Name : Aditya Shashikant Raul
MIS : 142203016
Branch : Computer Engineering
Division : 2

==========

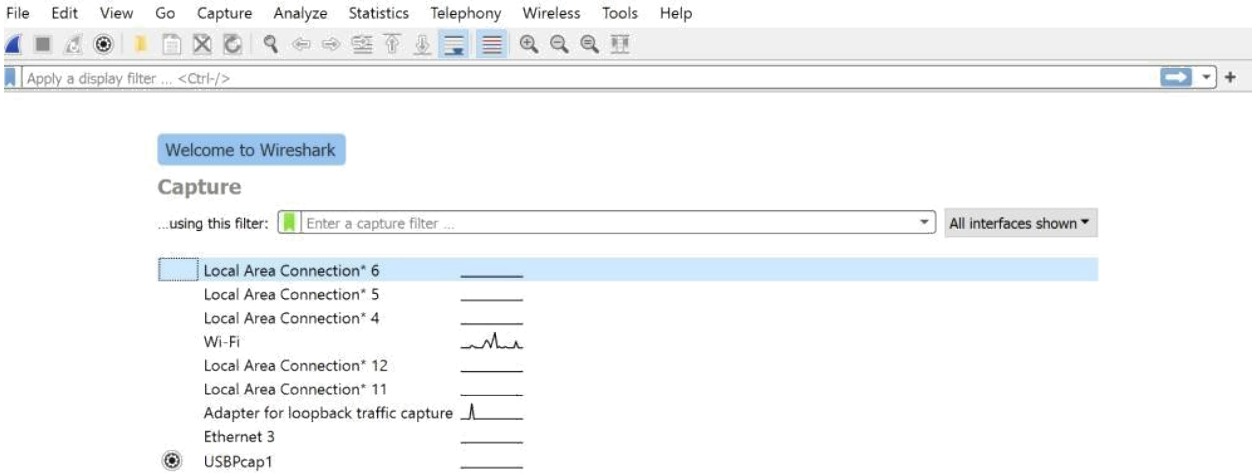
Analyze TCP, UDP and IPV4 Header using Wireshark

TCP Analysis using Wireshark:

TCP or Transmission Control Protocol is one of the most important protocols or standards for enabling communication possible amongst devices present over a particular network. It has algorithms that solve complex errors arising in packet communications, i.e. corrupted packets, invalid packets, duplicates, etc. Since it is used with IP(Internet Protocol), many times it is also referred to as TCP/IP. In order to start a communication, the TCP first establishes a connection using the three-way-handshake. TCP's efficiency over other protocols lies in its error detecting and correction attribute. Not only this, it organizes packets and segments larger data into a number of packets without disrupting the integrity of the data.

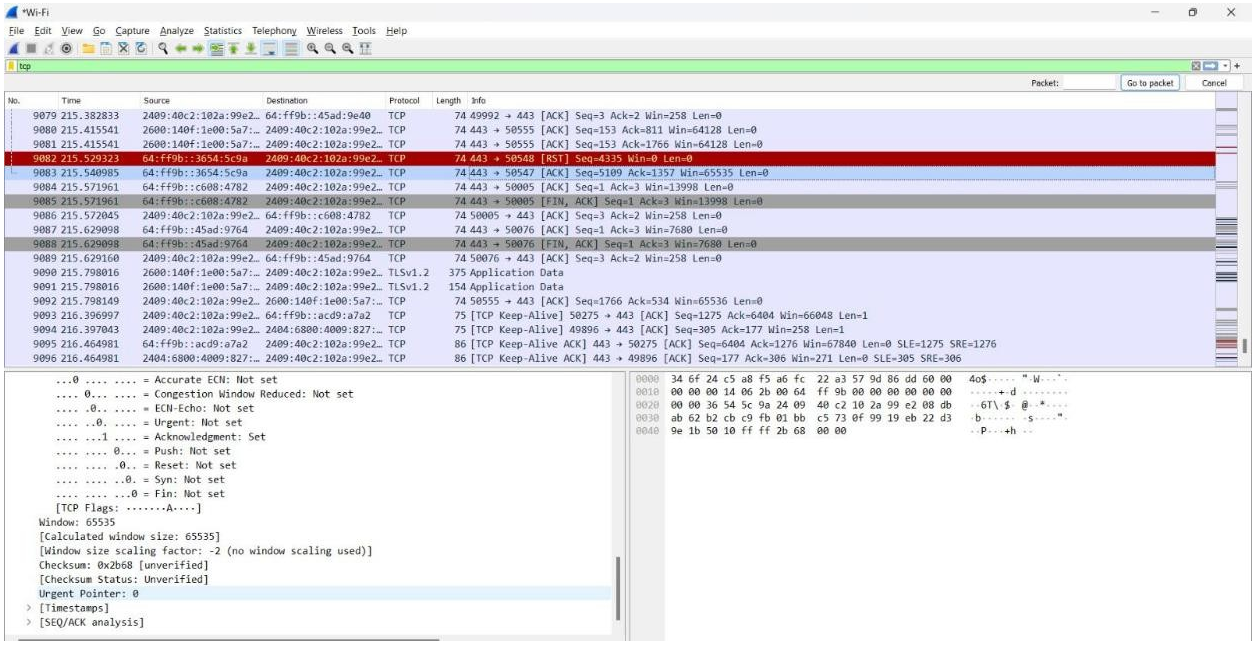
So now we are a bit familiar with TCP, let's look at how we can analyze TCP using Wireshark, which is the most widely used protocol analyzer in the world. In order to analyze TCP, you first need to launch Wireshark and follow the steps given below:

- From the menu bar, select capture -> options -> interfaces.
- In the interfaces, choose a particular Ethernet adapter and note down its IP, and click the start button of the selected adapter.
- Now we shall be capturing packets. Browse to a particular web address to generate traffic to capture packets from the communication for e.g. geeksforgeeks.org and return to Wireshark and stop the capture by selecting stop from the capture menu. You can have a look at it in the image below.

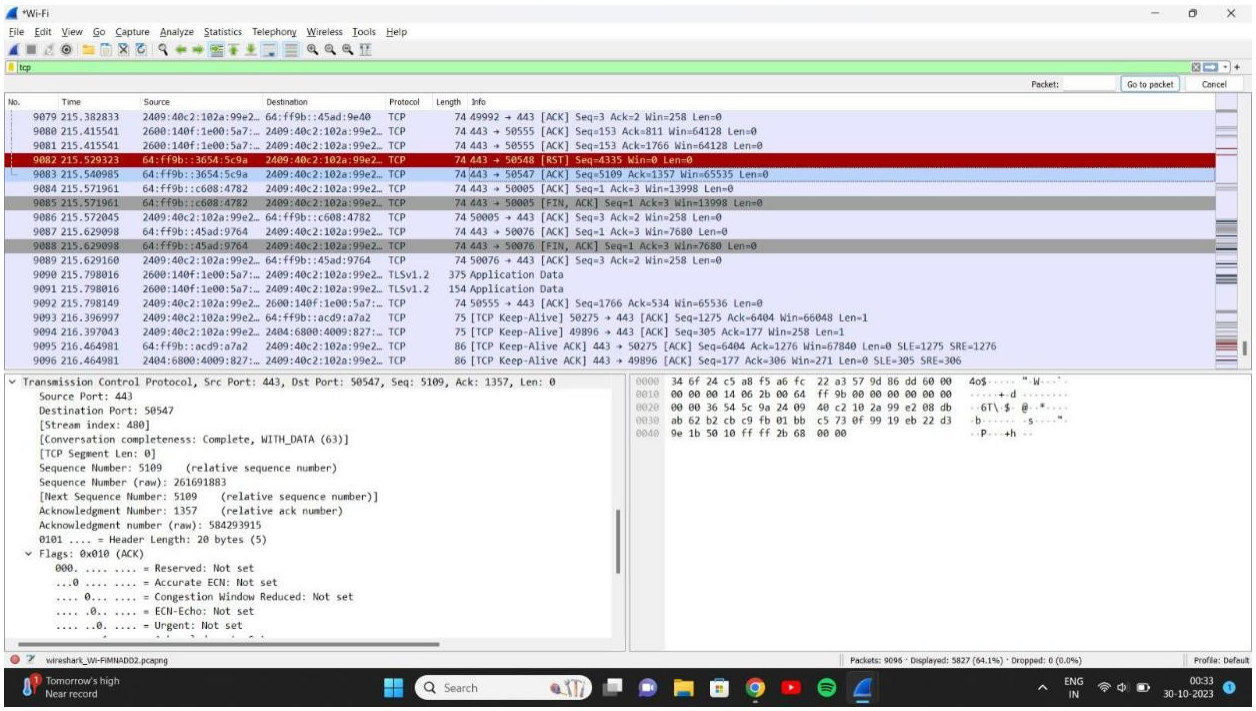


Now we have the captured packets and you will be having the captured packet list on the screen. Since we are concerned here with only TCP packets as we are doing TCP analysis, we shall be filtering out TCP packets from the packet pool. You can apply a filter in any of the following ways:

In the display filter bar on the screen, enter TCP and apply the filter.



From analyzing the menu in the menu bar select display filters or from capture select capture filters and then TCP only and ok.



Here you will have the list of TCP packets. The first three packets of this list are part of the three-way handshake mechanism of TCP to establish a connection. Let's get a basic knowledge of this mechanism which happens in the following 3 steps:

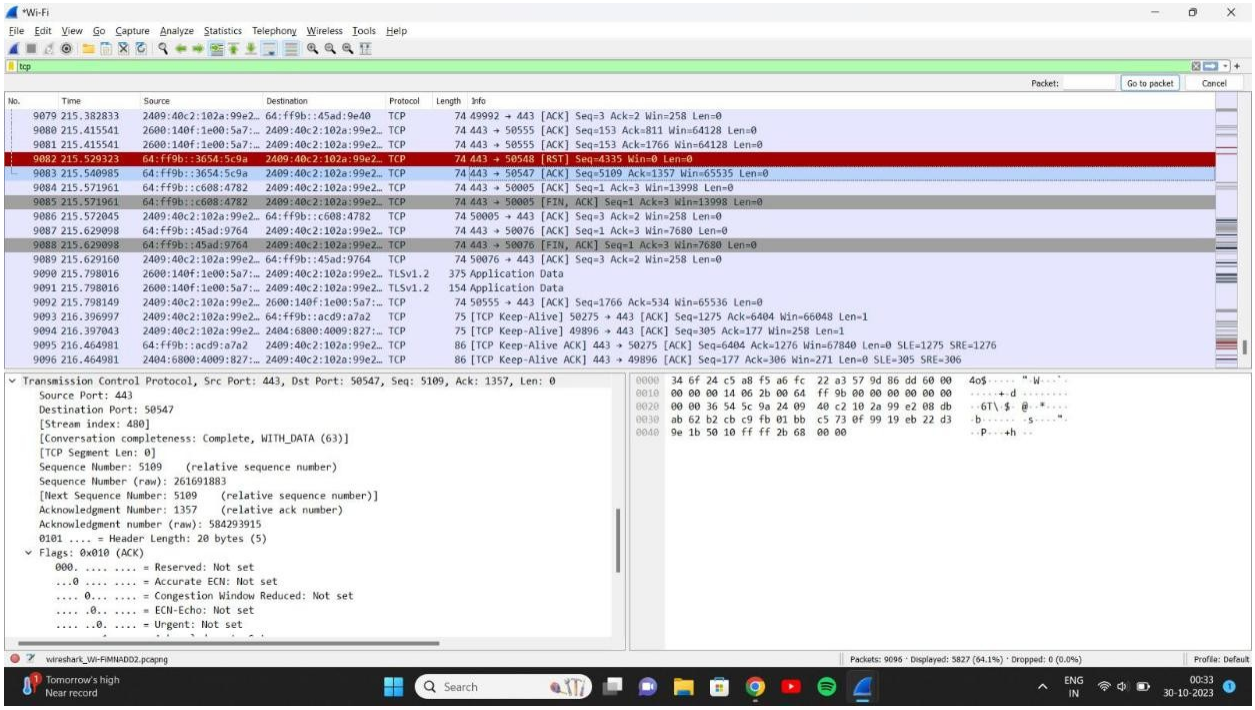
A synchronization packet (SYN) is sent by your local host IP to the server it desires to connect to.
The server reciprocates by sending an acknowledgment packet (ACK) to the local host signaling that it has received the SYN request of the host IP to connect and also sends a synchronization packet (SYN) to the local host to confirm the connection. So this one is basically an SYN+ACK packet.
The host answers this request by sending the ACK on receiving the SYN of the server.

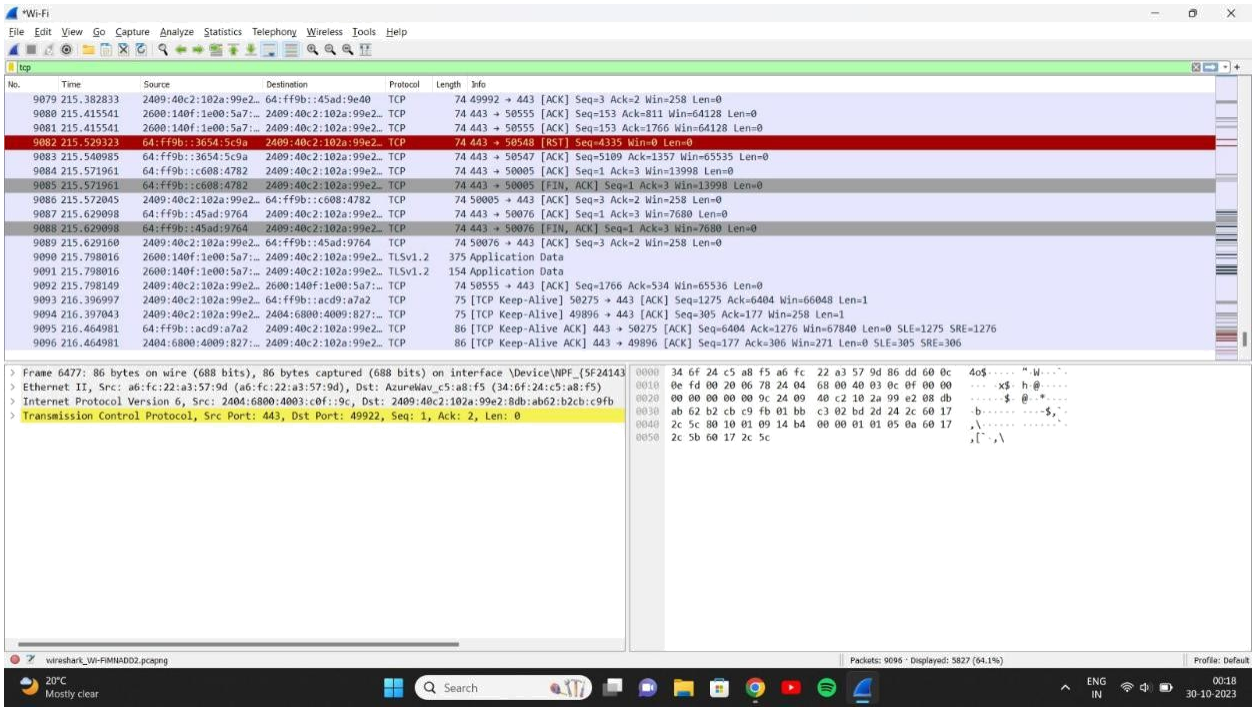
Source port: This is the port of your host network used for communication.
Destination port: This is the port of the destination server.
TCP segment length: It represents the data length in the selected packet.

Sequence number: It is a method used by Wireshark to give particular indexing to each packet for tracking packets with ease. This indexing starts from 0.

Next sequence number: It is the sum of the sequence number and the segment length of the current packet.
Acknowledgment number: It contains the byte length of data received.
Header length: It is the length of the TCP header and can vary from 20 to 60.

We can observe three connection establishment steps in the first three packets of the TCP list where each of the packet types i.e. ACK, SYN, SYN-ACK are listed on their respective sides. Now to examine a packet closely we shall select a packet and in the expert view in the packet detail section just below the packet list we shall be having the TCP parameters as you can see in the below diagram.



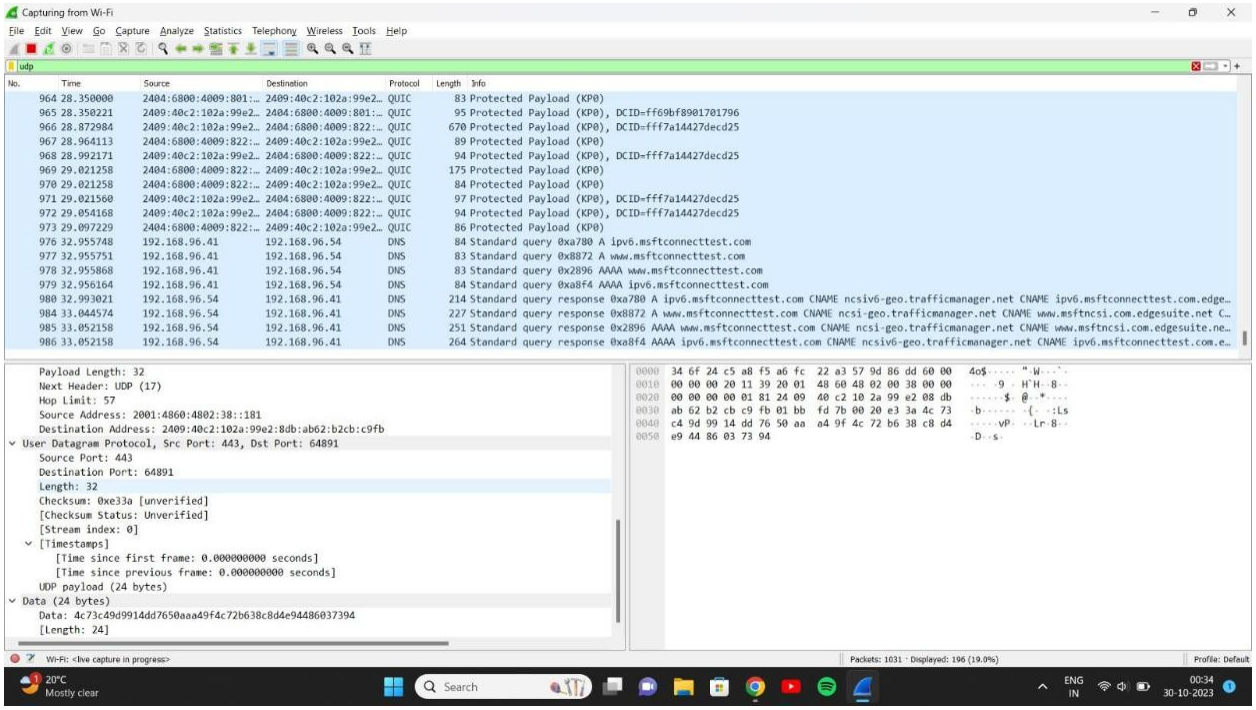


UDP Analysis using Wireshark:

The User Datagram Protocol, or UDP, is a communication protocol used across the Internet for especially time-sensitive transmissions such as video playback or DNS lookups. It speeds up communications by not formally establishing a connection before data is transferred. This allows data to be transferred very quickly, but it can also cause packets to become lost in transit — and create opportunities for exploitation in the form of DDoS attacks.

Like all networking protocols, UDP is a standardized method for transferring data between two computers in a network. Compared to other protocols, UDP accomplishes this process in a simple fashion: it sends packets (units of data transmission) directly to a target computer, without establishing a connection frst, indicating the order of said packets, or checking whether they arrived as intended. (UDP packets are referred to as ‘datagrams’.)

We have to follow the same steps for udp analysis .The only difference is we are fltering the packets based on udp protocol.



UDP Header –

UDP header is an 8-bytes fxed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The frst 8 Bytes contains all necessary header information and the remaining part consist of data. UDP port number felds are each 16 bits long, therefore the range for port numbers is defned from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.

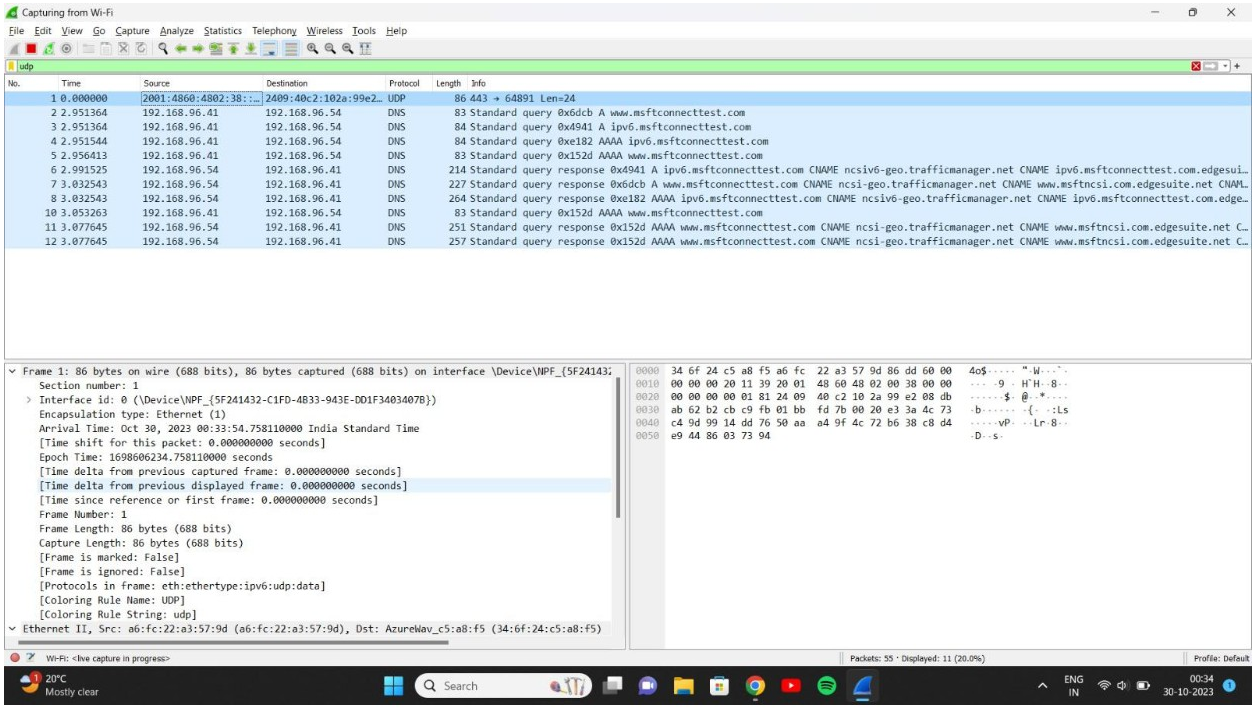
Source Port: Source Port is a 2 Byte long feld used to identify the port number of the source.

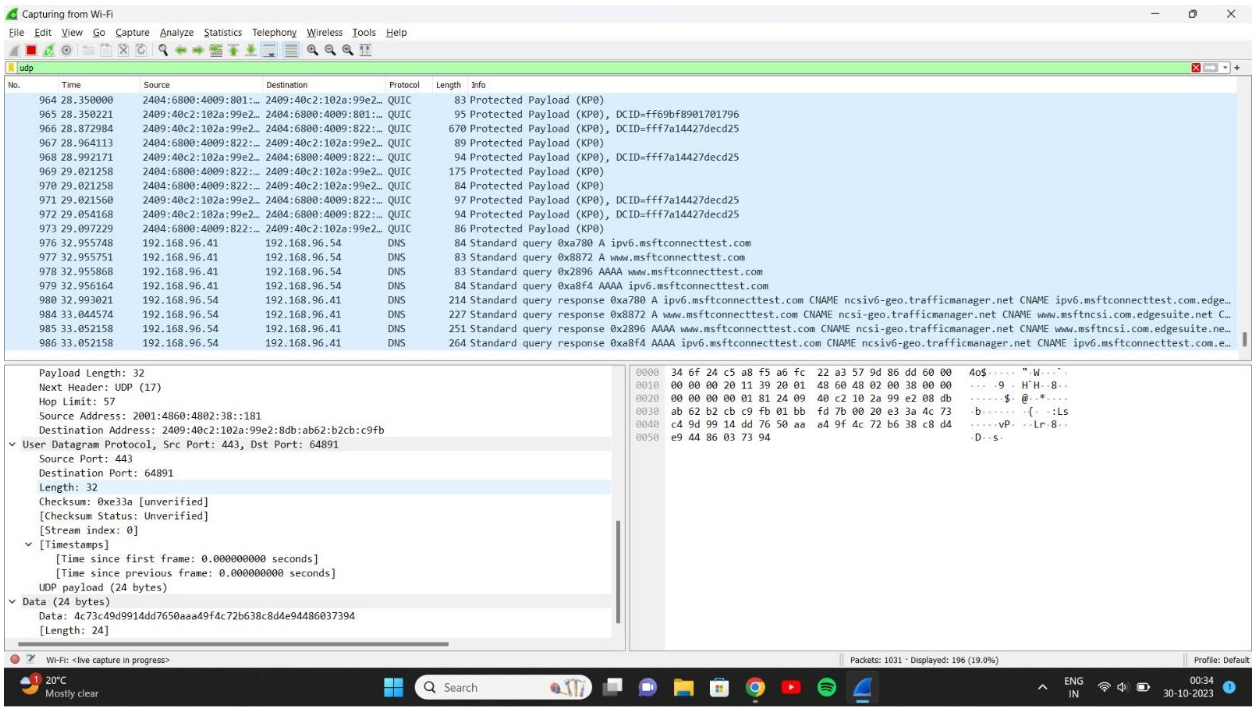
Destination Port: It is a 2 Byte long feld, used to identify the port of the destined packet.

Length: Length is the length of UDP including the header and the data. It is a 16-bits feld.

Checksum: Checksum is 2 Bytes long feld. It is the 16-bit one’s complement of the one’s complement sum of the UDP header, the pseudo-header of information

from the IP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.





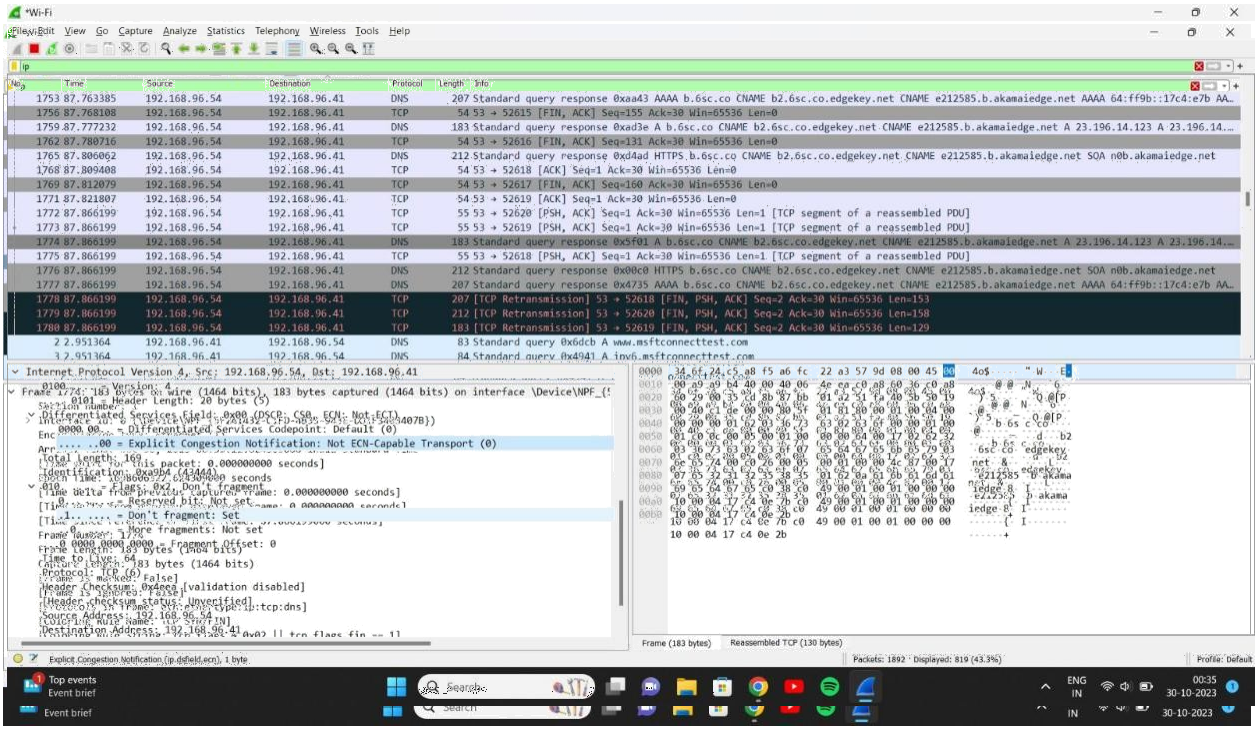
IPv4 Analysis using Wireshark:

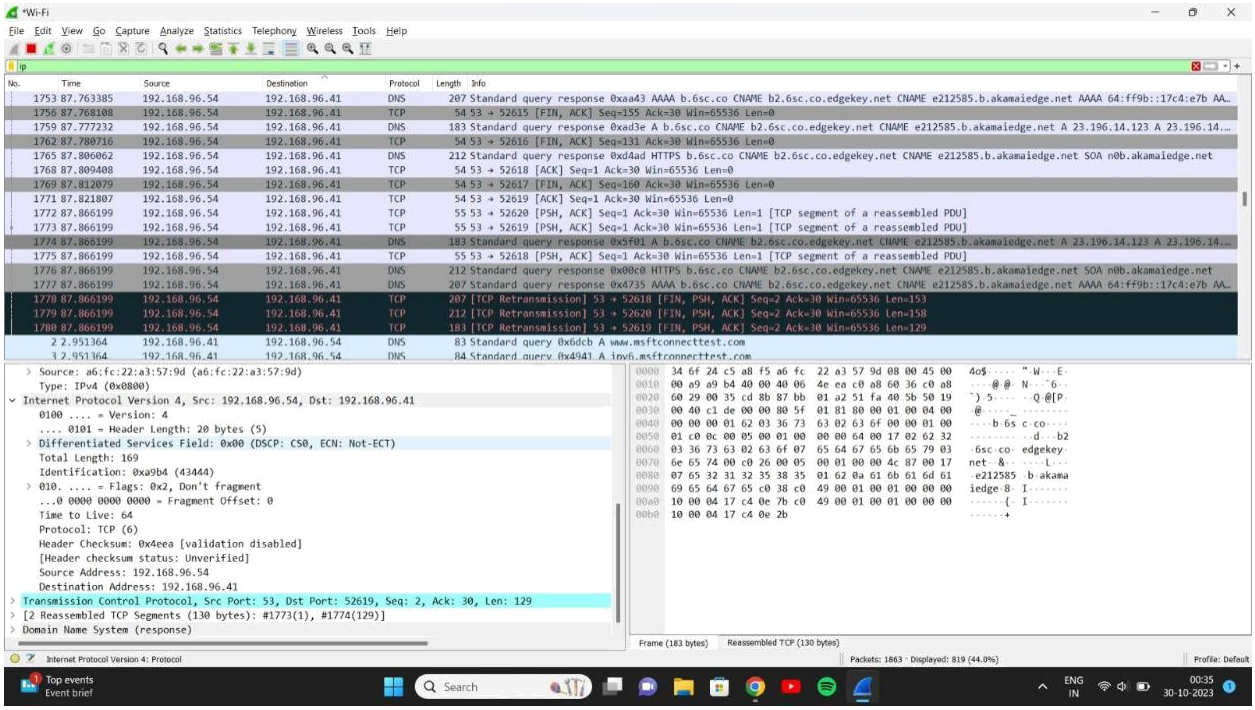
IP stands for Internet Protocol and v4 stands for Version Four (IPv4). IPv4 was the primary version brought into action for production within the ARPANET in 1983. IP version four addresses are 32-bit integers which will be expressed in decimal notation.
Example- 192.0.2.126 could be an IPv4 address.

Parts of IPv4
Network part:
The network part indicates the distinctive variety that's appointed to the network. The network part conjointly identifes the category of the network that's assigned.
Host Part:
The host part uniquely identifes the machine on your network. This part of the IPv4 address is assigned to every host.
For each host on the network, the network part is the same, however, the host half must vary.
Subnet number:
This is the nonobligatory part of IPv4. Local networks that have massive numbers of hosts are divided into subnets and subnet numbers are appointed to that.

Characteristics of IPv4

- IPv4 could be a 32-Bit IP Address.
- IPv4 could be a numeric address, and its bits are separated by a dot.
- The number of header felds is twelve and the length of the header feld is twenty.
- It has Unicast, broadcast, and multicast style of addresses.
- IPv4 supports VLSM (Virtual Length Subnet Mask).
- IPv4 uses the Post Address Resolution Protocol to map to the MAC address.
- RIP may be a routing protocol supported by the routed daemon.
- Networks ought to be designed either manually or with DHCP.
- Packet fragmentation permits from routers and causing host.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip
No. Time Source Destination Protocol Length Info
1753 87.763385 192.168.96.54 192.168.96.41 DNS 207 Standard query response 0xaa43 AAAA b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net AAAA 64:ff9b::17c4:e7b AA…
1756 87.768108 192.168.96.54 192.168.96.41 TCP 54 53 → 52615 [FIN, ACK] Seq=155 Ack=30 Win=65536 Len=0
1759 87.777232 192.168.96.54 192.168.96.41 DNS 183 Standard query response 0xad3e A b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net A 23.196.14.123 A 23.196.14.…
1762 87.780716 192.168.96.54 192.168.96.41 TCP 54 53 → 52616 [FIN, ACK] Seq=131 Ack=30 Win=65536 Len=0
1765 87.806062 192.168.96.54 192.168.96.41 DNS 212 Standard query response 0xd4ad HTTPS b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net SOA n0b.akamaiedge.net
1768 87.809408 192.168.96.54 192.168.96.41 TCP 54 53 → 52618 [ACK] Seq=1 Ack=30 Win=65536 Len=0
1769 87.812079 192.168.96.54 192.168.96.41 TCP 54 53 → 52617 [FIN, ACK] Seq=160 Ack=30 Win=65536 Len=0
1771 87.821807 192.168.96.54 192.168.96.41 TCP 54 53 → 52619 [ACK] Seq=1 Ack=30 Win=65536 Len=0
1772 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52620 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1773 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52619 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1774 87.866199 192.168.96.54 192.168.96.41 DNS 183 Standard query response 0x5f01 A b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net A 23.196.14.123 A 23.196.14.…
1775 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52618 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1776 87.866199 192.168.96.54 192.168.96.41 DNS 212 Standard query response 0x00c0 HTTPS b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net SOA n0b.akamaiedge.net
1777 87.866199 192.168.96.54 192.168.96.41 DNS 207 Standard query response 0x4735 AAAA b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net AAAA 64:ff9b::17c4:e7b AA…
1778 87.866199 192.168.96.54 192.168.96.41 TCP 207 [TCP Retransmission] 53 → 52618 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=153
1779 87.866199 192.168.96.54 192.168.96.41 TCP 212 [TCP Retransmission] 53 → 52620 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=158
1780 87.866199 192.168.96.54 192.168.96.41 TCP 183 [TCP Retransmission] 53 → 52619 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=129
2 2.951364 192.168.96.41 192.168.96.54 DNS 83 Standard query 0x6dcb A www.msftconnecttest.com
3 2.951364 192.168.96.41 192.168.96.54 DNS 84 Standard query 0x4941 A ipv6.msftconnecttest.com
Internet Protocol Version 4, Src: 192.168.96.54, Dst: 192.168.96.41
Frame 1774: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface \Device\NPF_{
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 169
Identification: 0xa9b4 (43444)
010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x4eea [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.96.54
Destination Address: 192.168.96.41
Explicit Congestion Notification (ip.dsfield.ecn), 1 byte
Frame (183 bytes)
Reassembled TCP (130 bytes)
Packets: 1892 · Displayed: 819 (43.3%)
Profile: Default
Top events
Event brief
Search
ENG
IN
00:35
30-10-2023


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip
No. Time Source Destination Protocol Length Info
1753 87.763385 192.168.96.54 192.168.96.41 DNS 207 Standard query response 0xaa43 AAAA b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net AAAA 64:ff9b::17c4:e7b AA…
1756 87.768108 192.168.96.54 192.168.96.41 TCP 54 53 → 52615 [FIN, ACK] Seq=155 Ack=30 Win=65536 Len=0
1759 87.777232 192.168.96.54 192.168.96.41 DNS 183 Standard query response 0xad3e A b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net A 23.196.14.123 A 23.196.14.…
1762 87.780716 192.168.96.54 192.168.96.41 TCP 54 53 → 52616 [FIN, ACK] Seq=131 Ack=30 Win=65536 Len=0
1765 87.806062 192.168.96.54 192.168.96.41 DNS 212 Standard query response 0xd4ad HTTPS b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net SOA n0b.akamaiedge.net
1768 87.809408 192.168.96.54 192.168.96.41 TCP 54 53 → 52618 [ACK] Seq=1 Ack=30 Win=65536 Len=0
1769 87.812079 192.168.96.54 192.168.96.41 TCP 54 53 → 52617 [FIN, ACK] Seq=160 Ack=30 Win=65536 Len=0
1771 87.821807 192.168.96.54 192.168.96.41 TCP 54 53 → 52619 [ACK] Seq=1 Ack=30 Win=65536 Len=0
1772 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52620 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1773 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52619 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1774 87.866199 192.168.96.54 192.168.96.41 DNS 183 Standard query response 0x5f01 A b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net A 23.196.14.123 A 23.196.14.…
1775 87.866199 192.168.96.54 192.168.96.41 TCP 55 53 → 52618 [PSH, ACK] Seq=1 Ack=30 Win=65536 Len=1 [TCP segment of a reassembled PDU]
1776 87.866199 192.168.96.54 192.168.96.41 DNS 212 Standard query response 0x00c0 HTTPS b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net SOA n0b.akamaiedge.net
1777 87.866199 192.168.96.54 192.168.96.41 DNS 207 Standard query response 0x4735 AAAA b.6sc.co CNAME b2.6sc.co.edgekey.net CNAME e212585.b.akamaiedge.net AAAA 64:ff9b::17c4:e7b AA…
1778 87.866199 192.168.96.54 192.168.96.41 TCP 207 [TCP Retransmission] 53 → 52618 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=153
1779 87.866199 192.168.96.54 192.168.96.41 TCP 212 [TCP Retransmission] 53 → 52620 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=158
1780 87.866199 192.168.96.54 192.168.96.41 TCP 183 [TCP Retransmission] 53 → 52619 [FIN, PSH, ACK] Seq=2 Ack=30 Win=65536 Len=129
2 2.951364 192.168.96.41 192.168.96.54 DNS 83 Standard query 0x6dcb A www.msftconnecttest.com
3 2.951364 192.168.96.41 192.168.96.54 DNS 84 Standard query 0x4941 A ipv6.msftconnecttest.com
Source: a6:fc:22:a3:57:9d (a6:fc:22:a3:57:9d)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.96.54, Dst: 192.168.96.41
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 169
Identification: 0xa9b4 (43444)
010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x4eea [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.96.54
Destination Address: 192.168.96.41
Transmission Control Protocol, Src Port: 53, Dst Port: 52619, Seq: 2, Ack: 30, Len: 129
[2 Reassembled TCP Segments (130 bytes): #1773(1), #1774(129)]
Domain Name System (response)
Internet Protocol Version 4: Protocol
Frame (183 bytes)
Reassembled TCP (130 bytes)
Packets: 1863 · Displayed: 819 (44.0%)
Profile: Default
Top events
Event brief
Search
ENG
IN
00:35
30-10-2023