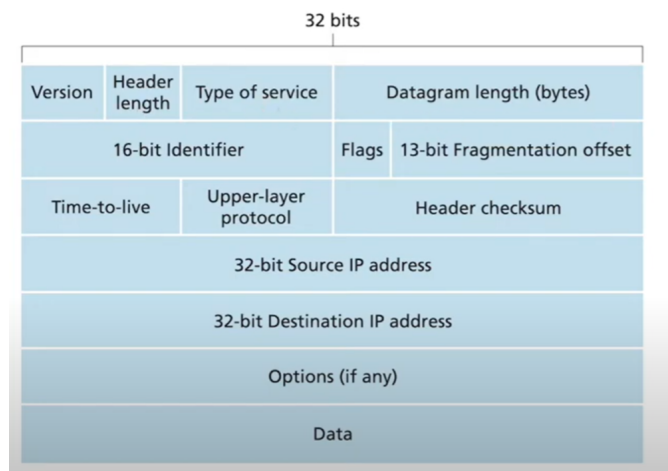


IP Packet capturing using wireshark

- 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)

```

Internet Protocol Version 4, Src: 202.65.141.245, Dst: 192.168.10.9
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 831
    Identification: 0xe848 (59464)
  > Flags: 0x40, Don't fragment
    Fragment Offset: 0
    Time to Live: 55
    Protocol: TCP (6)
    Header Checksum: 0x3588 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 202.65.141.245

0000 0c 7a 15 cc f1 a5 f4 f2 6d 69 d0 aa 08 00 45 00  .z.....mI...E
0010 03 3f e8 40 40 00 37 06 35 88 ca 41 8d f5 c0 a8  .?H0-7-5-A...
0020 0a 05 00 50 d4 1b 38 6f 69 f4 71 27 08 d3 50 18  .P..8o i q'..P
0030 00 36 39 f2 00 00 48 54 54 50 2f 31 2e 31 20 32  .69...HT TP/1.1 2
0040 30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 53 75 6e  00 OK..D ate: Sun
0050 2c 20 32 34 20 4f 63 74 20 32 30 32 31 20 31 36  , 24 Oct 2021 16
0060 3a 33 31 3a 35 31 20 47 4d 54 0d 0a 53 65 72 76  :31:51 G HT..Serv
0070 65 72 3a 20 41 70 61 63 68 65 2f 32 2e 32 2e 33  er: Apac he/2.2.3
0080 20 28 52 65 64 20 48 61 74 29 0d 0a 4c 61 73 74  (Red Ha t)..Last
0090 2d 4d 6f 64 69 66 69 65 64 3a 20 53 61 74 2c 20  -Modifie d: Sat,
00a0 30 33 20 4e 6f 76 20 32 30 31 32 20 30 39 3a 32  03 Nov 2 012 09:2
00b0 33 3a 31 30 20 47 4d 54 0d 0a 45 54 61 67 3a 20  3:10 GMT ..ETag:
00c0 22 31 35 39 30 37 31 36 2d 32 31 64 2d 63 63 66  "1590716 -21d-ccf

```

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