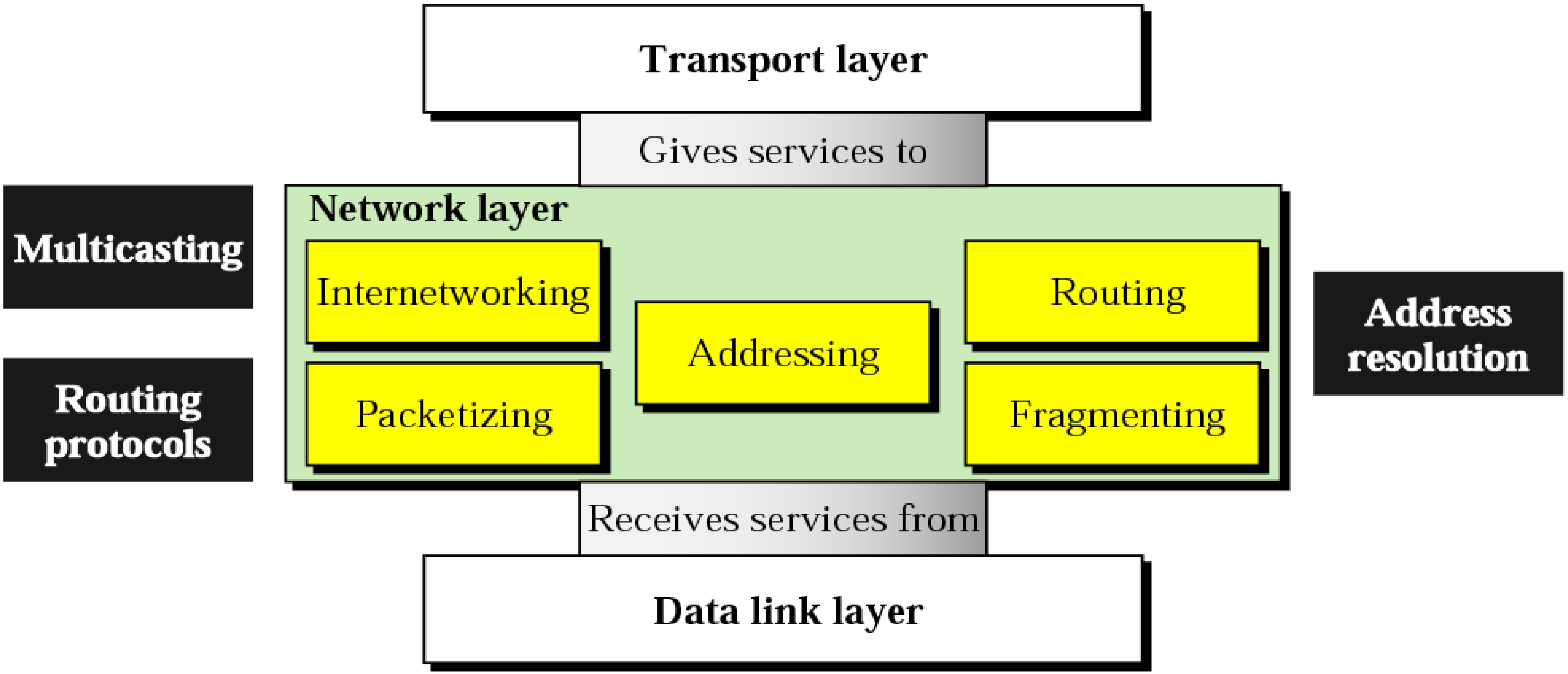
— Network Layer —

- **Text Book**

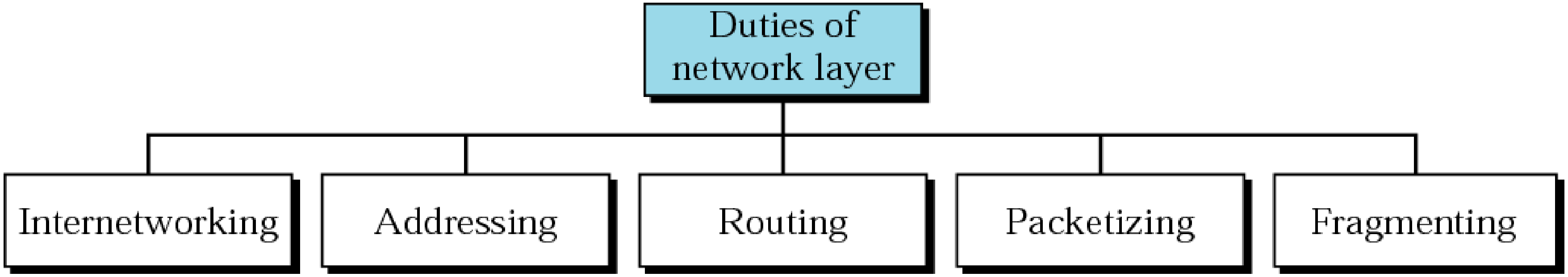
  Behrouz .A. Forouzan, “Data communication and Networking”, Tata McGrawHill, 2004

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

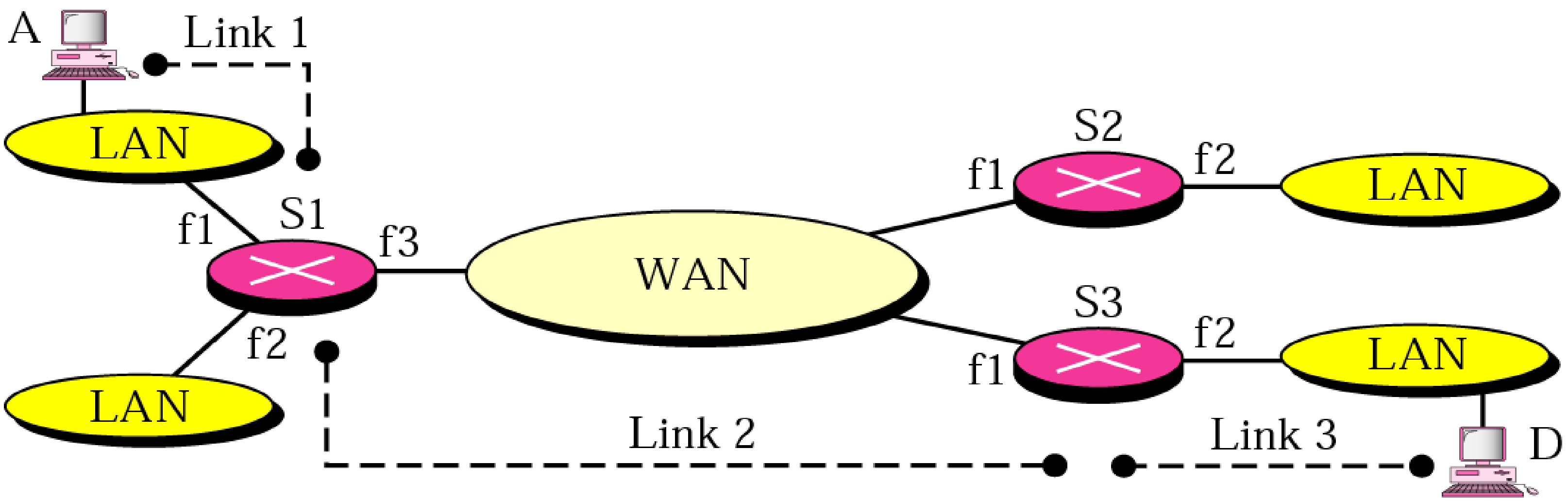


Figure 19.2 Links in an internetwork

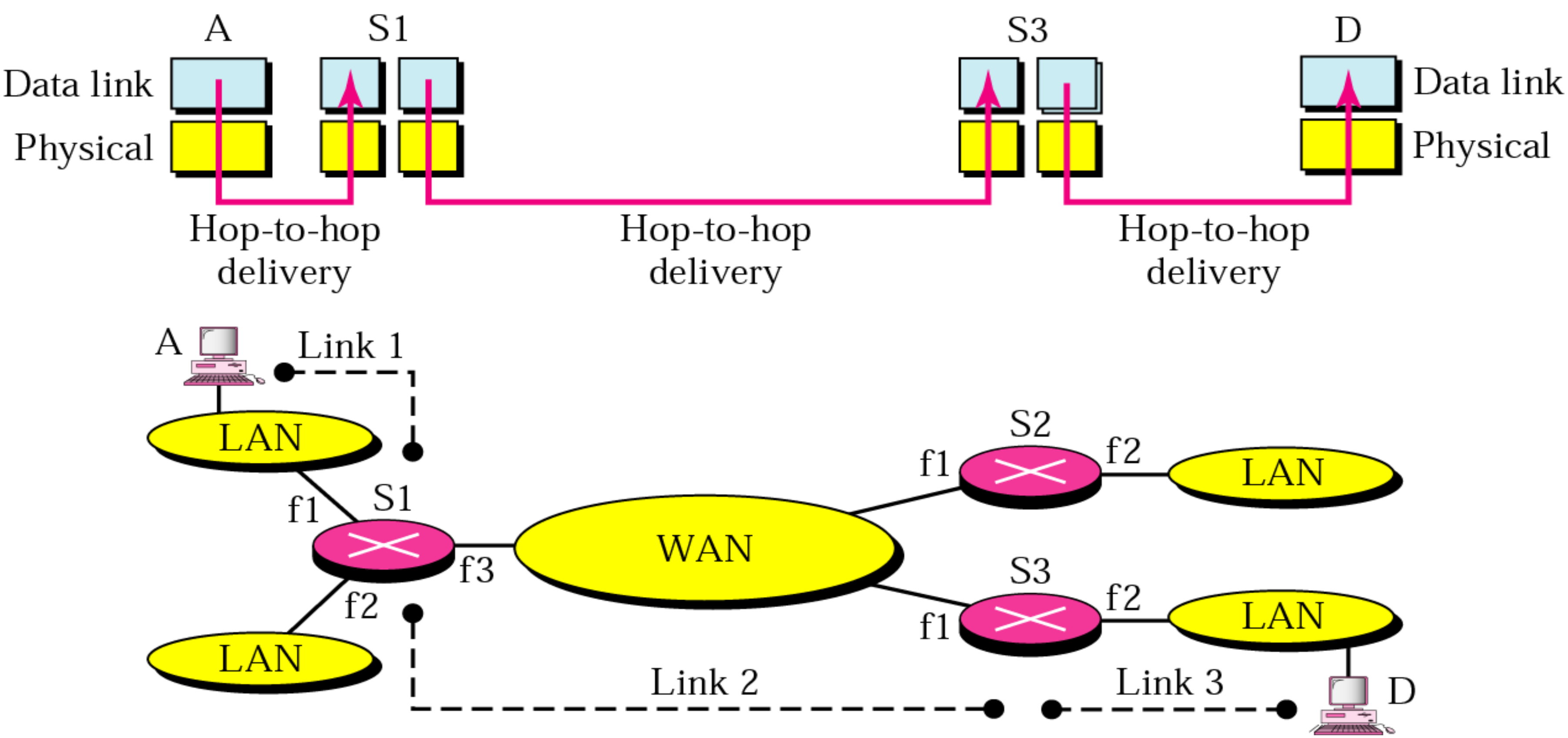


Figure 19.3 Network layer in an internetwork

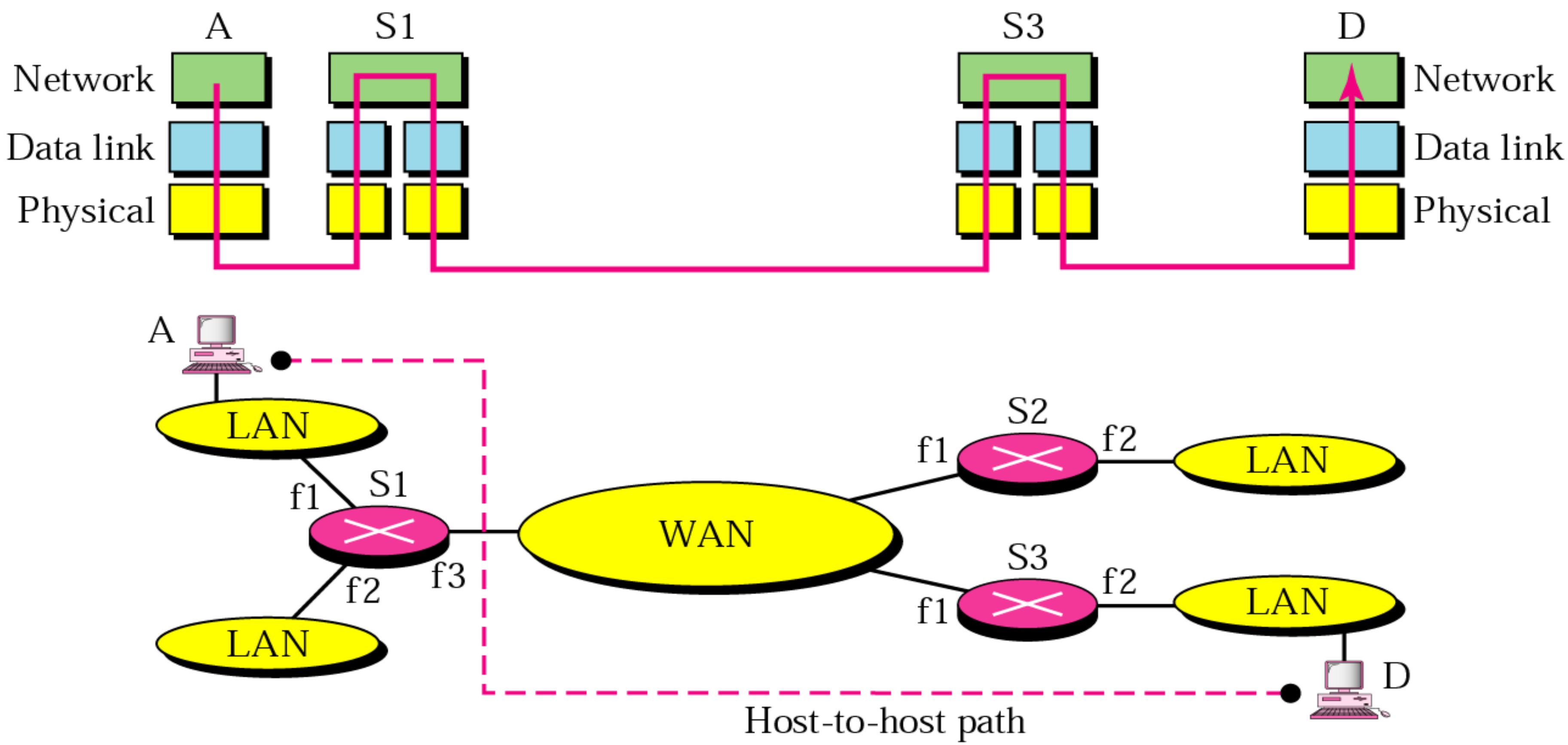


Figure 19.4 Network layer at the source

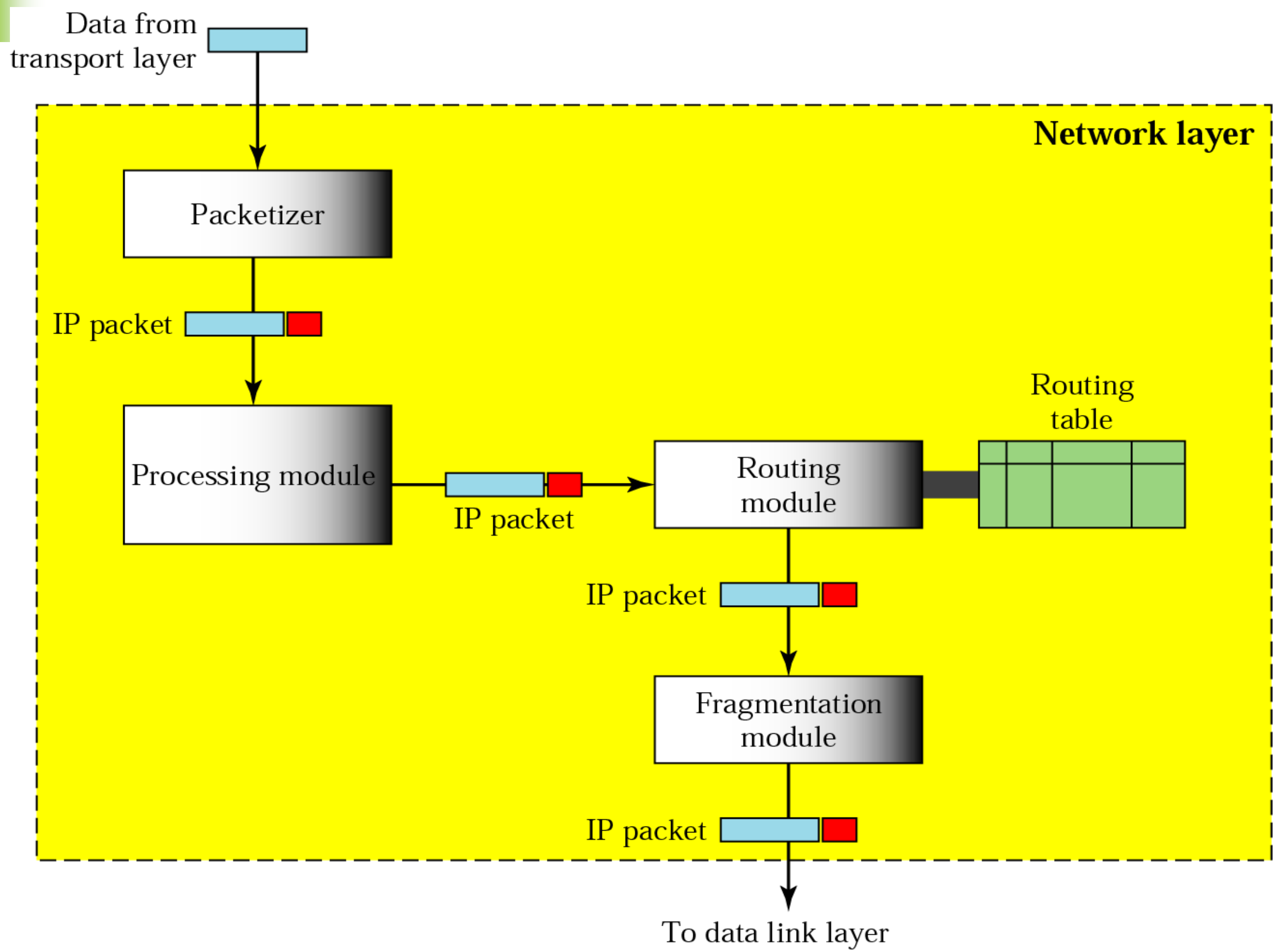


Figure 19.5 Network layer at a router

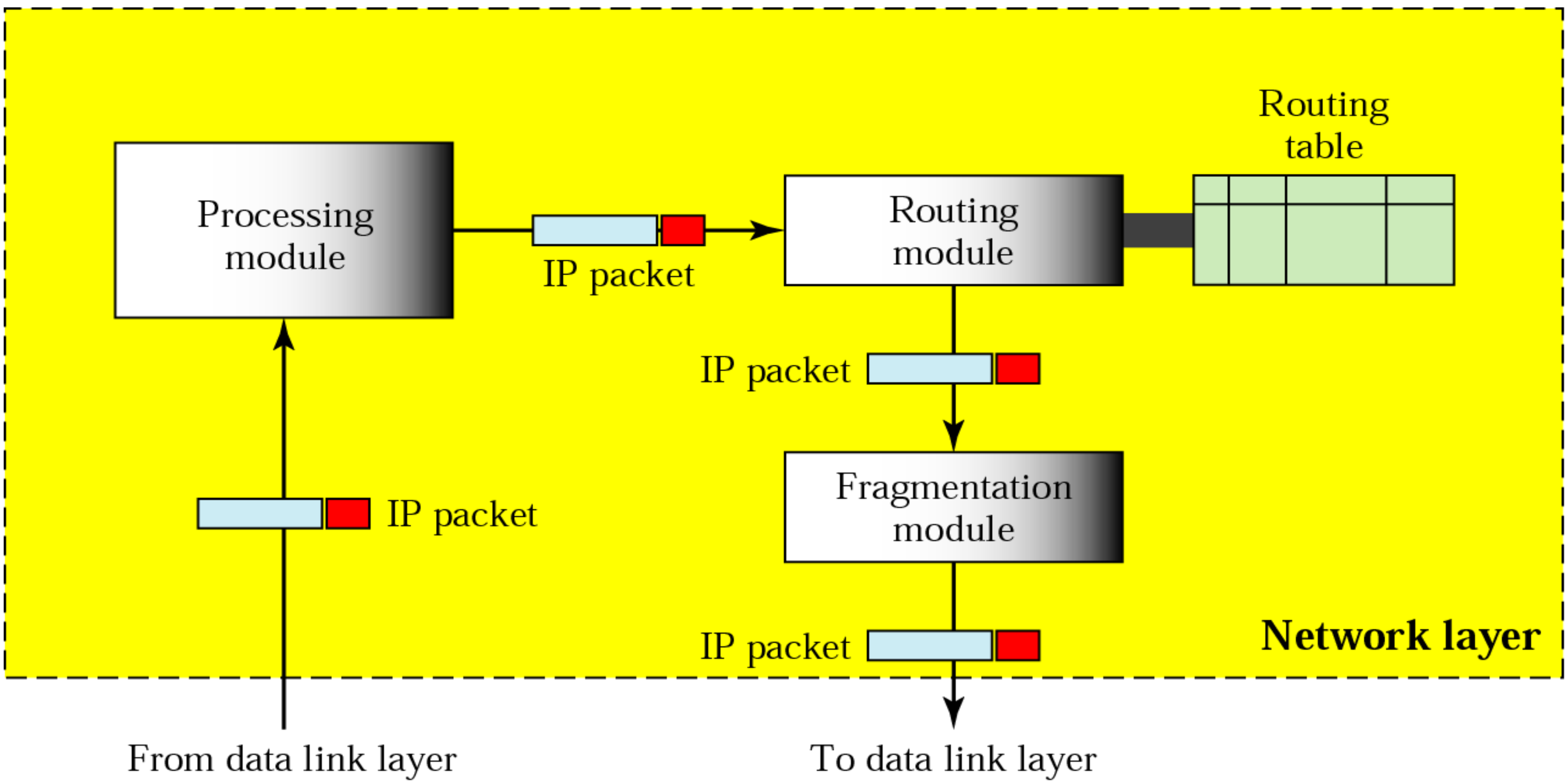


Figure 19.6 Network layer at the destination

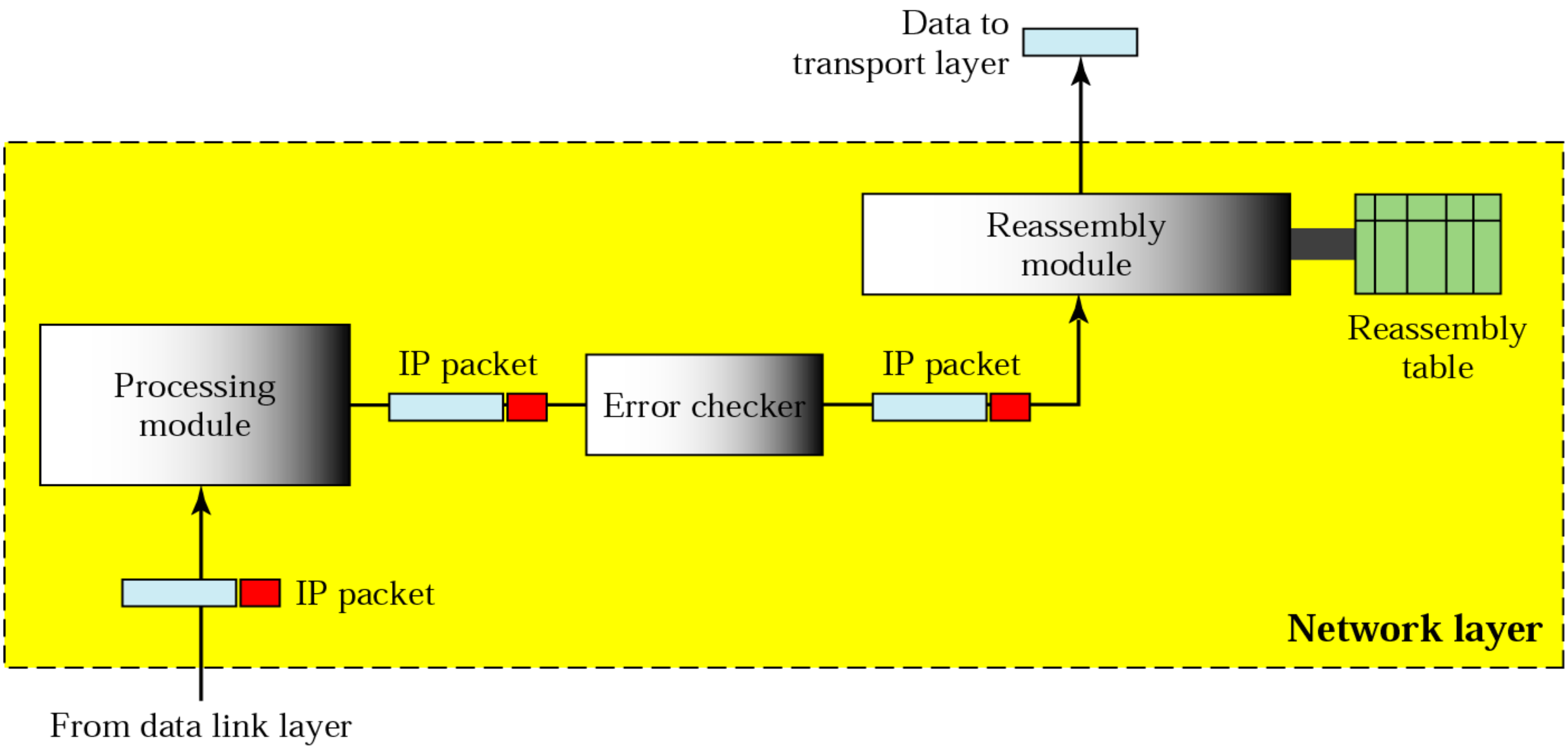
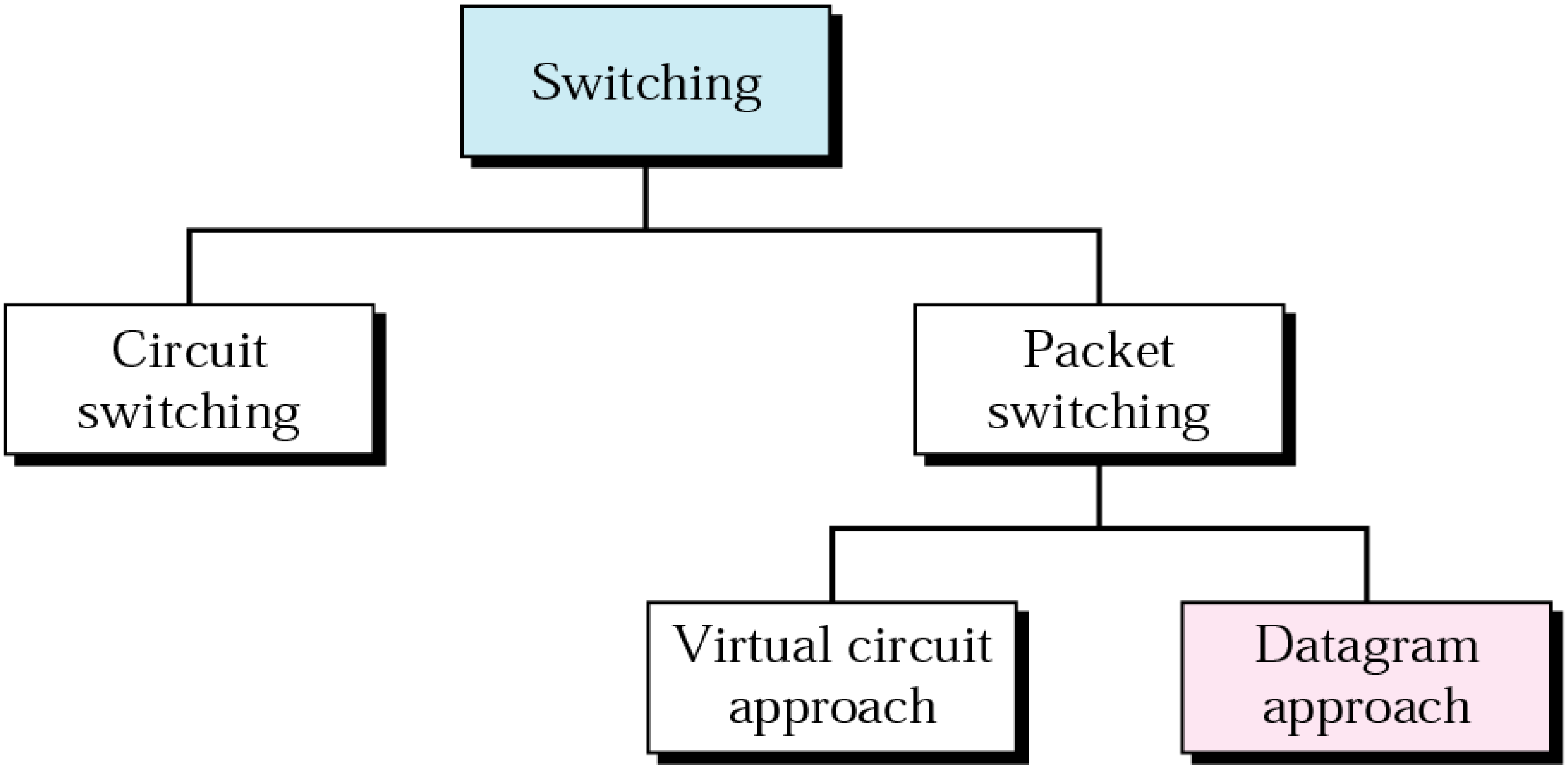


Figure 19.7 Switching



Connection, connection-less service

- *datagram* network provides network-layer *connectionless* service
- *virtual-circuit* network provides network-layer *connection* service
- analogous to TCP/UDP connecton-oriented / connectionless transport-layer services, but:
  - *service:* host-to-host
  - *no choice:* network provides one or the other
  - *implementation:* in network core

Virtual circuits

"source-to-dest path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

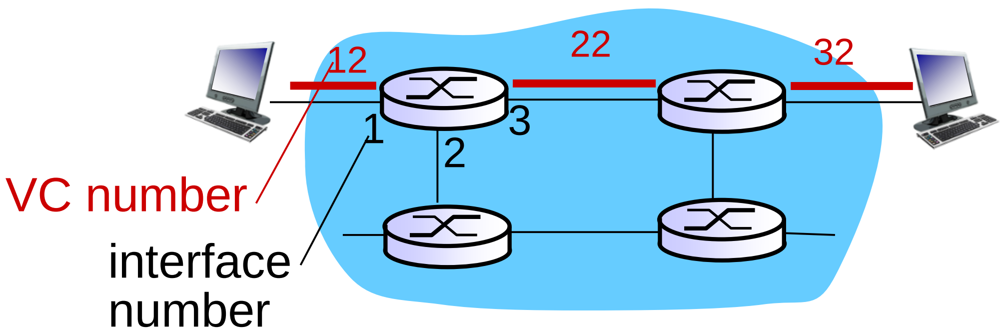
VC implementation

a VC consists of:

1. *path* from source to destination
2. *VC numbers*, one number for each link along path
3. *entries in forwarding tables* in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - new VC number comes from forwarding table

VC forwarding table



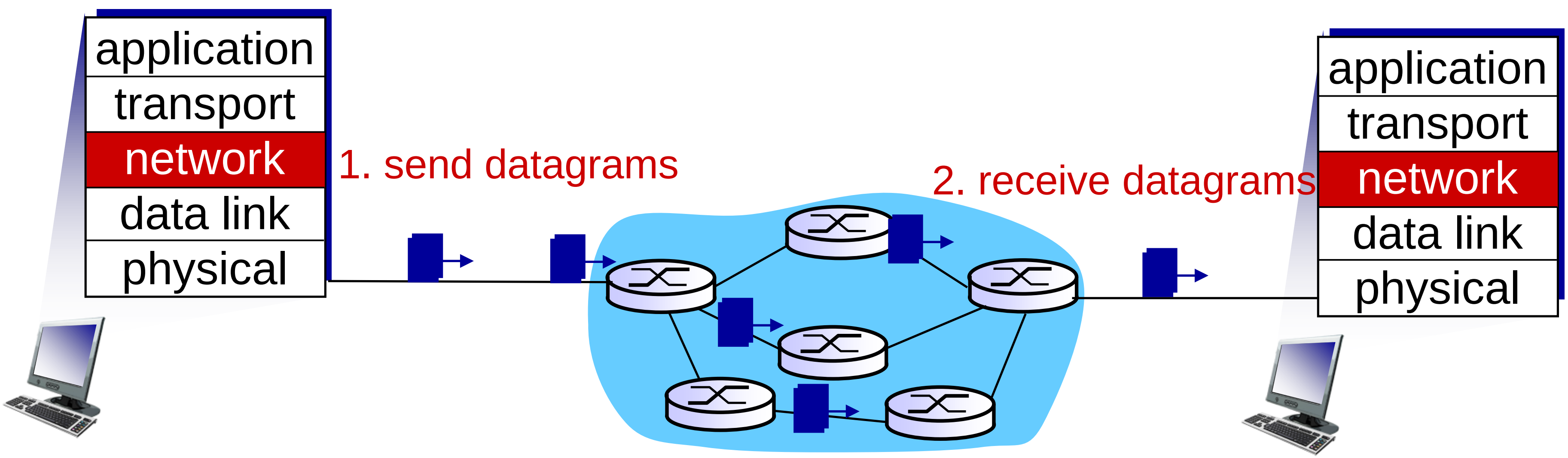
forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

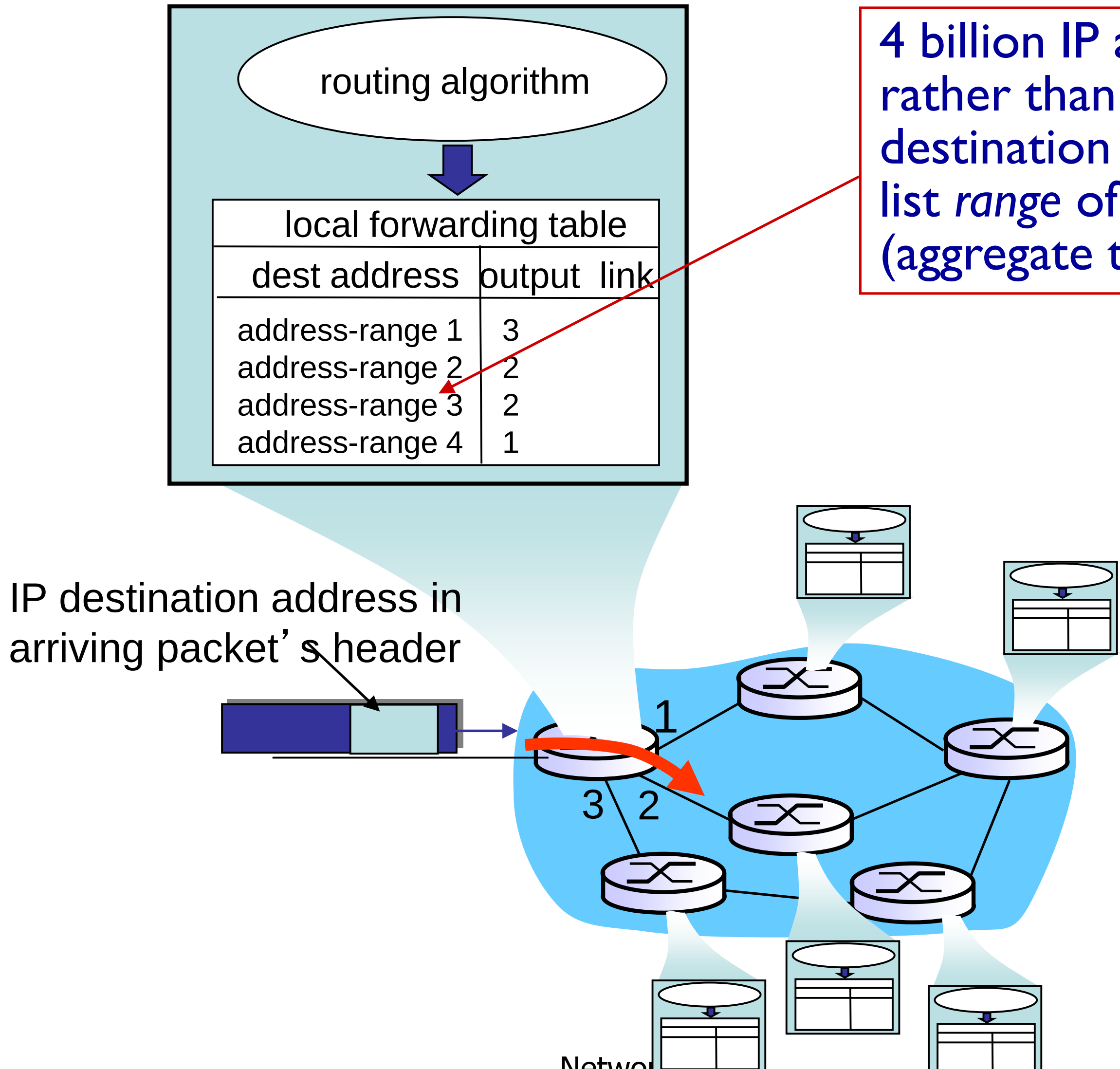
VC routers maintain connection state information!

Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address

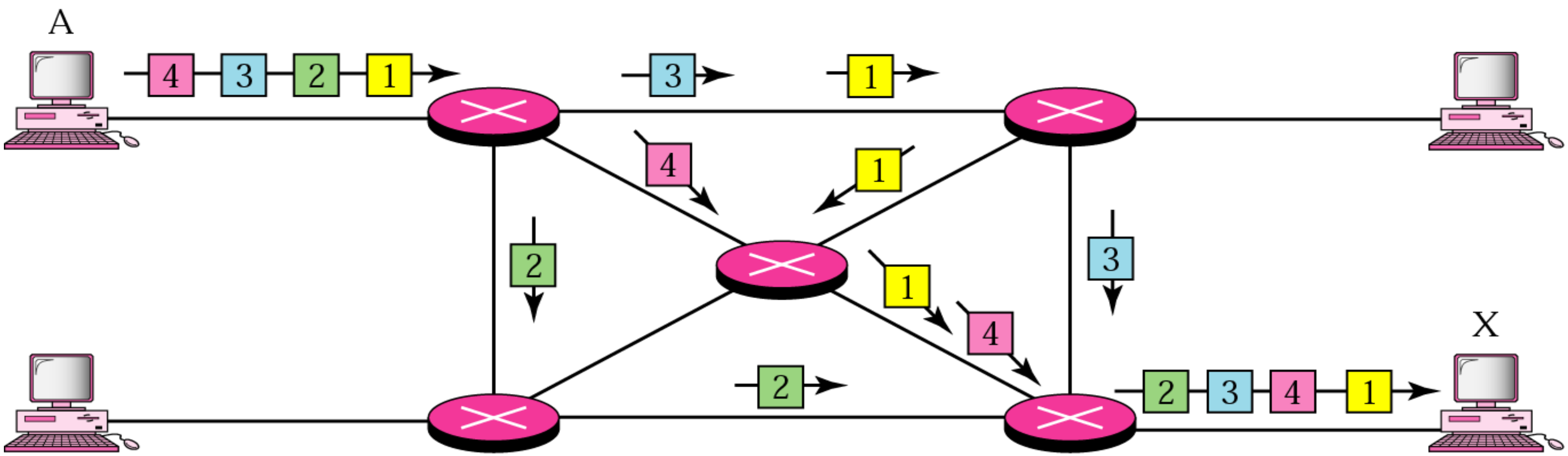


Datagram forwarding table



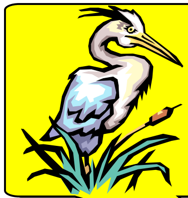
4 billion IP addresses, so rather than list individual destination address list *range* of addresses (aggregate table entries)

Figure 19.8 Datagram approach



Note:

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



Note:

Communication at the network layer in the Internet is connectionless.

19.2 Addressing

Internet Address

Classful Addressing

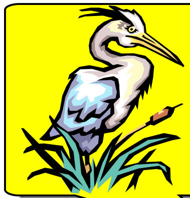
Subnetting

Supernetting

Classless Addressing

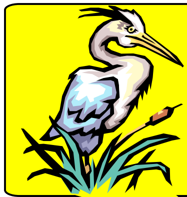
Dynamic Address Configuration

Network Address Translation



Note:

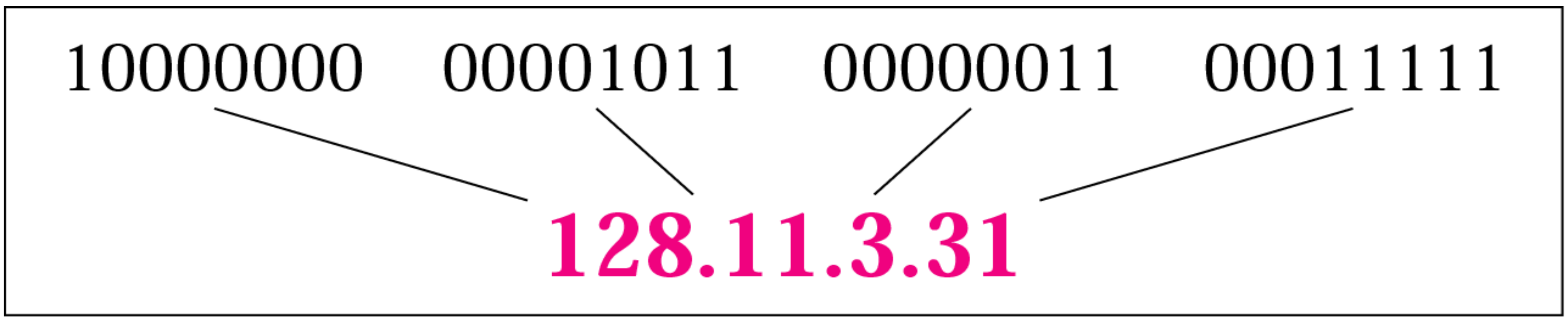
An IP address is a 32-bit address.



Note:

The IP addresses are unique and universal.

Figure 19.9 Dotted-decimal notation



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

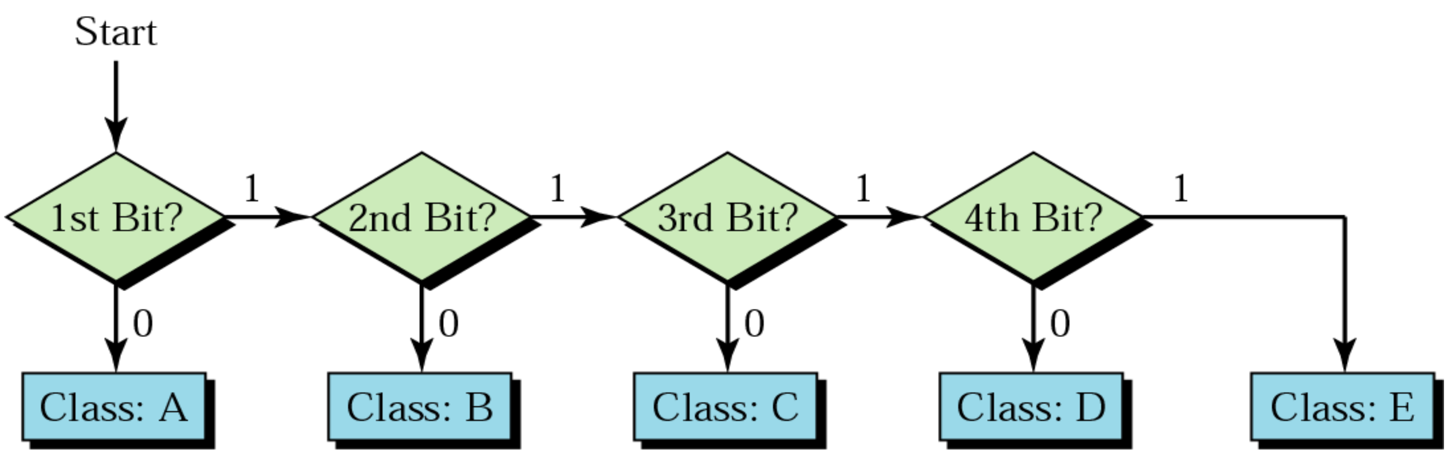
Note:

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

	First byte	Second byte	Third byte	Fourth byte
Class A	**0**			
Class B	**10**			
Class C	**110**			
Class D	**1110**			
Class E	**1111**			

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

	First byte	Second byte	Third byte	Fourth byte
Class A	0 to 127			
Class B	128 to 191			
Class C	192 to 223			
Class D	224 to 239			
Class E	240 to 255			

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

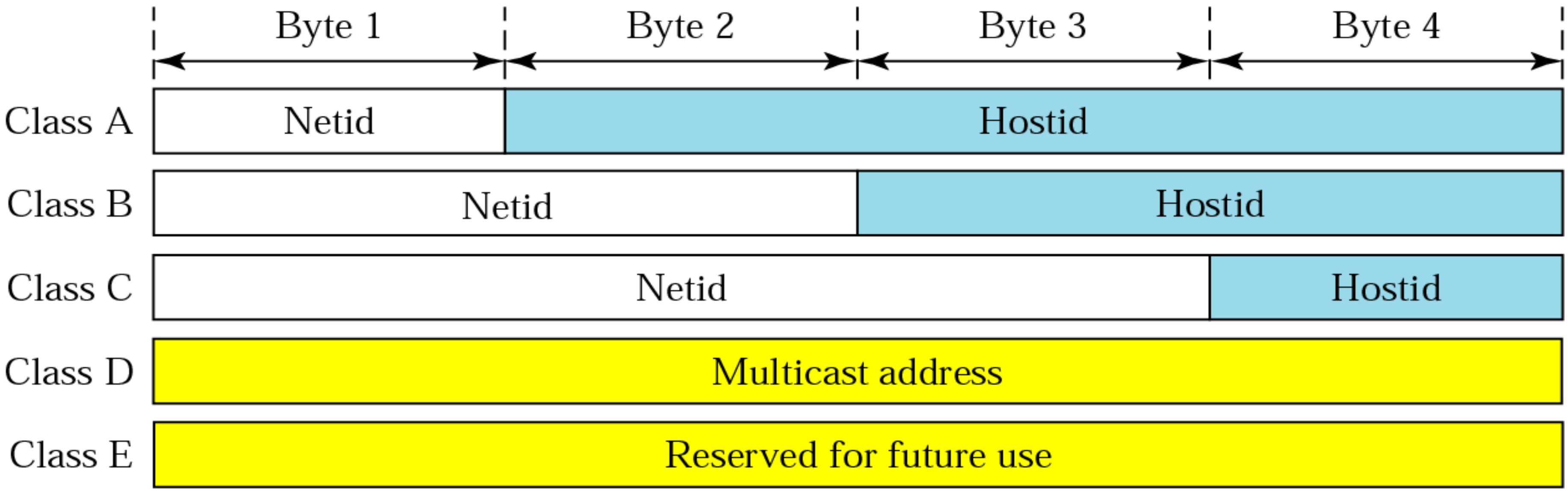
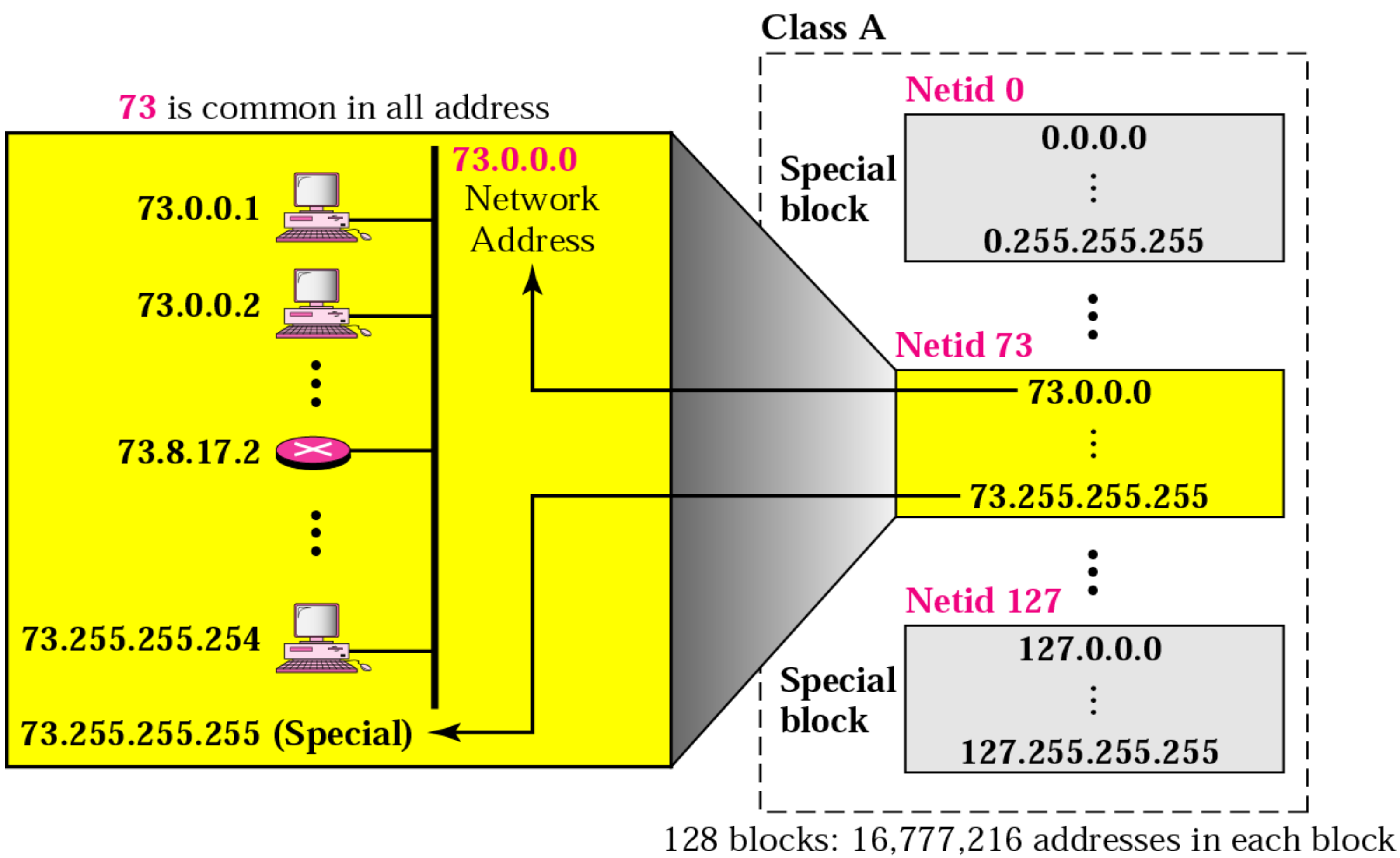
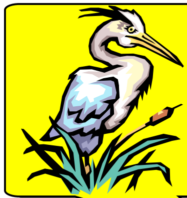


Figure 19.14 Blocks in class A

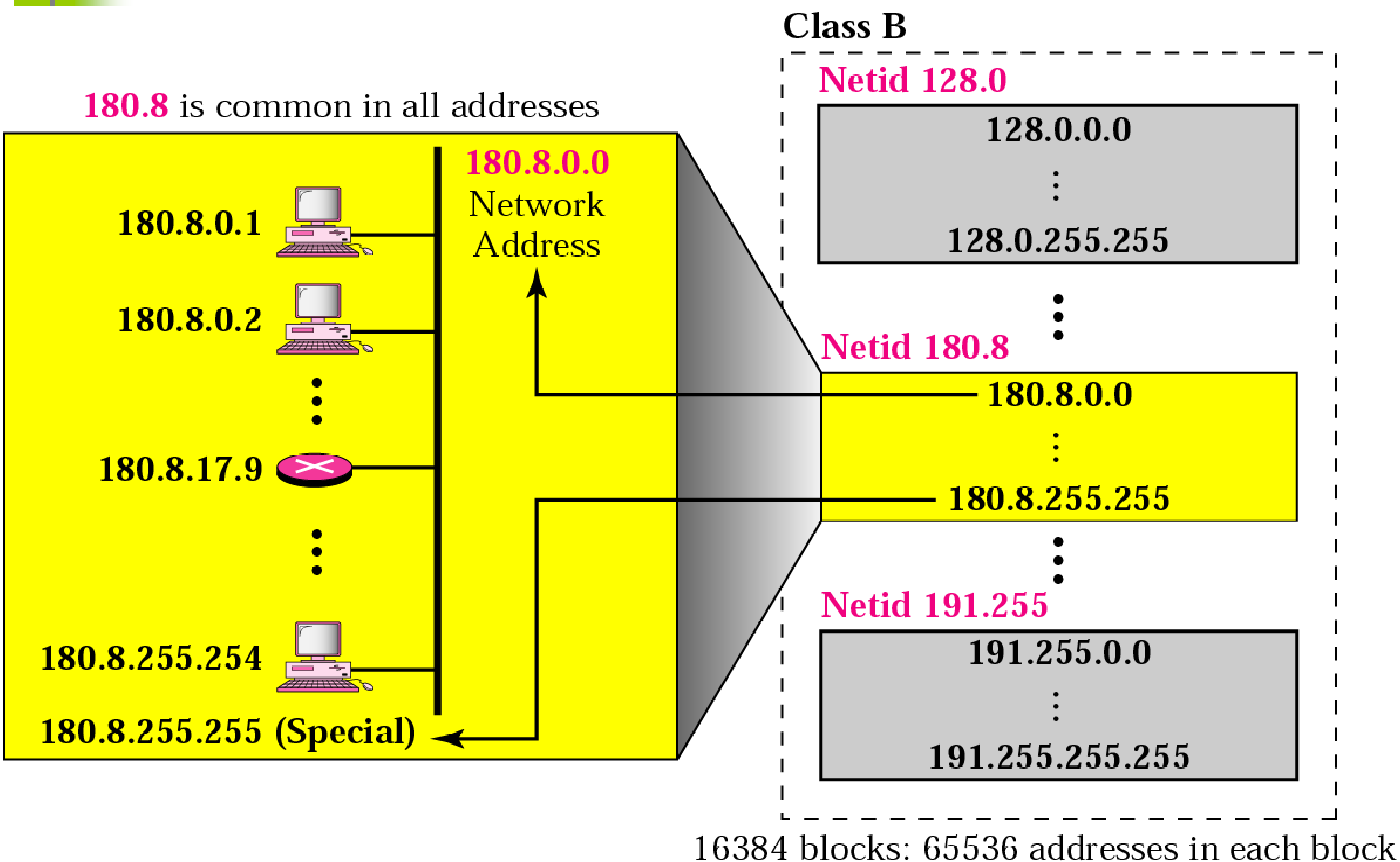


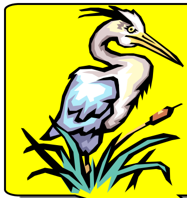


Note:

Millions of class A addresses are wasted.

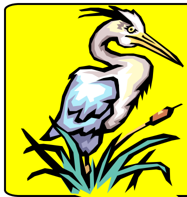
Figure 19.15 Blocks in class B





Note:

Many class B addresses are wasted.



Note:

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

Figure 19.16 Blocks in class C

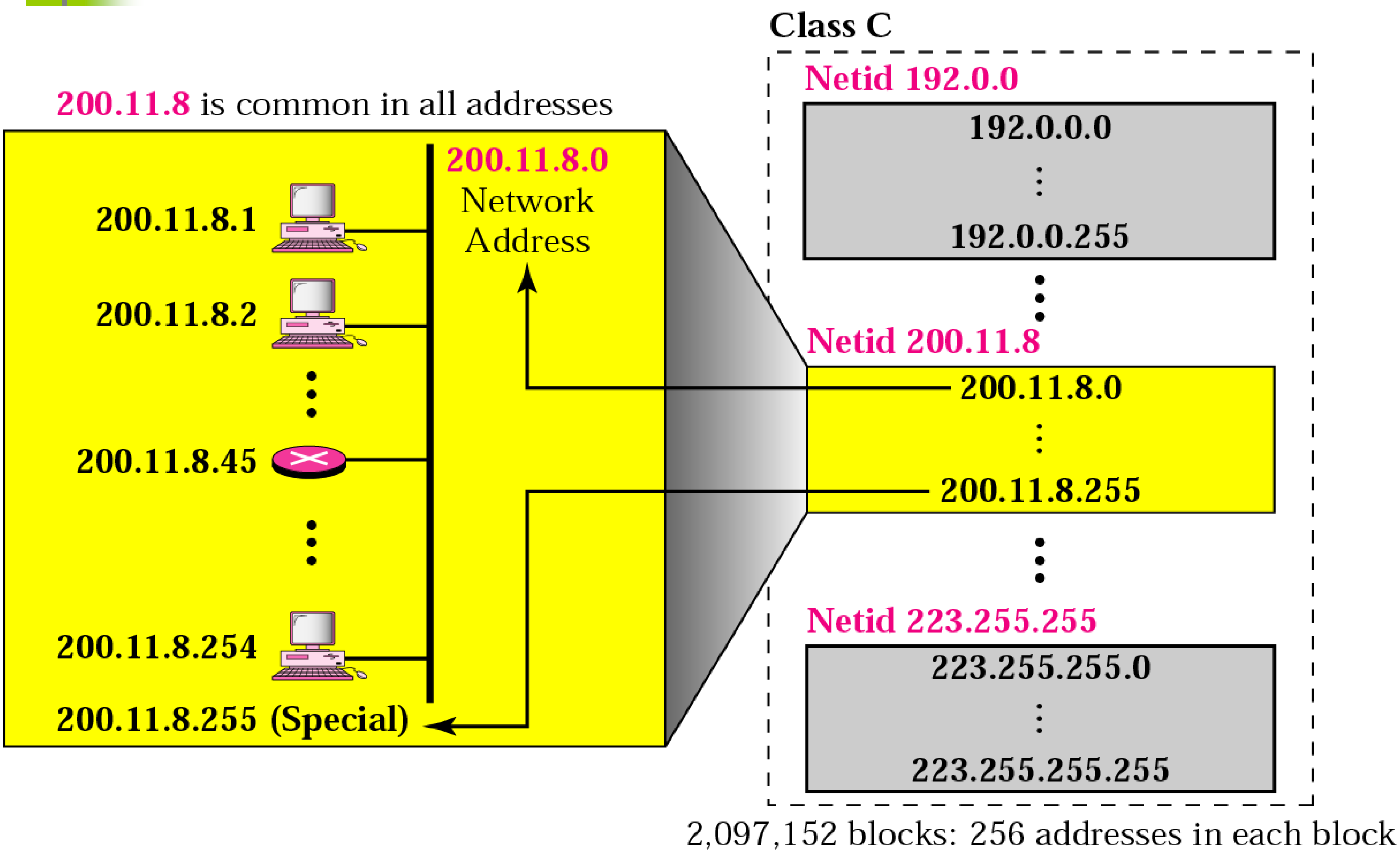
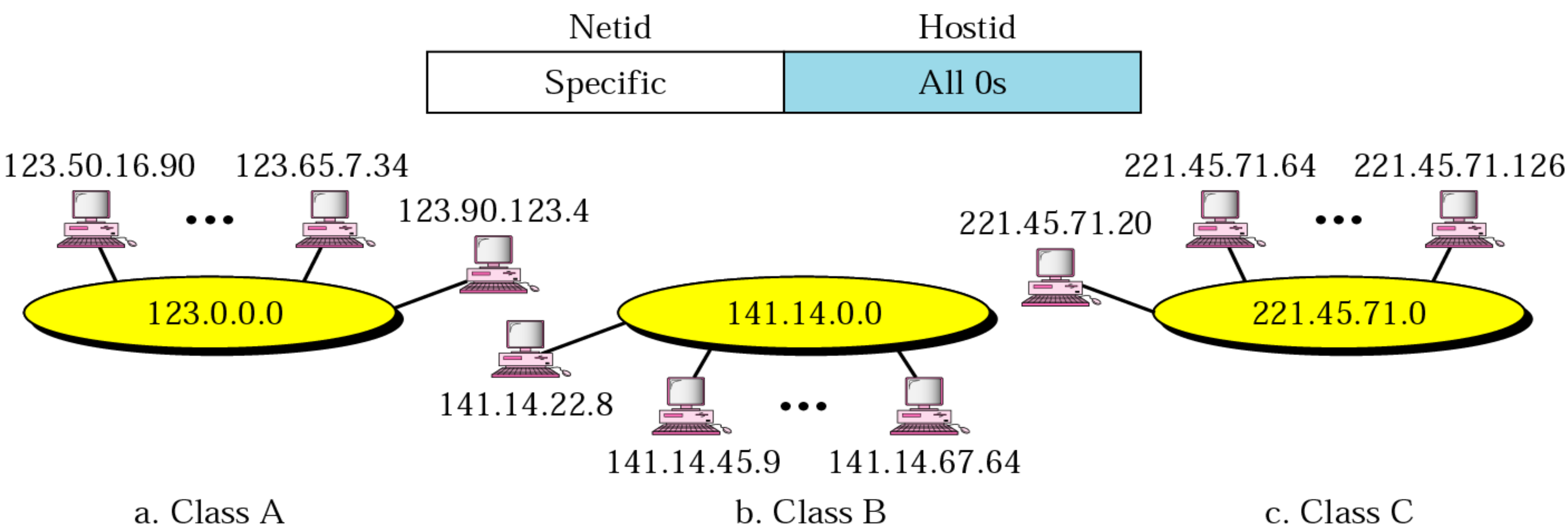
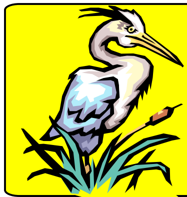


Figure 19.17 Network address





Note:

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.

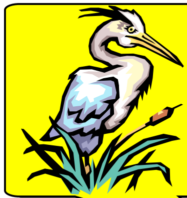
Solution

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

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. *The first group has 64 customers; each needs 256 addresses.*
b. *The second group has 128 customers; each needs 128 addresses.*
c. *The third group has 128 customers; each needs 64 addresses.*

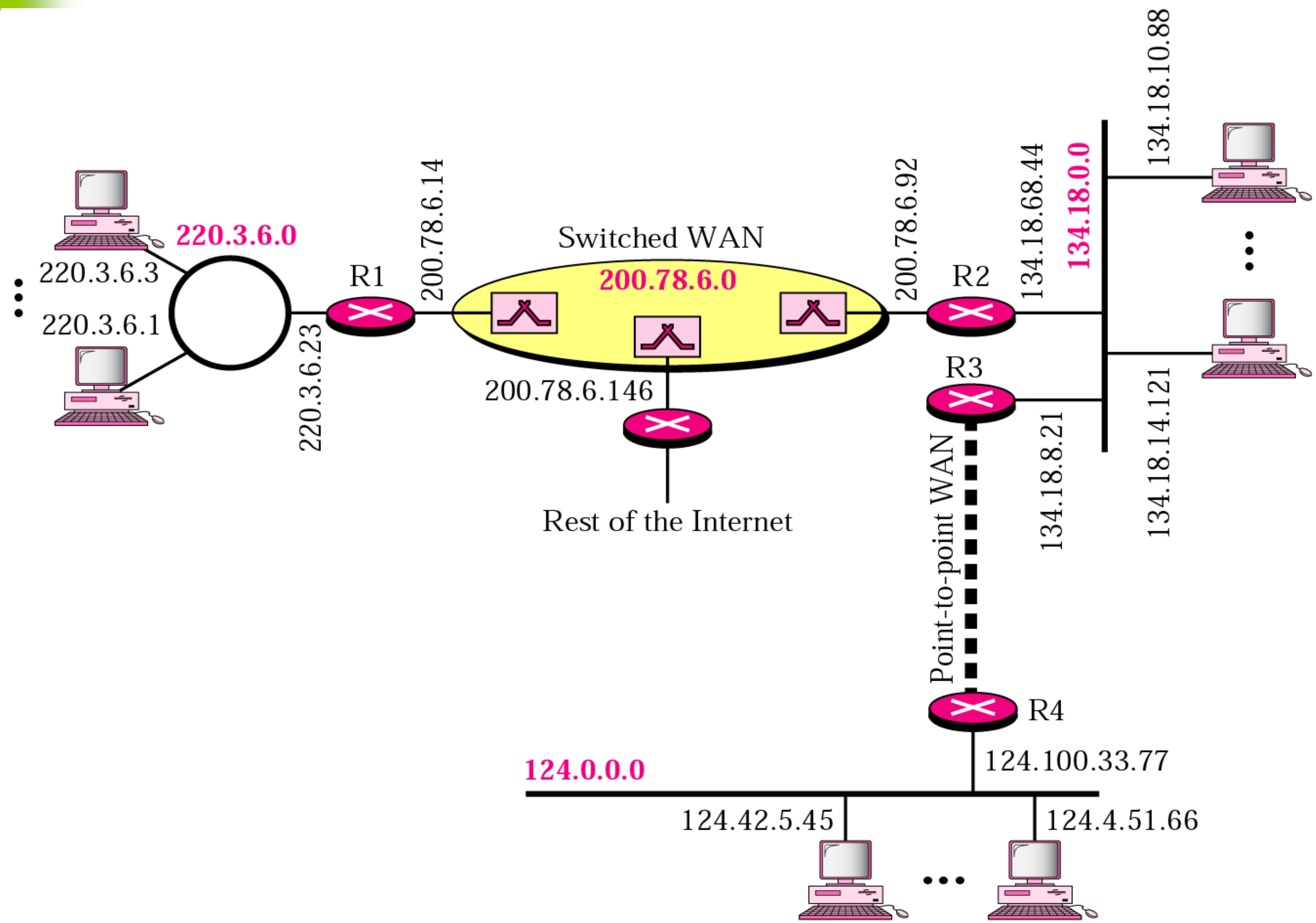
Design the subblocks and find out how many addresses are still available after these allocations.

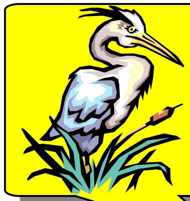


Note:

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

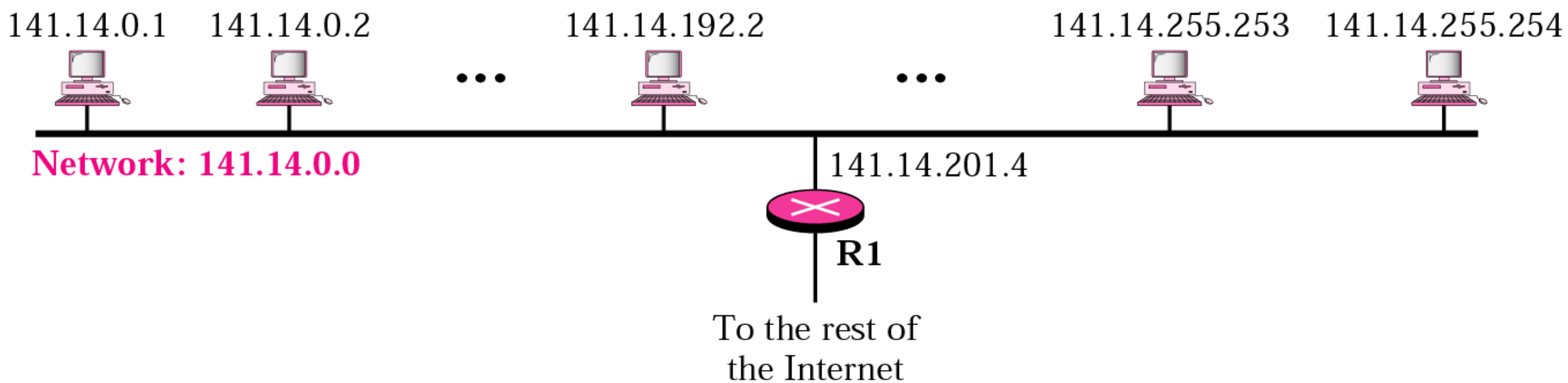


Figure 19.20 A network with three levels of hierarchy (subnetted)

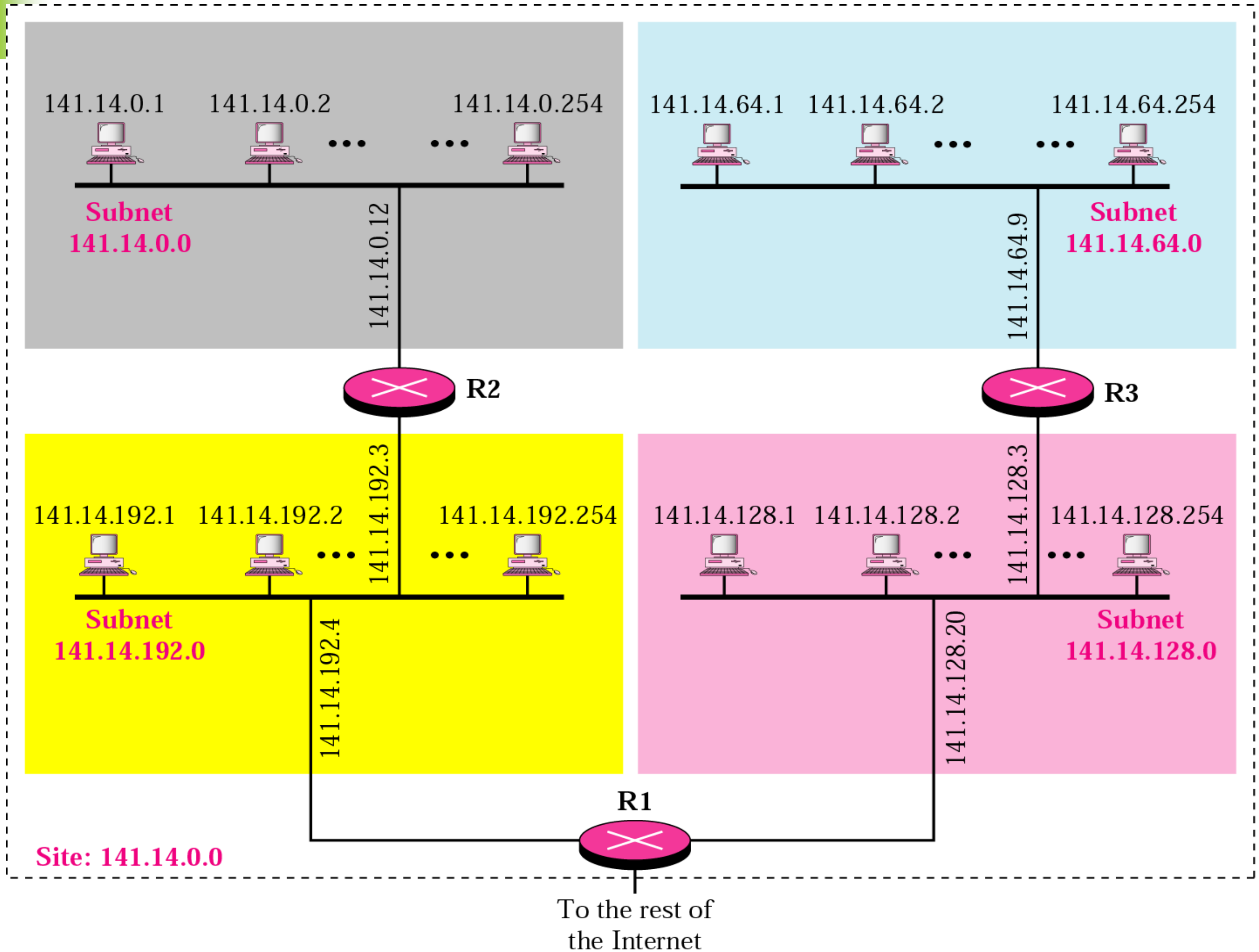


Figure 19.21 Addresses in a network with and without subnetting

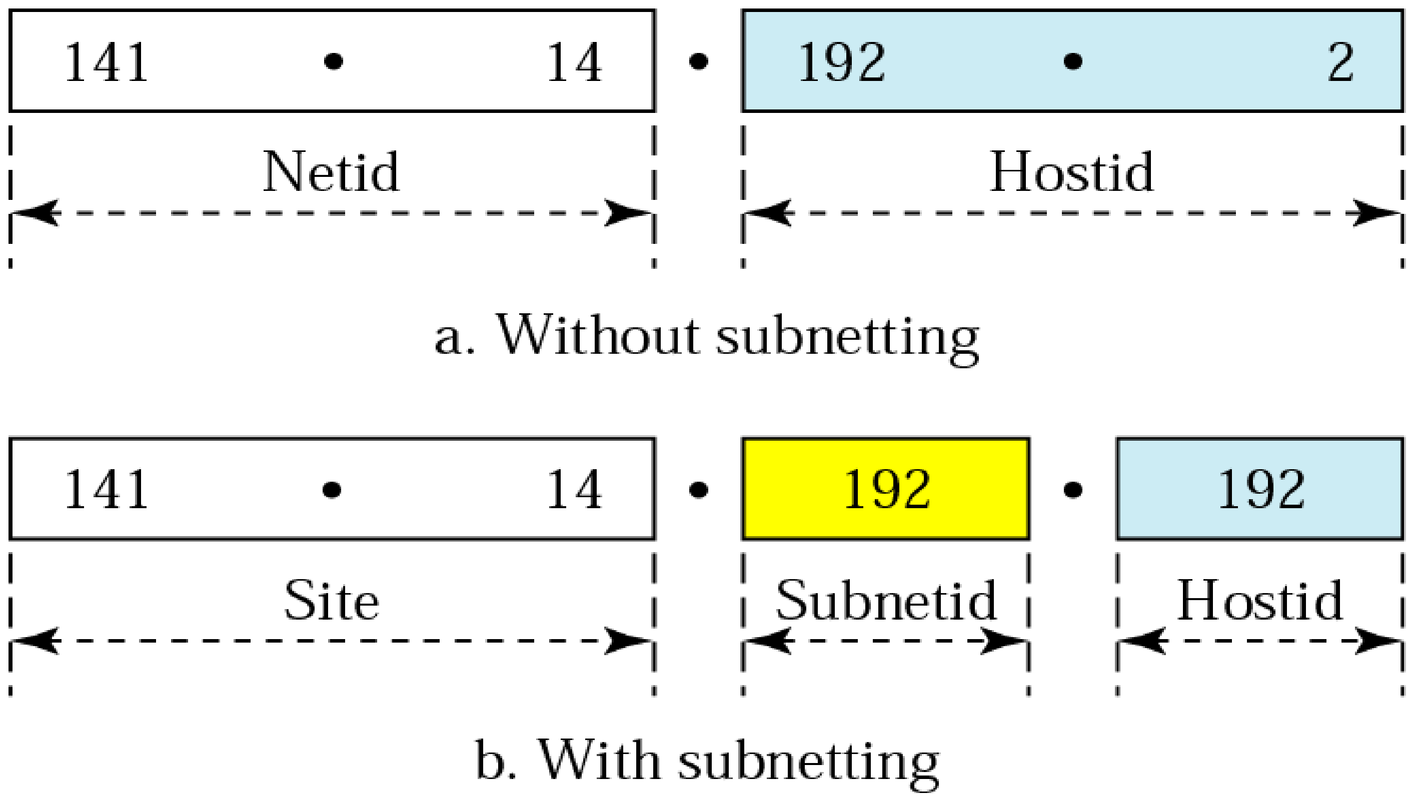
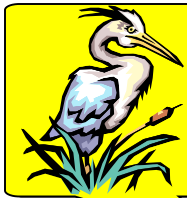


Table 19.1 Default masks

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

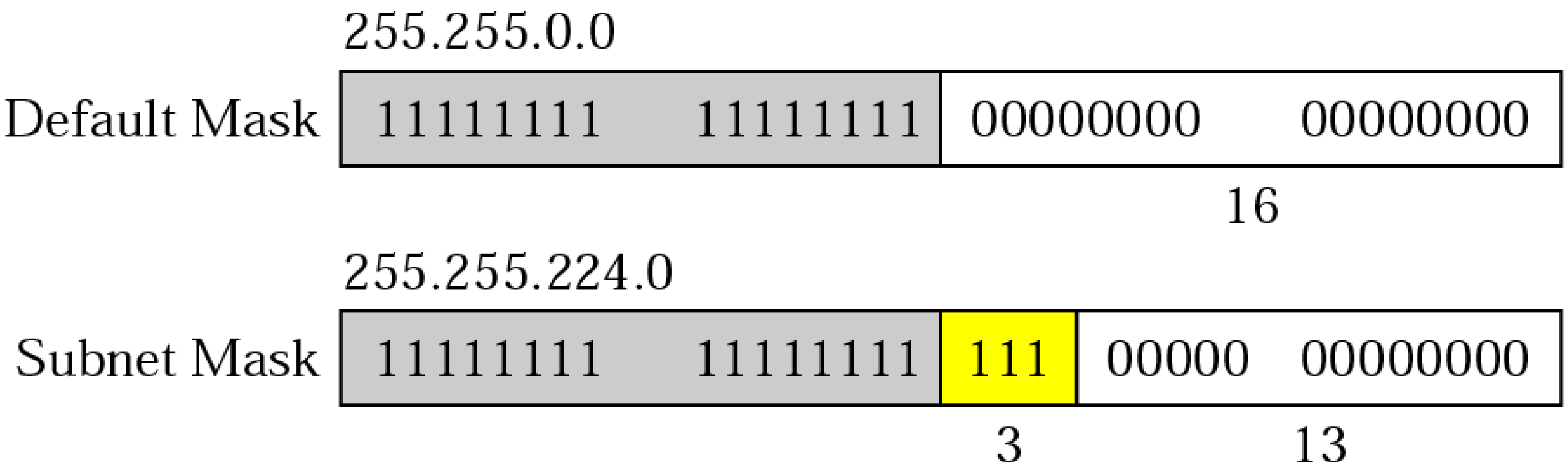
A router outside the organization receives a packet with destination address 190.240.7.91. Show how it finds the network address to route the packet.

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.**

Figure 19.25 NAT

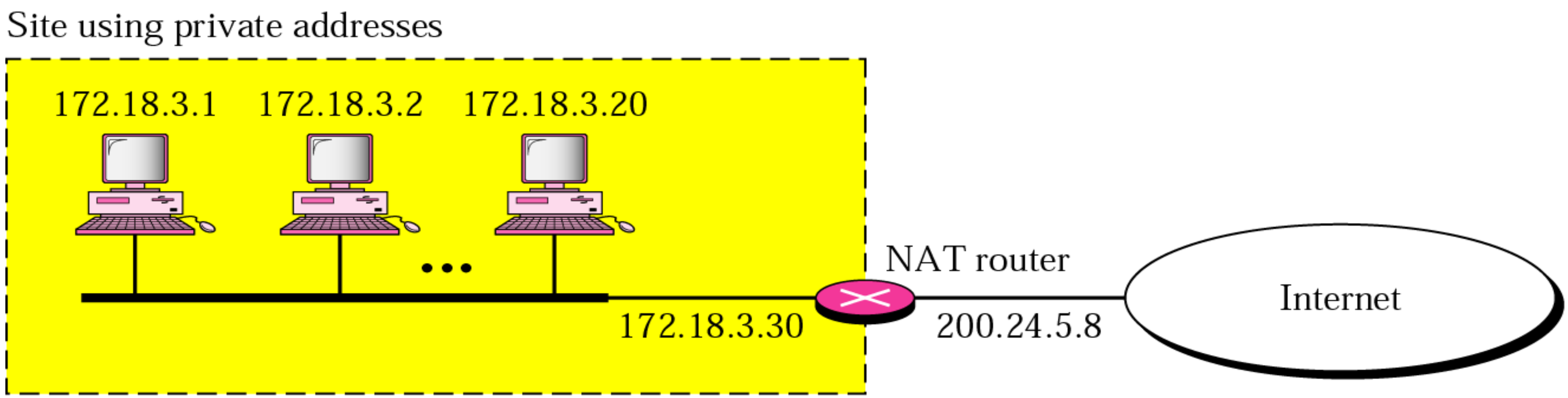


Figure 19.26 Address translation

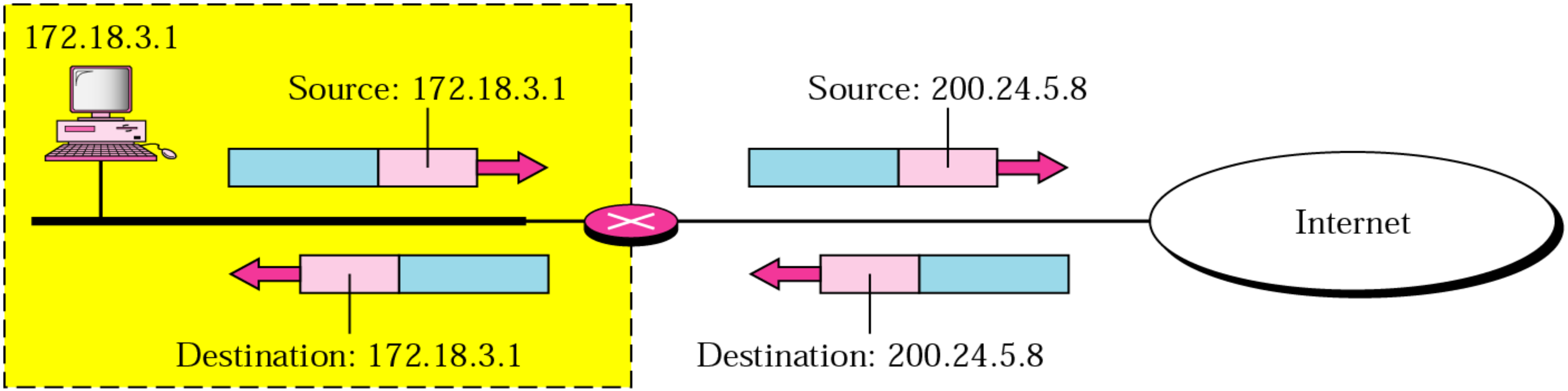


Figure 19.27 Translation

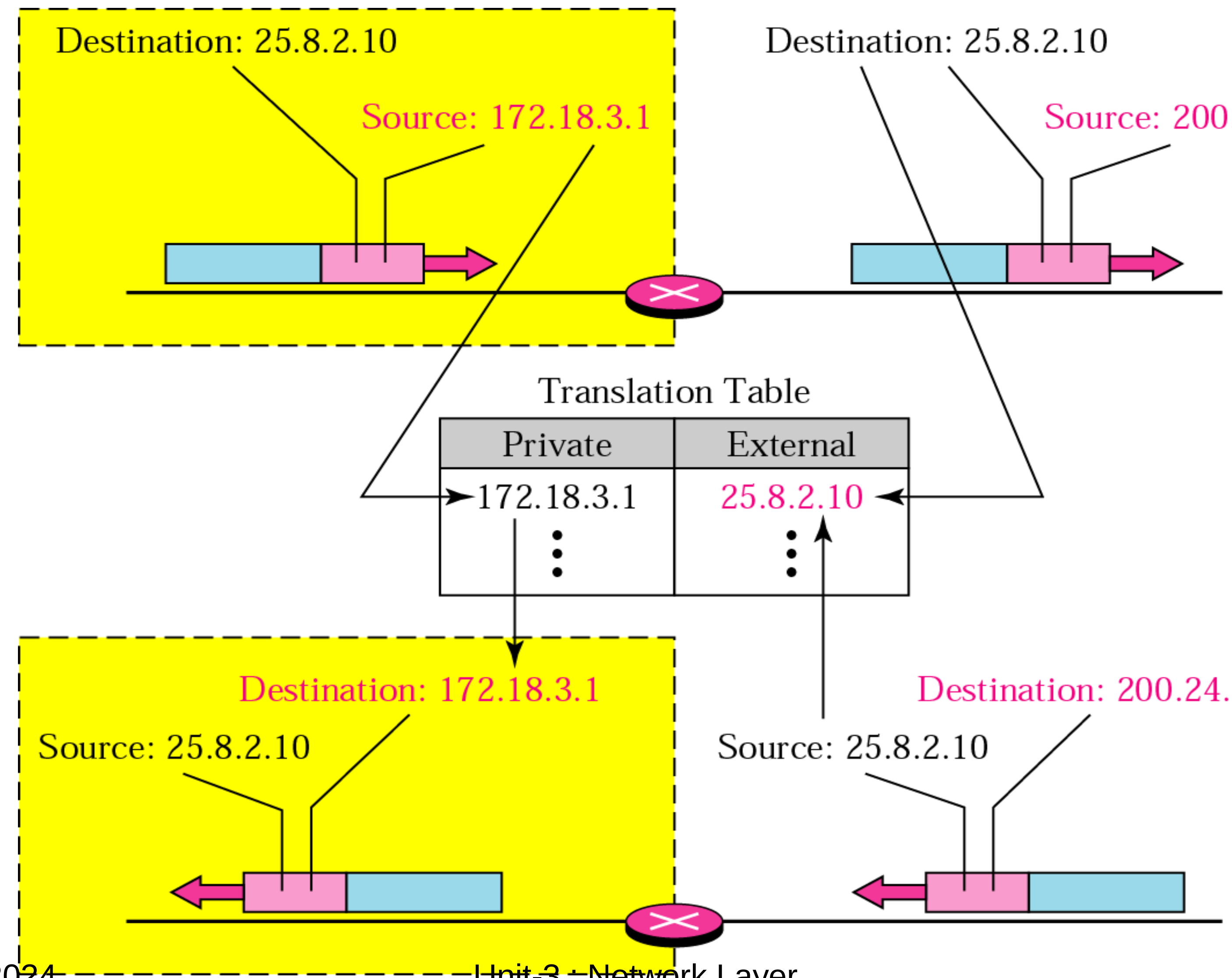


Table 19.3 Five-column translation table

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

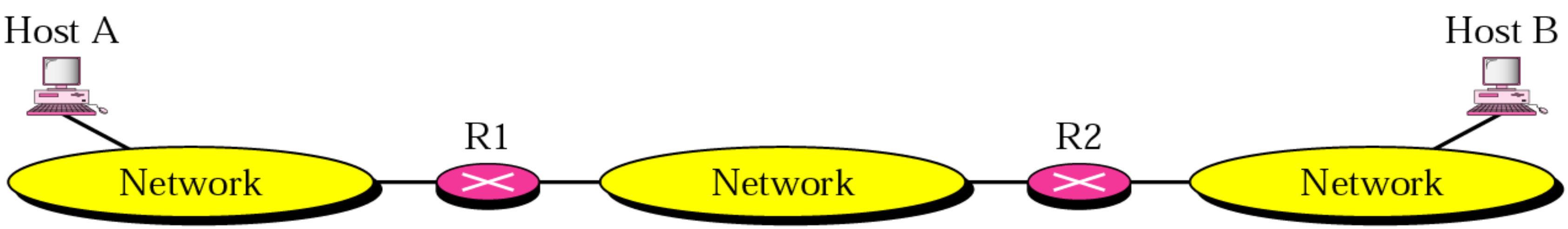
Routing table for R1

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

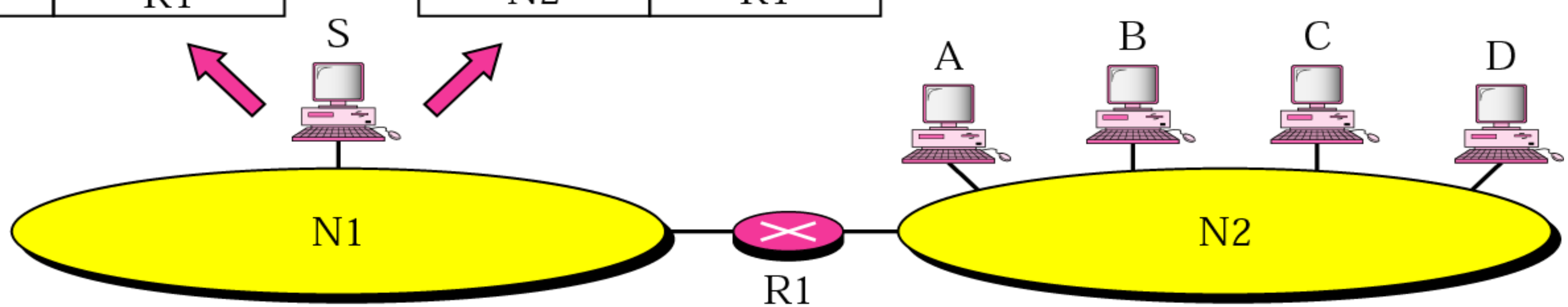


Figure 19.30 Host-specific routing

Routing table for host A

Destination	Next Hop
Host B	R3
N2	R1
N3	R3
......	......

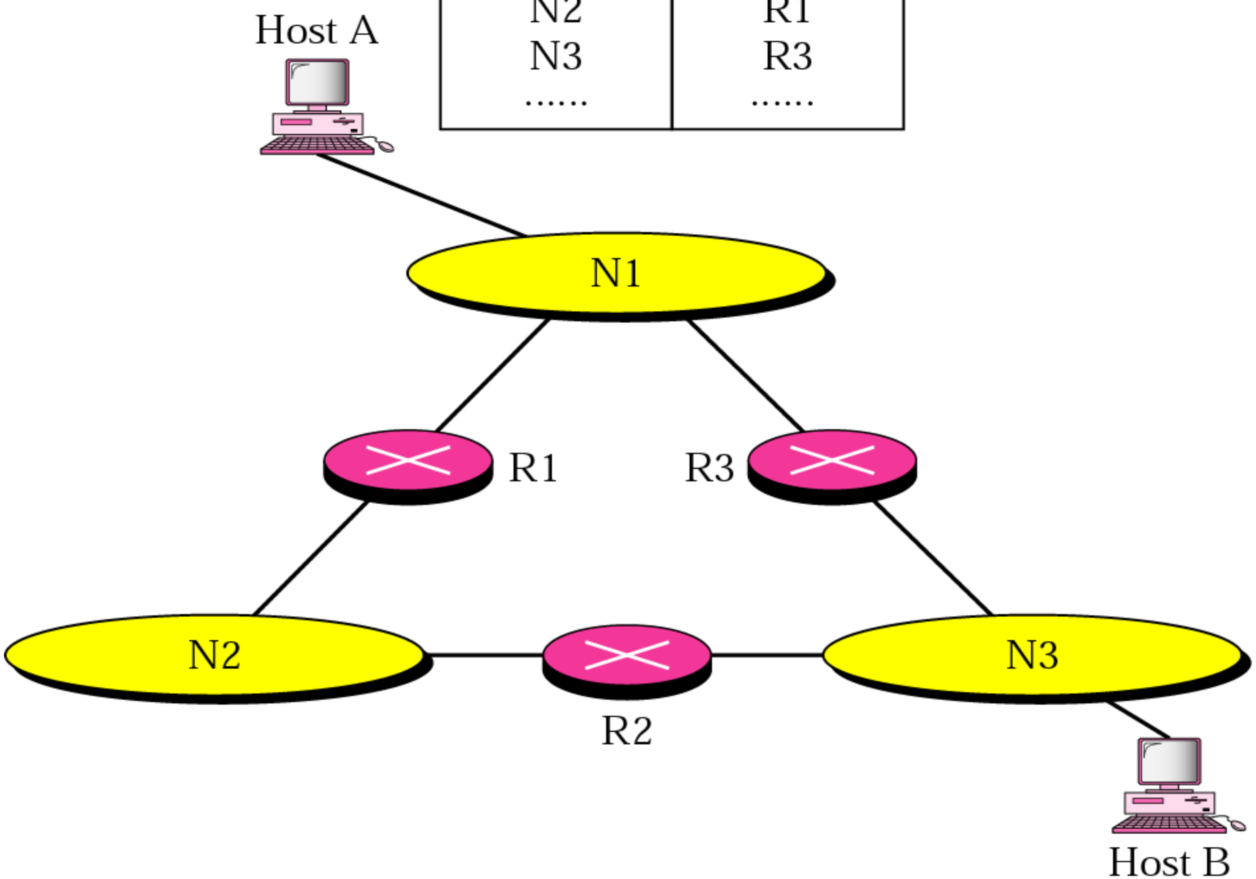


Figure 19.31 Default routing

Routing table for host A

Destination	Next Hop
N2	R1
……	……
Default	R2

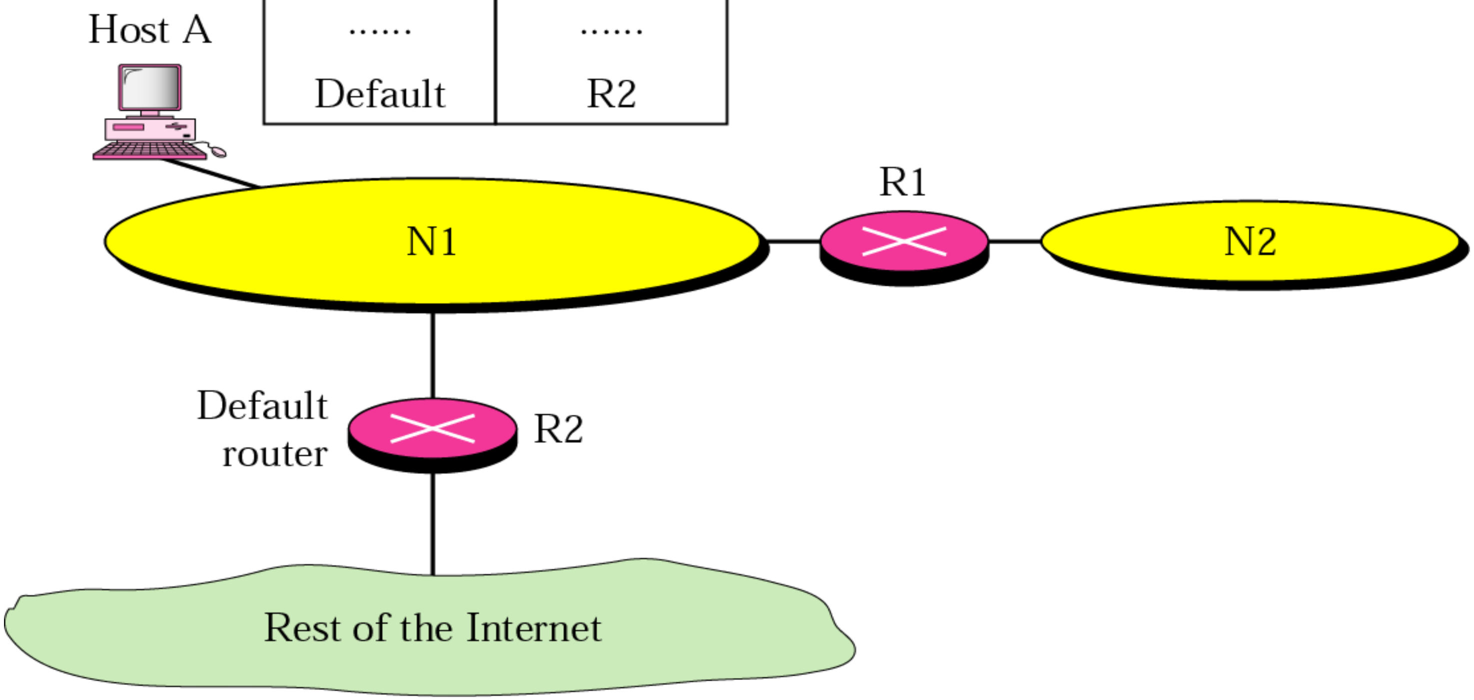


Figure 19.32 Classful addressing routing table

	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

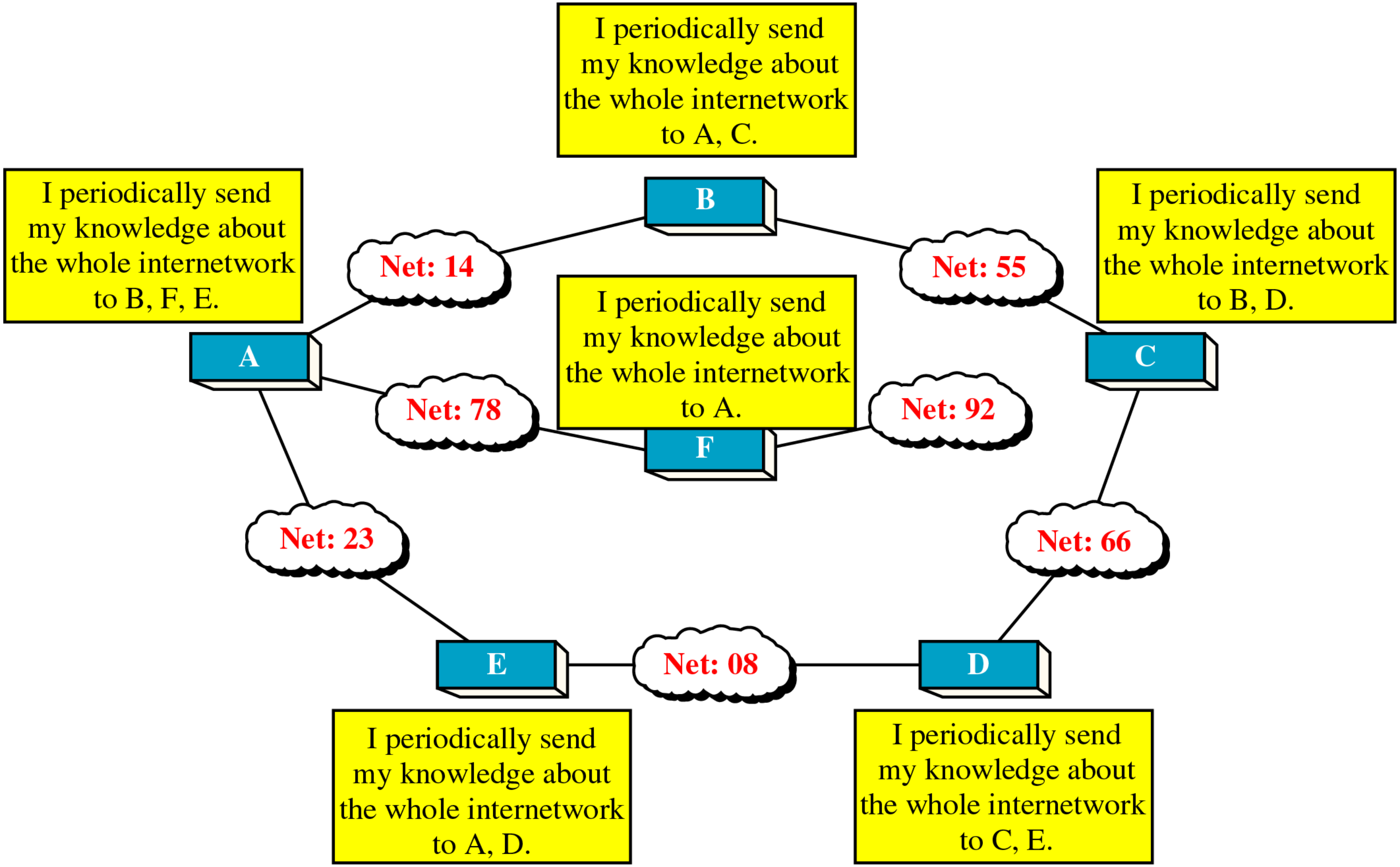


Figure 21-19

Distance Vector Routing Table

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

Figure 21-20

Routing Table Distribution

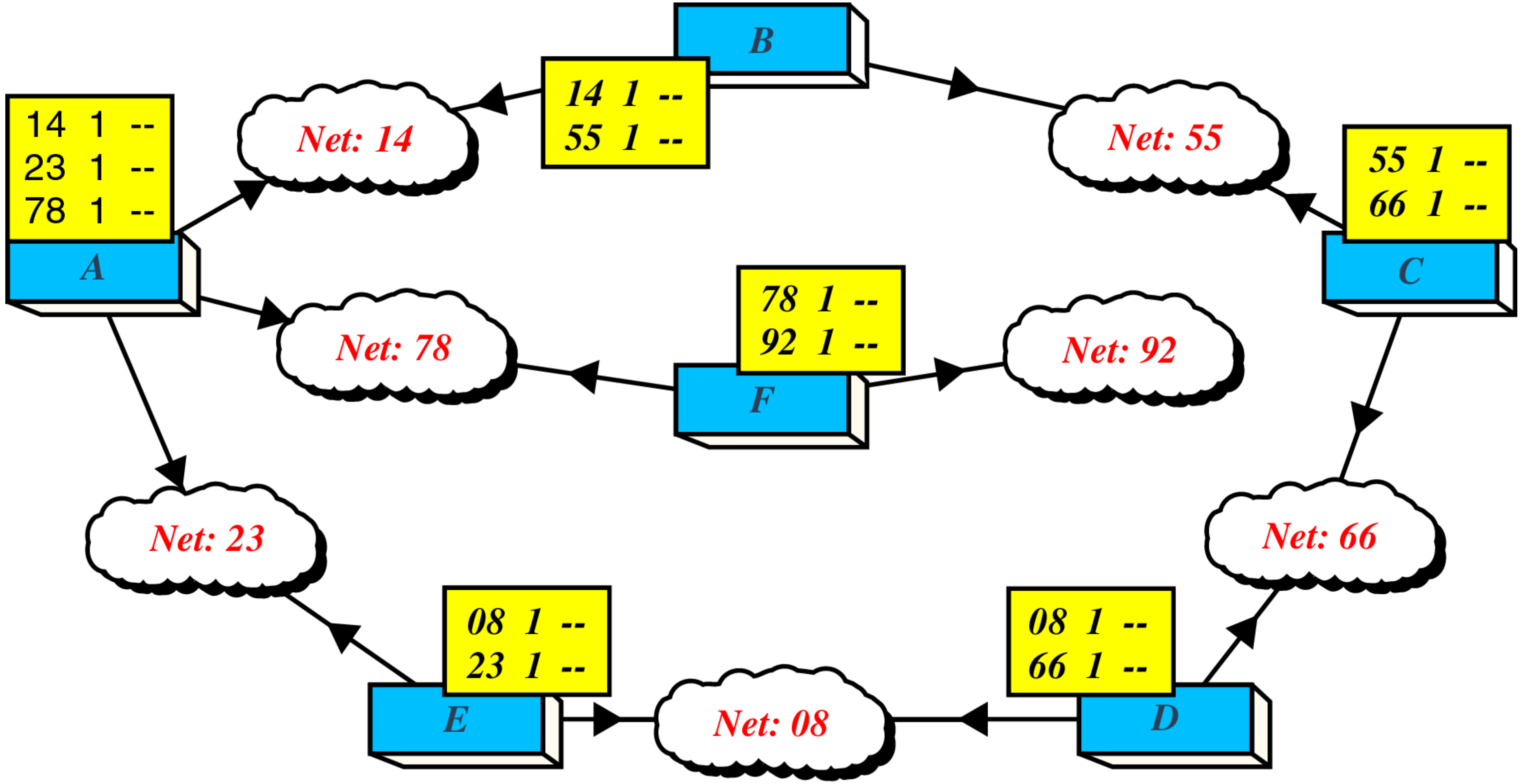


Figure 21-21

Updating Routing Table for Router A

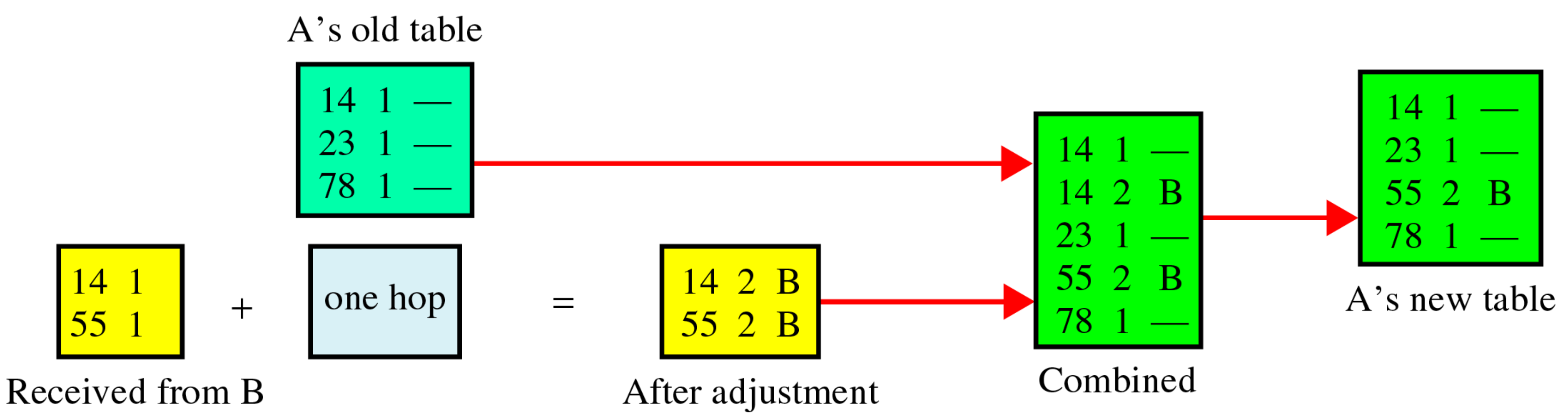


Figure 21-22

Final Routing Tables

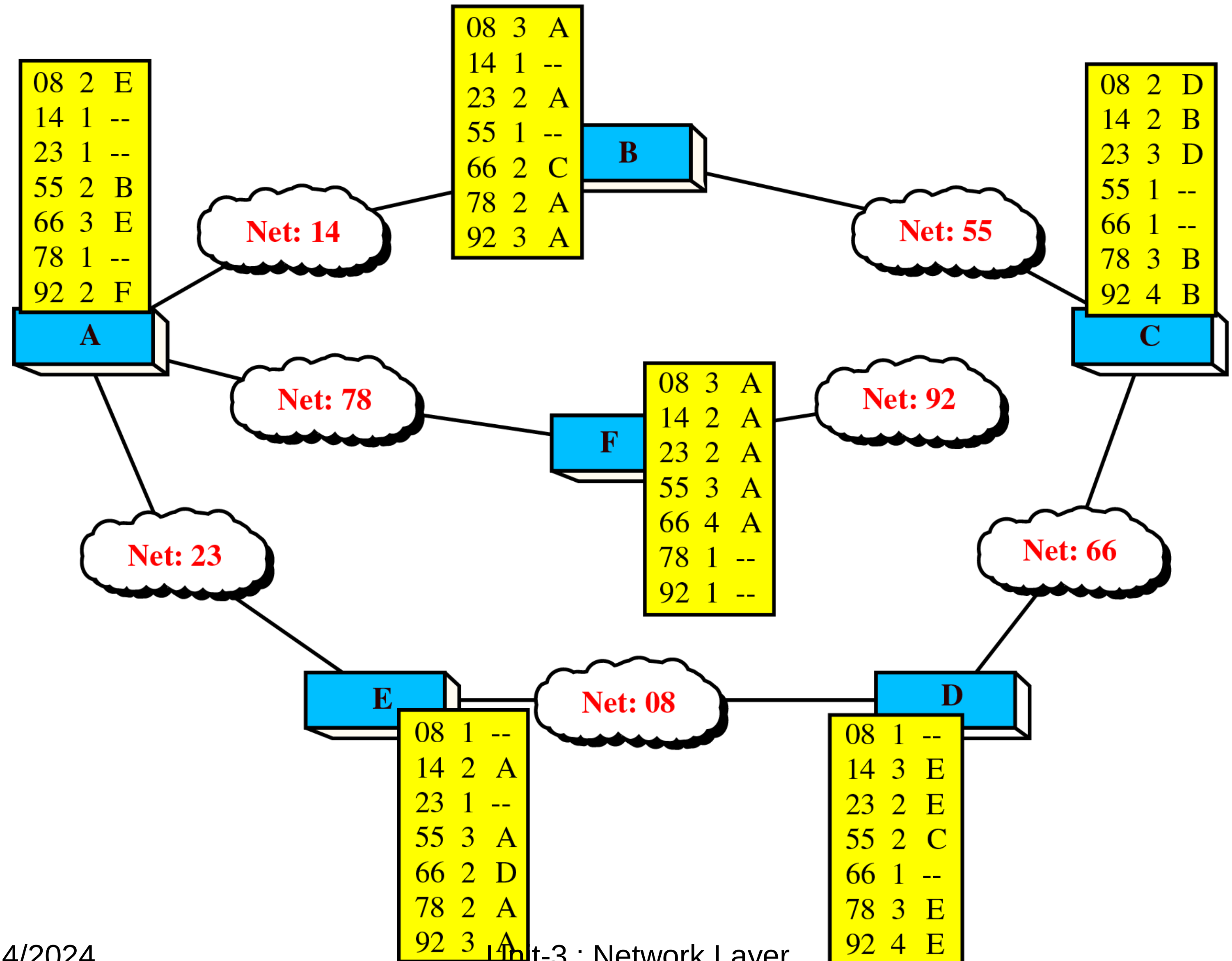


Figure 21-24

Concept of Link State Routing

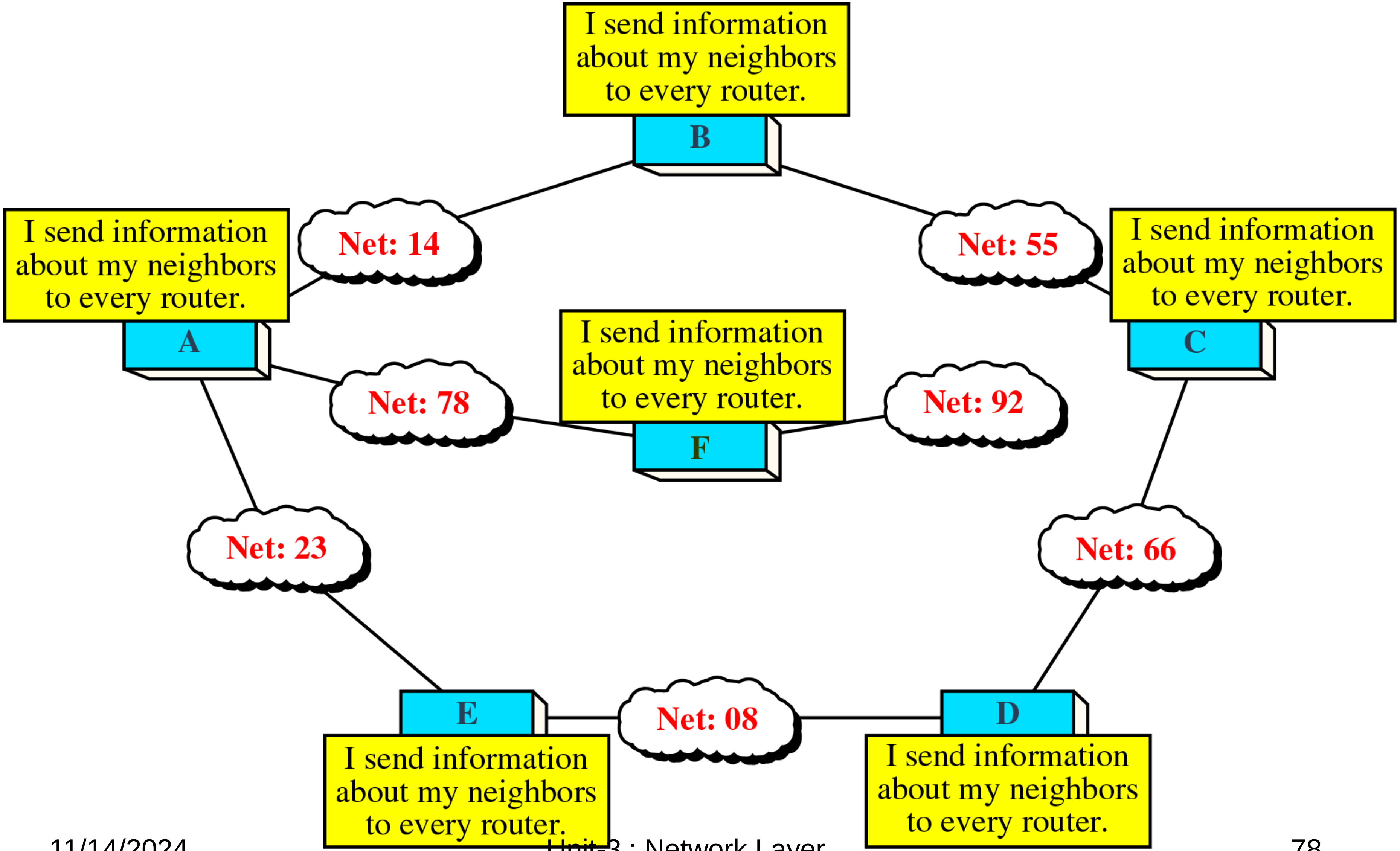


Figure 21-25

Cost in Link State Routing

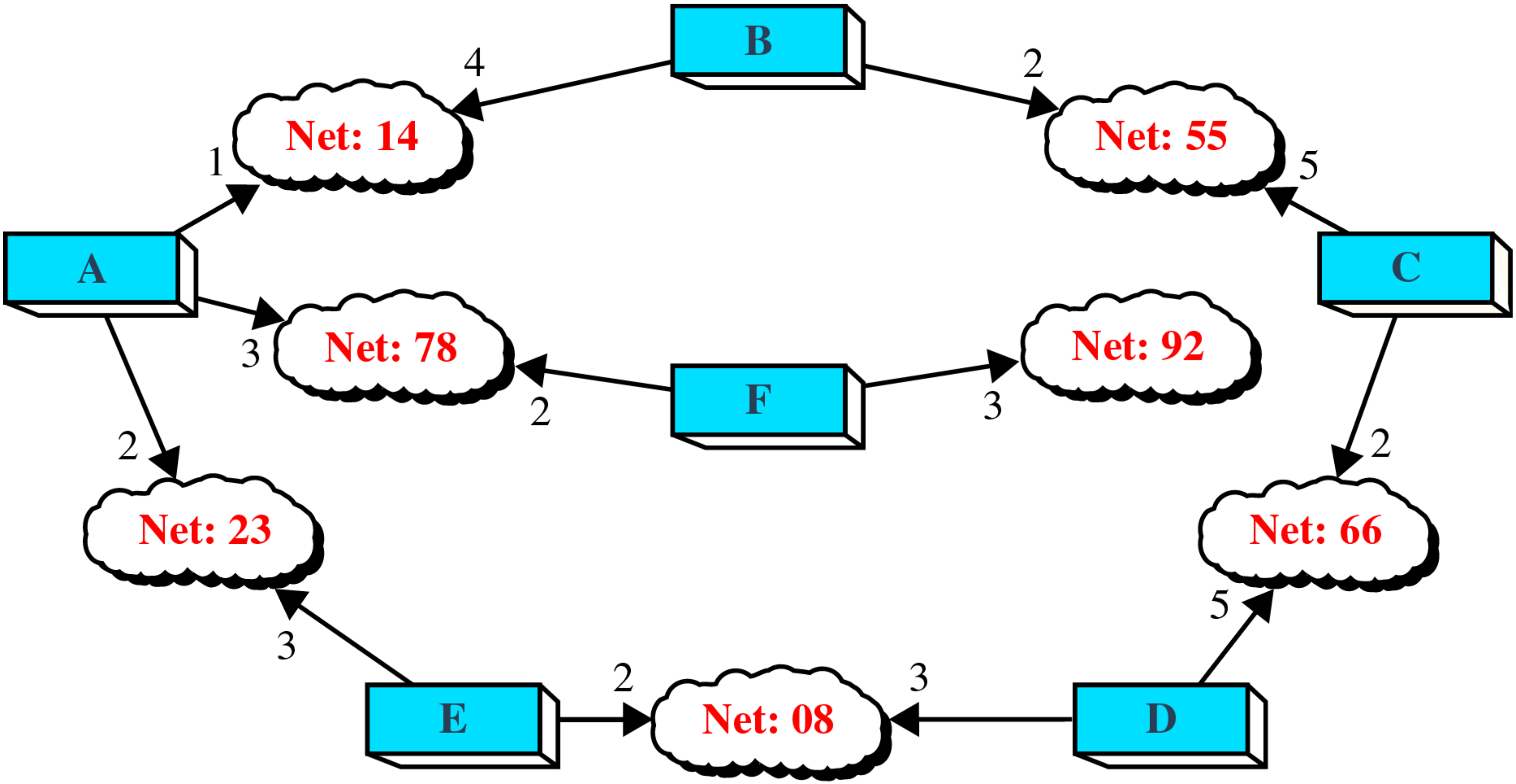


Figure 21-26

Link State Packet

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

Figure 21-27

Flooding of A's LSP

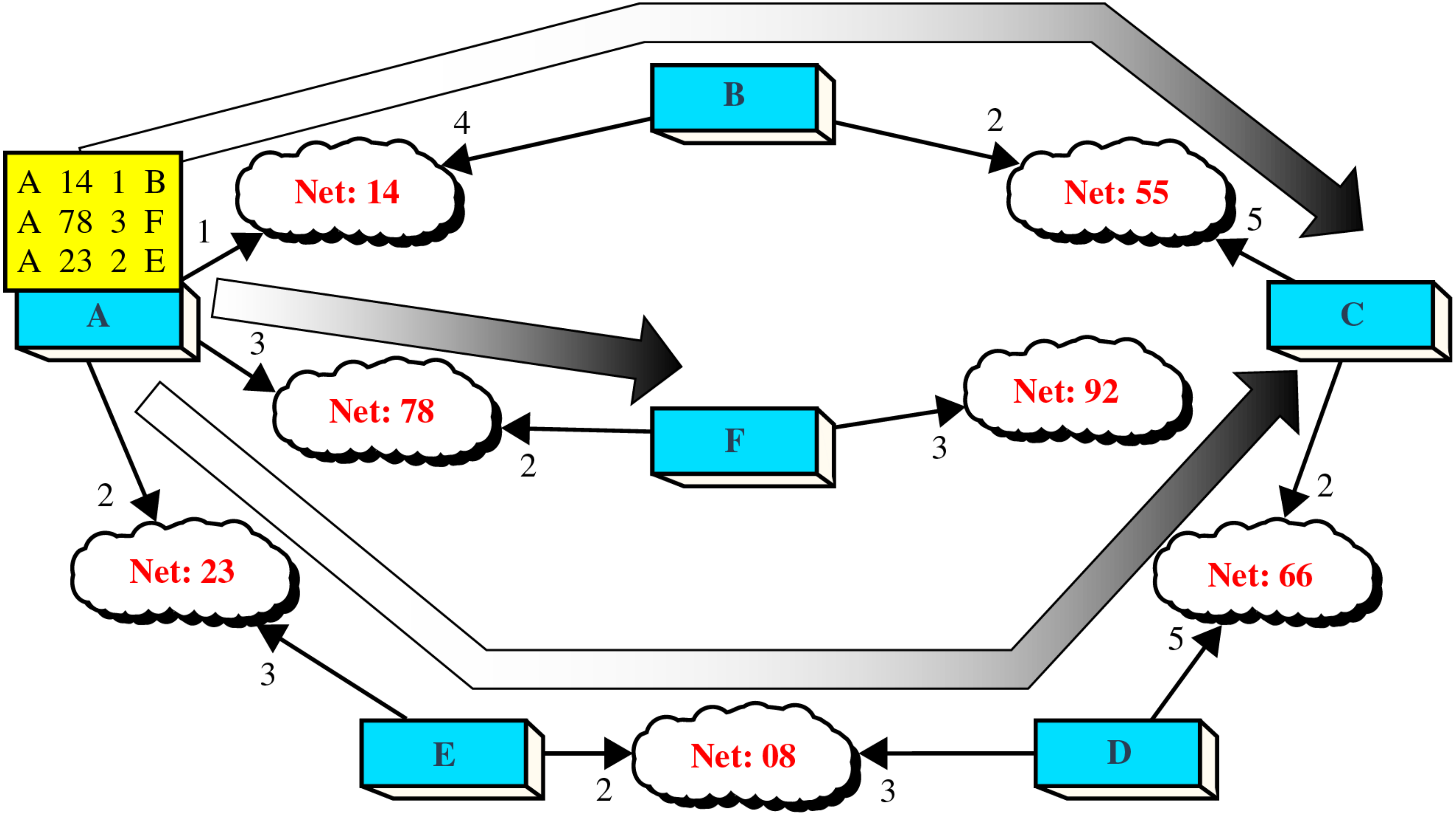
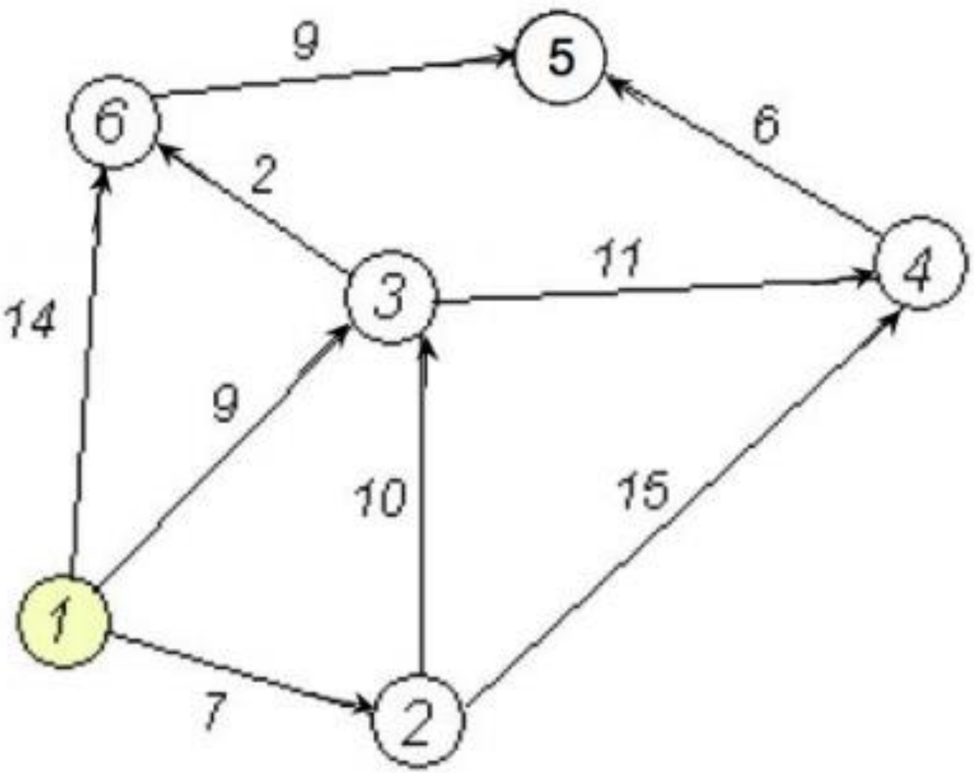


Figure 21-28

Link State Database

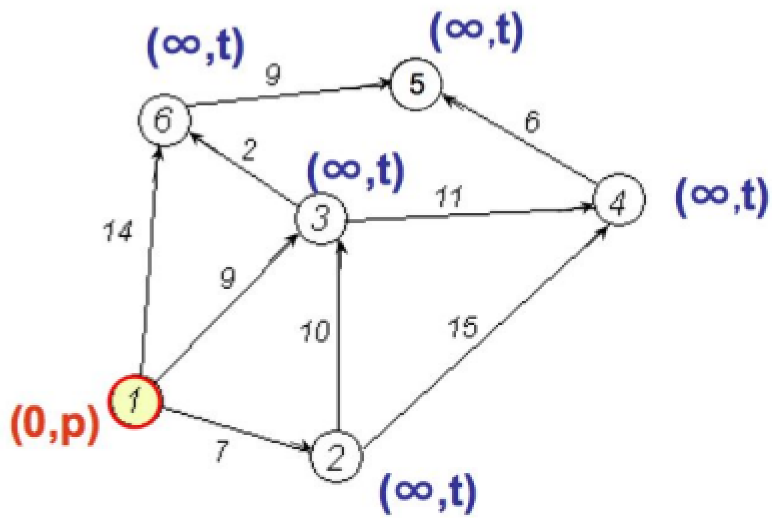
Advertiser	Network	Cost	Neighbor
A	14	1	B
A	78	3	F
A	23	2	E
B	14	4	A
B	55	2	C
C	55	5	B
C	66	2	D
D	66	5	C
D	08	3	E
E	23	3	A
E	08	2	D
F	78	2	A
F	92	3	—

Dijkstra's Algorithm: Example



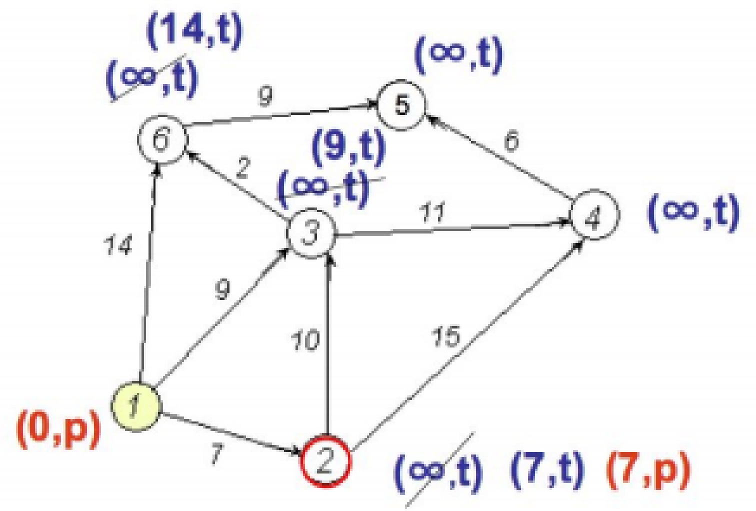
Initialization - Step 1

- Node 1 is designated as the current node
- The state of node 1 is $(0, p)$
- Every other node has state (∞, t)



Step 2

- Nodes 2, 3,and 6 can be reached from the current node 1
- Update distance values for these nodes

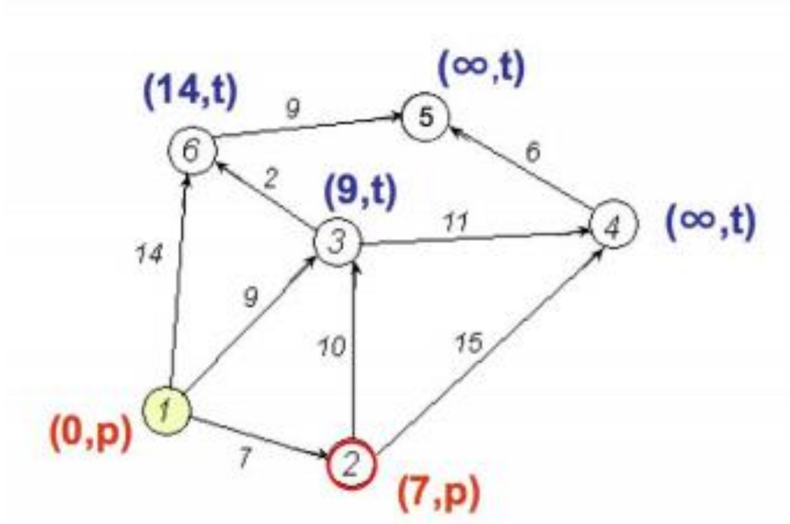


$$d_2 = \min\{\infty, 0+7\} = 7$$
$$d_3 = \min\{\infty, 0+9\} = 9$$
$$d_6 = \min\{\infty, 0+14\} = 14$$

- Now, among the nodes 2, 3, and 6, node 2 has the smallest distance value
- The status label of node 2 changes to permanent, so its state is $(7, p)$, while the status of 3 and 6 remains temporary
- Node 2 becomes the current node

Step 3

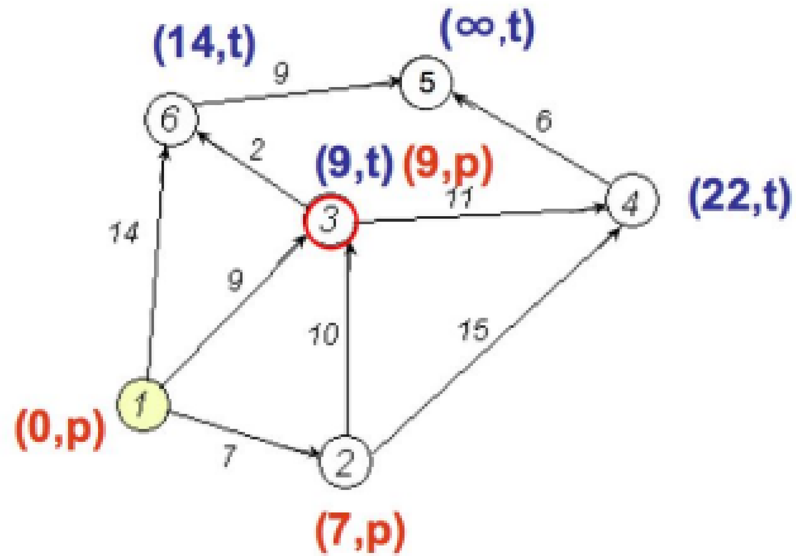
Graph at the end of Step 2



We are not done, not all nodes have been reached from node 1, so we perform another iteration (back to Step 2)

Another Implementation of Step 2

- Nodes 3 and 4 can be reached from the current node 2
- Update distance values for these nodes



$$d_3 = \min\{9, 7 + 10\} = 9$$
$$d_6 = \min\{\infty, 7 + 15\} = 22$$

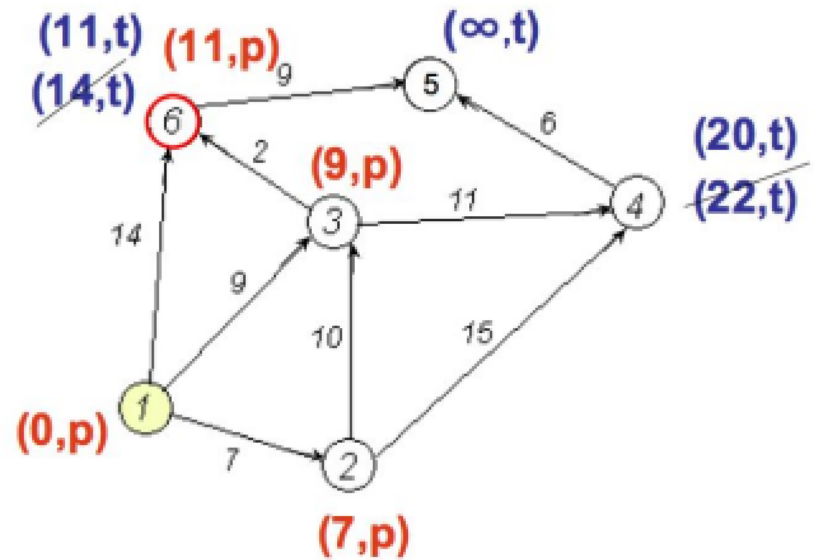
- Now, between the nodes 3 and 4 node 3 has the smallest distance value
- The status label of node 3 changes to permanent, while the status of 6 remains temporary
- Node 3 becomes the current node

We are not done (Step 3 fails), so we perform another Step 2

Another Step 2

- Nodes 6 and 4 can be reached from the current node 3
- Update distance values for them

$$d_4 = \min\{22, 9+11\} = 20$$
$$d_6 = \min\{14, 9+2\} = 11$$

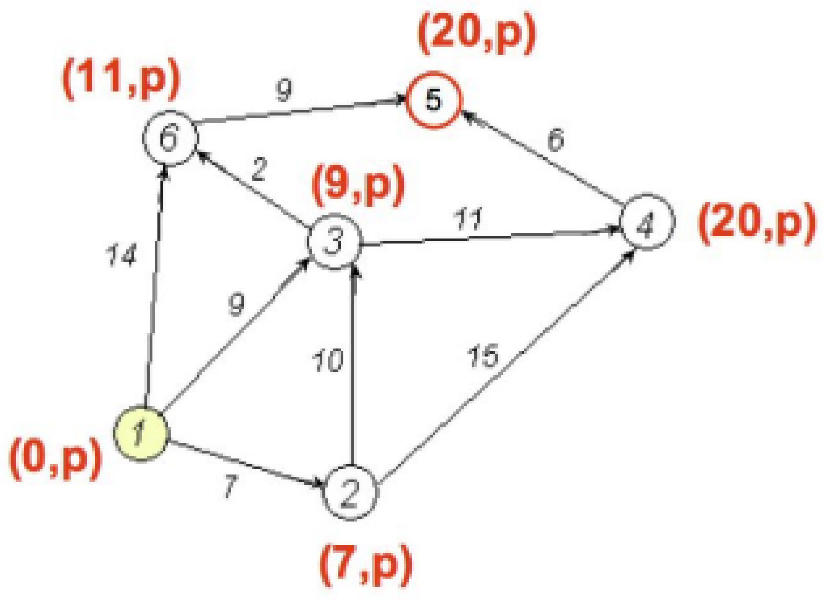


- Now, between the nodes 6 and 4 node 6 has the smallest distance value
- The status label of node 6 changes to permanent, while the status of 4 remains temporary
- Node 6 becomes the current node

We are not done (Step 3 fails), so we perform another Step 2

Another Step 2

- Node 5 can be reached from the current node 6
- Update distance value for node 5



$$d_5 = \min\{\infty, 11 + 9\} = 20$$

- Now, node 5 is the only candidate, so its status changes to permanent

- Node 5 becomes the current node

From node 5 we cannot reach any other node. Hence, node 4 gets permanently labeled and we are done.

Figure 21-29

Costs in the Dijkstra Algorithm

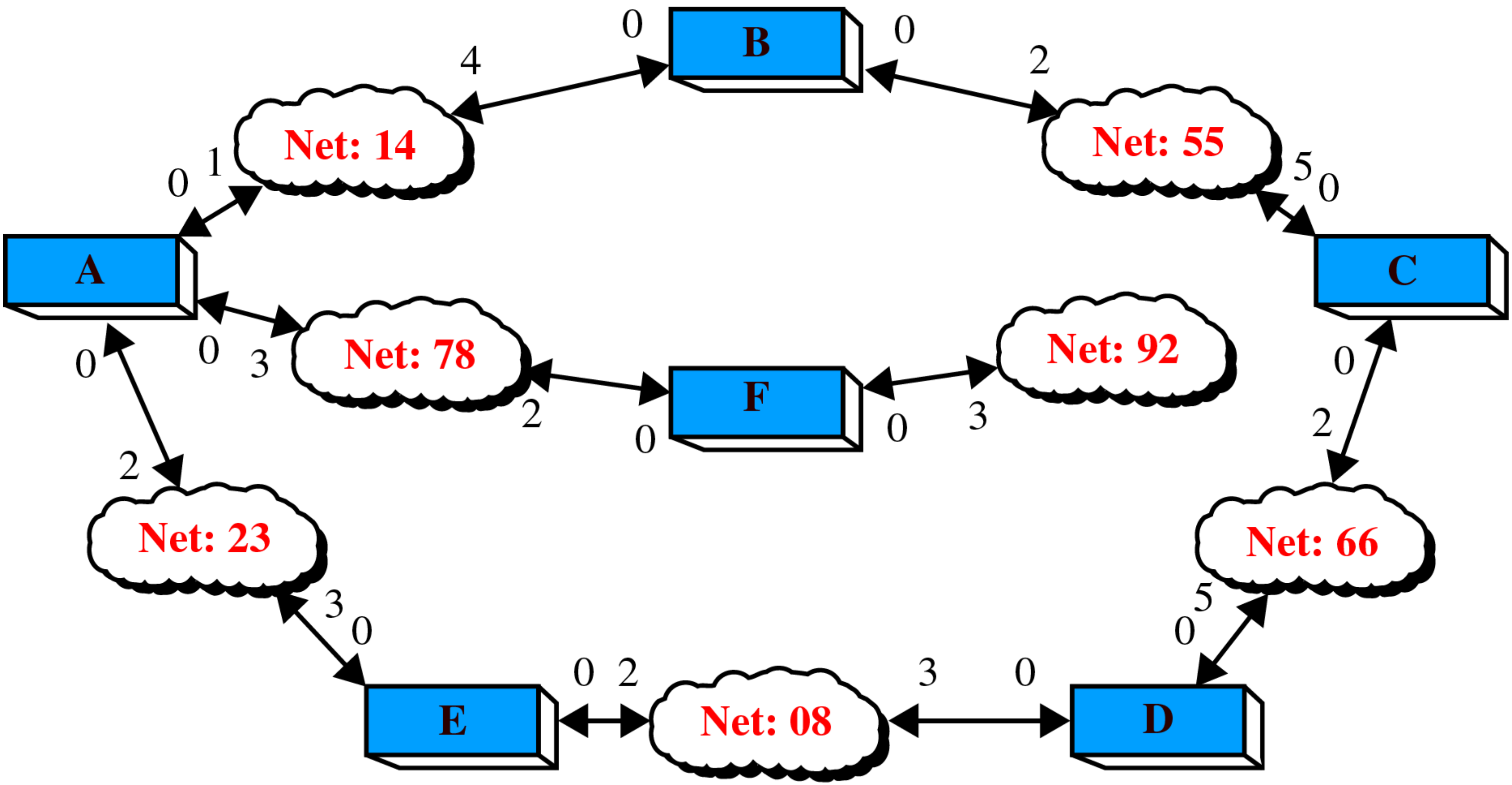
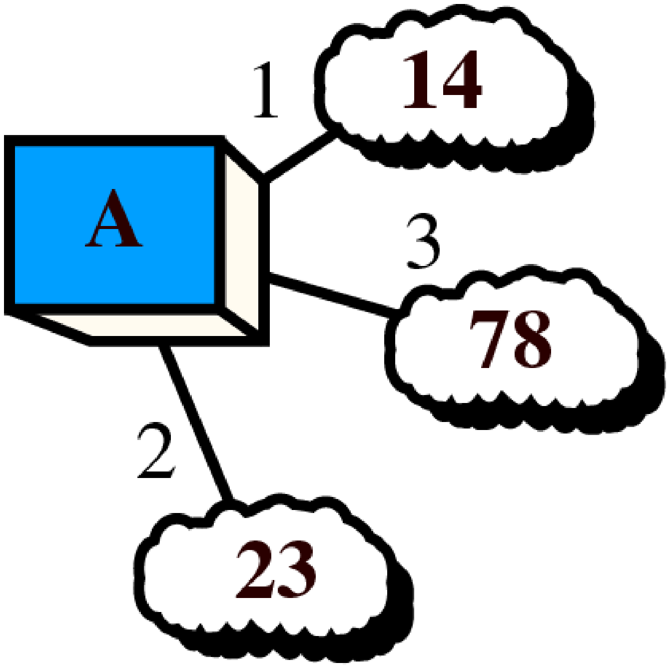


Figure 21-30, Part I

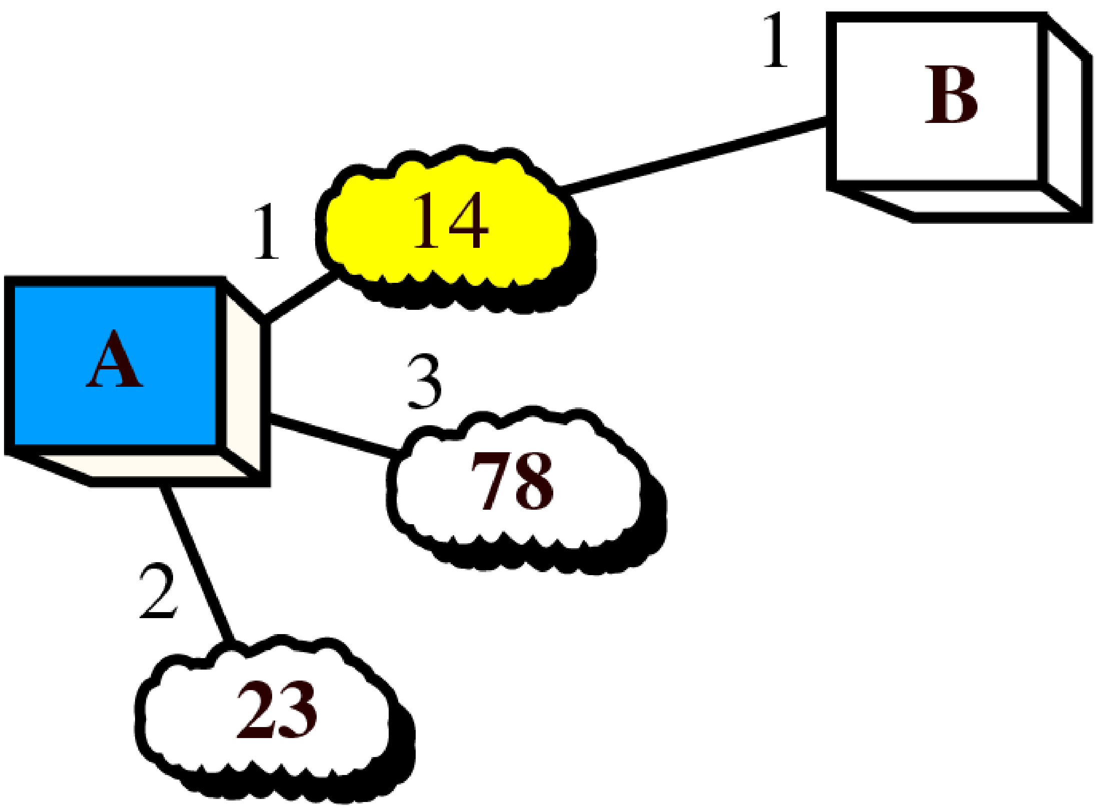
Shortest Path Calculation, Part I



Root is A, networks
14, 78, 23 added

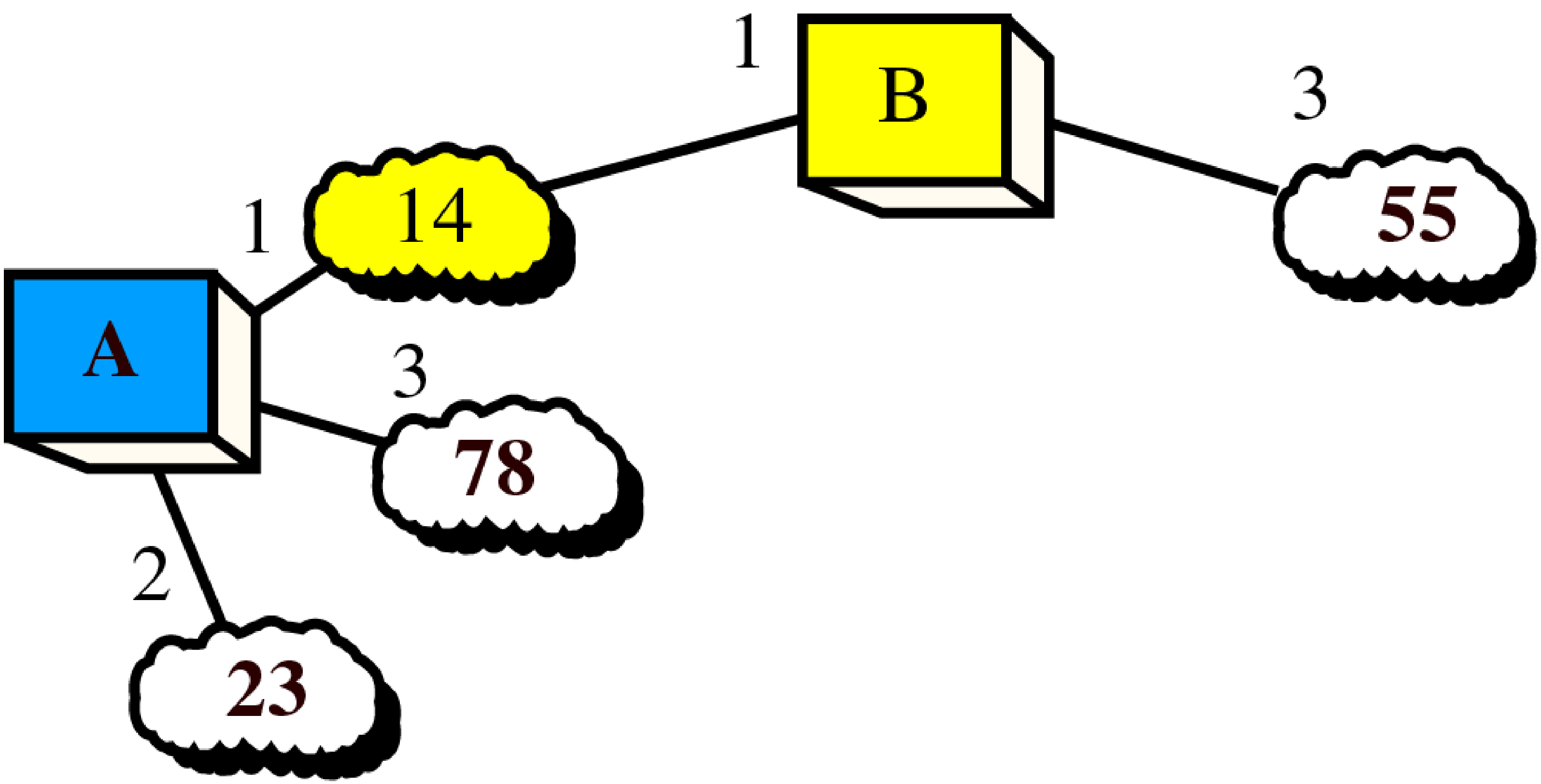
Figure 21-30, Part II

Shortest Path Calculation, Part II



14 permanent, B added

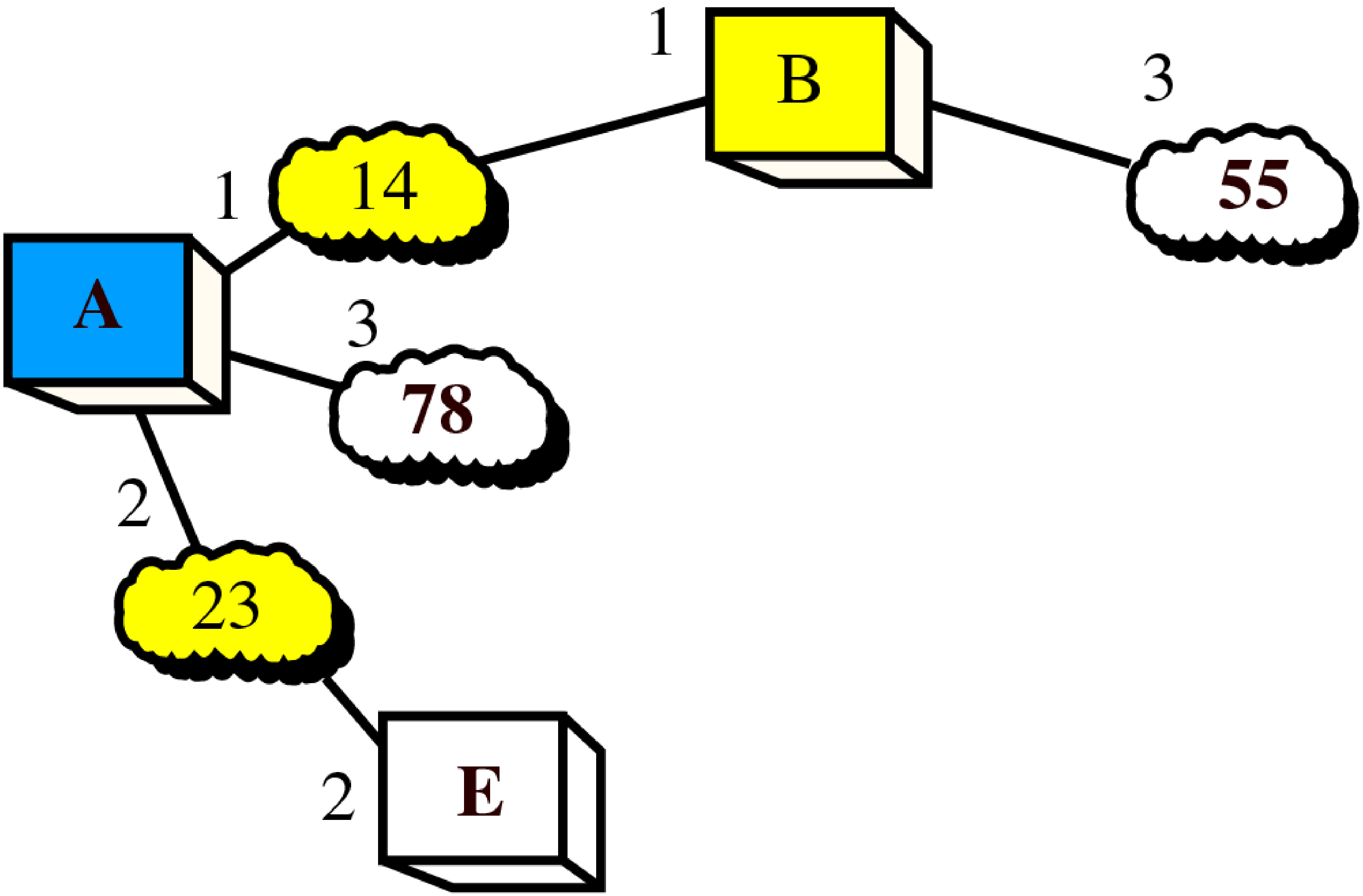
Shortest Path Calculation, Part III



B Permanent, 55 added

Figure 21-30, Part IV

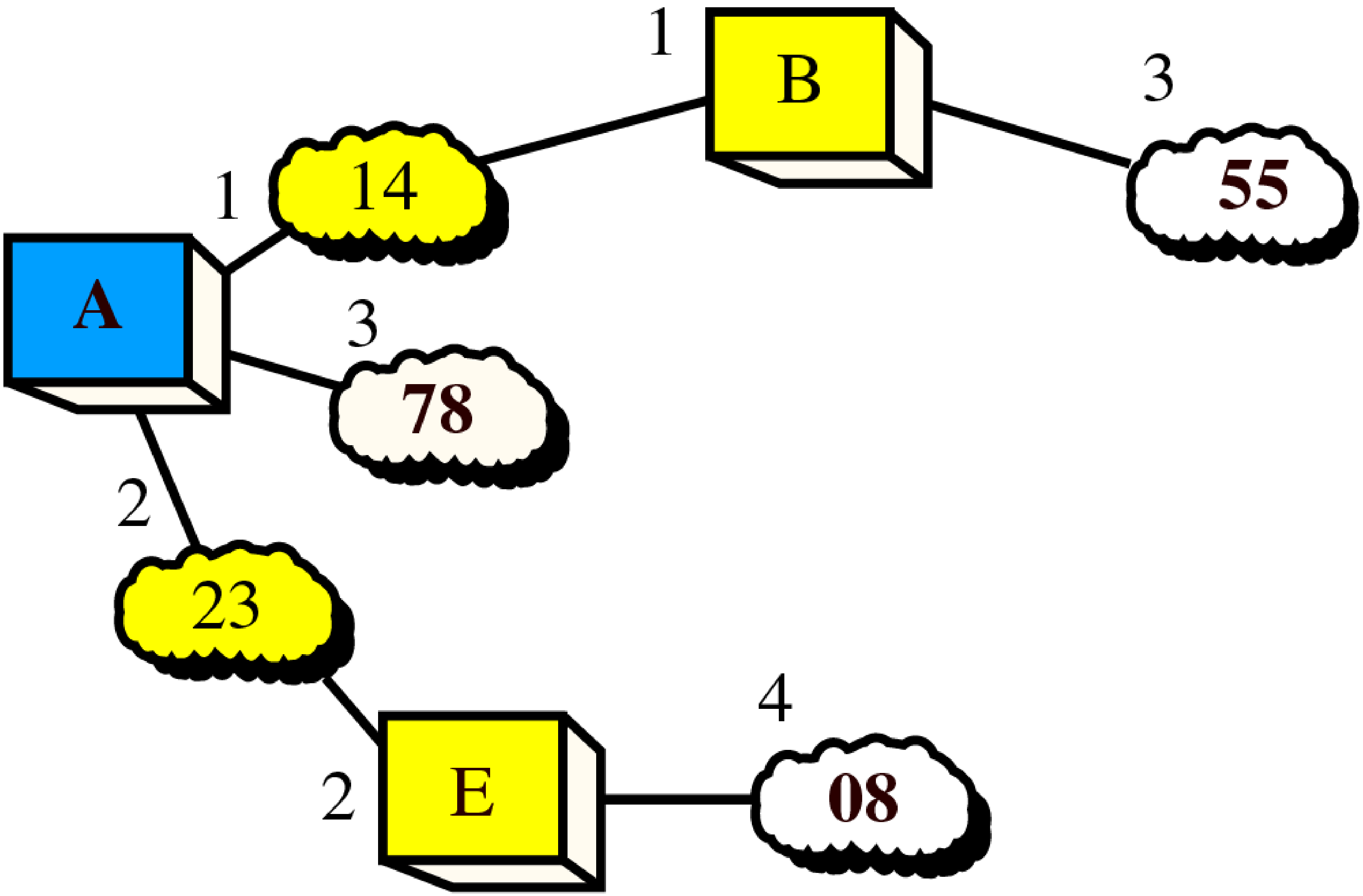
Shortest Path Calculation, Part IV



23 permanent, E added

Figure 21-30, Part V

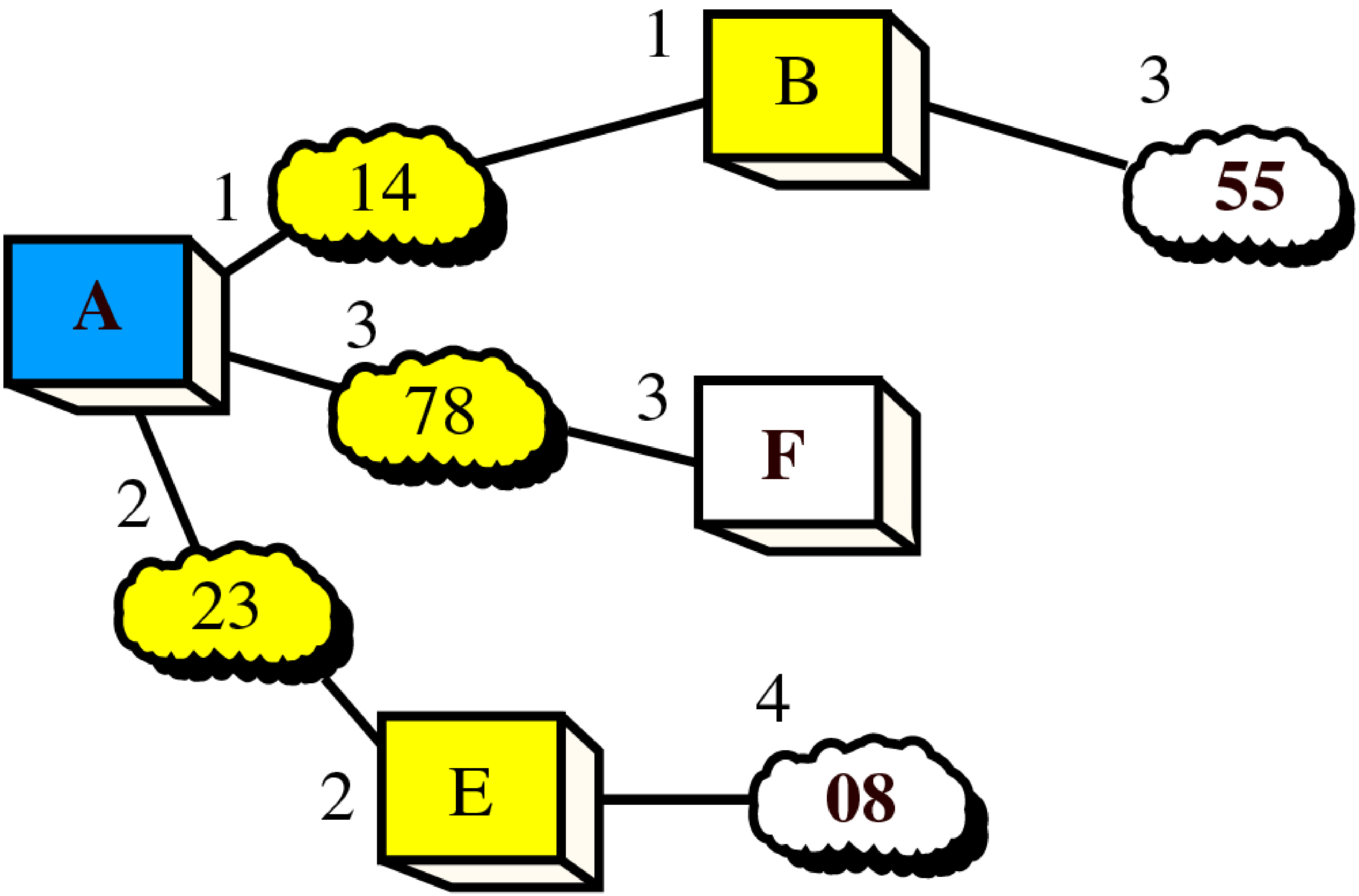
Shortest Path Calculation, Part V



E permanent, 08 added

Figure 21-30, Part VI

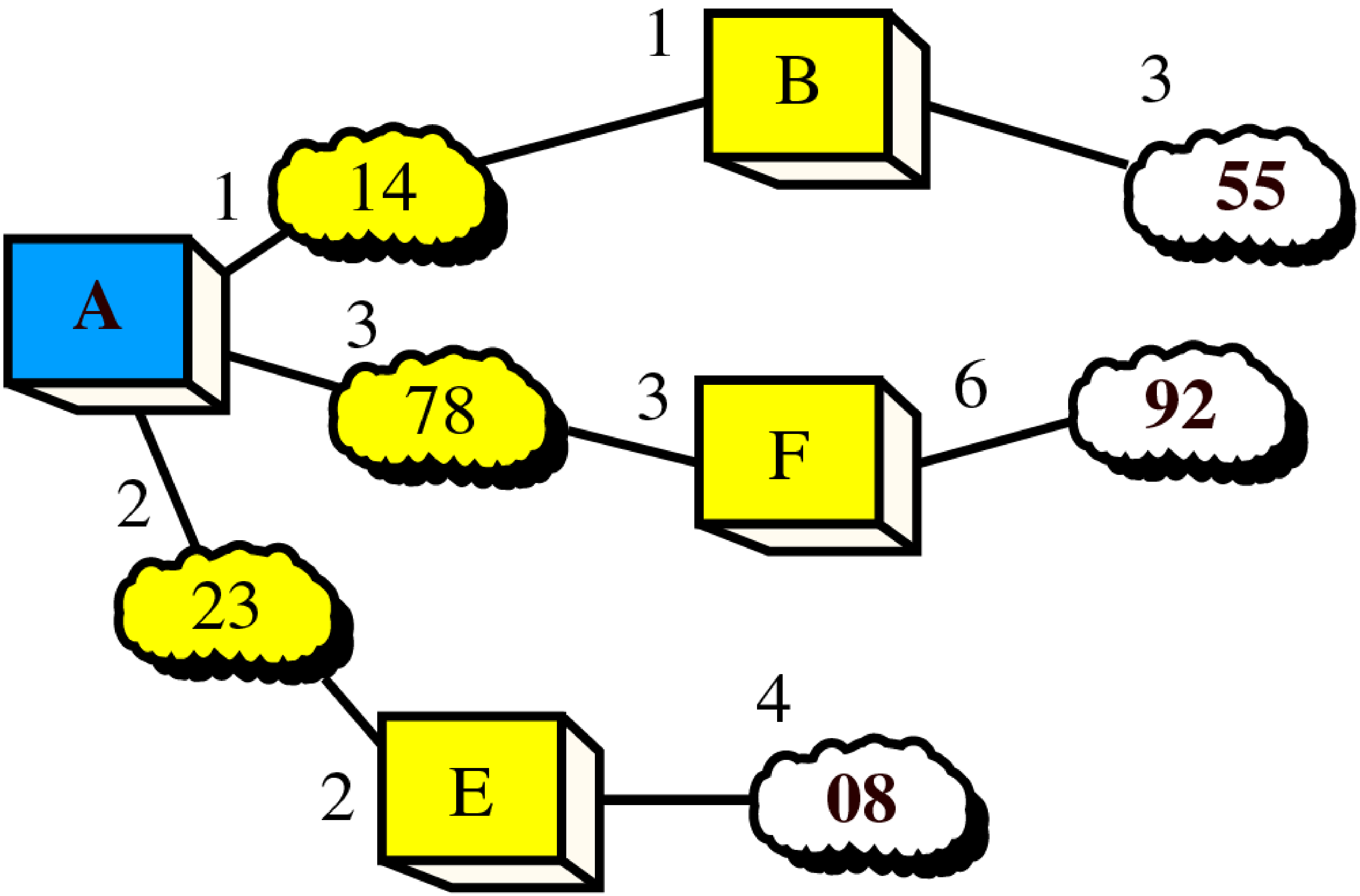
Shortest Path Calculation, Part VI



78 permanent, F added

Figure 21-31, Part VII

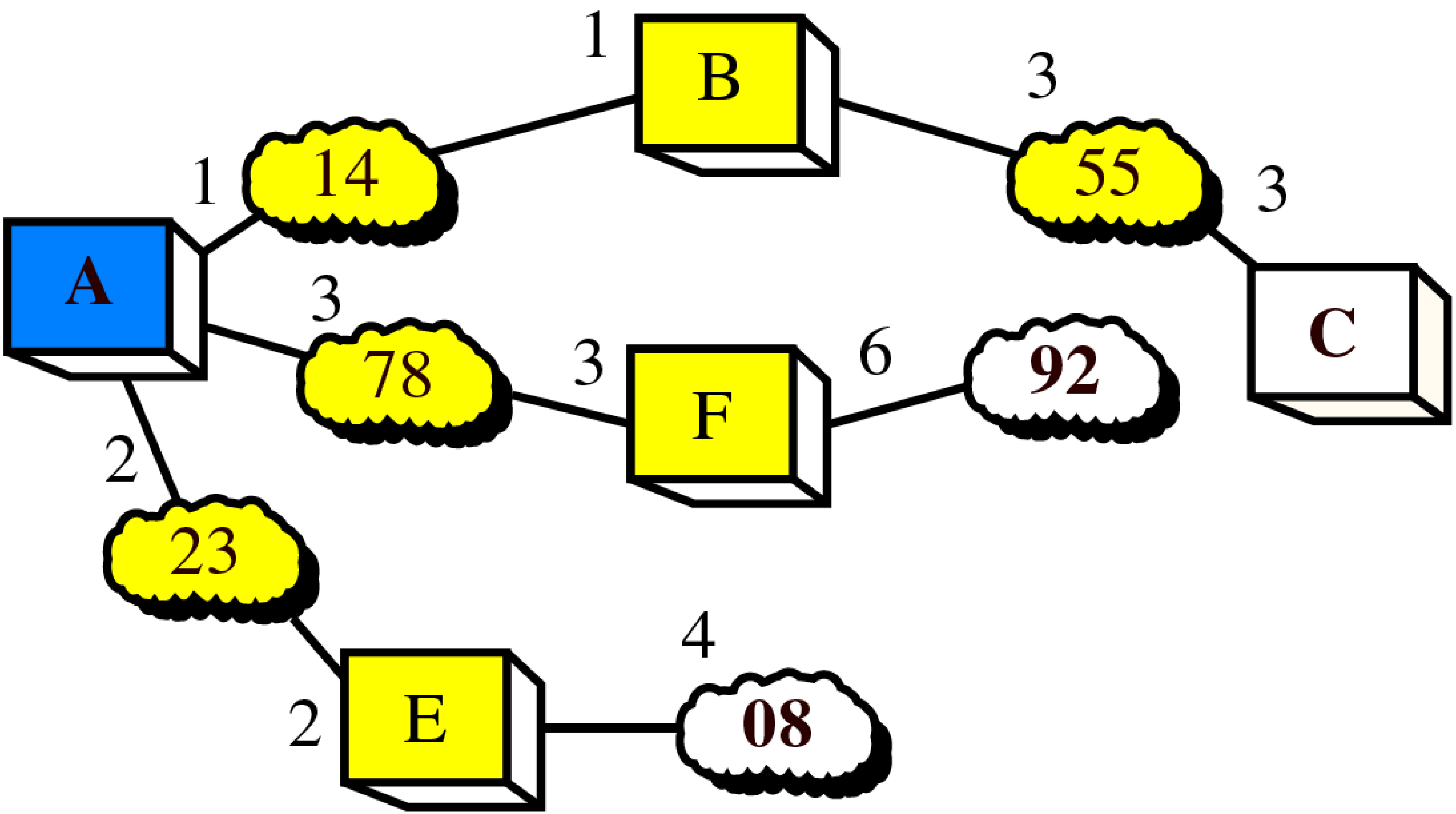
Shortest Path Calculation, Part VII



F permanent, 92 added

Figure 21-31, Part I

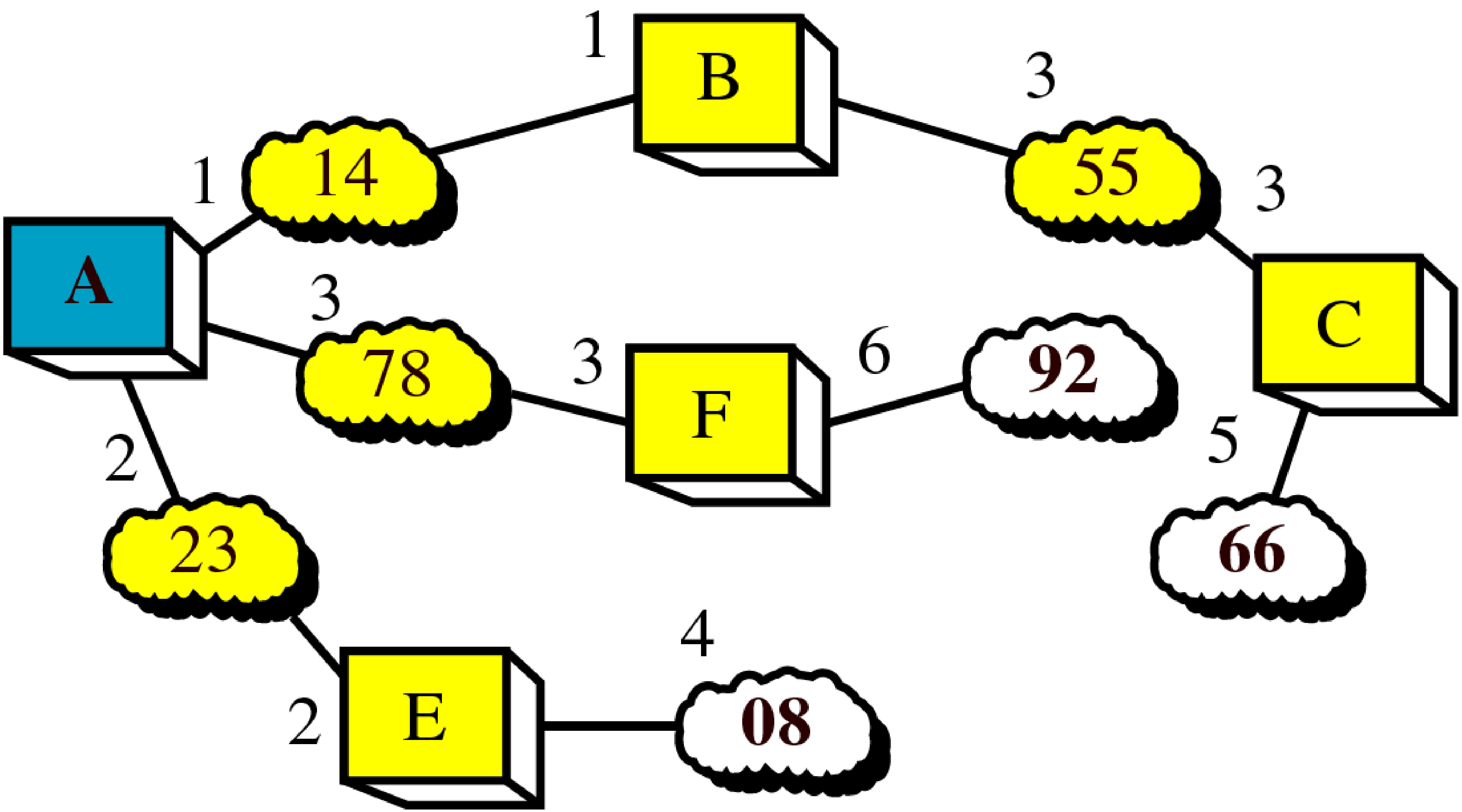
Shortest Path Calculation, Part VIII



55 permanent, C added

Figure 21-31, Part II

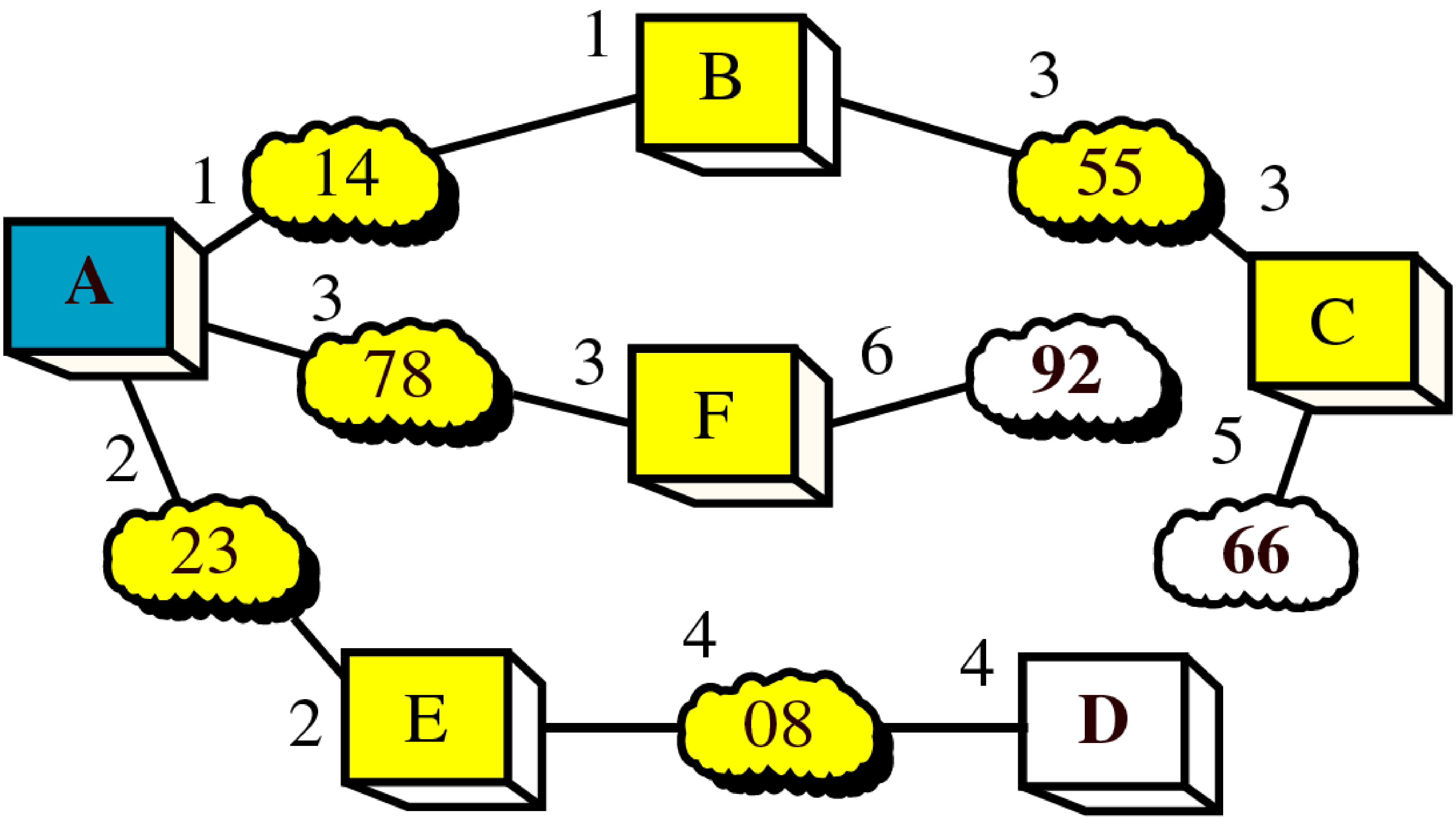
Shortest Path Calculation, Part IX



C permanent, 66 added

Figure 21-31, Part III

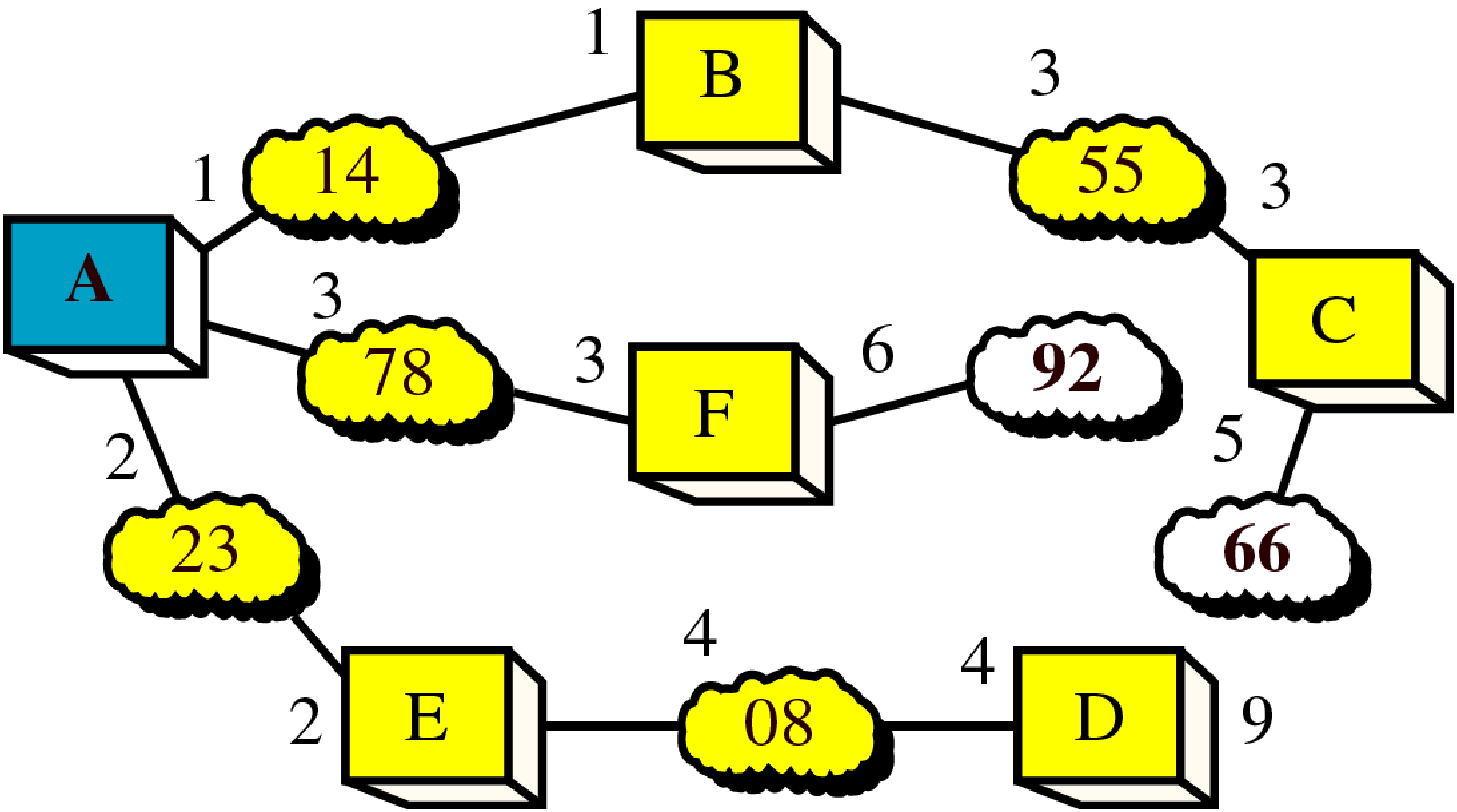
Shortest Path Calculation, Part X



08 permanent, D added

Figure 21-31, Part IV

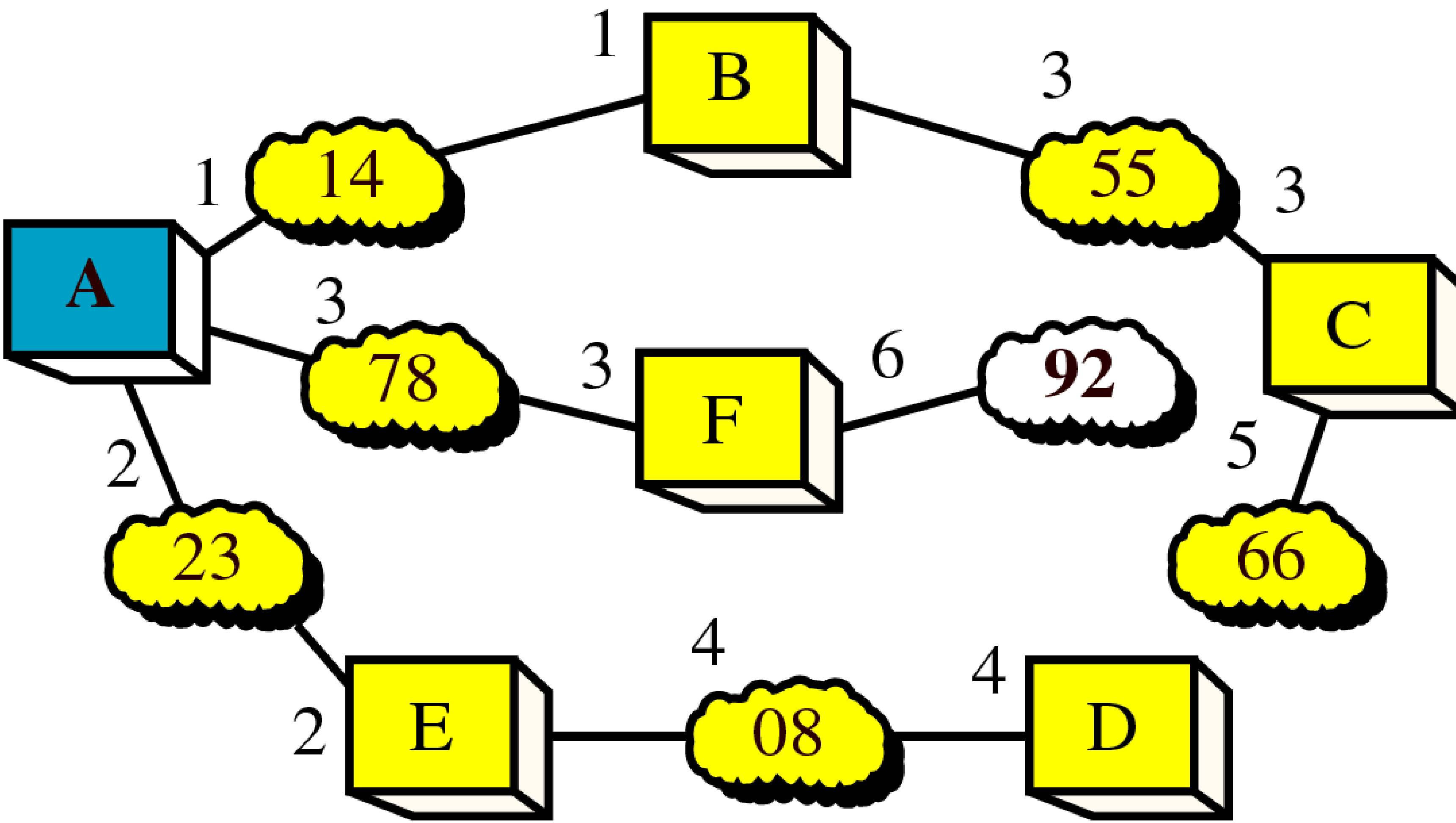
Shortest Path Calculation, Part XI



D permanent, 66 added.
But 9 > 5, so that link deleted

Figure 21-31, Part V

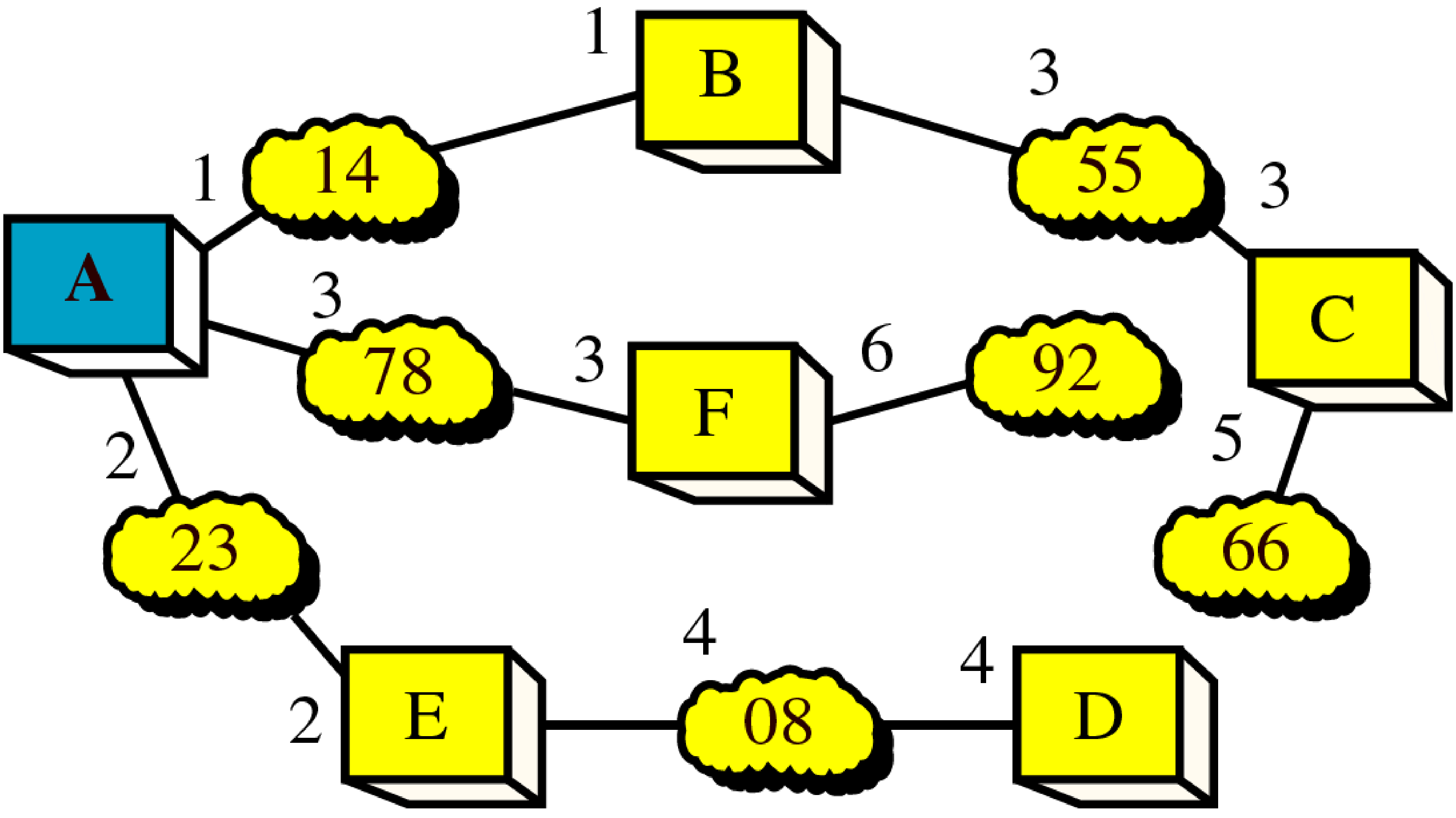
Shortest Path Calculation, Part XII



66 permanent

Figure 21-31, Part VI

Shortest Path Calculation, Part XIII



92 permanent

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