Module 5

Logical Addressing, Internet Protocol, Address mapping and Error reporting, Delivery and forwarding, Unicast and multicast routing protocols

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.
- **IPv4 addresses are unique.** They are unique in the sense that each address defines one, and only one, connection to the Internet.
- Two devices on the Internet can never have the same address at the same time
- If a device operating at the network layer has *m* connections to the Internet, it needs to have *m* addresses.
- The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

- A protocol such as IPv4 that defines addresses has an address space.
- An **address space** is the total number of addresses used by the protocol.
- If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) and N bits can have 2^N values.
- **IPv4 uses 32-bit addresses**, which means that the **address space is 2^{32}** or 4,294,967,296.

Notations

There are two prevalent notations to show an IPv4 address: **binary notation and dotted decimal notation.**

Binary Notation

- In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte.
- So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address.
- The following is an example of an IPv4 address in binary notation:

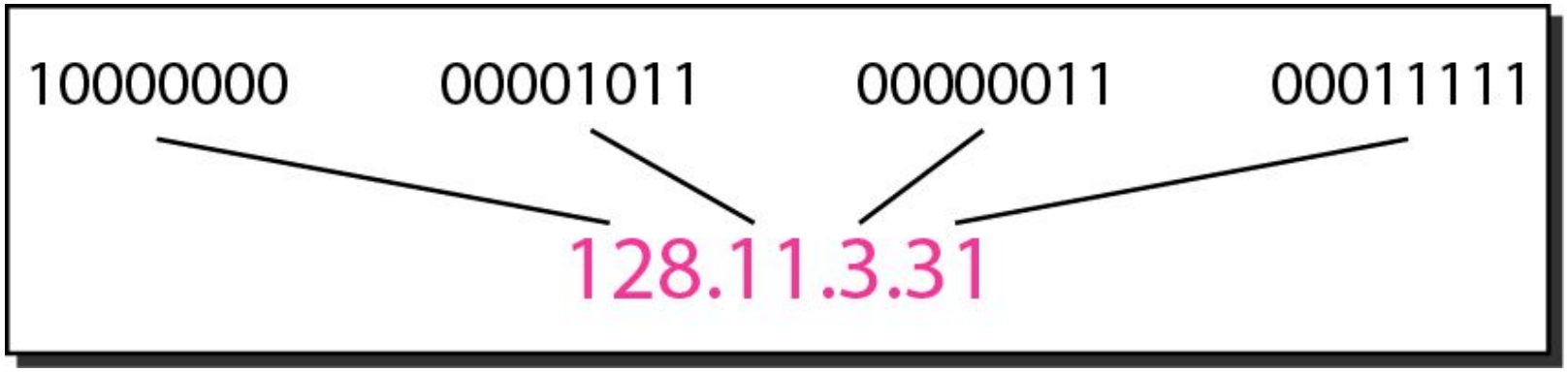
  01110101 10010101 00011101 00000010

Dotted-Decimal Notation

- To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes.
- The following is the dotted-decimal notation of the above address:

  117.149.29.2

Dotted-decimal notation and binary notation for an IPv4 address



Example

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

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

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

a. 111.56.45.78

b. 221.34.7.82

Example

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

a. 111.56.045.78

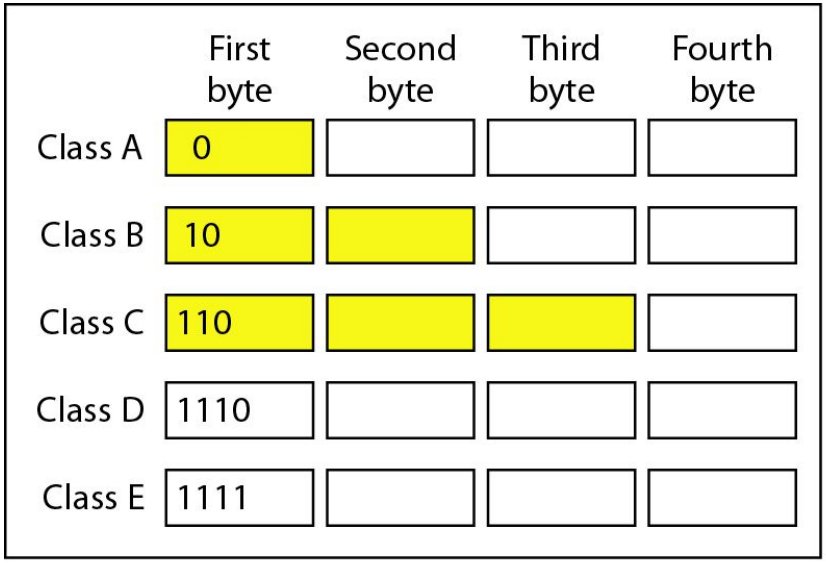
b. 221.34.7.8.20

c. 75.45.301.14

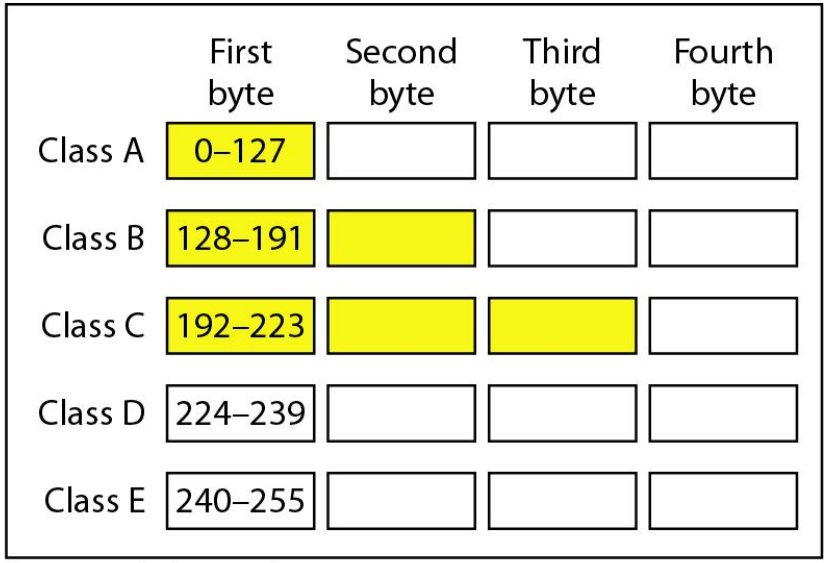
d. 11100010.23.14.67

Classful Addressing

- IPv4 addressing, at its inception, used the concept of classes. This architecture is called **classful addressing**.
- In classful addressing, the address space is divided into five classes: **A, B, C, D, and E.** Each class occupies some part of the address space.



a. Binary notation



b. Dotted-decimal notation

Example

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

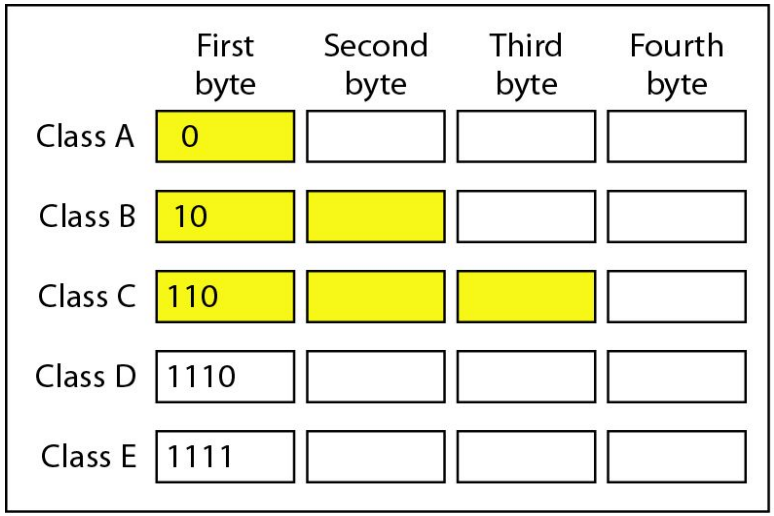
Classes and Blocks

- One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size, a large part of the available addresses were wasted.
- **Classes A,B,C – unicast communication** - from one source to one destination.
- A host needs to have at least one unicast address to be able to send or receive packets.
- Addresses **in class D are for multicast communication** from one source to a group of destinations. A multicast address can be used only as a destination address but never as a source address.
- Addresses in **class E are reserved** – used for special purpose.

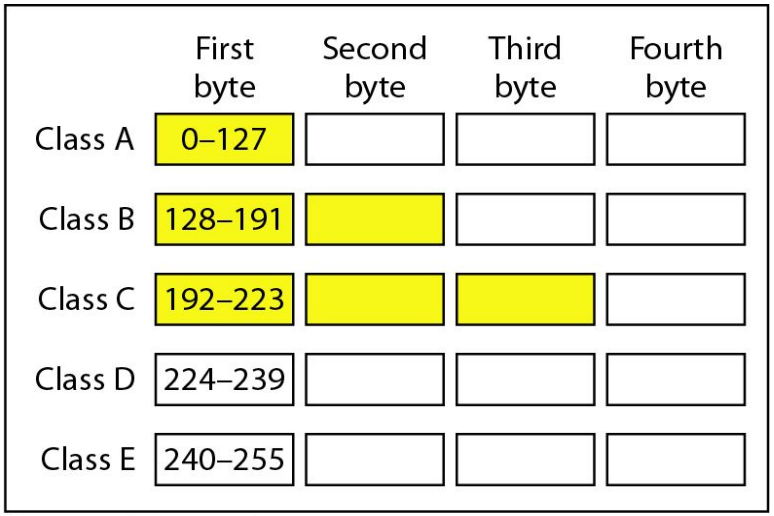
Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Netid and Hostid

- In classful addressing, an IP address in class A, B, or C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address.
- The netid is in color, the hostid is in white.
- It does not apply to classes D and E.
- In **class A, one byte** defines the **netid** and **three bytes** define the **hostid**.
- In **class B**, **two bytes** define the **netid** and **two bytes** define the **hostid**.
- In **class C**, **three bytes** define the **netid** and **one byte** defines the **hostid**.



a. Binary notation



b. Dotted-decimal notation

Mask

- The length of the netid and hostid (in bits) is predetermined in classful addressing, can also use a mask (also called the default mask), a 32-bit number made of contiguous 1s followed by contiguous 0s.
- **The mask can help us to find the netid and the hostid.** For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.
- The last column of Table shows the mask in the form */n* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called slash notation or **Classless Interdomain Routing** (CIDR) notation. The notation is used in **classless 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

Subnetting, Supernetting

- During the era of classful addressing, subnetting was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors. **Subnetting increases the number of 1s in the mask.**
- The time came when most of the class A and class B addresses were depleted; however, there was still a huge demand for midsize blocks.
- In **supernetting**, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a supernet.
- An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork.
- **Supernetting decreases the number of 1s in the mask.**

Classless Addressing

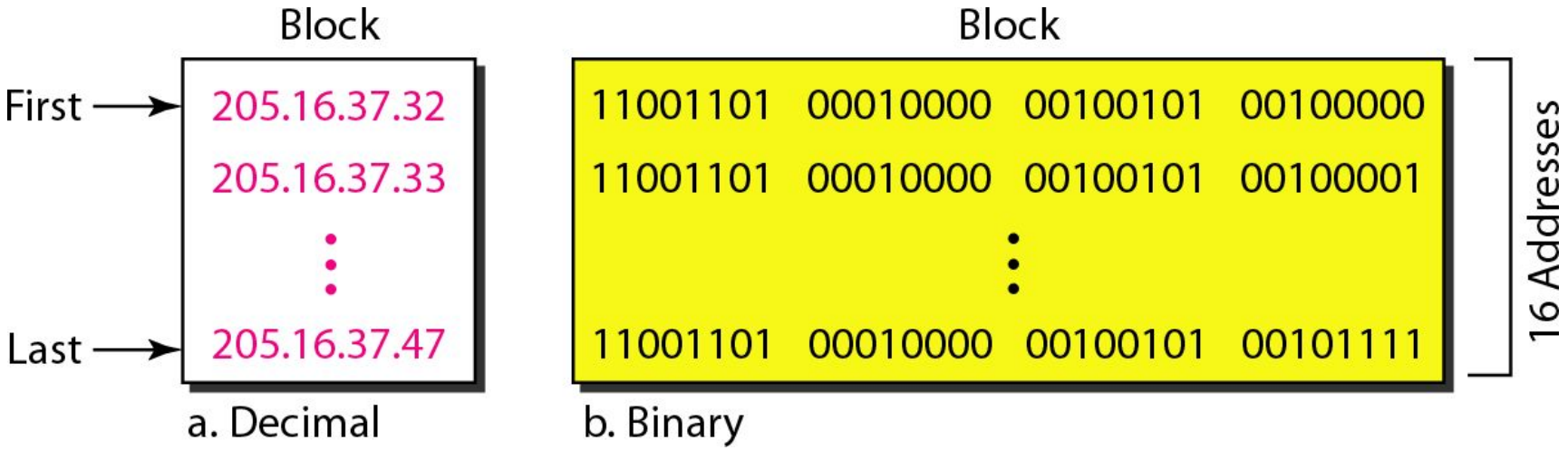
- To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

- In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is **granted a block (range) of addresses**. The **size of the block** (the number of addresses) varies based on the **nature and size of the entity**.
- An ISP, as the **Internet service provider**, may be given thousands or hundreds of thousands based on the number of customers it may serve.
- To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1. The **addresses in a block must be contiguous**, one after another.
2. The **number of addresses in a block must be a power of 2** (1, 2, 4, 8, ...).
3. The **first address must be evenly divisible** by the number of addresses.

A block of 16 addresses granted to a small organization



Network Addresses

- A very important concept in IP addressing is the network address. When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet.
- The **first address** in the class, however, is normally (not always) **treated as a special address**. The **first address** is called the **network address** and **defines the organization network.** It defines the organization itself to the rest of the world.

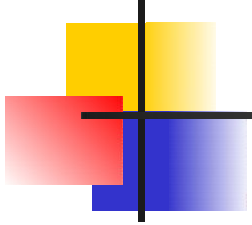
In IPv4 addressing, a block of addresses can be defined as

x.y.z.t /*n*

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

Note

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



Note

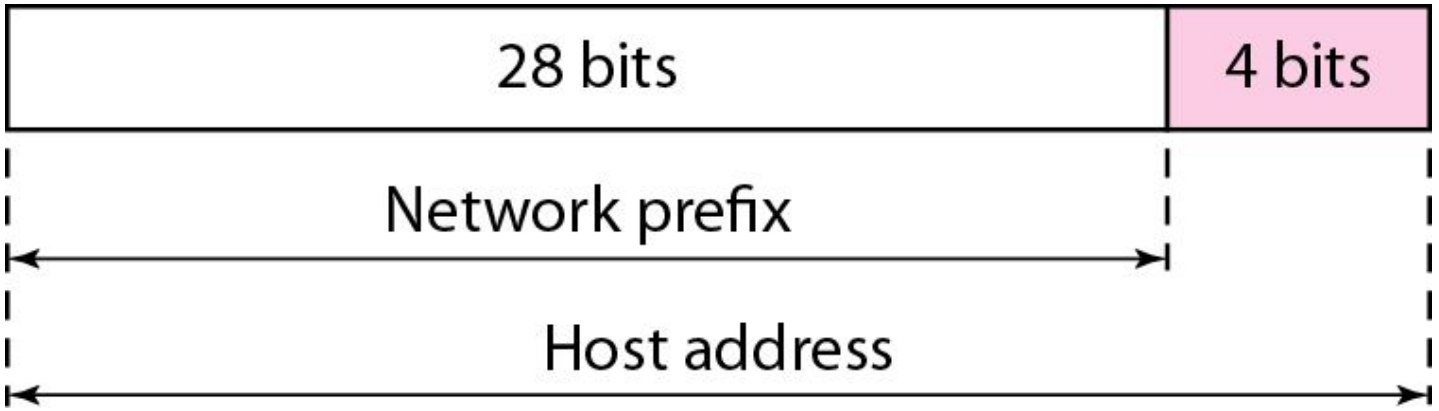
The last address in the block can be found by setting the rightmost 32 − n bits to 1s.

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.

Two-Level Hierarchy: No Subnetting

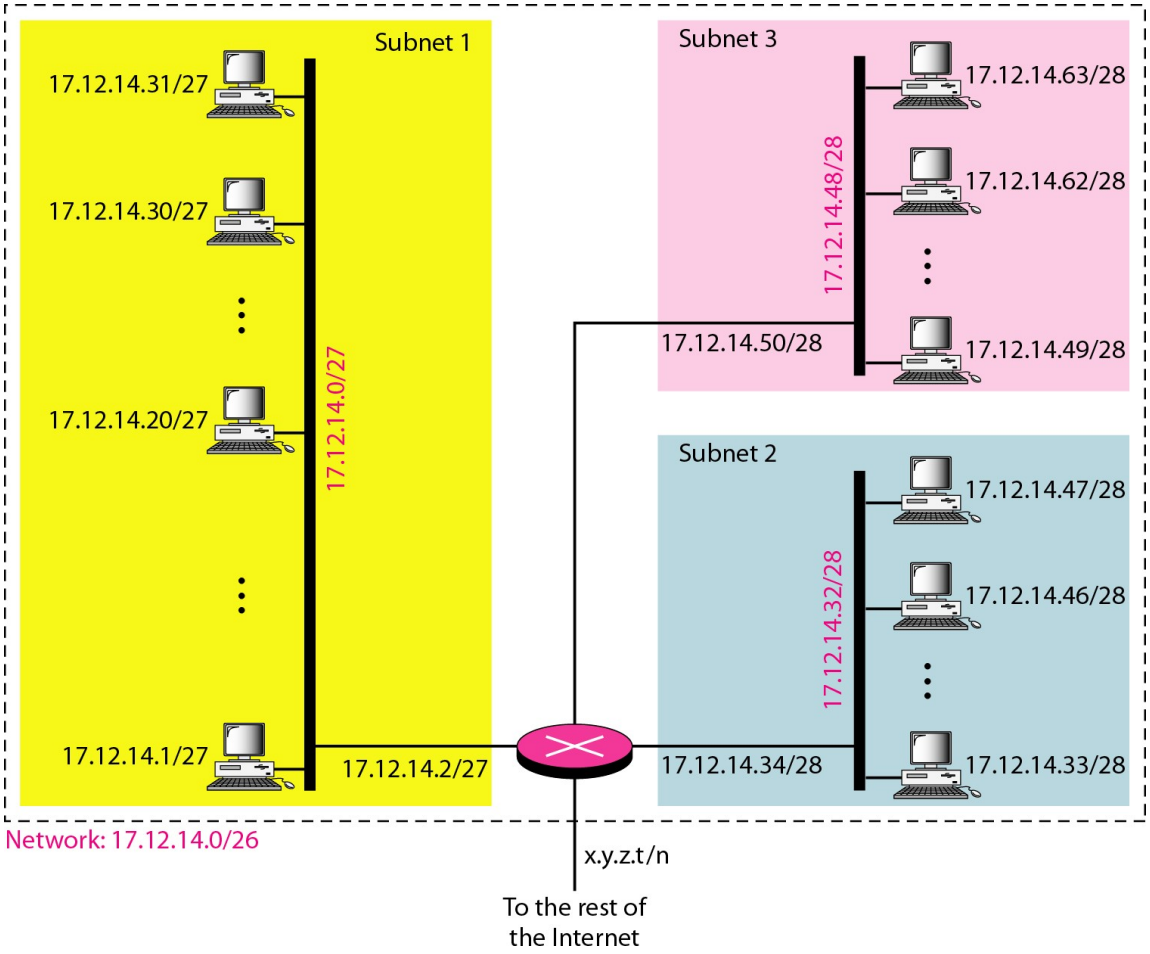
- An IP address can define only two levels of hierarchy when not subnetted.
- The part of the address that **defines the network is called the prefix**; the part that **defines the host is called the suffix.**
- The prefix is common to all addresses in the network; the suffix changes from one device to another.



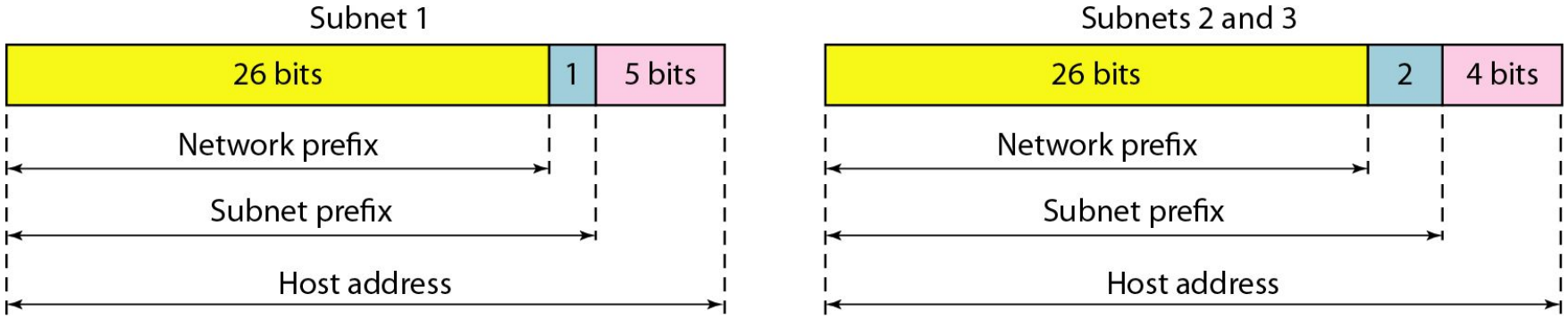
Note

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.

Figure 19.7 *Configuration and addresses in a subnetted network*



Three-level hierarchy in an IPv4 address



Address Allocation

The next issue in classless addressing is address allocation.

The ultimate responsibility of address allocation is given to a global authority called the *Internet Corporation for Assigned Names and Addresses* (ICANN).

However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP.

Each ISP, in turn, divides its assigned block into smaller subblocks and grants the subblocks to its customers. In other words, an ISP receives one large block to be distributed to its Internet users. This is called **address aggregation**: many blocks of addresses are aggregated in one block and granted to one ISP.

Network Address Translation(NAT)

- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.
- To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses.
- Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private networks.
- They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address. The site must have only one single connection to the global Internet through a router that runs the NAT software.

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

A NAT implementation

Address Translation

All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address

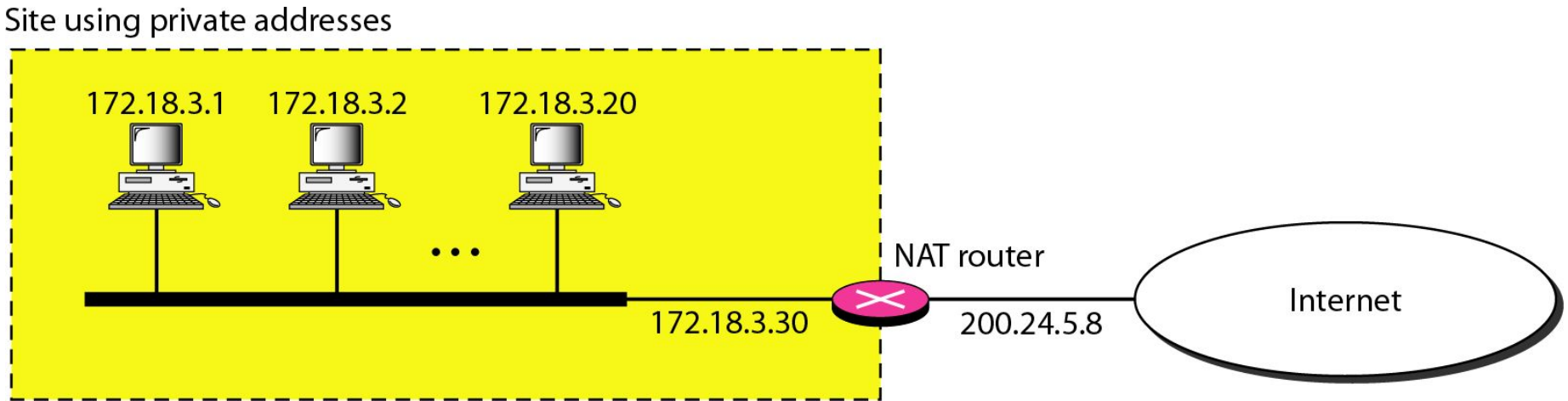


Figure 19.11 *Addresses in a NAT*

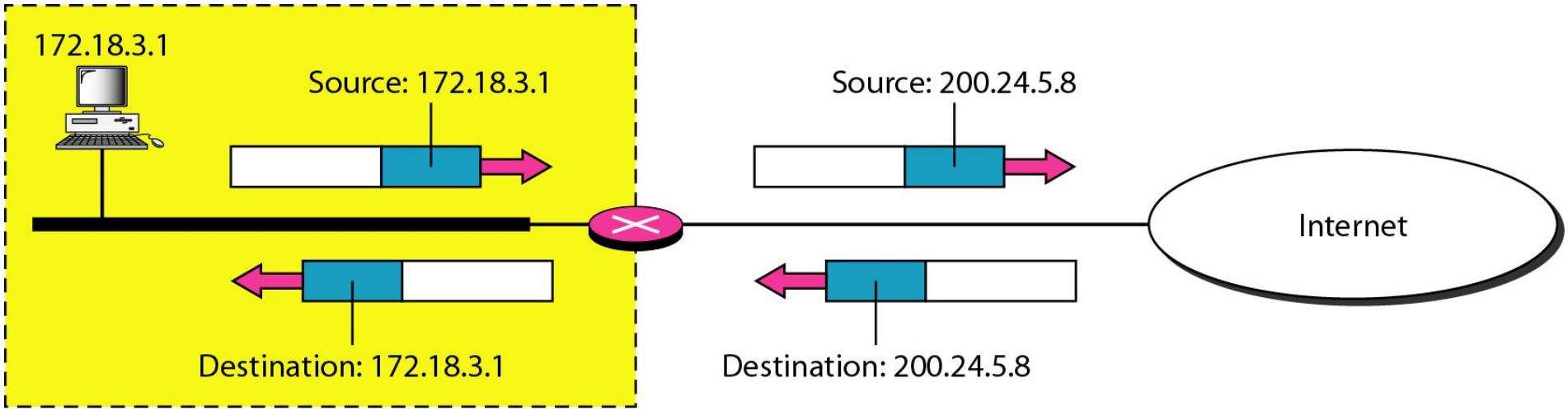


Figure 19.12 *NAT address translation*

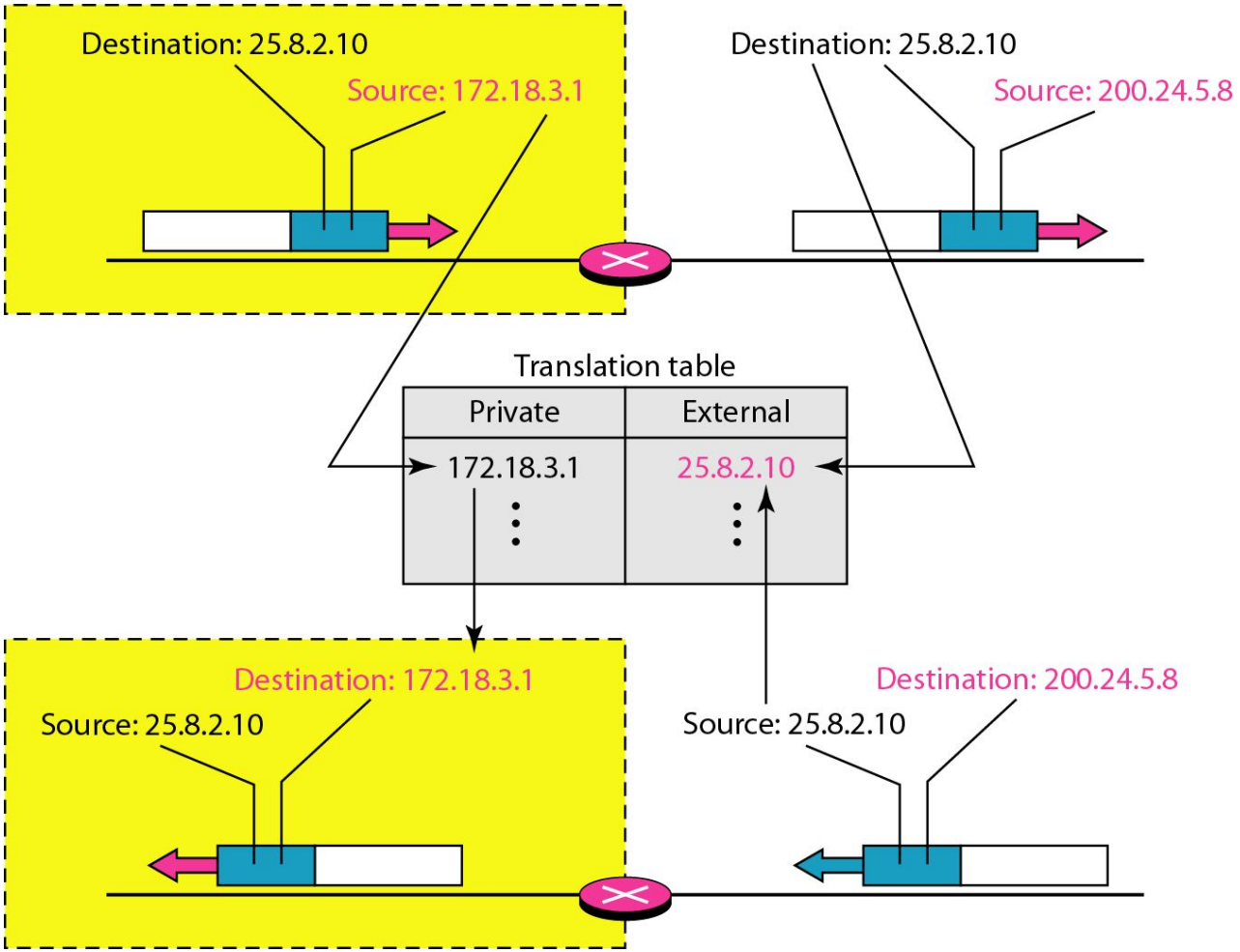
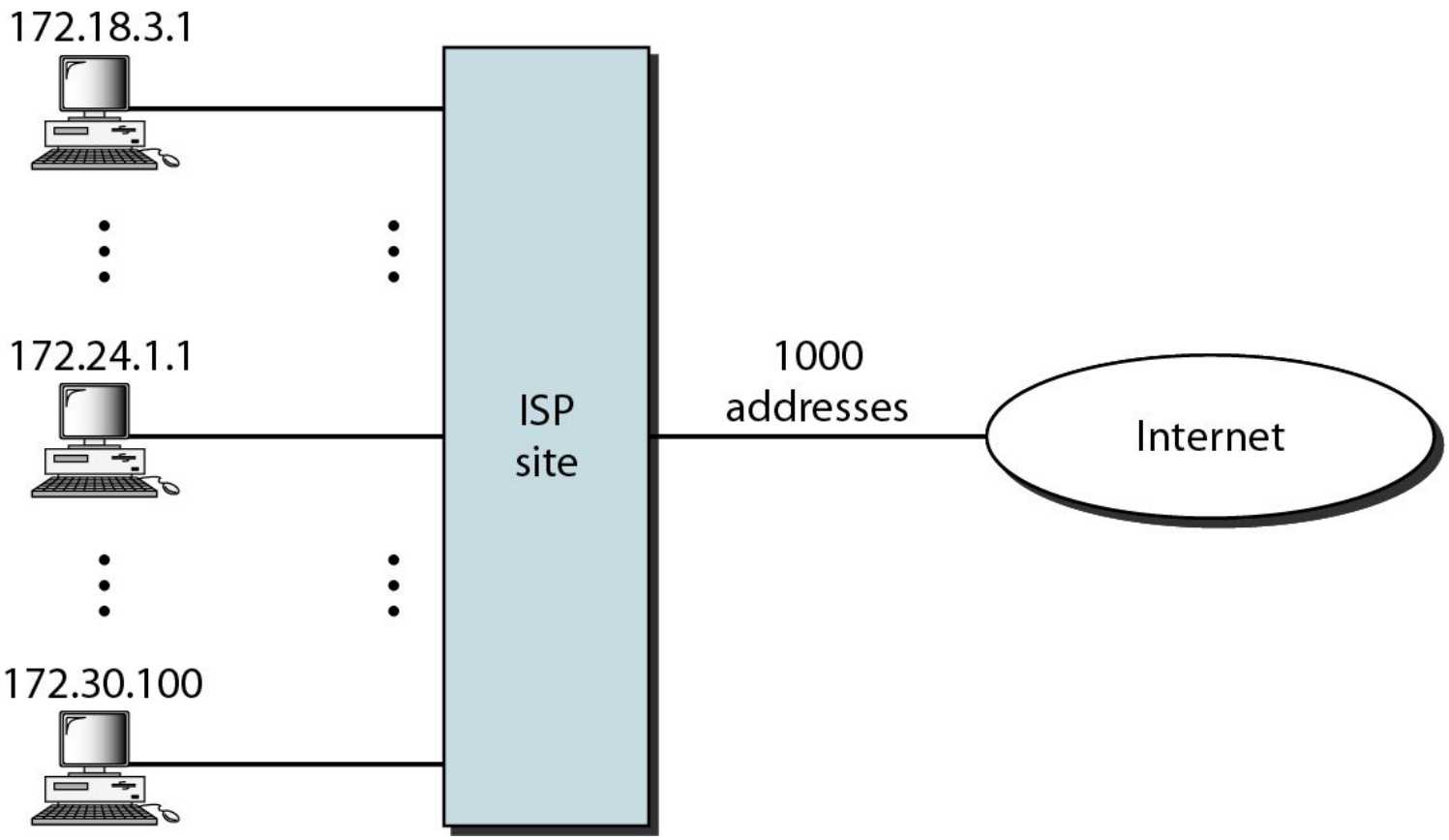


Table 19.4 ***Five-column translation table***

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
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*

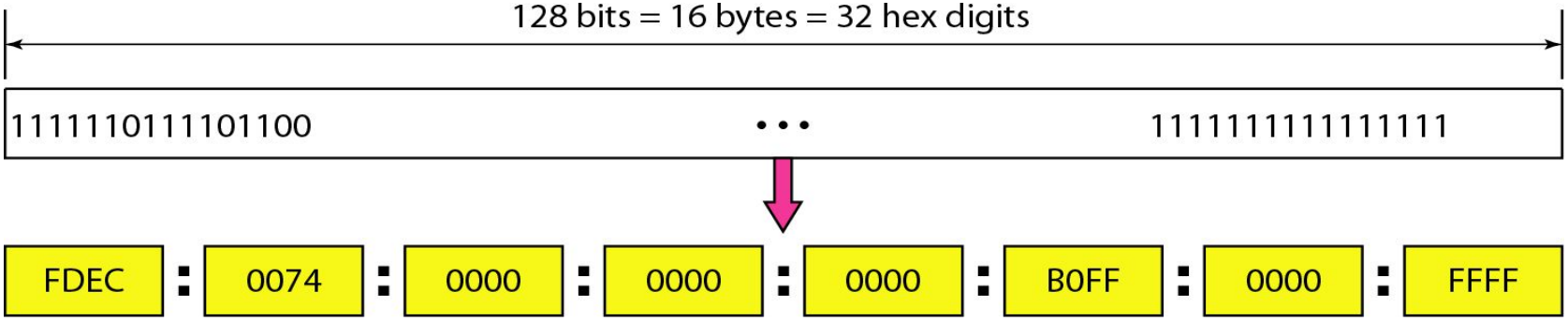


IPv6ADDRESSES

- An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

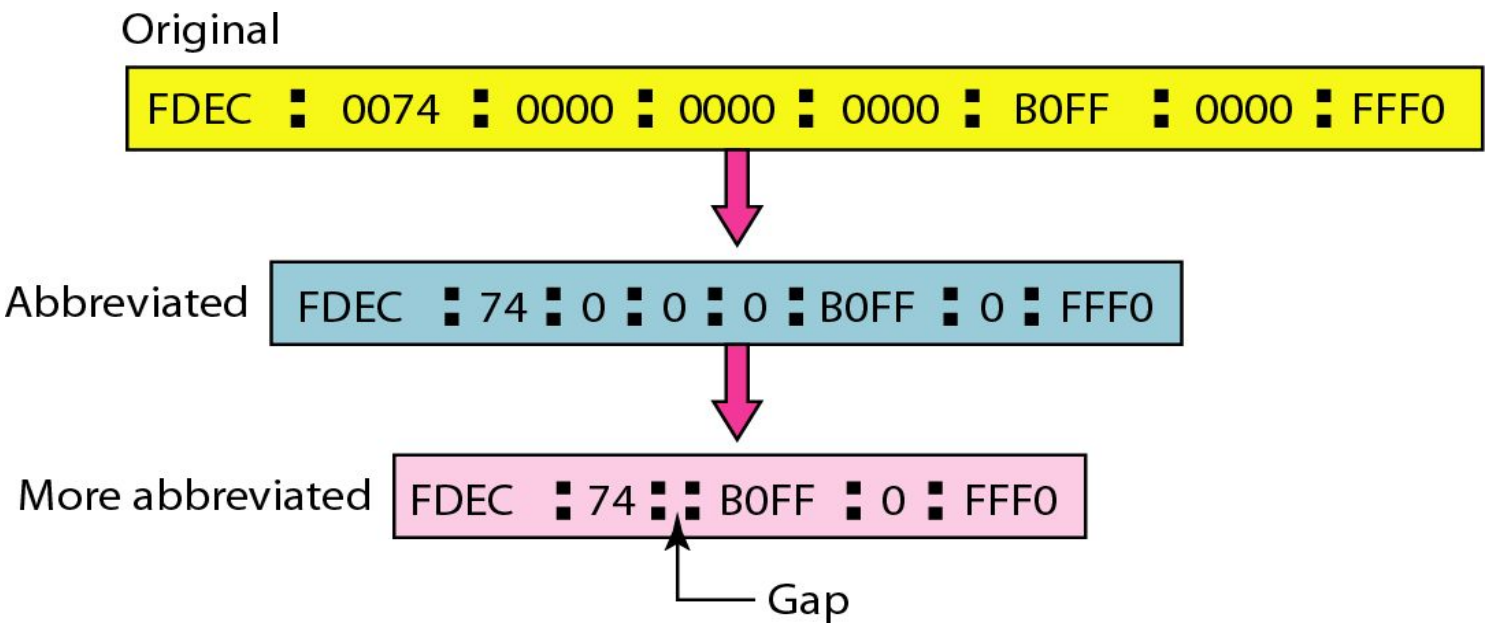
Hexadecimal Colon Notation

- To make addresses more readable, IPv6 specifies **hexadecimal colon notation.**
- In this notation, **128 bits** is divided into eight sections, each 2 bytes in length.
- Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



Abbreviated IPv6 addresses

- Although the IP address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address.
- The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros.



Example

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

```
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
```

Table 19.5 *Type prefixes for IPv6 addresses*

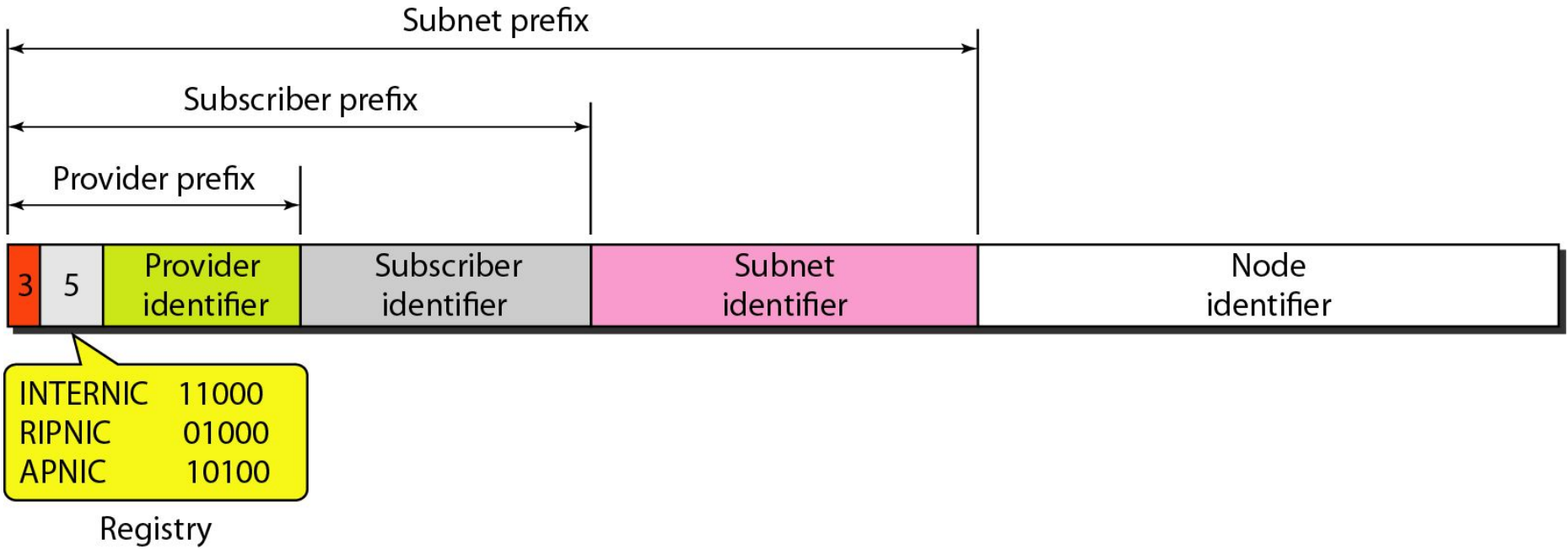
Type Prefix	*Type*	*Fraction*
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	**Provider-based unicast addresses**	**1/8**

Table 19.5 *Type prefixes for IPv6 addresses*

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

Unicast address

- A **unicast address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: **geographically based and provider-based**. The **provider-based address** is generally used by a **normal host as a unicast address**.
- **Type identifier.** This **3-bit field** defines the **address as a provider-base address.**
- **Registry identifier.** This **5-bit field** indicates the **agency that has registered the address**. Currently three registry centers have been defined. **INTERNIC (code 11000) is the center for North America; RIPNIC (code 01000) is the center for European registration; and APNIC (code 10100) is for Asian and Pacific countries**.
- **Provider identifier**. This **variable-length** field **identifies the provider for Internet access** (such as an ISP). **A 16-bit length** is recommended for this field.
- **Subscriber identifier**. When an **organization subscribes to the Internet through a provider**, it is assigned a **subscriber identification**. **A 24-bit length** is recommended for this field.
- **Subnet identifier**. Each **subscriber can have many different subnetworks**, and each subnetwork can have an identifier. The subnet identifier defines a specific subnetwork under the territory of the subscriber. A **32-bit length** is recommended for this field.
- **Node identifier**. The last field defines **the identity of the node** connected to a subnet. A **length of 48 bits** is recommended for this field to make it compatible with the 48-bit link (physical) address used by Ethernet. In the future, this link address will probably be the same as the node physical address.


Subnet prefix
Subscriber prefix
Provider prefix
3
5
Provider identifier
Subscriber identifier
Subnet identifier
Node identifier
INTERNIC 11000
RIPNIC 01000
APNIC 10100
Registry

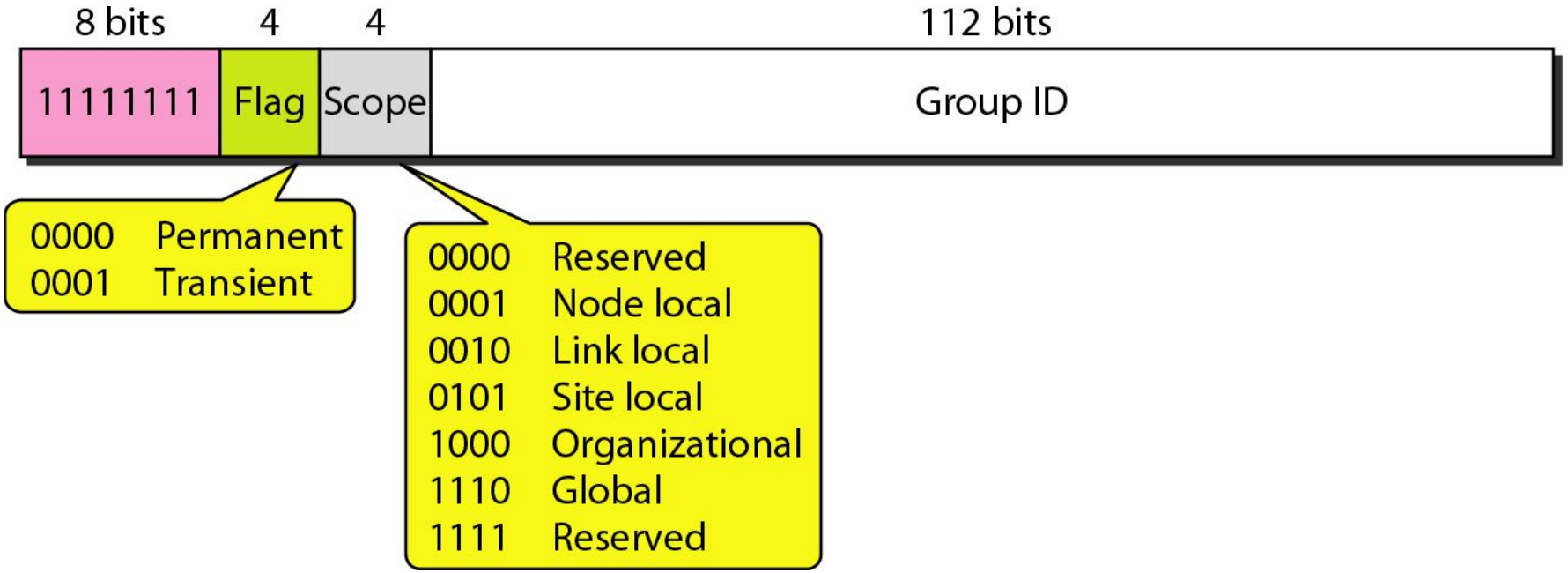
Multicast Addresses

Multicast addresses are used to define a **group of hosts** instead of just one. A packet sent to a multicast address must be delivered to each member of the group.

The second field is a flag that defines the group address as either **permanent or transient.** A **permanent group address** is defined by the **Internet authorities** and can be accessed at all times. A **transient group** address, on the other hand, is used only **temporarily.**

Systems engaged in a teleconference, for example, can use a transient group address.

The third field defines **the scope of the group address.**

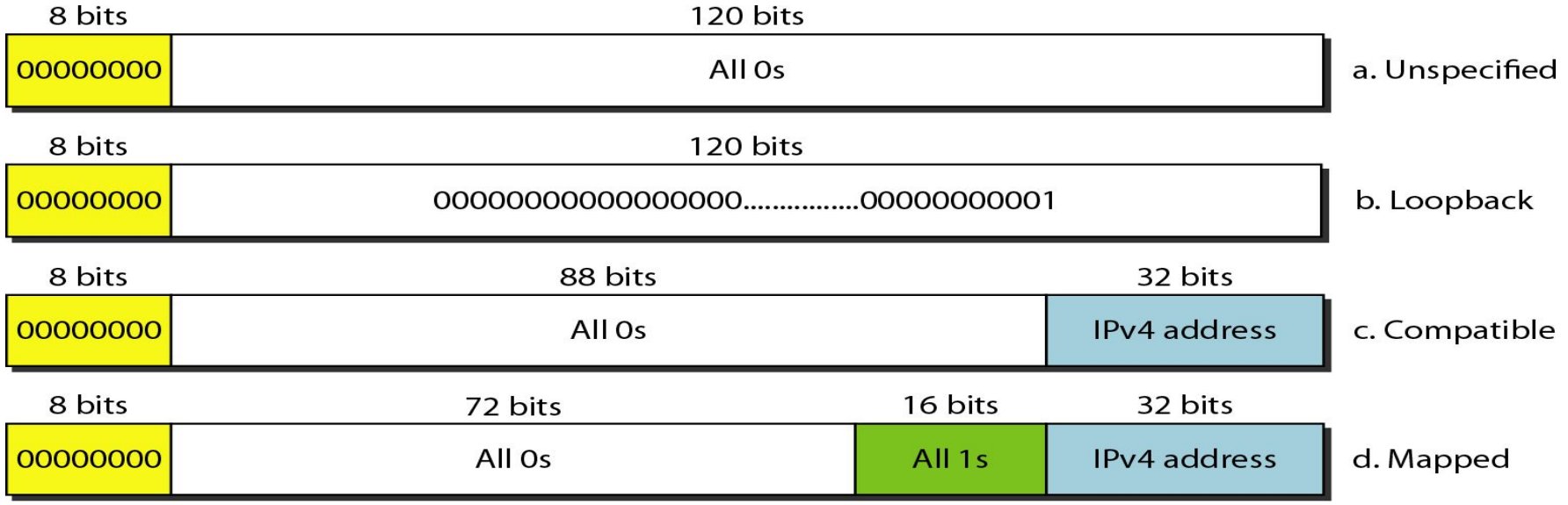


Anycast Addresses

- IPv6 also defines anycast addresses.
- An anycast address, like a multicast address, also defines a group of nodes.
- However, a packet destined for an anycast address is delivered to only **one of the members of the anycast group, the nearest one** (the one with the shortest route).
- one possible use is to assign an anycast address to all routers of an ISP that covers a large logical area in the Internet.
- The routers outside the ISP deliver a packet destined for the ISP to the nearest ISP router. No block is assigned for anycast addresses.

Reserved addresses in IPv6

Another category in the address space is the reserved address. These addresses start with eight 0s (type prefix is 00000000). An **unspecified address** is used when a **host does not know its own address** and sends an inquiry to find its address. A **loopback address** is used by a **host to test itself** without going into the network. A **compatible address** is used during the **transition from IPv4 to IPv6**. It is used when a computer using IPv6 wants to send a message to another computer using IPv6, but the message needs to pass through a part of the network that still operates in IPv4. A **mapped address** is also used during transition. However, it is used when a computer that has migrated to **IPv6** wants to send a packet to a computer still using **IPv4.**

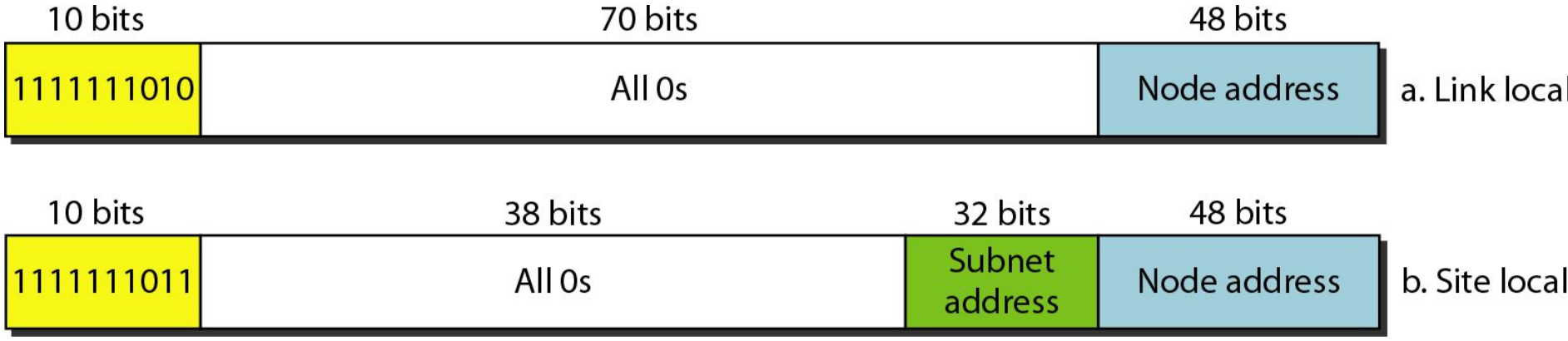


Local addresses in IPv6

These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet. In other words, they provide addressing for private networks.

Nobody outside the organization can send a message to the nodes using these addresses.

A **link local address is used in an isolated subnet**; a **site local address** is used in an **isolated site with several subnets**.



Internet Protocol

IP is the **host-to-host network layer delivery protocol** for the Internet.

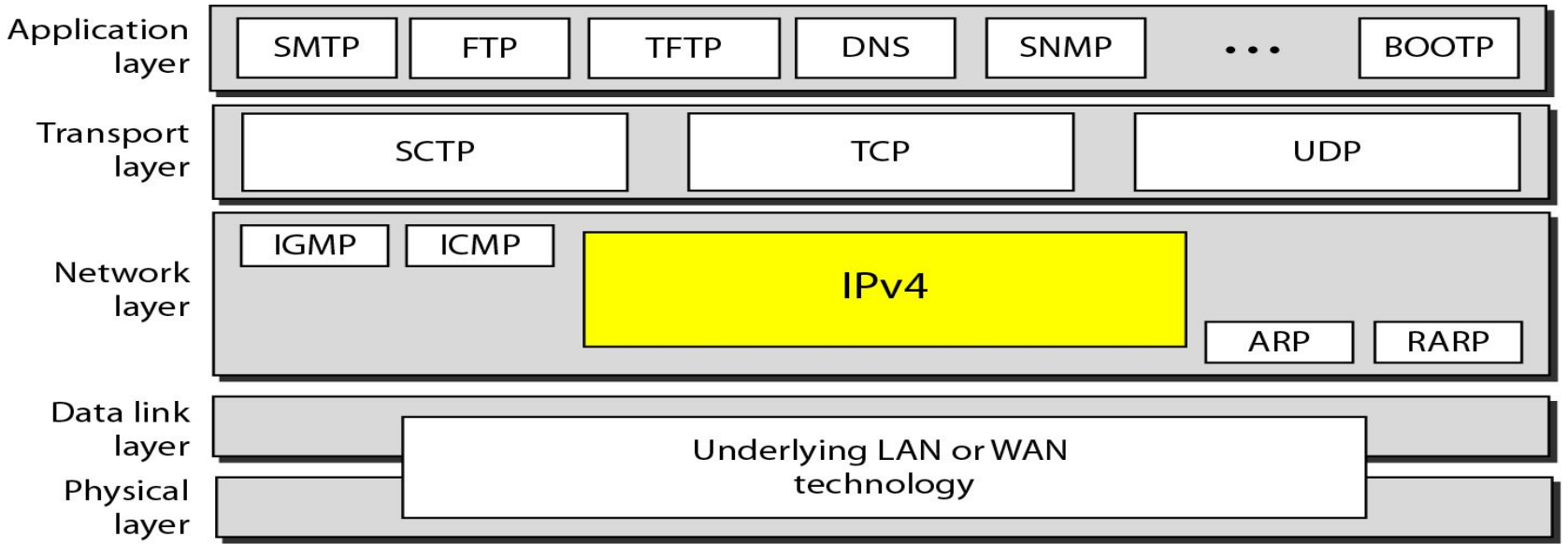
IP is **unreliable and connectionless datagram proto**col – a **best-effort delivery service.**

The term **best-effort** means that IP provides no error control or flow control.

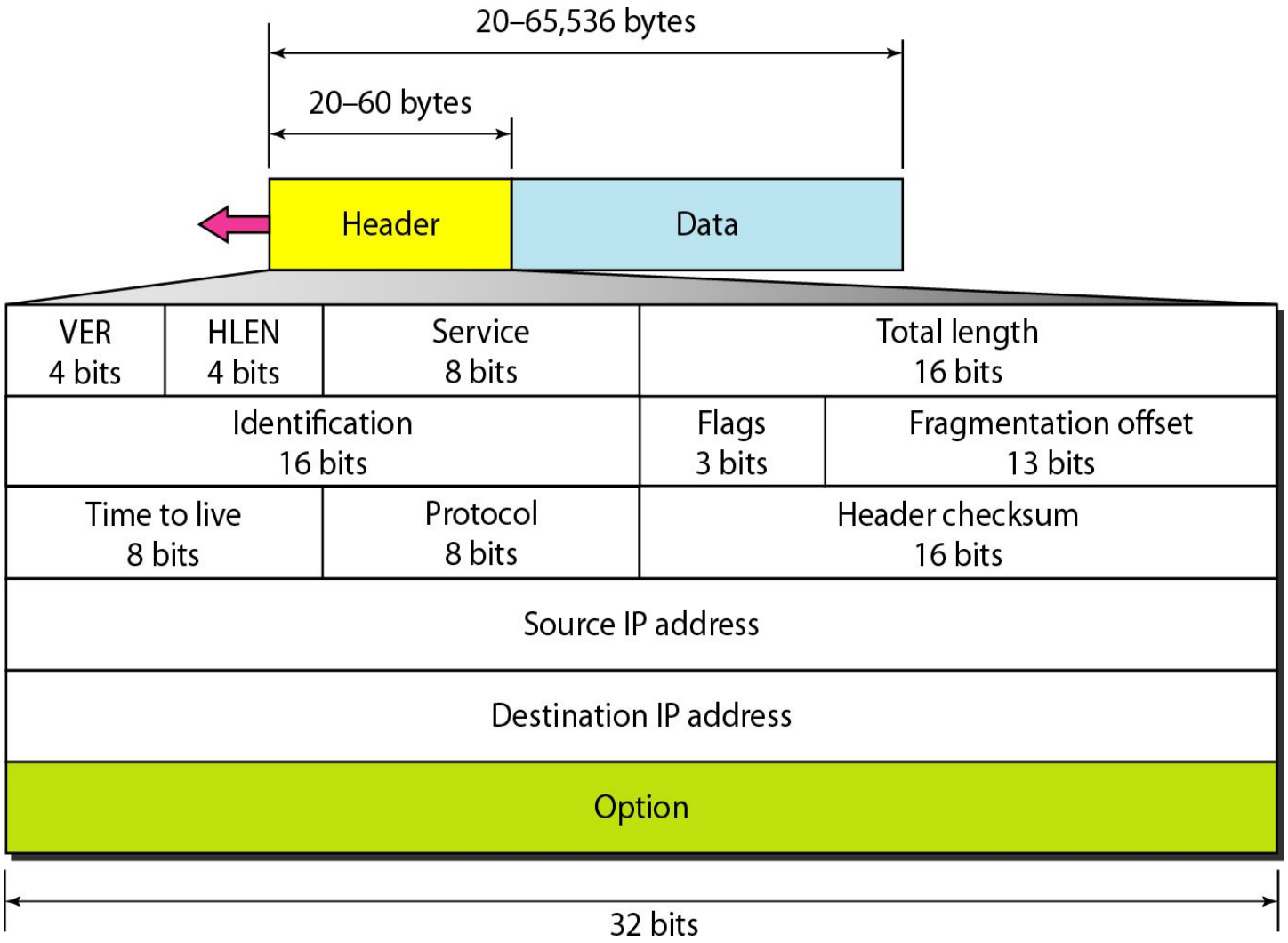
Packets in the IP layer are called **datagrams.**

IPv4

- The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the **TCP/IP protocols**.
- IPv4 is also a **connectionless protocol** for a packet-switching network that uses the datagram approach. This means that each **datagram is handled independently,** and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. IPv4 relies on a higher-level protocol to take care of all these problems.



IPv4 datagram format



Datagram

- Packets in the IPv4 layer are called **datagrams**.
- A datagram is a **variable-length packet** consisting of two parts: **header and data.**
- The header is **20 to 60 bytes in length** and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections.
- **Version (VER)**

This **4-bit field** defines the version of the IPv4 protocol. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4.

- **Header length (HLEN)**

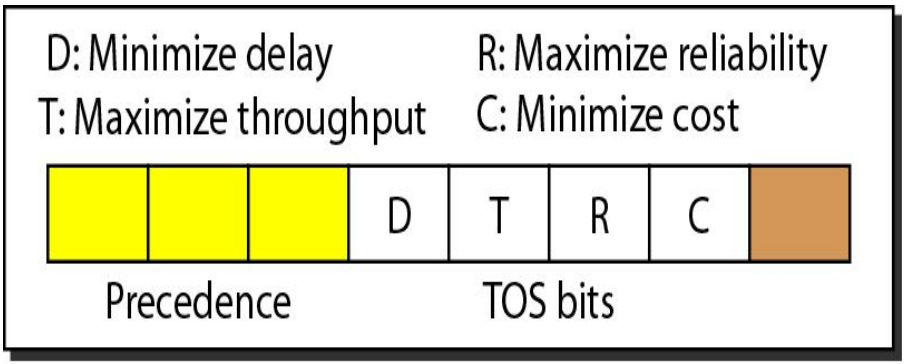
This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are **no options**, the header length is **20 bytes**, and the value of this field is **5** (5 x 4 = 20). When the **option field is at its maximum si**ze, the value of this field is **15** (15 x 4 = 60).

- **Services** IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services.

Service Type

In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

- **Precedence** is a **3-bit subfield** ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the **priority of the datagram** in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first. Some datagrams in the Internet are more important than others.
- **TOS** bits is a **4-bit subfield** with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. With only 1 bit set at a time, we can have five different types of services.



Service type

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Default types of service

Application programs can request a specific type of service. The defaults for some applications:

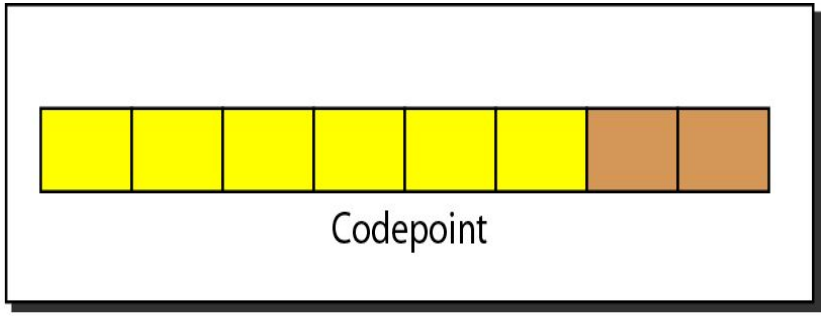
Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Differentiated Services

In this interpretation, the **first 6 bits make up the codepoint** subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.

a. When the **3 rightmost bits are 0s**, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.

b. When **the 3 rightmost bits are not all Os**, the 6 bits define 64 services based on the **priority assignment by the Internet or local authorities.**



Total length

- This is a **16-bit field** that defines the total length (header plus data) of the IPv4 datagram in bytes.
- To find the length of the data coming from the upper layer, subtract the header length from the total length.
- The header length can be found by multiplying the value in the HLEN field by 4.

  Length of data =total length - header length
- Since the field length is 16 bits, the total length of the IPv4 datagram is limited to **65,535 (2^{16} - 1) bytes**.

Identification:

- This field is used in fragmentation. This **16-bit field** identifies a datagram originating from the **source host**. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. All fragments have the same identification number, the same as the original datagram. The identification number helps the destination in reassembling the datagram. It knows that all fragments having the same identification value must be assembled into one datagram.

Flags:

- This field is used in fragmentation.

Fragmentation offset:

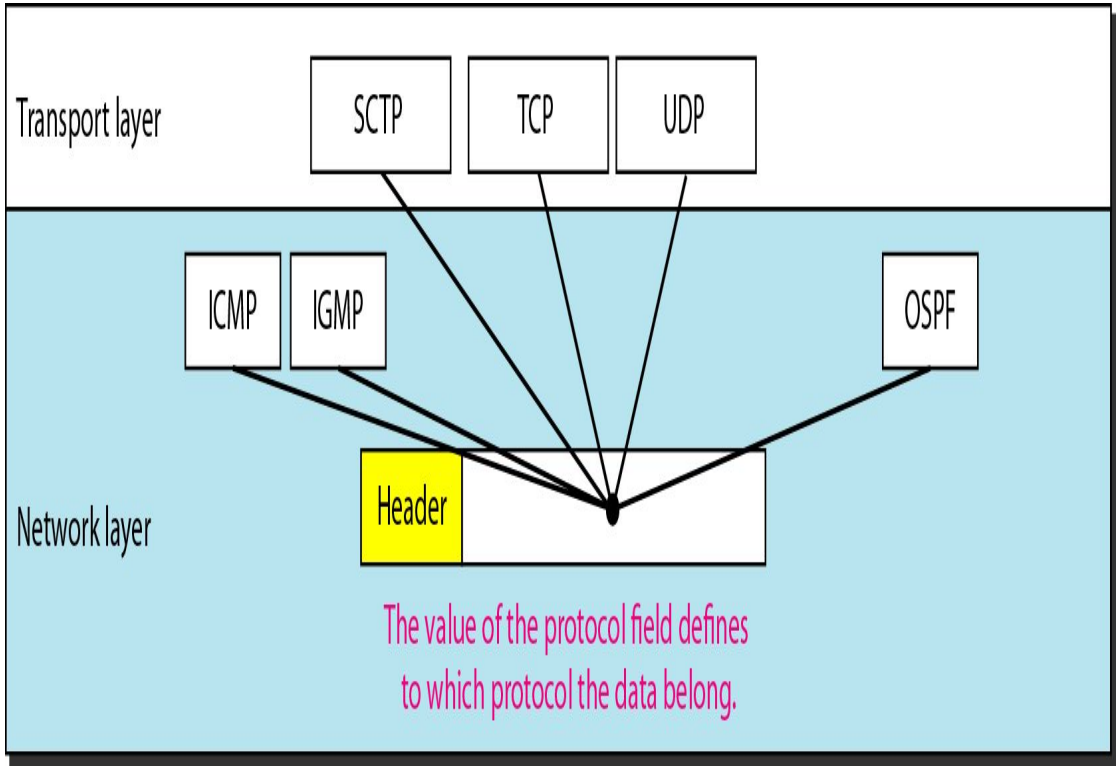
- This field is used in fragmentation.

Time to live:

- To control the **maximum number of hops(routers) visited by the datagram.**
- This field was originally **designed to hold a timestamp**, which was **decremented by each visited router**.
- The datagram was **discarded** when the **value became zero.** this field is used mostly to control the maximum number of hops (routers) visited by the datagram.
- To intentionally limit the journey of the packet

Protocol field and encapsulated data

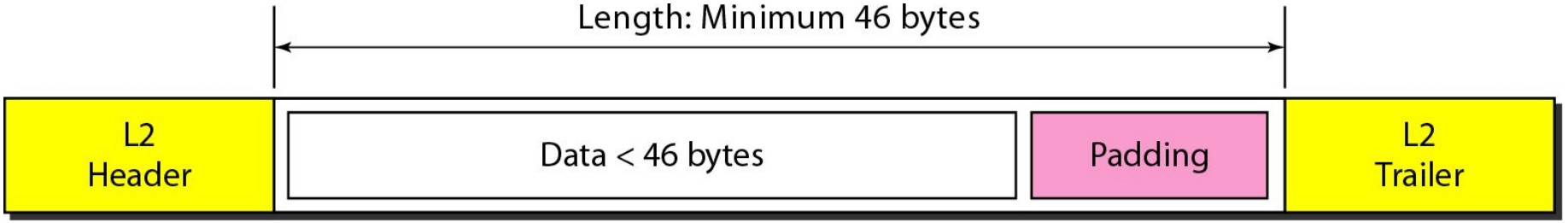
Protocol: This **8-bit field** defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered. In other words, since the IPv4 protocol carries data from different other protocols, the value of this field helps the receiving network layer know to which protocol the data belong.



Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

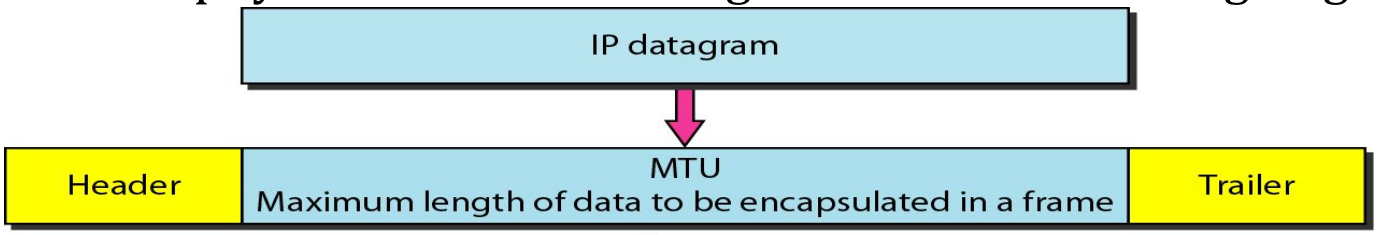
- **Checksum:** The implementation of the checksum in the IPv4 packet follows the same principles. First, the value of the checksum field is set to 0. Then the entire header is divided into 16-bit sections and added together. The result (sum) is complemented and inserted into thechecksum field.
- **Source address:** This **32-bit field** defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Destination address:** This **32-bit field** defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

Figure 20.7 ***Encapsulation of a small datagram in an Ethernet frame***



Fragmentation

- A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, and then encapsulates it in another frame.
- The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
- The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.



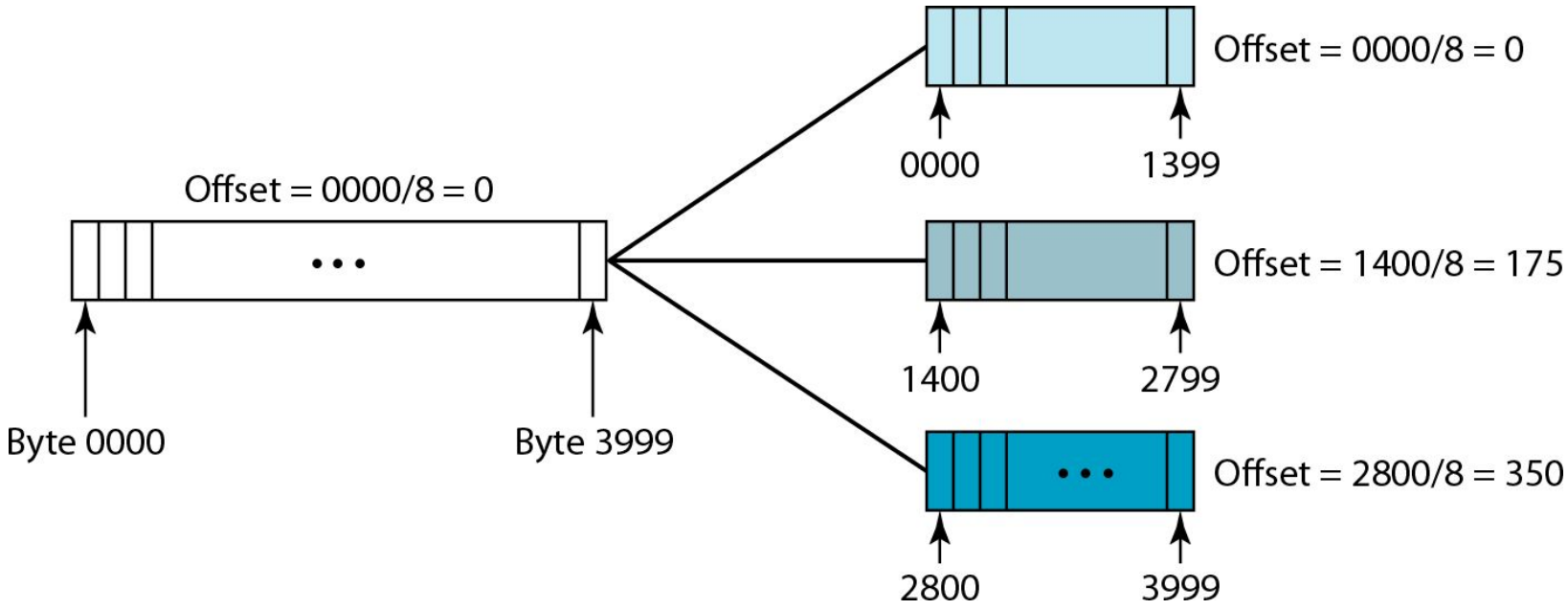
Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

Flags used in fragmentation

- This is a **3-bit field**.
- The **first bit** is **reserved**.
- The **second bit** is called the ***do notfragment*** **bit.** If its value is **1**, the machine **must not fragment the datagram.** If it cannot pass the datagram through any available physical network, it discards the datagram and sends an **ICMP error** message to the source host.
- If its value is **0**, the datagram **can be fragmented** if necessary.
- The **third bit** is called the ***more fragment*** **bit.** If its value is **1**, it means the datagram is **not the last fragment**; there are more fragments after this one. If its value is **0**, it means this is the **last or only fragment.**

- **Fragmentation offset:**
  This **13-bit field** shows the **relative position of this fragment** with respect to the whole datagram.
  It is the offset of the data in the original datagram measured in units of 8 bytes.
  The value of the **offset is measured in units of 8 bytes**. This is done because the length of the offset field is only 13 bits and cannot represent a sequence of bytes greater than 8191. This forces hosts or routers that fragment datagrams to choose a fragment size so that the first byte number is divisible by 8.

A datagram with a data size of 4000 bytes fragmented into three fragments



Options

The header of the IPv4 datagram is made of two parts: **a fixed part and a variable part.**

The **fixed part** is **20 bytes long.** The **variable part** comprises the options that can be a maximum of **40 bytes.**

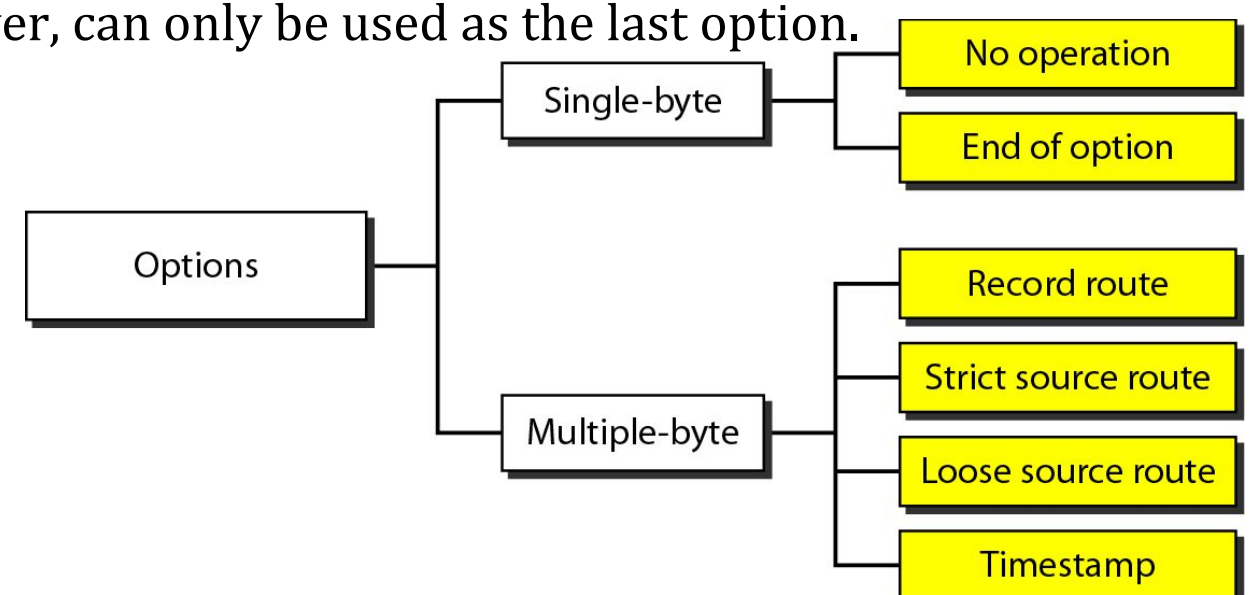
Options, as the name implies, are not required for a datagram. They can be used for **network testing and debugging.** Although options are not a required part of the IPv4 header, option processing is required of the IPv4 software. This means that all implementations must be able to handle options if they are present in the header.

No Operation

A **no-operation option** is a **1-byte option** used as a **filler between options.**

End of Option

An end-of-option option is a **1-byte option used for padding** at the end of the option field. It, however, can only be used as the last option.



- **Record Route**
  A record route option is used to **record the Internet routers** that handle the datagram. It can list up **to nine router addresses**. It can be used for **debugging and management purposes.**

  Strict Source Route
  A strict source route option is **used by the source to predetermine a route** for the **datagram** as it travels through the Internet. Dictation of a route by the source can be useful for several purposes. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose. If a datagram specifies a strict source route, **all the routers defined in the option must be visited by the datagram**. A router must not be visited if its IPv4 address is not listed in the datagram. If the datagram visits a router that is not on the list, the datagram is discarded and an error message is issued. If the datagram arrives at the destination and some of the entries were not visited, it will also be discarded and an error message issued.

  Loose Source Route
  A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.

  Timestamp
  A timestamp option is used to **record the time of datagram processing by a router**. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet.

IPv6

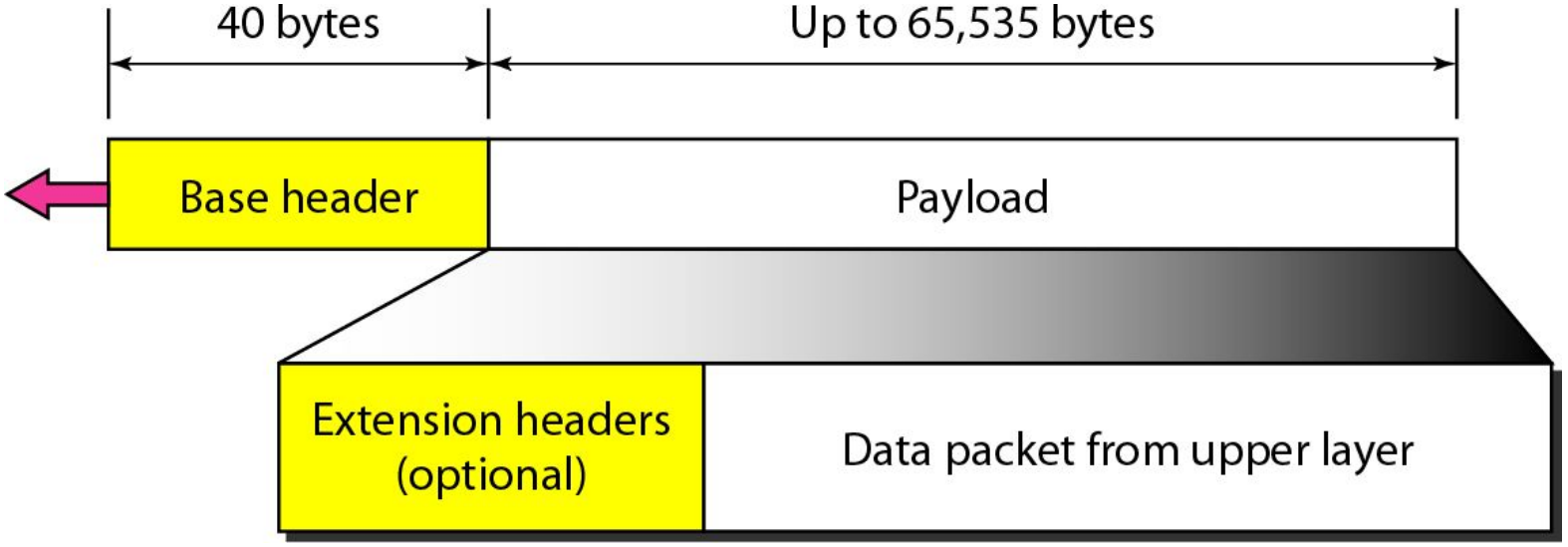
- IPv6 (Internetworking Protocol, version 6), also known as **Ipng** (Internetworking Protocol, next generation), was proposed and is now a standard.

Advantages

- **Larger address space**: An IPv6 address is 128 bits long , compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- **Better header format**: IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options:** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension.** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation:** In IPv6, the type-of-service field has been removed, but a mechanism has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security.** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

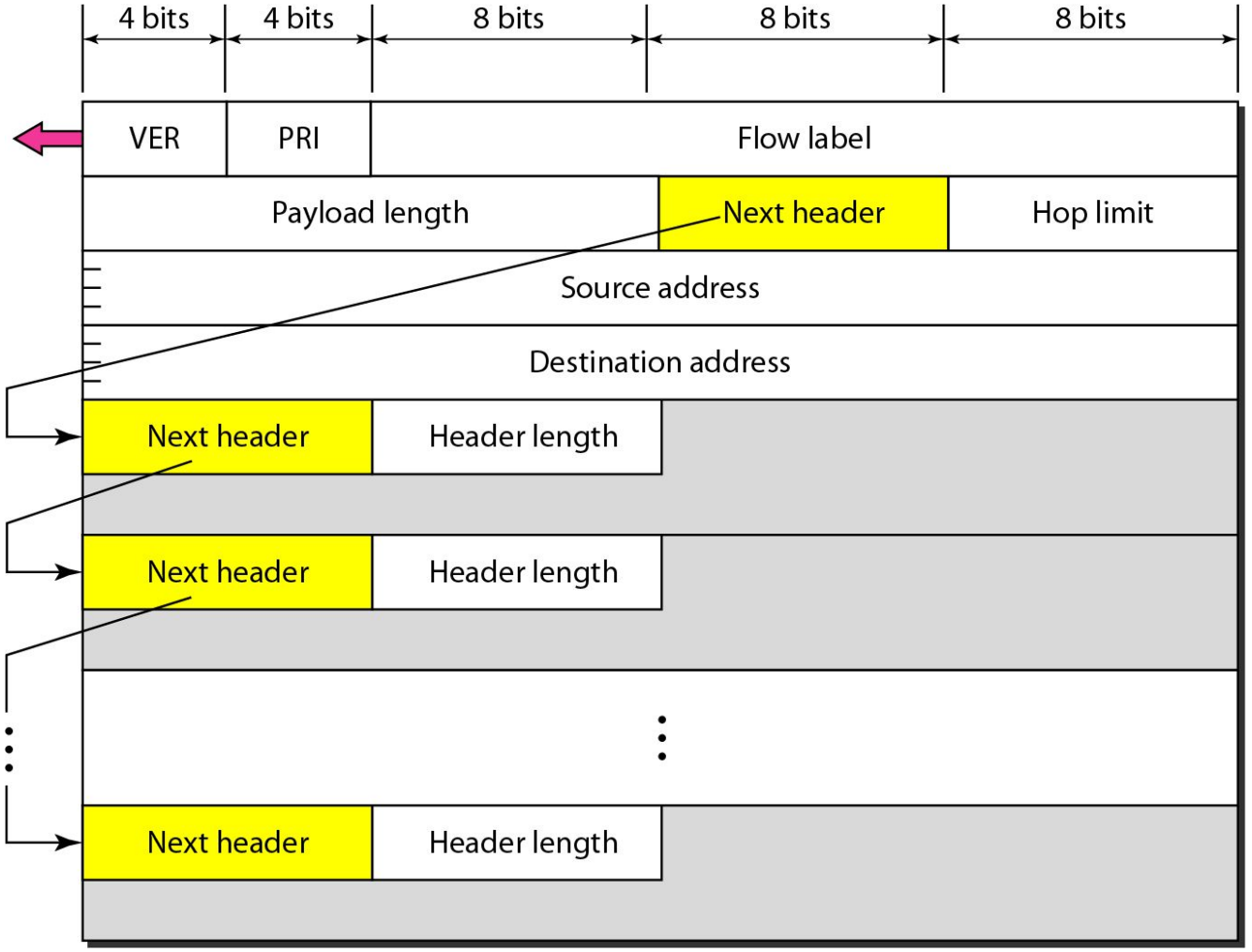
IPv6 datagram header and payload

- Each packet is composed of a mandatory **base header followed by the payload.**
- The payload consists of two parts: **optional extension headers and data from an upper layer.**
- The **base header** occupies **40 bytes**, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



- **Version:** This **4-bit field** defines the version number of the IP. For IPv6, the value is **6**.
- **Priority:** The **4-bit priority** field defines the priority of the packet with respect to **traffic congestion**.
- **Flow label:** The flow label is a **3-byte (24-bit)** field that is designed to provide special handling for a particular flow of data.
- **Payload length:** The **2-byte payload** length field defines the length of the IP datagram excluding the base header.
- **Next header:** The next header is an **8-bit field** defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. Note that this field in version 4 is called the *protocol.*
- **Hop limit:** This **8-bit hop limit** field serves the same purpose as the **TTL field** in IPv4.
- **Source address:** The source address field is a **16-byte** (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address:** The destination address field is a **16-byte** (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

Format of an IPv6 datagram



Next header codes for IPv6

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Priority

- The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source. For example, if one of two consecutive datagrams must be discarded due to congestion, the datagram with the lower **packet priority** will be discarded.
- IPv6 divides traffic into two broad categories: **congestion-controlled and non-congestion-controlled.**
- **Congestion-Controlled Traffic:** If a **source adapts itself to traffic slowdown** when there is congestion, the traffic is referred to as **congestion-controlled traffic. In** congestion-controlled traffic, it is understood that packets may arrive delayed, lost, or out of order. Congestion-controlled data are assigned priorities from 0 to 7. A **priorit** **st.**

Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

The priority descriptions are as follows:

o **No specific traffic.** A priority of 0 is assigned to a packet when the process **does not define a priority.**

o **Background data.** This group (priority 1) defines **data that are usually delivered in the background.** Delivery of the news is a good example.

o **Unattended data traffic.** If the **user is not waiting** (attending) for the data to be received, the packet will be given a priority of 2. E-mail belongs to this group. The recipient of an e-mail does not know when a message has arrived. In addition, an e-mail is usually **stored before it is forwarded.** A little bit of delay is of little consequence.

o **Attended bulk data traffic.** A protocol that transfers data while the **user is waiting (attending) to receive the data** (possibly with delay) is given a priority of 4. FTP and HTTP belong to this group.

o **Interactive traffic.** Protocols such as **TELNET that need user** interaction are assigned the second-highest priority (6) in this group.

o **Control traffic.** Control traffic is given the highest priority (7). Routing protocols such as **OSPF and RIP** and management protocols such as **SNMP** have this priority.

Noncongestion-controlled traffic

- **Noncongestion-controlled traffic**: This refers to a type of traffic that expects **minimum delay.**
- **Discarding of packets** is not desirable**.**
- **Retransmission** in most cases is impossible**.**
- In other words, the **source does not adapt itself** to congestion**.**
- Real-time audio and video are examples of this type of traffic**.**
- Priority numbers from 8 to 15 are assigned to noncongestion-controlled traffic.

Priority	*Meaning*
8	Data with greatest redundancy
. . .	. . .
15	Data with least redundancy

Flow Label

- A **sequence of packets**, sent from a **particular source** to a particular **destination,** that needs **special handling by routers** is called a *flow* of packets.
- The **combination of the source address** and the **value of the *flow label*** uniquely defines a flow of packets.
- To allow the effective use of flow labels, three rules have been defined:

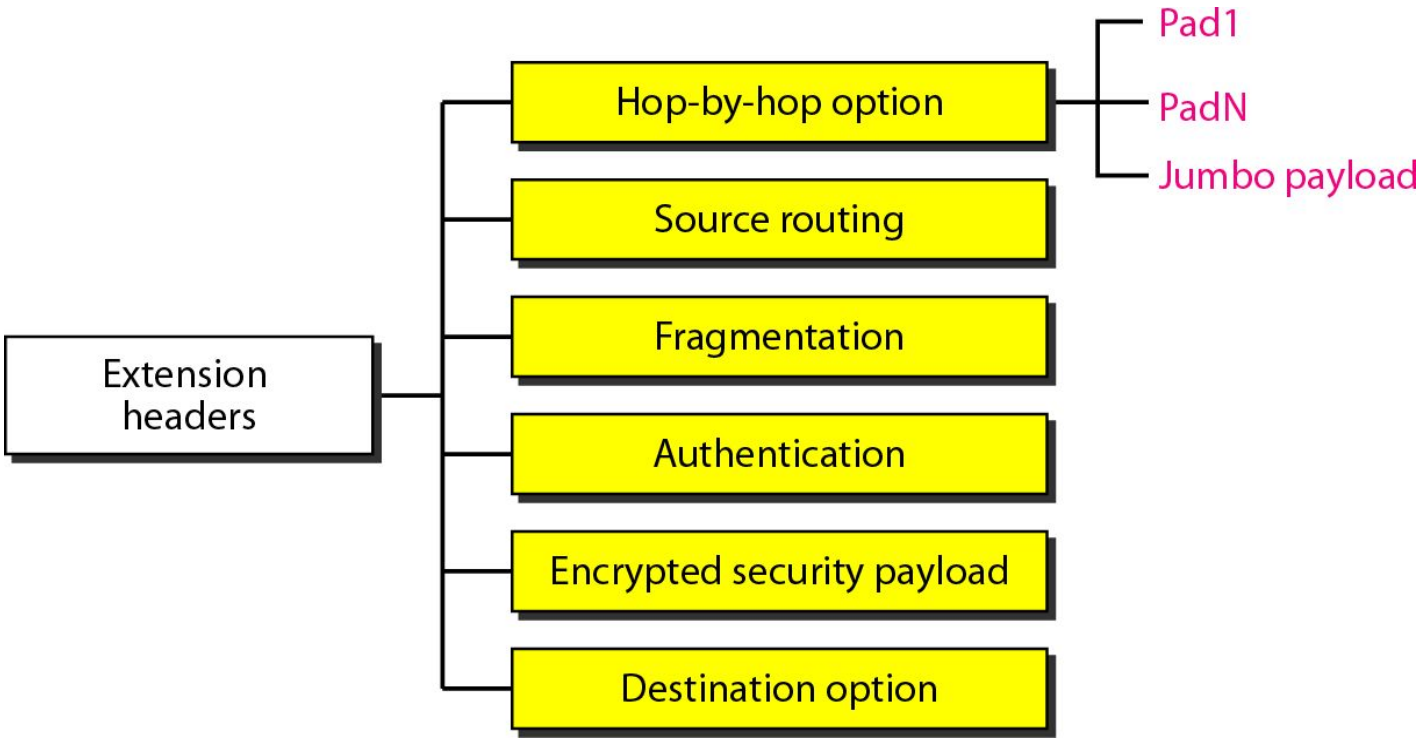
1. The **flow label** is assigned to a **packet** by the **source host.** The label is a random number between **1 and 2^{24} - 1**. A source must not reuse a flow label for a new flow while the existing flow is still active.

2. If a **host does not support the flow label**, it sets this field **to zero.** If a **router does not support the flow label**, it simply **ignores** it.

3. All **packets** belonging to the **same flow** have the **same source**, **same destination, same priority, and same options.**

Extension headers

- The length of the base header is fixed at 40 bytes. However, to give greater functionality to the IP datagram, the base header can be followed by up to **six extension headers**.
- Many of these headers are options in IPv4.



Hop-by-Hop Option

The **hop-by-hop option** is used when the **source** needs to pass information to all **routers visited by the datagram**. So far, only three options have been defined: **Pad1, PadN, and jumbo payload.**

The **Pad1 option** is **1 byte long** and is designed for **alignment purposes.** PadN is similar in concept to Pad1. The difference is that PadN is used when **2 or**

more bytes is needed for alignment. The **jumbo payload option** is used to define a **payload longer than 65,535 bytes**.

Source Routing The source routing extension header combines the concepts of the **strict source route and the loose source route** options of IPv4.

Fragmentation

The concept of fragmentation is the same as that in IPv4. However, the place where fragmentation occurs differs. In IPv4, the source or a router is required to fragment if the size of the datagram is larger than the MTU of the network over which the datagram travels. In IPv6, only **the original source can fragment**. A source must use a path **MTU discovery technique to find the smallest MTU** supported by any network on the path. The source then fragments using this knowledge.

Authentication

The authentication extension header has a dual purpose: it **validates the message sender and ensures the integrity of data.**

Encrypted Security Payload

The **encrypted security payload** (ESP) is an extension that provides **confidentiality** and **guards against eavesdropping.**

Destination Option The destination option is used when **the source needs to pass information to the destination** only.

Intermediate routers are not permitted access to this information.

Comparison between IPv4 and IPv6 packet headers

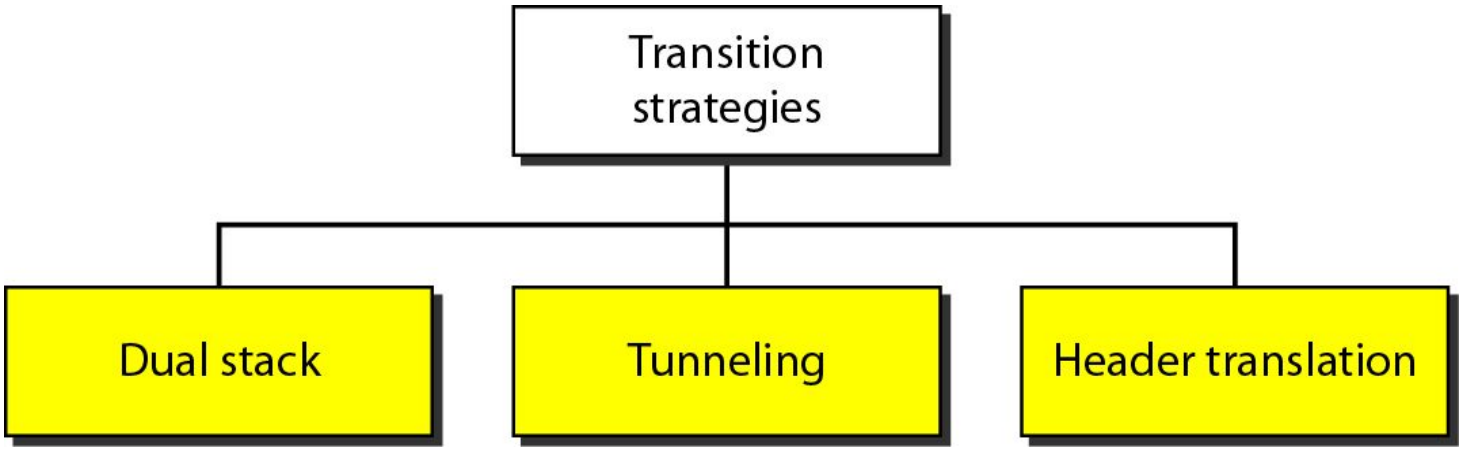
Comparison
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

Comparison between IPv4 options and IPv6 extension headers

Comparison
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

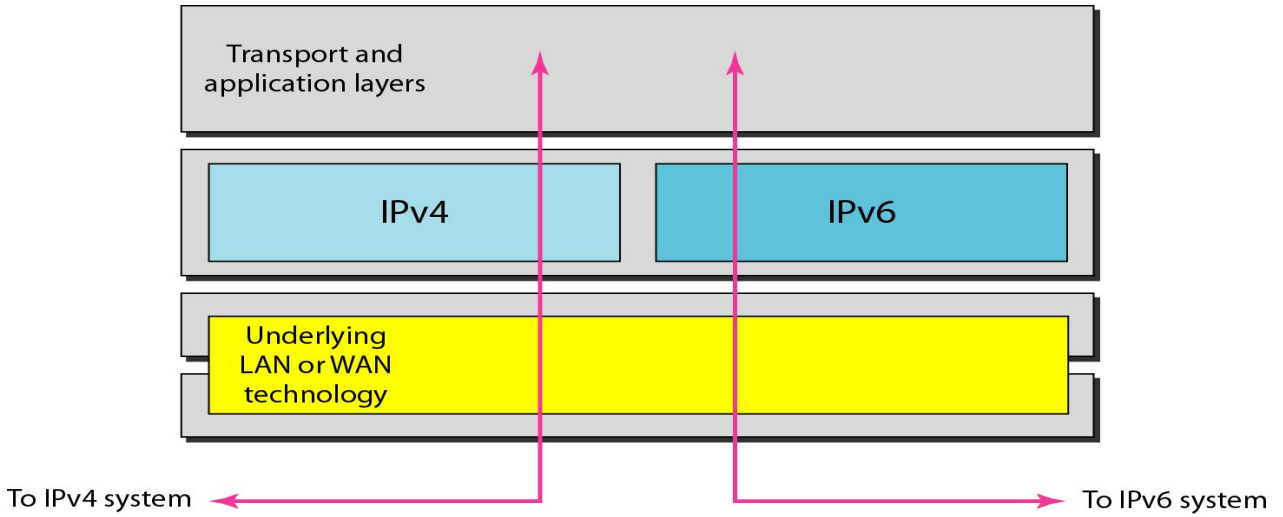
Transition from IPv4 to IPv6

- Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.
- It takes a considerable amount of time before every system in the Internet can move from IPv4 to IPv6.
- The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.
- Three strategies have been devised by the IETF to help the transition:



Dual stack

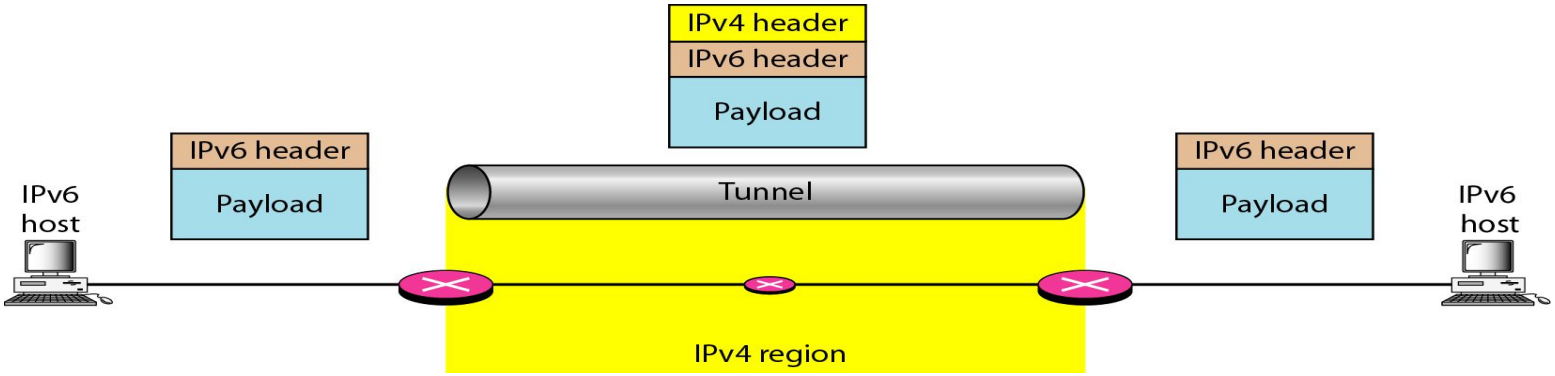
- It is recommended that all hosts, before migrating completely to version 6, have a **dual stack of protocols**.
- In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.
- To determine which version to use when sending a packet to a destination, the **source host queries the DNS**. If the **DNS returns an IPv4 address**, the **source host sends an IPv4 packet**. If the **DNS returns an IPv6 address**, the **source host sends an IPv6 packet**.



Tunneling strategy

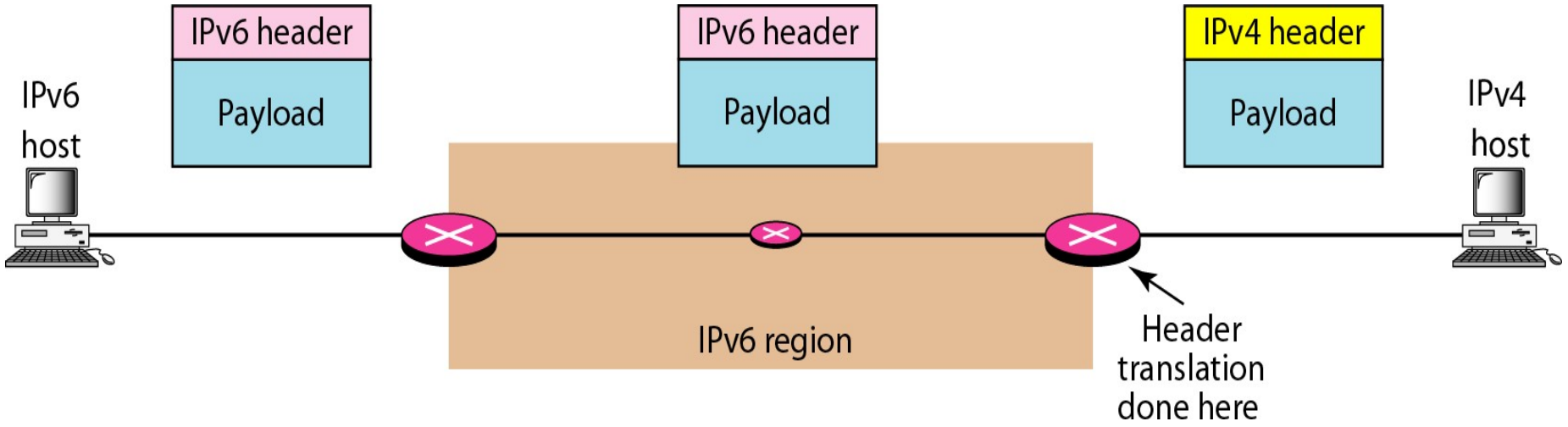
Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4.

To make it clear that the IPv4 packet is carrying an IPv6 packet as data, the **protocol value is set to 41**.



Header translation strategy

- Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4.
- The header of the IPv6 packet is converted to an IPv4 header. Header translation uses the **mapped address** to translate an IPv6 address to an IPv4 address.



Header translation

Header Translation Procedure
1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.

Address Mapping

- The delivery of a packet to a host or a router requires two levels of addressing: logical and physical.
- An internet is made of a combination of physical networks connected by internetworking devices such as routers.
- The hosts and routers are recognized at the network level by their logical (IP) addresses.
- Packets pass through physical networks to reach these hosts and routers.
- At the physical level, the hosts and routers are recognized by their physical (MAC) addresses.
- A **physical address is a local address**. Its jurisdiction is a local network. It must be unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware. An example of a **physical address is the 48-bit MAC address in** the Ethernet protocol, which is imprinted on the NIC installed in the host or router.
- 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. These can be done by using either **static or dynamic mapping**.

Static Mapping

- involves in the **creation of a table** that associates a **logical address with a physical address.**
- This table is stored in each machine on the network.
- Each machine that knows, the IP address of another machine but not its physical address can look it up in the table.
- This has some limitations because physical addresses may change in the following ways:

1. A machine could change its NIC, resulting in a new physical address.

2. In some LANs, such as LocalTalk, the physical address changes every time the computer is turned on.

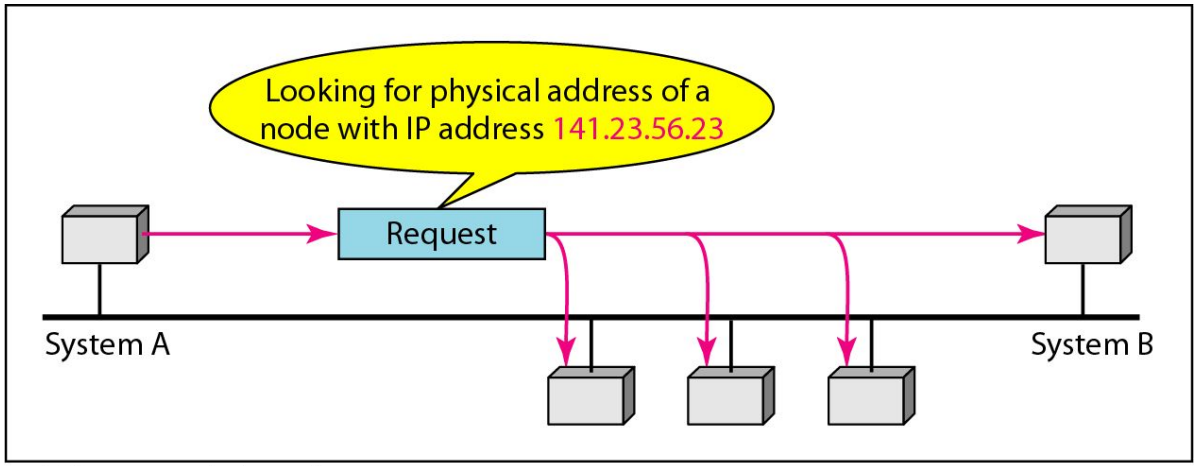
3. A mobile computer can move from one physical network to another,resulting in a change in its physical address.

- To implement these changes, a static mapping table must be **updated periodically.** This **overhead could affect network performance**.

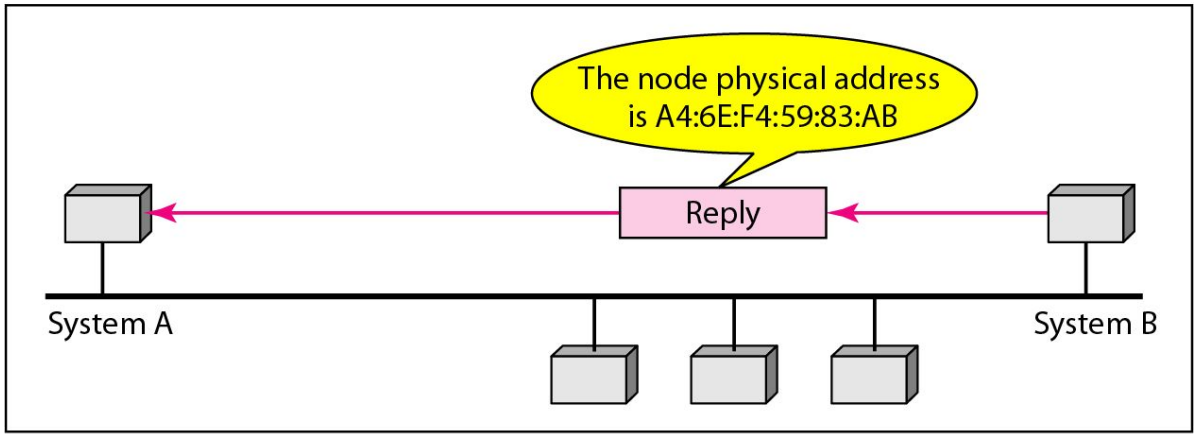
Dynamic Mapping

- In **dynamic mapping,** each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one.
- Two protocols are designed to perform dynamic mapping:
  (i) **ARP –** maps an IP address to a MAC address
  (ii) **RARP –** maps a MAC address to an IP address
- Anytime a host or a router needs to **find the MAC address** of another host or router in the network, it sends an **ARP query packet.**
- The packet includes the physical and **IP addresses of the sender and the IP address of the receiver.** Because the sender does not know the physical address of the receiver, the **query is broadcast** over the network.
- Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an **ARP response packet**.
- The **response packet** contains the **recipient's IP and physical addresses.** The packet is **unicast directly** to the inquirer by using the physical address received in the query packet.

ARP operation

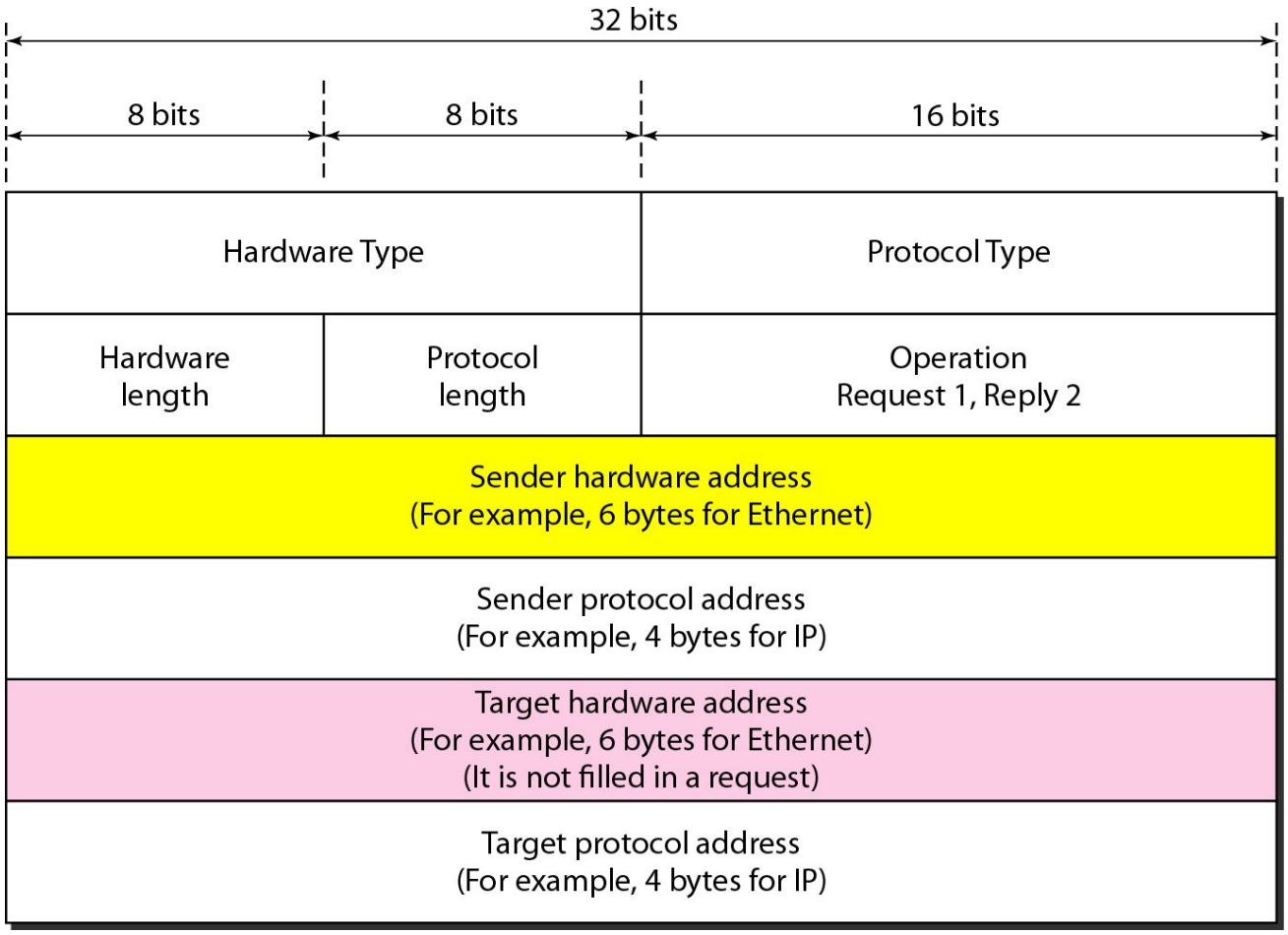


a. ARP request is broadcast



b. ARP reply is unicast

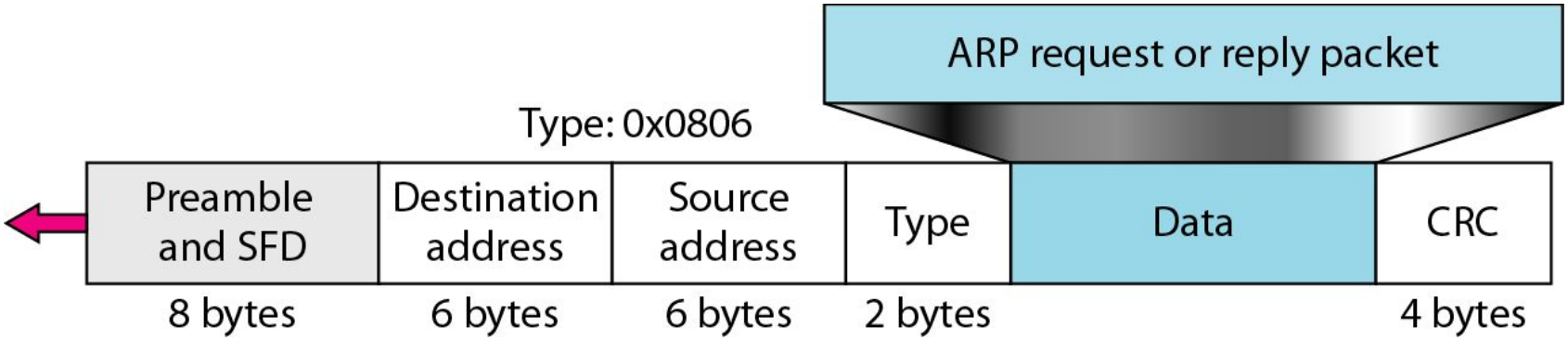
ARP packet


32 bits
8 bits
8 bits
16 bits
Hardware Type
Protocol Type
Hardware length
Protocol length
Operation
Request 1, Reply 2
Sender hardware address
(For example, 6 bytes for Ethernet)
Sender protocol address
(For example, 4 bytes for IP)
Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)
Target protocol address
(For example, 4 bytes for IP)

- **Hardware type (HTYPE):** This is a **16-bit field** defining the type of the network on which ARP is running.
- **Protocol type (PTYPE):** This is a **16-bit field** defining the protocol. **Hardware length (HLEN):** This is an **8-bit field** defining the length of the physical address in bytes.
- **Protocol length (PLEN):** This is an **8-bit field** defining the length of the logical address in bytes.
- **Operation (OPER):** This is a **16-bit field** defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).
- **Sender hardware address (SHA):** This is a **variable-length fiel**d defining the physical address of the sender.
- **Sender protocol address (SPA):** This is a **variable-length field** defining the logical (for example, IP) address of the sender.
- **Target hardware address (THA):** This is a **variable-length field** defining the physical address of the target. For an ARP request message, this field is alI Os because the sender does not know the physical address of the target.
- **Target protocol address (TPA):** This is a **variable-length field** defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

Encapsulation

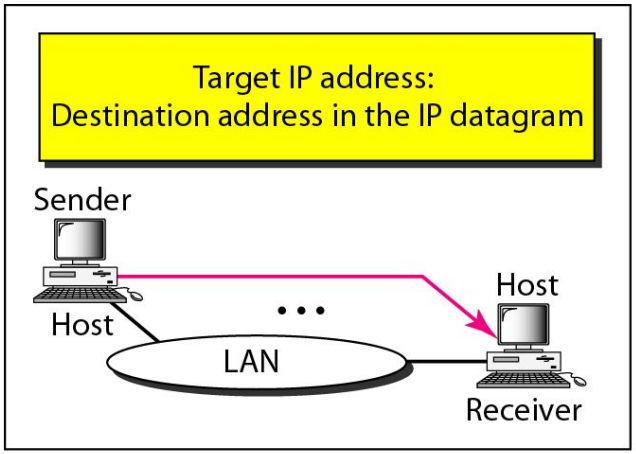
- An ARP packet is encapsulated directly into a data link frame.
- The type field indicates that the data carried by the frame are an ARP packet.



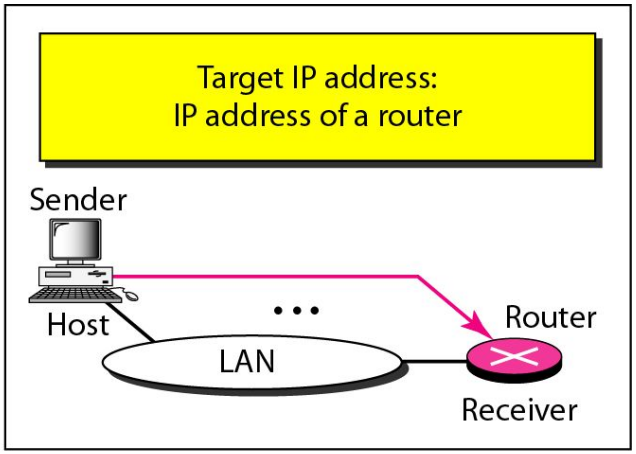
Steps

1. The sender knows the IP address of the target.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

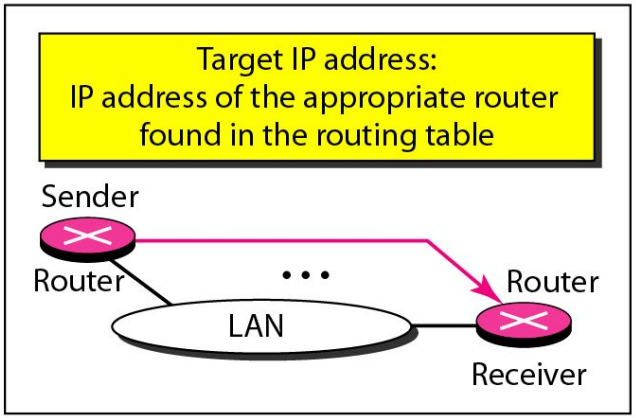
Figure 21.4 *Four cases using ARP*



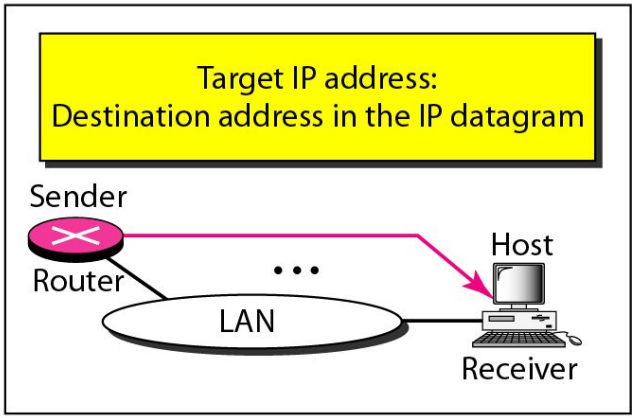
Case 1. A host has a packet to send to another host on the same network.



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



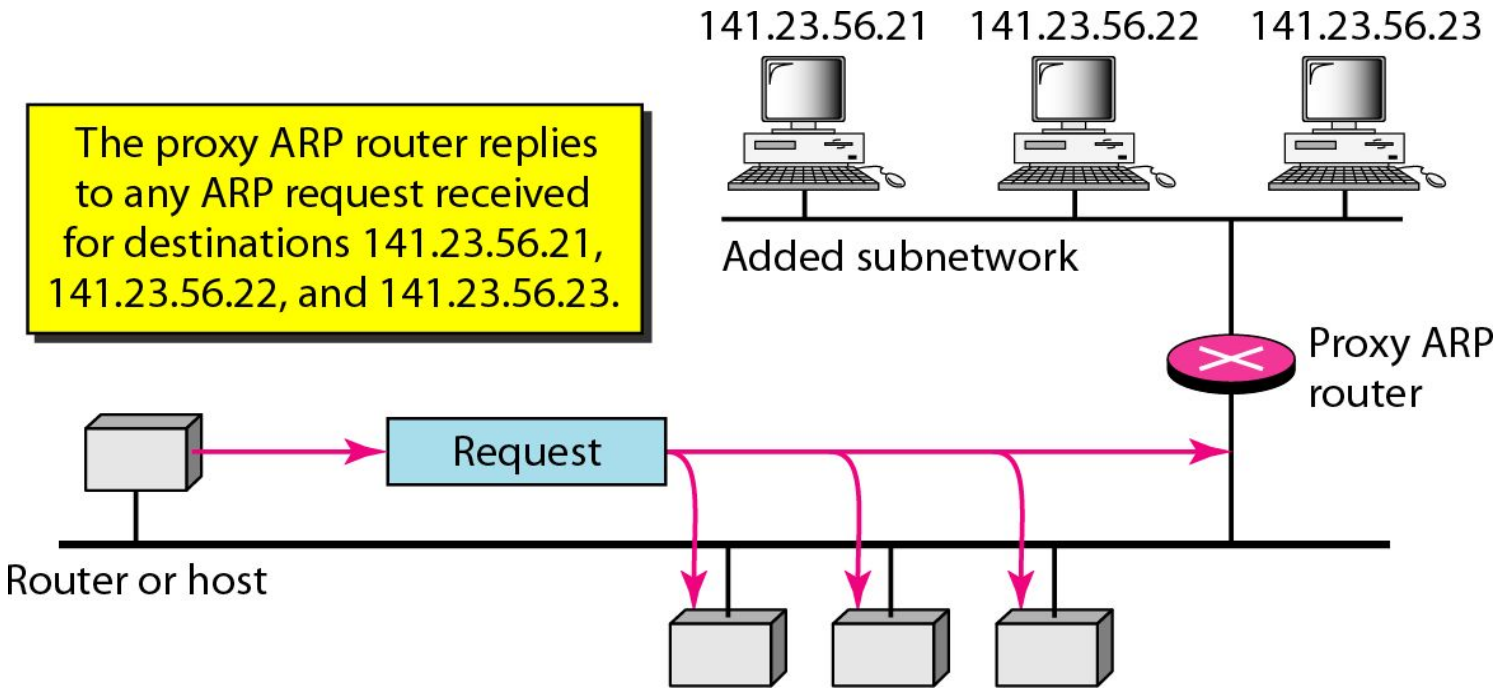
Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.



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

Proxy ARP

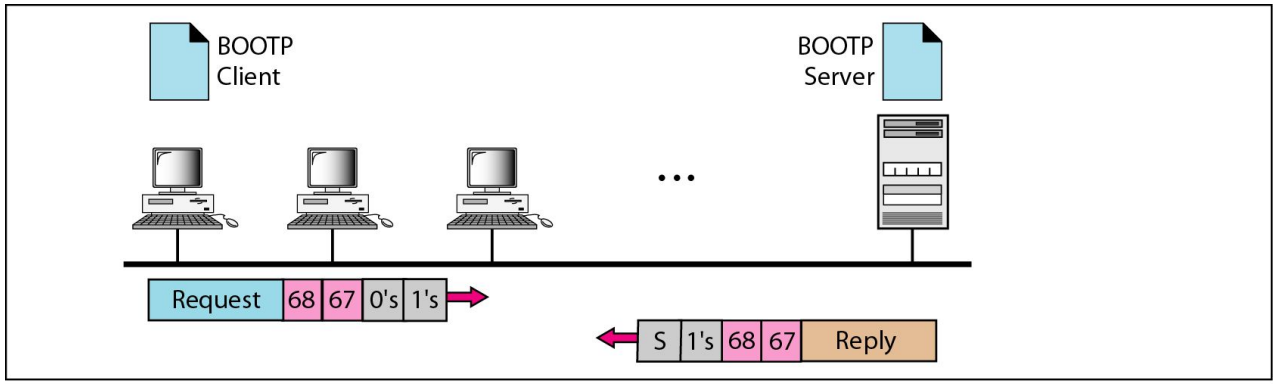
- A technique called ***proxy* ARP** is used to create a subnetting effect.
- A proxy ARP is an ARP that acts on behalf of a set of hosts.
- Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address. After the router receives the actual IP packet, it sends the packet to the appropriate host or router.



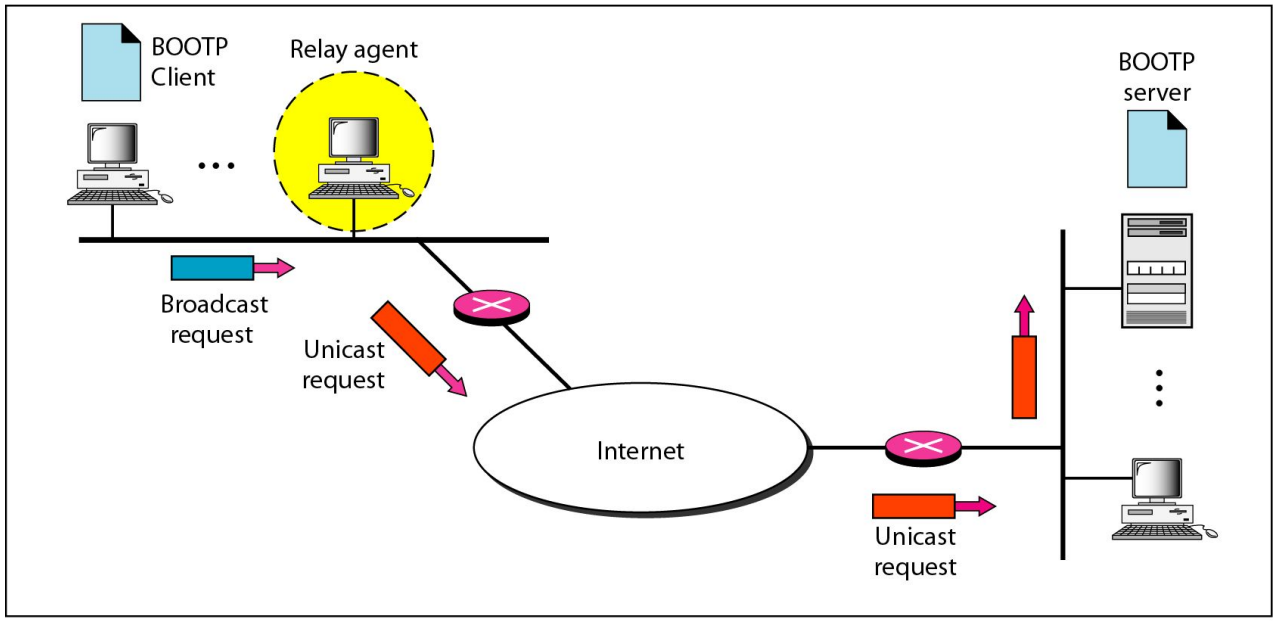
RARP

- Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address.
- Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine.
- To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.
- The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network.
- Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.
- There is a serious problem with RARP: Broadcasting is done at the data link layer. This is the reason that RARP is almost obsolete.
- Two protocols, BOOTP and DHCP, are replacing RARP.

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.

ICMP

- IP has two deficiencies: **lack of error control and lack of assistance mechanisms.**
- The IP protocol has no error-reporting or error-correcting mechanism. What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value? What happens if the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit? These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.
- 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.**

Types of messages

Error-reporting messages

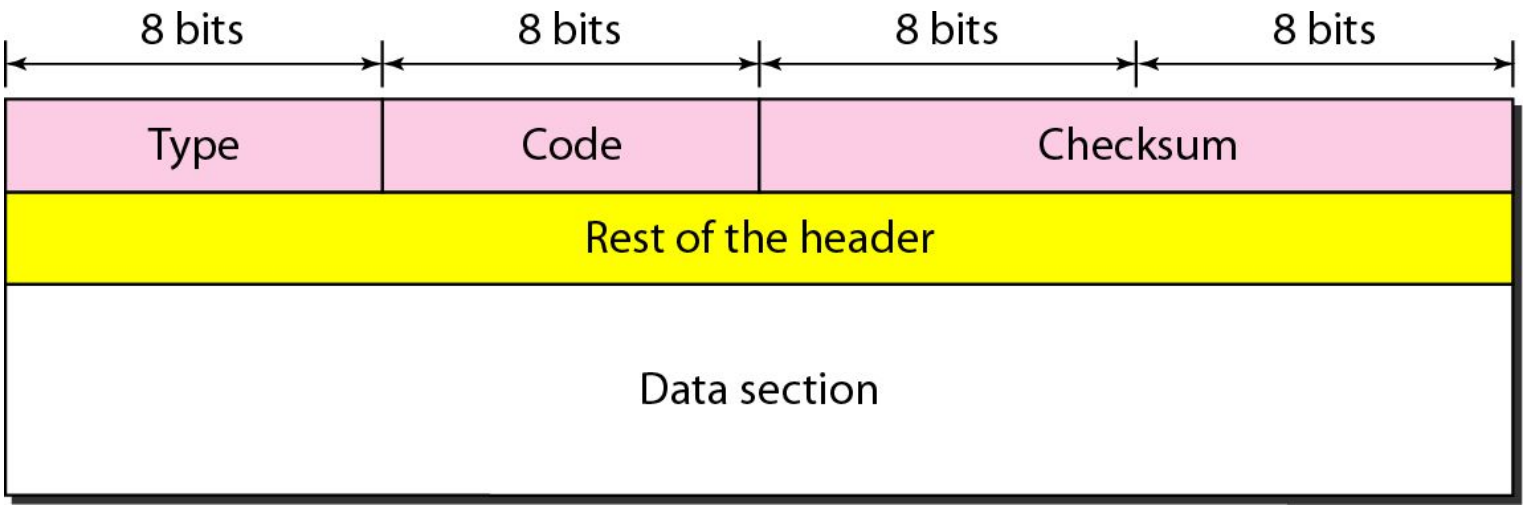
- report problems that a router or a host may encounter when it processes an IP packet.

Query messages

- occur in pairs, help a host or a network manager get specific information from a router or another host
- Hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

Message Format

- An ICMP message has an **8-byte header** and a variable-size data section.



Error reporting

- ICMP does not correct errors-it simply reports them.
- Error correction is left to the higher-level protocols.
- Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses. **ICMP uses the source IP address to send the error message to the source (originator) of the datagram**.
- **Five types of errors are handled**: destination unreachable, source quench, time exceeded, parameter problems, and redirection

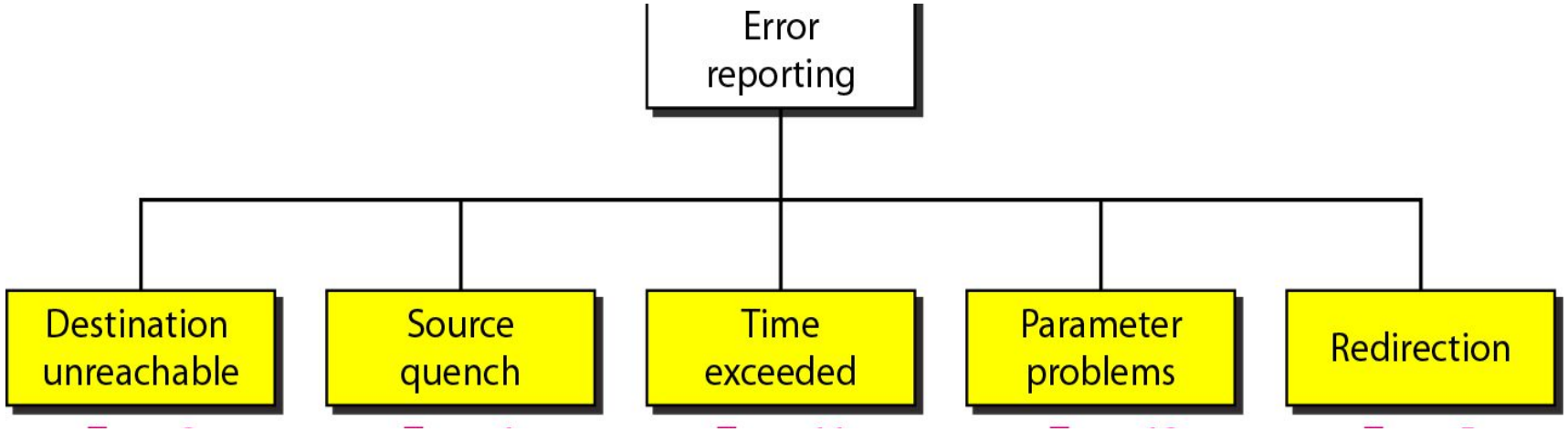
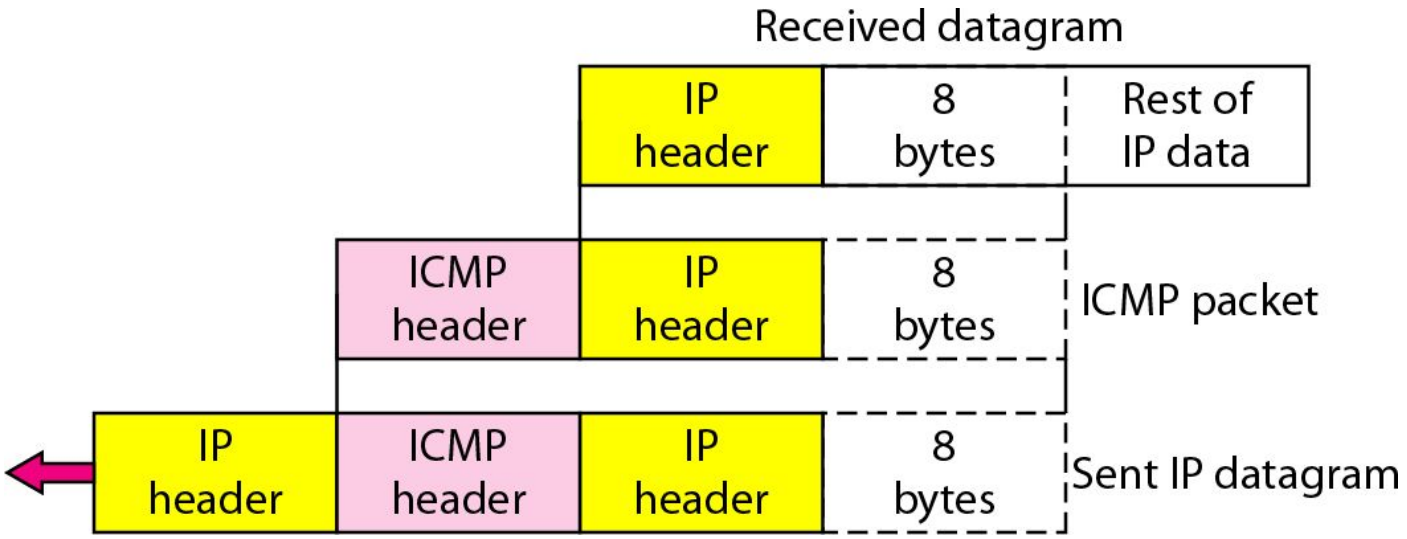


Figure 21.10 *Contents of data field for the error messages*



Types of errors

Destination unreachable

- A **router cannot route a datagram** or **a host cannot deliver a datagram**
- the **datagram is discarded** and the router or the host sends a **destination-unreachable message** back to the source host that initiated the datagram.
- **destination-unreachable messages** can be created by either a **router** or the **destination host.**

Source Quench

- The source-quench message in ICMP was designed to add a kind of **flow control to the IP.**
- When a **router or host discards a datagram** due to **congestion**, it sends a **source-quench message** to the sender of the datagram**.**
- Two purposes:
- (i) it informs the **source that the datagram has been discarded.**
- (ii) it warns the source that there is **congestion** somewhere in the path and that the source should slow down (quench) the sending process**.**

Time Exceeded

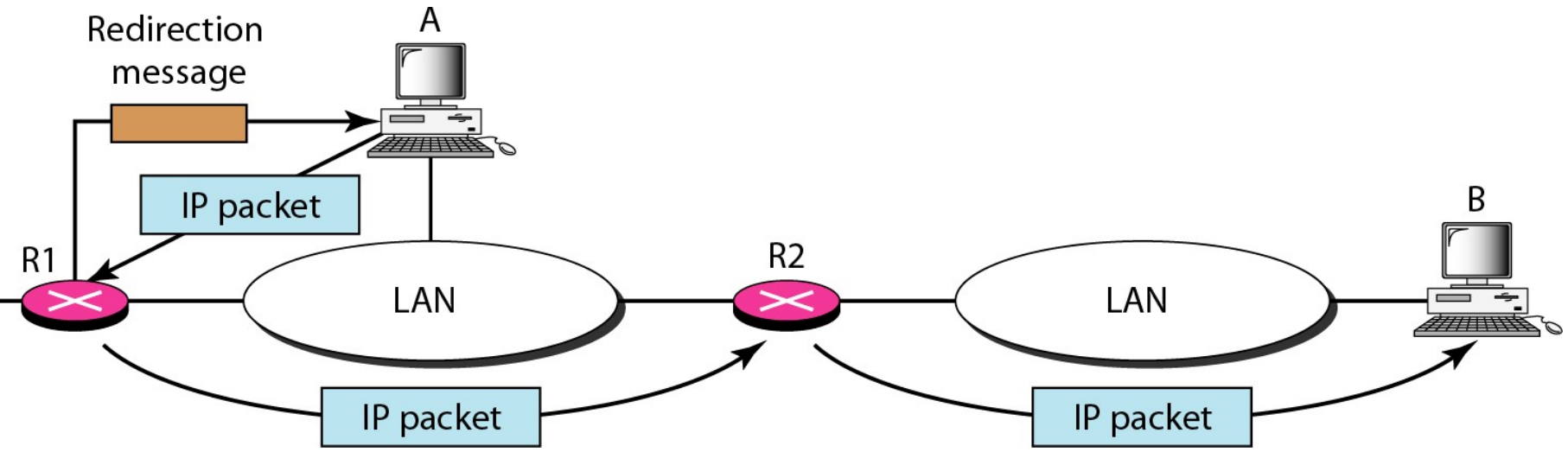
- Routers use routing tables to find the next hop (next router) that must receive the packet.
- If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly.
- When a datagram visits a router, the value of this field is decremented by 1.
- When the **time-to-live value reaches 0**, after decrementing, the router **discards the datagram**.
- When the datagram is discarded, **a time-exceeded message** must be sent by the **router to the original source**.
- A time-exceeded message is also generated **when not all fragments** that make up a message arrive at the destination host within a certain time limit.

Parameter Problem

- If a **router or the destination host discovers an ambiguous or missing value in any field of the datagram,** it discards the datagram and sends a **parameter-problem message** back to the source.

Redirection

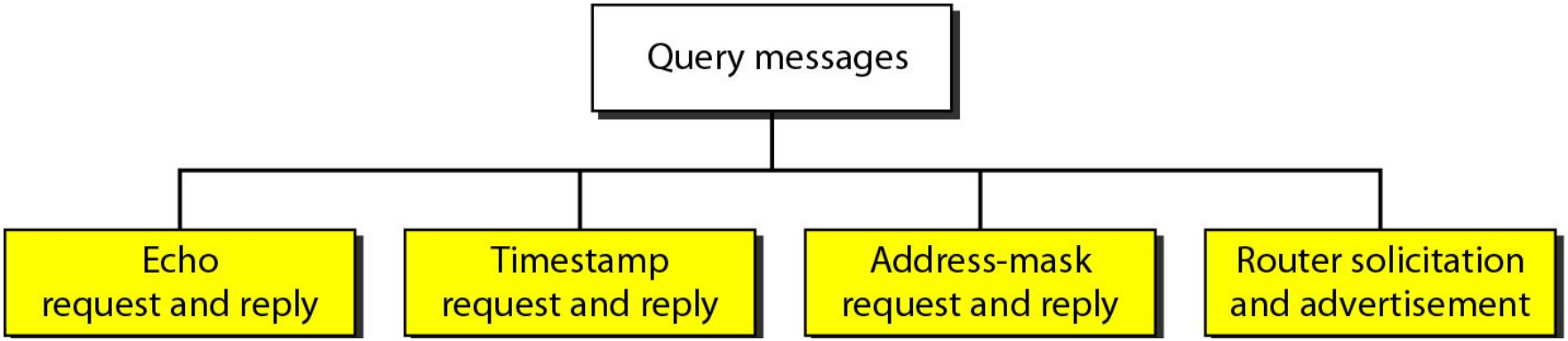
- Routers take part in the routing update process, update constantly. Routing is dynamic.
- Hosts use static routing.
- When a host comes up, its routing table has a limited number of entries. It usually knows the IP address of only one router, the default router.
- For this reason, the host may send a datagram, which is destined for another network, to the wrong router. In this case, the router that receives the datagram will forward the datagram to the correct router.
- However, to **update the routing table of the host**, it sends a redirection message to the host.



Query

In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.

A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame.



Echo Request and Reply

- The echo-request and echo-reply messages are designed for **diagnostic purposes.**
- Network managers and users utilize this pair of messages to **identify network problems.**
- The combination of echo-request and echo-reply messages determines whether two systems can communicate with each other.
- The echo-request and echo-reply messages can be used to determine if there is communication at the IP level.
- Because ICMP messages are encapsulated in IP datagrams, the receipt of an echo-reply message by the machine that sent the echo request is proof that the IP protocols in the sender and receiver are communicating with each other using the IP datagram.
- Also, it is proof that the intermediate routers are receiving, processing, and forwarding IP datagrams.
- Today, most systems provide a version of the ping command that can create a series (instead of just one) of echo-request and echo-reply messages, providing statistical information.

Timestamp Request and Reply

- Two machines (hosts or routers) can use the **timestamp request and timestamp reply messages** to determine the **round-trip time** needed for an IP datagram to travel between them.
- It can also be used to **synchronize the clocks in two machines.**

Address-Mask Request and Reply

- A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24.
- To obtain its **mask,** a **host** sends an **address-mask-request message** to a **router** on the LAN.
- If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message. The router receiving the address-mask-request message responds with an address-mask-reply message, providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

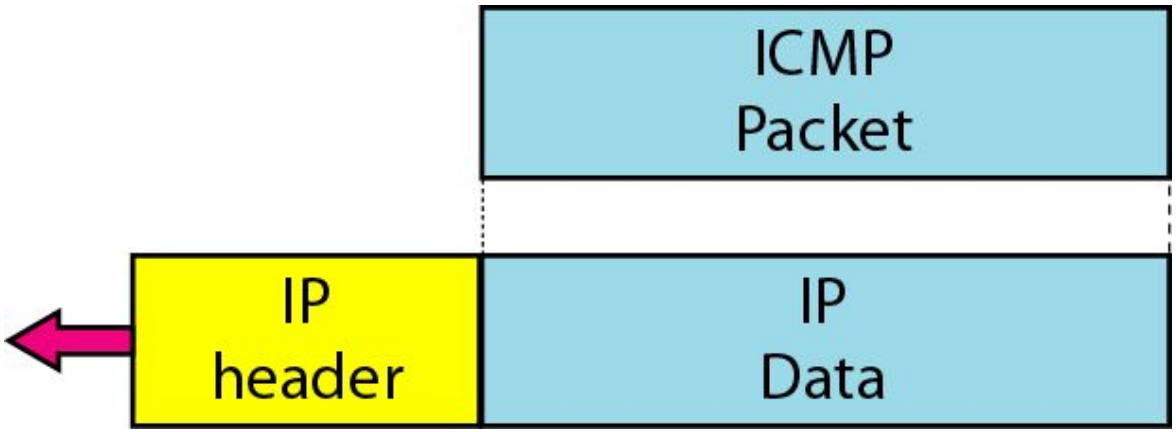
The **host** must know if the **routers are alive and functioning**. **The router-solicitation and router-advertisement messages** can help in this situation. A host can broadcast (or multicast) a router-solicitation message. The **router or routers** that receive **the solicitation message** broadcast their routing information using the **router-advertisement message**. A router can also periodically send router-advertisement messages even if no host has solicited.

Note that when a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

Checksum

In ICMP the checksum is calculated over the entire message (header and data).

Figure 21.13 *Encapsulation of ICMP query messages*



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.

Figure 21.16 *IGMP message types*

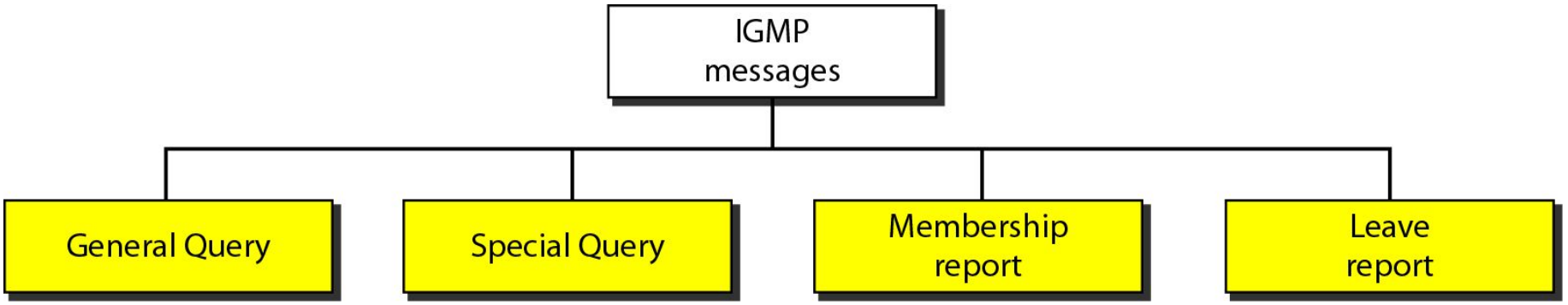


Figure 21.17 *IGMP message format*

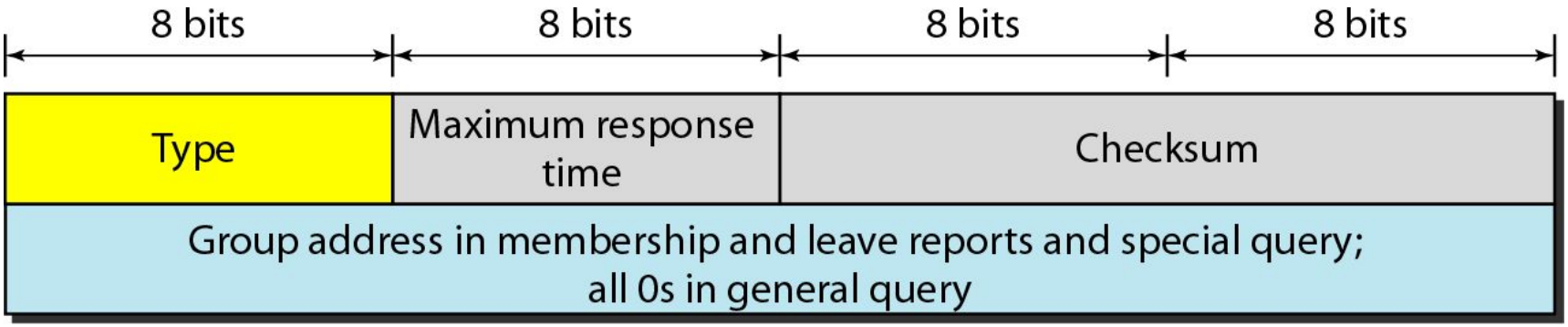


Table 21.1 *IGMP type field*

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

Figure 21.18 *IGMP operation*

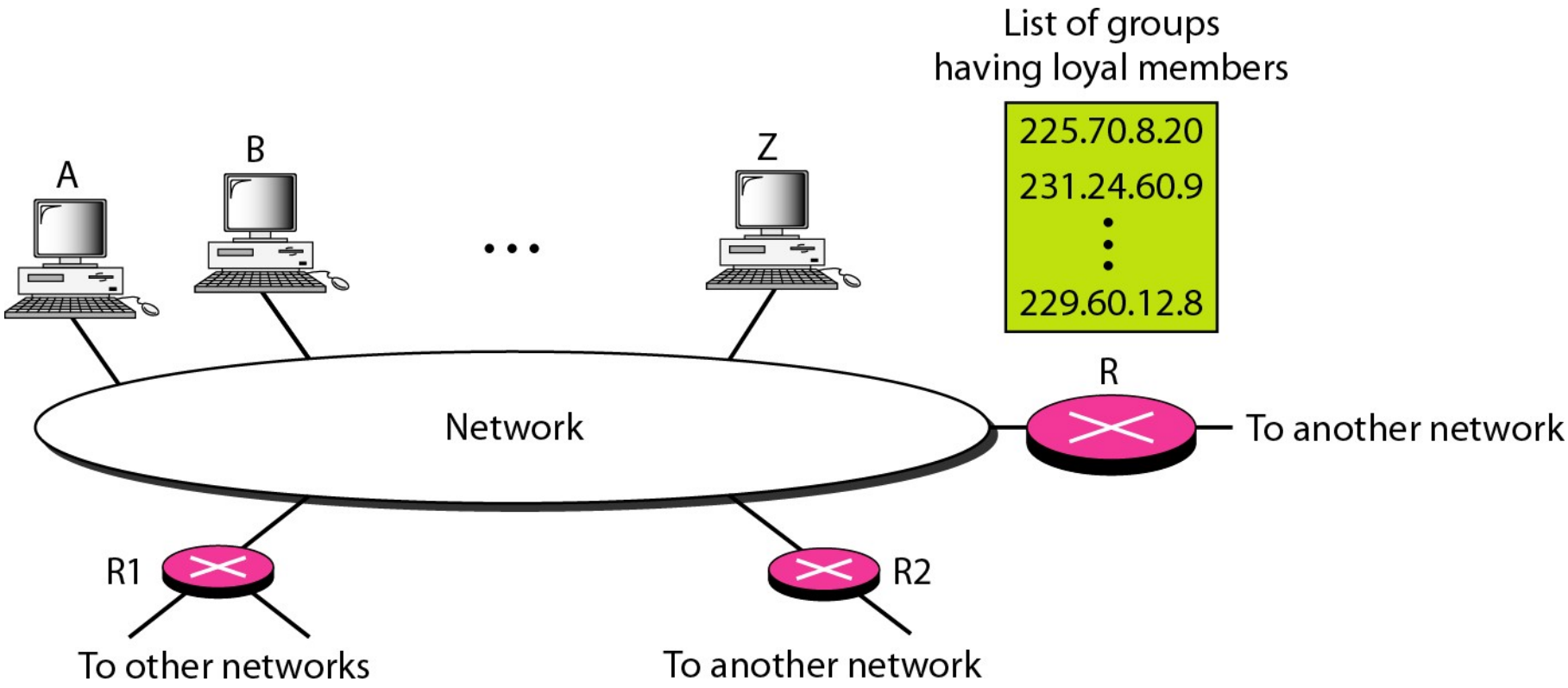
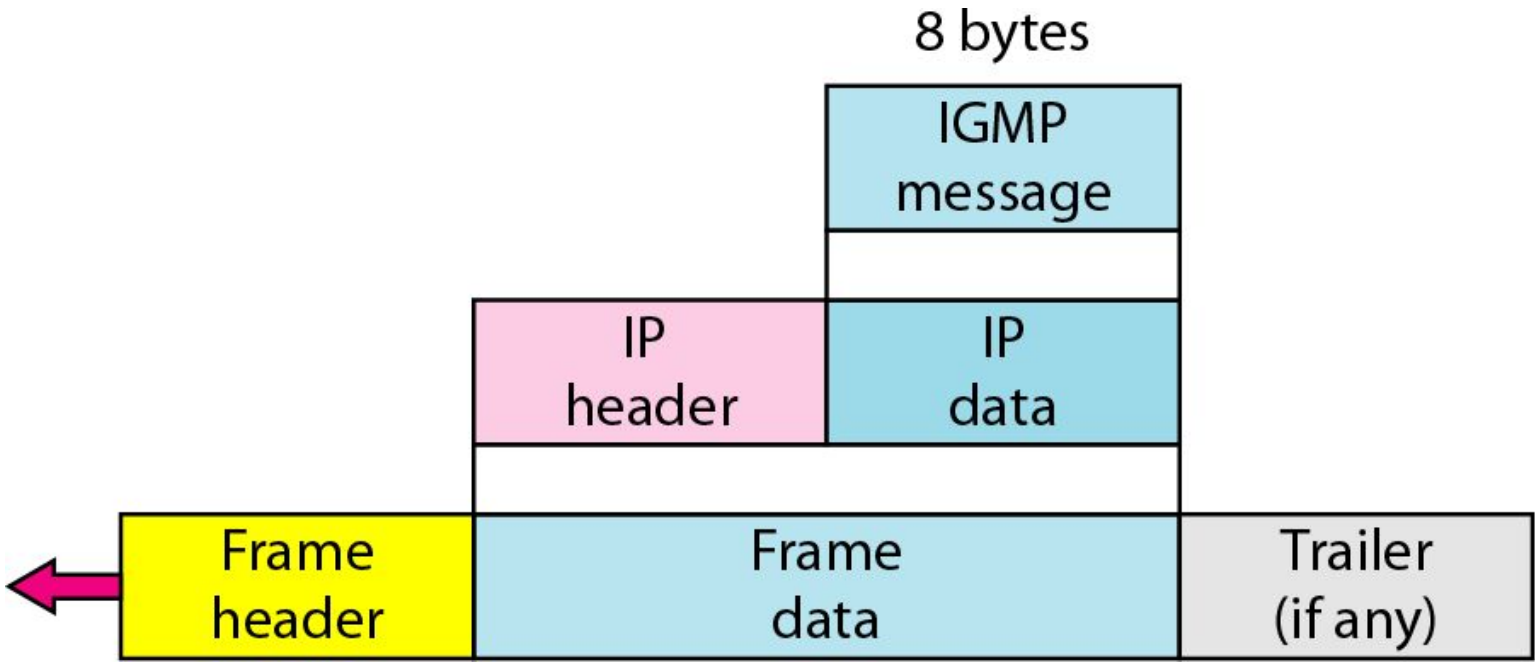


Figure 21.20 *Encapsulation of IGMP packet*



Note

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

Table 21.2 ***Destination IP addresses***

Type	*IP Destination Address*
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

Note

An Ethernet multicast physical address is in the range 01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF.

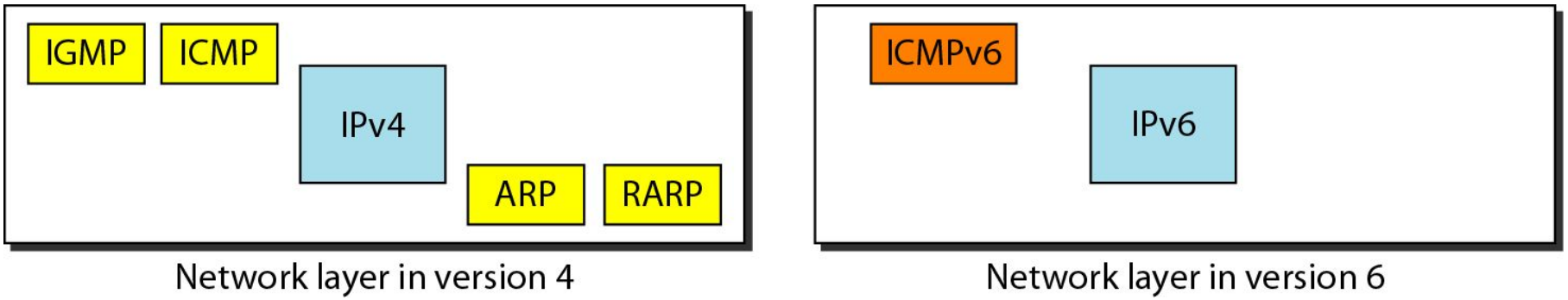
ICMPv6

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*



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

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

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.

Intradomain & Interdomain routing

An internet is divided into **autonomous systems.**

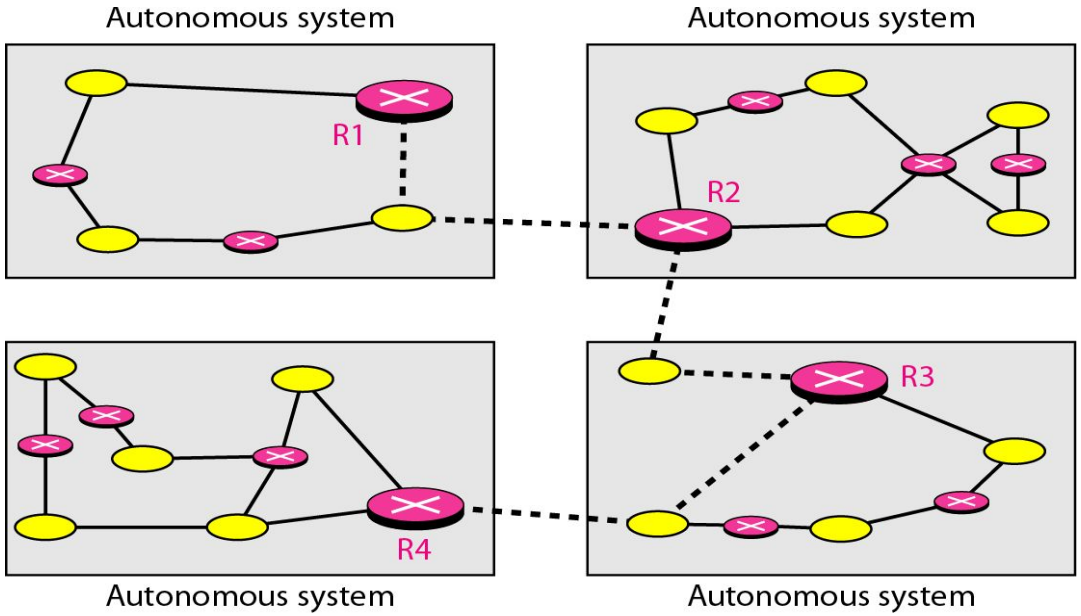
An **autonomous system** (AS) is a group of **networks and routers** under the authority of a **single administration.**

Routing **inside an autonomous system** is referred to as **intradomain routing.**

Routing **between autonomous systems** is referred to as **interdomain routing.**

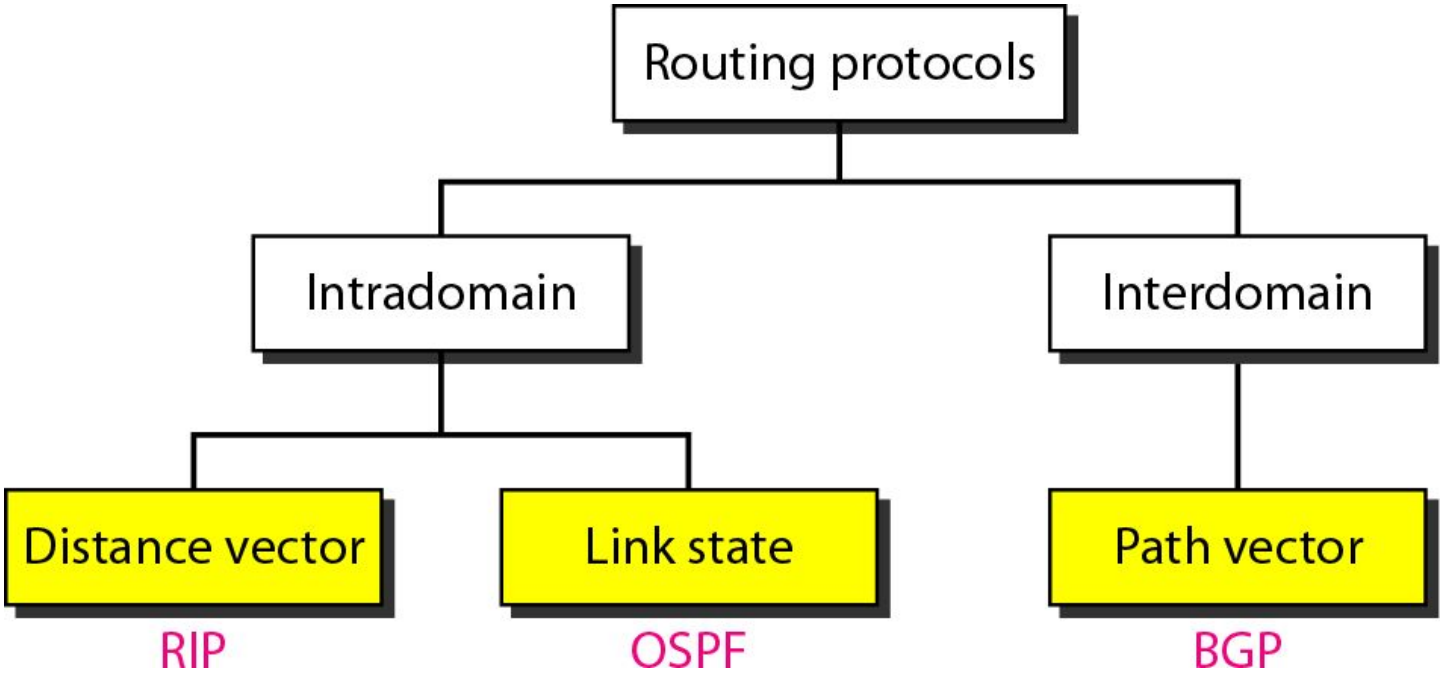
Each autonomous system can choose one or more intradomain routing protocols to handle routing inside the autonomous system.

Only one interdomain routing protocol handles routing between autonomous systems.



Popular routing protocols

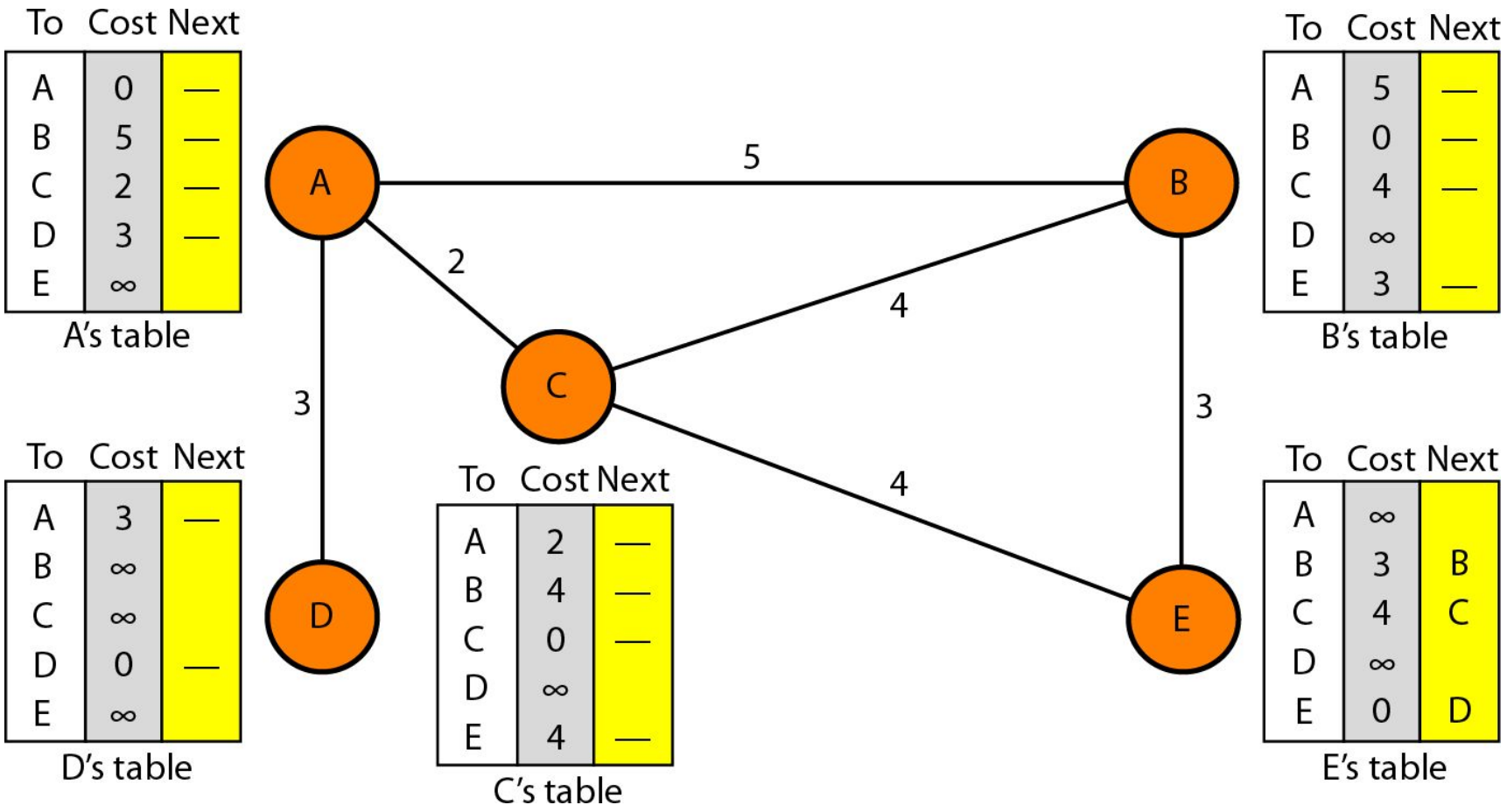
- **Routing Information Protocol** (RIP) is an implementation of the **distance vector protocol.**
- **Open Shortest Path First (OSPF)** is an implementation of the **link state protocol.**
- **Border Gateway Protocol (BGP**) is an implementation of the **path vector protocol.**



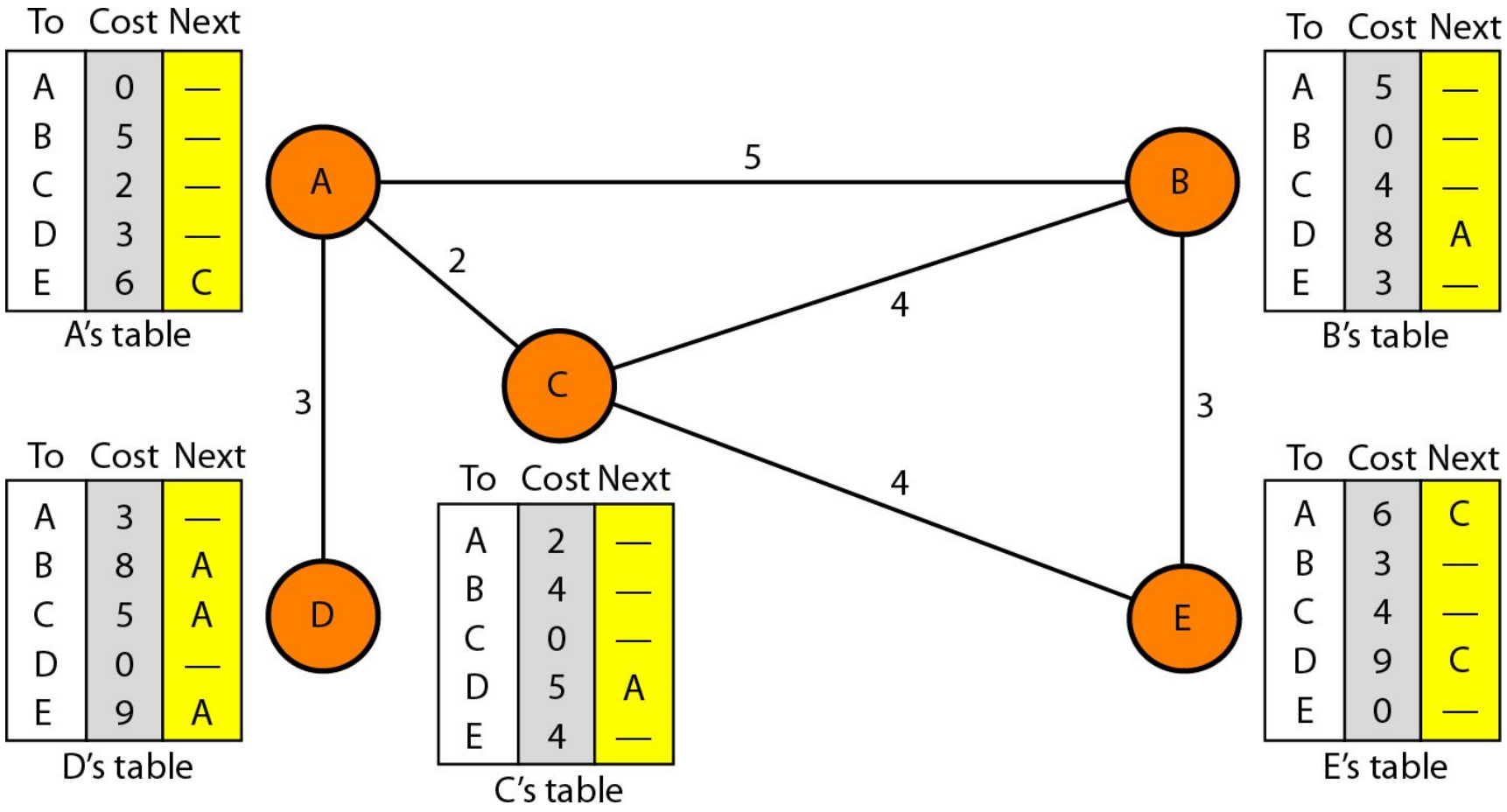
Distance Vector Routing

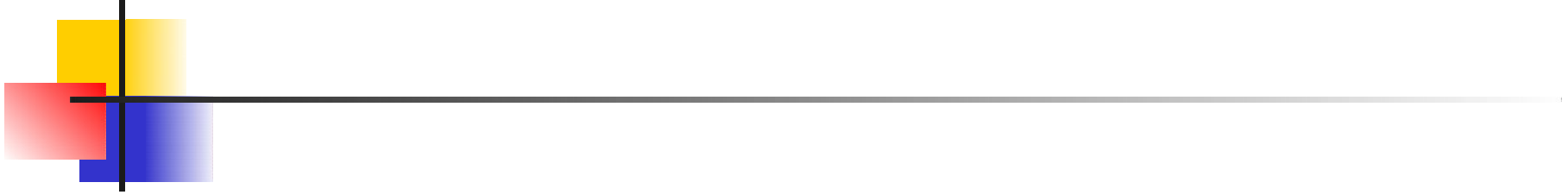
- In distance vector routing, the **least-cost route** between any two nodes is the route with **minimum distance**.
- In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.
- The table at each node also guides the packets to the desired node by showing the next stop in the route.
- Each node can know only the distance between itself and its immediate neighbors, those directly connected to it.
- Each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.
- The distance for any entry that is not a neighbor is marked as infinite.
- The whole idea of distance vector routing is the sharing of information between neighbors.
- Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

Initialization of tables in distance vector routing



Distance vector routing tables





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

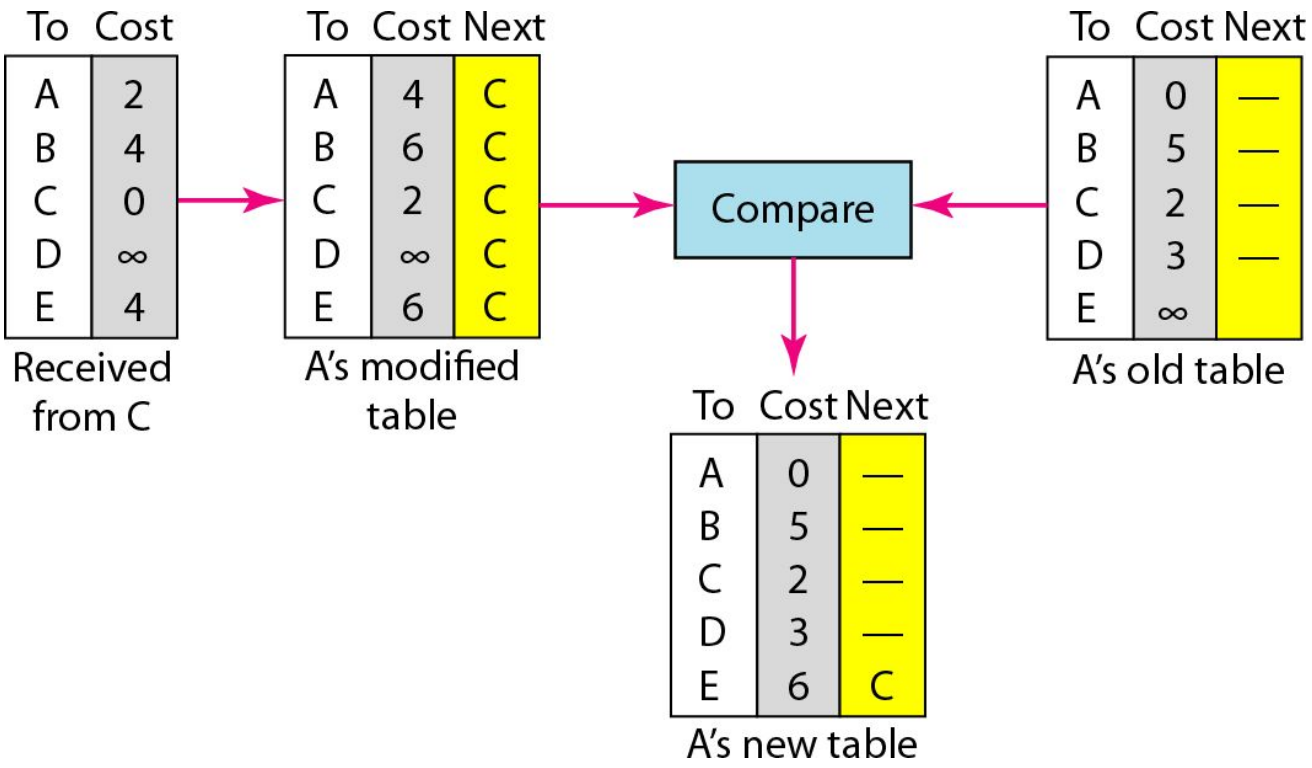
Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is an intradomain 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 than 15 hops.
5. The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Updating in distance vector routing



Link state routing

- In link state routing, if each node in the domain has the entire topology of the domain - the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.

Building Routing Tables

- In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

1. Creation of the states of the links by each node, called the **link state packet** (LSP).
2. Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Creation of Link State Packet (LSP)

A link state packet can carry a large amount of information. For the moment, however, we assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time. LSPs are generated on two occasions:

1. When there is a change in the topology of the domain. Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.

2. On a periodic basis. The period in this case is much longer compared to distance vector routing. As a matter of fact, there is no actual need for this type of LSP dissemination.

It is done to ensure that old information is removed from the domain.

The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

Flooding of LSPs

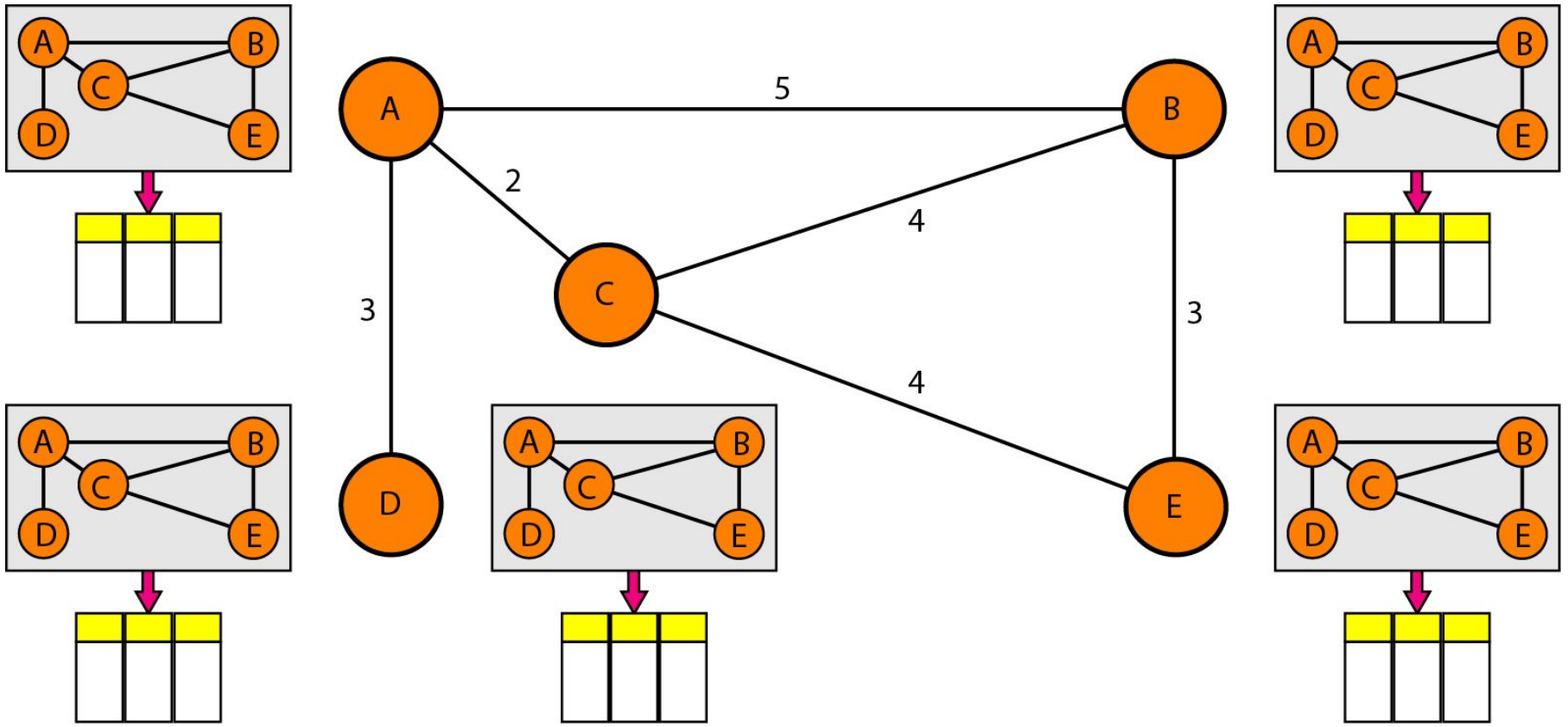
After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:

1. The creating node sends a copy of the LSP out of each interface.
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:

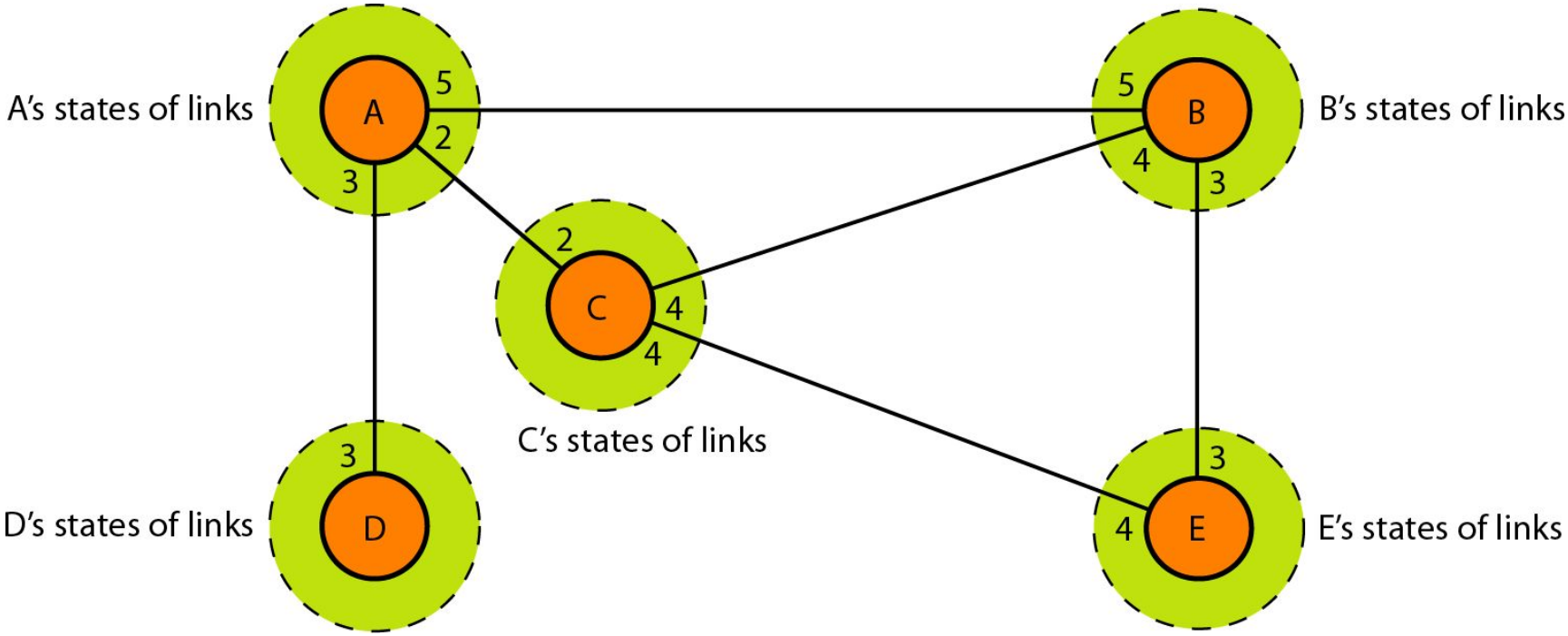
a. It discards the old LSP and keeps the new one.

b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain.

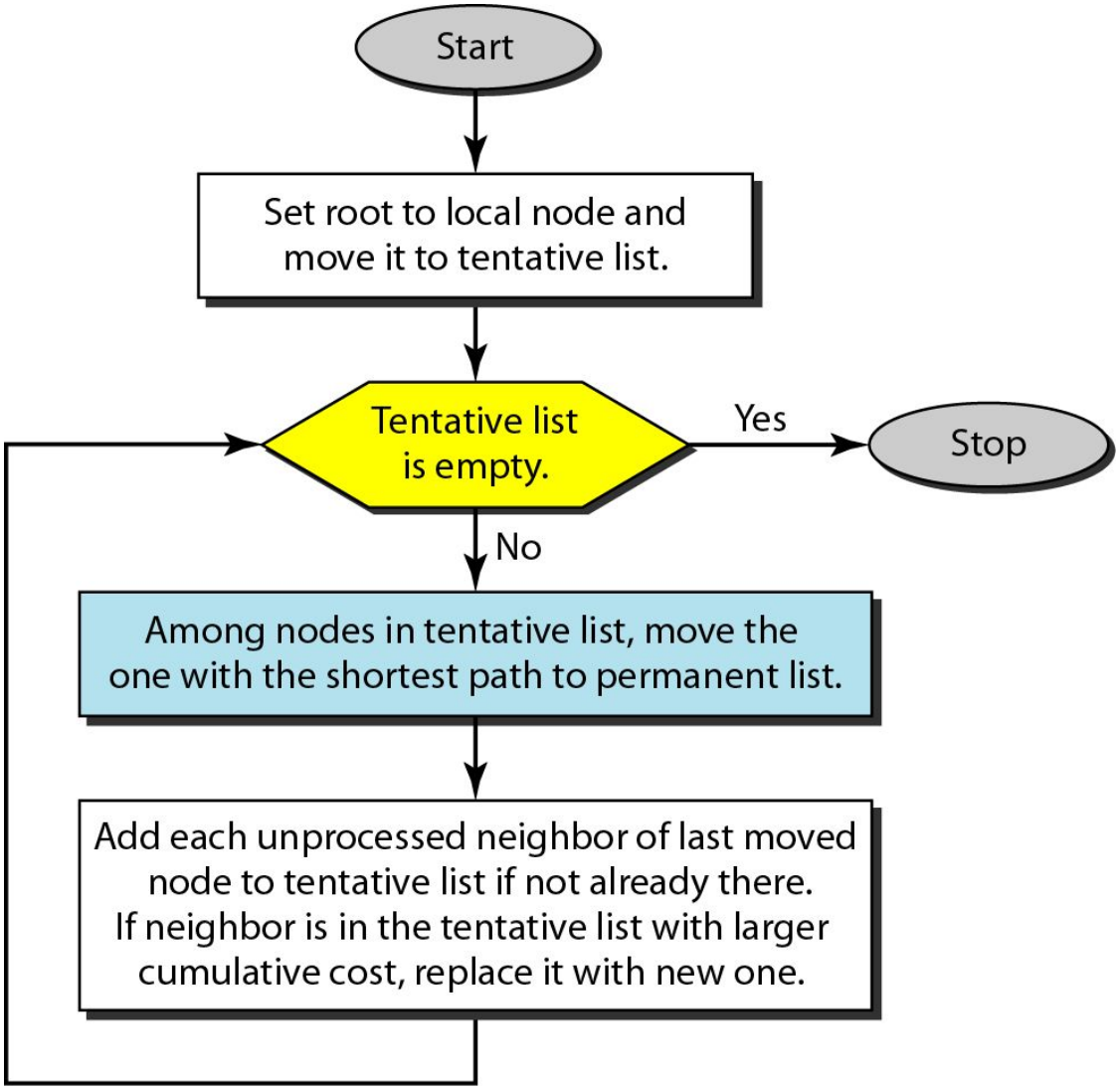
Concept of link state routing



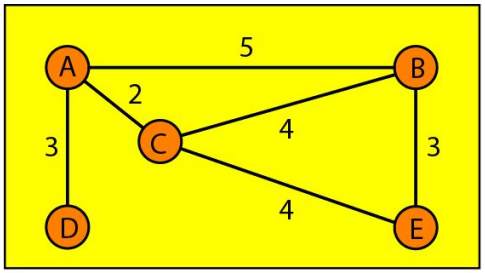
Link state knowledge



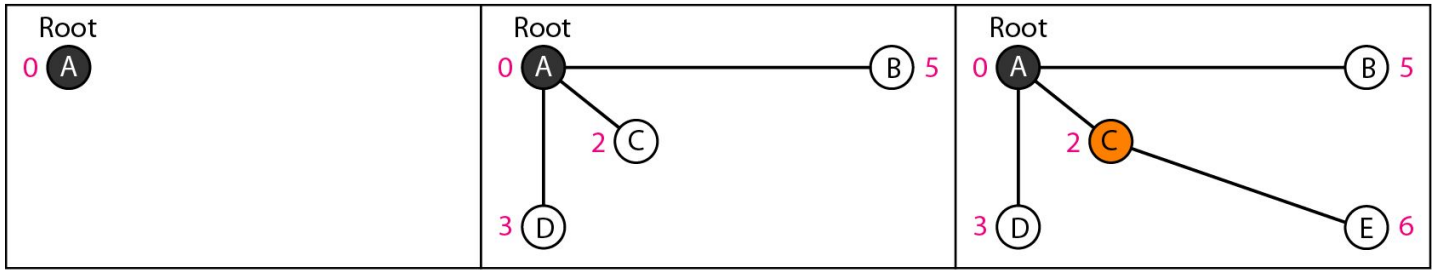
Dijkstra algorithm



Example of formation of shortest path tree



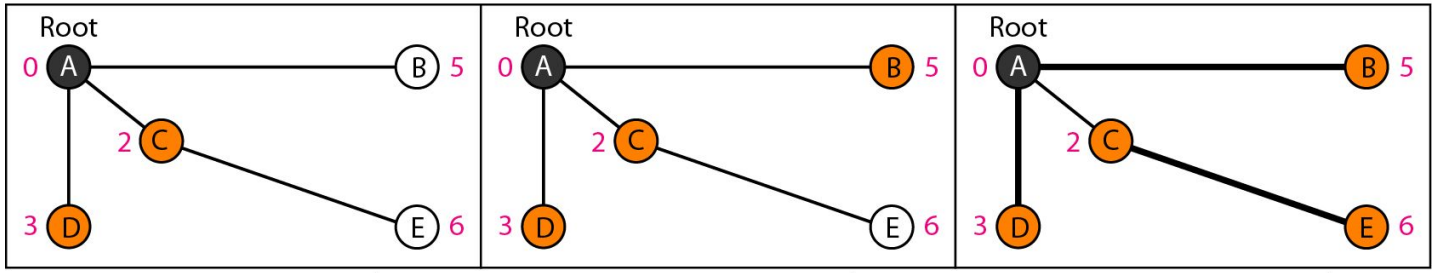
Topology



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.



4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

Calculation of Routing Table from Shortest Path Tree

Each node uses the shortest path tree protocol to construct its routing table.

The routing table shows the cost of reaching each node from the root.

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

Routing table for node A

OSPF

- The Open Shortest Path First or OSPF protocol is an intradomain routing protocol based on link state routing.
- Its domain is also an autonomous system.
- To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas.
- An area is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be divided into many different areas. All networks inside an area must be connected.

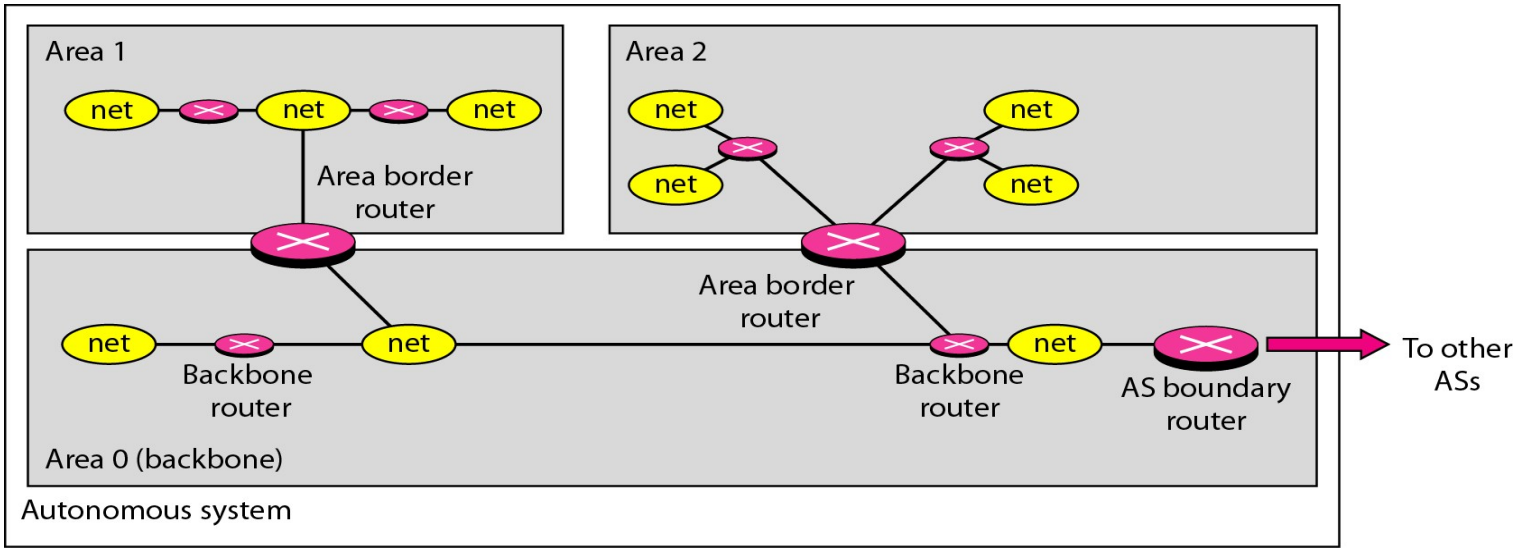
Areas in an autonomous system

Routers inside an area flood the area with routing information. At the border of an area, special routers called **area border routers** summarize the information about the area and send it to other areas.

Among the areas inside an autonomous system is a special area called the ***backbone;*** all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a **primary area** and the other areas as **secondary areas**. This does not mean that the routers within areas cannot be connected to each other, however. The routers inside the backbone are called the backbone routers. Note that a backbone router can also be an area border router.

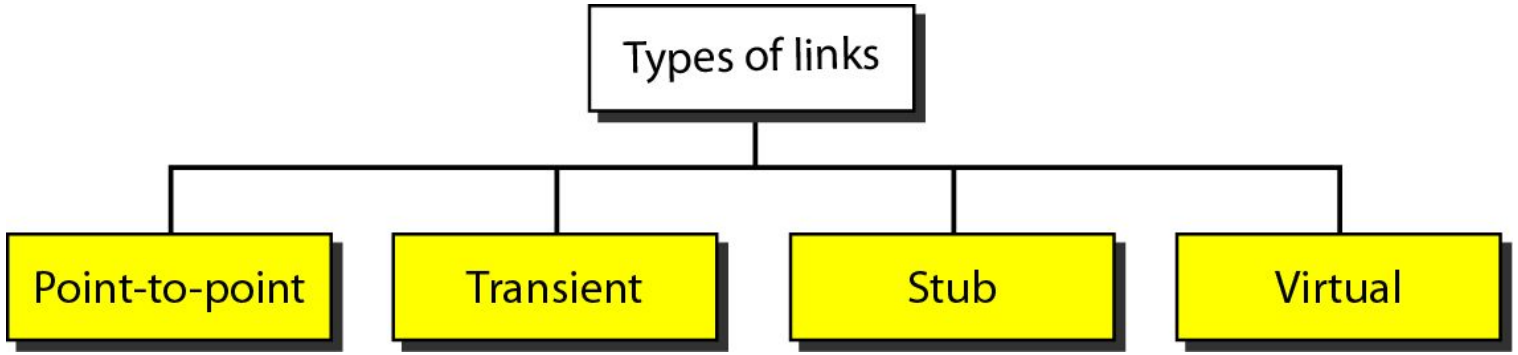
If, because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area.

Each area has an area identification. The area identification of the backbone is zero.



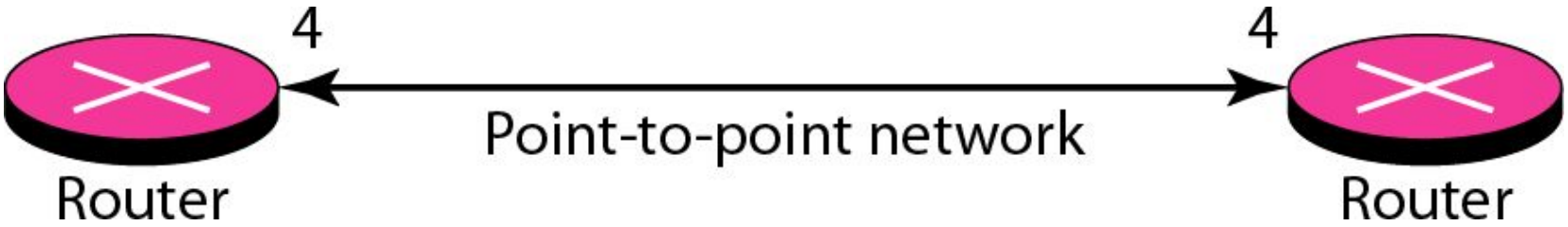
Types of links

A connection is called a link.



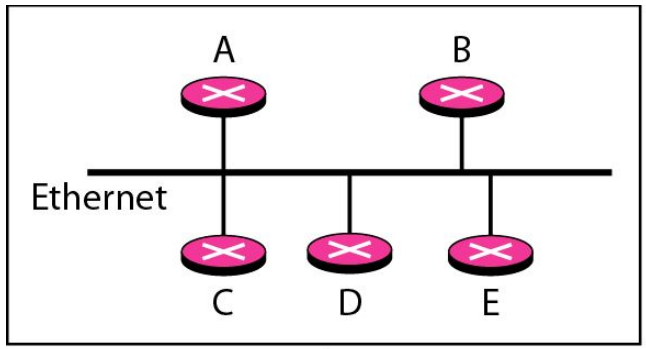
Point-to-point link

- A point-to-point link connects two routers without any other host or router in between.
- An example of this type of link is two routers connected by a telephone line or a T line. There is no need to assign a network address to this type of link.
- Graphically, the routers are represented by nodes, and the link is represented by a bidirectional edge connecting the nodes.
- The metrics, which are usually the same, are shown at the two ends, one for each direction.
- Each router has only one neighbor at the other side of the link.

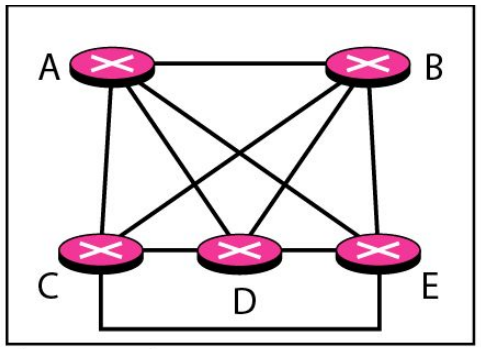


Transient link

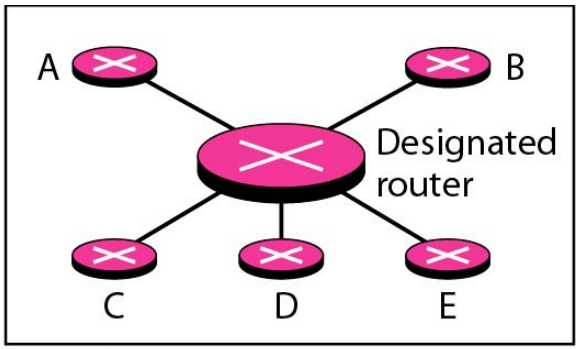
- A transient link is a network with several routers attached to it.
- The data can enter through any of the routers and leave through any router.
- All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors.
- For example, consider the Ethernet, Router A has routers B, C, D, and E as neighbors.
- Router B has routers A, C, D, and E as neighbors.
- If we want to show the neighborhood relationship in this situation, we have the graph.



a. Transient network



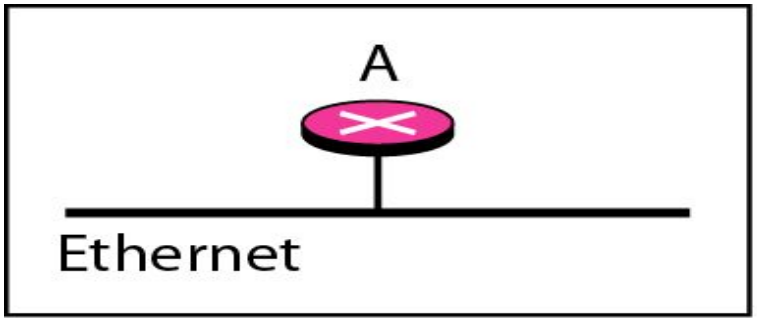
b. Unrealistic representation



c. Realistic representation

Stub link

- A **stub link** is a network that is connected to only one router.
- The data packets enter the network through this single router and leave the network through this same router.
- This is a special case of the transient network. the router is a node and using the designated router for the network.
- The link is only one-directional, from the router to the network



a. Stub network

b. Representation

- When the link between two routers is broken, the administration may create a **virtual link** between them, using a longer path that probably goes through several routers.

Path vector routing

- Path vector routing is useful for **interdomain routing.**
- The principle of path vector routing is similar to that of distance vector routing.
- In path vector routing, that there is **one node** in each autonomous system that **acts on behalf** of the entire autonomous system, known as **speaker node**.
- The speaker node in an AS creates a **routing table** and **advertises** it to speaker nodes in the neighboring ASs.
- The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other.
- A speaker node **advertises the path**, not the metric of the nodes, in its autonomous system or other autonomous system.

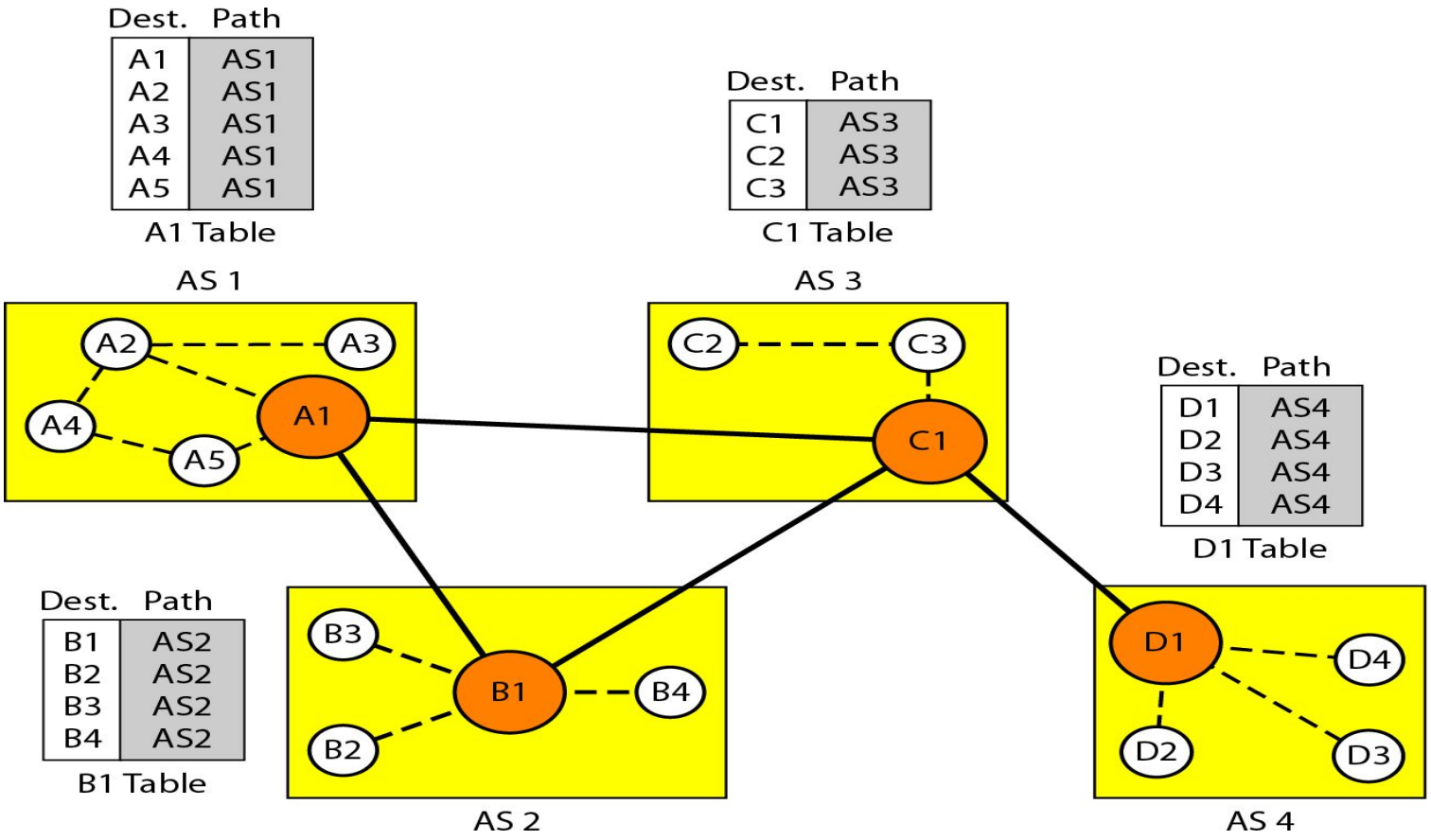
Initial routing tables in path vector routing

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system.

Node A1is the speaker node for AS1, B1 for AS2, C1 for AS3, and D1 for AS4.

Node A1 creates an initial table that shows A1to A5 are located in AS1 and can be reached through it.

Node B1 advertises that B1 to B4 are located in AS2 and can be reached through B1.



Sharing

Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

Node A1 shares its table with nodes B1 and C1.

Node C1 shares its table with nodes D1, B1, and A1.

Node B1 shares its table with C1 and A1.

Node D1 shares its table with C1.

Updating

When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table.

After a while each speaker has a table and knows how to reach each node in other ASs.

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

A1 Table

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

B1 Table

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

C1 Table

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

D1 Table

Border Gateway Protocol

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing.

In BGP, autonomous systems are classified into three categories: stub, multihomed, and transit.

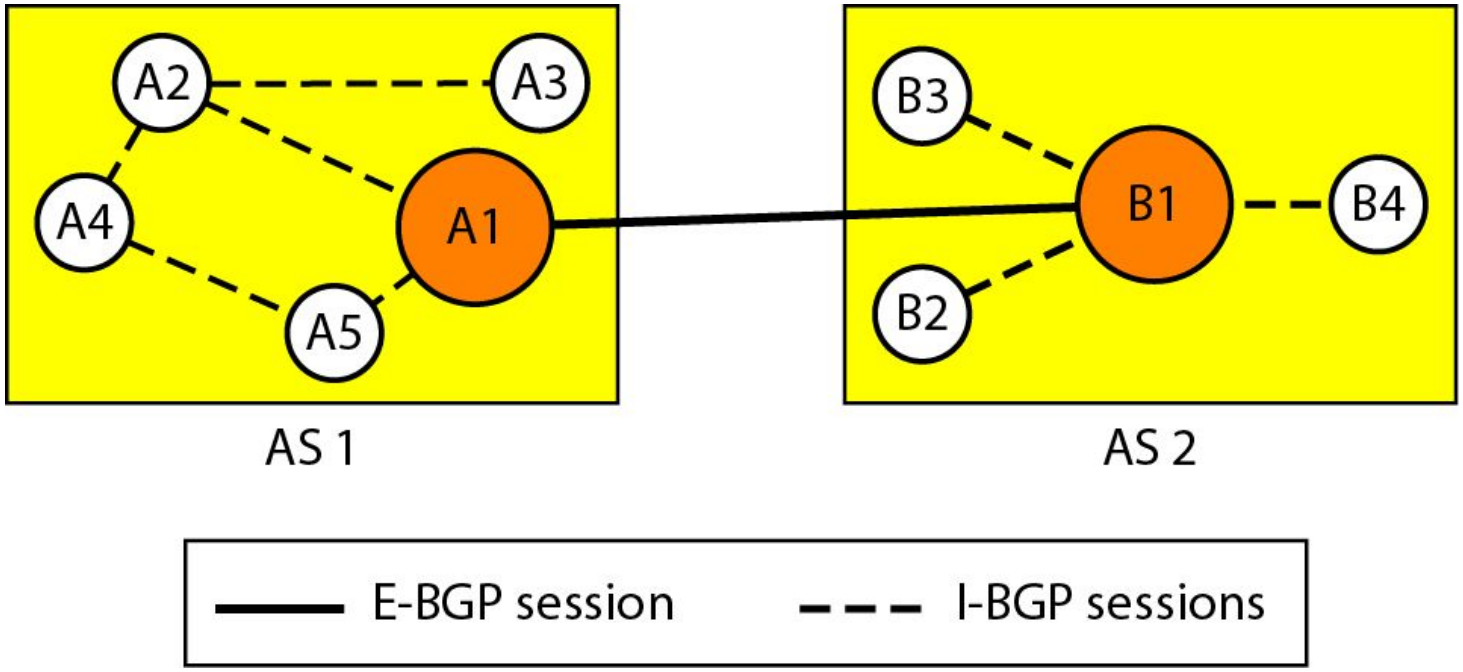
Stub AS: A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink.

Multihomed AS: A multihomed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS.

Transit AS: A transit AS is a multihomed AS that also allows transient traffic.

Internal and external BGP sessions

- BGP can have two types of sessions: **external BGP (E-BGP) and internal BGP (I-BGP) sessions.**
- The **E-BGP** session is used to exchange information between two speaker nodes belonging to two different autonomous systems.
- The **I-BGP** session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.



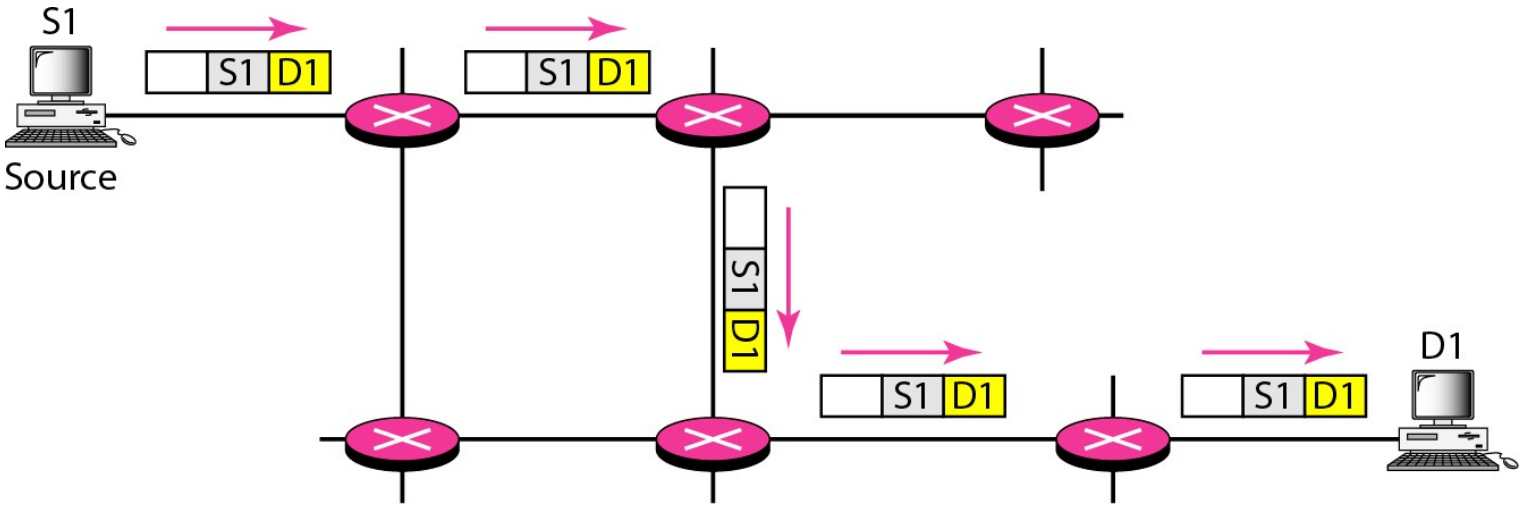
22-4 MULTICASTROUTINGPROTOCOLS

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

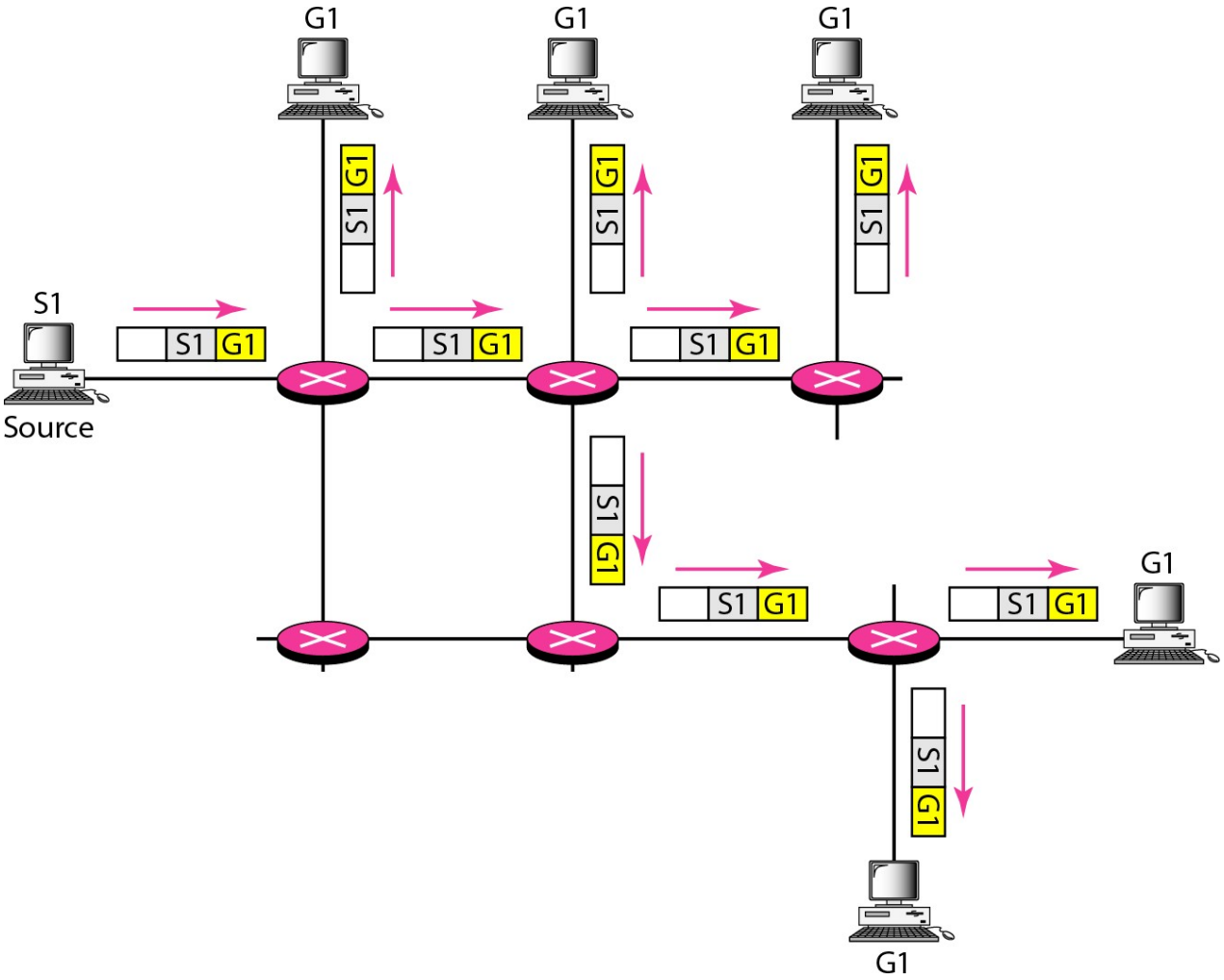


Note

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

Figure 22.34

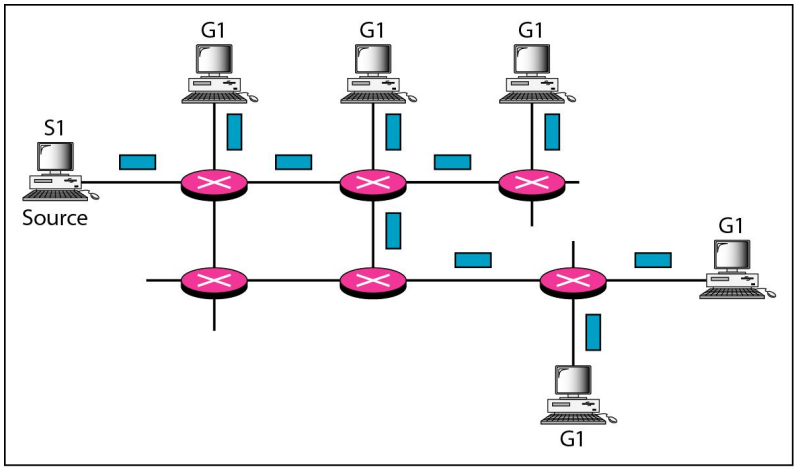
Multicasting



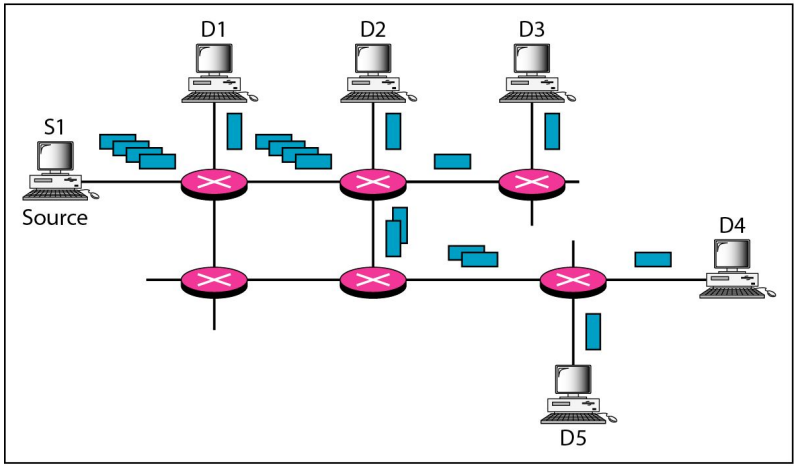
Note

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

Figure 22.35 *Multicasting versus multiple unicasting*



a. Multicasting



b. Multiple unicasting

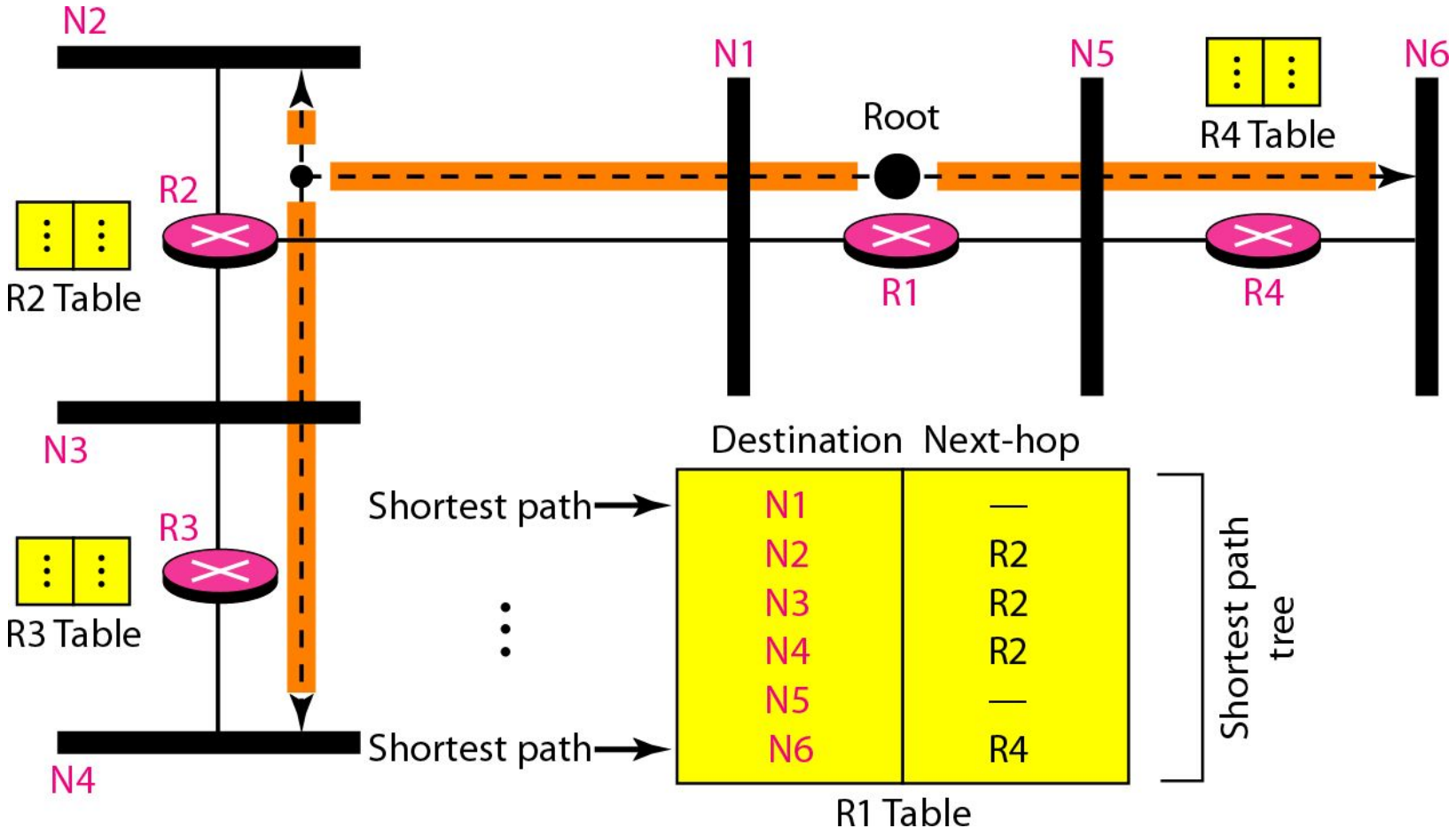
Note

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.

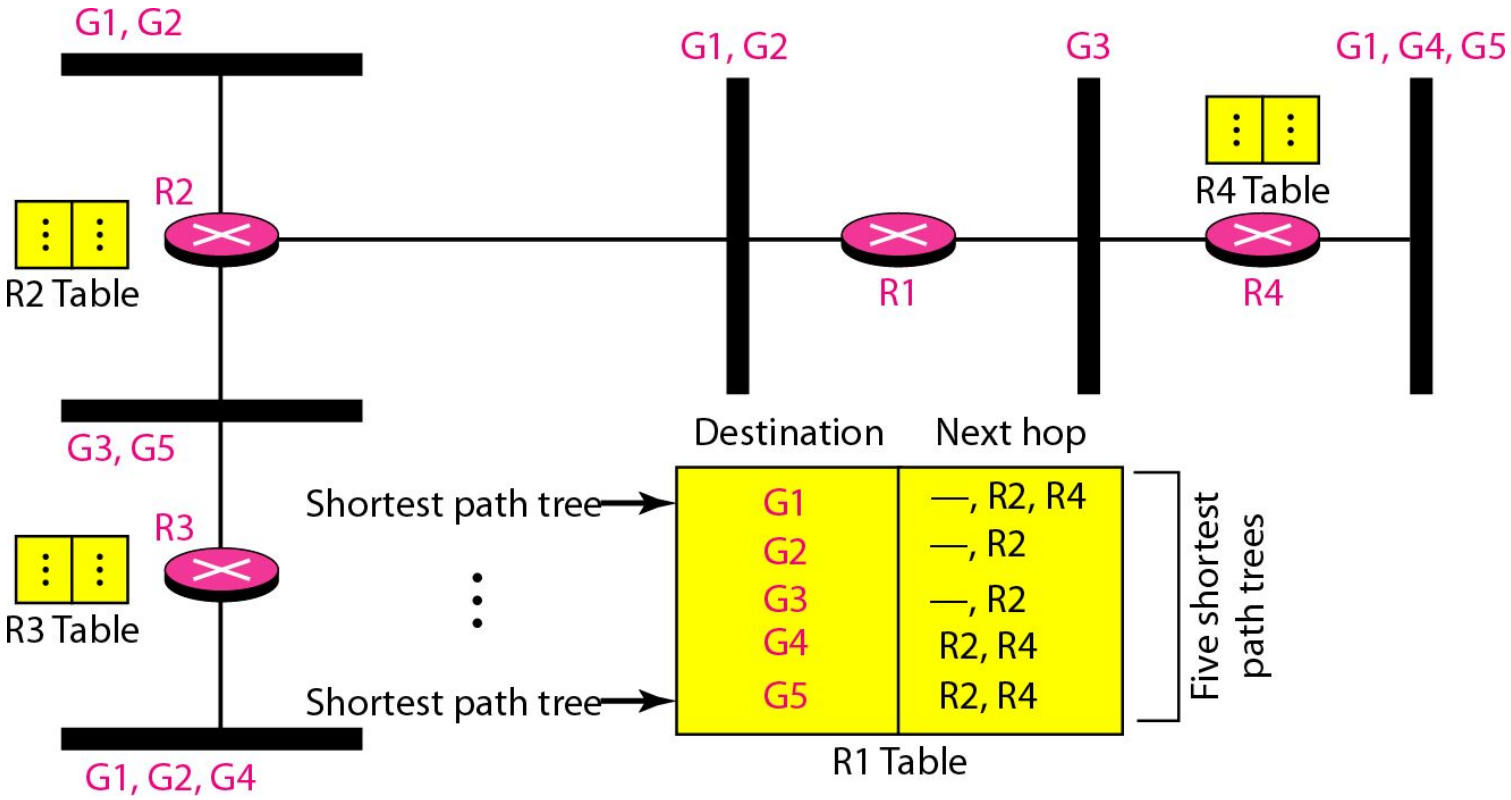
Figure 22.36 *Shortest path tree in unicast routing*



Note

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

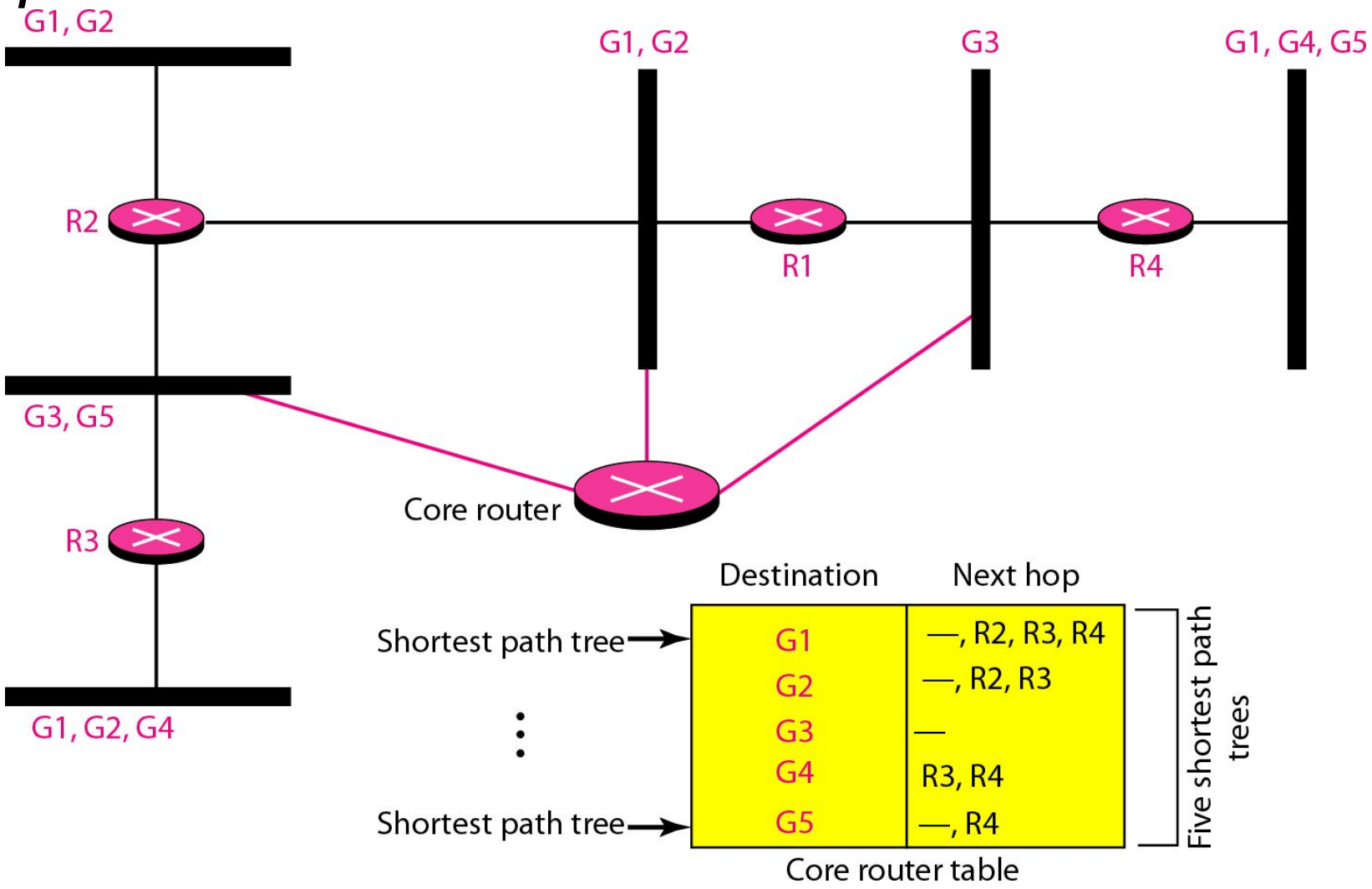
Figure 22.37 *Source-based tree approach*



Note

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

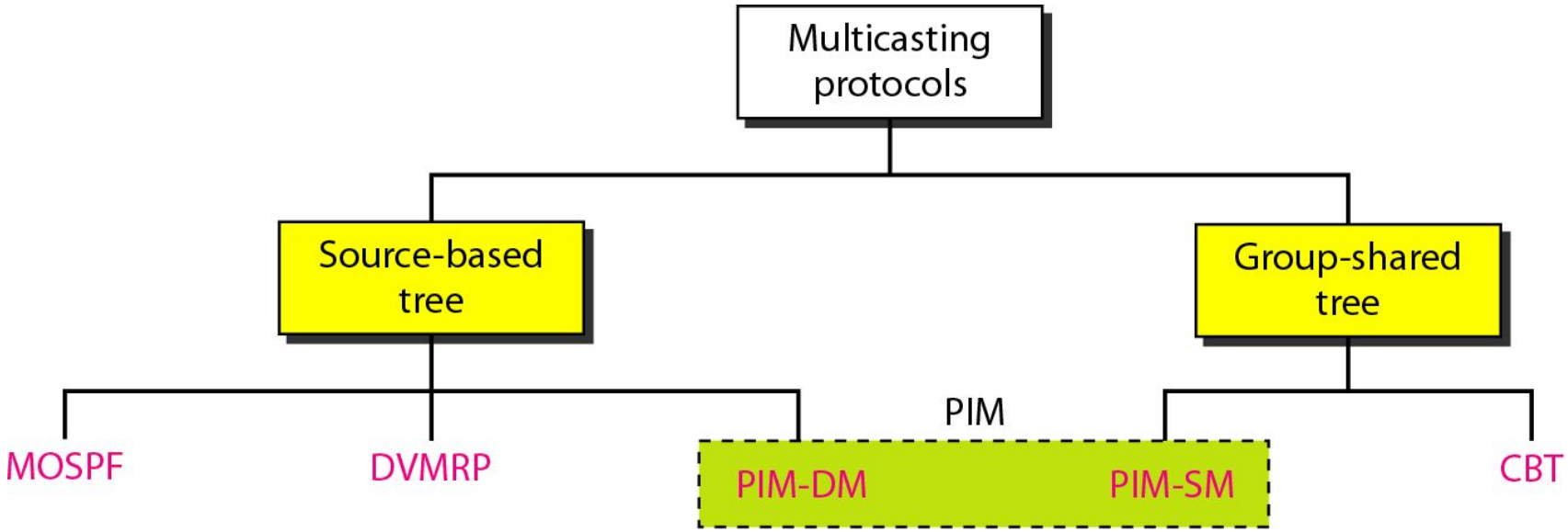
Figure 22.38 *Group-shared tree approach*



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)*

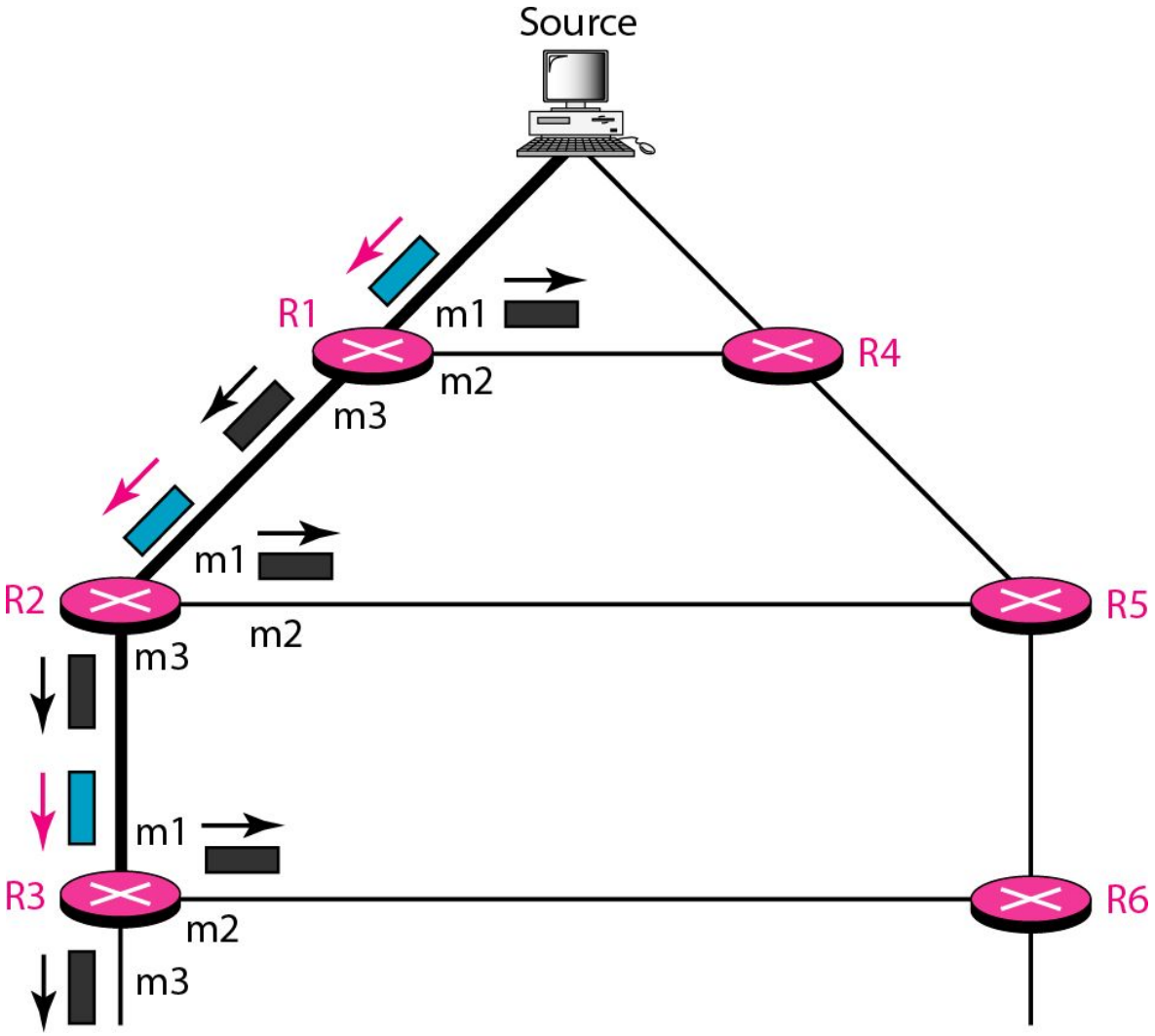
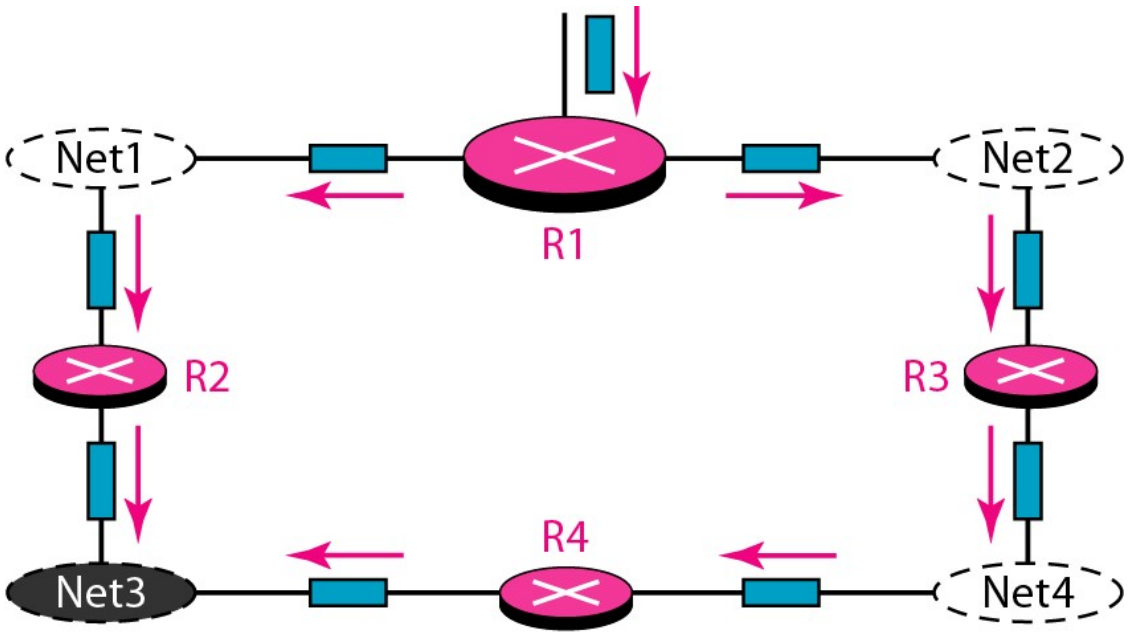
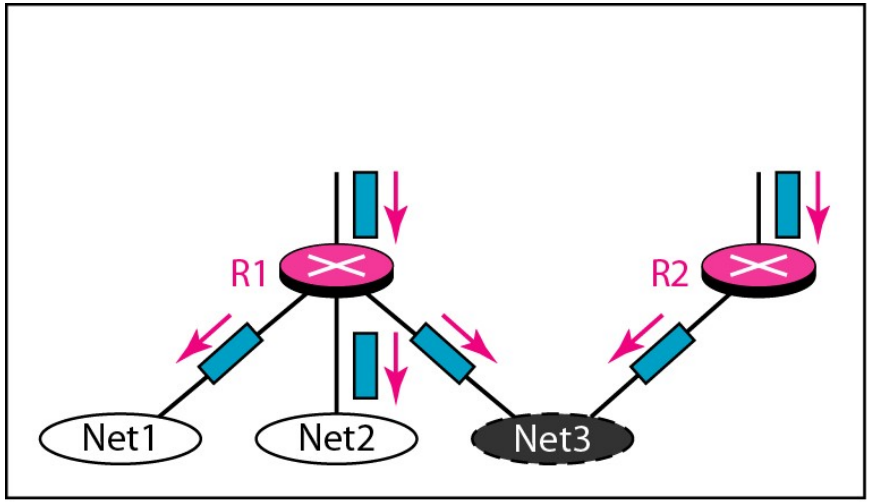


Figure 22.41 *Problem with RPF*

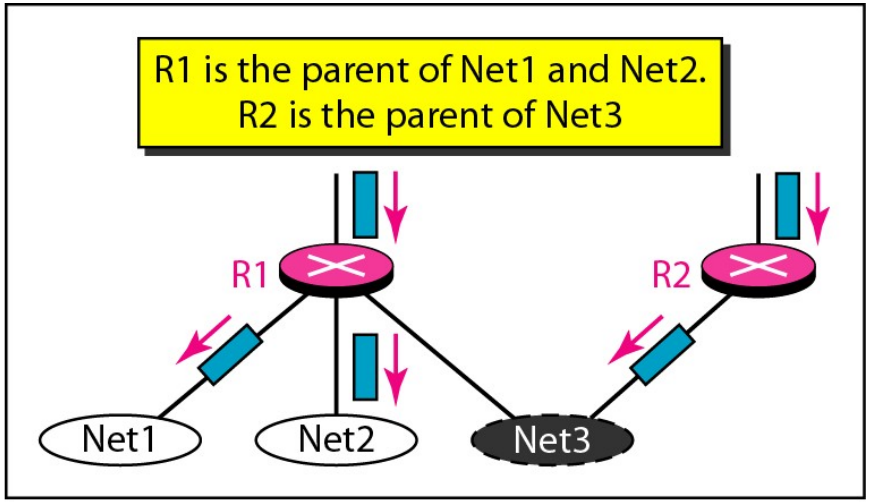


Net3 receives two copies of the packet

Figure 22.42 *RPF Versus RPB*



a. RPF

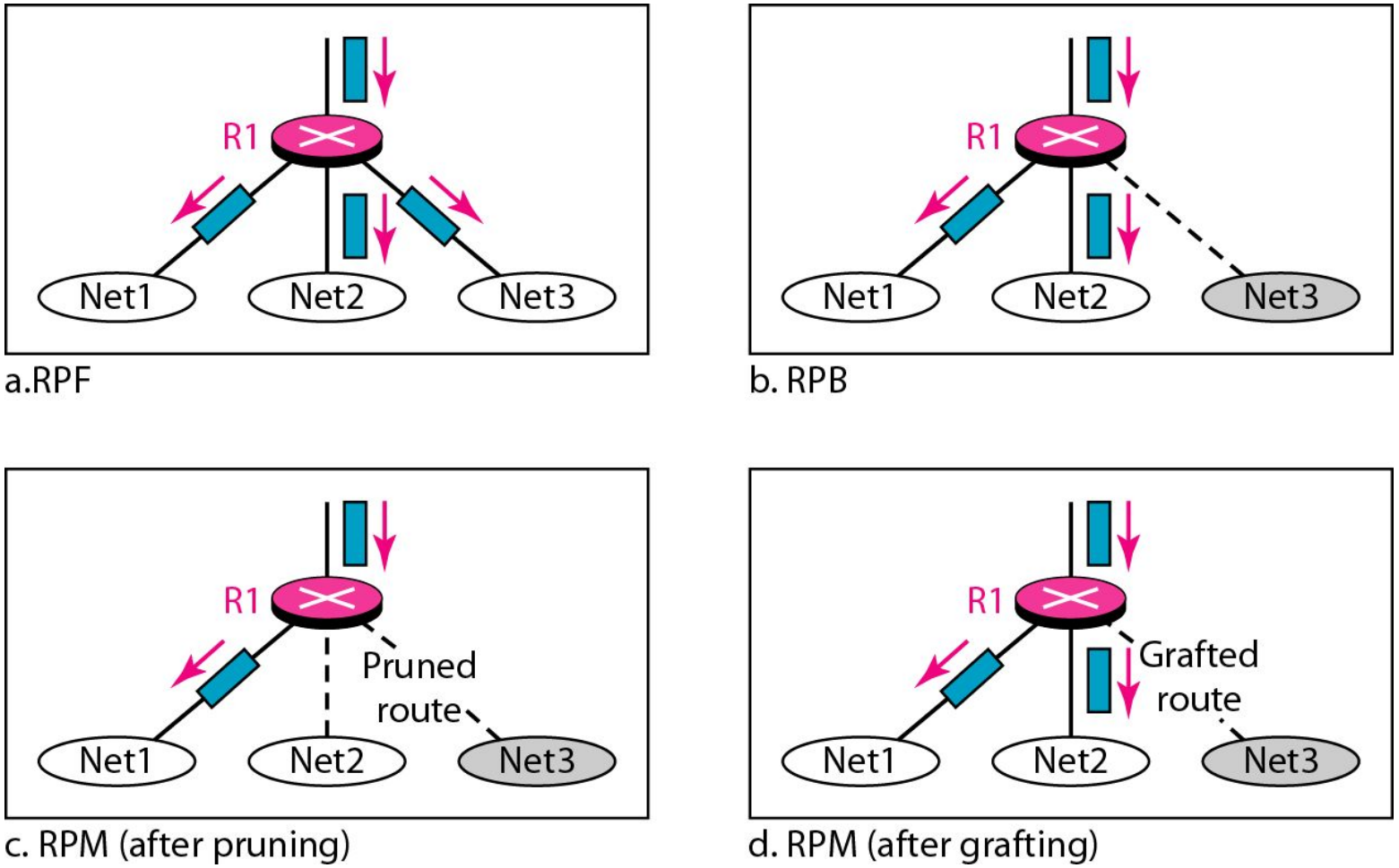


b. RPB

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*



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*

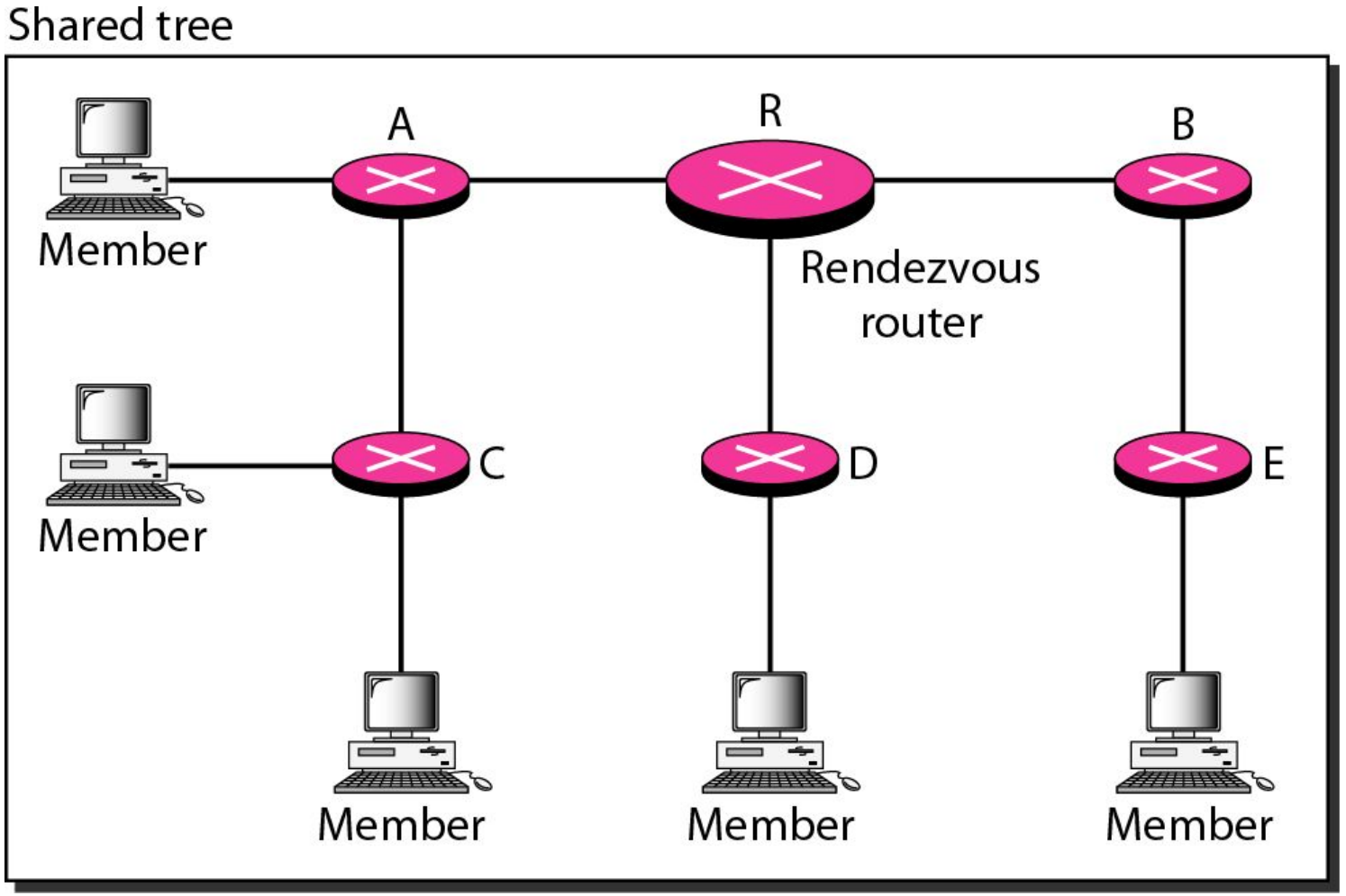
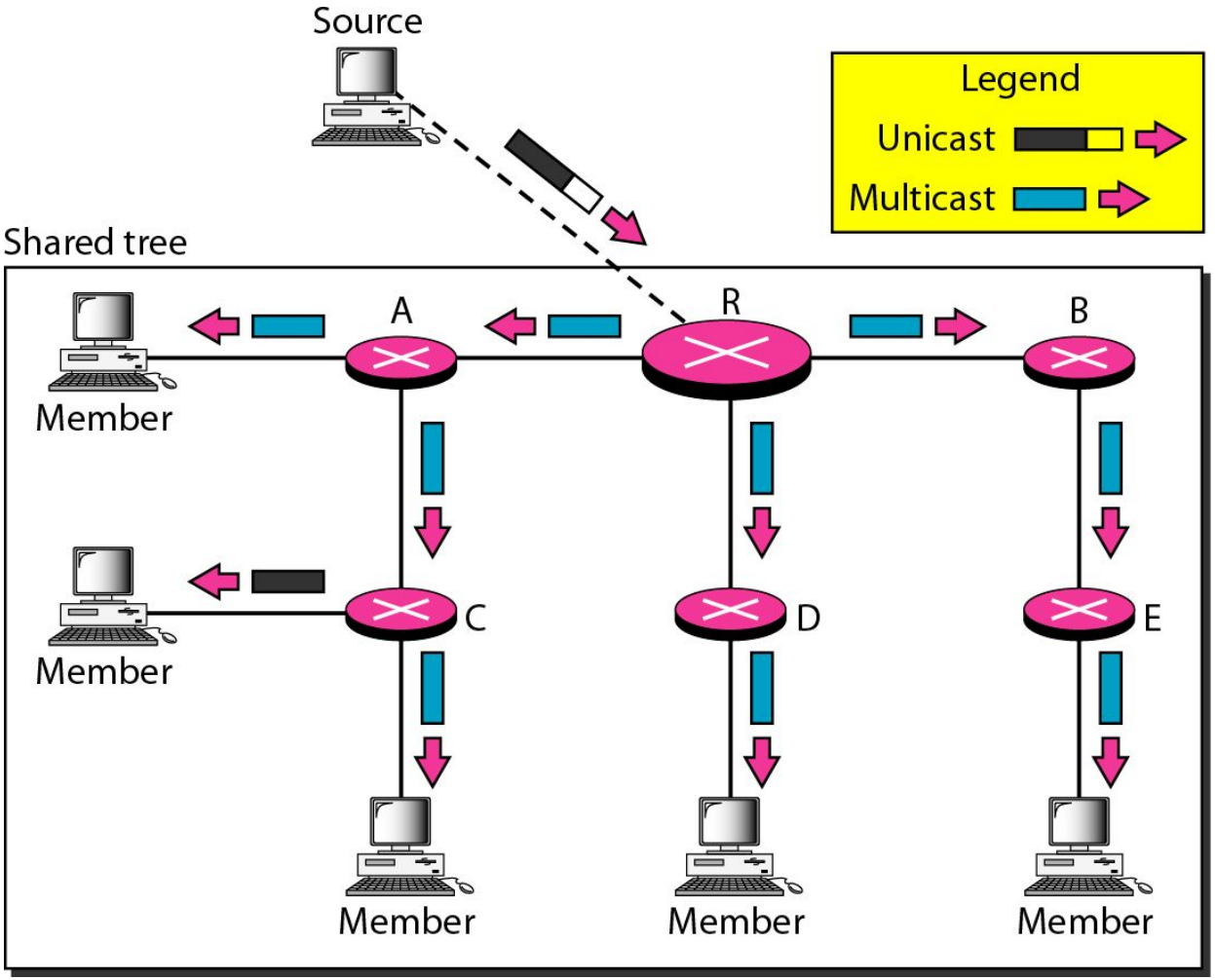


Figure 22.45 *Sending a multicast packet to the rendezvous router*



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.

Note

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

Note

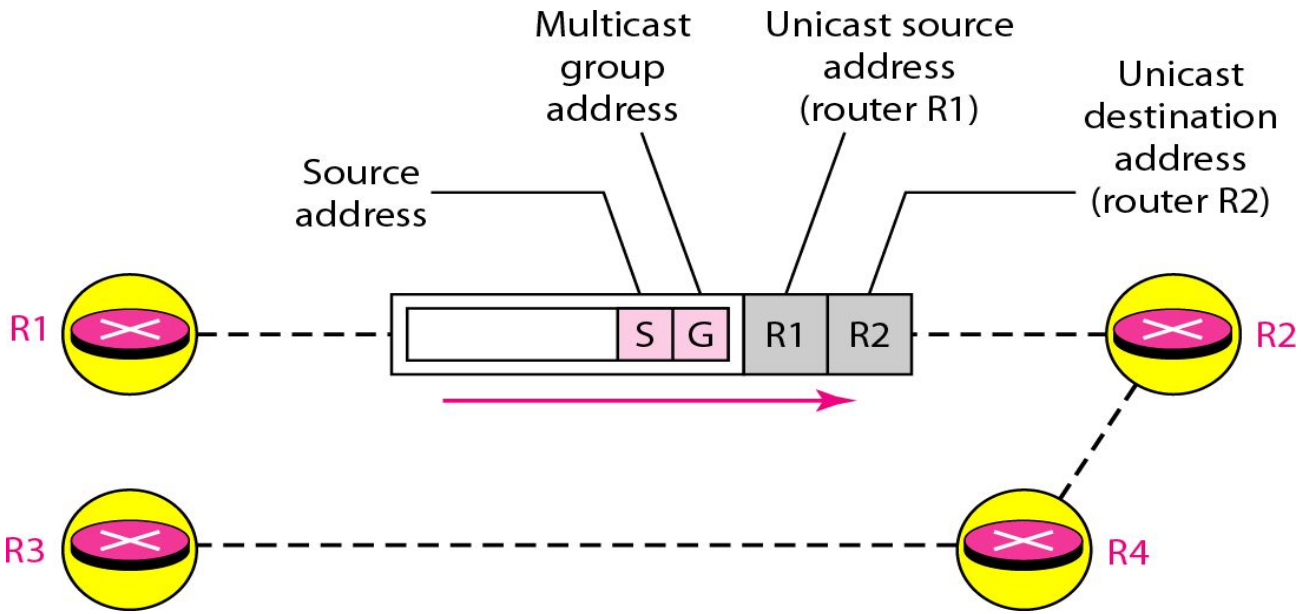
PIM-SM is used in a sparse multicast environment such as a WAN.

Note

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

MBONE

- Multimedia and real-time communication have increased the need for multicasting in the Internet.
- A multicast router may not find another multicast router in the neighborhood to forward the multicast packet.
- The multicast routers are seen as a group of routers on top of unicast routers. The multicast routers may not be connected directly, but they are connected logically. Only the routers enclosed in the shaded circles are capable of multicasting. Without tunneling, these routers are isolated islands.
- To enable multicasting, we make a multicast backbone (MBONE) out of these isolated routers by using the concept of tunneling.



A logical tunnel is established by encapsulating the multicast packet inside a unicast packet.
The multicast packet becomes the payload (data) of the unicast packet.
The intermediate (nonmulticast) routers forward the packet as unicast routers and deliver the packet from one island to another.
It's as if the unicast routers do not exist and the two multicast routers are neighbors.
The only protocol that supports MBONE and tunneling is DVMRP.