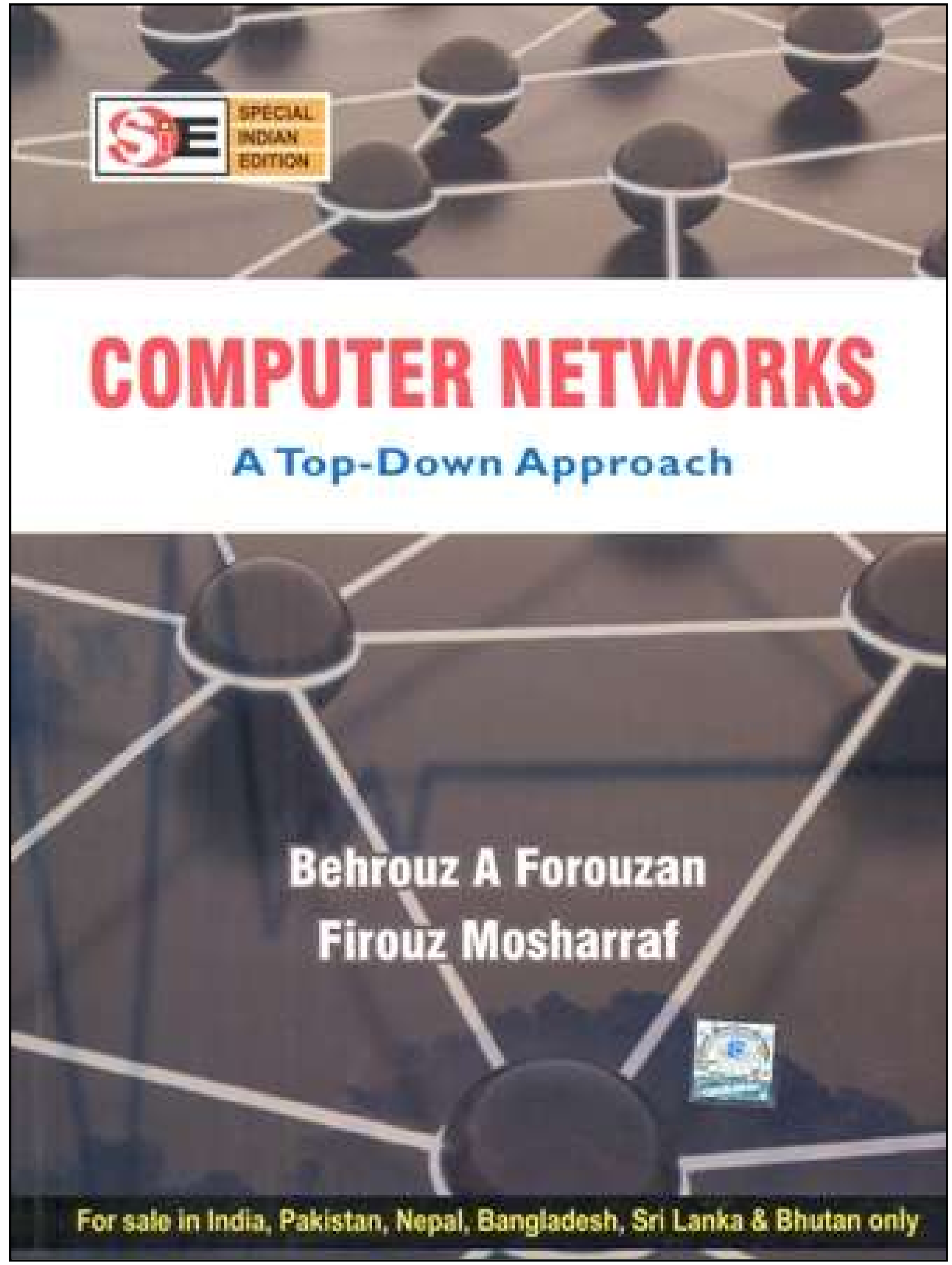
The McGraw-Hill Companies

Chapter 4

Network Layer



4-3 UNICAST ROUTING

In an internet, the goal of the network layer is to deliver a datagram from its source to its destination or destinations. If a datagram is destined for only one destination (one-to-one delivery), we have unicast routing. In this section and the next, we discuss only unicast routing; multicast and broadcast routing will be discussed later in the chapter.

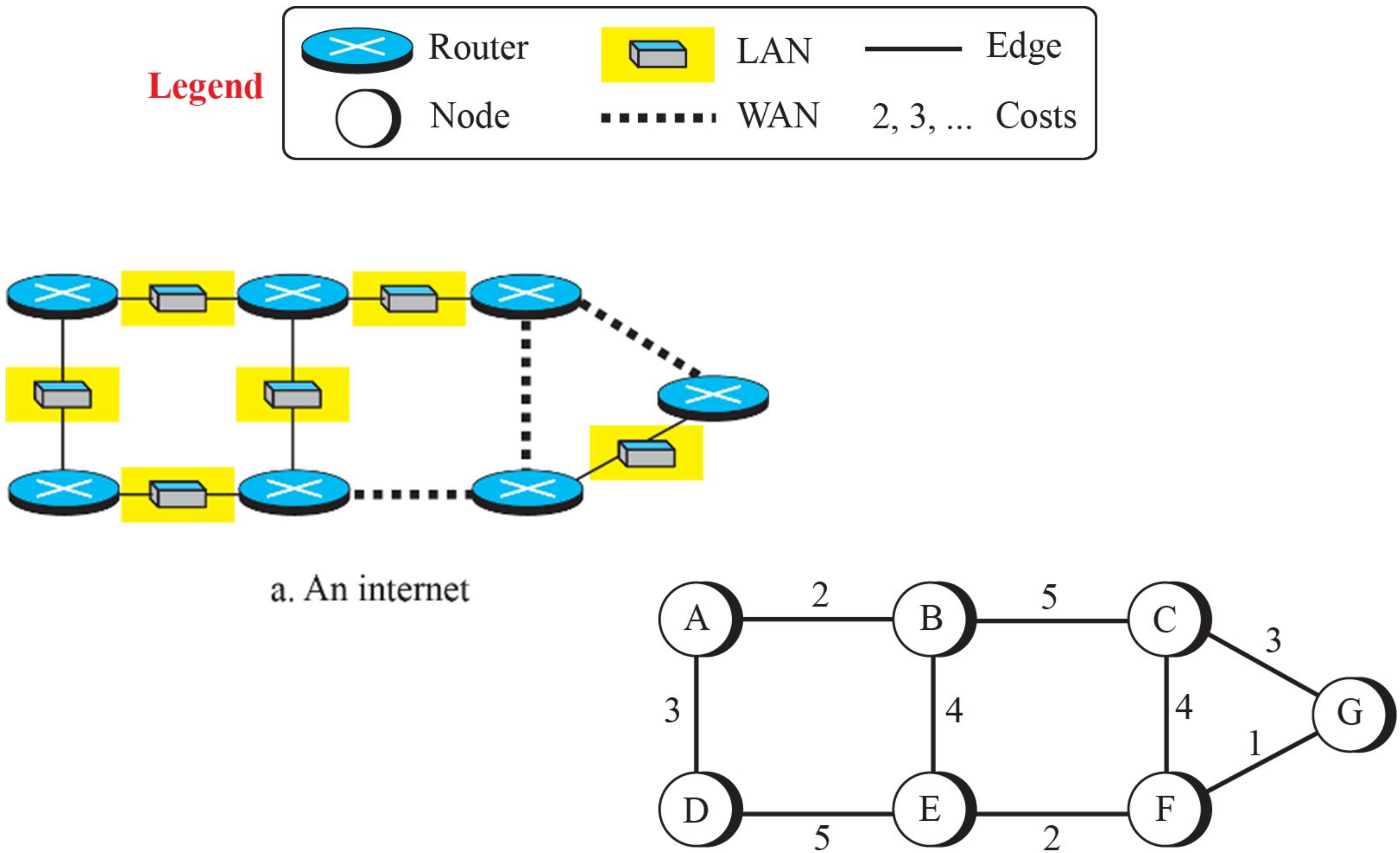
4.3.1 General Idea

In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables. The source host needs no forwarding table because it delivers its packet to the default router in its local network. The destination host needs no forwarding table either because it receives the packet from its default router in its local network. This means that only the routers that glue together the networks in the internet need forwarding tables.

4.3.1 (continued)

- An Internet as a Graph
- Least-Cost Routing
- Least-Cost Trees

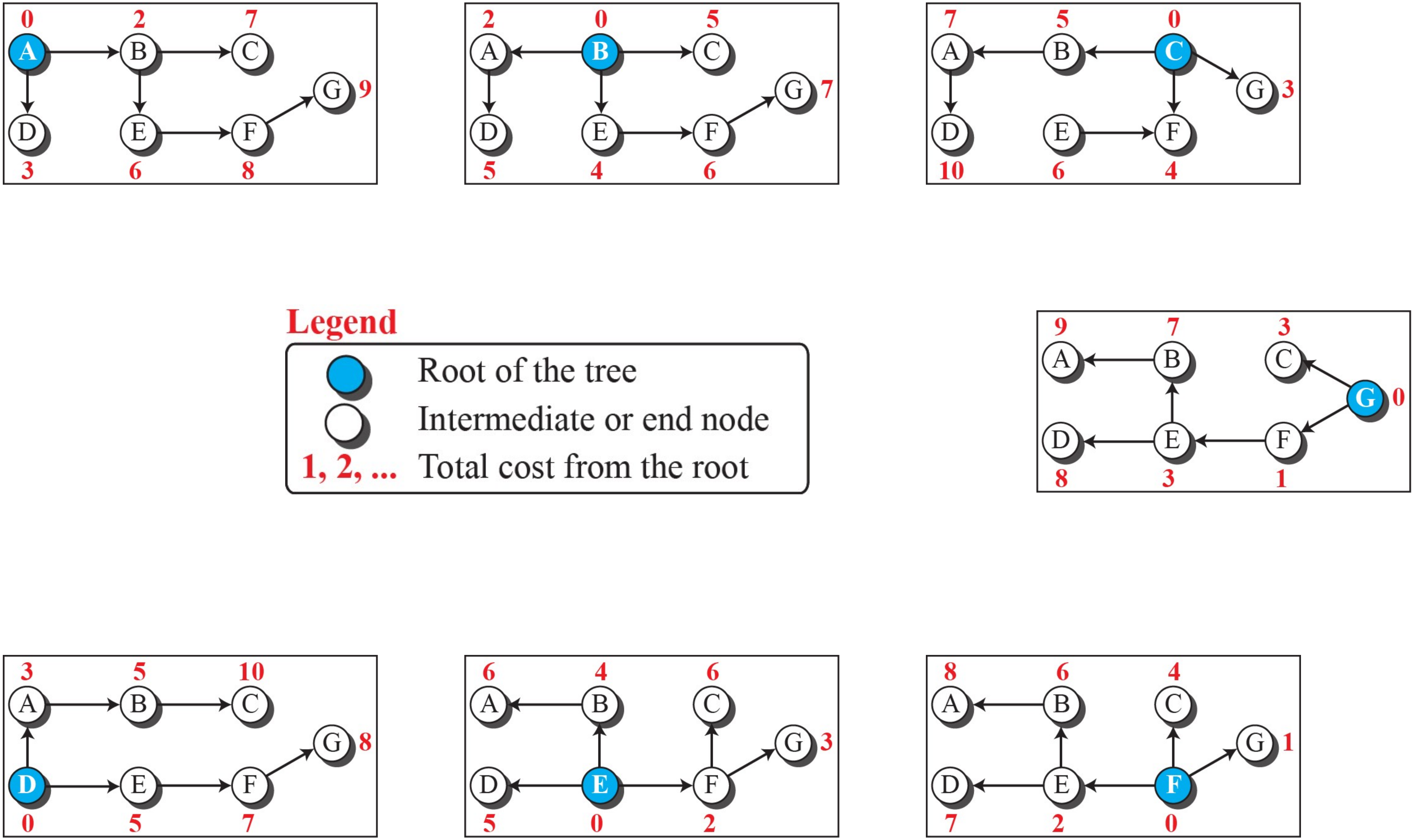
Figure 4.56: An internet and its graphical representation



a. An internet

b. The weighted graph

Figure 4.57: Least-cost trees for nodes in the internet of Figure 4.56



4.3.2 Routing Algorithm

After discussing the general idea behind least-cost trees and the forwarding tables that can be made from them, now we concentrate on the routing algorithms. Several routing algorithms have been designed in the past. The differences between these methods are in the way they interpret the least cost and the way they create the least-cost tree for each node. In this section, we discuss the common algorithm; later we show how a routing protocol in the Internet implements one of these algorithms.

4.3.2 (continued)

Distance-Vector Routing

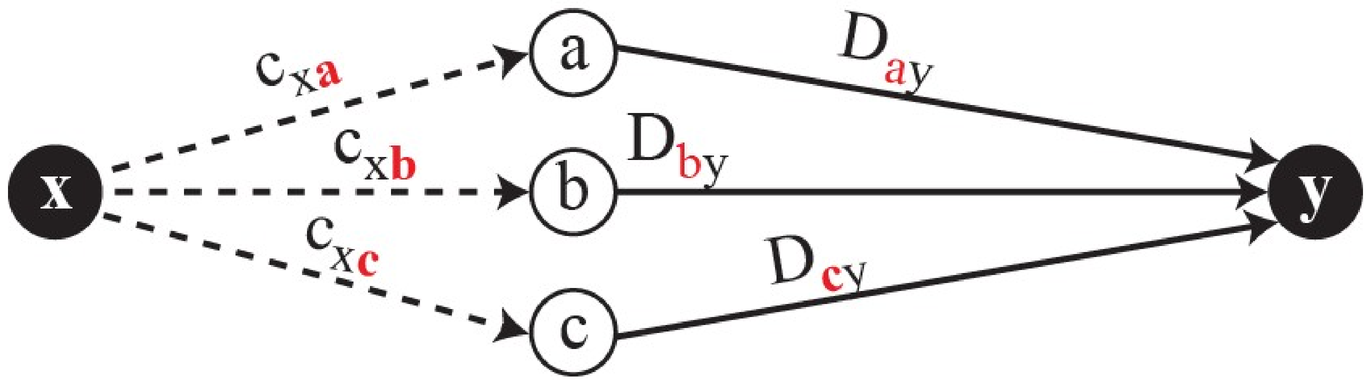
- ***Bellman-Ford Equation***
- ***Distance Vectors***
- ***Distance-Vector Routing Algorithm***
- ***Count to Infinity***
- ***Two-Node Loop***
  - ***Split Horizon***
  - ***Poisoned Reverse***
- ***Three-Node Instability***

4.3.2 (continued)

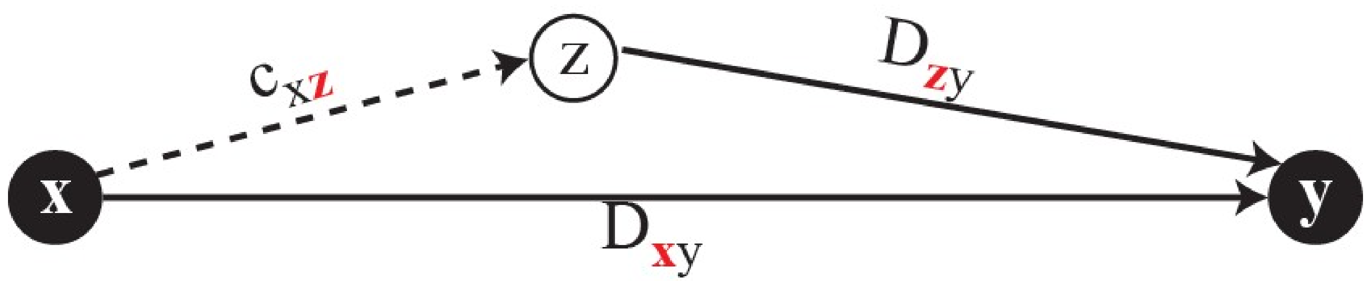
- ***Link-State Routing***
  - ***Link-State Database (LSDB)***
  - ***Least-Cost Trees (Dijkstra's Algorithm)***

- ***Path-Vector Routing***
  - ***Spanning Trees***
  - ***Creation of Spanning Trees***
  - ***Path-Vector Algorithm***

Figure 4.58: ***Graphical idea behind Bellman-Ford equation***

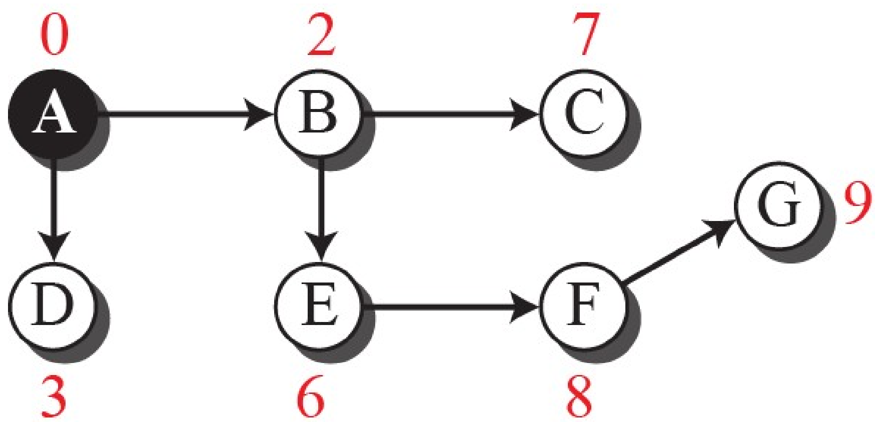


a. General case with three intermediate nodes



b. Updating a path with a new route

Figure 4.59: The distance vector corresponding to a tree



a. Tree for node A

	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

b. Distance vector for node A

Figure 4.60: The first distance vector for an internet

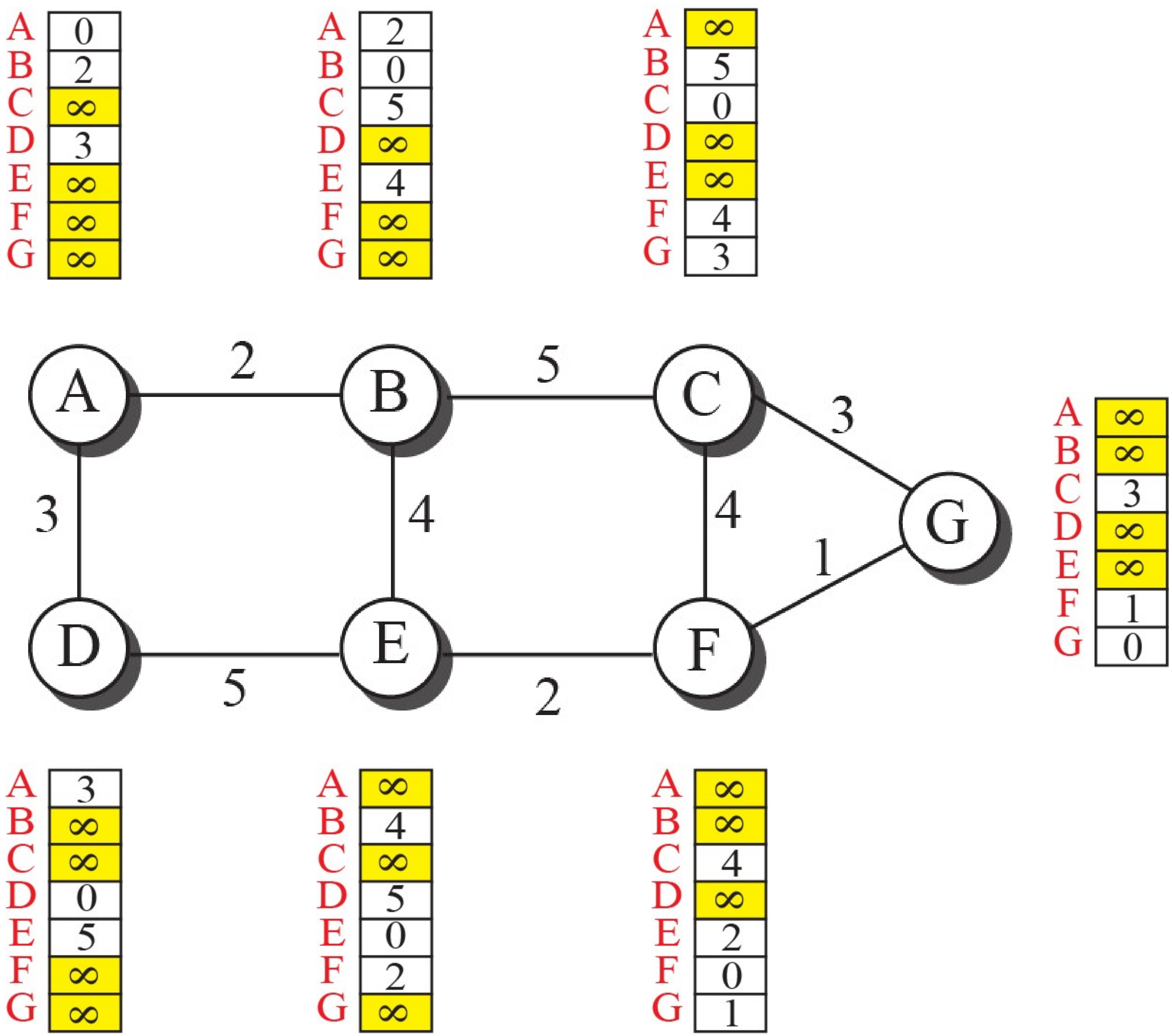


Figure 4.61: Updating distance vectors

	New B	Old B	A
A	2	2	0
B	0	0	2
C	5	5	∞
D	5	∞	3
E	4	4	∞
F	∞	∞	∞
G	∞	∞	∞

B[ ] = min (B[ ] , 2 + A[ ])

a. First event: B receives a copy of A's vector.

Note:
X[ ]: the whole vector

	New B	Old B	E
A	2	2	∞
B	0	0	4
C	5	5	∞
D	5	5	5
E	4	4	0
F	6	∞	2
G	∞	∞	∞

B[ ] = min (B[ ] , 4 + E[ ])

b. Second event: B receives a copy of E's vector.

Table 4.4: Distance-Vector Routing Algorithm for A Node

```
Distance_Vector_Routing ( )
{
    // Initialize (create initial vectors for the node)
    D[myself] = 0
    for (y = 1 to N)
    {
        if (y is a neighbor)
            D[y] = c[myself][y]
        else
            D[y] = ∞
    }
    send vector {D[1], D[2], …, D[N]} to all neighbors
    // Update (improve the vector with the vector received from a neighbor)
    repeat (forever)
    {
        wait (for a vector D_w from a neighbor w or any change in the link)
        for (y = 1 to N)
        {
            D[y] = min [D[y], (c[myself][w] + D_w[y])]        // Bellman-Ford equation
        }
        if (any change in the vector)
            send vector {D[1], D[2], …, D[N]} to all neighbors
    }
} // End of Distance Vector
```

Figure 4.62: Two-node instability

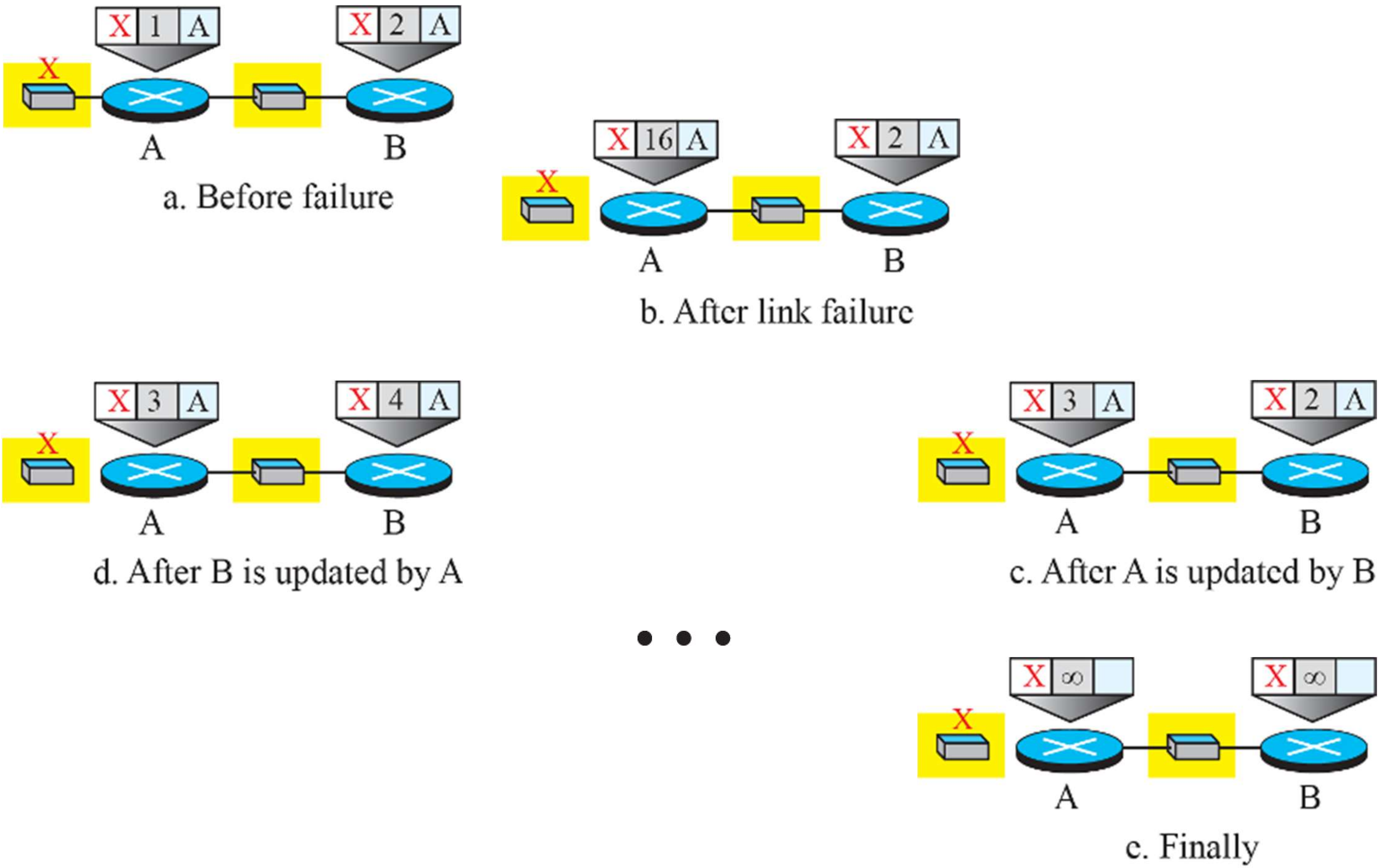
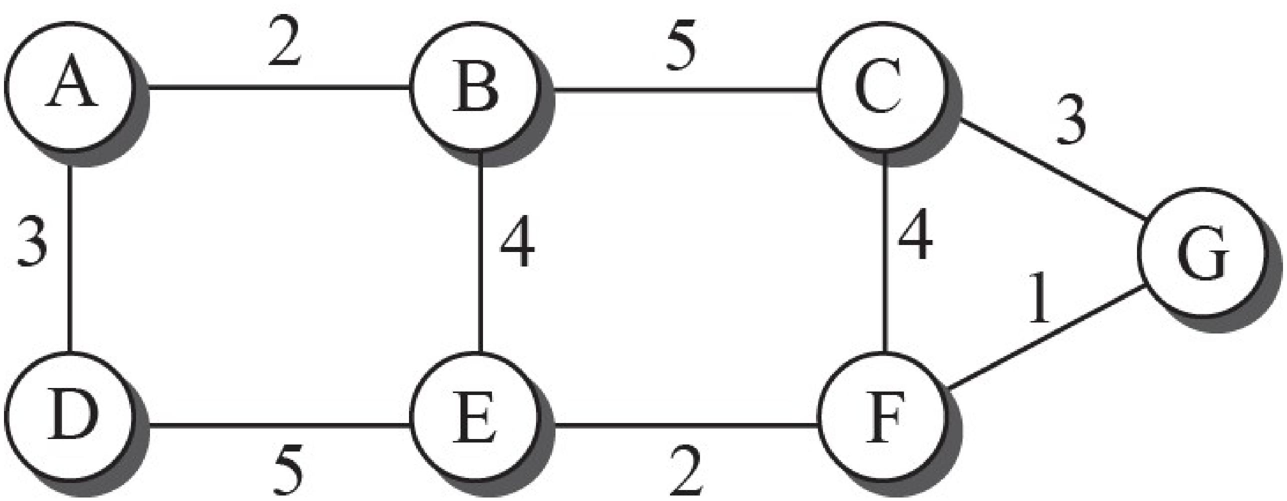


Figure 4.63: ***Example of a link-state database***



a. The weighted graph

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

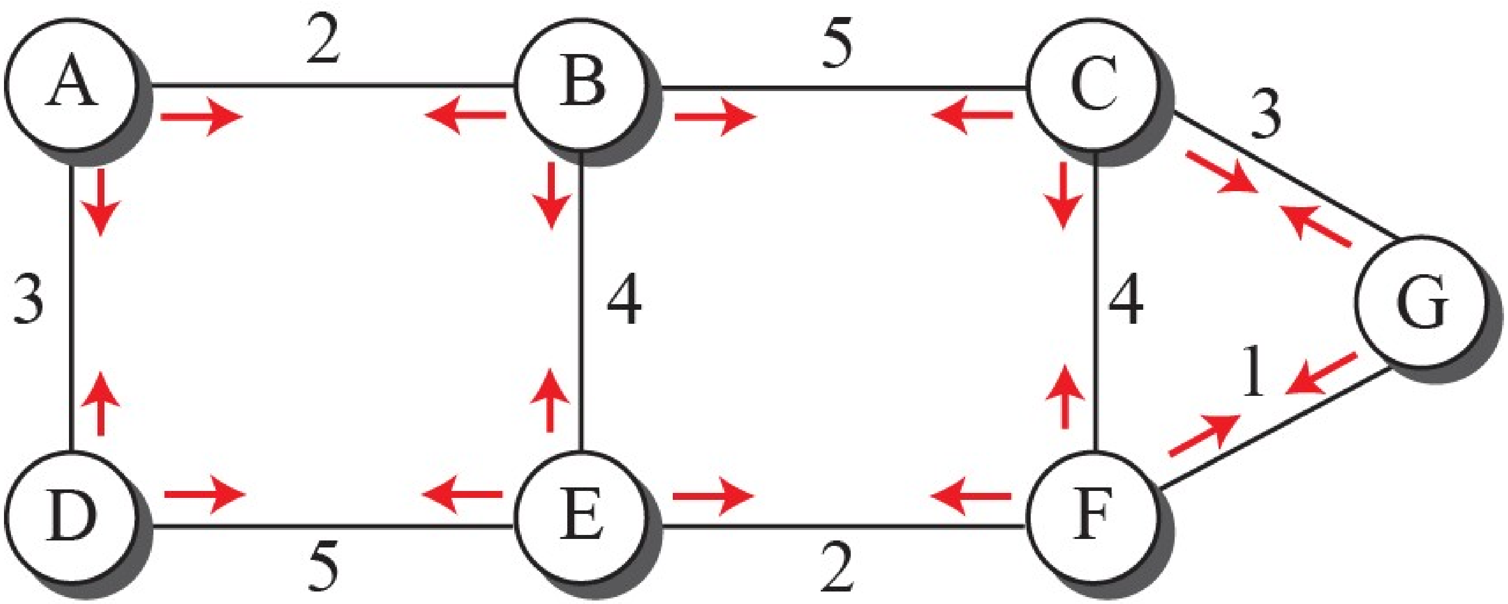
b. Link state database

Figure 4.64: ***LSPs created and sent out by each node to build LSDB***

Node	Cost
B	2
D	3

Node	Cost
A	2
C	5
E	4

Node	Cost
B	5
F	4
G	3



Node	Cost
C	3
F	1

Node	Cost
A	3
E	5

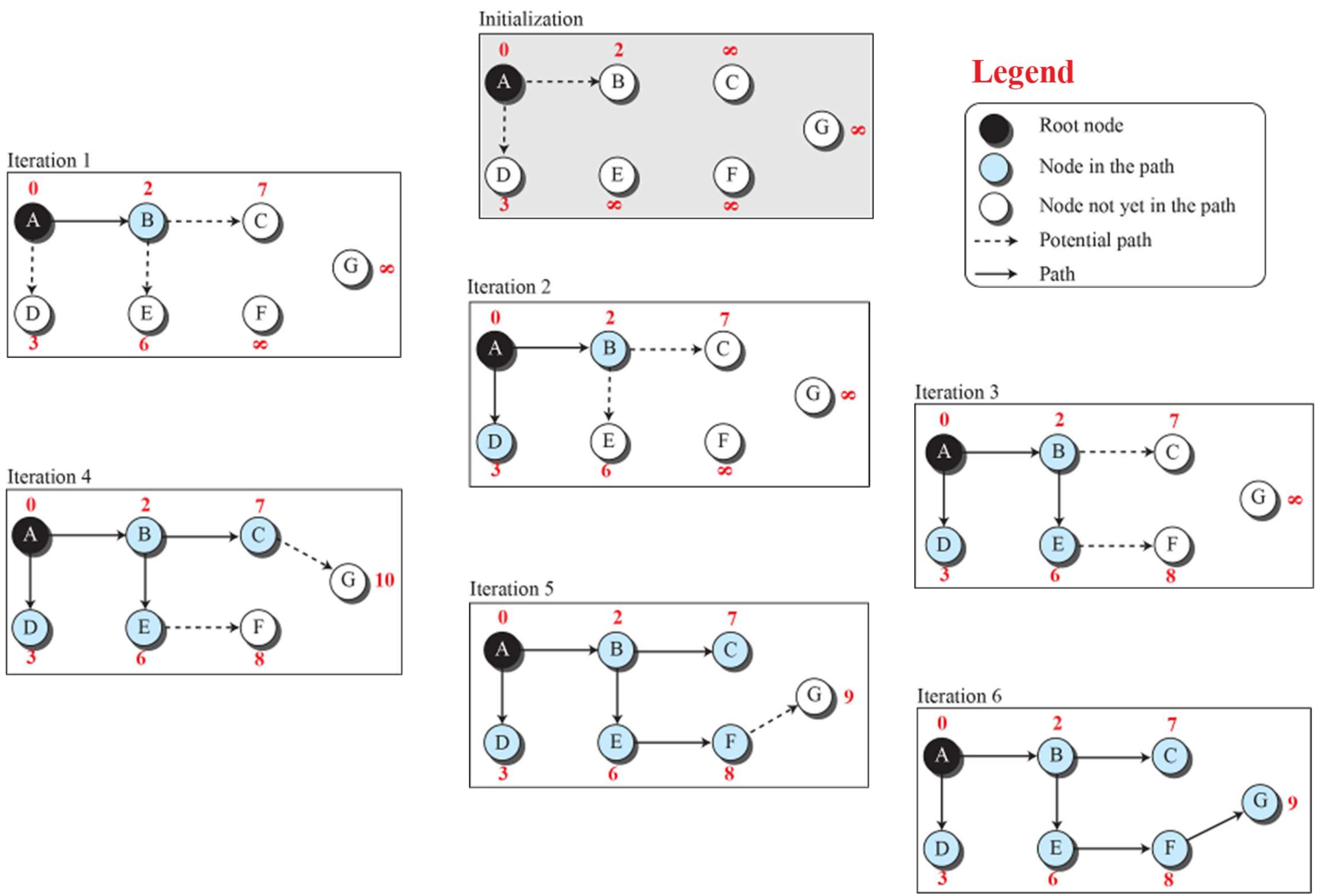
Node	Cost
B	4
D	5
E	2

Node	Cost
C	4
E	2
G	1

Table 4.5: Dijkstra's Algorithm

```
Dijkstra's Algorithm ( )
{
        // Initialization
        Tree = {root}                     // Tree is made only of the root
        for (y = 1 to N)                  // N is the number of nodes
        {
                if (y is the root)
                        D[y] = 0          // D[y] is shortest distance from root to node y
                else if (y is a neighbor)
                        D[y] = c[root][y] // c[x][y] is cost between nodes x and y in LSDB
                else
                        D[y] = ∞
        }
        // Calculation
        repeat
        {
                find a node w, with D[w] minimum among all nodes not in the Tree
                Tree = Tree ∪ {w}         // Add w to tree
                // Update distances for all neighbor of w
                for (every node x, which is neighbor of w and not in the Tree)
                {
                        D[x] = min{D[x], (D[w] + c[w][x])}
                }
        } until (all nodes included in the Tree)
} // End of Dijkstra
```

Figure 4.65: ***Least-cost tree***



4.3.3 Unicast Routing Protocols

In the previous section, we discussed unicast routing algorithms; in this section, we discuss unicast routing protocols used in the Internet. Although three protocols we discuss here are based on the corresponding algorithms we discussed before, a protocol is more than an algorithm. A protocol needs to define its domain of operation, the messages exchanged, communication between routers, and interaction with protocols in other domains. After an introduction, we discuss three common protocols used in the Internet: RIP, OSPF, and BGP.

4.3.3 (continued)

- **Internet Structure**
  - Hierarchical Routing
  - Autonomous Systems

- **Routing Information Protocol (RIP)**
  - Hop Count
  - Forwarding Tables
  - RIP Implementation
  - Performance

4.3.3 (continued)

- **Open Shortest Path First (OSPF)**
  - Metric
  - Forwarding Tables
  - Areas
  - Link-State Advertisement
  - OSPF Implementation
  - Performance
- **Border Gateway Protocol Version 4 (BGP4)**
  - Introduction
  - Path Attributes
  - Route Selection
  - Messages
  - Performance

Figure 4.69: *Internet structure*

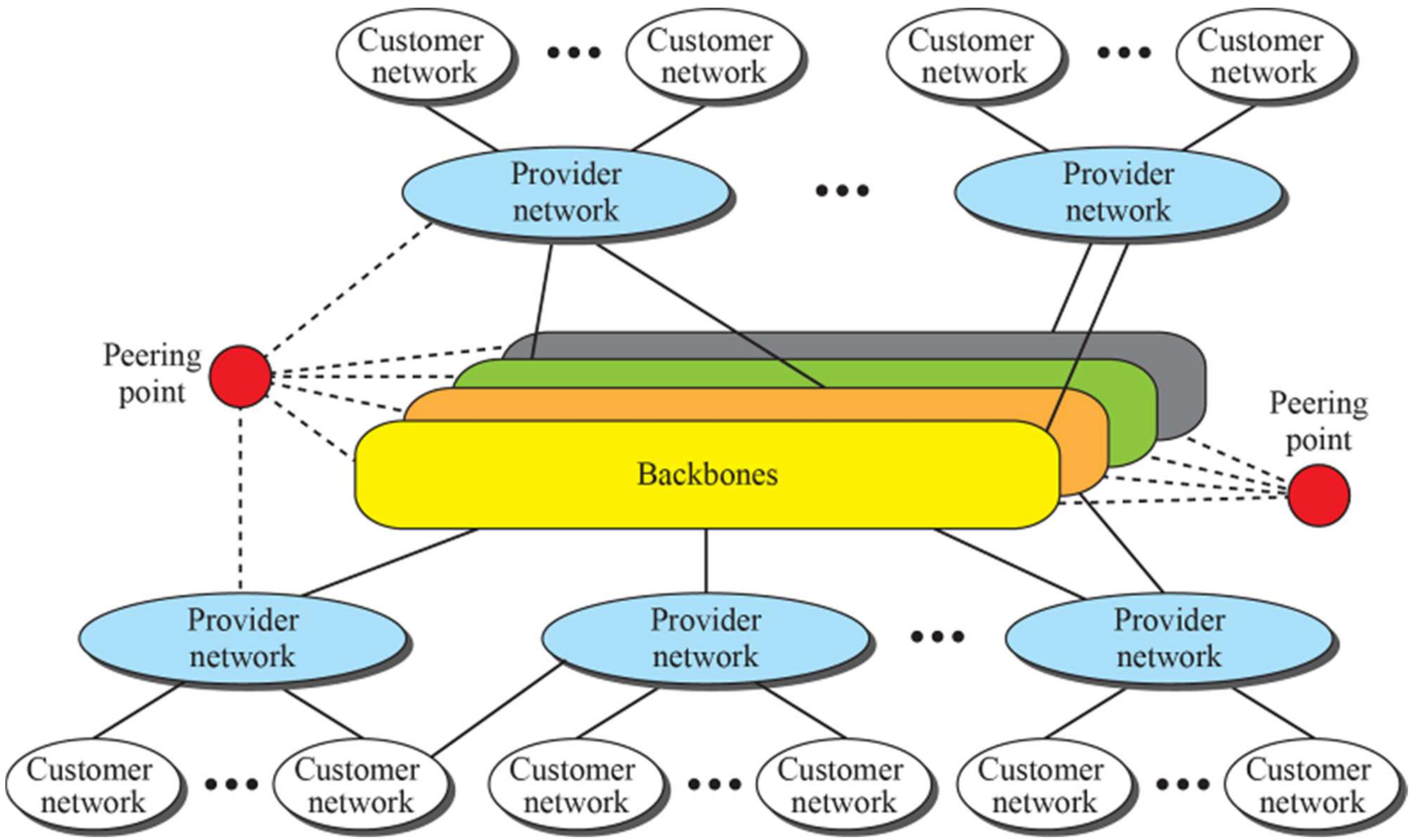


***Figure 4.70:* Hop counts in RIP (Distance vector)**

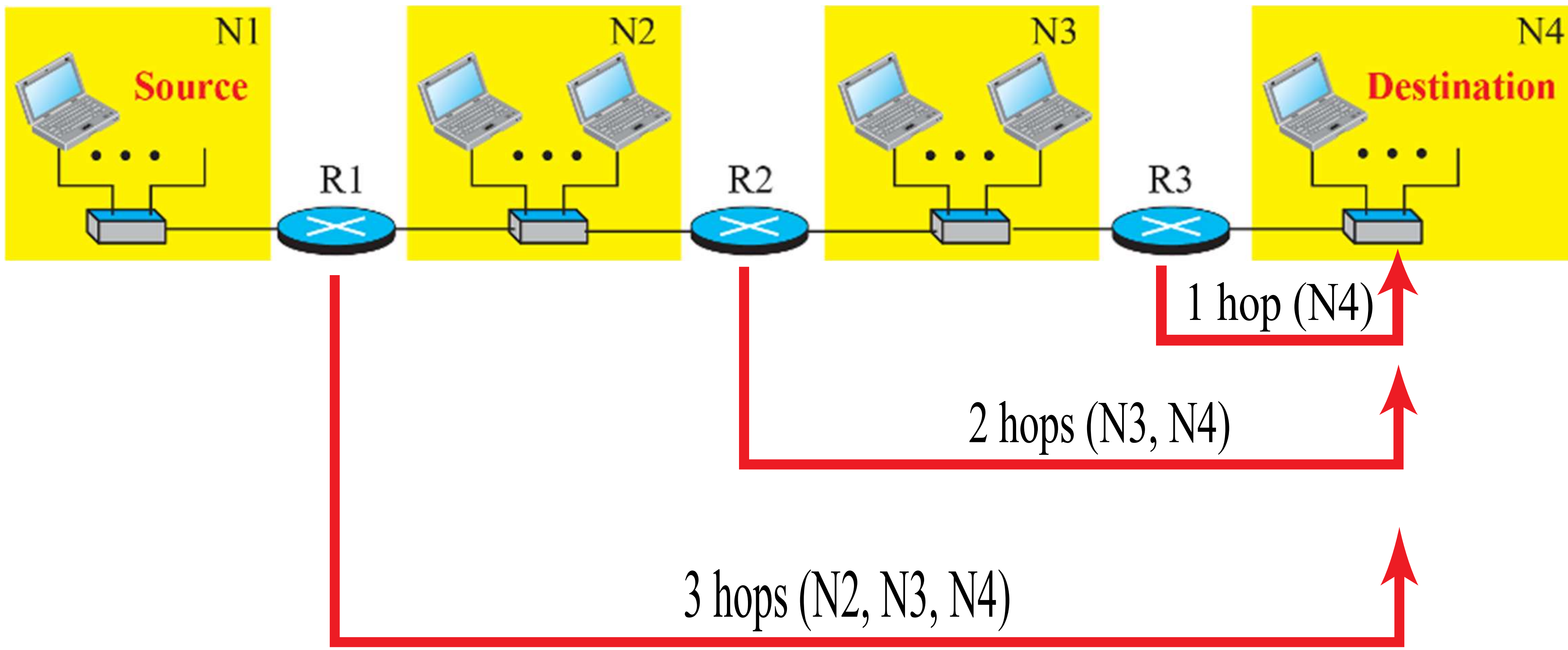


Figure 4.71: Forwarding tables

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

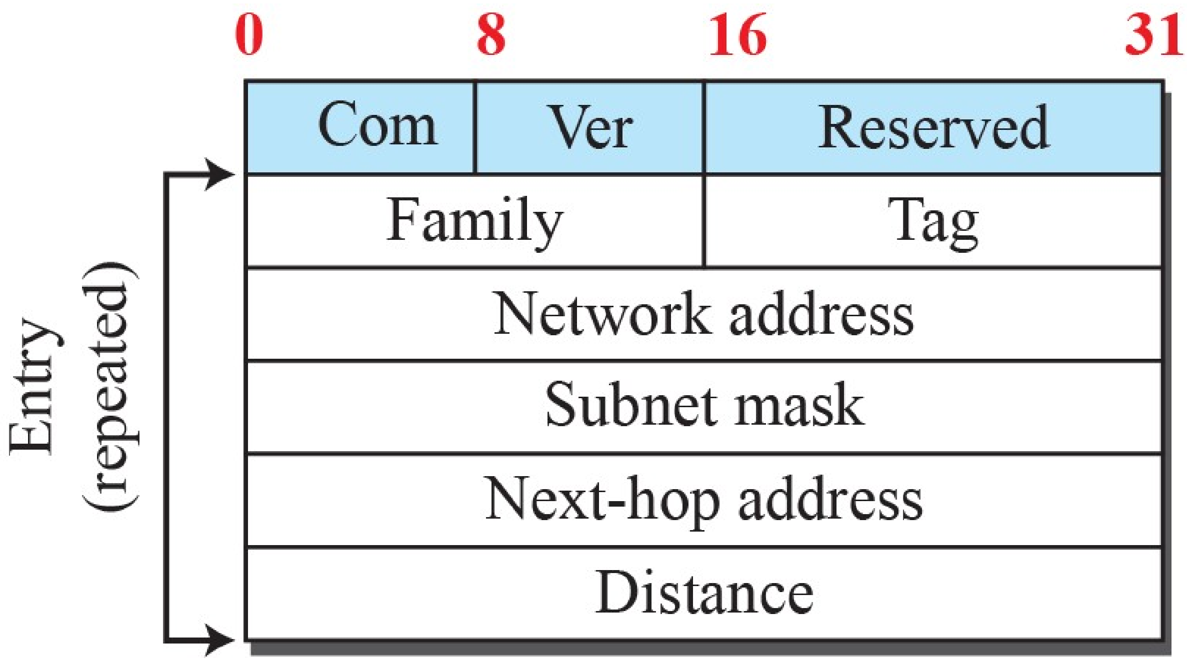
Forwarding table for R2

Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	——	1
N4	——	1

Figure 4.72: RIP message format



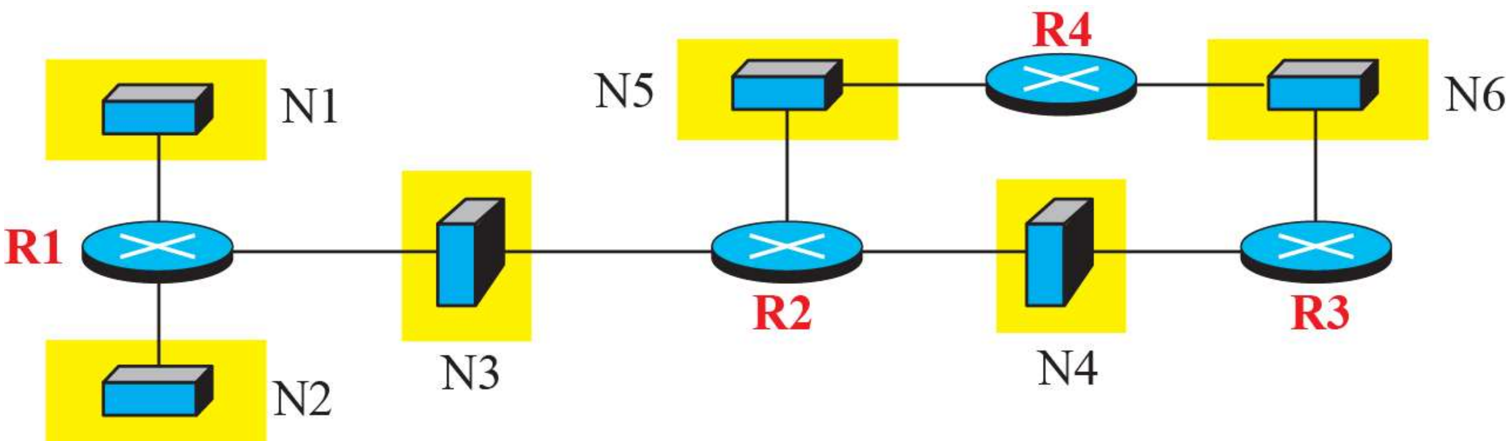
Fields

Com: Command, request (1), response (2)
Ver: Version, current version is 2
Family: Family of protocol, for TCP/IP value is 2
Tag: Information about autonomous system
Network address: Destination address
Subnet mask: Prefix length
Next-hop address: Address length
Distance: Number of hops to the destination

Example 4.15

Figure 4.73 shows a more realistic example of the operation of RIP in an autonomous system. First, the figure shows all forwarding tables after all routers have been booted. Then we show changes in some tables when some update messages have been exchanged. Finally, we show the stabilized forwarding tables when there is no more change.

Figure 4.73: Example of an autonomous system using RIP (Part I)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops

R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2

Des.	N. R.	Cost
N3	—	1
N4	—	1
N5	—	1

R3

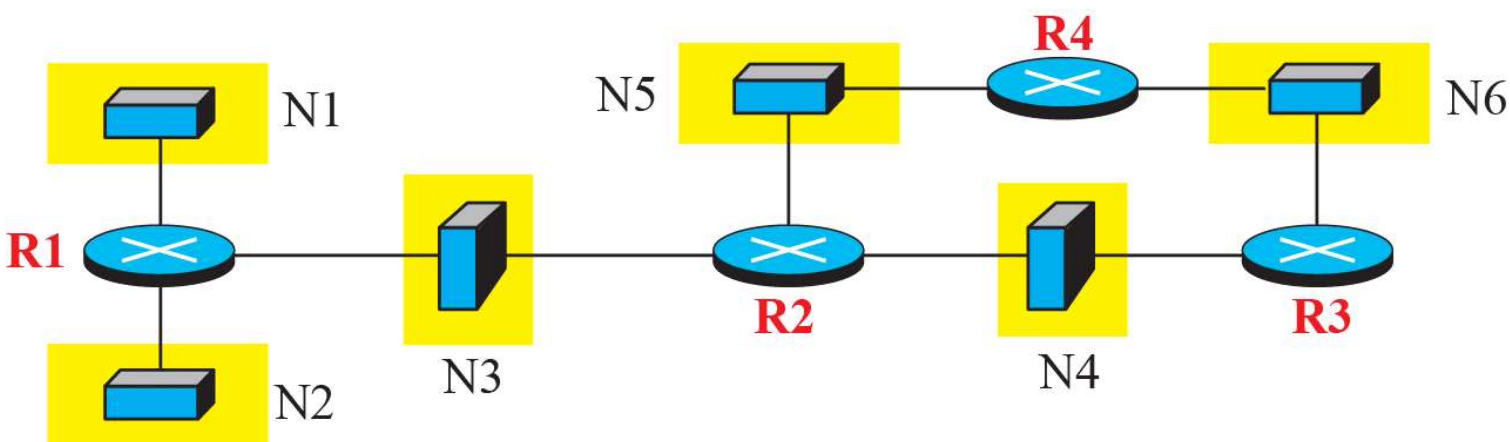
Des.	N. R.	Cost
N4	—	1
N6	—	1

R4

Des.	N. R.	Cost
N5	—	1
N6	—	1

Forwarding tables after all routers booted

Figure 4.73: Example of an autonomous system using RIP (Part II)



Legend

Des.: Destination network
N. R.: Next router
Cost: Cost in hops
← : New route
← : Old route

New R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1
N4	R2	2
N5	R2	2

Old R1

Des.	N. R.	Cost
N1	—	1
N2	—	1
N3	—	1

R2 Seen by R1

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R3

Des.	N. R.	Cost
N3	R2	2
N4	—	1
N5	R2	2
N6	—	1

Old R3

Des.	N. R.	Cost
N4	—	1
N6	—	1

R2 Seen by R3

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

New R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	—	1
N6	—	1

Old R4

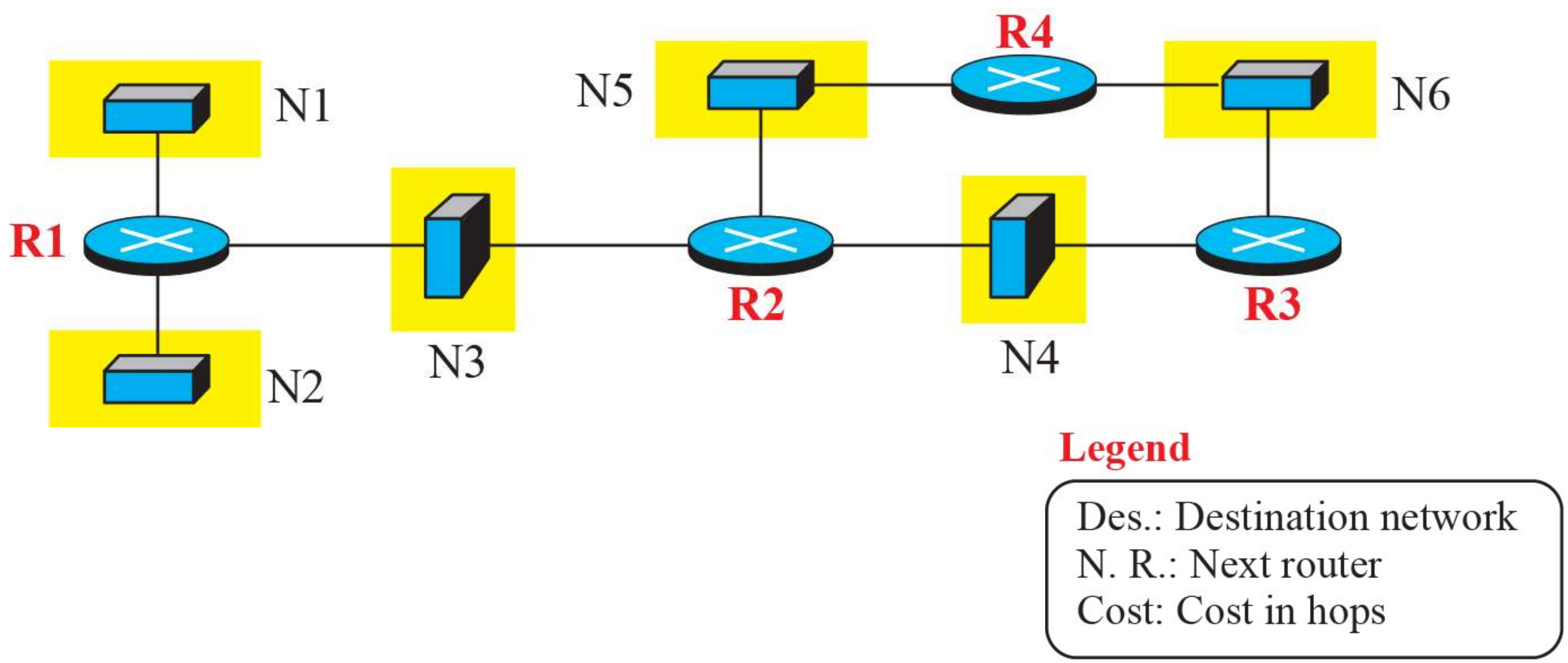
Des.	N. R.	Cost
N5	—	1
N6	—	1

R2 Seen by R4

Des.	N. R.	Cost
N3	R2	2
N4	R2	2
N5	R2	2

Changes in the forwarding tables of R1, R3, and R4 after they receive a copy of R2's table

Figure 4.73: Example of an autonomous system using RIP (Part III)



Forwarding tables for all routers
after they have been stablized

Final R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1
N4	R2	2
N5	R2	2
N6	R2	3

Final R2

Des.	N. R.	Cost
N1	R1	2
N2	R1	2
N3	———	1
N4	———	1
N5	———	1
N6	R3	2

Final R3

Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	———	1
N5	R2	2
N6	———	1

Final R4

Des.	N. R.	Cost
N1	R2	3
N2	R2	3
N3	R2	2
N4	R2	2
N5	———	1
N6	———	1

Figure 4.74: *Metric in OSPF (Link state)*

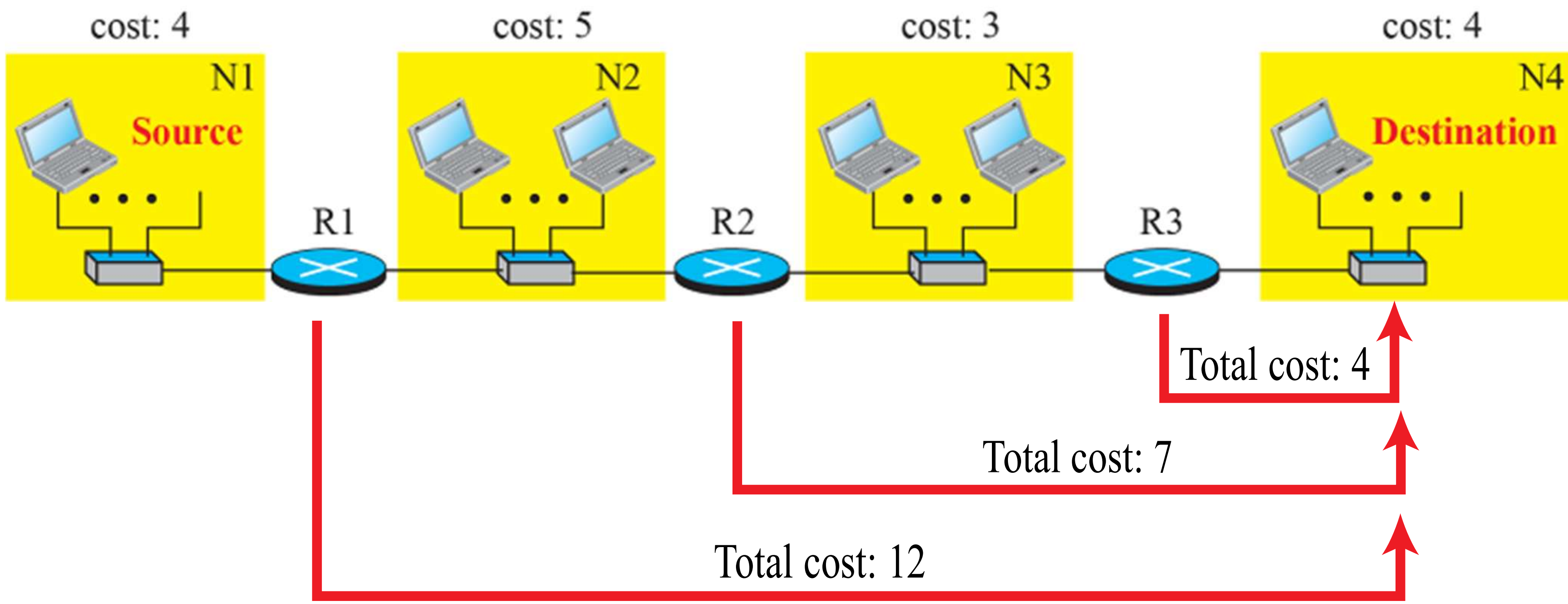
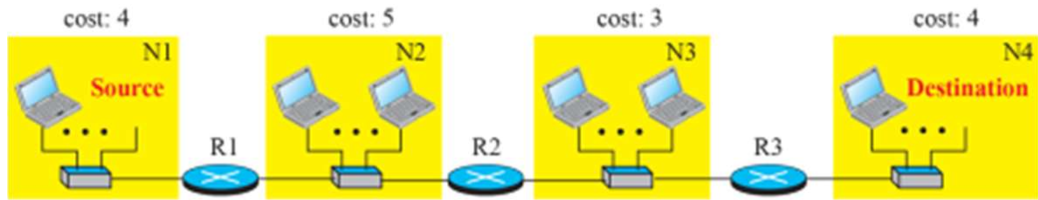


Figure 4.75: Forwarding tables in OSPF

Forwarding table for R1

Destination network	Next router	Cost
N1	——	
N2	——	
N3	R2	
N4	R2	12



The internet from previous figure

Forwarding table for R2

Destination network	Next router	Cost
N1	R1	9
N2	——	5
N3	——	3
N4	R3	7

Forwarding table for R3

Destination network	Next router	Cost
N1	R2	12
N2	R2	8
N3	——	3
N4	——	4

Figure 4.76: Areas in an autonomous system

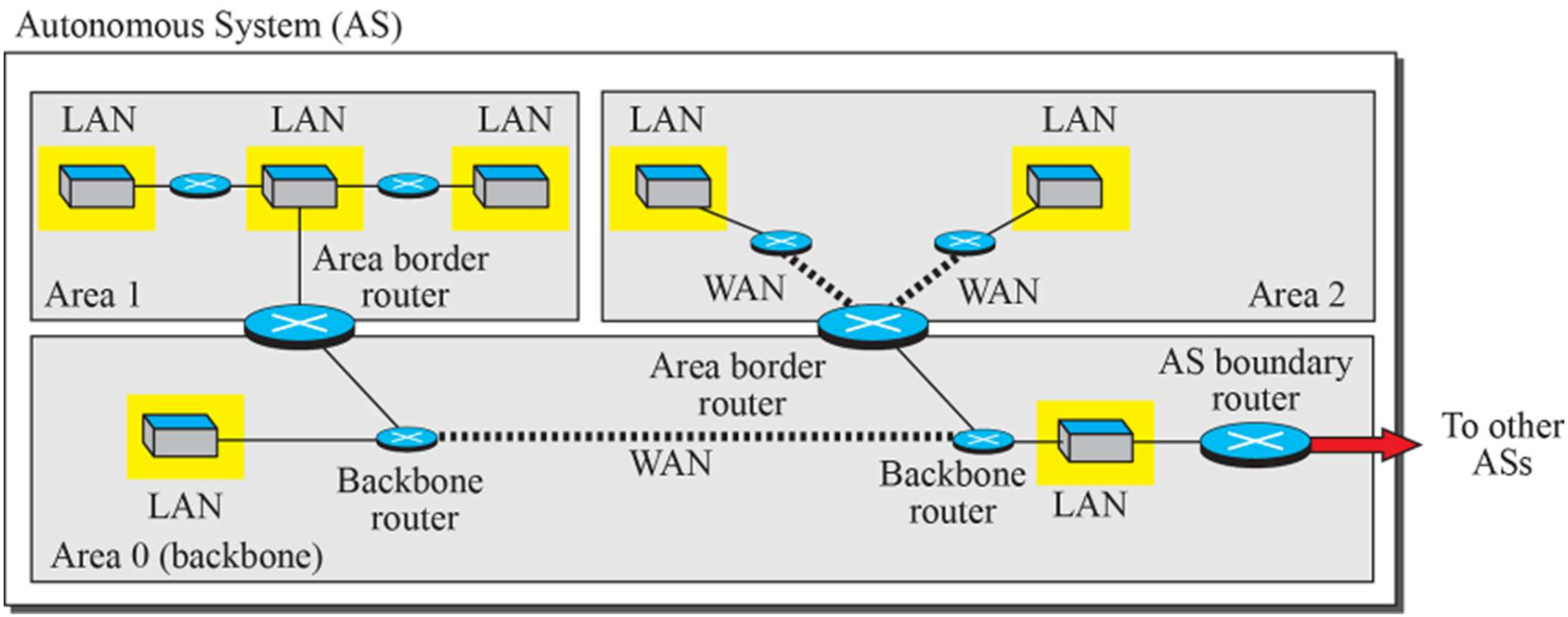


Figure 4.79: A sample internet with four ASs

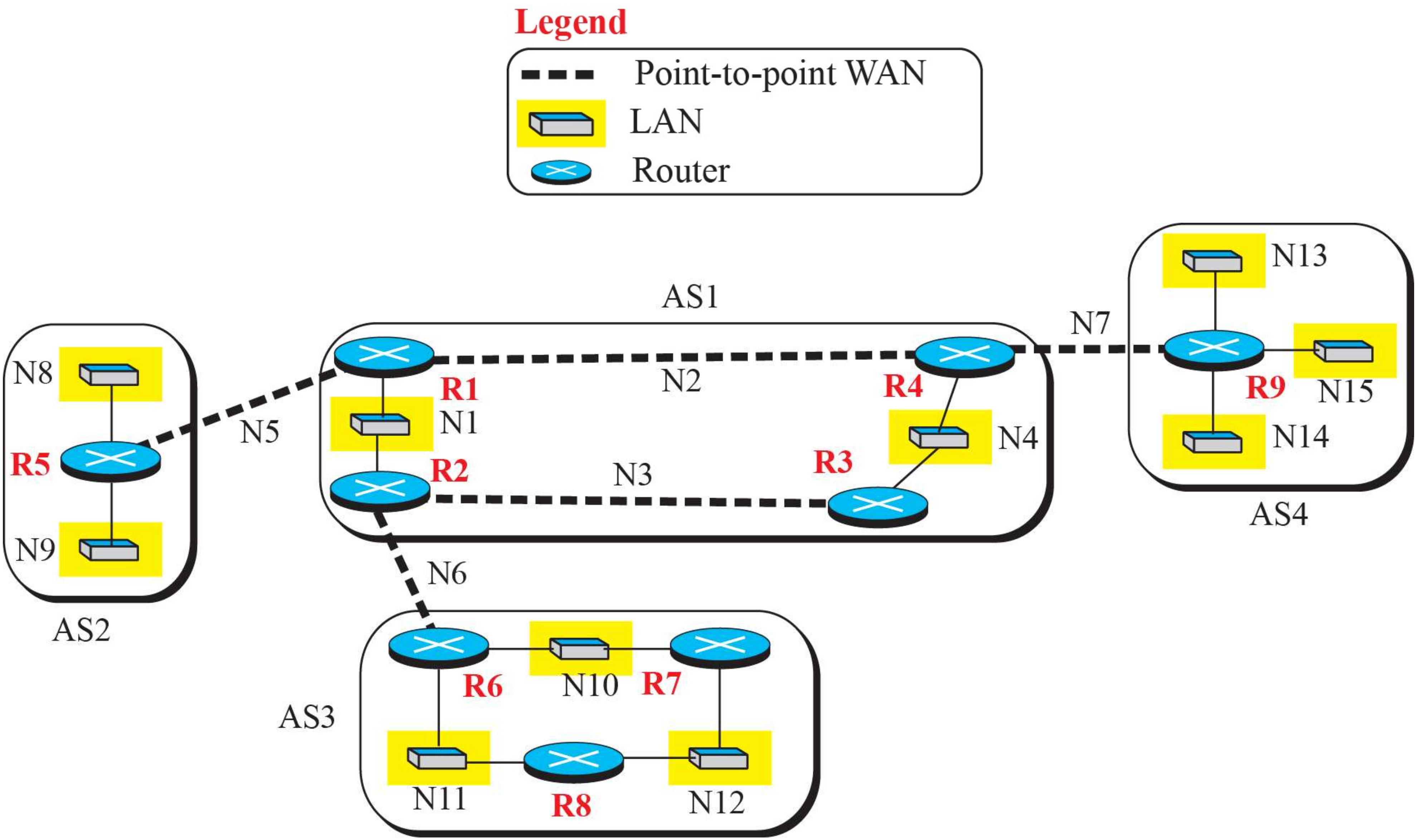
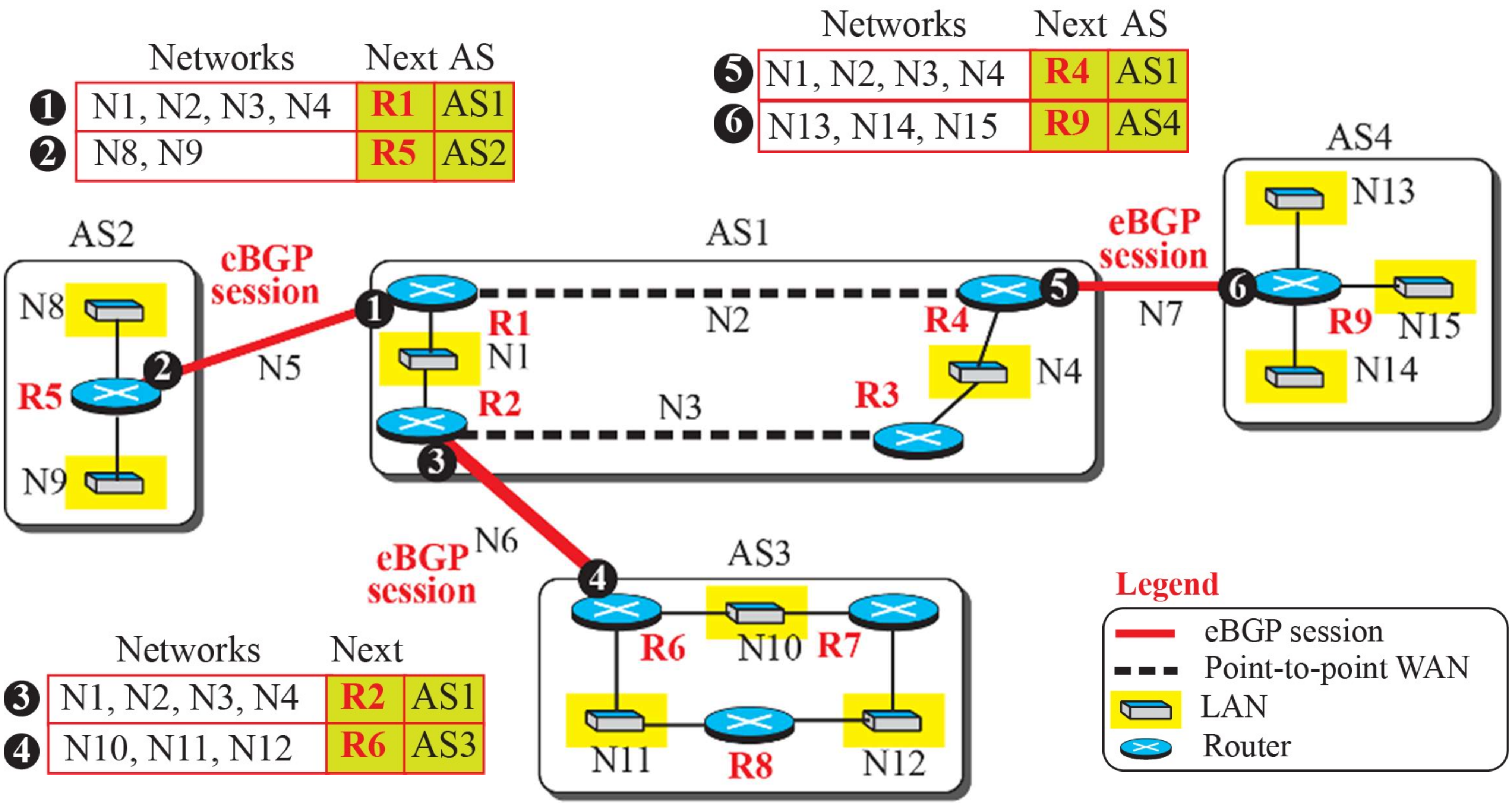


Figure 4.80: eBGP operation



Chapter 4: Summary

- ***The Internet is made of many networks (or links) connected through connecting devices, each of which acts as a router or as a switch. Two types of switching are traditionally used in networking: circuit switching and packet switching. The network layer is designed as a packet-switched network.***

- ***The network layer supervises the handling of packets by the underlying physical networks. The delivery of a packet can be direct or indirect. Two categories of forwarding are defined: forwarding based on the destination address of the IP datagram and forwarding based on the label attached to an IP datagram.***

Chapter 4: Summary (continued)

- ***IPv4 is an unreliable connectionless protocol responsible for source-to-destination delivery. Packets in the IP layer are called datagrams. The identifiers used in the IP layer of the TCP/IP protocol suite are called the IP addresses. An IPv4 address is 32 bits long, divided into two parts: the prefix and the suffix. All addresses in the block have the same prefix; each address has a different suffix.***

- ***The Internet Control Message Protocol (ICMP) supports the unreliable and connectionless Internet Protocol (IP).***

- ***To be able to route a packet, a router needs a forwarding table. Routing protocols are specific application programs that have the duty of updating forwarding tables.***

Chapter 4: Summary (continued)

- ***Multicasting is the sending of the same message to more than one receiver simultaneously. The Internet Group Management Protocol (IGMP) is involved in collecting local membership group information.***

- ***IPv6, the latest version of the Internet Protocol, has a 128-bit address space. IPv6 uses hexadecimal colon notation with abbreviation methods available.***