Ethernet Framing

Ethernet II

This frame type was designed by Digital, Intel, and Xerox and is in its second revision, thus the name Ethernet II.

Preamble

This is an alternating pattern of ones and zeros used to synchronize the receiving stations and to indicate that data is about to begin. The Media Access Unit (MAU) generates the preamble.

Destination and Source Address

These fields contain the addresses of the sender and intended receiver, respectively. The destination address can be either two or six bytes in length, with the latter being the most common, and it can specify a single host, multiple hosts, or all hosts on a network. These addresses are commonly referred to as unicast, multicast, or broadcast, respectively. The Source address, like the destination address, is six bytes in length. The IEEE has assigned each hardware vendor a unique three-byte code (Vendor Code) to be incorporated into each NIC's six-byte address. The hardware vendor is responsible for assigning the last three bytes of the NIC's address. The result is a unique six-byte address.

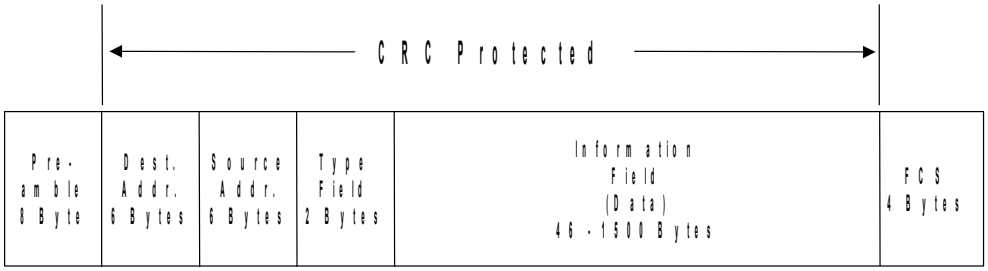
Frame Type

This field indicates which upper layer protocol should be used to interpret the information found in the data portion of the packet. The values found in this field have been defined and managed by Xerox. The frame type assigned to Novell for Ethernet encapsulation is 8137.

Data

The data part of an Ethernet packet contains the protocol header and the actual information being transmitted. The minimum size of an Ethernet packet is sixty-four bytes. When too few data bytes are to be transmitted, the transmitter pads the data field to the minimum size. This minimum frame size is needed in order to detect collisions. Receivers discard any frames shorter than the sixty-four-byte minimum. In the example on the next page, the data field contains an IPX header with its associated upper layer protocol information in its data field.

Ethernet II



Frame Check Sequence

This field is used to check the integrity of the packet. Before the transmitting station places the packet out on the wire, it takes all the bytes within the packet, excluding the preamble and the FCS field, and performs a mathematical calculation called a cyclical redundancy check. The resulting value is then placed in this field. When the packet arrives at the receiving station, it also calculates a separate CRC value on the bytes received. The two values are then compared. If they are equal, the packet is accepted. If not, it is assumed that something has been corrupted and the packet is discarded. This is known as bit level error checking.

IEEE 802.3

An IEEE 802.3 frame is nearly identical to an Ethernet II frame. This is because the IEEE used the original Ethernet standard as the basis for their final product, named for the committee that worked on it, 802.3. The differences between the Ethernet II standard and the 802.3 frame definition are discussed below and the frame is illustrated on the opposite page.

Preamble

This preamble is similar to the preamble used with Ethernet II; the major difference is the length. It is seven bytes rather than eight and is followed by a Start Frame Delimiter.

Start Frame Delimiter

This is a one-byte field of alternating ones and zeros that end with two consecutive ones. These bits are used to signal the beginning of a frame.

Source and Destination Addresses

These fields are identical to the Ethernet II source and destination addresses previously explained.

Frame Length

The Frame Length field replaces the Frame Type field used in an Ethernet II frame. This field indicates the length of the data portion of an 802.3 packet. The only way that a router can differentiate between an Ethernet II frame and an 802.3 frame is to look at the value in the Frame Length or Type field. In an Ethernet II frame, the value found in the Type field would always be greater than 1500 decimals. Since the maximum length of an Ethernet II frame is 1518 bytes (decimal), the length field in an 802.3 frame will always contain a value less than that.

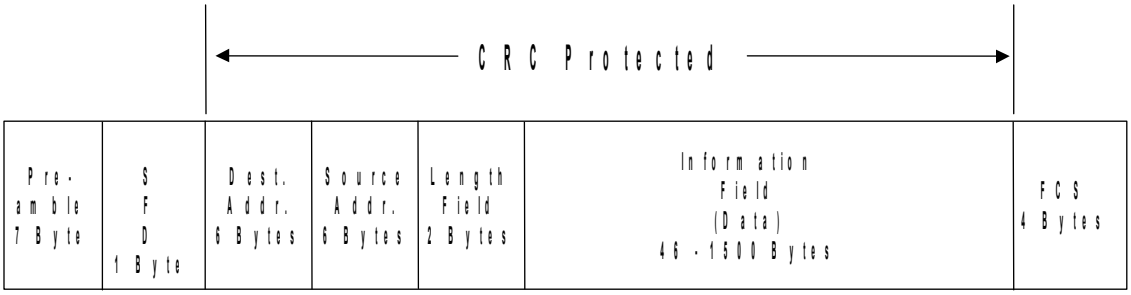
Data Unit

The data unit field in an 802.3 frame can contain either a Logical Link Control header (LLC) or a Subnetwork Access Protocol header (SNAP).

FCS

This field is identical to the FCS field in an Ethernet II frame.

802.3 Frame



IEEE 802.2

The IEEE 802.2 standard provides the information necessary to properly route an 802.3 packet. This standard was developed quite some time after the 802.3 standard. The 802.2 or LLC header envelops the data prior to being encapsulated by the 802.3 header. Because the 802.2 fields make up the Logical Link Control layer, the framed data is sometimes referred to as a Logical Link Control Protocol Data Unit (L-PDU) or just PDU. The LLC frame format adds several additional fields to the header, which are described on the next page.

Destination and Source Service Access Points

These fields indicate the point of service the packet is destined for, or what upper-layer protocol will use the data contained in the information field of the LLC header. Both the DSAP and SSAP fields contain values that identify the upper-layer protocol packet types. The combination of both the DSAP and SSAP are sometimes referred to as Logical Service Access Points (LSAPs). Each of these fields is one byte in length, but only six bits are used for the actual SAP. The first bit in the DSAP indicates whether the destination address is an individual or group address, while the first bit in the SSAP indicates whether the Protocol Data Unit (PDU) contains a request or a response frame. The second bit in both the DSAP and SSAP, if set to a value of one indicates an IEEE-assigned value is contained in the remaining bits. The LLC uses these bits to determine how to process certain bits in the next field, the Control field. For an LLC frame containing Novell information, the IEEE has assigned the value of E0 (hex) to indicate that the data contained in the Information field is a Novell IPX header.

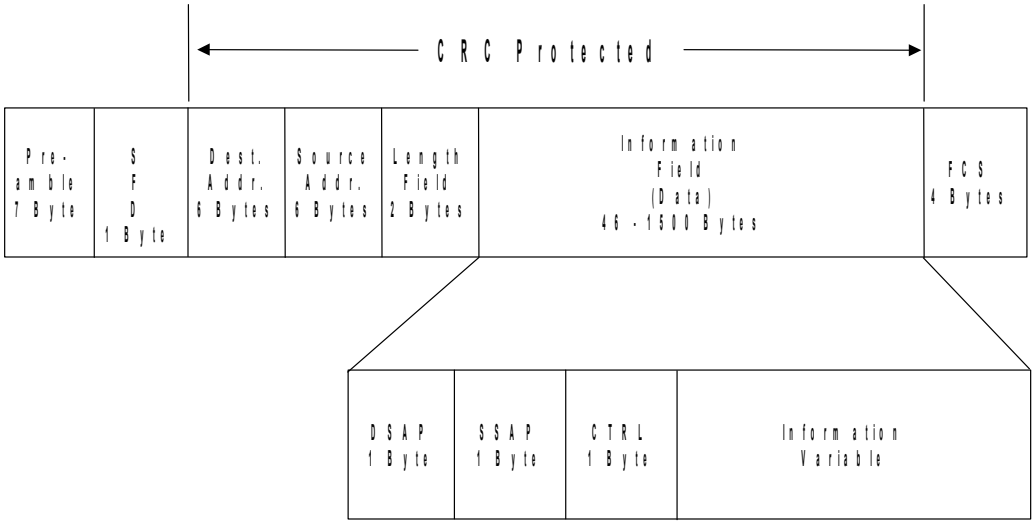
Control Field

The Control Field is used by certain protocols for administrative purposes. It follows the standard defined by High-level Data Link Control (HDLC) and defines three types of data that may be contained in the information field. Information frames are indicated by a 01 (hex) in the Control field, whereas Supervisory frames use a value of 02 (hex), and Unnumbered frames are indicated by the value of 03 (hex). Currently, NetWare's IPX/SPX protocols do not use this field other than to set its value to 03, to denote an 802.2 unnumbered frame format.

Information Field

As shown in the example, this field contains the IPX header.

802.3 Frame w/802.2 Header



SubNetwork Access Protocol (SNAP)

The SNAP standard was developed to ensure that an adequate amount of space was set aside for protocol identification in both Ethernet and Token-Ring headers. Up to this point, the IEEE had only defined a one-byte DSAP and SSAP fields for 802.2 to identify upper layer protocols.

DSAP and SSAP

In order to differentiate a SNAP header from an 802.2 header, the DSAP and SSAP fields are set to a fixed value of AA in a SNAP header.

Organizational Unit Identifier (OUI)

The OUI field is a three-byte field used to identify an organization whose upper layer protocol is identified in the PID field. Some examples are: 000000 for IP & IPX, 08002B Digital, and 080007 Apple Computer.

Protocol Identifier (PID)

A node receiving either an 802.3 or a Token-Ring frame with the DSAP and SSAP fields both set to AA will now to look into the Protocol Identification field for the protocol type information. Examples include 8137 IPX, 0800 IP, 0BAD Banyan VINES, and 809B AppleTalk phase 2.

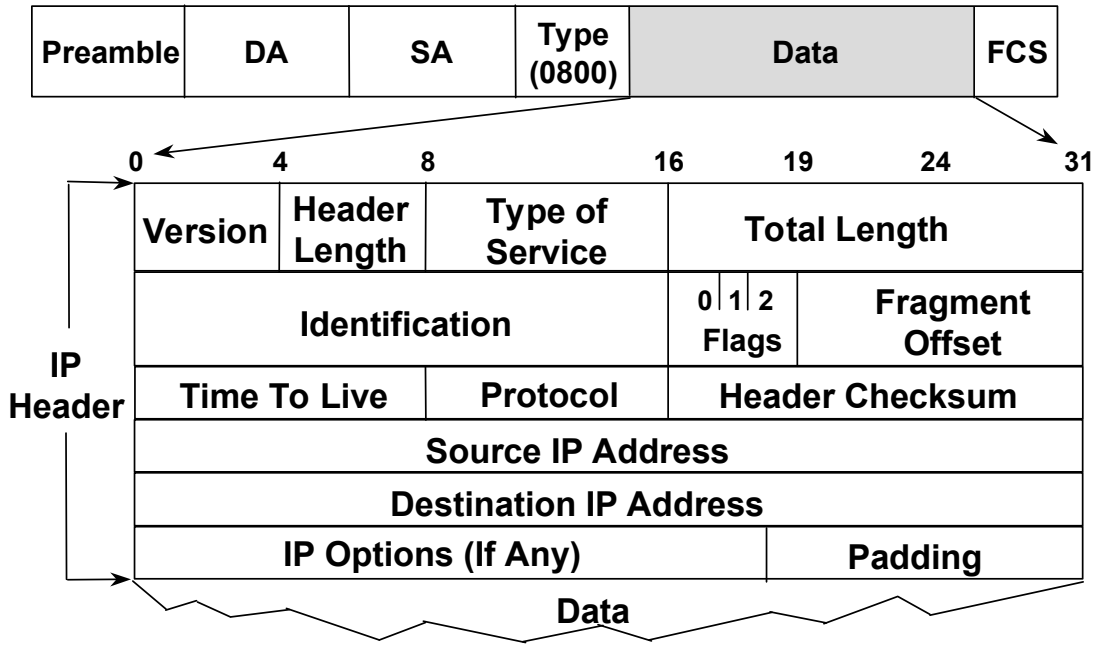
IP Header

Description

For completeness, the remaining elements of an IP datagram are defined as follows:

- **Version—**A 4-bit field that determines the IP version of the protocol used to create the datagram. To be able to interpret the datagram properly, all nodes and gateways must agree on the IP version. Versions that do not match are rejected. The current IP is version is 4.
- **Header length—**A 4-bit field that gives the length of the datagram header. This value will usually be either 5 or more (indicating either five or more 32-bit words in the header).
- **Type of service—**Specifies the handling of the datagram. Different values might indicate precedence, delay, throughput, reliability, or minimum cost.
- **Total length—**The length in octets of the IP datagram, which includes the octets found in the header and data area of the datagram. Based on this 16-bit field, the maximum size of a datagram can be 65,535 octets.
- **Identification—**A unique integer used to identify a datagram. In the case of a fragmented datagram, it is used to correlate arriving fragments of the original datagram in the reassembly process.

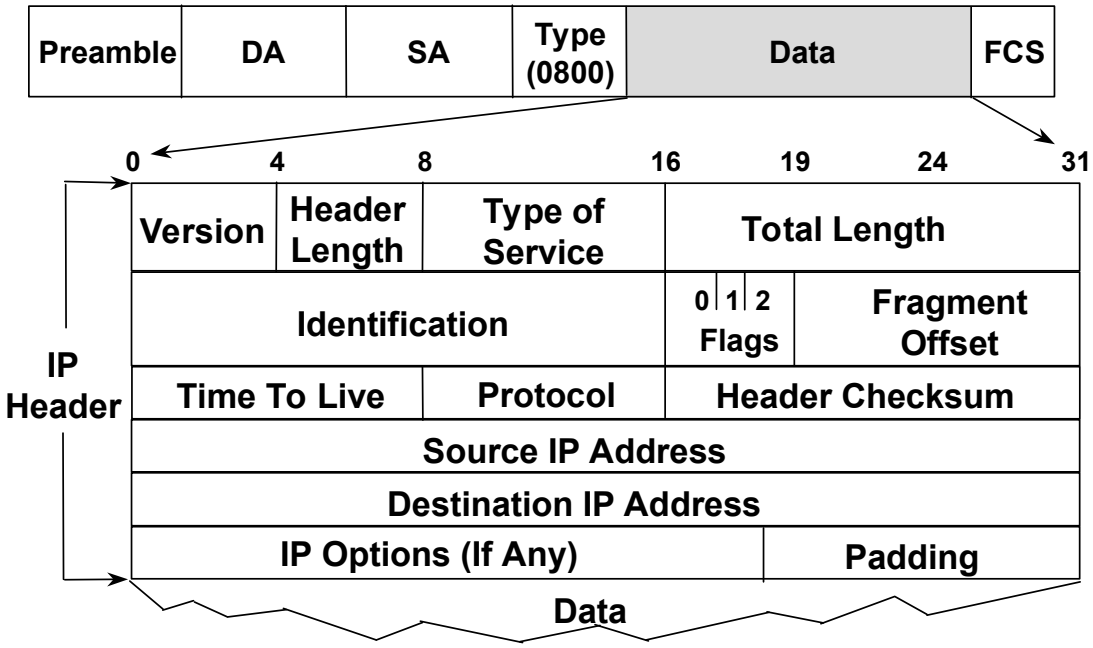
IP Datagram Format



IP Datagram Format (continued)

- **Flags—**A 3-bit field used to control fragmentation of a datagram. Bit 0 of this field is reserved and must be zero. Bit 1 is used to indicate whether or not a datagram may be fragmented. A value of one indicates that it cannot be fragmented. Bit 2 is used to tell the receiving station when it has received the last fragment that belongs to a single datagram. A nonzero value indicates more fragments.
- **Fragment offset—**Measured in units of eight octets starting at offset 0. It specifies an offset value for each data fragment and is used in the reassembly process.
- **Time to live—**Specifies how many seconds the datagram can remain in the Internet. Routers that forward a datagram will decrease this time to live counter by one, as well as decrease the time the datagram waited in the router for processing. If this counter reaches zero, the datagram is discarded and an Internet Control Message Protocol (ICMP) message is sent back to the originating host.
- **Protocol—**Defines the upper layer protocol type that is being transmitted in the data portion of an IP datagram (for example, 17 for UDP or 6 for TCP).
- **Header checksum—**Checks the integrity of the datagram header, not the data in the data field.
- **Source and destination IP addresses—**Contains the 32-bit network address of the sender and receiver of the datagram.
- **IP options—**Used primarily for testing and debugging a network. For example, the record route option allows the source to create an empty list of IP addresses within the header and to arrange for each gateway that handles the datagram to add its IP address to the list.
- **Padding—**Because of the varying sizes and the need that all datagrams be within a 32-bit boundary and multiples of 8 octets, a padding field is used to ensure that this 32-bit boundary is maintained in the datagram.

IP Datagram Format (continued)

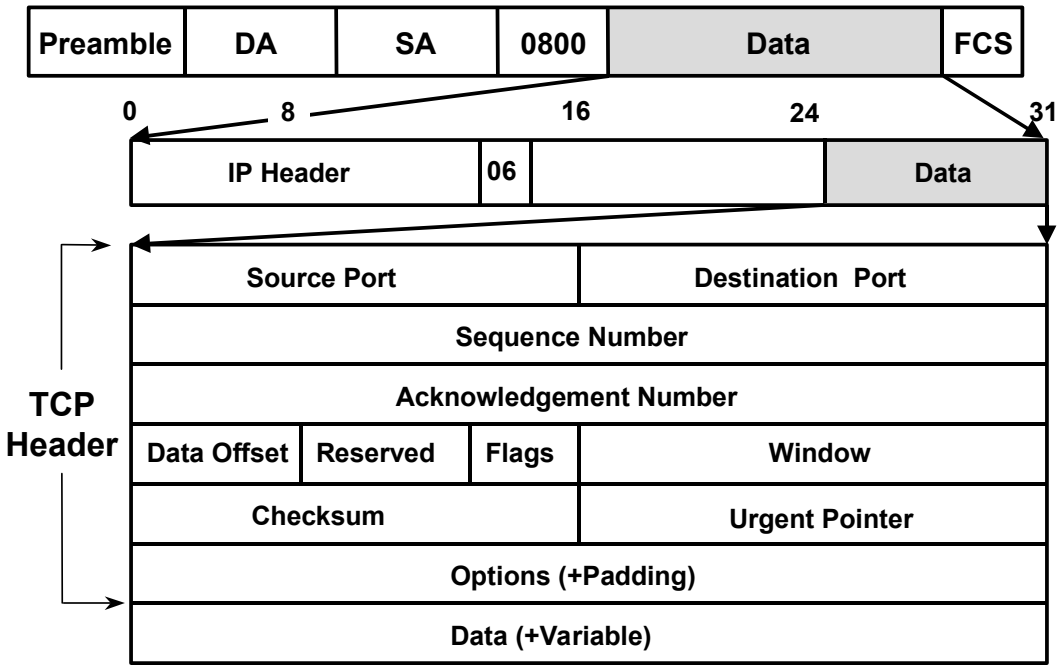


TCP Header

Description

- **Source Port—**Indicates the port number assigned to the sending process. It is typically the destination port number during a reply.
- **Destination Port—**Indicates the port number of the process in the destination host.
- **Sequence Number—**Is the sequence number of the first data octet in this segment. As an example, a source node might be sending 400 octets of data and has assigned sequence number 250 to this segment, the next segment would start at sequence number 651.
- **Acknowledgement Number—**This is the value of the next sequence number the sender of this segment is expected to receive. This is used to acknowledge receipt of all octets up to this value minus one.
- **Data Offset—**This is the number of 32 bit words in the TCP header. This is used to indicate where the data begins. If no options are used, this value must be a minimum of five.
- **Reserved—**Reserved for future use. It must be set to a value of zero.
- **Flags—**Carries a variety of control information. The field is six bits in length.
- **Window—**Indicates the number of data octets that the sender of this segment will accept.
- **Checksum—**Indicates whether the header was damaged in transit.
- **Urgent Pointer—**Points to the first urgent data octet in the packet.
- **Options—**Specifies various TCP options and is a variable length field.
- **Padding—**Used if option were present in the packet to ensure 32 bit word structure is maintained.
- **Data—**A variable length field that contains the upper layer information.

TCP Header


Preamble
DA
SA
0800
Data
FCS
0
8
16
24
31
IP Header
06
Data
TCP Header
Source Port
Destination Port
Sequence Number
Acknowledgement Number
Data Offset
Reserved
Flags
Window
Checksum
Urgent Pointer
Options (+Padding)
Data (+Variable)

User Datagram Protocol (UDP)

Description

User Datagram Protocol (UDP) is a connectionless transport layer protocol. It is an interface between IP and upper layer processes. UDP protocol ports distinguish multiple applications running on a single device from one another.

Unlike TCP, UDP adds no reliability, flow-control, or error recovery functionality to IP. UDP is simplistic; UDP headers contain fewer bytes and consume less network overhead than TCP.

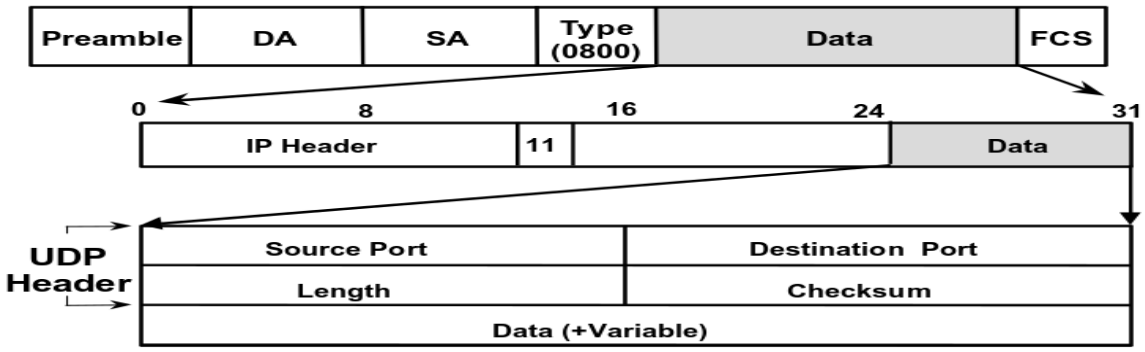
UDP is used in situations where the reliability mechanisms of TCP are not necessary, such as in cases where a higher layer protocol might provide error and flow control.

UDP is the transport protocol for several well-known application layer protocols, including SNMP (port 161), TFTP (port 69), DHCP (port 67) and RIP (port 520).

The UDP packet format is shown on the next page, it contains:

- **Source & Destination Ports—**Contain 16-bit UDP protocol port numbers used to demultiplex datagrams for receiving application layer processes. Port numbers are defined in the SERVICES file on a host.
- **Length—**Specifies the length of the UDP header and data.
- **Checksum—**Provides an (optional) integrity check on the UDP header and data.

UDP Header


Preamble
DA
SA
Type (0800)
Data
FCS
0
8
16
24
31
IP Header
11
Data
UDP Header
Source Port
Destination Port
Length
Checksum
Data (+Variable)

Internet Protocol