Ethernet frame

Ethernet packet. The SFD (start frame delimiter) marks the end of the packet preamble. It is immediately followed by the Ethernet frame, which starts with the destination MAC address.[1]

In computer networking, an **Ethernet frame** is a data link layer protocol data unit and uses the underlying Ethernet physical layer transport mechanisms. In other words, a data unit on an Ethernet link transports an Ethernet frame as its payload.[2]

An Ethernet frame is preceded by a preamble and start frame delimiter (SFD), which are both part of the Ethernet packet at the physical layer. Each Ethernet frame starts with an Ethernet header, which contains destination and source MAC addresses as its first two fields. The middle section of the frame is payload data including any headers for other protocols (for example, Internet Protocol) carried in the frame. The frame ends with a frame check sequence (FCS), which is a 32-bit cyclic redundancy check used to detect any in-transit corruption of data.

Structure[edit]

See also: Physical Coding Sublayer

A data packet on the wire and the frame as its payload consist of binary data. Ethernet transmits data with the most-significant octet (byte) first; within each octet, however, the least-significant bit is transmitted first.[a]

The internal structure of an Ethernet frame is specified in IEEE 802.3.[2] The table below shows the complete Ethernet packet and the frame inside, as transmitted, for the payload size up to the MTU of 1500 octets.[b] Some implementations of Gigabit Ethernet and other higher-speed variants of Ethernet support larger frames, known as jumbo frames.

802.3 Ethernet packet and frame structure

Layer	Preamble	Start frame delimiter (SFD)	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap (IPG)
Length (octets)	7	1	6	6	(4)	2	42–1500[c]	4	12

Layer 2 Ethernet frame	(not part of the frame)	← 64–1522 octets →	(not part of the frame)
Layer 1 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 (Q-in-Q) allows for multiple tags in each frame. This option is not illustrated here.

Ethernet packet – physical layer[edit]

Preamble and start frame delimiter[edit]

See also: Syncword

An Ethernet packet starts with a seven-octet (56-bit) preamble and one-octet (8-bit) *start frame delimiter* (SFD).[d] The preamble bit values alternate 1 and 0, allowing receivers to synchronize their clock at the bit-level with the transmitter. The preamble is followed by the SFD which ends with a 1 instead of 0, to break the bit pattern of the preamble and signal the start of the actual frame.[1]: section 4.2.5

Physical layer transceiver circuitry (PHY for short) is required to connect the Ethernet MAC to the physical medium. The connection between a PHY and MAC is independent of the physical medium and uses a bus from the media independent interface family (MII, GMII, RGMII, SGMII, XGMII). The preamble and SFD representation depends on the width of the bus:

Preamble and SFD representations as bits, decimal, bytes, and nibbles

Representation	56-bit (7-byte) Preamble							SFD byte
uncoded on-the-wire **bit pattern** transmitted from left to right (used by Ethernet variants transmitting serial bits instead of	10101010	10101010	10101010	10101010	10101010	10101010	10101010	10101011

larger symbols)[1]:sections 4.2.5 and 3.2.2																
decimal in Ethernet LSb-first ordering[1]:sections 3.2.2, 3.3 and 4.2.6	85		85		85		85		85		85		85		213	
hexadecimal LSb-first **bytes** for 8-bit wide busses (GMII bus for Gigabit Ethernet transceivers)	0x55		0x55		0x55		0x55		0x55		0x55		0x55		0xD5	
hexadecimal LSb-first **nibbles** for 4-bit wide busses (MII bus for Fast Ethernet or RGMII for gigabit transceivers)	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0x5	0xD

The SFD is immediately followed by the destination MAC address, which is the first field in an Ethernet frame.

Frame – data link layer[edit]

Header[edit]

The header features destination and source MAC addresses (each six octets in length), the EtherType field and, optionally, an IEEE 802.1Q tag or IEEE 802.1ad tag.

The EtherType field is two octets long and it can be used for two different purposes. Values of 1500 and below mean that it is used to indicate the size of the payload in octets, while values of 1536 and above indicate that it is used as an EtherType, to indicate which protocol is encapsulated in the payload of the frame. When used as EtherType, the length of the frame is determined by the location of the interpacket gap and valid frame check sequence (FCS).

The IEEE 802.1Q tag or IEEE 802.1ad tag, if present, is a four-octet field that indicates virtual LAN (VLAN) membership and IEEE 802.1p priority. The first two octets of the tag are called the **T**ag **P**rotocol **ID**entifier (TPID) and double as the EtherType field indicating that the frame is either 802.1Q or 802.1ad tagged. 802.1Q uses a TPID of 0x8100. 802.1ad uses a TPID of 0x88a8.

Payload[edit]

Payload is a variable-length field. Its minimum size is governed by a requirement for a minimum frame transmission of 64 octets (bytes).[e] With header and FCS taken into account, the minimum payload is 42 octets when an 802.1Q tag is present[f] and 46

octets when absent. When the actual payload is less than the minimum, padding octets are added accordingly. IEEE standards specify a maximum payload of 1500 octets. Non-standard jumbo frames allow for larger payloads on networks built to support them.

Frame check sequence[edit]

The frame check sequence (FCS) is a four-octet cyclic redundancy check (CRC) that allows detection of corrupted data within the entire frame as received on the receiver side. According to the standard, the FCS value is computed as a function of the protected MAC frame fields: source and destination address, length/type field, MAC client data and padding (that is, all fields except the FCS).

Per the standard, this computation is done using the left shifting CRC-32 (polynomial = 0x4C11DB7, initial CRC = 0xFFFFFFFF, CRC is post complemented, verify value = 0x38FB2284) algorithm. The standard states that data is transmitted least significant bit (bit 0) first, while the FCS is transmitted most significant bit (bit 31) first.[1]:section 3.2.9 An alternative is to calculate a CRC using the right shifting CRC-32 (polynomial = 0xEDB88320, initial CRC = 0xFFFFFFFF, CRC is post complemented, verify value = 0x2144DF1C), which will result in a CRC that is a bit reversal of the FCS, and transmit both data and the CRC least significant bit first, resulting in identical transmissions.

The standard states that the receiver should calculate a new FCS as data is received and then compare the received FCS with the FCS the receiver has calculated. An alternative is to calculate a CRC on both the received data and the FCS, which will result in a fixed non-zero "verify" value. (The result is non-zero because the CRC is post complemented during CRC generation). Since the data is received least significant bit first, and to avoid having to buffer octets of data, the receiver typically uses the right shifting CRC-32. This makes the "verify" value (sometimes called "magic check") 0x2144DF1C.[5]

However, hardware implementation of a logically right shifting CRC may use a left shifting Linear Feedback Shift Register as the basis for calculating the CRC, reversing the bits and resulting in a verify value of 0x38FB2284. Since the complementing of the CRC may be performed post calculation and during transmission, what remains in the hardware register is a non-complemented result, so the residue for a right shifting implementation would be the complement of 0x2144DF1C = 0xDEBB20E3, and for a left shifting implementation, the complement of 0x38FB2284 = 0xC704DD7B.

End of frame – physical layer[edit]

The *end of a frame* is usually indicated by the end-of-data-stream symbol at the physical layer or by loss of the carrier signal; an example is 10BASE-T, where the receiving station detects the end of a transmitted frame by loss of the carrier. Later physical layers use an explicit *end of data* or *end of stream* symbol or sequence to avoid ambiguity, especially where the carrier is continually sent between frames; an example is Gigabit Ethernet with its 8b/10b encoding scheme that uses special symbols which are transmitted before and after a frame is transmitted.[6][7]

Interpacket gap – physical layer[edit]

Interpacket gap (IPG) is idle time between packets. After a packet has been sent, transmitters are required to transmit a minimum of 96 bits (12 octets) of idle line state before transmitting the next packet.

Types[edit]

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

There are several types of Ethernet frames:

- Ethernet II frame, or Ethernet Version 2,[q] 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

The different frame types have different formats and MTU values, but can coexist on the same physical medium. Differentiation between frame types is possible based on the table on the right.

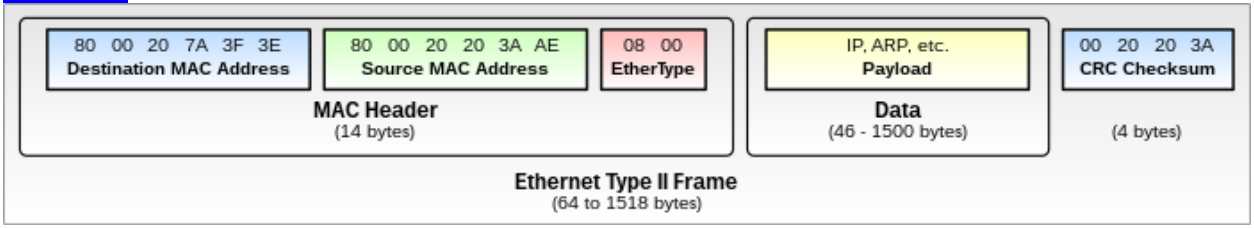
In addition, all four Ethernet frame types may optionally contain an IEEE 802.1Q tag to identify what VLAN it belongs to and its priority (quality of service). This encapsulation is defined in the IEEE 802.3ac specification and increases the maximum frame by 4 octets.

The IEEE 802.1Q tag, if present, is placed between the Source Address and the EtherType or Length fields. The first two octets of the tag are the Tag Protocol Identifier (TPID) value of 0x8100. This is located in the same place as the EtherType/Length field in untagged frames, so an EtherType value of 0x8100 means the frame is tagged, and the true EtherType/Length is located after the Q-tag. The TPID is followed by two octets containing the Tag Control Information (TCI) (the IEEE 802.1p priority (quality of service) and VLAN id). The Q-tag is followed by the rest of the frame, using one of the types described above.

Ethernet II[edit]

Ethernet II framing (also known as **DIX Ethernet**, named after DEC, Intel and Xerox, the major participants in its design[8]), defines the two-octet EtherType field in an Ethernet frame, preceded by destination and source MAC addresses, that identifies an upper layer protocol encapsulated by the frame data. Most notably, an EtherType value of 0x0800 indicates that the frame contains an IPv4 datagram, 0x0806 indicates

an ARP datagram, and 0x86DD indicates an IPv6 datagram. See EtherType § Values for more.



The most common Ethernet frame format, type II

As this industry-developed standard went through a formal IEEE standardization process, the EtherType field was changed to a (data) length field in the new 802.3 standard.[h] Since the recipient still needs to know how to interpret the frame, the standard required an IEEE 802.2 header to follow the length and specify the type. Many years later, the 802.3x-1997 standard, and later versions of the 802.3 standard, formally approved of both types of framing. Ethernet II framing is the most common in Ethernet local area networks, due to its simplicity and lower overhead.

In order to allow some frames using Ethernet II framing and some using the original version of 802.3 framing to be used on the same Ethernet segment, EtherType values must be greater than or equal to 1536 (0x0600). That value was chosen because the maximum length of the payload field of an Ethernet 802.3 frame is 1500 octets (0x05DC). Thus if the field's value is greater than or equal to 1536, the frame must be an Ethernet II frame, with that field being a type field.[9] If it's less than or equal to 1500, it must be an IEEE 802.3 frame, with that field being a length field. Values between 1500 and 1536, exclusive, are undefined.[10] This convention allows software to determine whether a frame is an Ethernet II frame or an IEEE 802.3 frame, allowing the coexistence of both standards on the same physical medium.

Novell raw IEEE 802.3[edit]

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 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 which got confused by this.

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]

IEEE 802.2 LLC[edit]

Main article: IEEE 802.2

Some protocols, such as those designed for the OSI stack, 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 installations that have not yet migrated to NetWare over IP. In the past, many corporate networks used IEEE 802.2 to support transparent translating bridges between Ethernet and Token Ring or FDDI networks.

There exists an Internet standard for encapsulating IPv4 traffic in IEEE 802.2 LLC SAP/SNAP frames.[12] It is almost never implemented on Ethernet, although it is used on FDDI, Token Ring, IEEE 802.11 (with the exception of the 5.9 GHz band, where it uses EtherType)[13] and other 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 SNAP[edit]

Main article: 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 v2 protocol suite on Ethernet ("EtherTalk") uses IEEE 802.2 LLC + SNAP encapsulation.

Maximum throughput[edit]

We may calculate the protocol overhead for Ethernet as a percentage (packet size including IPG)

We may calculate the *protocol efficiency* for Ethernet

Maximum efficiency is achieved with largest allowed payload size and is:

for untagged frames, since the packet size is maximum 1500 octet payload + 8 octet preamble + 14 octet header + 4 octet trailer + minimum interpacket gap corresponding to 12 octets = 1538 octets. The maximum efficiency is:

when 802.1Q VLAN tagging is used.

The throughput may be calculated from the efficiency

,

where the physical layer net bit rate (the wire bit rate) depends on the Ethernet physical layer standard, and may be 10 Mbit/s, 100 Mbit/s, 1 Gbit/s or 10 Gbit/s. Maximum throughput for 100BASE-TX Ethernet is consequently 97.53 Mbit/s without 802.1Q, and 97.28 Mbit/s with 802.1Q.

Channel utilization is a concept often confused with protocol efficiency. It considers only the use of the channel disregarding the nature of the data transmitted – either payload or overhead. At the physical layer, the link channel and equipment do not know the difference between data and control frames. We may calculate the channel utilization:

The total time considers the round trip time along the channel, the processing time in the hosts and the time transmitting data and acknowledgements. The time spent transmitting data includes data and acknowledgements.

Runt frames[edit]

A runt frame is an Ethernet frame that is less than the IEEE 802.3's minimum length of 64 octets. Runt frames are most commonly caused by collisions; other possible causes are a malfunctioning network card, buffer underrun, duplex mismatch or software issues.[14]