.661966  192.168.68.203     20.44.229.112      TCP        54 49787 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
  354 48.664108  192.168.68.203     20.44.229.112      TLSv1     330 Client Hello
  355 48.670143  20.44.229.112      192.168.68.203     TCP        54 443 → 49787 [ACK] Seq=1 Ack=277 Win=66048 Len=0
  356 48.726556  20.44.229.112      192.168.68.203     TCP        54 443 → 49787 [RST, ACK] Seq=1 Ack=277 Win=66048 Len=0
  357 48.729507  192.168.68.203     20.44.229.112      TCP        66 49788 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
  358 48.791019  20.44.229.112      192.168.68.203     TCP        66 443 → 49788 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 WS=256 SACK_PERM
  359 48.791183  192.168.68.203     20.44.229.112      TCP        54 49788 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
  360 48.792592  192.168.68.203     20.44.229.112      TLSv1.2   239 Client Hello
  361 48.797067  20.44.229.112      192.168.68.203     TCP        54 443 → 49788 [ACK] Seq=1 Ack=186 Win=66048 Len=0
  362 48.857399  20.44.229.112      192.168.68.203     TCP      1304 443 → 49788 [ACK] Seq=1 Ack=186 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
  363 48.857399  20.44.229.112      192.168.68.203     TCP      1304 443 → 49788 [ACK] Seq=1251 Ack=186 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
  364 48.857399  20.44.229.112      192.168.68.203     TCP      1304 443 → 49788 [ACK] Seq=2501 Ack=186 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
  365 48.857399  20.44.229.112      192.168.68.203     TCP      1304 443 → 49788 [ACK] Seq=3751 Ack=186 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
  366 48.857399  20.44.229.112      192.168.68.203     TCP      1304 443 → 49788 [ACK] Seq=5001 Ack=186 Win=66048 Len=1250 [TCP segment of a reassembled PDU]
```

```
> Frame 1: 42 bytes on wire (336 bits), 42
> Ethernet II, Src: 86:b4:b5:c2:5d:67 (86:t
> Address Resolution Protocol (request)
```

```
0000  76 4c 87 9f 11 d2 86 b4  b5 c2 5d 67 08 06 00 01   vL······ ··]g····
0010  08 00 06 04 00 01 86 b4  b5 c2 5d 67 c0 a8 44 de   ········ ··]g··D·
0020  00 00 00 00 00 00 c0 a8  44 cb                     ········ D·
```

Exp - 5

Name: Shreerang Mhatre
Rollno: 52
Batch: A3
class : TY
Date : 25/09/2023

* Post -Lab Question

Q1) Enlist other packet software tools you know

-> Packet software tools-

① tcp dump
② tshark
③ Ethereal
④ Packet Tracer
⑤ Colasoft capsa
⑥ NetMon
⑦ Snort
⑧ ngrep
⑨ ettercap
⑩ omni peek
⑪ PacketTotal

->

Q2) what do we need to do packet capturing and its analysis?

-> ~~A network interface~~
To perform packet Capturing & Analysis, we need to follow -

i) A network interface card that support bromisious mode, This allows the NIC to capture all trafic on the network even if it is not addressed to the Mc itself

ii) A packet capture tool, such as tcp dump or wire shark

iii) A computer to run the packet capture tool on.

Q3) If you capture ICP and UDP packets, what change your will observer in these packets.

-> ① when capturing both TCP & UDP packets, differences become apparent in various ascpects.

② TCP, aconnection - orinted protocol, ensures reliable and ordered data transfer with features like connection establishment, retransmission, and

③ UDP, a connection less protocol, lacks these features, making it faster but less reliable

④ Key differences include header information, reliability mechanisms, data transfer, efficiency etc.

⑤ TCP is commonly used for applications requiring data integrity, while UDP excels in real-time applications where low latency is crucial but data integrity is less critical.

Q4) What is the role of Filters in wireshark?

-> Filters in wire shark play a crucial role in packet analysis by allowing you to focus on specific packets that meet certain criteria while ignoring others. These filters help you efficiently search for an examine packets of interest, making the packet analysis process more manageable and insightful.

Q5) How can one use Wire shark software in day to day life and in Industry?

-> In Day-to-Day Life -

1. Network Troubleshooting
2. Security Awareness
3. Ophimizing Band width
4. Parental Controls
5. Home Lab & learning

In the Industry

1. Network Monitoring & Troubleshooting
2. security Analysis
3. compliance & Auditing
4. Quality of Service (QoS) Analysis
5. Performance Monitoring
6. Debugging Protocols & services
7. Capacity Planning
8. Training & Education