Encapsulation

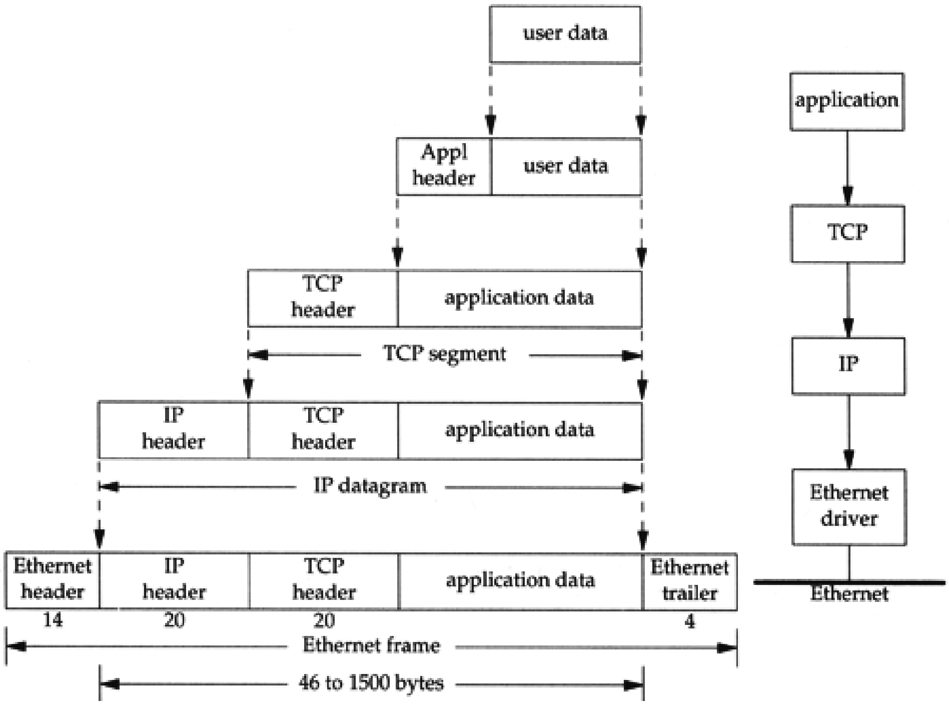


Figure . Encapsulation of data as it goes down the protocol stack.

Source: https://flylib.com/books/en/3.223.1.18/1/

What are Ethernet, IP and TCP Headers

1. Ethernet II – Layer 2
2. IP Header – Layer 3
3. TCP Header -Layer 4. I left out UDP since connectionless headers are quite simpler, e.g. Source Port, Destination Port, Length and Checksum.

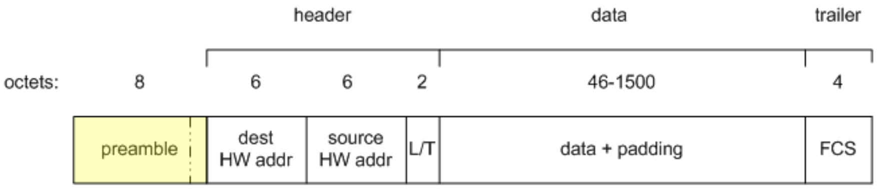
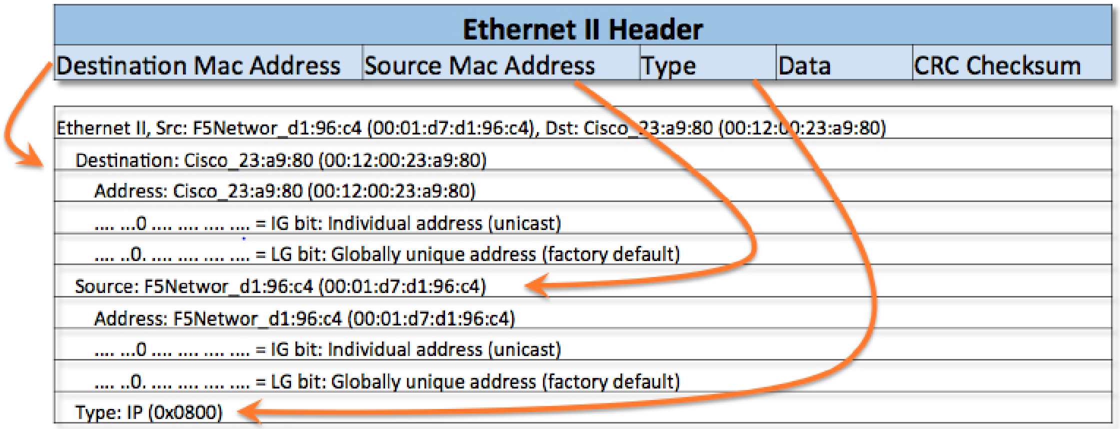


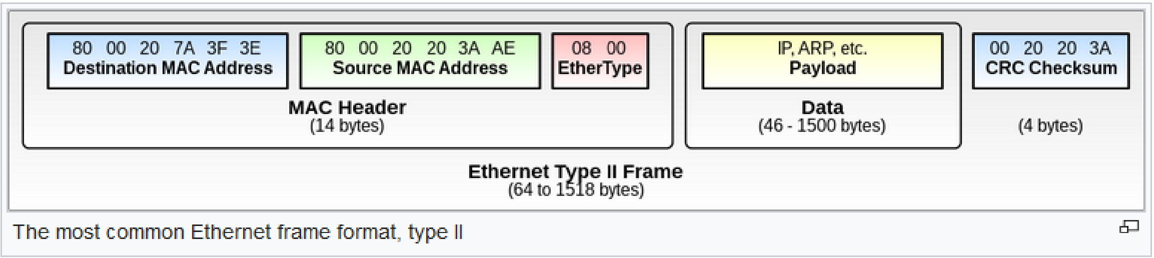
Figure 2: Full Ethernet Frame Format



Ethernet II Header				
Destination Mac Address	Source Mac Address	Type	Data	CRC Checksum

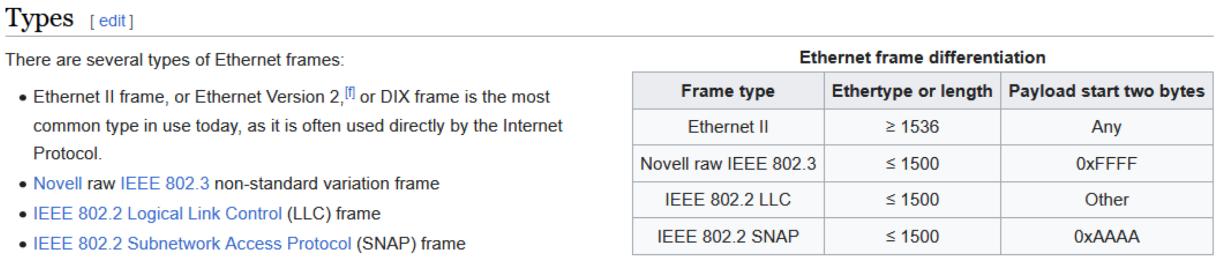
```
Ethernet II, Src: F5Networ_d1:96:c4 (00:01:d7:d1:96:c4), Dst: Cisco_23:a9:80 (00:12:00:23:a9:80)
  Destination: Cisco_23:a9:80 (00:12:00:23:a9:80)
    Address: Cisco_23:a9:80 (00:12:00:23:a9:80)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  Source: F5Networ_d1:96:c4 (00:01:d7:d1:96:c4)
    Address: F5Networ_d1:96:c4 (00:01:d7:d1:96:c4)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
```

Figure 3: Ethernet Header example with WireShark



The most common Ethernet frame format, type II

Figure 3: Another Visual Representation of an Ethernet Frame



Types [edit]

There are several types of Ethernet frames:

- Ethernet II frame, or Ethernet Version 2,[f] or DIX frame is the most common type in use today, as it is often used directly by the Internet Protocol.
- Novell raw IEEE 802.3 non-standard variation frame
- IEEE 802.2 Logical Link Control (LLC) frame
- IEEE 802.2 Subnetwork Access Protocol (SNAP) frame

Ethernet frame differentiation

Frame type	Ethertype or length	Payload start two bytes
Ethernet II	≥ 1536	Any
Novell raw IEEE 802.3	≤ 1500	0xFFFF
IEEE 802.2 LLC	≤ 1500	Other
IEEE 802.2 SNAP	≤ 1500	0xAAAA

Source: https://en.wikipedia.org/wiki/Ethernet_frame

Ethernet Frame Field Descriptions

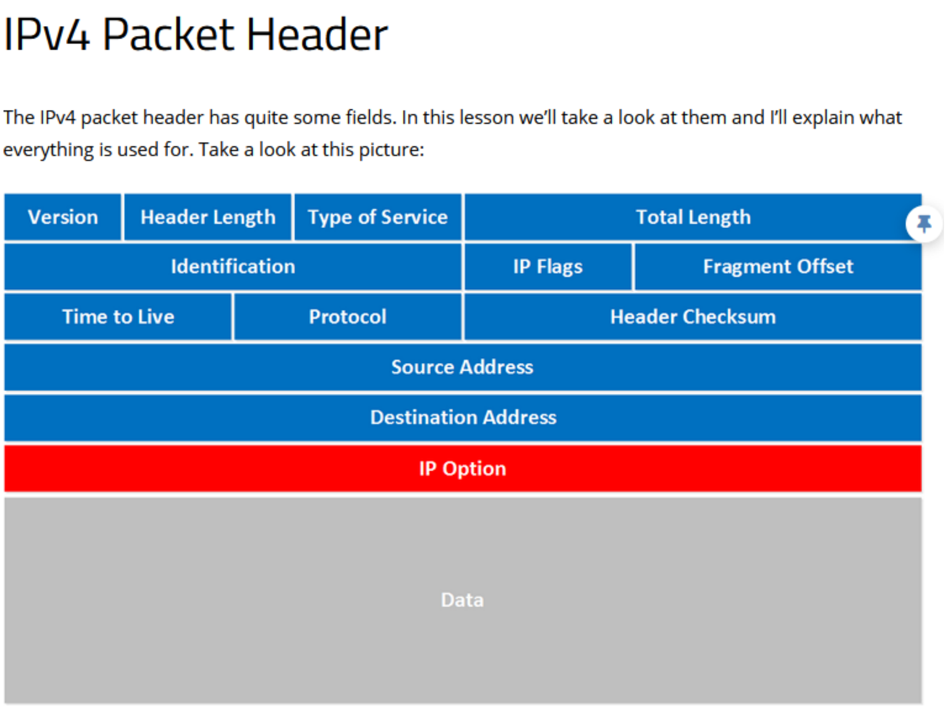
- *Preamble*---preamble is used to inform the receiving stations that a frame is coming, and provide a means to synchronize the frame-reception portions of receivers physical layers. The preamble consists of 7 octets of alternating ones & zeros followed by a single octet, called the start of frame delimiter (SOF) which consists of 6 alternating bits ending in two one-bit values. **Note:** this is rarely shown in examples.
- *Destination address*---MAC address that identifies which node(s) should receive the frame.
- *Source address*---MAC address that identifies the sending node of the frame.

- *Length or Type*---indicates either the length of the data portion of the frame in octets, **or** the type of the data contained within the data portion of the frame. If the value is <= 1500 (x05DC), the value is a length value. If the value is >=1536 (x0600), the value is a type value. Example type values are:
- `x0800: IPv4`
- `x0806: ARP`
- `x8137: IPX`
- `x86DD: IPv6`
- `x9000: Loopback`

  (see http://standards.ieee.org/develop/regauth/ethertype/eth.txt or http://www.iana.org/assignments/ethernet-numbers)

- *Data*---a sequence of n octets, where n must be <= 1500. If the length of n is < 46, data is added to "pad" the length of the data to 46 bytes.
- *Frame Check Sequence*---specifically for 802.3, this is a 32-bit Cyclic Redundancy Check (CRC-32).

Source: https://homepages.uc.edu/~thomam/Net1/Packet_Formats/ethernet.html



IPv4 Packet Header

The IPv4 packet header has quite some fields. In this lesson we'll take a look at them and I'll explain what everything is used for. Take a look at this picture:

<table>
<tr><td>Version</td><td colspan="2">Header Length</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="4">Identification</td><td>IP Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="2">Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="6">IP Option</td></tr>
<tr><td colspan="6">Data</td></tr>
</table>

Figure 4: IP Header

- **Version**: the first field tells us which IP version we are using, only IPv4 uses this header so you will always find decimal value 4 here.
- **Header Length**: this 4 bit field tells us the length of the IP header in 32 bit increments. The minimum length of an IP header is 20 bytes so with 32 bit increments, you would see value of 5 here. The maximum value we can create with 4 bits is 15 so with 32 bit increments, that would be a header length of 60 bytes. This field is also called the **Internet Header Length (IHL)**.

- **Type of Service**: this is used for QoS (Quality of Service). There are 8 bits that we can use to mark the packet which we can use to give the packet a certain treatment. You can read more about this field in my IP precedence and DSCP lesson.
- **Total Length**: this 16-bit field indicates the entire size of the IP packet (header and data) in bytes. The minimum size is 20 bytes (if you have no data) and the maximum size is 65.535 bytes, that's the highest value you can create with 16 bits.
- **Identification**: If the IP packet is fragmented then each fragmented packet will use the same 16 bit identification number to identify to which IP packet they belong to.
- **IP Flags**: These 3 bits are used for fragmentation:
  - The first bit is always set to 0.
  - The second bit is called the **DF (Don't Fragment) bit** and indicates that this packet should not be fragmented.
  - The third bit is called the **MF (More Fragments)** bit and is set on all fragmented packets except the last one.
- **Fragment Offset**: this 13 bit field specifies the position of the fragment in the original fragmented IP packet.
- **Time to Live**: Everytime an IP packet passes through a router, the time to live field is decremented by 1. Once it hits 0 the router will drop the packet and sends an ICMP time exceeded message to the sender. The time to live field has 8 bits and is used to prevent packets from looping around forever (if you have a routing loop).
- **Protocol**: this 8 bit field tells us which protocol is enapsulated in the IP packet, for example TCP has value 6 and UDP has value 17.
- **Header Checksum**: this 16 bit field is used to store a checksum of the header. The receiver can use the checksum to check if there are any errors in the header.
- **Source Address**: here you will find the 32 bit source IP address.
- **Destination Address**: and here's the 32 bit destination IP address.
- **IP Option**: this field is not used often, is optional and has a variable length based on the options that were used. When you use this field, the value in the header length field will increase. An example of a possible option is "source route" where the sender requests for a certain routing path.

Source: https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/ipv4-packet-header

TCP Header

TCP (Transmission Control Protocol) is a reliable transport protocol as it establishes a connection before sending any data and everything that it sends is acknowledged by the receiver. In this lesson we will take a closer look at the TCP header and its different fields. Here's what it looks like:

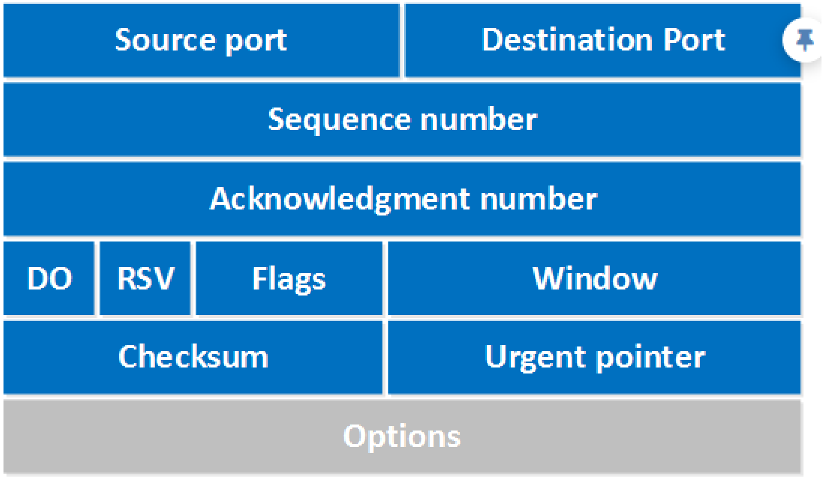


Figure 5: TCP Header

Let's walk through all these fields:

- Source port: this is a 16 bit field that specifies the port number of the sender.
- Destination port: this is a 16 bit field that specifies the port number of the receiver.
- Sequence number: the sequence number is a 32 bit field that indicates how much data is sent during the TCP session. When you establish a new TCP connection (3 way handshake) then the initial sequence number is a random 32 bit value. The receiver will use this sequence number and sends back an acknowledgment. Protocol analyzers like wireshark will often use a relative sequence number of 0 since it's easier to read than some high random number.
- Acknowledgment number: this 32 bit field is used by the receiver to request the next TCP segment. This value will be the sequence number incremented by 1.
- DO: this is the 4 bit data offset field, also known as the header length. It indicates the length of the TCP header so that we know where the actual data begins.
- RSV: these are 3 bits for the reserved field. They are unused and are always set to 0.
- Flags: there are 9 bits for flags, we also call them control bits. We use them to establish connections, send data and terminate connections:
- URG: urgent pointer. When this bit is set, the data should be treated as priority over other data.
- ACK: used for the acknowledgment.
- PSH: this is the push function. This tells an application that the data should be transmitted immediately and that we don't want to wait to fill the entire TCP segment.
- RST: this resets the connection, when you receive this you have to terminate the connection right away. This is only used when there are unrecoverable errors and it's not a normal way to finish the TCP connection.
- SYN: we use this for the initial three way handshake and it's used to set the initial sequence number.
- FIN: this finish bit is used to end the TCP connection. TCP is full duplex so both parties will have to use the FIN bit to end the connection. This is the normal method how we end an connection.

- Window: the 16 bit window field specifies how many bytes the receiver is willing to receive. It is used so the receiver can tell the sender that it would like to receive more data than what it is currently receiving. It does so by specifying the number of bytes beyond the sequence number in the acknowledgment field.
- Checksum: 16 bits are used for a checksum to check if the TCP header is OK or not.
- Urgent pointer: these 16 bits are used when the URG bit has been set, the urgent pointer is used to indicate where the urgent data ends.
- Options: this field is optional and can be anywhere between 0 and 320 bits.

Source: https://networklessons.com/cisco/ccie-routing-switching-written/tcp-header