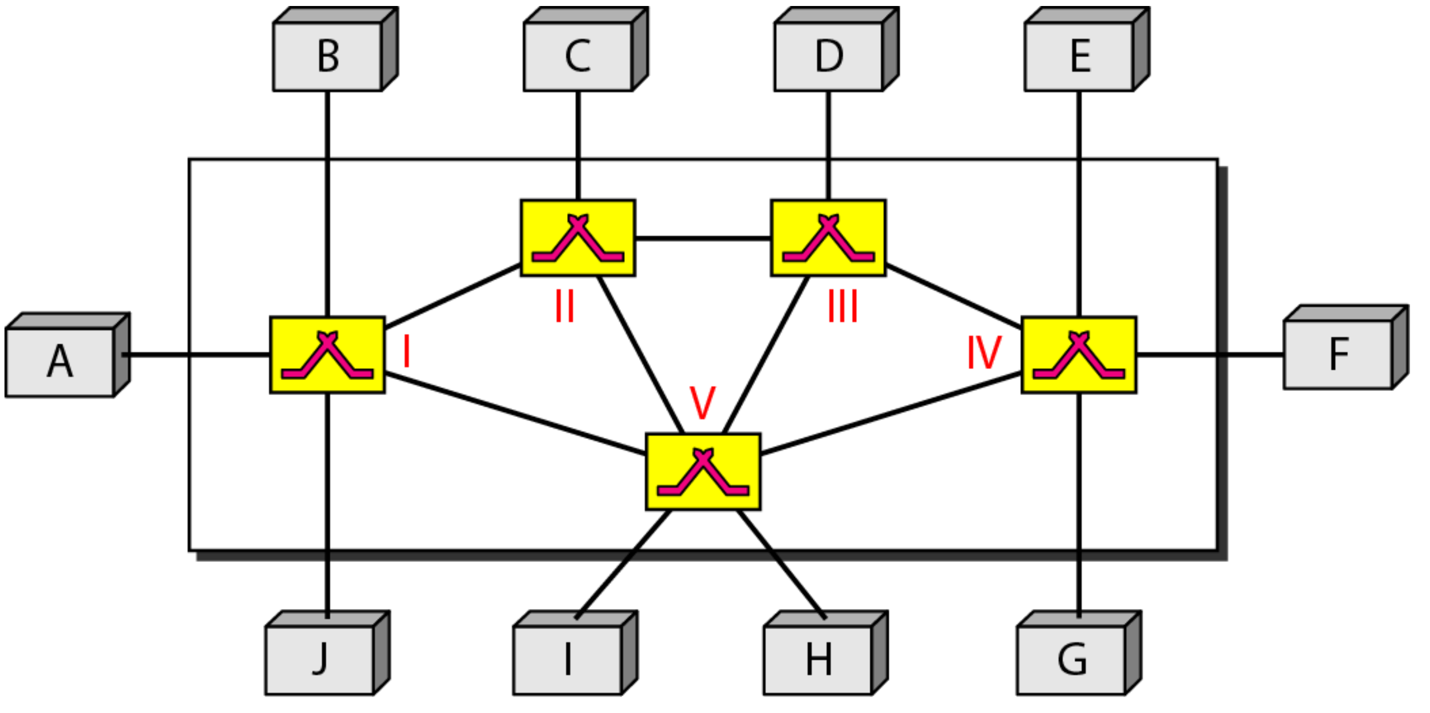
# Lecture 8

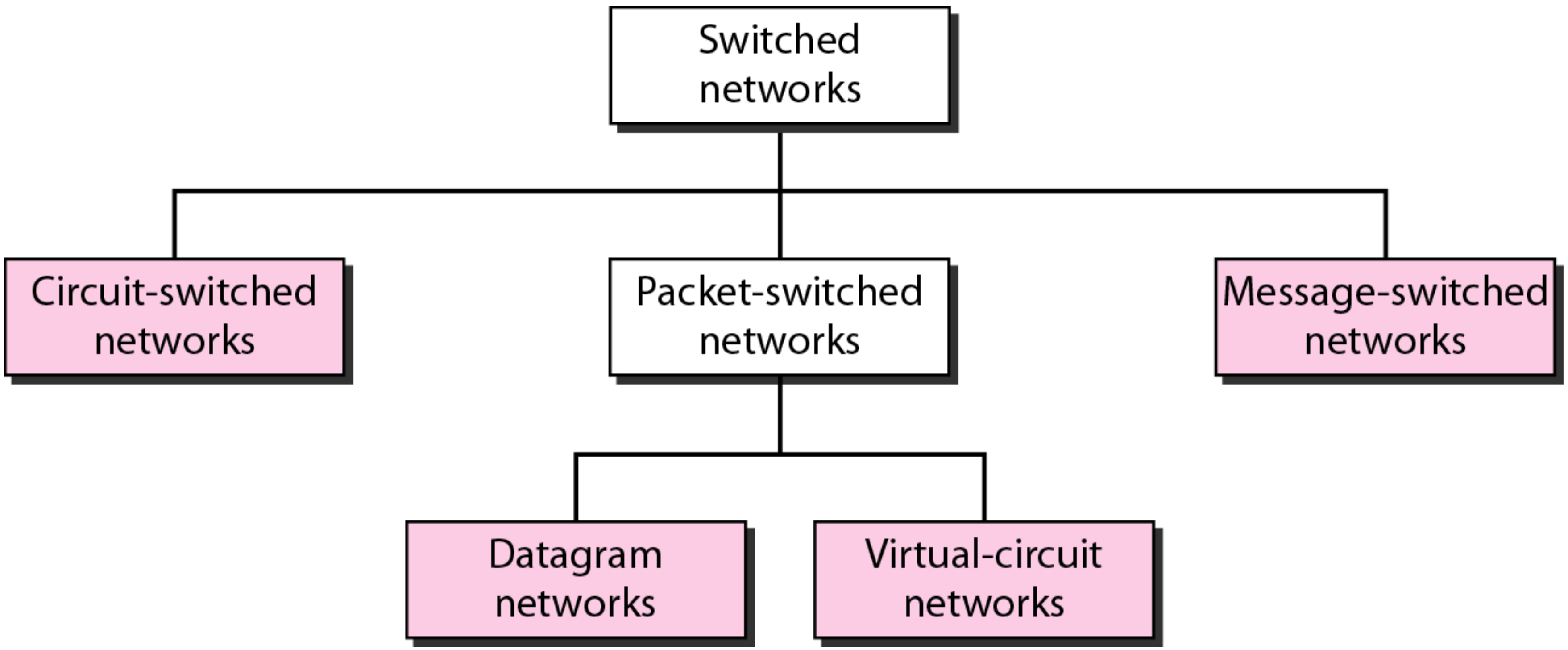
# Packet Switched Networks

## (Chapters 8, 22, 24 of “Data Communications and Networking” [B. A. Forouzan])

## Figure 8.1 *Switched network*



## Figure 8.2 *Taxonomy of switched networks*



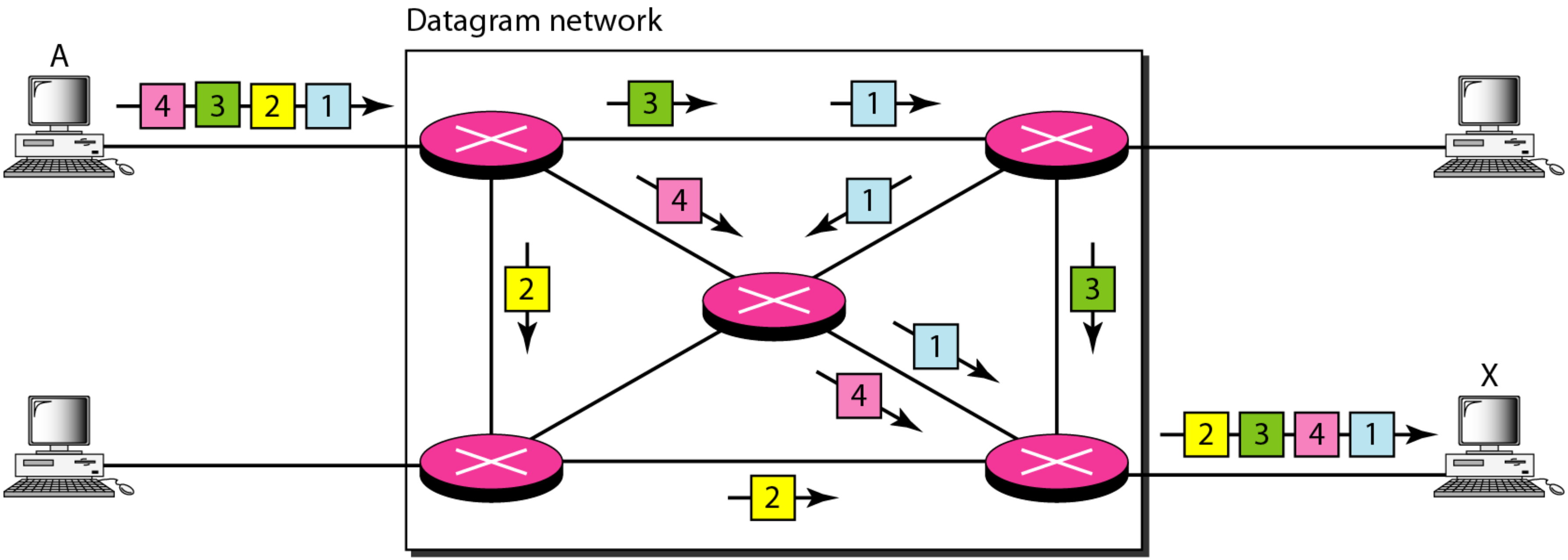
# 8-2 DATAGRAM NETWORKS

***In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol.***

**Note**

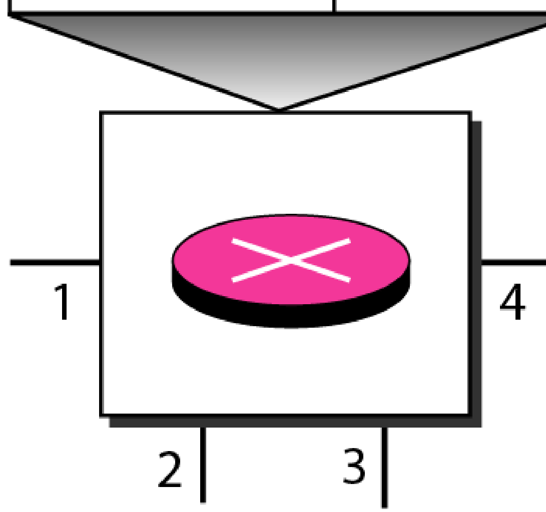
**In a packet-switched network, there is no resource reservation; resources are allocated on demand.**

## Figure 8.7 *A datagram network with five switches (routers)*



## **Figure 8.8** *Routing table in a datagram network*

| Destination address | Output port |
|---|---|
| 1232 | 1 |
| 4150 | 2 |
| ⋮ | ⋮ |
| 9130 | 3 |



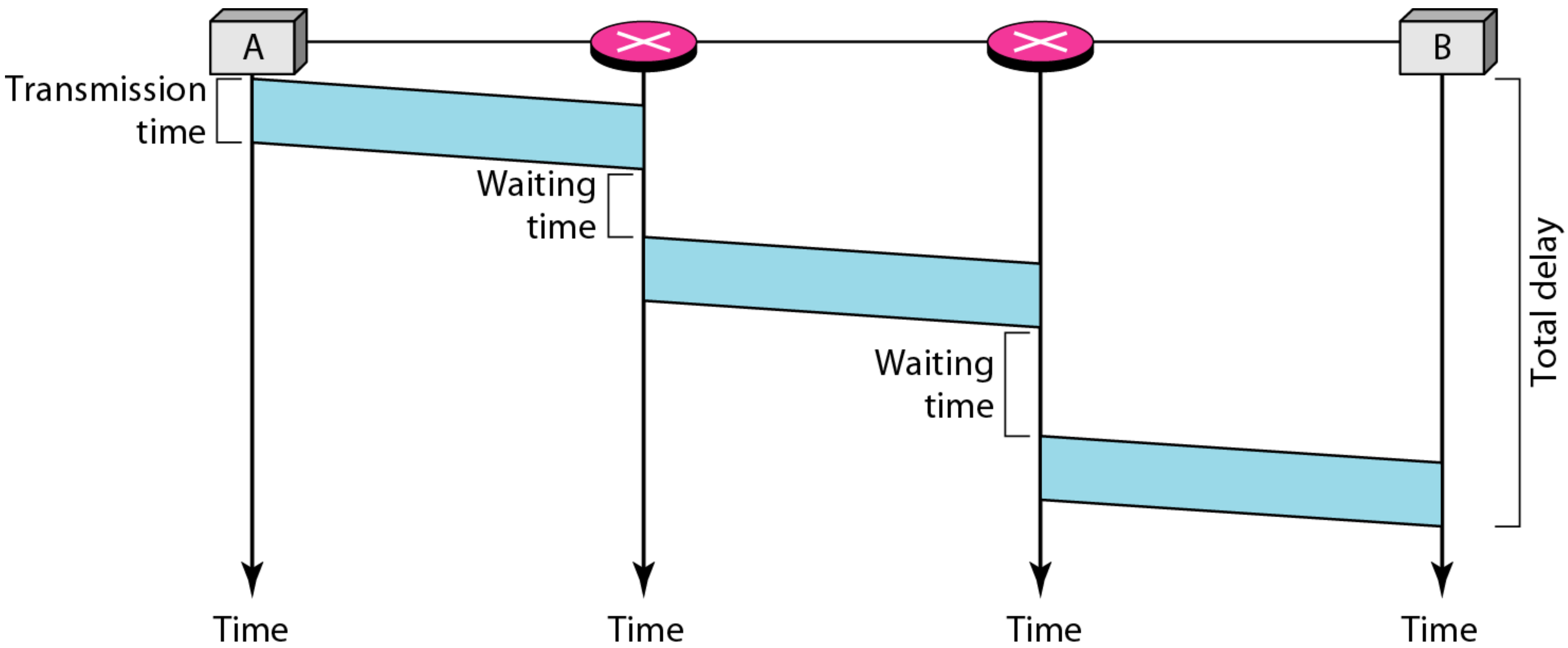
**Note**

**A switch in a datagram network uses a routing table that is based on the destination address.**

***Note***

**The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet.**

## Figure 8.9 *Delay in a datagram network*




**Total delay = 3 transmission times + 3 propagation times + 2 waiting times**

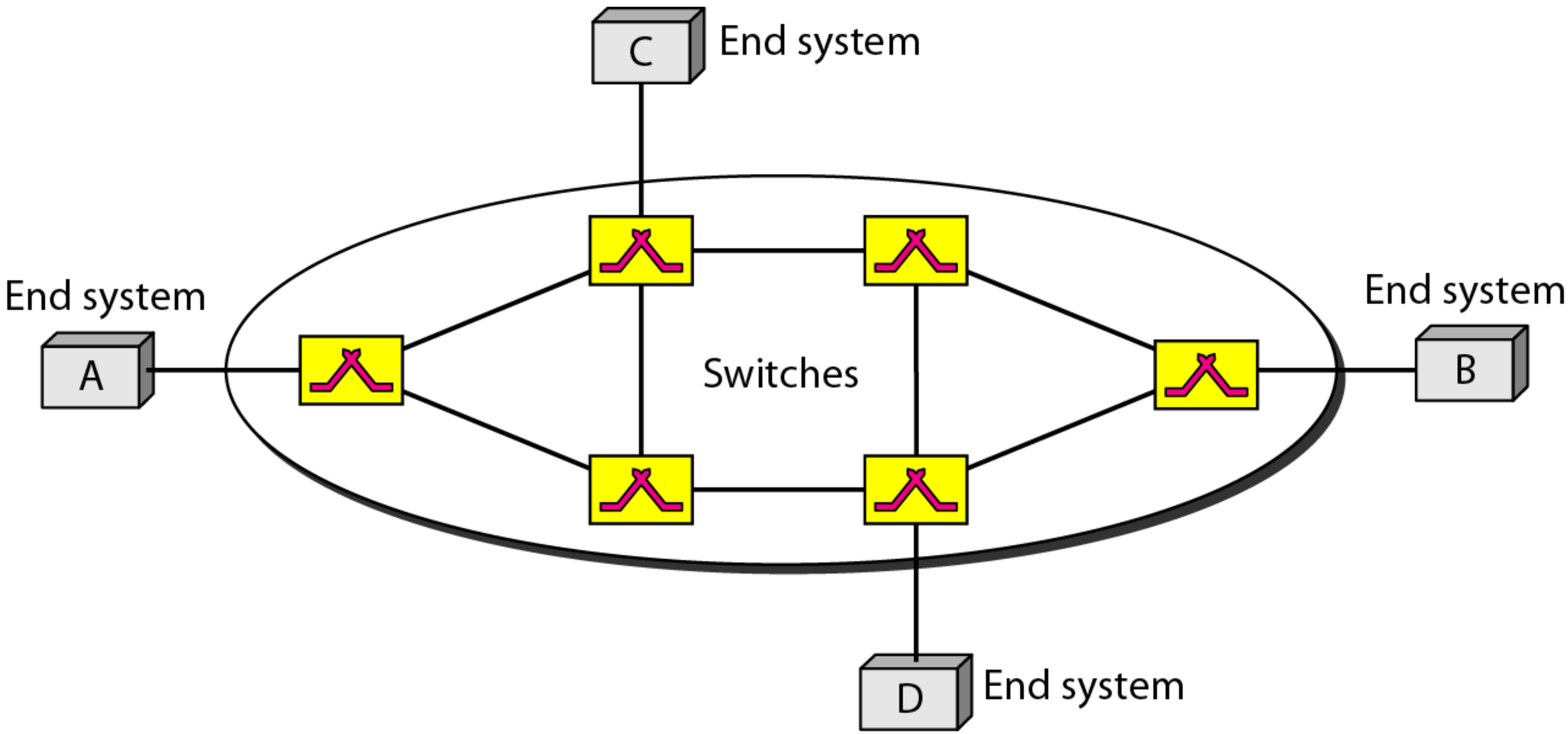
*Note*

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

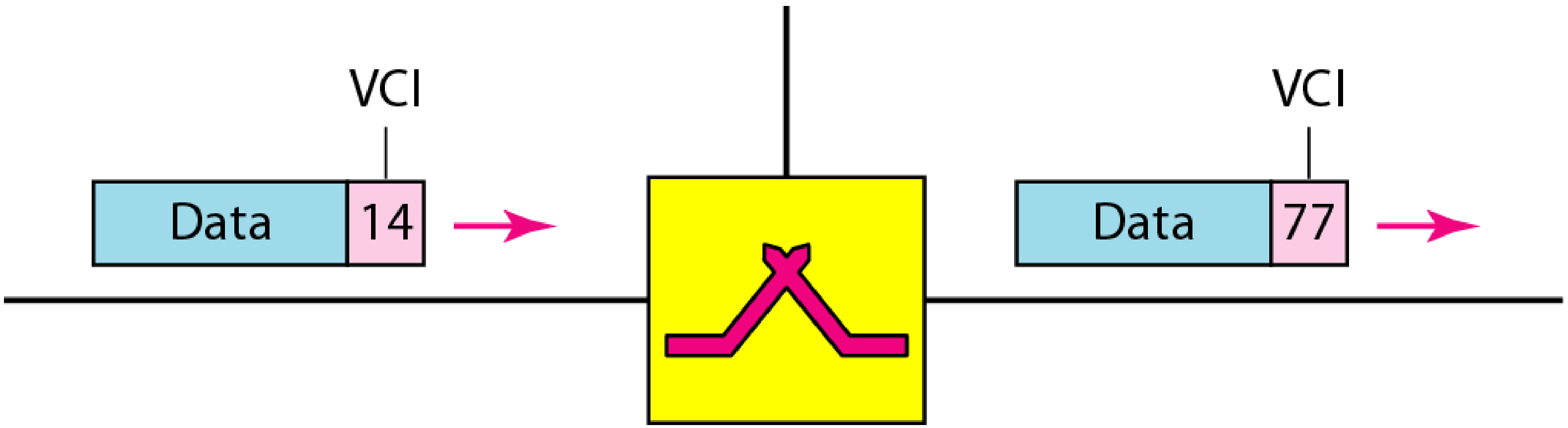
# 8-3 VIRTUAL-CIRCUIT NETWORKS

***A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.***

## **Figure 8.10** *Virtual-circuit network*

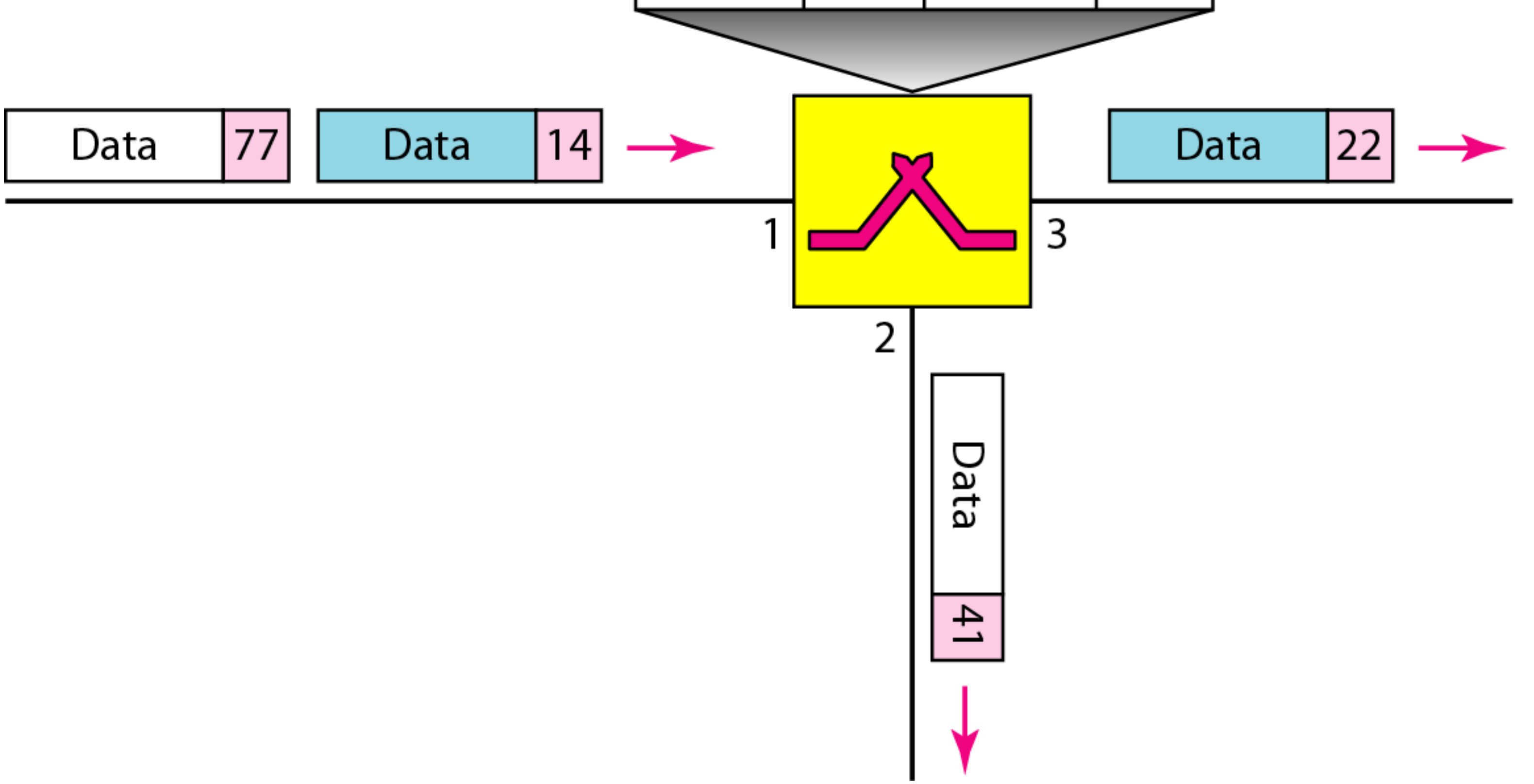


## Figure 8.11 *Virtual-circuit identifier*

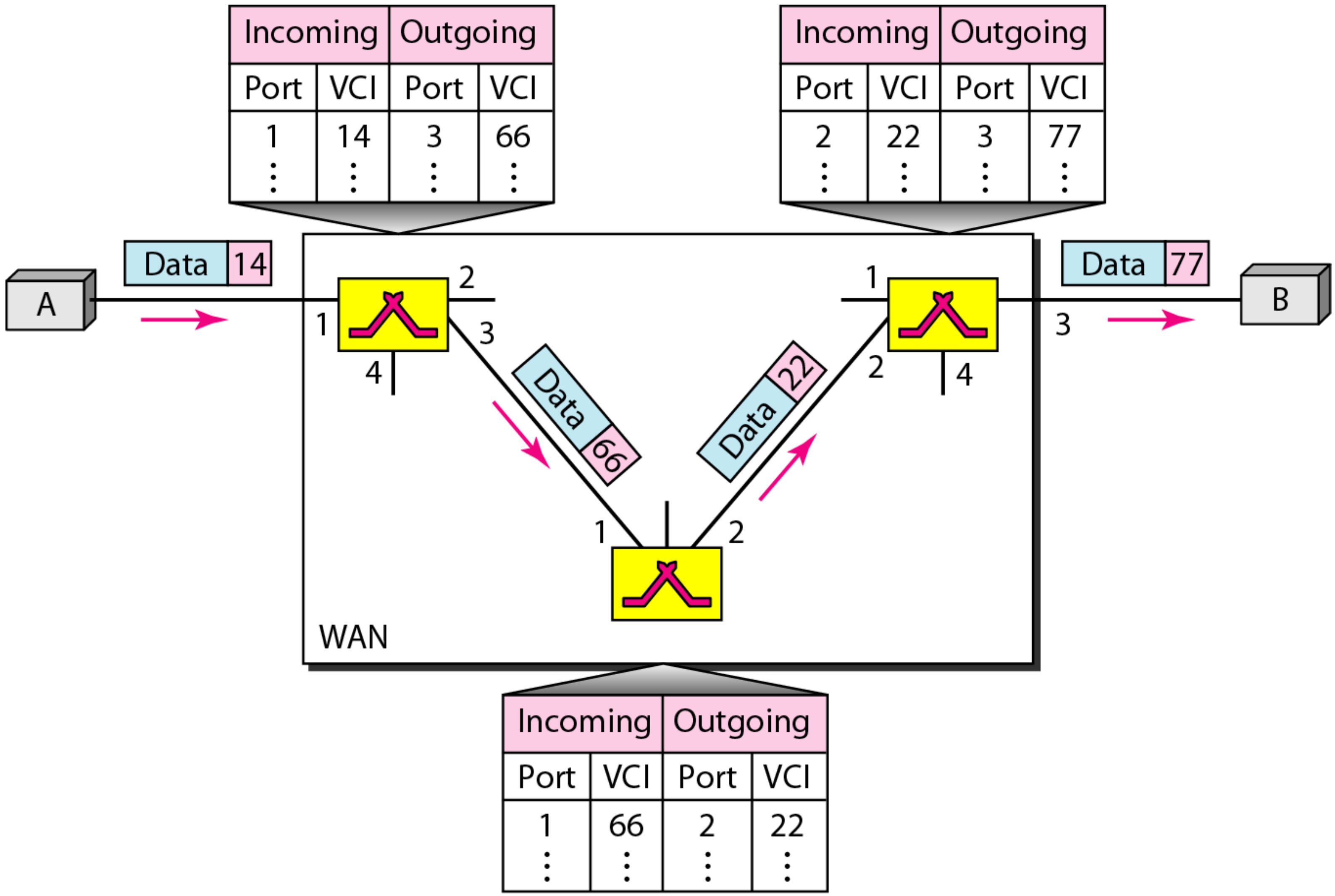


## **Figure 8.12** *Switch and tables in a virtual-circuit network*

| Incoming | | Outgoing | |
|---|---|---|---|
| Port | VCI | Port | VCI |
| 1 | 14 | 3 | 22 |
| 1 | 77 | 2 | 41 |



# Figure 8.13 *Source-to-destination data transfer in a virtual-circuit network*

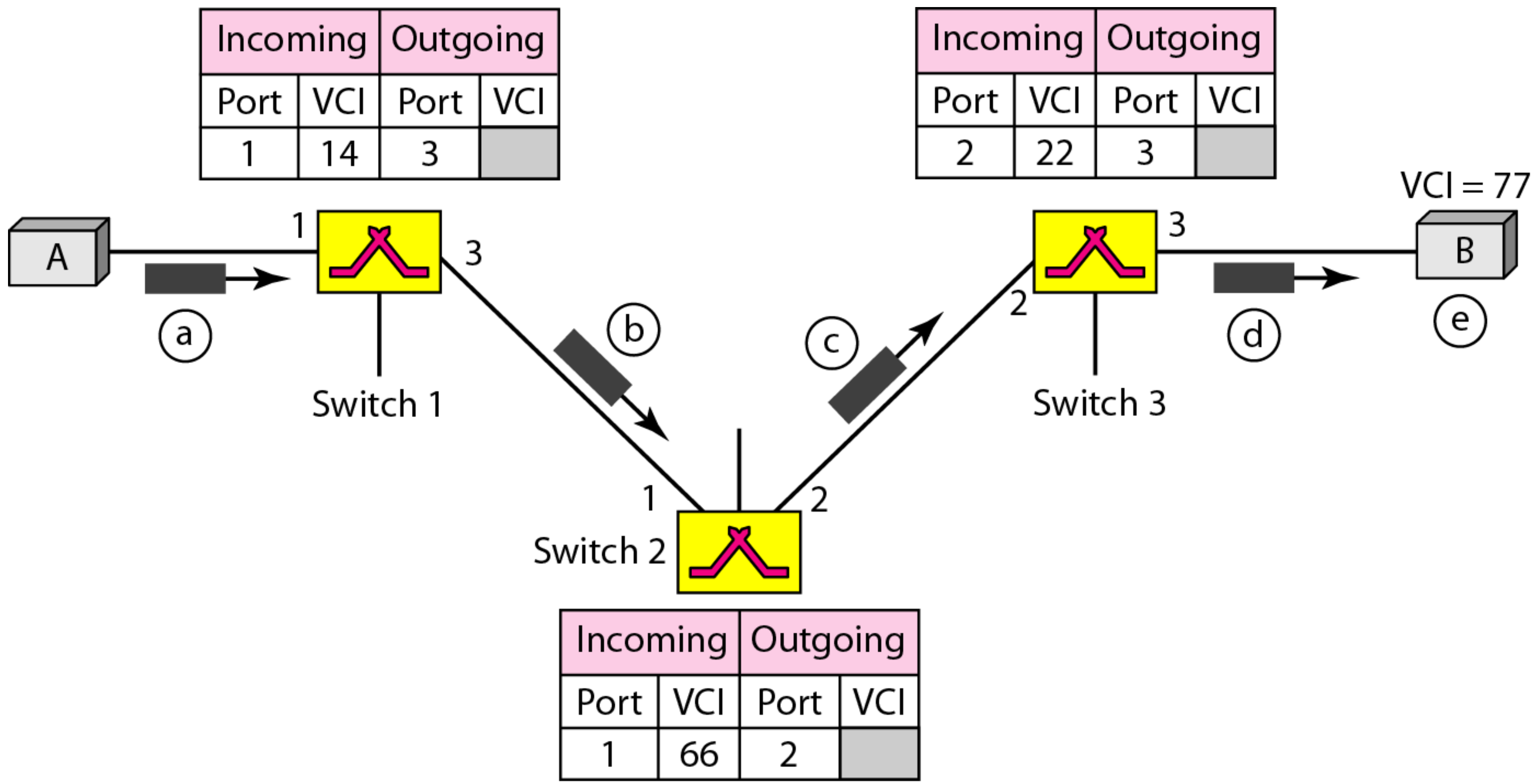



| Incoming | | Outgoing | |
|---|---|---|---|
| Port | VCI | Port | VCI |
| 1 | 14 | 3 | 66 |
| ⋮ | ⋮ | ⋮ | ⋮ |

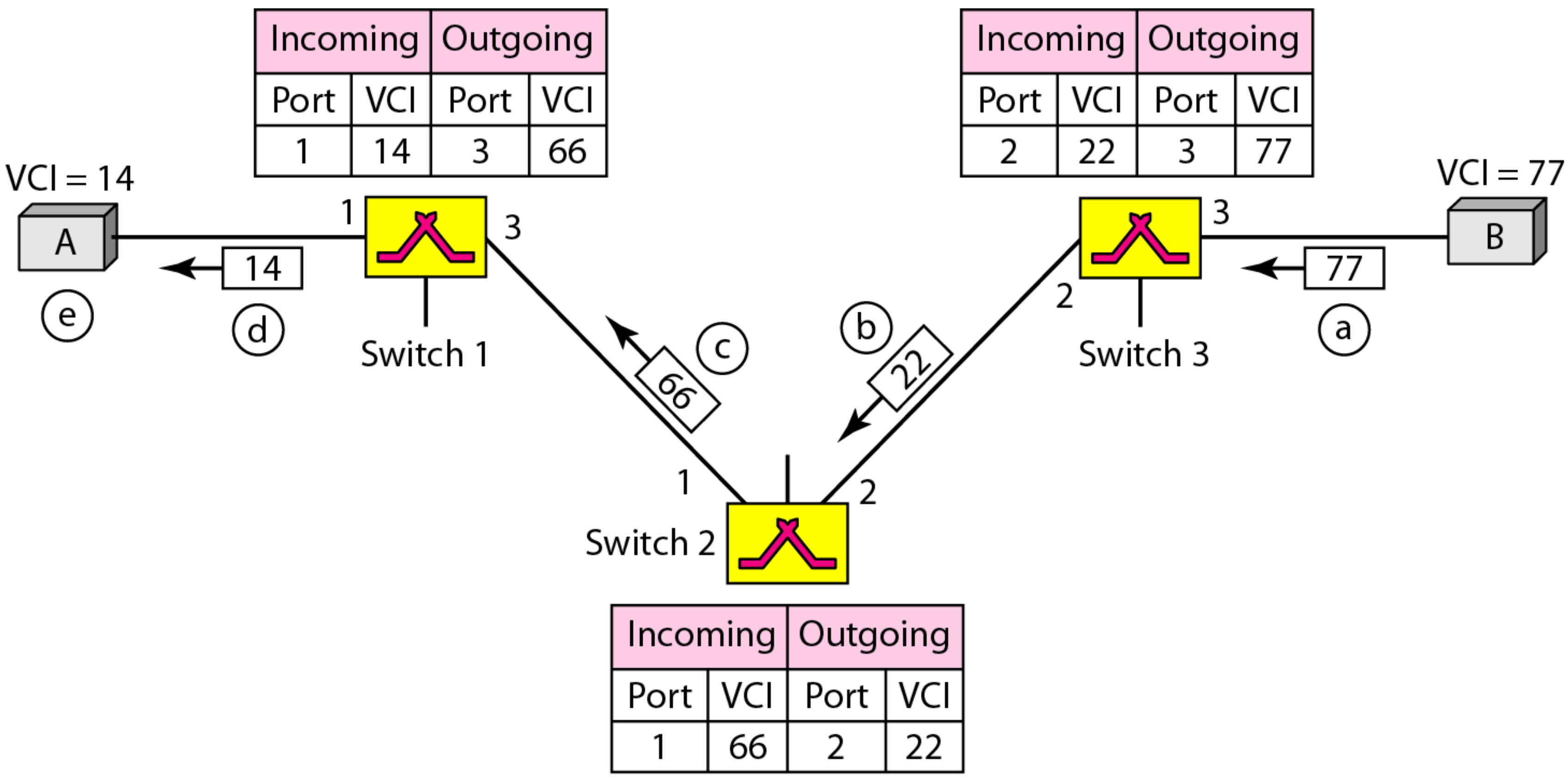
| Incoming | | Outgoing | |
|---|---|---|---|
| Port | VCI | Port | VCI |
| 2 | 22 | 3 | 77 |
| ⋮ | ⋮ | ⋮ | ⋮ |

| Incoming | | Outgoing | |
|---|---|---|---|
| Port | VCI | Port | VCI |
| 1 | 66 | 2 | 22 |
| ⋮ | ⋮ | ⋮ | ⋮ |

## Figure 8.14 *Setup request in a virtual-circuit network*



# **Figure 8.15** *Setup acknowledgment in a virtual-circuit network*



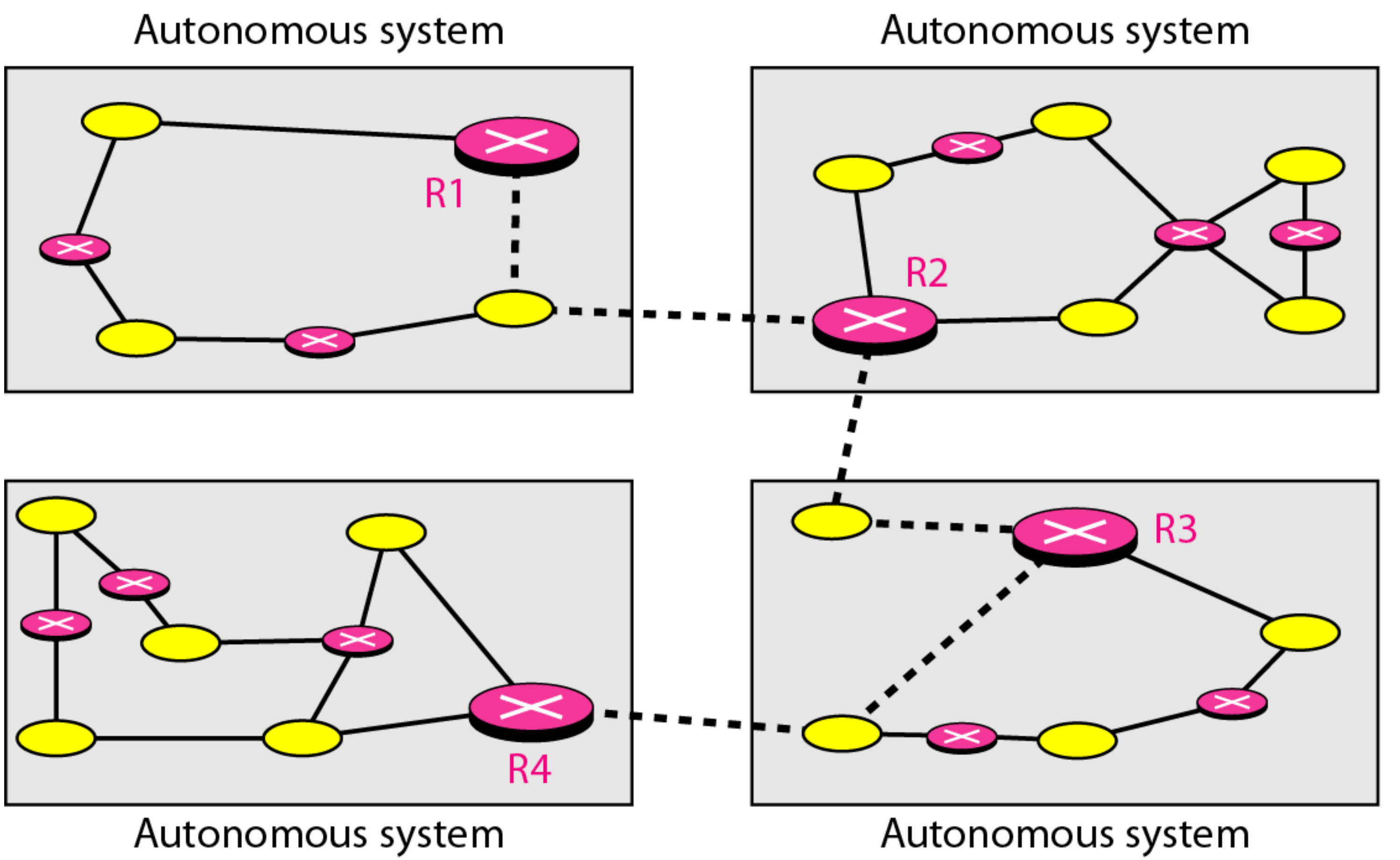
***Note***

**In virtual-circuit switching, all packets belonging to the same source and destination travel the same path; but the packets may arrive at the destination with different delays if resource allocation is on demand.**

# 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.***

## Figure 22.12 *Autonomous systems*




**Autonomous system: a group of networks and routers under the authority of a single administration.**

## **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

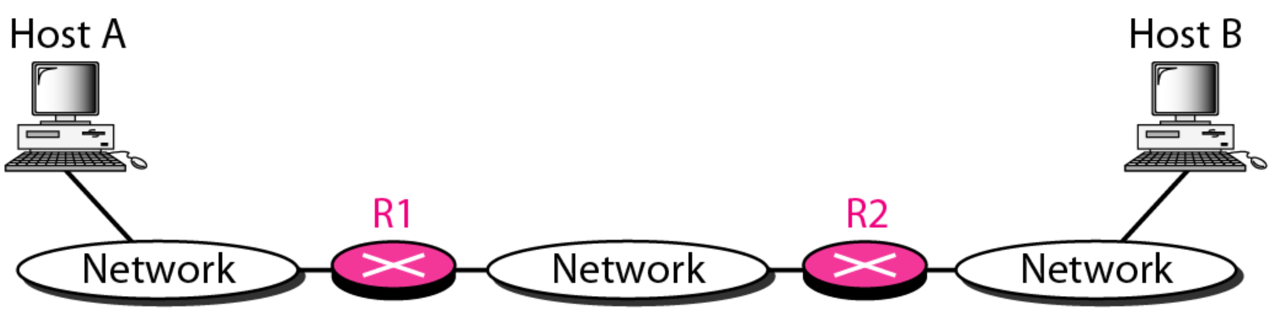
| Destination | Next hop |
|---|---|
| Host B | R1 |

Routing table for R1

| Destination | Next hop |
|---|---|
| Host B | R2 |

Routing table for R2

| Destination | Next hop |
|---|---|
| Host B | --- |



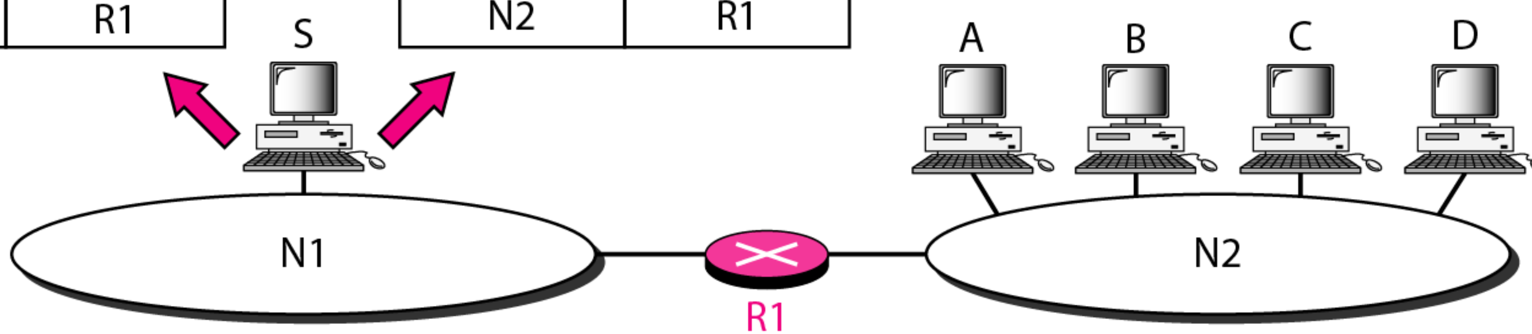
## **Figure 22.3** *Host-specific versus network-specific method*

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

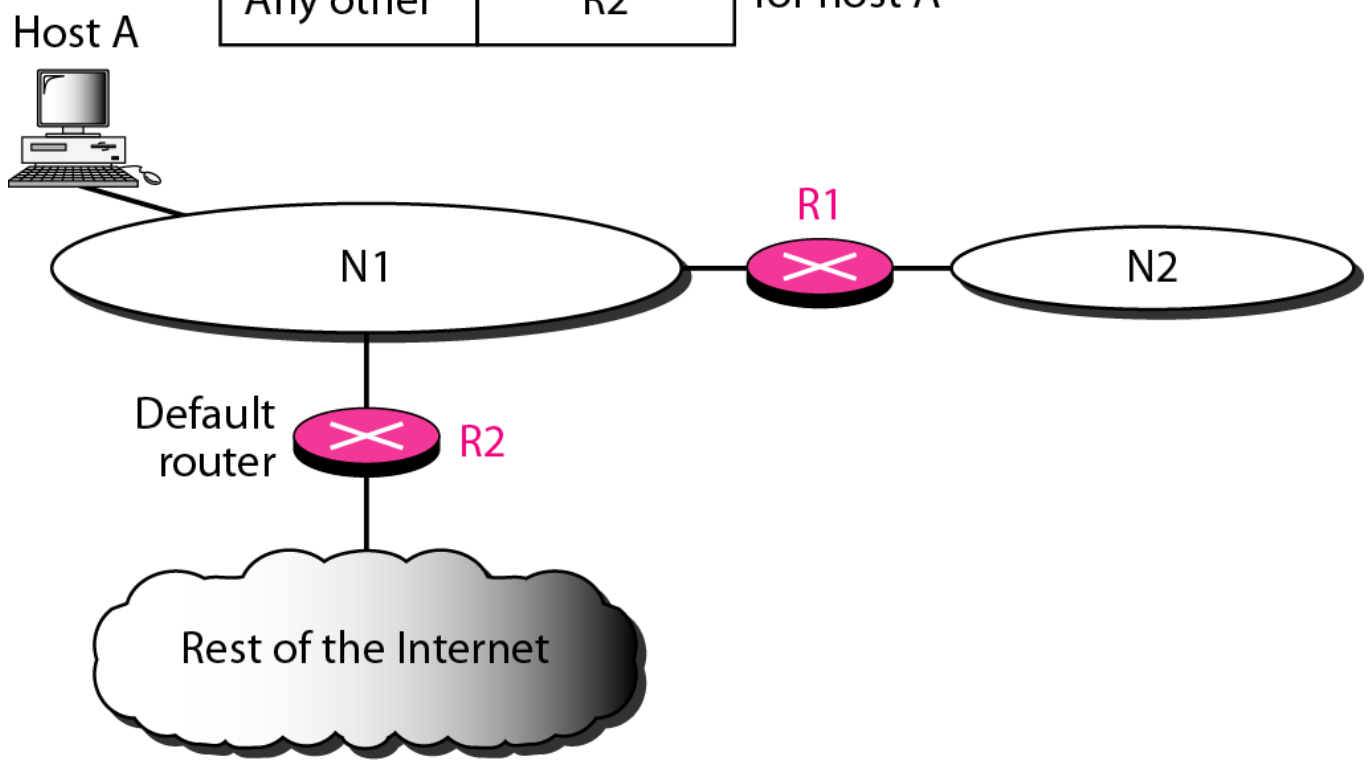
| Destination | Next hop |
|---|---|
| N2 | R1 |



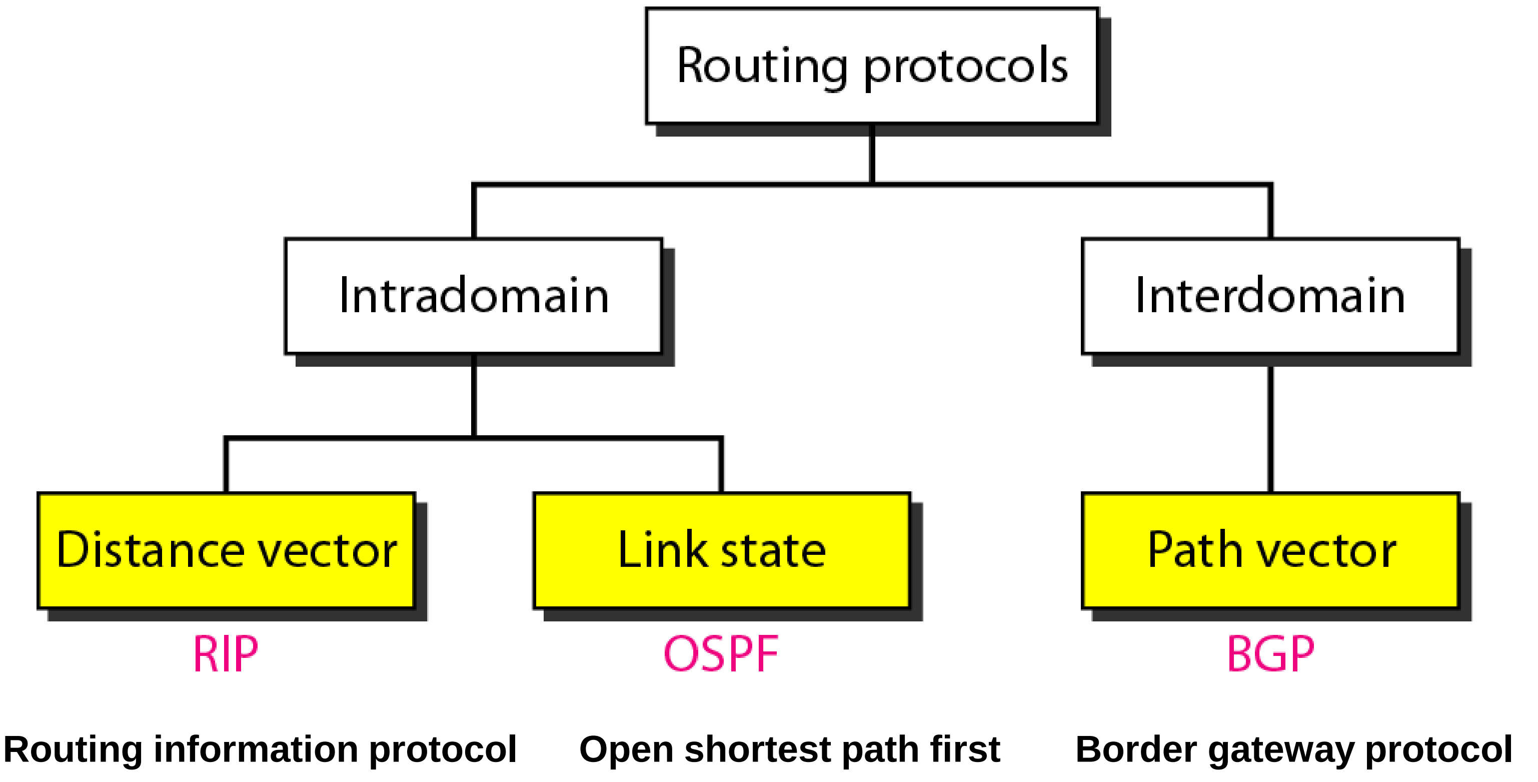
# **Figure 22.4** *Default method*

| Destination | Next hop |
|---|---|
| N2 | R1 |
| Any other | R2 |

Routing table for host A

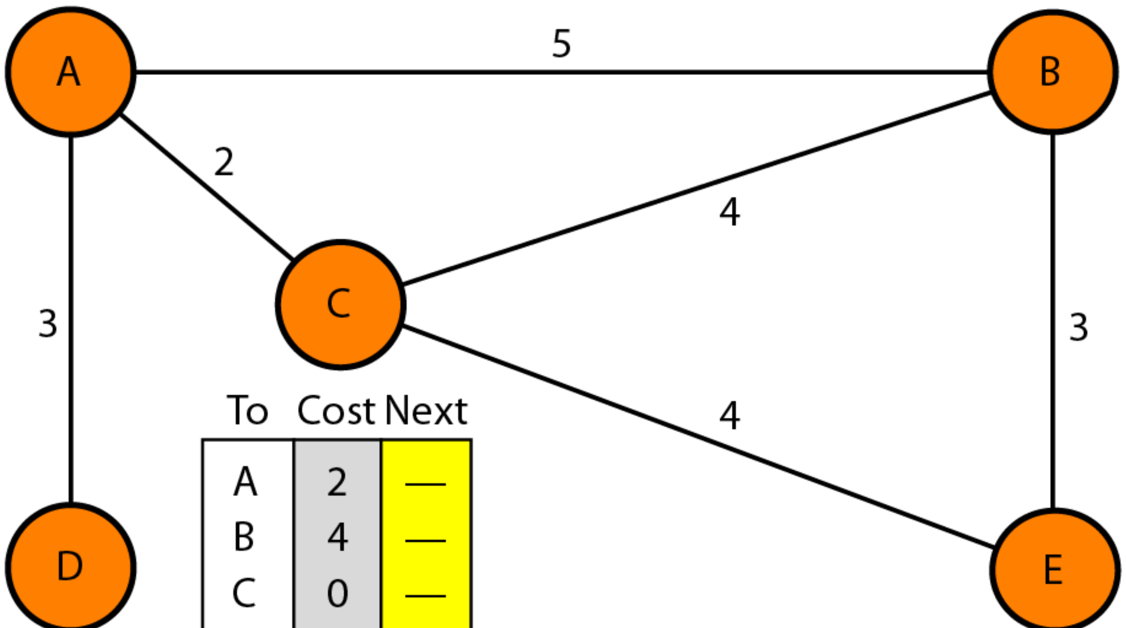


## **Figure 22.13** *Popular routing protocols*



**In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. Each node maintains a vector (table) of minimum distances to every node. The table 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 | 2 | — |
| B | 4 | — |
| C | 0 | — |
| D | 5 | A |
| E | 4 | — |

C's table

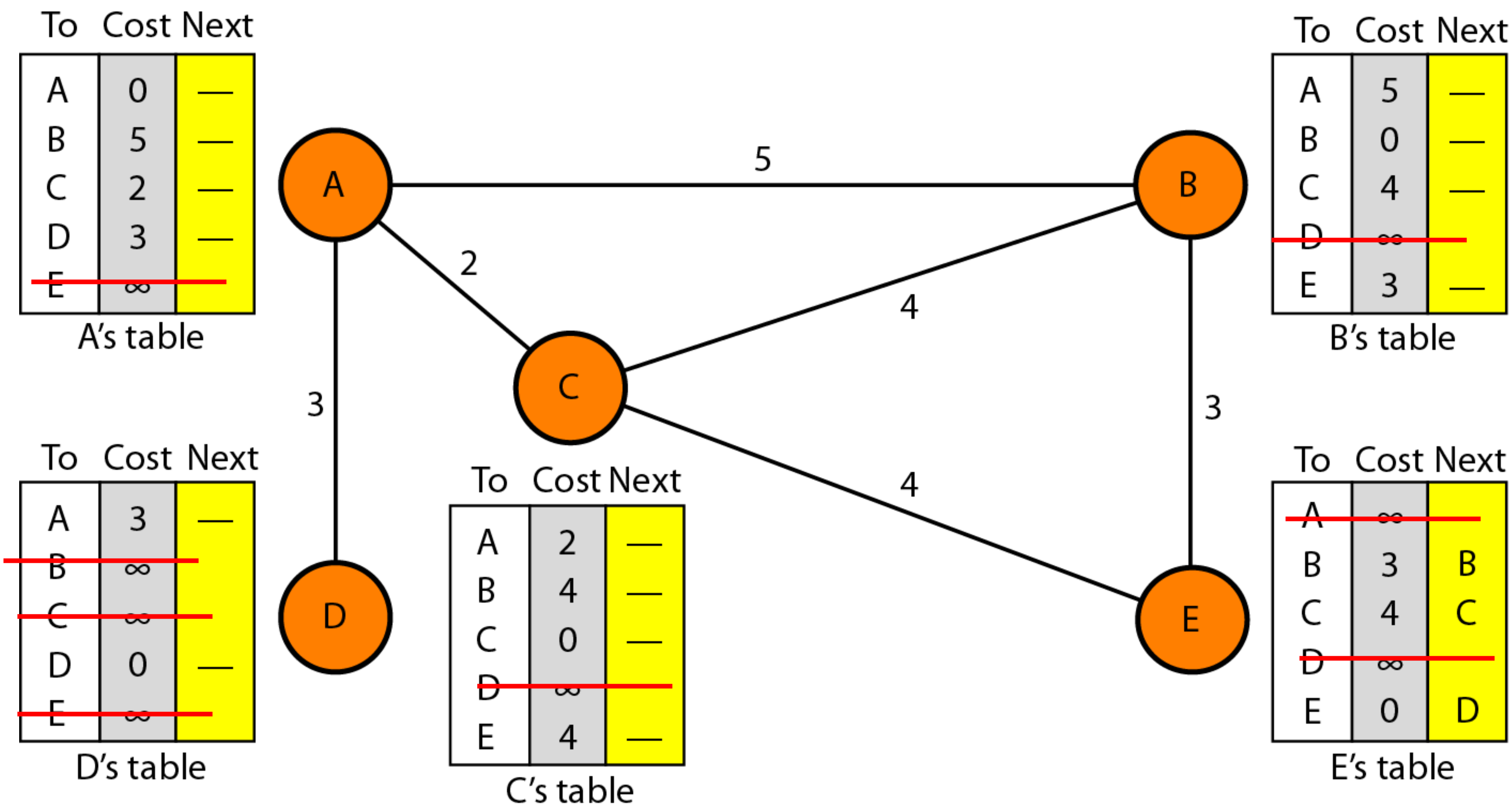
| To | Cost | Next |
|---|---|---|
| A | 3 | — |
| B | 8 | A |
| C | 5 | A |
| D | 0 | — |
| E | 9 | A |

D's table

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

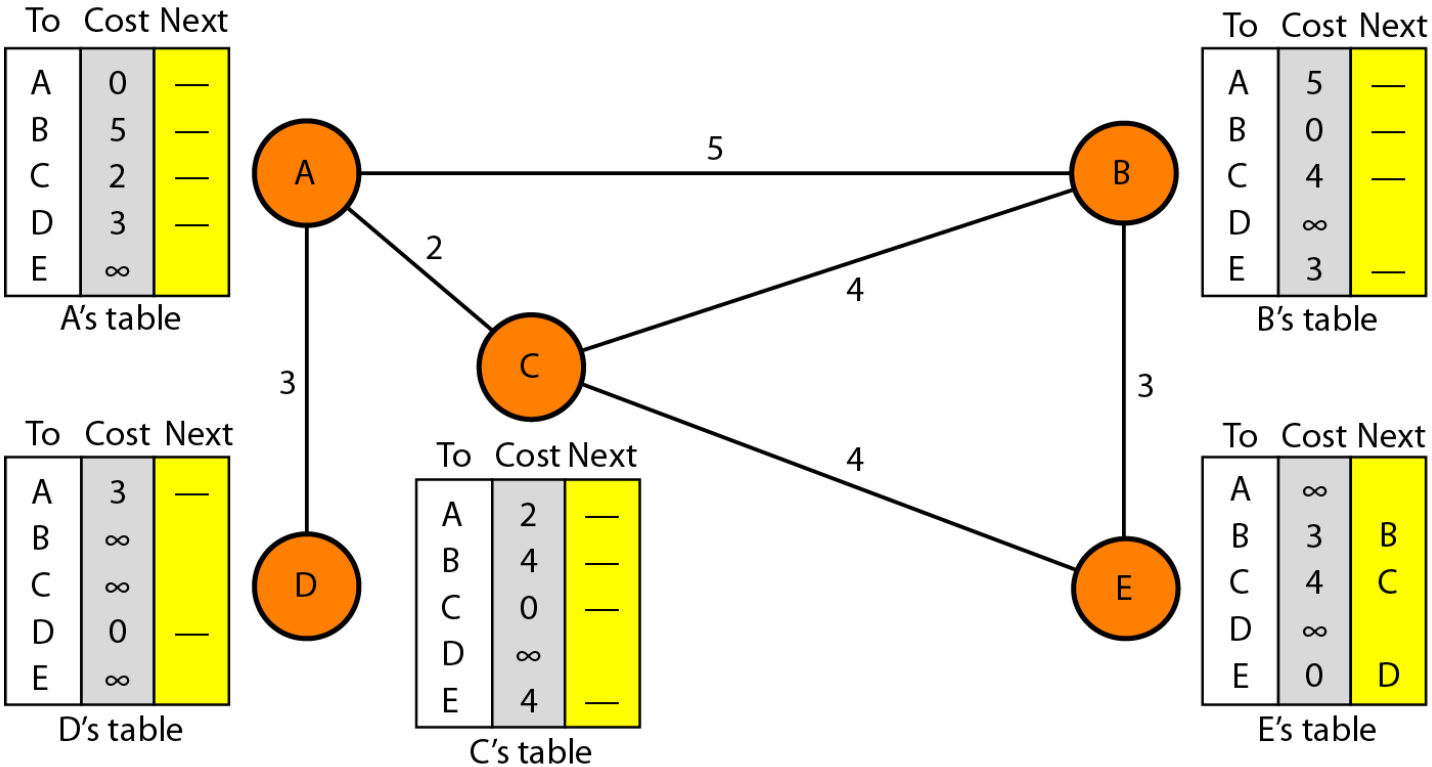
E's table

## **Figure 22.15** *Initialization of tables in distance vector routing*



**Note**

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



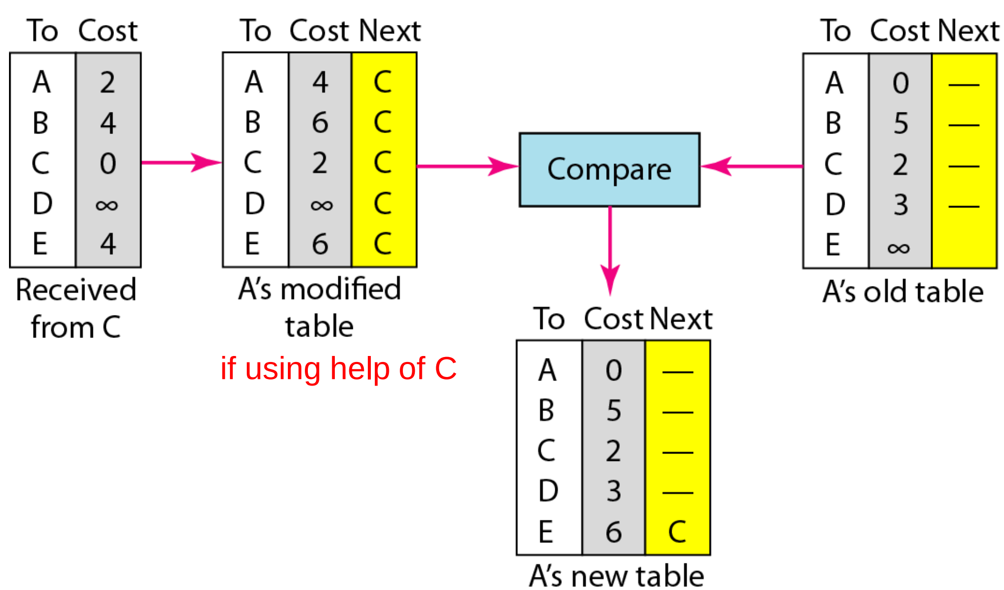
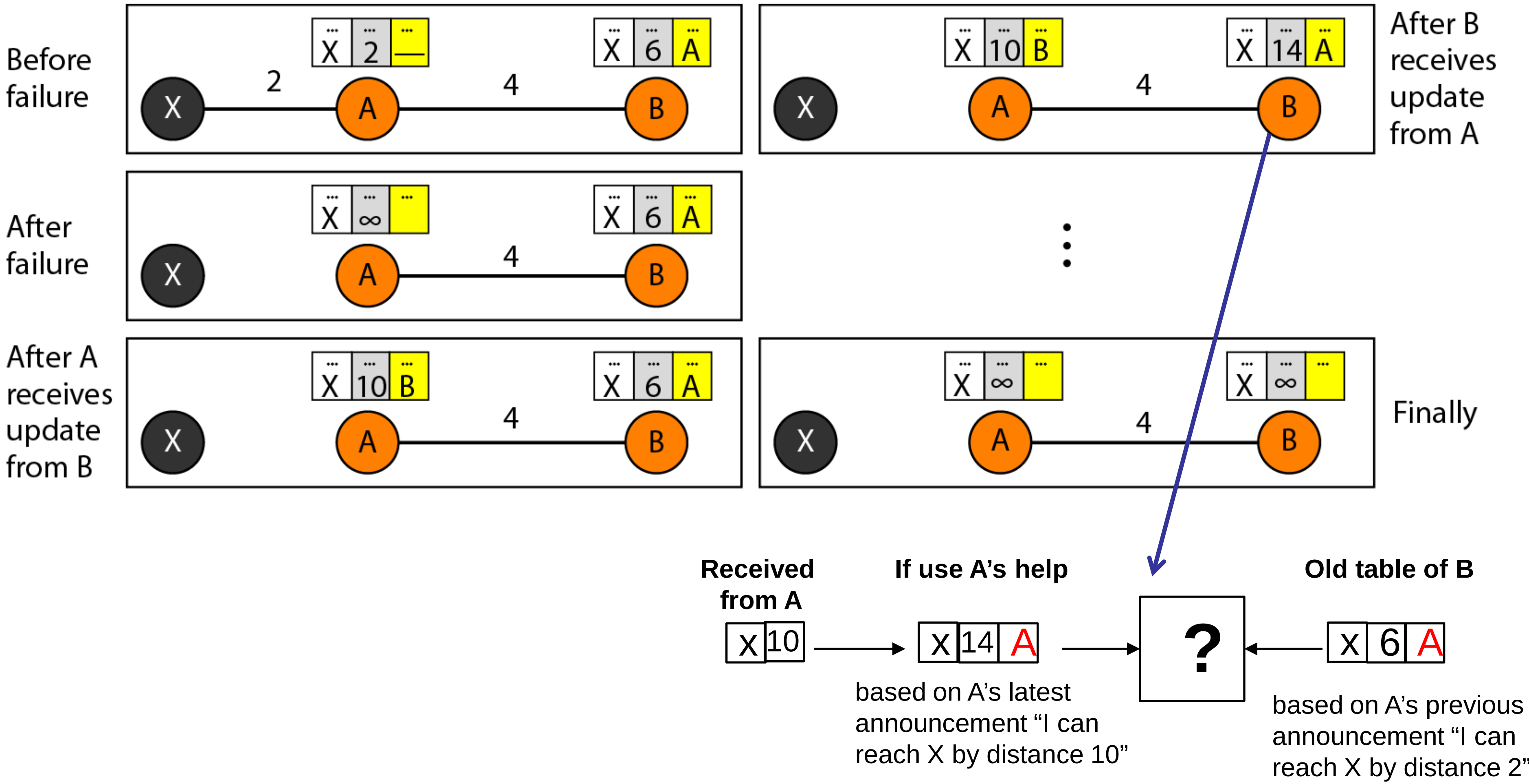




**Figure 22.16** ***Updating in distance vector routing***

# **Figure 22.17** *Two-node instability*

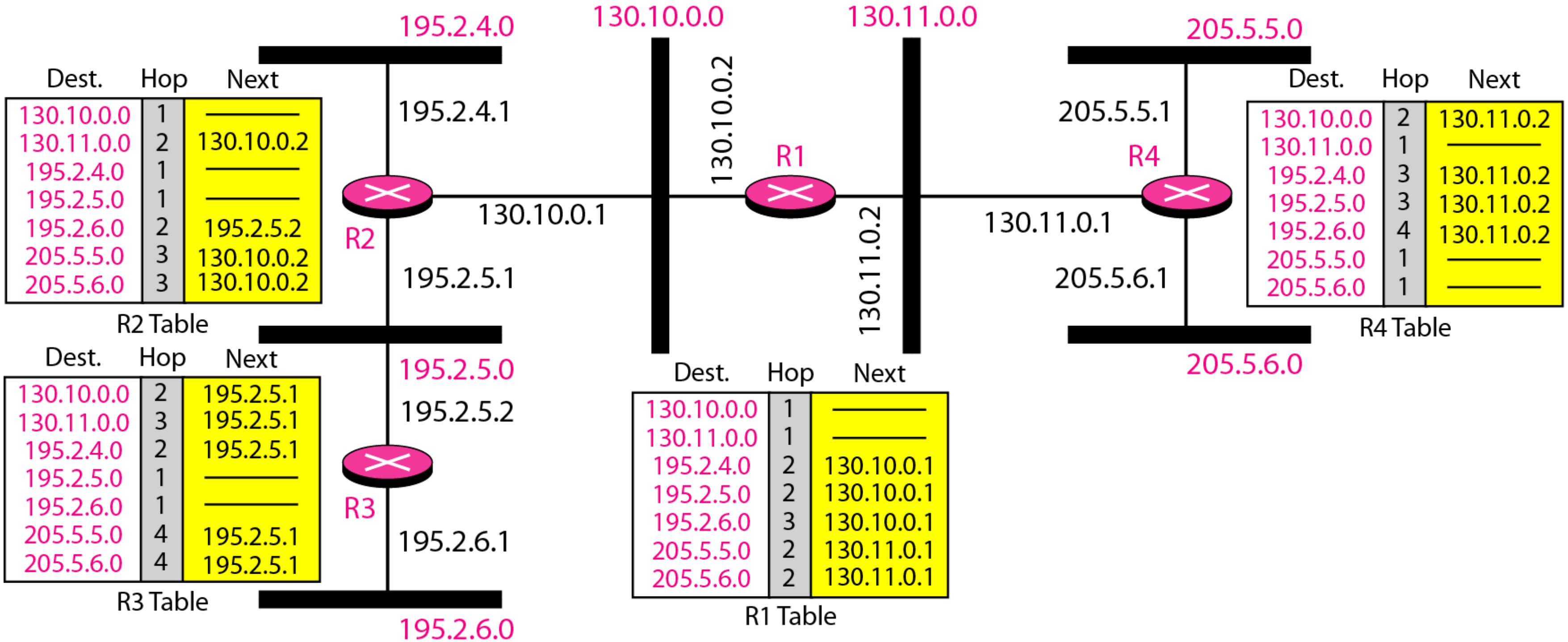



**One solution: redefine infinity to a smaller number, e.g., 16 (cannot used in large networks)**

# Routing Information Protocol (RIP)

1. The distance is defined as the number of links to reach the destination. This metric is called a hop count
2. Infinity is defined as 16 (which means any route cannot have more than 15 hops)
3. The next-node column defines the address of the next router (Q: physical or logic address?)

## **Figure 22.19** *Example of a domain using RIP*




R2 Table

| 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 |

R3 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 |

R1 Table

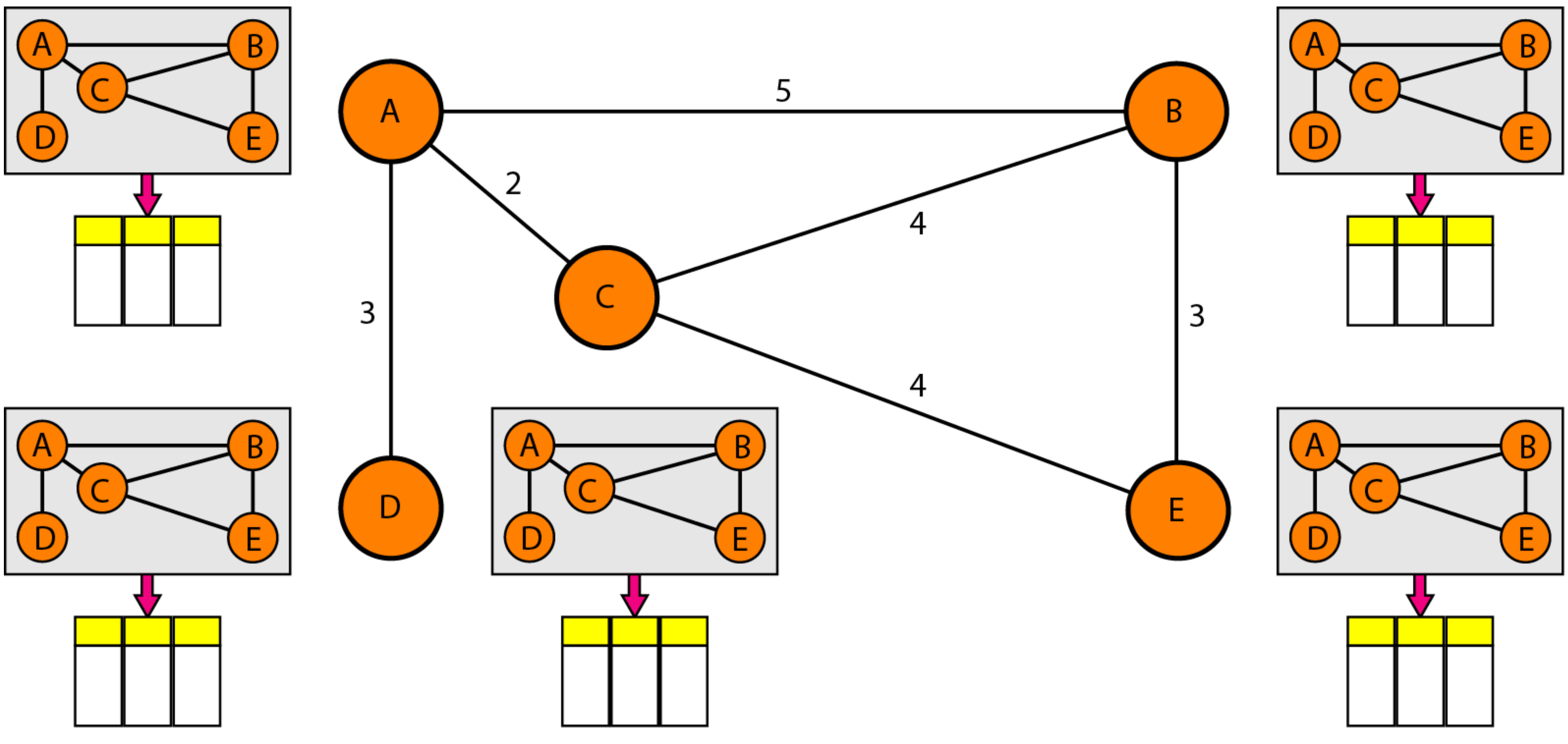
| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 1 | ——— |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 2 | 130.10.0.1 |
| 195.2.5.0 | 2 | 130.10.0.1 |
| 195.2.6.0 | 3 | 130.10.0.1 |
| 205.5.5.0 | 2 | 130.11.0.1 |
| 205.5.6.0 | 2 | 130.11.0.1 |

R4 Table

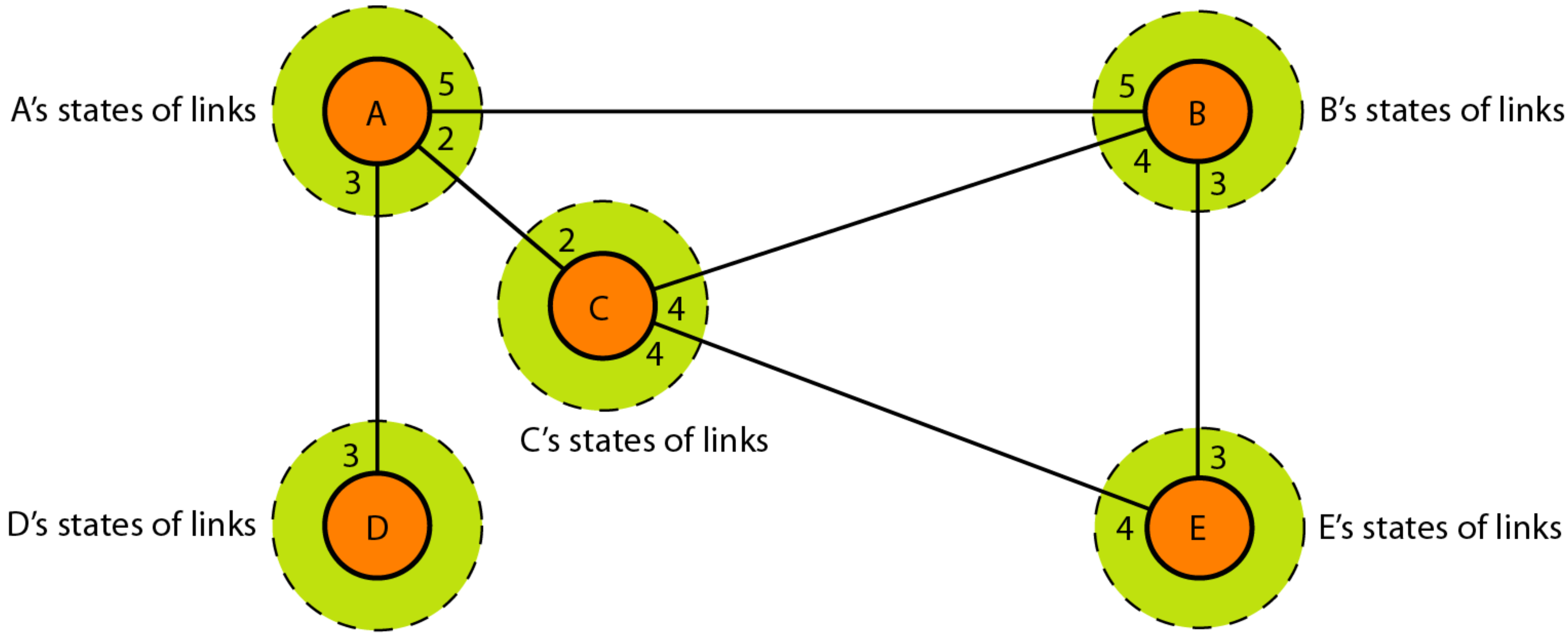
| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 2 | 130.11.0.2 |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 3 | 130.11.0.2 |
| 195.2.5.0 | 3 | 130.11.0.2 |
| 195.2.6.0 | 4 | 130.11.0.2 |
| 205.5.5.0 | 1 | ——— |
| 205.5.6.0 | 1 | ——— |

**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 (including 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.**

## **Figure 22.20** *Concept of link state routing*



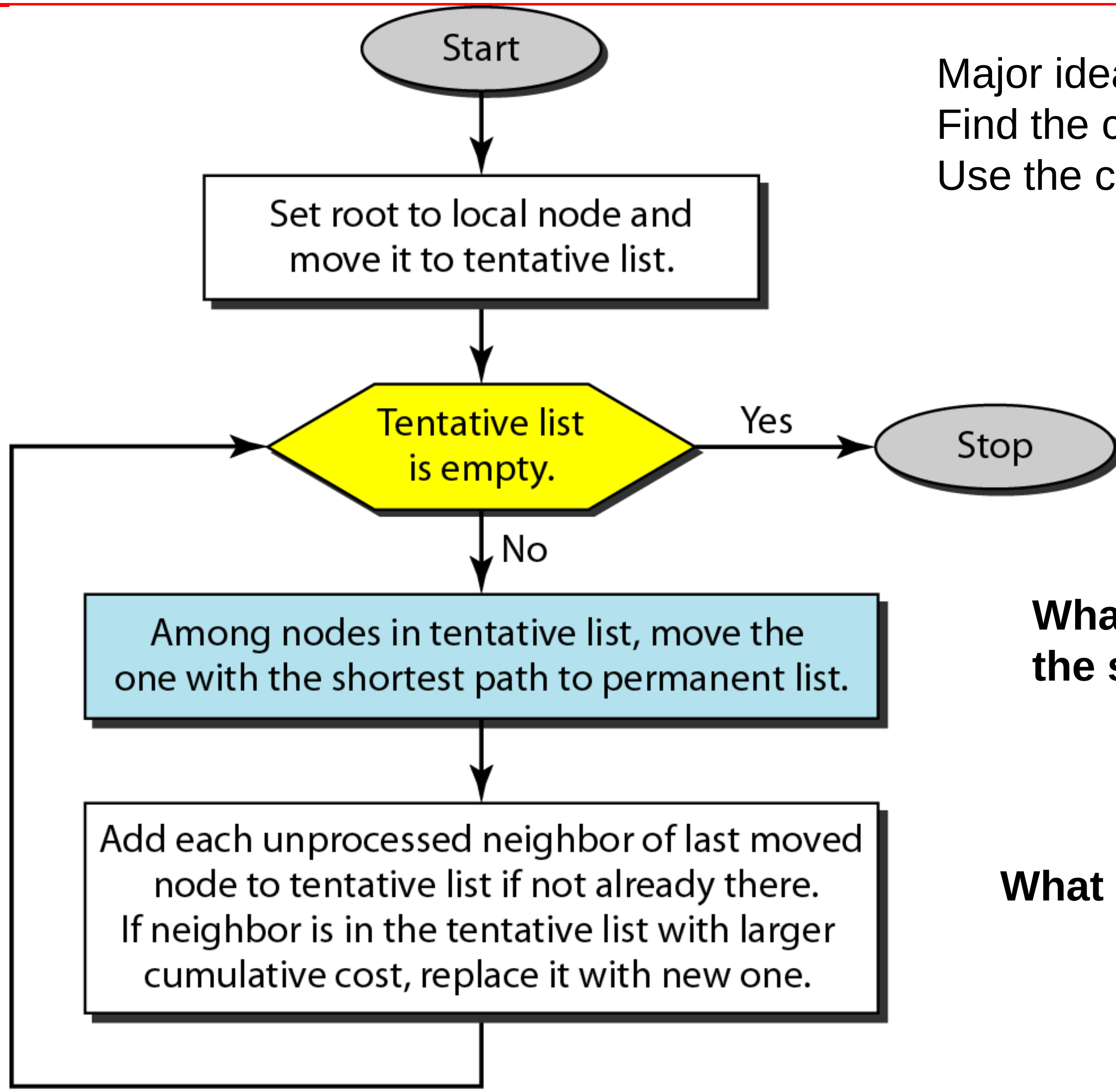
## **Figure 22.21** *Link state knowledge*



## *Building routing tables in link state routing*

1. Creation of the states of the links by each node, called the link state packet (LSP). An LSP includes
   i) the node identity
   ii) the list of links
   iii) a sequence number (time stamp of the LSP)
2. Each node disseminates its LSP to all other nodes, not only to its neighbors, called flooding.
   i) the creating node sends a copy of its LSP out of each interface
   ii) a node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the arrived LSP; otherwise, it will update its LSP record and sends a copy of the LSP out of each interface except the one from which the LSP arrived.
3. Each node forms a shortest path tree
4. Each node calculates a routing table based on the shortest path tree

## **Figure 22.22** *Dijkstra algorithm*




Major idea:
Find the closest node;
Use the closest node to reach other nodes.

**What if more than one nodes have the same shortest path?**

**What if the two costs are the same?**

# **Figure 22.23** *Example of formation of shortest path tree*

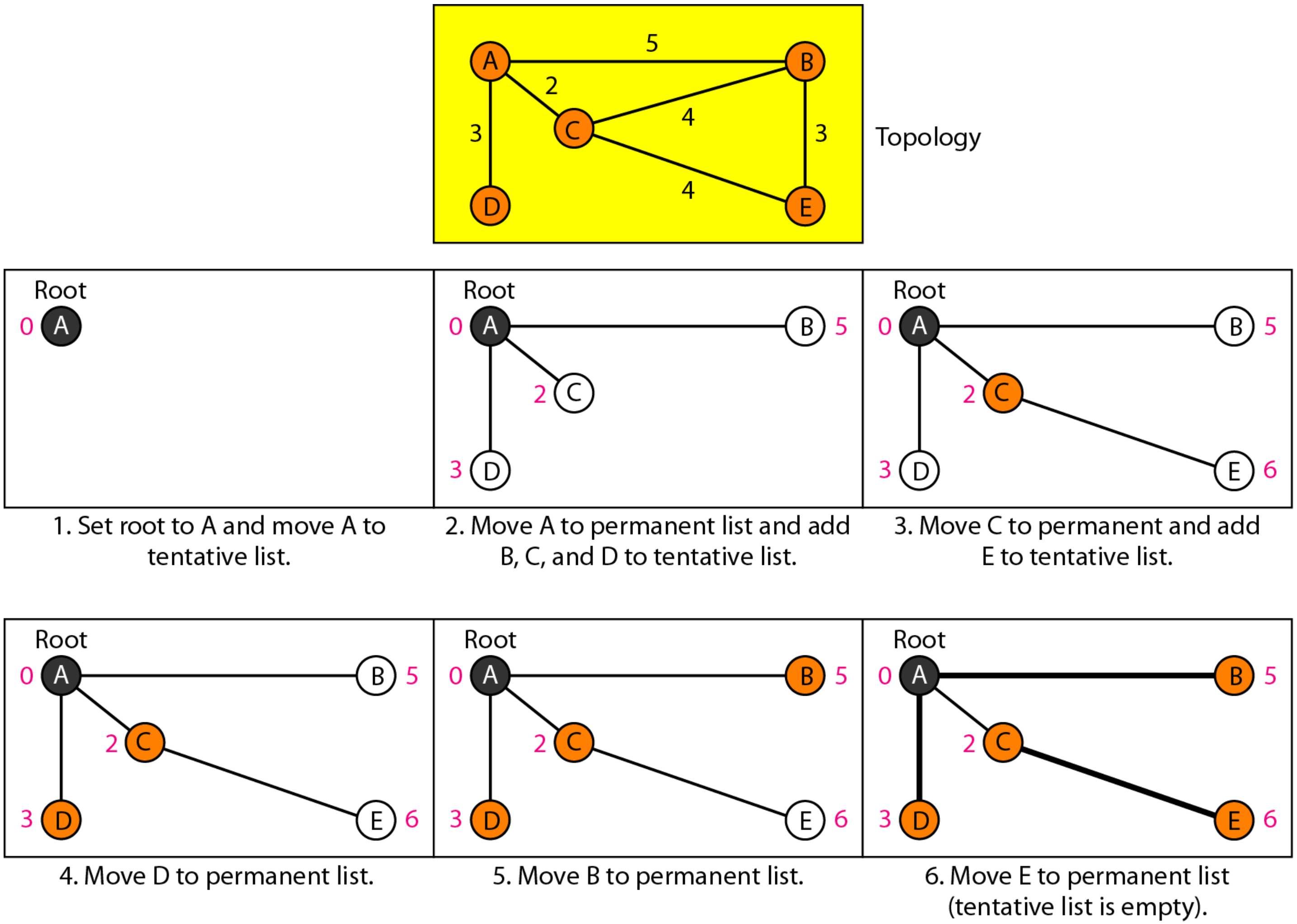
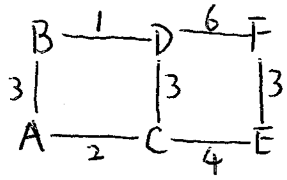


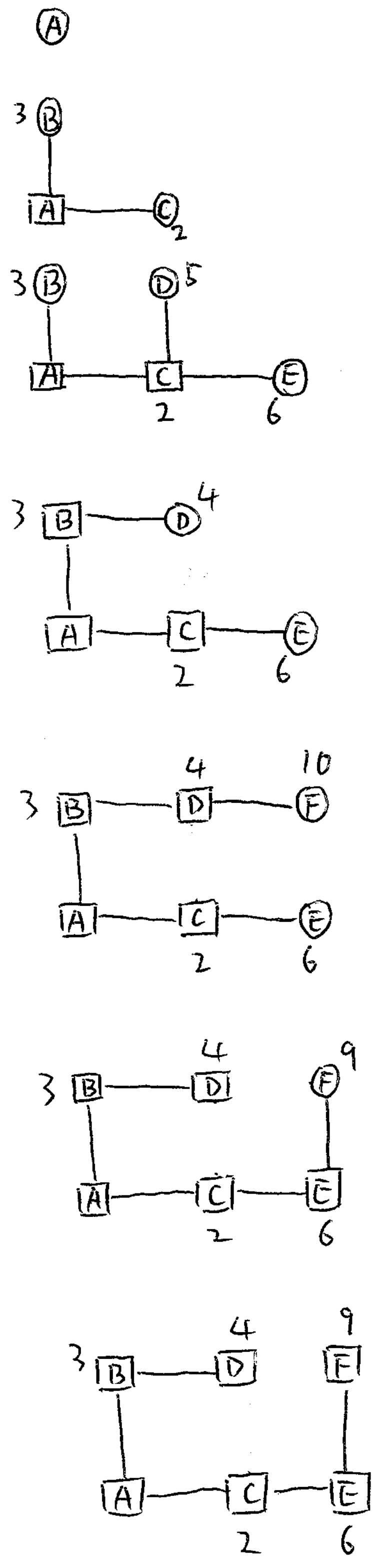
**Table 22.2** ***Routing table for node A***

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



O: tentative

□: permanent



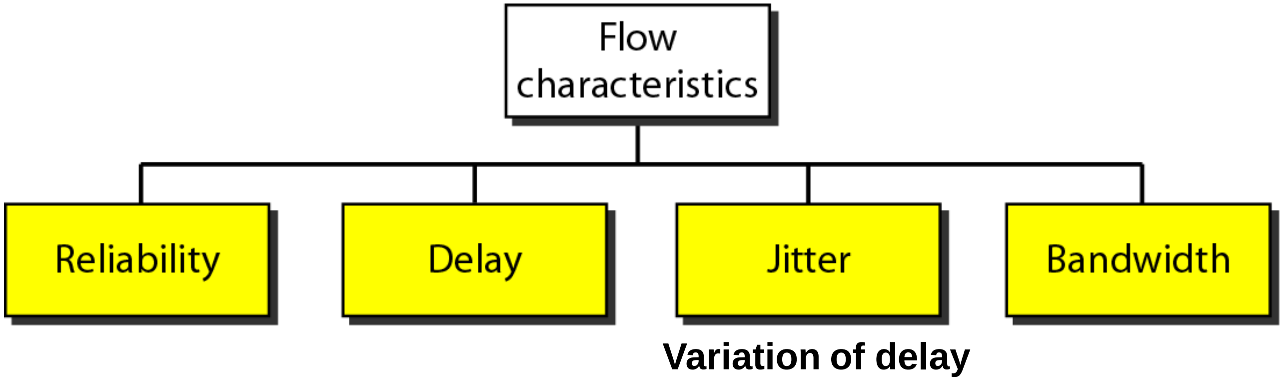
Routing table at A:

| node | cost | Next node |
|---|---|---|
| A | 0 | — |
| B | 3 | — |
| C | 2 | — |
| D | 4 | B |
| E | 6 | C |
| F | 9 | C |

# 24-5 QUALITY OF SERVICE

***Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.***

## Figure 24.15 *Flow characteristics*

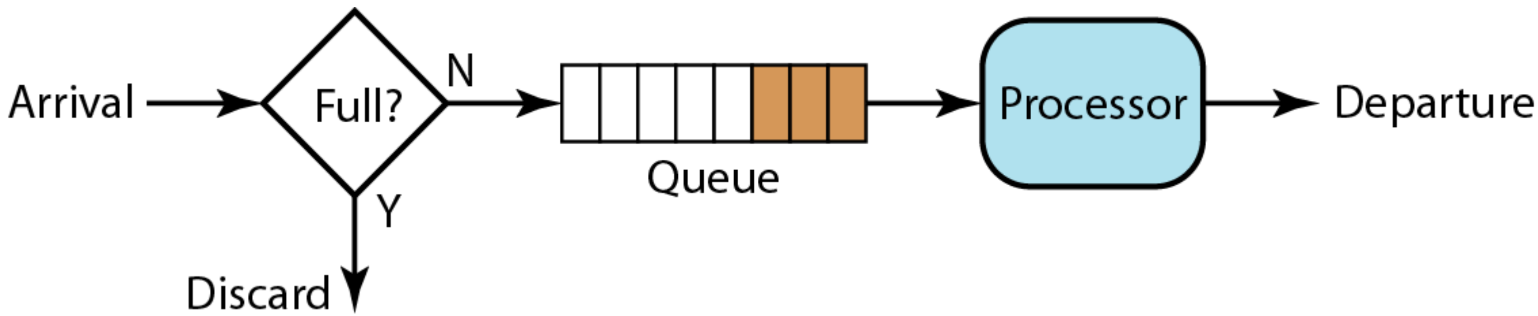



**Non-realtime applications (email, file transfer, Internet access): need reliable transmissions, but delay tolerant.**
**Realtime applications (telephony or audio/video conferencing): need timely delivery. Can tolerate a certain level of packet losses.**
**Realtime applications: jitter is more annoying than delay.**
**Assume 4 packets departing at times 0, 1, 2, and 3. If they arrive at times 20, 21, 22, and 23, then delay is 20 for each packet. Jitter is zero. It is acceptable. If they arrive at times 21, 23, 21, and 28. Delay is 21, 22, 19, and 25 for the four packets. It is unacceptable.**

# 24-6 TECHNIQUES TO IMPROVE QoS

***In Section 24.5 we tried to define QoS in terms of its characteristics. In this section, we discuss some techniques that can be used to improve the quality of service. We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.***

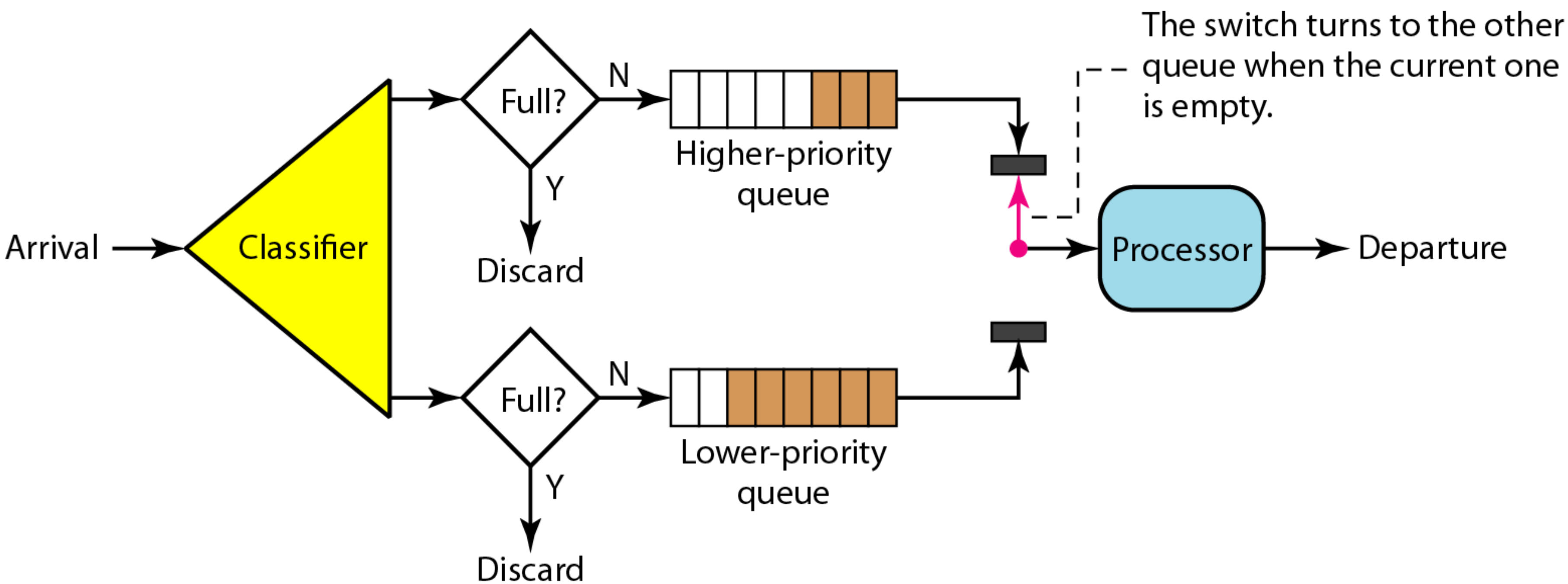
## Figure 24.16 *FIFO queue*



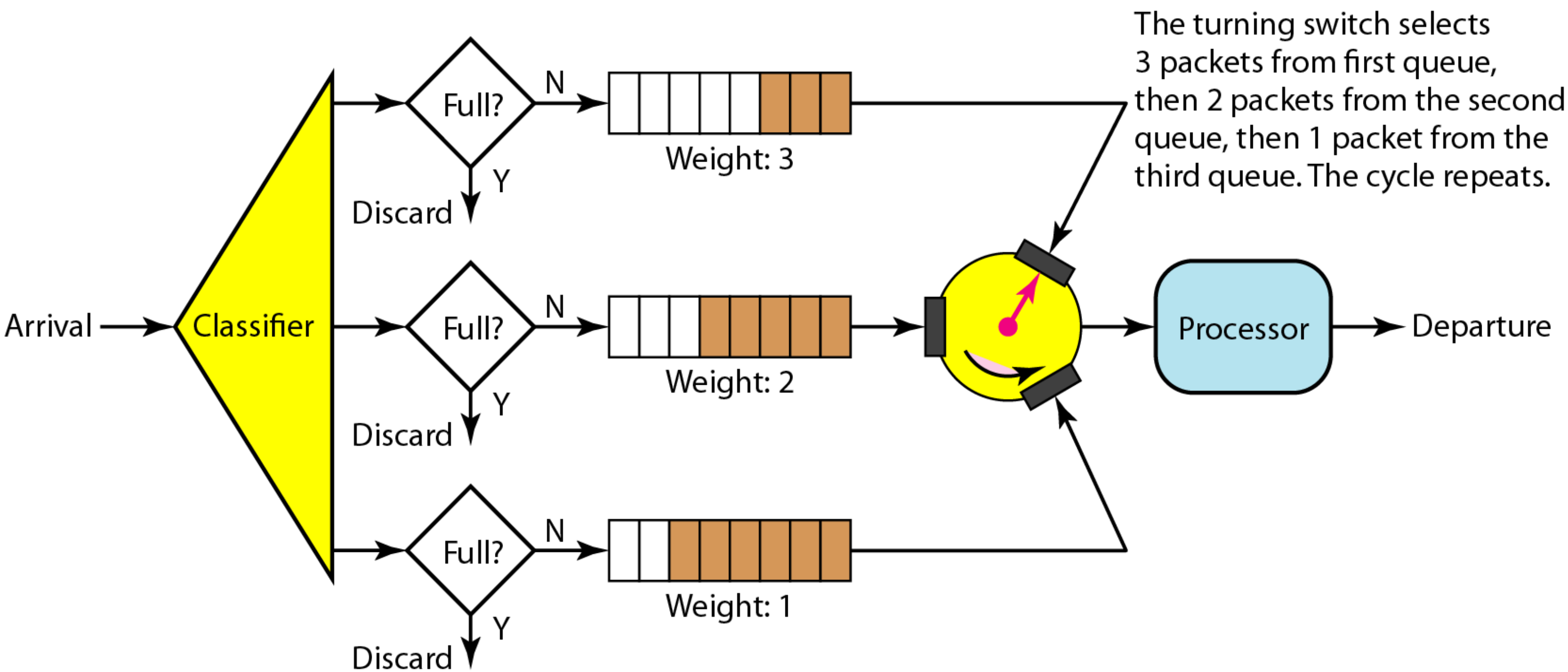

**Packet from different flows arrive at a router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner.**

**Packets wait in a buffer (queue) until being served.**
**If average arrival rate is higher than the average processing rate, the queue will fill up, and new packets will be discarded.**
**Can smooth bursty traffic.**

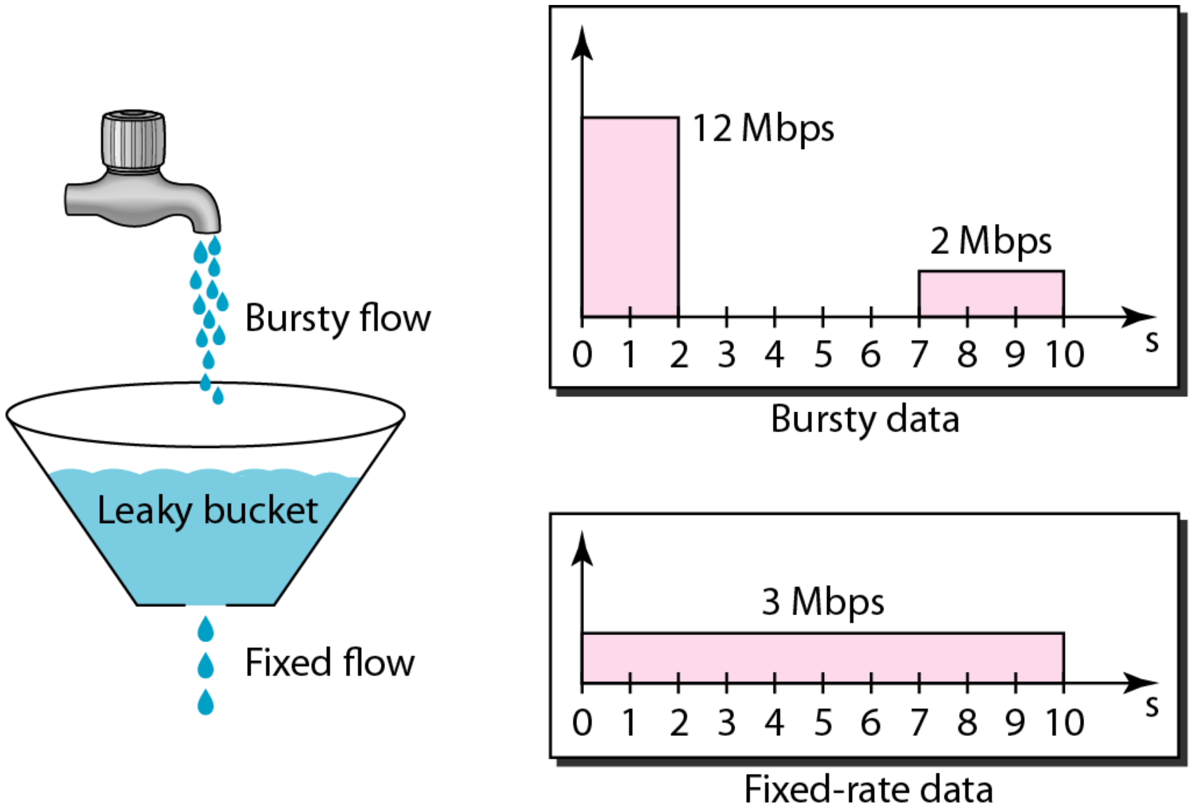
## **Figure 24.17** *Priority queuing*



## **Figure 24.18** *Weighted fair queuing*

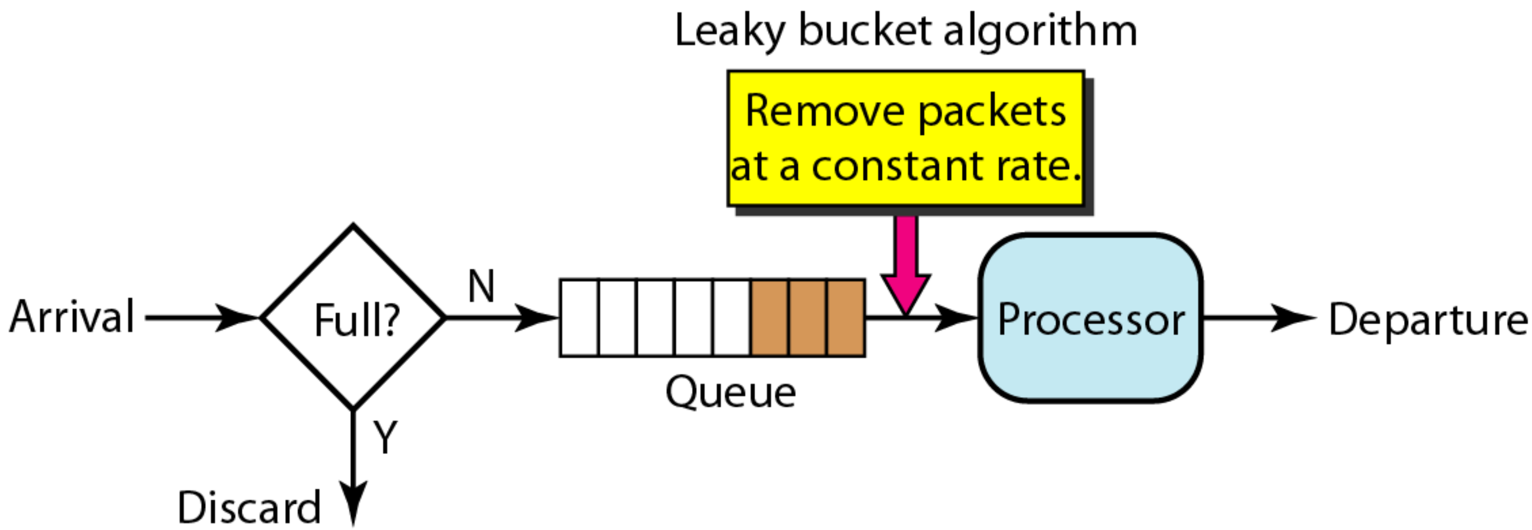


# Figure 24.19 *Leaky bucket*




**Traffic shaping: To control the amount and rate of traffic sent to the network**

## **Figure 24.20** *Leaky bucket implementation*




FIFO with fixed service rate

**Note**

**A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.**

# 24-7 INTEGRATED SERVICES

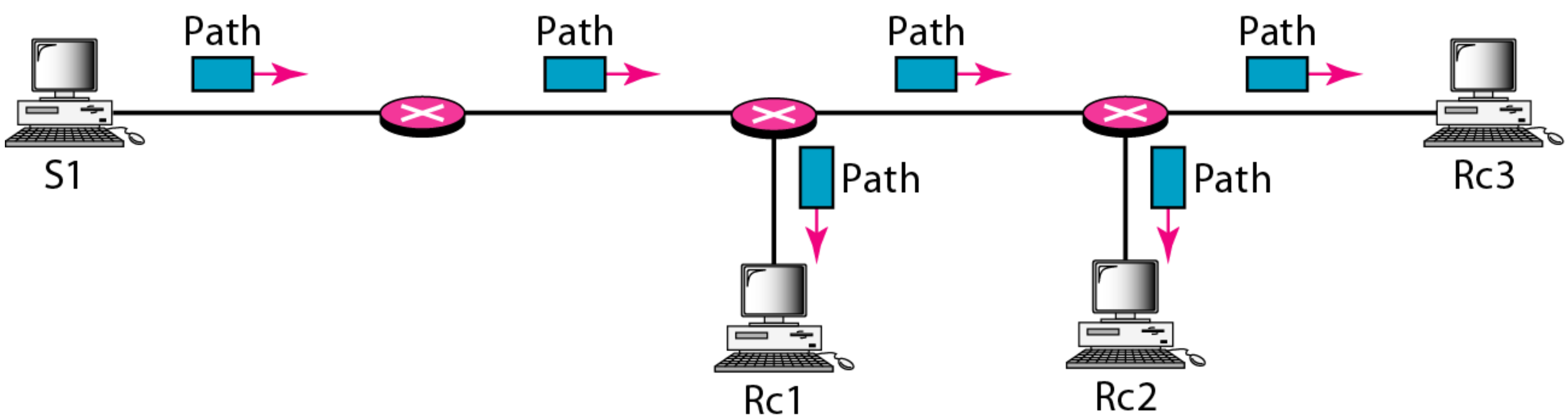
***Two models have been designed to provide quality of service in the Internet: Integrated Services and Differentiated Services. We discuss the first model here.***

**In IntServ, we need to create a flow, a kind of virtual circuit network, out of the IP, which was originally designed as a datagram packet-switched network.**
**Resource Reservation Protocol (RSVP) is a signaling protocol to help IP create a flow and consequently make a resource reservation.**
**RSVP is designed for multicasting. It can be used for unicasting as a special case of multicasting.**

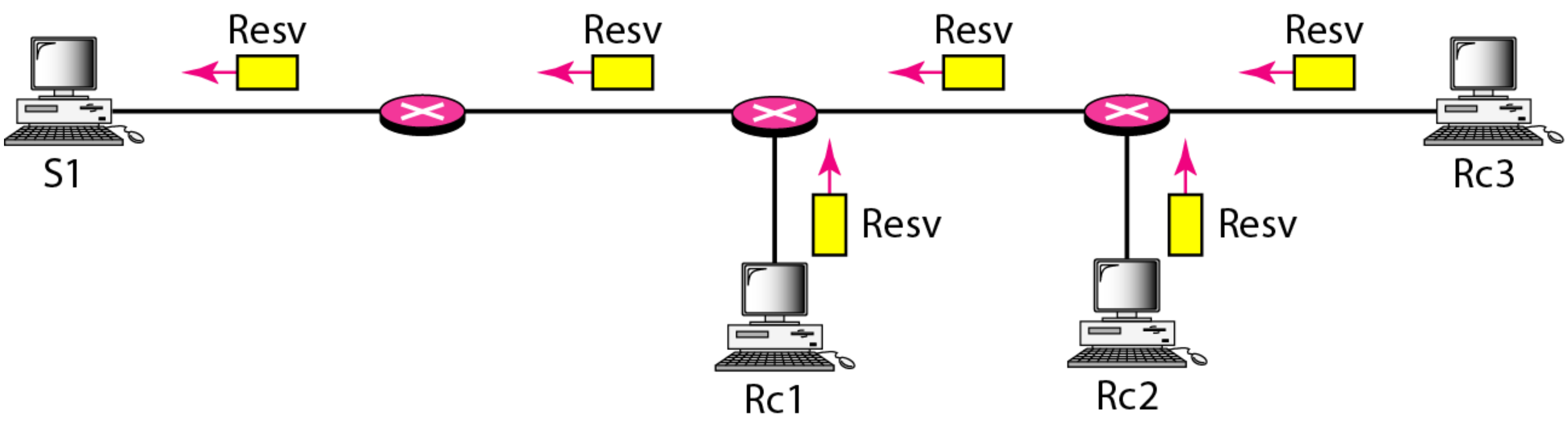
**Note**

**Integrated Services is a flow-based QoS model designed for IP.**

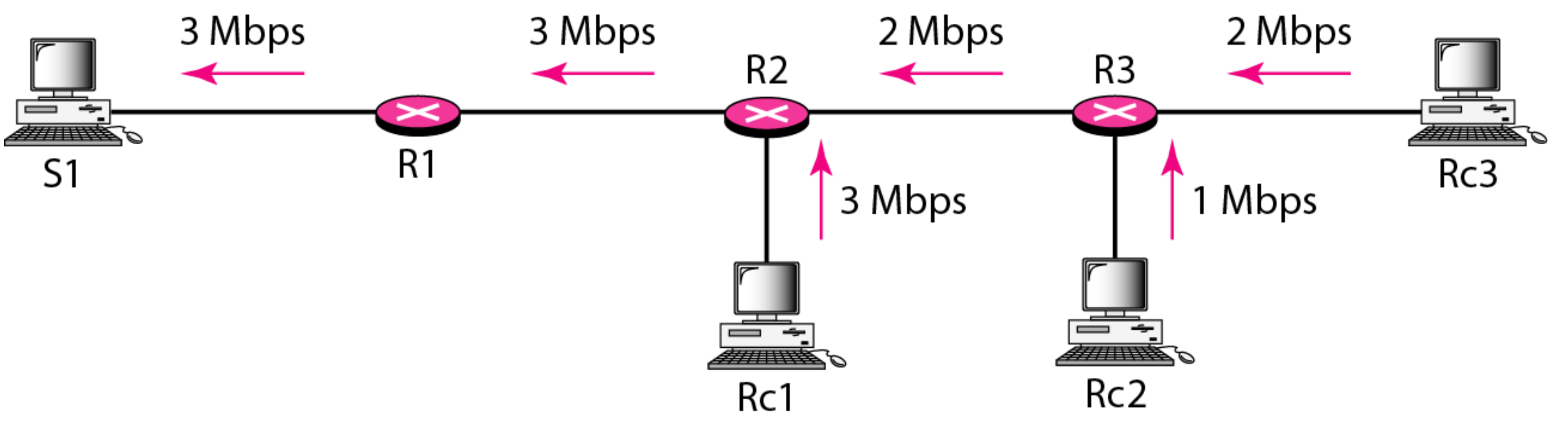
## Figure 24.22 *Path messages*



## Figure 24.23 *Resv messages*



## Figure 24.24 *Reservation merging*



# 24-8 DIFFERENTIATED SERVICES

***Differentiated Services (DS or Diffserv) was introduced by the IETF (Internet Engineering Task Force) to handle the shortcomings of Integrated Services.***

**Major Problem of IntServ:**
**Scalability: each router keeps info. for each flow.**

**Note**

**Differentiated Services is a class-based QoS model designed for IP.**

## Figure 24.26 *DS field*


**DS field is included in each packet.**

**DSCP (differentiated service code point): 6 bits, to define the per-hop behavior (PHB)**

- **for example, assured forwarding PHB: deliver the packets with a high assurance as long as the class traffic does not exceed the traffic profile of the node.**

**CU (currently unused)**

## Figure 24.27 *Traffic conditioner*