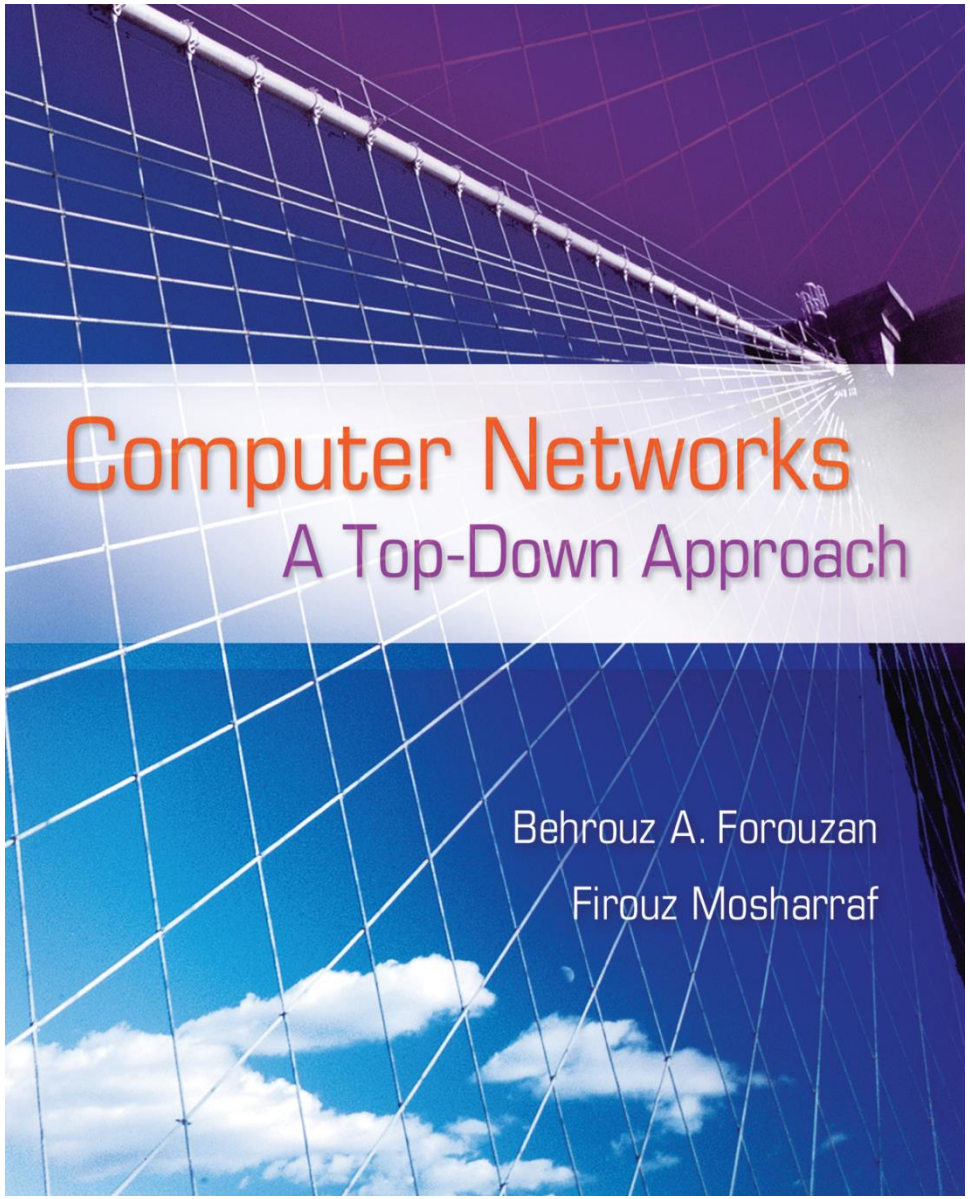
Chapter 4

Network Layer



Chapter 4: Outline

4.1 INTRODUCTION

4.2 NETWORK-LAYER PROTOCOLS

4.3 INTRODUCTION TO UNICAST ROUTING

4.4 INTRODUCTION TO MULTICAST ROUTING

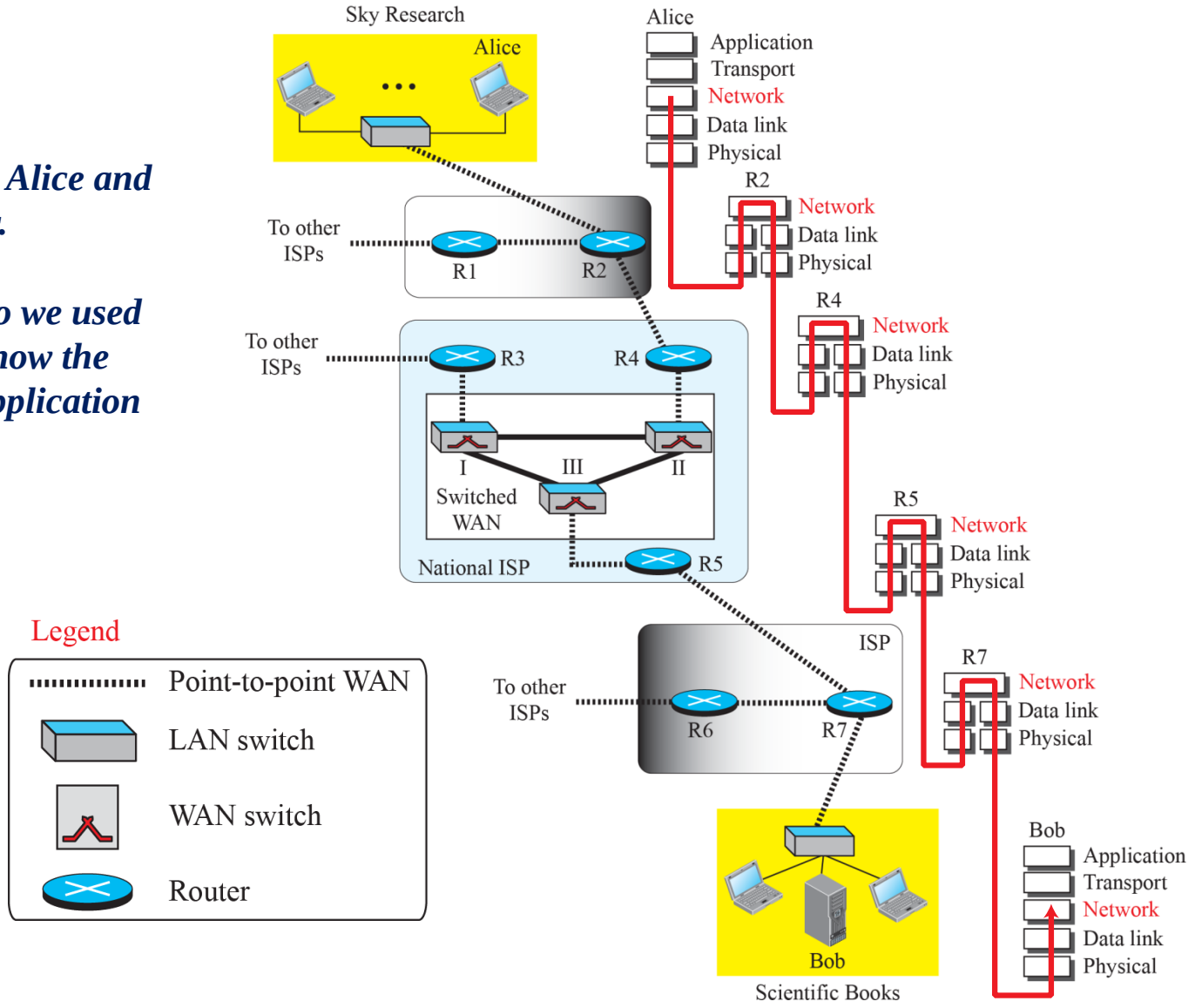
4.5 NEXT GENERATION IP

Chapter 4: Objective

- *We first discuss services that can be provided at the network layer: packetizing, routing, and forwarding.*
- *We then discuss the network layer at the TCP/IP suite: IPv4 and ICMPv4. We also discuss IPv4 addressing and related issues.*
- *We then concentrate on the unicast routing and unicast routing protocols.*
- *We then move to multicasting and multicast routing protocols protocol.*
- *We finally discuss the new generation of network-layer protocols, IPv6 and ICMPv6.*

Figure 4.1: Introduction: Communication at the network layer

- *Figure 4.1 shows the communication between Alice and Bob at the network layer.*
- *This is the same scenario we used in Chapters 2 and 3 to show the communication at the application and the transport layers, respectively.*



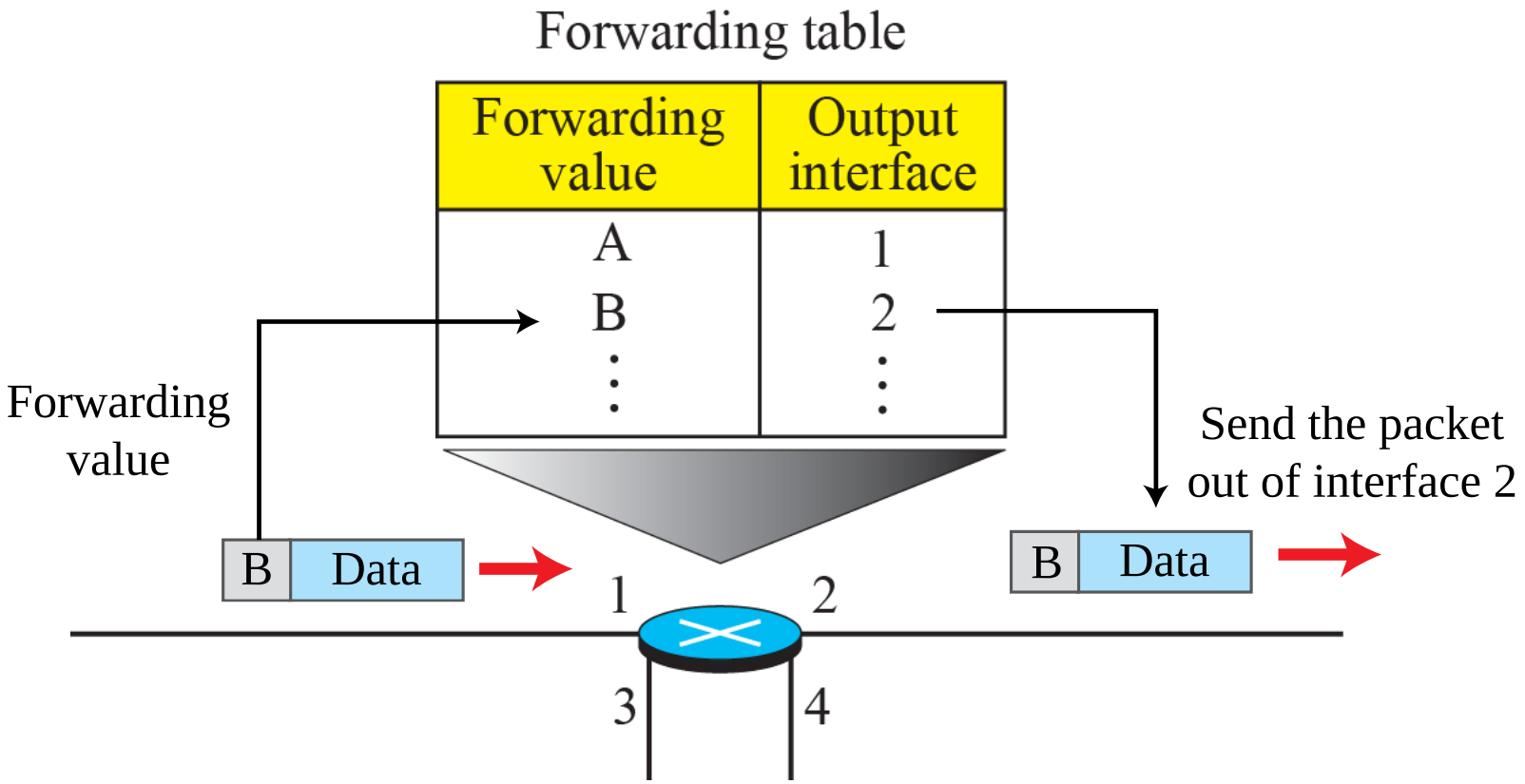
4.1.1 Network-Layer Services

- *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.*

  - ❑ *Packetizing*
  - ❑ *Routing*
  - ❑ *Forwarding*

  - ❑ *Error & Flow Control*
  - ❑ *Congestion Control*
  - ❑ *Security*

Figure 4.2: Forwarding process



4.1.2 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.*

  - ❑ *Datagram Approach*
  - ❑ *Virtual-Circuit Approach*
    - ❖ *Setup Phase*
    - ❖ *Data-Transfer Phase*
    - ❖ *Teardown Phase*

Figure 4.3: A connectionless packet-switched network

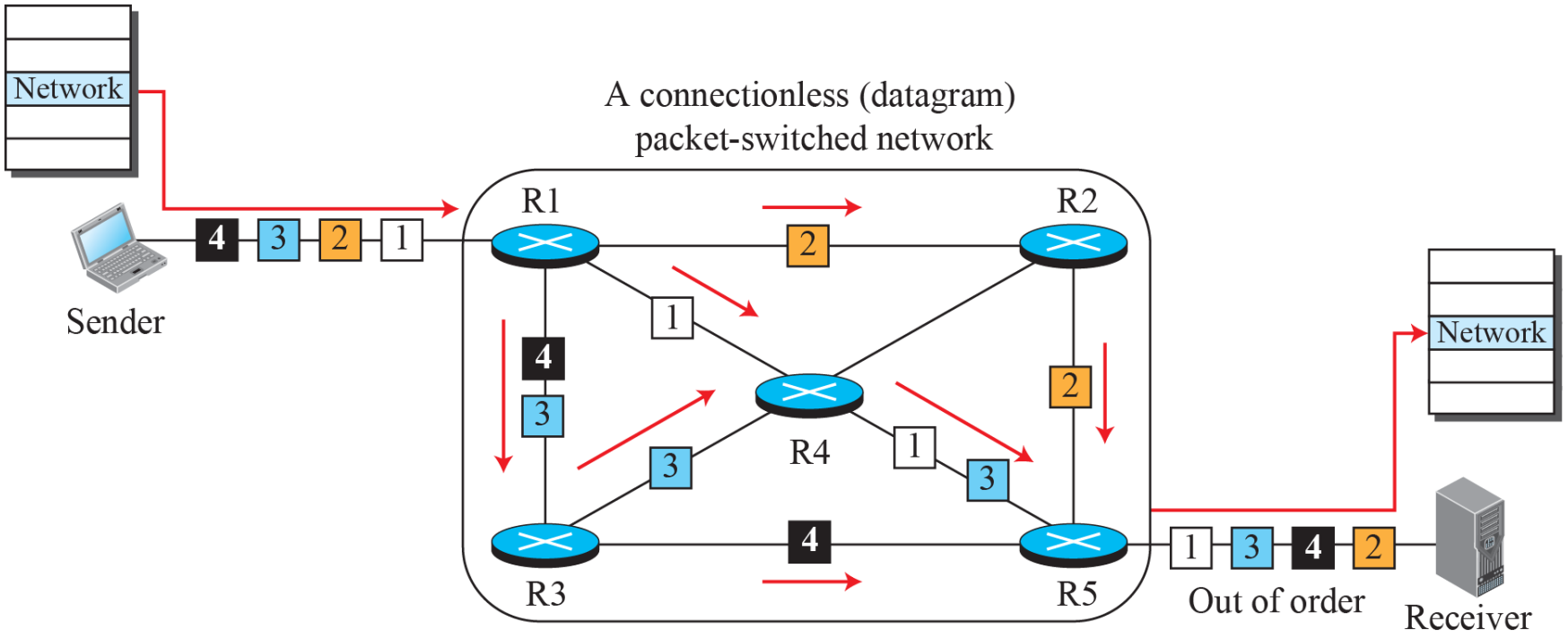


Figure 4.4: Forwarding process in a router when used in a connectionless network

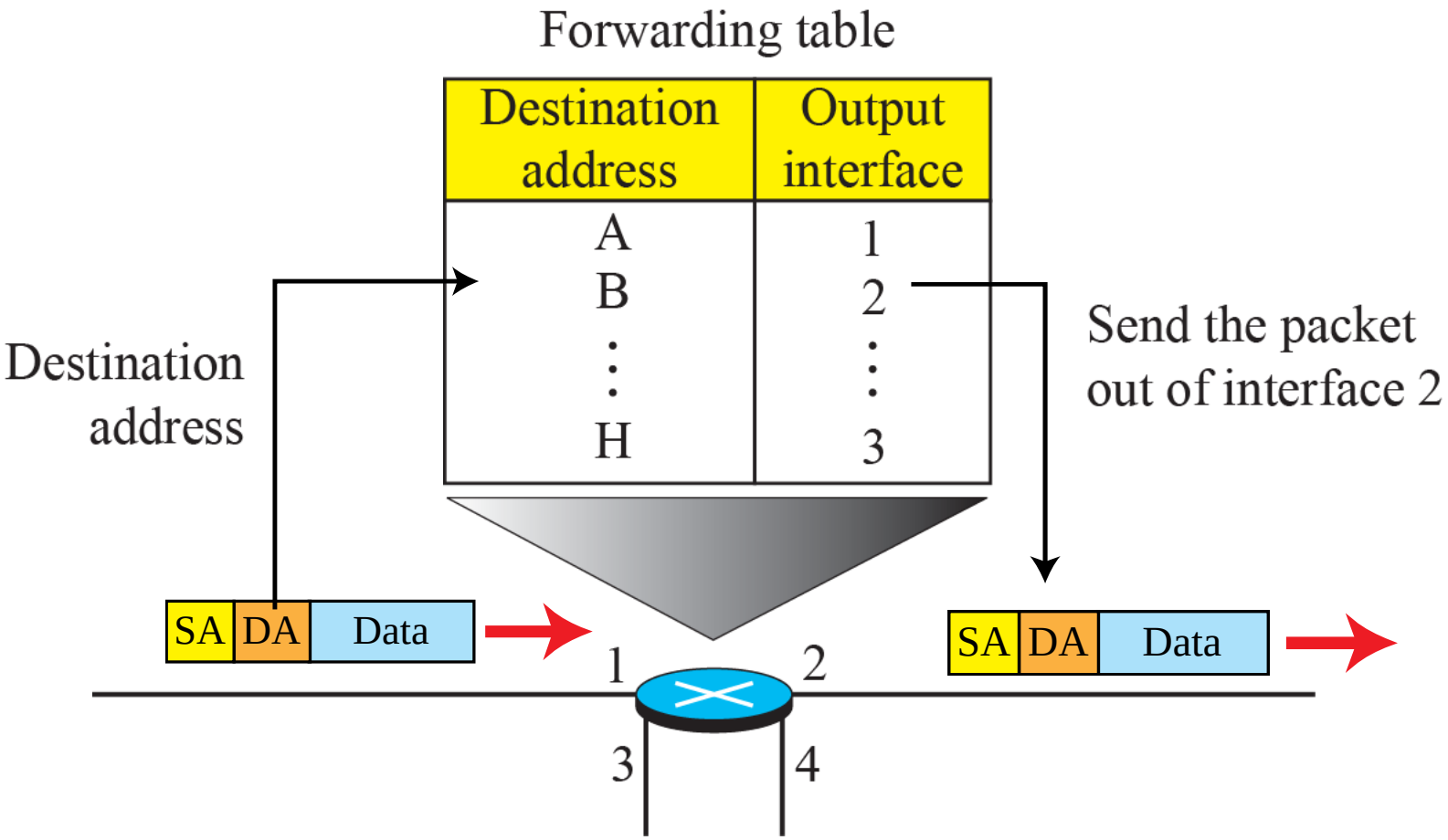


Figure 4.5: A virtual-circuit packet-switched network

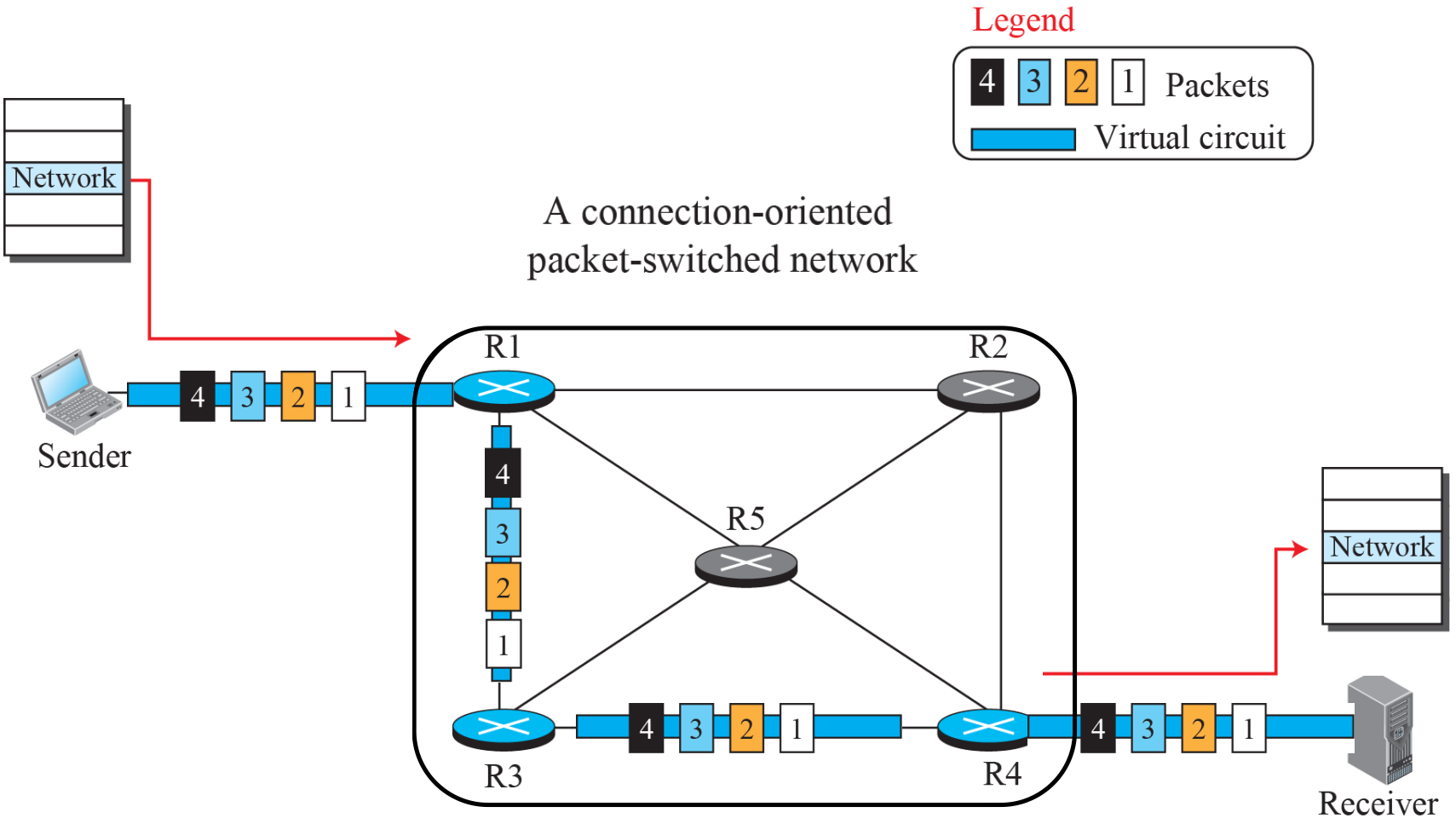


Figure 4.6: Forwarding process in a router when used in a virtual circuit network

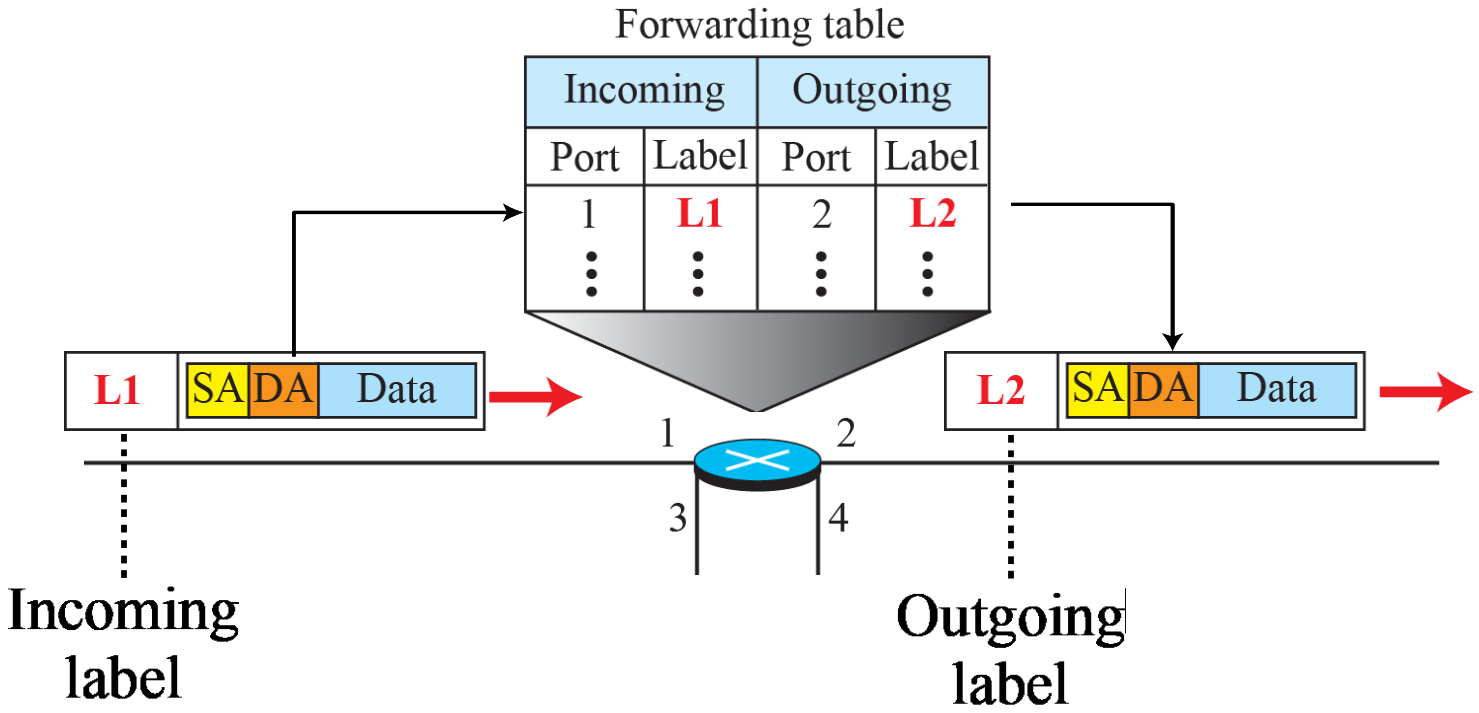


Figure 4.7: Sending request packet in a virtual-circuit network

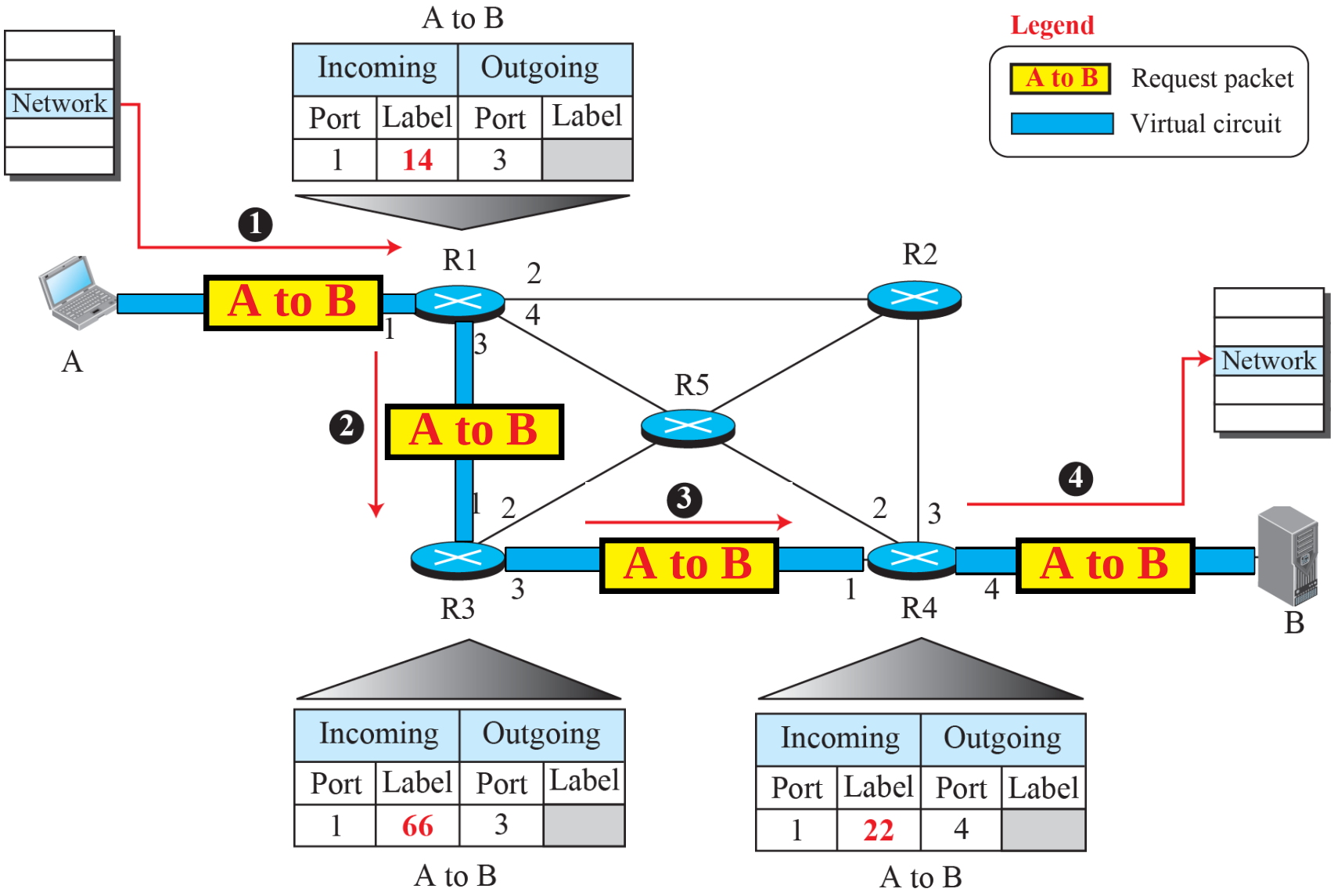


Figure 4.8: Sending acknowledgments in a virtual-circuit network

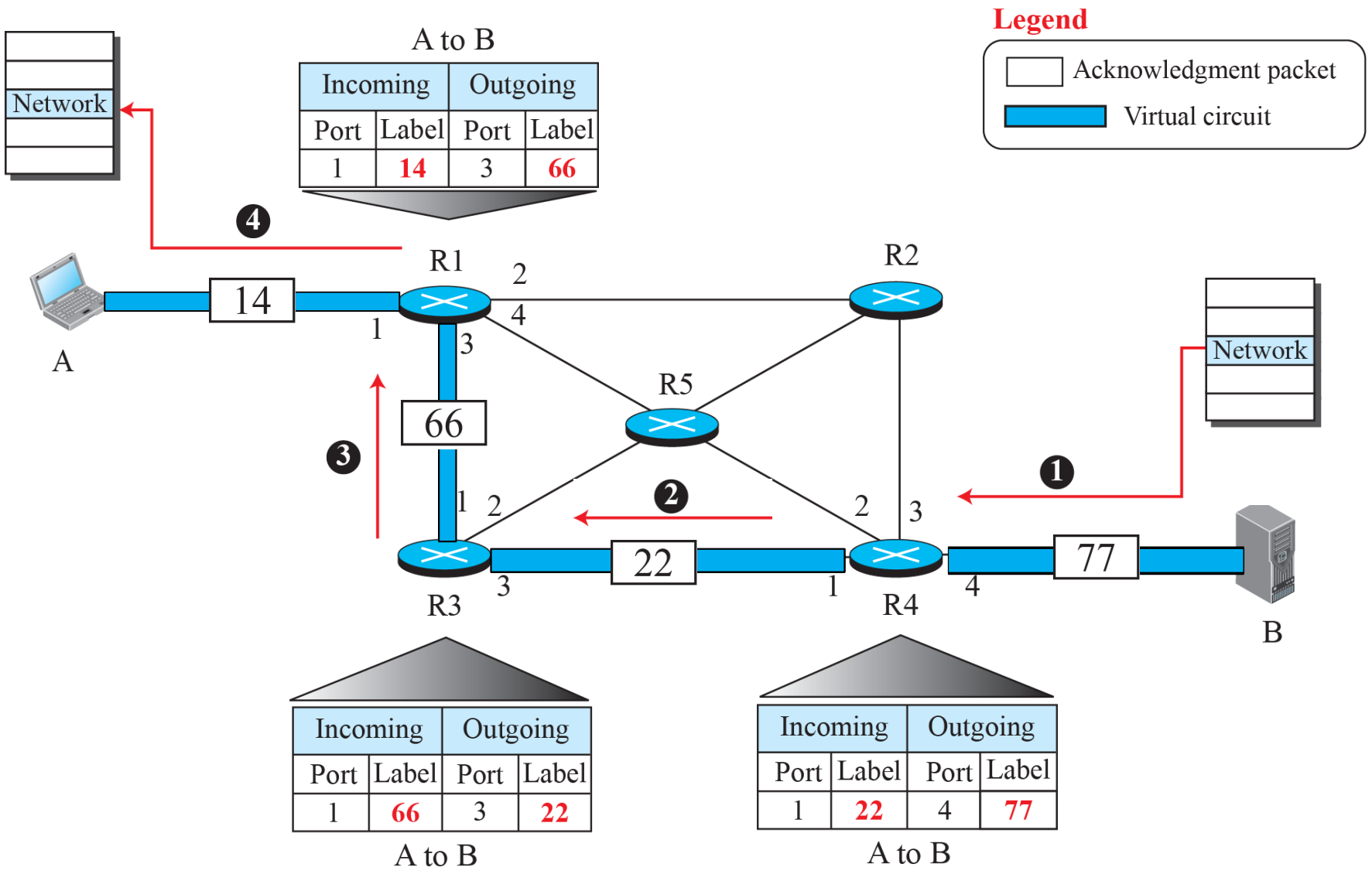
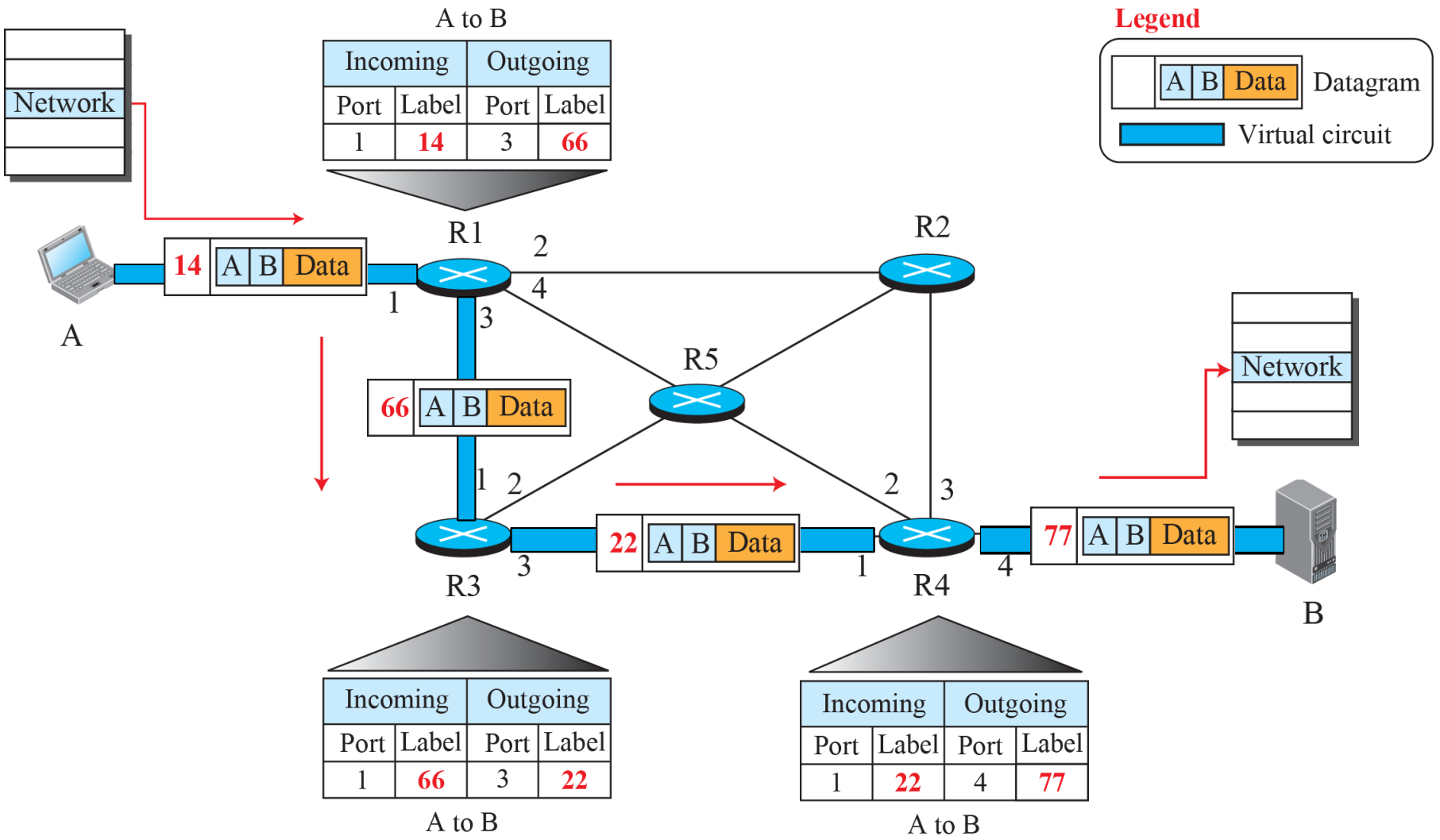


Figure 4.8: Sending acknowledgments in a virtual-circuit network



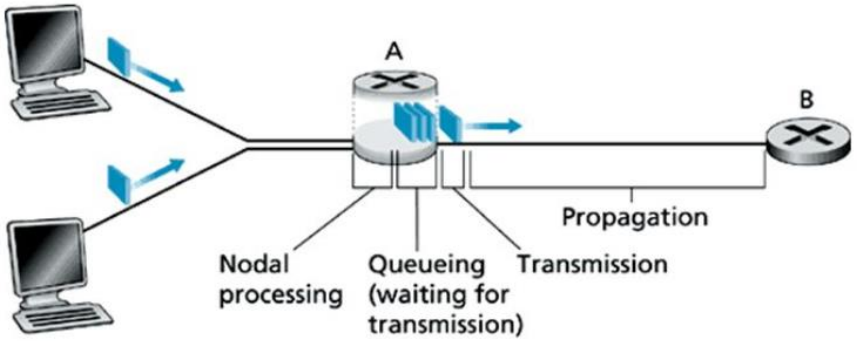
4.1.3 Network-Layer Performance

- *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*
- *We first define these three terms in a packet-switched network before we discuss their effects on performance.*

4.1.3 (continued)

Delay

- *Queuing Delay:* time waiting for its turn at output link
- *Processing/Nodal Delay:* Check for packet error & routing decision
- *Transmission Delay:* time to pump the packet onto a link at link speed
- *Propagation Delay:* router/node–to-router/node propagation



Total Delay=sum of all the delays-above

Throughput

Packet Loss

Figure 4.10: Throughput in a path with three links in a series

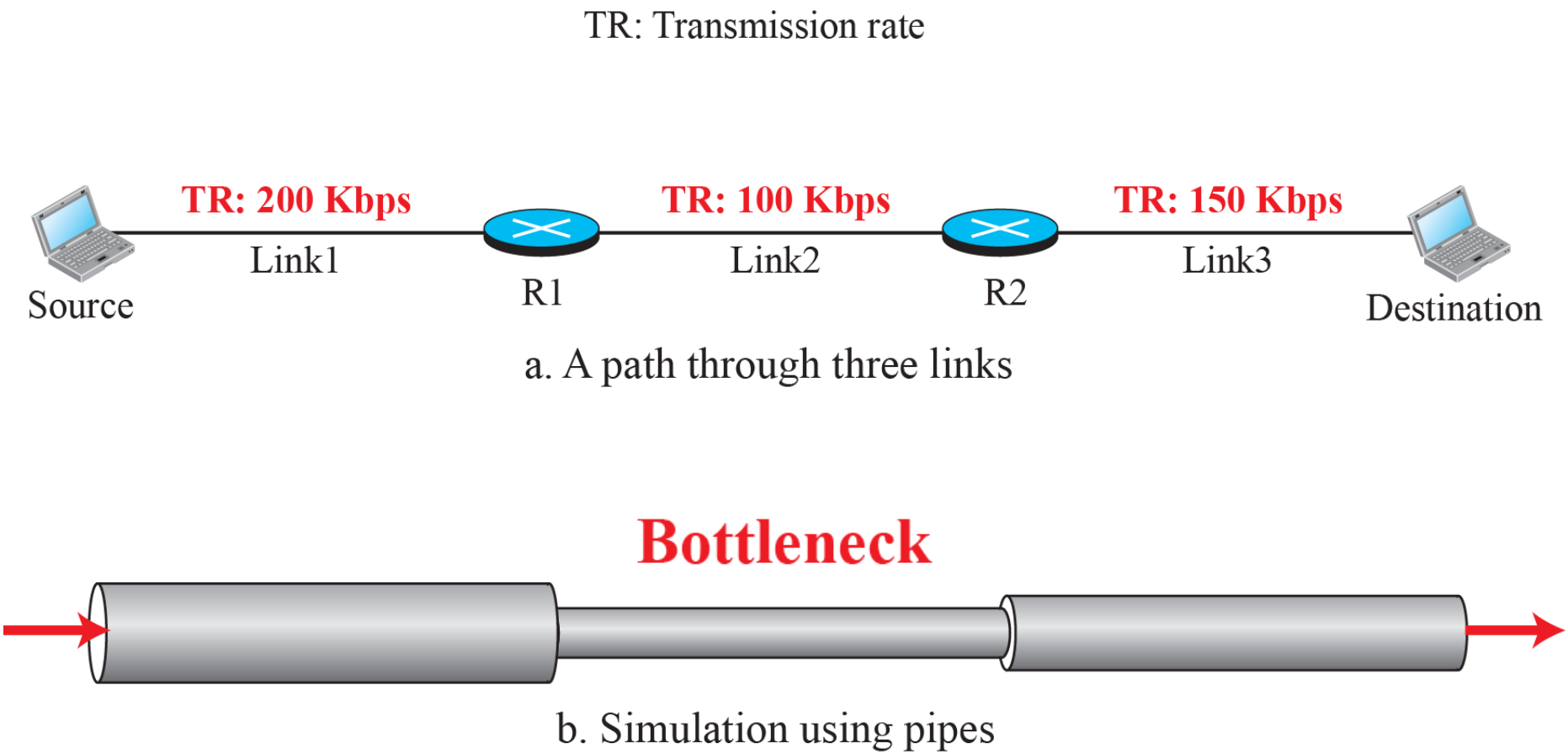
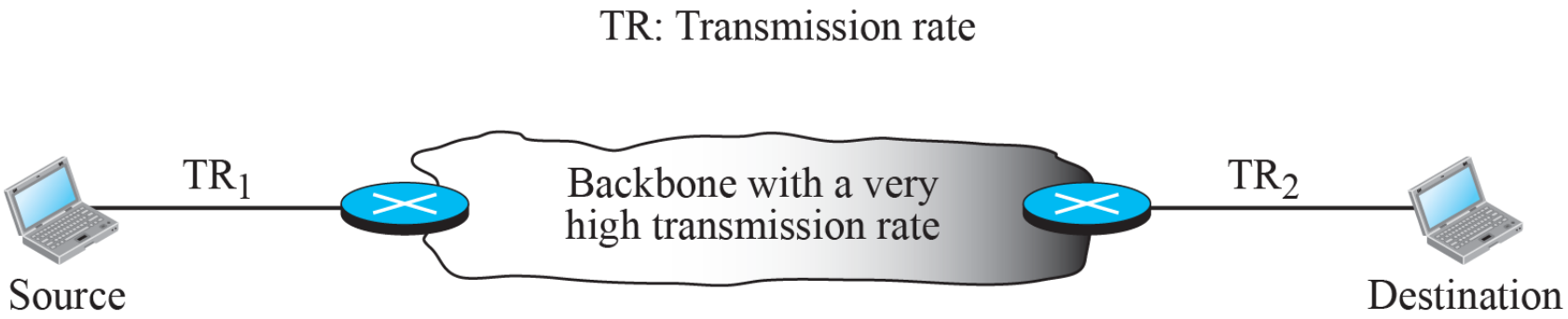


Figure 4.11: A path through the Internet backbone



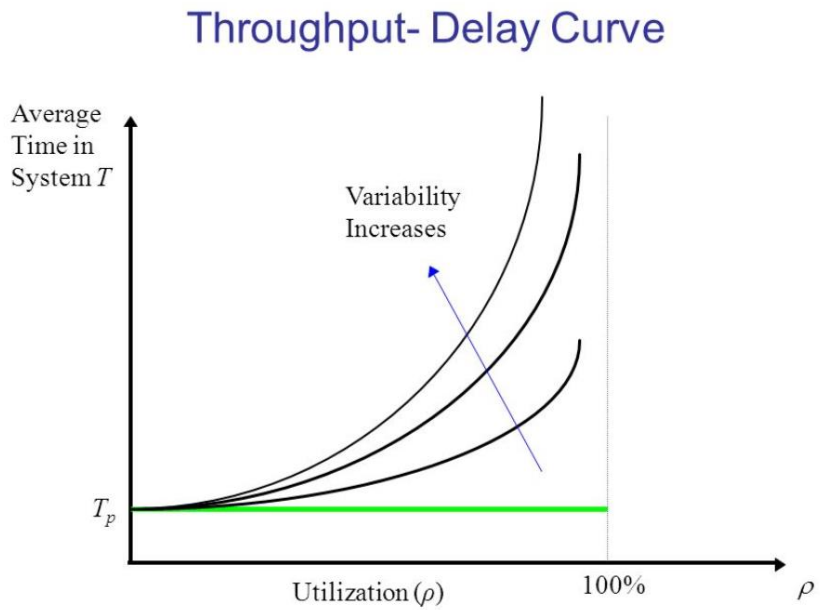
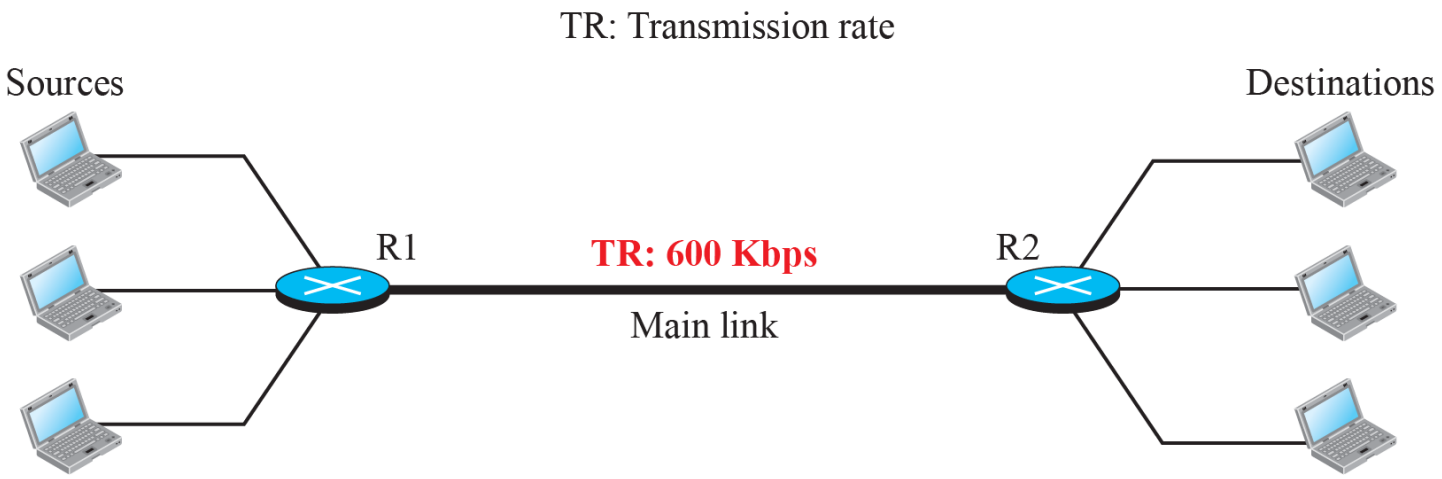
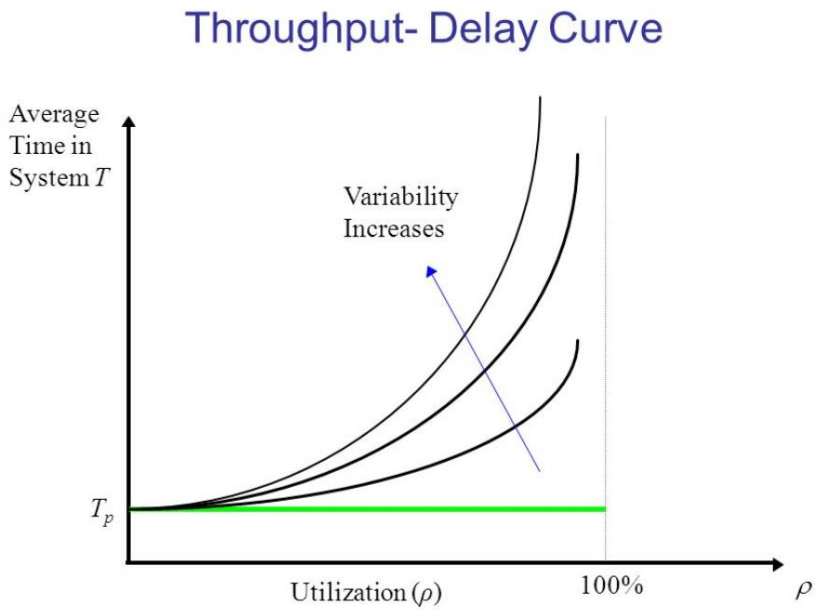


Figure 4.12: Effect of throughput in shared links





4.1.4 Network-Layer Congestions

- *In the previous lecture, we discussed 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 it previously.*

4.1.4 (continued)

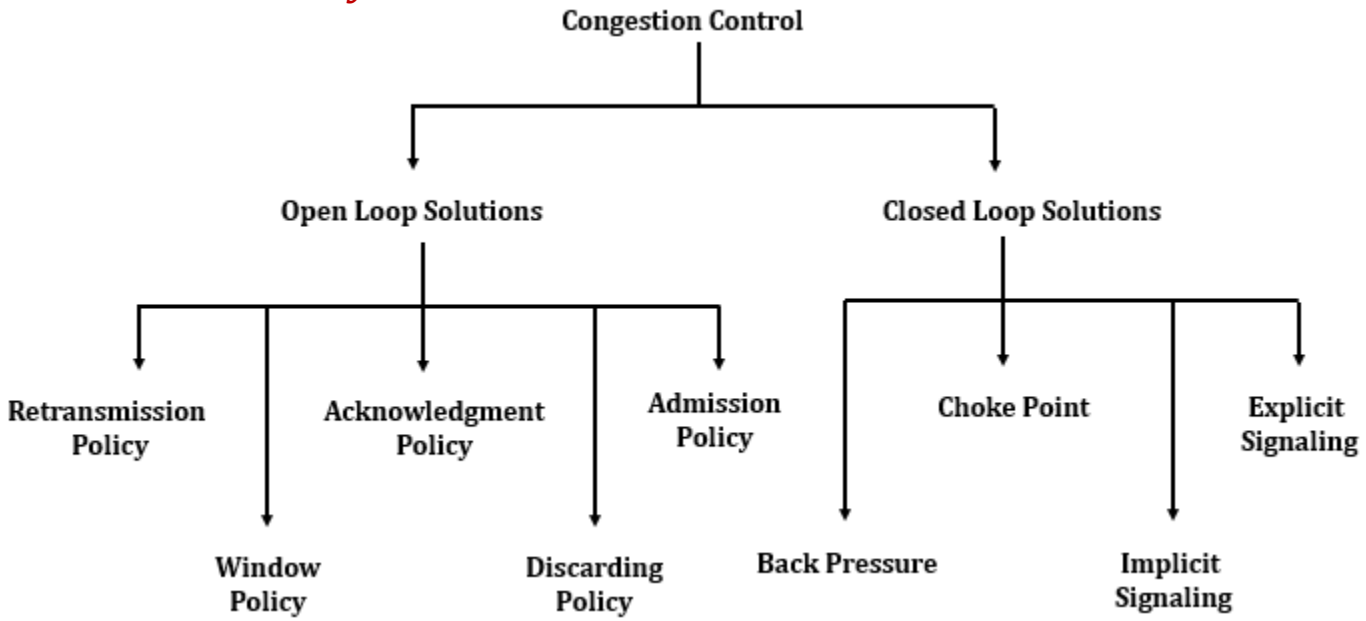
❑ Congestion Control

❖ *Open-Loop Congestion Control (Prevention)*

- *Retransmission Policy*
- *Window Policy*
- *Acknowledgment Policy*
- *Discarding Policy*
- *Admission Policy*

❖ *Closed-Loop Congestion Control* (removal)

- *Backpressure*
- *Choke Packet*
- *Implicit Signaling*
- *Explicit Signaling*



Open Loop Congestion Control:

• In this method, policies are used to prevent the congestion ***before it happens.***
• Congestion control is handled either by the source or by the destination.
• The various methods used for open loop congestion control are:

- ***Retransmission Policy:*** *The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted. The retransmission policy and the retransmission timers need to be designed to optimize efficiency*
- ***Window Policy****: To implement window policy, selective reject window method is used for congestion control. Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this duplication may make congestion worse. Selective reject method sends only the specific lost or damaged packets.*
- ***Acknowledgement Policy****: positive and negative acknowledge approaches can be used,*
- ***Discarding Policy****: A router may discard less sensitive packets when congestion is likely to happen.*
- ***Admission Policy:*** *An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.*

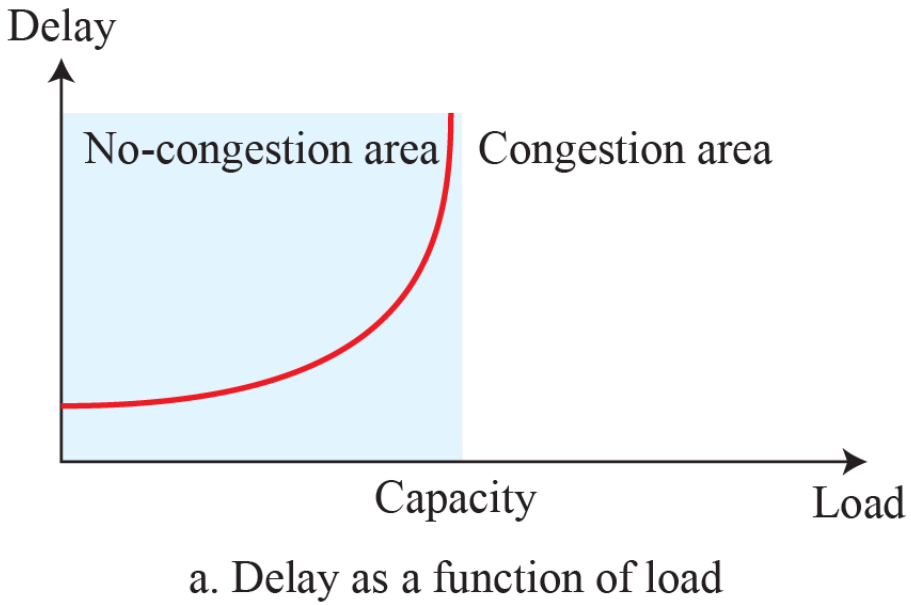
Closed Loop Congestion Control :

• Closed loop congestion control mechanisms try to remove the congestion ***after it happens.***

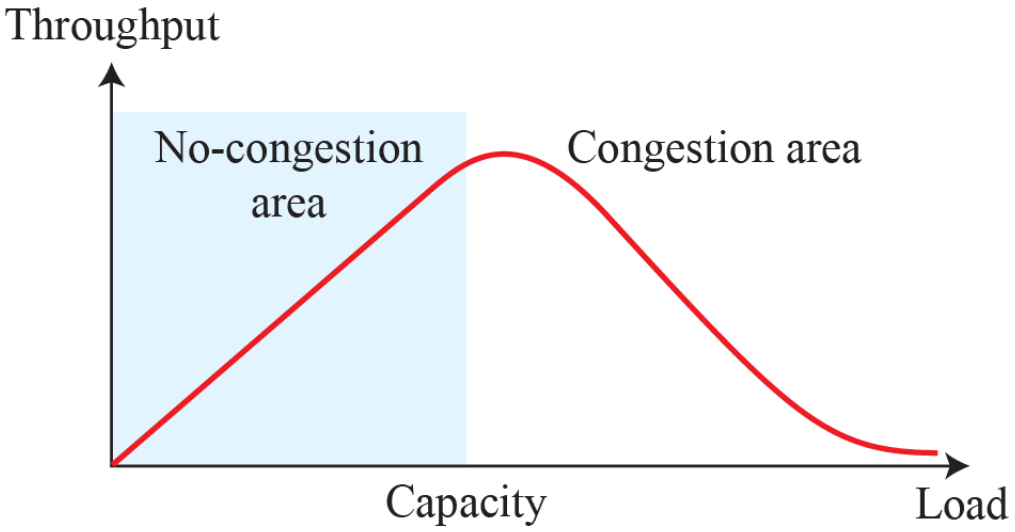
The various methods used for closed loop congestion control are:

- ***Backpressure:*** *Backpressure is a node-to-node congestion control that starts with a node and propagates, in the* ***opposite*** *direction of data flow.*
- ***Choke Packet:*** *In this method of congestion control, congested router or node* ***sends a special type of packet called choke packet*** *to the source to inform it about the congestion.*
- ***Implicit Signalling:*** *In implicit signalling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network when it does not receive any acknowledgment. This type of congestion control policy is used by TCP.*
- ***Explicit Signalling:*** *In this method, the congested nodes explicitly send a signal to the source or destination to inform about the congestion. Explicit signalling can occur in either the forward direction or the backward direction .*

Figure 4.13: Packet delay and throughput as functions of load



a. Delay as a function of load



b. Throughput as a function of load

Figure 4.14: Backpressure method for alleviating congestion

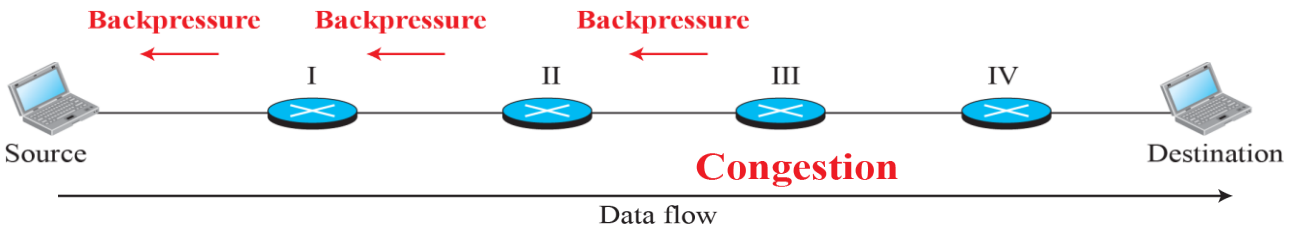
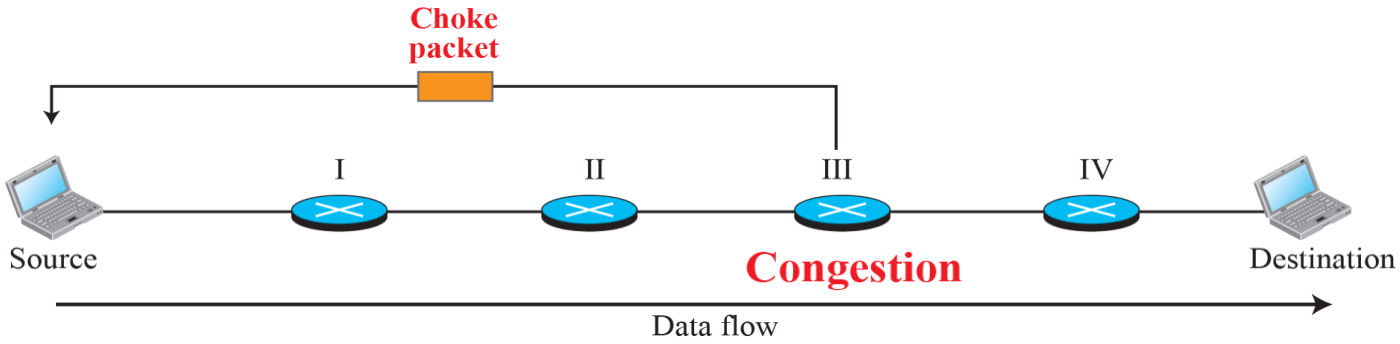


Figure 4.15: Choke packet



4.1.5 Structure of A Router

*Layer-3 (Network Layer Device - **forwarding** and **routing***

- *We represented a router as a black box*
- *That accepts **incoming** packets from one of the **input ports** (interfaces)*
- *Uses a **forwarding table** to find the **output port** from which the packet departs, and sends the packet from this output port.*
- *In this section we open the black box and look inside.*
- *However, our discussion **won't be very detailed**; entire books have been written about routers. We just give an overview.*

4.1.5 (continued)

- **Components**
  - Input Ports
  - Output Ports
  - Routing Processor
  - Switching Fabrics
    - Crossbar Switch
    - Banyan Switch
    - Batcher-Banyan Switch

Figure 4.16: Router components

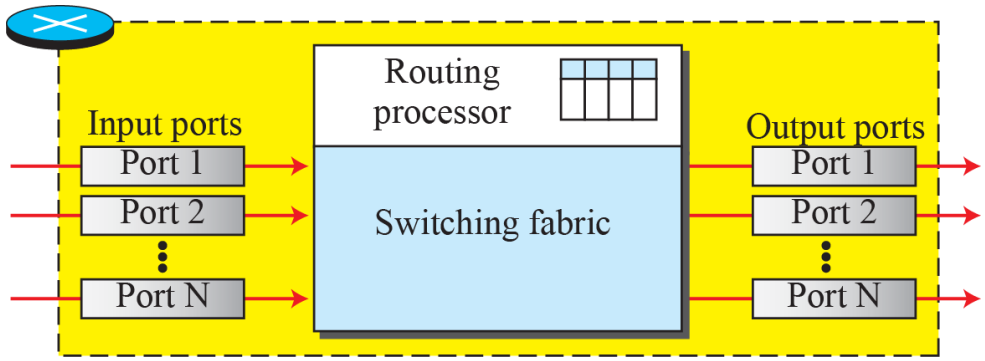


Figure 4.17: Input port

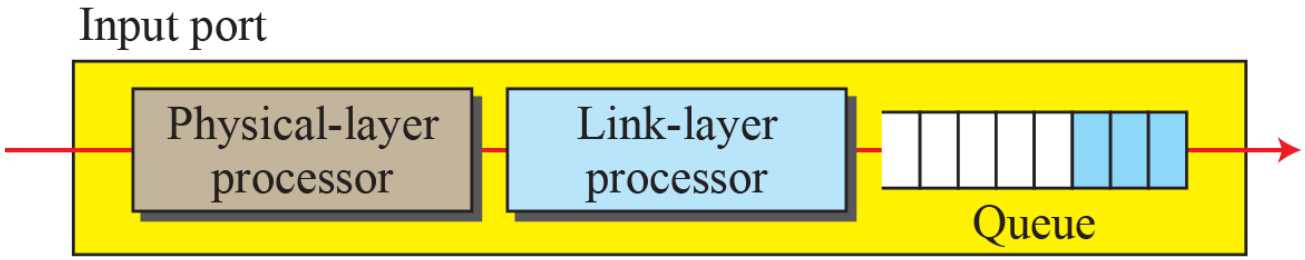


Figure 4.18: Output port

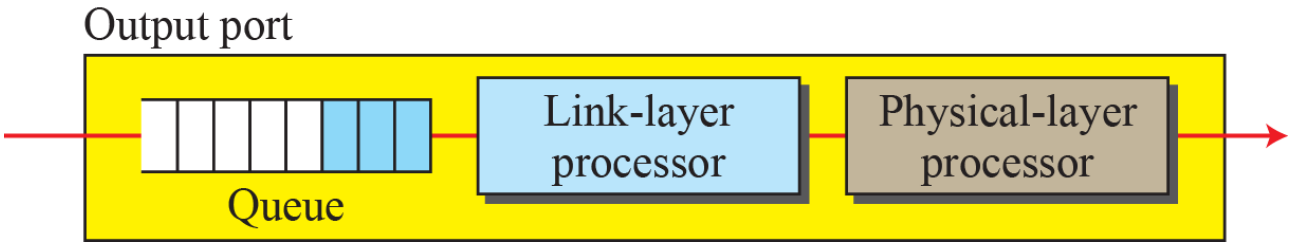


Figure 4.19: Crossbar switch

- *A crossbar switch is a switch connecting multiple inputs to multiple outputs in a matrix manner.*
- *Architecture of a unidirectional crossbar switch: The crossbar switch can switch inputs to the outputs - imagine each box is a CPU or a memory module.*

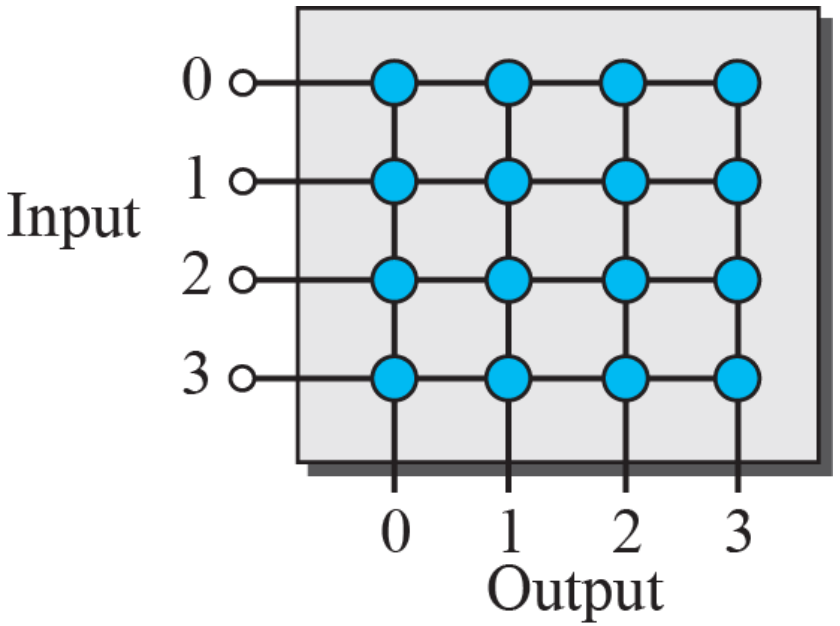
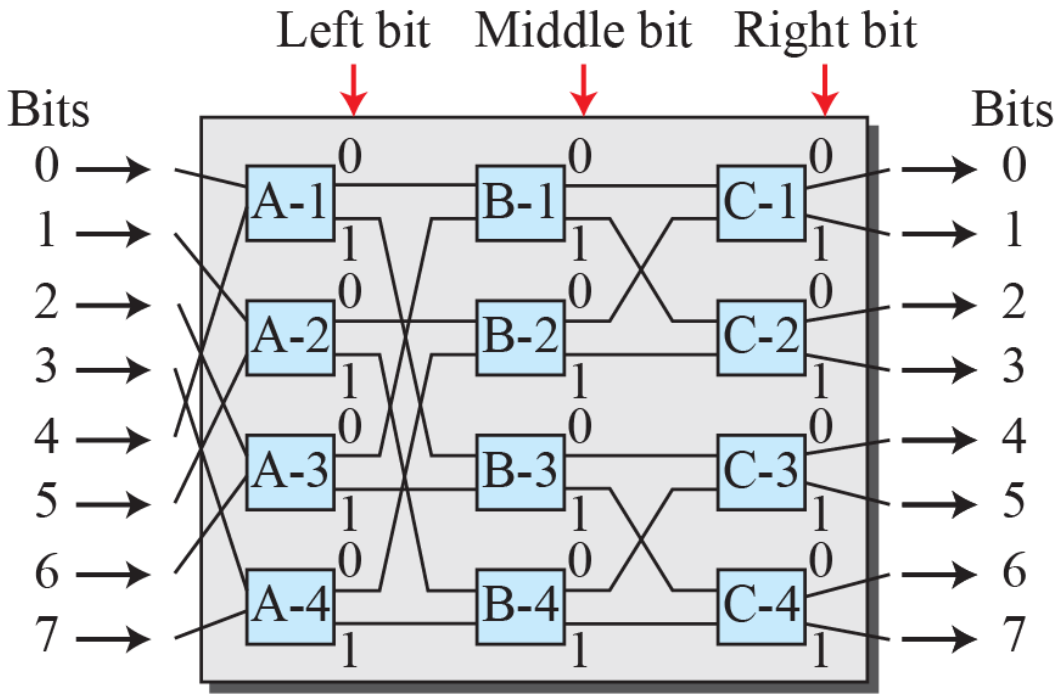
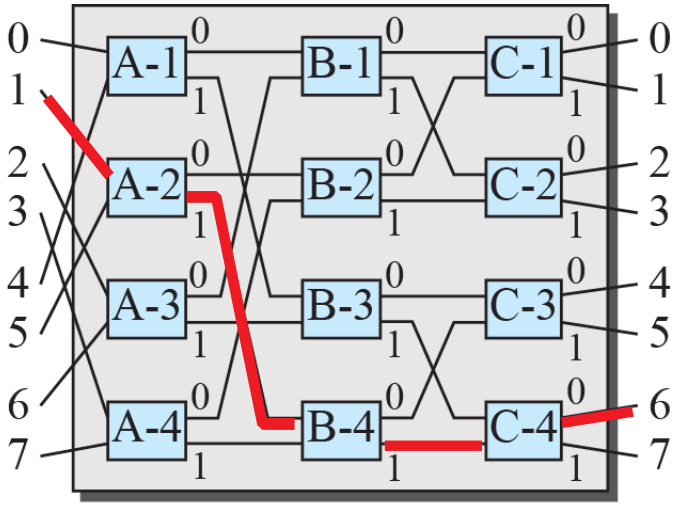


Figure 4.20: Banyan switch

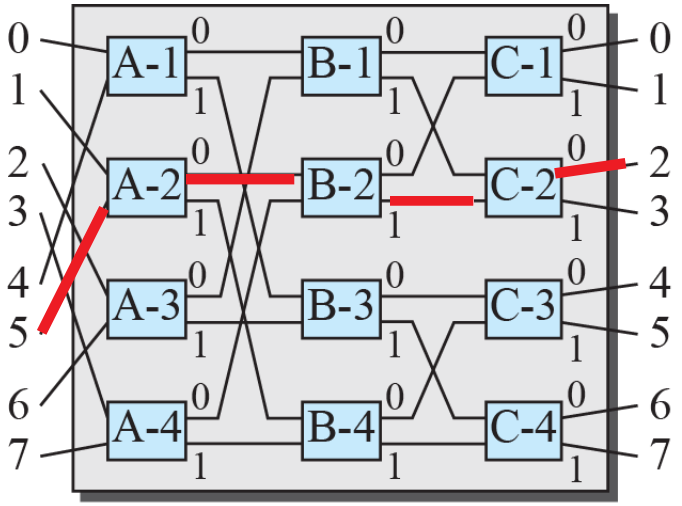


- The **first stage** routes the packet based on the **high-order bit** of the binary string.
- The **second stage** routes the packet based on the **second high-order bit**, and so on.
- Figure shows a banyan switch with eight inputs and eight outputs. The number of stages is $\text{Log}_2(8) = \mathbf{3}$.

Figure 4.21: Examples of routing in a banyan switch

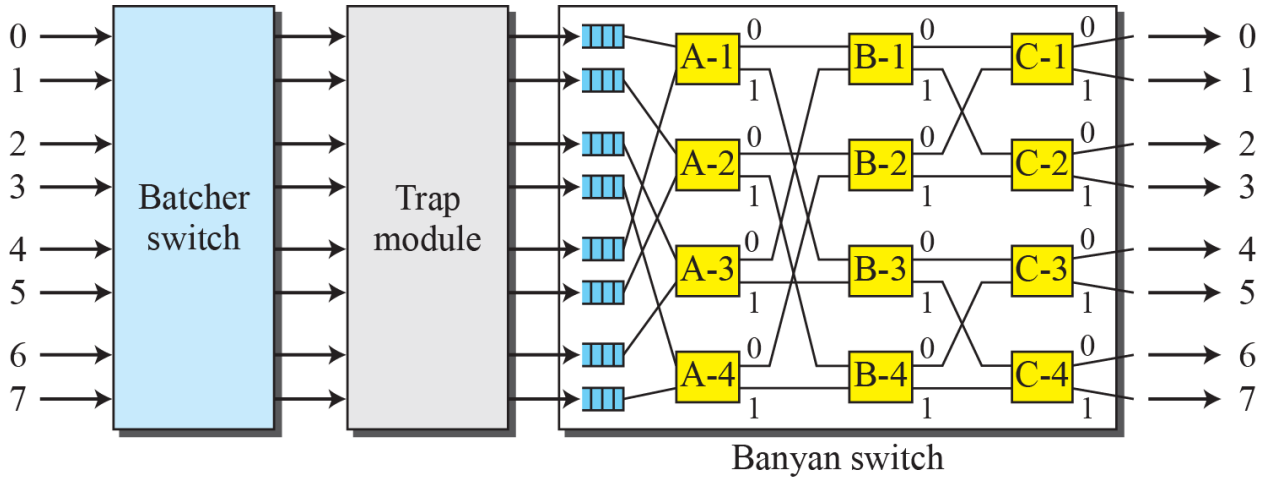


a. Input 1 sending to output 6 (110)



b. Input 5 sending to output 2 (010)

Figure 4.22: Batcher-banyan switch

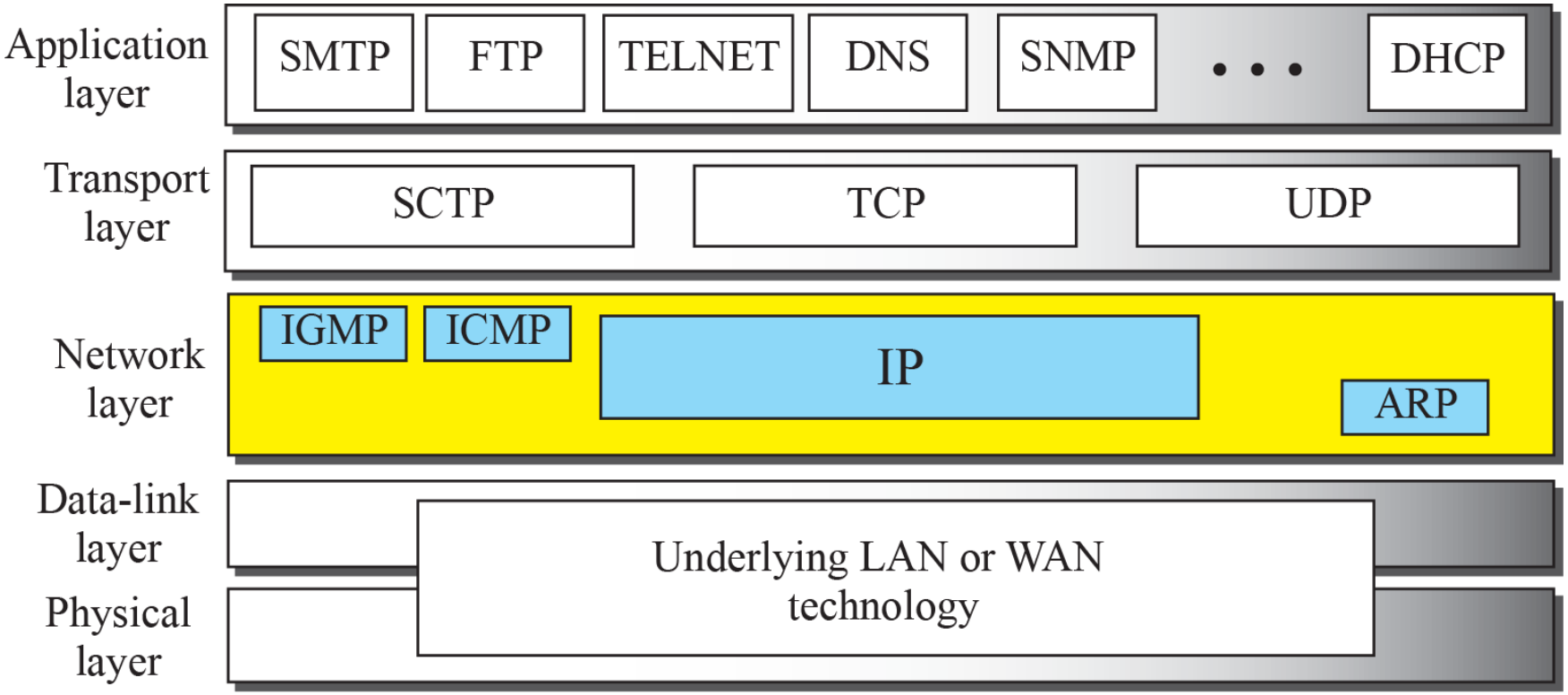


- **Batcher-Banyan switch:** On each switch cycle, each input line at the top puts one cell into the Batcher sorter switch.
- **The Batcher sort switch:** Sorts the cells and passes them onto the trap module.
- **The trap module:** detects duplicate cells to the same destination and selects one cell for each destination to pass on to the banyan.
- **The trap module** then compresses the list of sorted cells so there are no gaps and place's the cells on the input lines into the banyan switch.
- **The banyan switch** routes the cells to their appropriate outputs.
- The feature that tends to differentiate Batcher-Banyan designs is the design of their trap modules.
- The challenge of the trap module is to avoid discarding cells when there is more than one cell for the same destination while making sure cells do not get reordered.

4-2 NETWORK-LAYER PROTOCOLS

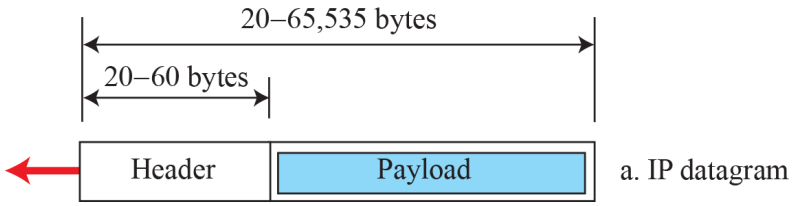
- *Here in this part we will see how the network layer is implemented in the TCP/IP protocol suite.*
- *The network layer protocols has gone through to changes to arrive at these major versions;*
  - *in this section, we concentrate on the current version (**IPv4**),*
  - *in the last section of this chapter, we briefly discuss version (**IPv6**), which is on the horizon.*

Figure 4.23: ***Position of IP and other network-layer protocols in TCP/IP protocol suite***



4.2.1 IPv4 Datagram Format

- *Packets used by the IP are called IP_datagrams. Figure 4.24 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.*

- ***Fragmentation***
  - ***Maximum Transfer Unit (MTU)***
  - ***Fields Related to Fragmentation***

- ***Security of IPv4 Datagrams***
  - ***Packet Sniffing***
  - ***Packet Modification***
  - ***IP Spoofing***
  - ***IPSec***

Figure 4.24: IP datagram

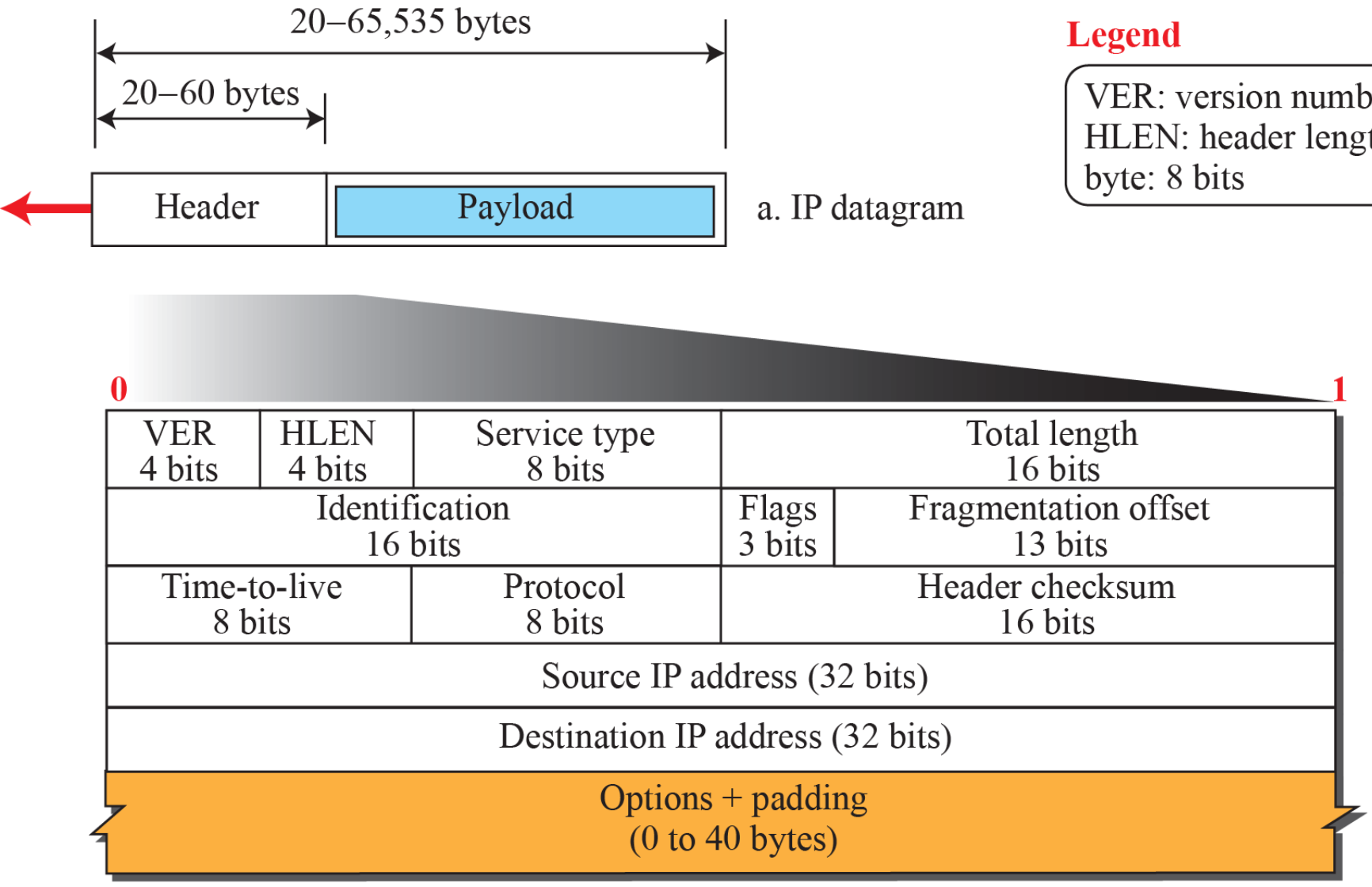
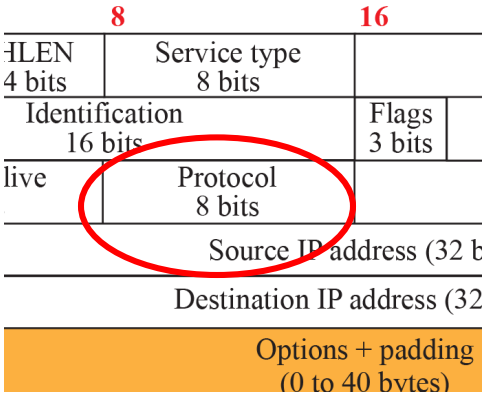


Figure 4.25: Multiplexing and demultiplexing using the value of the protocol field (8-bits)

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

Some protocol values



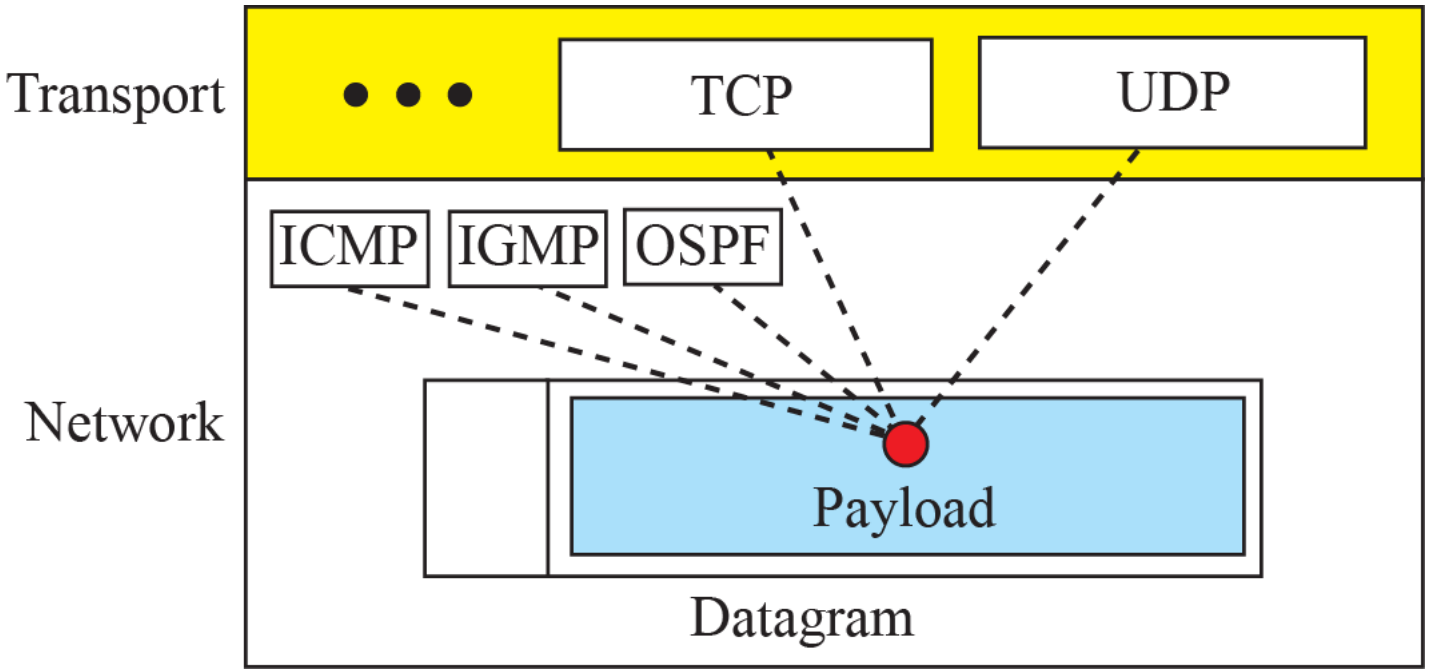
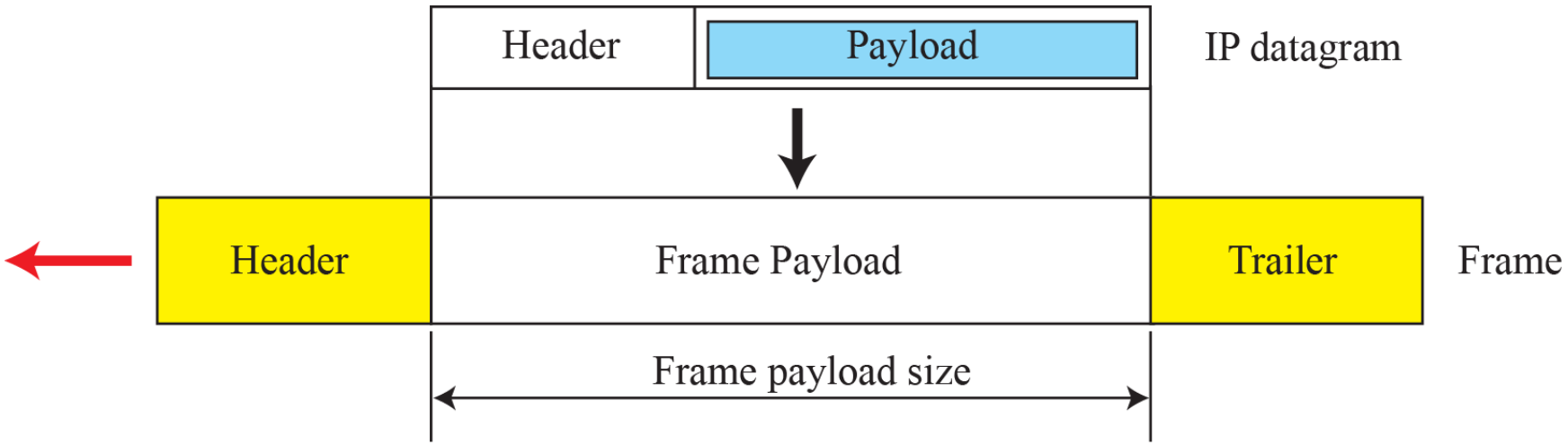
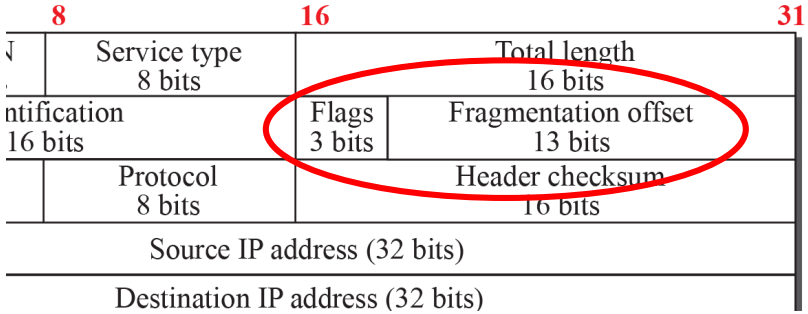


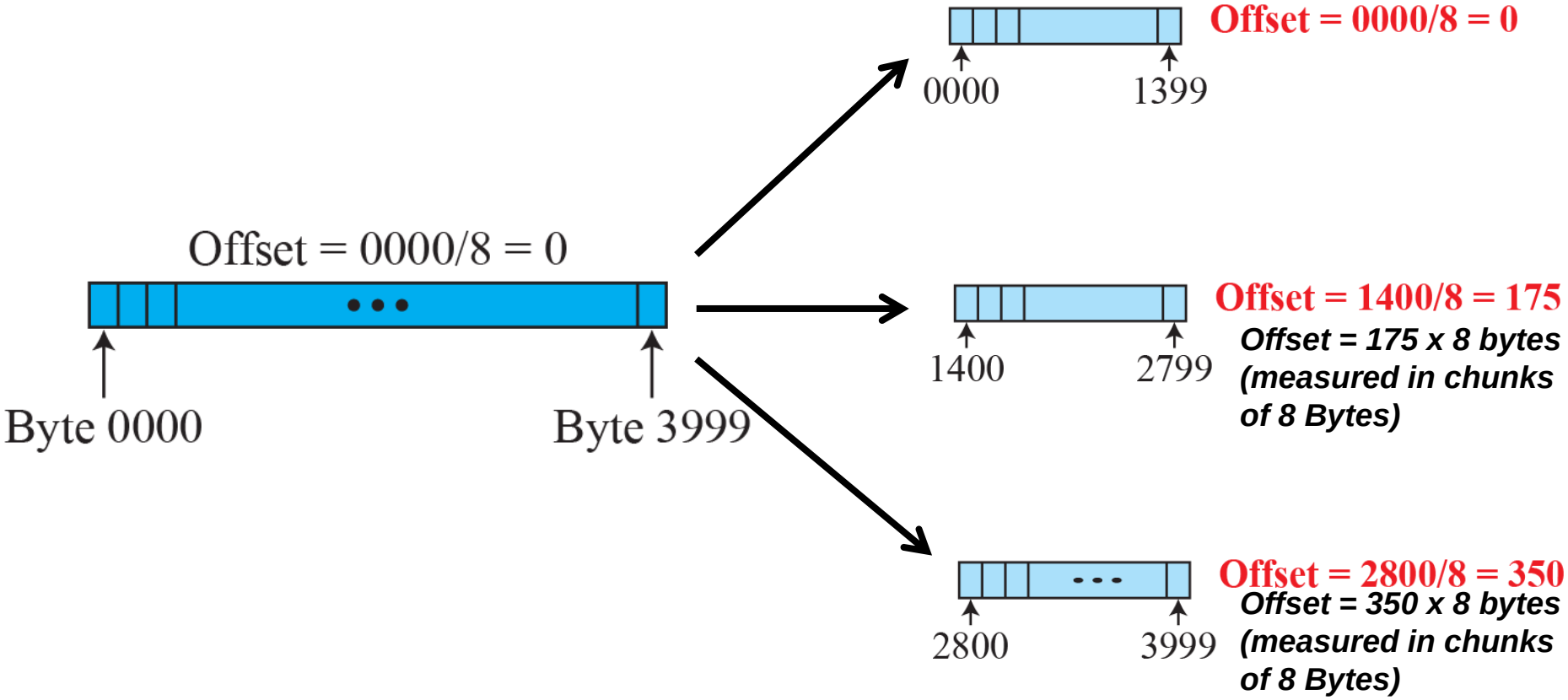
Figure 4.26: Maximum transfer unit (MTU)



MTU: Maximum size of frame payload

Figure 4.27: Fragmentation example





4.2.2 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, because if the device is moved to another network, the IP address may be changed.*

4.2.2 (continued)

- *Address Space*
- *Notation*
- *Hierarchy in Addressing*
- *Classful Addressing*
  - *Address Depletion*
  - *Subnetting and Supernetting*
  - *Advantage of Classful Addressing*
- *Classless Addressing*
  - *Prefix Length: Slash Notation*
  - *Extracting information from an address*
  - *Address Mask*
  - *Network Address*
  - *Subnetting*
  - *Address Aggregation*
  - *Special Addresses*
- *Dynamic Host Configuration Protocol (DHCP)*
  - *DHCP Message Format*
  - *DHCP Operation*
  - *Two Well-Known Ports*
  - *Using FTP*
  - *Error Control*
  - *Transition States*
- *NAT*
  - *Address Translation*
  - *Translation Table*

Figure 4.29: Three different notations in IPv4 addressing

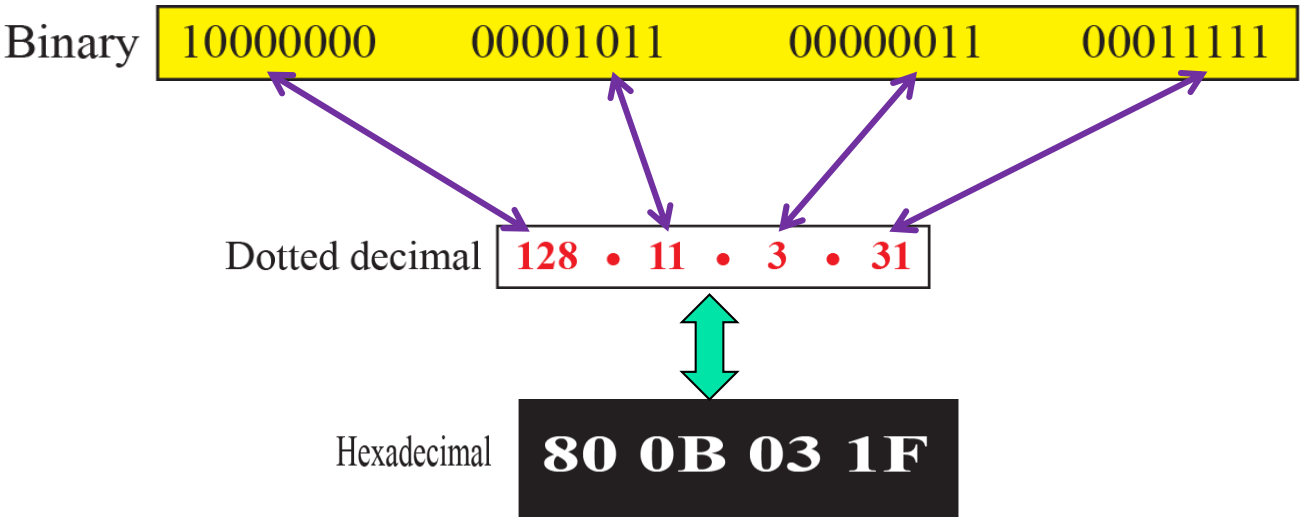


Figure 4.30: Hierarchy in addressing

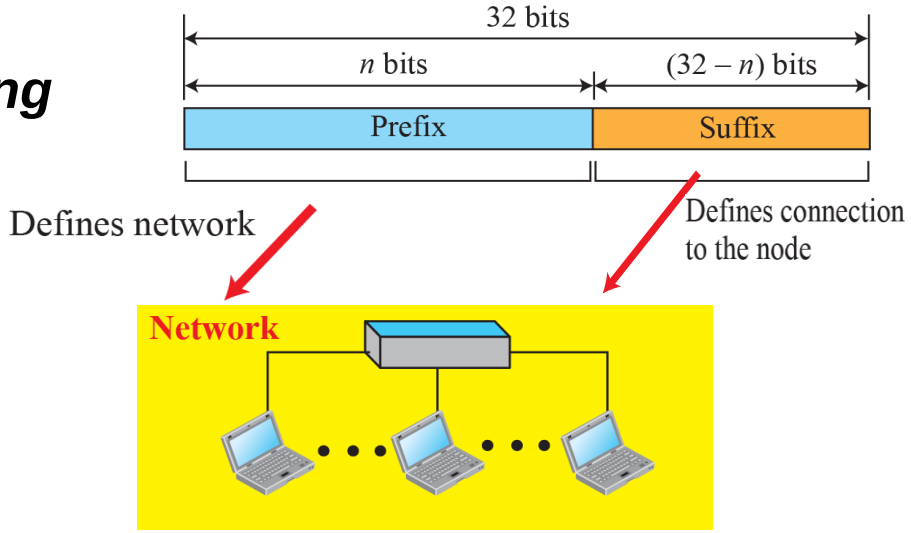
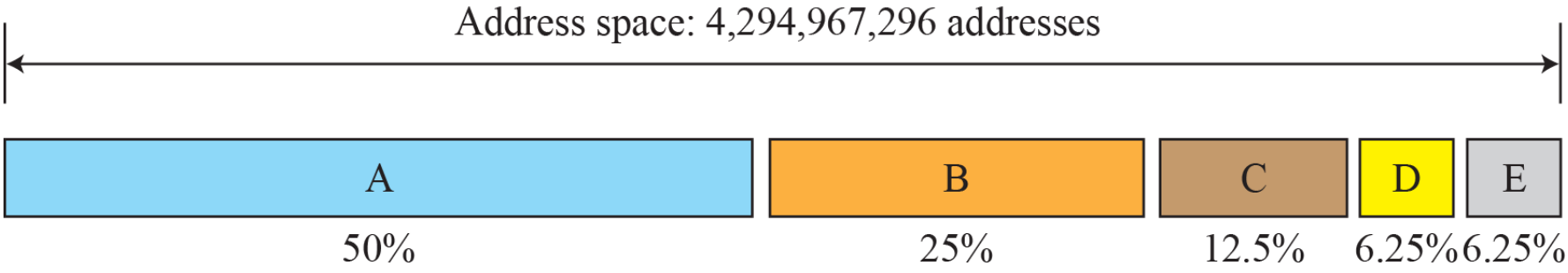
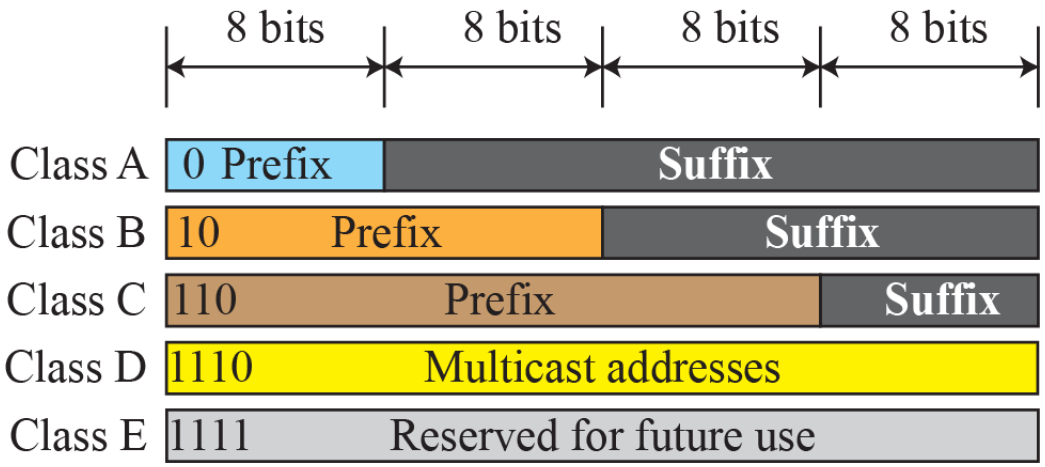


Figure 4.31: Occupation of the address space in *classful* addressing





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

Figure 4.33: Slash notation (CIDR)

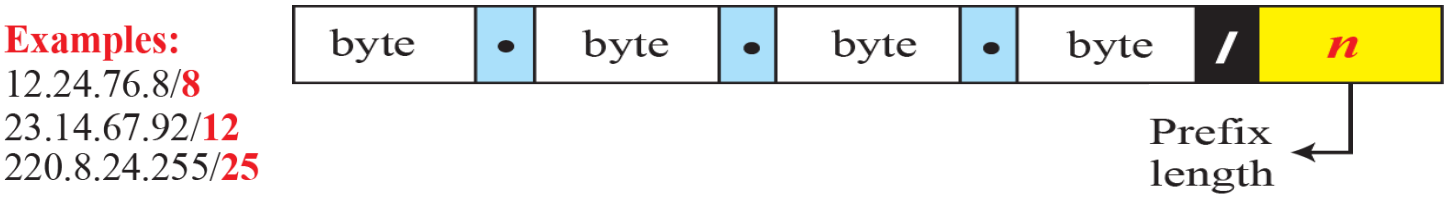
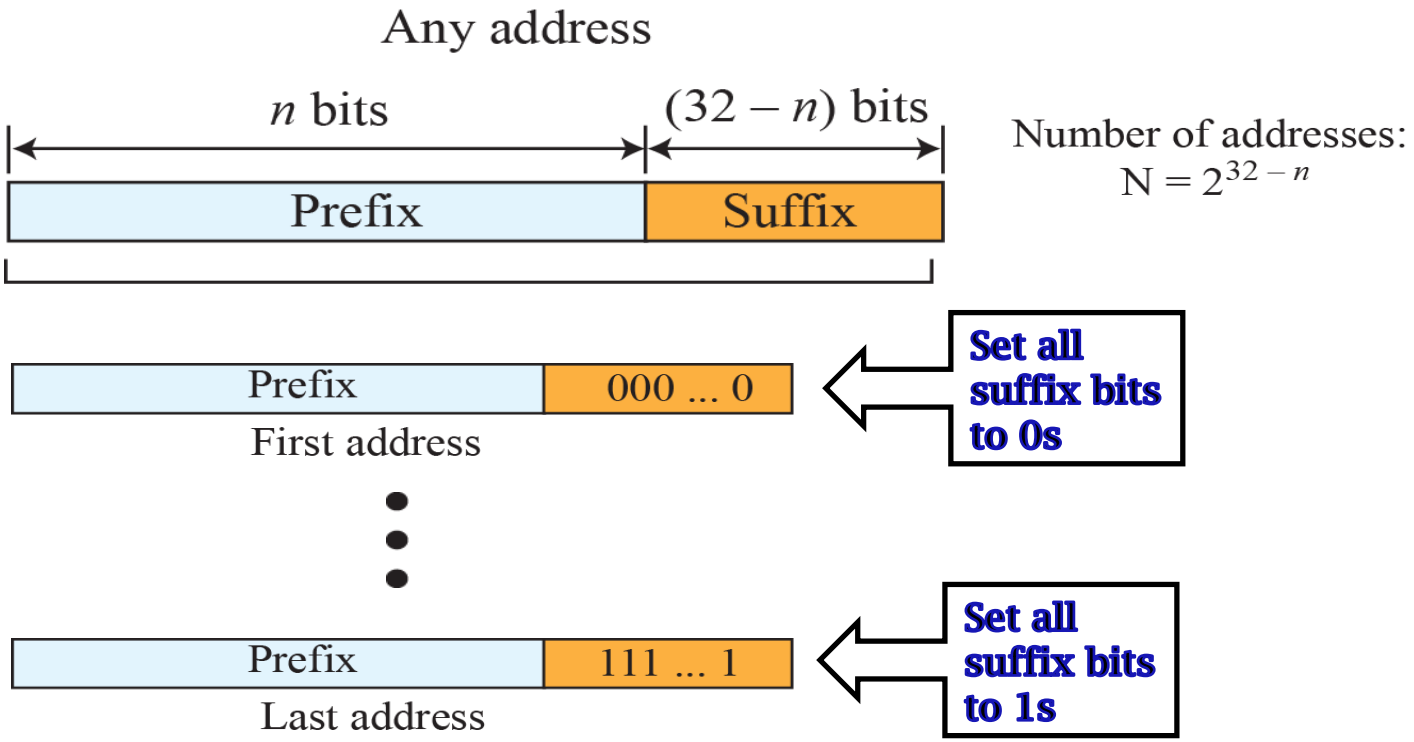


Figure 4.34: Information extraction in classless addressing



Example 4.1

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, where $n=27$

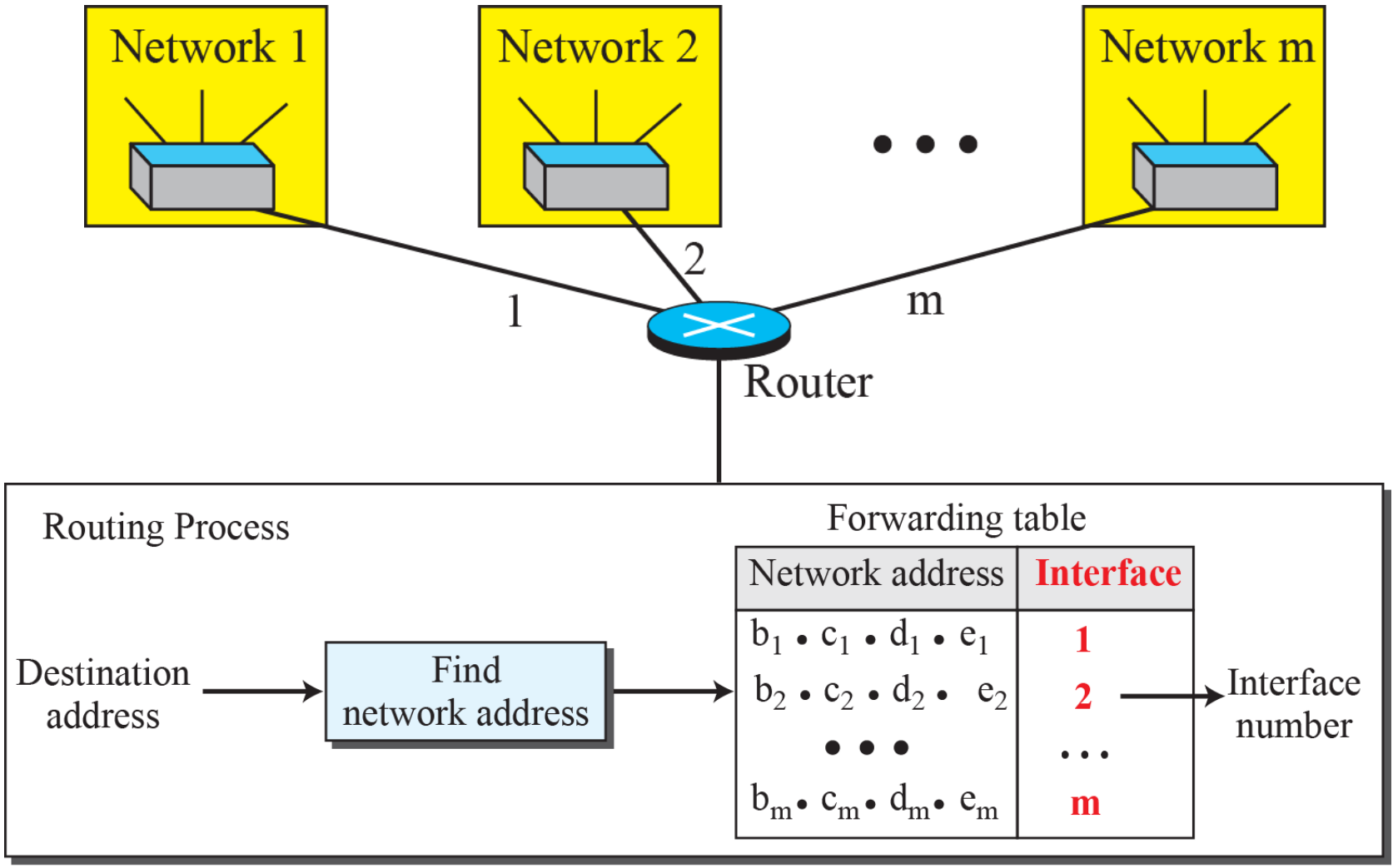
The **first address** can be found by keeping the first **27 bits** and changing the rest of the **bits to 0s**.

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 4.35: Network address



Example 4.5

- 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**:
  1. one subblock of 10 addresses,
  2. one subblock of 60 addresses, and
  3. one subblock of 120 addresses.
- Design the **subblocks**.

Solution:-

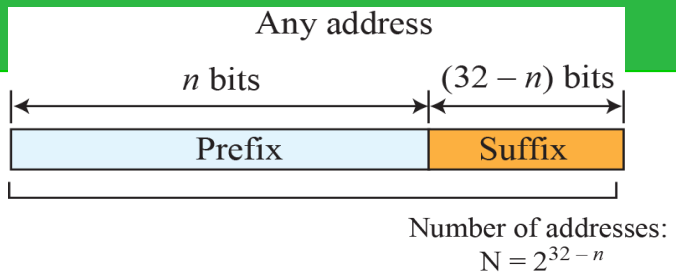
☞ Total # of addresses are $2^{32-24} = 256$ addresses in whole complete block.

☞ 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 sub-blocks, starting with the largest and ending with the smallest one.

Continued ….

Example 4.5 (continued)



Solution (continued..)

a. one subblock of 10 addresses,
b. one subblock of 60 addresses, and
c. one subblock of 120 addresses.

- c. The number of addresses in the **largest sub-block**, which requires **120 addresses**, is not a power of 2. **We allocate 128 addresses**. The subnet mask for this subnet can be found as **netmask1** = 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**.

0 – 127 = 128 address

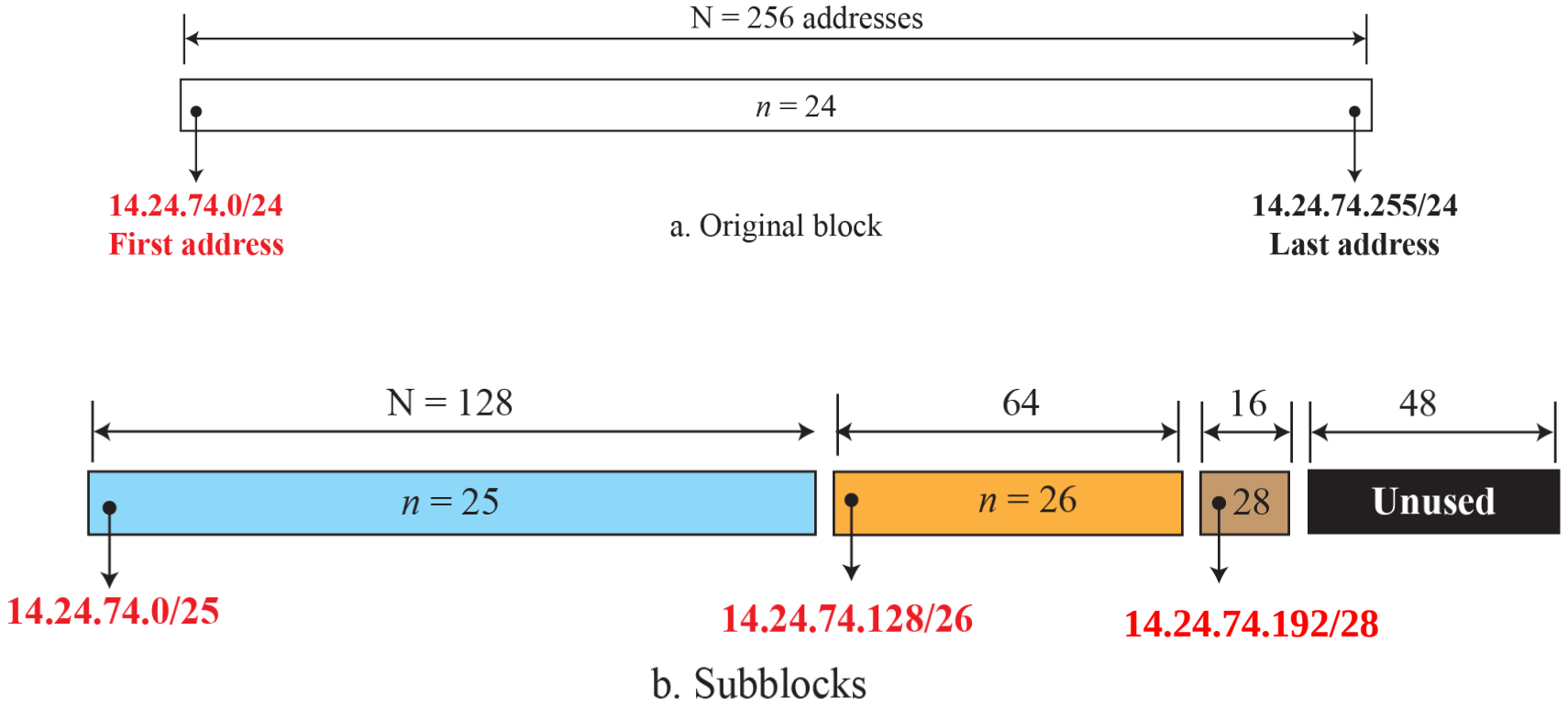
- b. The number of addresses in the **second largest sub-block**, which requires **60 addresses**, is not a power of 2 either. **We allocate 64 addresses**. The subnet mask for this subnet can be found as netmask2 = 32 − $\log_2 64$ = **26**.
  - The **first** address in this block is 14.24.74.128/**26**;
  - the **last** address is 14.24.74.191/**26**.

128 – 191 = 64 address

- a. The number of addresses in the **smallest sub-block**, which requires **10 addresses**, is not a power of 2. **We allocate 16 addresses**. The subnet mask for this subnet can be found as netmask3= 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**.

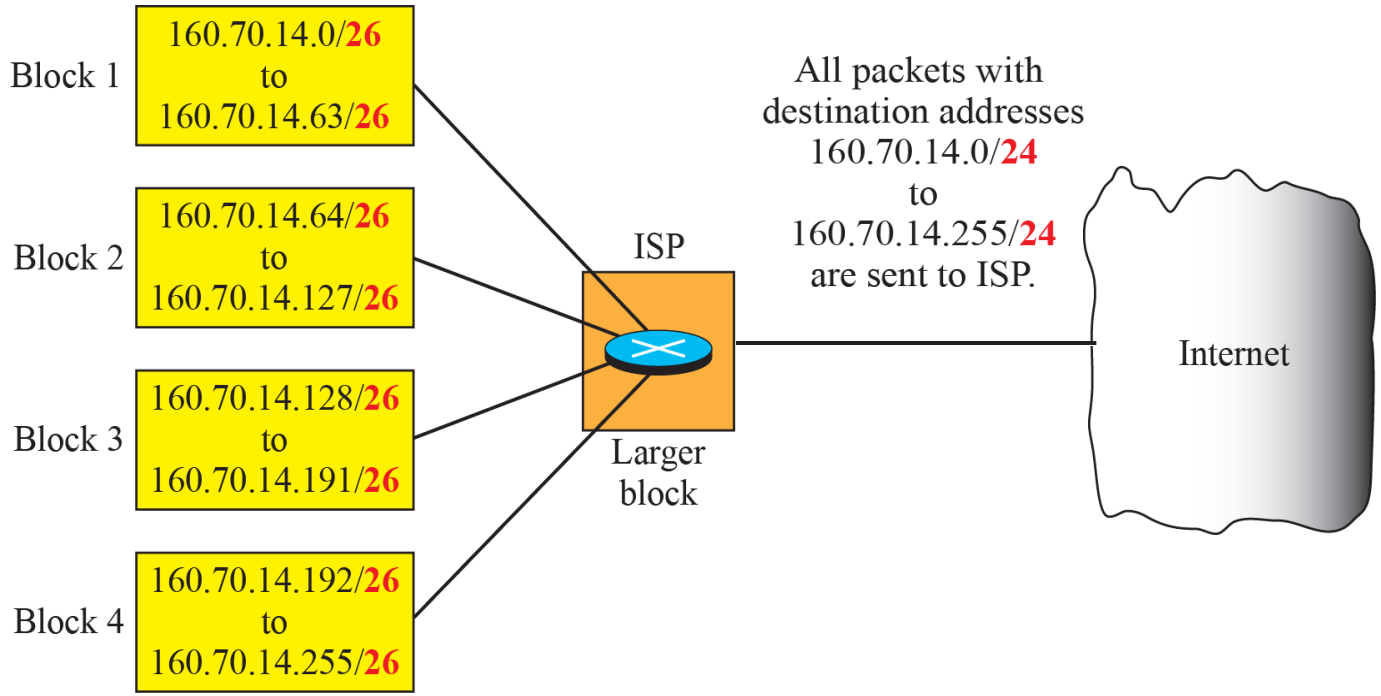
192 – 207 = 16 address

Figure 4.36: Solution to Example 4.5



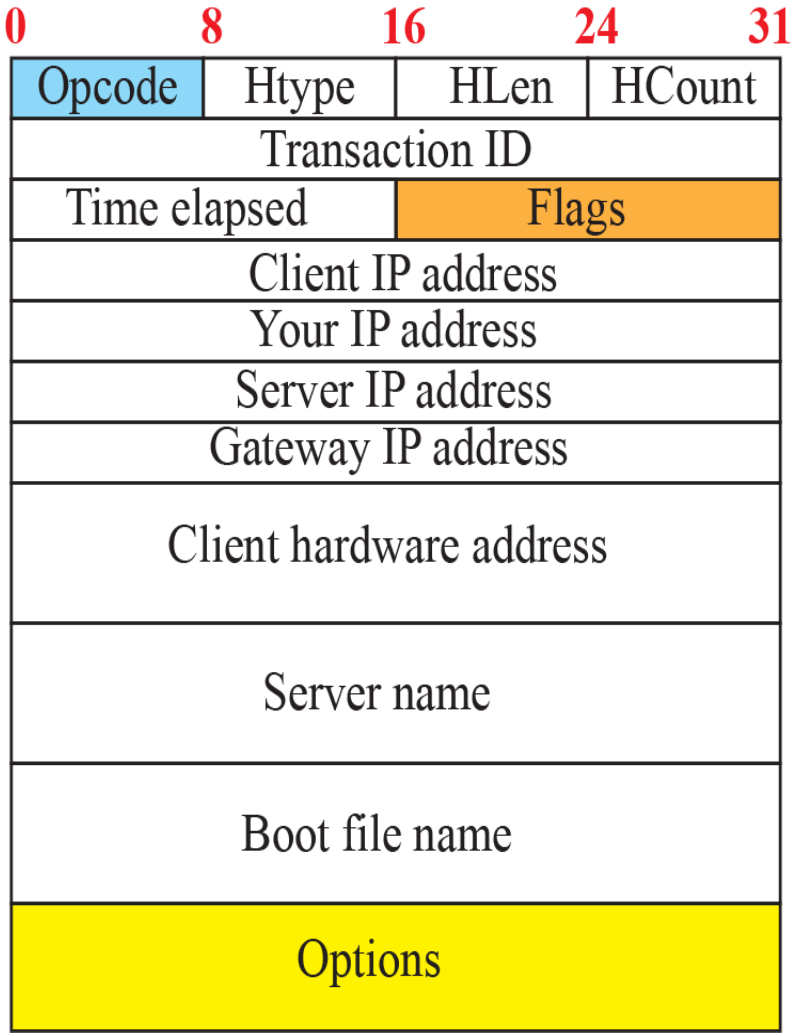
- If we add all addresses in all the subblocks, the result is 208 addresses, which means **48 addresses are left in reserve or unused**.
  - The first address of **unused range is 14.24.74.208**.
  - The **last address is 14.24.74.255.**
- Figure 4.36 shows the configuration of sub-blocks. Figure shows the first address in each block.

Figure 4.37: Example 4.6 of address aggregation



- ✓ *Figure 4.37 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. All packages coming from internet are sent first to the ISP and then distributed to the corresponding networks behind the ISP.*

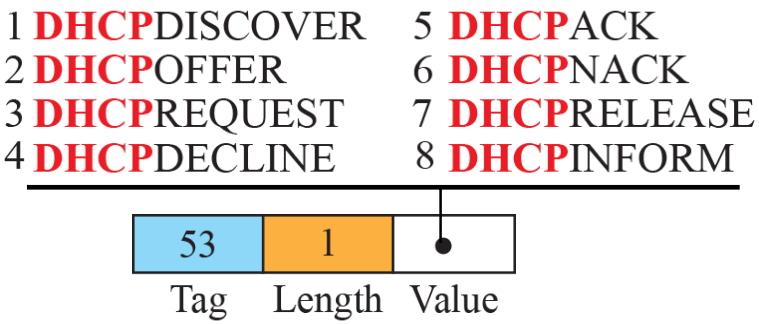
Figure 4.38: DHCP message format



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 4.39: Option format



DHCP Msg Type	Description
DHCPDiscover	The first time a DHCP client computer attempts to log on to the network, it requests IP address information from a DHCP server by broadcasting a **DHCPDiscover** packet. The source IP address in the packet is 0.0.0.0 because the client does not yet have an IP address.
DHCPOffer	Each DHCP server that receives the client DHCPDiscover packet responds with a **DHCPOffer** packet containing an unleased IP address and additional TCP/IP configuration information, such as the subnet mask and default gateway.
DHCPRequest	When a DHCP client receives a DHCPOffer packet, it responds by broadcasting a **DHCPRequest** packet that contains the offered IP address, and shows acceptance of the offered IP address.
DHCPAcknowledge (DHCPAck)	The selected DHCP server acknowledges the client DHCPRequest for the IP address by sending a **DHCPAck** packet.
DHCPNak	Whenever a DHCP server receives a request for an IP address (renewal) that is invalid according to the scopes that it is configured with, it sends a **DHCPNak** message to the client.
DHCPDecline	If the DHCP client determines the offered configuration parameters are invalid, it sends a **DHCPDecline** packet to the server, and the client must begin the lease process again.
DHCPRelease	A DHCP client sends a **DHCPRelease** packet to the server to release the IP address and cancel any remaining lease.
DHCPInform	**DHCPInform** is a new DHCP message type, defined in RFC2131, used by computers on the network to request and obtain information from a DHCP server for use in their local configuration.

Figure 4.40: Operation of DHCP

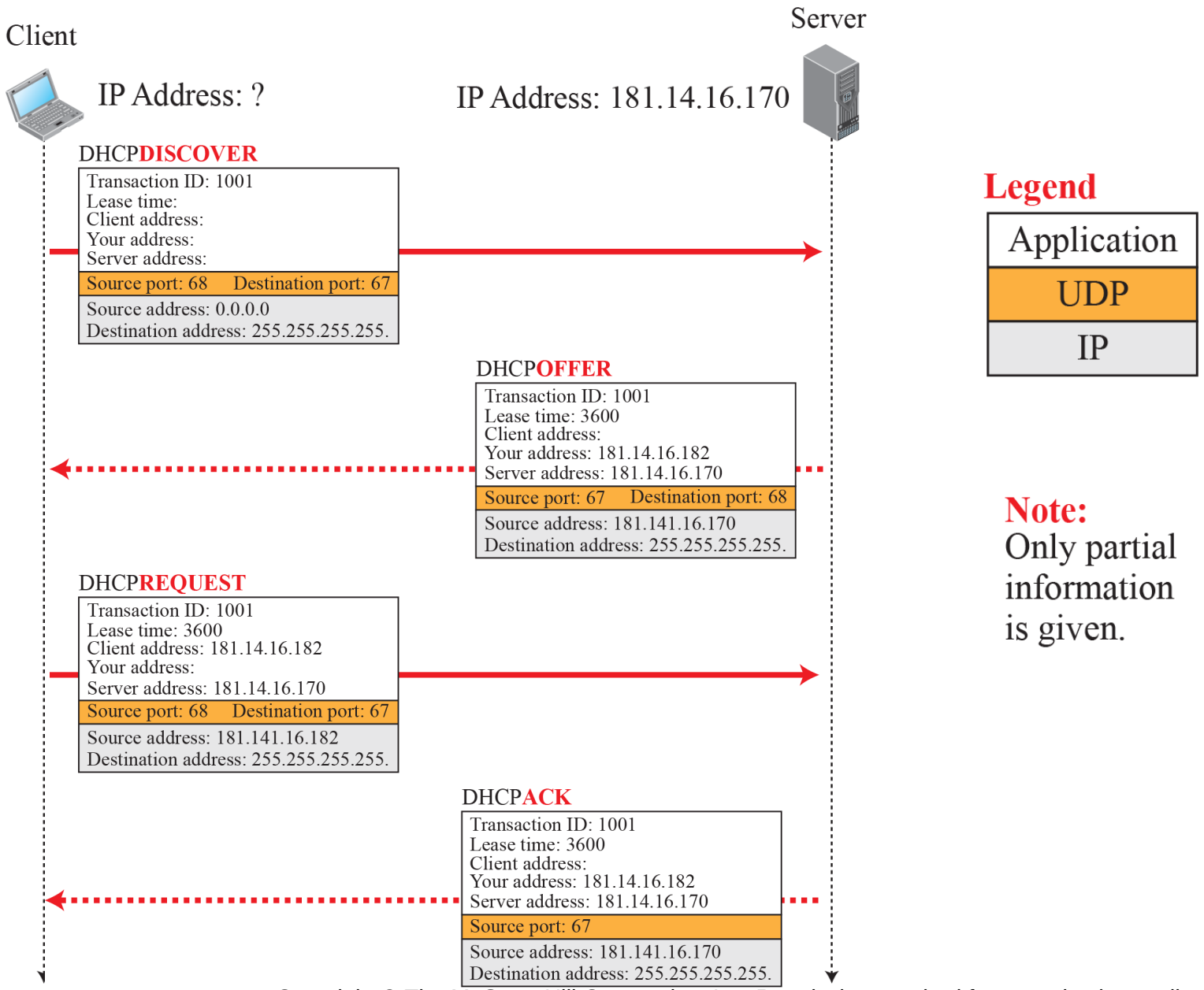
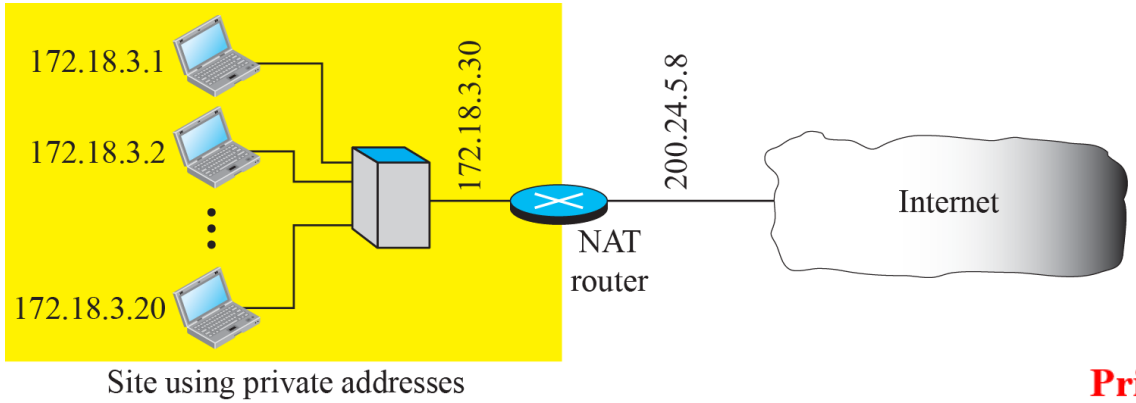


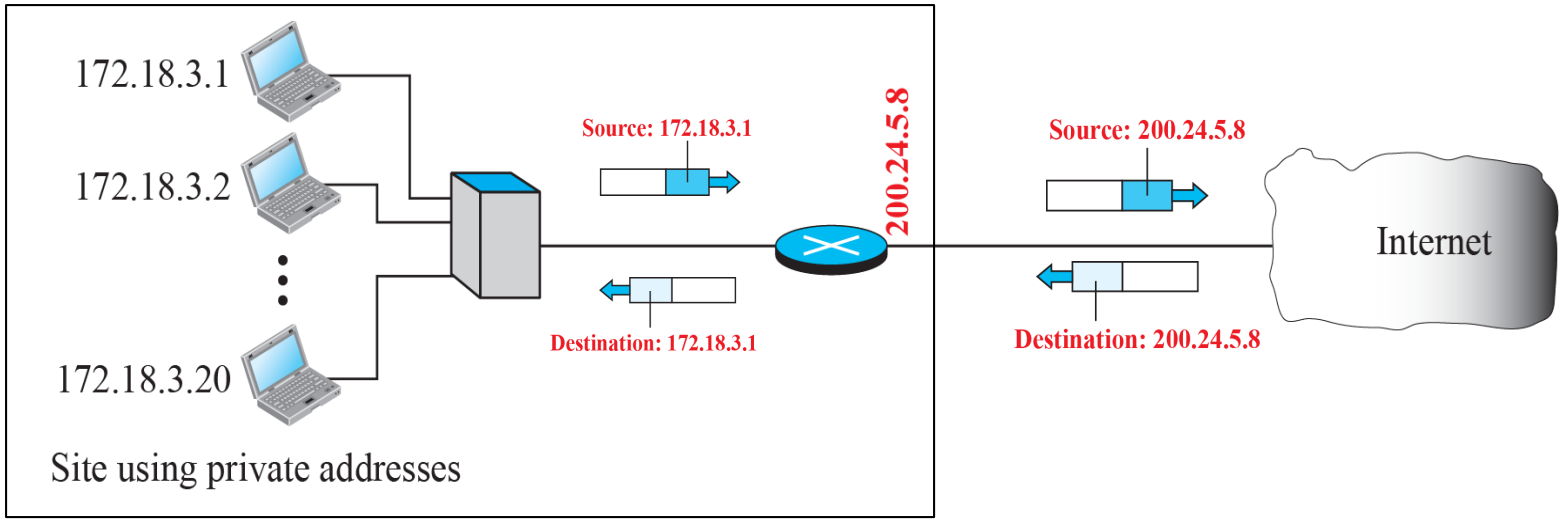
Figure 4.42: NAT

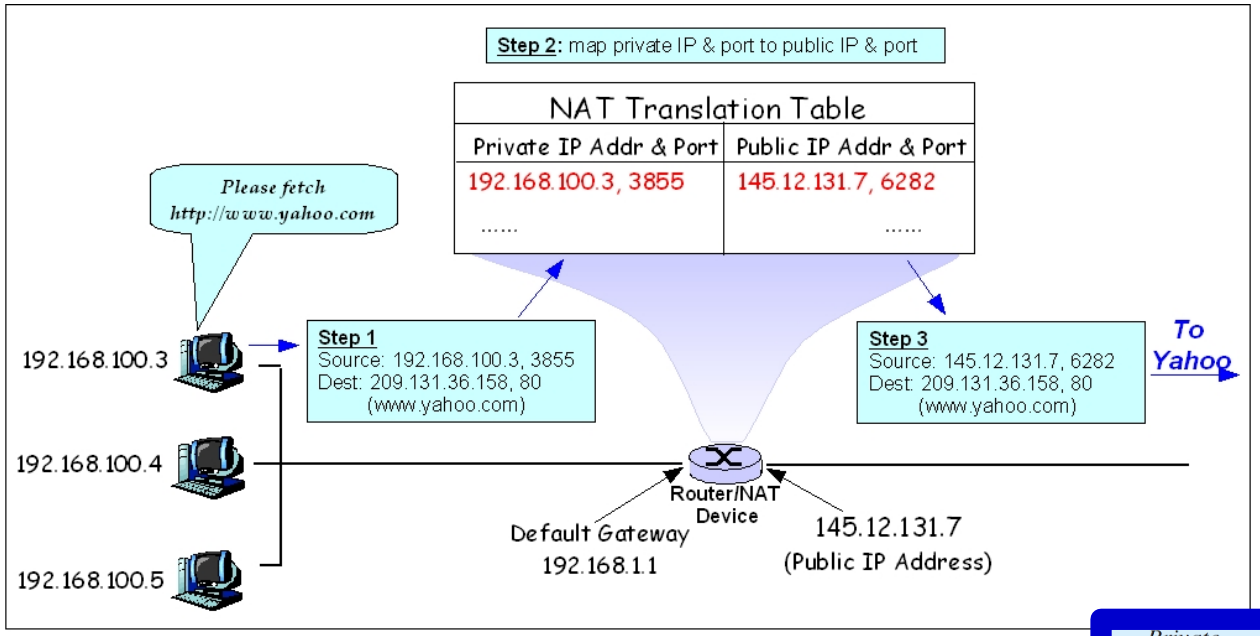


Private IPv4 address spaces

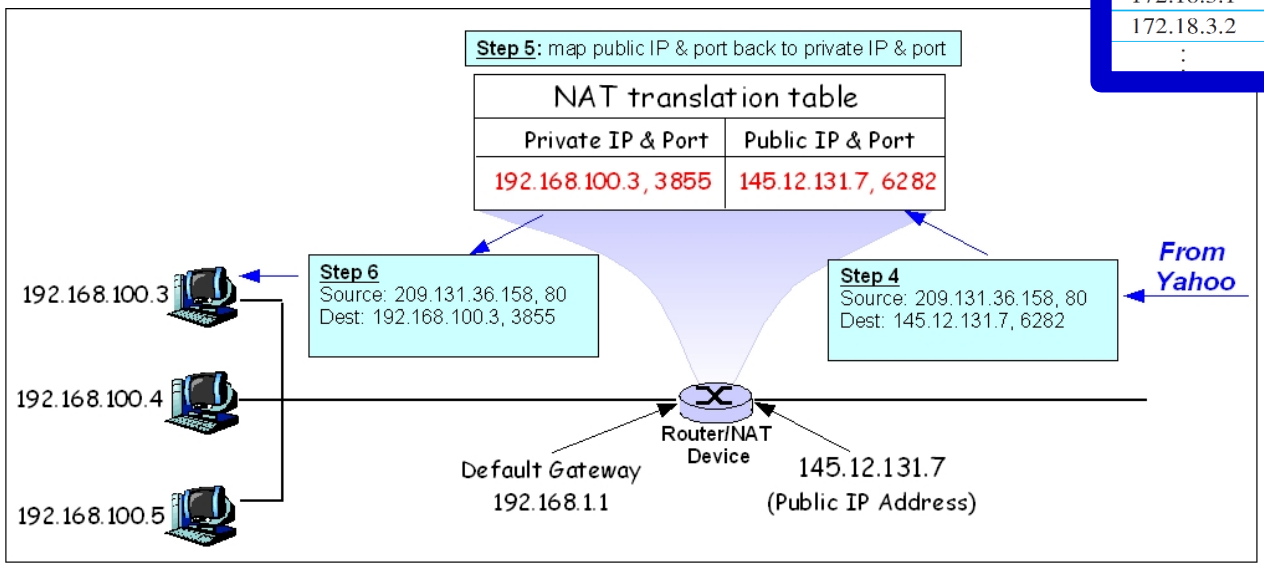
RFC1918 name	IP address range
24-bit block	10.0.0.0 – 10.255.255.255
20-bit block	172.16.0.0 – 172.31.255.255
16-bit block	192.168.0.0 – 192.168.255.255

Figure 4.43: Address translation





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
:	:	:	:	:



Reference: https://commons.wikimedia.org/wiki/File:Network_Address_Translation_(file2).jpg
Reference: https://upload.wikimedia.org/wikipedia/commons/a/a4/NAT3.jpg

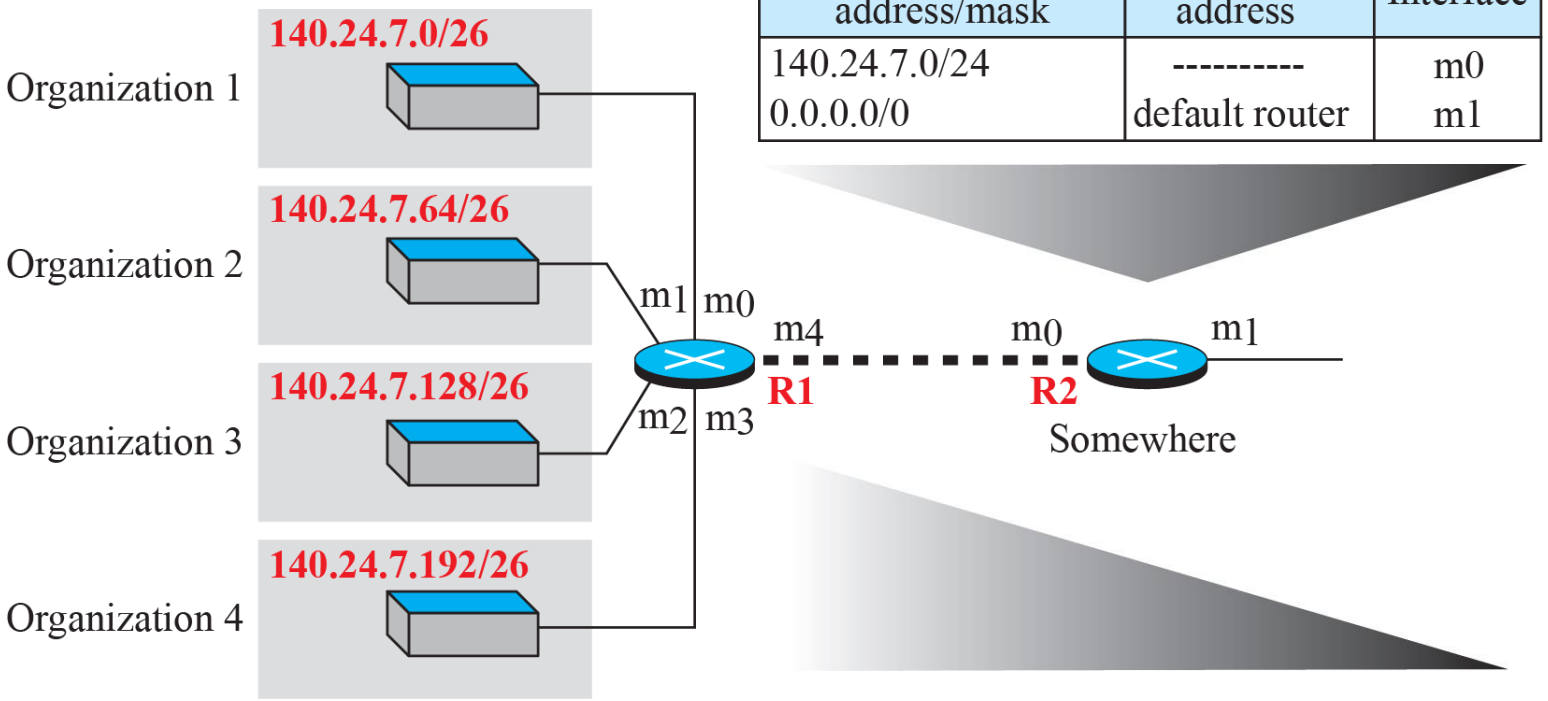
4.2.3 Forwarding of IP Packets

- *Concept of **forwarding** packet at the network layer.*
- *Extend the concept to include the **role of IP addresses** in forwarding.*
- *Forwarding means to placing the packet in its route to its **destination**.*
- *Since the Internet is complex combination of links (networks), **forwarding means to deliver the packet to the next hop** (which can be the **final** destination or the **intermediate** connecting device).*

4.2.3 (continued)

- **Forwarding Based On Destination Address**
  - Address Aggregation
  - Longest Mask Matching
  - Hierarchical Routing
  - Geographical Routing
  - Forwarding Table Search Algorithms
- **Forwarding Based on Label**
  - Multi-Protocol Label Switching (MPLS)
  - A New Header
  - Hierarchical Routing

Figure 4.47: Forwarding by Address aggregation



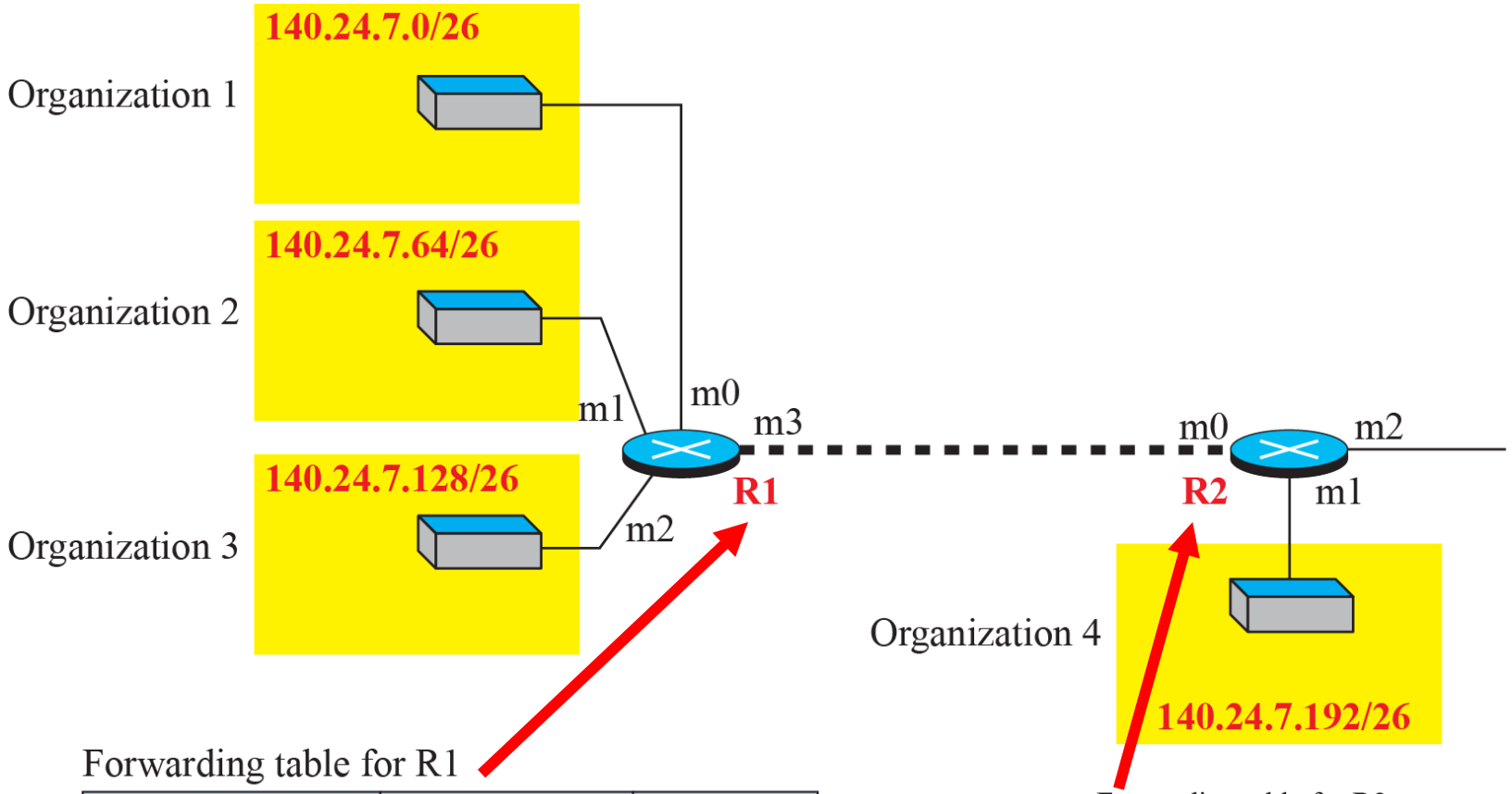
Forwarding table for R2

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 4.48: Longest mask matching



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

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

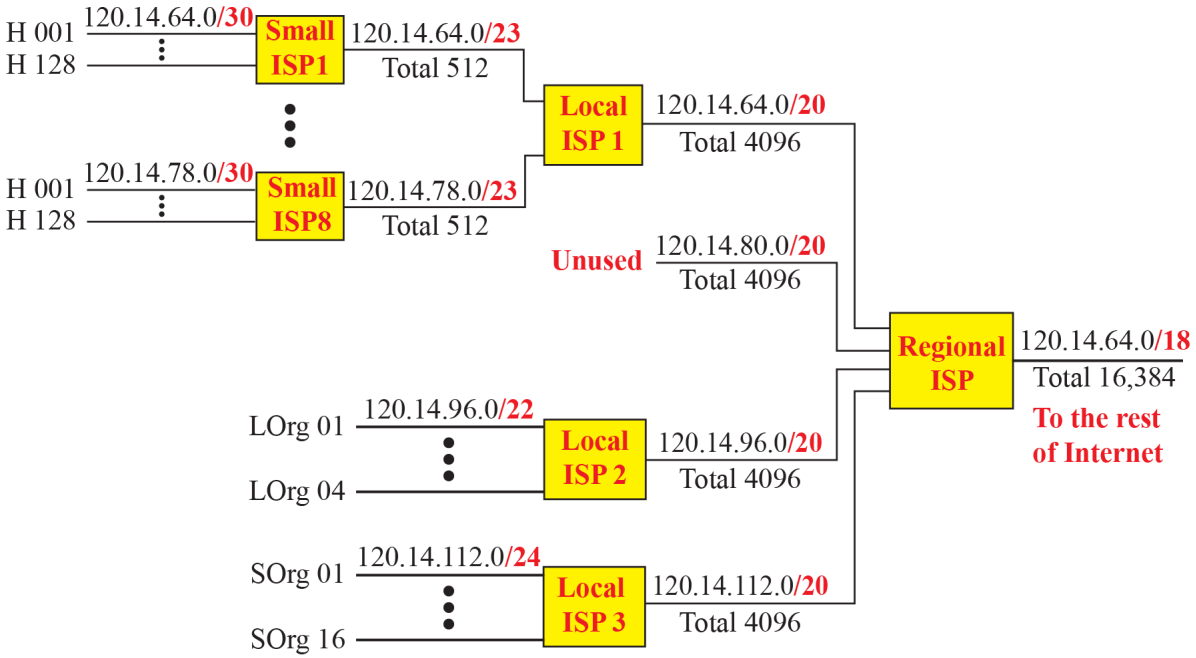


Figure 4.49: ***Hierarchical routing with ISPs***

- ✓ **Example of hierarchical routing in Figure 4.49.**
- ✓ *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 sub-blocks are assigned to **three local ISPs**,*
- ✓ *the second sub-block 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 4.51: Example 4.12: Forwarding based on label

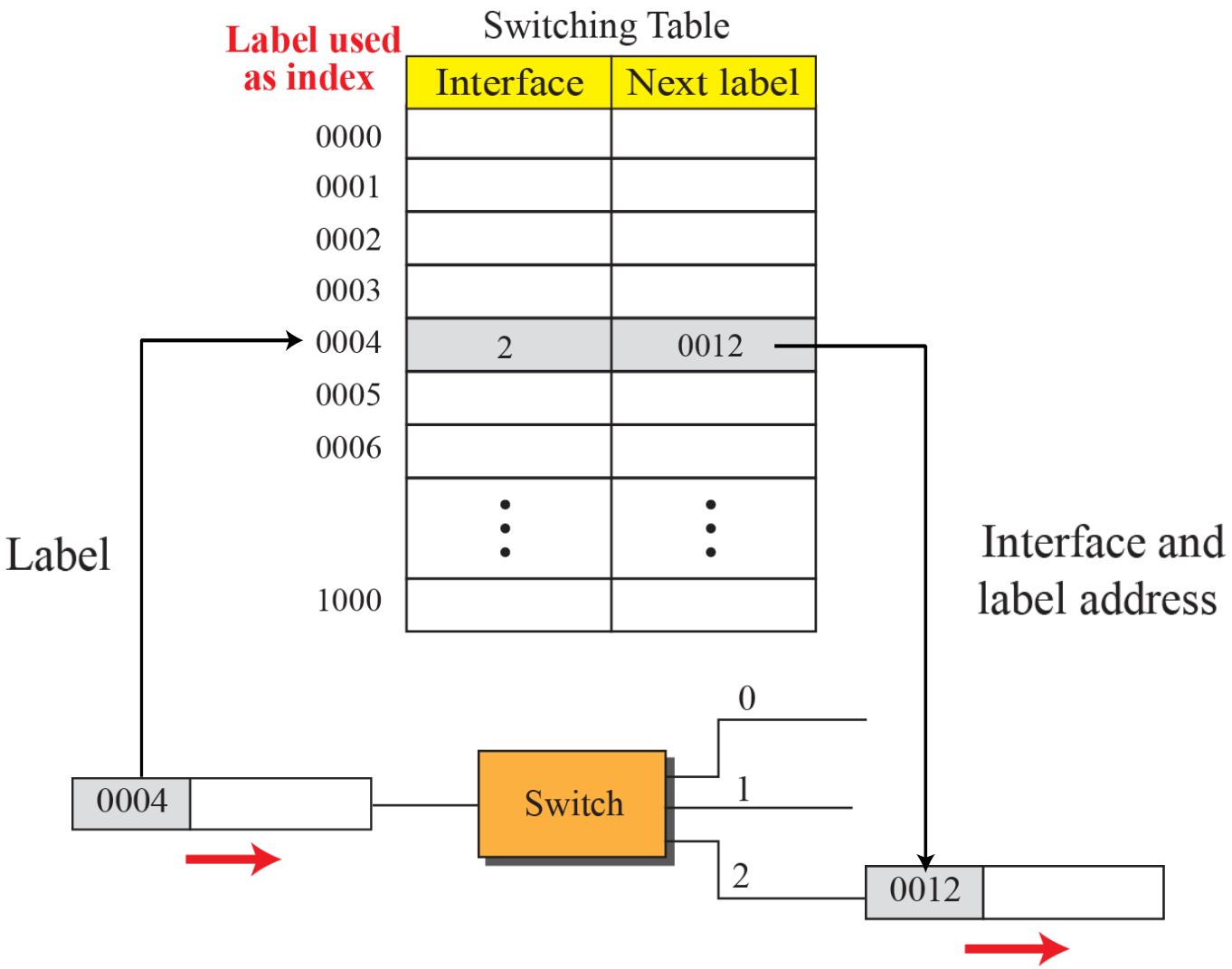


Figure 4.52: MPLS header added to an IP packet

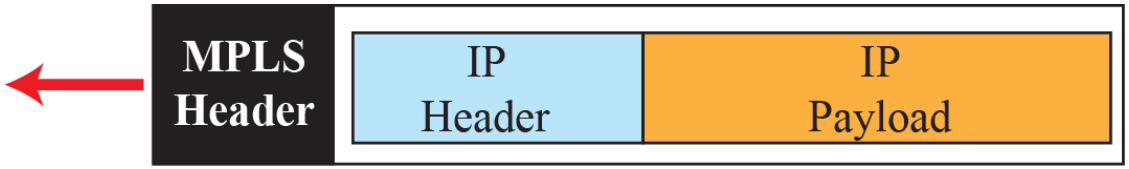
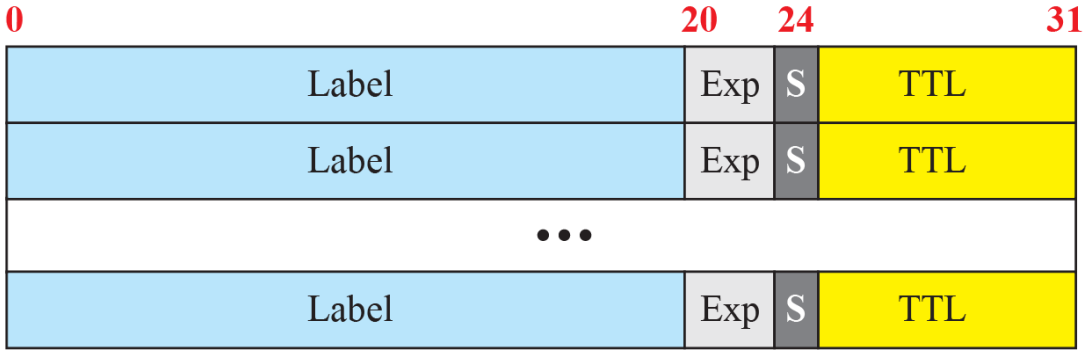


Figure 4.53: MPLS header made of a stack of labels



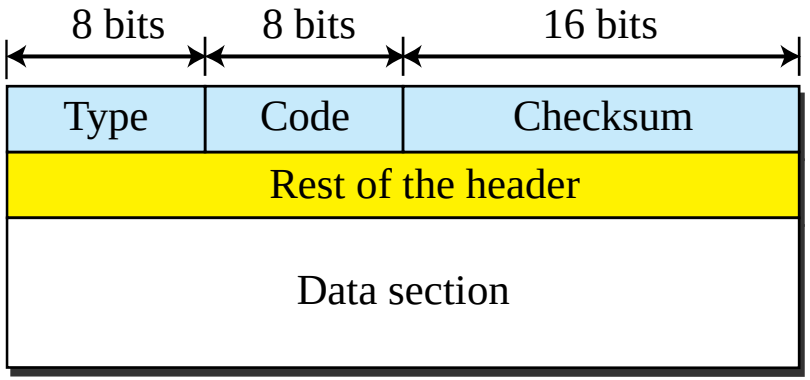
4.2.4 ICMPv4

- *The IPv4 has no error-reporting or error-correcting mechanism.*
- *What happens if something goes wrong?*
- *There are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.*
- *The IP protocol also lacks a mechanism for host and management queries.*

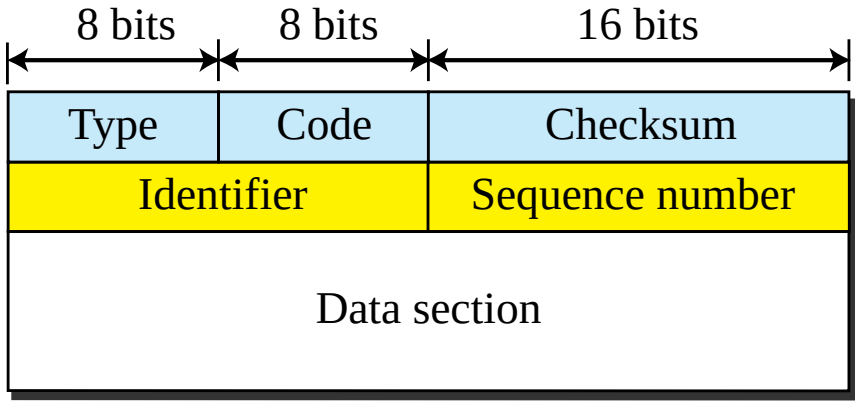
The Internet Control Message Protocol version 4 (ICMPv4) has been designed to compensate for the above deficiencies.

- ***Messages***
  - ***Message Format***
  - ***Error Reporting Messages***
  - ***Query Messages***

Figure 4.54: 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)

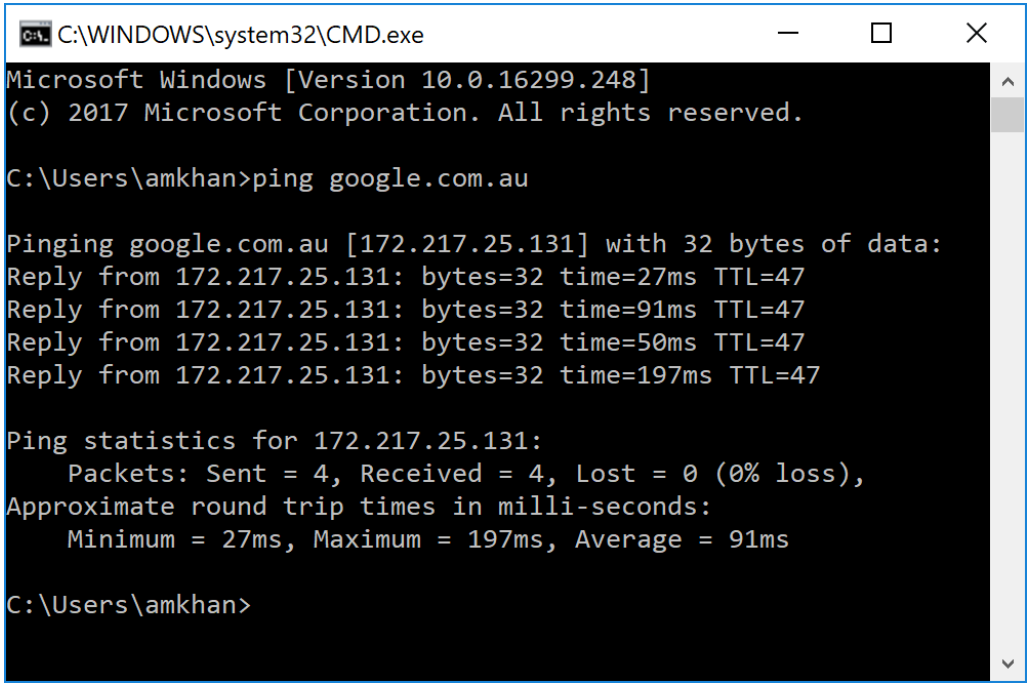
Query messages

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.

Example 4.13

- *One of the tools that a host can use to test the liveliness of another host is the **ping** program.*
- *The **ping** program takes advantage of the **ICMP** echo **request** and echo **reply** messages.*
- *A host can send an echo request (type 8, code 0) message to another host, which, if alive, can send back an echo reply (type 0, code 0) message.*
- *To some extent, the ping program can also measure the reliability and congestion of the router between the two hosts by sending a set of request-reply messages.*



Example 4.14

- The ***traceroute*** program in UNIX or
- The ***tracert*** in Windows

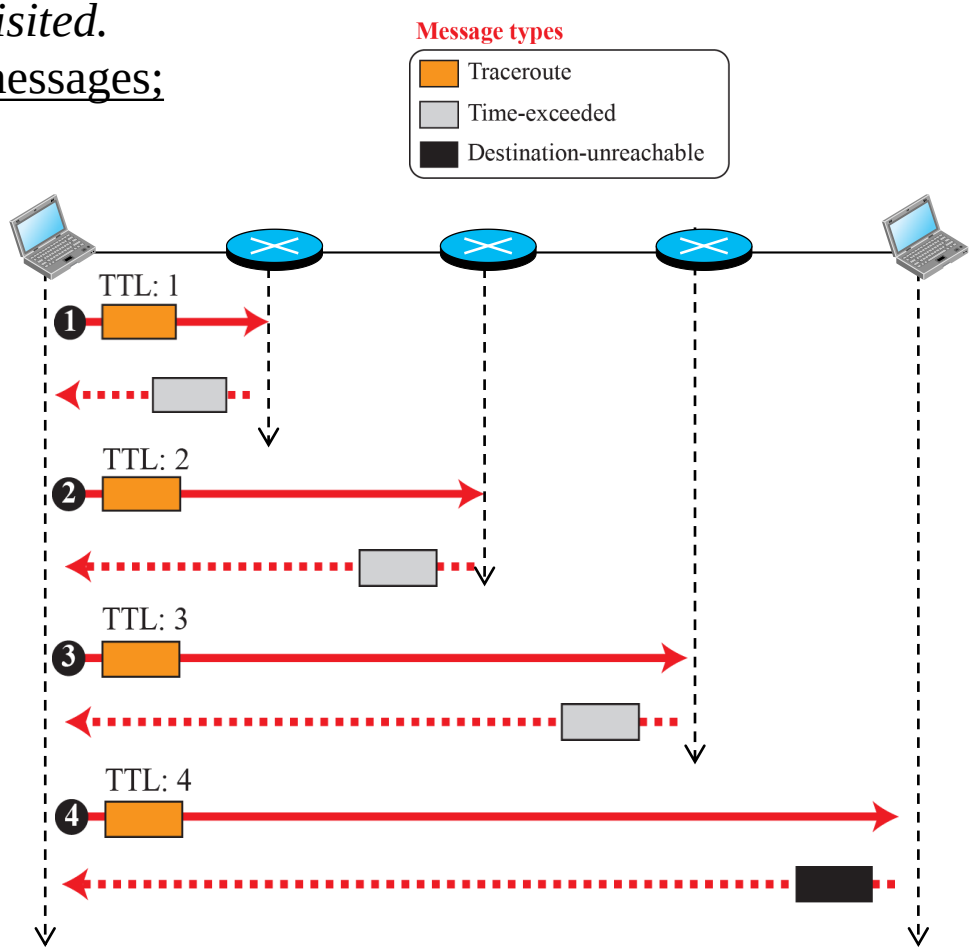
It can be used to trace the path of a packet from a source to the destination. It can find the IP addresses of all the routers that are visited along the path. The program is usually set to check for the ***maximum of 30 hops*** *(routers) to be visited.*

The ping program gets help from two query messages;

two error-reporting messages:

- **time-exceeded and**
- **destination-unreachable**

Figure 4.55 shows an example in which;
n = 3 or three intermediate routers.



Example 4.14 (continued)

- *The **traceroute** program also sets a timer to find the round-trip time for each router and the destination.*
- *Most **traceroute** programs send three messages to each device, with the same TTL value, to be able to find a better estimate for the round-trip time.*
- *The following output shows an example of a **traceroute** program, which uses **three probes** for each device and gets **three** RTTs.*

```
$ traceroute printers.com
traceroute to printers.com (13.1.69.93), 30 hops max, 38 byte packets
1  route.front.edu      (153.18.31.254)   0.622 ms   0.891 ms   0.875 ms
2  ceneric.net          (137.164.32.140)  3.069 ms   2.875 ms   2.930 ms
3  satire.net           (132.16.132.20)   3.071 ms   2.876 ms   2.929 ms
4  alpha.printers.com   (13.1.69.93)      5.922 ms   5.048 ms   4.922 ms
```

4-3 UNICAST ROUTING

- In an internet, the goal of the network layer is to deliver a datagram from its **source** to its **destination** or **destinations**.
- *If a datagram is destined for only one destination (one-to-one delivery), we have unicast routing.*
- This section discusses unicast routing.

4.3.1 General Idea

- *In unicast routing, **a packet is routed, hop by hop, from its source to its destination** by the help of forwarding tables.*
- *The **source host needs no forwarding table** because it delivers its packet to the **default router/gateway** in its local network.*
- *The **destination host needs no forwarding table either** because it **receives the packet from its default router** in its local network.*
- *This means that only the routers that glue together the networks in the internet need forwarding tables.*

  - ***An Internet as a Graph***
  - ***Least-Cost Routing***
  - ***Least-Cost Trees***

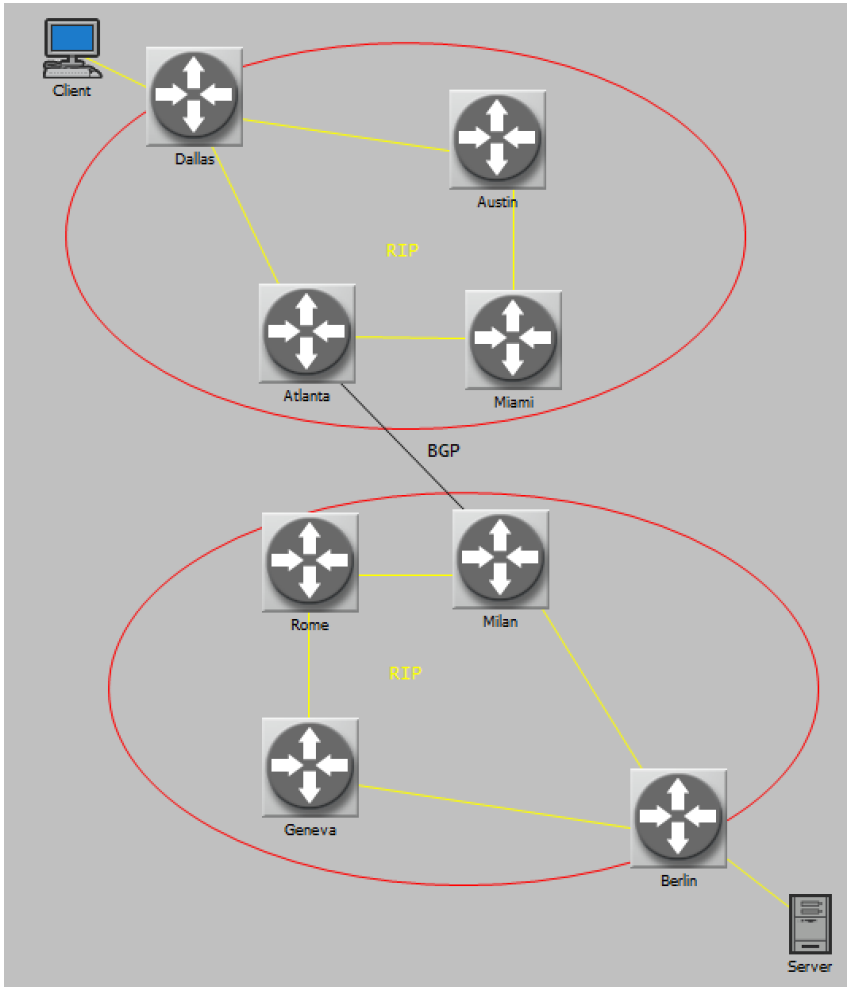
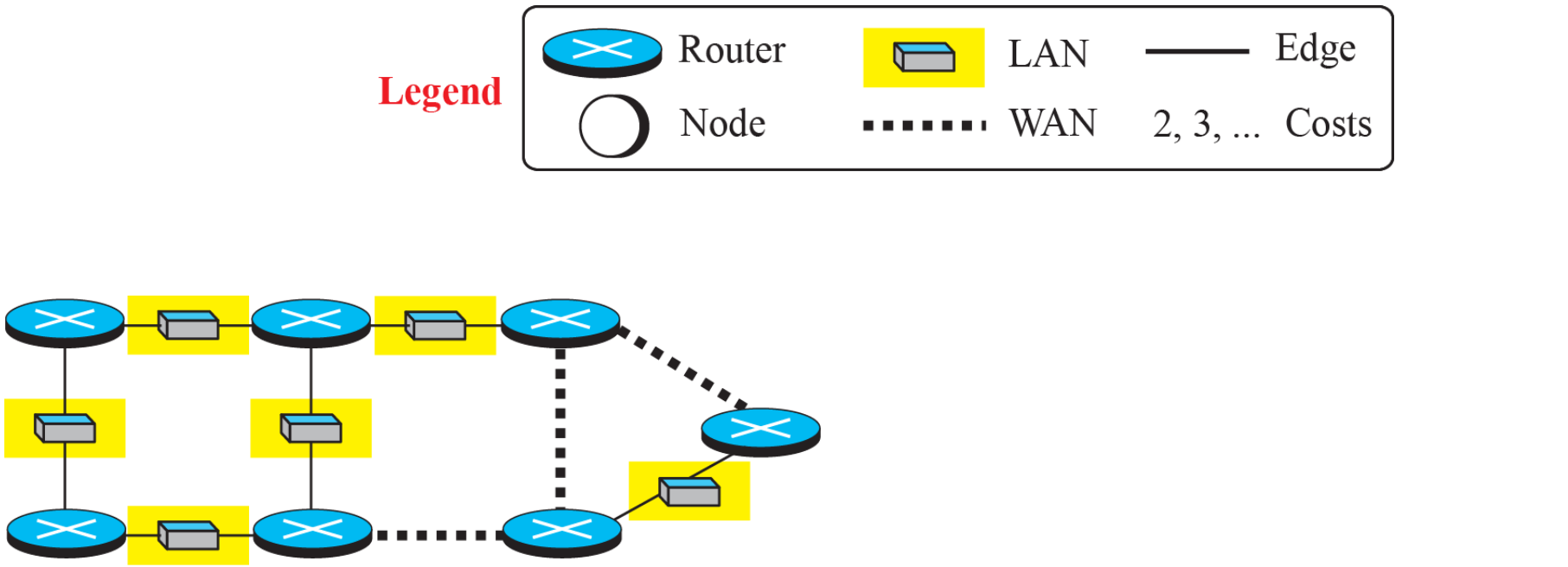


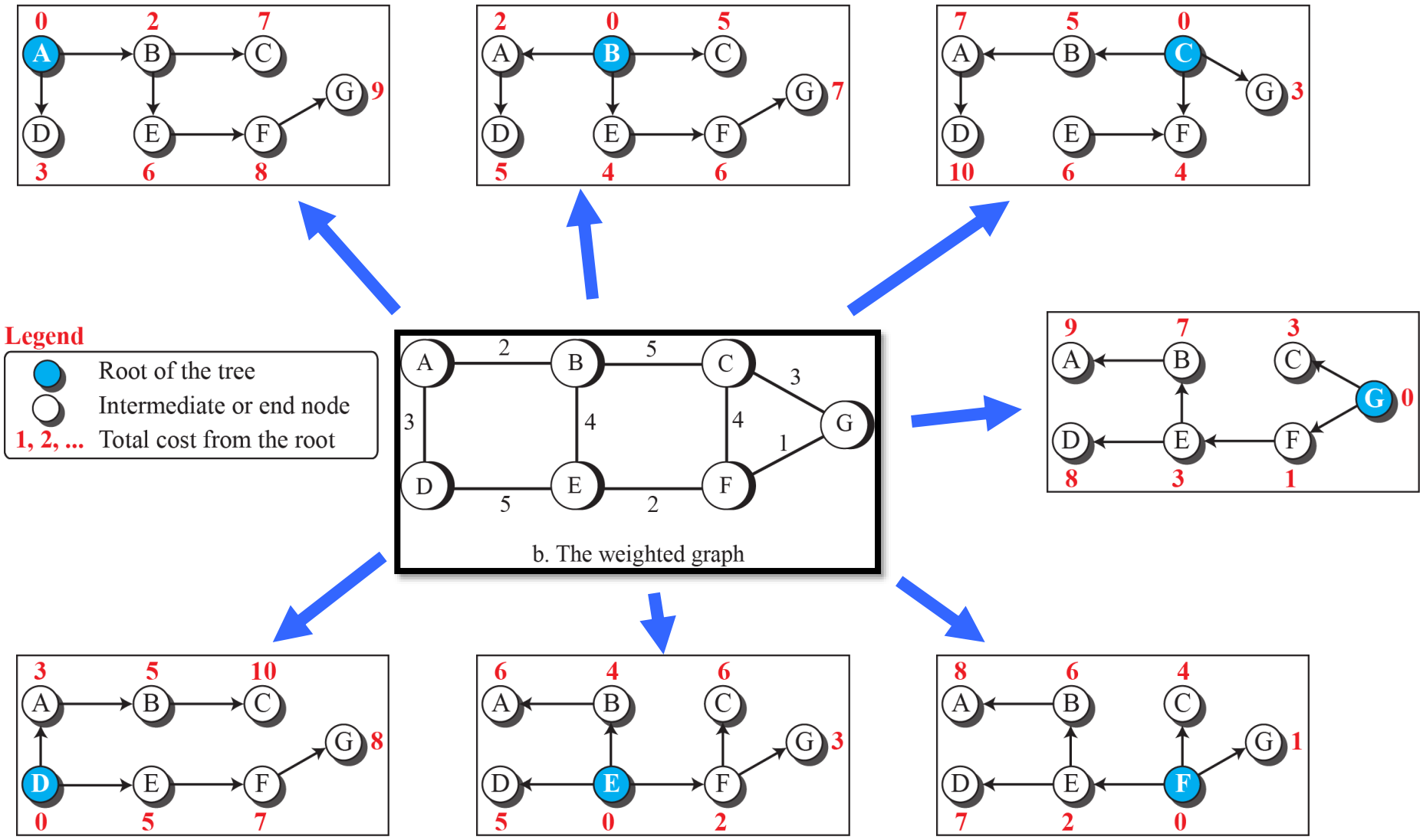
Figure 4.56: An internet and its graphical representation



a. An internet

b. The weighted graph

Figure 4.57: Least-cost trees for nodes in the internet of Figure 4.56



4.3.2 Routing Algorithm

- *After discussing the general idea behind least-cost trees and the forwarding tables can be developed from them.*
- *Lets concentrate on the routing algorithms.*
- *Several routing algorithms have been designed in the past. The differences between these methods are in the way they interpret the least cost and the way they create the least-cost tree for each node.*
- *In this section lets discuss the common algorithm; later we show how a routing protocol in the Internet implements one of these algorithms.*

4.3.2 (continued)

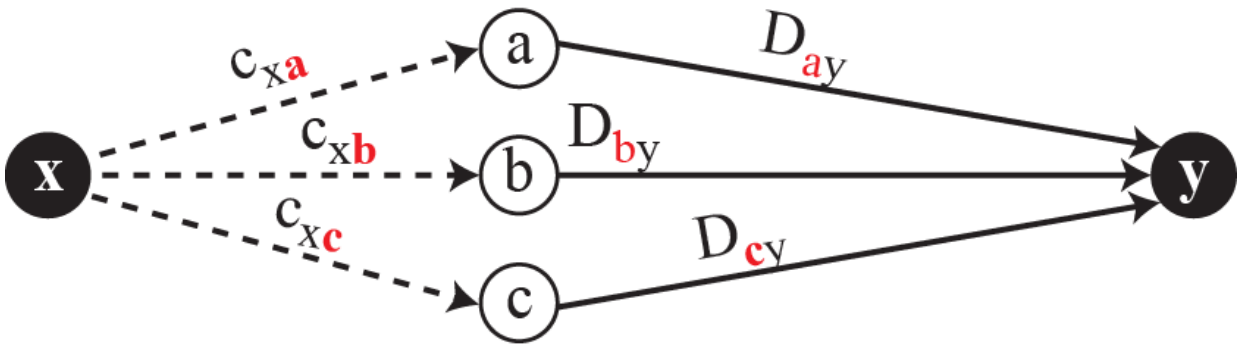
Distance-Vector Routing

- Bellman-Ford Equation
- Distance-Vectors & Routing Algorithm
- Count to Infinity
- Two-Node Loop
  - Split Horizon
  - Poisoned Reverse
- Three-Node Instability

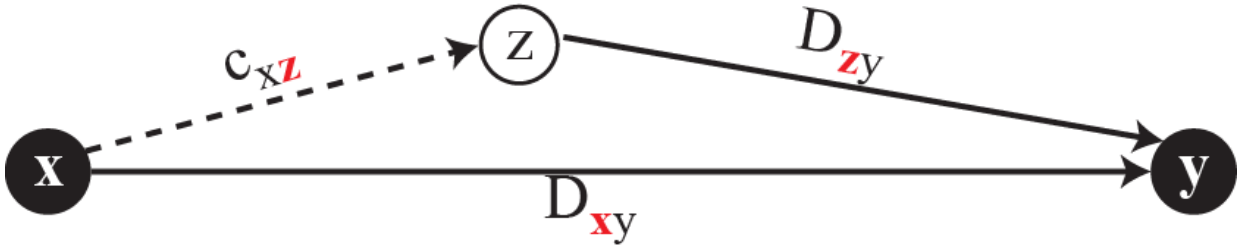
4.3.2 (continued)

- **Link-State Routing**
  - Link-State Database (LSDB)
  - Least-Cost Trees (Dijkstra's Algorithm)
- **Path-Vector Routing**
  - Spanning Trees
  - Creation of Spanning Trees
  - Path-Vector Algorithm

Figure 4.58: Graphical idea behind Bellman-Ford equation

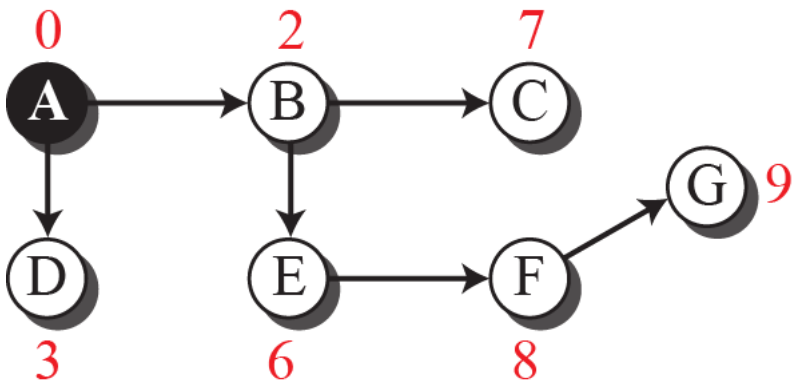


a. General case with three intermediate nodes



b. Updating a path with a new route

Figure 4.59: The distance vector corresponding to a tree



a. Tree for node A

	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

b. Distance vector for node A

Figure 4.60: The first distance vector for an internet

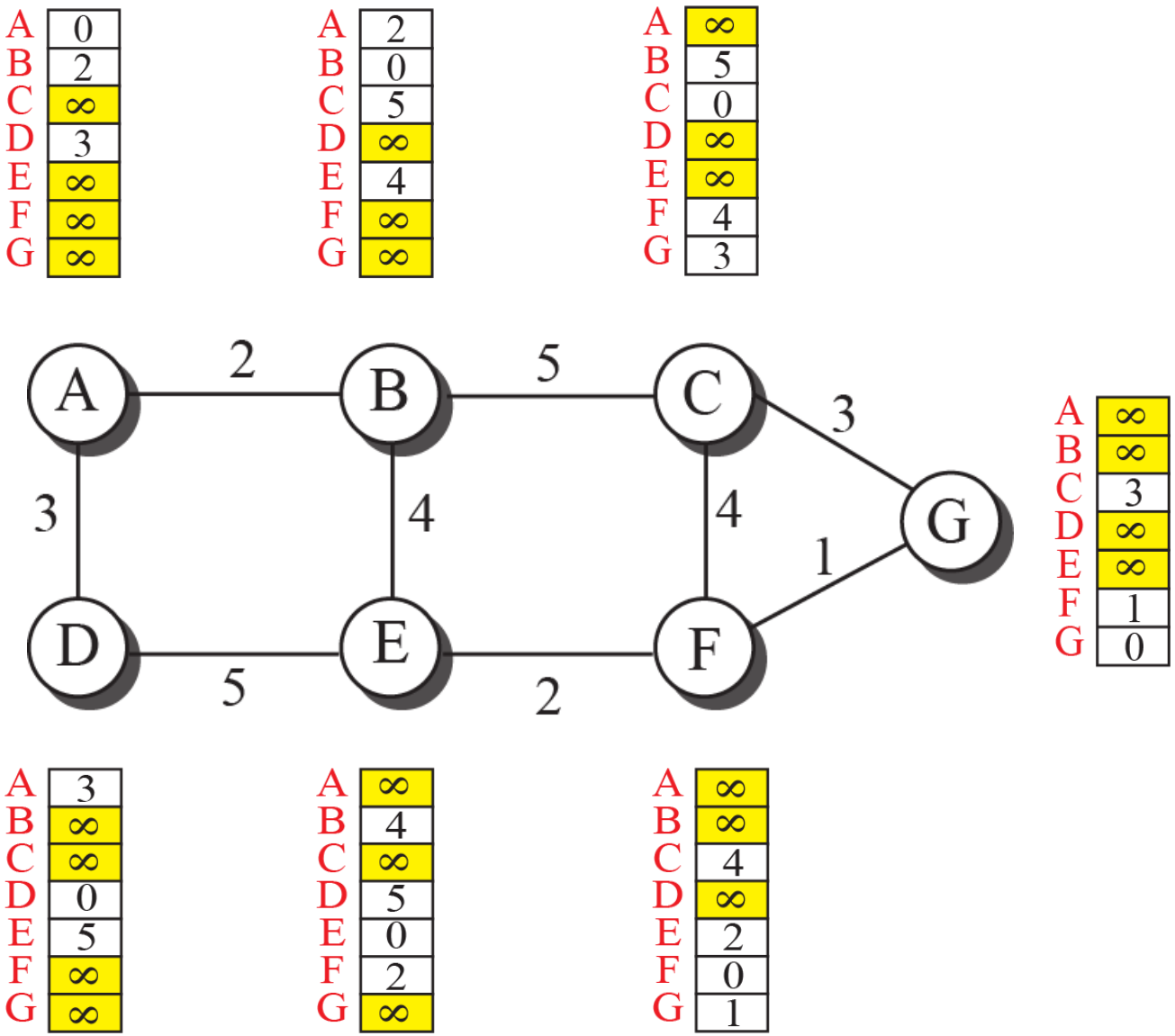
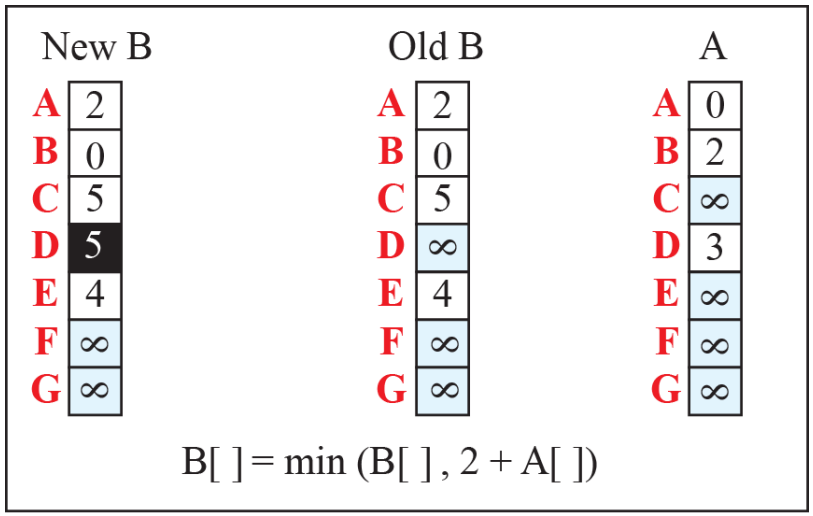
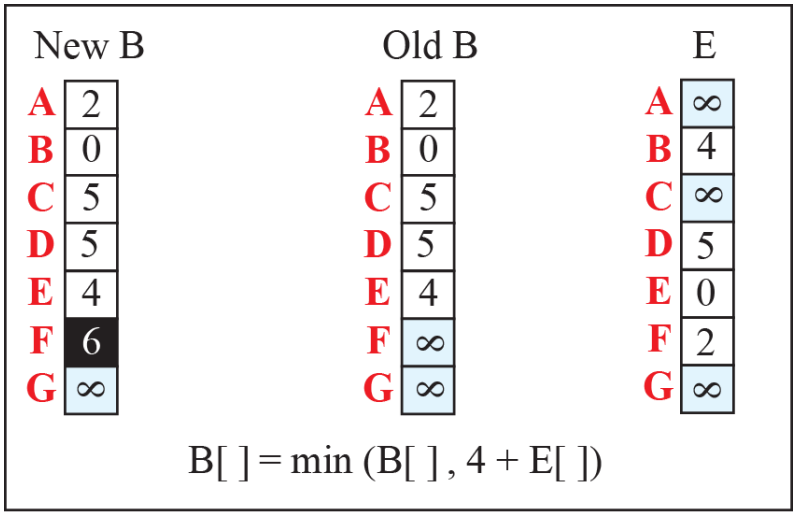


Figure 4.61: Updating distance vectors



a. First event: B receives a copy of A's vector.

Note:
X[ ]: the whole vector



b. Second event: B receives a copy of E's vector.

Figure 4.62: Two-node instability

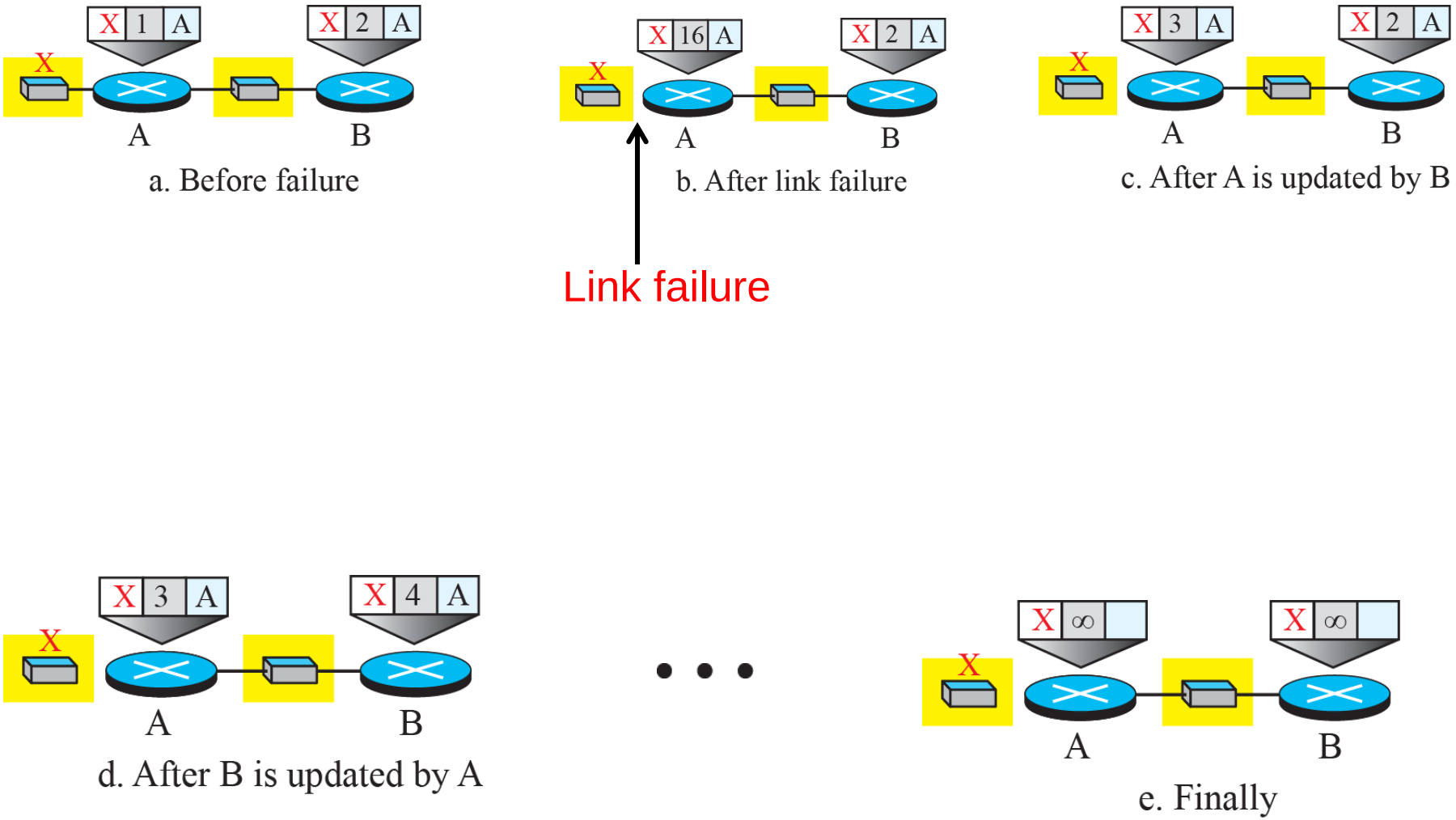
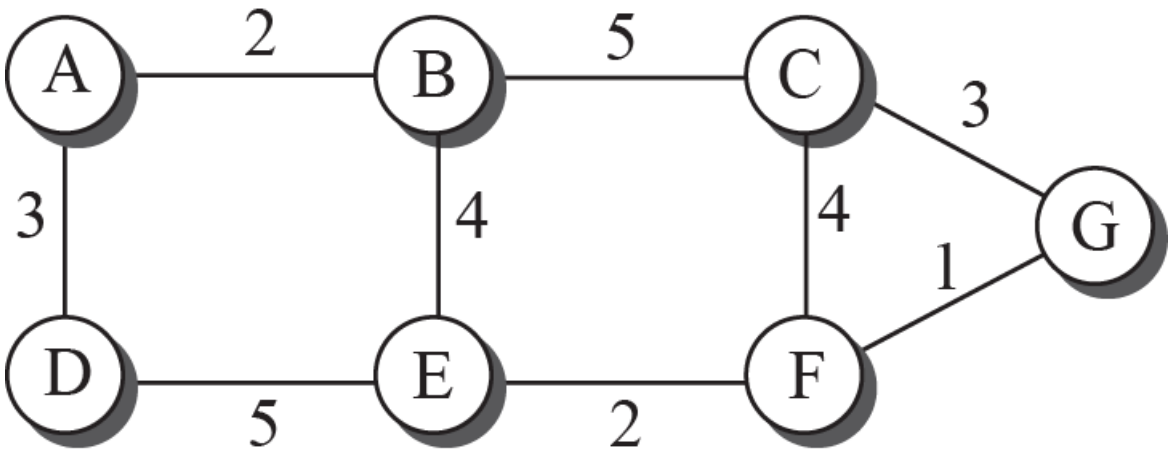


Figure 4.63: Example of a link-state database



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

Figure 4.64: LSPs created and sent out by each node to build LSDB

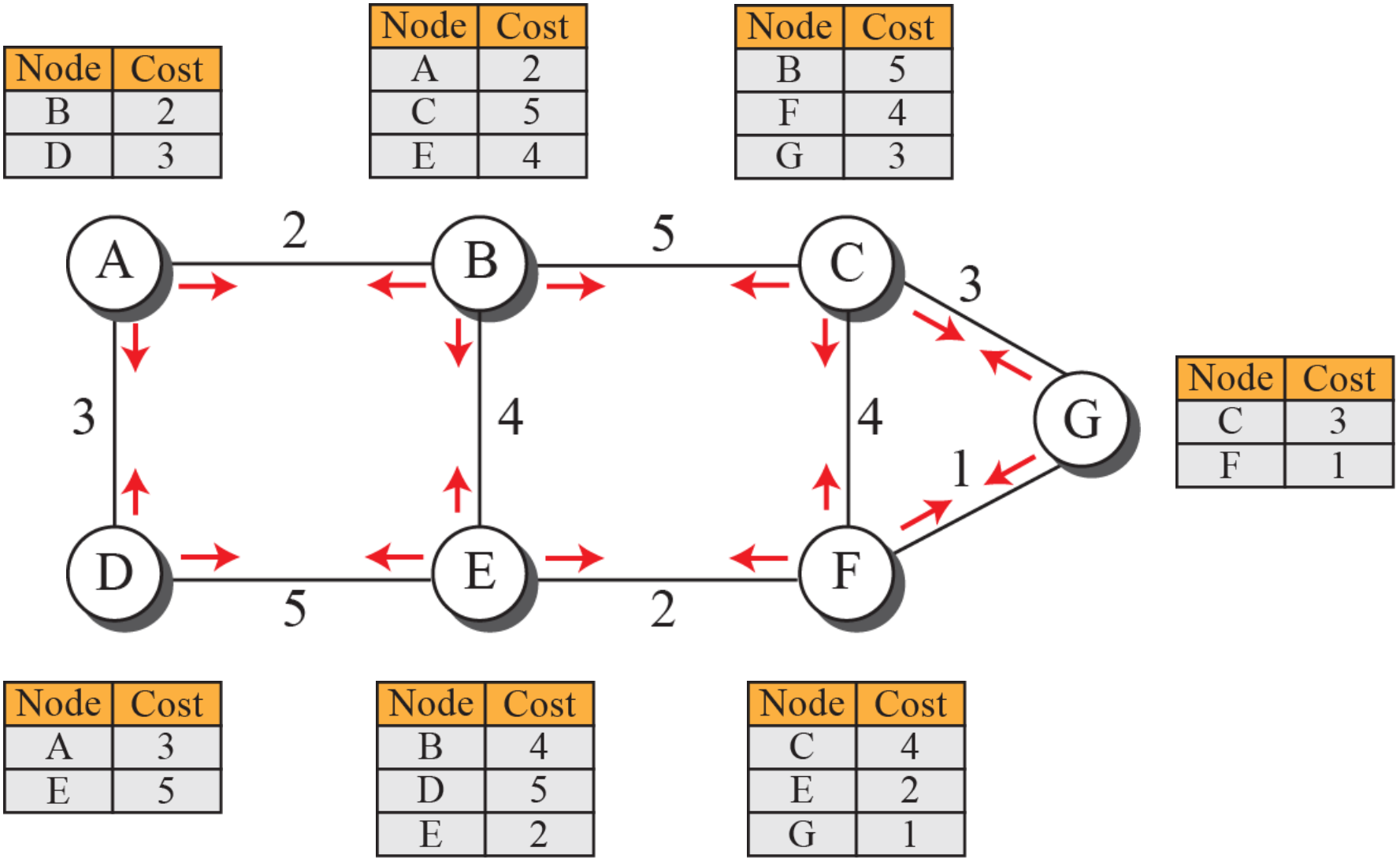


Figure 4.65: Least-cost tree

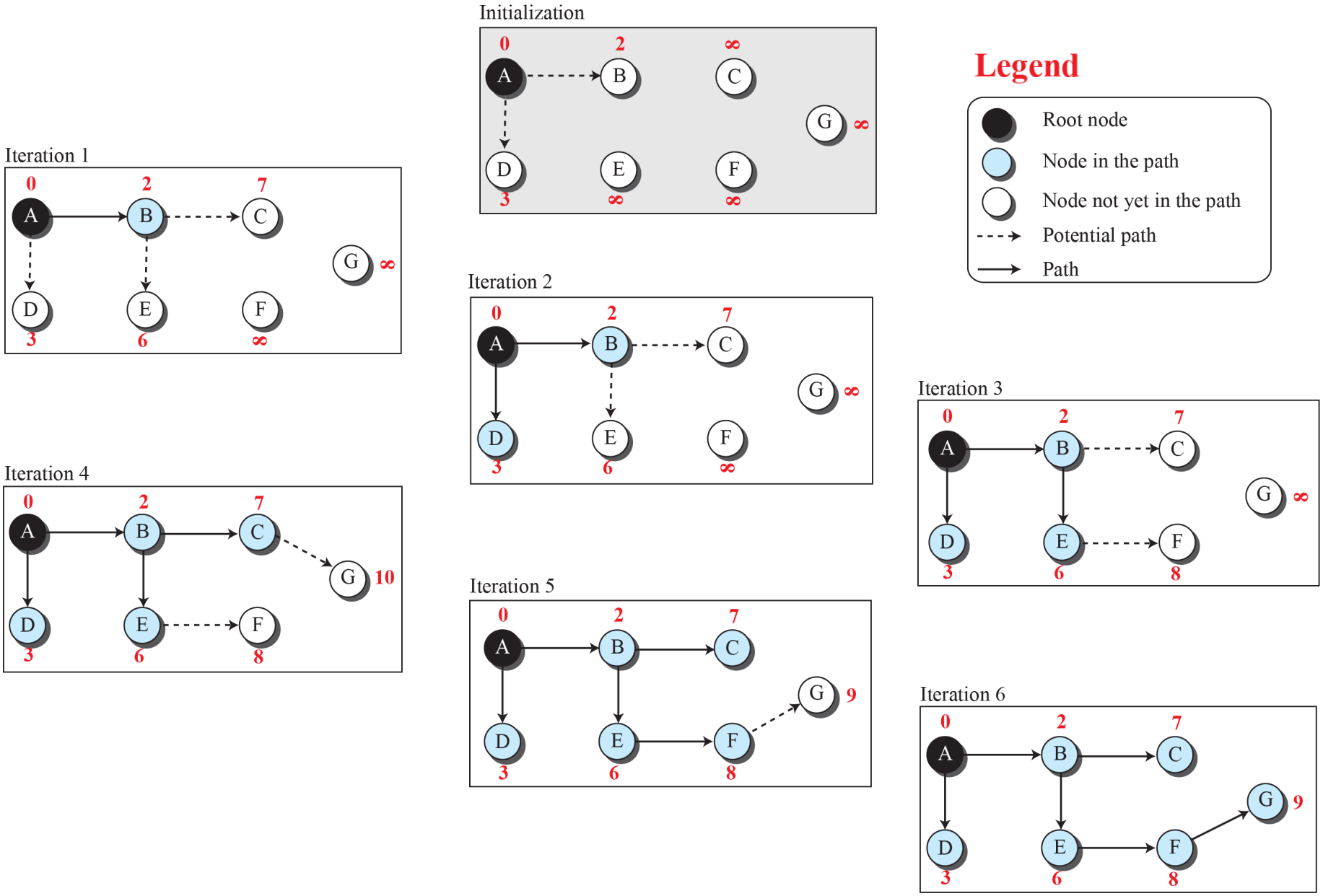


Figure 4.66: Spanning trees in path-vector routing

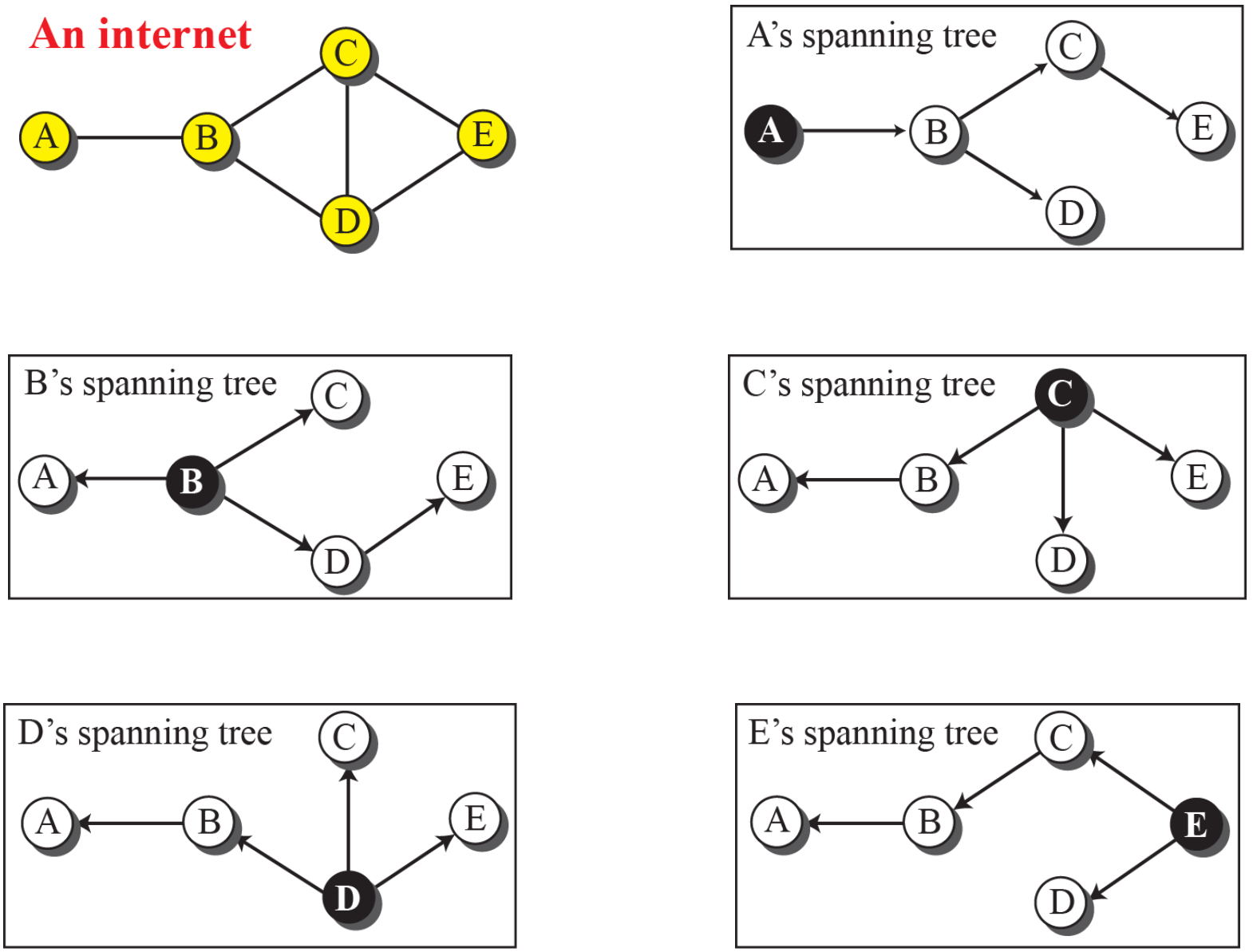


Figure 4.67: Path vectors made at booting time

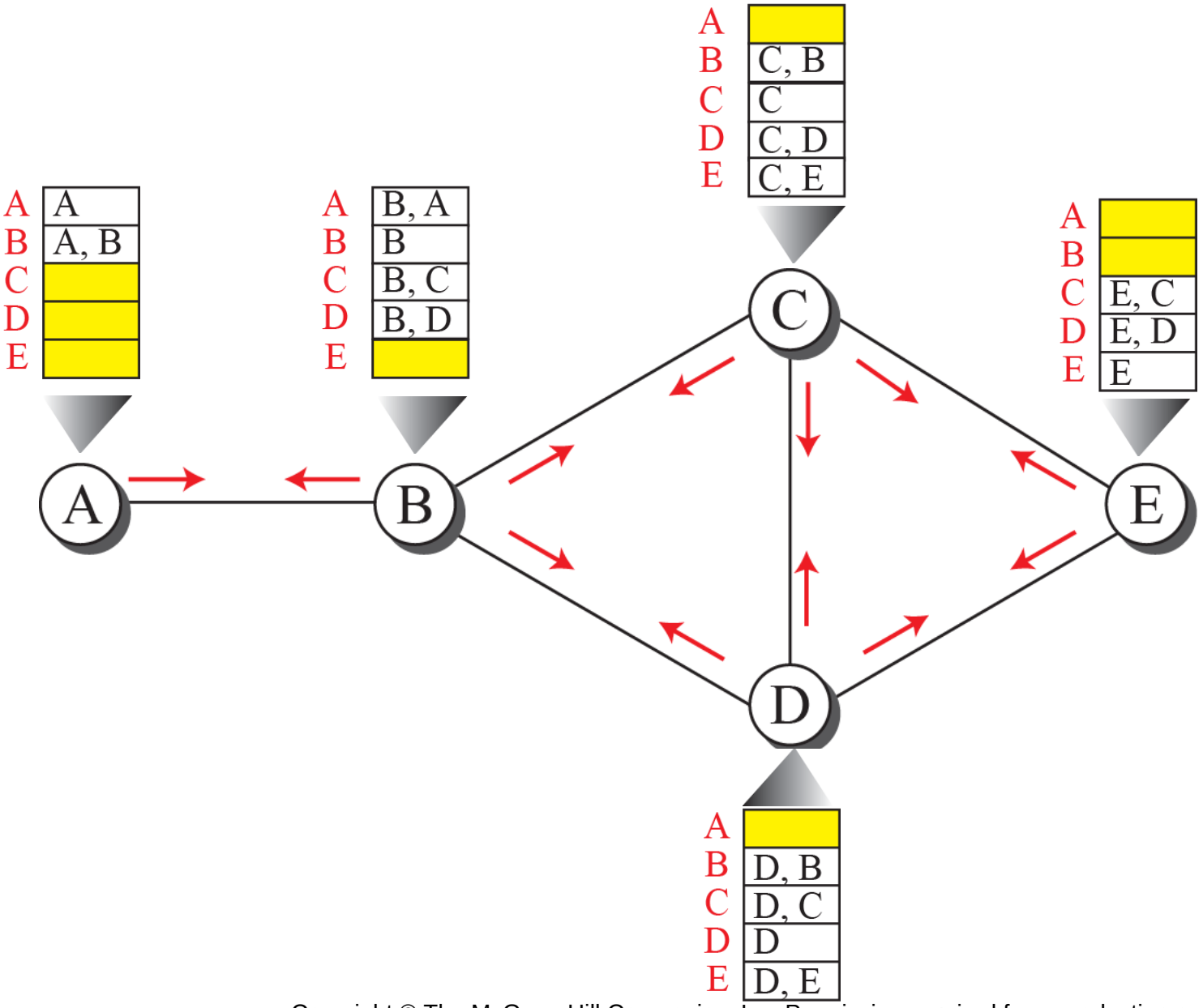
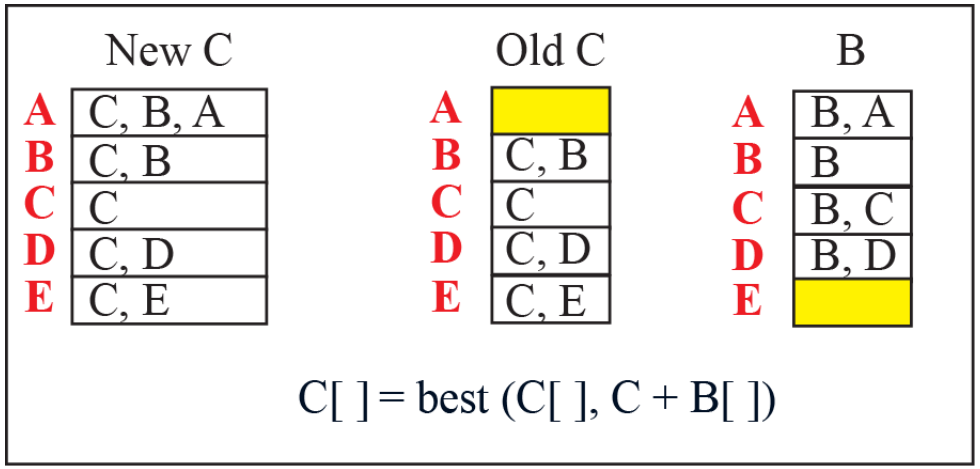
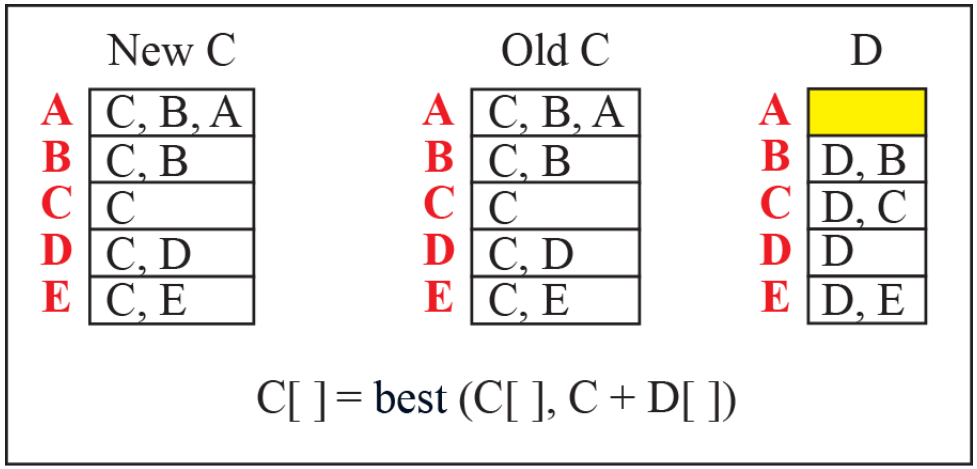


Figure 4.68: Updating path vectors



Event 1: C receives a copy of B's vector

Note:
X [ ]: vector X
Y: node Y



Event 2: C receives a copy of D's vector

4.3.3 Unicast Routing Protocols

- *Discuss unicast routing protocols used in the Internet.*
- *Two protocols discuss here are based on the corresponding algorithms we discussed before, a protocol is more than an algorithm.*
- *A protocol needs to define*
  - ✓ *its domain of operation,*
  - ✓ *the messages exchanged,*
  - ✓ *communication between routers, and*
  - ✓ *interaction with protocols in other domains.*
- *Intro and discuss two protocols used in the Internet:*
  - ✓ *RIP &*
  - ✓ *OSPF*

4.3.3 (continued)

- ***Internet Structure***
  - ***Hierarchical Routing***
  - ***Autonomous Systems***
- ***Routing Information Protocol (RIP)***
  - ***Hop Count***
  - ***Forwarding Tables***
  - ***RIP Implementation***
  - ***Performance***
- ***Open Shortest Path First (OSPF)***
  - ***Metric***
  - ***Forwarding Tables***
  - ***Areas***
  - ***Link-State Advertisement***
  - ***OSPF Implementation***
  - ***Performance***

Figure 4.69: Internet structure

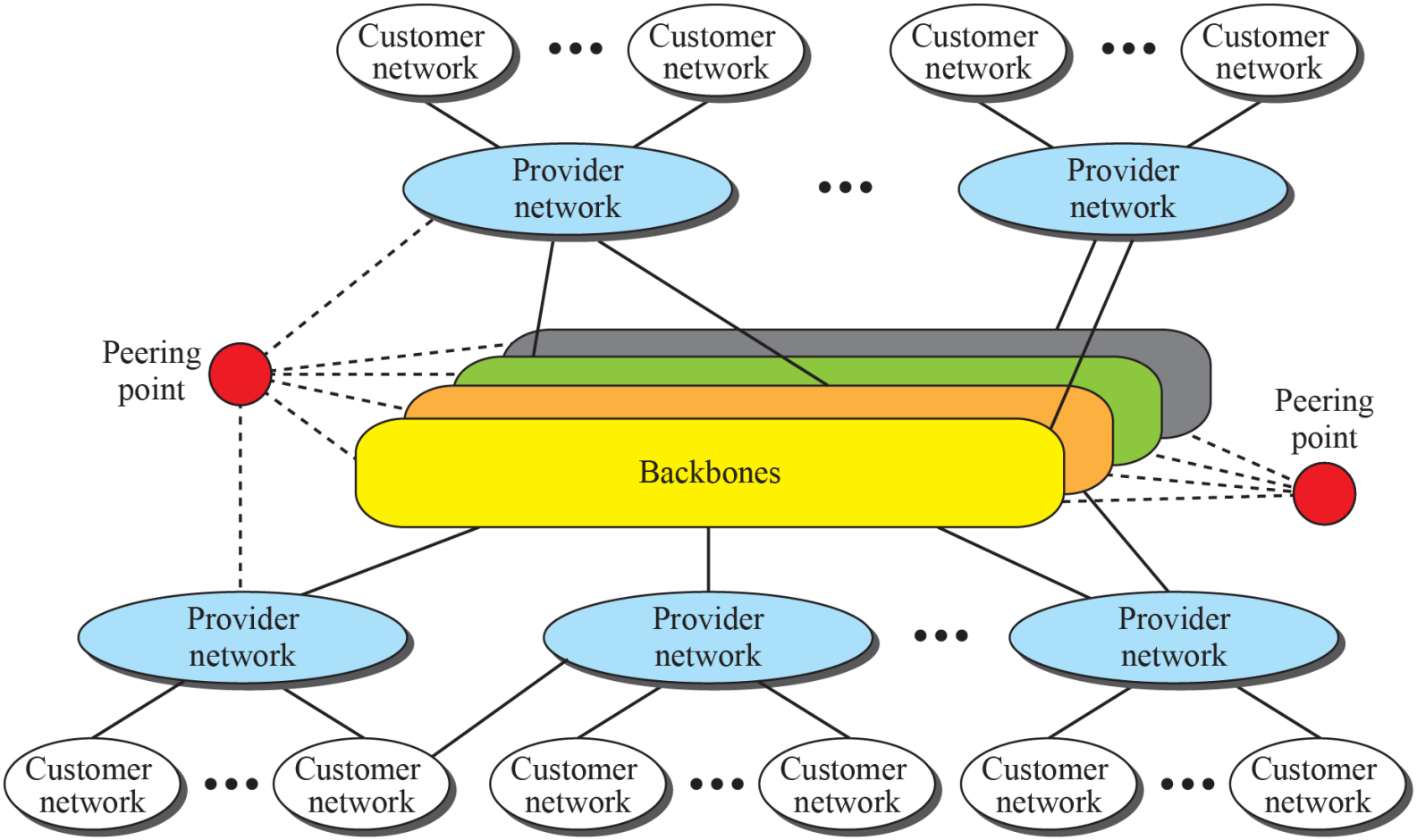
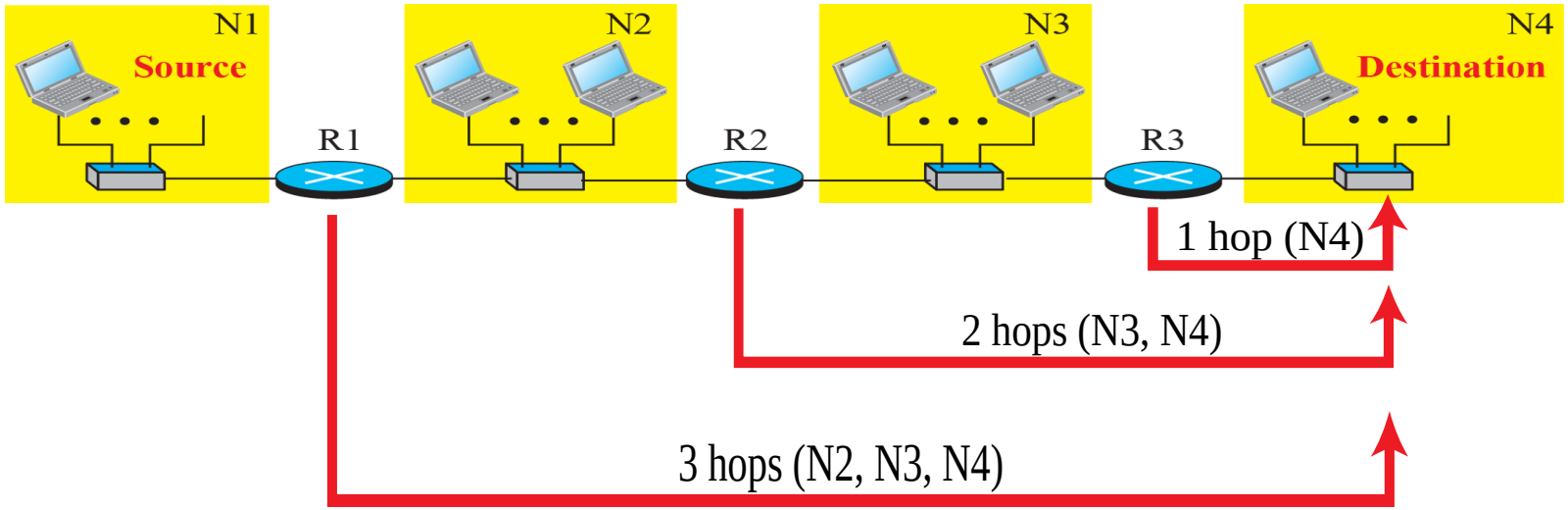
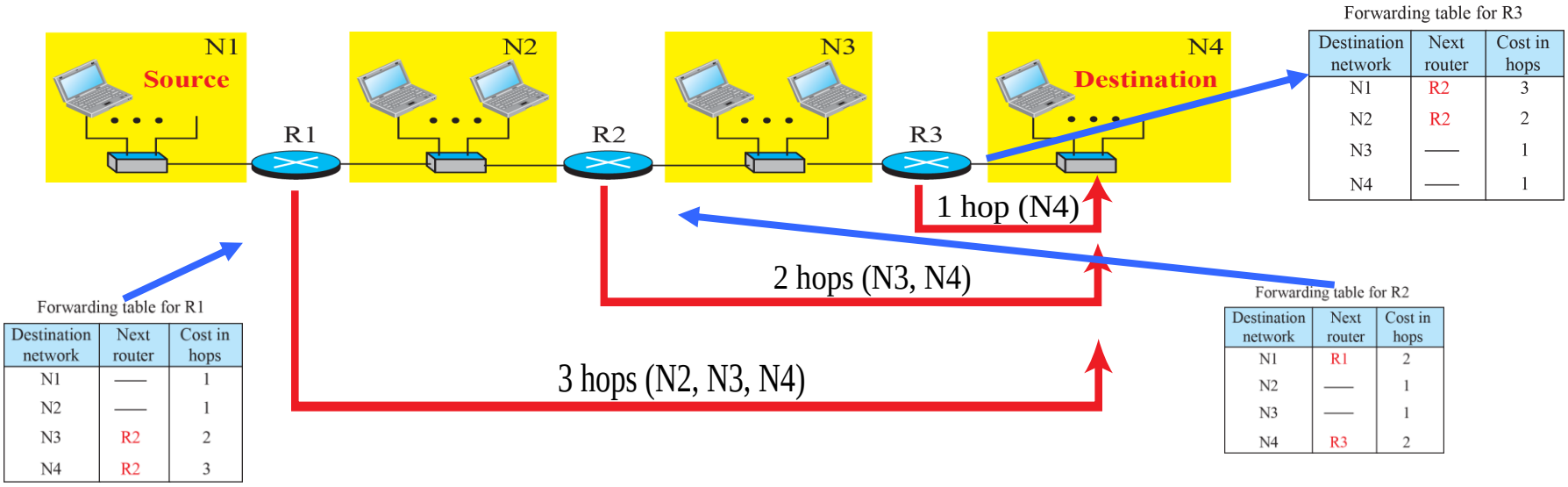


Figure 4.70: Hop counts in RIP



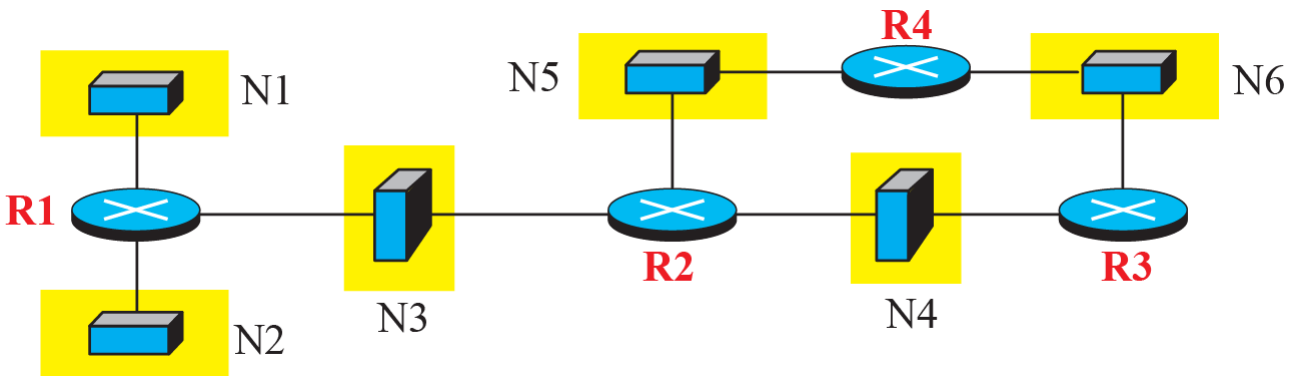
- *RIP is based on the **distance vector routing algorithm**, one of several common routing algorithms that routers use to **dynamically calculate the cost or metric of each possible path** through an internetwork.*
- *RIP is designed for **intra-domain routing** (routing within a flat routing space or routing domain).*
- *Routing tables in RIP-enabled routers are calculated on the **basis of the number of hops** to the destination network.*
- *RIP routers **do not use** other routing metrics such as load, bandwidth, latency, or Maximum Transmission Unit (MTU) in calculating routing costs.*
- *The **routing table of a RIP router contains the cost in hops of every path to every destination** network in the internetwork.*

Figure 4.70: Hop counts in RIP Figure 4.71: Forwarding tables



- *RIP-enabled routers broadcast their routing tables updates every 30 seconds over UDP port **520** using RIP advertisements.*
- *Adds some overhead to network traffic as broadcast information but propagates only in local network and received only by routers that have a routing interface to the LAN.*
- *RIP does not support multipath routing. If a routing table has multiple routes for a single network ID, RIP stores the route with the **lowest** metric (number of hops to destination).*

Example 4.15: Figure 4.73: Example of an autonomous system using RIP (Part I)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops

R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1

R2

Des.	N. R.	Cost
N3	———	1
N4	———	1
N5	———	1

R3

Des.	N. R.	Cost
N4	———	1
N6	———	1

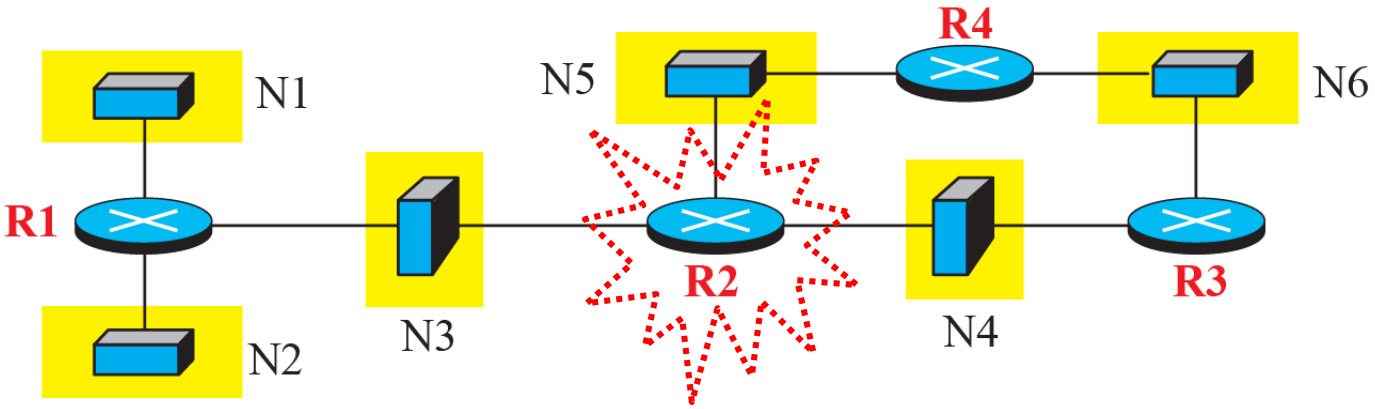
R4

Des.	N. R.	Cost
N5	———	1
N6	———	1

Forwarding tables after all routers booted

- *Figure 4.73 shows a more realistic example of the operation of RIP in an autonomous system.*
- *First, the figure shows all forwarding tables after all routers have been booted.*
- *Then we show changes in some tables when some update messages have been exchanged.*
- *Finally, we show the stabilized forwarding tables when there is no more change.*

Figure 4.73: Example of an autonomous system using RIP (Part II)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops
: New route
: Old route

New R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1
N4	R2	2
N5	R2	2

Old R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2 Seen by R1

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R3

Des.	N. R.	Cost
N3	R2	2
N4	—	1
N5	R2	2
N6	—	1

Old R3

Des.	N. R.	Cost
N4	—	1
N6	—	1

R2 Seen by R3

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	—	1
N6	—	1

Old R4

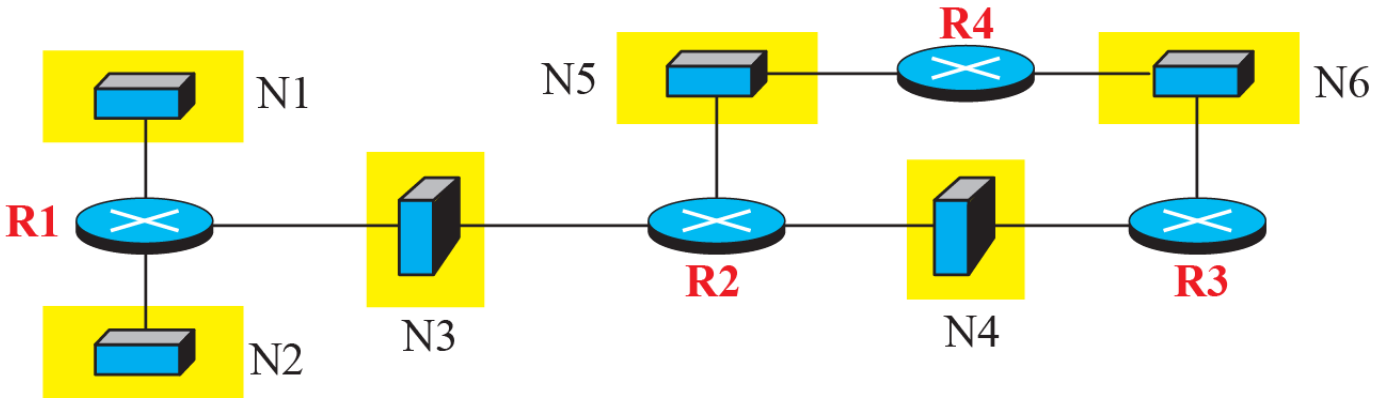
Des.	N. R.	Cost
N5	—	1
N6	—	1

R2 Seen by R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

Changes in the forwarding tables of R1, R3, and R4 after they receive a copy of R2's table

Figure 4.73: Example of an autonomous system using RIP (Part III)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops

Forwarding tables for all routers after they have been stablized

Final R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1
N4	R2	2
N5	R2	2
N6	R2	3

Final R2

Des.	N. R.	Cost
N1	R1	2
N2	R1	2
N3	———	1
N4	———	1
N5	———	1
N6	R3	2

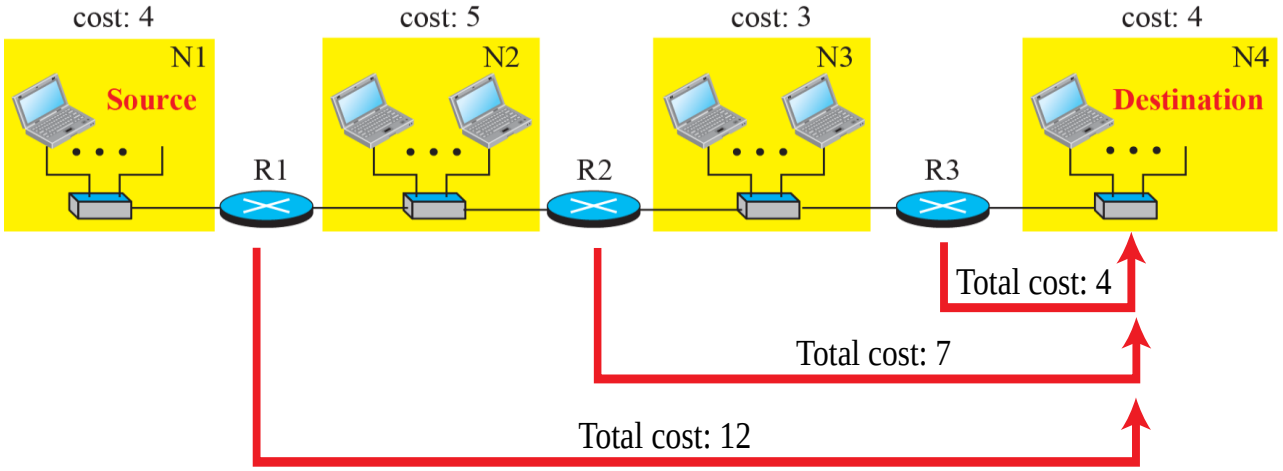
Final R3

Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	———	1
N5	R2	2
N6	———	1

Final R4

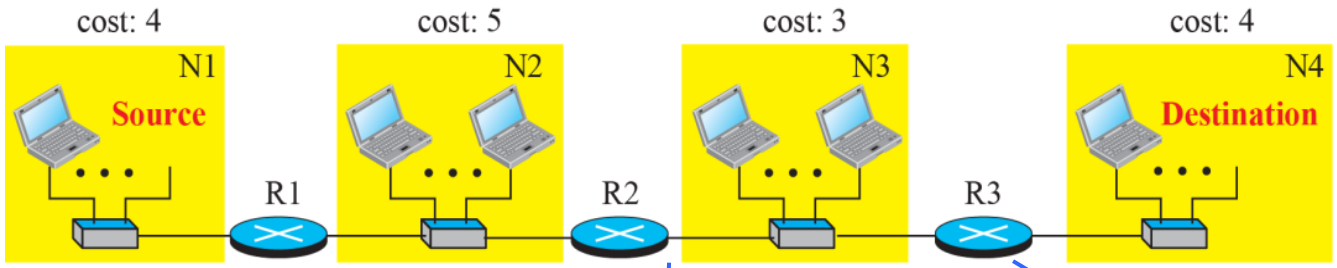
Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	R2	2
N5	———	1
N6	———	1

Figure 4.74: Metric in OSPF



- *Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks.*
- *OSPF uses a **link state routing (LSR) algorithm** and falls into the group of interior gateway protocols (IGPs), operating within a single autonomous system (AS).*
- *OSPF **detects changes in the topology**, such as **link failures**, and **converges** on a new loop-free routing structure within seconds.*
- *OSPF computes the **shortest-path tree** for each route using a method based on **Dijkstra's** algorithm. (Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by **Dijkstra** in 1956.)*
- *The OSPF routing policies for constructing a route table are **governed by link metrics associated with each routing interface.***
- ***Cost factors may be the distance of a router (round-trip time), data throughput of a link, or link availability and reliability, expressed as simple unit-less numbers.** OSPF provides a dynamic process of traffic load balancing between routes of equal cost.*

Figure 4.75: Forwarding tables in OSPF



The internet from previous figure

Forwarding table for R1

Destination network	Next router	Cost
N1	——	
N2	——	
N3	R2	8
N4	R2	12

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	——	5
N3	——	3
N4	R3	7

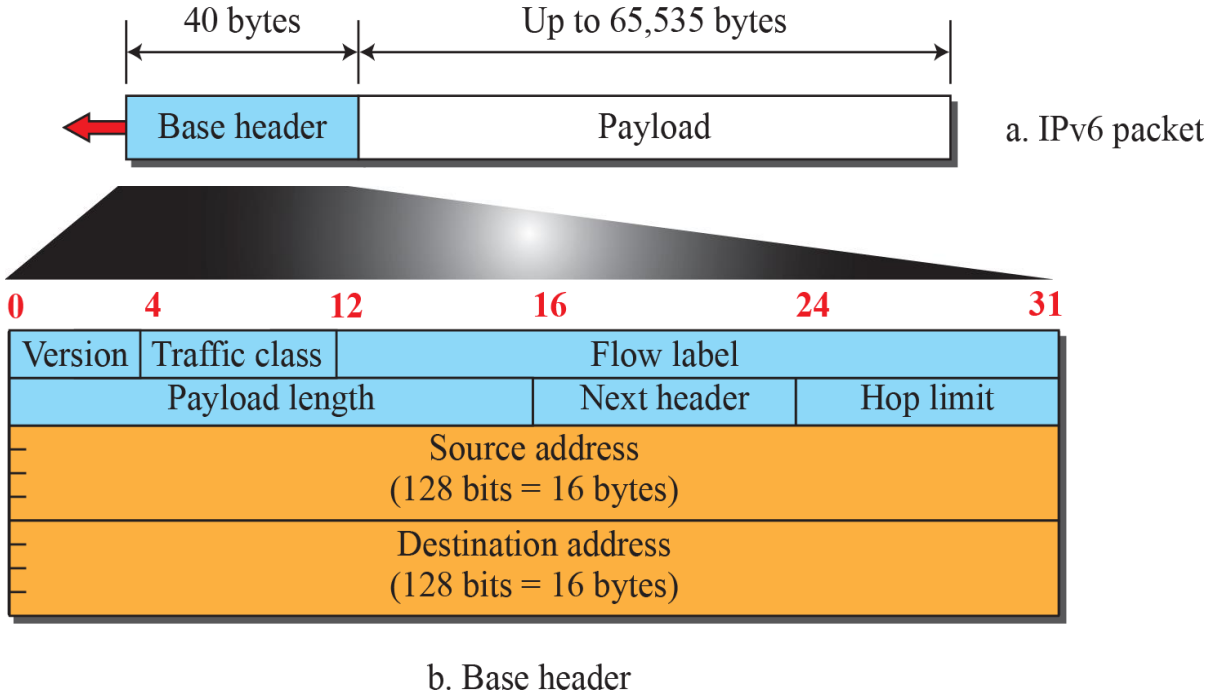
Forwarding table for R3

Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	——	3
N4	——	4

4-5 NEXT GENERATION IP

- *The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP protocol in the early 1990s.*
- *The new version, which is called Internet Protocol version 6 (IPv6) or IP new generation (IPng) was a proposal to augment the address space of IPv4*
- *Redesign the format of the IP packet and revise some auxiliary protocols such as ICMP.*

Figure 4.101: IPv6 datagram



4.5.1 Packet Format

The IPv6 packet is shown in Figure 4.101. Each packet is composed of a base header followed by the payload. The base header occupies 40 bytes, whereas payload can be up to 65,535 bytes of information.

- ***Concept of Flow and Priority in IPv6***
- ***Fragmentation and Reassembly***
- ***Extension Headers***

4.5.2 IPv6 Addressing

- *The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4.*
- *In this section, we show how the huge address space of IPv6 prevents address depletion in the future.*
- *We also discuss how the new addressing responds to some problems in the IPv4 addressing mechanism.*
- *An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.*

4.5.2 (continued)

- *Address Space*
- *Three Address Types*
- *Address Space Allocation*
  - *Global Unicast Addresses*
  - *Special Addresses*
  - *Other Assigned Blocks*

Table 4.7: Prefixes for assigned IPv6 addresses

Block prefix	*CIDR*	*Block assignment*	*Fraction*
0000 0000	0000::/8	Special addresses	1/256
001	**2000::/3**	**Global unicast**	**1/8**
1111 110	FC00::/7	Unique local unicast	1/128
1111 1110 10	FE80::/10	Link local addresses	1/1024
1111 1111	FF00::/8	Multicast addresses	1/256

Figure 4.103: Global unicast address

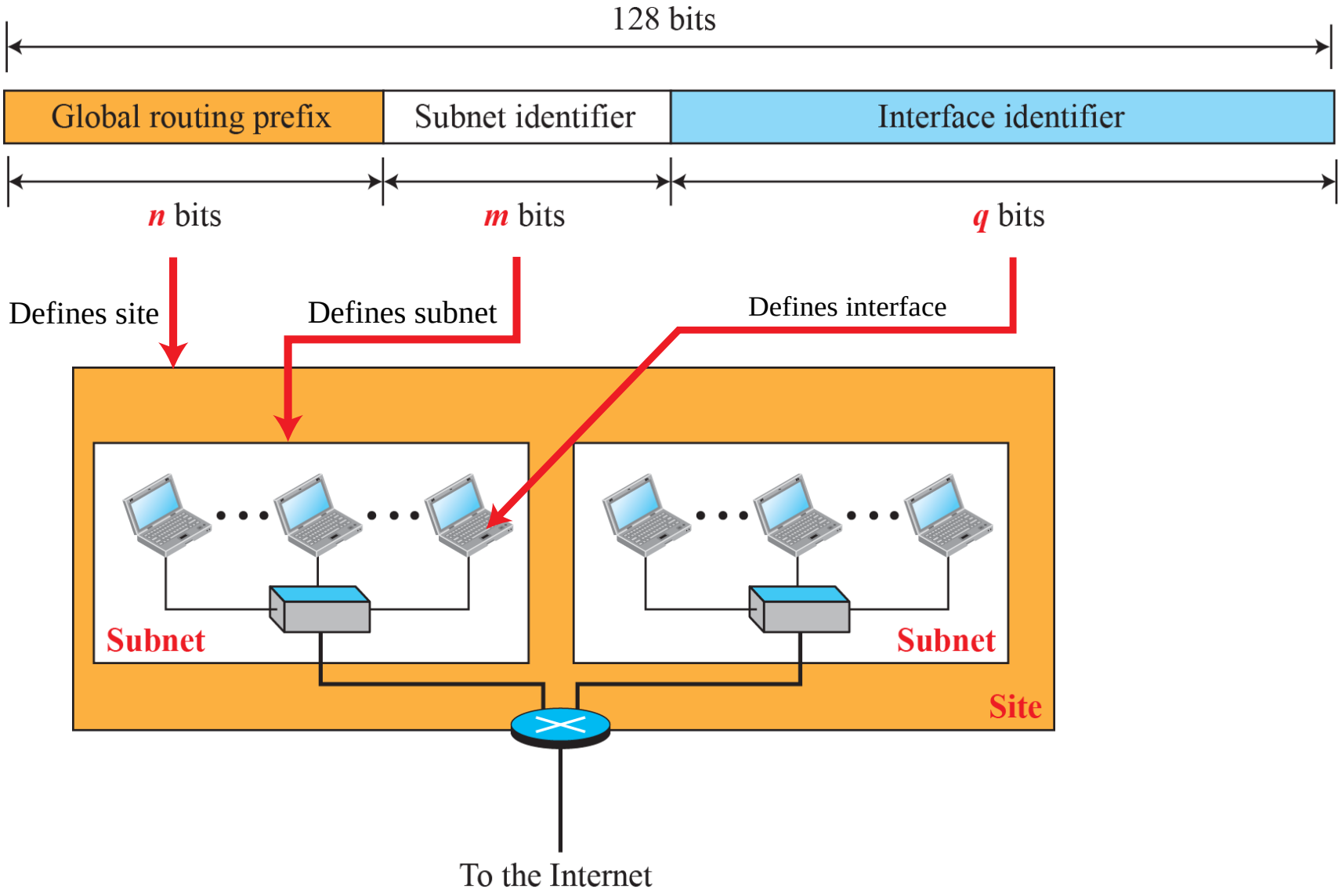
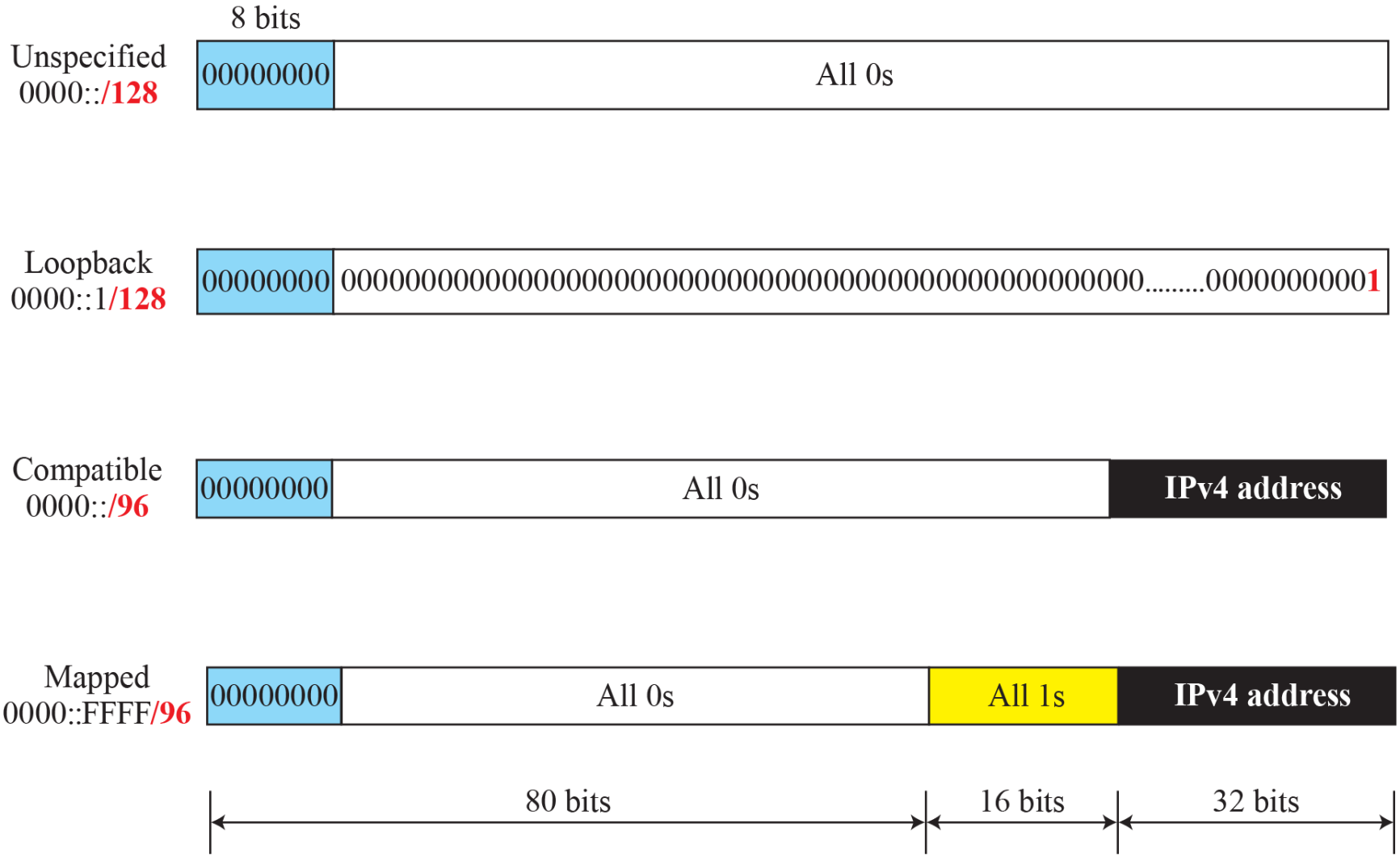


Figure 4.104: Special addresses

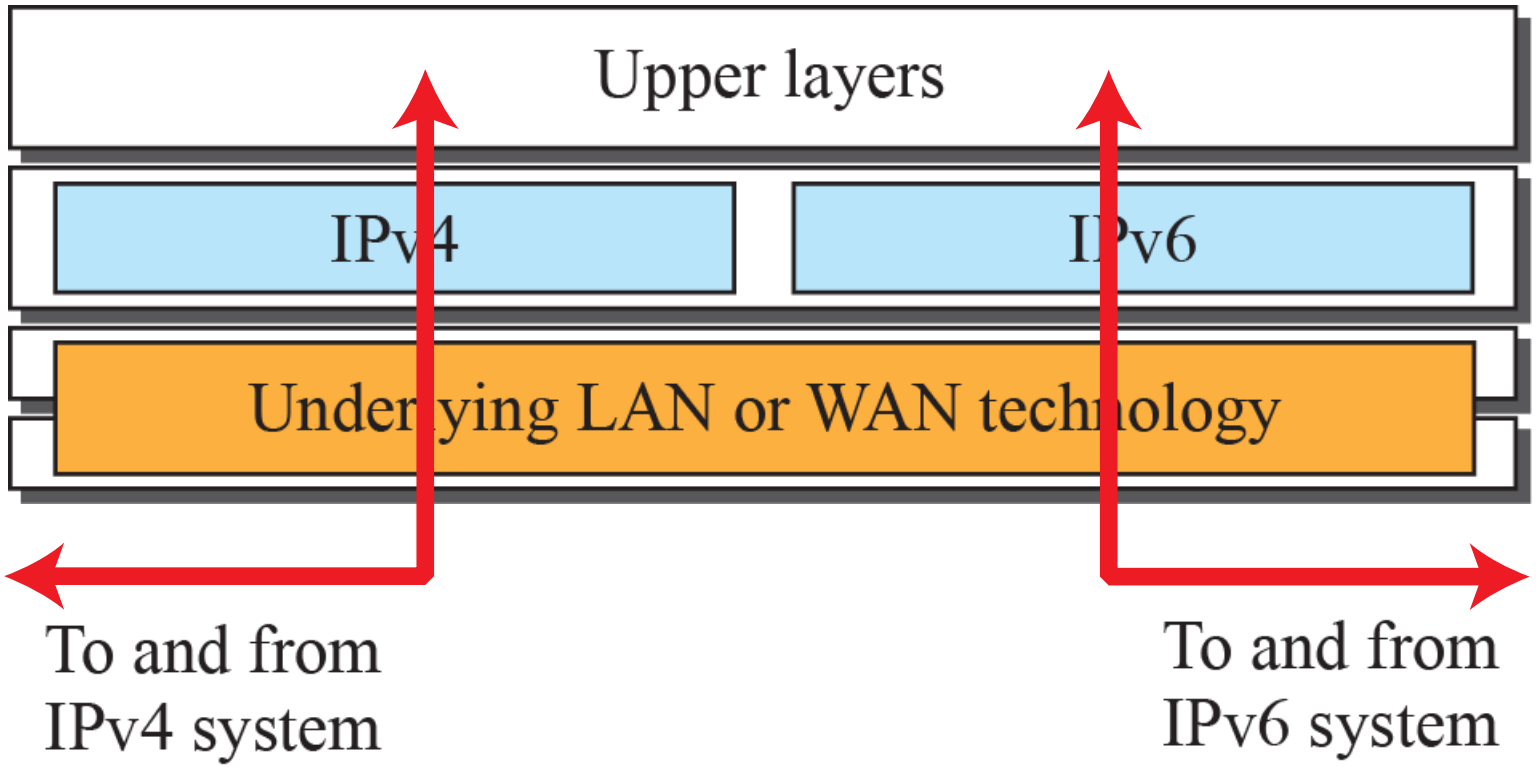


4.5.3 Transition from IPv4 to IPv6

Although we have a new version of the IP protocol, how can we make the transition to stop using IPv4 and start using IPv6? The first solution that comes to mind is to define a transition day on which every host or router should stop using the old version and start using the new version.

- ***Dual Stack***
  - ***Tunneling***
  - ***Header Translation***

Figure 4.106: Dual stack



Chapter 4: Summary

- ***The Internet is made of many networks (or links) connected through connecting devices, each of which acts as a router or as a switch. Two types of switching are traditionally used in networking: circuit switching and packet switching. The network layer is designed as a packet-switched network.***

- ***The network layer supervises the handling of packets by the underlying physical networks. The delivery of a packet can be direct or indirect. Two categories of forwarding are defined: forwarding based on the destination address of the IP datagram and forwarding based on the label attached to an IP datagram.***

- ***IPv4 is an unreliable connectionless protocol responsible for source-to-destination delivery. Packets in the IP layer are called datagrams. The identifiers used in the IP layer of the TCP/IP protocol suite are called the IP addresses. An IPv4 address is 32 bits long, divided into two parts: the prefix and the suffix. All addresses in the block have the same prefix; each address has a different suffix.***

Chapter 4: Summary (continued)

- *The Internet Control Message Protocol (ICMP) supports the unreliable and connectionless Internet Protocol (IP).*
- *To be able to route a packet, a router needs a forwarding table. Routing protocols are specific application programs that have the duty of updating forwarding tables.*
- *IPv6, the latest version of the Internet Protocol, has a 128-bit address space. IPv6 uses hexadecimal colon notation with abbreviation methods available.*
- *Multicasting is the sending of the same message to more than one receiver simultaneously. The Internet Group Management Protocol (IGMP) is involved in collecting local membership group information.*
- *IPv6, the latest version of the Internet Protocol, has a 128-bit address space. IPv6 uses hexadecimal colon notation with abbreviation methods available.*