Chapter 18: Introduction to Network Layer

Outline

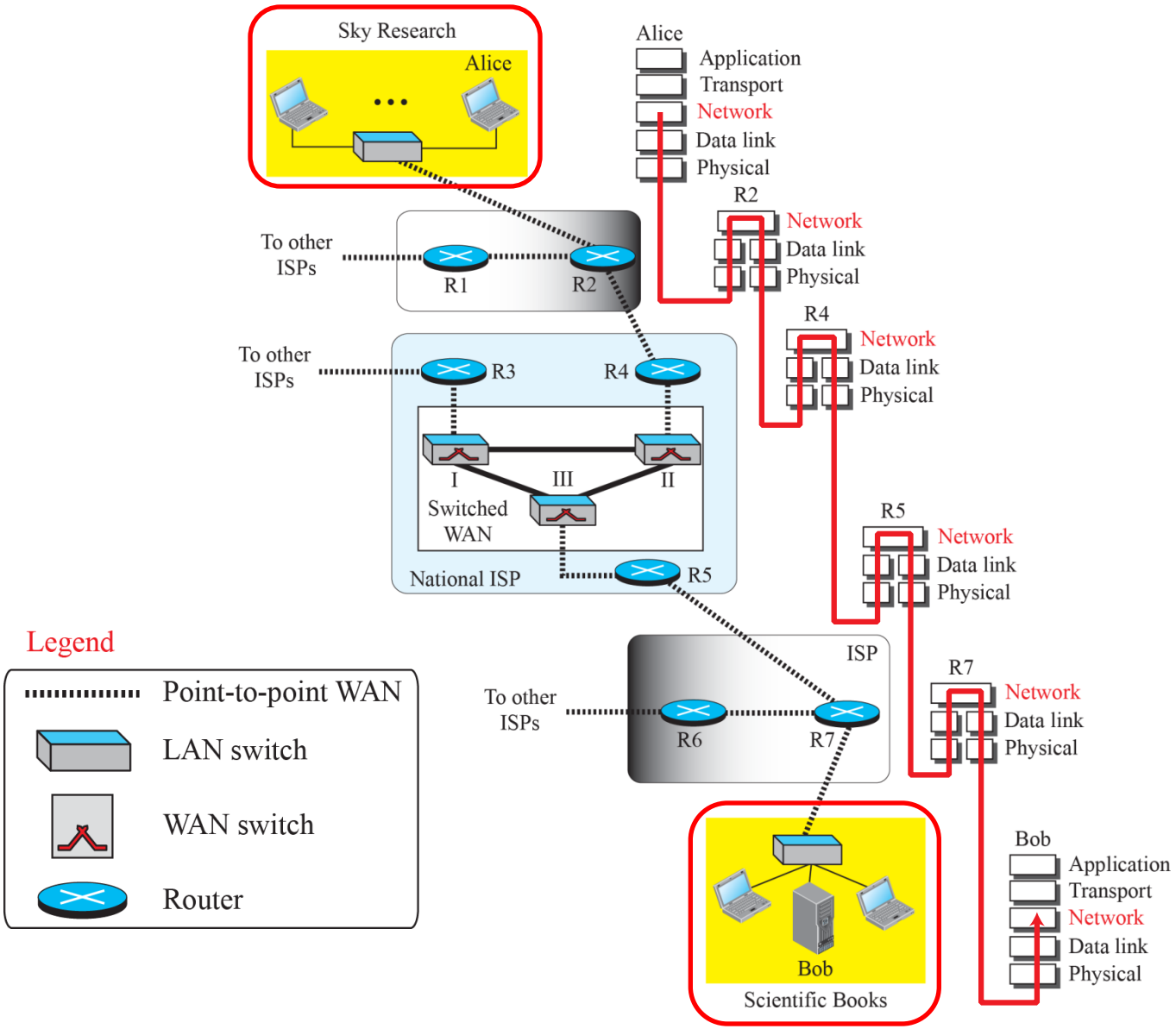
18.1 NETWORK-LAYER SERVICES

18.2 PACKET SWITCHING

18.4 IPv4 ADDRESSES

18.3 NETWORK-LAYER PERFORMANCE

Figure 18.1: Communication at the network layer



18.1 Network-Layer Services

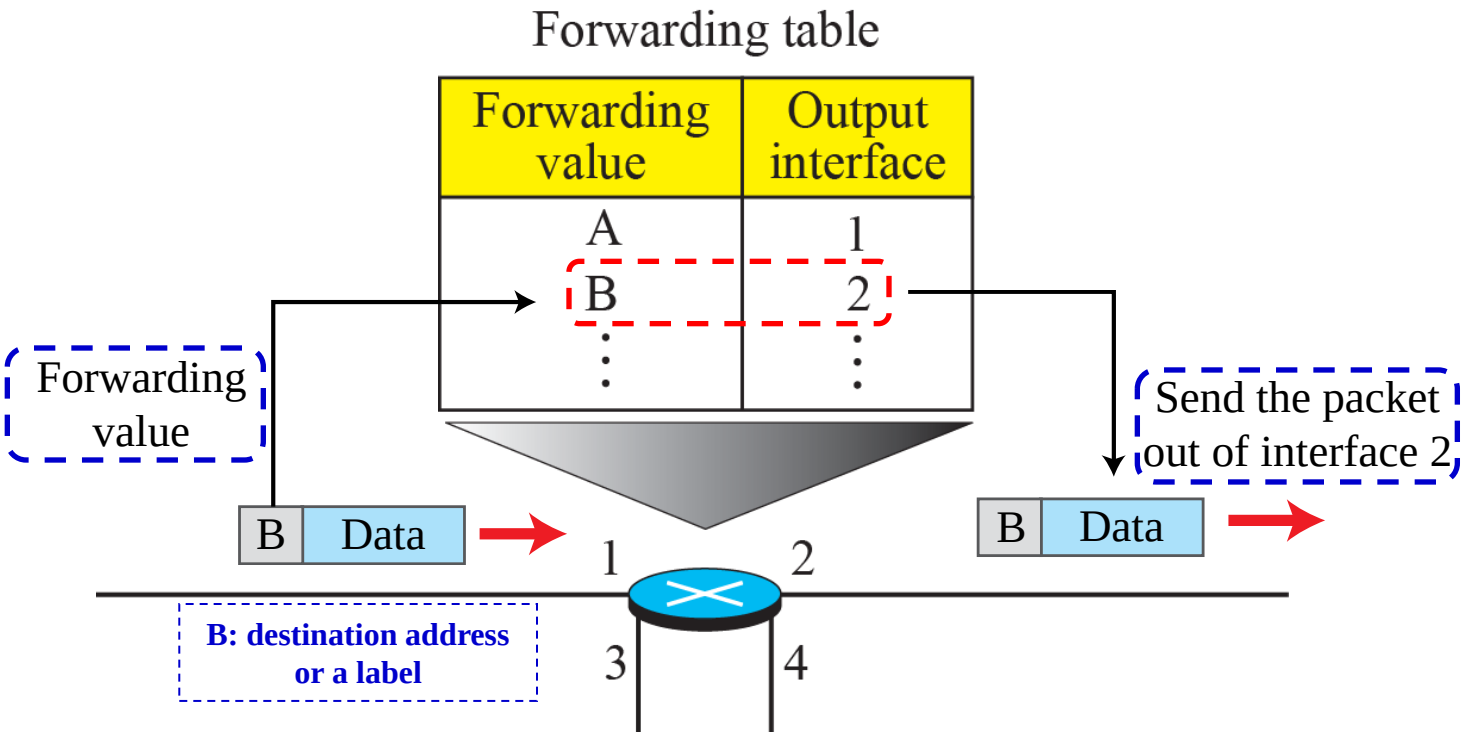
Before discussing the network layer in the Internet today, let's briefly discuss the network-layer services (packetizing, routing, forwarding) that, in general, are expected from a network-layer protocol. In addition, other services (error control, flow control, congestion control, quality of service 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 action applied by each router when a packet arrives at one of its interfaces, i.e., to forward the packet to another (unicast) or some (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.

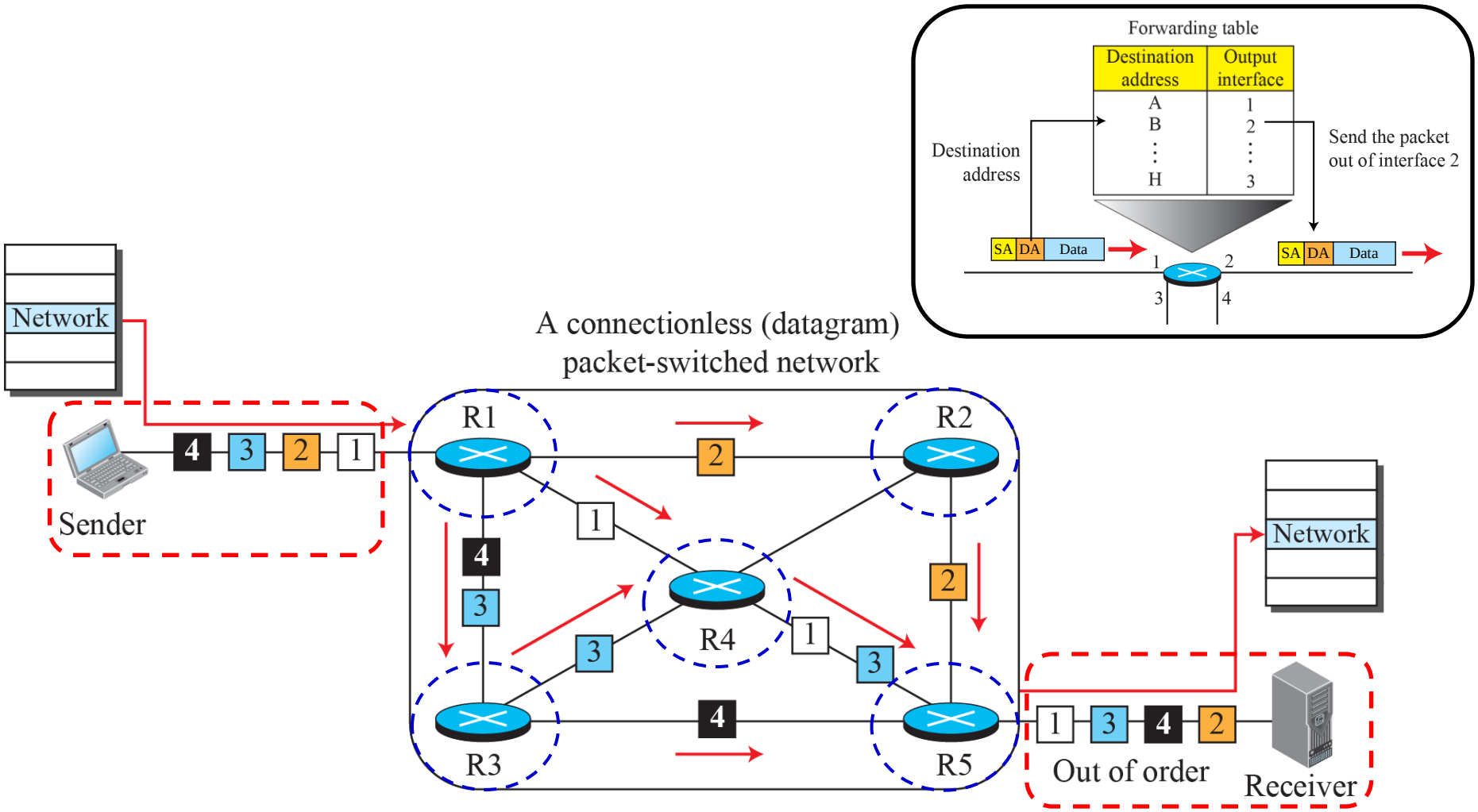
A router, in fact, is a switch that creates a connection between an input port and an output port (or a set of output ports), just as an electrical switch connects the input to the output to let electricity flow.

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

The IP address is the address of the connection, not the host or the router.

18.4.1 Address Space

A protocol like IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. IPv4 uses 32-bit addresses, 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.

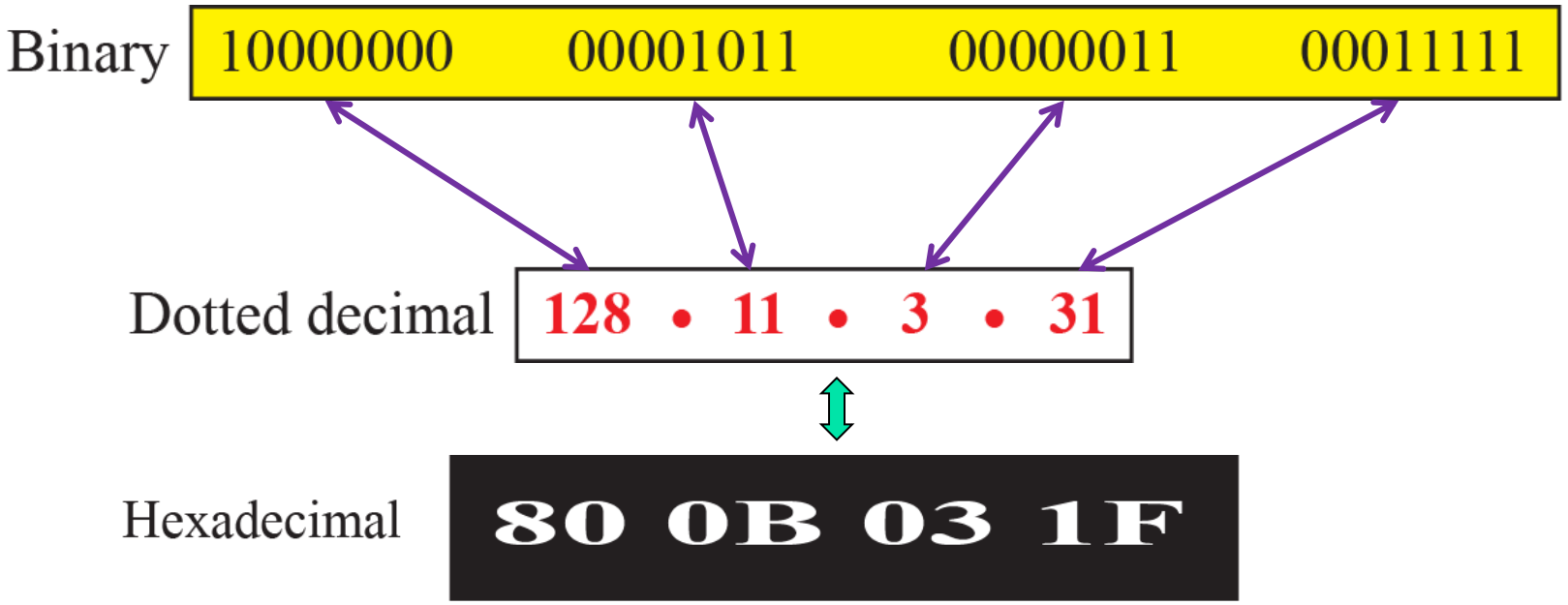
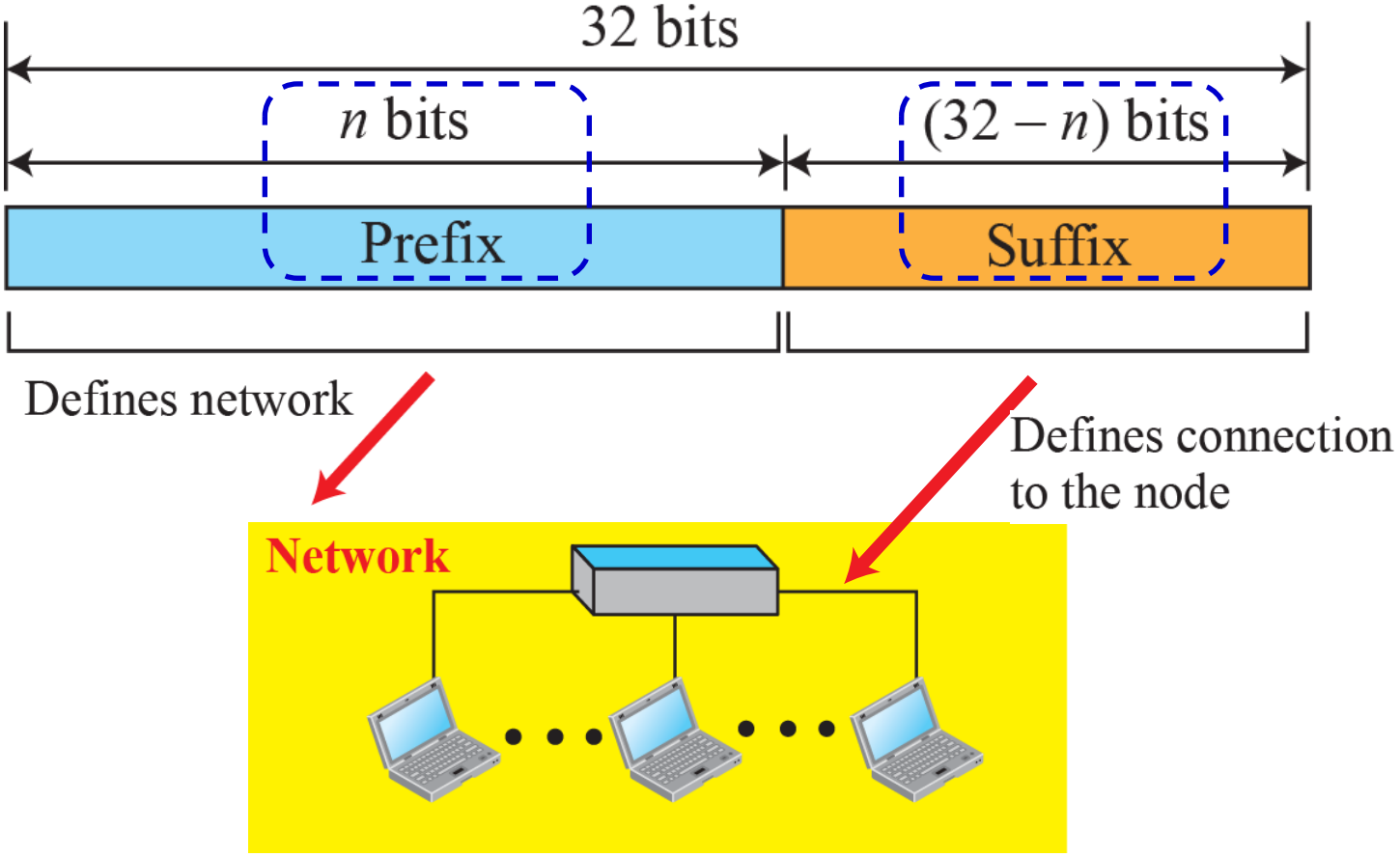


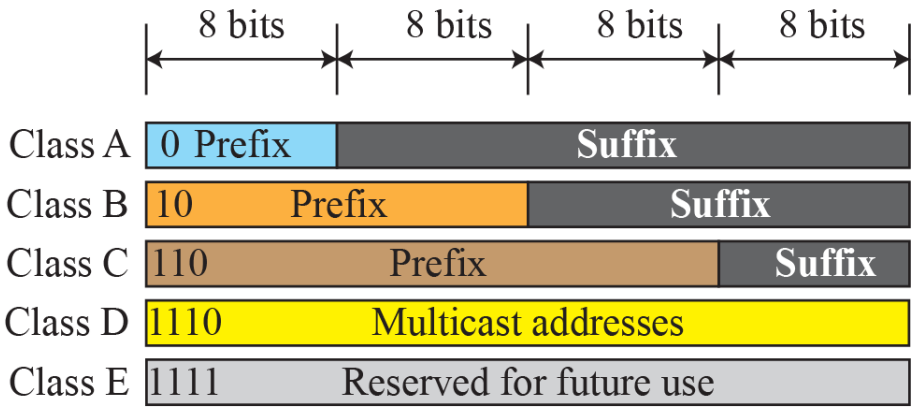
Figure 18.17: Hierarchy in addressing

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.

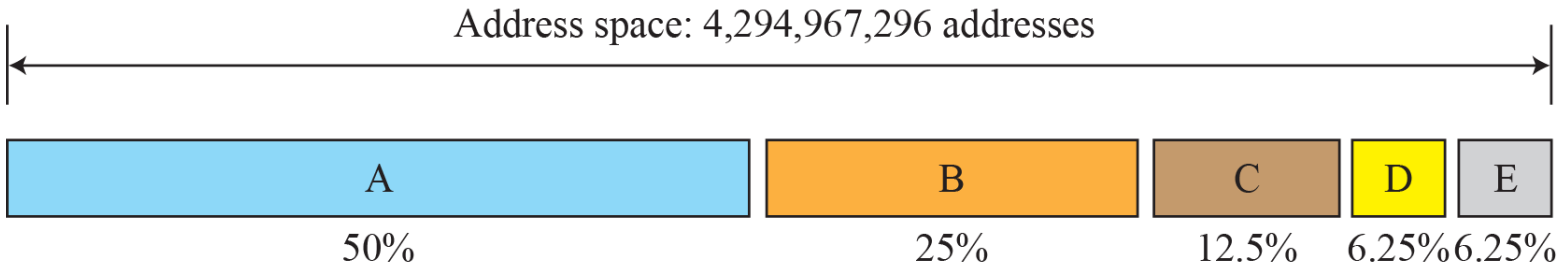


18.4.2 Classful Addressing

When the Internet started, an IPv4 address was designed with a fixed-length prefix: 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 classful addressing but is obsolete.



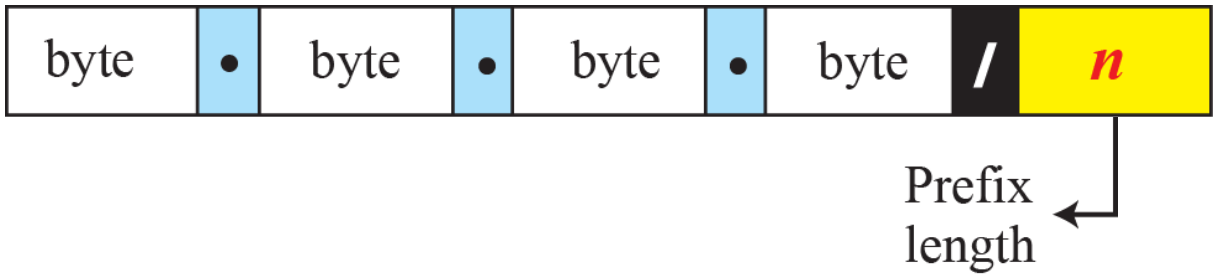
Class	Prefixes	First byte
A	*n* = 8 bits	0 to 127
B	*n* = 16 bits	128 to 191
C	*n* = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255



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 $2^{128} = 340 \times 10^{36}$), 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.

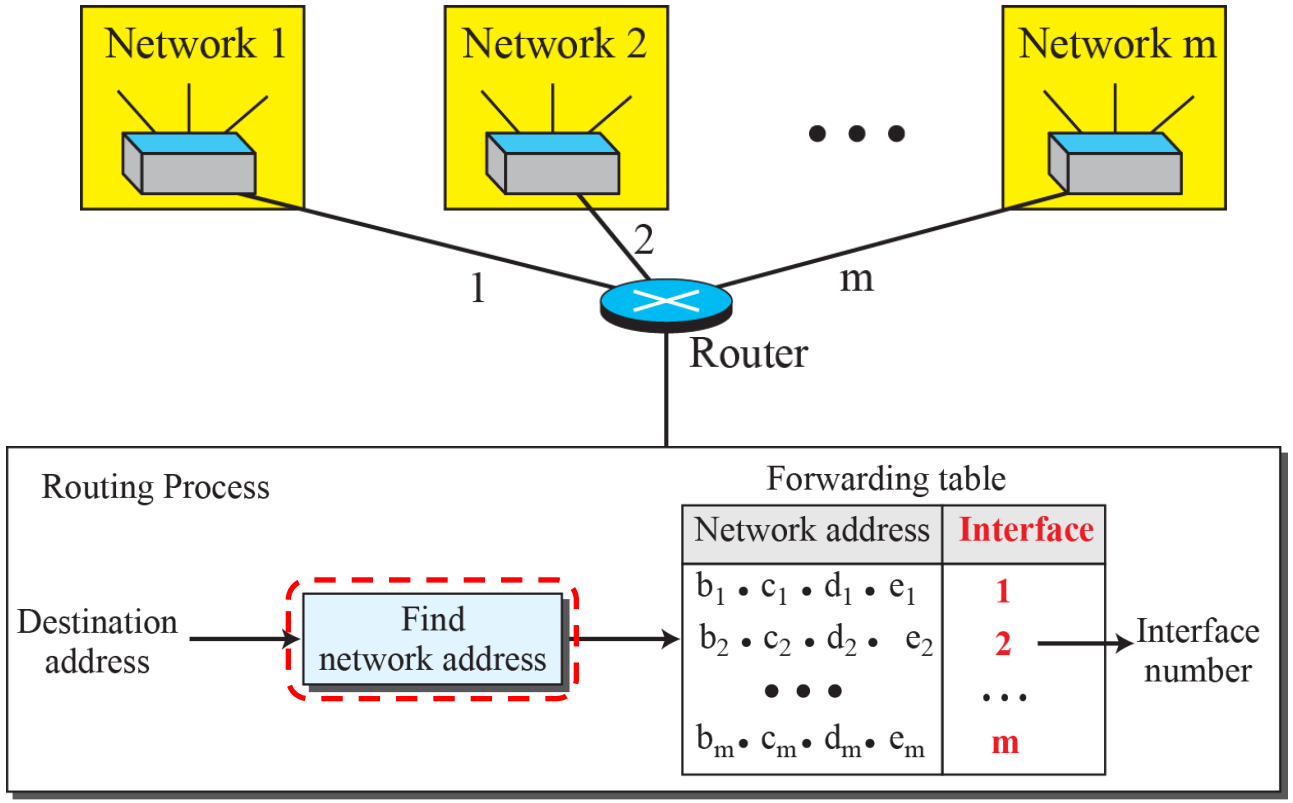
Address: 167.199.170.82/**27**	10100111	11000111	10101010	01010010
First address: 167.199.170.64/**27**	10100111	11000111	10101010	01000000

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

Address: 167.199.170.82/**27**	10100111	11000111	10101010	01011111
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.



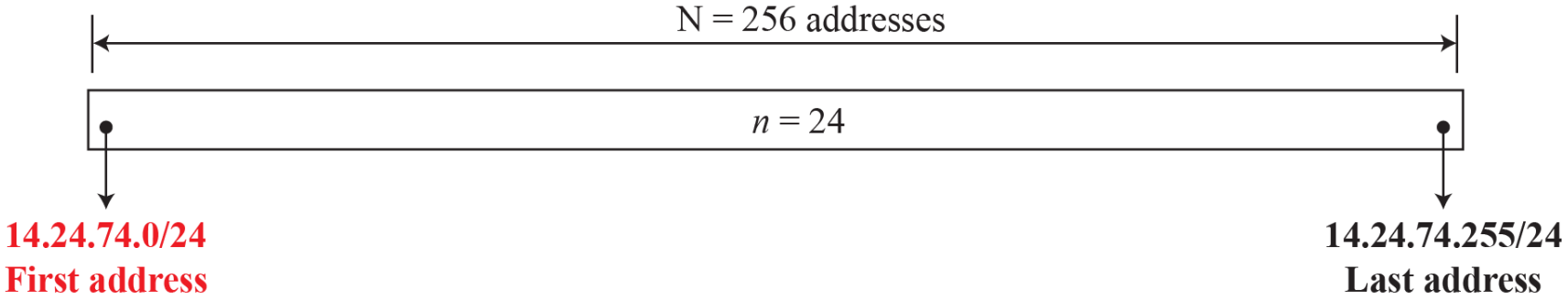
Additional Information: Chapter 18.5 (Forwarding of IP Packets, Longest Mask Matching):
To determine the network address (and hence the corresponding interface to send the packet out), 1) mask the prefix length with the destination address using the logical AND operation and 2) select the interface with 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.

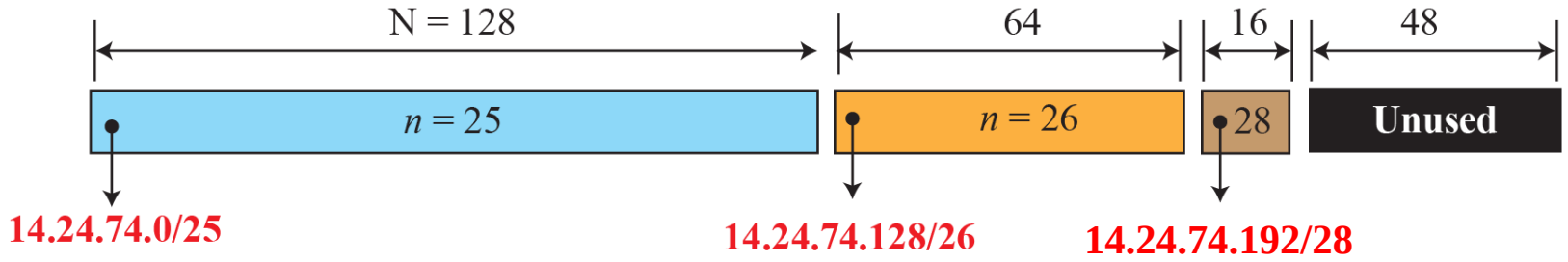


Problem (cont'd)

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**.

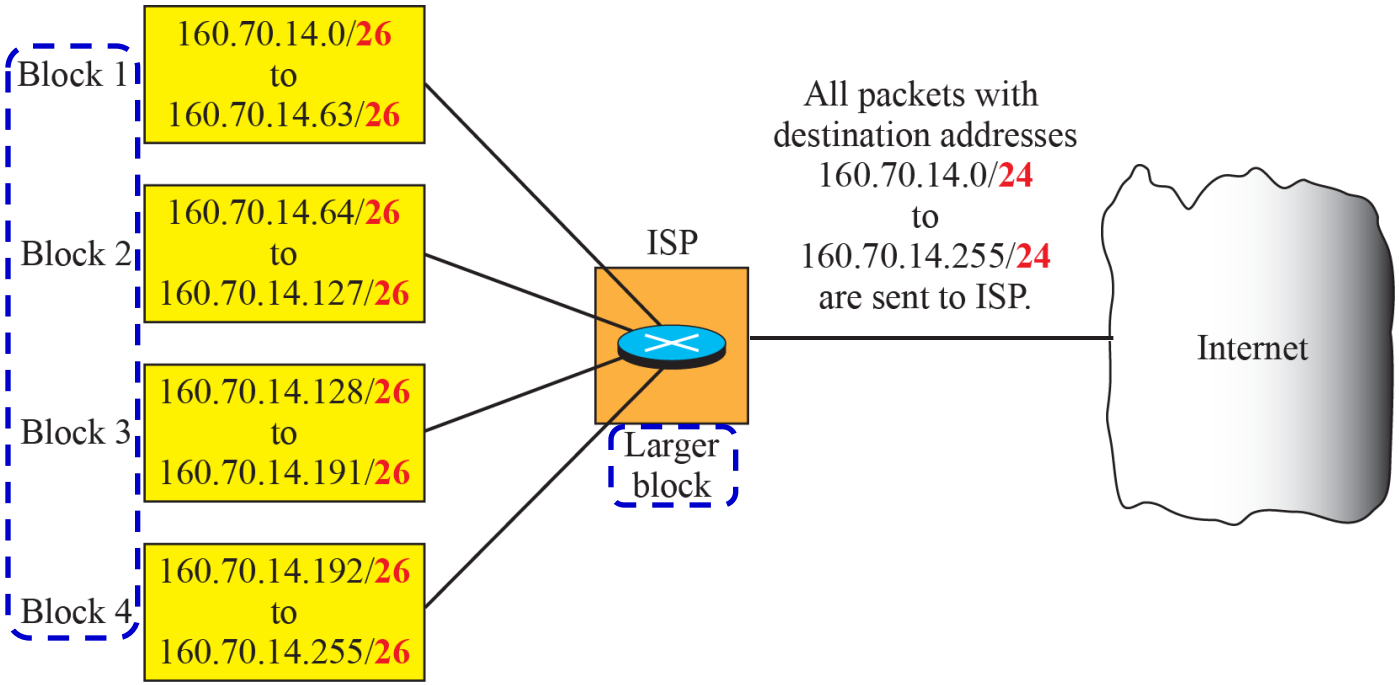
b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128/**26**; the last address is 14.24.74.191/**26**.

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.



18-3 NETWORK-LAYER PERFORMANCE

The upper-layer protocols that use the service of the network layer expect to receive an ideal service, but the network layer is not perfect. The performance of a network can be measured in terms of delay, throughput and packet loss.

Congestion control is an issue that can improve the performance.

18.3.1 Delay

All of us expect instantaneous response from a network, but a packet, from its source to its destination, encounters delays. Recall that the delays in a network can be divided into four types:

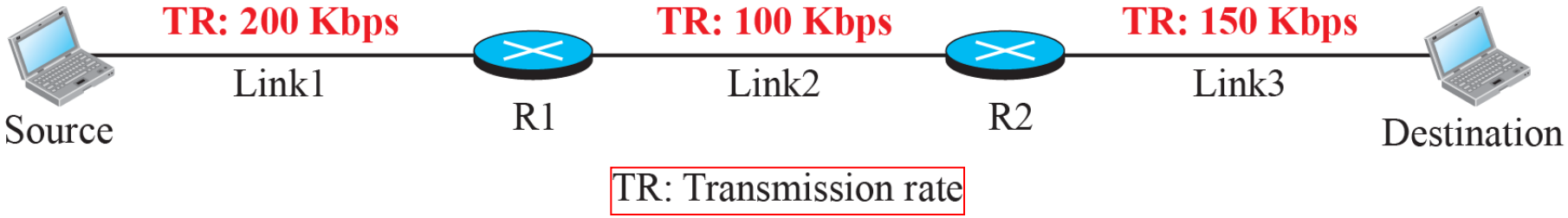
- D_{tr} (transmission): (Packet length) / (Transmission rate)
- D_{pg} (propagation): (Distance) / (Propagation speed)
- D_{pr} (processing): Time required to process a packet in a router or destination host
- D_{qu} (queuing): Time a packet waits in input and output queues in a router

Note that in a network with n routers, there are (n + 1) links connecting the routers. Assuming equal delays for the sender, routers and receiver, the total delay a packet encounters from source to destination with n routers is

$$\text{Total Delay} = (n+1)(D_{tr} + D_{pg} + D_{pr}) + (n)(D_{qu})$$

18.3.2 Throughput

Throughput at any point in a network is defined as the number of bits passing through the point in a second, which is actually the transmission rate of data at that point. In a path from source to destination, a packet may pass through several links, each with a different transmission rate.



$$\text{Throughput} = \text{minimum}\,(TR_1, TR_2, ..., TR_n\})$$

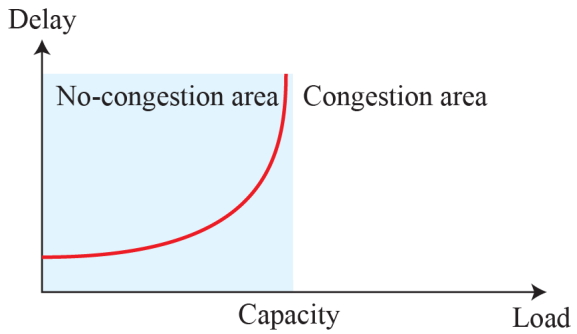
18.3.3 Packet Loss

Another issue that severely affects the performance of communication is the number of packets lost during transmission. When a router receives a packet while processing another packet, the received packet needs to be stored in the input buffer waiting for its turn. A router, however, has an input buffer with a limited size. A time may come when the buffer is full and the next packet needs to be dropped.

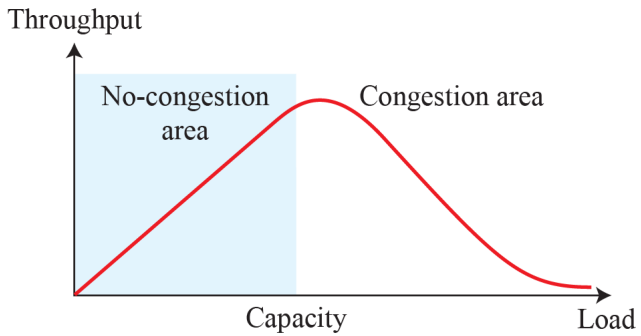
The effect of packet loss is that the packet needs to be resent, which in turn may create queue overflow and cause more packet loss.

18.3.4 Congestion Control

Congestion at the network layer is related to two issues: delay and throughput.



a. Delay as a function of load



b. Throughput as a function of load

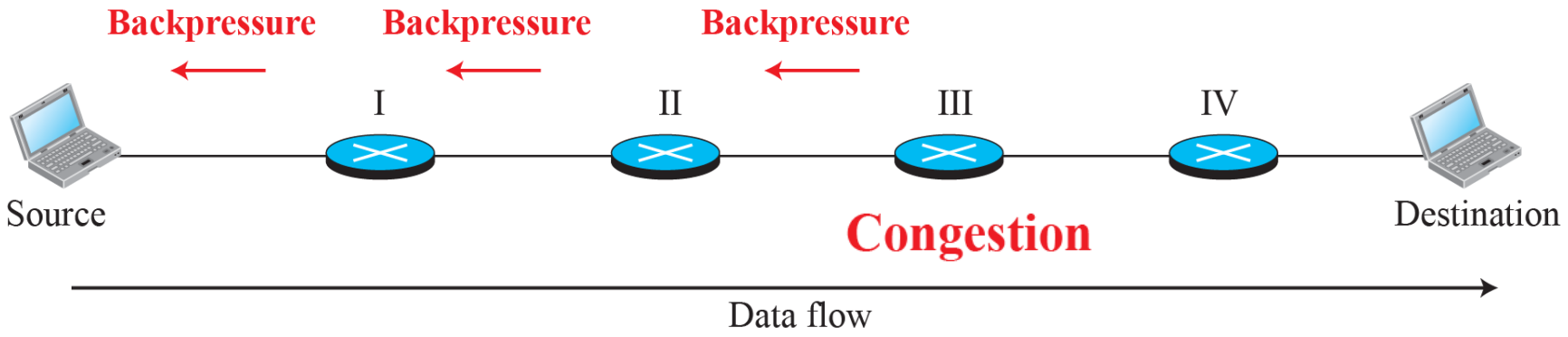
When the load is much less than the capacity of the network, the delay is at a minimum. When the load increases and reaches the network capacity, the delay increases sharply because we need to consider *queuing delay* (in addition to the processing delay and propagation delay). Note that the delay becomes infinite when the load is greater than the capacity.

When the load is below the capacity of the network, the throughput increases proportionally with the load. We would expect that the throughput to remain constant after the load reaches the capacity, but instead the throughput declines sharply. The reason is the *discarding of packets by the routers*. Note that discarding packets does not reduce the number of packets in the network as sources retransmit packets when packets do not reach the destinations.

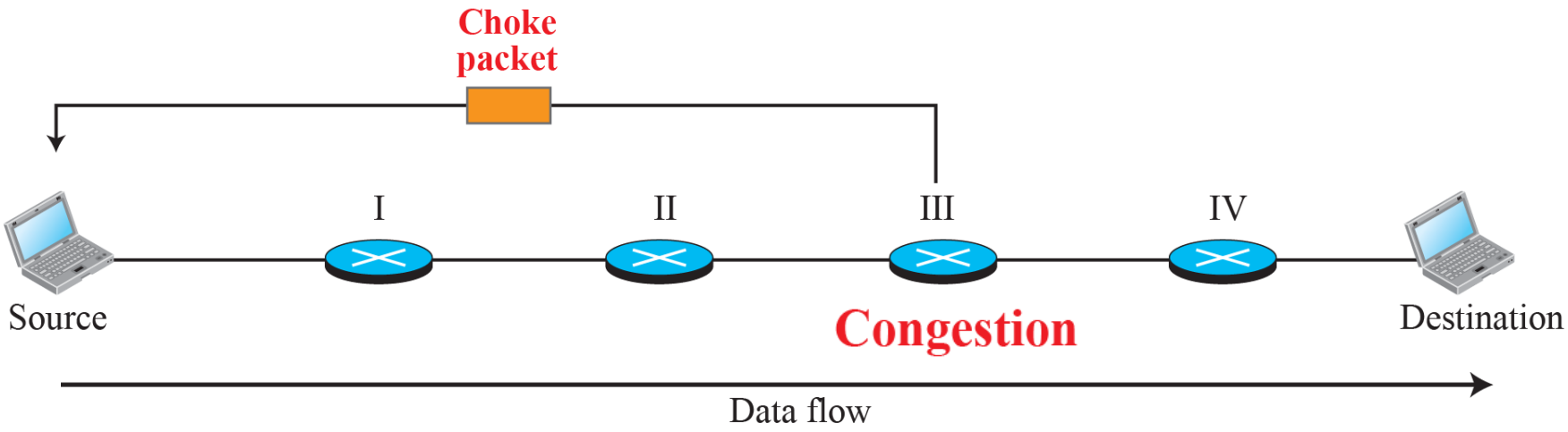
Congestion control is a mechanism for improving performance and refers to techniques and mechanisms that can either prevent congestion before it happens (open-loop congestion control: retransmission, window, acknowledgement, discarding and admission policies) or remove congestion after it has happened (closed-loop congestion control: backpressure, choke packet, implicit/explicit signaling).

Figure 18.14: Mechanisms for alleviating (removing) congestion

Backpressure: a node-to-node congestion control that starts with a congested node and propagates, in opposite directions of data flow, to the source. Note that the backpressure technique can only be applied to virtual circuit networks where each node knows the upstream node from which a flow of data is coming.



Choke packet: in this mechanism, a choke packet (quench Internet Control Message Protocol (ICMP) message) is sent from the router which has encountered congestion, directly to the source station.



Chapter 19: Network Layer Protocols

Outline

19.1 IPv4

19.2 ICMPv4

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.

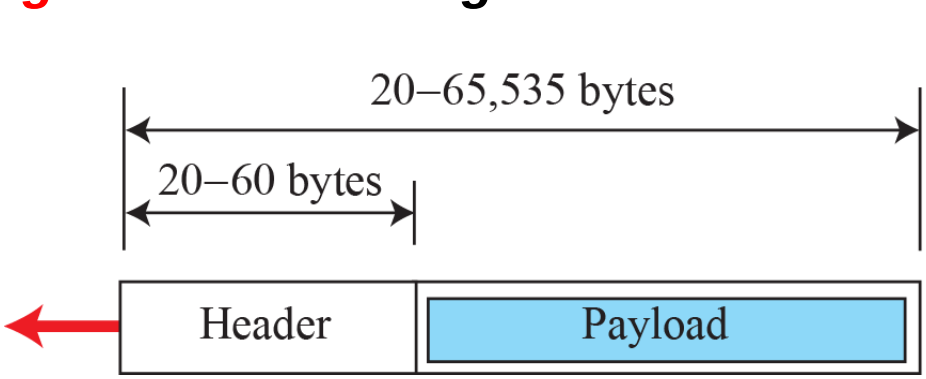
The Internet Contol Message Protocol version 4 (ICMPv4), a network layer protocol, is a companion to IPv4 and helps IPv4 to handle some errors that may occur in delivery.

19.1.1 Datagram Format

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

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

Figure 19.2: IP datagram



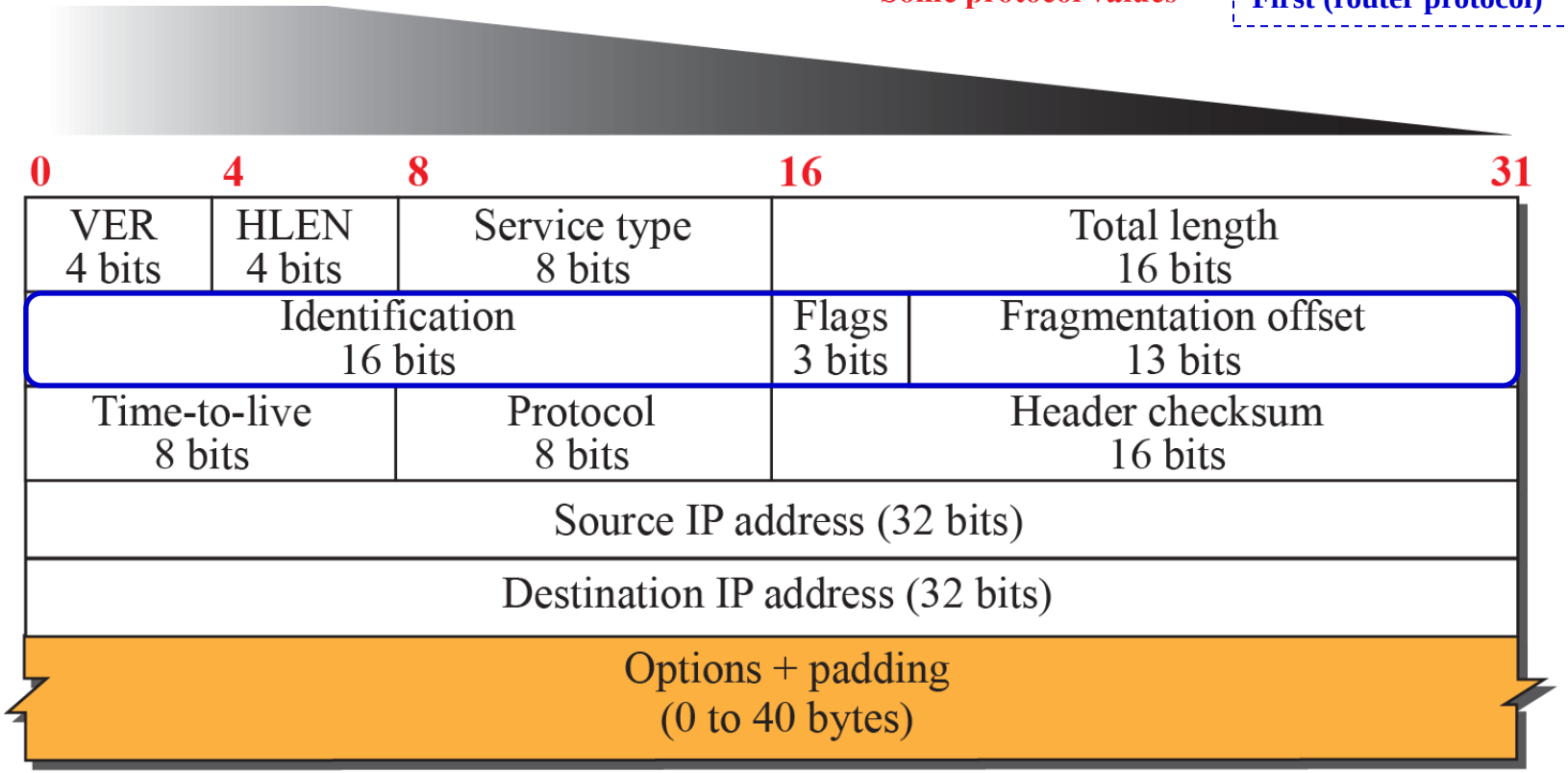
Legend

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

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

Some protocol values

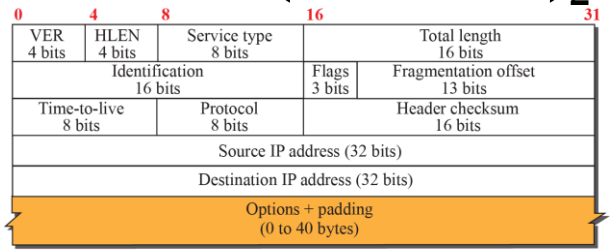
IGMP: Internet Group Management Protocol (IP multicast)
OSPF: Open Shortest Path First (router protocol)



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

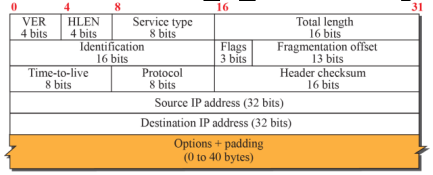
Problems

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 $40 - 20 =$ 20 bytes 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
IGMP: 02
TCP: 06
UDP: 17
OSPF: 89

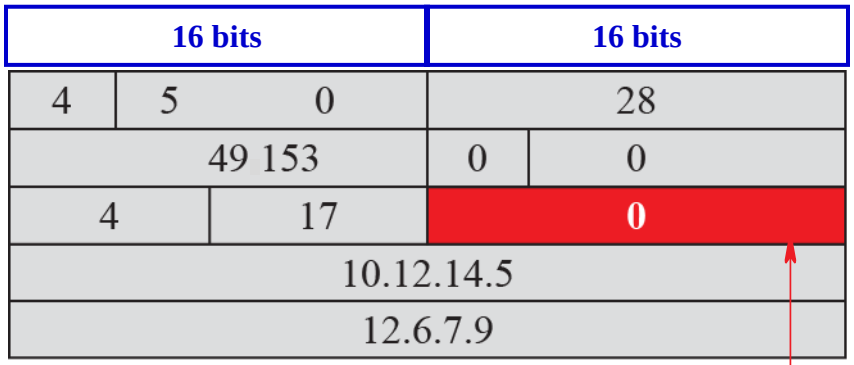
Some protocol values

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

Example

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



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**

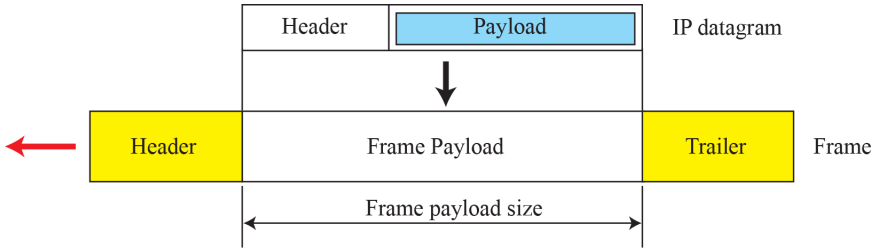
The header is divided into 16-bit sections. All the sections are added and the sum is complemented after wrapping the leftmost digit. The result is inserted in the checksum field.

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

Wrapped Sum = Sum mod $FFFF_{16}$
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.

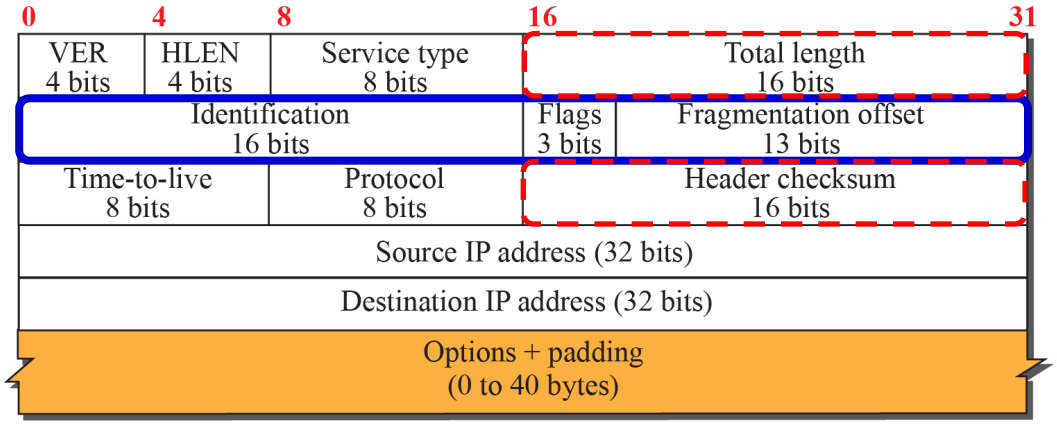


MTU: Maximum size of frame payload

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 whole datagram and is measured in units of 8-bytes.

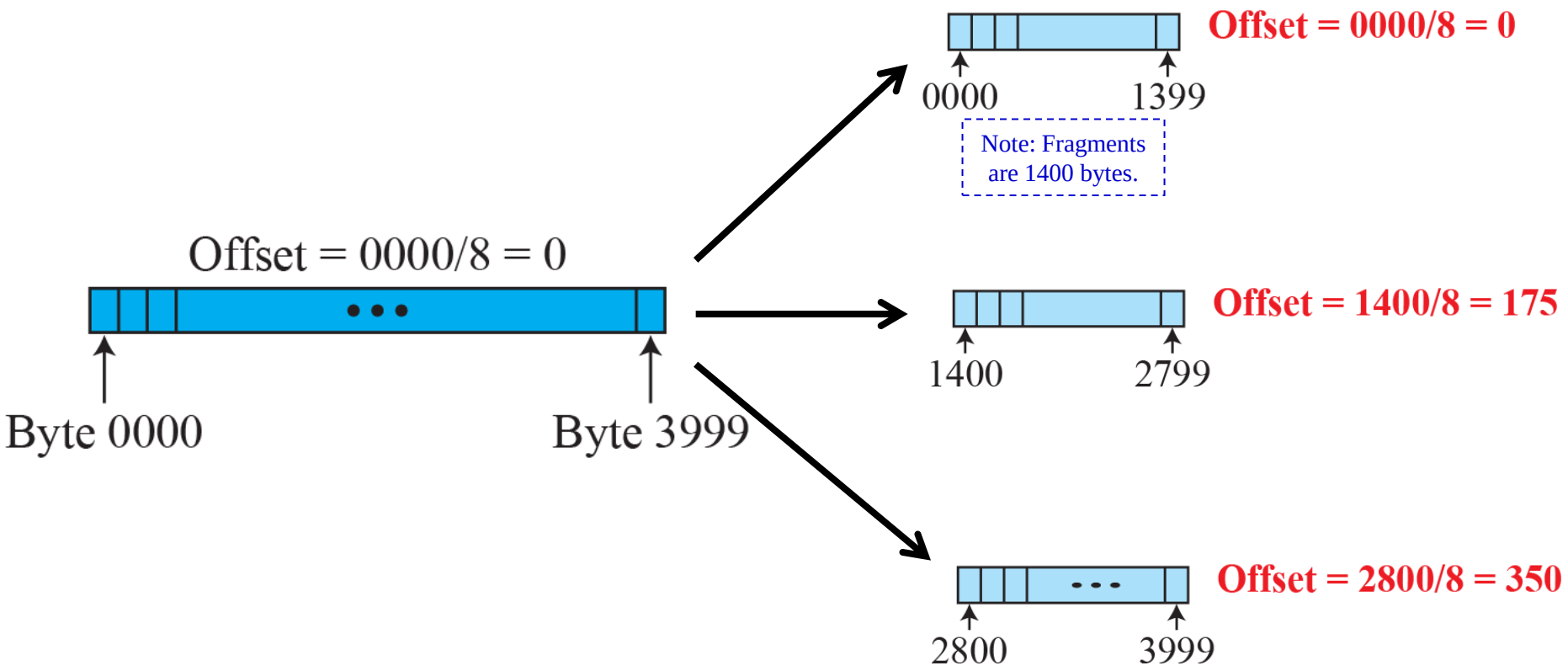
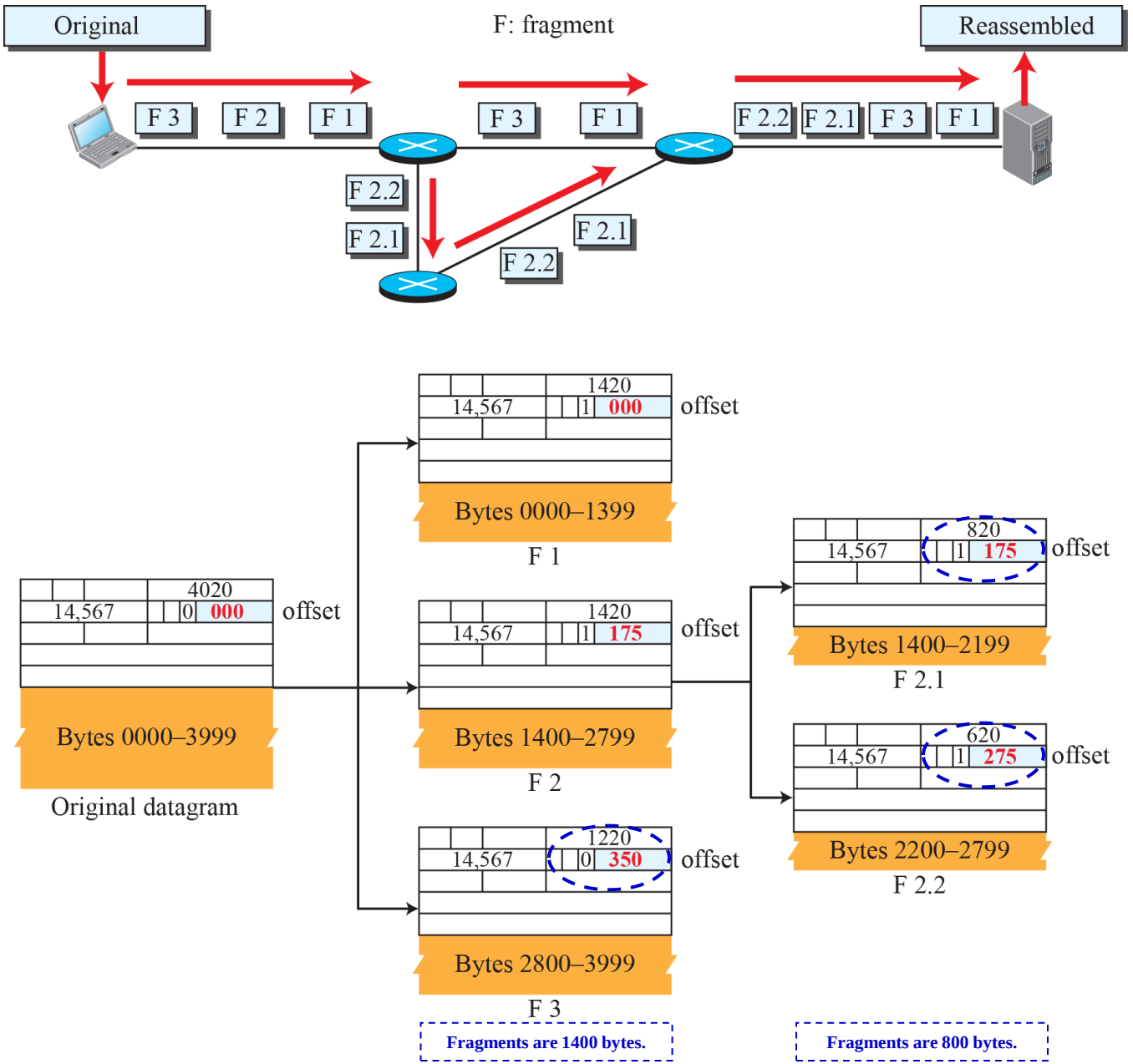


Figure 19.7: Detailed fragmentation example



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 options and can be a maximum of 40 bytes (in multiples of 4-bytes).

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

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

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) is a companion to the IP protocol and has been designed to compensate for the above two deficiencies. ICMP messages are encapsulated inside IP datagrams with protocol value = 01 and are divided into two broad categories: error-reporting messages and query messages.

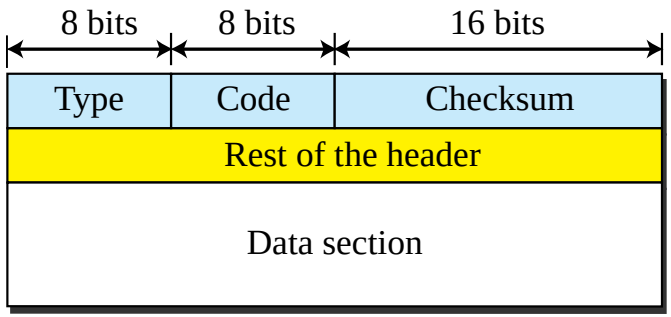
Error-reporting messages: report problems that a router or a host may encounter when it processes an IP packet (e.g., destination unreachable, source quench, etc.).

Query messages: occur in request/reply pairs, help a host or a network manager get specific information from a router or another host (e.g., ping, traceroute, etc.).

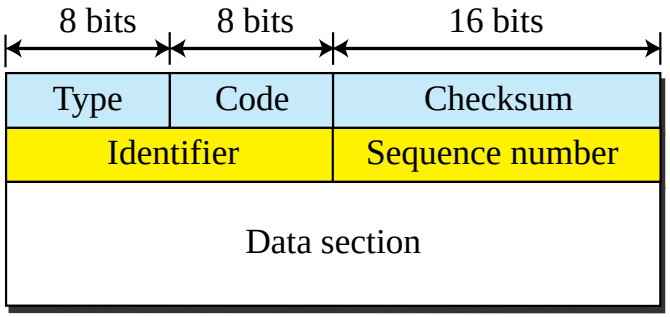
Figure 19.8: General format of ICMP messages

An ICMP message has an 8-byte header and a variable-size data section. The first byte, *type field*, defines the type of the message and the second byte, *code field*, specifies the reason for the particular message type. The last common field is the *checksum field*.

The rest of the header is specific for each message type. The data section in error messages carries information for finding the original packet that had the error. The data section in query messages carries extra information based on the type of query.



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

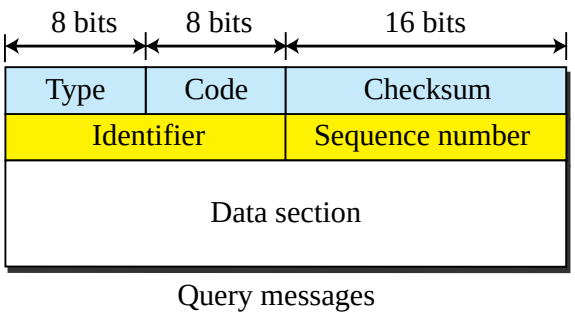
08 and 00: Echo request and reply (only code 0)
13 and 14: Timestamp request and reply (only code 0)

The complete discussion of messages in ICMPv4 is included in the book website under Extra Materials for Chapter 19.

19.2.3 ICMP Checksum

In ICMP the checksum is calculated over the <u>entire message</u> (header and data).

As an example, a query message with identifier = 1, sequence number = 9 and data section = TEST. The data is ASCII-encoded and the message is divided into 16-bit words. The words are added and the sum is complemented for inclusion in the Checksum field.



Query messages