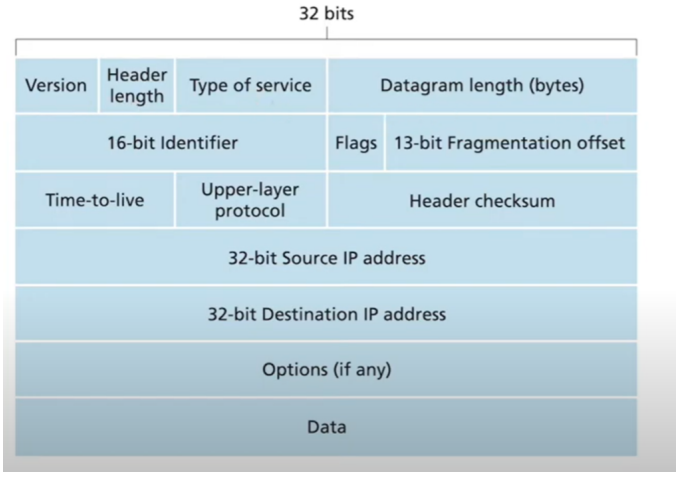
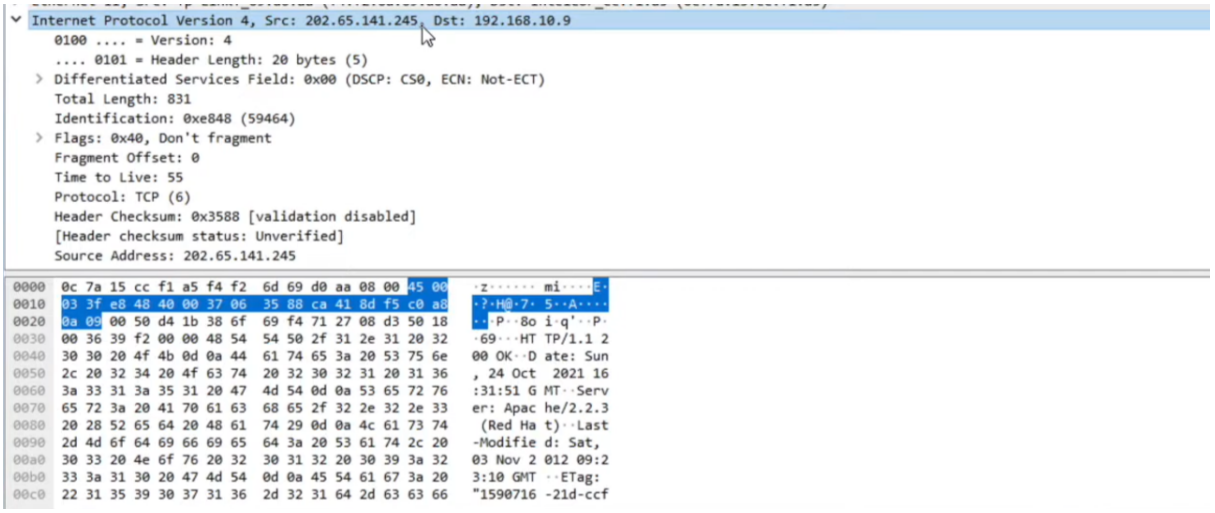
IP Packet capturing using wireshark

- To generate IP packets, we will use HTTPS(using browser)
- We will type url over the browser so that it will generate HTTP traffic which in turn will generate TCP and IP
- Ethernet or WiFi will capture the packet from the datalink layer, this will be the main component of the frame that will be captured by the wireshark.

Contents of the IP packet



Packet(wach row is 4 bytes, and minimum 5 rows, so minimum 20 bytes)



Field name	Field leangth(# of bits)	F ield value(content carried)
Version	4 bits	0100 ....
Header Length	4 bits	0101, 5
Type of Services	8 bits	0x00
Datagram Length	16 bits	0x033f
16-bit identifier	16 bits	010
Flags	3 bits	000
13-bt fragmentation offset	13 bits	0
Time to live	8 bits	55
Upper Layer Protocol	8 bits	6 = TCP
Header Checksum	16 bits	0x3588
32-bit Source IP address	32 bits	202.65.141.245
32-bit destination IP address	32 bits	192.168.10.9
Option(if any)	_	_
Data	831 bytes	TCP data

IP Header Values and Calculation

1)Version (4 bits)

Calculation: Set to 4 for IPv4 or 6 for IPv6. This field is not calculated but rather set based on the IP version being used.

2)IHL (Internet Header Length) (4 bits)

Calculation: Calculated as the number of 32-bit words in the IP header. For a standard IPv4 header without options, it is 5 (20 bytes). If options are present, this value increases to reflect the total header length.

3)Type of Service (ToS) (8 bits)

Calculation: Set based on desired QoS (Quality of Service) parameters. Typically, this is set to 0 for default service, but it can include fields for priority, delay, throughput, and reliability.

4)Total Length (16 bits)

Calculation: This is the length of the entire IP packet, including the header and the data payload, in bytes. It is calculated as the sum of the header length (IHL field) and the data length.

5)Identification (16 bits)

Calculation: A unique value assigned to each packet. It helps in reassembling fragmented packets. This value is typically assigned by the sender and incremented for each new packet.

6)Flags (3 bits)

Calculation: Set to control fragmentation. Common values include:

- 0x2 for "Don't Fragment" (DF)
- 0x1 for "More Fragments" (MF)

These flags are set based on whether the packet is fragmented or if

fragmentation is allowed.

7)Fragment Offset (13 bits)

Calculation: Indicates the offset of the fragment relative to the start of the original packet. It is calculated as the position of the fragment in the original packet, measured in 8-byte units

8)Time to Live (TTL) (8 bits)

Calculation: Set by the sender and decremented by each router along the path. It prevents packets from circulating indefinitely. Typical starting values are 64, 128, or 255.

9)Protocol (8 bits)

Calculation: Indicates the type of protocol in the data portion of the IP packet (e.g., 6 for TCP, 17 for UDP). This value is set based on the higher-layer protocol being used.

10)Header Checksum (16 bits)

Calculation: Computed by taking the one's complement sum of all 16-bit words in the header. The checksum helps detect errors in the header. To calculate:

-Set all checksum bits to 0 and sum all 16-bit header words.
-Add any overflow bits to the result.
-Take the one's complement of the result.
-This value is placed in the checksum field.

11)Source IP Address (32 bits)

Calculation: Set to the IP address of the sender. This is manually assigned or obtained dynamically.

12)Destination IP Address (32 bits)

Calculation: Set to the IP address of the intended recipient. Like the source address, it is manually assigned or obtained.

13)Options (variable)

Calculation: Optional fields that can be used for various functions. The length and content depend on specific requirements or network protocols.

14)Data