

Ethernet frame

From Wikipedia, the free encyclopedia

A data packet on an Ethernet link is called an *Ethernet packet*, which transports an **Ethernet frame** as payload.^[1]

An Ethernet frame is preceded by a preamble and start frame delimiter (SFD), which are both part of the layer 1 Ethernet packet. 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.

Contents

- 1 Structure
 - 1.1 Preamble and start frame delimiter
 - 1.2 Header
 - 1.3 Payload
 - 1.4 Frame check sequence
 - 1.5 End of frame
 - 1.6 Interpacket gap
- 2 Ethernet frame types
 - 2.1 Ethernet II
 - 2.2 Novell raw IEEE 802.3
 - 2.3 IEEE 802.2 LLC
 - 2.4 IEEE 802.2 SNAP
- 3 Maximum throughput
- 4 Runt frames
- 5 Notes
- 6 References
- 7 Further reading

Structure

A data packet on the wire and the frame as its payload consist of binary data. Data on Ethernet is transmitted most-significant octet (byte) first. Within each octet, however, the least-significant bit is transmitted first.^[2]

The table below shows the complete Ethernet frame, as transmitted, for the payload size up to the MTU of 1500 octets.^[a] Some implementations of Gigabit Ethernet (and higher speed ethernets) support larger frames, known as jumbo frames.

The internal structure of an Ethernet frame is specified in IEEE 802.3-2012.^[1]

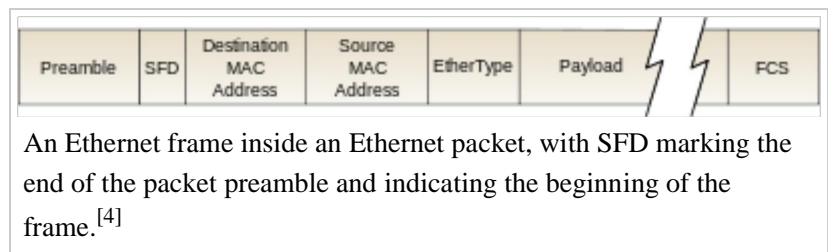
802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46(42) ^[b] –1500 octets	4 octets
Layer 2 Ethernet frame	← 64–1518(1522) octets →							
Layer 1 Ethernet packet	← 72–1526(1530) octets →							

Preamble and start frame delimiter

An Ethernet frame starts following a 7-octet preamble and 1-octet *start frame delimiter* (SFD), both of which are part of the Ethernet packet enveloping the frame.^[c] Prior to Fast Ethernet, the on-the-wire bit pattern for this portion of the frame is 10101010 10101010 10101010 10101010 10101010

10101010 10101010 10101011.^{[4]:sections 4.2.5 and 3.2.2} Since octets are transmitted least-significant bit first, the corresponding hexadecimal representation is 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0xD5.



The SFD is an 8-bit (1-byte) value marking the end of the preamble, which is the first field of an Ethernet packet, and indicating the beginning of the Ethernet frame. The SFD is immediately followed by the destination MAC address, which is the first field in an Ethernet frame. SFD has the value of 171 (10101011 in binary notation), which is transmitted with least-significant bit first as 213 (0xD5).^{[4]:sections 3.1.1 and 3.2}

The preamble of an Ethernet packet consists of a 56-bit (7-byte) pattern of alternating 1 and 0 bits, which allows devices on the network to easily detect a new incoming frame. The SFD is designed to break this pattern and signal the start of the actual frame.^{[4]:section 4.2.5}

Physical layer transceiver chips (PHYs for short) are necessary 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).

Fast Ethernet transceiver chips utilize the MII bus, which is a 4-bit (one nibble) wide bus, therefore the preamble is represented as 14 instances of 0x5, and the start frame delimiter is 0x5 0xD (as nibbles). Gigabit Ethernet transceiver chips use the GMII bus, which is an 8-bit wide interface, thus the sequence would be 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0xD5 (as bytes).

Header

The header features destination and source MAC addresses (each six octets in length), the EtherType field and, optionally, a IEEE 802.1Q 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, if present, is a four-octet field that indicates Virtual LAN (VLAN) membership and IEEE 802.1p priority.

Payload

The minimum payload is 42 octets when an 802.1Q tag is present and 46 octets when absent.^{[3][d][e]} The maximum payload is 1500 octets. Non-standard jumbo frames allow for larger maximum payload size.

Frame check sequence

The frame check sequence (FCS) is a 4-octet cyclic redundancy check which allows detection of corrupted data within the entire frame. Running the FCS algorithm over the received frame data including the FCS will always result in the *magic number* or *CRC32 residue* 0xC704DD7B when data has been transmitted correctly. This allows for receiving a frame and validating the FCS without knowing where the FCS field actually starts.^[6]

End of frame

The *end of a frame* is usually indicated by the end of data stream 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. Some physical layers use an explicit *end of data* or *end of stream* symbol or sequence to avoid ambiguity; 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.^{[7][8]}

Interpacket gap

Interpacket gap 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.

Ethernet frame types

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,^[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

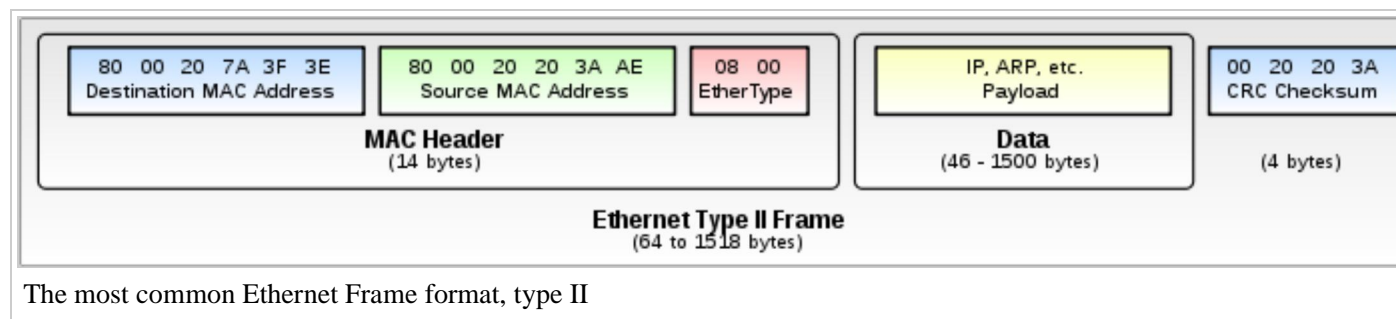
The different frame types have different formats and MTU values, but can coexist on the same physical medium. Differentiation between frame types is possible based on the table on the right.

In addition, all four Ethernet frames 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

Ethernet II framing (also known as **DIX Ethernet**, named after DEC, Intel and Xerox, the major participants in its design^[9]), defines the two-octet EtherType field in an Ethernet frame, preceded by destination and source MAC addresses, that identifies an upper layer protocol encapsulating the frame data. For example, an EtherType value of 0x0800 signals that the frame contains an IPv4 datagram. Likewise, an EtherType of 0x0806 indicates an ARP frame, 0x8100 indicates an IEEE 802.1Q frame and 0x86DD indicates an IPv6 frame.



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.^[g] 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. In practice, both formats are in wide use, with original Ethernet framing the most common in Ethernet local area networks, due to its simplicity and lower overhead.

In order to allow some frames using Ethernet v2 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 v2 frame, with that field being a type field.^[10] 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.^[11] 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

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 has always FF at the first two octets (while in IEEE 802.2 LLC that pattern is theoretically possible but extremely unlikely), in practice this mostly 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 now defaults to IEEE 802.2 with LLC (NetWare Frame Type Ethernet_802.2) when using IPX.^[12]

IEEE 802.2 LLC

Some protocols, 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.^[13] It is almost never implemented on Ethernet, although it is used on FDDI, Token Ring, IEEE 802.11 and other IEEE 802 LANs. IP traffic cannot be encapsulated in IEEE 802.2 LLC frames without SNAP because, although there is a LLC SAP protocol type for IP, there is no such type for ARP, which is required for operation of any medium to large network. 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

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 additional 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 SNAP service is requested. 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.

Mac OS uses IEEE 802.2 LLC SAP/SNAP encapsulation for the AppleTalk v2 protocol suite on Ethernet ("EtherTalk").

Maximum throughput

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

$$\text{Protocol overhead} = \frac{\text{Packet size} - \text{Payload size}}{\text{Packet size}}$$

We may calculate the *protocol efficiency* for Ethernet

$$\text{Protocol efficiency} = \frac{\text{Payload size}}{\text{Packet size}}$$

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

$$\frac{1500}{1538} = 97.53\%$$

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:

$$\frac{1500}{1542} = 97.28\%$$

when 802.1Q VLAN tagging is used.

The throughput may be calculated from the efficiency

$$\text{Throughput} = \text{Efficiency} \times \text{Net bit rate},$$

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:

$$\text{Channel utilization} = \frac{\text{Time spent transmitting data}}{\text{Total time}}$$

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

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 underruns, a bad network card, or software bugs.^[14]

Notes

- a. ^ The bit patterns in the preamble and start of frame delimiter are written as bit strings, with the first bit transmitted on the left (*not* as octet values, which in Ethernet are transmitted least significant bit(s) first). This notation matches the one used in the IEEE 802.3 standard.
- b. ^ 42 octet minimum applies when 802.1Q is present. When absent, 46 octet minimum applies.^[3]

- c. ^ Preamble and start frame delimiter are not displayed by packet sniffing software because these bits are stripped away at OSI Layer 1 by the network interface controller before being passed on to the OSI layer 2 which is where packet sniffers collect their data. There are layer-2 sniffers which can capture and display the preamble and start frame delimiter but they are expensive and mainly used to detect physical related problems.
- d. ^ Minimum payload size is dictated by the 512-bit slot time used for collision detection in the Ethernet LAN architecture.
- e. ^ Both 42 and 46 octet minimums are valid when 802.1Q is present.^[5]
- f. ^ A version 1 Ethernet frame was used for early Ethernet prototypes and featured 8-bit MAC addresses and was never commercially deployed.
- g. ^ Original Ethernet frames define their length with the framing that surrounds it, rather than with an explicit length count.

References

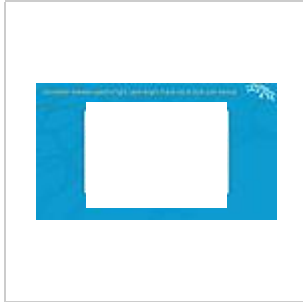
1. ^ ^a ^b "3.1.1 Packet format" (http://standards.ieee.org/getieee802/download/802.3-2012_section1.pdf) (PDF). *IEEE Standard for Ethernet, 802.3-2012 – section one*. 2012-12-28. p. 53. Retrieved 2014-07-06.
2. ^ "Ethernet Frame" (<http://www.infocellar.com/networks/ethernet/frame.htm>). Retrieved 2012-03-20. "Ethernet transmission is strange, in that the octet order is big-endian (leftmost octet is sent first), but bit order little-endian (rightmost, or LSB (Least Significant Bit) of the octet is sent first)."
3. ^ ^a ^b IEEE 802.3-2005 Clause 3.5
4. ^ ^a ^b ^c ^d "802.3-2012 - IEEE Standard for Ethernet" (<http://standards.ieee.org/findstds/standard/802.3-2012.html>) (PDF). *ieee.org*. IEEE Standards Association. 2012-12-28. Retrieved 2014-02-09.
5. ^ IEEE 802.1Q-2011, Annex G
6. ^ Nanditha Jayarajan (2007-04-20). "Configurable LocalLink CRC Reference Design" (http://www.xilinx.com/support/documentation/application_notes/xapp562.pdf) (PDF). xilinx.com. p. 14. Retrieved 2014-06-30.
7. ^ Charles E. Spurgeon (February 2000). "Ethernet: The Definitive Guide" (http://books.google.com/books?id=MRChaUQr0Q0C&printsec=frontcover&source=gbs_ge_summary_r&redir_esc=y#v=onepage&q&f=false). O'Reilly. pp. 41, 47. Retrieved 2014-06-30.
8. ^ "40.1.3.1 Physical Coding Sublayer (PCS)" (http://standards.ieee.org/getieee802/download/802.3-2012_section3.pdf) (PDF). *IEEE Standard for Ethernet, 802.3-2012 – section three*. 2012-12-28. p. 183. Retrieved 2014-07-06.
9. ^ Drew Heywood; Zubair Ahmad (2001). *Drew Heywood's Windows 2000 Network Services*. Sams. p. 53. ISBN 0-672-31741-9.
10. ^ LAN MAN Standards Committee of the IEEE Computer Society (20 March 1997). *IEEE Std 802.3x-1997 and IEEE Std 802.3y-1997*. The Institute of Electrical and Electronics Engineers, Inc. pp. 28–31.
11. ^ IEEE Std 802.3-2005, 3.2.6
12. ^ Don Provan (17 September 1993). "Ethernet Framing (news:1993Sep17.190654.13335@novell.com)". comp.sys.novell (news:comp.sys.novell). Web link (http://groups.google.com/group/bit.listserv.novell/browse_thread/thread/d00a24530625714c). — a classic series of Usenet postings by Novell's Don Provan that have found their way into numerous FAQs and are widely considered the definitive answer to the Novell Frame Type usage.
13. ^ "RFC1042: A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" (<http://tools.ietf.org/html/rfc1042>). Network Working Group of the IETF. February 1988.

14. ^ "Troubleshooting Ethernet" (<http://www.cisco.com/en/US/docs/internetworking/troubleshooting/guide/tr1904.html>). Cisco Systems.

Further reading



Video which explains
how to build an
Ethernet Frame



Minimum Frame
Length in Ethernet
explained

Retrieved from "http://en.wikipedia.org/w/index.php?title=Ethernet_frame&oldid=615789139"

Categories: Ethernet

-
- This page was last modified on 6 July 2014 at 06:50.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.