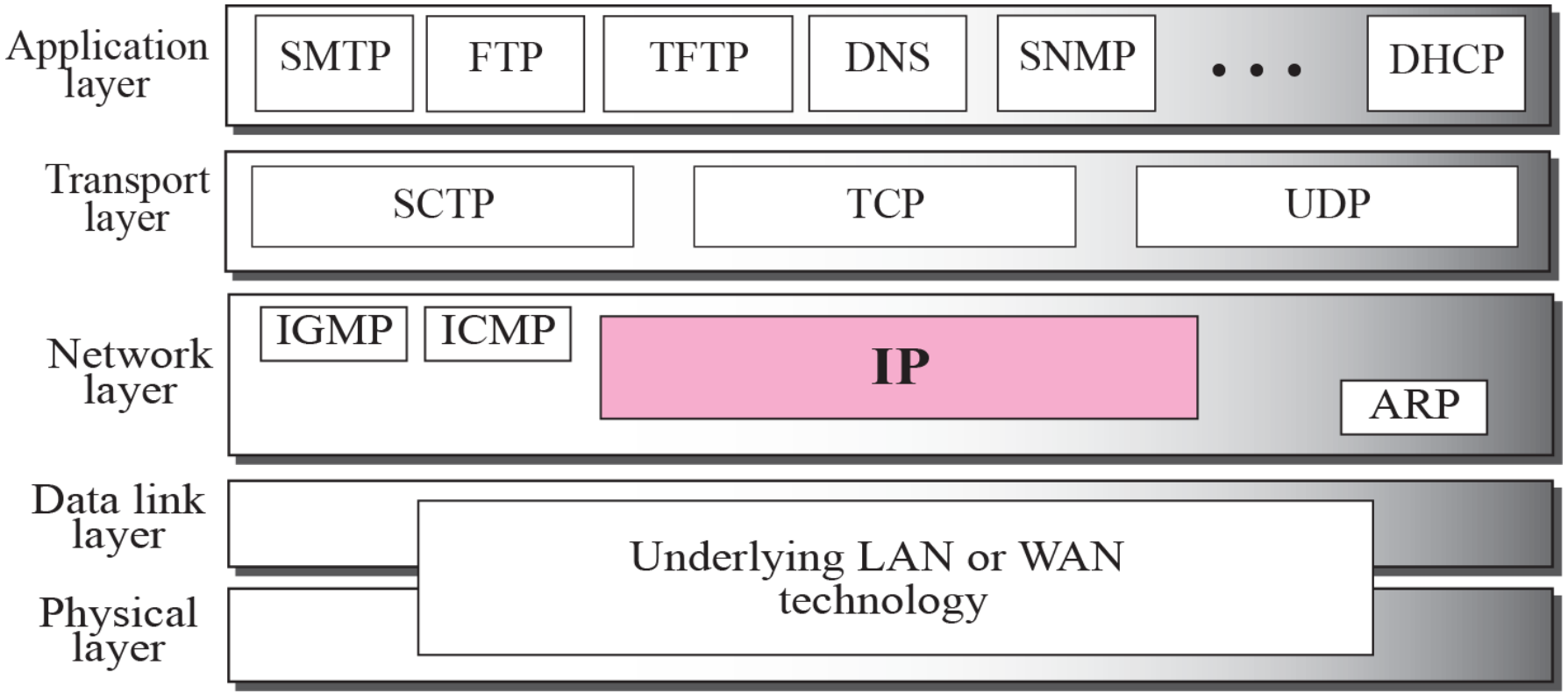
Internet Protocol Version4 (IPv4)

7-1 INTRODUCTION

- The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols at the network layer.

- IP is unreliable and connectionless protocol- **a best effort delivery service.**

Figure 7.1 *Position of IP in TCP/IP protocol suite*



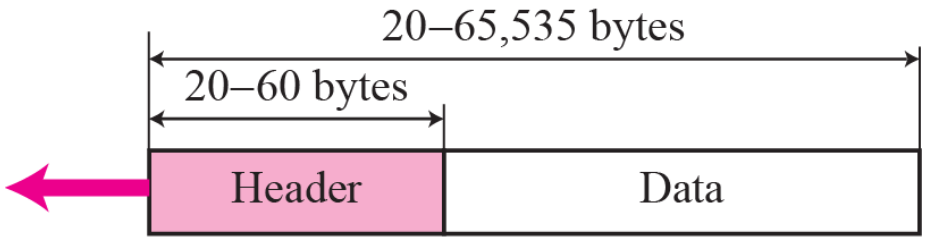
7-2 DATAGRAMS

- **Packets in the network (internet) layer are called *datagrams*. A datagram is a variable-length packet consisting of two parts: header and data.**

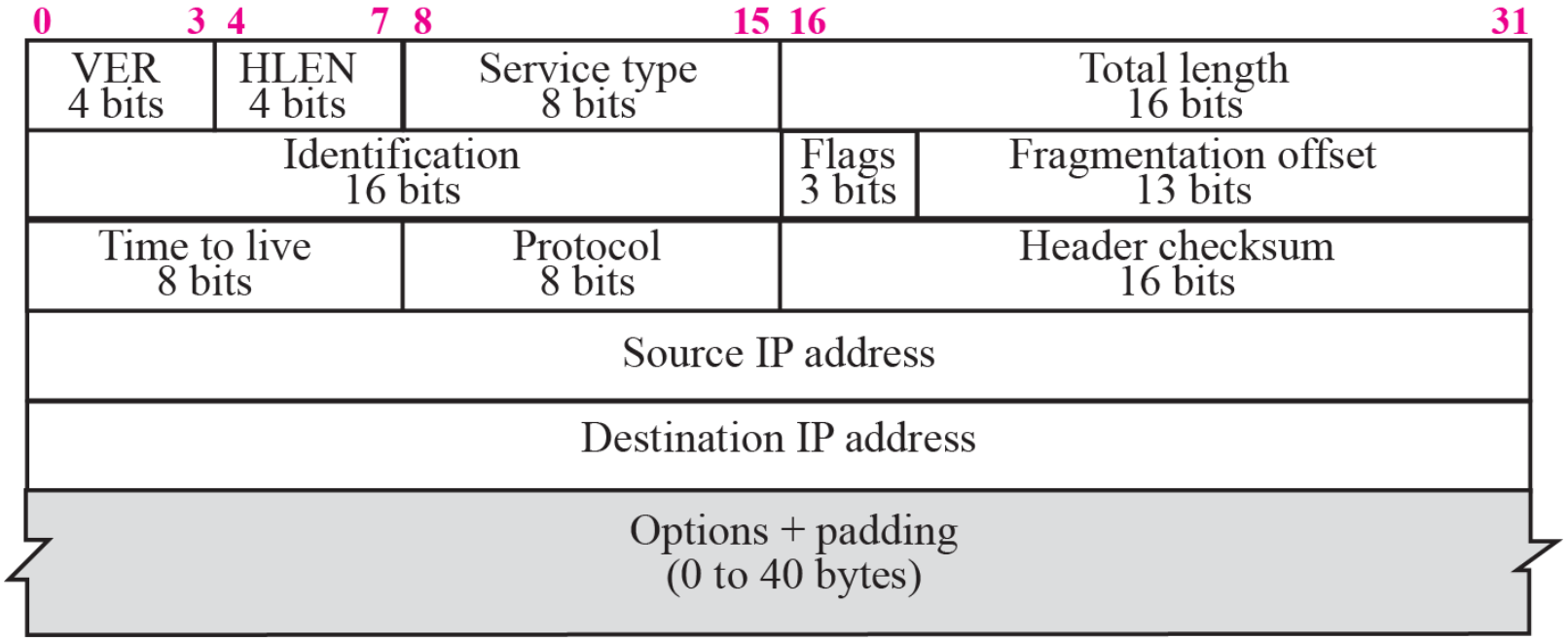
- **The header is 20 to 60 bytes in length and contains information essential to routing and delivery.**

- **It is customary in TCP/IP to show the header in 4 -byte sections. A brief description of each field is in order.**

Figure 7.2 *IP datagram*



a. IP datagram



b. Header format

1. **VER (Version):** Defines version of IP protocol
2. **HLEN (Header Length) :**defines total length of datagram
3. **Service Type (ToS or DSCP)**

TOS (Type of Service): defines how datagram should be handled

ToS Value	ToS Description
0 (000)	Routine
1 (001)	Priority
2 (010)	Immediate
3 (011)	Flash
4 (100)	Flash Override
5 (101)	CRITIC/ECP
6 (110)	Internet Control
7 (111)	Network Control

Precedence defines 8 level priority of datagram (0-7) in issues such as congestion

Figure 7.3 *Service type*

Differentiated Services:

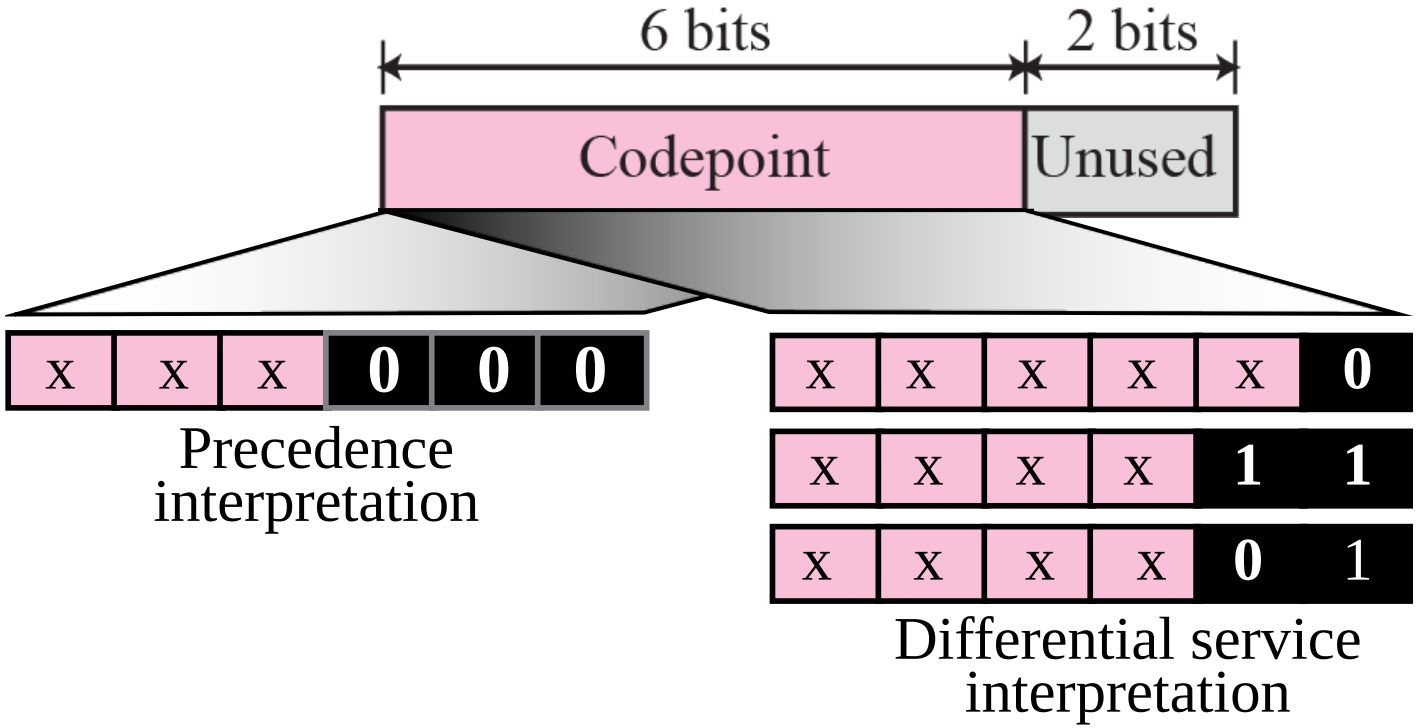


Table 7.1 *Values for codepoints*

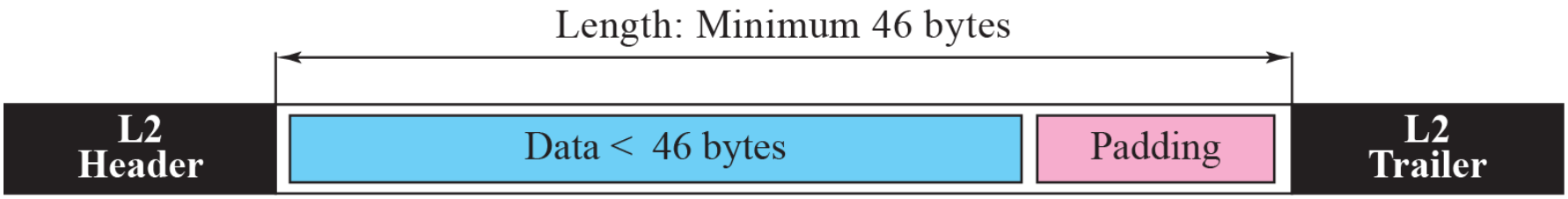
Category	*Codepoint*	*Assigning Authority*
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

Total Length

- **Total Length :** Header + data (in bytes)
- **Field length =**16 bit
- **Therefore , length of IP datagram is limited to (2 ^{16} -1) = 65,535 bytes**

The total length field defines the total length of the datagram including the header.

Figure 7.4 *Encapsulation of a small datagram in an Ethernet frame*



1. **Identification**
2. **Flags**
3. **Fragmentation Offset**
4. **TTL (Time to live)**
5. **Protocol**
6. **Source Address**
7. **Destination Address**

Figure 7.5 *Multiplexing*

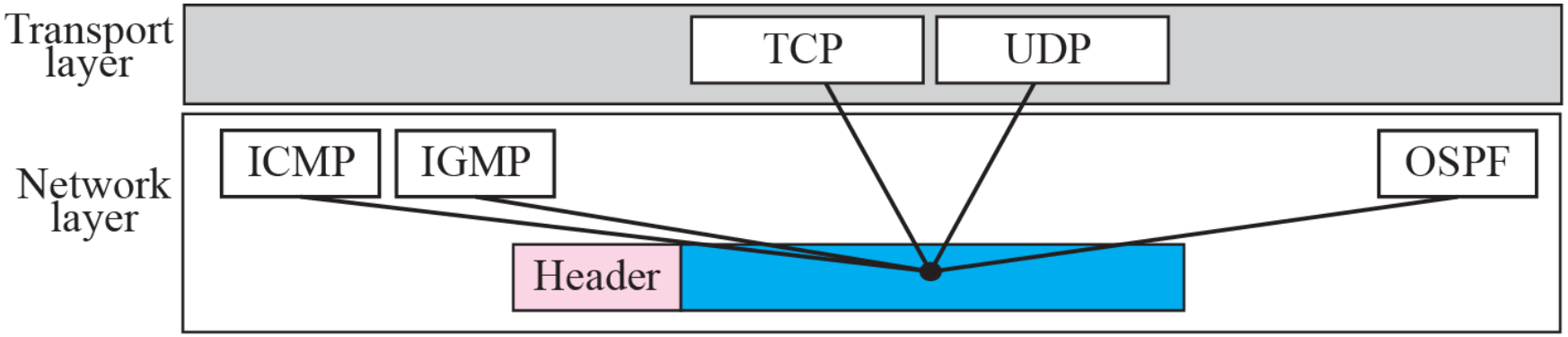


Table 7.2 *Protocols*

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

Example 7.1

An IP 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 left-most bits (0100) show the version, which is correct. The next 4 bits (0010) show the wrong 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 7.2

In an IP 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.

Example 7.3

In an IP packet, the value of HLEN is 5_{16} and the value of the total length field is 0028_{16}. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4 or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example 7.4

An IP packet has arrived with the first few hexadecimal digits as shown below:

```
45000028000100000102 . . .
```

How many hops can this packet travel before being dropped? The data belong to what upper layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes (16 hexadecimal digits). The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper layer protocol is IGMP (see Table 7.2)

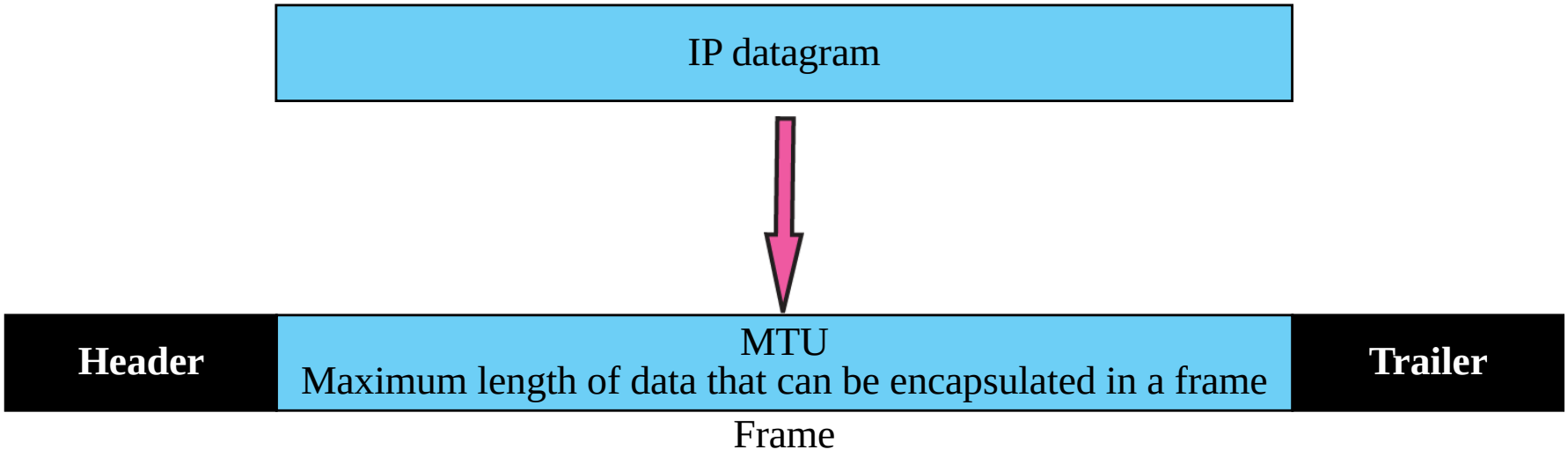
7-3 FRAGMENTATION

A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.

Topics Discussed in the Section

- ✓ **Maximum Transfer Unit (MTU)**
- ✓ **Fields Related to Fragmentation**

Figure 7.6 *MTU*



Note

Only data in a datagram is fragmented.

Figure 7.7 *Flags field*

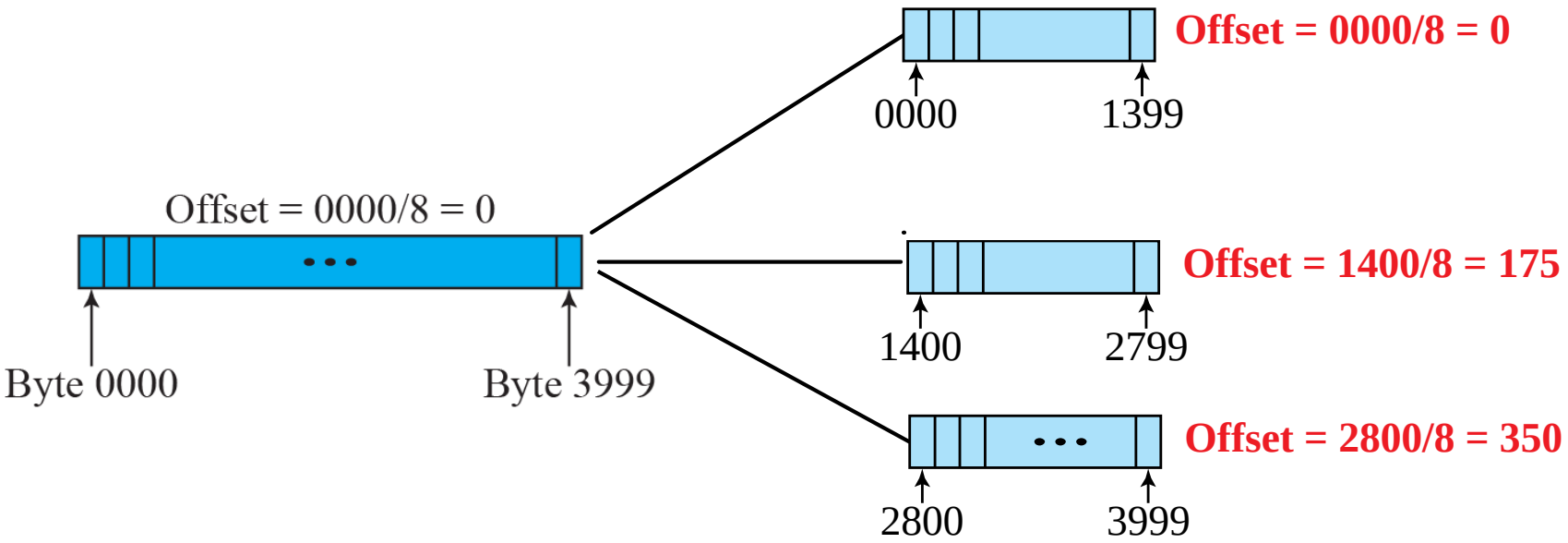
Identification Field:

- **Combination of Identification and Source IP must be unique**
- **IP protocol uses a counter to label datagrams**

Flag Fields :

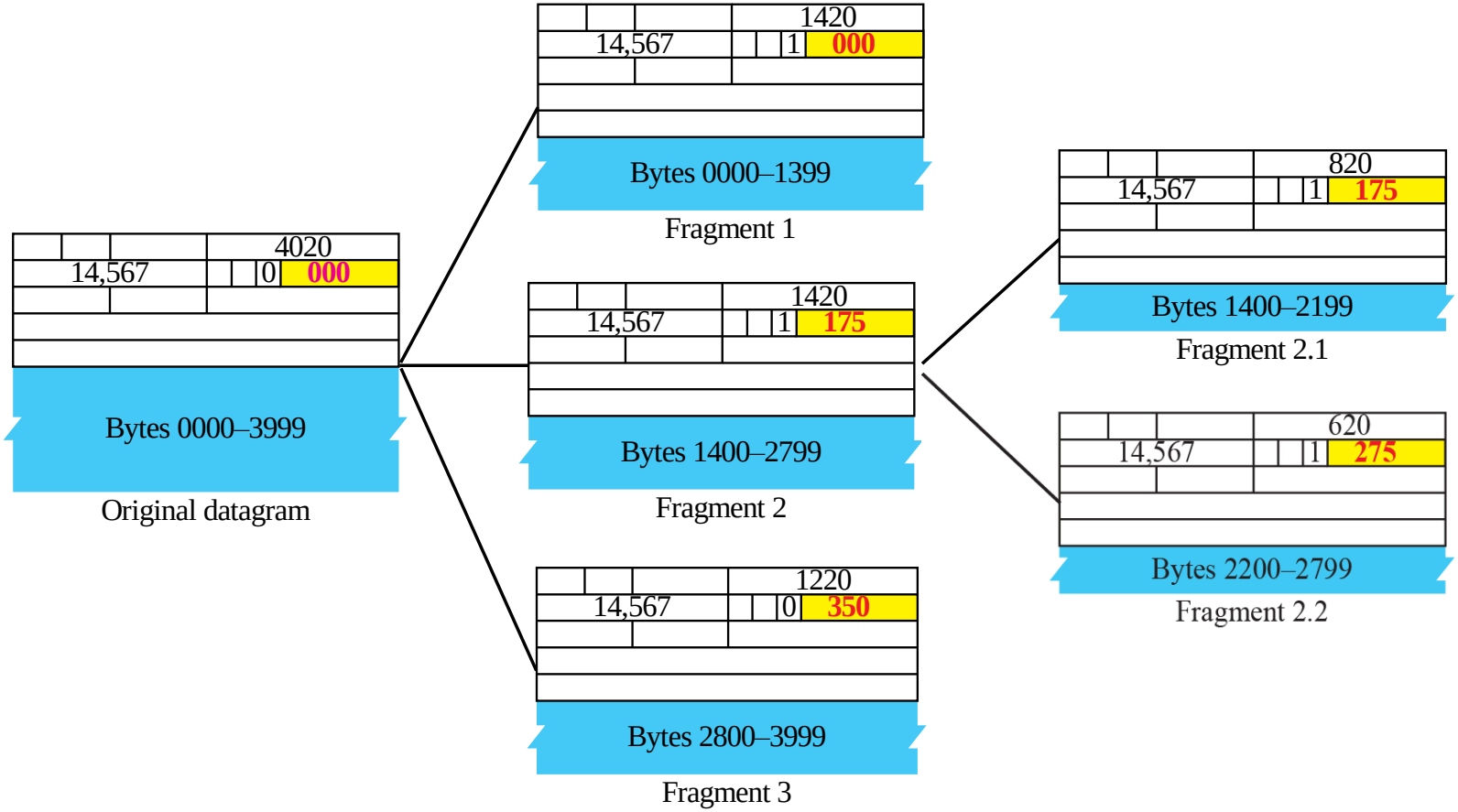
D: Do not fragment
M: More fragments

Figure 7.8 *Fragmentation example*



Fragmentation offset Field shows relative position of the fragment w.r.t whole datagram.

Figure 7.9 *Detailed fragmentation example*



Example 7.5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A nonfragmented packet is considered the last fragment.

Example 7.6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset). See also the next example.

Example 7.7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example 7.8

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

Example 7.9

A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte?

Solution

The first byte number is 100 × 8 = 800. The total length is 100 bytes and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

7-5 CHECKSUM

The error detection method used by most TCP/IP protocols is called the checksum. The checksum protects against the corruption that may occur during the transmission of a packet. It is redundant information added to the packet. The checksum is calculated at the sender and the value obtained is sent with the packet. The receiver repeats the same calculation on the whole packet including the checksum. If the result is satisfactory (see below), the packet is accepted; otherwise, it is rejected.

Topics Discussed in the Section

- ✓ **Checksum Calculation at the Sender**
- ✓ **Checksum Calculation at the Receiver**
- ✓ **Checksum in the Packet**

Figure 7.22 *Checksum concept*

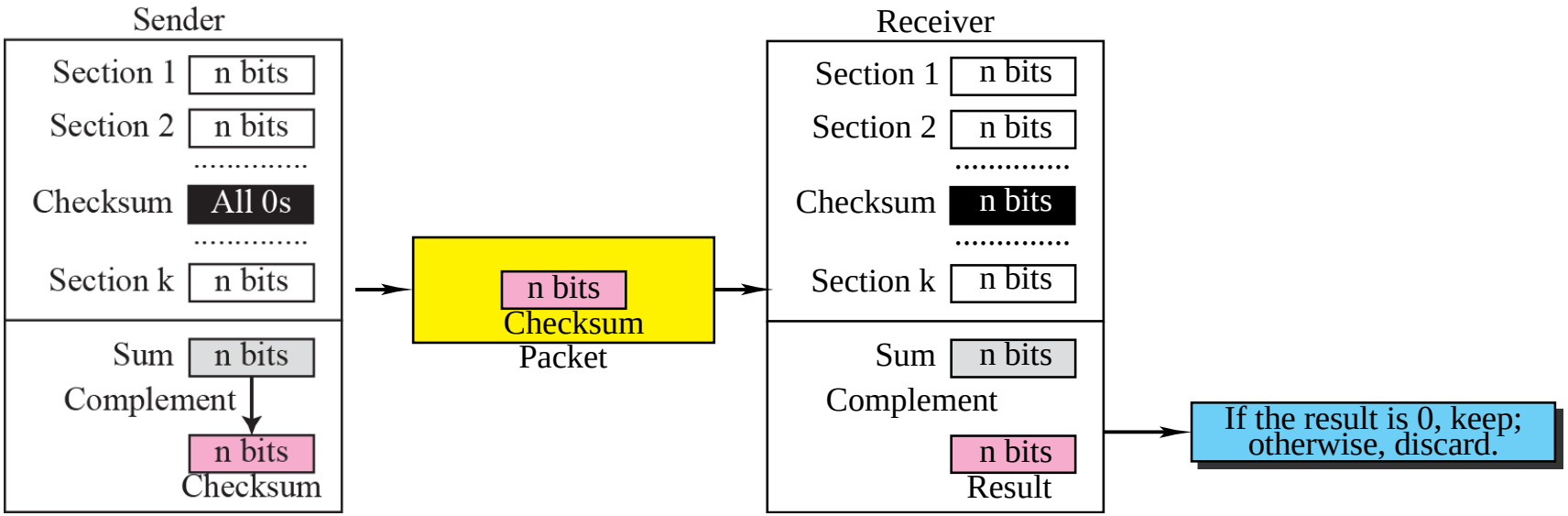
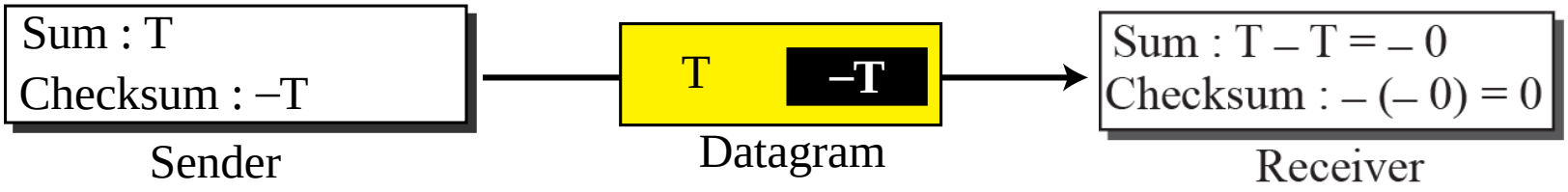


Figure 7.23 *Checksum in one's complement arithmetic*



Note

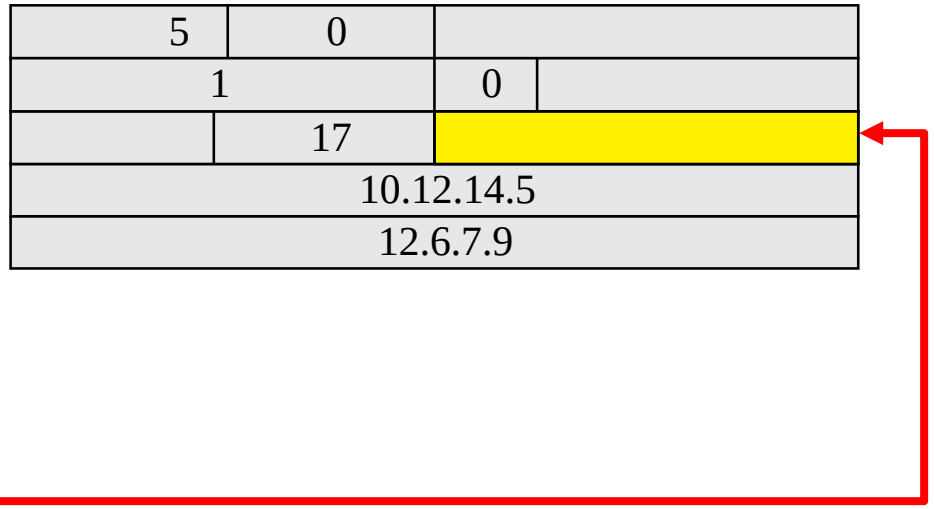
Checksum in IP covers only the header, not the data.

Example 7.17

Figure 7.24 shows an example of a checksum calculation at the sender site for an IP header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

Figure 7.24 *Example of checksum calculation at the sender*

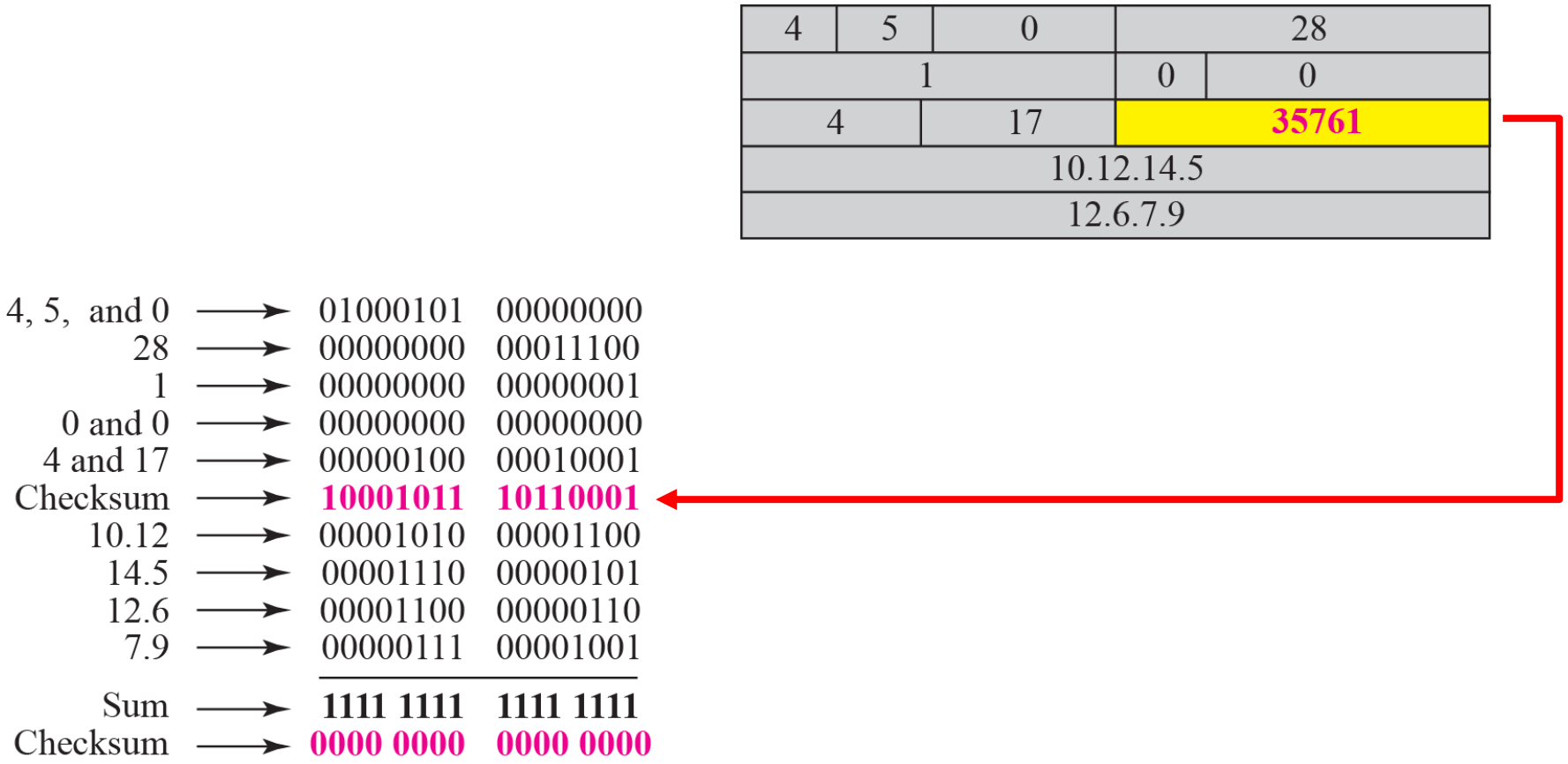
4, 5, and 0	01000101	00000000
28	00000000	00011100
1	00000000	00000001
0 and 0	00000000	00000000
4 and 17	00000100	00010001
0	00000000	00000000
10.12	00001010	00001100
14.5	00001110	00000101
12.6	00001100	00000110
7.9	00000111	00001001
Sum	**01110100**	**01001110**
Checksum	**10001011**	**10110001**



Example 7.18

Figure 7.25 shows the checking of checksum calculation at the receiver site (or intermediate router) assuming that no errors occurred in the header. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. Since the result is 16 0s, the packet is accepted.

Figure 7.25 *Example of checksum calculation at the receiver*



Note

Appendix D gives an algorithm for checksum calculation.