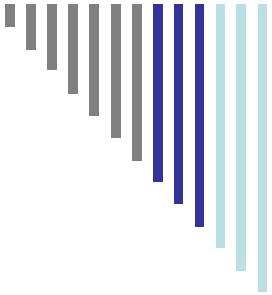
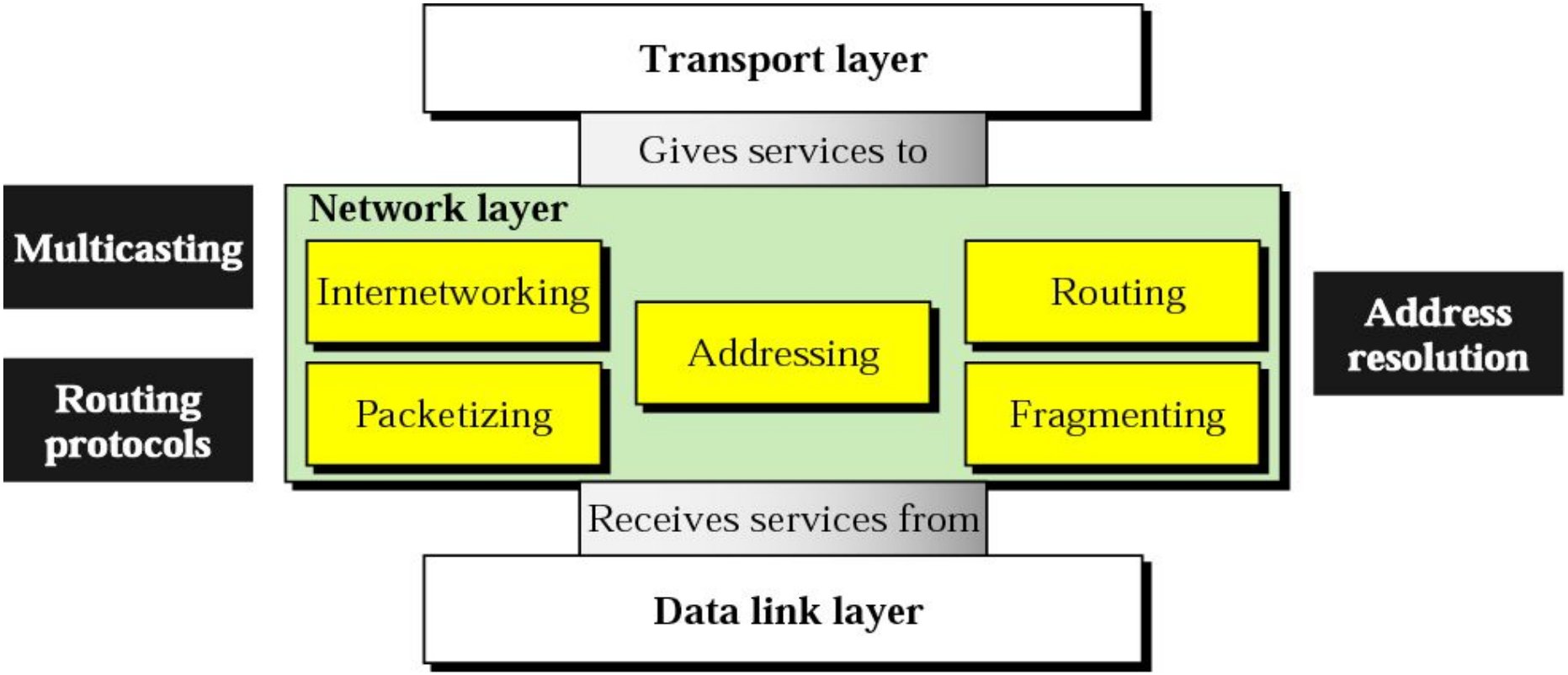
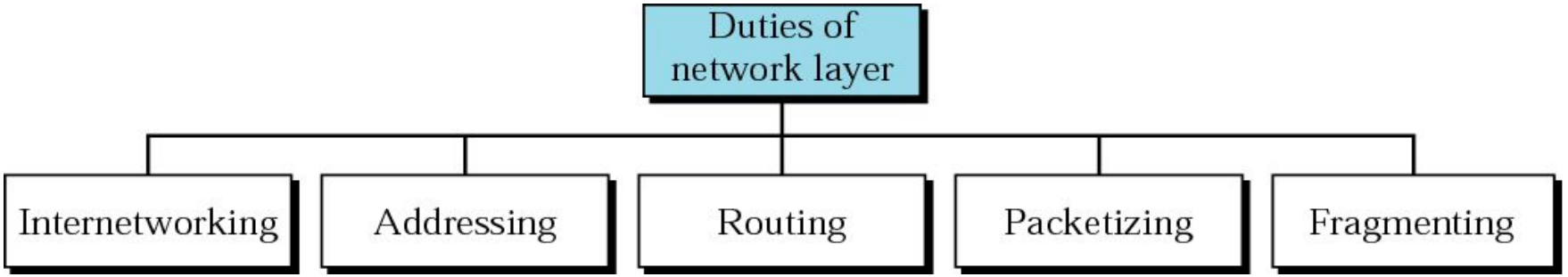


Network Layer

Position of network layer



Network layer duties



Host-to-Host Delivery:

Internetworking, Addressing, and Routing

19.1 Internetworks

Need For Network Layer

Internet As A Packet-Switched Network

Internet As A Connectionless Network

Figure 19.1 Internetwork

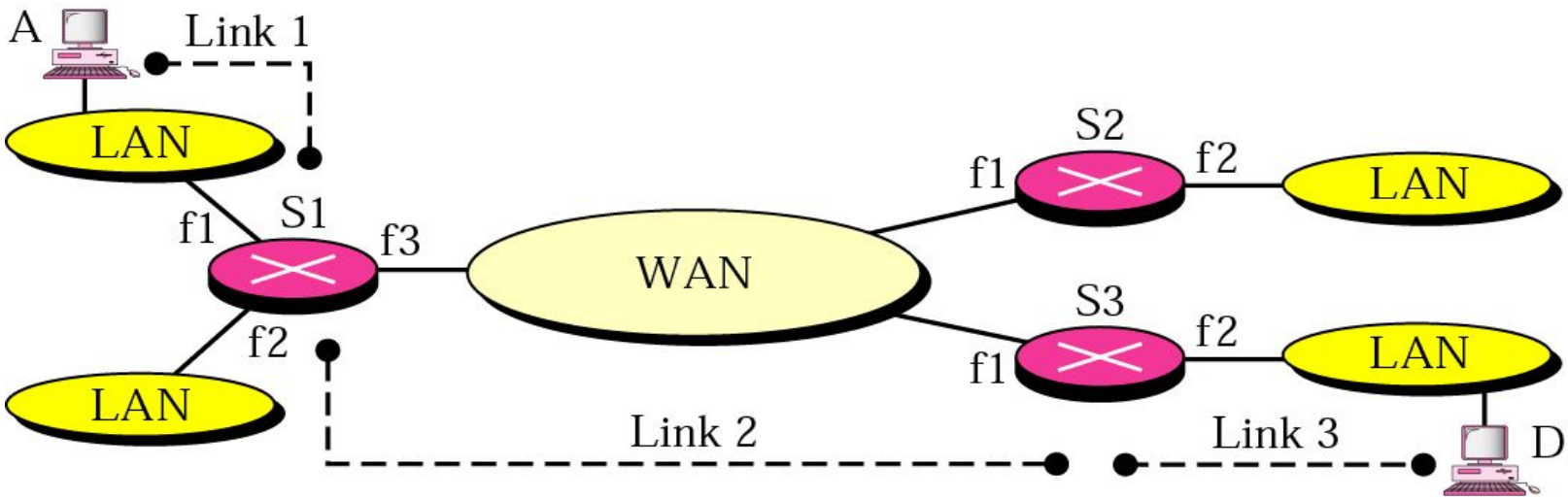


Figure 19.2 Links in an internetwork

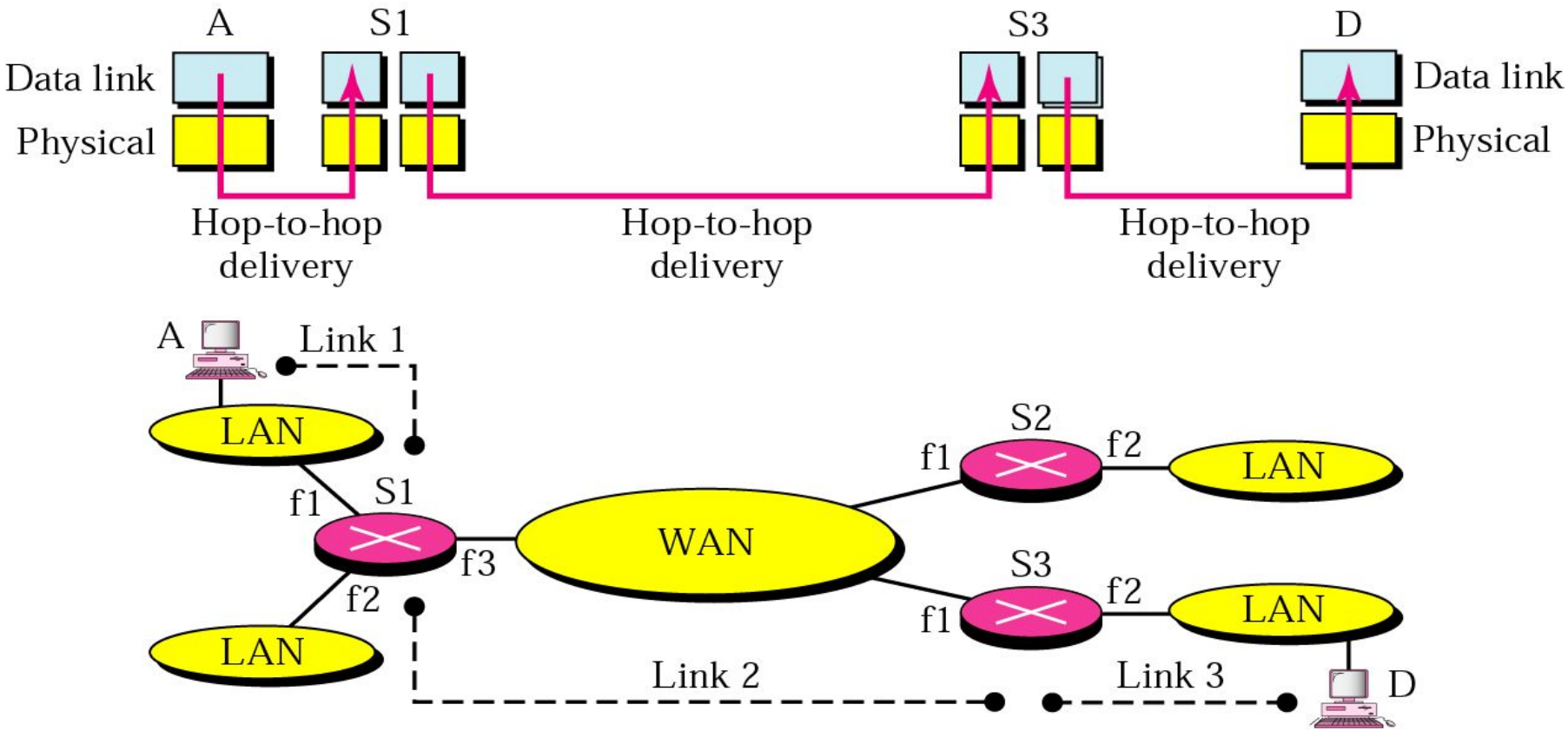


Figure 19.3 Network layer in an internetwork

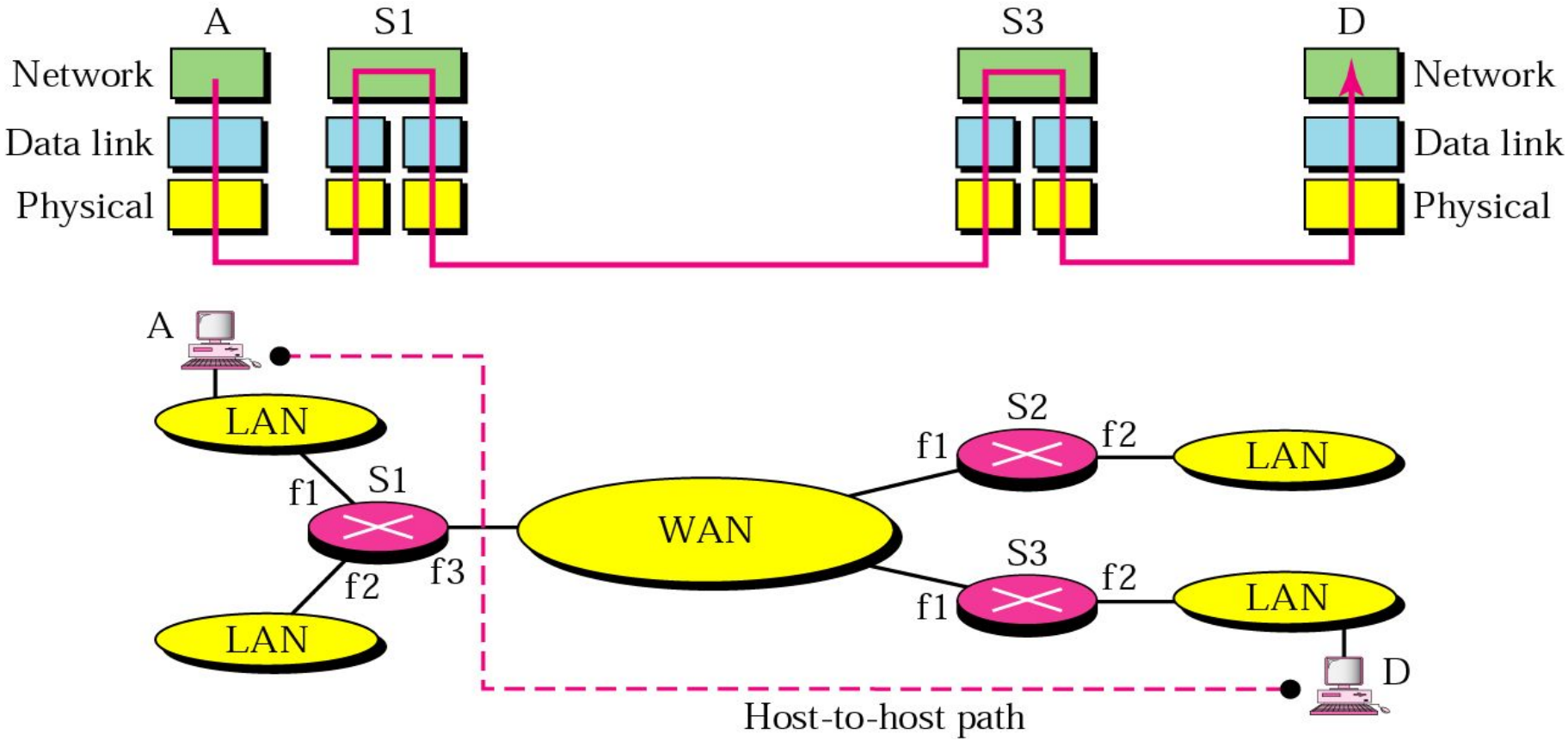


Figure 19.4 Network layer at the source

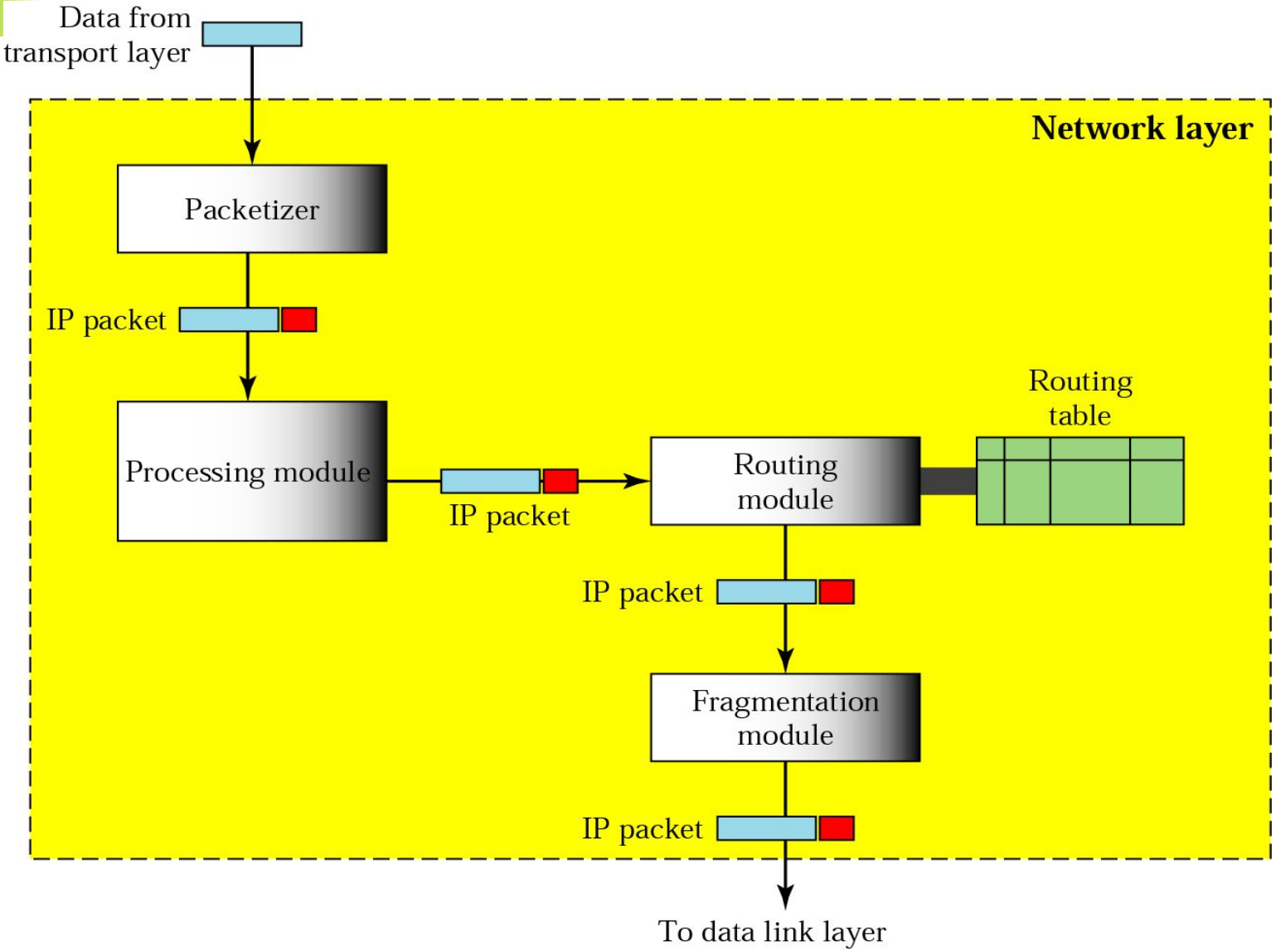


Figure 19.5 Network layer at a router

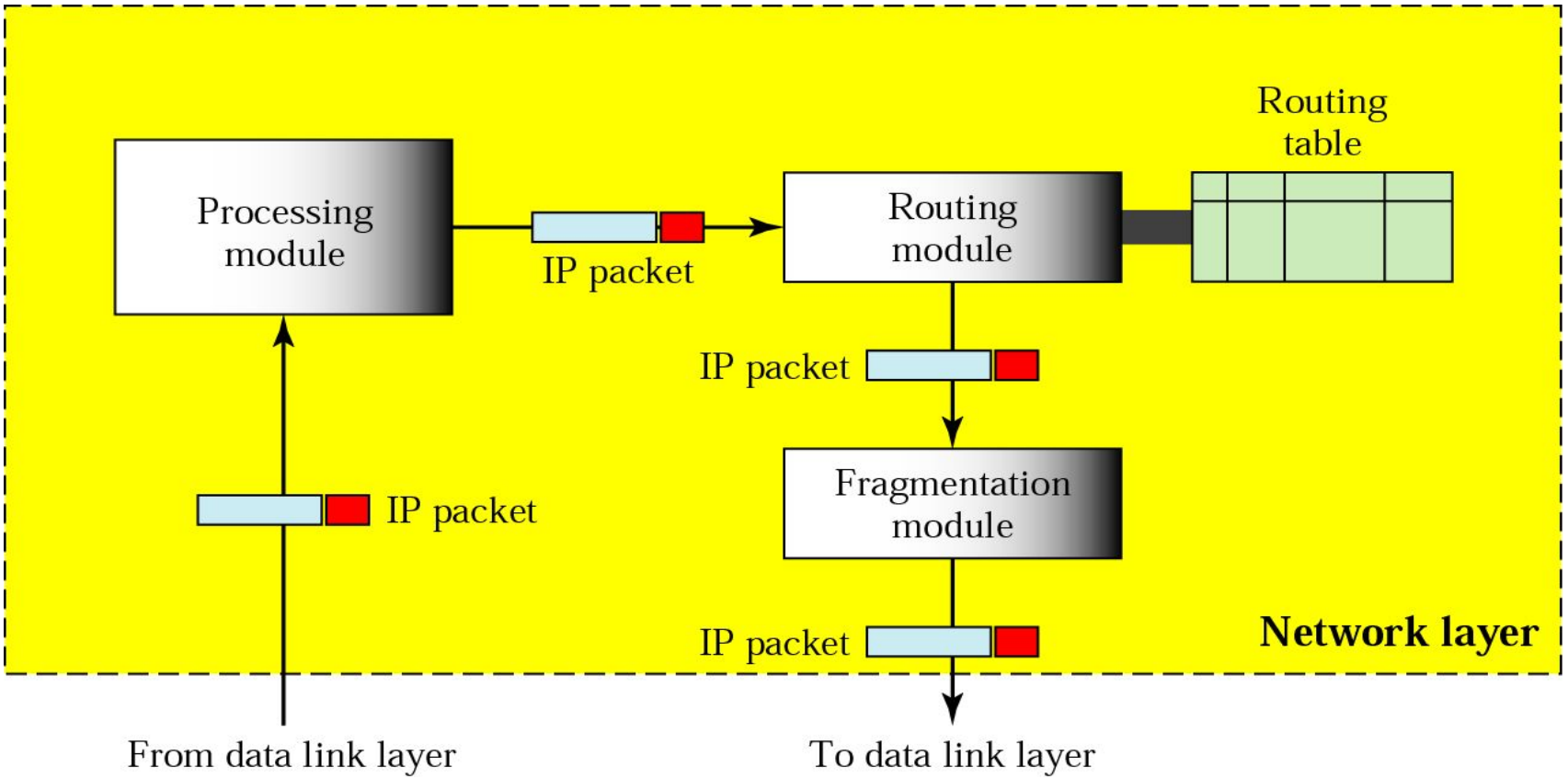


Figure 19.6 Network layer at the destination

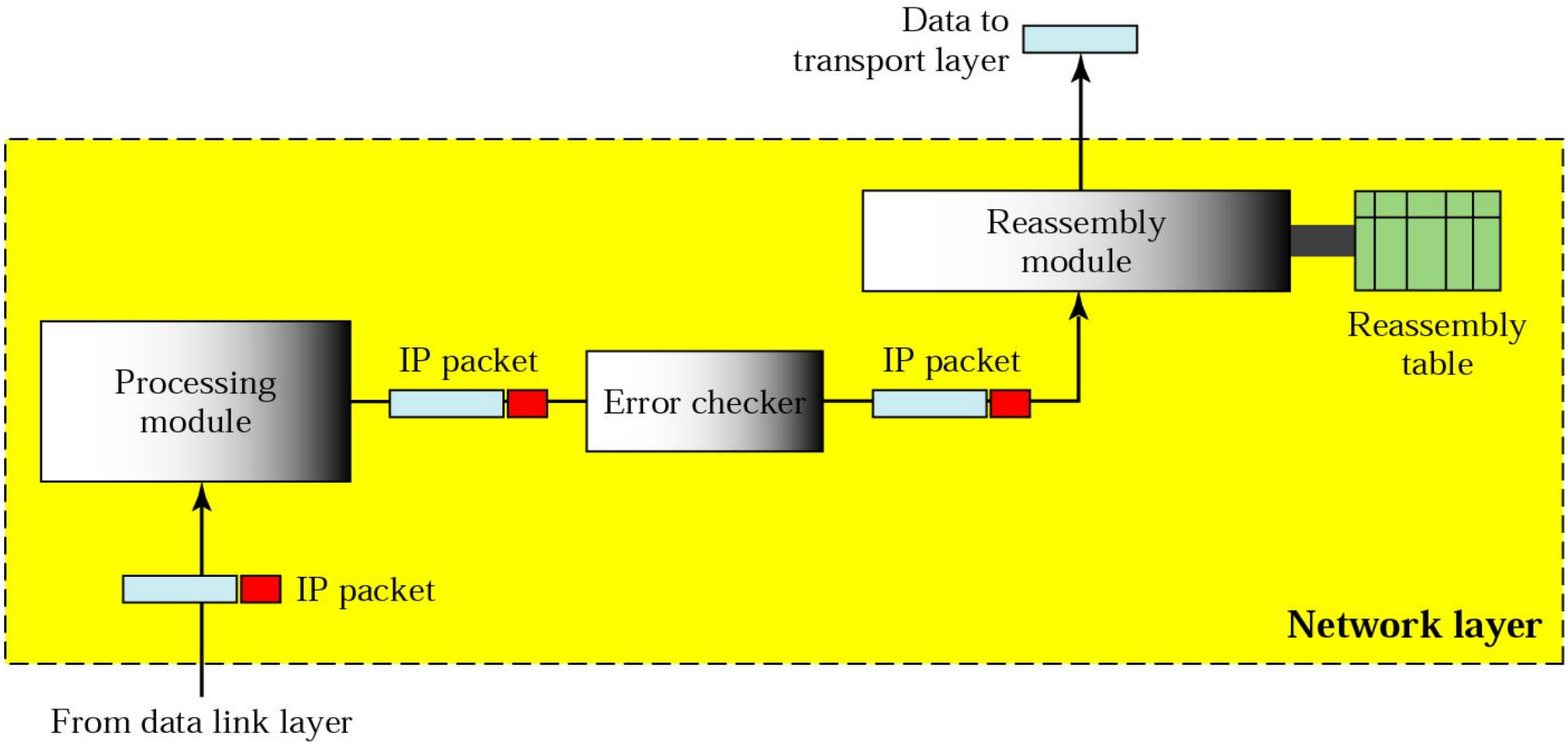


Figure 19.7 Switching

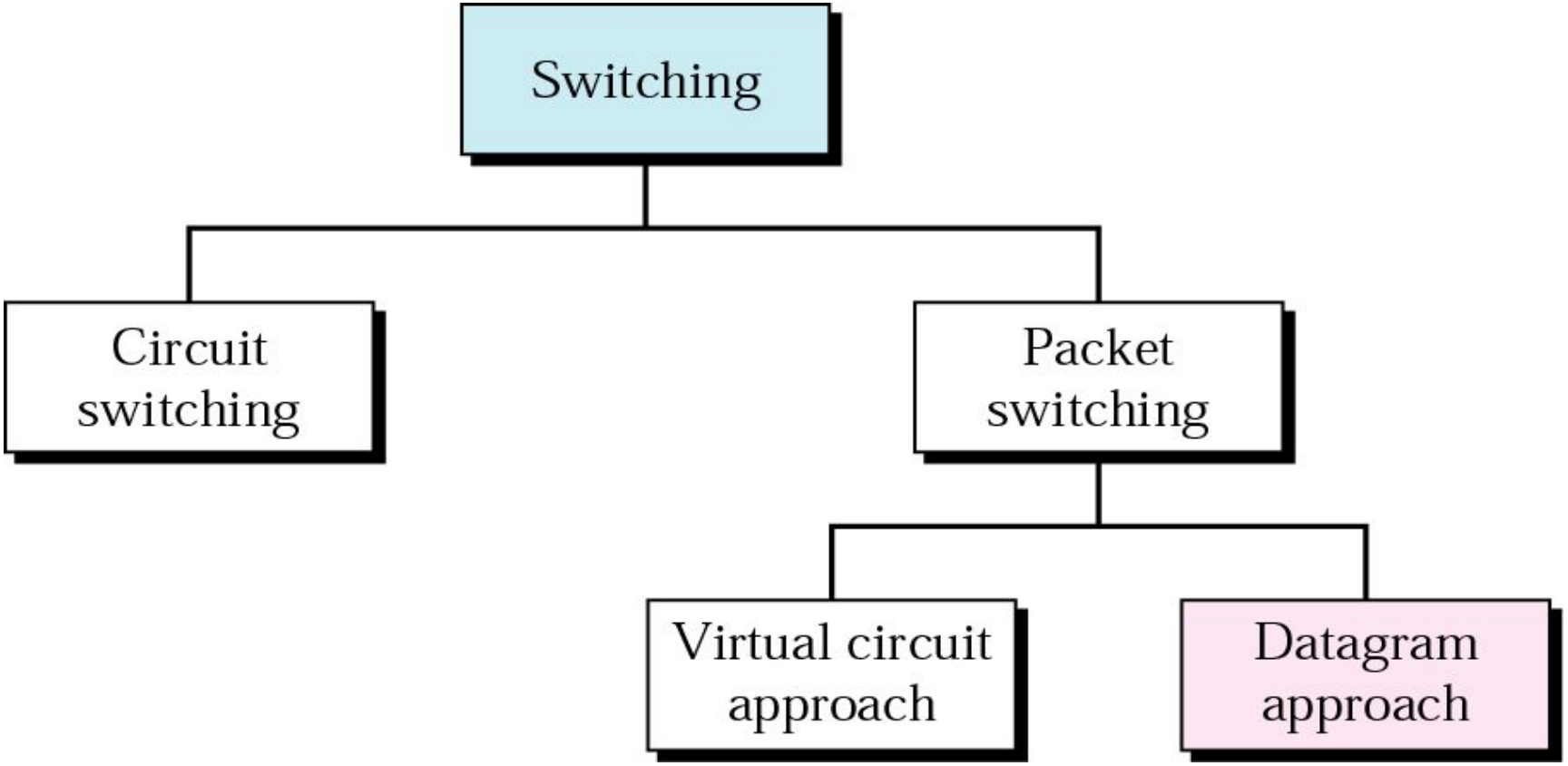
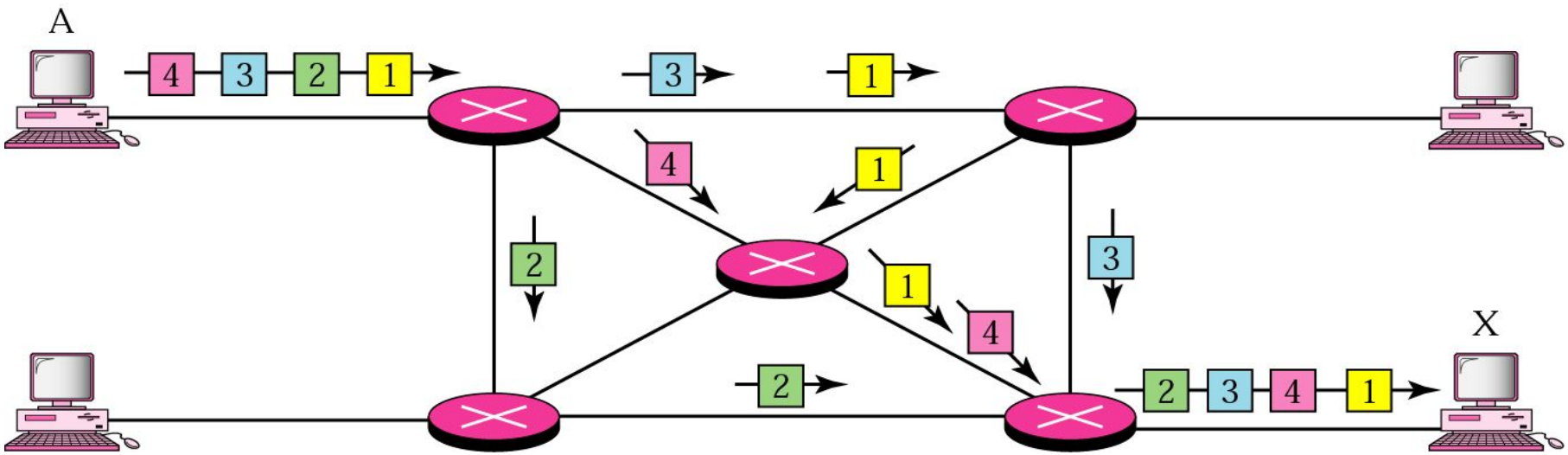


Figure 19.8 Datagram approach



Switching at the network layer in the Internet is done using the datagram approach to packet switching.

Communication at the network layer in the Internet is connectionless.

19.2 Addressing

Internet Address

Classful Addressing

Subnetting

Supernetting

Classless Addressing

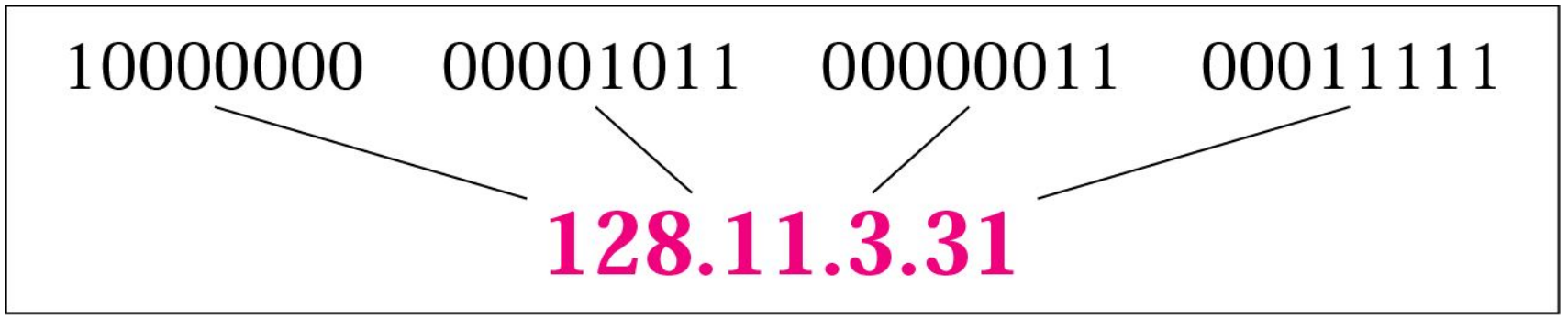
Dynamic Address Configuration

Network Address Translation

An IP address is a 32-bit address.

The IP addresses are unique and universal.

Figure 19.9 Dotted-decimal notation



Note:

The binary, decimal, and hexadecimal number systems are reviewed in Appendix B.

Example 1

Change the following IP addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation:

a. 129.11.11.239

b. 249.155.251.15

Example 2

Change the following IP addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 75.45.34.78

Solution

We replace each decimal number with its binary equivalent (see Appendix B):

a. 01101111 00111000 00101101 01001110
b. 01001011 00101101 00100010 01001110

Note:

In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

Figure 19.10 Finding the class in binary notation

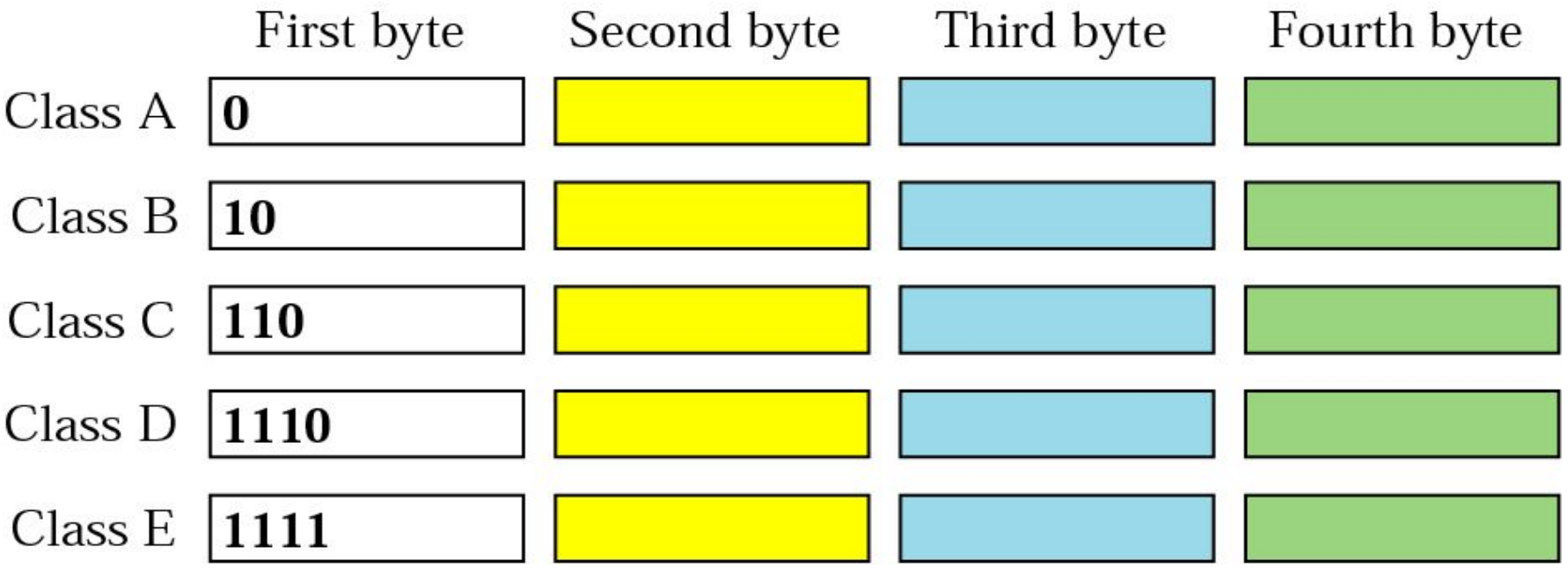
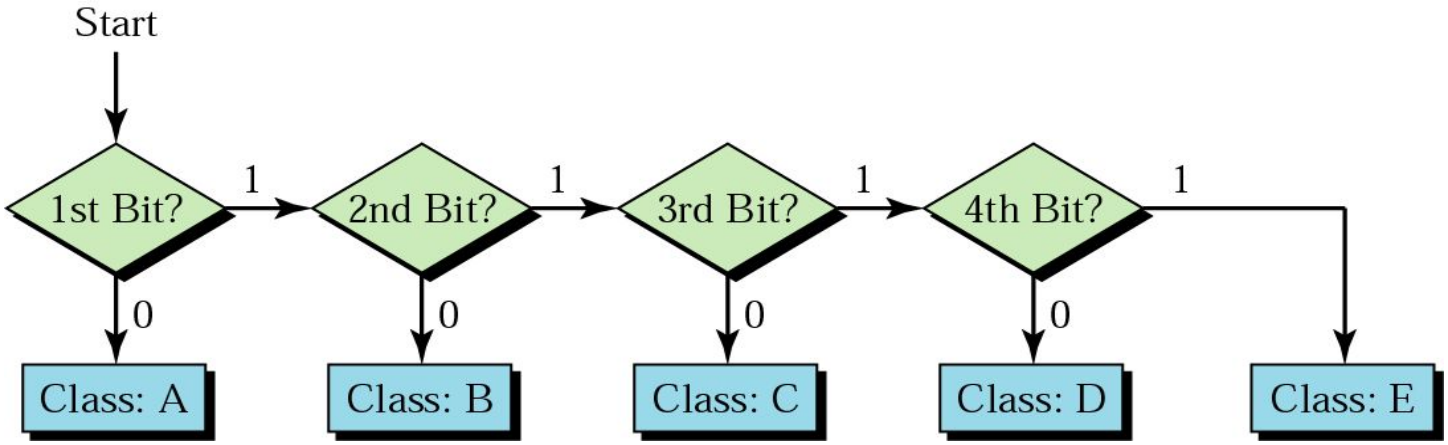


Figure 19.11 Finding the address class



Example 3

Find the class of each address:

a. **0**0000001 00001011 00001011 11101111

b. **1111**0011 10011011 11111011 00001111

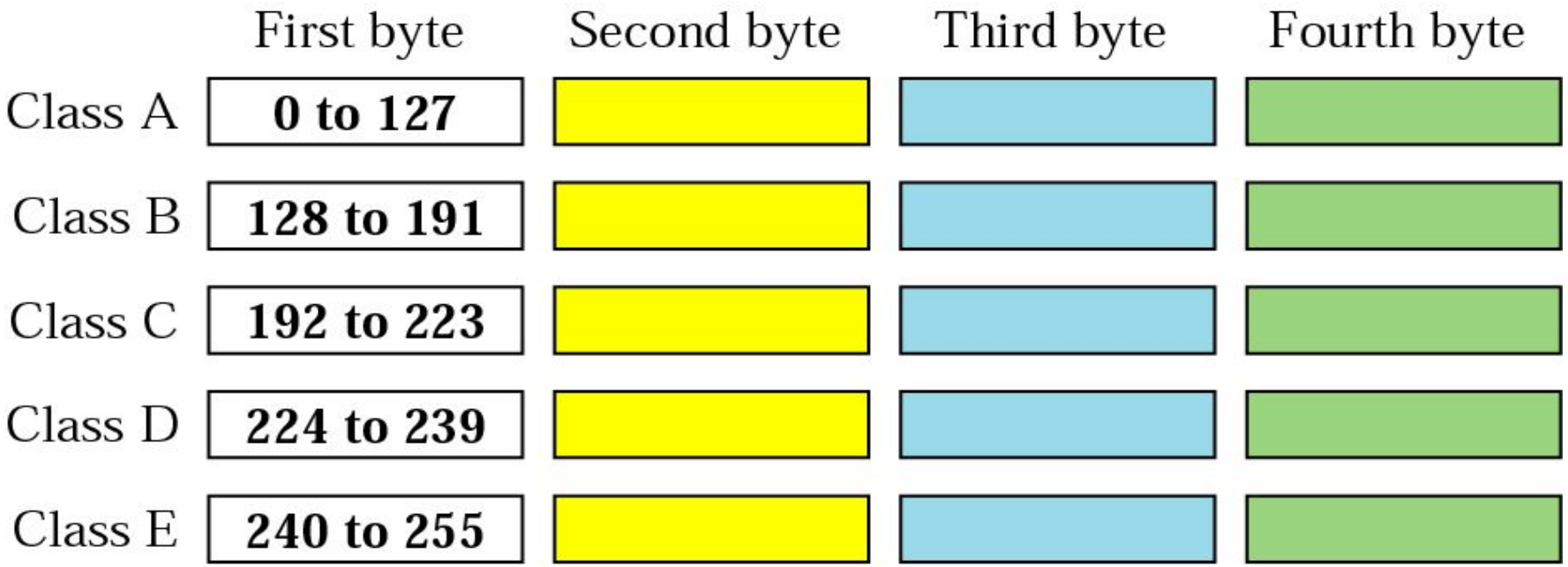
Solution

See the procedure in Figure 19.11.

a. The first bit is 0; this is a class A address.

b. The first 4 bits are 1s; this is a class E address.

Figure 19.12 **Finding the class in decimal notation**



Example 4

Find the class of each address:

a. **227**.12.14.87

b. **252**.5.15.111

c. **134**.11.78.56

Solution

a. The first byte is 227 (between 224 and 239); the class is D.
b. The first byte is 252 (between 240 and 255); the class is E.
c. The first byte is 134 (between 128 and 191); the class is B.

Figure 19.13 Netid and hostid

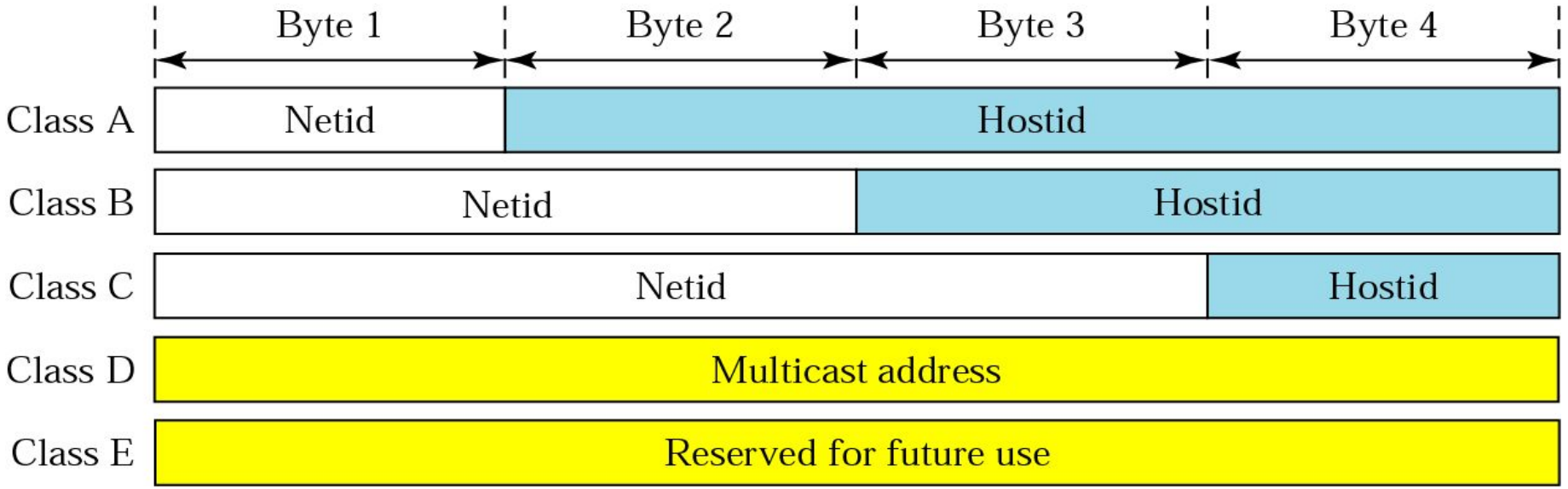
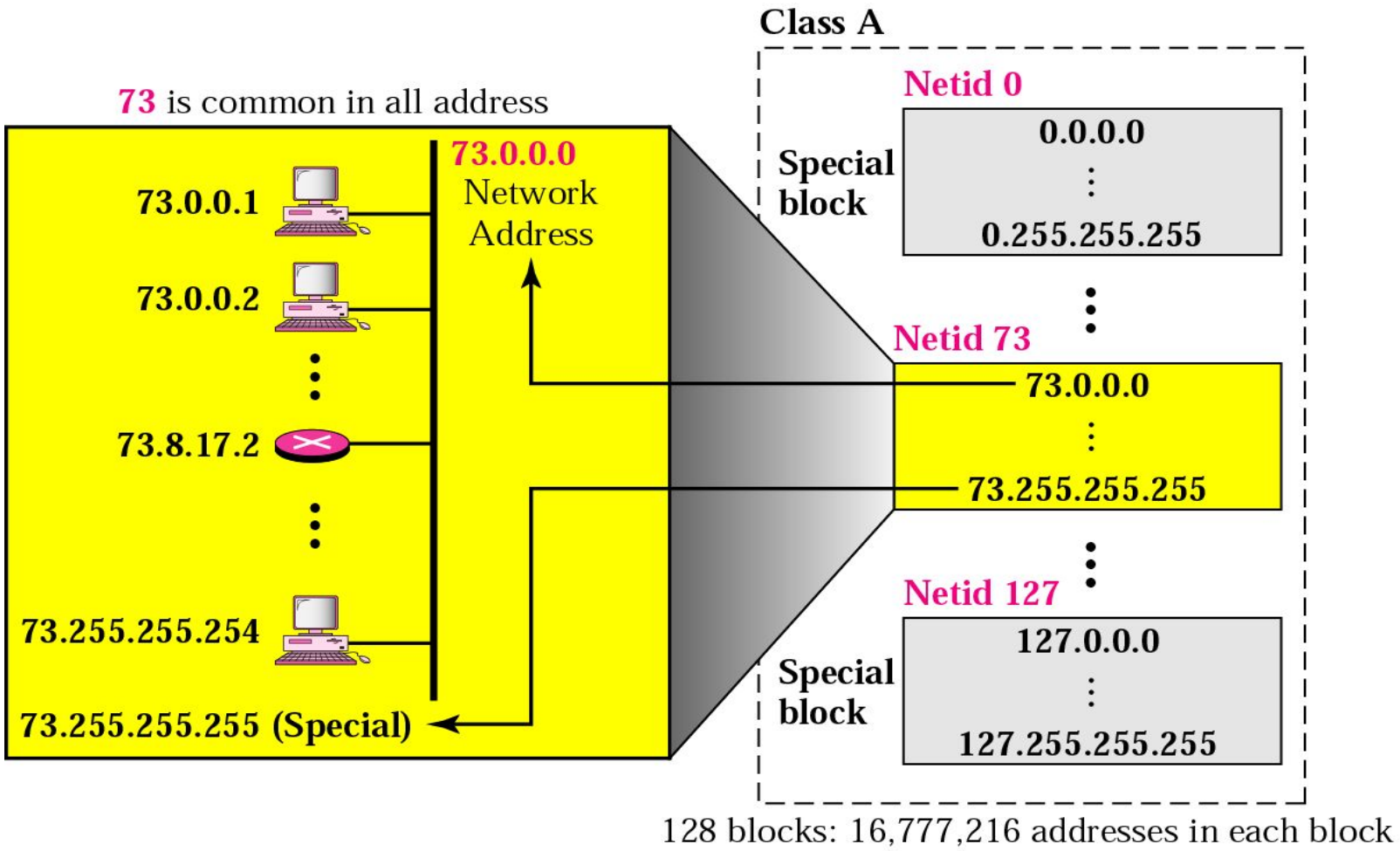
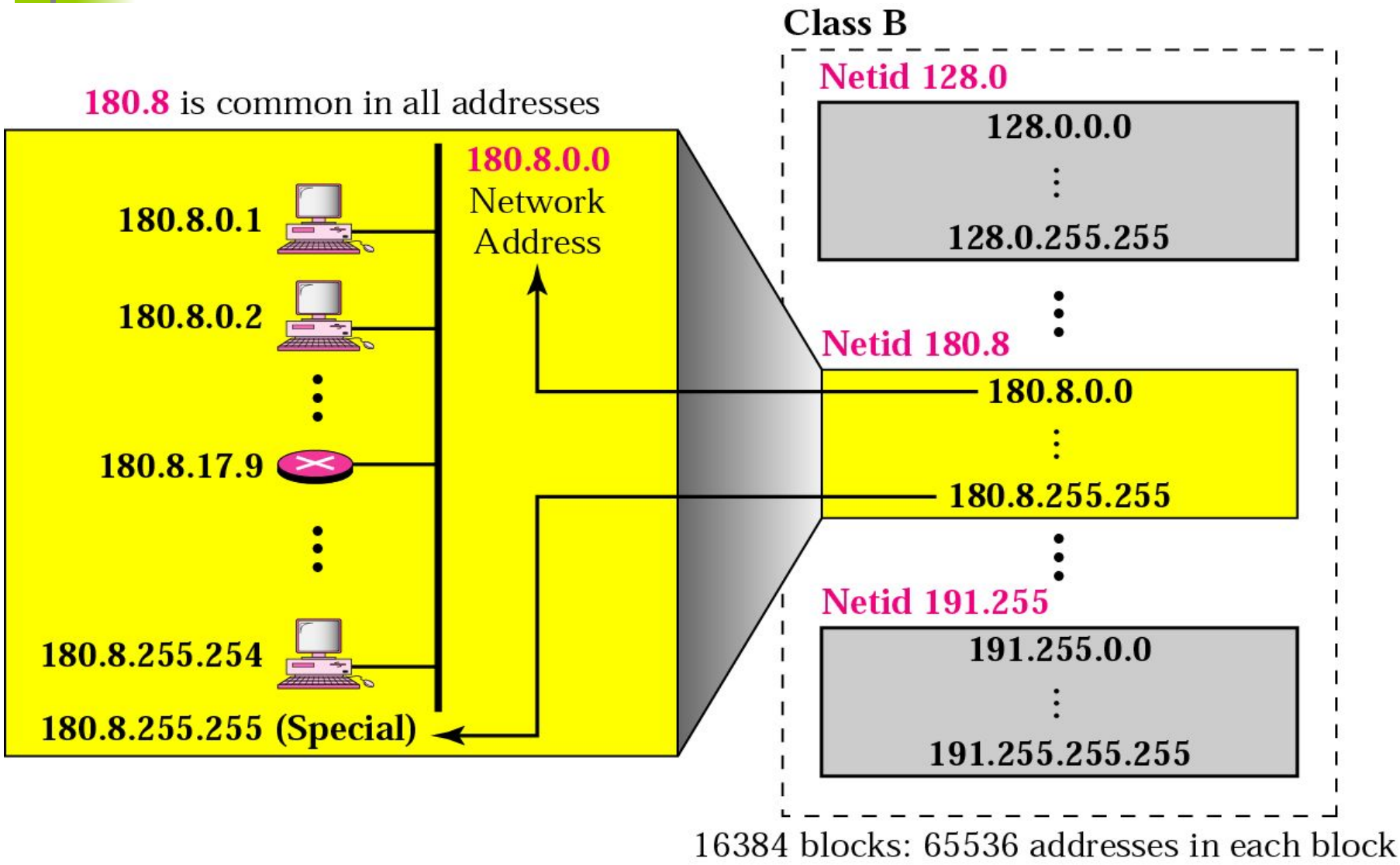


Figure 19.14 Blocks in class A



Millions of class A addresses are wasted.

Figure 19.15 Blocks in class B



Many class B addresses are wasted.

The number of addresses in class C is smaller than the needs of most organizations.

Figure 19.16 Blocks in class C

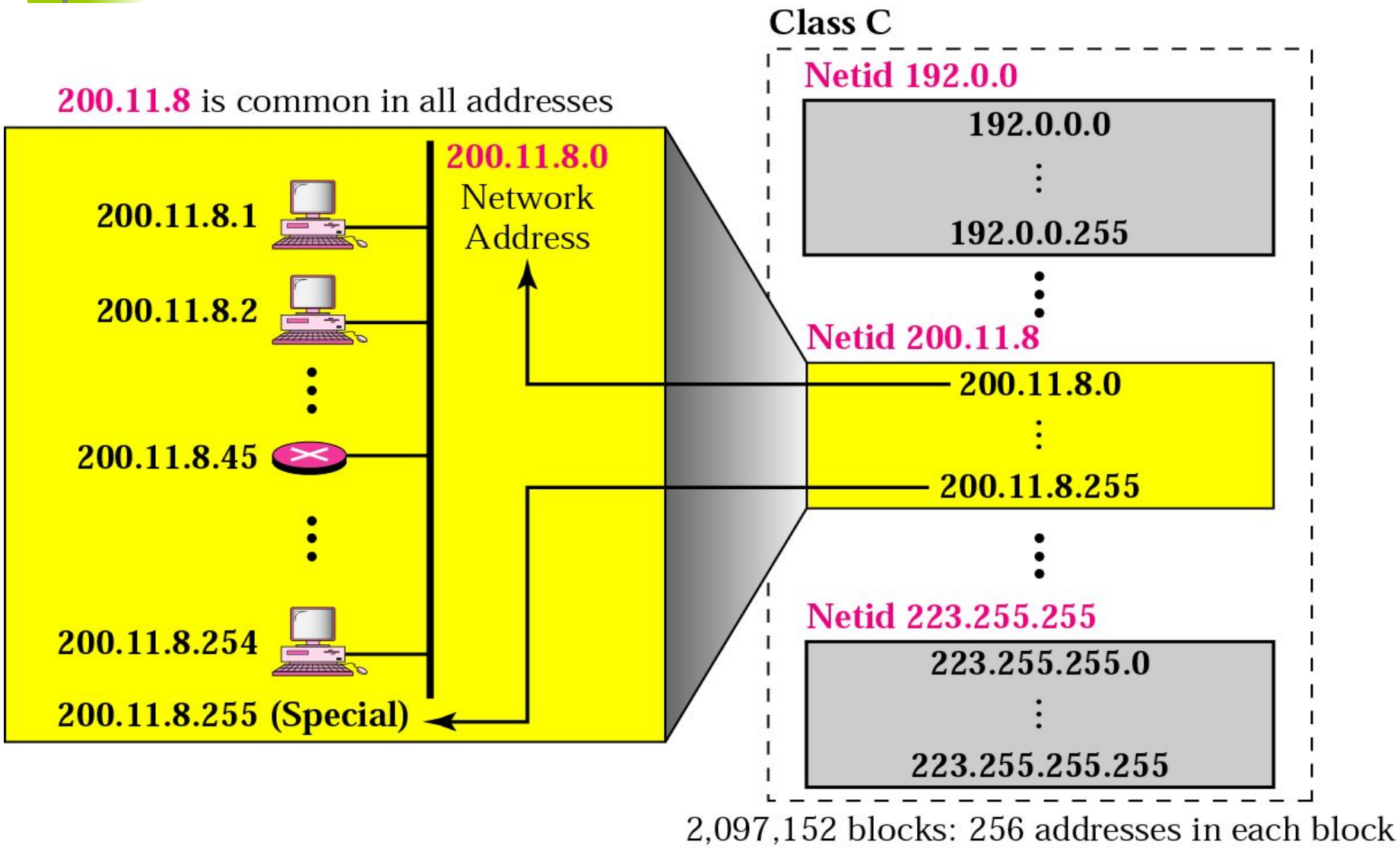
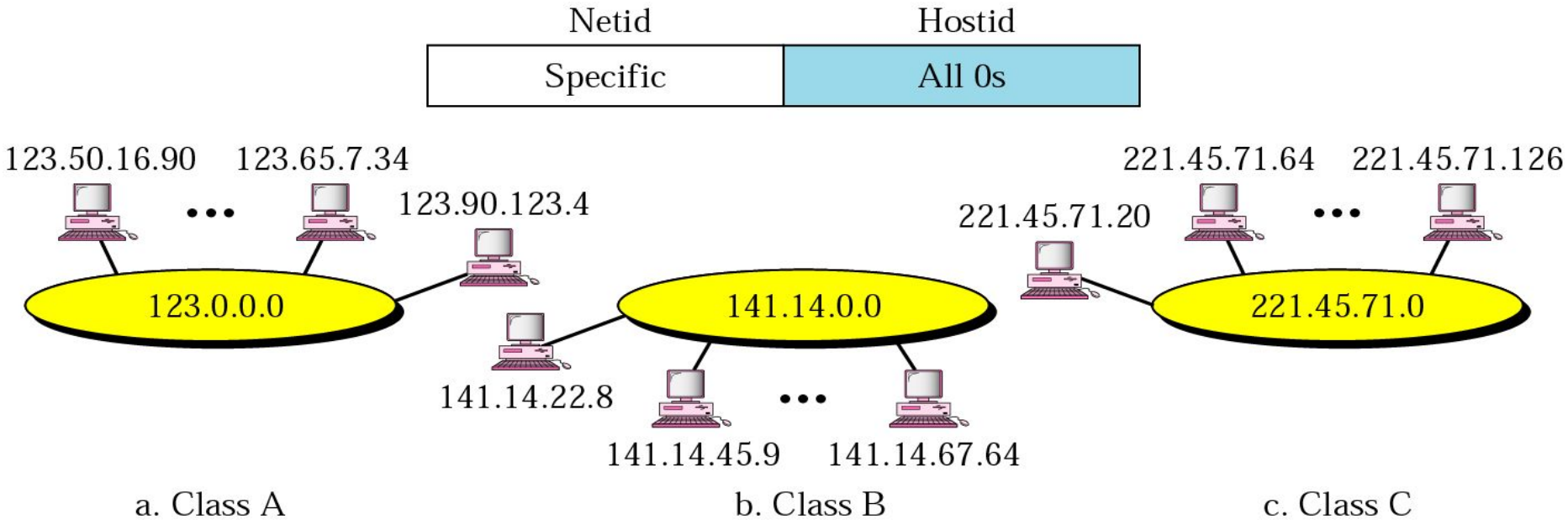


Figure 19.17 Network address



In classful addressing, the network address is the one that is assigned to the organization.

Example 5

Given the address 23.56.7.91, find the network address.

Solution

The class is A. Only the first byte defines the netid. We can find the network address by replacing the hostid bytes (56.7.91) with 0s. Therefore, the network address is 23.0.0.0.

Example 6

Given the address 132.6.17.85, find the network address.

Solution

The class is B. The first 2 bytes defines the netid. We can find the network address by replacing the hostid bytes (17.85) with 0s. Therefore, the network address is 132.6.0.0.

Example 7

Given the network address 17.0.0.0, find the class.

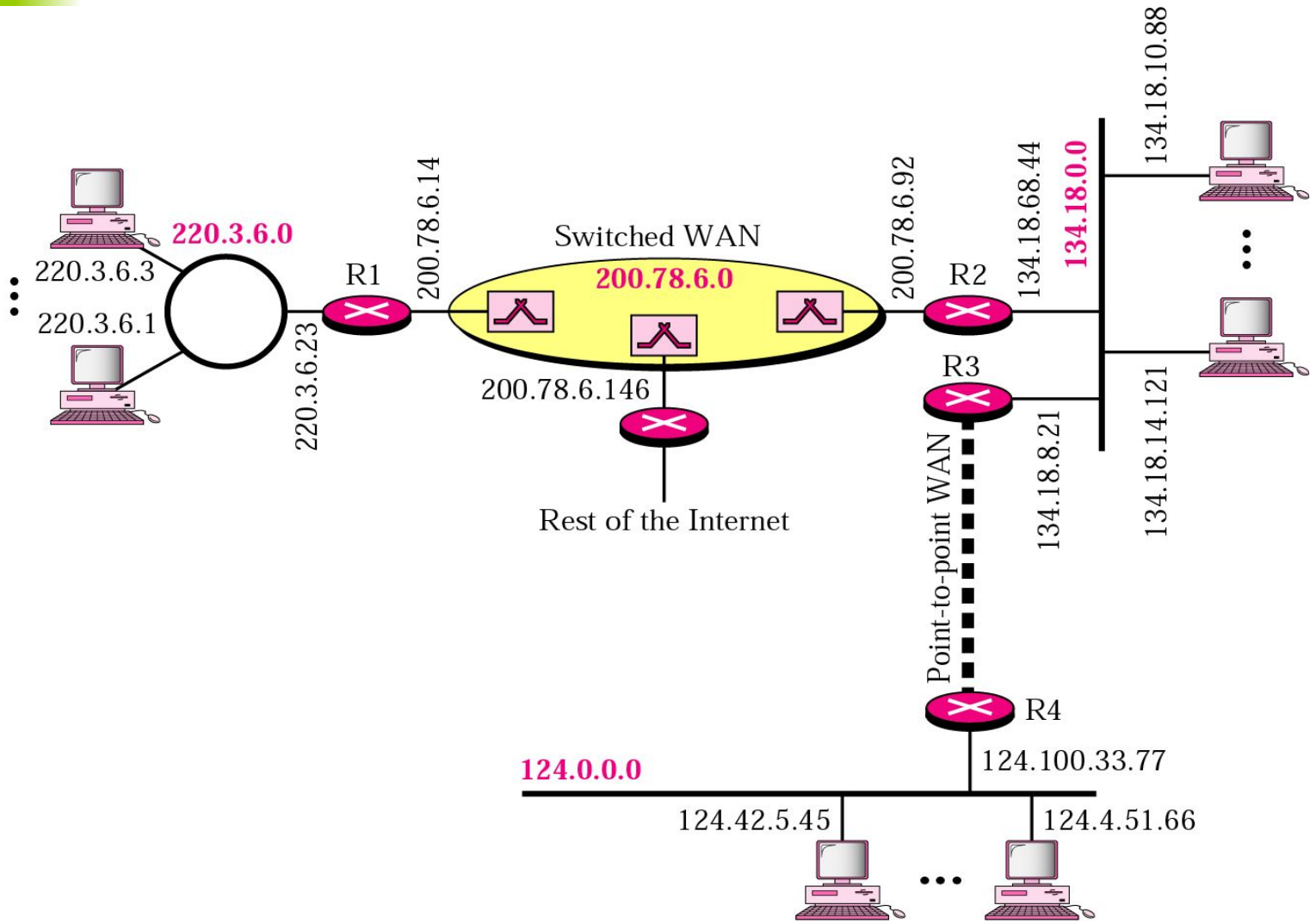
Solution

The class is A because the netid is only 1 byte.

Note:

A network address is different from a netid. A network address has both netid and hostid, with 0s for the hostid.

Figure 19.18 Sample internet



Note:

IP addresses are designed with two levels of hierarchy.

Figure 19.19 A network with two levels of hierarchy

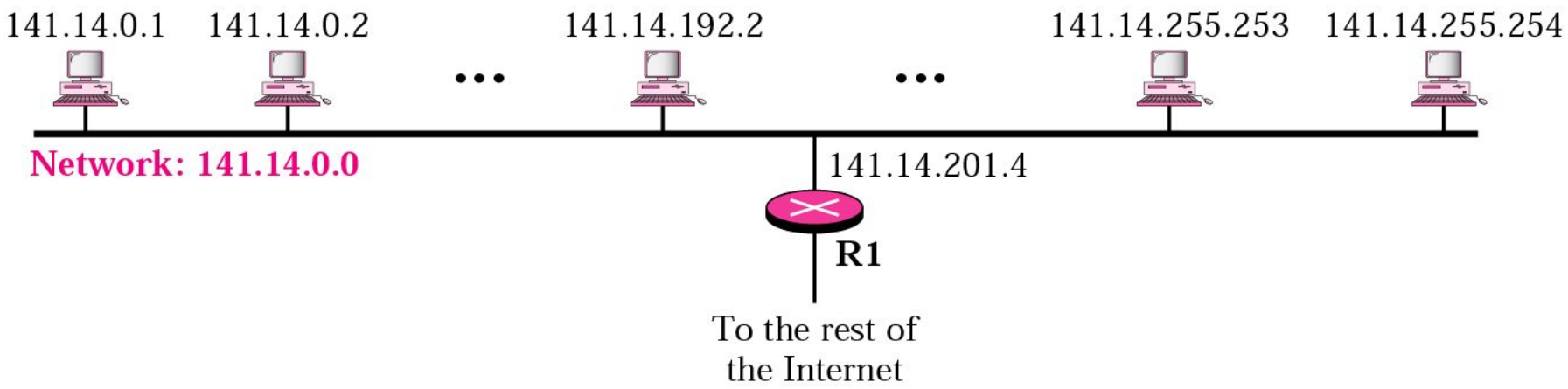


Figure 19.20 A network with three levels of hierarchy (subnetted)

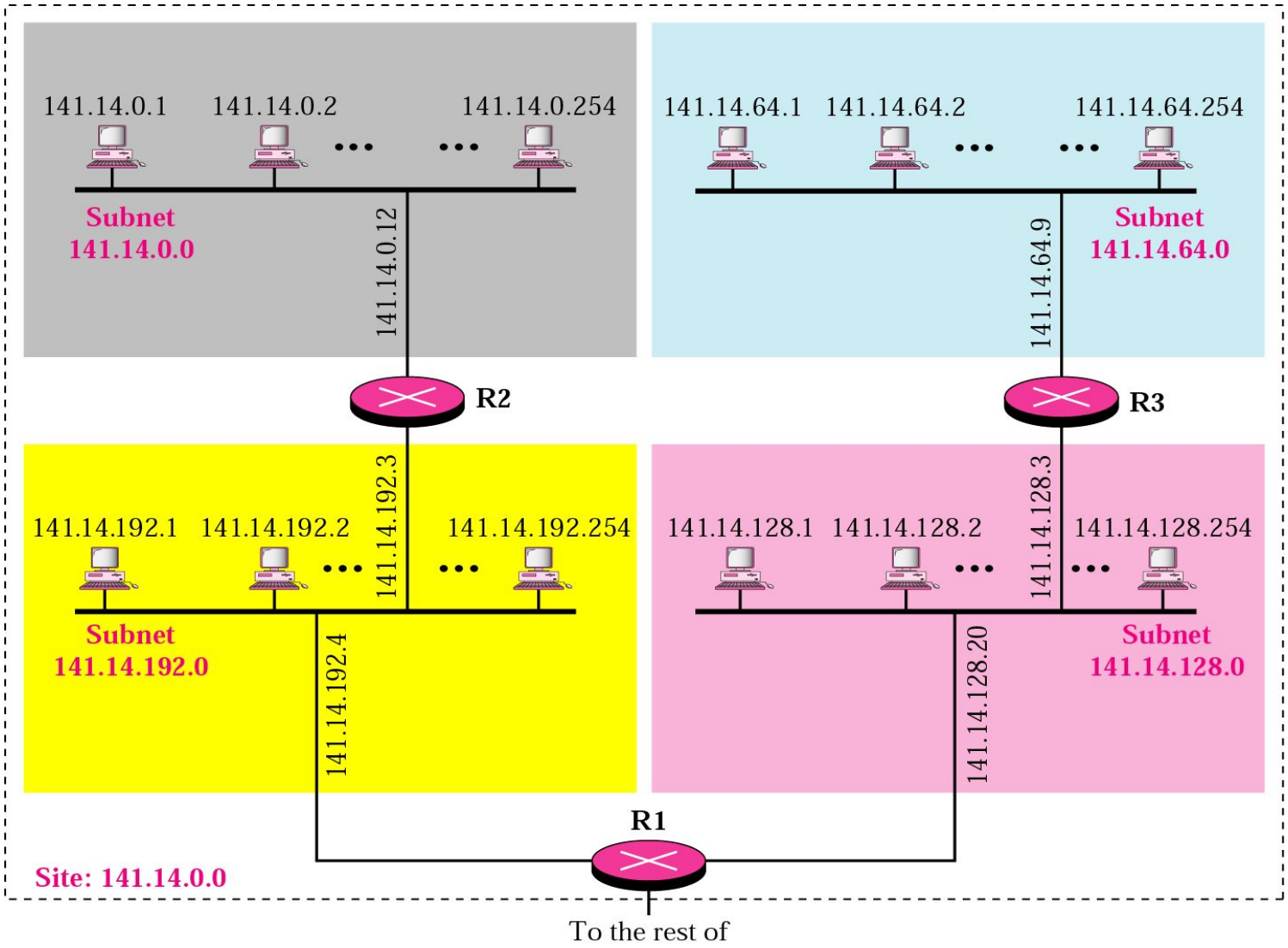


Figure 19.21 Addresses in a network with and without subnetting

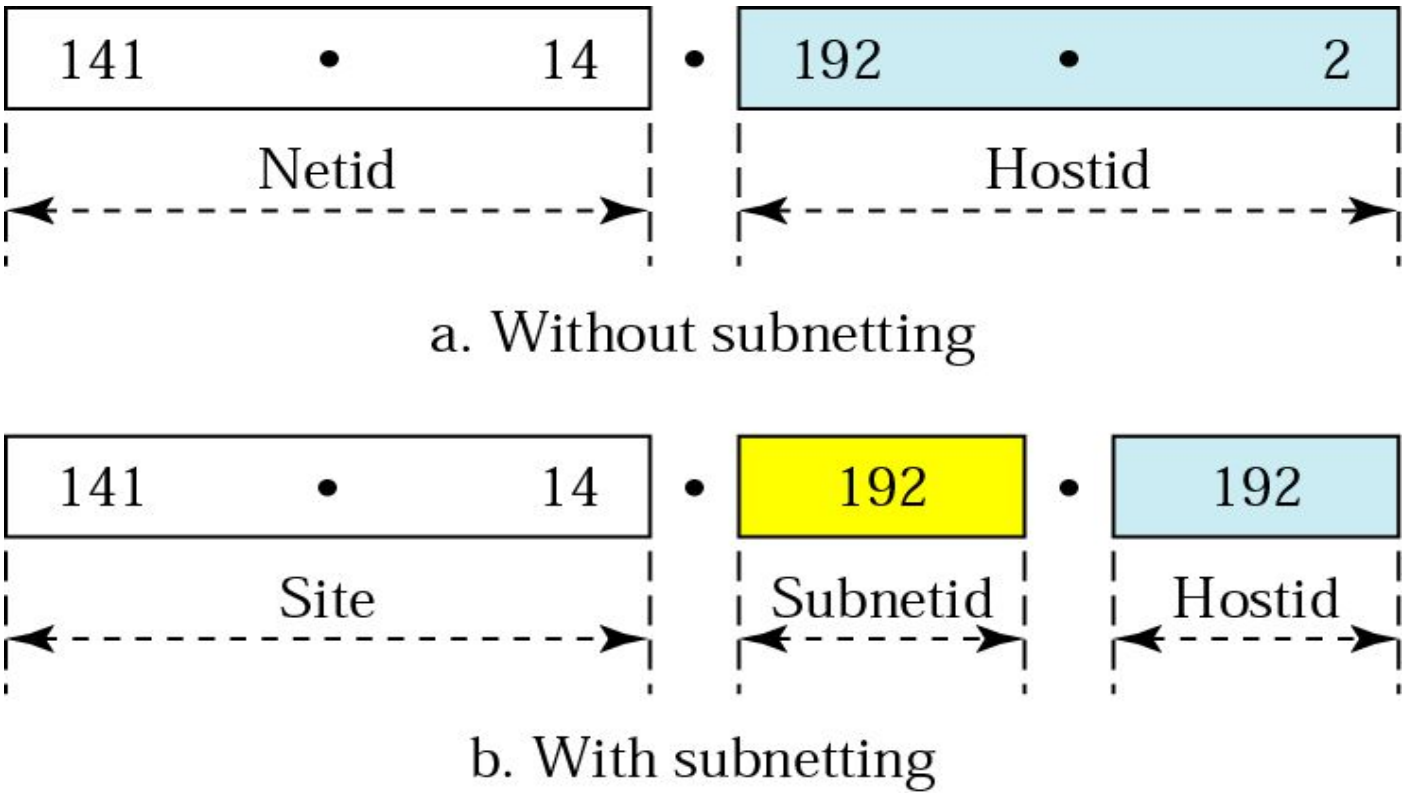


Figure 19.22 Hierarchy concept in a telephone number

Table 19.1 Default masks

Class	*In Binary*	*In Dotted-Decimal*	*Using Slash*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 111111111 11111111 00000000	255.255.255.0	/24

Note:

The network address can be found by applying the default mask to any address in the block (including itself). It retains the netid of the block and sets the hostid to 0s.

Example 8

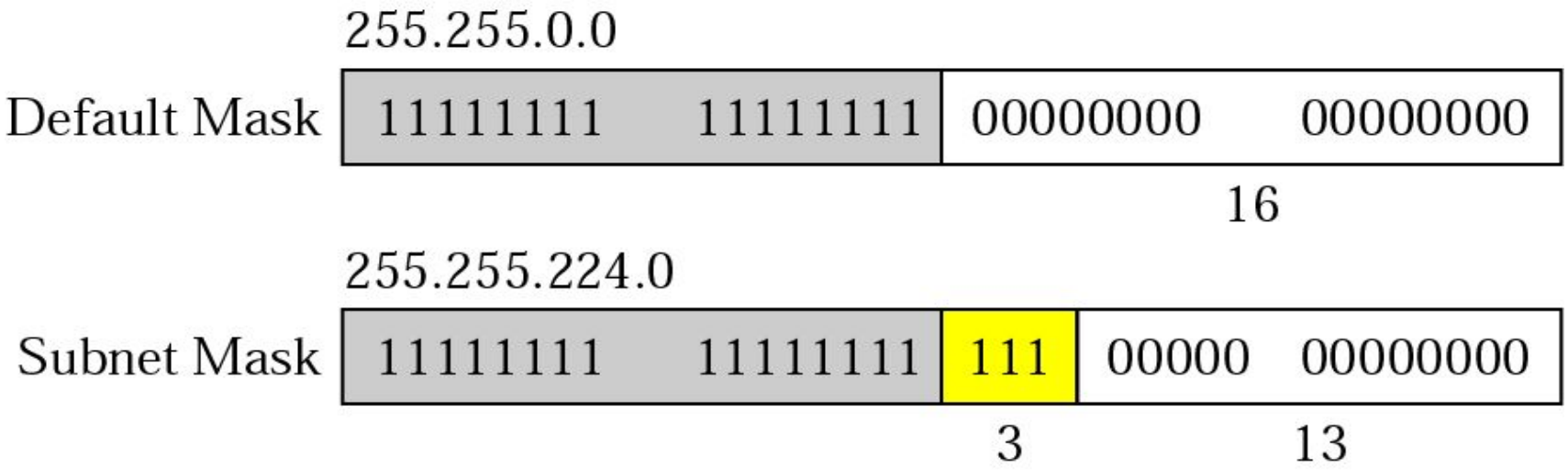
A router outside the organization receives a packet with destination address 190.240.7.91. Show how it finds the network address to route the packet.

Solution

The router follows three steps:

1. The router looks at the first byte of the address to find the class. It is class B.
2. The default mask for class B is 255.255.0.0. The router ANDs this mask with the address to get 190.240.0.0.
3. The router looks in its routing table to find out how to route the packet to this destination. Later, we will see what happens if this destination does not exist.

Figure 19.23 Subnet mask



Example 9

A router inside the organization receives the same packet with destination address 190.240.33.91. Show how it finds the subnetwork address to route the packet.

Solution

The router follows three steps:

1. **The router must know the mask. We assume it is /19, as shown in Figure 19.23.**
2. **The router applies the mask to the address, 190.240.33.91. The subnet address is 190.240.32.0.**
3. **The router looks in its routing table to find how to route the packet to this destination. Later, we will see what happens if this destination does not exist.**

Table 19.2 Default masks

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

Figure 19.25 NAT

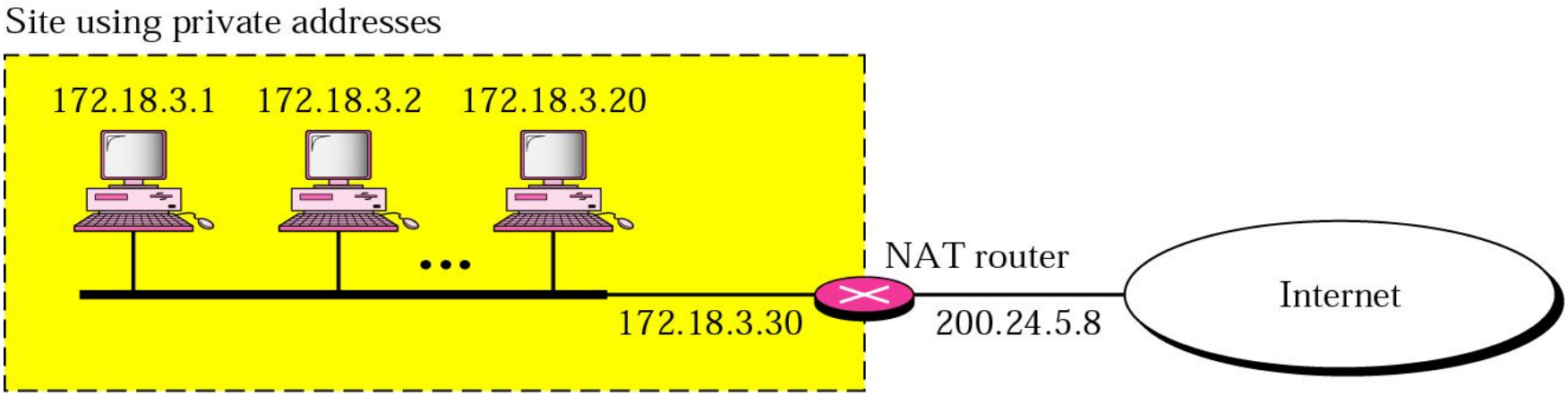


Figure 19.26 Address translation

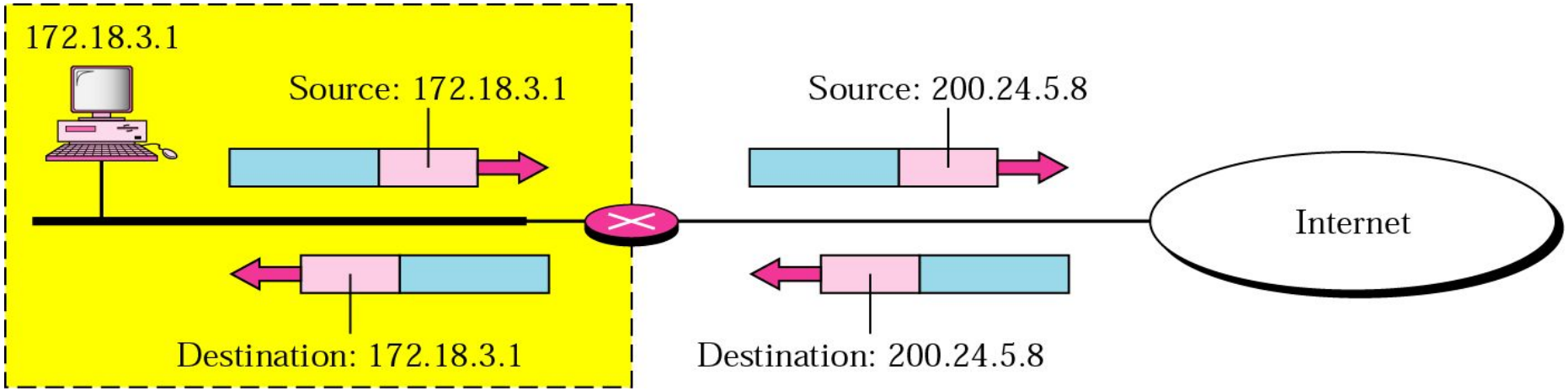


Figure 19.27 Translation

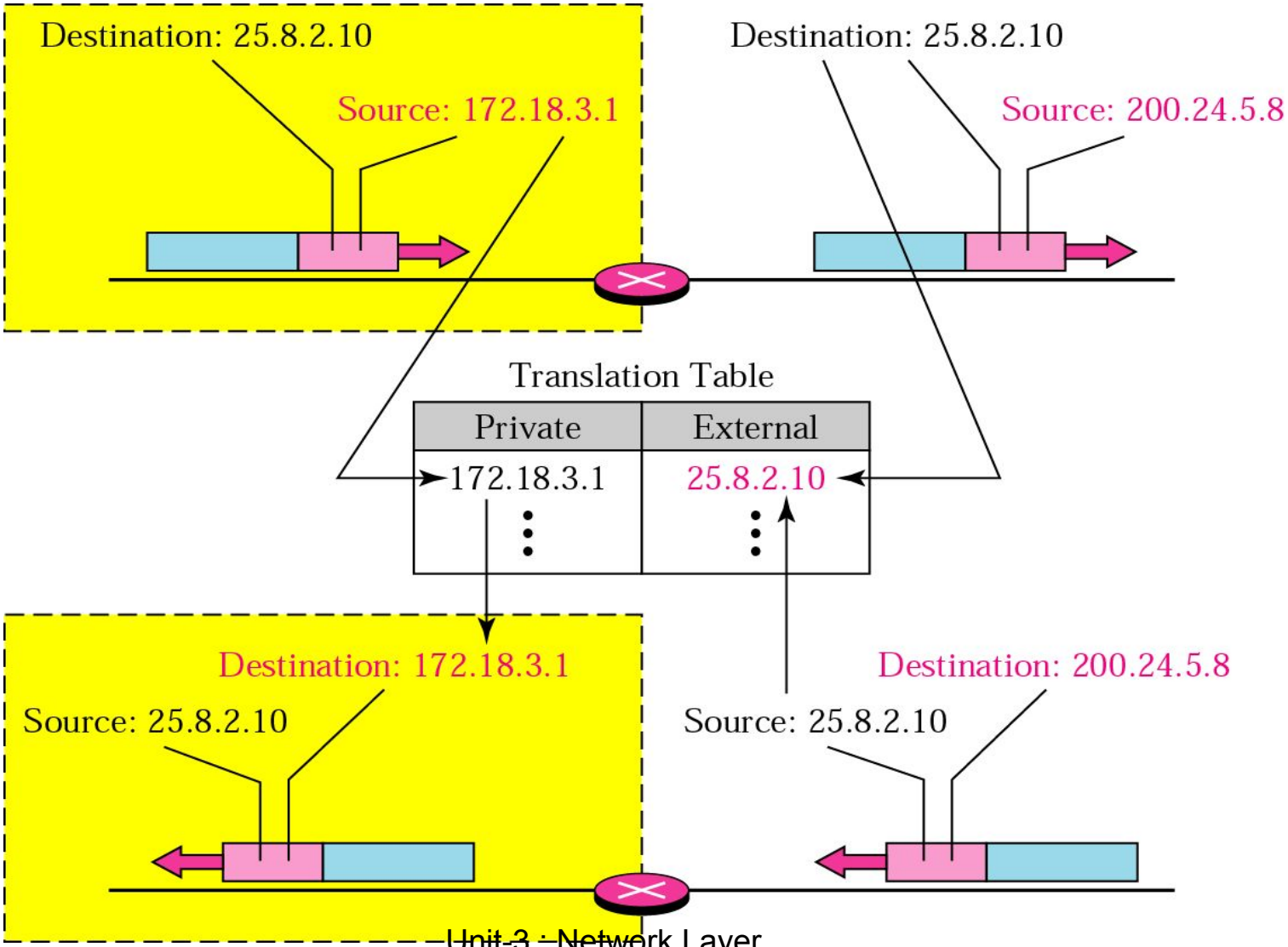


Table 19.3 Five-column translation table

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
...	...	...	...	...

19.3 Routing

Routing Techniques

Static Versus Dynamic Routing

Routing Table for Classful Addressing

Routing Table for Classless Addressing

Figure 19.28 Next-hop routing

Routing table for host A

Destination	Route
Host B	R1, R2, Host B

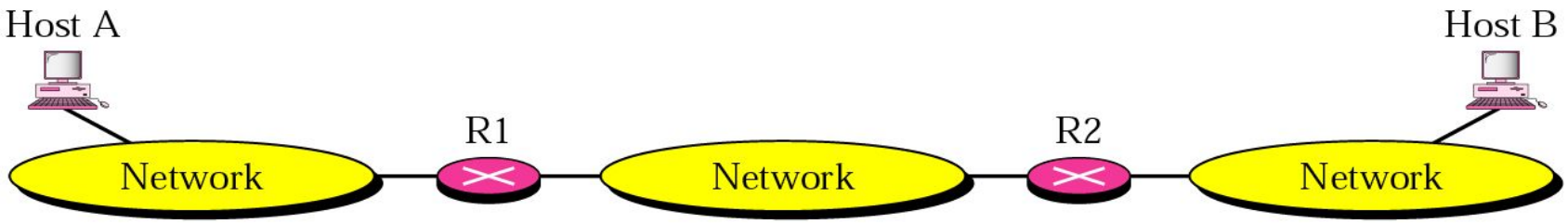
Routing table for R1

Destination	Route
Host B	R2, Host B

Routing table for R2

Destination	Route
Host B	Host B

a. Routing tables based on route



Routing table for host A

Destination	Next Hop
Host B	R1

Routing table for R1

Destination	Next Hop
Host B	R2

Routing table for R2

Destination	Next Hop
Host B	—

b. Routing tables based on next hop

Figure 19.29 Network-specific routing

Routing table for host S based on host-specific routing

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific routing

Destination	Next Hop
N2	R1

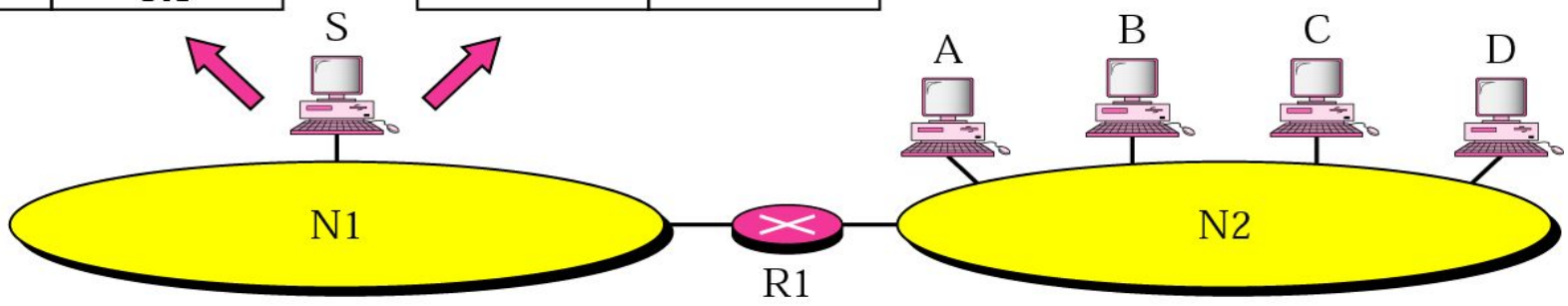


Figure 19.30 Host-specific routing

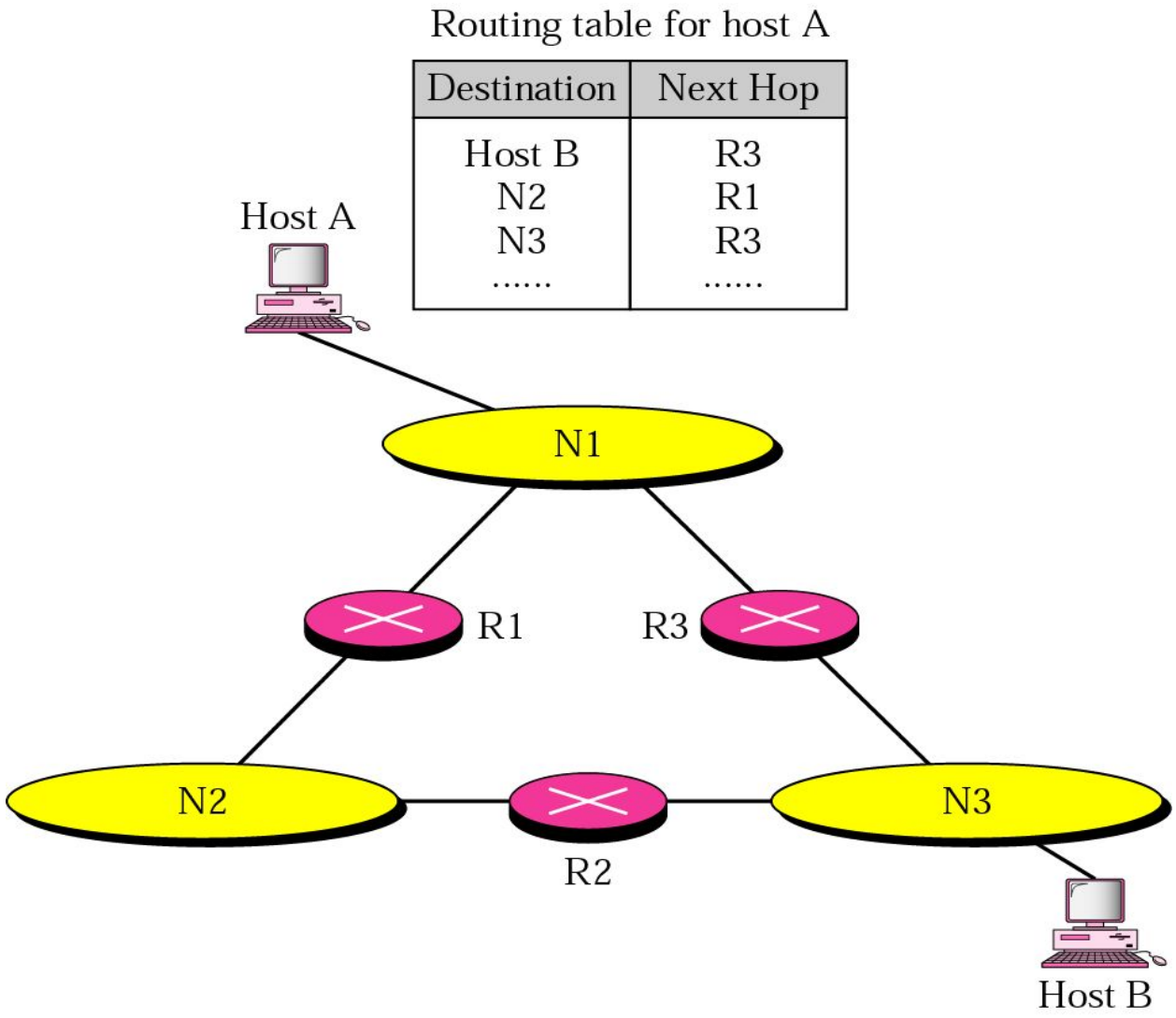


Figure 19.31 Default routing

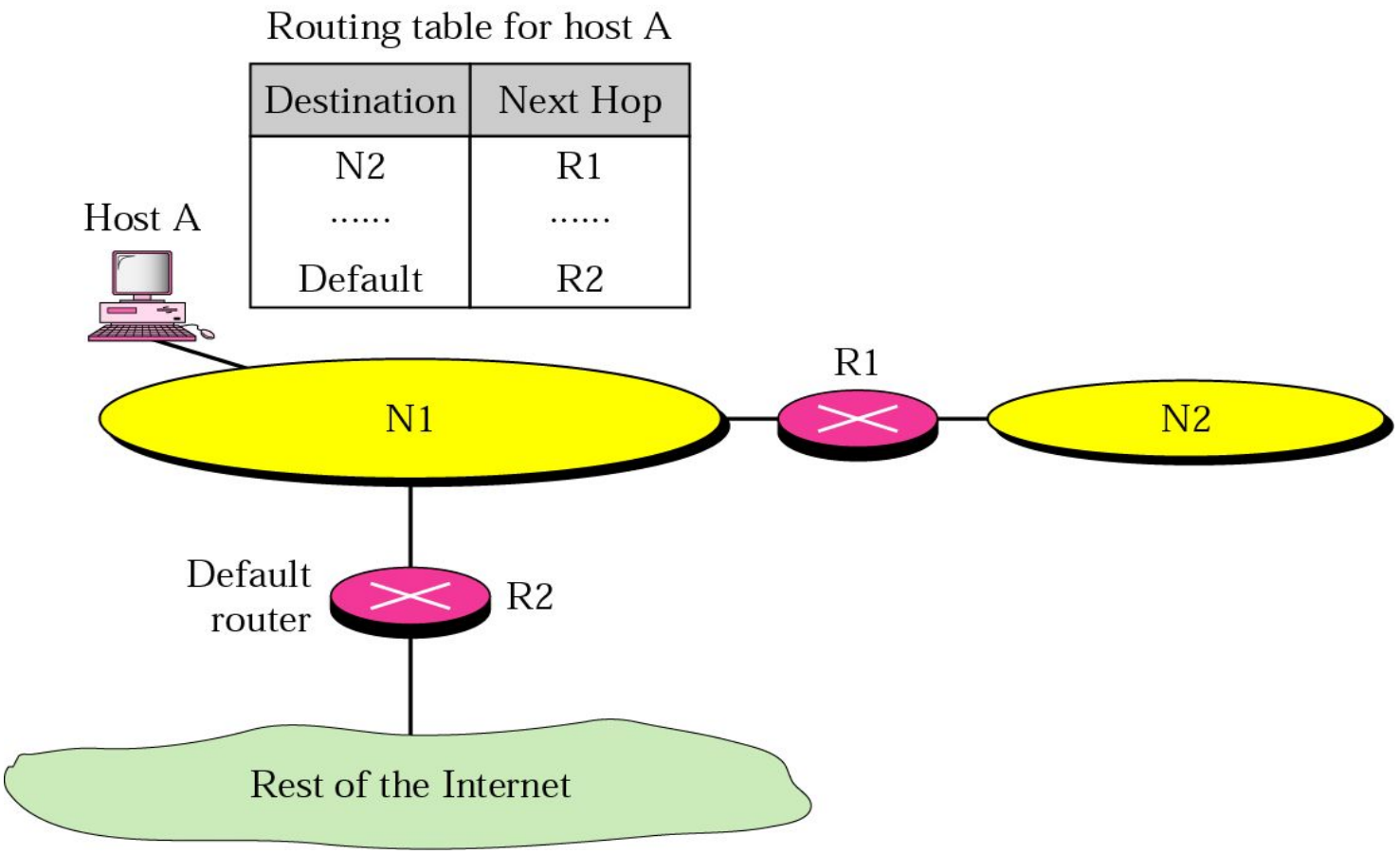


Figure 19.32 Classful addressing routing table

	Mask	Destination address	Next-hop address	Interface
	/8	14.0.0.0	118.45.23.8	m1
Host-specific →	/32	192.16.7.1	202.45.9.3	m0
	/24	193.14.5.0	84.78.4.12	m2
Default →	/0	/0	145.11.10.6	m0

Example 10

Using the table in Figure 19.32, the router receives a packet for destination 192.16.7.1. For each row, the mask is applied to the destination address until a match with the destination address is found. In this example, the router sends the packet through interface m0 (host specific).

Example 11

Using the table in Figure 19.32, the router receives a packet for destination 193.14.5.22. For each row, the mask is applied to the destination address until a match with the next-hop address is found. In this example, the router sends the packet through interface m2 (network specific).

Example 12

Using the table in Figure 19.32, the router receives a packet for destination 200.34.12.34. For each row, the mask is applied to the destination address, but no match is found. In this example, the router sends the packet through the default interface m0.

Routing Algorithms

1. Distance Vector Routing
2. Link State Routing

Figure 21-18

The Concept of Distance Vector Routing

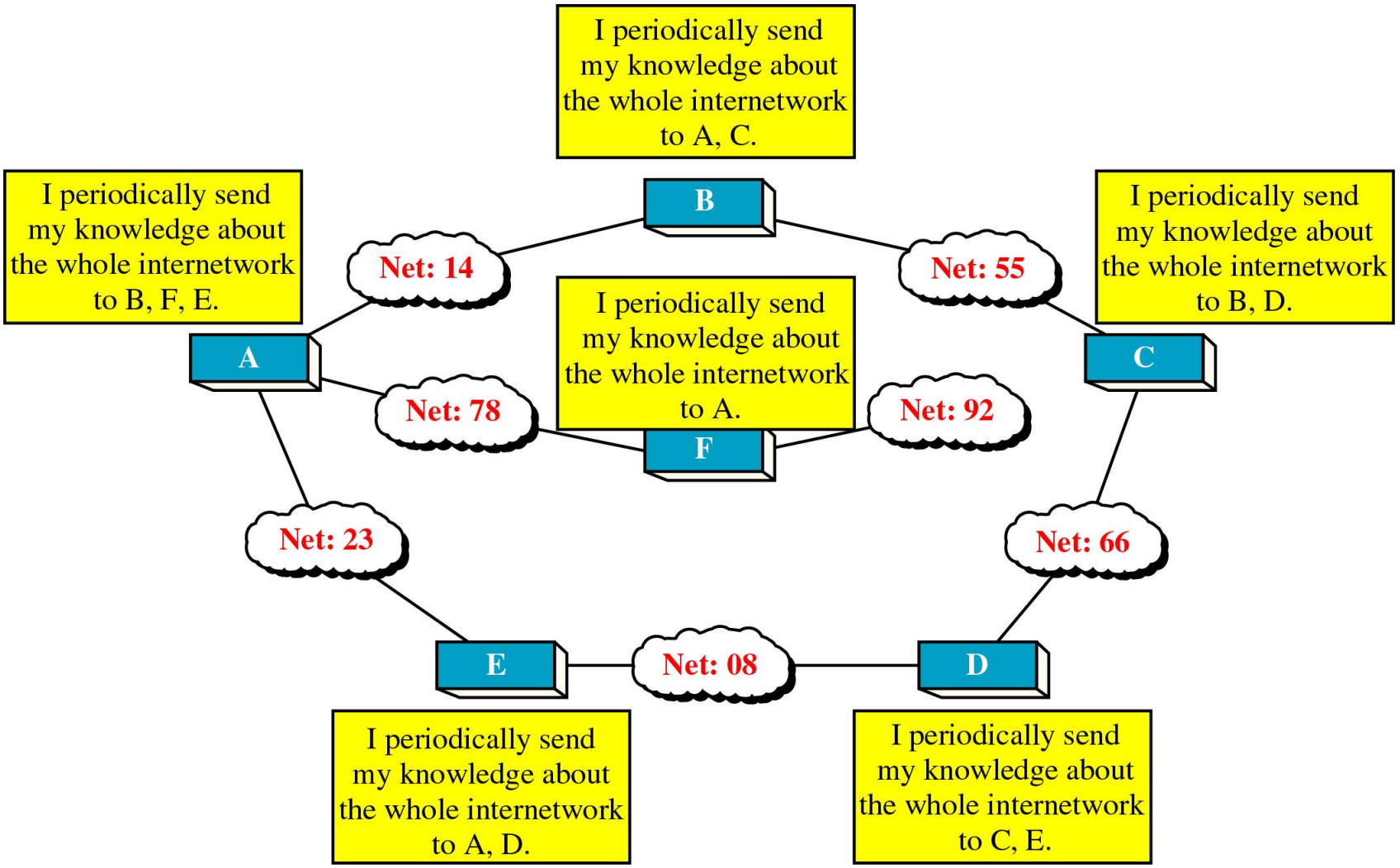


Figure 21-19

Distance Vector Routing Table

Network ID	Cost	Next Hop
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .

Figure 21-20

Routing Table Distribution

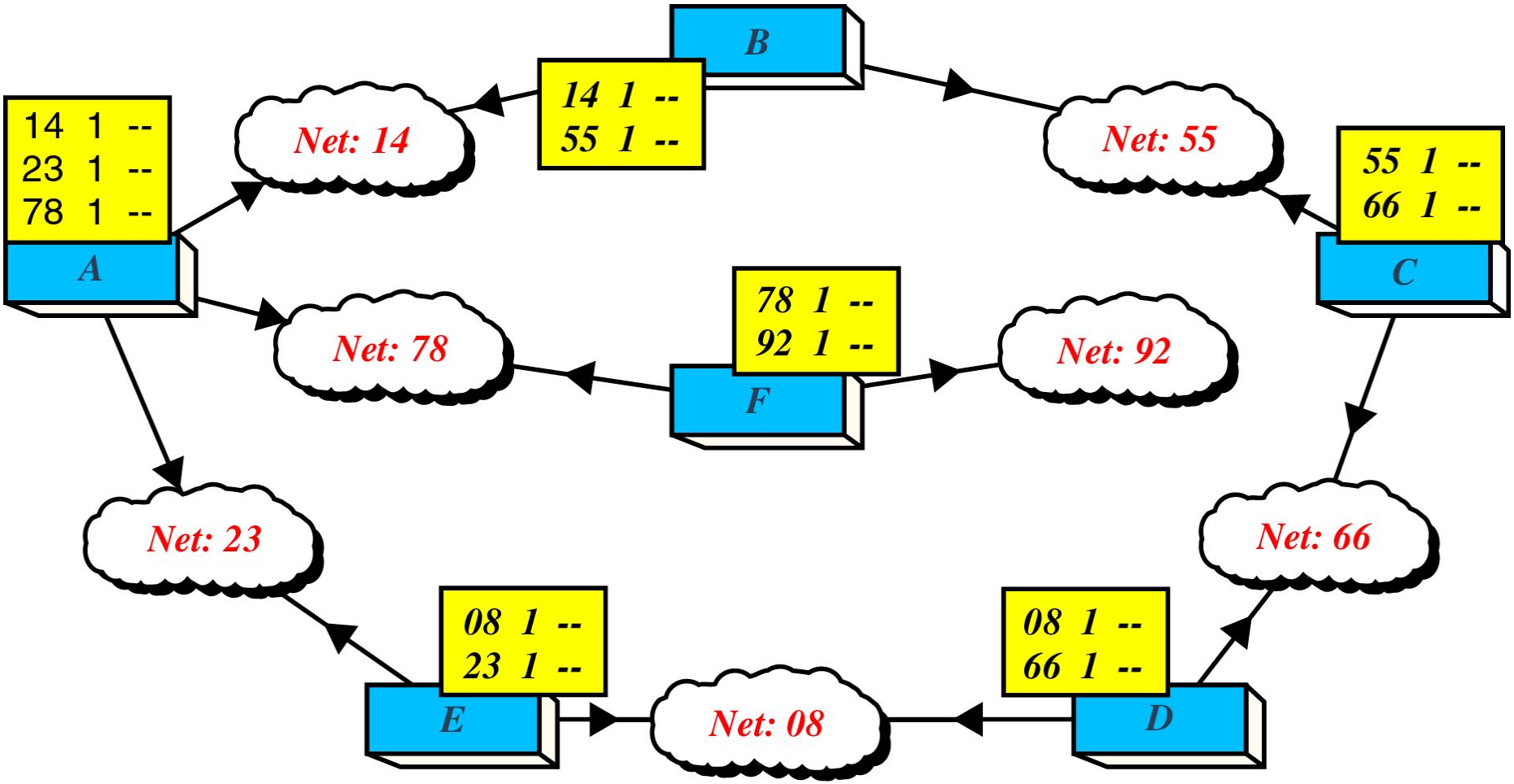


Figure 21-21

Updating Routing Table for Router A

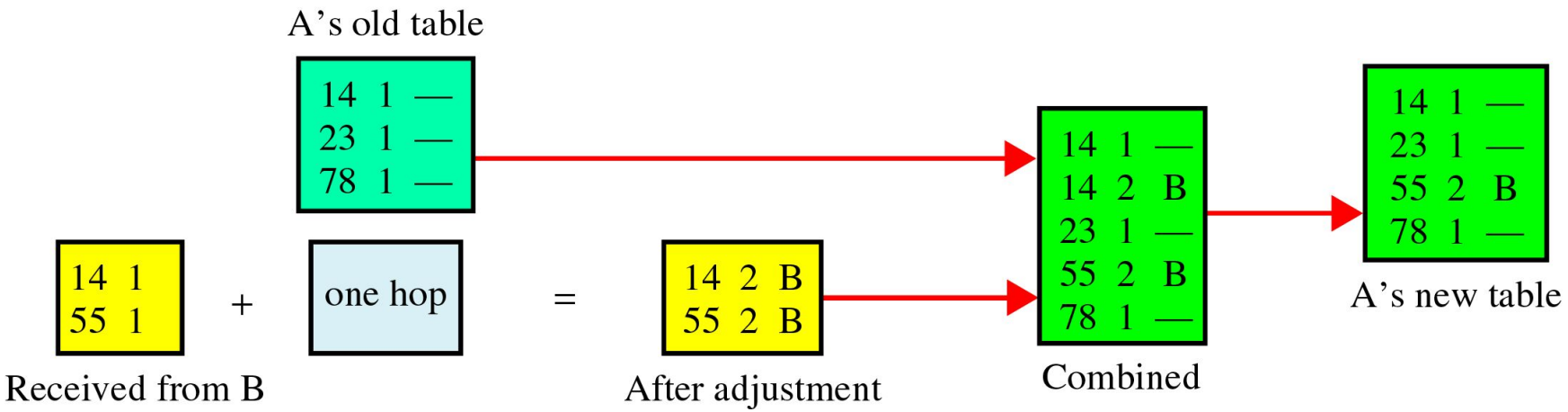


Figure 21-22

Final Routing Tables

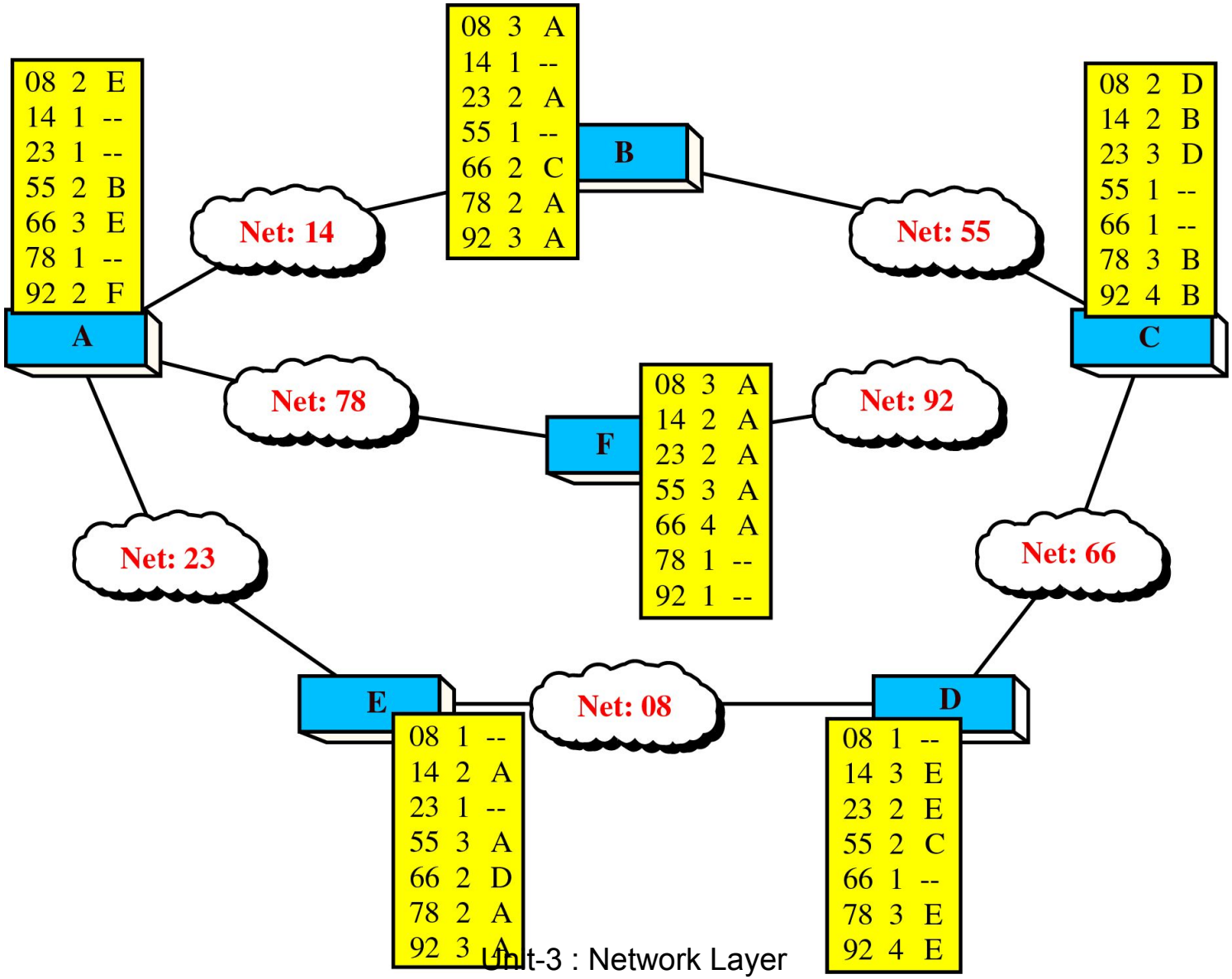
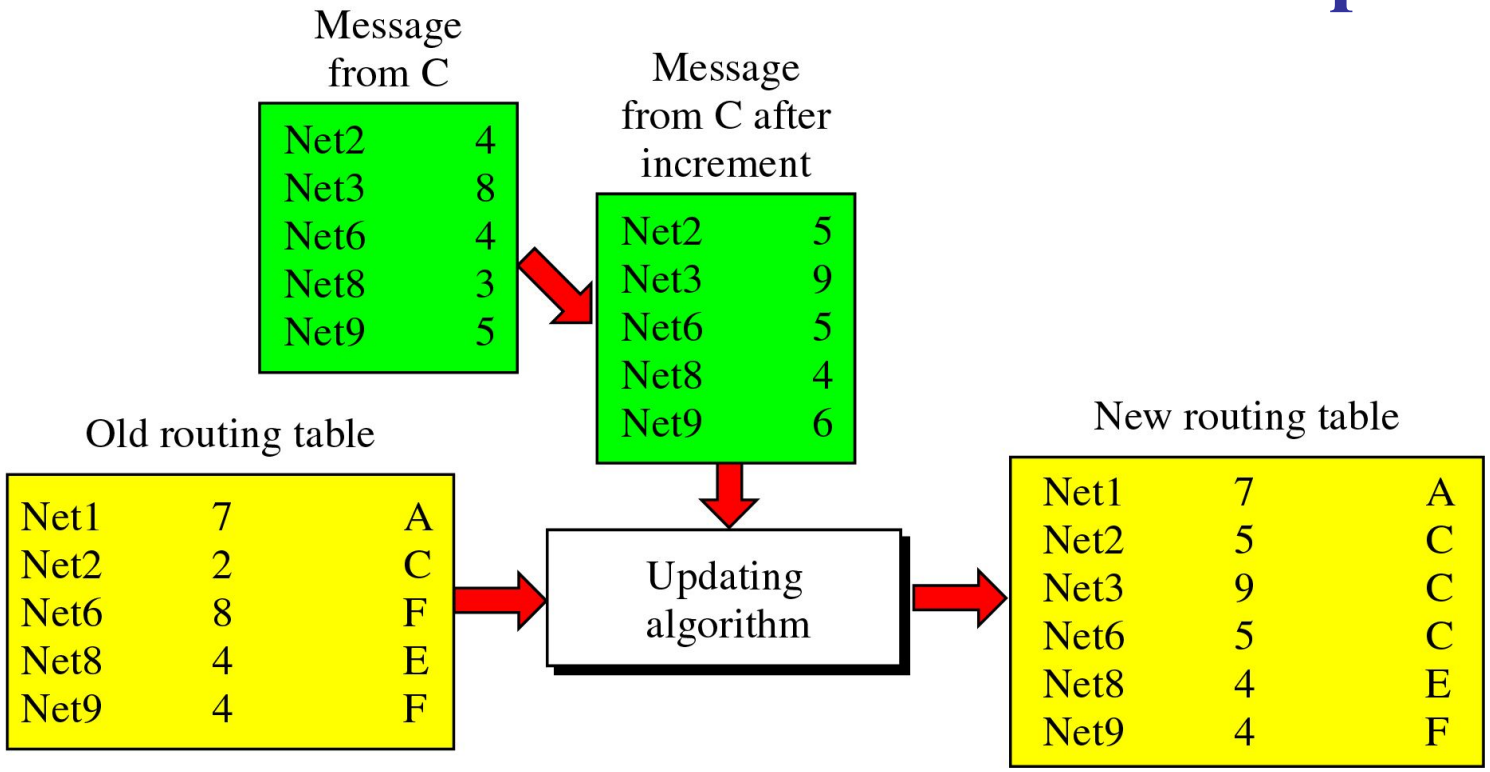


Figure 21-23

Example 21.1



Rules

Net2: Replace **(Rule 2.a)**
Net3: Add **(Rule 1)**
Net6: Replace **(Rule 2.b.i)**
Net8: No change **(Rule 2.b.ii)**
Net9: No change **(Rule 2.b.ii)**

Note that there is no news about Net1 in the advertised message, so none of the rules apply to this entry.

Figure 21-24

Concept of Link State Routing

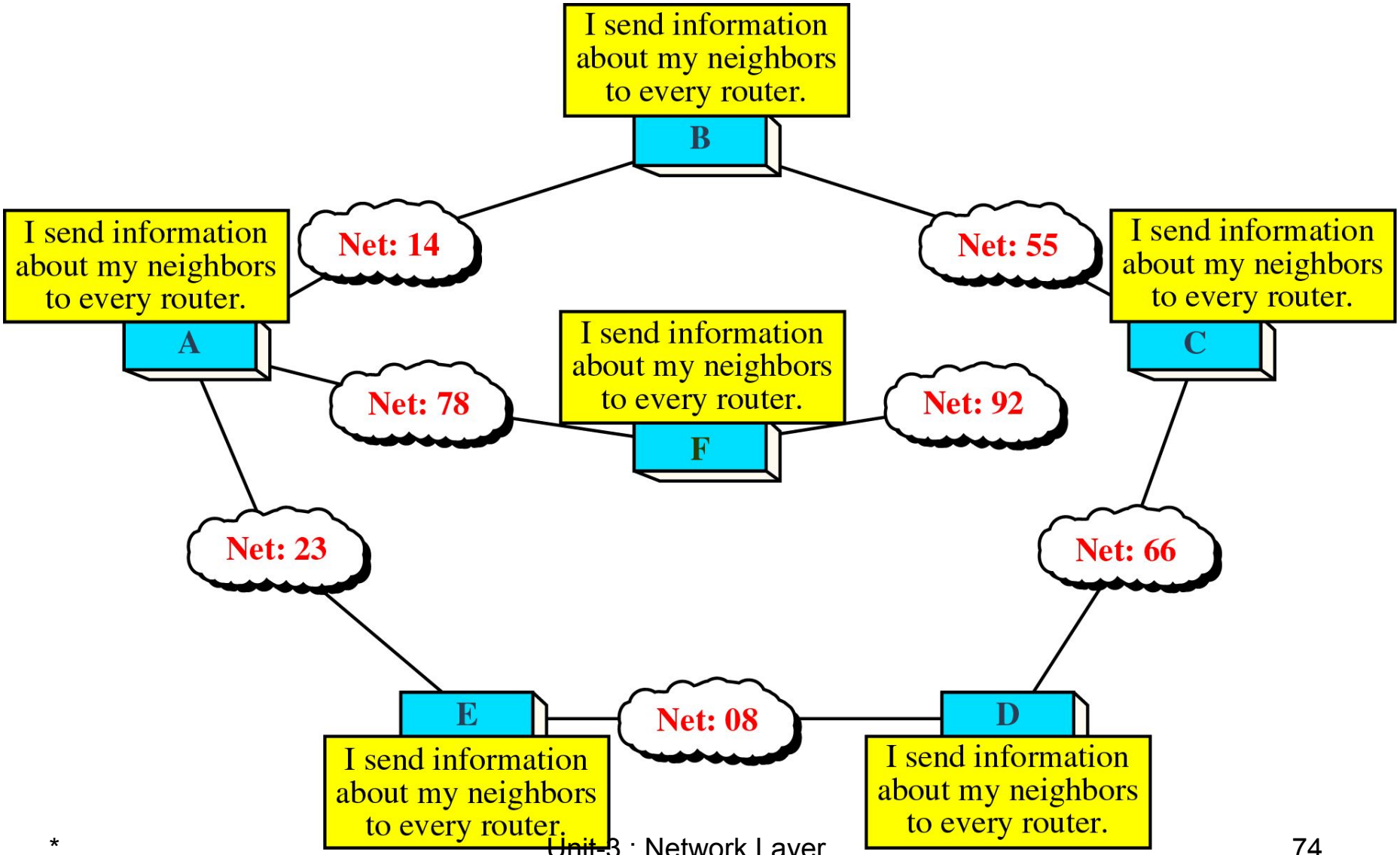


Figure 21-25

Cost in Link State Routing

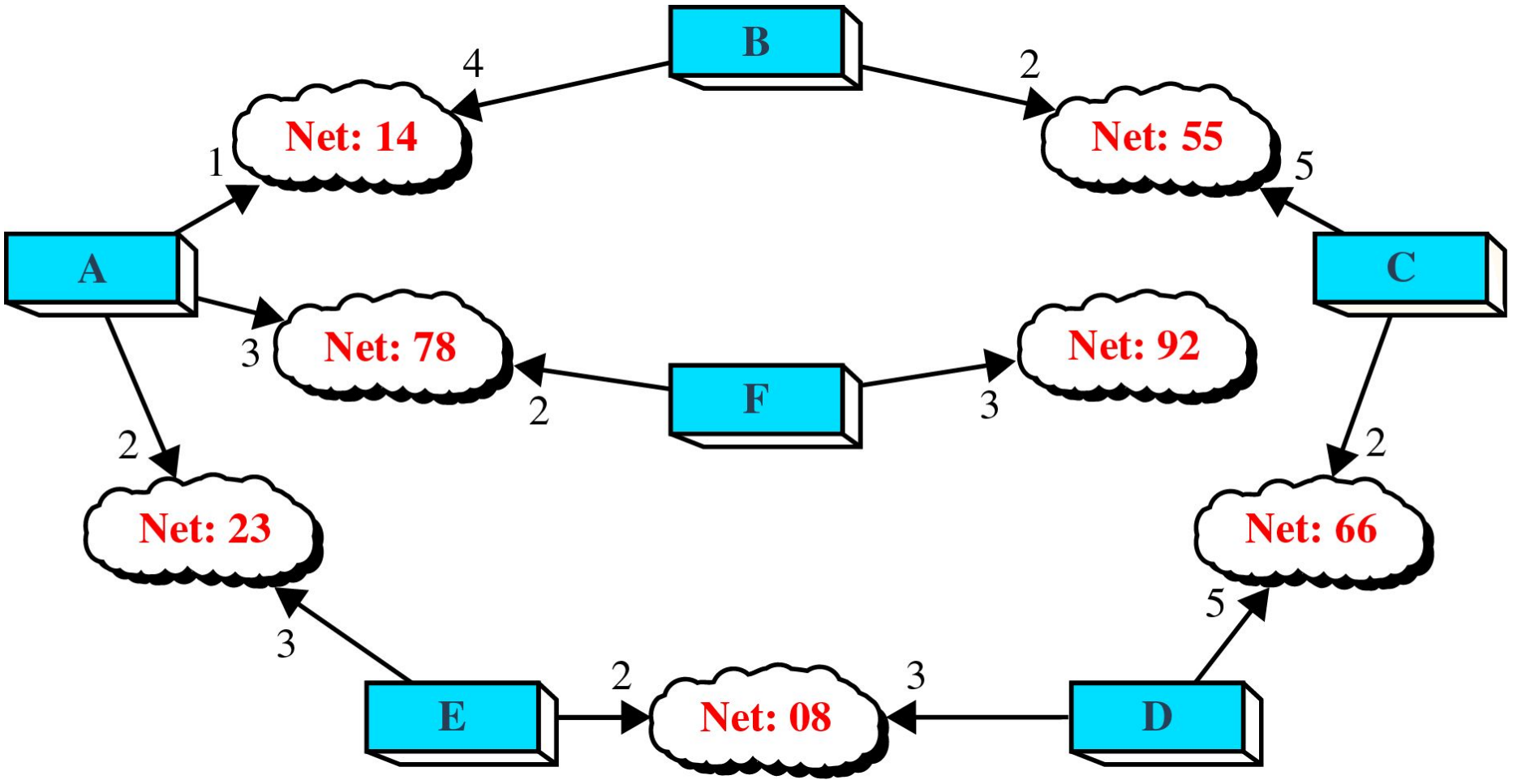


Figure 21-26

Link State Packet

Advertiser	Network	Cost	Neighbor
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .
. . . . .	. . . . .	. . . . .	. . . . .

Figure 21-27

Flooding of A's LSP

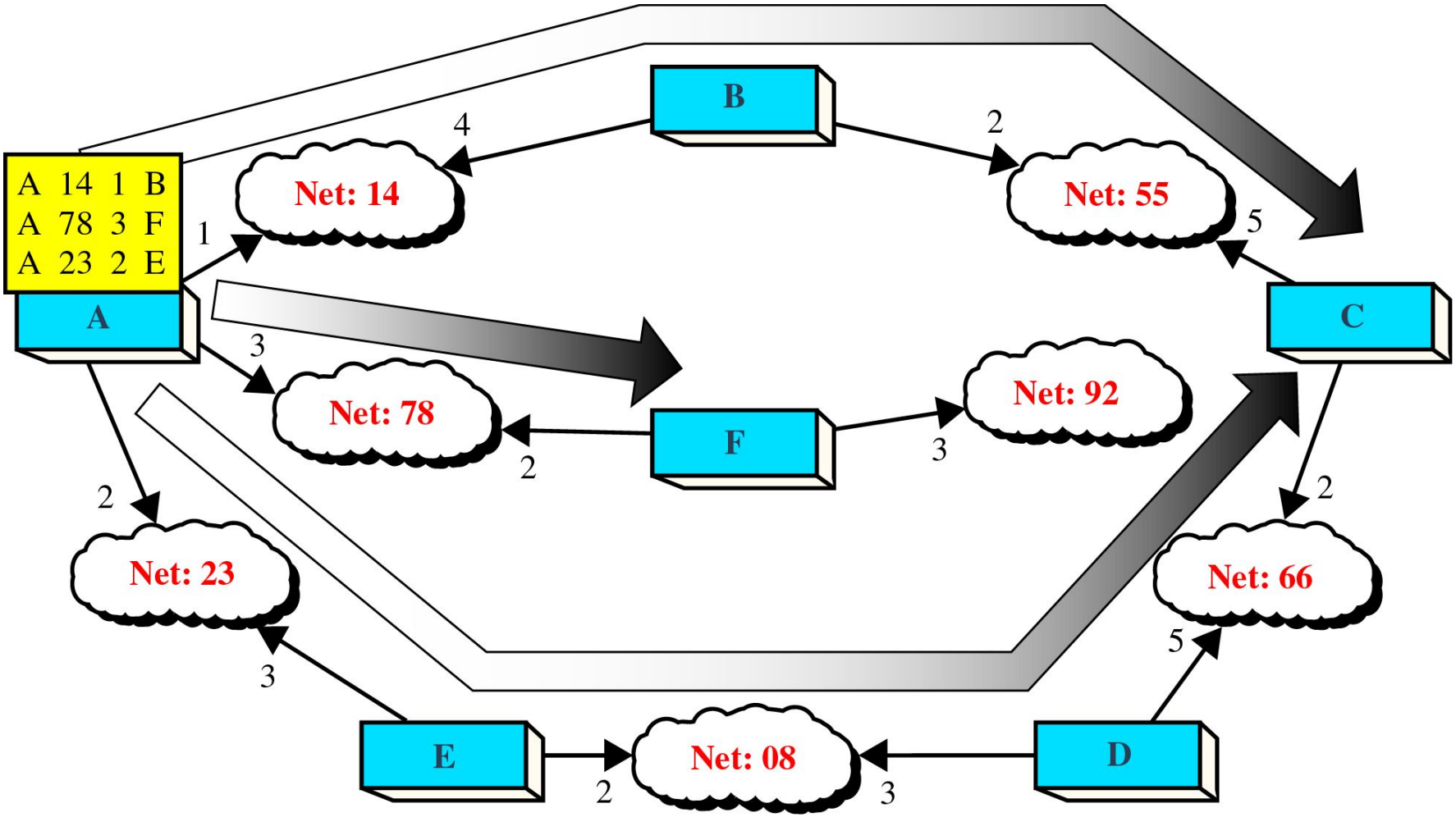


Figure 21-28

Link State Database

Advertiser	Network	Cost	Neighbor
A	14	1	B
A	78	3	F
A	23	2	E
B	**14**	**4**	**A**
B	**55**	**2**	**C**
C	55	5	B
C	66	2	D
D	**66**	**5**	**C**
D	**08**	**3**	**E**
E	23	3	A
E	08	2	D
F	**78**	**2**	**A**
F	**92**	**3**	**—**

Figure 21-29

Costs in the Dijkstra Algorithm

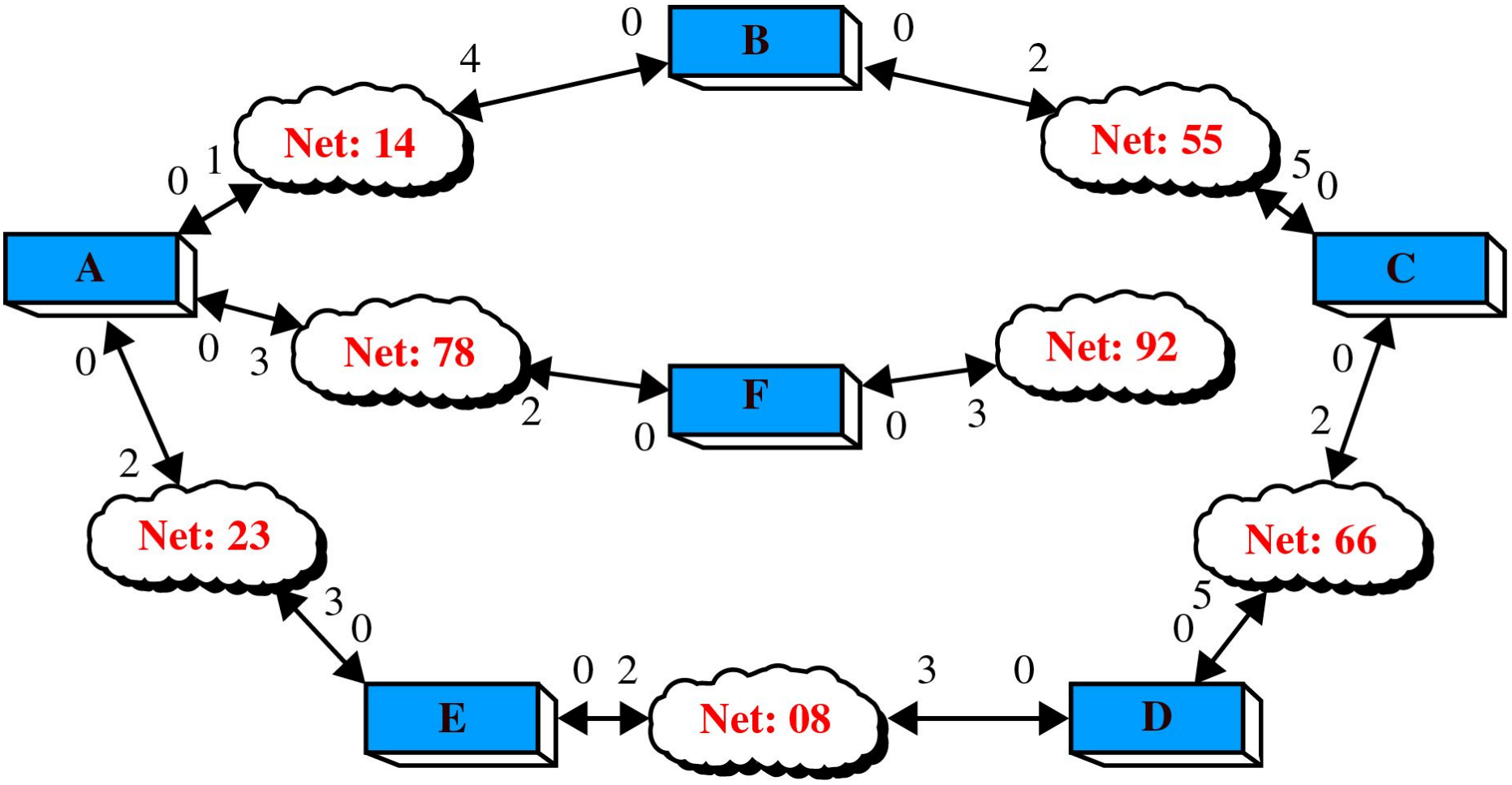
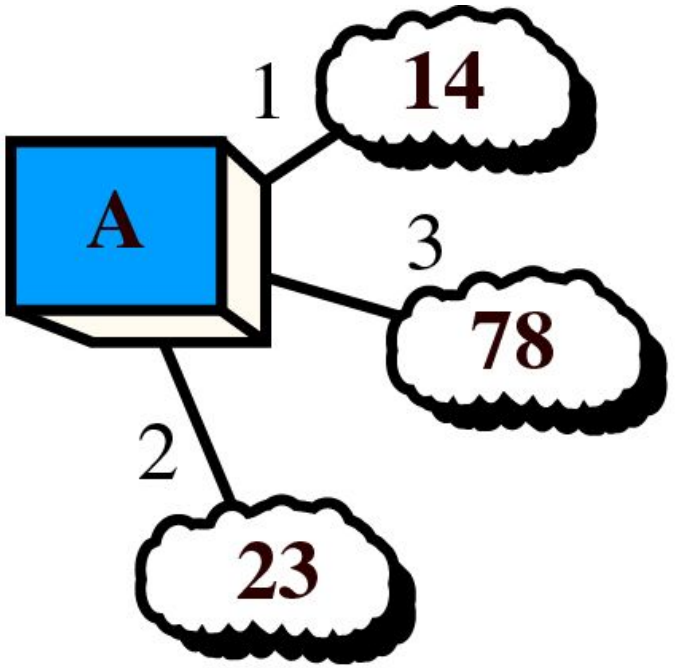


Figure 21-30, Part I

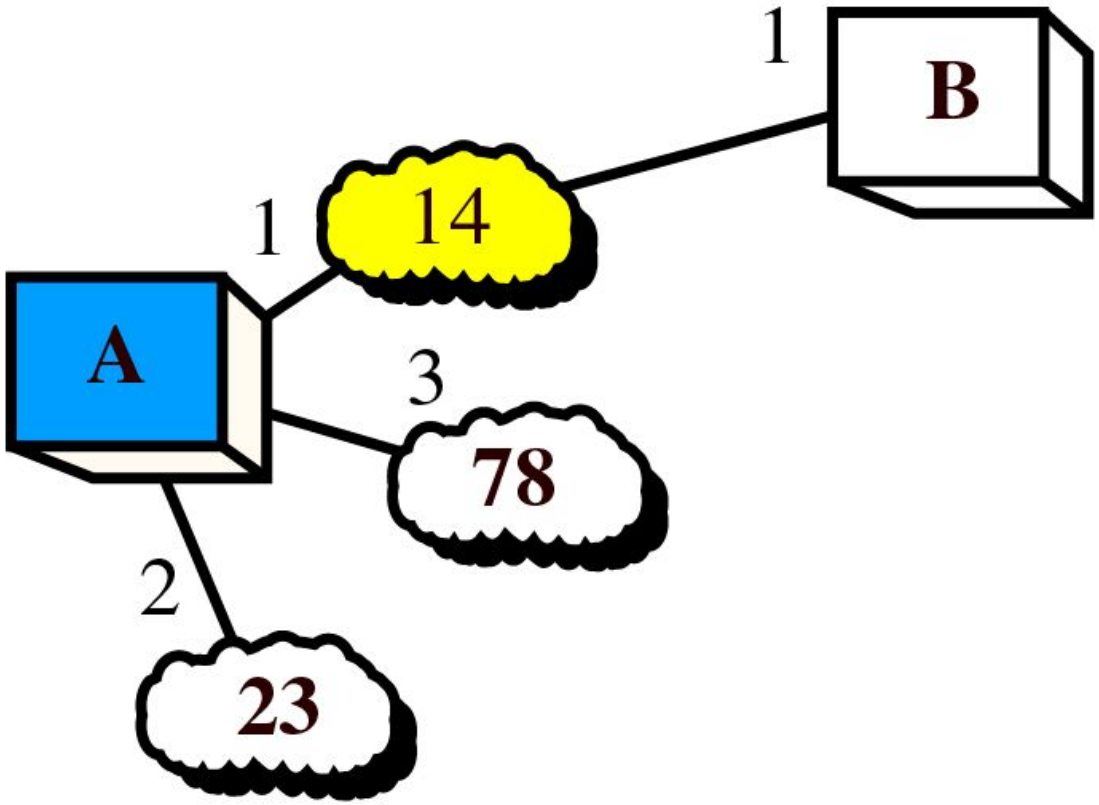
Shortest Path Calculation, Part I



Root is A, networks
14, 78, 23 added

Figure 21-30, Part II

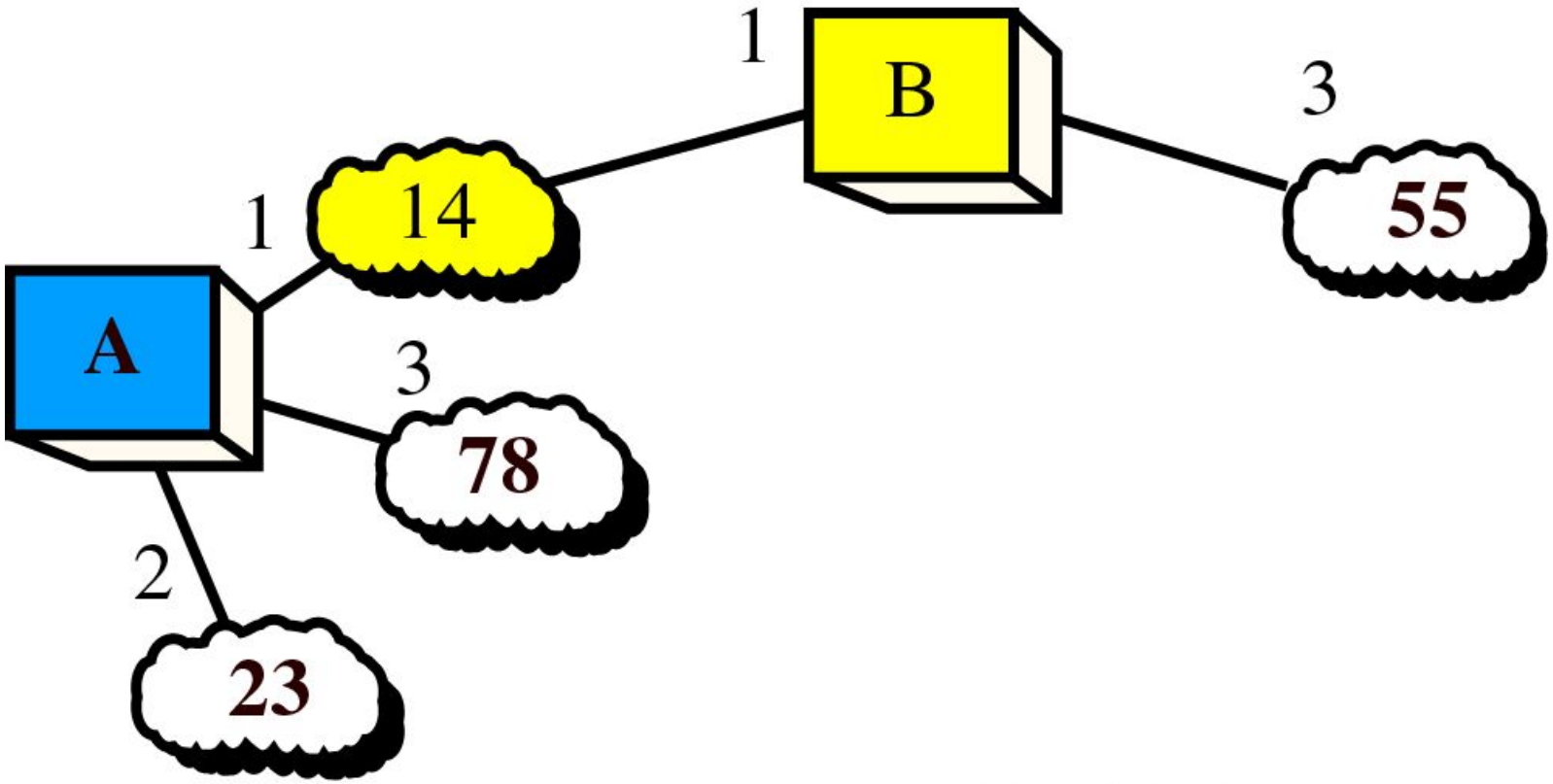
Shortest Path Calculation, Part II



14 permanent, B added

Figure 21-30, Part III

Shortest Path Calculation, Part III



B Permanent, 55 added

Figure 21-30, Part IV

Shortest Path Calculation, Part IV

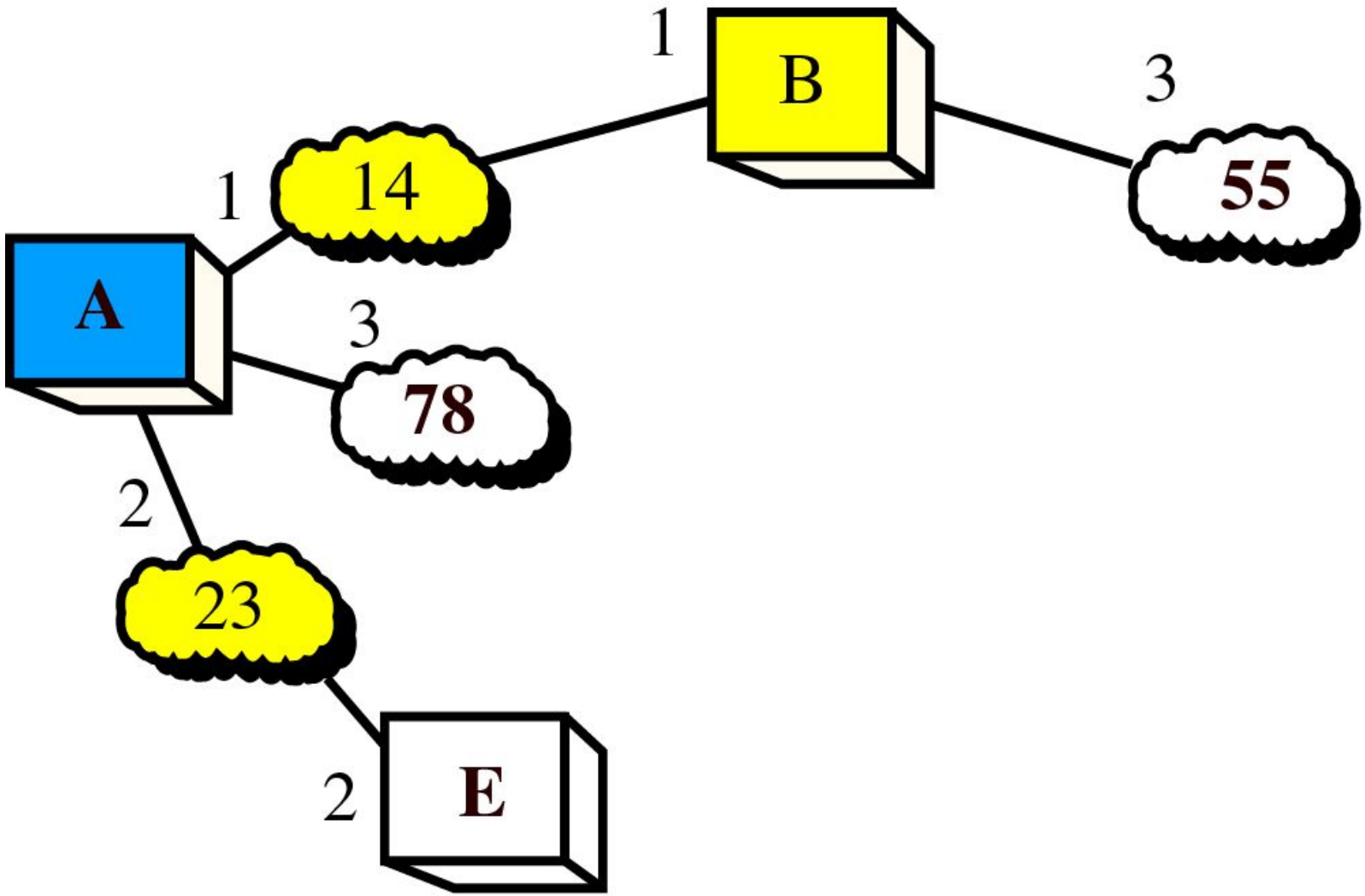


Figure 21-30, Part V

Shortest Path Calculation, Part V

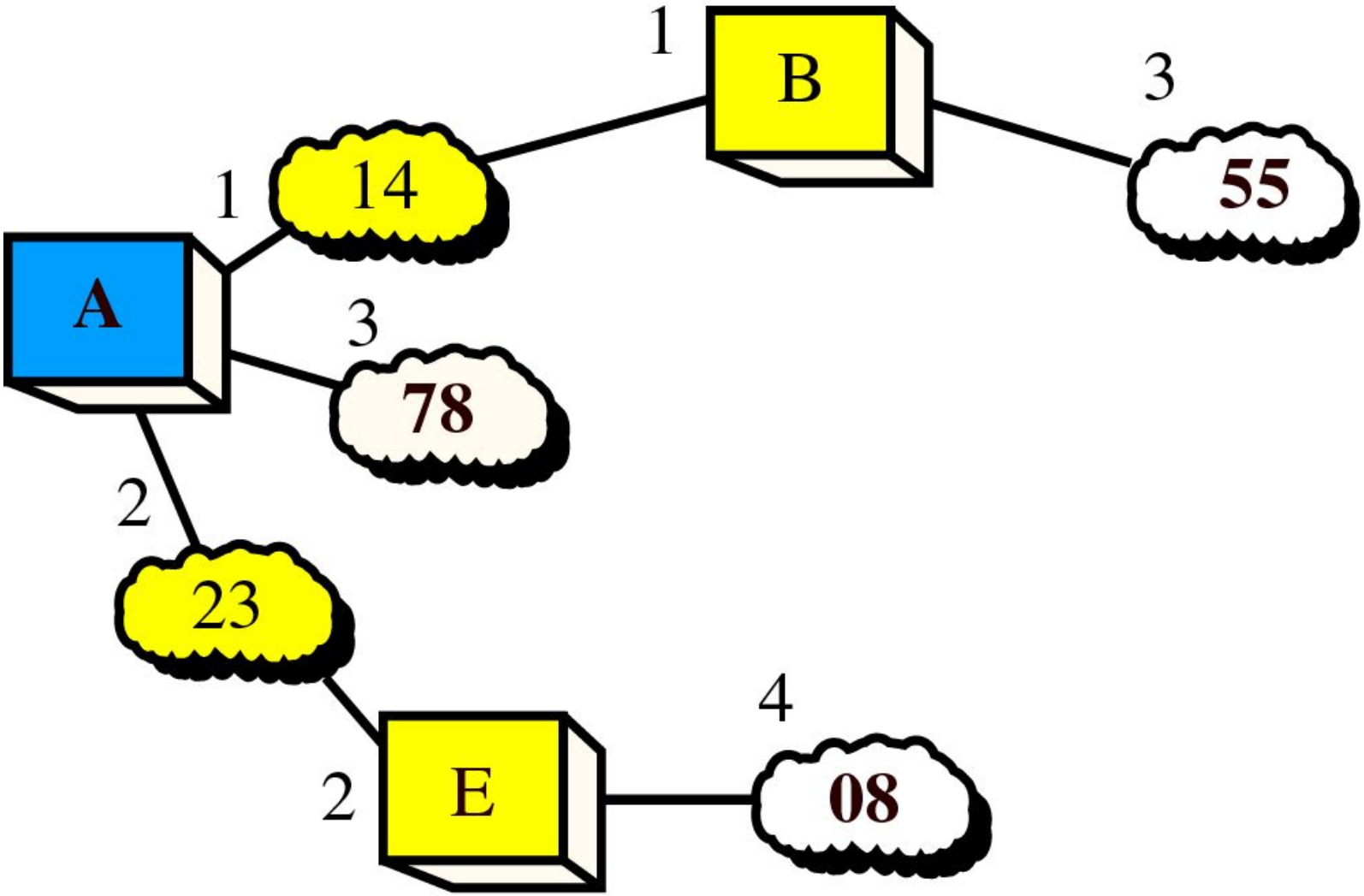


Figure 21-30, Part VI

Shortest Path Calculation, Part VI

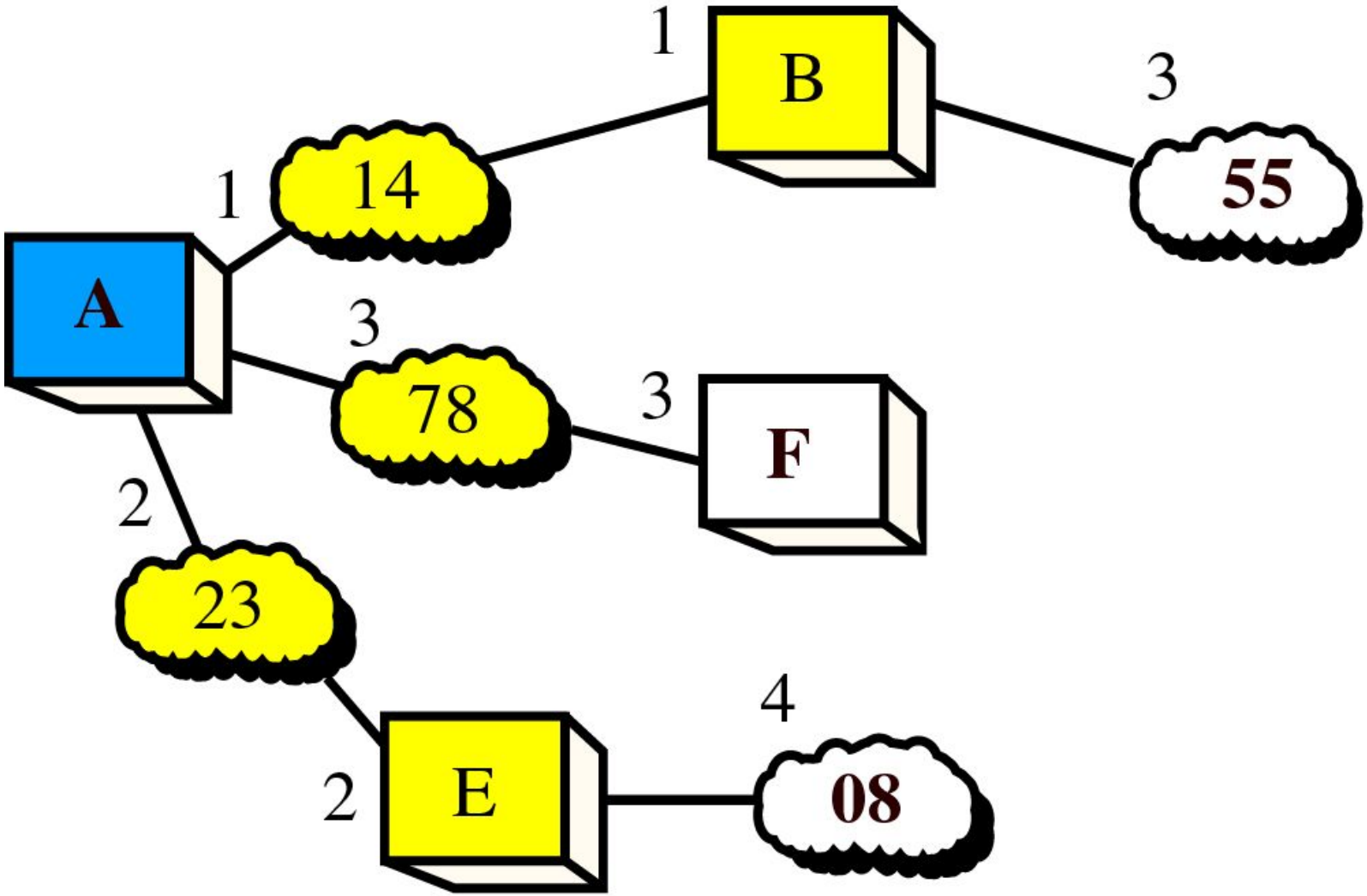


Figure 21-31, Part VII

Shortest Path Calculation, Part VII

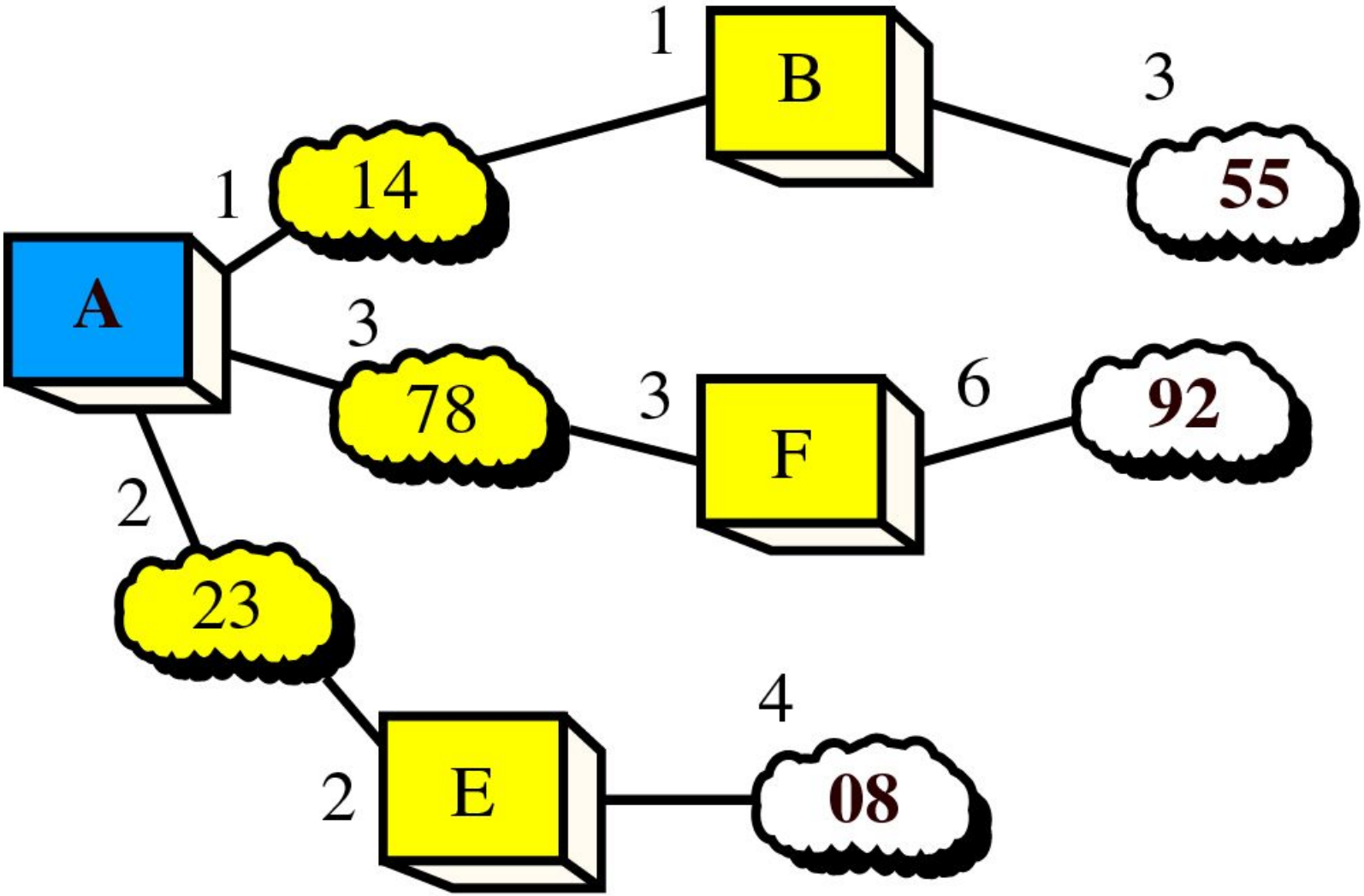


Figure 21-31, Part I

Shortest Path Calculation, Part VIII

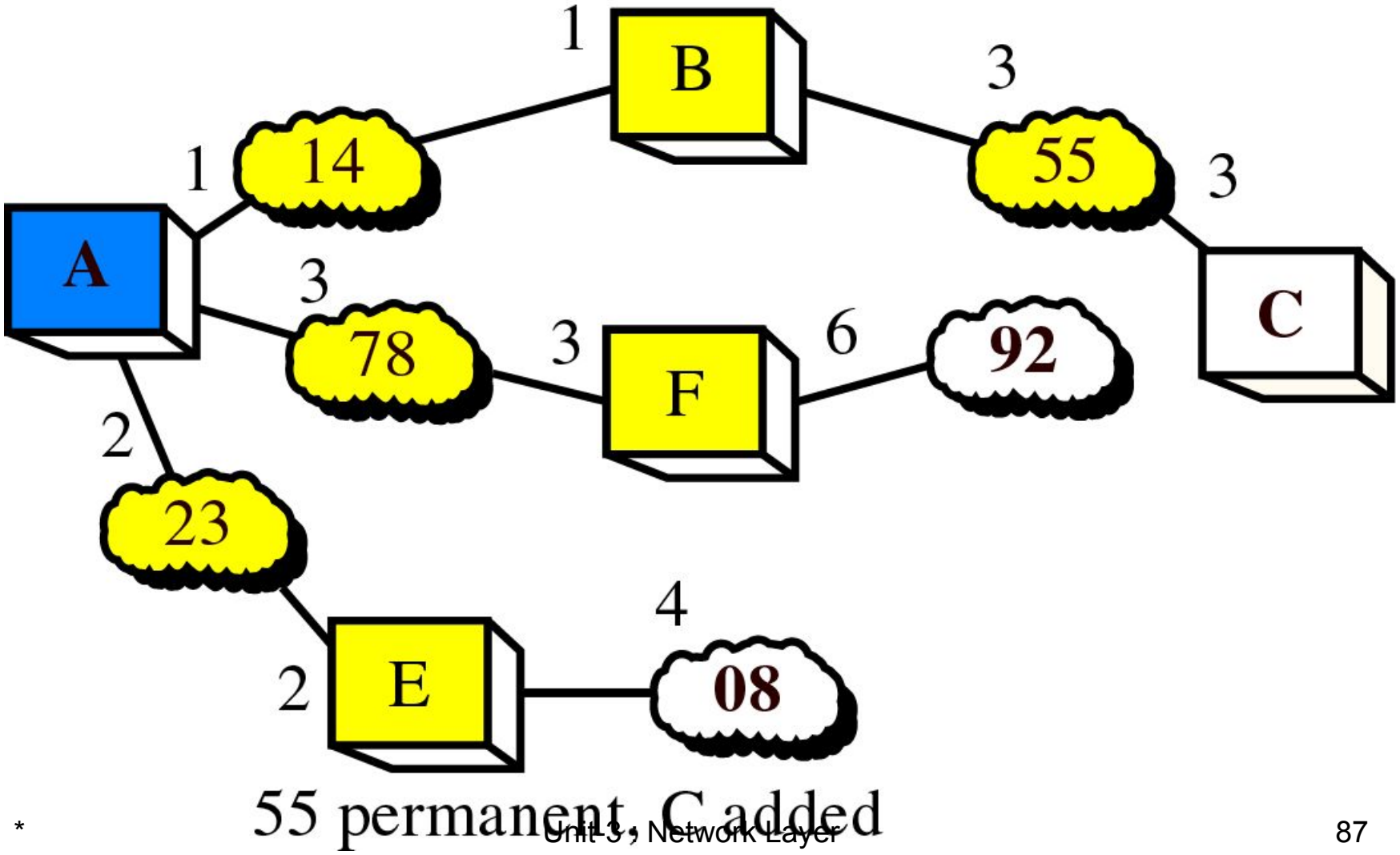


Figure 21-31, Part II

Shortest Path Calculation, Part IX

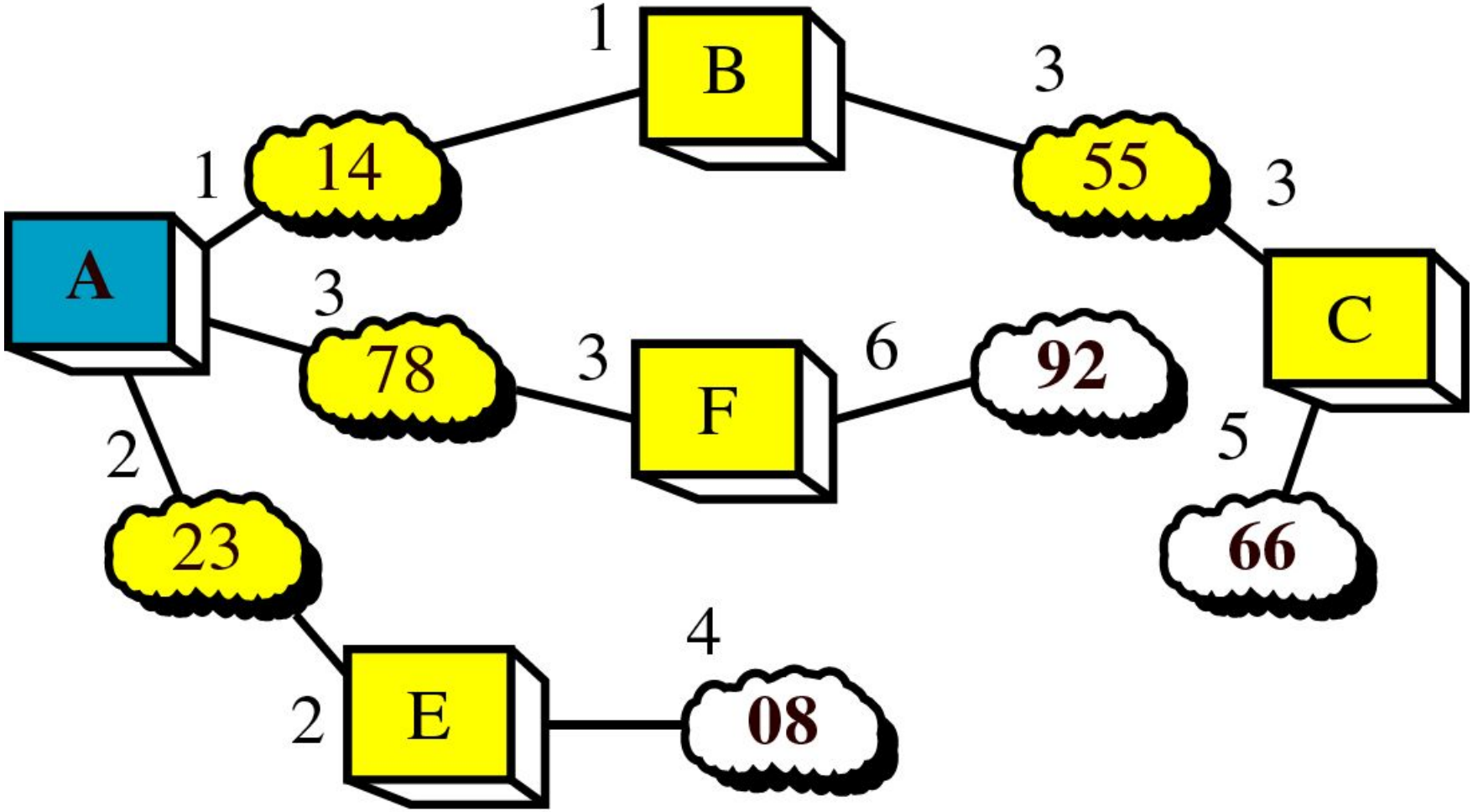
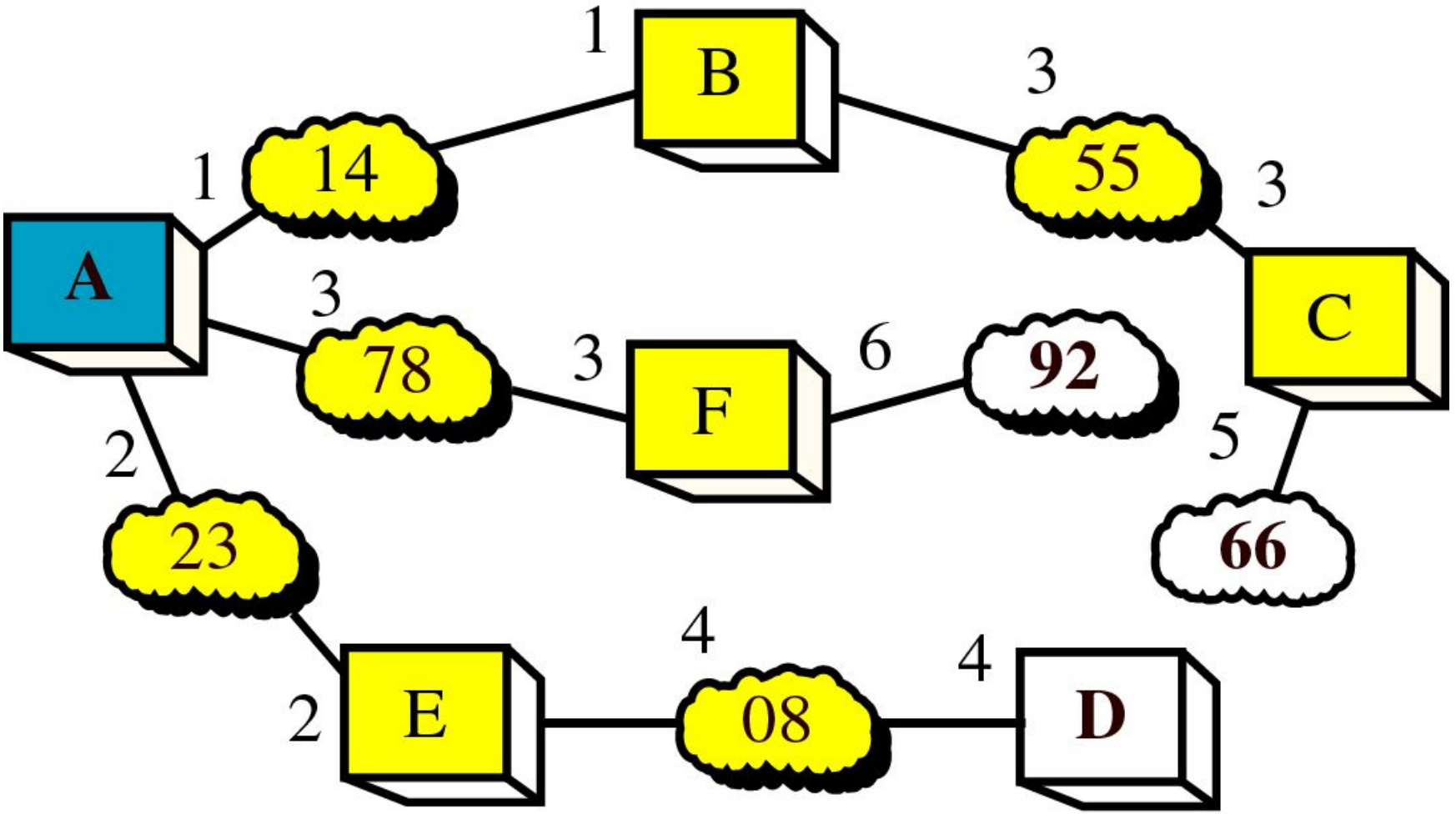


Figure 21-31, Part III

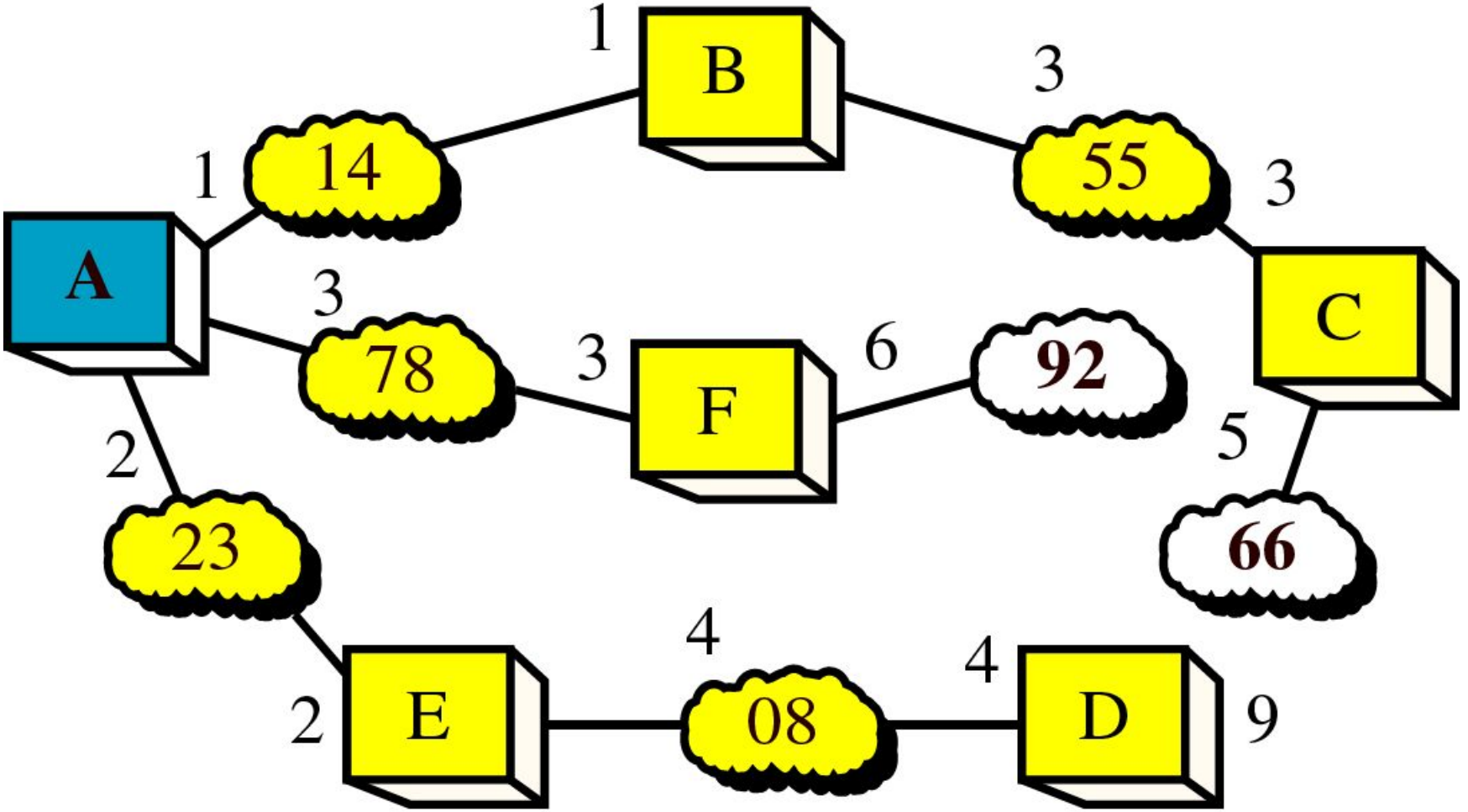
Shortest Path Calculation, Part X



08 permanent, D added

Figure 21-31, Part IV

Shortest Path Calculation, Part XI



D permanent, 66 added.

But 9 > 5, so that link deleted

Figure 21-31, Part V

Shortest Path Calculation, Part XII

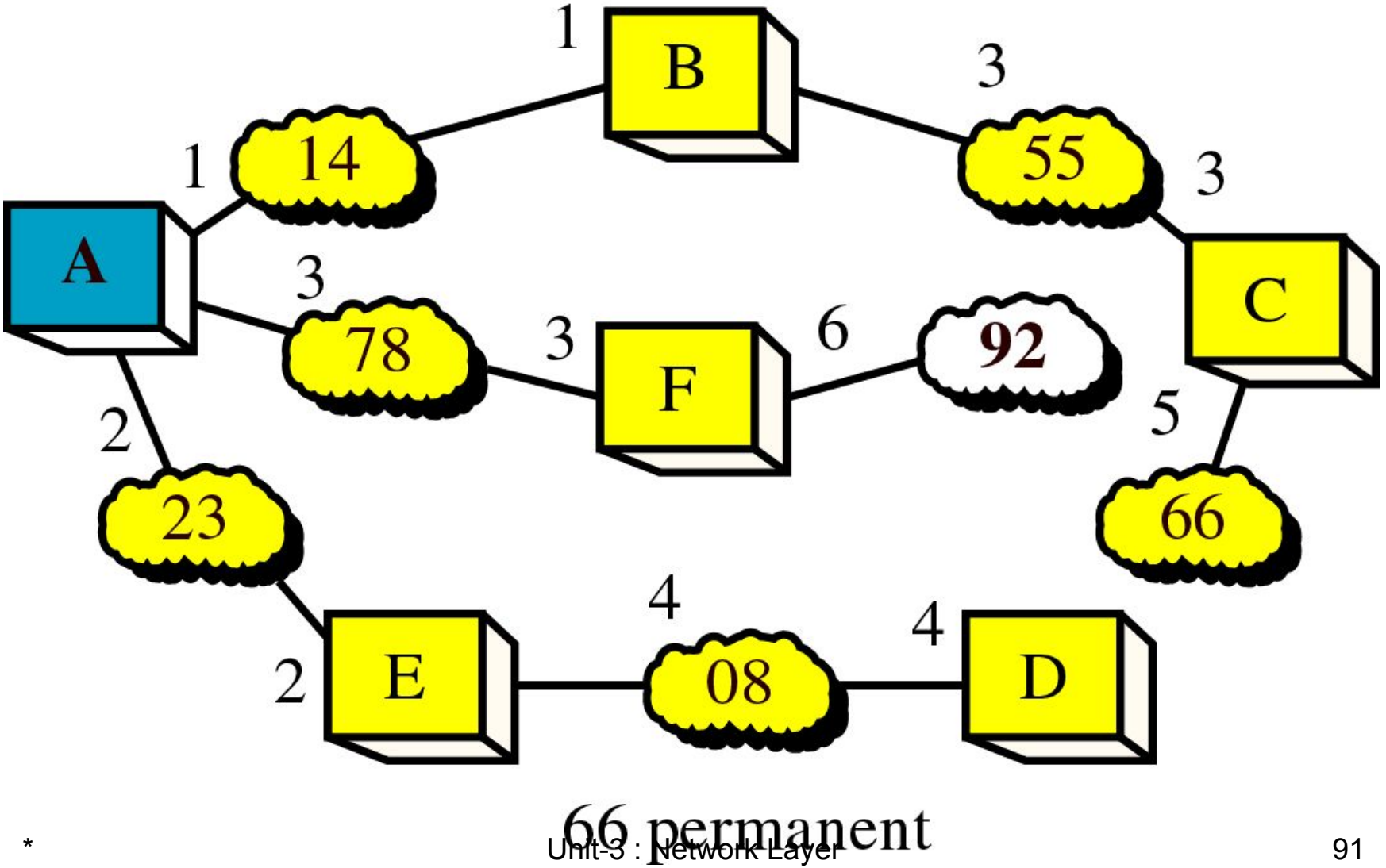


Figure 21-31, Part VI

Shortest Path Calculation, Part XIII

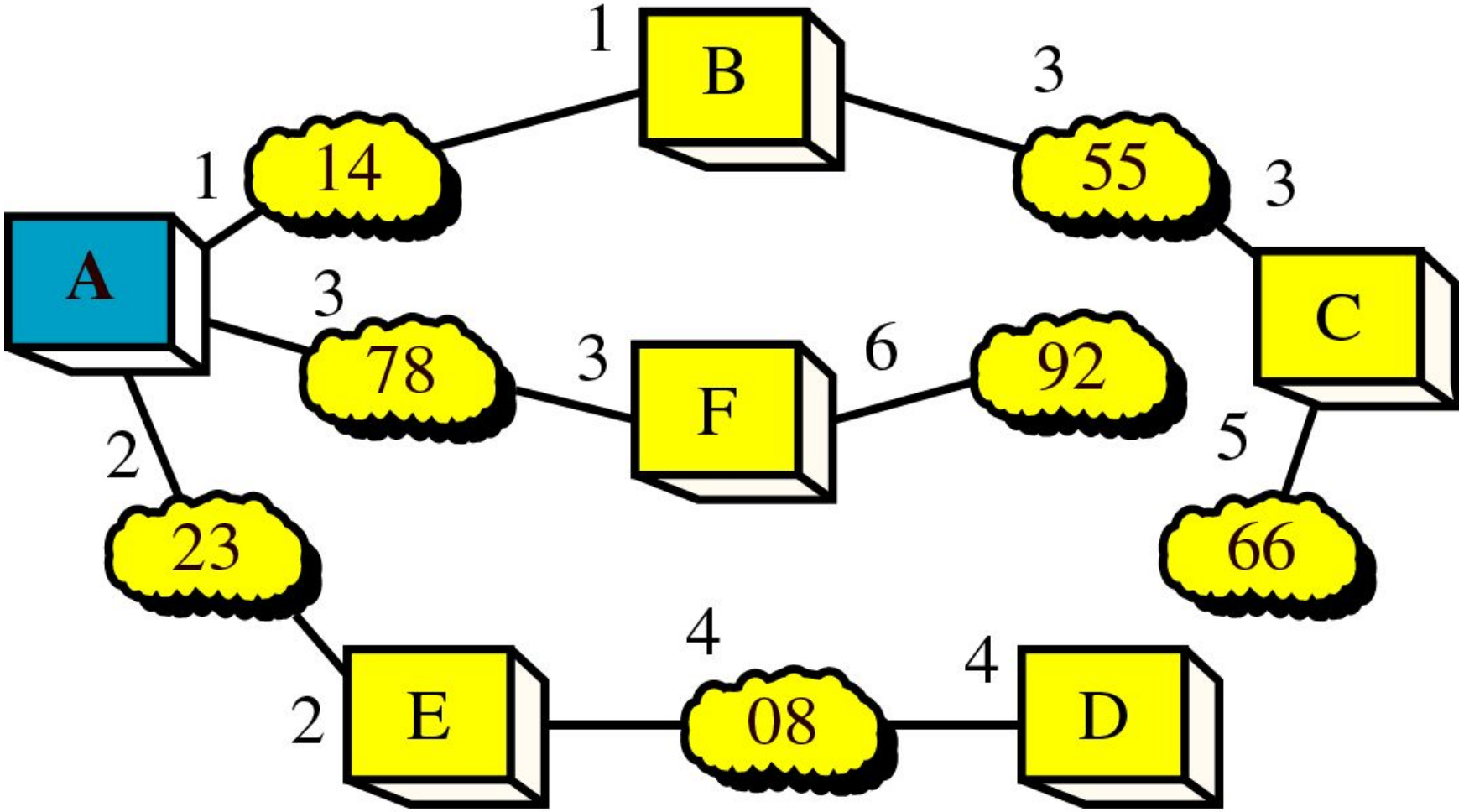


Figure 21-32

Routing Table for Router A

Net	Cost	Next router
08	4	E
14	1	--
23	2	--
55	3	B
66	5	B
78	3	--
92	6	F

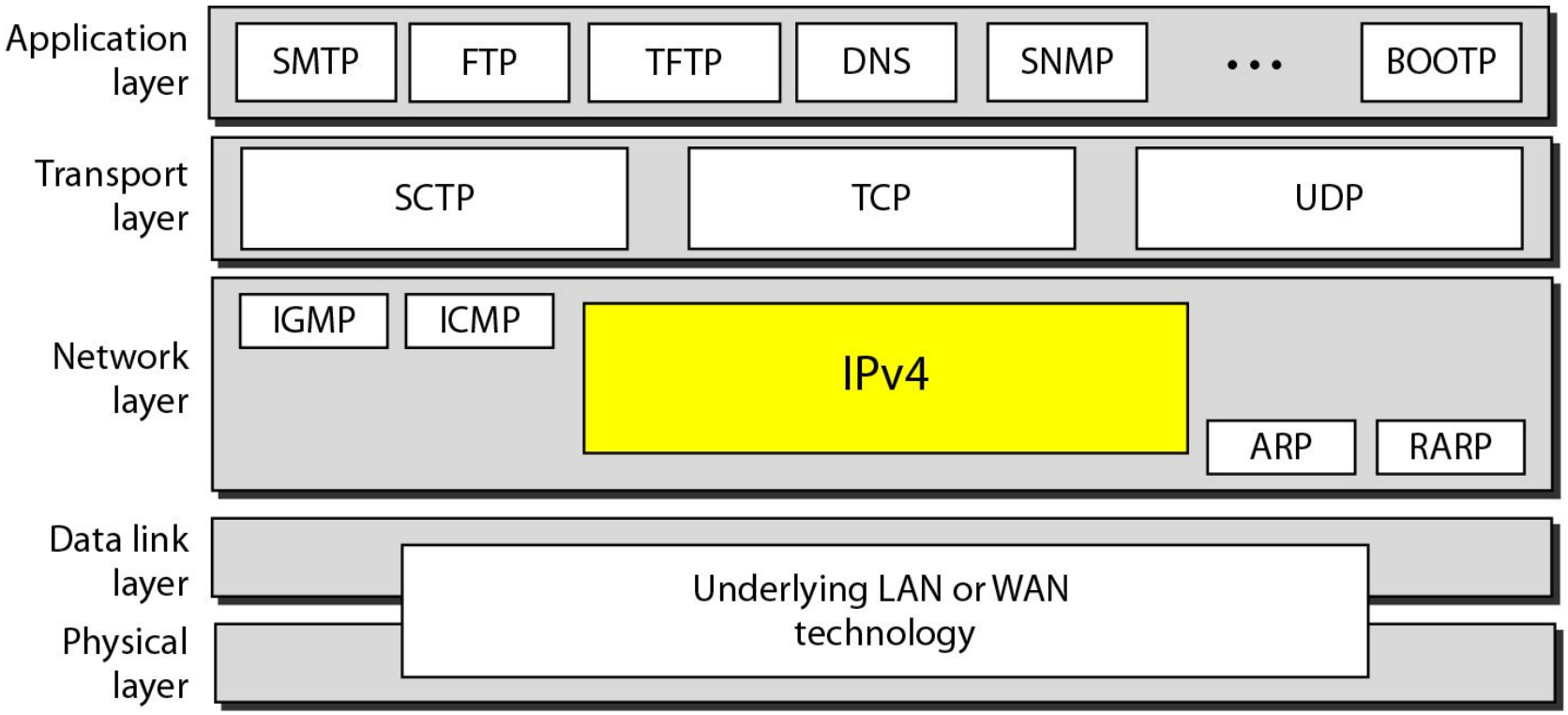
20-2 IPv4

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

Topics discussed in this section:

Datagram
Fragmentation
Checksum
Options

Figure 20.4 *Position of IPv4 in TCP/IP protocol suite*



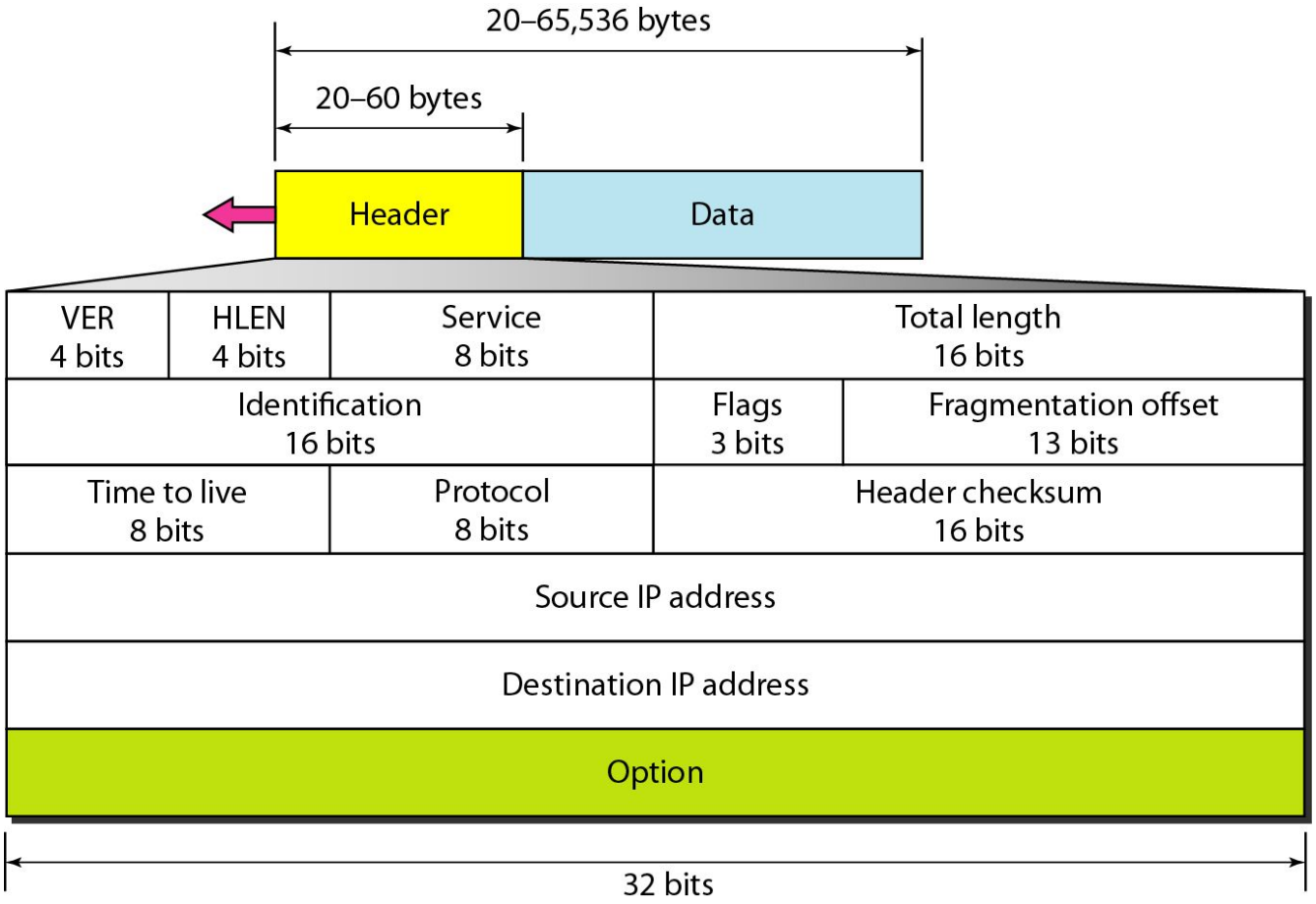
Note

IPv4 is an unreliable and connectionless datagram protocol – a best effort delivery

Best effort means that IPv4 provides no error control (except for error detection on the header) or flow control

IPv4 does its best to get a transmission through to its destination, but with no guarantees

Figure 20.5 *IPv4 datagram format*



IPv4 Datagram Format

- Version (VER): version of the IP protocol. Currently, the version is 4.
- Header length (HLEN): the total length of the datagram header in 4-byte words.
- Services: service type or differentiated services (not used now).
- Total length: total length (header plus data) of the datagram in bytes.
  - Total length of data = total length – header length

IPv4 Datagram Format

- Identification: used in fragmentation (discussed later).
- Flags: used in fragmentation (discussed later).
- Fragmentation offset: used in fragmentation (discussed later).
- Time to live: it is used to control the maximum number hops visited by the datagram.
- Protocol: defines the higher-level protocol that uses the services of the IPV4 layer.

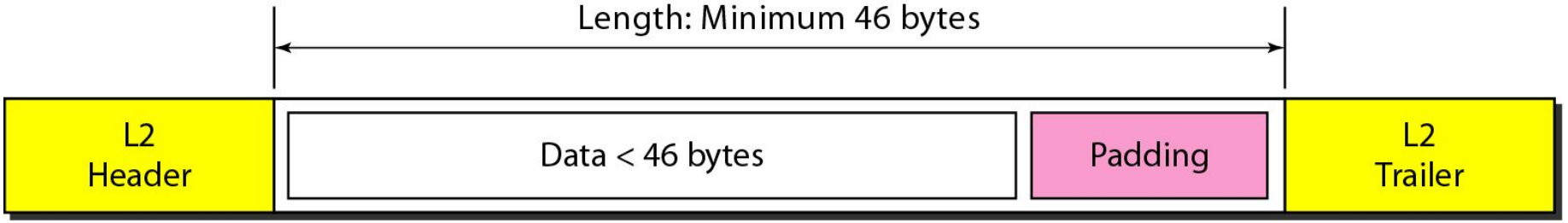
IPv4 Datagram Format

- Checksum: 1's compliment checksum (introduced in Chapter 10).
- Source address: is the IPv4 address of the source.
- Destination address: is the IPv4 address of the source.

Note

The total length field defines the total length of the datagram including the header.

Figure 20.7 *Encapsulation of a small datagram in an Ethernet frame*



One of the reason why “total length” field is required.

Figure 20.8 *Protocol field and encapsulated data*

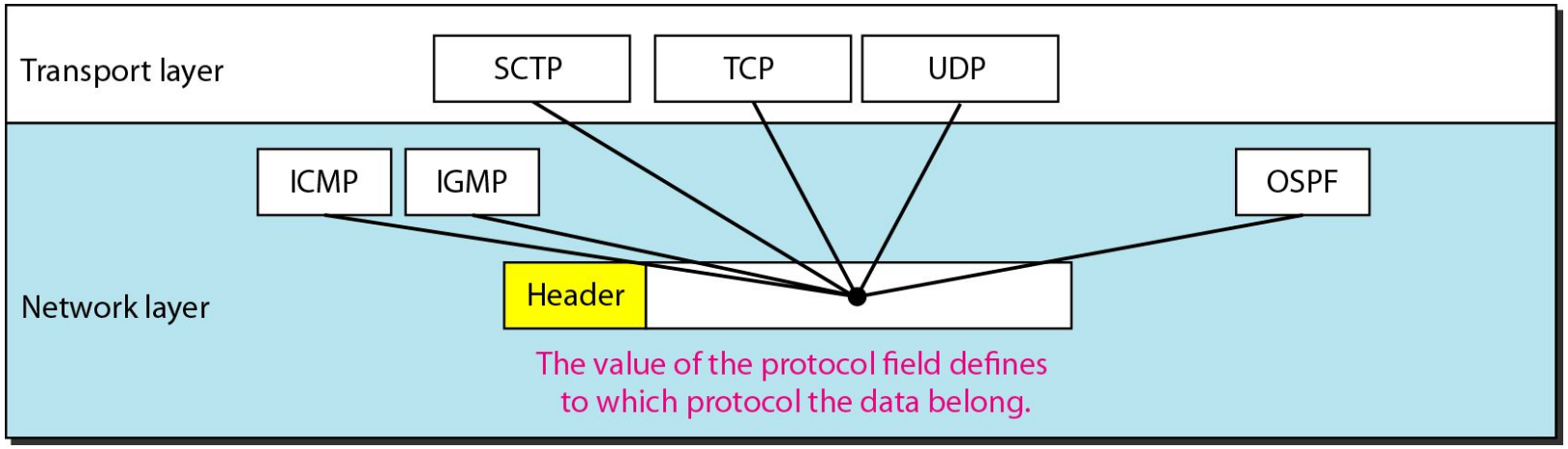


Table 20.4 *Protocol values*

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example 20.1

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Figure 20.9 *Maximum transfer unit (MTU)*

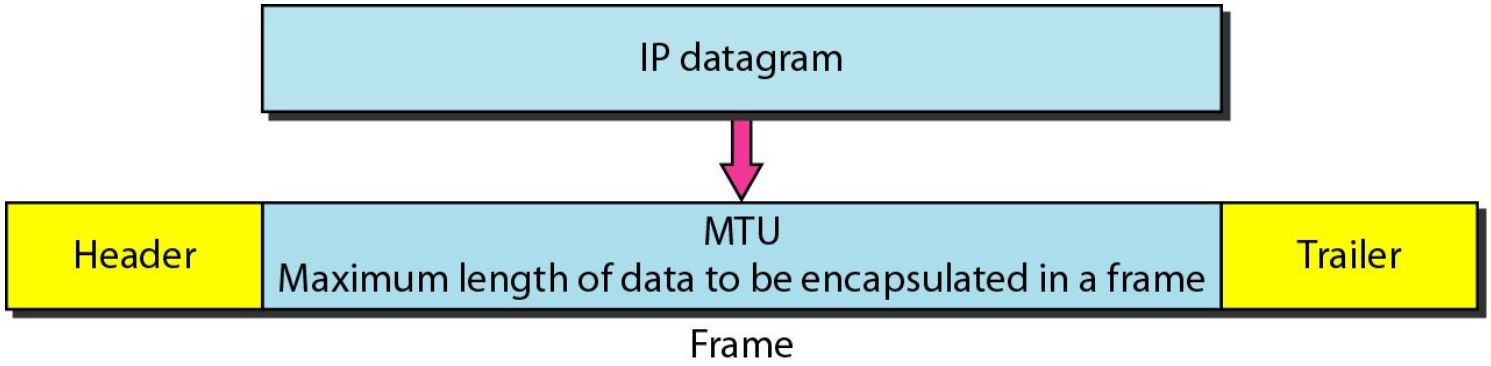


Table 20.5 *MTUs for some networks*

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Fields Related to Fragmentation

- **Identification:** identifies a datagram originating from the source host. A combination of the identification and source address must uniquely define a datagram as it leaves the source node.
- **Flags:** see next slide.
- **Fragmentation offset:** is the offset of the data in the original datagram measured in units of 8 bytes.

20-3 IPv6

The network layer protocol in the TCP/IP protocol suite is currently IPv4. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

Topics discussed in this section:

Advantages
Packet Format
Extension Headers

IPv6: Advantages

- Larger address space.
- Better header format.
- New options.
- Allowance for extensions.
- Support for resource allocation.
- Support for more security.

Figure 20.15 *IPv6 datagram header and payload*

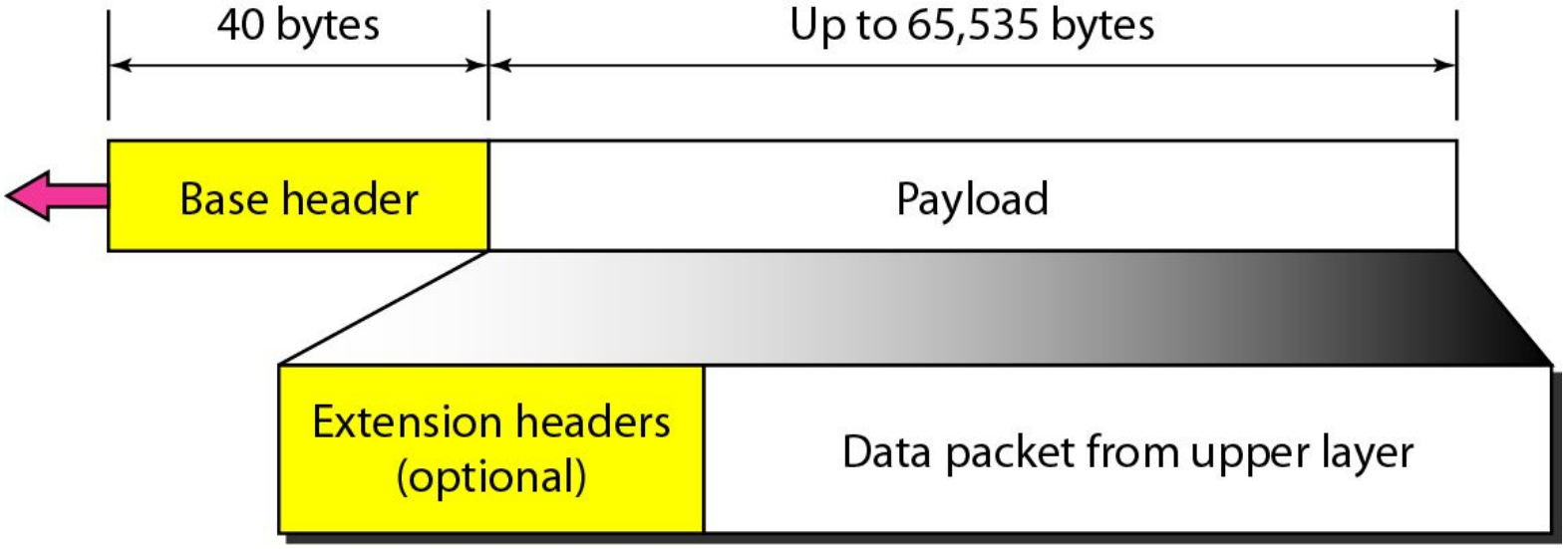


Figure 20.16 *Format of an IPv6 datagram*

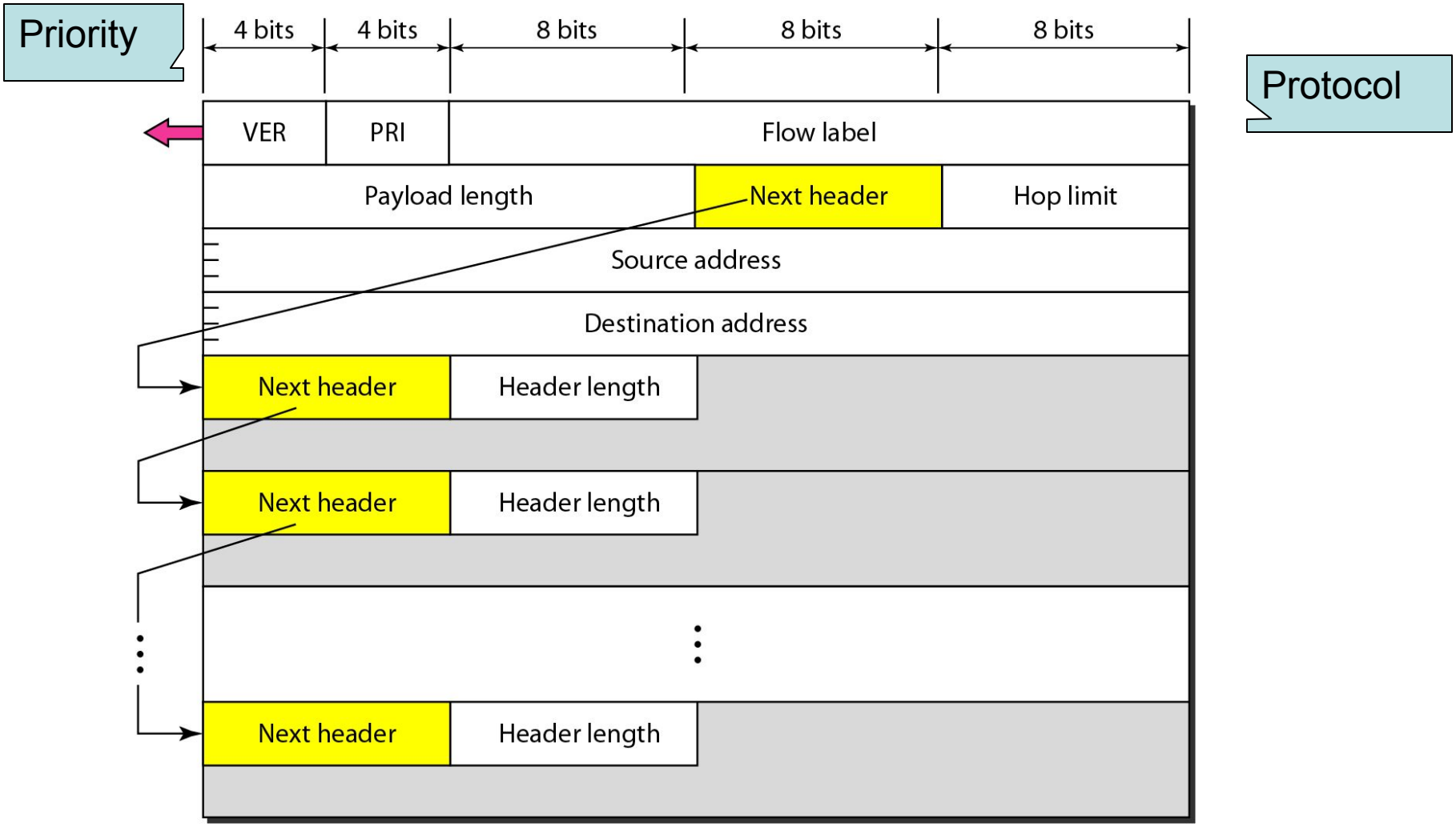
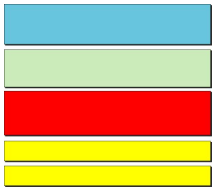


Table 20.9 *Comparison between IPv4 and IPv6 packet headers*

Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.



Chapter 14

Unicast Routing Protocols: RIP, OSPF, and BGP

Objectives

Upon completion you will be able to:

- *Distinguish between intra and interdomain routing*
- *Understand distance vector routing and RIP*
- *Understand link state routing and OSPF*
- *Understand path vector routing and BGP*

14.1 INTRA- AND INTERDOMAIN ROUTING

Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing.

Figure 14.1 *Autonomous systems*

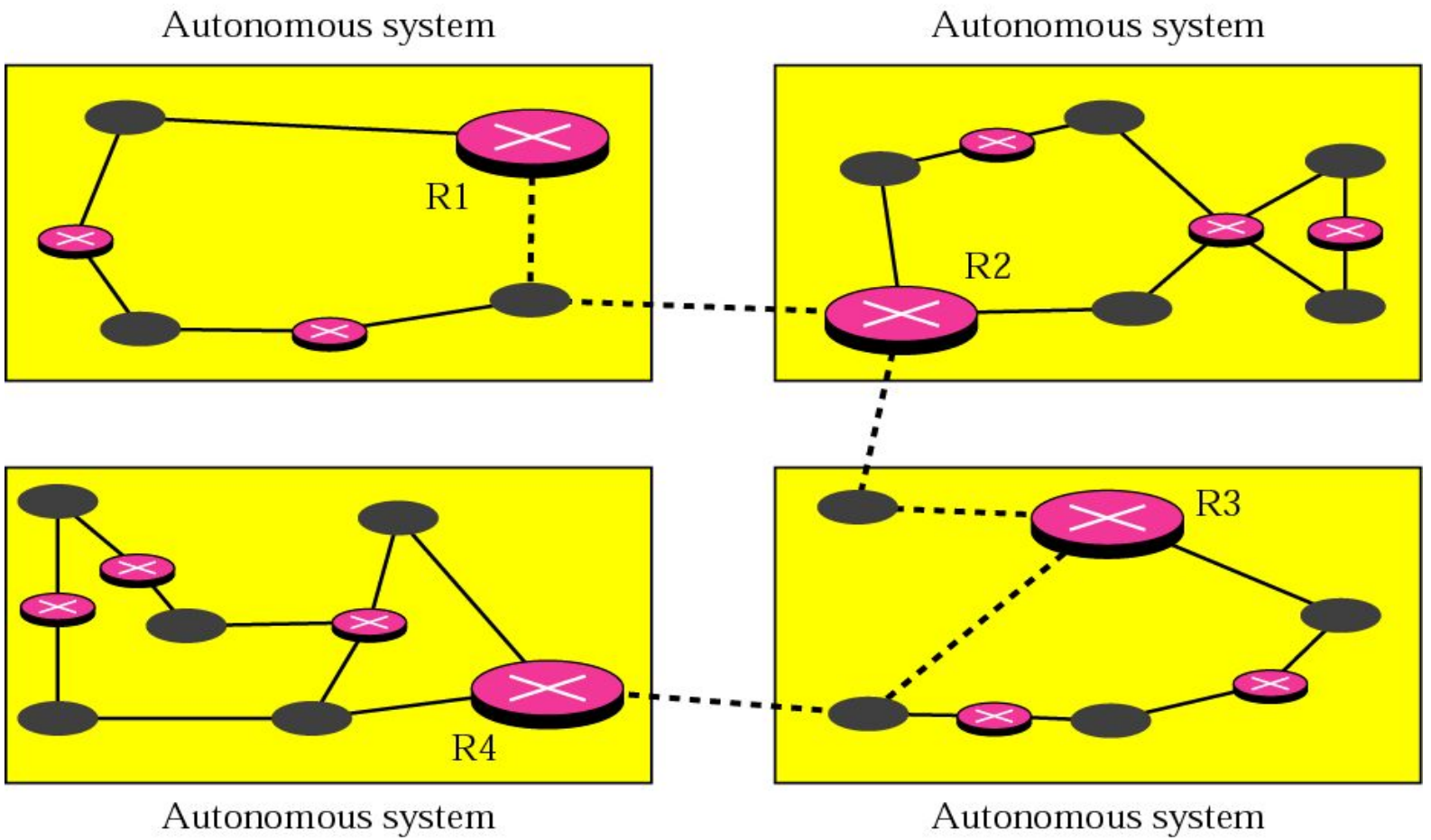
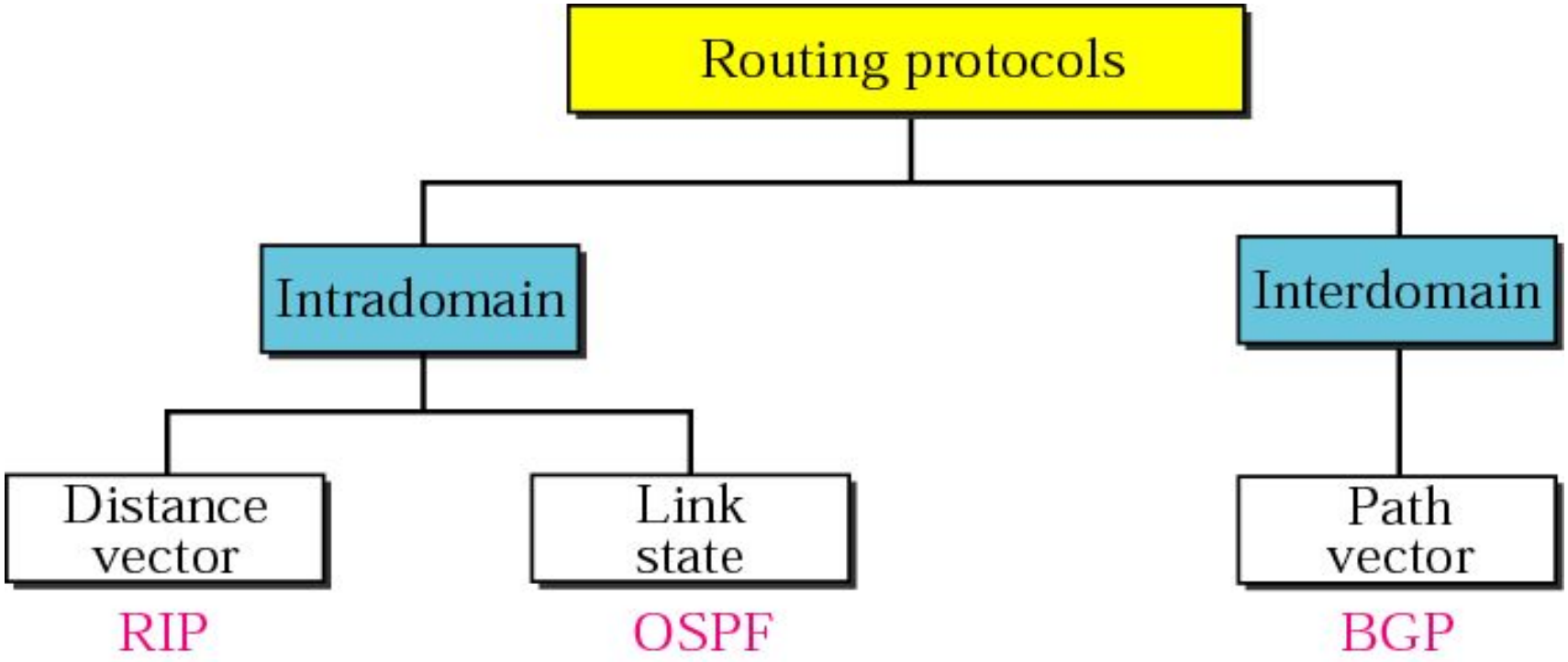


Figure 14.2 *Popular routing protocols*



14.2 DISTANCE VECTOR ROUTING

In distance vector routing, the least cost route between any two nodes is the route with minimum distance. In this protocol each node maintains a vector (table) of minimum distances to every node

The topics discussed in this section include:

Initialization
Sharing
Updating
When to Share
Two-Node Loop Instability
Three-Node Instability

Figure 14.3 *Distance vector routing tables*

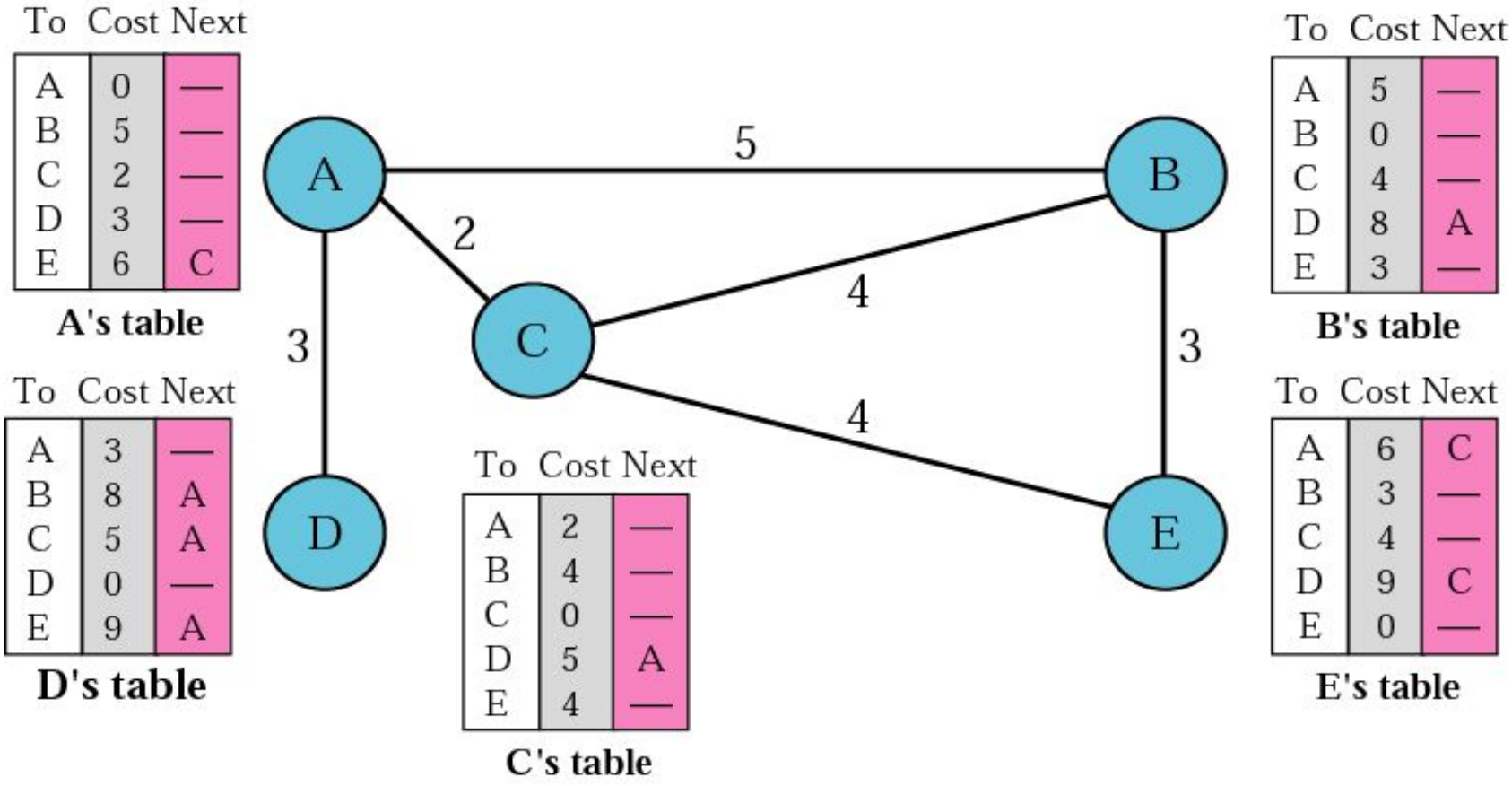
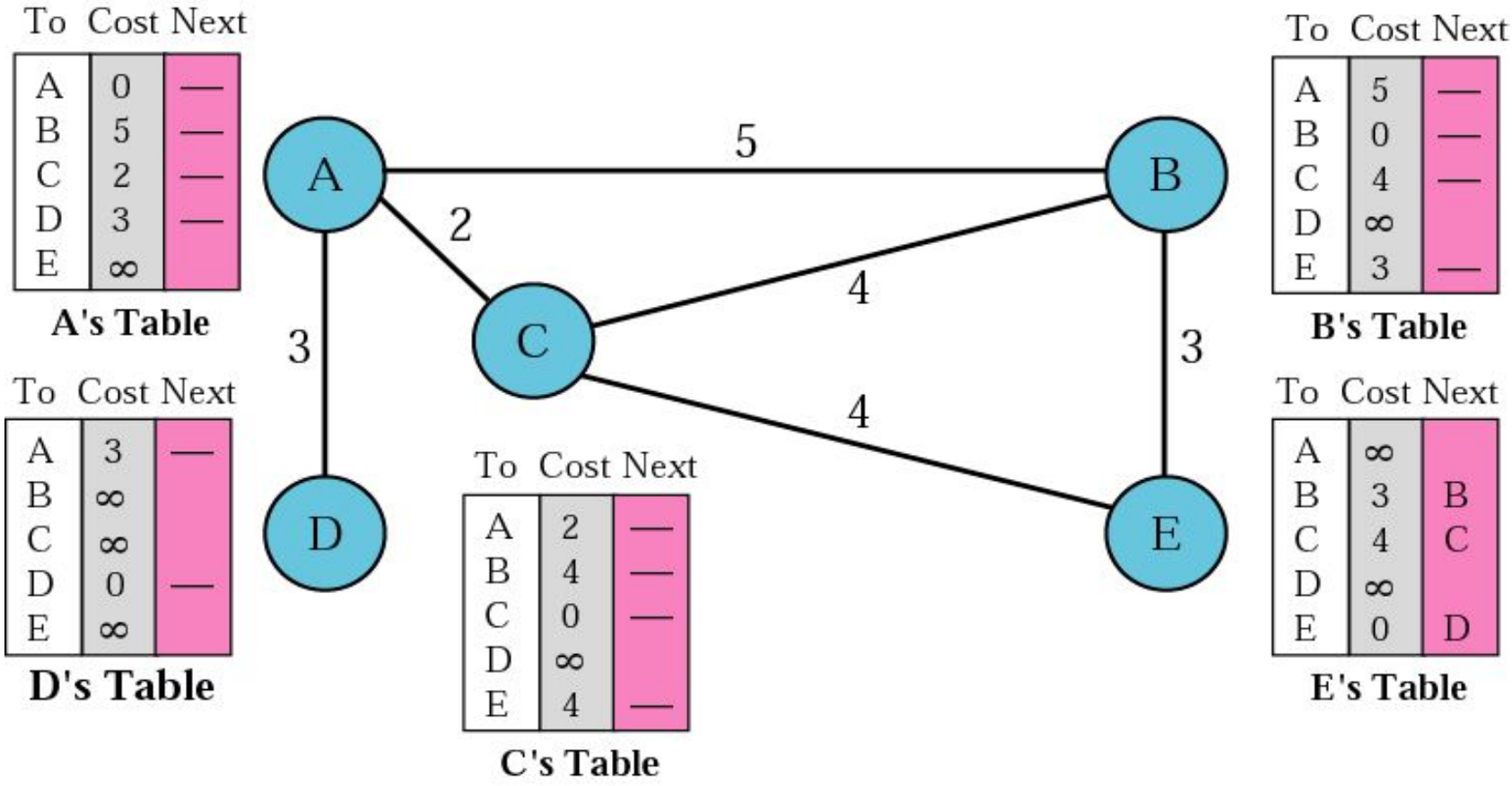
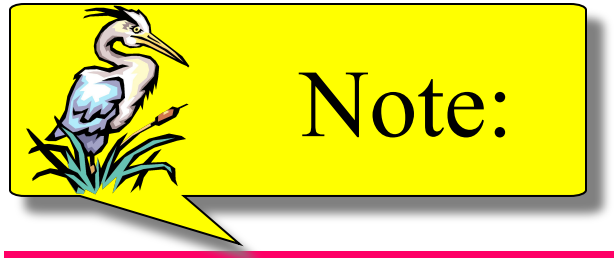
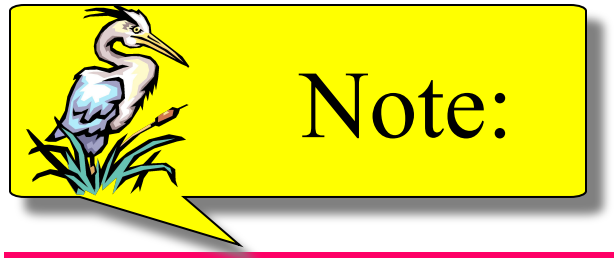


Figure 14.4 *Initialization of tables in distance vector routing*





In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.



RIP uses the services of UDP on well-known port 520.

14.4 LINK STATE ROUTING

In link state routing, if each node in the domain has the entire topology of the domain, the node can use Dijkstra's algorithm to build a routing table.

The topics discussed in this section include:

Building Routing Tables

Figure 14.15 *Concept of link state routing*

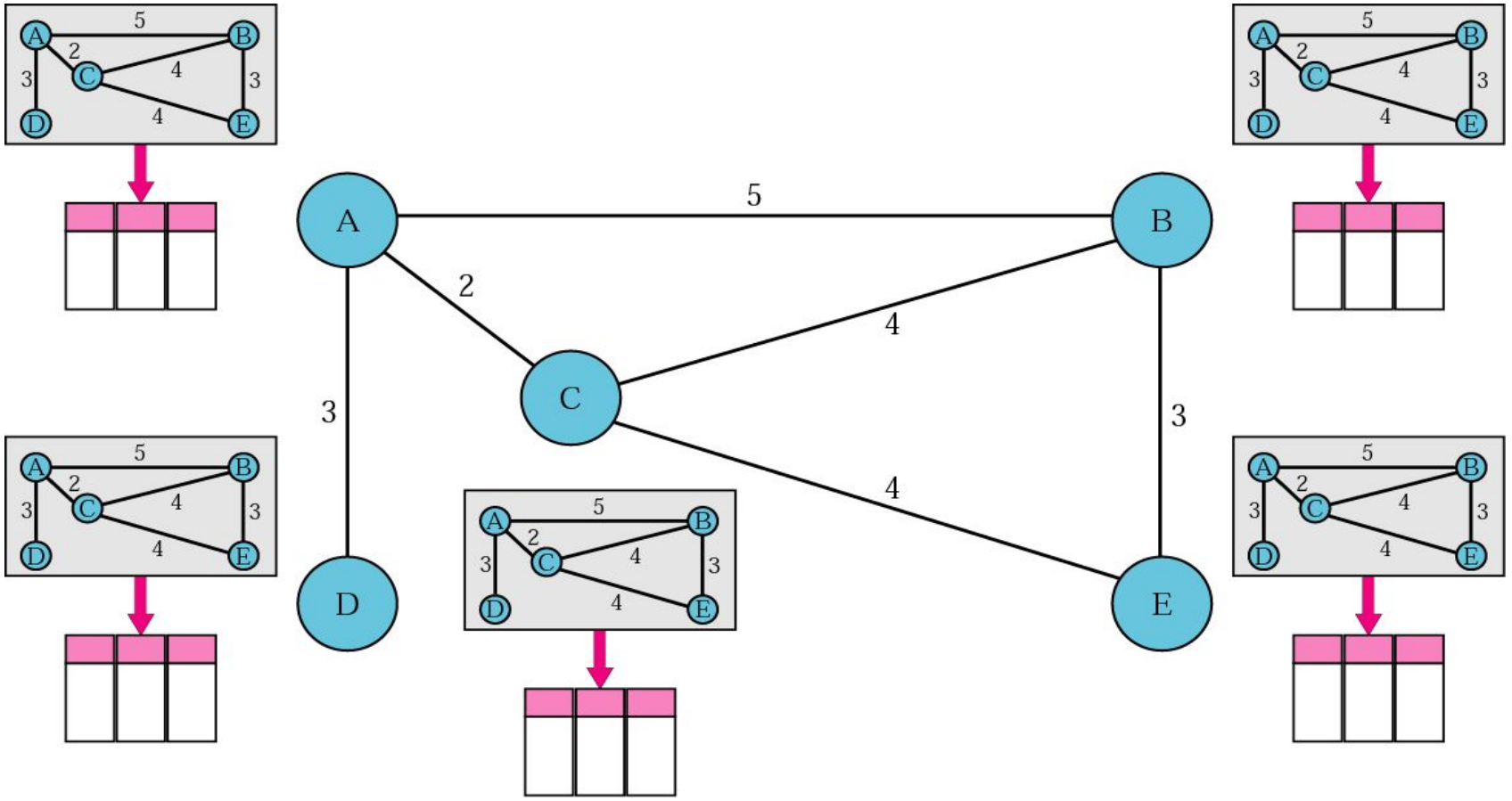


Figure 14.16 *Link state knowledge*

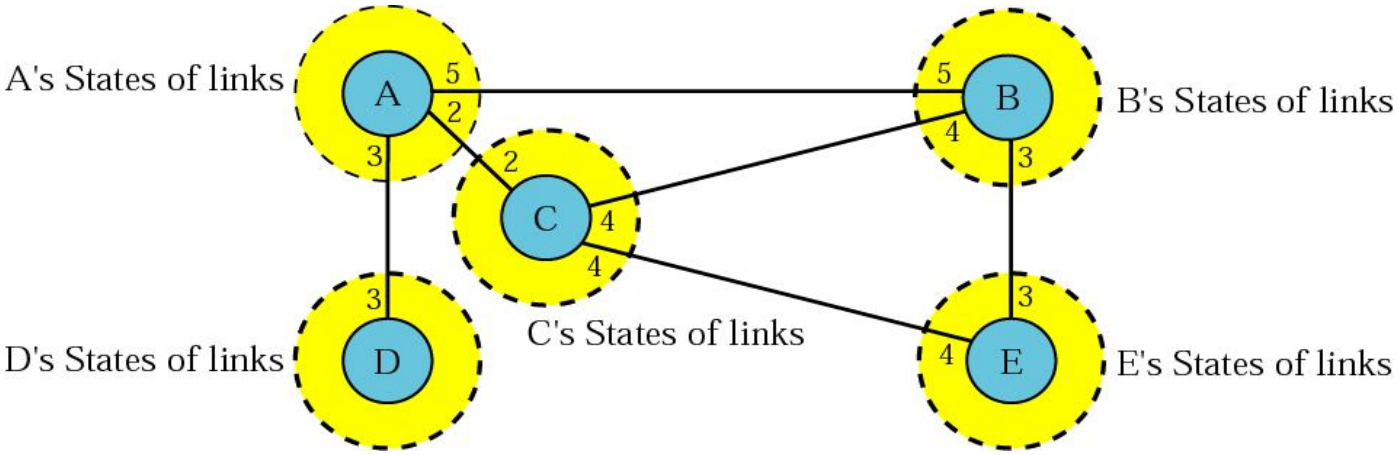


Figure 14.17 *Dijkstra algorithm*

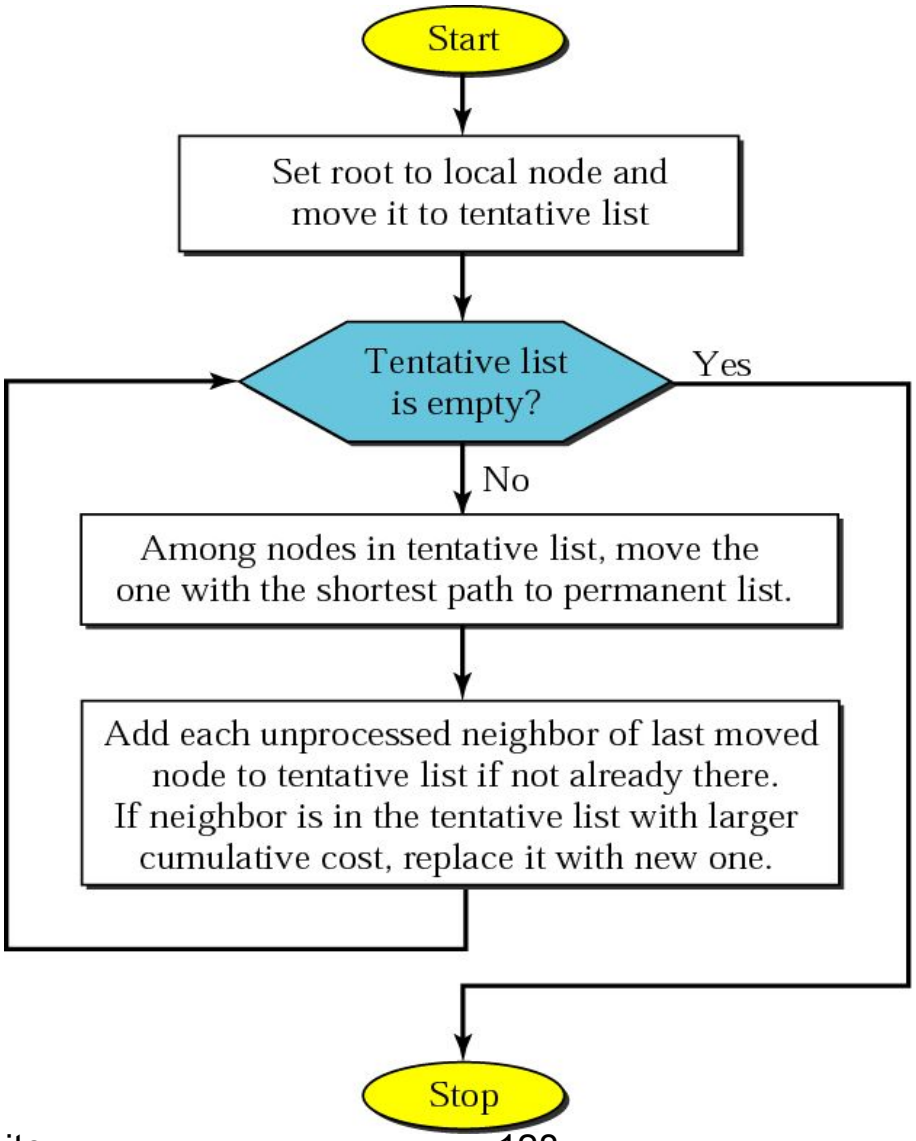
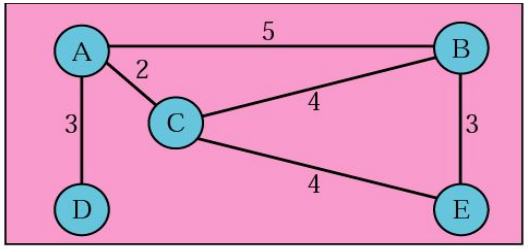
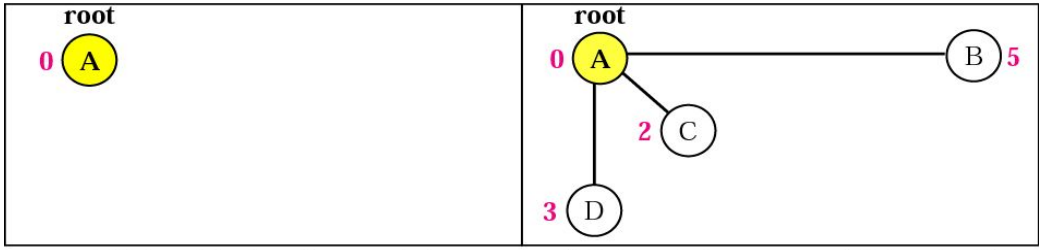


Figure 14.18 *Example of formation of shortest path tree*

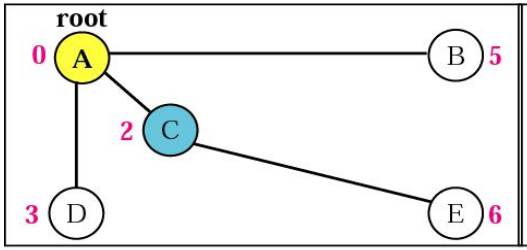


Topology

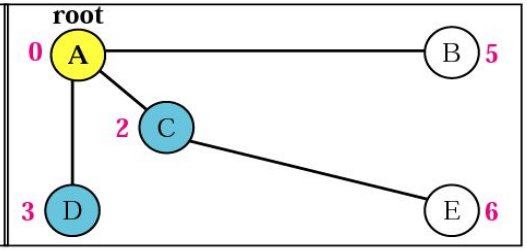


1. Set root to A and move A to tentative list

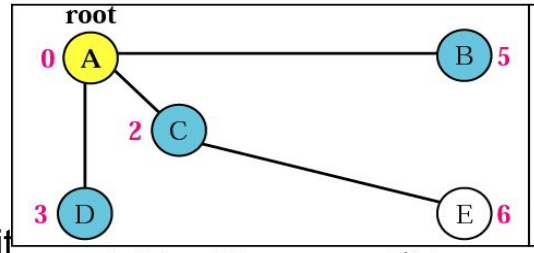
2. Move A to permanent list and add B, C, and D to tentative list



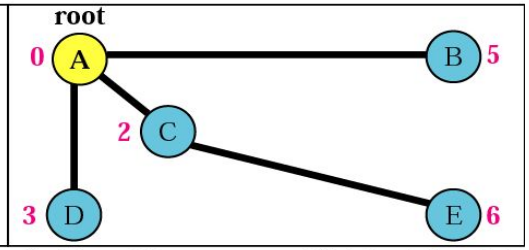
3. Move C to permanent and add E to tentative list



4. Move D to permanent list.



5. Move B to permanent list



6. Move E to permanent list (tentative list is empty)

Table 14.1 Routing table for node A

Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

14.5 OSPF

The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

The topics discussed in this section include:

Areas
Metric
Types of Links
Graphical Representation
OSPF Packets
Link State Update Packet
Other Packets
Encapsulation

Figure 14.19 *Areas in an autonomous system*

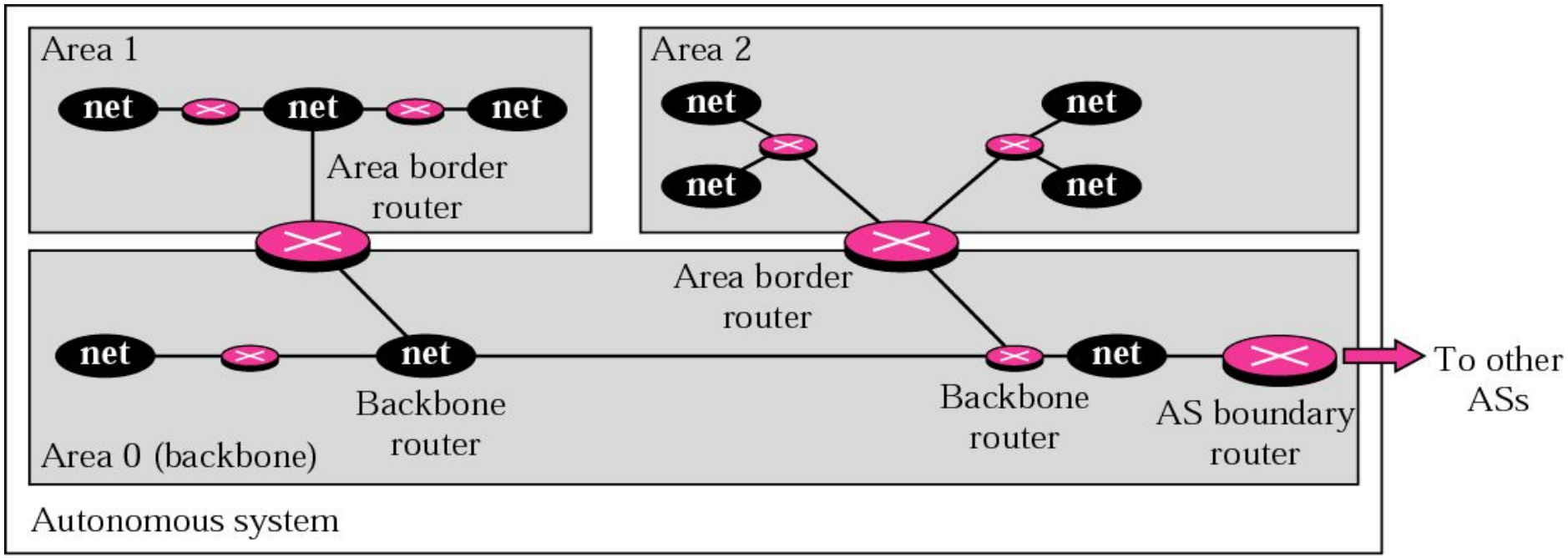


Figure 14.20 *Types of links*

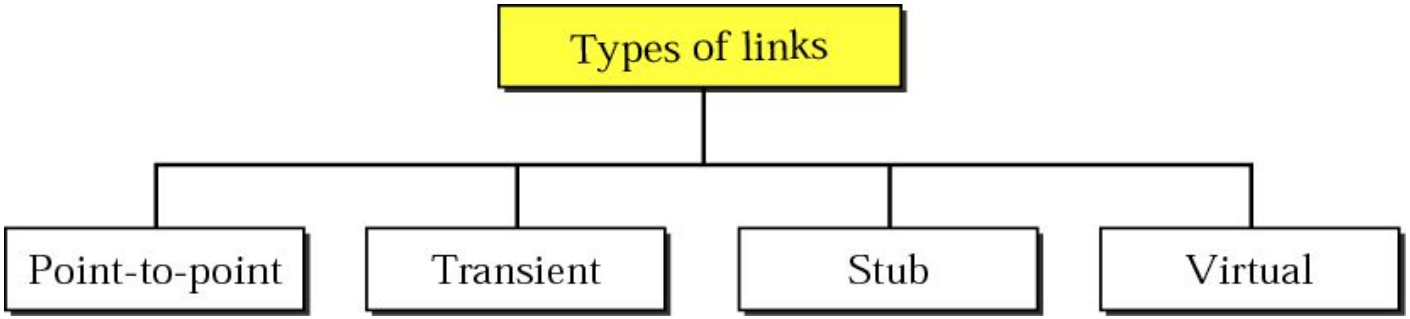


Figure 14.21 *Point-to-point link*

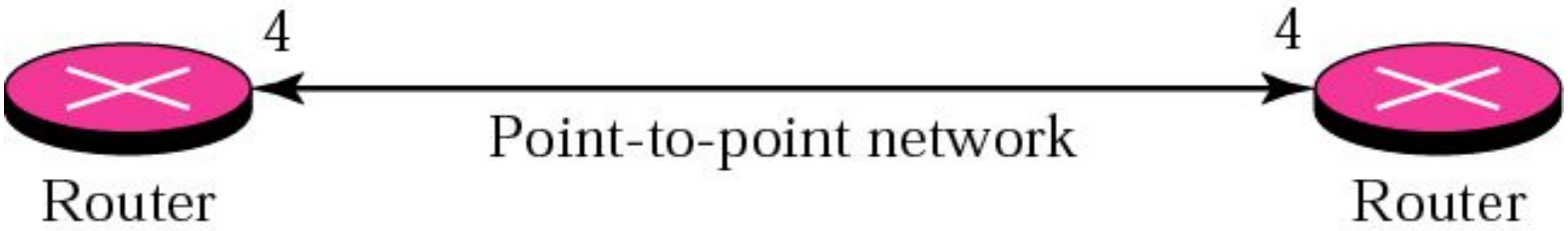
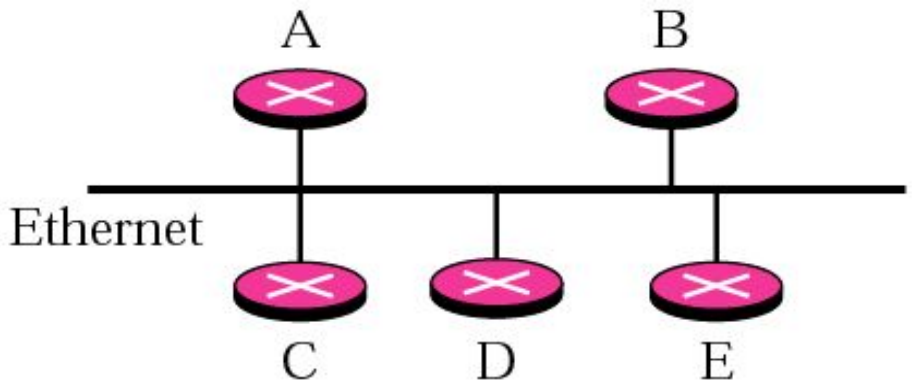
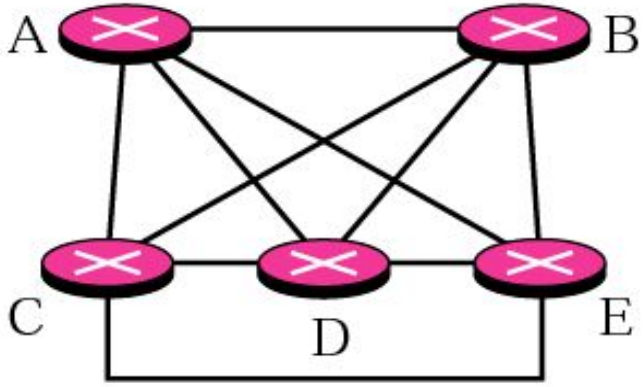


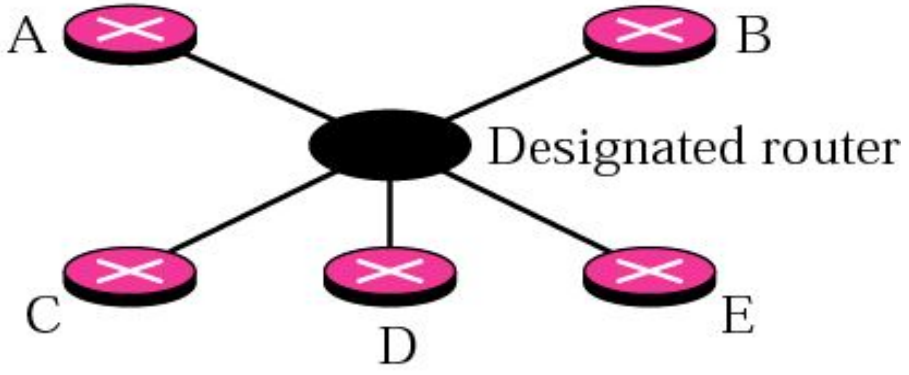
Figure 14.22 *Transient link*



a. Transient network

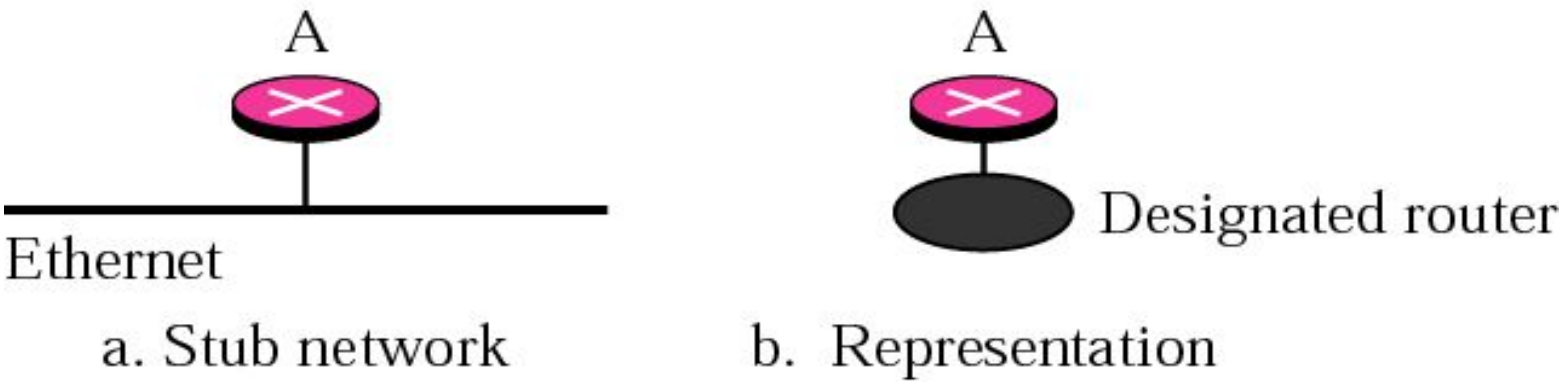


b. Unrealistic representation



c. Realistic representation

Figure 14.23 *Stub link*



a. Stub network b. Representation

Figure 14.25 *Types of OSPF packets*

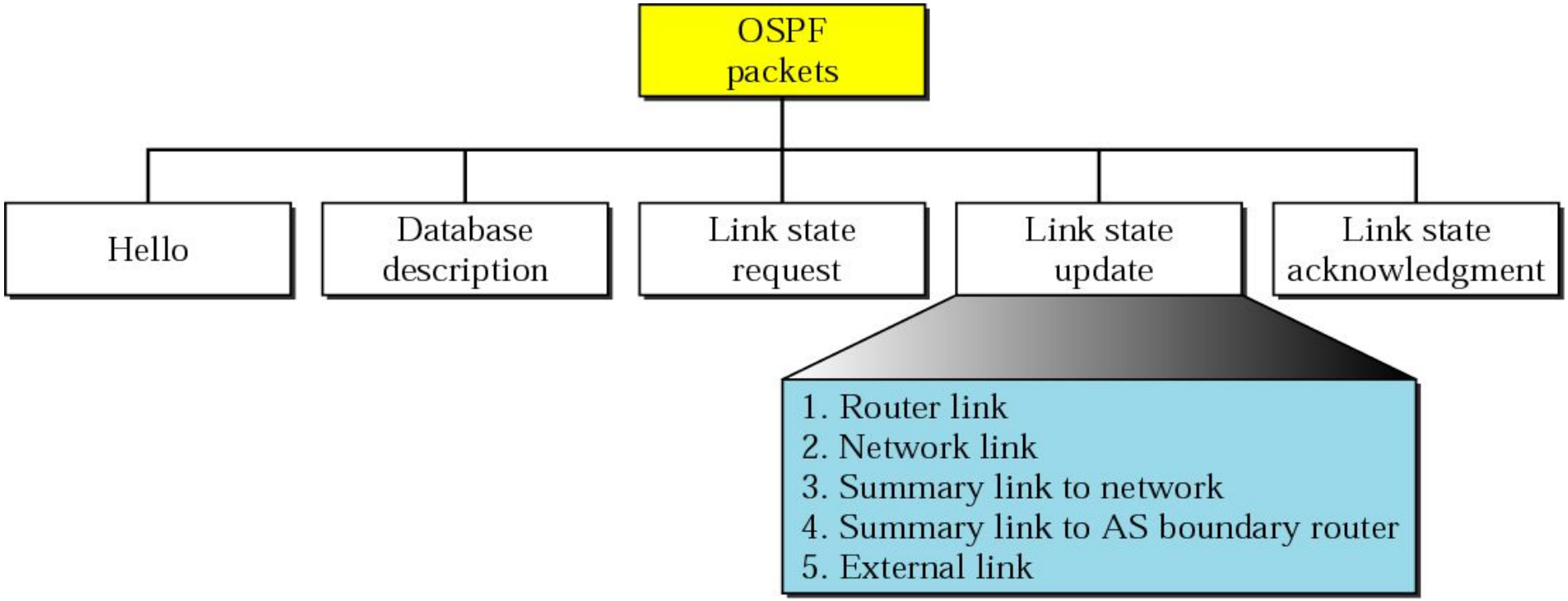
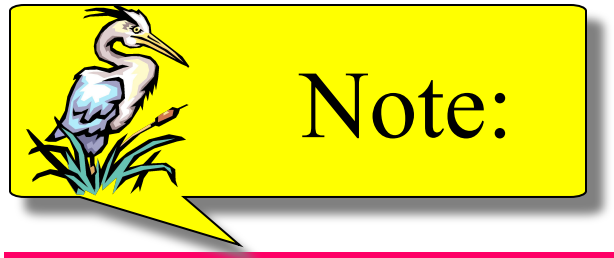


Table 14.2 Link types, link identification, and link data

Link Type	*Link Identification*	*Link Data*
Type 1: Point-to-point	Address of neighbor router	Interface number
Type 2: Transient	Address of designated router	Router address
Type 3: Stub	Network address	Network mask
Type 4: Virtual	Address of neighbor router	Router address



OSPF packets are encapsulated in IP datagrams.

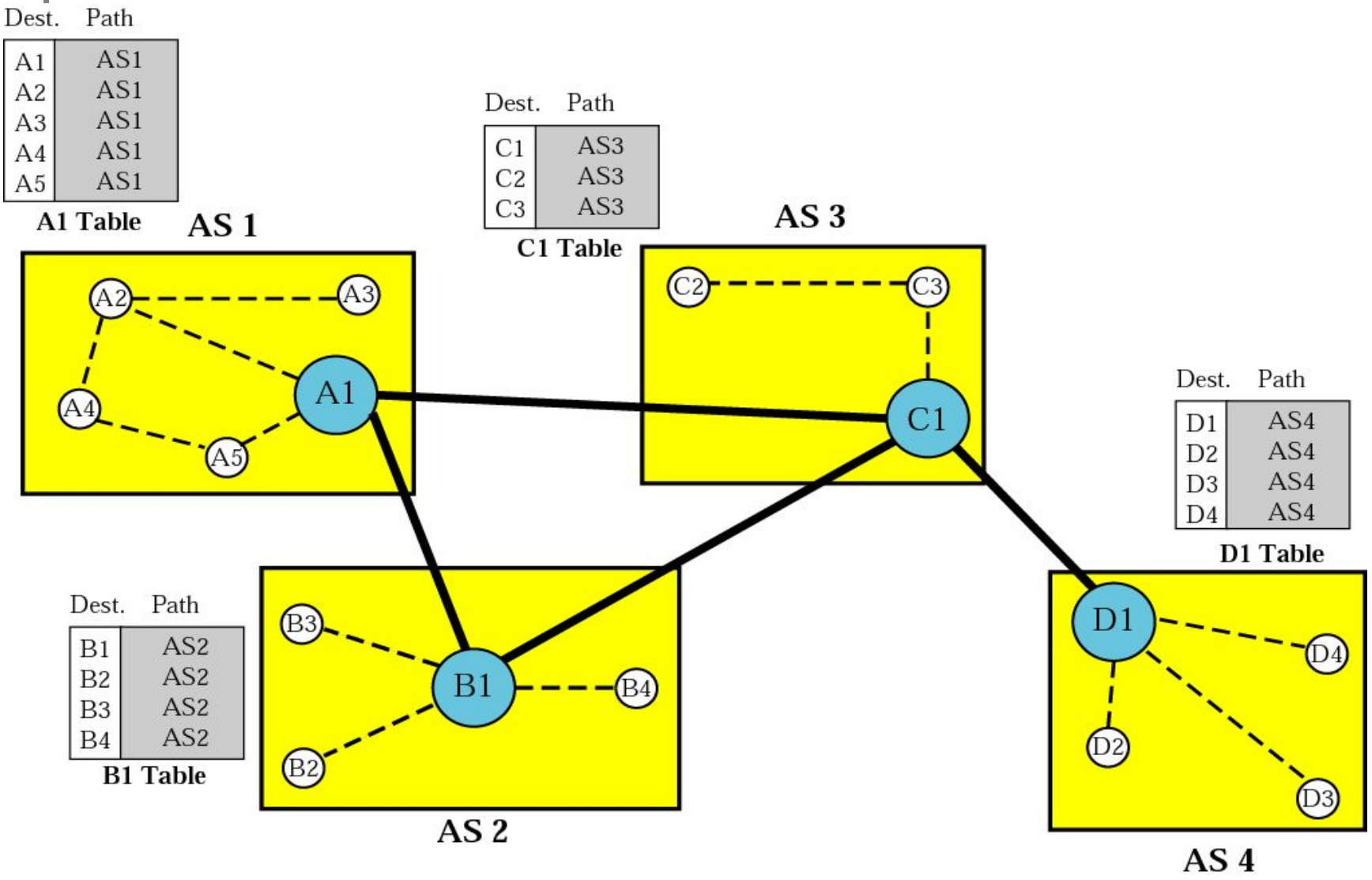
14.6 PATH VECTOR ROUTING

Path vector routing is similar to distance vector routing. There is at least one node, called the speaker node, in each AS that creates a routing table and advertises it to speaker nodes in the neighboring ASs..

The topics discussed in this section include:

Initialization
Sharing
Updating

Figure 14.48 *Initial routing tables in path vector routing*



14.7 BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.

The topics discussed in this section include:

Types of Autonomous Systems
Path Attributes
BGP Sessions
External and Internal BGP
Types of Packets
Packet Format
Encapsulation

Figure 14.50 *Internal and external BGP sessions*

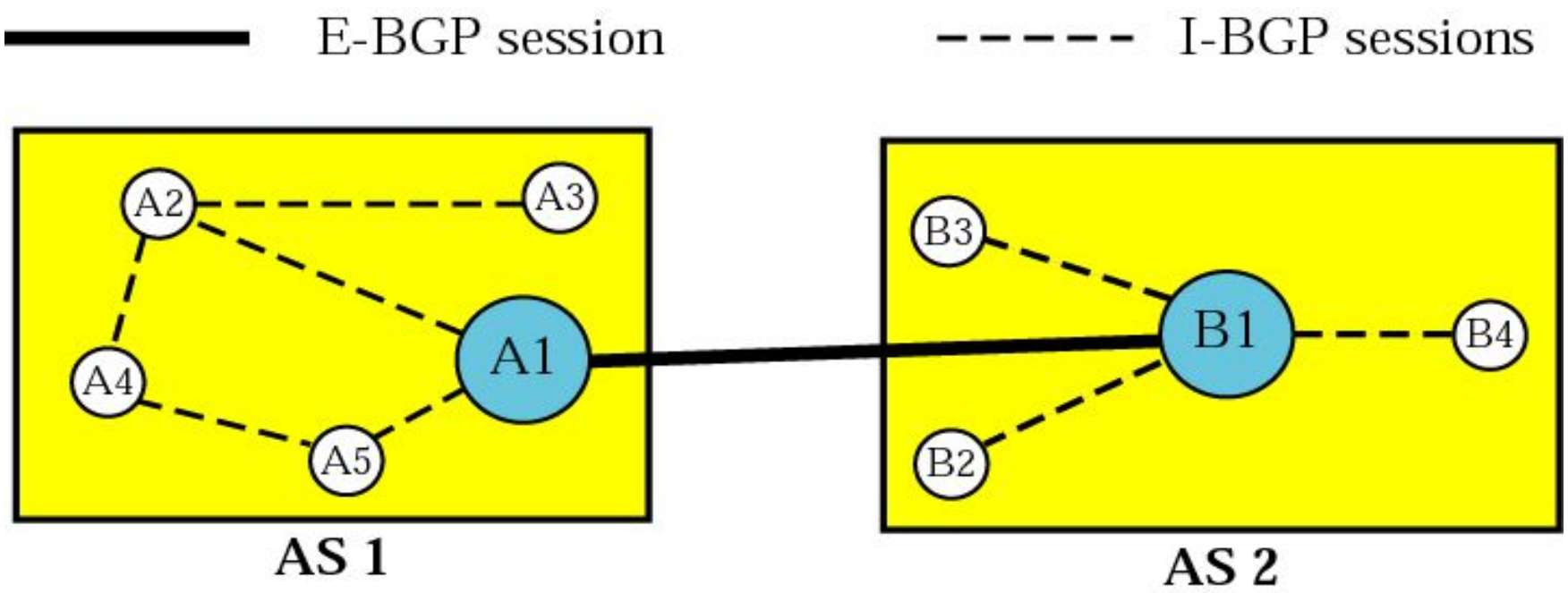
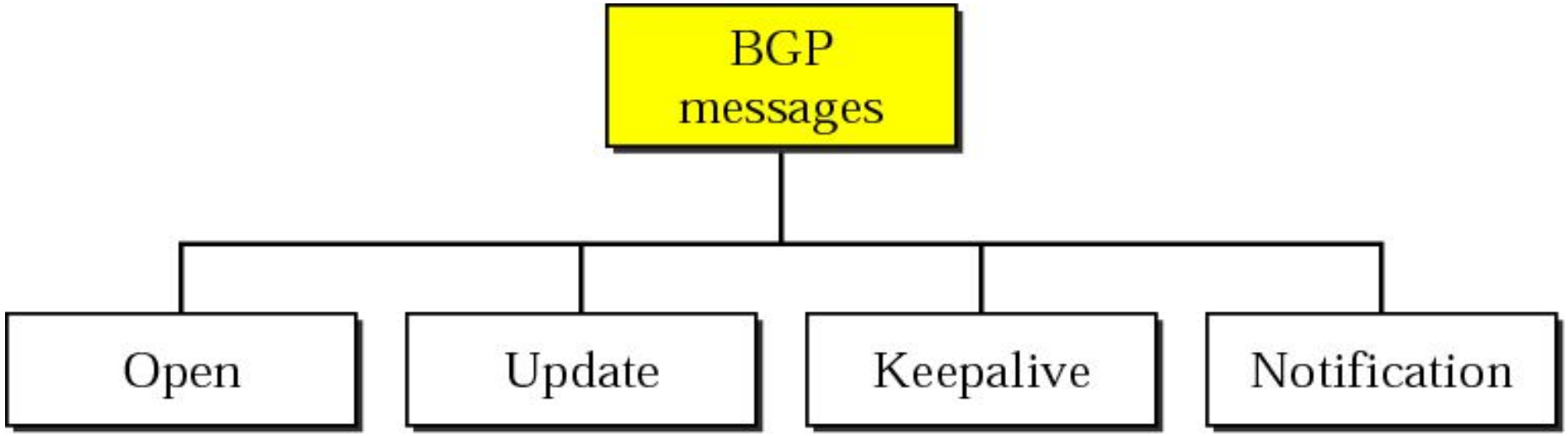


Figure 14.51 *Types of BGP messages*



BGP supports classless addressing and CIDR.

BGP uses the services of TCP on port 179.