Data Communications and Networking Fourth Edition

Forouzan

Chapter 22

Network Layer: Delivery, Forwarding, and Routing

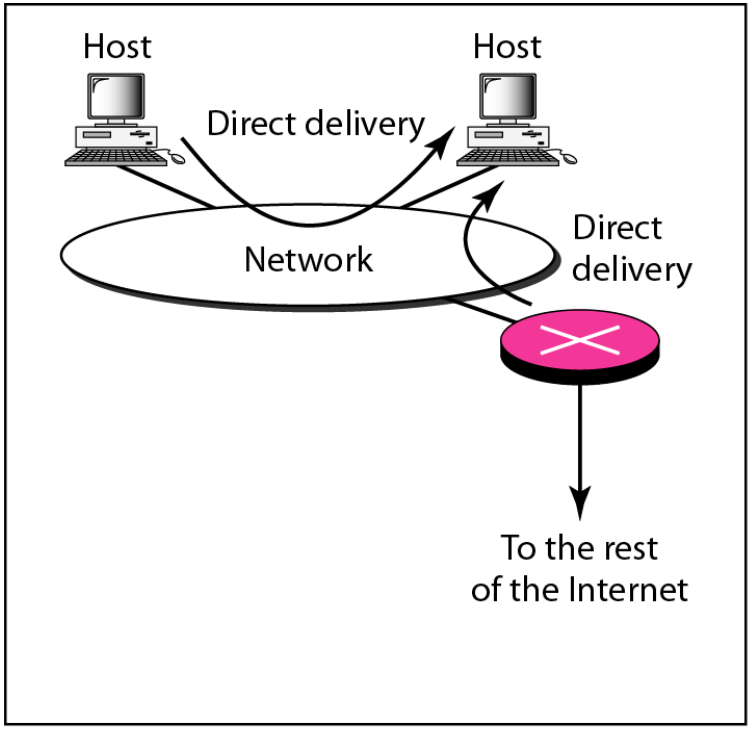
22-1 DELIVERY

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

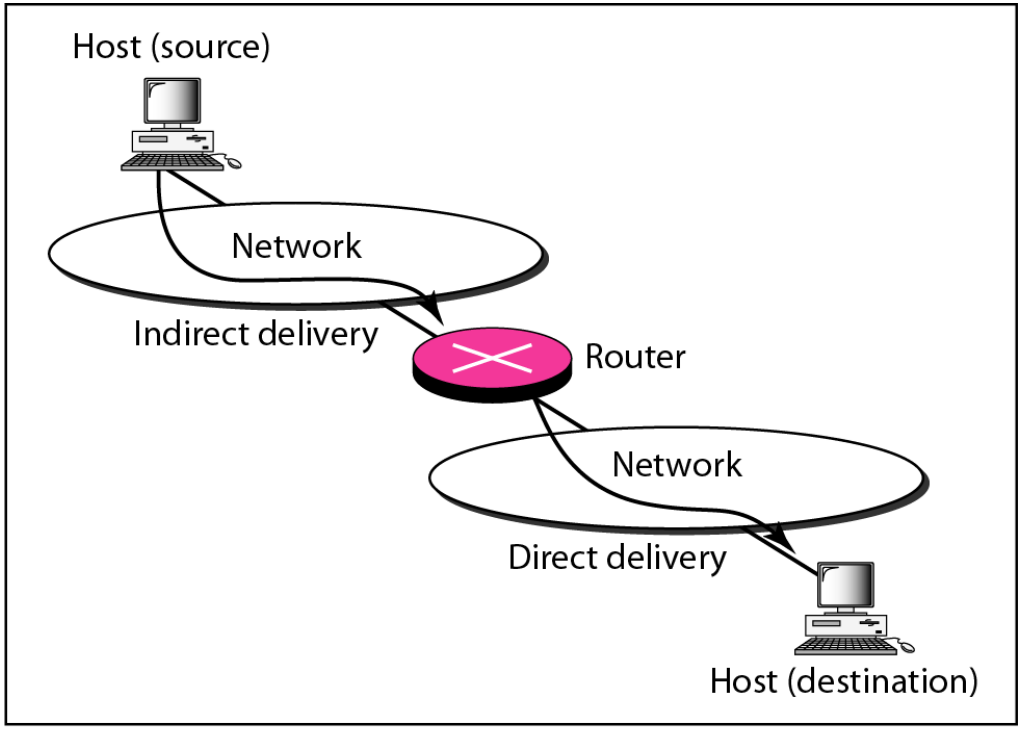
Topics discussed in this section:

Direct Versus Indirect Delivery

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

Figure 22.2 *Route method versus next-hop method*

a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table for host A

Destination	Route
Host B	R2, host B

Routing table for R1

Destination	Route
Host B	Host B

Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---

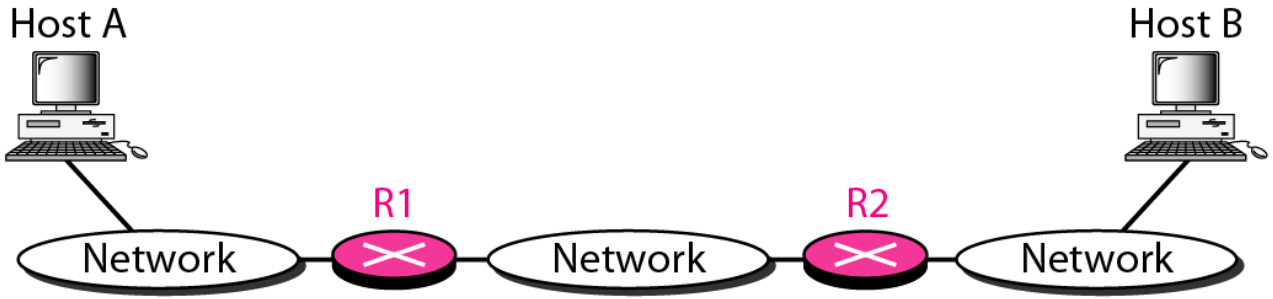


Figure 22.3 *Host-specific versus network-specific method*

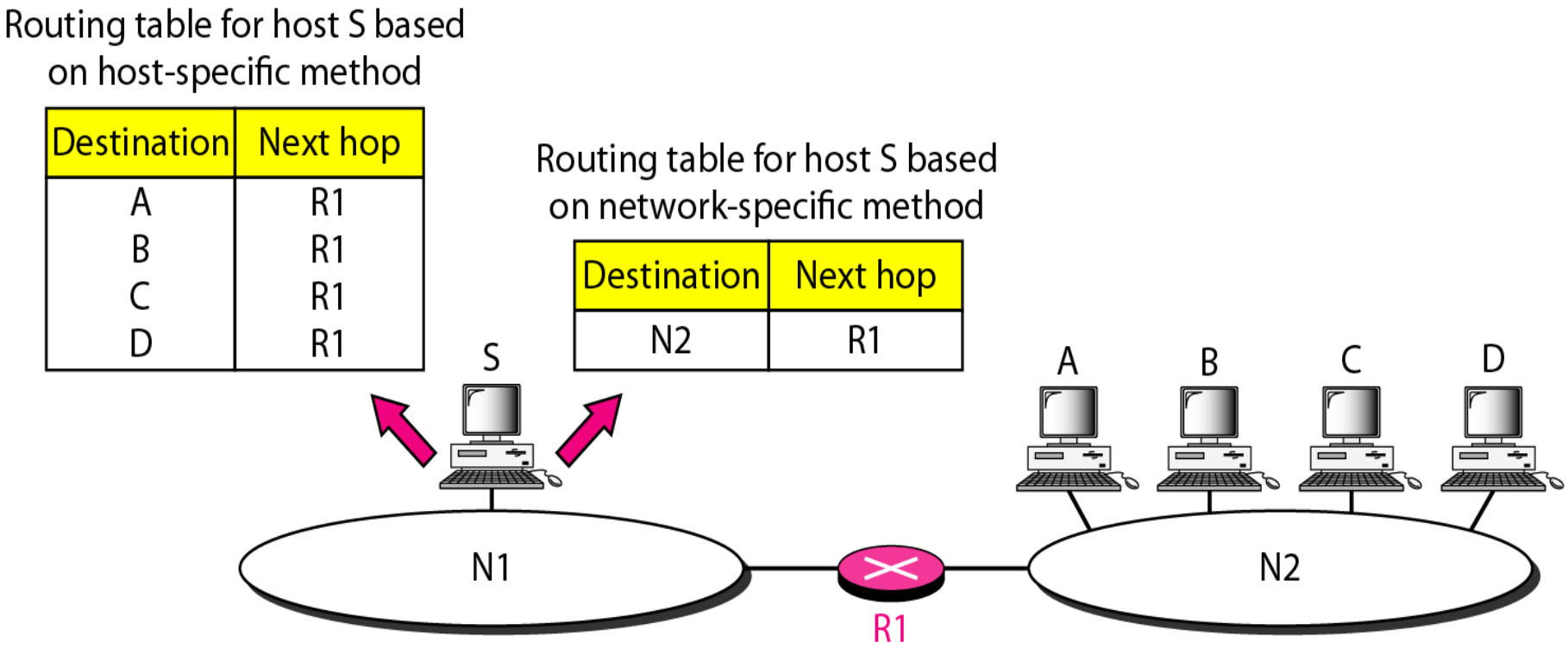


Figure 22.4 *Default method*

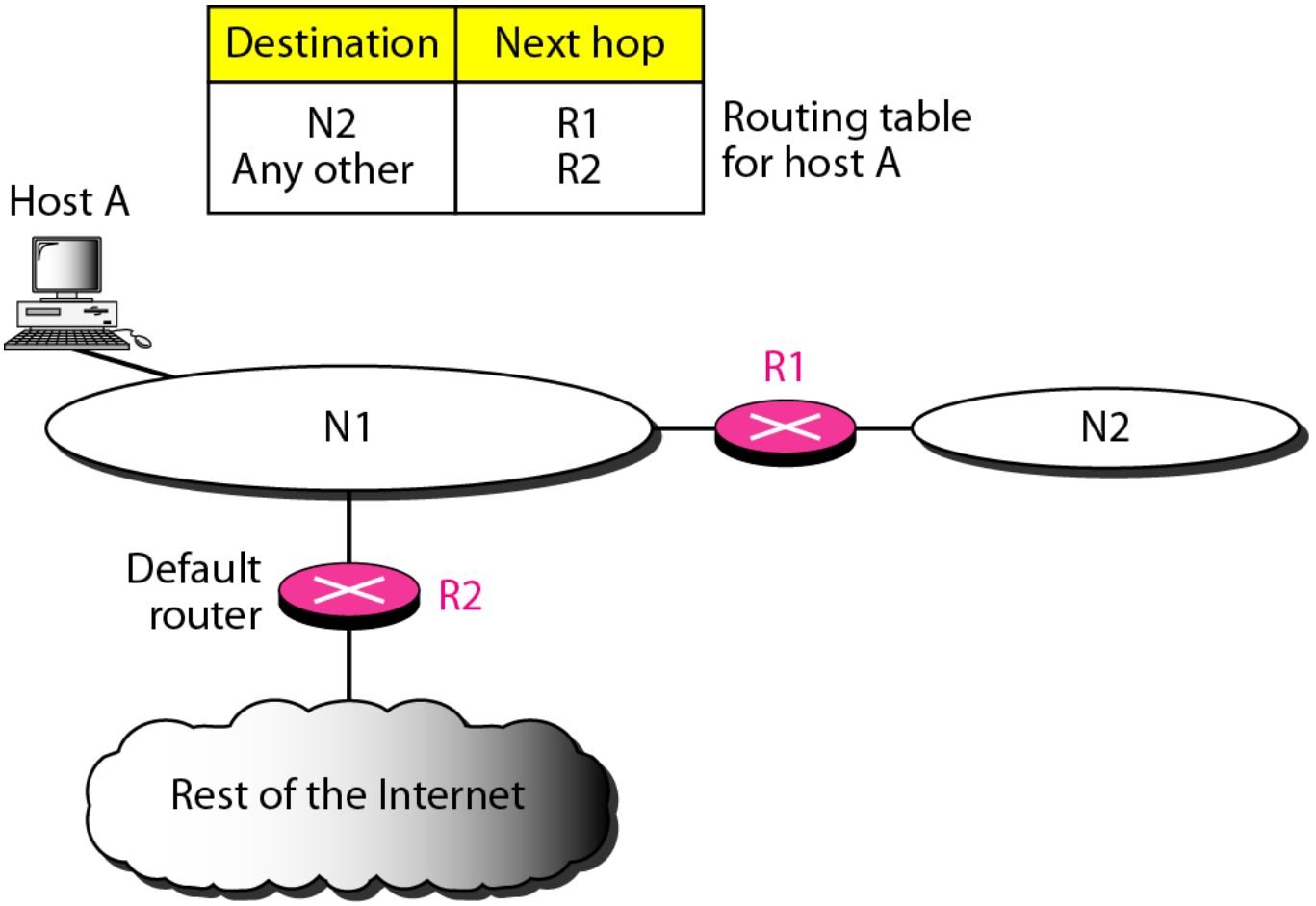
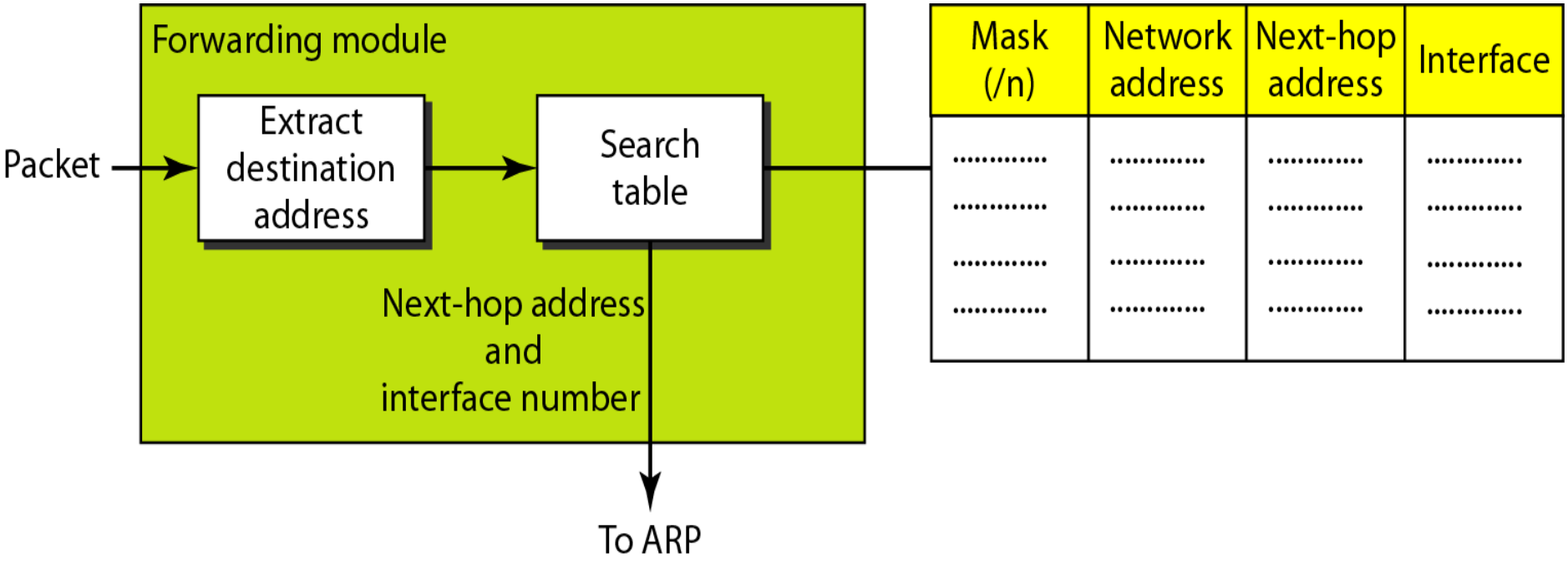


Figure 22.5 *Simplified forwarding module in classless address*



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.

Figure 22.6 *Configuration for Example 22.1*

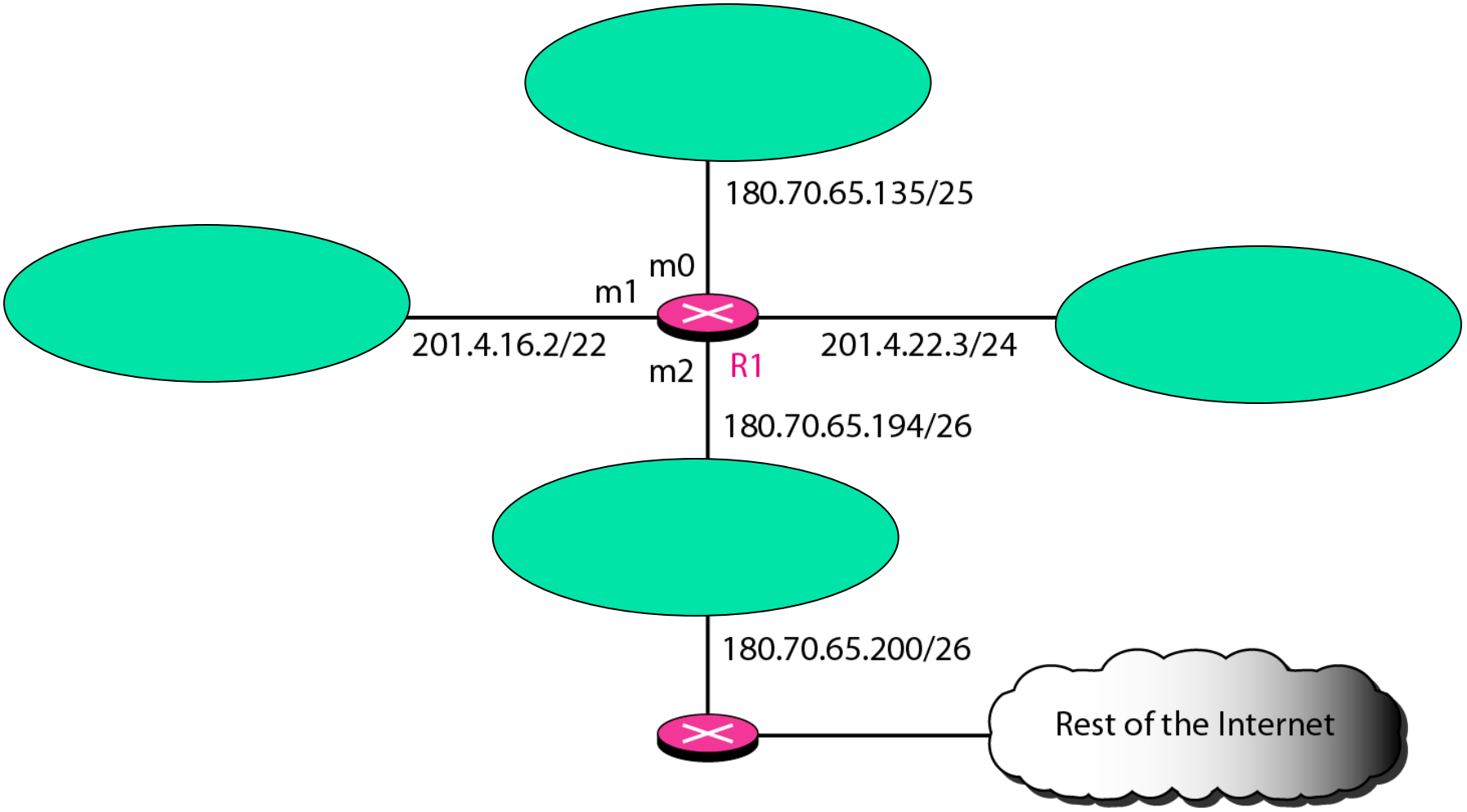


Figure 22.6 *Configuration for Example 22.1*

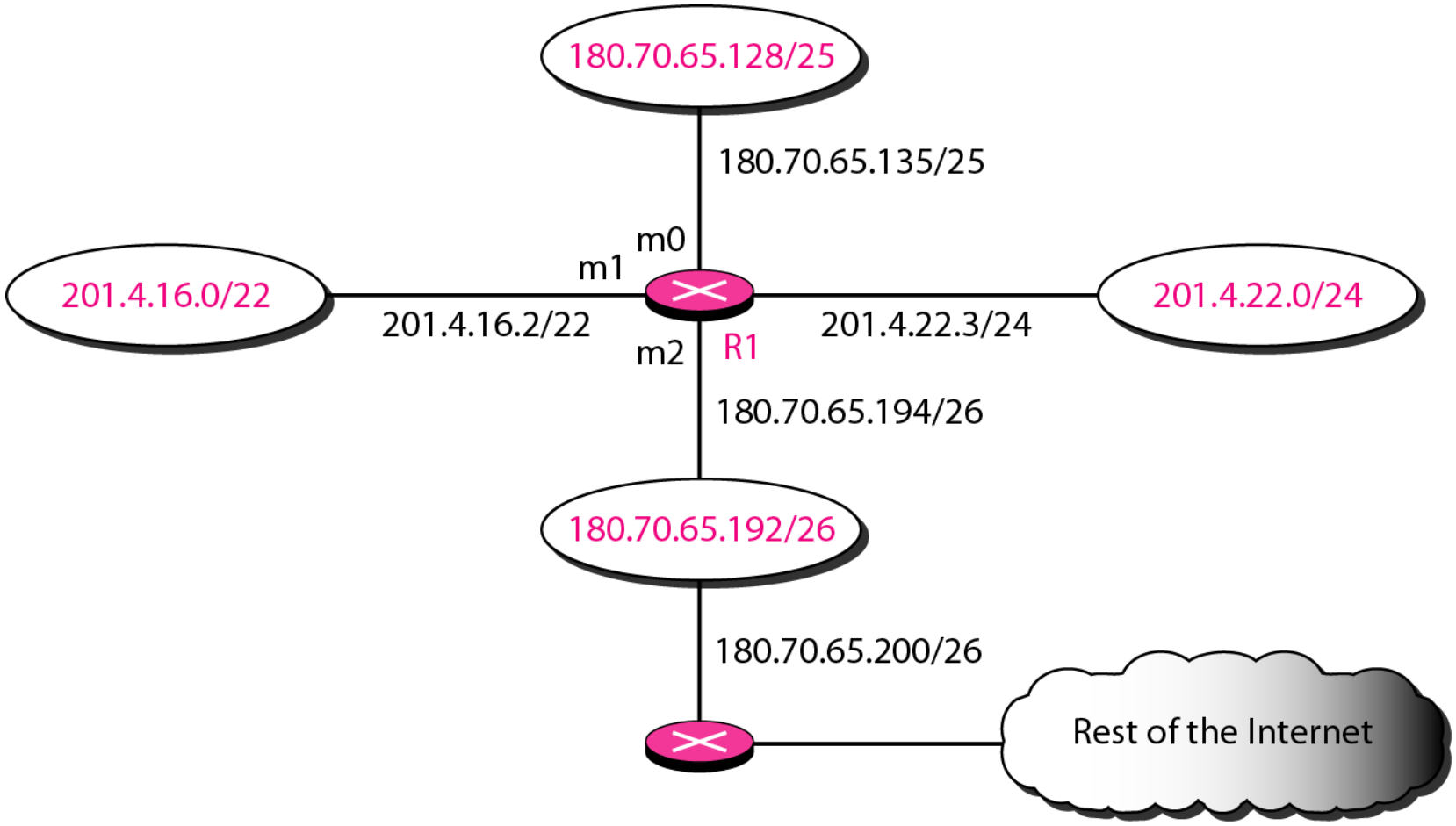
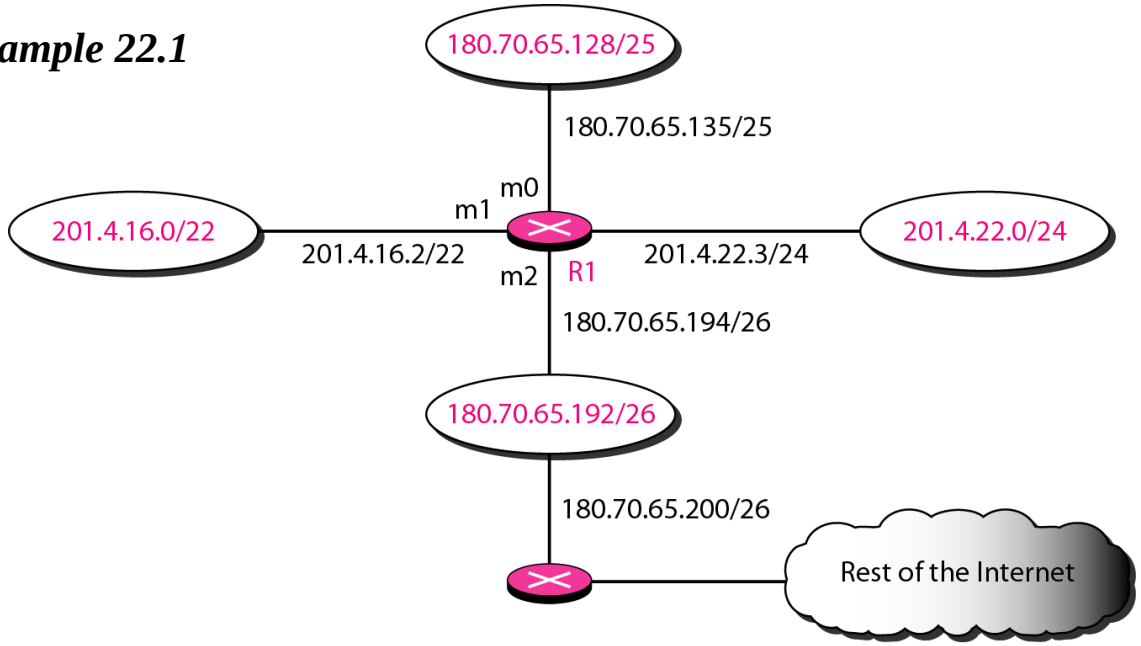


Figure 22.6 *Configuration for Example 22.1*



Solution

Table 22.1 shows the corresponding table.

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

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

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.

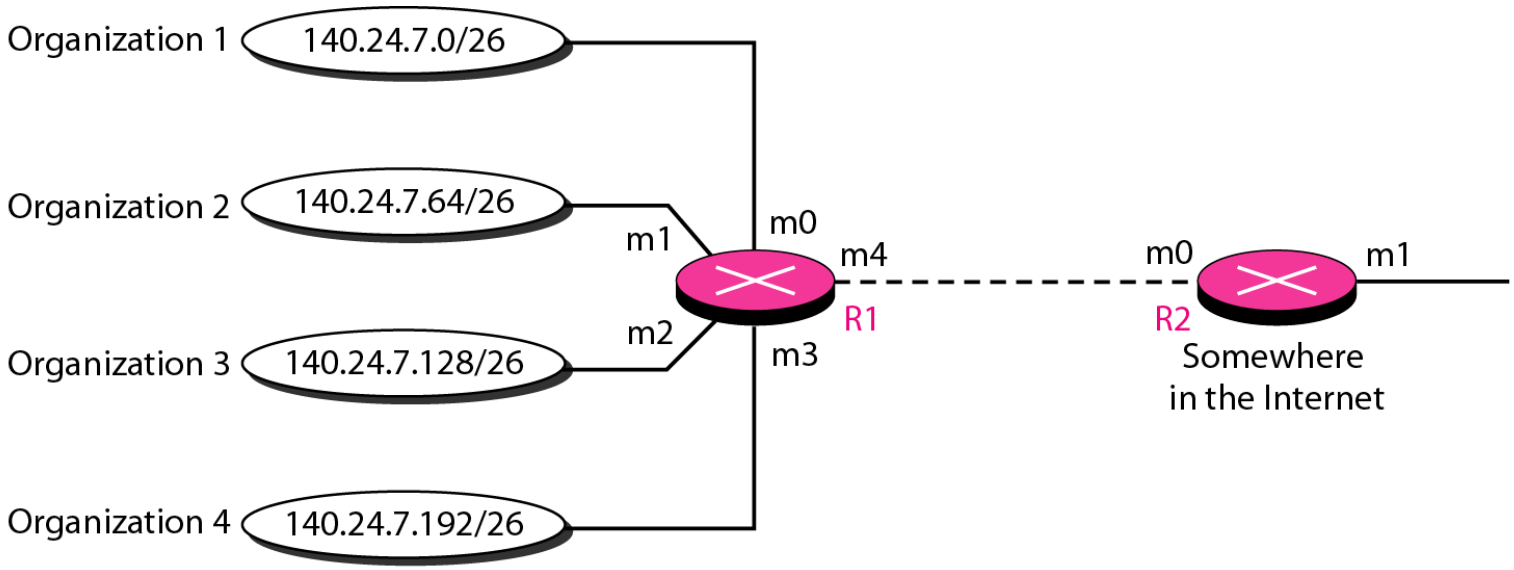
Solution

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.

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

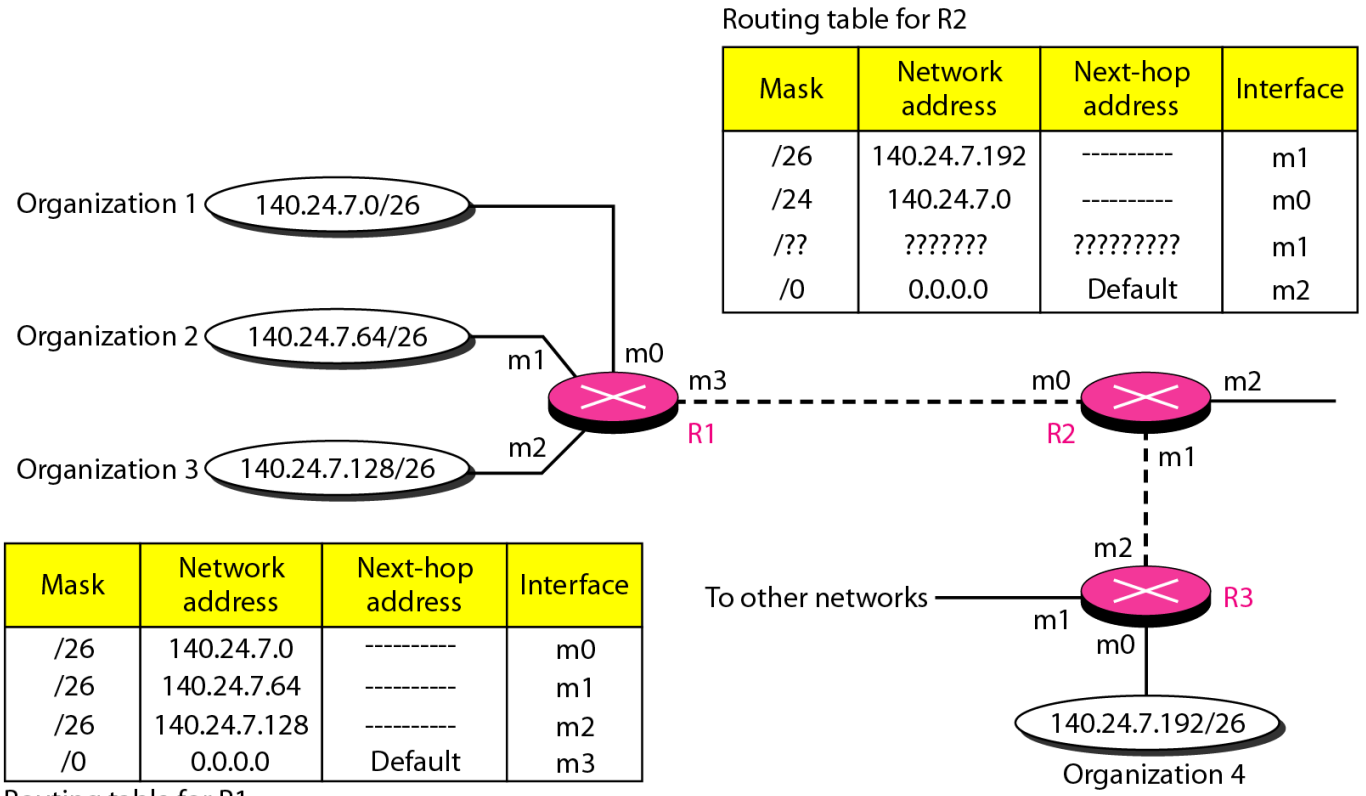
Routing table for R1

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

Routing table for R2

22.19

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*

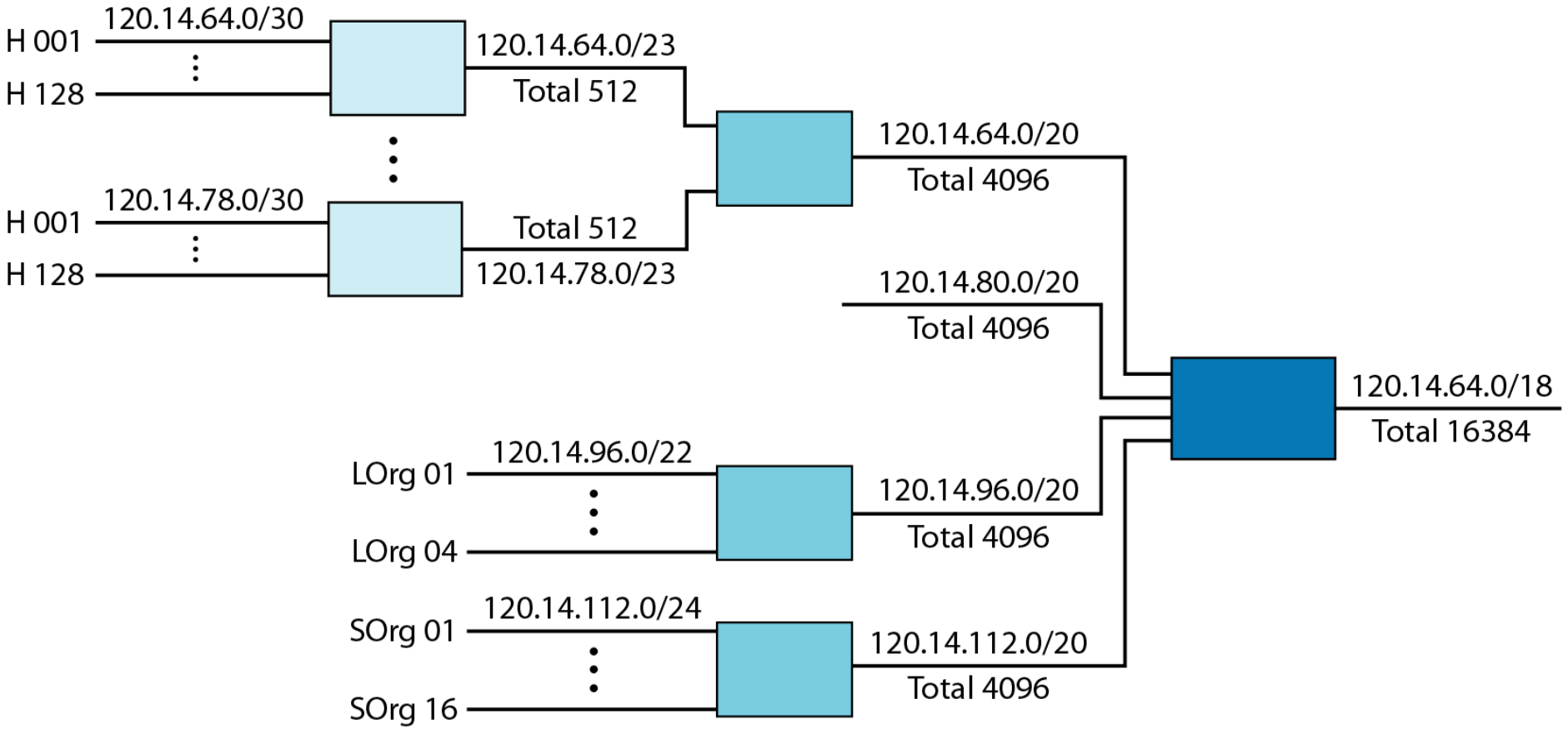


Figure 22.10 *Common fields in a routing table*

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

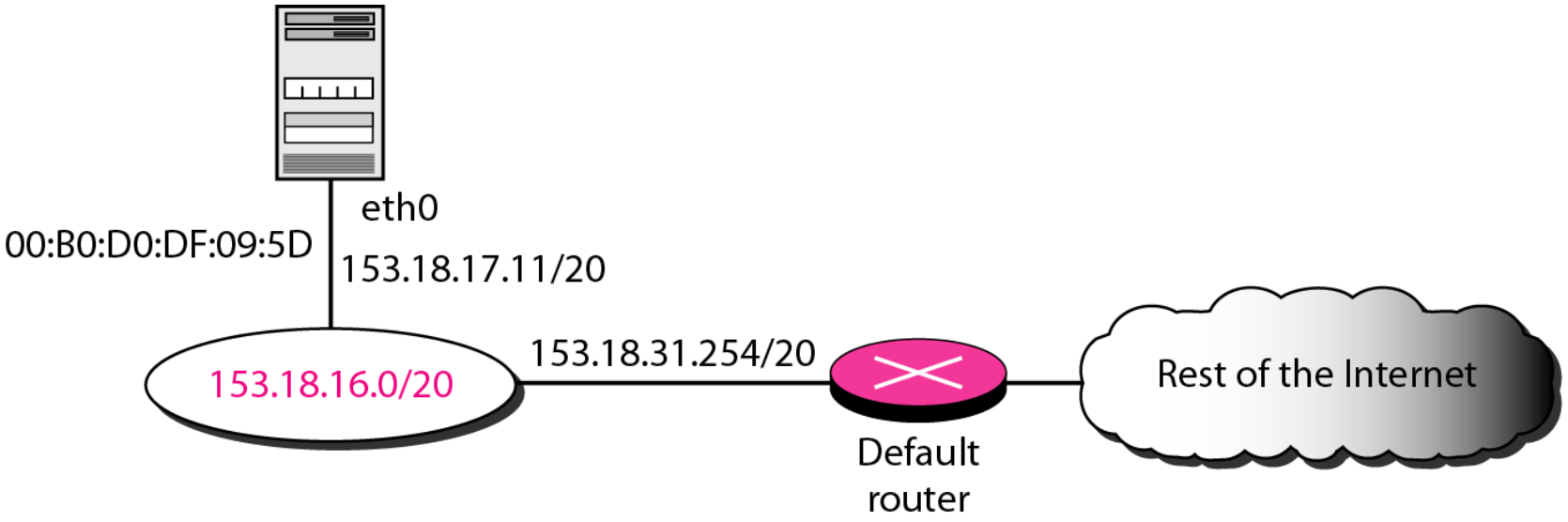
The destination column here defines the network address. The term gateway used by UNIX is synonymous with router. This column actually defines the address of the next hop. The value 0.0.0.0 shows that the delivery is direct. The last entry has a flag of G, which means that the destination can be reached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



22-3 UNICAST ROUTING PROTOCOLS

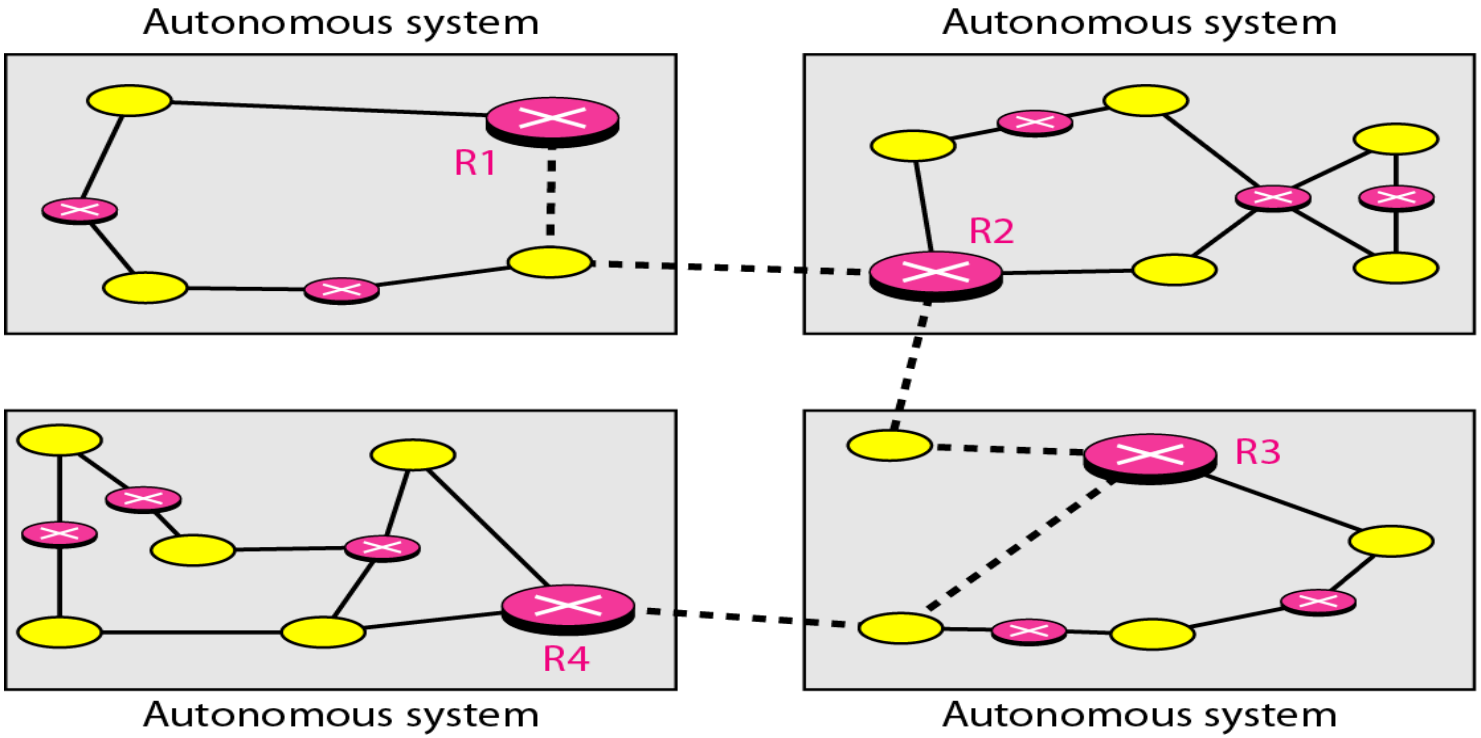
- ✓ *A routing table can be either **static or dynamic**.*
- ✓ *A **static** table is one with manual entries.*
- ✓ *A **dynamic** table is one that is updated automatically when there is a change somewhere in the Internet.*

- ✓ *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:

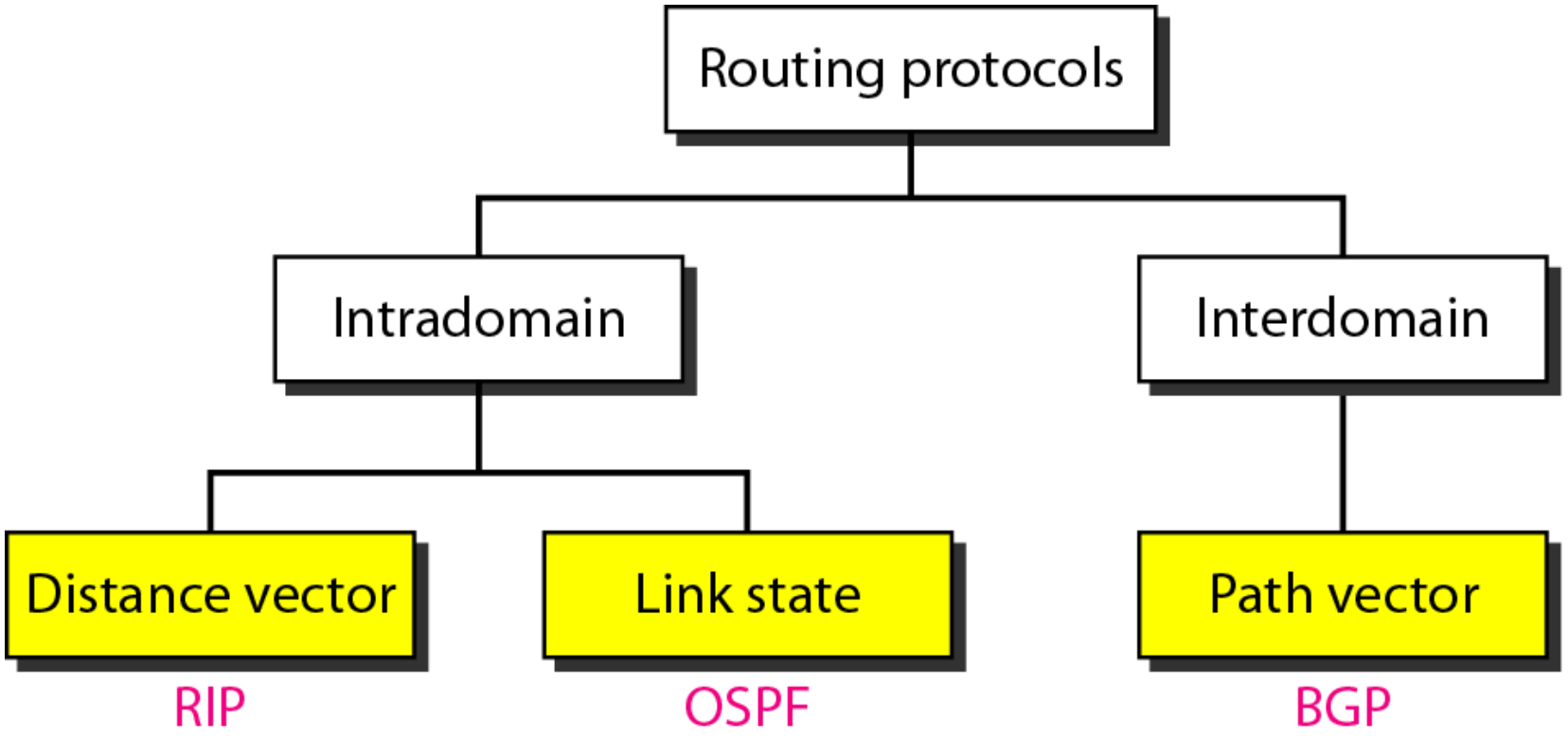
Optimization
Intra- and Interdomain Routing
Distance Vector Routing and RIP
Link State Routing and OSPF
Path Vector Routing and BGP

Figure 22.12 *Autonomous systems*



- ✓ Routing inside an autonomous system is referred to as **intradomain routing.**
- ✓ Routing between an autonomous system is referred to as **interdomain routing.**
- ✓ Each **autonomous system** can choose one or **more intradomain routing protocols** to handle routing
- ✓ **Only one interdomain routing protocol** handles routing between autonomous systems

Figure 22.13 *Popular routing protocols*



Distance Vector Routing

The distance vector routing algorithm is commonly known as the distributed **Bellman-Ford** routing algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962).

Figure 22.14 *Distance Vector Routing* tables *after convergence (or Stable)*

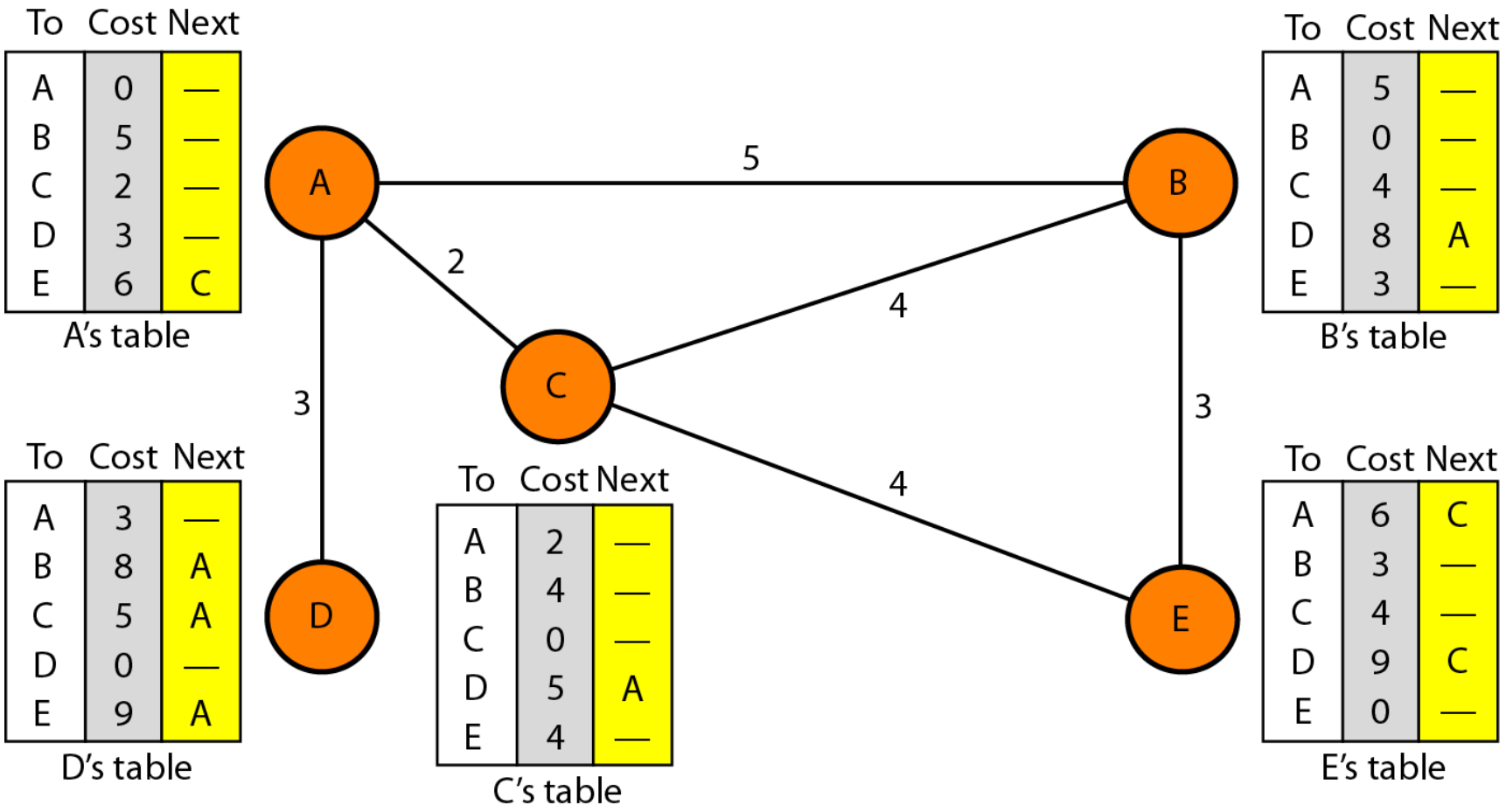
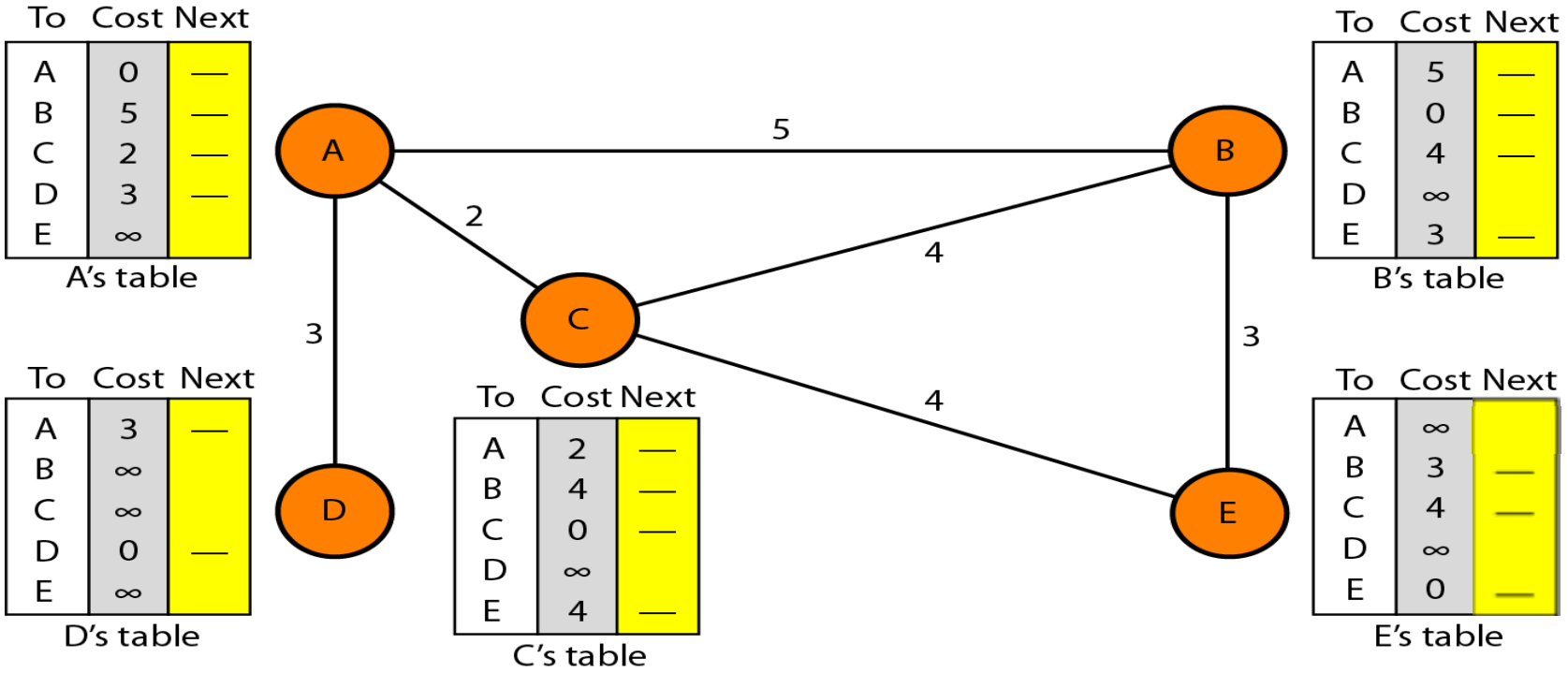


Figure 22.15 *Initialization of tables in Distance Vector Routing*



- ✓ Every router will share with neighbours only the first two columns.
- ✓ Then, apply Bellman-Ford equation to compute the updated cost.

Note

In Distance Vector Routing, each node shares its routing table with its immediate neighbors periodically (normally every 30 s) and when there is a change.

Figure 22.16 *Updating in Distance Vector Routing*

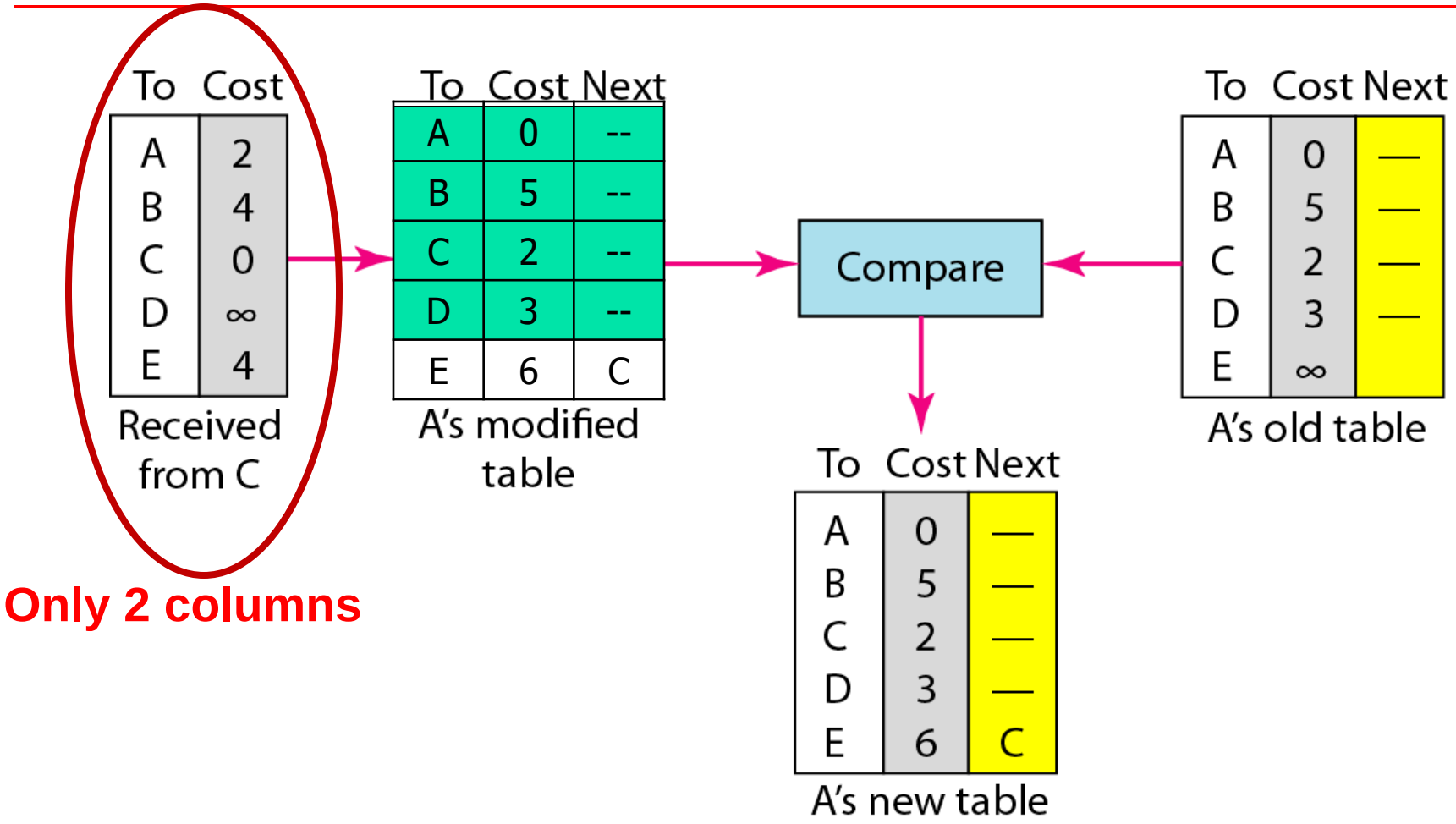
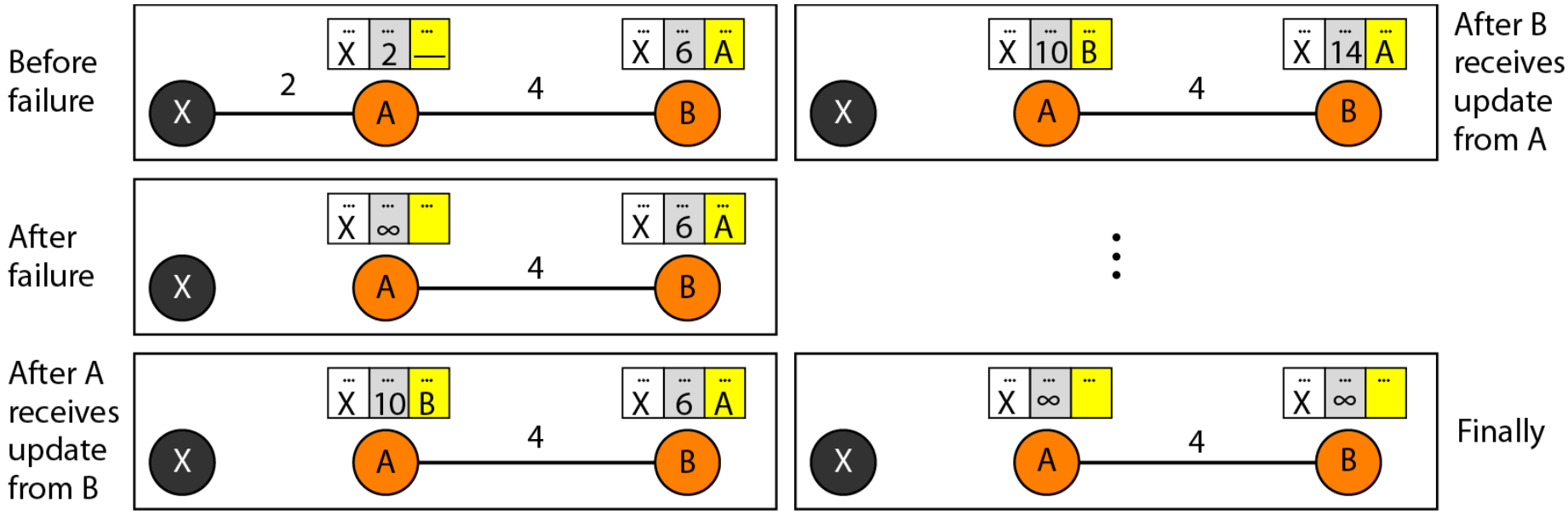


Figure 22.17 Two-node instability (Count-to-Infinity Problem)



- ✓ At the beginning, both nodes A and B know how to reach node X.
- ✓ But suddenly, **the link between A and X fails**.
- ✓ Node A changes its table. **If A can send its table to B immediately, everything is fine.**
- ✓ However, the system becomes unstable if B sends its routing table to A before receiving A's routing table.
- ✓ A sends its new update to B. Now B updates its routing table.
- ✓ The cost of reaching X increases gradually until it reaches infinity. At this moment, both A and B know that X cannot be reached. During this time the system is not stable.

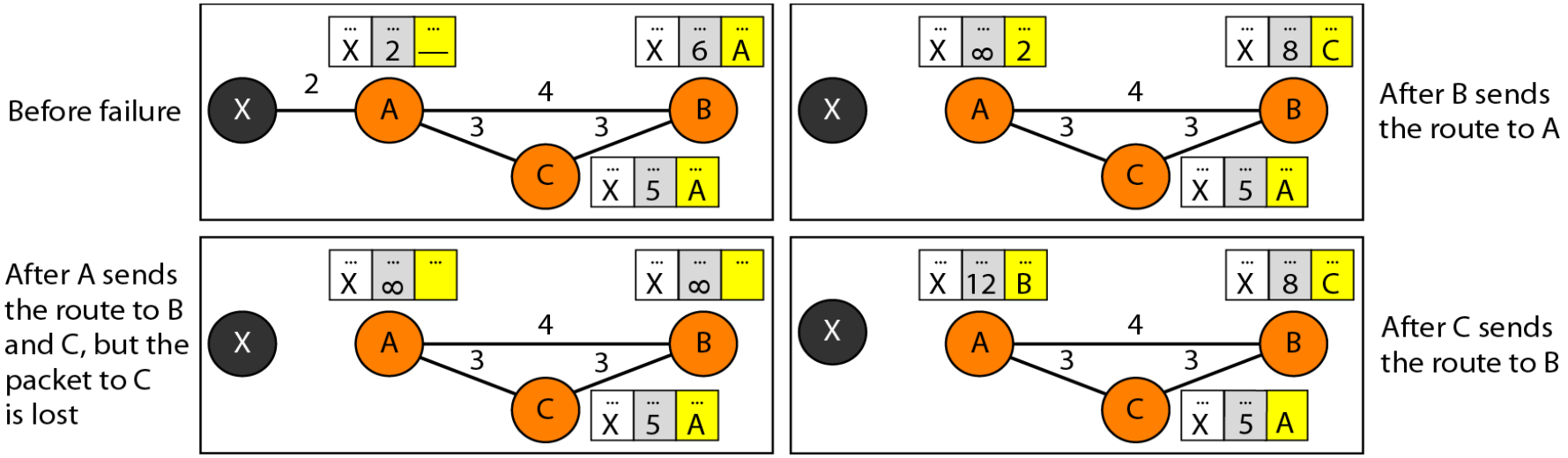
Solutions to Count-to-Infinity Problem

- ✓ **Defining Infinity**: to make the system stable quickly– reduce the infinity value. Most implementations of the distance vector protocol define the infinity value as 16 (15 hops is valid).

- ✓ Another solution **is Split Horizon**
  - ❖ **Strategy:** instead of flooding the table through each interface, each node sends only part of its table. If, according to its table, B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows).
  - ❖ In this case, A keeps the value of infinity as the distance to X. Later when A sends its routing table to B, B also corrects its routing table. The system becomes stable after the first update.

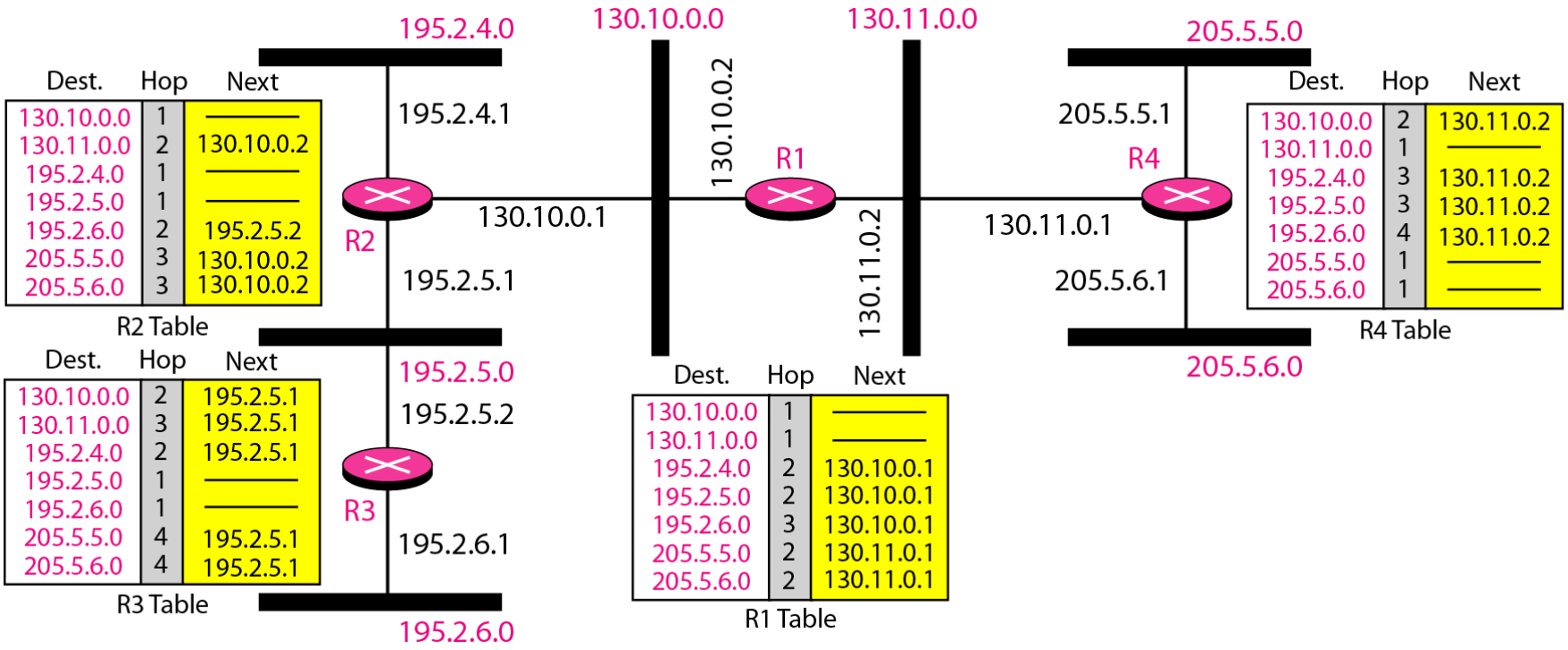
- ✓ **Split Horizon and Poison Reverse**

Figure 22.18 *Three-node instability*



- ✓ Let X becomes not reachable, A sends a packet to B and C to inform this.
- ✓ B updates its table, **but the packet to C is lost** and never reaches C.
- ✓ So, C still thinks that there is a route to X via A with a distance of 5.
- ✓ After a while, C sends to B its routing table, the route to X. **B is totally fooled here**.
- ✓ Now, B advertise this route to A, now A is fooled and update its table.
- ✓ Now A advertise this route to C with increased cost, but not to B.
- ✓ This loop continues and stops when it reaches infinity.

Figure 22.19 *Example of a domain using RIP*



- ✓ Destination in a routing table is a **network address**
- ✓ **Metric** used is **hop count**
- ✓ **Infinity is defined as 16**, so cannot have more than 15 hops
- ✓ **Next-node column** defines the address of the router's interface, the packet to be sent to reach its destination.

Link State Routing

- ✓ In link state routing, each node has the complete network topology, including all nodes, links, and link details (type, cost, status).
- ✓ Using this information, a node can apply Dijkstra's algorithm to build its routing table.
- ✓ Link-state routing assumes that while global knowledge of the topology isn't fully clear, each node has partial knowledge, specifically the status and cost of its own links.
- ✓ This partial information from each node **combines to** form the **full network** view.

Figure 22.20 *Concept of link state routing*

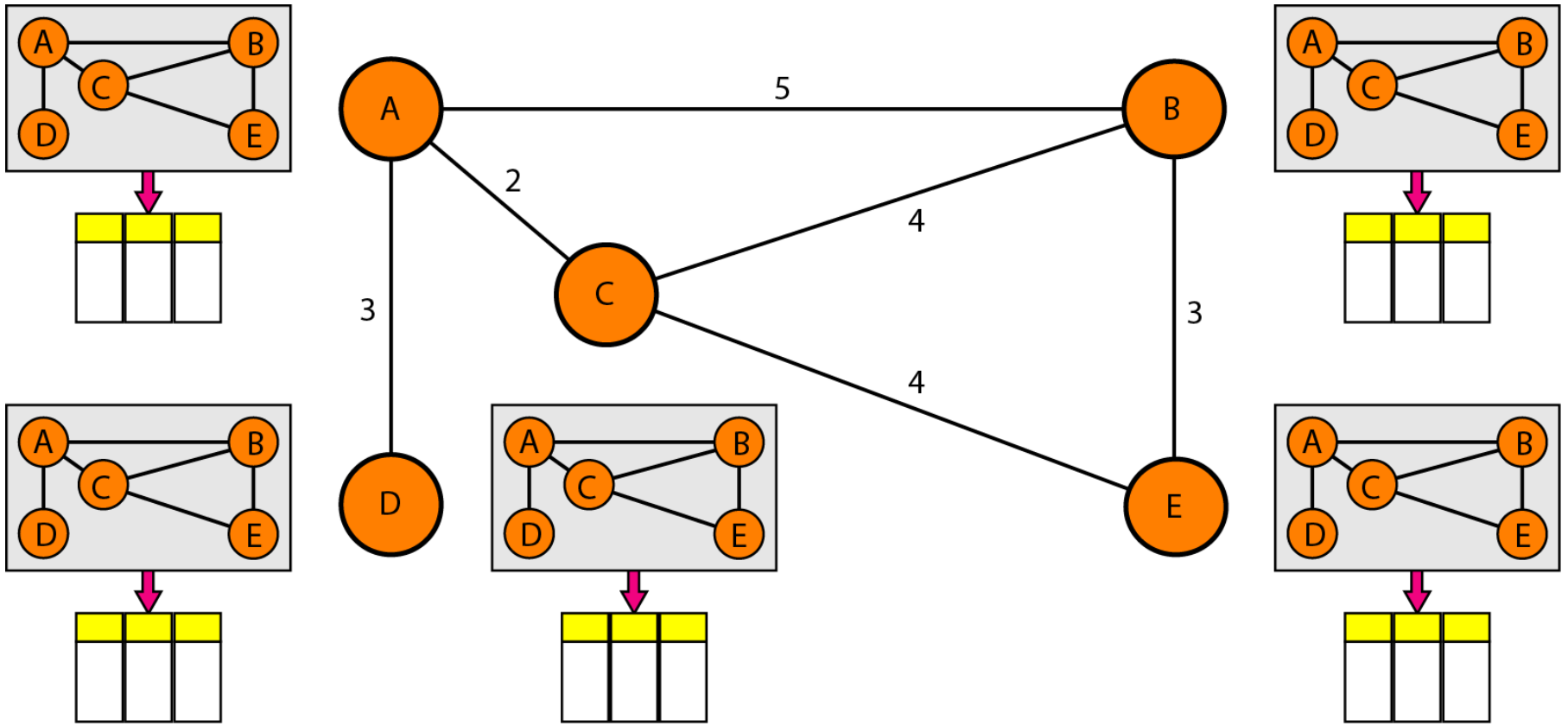
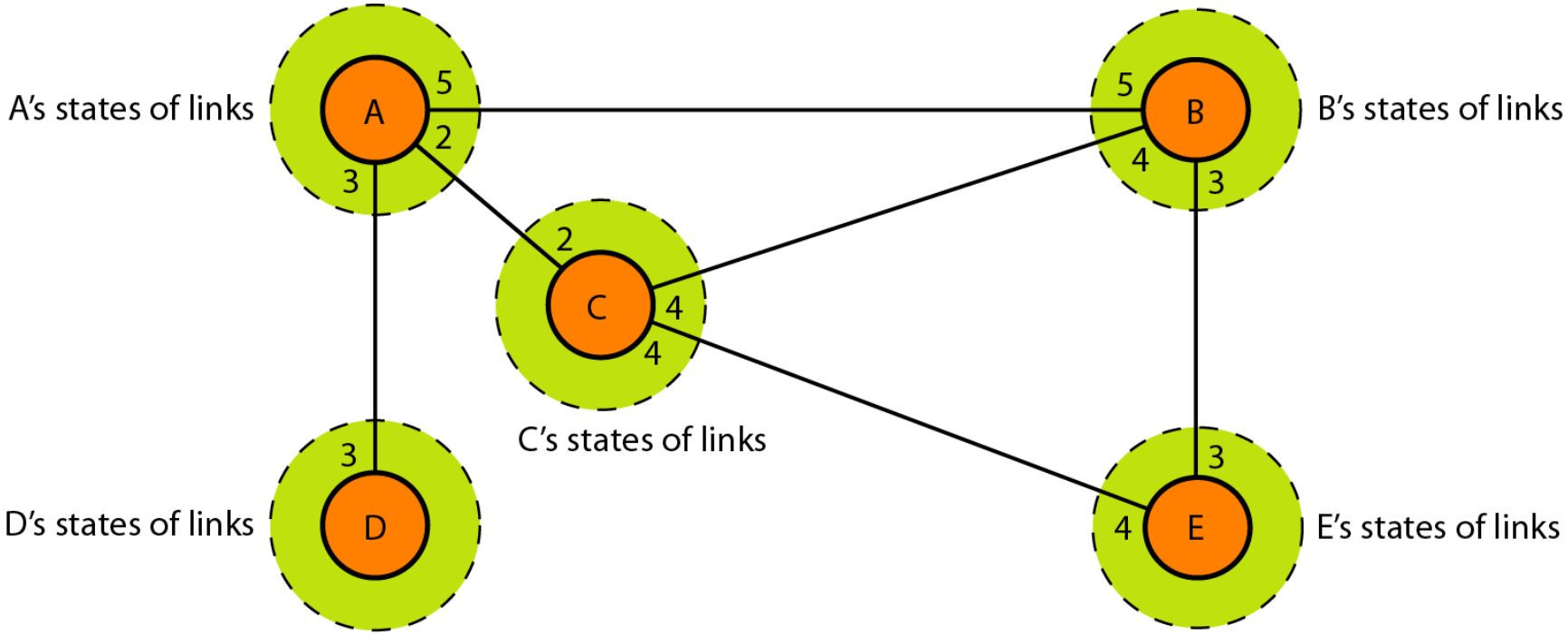


Figure 22.21 *Link state knowledge*



Creation of a common topology using the shared information

Building Routing Tables

Four sets of actions are required:

1. Creation of the **states of the links** by each node, called the **link state packet (LSP**).
2. Dissemination of LSPs to every other router, using **flooding.**
3. Formation of a **shortest path tree for each node.**
4. **Calculation of a routing table** based on the shortest path tree.

Creation of Link State Packet (LSP)

LSP include:

- ✓ Node identity,
- ✓ List of links,
- ✓ Sequence number: facilitates flooding and distinguishes new LSP from old ones.
- ✓ Age (TTL): prevents old LSPs from remaining in the domain for a long time.

Node Identity: A	
Sequence Number	
TTL	
A	0
B	5
C	2
D	3

LSP of router A

LSPs are generated on two occasions:

1. When there is a change in topology
2. On a periodic basis (1 to 2 hrs)

Figure 22.22 *Dijkstra algorithm*

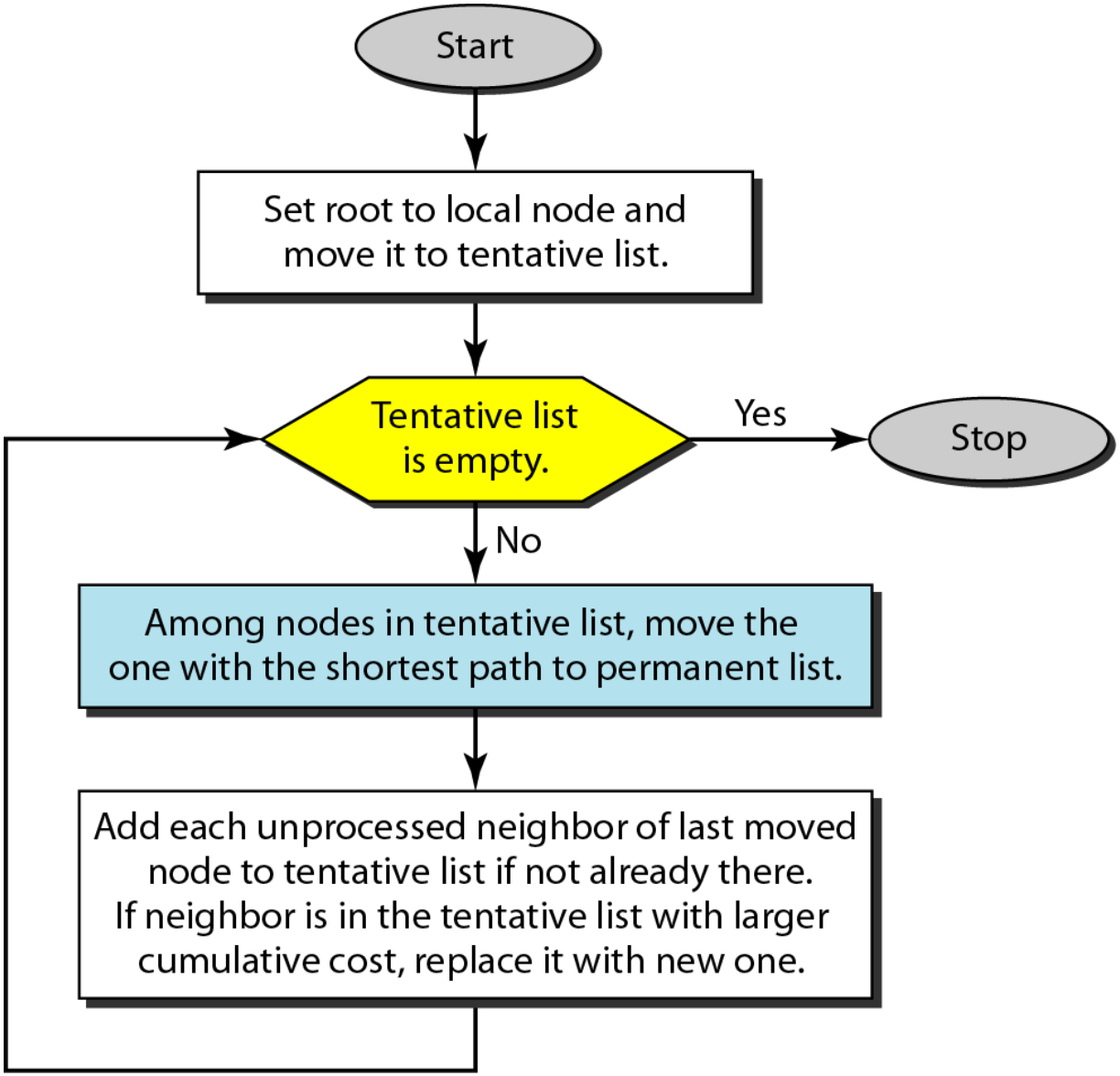
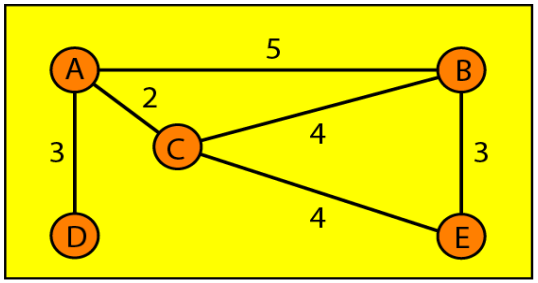
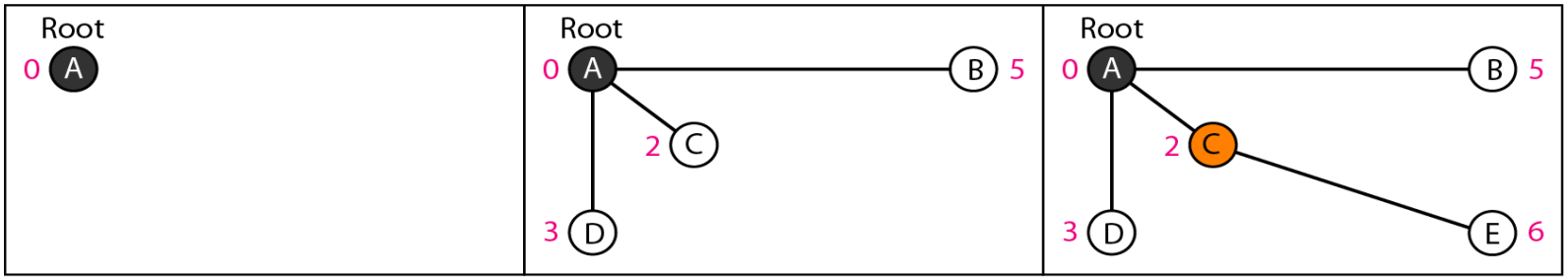


Figure 22.23 *Example of formation of shortest path tree*



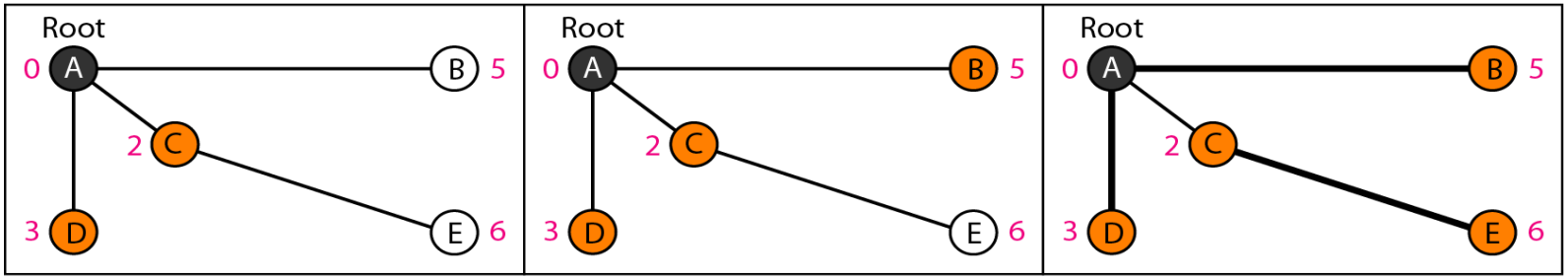
Topology



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.

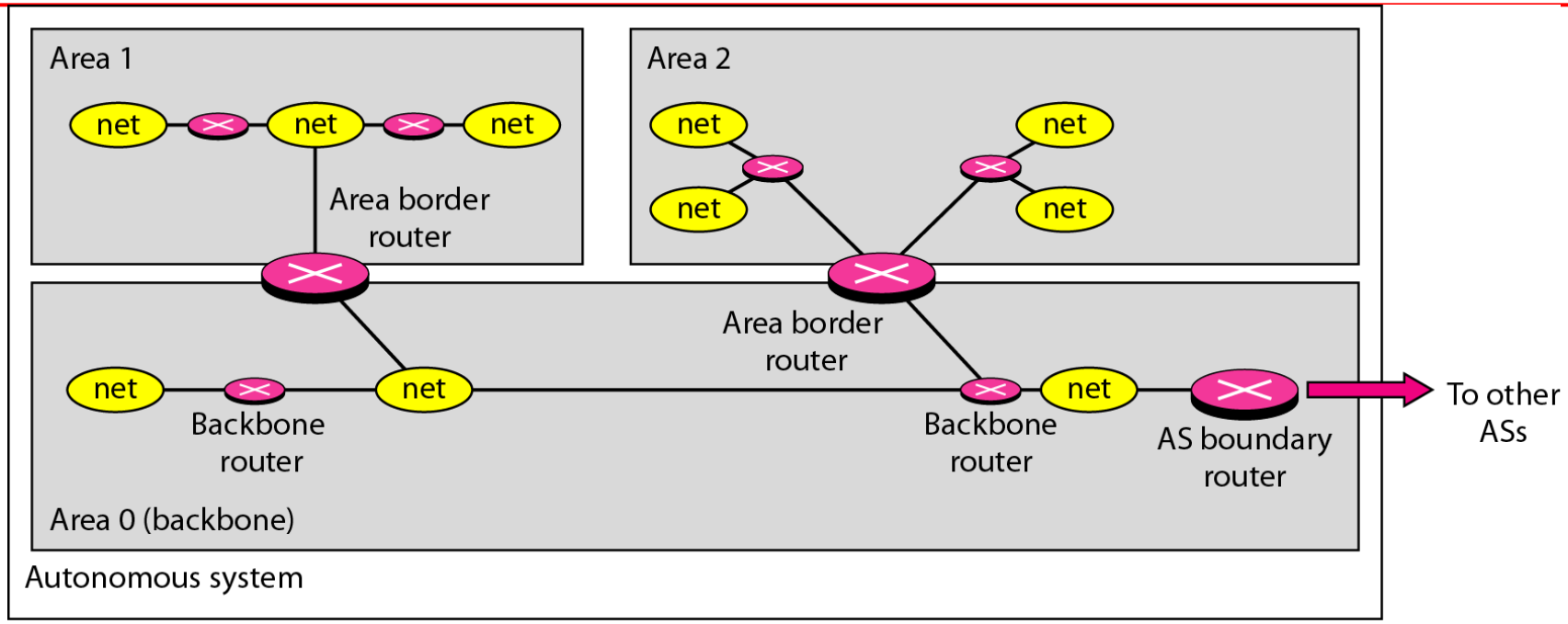
5. Move B to permanent list.

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

Table 22.2 *Routing table for node A*

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

Figure 22.24 *Areas in an autonomous system*



- ✓ OSPF divides an autonomous system into areas.
- ✓ At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas.
- ✓ Areas inside an autonomous system is a special area called the ***backbone***
- ✓ Routers inside the backbone are called the backbone routers. Backbone router can also be an area border router.
- ✓ Area identification of the backbone is zero.
- ✓ **Metric** can be based on a type of service (delay, throughput, and so on).

Figure 22.25 *Types of links*

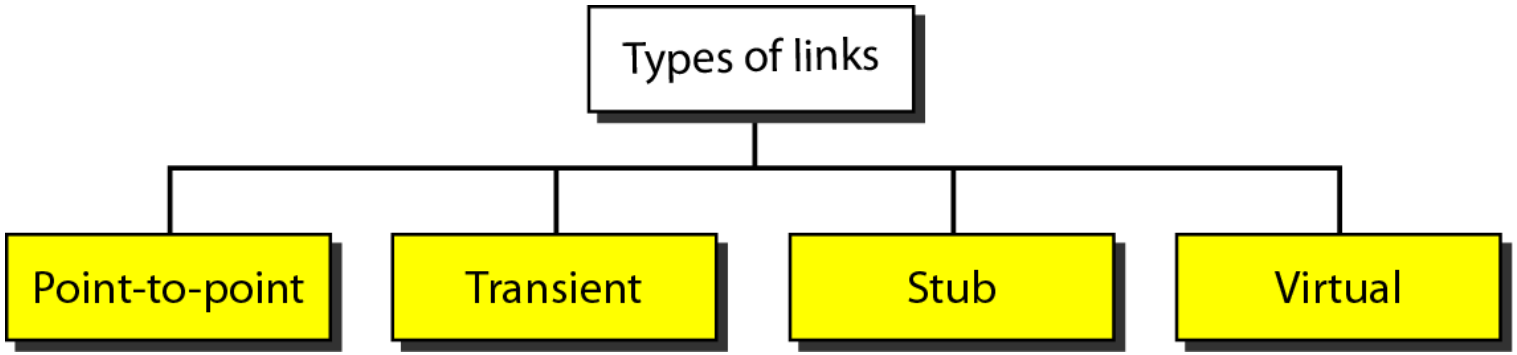
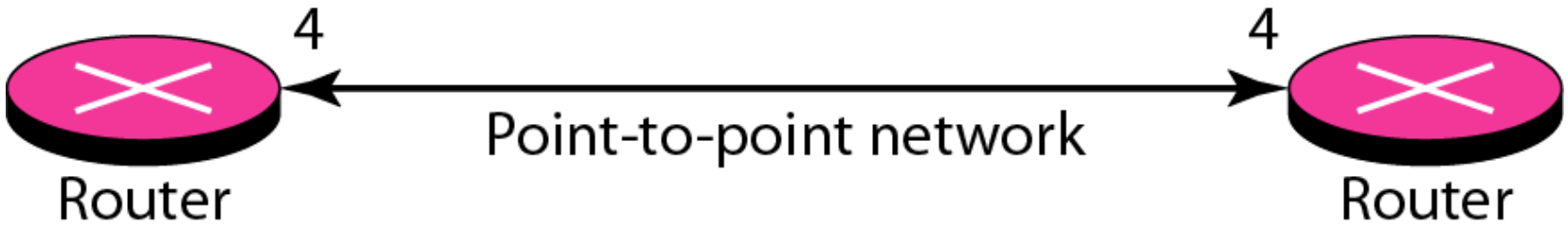
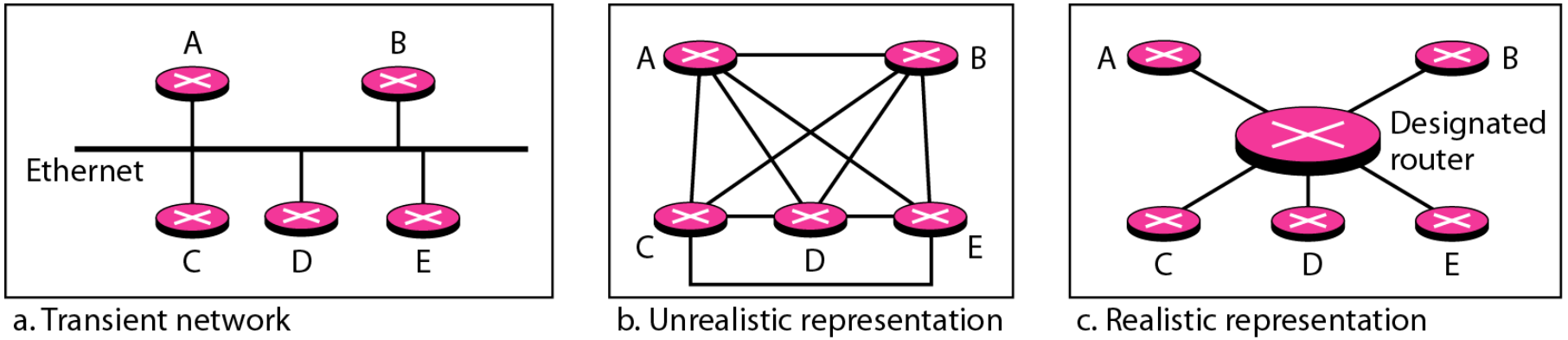


Figure 22.26 *Point-to-point link*



A point-to-point link connects two routers without any other host or router in between.

Figure 22.27 *Transient link*



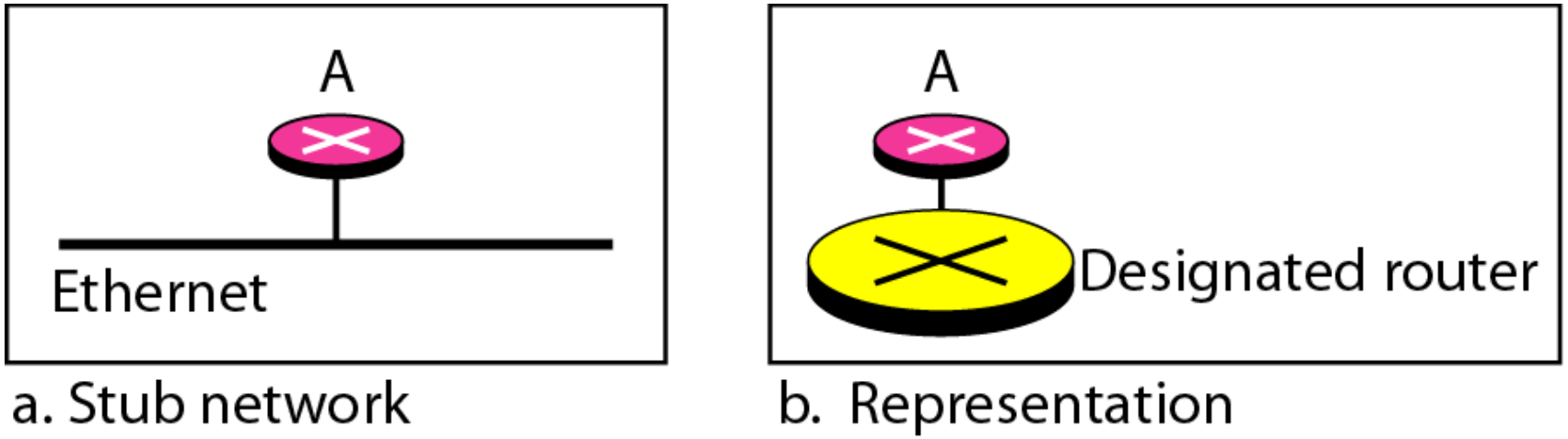
a. Transient network

b. Unrealistic representation

c. Realistic representation

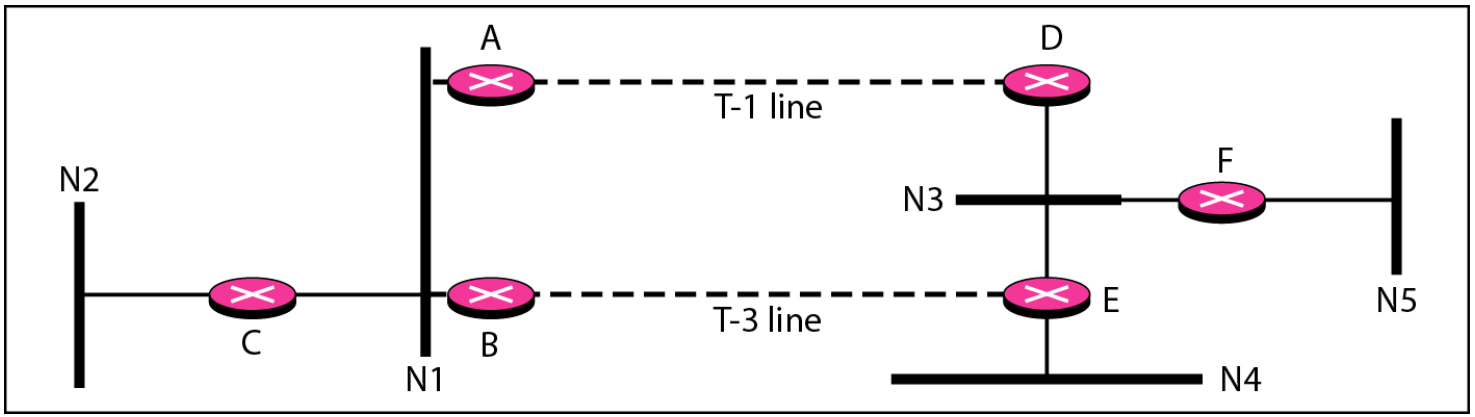
A transient link **is a network** with several routers attached to it

Figure 22.28 *Stub link*

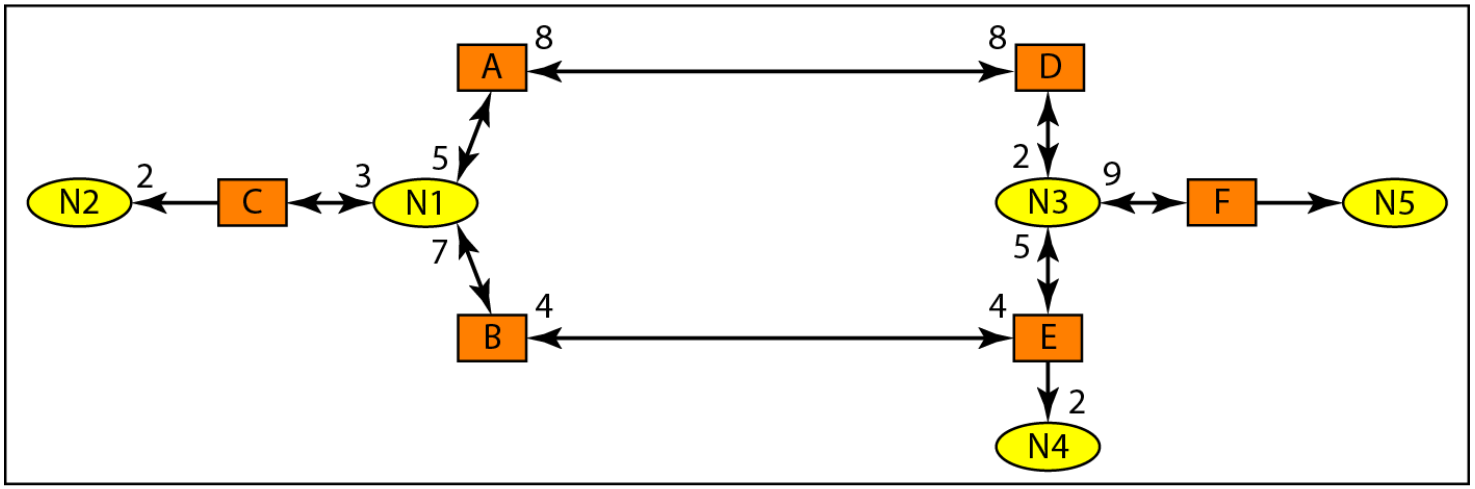


- ✓ A **stub link** is a network that is connected to only one router.
- ✓ The data packets enter the network through this single router and leave the network through this same router.
- ✓ A special case of the transient network.
- ✓ Can show this situation using the router as a node and using the designated router for the network

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



a. Autonomous system



b. Graphical representation

Path Vector Routing

- ✓ It is an Interdomain routing protocol based on path vector routing.
- ✓ There is one **speaker node** in each autonomous system that acts on behalf of the entire autonomous system.
- ✓ 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,

Figure 22.30 Initial routing tables in Path Vector Routing

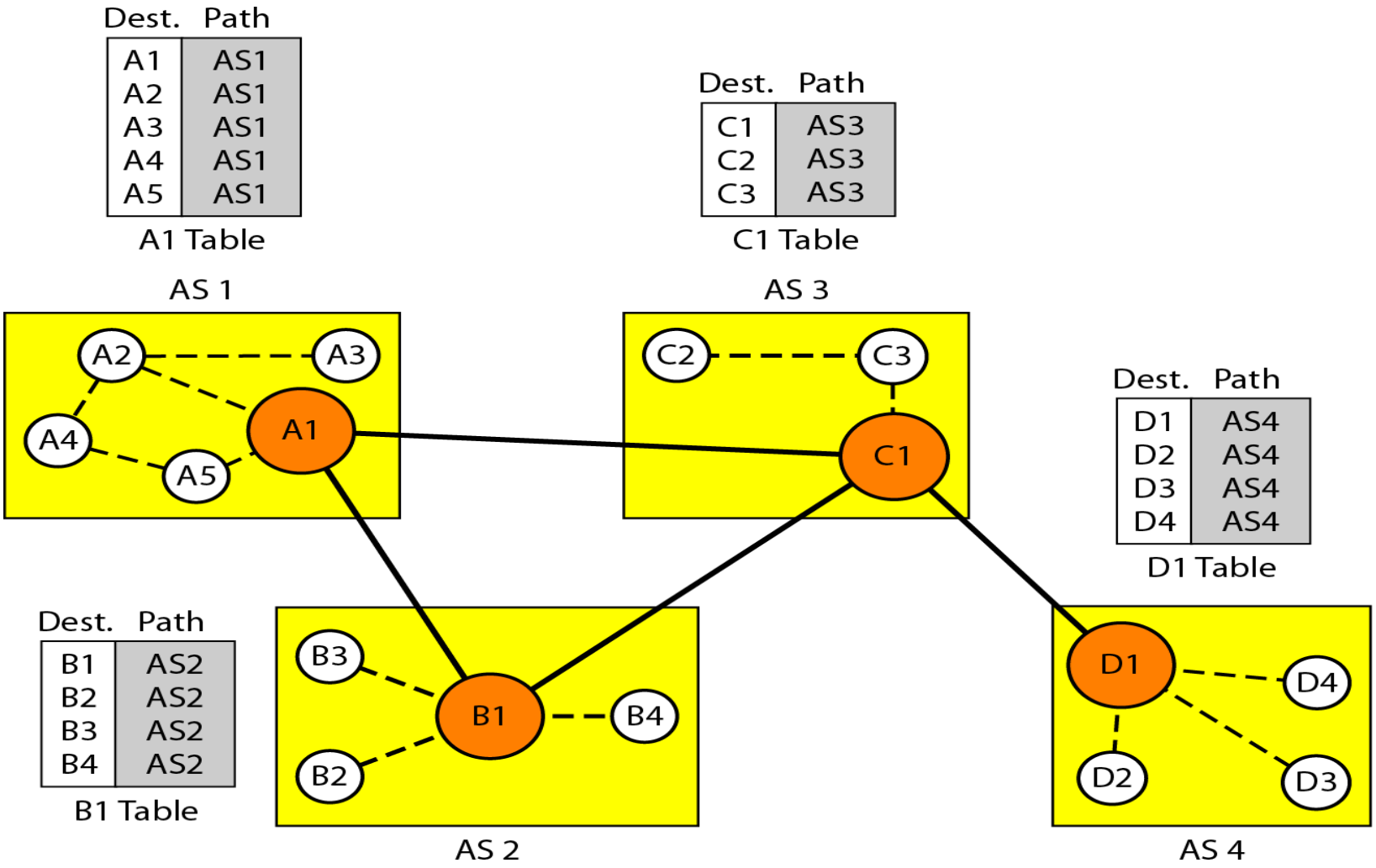


Figure 22.31 *Stabilized tables for three autonomous systems*

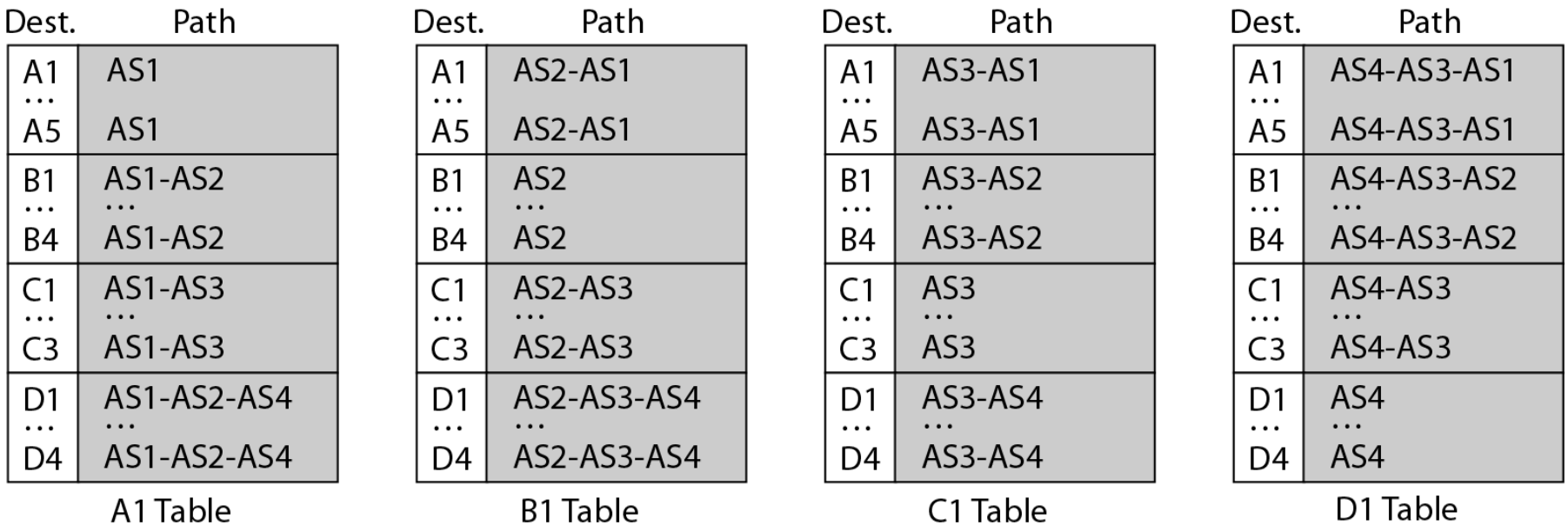
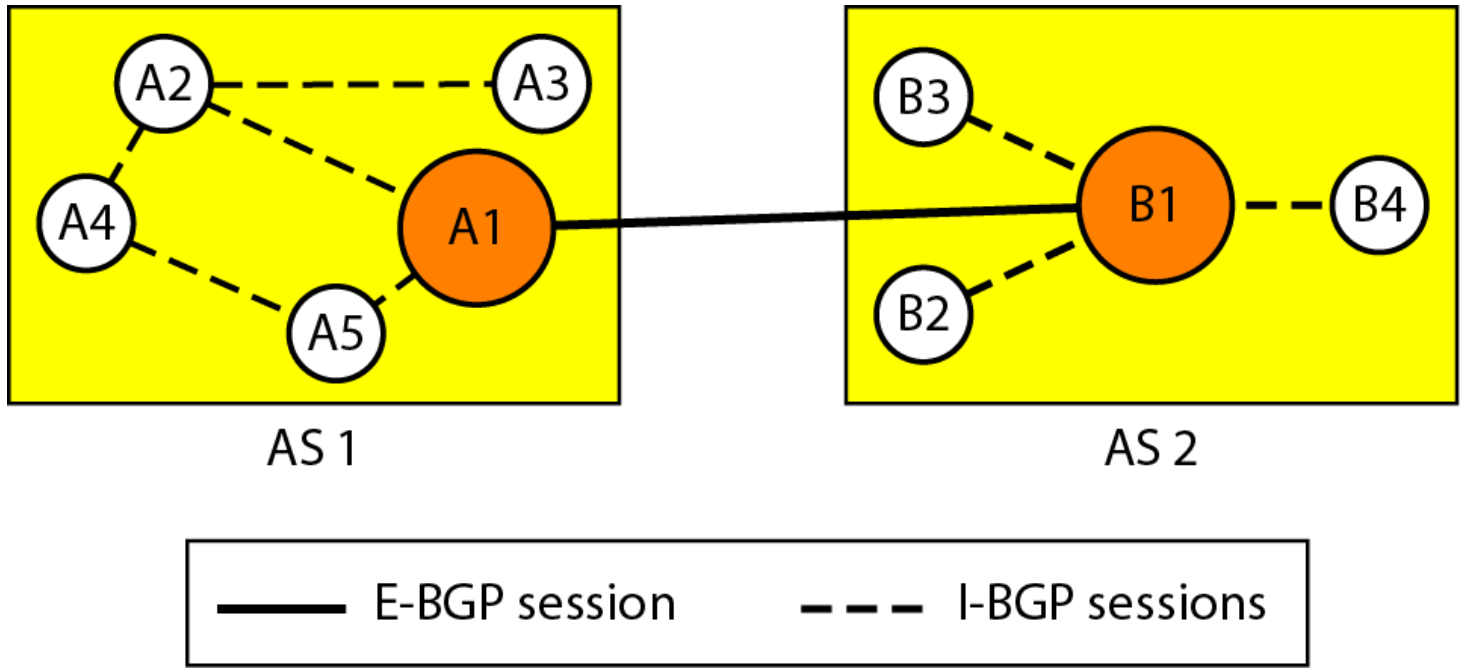


Figure 22.32 *Internal and external BGP sessions*



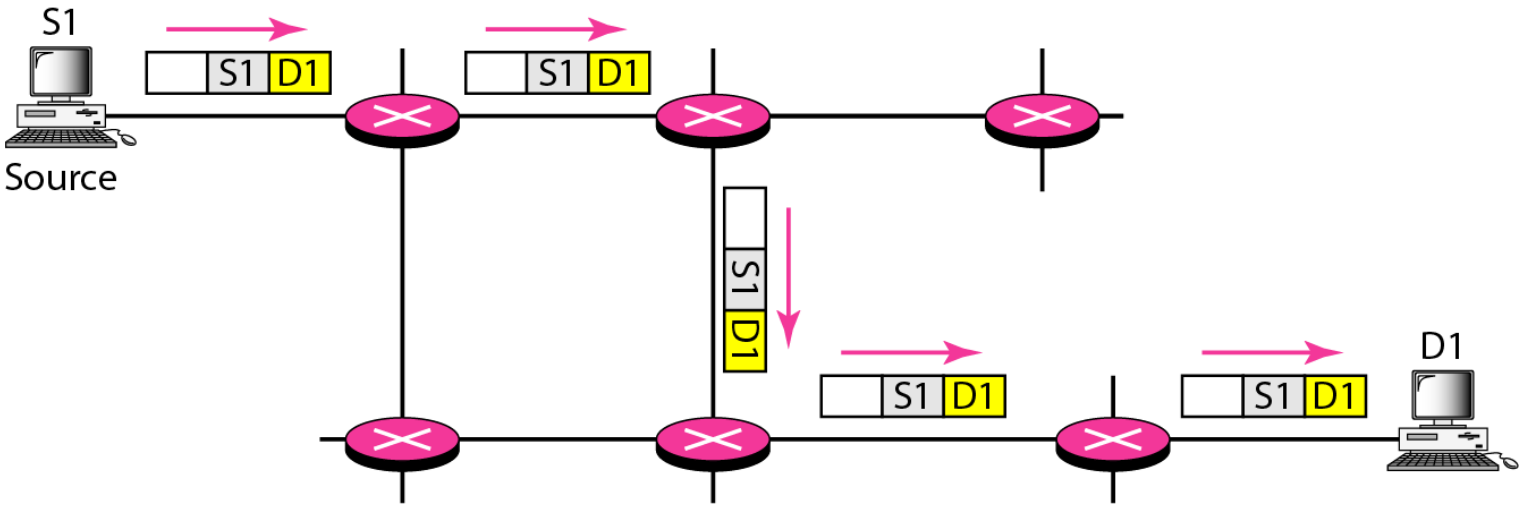
22-4 MULTICAST ROUTING PROTOCOLS

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

Topics discussed in this section:

Unicast, Multicast, and Broadcast
Applications
Multicast Routing
Routing Protocols

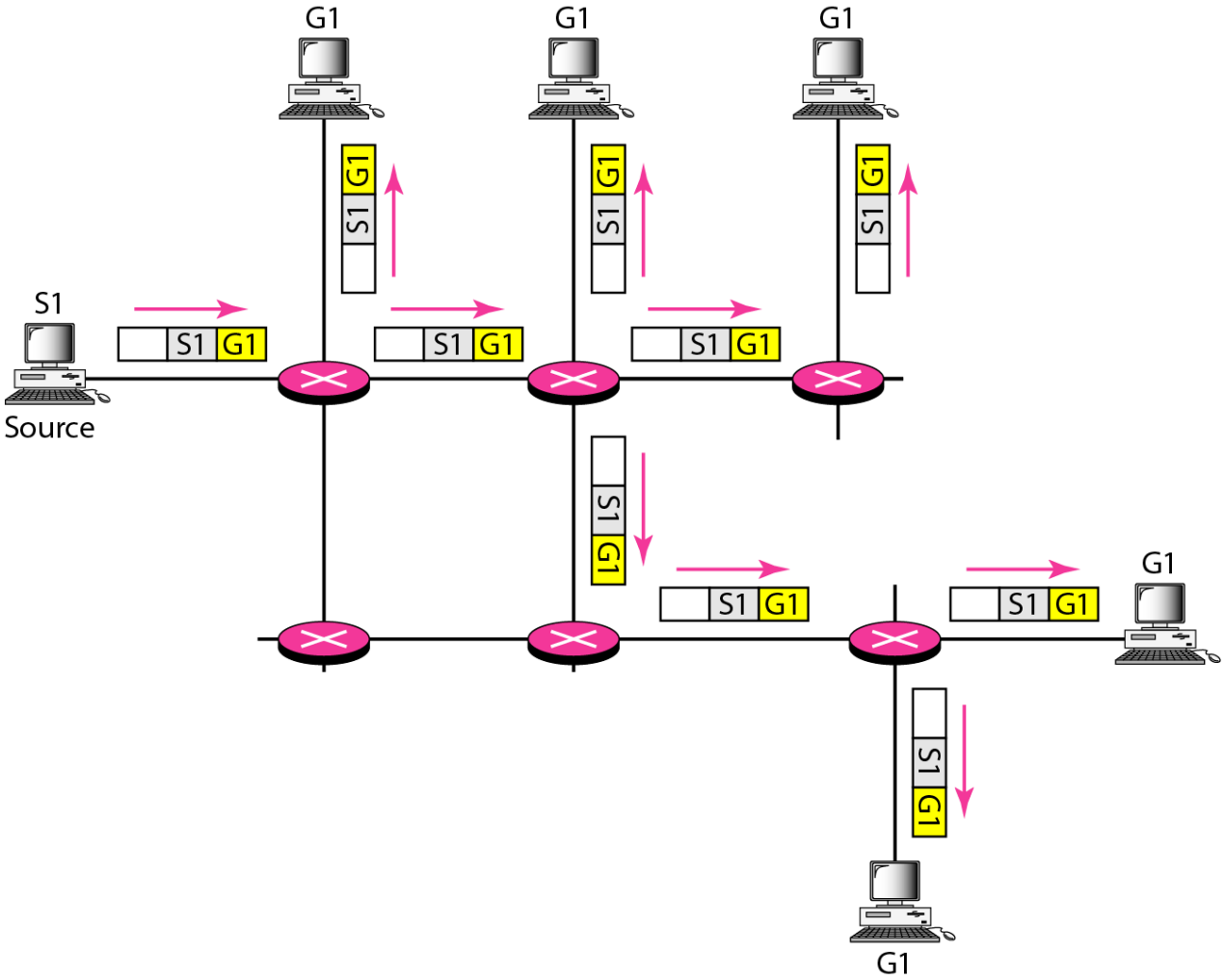
Figure 22.33 *Unicasting*



Note

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

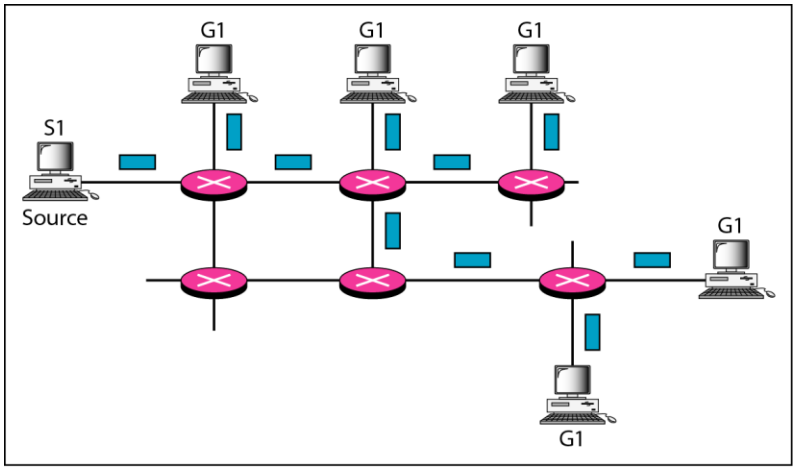
Figure 22.34 *Multicasting*



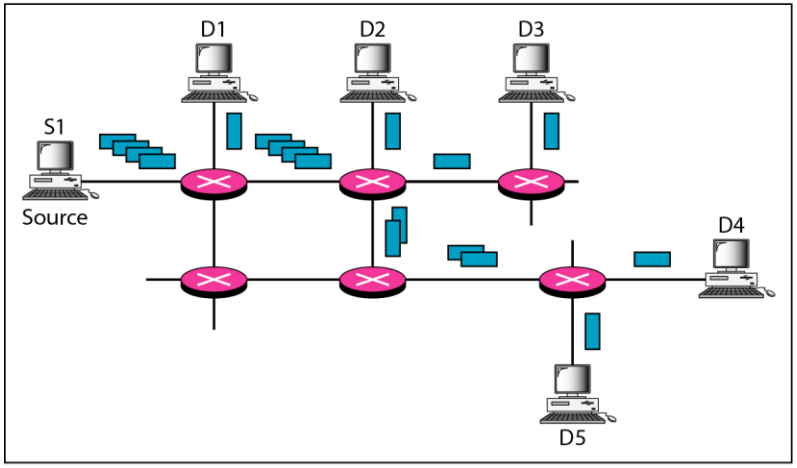
Note

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

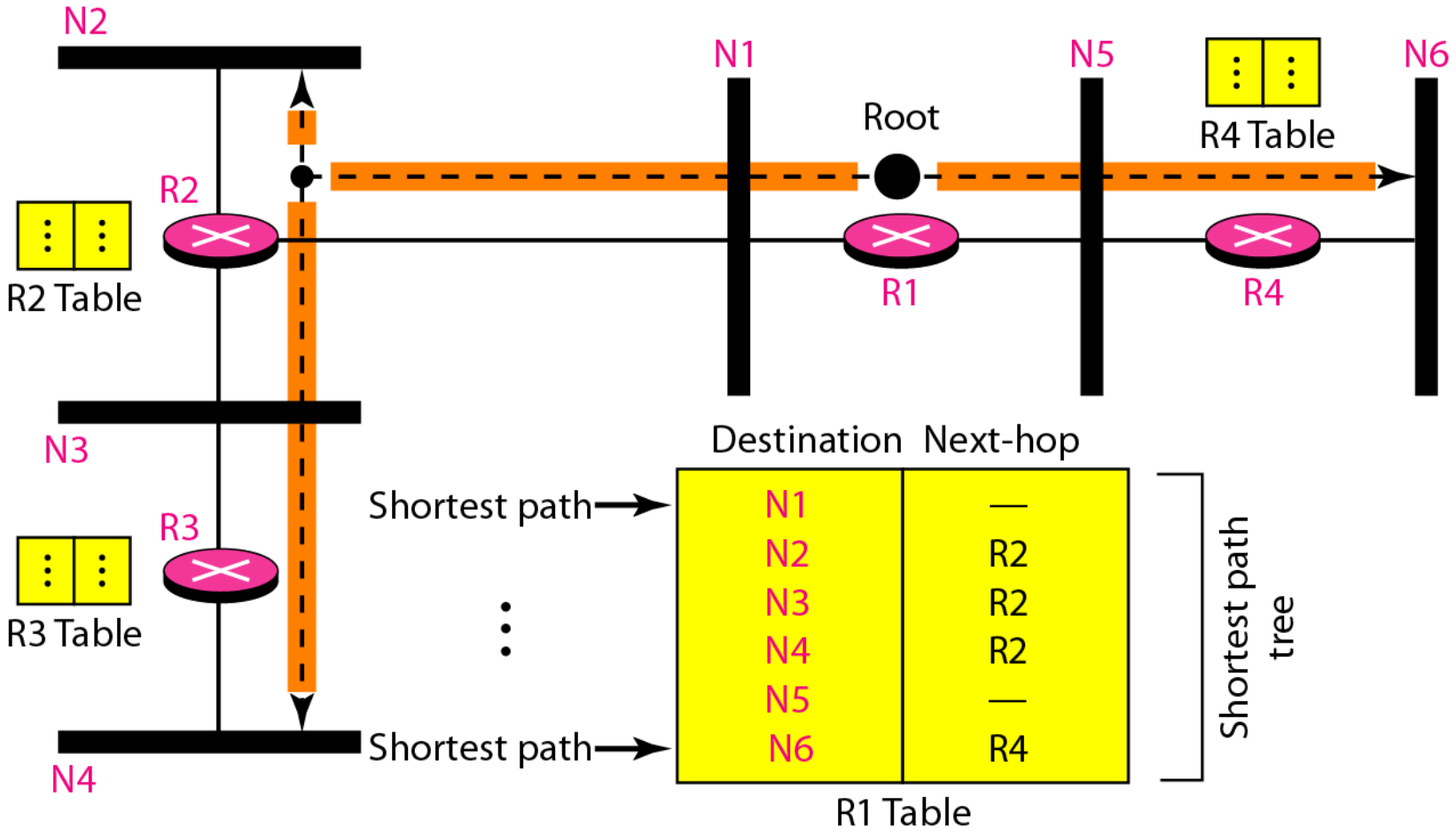
Note

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.

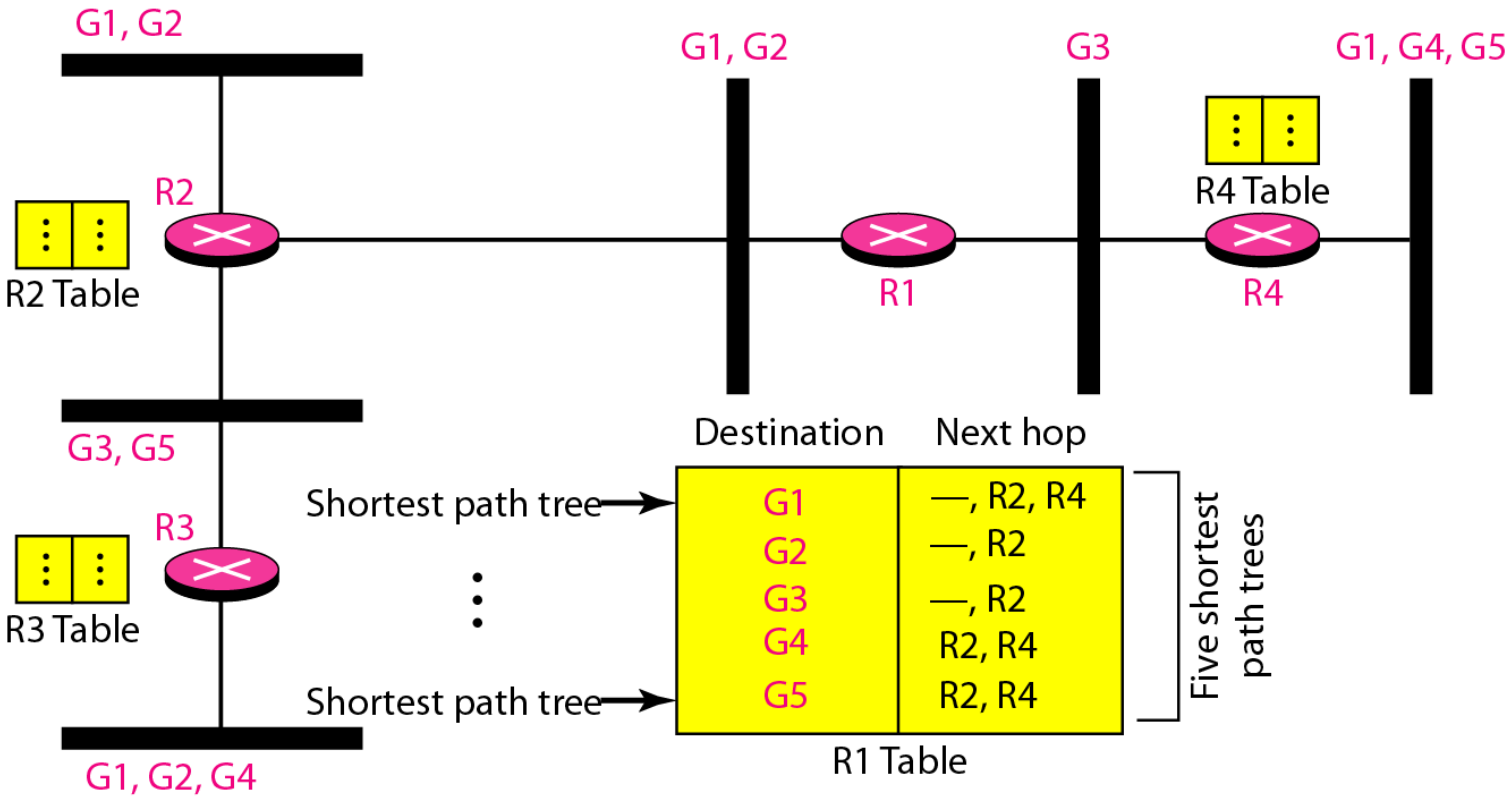
Figure 22.36 *Shortest path tree in unicast routing*



Note

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

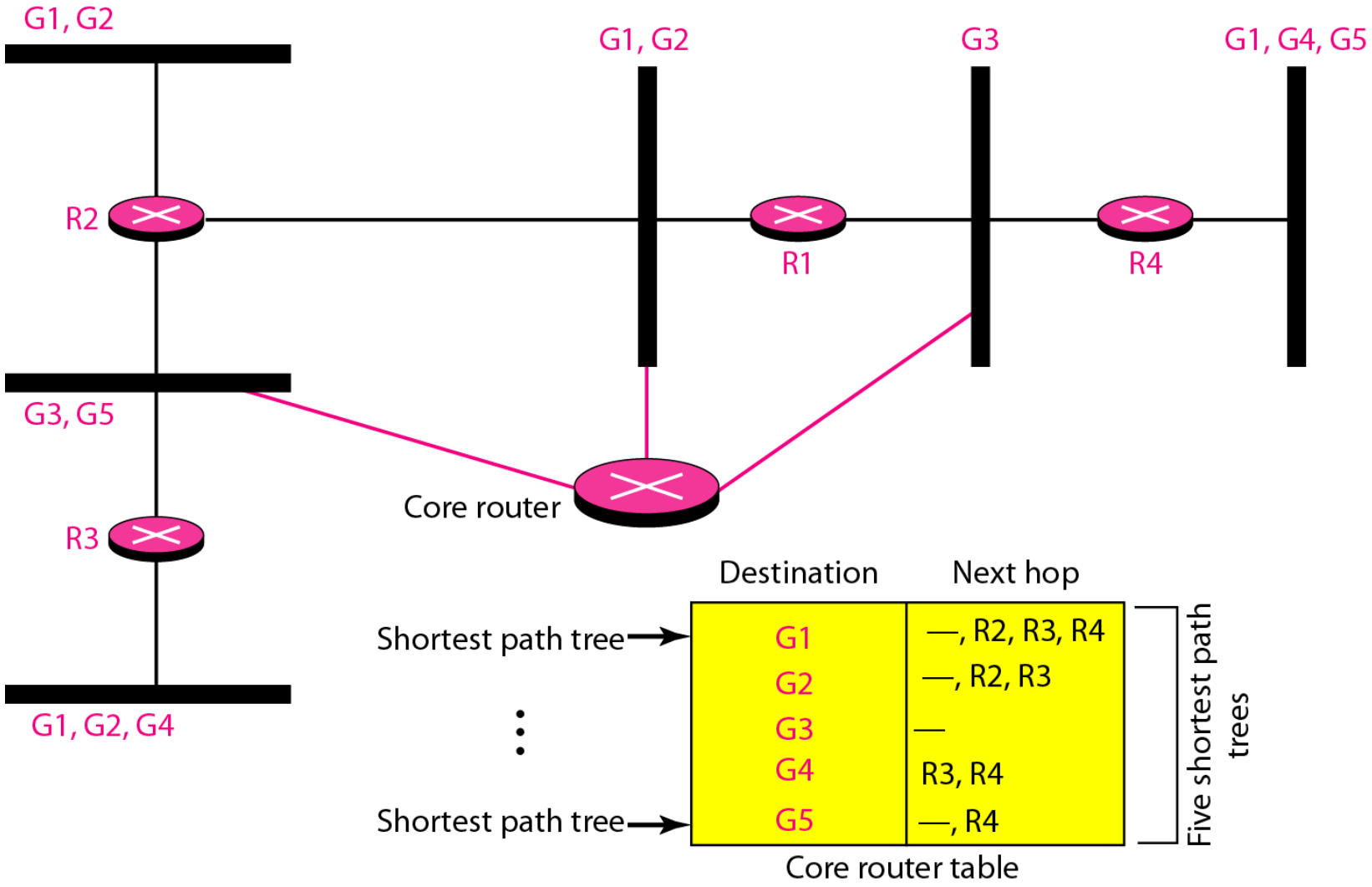
Figure 22.37 *Source-based tree approach*



Note

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

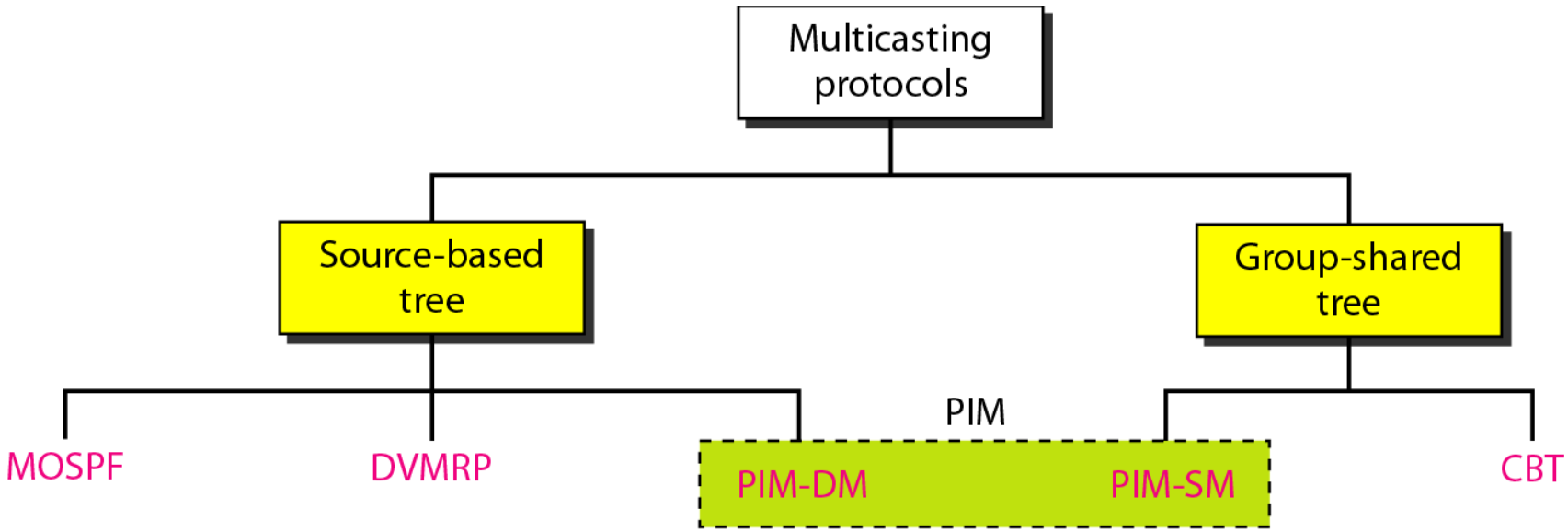
Figure 22.38 *Group-shared tree approach*



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

Multicast link state 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.

Figure 22.40 *Reverse path forwarding (RPF)*

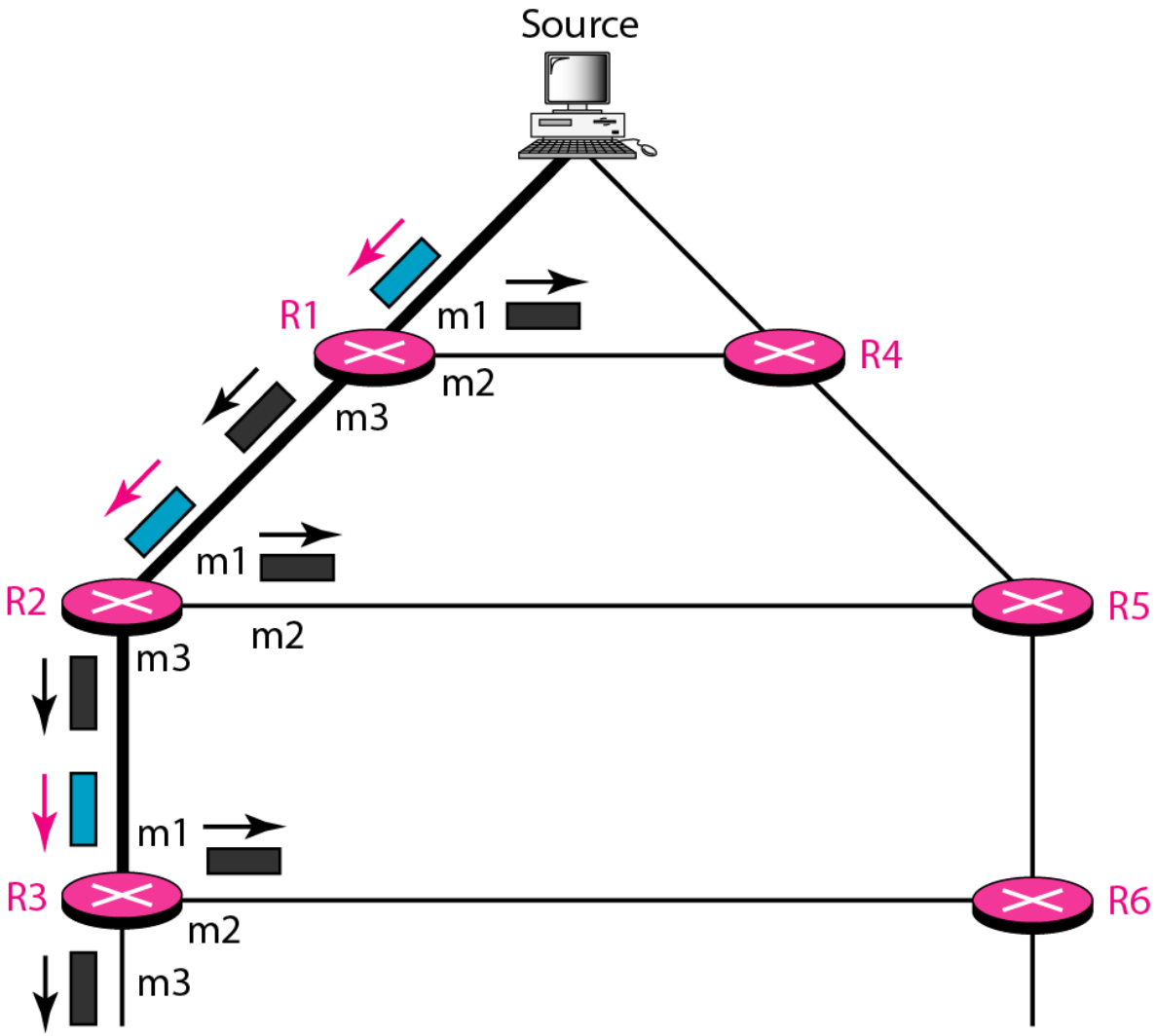
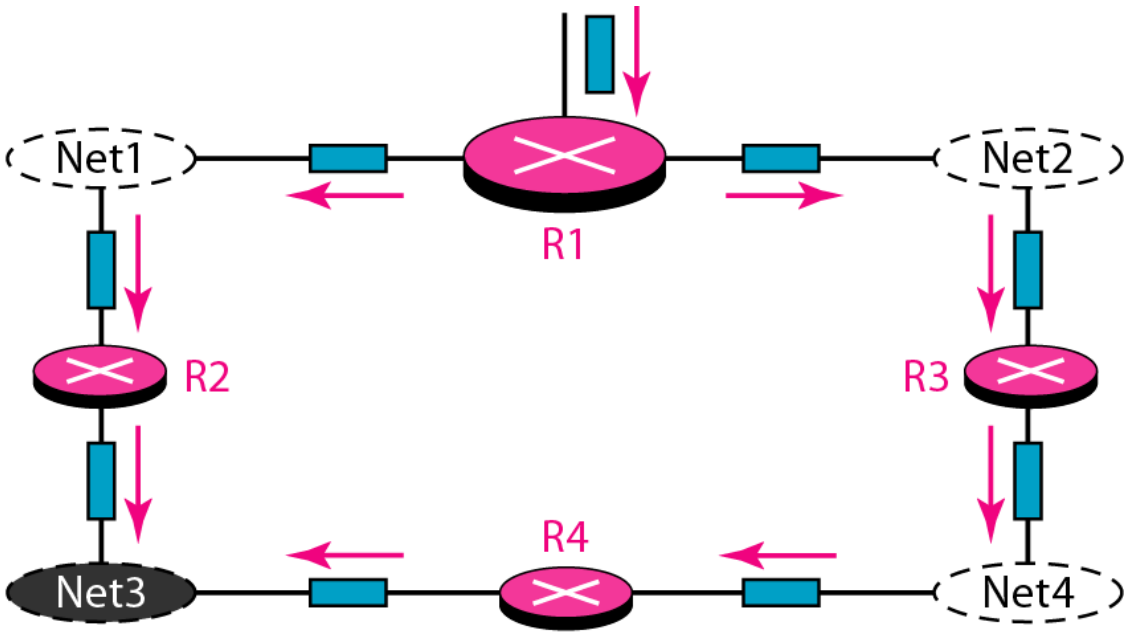
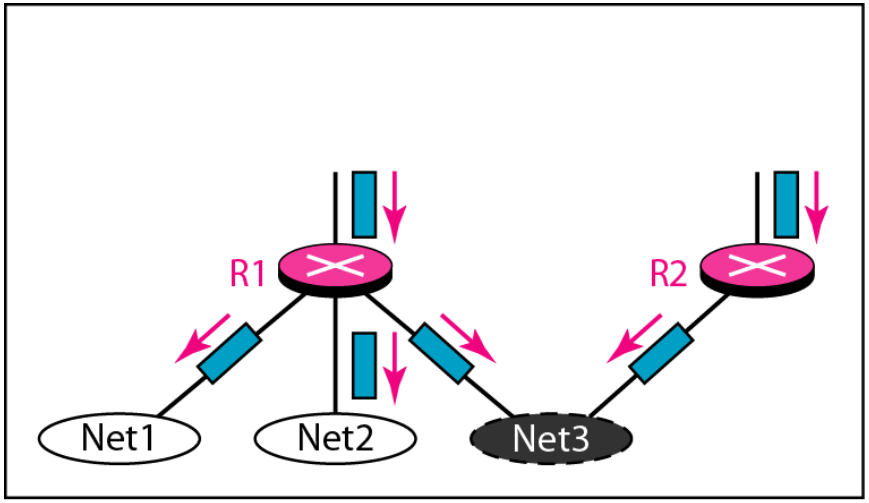


Figure 22.41 *Problem with RPF*

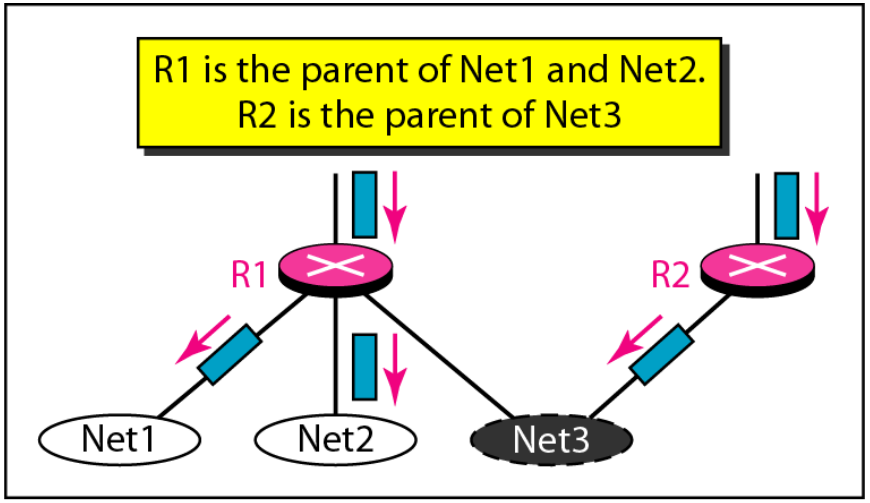


Net3 receives two copies of the packet

Figure 22.42 *RPF Versus RPB*



a. RPF

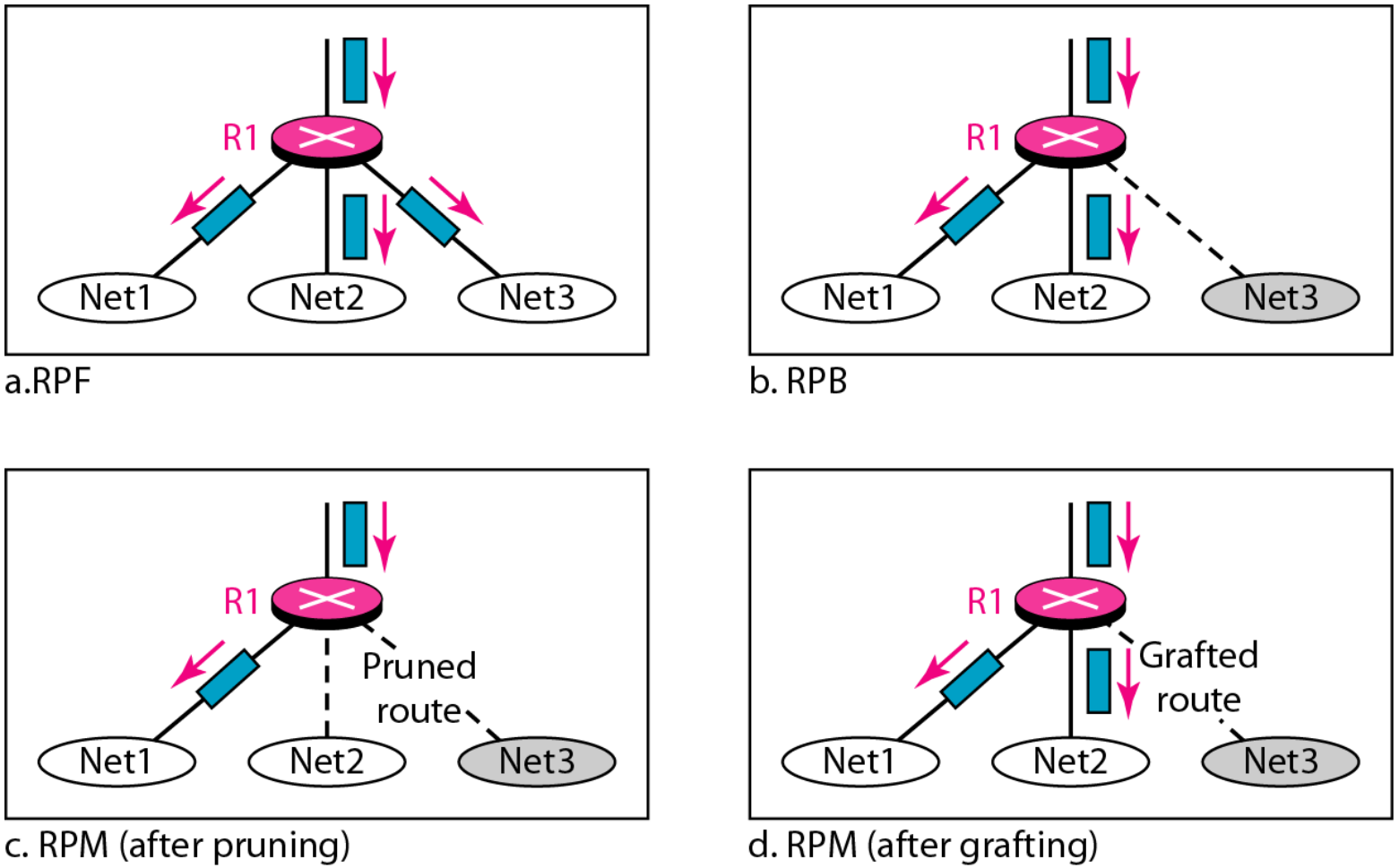


b. RPB

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*

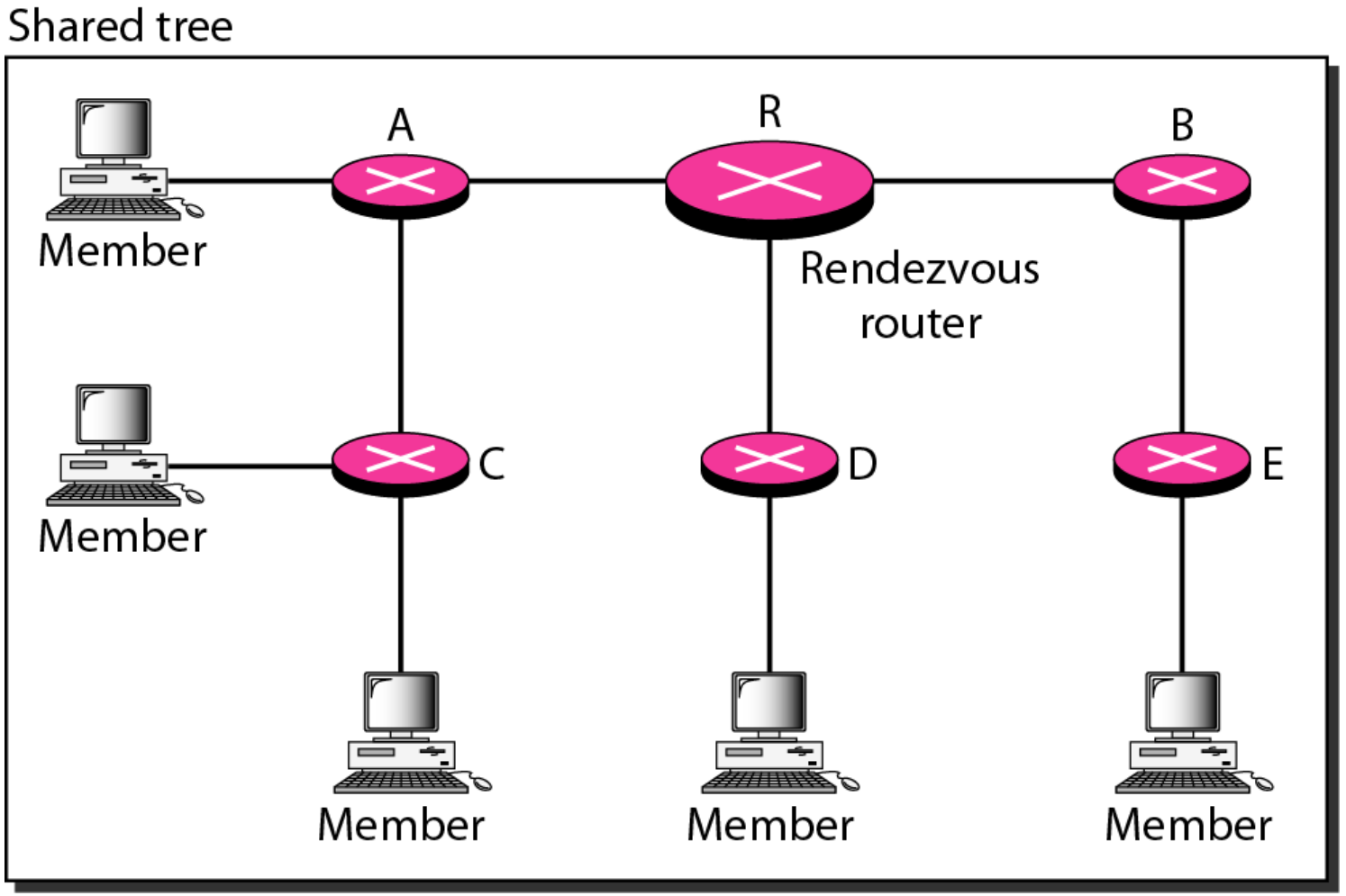
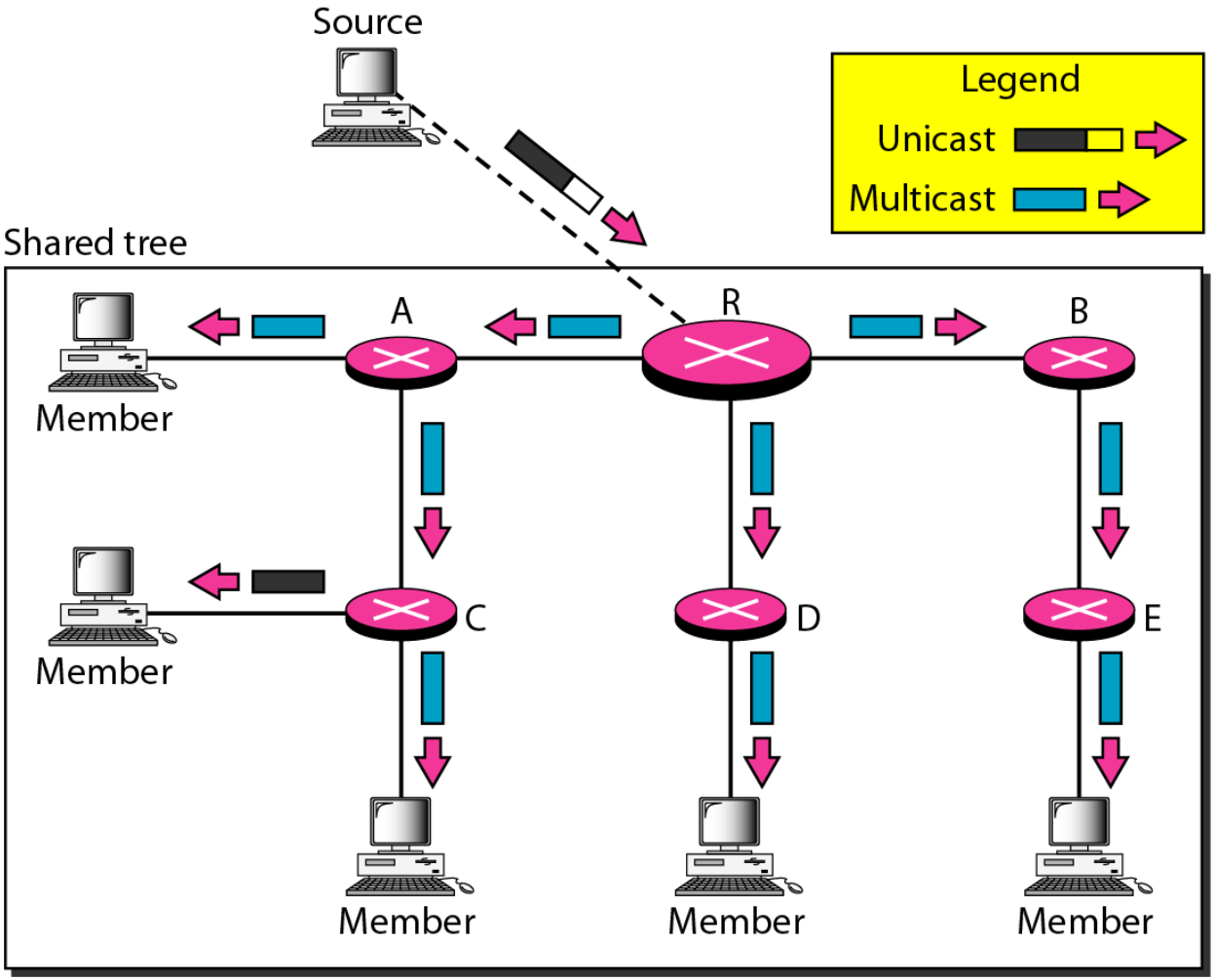


Figure 22.45 *Sending a multicast packet to the rendezvous router*



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.

Note

PIM-DM is used in a dense multicast environment, such as a LAN.

Note

PIM-DM uses RPF and pruning and grafting strategies to handle multicasting.
However, it is independent of the underlying unicast protocol.

Note

PIM-SM is used in a sparse multicast environment such as a WAN.

Note

PIM-SM is similar to CBT but uses a simpler procedure.

Figure 22.46 *Logical tunneling*

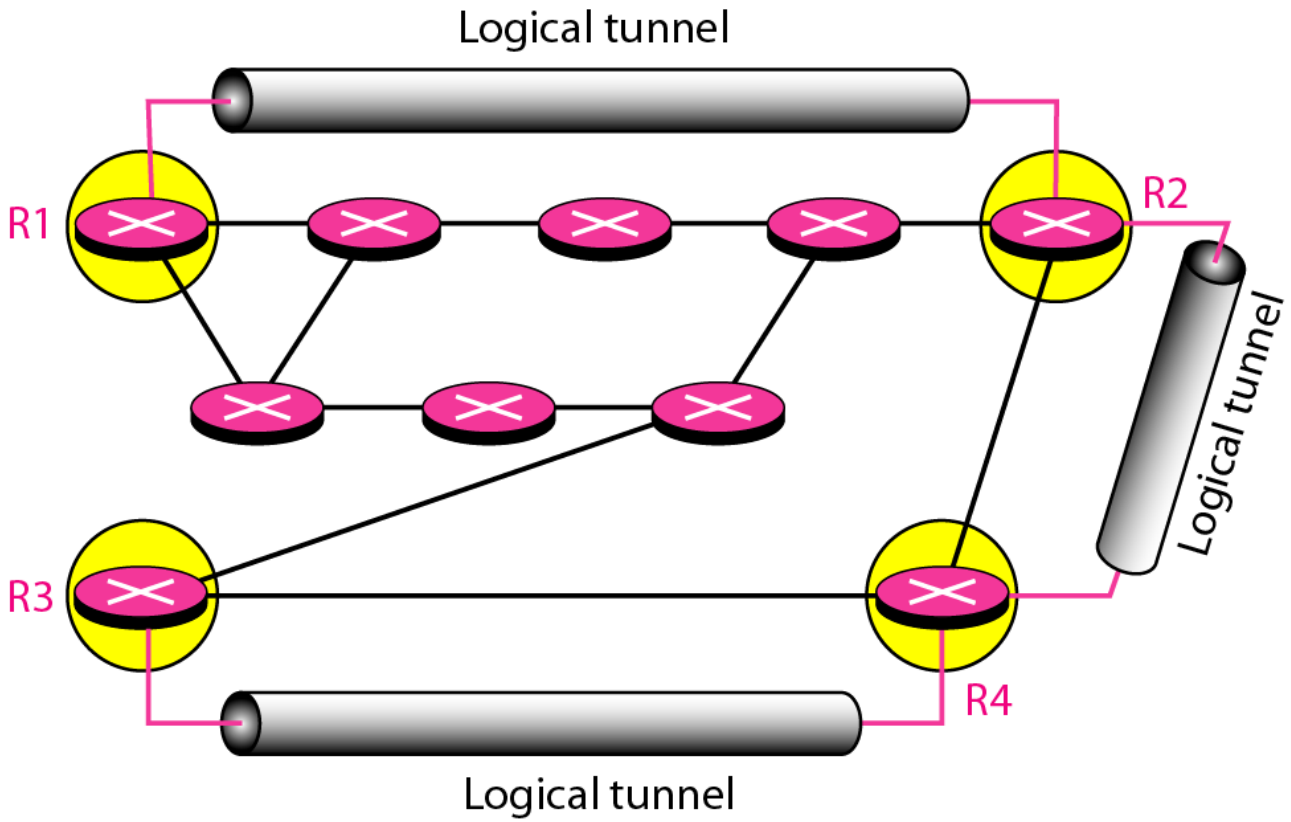


Figure 22.47 *MBONE*