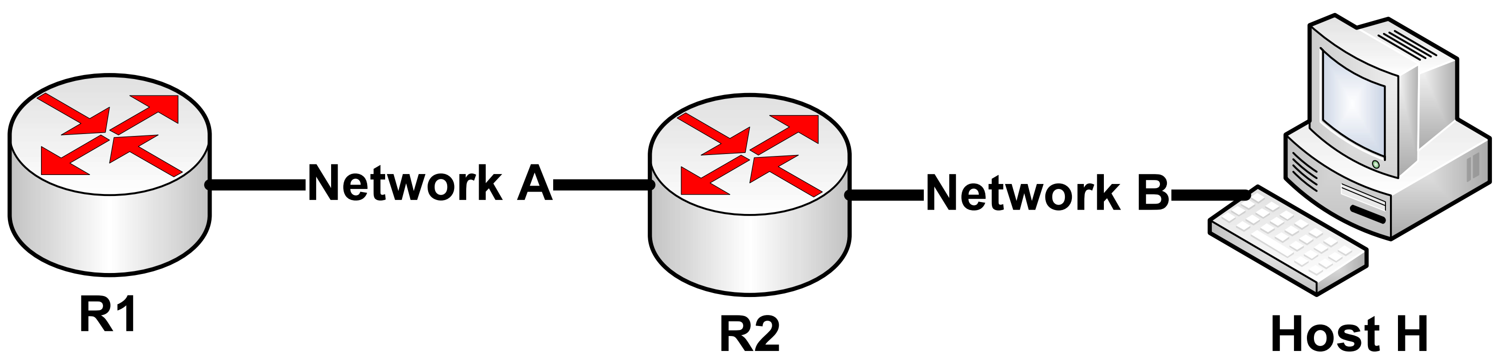
# Example 7.9

Considering the following scenario show all the IP Packets generated by router R1 and R2 for the network shown in Figure Q3A. Assume an MTU of 1500 Bytes for network A and 535 bytes for network B respectively. R1 receives an IP packet P directed at H and containing a 2000 bytes TCP segment. IP header is of 20 bytes. Describe the M bit and the Offset field for IP fragments of packet P sent from R1 on network A and R2 over network B.



# Example 7.9

## *Solution*

**R1 on network A:**

**MTU = 1500**

**20 header + 1480 payload**

**20 header + 520 payload - 2 fragments**

- **First fragment: M=1 , Offset= 0**
- **Second fragment: M=0 , Offset= 1480/8 = 185**

**R2 over network B:**

- **First packet originating from R1: 20+512+**
  **20+512+**
  **20+456 3 fragments**
  - **M = 1 , Offset= 0**
  - **M = 1, Offset = 512/8 = 64**
  - **M = 1, Offset = 1024/8 = 128**
- **Second packet originating from R1: 20+512+**
  **20+8 2 fragments**
  - **M=1, Offset = 1480/8 = 185**
  - **M=0, Offset = 1992/8 = 249**

# 7-4 OPTIONS

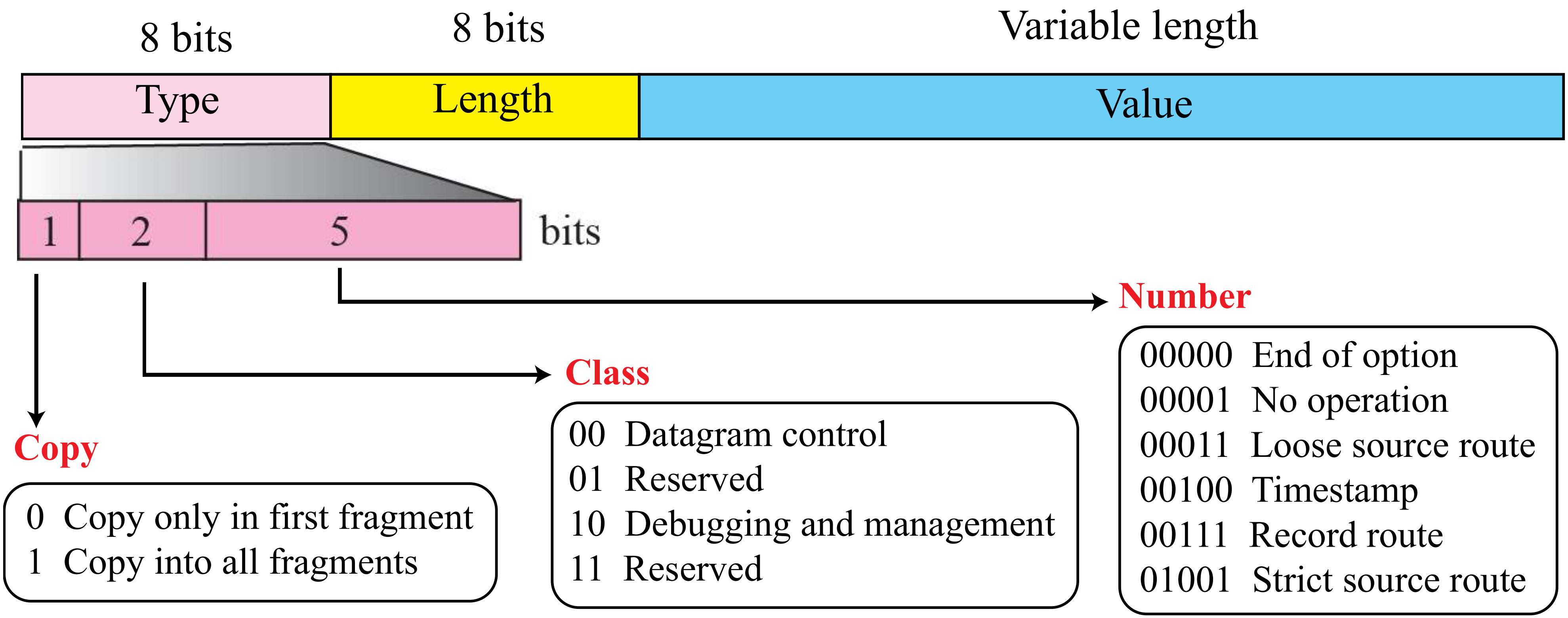
The header of the IP datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options, which can be a maximum of 40 bytes.

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 IP header, option processing is required of the IP software.

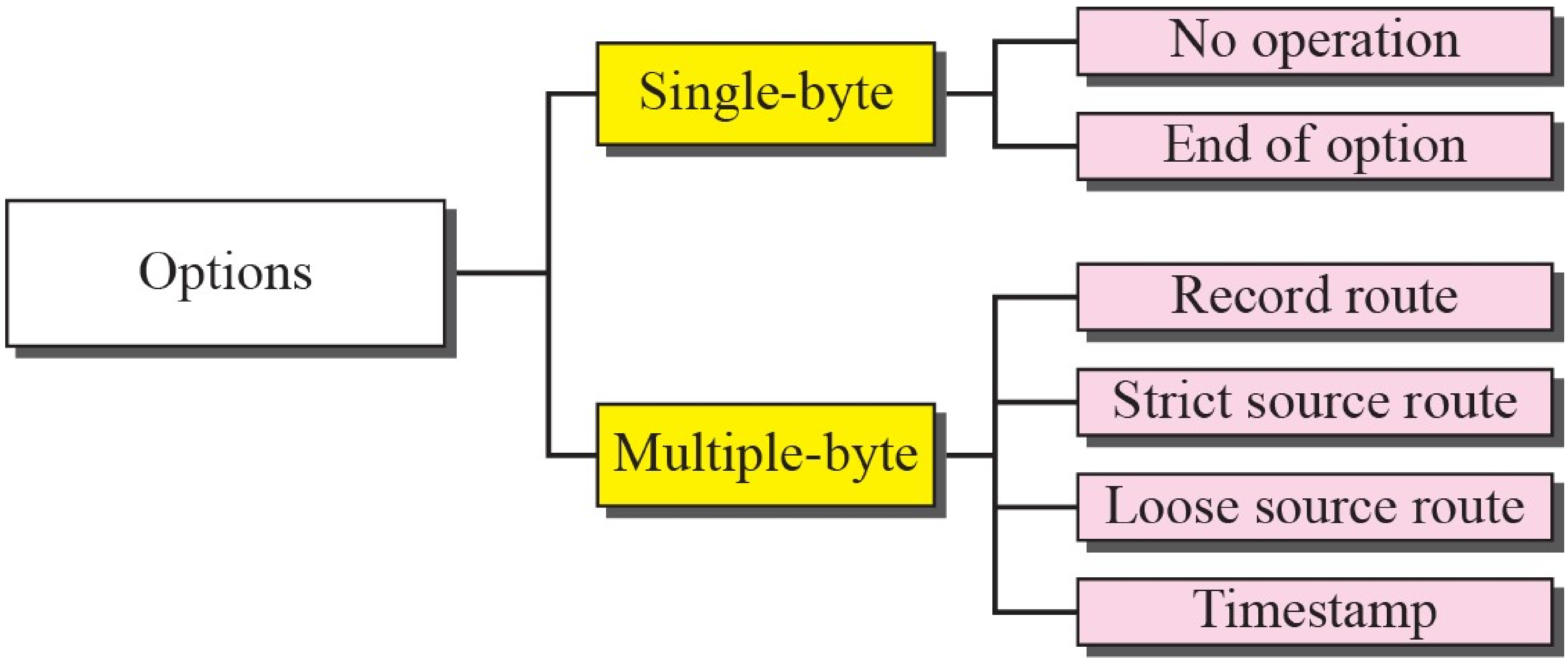
# *Topics Discussed in the Section*

- ✓ **Format**
- ✓ **Option Types**

## Figure 7.10 *Option format*



# **Figure 7.11** *Categories of options*



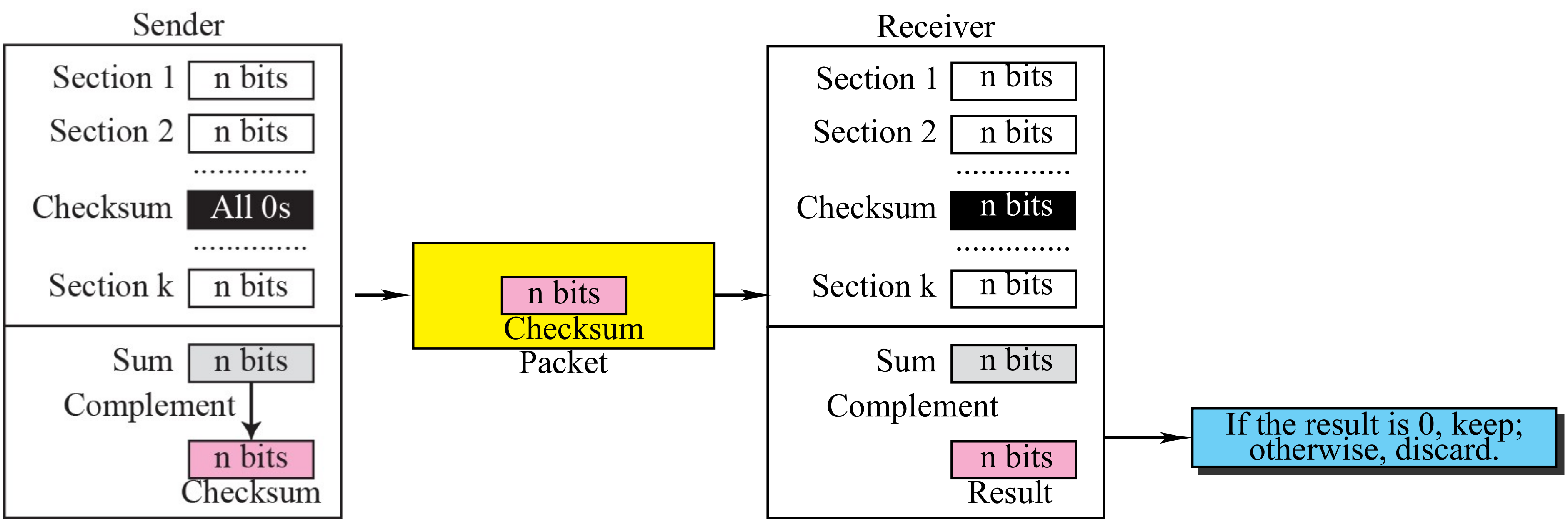
# 7-5 CHECKSUM

**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.**

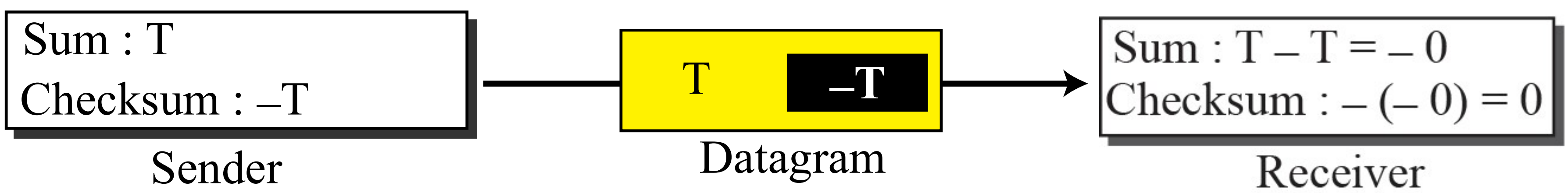
# *Topics Discussed in the Section*

- ✓ **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*

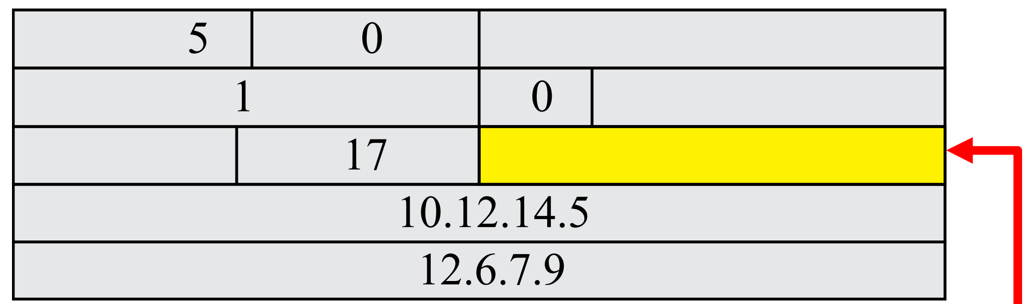
***Checksum in IP covers only the header, not the data.***

# 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*

| | | | |
|---|---|---|---|
| 4 | 5 | 0 | 28 |
| 1 | | 0 | 0 |
| 4 | 17 | 35761 | |
| 10.12.14.5 | | | |
| 12.6.7.9 | | | |

| | | |
|---|---|---|
| 4, 5, and 0 → | 01000101 | 00000000 |
| 28 → | 00000000 | 00011100 |
| 1 → | 00000000 | 00000001 |
| 0 and 0 → | 00000000 | 00000000 |
| 4 and 17 → | 00000100 | 00010001 |
| Checksum → | **10001011** | **10110001** |
| 10.12 → | 00001010 | 00001100 |
| 14.5 → | 00001110 | 00000101 |
| 12.6 → | 00001100 | 00000110 |
| 7.9 → | 00000111 | 00001001 |
| Sum → | **1111 1111** | **1111 1111** |
| Checksum → | **0000 0000** | **0000 0000** |