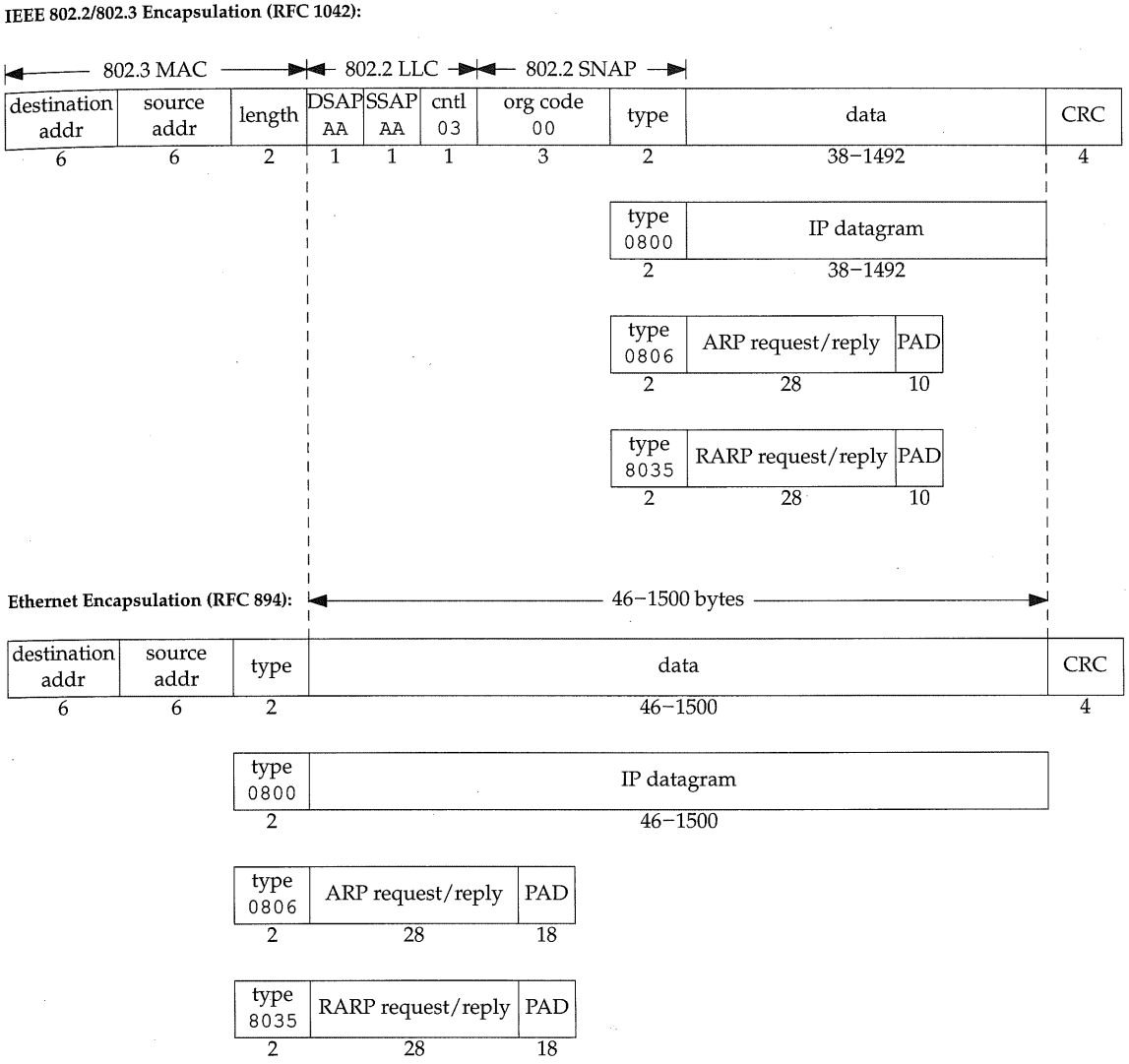


Figure 2.1 IEEE 802.2/802.3 encapsulation (RFC 1042) and Ethernet encapsulation (RFC 894).

2.3 Trailer Encapsulation

RFC 893 [Leffler and Karels 1984] describes another form of encapsulation used on Ethernets, called *trailer encapsulation*. It was an experiment with early BSD systems on DEC VAXes that improved performance by rearranging the order of the fields in the IP datagram. The variable-length fields at the beginning of the data portion of the Ethernet frame (the IP header and the TCP header) were moved to the end (right before the CRC). This allows the data portion of the frame to be mapped to a hardware page,

2.6 PPP: Point-to-Point Protocol

PPP, the Point-to-Point Protocol, corrects all the deficiencies in SLIP. PPP consists of three components.

1. A way to encapsulate IP datagrams on a serial link. PPP supports either an asynchronous link with 8 bits of data and no parity (i.e., the ubiquitous serial interface found on most computers) or bit-oriented synchronous links.
2. A *link control protocol* (LCP) to establish, configure, and test the data-link connection. This allows each end to negotiate various options.
3. A family of *network control protocols* (NCPs) specific to different network layer protocols. RFCs currently exist for IP, the OSI network layer, DECnet, and AppleTalk. The IP NCP, for example, allows each end to specify if it can perform header compression, similar to CSLIP. (The acronym NCP was also used for the predecessor to TCP.)

RFC 1331 [Simpson 1992] specifies the encapsulation method and the link control protocol. RFC 1332 [McGregor 1992] specifies the network control protocol for IP.

The format of the PPP frames was chosen to look like the ISO HDLC standard (high-level data link control). Figure 2.3 shows the format of PPP frames.

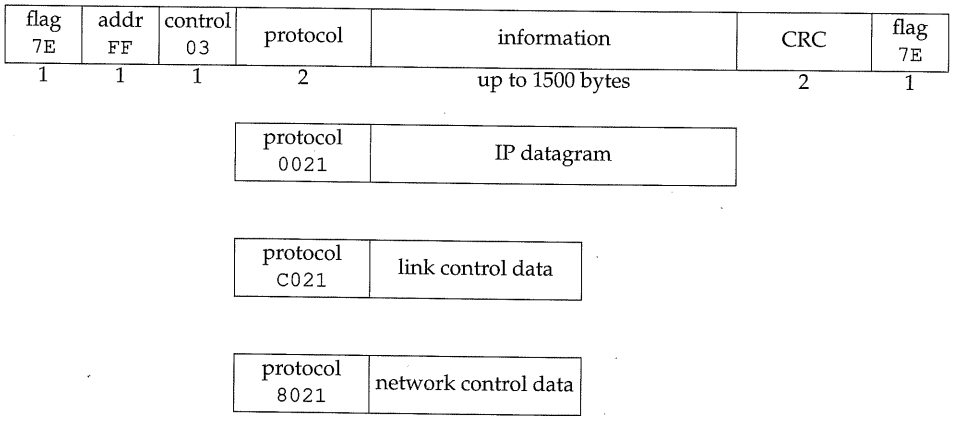
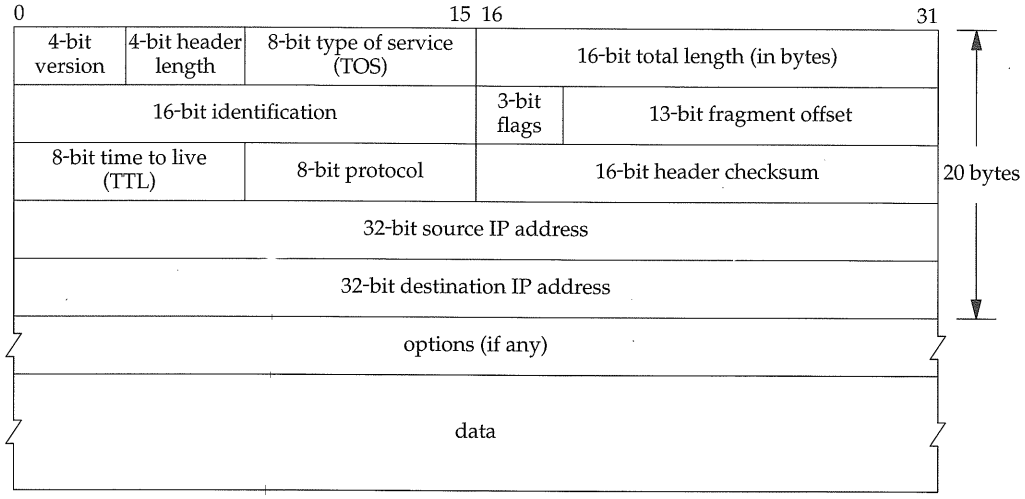


Figure 2.3 Format of PPP frames.

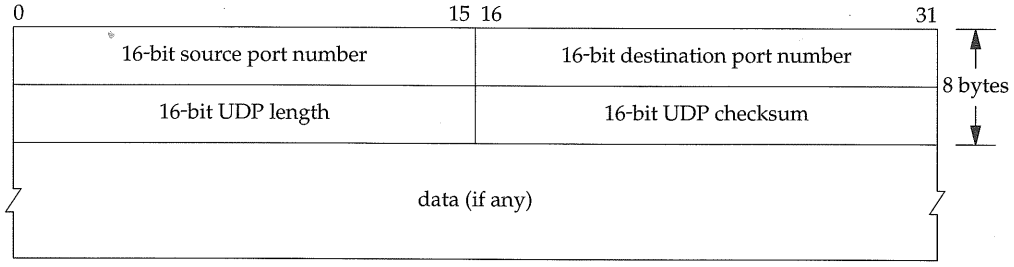
Each frame begins and ends with a *flag* byte whose value is `0x7e`. This is followed by an *address* byte whose value is always `0xff`, and then a *control* byte, with a value of `0x03`.

Next comes the *protocol* field, similar in function to the Ethernet *type* field. A value of `0x0021` means the *information* field is an IP datagram, a value of `0xc021` means the *information* field is link control data, and a value of `0x8021` is for network control data.

IP Header



UDP Header



TCP Header