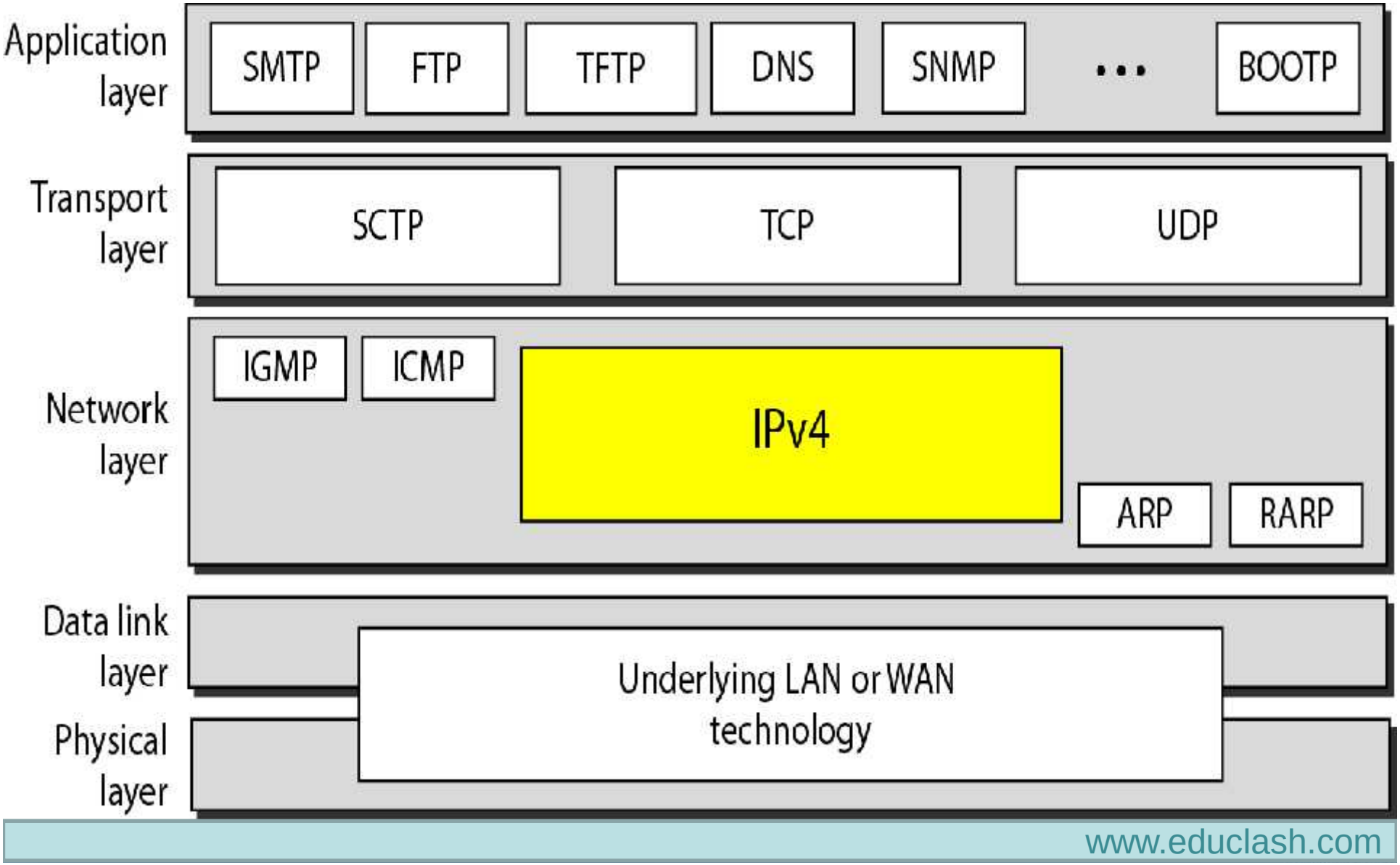
# IPv4 Packet Format

# IPv4 Packet Format

## Topics Covered:

1. Introduction
   - Position of IPv4 in TCP/IP
   - IPv4 Shortcomings
2. Packet format in IPv4
3. Service type or differentiated services
4. Fragmentation
   - MTU
   - Need of fragmentation
   - Fields Related to Fragmentation
     - Identification
     - Flags
     - Fragmentation offset
   - Fragmentation Example
5. Options
   - Single Byte Options
   - Multiple Byte Options

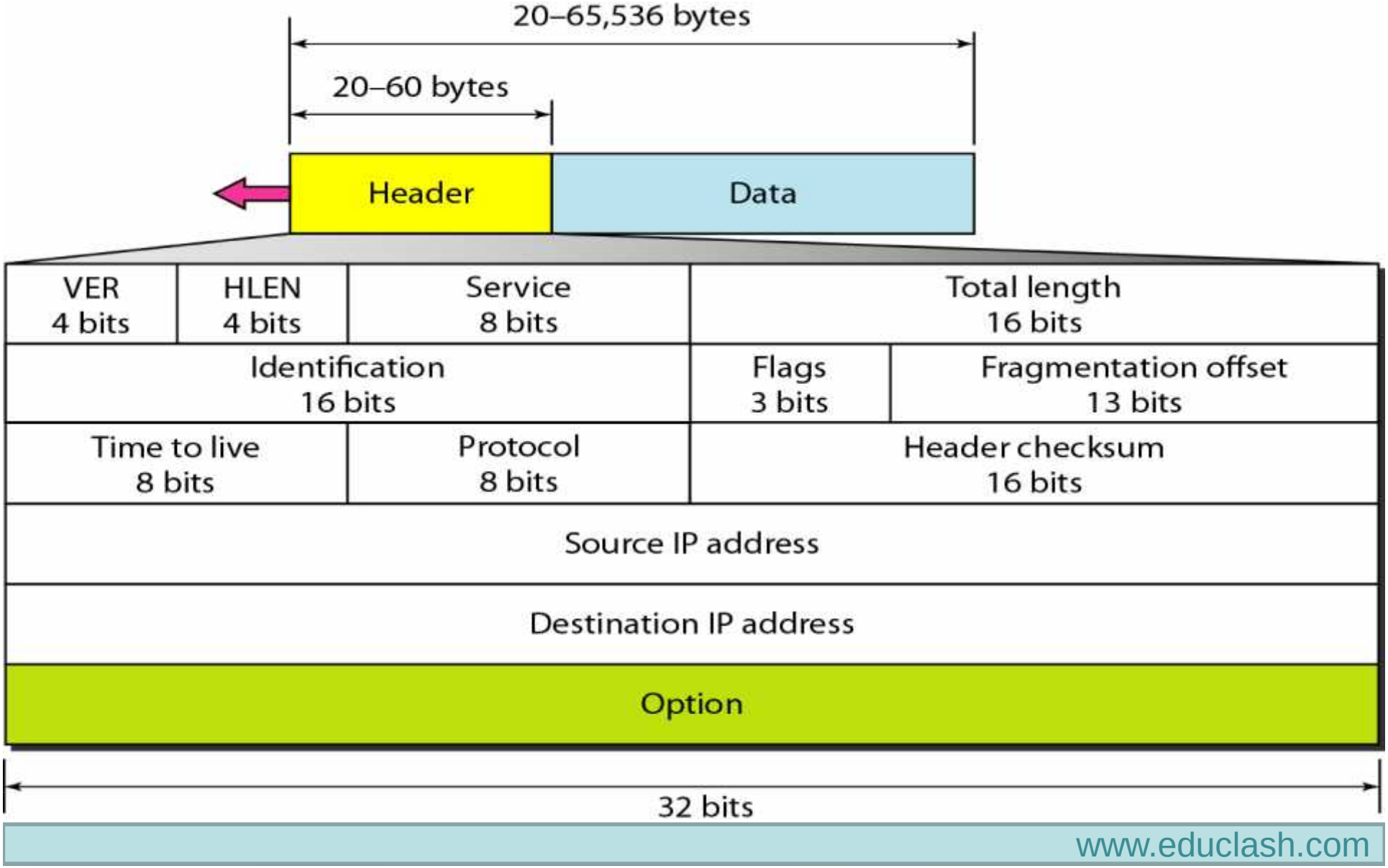
# Position of IPv4 in TCP/IP



# IPv4 Shortcomings

- IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service.
  - The term *best-effort* means that IPv4 provides no error control or flow control (except for error detection on the header).
  - IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.
  - If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.
- IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach.
  - This means that each datagram is handled independently, and each datagram can follow a different route to the destination.
  - This implies that datagrams sent by the same source to the same destination could arrive out of order.
  - Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems

# Packet format in IPv4



# Fields of Ipv4

- Version (VER):
  - This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPv6) may totally replace version 4 in the future. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4. All fields must be interpreted as specified in the fourth version of the protocol. If the machine is using some other version of IPv4, the datagram is discarded rather than interpreted incorrectly.
- Header length (HLEN):
  - This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).

# Fields of IPv4

- Services:
  - IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.
- Total length:
  - This is a In-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

**Length of data =total length - header length**

  - 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. Though a size of 65,535 bytes might seem large, the size of the IPv4 datagram may increase in the near future as the underlying technologies allow even more throughput (greater bandwidth).

# Fields of IPv4

- Identification:
  - This field is used in fragmentation.
- Flags:
  - This field is used in fragmentation.
- Fragmentation offset:
  - This field is used in fragmentation.

# Fields of IPv4

- Time to live:
  - 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. This field is needed because routing tables in the Internet can become corrupted.

# Fields of IPv4

  - A datagram may travel between two or more routers for a long time without ever getting delivered to the destination host. This field limits the lifetime of a datagram. Another use of this field is to intentionally limit the journey of the packet. For example, if the source wants to confine the packet to the local network, it can store 1 in this field. When the packet arrives at the first router, this value is decremented to 0, and the datagram is discarded.

- 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. In other words, since the IPv4 protocol carries data from different other protocols, the value of this field helps the receiving network layer know to which protocol the data belong

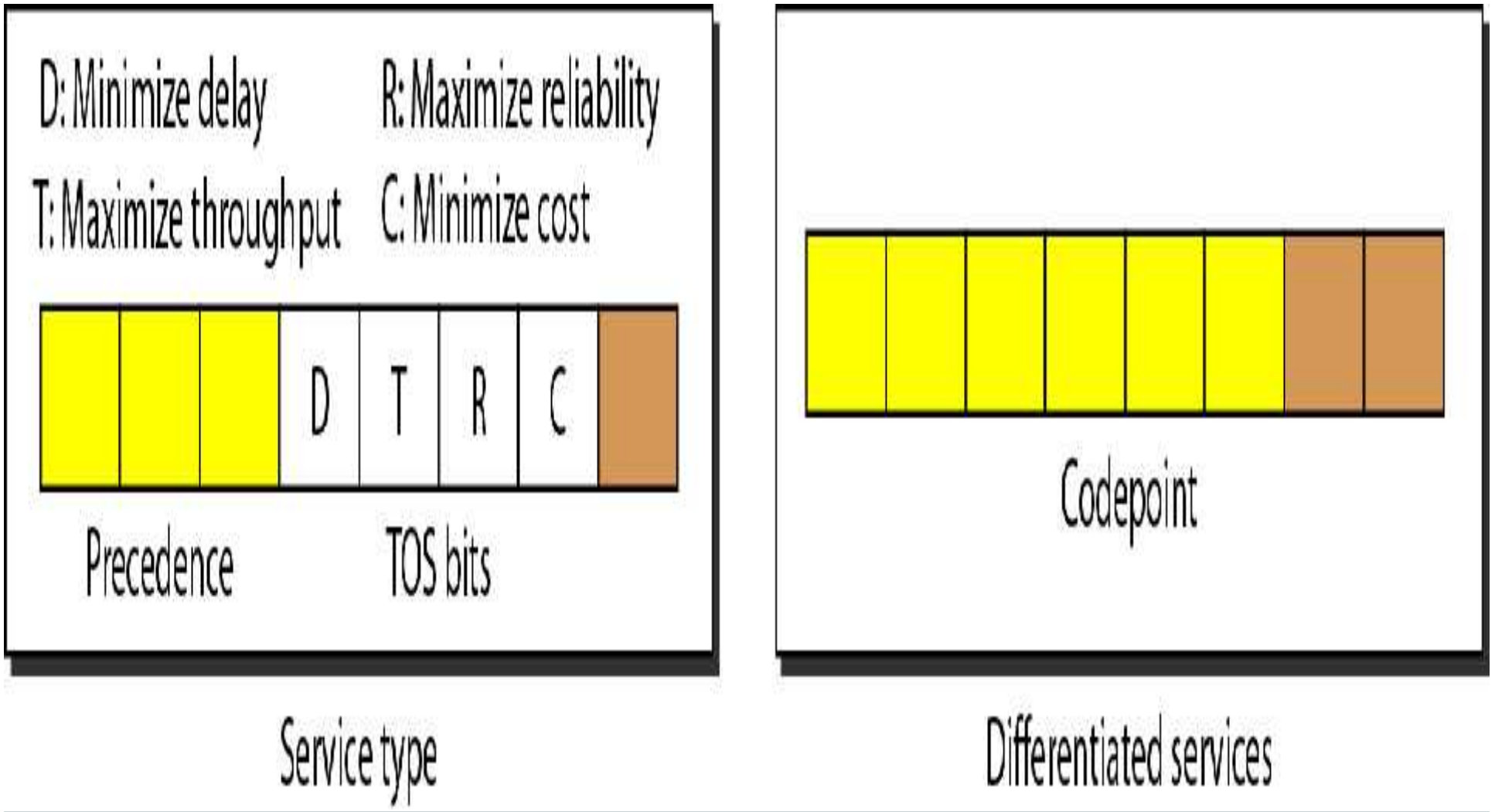
# Fields of IPv4

- Checksum:
  - The checksum concept and its calculation are discussed later .

- 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.

# Service type or differentiated services



# Service type or differentiated services

- In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.
  - Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first. Some datagrams in the Internet are more important than others. For example, a datagram used for network management is much more urgent and important than a datagram containing optional information for a group.
  - TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in Table 20.1. With only 1 bit set at a time, we can have five different types of services

# Service type or differentiated services

- Differentiated Services
  - In this interpretation, the first 6 bits make up the code point subfield, and the last 2 bits are not used. The code point subfield can be used in two different ways.
  - When the 3 rightmost bits are Os, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.
  - When the 3 rightmost bits are not all 0s, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities according to Table 20.3.
    - The first category contains 32 service types; the second and the third each contain 16. The first category (numbers 0, 2,4, ... ,62) is assigned by the Internet authorities (IETF).
    - The second category (3, 7, 11, 15, , 63) can be used by local authorities (organizations).
    - The third category (1, 5, 9, ,61) is temporary and can be used for experimental purposes

# Fragmentation

- A datagram can travel through different networks.
- Each router decapsulates the IPv4 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.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

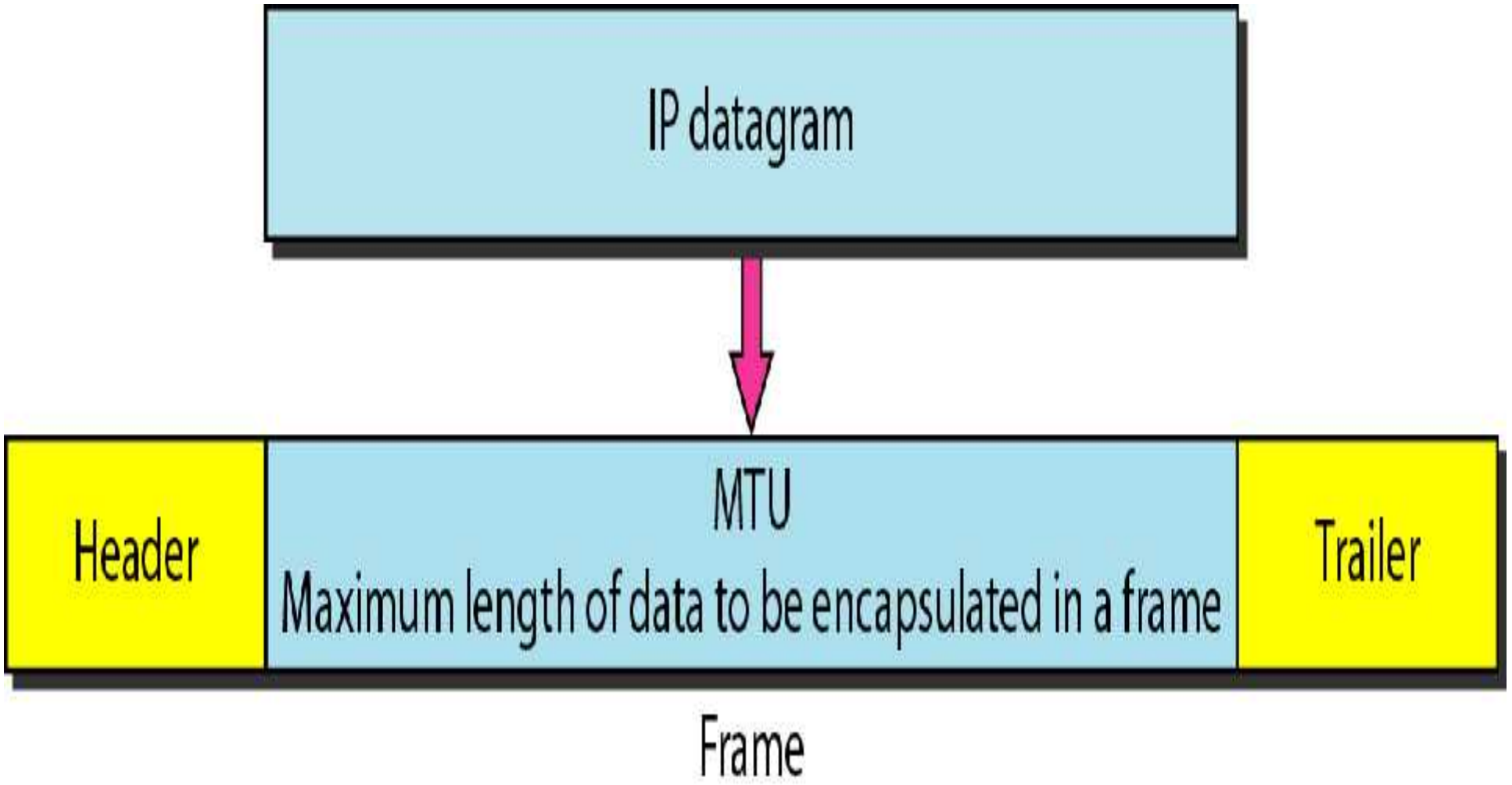
# Maximum Transfer Unit (MTU)

- Each data link layer protocol has its own frame format in most protocols. One of the fields defined in the format is the maximum size of the data field.

- In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network (see Figure 20.9).

- The value of the MTU depends on the physical network protocol. Table 20.5 shows the values for some protocols.

# MTU


IP datagram
Header
MTU
Maximum length of data to be encapsulated in a frame
Trailer
Frame

# Need of fragmentation

- To make the IPv4 protocol independent of the physical network, the designers decided to make the maximum length of the IPv4 datagram equal to 65,535 bytes.
- This makes transmission more efficient if we use a protocol with an MTU of this size.
- However, for other physical networks, we must divide the datagram to make it possible to pass through these networks.
- This is called fragmentation

# Points to remember

- The source usually does not fragment the IPv4 packet.
- The transport layer will instead segment the data into a size that can be accommodated by IPv4 and the data link layer in use.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed
- The reassembly of the datagram, however, is done only by the destination host because each fragment becomes an independent datagram.
  - Whereas the fragmented datagram can travel through different routes, and we can never control or guarantee which route a fragmented datagram may take, all the fragments belonging to the same datagram should finally arrive at the destination host.
  - So it is logical to do the reassembly at the final destination

# Fields Related to Fragmentation

- Identification
- Flags
- Fragmentation offset

# 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.
- To guarantee uniqueness, the IPv4 protocol uses a counter to label the datagrams. The counter is initialized to a positive number. When the IPv4 protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by 1. As long as the counter is kept in the main memory, uniqueness is guaranteed.
- When a datagram is fragmented, the value in the identification field is copied to all fragments. In other words, all fragments have the same identification number, the same as the original datagram.
- The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.

# 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

Figure 20.10 *Flags used in fragmentation*

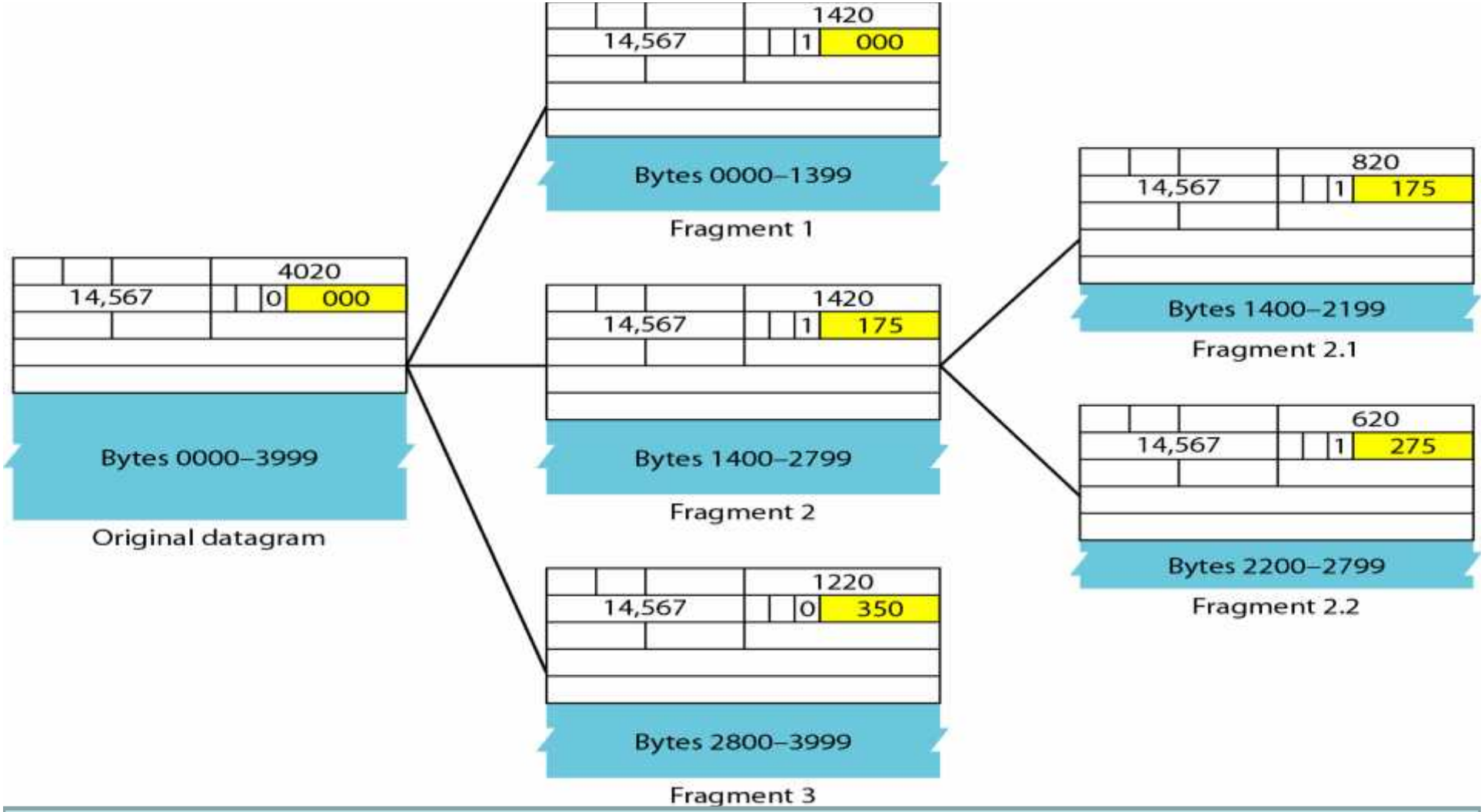
# Fragmentation offset:

- 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 20.11 shows a datagram with a data size of 4000 bytes fragmented into three fragments.
  - The bytes in the original datagram are numbered 0 to 3999.
  - The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = 0.
  - The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175.
  - Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.

# Fragmentation offset:

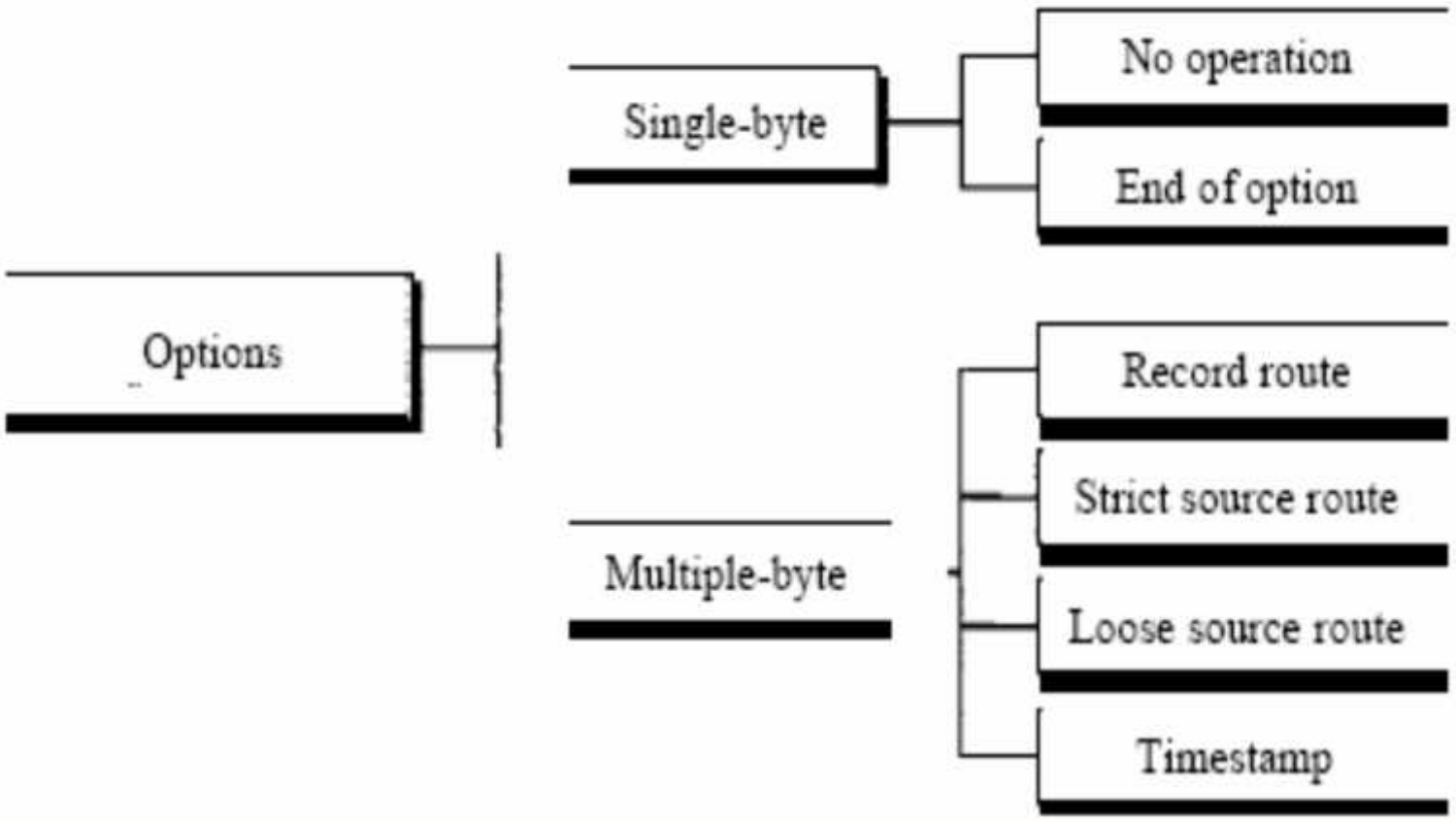
- Remember that the value of the offset is measured in units of 8 bytes. This is done because the length of the offset field is only 13 bits and cannot represent a sequence of bytes greater than 8191. This forces hosts or routers that fragment datagrams to choose a fragment size so that the first byte number is divisible by 8. Figure 20.12 shows an expanded view of the fragments in Figure 20.11.
- Notice the value of the identification field is the same in all fragments.
- Notice the value of the flags field with the *more* bit set for all fragments except the last.
- Also, the value of the offset field for each fragment is shown.

# Fragmentation Example



# Options

Figure 20.14 *Taxonomy of options in IPv4*



# Single Byte Options

- No Operation
  - A no-operation option is a 1-byte option used as a filler between options.

- End of Option
  - An end-of-option option is a 1-byte option used for padding at the end of the option field. It, however, can only be used as the last option.

# Multiple Byte Options

- Record Route
  - A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.
- Loose Source Route:
  - A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
- Timestamp
  - A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet. We can estimate the time it takes for a datagram to go from one ~outer to another. We say *estimate* because, although all routers may use Universal time, their local clocks may not be synchronized

# Multiple Byte Options

- Strict Source Route
  - A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet.
  - Dictation of a route by the source can be useful for several purposes. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput.
  - Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. For example, a sender can choose a route so that its datagram does not travel through a competitor's network.
  - If a datagram specifies a strict source route, all the routers defined in the option must be visited by the datagram. A router must not be visited if its IPv4 address is not listed in the datagram.
  - If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued. If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued

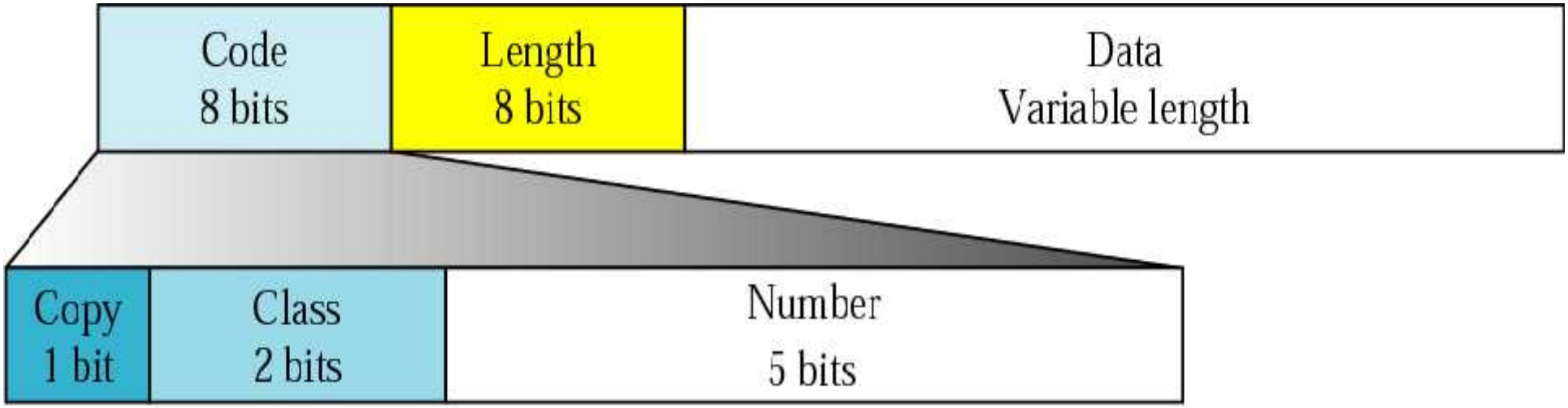
# 8.3 OPTIONS

*The header of the IP datagram is made of two parts: a fixed part and a variable part. The variable part comprises the options that can be a maximum of 40 bytes.*

*The topics discussed in this section include:*

*Format*
*Option Types*

# *Option format*

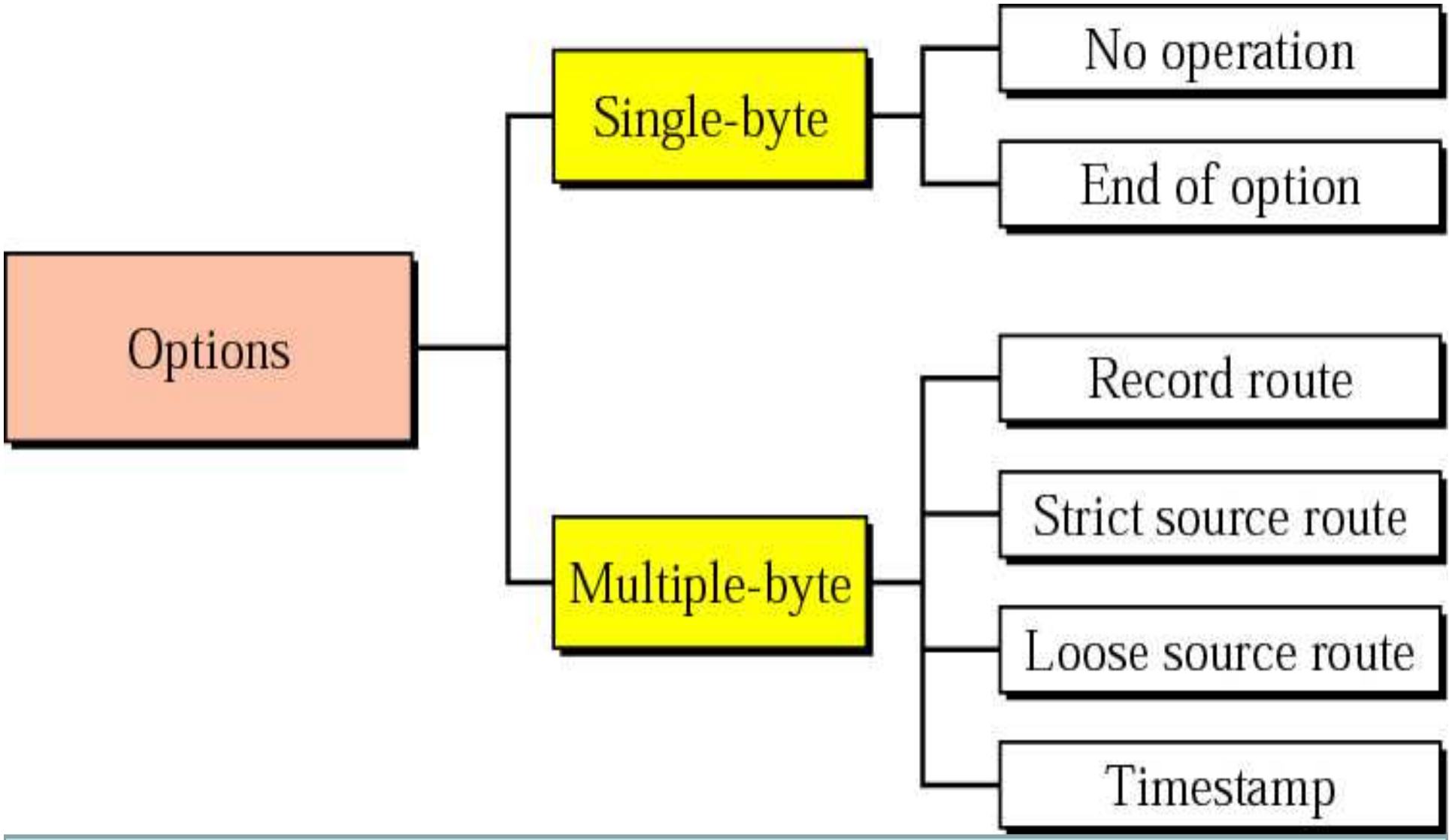



Copy
0 Copy only in first fragment
1 Copy into all fragments

Class
00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

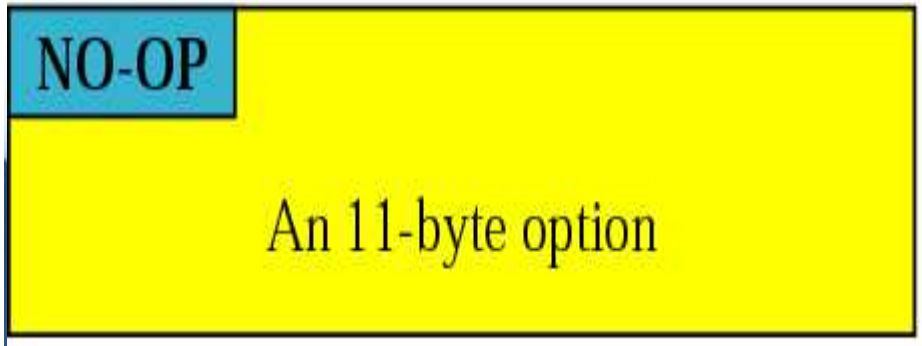
Number
00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

*Categories of options*

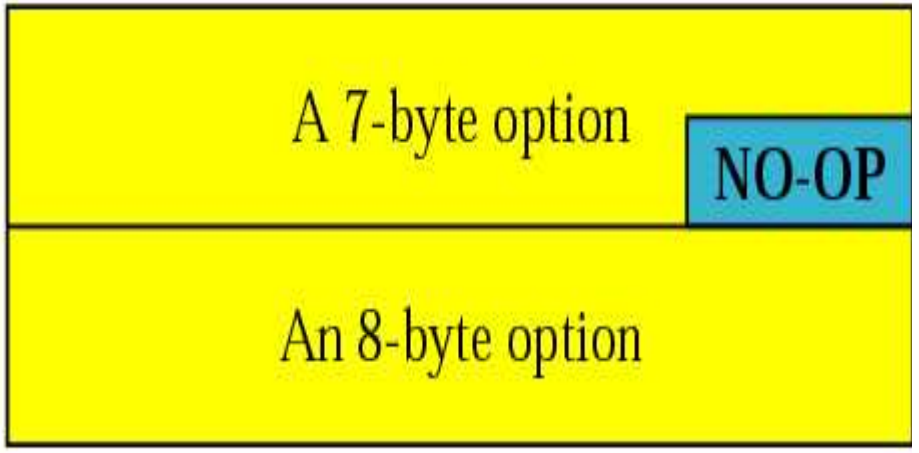


# *No operation option*


a. No operation option

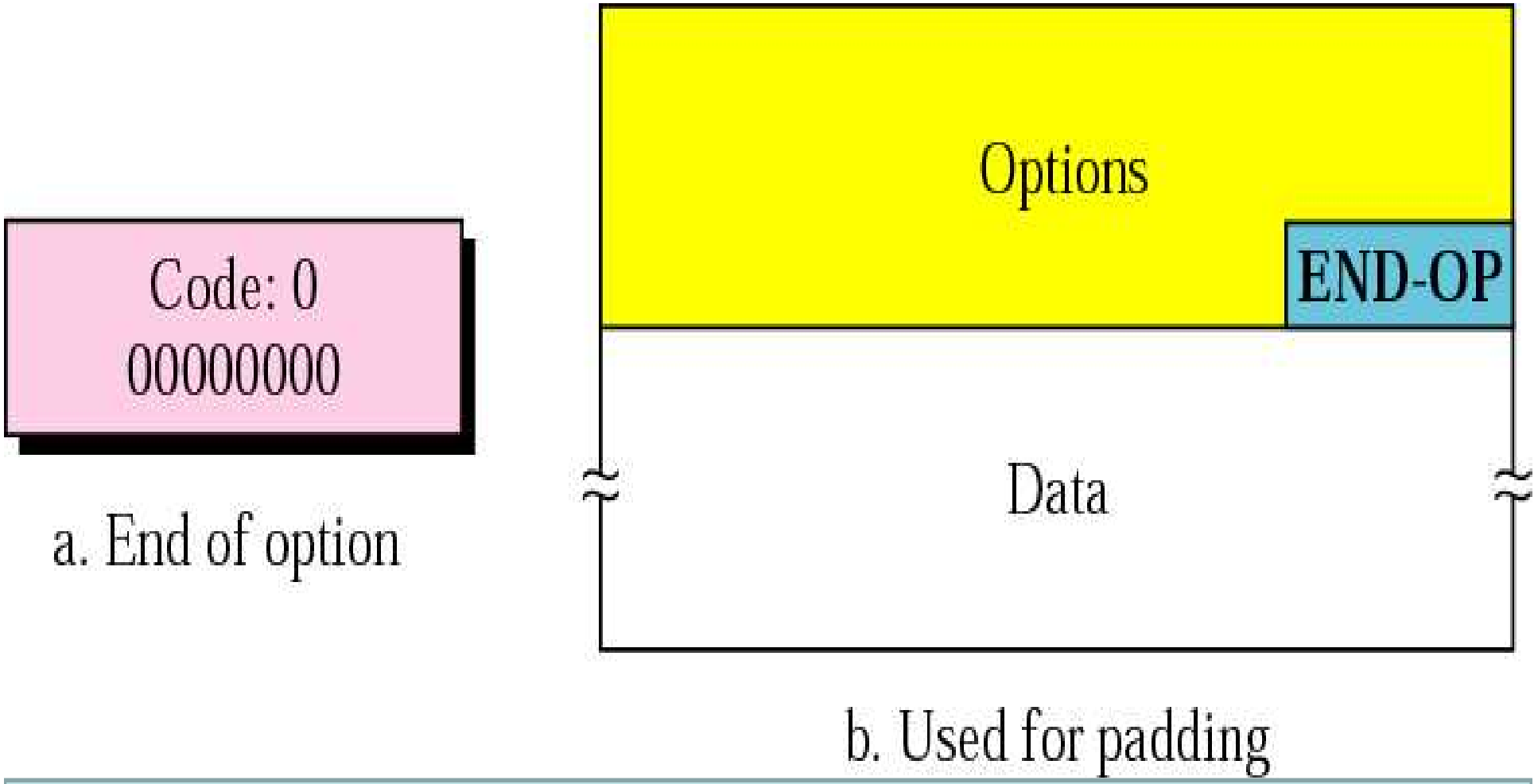



b. Used to align beginning of an option

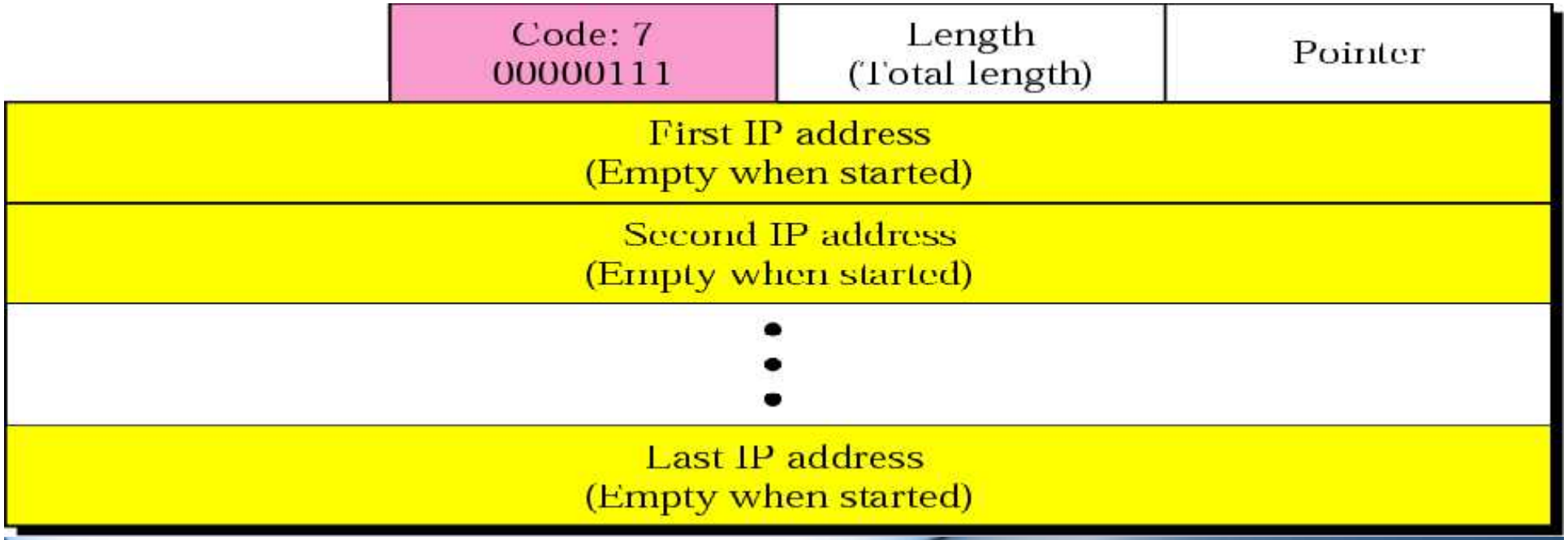



c. Used to align the next option

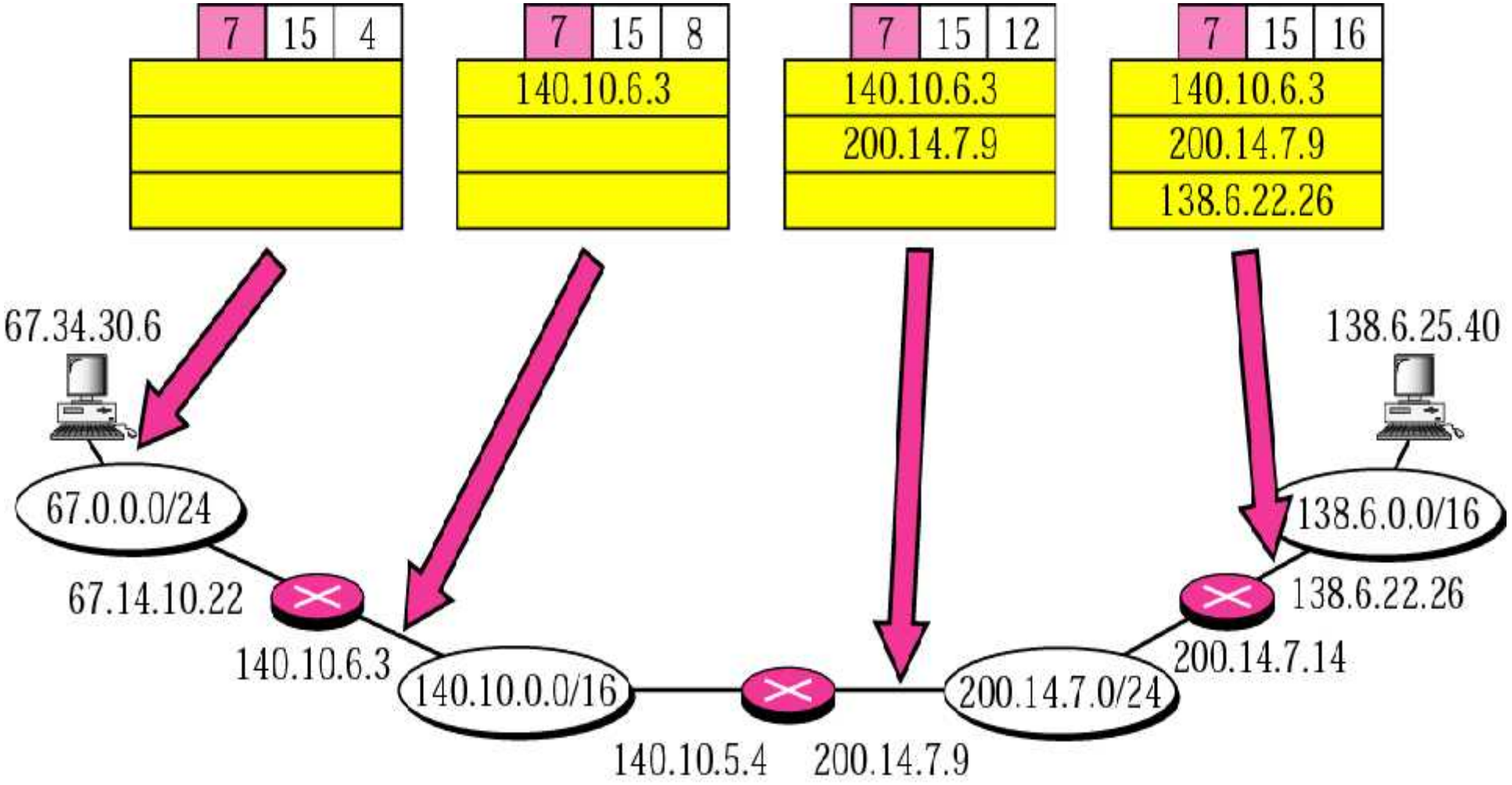
# *End of option option*



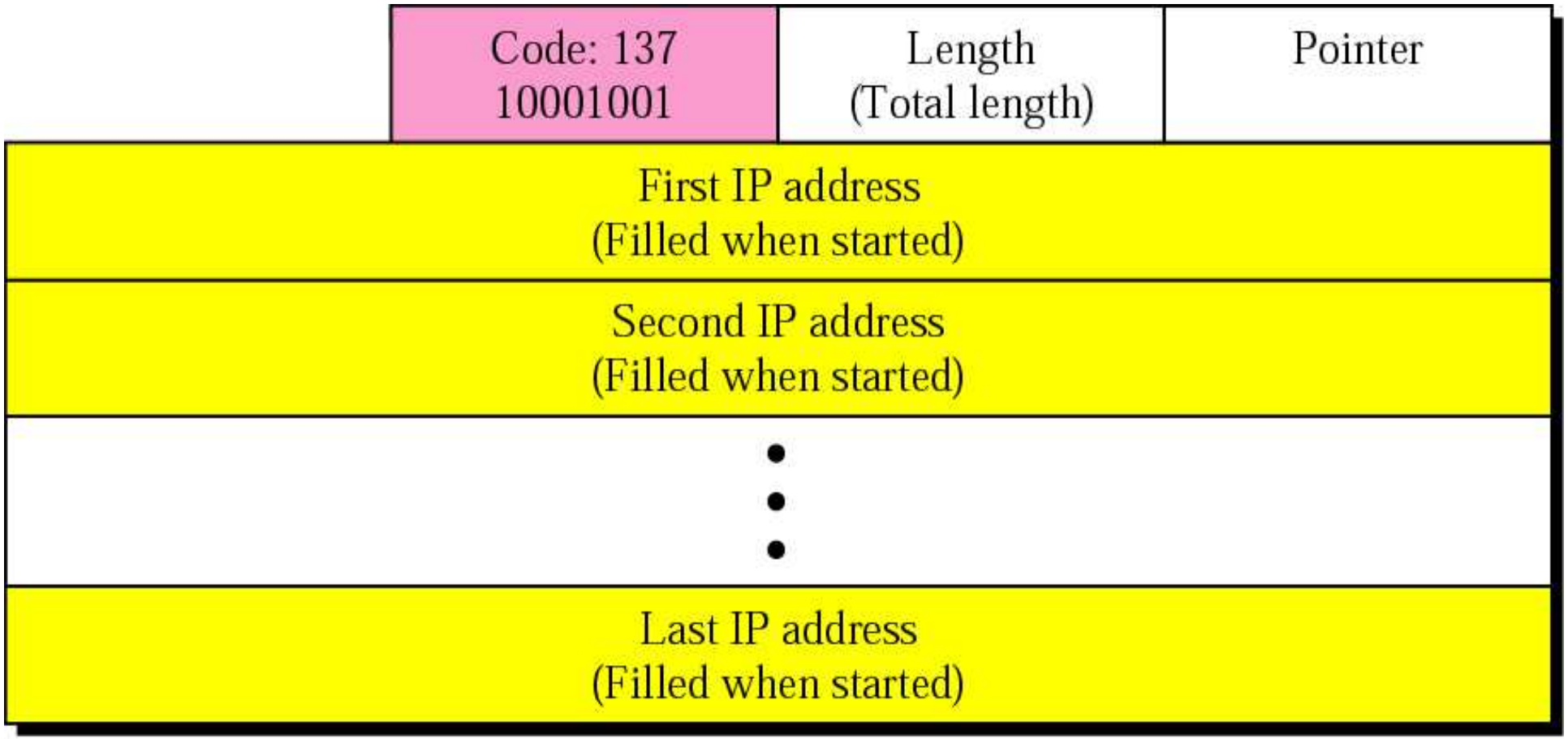
# *Record route option*



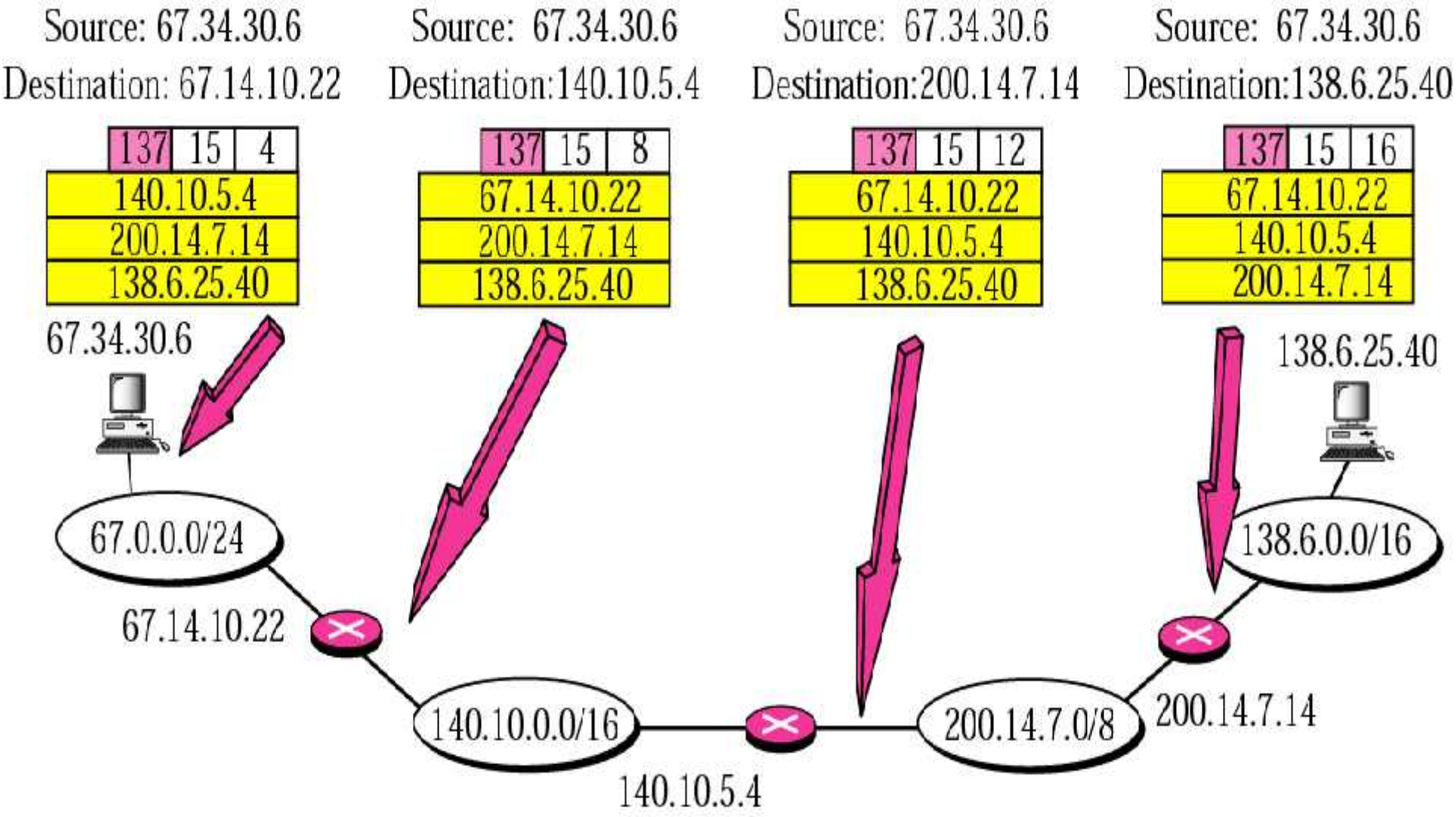
# *Record route concept*



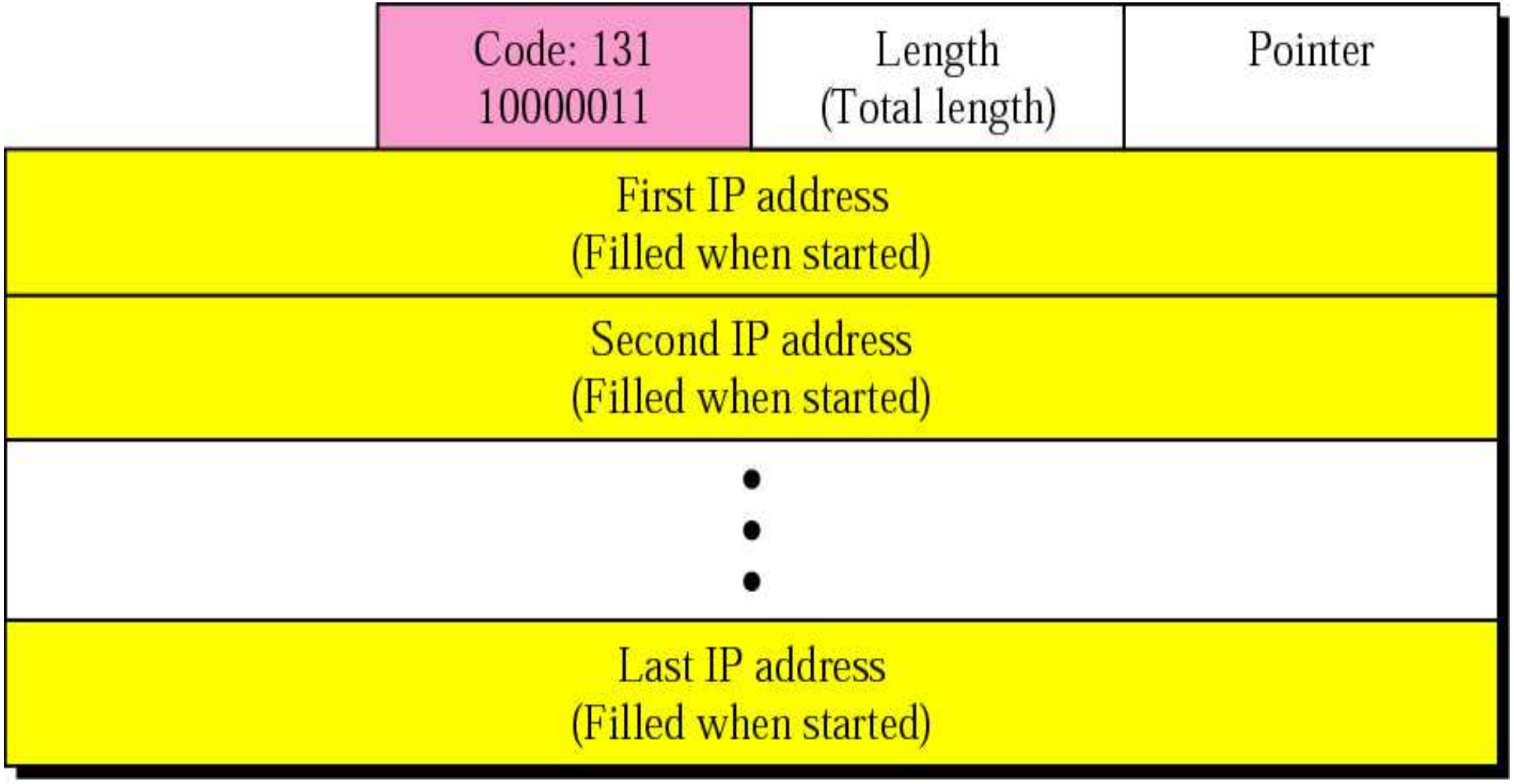
# *Strict source route option*



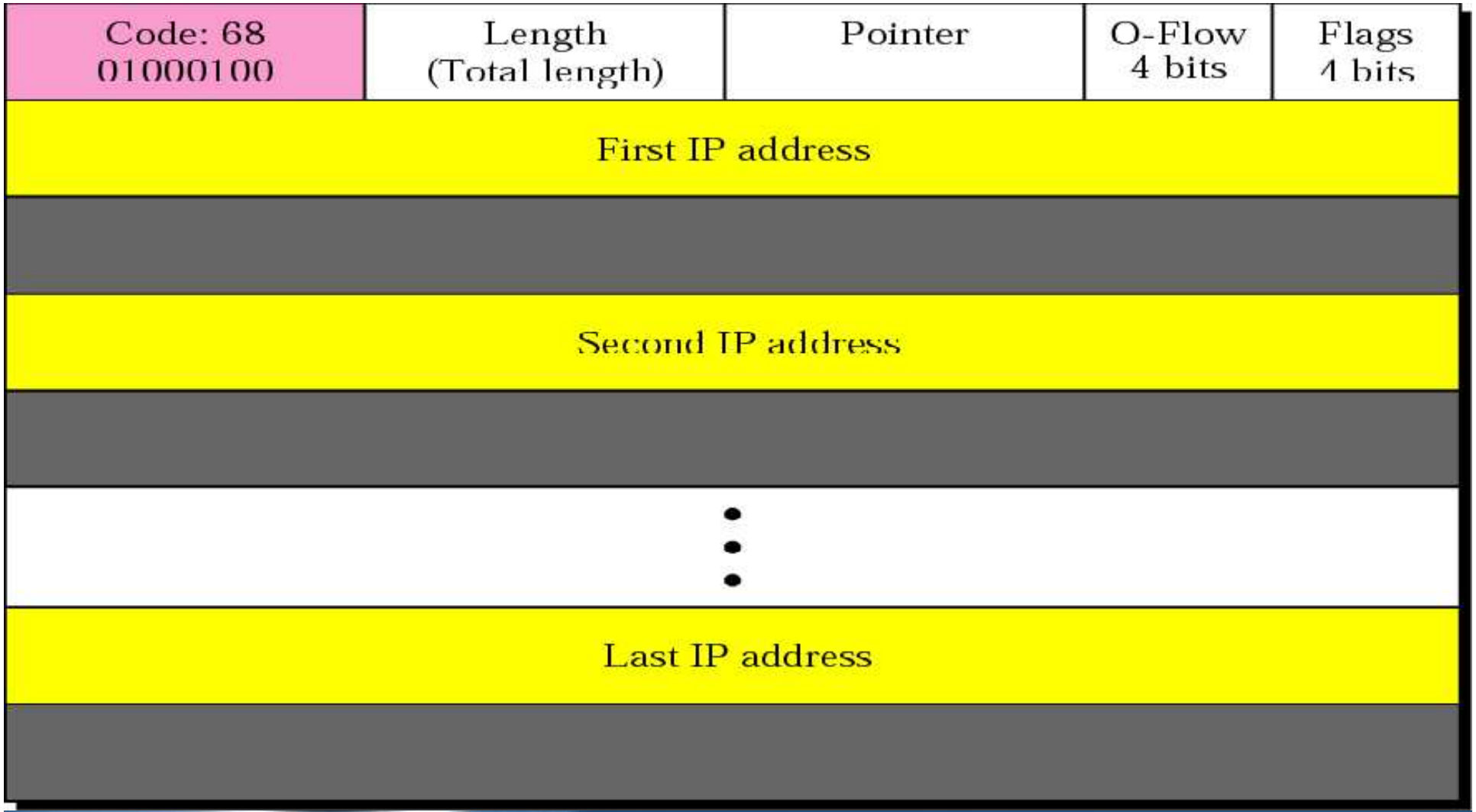
# *Strict source route concept*



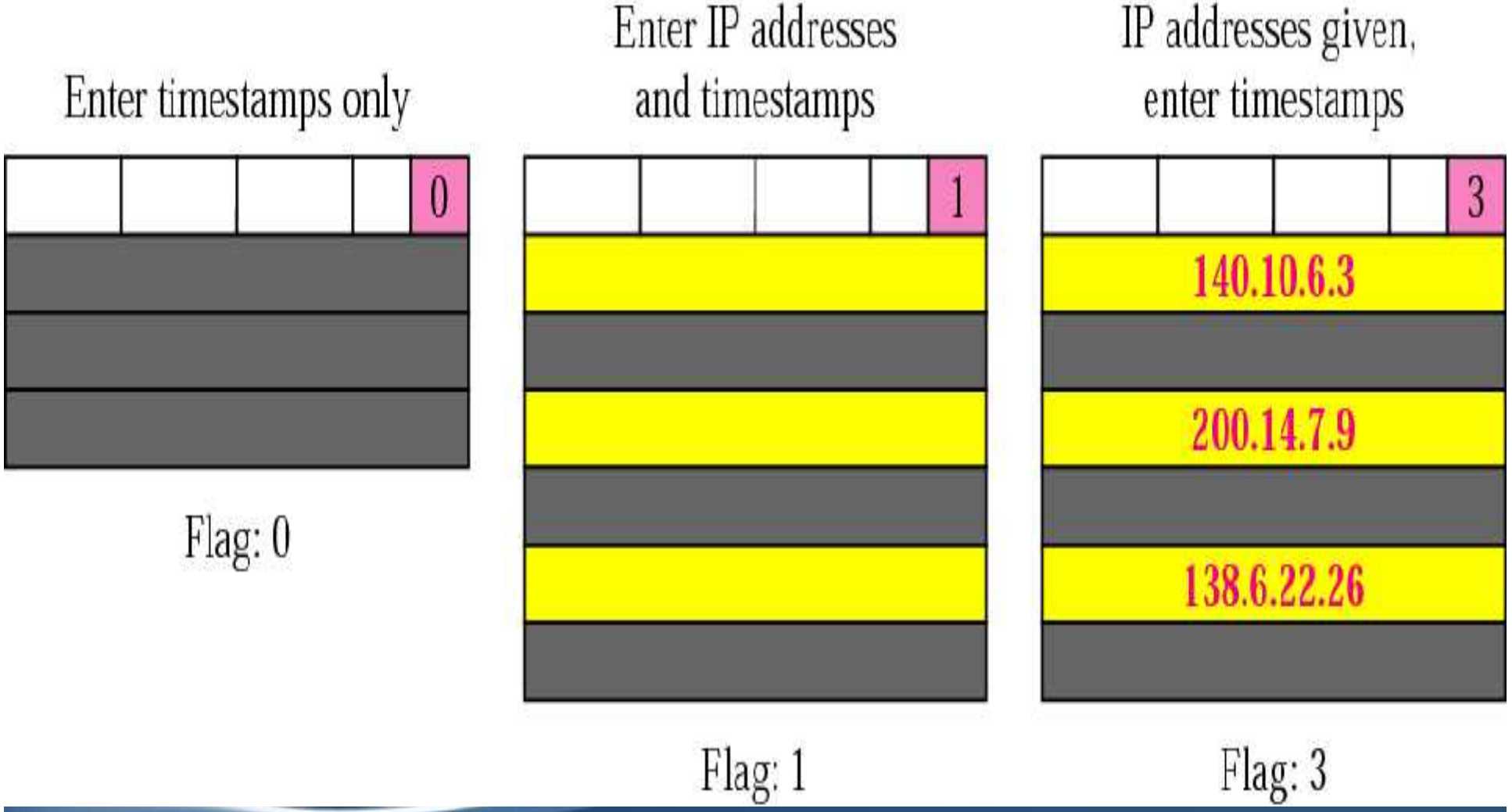
# *Loose source route option*



# *Timestamp option*



# *Use of flag in timestamp*



# *Timestamp concept*