Since the field length is 16 bits, *the total length of the IPv4 datagram is limited to 65,535 (2^{16} - 1) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer*. Some physical networks are not able to encapsulate a datagram of 65,535 bytes in their frames. The datagram must be ***fragmented*** to be able to pass through those networks. For example, the Ethernet protocol has a minimum and maximum restriction on the size of data that can be encapsulated in a frame (46 to 1500 bytes). If the size of an IPv4 datagram is less than 46 bytes, some padding will be added to meet this requirement, as shown in Figure 6.3.

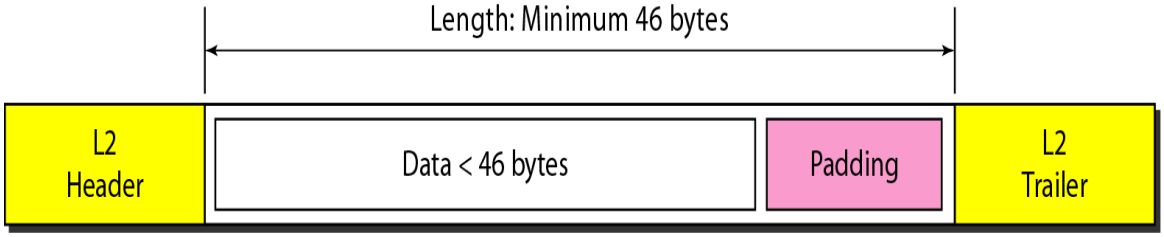

Figure 6.3 Encapsulation of a small datagram in an Ethernet frame

- ***Identification.*** *This 16-bit field identifies a datagram originating from the source* host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. *When a datagram is fragmented, the value in the identification field is copied to all fragments.*
- ***Flags.*** This is a 3-bit field. The first bit is reserved. The second bit is called the *do not fragment* bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary. The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment, as shown in Figure 6.4.

Figure 6.4 Flags used in fragmentation

- ***Fragmentation offset.*** This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure 6.5 shows a datagram with a data size of 4000 bytes fragmented into three fragments.

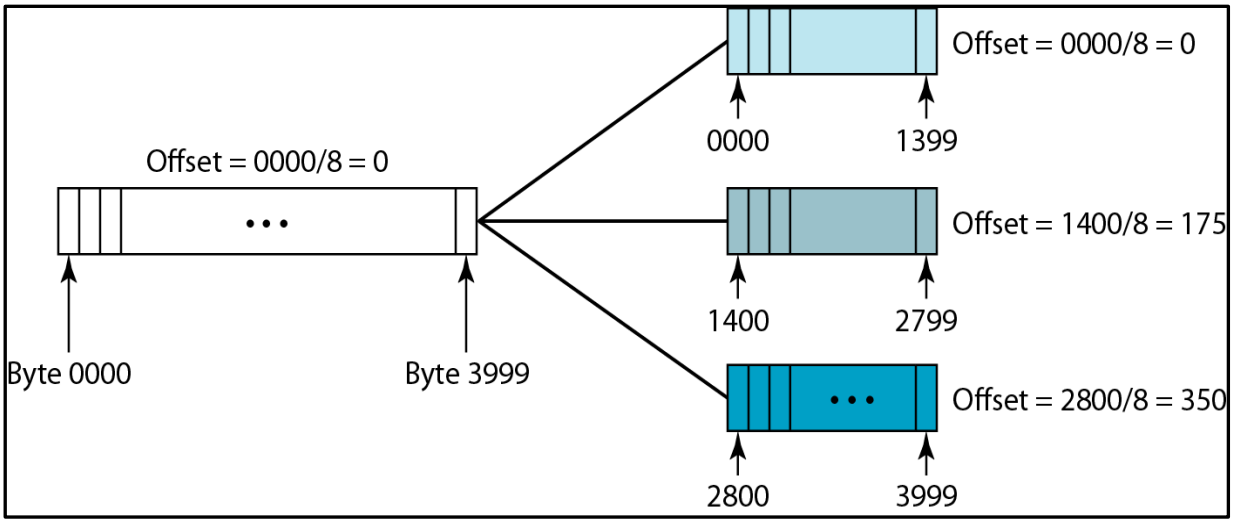

Figure 6.5 Fragmentation example

- ***Time to live.*** *This is 8-bit field*; a datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. *The datagram was discarded when the value became zero*. However, for this scheme, all the machines must have synchronized clocks and must know how long it takes for a datagram to go from one machine to another. *Today, this field is used mostly to control the maximum number of hops (routers) visited by the datagram.* When a source host sends the datagram, it stores a number in this field. This value is approximately 2 times the maximum number of routes between any two hosts. Each router that processes the datagram decrements this number by 1. If this value, after being decremented, is zero, the router discards the datagram.
- ***Protocol.*** *This 8-bit field defines the higher-level protocol* that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols

such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered, as shown in Figures 6.6. and 6.7.

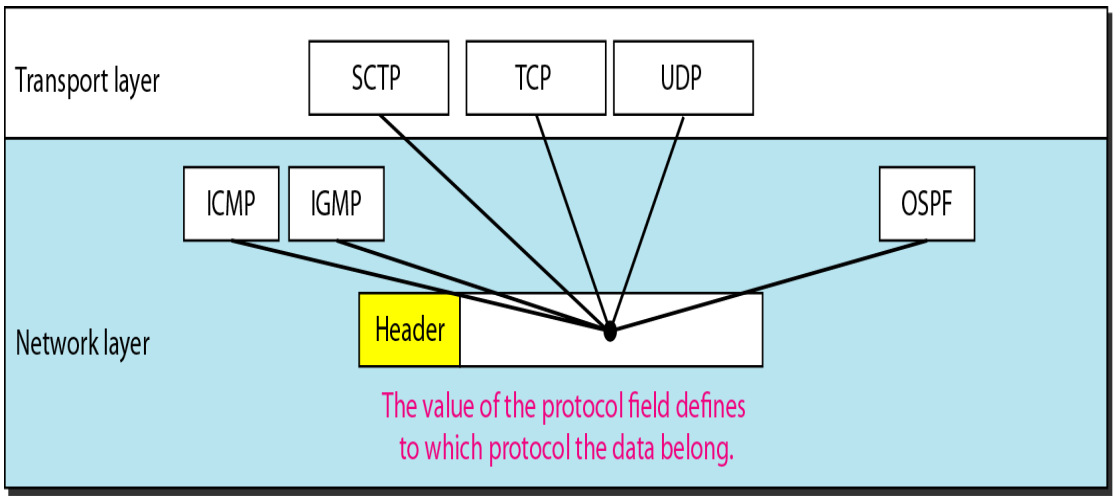

Figure 6.6 Protocol field and encapsulated data

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Figure 6.7 Protocol values

- ***Header Checksum.*** This is 16-bit field; the checksum *detects the corruption in the header of IPv4 packets*. It is carried in the IP packet header, and represents the 16-bit result of summation of the header word.

- ***Source address.*** *This 32-bit field defines the IPv4 address of the source*. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

- Destination address. *This 32-bit field defines the IPv4 address of the destination*. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- ***Option.*** The header of the IPv4 datagram is made of two parts: a fixed part and a variable part. *The fixed part is 20 bytes long* and was discussed in the previous section. ***The variable part comprises the options that can be a maximum of 40 bytes.*** Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software. This means that all implementations must be able to handle options if they are present in the header.

Example 6.1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution:

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 6.2

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution:

The HLEN value is 8, which means the total number of bytes in the header is 8×4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.