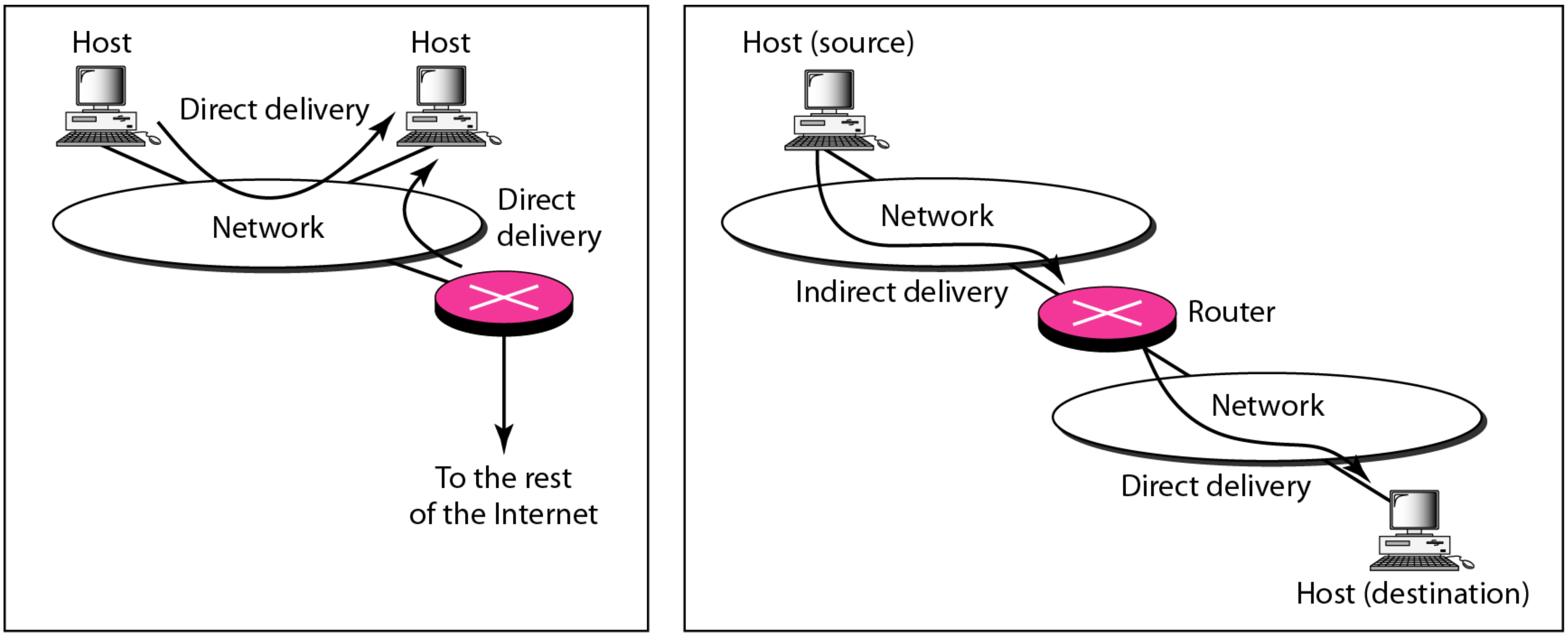
Network Layer: Delivery, Forwarding, and Routing

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.

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.***

Figure 22.2 *Route method versus next-hop method*

a. Routing tables based on route

b. Routing tables based on next hop

Routing table for host A

Destination	Route
Host B	R1, R2, host B

Destination	Next hop
Host B	R1

Routing table for R1

Destination	Route
Host B	R2, host B

Destination	Next hop
Host B	R2

Routing table for R2

Destination	Route
Host B	Host B

Destination	Next hop
Host B	---

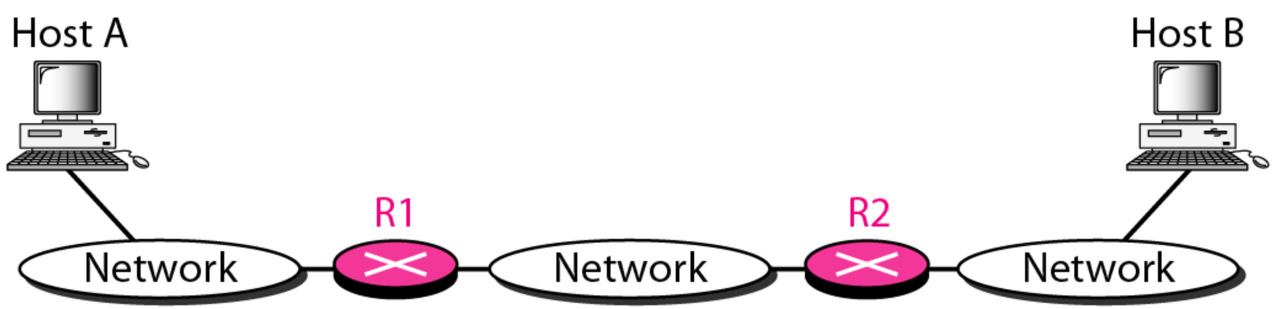


Figure 22.3 *Host-specific versus network-specific method*

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

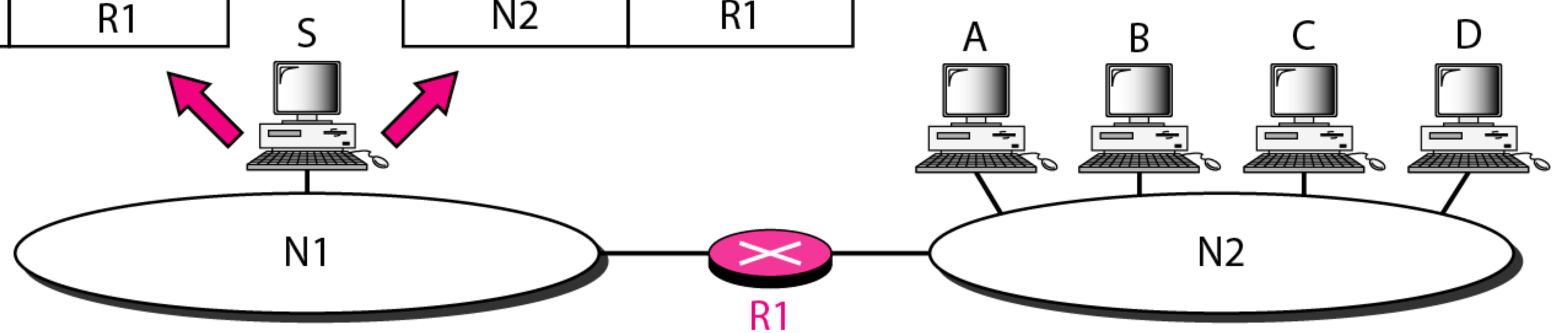


Figure 22.4 *Default method*

Destination	Next hop
N2	R1
Any other	R2

Routing table for host A

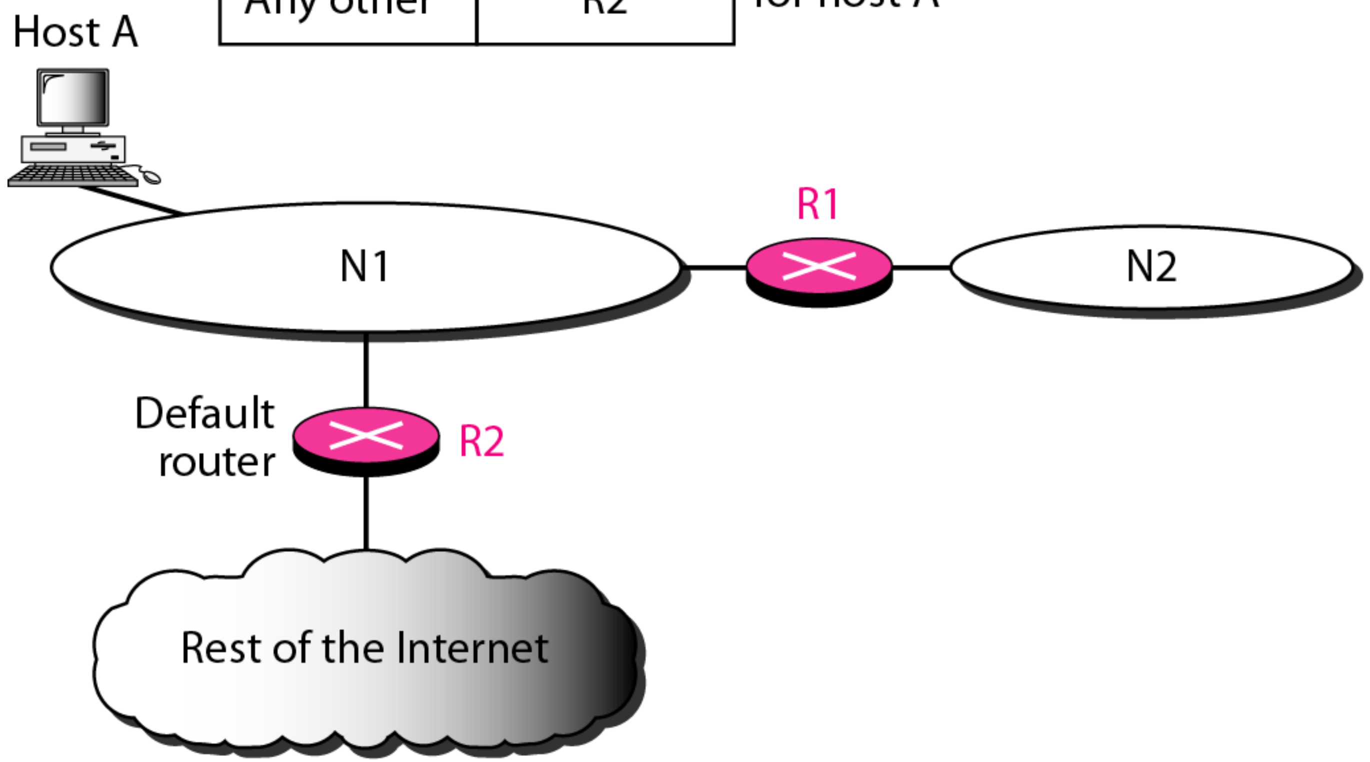
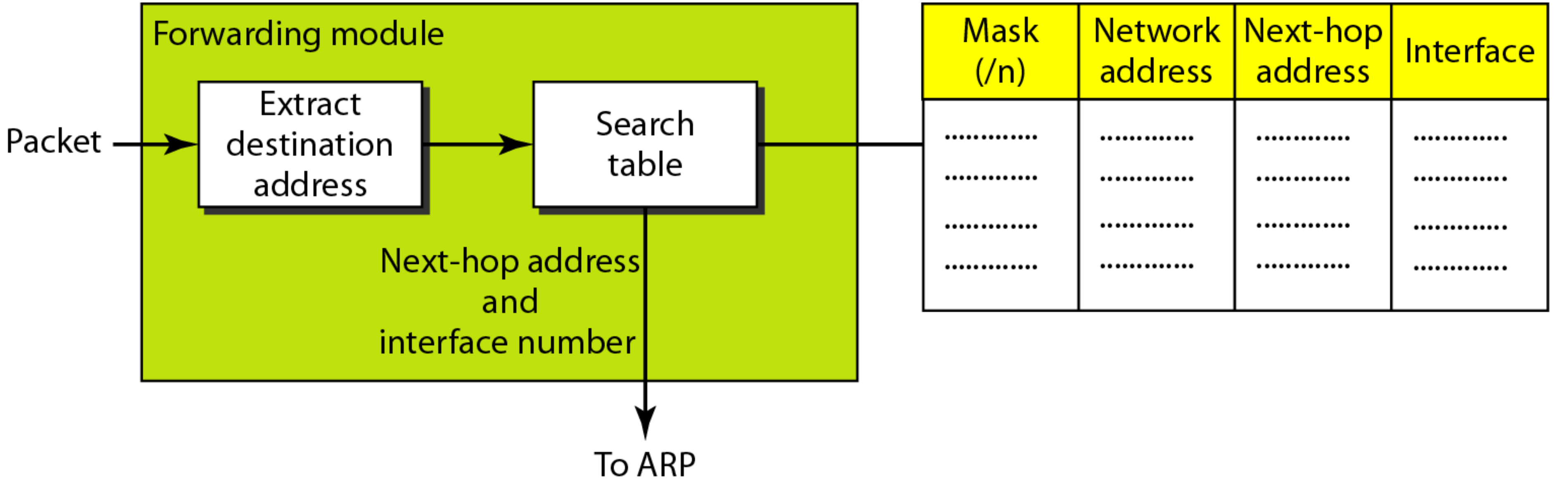


Figure 22.5 *Simplified forwarding module in classless address*



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*

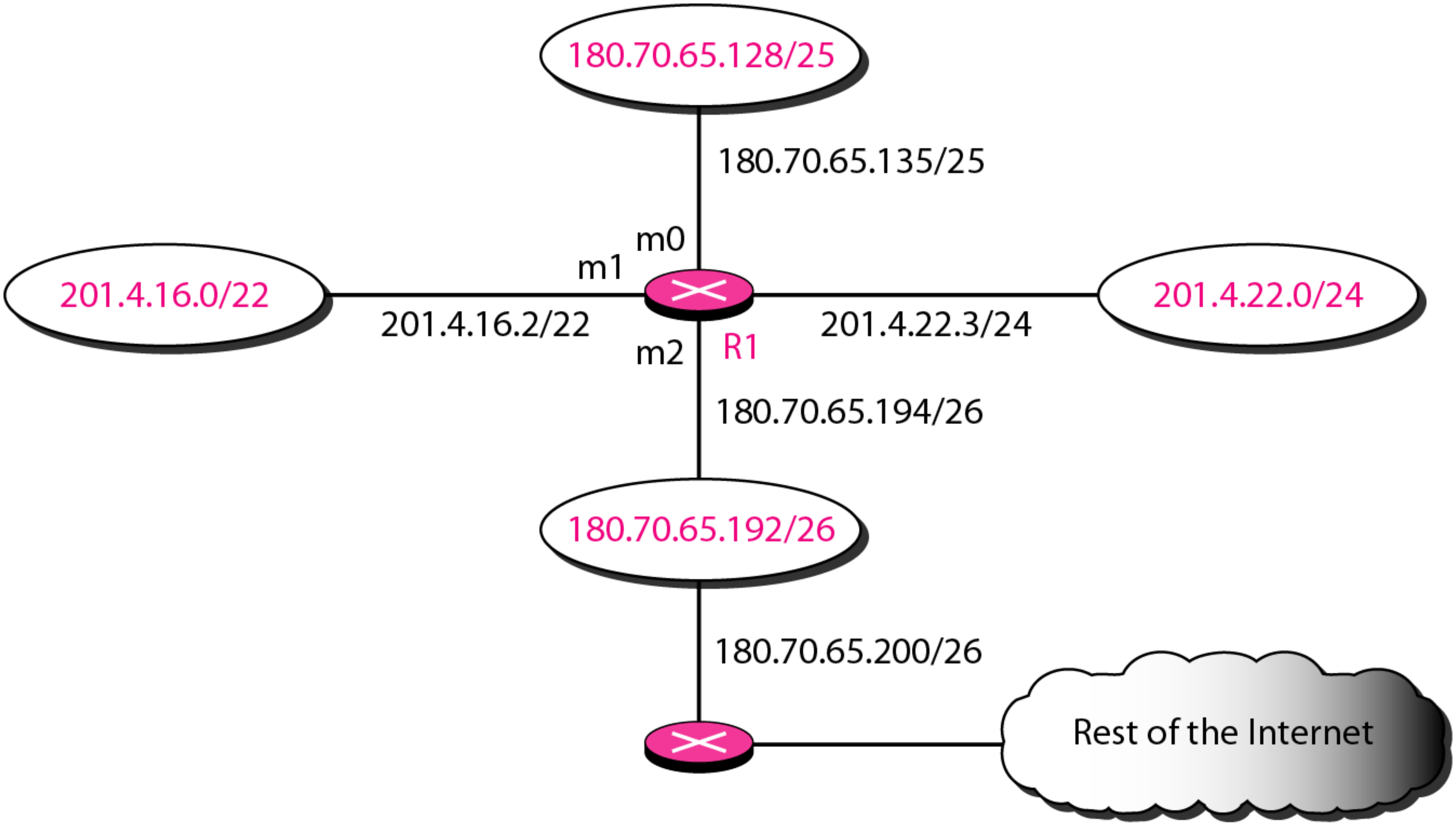


Table 22.1 ***Routing table for router R1 in Figure 22.6***

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

Figure 22.10 *Common fields in a dynamic routing table*

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

- **Mask:** Mask applied for the entry
- **Network Address:** This field defines the network address to which the packet is finally delivered.
- **Next-hop Address:** Address of the next-hop router to which the packet is delivered
- **Interface:** Name of interface

- **Flags:**
  - **U (Up):** It indicates that router is up and running.

  - **G (Gateway):** The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery (indirect). When this flag is missing, it means the destination is in this network (direct delivery).

- **H (Host-specific):** It indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.
- **D (added by redirection):** The D indicates that routing information for this destination has been added to the host routing table by a redirection table by a redirection message from ICMP.
- **M (Modified by redirection):** It indicates that the routing information for this destination has been modified by a redirection message from ICMP.

- **Reference Count:** Indicates number of users of this route at the moment.
- **Use:** Indicates number of packets transmitted through this route for corresponding destination.

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

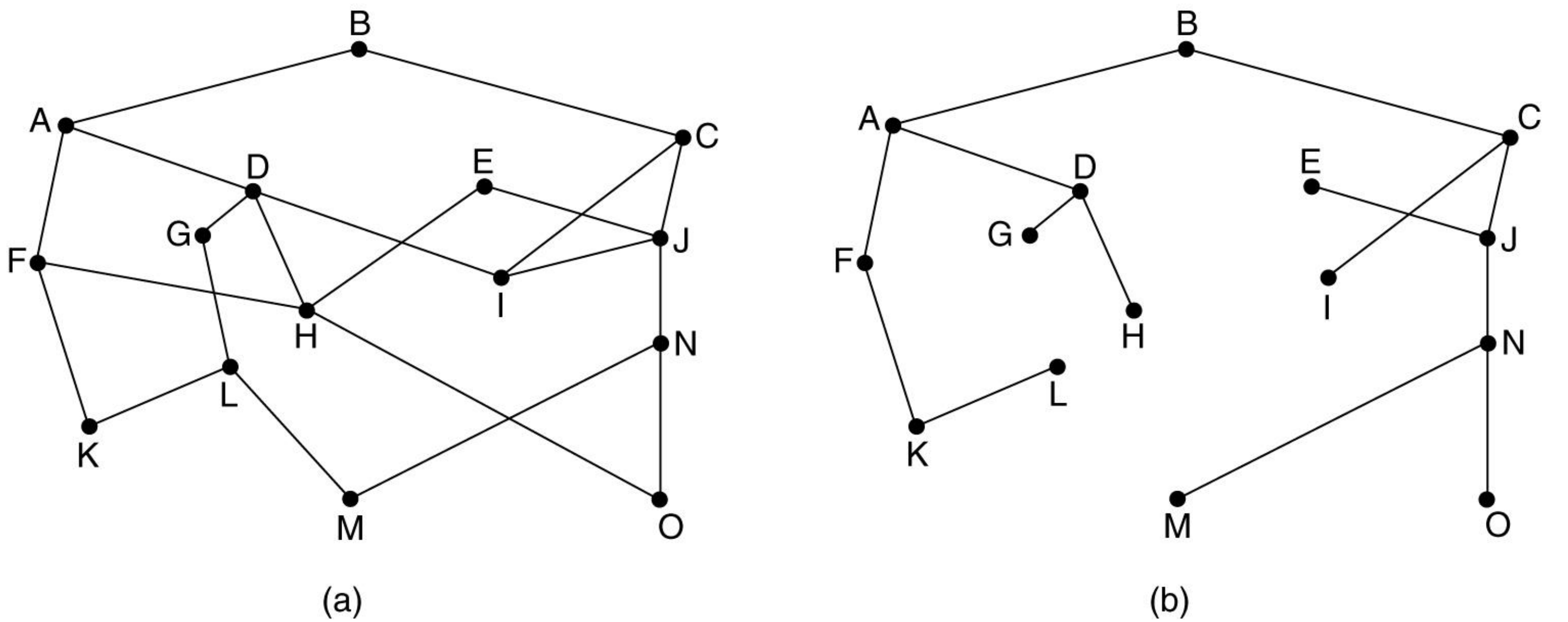
The Optimality Principle

- Before we get into specific algorithms, it may be helpful to note that one can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the **optimality principle** (Bellman,1957).

- It states that if router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. To see this, call the part of the route from I to J $r1$ and the rest of the route $r2$. If a route better than $r2$ existed from J to K, it could be concatenated with $r1$ to improve the route from I to K, contradicting our statement that $r1\ r\ 2$ is optimal.

- As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree.** Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist.

The Optimality Principle



(a) A network. (b) A sink tree for router *B*.

Figure 22.12 *Autonomous systems*

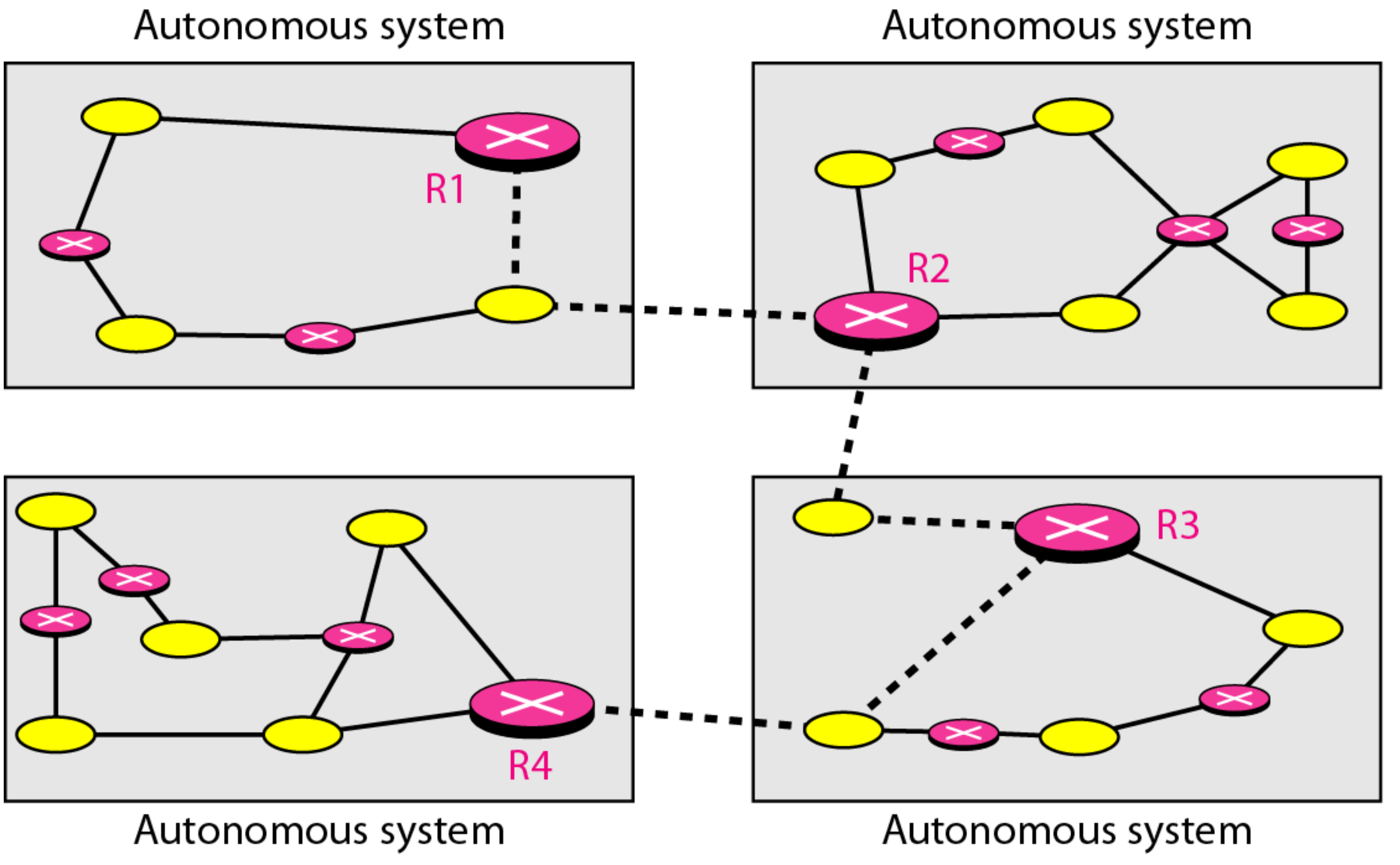


Figure 22.13 *Popular routing protocols*

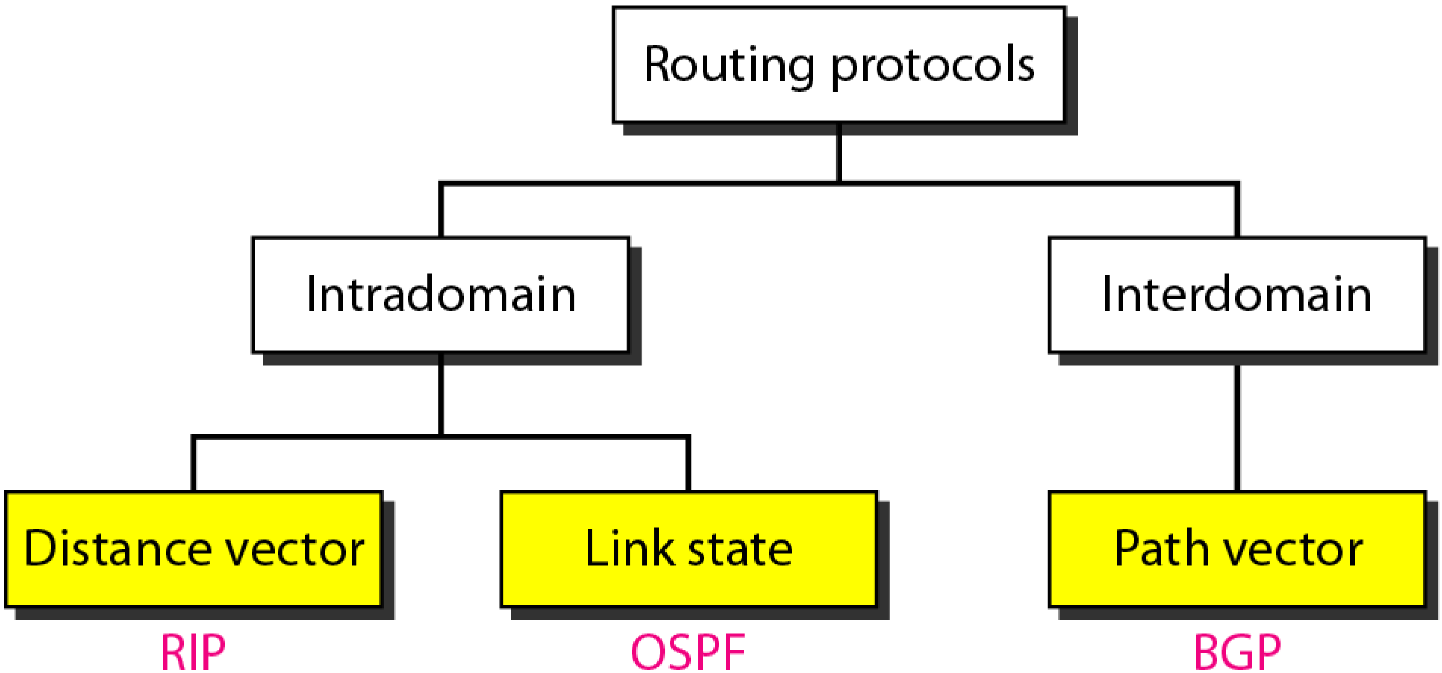
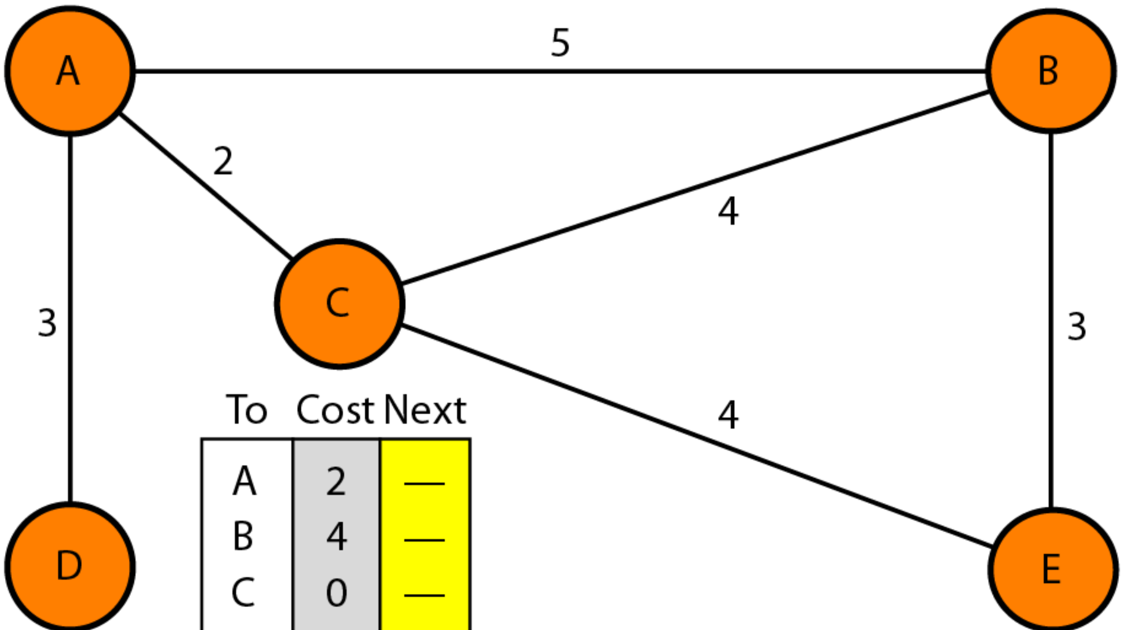


Figure 22.14 *Distance vector routing tables*



A's table

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

B's table

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

C's table

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

D's table

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

E's table

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

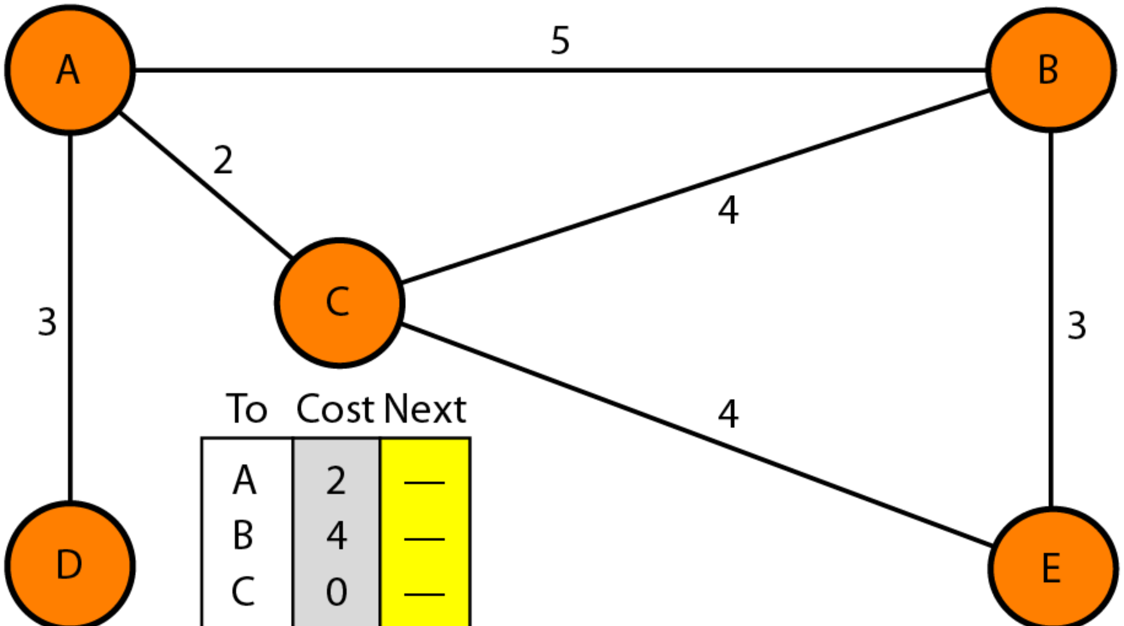
Figure 22.15 *Initialization of tables in distance vector routing*

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.

The heart of distance-vector routing is the famous **Bellman-Ford** equation.

This equation is used to find the least cost (shortest distance) between a source node, ***x***, and a destination node, ***y***, through some intermediary nodes (**a, b, c,** . . .) when the costs between the source and the intermediary nodes and the least costs between the intermediary nodes and the destination are given.

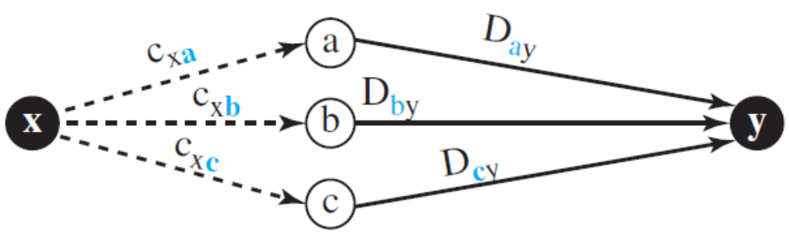
The following shows the general case in which D_{ij} is the shortest distance and c_{ij} is the cost between nodes ***i*** and ***j***.

$$Dxy = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), ..\}$$

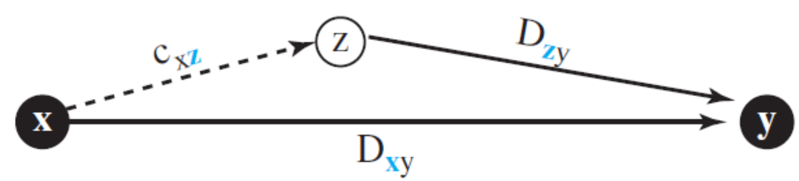
Bellman-Ford Equation

In distance-vector routing, normally we want to update an existing least cost with a least cost through an intermediary node, such as *z*, if the latter is shorter. In this case, the equation becomes simpler, as shown below:

$$D_{xy} = \min\{D_{xy}, (c_{xz} + D_{zy})\}$$



a. General case with three intermediate nodes



b. Updating a path with a new route

Figure 22.16 *Updating in distance vector routing*

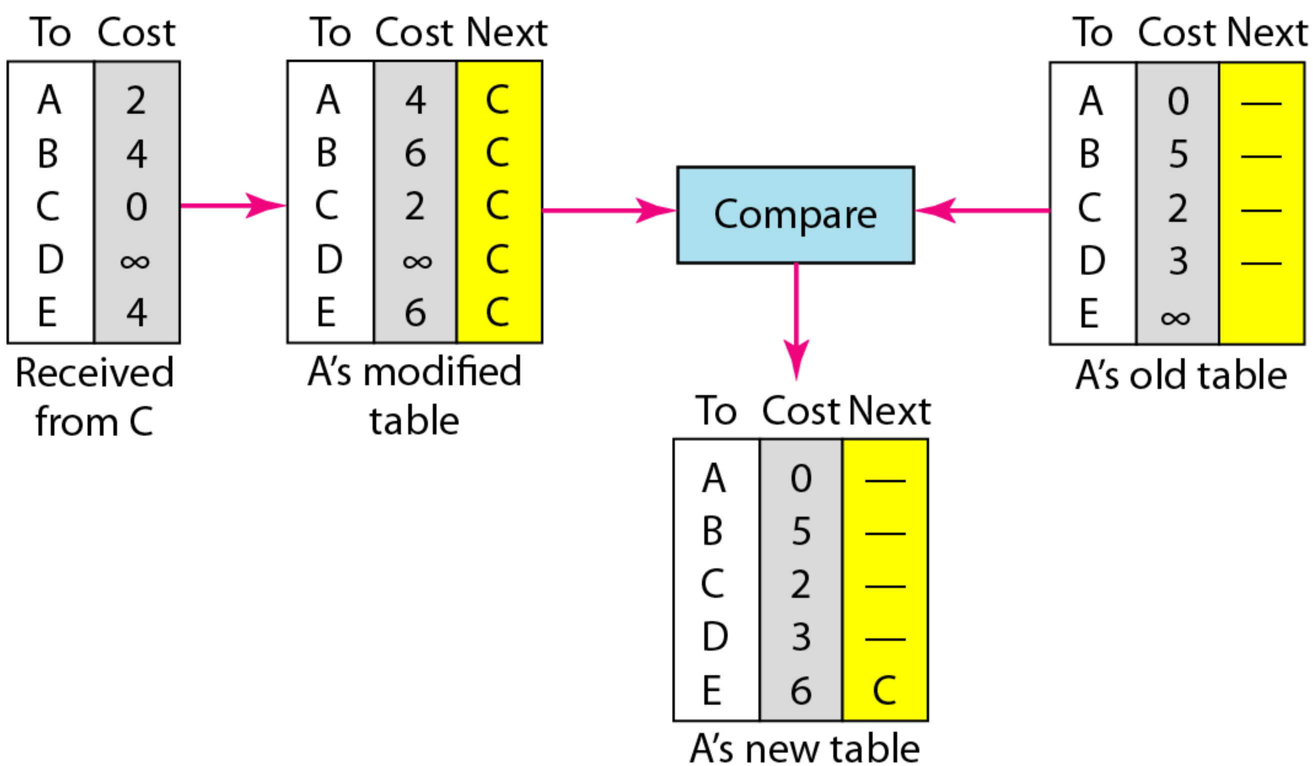


Figure 22.17 *Two-node instability*

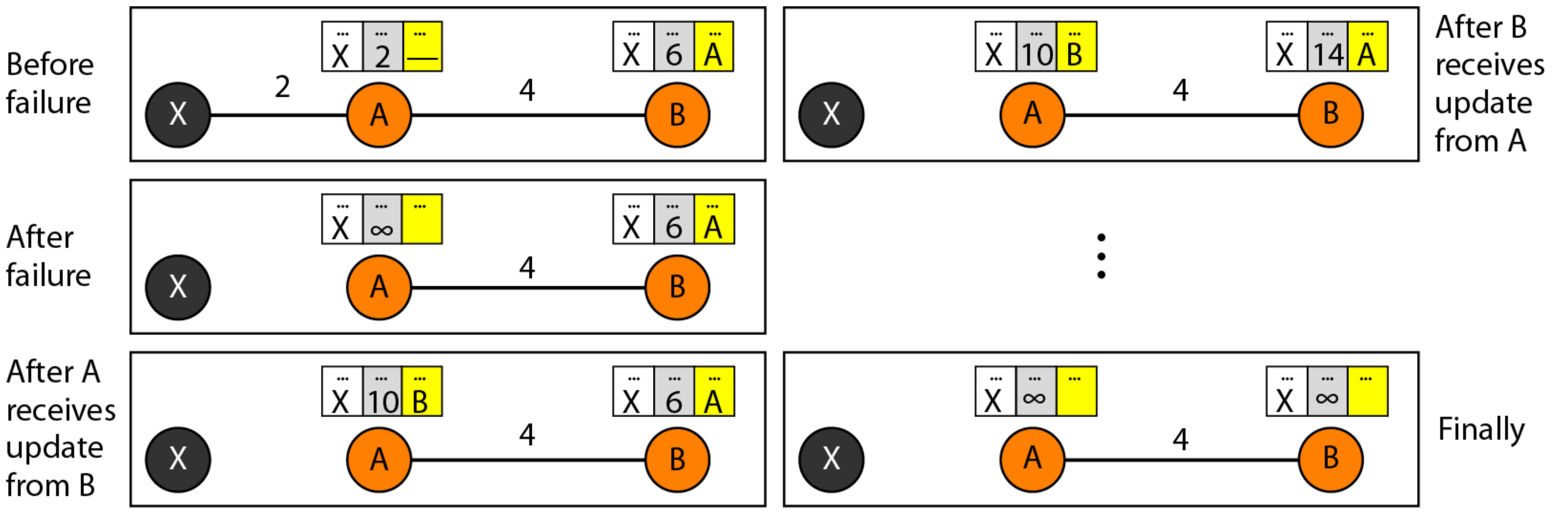


Figure 22.18 *Three-node instability*

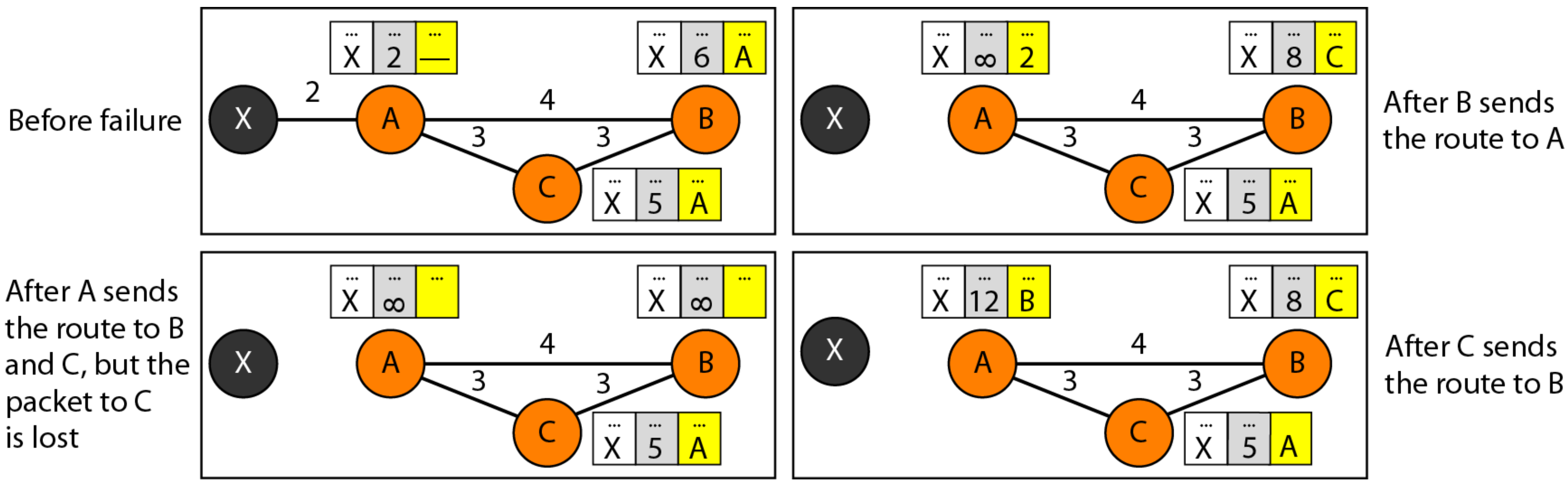
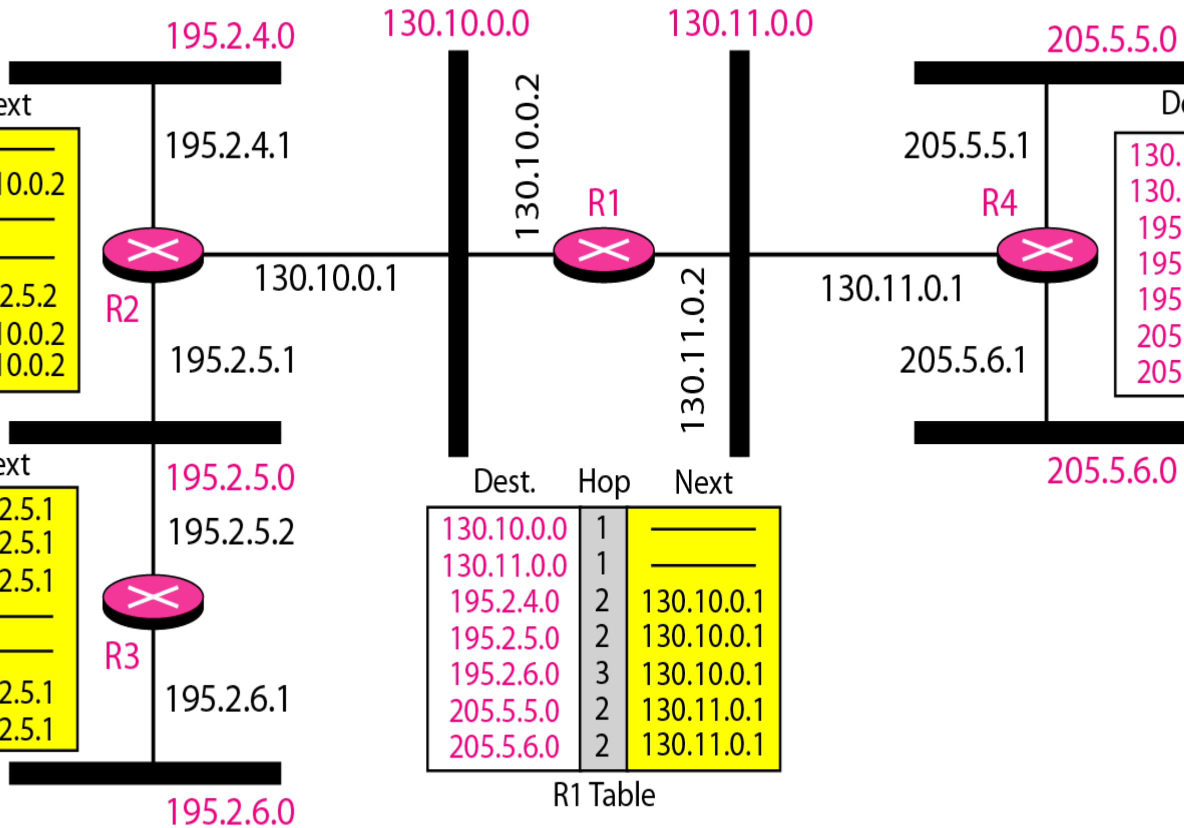


Figure 22.19 *Example of a domain using RIP*



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

Figure 22.20 *Concept of link state routing*

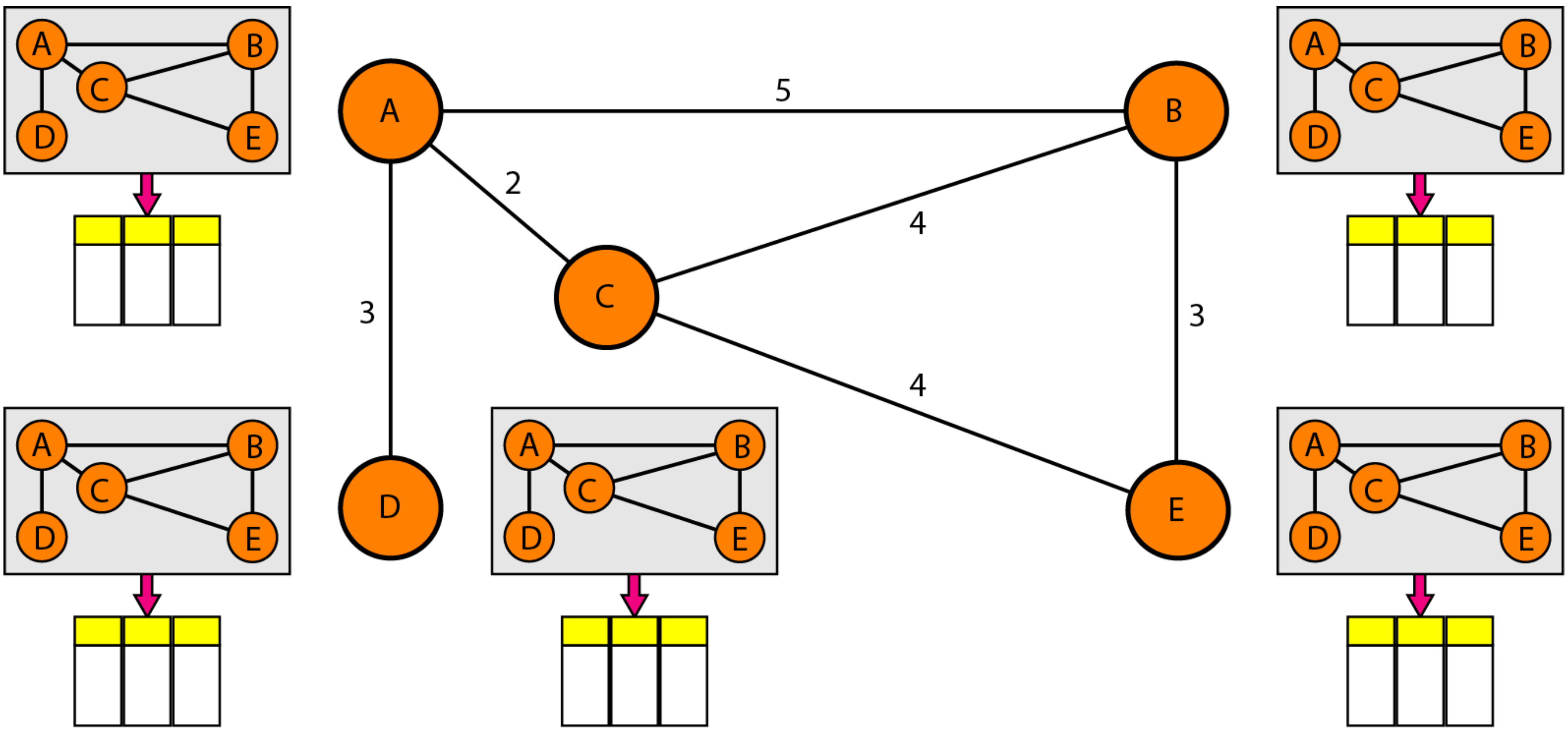


Figure 22.21 *Link state knowledge*

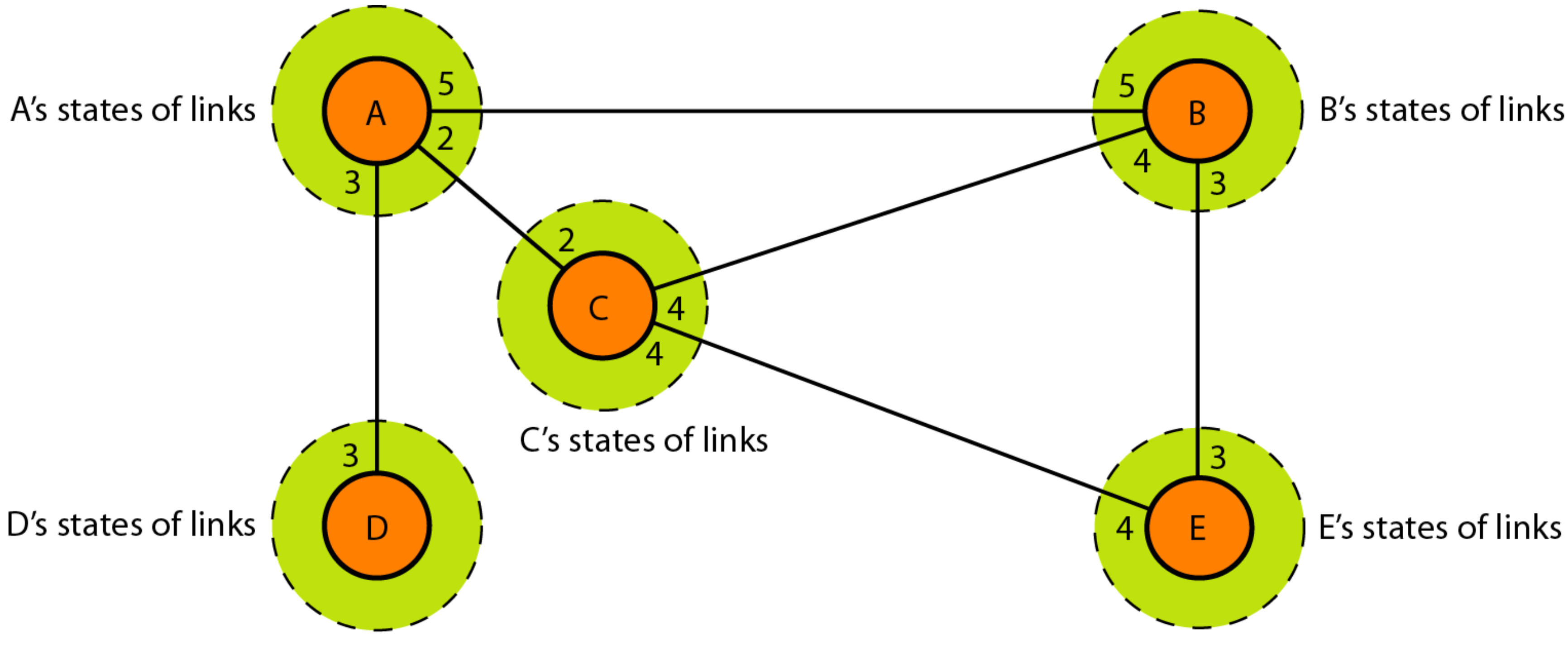


Figure 22.22 *Dijkstra algorithm*

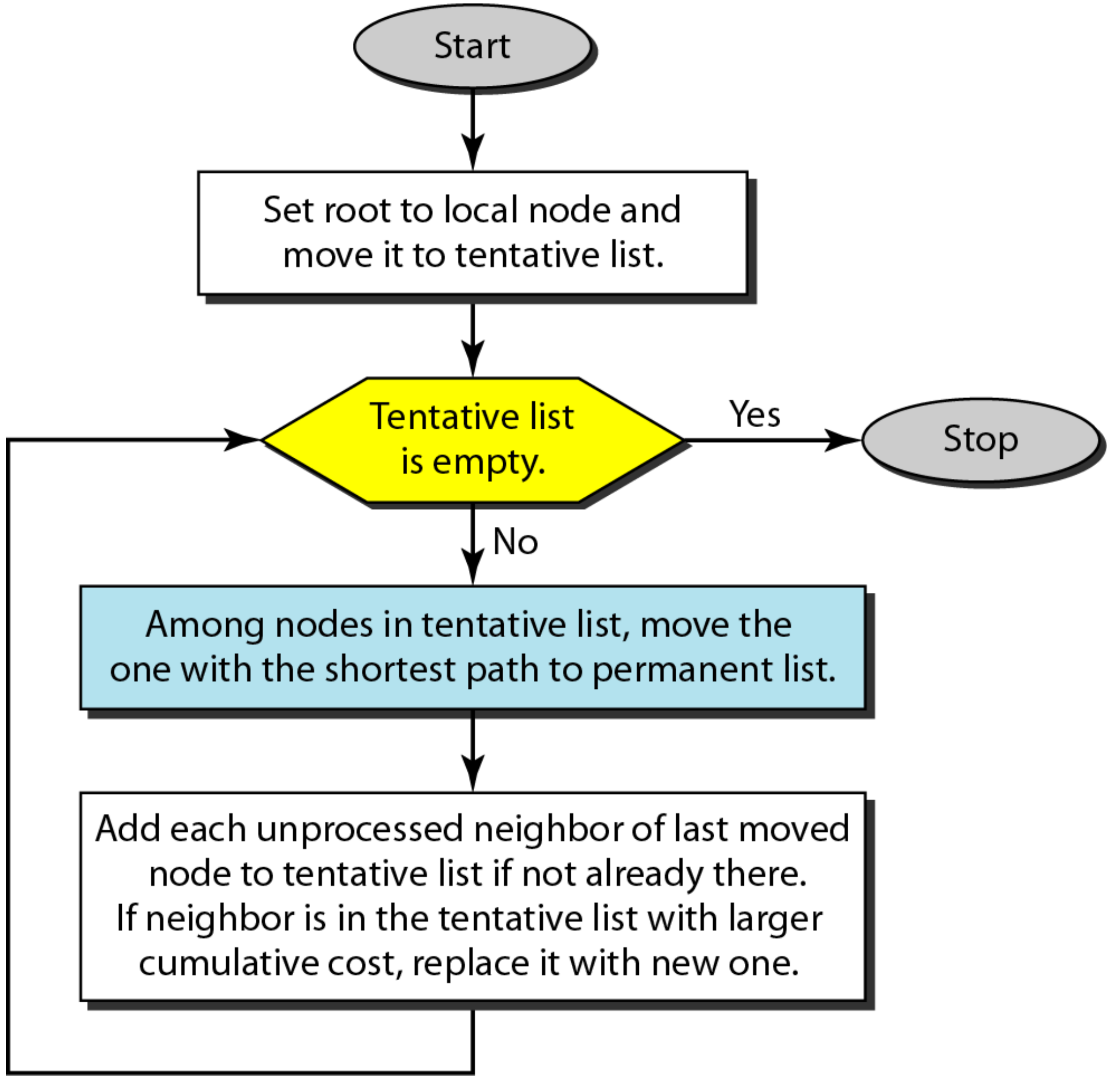


Figure 22.23 *Example of formation of shortest path tree*

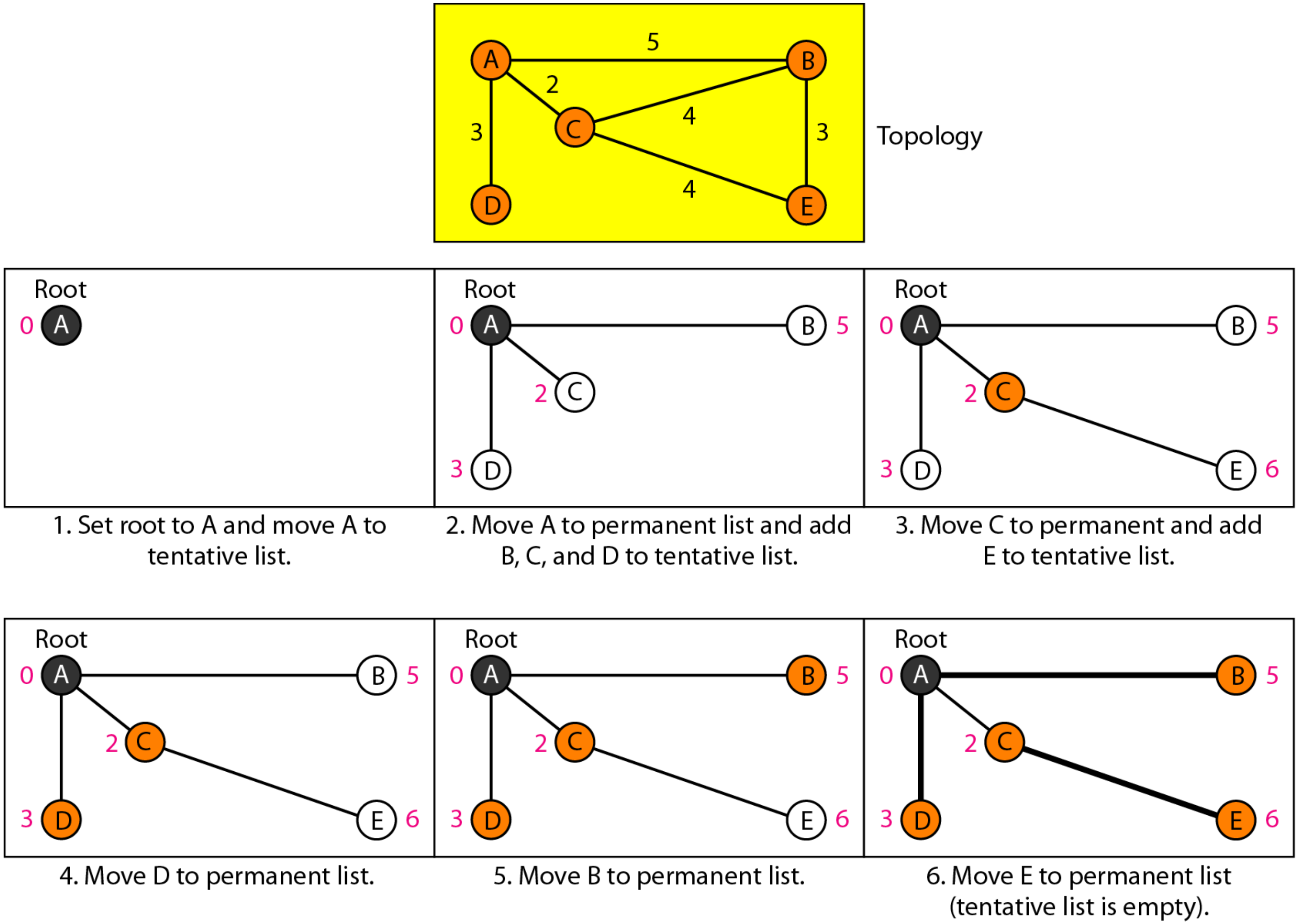
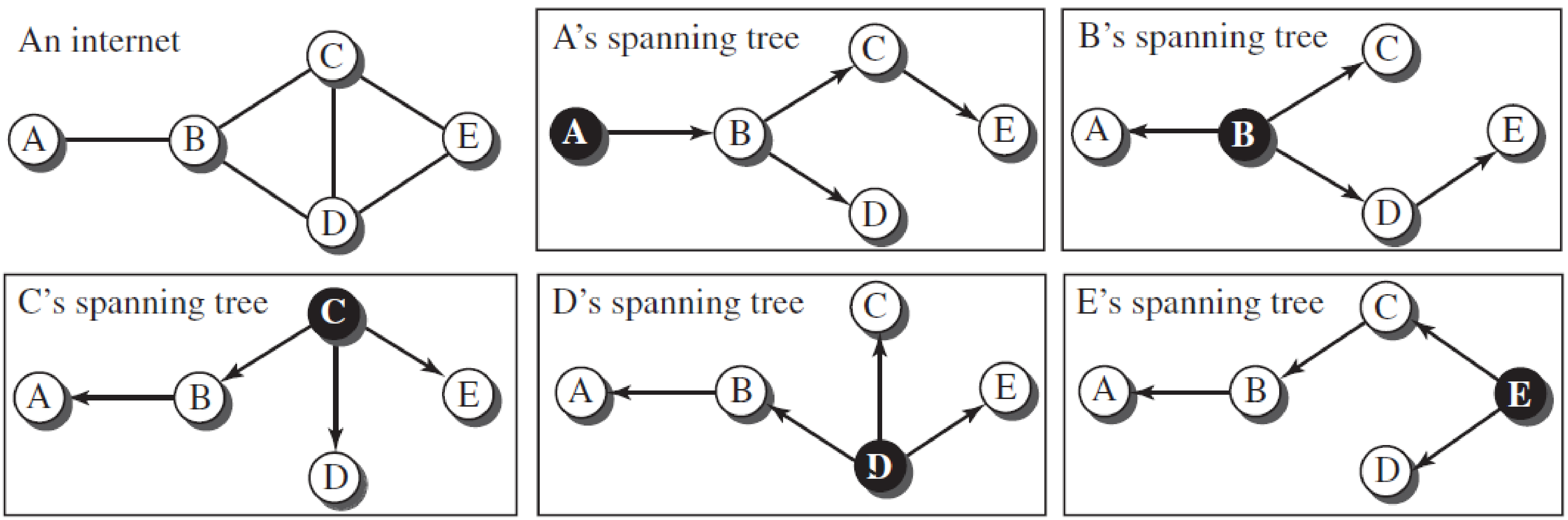


Table 22.2 ***Routing table for node A***

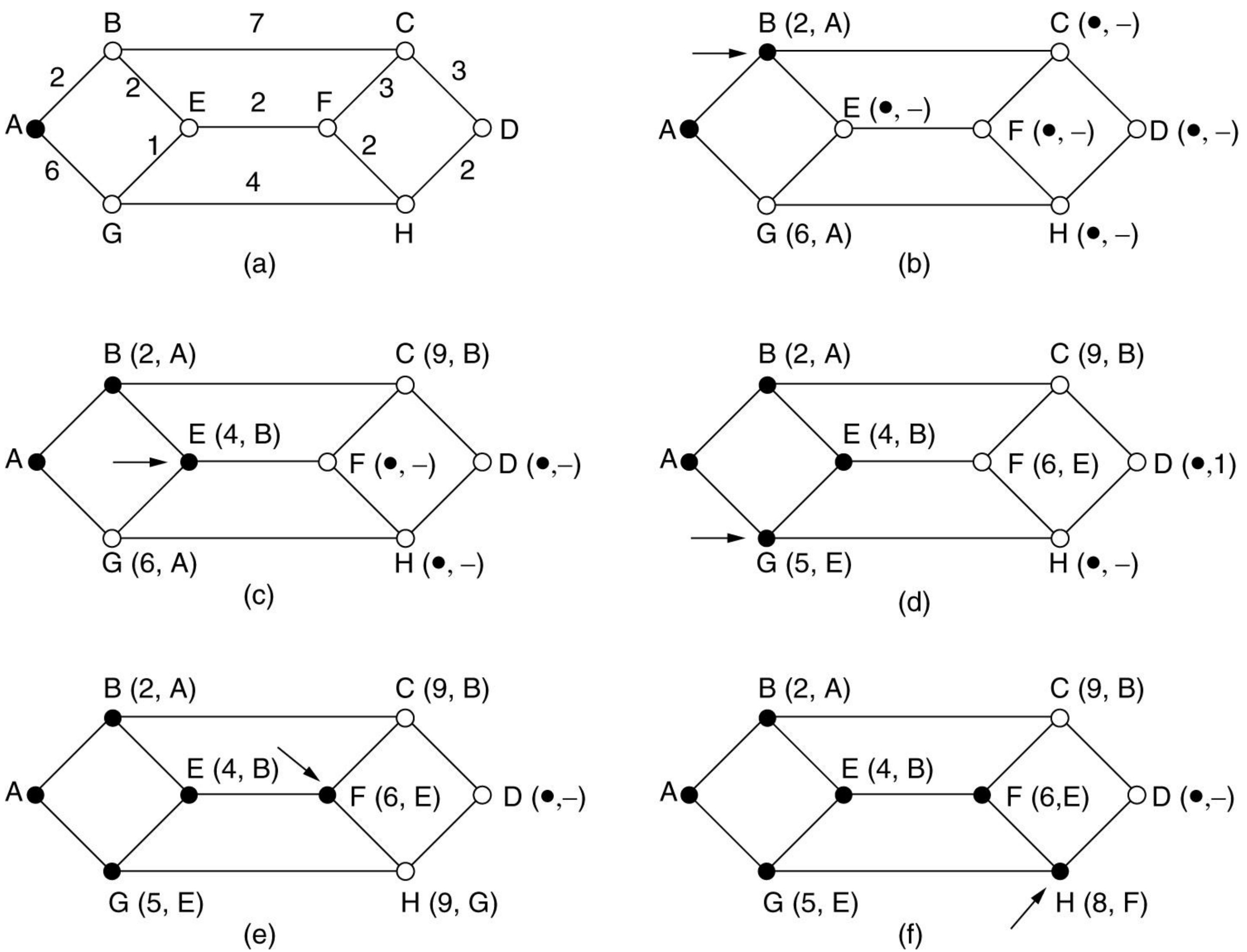
Node	*Cost*	*Next Router*
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Path vector routing



Path($\boldsymbol{x}$, $\boldsymbol{y}$) = best {Path(x, y), [($\boldsymbol{x}$ + Path($\mathbf{v}$, $\boldsymbol{y}$)]} for all $\mathbf{v}$'s in the internet

Shortest Path Routing



The first 5 steps used in computing the shortest path from A to D.
The arrows indicate the working node.

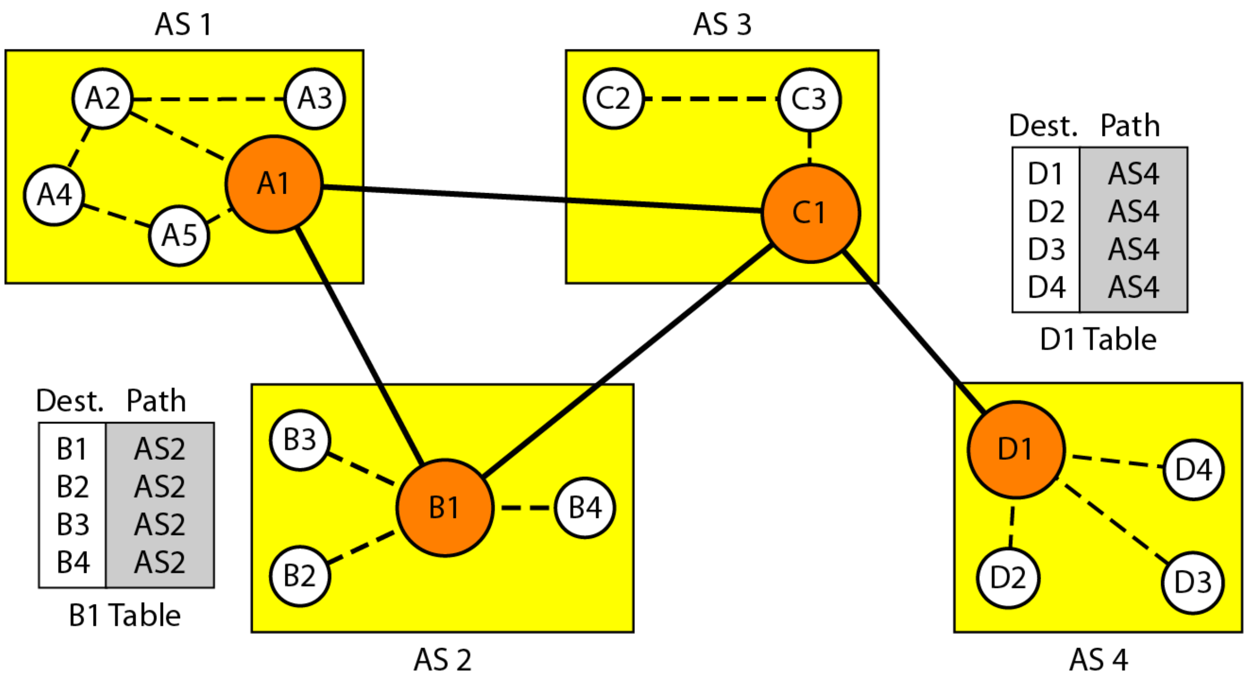
Figure 22.30 *Initial routing tables in path vector routing*

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*

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

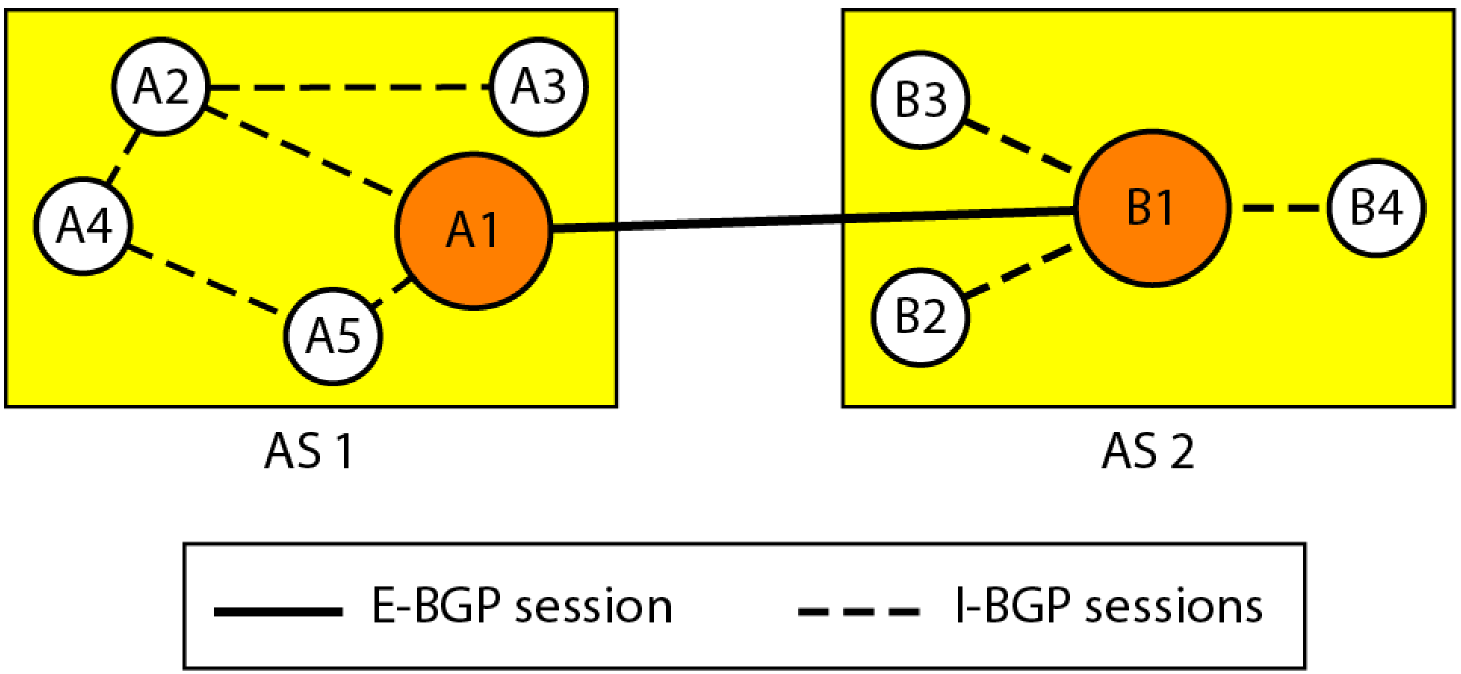
Dest.	Path
A1	AS3-AS1
…	
A5	AS3-AS1
B1	AS3-AS2
…	…
B4	AS3-AS2
C1	AS3
…	…
C3	AS3
D1	AS3-AS4
…	…
D4	AS3-AS4

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

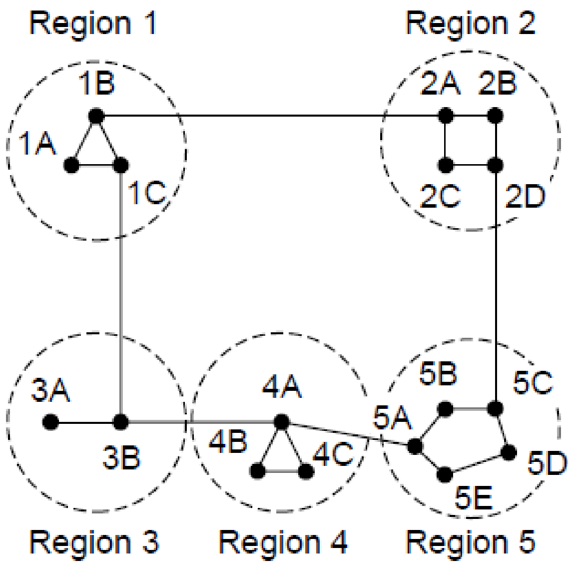
Figure 22.32 *Internal and external BGP sessions*



Hierarchical Routing

- **As networks grow in size, the router routing tables grow proportionally.**
- Not only is router memory consumed by ever-increasing tables, but **more CPU time** is needed to scan them and more bandwidth is needed to send status reports about them.
- At a certain point, the **network may grow to the point where it is no longer feasible** for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.
- When **hierarchical routing is used, the routers are divided into what we will call regions.**
- Each router knows all the details about how to route packets to destinations within its own region but knows nothing about the internal structure of other regions.
- When different networks are interconnected, it is natural to regard each one as a separate region to free the routers in one network from having to know the topological structure of the other ones.
- For huge networks, a two-level hierarchy may be insufficient; it may be necessary to **group the regions into clusters, the clusters into zones, the zones into groups, and so on,** until we run out of names for aggregations.

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)

Hierarchical routing.