ETHERNET FRAMES

DEFINITION:

Ethernet frames are the basic units of data transmitted over an Ethernet network. They encapsulate the data to be sent across the network and include various fields containing information necessary for proper transmission and reception. Here are the key components typically found in an Ethernet frame

Header:

The Ethernet frame header contains essential information for proper communication and handling of the frame. It typically includes:

Preamble: A 7-byte (56-bit) sequence of alternating 1s and 0s used for synchronization and preamble detection.

Start Frame Delimiter (SFD): A specific bit sequence (1 byte) indicating the end of the preamble and the start of the frame's actual data.

Destination MAC Address: A 6-byte (48-bit) field specifying the MAC address of the intended recipient device.

Source MAC Address: A 6-byte (48-bit) field specifying the MAC address of the sender device.

Type/Length Field: A 2-byte (16-bit) field that indicates the type of payload carried within the frame (for Ethernet II frames) or the length of the payload (for IEEE 802.3 frames).

Payload:

This section of the Ethernet frame carries the actual data to be transmitted. The size of the payload can vary, typically ranging from 46 to 1500 bytes (excluding headers and trailers) in traditional Ethernet frames.

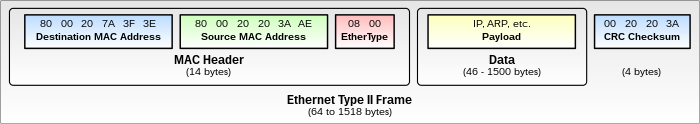
Footer (Trailer):

The Ethernet frame trailer includes:

Frame Check Sequence (FCS): A 4-byte (32-bit) field containing a cyclic redundancy check (CRC) value. It is used for error detection, allowing the receiving device to verify if the frame has been transmitted without errors.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **802.3 Ethernet packet and frame structure** | | | | | | | | | |
| **Layer** | **Preamble** | **Start frame delimiter (SFD)** | **MAC destination** | **MAC source** | [**802.1Q**](https://en.wikipedia.org/wiki/802.1Q)**tag (optional)** | [**Ethertype**](https://en.wikipedia.org/wiki/Ethertype)**(**[**Ethernet II**](https://en.wikipedia.org/wiki/Ethernet_II)**) or length (**[**IEEE 802.3**](https://en.wikipedia.org/wiki/IEEE_802.3)**)** | **Payload** | [**Frame check sequence**](https://en.wikipedia.org/wiki/Frame_check_sequence)**(32‑bit**[**CRC**](https://en.wikipedia.org/wiki/Cyclic_redundancy_check)**)** | [**Interpacket gap (IPG)**](https://en.wikipedia.org/wiki/Interpacket_gap) |
|  | 7 [octets](https://en.wikipedia.org/wiki/Octet_(computing)) | 1 octet | 6 octets | 6 octets | (4 octets) | 2 octets | 46–1500 octets | 4 octets | 12 octets |
| [Layer 2](https://en.wikipedia.org/wiki/Data_link_layer) Ethernet frame | (not part of the frame) | | ← 64–1522 octets → | | | | | | (not part of the frame) |
| [Layer 1](https://en.wikipedia.org/wiki/Physical_layer) Ethernet packet & IPG | ← 72–1530 octets → | | | | | | | | ← 12 octets → |

The optional 802.1Q tag consumes additional space in the frame. Field sizes for this option are shown in brackets in the table above. [IEEE 802.1ad](https://en.wikipedia.org/wiki/IEEE_802.1ad) (Q-in-Q) allows for multiple tags in each frame.

[](https://en.wikipedia.org/wiki/File:Ethernet_Type_II_Frame_format.svg)

**Novell raw IEEE 802.3**

* Novell's "raw" 802.3 frame format was based on early IEEE 802.3 work. Novell used this as a starting point to create the first implementation of its own [IPX](https://en.wikipedia.org/wiki/Internetwork_Packet_Exchange) Network Protocol over Ethernet.
* They did not use any LLC header but started the IPX packet directly after the length field. This does not conform to the IEEE 802.3 standard, but since IPX always has FF as the first two octets (while in IEEE 802.2 LLC that pattern is theoretically possible but extremely unlikely), in practice this usually coexists on the wire with other Ethernet implementations, with the notable exception of some early forms of [DECnet](https://en.wikipedia.org/wiki/DECnet" \o "DECnet) which got confused by this.
* [Novell NetWare](https://en.wikipedia.org/wiki/Novell_NetWare) used this frame type by default until the mid-nineties, and since NetWare was then very widespread, while IP was not, at some point in time most of the world's Ethernet traffic ran over "raw" 802.3 carrying IPX. Since NetWare 4.10, NetWare defaults to IEEE 802.2 with LLC (NetWare Frame Type Ethernet\_802.2) when using IPX.[[11]](https://en.wikipedia.org/wiki/Ethernet_frame#cite_note-18)

**IEEE 802.2 LLC**

*Main article:*[*IEEE 802.2*](https://en.wikipedia.org/wiki/IEEE_802.2)

* Some protocols, such as those designed for the [OSI stack](https://en.wikipedia.org/wiki/OSI_model), operate directly on top of IEEE 802.2 LLC encapsulation, which provides both connection-oriented and connectionless network services.
* IEEE 802.2 LLC encapsulation is not in widespread use on common networks currently, with the exception of large corporate [NetWare](https://en.wikipedia.org/wiki/NetWare) installations that have not yet migrated to NetWare over [IP](https://en.wikipedia.org/wiki/Internet_Protocol). In the past, many corporate networks used IEEE 802.2 to support transparent translating bridges between Ethernet and [Token Ring](https://en.wikipedia.org/wiki/Token_Ring) or [FDDI](https://en.wikipedia.org/wiki/FDDI) networks.
* There exists an [Internet standard](https://en.wikipedia.org/wiki/Internet_standard) for encapsulating IPv4 traffic in IEEE 802.2 LLC SAP/SNAP frames.[[12]](https://en.wikipedia.org/wiki/Ethernet_frame#cite_note-19) It is almost never implemented on Ethernet, although it is used on FDDI, Token Ring, [IEEE 802.11](https://en.wikipedia.org/wiki/IEEE_802.11) (with the exception of the [5.9 GHz band](https://en.wikipedia.org/wiki/IEEE_802.11p), where it uses EtherType)[[13]](https://en.wikipedia.org/wiki/Ethernet_frame#cite_note-20) and other [IEEE 802](https://en.wikipedia.org/wiki/IEEE_802) LANs. IPv6 can also be transmitted over Ethernet using IEEE 802.2 LLC SAP/SNAP, but, again, that's almost never used.

**IEEE 802.2 SNAPfd**

*Main article:*[*Subnetwork Access Protocol*](https://en.wikipedia.org/wiki/Subnetwork_Access_Protocol)

* By examining the 802.2 LLC header, it is possible to determine whether it is followed by a SNAP header. The LLC header includes two eight-bit address fields, called *service access points* (SAPs) in OSI terminology; when both source and destination SAP are set to the value 0xAA, the LLC header is followed by a SNAP header. The SNAP header allows EtherType values to be used with all IEEE 802 protocols, as well as supporting private protocol ID spaces.
* In IEEE 802.3x-1997, the IEEE Ethernet standard was changed to explicitly allow the use of the 16-bit field after the MAC addresses to be used as a length field or a type field.
* The [AppleTalk](https://en.wikipedia.org/wiki/AppleTalk) v2 protocol suite on Ethernet ("[EtherTalk](https://en.wikipedia.org/wiki/AppleTalk" \l "EtherTalk,_TokenTalk_and_AppleShare" \o "AppleTalk)") uses IEEE 802.2 LLC + SNAP encapsulation.