Data Communications and Networking Fourth Edition

Forouzan

Chapter 22

Network Layer: Delivery, Forwarding, and Routing

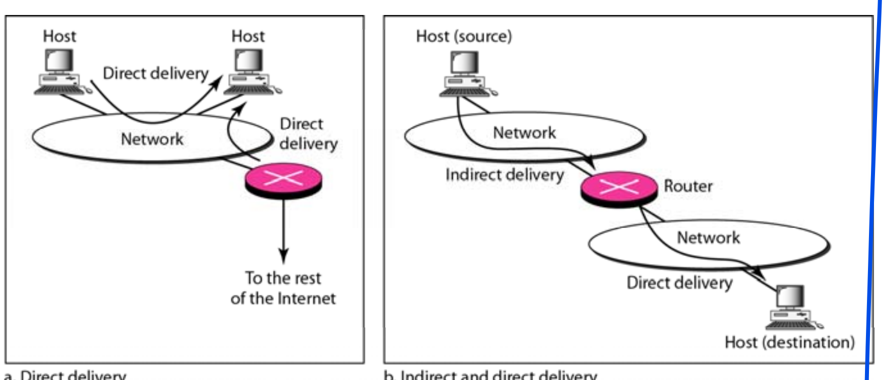
22-1 DELIVERY

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***



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

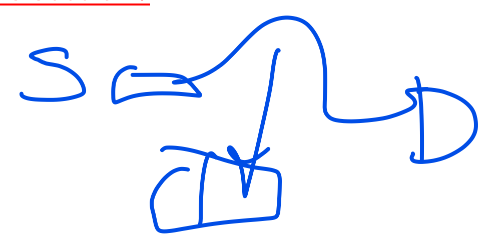


Figure 22.2 *Route method versus next-hop method*

a. Routing tables based on route

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	---

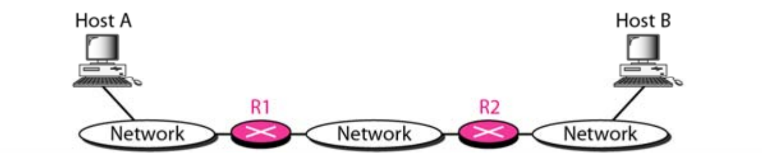


Figure 22.3 *Host-specific versus network-specific method*

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

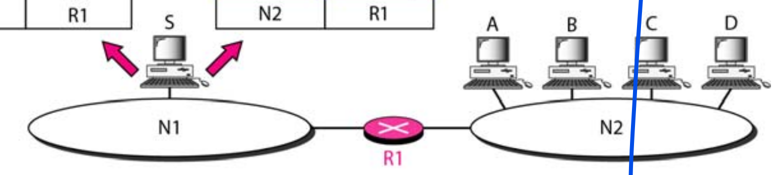


Figure 22.4 *Default method*

Destination	Next hop
N2	R1
Any other	R2

Routing table for host A

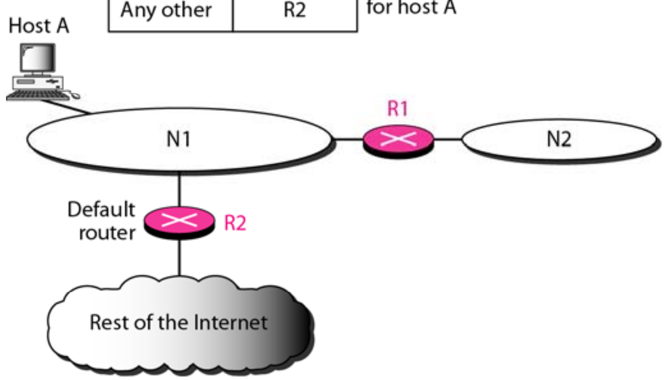
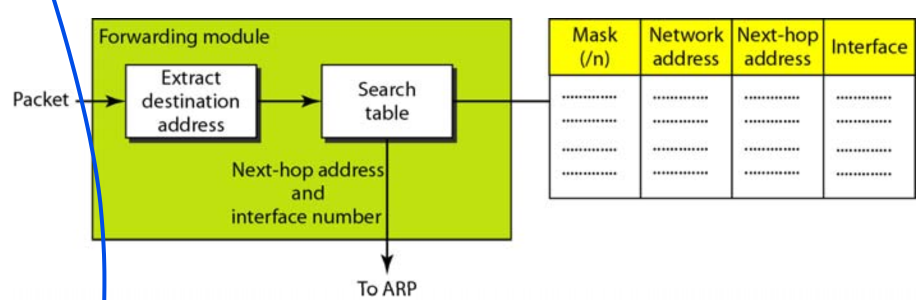


Figure 22.5 *Simplified forwarding module in classless address*



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*

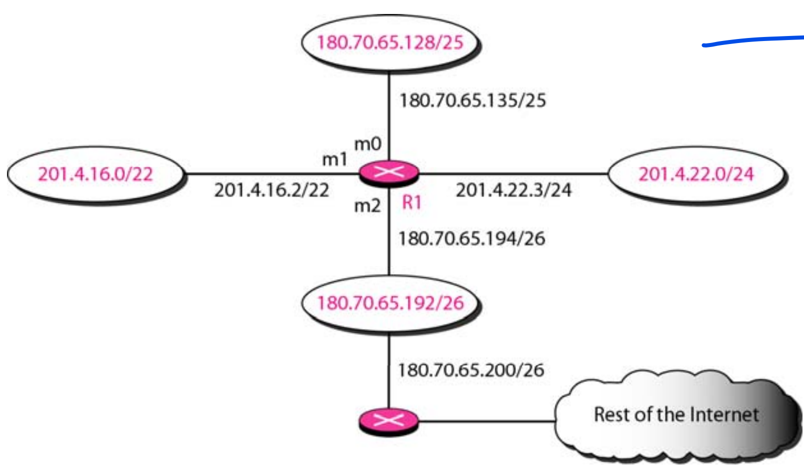


Table 22.1 *Routing table for router R1 in Figure 22.6*

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

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.*

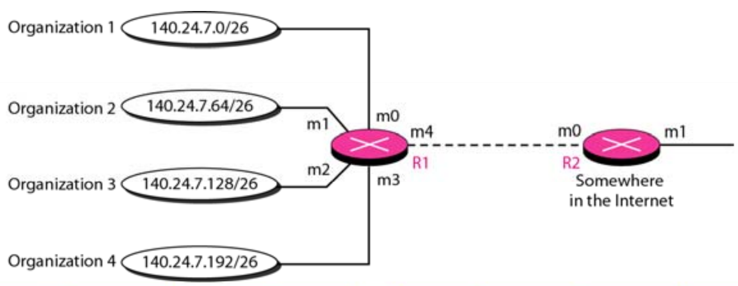
Example 22.4

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*



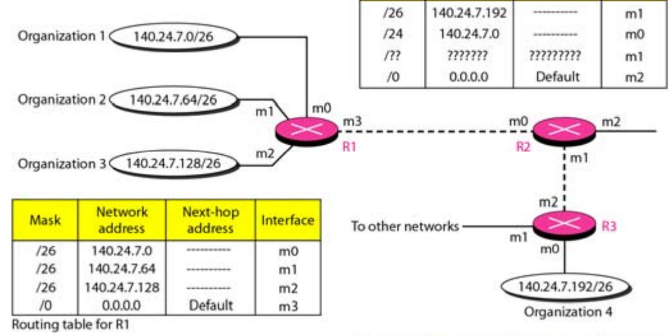
Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	Default	m4

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

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.

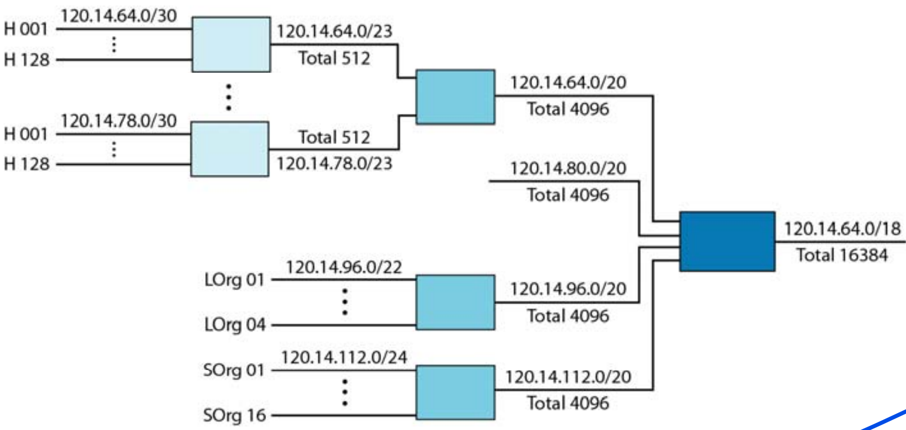
Example 22.5 (continued)

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*



Routing table:

- ***Static routing table: information entered manually***
- ***Dynamic routing table: updated periodically using routing protocols***

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

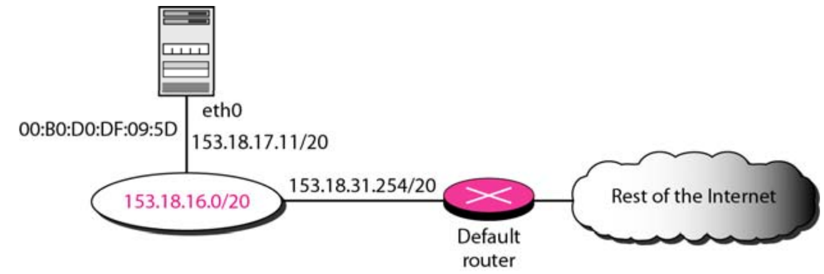
The destination column here defines the network address. The term gateway used by UNIX is synonymous with router. This column actually defines the address of the next hop. The value 0.0.0.0 shows that the delivery is direct. The last entry has a flag of G, which means that the destination can be reached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

Example 22.6 (continued)

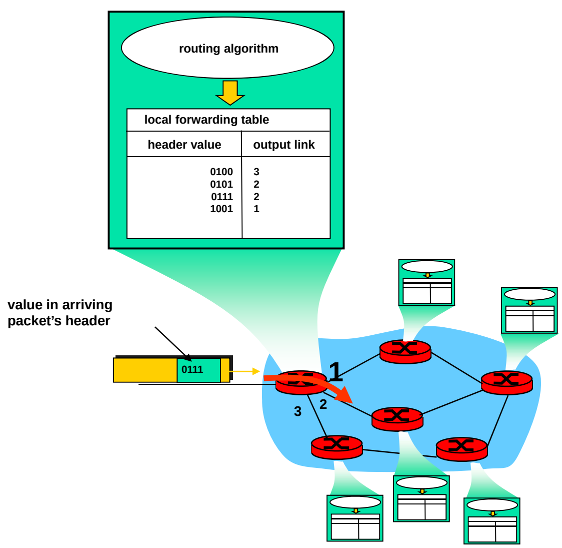
More information about the IP address and physical address of the server can be found by using the ifconfig command on the given interface (eth0).

```
$ 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
. . .
```

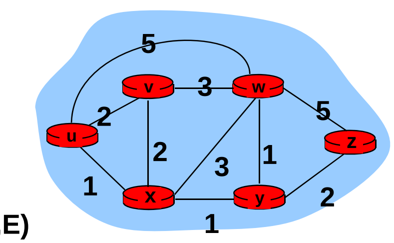

Figure 22.11 *Configuration of the server for Example 22.6*



Interplay between routing, forwarding



Graph abstraction



Graph: G = (N,E)

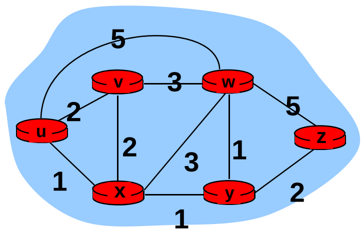
N = set of routers = { u, v, w, x, y, z }

E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z),

Remark: Graph abstraction is useful in other network contexts

Example: P2P, where N is set of peers and E is set of TCP connections

Graph abstraction: costs



- c(x,x') = cost of link (x,x')

  - e.g., c(w,z) = 5

- cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

Cost of path (x_1, x_2, x_3,…, x_p) = c(x_1,x_2) + c(x_2,x_3) + … + c(x_{p-1},x_p)

Question: What's the least-cost path between u and z ?

Routing algorithm: algorithm that finds least-cost path

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

22-3 UNICAST ROUTING PROTOCOLS

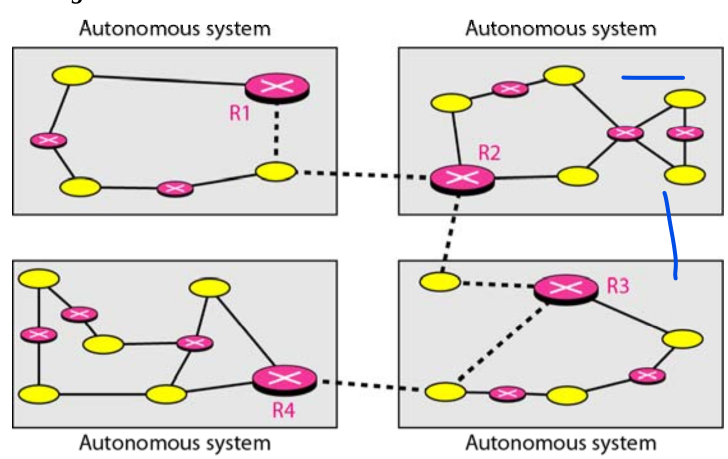
A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table is one that is updated automatically when there is a change somewhere in the Internet. A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each 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***

Autonomous system: a group of networks and routers under the authority of a singe administration



Ia

Ie

Figure 22.13 *Popular routing protocols*

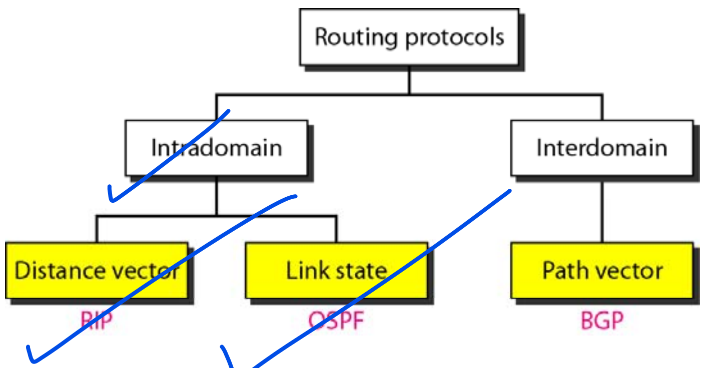
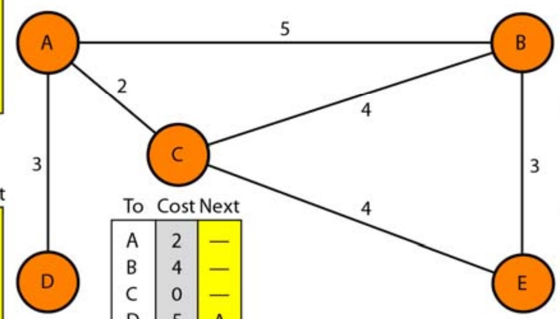


Figure 22.14 *Distance vector routing tables*



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	—

D's table

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

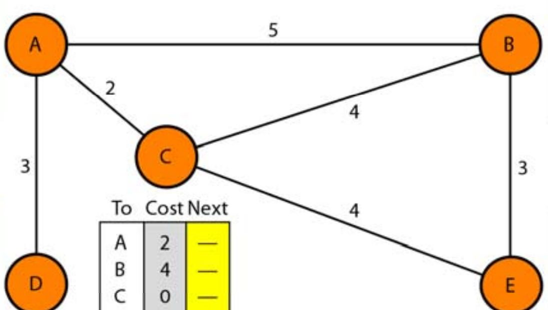
C's table

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

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*



A's table

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

B's table

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

D's table

To	Cost	Next
A	3	—
B	∞	
C	∞	
D	0	—
E	∞	

C's table

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

E's table

To	Cost	Next
A	∞	
B	3	B
C	4	C
D	∞	
E	0	D

Note

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

RIP Updating Algorithm
Receive: a response RIP message
1. Add one hop to the hop count for each advertised destination.
2. Repeat the following steps for each advertised destination:
1. If (destination not in the routing table)
1. Add the advertised information to the table.
2. Else
1. If (next-hop field is the same)
1. Replace entry in the table with the advertised one.
2. Else
1. If (advertised hop count smaller than one in the table)
1. Replace entry in the routing table.
3. Return.

Figure 22.16 *Updating in distance vector routing*

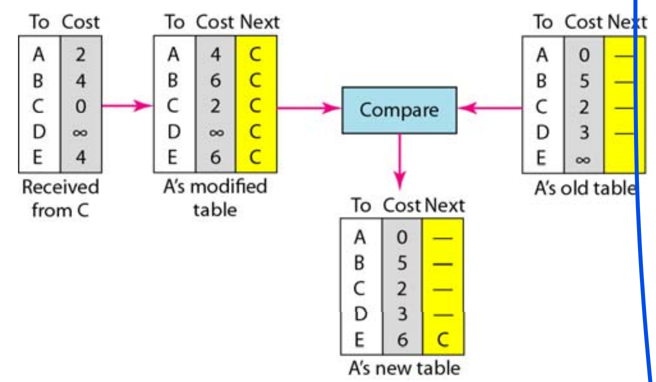
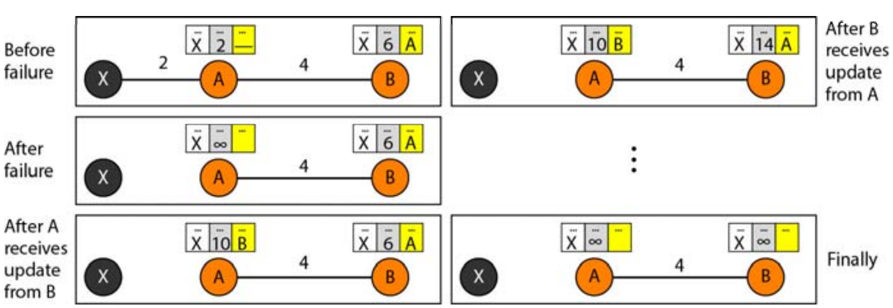


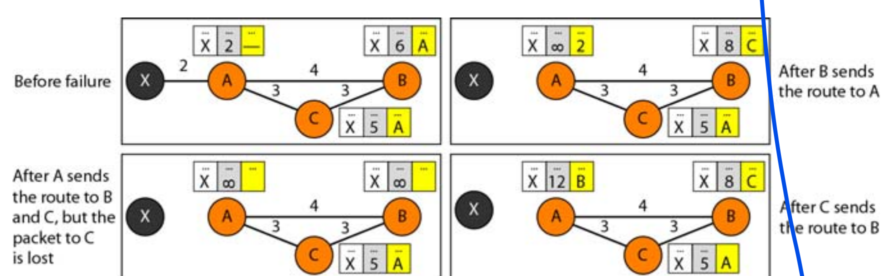
Figure 22.17 *Two-node instability*



Solutions:

Defining infinity, Split horizon, poison reverse

Figure 22.18 *Three-node instability*



Sharing tables distance vector

- When to Share
- The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors?
- The table is sent both periodically and when there is a change in the table.
- Periodic Update: A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.
- Triggered Update: A node sends its two-column routing table to its neighbors any- time there is a change in its routing table. This is called a triggered update.
- The change can result from the following.
  - A node receives a table from a neighbor, resulting in changes in its own table after updating.
  - A node detects some failure in the neighboring links which results in a distance change to infinity

RIP Protocol

- The Routing Information Protocol (RIP) is an intra-domain 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:
- 1. In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
- 2. The destination in a routing table is a network, which means the first column defines a network address.
- 3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
- 4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than15 hops

Figure 22.19 *Example of a domain using RIP*

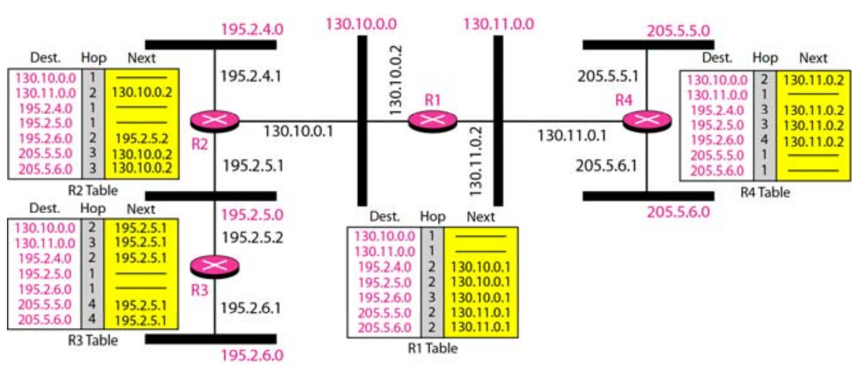


Figure 22.20 *Concept of link state routing*

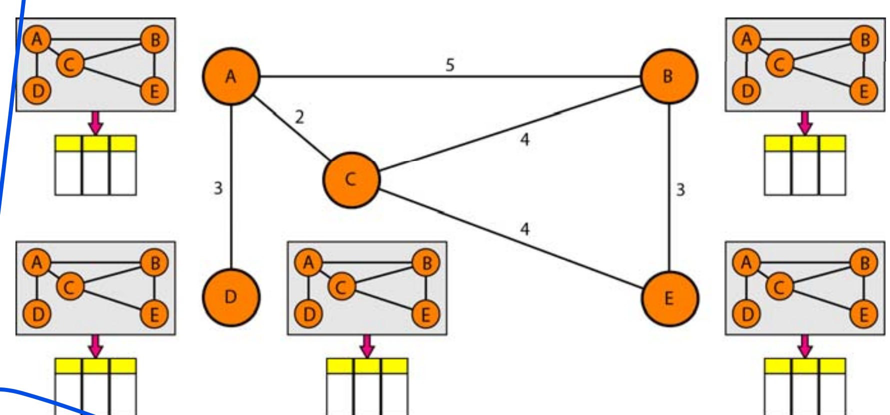


Figure 22.21 *Link state knowledge*

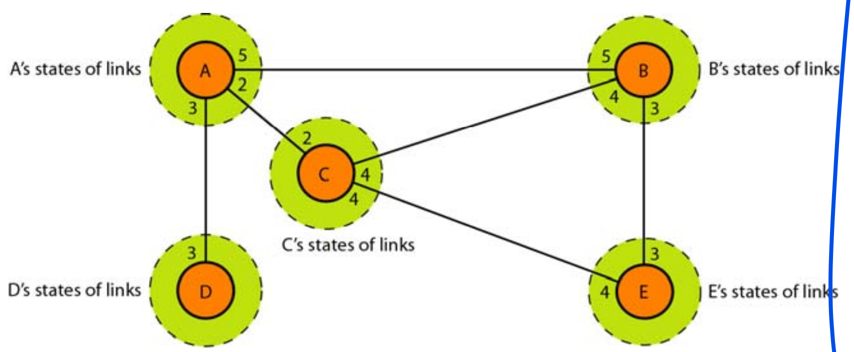


Figure 22.22 *Dijkstra algorithm*

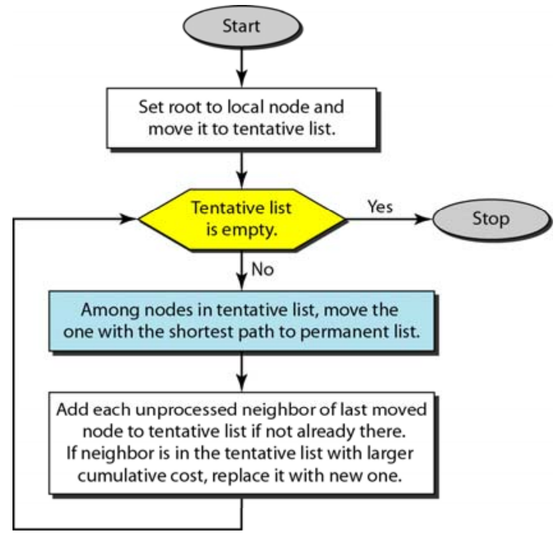


Figure 22.23 *Example of formation of shortest path tree*

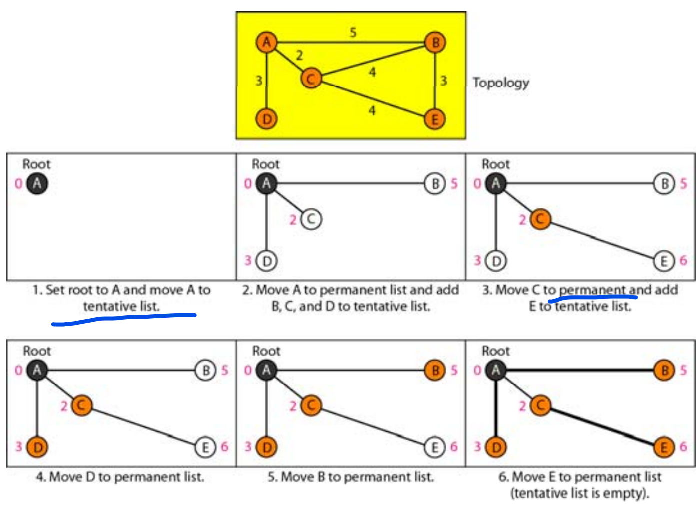


Table 22.2 *Routing table for node A*

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

Figure 22.24 ***Areas in an autonomous system***

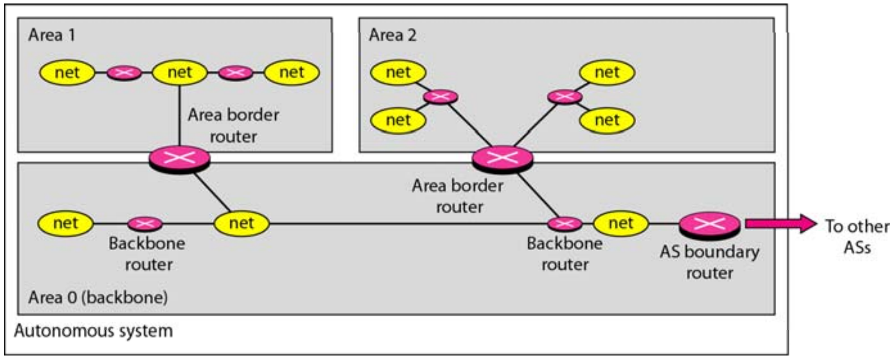


Figure 22.25 ***Types of links***

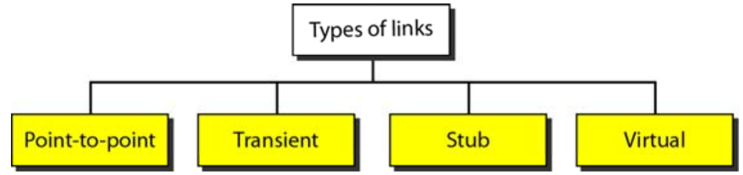


Figure 22.26 ***Point-to-point link***

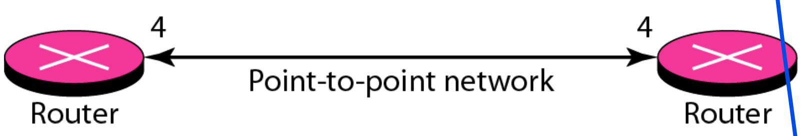


Figure 22.27 ***Transient link***

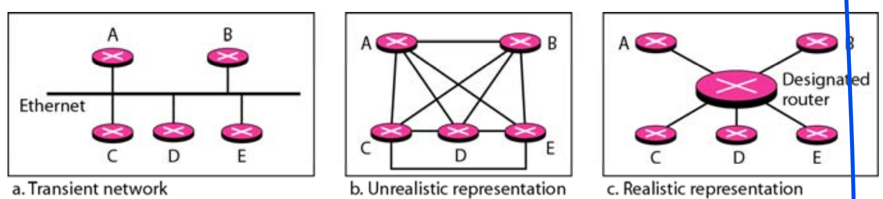


Figure 22.28 *Stub link*

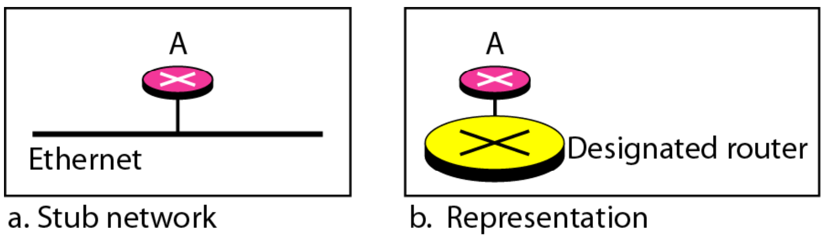


Figure 22.29 *Example of an AS and its graphical representation in OSPF*

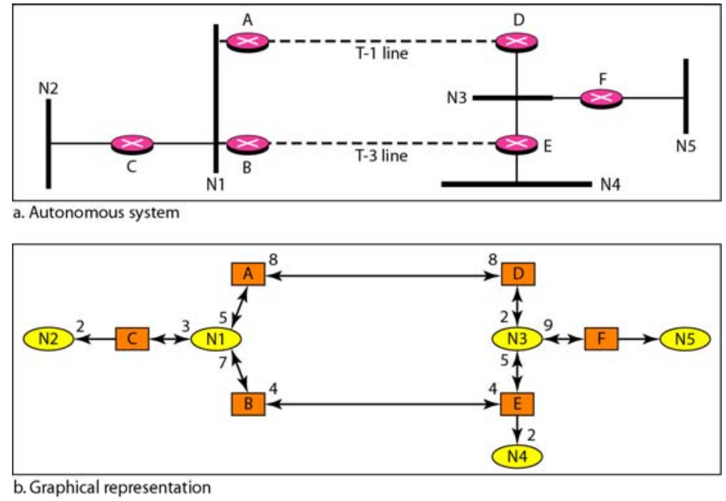


Figure 22.30 *Initial routing tables in path vector routing - speaker node*

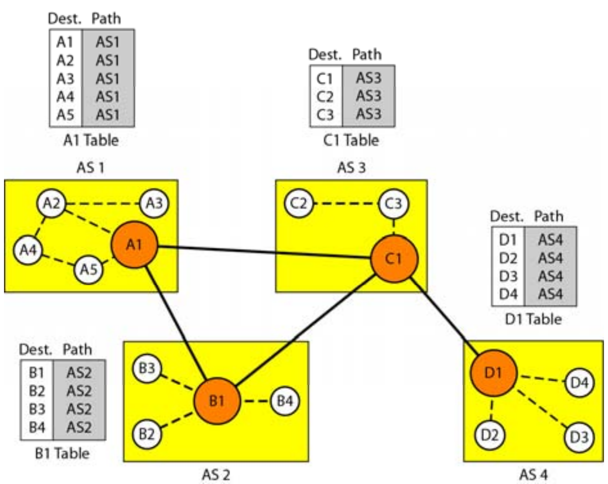


Figure 22.31 *Stabilized tables for three autonomous systems*

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

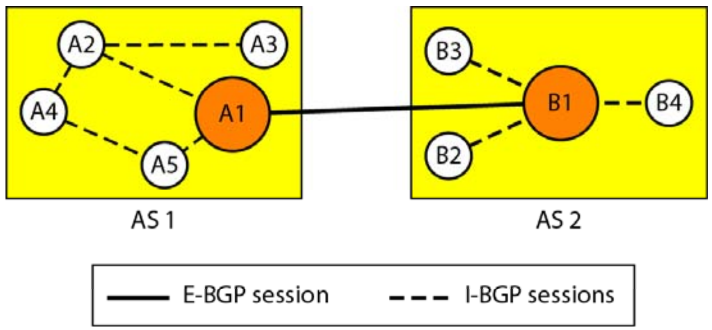
Dest.	Path
A1	AS3-AS1
...	...
A5	AS3-AS1
B1	AS3-AS2
...	...
B4	AS3-AS2
C1	AS3
...	...
C3	AS3
D1	AS3-AS4
...	...
D4	AS3-AS4

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*



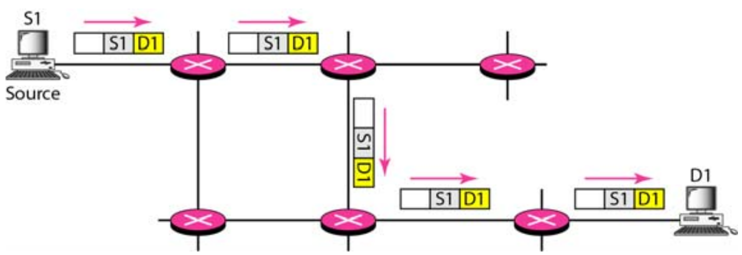
22-4 MULTICAST ROUTING PROTOCOLS

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

Topics discussed in this section:

Unicast, Multicast, and Broadcast
Applications
Multicast Routing
Routing Protocols

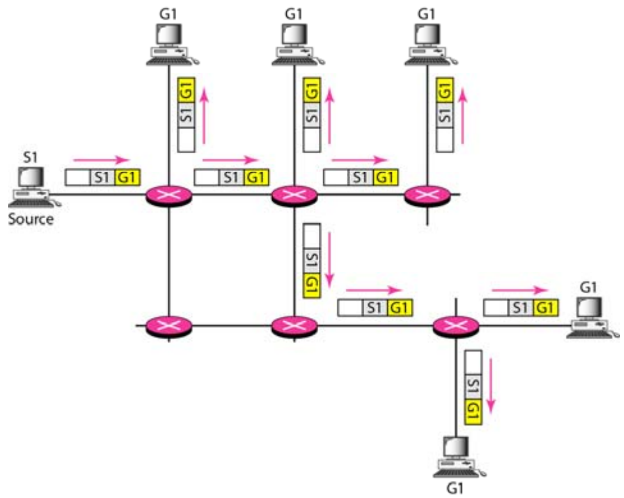
Figure 22.33 *Unicasting*



Note

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

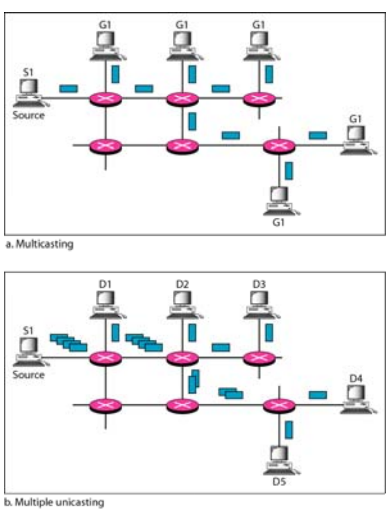
Figure 22.34 *Multicasting*



Note

In multicasting, the router may forward the received packet through several of its interfaces.

Figure 22.35 *Multicasting versus multiple unicasting*



Note

Emulation of multicasting through multiple unicasting is not efficient and may create long delays, particularly with a large group.

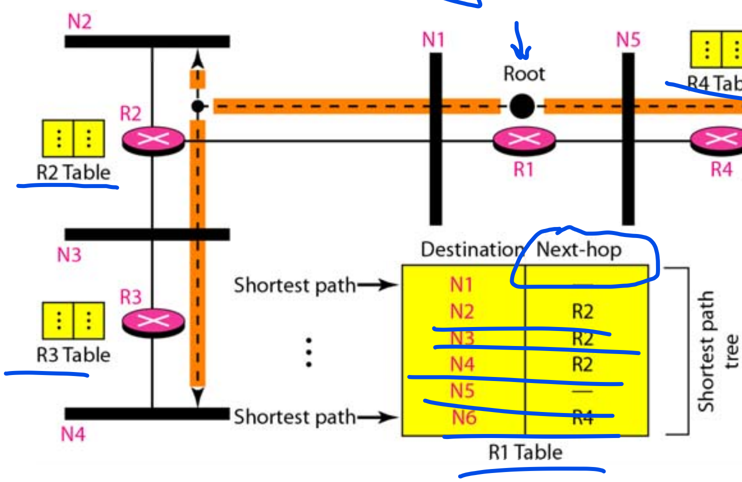
MC → eff ↑ BW ↓ npd ↓

Note

In unicast routing, each router in the domain has a table that defines a shortest path tree to possible destinations.

SPT

Figure 22.36 *Shortest path tree in unicast routing*



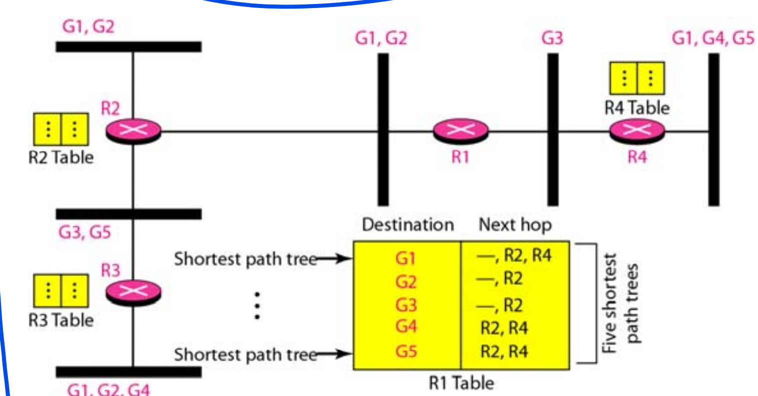
SPT

G1

Note

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

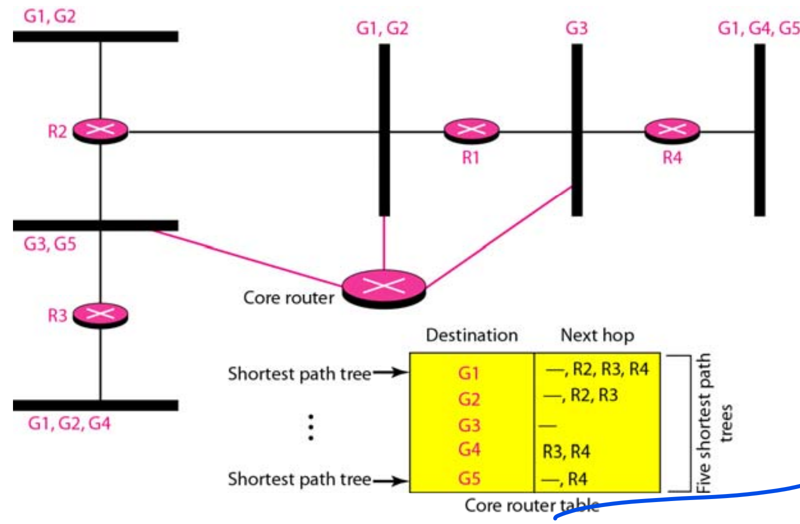
Figure 22.37 *Source-based tree approach*



Note

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

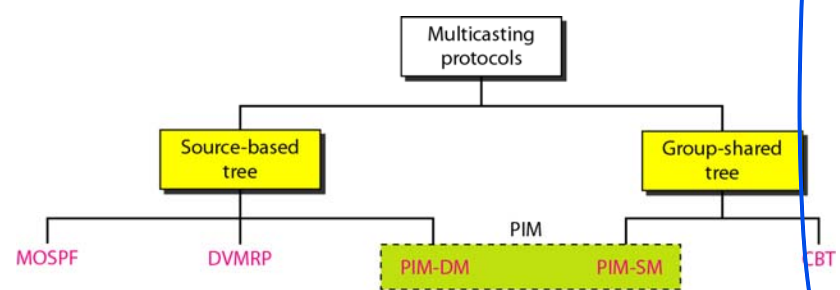
Figure 22.38 *Group-shared tree approach*



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)*

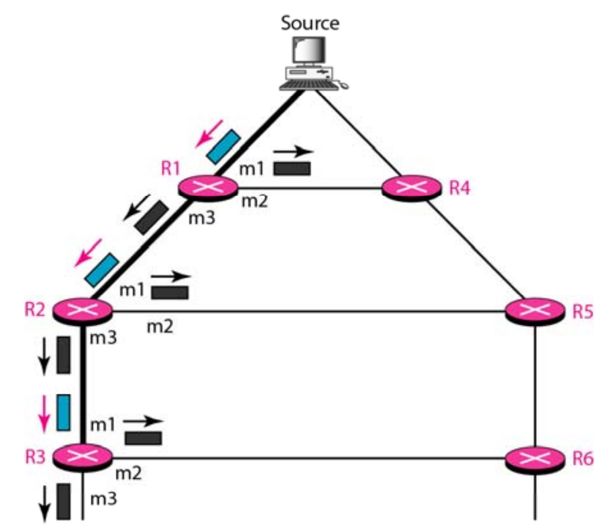


Figure 22.41 *Problem with RPF*

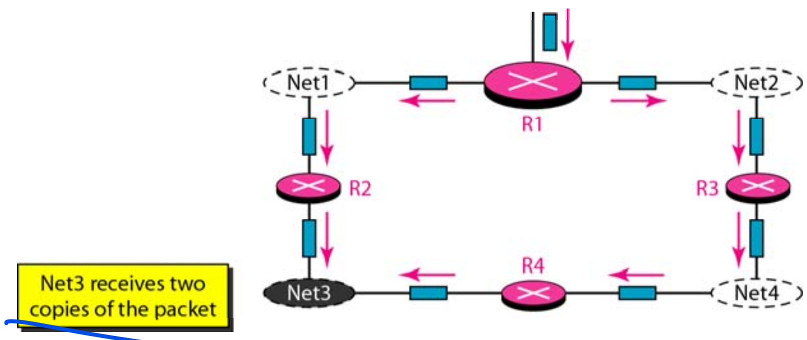
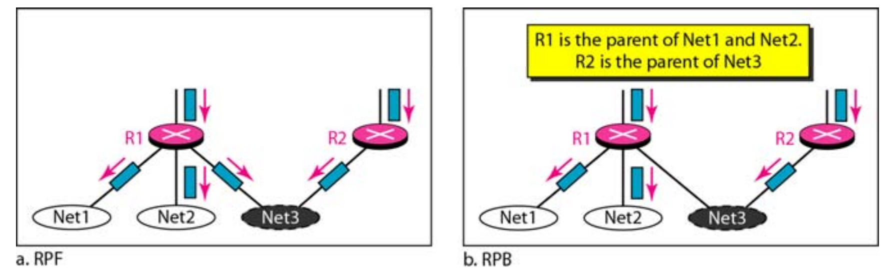


Figure 22.42 *RPF Versus RPB*

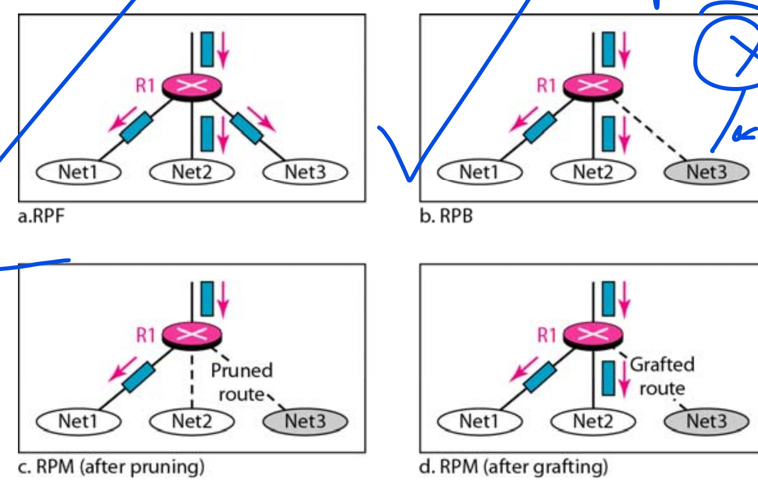


Note

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*

parent



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*

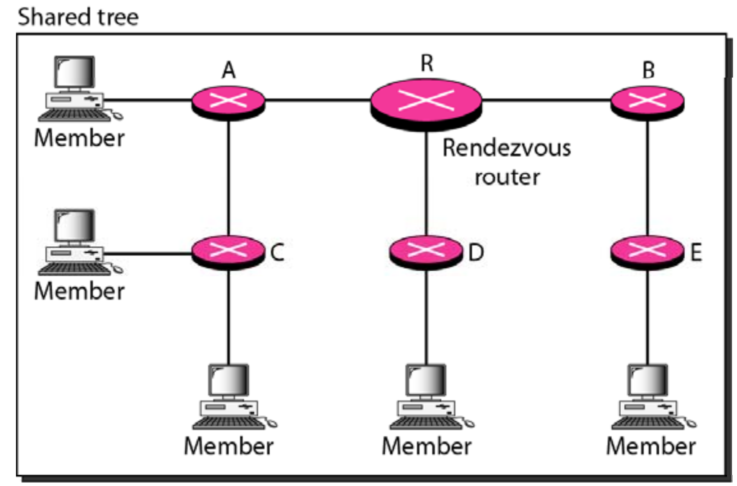
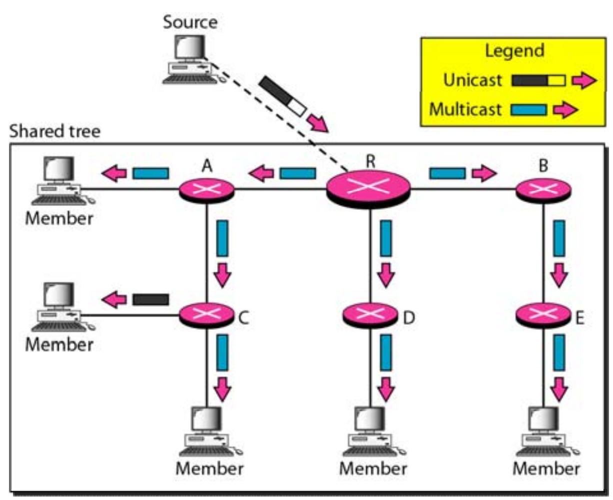


Figure 22.45 *Sending a multicast packet to the rendezvous router*



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.

85

Note

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

86

Note

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

87

Note

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

88

Figure 22.46 *Logical tunneling*

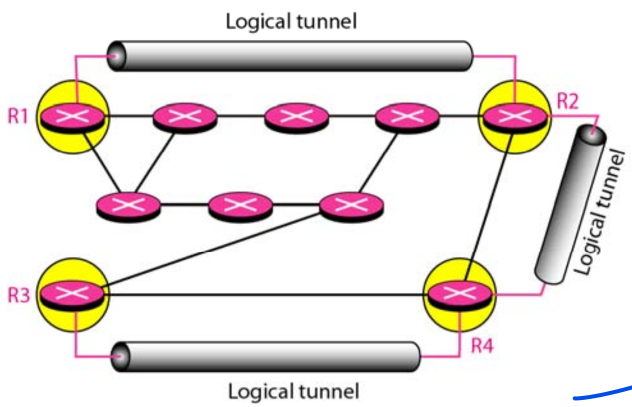
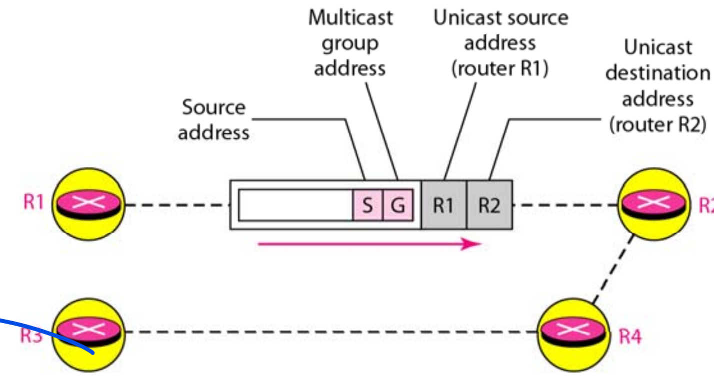
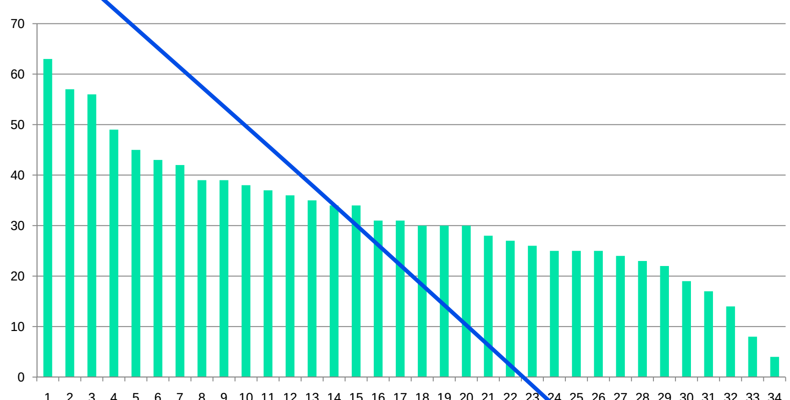


Figure 22.47 *MBONE*





Mean	31.94117647
Standard Error	2.232853532
Median	30.5
Mode	30
Standard Deviation	13.01966153
Sample Variance	169.5115865
Kurtosis	0.444990404
Skewness	0.290623412
Range	59
Minimum	4
Maximum	63
Sum	1086
Count	34