COMPUTER NETWORKS

V SEM (BCS502)

Course-Coordinator :
Dr. Kumaresh Sheelavant
Associate Professor
Dept.of.CSE(AI&ML)

Module 3
NETWORK LAYER

Chapter 18
Introduction to Network Layer

Introduction to Network Layer :

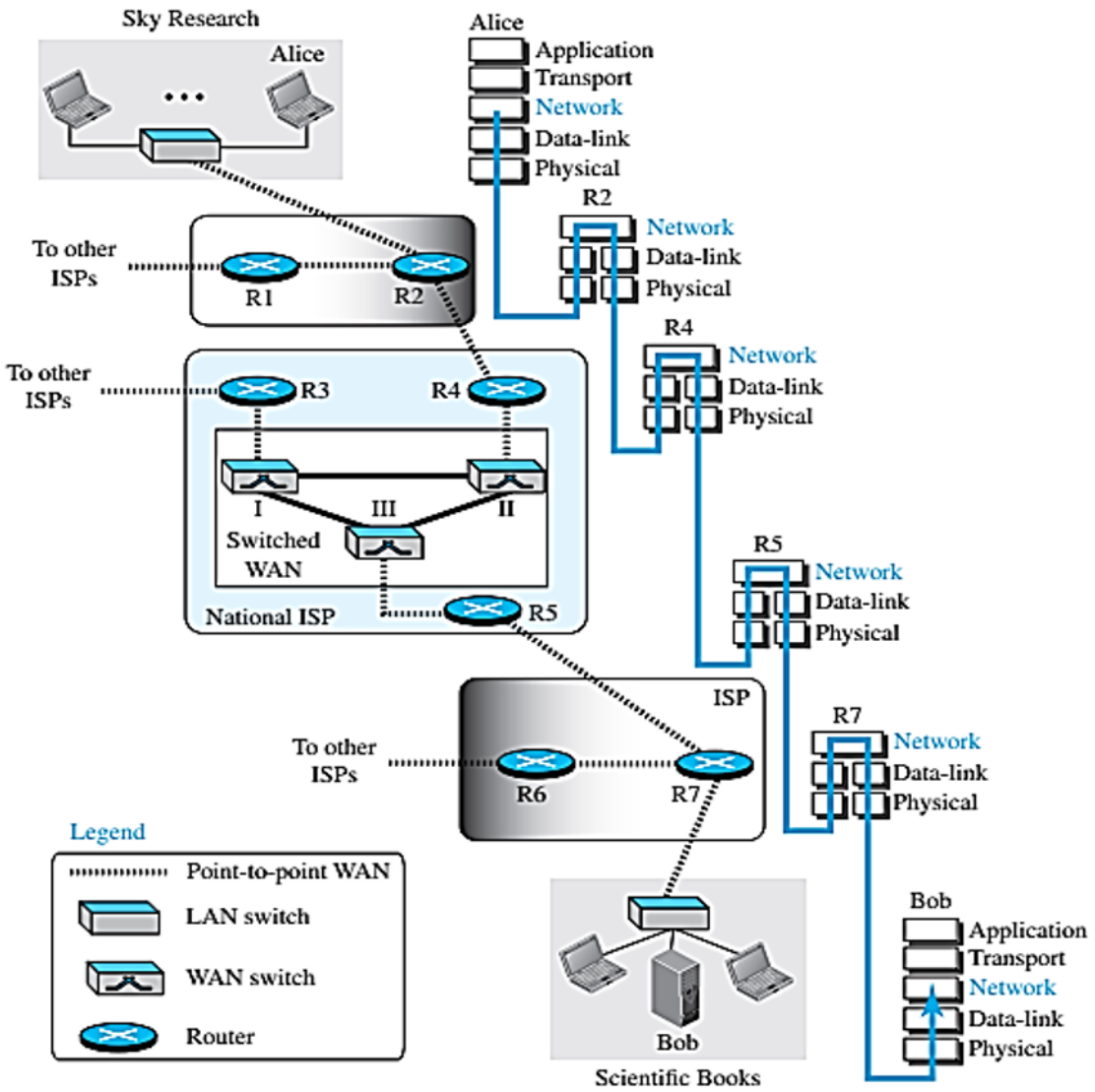
The network layer in the TCP/IP protocol suite is responsible for the host-to-host delivery of datagrams. It provides services to the transport layer and receives services from the data-link layer.

Topics discussed in this section:

- Network Layer Services
- Packet Switching
- IPv4 Addressing

Network Layer Services:

Figure 18.1 *Communication at the network layer*



Network Layer Services:

- As the figure shows, the network layer is involved at the source host, destination host, and all routers in the path (R2, R4, R5, and R7).
- At the source host (Alice), the network layer accepts a packet from a transport layer, encapsulates the packet in a datagram, and delivers the packet to the data-link layer.
- At the destination host (Bob), the datagram is decapsulated, and the packet is extracted and delivered to the corresponding transport layer.
- Although the source and destination hosts are involved in all five layers of the TCP/IP suite, the routers use three layers if they are routing packets only; however, they may need the transport and application layers for control purposes.
- A router in the path is normally shown with two data-link layers and two physical layers, because it receives a packet from one network and delivers it to another network.

Packetizing:

- The source host receives the payload from an upper-layer protocol, adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol, encapsulate it and delivers the packet to the data-link layer.

- The destination host receives the network-layer packet from its data-link layer, decapsulates the packet, and delivers the payload to the corresponding upper-layer protocol.

- If the packet is fragmented at the source or at routers along the path, the network layer is responsible for waiting until all fragments arrive, reassembling them, and delivering them to the upper-layer protocol

  The routers in the path are not allowed to decapsulate the packets they received unless the packets need to be fragmented.

Routing and Forwarding:

Routing:

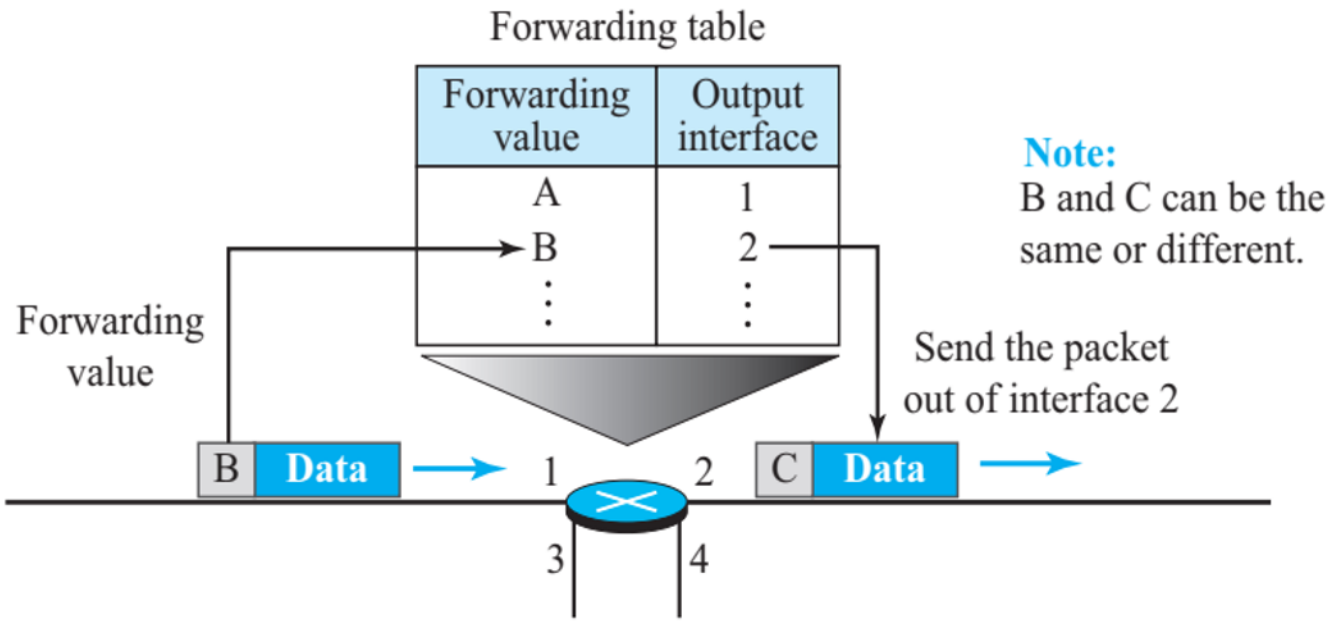
- The network layer is responsible for routing the packet from its source to the destination.
- The network layer is responsible for finding the best one among these possible routes.
- The network layer needs to have some specific strategies for defining the best route.

Forwarding:

- If routing is applying strategies and running some routing protocols to create the decision-making tables for each router, forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces.
- The decision-making table a router normally uses for applying this action is sometimes called the forwarding table and sometimes the routing table.

Forwarding :

Figure 18.2 *Forwarding process*



Other Services:

- *Error Control*
- *Flow Control*
- *Congestion Control*
- *Quality of Services*
- *Security*

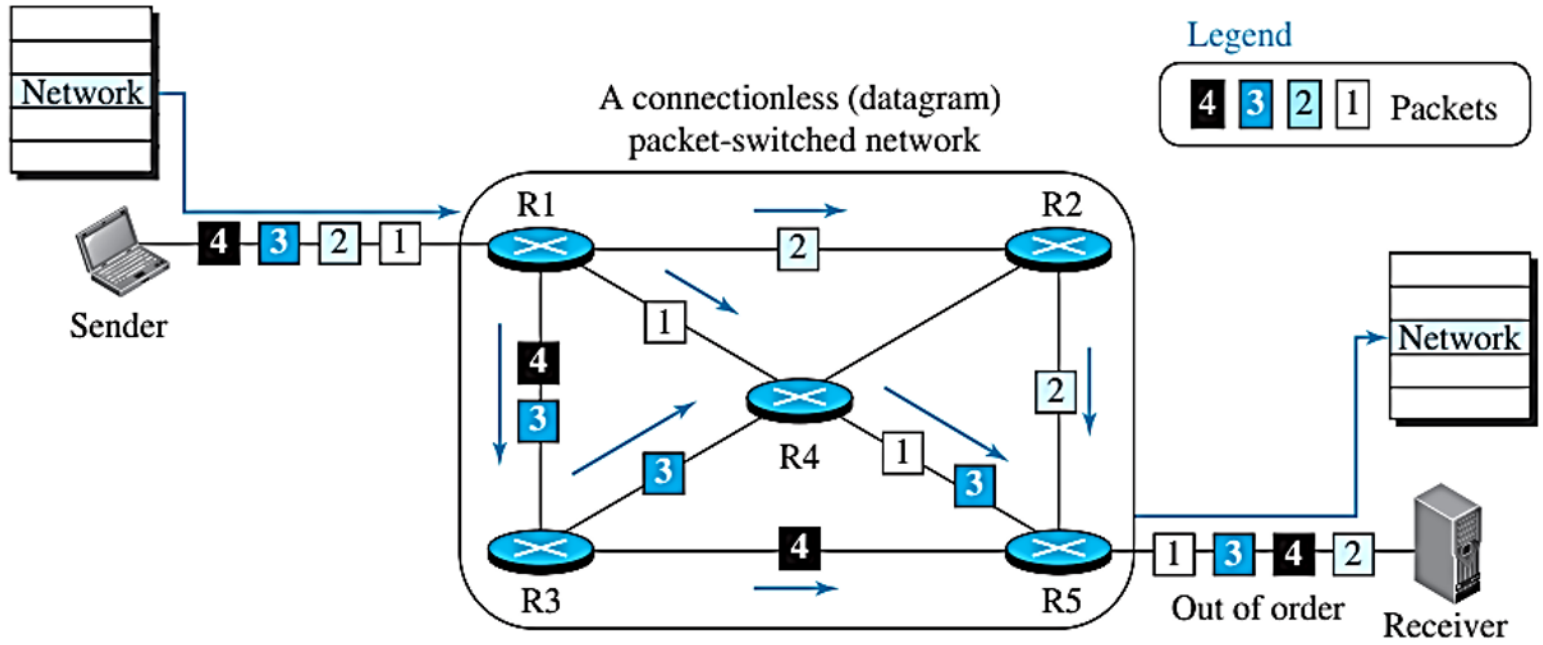
Packet Switching:

- *The source of the message sends the packets one by one; the destination of the message receives the packets one by one.*
- *The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer.*
- *The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.*
- *Two Approaches:*
  - *Datagram Approach: Connectionless Service*
  - *Virtual-Circuit Approach: Connection-Oriented Service*

Datagram Approach: Connectionless Service :

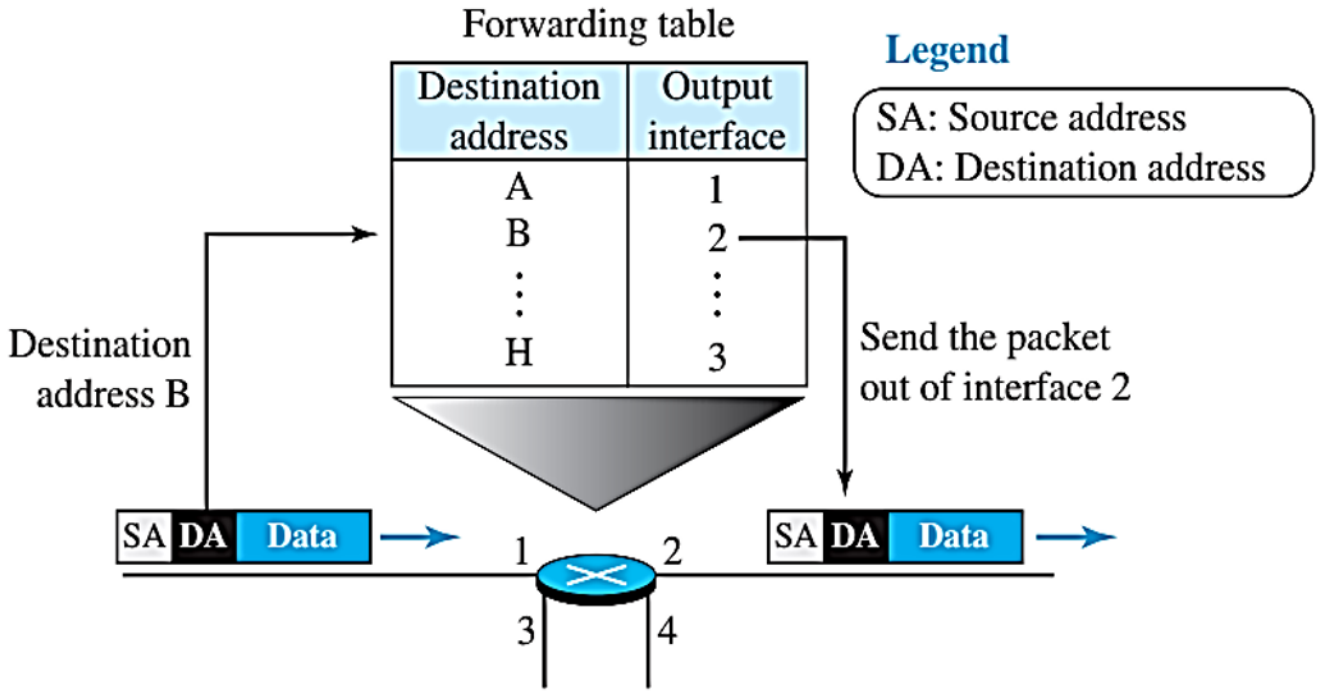
- **The network layer was designed to provide a connectionless service in which the network-layer protocol treats each packet independently, with each packet having no relationship to any other packet.**

Figure 18.3 *A connectionless packet-switched network*



Datagram Approach: Connectionless Service :

Figure 18.4 *Forwarding process in a router when used in a connectionless network*



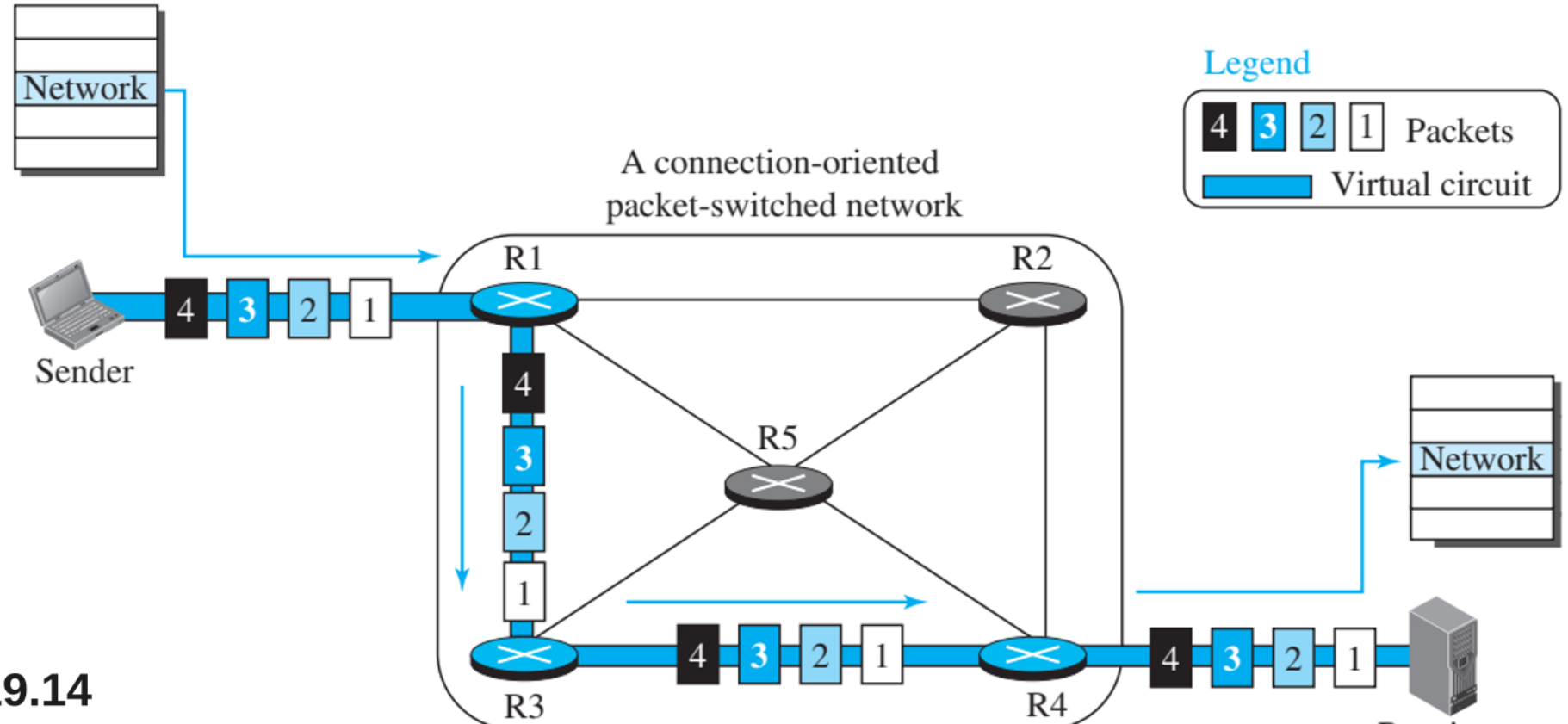
Virtual Circuit Approach: Connection-Oriented Service :

- **Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams.**
- **After *connection setup*, the datagrams can all follow the same path.**
- **In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.**
- **we assume that the packet carries this label. Although it looks as though the use of the label may make the source and destination addresses unnecessary during the data transfer phase, parts of the Internet at the network layer still keep these addresses.**

Virtual Circuit Approach: Connection-Oriented Service :

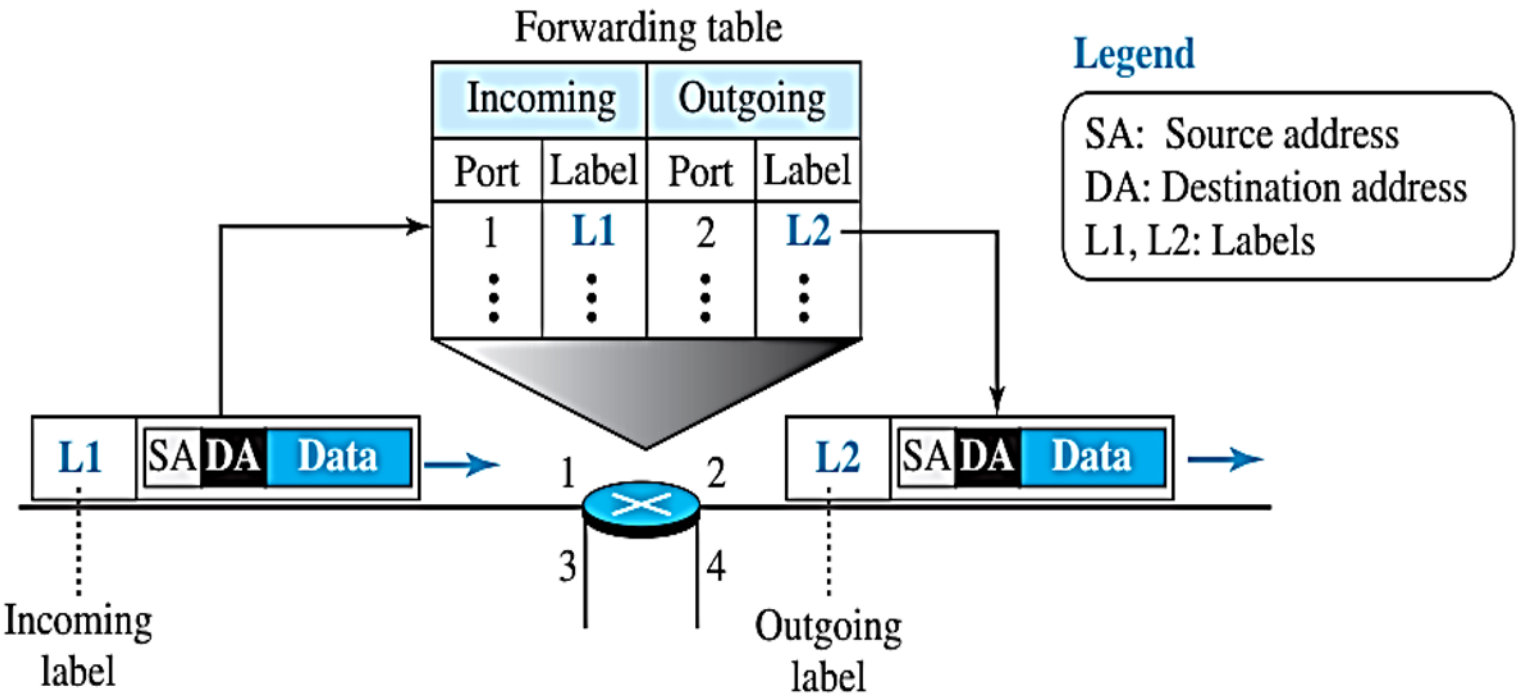
In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message.

Figure 18.5 *A virtual-circuit packet-switched network*



Virtual Circuit Approach: Connection-Oriented Service :

Figure 18.6 *Forwarding process in a router when used in a virtual-circuit network*



Virtual Circuit Approach: Connection-Oriented Service :

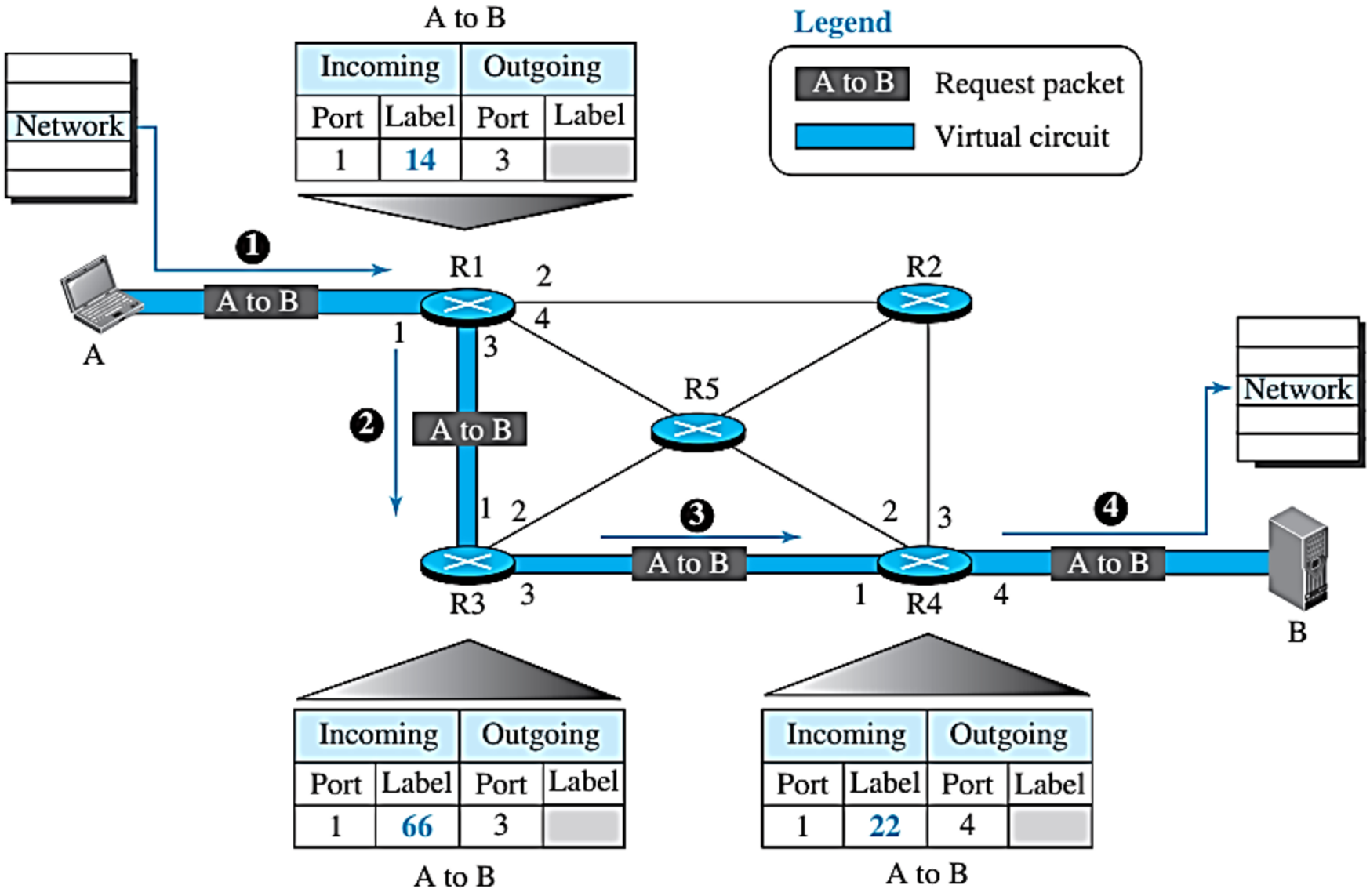
Setup Phase:

- In the setup phase, a router creates an entry for a virtual circuit.
- For example, suppose source A needs to create a virtual circuit to destination B.
- Two auxiliary packets need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet

Request packet :

- A request packet is sent from the source to the destination.
- This auxiliary packet carries the source and destination addresses

Figure 18.7 *Sending request packet in a virtual-circuit network*



Connection-Oriented Service : Request PhaSe

1. **Source A sends a request packet to router R1.**

2. **Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. How the router has obtained this information is a point covered later. The router assigns the incoming port (1) and chooses an available incoming label (14) and the outgoing port (3). It does not yet know the outgoing label, which will be found during the acknowledgment step. The router then forwards the packet through port 3 to router R3.**

3. **Router R3 receives the setup request packet. The same events happen here as at router R1; three columns of the table are completed: in this case, incoming port (1), incoming label (66), and outgoing port (3).**

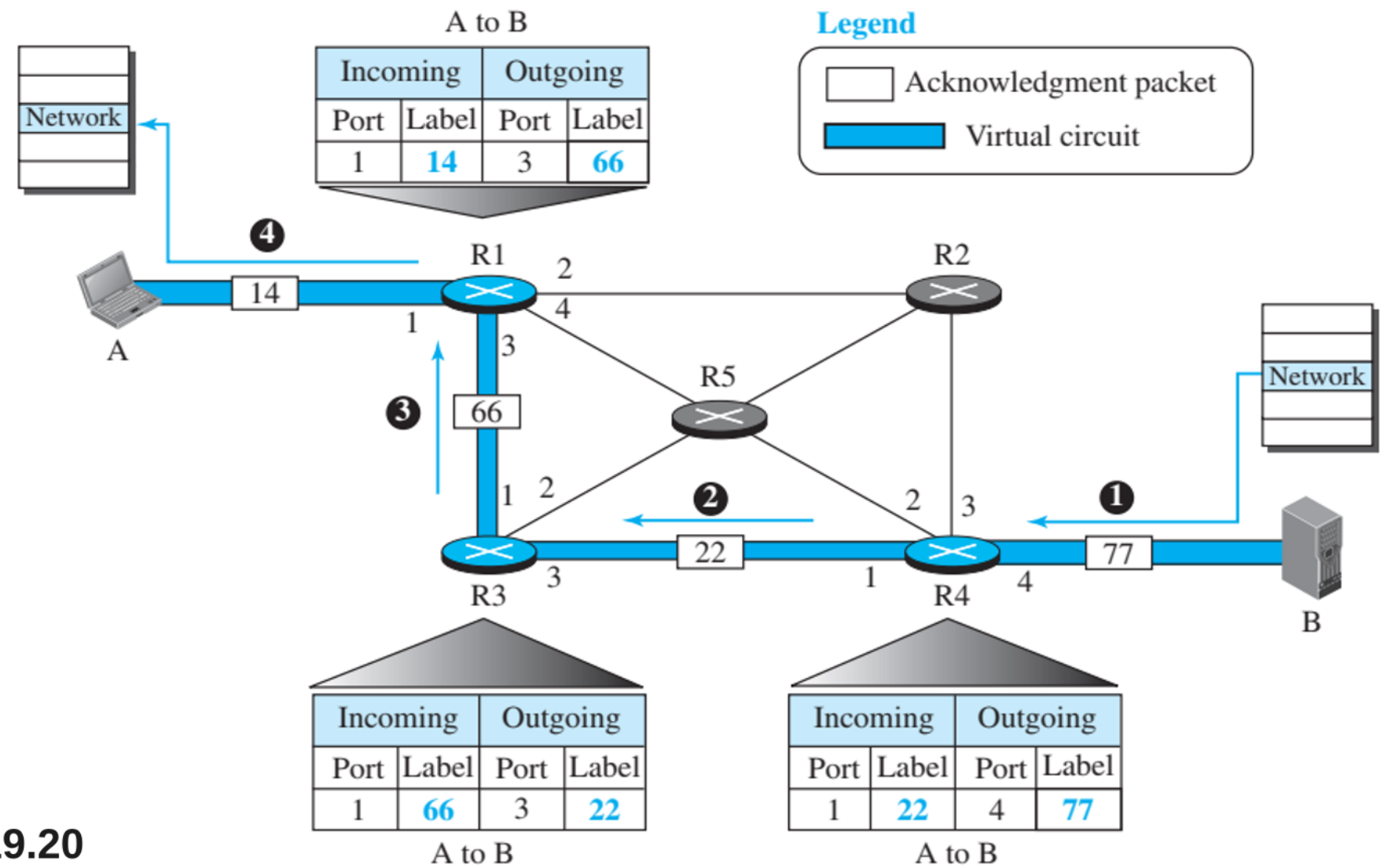
Connection-Oriented Service : Request PhaSe

4. **Router R4 receives the setup request packet. Again, three columns are completed: incoming port (1), incoming label (22), and outgoing port (4).**

5. **Destination B receives the setup packet, and if it is ready to receive packets from A, it assigns a label to the incoming packets that come from A, in this case 77, as shown in Figure 18.8. This label lets the destination know that the packets come from A, and not from other sources.**

Connection-Oriented Service : Acknowledgement phase:

Figure 18.8 *Sending acknowledgments in a virtual-circuit network*



Connection-Oriented Service : Acknowledgement phase:

1. The destination sends an acknowledgment to router R4. The acknowledgment carries the global source and destination addresses so the router knows which entry in the table is to be completed. The packet also carries label 77, chosen by the destination as the incoming label for packets from A. Router R4 uses this label to complete the outgoing label column for this entry. Note that 77 is the incoming label for destination B, but the outgoing label for router R4.
2. Router R4 sends an acknowledgment to router R3 that contains its incoming label in the table, chosen in the setup phase. Router R3 uses this as the outgoing label in the table.
3. Router R3 sends an acknowledgment to router R1 that contains its incoming label in the table, chosen in the setup phase. Router R1 uses this as the outgoing label in the table.
4. Finally router R1 sends an acknowledgment to source A that contains its incoming label in the table, chosen in the setup phase.
5. The source uses this as the outgoing label for the data packets to be sent to destination B

Connection-Oriented Service : Data Transfer phase:

Figure 18.9 *Flow of one packet in an established virtual circuit*

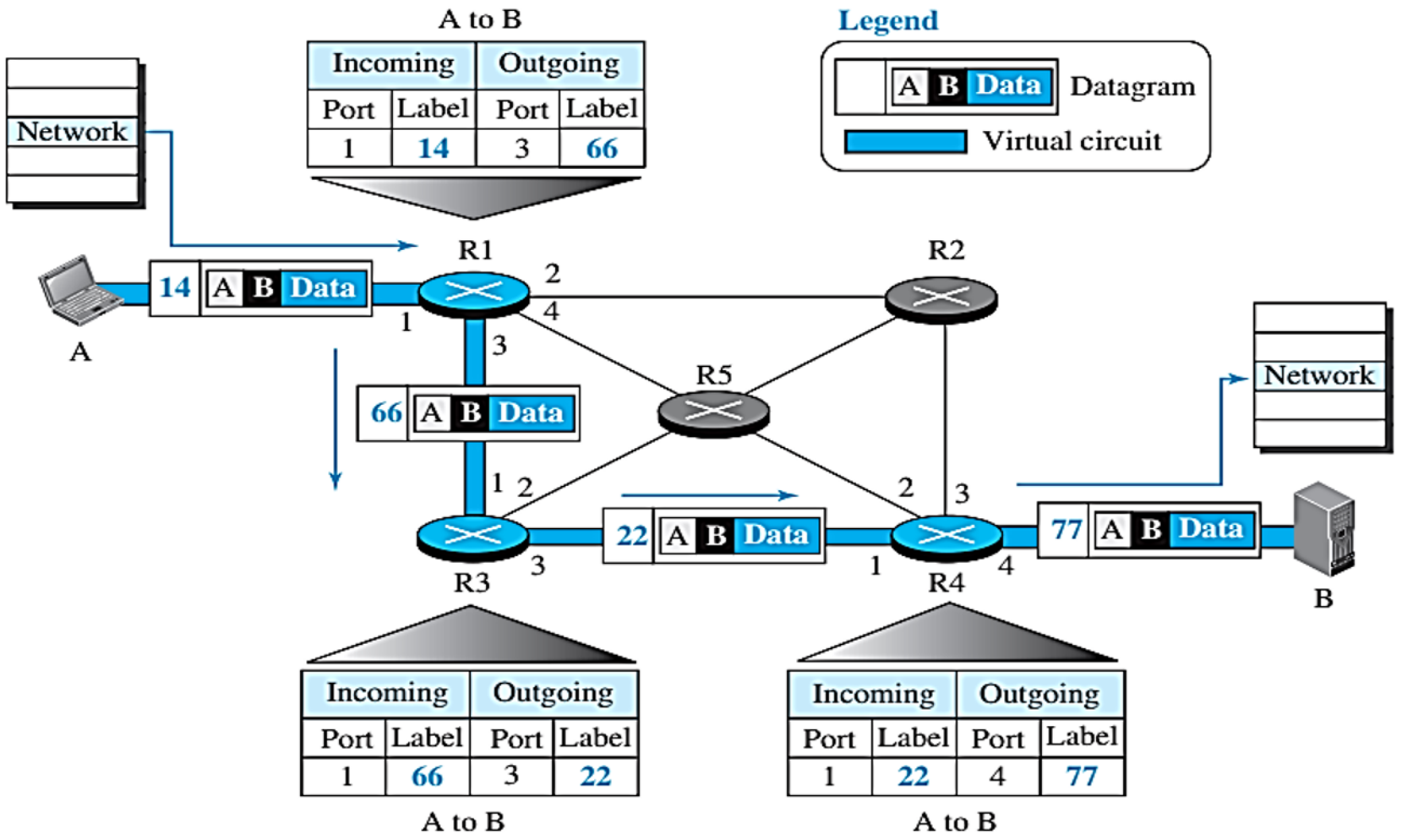
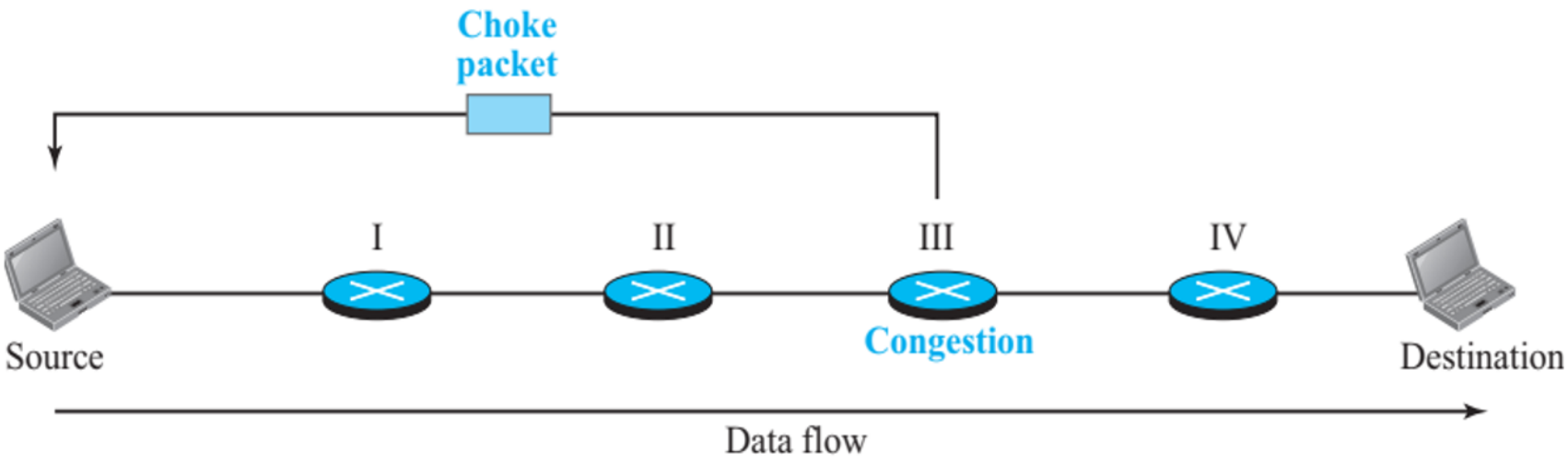


Figure 18.15 *Choke packet*



19-1 IPv4 ADDRESSES

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

Topics discussed in this section:

Address Space
Notations
Classful Addressing
Classless Addressing
Network Address Translation (NAT)

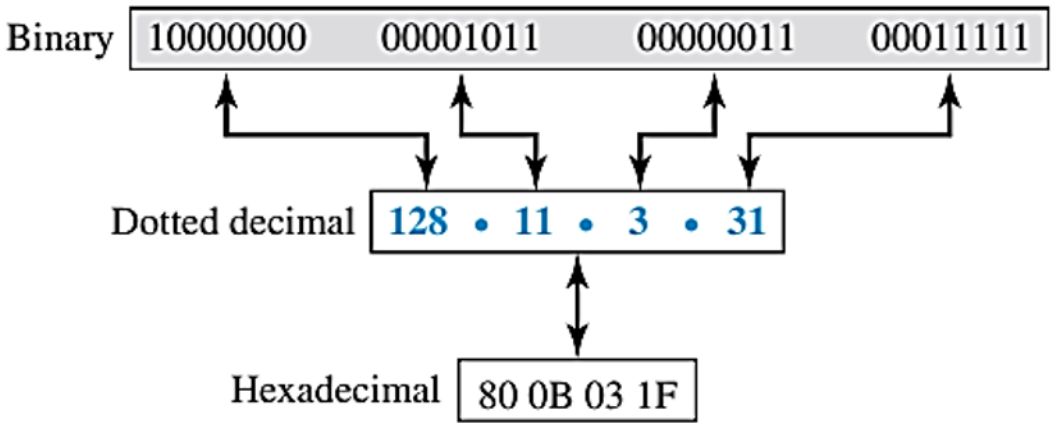
ADDRESS SPACE:

- An address space is the total number of addresses used by the protocol.
- If a protocol uses b bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than four billion).
- If there were no restrictions, more than 4 billion devices could be connected to the Internet

ADDRESS SPACE: *Notations*

There are three common notations to show an IPv4 address:

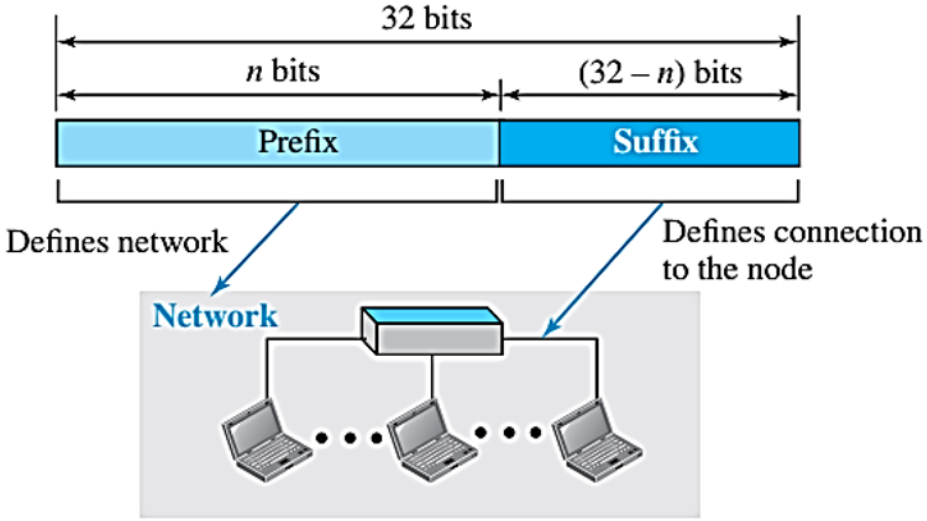
- **binary notation (base 2),**
- **dotted-decimal notation (base 256),**
- **hexadecimal notation (base 16)**



ADDRESS SPACE: *Hierarchy in Addressing*

- A 32-bit IPv4 address is also hierarchical, but divided only into two parts.
- The first part of the address, called the prefix, defines the network; the second part of the address, called the suffix, defines the node (connection of a device to the Internet).
- Figure 18.17 shows the prefix and suffix of a 32-bit IPv4 address. The prefix length is n bits and the suffix length is (32 - n) bits

Figure 18.17 *Hierarchy in addressing*



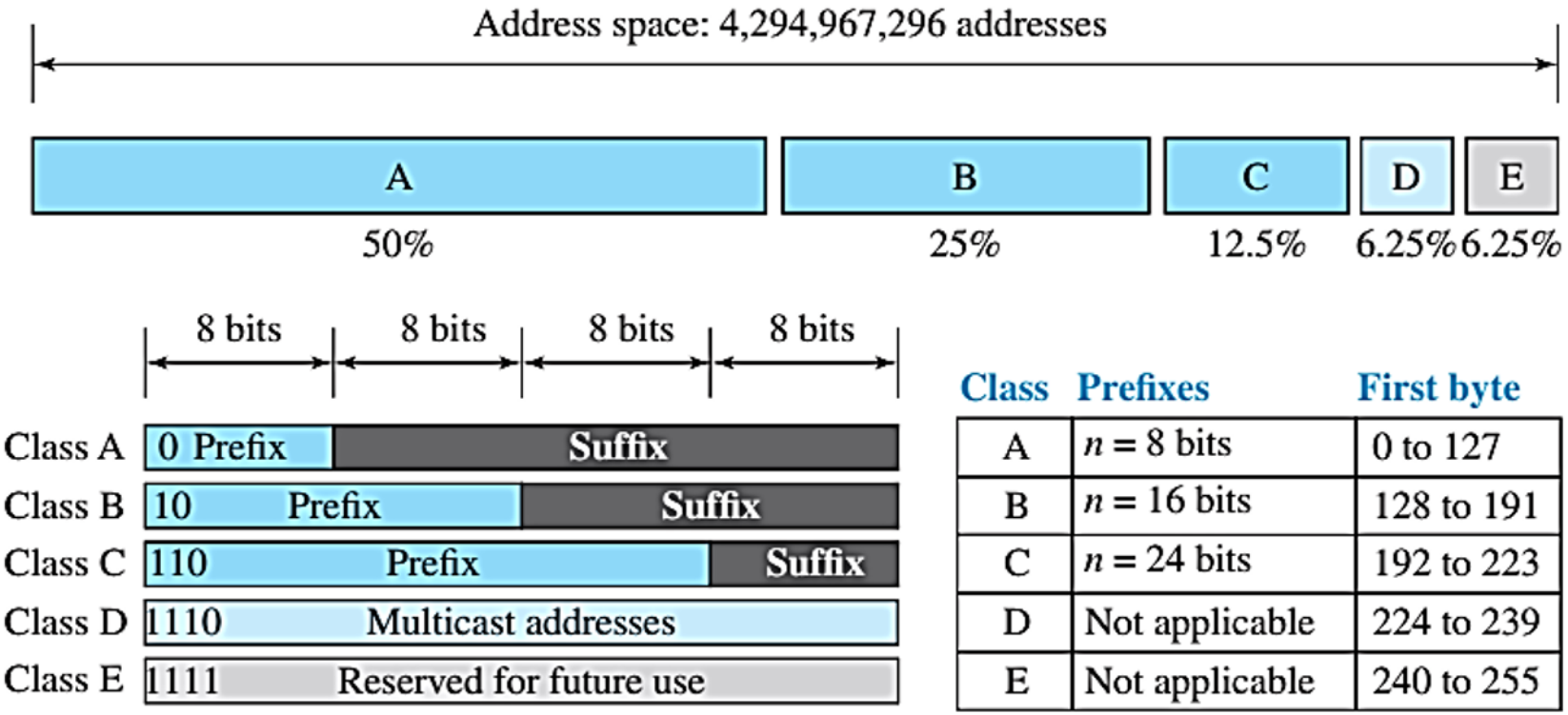
ADDRESS SPACE: *Classful Addressing*

- When the Internet started, an IPv4 address was designed with a fixed-length prefix,
- But to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24).
- The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure 18.18. This scheme is referred to as classful addressing.

  - In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier.
  - This means there are only 2^7 = 128 networks in the world that can have a class A address

ADDRESS SPACE: *Classful Addressing*

Figure 18.18 *Occupation of the address space in classful addressing*



Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

ADDRESS SPACE: *Classful Addressing: Address depletion*

- The reason that classful addressing has become obsolete is address depletion.
- Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet

- To understand the problem, let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network with 16,777,216 nodes (computers in this single network).
- Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused

To alleviate address depletion, two strategies were proposed and, to some extent, implemented:

Subnetting :

- **In Subnetting, for example, if a network in class A is divided into four subnets, each subnet has a prefix of nsub = 10.**
- **At the same time, if all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.**
- **This idea did not work because most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.**

Supernetting **was devised to combine several class C blocks into a larger block to be attractive to organizations that need more than the 256 addresses available in a class C block. This idea did not work either because it makes the routing of packets more difficult.**

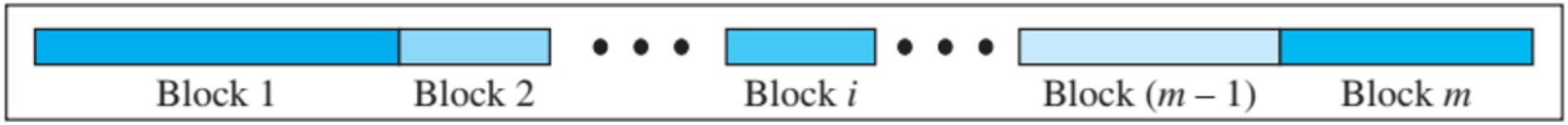
ADDRESS SPACE: *Classless Addressing:*

- With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution.
- The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed.
- The short-term solution still uses IPv4 addresses, but it is called classless addressing.
- In other words, the class privilege was removed from the distribution to compensate for the address depletion.

- In 1996, the Internet authorities announced a new architecture called classless addressing.
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.
- In classless addressing, the whole address space is divided into variable length blocks.

ADDRESS SPACE: *Classless Addressing:*

Figure 18.19 *Variable-length blocks in classless addressing*



Address space

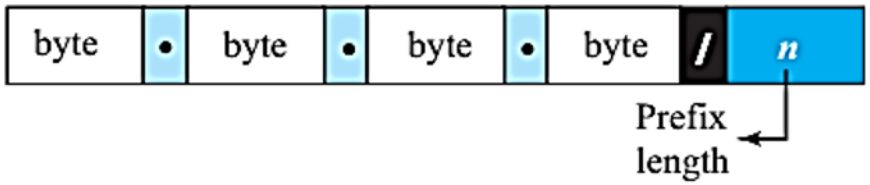
- Unlike classful addressing, the prefix length in classless addressing is variable.
- We can have a prefix length that ranges from 0 to 32. The size of the network is inversely proportional to the length of the prefix.
- A small prefix means a larger network; a large prefix means a smaller network.
- We need to emphasize that the idea of classless addressing can be easily applied to classful addressing.
- An address in class A can be thought of as a classless address in which the prefix length is 8. An address in class B can be thought of as a classless address in which the prefix is 16, and so on.

ADDRESS SPACE: *Prefix Length: Slash Notation:*

The first question that we need to answer in classless addressing is how to find the prefix length if an address is given.

- **Since the prefix length is not inherent in the address, we need to separately give the length of the prefix. In this case,**
- **The prefix length, n, is added to the address, separated by a slash.**
- **The notation is informally referred to as slash notation and formally as classless interdomain routing or CIDR**

Figure 18.20 *Slash notation (CIDR)*



Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

ADDRESS SPACE: *Prefix Length: Slash Notation:*

Extracting Information from an Address

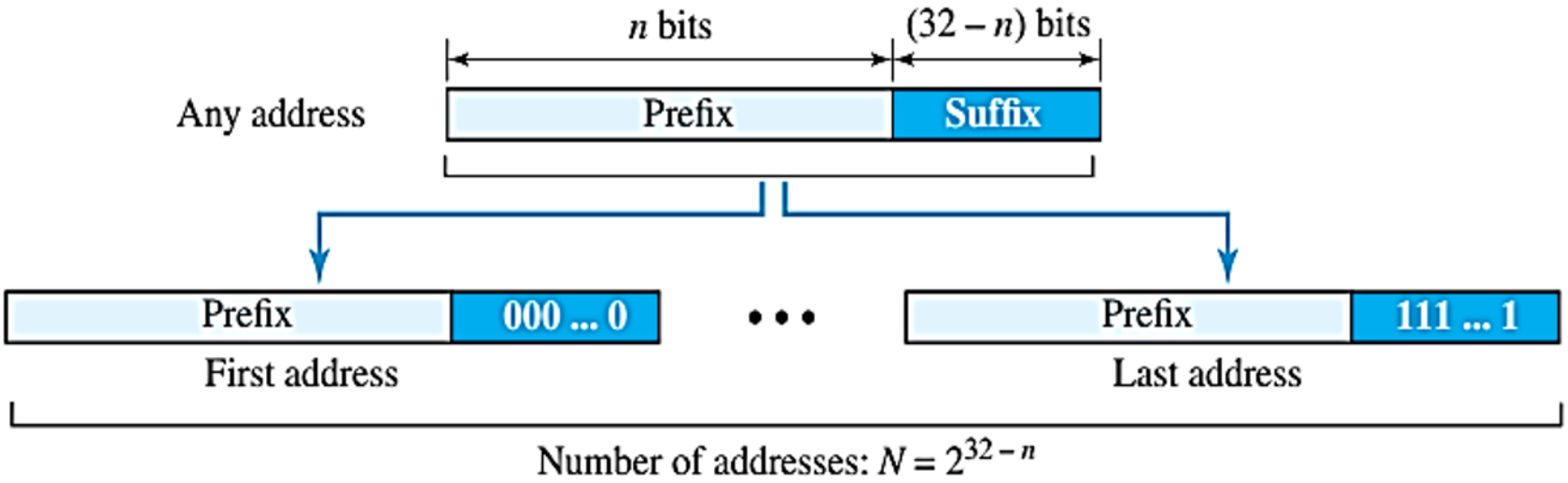
Given any address in the block, we normally like to know three pieces of information about the block to which the address belongs:

- **The number of addresses, the first address in the block, and the last address.**
- **Since the value of prefix length, n, is given, we can easily find these three pieces of information**

1. **The number of addresses in the block is found as $N = 2^{32-n}$.**
2. **To find the first address, we keep the *n* leftmost bits and set the (32 – *n*) rightmost bits all to 0s.**
3. **To find the last address, we keep the *n* leftmost bits and set the (32 – *n*) rightmost bits all to 1s.**

ADDRESS SPACE: *Prefix Length: Slash Notation:*

Figure 18.21 *Information extraction in classless addressing*



ADDRESS SPACE: *Prefix Length: Slash Notation:*

Example 18.1

A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows.

The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses.

The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

ADDRESS SPACE: *Address Mask:*

- Another way to find the first and last addresses in the block is to use the address mask.
- The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 − n) are set to 0s.
- A computer can easily find the address mask because it is the complement of $(2^{32-n} - 1)$.
- The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR

1. The number of addresses in the block N = NOT (mask) + 1.
2. The first address in the block = (Any address in the block) AND (mask).
3. The last address in the block = (Any address in the block) OR [(NOT (mask)

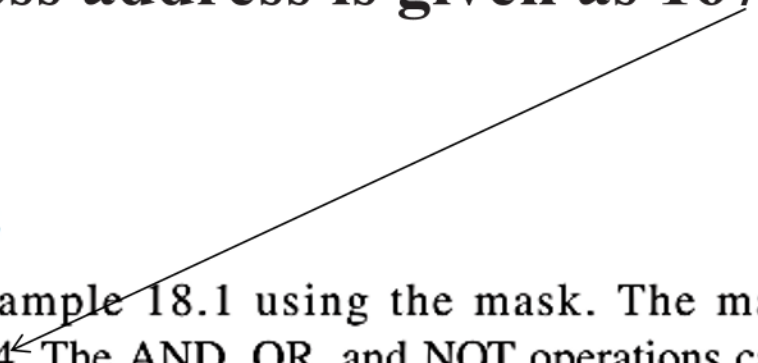
ADDRESS SPACE: *Address Mask:*

- Another way to find the first and last addresses in the block is to use the address mask.
- The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits (32 − n) are set to 0s.
- A computer can easily find the address mask because it is the complement of $(2^{32-n} - 1)$.
- The reason for defining a mask in this way is that it can be used by a computer program to extract the information in a block, using the three bit-wise operations NOT, AND, and OR

1. The number of addresses in the block N = NOT (mask) + 1.
2. The first address in the block = (Any address in the block) AND (mask).
3. The last address in the block = (Any address in the block) OR [(NOT (mask)

ADDRESS SPACE: *Address Mask:*

A classless address is given as 167.199.170.82/27.



Example 18.2

We repeat Example 18.1 using the mask. The mask in dotted-decimal notation is 256.256.256.224. The AND, OR, and NOT operations can be applied to individual bytes using calculators and applets at the book website.

Number of addresses in the block:	N = **NOT** (mask) + 1 = 0.0.0.31 + 1 = 32 addresses
First address:	First = (address) **AND** (mask) = **167.199.170.64**
Last address:	Last = (address) **OR** (**NOT** mask) = **167.199.170.95**

Note

In IPv4 addressing, a block of addresses can be defined as x.y.z.t /n in which x.y.z.t defines one of the addresses and the /n defines the mask.

Note

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

Example 19.6

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

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 0010000

or

205.16.37.32.

This is actually the block shown in Figure 19.3.

Note

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

Example 19.7

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is

11001101 00010000 00100101 00100111

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

11001101 00010000 00100101 00101111

or

205.16.37.47

This is actually the block shown in Figure 19.3.

Note

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

Example 19.8

Find the number of addresses in Example 19.6.

Solution

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

Example 19.9

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as

11111111 11111111 11111111 11110000

(twenty-eight 1s and four 0s).

Find

a. The first address
b. The last address
c. The number of addresses.

Example 19.9 (continued)

Solution

a. The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.

Address:	11001101	00010000	00100101	00100111
Mask:	**11111111**	**11111111**	**11111111**	**11110000**
First address:	11001101	00010000	00100101	00100000

Example 19.9 (continued)

b. **The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.**

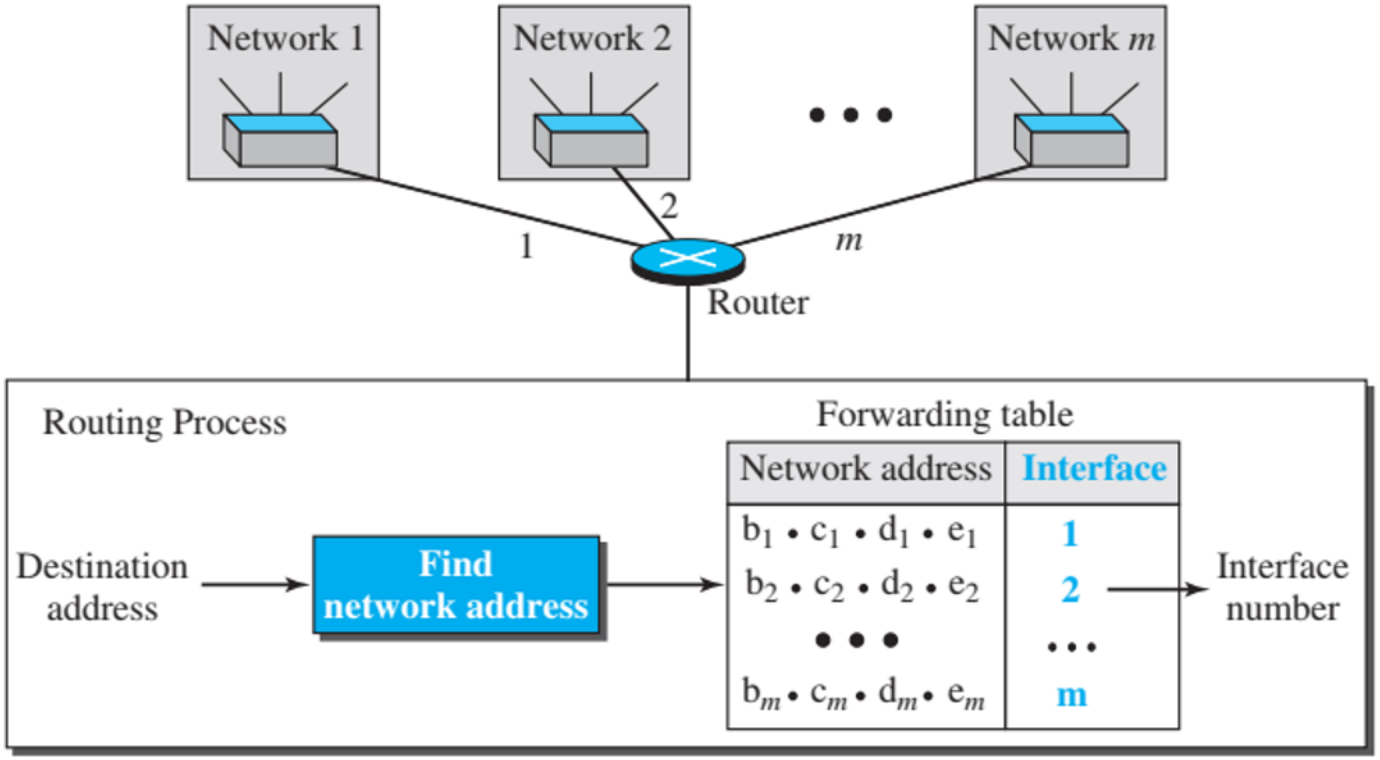
Address:	11001101	00010000	00100101	00100111
Mask complement:	**00000000**	**00000000**	**00000000**	**00001111**
Last address:	11001101	00010000	00100101	00101111

ADDRESS SPACE: *Network Address:*

- The first address, the network address, is particularly important because it is used in routing a packet to its destination network.
- For the moment, let us assume that an internet is made of m networks and a router with m interfaces.
- When a packet arrives at the router from any source host, the router needs to know to which network the packet should be sent: from which interface the packet should be sent out.
- When the packet arrives at the network, it reaches its destination host using another strategy
- After the network address has been found, the router consult its forwarding table to find the corresponding interface from which the packet should be sent out.
- The network address is actually the identifier of the network; each network is identified by its network address.

ADDRESS SPACE: *Network Address:*

Figure 18.22 *Network address*

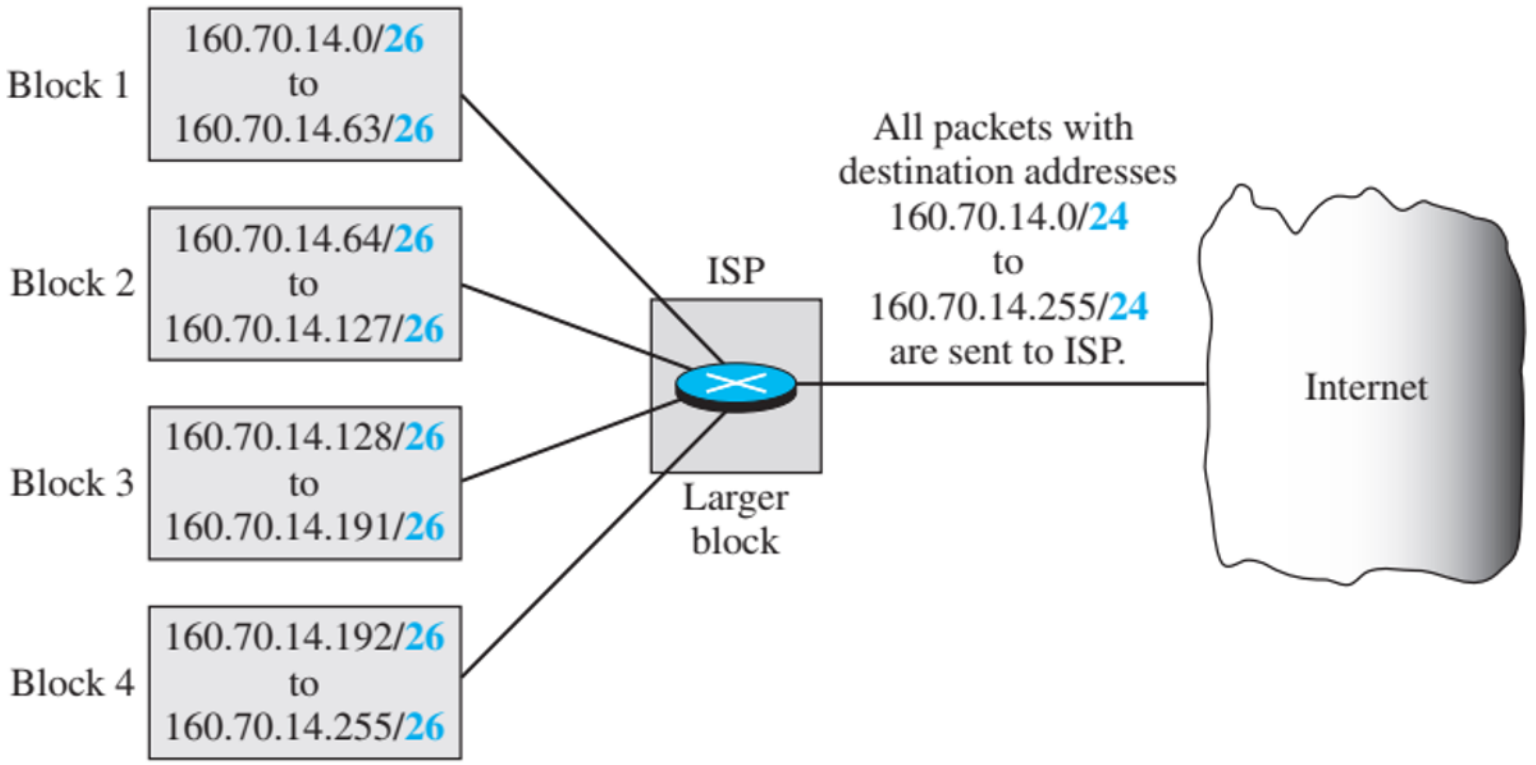


IPv4 ADDRESS AGGREGATION:

- One of the advantages of the CIDR (Classless Inter-Domain Routing) strategy is address aggregation (sometimes called address summarization or route summarization).
- When blocks of addresses are combined to create a larger block, routing can be done based on the prefix of the larger block.
- ICANN (Internet Corporation for Assigned Names and Numbers) 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.

IPv4 ADDRESS AGGREGATION:

Figure 18.24 *Example of address aggregation*



IPv4: SPECIAL ADDRESSES

This-host Address

The only address in the block **0.0.0.0/32** is called the ***this-host* address**. It is used whenever a host needs to send an IP datagram but it does not know its own address to use as the source address.

Limited-broadcast Address

The only address in the block **255.255.255.255/32** is called the *limited-broadcast* address. It is used whenever a router or a host needs to send a datagram to all devices in a network.

Loopback Address

The block **127.0.0.0/8** is called the *loopback* address. A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host.

Private Addresses

Four blocks are assigned as private addresses: **10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, and 169.254.0.0/16**

Multicast Addresses

The block 224.0.0.0/**4** is reserved for multicast addresses

19-2 IPv6 ADDRESSES

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

Topics discussed in this section:

Structure
Address Space

IPv6:

- ***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 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.

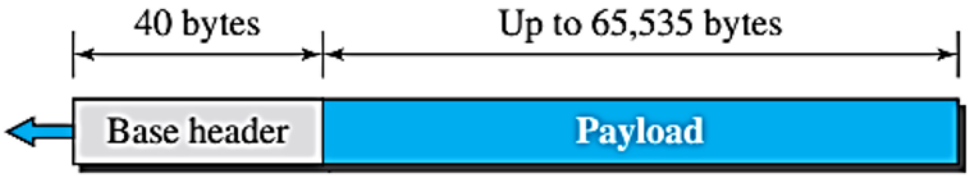
- ***Allowance for extension.*** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

- ***Support for resource allocation.*** In IPv6, the type-of-service field has been removed, but two new fields, traffic class and flow label, have 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.

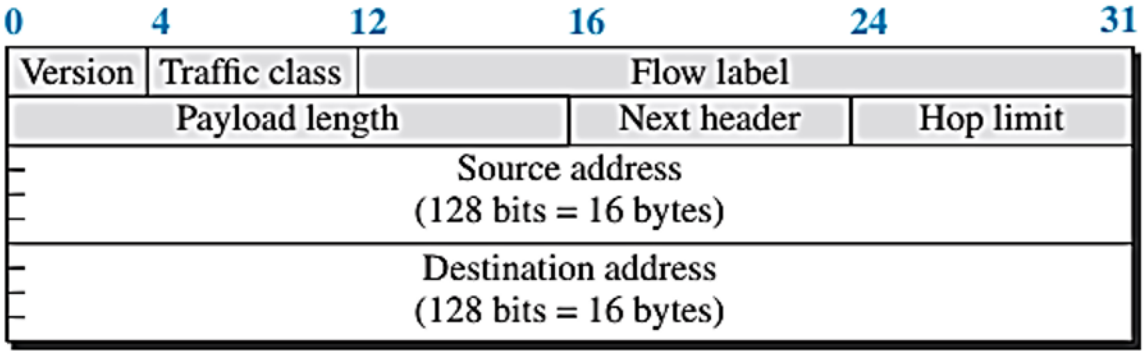
- ***Support for more security.*** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet

IPv6: Packet Format

Figure 22.6 *IPv6 datagram*



a. IPv6 packet



b. Base header

IPv6: Packet Format

- **Version.** The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.

- **Traffic class.** The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements. It replaces the type-of-service field in IPv4.

- **Flow label.** The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data. We will discuss this field later.

- **Payload length.** The 2-byte payload length field defines the length of the IP datagram excluding the header.
  - Note that IPv4 defines two fields related to the length: header length and total length.
  - In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined.

IPv6: Packet Format

- **Next header.** The next header is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram. This field is similar to the protocol field in IPv4, but we talk more about it when we discuss the payload.

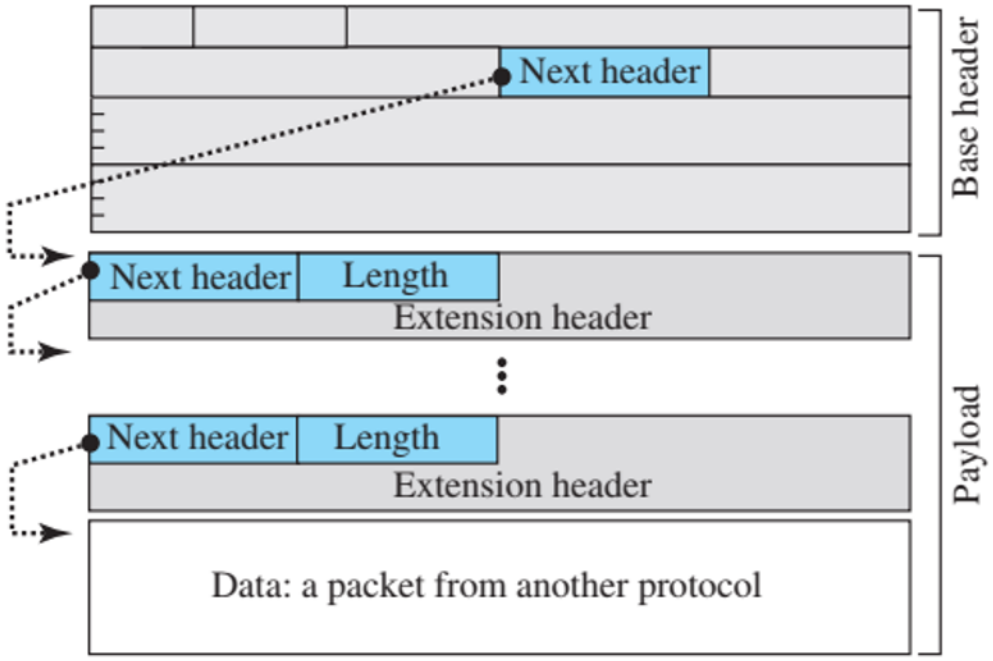
- **Hop limit.** The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.

- **Source and destination addresses.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destinationaddress field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.

- **Payload.** Compared to IPv4, the payload field in IPv6 has a different format and meaning, as shown in Figure 22.7..

IPv6: Packet Format

Figure 22.7 *Payload in an IPv6 datagram*



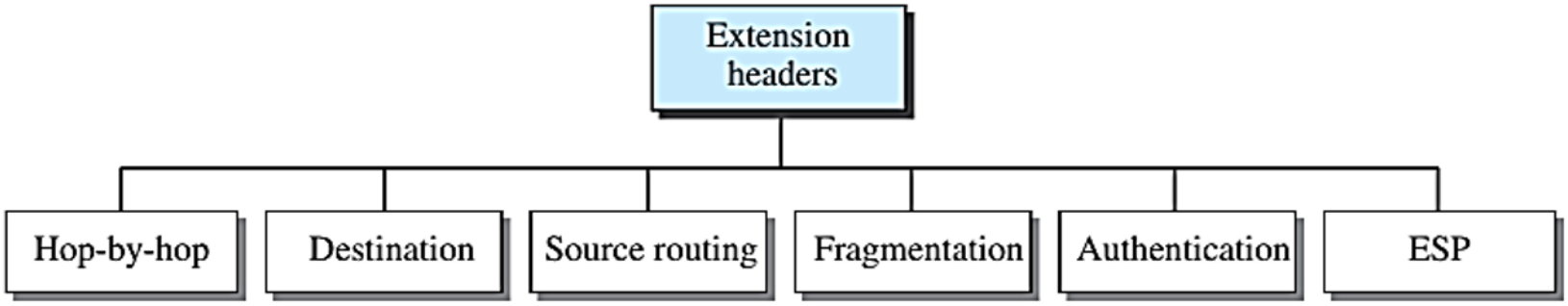
Some next-header codes

00: Hop-by-hop option
02: ICMPv6
06: TCP
17: UDP
43: Source-routing option
44: Fragmentation option
50: Encrypted security payload
51:Authentication header
59: Null (no next header)
60: Destination option

IPv6: Extension Header

- An IPv6 packet is made of a base header and some extension headers.
- The length of the base header is fixed at 40 bytes. However, to give more functionality to the IP datagram, the base header can be followed by up to six extension headers.

Figure 22.8 *Extension header types*



IPv6: Extension Header

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.
- For example, perhaps routers must be informed about certain management, debugging, or control functions.
- So far, only three hopby-hop options have been defined: Pad1, PadN, and jumbo payload.

- **Pad1**. This option is 1 byte long and is designed for alignment purposes.
- **PadN**. PadN is similar in concept to Pad1. The difference is that PadN is used when 2 or more bytes are needed for alignment.
- **Jumbo payload**. Recall that the length of the payload in the IP datagram can be a maximum of 65,535 bytes. However, if for any reason a longer payload is required, we can use the jumbo payload option to define this longer length

IPv6: Extension Header

- **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

- **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 in IPv6 is the same as that in IPv4. In IPv4, the source or a router is required to fragment In IPv6, only the original source can fragment.

- **Authentication**

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

- **Encrypted Security Payload**

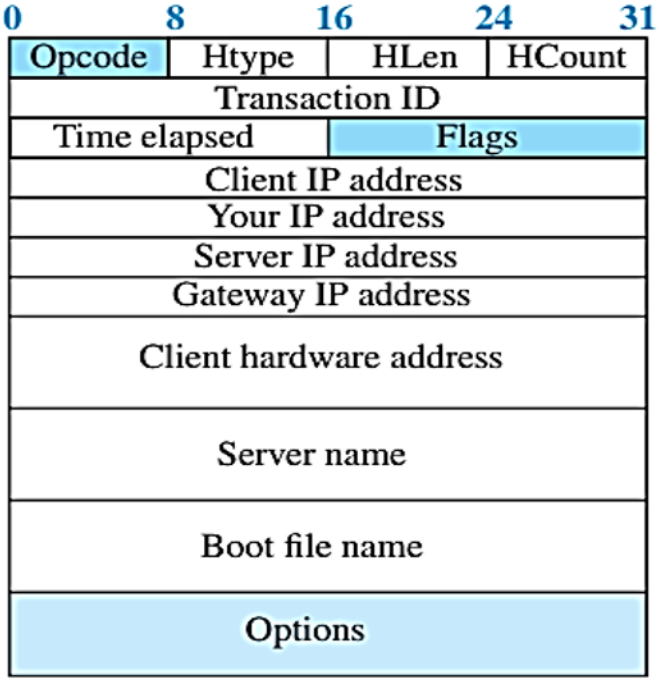
The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping

Dynamic Host Control Protocol : DHCP

- After a block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers.
- However, address assignment in an organization can be done automatically using the Dynamic Host Configuration Protocol (DHCP).
- DHCP has found such widespread use in the Internet that it is often called a plug and-play protocol.
- In can be used in many situations. A network manager can configure DHCP to assign permanent IP addresses to the host and routers.
- DHCP can also be configured to provide temporary, on demand, IP addresses to hosts.
- The second capability can provide a temporary IP address to a traveller to connect her laptop to the Internet while she is staying in the hotel

DHCP: Message Format

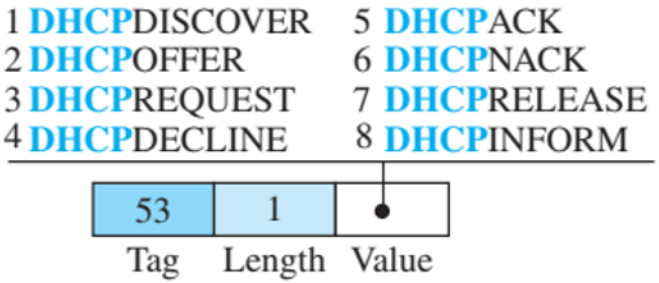
Figure 18.25 *DHCP message format*



Fields:

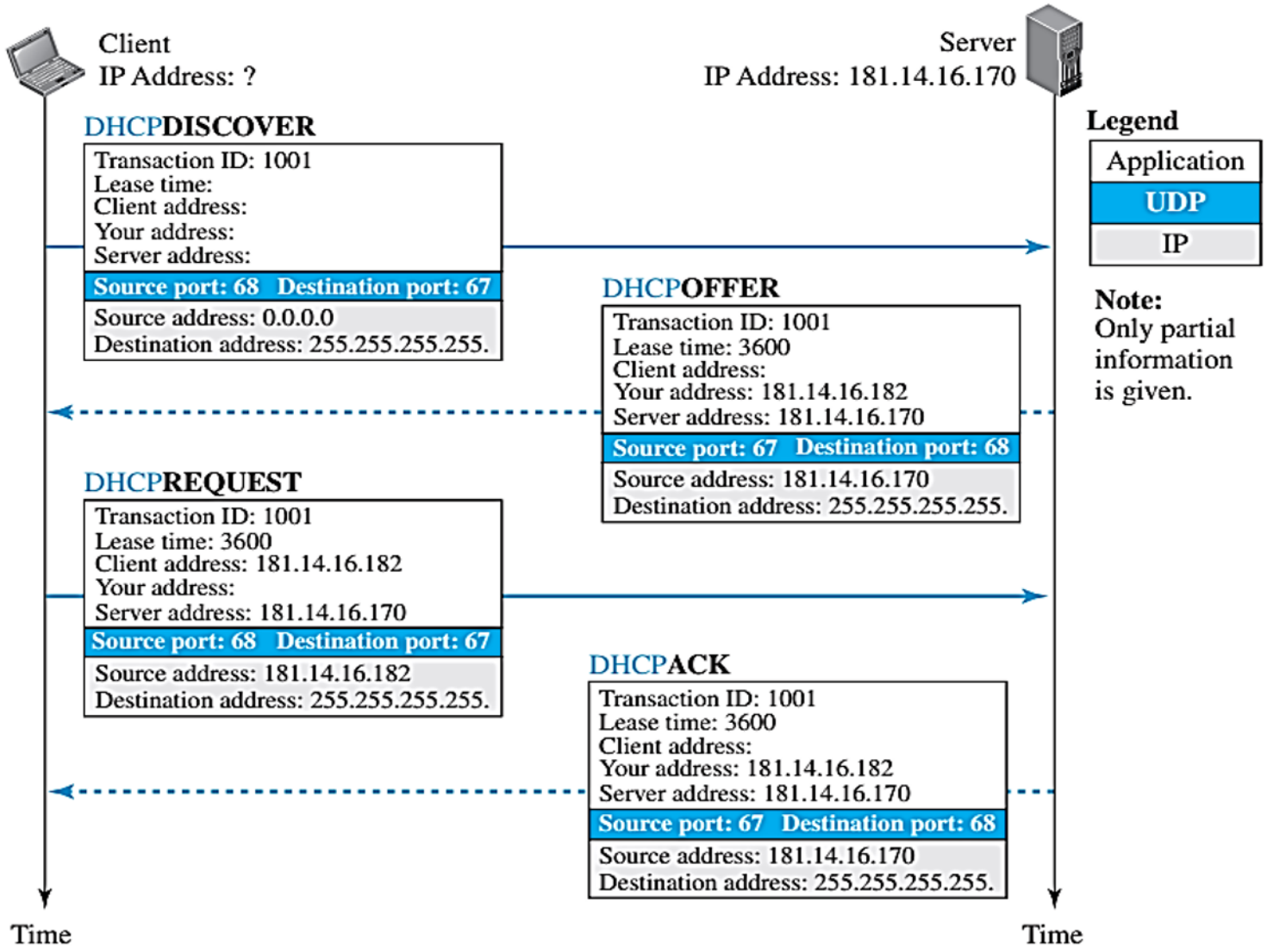
Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Length of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by the client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

Figure 18.26 *Option format*



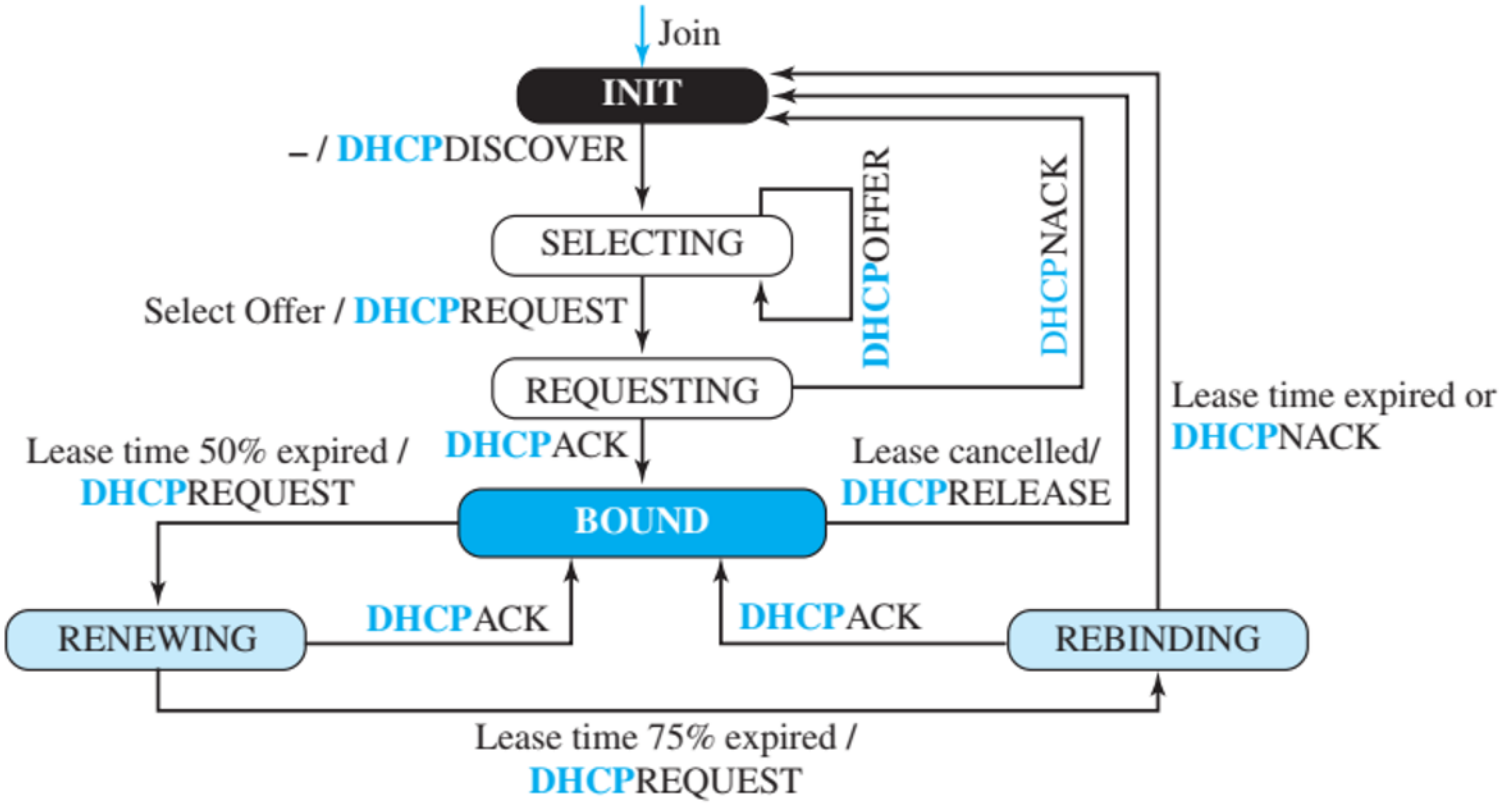
DHCP: Operation

Figure 18.27 *Operation of DHCP*



DHCP: FSM

Figure 18.28 *FSM for the DHCP client*



DHCP: FSM

- When the DHCP client first starts, it is in the INIT state (initializing state).
- The client broadcasts a discover message. When it receives an offer, the client goes to the SELECTING state. While it is there, it may receive more offers.
- After it selects an offer, it sends a request message and goes to the REQUESTING state.
- If an ACK arrives while the client is in this state, it goes to the BOUND state and uses the IP address.
- When the lease is 50 percent expired, the client tries to renew it by moving to the RENEWING state.
- If the server renews the lease, the client moves to the BOUND state again.

DHCP: FSM

- If the lease is not renewed and the lease time is 75 percent expired, the client moves to the REBINDING state.
- If the server agrees with the lease (ACK message arrives), the client moves to the BOUND state and continues using the IP address; otherwise, the client moves to the INIT state and requests another IP address.
- Note that the client can use the IP address only when it is in the BOUND, RENEWING, or REBINDING state.
- The above procedure requires that the client uses three timers: renewal timer (set to 50 percent of the lease time), rebinding timer (set to 75 percent of the lease time), and expiration timer (set to the lease time).

Network Address Translation : NTA

Problem:

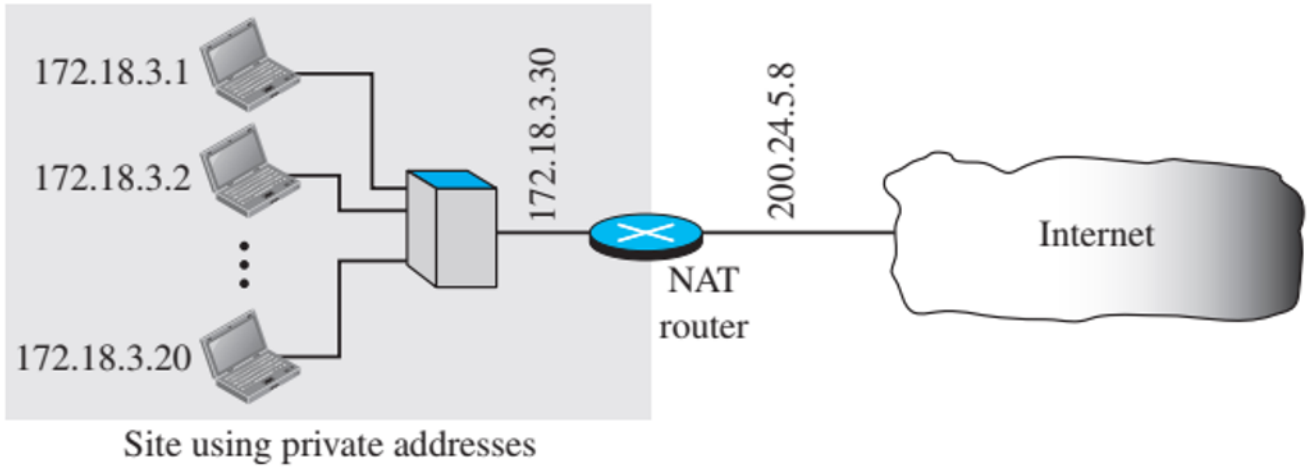
- The distribution of addresses through ISPs has created a new problem.
- Assume that an ISP has granted a small range of addresses to a small business or a household.
- If the business grows or the household needs a larger range, the ISP may not be able to grant the demand because the addresses before and after the range may have already been allocated to other networks.

Solution:

- A technology that can provide the mapping between the private and universal addresses, and at the same time support virtual private networks, is Network Address Translation (NAT).
- The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.

Network Address Translation :

Figure 18.29 *NAT*

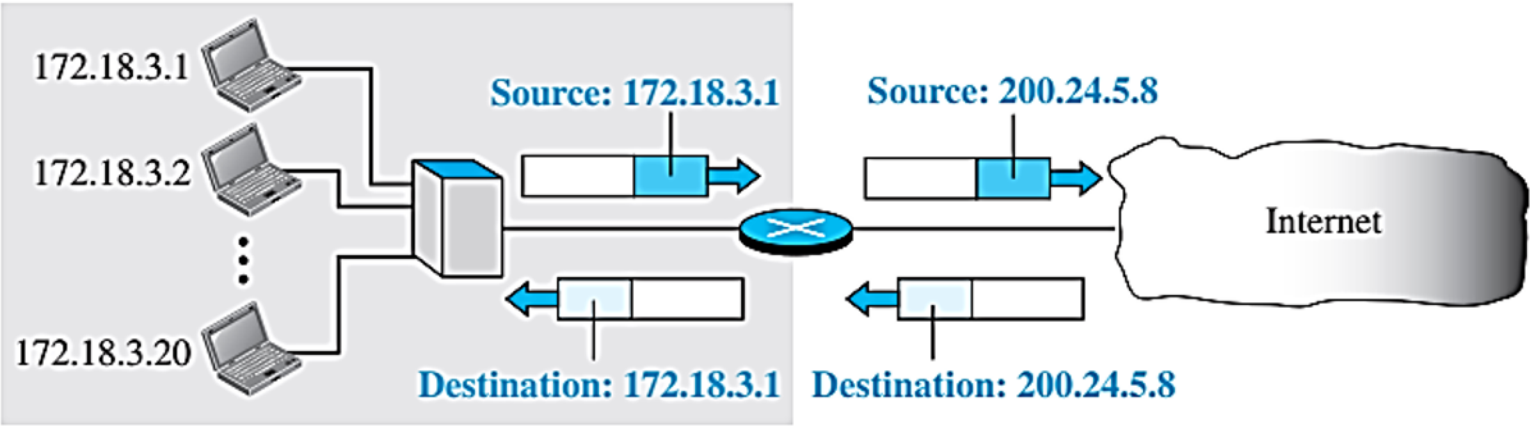


- As the figure shows, the private network uses private addresses.
- The router that connects the network to the global address uses one private address and one global address.
- The private network is invisible to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

NAT: Address Translation

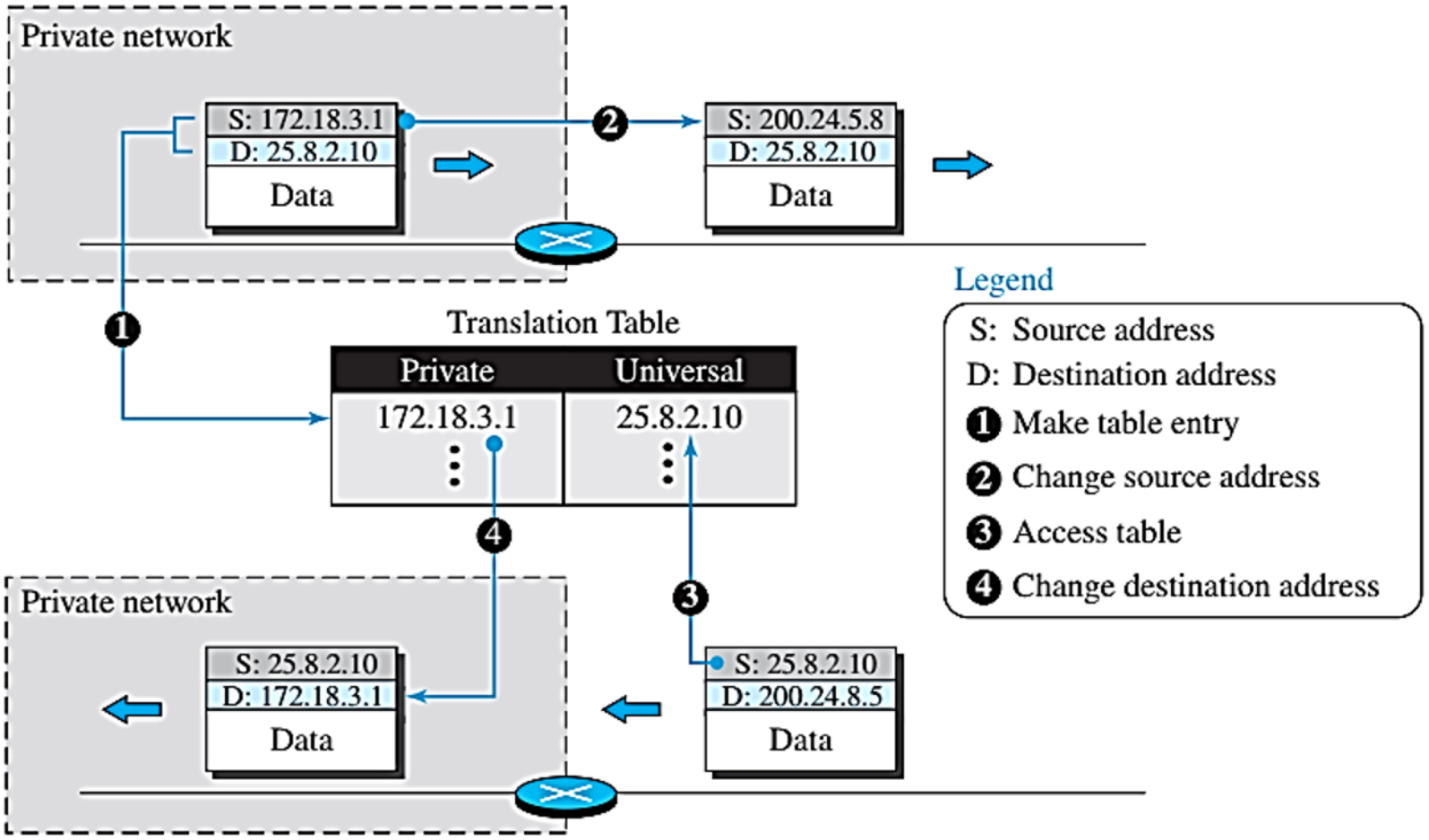
- All of the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private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

Figure 18.30 *Address translation*



NAT: Translation

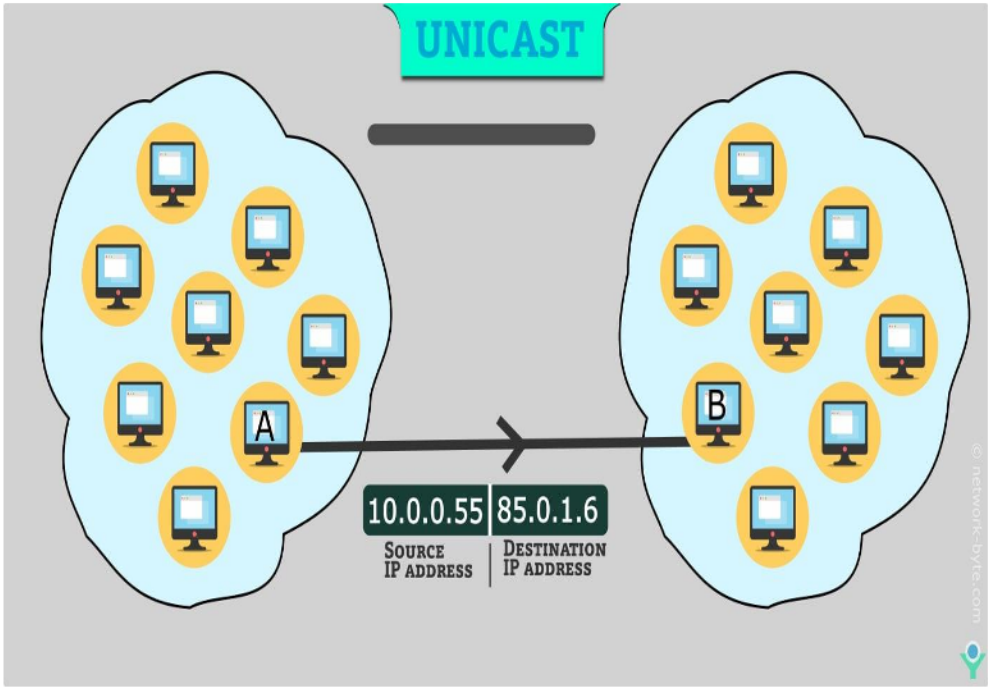
Figure 18.31 *Translation*

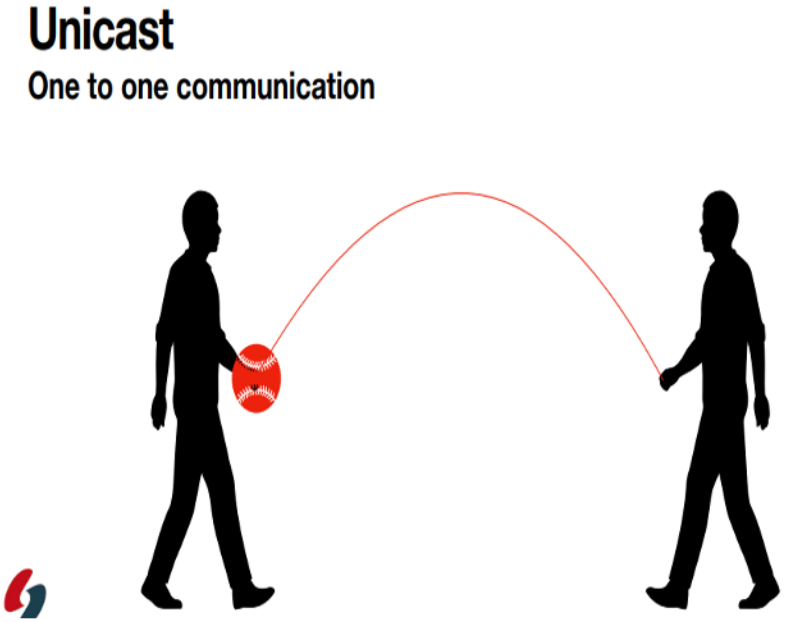


Chapter 21
Unicast Routing

Unicast Routing

- **The prefix "Uni" means single where data is sent from one sender to one receiver. It is also known as one-to-one transmission.**





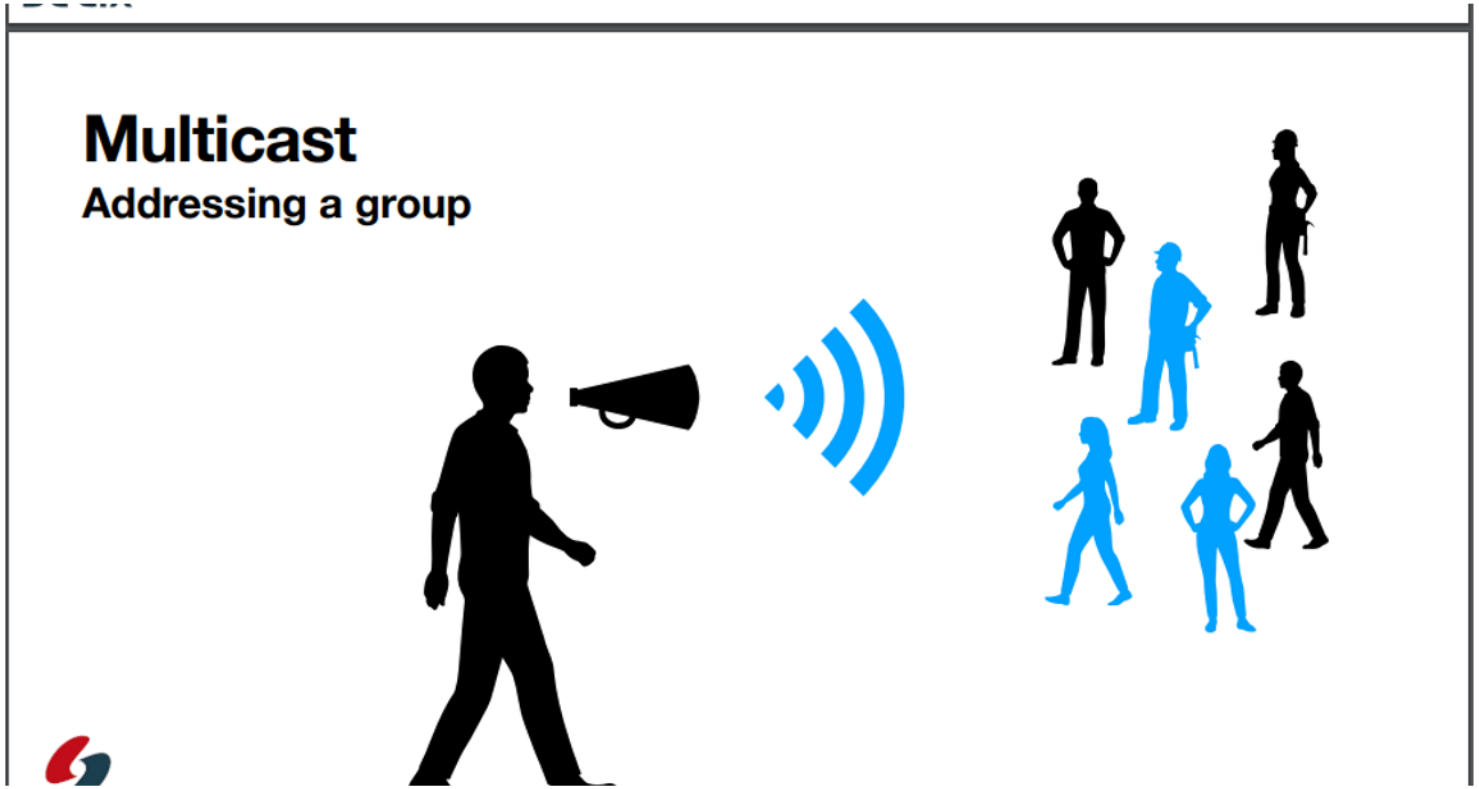
- **Here in the above figure Host, A having IP address 10.0.0.55 want to send data to Host B having IP address 85.0.1.6 then this connection established is known as Unicast.**
- In unicast both sender and receiver have a unique IP address.

Broadcast Routing

The broadcast routing method considers the communication through a network that involves a single sender (source) and multiple receivers (destinations). By default, the broadcast receivers are every device connected to the same network as the sender.

Multicast Routing

Multicasting route messages for a specific group of devices in a network. Note that, even if a group contains all the devices in a network, multicast is theoretically different from the broadcast. This difference consists that, in the multicast case, devices effectively subscribe to receive messages. In the broadcast case, however, devices receive messages regardless of whether or not they want to

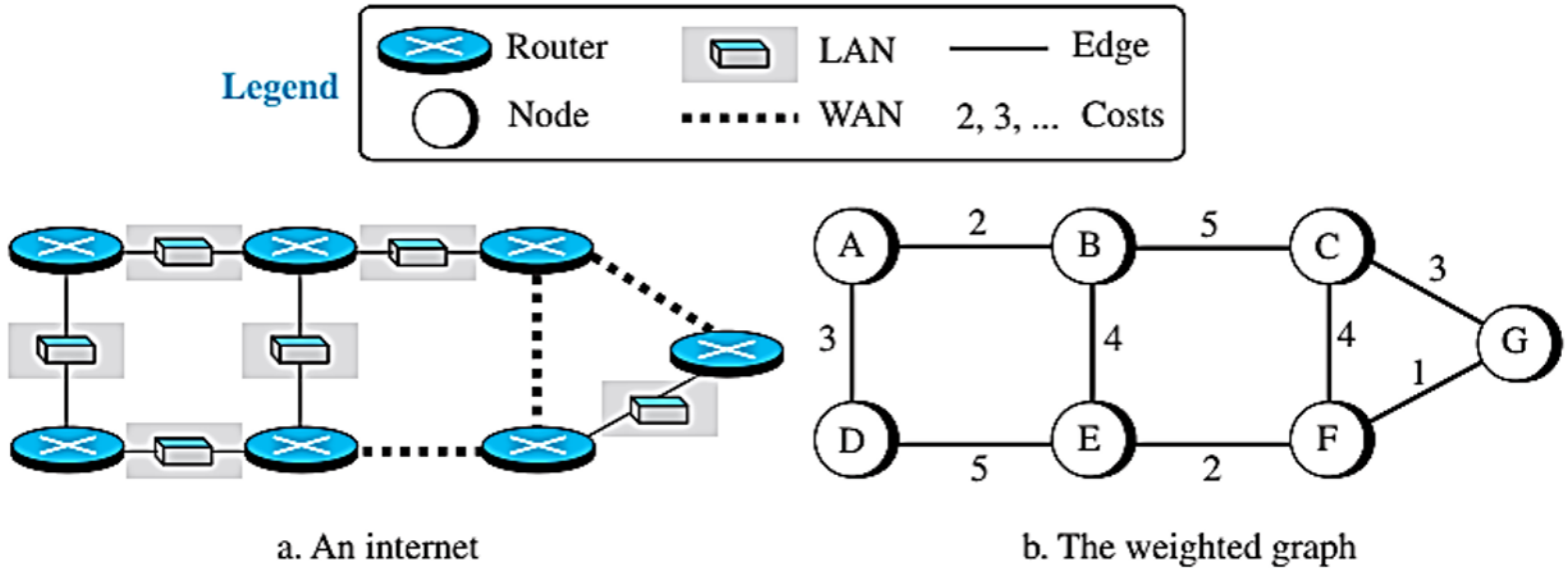


Unicast Routing : Introduction

- Unicast routing in the Internet, with a large number of routers and a huge number of hosts, can be done only by using hierarchical routing: routing in several steps using different routing algorithms.
- In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables.
- The source host needs no forwarding table because it delivers its packet to the default router in its local network.
- To find the best route, an internet can be modeled as a graph.
- To model an internet as a graph, we can think of each router as a node and each network between a pair of routers as an edge.
- An internet is, in fact, modeled as a weighted graph, in which each edge is associated with a cost.

Unicast Routing : Least Cost Routing (LCR)

- One of the ways to interpret the best route from the source router to the destination router is to find the least cost between the two.
- In other words, the source router chooses a route to the destination router in such a way that the total cost for the route is the least cost among all possible routes.
- In Figure 20.1, the best route between A and E is A-B-E,

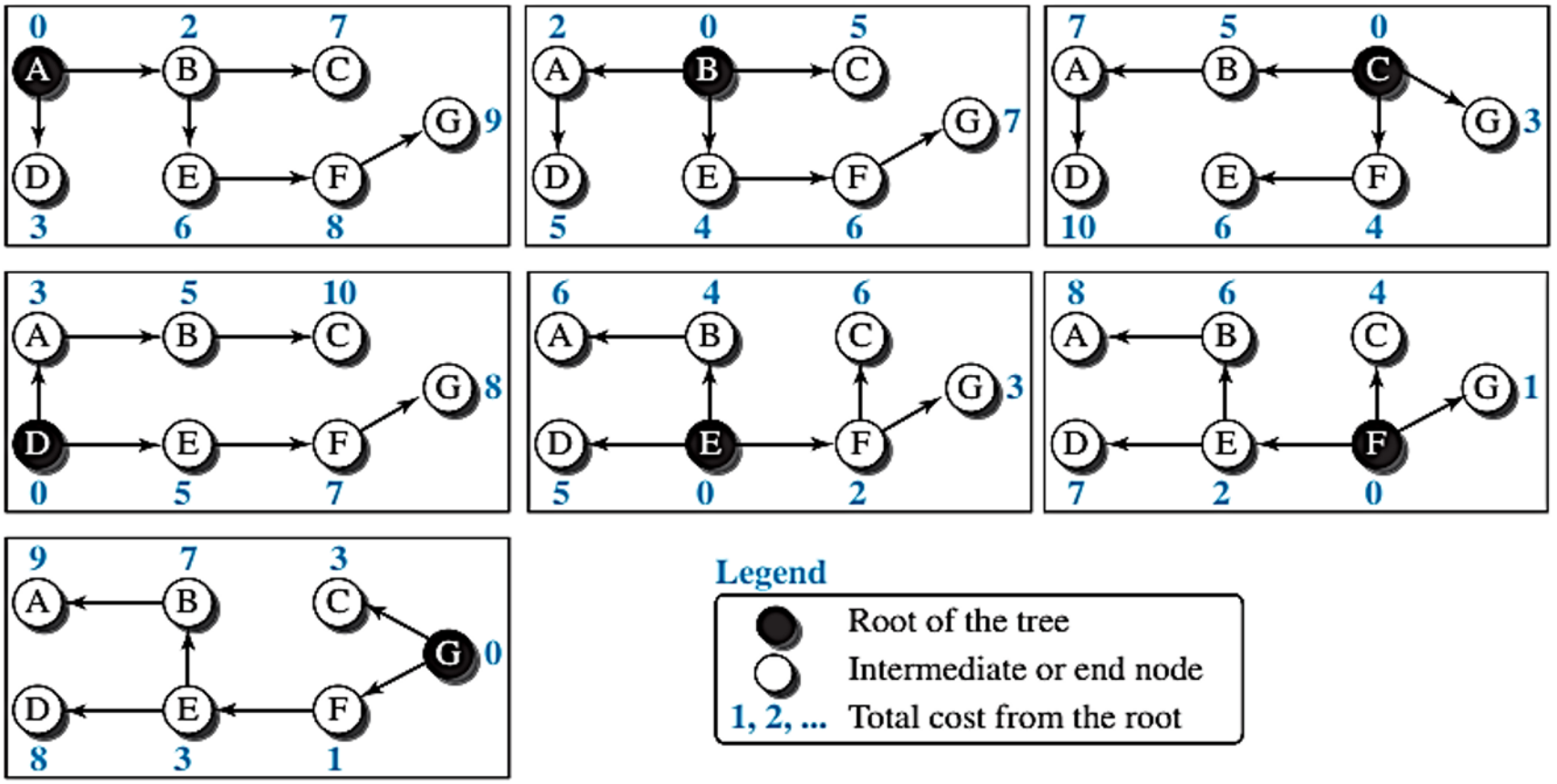


a. An internet

b. The weighted graph

Unicast Routing : Least Cost Routing (LCR)

Figure 20.2 *Least-cost trees for nodes in the internet of Figure 20.1*



Unicast Routing : Routing Algorithms

Distance-Vector Routing :

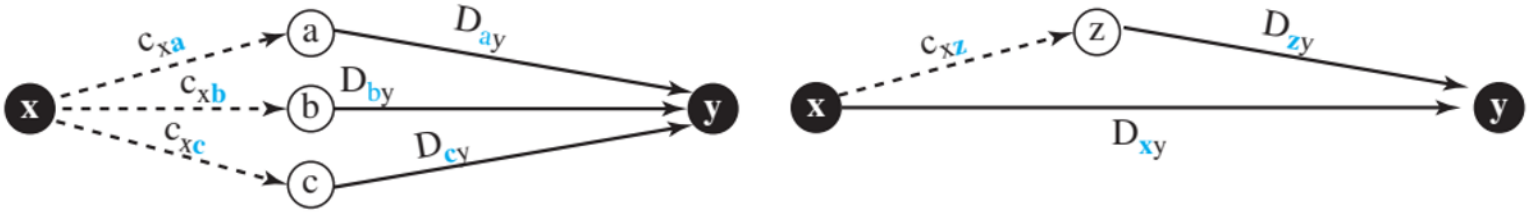
- The distance-vector (DV) routing is to find the best route.
- In distance-vector routing, the first thing each node creates is its own least-cost tree with the rudimentary information it has about its immediate neighbors.
- The incomplete trees are exchanged between immediate neighbors to make the trees more and more complete and to represent the whole internet.
- We can say that in distance-vector routing, a router continuously tells all of its neighbors what it knows about the whole internet (although the knowledge can be incomplete).

- *Bellman-Ford Equation*
- *Distance vector Routing*

Unicast Routing :Bellman Ford

- The heart of distance-vector routing is the famous Bellman-Ford equation.
- This equation is used to find the least cost (shortest distance) between a source node, x, and a destination node, y, through some intermediary nodes (a, b, c, . . .) when the costs between the source and the intermediary nodes and the least costs between the intermediary nodes and the destination are given.

Figure 20.3 *Graphical idea behind Bellman-Ford equation*



a. General case with three intermediate nodes

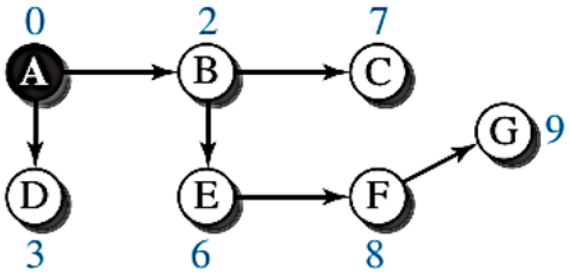
b. Updating a path with a new route

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

Unicast Routing :Distance Vectors

- ✓ A least-cost tree is a combination of least-cost paths from the root of the tree to all destinations.
- ✓ These paths are graphically glued together to form the tree. Distance-vector routing unglues these paths and creates a distance vector, a one-dimensional array to represent the tree.

Figure 20.4 *The distance vector corresponding to a tree*



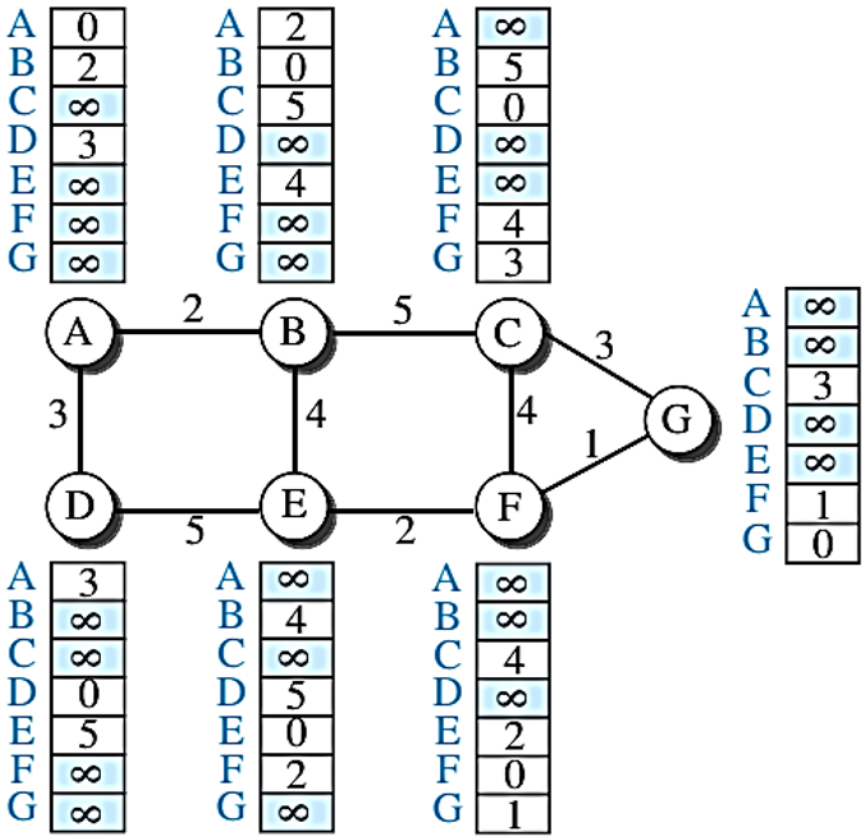
a. Tree for node A

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

b. Distance vector for node A

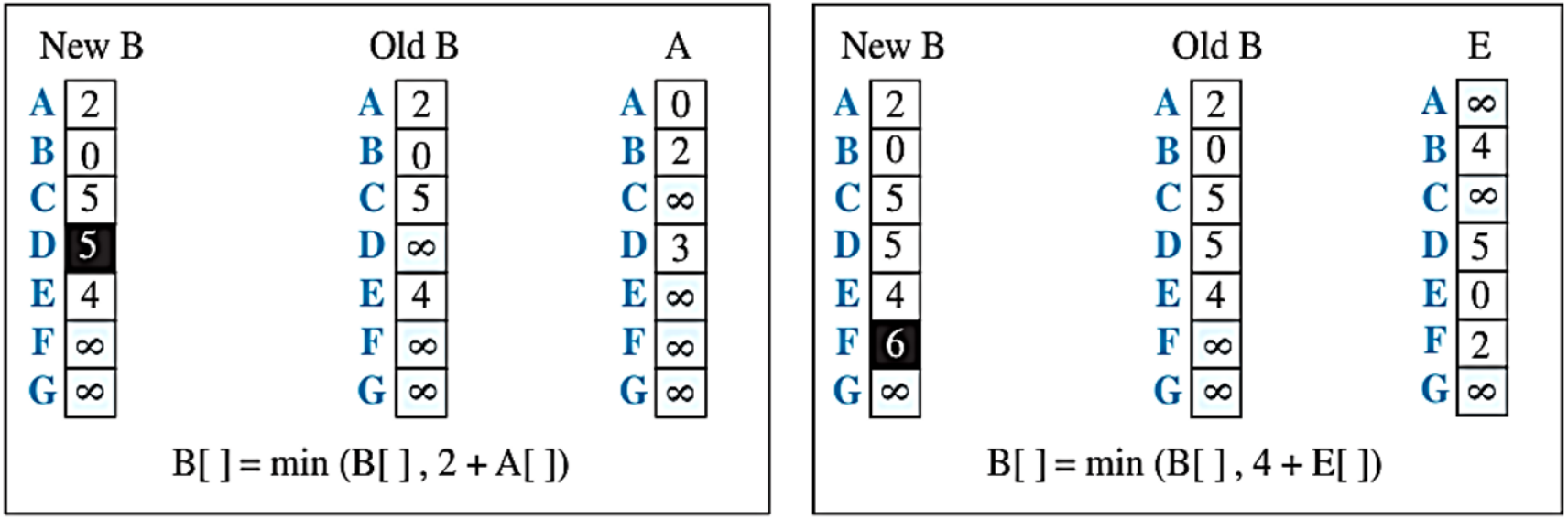
Unicast Routing :Distance Vectors

Figure 20.5 *The first distance vector for an internet*



Unicast Routing :Distance Vectors

Figure 20.6 *Updating distance vectors*



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

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

Note:
X[ ]: the whole vector

Unicast Routing :Distance Vector Routing Algo

```
1   Distance_Vector_Routing ( )
2   {
3       // Initialize (create initial vectors for the node)
4       D[myself] = 0
5       for (y = 1 to N)
6       {
7           if (y is a neighbor)
8               D[y] = c[myself][y]
9           else
10              D[y] = ∞
11      }
12      send vector {D[1], D[2], ..., D[N]} to all neighbors
13      // Update (improve the vector with the vector received from a neighbor)
14      repeat (forever)
15      {
16          wait (for a vector D_w from a neighbor w or any change in the link)
17          for (y = 1 to N)
18          {
19              D[y] = min [D[y], (c[myself][w] + D_w[y])]      // Bellman-Ford equation
20          }
21          if (any change in the vector)
22              send vector {D[1], D[2], ..., D[N]} to all neighbors
23      }
24  } // End of Distance Vector
```

Distance Vector Routing : Problems

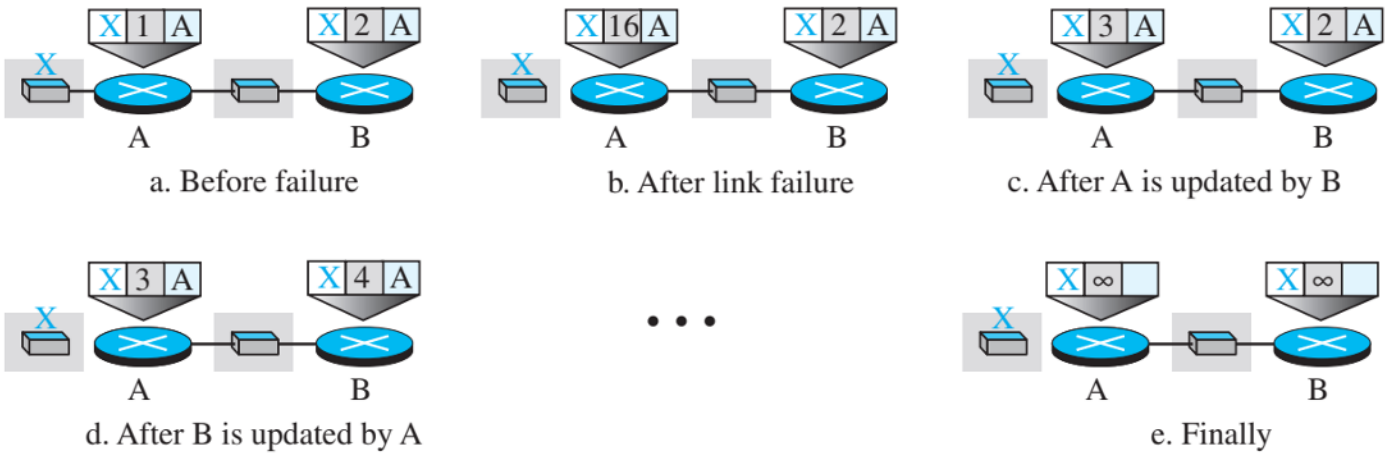
Count to Infinity

- A problem with distance-vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) will propagate slowly. but in distance-vector routing, this takes some time. The problem is referred to as **count to infinity.**

Two-Node Loop

- One example of count to infinity is the two-node loop problem. To understand the problem, let us look at the scenario depicted in Figure 20.7.

Figure 20.7 *Two-node instability*

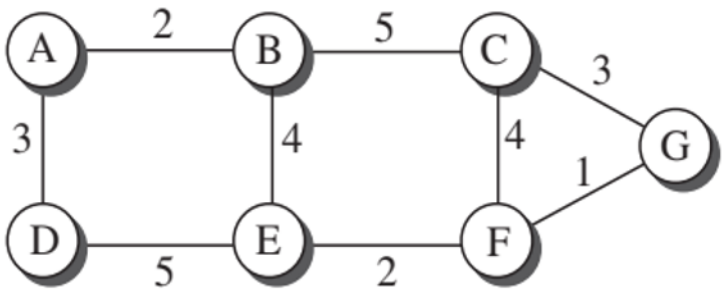


Unicast Routing : Link State Routing (LSR)

- This method uses the term link-state to define the characteristic of a link (an edge) that represents a network in the internet.
- In this algorithm the cost associated with an edge defines the state of the link.
- To create a least-cost tree with this method, each node needs to have a complete map of the network, which means it needs to know the state of each link.
- The collection of states for all links is called the link-state database (LSDB).
- There is only one LSDB for the whole internet; each node needs to have a duplicate of it to be able to create the least-cost tree

Unicast Routing : Link State Routing (LSR)

Figure 20.8 *Example of a link-state database*



a. The weighted graph

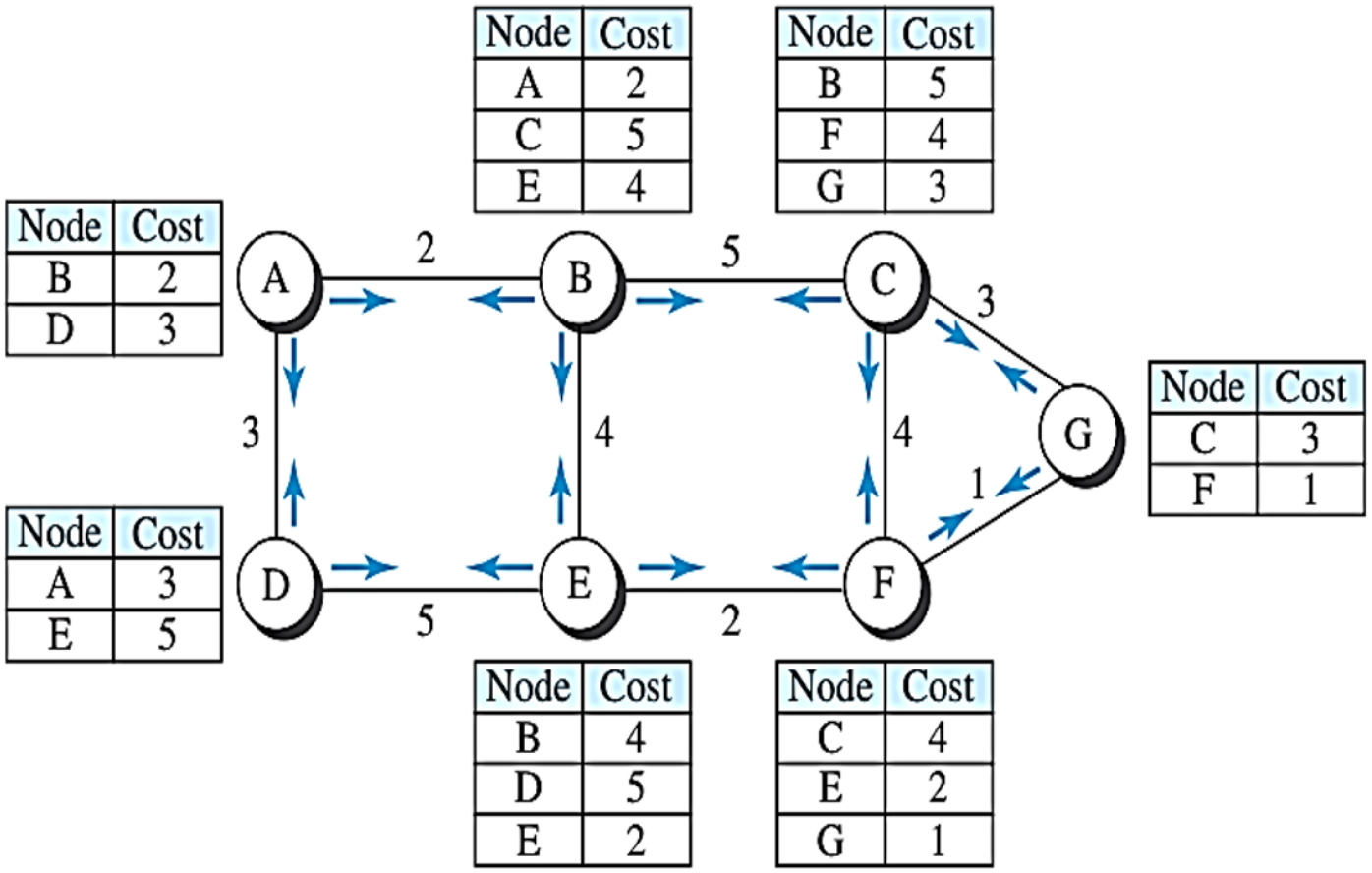
	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

b. Link state database

- **Now the question is how each node can create this LSDB that contains information about the whole internet.**
- **This can be done by a process called flooding.**
- **Each node can send some greeting messages to all its immediate neighbors to collect two pieces of information for each neighboring node: the identity of the node and the cost of the link**

Unicast Routing : Link State Routing (LSR)

Figure 20.9 *LSPs created and sent out by each node to build LSDB*



Unicast Routing : Path Vector Routing (PVR)

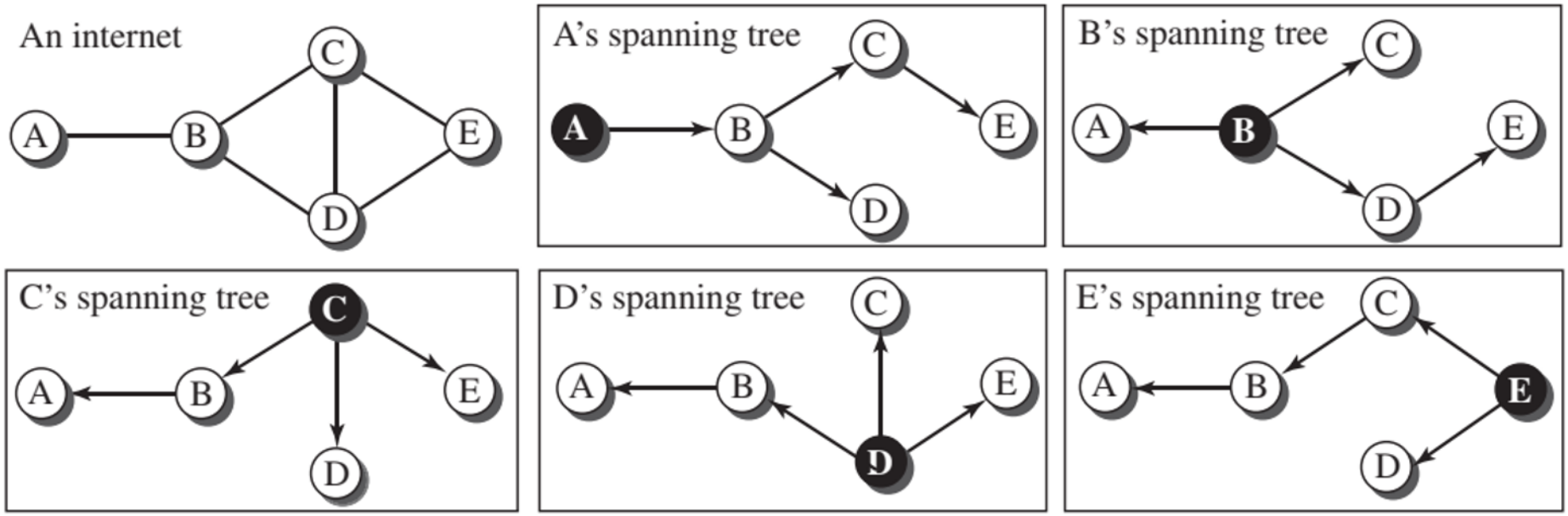
- Path-vector routing does not have the drawbacks of LS or DV routing as described above because it is not based on least-cost routing.
- The best route is determined by the source using the policy it imposes on the route.
- In other words, the source can control the path.
- Although path-vector routing is not actually used in an internet, and is mostly designed to route a packet between ISPs, we discuss the principle of this method in this section as though applied to an internet.

Unicast Routing : Path Vector Routing (PVR)

Spanning Trees

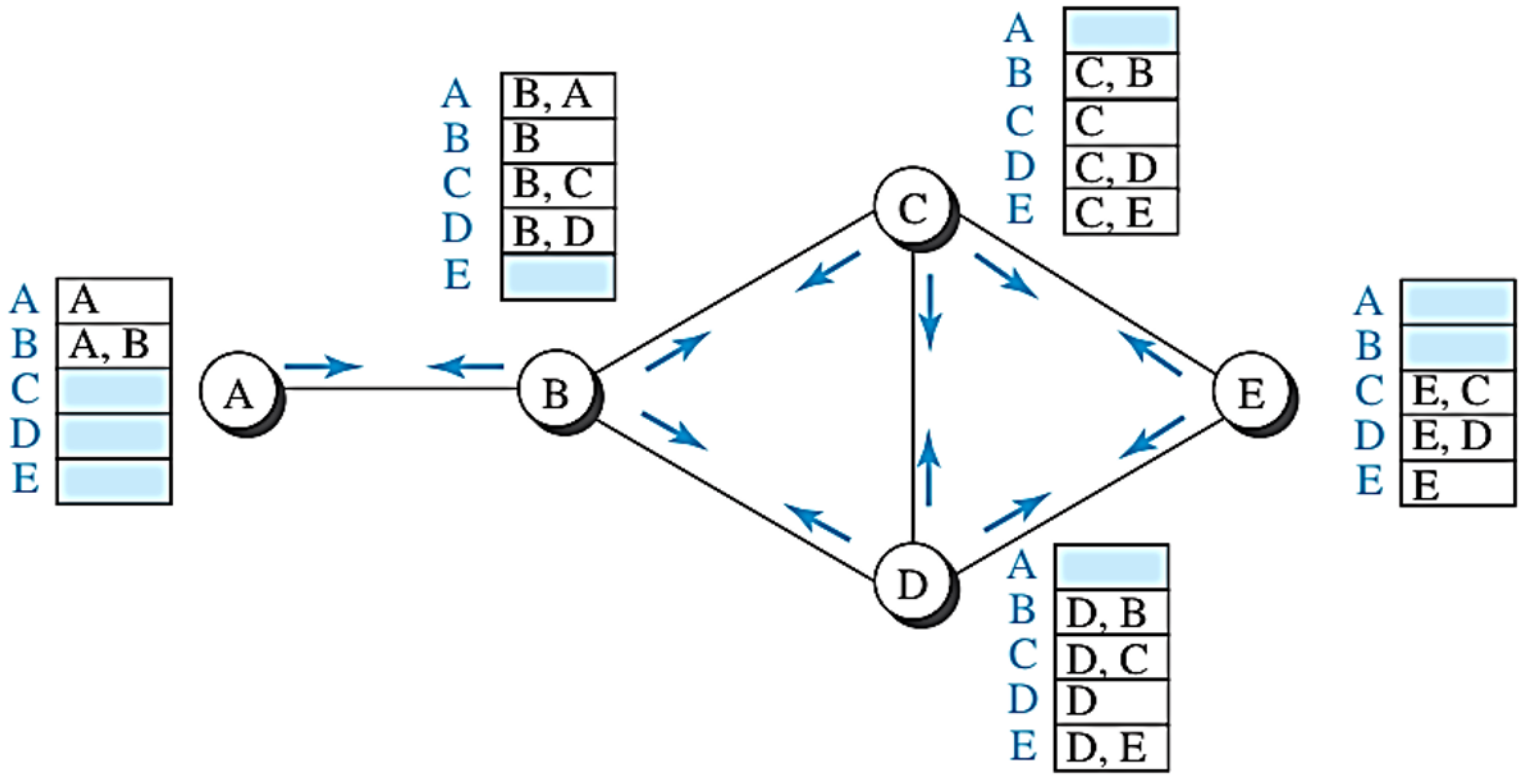
In path-vector routing, the path from a source to all destinations is also determined by the best spanning tree.

Figure 20.11 *Spanning trees in path-vector routing*



Unicast Routing : Path Vector Routing (PVR)

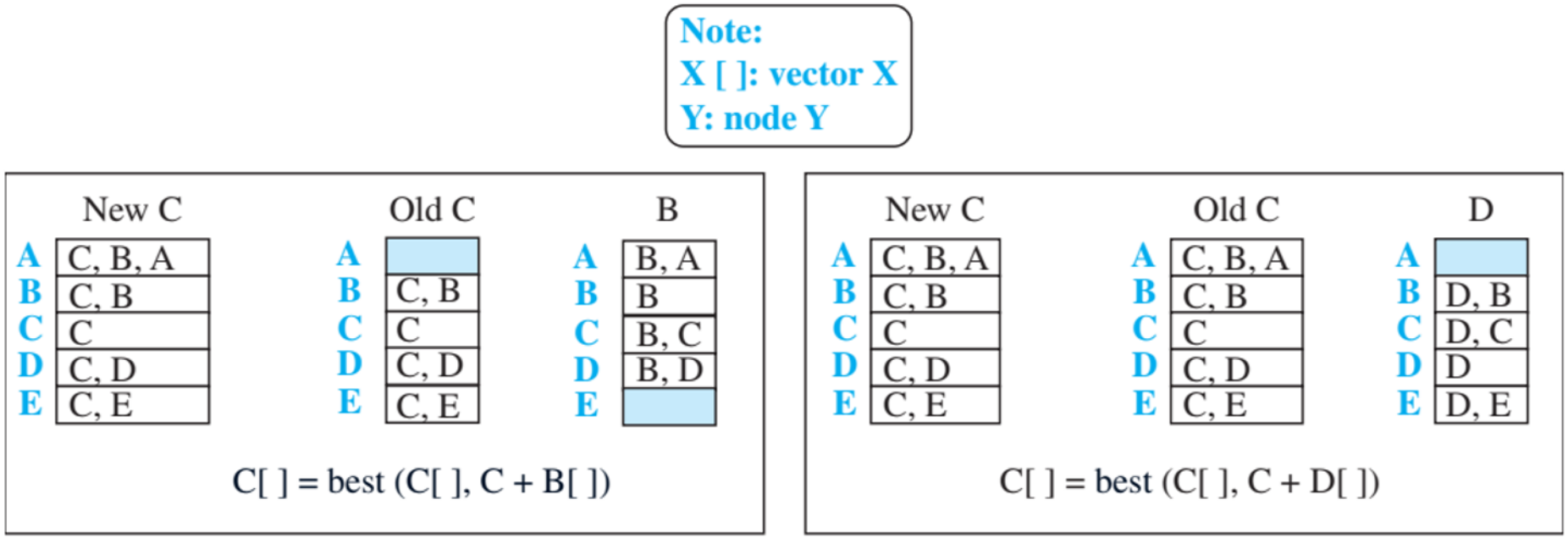
Figure 20.12 *Path vectors made at booting time*



Unicast Routing : Path Vector Routing (PSR)

$$\text{Path}(x, y) = \text{best}\ \{\text{Path}(x, y), [(x + \text{Path}(v, y)]\} \quad \text{for all } v\text{'s in the internet.}$$

Figure 20.13 *Updating path vectors*



Event 1: C receives a copy of B's vector

Event 2: C receives a copy of D's vector

Unicast Routing : Path Vector Routing (PSR)

Table 20.3 *Path-vector algorithm for a node*

```
1  Path_Vector_Routing ( )
2  {
3     // Initialization
4     for (y = 1 to N)
5     {
6        if (y is myself)
7           Path[y] = myself
8        else if (y is a neighbor)
9           Path[y] = myself + neighbor node
10       else
11          Path[y] = empty
12    }
13    Send vector {Path[1], Path[2], …, Path[y]} to all neighbors
14    // Update
15    repeat (forever)
16    {
17       wait (for a vector Path_w from a neighbor w)
18       for (y = 1 to N)
19       {
20          if (Path_w includes myself)
21             discard the path                    // Avoid any loop
22          else
23             Path[y] = best {Path[y], (myself + Path_w[y])}
24       }
25       If (there is a change in the vector)
26          Send vector {Path[1], Path[2], …, Path[y]} to all neighbors
27    }
28 } // End of Path Vector
```

UNICAST ROUTING PROTOCOLS

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

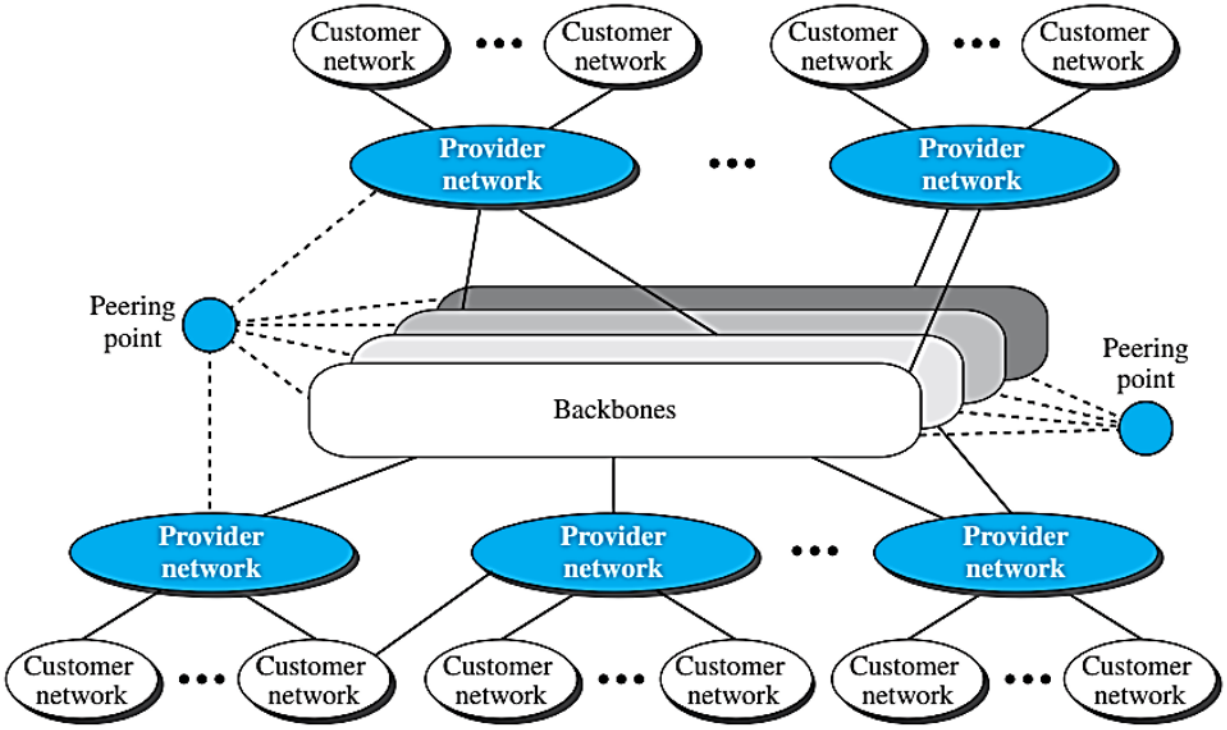
Topics discussed in this section:

1. Routing Information Protocol (RIP)
2. Open Shortest Path First (OSPF)
3. Border Gateway Protocol (BGP)
4. Multicast Open Shortest Path First (MOSPF)

Unicast Routing Protocols :

- Before discussing unicast routing protocols, we need to understand the structure of today's Internet.
- The Internet has changed from a tree-like structure, with a single backbone, to a multi-backbone structure run by different private corporations today.

Figure 20.14 *Internet structure*



Autonomous Systems:AS

- As we said before, each ISP is an autonomous system when it comes to managing networks and routers under its control.
- Each AS is given an autonomous number (ASN) by the ICANN.
- Each ASN is a 16-bit unsigned integer that uniquely defines an AS. The autonomous systems, however, are not categorized according to their size; they are categorized according to the way they are connected to other ASs.

❑ **Stub AS**. A stub AS has only one connection to another AS. The data traffic can be either initiated or terminated in a stub AS; the data cannot pass through it

❑ **Multihomed AS**. A multihomed AS can have more than one connection to other ASs, but it does not allow data traffic to pass through it.

❑ **Transient AS**. A transient AS is connected to more than one other AS and also allows the traffic to pass through.

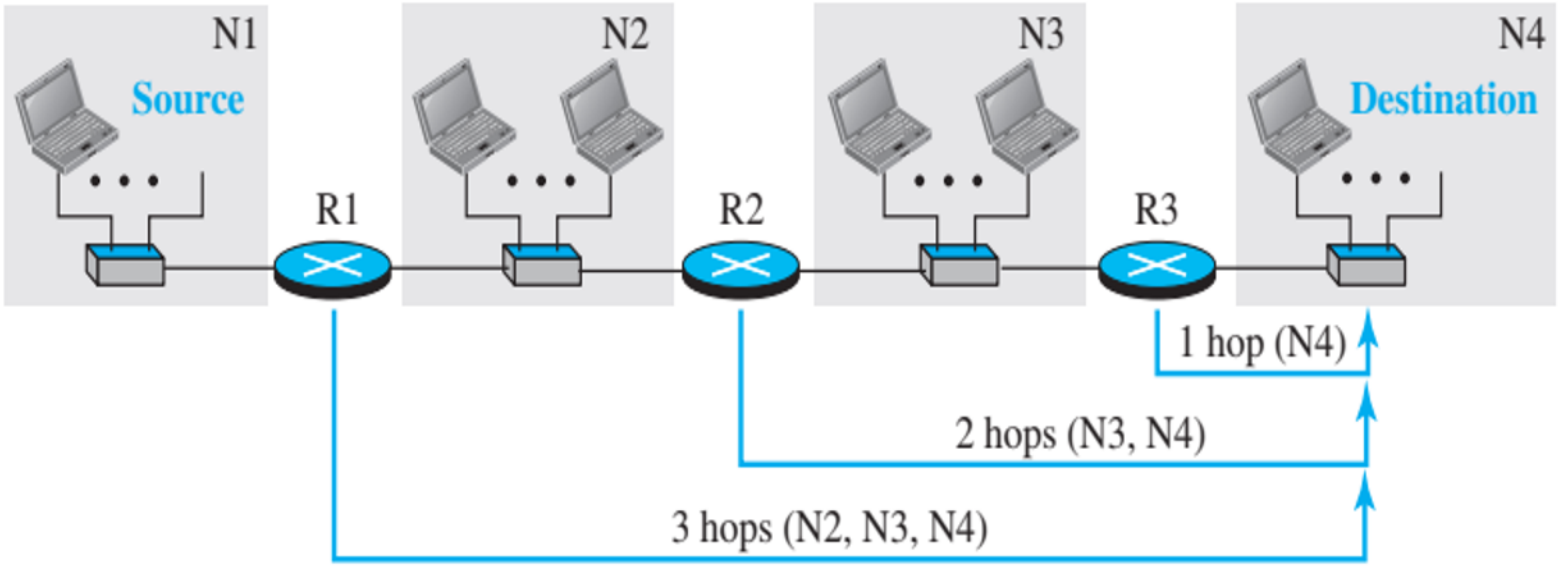
Unicast Routing Protocols :Routing Information Protocol (RIP)

- The Routing Information Protocol (RIP) is one of the most widely used intradomain routing protocols based on the distance-vector routing algorithm.
- RIP was started as part of the Xerox Network System (XNS), but it was the Berkeley Software Distribution (BSD) version of UNIX that helped make the use of RIP widespread.

Hop Count

- A router in this protocol basically implements the distance-vector routing algorithm shown in Table 20.1.
- However, the algorithm has been modified as described below.
- First, since a router in an AS needs to know how to forward a packet to different networks (subnets) in an AS, RIP routers advertise the cost of reaching different networks instead of reaching other nodes in a theoretical graph.
- Second, to make the implementation of the cost simpler the cost is defined as the number of hops, which means the number of networks (subnets) a packet needs to travel through from the source router to the final destination host.

Figure 20.15 *Hop counts in RIP*



Forwarding Tables

- The routers in an autonomous system need to keep forwarding tables to forward packets to their destination networks.
- A forwarding table in RIP is a three-column table in which the first column is the address of the destination network, the second column is the address of the next router to which the packet should be forwarded, and the third column is the cost (the number of hops) to reach the destination network.

Figure 20.16 *Forwarding tables*

Forwarding table for R1

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

Forwarding table for R2

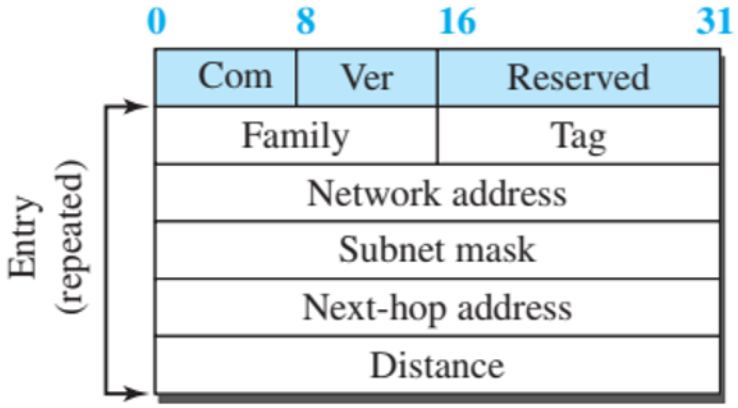
Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

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

RIP Messages

Two RIP processes, a client and a server, like any other processes, need to exchange messages.
RIP-2 defines the format of the message, as shown in Figure 20.17



Fields

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

- **RIP has two types of messages: request and response.**
- **A request message is sent by a router that has just come up or by a router that has some time-out entries.**
- **A request message can ask about specific entries or all entries.**
- **A response (or update) message can be either solicited or unsolicited. A solicited response message is sent only in answer to a request message.**

RIP Algorithm

- RIP implements the same algorithm as the distance-vector routing algorithm. However, some changes need to be made to the algorithm to enable a router to update its forwarding table:
  - Instead of sending only distance vectors, a router needs to send the whole contents of its forwarding table in a response message.
  - The receiver adds one hop to each cost and changes the next router field to the address of the sending router.

1. If the received route does not exist in the old forwarding table, it should be added to the route.
2. If the cost of the received route is lower than the cost of the old one, the received route should be selected as the new one.
3. If the cost of the received route is higher than the cost of the old one, but the value of the next router is the same in both routes, the received route should be selected as the new one.

Routing Information Protocol : Timers

The periodic timer controls the advertising of regular update messages. Each router has one periodic timer that is randomly set to a number between 25 and 35 seconds The timer counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

The expiration timer governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 seconds for that particular route. Every time a new update for the route is received, the timer is reset.

The garbage collection timer is used to purge a route from the forwarding table. When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16.

Performance

❑ **Update Messages. The update messages in RIP have a very simple format and are sent only to neighbors; they are local. They do not normally create traffic because the routers try to avoid sending them at the same time.**

❑ **Convergence of Forwarding Tables. RIP uses the distance-vector algorithm, which can converge slowly if the domain is large, but, since RIP allows only 15 hops in a domain (16 is considered as infinity), there is normally no problem in convergence.**

❑ **Robustness. As we said before, distance-vector routing is based on the concept that each router sends what it knows about the whole domain to its neighbors.**

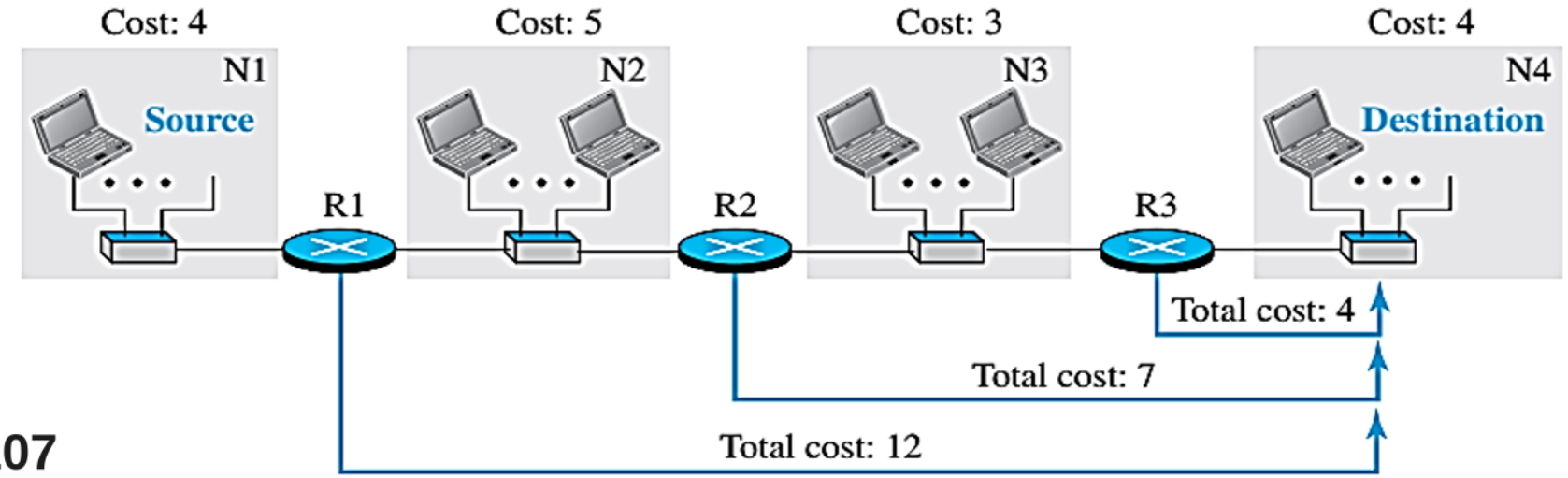
- **If there is a failure or corruption in one router, the problem will be propagated to all routers and the forwarding in each router will be affected.**

Unicast Routing Protocols : Open Shortest Path First (OSPF)

- OSPF is also an intradomain routing protocol like RIP, but it is based on the link-state routing protocol.
- OSPF is an open protocol, which means that the specification is a public document

Metric

- In OSPF, like RIP, the cost of reaching a destination from the host is calculated from the source router to the destination network.
- However, each link (network) can be assigned a weight based on the throughput, round-trip time, reliability, and so on



OSPF: Forwarding Tables

- Each OSPF router can create a forwarding table after finding the shortest-path tree between itself and the destination using Dijkstra's algorithm,
- Comparing the forwarding tables for the OSPF and RIP in the same AS, we find that the only difference is the cost values. In other words, if we use the hop count for OSPF, the tables will be exactly the same.

Figure 20.20 *Forwarding tables in OSPF*

Forwarding table for R1

Destination network	Next router	Cost
N1	——	4
N2	——	5
N3	R2	8
N4	R2	12

Forwarding table for R2

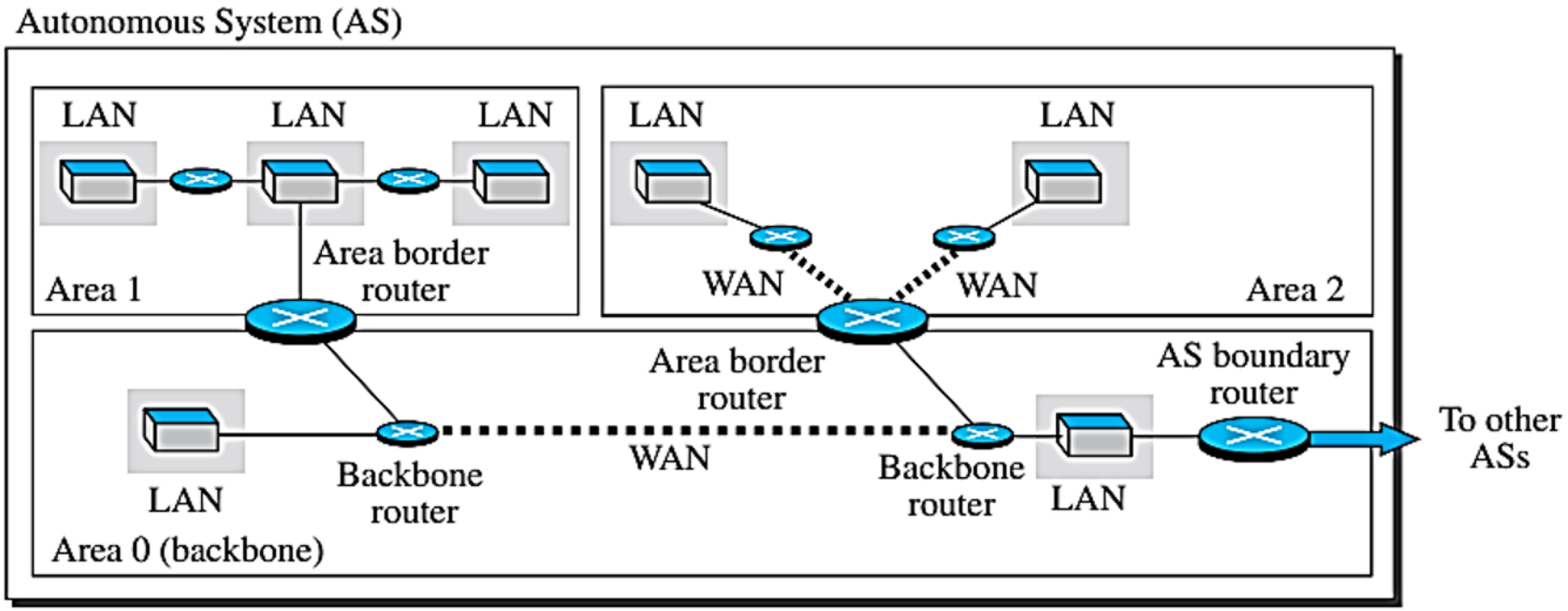
Destination network	Next router	Cost
N1	R1	9
N2	——	5
N3	——	3
N4	R3	7

Forwarding table for R3

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

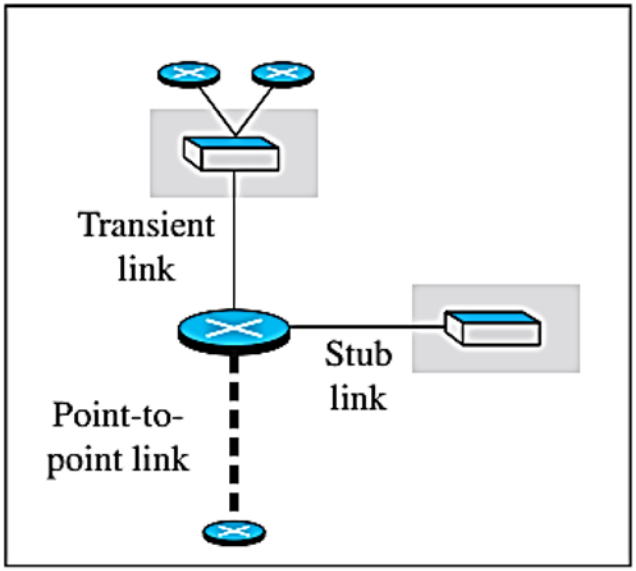
OSPF: Areas

- Compared with RIP, which is normally used in small ASs, OSPF was designed to be able to handle routing in a small or large autonomous system
- To prevent volume or traffic this, the AS needs to be divided into small sections called areas. Each area acts as a small independent domain for flooding LSPs.

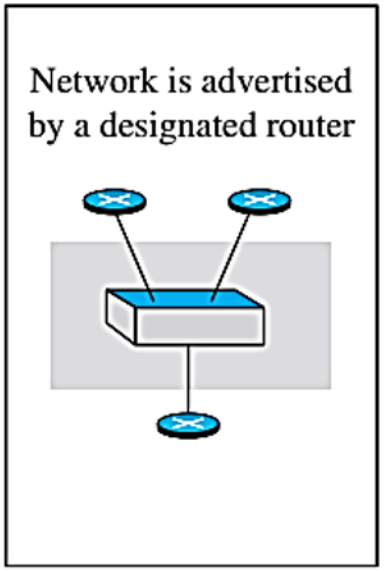


OSPF: Link state Advertisement

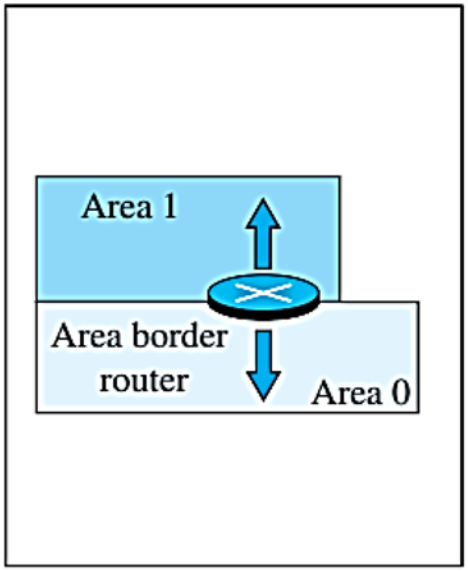
OSPF is based on the link-state routing algorithm, which requires that a router advertise the state of each link to all neighbors for the formation of the LSDB



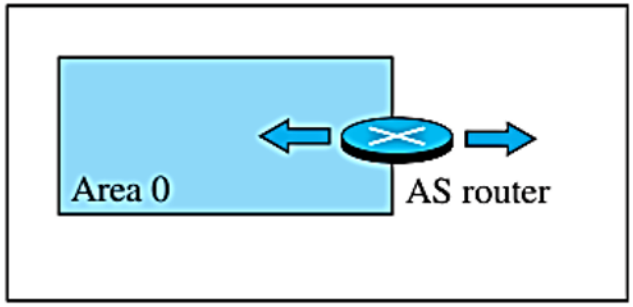
a. Router link



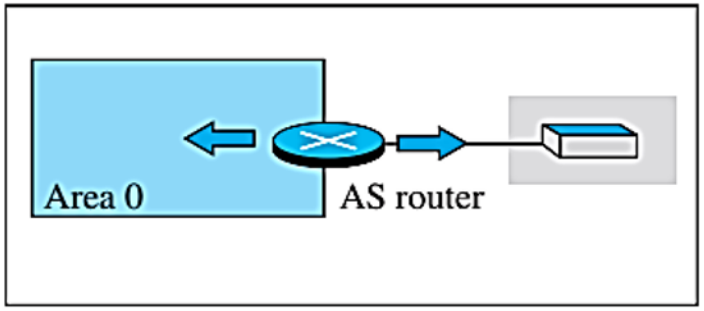
b. Network link



c. Summary link to network



d. Summary link to AS



e. External link

OSPF: Link state Advertisement

- **Router link**. A router link advertises the existence of a router as a node.
  - A transient link announces a link to a transient network, a network that is connected to the rest of the networks by one or more routers.
  - A stub link advertises a link to a stub network, a network that is not a through network.
- **Network link.** A network link advertises the network as a node. However, since a network cannot do announcements itself one of the routers is assigned as the designated router and does the advertising
- **Summary link to network.** This is done by an area border router; it advertises the summary of links collected by the backbone to an area
- **Summary link to AS**. This is done by an AS router that advertises the summary links from other ASs to the backbone area of the current AS
- **External link.** This is also done by an AS router to announce the existence of a single network outside the AS to the backbone area to be disseminated into the areas

Performance

❑ **Update Messages. The link-state messages in OSPF have a somewhat complex format. They also are flooded to the whole area. If the area is large, these messages may create heavy traffic and use a lot of bandwidth..**

❑ **Convergence of Forwarding Tables. When the flooding of LSPs is completed, each router can create its own shortest-path tree and forwarding table; convergence is fairly quick**

- **Robustness : The OSPF protocol is more robust than RIP because, after receiving the completed LSDB, each router is independent and does not depend on other routers in the area. Corruption or failure in one router does not affect other routers as seriously as in RIP**

Multicast Protocol: Multicast Open Shortest Path First(MOSPF)

- MOSPF is the extension of the Open Shortest Path First (OSPF) protocol, which is used in unicast routing. It also uses the source based tree approach to multicasting.
- Now a router goes through the following steps to forward a multicast packet received from source S and to be sent to destination G

1. The router uses the Dijkstra algorithm to create a shortest-path tree with S as the root and all destinations in the internet as the leaves.

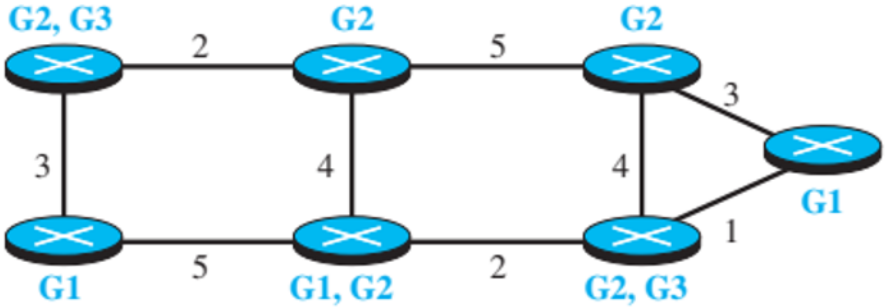
2. The router finds itself in the shortest-path tree created in the first step. In other words, the router creates a shortest-path subtree with itself as the root of the subtree

3. The shortest-path subtree is actually a broadcast subtree with the router as the root and all networks as the leaves.

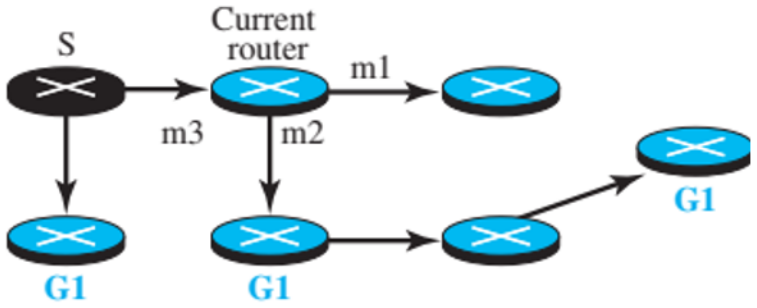
4. The router can now forward the received packet out of only those interfaces that correspond to the branches of the multicast tree.

Multicast Open Shortest Path First (MOSPF)

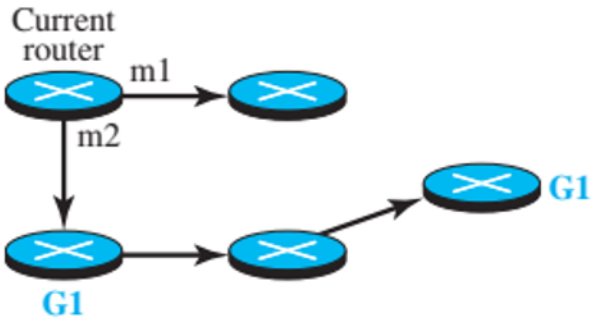
Figure 21.13 *Example of tree formation in MOSPF*



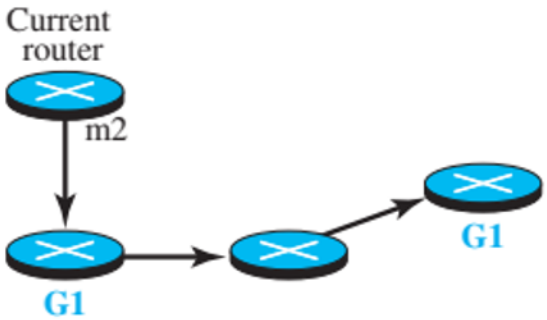
a. An internet with some active groups



b. S-G1 shortest-path tree



c. S-G1 subtree seen by current router



d. S-G1 pruned subtree

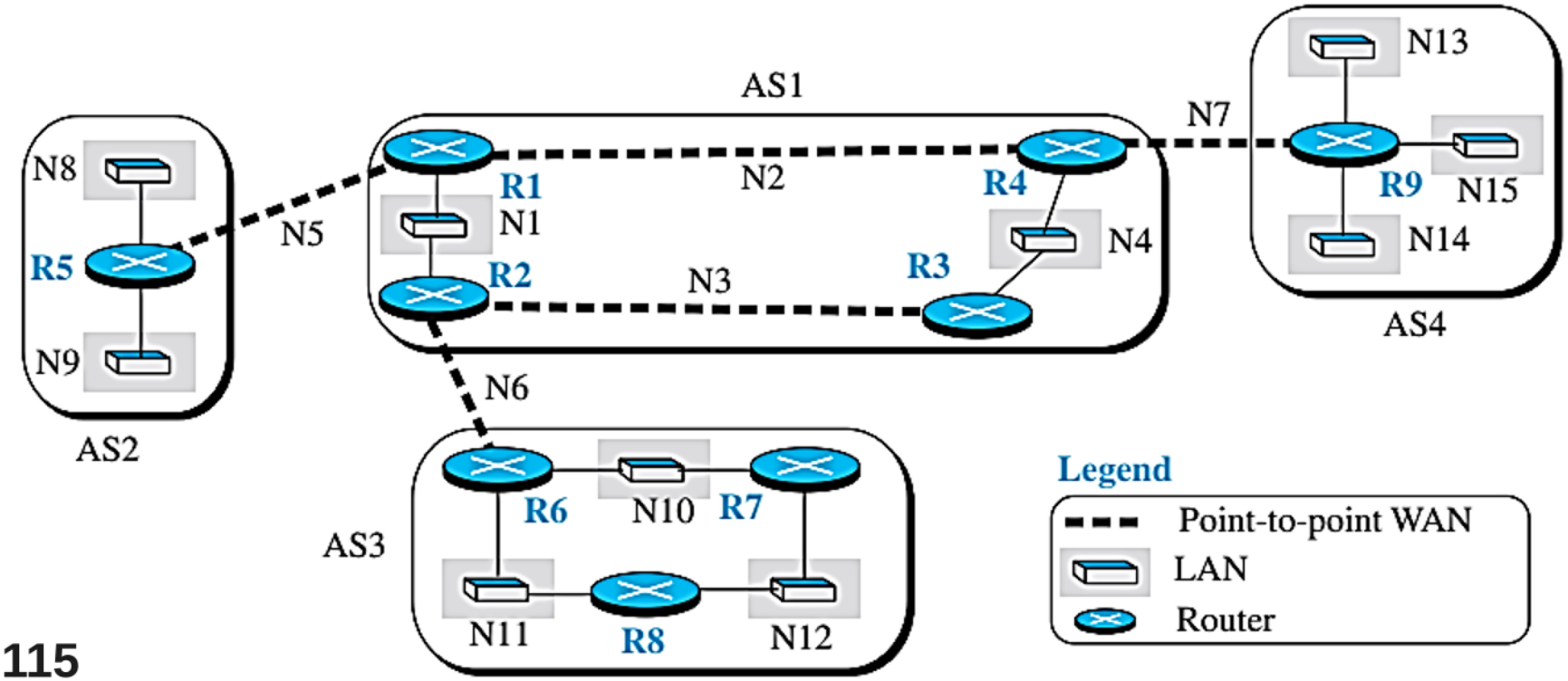
Forwarding table for current router

Group-Source	Interface
S, G1	m2
...	...

Border Gateway Protocol Version-4 (BGP4)

- BGP4 is the only interdomain routing protocol used in the Internet today. BGP4 is based on the path-vector algorithm
- AS2, AS3, and AS4 are stub autonomous systems; AS1 is a transient one. In our example, data exchange between AS2, AS3, and AS4 should pass through AS1.

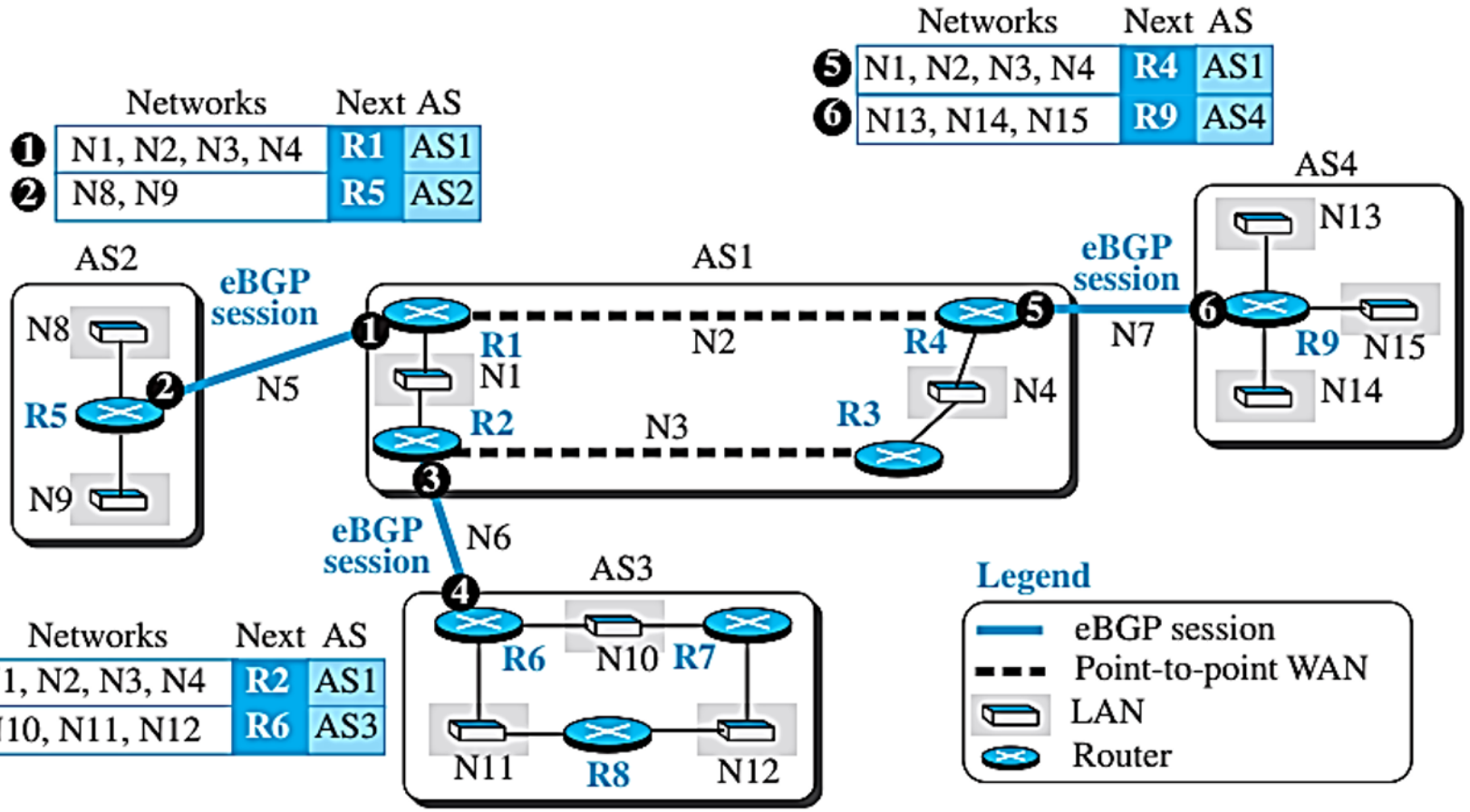
Figure 20.24 *A sample internet with four ASs*



BGP4: Operation of External BGP (eBGP)

The eBGP variation of BGP allows two physically connected border routers in two different ASs to form pairs of eBGP speakers and exchange messages

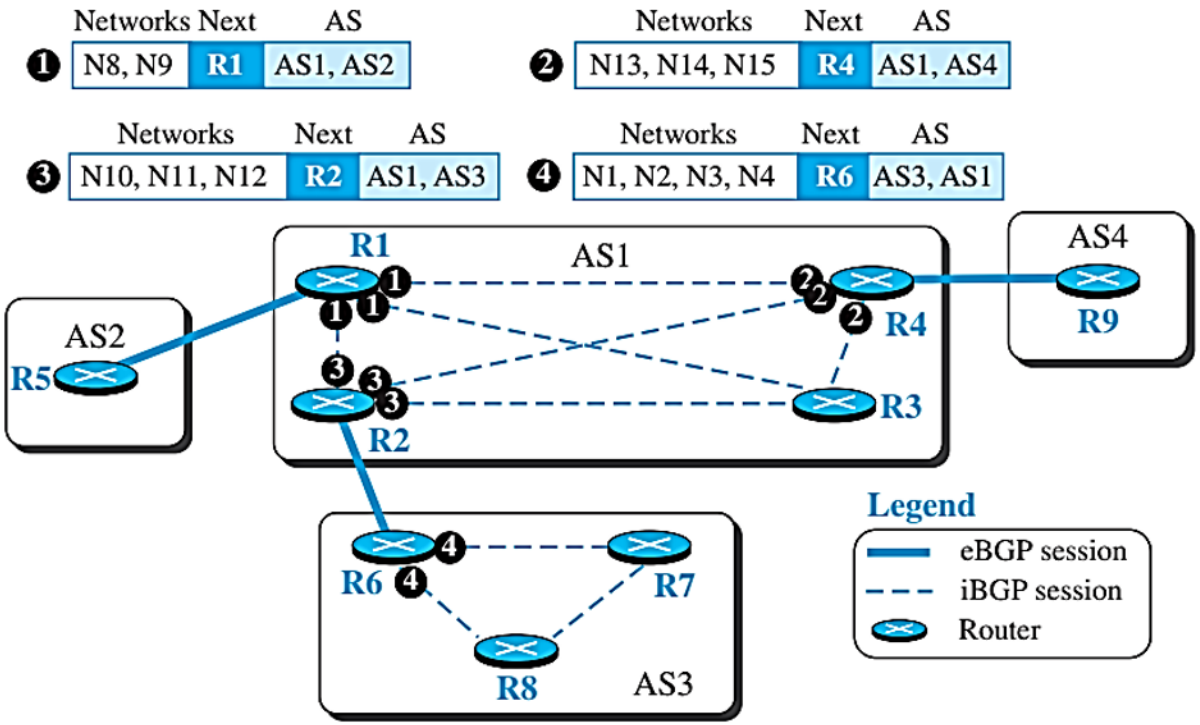
Figure 20.25 *eBGP operation*



BGP4: Operation of internal BGP (iBGP)

- The iBGP protocol creates a session between any possible pair of routers inside an autonomous system. However, some points should be made clear.
- First, if an AS has only one router, there cannot be an iBGP session. For example, we cannot create an iBGP session inside AS2 or AS4 in our internet.

Figure 20.26 *Combination of eBGP and iBGP sessions in our internet*



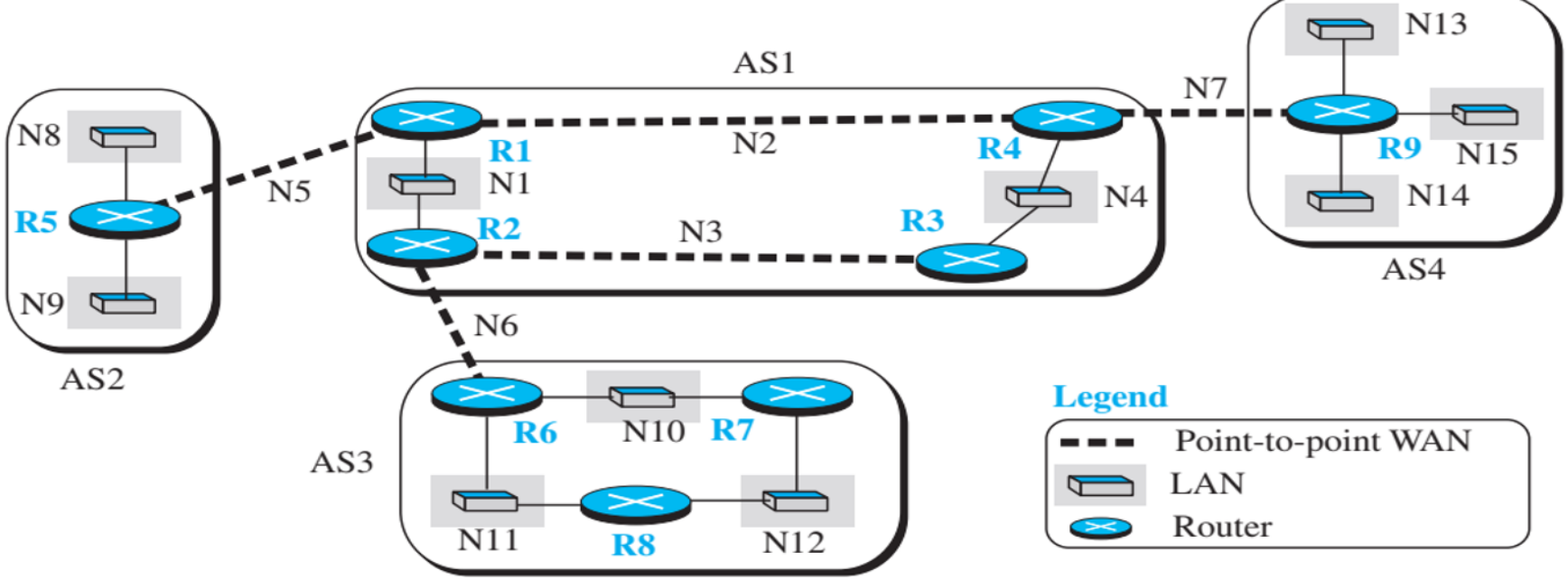


Figure 20.27 Finalized BGP path tables

Networks	Next	Path
N8, N9	R5	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R1

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R6	AS1, AS3
N13, N14, N15	R1	AS1, AS4

Path table for R2

Networks	Next	Path
N8, N9	R2	AS1, AS2
N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4

Path table for R3

Networks	Next	Path
N8, N9	R1	AS1, AS2
N10, N11, N12	R1	AS1, AS3
N13, N14, N15	R9	AS1, AS4

Path table for R4

Networks	Next	Path
N1, N2, N3, N4	R1	AS2, AS1
N10, N11, N12	R1	AS2, AS1, AS3
N13, N14, N15	R1	AS2, AS1, AS4

Path table for R5

Networks	Next	Path
N1, N2, N3, N4	R2	AS3, AS1
N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R2	AS3, AS1, AS4

Path table for R6

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R7

Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1
N8, N9	R6	AS3, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4

Path table for R8

Networks	Next	Path
N1, N2, N3, N4	R4	AS4, AS1
N8, N9	R4	AS4, AS1, AS2
N10, N11, N12	R4	AS4, AS1, AS3

Path table for R9

Figure 20.28 *Forwarding tables after injection from BGP*

Des.	Next	Cost
N1	—	1
N4	R4	2
N8	R5	1
N9	R5	1
N10	R2	2
N11	R2	2
N12	R2	2
N13	R4	2
N14	R4	2
N15	R4	2

Table for R1

Des.	Next	Cost
N1	—	1
N4	R3	2
N8	R1	2
N9	R1	2
N10	R6	1
N11	R6	1
N12	R6	1
N13	R3	3
N14	R3	3
N15	R3	3

Table for R2

Des.	Next	Cost
N1	R2	2
N4	—	1
N8	R2	3
N9	R2	3
N10	R2	2
N11	R2	2
N12	R2	2
N13	R4	2
N14	R4	2
N15	R4	2

Table for R3

Des.	Next	Cost
N1	R1	2
N4	—	1
N8	R1	2
N9	R1	2
N10	R3	3
N11	R3	3
N12	R3	3
N13	R9	1
N14	R9	1
N15	R9	1

Table for R4

Des.	Next	Cost
N8	—	1
N9	—	1
0	R1	1

Table for R5

Des.	Next	Cost
N10	—	1
N11	—	1
N12	R7	2
0	R2	1

Table for R6

Des.	Next	Cost
N10	—	1
N11	R6	2
N12	—	1
0	R6	2

Table for R7

Des.	Next	Cost
N10	R6	2
N11	—	1
N12	—	1
0	R6	2

Table for R8

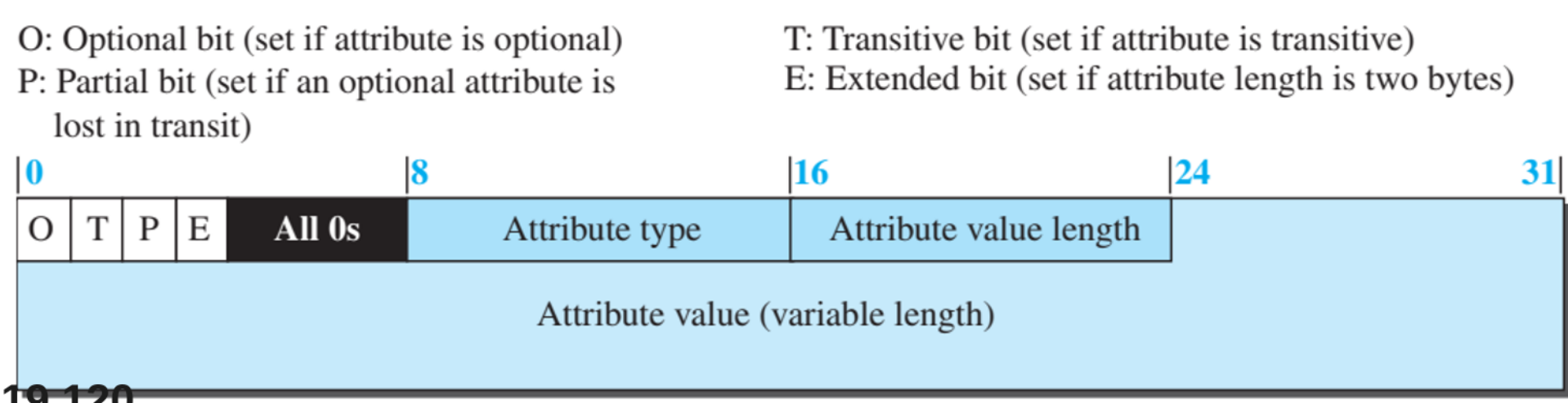
Des.	Next	Cost
N13	—	1
N14	—	1
N15	—	1
0	R4	1

Table for R9

BGP4: Path attributes

- Interdomain routing is more involved and naturally needs more information about how to reach the final destination.
- In BGP these pieces are called path attributes. BGP allows a destination to be associated with up to seven path attributes.
- Path attributes are divided into two broad categories: well-known and optional. A well-known attribute must be recognized by all routers; an optional attribute need not be.

Figure 20.29 *Format of path attribute*



BGP4: Attribute Types

- **ORIGIN (type 1).** This is a well-known mandatory attribute, which defines the source of the routing information.

This attribute can be defined by one of the three values: 1, 2, 3.

- ✓ Value 1 means that the information about the path has been taken from an intradomain protocol (RIP or OSPF).
- ✓ Value 2 means that the information comes from BGP.
- ✓ Value 3 means that it comes from an unknown source.

- **AS-PATH (type 2).** This is a well-known mandatory attribute, which defines the list of autonomous systems through which the destination can be reached.

- **NEXT-HOP (type 3).** This is a well-known mandatory attribute, which defines the next router to which the data packet should be forwarded.

- **MULT-EXIT-DISC (type 4).** The multiple-exit discriminator is an optional intransitive attribute, which discriminates among multiple exit paths to a destination.

BGP4: Attribute Types

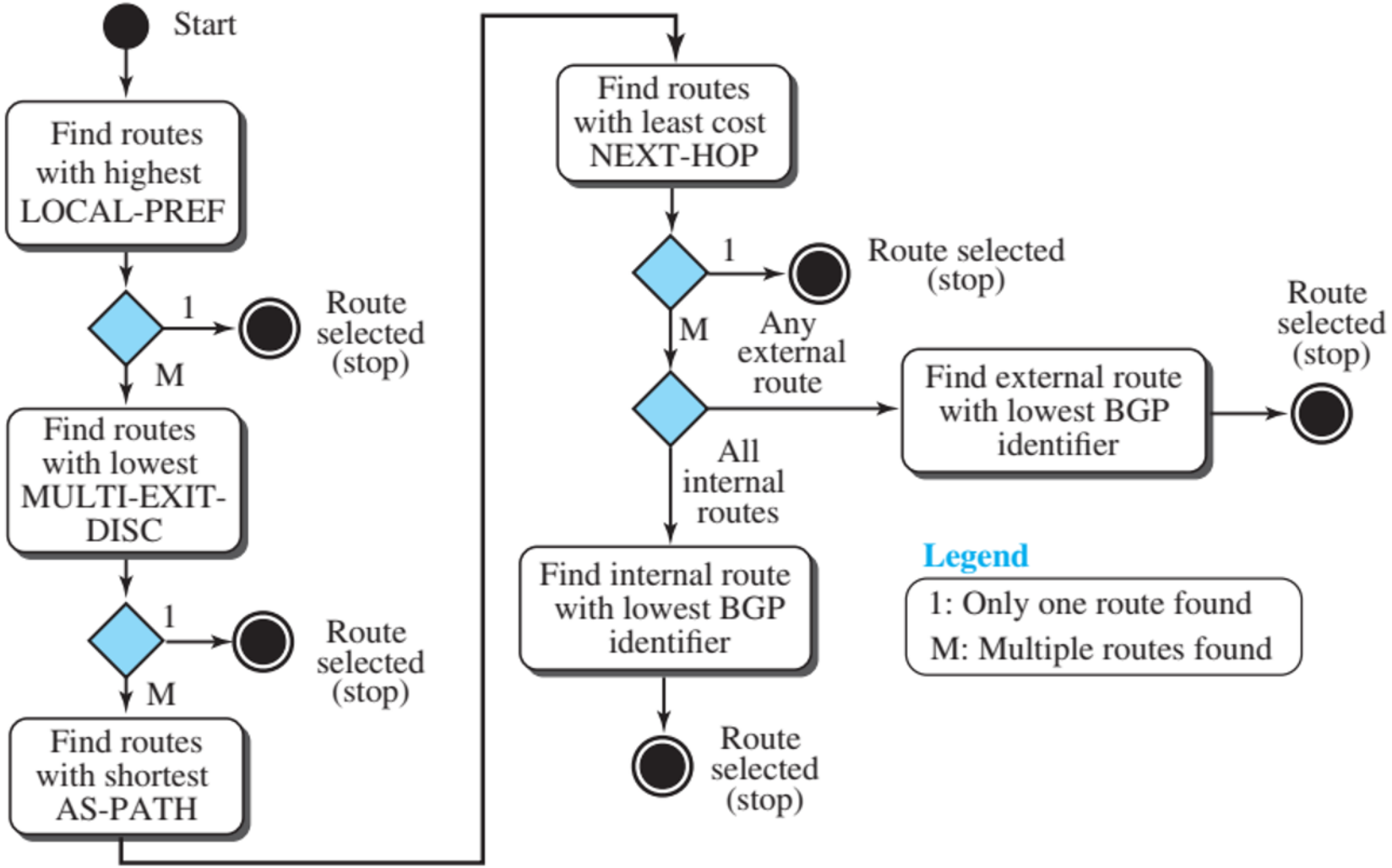
LOCAL-PREF (type 5). The local preference attribute is a well-known discretionary attribute. It is normally set by the administrator, based on the organization policy. The routes the administrator prefers are given a higher local preference value

ATOMIC-AGGREGATE (type 6). This is a well-known discretionary attribute, which defines the destination prefix as not aggregate; it only defines a single destination network. This attribute has no value field, which means the value of the
length field is zero

AGGREGATOR (type 7). This is an optional transitive attribute, which emphasizes that the destination prefix is an aggregate. The attribute value gives the number of the last AS that did the aggregation followed by the IP address of the router that did so

BGP4: Route Selection BGP (BGP)

Figure 20.30 *Flow diagram for route selection*



BGP4: Messages

BGP uses four types of messages for communication between the BGP speakers across the ASs and inside an AS:

- **Open Message.** To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message.

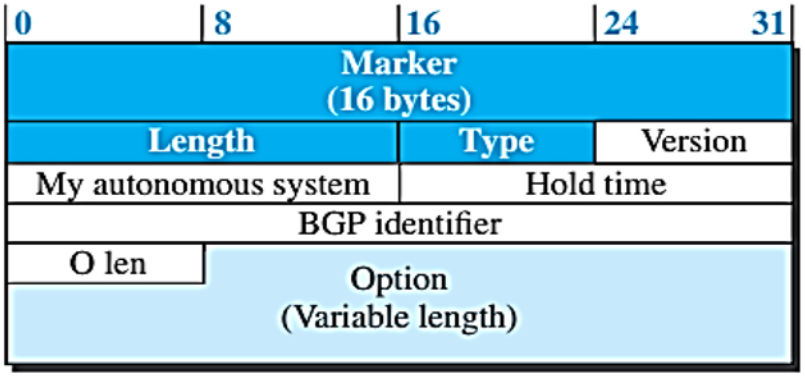
- **Update Message.** The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, to announce a route to a new destination, or both.

- **Keepalive Message.** The BGP peers that are running exchange keepalive messages regularly (before their hold time expires) to tell each other that they are alive.

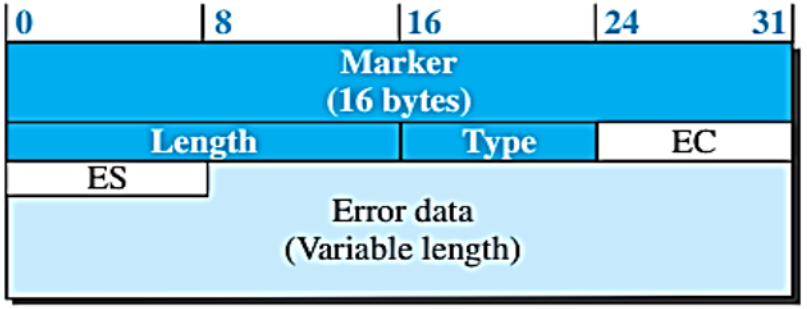
- **Notification.** A notification message is sent by a router whenever an error condition is detected or a router wants to close the session.

BGP4: Messages

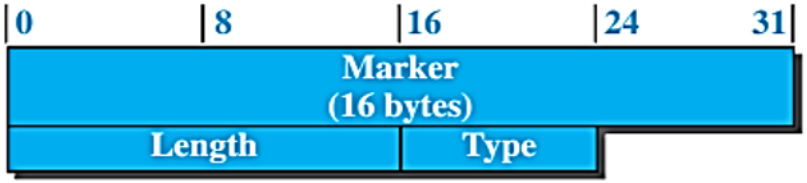
Figure 20.31 *BGP messages*



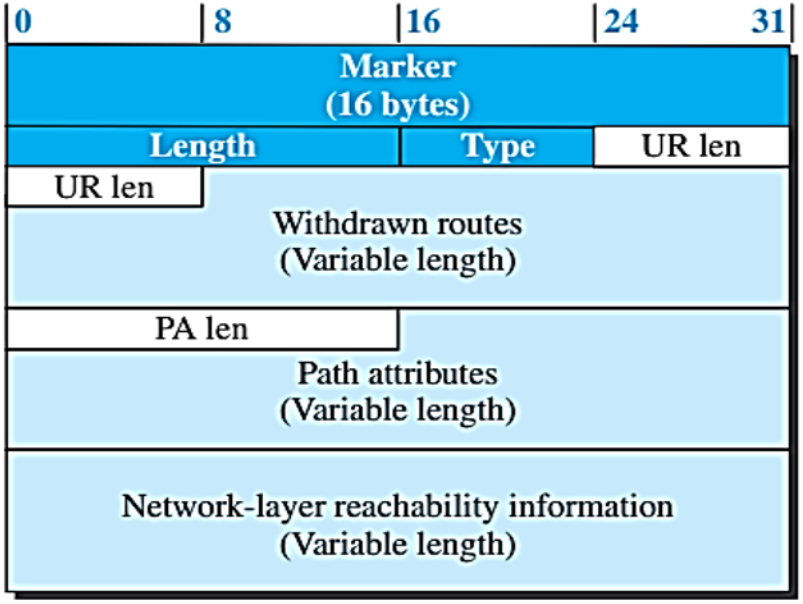
Open message (type 1)



Notification message (type 3)



Keepalive message (type 4)



Update message (type 2)

Fields in common header

Marker: Reserved for authentication
Length: Length of total message in bytes
Type: Type of message (1 to 4)

Abbreviations

O len: Option length
EC: Error code
ES: Error subcode
UR len: Unfeasible route length
PA len: Path attribute length

End of Module-3

MODULE-1

Introduction: Data Communications, Networks, Network Types, Networks Models: Protocol Layering, TCP/IP Protocol suite, The OSI model, Introduction to Physical Layer: Transmission media, Guided Media, Unguided Media: Wireless. Switching: Packet Switching and its types.

MODULE-2

Data Link Layer: Error Detection and Correction: Introduction, Block Coding, Cyclic Codes. Data link control: DLC Services: Framing, Flow Control, Error Control, Connectionless and Connection Oriented, Data link layer protocols, High Level Data Link Control. Media Access Control: Random Access, Controlled Access. Check Sum and Point to Point Protocol

MODULE-3

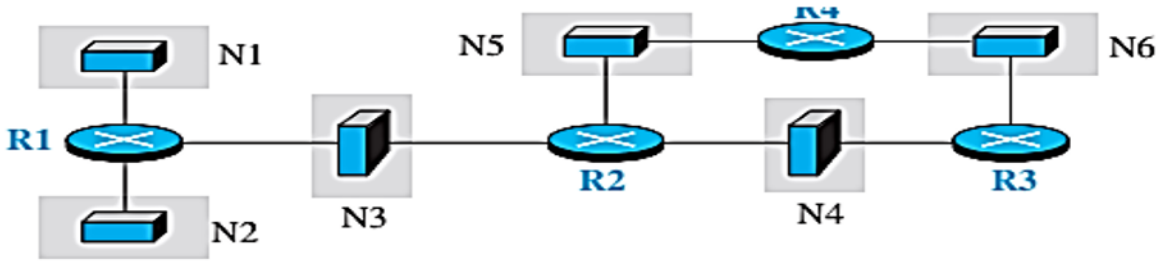
Network Layer: Network layer Services, Packet Switching, IPv4 Address, IPv4 Datagram, IPv6 Datagram, Introduction to Routing Algorithms, Unicast Routing Protocols: DVR, LSR, PVR, Unicast Routing protocols: RIP, OSPF, BGP, Multicasting Routing-MOSPF

MODULE-4

Introduction to Transport Layer: Introduction, Transport-Layer Protocols: Introduction, User Datagram Protocol, Transmission Control Protocol: services, features, segments, TCP connections, flow control, Error control, Congestion control.

MODULE-5

Introduction to Application Layer: Introduction, Client-Server Programming, Standard ClientServer Protocols: World Wide Web and HTTP, FTP, Electronic Mail, Domain Name System (DNS), TELNET, Secure Shell (SSH)



R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1

R2

Des.	N. R.	Cost
N3	———	1
N4	———	1
N5	———	1

R3

Des.	N. R.	Cost
N4	———	1
N6	———	1

R4

Des.	N. R.	Cost
N5	———	1
N6	———	1

Forwarding tables after all routers booted

New R1

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

Old R1

Des.	N. R.	Cost
N1	———	1
N2	———	1
N3	———	1

R2 Seen by R1

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

New R3

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

Old R3

Des.	N. R.	Cost
N4	———	1
N6	———	1

R2 Seen by R3

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

New R4

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

Old R4

Des.	N. R.	Cost
N5	———	1
N6	———	1

R2 Seen by R4

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

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

Final R1

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

Final R2

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

Final R3

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

Final R4

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

Forwarding tables for all routers after they have been stablized