Module 2

Network Layer: Delivery, Forwarding, and Routing

22-1 DELIVERY

The network layer <u>supervises</u> the <u>handling</u> of the packets by the underlying physical networks.

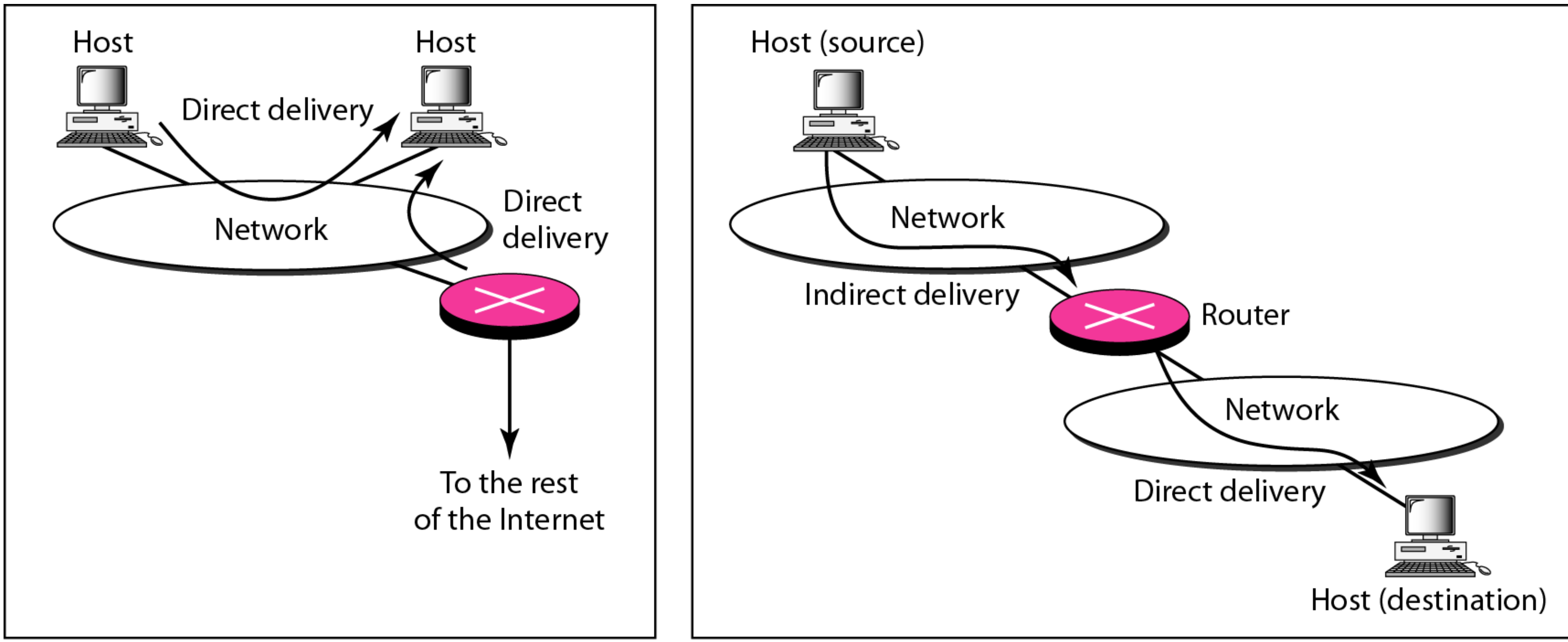
→ ***defined as the delivery of a packet.***

Direct Versus Indirect Delivery

Direct delivery occurs when **source and destination of packet are located on the same physical network** *OR*
When the delivery is between last Router and destination Host

In indirect delivery, **packet travels from router to router** until it reaches the one connected to same physical network as the 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

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 refers to this table to find the route to the final destination.

Topics discussed in this section:

Forwarding Techniques
Forwarding Process
Routing Table

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	---

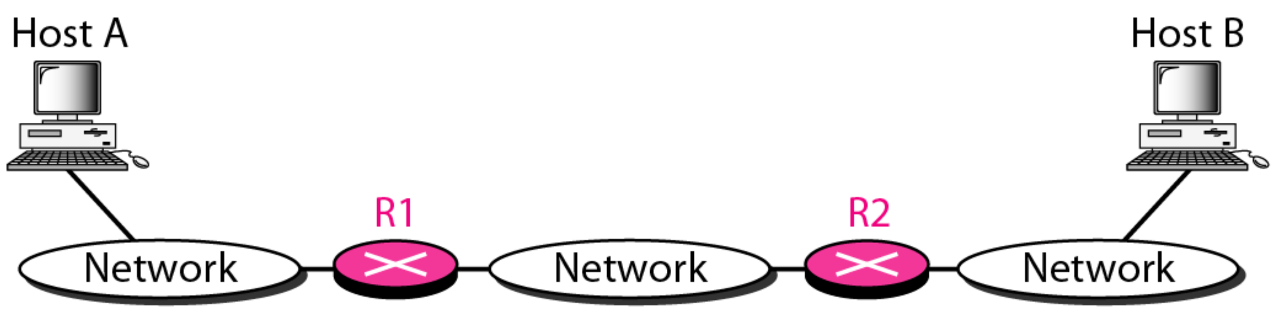


Figure 22.3 *Host-specific versus network-specific method*

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

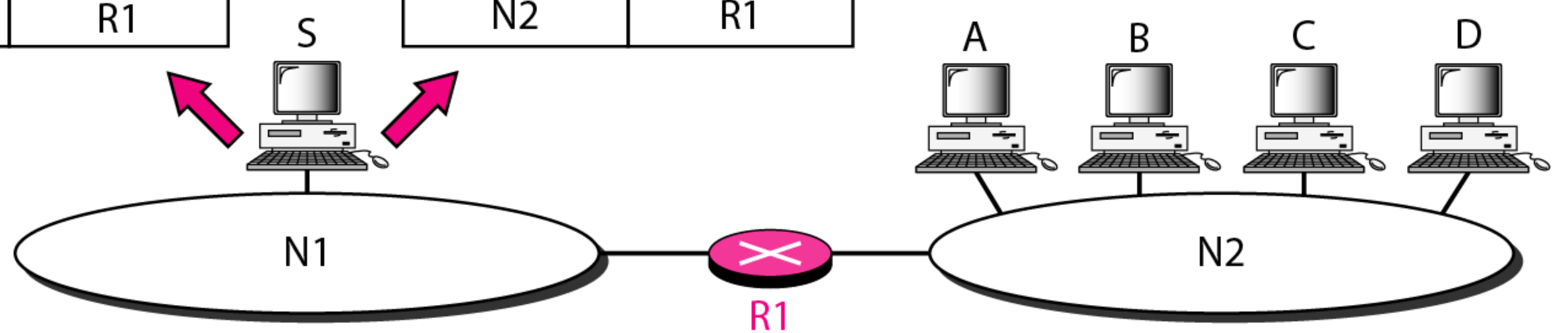


Figure 22.4 *Default method*

Destination	Next hop
N2	R1
Any other	R2

Routing table for host A

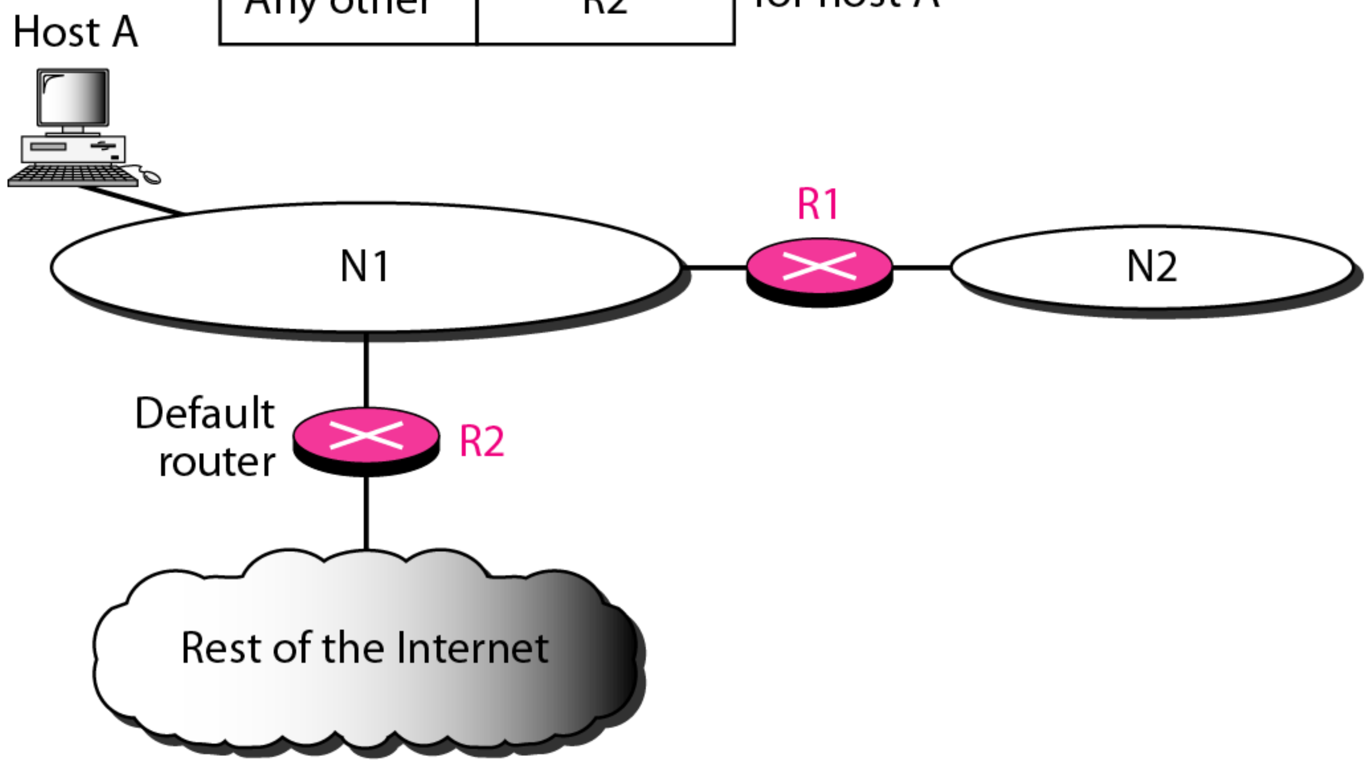
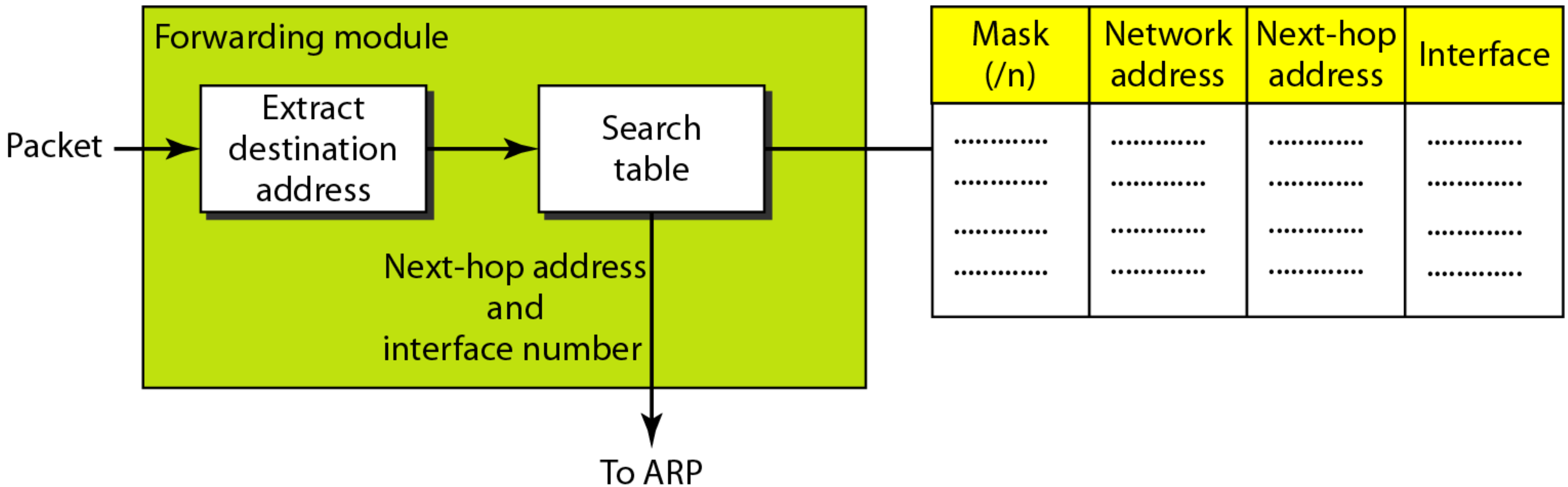


Figure 22.5 *Simplified forwarding module in classless address*



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*

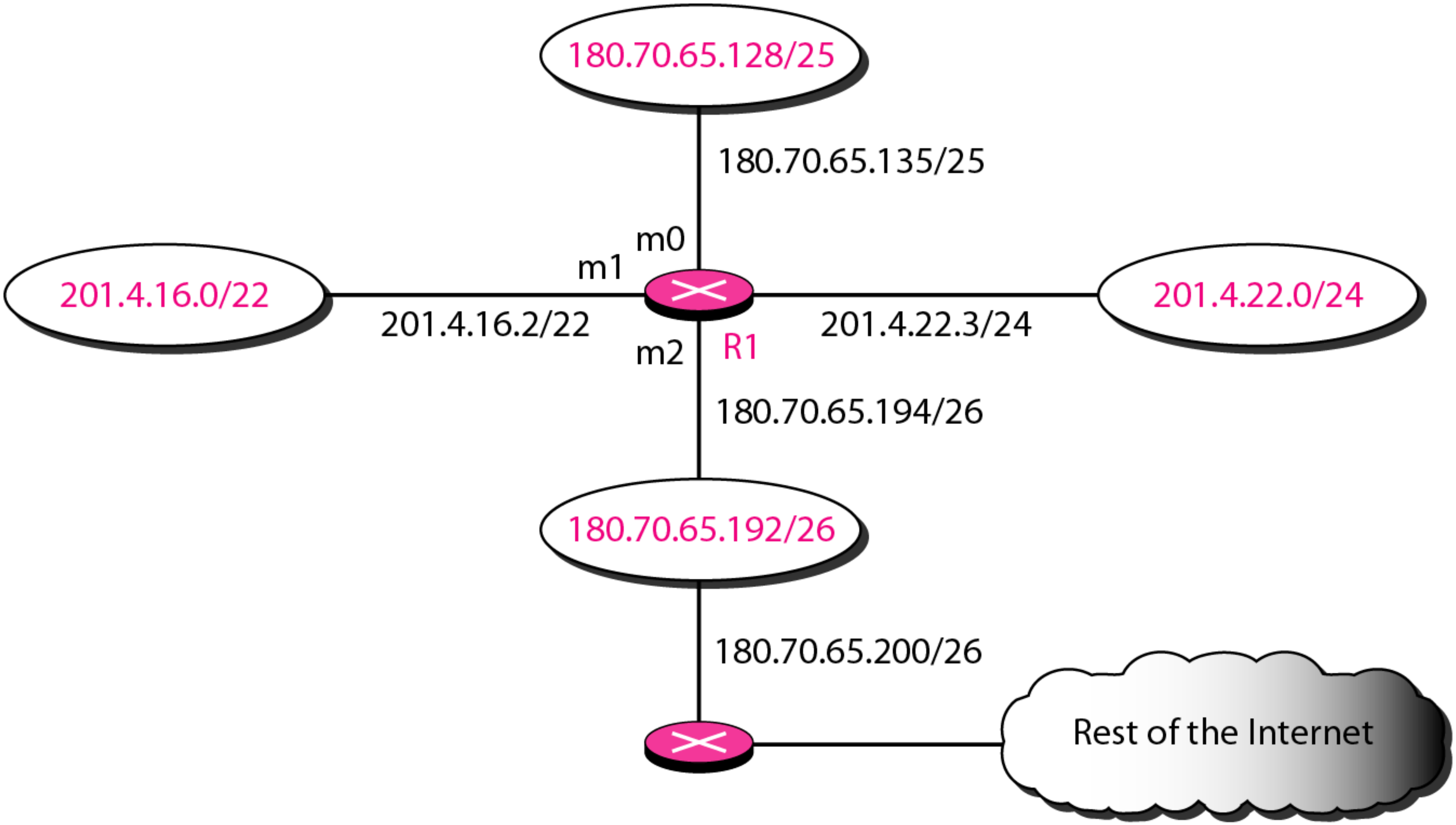


Table 22.1 ***Routing table for router R1 in Figure 22.6***

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

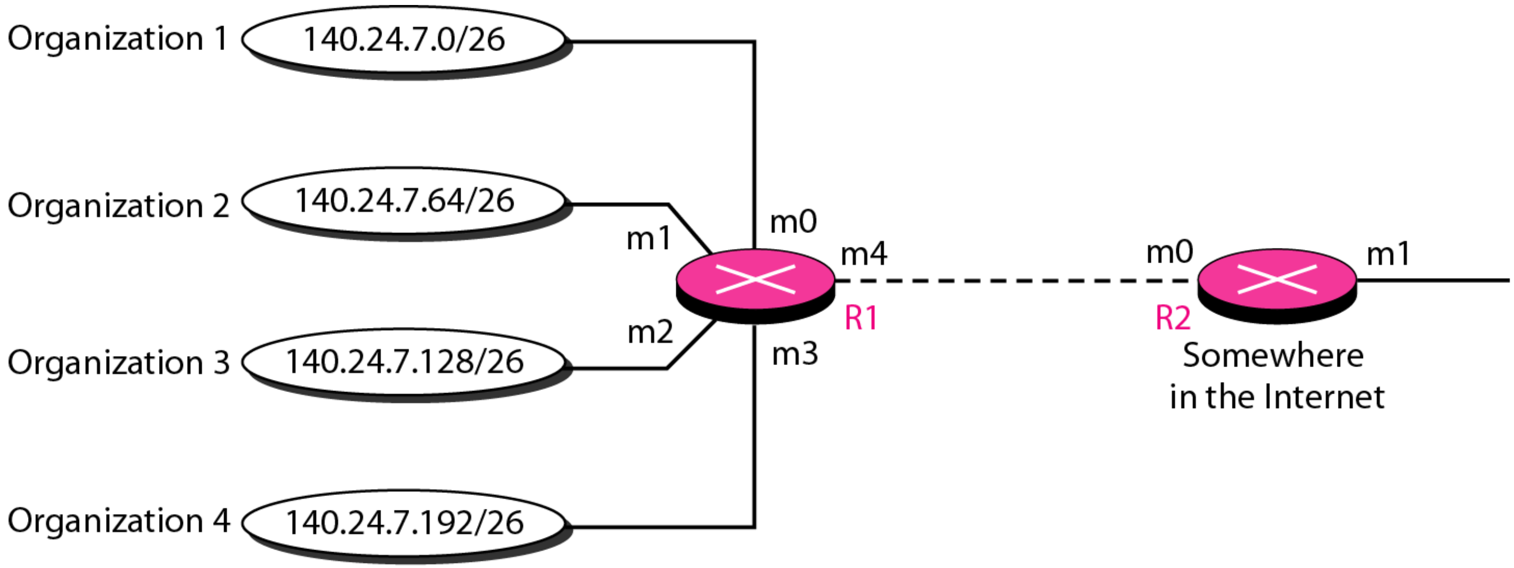
Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 18.24.32.78.

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 packet that needs to be sent, via the default router, to someplace elsewhere in the Internet.

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

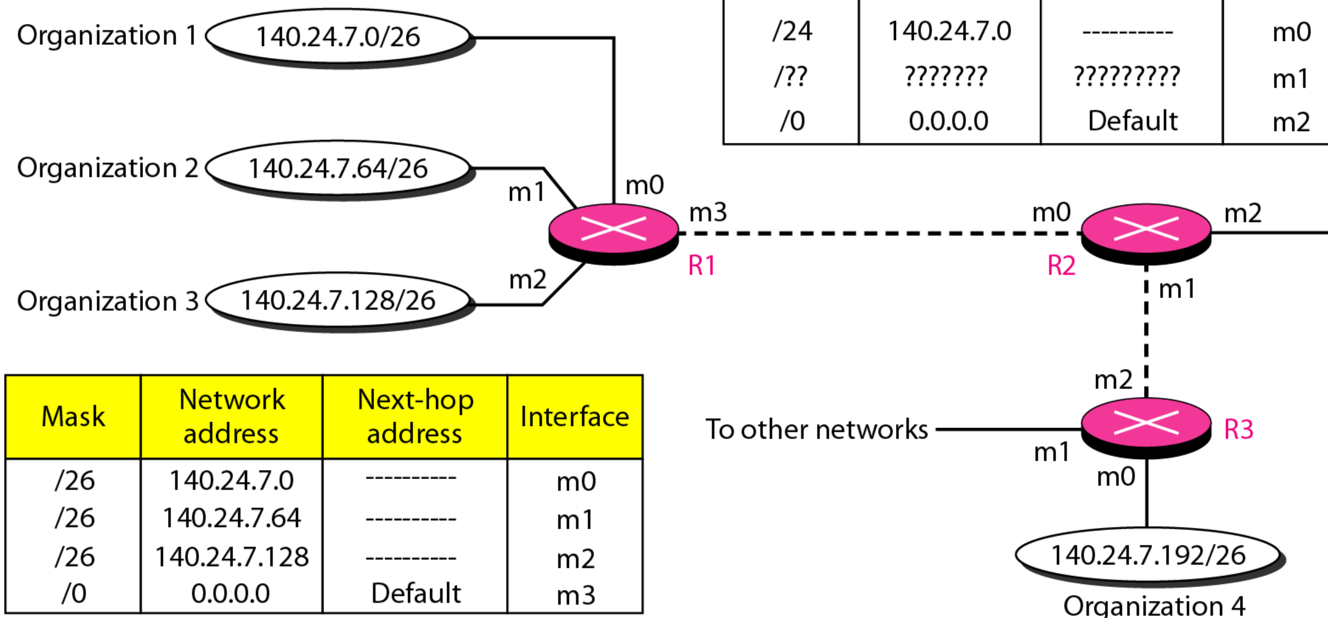
Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	Default	m1

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



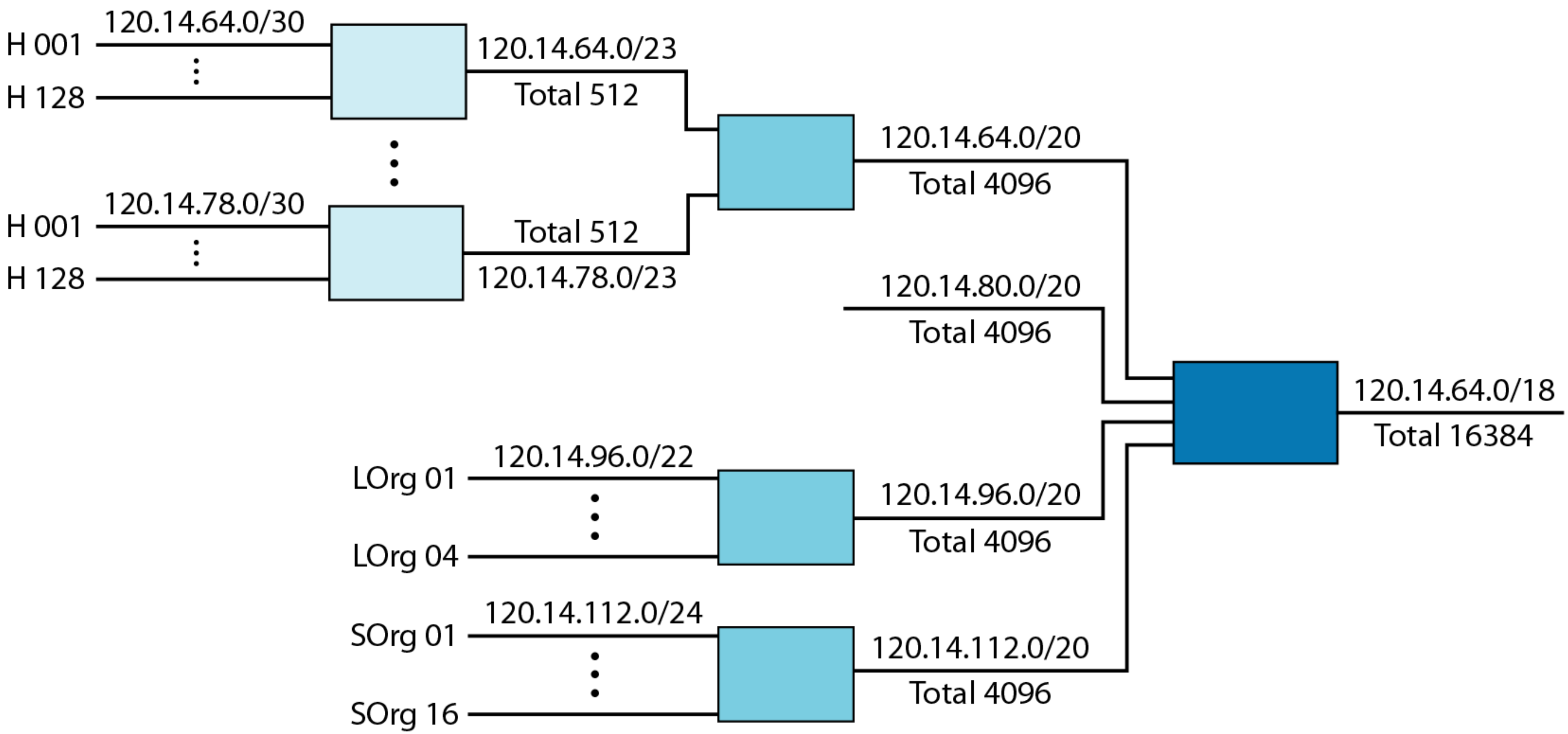
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	Default	m3

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

Figure 22.9 *Hierarchical routing with ISPs*



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 120.14.64.0/18

The regional ISP divides 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 → mask /23

Each small ISP provides services to 128 households, each using 4 addresses → mask /30

Example 22.5 (continued)

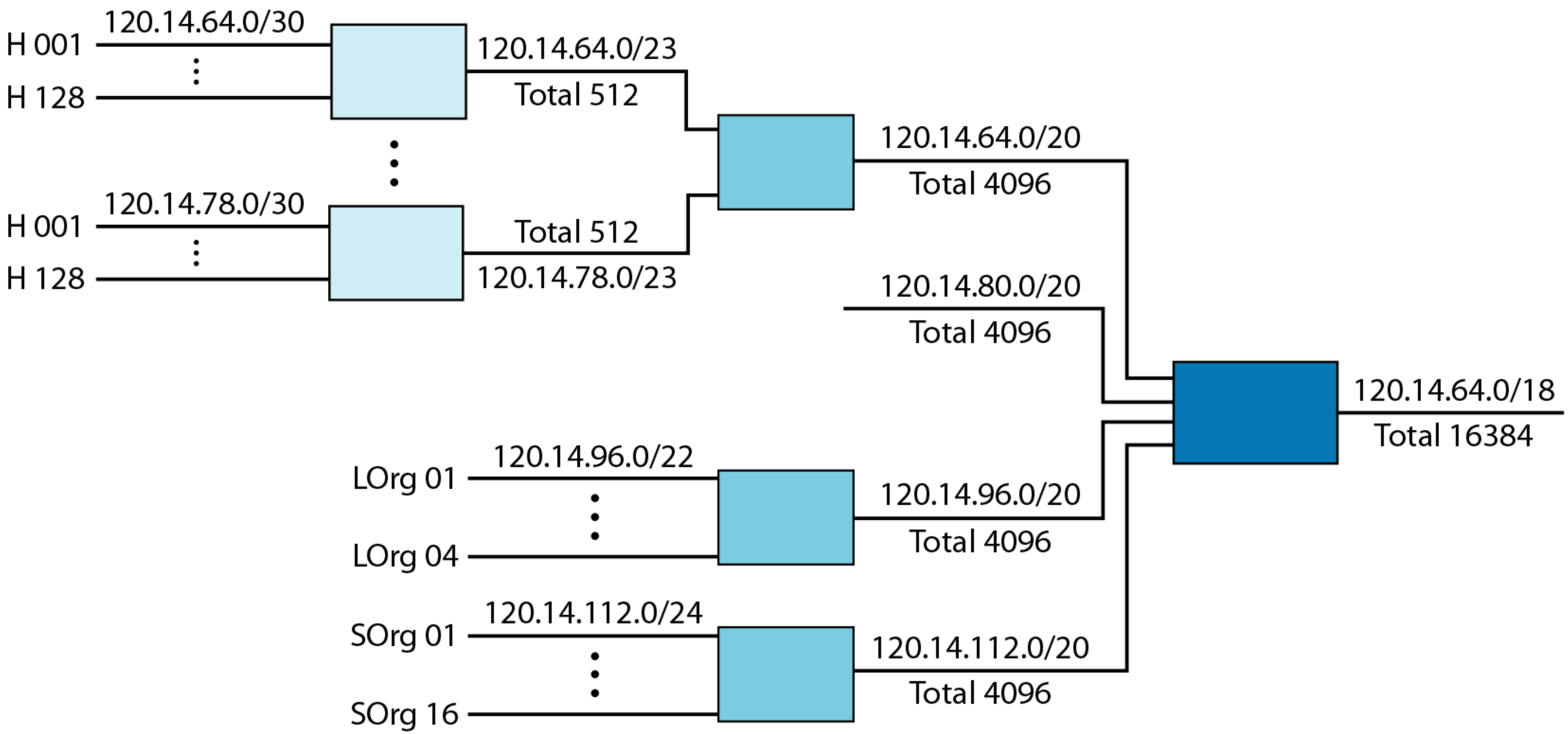
The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations → addresses 1024; mask is /22

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 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*



Routing Table

Static Routing tables- ***Entries done manually; fixed- cannot update automatically when there is a change in the network***

Dynamic Routing tables- ***Updated periodically (or whenever there is a change in the network) using Dynamic Routing protocol e.g RIP, OSPF etc.***

Common fields in a routing table

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

Example 22.6

Utility that can be used to find the contents of a routing table for a host or router is netstat in UNIX or LINUX.

Two options, r and n; 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

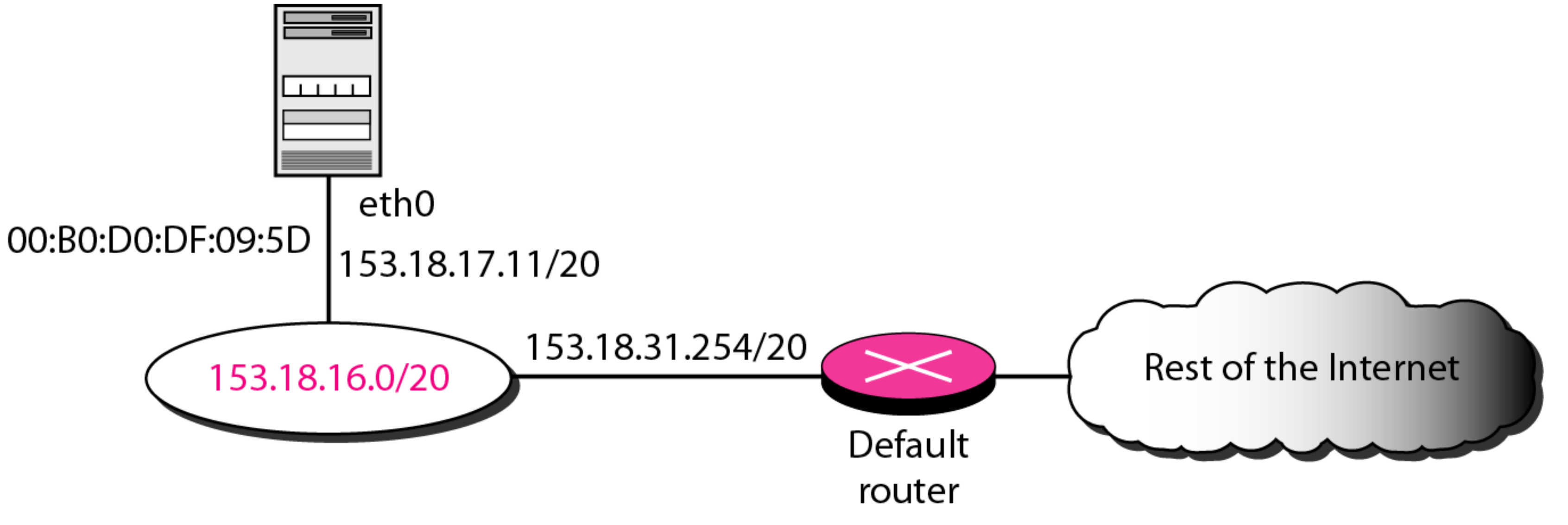
The Destination column here defines the network address. The term Gateway used by UNIX is synonymous with router that actually defines the next hop address. The value 0.0.0.0 shows that the delivery is direct. The last entry has a flag of G, which means that the destination can be reached through a router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



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

Figure 22.12 ***Autonomous systems***

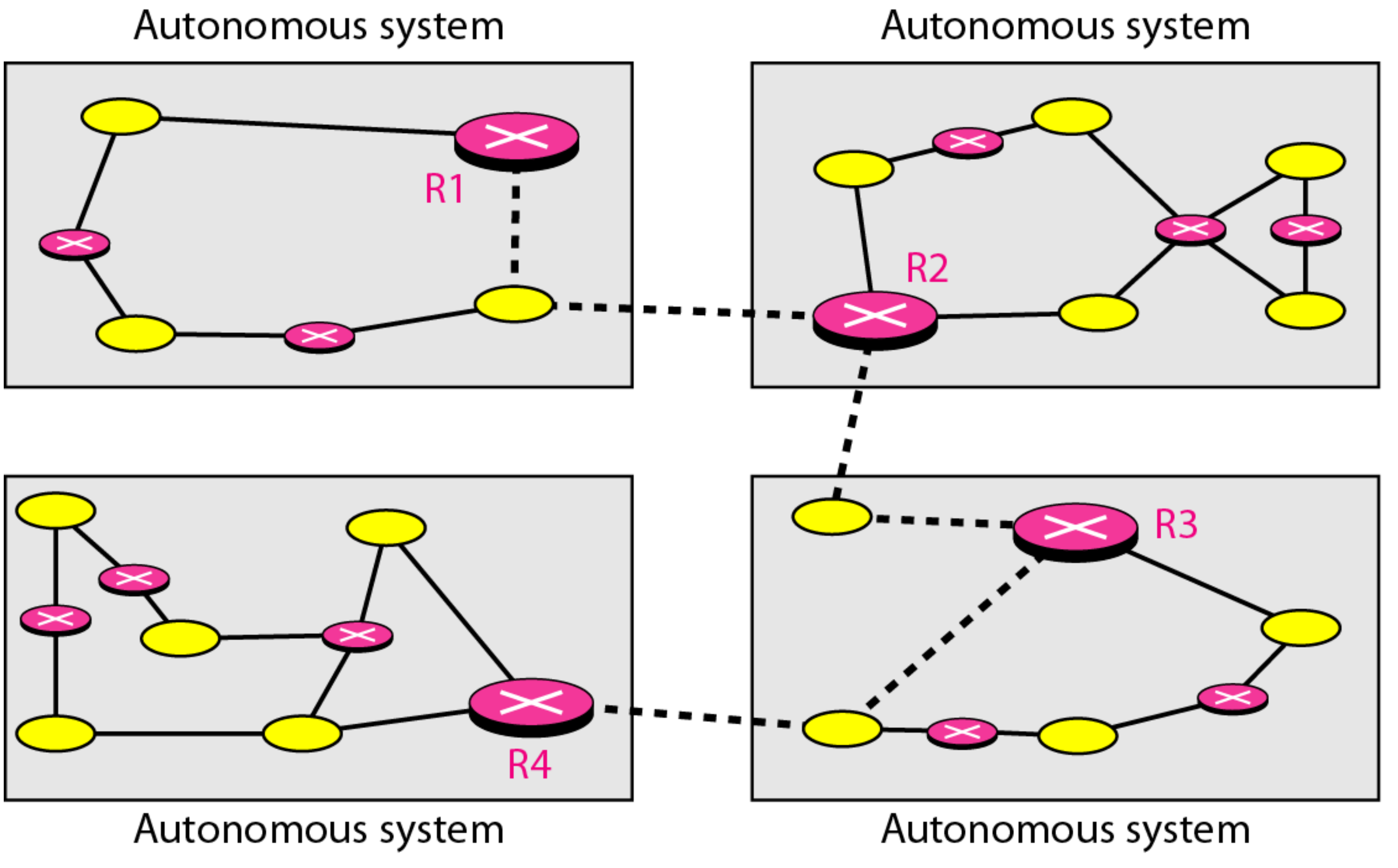
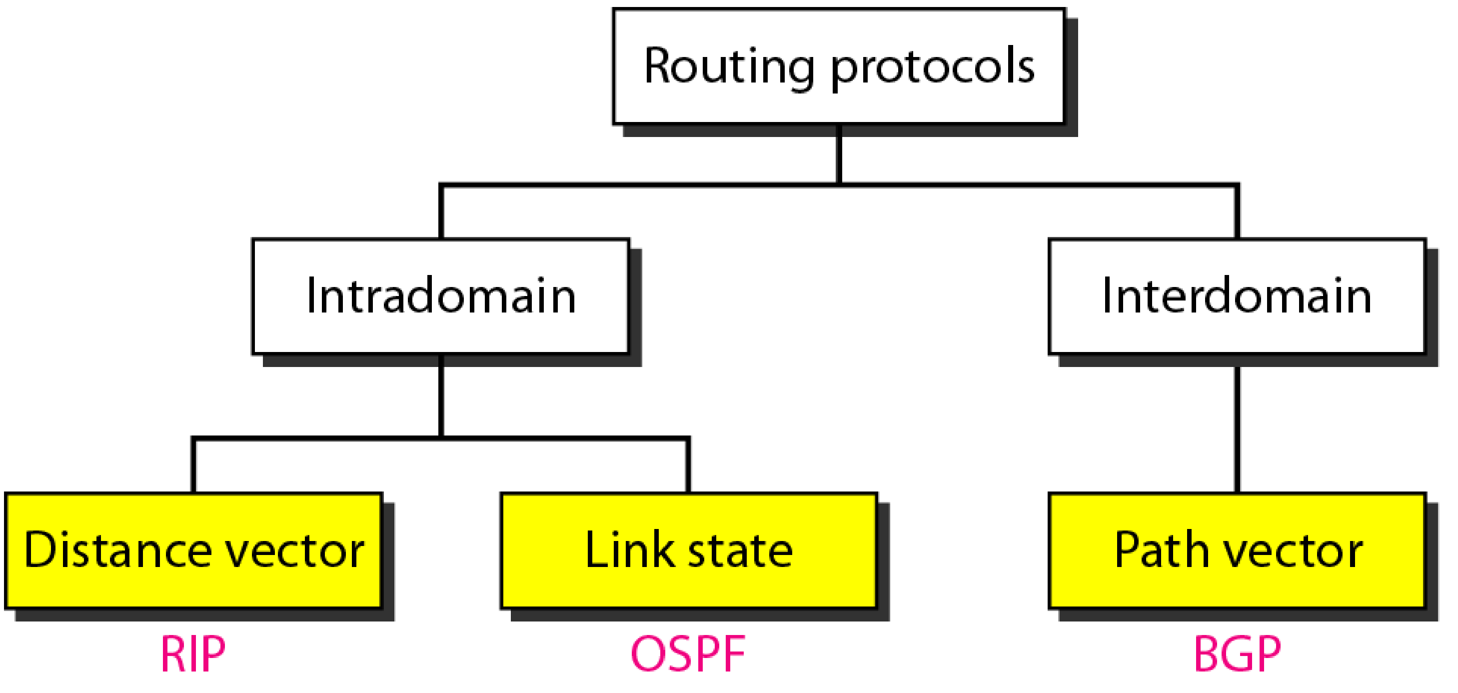


Figure 22.13 *Popular routing protocols*



Distance Vector Routing

Goal

•To find the best route (least cost/distance route) to the destination

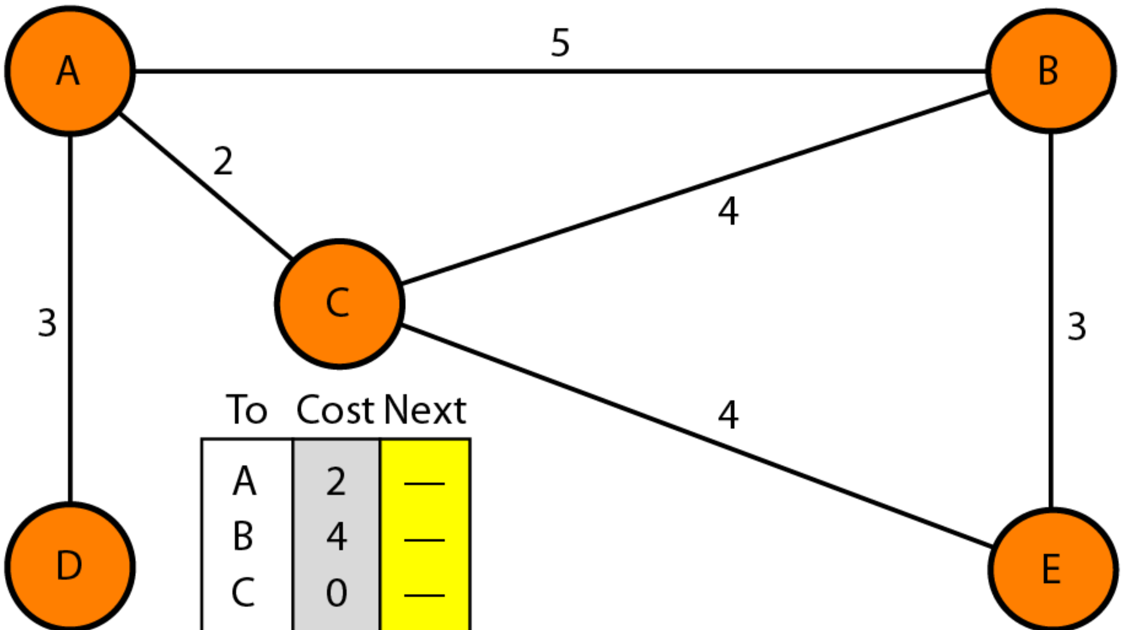
Principle

•Nodes share the local knowledge to acquire the global

Steps

- ***Acquire the local knowledge (learn)***
- ***Share this with others (cooperate)***
- ***Update global knowledge***

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

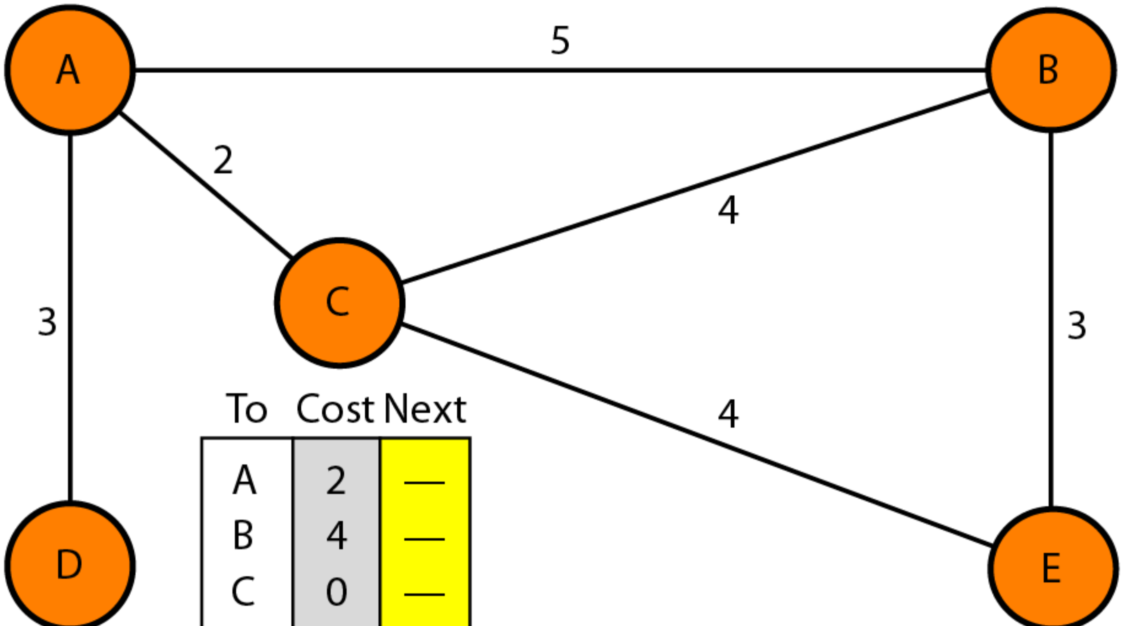
To	Cost	Next
A	2	—
B	4	—
C	0	—
D	5	A
E	4	—

C's table

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

E's table

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 the table

Comparison

- ***If next node entry is different, receiving node chooses the entry with smaller cost.***

- ***If next node entry is same, receiving node chooses the new cost***

Figure 22.16 *Updating in distance vector routing*

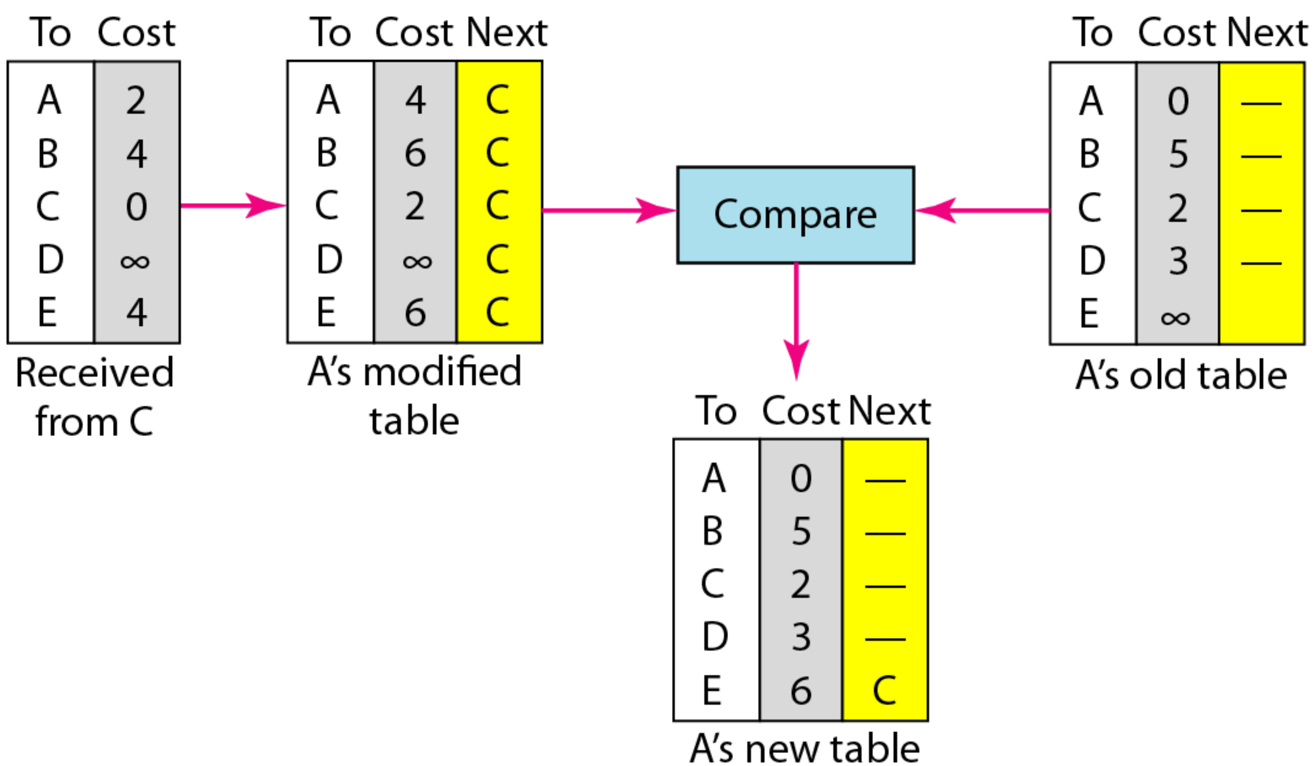
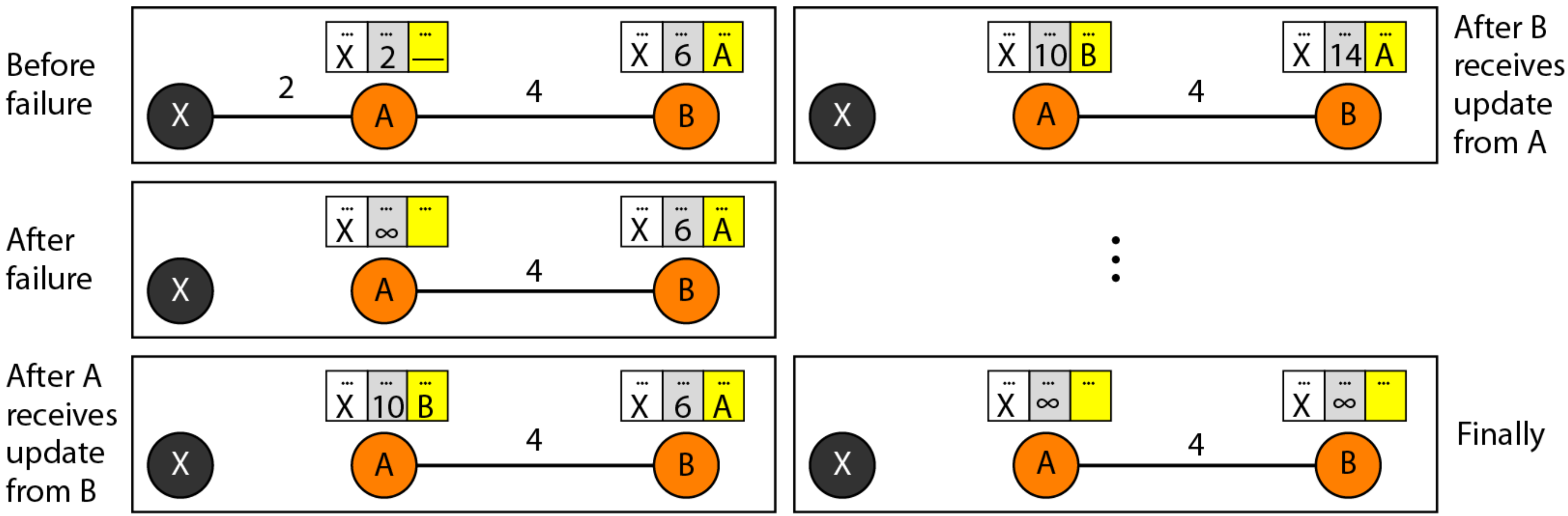


Figure 22.17 *Two-node instability*



Solution for Instability

- ***Defining Infinity :***
- 16 as infinity defined for the Distance vector routing (RIP)

Split Horizon: Each node sends only part of its table through the interface.

- B will not send the entry for X to A (since this info came from A)

- ***Split Horizon and Poison Reverse :***
  - If node does not get news about the route for some predefined time, the node deletes the route from its table (node A cant guess this is due to split horizon)
  - Poison Reverse : B can advertise the value for X but the distance it will mark as infinity for A → warning to A

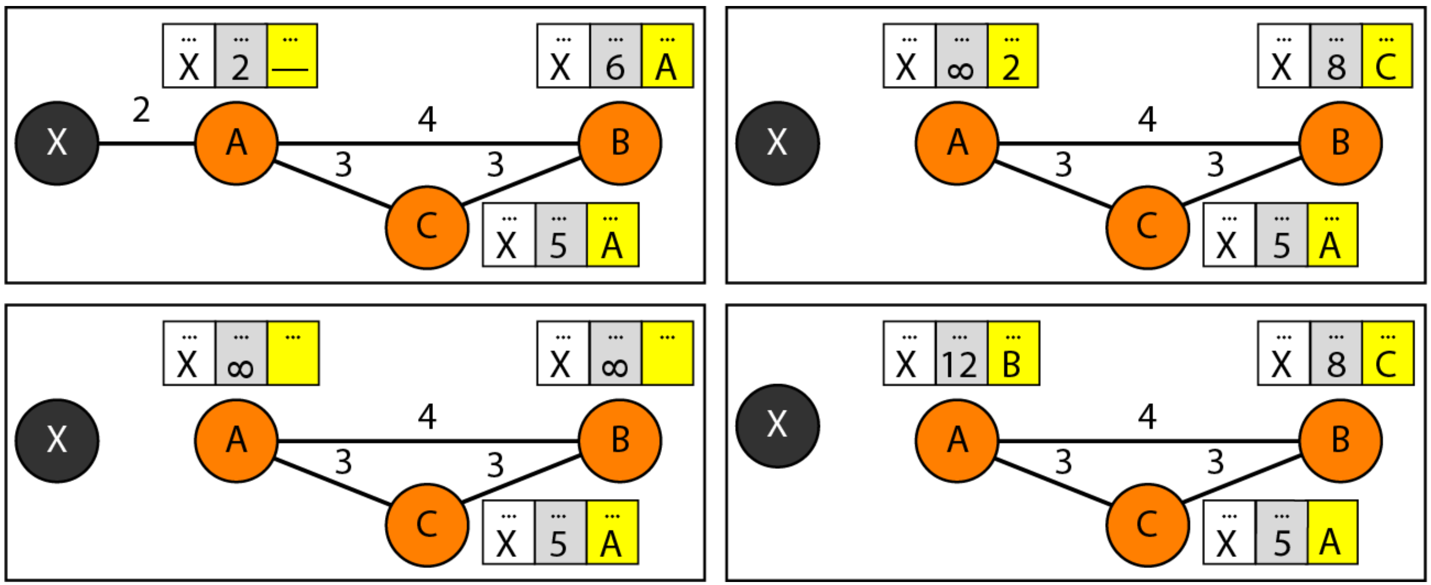
Figure 22.18 *Three-node instability*

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

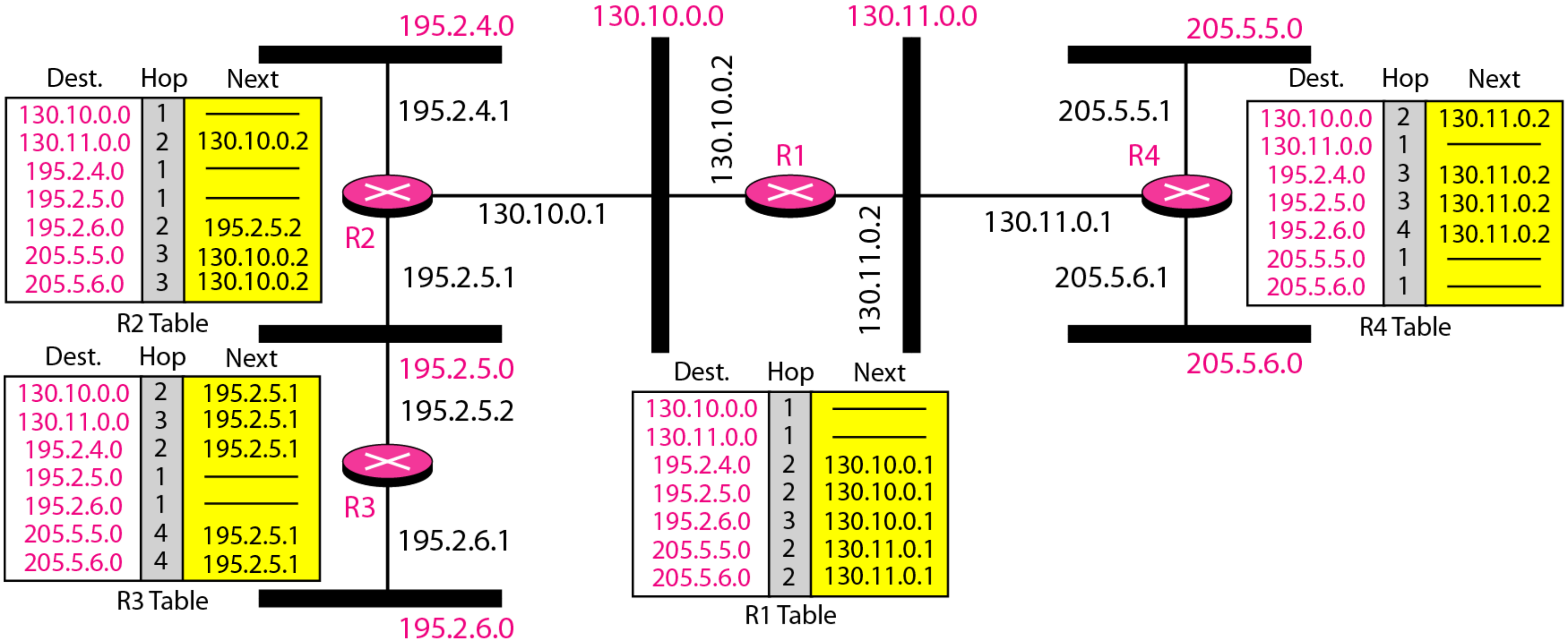


Routing Information Protocol (RIP)

• *Intra-domain Routing protocol used in autonomous system→ implements DV with following considerations*

1. *Routers have routing tables; networks do not*
2. *First column defines network address of destination network*
3. *Next node column - address of router to which packet needs to be sent to reach the destination*
4. *Metric: No. of networks to travel to reach the destination → hop-count*
5. *Infinity defined as 16*

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

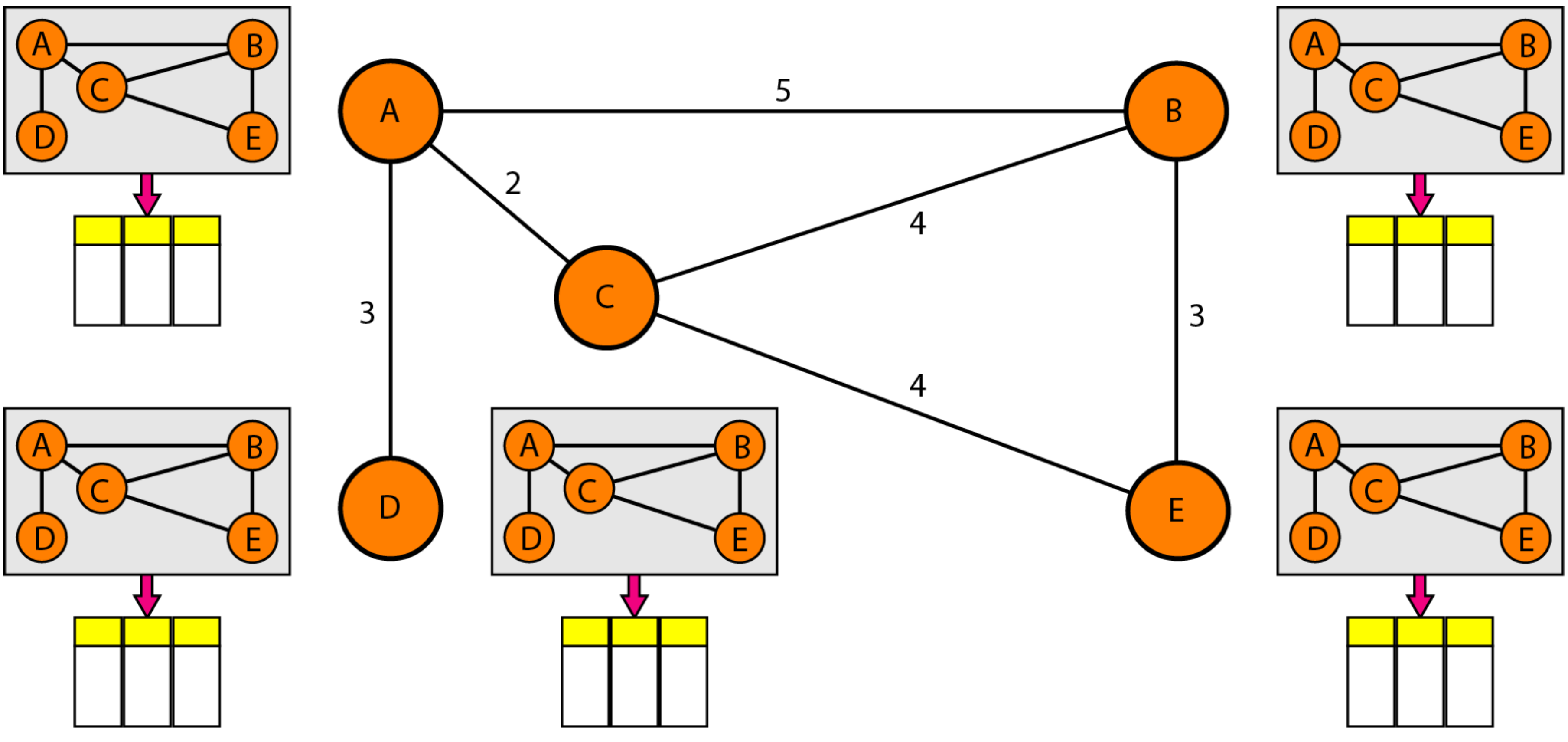
Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	—
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	—
205.5.6.0	1	—

R4 Table

Link State Routing

- *Follows the notion of state (type, condition, cost) of node/link*
- *No node has global knowledge of entire topology*
- *Each node has partial knowledge; knows the state of its (own) links*
- *Whole topology compiled from partial knowledge of each node*

Figure 22.20 *Concept of link state routing*

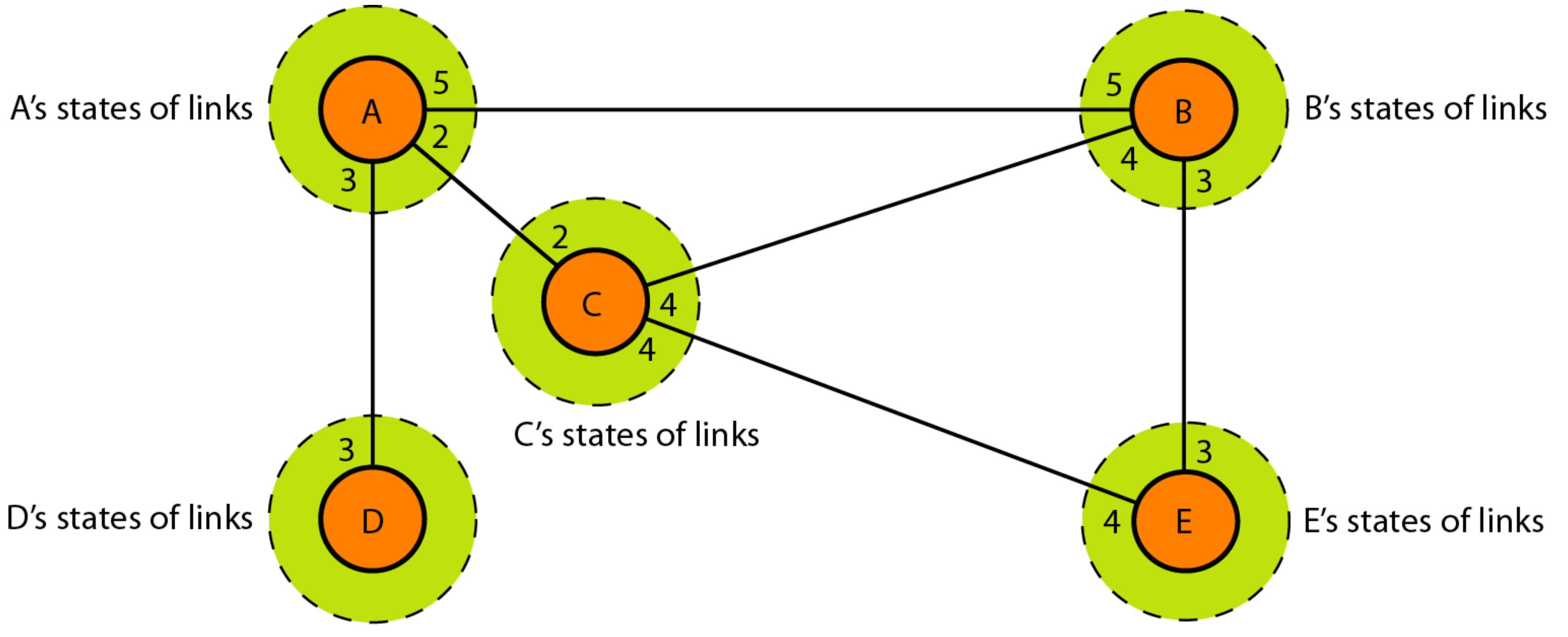


Steps in Link State Routing

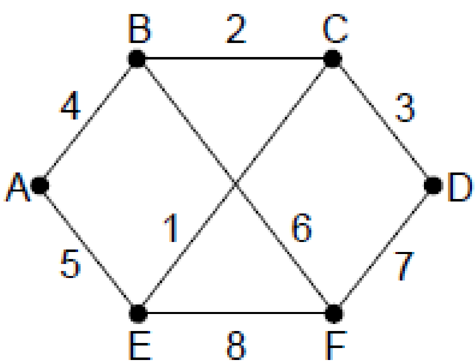
The idea behind link state routing be stated as five parts. Each router must do the following:

- ***Discover its neighbors and learn their network addresses***
- ***Measure the delay or cost to each of its neighbors Construct a packet telling all it has just learned (Creation of packet that has state of the links; LSP)***
- ***Send this packet to all other routers; LSP dissemination (Flooding in efficient and reliable way to other routers)***
- ***Compute the shortest path to every other router***
  - *Formation of a shortest path tree for each node.*
- ***Calculation of a routing table based on the shortest path tree***

Figure 22.21 *Link state knowledge*



Link State Packet and Flooding



(a)

Link State Packets

A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

(b)

Figure 22.22 *Dijkstra algorithm*

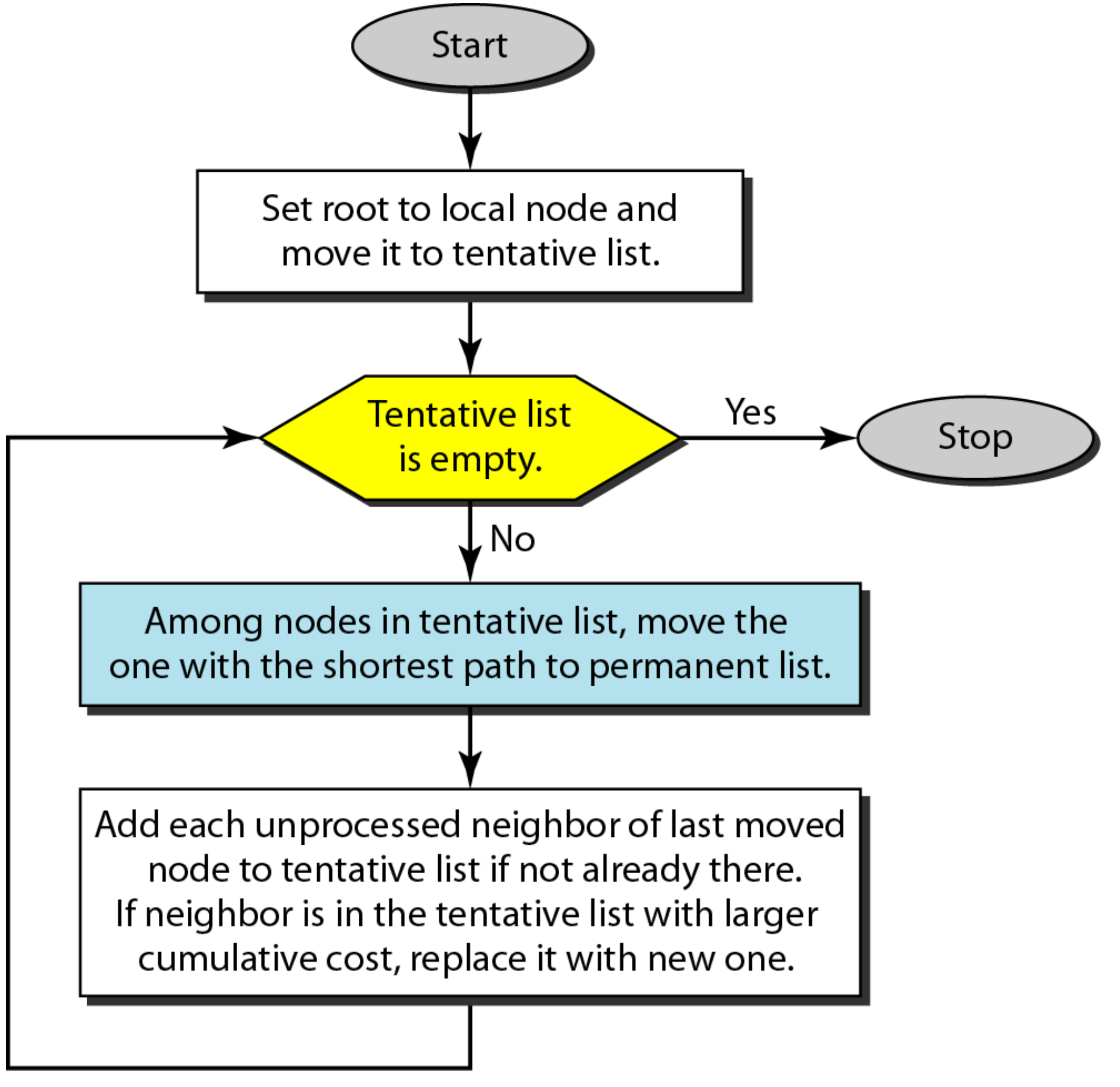
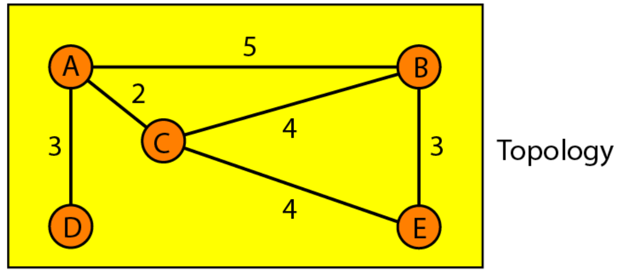
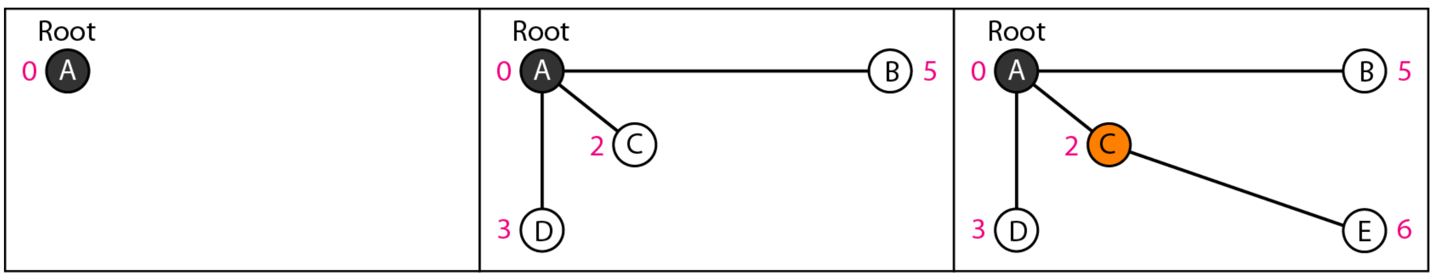


Figure 22.23 *Example of formation of shortest path tree*

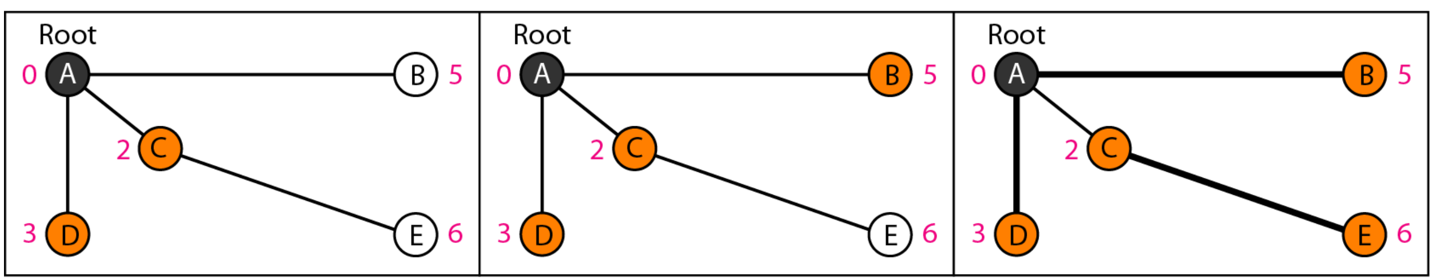




1. Set root to A and move A to tentative list.

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: Figure 22.24 *Areas in an autonomous system*

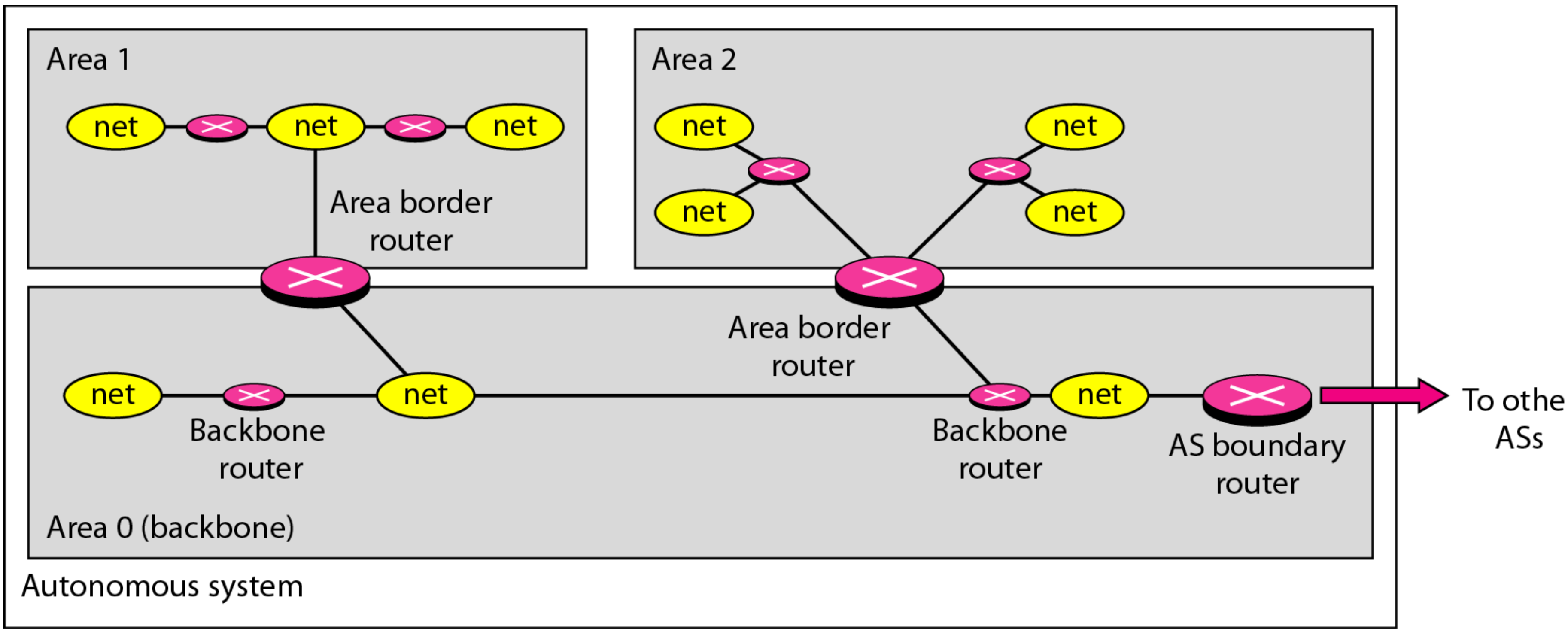


Figure 22.25 *Types of links*

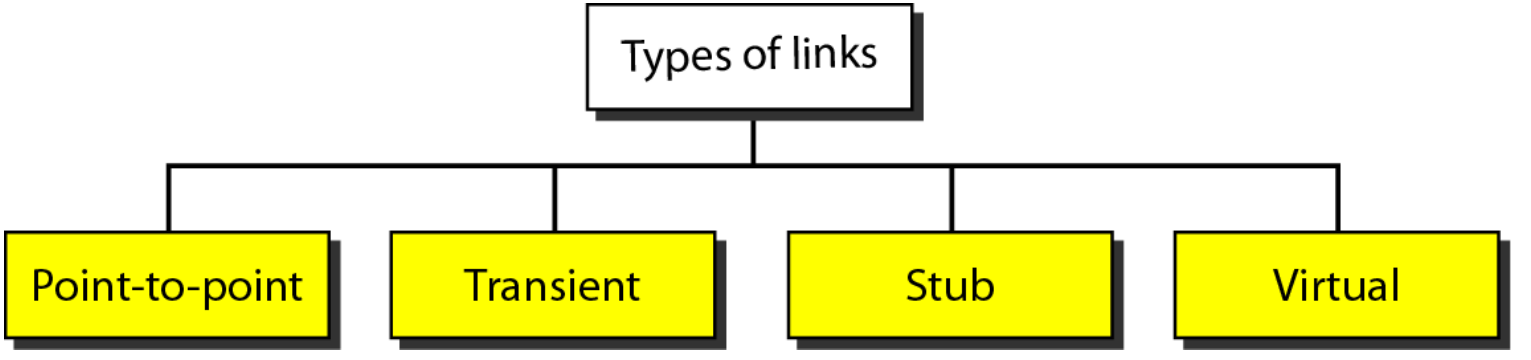


Figure 22.26 *Point-to-point link*

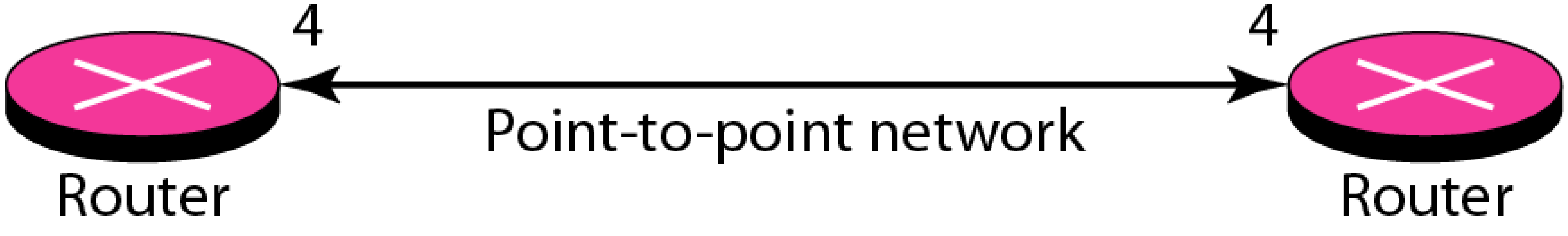


Figure 22.27 *Transient link*

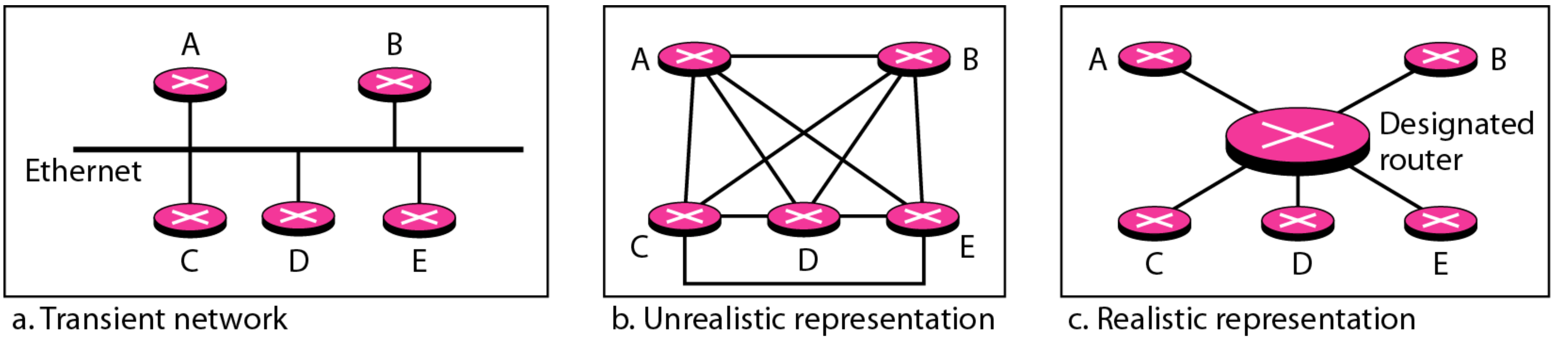


Figure 22.28 *Stub link*

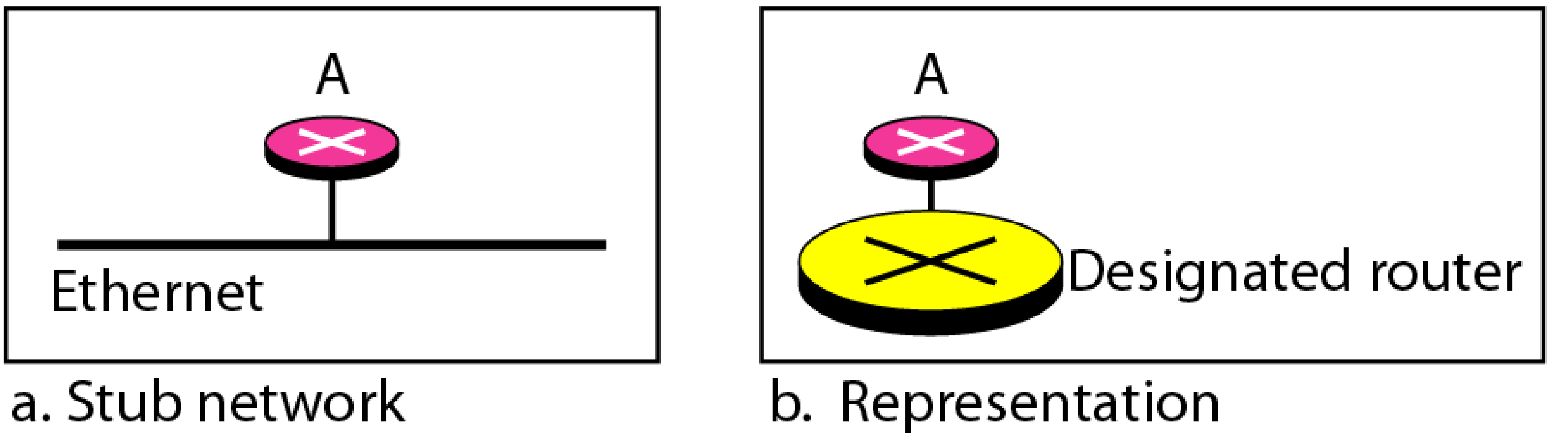
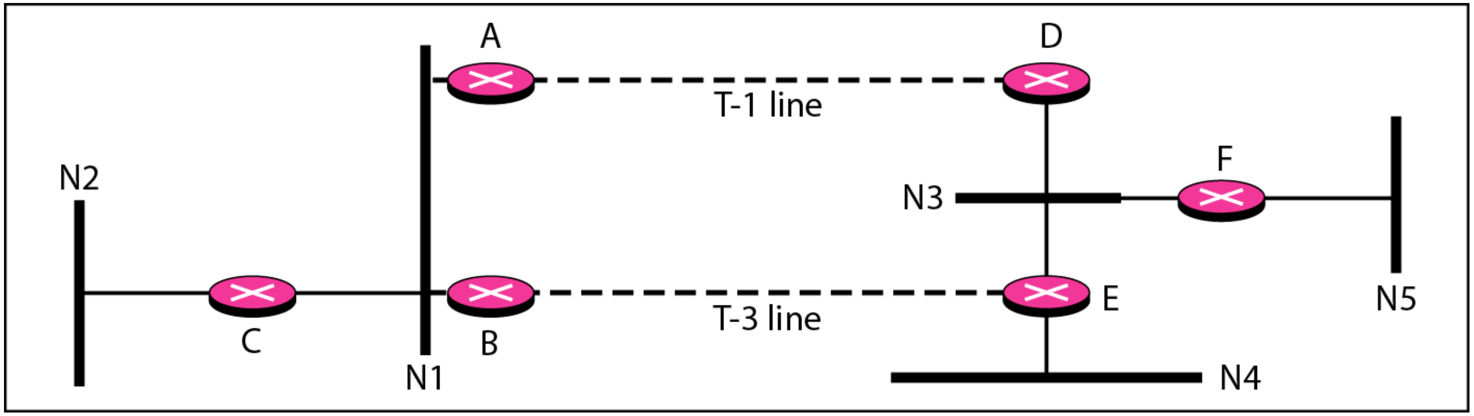
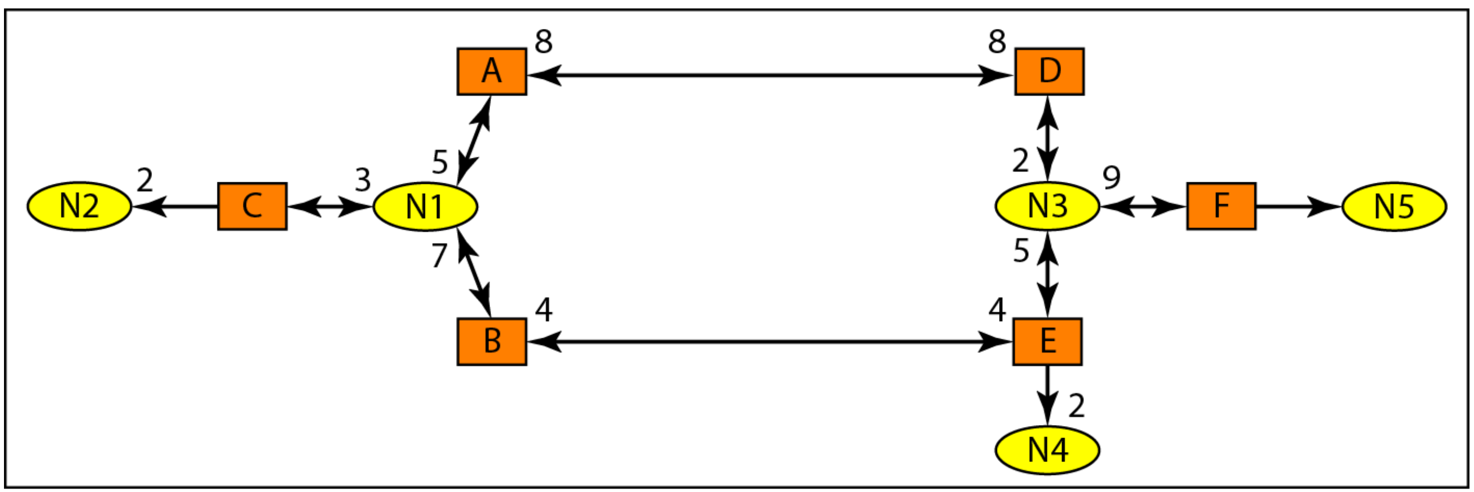


Figure 22.29 *Example of an AS and its graphical representation in OSPF*



a. Autonomous system



b. Graphical representation

Figure 22.30 *Initial routing tables in path vector routing*

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

A1 Table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

C1 Table

Dest.	Path
D1	AS4
D2	AS4
D3	AS4
D4	AS4

D1 Table

Dest.	Path
B1	AS2
B2	AS2
B3	AS2
B4	AS2

B1 Table

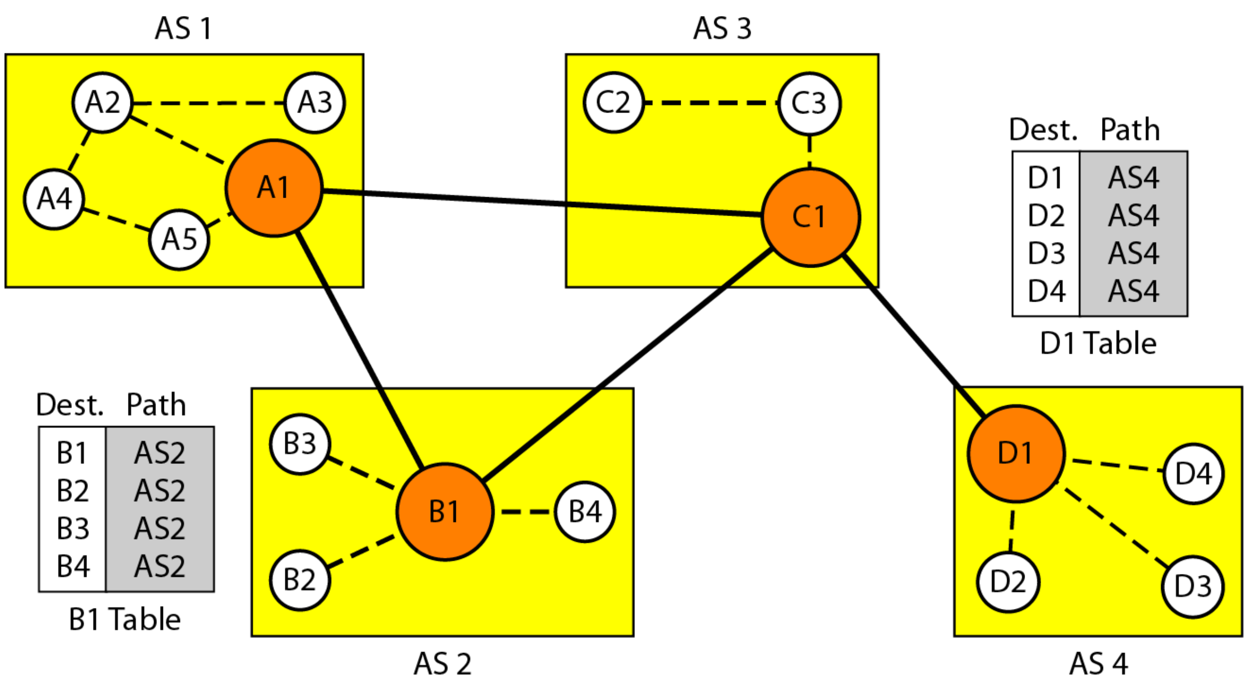


Figure 22.31 *Stabilized tables for three autonomous systems*

A1 Table

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

B1 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

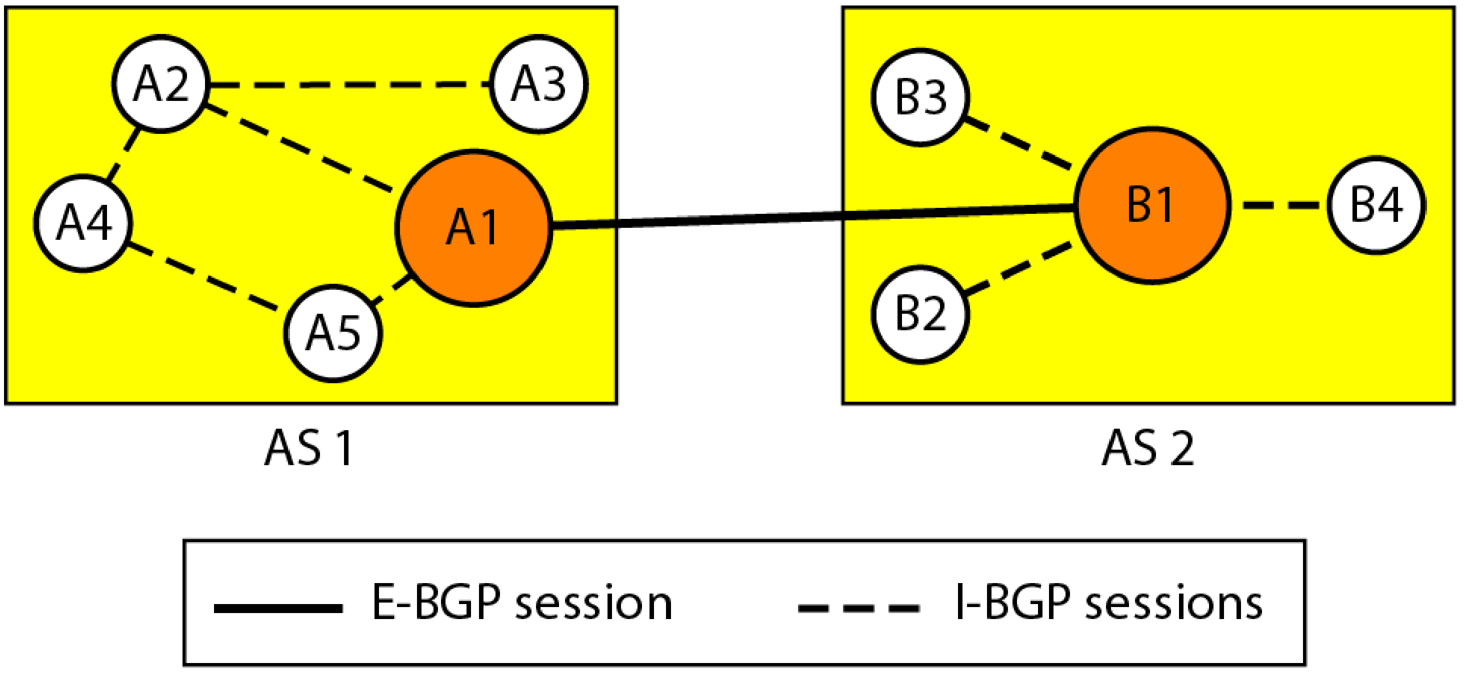
C1 Table

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

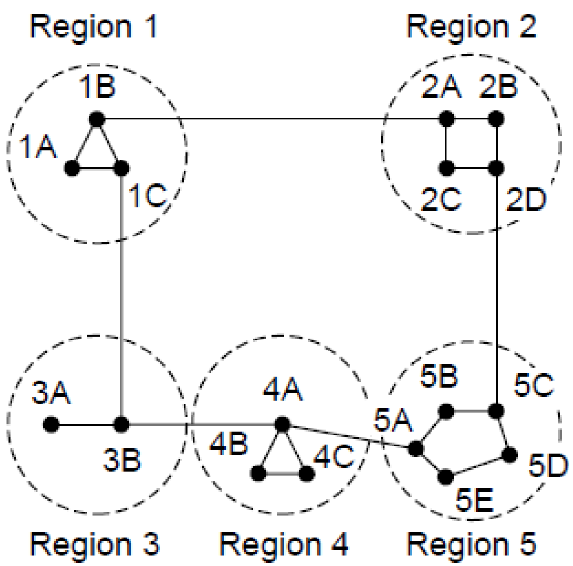
D1 Table

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

Figure 22.32 *Internal and external BGP sessions*



Hierarchical Routing



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

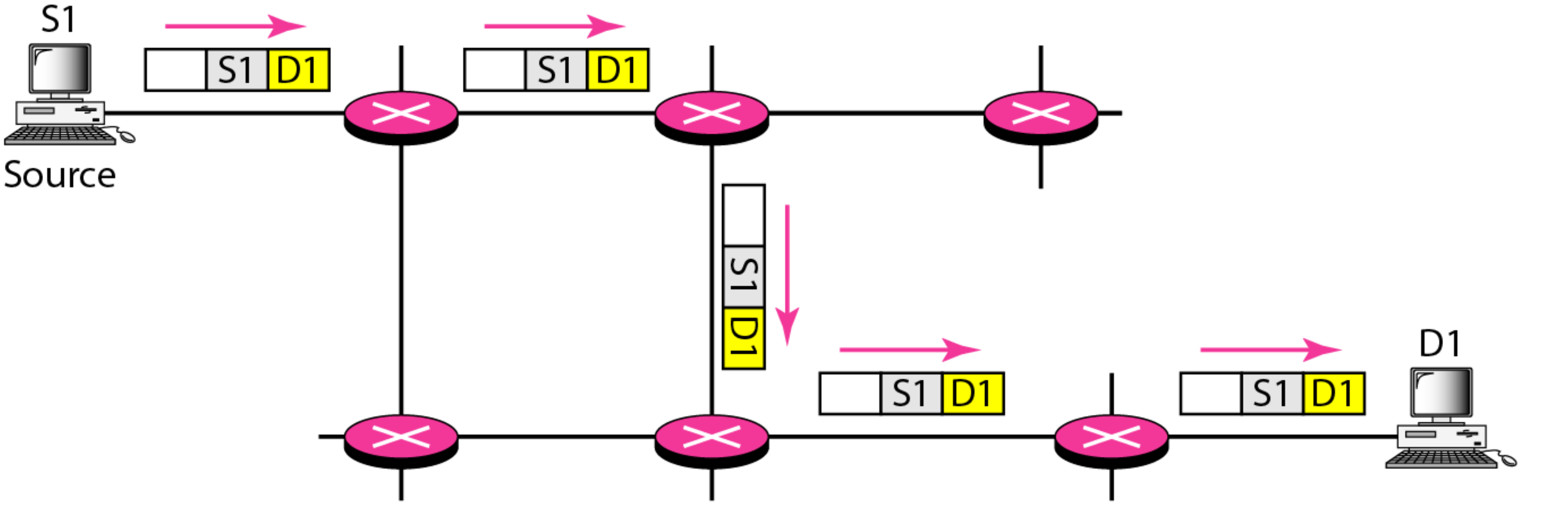
(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

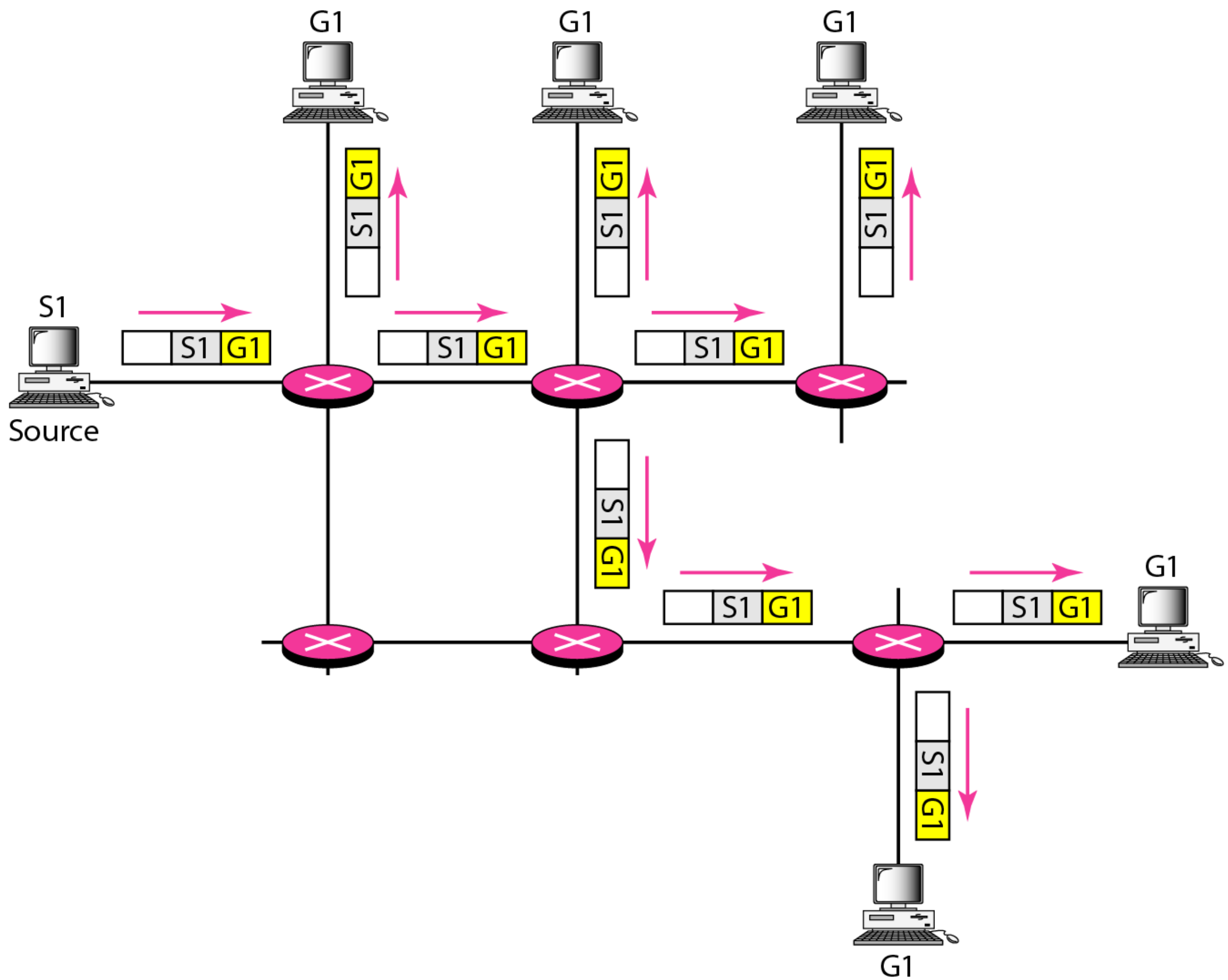
Figure 22.33 *Unicasting*



Note

In unicasting, the router forwards the received packet through only one of its interfaces.

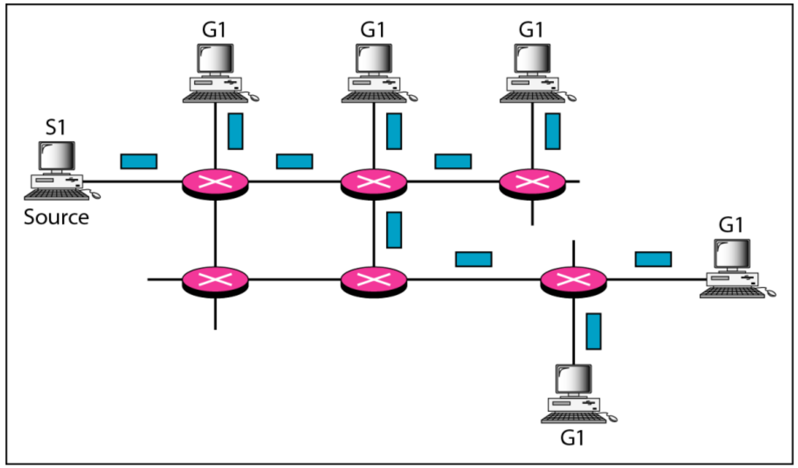
Figure 22.34 *Multicasting*



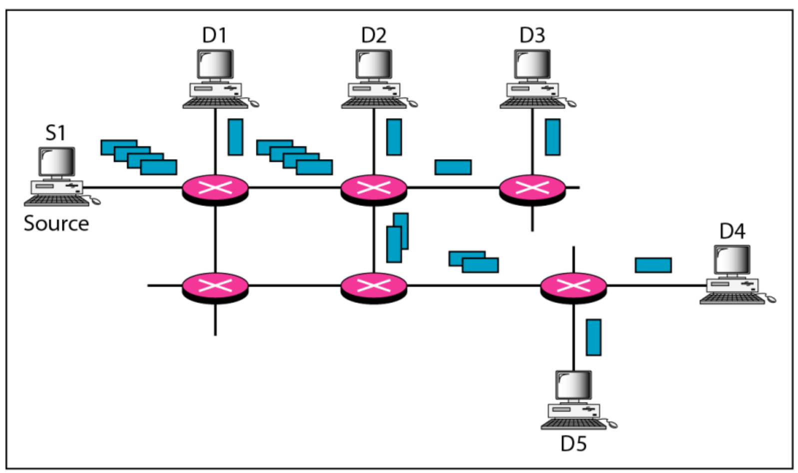
Note

In multicasting, the router may forward the received packet through several of its interfaces.

Figure 22.35 *Multicasting versus multiple unicasting*



a. Multicasting



b. Multiple unicasting

22-4 MULTICAST ROUTING PROTOCOLS

In this section, we discuss multicasting and multicast routing protocols.

<u>Topics discussed in this section:</u>

Unicast, Multicast, and Broadcast
Applications
Multicast Routing
Routing Protocols

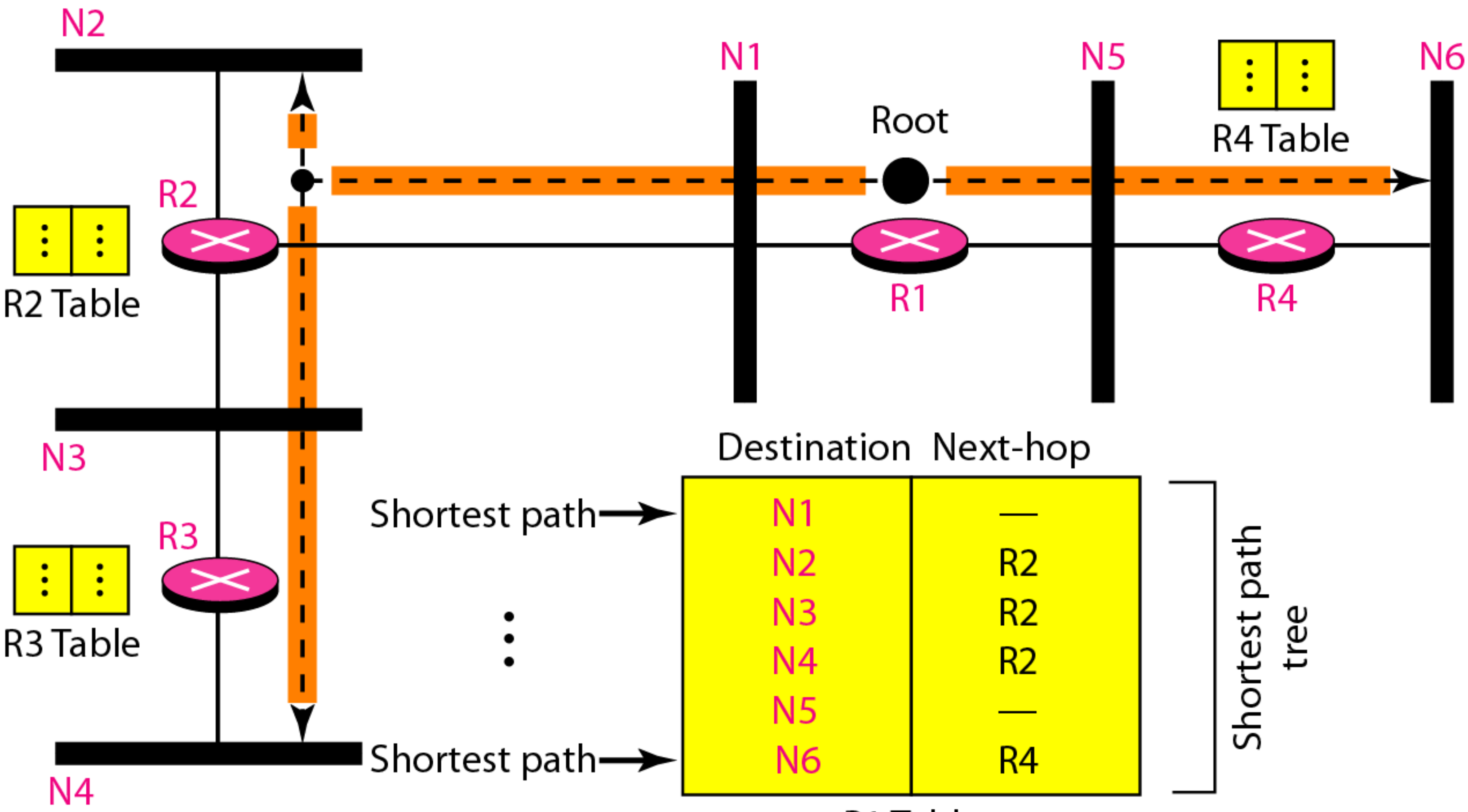
Note

Emulation of multicasting through multiple unicasting is not efficient and may create long delays, particularly with a large group.

Note

In unicast routing, each router in the domain has a table that defines a shortest path tree to possible destinations.

Figure 22.36 *Shortest path tree in unicast routing*



Shortest path →

Destination	Next-hop
N1	—
N2	R2
N3	R2
N4	R2
N5	—
N6	R4

Shortest path →

R1 Table

Shortest path tree

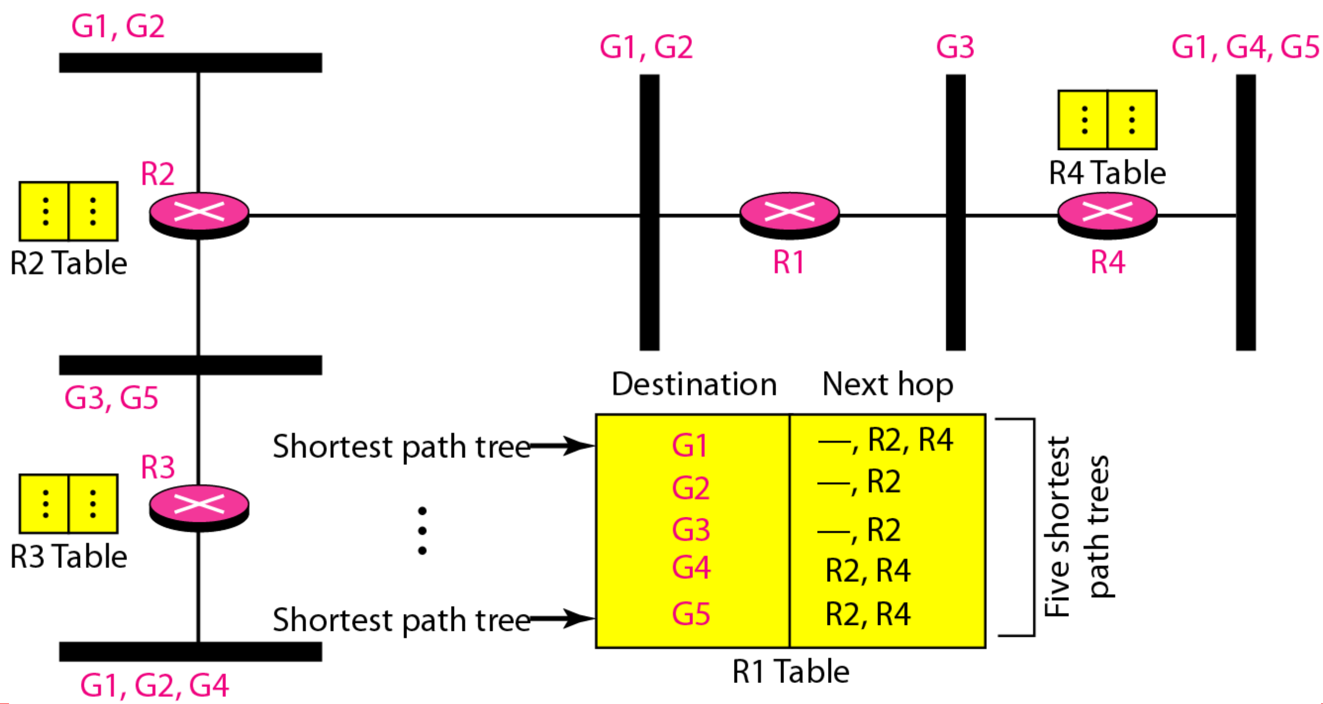
Note

In multicast routing, each involved router needs to construct a shortest path tree for each group.

Figure 22.37 *Source-based tree approach*

-Each Router needs to have one shortest path tree for each group

-Shortest path tree for a group defines the next hop for each network that has loyal members for that group



Note

In the source-based tree approach, each router needs to have one shortest path tree for each group

If no. of groups = *m*; each router needs to have *m* shortest path trees

Group-Shared Tree approach

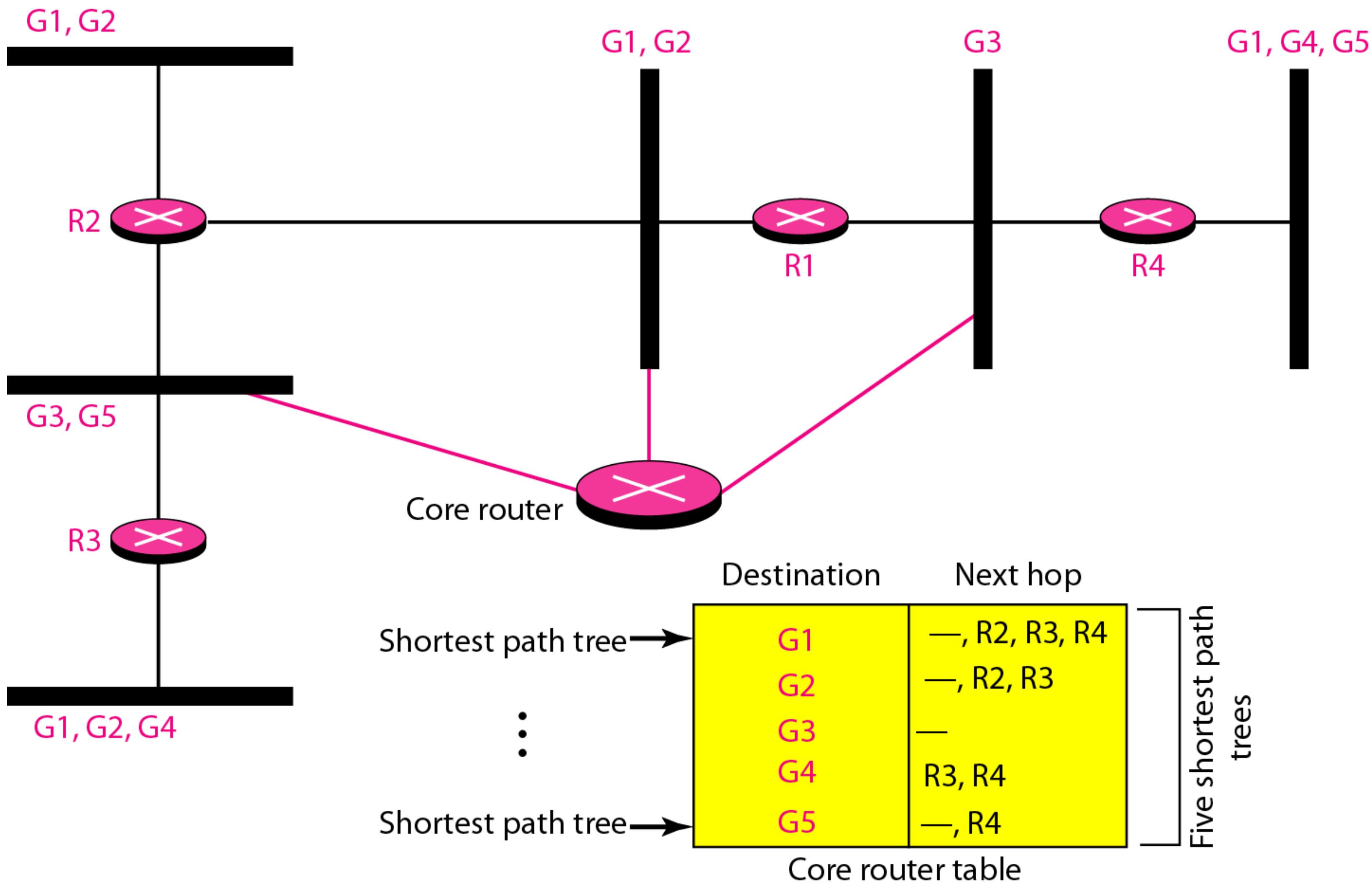
-In Group-shared tree approach, instead of each router having m shortest path trees, only 1 designated router→ the center core or rendezvous router responsible for distributing multicast traffic

-It has m shortest path trees in its routing table; rest of the routers have none

-If a router receives a multicast pkt, it encapsulates it into unicast pkt and sends to core

- Core router de-capsulates and removes the multicast pkt; consults it routing table to route it

Figure 22.38 ***Group-shared tree approach***



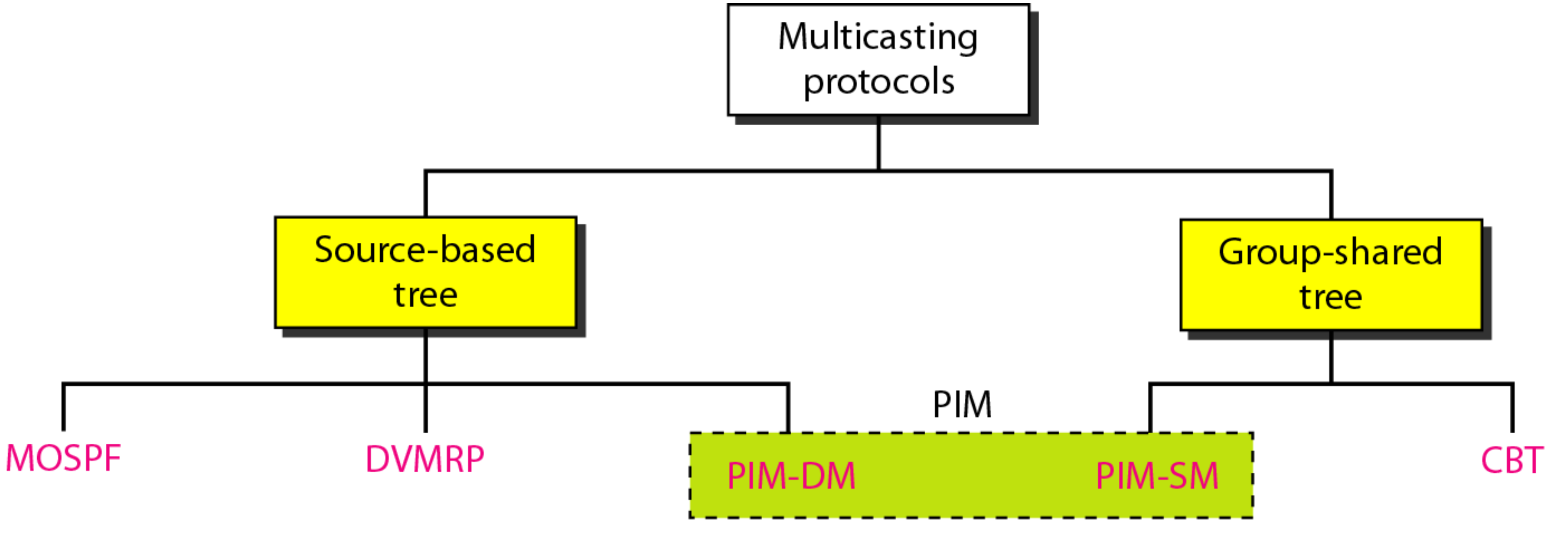
Destination	Next hop
G1	—, R2, R3, R4
G2	—, R2, R3
G3	—
G4	R3, R4
G5	—, R4

Core router table

Note

In the group-shared tree approach, only the core router, which has a shortest path tree for each group, is involved in multicasting.

Figure 22.39 *Taxonomy of common multicast protocols*



Note

DVMRP

Multicast distance vector routing uses the source-based tree approach.

Note

Flooding broadcasts packets, but creates loops in the systems.

Note

RPF eliminates the loop in the flooding process
Only one copy is forwarded (only the one that travelled the shortest path); others are dropped

Figure 22.40 *Reverse path forwarding (RPF)*

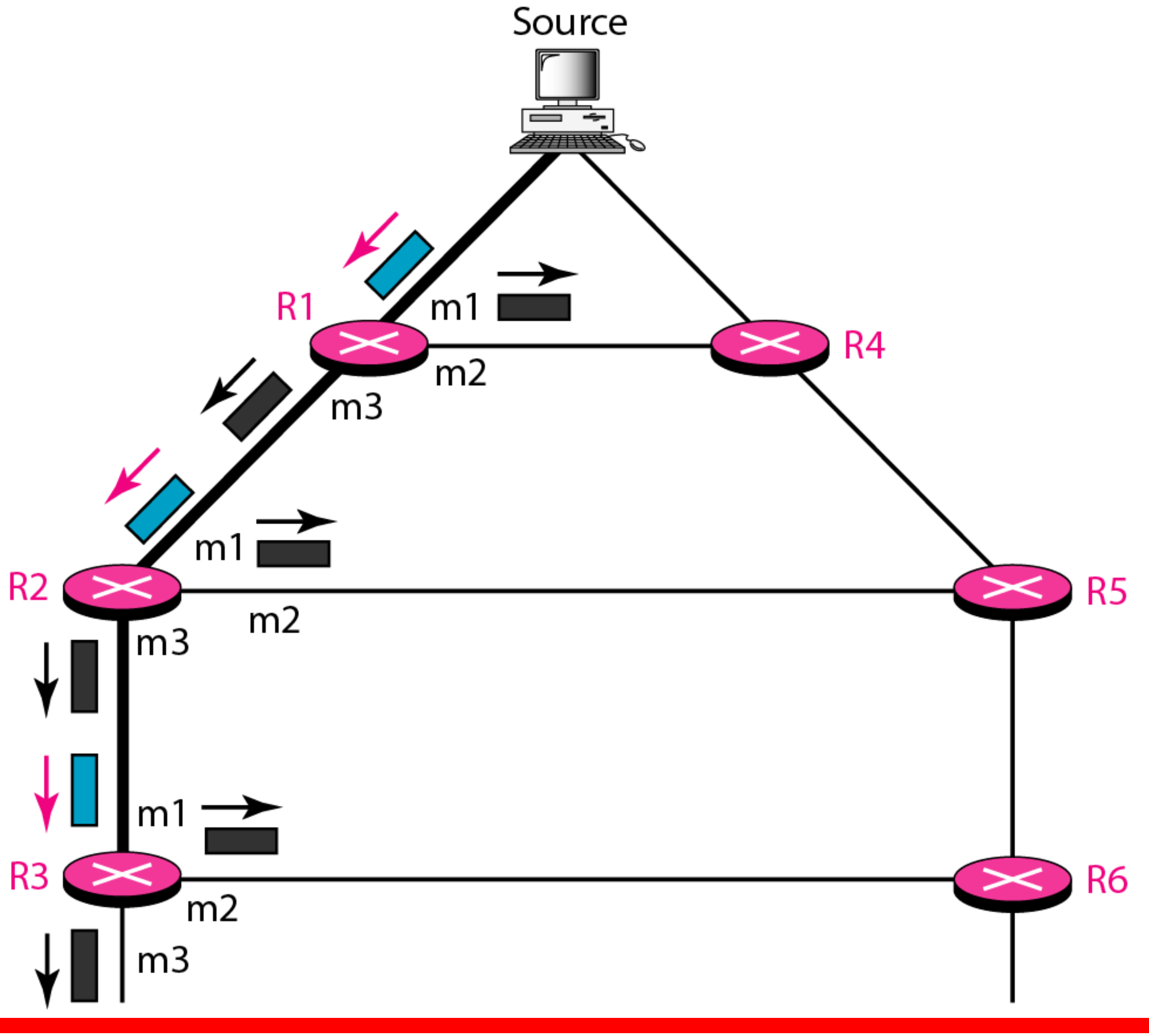


Figure 22.41 *Problem with RPF*

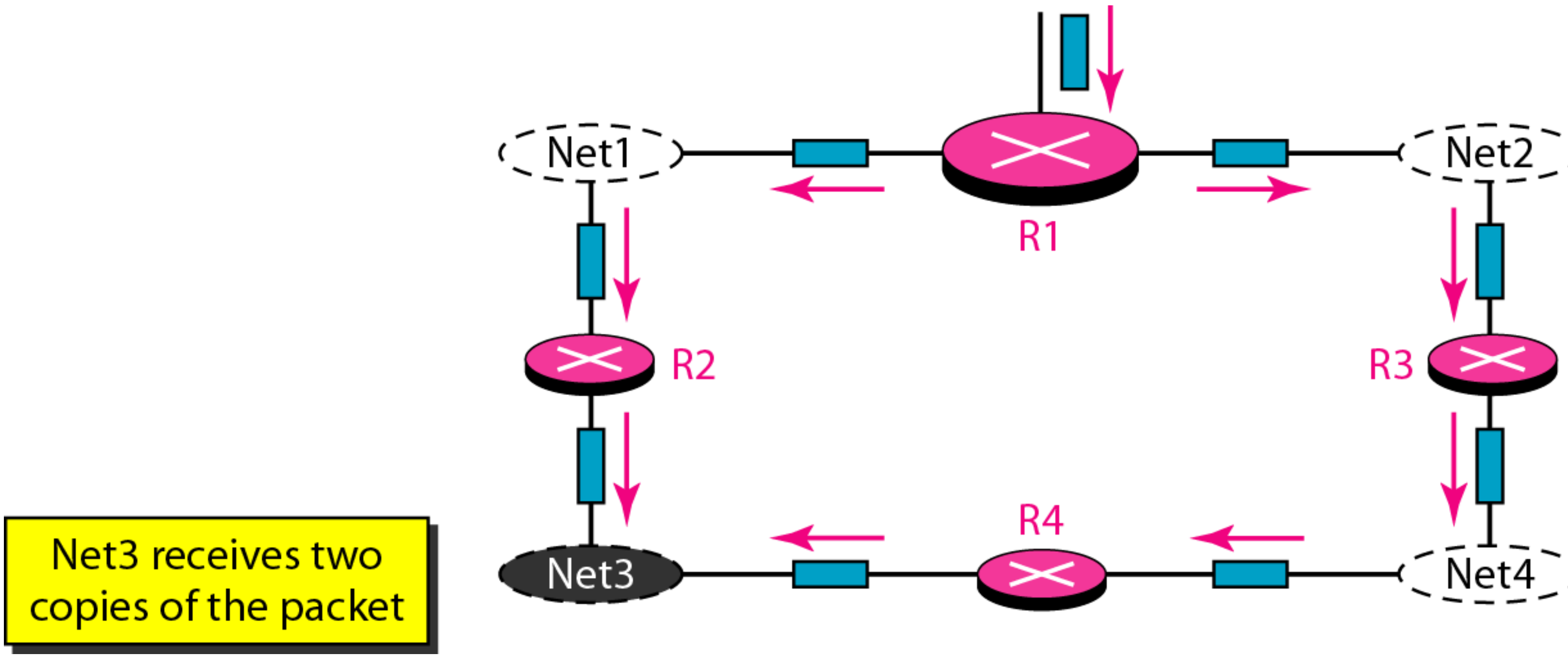
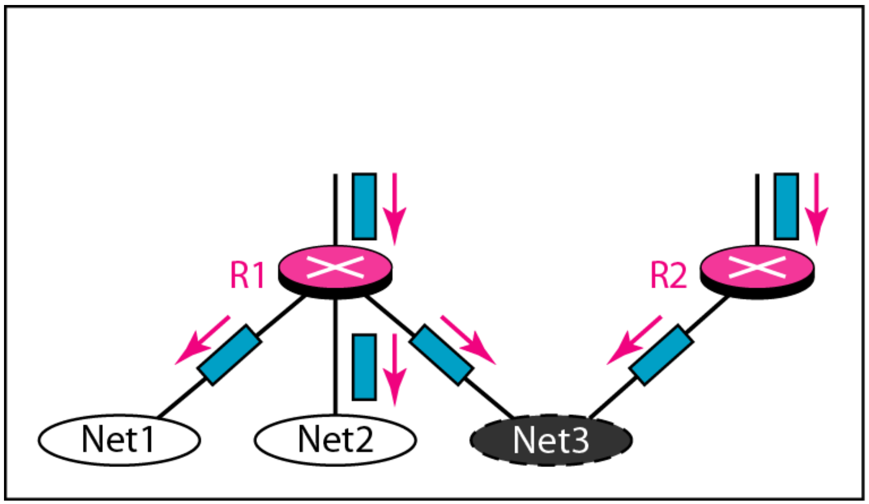
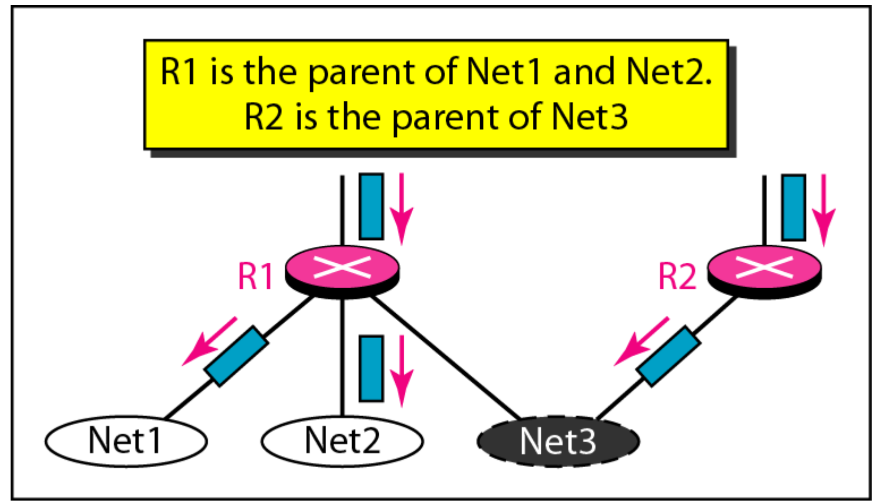


Figure 22.42 *RPF Versus RPB*



a. RPF

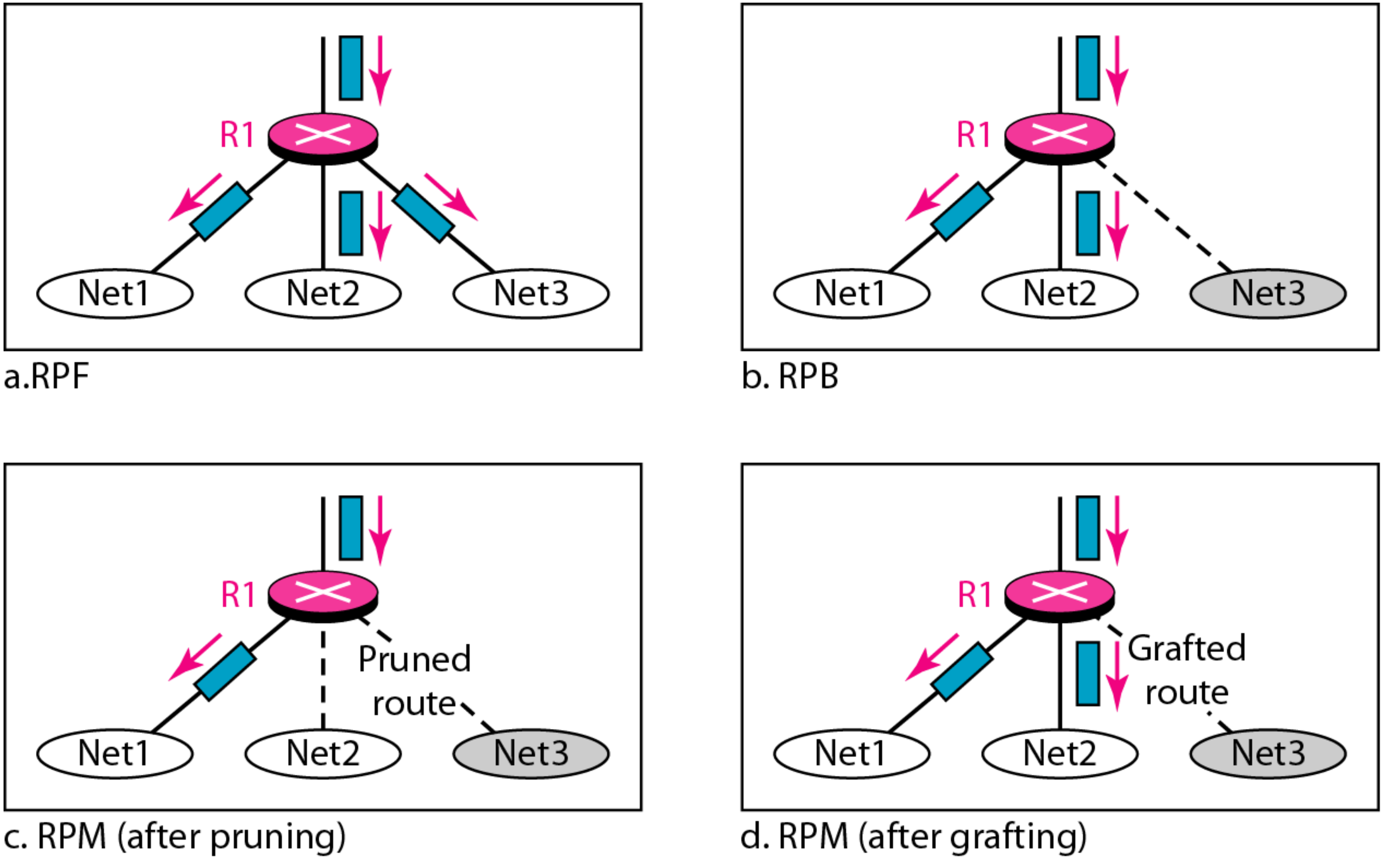


b. RPB

Note

RPB creates a shortest path broadcast tree from the source to each destination. It guarantees that each destination receives one and only one copy of the packet.

Figure 22.43 *RPF, RPB, and RPM*



Note

RPM adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

Figure 22.44 *Group-shared tree with rendezvous router*

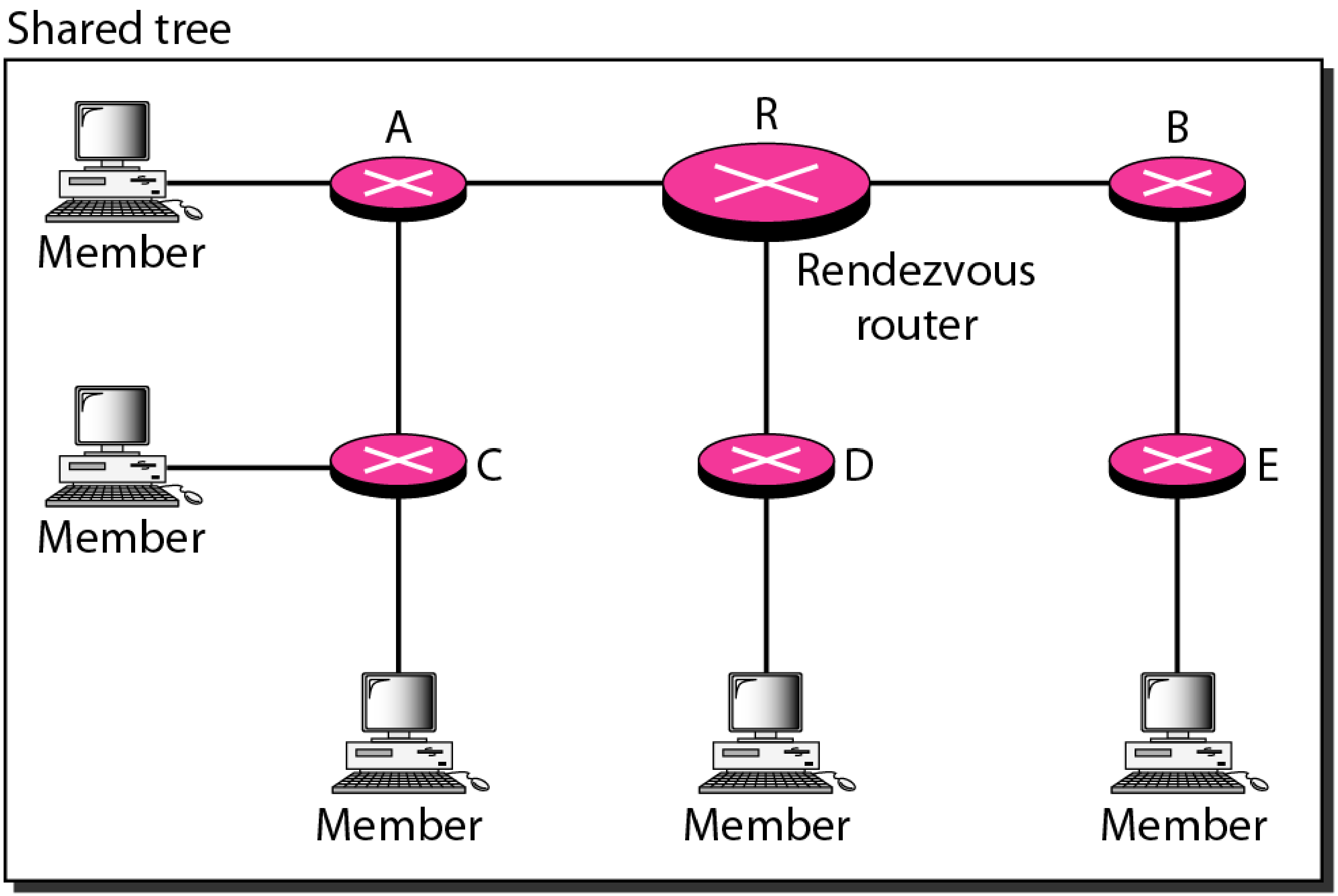
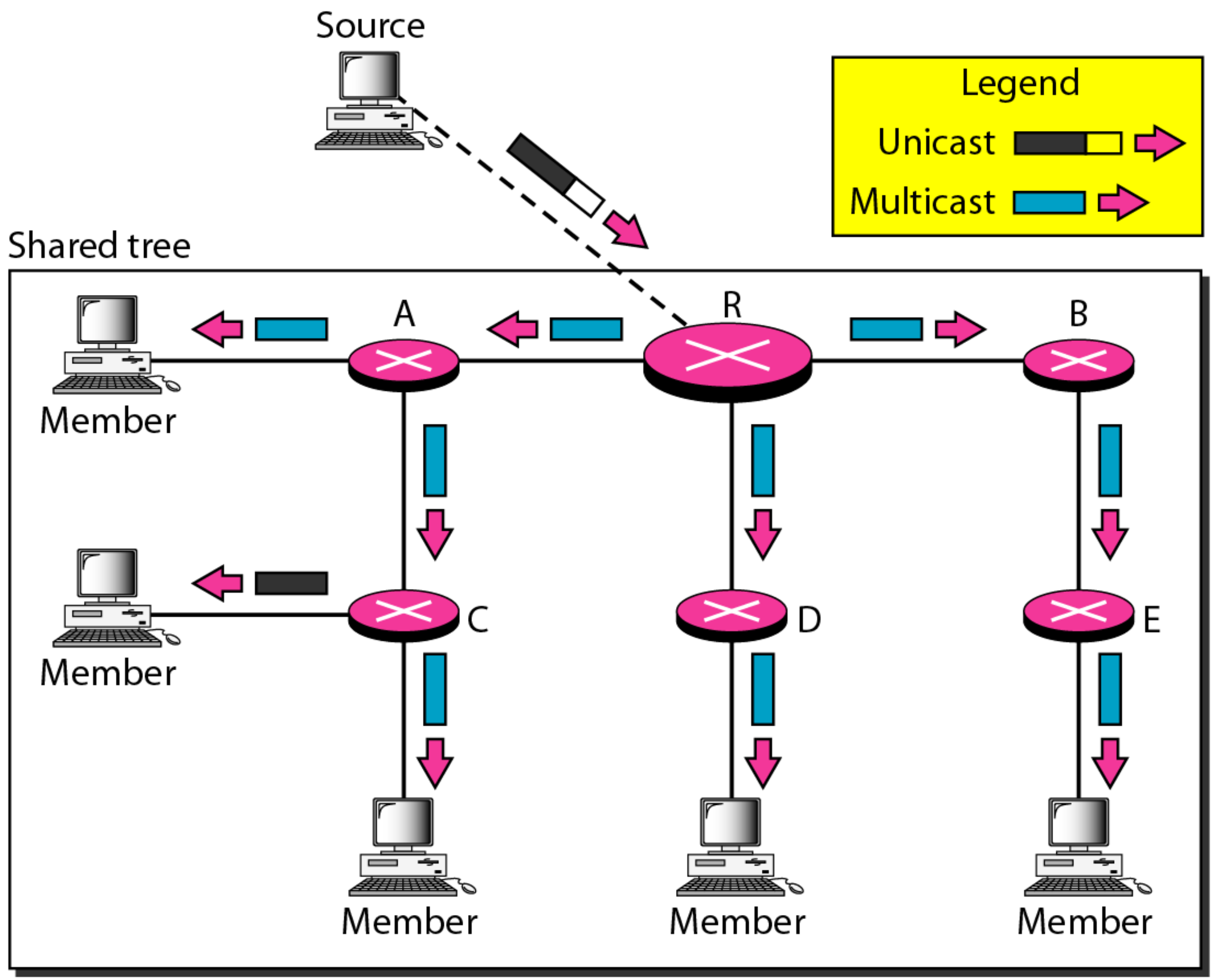


Figure 22.45 *Sending a multicast packet to the rendezvous router*



Note

In CBT, the source sends the multicast packet (encapsulated in a unicast packet) to the core router. The core router decapsulates the packet and forwards it to all interested interfaces.