Chapter 18: Introduction to Network Layer

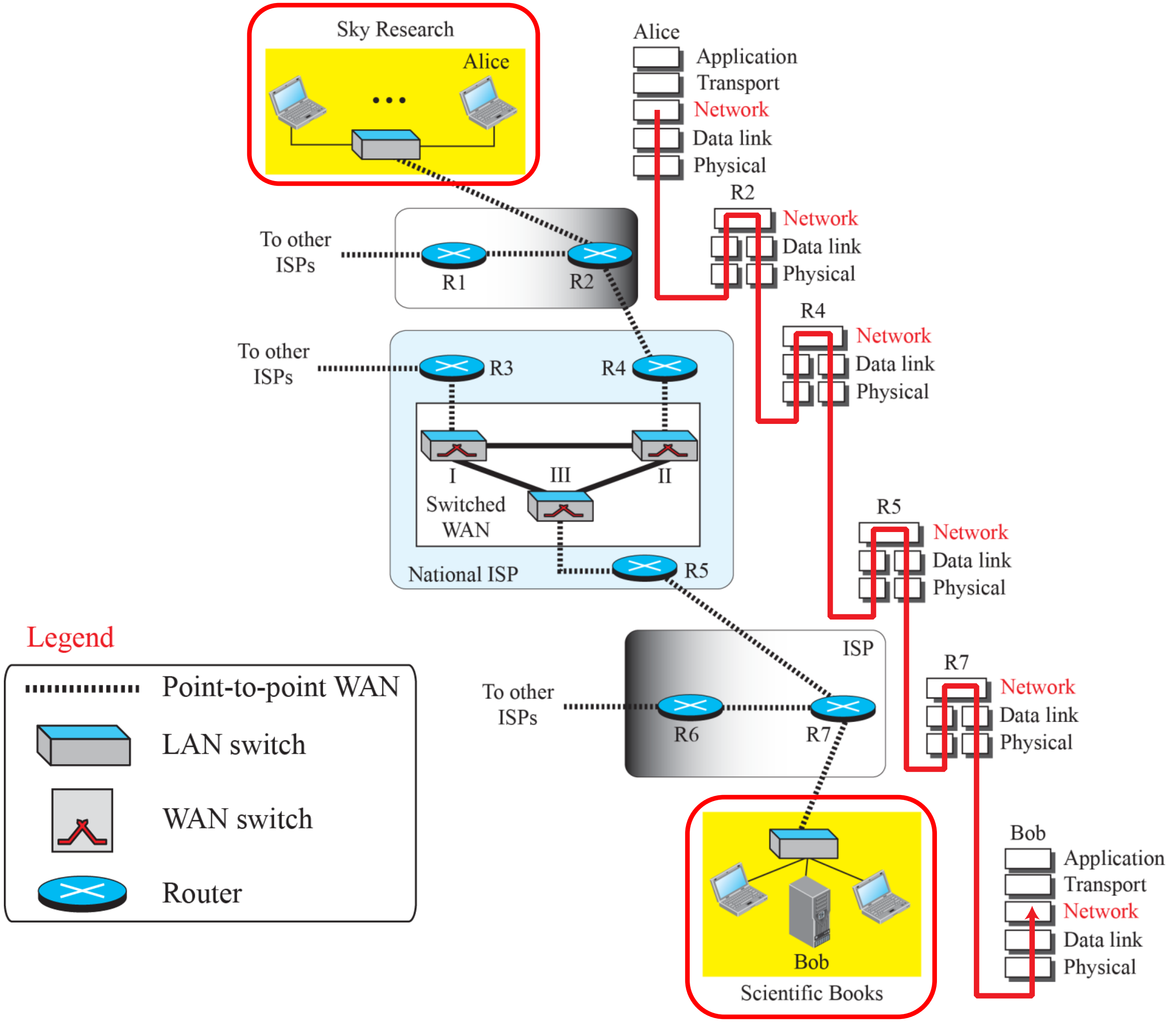
Outline

18.1 NETWORK-LAYER SERVICES

18.2 PACKET SWITCHING

18.4 IPv4 ADDRESSES

Figure 18.1: Communication at the network layer



18.1 Network-Layer Services

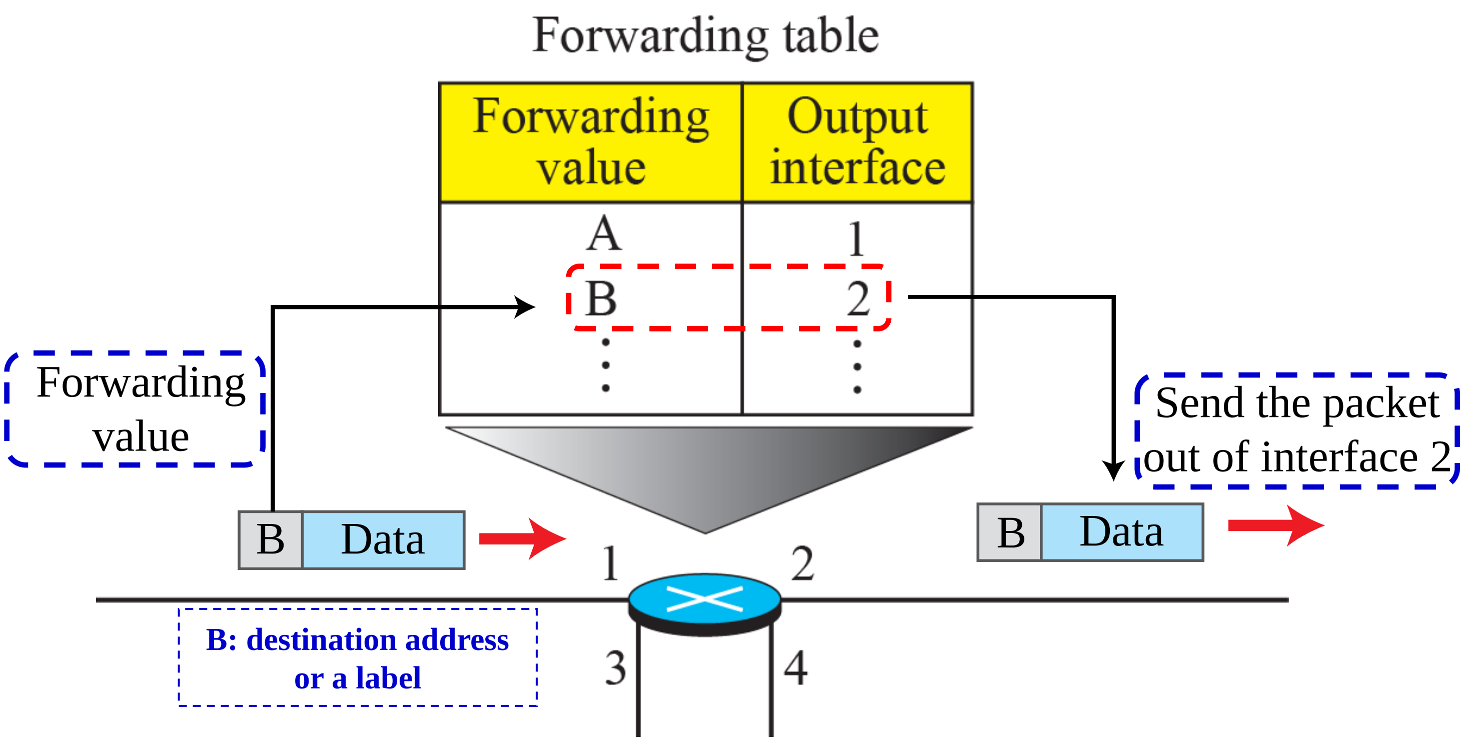
Before discussing the network layer in the Internet today, let's briefly discuss the network-layer services (packetizing, routing, forwarding) that, in general, are expected from a network-layer protocol. In addition, other services (error control, flow control, congestion control, quality of service and security) may also be expected.

Packetizing: encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination. Note that the network layer carries a payload from the source to the destination without changing or using it.

Routing: there is more than one route from the source to the destination. The network layer is responsible for applying strategies and running routing protocols to find the best one among these possible routes and create routing tables for each router.

18.1 Network-Layer Services

***Forwarding*: is the *<u>action applied</u>* by each router when *<u>a packet arrives</u>* at one of its *<u>interfaces</u>*, i.e., to *<u>forward the packet</u>* to *<u>another</u>* (unicast) or *<u>some</u>* (multicast) attached network(s).**



18-2 PACKET SWITCHING

From the discussion of routing and forwarding, we infer that a kind of switching occurs at the network layer.

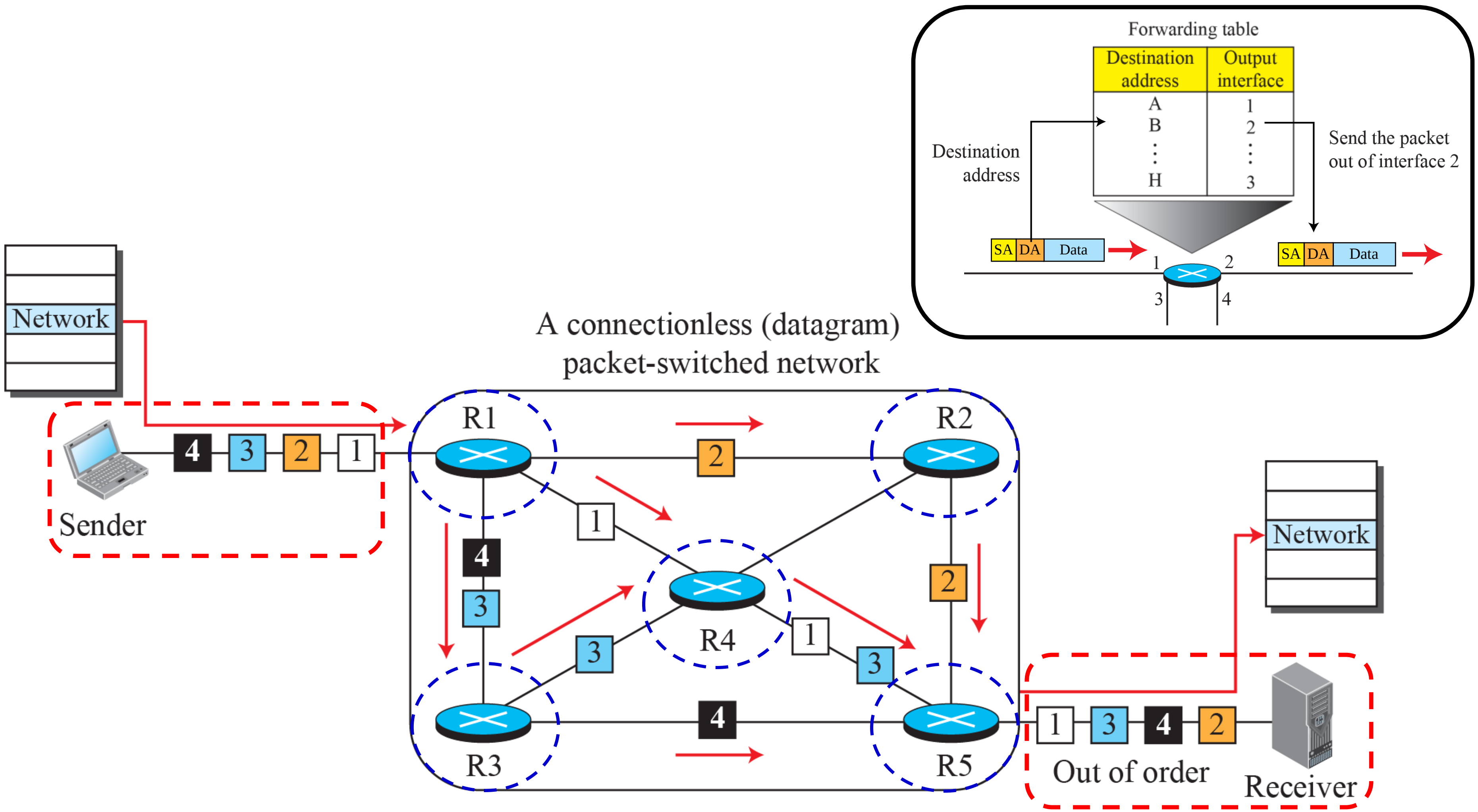
A <u>router</u>, in fact, is a <u>switch</u> that <u>creates a connection</u> between an <u>input port</u> and an <u>output port</u> (or a set of output ports), just as an electrical switch connects the input to the output to let electricity flow.

18.2.1 Datagram Approach

When the Internet started, the network layer was designed to provide a connectionless service in which the network-layer protocol treats each packet independently, with each packet having no relationship to any other packet. The idea was that the network layer is only responsible for delivery of packets from the source to the destination.

In this approach, the packets in a message may or may not travel the same path to their destination.

Figure 18.3: A connectionless packet-switched network



18-4 IPv4 ADDRESSES

The <u>identifier</u> used in the <u>network layer</u> of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or <u>IP address</u>. An IPv4 address is a <u>32-bit address</u> that <u>uniquely and universally</u> defines the <u>connection</u> of a host or a router to the Internet.

The IP address is the address of the <u>connection</u>, <u>not the host or the router</u>.

18.4.1 Address Space

A protocol like IPv4 that defines addresses has an <u>address space</u>. An address space is the <u>total number of addresses used by the protocol</u>. IPv4 uses <u>32-bit addresses</u>, which means that the address space is 2^{32}. If there were no restrictions, more than 4 billion devices could be connected to the Internet. A 32-bit IPv4 address can be notated using binary, dotted decimal and hexadecimal.

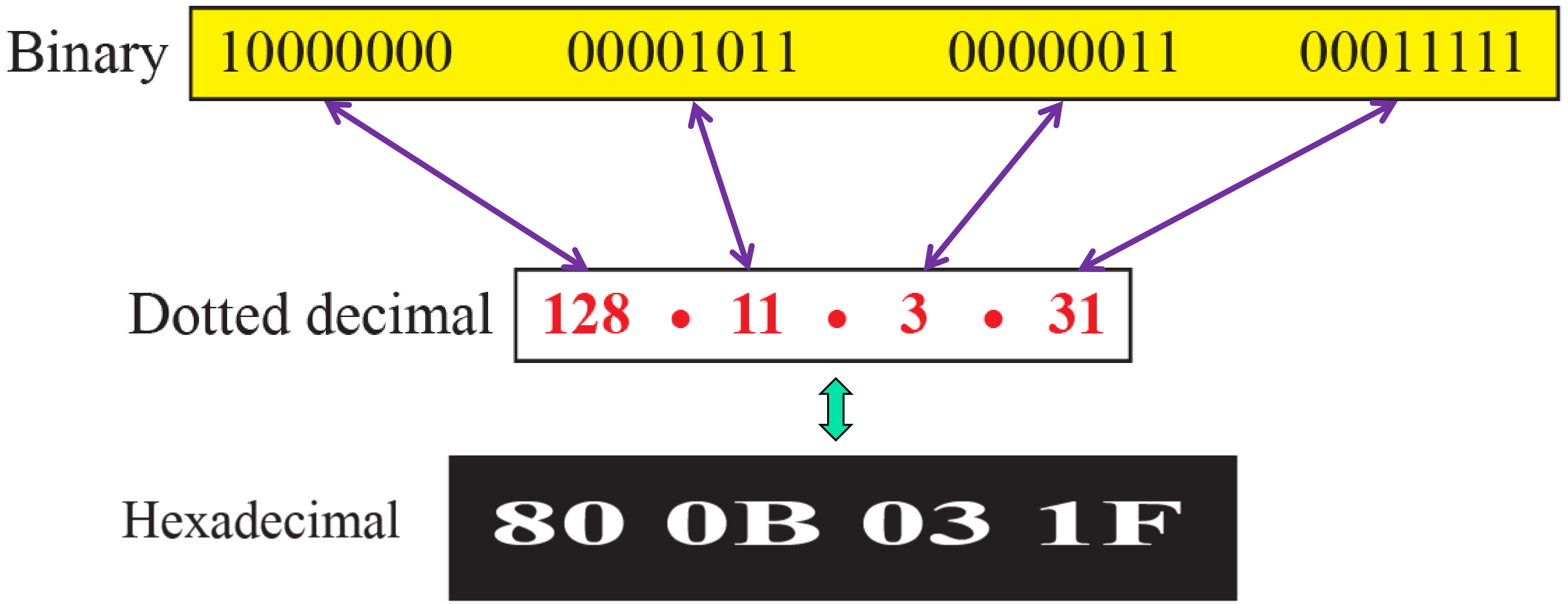
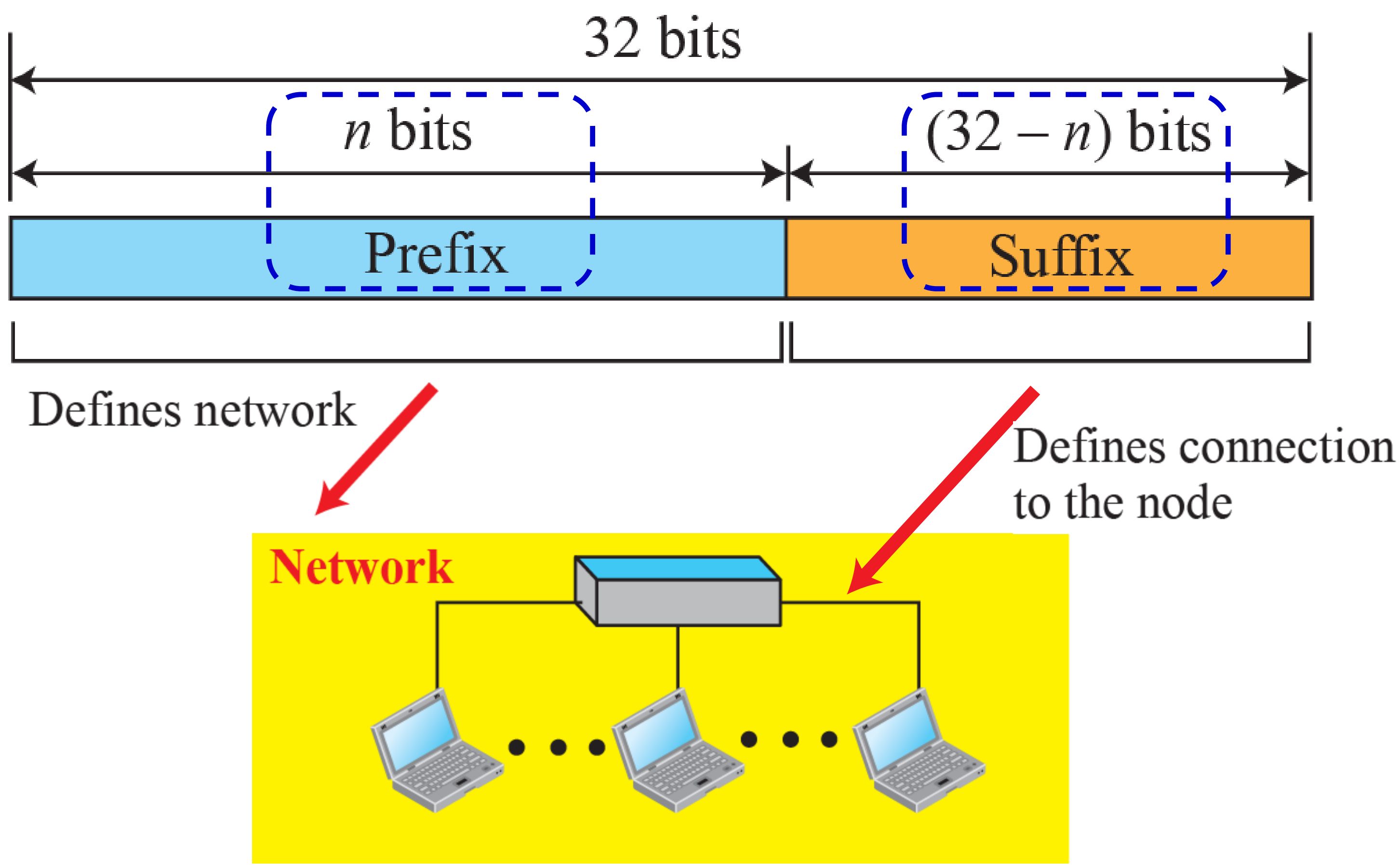


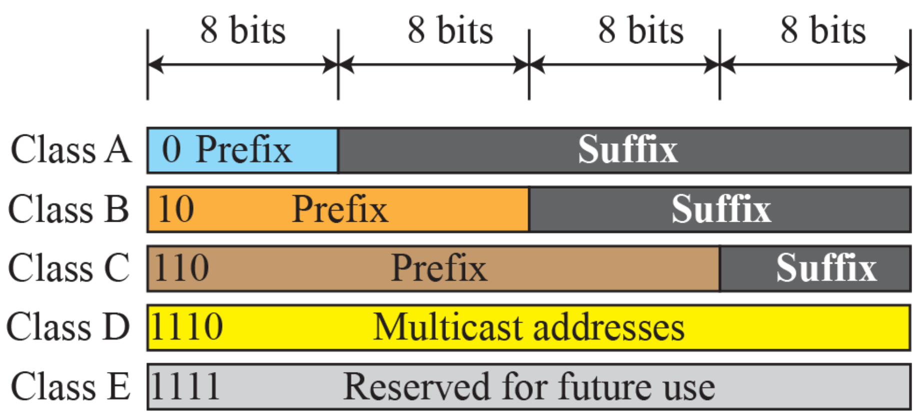
Figure 18.17: ***Hierarchy in addressing***

A 32-bit IPv4 address is hierarchical and divided into two parts: the first part of the address is called the prefix (fixed- or variable- length) and defines the network; the second part of the address is called the suffix and defines the connection to the node.

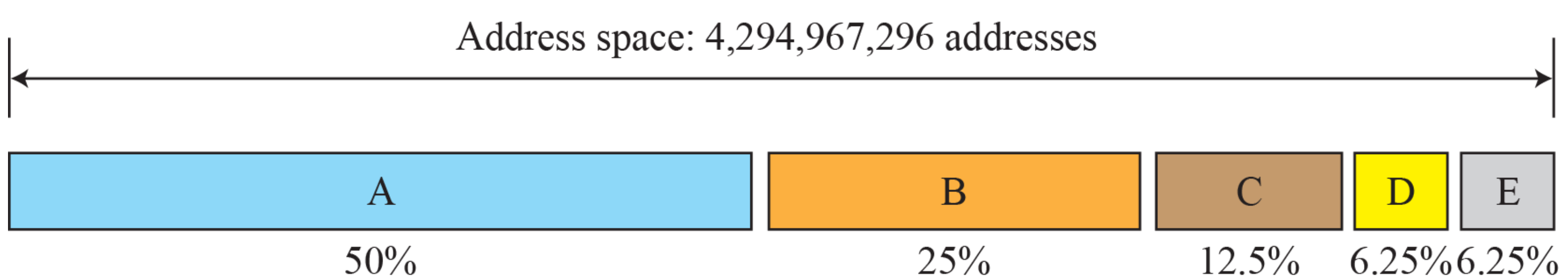


18.4.2 Classful Addressing

When the Internet started, an IPv4 address was designed with a <u>fixed-length prefix</u>: to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24). This scheme is referred to as <u>classful addressing</u> but is <u>obsolete</u>.



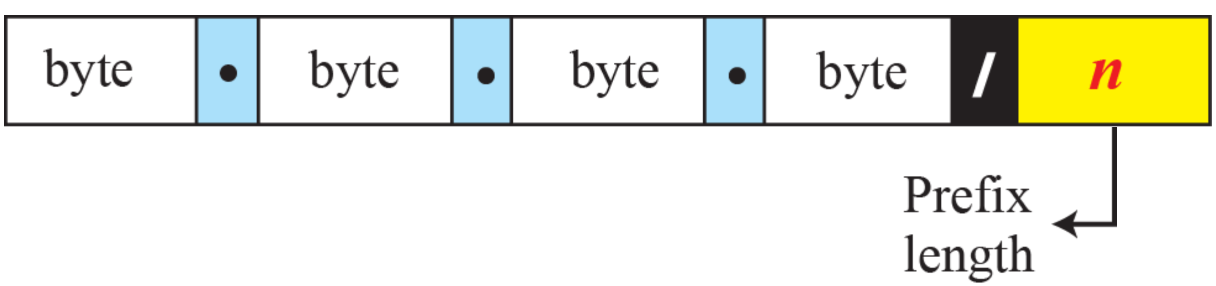
Class	Prefixes	First byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255



18.4.3 Classless Addressing

With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. Although the long-term solution has already been devised and is called IPv6 (128-bit addresses with $2^{128} = 340 \text{ x } 10^{36}$), a short-term solution was also devised to use the same address space but to change the distribution of addresses (as well as subnetting and supernetting) to provide a fair share to each organization.

The short-term solution still uses IPv4 addresses and is referred to as classless addressing. Note that since the prefix length is not inherent in the address, it is added to the address separated by a slash. The notation is formally know as classless interdomain routing or CIDR.



Examples:
12.24.76.8/**8**
23.14.67.92/**12**
220.8.24.255/**25**

Example

A classless address is given as 167.199.170.82/**27**.

a) How many addresses are there in the network?

b) What is the first address and what is the last address?

Solution:

a) The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses.

b) The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

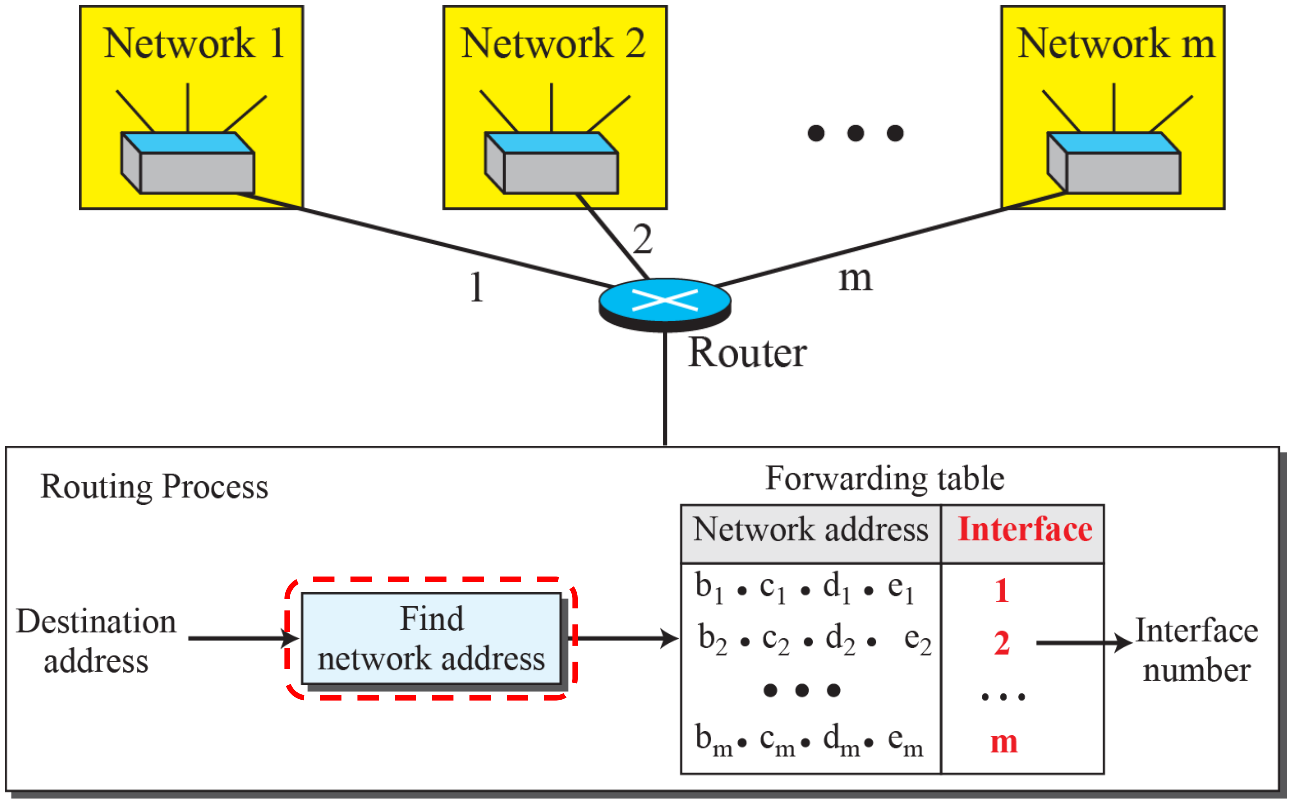
Address: 167.199.170.82/**27**	10100111	11000111	10101010	01010010
First address: 167.199.170.64/**27**	10100111	11000111	10101010	01000000

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Last address: 167.199.170.95/**27**	10100111	11000111	10101010	01011111

Figure 18.22: Network address

When a packet arrives at the router from any source host, the router needs to know which interface (i.e., to which network) the packet should be sent out.



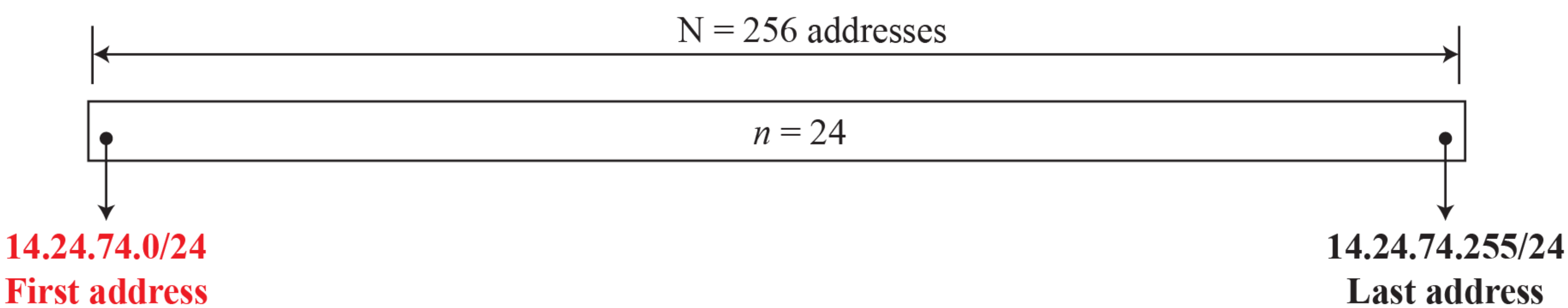
Additional Information: Chapter 18.5 (Forwarding of IP Packets, Longest Mask Matching):
To determine the network address (and hence the corresponding interface to send the packet out), 1) mask the prefix length with the destination address using the logical AND operation and 2) select the interface with the longest mask match. (The prefix length (network part of the address), is indicated by the number of msb 1s in the mask: e.g., "/16" denotes 11111111 11111111 00000000 00000000.)

Problem

An organization is granted a block of addresses with the beginning address 14.24.74.0/**24**. The organization needs to have 3 subblocks of addresses to use in its three **subnets**: one subblock of 10 addresses, one subblock of 60 addresses and one subblock of 120 addresses. Design the subblocks by assigning addresses to subblocks, starting with the largest and ending with the smallest one.

Solution

There are $2^{32-24} = 256$ addresses in this block. The first address is 14.24.74.0/**24**; the last address is 14.24.74.255/**24**.

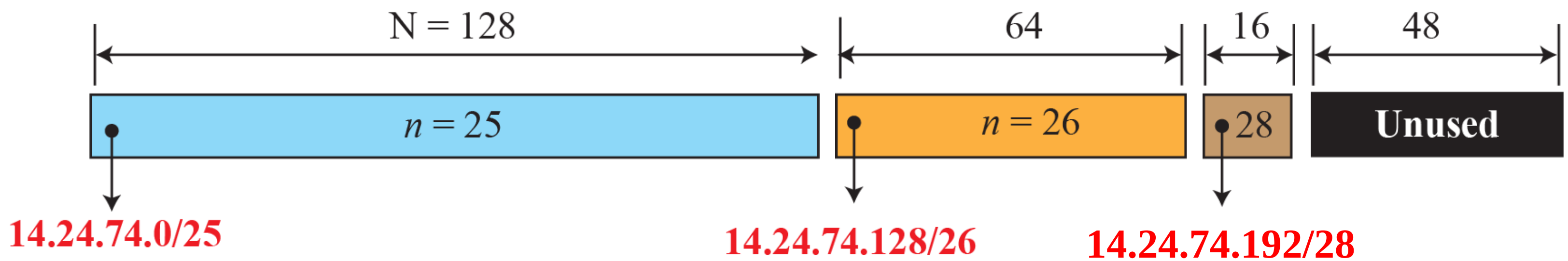


Problem (cont'd)

a. The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as $n_1 = 32 - \log_2 128 = 25$. The first address in this block is 14.24.74.0/**25**; the last address is 14.24.74.127/**25**.

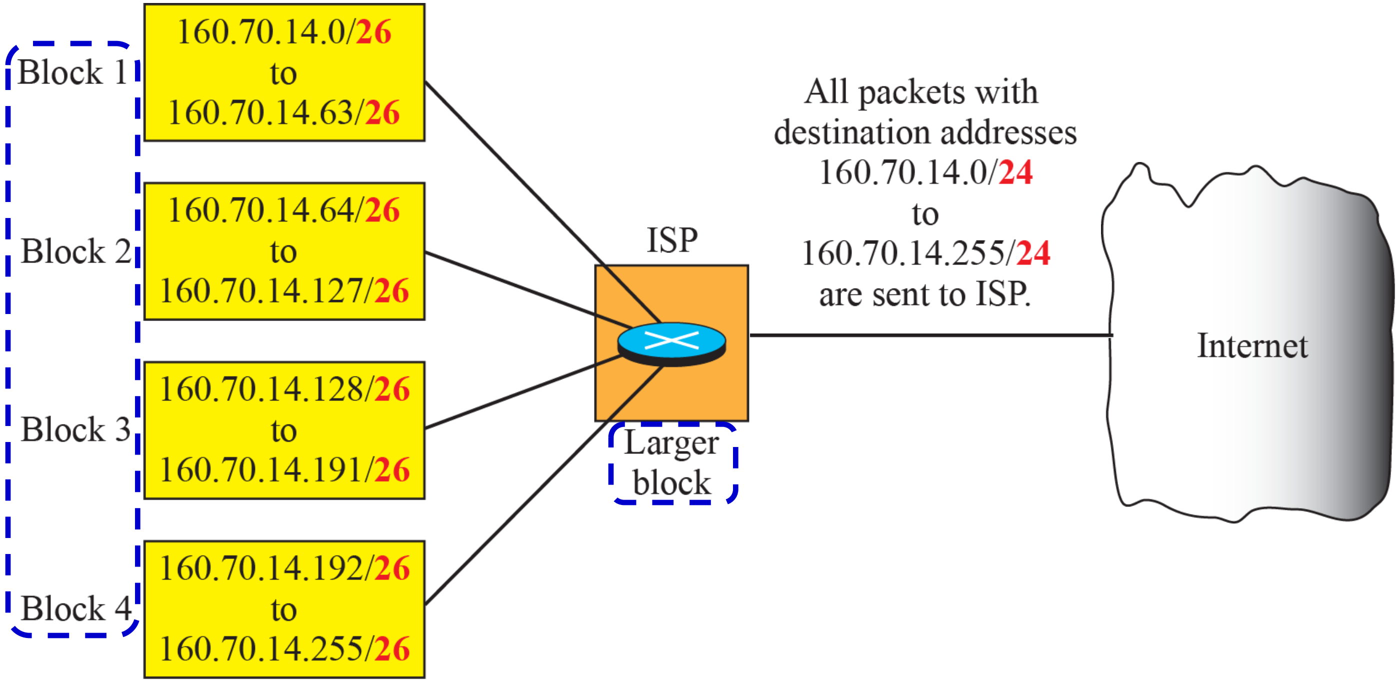
b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128/**26**; the last address is 14.24.74.191/**26**.

c. The number of addresses in the smallest subblock, which requires 10 addresses, is not a power of 2 either. We allocate 16 addresses. The subnet mask for this subnet can be found as $n_1 = 32 - \log_2 16 = 28$. The first address in this block is 14.24.74.192/**28**; the last address is 14.24.74.207/**28**.



Example

The following diagram shows how four small blocks of addresses are assigned to four organizations by an internet service provider (ISP). The ISP combines these four blocks into one single block (**supernet**). Any packet destined for this larger block should be sent to this ISP. It is the responsibility of the ISP to forward the packet to the appropriate organization.



Chapter 19: Network Layer Protocols

Outline

19.1 IPv4

19.1 NETWORK-LAYER PROTOCOLS

The main protocol in the network layer, Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet. It is an unreliable and a connectionless datagram protocol.

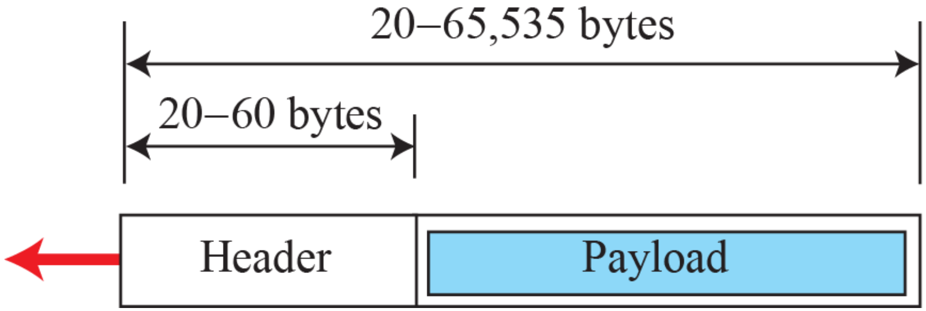
The Internet Contol Message Protocol version 4 (ICMPv4), a network layer protocol, is a companion to IPv4 and helps IPv4 to handle some errors that may occur in delivery.

19.1.1 Datagram Format

Packets used by the IP are called <u>datagrams</u>. A datagram is a <u>variable-length</u> packet consisting of two parts: <u>header</u> and <u>payload</u> (data). The <u>header</u> is a <u>minimum of 20 bytes</u> and <u>up to 60 bytes</u> in length and contains information essential to <u>routing</u> and <u>delivery</u>.

It is customary in TCP/IP to show the IP header in <u>32-bit</u> sections.

Figure 19.2: ***IP datagram***



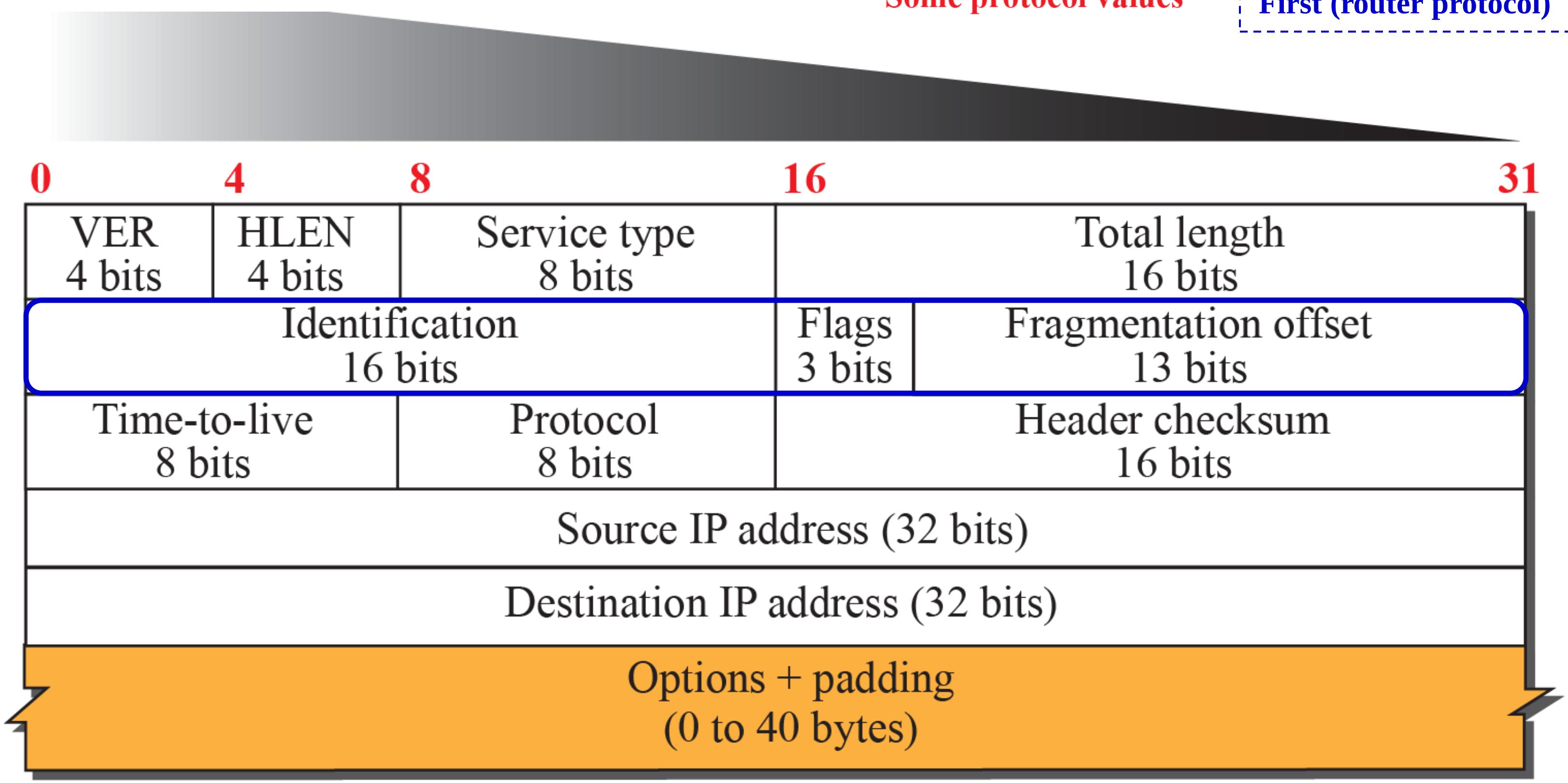
Legend

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

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

Some protocol values

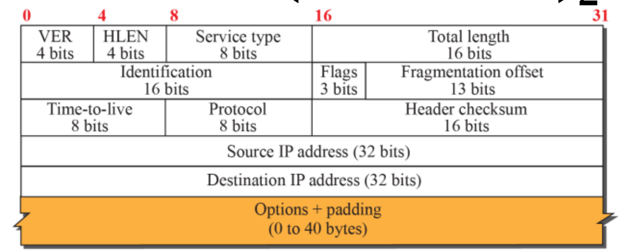
IGMP: Internet Group Management Protocol (IP multicast)
OSPF: Open Shortest Path First (router protocol)



Notes: 1)* HLEN *is in units of* <u>*4-bytes*</u>*. 2)* Total length *includes both the* <u>*header*</u> *and* <u>*payload*</u> *in bytes.

Example

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



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 of 8 bytes (2×4). The minimum number of bytes in the header must be 20 bytes.

Q) 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 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

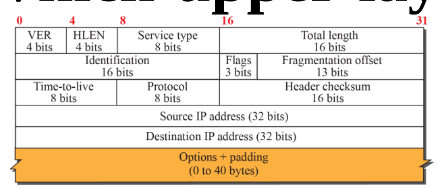
Problems

Q) 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 20 bytes, i.e., no options. The total length is $(0028)_{16}$ or 40 bytes, which means the packet is carrying $40 - 20 =$ <u>20 bytes</u> of data.

Q) An IPv4 packet has arrived with the first few hexadecimal digits as $(4500\ 0028\ 0001\ 0000\ 0106)_{16}$. How many hops can this packet travel before being dropped? Which upper-layer protocol does the data belong to?



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

Some protocol values

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 <u>one hop</u>. The protocol field is the next byte $(06)_{16}$, which means that the upper-layer protocol is <u>TCP</u>.

Example

An example of a checksum calculation for an IPv4 header without options is shown:

16 bits		16 bits	
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
49153 →		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**

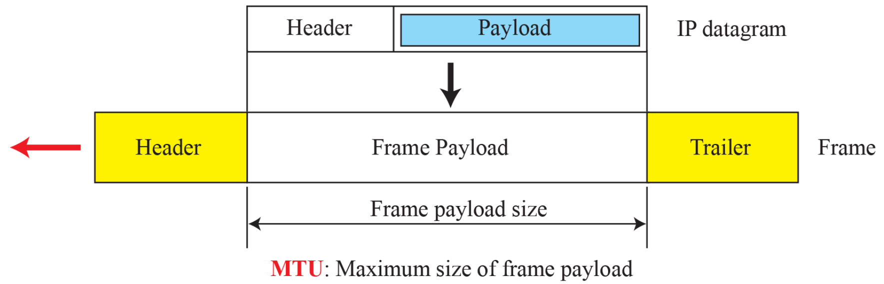
The header is divided into <u>16-bit sections</u>. All the sections are added and the sum is complemented after wrapping the leftmost digit. The result is inserted in the checksum field.

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

Wrapped Sum = Sum mod $FFFF_{16}$
Checksum = $FFFF_{16}$ – Wrapped Sum

19.1.2 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__.



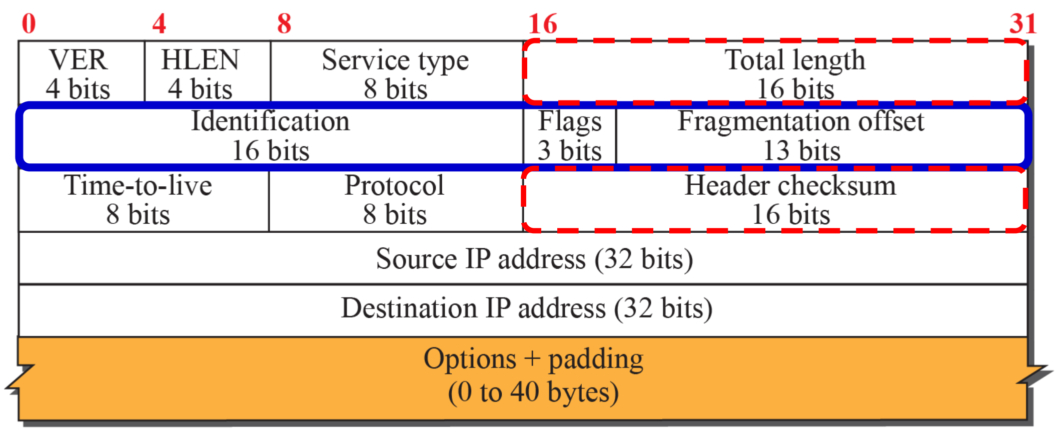
Each link-layer protocol has its own frame format. The value of the __maximum transfer unit__ (MTU) differs from one physical network protocol to another. The datagram has to be divided, i.e., __fragmentation__, to make it possible to pass through these networks. The __reassembly__ of the datagram is done only by the __destination host__.

The __format and size__ of the

(i) received frame: depend on the protocol used by the physical network through which the __frame has just traveled__.

(ii) sent frame: depend on the protocol used by the physical network through which the __frame is going to travel__.

Figure 19.6: *Fragmentation example*



Identification: The 16-bit *identification field* identifies a datagram originating from the source host (Uniqueness: Source IP addr + Identification).

Flags: The 3-bit *flags field* identifies 3 flags:

- **The leftmost bit is reserved (not used).**
- **The second bit (D bit) is the *do not fragment* bit.**
- **The third bit (M bit) is the *more fragment* bit.**

Fragmentation offset: The 13-bit *fragmentation offset field* shows the relative position of this fragment with respect to the original payload and is measured in units of 8-bytes.

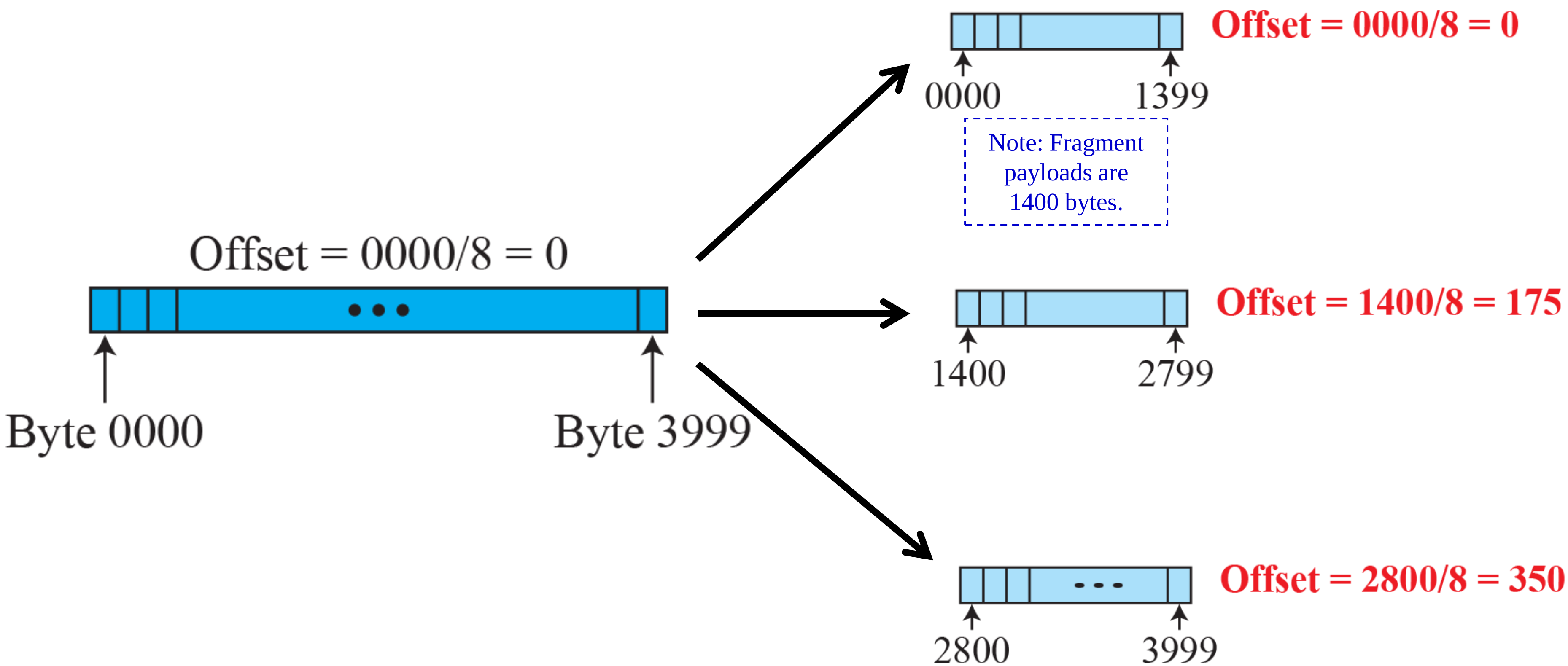
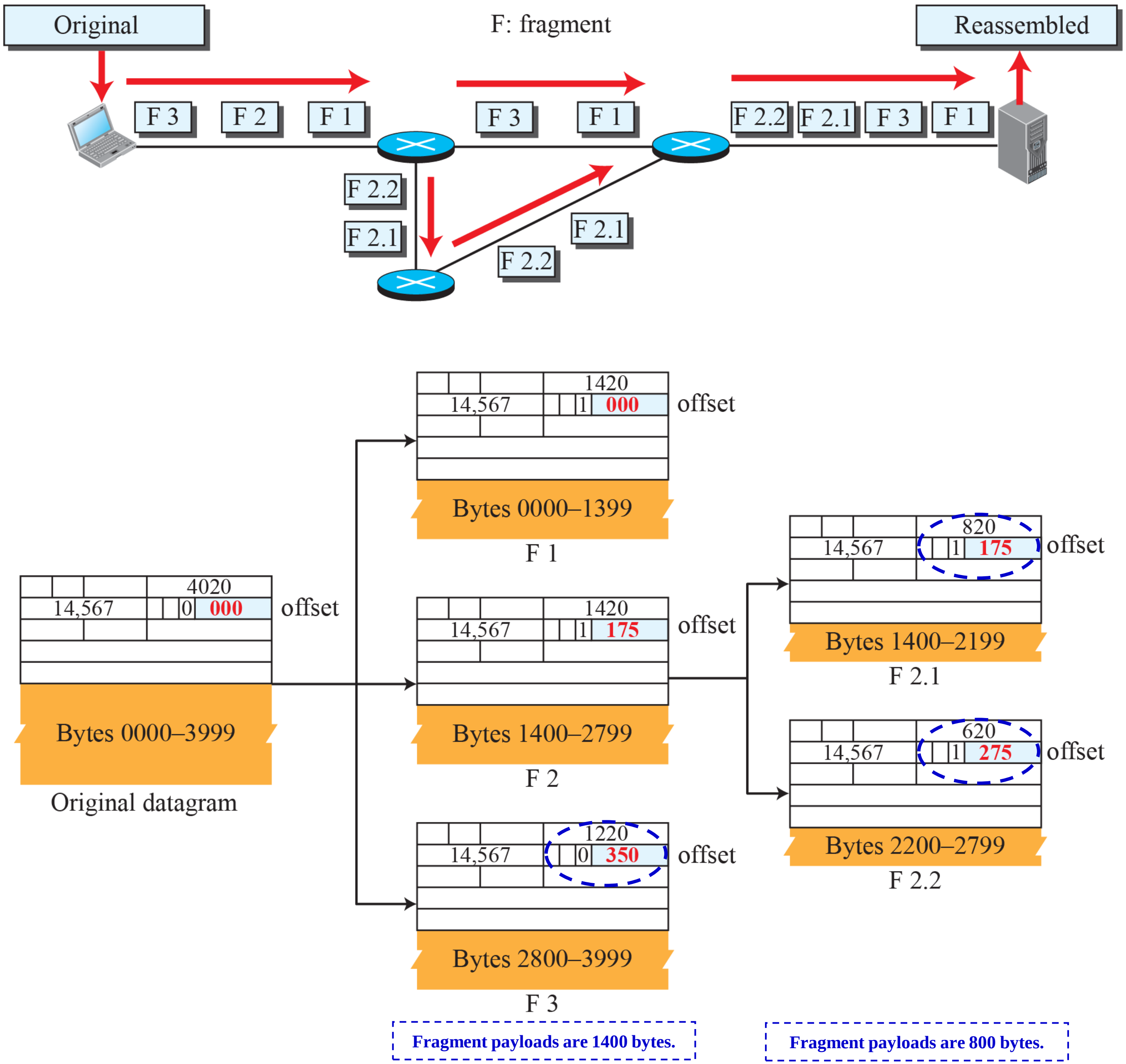


Figure 19.7: Detailed fragmentation example



Example

Q) A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment or a middle fragment? Was the packet fragmented?

Solution

If the M bit is 0, it means that there are no more fragments, i.e., the fragment is the last one. However, there is insufficient information to determine if the original packet was fragmented or not. Note that a non-fragmented packet is considered the last fragment.

Q) A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment or a middle fragment? Was the packet 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. However, there is insufficient information (require the value of the fragmentation offset) to determine if it is the first one or a middle one.

Problems

Q) 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. Since the offset value is 0, it is the first fragment.

Q) A packet has arrived in which the offset value is 100. What is the number of the first byte? What is 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. The number of the last byte cannot be determined unless we know the length of the data.

19.1.3 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. The variable part comprises the <u>options</u> and can be a maximum of <u>40 bytes</u> (in multiples of 4-bytes).

Options are <u>not required</u> for a datagram but can be used for <u>network testing</u> and <u>debugging</u>. Note that although options are not a required part of the IPv4 header, all <u>implementations</u> of IPv4 software must be <u>able to handle options</u>.

The complete discussion of options in IPv4 is included in the book website under <u>Extra Materials</u> for Chapter 19.