

Ethernet frame

A data packet on an Ethernet link is called an **Ethernet frame**. A frame begins with preamble and start frame delimiter. Following which, each Ethernet frame continues with an Ethernet header featuring destination and source MAC addresses. The middle section of the frame is payload data including any headers for other protocols (e.g. Internet Protocol) carried in the frame. The frame ends with a 32-bit cyclic redundancy check which is used to detect any corruption of data in transit.

Structure

A data packet on the wire is called a frame and consists of binary data. Data on Ethernet is transmitted most-significant byte first. Within each byte, however, the least-significant bit is transmitted first.^[1]

The table below shows the complete Ethernet frame, as transmitted, for the payload size up to the MTU of 1500 octets.^[2] Some implementations of Gigabit Ethernet (and higher speed ethernets) support larger frames, known as jumbo frames.

802.3 Ethernet frame structure

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)	Interframe gap
7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	42 ^[3] –1500 octets	4 octets	12 octets
← 64–1522 octets →								
← 72–1530 octets →								
← 84–1542 octets →								

Preamble and start frame delimiter

A frame starts with a 7-octet preamble and 1-octet start frame delimiter (SFD).^[4] 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.^[5] Since octets are transmitted least-significant bit first the corresponding hexadecimal representation is 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0xD5.

PHY transceiver chips used for Fast Ethernet feature a 4-bit (one nibble) Media Independent Interface. Therefore the preamble will consist of 14 instances of 0x5, and the start frame delimiter 0x5 0xD. Gigabit Ethernet transceiver chips use a Gigabit Media Independent Interface that works 8-bits at a time, and 10 Gbit/s (XGMII) PHY works with 32-bits at a time.

Header

The header features source and destination MAC addresses which have 6 octets each, the EtherType protocol identifier field and optional IEEE 802.1Q tag.

802.1Q tag

The IEEE 802.1Q tag is an optional 4-octet field that indicates Virtual LAN (VLAN) membership and IEEE 802.1p priority.

EtherType or length

EtherType is a two-octet field in an Ethernet frame. It is used to indicate which protocol is encapsulated in the payload of an Ethernet Frame.

Payload

The minimum payload is 42 octets when 802.1Q tag is present and 46 octets when absent.^[6] and the maximum payload is 1500 octets. Non-standard jumbo frames allow for larger maximum payload size.

Frame check sequence

The frame check sequence is a 4-octet cyclic redundancy check which allows detection of corrupted data within the entire frame.

Interframe gap

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

Ethernet frame types

There are several types of Ethernet frames. The different frame types have different formats and MTU values, but can coexist on the same physical medium.

- The Ethernet Version 2^[7] or Ethernet II frame or DIX frame is the most common type in use today, as it is often used directly by the Internet Protocol.
- Novell's non-standard variation of raw IEEE 802.3 frame
- IEEE 802.2 Logical Link Control (LLC) frame
- Subnetwork Access Protocol (SNAP) frame

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^[8]), 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.^[9] Since the packet recipient still needs to know how to interpret the packet, the standard required an IEEE 802.2 header to follow the length and specify the packet 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 packets using Ethernet v2 framing and some packets 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.

802.2 LLC

Some protocols, particularly 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, 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.

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").

Novell raw 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 very widespread back then, 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.^[13]

Maximum throughput

We may calculate the protocol overhead for Ethernet as a percentage

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

We may calculate the *protocol efficiency* for Ethernet

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

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

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

for untagged Ethernet packets, since the frame size is maximum 1500 byte payload + 8 byte preamble + 14 byte header + 4 byte trailer + minimum interframe gap corresponding to 12 bytes = 1538 bytes. 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 acknowledge packets. The time spent transmitting data includes data and acknowledge packets.

Runt frames

A runt frame is an Ethernet frame that is less than the IEEE 802.3 minimum length of 64 octets. Possible causes are collision, underruns, a bad network card or software.^[14]

Notes

- [1] "Ethernet Frame" (<http://www.infocellar.com/networks/ethernet/frame.htm>). . Retrieved 2012-03-20. "Ethernet transmission is strange, in that the byte order is big-endian (leftmost byte is sent first), but bit order little-endian (rightmost, or LSB (Least Significant Bit) of the byte is sent first)."
- [2] 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 byte values, which in Ethernet are transmitted least significant bit(s) first). This notation matches the one used in the IEEE 802.3 standard.
- [3] 42 octet minimum applies when 802.1Q is present. When absent, 46 octet minimum applies. IEEE 802.3-2005 Clause 3.5
- [4] 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.
- [5] IEEE 802.3 Clause 4.2.5
- [6] Minimum payload size is dictated by the slot time used for collision detection in the Ethernet LAN architecture.
- [7] A version 1 Ethernet frame was used for early Ethernet prototypes and featured 8-bit MAC addresses and was never commercially deployed.
- [8] Drew Heywood; Zubair Ahmad (2001). *Drew Heywood's Windows 2000 Network Services*. Sams. p. 53. ISBN 0-672-31741-9.
- [9] Original Ethernet packets define their length with the framing that surrounds it, rather than with an explicit length count.
- [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] "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. .
- [13] Don Provan (1993-09-17). "ep17.190654.13335@novell.com Ethernet Framing (news:1993S)". [news:comp.sys.novell 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.
- [14] "Troubleshooting Ethernet" (<http://www.cisco.com/en/US/docs/internetworking/troubleshooting/guide/tr1904.html>). Cisco Systems.

References

Article Sources and Contributors

Ethernet frame *Source:* <http://en.wikipedia.org/w/index.php?oldid=540790267> *Contributors:* Alexh19740110, Bananastalktome, Bostonvaulter, Buchs, Dancysoft, Dewritech, Electron9, EoGuy, Glenn, Guy Harris, Gyro Copter, Hawaiian717, I340793, Intr, Jiemingwei, Kubanczyk, Kvng, M1ss1ontomars2k4, Magioladitis, Mange01, Mayur, Myjkccd, Ocdex, QueBurro, Quinn d, Shashiskills, Staaki, Systemparadox, Tchai, Tomaxer, Torzsmokus, W Nowicki, Wjwatson67, Xezbeth, Yemisilin, 94 anonymous edits

License

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)
