

Module 2

Network Layer: Delivery, Forwarding, and Routing

22-1 DELIVERY

The network layer supervises the handling of the packets by the underlying physical networks.

→ defined as the **delivery of a packet**.

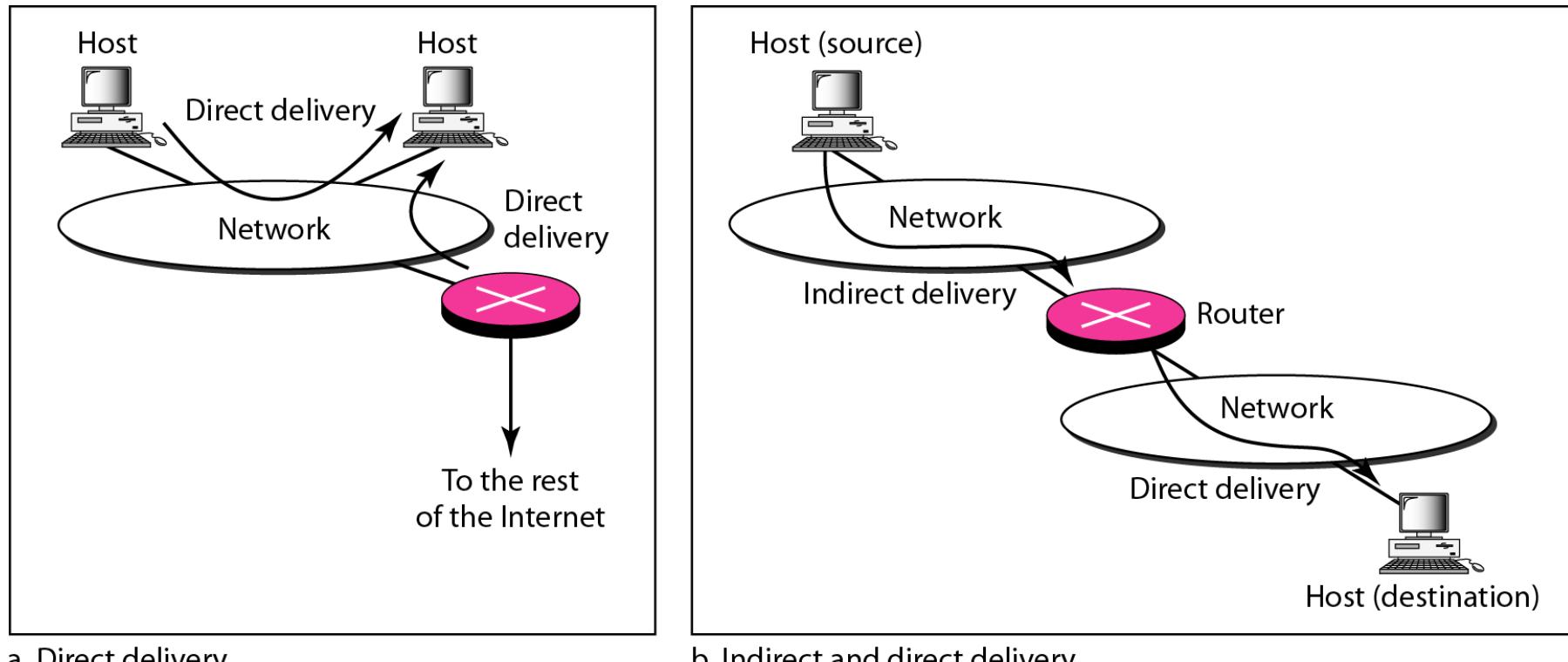
Direct Versus Indirect Delivery

Direct delivery occurs when **source and destination of packet are located on the same physical network OR**

When the delivery is between last Router and destination Host

In indirect delivery, **packet travels from router to router** until it reaches the one connected to same physical network as the destination

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

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 refers to 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

Destination	Route
Host B	R1, R2, host B

Routing table
for host A

Destination	Route
Host B	R2, host B

Routing table
for R1

Destination	Route
Host B	Host B

Routing table
for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

Destination	Next hop
Host B	---

Host A



Network

R1

Host B



Network

R2

Network

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

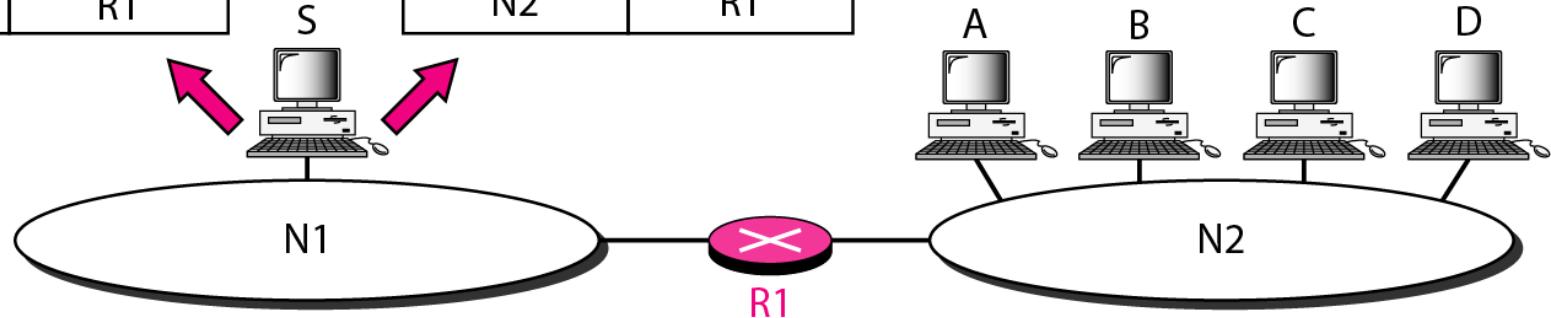


Figure 22.4 Default method

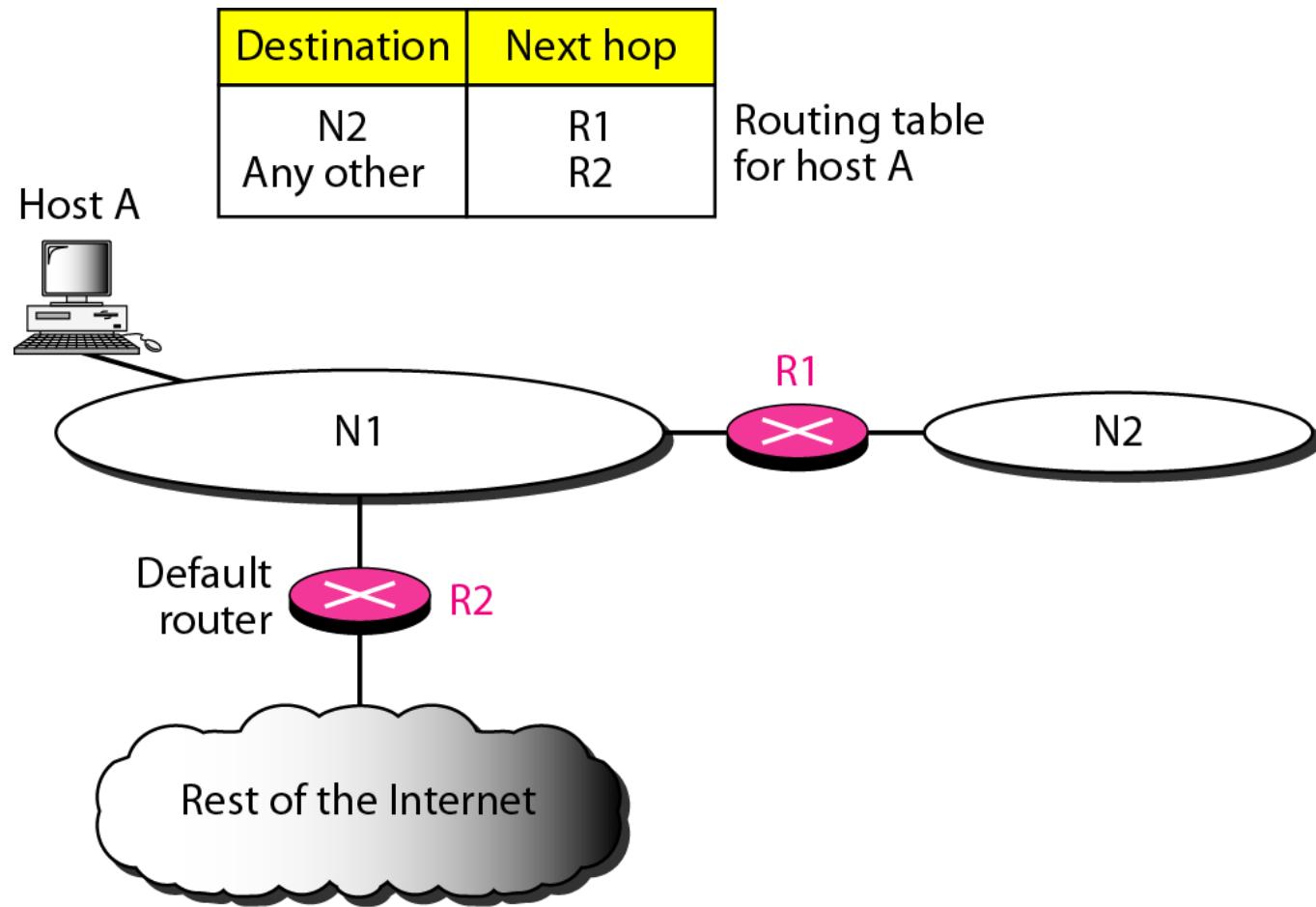
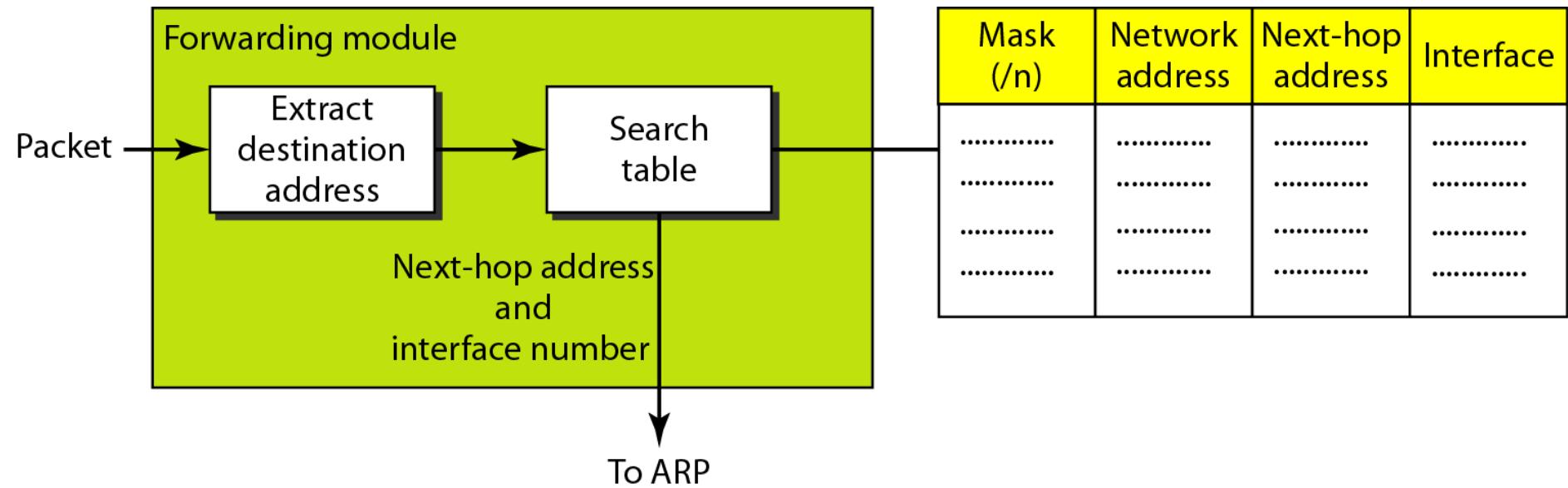
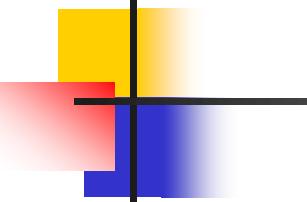


Figure 22.5 Simplified forwarding module in classless address





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

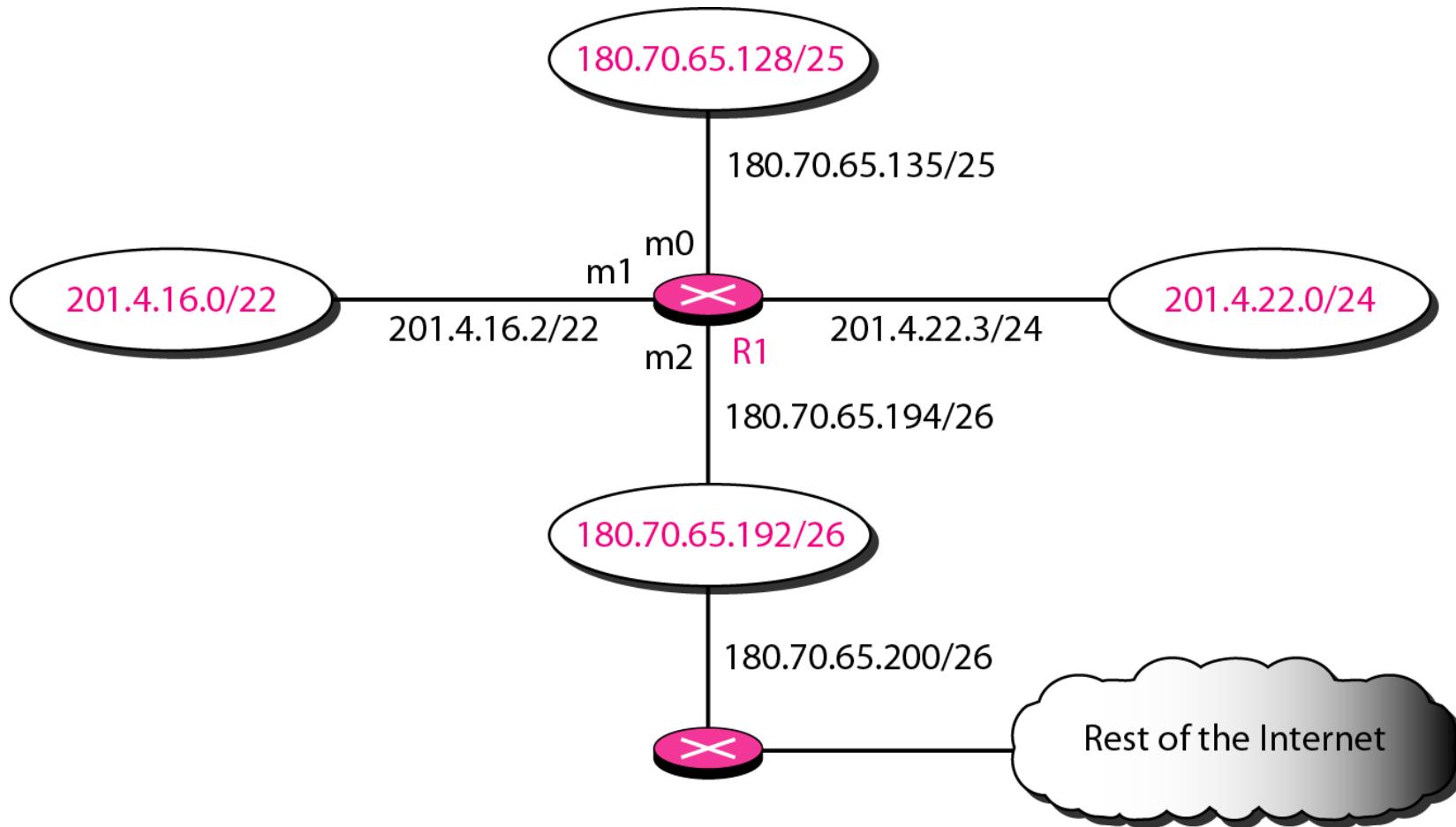
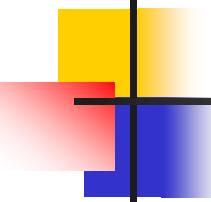


Table 22.1 *Routing table for router R1 in Figure 22.6*

<i>Mask</i>	<i>Network Address</i>	<i>Next Hop</i>	<i>Interface</i>
/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



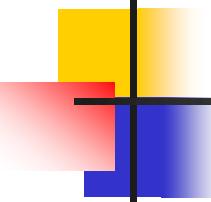
Example 22.2

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.***



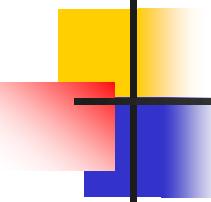
Example 22.3

Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 201.4.22.35.

Solution

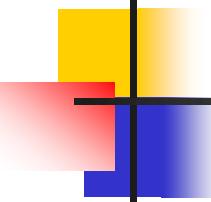
The router performs the following steps:

- 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

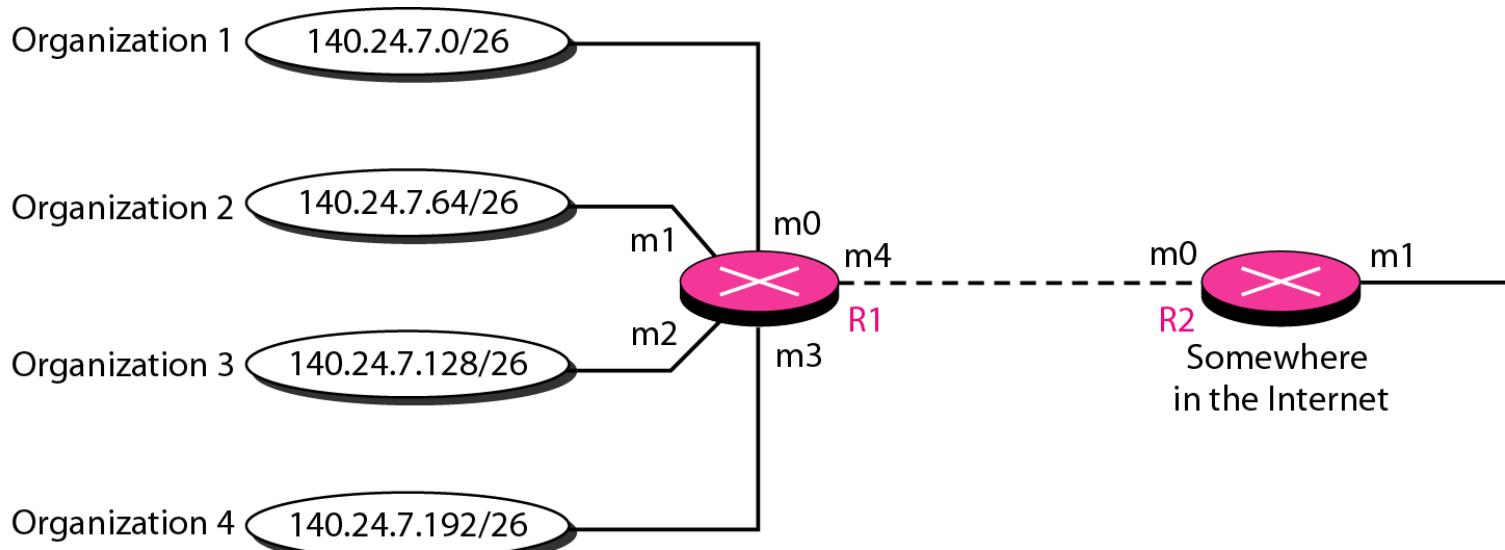
Show the forwarding process if a packet arrives at R1 in Figure 22.6 with the destination address 18.24.32.78.

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 packet that needs to be sent, via the default router, to someplace elsewhere 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

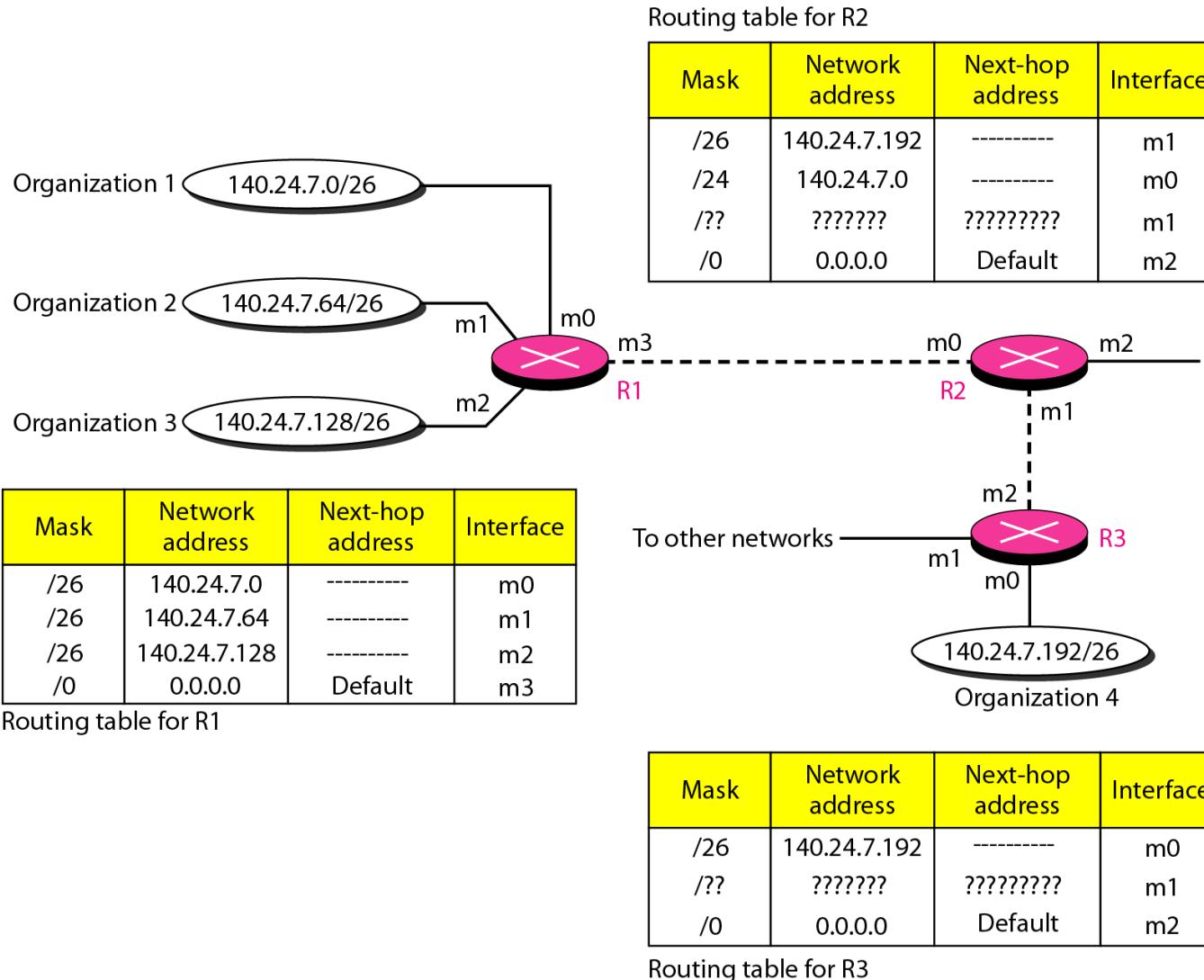
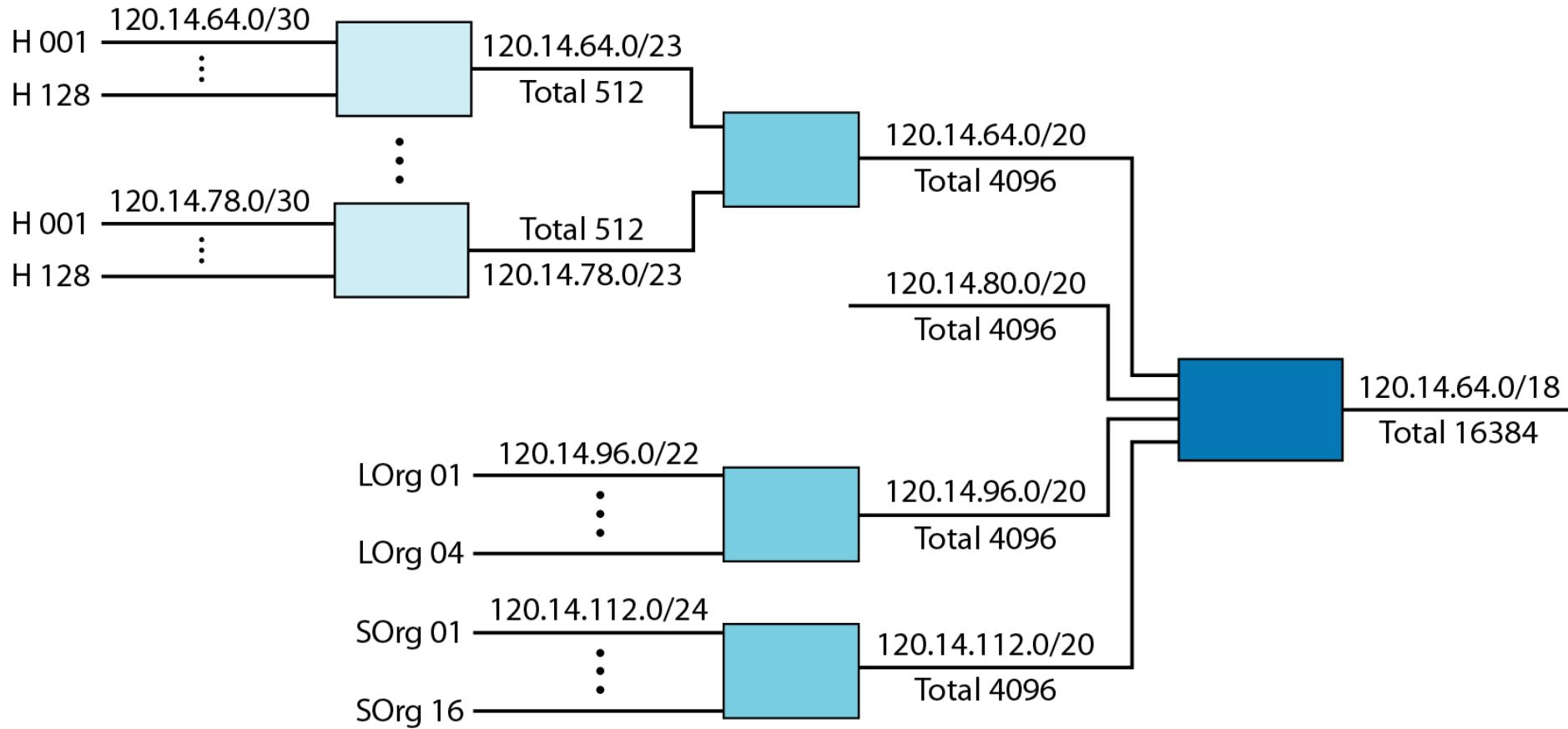


Figure 22.9 *Hierarchical routing with ISPs*



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 120.14.64.0/18*

*The regional ISP divides 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 → mask /23

*Each small ISP provides services to 128 households, each using **4 addresses** → mask /30*

Example 22.5 (continued)

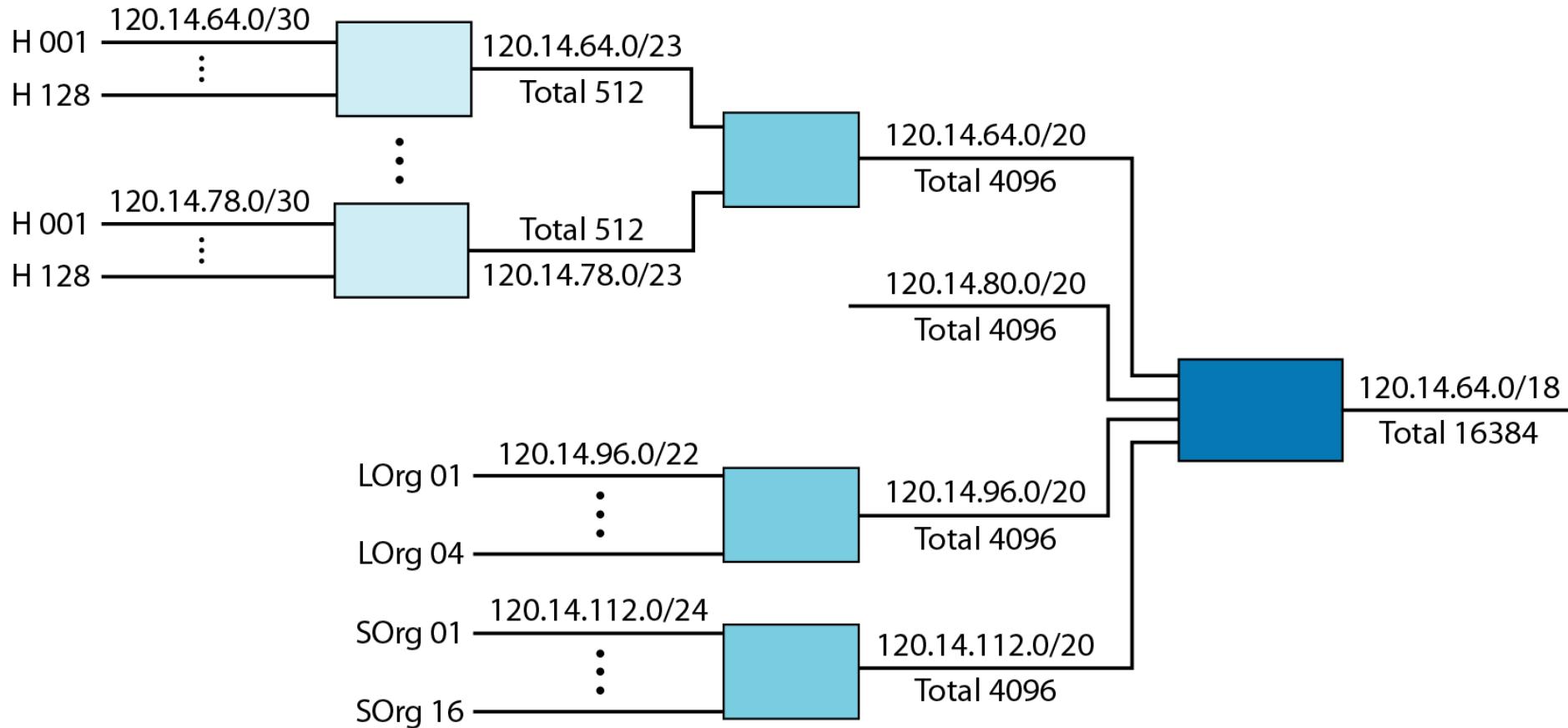
The second local ISP has divided its block into 4 blocks and has assigned the addresses to four large organizations → addresses 1024; mask is /22

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 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*



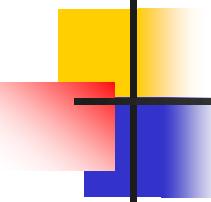
Routing Table

Static Routing tables- Entries done manually; fixed- cannot update automatically when there is a change in the network

Dynamic Routing tables- Updated periodically (or whenever there is a change in the network) using Dynamic Routing protocol e.g RIP, OSPF etc.

Common fields in a routing table

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



Example 22.6

*Utility that can be used to find the contents of a routing table for a host or router is **netstat** in UNIX or LINUX.*

*Two options, **r** and **n**; 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** that actually defines the **next hop address**. 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.*

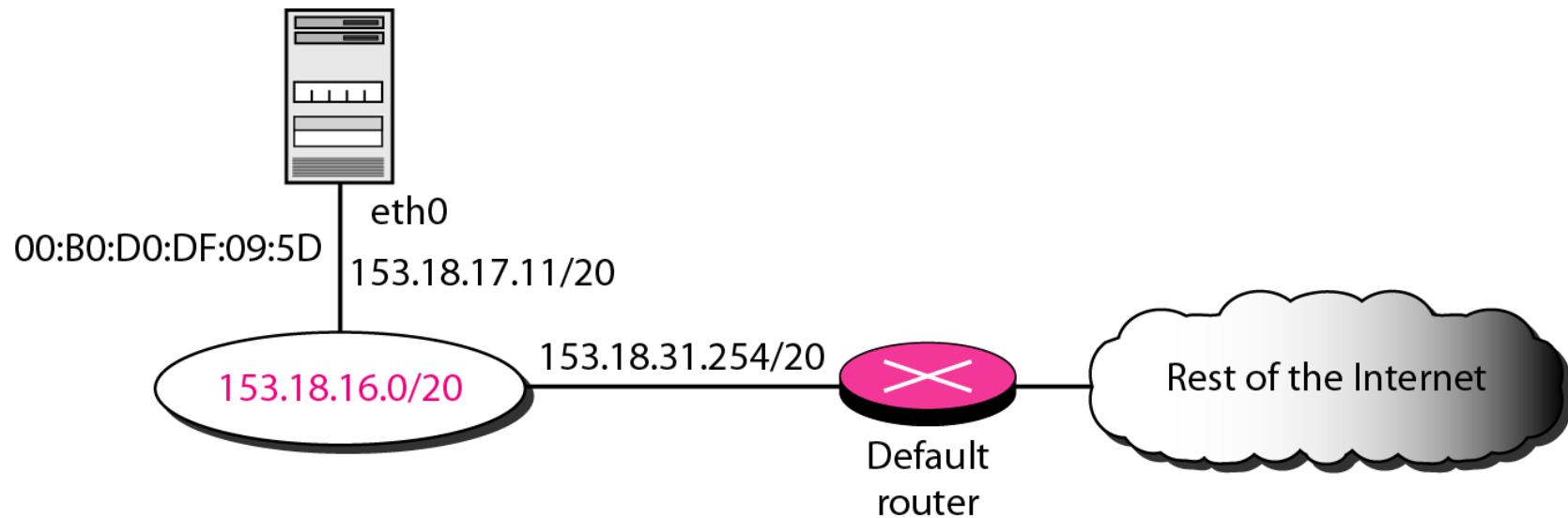
Example 22.6 (continued)

*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



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

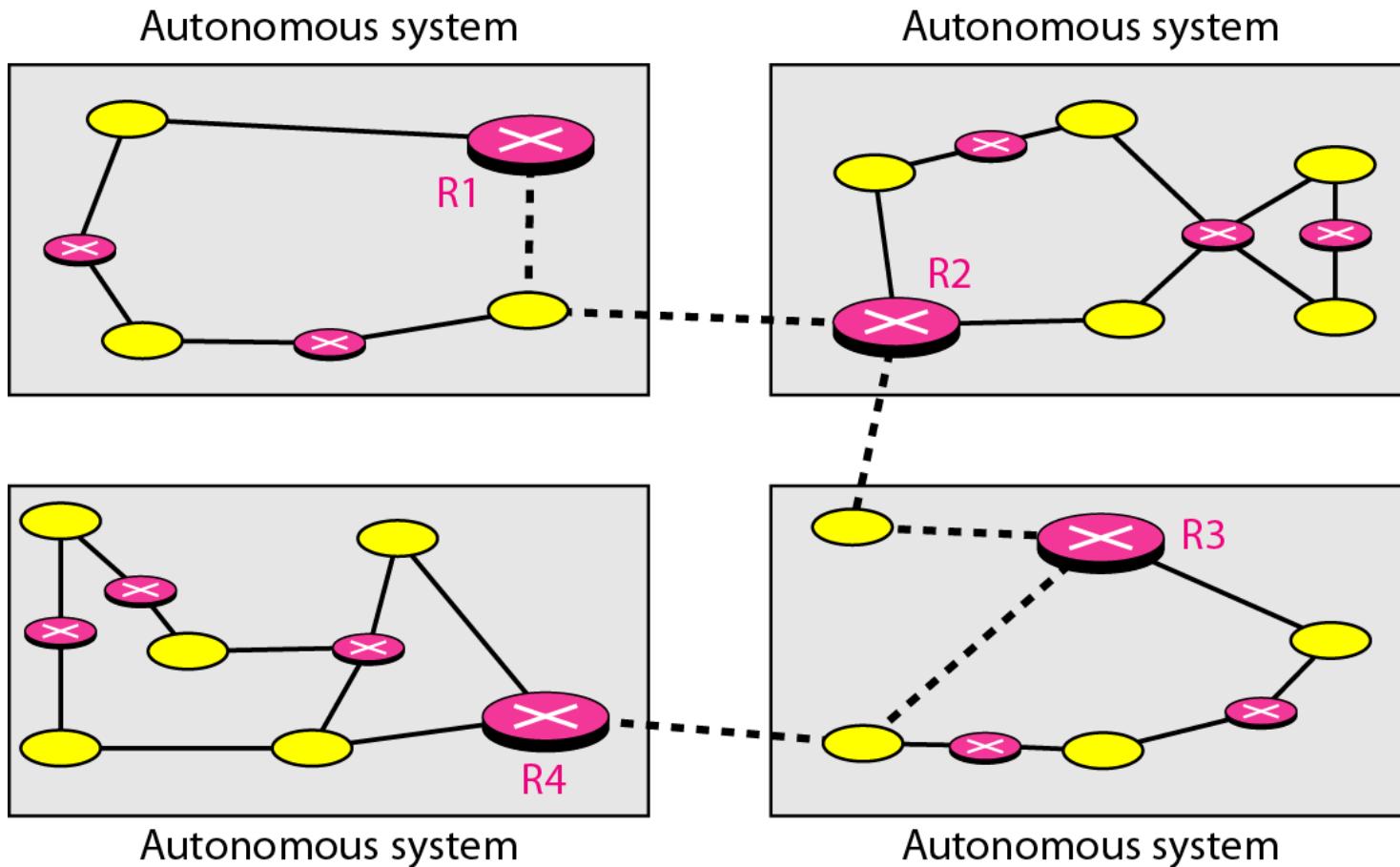
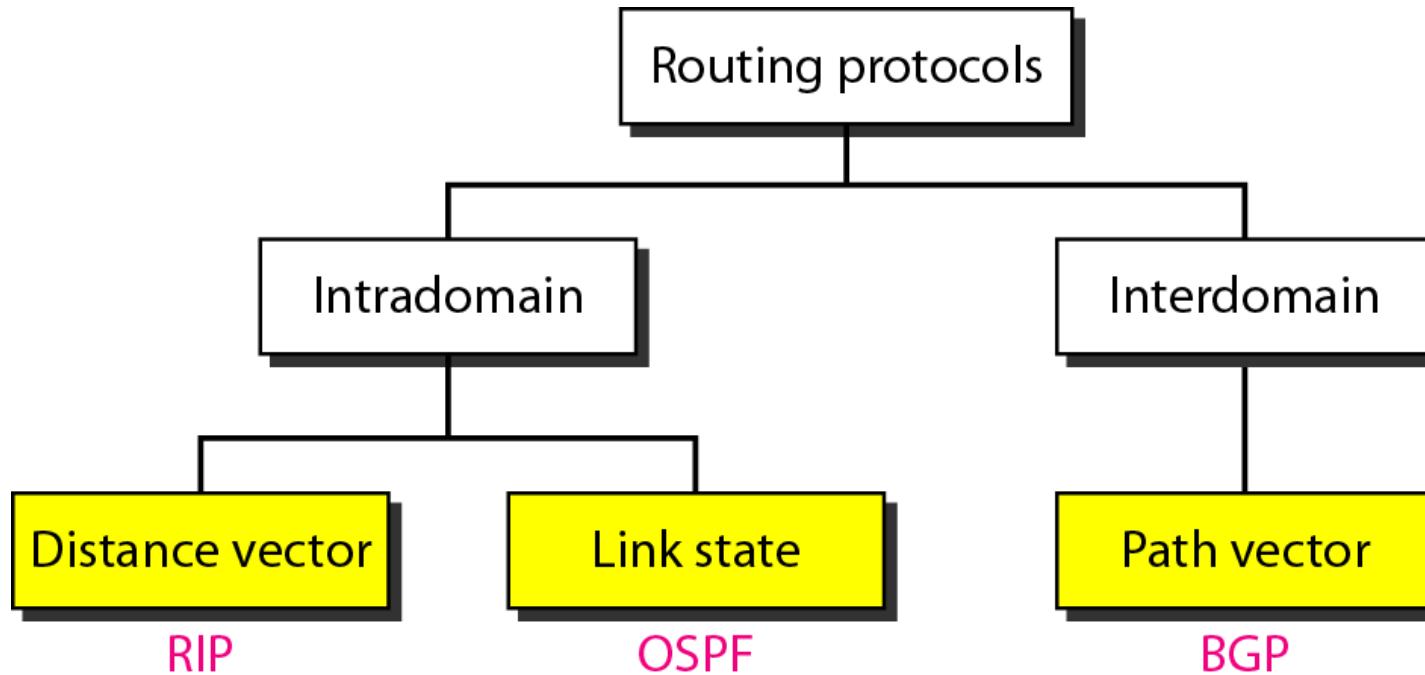


Figure 22.13 *Popular routing protocols*



Distance Vector Routing

Goal

- *To find the best route (least cost/distance route) to the destination*

Principle

- *Nodes share the local knowledge to acquire the global*

Steps

- *Acquire the local knowledge (learn)*
- *Share this with others (cooperate)*
- *Update global knowledge*

Figure 22.14 Distance vector routing tables

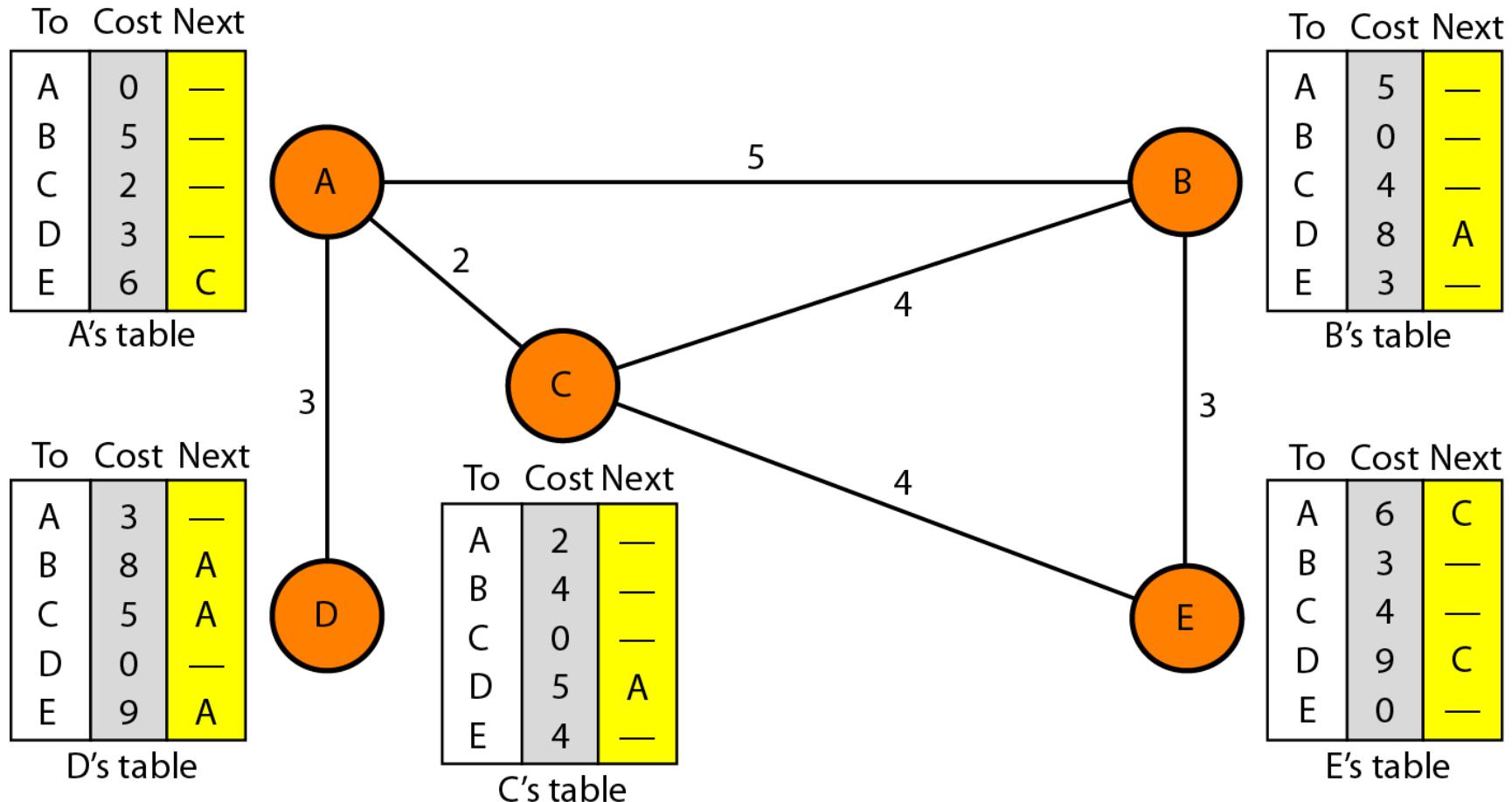
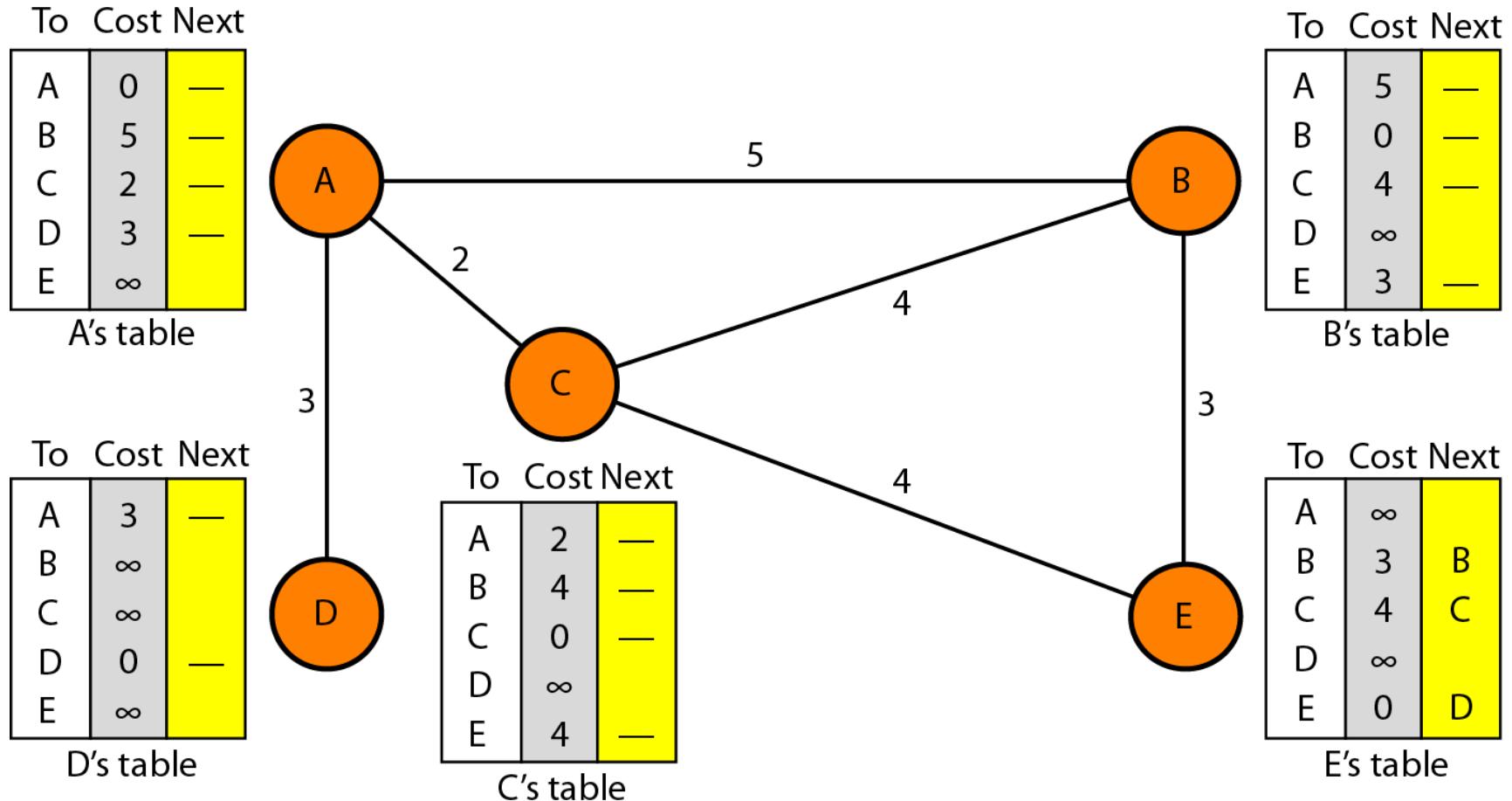
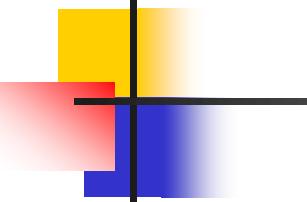


Figure 22.15 Initialization of tables in distance vector routing





Note

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

Updating the table

Comparison

- *If next node entry is different, receiving node chooses the entry with smaller cost.*
- *If next node entry is same, receiving node chooses the new cost*

Figure 22.16 Updating in distance vector routing

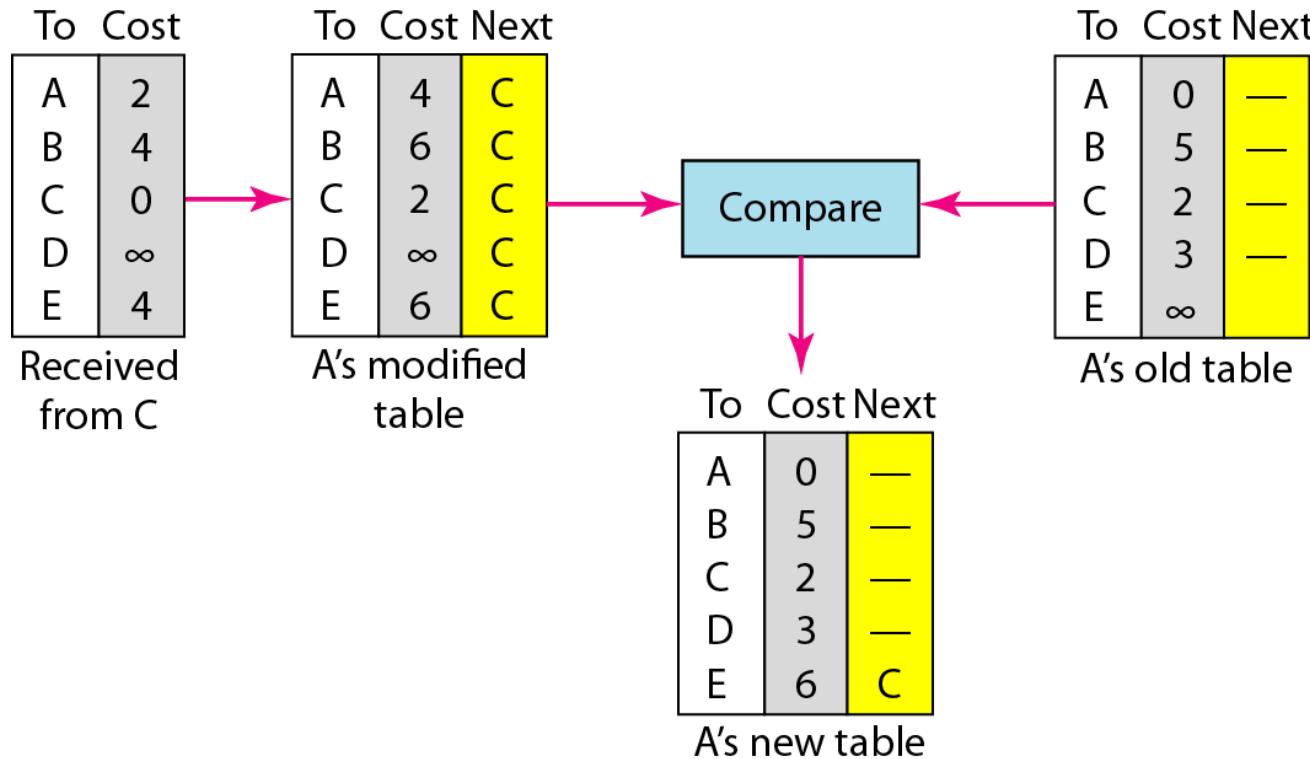
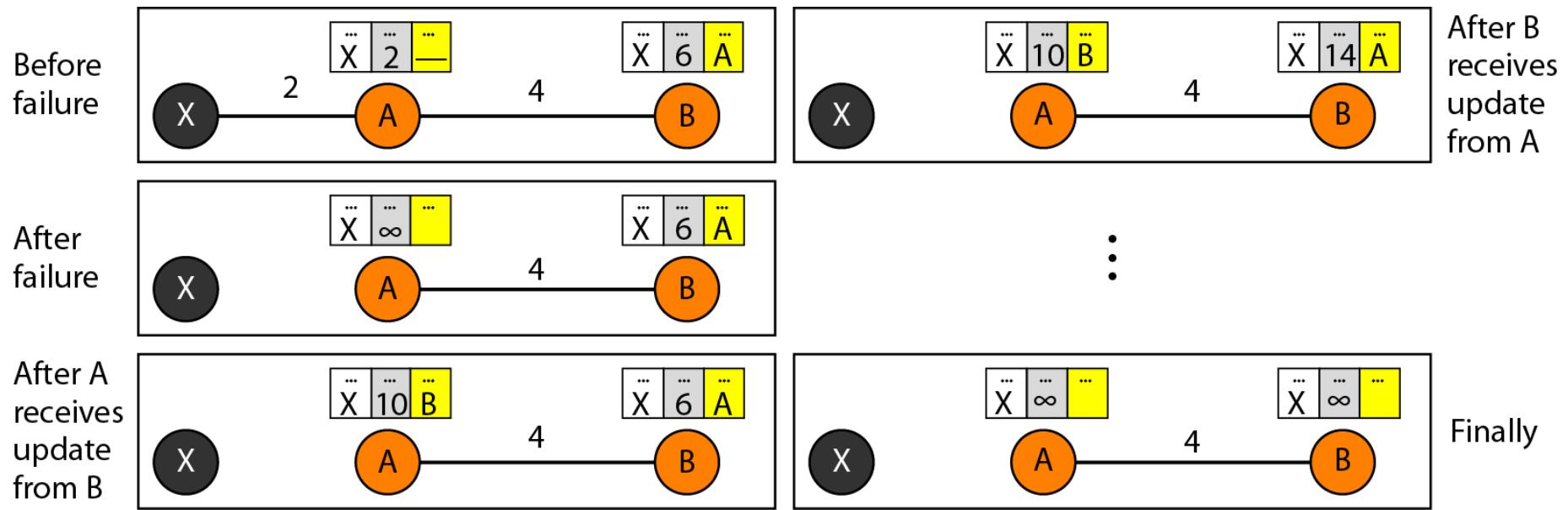


Figure 22.17 Two-node instability



Solution for Instability

- **Defining Infinity :**

- 16 as infinity defined for the Distance vector routing (RIP)

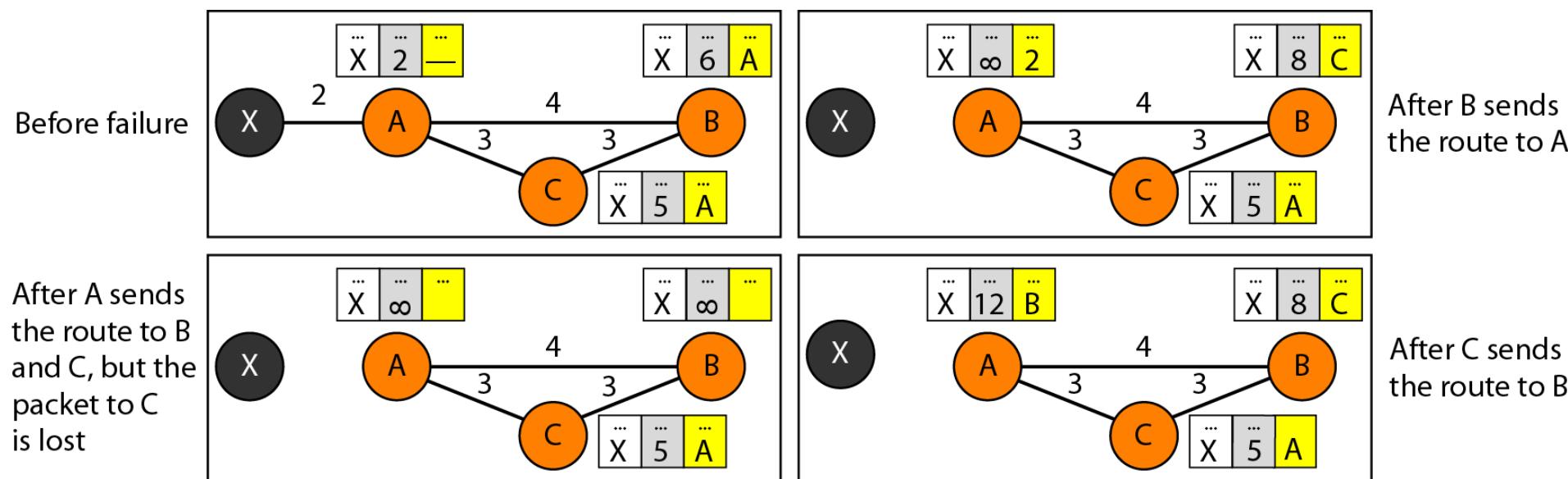
Split Horizon: Each node sends only part of its table through the interface.

- B will not send the entry for X to A (since this info came from A)

- **Split Horizon and Poison Reverse :**

- If node does not get news about the route for some predefined time, the node deletes the route from its table (node A cant guess this is due to split horizon)
- **Poison Reverse :** B can advertise the value for X but the distance it will mark as infinity for A → warning to A

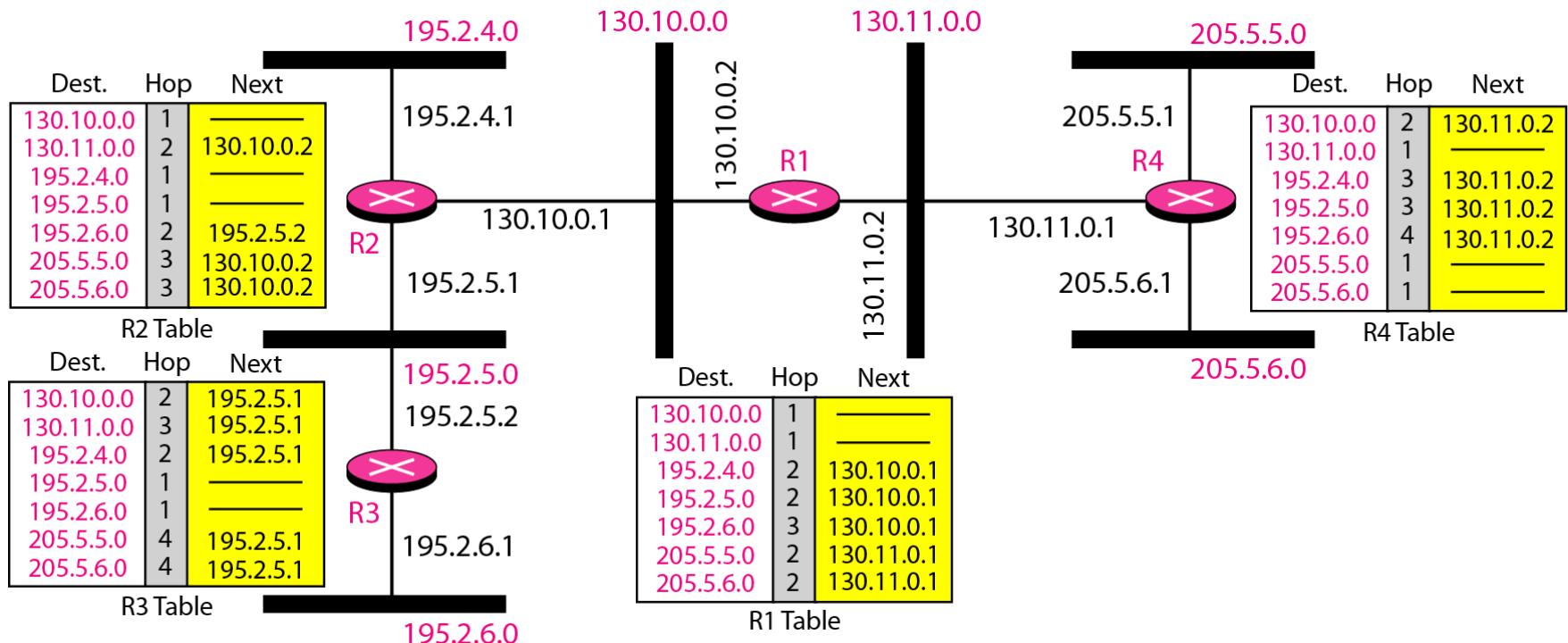
Figure 22.18 Three-node instability



Routing Information Protocol (RIP)

- *Intra-domain Routing protocol used in autonomous system → implements DV with following considerations*
 1. *Routers have routing tables; networks do not*
 2. *First column defines network address of destination network*
 3. *Next node column - address of router to which packet needs to be sent to reach the destination*
 4. *Metric: No. of networks to travel to reach the destination → hop-count*
 5. *Infinity defined as 16*

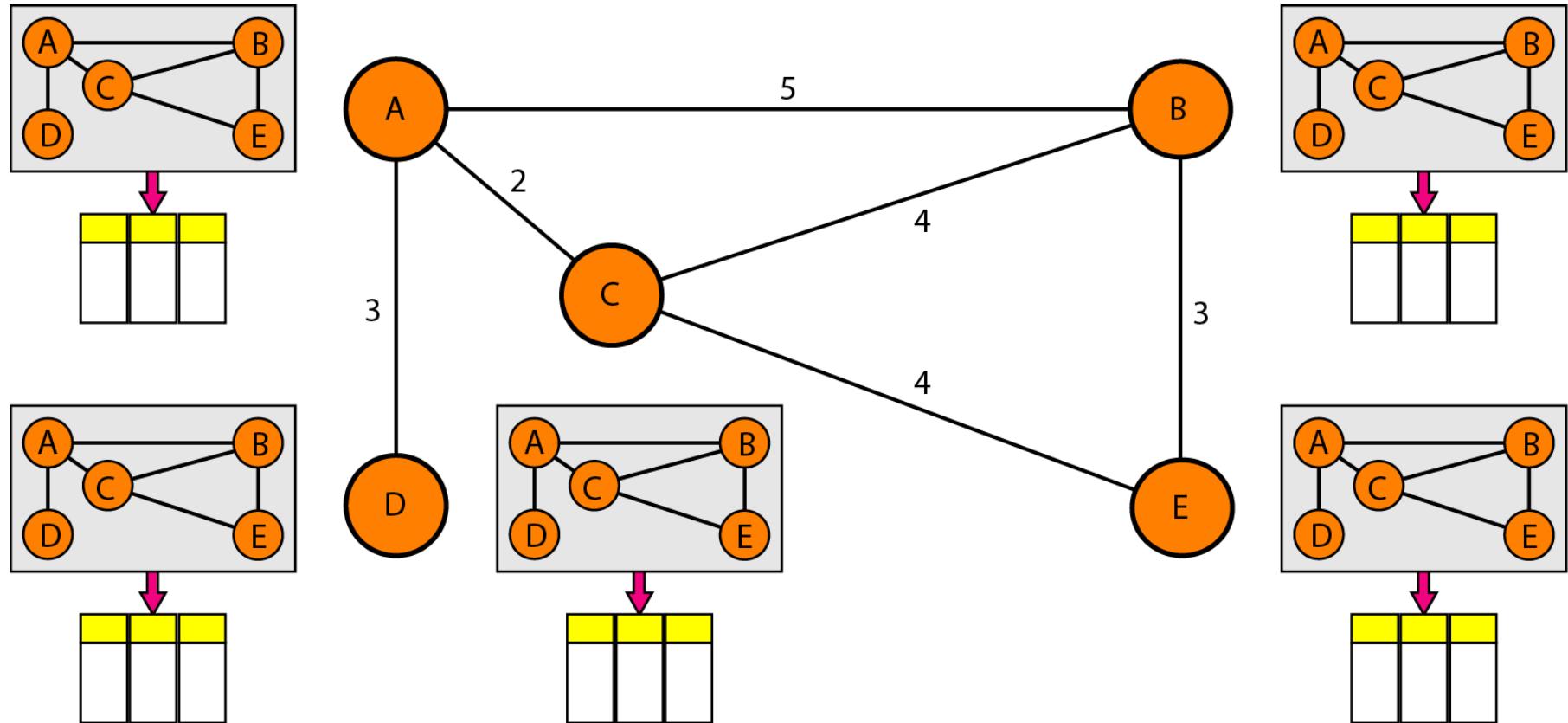
Figure 22.19 Example of a domain using RIP



Link State Routing

- *Follows the notion of state (type, condition, cost) of node/link*
- *No node has global knowledge of entire topology*
- *Each node has partial knowledge; knows the state of its (own) links*
- *Whole topology compiled from partial knowledge of each node*

Figure 22.20 Concept of link state routing

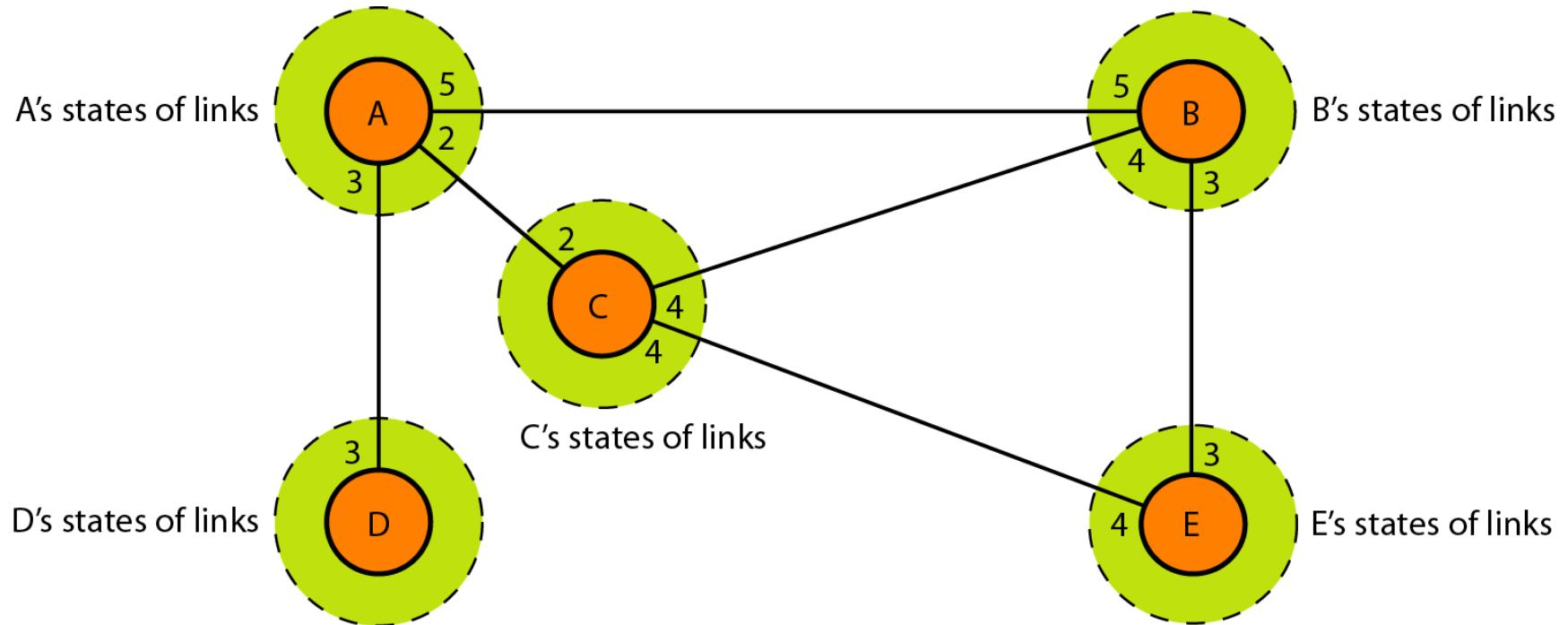


Steps in Link State Routing

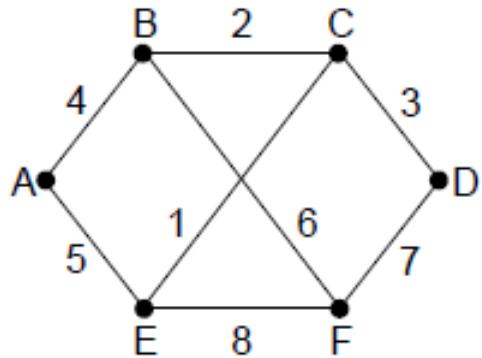
*The idea behind link state routing be stated as five parts.
Each router must do the following:*

- *Discover its neighbors and learn their network addresses*
- *Measure the delay or cost to each of its neighbors*
Construct a packet telling all it has just learned
(Creation of packet that has state of the links; LSP)
- **Send this packet to all other routers; LSP dissemination**
(Flooding in efficient and reliable way to other routers)
- *Compute the shortest path to every other router*
 - Formation of a shortest path tree for each node.
- **Calculation of a routing table based on the shortest path tree**

Figure 22.21 *Link state knowledge*



Link State Packet and Flooding



(a)

	Link	State	Packets
A			
B	Seq.		
C	Age		
D		Seq.	
E	Age	Age	
F			Seq.

The tables show the sequence and age values for each node:

- Link table:
 - A: Seq.
 - B: Age
 - C: B 4
 - D: A 4
 - E: B 5
 - F: C 2
- State table:
 - A: Age
 - B: B 4
 - C: C 2
 - D: D 3
 - E: F 6
 - F: E 1
- Packets table:
 - A: C 3
 - B: F 7
 - C: A 5
 - D: C 1
 - E: F 8
 - F: E 8

(b)

Figure 22.22 Dijkstra algorithm

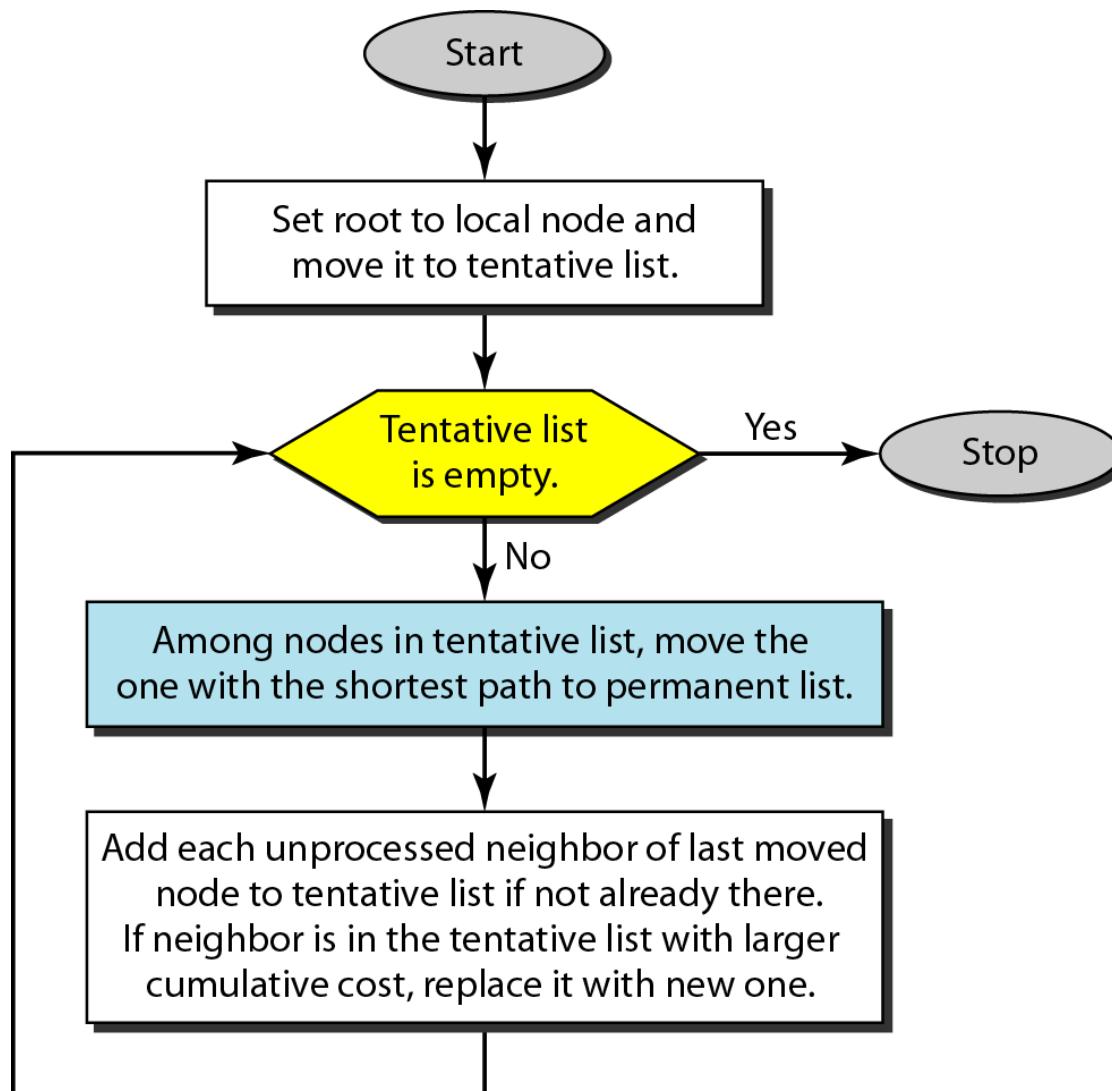


Figure 22.23 Example of formation of shortest path tree

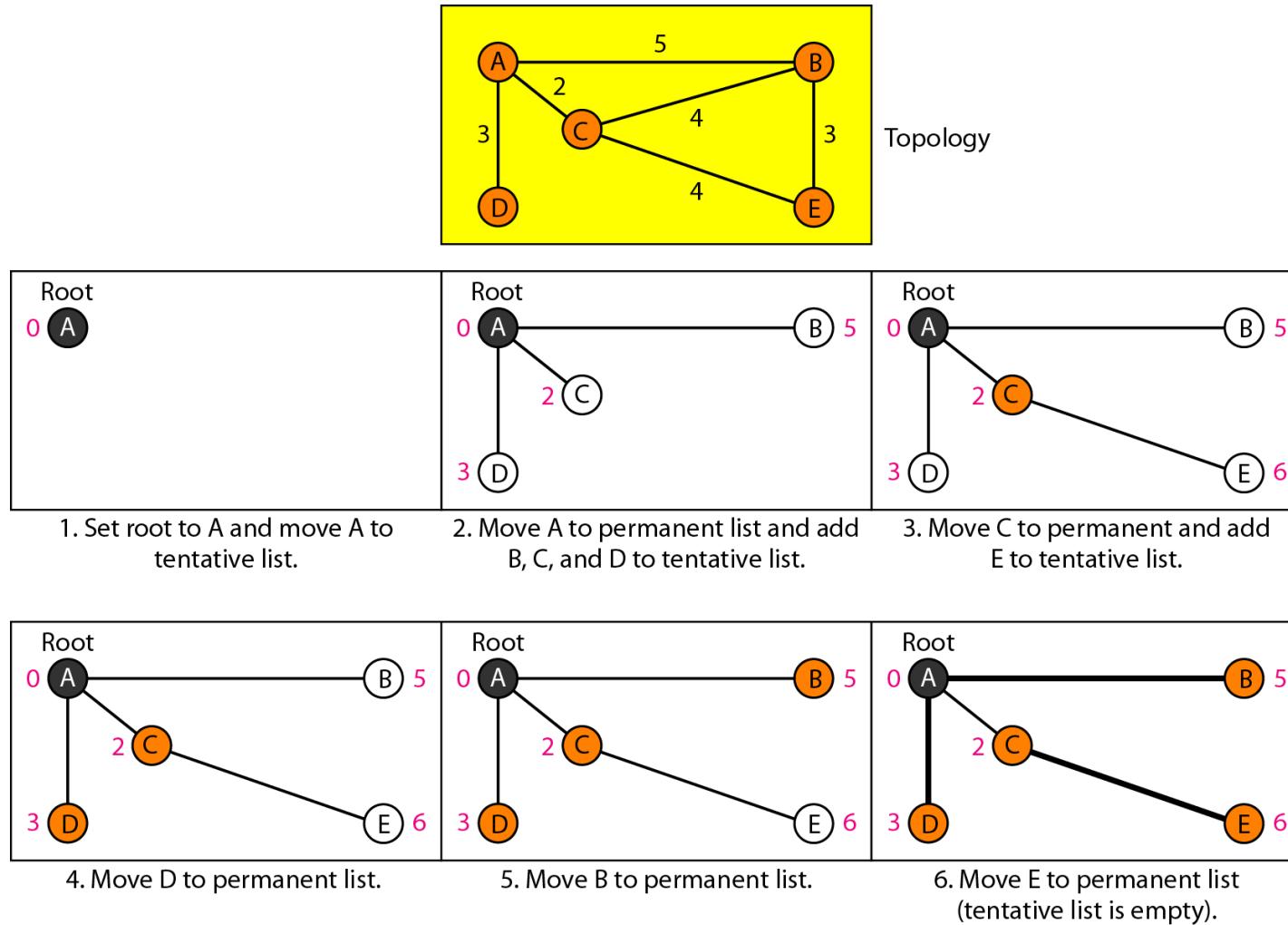


Table 22.2 *Routing table for node A*

<i>Node</i>	<i>Cost</i>	<i>Next Router</i>
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

OSPF: Figure 22.24 Areas in an autonomous system

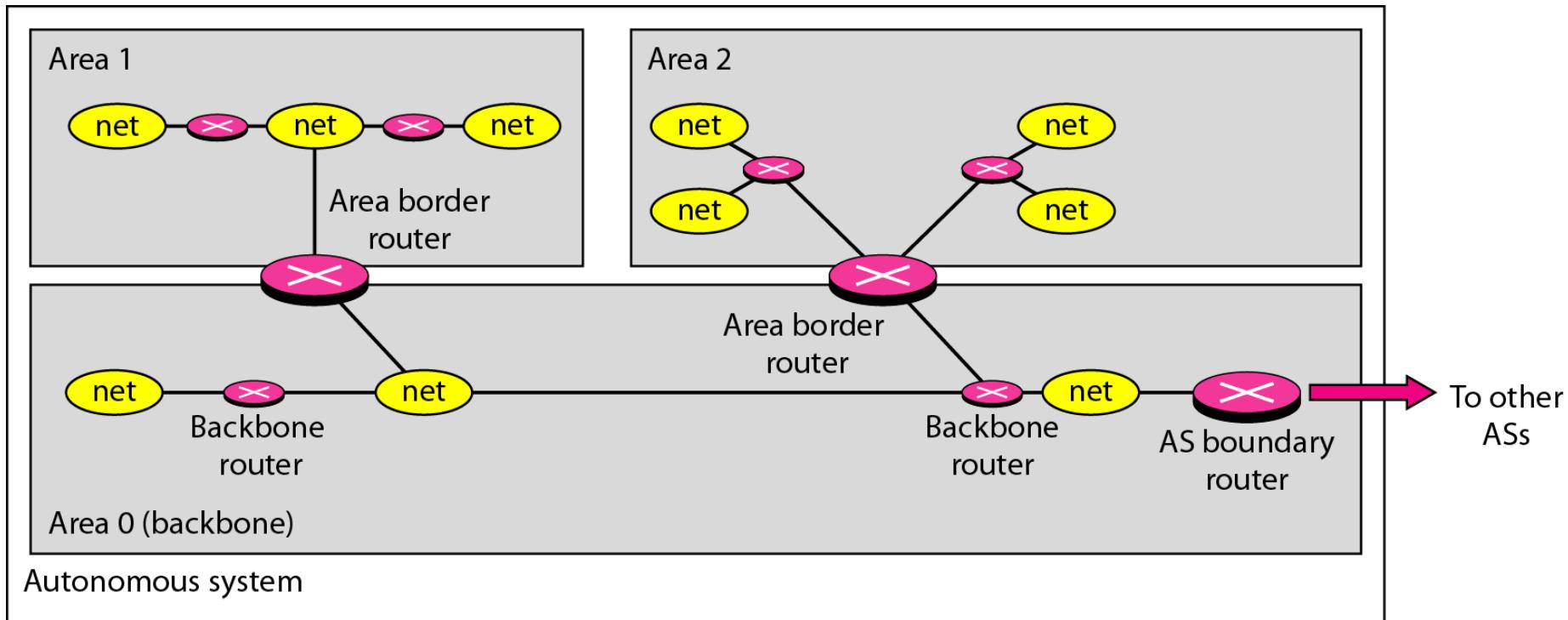


Figure 22.25 *Types of links*

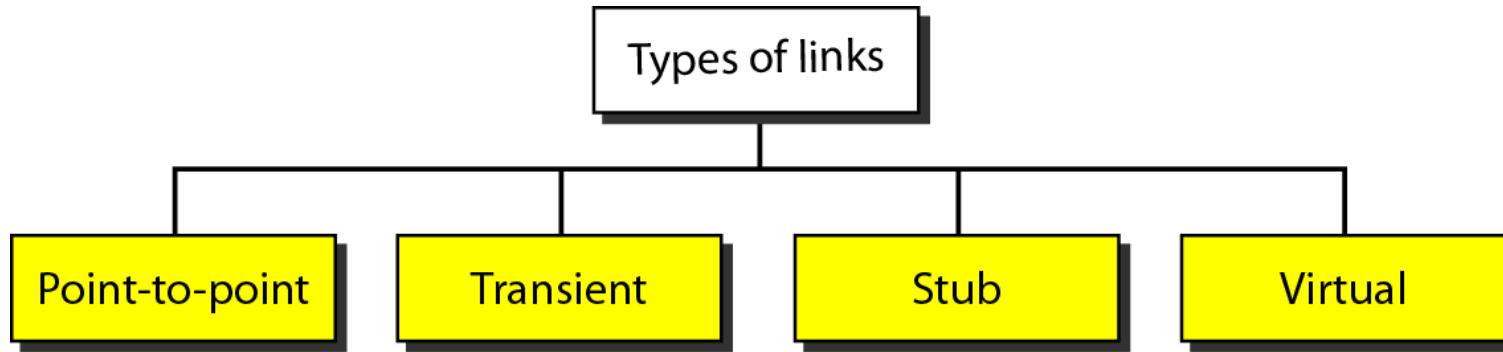


Figure 22.26 *Point-to-point link*

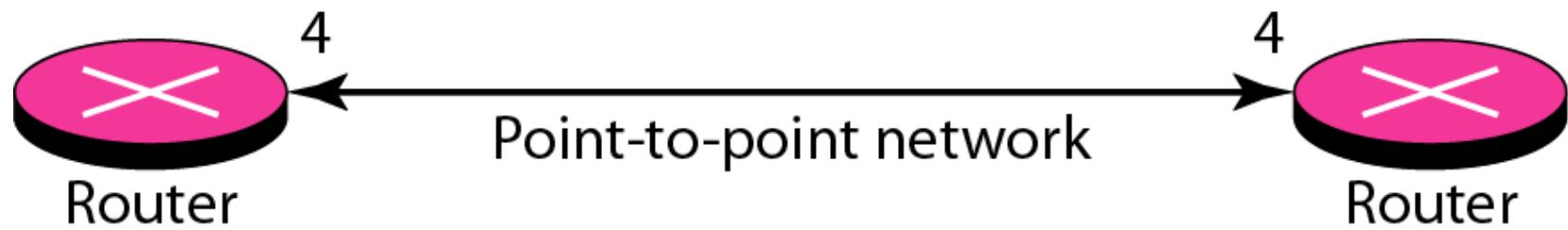


Figure 22.27 Transient link

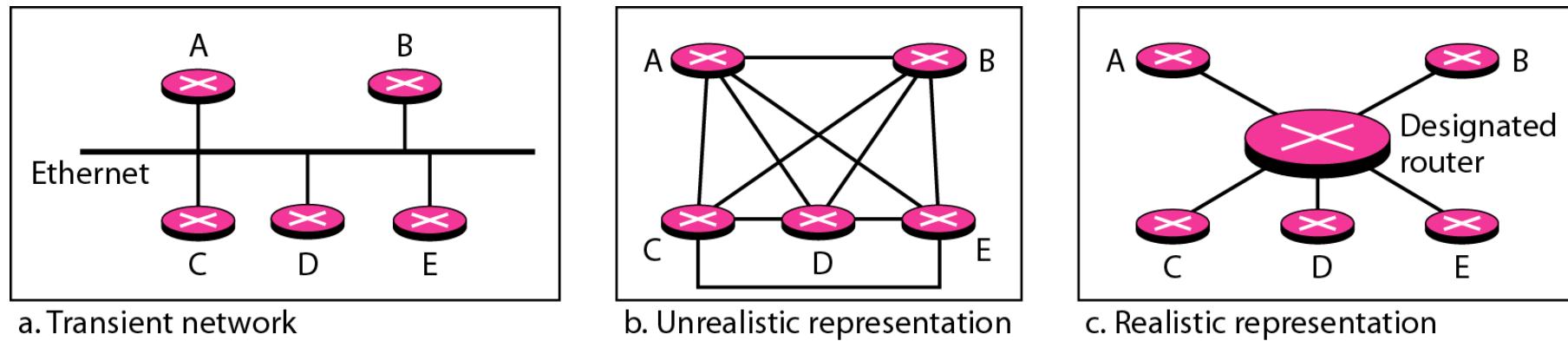
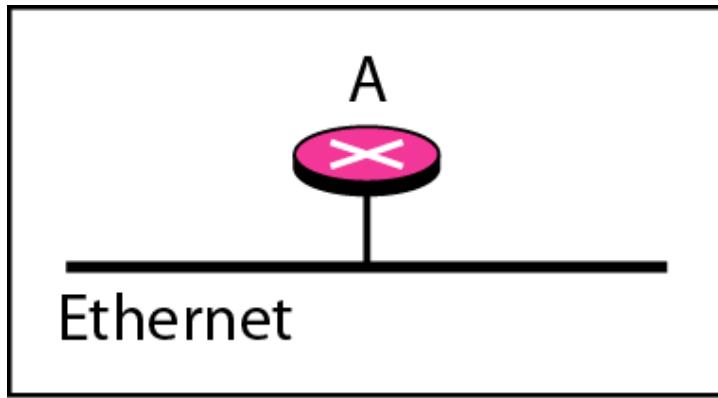


Figure 22.28 *Stub link*

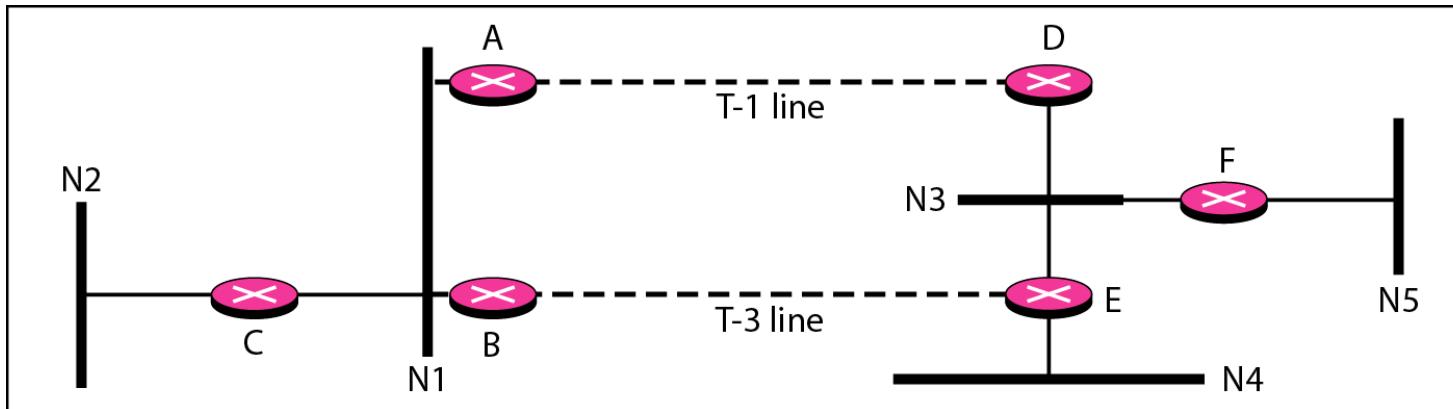


a. Stub network

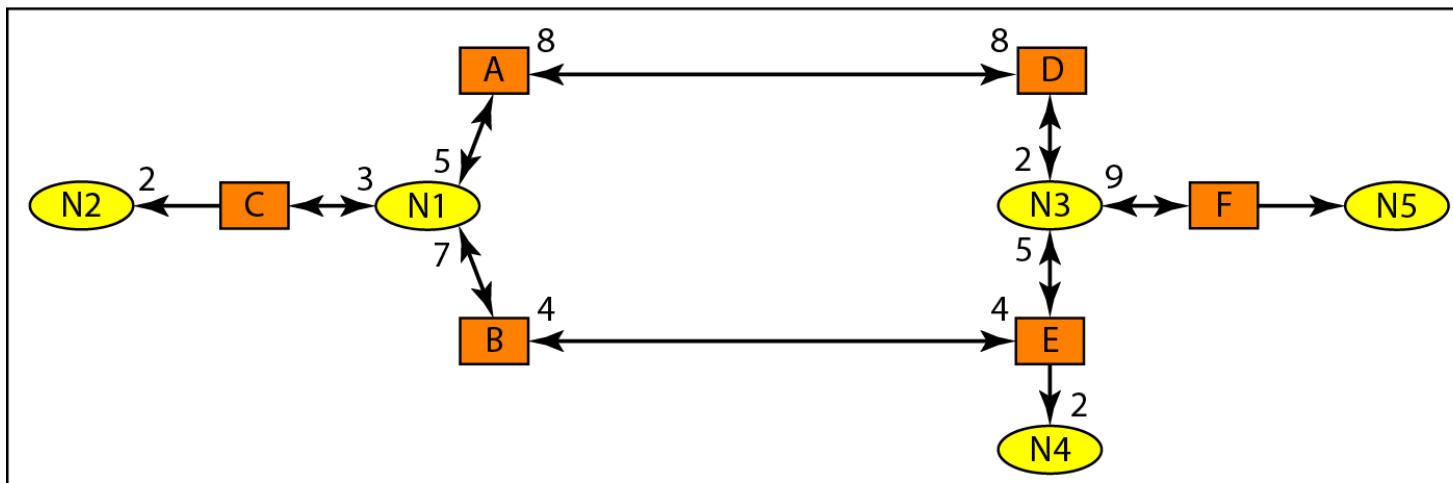


b. Representation

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



a. Autonomous system



b. Graphical representation

Figure 22.30 Initial routing tables in path vector routing

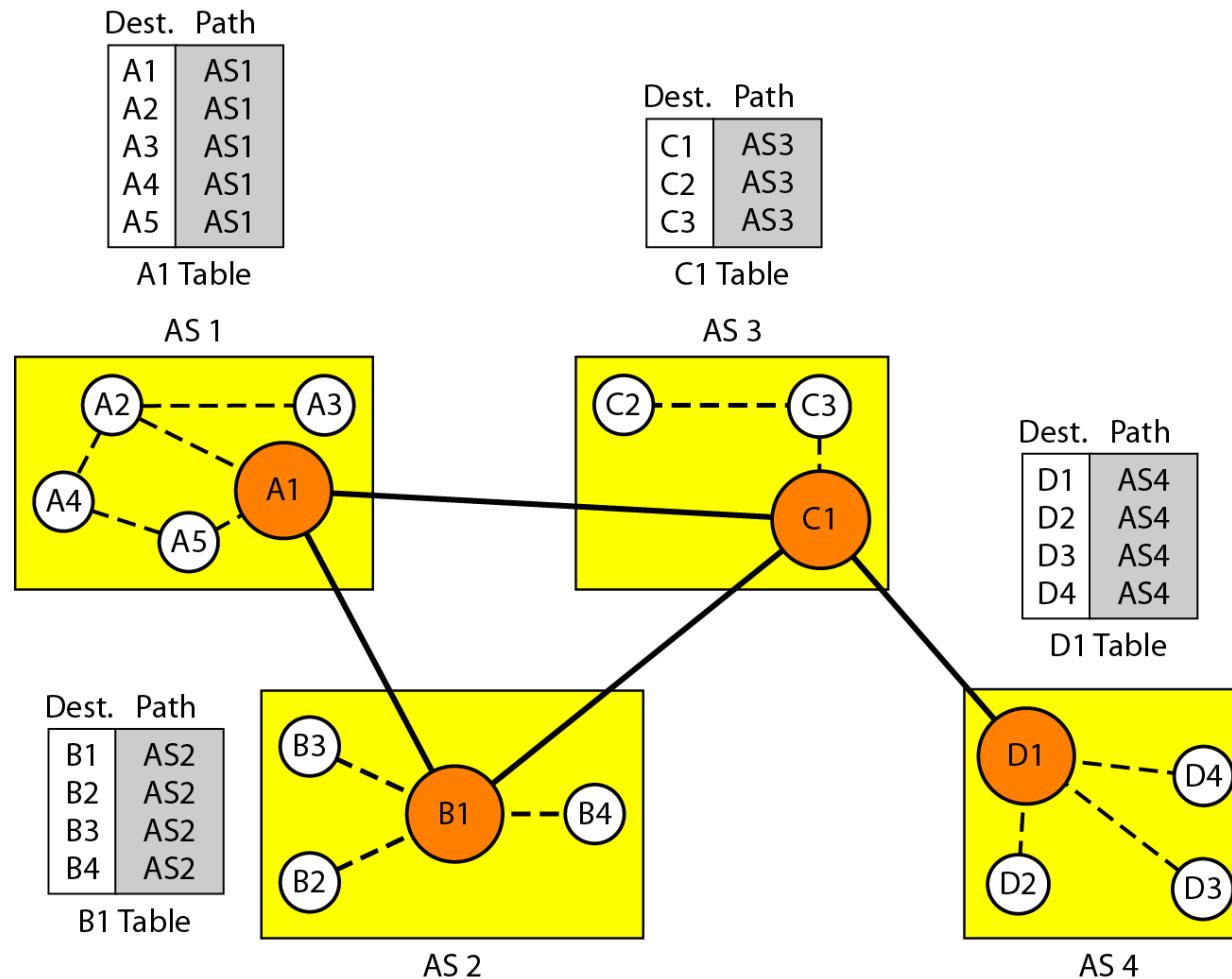


Figure 22.31 *Stabilized tables for three autonomous systems*

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

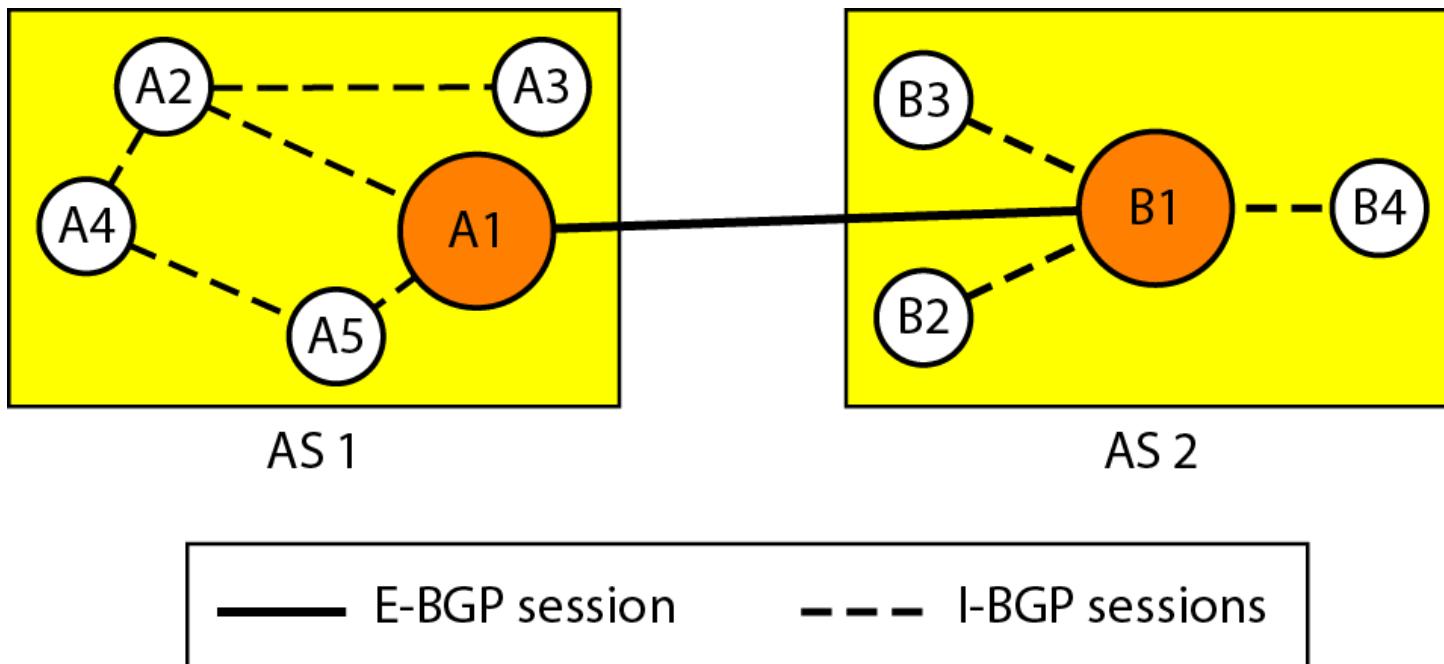
Dest.	Path
A1	AS3-AS1
...	
A5	AS3-AS1
B1	AS3-AS2
...	
B4	AS3-AS2
C1	AS3
...	
C3	AS3
D1	AS3-AS4
...	
D4	AS3-AS4

C1 Table

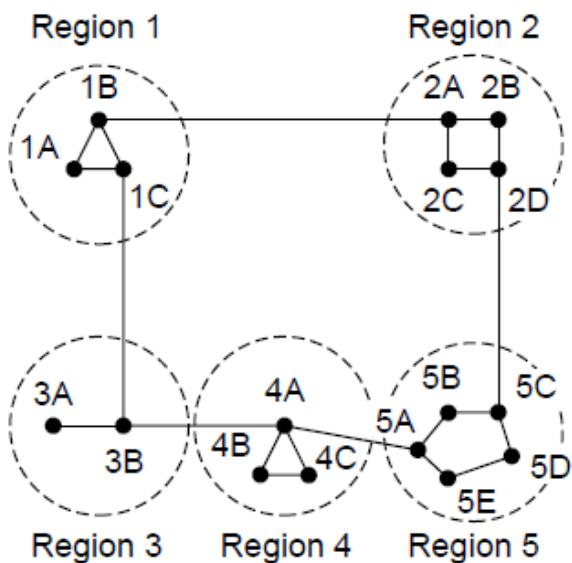
Dest.	Path
A1	AS4-AS3-AS1
...	
A5	AS4-AS3-AS1
B1	AS4-AS3-AS2
...	
B4	AS4-AS3-AS2
C1	AS4-AS3
...	
C3	AS4-AS3
D1	AS4
...	
D4	AS4

D1 Table

Figure 22.32 Internal and external BGP sessions



Hierarchical Routing



(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

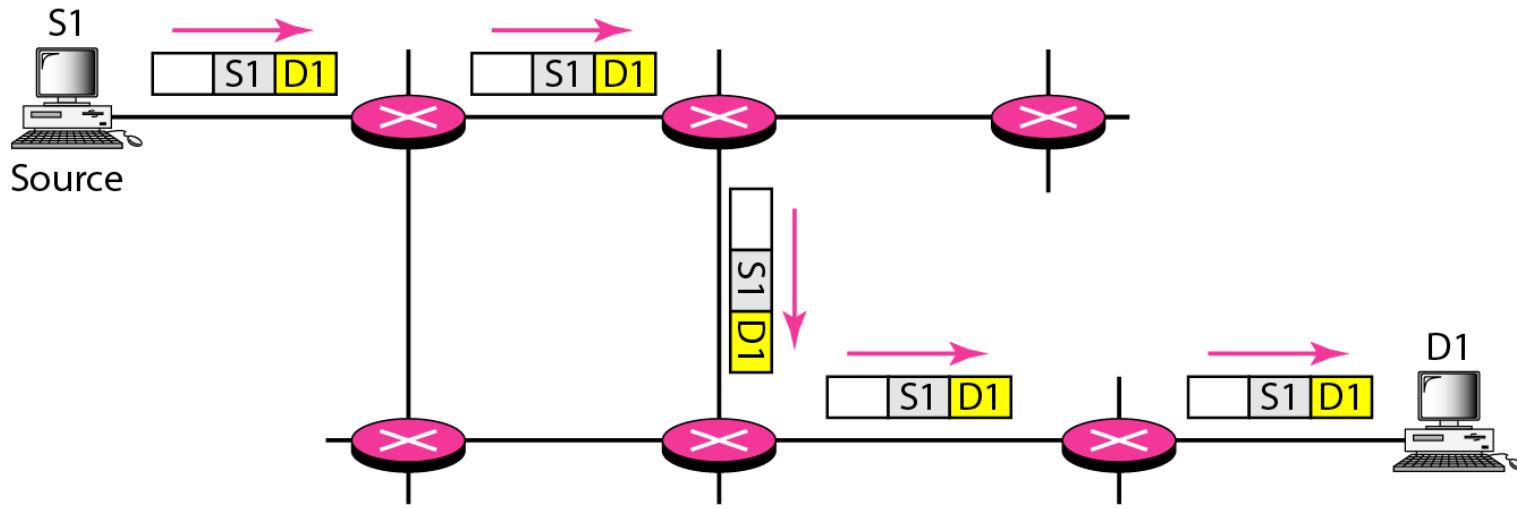
(b)

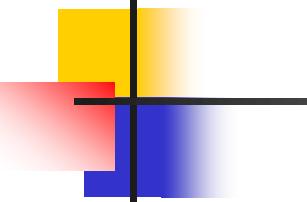
Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Figure 22.33 Unicasting

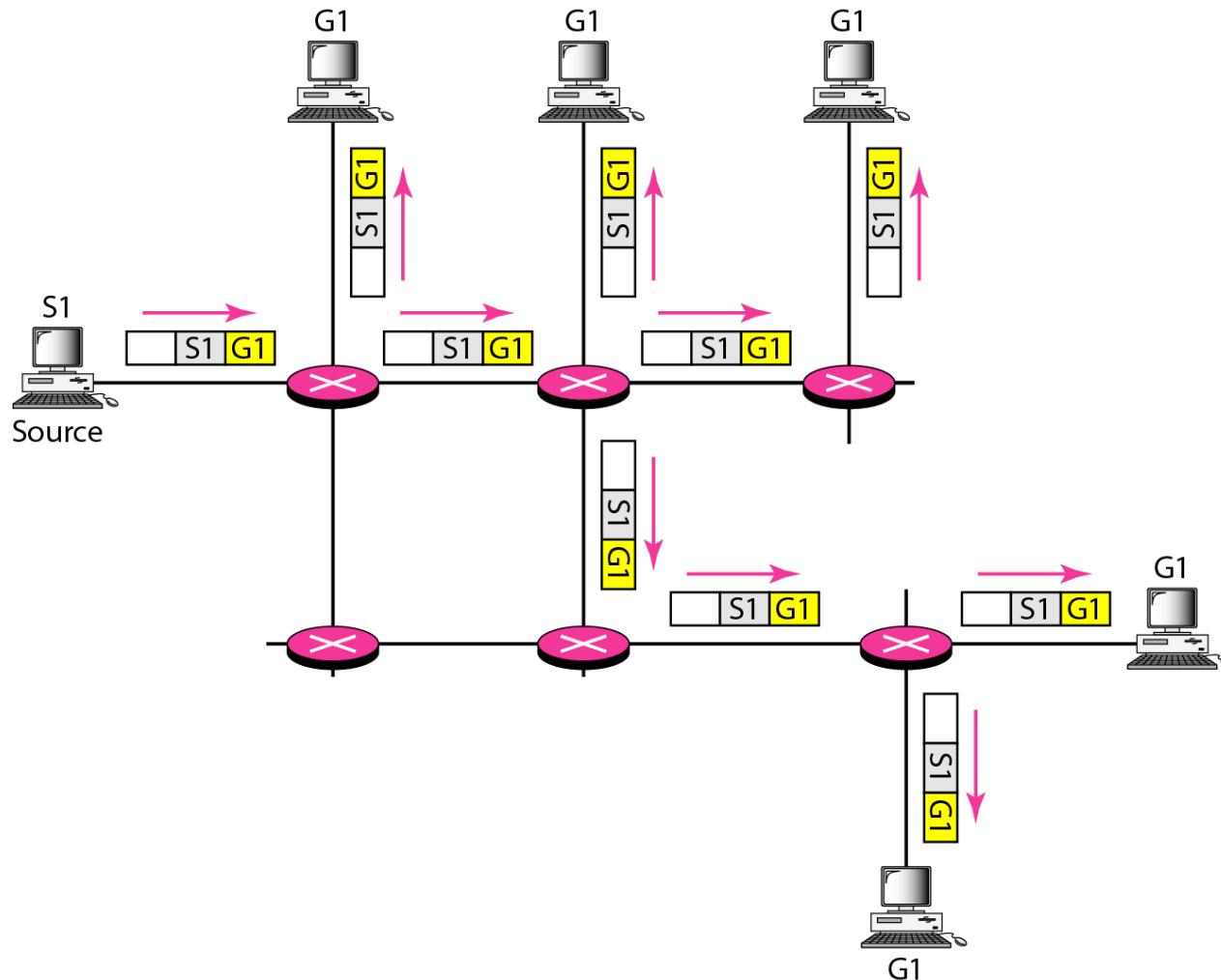


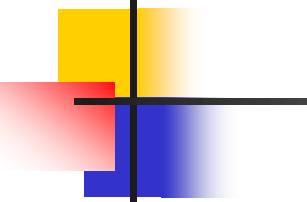


Note

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

Figure 22.34 Multicasting

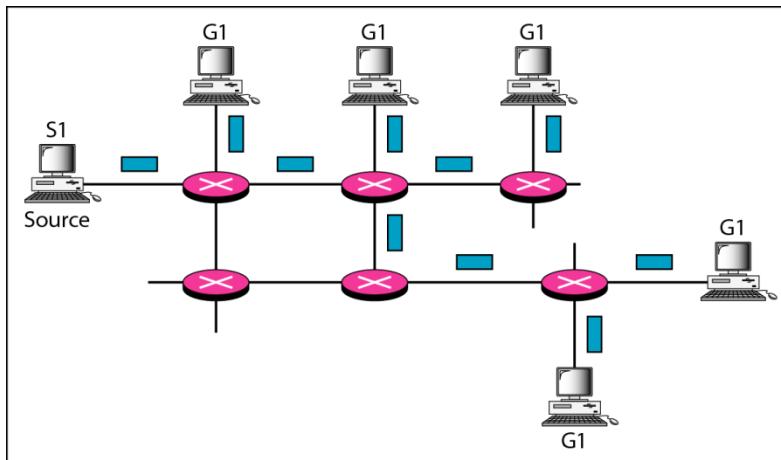




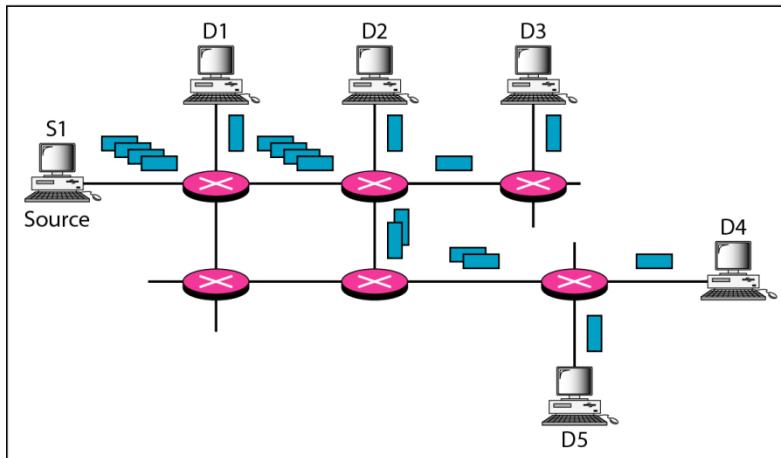
Note

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

Figure 22.35 Multicasting versus multiple unicasting



a. Multicasting



b. Multiple unicasting

22-4 MULTICAST ROUTING PROTOCOLS

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

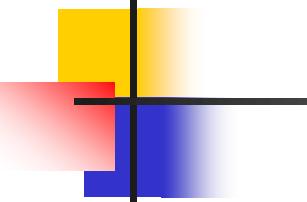
Topics discussed in this section:

Unicast, Multicast, and Broadcast

Applications

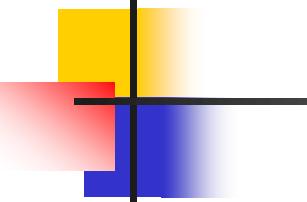
Multicast Routing

Routing Protocols



Note

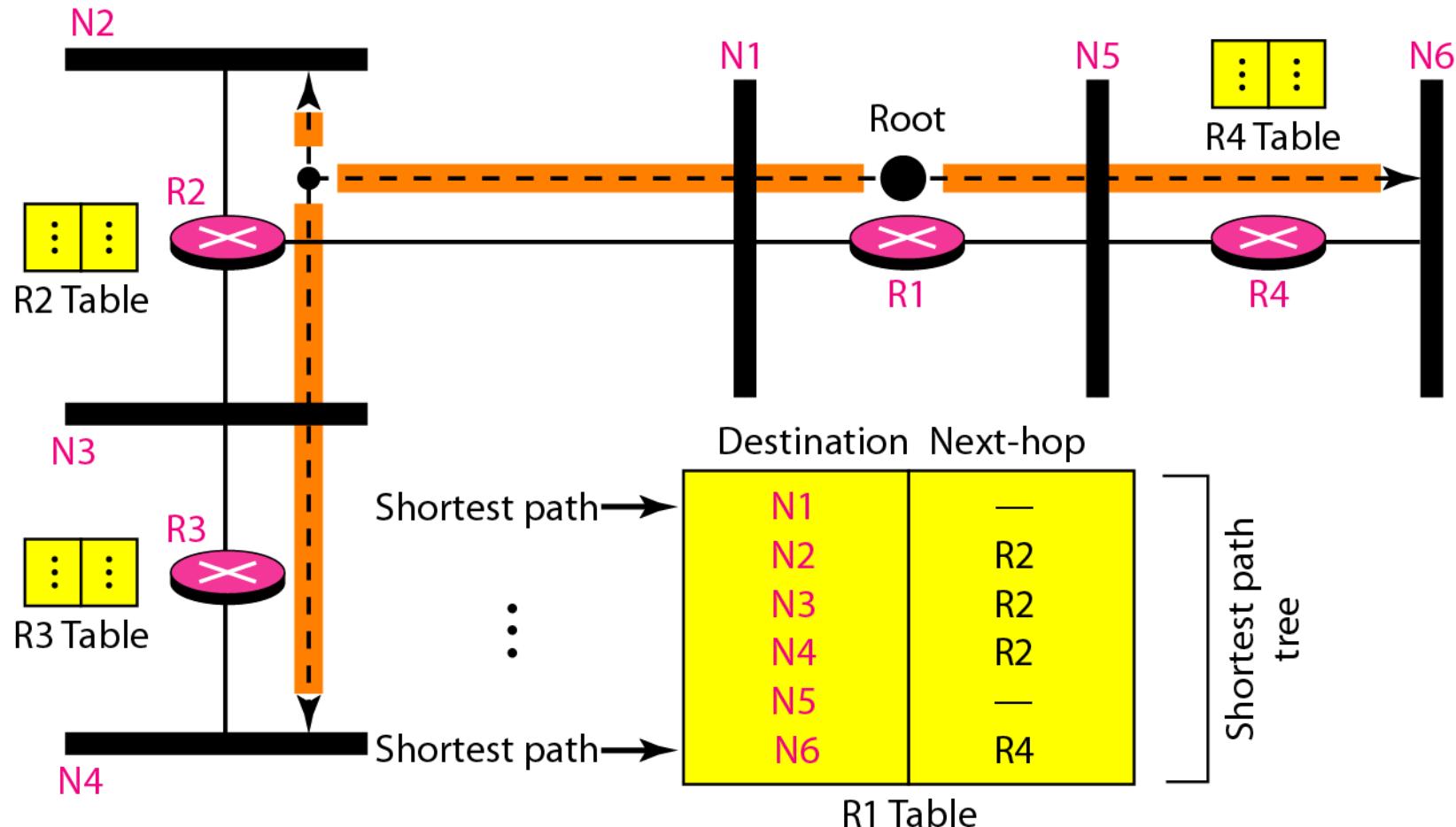
Emulation of multicasting through multiple unicasting is not efficient and may create long delays, particularly with a large group.

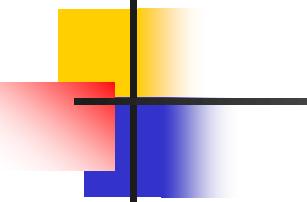


Note

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

Figure 22.36 Shortest path tree in unicast routing



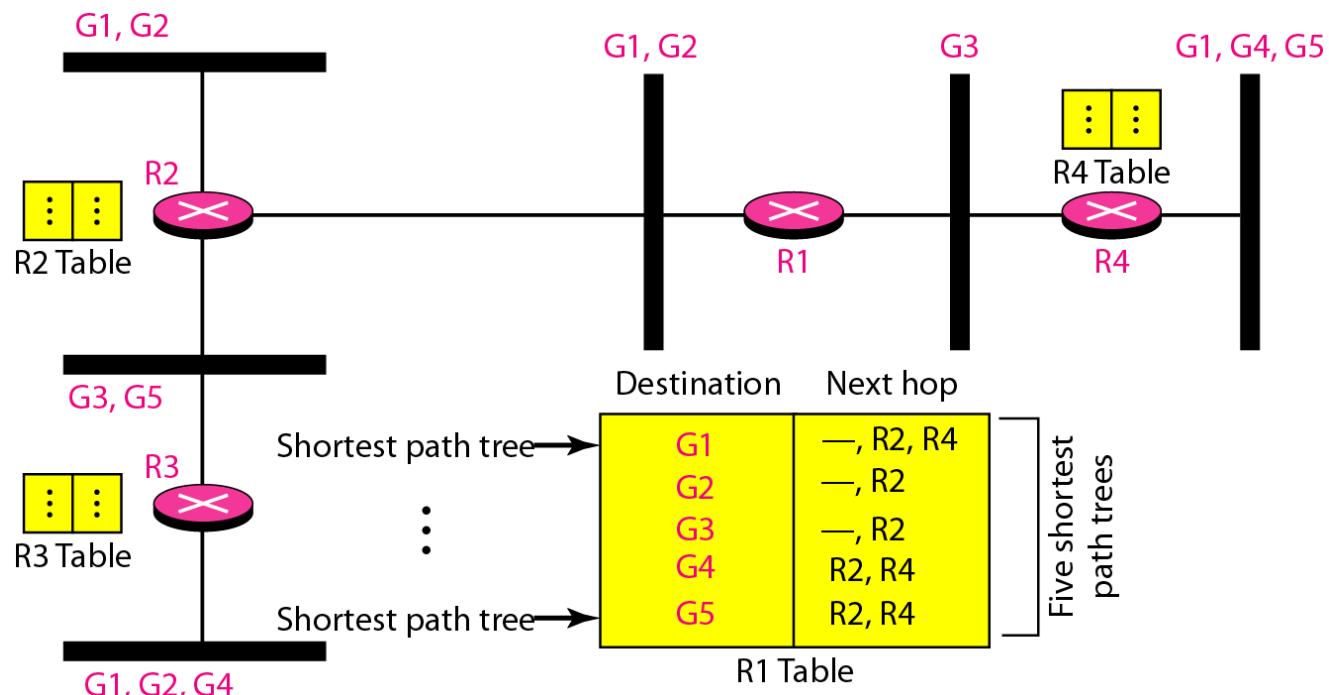


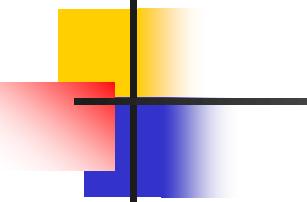
Note

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

Figure 22.37 Source-based tree approach

- Each Router needs to have one shortest path tree for each group
- Shortest path tree for a group defines the next hop for each network that has loyal members for that group





Note

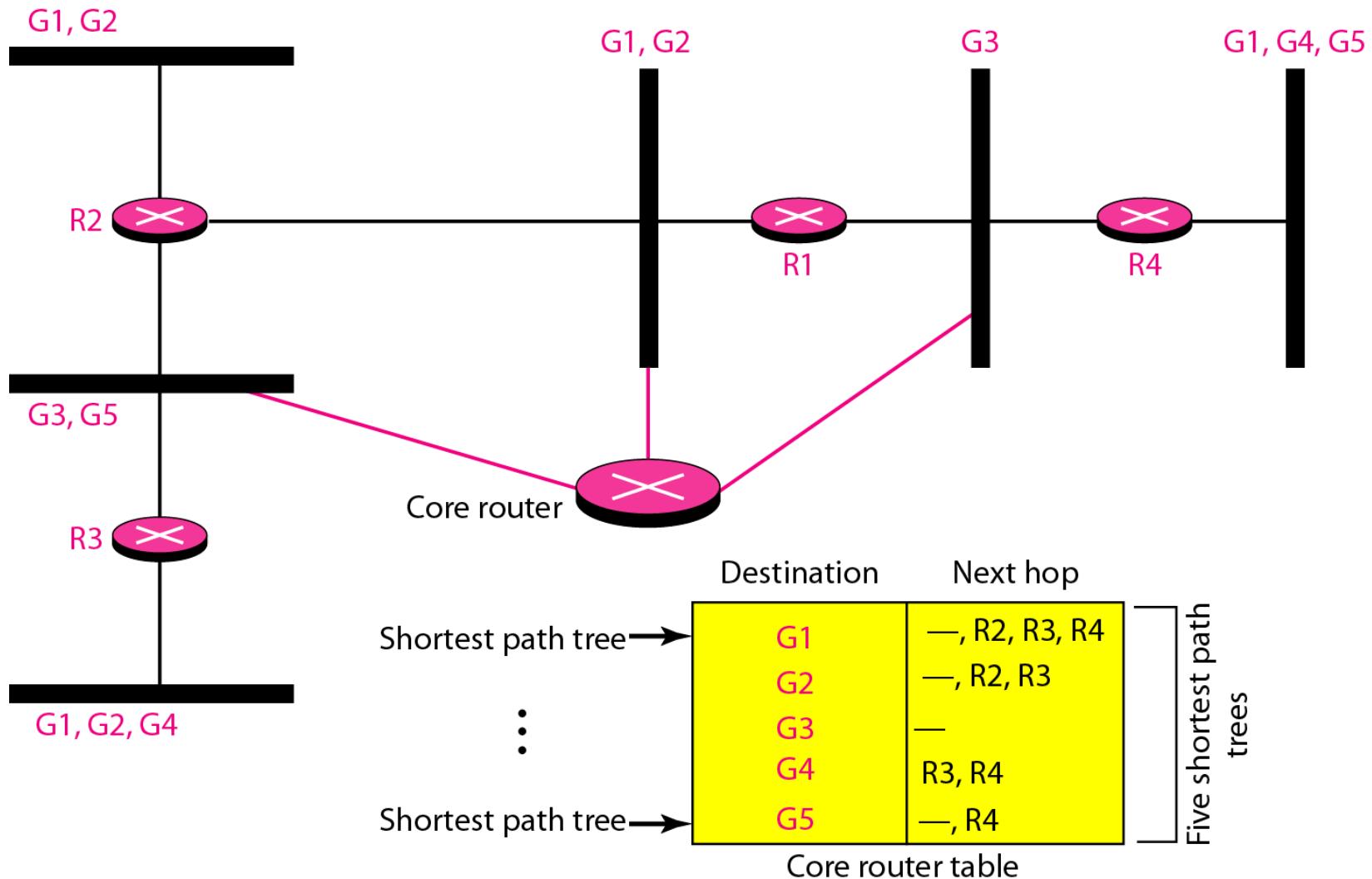
In the source-based tree approach, each router needs to have one shortest path tree for each group

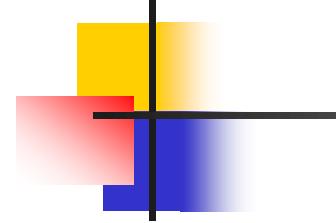
- If no. of groups = m ; each router needs to have m shortest path trees**

Group-Shared Tree approach

- In Group-shared tree approach, instead of each router having m shortest path trees, only 1 designated router → the **center core or rendezvous router** responsible for distributing multicast traffic
- It has m shortest path trees in its routing table; rest of the routers have none
- If a router receives a multicast pkt, it encapsulates it into unicast pkt and sends to core
 - Core router de-capsulates and removes the multicast pkt; consults its routing table to route it

Figure 22.38 Group-shared tree approach

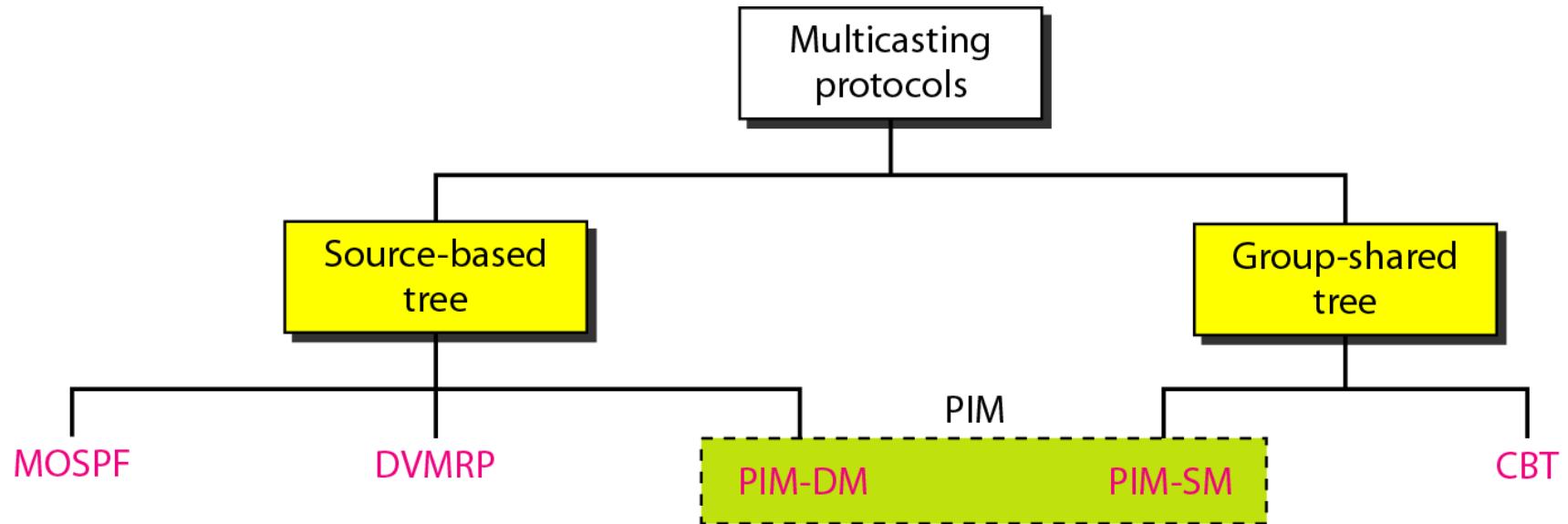


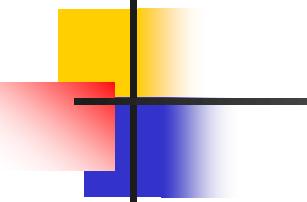


Note

In the group-shared tree approach, only the core router, which has a shortest path tree for each group, is involved in multicasting.

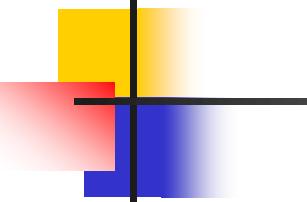
Figure 22.39 *Taxonomy of common multicast protocols*





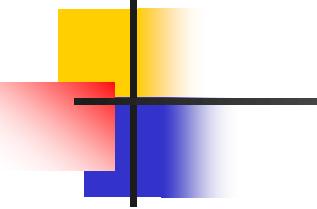
Note

DVMRP
**Multicast distance vector 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

Only one copy is forwarded (only the one that travelled the shortest path); others are dropped

Figure 22.40 Reverse path forwarding (RPF)

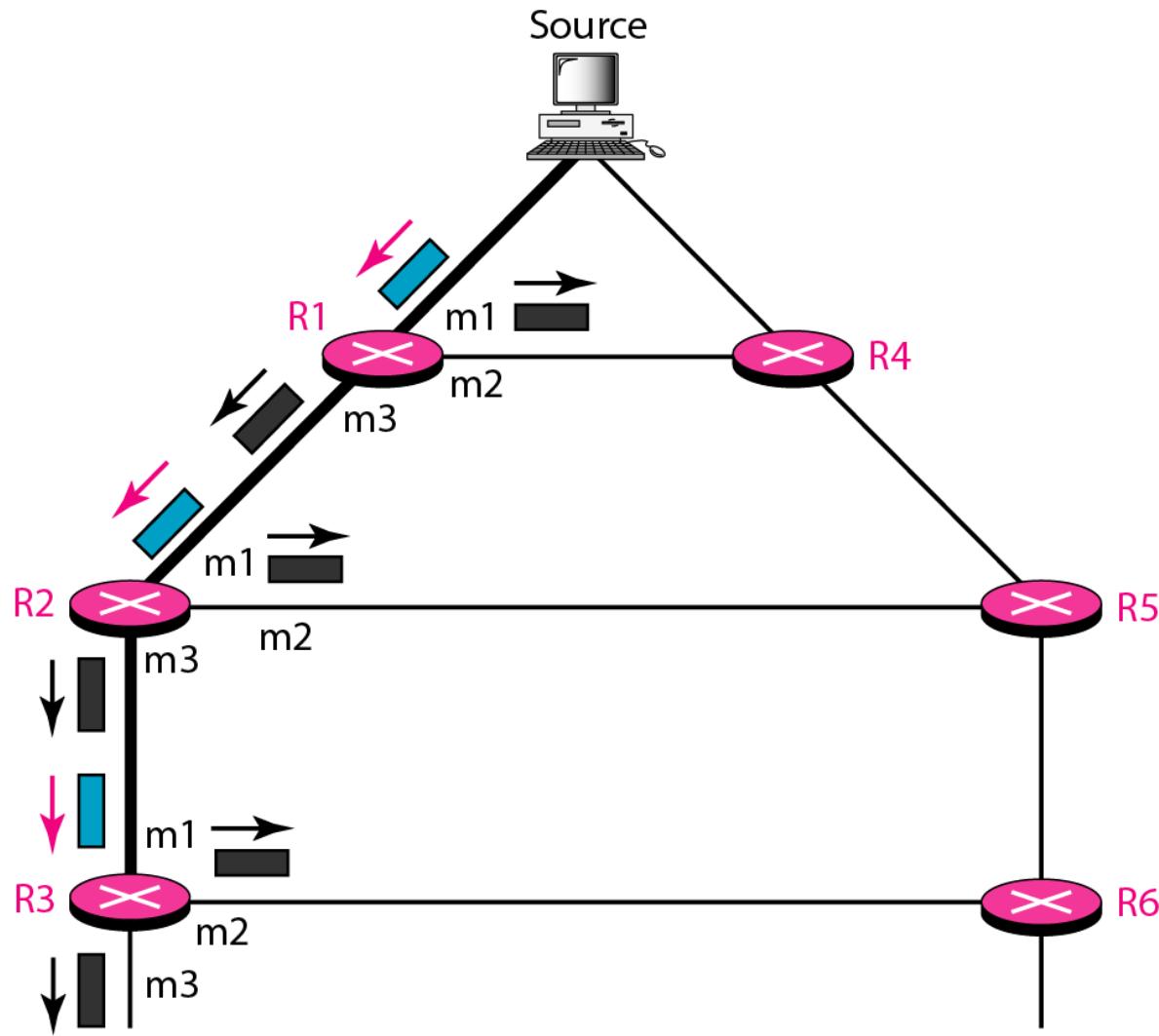


Figure 22.41 Problem with RPF

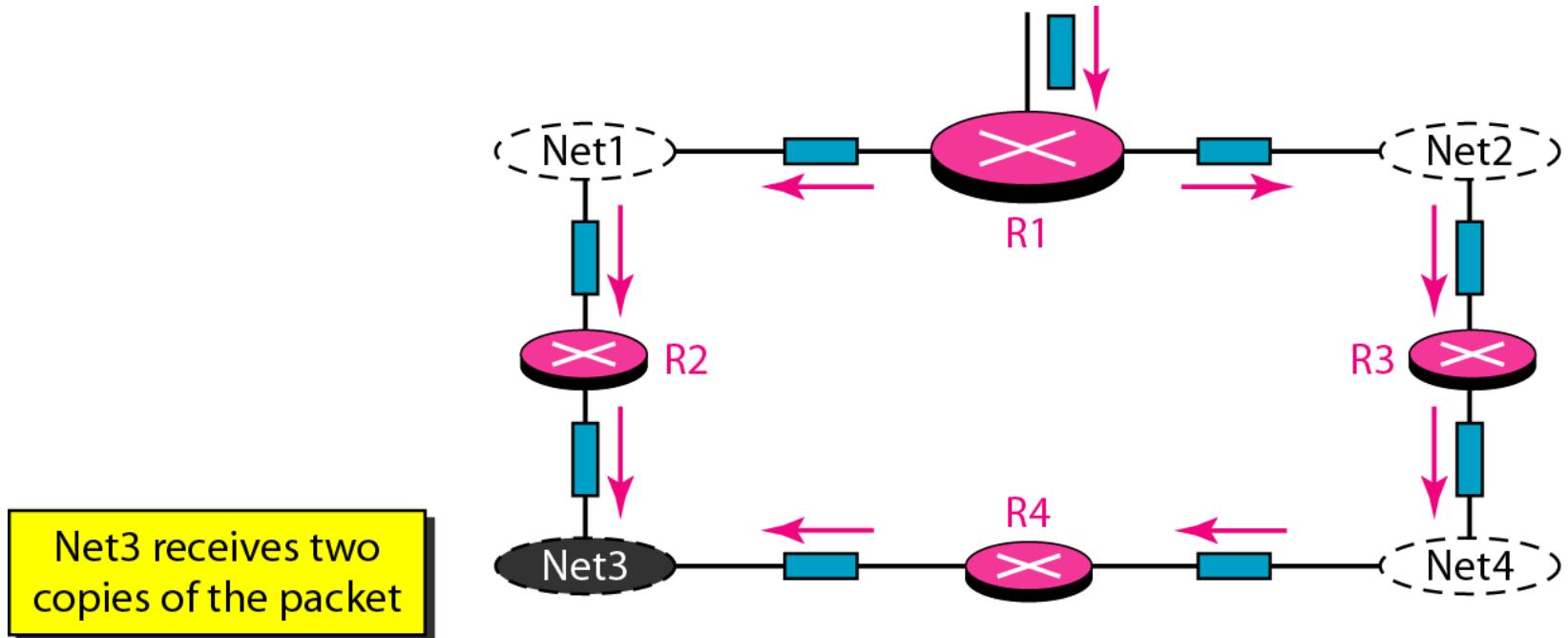
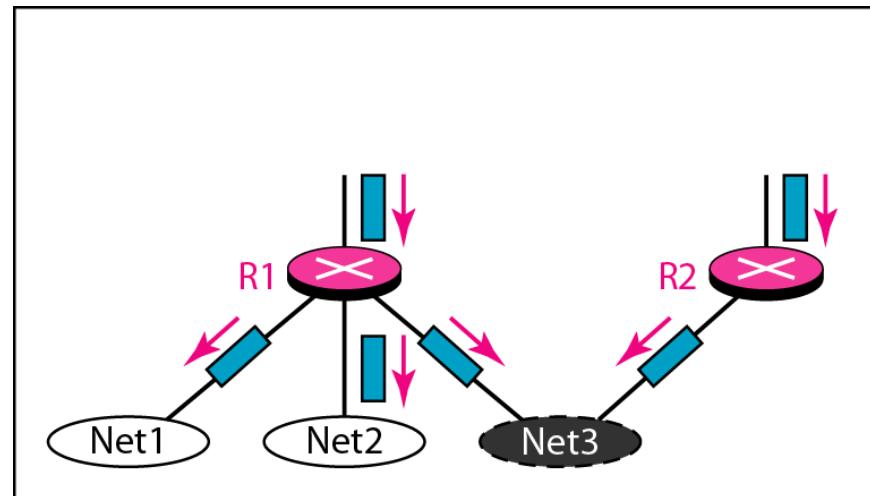
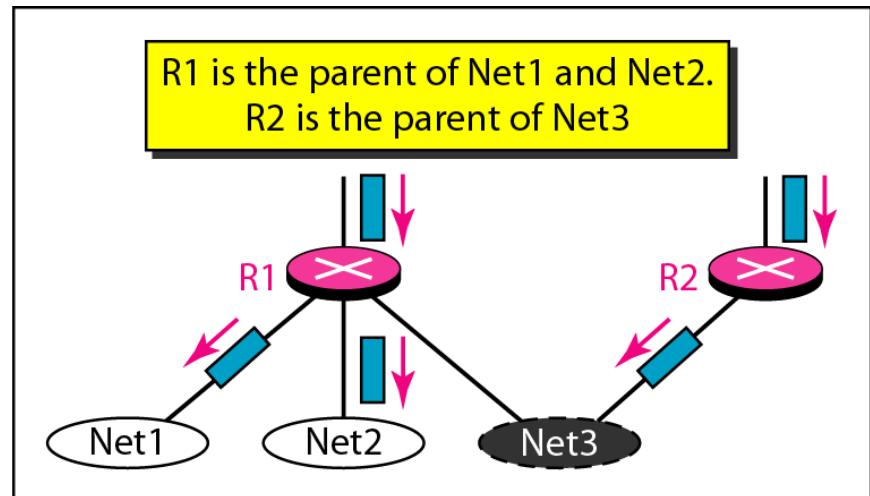


Figure 22.42 RPF Versus RPB

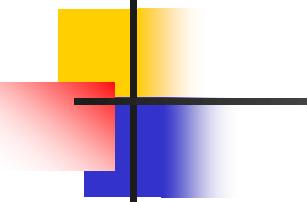


a. RPF



b. RPB

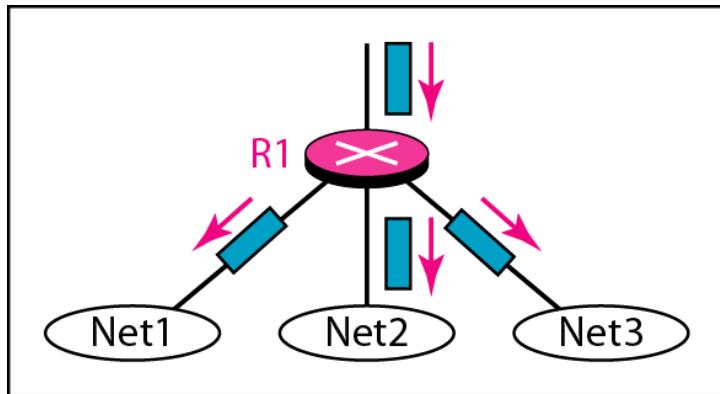
R1 is the parent of Net1 and Net2.
R2 is the parent of Net3



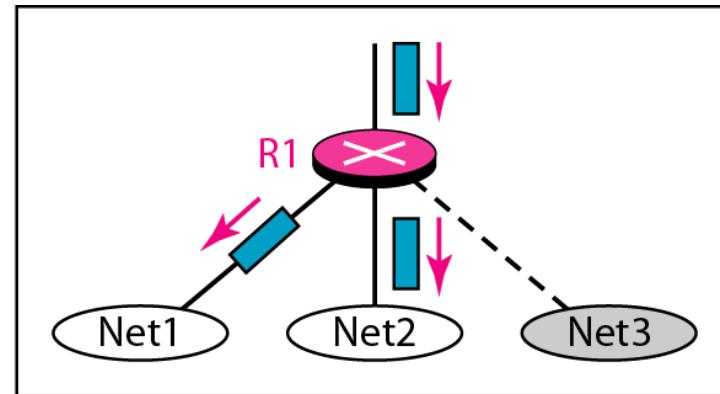
Note

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.

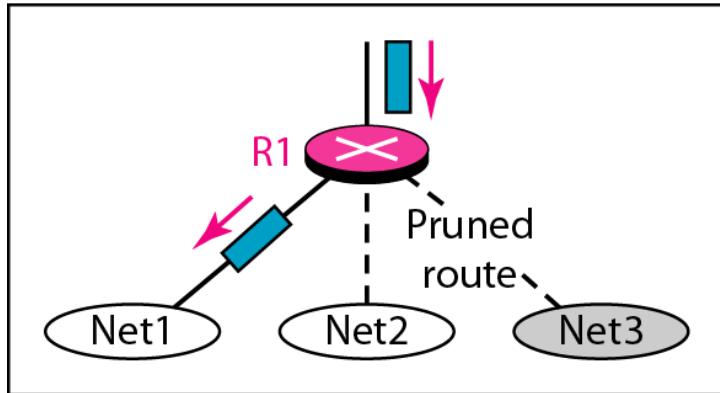
Figure 22.43 RPF, RPB, and RPM



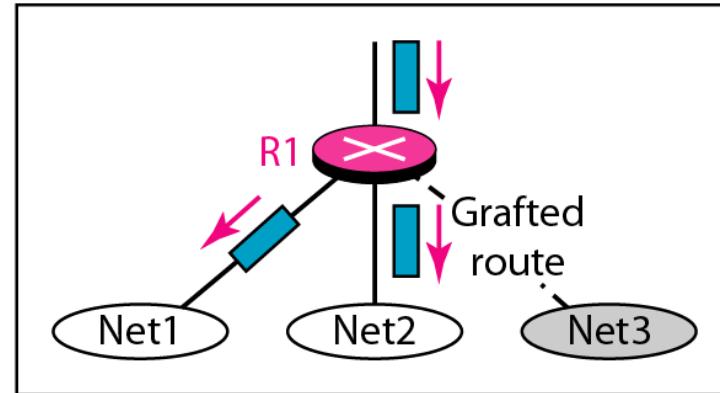
a. RPF



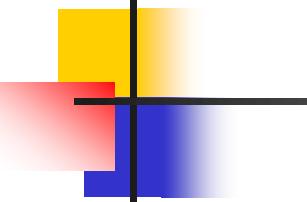
b. RPB



c. RPM (after pruning)



d. RPM (after grafting)



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

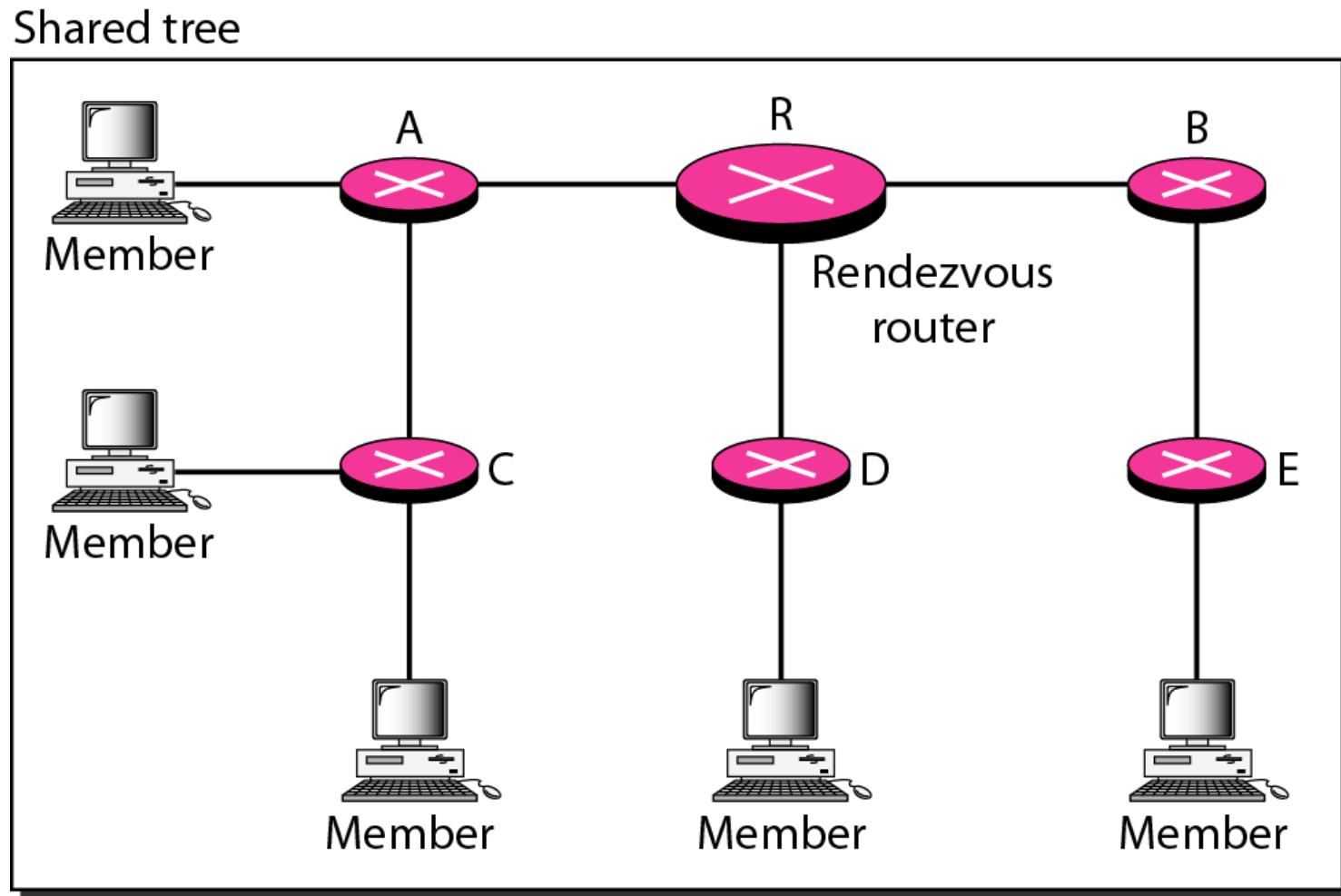
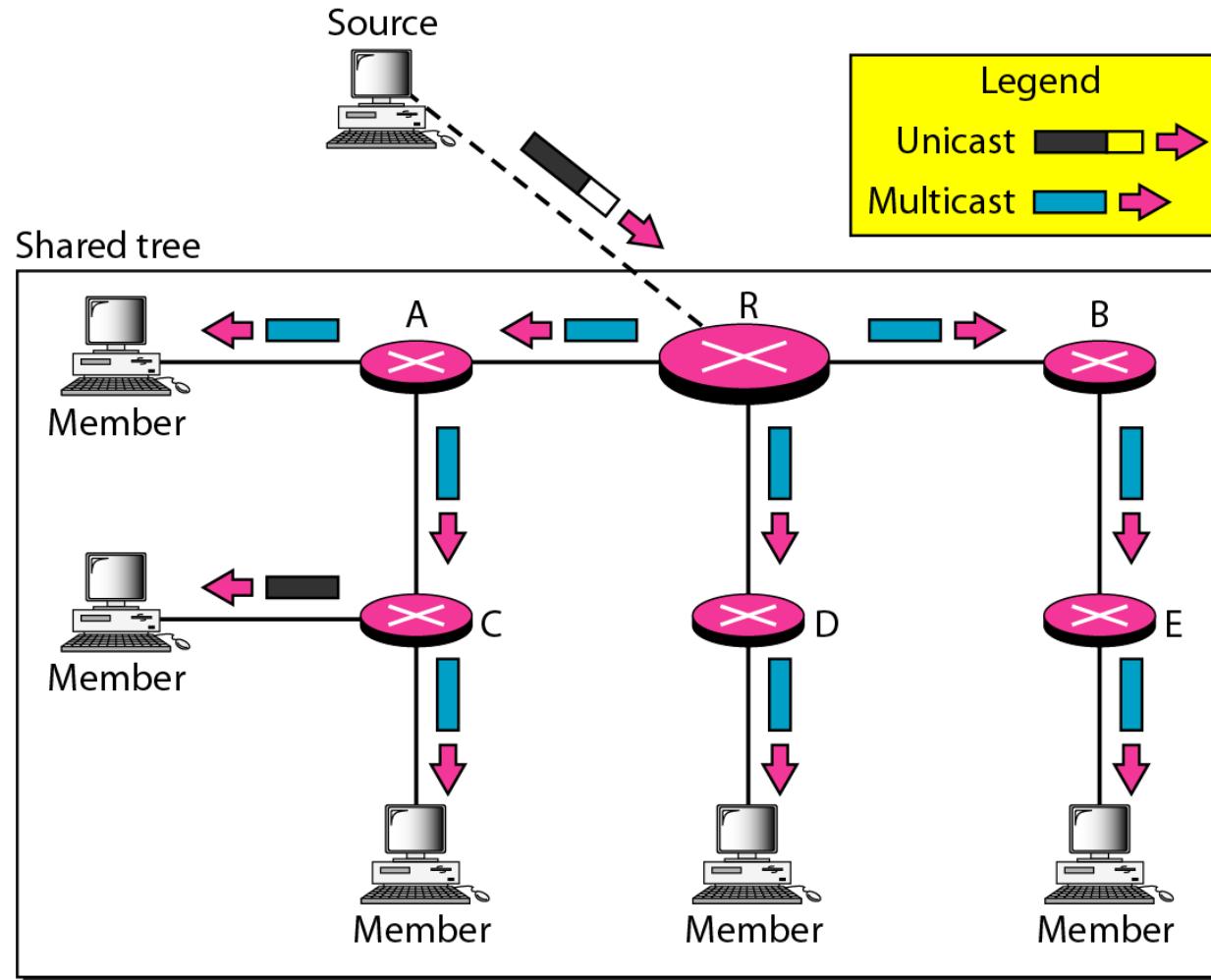
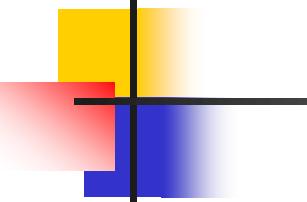


Figure 22.45 *Sending a multicast packet to the rendezvous router*





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.

Module 3

Network Layer: Address Mapping, Error Reporting, and Multicasting

21-1 ADDRESS MAPPING

*The delivery of a packet to a host or a router requires two levels of addressing: **logical** and **physical**. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.*

Topics discussed in this section:

Mapping Logical to Physical Address

Mapping Physical to Logical Address

Figure 21.1 ARP operation

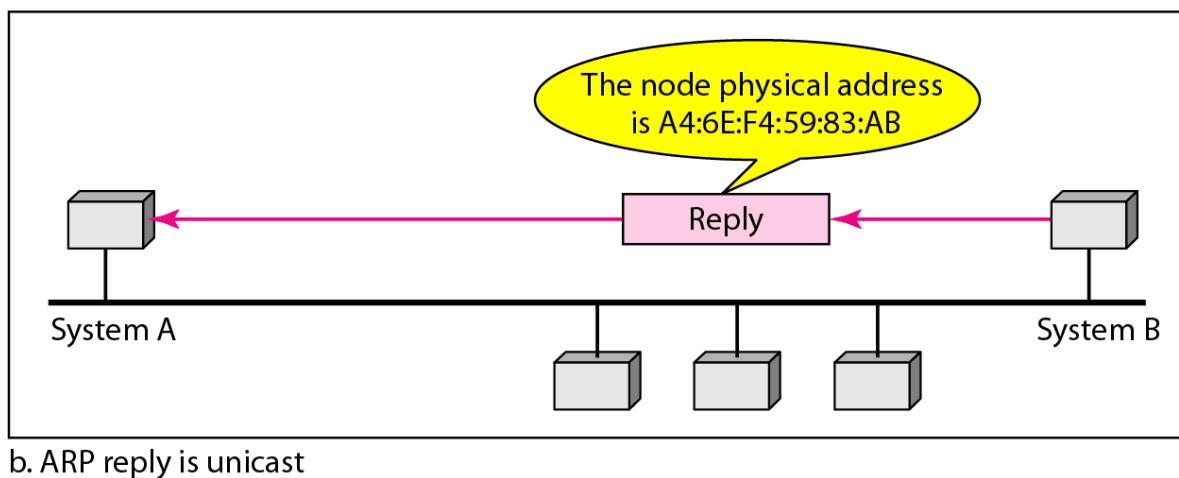
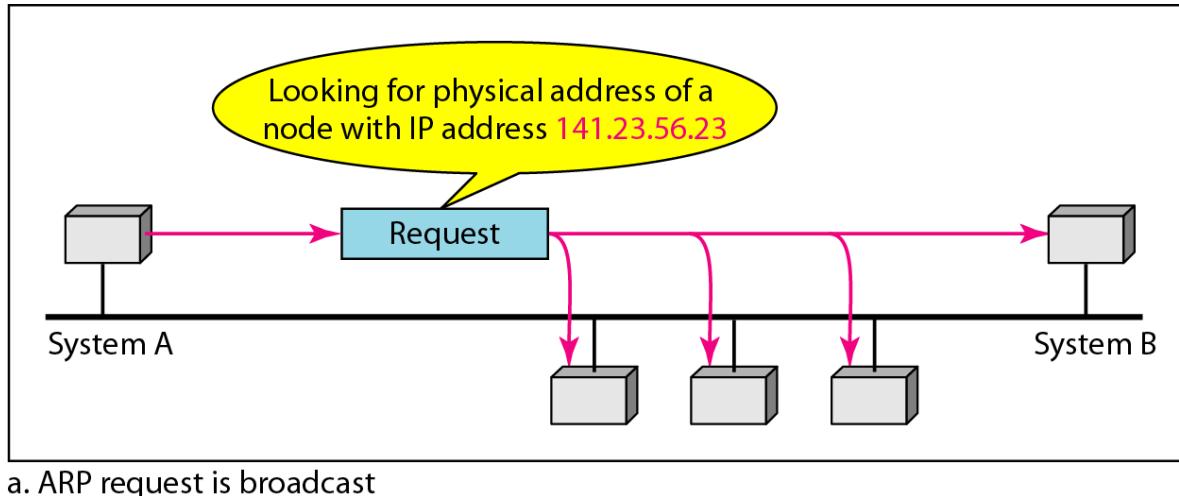


Figure 21.2 ARP packet

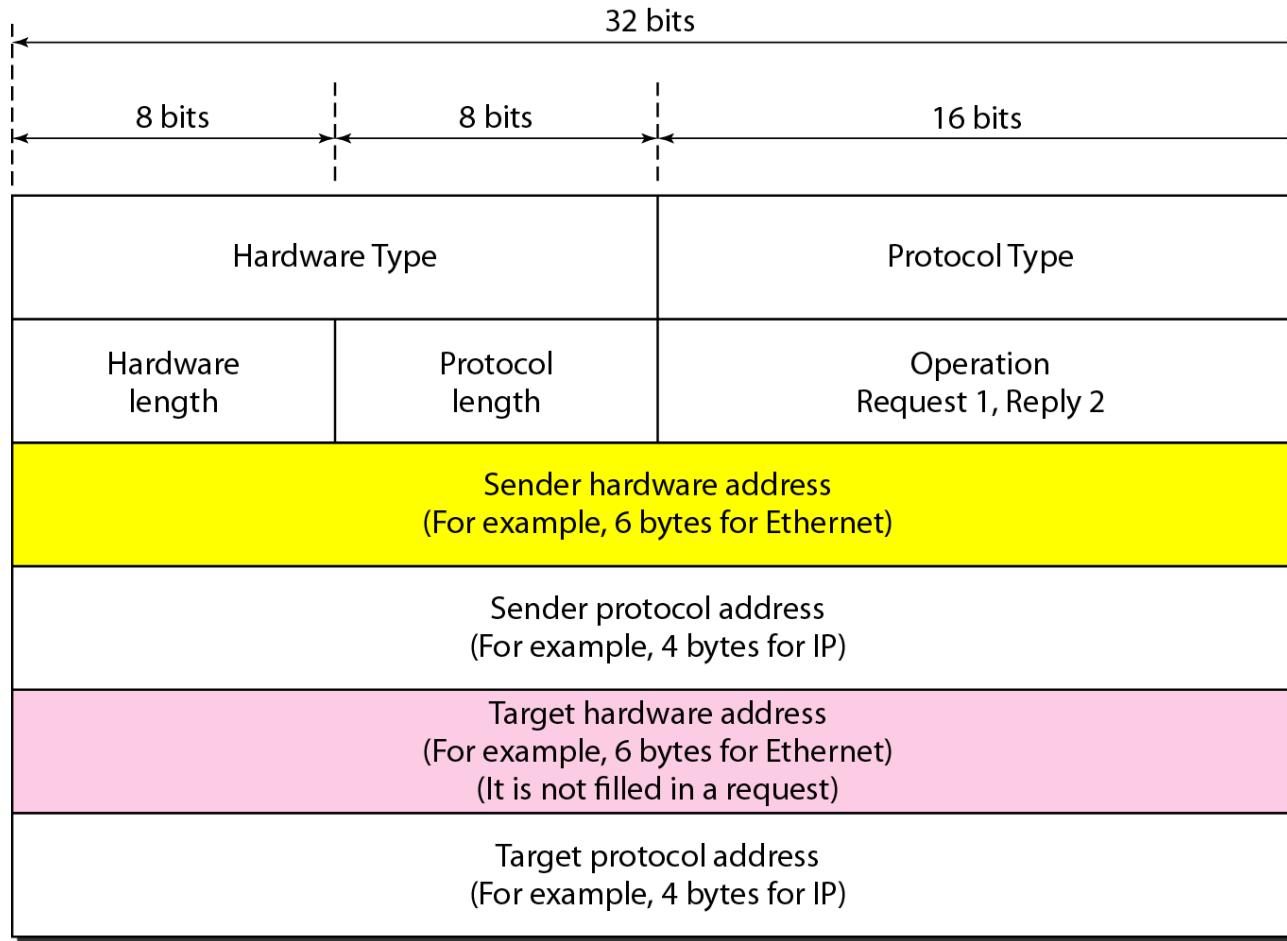


Figure 21.3 *Encapsulation of ARP packet*

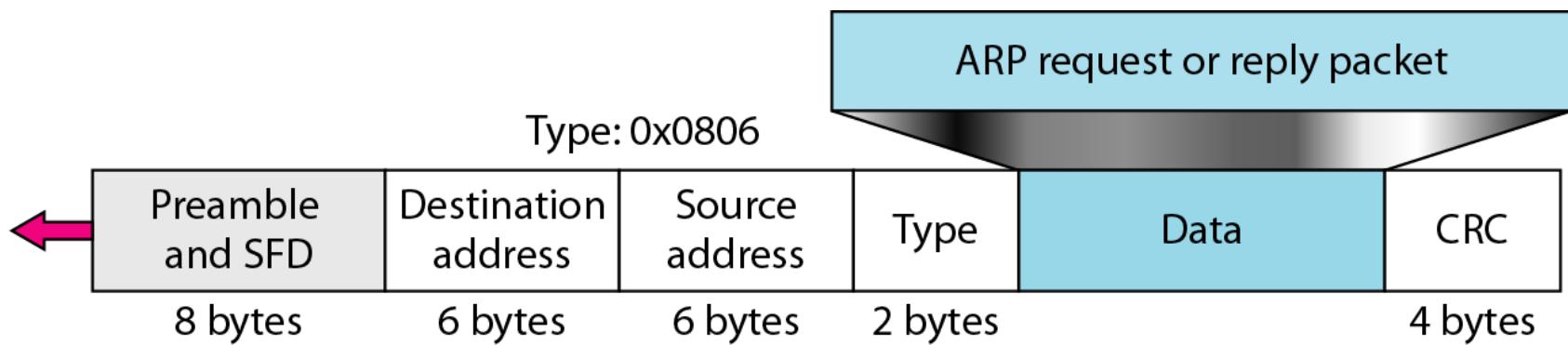
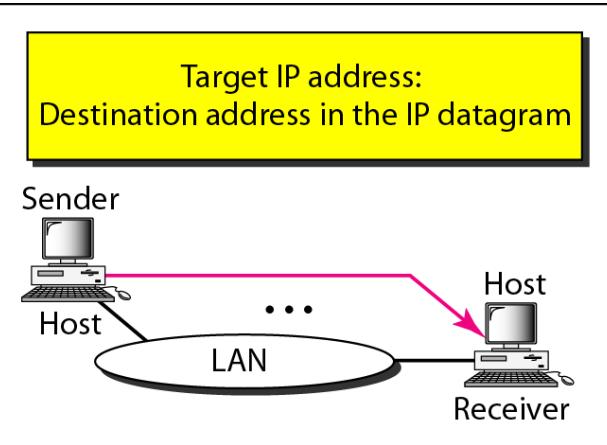
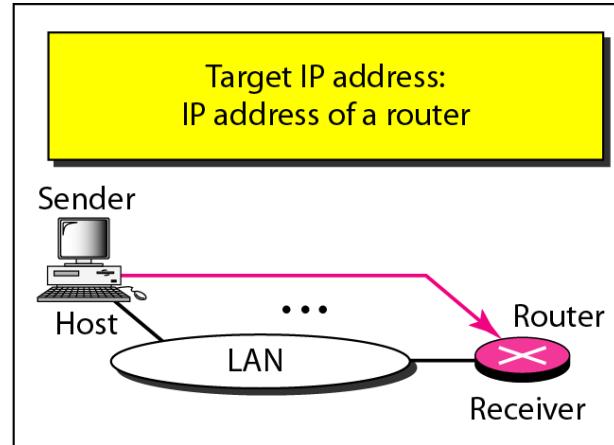


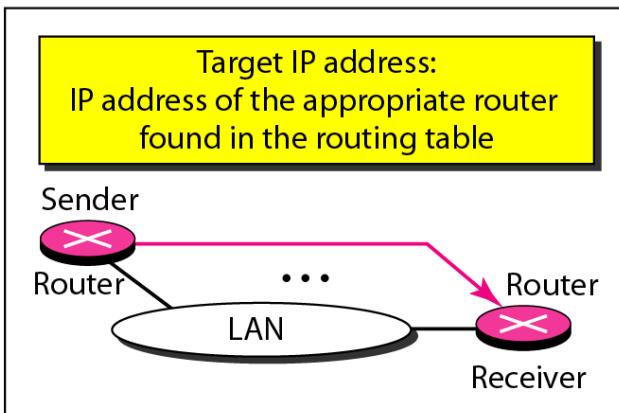
Figure 21.4 Four cases using ARP



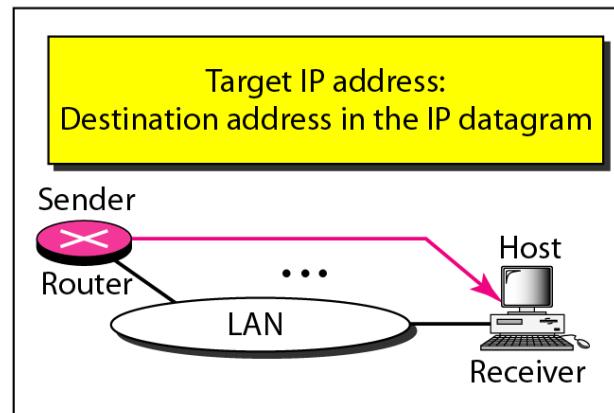
Case 1. A host has a packet to send to another host on the same network.



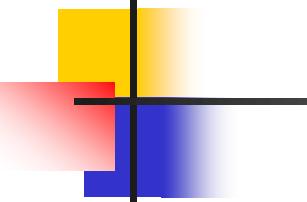
Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to a router.



Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.



Note

**An ARP request is broadcast;
an ARP reply is unicast.**

Example 21.1

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 21.5 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

Figure 21.5 Example 21.1, an ARP request and reply

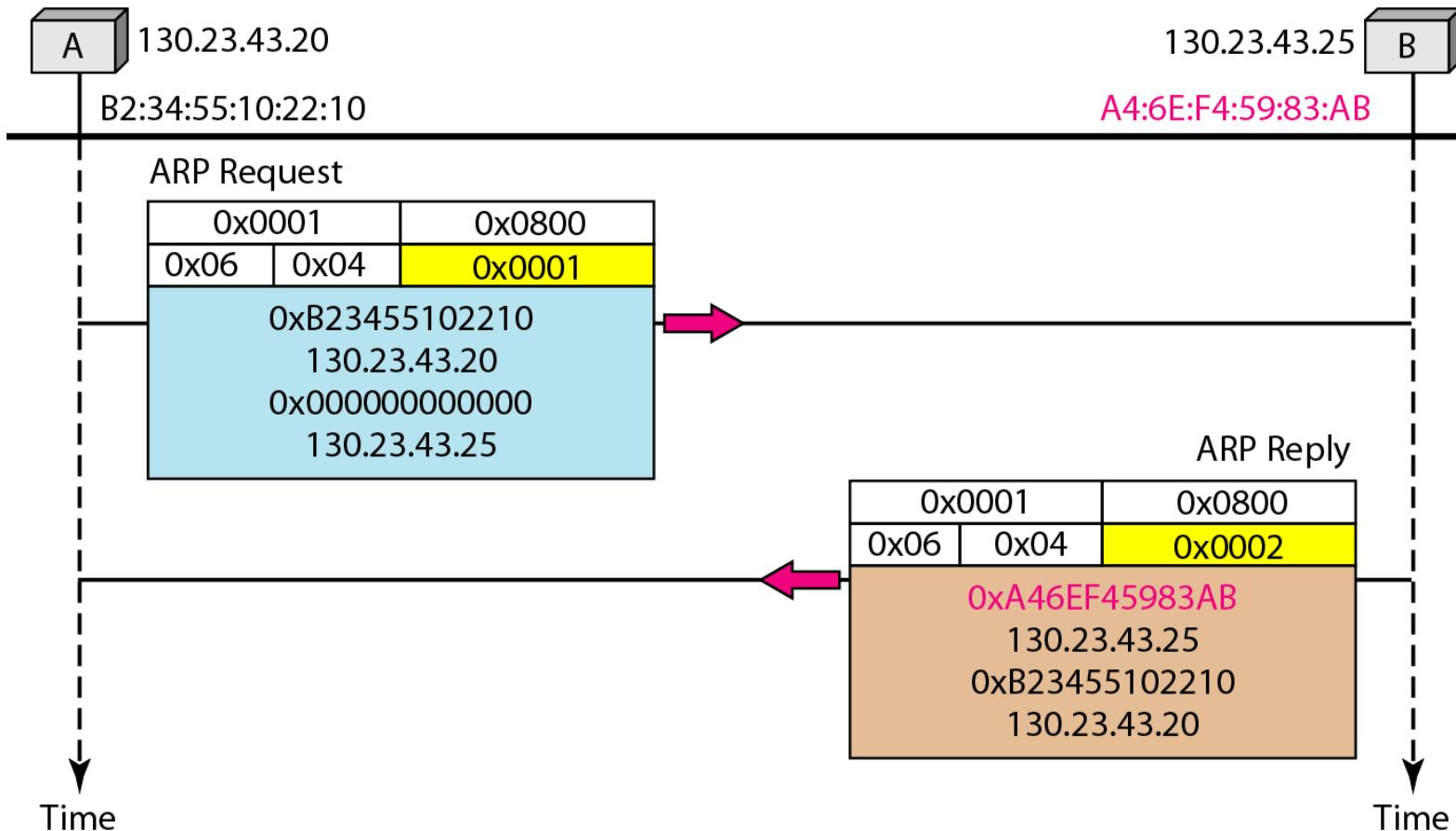


Figure 21.6 Proxy ARP

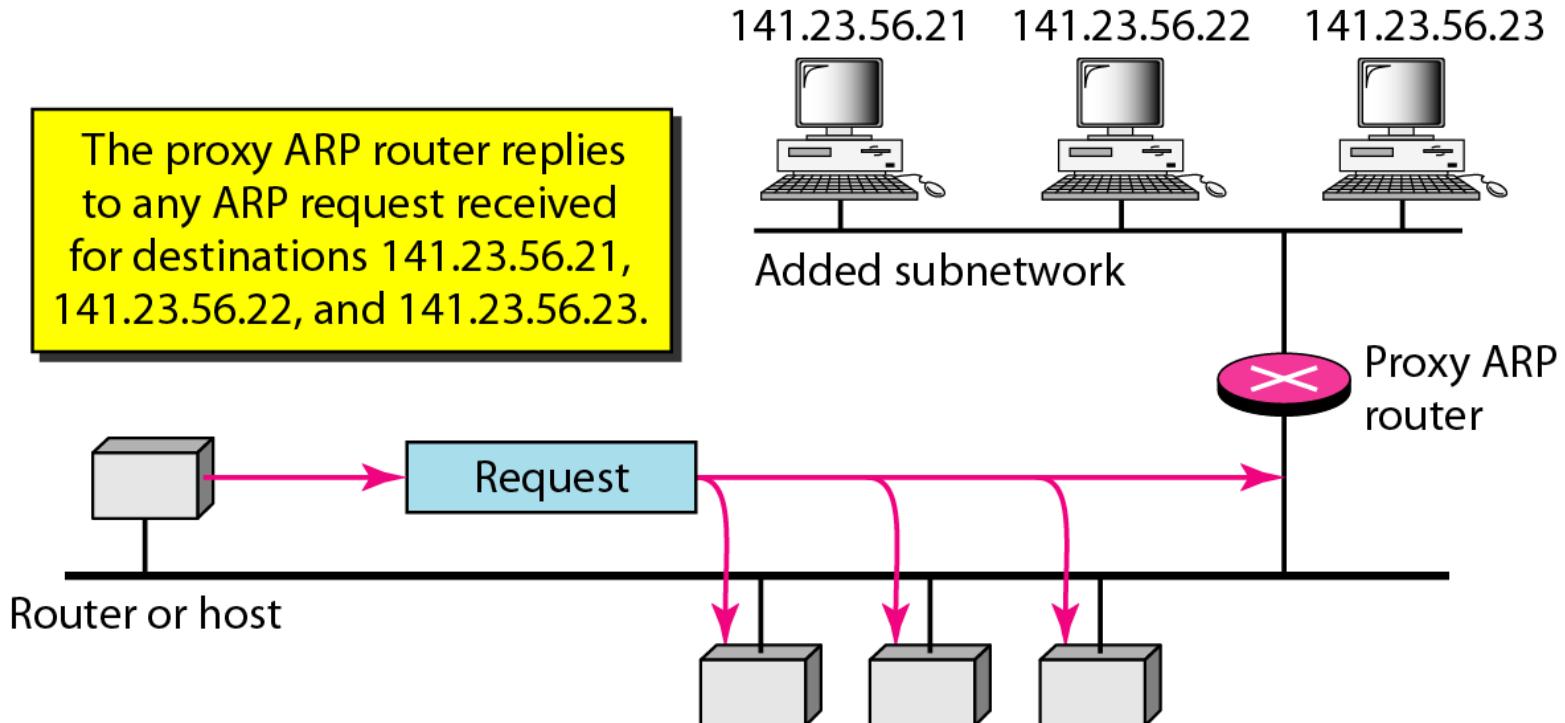
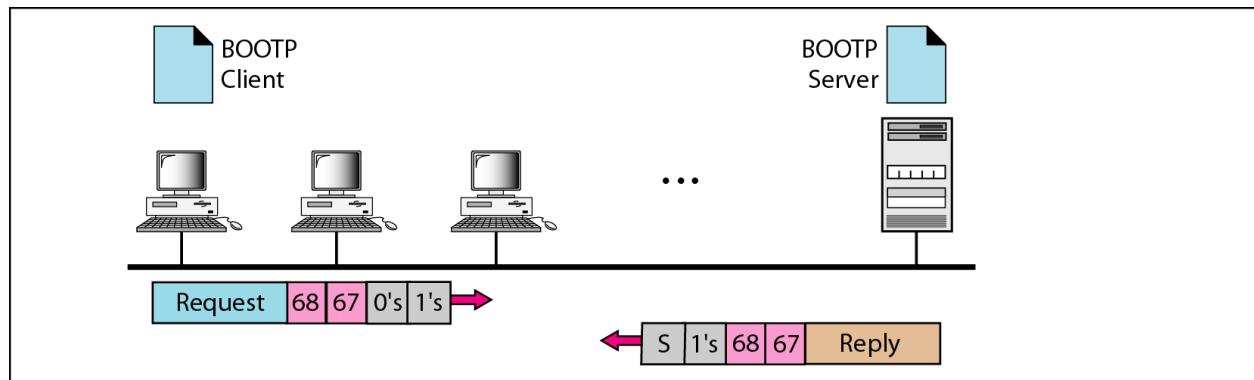
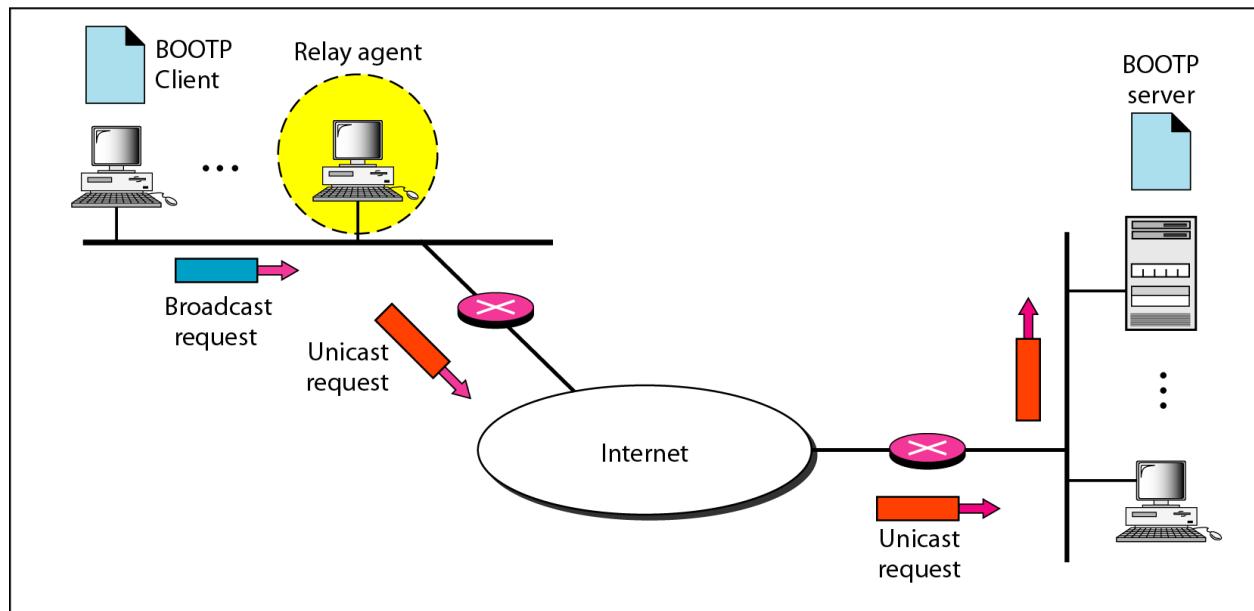


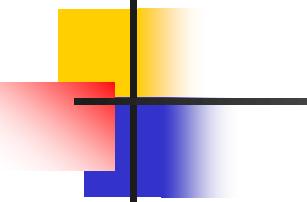
Figure 21.7 *BOOTP client and server on the same and different networks*



a. Client and server on the same network



b. Client and server on different networks



Note

DHCP provides static and dynamic address allocation that can be manual or automatic.

21-2 ICMP

The IP protocol has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

Topics discussed in this section:

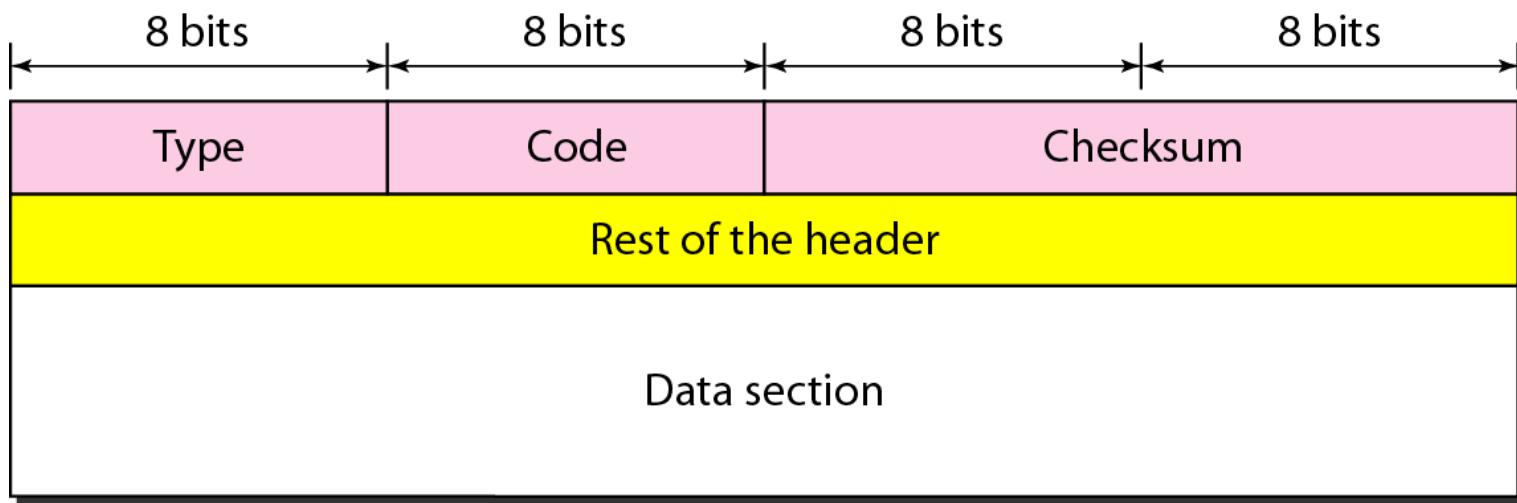
Types of Messages

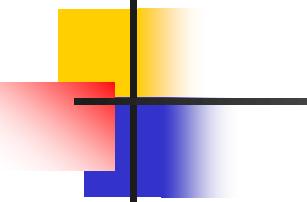
Message Format

Error Reporting and Query

Debugging Tools

Figure 21.8 General format of ICMP messages

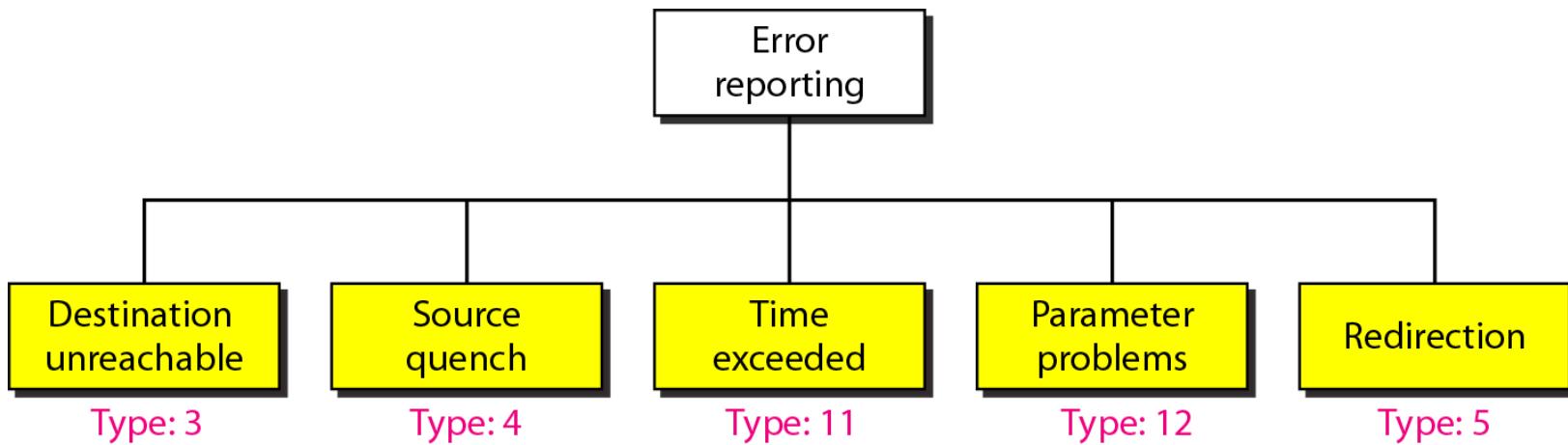


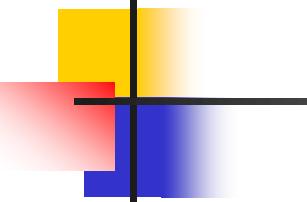


Note

ICMP always reports error messages to the original source.

Figure 21.9 *Error-reporting messages*





Note

Important points about ICMP error messages:

- No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- No ICMP error message will be generated for a datagram having a multicast address.
- No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

Figure 21.10 *Contents of data field for the error messages*

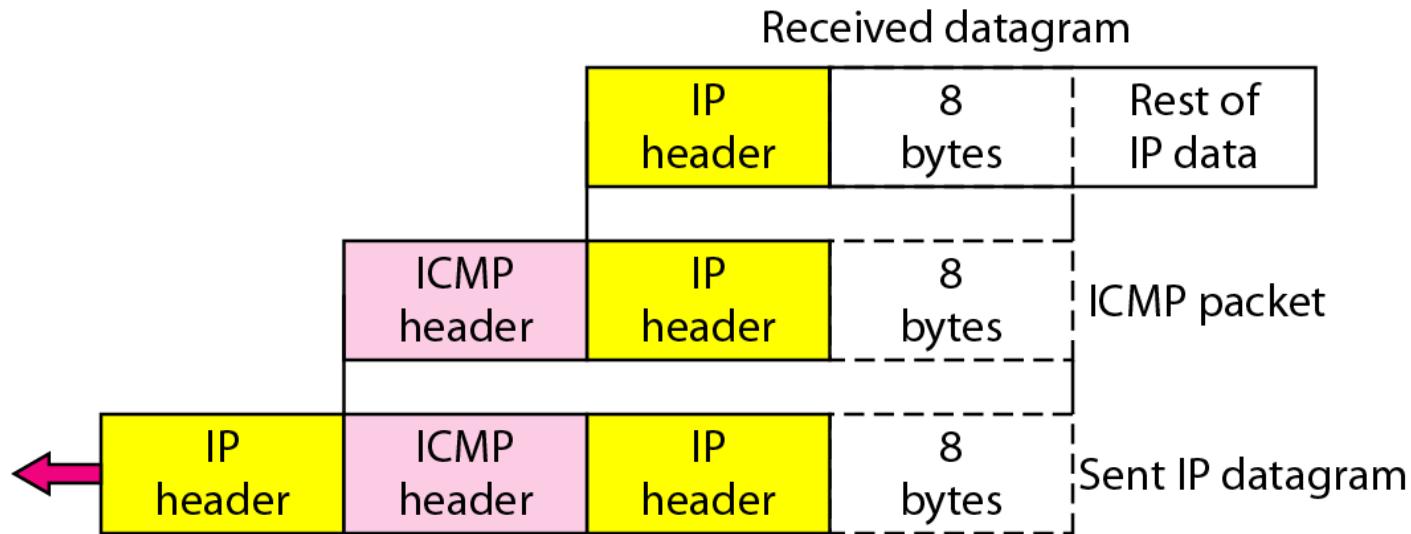


Figure 21.11 *Redirection concept*

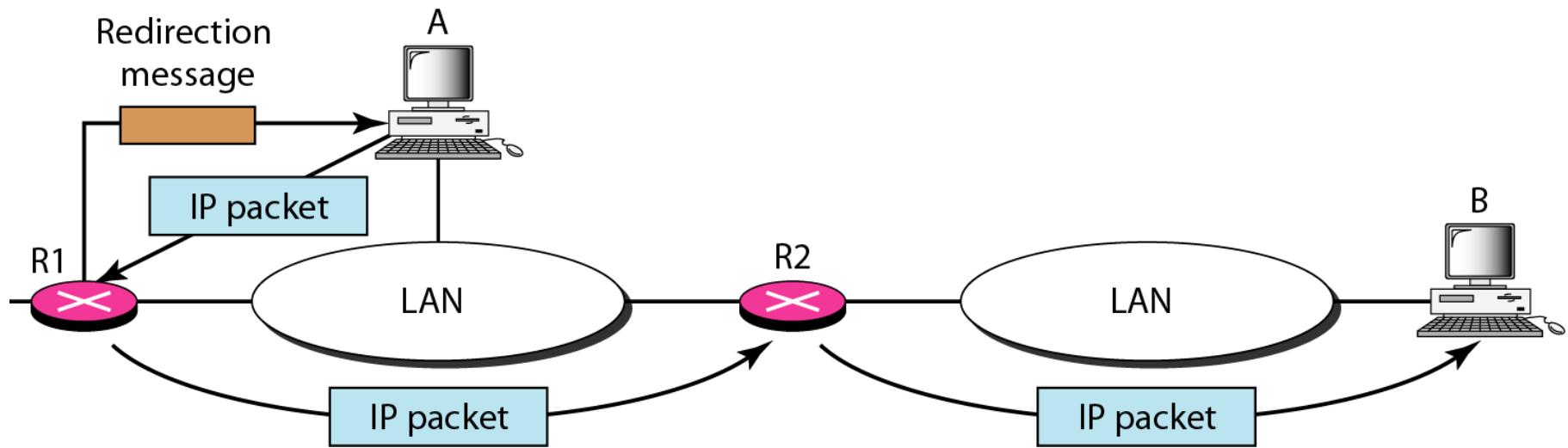


Figure 21.12 *Query messages*

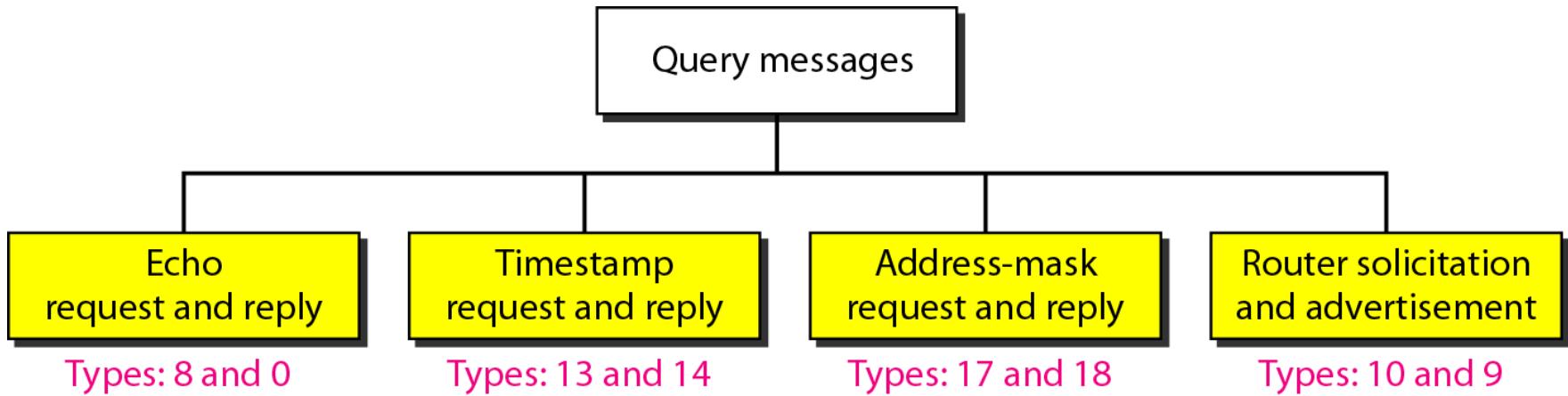


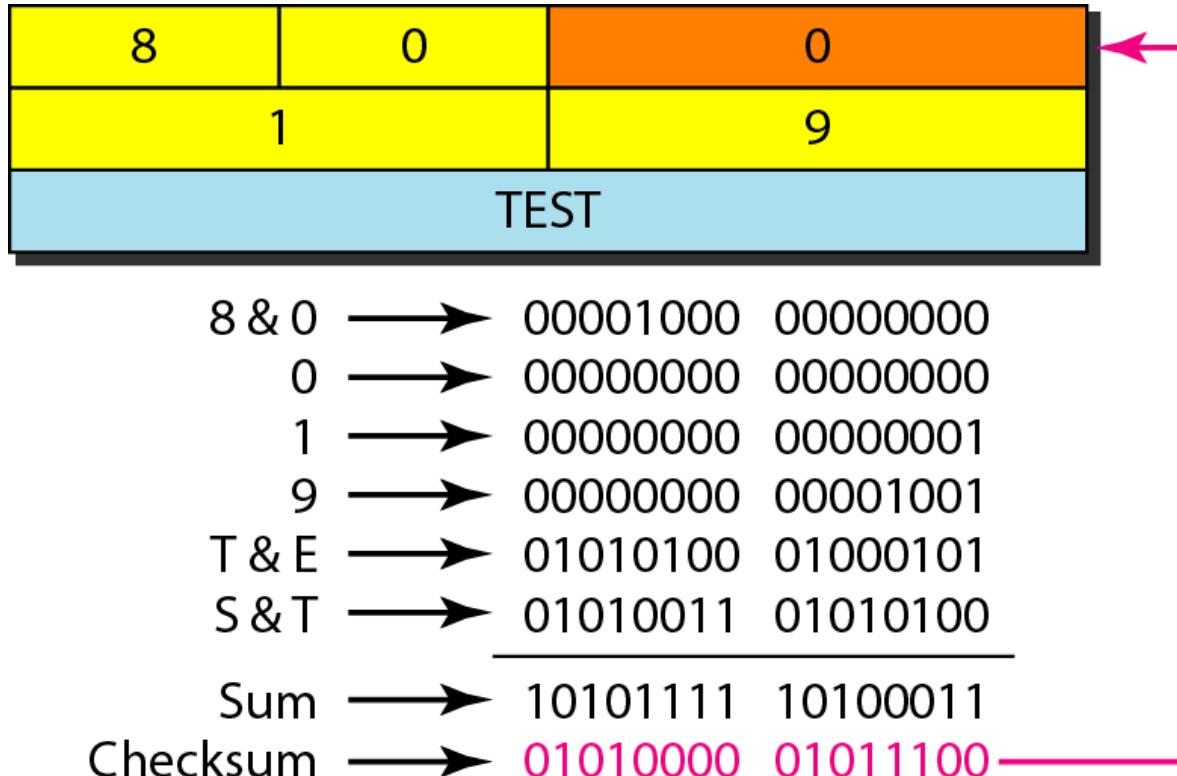
Figure 21.13 *Encapsulation of ICMP query messages*



Example 21.2

Figure 21.14 shows an example of checksum calculation for a simple echo-request message. We randomly chose the identifier to be 1 and the sequence number to be 9. The message is divided into 16-bit (2-byte) words. The words are added and the sum is complemented. Now the sender can put this value in the checksum field.

Figure 21.14 Example of checksum calculation



Example 21.3

We use the ping program to test the server fhda.edu. The result is shown on the next slide. The ping program sends messages with sequence numbers starting from 0. For each probe it gives us the RTT time. The TTL (time to live) field in the IP datagram that encapsulates an ICMP message has been set to 62. At the beginning, ping defines the number of data bytes as 56 and the total number of bytes as 84. It is obvious that if we add 8 bytes of ICMP header and 20 bytes of IP header to 56, the result is 84. However, note that in each probe ping defines the number of bytes as 64. This is the total number of bytes in the ICMP packet ($56 + 8$).

Example 21.3 (continued)

```
$ ping fhda.edu
```

PING fhda.edu (153.18.8.1) 56 (84) bytes of data.

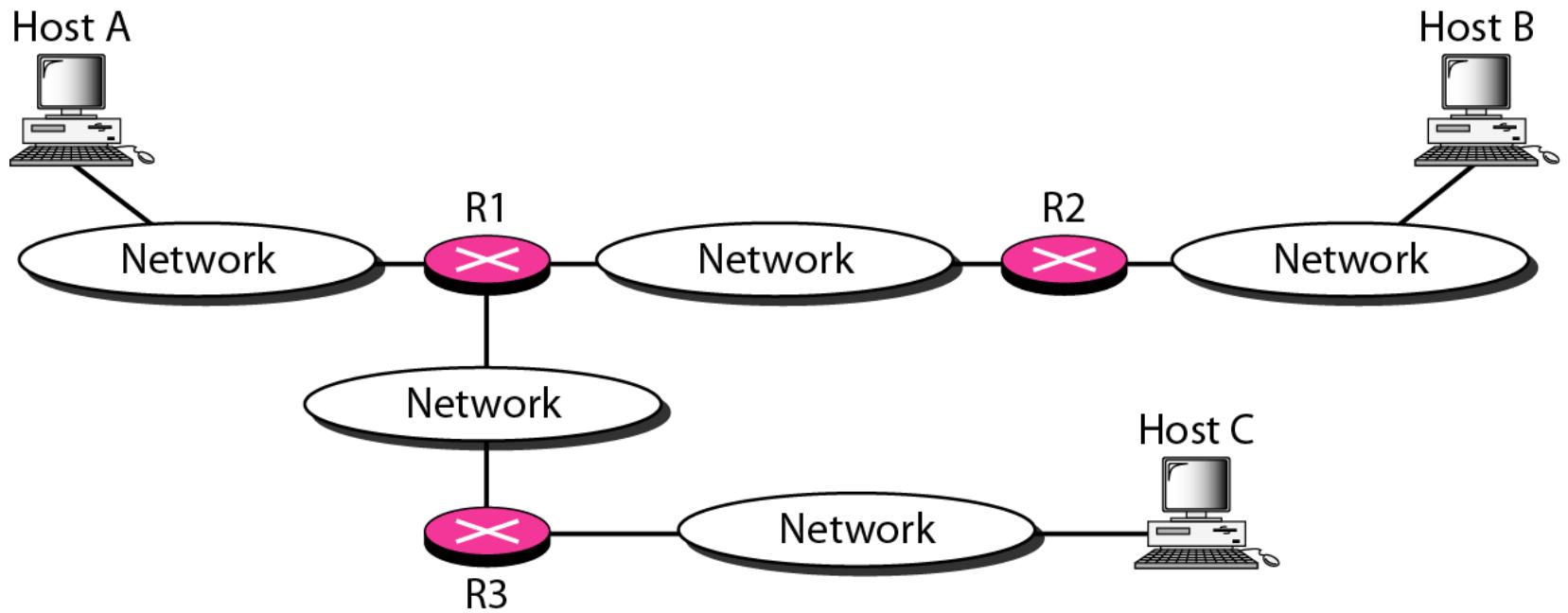
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0	ttl=62	time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1	ttl=62	time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2	ttl=62	time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4	ttl=62	time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5	ttl=62	time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9	ttl=62	time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10	ttl=62	time=1.98 ms

--- fhda.edu ping statistics ---

11 packets transmitted, 11 received, 0% packet loss, time 10103ms

rtt min/avg/max = 1.899/1.955/2.041 ms

Figure 21.15 *The traceroute program operation*

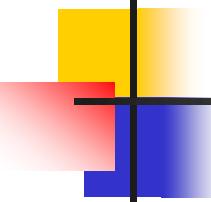


Example 21.4

We use the traceroute program to find the route from the computer voyager.deanza.edu to the server fhda.edu. The following shows the result:

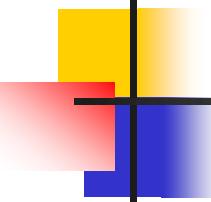
```
$ traceroute fhda.edu
traceroute to fhda.edu    (153.18.8.1), 30 hops max, 38 byte packets
 1 Dcore.fhda.edu        (153.18.31.254)   0.995 ms   0.899 ms   0.878 ms
 2 Dbackup.fhda.edu      (153.18.251.4)    1.039 ms   1.064 ms   1.083 ms
 3 tiptoe.fhda.edu       (153.18.8.1)      1.797 ms   1.642 ms   1.757 ms
```

The unnumbered line after the command shows that the destination is 153.18.8.1. The packet contains 38 bytes: 20 bytes of IP header, 8 bytes of UDP header, and 10 bytes of application data. The application data are used by traceroute to keep track of the packets.



Example 21.4 (continued)

The first line shows the first router visited. The router is named Dcore.fhda.edu with IP address 153.18.31.254. The first round-trip time was 0.995 ms, the second was 0.899 ms, and the third was 0.878 ms. The second line shows the second router visited. The router is named Dbackup.fhda.edu with IP address 153.18.251.4. The three round-trip times are also shown. The third line shows the destination host. We know that this is the destination host because there are no more lines. The destination host is the server fhda.edu, but it is named tiptoe.fhda.edu with the IP address 153.18.8.1. The three round-trip times are also shown.



Example 21.5

In this example, we trace a longer route, the route to xerox.com (see next slide). Here there are 17 hops between source and destination. Note that some round-trip times look unusual. It could be that a router was too busy to process the packet immediately.

Example 21.5 (continued)

\$ traceroute xerox.com

traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets

1	Dcore.fhda.edu	(153.18.31.254)	0.622 ms	0.891 ms	0.875 ms
2	Ddmz.fhda.edu	(153.18.251.40)	2.132 ms	2.266 ms	2.094 ms
3	Cinic.fhda.edu	(153.18.253.126)	2.110 ms	2.145 ms	1.763 ms
4	cenic.net	(137.164.32.140)	3.069 ms	2.875 ms	2.930 ms
5	cenic.net	(137.164.22.31)	4.205 ms	4.870 ms	4.197 ms
...
14	snfc21.pbi.net	(151.164.191.49)	7.656 ms	7.129 ms	6.866 ms
15	sbcglobal.net	(151.164.243.58)	7.844 ms	7.545 ms	7.353 ms
16	pacbell.net	(209.232.138.114)	9.857 ms	9.535 ms	9.603 ms
17	209.233.48.223	(209.233.48.223)	10.634 ms	10.771 ms	10.592 ms
18	alpha.Xerox.COM	(13.1.64.93)	11.172 ms	11.048 ms	10.922 ms

21-3 IGMP

The IP protocol can be involved in two types of communication: unicasting and multicasting. The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient, protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

Topics discussed in this section:

Group Management

IGMP Messages and IGMP Operation

Encapsulation

Netstat Utility

Figure 21.16 *IGMP message types*

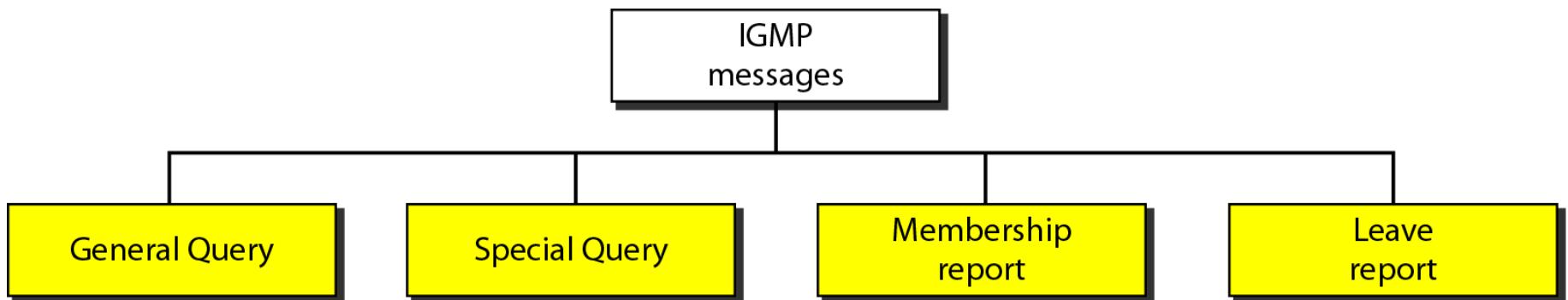


Figure 21.17 IGMP message format

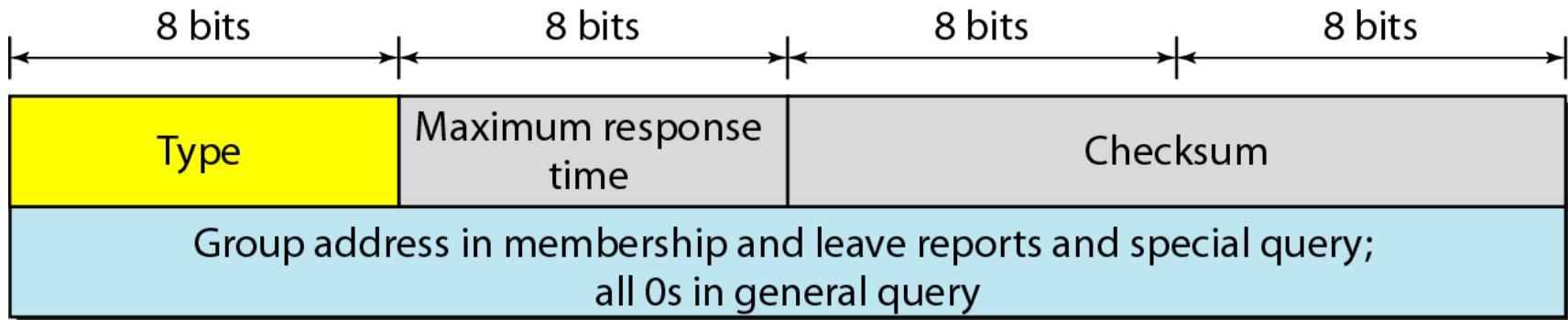
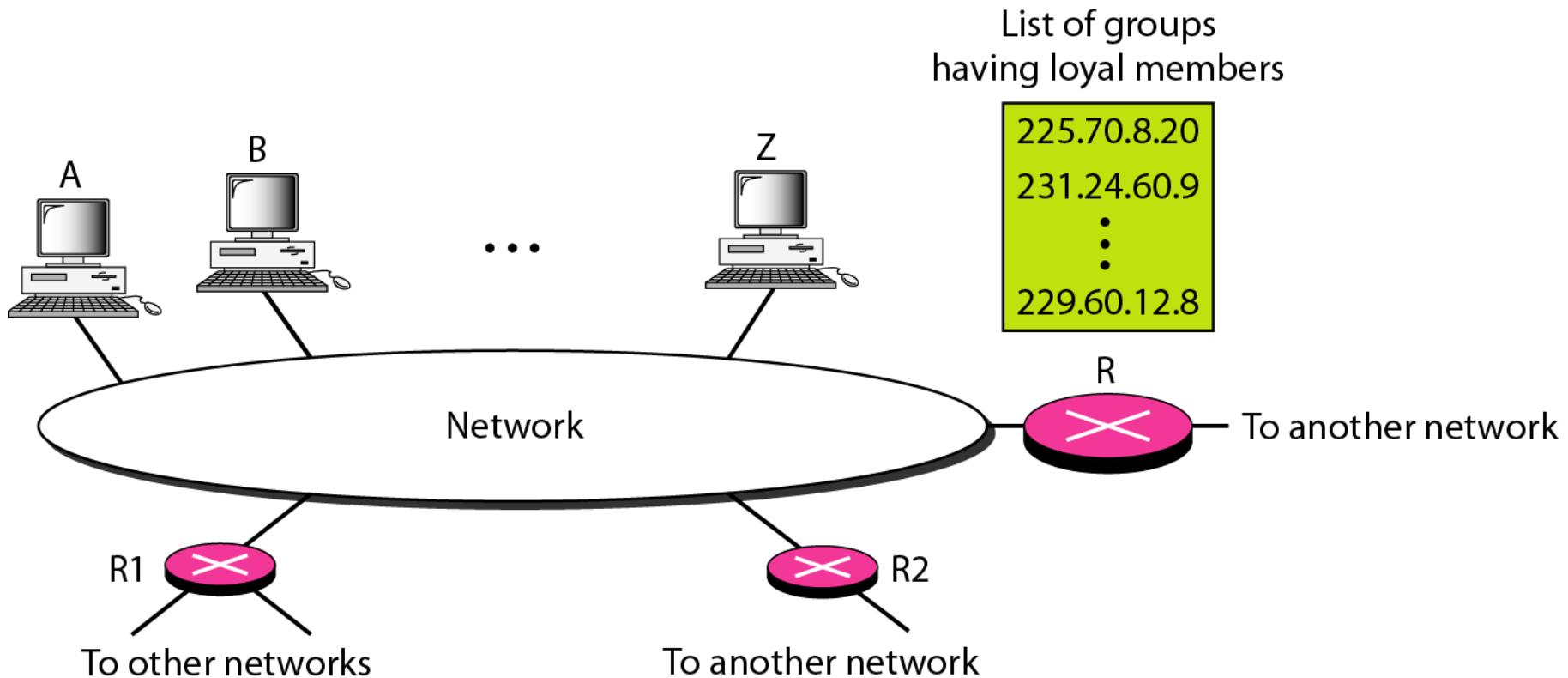
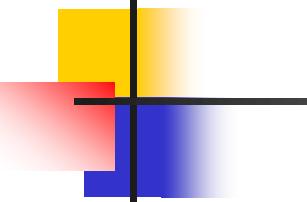


Table 21.1 *IGMP type field*

Type	Value
General or special query	0x11 or 00010001
Membership report	0x16 or 00010110
Leave report	0x17 or 00010111

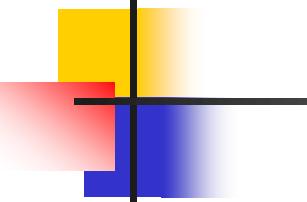
Figure 21.18 IGMP operation





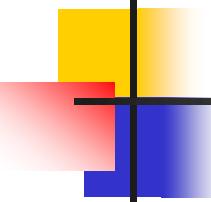
Note

In IGMP, a membership report is sent twice, one after the other.



Note

The general query message does not define a particular group.



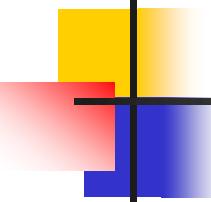
Example 21.6

Imagine there are three hosts in a network, as shown in Figure 21.19. A query message was received at time 0; the random delay time (in tenths of seconds) for each group is shown next to the group address. Show the sequence of report messages.

Solution

The events occur in this sequence:

- a. *Time 12: The timer for 228.42.0.0 in host A expires, and a membership report is sent, which is received by the router and every host including host B which cancels its timer for 228.42.0.0.*



Example 21.6 (continued)

- b. Time 30: The timer for 225.14.0.0 in host A expires, and a membership report is sent which is received by the router and every host including host C which cancels its timer for 225.14.0.0.*
- c. Time 50: The timer for 238.71.0.0 in host B expires, and a membership report is sent, which is received by the router and every host.*
- d. Time 70: The timer for 230.43.0.0 in host C expires, and a membership report is sent, which is received by the router and every host including host A which cancels its timer for 230.43.0.0.*

Figure 21.19 Example 21.6

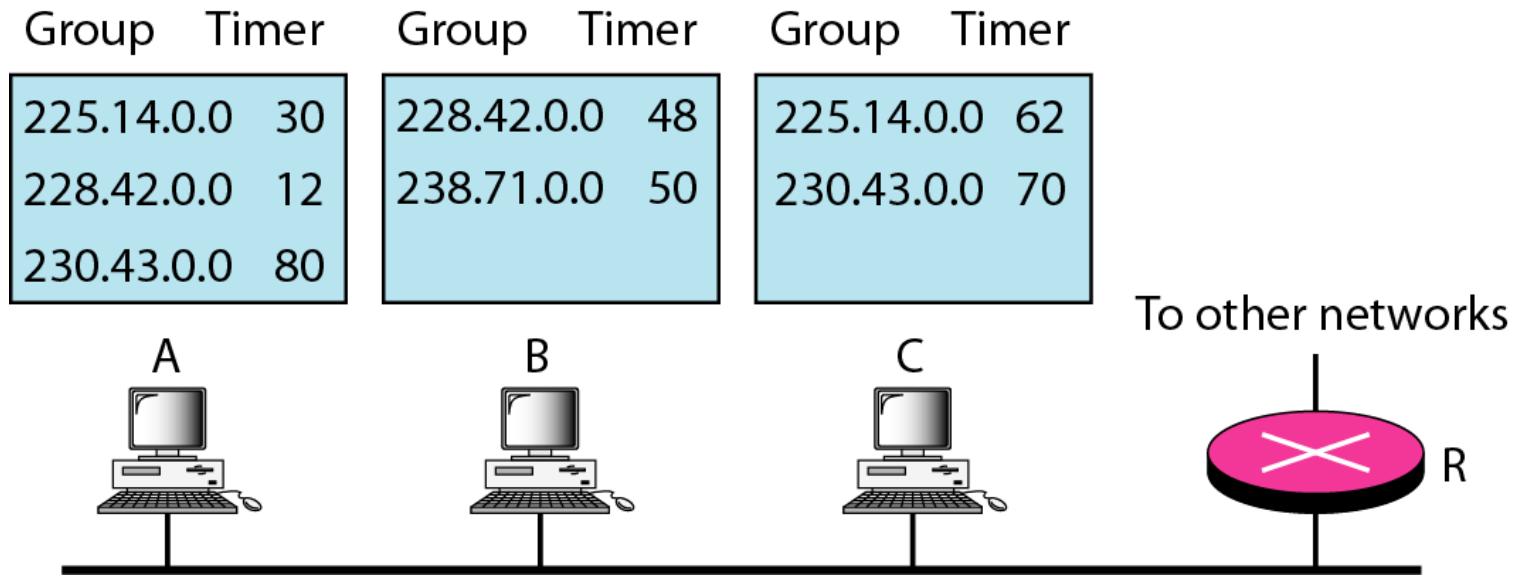
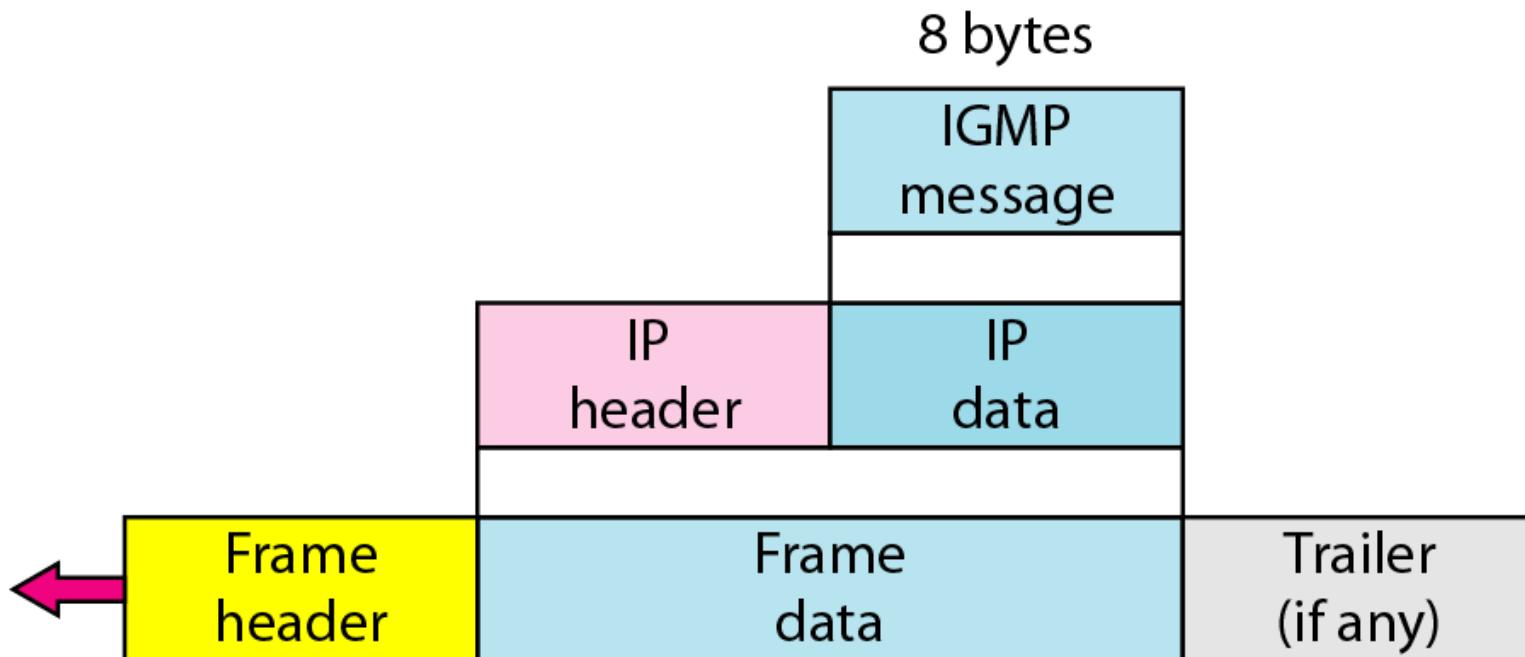
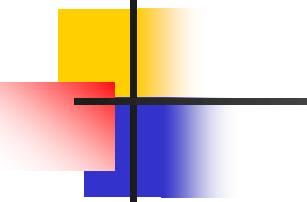


Figure 21.20 *Encapsulation of IGMP packet*





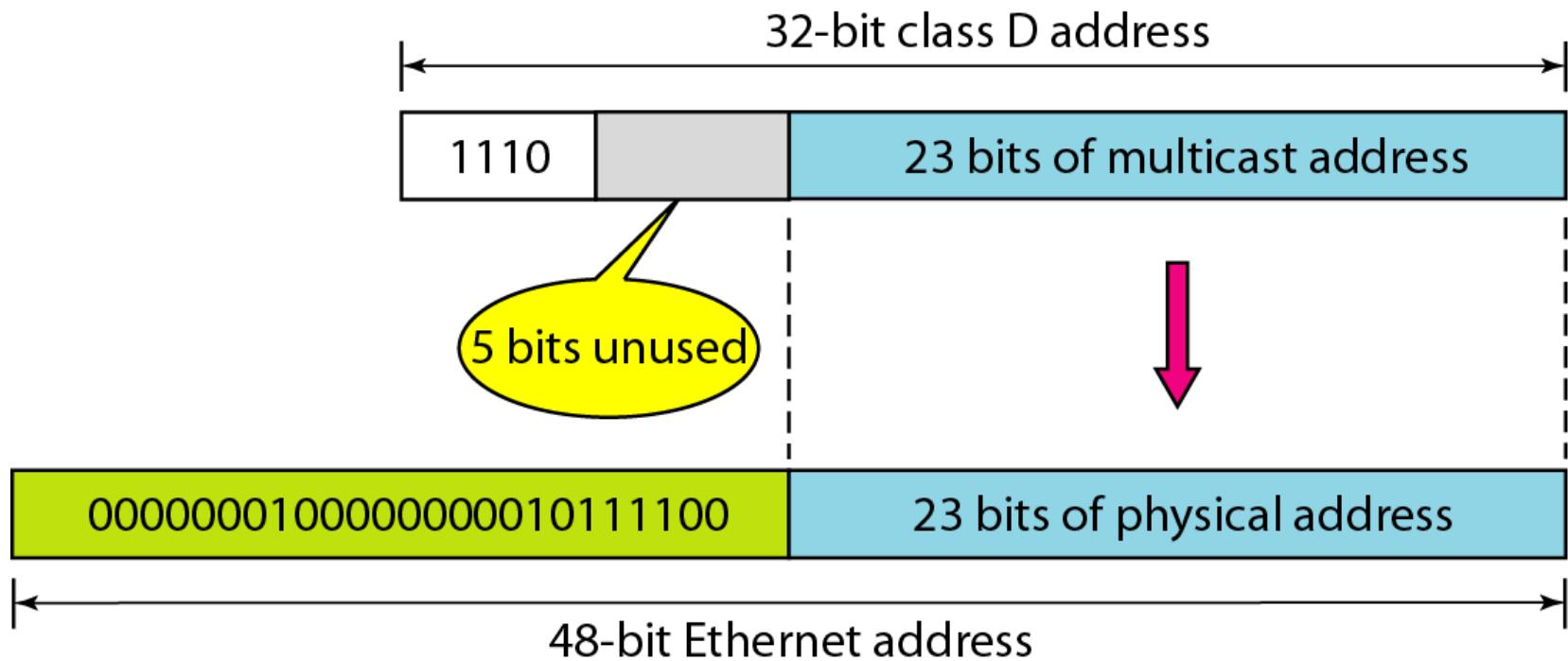
Note

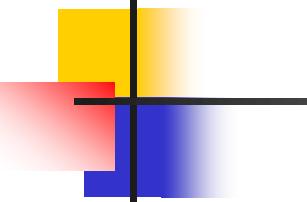
The IP packet that carries an IGMP packet has a value of 1 in its TTL field.

Table 21.2 *Destination IP addresses*

<i>Type</i>	<i>IP Destination Address</i>
Query	224.0.0.1 All systems on this subnet
Membership report	The multicast address of the group
Leave report	224.0.0.2 All routers on this subnet

Figure 21.21 *Mapping class D to Ethernet physical address*





Note

**An Ethernet multicast physical address
is in the range**

01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF.

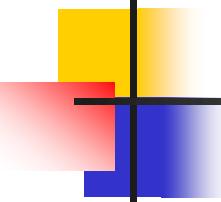
Example 21.7

Change the multicast IP address 230.43.14.7 to an Ethernet multicast physical address.

Solution

We can do this in two steps:

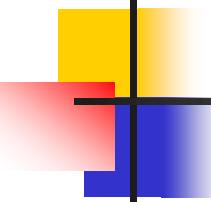
- a. *We write the rightmost 23 bits of the IP address in hexadecimal. This can be done by changing the rightmost 3 bytes to hexadecimal and then subtracting 8 from the leftmost digit if it is greater than or equal to 8. In our example, the result is 2B:0E:07.*



Example 21.7 (continued)

- b.** We add the result of part a to the starting Ethernet multicast address, which is 01:00:5E:00:00:00. The result is

01:00:5E:2B:0E:07



Example 21.8

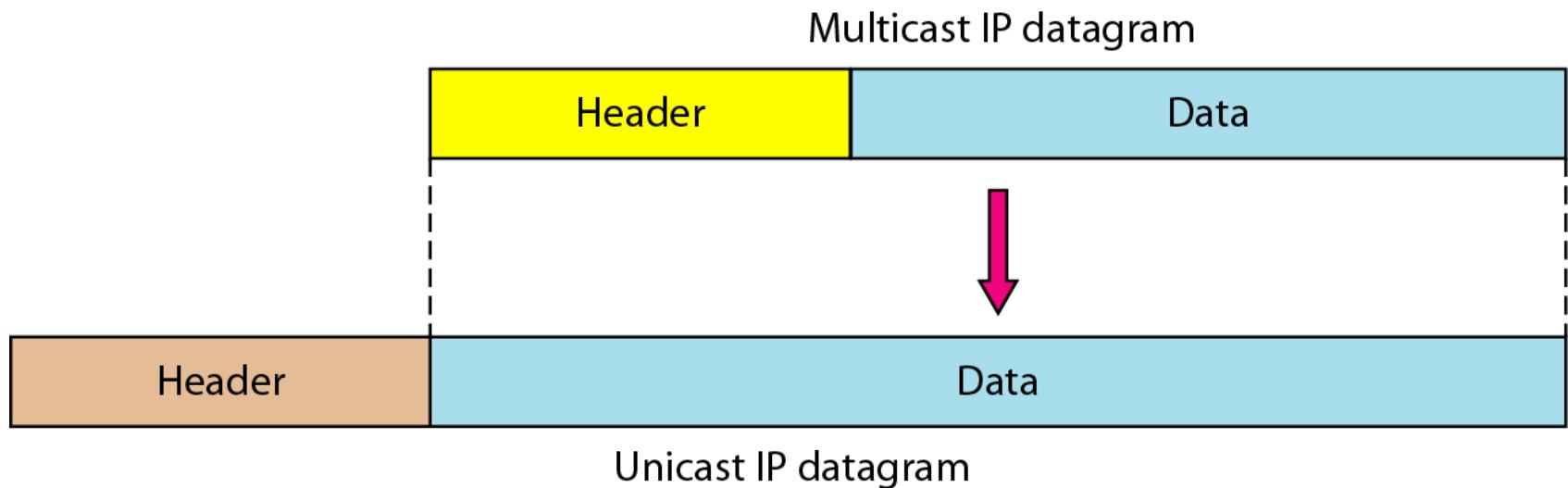
Change the multicast IP address 238.212.24.9 to an Ethernet multicast address.

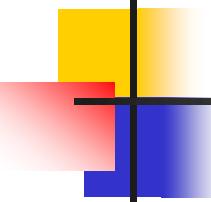
Solution

- a. The rightmost 3 bytes in hexadecimal is D4:18:09. We need to subtract 8 from the leftmost digit, resulting in 54:18:09.*
- b. We add the result of part a to the Ethernet multicast starting address. The result is*

01:00:5E:54:18:09

Figure 21.22 Tunneling





Example 21.9

We use *netstat* (see next slide) with three options: *-n*, *-r*, and *-a*. The *-n* option gives the numeric versions of IP addresses, the *-r* option gives the routing table, and the *-a* option gives all addresses (unicast and multicast). Note that we show only the fields relative to our discussion. “Gateway” defines the router, “Iface” defines the interface.

Note that the multicast address is shown in color. Any packet with a multicast address from 224.0.0.0 to 239.255.255.255 is masked and delivered to the Ethernet interface.

Example 21.9 (continued)

```
$ netstat -nra
```

Kernel IP routing table

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
224.0.0.0	0.0.0.0	224.0.0.0	U	eth0
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

21-4 ICMPv6

We discussed IPv6 in Chapter 20. Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP (ICMPv6). This new version follows the same strategy and purposes of version 4.

Topics discussed in this section:

Error Reporting

Query

Figure 21.23 *Comparison of network layers in version 4 and version 6*

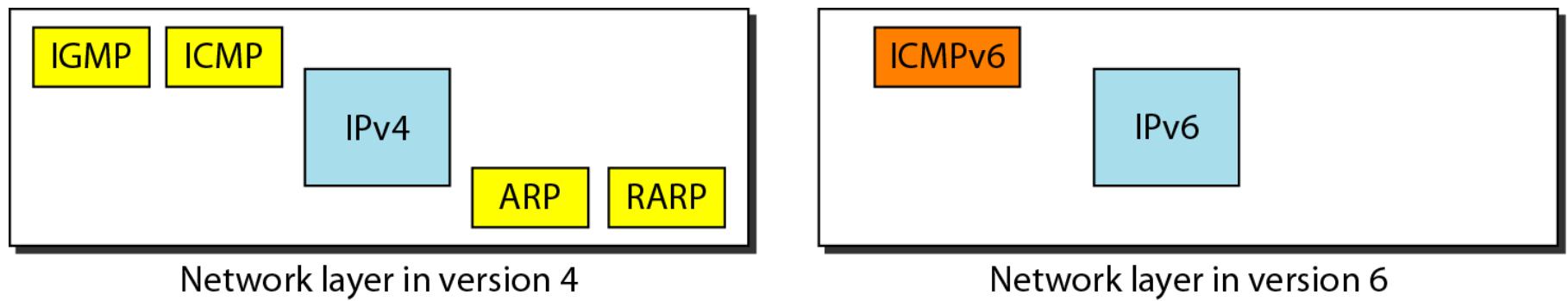


Table 21.3 *Comparison of error-reporting messages in ICMPv4 and ICMPv6*

Type of Message	Version 4	Version 6
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Table 21.4 *Comparison of query messages in ICMPv4 and ICMPv6*

Type of Message	Version 4	Version 6
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address-mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbor solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes

Module 3

Network Layer:

Logical Addressing

19-1 IPv4 ADDRESSES

An **IPv4 address** is a **32-bit** address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

Topics discussed in this section:

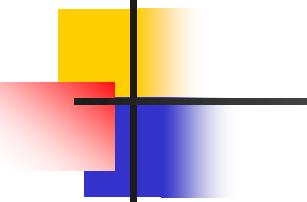
Address Space

Notations

Classful Addressing

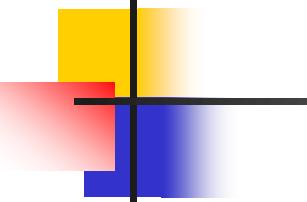
Classless Addressing

Network Address Translation (NAT)



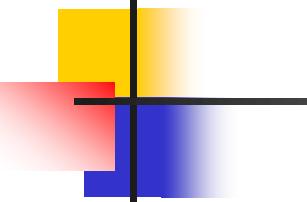
Note

An IPv4 address is 32 bits long.



Note

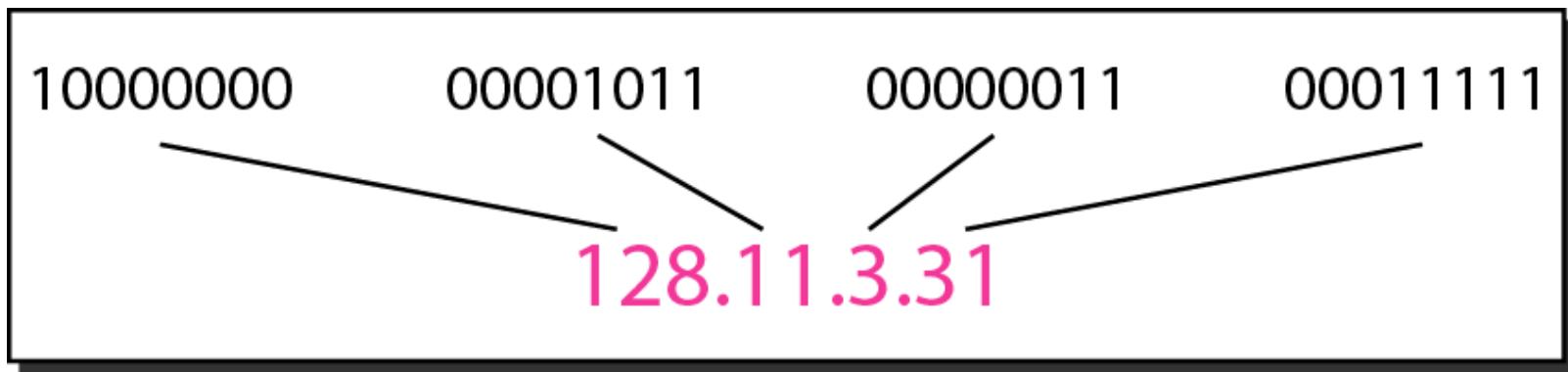
**The IPv4 addresses are unique
and universal.**

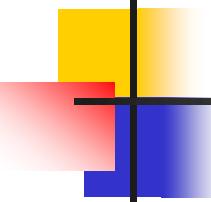


Note

**The address space of IPv4 is
 2^{32} or 4,294,967,296.**

Figure 19.1 Dotted-decimal notation and binary notation for an IPv4 address





Example 19.1

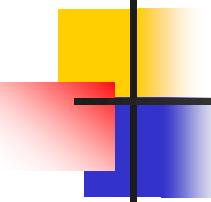
Change the following IPv4 addresses from binary notation to dotted-decimal notation.

- a. 10000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation.

- a. 129.11.11.239
- b. 193.131.27.255



Example 19.2

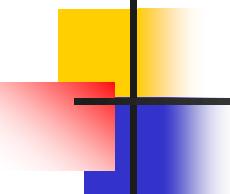
Change the following IPv4 addresses from dotted-decimal notation to binary notation.

- a. 111.56.45.78
- b. 221.34.7.82

Solution

We replace each decimal number with its binary equivalent (8bit).

- a. 01101111 00111000 00101101 01001110
- b. 11011101 00100010 00000111 01010010



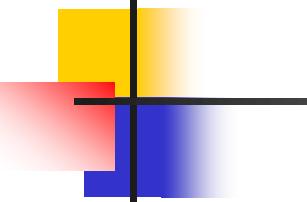
Example 19.3

Find the error, if any, in the following IPv4 addresses.

- a. 111.56.045.78
- b. 221.34.7.8.20
- c. 75.45.301.14
- d. 11100010.23.14.67

Solution

- a. *There must be no leading zero (045).*
- b. *There can be no more than four numbers.*
- c. *Each number needs to be less than or equal to 255.*
- d. *A mixture of binary notation and dotted-decimal notation is not allowed.*



Note

**In classful addressing, the address space is divided into five classes:
A, B, C, D, and E.**

Figure 19.2 Finding the classes in binary and dotted-decimal notation

10101100 172.x.x.x

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

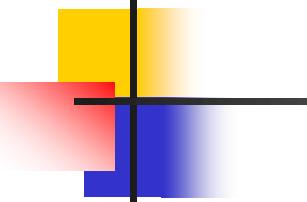
a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Figure: Hierarchy in IP Address



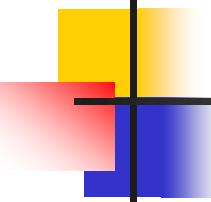


Note

Each address in the block can be considered as a two-level hierarchical structure:

the leftmost n bits (prefix) define the network;

the rightmost $32 - n$ bits define the host



Example 19.4

Find the class of each address.

- a. 00000001 00001011 00001011 11101111
- b. 11000001 10000011 00011011 11111111
- c. 14.23.120.8
- d. 252.5.15.111

Solution

- a. *The first bit is 0. This is a class A address.*
- b. *The first 2 bits are 1; the third bit is 0. This is a class C address.*
- c. *The first byte is 14; the class is A.*
- d. *The first byte is 252; the class is E.*

Table 19.1 *Number of blocks and block size in classful IPv4 addressing*

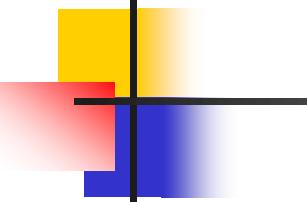
<i>Class</i>	<i>Number of Blocks</i>	<i>Block Size</i>	<i>Application</i>
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Mask

- It is a 32 bit number in which the 'n' leftmost bits are 1s and 32-n rightmost bits are 0s

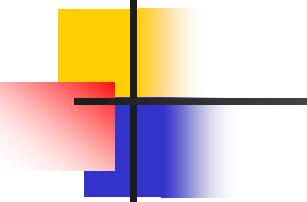
Table *Default masks for classful addressing*

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24



Note

In classful addressing, a large part of the available addresses were wasted.



Note

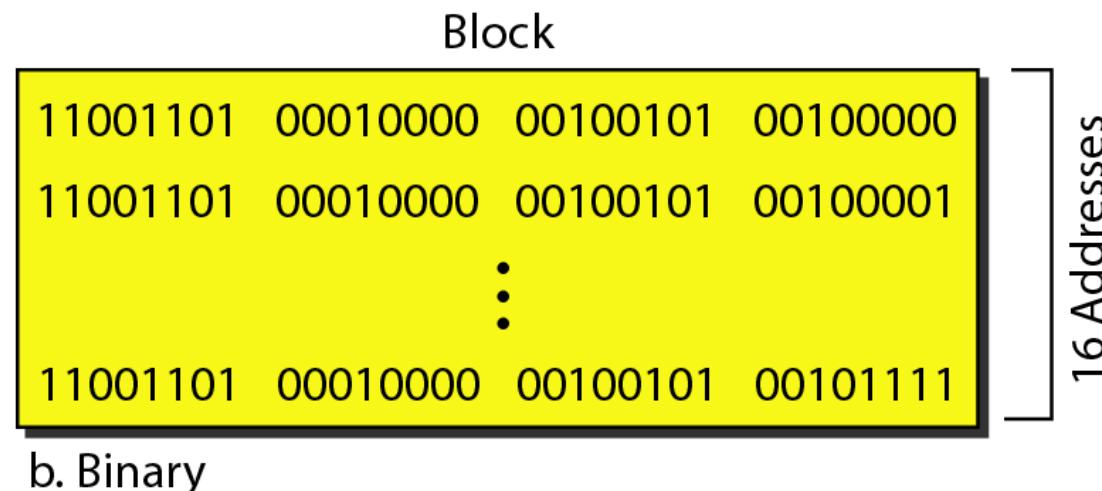
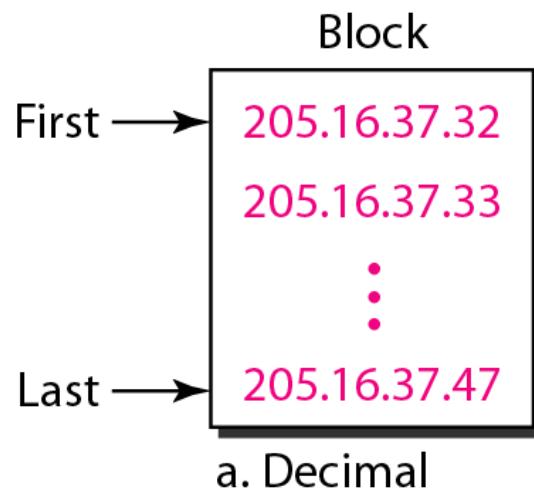
Classful addressing, which is almost obsolete, is replaced with classless addressing.

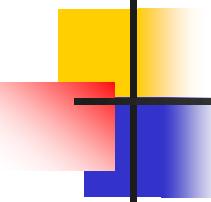
Sub-netting and Super-netting

- **Sub-netting:** Dividing the addresses into several contiguous groups and assigning each group to smaller networks, called subnets
 - Increases No. of 1s in the mask
- **Super-netting:** Combining several (class-C) address blocks to create a larger network; super-net
 - Decreases No. of 1s in the mask

Ex19.5, Fig 19.3 A block of 16 addresses granted to a small organization

Figure shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.



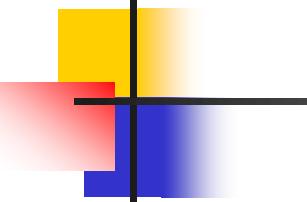


Example 19.5

Restrictions applied to this block

- *The addresses must be contiguous*
- *The number of addresses must be power of 2 (here, $16 = 2^4$),
and*
- *The first address must be evenly divisible by number of addresses
(here, divisible by 16)*

The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210.

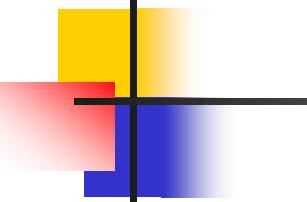


Note

In IPv4 addressing, a block of addresses can be defined as

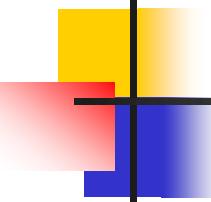
x.y.z.t /n

in which x.y.z.t defines one of the addresses and the /n defines the mask.



Note

The first address in the block can be found by setting the **rightmost $32 - n$ bits to 0s.**



Example 19.6

A block of addresses is granted to a small organization. One of the addresses is 205.16.37.39/28. What is the first address in the block?

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

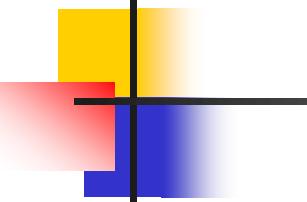
If we set 32–28 (i.e 4) rightmost bits to 0, we get

11001101 00010000 00100101 00100000

or

205.16.37.32.

This is actually the block shown in Figure 19.3.



Note

The last address in the block can be found by setting the **rightmost $32 - n$ bits to 1s.**

Example 19.7

Find the last address for the block if one of the address is 205.16.37.39/28 (Example 19.6).

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

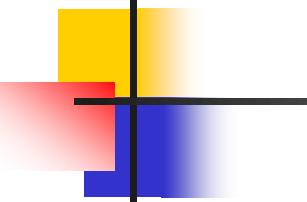
If we set 32 – 28 rightmost bits to 1, we get

11001101 00010000 00100101 00101111

or

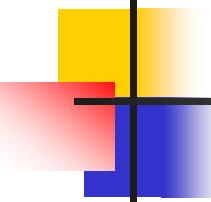
205.16.37.47

This is actually the block shown in Figure 19.3.



Note

**The number of addresses in the block
can be found by using the formula
 2^{32-n} .**



Example 19.8

Find the number of addresses in Example 19.6.

Solution

The value of n is 28, which means that number of addresses is 2^{32-28} or 16.

Example 19.9 Alternate Method

Another way to find the first address, the last address, and the number of addresses particularly useful when we are writing a program to find this information.

Methodology

Represent the mask as a 32-bit binary (or 8-digit hexadecimal) number

In Example 19.5 the /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s)

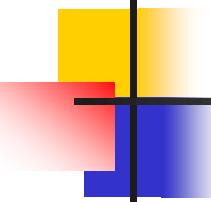
Example 19.9 (continued)

- a. The first address can be found by *ANDing the given addresses with the mask*. ANDing here is done bit by bit.

Address: 11001101 00010000 00100101 00100111

Mask: **11111111 11111111 11111111 11110000**

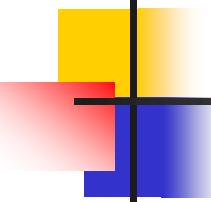
First address: 11001101 00010000 00100101 00100000



Example 19.9 (continued)

- b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit.*

Address:	11001101	00010000	00100101	00100111
Mask complement:	00000000	00000000	00000000	00001111
Last address:	11001101	00010000	00100101	00101111



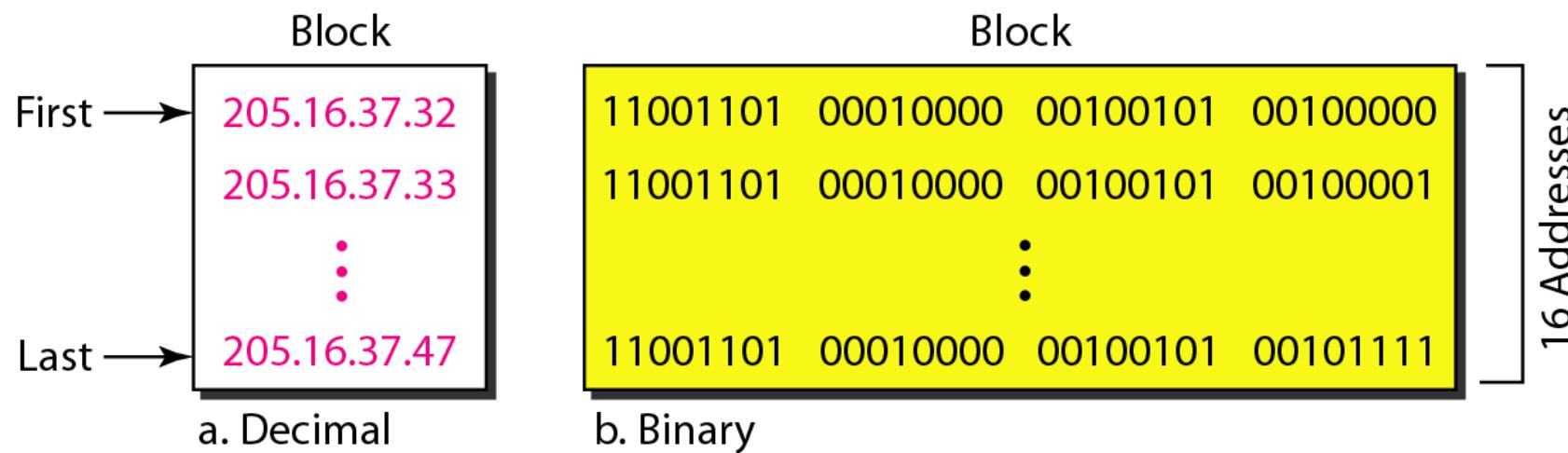
Example 19.9 (continued)

- c. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

Mask complement: 00000000 00000000 00000000 00001111

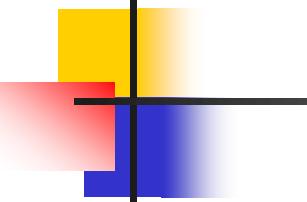
Number of addresses: $15 + 1 = 16$

Figure 19.4 A network configuration for the block 205.16.37.32/28



Example: Find first and last IP address for given following address

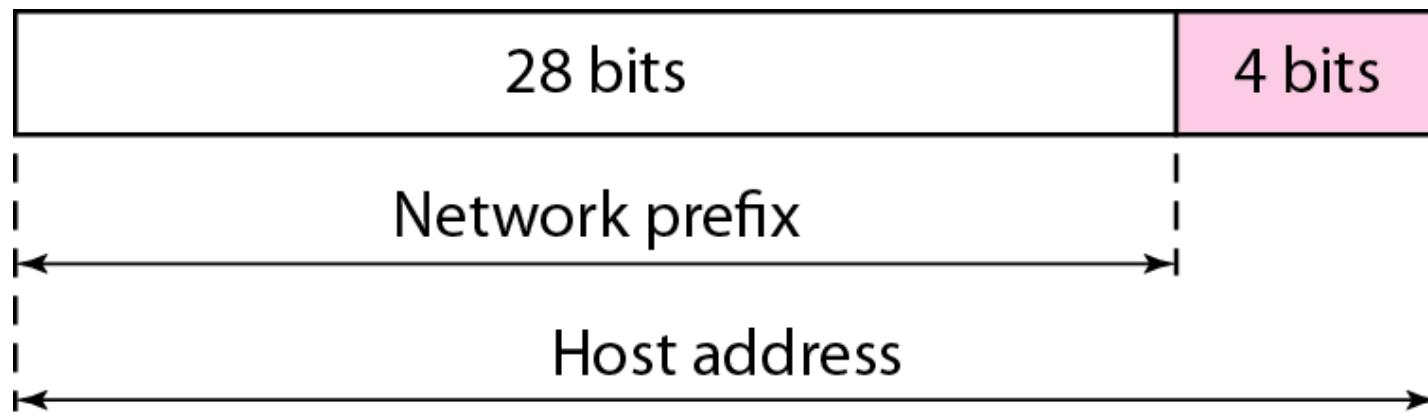
- 192.168.14.5 / 255.255.255.0
- 11000000 10101000 00001110 00000101
- 11111111 11111111 11111111 00000000
- first IP → 192.168.14.0
- Last address- compliment the mask and perform bit wise OR operation with given IP address
- 0000000 00000000 00000000 11111111
- Last IP add → 192.168.14.255

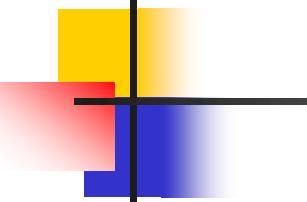


Note

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Figure 19.6 *Two Level Hierarchy in IPv4 Address*



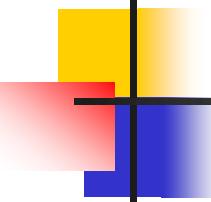


Note

Each address in the block can be considered as a two-level hierarchical structure:

the leftmost n bits (prefix) define the network;

the rightmost $32 - n$ bits define the host



Example 19.10

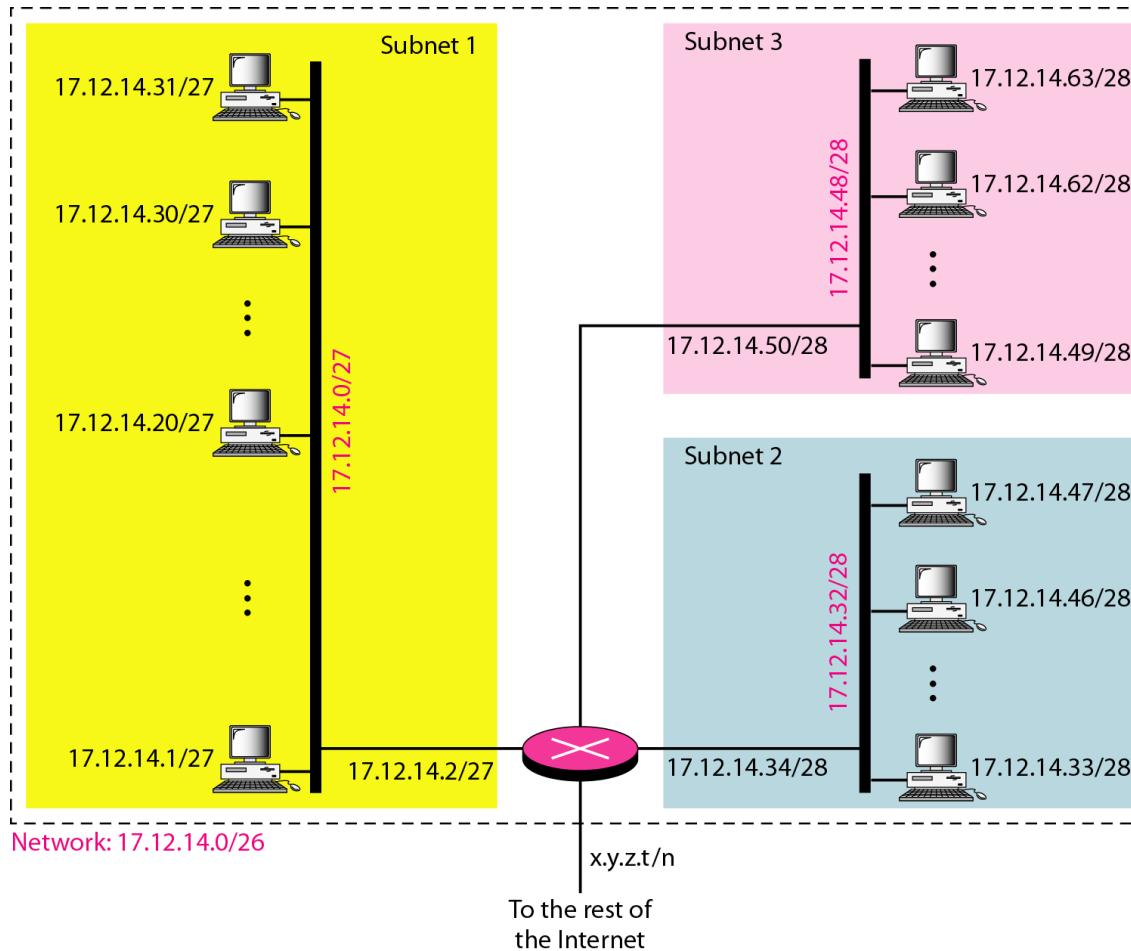
An organization is given a block of addresses starting with 17.12.40.0/26 (64 addresses). The organization has 3 offices and needs to distribute these addresses into three sub-blocks of 32, 16 and 16 addresses.

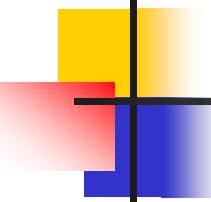
Carry out the allocation.

Solution:

- Given mask is /26 to define 64 addresses
- Calculate new mask for each sub-block (subnet)
- 32 addresses in sub-block 1 → 5 bits reqd
new mask = /27
- 16 addresses in sub-blocks 2 and 3 → 4 bits reqd /28

Figure 19.7 Configuration and addresses in a subnetted network





Example 19.10

An organization is given a block of addresses starting with 172.17.14.15/23

The organization has 3 labs and needs to distribute these addresses into three sub-blocks of 255, 125, and 63 addresses.

10101100 00010001 00001110 00001111

00000000 00000000 00000001 11111111

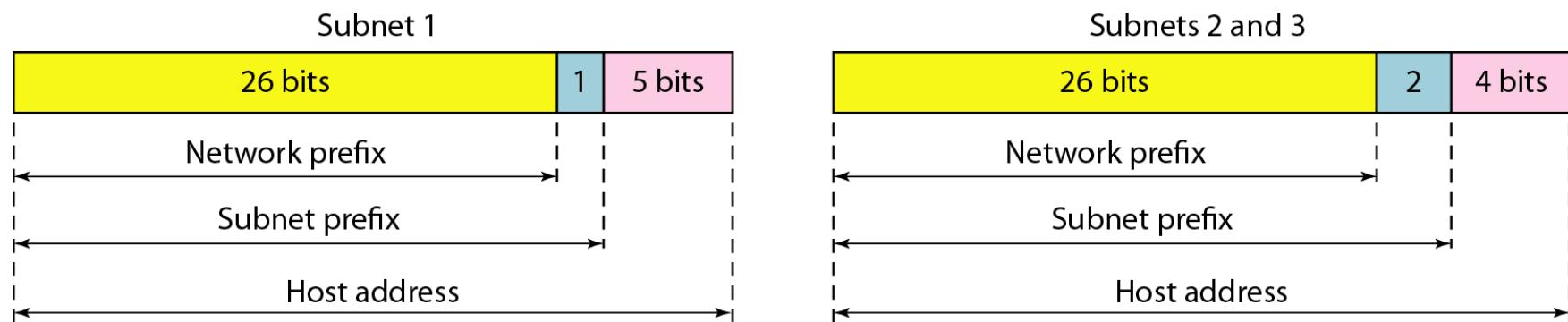
Ist address → 172.17.14.0

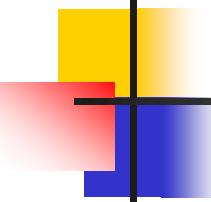
Last address → 172.17.15.255

Solution:

- *Given mask is /23 → 512 addresses*
- *Calculate new mask for each sub-block (subnet)*
- *255 addresses in sub-block 1 → 8 bits reqd, new mask*

Figure 19.8 *Three-level hierarchy in an IPv4 address*





Example 19.10

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

- a. The first group has 64 customers; each needs 256 addresses.*
- b. The second group has 128 customers; each needs 128 addresses.*
- c. The third group has 128 customers; each needs 64 addresses.*

Design the subblocks and find out how many addresses are still available after these allocations.

Example 19.10 (continued)

Solution

Figure 19.9 shows the situation.

Group 1

For this group, each customer needs 256 addresses. This means that $8 (\log_2 256)$ bits are needed to define each host. The prefix length is then $32 - 8 = 24$. The addresses are

1st Customer: 190.100.0.0/24 190.100.0.255/24

2nd Customer: 190.100.1.0/24 190.100.1.255/24

...

64th Customer: 190.100.63.0/24 190.100.63.255/24

Total = $64 \times 256 = 16,384$

Example 19.10 (continued)

Group 2

For this group, each customer needs 128 addresses. This means that 7 ($\log_2 128$) bits are needed to define each host. The prefix length is then $32 - 7 = 25$. The addresses are

1st Customer: 190.100.64.0/25 190.100.64.127/25

2nd Customer: 190.100.64.128/25 190.100.64.255/25

...

128th Customer: 190.100.127.128/25 190.100.127.255/25

Total = $128 \times 128 = 16,384$

Example 19.10 (continued)

Group 3

For this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host. The prefix length is then $32 - 6 = 26$. The addresses are

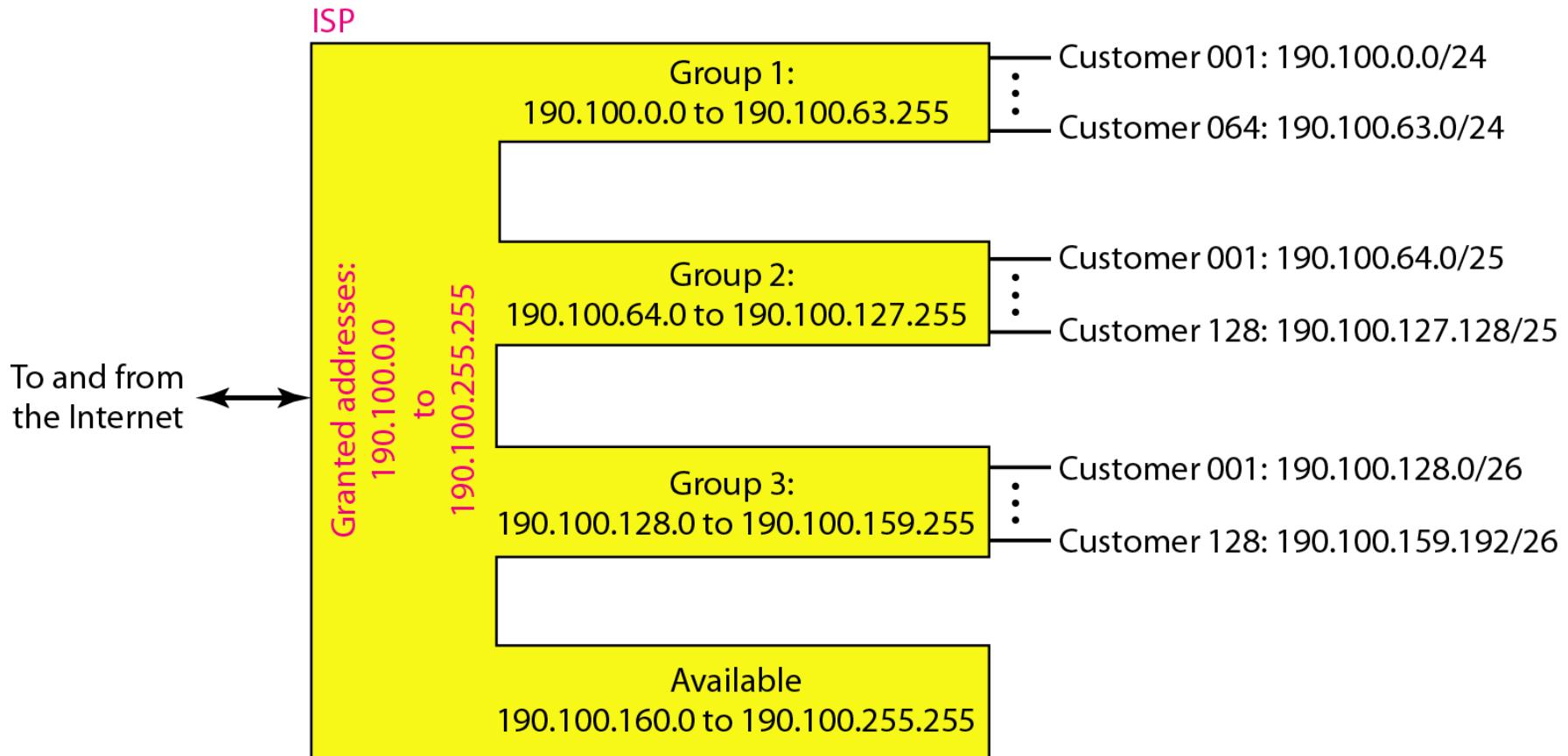
<i>1st Customer:</i>	$190.100.128.0/26$	$190.100.128.63/26$
<i>2nd Customer:</i>	$190.100.128.64/26$	$190.100.128.127/26$
...		
<i>128th Customer:</i>	$190.100.159.192/26$	$190.100.159.255/26$
<i>Total = $128 \times 64 = 8192$</i>		

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

Figure 19.9 An example of address allocation and distribution by an ISP



Network Address Translation (NAT)

- Shortage of addresses
- NAT enables a user to have a **large set of addresses internally and one address/ small set of addresses externally**

Table 19.3 Addresses for private networks (Private IP address range)

<i>Range</i>		<i>Total</i>
10.0.0.0	to	2^{24}
172.16.0.0	to	2^{20}
192.168.0.0	to	2^{16}

Figure 19.10 A NAT implementation

Site using private addresses

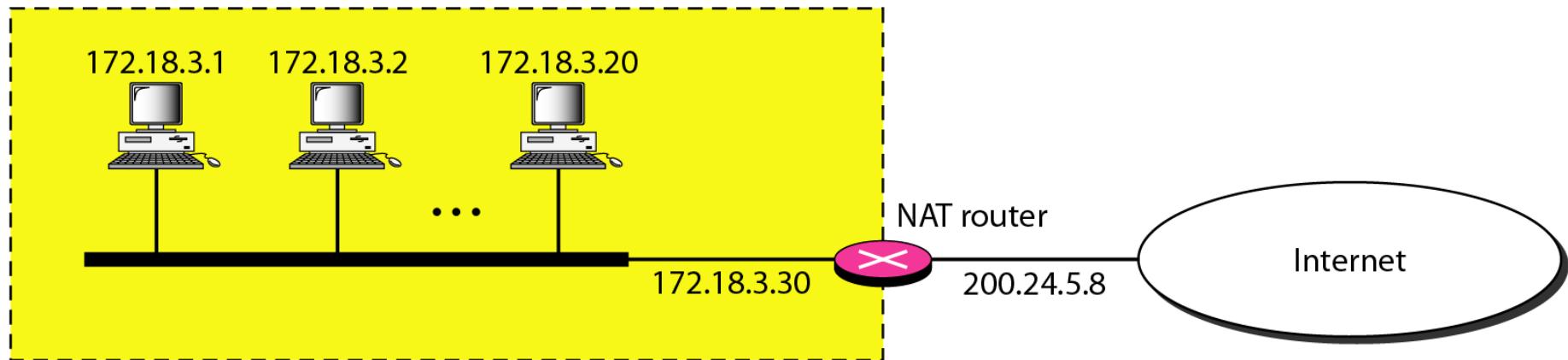


Figure 19.11 Addresses in a NAT

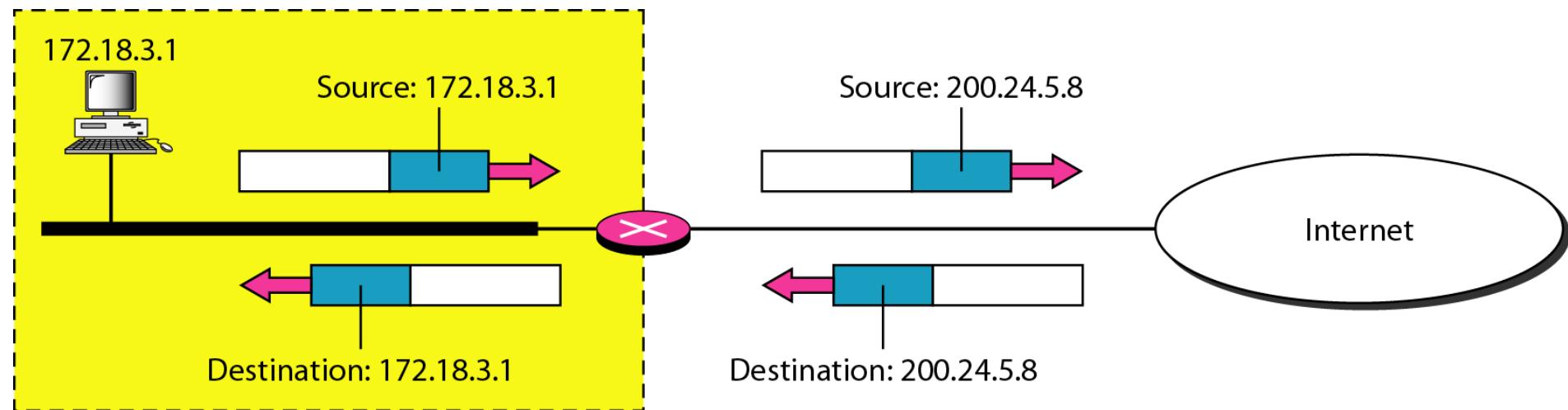
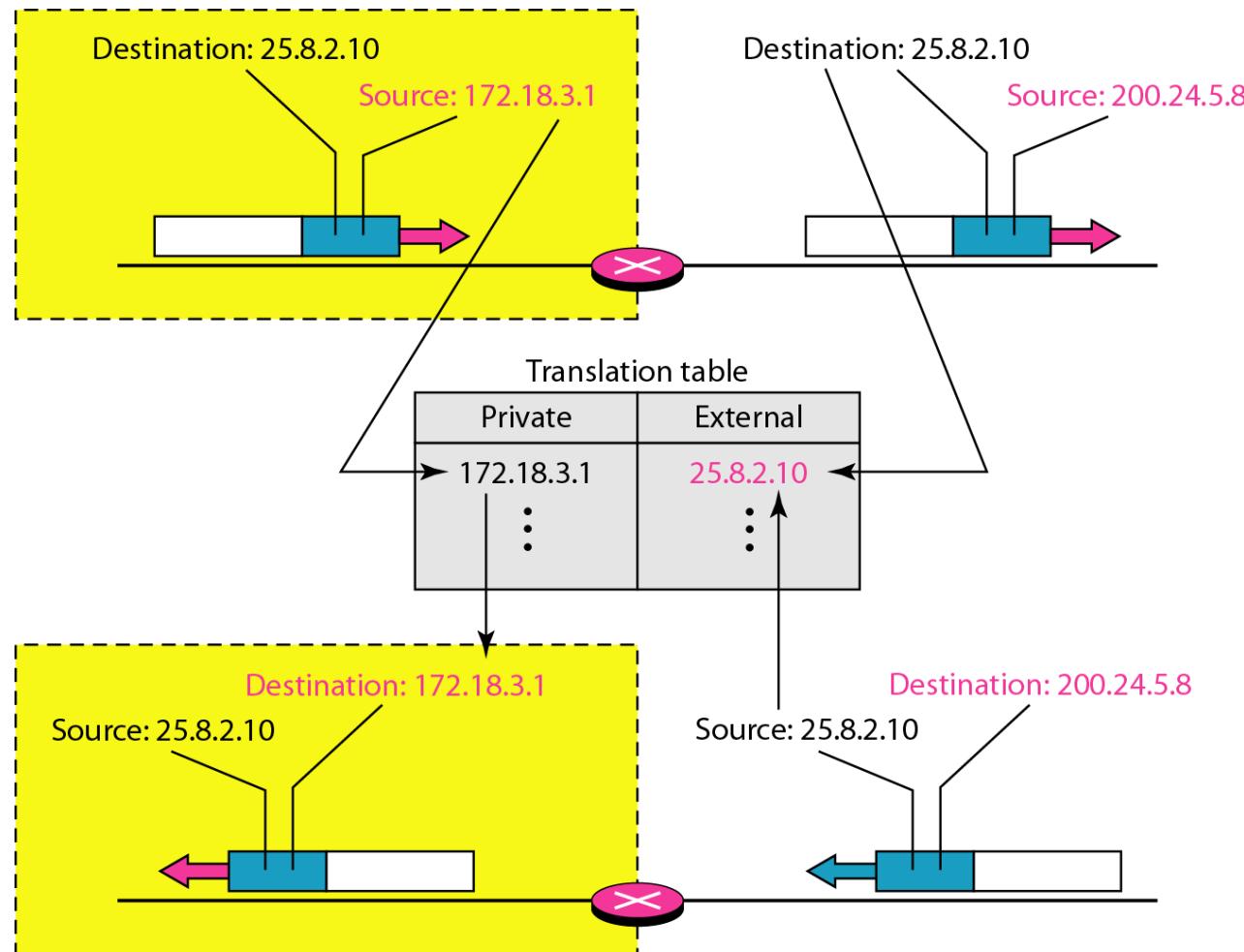


Figure 19.12 NAT address translation



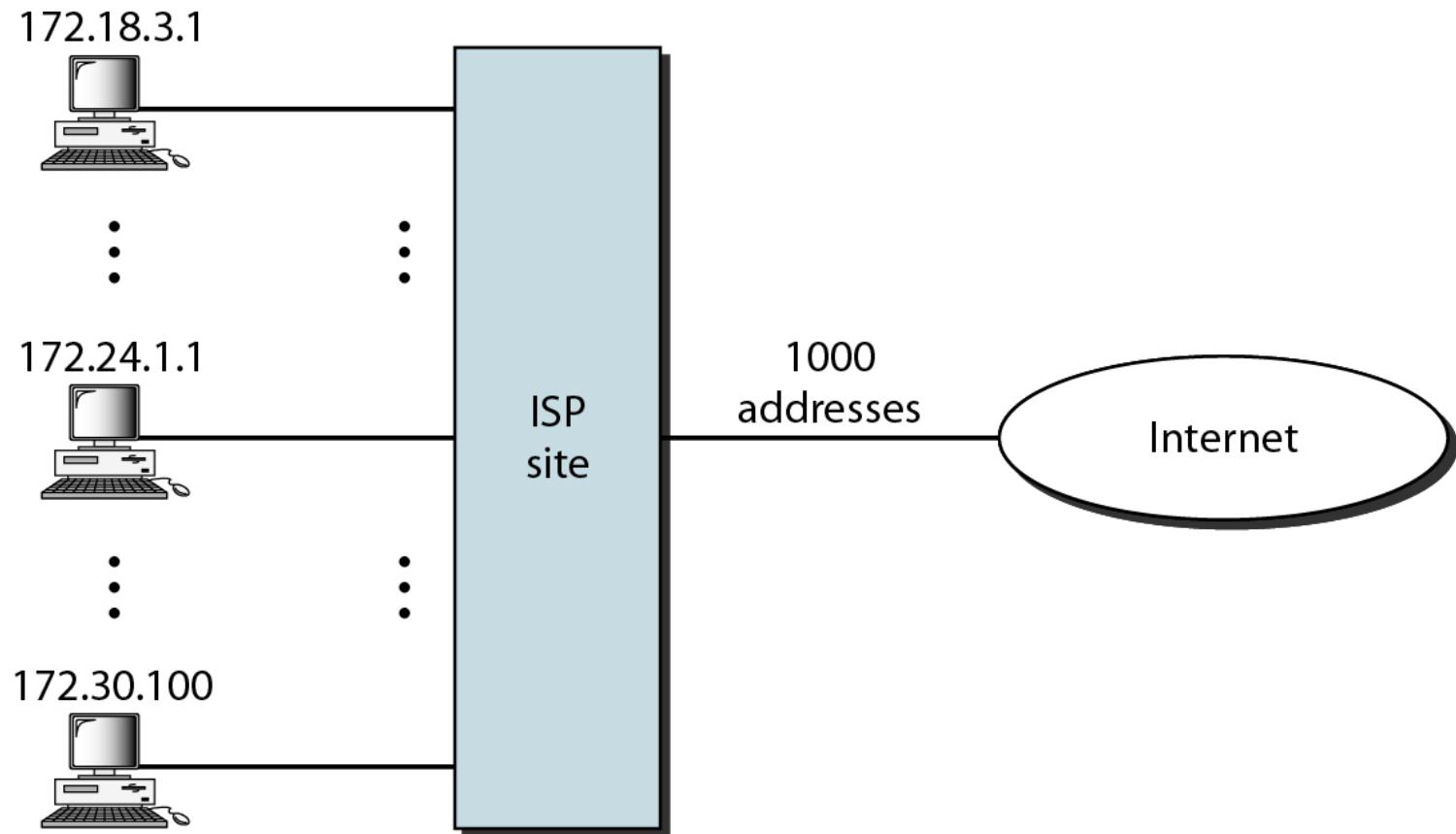
NAT...

- If NAT router has only 1 global address, only 1 private network host can access same external host
- To allow many to many relationship, more information reqd in translation table
- Combination of source address and port number defines the private network host

Table 19.4 *Five-column translation table*

<i>Private Address</i>	<i>Private Port</i>	<i>External Address</i>	<i>External Port</i>	<i>Transport Protocol</i>
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
...

Figure 19.13 An ISP and NAT



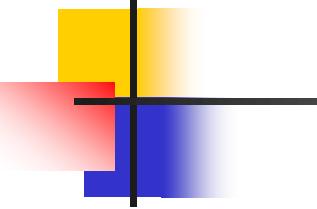
19-2 IPv6 ADDRESSES

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Topics discussed in this section:

Structure

Address Space



Note

An IPv6 address is 128 bits long.

Figure 19.14 IPv6 address in binary and hexadecimal colon notation

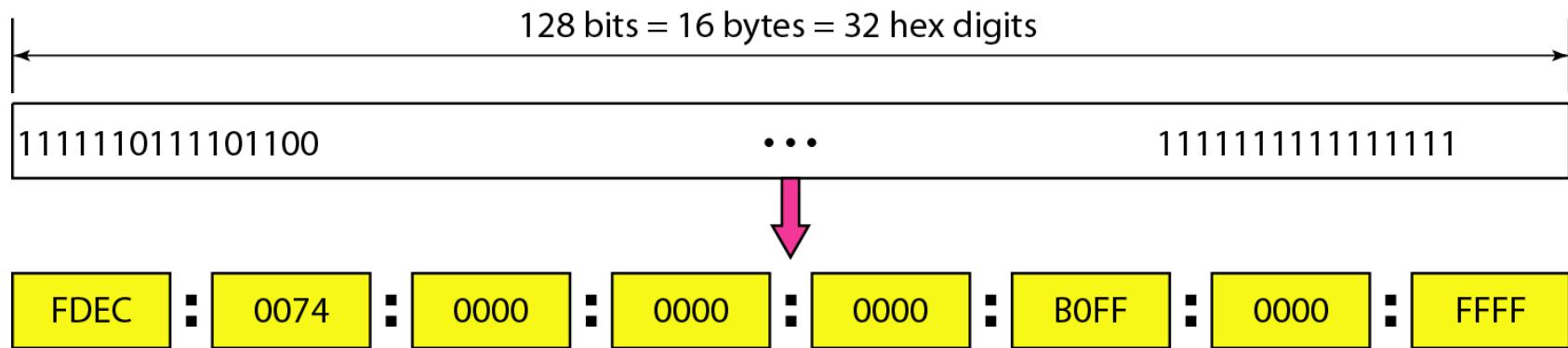
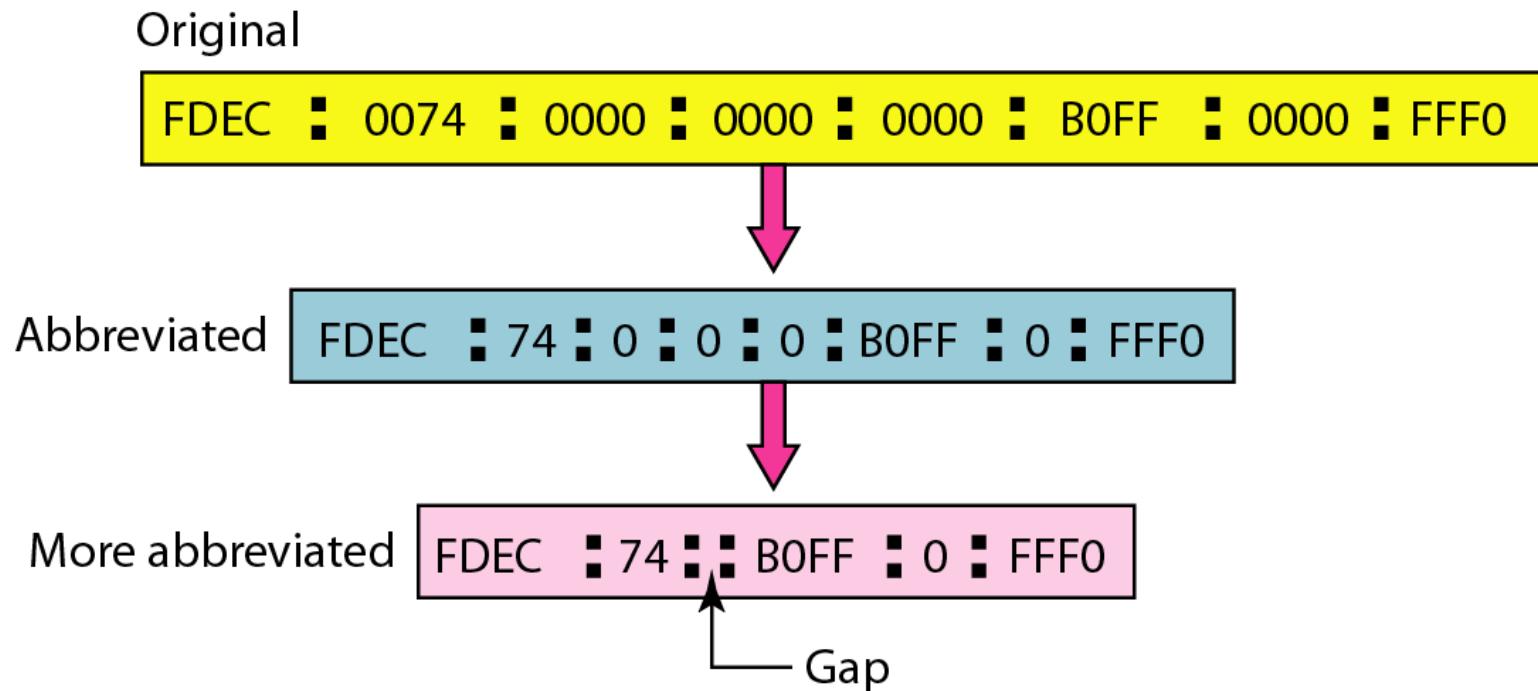


Figure 19.15 Abbreviated IPv6 addresses



Example 19.11

Expand the address 0:15::1:12:1213 to its original.

Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find how many 0s we need to replace the double colon.

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

0: 15: : 1: 12:1213

This means that the original address is-

0000:0015:0000:0000:0000:0001:0012:1213

IPv6 Address Categories

Table 19.5 *Type prefixes for IPv6 addresses*

Type Prefix	Type	Fraction
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	Provider-based unicast addresses	1/8

Table 19.5 Type prefixes for IPv6 addresses (continued)

Type Prefix	Type	Fraction
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

Figure 19.16 Prefixes for provider-based unicast address

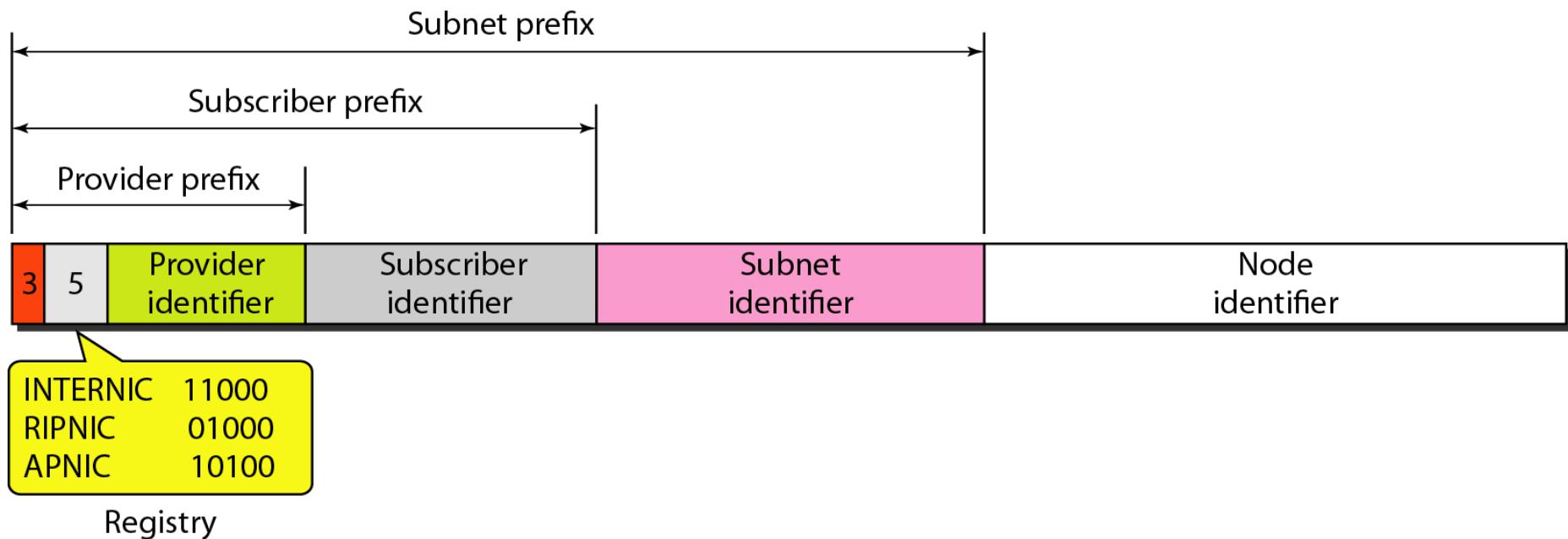


Figure 19.17 Multicast address and Anycast IPv6

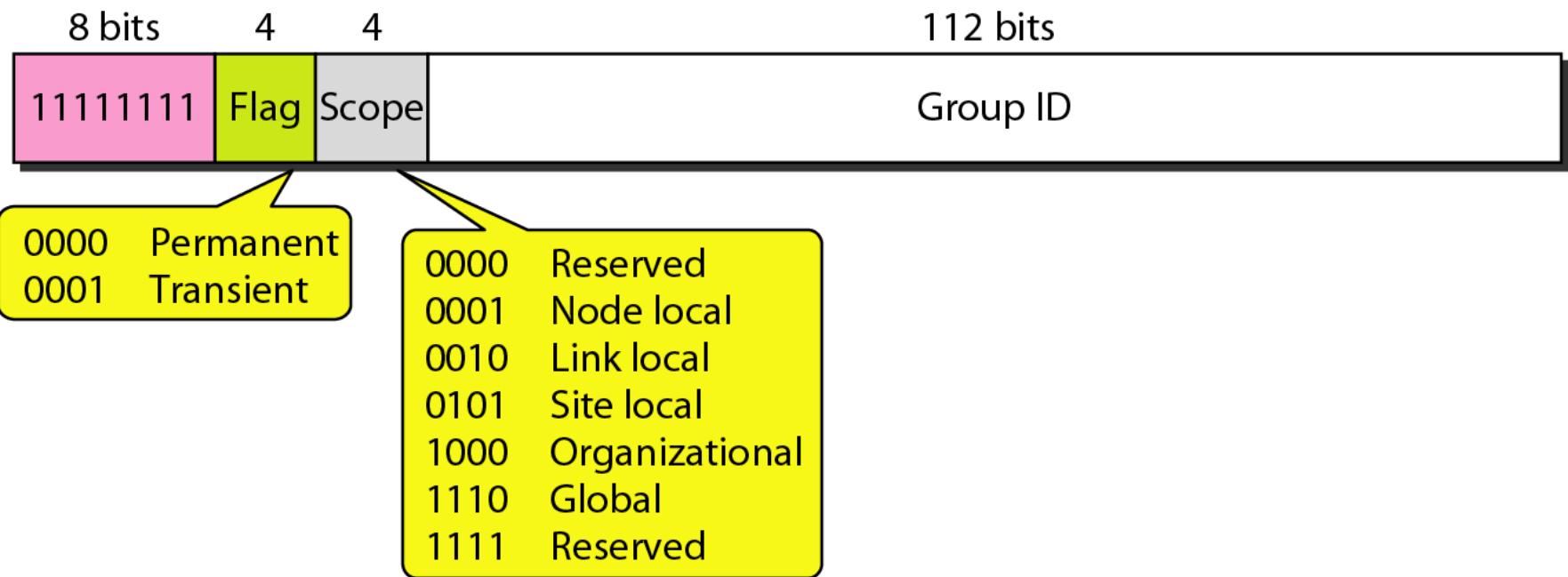


Figure 19.18 *Reserved addresses in IPv6*

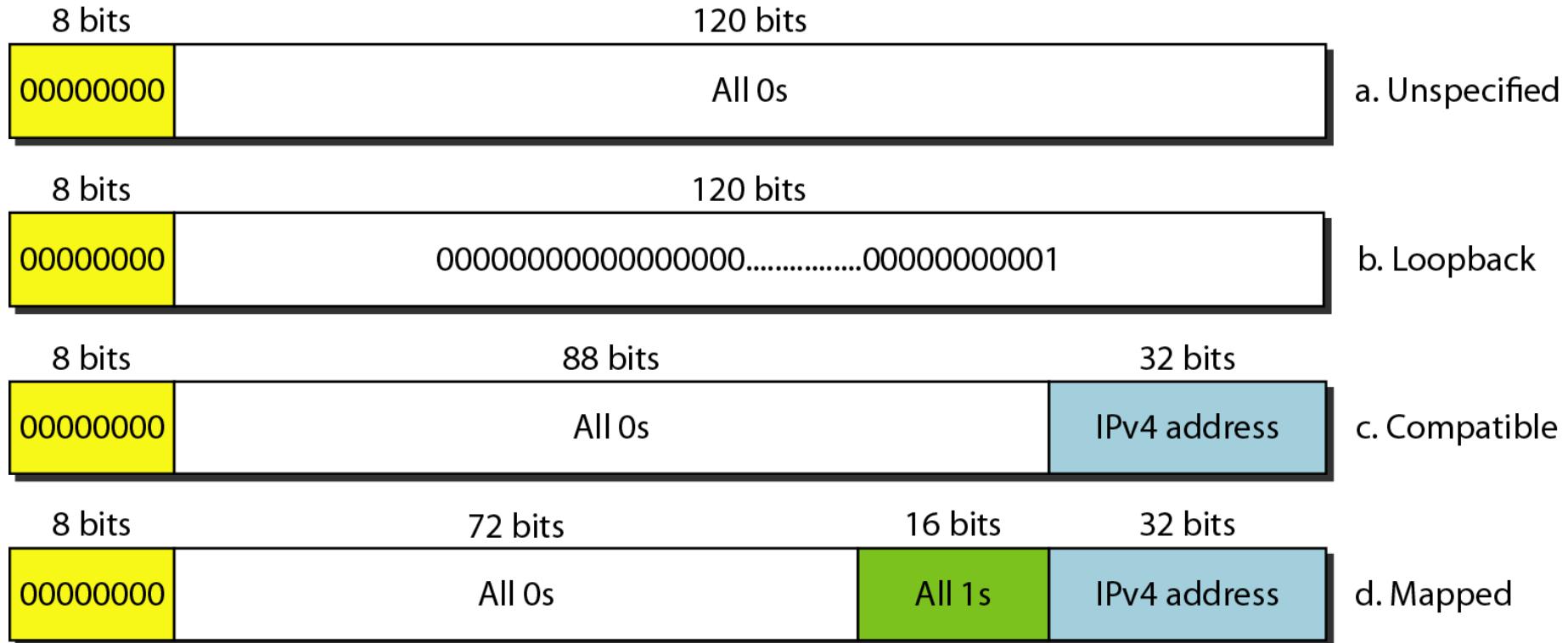
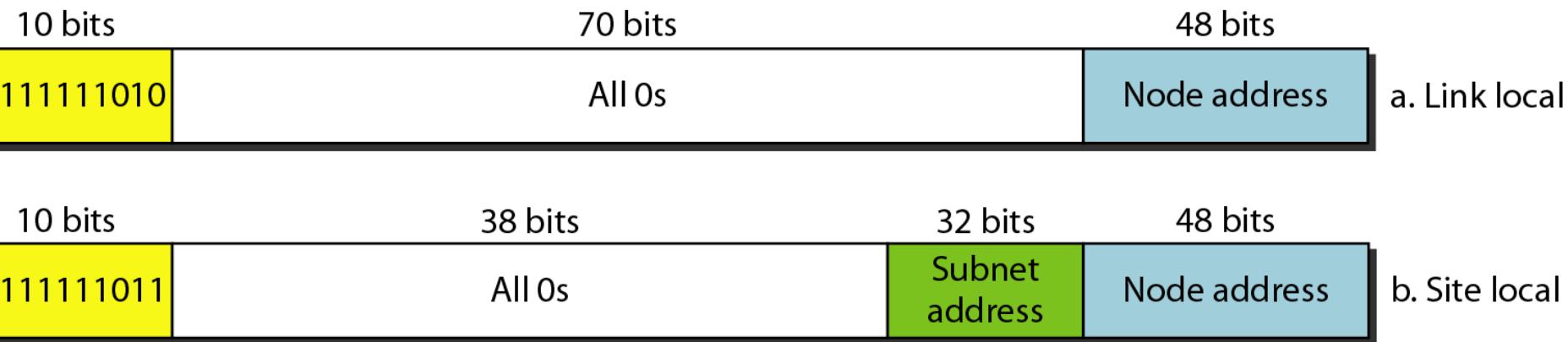


Figure 19.19 Local addresses in IPv6: (private addresses)



■ IPv4 Datagram Header

