Chapter 22

Network Layer: Delivery, Forwarding, and Routing

- Delivery refers to the way a packet is handled by the underlying networks under the control of the network layer.
- Forwarding refers to the way a packet is delivered to the next station.
- Routing refers to the way routing tables are created to help in forwarding. Routing protocols are used to continuously update the routing tables that are consulted for forwarding and routing.

22-1 DELIVERY

The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet.

Topics discussed in this section:

Direct Versus Indirect Delivery

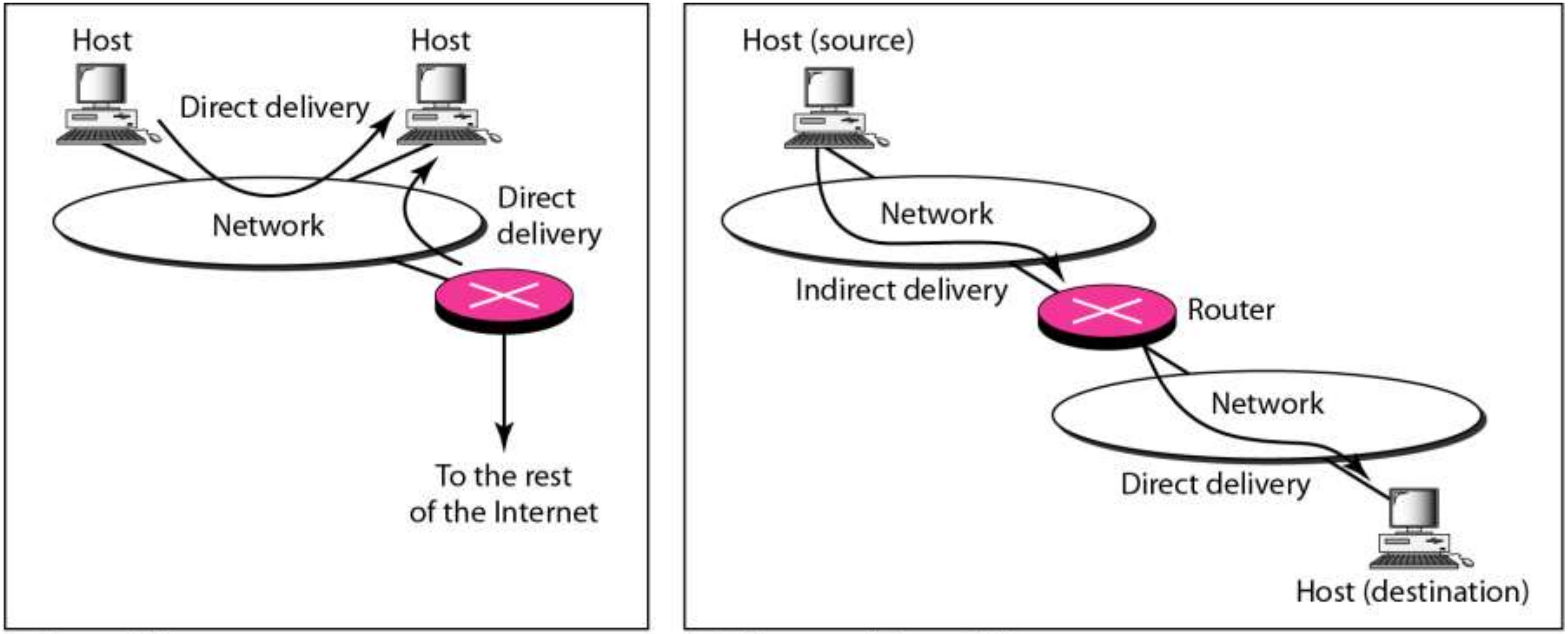
Direct Delivery

- Host-to-host
- Router-to- host
- In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer.
- Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host.
- The sender can easily determine if the delivery is direct. It can extract the network address of the destination (using the mask) and compare this address with the addresses of the networks to which it is

Indirect Delivery

- If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination.

Figure 22.1 *Direct and indirect delivery*



a. Direct delivery

b. Indirect and direct delivery

22-2 FORWARDING

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Topics discussed in this section:

Forwarding Techniques
Forwarding Process
Routing Table

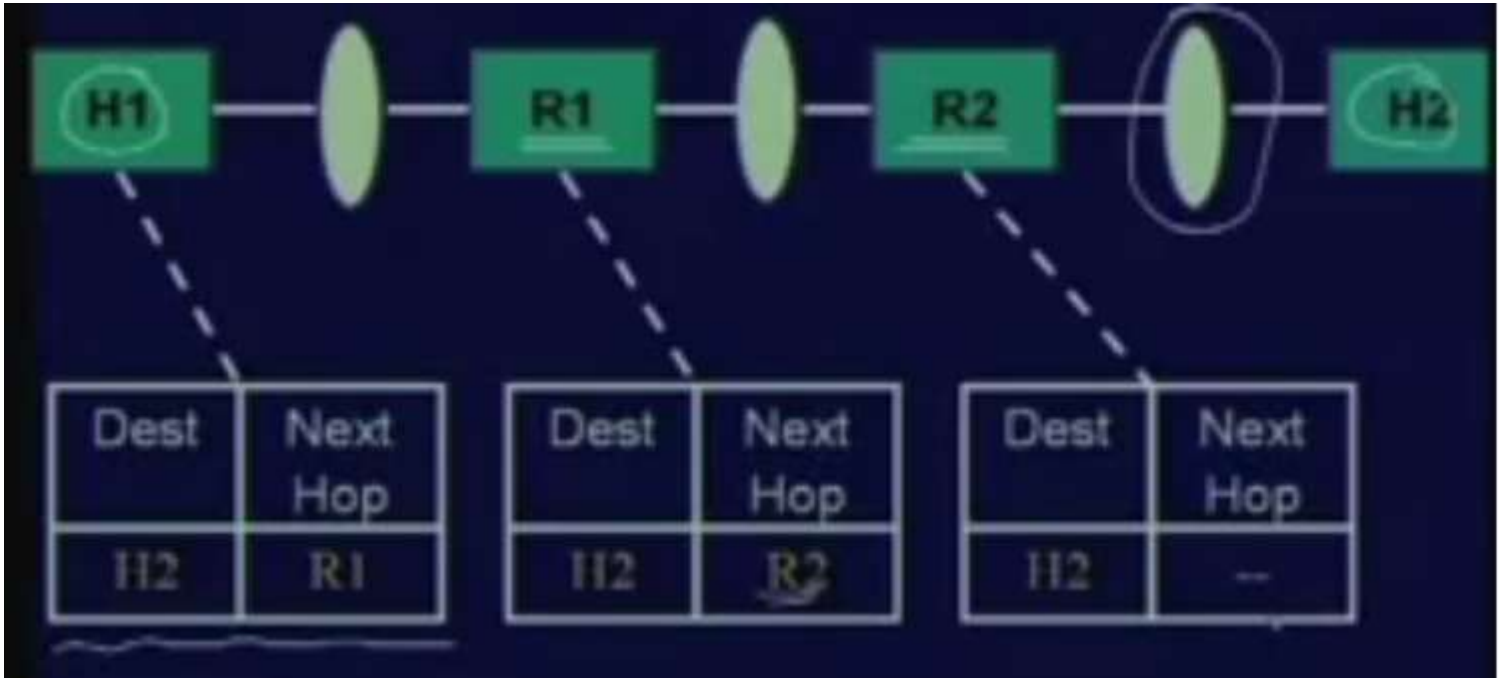
Forwarding Techniques

Several techniques can make the size of the routing table manageable and also handle issues such as security.

Next-Hop Method Versus Route Method

- One technique to reduce the contents of a routing table is called the next-hop method.
- In this technique, the routing table holds only the address of the next hop instead of information about the complete route (route method).

Routing tables based on next hop



Dest	Next Hop
H2	R1

Dest	Next Hop
H2	R2

Dest	Next Hop
H2	--

Network-Specific Method Versus Host-Specific Method

- A second technique to reduce the routing table and simplify the searching process is called the network-specific method.
- Here, instead of having an entry for every destination host connected to the same physical network (host-specific method), we have only one entry that defines the address of the destination network itself.
- In other words, we treat all hosts connected to the same network as one single entity.

Default Method

- Another technique to simplify routing is called the default method.
- Follows a default path if there is no match.
- In Fig host A is connected to a network with two routers. Router Rl routes the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 is used. So instead of listing all networks in the entire Internet, host A can just have one entry called the default (normally defined as network address 0.0.0.0).

Figure 22.2 *Route method versus next-hop method*

a. Routing tables based on route

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

b. Routing tables based on next hop

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	---

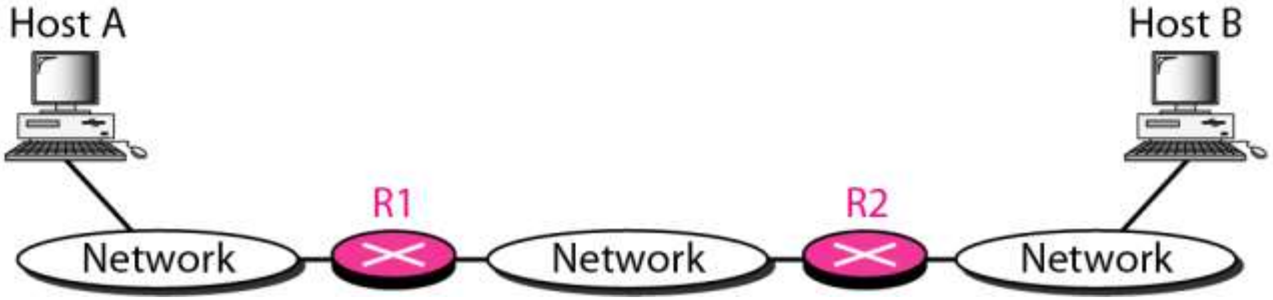
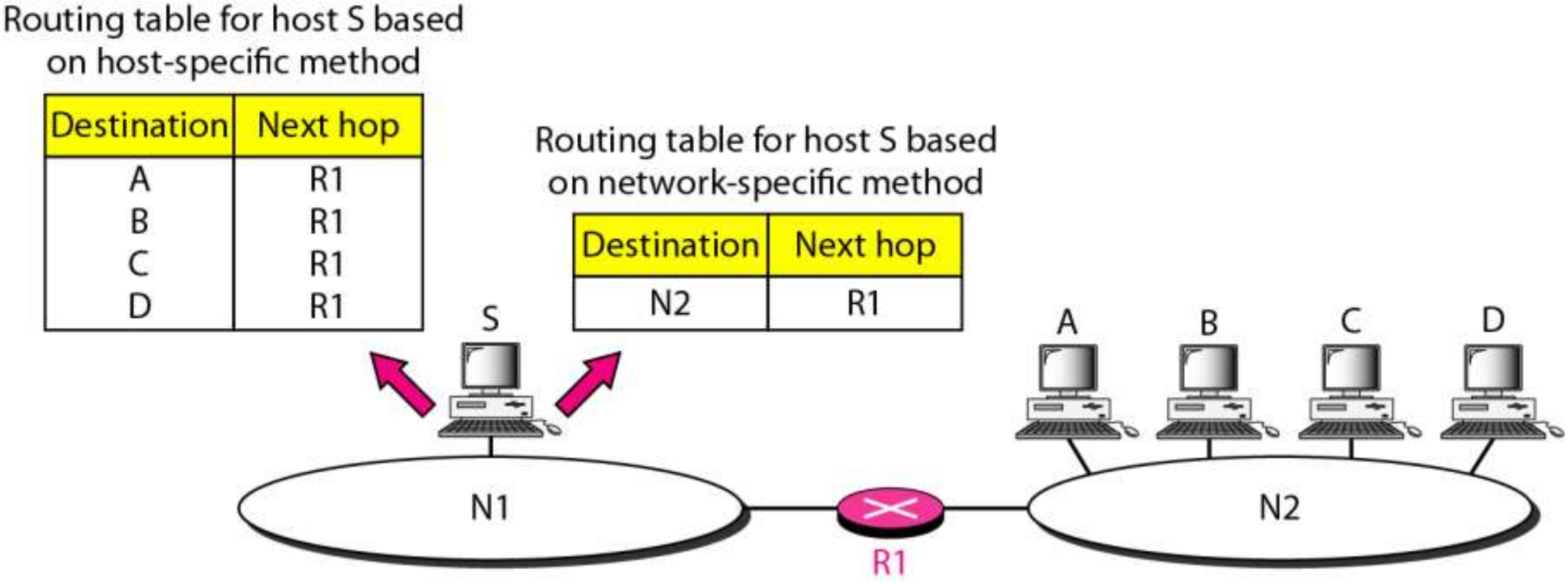


Figure 22.3 *Host-specific versus network-specific method*



Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

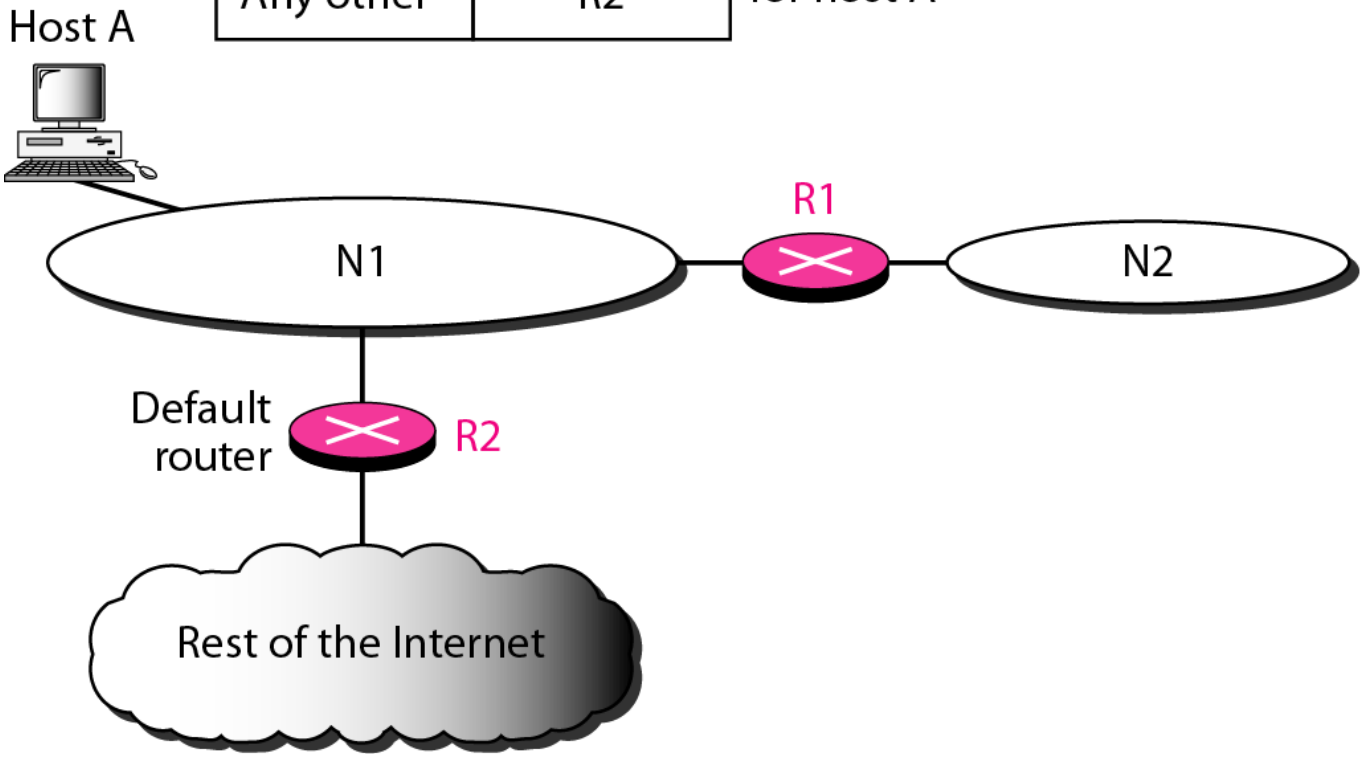
Routing table for host S based on network-specific method

Destination	Next hop
N2	R1

Figure 22.4 *Default method*

Destination	Next hop
N2	R1
Any other	R2

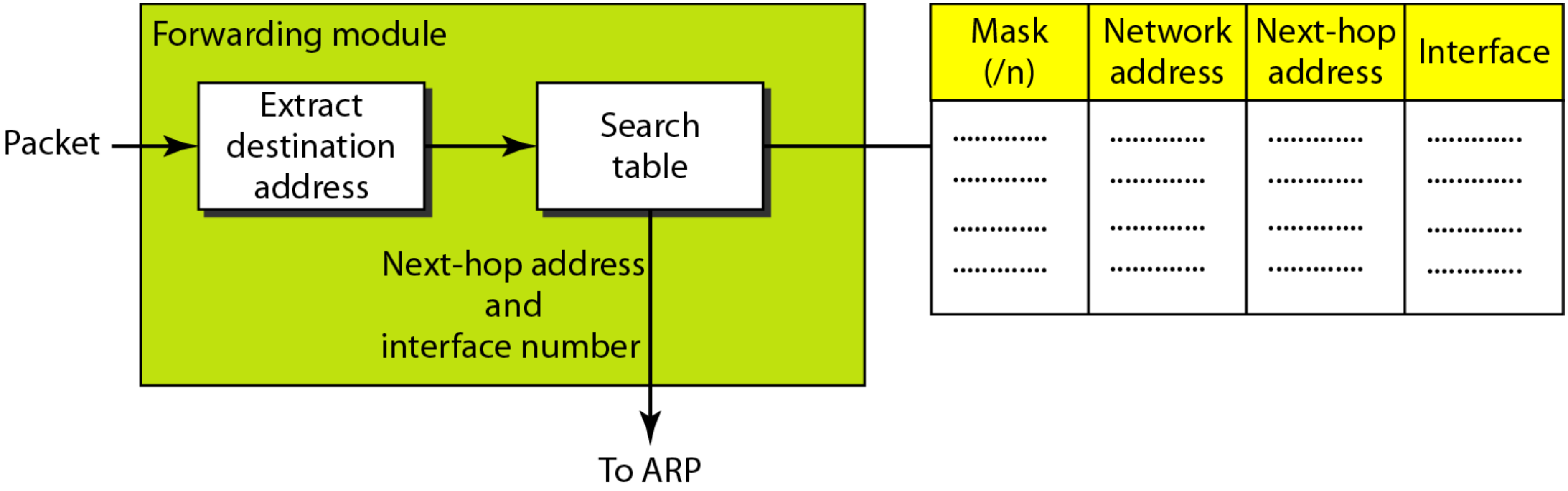
Routing table for host A



Forwarding Process

- In classless addressing, the routing table needs to have one row of information for each block involved. The table needs to be searched based on the network address (first address in the block). Unfortunately, the destination address in the packet gives no clue about the network address.
- To solve the problem, we need to include the mask (In) in the table; we need to have an extra column that includes the mask for the corresponding block.
- Figure shows a simple forwarding module for classless addressing.

Figure 22.5 *Simplified forwarding module in classless address*



Forwarding module

Packet

Extract destination address

Search table

Next-hop address and interface number

To ARP

Mask (/n)	Network address	Next-hop address	Interface
....	....	....	....
....	....	....	....
....	....	....	....
....	....	....	....

Note

In classless addressing, we need at least four columns in a routing table.

Example 22.1

Make a routing table for router R1, using the configuration in Figure 22.6.

Solution

Table 22.1 shows the corresponding table.

Figure 22.6 *Configuration for Example 22.1*

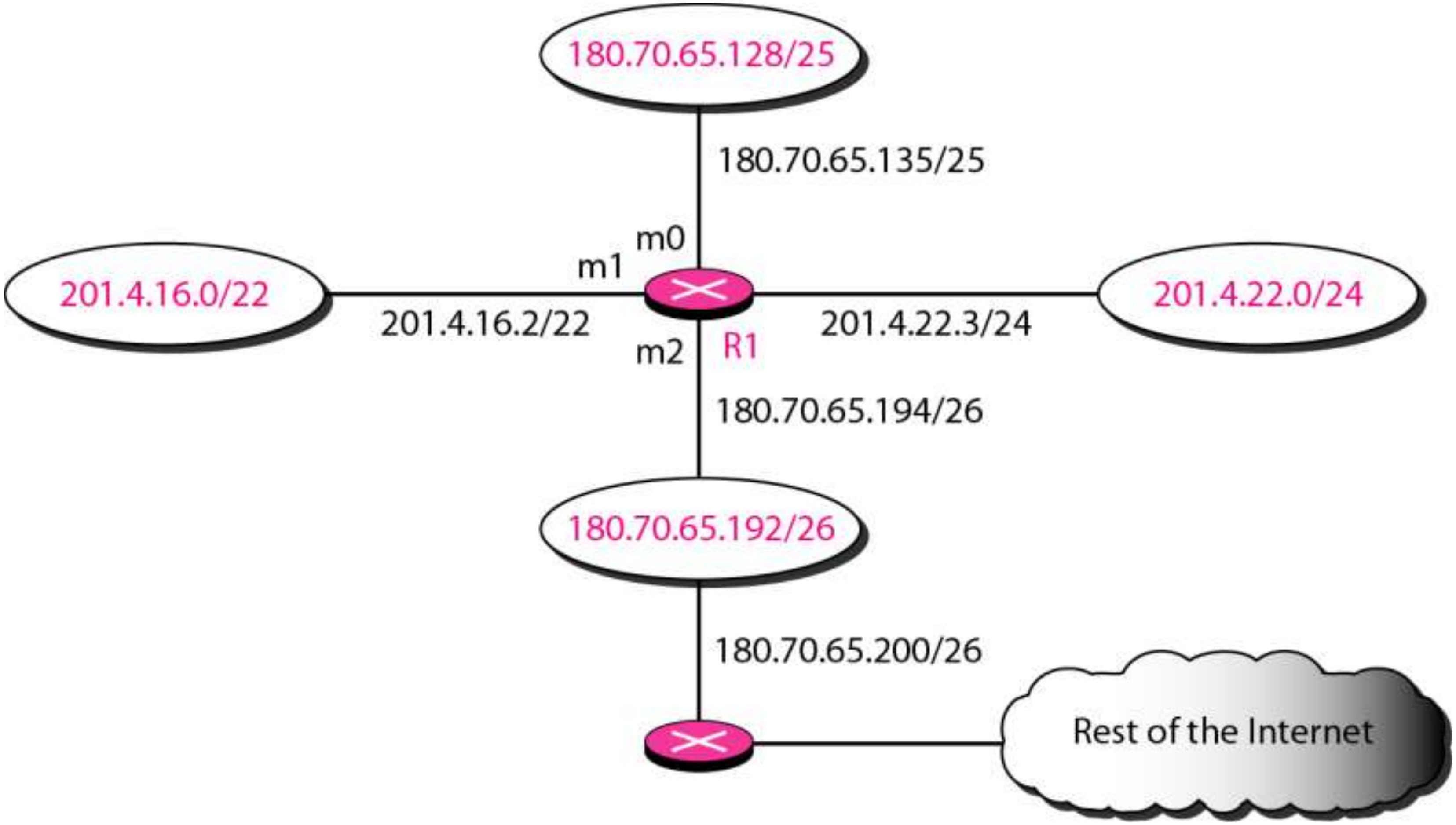
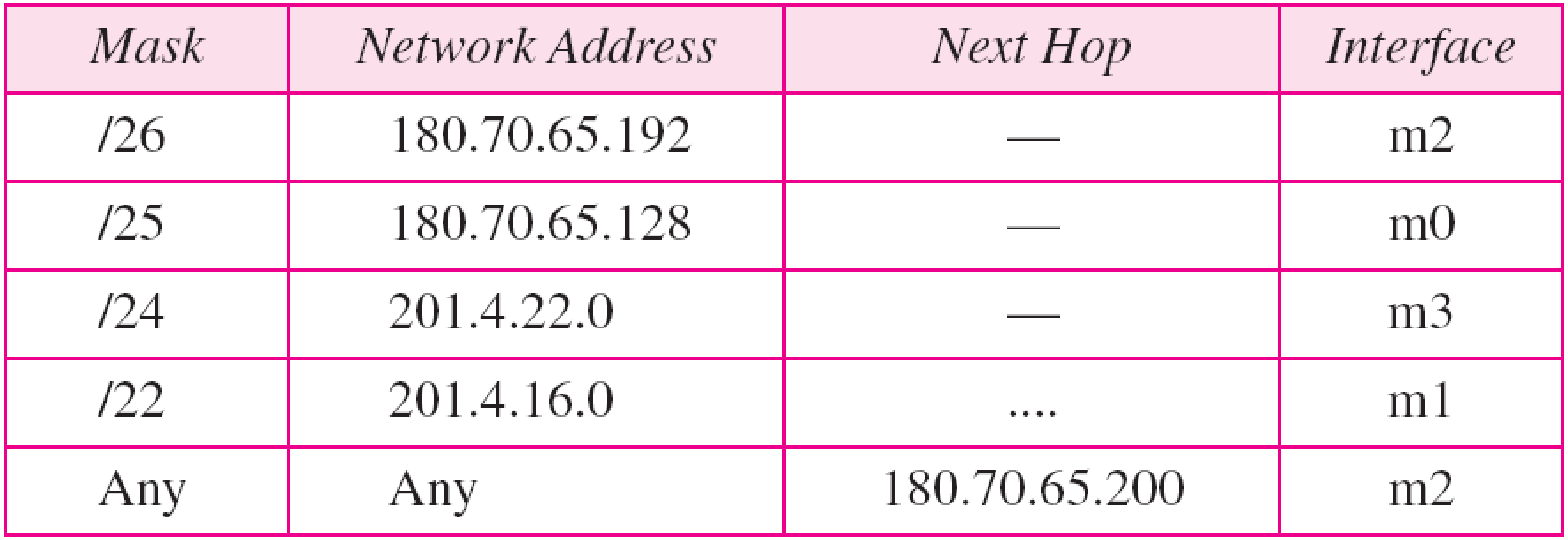


Table 22.1 *Routing table for router R1 in Figure 22.6*



Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Example 22.2

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

1. ***The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.***
2. ***The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address and the interface number m0 are passed to ARP for further processing.***

Example 22.3

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address.

2. The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).

Example 22.3 (continued)

***3.** The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.*

Example 22.4

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 18.24.32.78.

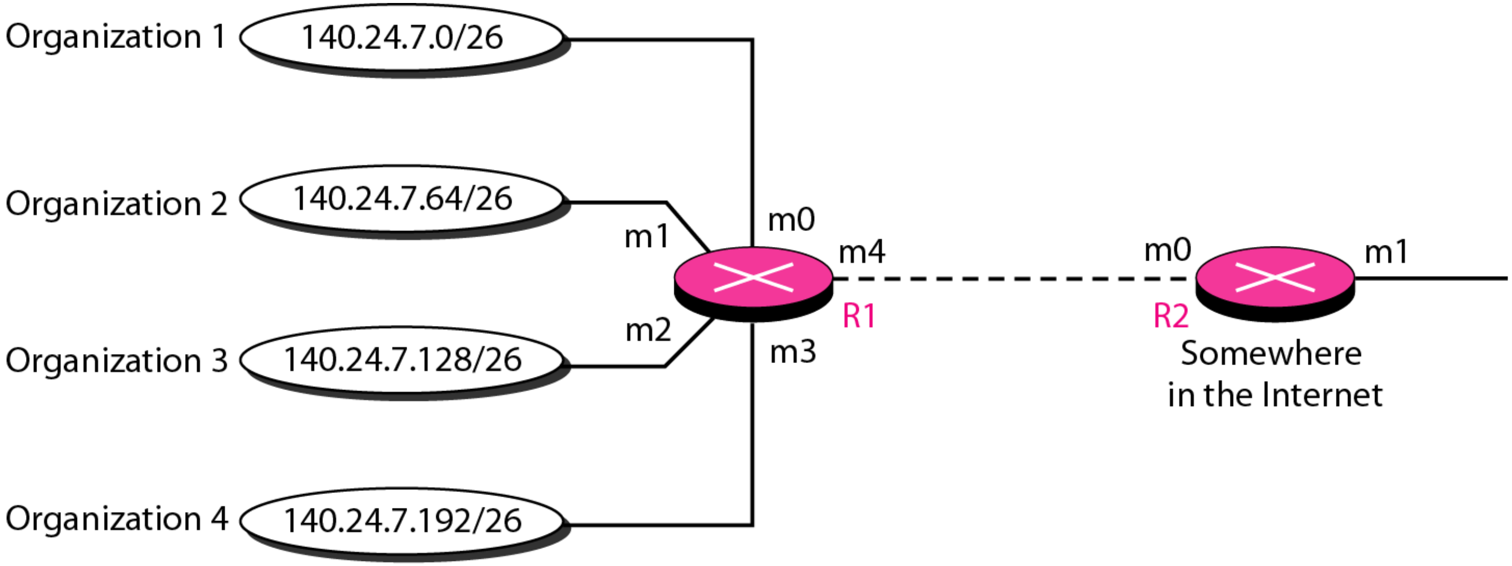
Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Address Aggregation:

- When we use classless addressing, it is likely that the number of routing table entries will increase. This is so because the intent of classless addressing is to divide up the whole address space into manageable blocks. The increased size of the table results in an increase in the amount of time needed to search the table. To alleviate the problem, the idea of address aggregation was designed.

Figure 22.7 *Address aggregation*



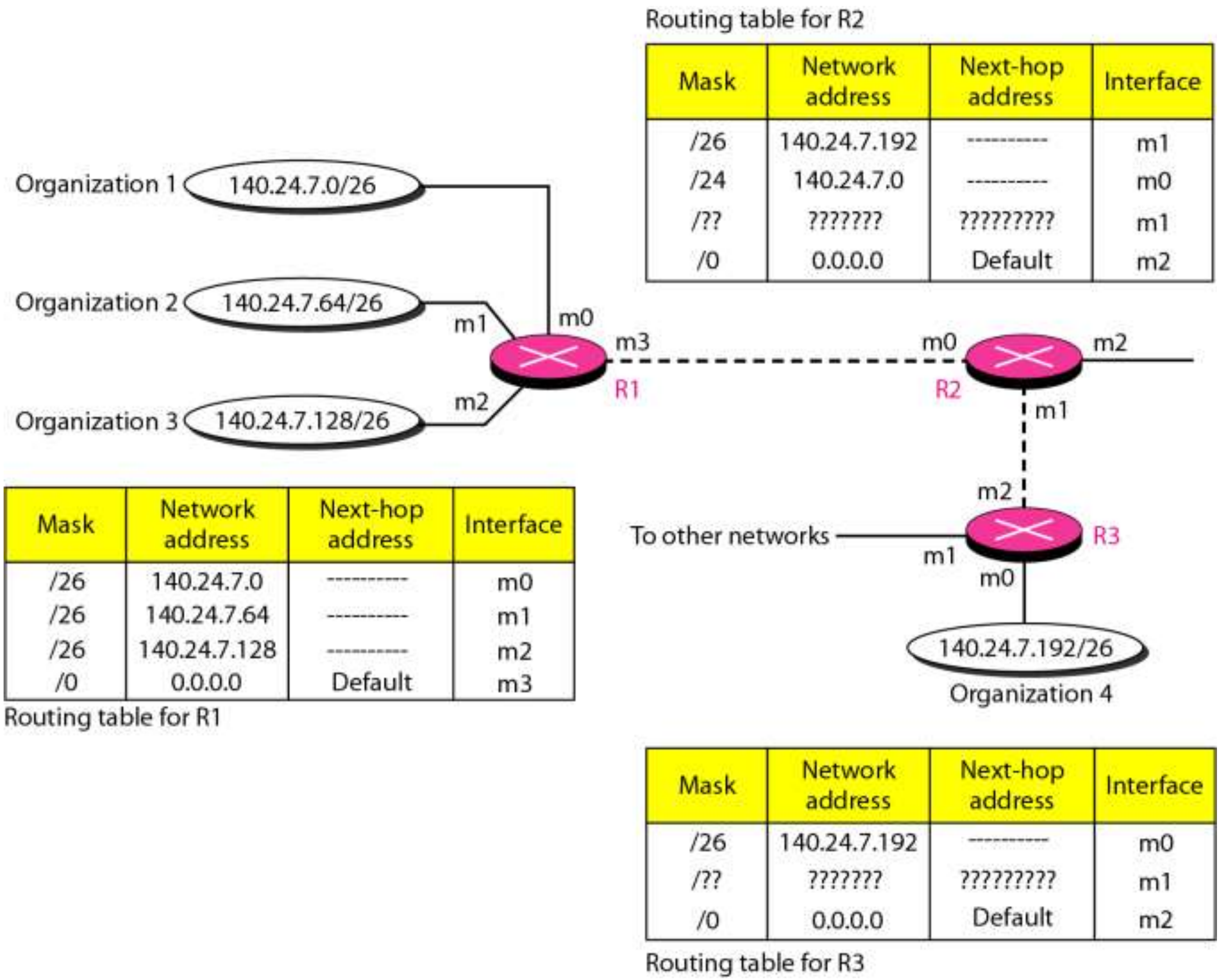
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Default	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	Default	m1

Routing table for R2

Figure 22.8 *Longest mask matching*



Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m1
/24	140.24.7.0	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Default	m3

Routing table for R1

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	Default	m2

Routing table for R3

Example 22.5

As an example of hierarchical routing, let us consider Figure 22.9. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into four subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs; the second subblock is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.

The first local ISP has divided its assigned subblock into 8 smaller blocks and assigned each to a small ISP. Each small ISP provides services to 128 households, each using four addresses.

Example 22.5 (continued)

The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations.

The third local ISP has divided its block into 16 blocks and assigned each block to a small organization. Each small organization has 256 addresses, and the mask is /24.

There is a sense of hierarchy in this configuration. All routers in the Internet send a packet with destination address 120.14.64.0 to 120.14.127.255 to the regional ISP.

Figure 22.9 *Hierarchical routing with ISPs*

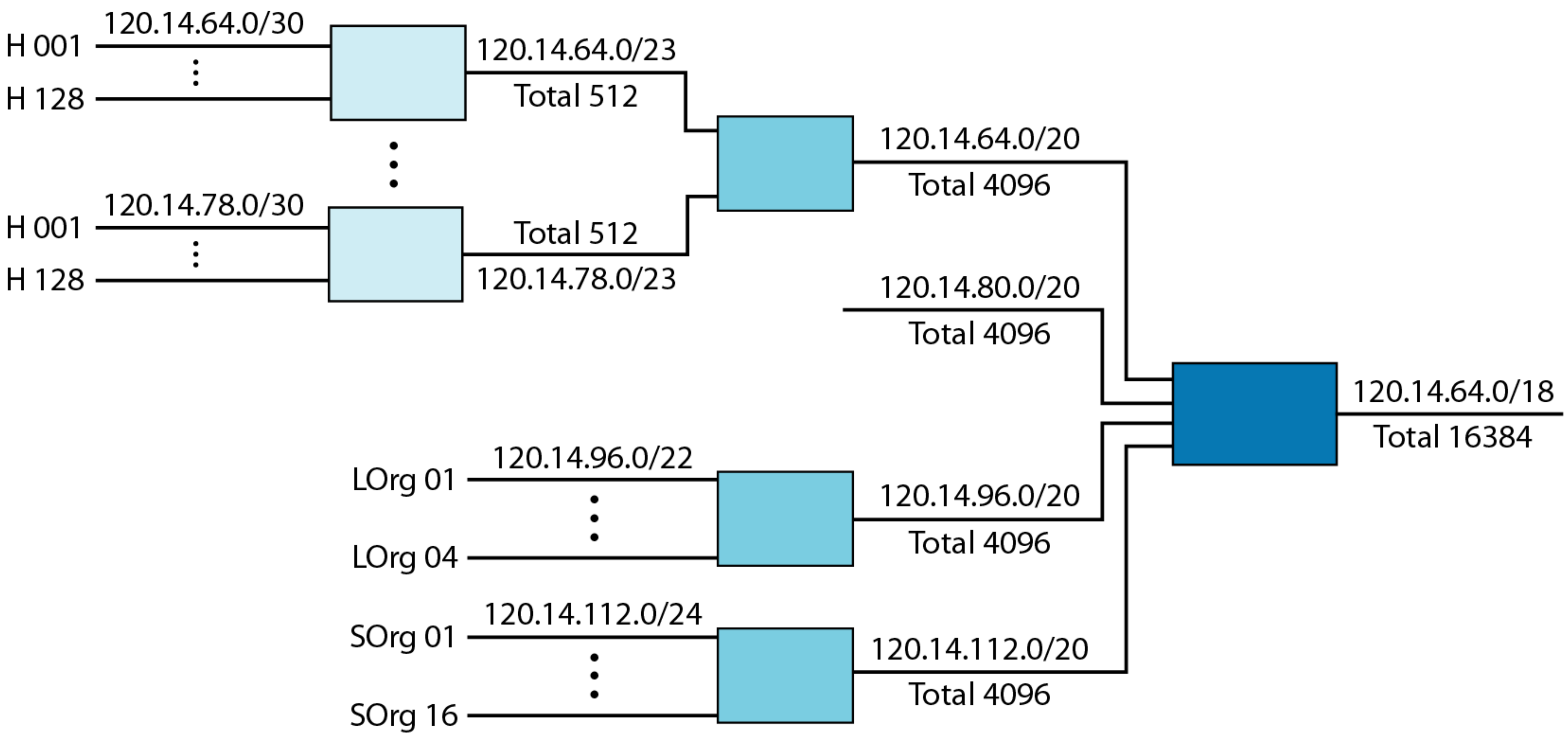
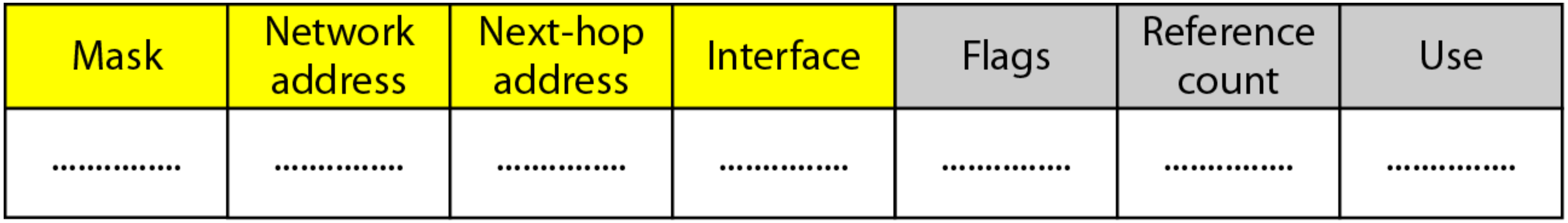


Figure 22.10 *Common fields in a routing table*

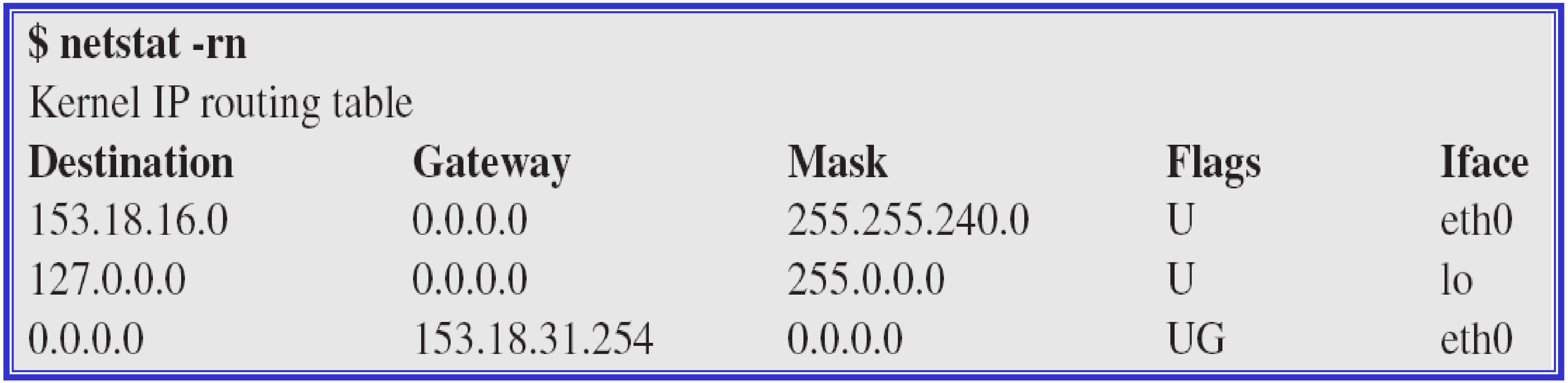


Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
....	....	....	....	....	....	....

Example 22.6

One utility that can be used to find the contents of a routing table for a host or router is netstat in UNIX or LINUX. The next slide shows the list of the contents of a default server. We have used two options, r and n. The option r indicates that we are interested in the routing table, and the option n indicates that we are looking for numeric addresses. Note that this is a routing table for a host, not a router. Although we discussed the routing table for a router throughout the chapter, a host also needs a routing table.

Example 22.6 (continued)

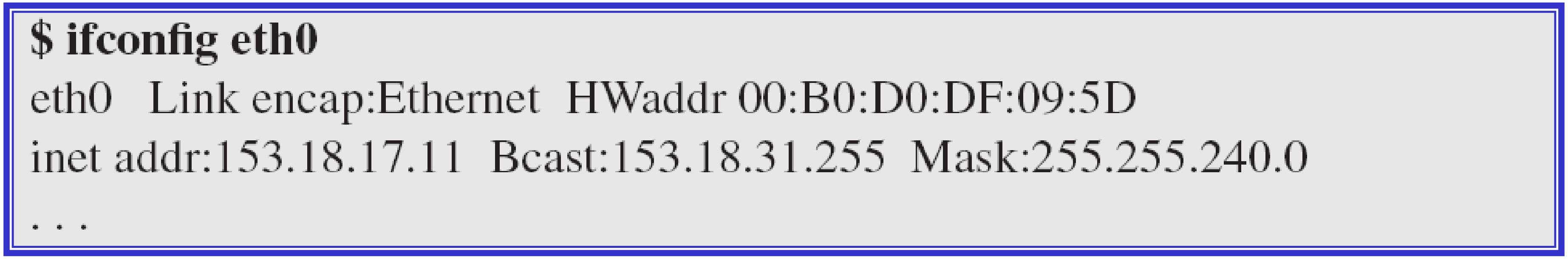


```
$ netstat -rn
Kernel IP routing table
Destination      Gateway          Mask             Flags      Iface
153.18.16.0      0.0.0.0          255.255.240.0    U          eth0
127.0.0.0        0.0.0.0          255.0.0.0        U          lo
0.0.0.0          153.18.31.254    0.0.0.0          UG         eth0
```

The destination column here defines the network address. The term gateway used by UNIX is synonymous with router. This column actually defines the address of the next hop. The value 0.0.0.0 shows that the delivery is direct. The last entry has a flag of G, which means that the destination can be reached through a router (default router). The Iface defines the interface.

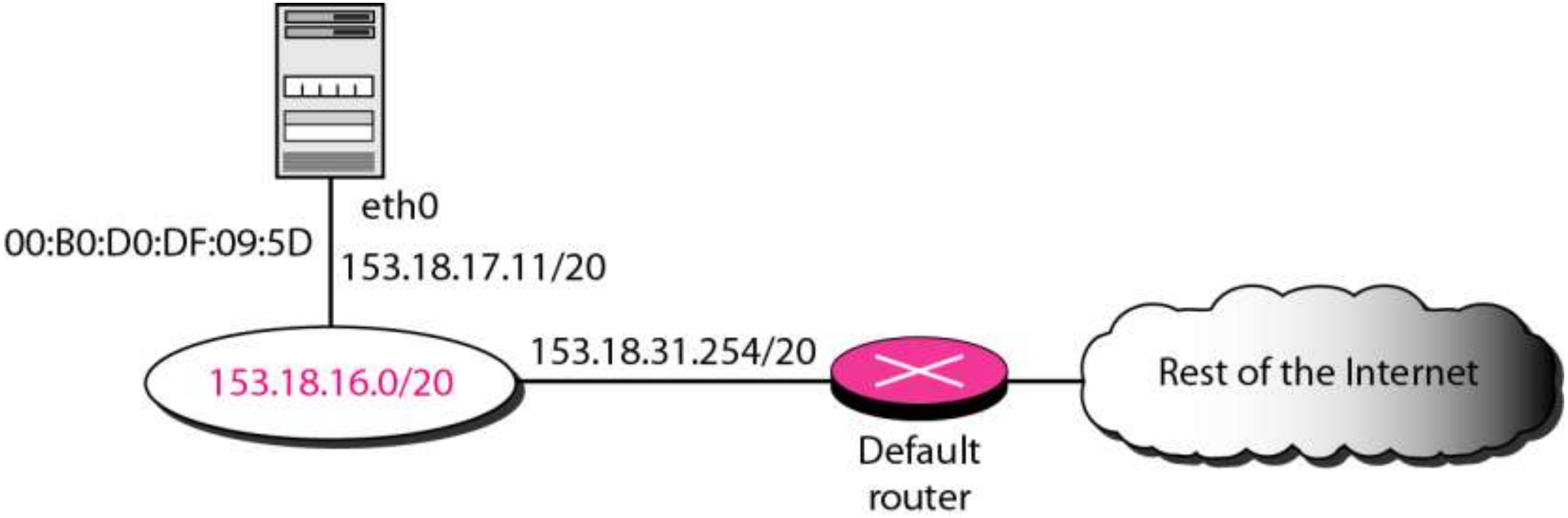
Example 22.6 (continued)

More information about the IP address and physical address of the server can be found by using the ifconfig command on the given interface (eth0).



```
$ ifconfig eth0
eth0   Link encap:Ethernet  HWaddr 00:B0:D0:DF:09:5D
inet addr:153.18.17.11  Bcast:153.18.31.255  Mask:255.255.240.0
. . .
```

Figure 22.11 *Configuration of the server for Example 22.6*



Types of Routing tables:

1.Static

- Contains information inserted manually
- Doesn’t change with time.

2.Dynamic

- Updated periodically based on network condition
- Uses protocols like RIP,OSPF,BGP,etc.

22-3 UNICAST ROUTING PROTOCOLS

A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table is one that is updated automatically when there is a change somewhere in the Internet. A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

Topics discussed in this section:

Optimization
Intra- and Interdomain Routing
Distance Vector Routing and RIP
Link State Routing and OSPF
Path Vector Routing and BGP

Inter Domain and Intra Domain routing:

- An internet is divided into autonomous systems. An autonomous system (AS) is a group of networks and routers under the authority of a single administration.
- Routing inside an autonomous system is referred to as intradomain routing.
- Routing between autonomous systems is referred to as interdomain routing. Each autonomous system can choose one or more intradomain routing protocols to handle routing inside the autonomous system.

Figure 22.12 *Autonomous systems*

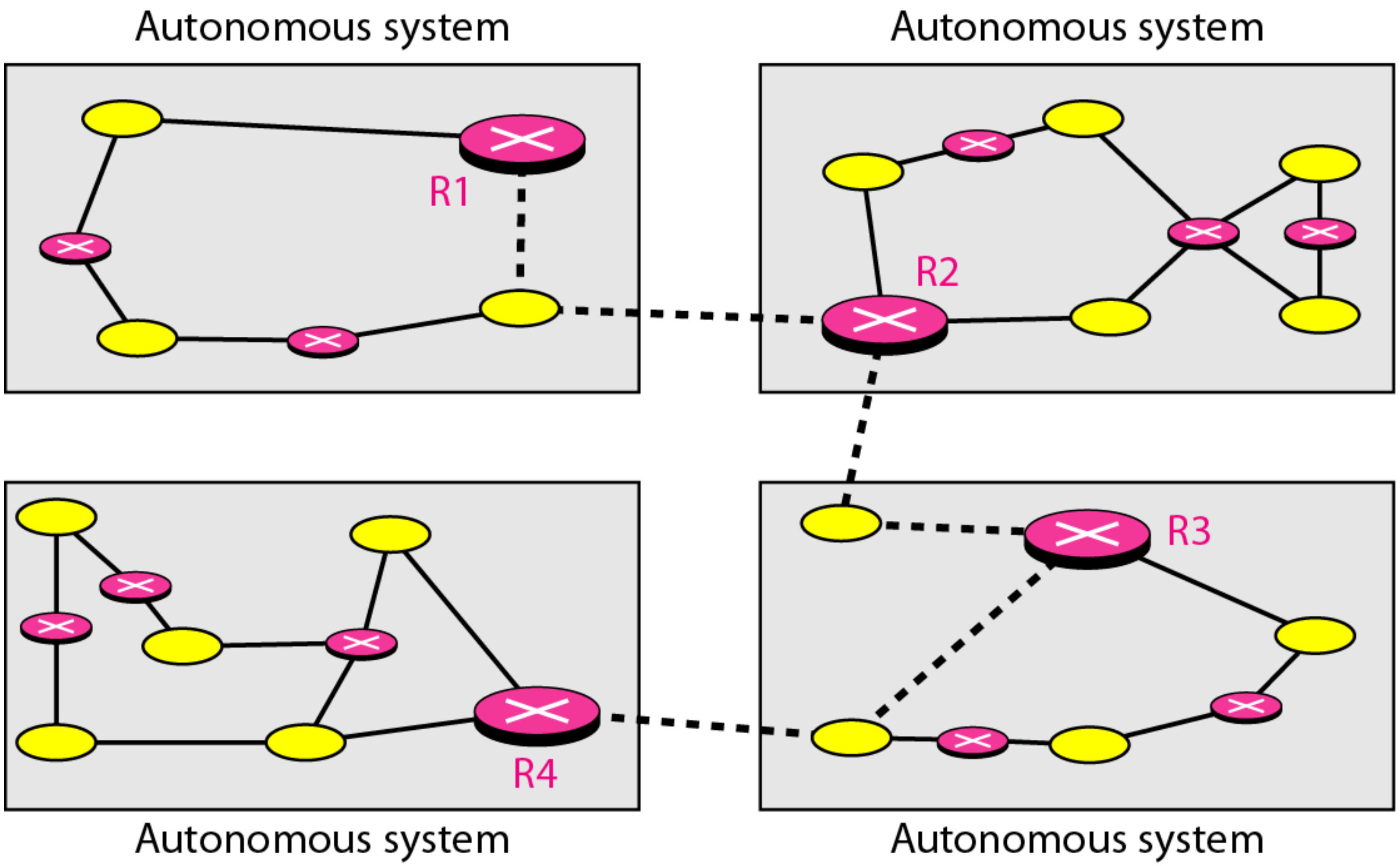
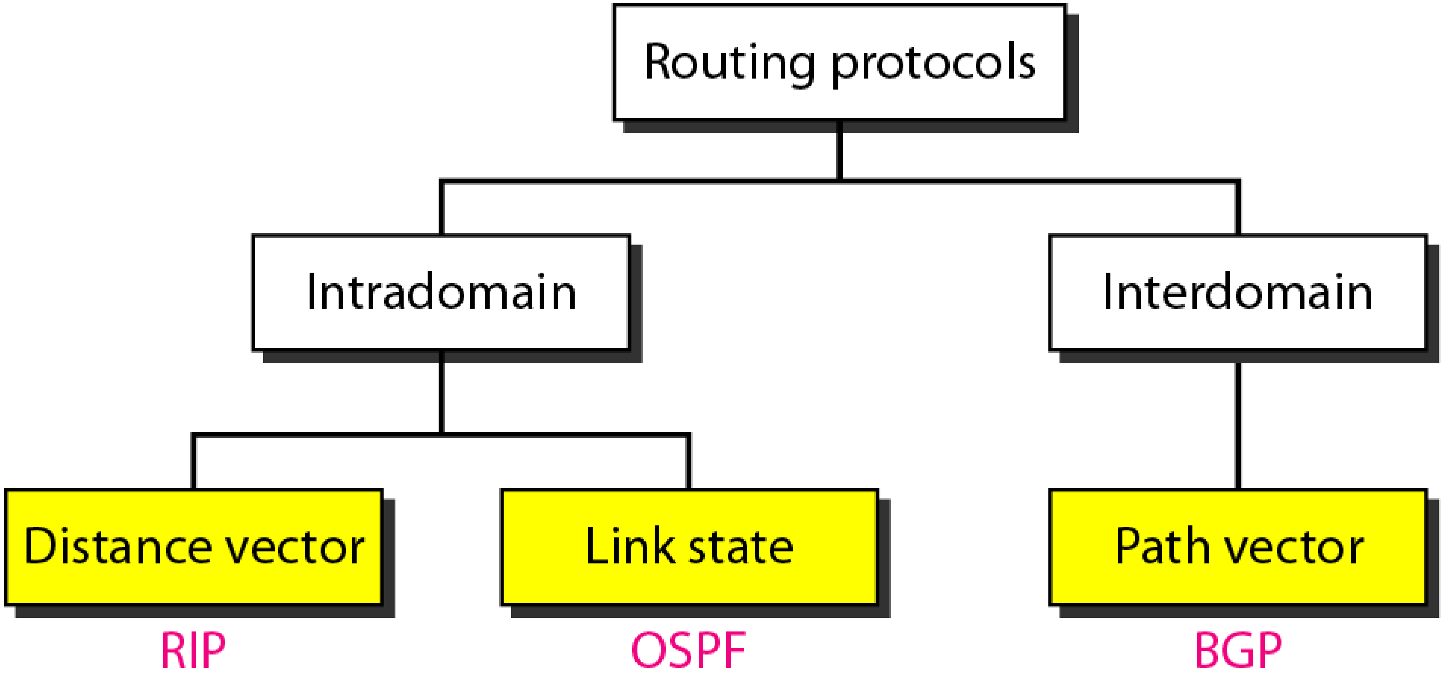


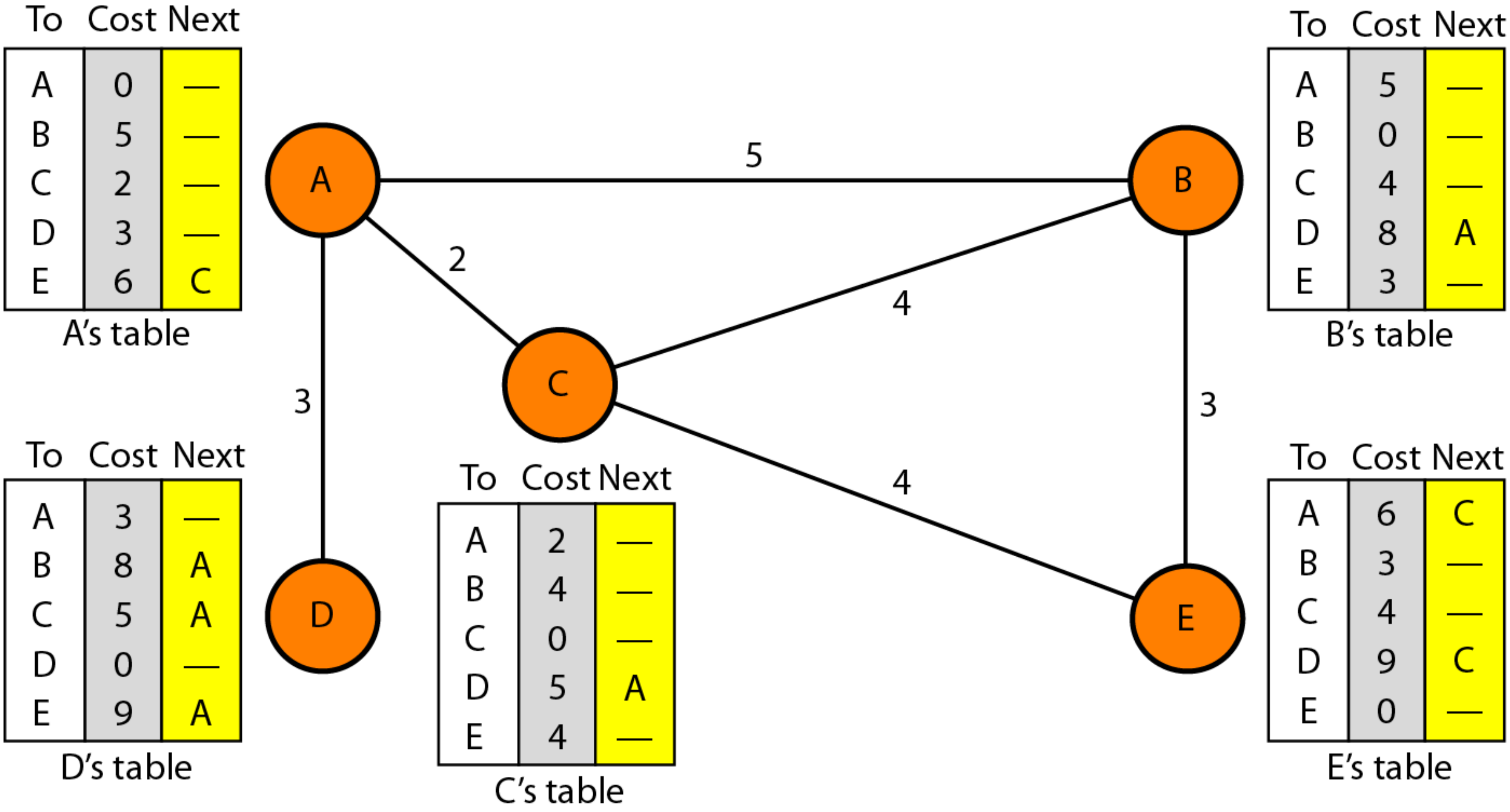
Figure 22.13 *Popular routing protocols*



Distance Vector Routing:

- In this, 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 table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

Figure 22.14 *Distance vector routing tables*



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A’s table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B’s table

To	Cost	Next
A	3	—
B	8	A
C	5	A
D	0	—
E	9	A

D’s table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C’s table

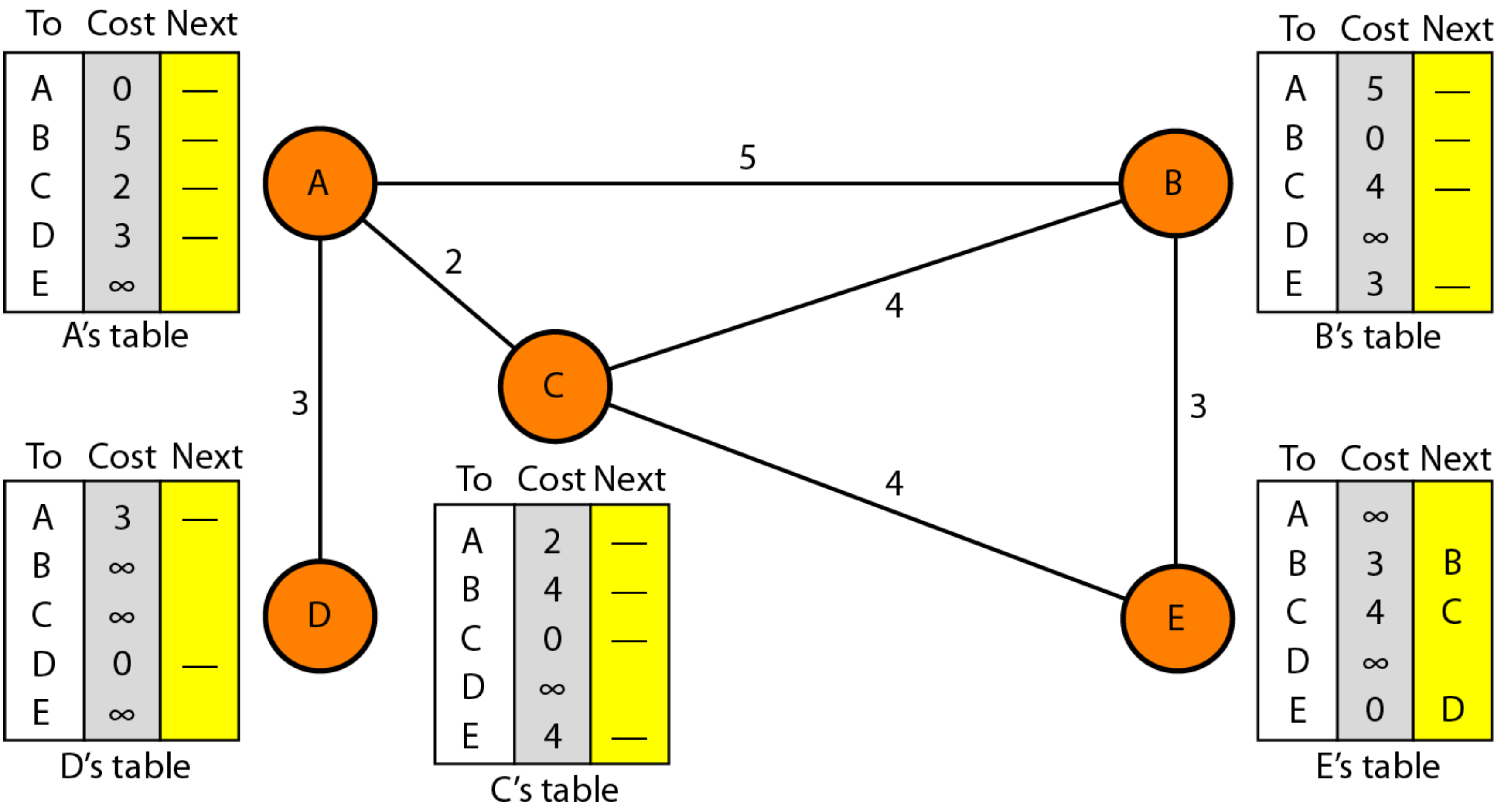
To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

E’s table

Initialization

- At the beginning, each node can know only the distance between itself and its immediate neighbors, those directly connected to it.
- So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.
- The distance for any entry that is not a neighbor is marked as infinite (unreachable).

Figure 22.15 *Initialization of tables in distance vector routing*



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	∞	
E	3	—

B's table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

D's table

To	Cost	Next
A	2	—
B	4	—
C	0	—
D	∞	
E	4	—

C's table

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

E's table

Note

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

Updating

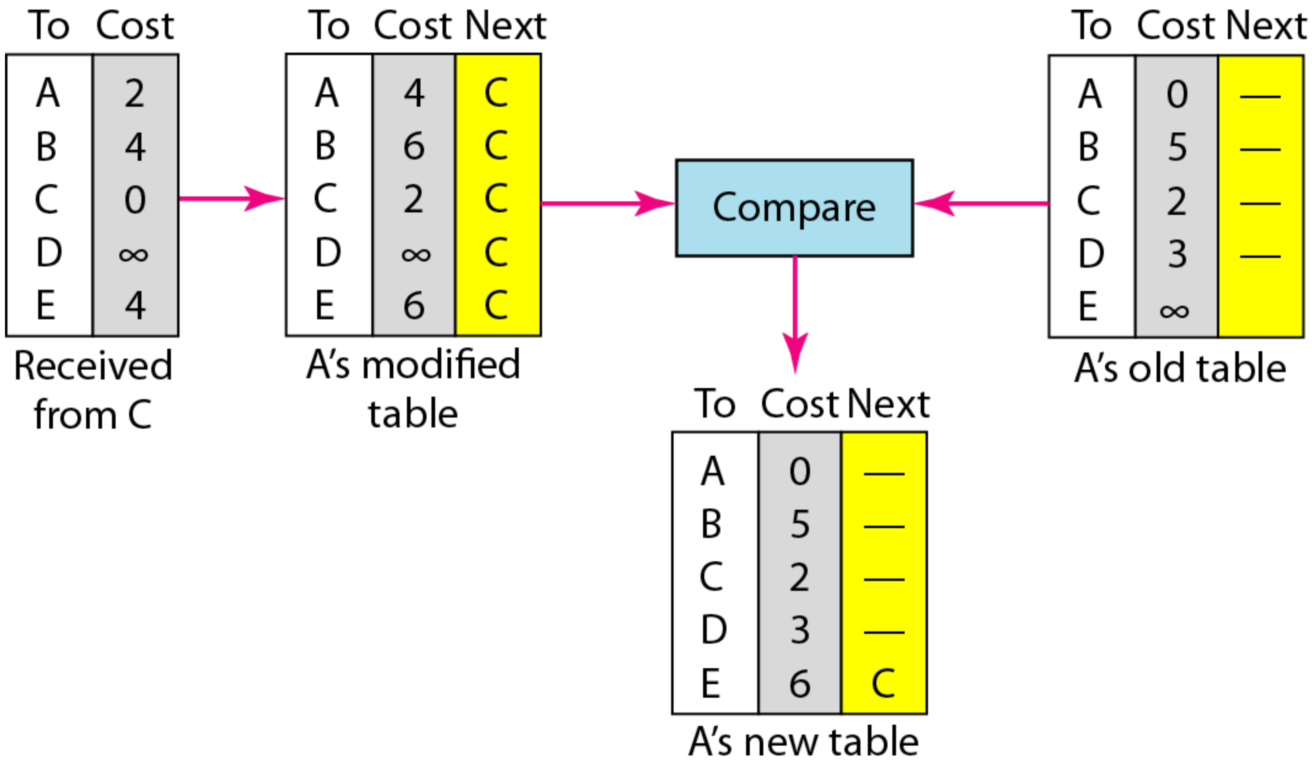
When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b. If the next-node entry is the same, the receiving node chooses the new row.

Figure 22.16 *Updating in distance vector routing*



To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's modified table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	∞	

A's old table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's new table

Figure 22.17 *Two-node instability*

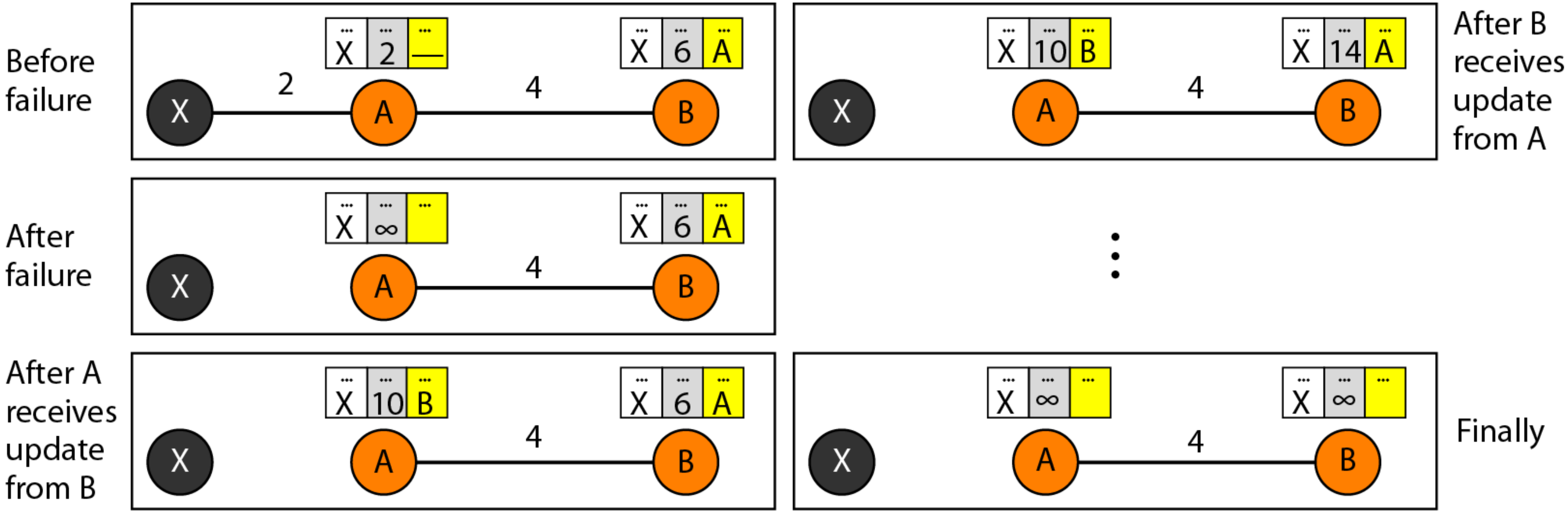
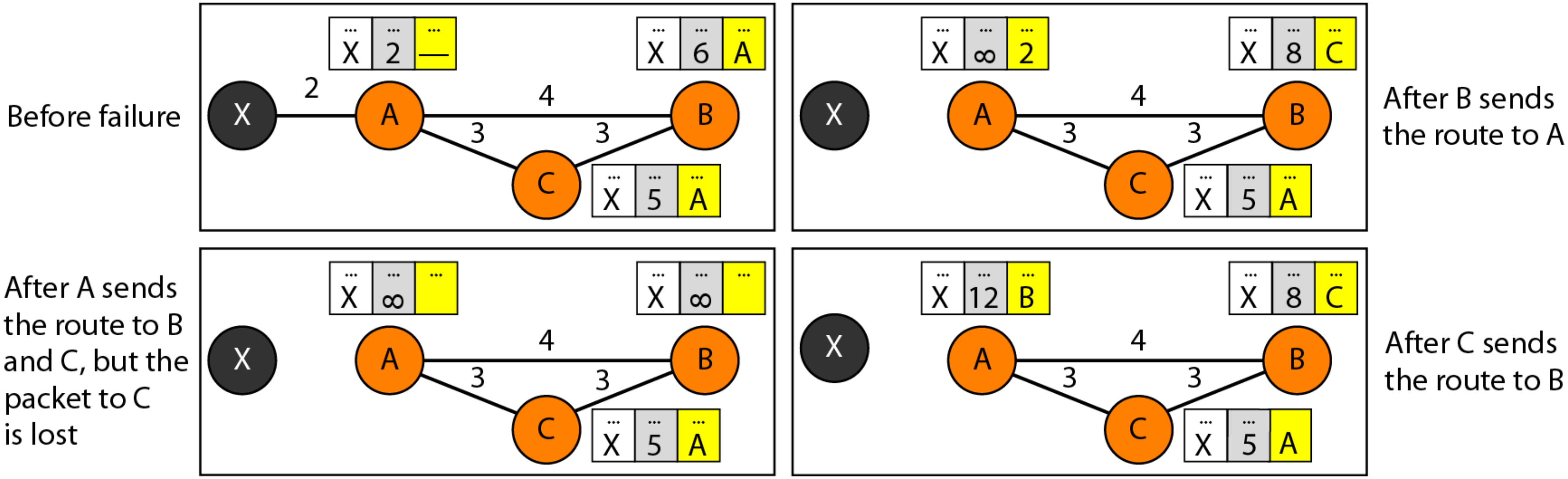


Figure 22.18 *Three-node instability*


Before failure
After B sends the route to A
After A sends the route to B and C, but the packet to C is lost
After C sends the route to B
X
A
B
C
2
4
3
3
X 2 —
X 6 A
X 5 A
X ∞ 2
X 8 C
X ∞
X 12 B

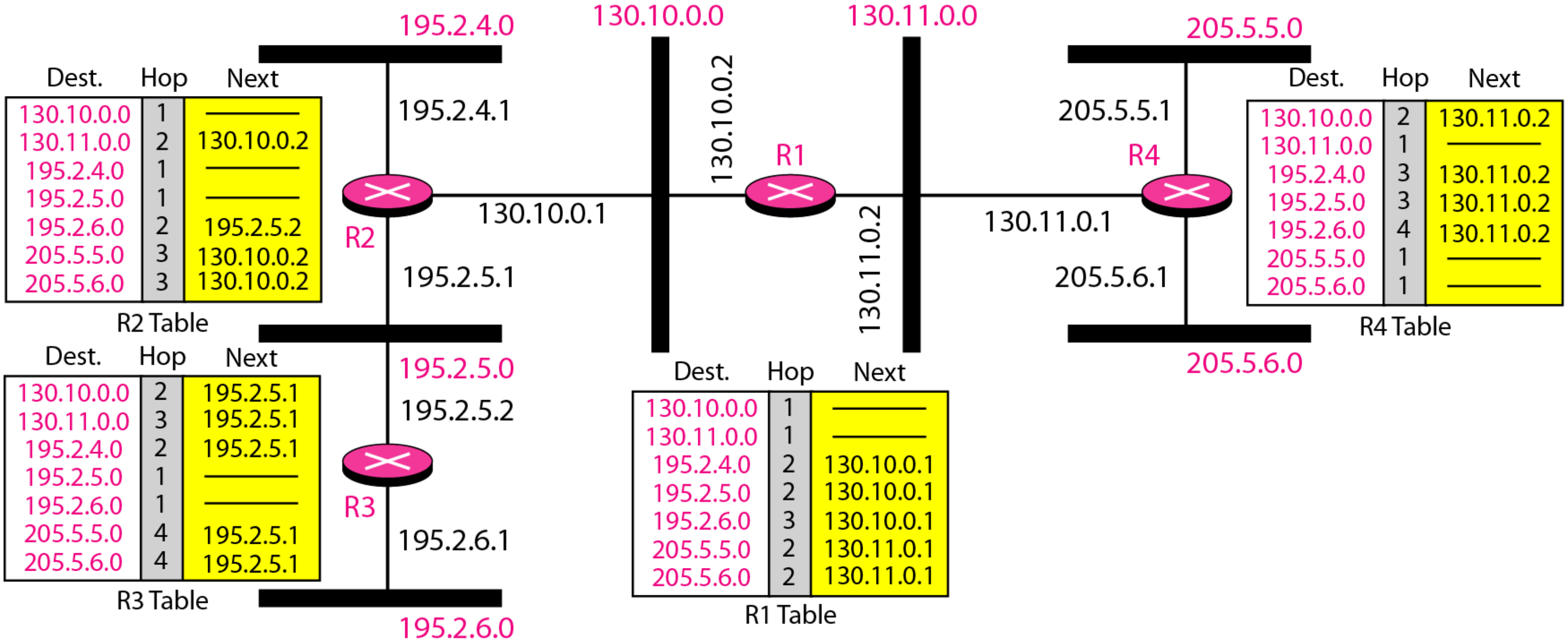
RIP

- The Routing Information Protocol (RIP) is an intradomain routing protocol used inside an autonomous system.
- It is a very simple protocol based on distance vector routing using hop count
- Table entries updated using values received from neighbors
- Maintain timers to detect failed links
- Used in first generation ARPANET

RIP implements distance vector routing directly with some considerations:

1. In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Figure 22.19 *Example of a domain using RIP*



Dest.	Hop	Next
130.10.0.0	1	———
130.11.0.0	2	130.10.0.2
195.2.4.0	1	———
195.2.5.0	1	———
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R2 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	———
195.2.6.0	1	———
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R3 Table

Dest.	Hop	Next
130.10.0.0	1	———
130.11.0.0	1	———
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

R1 Table

Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	———
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	———
205.5.6.0	1	———

R4 Table

Problems:

- Slow convergence for larger networks
- If a network becomes inaccessible, it may take a long time for all other routing tables to know this.(after a no of msg transfers)
- Routing loops may take a long time to be detected (counting to infinity problem)
- Too much bandwidth consumed by routing updates.

Link State Routing

- Link state routing has a different philosophy from that of distance vector routing.
- In link state routing, if each node in the domain has the entire topology of the domain,the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.
- Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology

Figure 22.20 *Concept of link state routing*

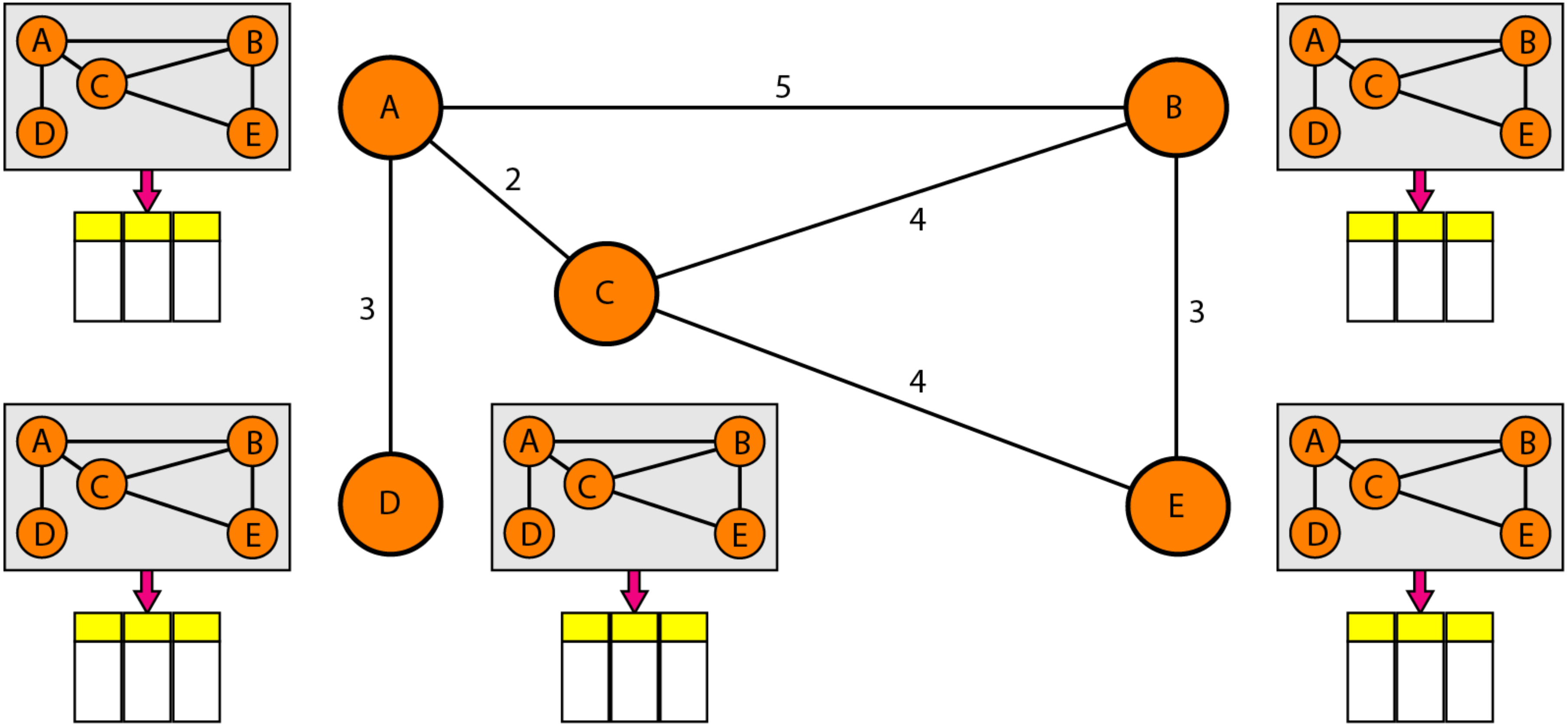


Figure 22.21 *Link state knowledge*

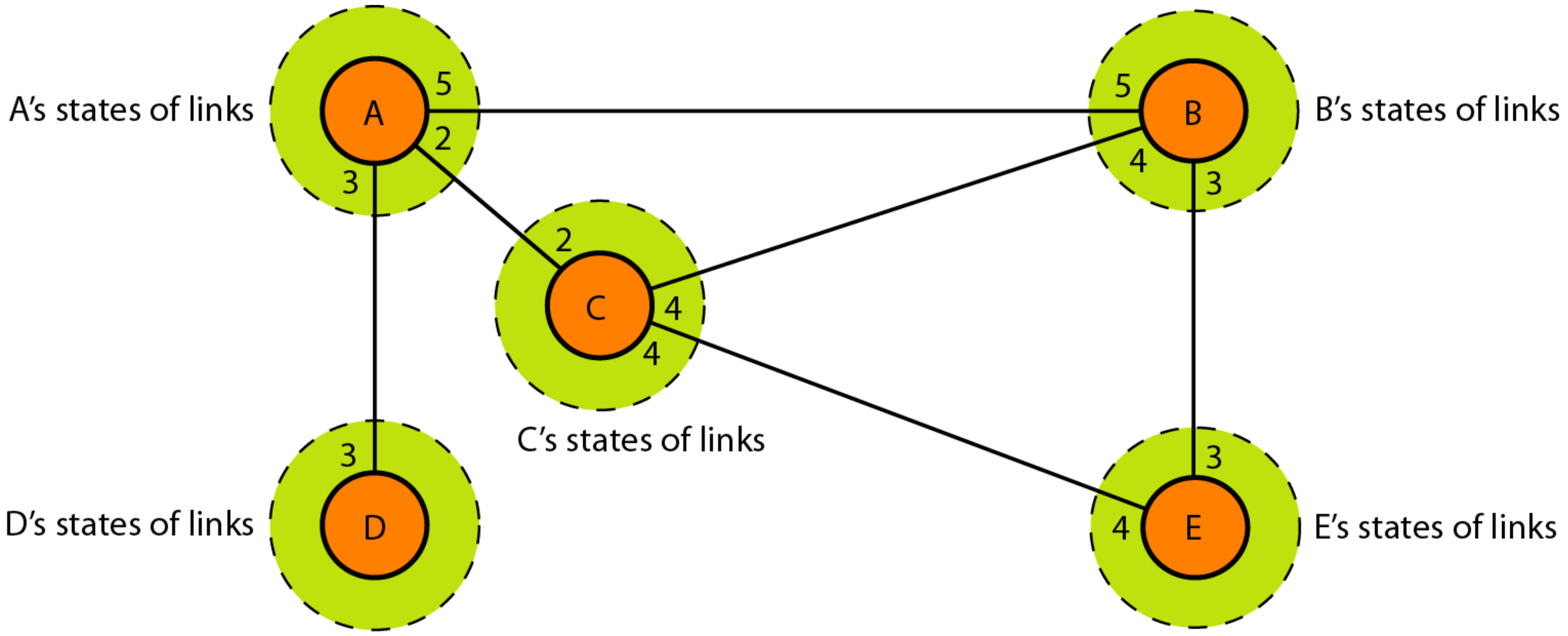
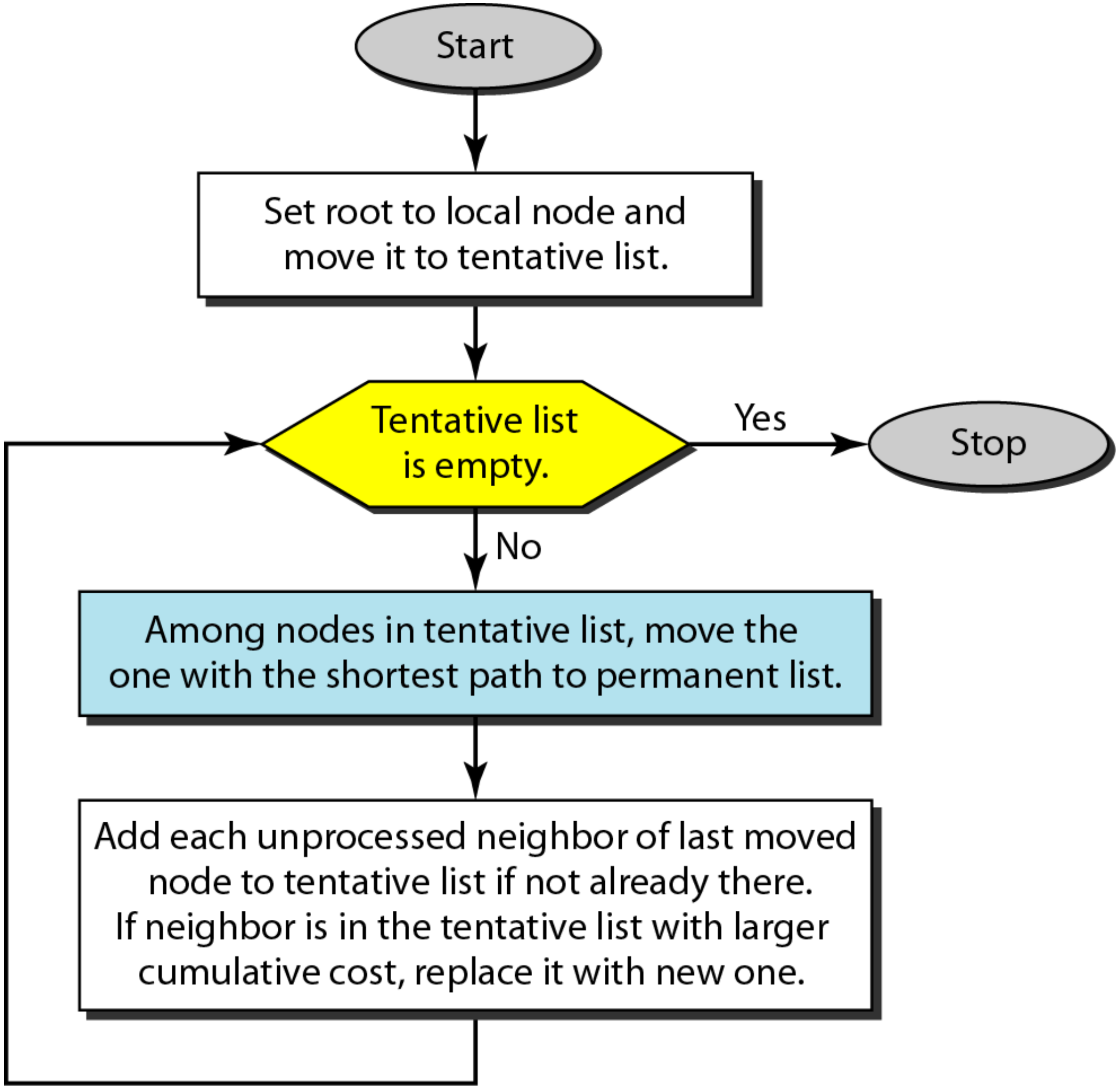
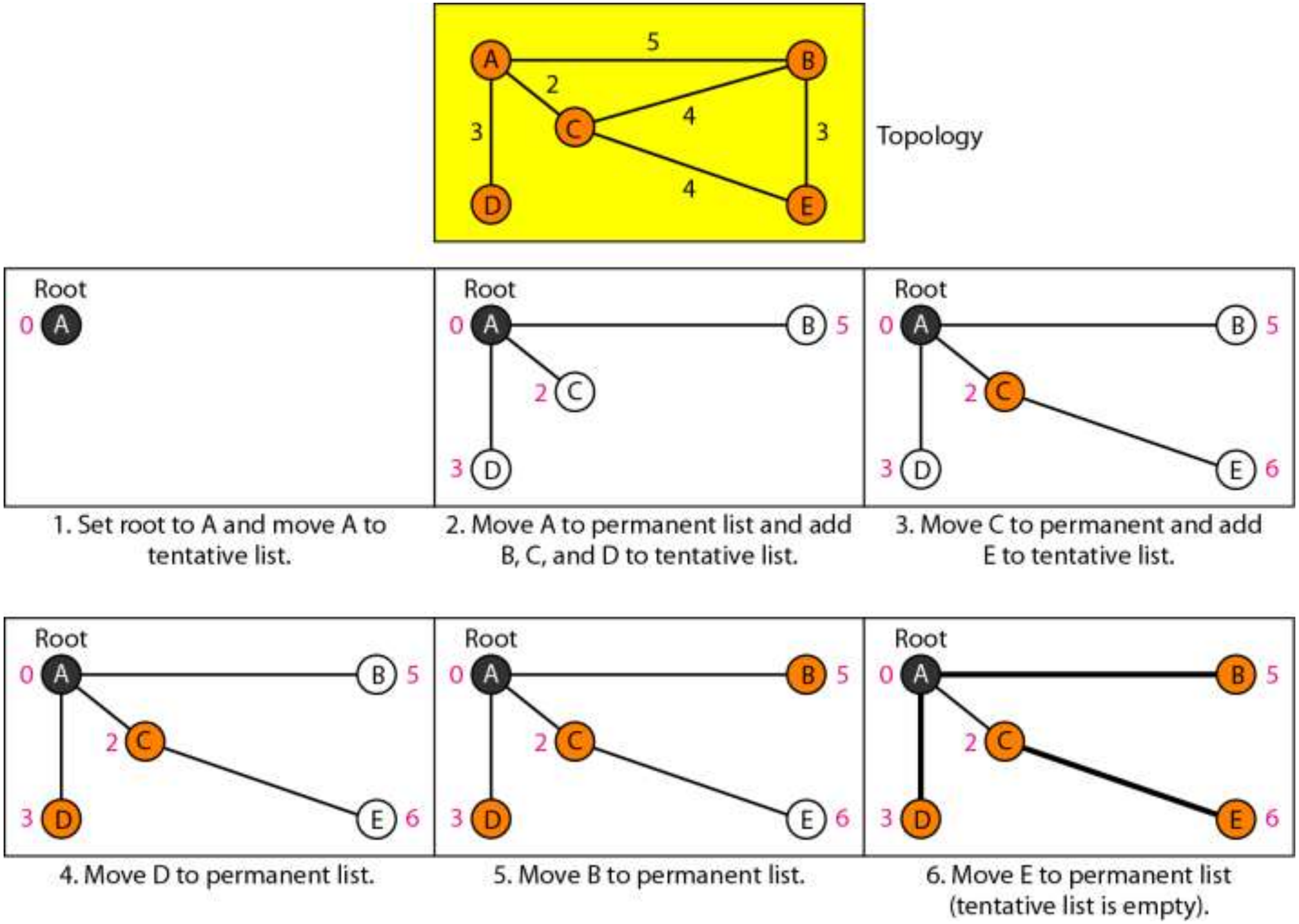


Figure 22.22 *Dijkstra algorithm*



Start
Set root to local node and move it to tentative list.
Tentative list is empty.
Yes
Stop
No
Among nodes in tentative list, move the one with the shortest path to permanent list.
Add each unprocessed neighbor of last moved node to tentative list if not already there.
If neighbor is in the tentative list with larger cumulative cost, replace it with new one.

Figure 22.23 *Example of formation of shortest path tree*



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.

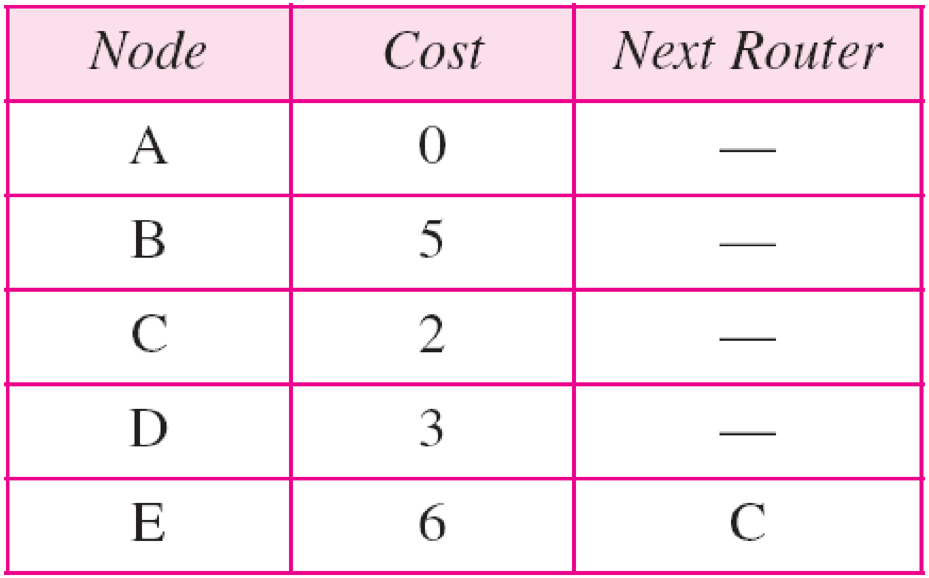
3. Move C to permanent and add E to tentative list.

4. Move D to permanent list.

5. Move B to permanent list.

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

Table 22.2 ***Routing table for node A***



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

OSPF

- The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.
- Widely used as the interior router protocol in TCP/IP networks
- Basic Concept:

➢ Computes a route that incurs the least cost.

✓ User configurable: delay, data rate, cost, etc.

➢ Each router maintains a database

✓ Topology of the autonomous system to which the router belongs

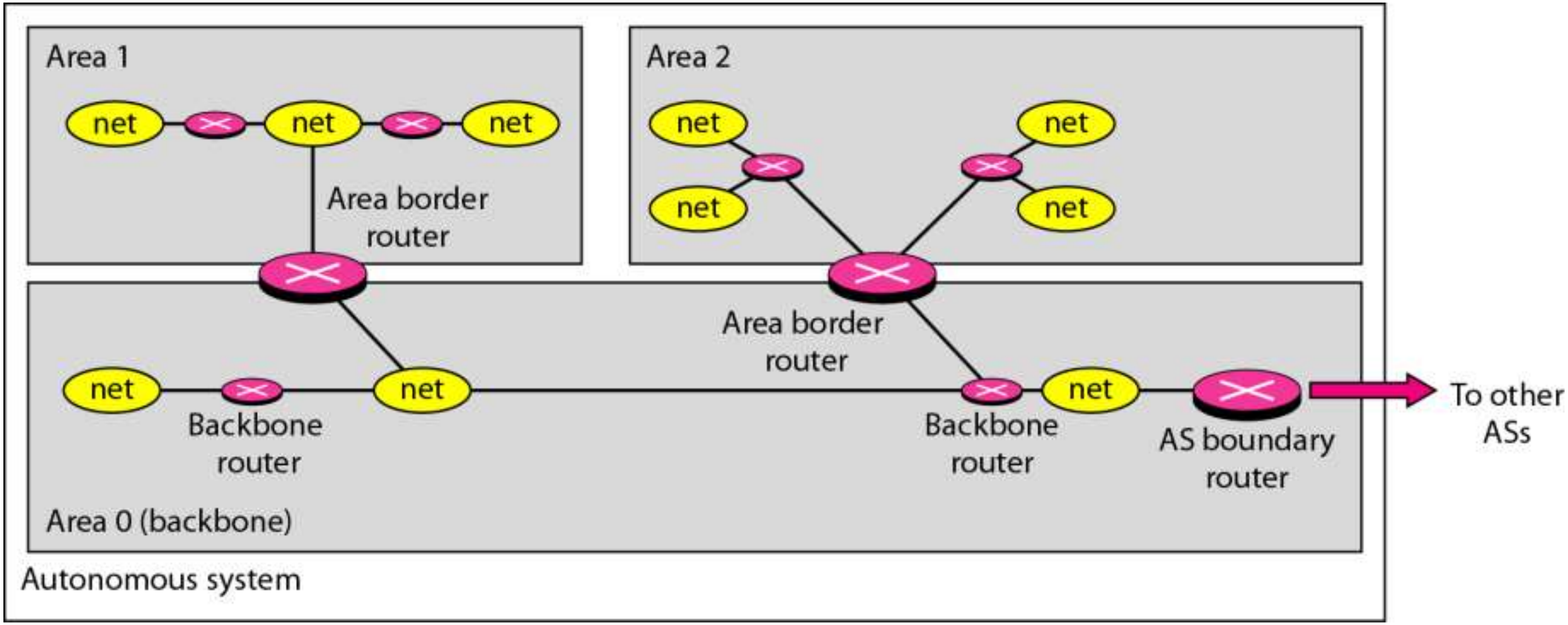
✓ Vertices and edges.

Areas To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system.

- An autonomous system can be divided into many different areas. All networks inside an area must be connected

- Routers inside an area flood the area with routing information.
- At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas.
- Among the areas inside an autonomous system , there is a special area called the backbone; all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas.
- The routers inside the backbone are called the backbone routers. A backbone router can also be an area border router
- Each area has an area identification. The area identification of the backbone is zero.

Figure 22.24 *Areas in an autonomous system*



- Metric The OSPF protocol allows the administrator to assign a cost, called the metric, to each route.
- The metric can be based on a type of service (minimum delay, maximum throughput, and so on).
- Types of Links In OSPF terminology, a connection is called a link. Four types of links have been defined: point-to-point, transient, stub, and virtual

Figure 22.25 *Types of links*

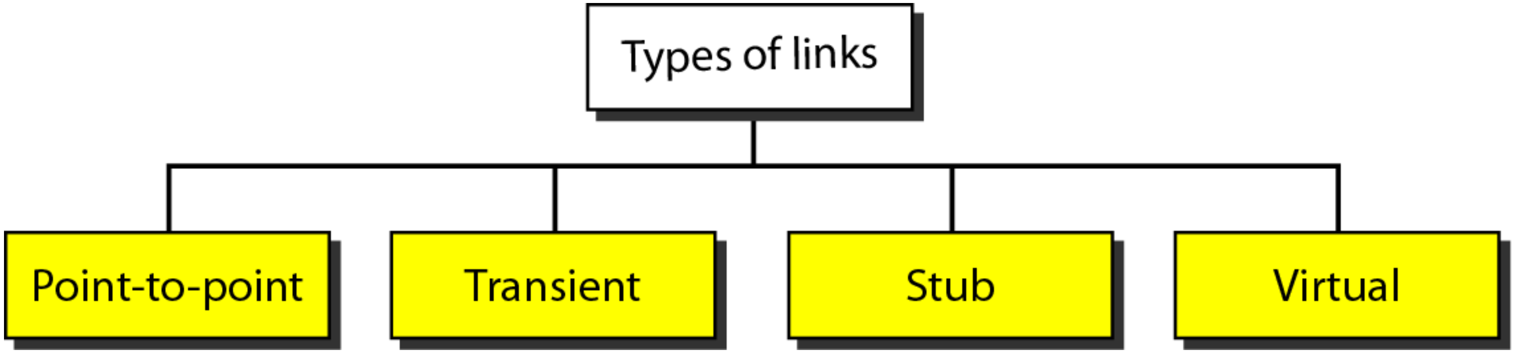


Figure 22.26 *Point-to-point link*

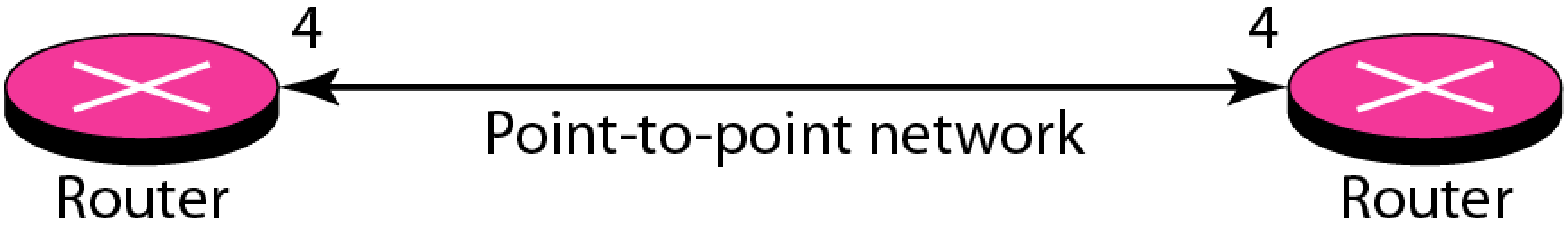
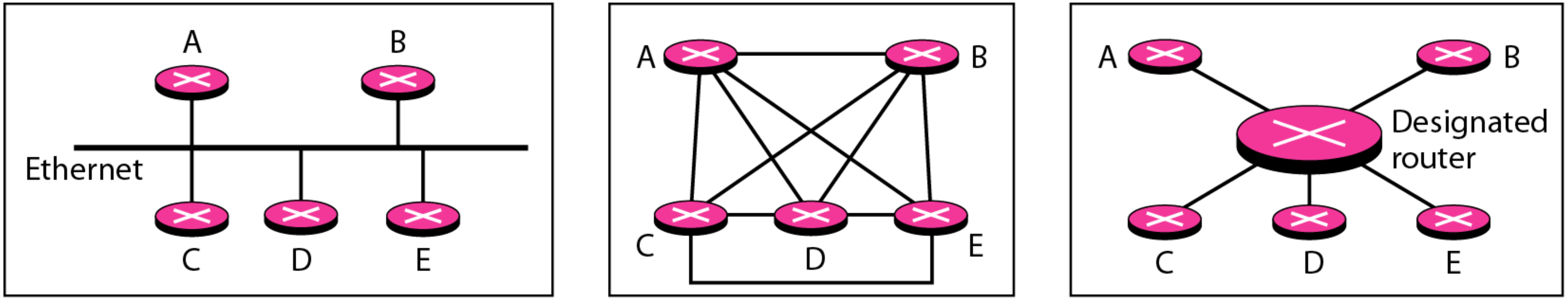


Figure 22.27 *Transient link*



a. Transient network

b. Unrealistic representation

c. Realistic representation

Figure 22.28 *Stub link*

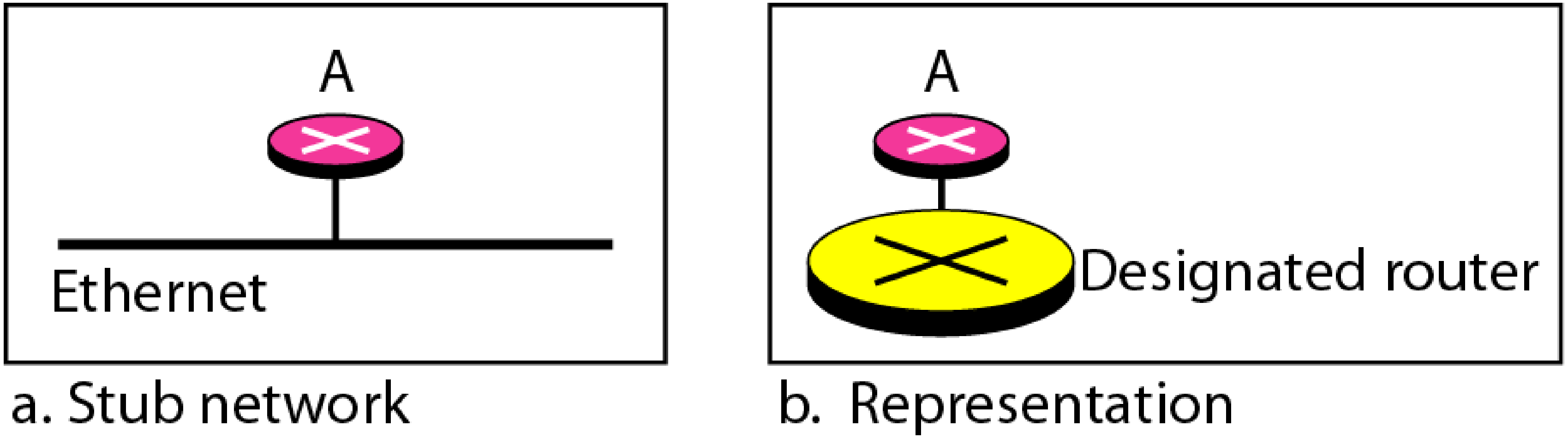
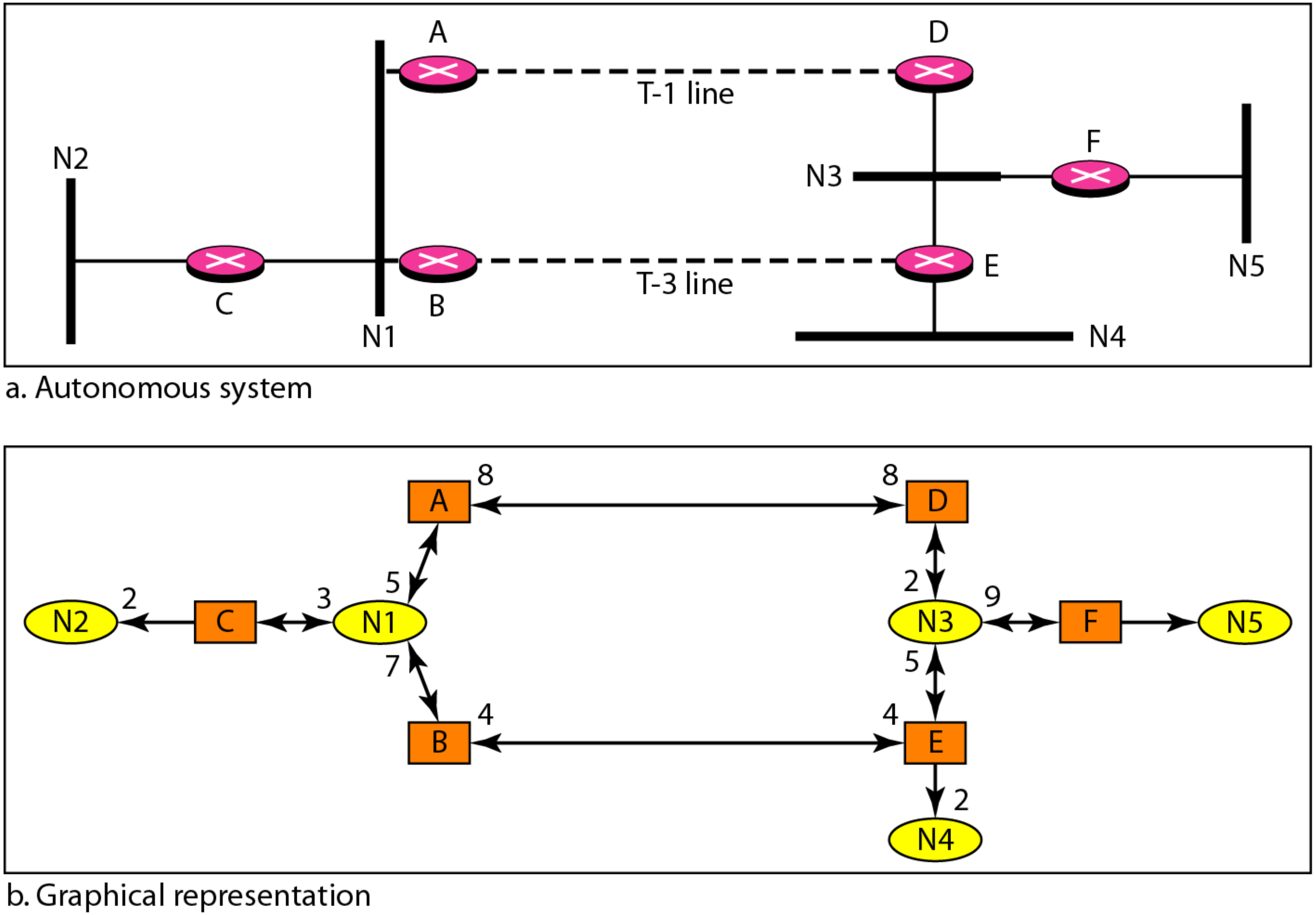


Figure 22.29 *Example of an AS and its graphical representation in OSPF*



a. Autonomous system

b. Graphical representation

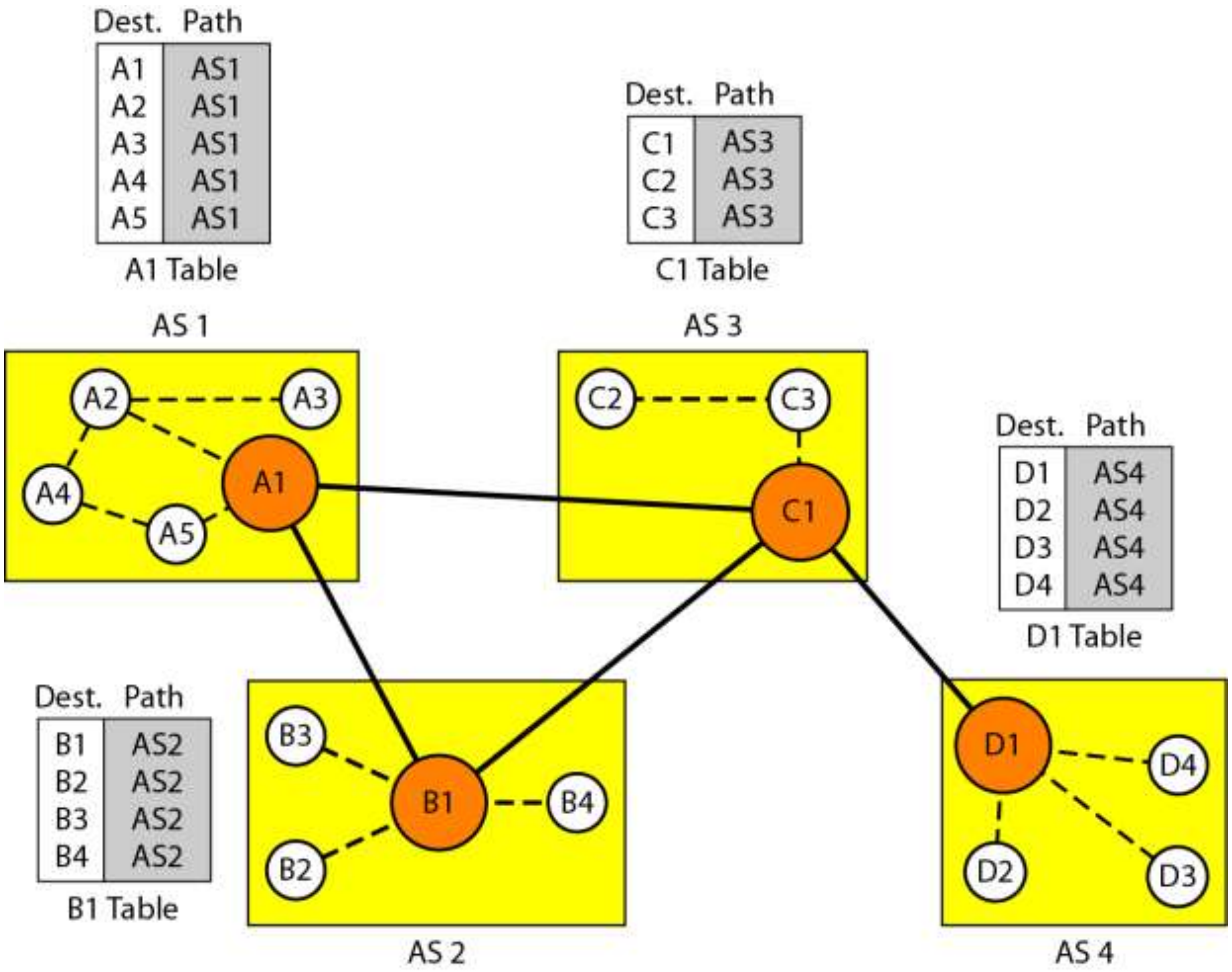
Path Vector Routing

- Path vector routing proved to be useful for interdomain routing.
- The principle of path vector routing is similar to that of distance vector routing.
- In path vector routing, we assume that there is one node in each autonomous system that acts on behalf of the entire autonomous system which is called speaker node.
- The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs.
- A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems

Initialization

- At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system.
- Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4.
- Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it.
- Node Bl advertises that Bl to B4 are located in AS2 and can be reached through Bl. And so on.

Figure 22.30 *Initial routing tables in path vector routing*



A1 Table

Dest.	Path
A1	AS1
A2	AS1
A3	AS1
A4	AS1
A5	AS1

C1 Table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

D1 Table

Dest.	Path
D1	AS4
D2	AS4
D3	AS4
D4	AS4

B1 Table

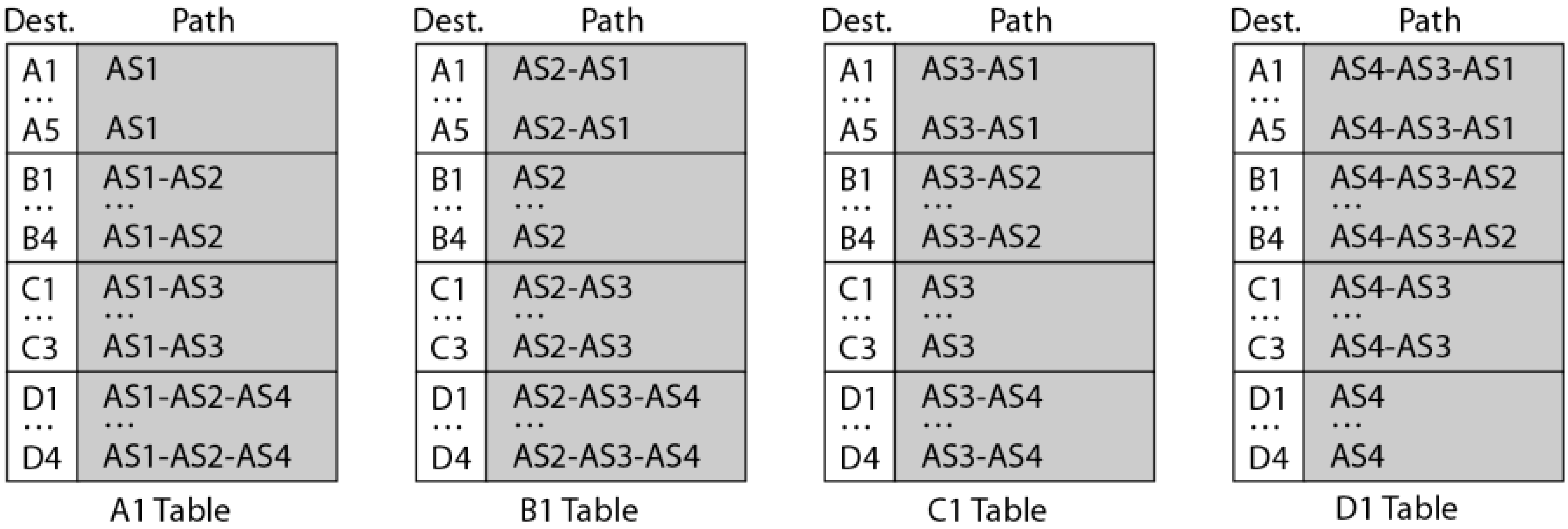
Dest.	Path
B1	AS2
B2	AS2
B3	AS2
B4	AS2

Sharing Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

Updating When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other ASs.

Loop prevention The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.

Figure 22.31 *Stabilized tables for three autonomous systems*



Dest.	Path
A1	AS1
…	
A5	AS1
B1	AS1-AS2
…	…
B4	AS1-AS2
C1	AS1-AS3
…	…
C3	AS1-AS3
D1	AS1-AS2-AS4
…	…
D4	AS1-AS2-AS4

A1 Table

Dest.	Path
A1	AS2-AS1
…	
A5	AS2-AS1
B1	AS2
…	…
B4	AS2
C1	AS2-AS3
…	…
C3	AS2-AS3
D1	AS2-AS3-AS4
…	…
D4	AS2-AS3-AS4

B1 Table

Dest.	Path
A1	AS3-AS1
…	
A5	AS3-AS1
B1	AS3-AS2
…	…
B4	AS3-AS2
C1	AS3
…	…
C3	AS3
D1	AS3-AS4
…	…
D4	AS3-AS4

C1 Table

Dest.	Path
A1	AS4-AS3-AS1
…	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
…	…
B4	AS4-AS3-AS2
C1	AS4-AS3
…	…
C3	AS4-AS3
D1	AS4
…	…
D4	AS4

D1 Table

BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing.

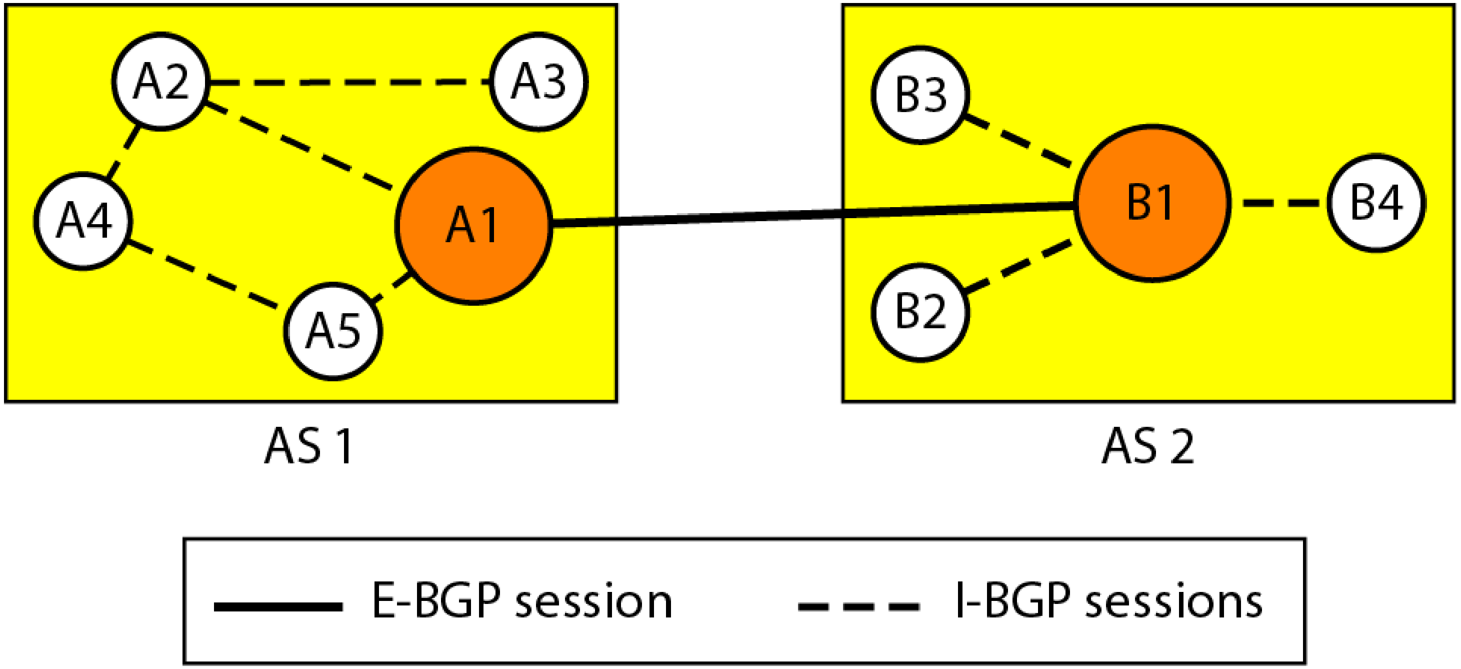
BGP Sessions The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment,BGP uses the services of TCP.

External and Internal BGP

BGP can have two types of sessions:

- ✓ external BGP (E-BGP) and
- ✓ internal BGP (I-BGP) sessions.
- The E-BGP session is used to exchange information between two speaker nodes belonging to two different autonomou systems.
- The I-BGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.

Figure 22.32 *Internal and external BGP sessions*



Message types in BGP:

1. Open-used to open a neighbor connection with another router.
2. Update-used to transmit information about a single route
3. Keepalive-used to periodically confirm the neighbor connection
4. Notification-used to notify about some error condition

Functional procedures in BGP:

- Neighbor acquisition-two routes agree to be neighbors by exchanging messages
- Neighbor reachability-check if the neighbor is still alive and is maintaining the relationship
- Network reachability-each router maintains a list of networks that it can reach and the preferred routes.