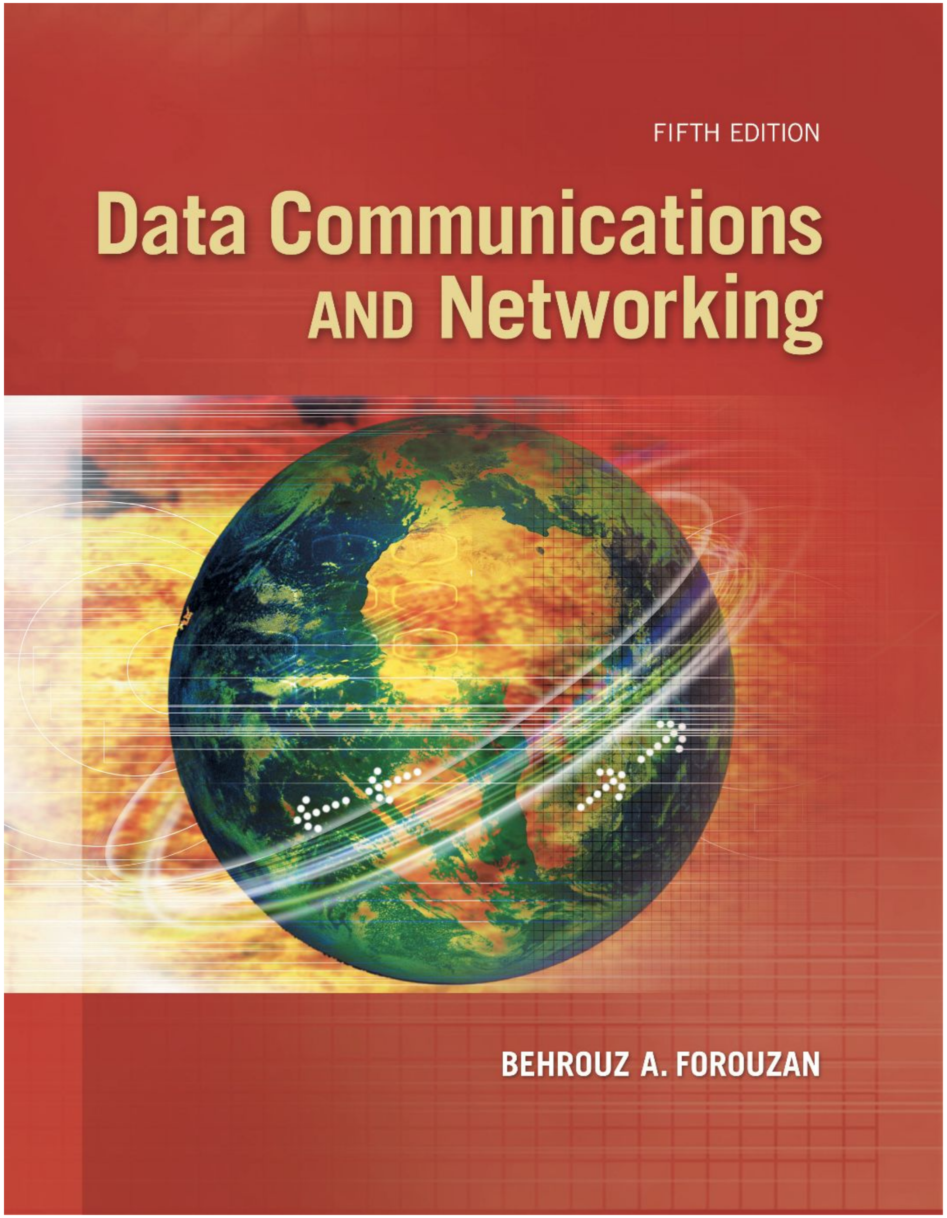
The McGraw-Hill Companies

Chapter 19

Network Layer Protocols



Chapter 4: Outline

19.1 IPv4

19.2 ICMPv4

19.3 Mobile IP

Chapter 19: Objective

- *The first section discusses the IPv4 protocol. It first describes the IPv4 datagram format. It then explains the purpose of fragmentation in a datagram. The section then briefly discusses options fields and their purpose in a datagram. The section finally mentions some security issues in IPv4, which are addressed in Chapter 32.*

- *The second section discusses ICMPv4, one of the auxiliary protocols used in the network layer to help IPv4. First, it briefly discusses the purpose of each option. The section then shows how ICMP can be used as a debugging tool. The section finally shows how the checksum is calculated for an ICMPv4 message.*

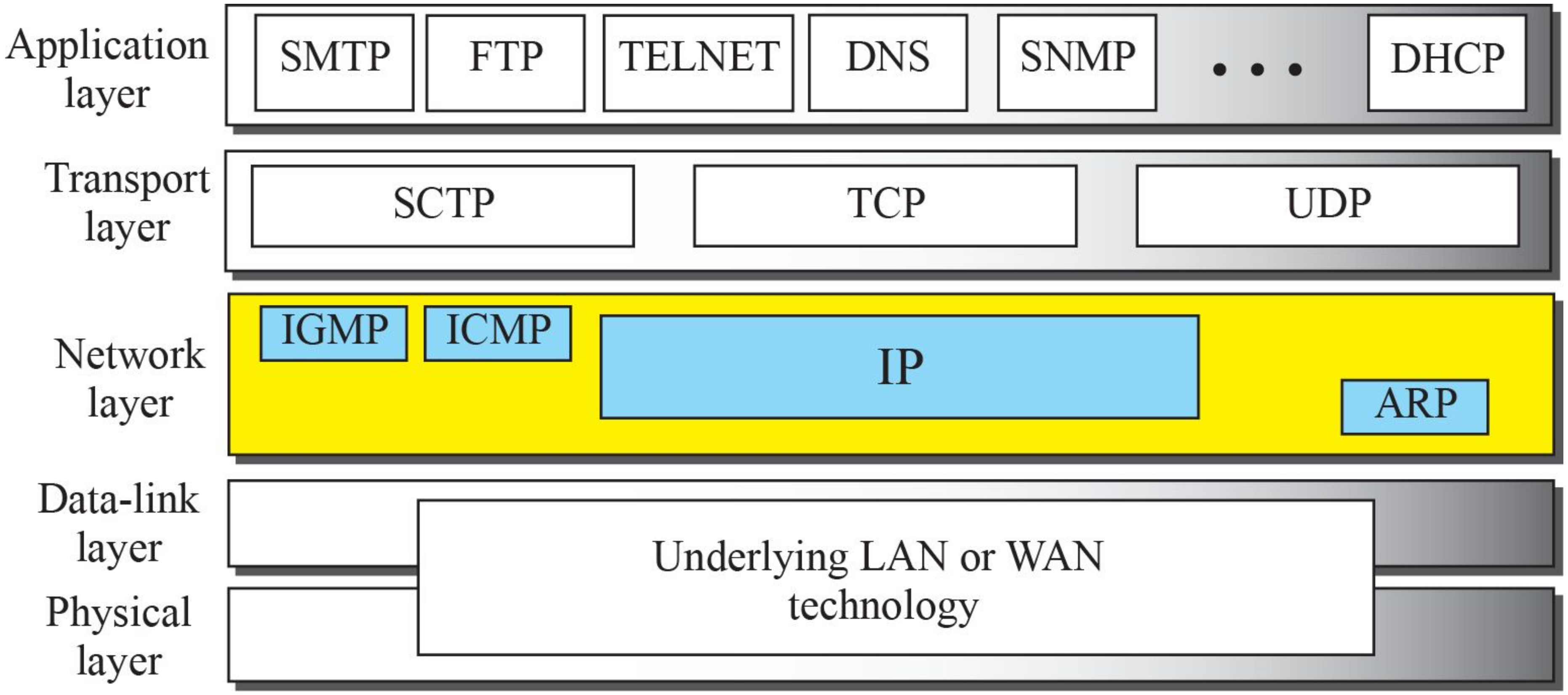
Chapter 19: Objective (continued)

- ***The third section discusses the mobile IP, whose use is increasing every day when people temporarily move their computers from one place to another. The section first describes the issue of address change in this situation. It then shows the three phases involved in the process. The section finally explains the inefficiency involved in this process and some solutions.***

19.1 NETWORK-LAYER PROTOCOLS

The network layer in version 4 can be thought of as one main protocol and three auxiliary ones. The main protocol, IPv4, is responsible for packetizing, forwarding, and delivery of a packet. The ICMPv4 helps IPv4 to handle some errors that may occur in delivery. The IGMP is used to help IPv4 in multicasting. ARP is used in address mapping.

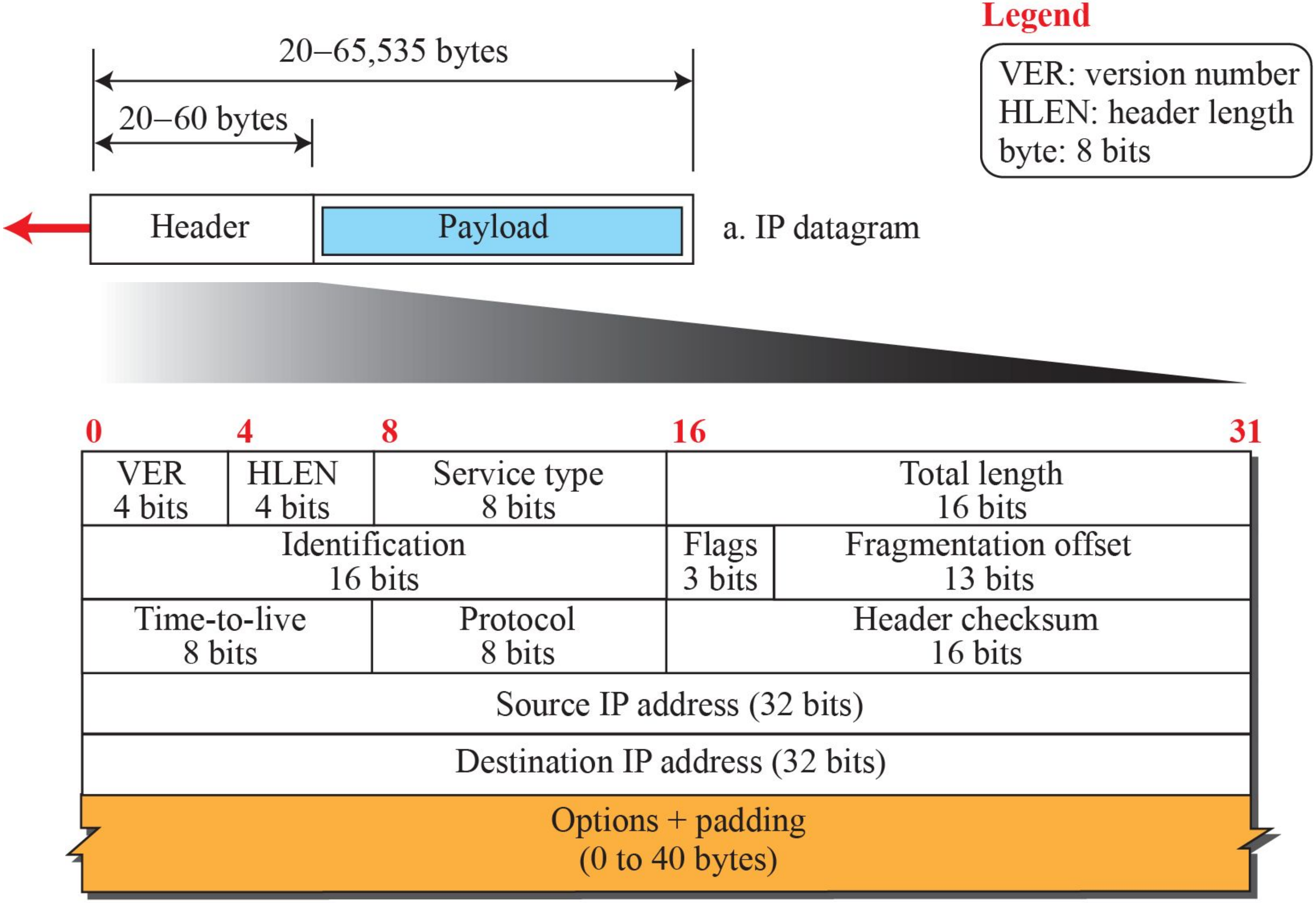
Figure 19.1: ***Position of IP and other network-layer protocols in TCP/IP protocol suite***



19.19.1 Datagram Format

Packets used by the IP are called datagrams. Figure 19.2 shows the IPv4 datagram format. A datagram is a variable-length packet consisting of two parts: header and payload (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..

Figure 19.2: IP datagram



a. IP datagram

Legend

VER: version number
HLEN: header length
byte: 8 bits

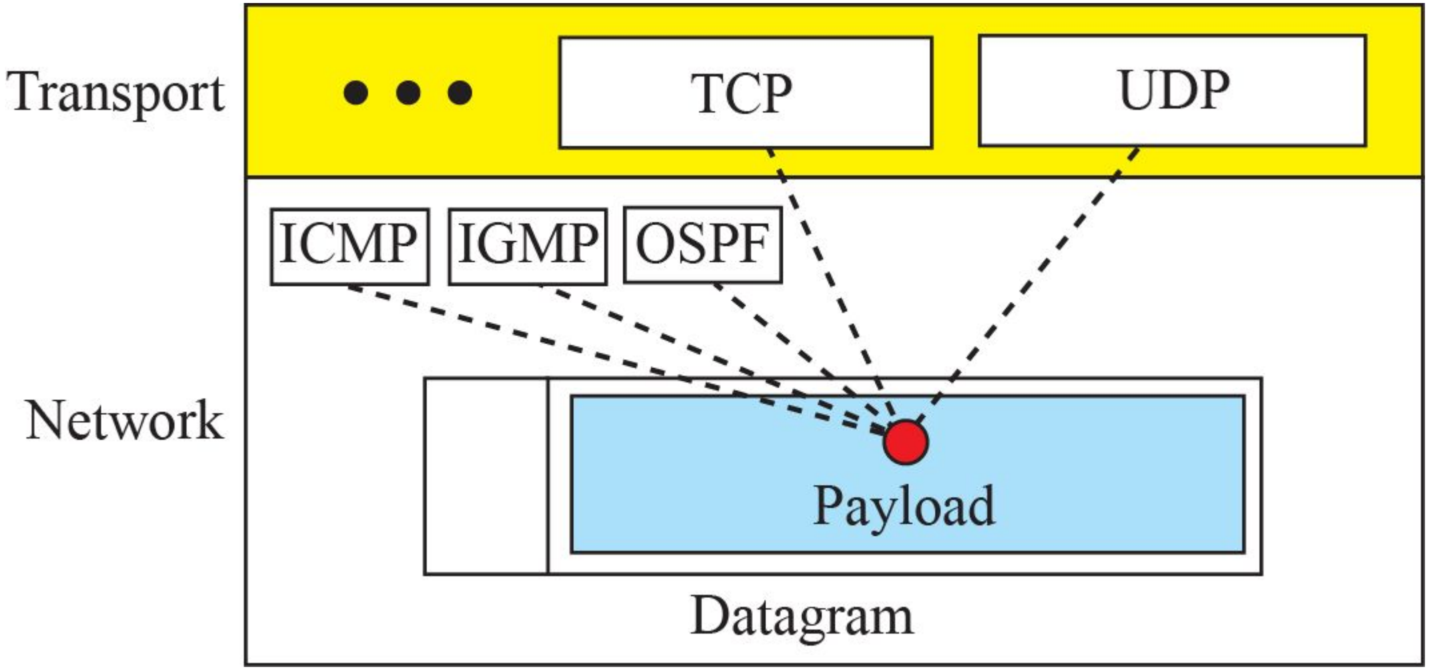
VER 4 bits	HLEN 4 bits	Service type 8 bits		Total length 16 bits
Identification 16 bits			Flags 3 bits	Fragmentation offset 13 bits
Time-to-live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address (32 bits)				
Destination IP address (32 bits)				
Options + padding (0 to 40 bytes)				

b. Header format

Figure 19.3: Multiplexing and demultiplexing using the value of the protocol field

ICMP: 01
IGMP: 02
TCP: 06
UDP: 17
OSPF: 89

Some protocol values



Example 19.1

An IPv4 packet has arrived with the first 8 bits as $(01000010)_2$ The receiver discards the packet. Why?.

Solution

There is an error in this packet. The 4 leftmost bits $(0100)_2$ show the version, which is correct. The next 4 bits $(0010)_2$ 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 19.2

In an IPv4 packet, the value of HLEN is $(1000)_2$. 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 19.3

In an IPv4 packet, the value of HLEN is 5, 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 $(0028)_{16}$ or 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example 19.4

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

$$(45000028000100000102\ \ldots\ )_{16}$$

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

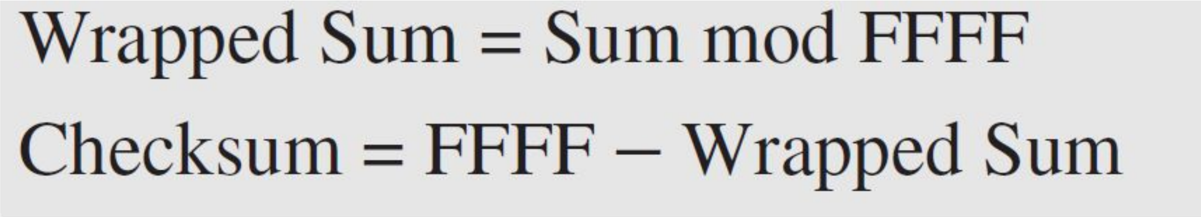
Solution

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 $(01)_{16}$. This means the packet can travel only one hop. The protocol field is the next byte $(02)_{16}$, which means that the upper-layer protocol is IGMP.

Example 19.5

Figure 19.4 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented after wrapping the leftmost digit. The result is inserted in the checksum field.

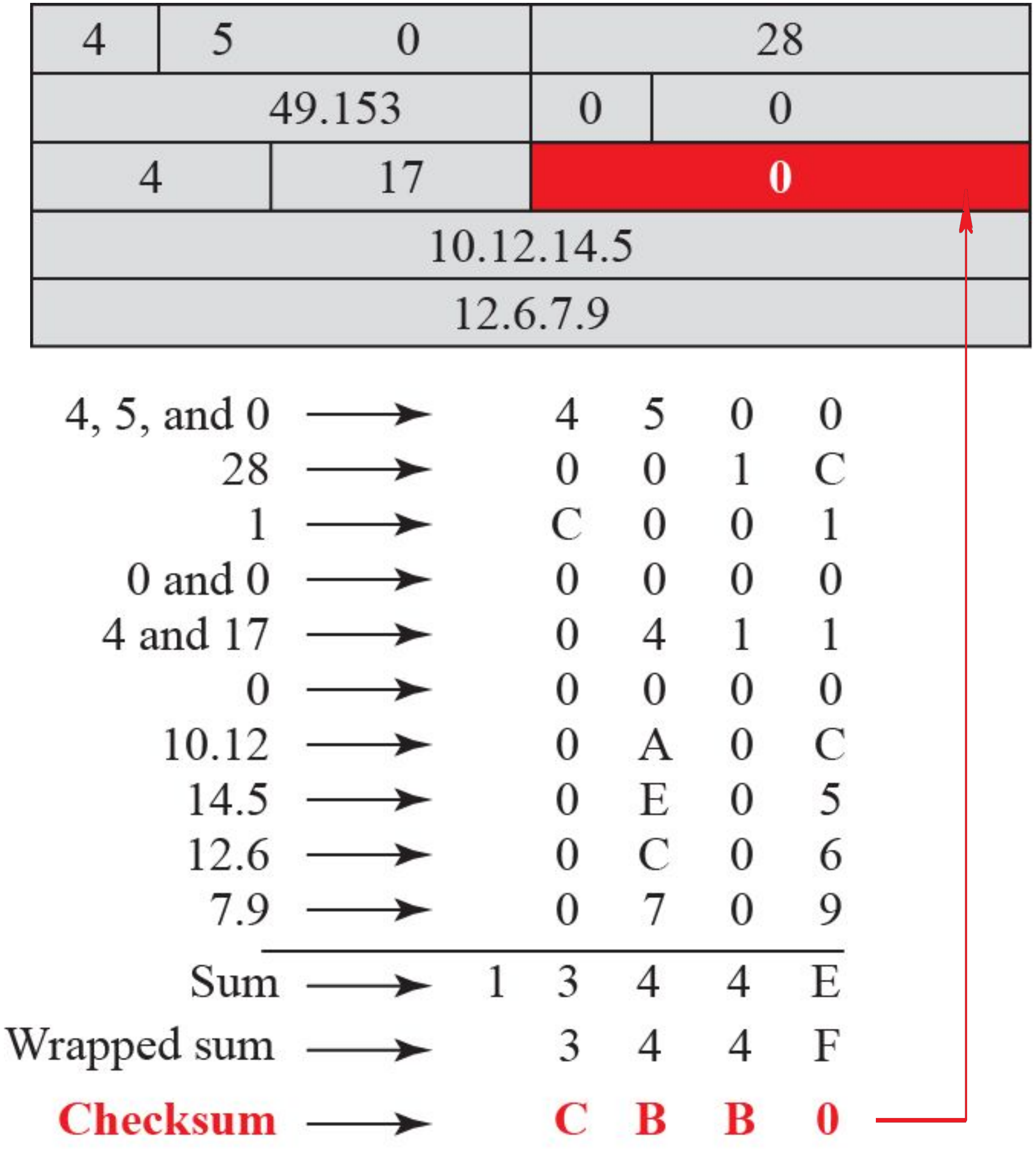
Note that the calculation of wrapped sum and checksum can also be done as follows in hexadecimal:



Wrapped Sum = Sum mod FFFF

Checksum = FFFF – Wrapped Sum

Figure 19.4:



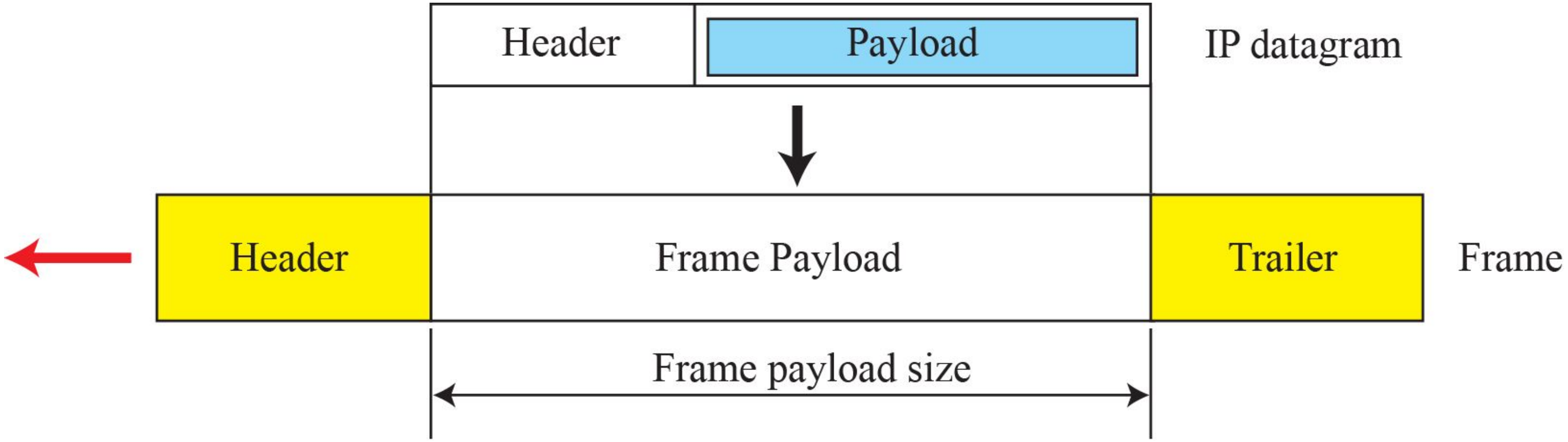
4	5 0	28
49.153	0	0
4	17	0
10.12.14.5		
12.6.7.9		

4, 5, and 0	→		4	5	0	0
28	→		0	0	1	C
1	→		C	0	0	1
0 and 0	→		0	0	0	0
4 and 17	→		0	4	1	1
0	→		0	0	0	0
10.12	→		0	A	0	C
14.5	→		0	E	0	5
12.6	→		0	C	0	6
7.9	→		0	7	0	9
Sum	→	1	3	4	4	E
Wrapped sum	→		3	4	4	F
Checksum	→		**C**	**B**	**B**	**0**

19.19.2 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. 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.

Figure 19.5: Maximum transfer unit (MTU)



MTU: Maximum size of frame payload

Figure 19.6: *Fragmentation example*

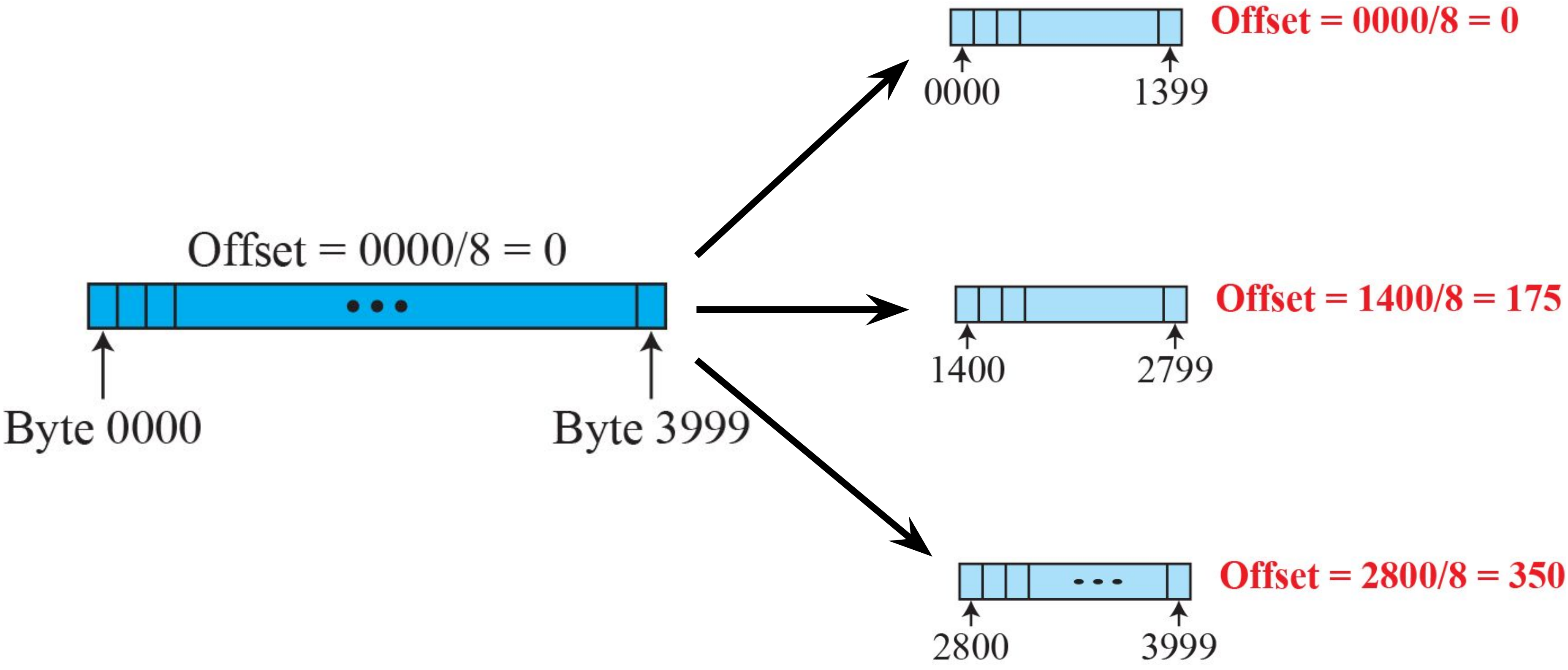
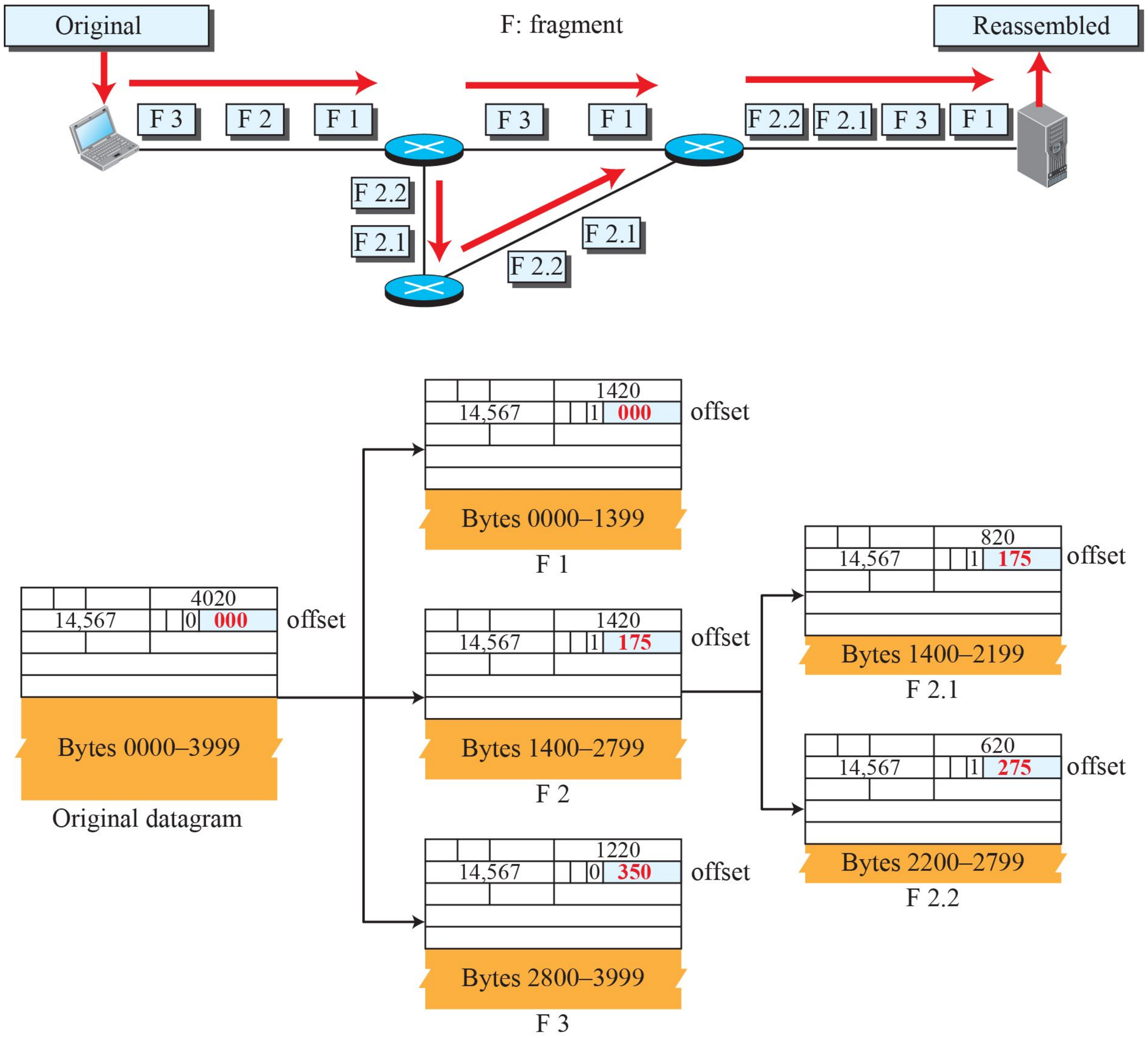


Figure 19.7: Detailed fragmentation example



Example 19.6

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 19.7

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).

Example 19.8

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. 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 19.9

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 19.10

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 are the numbers 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.

19.19.3 Options

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 (in multiples of 4-bytes) to preserve the boundary of the header.

19.19.4 Security of IPv4 Datagrams

The IPv4 protocol, as well as the whole Internet, was started when the Internet users trusted each other. No security was provided for the IPv4 protocol. Today, however, the situation is different; the Internet is not secure anymore. Although we will discuss network security in general and IP security in particular in Chapters 31 and 32, here we give a brief idea about the security issues in IP protocol and the solutions. There are three security issues that are particularly applicable to the IP protocol: packet sniffing, packet modification, and IP spoofing.

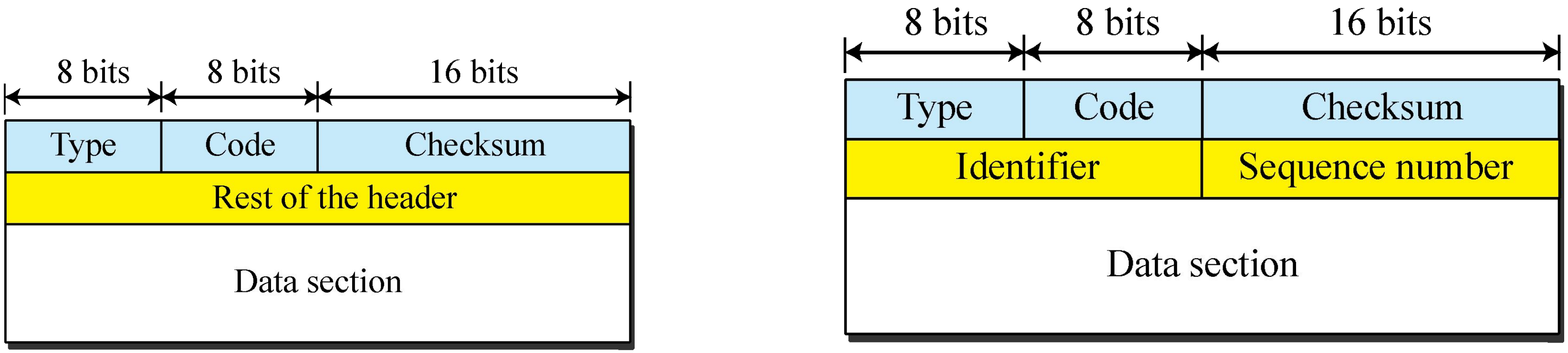
19.2 ICMPv4

The IPv4 has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above two deficiencies.

19.2.1 MESSAGES

ICMP messages are divided into two broad categories: error-reporting messages and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.

Figure 19.8: General format of ICMP messages



Error-reporting messages

Query messages

Type and code values

Error-reporting messages

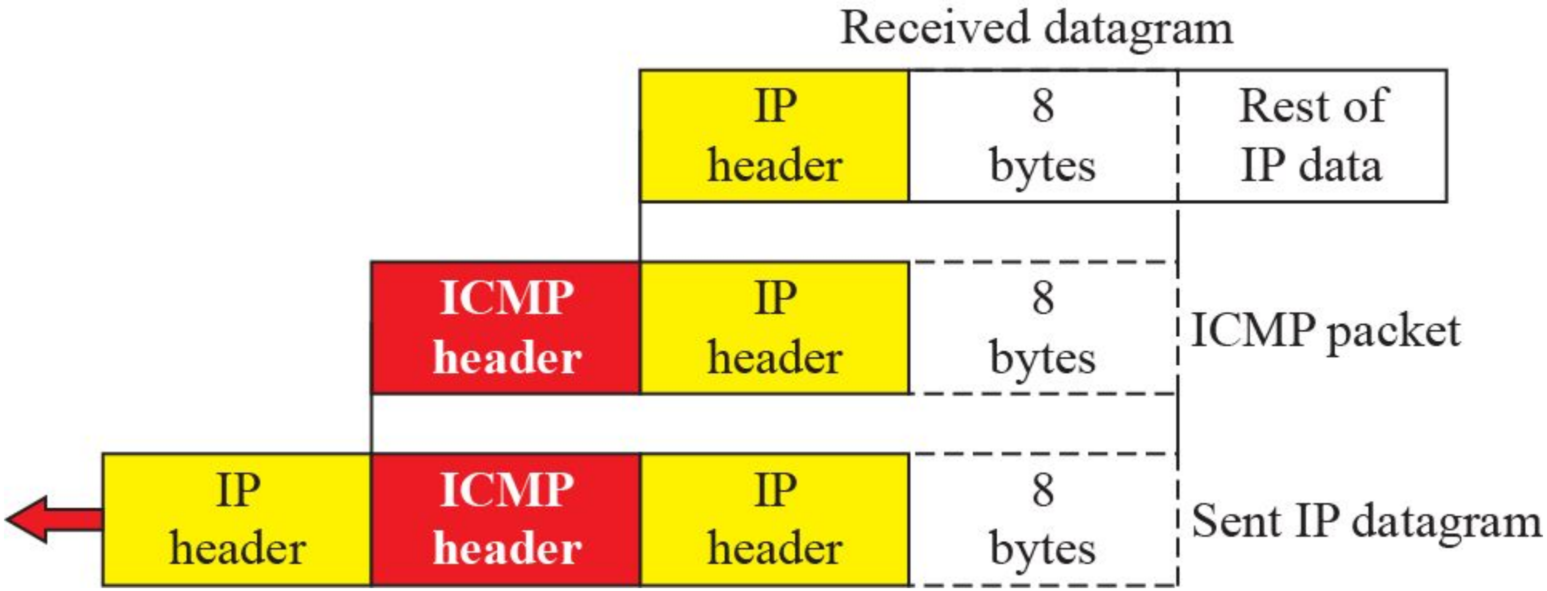
03: Destination unreachable (codes 0 to 15)
04: Source quench (only code 0)
05: Redirection (codes 0 to 3)
11: Time exceeded (codes 0 and 1)
12: Parameter problem (codes 0 and 1)

Query messages

08 and 00: Echo request and reply (only code 0)
13 and 14: Timestamp request and reply (only code 0)

Note: See the book website for more explanation about the code values.

Figure 19.9: Contents of data field for the error messages

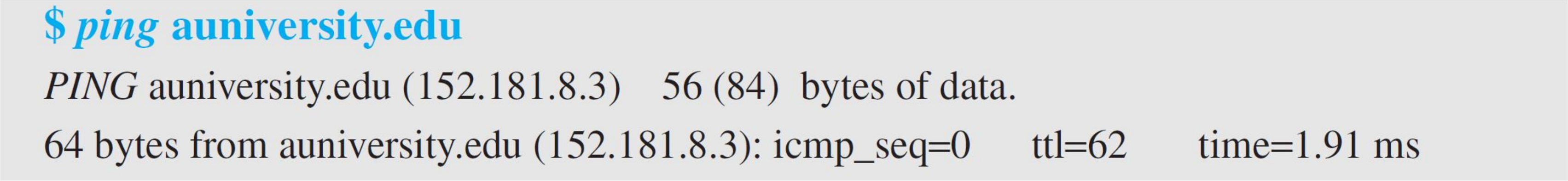


19.2.2 Debugging Tools

There are several tools that can be used in the Internet for debugging. We can determine the viability of a host or router. We can trace the route of a packet. We introduce two tools that use ICMP for debugging: ping and traceroute.

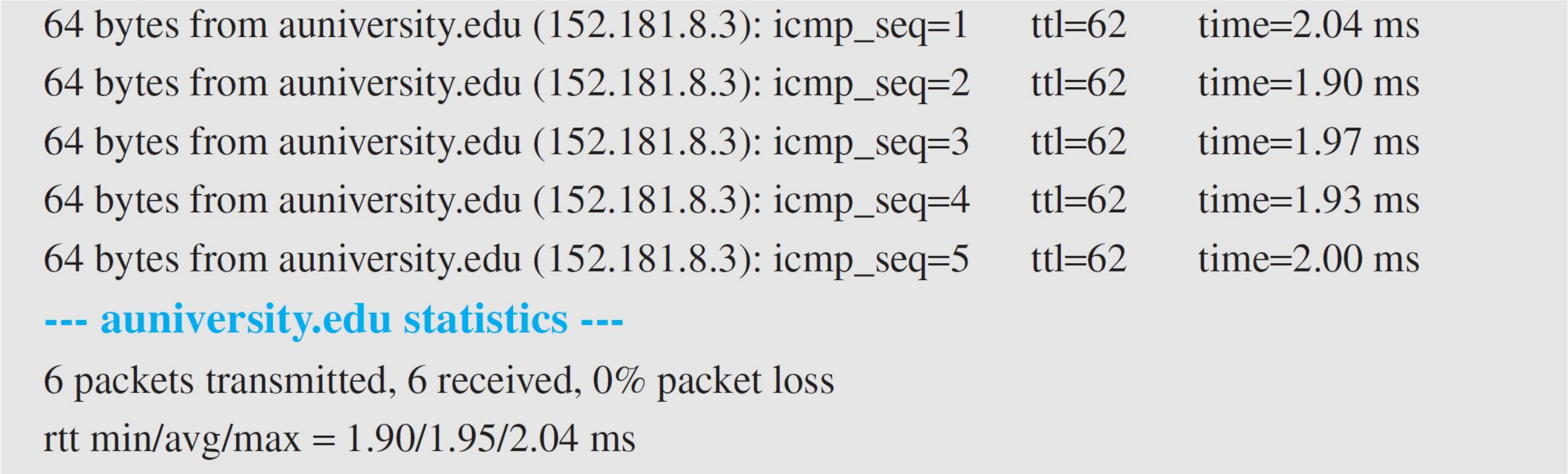
Example 19.11

The following shows how we send a ping message to the auniversity.edu site. We set the identifier field in the echo request and reply message and start the sequence number from 0; this number is incremented by one each time a new message is sent. Note that ping can calculate the round-trip time. It inserts the sending time in the data section of the message. When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (*rtt*).



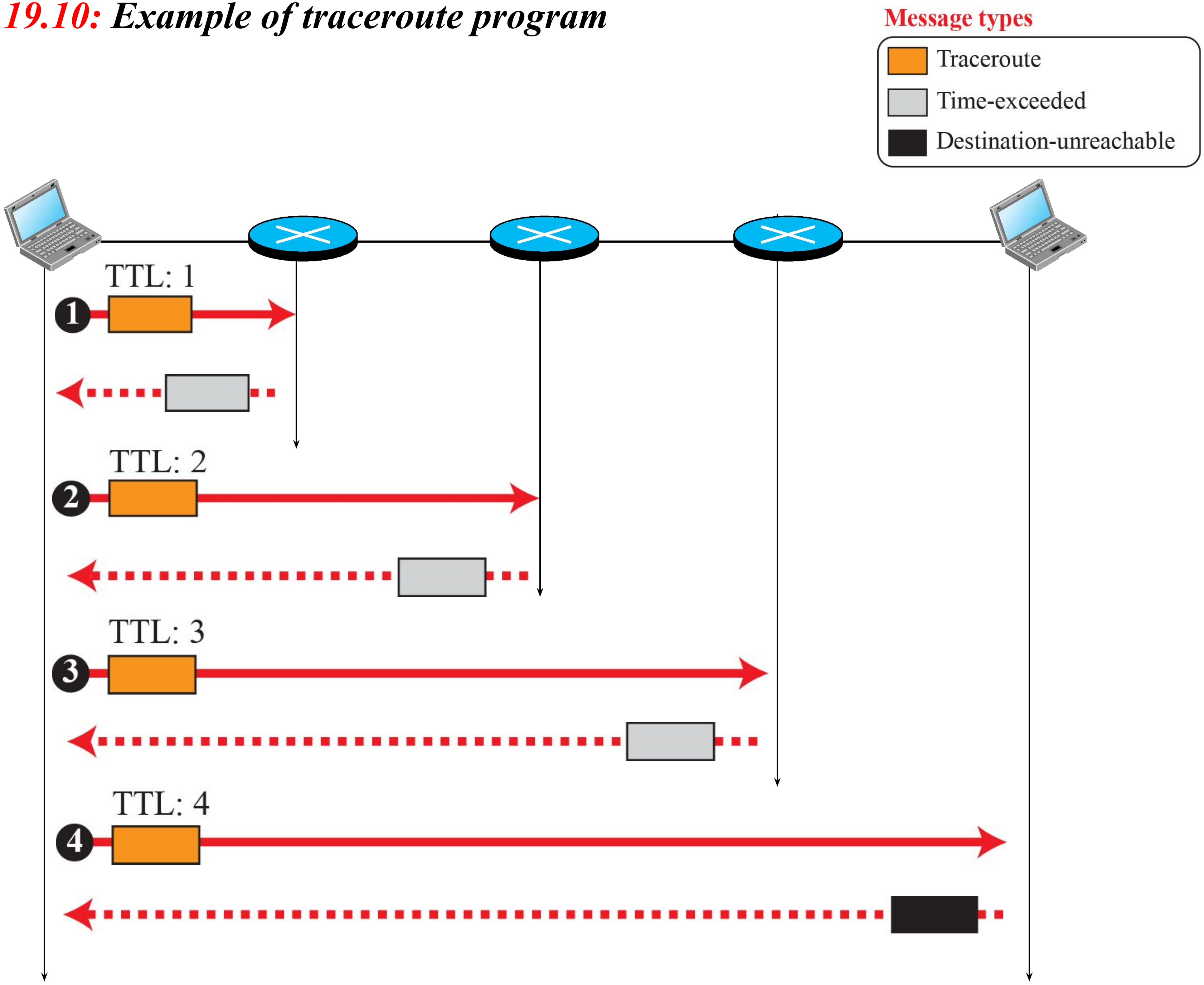
```
$ ping auniversity.edu
PING auniversity.edu (152.181.8.3)    56 (84)  bytes of data.
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=0     ttl=62      time=1.91 ms
```

Example 19.11(continued)



```
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=1    ttl=62     time=2.04 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=2    ttl=62     time=1.90 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=3    ttl=62     time=1.97 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=4    ttl=62     time=1.93 ms
64 bytes from auniversity.edu (152.181.8.3): icmp_seq=5    ttl=62     time=2.00 ms
--- auniversity.edu statistics ---
6 packets transmitted, 6 received, 0% packet loss
rtt min/avg/max = 1.90/1.95/2.04 ms
```

Figure 19.10: Example of traceroute program



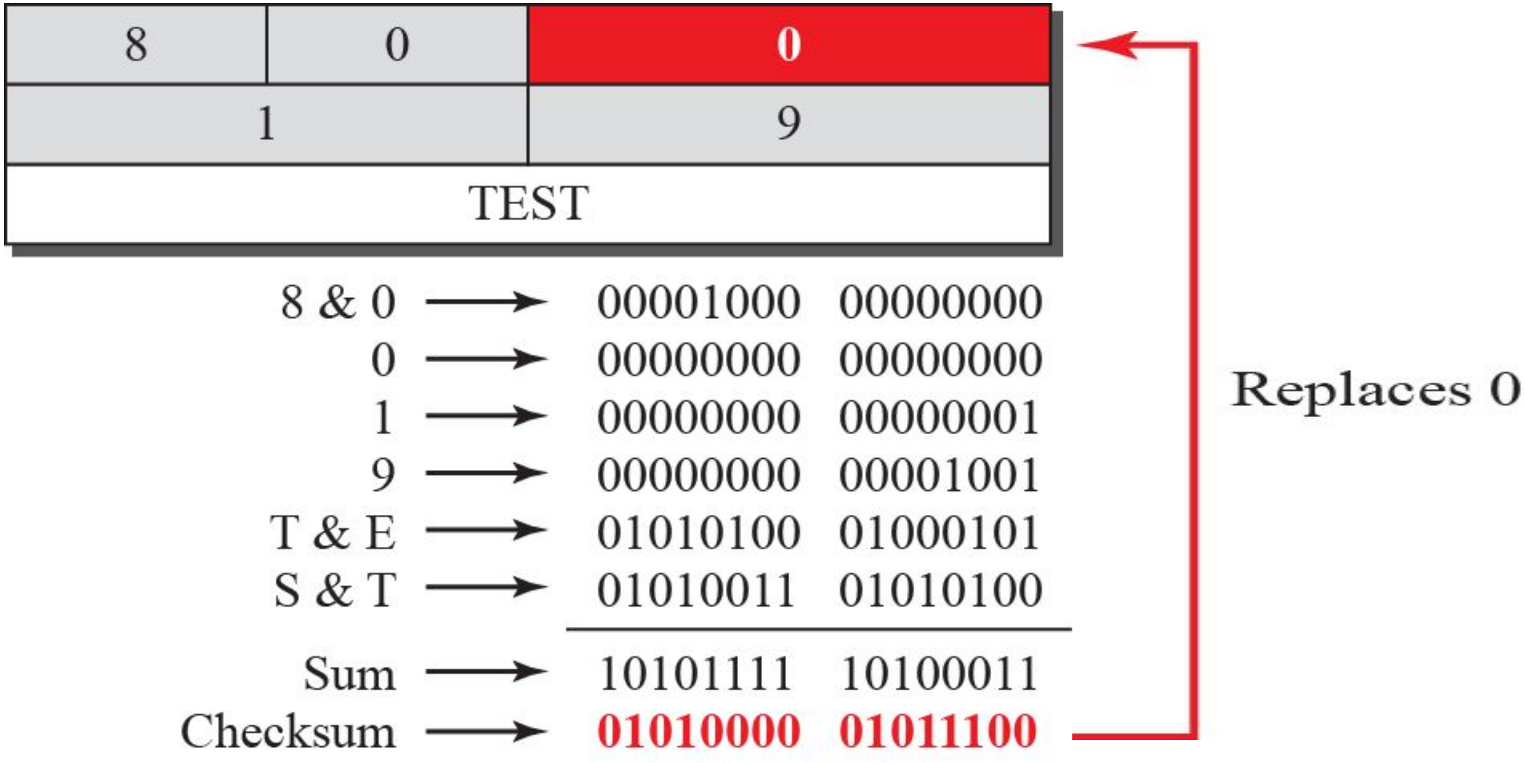
19.2.3 ICMP Checksum

In ICMP the checksum is calculated over the entire message (header and data).

Example 19.12

Figure 19.11 shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.

Figure 19.11: Example of checksum calculation



8	0	**0**
1		9
TEST		

8 & 0	→	00001000	00000000
0	→	00000000	00000000
1	→	00000000	00000001
9	→	00000000	00001001
T & E	→	01010100	01000101
S & T	→	01010011	01010100
Sum	→	10101111	10100011
Checksum	→	**01010000**	**01011100**

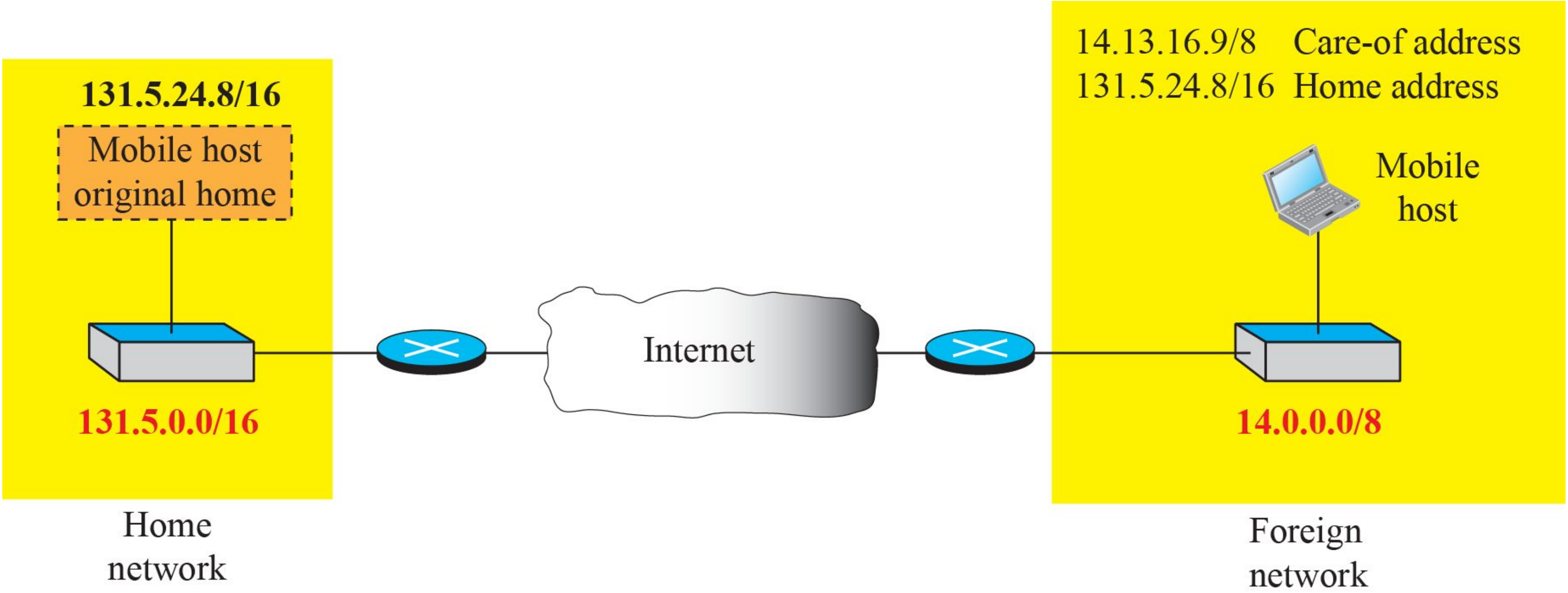
19-3 MOBILE IP

In the last section of this chapter, we discuss mobile IP. As mobile and personal computers such as notebooks become increasingly popular, we need to think about mobile IP, the extension of IP protocol that allows mobile computers to be connected to the Internet at any location where the connection is possible. In this section, we discuss this issue.

19.3.1 Addressing

The main problem that must be solved in providing mobile communication using the IP protocol is addressing.

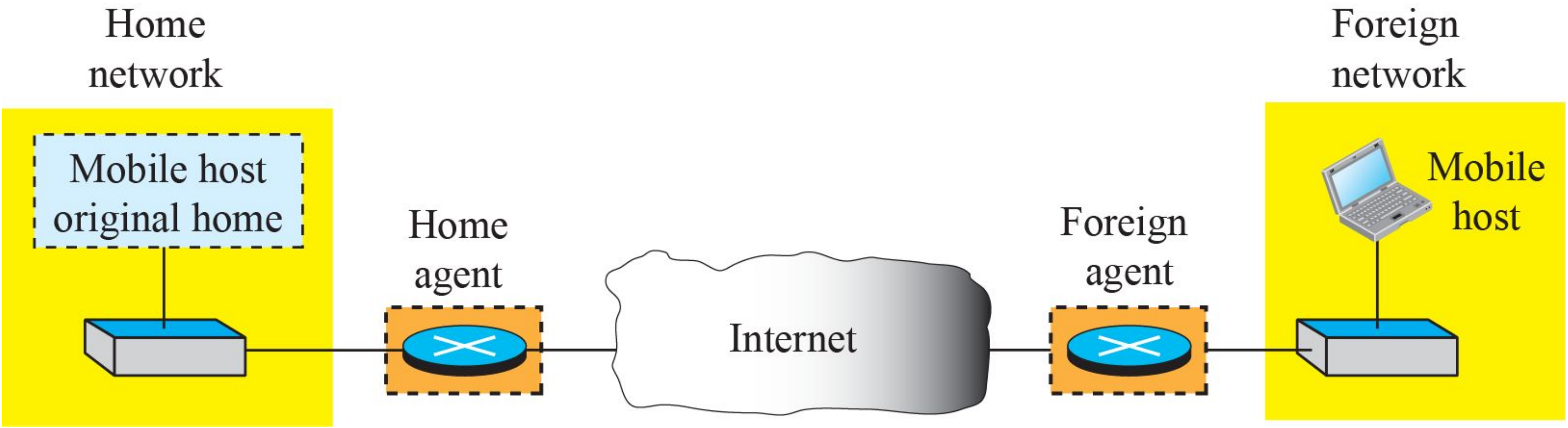
Figure 19.12: Home address and care-of address



19.3.2 Agents

To make the change of address transparent to the rest of the Internet requires a home agent and a foreign agent. Figure 19.13 shows the position of a home agent relative to the home network and a foreign agent relative to the foreign network..

Figure 19.13: Home agent and foreign agent



19.3.3 Three Phases

To communicate with a remote host, a mobile host goes through three phases: agent discovery, registration, and data transfer, as shown in Figure 19.14.

Figure19.14: Remote host and mobile host communication

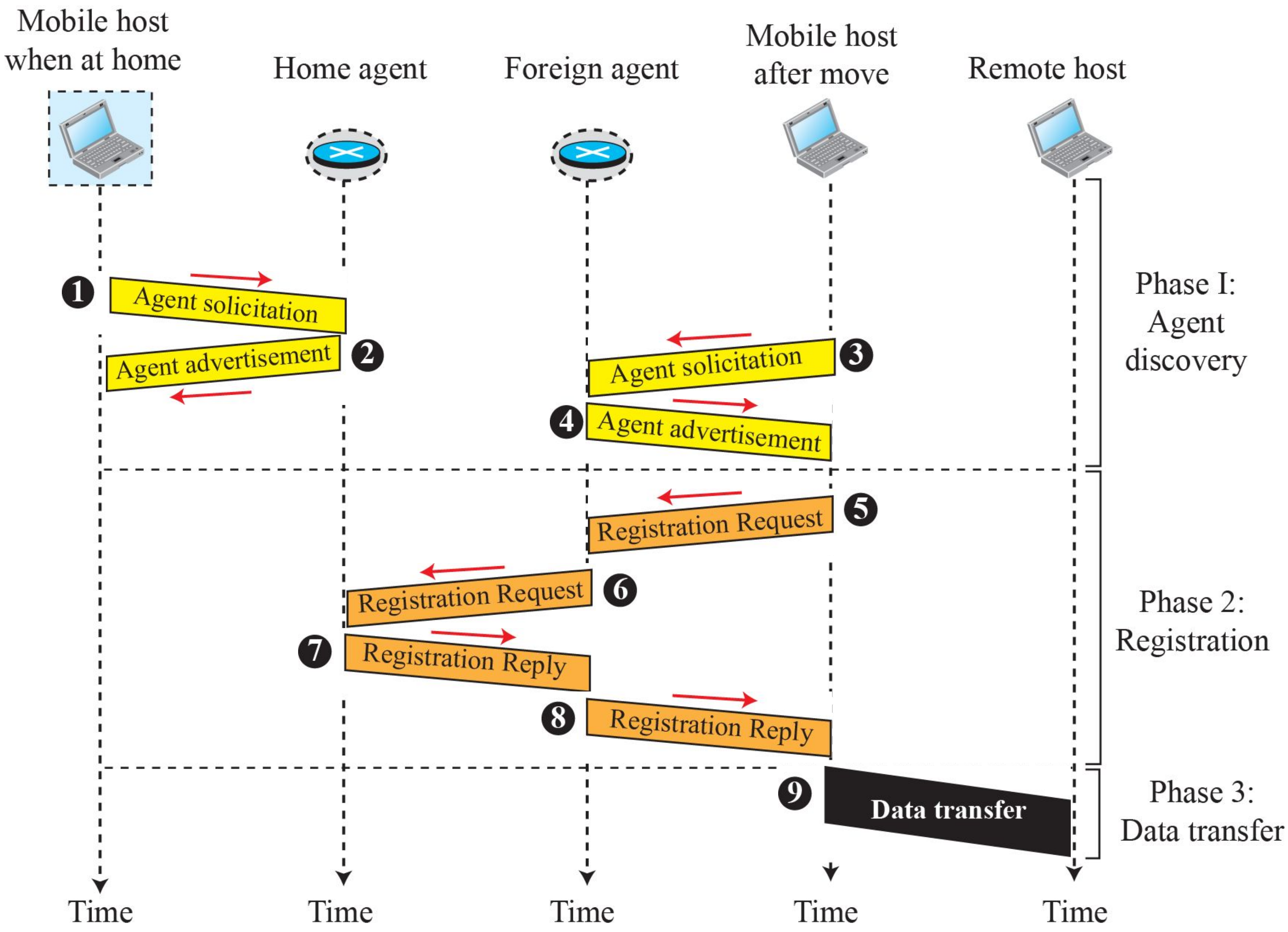
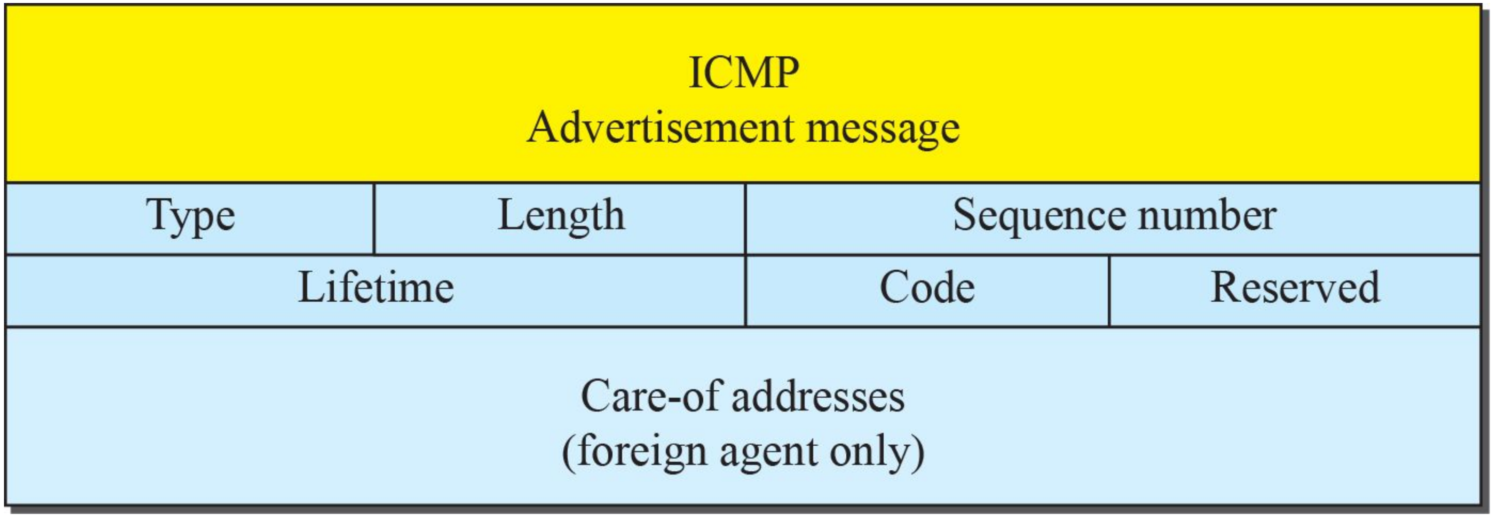


Figure 19.14: Agent advertisement



ICMP
Advertisement message

Type

Length

Sequence number

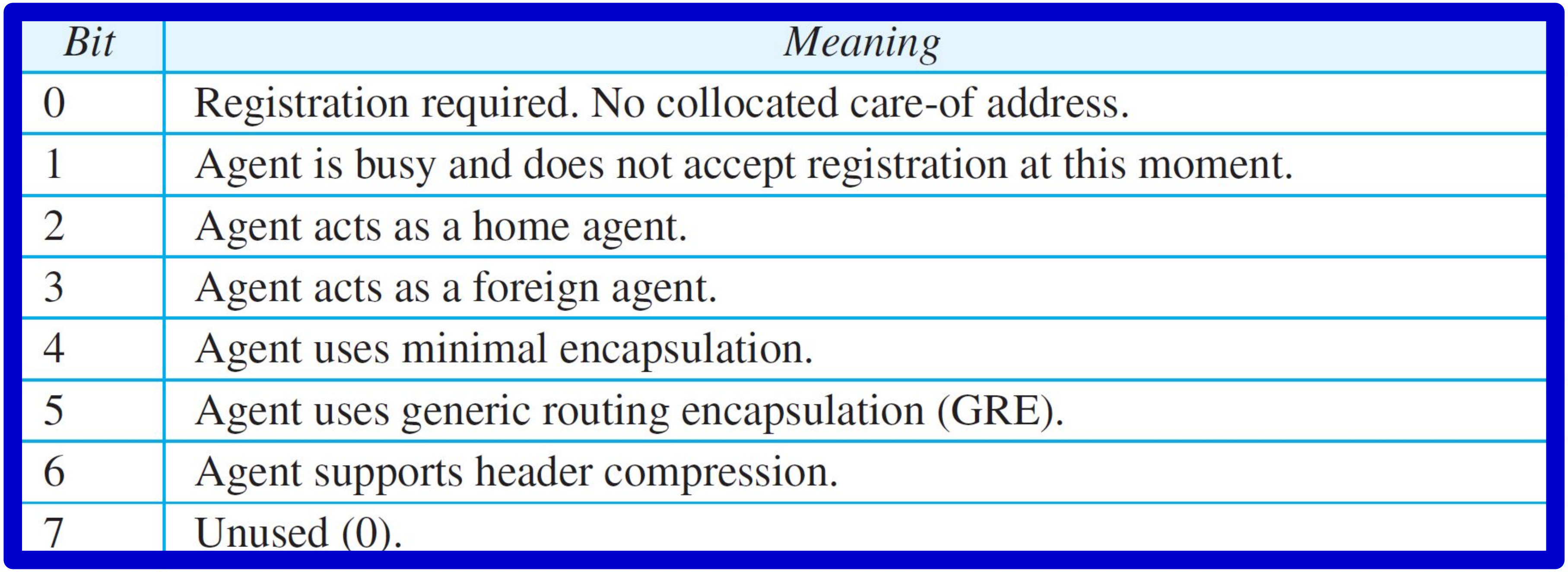
Lifetime

Code

Reserved

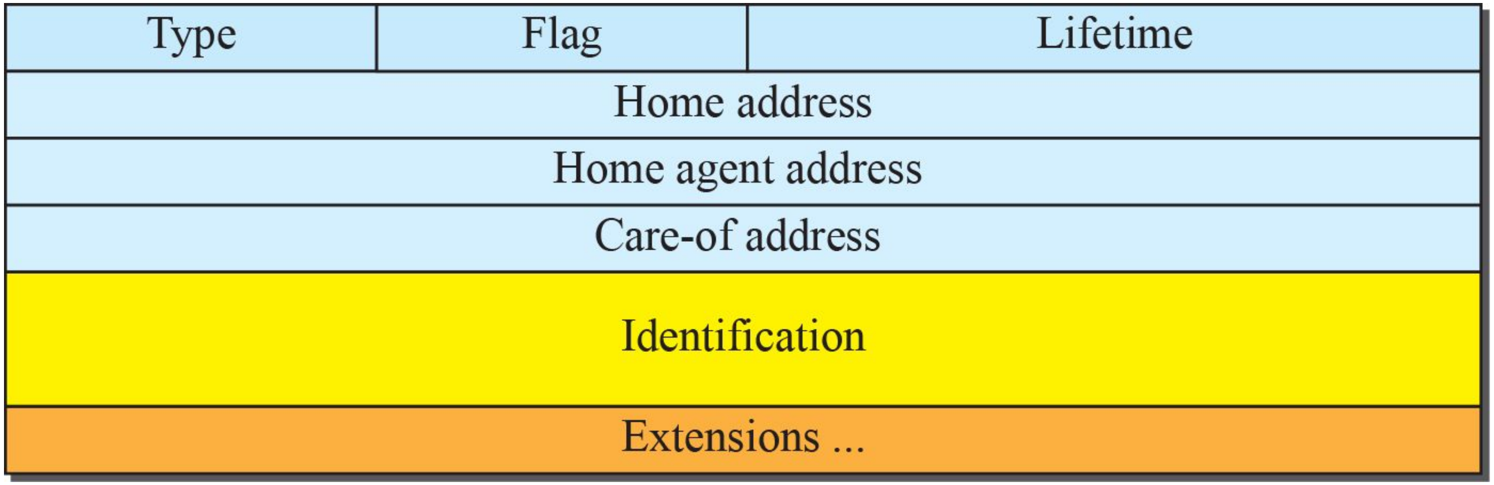
Care-of addresses
(foreign agent only)

Table 19.1: Code Bits



Bit	*Meaning*
0	Registration required. No collocated care-of address.
1	Agent is busy and does not accept registration at this moment.
2	Agent acts as a home agent.
3	Agent acts as a foreign agent.
4	Agent uses minimal encapsulation.
5	Agent uses generic routing encapsulation (GRE).
6	Agent supports header compression.
7	Unused (0).

Figure 19.16: Registration request format



Type

Flag

Lifetime

Home address

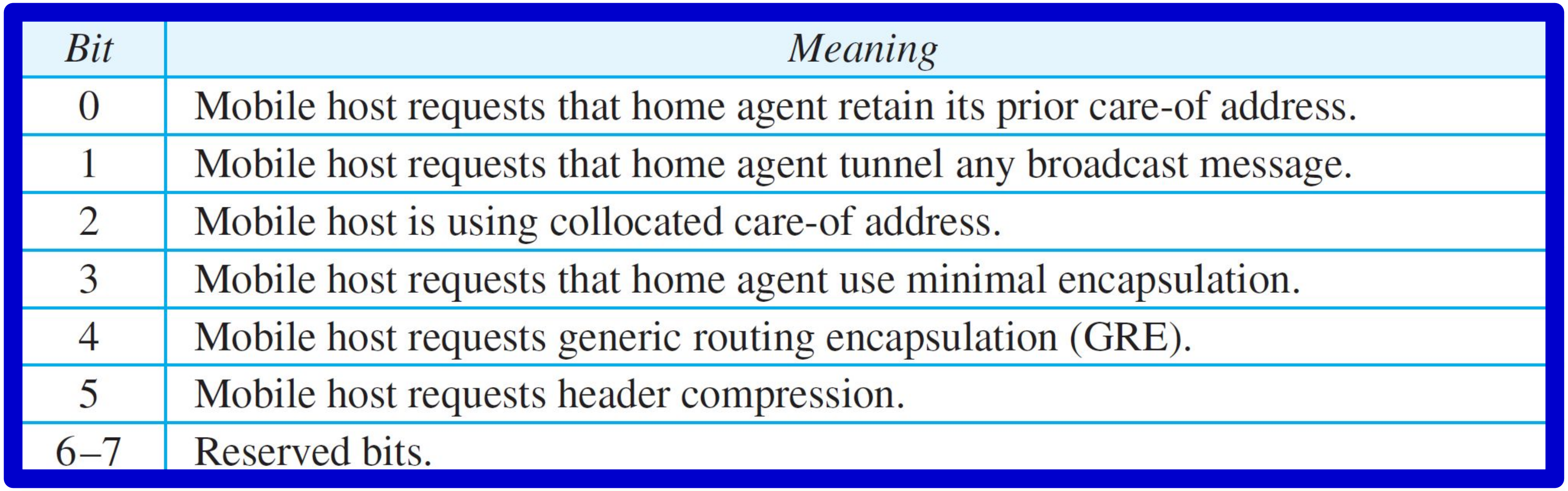
Home agent address

Care-of address

Identification

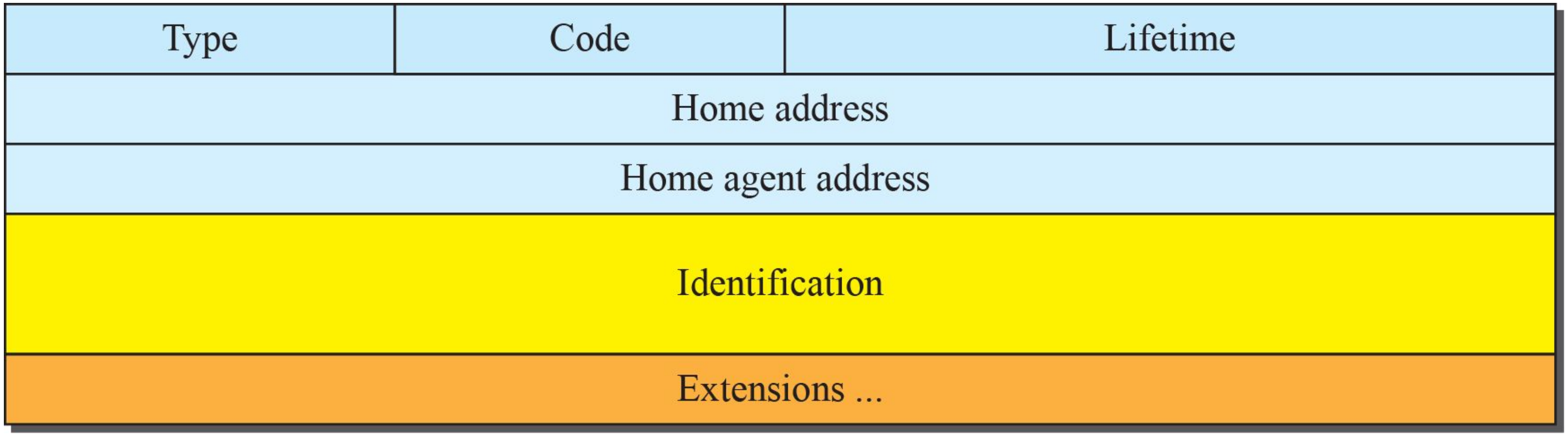
Extensions ...

Table 19.2: Registration request flag field bits



Bit	*Meaning*
0	Mobile host requests that home agent retain its prior care-of address.
1	Mobile host requests that home agent tunnel any broadcast message.
2	Mobile host is using collocated care-of address.
3	Mobile host requests that home agent use minimal encapsulation.
4	Mobile host requests generic routing encapsulation (GRE).
5	Mobile host requests header compression.
6–7	Reserved bits.

Figure 19,17: ***Registration reply format***



Type

Code

Lifetime

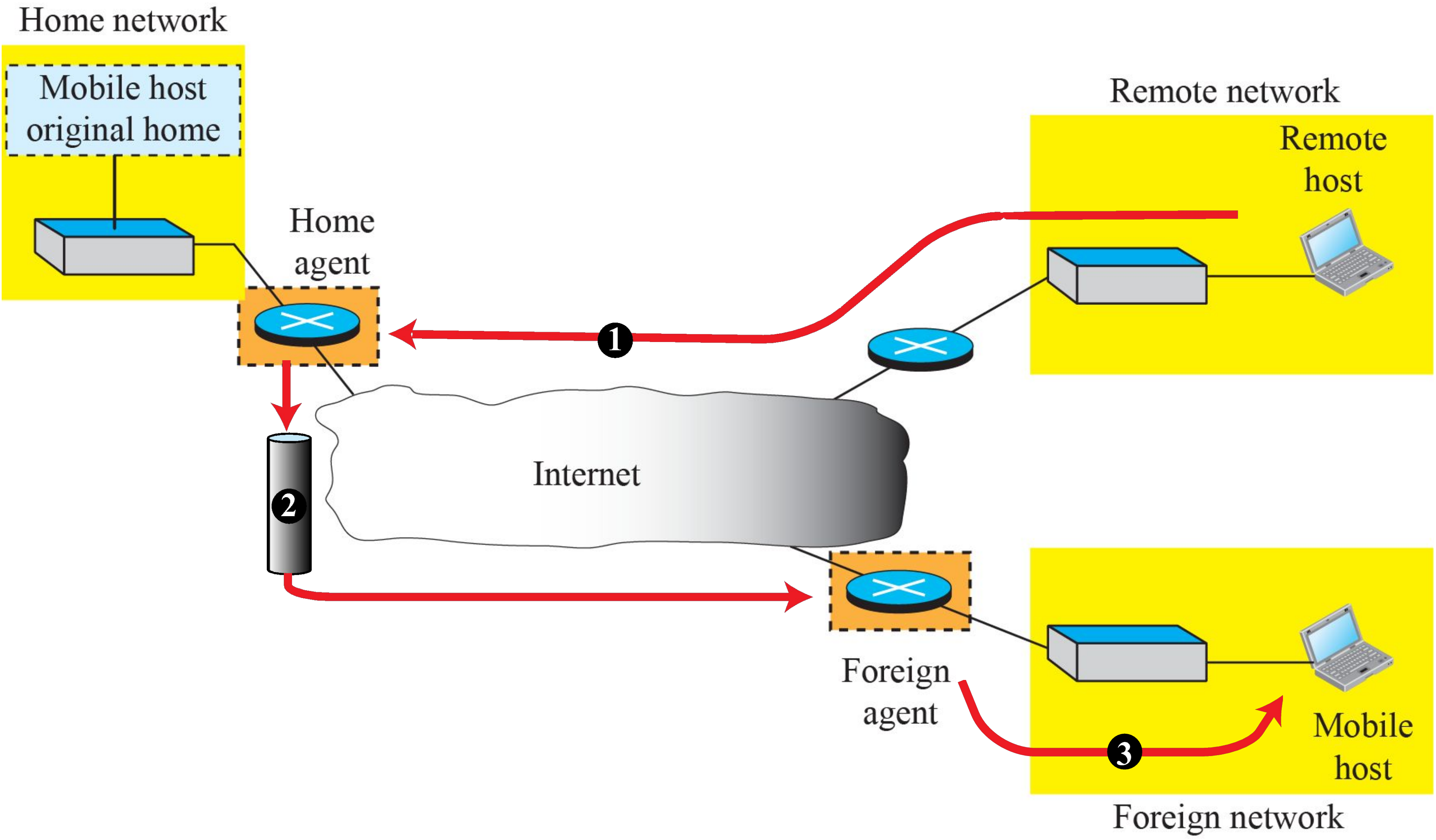
Home address

Home agent address

Identification

Extensions ...

Figure 19.18: Data transfer



19.3.4 Inefficiency in Mobile IP

Communication involving mobile IP can be inefficient. The inefficiency can be severe or moderate. The severe case is called double crossing or 2X. The moderate case is triangle routing or dog-leg routing.

Figure 19.19: Double crossing

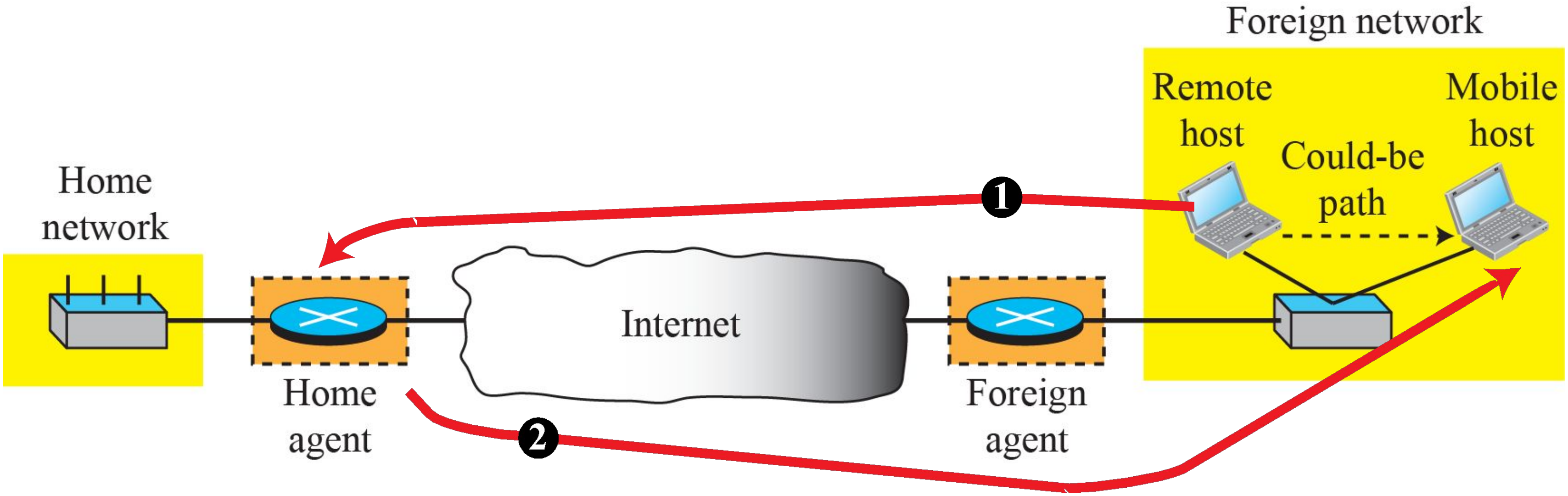
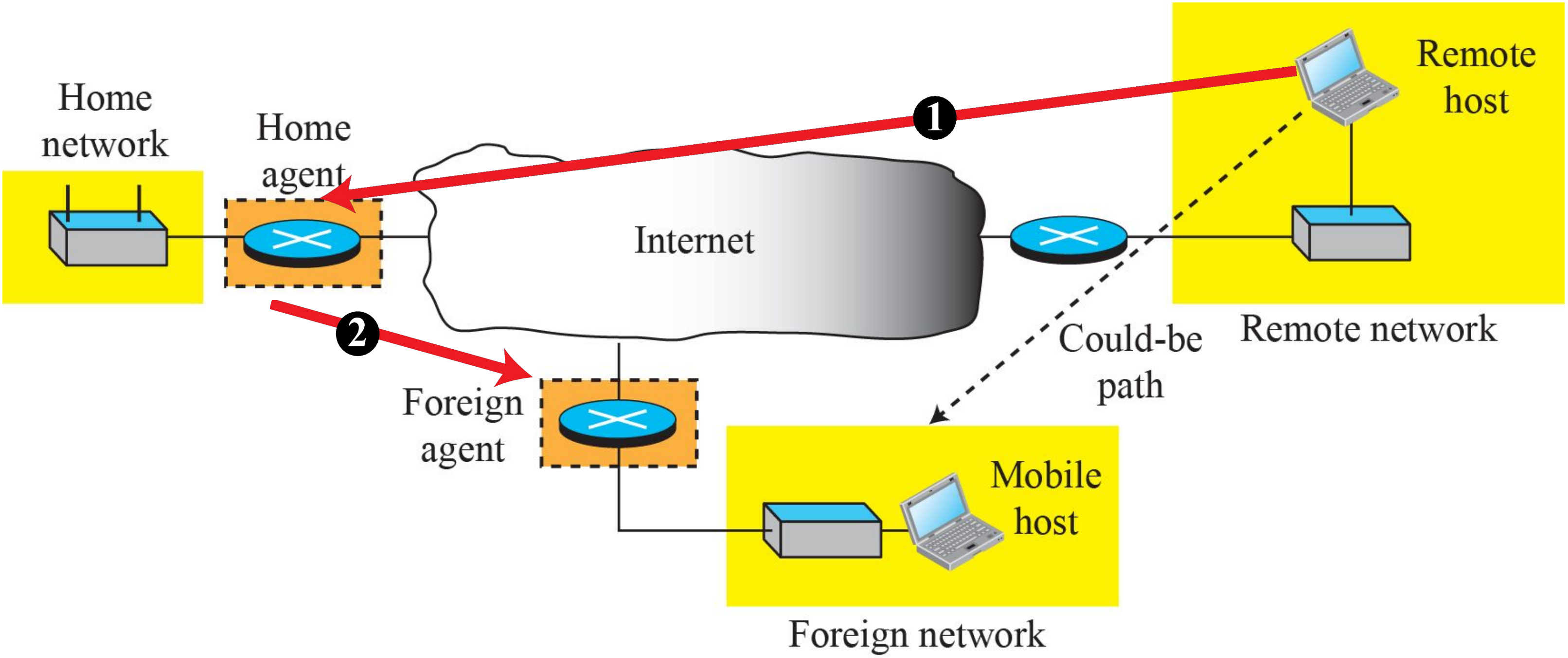


Figure 19.20: *Triangle routing*


Remote host
Home network
Home agent
1
Internet
Remote network
2
Could-be path
Foreign agent
Mobile host
Foreign network