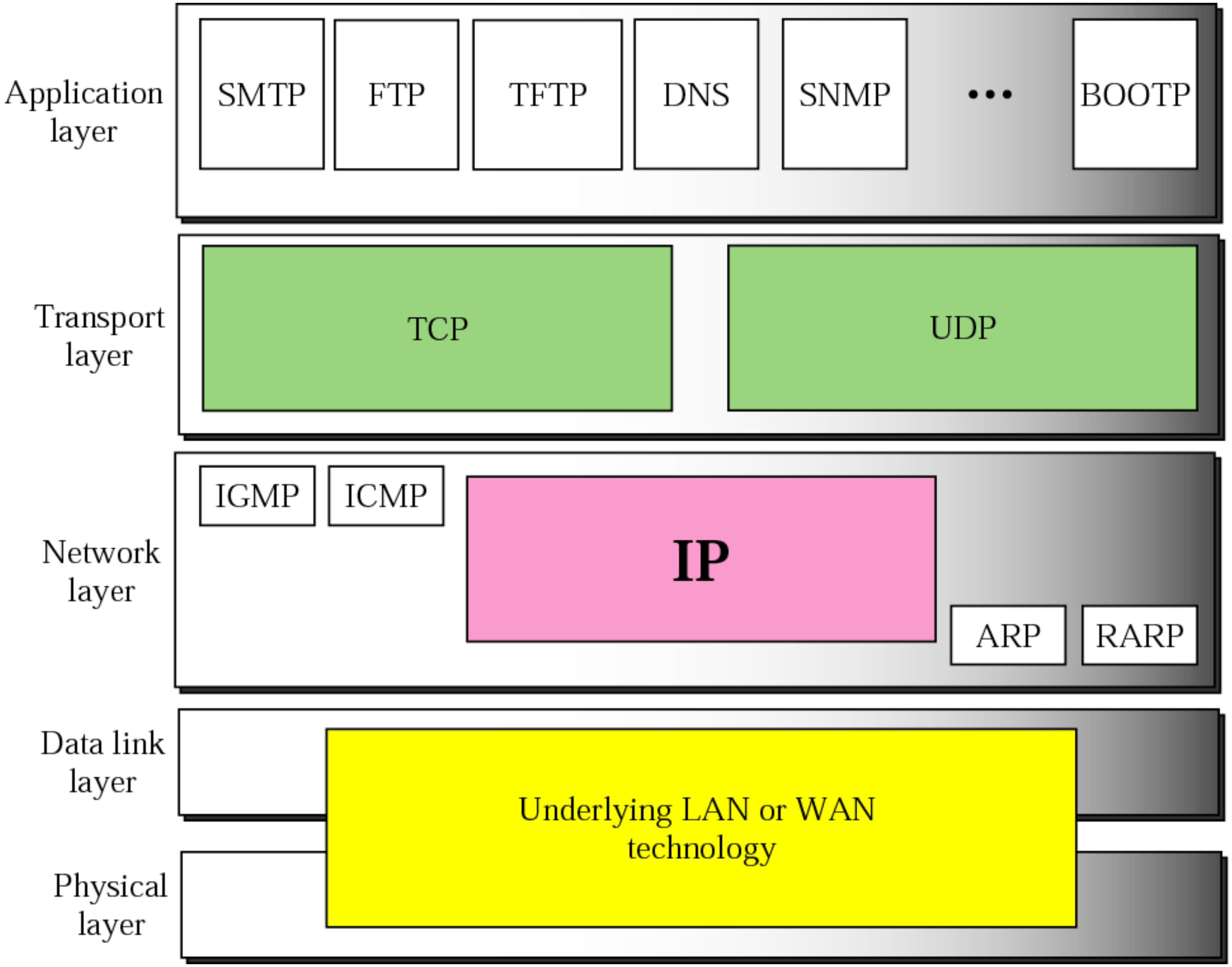
Chapter 8

Internet Protocol (IP)

CONTENTS

- **DATAGRAM**
- **FRAGMENTATION**
- **OPTIONS**
- **CHECKSUM**
- **IP PACKAGE**

Figure 8-1 Position of IP in TCP/IP protocol suite



8.1

DATAGRAM

Figure 8-2

IP datagram

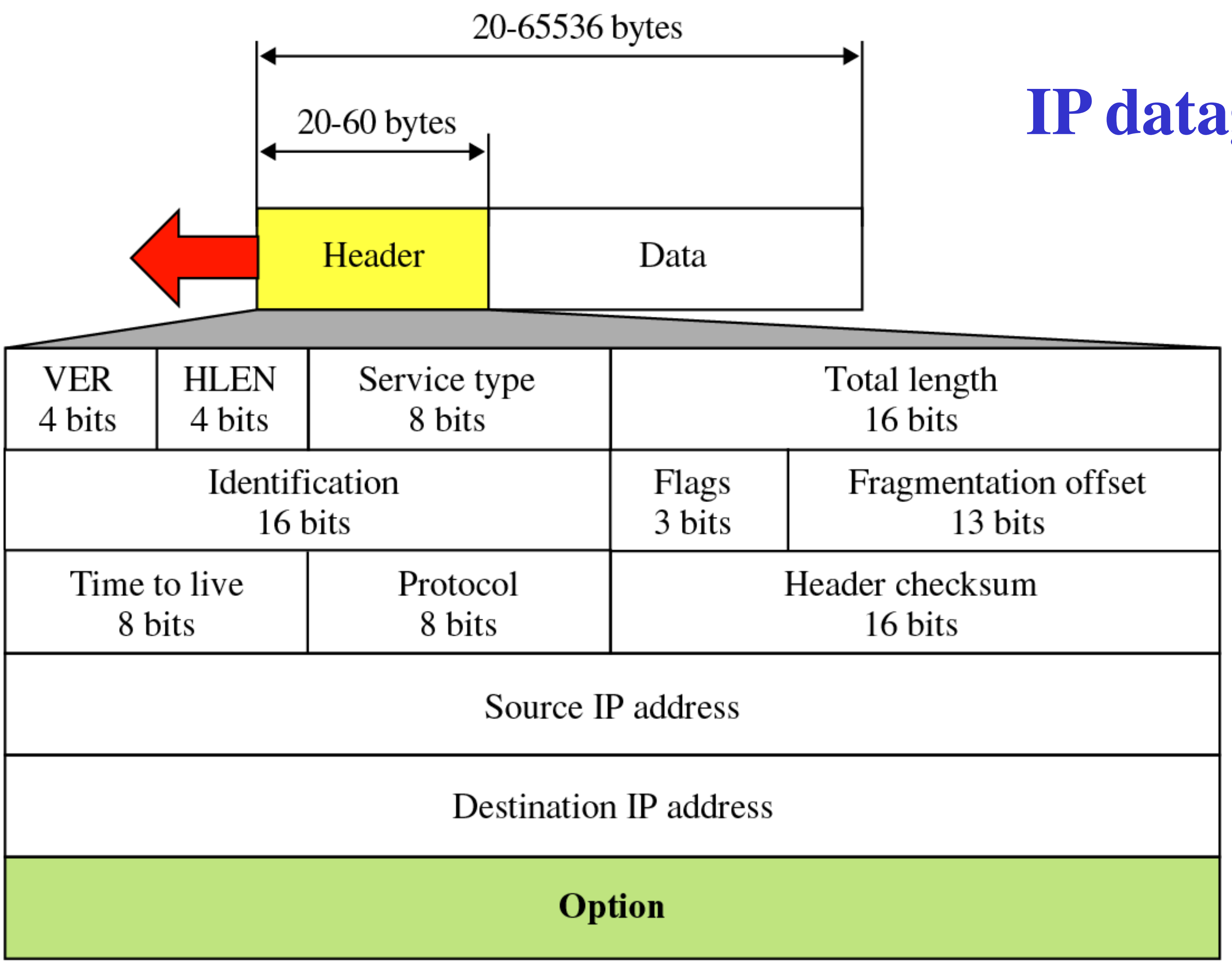
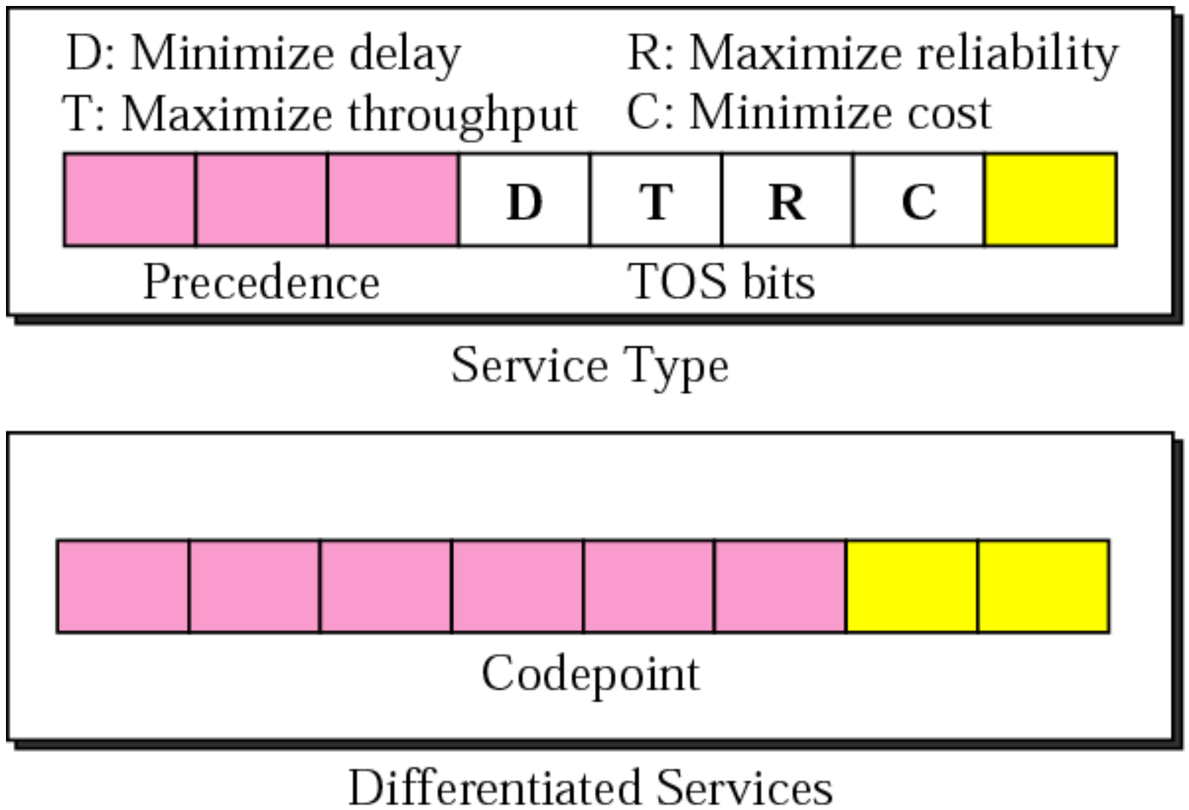


Figure 8-3

Service Type or Differentiated Services



Differentiated service (DS)

- xxx000
  - 3 left-most bits interpreted as the precedence bits in Service Type interpretation
- Else
  - Define 64 services
    - xxxxx0: Internet category - 32 services (0, 2, 4 … 62)
    - xxxx11: Local category – 16 services (3, 7, 11, … 63)
    - xxxx01: Temporary or exprimental – 16 services (1, 5, 9, … 61)

Note

The precedence subfield is not used in version 4.

Note

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

Figure 8-4

Encapsulation of a small datagram in an Ethernet frame

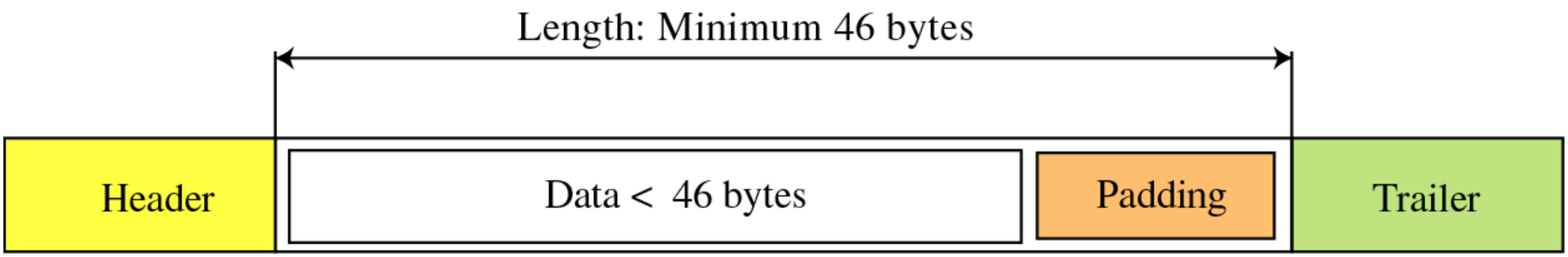
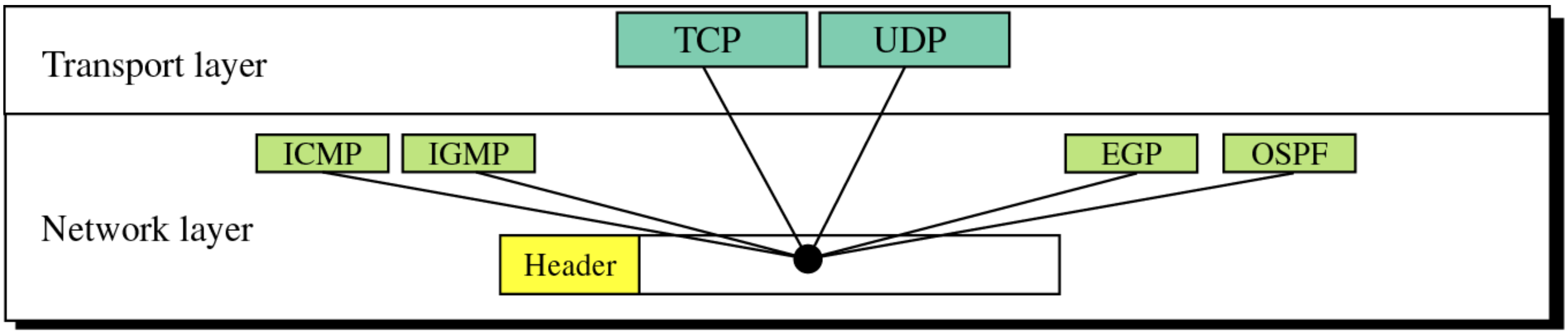


Figure 8-5

Multiplexing



Example 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 header length, which means ($2 \times 4 = 8$), which is wrong. The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 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 main header, the next 12 bytes are the options.

Example 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 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 should 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.

8.2

FRAGMENTATION

Figure 8-6

MTU

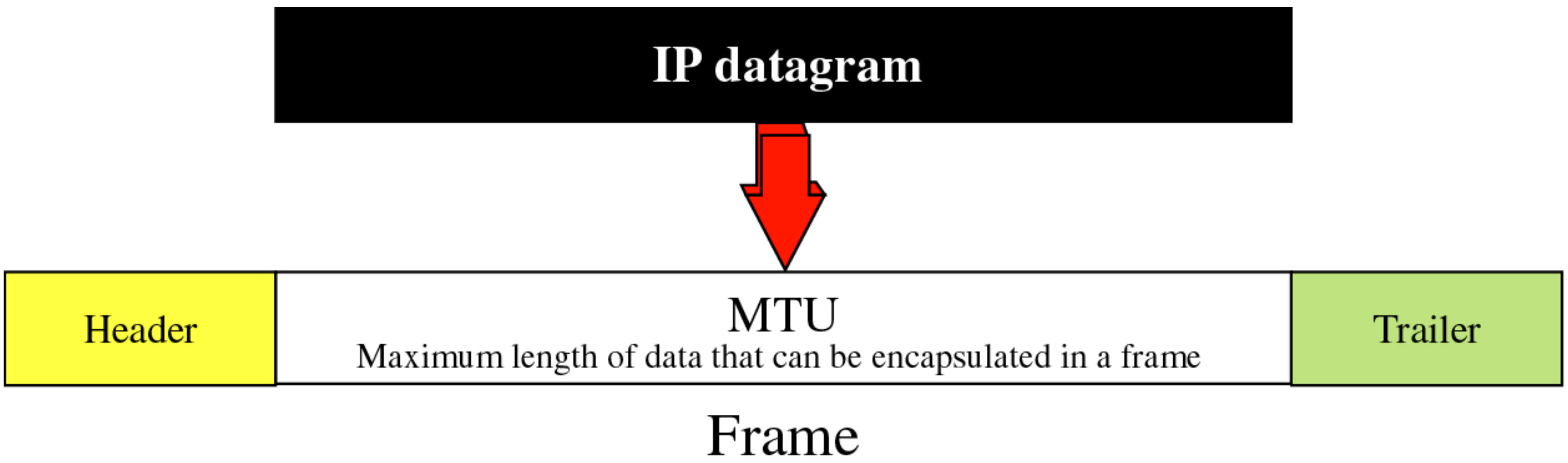


Figure 8-7

Flag field

D: Do not fragment

M: More fragments

Figure 8-8

Fragmentation example

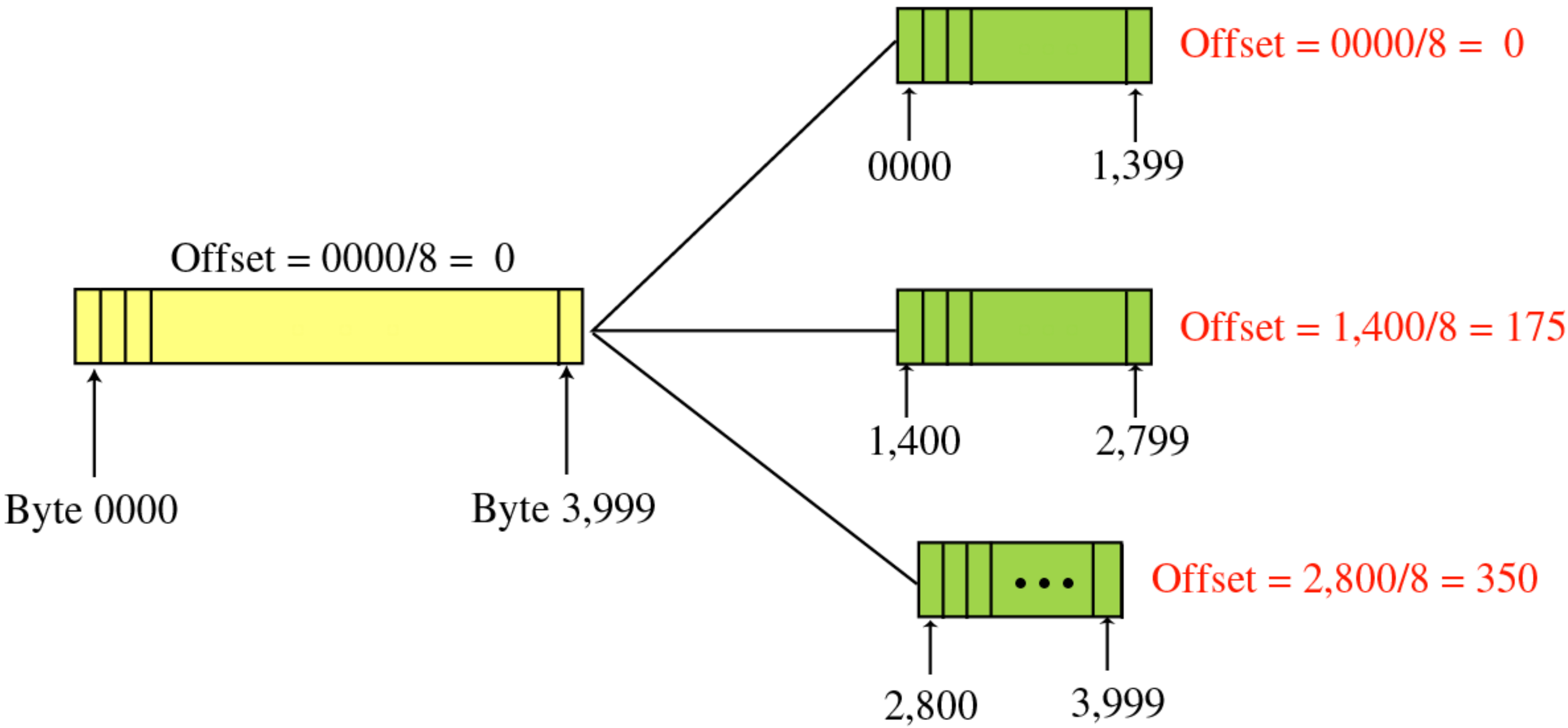
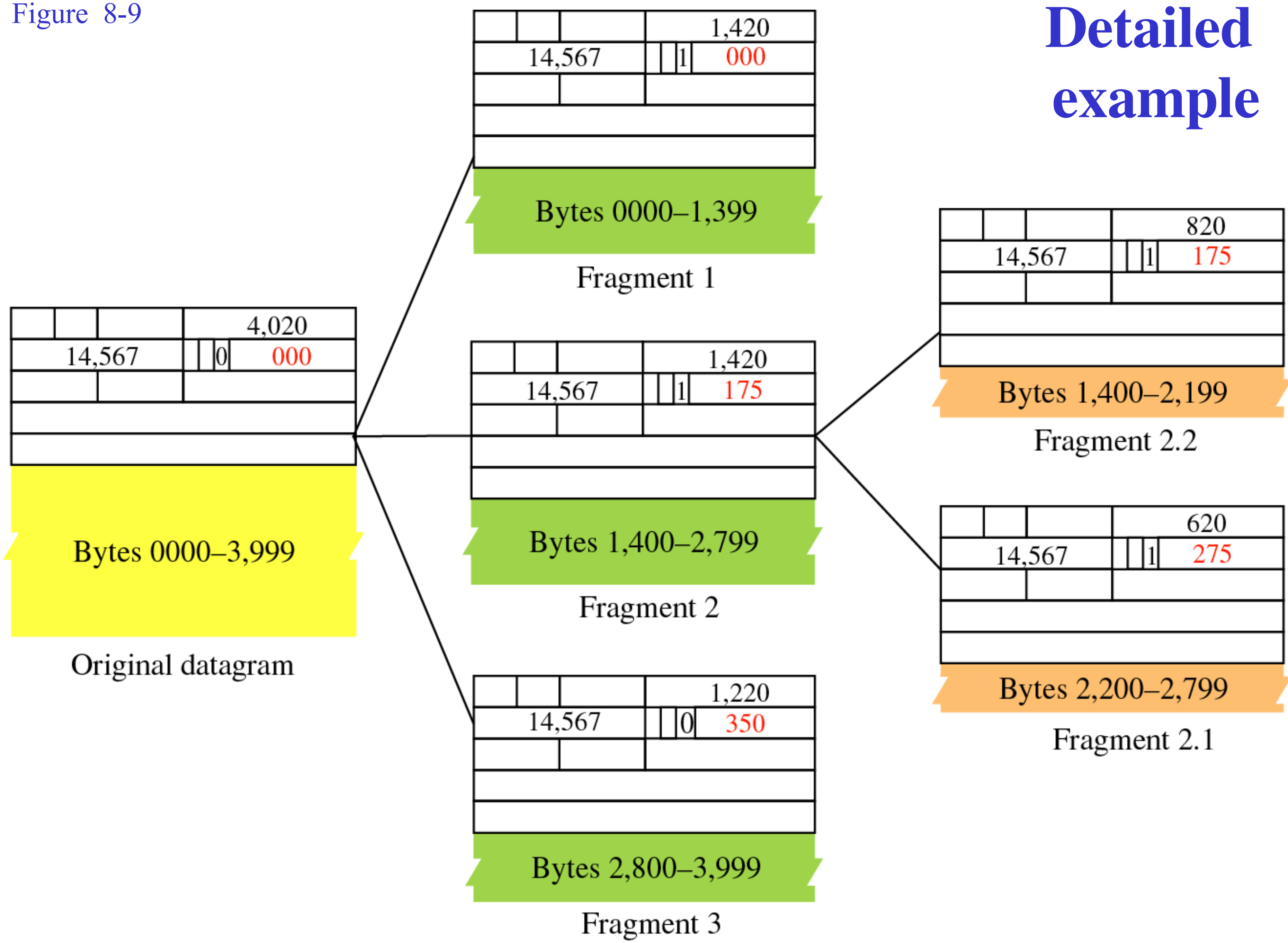


Figure 8-9

Detailed example



Example 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 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). However, we can definitely say the original packet has been fragmented because the *M* bit value is 1.

Example 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 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 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

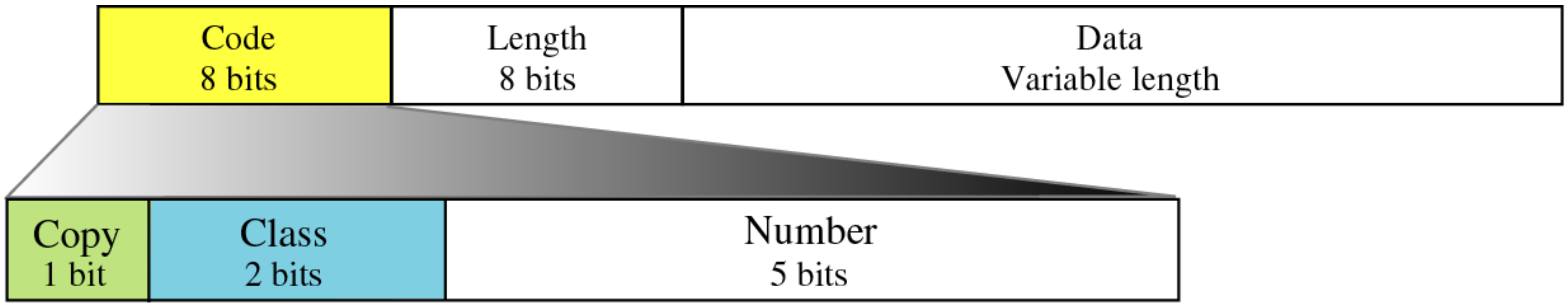
The first byte number is $100 \times 8 = 800$. The total length is 100 bytes and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must 879.

8.3

OPTIONS

Figure 8-10

Option format



Copy

0 Copy only in first fragment
1 Copy into all fragments

Class

00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

Number

00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

Figure 8-11

Categories of options

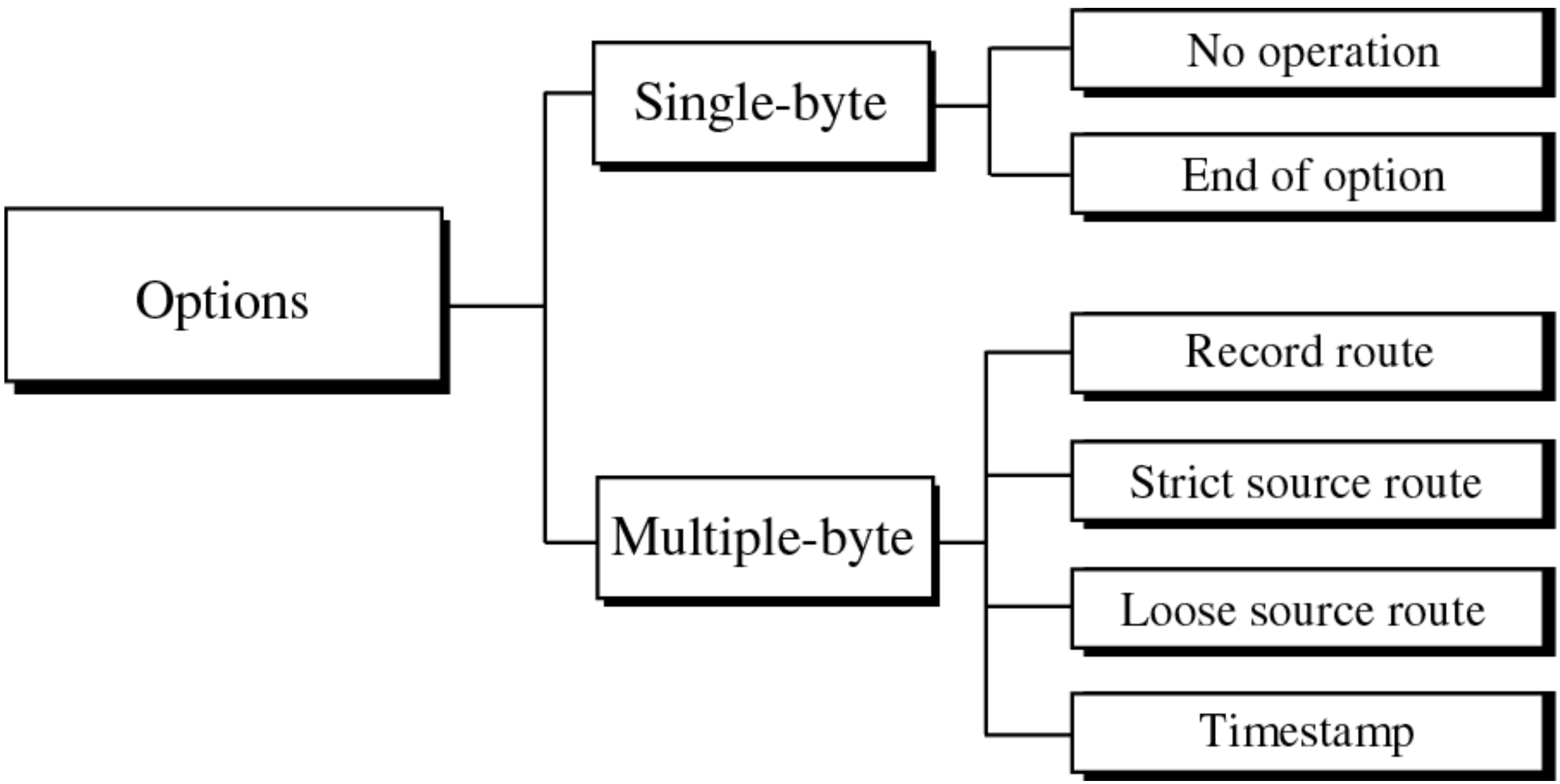
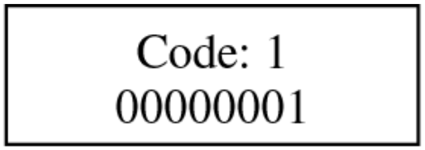
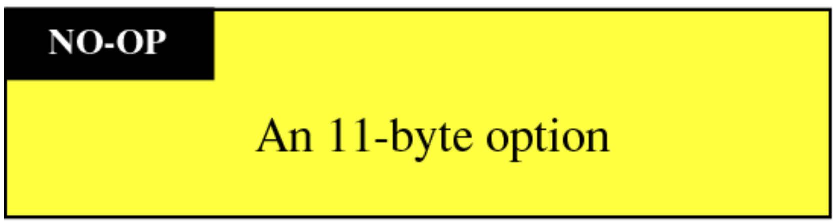


Figure 8-12

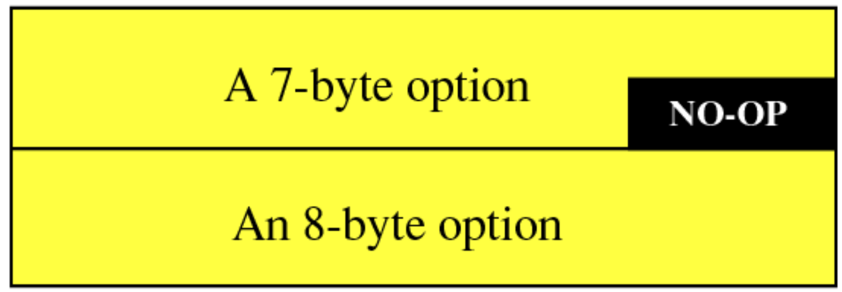
No operation option



a. No operation option



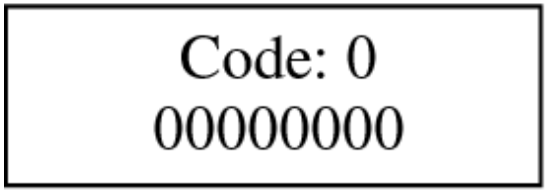
b. Used to align beginning of an option



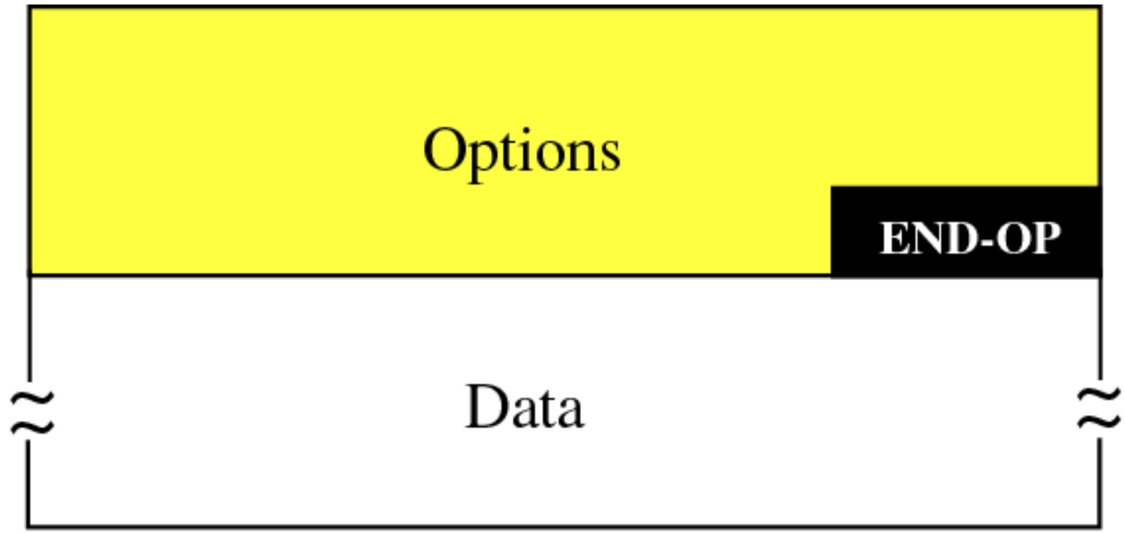
c. Used to align the next option

Figure 8-13

End of option option



a. End of option



b. Used for padding

Figure 8-14

Record route option

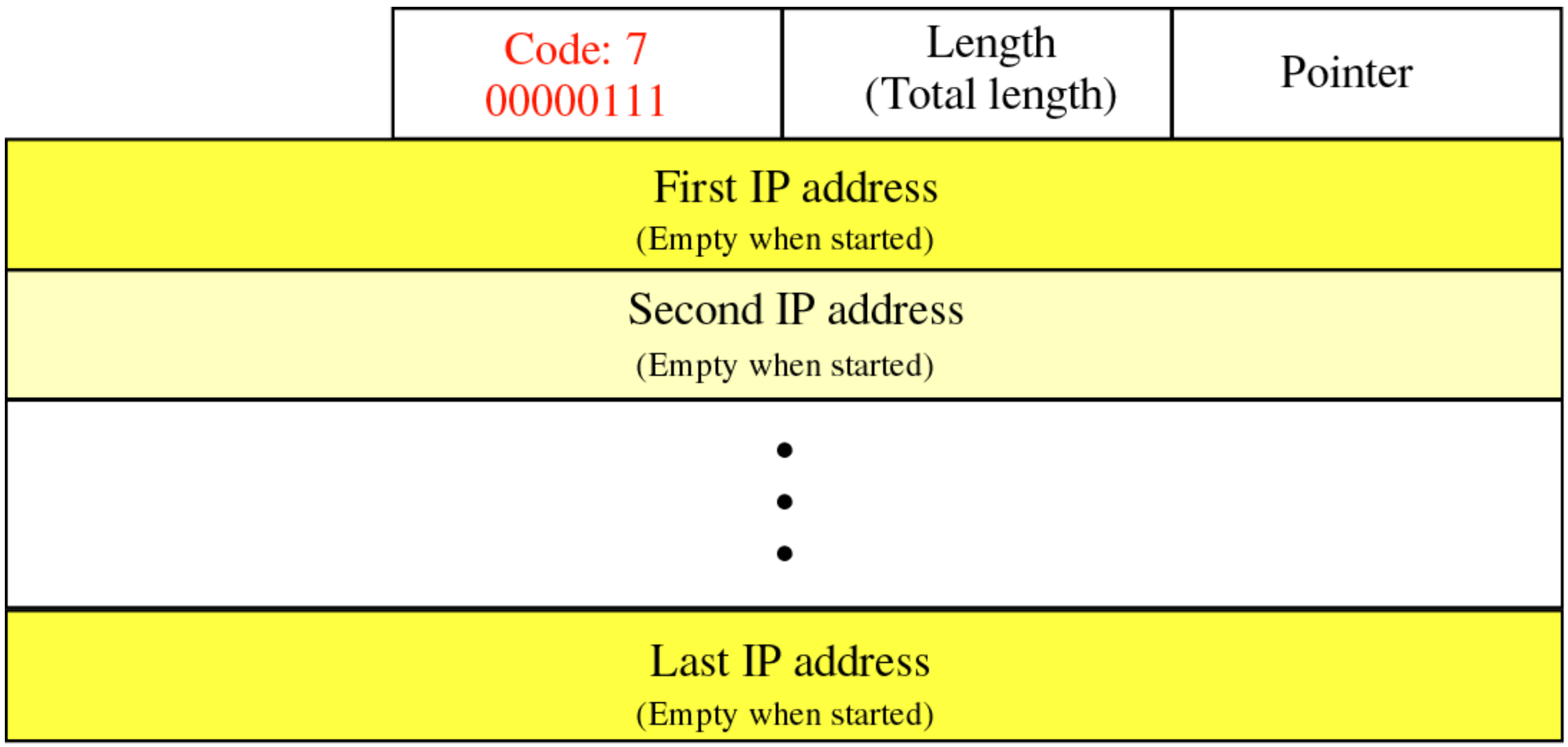


Figure 8-15

Record route concept

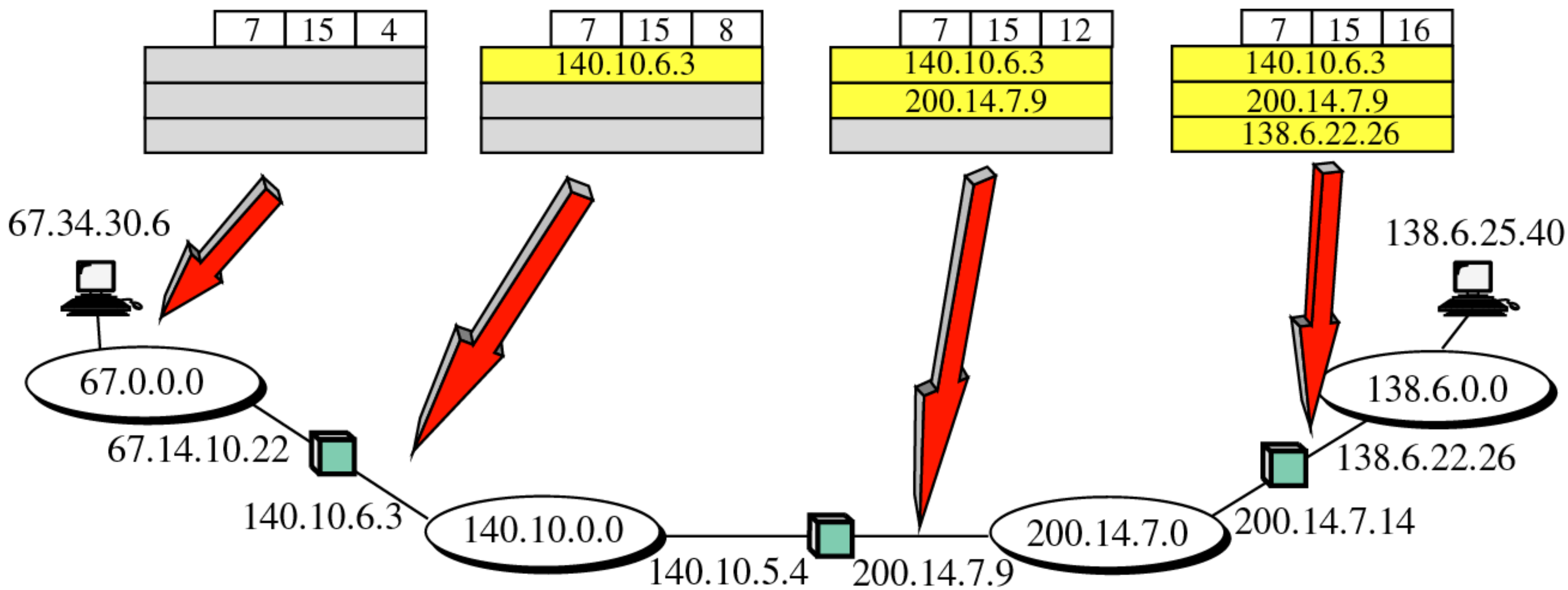


Figure 8-16

Strict source route option

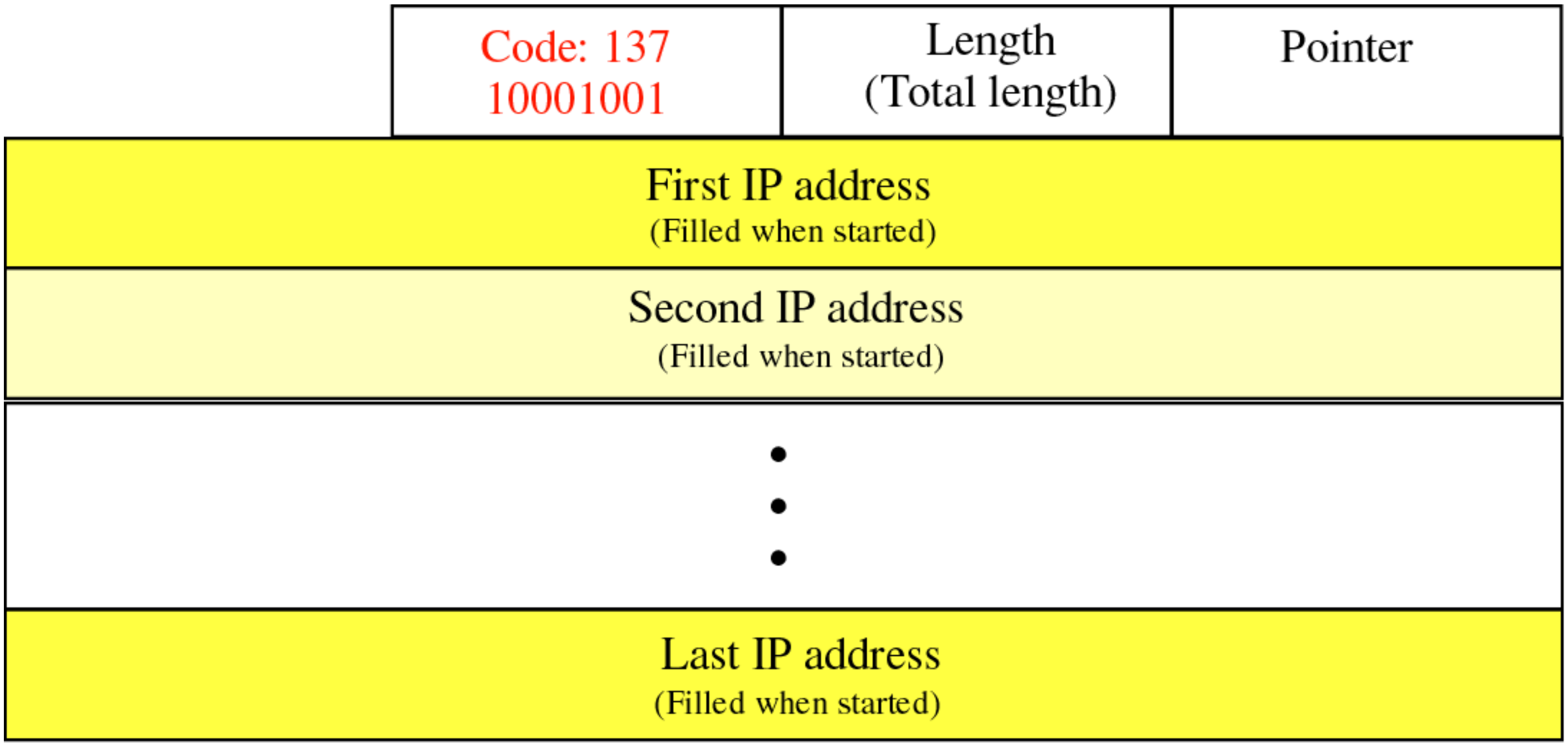


Figure 8-17

Strict source route concept

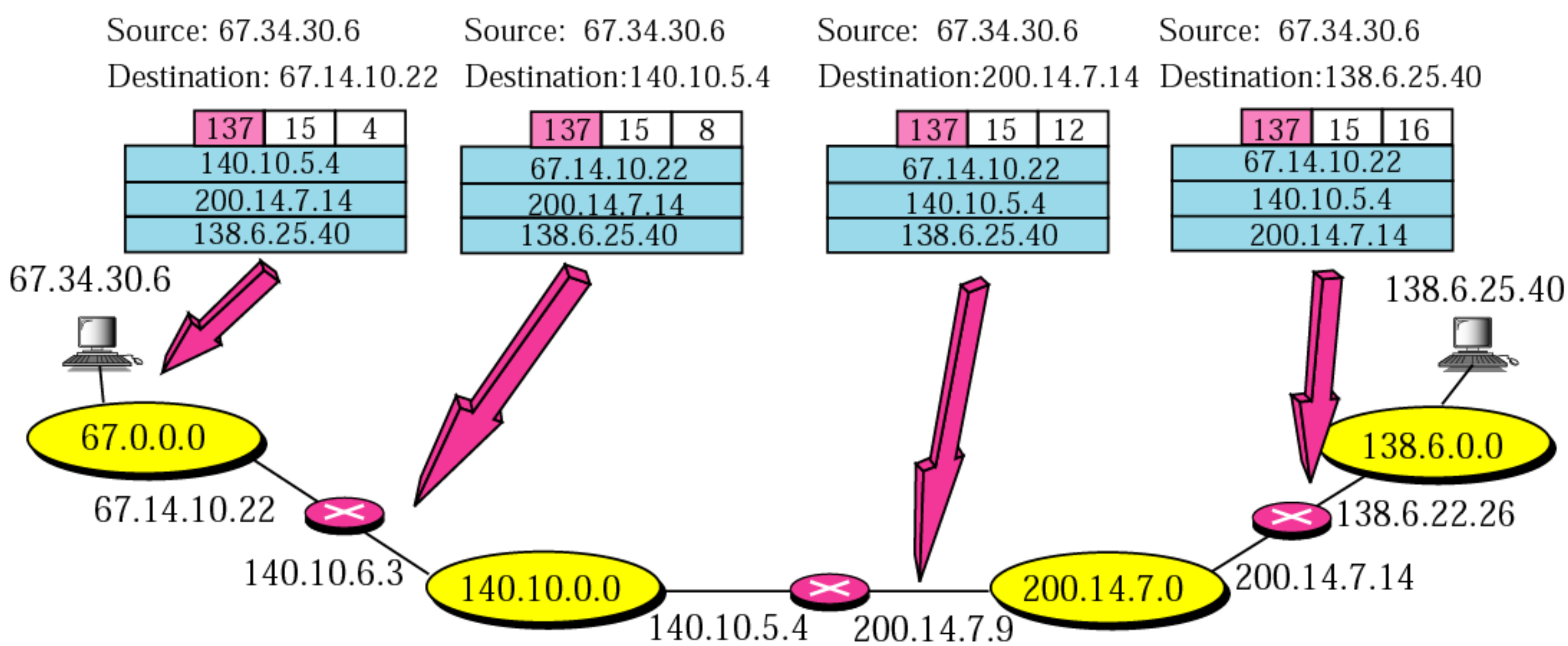


Figure 8-18

Loose source route option

	Code: 131 10000011	Length (Total length)	Pointer
First IP address (Filled when started)			
Second IP address (Filled when started)			
• • •			
Last IP address (Filled when started)			

Figure 8-19

Timestamp option

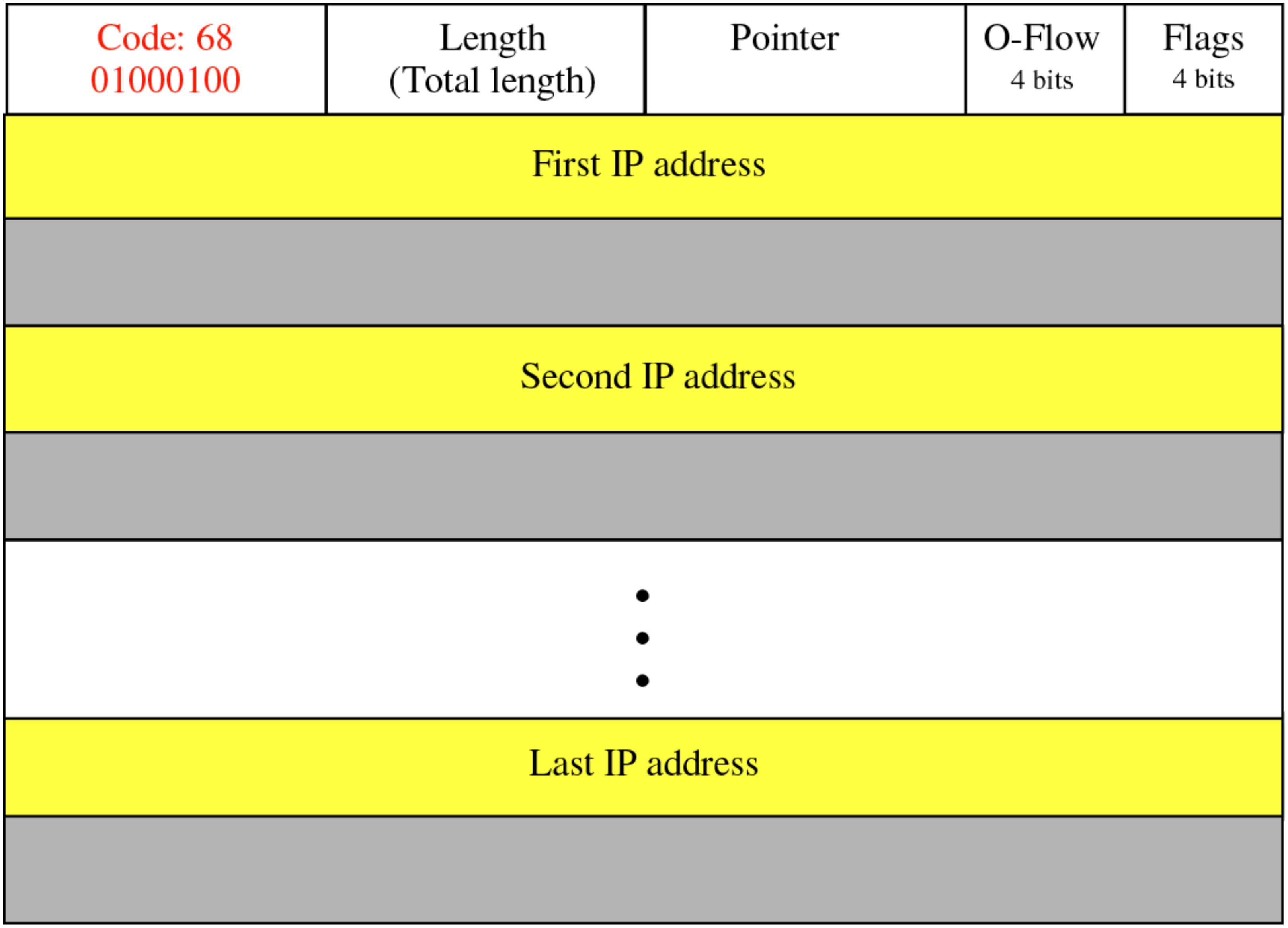


Figure 8-20

Use of flag in timestamp

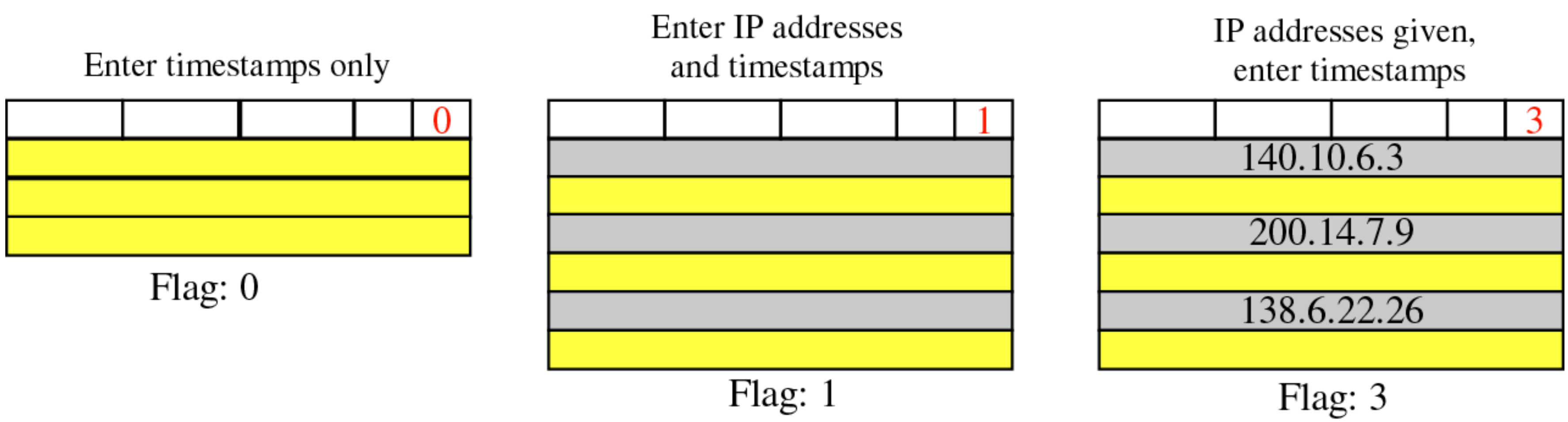
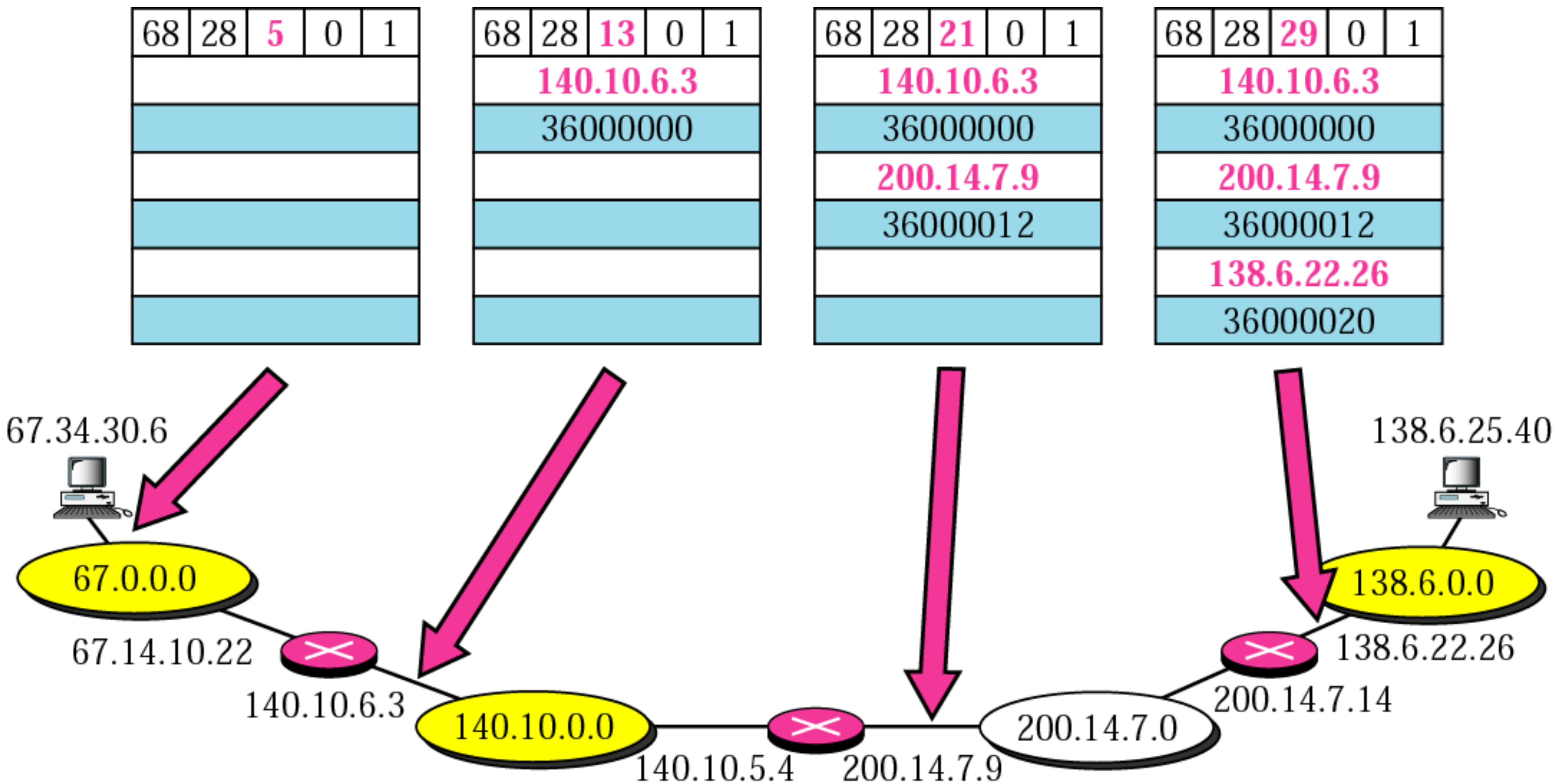


Figure 8-21

Timestamp concept



Example 10

Which of the six options must be copied to each fragment?

Solution

We look at the first (left-most) bit of the code for each option.

No operation: Code is **0**0000001; no copy.

End of option: Code is **0**0000000; no copy.

Record route: Code is **0**0000111; no copy.

Strict source route: Code is **1**0001001; copied.

Loose source route: Code is **1**0000011; copied.

Timestamp: Code is **0**1000100; no copy.

Example 11

Which of the six options are used for datagram control and which are used for debugging and management?

Solution

We look at the second and third (left-most) bits of the code.

No operation: Code is 0**00**00001; control.

End of option: Code is 0**00**00000; control.

Record route: Code is 0**00**00111; control.

Strict source route: Code is 1**00**01001; control.

Loose source route: Code is 1**00**00011; control.

Timestamp: Code is 0**10**00100; debugging

8.4

CHECKSUM

Note

To create the checksum the sender does the following:

1. ***The packet is divided into k sections, each of n bits.***
2. ***All sections are added together using one's complement arithmetic.***
3. ***The final result is complemented to make the checksum.***

Figure 8-22

Checksum concept

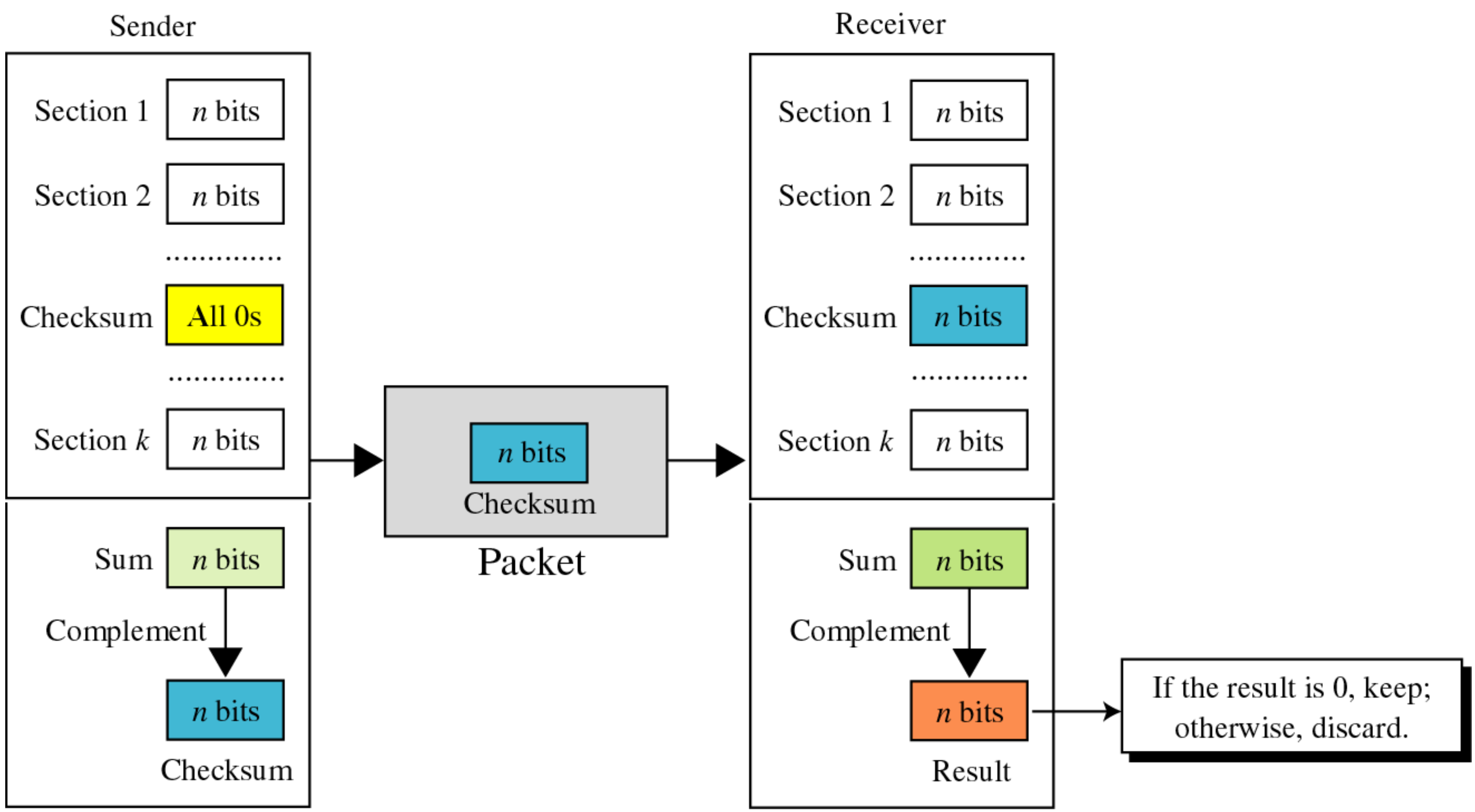


Figure 8-23

Checksum in one's complement arithmetic

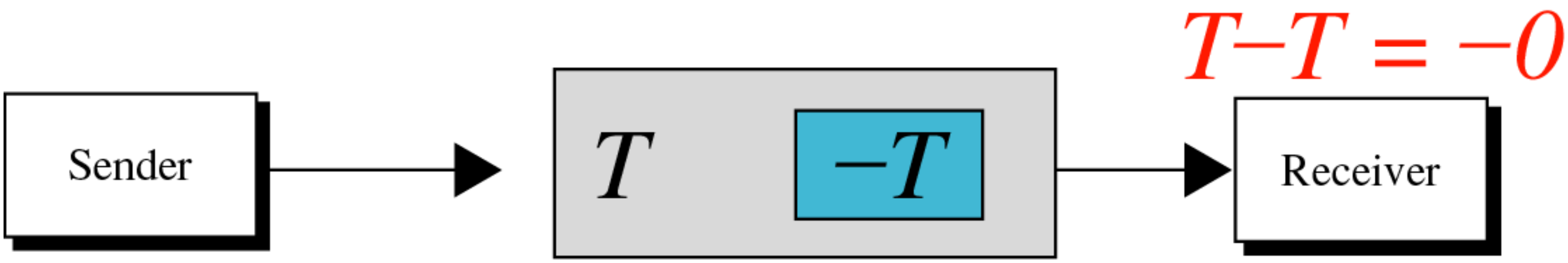
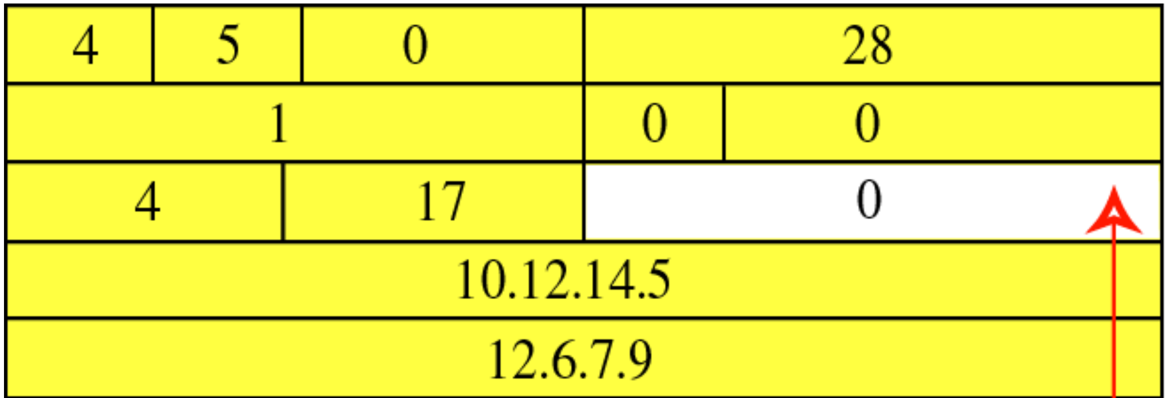


Figure 8-24

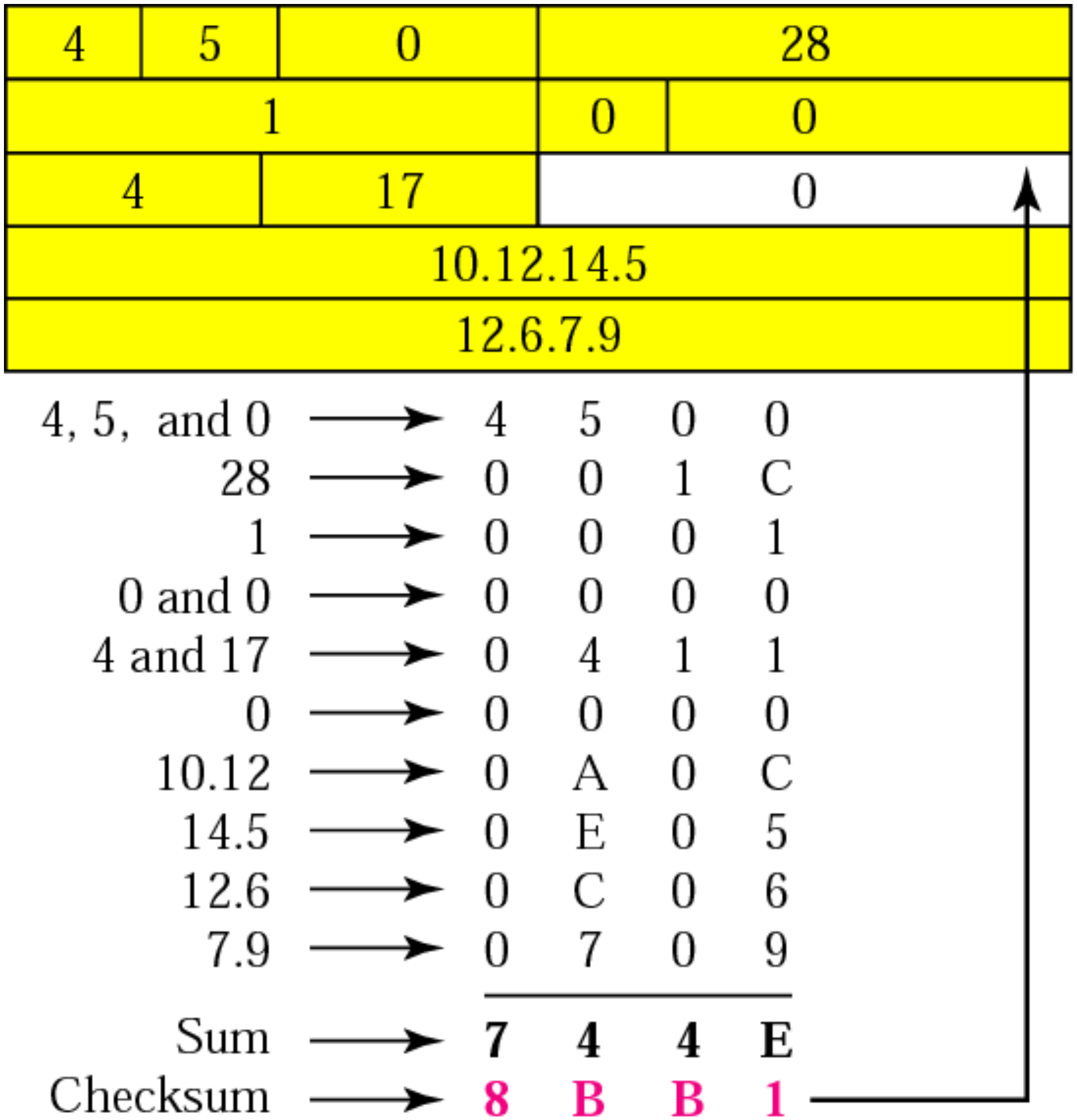
Example of checksum calculation in binary



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

Figure 8-25

Example of checksum calculation in hexadecimal



Note

Check Appendix C for a detailed description of checksum calculation and the handling of carries.

8.5

IP PACKAGE

Figure 8-26

IP components

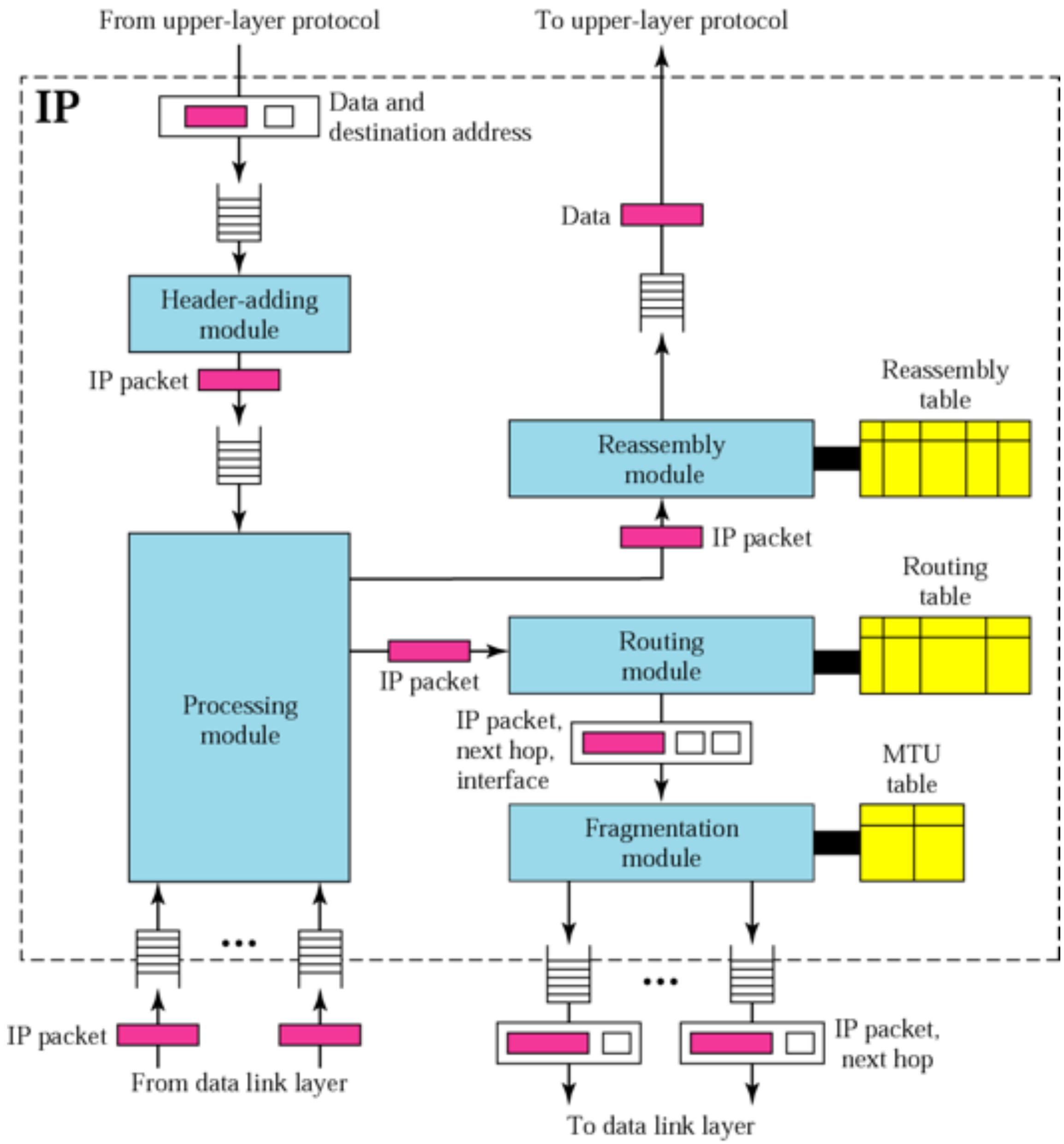


Figure 8-27

MTU table

Interface Number	MTU
....	....
....	....
....	....

Figure 8-28

Reassembly table

St.: State
S. A.: Source address
D. I.: Datagram ID

T. O.: Time-out
F.: Fragments