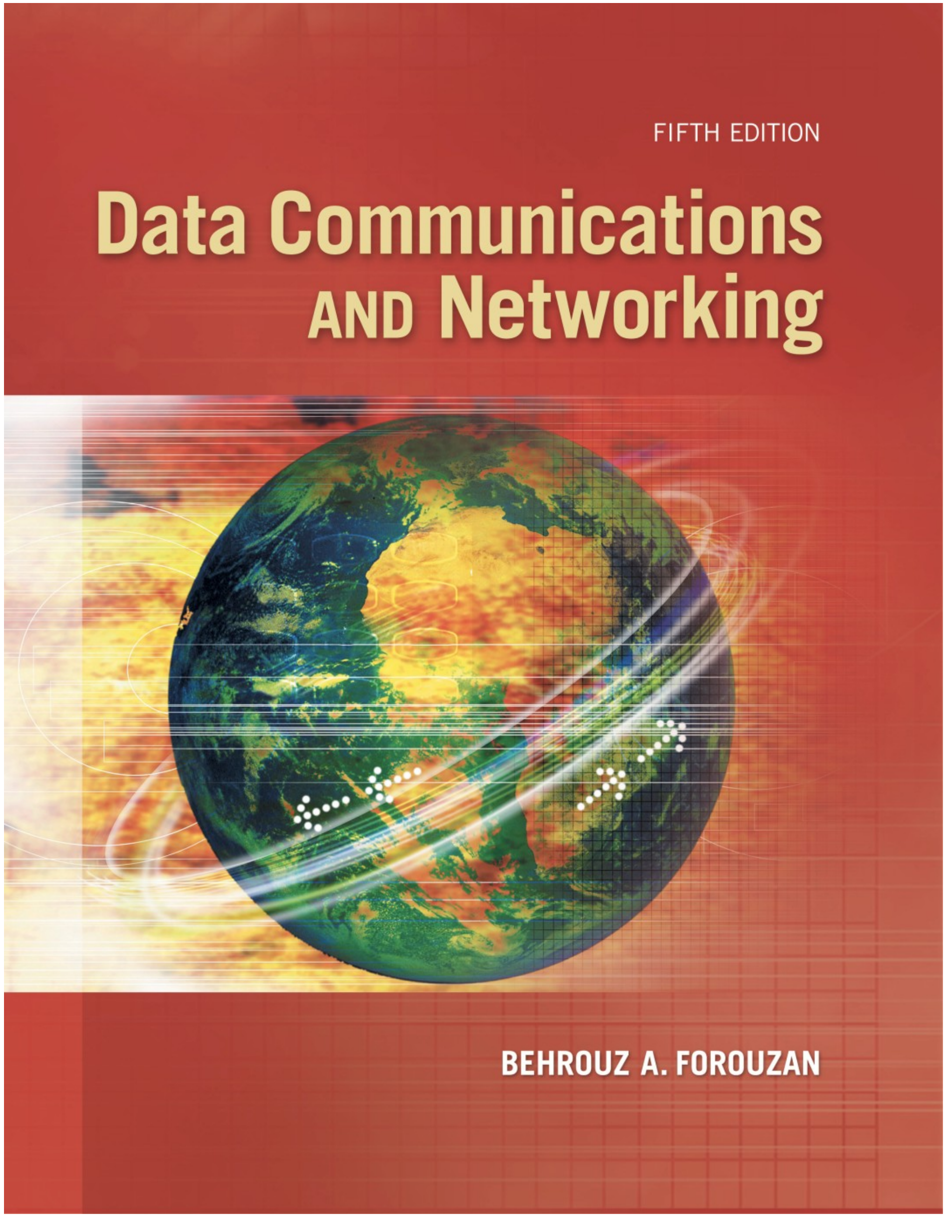
Chapter 18

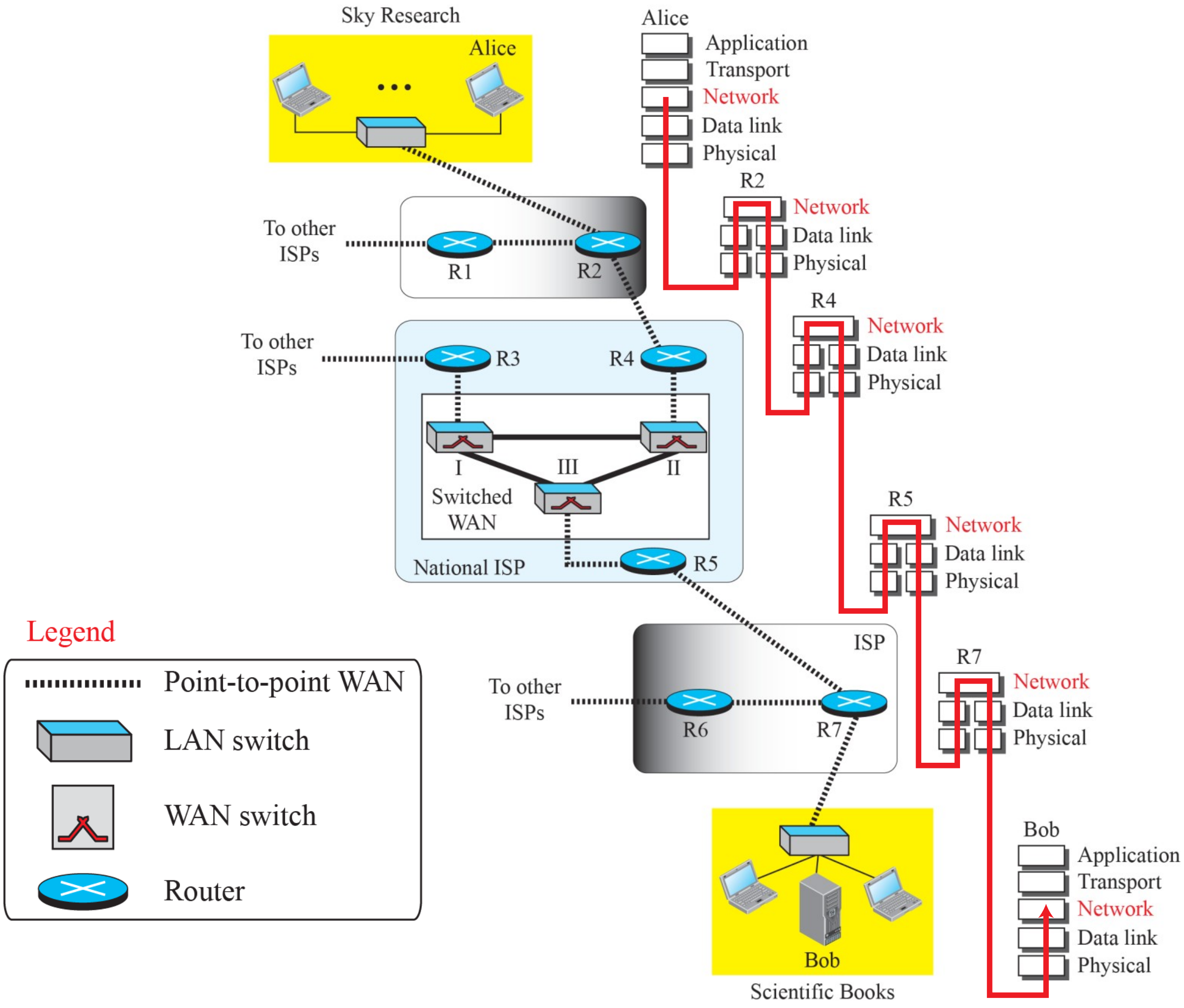
Introduction to Network Layer



18-1 NETWORK-LAYER SERVICES

Before discussing the network layer in the Internet today, let's briefly discuss the network-layer services that, in general, are expected from a network-layer protocol. Figure 18.1 shows the communication between Alice and Bob at the network layer. This is the same scenario we used in Chapters 3 and 9 to show the communication at the physical and the data-link layers, respectively.

Figure 18.1: Communication at the network layer



18.18.1 Packetizing

The first duty of the network layer is definitely packetizing: encapsulating the payload in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination. In other words, one duty of the network layer is to carry a payload from the source to the destination without changing it or using it. The network layer is doing the service of a carrier such as the postal office, which is responsible for delivery of packages from a sender to a receiver without changing or using the contents.

18.18.2 Routing and Forwarding

Other duties of the network layer, which are as important as the first, are routing and forwarding, which are directly related to each other.

Routing

- Network is a combination of LANs and WANs connected via Routers.
- More than one route from source to destination.
- Finding the best route.
- Coordinate knowledge about neighbors.
- Come up with consistent tables that are used when packets arrive.
- Routing protocols are:
  Routing Information **Protocol** (RIP)
- Interior Gateway **Protocol** (IGRP)
- Open Shortest Path First (OSPF)
- Exterior Gateway **Protocol** (EGP)
- Enhanced Interior Gateway **Routing Protocol** (EIGRP)

Forwarding

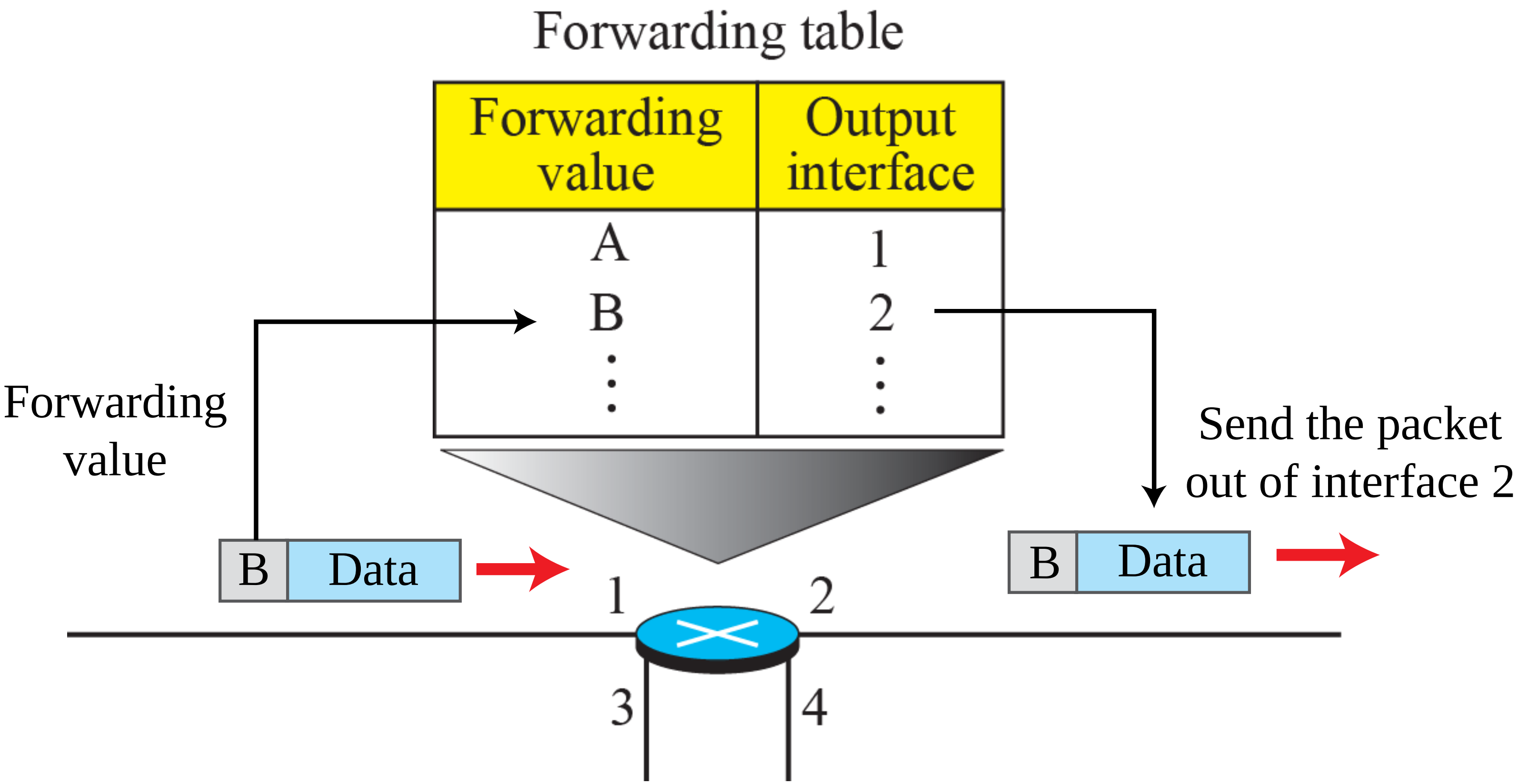
Routing is applying strategies and running some protocols to create the decision-making tables for each router.

Forwarding can be defined as the actions applied by router when a packet arrives at its interface.

Tables are called forwarding tables or Routing tables.

Use information in the packet header to find corresponding output interface using the tables.

Figure 18.2: Forwarding process



Forwarding table

Forwarding value	Output interface
A	1
B	2
⋮	⋮

18.18.3 Other Services

Let us briefly discuss other services expected from the network layer.

Error Control

- Can be implemented but usually ignored.
- Inefficient due to fragmentation of packets at routers.
- Checksum is there to check header.
- Whole datagram is not checked.
- **ICMP provides error control if datagrams are discarded or if there is some unknown information in the header.**

Flow Control

- Does not directly provide any flow control.
- Reasons are:
  - Job at receiver is simple so rare chances of overwhelming.
  - Transport layer have buffers to control flow.
  - Another level of flow control adds complications
  - System becomes less efficient.

18-2 PACKET SWITCHING

From the discussion of routing and forwarding in the previous section, 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.

Packet Switching

- Two Switching Techniques:
  - Circuit Switching(Use at Physical layers)
  - **Packet Switching(Network Layer)**
- Source and Destination sends and receives packets one by one.
- Destination waits for all the packets of the same message to arrive before sending it to Transport Layer.
- Two approaches to route packets:
  - Datagram Approach(connectionless)
  - Virtual Circuit Approach(connection oriented)

18.2.1 Datagram Approach

When the Internet started, to make it simple, 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 shows the idea..

Datagram Approach- Connectionless Service

- The switches in this type of network are called Routers.
- Each packet independent entity. No relationship between packets belonging to the same message.
- Packet is routed on the basis of source and destination information in the header.
- Forwarding is based on destination address.

Figure 18.3: A connectionless packet-switched network

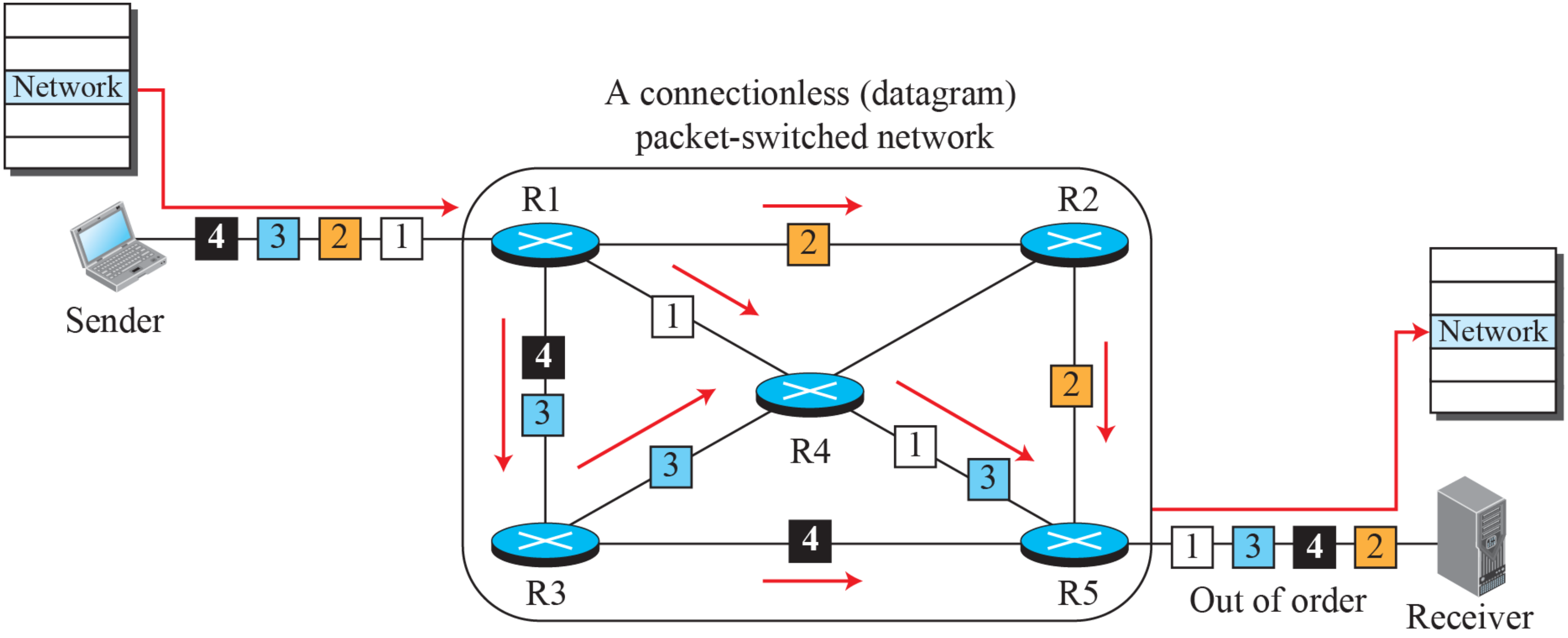
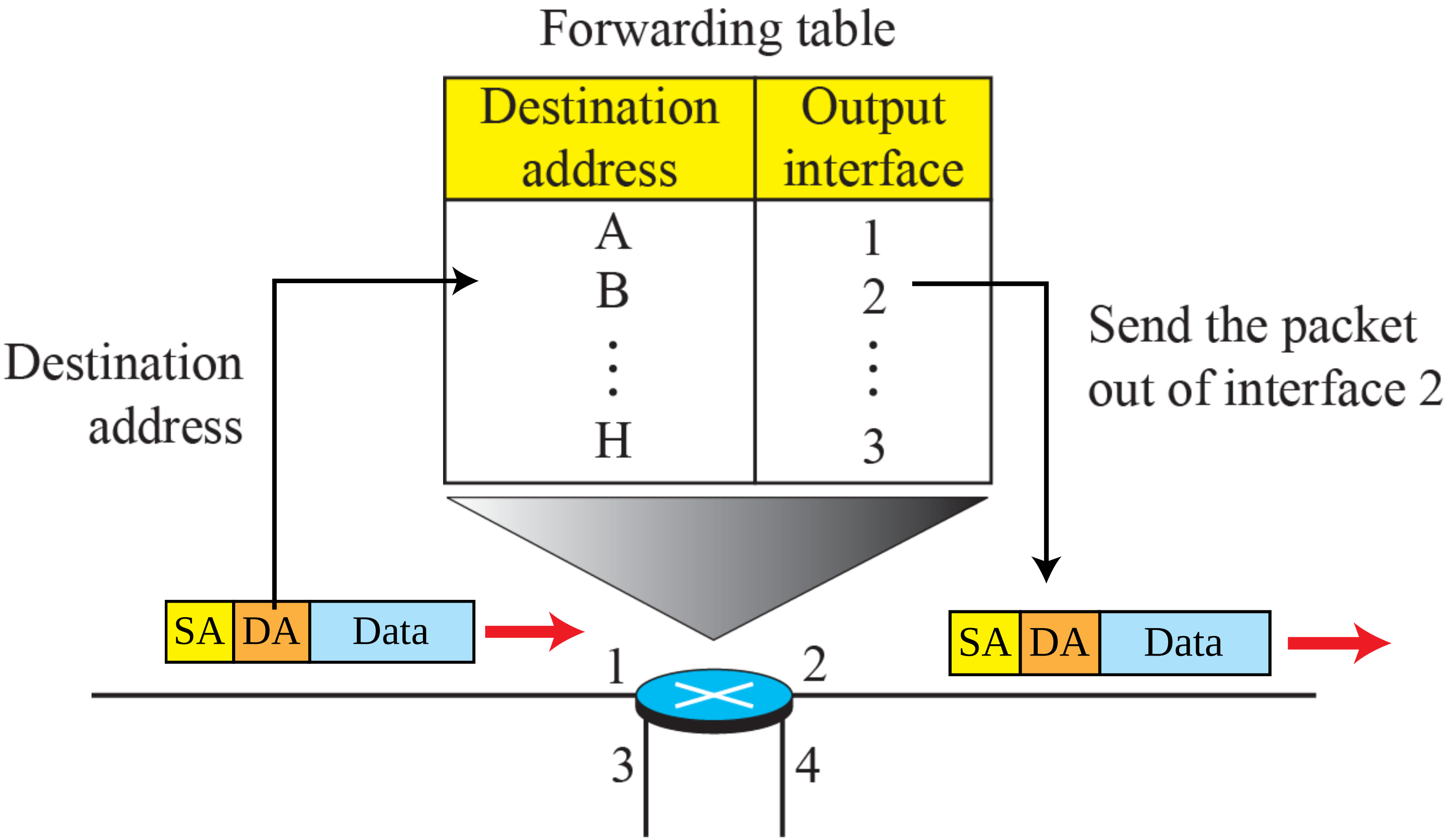


Figure 18.4: Forwarding process in a router when used in a connectionless network



Forwarding table

Destination address	Output interface
A	1
B	2
⋮	⋮
H	3

18.2.2 Virtual-Circuit Approach

In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path. In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.

18.2.2 Virtual-Circuit Approach

To Create a connection-Oriented Service a three phase process is used:

- Set up
- Data Transfer
- Tear Down

Set Up Phase: Source and Destination addresses are used to make table entries for the connection oriented service.

Tear Down Phase: Source and Destination inform the router to delete corresponding entries.

Figure 18.5: A virtual-circuit packet-switched network

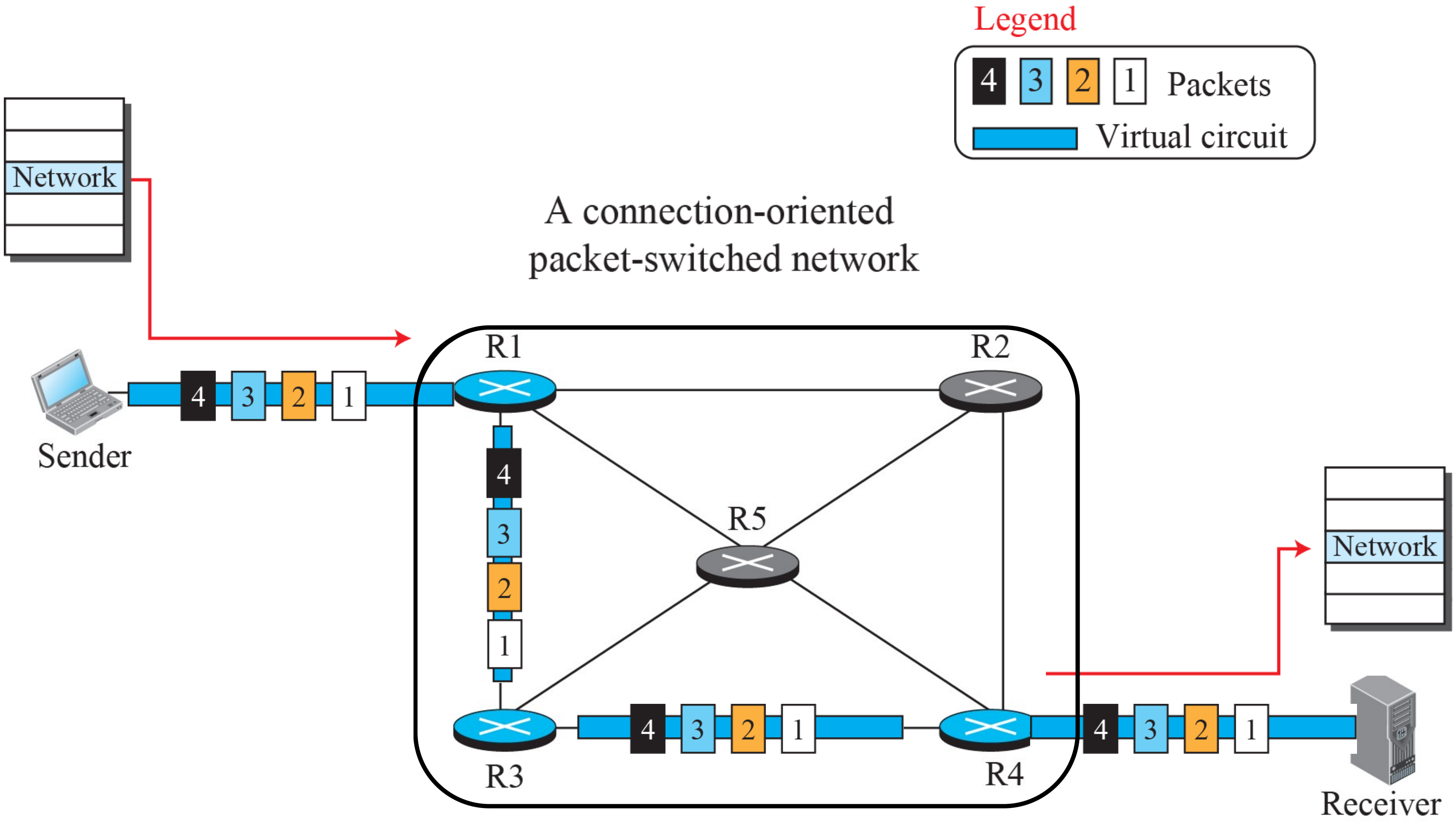
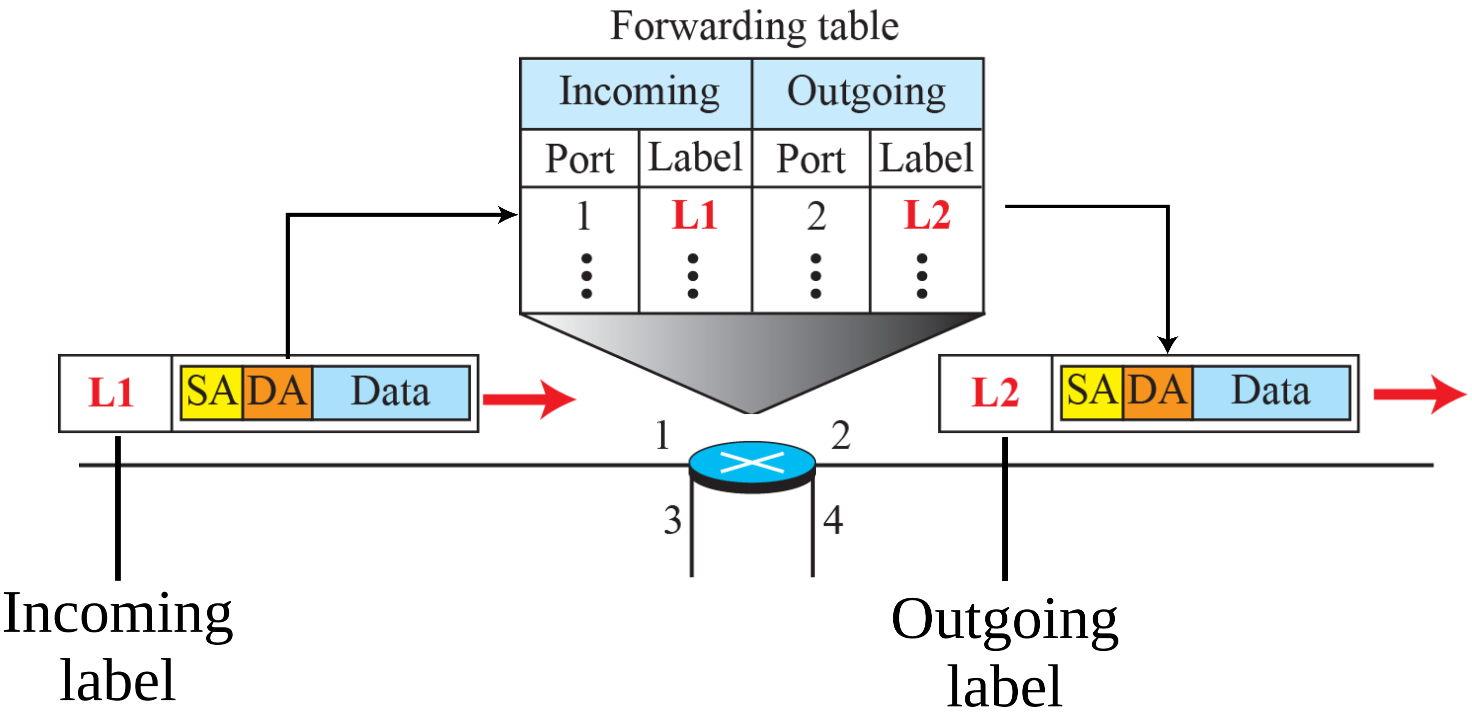


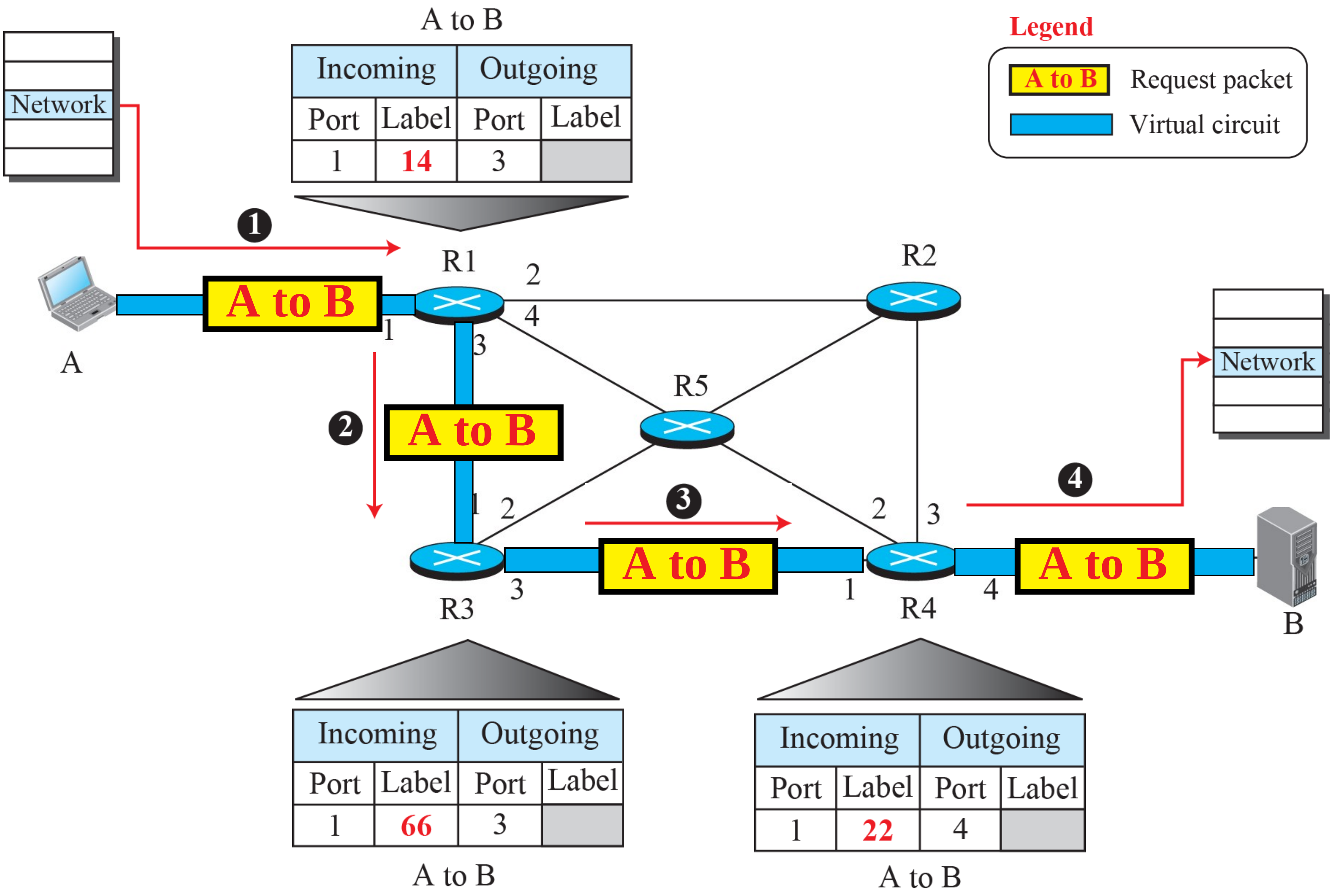
Figure 18.6: Forwarding process in a router when used in a virtual circuit network



Forwarding table

Incoming		Outgoing	
Port	Label	Port	Label
1	L1	2	L2
⋮	⋮	⋮	⋮

Figure 18.7: Sending request packet in a virtual-circuit network



A to B

Incoming		Outgoing	
Port	Label	Port	Label
1	14	3	

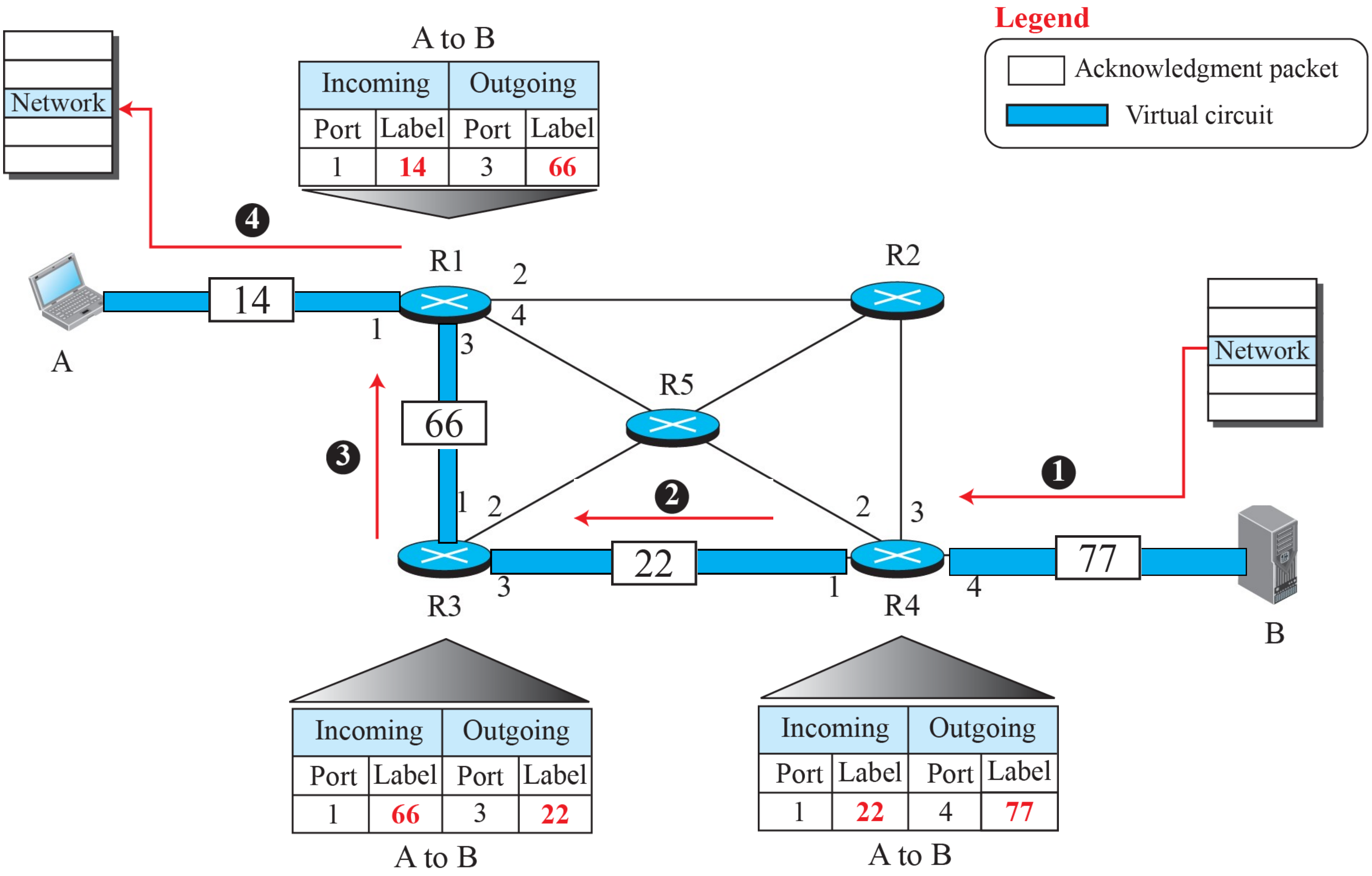
A to B

Incoming		Outgoing	
Port	Label	Port	Label
1	66	3	

A to B

Incoming		Outgoing	
Port	Label	Port	Label
1	22	4	

Figure 18.8: Sending acknowledgments in a virtual-circuit network



A to B

Incoming		Outgoing	
Port	Label	Port	Label
1	**14**	3	**66**

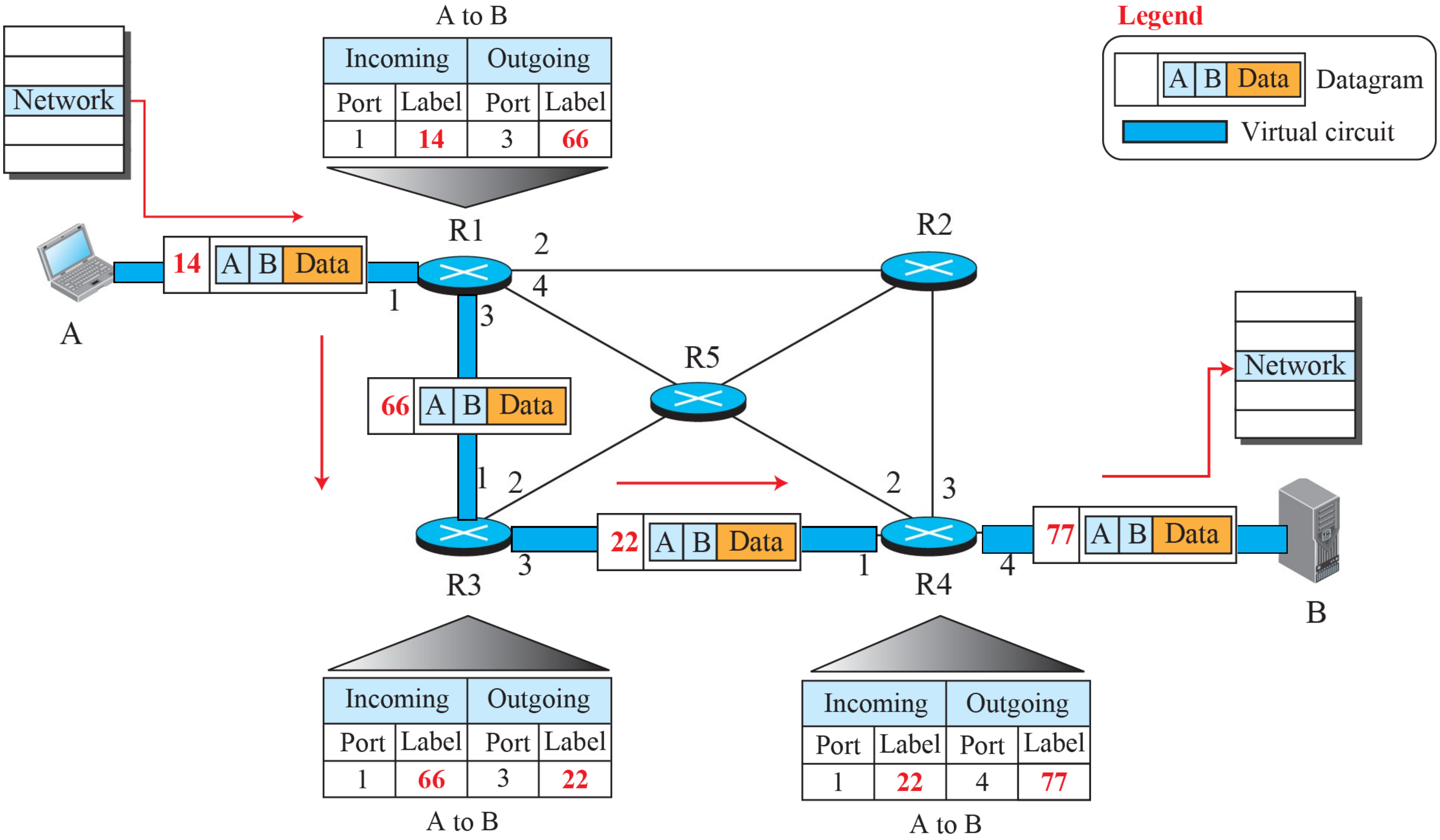
Incoming		Outgoing	
Port	Label	Port	Label
1	**66**	3	**22**

A to B

Incoming		Outgoing	
Port	Label	Port	Label
1	**22**	4	**77**

A to B

Figure 18.9: Flow of one packet in an established virtual circuit



A to B (R1)

Incoming		Outgoing	
Port	Label	Port	Label
1	14	3	66

A to B (R3)

Incoming		Outgoing	
Port	Label	Port	Label
1	66	3	22

A to B (R4)

Incoming		Outgoing	
Port	Label	Port	Label
1	22	4	77

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.

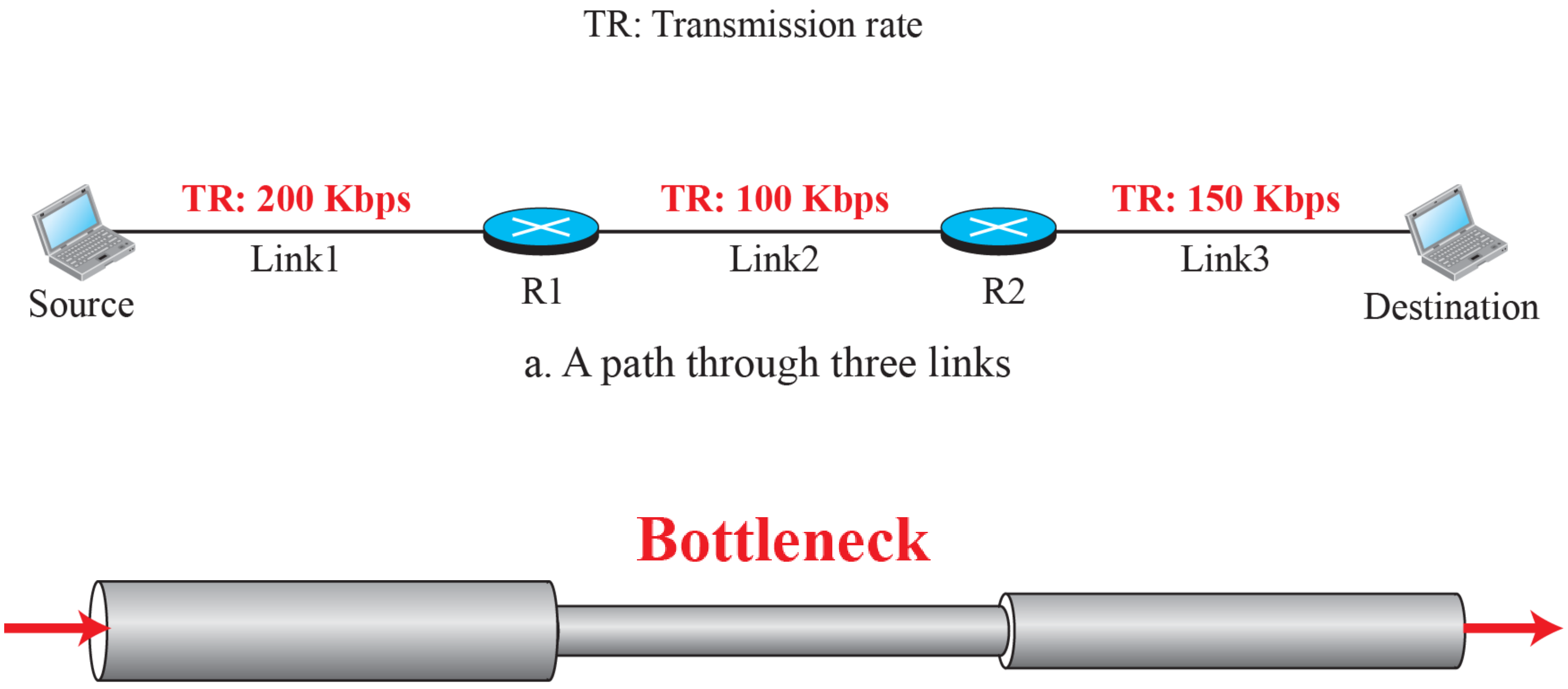
18.3.1 Delay

All of us expect instantaneous response from a network, but a packet, from its source to its destination, encounters delays. The delays in a network can be divided into four types: transmission delay, propagation delay, processing delay, and queuing delay. Let us first discuss each of these delay types and then show how to calculate a packet delay from the source to the destination..

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 (networks), each with a different transmission rate. How, then, can we determine the throughput of the whole path? To see the situation, assume that we have three links, each with a different transmission rate, as shown in Figure 18.10.

Figure 18.10: Throughput in a path with three links in a series



a. A path through three links

b. Simulation using pipes

Figure 18.11: A path through the Internet backbone

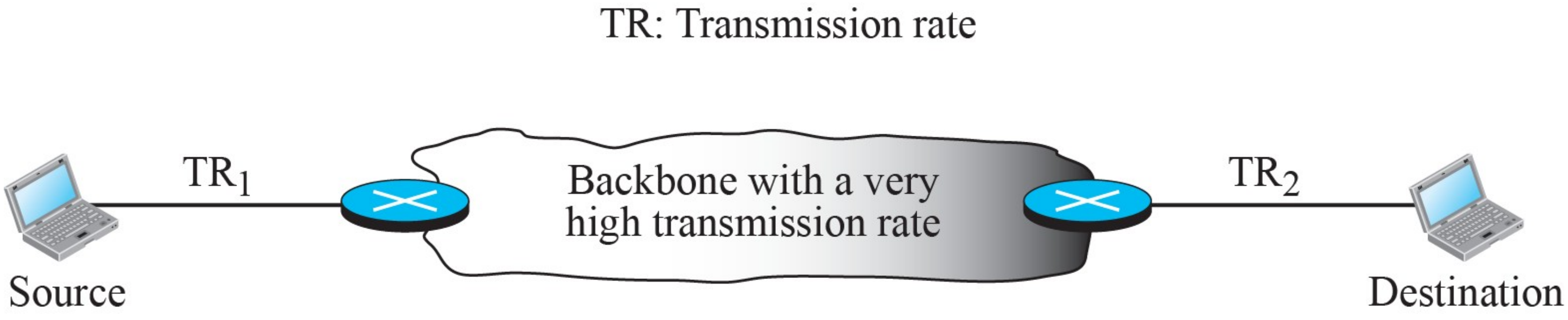
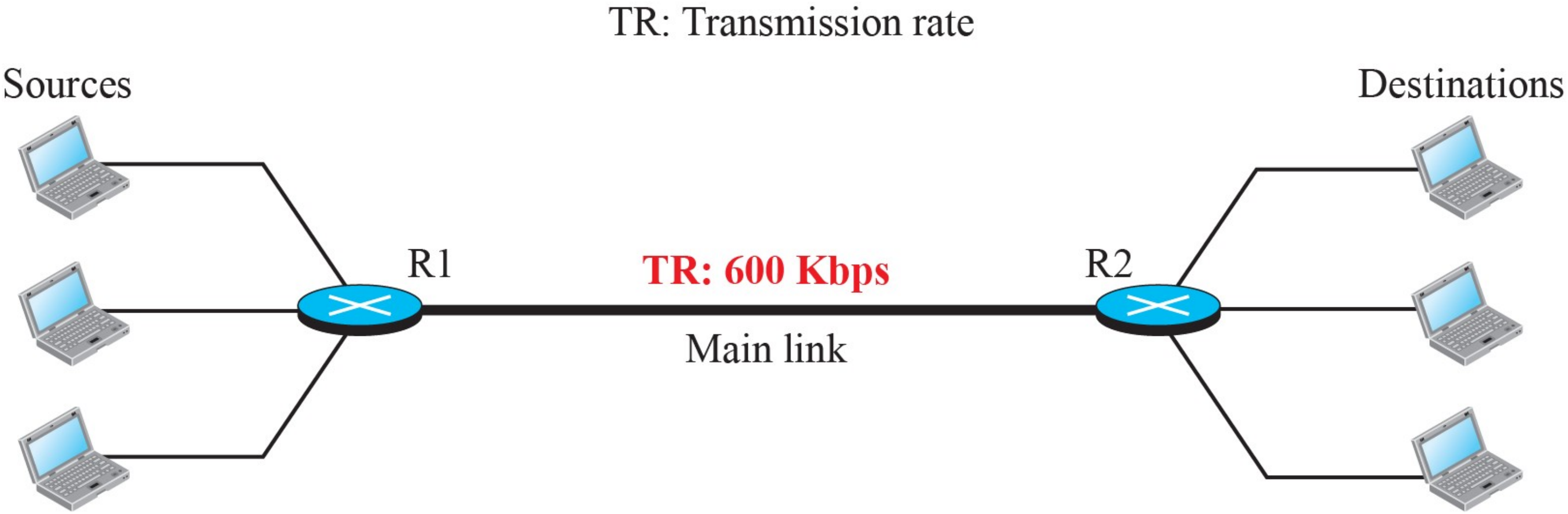


Figure 18.12: Effect of throughput in shared links



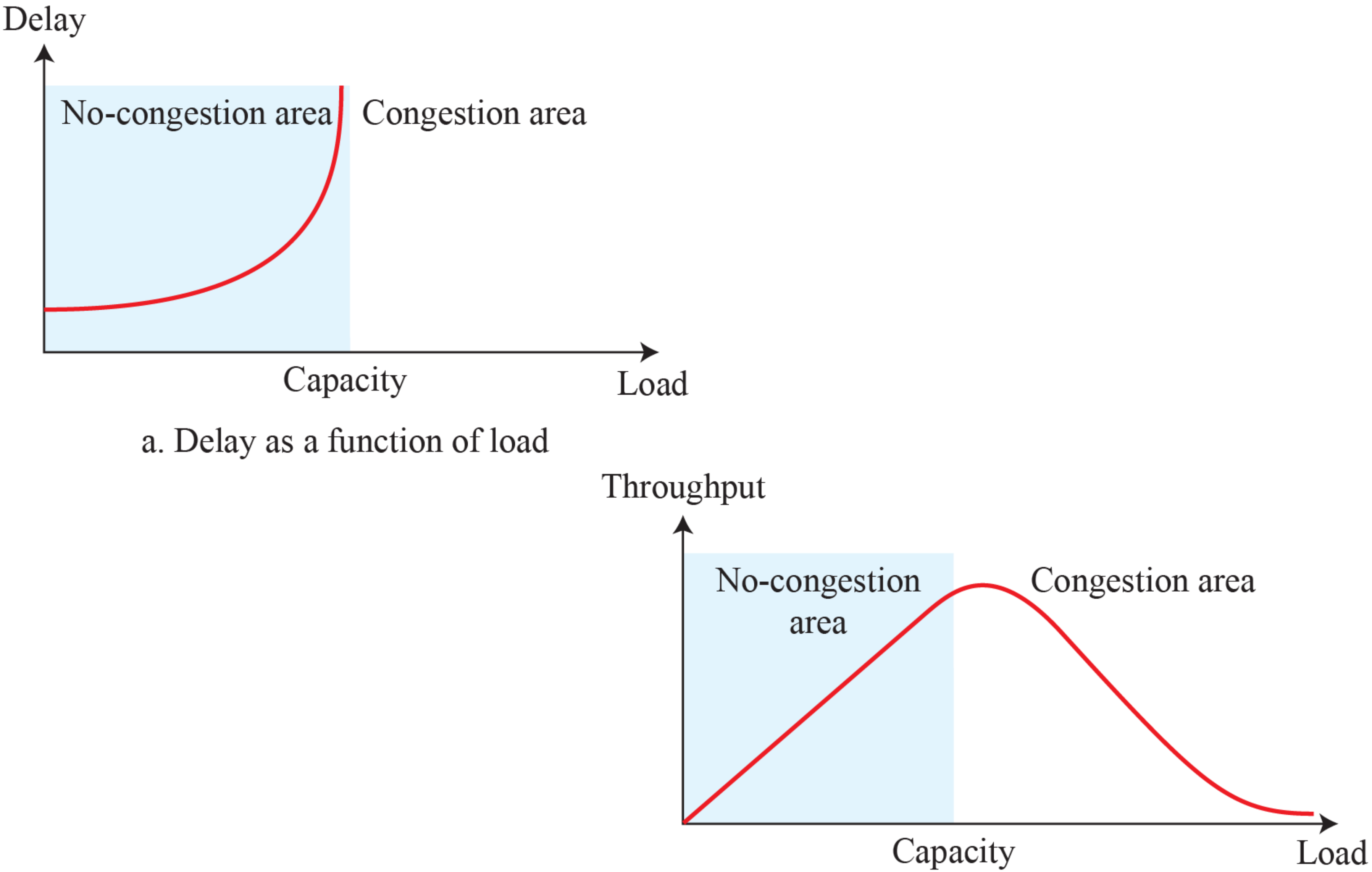
18.3.3 Packet Loss

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 on the Internet network layer is that the packet needs to be resent, which in turn may create overflow and cause more packet loss.

18.3.4 Congestion Control

Congestion control is a mechanism for improving performance. In Chapter 23, we will discuss congestion at the transport layer. Although congestion at the network layer is not explicitly addressed in the Internet model, the study of congestion at this layer may help us to better understand the cause of congestion at the transport layer and find possible remedies to be used at the network layer. Congestion at the network layer is related to two issues, throughput and delay, which we discussed in the previous section.

Figure 18.13. Packet delay and throughput as functions of load



a. Delay as a function of load

b. Throughput as a function of load

Figure 18.14: Backpressure method for alleviating congestion

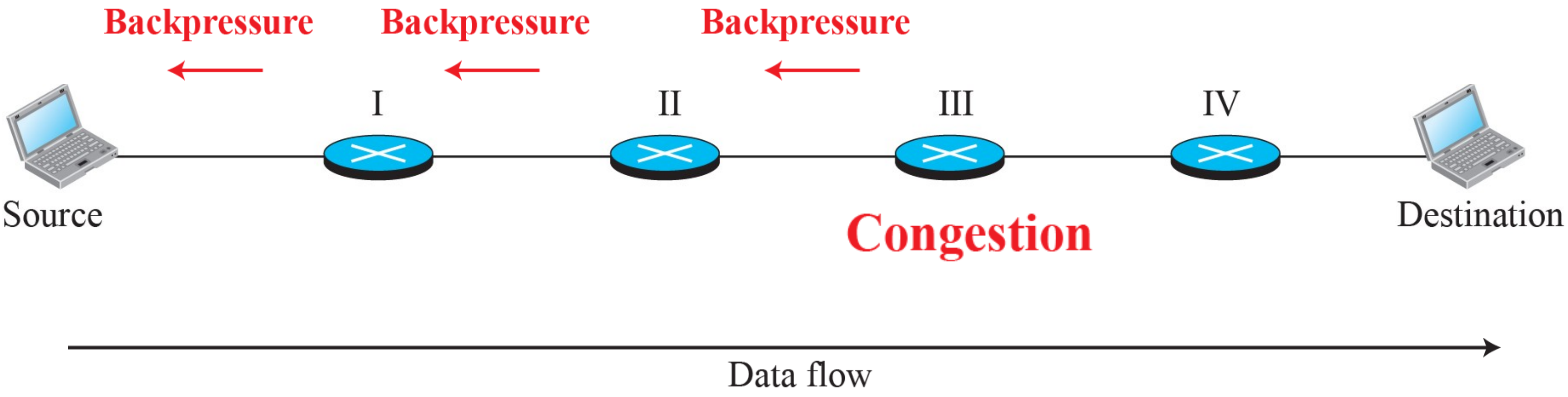
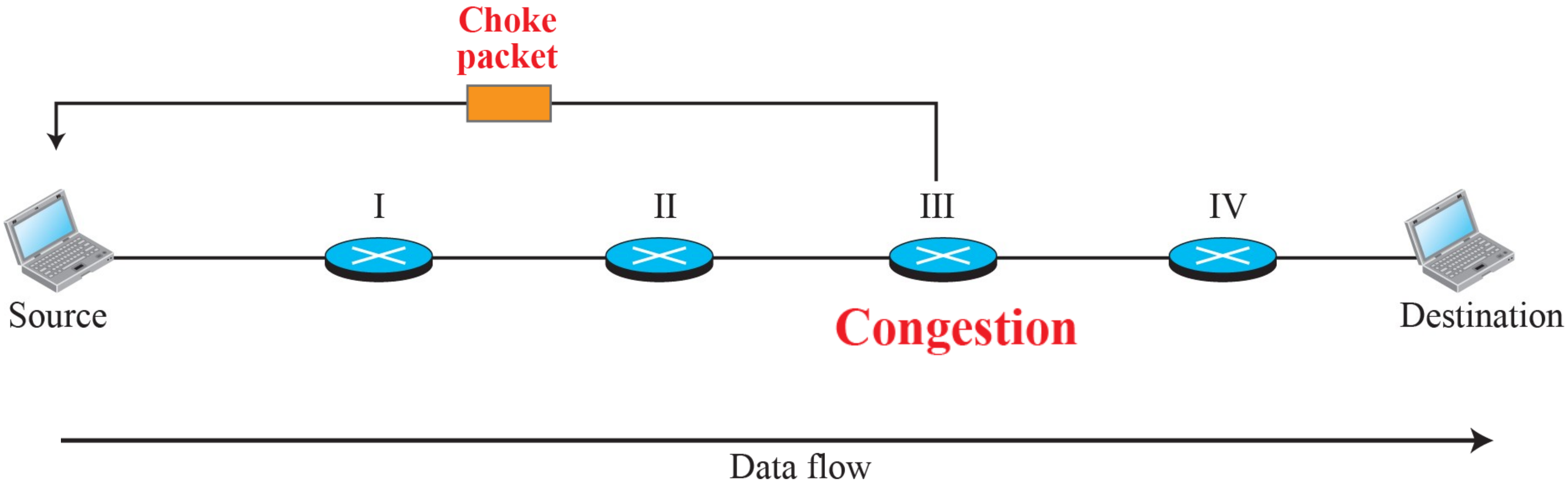


Figure 4.15: Choke packet



18-4 IPv4 ADDRESSES

The identifier used in the IP 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. If a protocol uses b bits to define an address, the address space is 2b because each bit can have two different values (0 or 1). IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than four billion). If there were no restrictions, more than 4 billion devices could be connected to the Internet.

Figure 18.16: Three different notations in IPv4 addressing

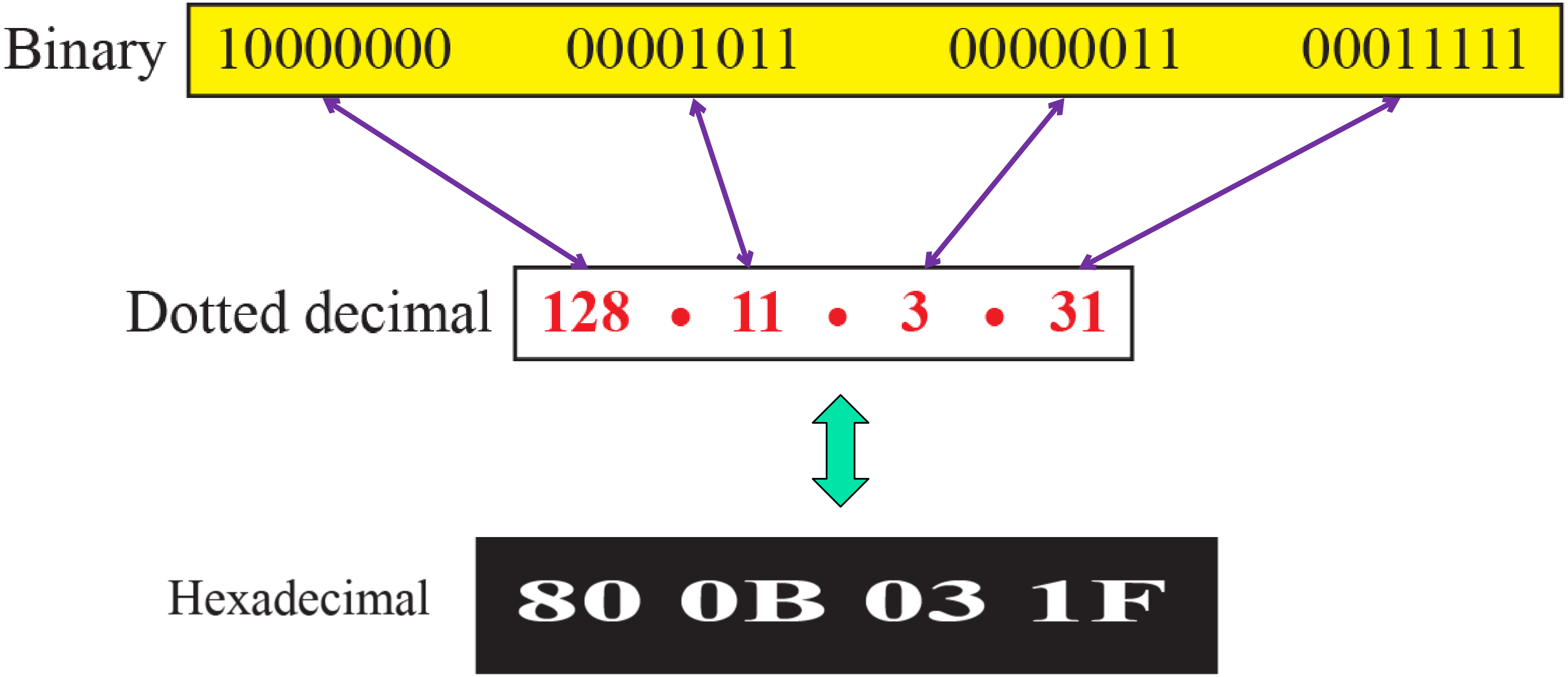
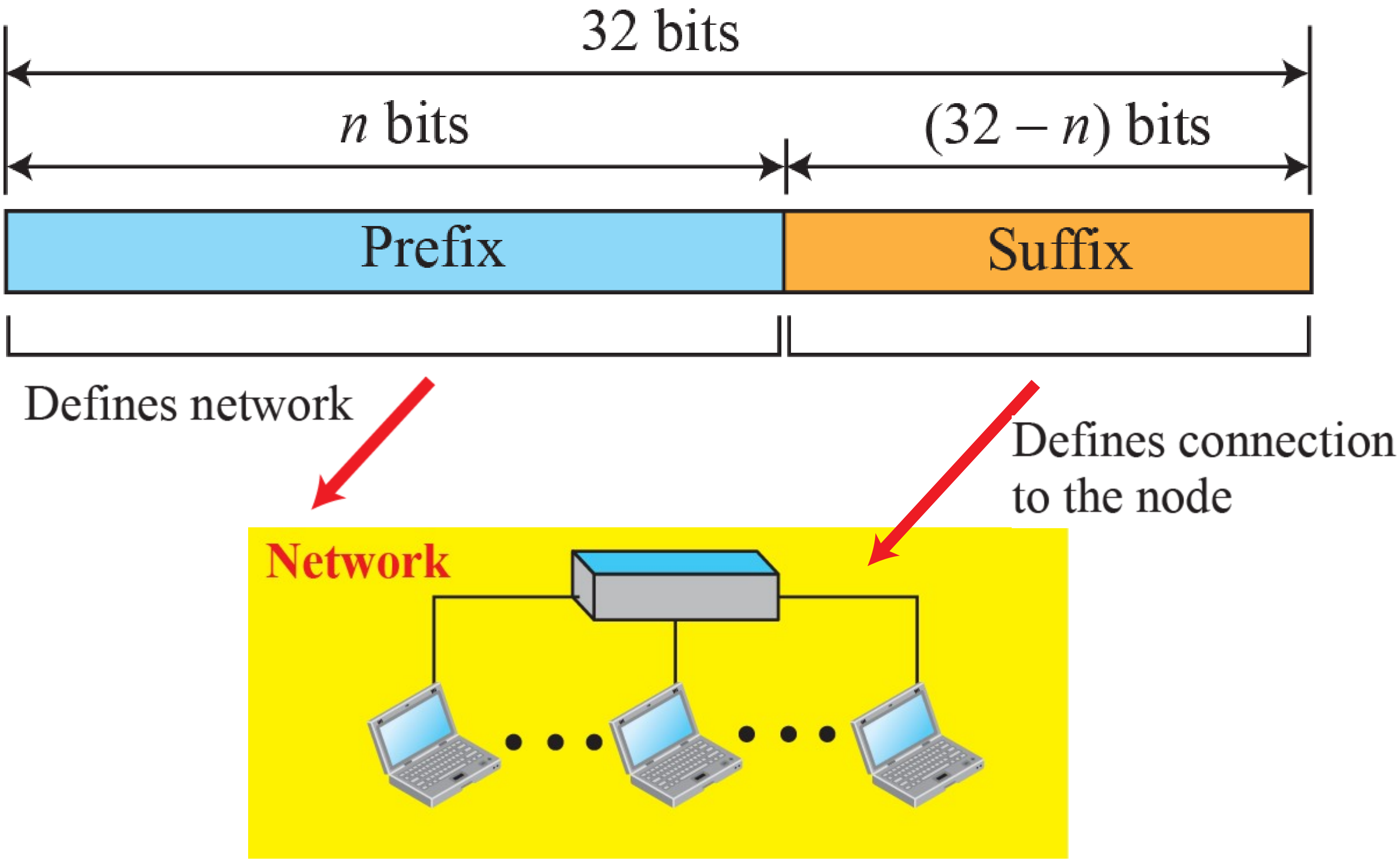


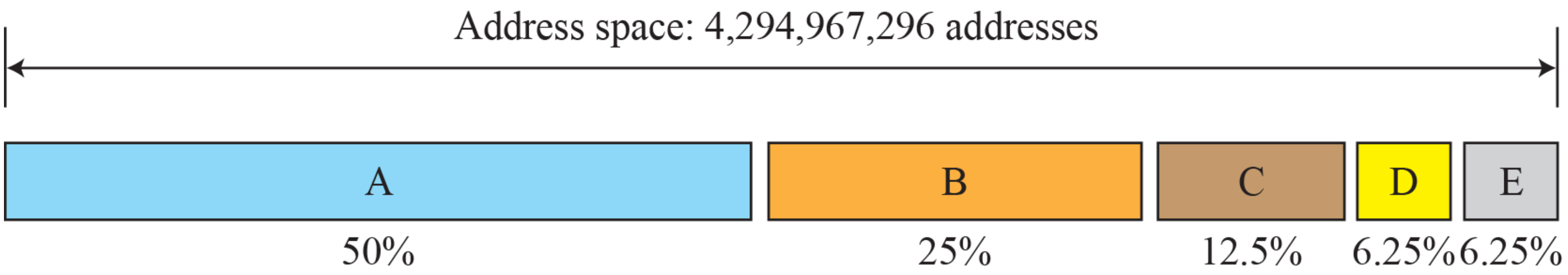
Figure 18.17: Hierarchy in addressing

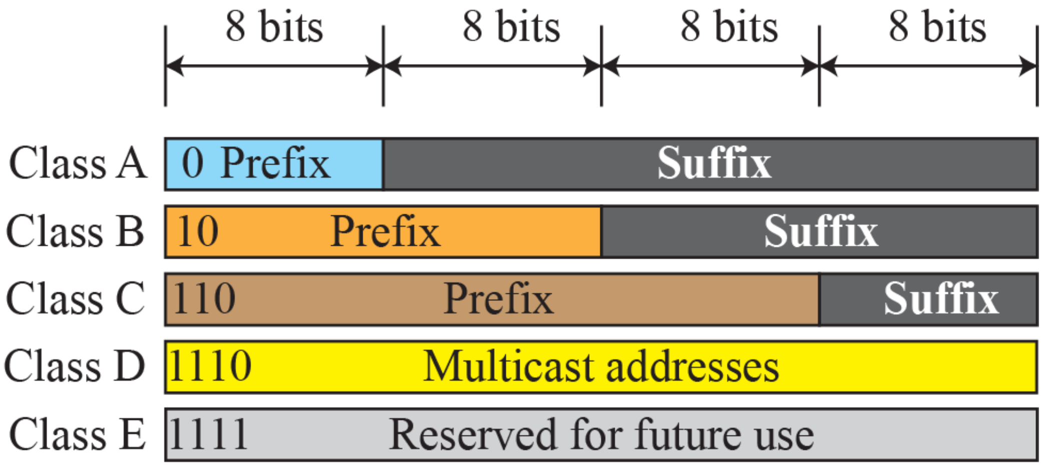


18.4.2 Classful Addressing

When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24). The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure 18.18. This scheme is referred to as classful addressing. Although classful addressing belongs to the past, it helps us to understand classless addressing, discussed later.

Figure 18.18: Occupation of the address space in classful addressing





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. The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing.

Figure 18.19: Variable-length blocks in classless addressing

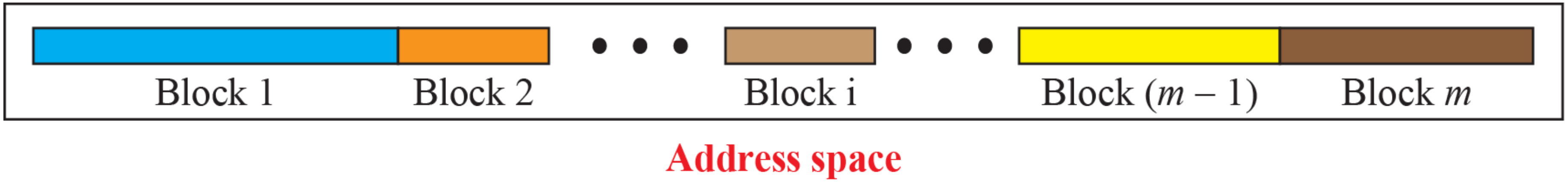
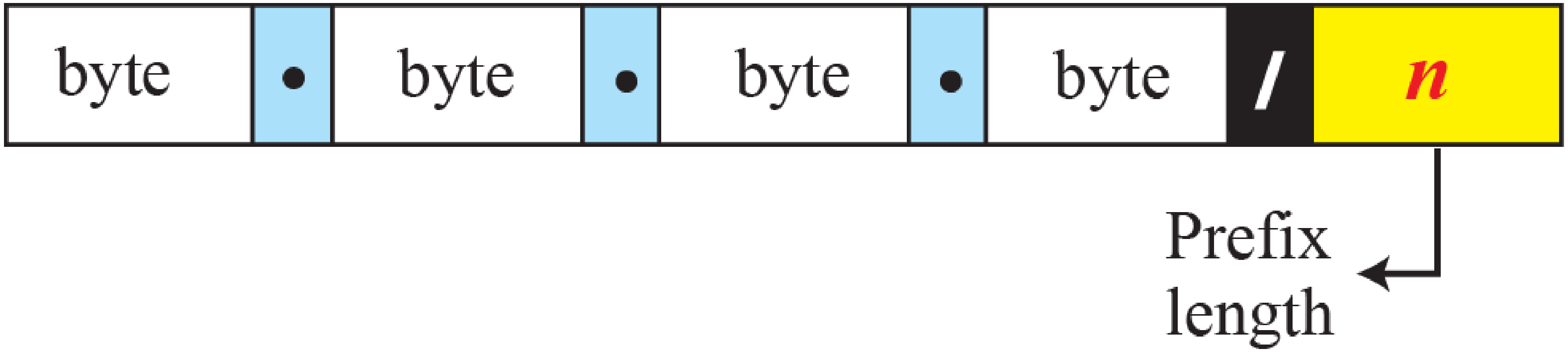


Figure 18.20: Slash notation (CIDR)



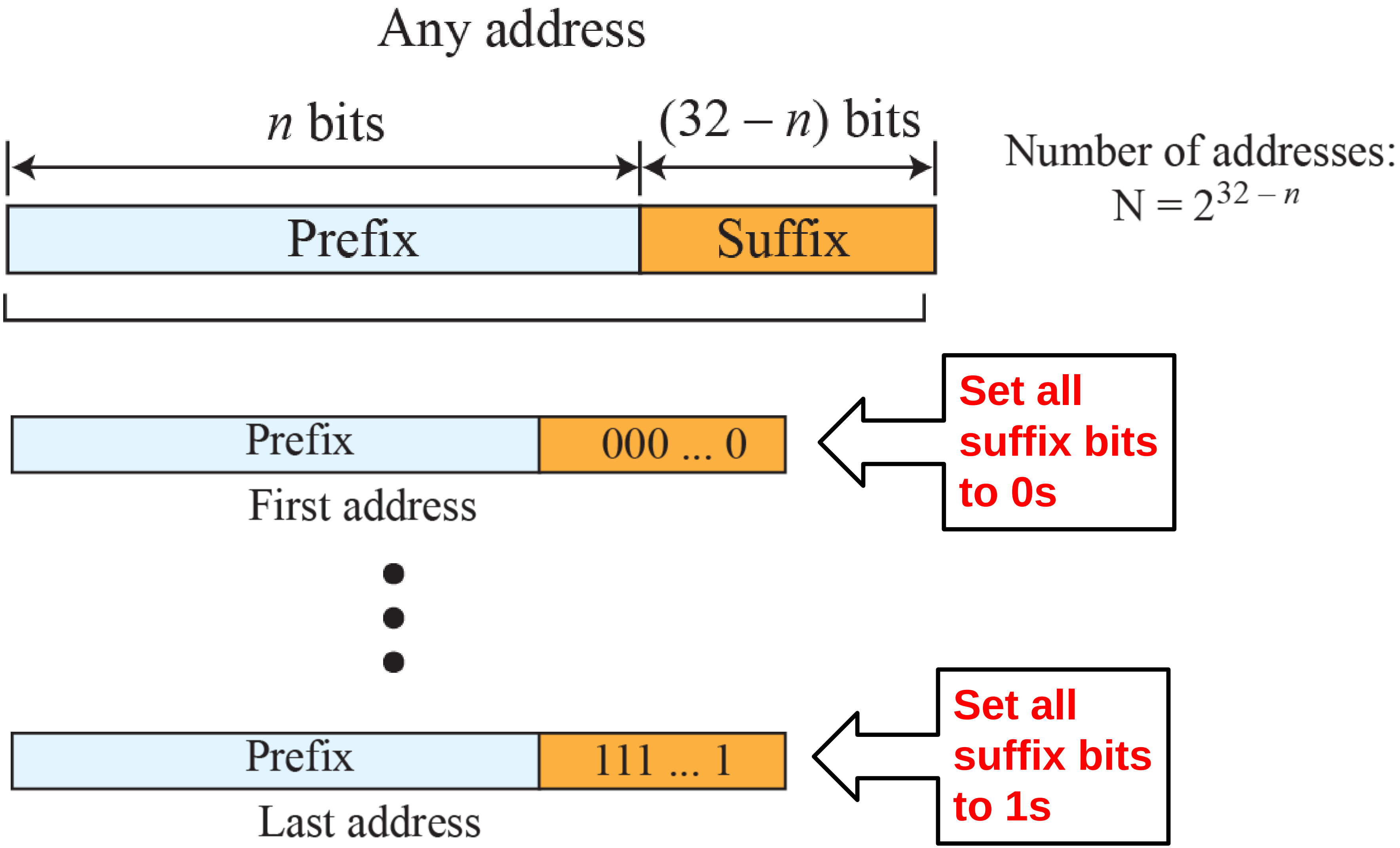
Examples:

12.24.76.8/**8**

23.14.67.92/**12**

220.8.24.255/**25**

Figure 18.21: Information extraction in classless addressing

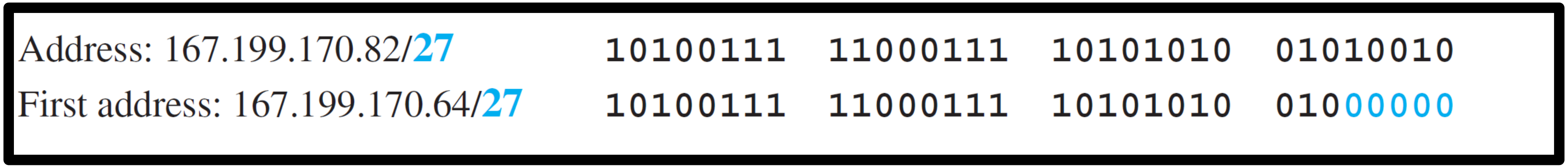


Set all suffix bits to 0s

Set all suffix bits to 1s

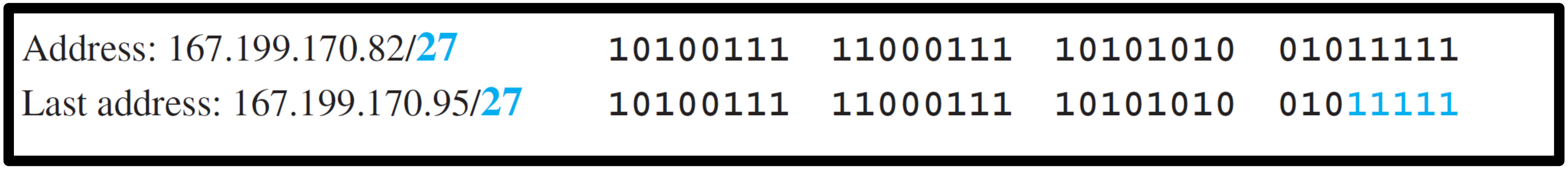
Example 18.1

A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows. The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses. 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

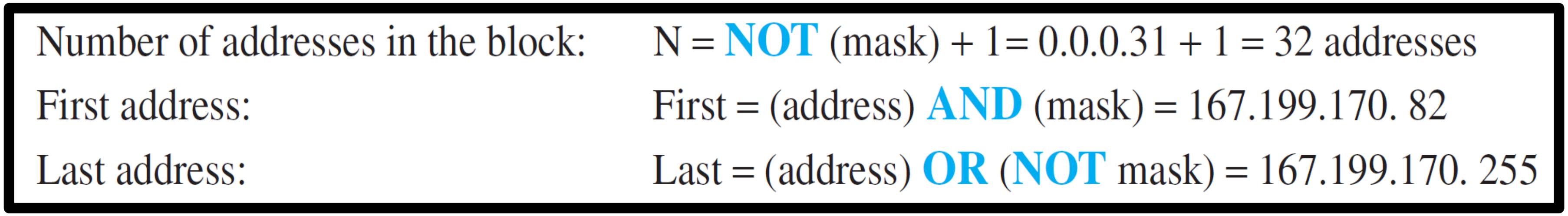
The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.



Address: 167.199.170.82/**27**	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/**27**	10100111	11000111	10101010	01011111

Example 18.2

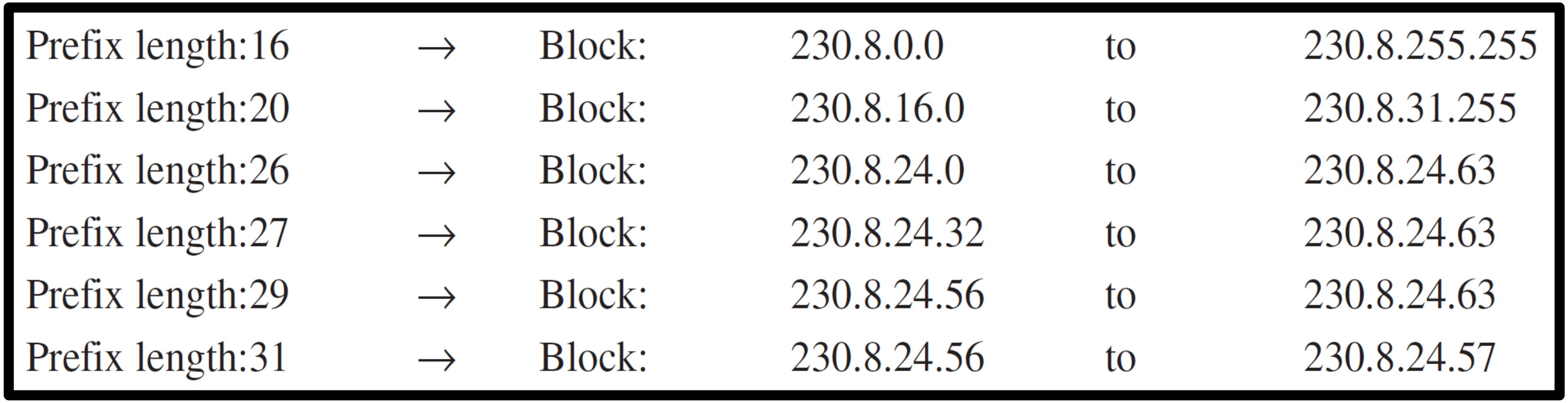
We repeat Example 18.1 using the mask. The mask in dotted-decimal notation is 256.256.256.224 The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.



Number of addresses in the block:	N = **NOT** (mask) + 1= 0.0.0.31 + 1 = 32 addresses
First address:	First = (address) **AND** (mask) = 167.199.170. 82
Last address:	Last = (address) **OR** (**NOT** mask) = 167.199.170. 255

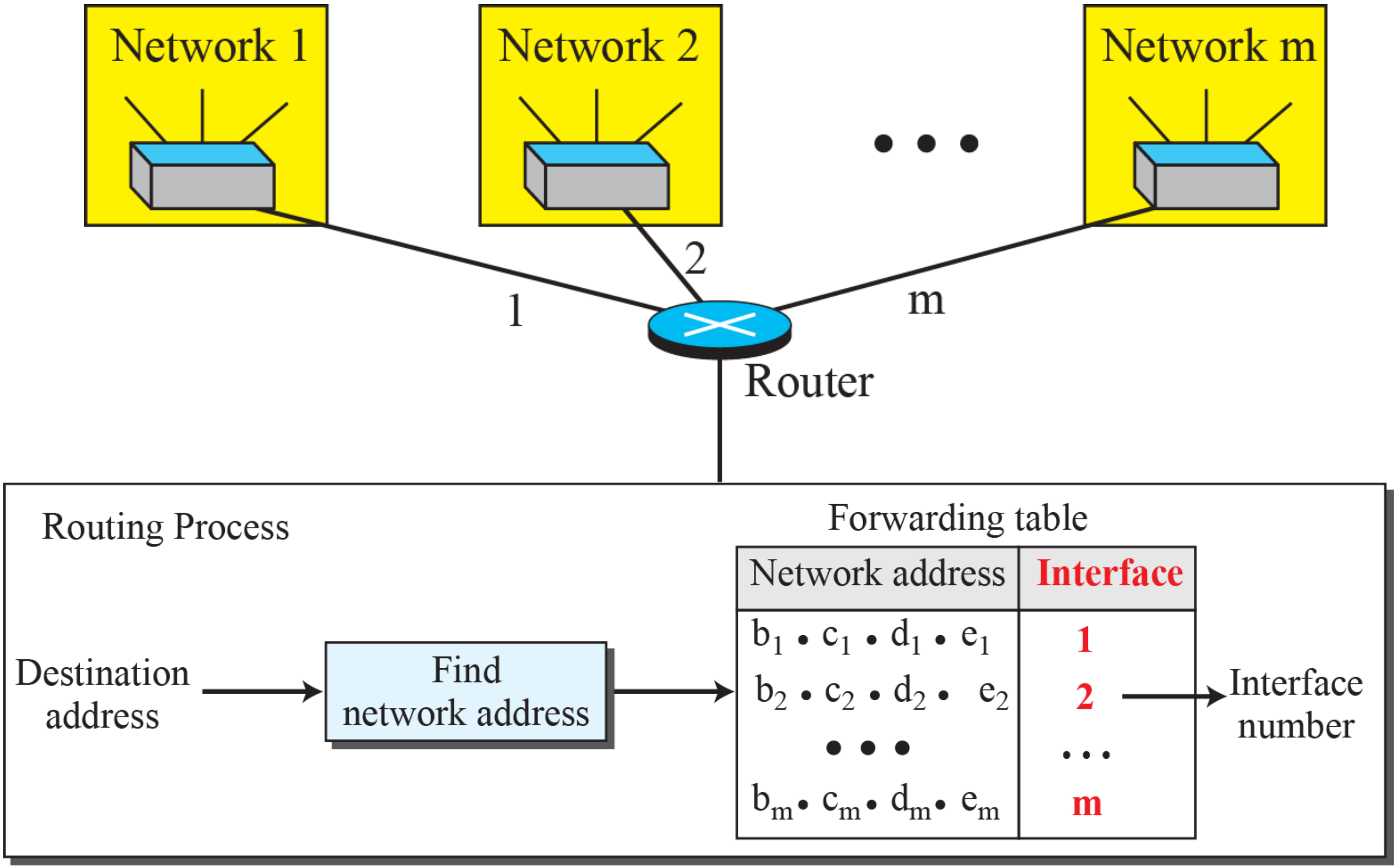
Example 18.3

In classless addressing, an address cannot per se define the block the address belongs to. For example, the address 230.8.24.56 can belong to many blocks. Some of them are shown below with the value of the prefix associated with that block.



Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

Figure 18.22: Network address



Forwarding table

Network address	Interface
$b_1 \bullet c_1 \bullet d_1 \bullet e_1$	1
$b_2 \bullet c_2 \bullet d_2 \bullet e_2$	2
• • •	. . .
$b_m \bullet c_m \bullet d_m \bullet e_m$	m

Example 18.4

An ISP has requested a block of 1000 addresses. Since 1000 is not a power of 2, 1024 addresses are granted. The prefix length is calculated as $n = 32 - \log_2 1024 = 22$. An available block, 18.14.12.0**/22**, is granted to the ISP. It can be seen that the first address in decimal is 302,910,464, which is divisible by 1024.

Example 18.5

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.

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**. To satisfy the third requirement, we assign addresses to subblocks, starting with the largest and ending with the smallest one.

xample 18.5 (continued)

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**.

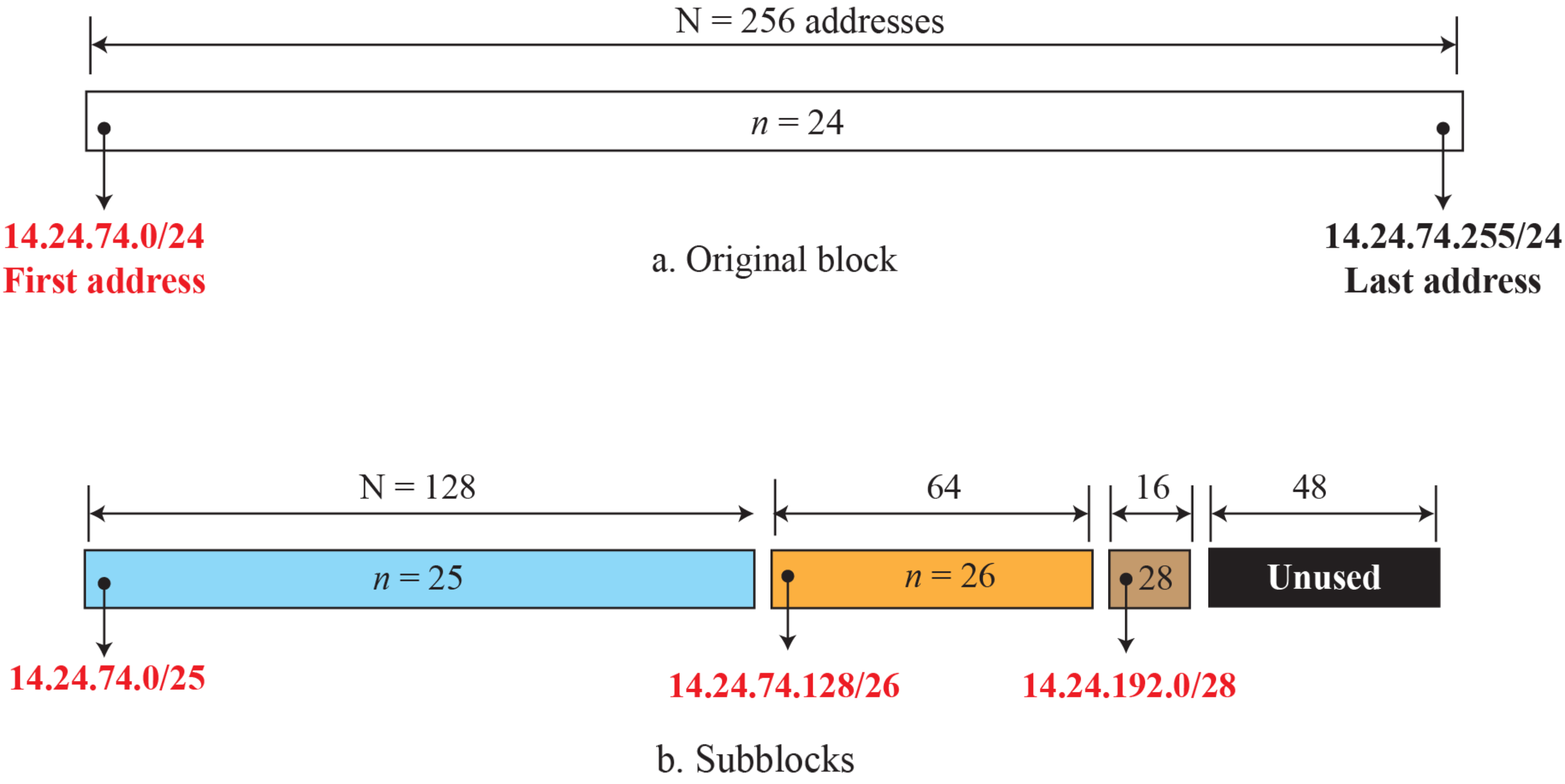
b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n_2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128/**26**; the last address is 14.24.74.191/**26**.

xample 18.5 (continued)

c. The number of addresses in the largest subblock, which requires 10 addresses, is not a power of 2. 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**.

If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255. We don’t know about the prefix length yet. Figure 18.23 shows the configuration of blocks. We have shown the first address in each block.

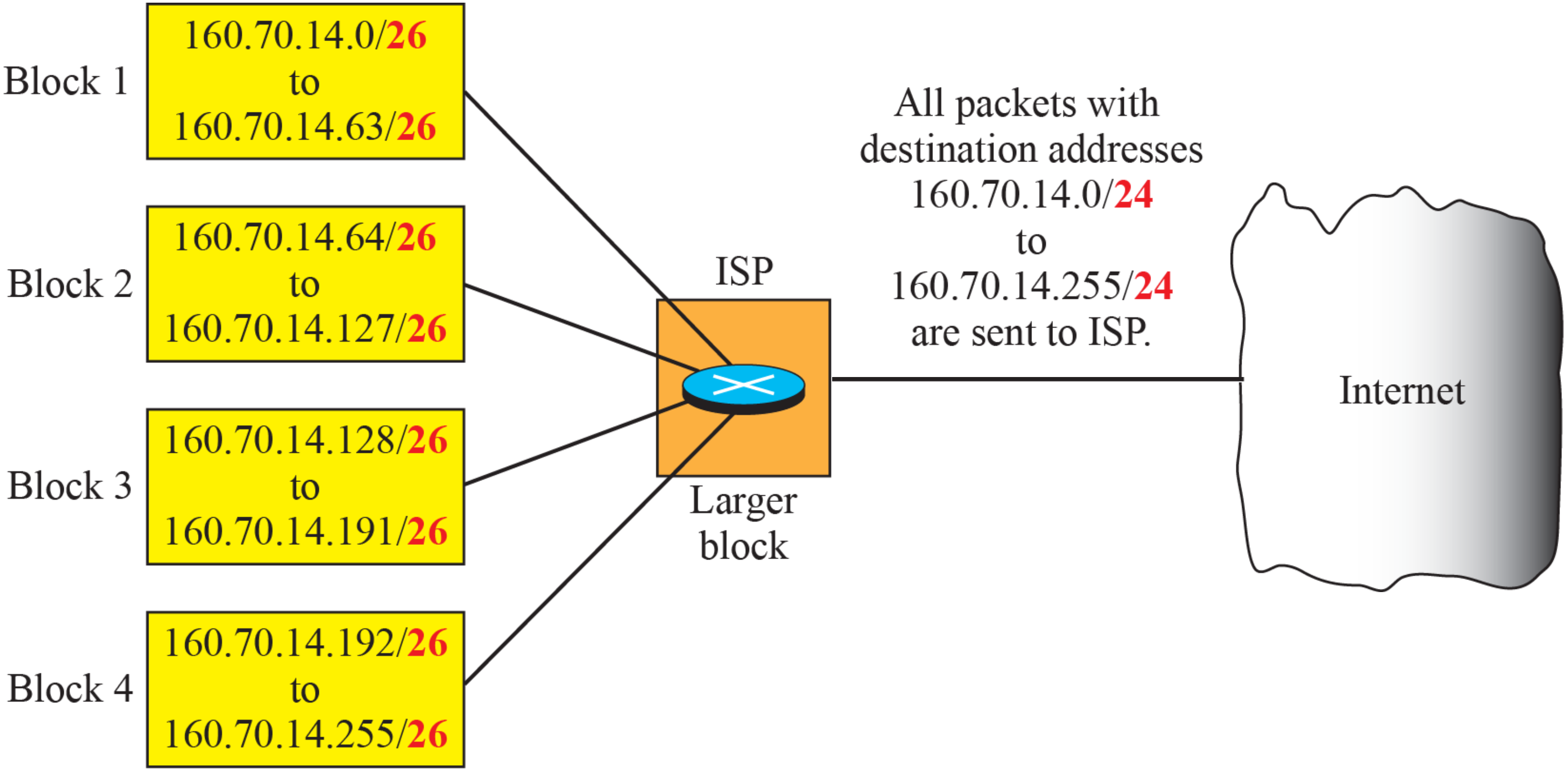
Figure 18.23: Solution to Example 4.5



Example 18.6

Figure 18.24 shows how four small blocks of addresses are assigned to four organizations by an ISP. The ISP combines these four blocks into one single block and advertises the larger block to the rest of the world. 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. This is similar to routing we can find in a postal network. All packages coming from outside a country are sent first to the capital and then distributed to the corresponding destination.

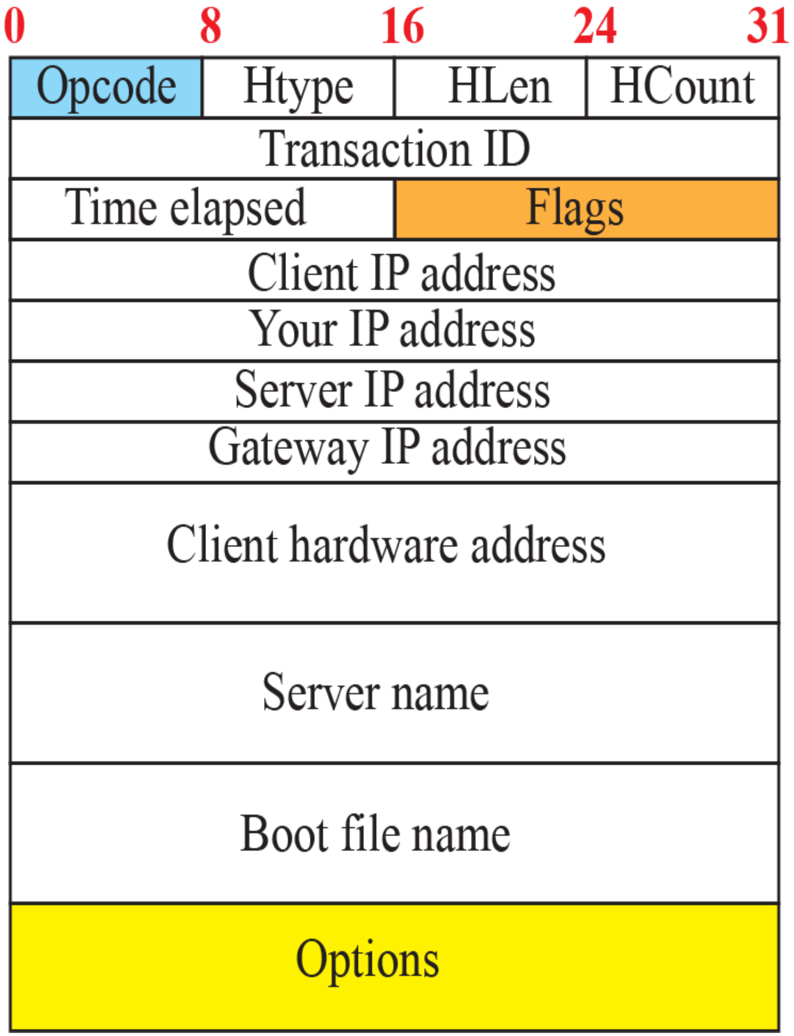
Figure 18.24: Example of address aggregation



18.4.4 DHCP

After a block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers. However, address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP). DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.

Figure 18.25: DHCP message format



0 8 16 24 31

Opcode Htype HLen HCount

Transaction ID

Time elapsed Flags

Client IP address

Your IP address

Server IP address

Gateway IP address

Client hardware address

Server name

Boot file name

Options

Fields:
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Lengh of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

Figure 18.26: Option format

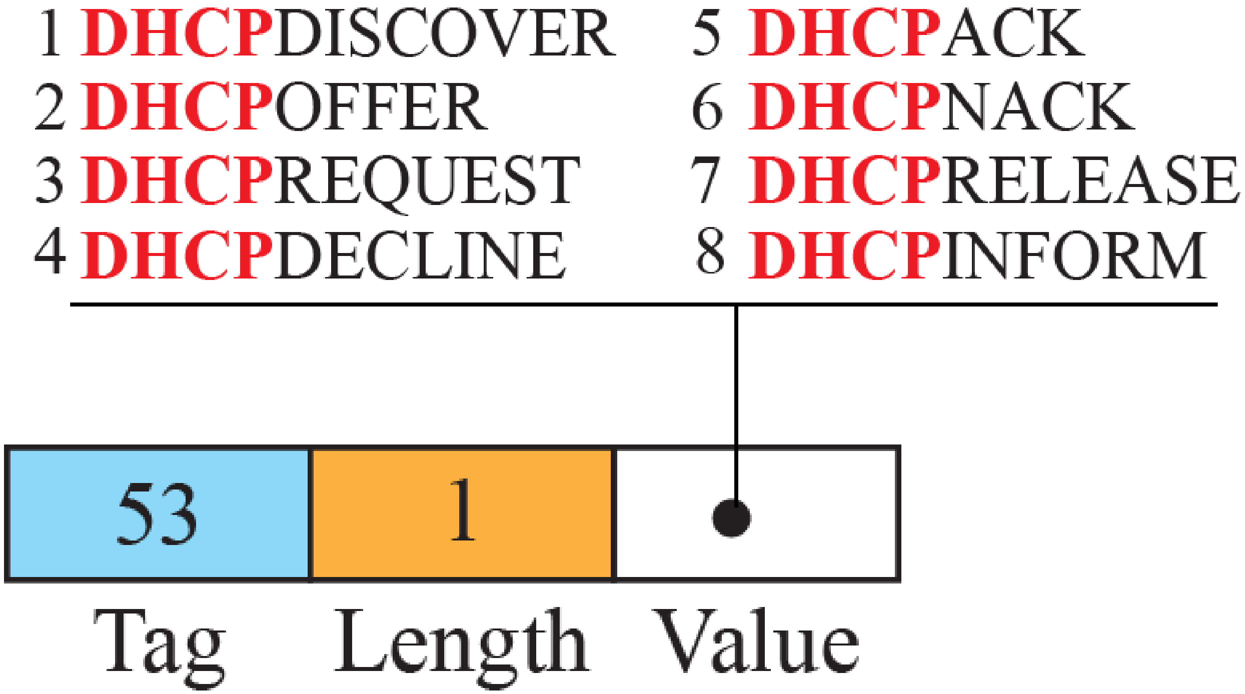
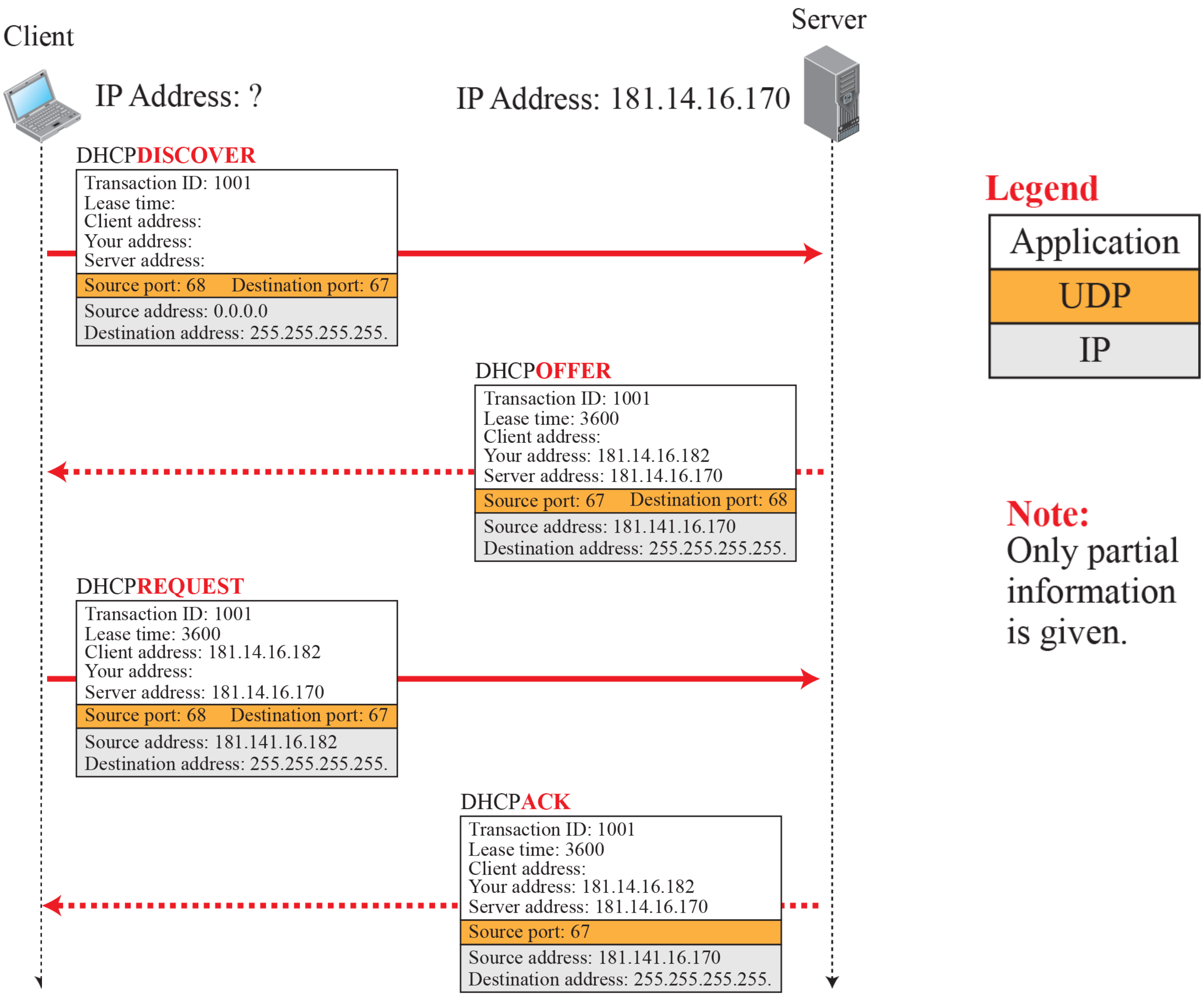


Figure 18.27: Operation of DHCP



DHCPDISCOVER

Transaction ID: 1001
Lease time:
Client address:
Your address:
Server address:
Source port: 68 Destination port: 67
Source address: 0.0.0.0
Destination address: 255.255.255.255.

DHCPOFFER

Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.141.16.170
Destination address: 255.255.255.255.

DHCPREQUEST

Transaction ID: 1001
Lease time: 3600
Client address: 181.14.16.182
Your address:
Server address: 181.14.16.170
Source port: 68 Destination port: 67
Source address: 181.141.16.182
Destination address: 255.255.255.255.

DHCPACK

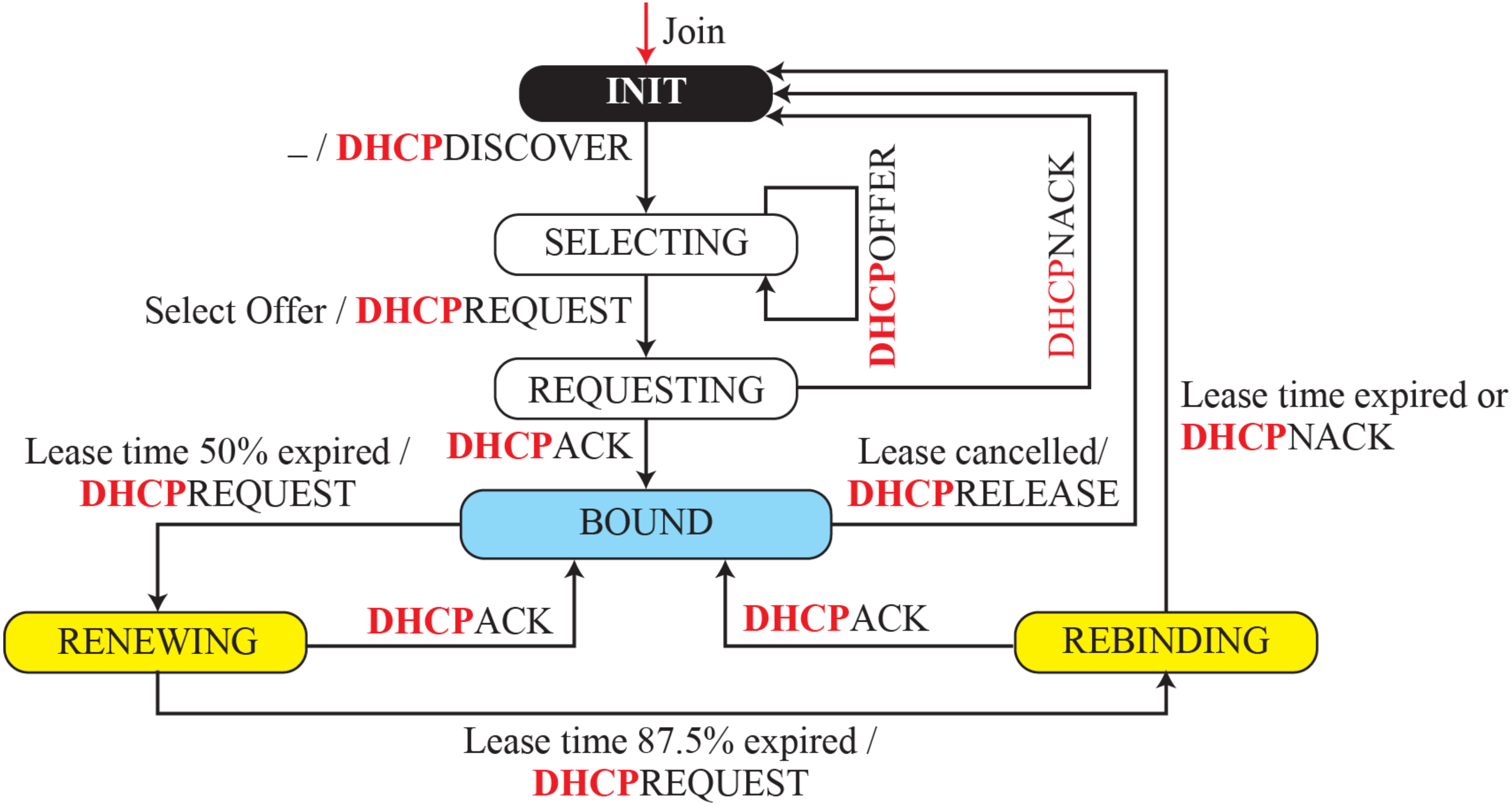
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67
Source address: 181.141.16.170
Destination address: 255.255.255.255.

Legend

Application
UDP
IP

Note:
Only partial information is given.

Figure 18.28: FSM for the DHCP client



18.4.5 NAT

In most situations, only a portion of computers in a small network need access to the Internet simultaneously. A technology that can provide the mapping between the private and universal addresses, and at the same time support virtual private networks, which we discuss in Chapter 32, is Network Address Translation (NAT). The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.

Figure 18.29: NAT

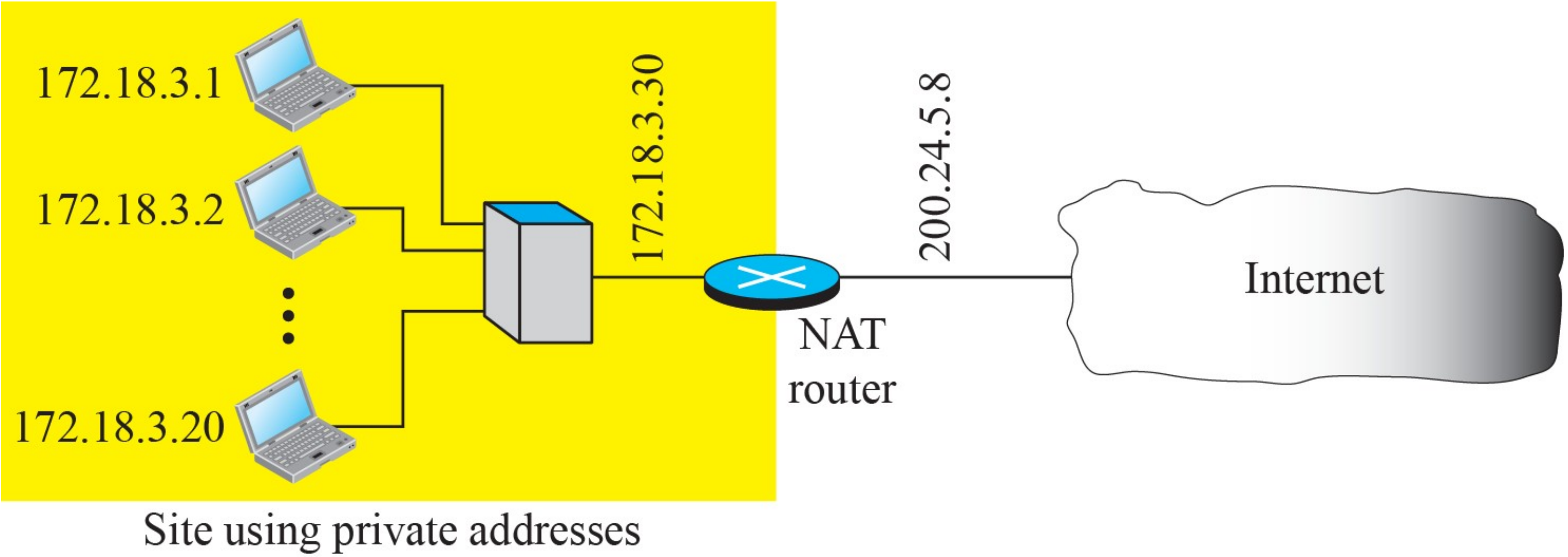


Figure 18.30: Address translation

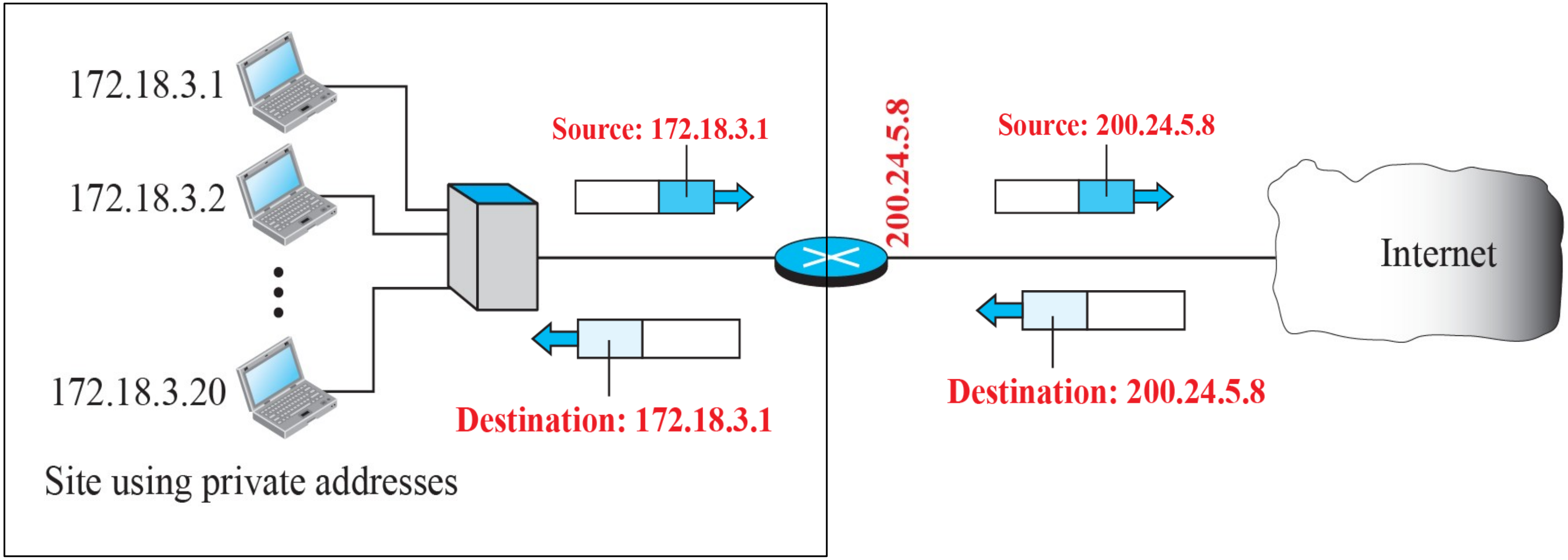
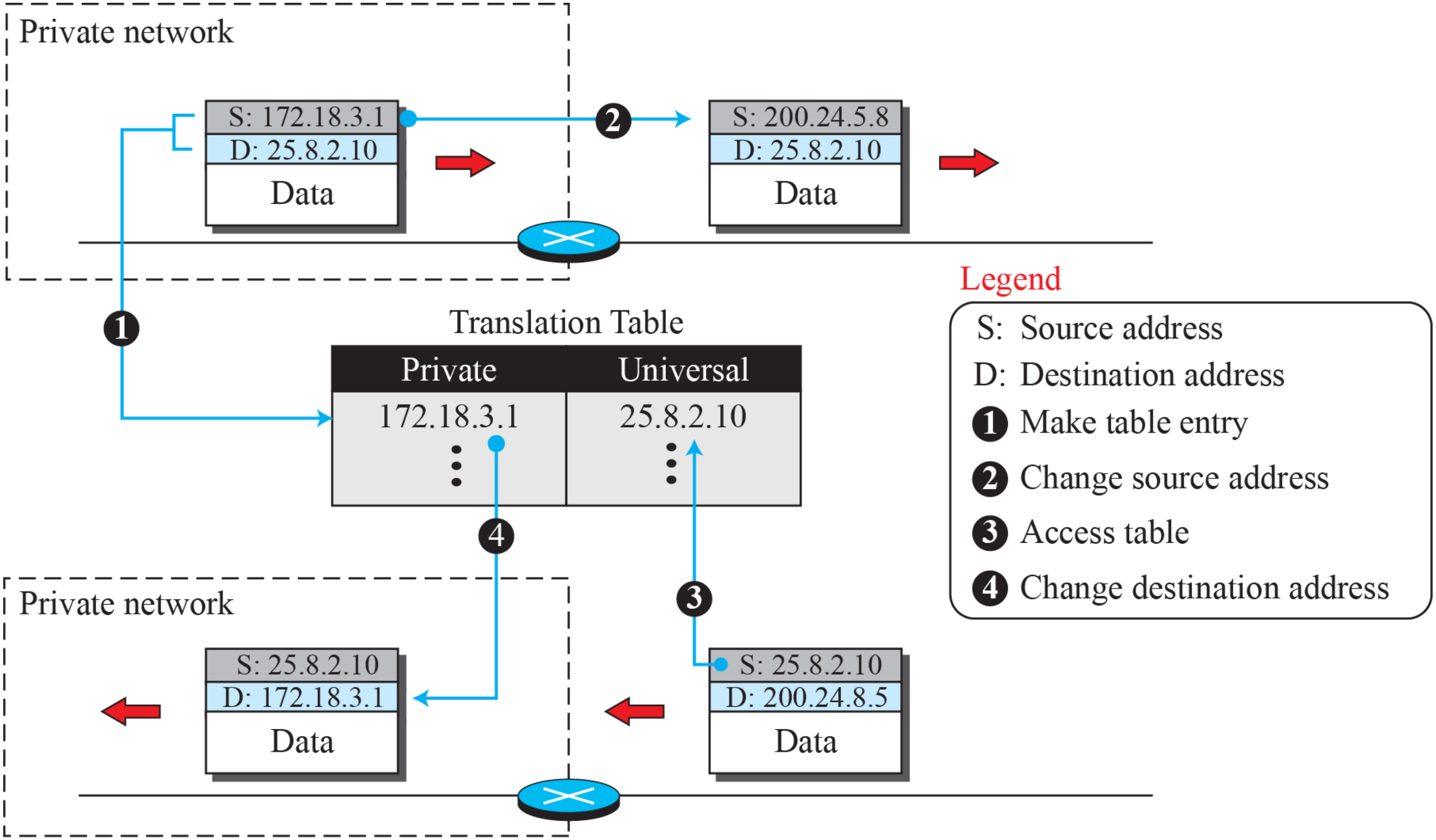


Figure 18.31: Translation



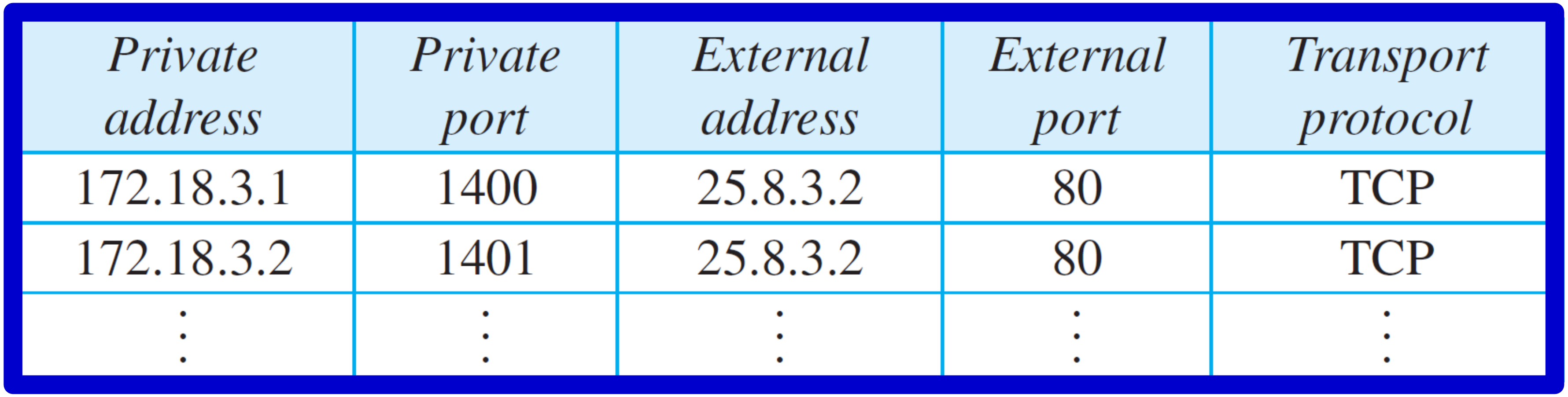
Translation Table

Private	Universal
172.18.3.1	25.8.2.10
⋮	⋮

Legend

S: Source address
D: Destination address
❶ Make table entry
❷ Change source address
❸ Access table
❹ Change destination address

Table 18.1: Five-column translation table



Private address	*Private port*	*External address*	*External port*	*Transport protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
⋮	⋮	⋮	⋮	⋮

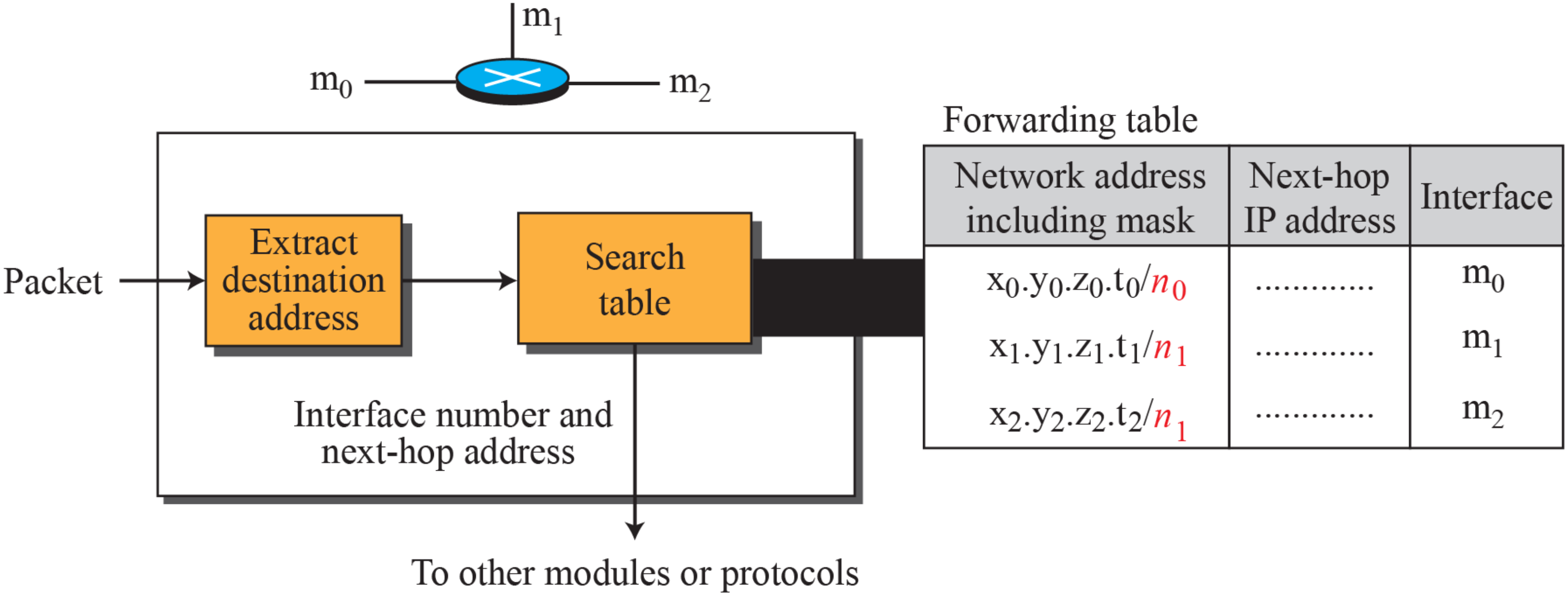
18-5 FORWARDING OF IP PACKETS

We discussed the concept of forwarding at the network layer earlier in this chapter. In this section, we extend the concept to include the role of IP addresses in forwarding. As we discussed before, forwarding means to place the packet in its route to its destination.

18.5.1 Destination Address Forwarding

We first discuss forwarding based on the destination address. This is a traditional approach, which is prevalent today. In this case, forwarding requires a host or a router to have a forwarding table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the next hop to deliver the packet to.

Figure 18.32: Simplified forwarding module in classless address



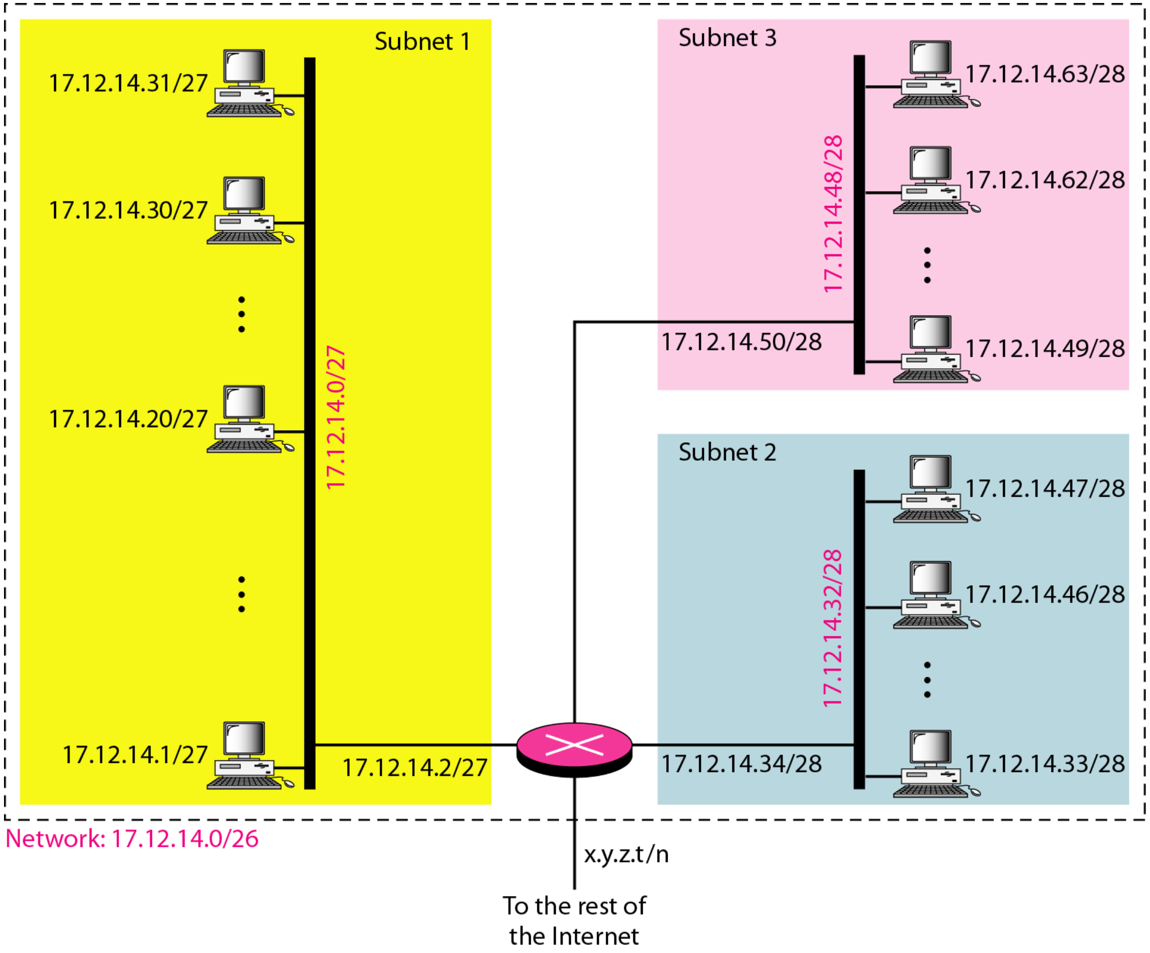
m_1
m_0
m_2
Packet
Extract destination address
Search table
Interface number and next-hop address
To other modules or protocols
Forwarding table

Network address including mask	Next-hop IP address	Interface
$x_0.y_0.z_0.t_0/n_0$	....	m_0
$x_1.y_1.z_1.t_1/n_1$	....	m_1
$x_2.y_2.z_2.t_2/n_1$	....	m_2

Note

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Configuration and addresses in a subnetted network



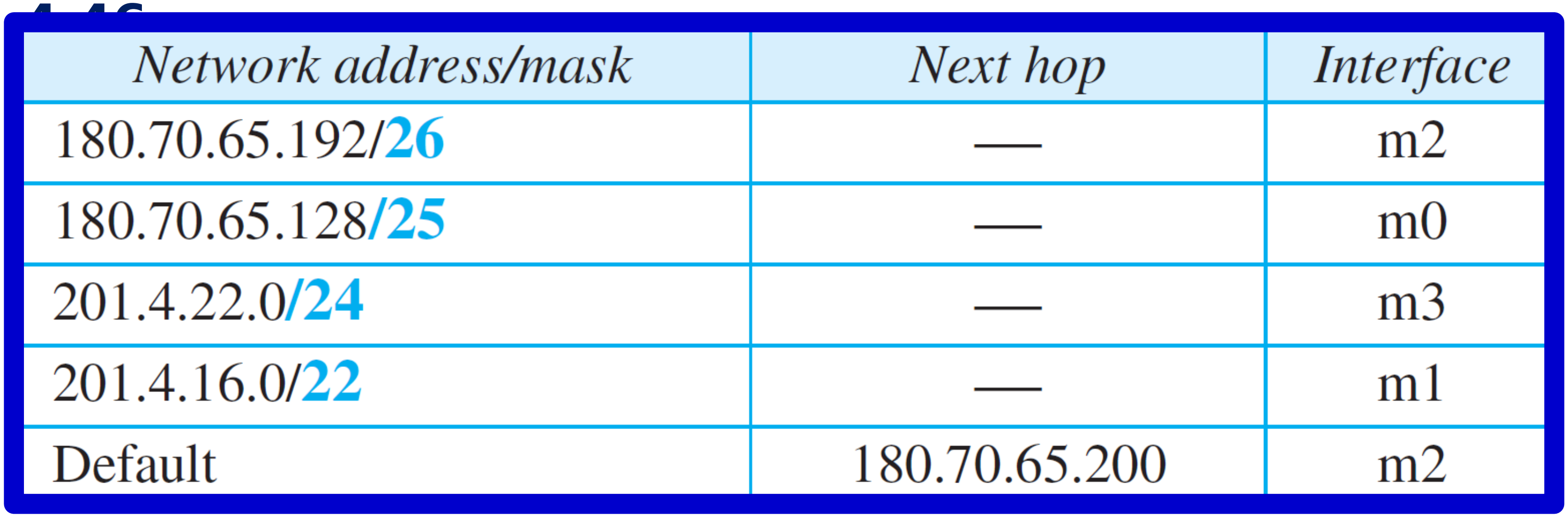
Example 18.7

Make a forwarding table for router R1 using the configuration in Figure 18.33.

Solution

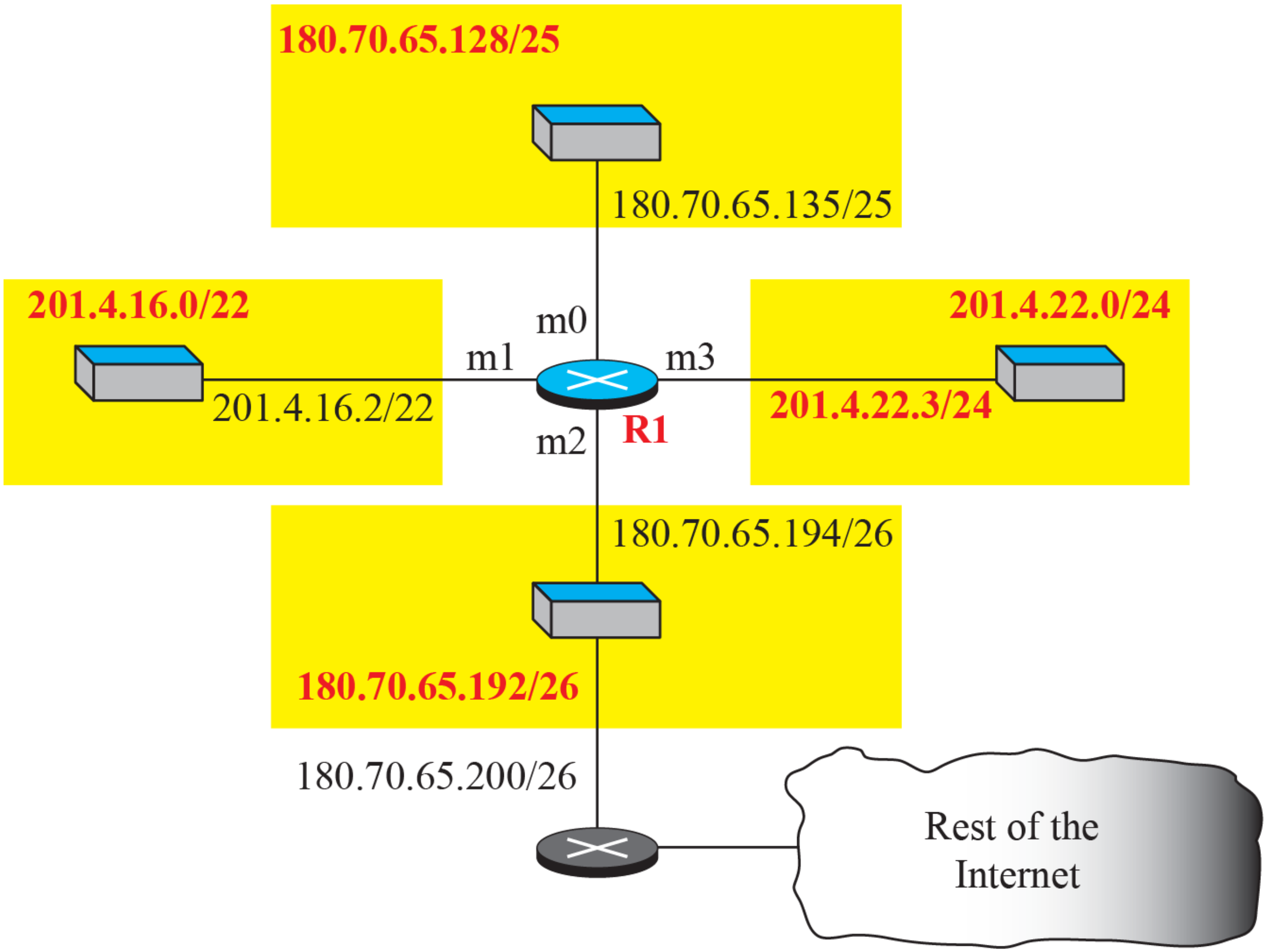
Table 18.2 shows the corresponding table.

Table 18.2: **Forwarding table for router R1 in Figure 4.46**



Network address/mask	*Next hop*	*Interface*
180.70.65.192/**26**	—	m2
180.70.65.128/**25**	—	m0
201.4.22.0/**24**	—	m3
201.4.16.0/**22**	—	m1
Default	180.70.65.200	m2

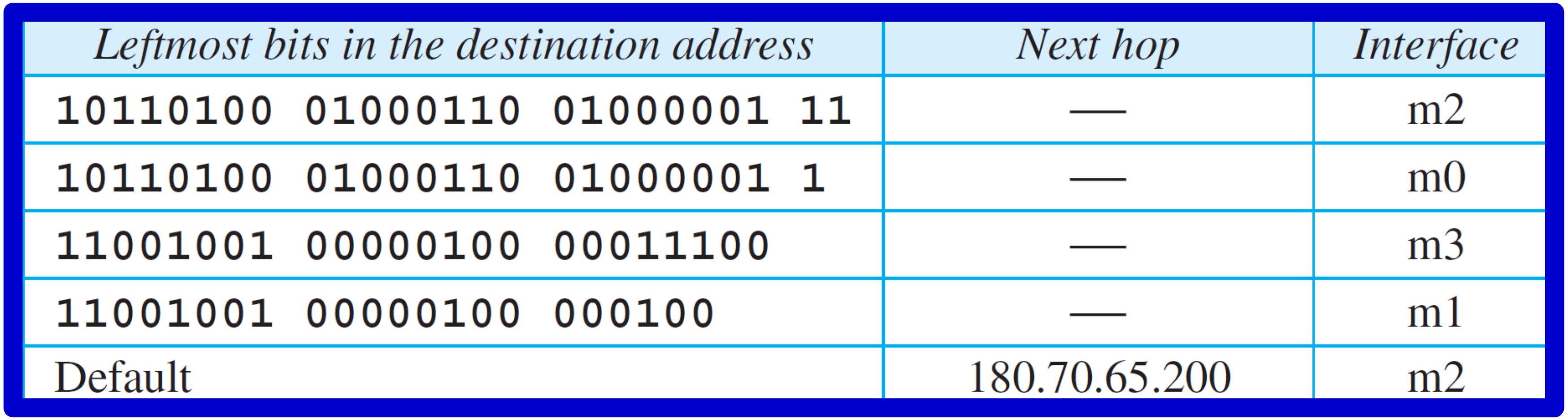
Figure 18.33: Configuration for Example 4.7



Example 18.8

Instead of Table 18.2, we can use Table 18.3, in which the network address/mask is given in bits.

Table 18.3: Forwarding table for router R1 using prefix bits



Leftmost bits in the destination address	*Next hop*	*Interface*
10110100 01000110 01000001 11	—	m2
10110100 01000110 01000001 1	—	m0
11001001 00000100 00011100	—	m3
11001001 00000100 000100	—	m1
Default	180.70.65.200	m2

When a packet arrives whose leftmost 26 bits in the destination address match the bits in the first row, the packet is sent out from interface m2. And so on.

Example 18.9

Show the forwarding process if a packet arrives at R1 in Figure 18.33 with the destination address 180.70.65.140.

Solution

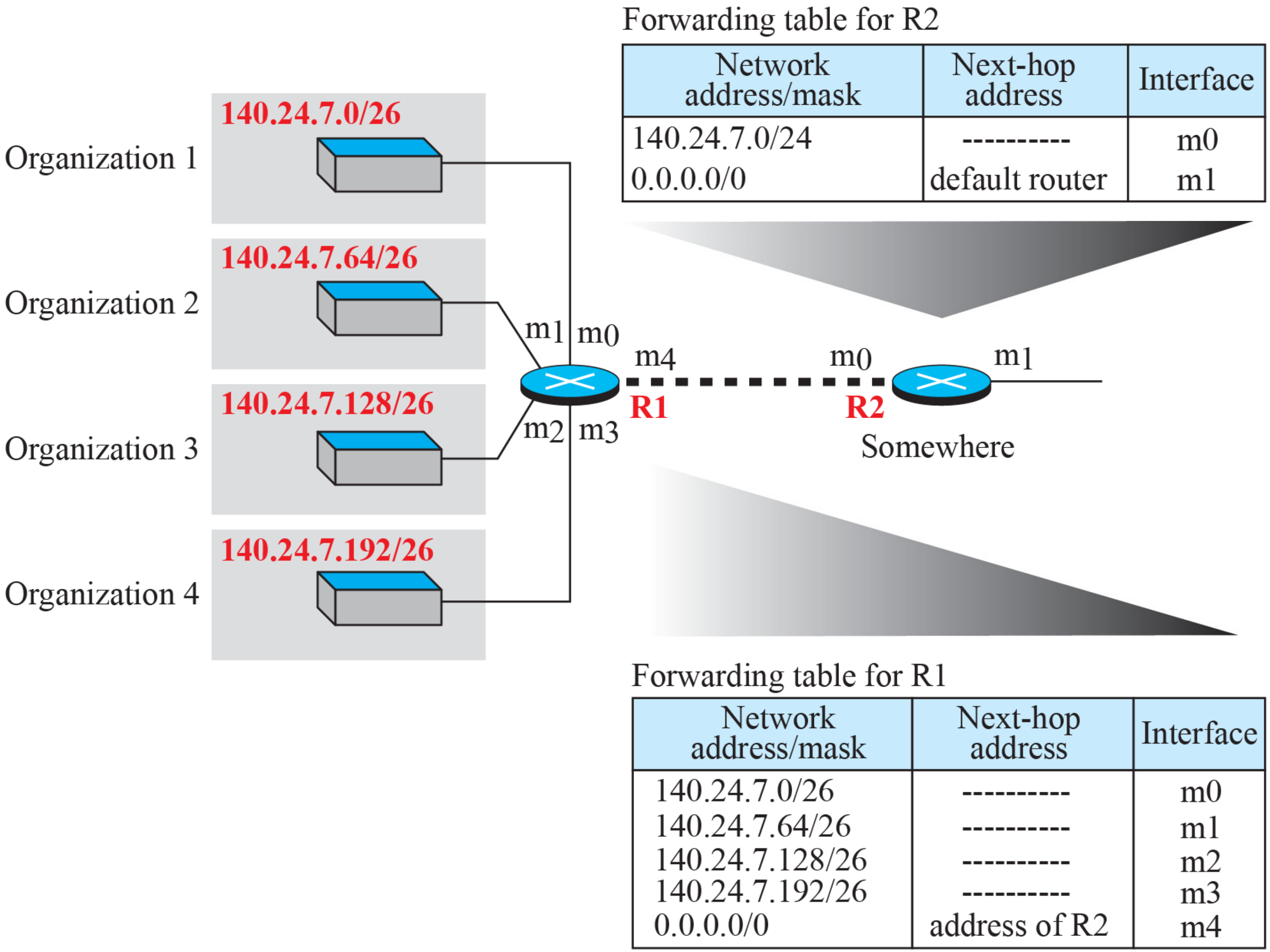
The router performs the following steps:

18. The first mask (/26) is applied to the destination address.

The result is 180.70.65.128, which does not match the corresponding network address.

2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are extracted for forwarding the packet (see Chapter 5).

Figure 18.34: Address aggregation



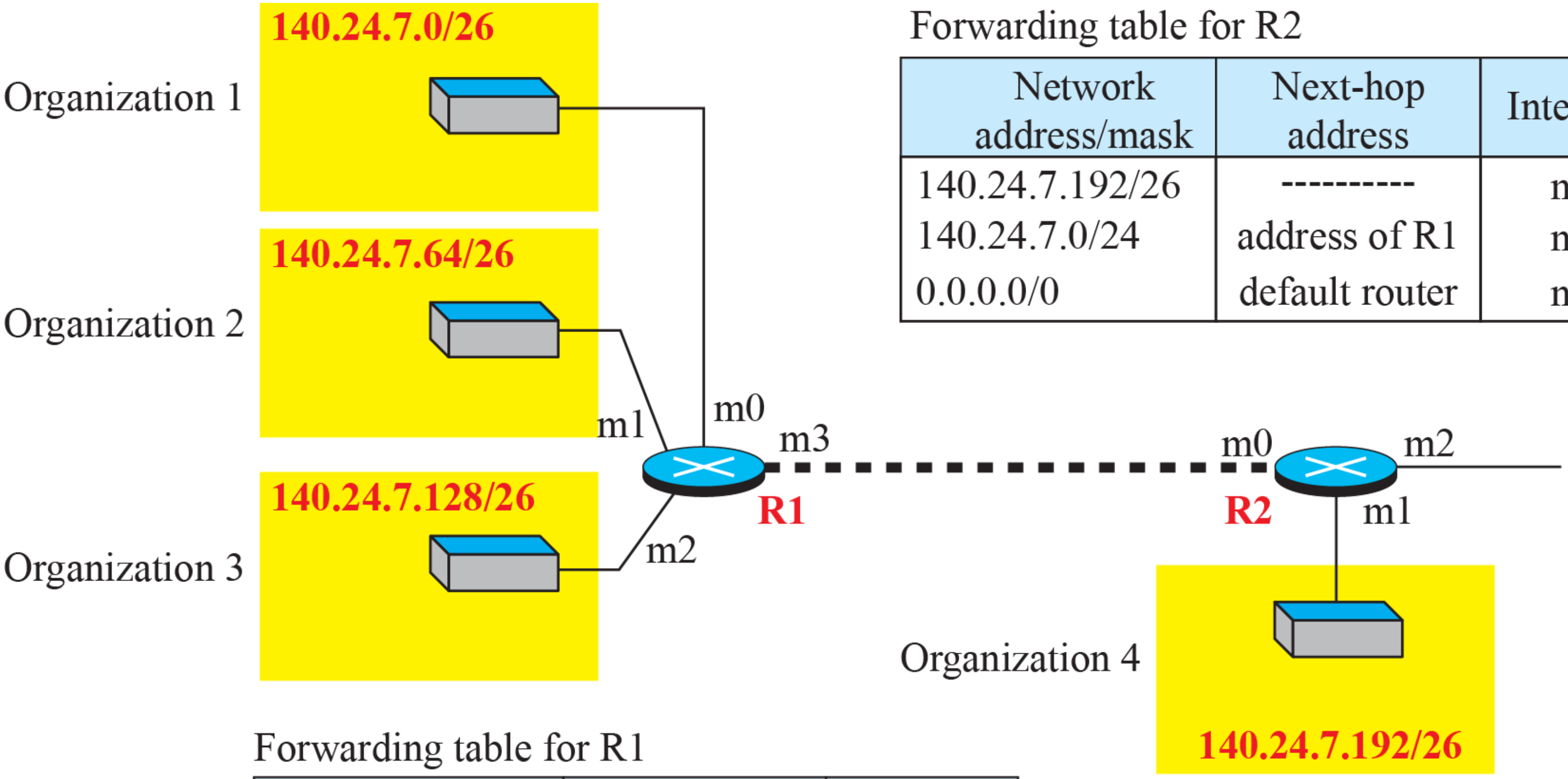
Forwarding table for R2

Network address/mask	Next-hop address	Interface
140.24.7.0/24	----------	m0
0.0.0.0/0	default router	m1

Forwarding table for R1

Network address/mask	Next-hop address	Interface
140.24.7.0/26	----------	m0
140.24.7.64/26	----------	m1
140.24.7.128/26	----------	m2
140.24.7.192/26	----------	m3
0.0.0.0/0	address of R2	m4

Figure 18.35: Longest mask matching



Forwarding table for R2

Network address/mask	Next-hop address	Interface
140.24.7.192/26	----------	m1
140.24.7.0/24	address of R1	m0
0.0.0.0/0	default router	m2

Forwarding table for R1

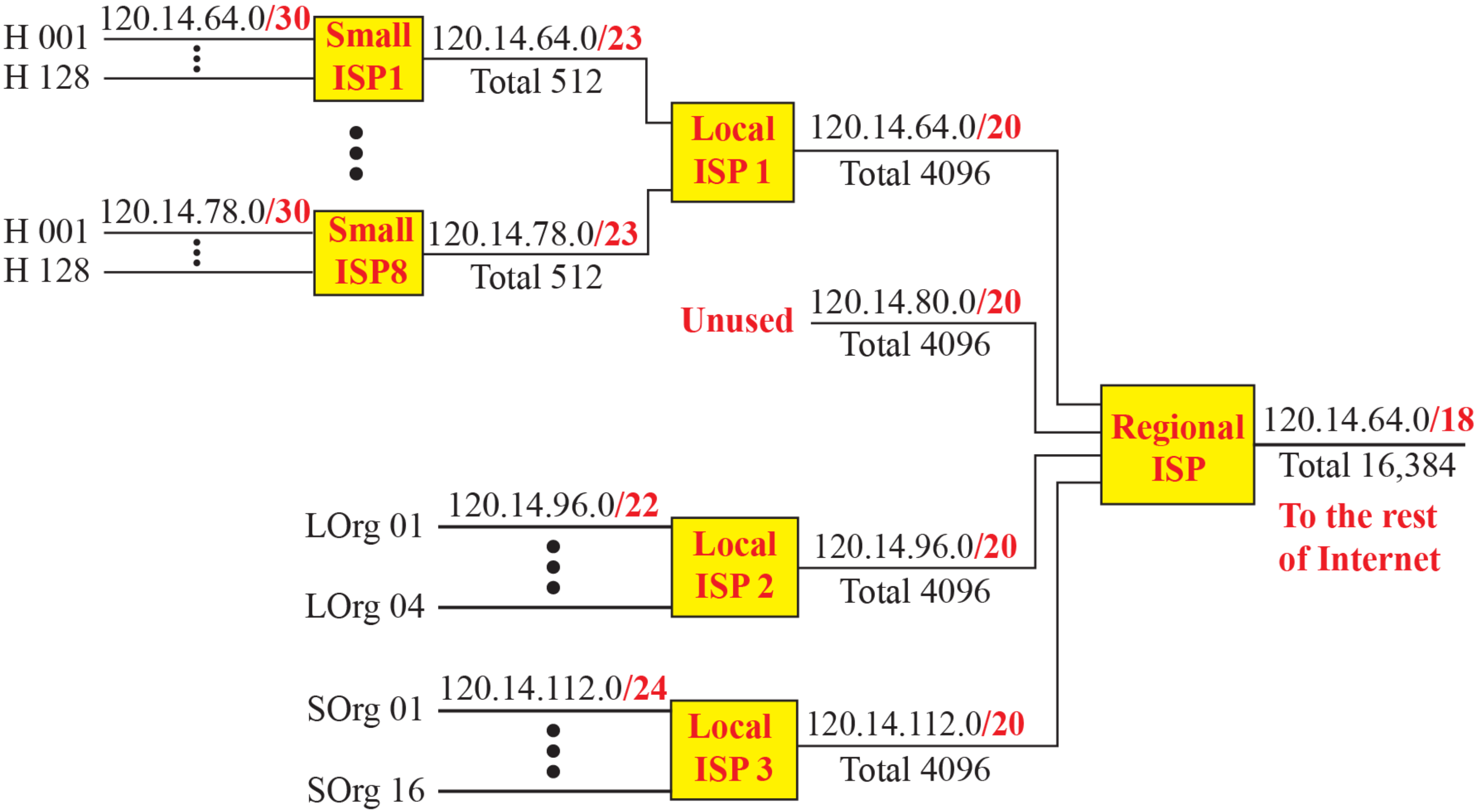
Network address/mask	Next-hop address	Interface
140.24.7.0/26	----------	m0
140.24.7.64/26	----------	m1
140.24.7.128/26	----------	m2
0.0.0.0/0	default router	m3

Example 18.10

As an example of hierarchical routing, let us consider Figure 18.36. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into 4 subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs, the second subblock is reserved for future use. Note that the mask for each block is **/20** because the original block with mask /18 is divided into 4 blocks.

The figure also shows how local and small ISPs have assigned addresses.

Figure 18.35: Hierarchical routing with ISPs


H 001
H 128
120.14.64.0/30
Small ISP1
120.14.64.0/23
Total 512
H 001
H 128
120.14.78.0/30
Small ISP8
120.14.78.0/23
Total 512
Local ISP 1
120.14.64.0/20
Total 4096
Unused
120.14.80.0/20
Total 4096
Regional ISP
120.14.64.0/18
Total 16,384
To the rest of Internet
LOrg 01
LOrg 04
120.14.96.0/22
Local ISP 2
120.14.96.0/20
Total 4096
SOrg 01
SOrg 16
120.14.112.0/24
Local ISP 3
120.14.112.0/20
Total 4096