Unicast Routing Protocols

Dr Mohit Kumar
Assistant Professor
Dept. of Information Technology
National Institute of Technology
Jalandhar-144008, Punjab India

Internet Protocol (IP)

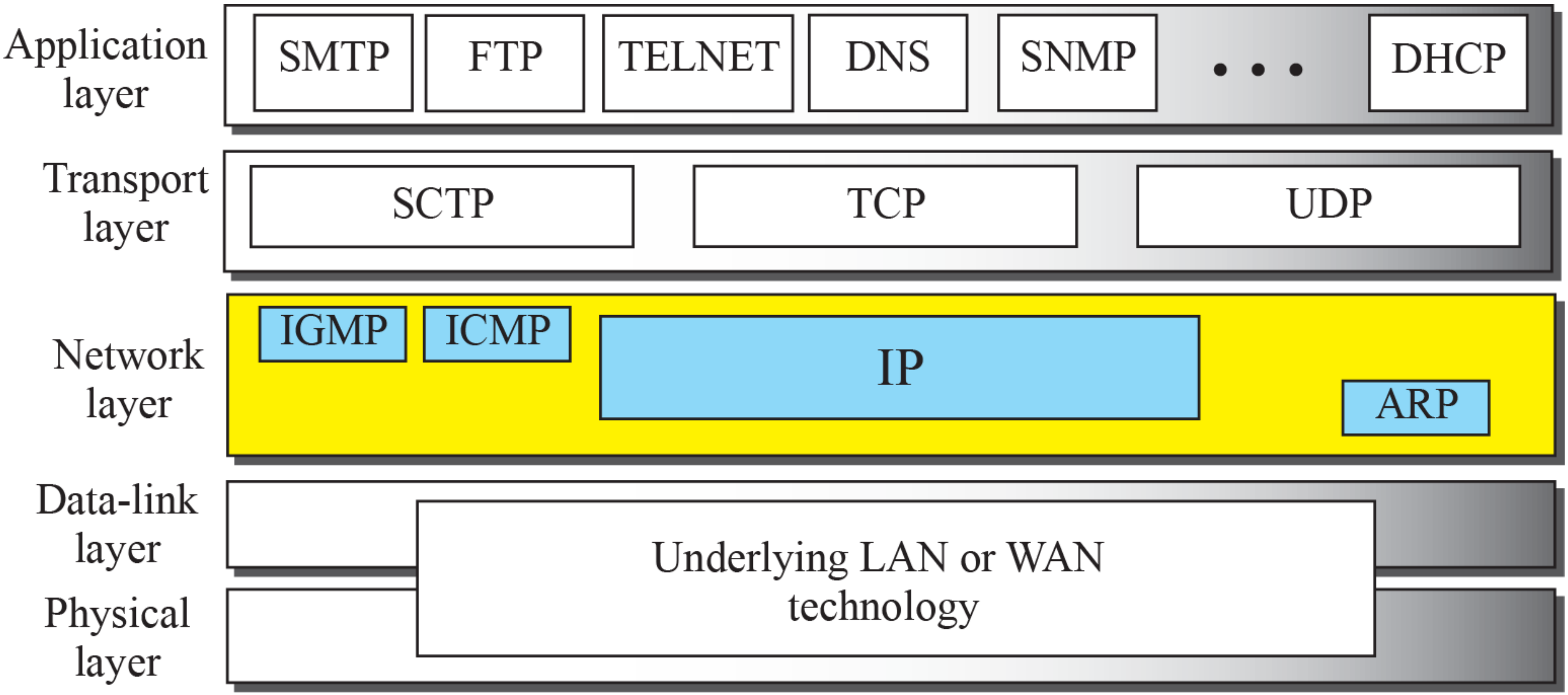


Figure 1: ***Position of IP and other network-layer protocols in TCP/IP protocol suite***

Datagram Format

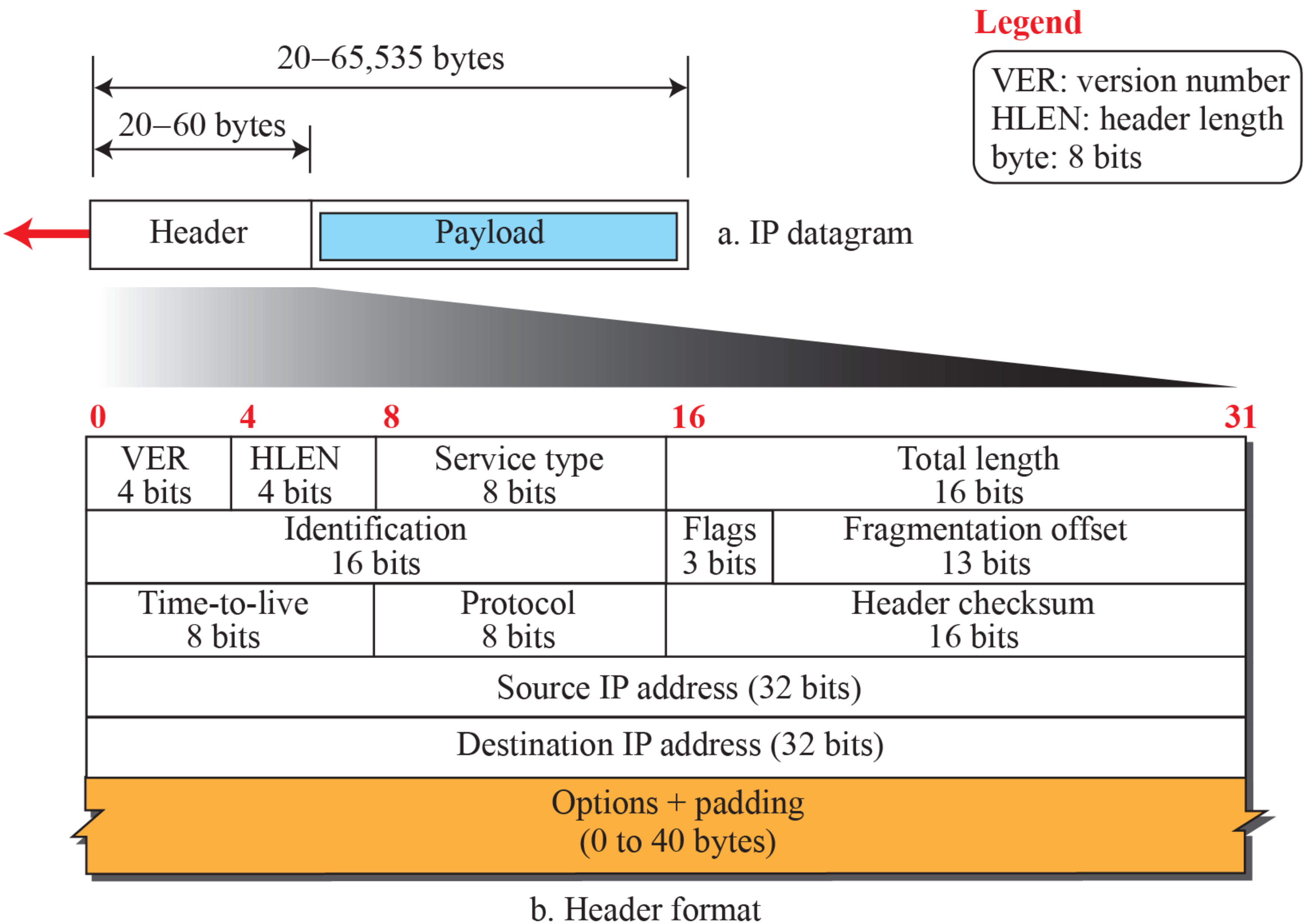
- ✓ ***Packets used by the IP are called datagrams. Figure 2 shows the IPv4 datagram format.***

- ✓ ***A datagram is a variable-length packet consisting of two parts: header and payload (data).***

- ✓ ***The header is 20 to 60 bytes in length and contains information essential to routing and delivery.***

- ✓ ***It is customary in TCP/IP to show the header in 4-byte sections.***

Figure 2: IP datagram



- An IPV4 packet, the value of HLEN is 1000 in binary. How many of options are being carried by this packet ?

- In an IPV4 packet, the value of HLEN is 5, and value of total length field is 0x0028. How many bytes of data are being carried by this packet.

- The host or datagram that fragments a datagram must change the values of three fields: flags, fragmentation offset, total length.
- Identification: This 16 bit field identifies a datagram originating from source host.
- The combination of identification and source IPV4 address must uniquely define a datagram as it leave the source host.

- **Fragment Offset** field (13 bits) is used to indicate the starting position of the data in the fragment in relation to the start of the data in the original packet.
- This information is used to reassemble the data from all the fragments (whether they arrive in order or not).
- In the first fragment the offset is 0 as the data in this packet starts in the same place as the data in the original packet (at the beginning).
- In subsequent fragments, the value is the offset of the data the fragment contains from the beginning of the data in the first fragment (offset 0), in 8 byte ‘blocks’.
- If a packet containing 800 bytes of data is split into two equal fragments carrying 400 bytes of data, the fragment offset of the first fragment is 0, of the second fragment 50 (400/8).
- The offset value must be the number of 8 byte blocks of data, which means the data in the prior fragment must be a multiple of 8 bytes.
- The last fragment can carry data that isn’t a multiple of 8 bytes as there won’t be a further fragment with an offset that must meet the 8 byte ‘rule’.

- In an IPv4 datagram, the M bit is 0, the value of HLEN is 10, the value of total length is 400 and the fragment offset value is 300. The position of the datagram, the sequence numbers of the first and the last bytes of the payload, respectively are

(A) Last fragment, 2400 and 2789
(B) First fragment, 2400 and 2759
(C) Last fragment, 2400 and 2759
(D) Middle fragment, 300 and 689

An IP router with a Maximum Transmission Unit (MTU) of 200 bytes has received an IP packet of size 520 bytes with an IP header of length 20 bytes. The values of the relevant fields in the IP header.

1 Introduction

2 Intra- and Inter-Domain Routing

3 Distance Vector Routing

4 RIP

5 Link State Routing

6 OSPF

7 Path Vector Routing

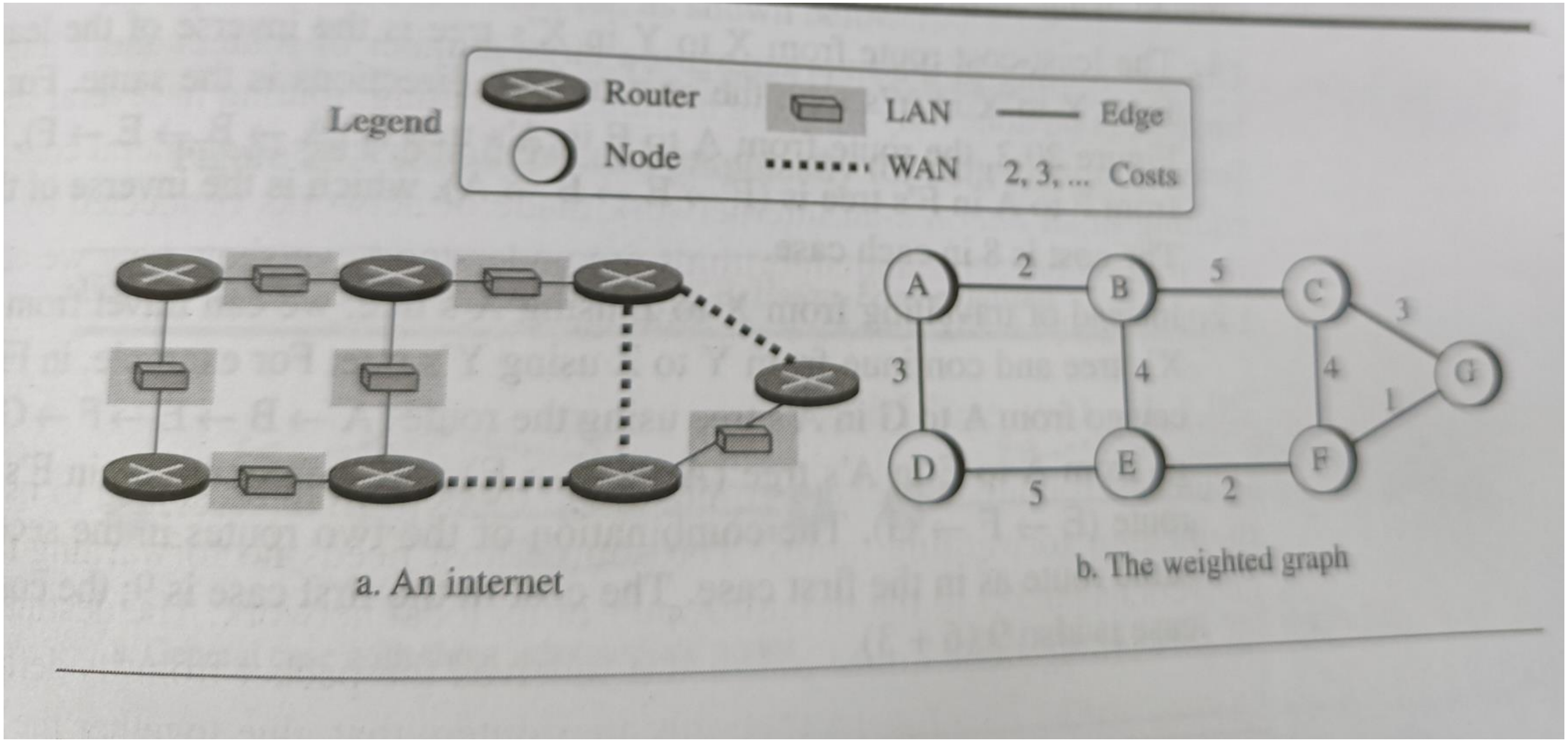
8 BGP

1 INTRODUCTION

An internet is a combination of networks connected by routers. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.

- ✓ **Cost or Metric**
- ✓ **Static versus Dynamic Routing Table**
- ✓ **Routing Protocol**

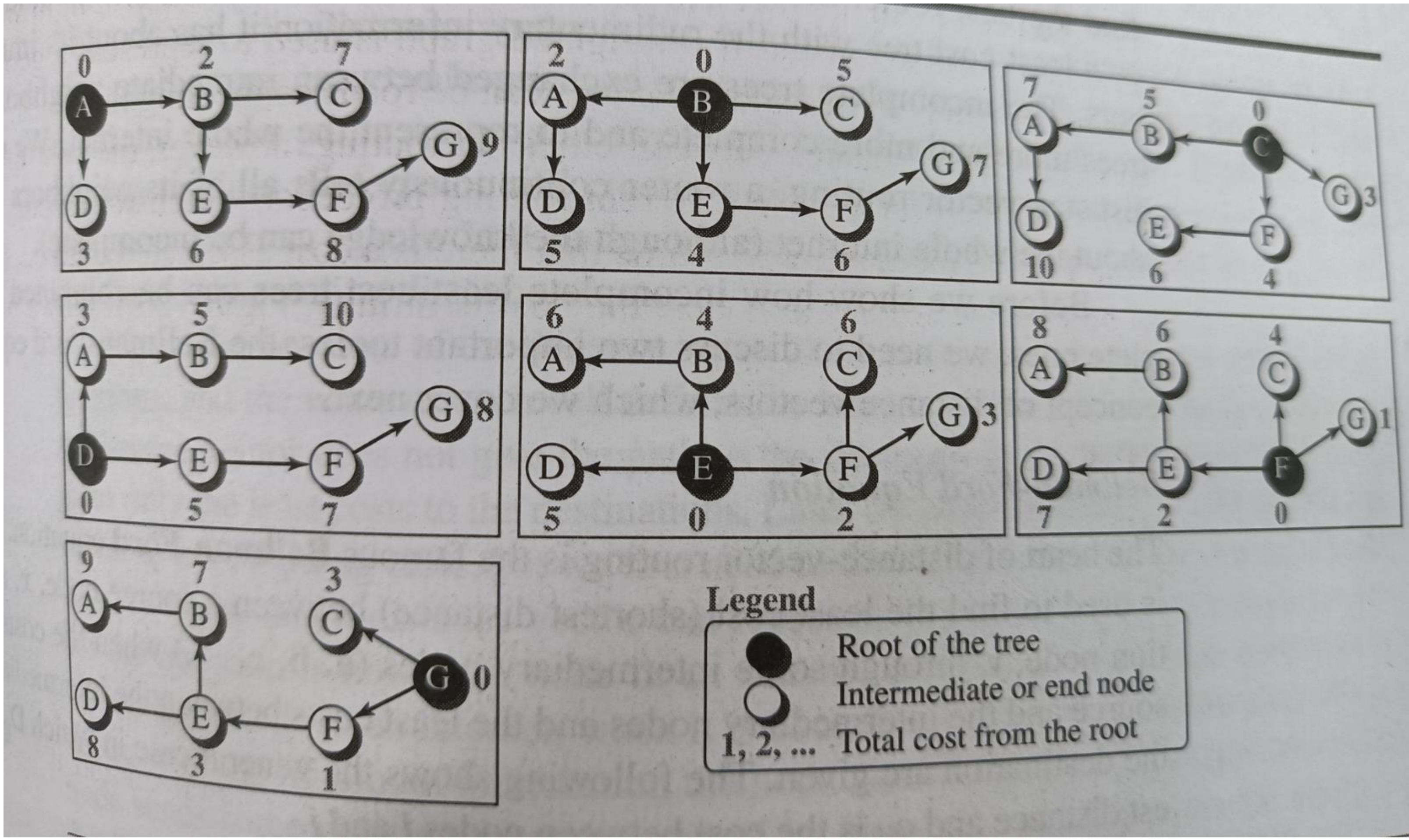
Least cost Routing



a. An internet

b. The weighted graph

Least-Cost Trees



2 INTER- AND INTRA-DOMAIN ROUTING

- **Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.**
- **For this reason, an internet is divided into autonomous systems.**
- **An autonomous system (AS) is a group of networks and routers under the authority of a single administration.**
- **Routing inside an autonomous system is called intra-domain routing.**
- **Routing between autonomous systems is called inter-domain routing**

Figure 1 *Autonomous systems*

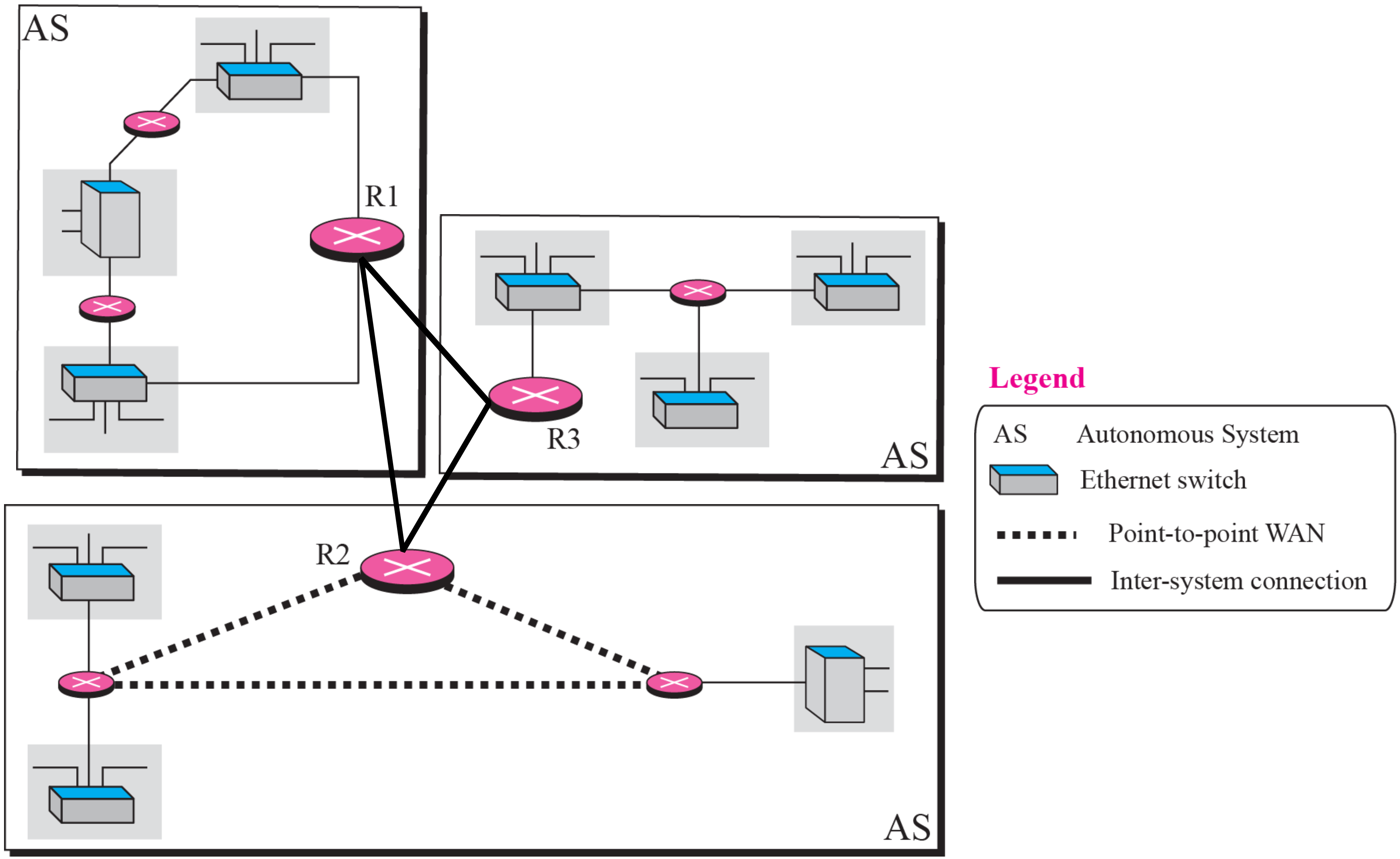
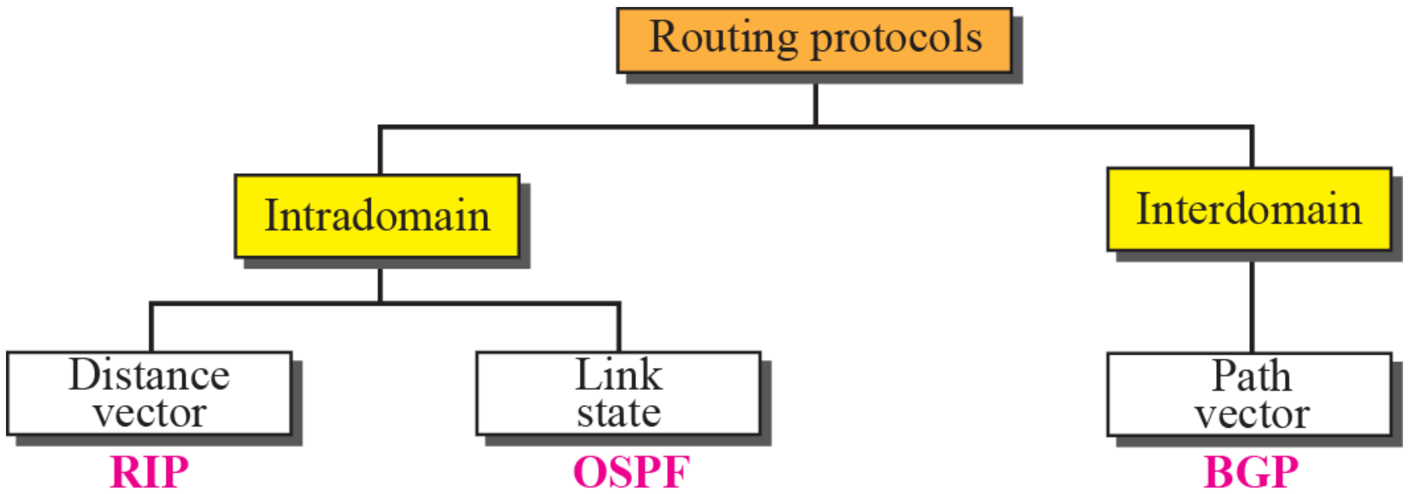


Figure 2 *Popular routing protocols*



3 DISTANCE VECTOR ROUTING

The distance-vector (DV) routing uses to find the best route.

- **DV is simple routing protocol which takes routing decision on the number of hops between source and destination.**
- **A route with less number of hops is considered as best route.**
- **Every router advertises its set best routes to other routers.**
- **Ultimately, all routers build up their network topology based on the advertisements of their peer routers**

  - ✓ **Bellman-Ford Algorithm**
  - ✓ **Distance Vector Routing Algorithm**
  - ✓ **Count to Infinity**

Figure 3 *A graph for Bellman-Ford algorithm*

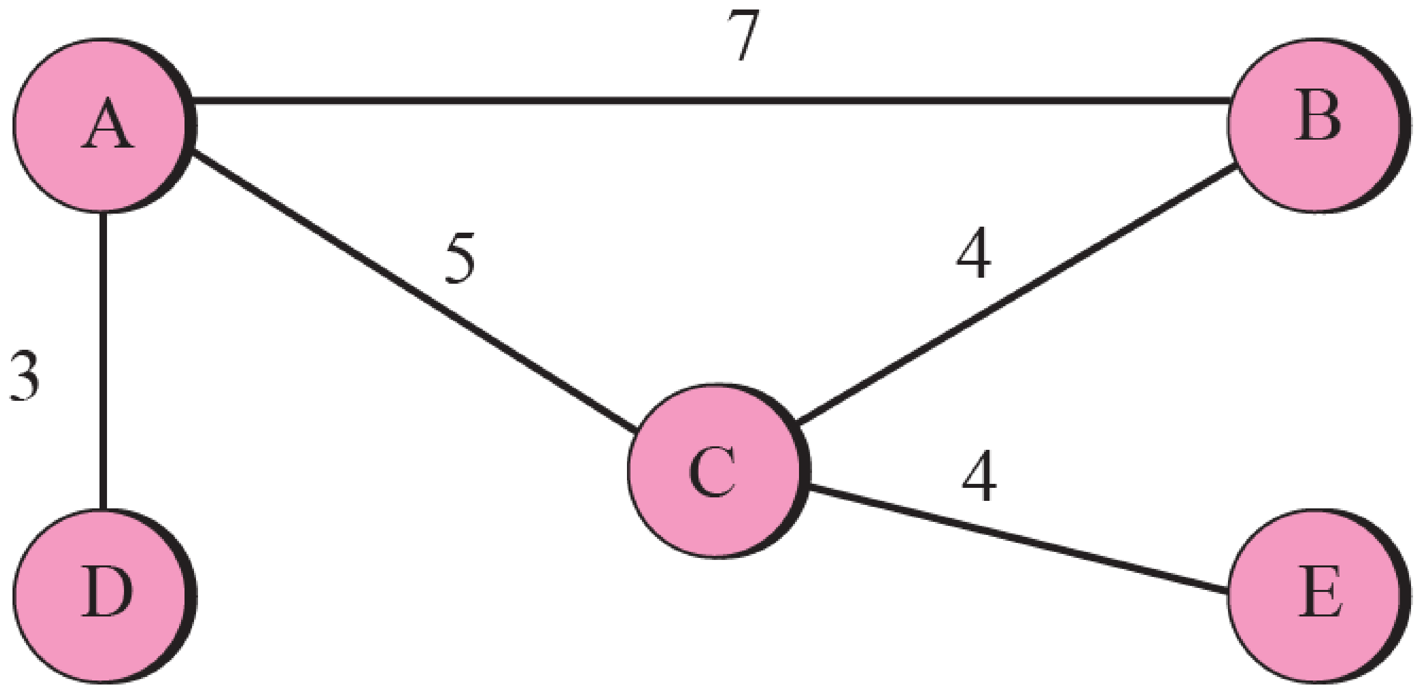
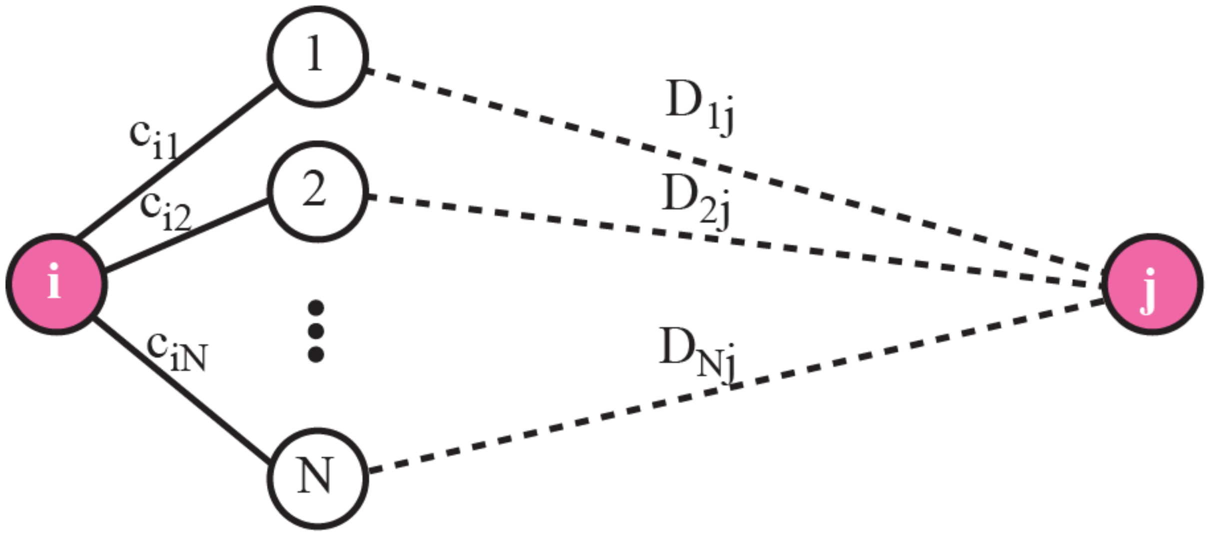


Figure 4 ***The fact behind Bellman-Ford algorithm***

$$D_{ij} = \text{minimum } \{(c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \ldots (c_{iN} + D_{Nj})\}$$



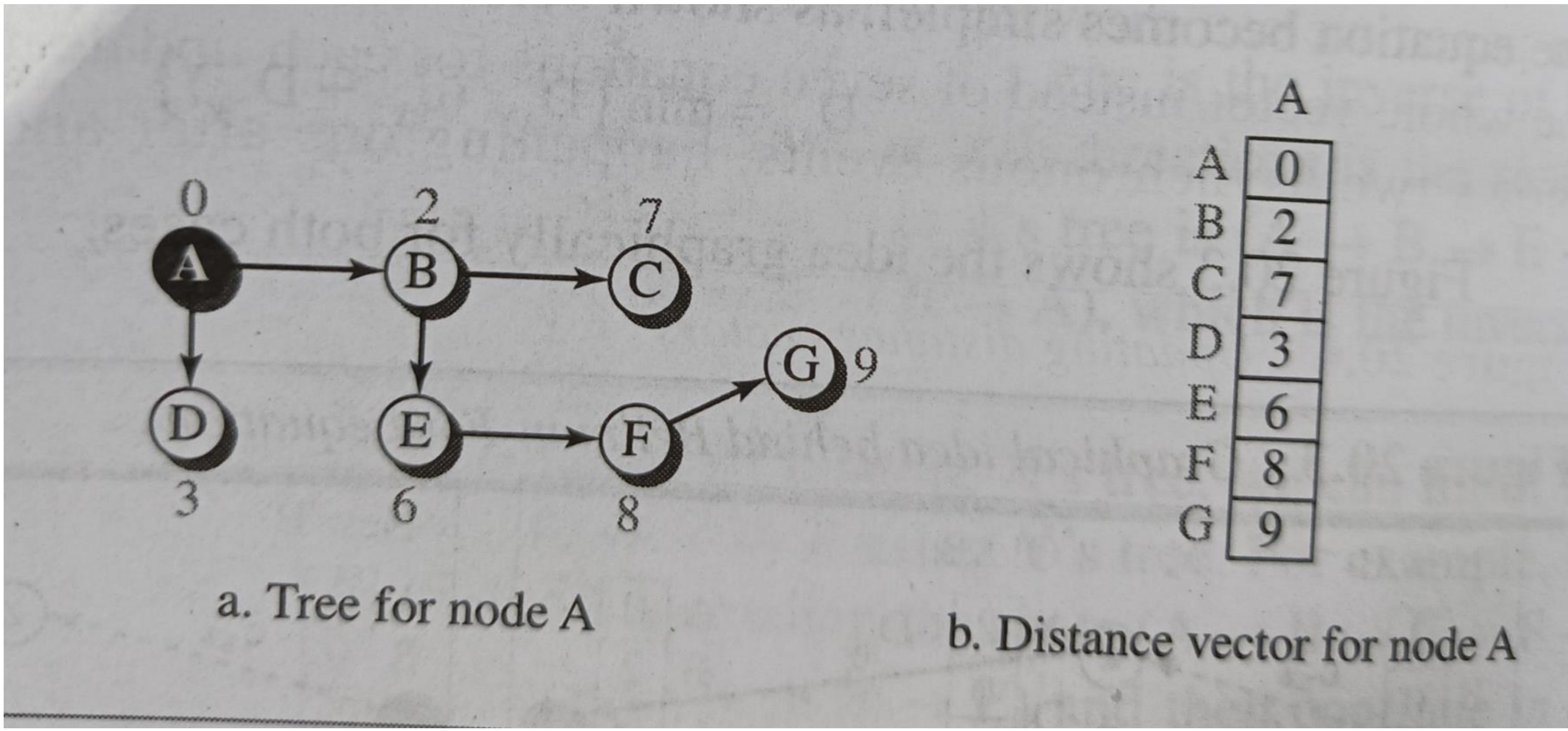
Legend

D_{ij} Shortest distance between i and j
c_{ij} Cost between i and j
N Number of nodes

- DVR, normally we want to update an existing least cost with a least cost through an intermediate node such as z

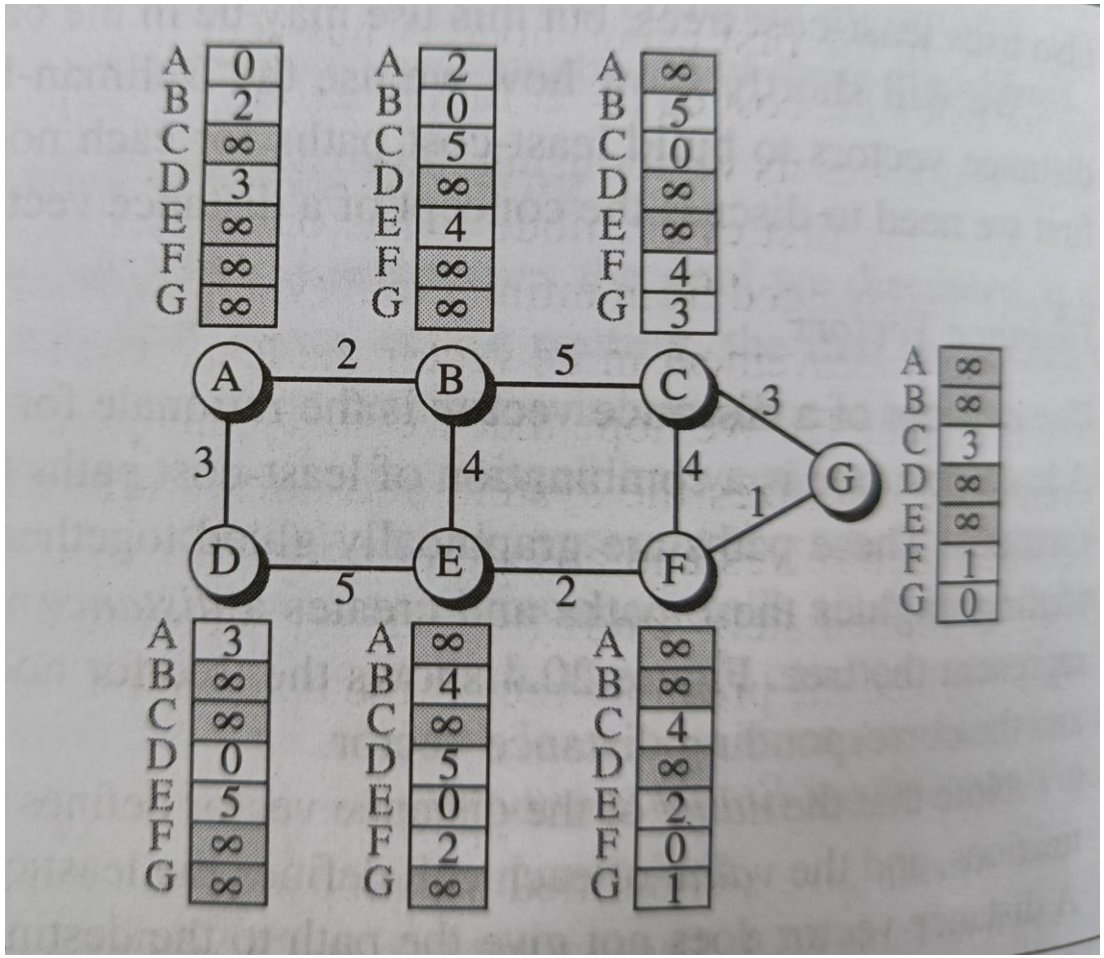
$$D_{xy} = \min \{D_{xy}, (c_{xz} + D_{zy})\}$$

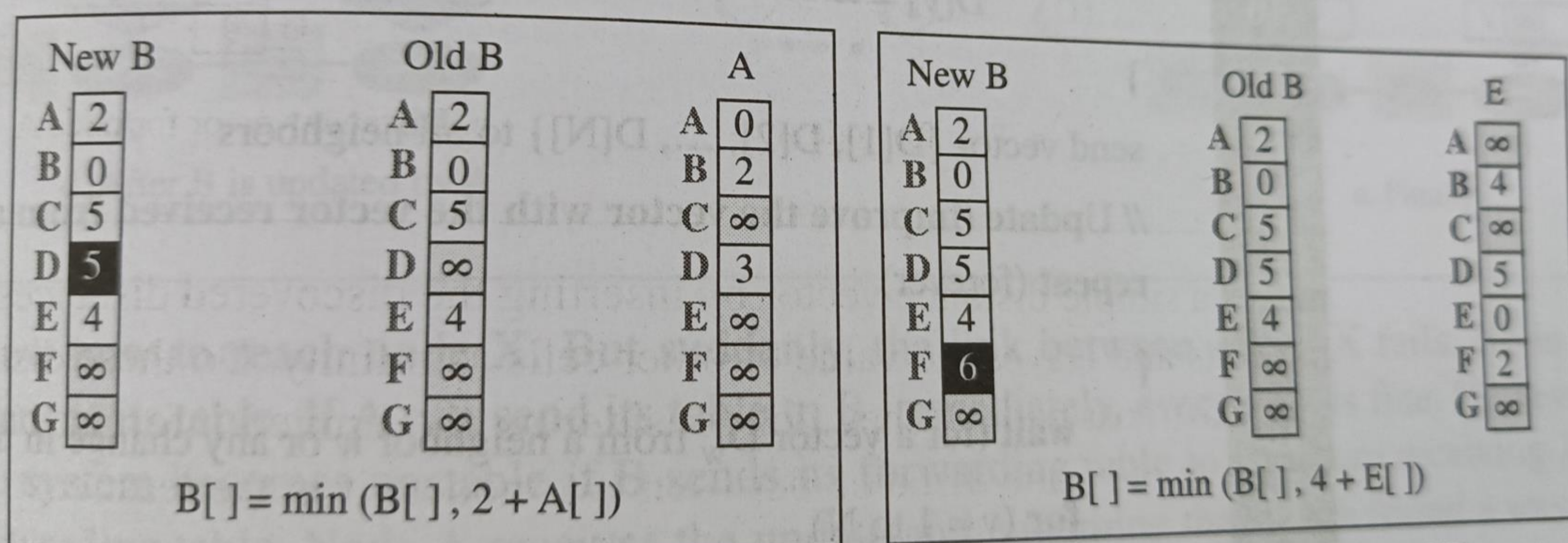
Distance vectors



a. Tree for node A

b. Distance vector for node A


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



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

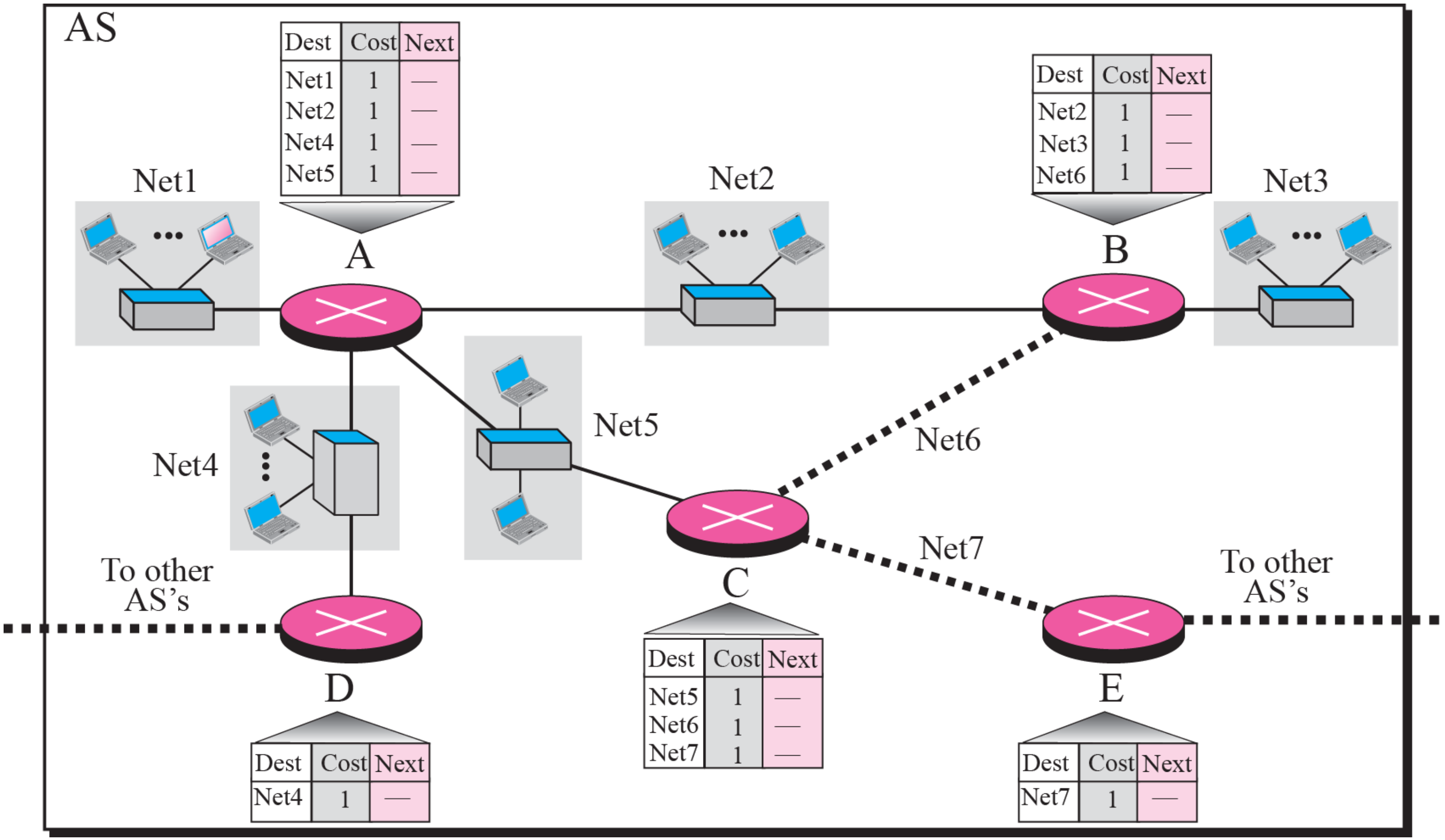
b. Second event: B receives a copy of E's vector.

Note:
X[ ]: the whole vector

Example 1

- **Next figure shows the initial routing table for an AS.**
- **Note that the figure does not mean that all routing tables have been created at the same time; each router creates its own routing table when it is booted.**

Figure 5 *Example 1*



Example 2

- **Now assume router A sends four records to its neighbors, routers B, D, and C.**

- **Next figure 6 shows the changes in B's routing table when it receives these records. We leave the changes in the routing tables of other neighbors as exercise.**

Figure 6 *Example 2*

Dest	Cost	Next
Net1	1	—
Net2	1	—
Net4	1	—
Net5	1	—

Dest	Cost	Next
Net2	1	—
Net3	1	—
Net6	1	—

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

After receiving record 1

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

After receiving record 2

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net6	1	—

After receiving record 3

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—

After receiving record 4

Example 3

Figure 7 shows the final routing tables for routers in Figure 5.

Figure 7 *Example 3*

A

Dest	Cost	Next
Net1	1	—
Net2	1	—
Net3	2	B
Net4	1	—
Net5	1	—
Net6	2	C
Net7	2	C

B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—
Net7	2	C

C

Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	2	B
Net4	2	A
Net5	1	—
Net6	1	—
Net7	1	—

D

Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	3	A
Net4	1	—
Net5	1	A
Net6	3	A
Net7	3	A

E

Dest	Cost	Next
Net1	3	C
Net2	3	C
Net3	3	C
Net4	3	C
Net5	2	C
Net6	2	C
Net7	1	—

Figure 8 *Two-node instability*

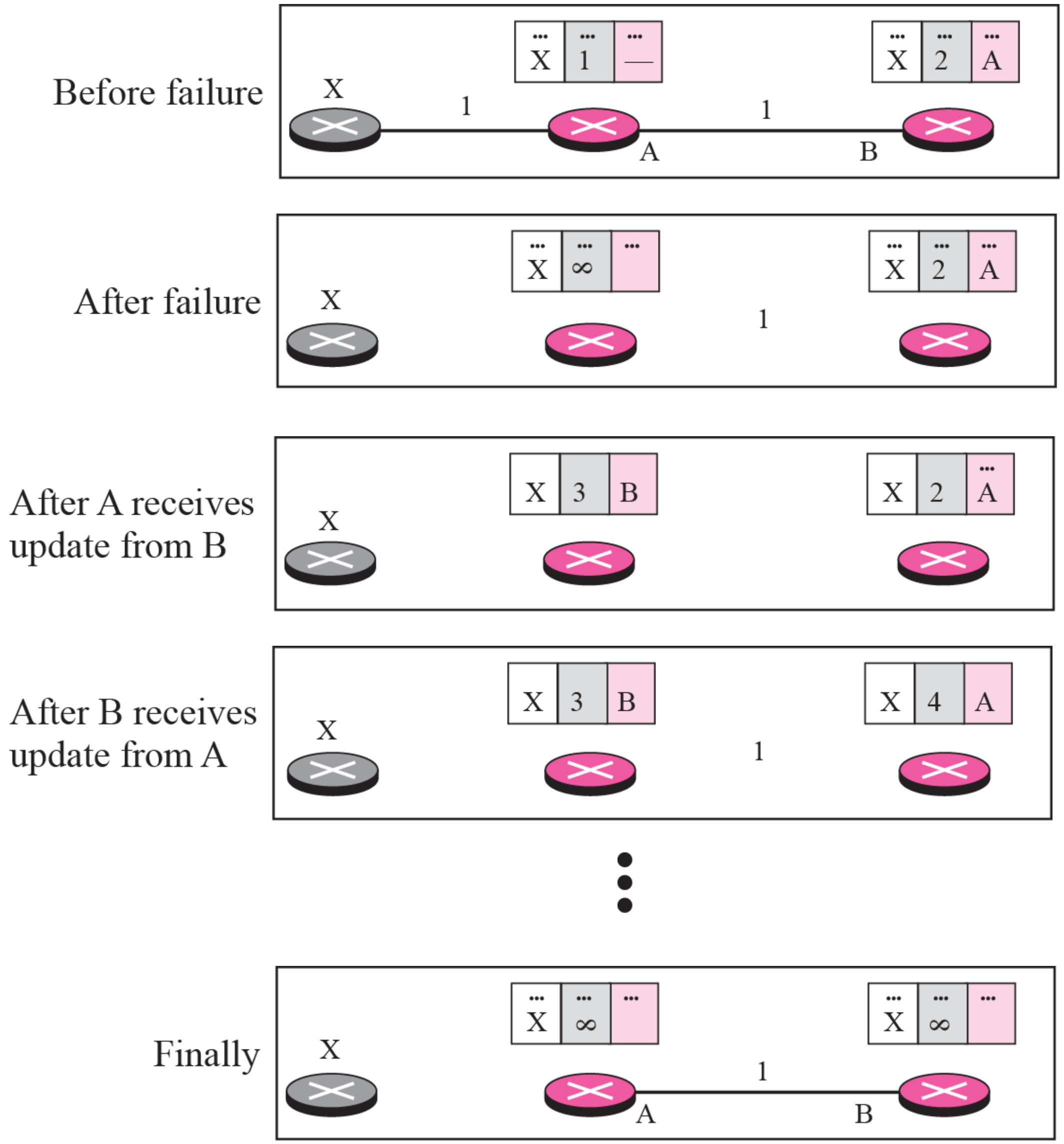
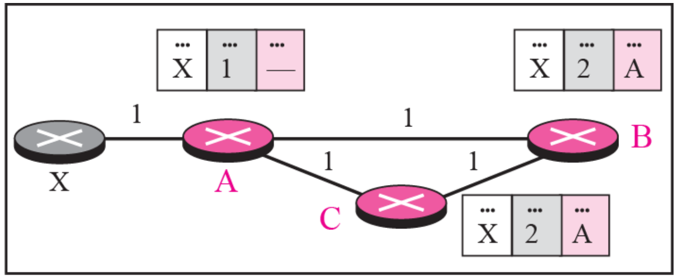
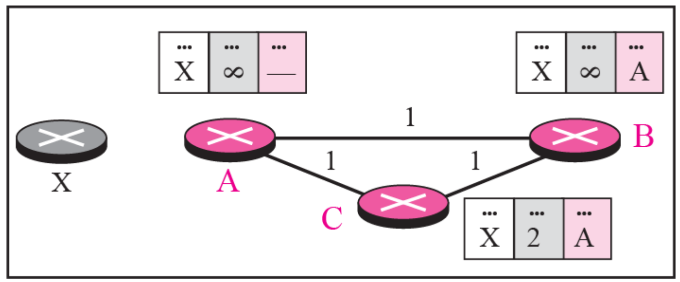


Figure 9 *Three-node instability*

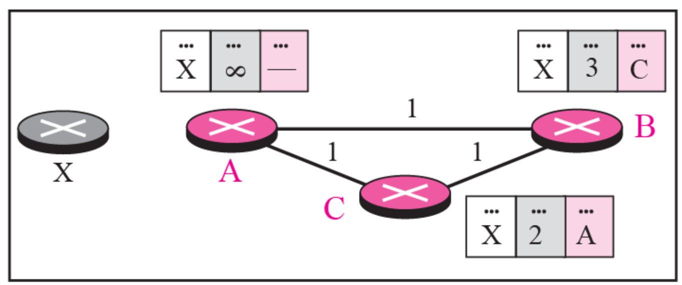
Before failure



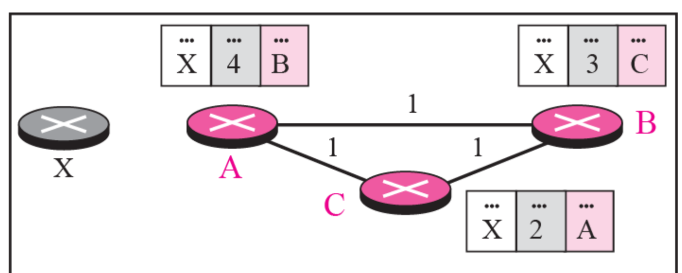
After A sends the route to B and C, but the packet to C is lost



After C sends the route to B



After B sends the route to A

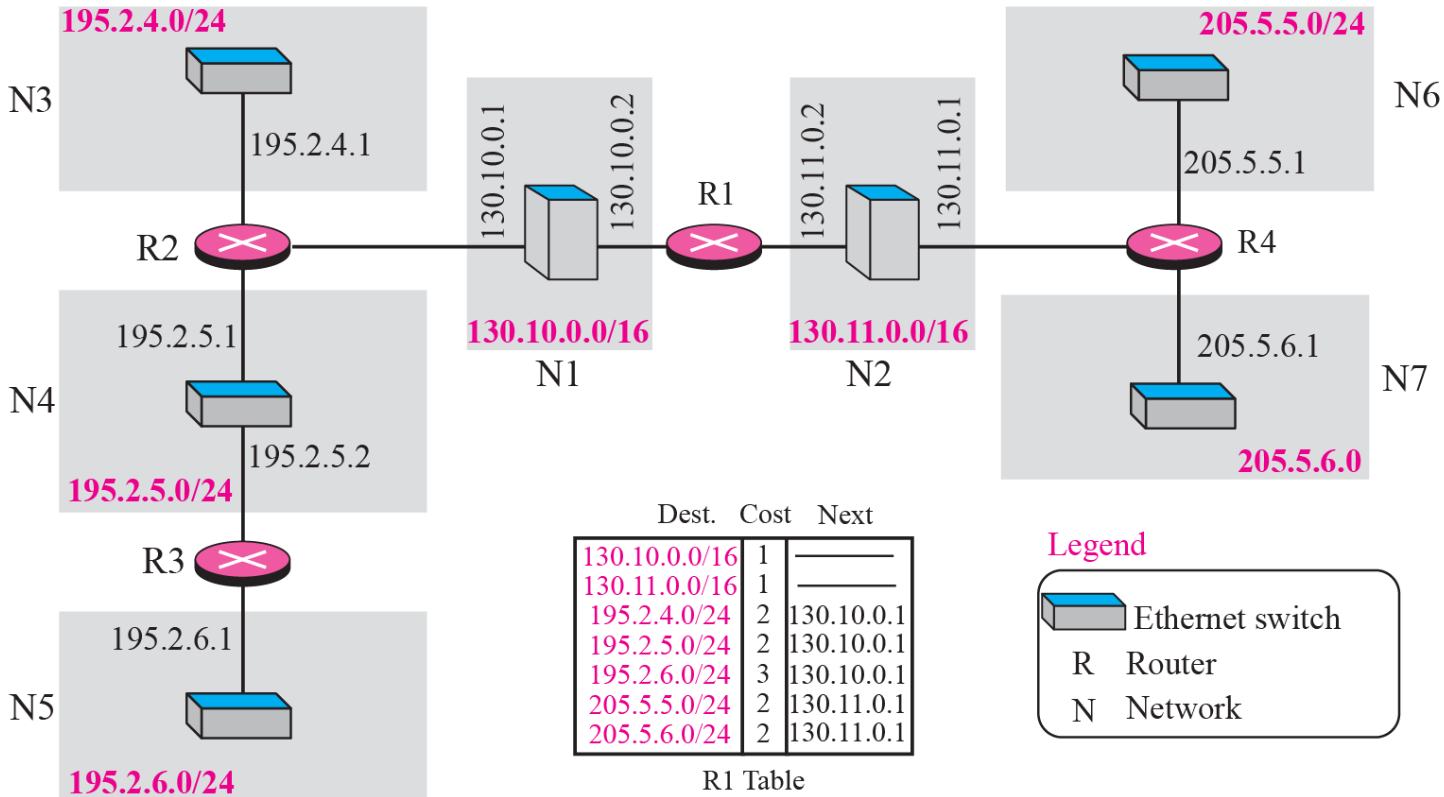


4 RIP

- **The Routing Information Protocol (RIP) is an intra-domain (interior) routing protocol used inside an autonomous system.**
- **It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations.**

  - ✓ **RIP Message Format**
  - ✓ **Request and Response**
  - ✓ **Timers in RIP**
  - ✓ **RIP Version 2**
  - ✓ **Encapsulation**

Figure 10 *Example of a domain using RIP*



Dest.	Cost	Next
130.10.0.0/16	1	—
130.11.0.0/16	1	—
195.2.4.0/24	2	130.10.0.1
195.2.5.0/24	2	130.10.0.1
195.2.6.0/24	3	130.10.0.1
205.5.5.0/24	2	130.11.0.1
205.5.6.0/24	2	130.11.0.1

R1 Table

Dest.	Cost	Next
130.10.0.0/16	1	—
130.11.0.0/16	2	130.10.0.2
195.2.4.0/24	1	—
195.2.5.0/24	1	—
195.2.6.0/24	2	195.2.5.2
205.5.5.0/24	3	130.10.0.2
205.5.6.0/24	3	130.10.0.2

R2 Table

Dest.	Cost	Next
130.10.0.0/16	2	195.2.5.1
130.11.0.0/16	3	195.2.5.1
195.2.4.0/24	2	195.2.5.1
195.2.5.0/24	1	—
195.2.6.0/24	1	—
205.5.5.0/24	4	195.2.5.1
205.5.6.0/24	4	195.2.5.1

R3 Table

Dest.	Cost	Next
130.10.0.0/16	2	130.11.0.2
130.11.0.0/16	1	—
195.2.4.0/24	3	130.11.0.2
195.2.5.0/24	3	130.11.0.2
195.2.6.0/24	4	130.11.0.2
205.5.5.0/24	1	—
205.5.6.0/24	1	—

R4 Table

Figure 11 *RIP message format*

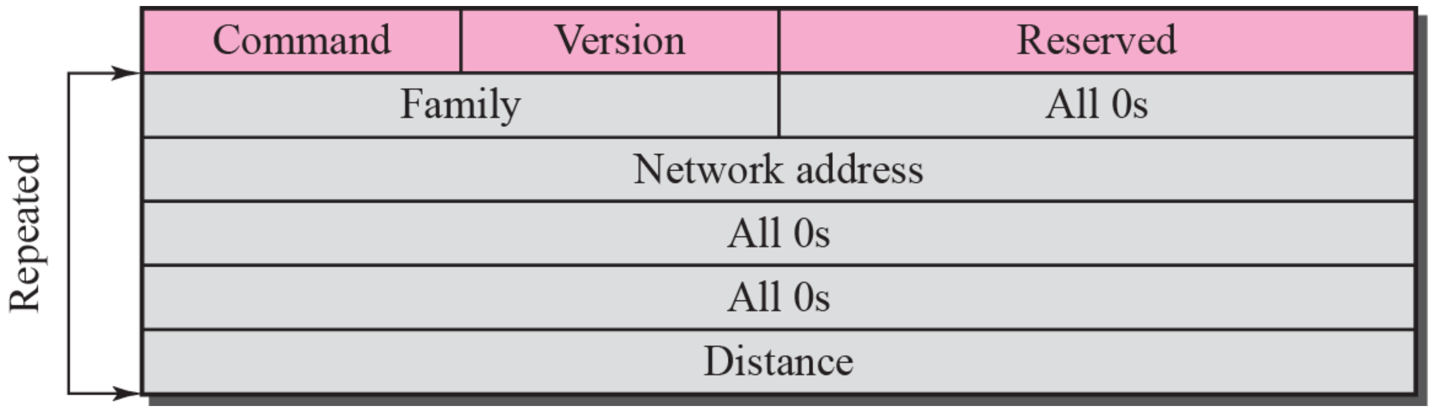


Figure 12 *Request messages*

Repeated

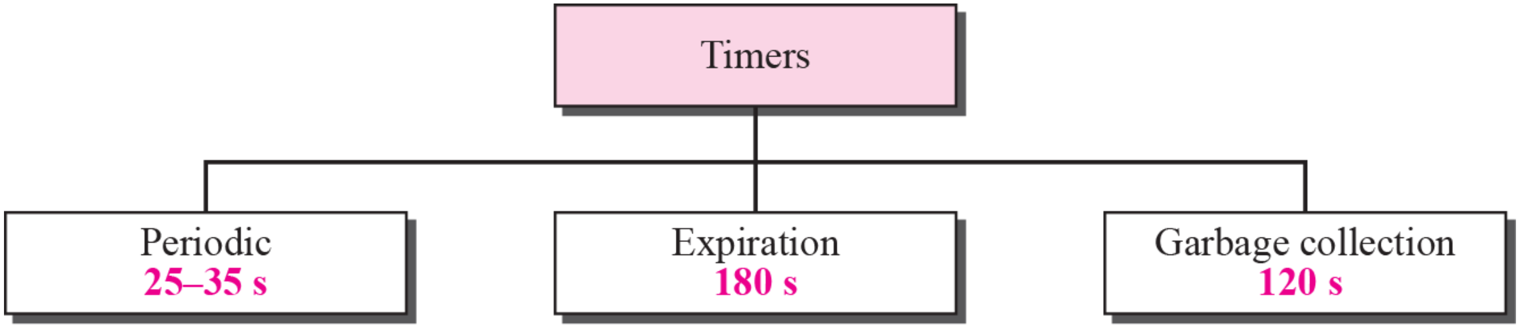
Com: 1	Version	Reserved
Family		All 0s
Network address		
All 0s		
All 0s		
All 0s		

a. Request for some

Com: 1	Version	Reserved
Family		All 0s
All 0s		
All 0s		
All 0s		
All 0s		

b. Request for all

Figure 14 *RIP timers*



Note

RIP uses the services of UDP on well-known port 520.

5 LINK STATE ROUTING

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—the list of nodes and links, how they are connected including the type, cost (metric), and the condition of the links (up or down)—the node can use the Dijkstra algorithm to build a routing table.

Figure 17 *Concept of Link state routing*

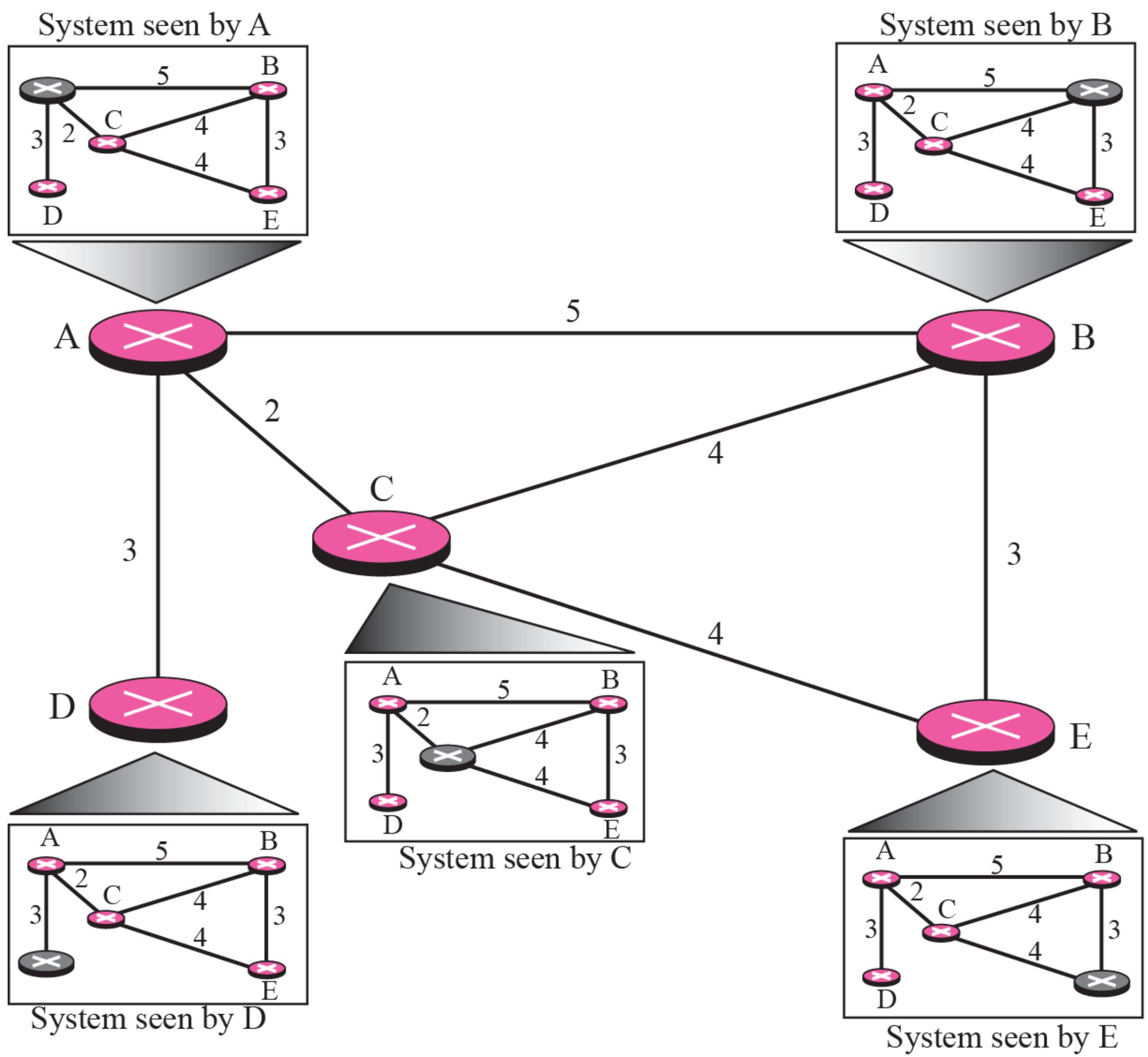
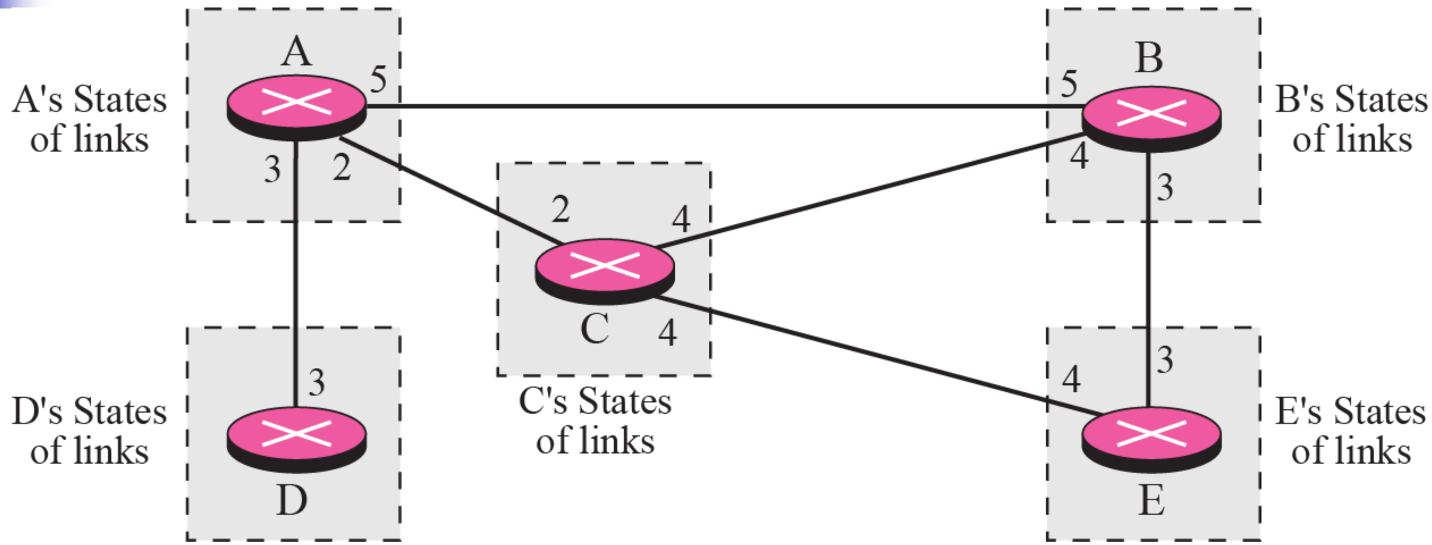


Figure 18 *Link state knowledge*

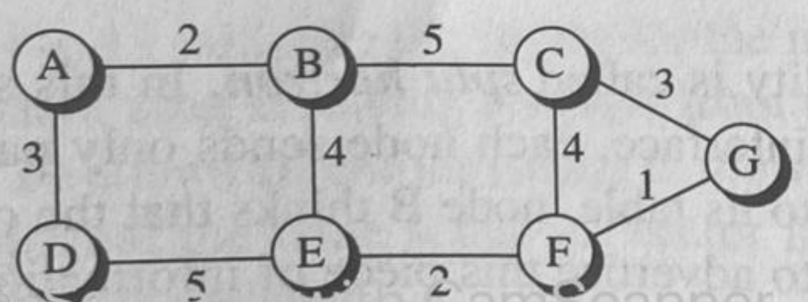


Building Routing Table:

1) Creation of link state packets (LSP)
2) Flooding of LSPs
3) Formation of shortest path tree
4) Calculation of routing table from shortest path tree

Creation of link state packets

- A link state packet can carry a large amount of information.
- For the moment, however we assume that it carries a minimum amount of data: node identity, cost of links, a sequence no and age.
- The first two are need to form the topology.
- LSP are generated on two occasions:
- When there is a change in the topology of the domain
- On a periodic basis



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

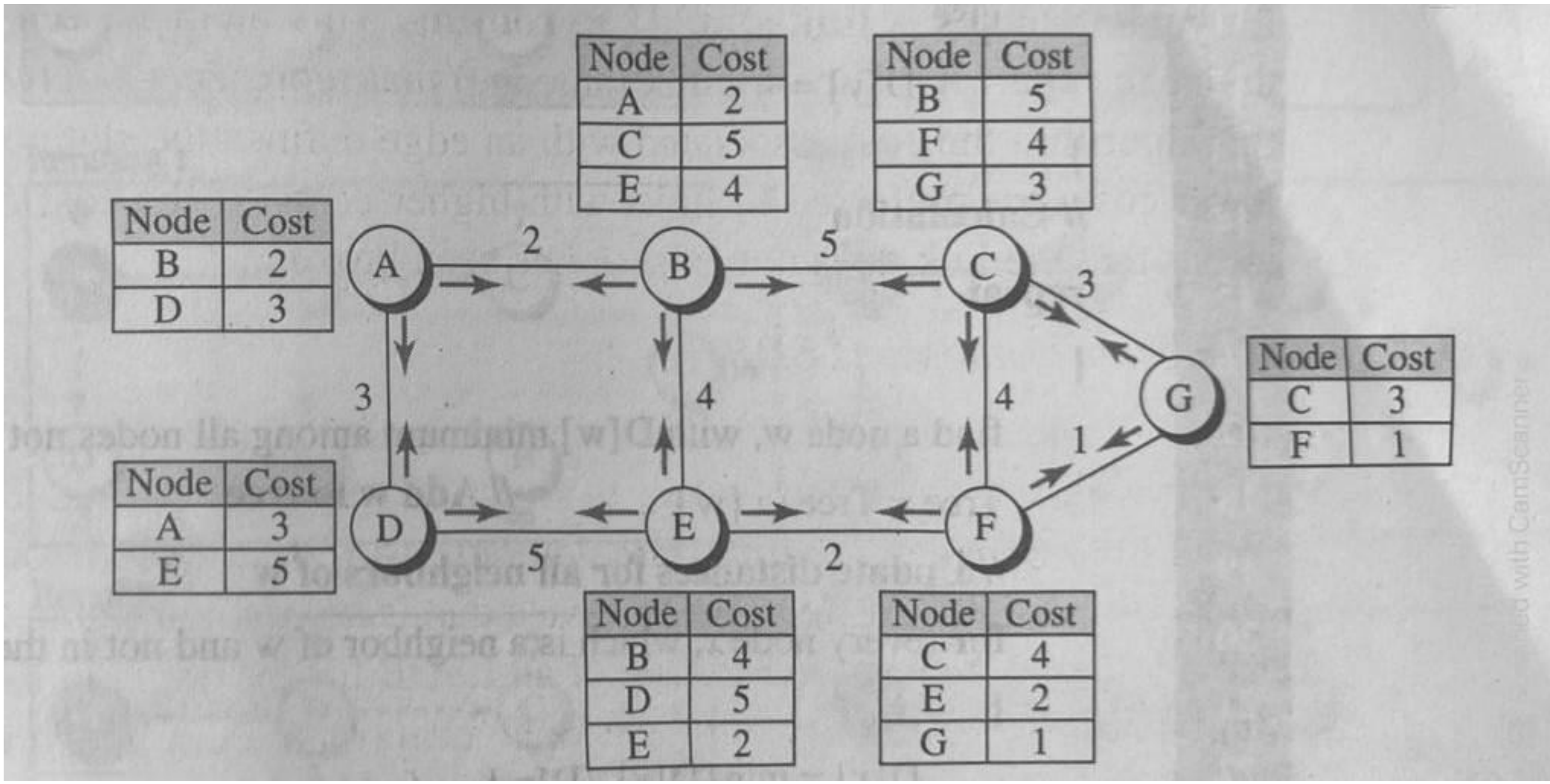
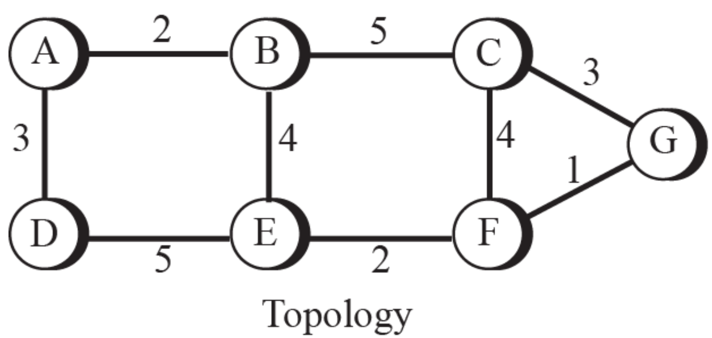
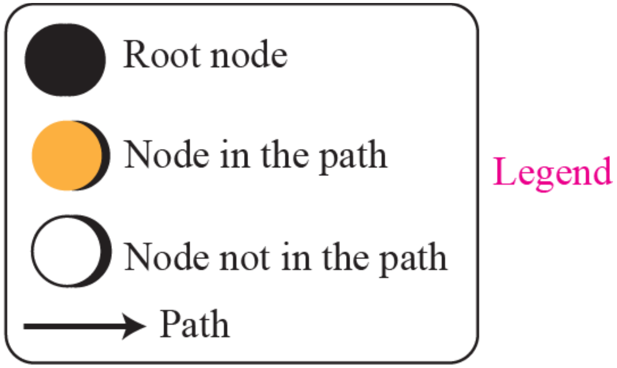

Node Cost
A 2
C 5
E 4
Node Cost
B 5
F 4
G 3
Node Cost
B 2
D 3
A
B
C
G
D
E
F
2
5
3
3
4
4
1
5
2
Node Cost
C 3
F 1
Node Cost
A 3
E 5
Node Cost
B 4
D 5
E 2
Node Cost
C 4
E 2
G 1

Figure 19 *Forming shortest path three for router A in a graph*





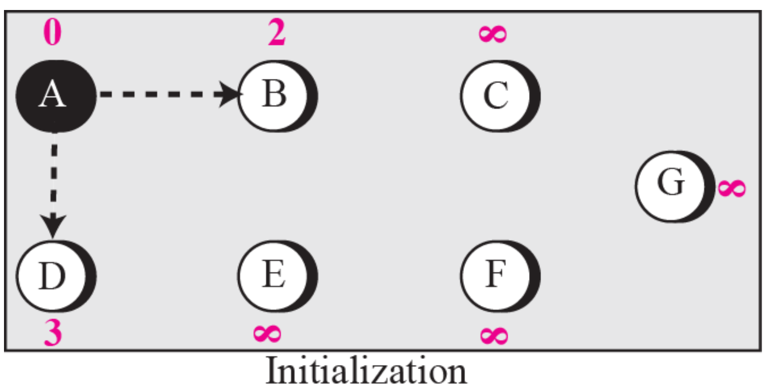
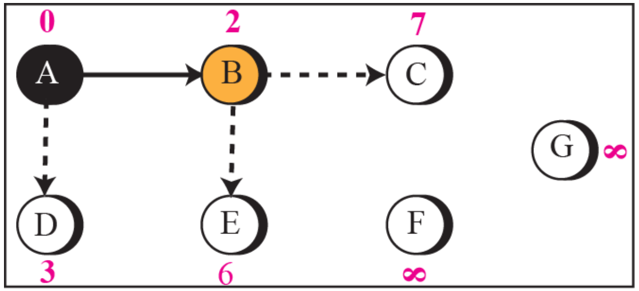
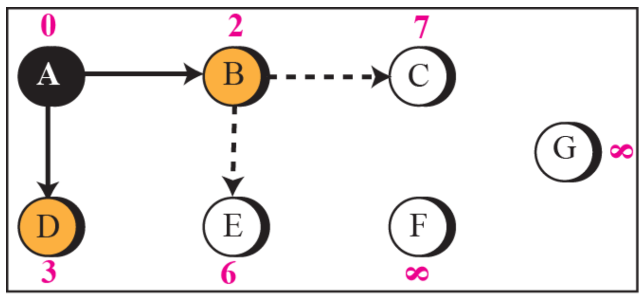


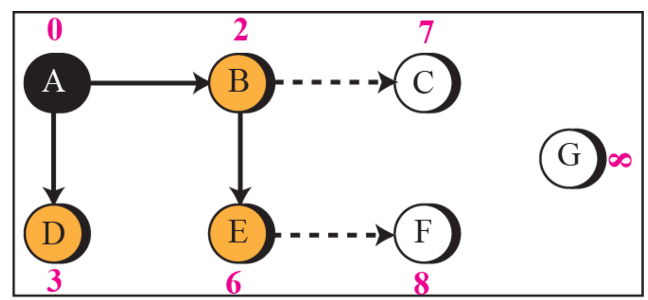
Figure 19 *Continued*



Iteration 1

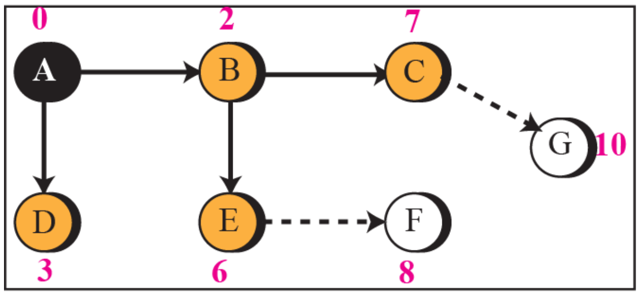


Iteration 2

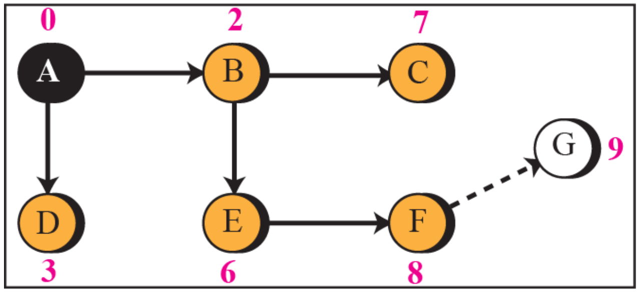


Iteration 3

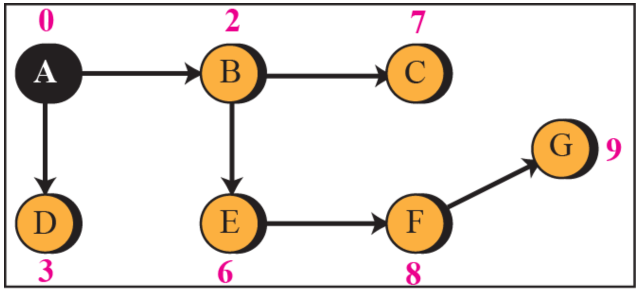
Figure 19 *Continued*



Iteration 4



Iteration 5



Iteration 6

Example 6

To show that the shortest path tree for each node is different, we found the shortest path tree as seen by node C (Figure 20). We leave the detail as an exercise.

Figure 20 *Example 6*

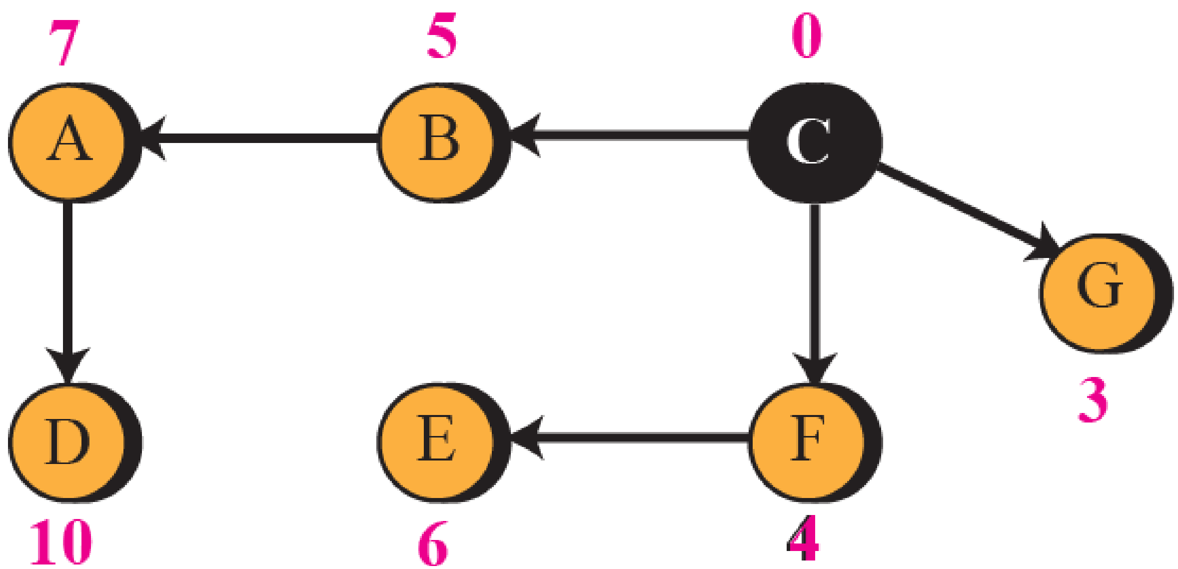


Table 11.4 *Routing Table for Node A*

Destination	*Cost*	*Next Router*
A	0	—
B	2	—
C	7	B
D	3	—
E	6	B
F	8	B
G	9	B

6 OSPF

The Open Shortest Path First (OSPF) protocol is an intra-domain routing protocol based on link state routing. Its domain is also an autonomous system.

- ✓ **Area**
- ✓ **Metric**
- ✓ **Types of Links**
- ✓ **Graphical Representation**
- ✓ **OSPF Packets**
- ✓ **Link State Update Packet**
- ✓ **Other Packets**
- ✓ **Encapsulation**

Figure 21 *Areas in an autonomous system*

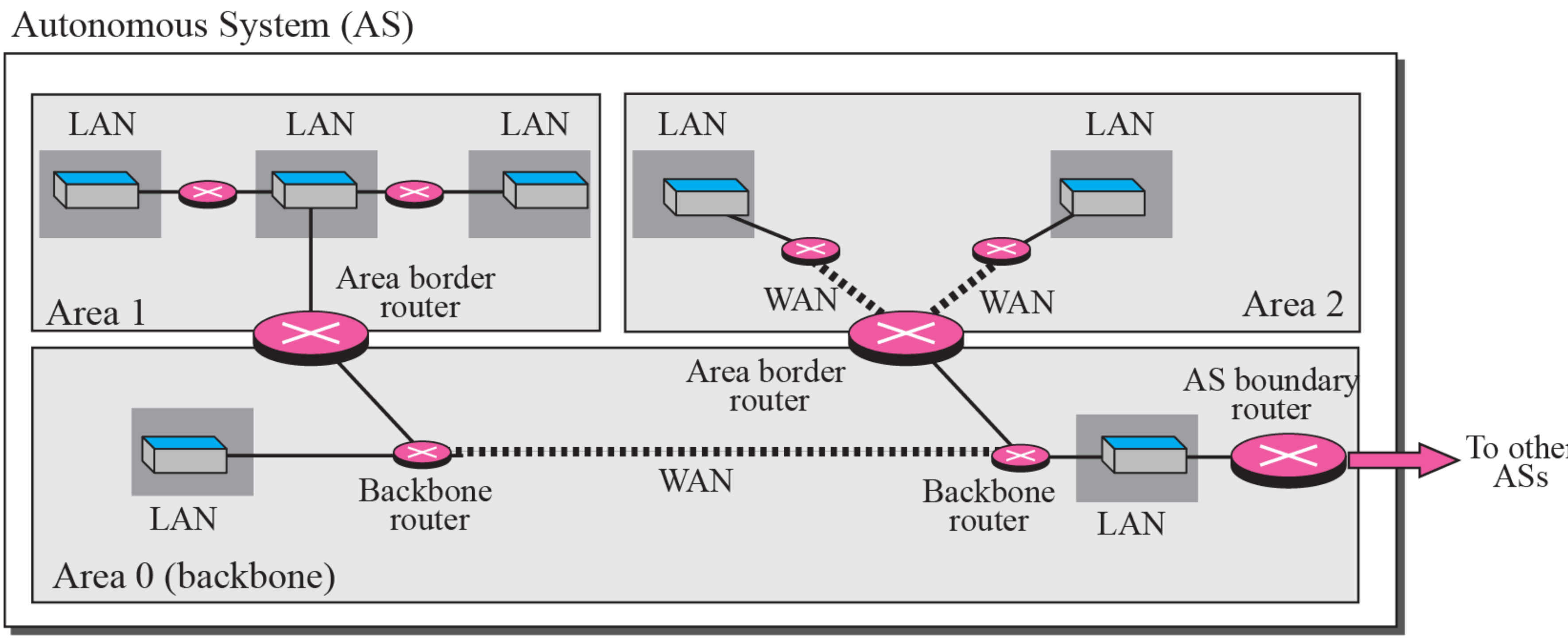


Figure 22 *Types of links*

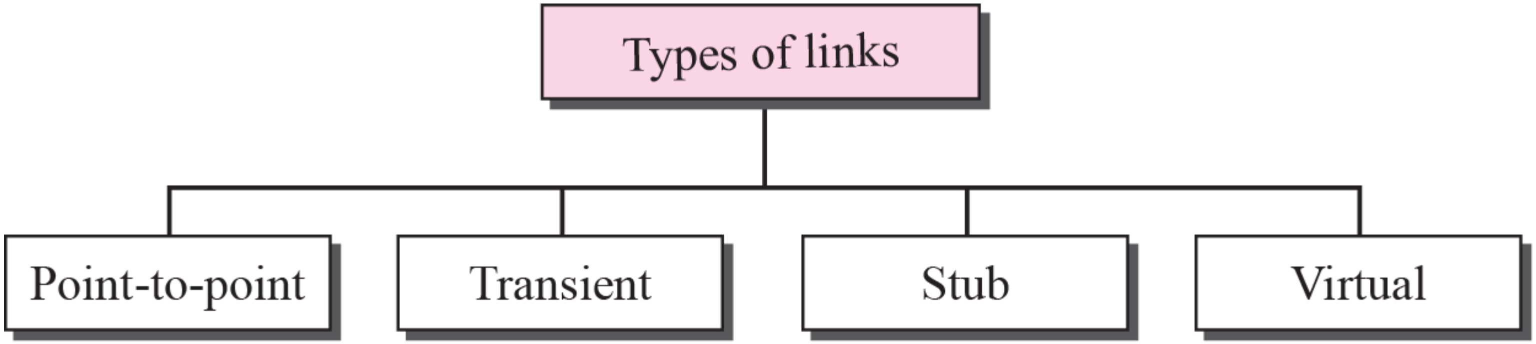
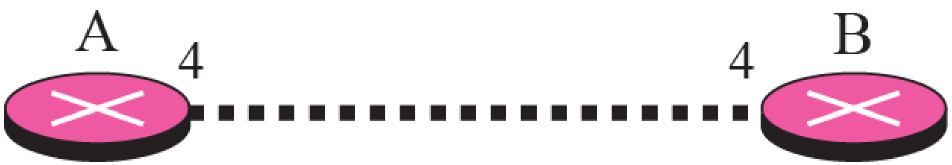


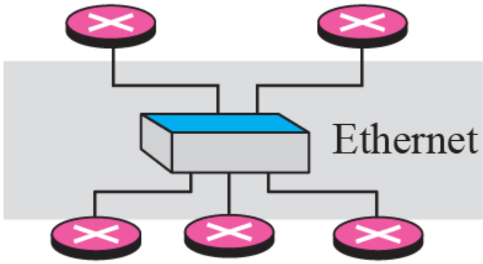
Figure 23 *Point-to-point link*



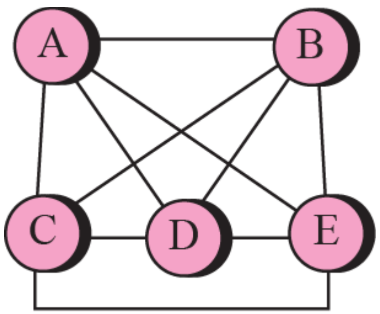
a. Point-to-point network

b. Representation

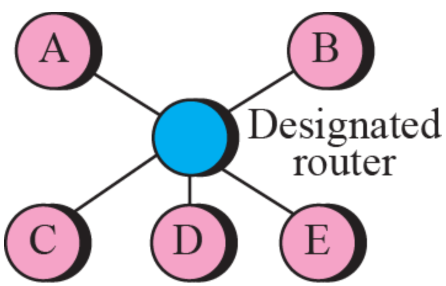
Figure 24 *Transient link*



a. Transient network



b. Unrealistic



c. Realistic

Figure 25 ***Stub link***

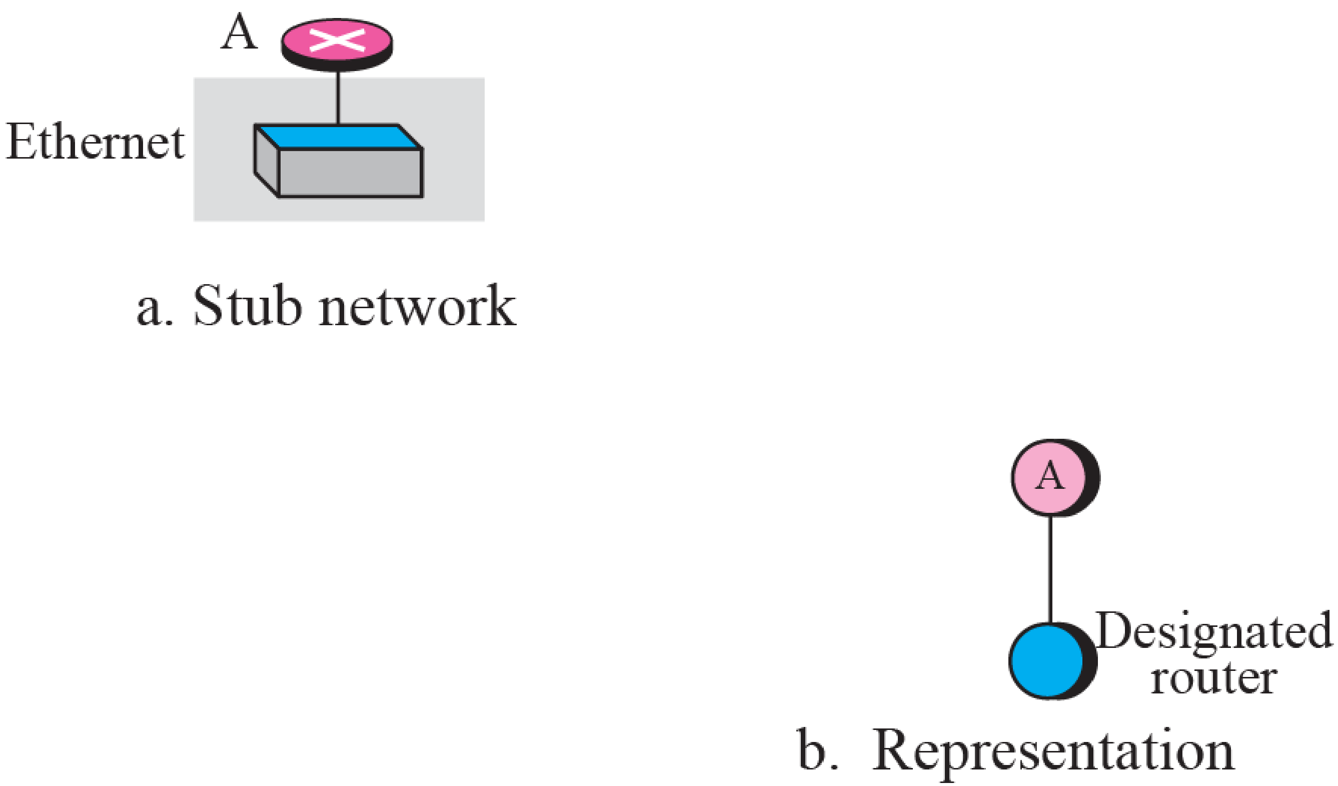
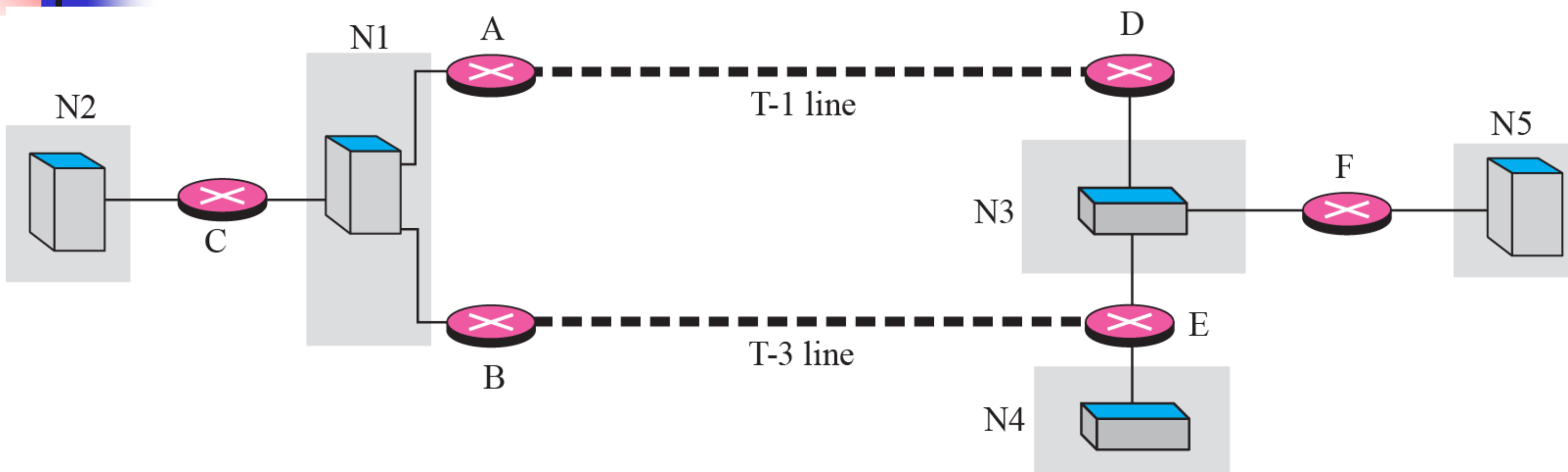
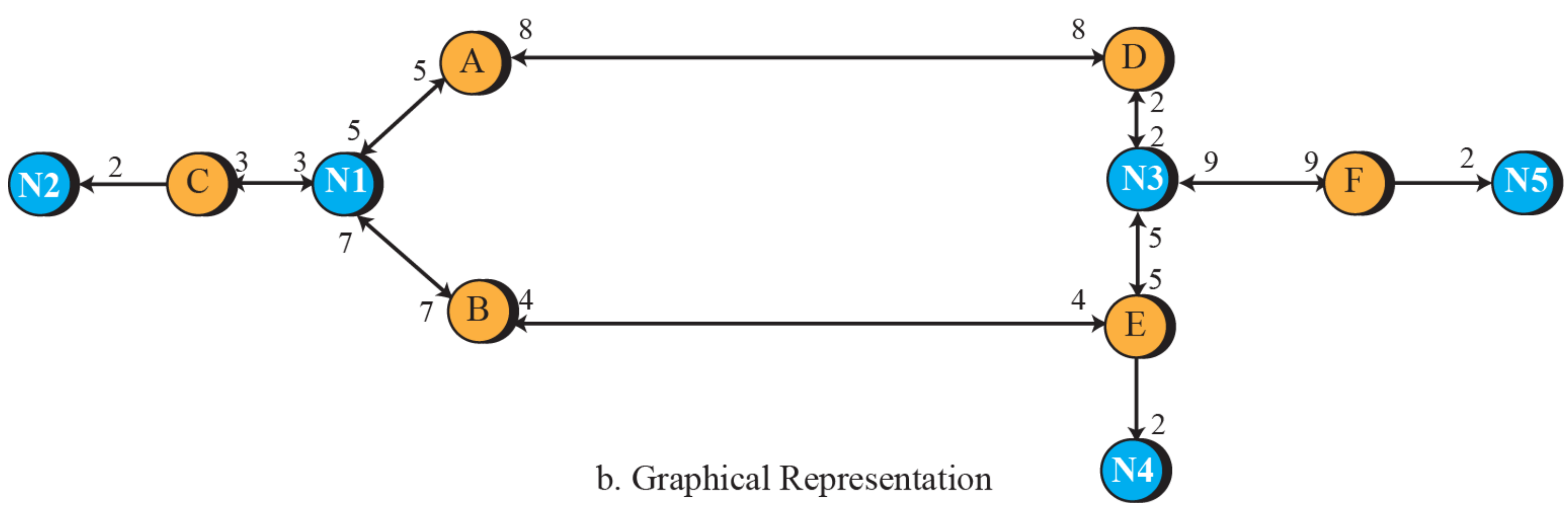


Figure 26 *Example of an AS and its graphical representation in OSPF*

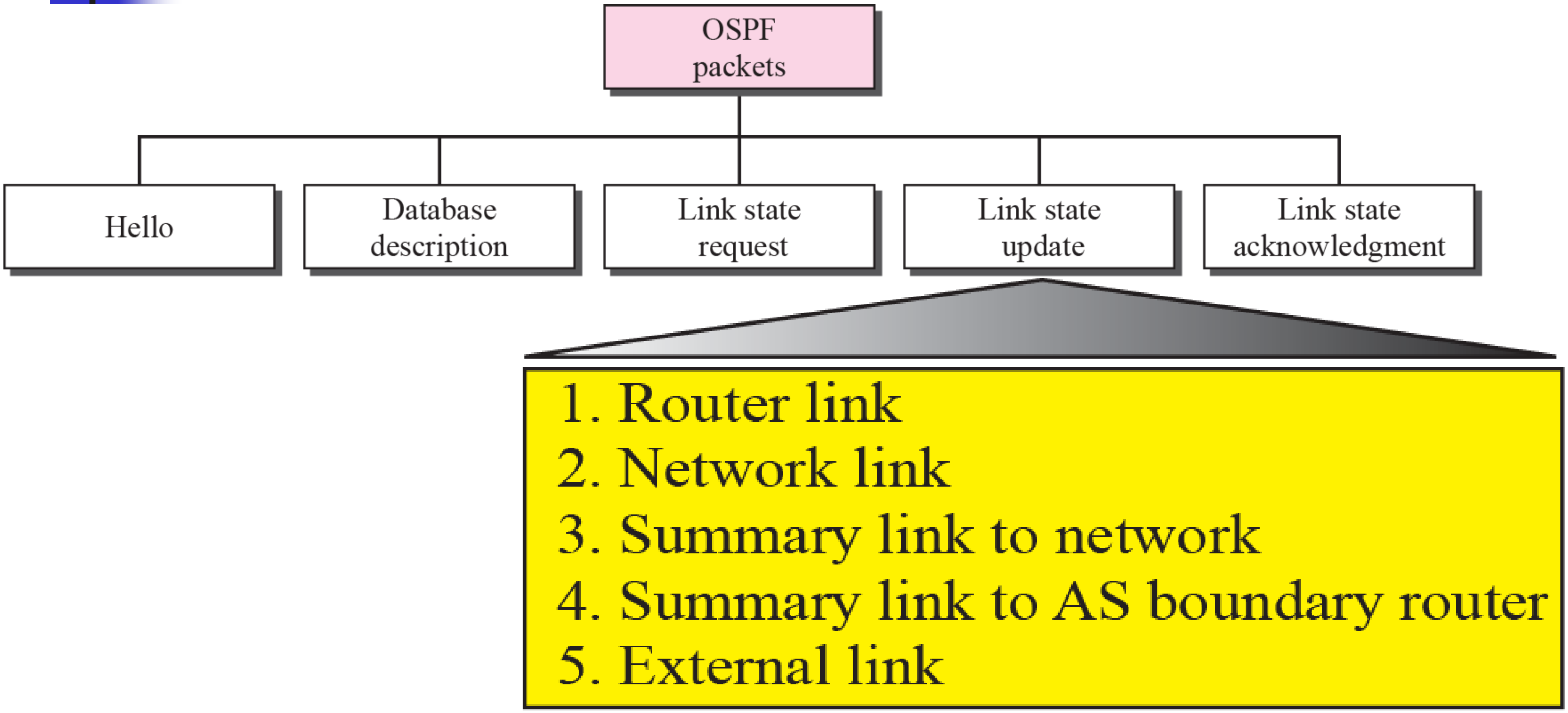


a. Autonomous System



b. Graphical Representation

Figure 27 *Types of OSPF packet*



Note

OSPF packets are encapsulated in IP datagrams.

7 PATH VECTOR ROUTING

Distance vector and link state routing are both **interior routing protocols**.

They can be used inside an autonomous system. Both of these routing protocols become intractable when the domain of operation becomes large.

Distance vector routing is subject to instability if there is more than a few hops in the domain of operation.

Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding.

There is a need for a third routing protocol which we call path vector routing.

PATH VECTOR ROUTING

Routing takes place between the two autonomous networks.

DV and Link State Routing ignore the internet outside the autonomous system whereas Path Vector routing assumes that internet consists of a collection of interconnected autonomous systems.

Protocol for inter-domain routing are also called as **exterior gateway protocols.**

- ✓ **Reachability**
- ✓ **Routing Table**

Example 10

The difference between the distance vector routing and path vector routing can be compared to the difference between a national map and an international map.

A national map can tell us the road to each city and the distance to be traveled if we choose a particular route;

An international map can tell us which cities exist in each country and which countries should be passed before reaching that city.

Figure 50 *Reachability*

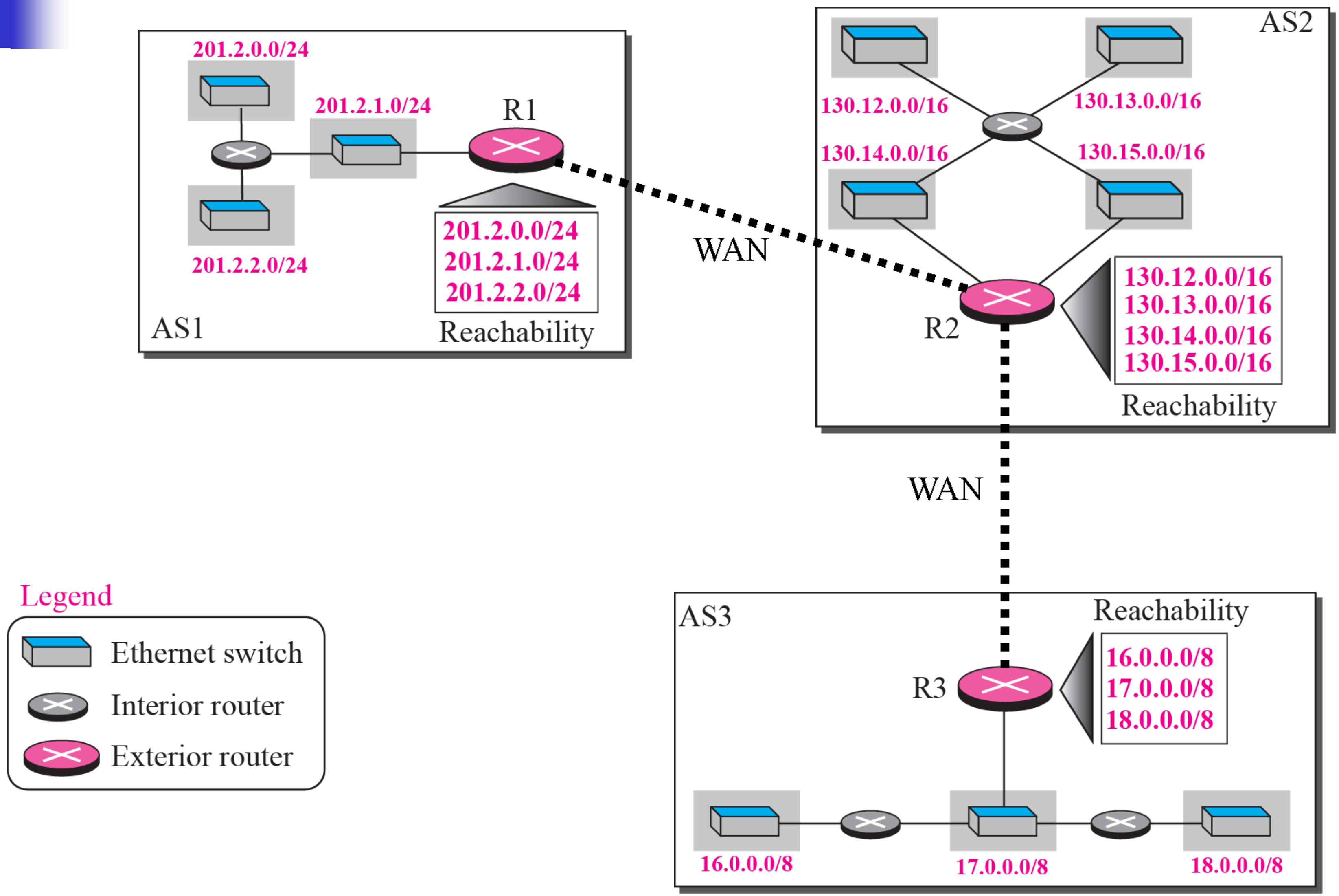


Figure 51 *Stabilized table for three autonomous system*

R1

Network	Path
201.2.0.0/24	**AS1** (This AS)
201.2.1.0/24	**AS1** (This AS)
201.2.2.0/24	**AS1** (This AS)
130.12.0.0/16	AS1, AS2
130.13.0.0/16	AS1, AS2
130.14.0.0/16	AS1, AS2
130.15.0.0/16	AS1, AS2
16.0.0.0/8	AS1, AS2, AS3
17.0.0.0/8	AS1, AS2, AS3
18.0.0.0/8	AS1, AS2, AS3

Path-Vector Routing Table

R2

Network	Path
201.2.0.0/24	AS2, AS1
201.2.1.0/24	AS2, AS1
201.2.2.0/24	AS2, AS1
130.12.0.0/16	**AS2** (This AS)
130.13.0.0/16	**AS2** (This AS)
130.14.0.0/16	**AS2** (This AS)
130.15.0.0/16	**AS2** (This AS)
16.0.0.0/8	AS2, AS3
17.0.0.0/8	AS2, AS3
18.0.0.0/8	AS2, AS3

Path-Vector Routing Table

R3

Network	Path
201.2.0.0/24	AS3, AS2, AS1
201.2.1.0/24	AS3, AS2, AS1
201.2.2.0/24	AS3, AS2, AS1
130.12.0.0/16	AS3, AS2
130.13.0.0/16	AS3, AS2
130.14.0.0/16	AS3, AS2
130.15.0.0/16	AS3, AS2
16.0.0.0/8	**AS3** (This AS)
17.0.0.0/8	**AS3** (This AS)
18.0.0.0/8	**AS3** (This AS)

Path-Vector Routing Table

Figure 52 *Routing tables after aggregation*

R1

Network	Path
201.2.0.0/22	**AS1** (This AS)
130.12.0.0/18	AS1, AS2
16.0.0.0/6	AS1, AS2, AS3

Path-Vector Routing Table

R2

Network	Path
201.2.0.0/22	AS2, AS1
130.12.0.0/18	**AS2** (This AS)
16.0.0.0/6	AS2, AS3

Path-Vector Routing Table

R3

Network	Path
201.2.0.0/22	AS3, AS2, AS1
130.12.0.0/18	AS3, AS2
16.0.0.0/6	**AS3** (This AS)

Path-Vector Routing Table

8 BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.

- ✓ **Types of Autonomous Systems**
- ✓ **Path Attributes**
- ✓ **BGP Sessions**
- ✓ **External and Internal BGP**
- ✓ **Types of Packets**
- ✓ **Packet Format**
- ✓ **Encapsulation**

Figure 53 *Internal and external BGP sessions*

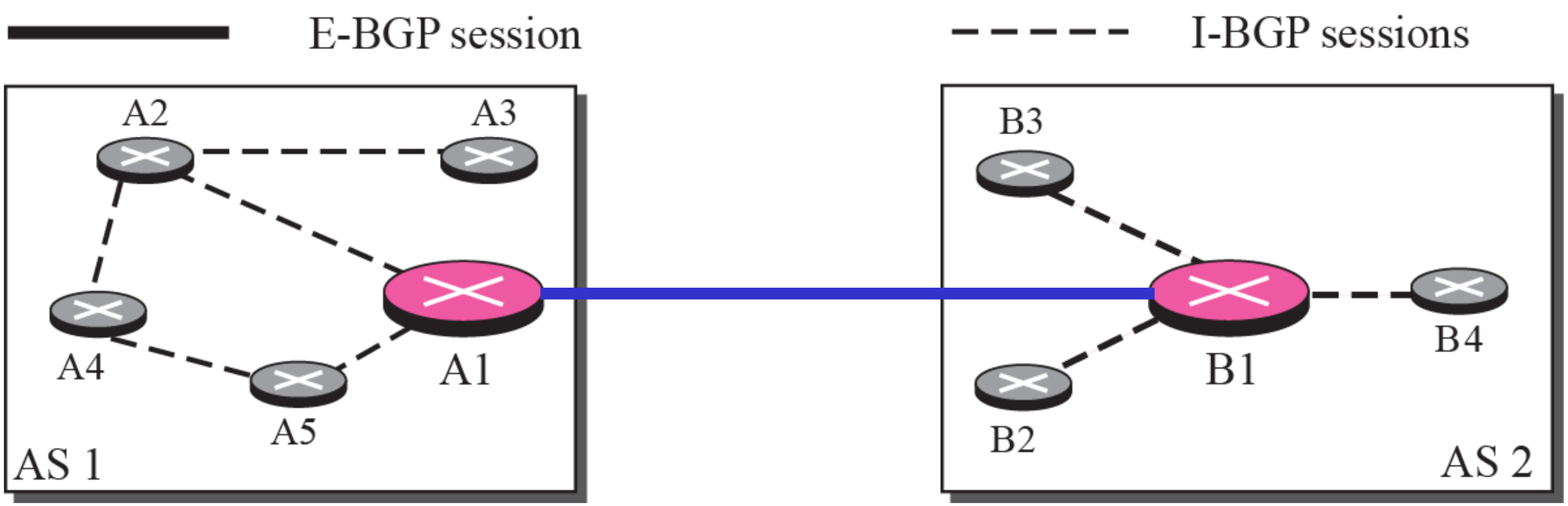
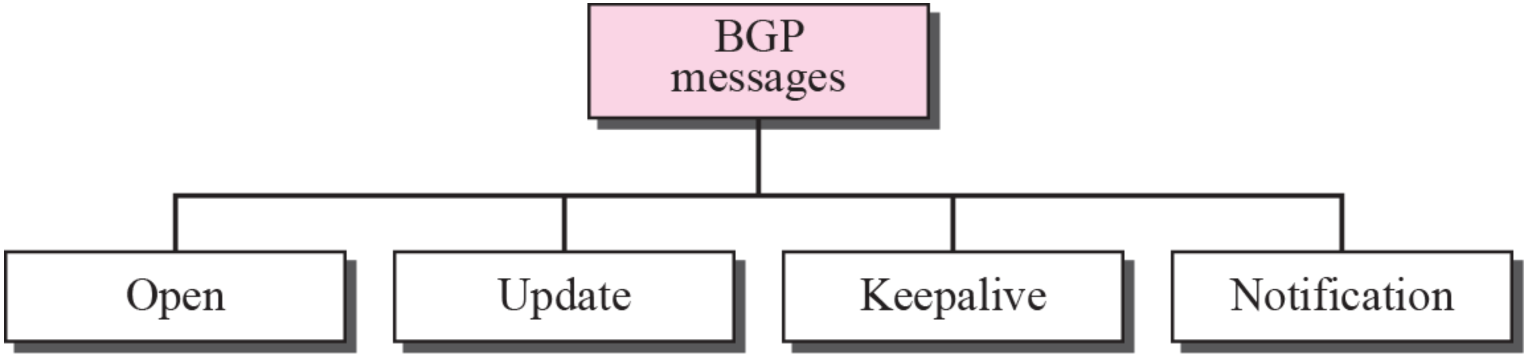


Figure 54 *Types of BGP messages*



Note

BGP supports classless addressing .

Note

BGP uses the services of TCP on port 179.