Unit III - Network Layer

AGENDA

- IPv4 Addresses: Address Space, Notations, Classful Addressing, Classless Addressing,
- Network Address Translation (NAT)
- Need for Network Layer
- Internet as a Datagram Network
- Internet as a Connectionless Network
- IPv4: Segment Header Format,
- Datagram, Fragmentation, Checksum, Options
- IPv6: Advantages
- Packet Format
- Extension Headers
- Forwarding Techniques
- Forwarding Process
- Routing Table.

- The Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses. These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses
- The need for more addresses, in addition to other concerns about the IP layer, motivated a new design of the IP layer called the new generation of IP or IPv6 (lP version 6).
- In this version, the Internet uses 128-bit addresses that give much greater flexibility in address allocation. These addresses are referred to as IPv6 (IP version 6) addresses.

IPv4 ADDRESSES

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address.
- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet.
- The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.
- If a device has two connections to the Internet, via two networks, it has two IPv4 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

IPv4 ADDRESSES

- IPv4 is also a connection-less protocol 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.
- 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.

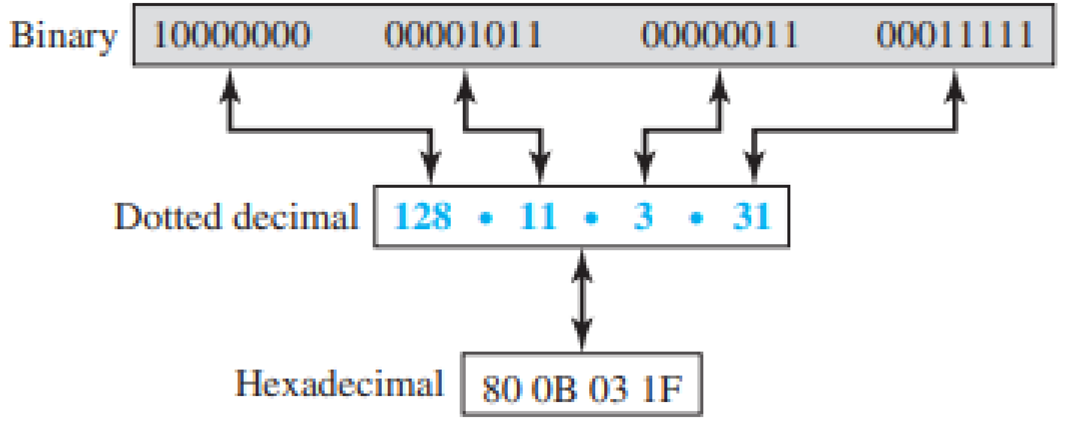
IPv4 ADDRESSES

- Address Space
- A protocol like 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 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 2^{32} 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.

Dotted-decimal notation and binary notation for an IPv4 address

- There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16).

Figure 18.16 *Three different notations in IPv4 addressing*



Dotted-decimal notation and binary notation for an IPv4 address

- In binary notation, an IPv4 address is displayed as 32 bits.
- To make the address more readable, one or more spaces are usually inserted between each octet (8 bits).
- Each octet is often referred to as a byte. To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes.
- This format is referred to as dotted-decimal notation.
- We sometimes see an IPv4 address in hexadecimal notation. Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.

IPv4 ADDRESSES

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

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

- We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation.

a. 129.11.11.239

b. 193.131.27.255

IPv4 ADDRESSES

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

a. 111.56.45.78

b. 221.34.7.82

- We replace each decimal number with its binary equivalent

a. 01101111 00111000 00101101 01001110

b. 11011101 00100010 00000111 01010010

IPv4 ADDRESSES

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

a. 111.56.045.78

b. 221.34.7.8.20

c. 75.45.301.14

d. 11100010.23.14.67

Solution

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

IPv4 ADDRESSES

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.
- A prefix can be fixed length or variable length.
- The network identifier in the IPv4 was first designed as a fixed-length prefix.
- This scheme, which is now obsolete, is referred to as classful addressing. The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.

IPv4 ADDRESSES

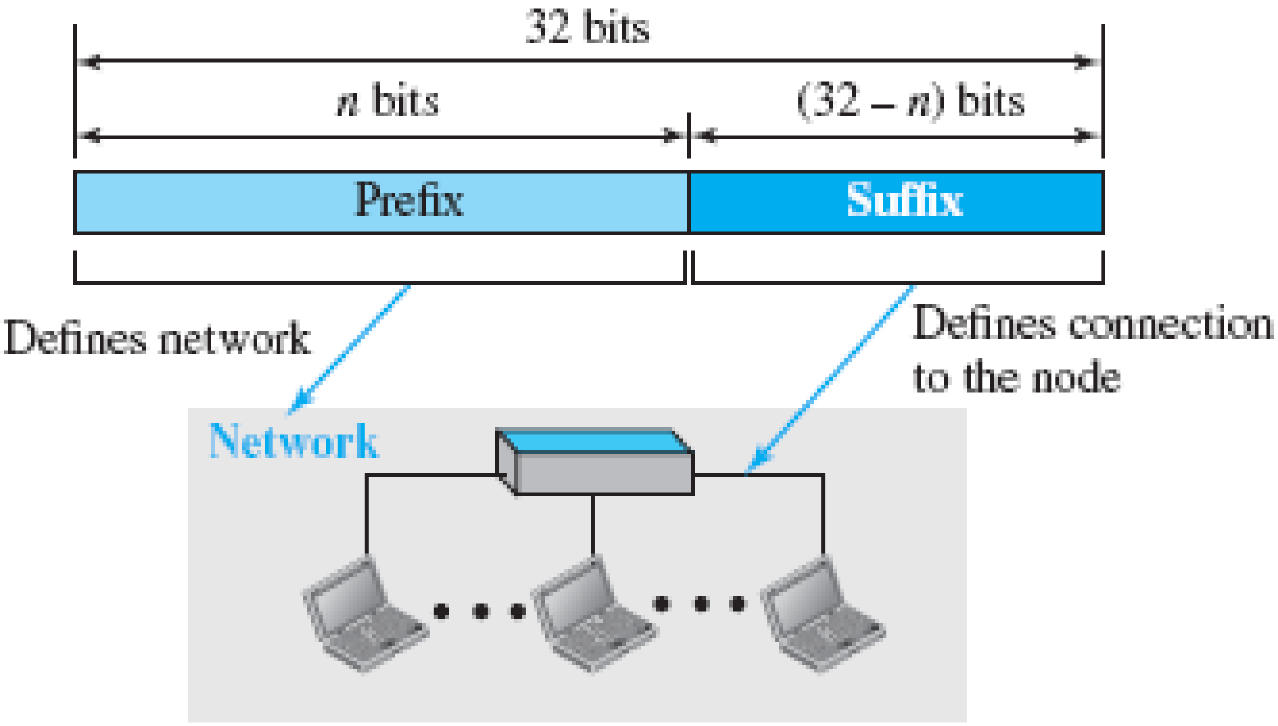


Figure 18.17 Hieararchy in
Addressing

IPv4 ADDRESSES

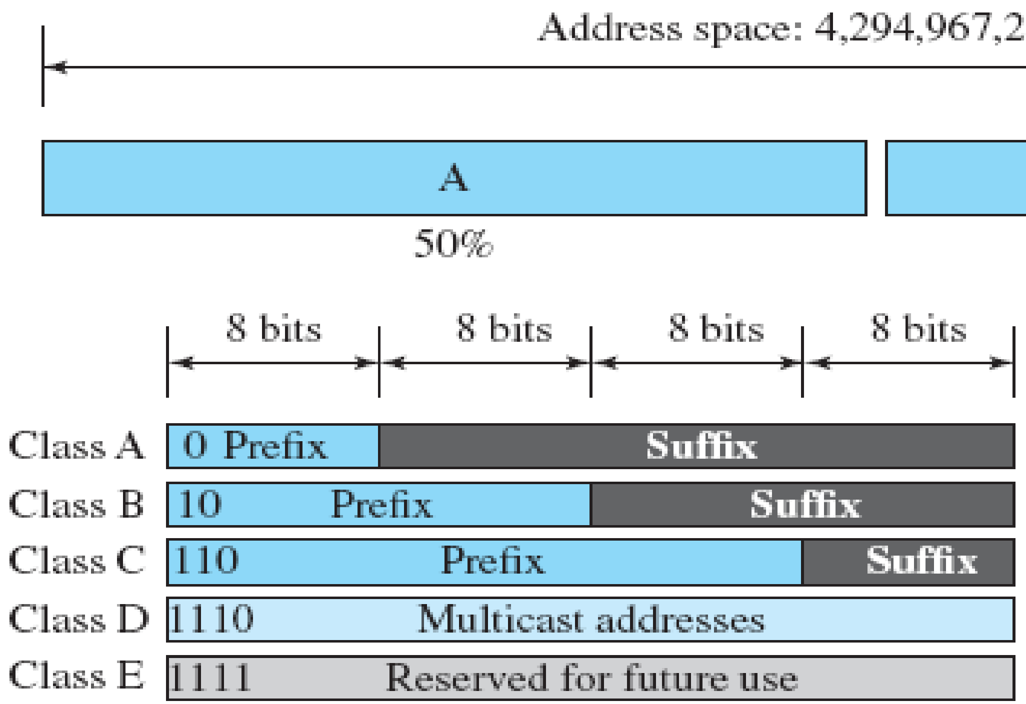
Classful Addressing

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

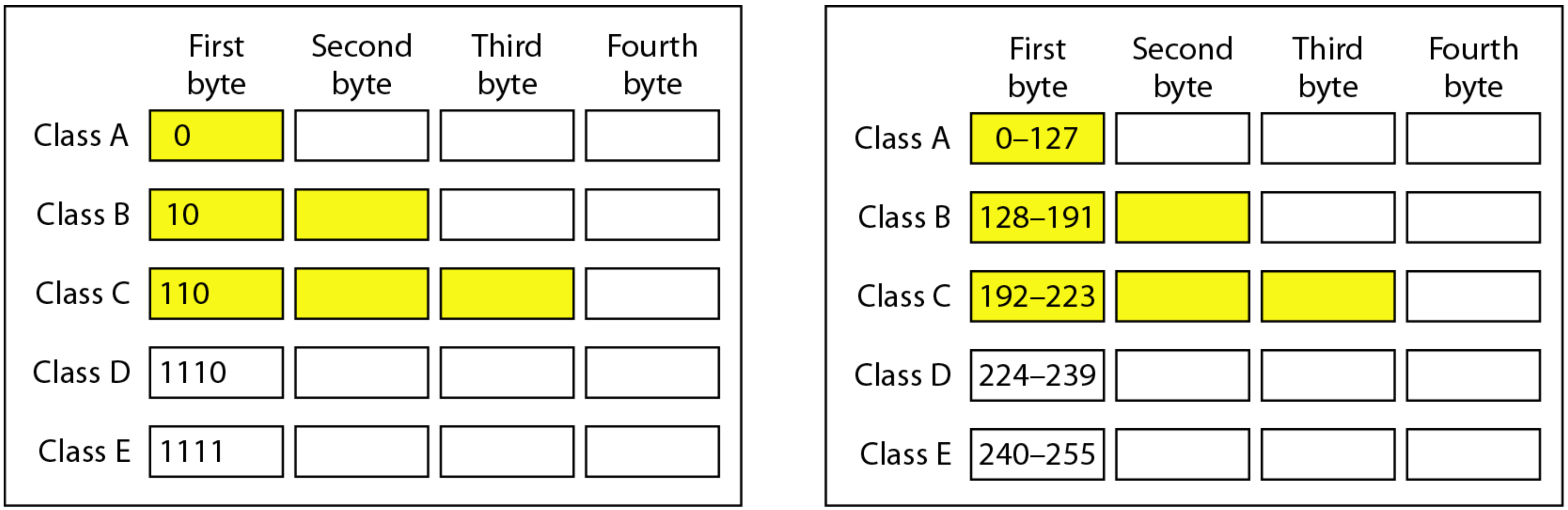
Classful Addressing

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

Finding the classes in binary and dotted-decimal notation



a. Binary notation

b. Dotted-decimal notation

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.**
- In class B, the network length is 16 bits, but since the first two bits, which are $(10)_2$, define the class, we can have only 14 bits as the network identifier.
- This means there are only $2^{14} = 16{,}384$ networks in the world that can have a class B address.
- All addresses that start with $(110)_2$ belong to class C.

- In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier.
- This means there are 2^{21} = 2,097,152 networks in the world that can have a class C address.
- Class D is not divided into prefix and suffix. It is used for multicast addresses.
- All addresses that start with 1111 in binary belong to class E.
- As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

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.

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

Class 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

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

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

- when an organization requested a block of addresses, it was granted one in class A, B, or C.
- Class A addresses were designed for large organizations with a large number of attached hosts or routers.
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.
- Class C addresses were designed for small organizations with a small number of attached hosts or routers.
- 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 (seen by the rest of the world) with 16,777,216 nodes (computers in this single network).

Classful Addressing

Address Depletion (reduction in something)

- To understand the problem, 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.
- Class C addresses
- The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

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. Figure 19.2 shows some netid and hostid bytes.
- the concept 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

Mask

Although the length of the netid and hostid (in bits) is predetermined in classful address ing, we can also use a mask (also called the default mask),

32 bit number is made of contiguous Is followed by contiguous 0s. The masks for classes A, B, and C are shown in Table

Table 19.2 ***Default masks for classful addressing***

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	**255**.0.0.0	/8
B	**11111111 11111111** 00000000 00000000	**255.255**.0.0	/16
C	**11111111 11111111 11111111** 00000000	**255.255.255**.0	/24

- 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 mask in the form In 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 also used in classless addressing

Classful Addressing

Subnetting and Supernetting

- To alleviate address depletion, two strategies were proposed and, to some extent, implemented: **subnetting and supernetting**.
  - In subnetting, a class A or class B block is divided into several subnets. Each subnet has a larger prefix length than the original network.
  - Subnetting increase the number of 1s in the mask
  - 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.

Classful Addressing

- While subnetting was devised to divide a large block into smaller ones,
- several networks are combined to create a super-network or a supemet. 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.
- This idea did not work either because it makes the routing of packets more difficult.
- The organization can then use these addresses to create one supernetwork. Supernetting decreases the number of Is in the mask . , the mask changes from /24 to /22.

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

Classless Addressing

- With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution.
- In this scheme , there are no classes but the addresses are still granted in blocks. Although the long-range solution has already been devised and is called IPv6 (discussed later),
- a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization.

- The short-term solution still uses IPv4 addresses, but it is 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.

Classless Addressing conti

- Address Block

In Classless addressing when entity ,small or large , need to be connected to the internet. It is granted a block(range) of addresses.

The size of the block (the number ofaddresses) varies based on the nature and size of the entity. For example, a household

may be given only two addresses; a large organization may be given thousands of addresses.

Restriction : To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

1.

Classless Addressing

1. The addresses in a block must be contiguous, one after another.

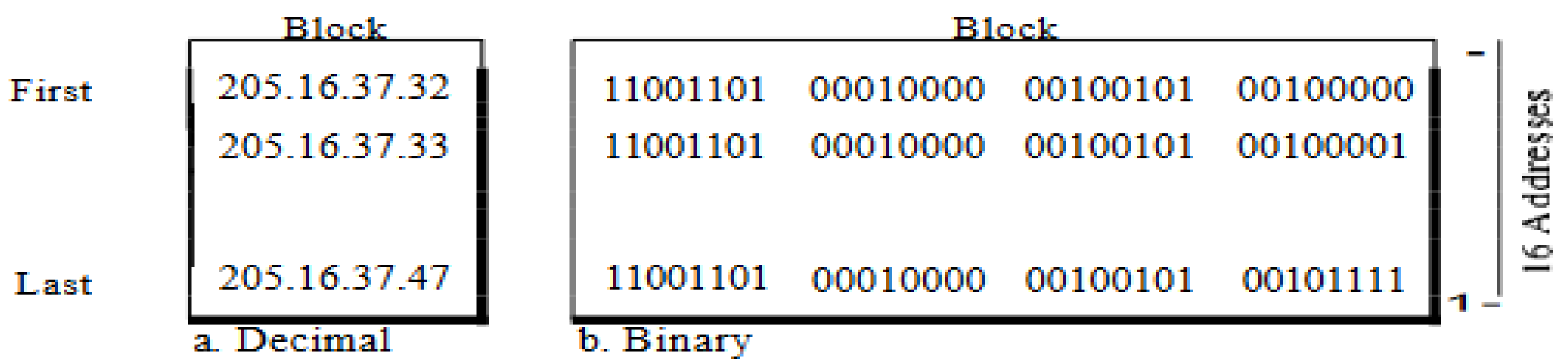
2. The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).

3. The first address must be evenly divisible by the number of addresses

In the figure :

The addresses are contiguous. The number of addresses is a power of 2 (16 = 24), and the first address is divisible by 16.

The first address, when converted to a decimal number, is 3,440,387,360, which when divided by16 results in 215,024,210



Classless Addressing

- In classless addressing, the whole address space is divided into variablelength blocks. The prefix in an address defines the block (network); the suffix defines the node (device).
- Theoretically, we can have a block of 2^0, 2^1, 2^2, . . . , 2^{32} addresses (Has to be in the power of 2).

- A small prefix means a larger network; a large prefix means a smaller network.

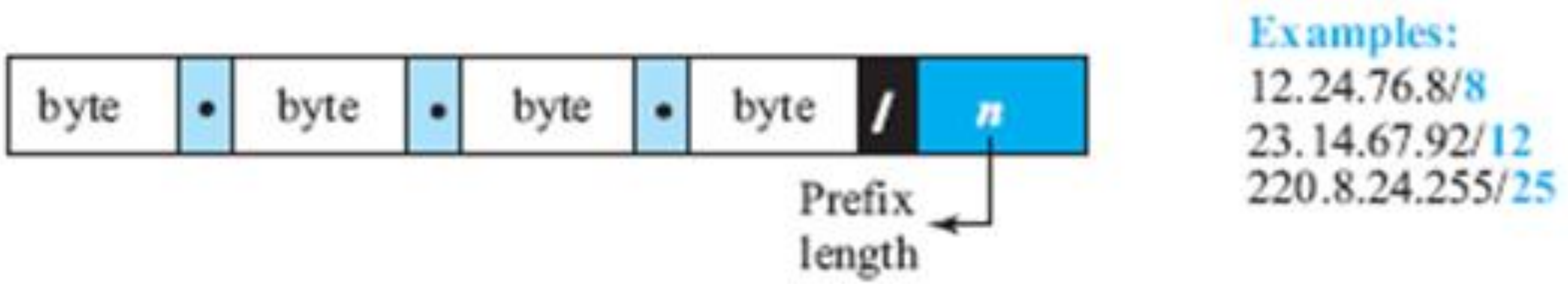
Classless Addressing

<u>Prefix Length: Slash Notation</u>

●How to find the prefix length if an address is given. 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 (pronounced cider) strategy.

●An address in classless addressing can then be represented as Figure



Classless Addressing

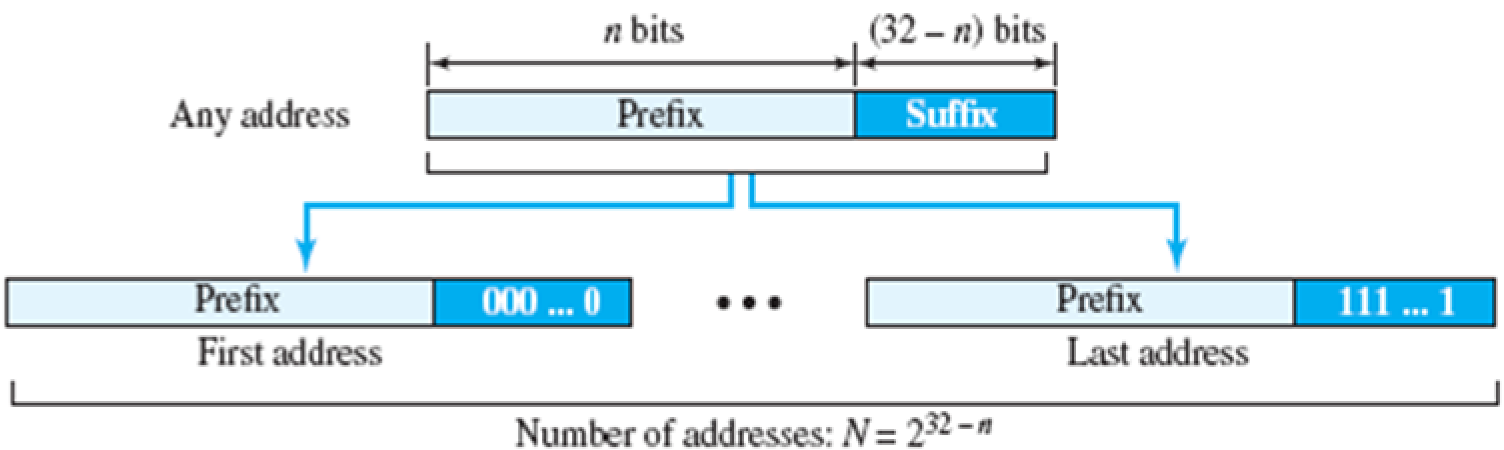
<u>Prefix Length: Slash Notation</u>

●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:

1. the number of addresses,
2. the first address in the block, and
3. the last address.

●Since the value of prefix length, n, is given, we can easily find these three pieces of information, as shown in figure

●



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 $(32 - n)$ right most 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.

Example

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

Example : 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.

Last Address The last address in the block can be found by setting the 32 - 28 right-most bits in the binary notation of the address to Is

11001101 00010000 00100101 00101111. or

205.16.37.47

Example : A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28 Find the number of addresses?

The value of n is 28, which means that number of addresses is

2^{32-28} or 16.

Classless Addressing

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)].

Example : A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. Find first address , last address and no of address using address masking.

Solution

Mask : /28 can be represented as

11111111 11111111 11111111 11110000 (twenty-eight Is and four Os).

A) The first address can be found by AND ing the given addresses with the mask.

Address: 11001101 00010000 00100101 00100111
Mask: 11111111 11111111 11111111 11110000
First Address 11001101 00010000 00100101 0010**0000**

Classless Addressing

B) Last Address : The last address can be found by ORing the given addresses with the complement of the mask.

Mask is :	11111111 11111111 11111111 11110000
Mask complement :	00000000 00000000 00000000 00001111

Now the OR operation

Address:	11001101 00010000 00100101 00100111
Mask complement:	00000000 00000000 00000000 00001111
Last address:	11001101 00010000 00100101 0010**1111**

C) No of Address

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

- Mask complement: 000000000 00000000 00000000 00001111
- Number of addresses: 15 + 1 =16

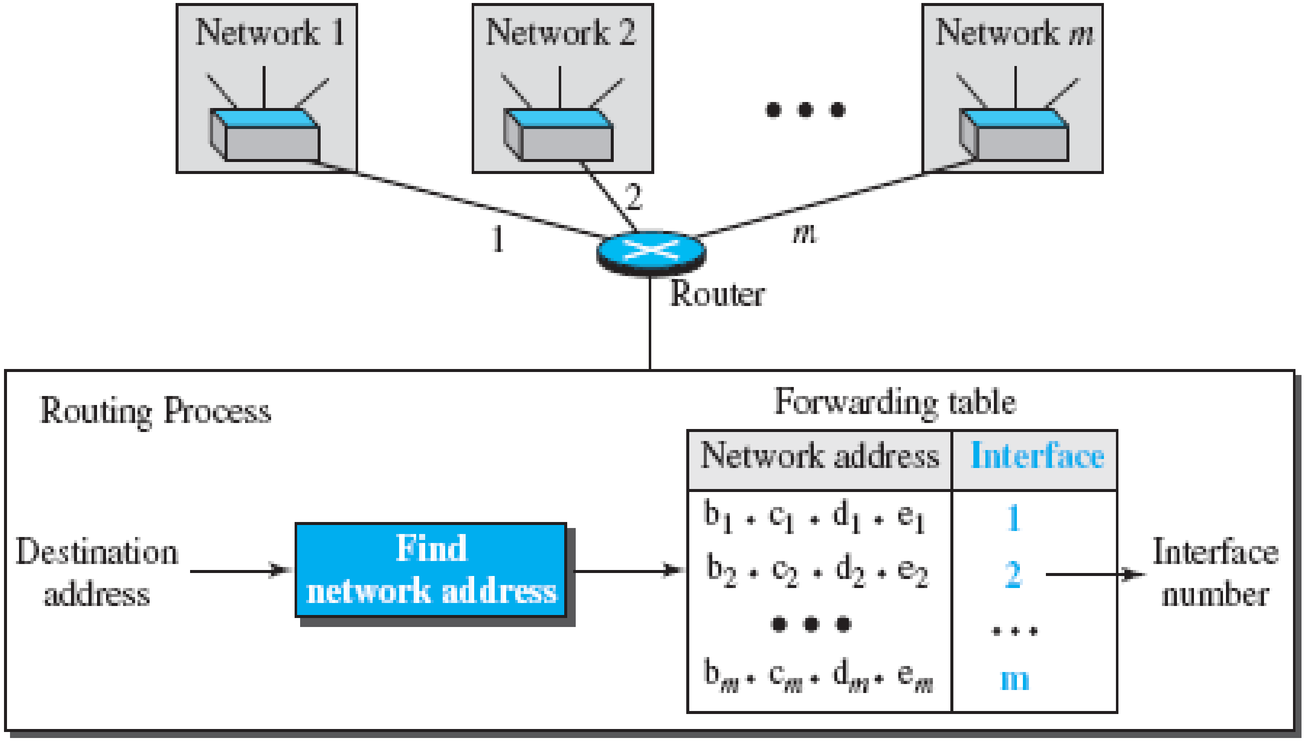
Classless Addressing : Network Address

- Network address 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.
- Figure shows the idea. After the network address has been found, the router consults 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.

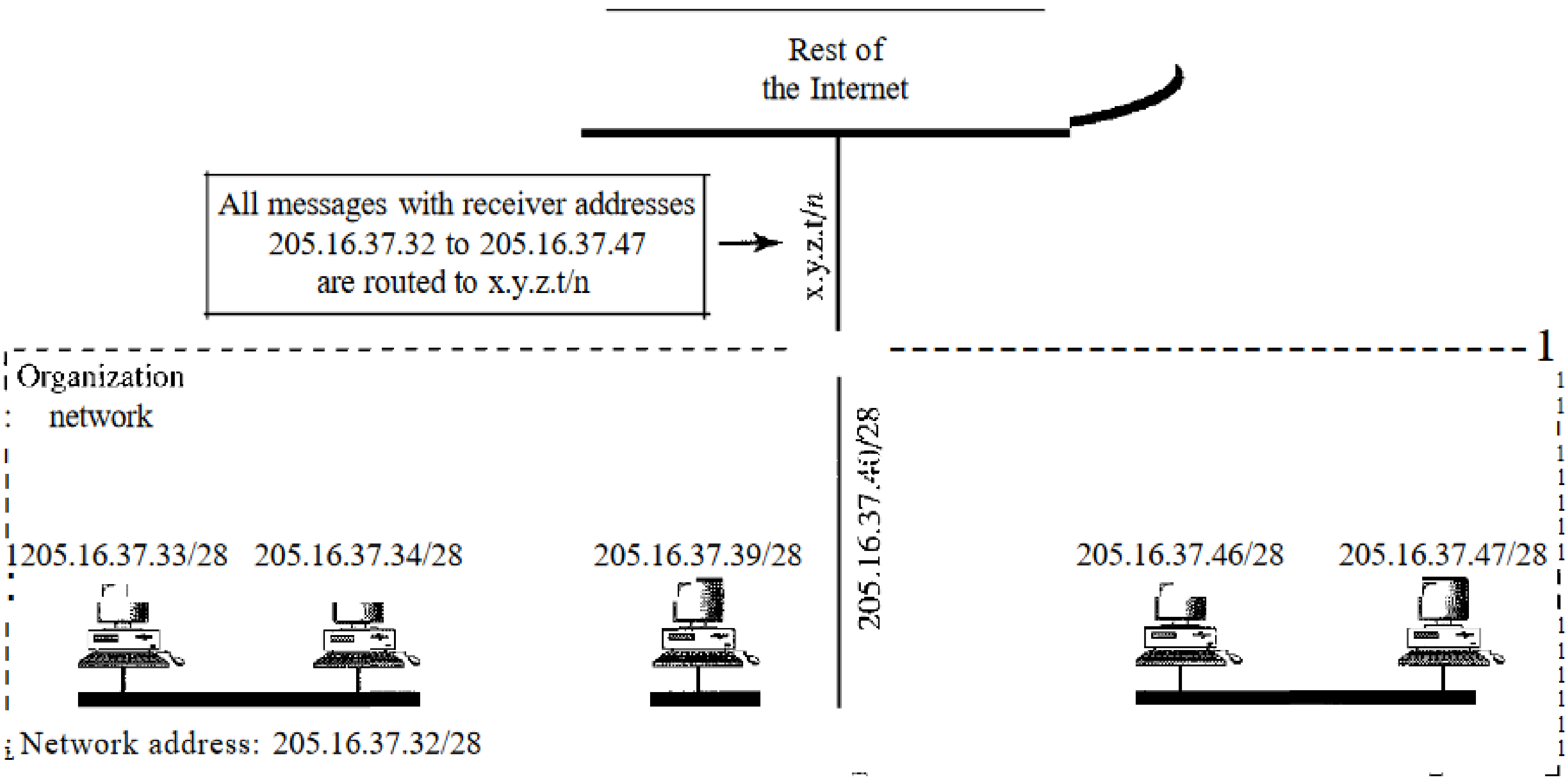
Classless Addressing

Network Address

Figure 18.22 *Network address*



A network configuration for the block 205.16.37.32/28



The organization network is connected to the Internet via a router. The router has two addresses.

One belongs to the granted block; the other belongs to the network that is at the other side of the router.

We call the second address x.y.z.t/n because we do not know anything about the network it is connected to at the other side. All messages destined for addresses in the organization block (205.16.37.32 to 205.16.37.47) are sent, directly or indirectly, to x.y.z.t/n

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.

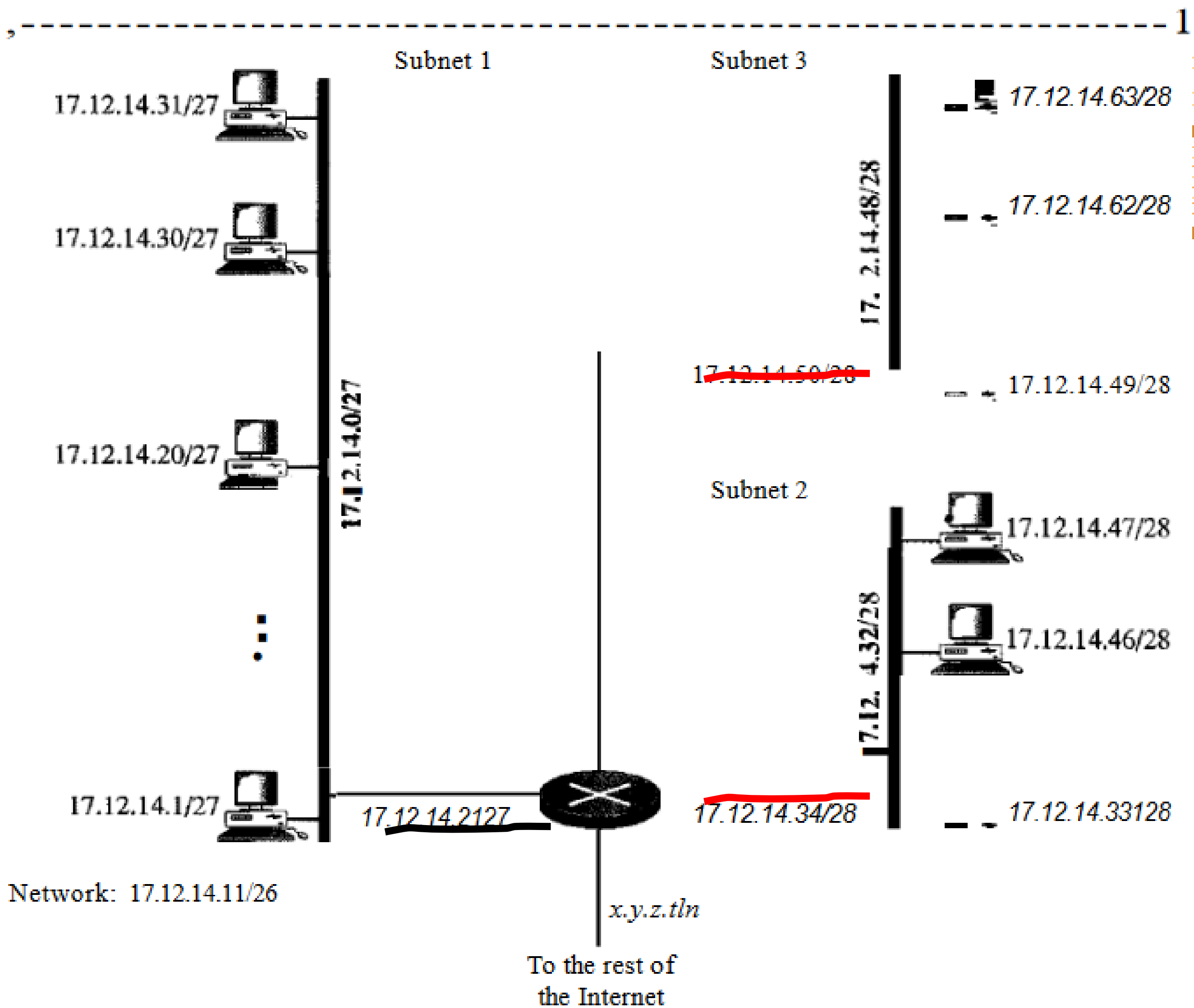
- Hierarchy
  - Ip address define the two level hierarchy when not subnetted.
  - The n left most bit define the network(organization network) and rest rightmost bit (32- n) define the particular host.
- Three level hierarchy (subnetting)
  - An organization that is granted a large block of addresses may want to create clusters of
  - networks (called subnets) and divide the addresses between the different subnets
  - The rest of the world still sees the organization as one entity; however, internally there are several subnets

- All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.
- The organization, however, needs to create small subblocks of addresses, each assigned to specific subnets
- suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses.
- The organization has three offices and needs to divide the addresses into three subblocks of 32, 16, and 16 addresses

1. Suppose the mask for the first subnet is n1, then 2^{32-n1} must be 32, which means that n1 = 27.
2. Suppose the mask for the second subnet is n2, then 2^{32-n2} must be 16, which means that n2 = 28.
3. Suppose the mask for the third subnet is n3, then 2^{32-n3} must be 16, which means that n3 = 28.

That is subnet have mask 27, 28,28 and organization mask is 26.

Figure 19.7 *Configuration and addresses in a subnetted network*



Subnet 1 : suppose the address 17.12.14.29/27

Then it can give us the subnet address one

Host: 00010001 00001100 00001110 00011101

Mask /27

Subnet : 00010001 00001100 00001110 00000000 .... (17.12.14.0

Subnet2 : The address 17.12.14.45/28 can give us the address of subnet

Host: 00010001 00001100 00001110 00101101

Mask /28

Subnet : 00010001 00001100 00001110 00100000 .... (17.12.32.0

Subnet3 : The address 17.12.14.50/28 can give us the subnet address

Host: 00010001 00001100 00001110 00110010

Mask /28

Subnet : 00010001 00001100 00001110 00110000 .... (17.12.48.0

- ***Network Address Block Allocation***
  - The main responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN).
  - ICANN does not allocate addresses to individual Internet users but to an ISP (or a larger organization that is considered an ISP in this case).
  - Each ISP, divides its assigned block into smaller subblocks and grants the subblocks to its customers.
  - ISP receives one large block to be distributed to its Internet users.

For the proper operation of the CIDR, two restrictions need to be applied to the allocated block.

1. The number of requested addresses, N, needs to be a power of 2. The reason is that $N = 2^{32-n}$ or $n = 32 - \log_2 N$. If N is not a power of 2, we cannot have an integer value for n.

2. The requested block needs to be allocated where there is an adequate number of contiguous addresses available in the address space.

However, there is a restriction that the first address needs to be divisible by the number of addresses in the block.

The reason is that the first address needs to be the prefix followed by $(32 - n)$ number of 0s. The decimal value of the first address is then,

$$\text{first address} = (\text{prefix in decimal}) \times 2^{32-n} = (\text{prefix in decimal})$$
$$N$$

Example 19.10

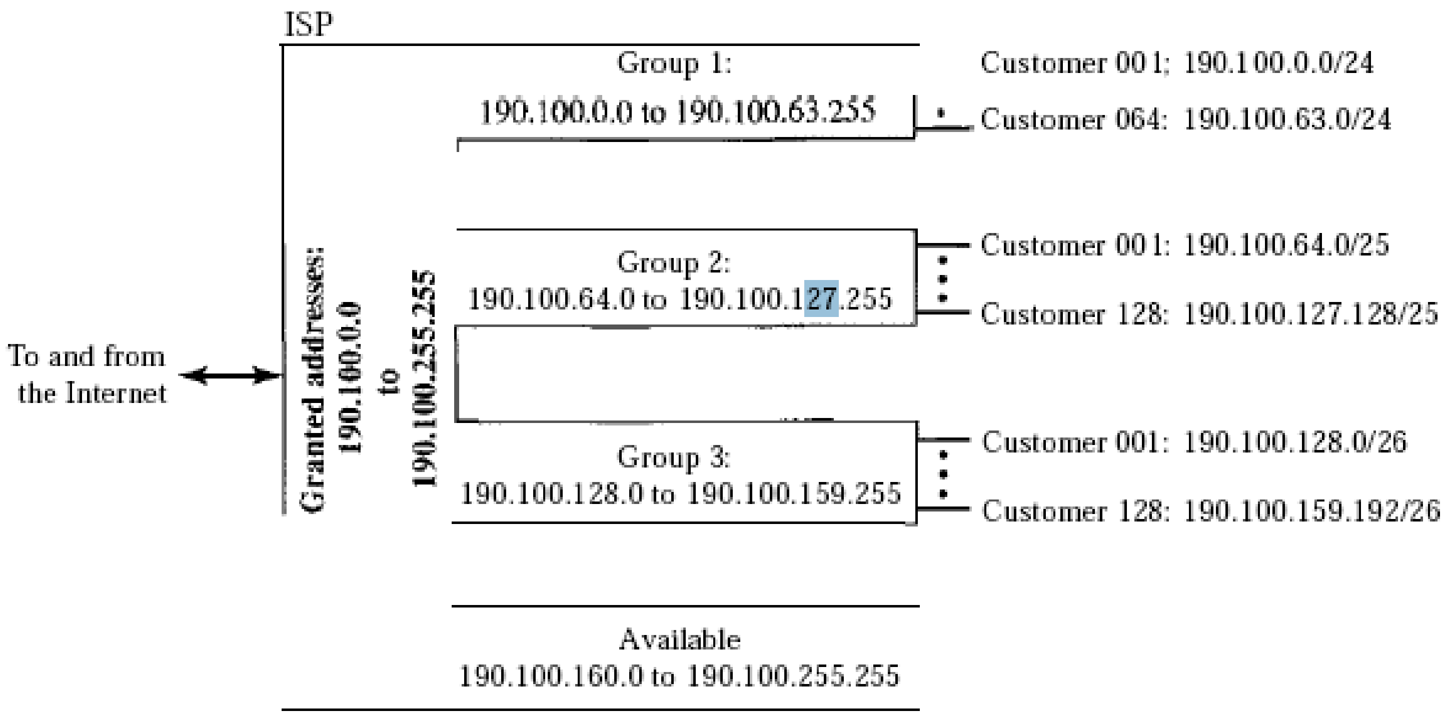
An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.

b. The second group has 128 customers; each needs 128 addresses.

c. The third group has 128 customers; each needs 64 addresses.

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



For group 1 :

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

1st Customer: 190.100.0.0/24 to 190.100.0.255/24

2nd Customer: 190.100.1.0/24 to 190.100.1.255/24

64th Customer: 190.100.63.0/24 190.100.63.255/24

Total = 64 X 256 =16,384

For group 2 :

or this group, each customer needs 128 addresses. This means that 7 (10g$_2$ 128) bits are needed to define each host. The prefix length is then 32 - 7 = 25. The addresses are

1st Customer: 190.100.64.0/25 to 190.100.64.127/25

2nd Customer: 190.100.64.128/25 to 190.100.64.255/25

128th Customer: 190.100.127.128/25 190.100.127.255/25

Total = 128 X 128 = 16,384

For group 3 :

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

1st Customer: 190.100.128.0/26 to 190.100.128.64/26

2nd Customer: 190.100.128.63/26 to 190.100.128.127/26

128th Customer: 190.100.159.192/26 190.100.159.255/26

Total = 128 X 64 = 8192

Number of granted addresses to the ISP: 65,536

Number of allocated addresses by the ISP: 40,960

Number of available addresses: 24,576

NAT- Network Address Translation

- The number of home users and small businesses that want to use the Internet is ever increasing.
- In the beginning, a user was connected to the Internet with a dial-up line. An ISP with a block of addresses could dynamically assign an address to this user.
- Home users and small businesses can be connected by an ADSL line or cable modem. In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host.
- Having shortage of addresses, is a serious problem. A quick solution to this problem is called 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.
- 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, shown in Table 19.3.

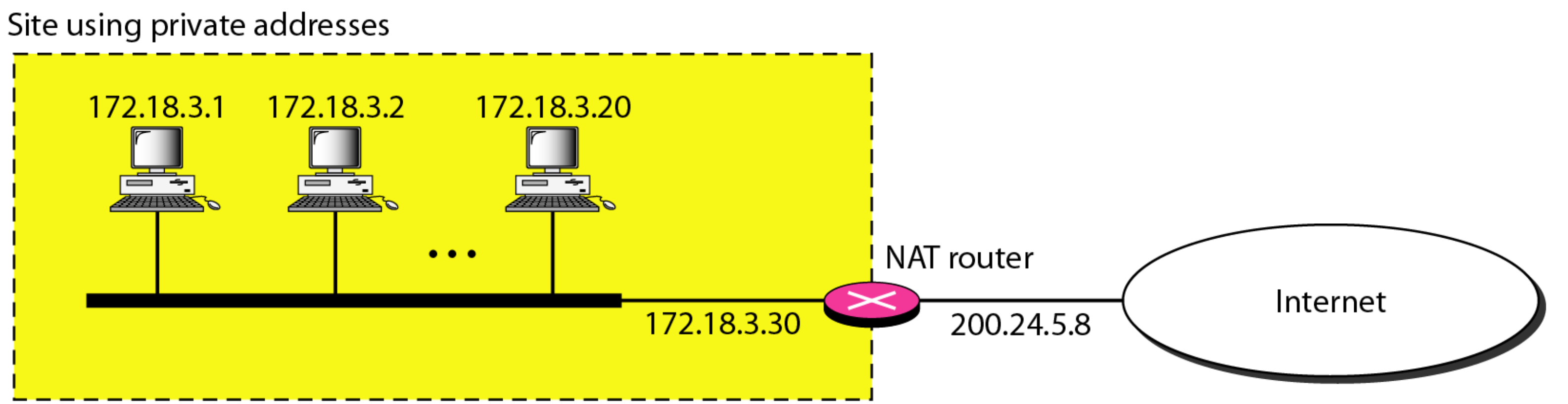
NAT- Network Address Translation

Table 19.3 *Addresses for private networks*

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}

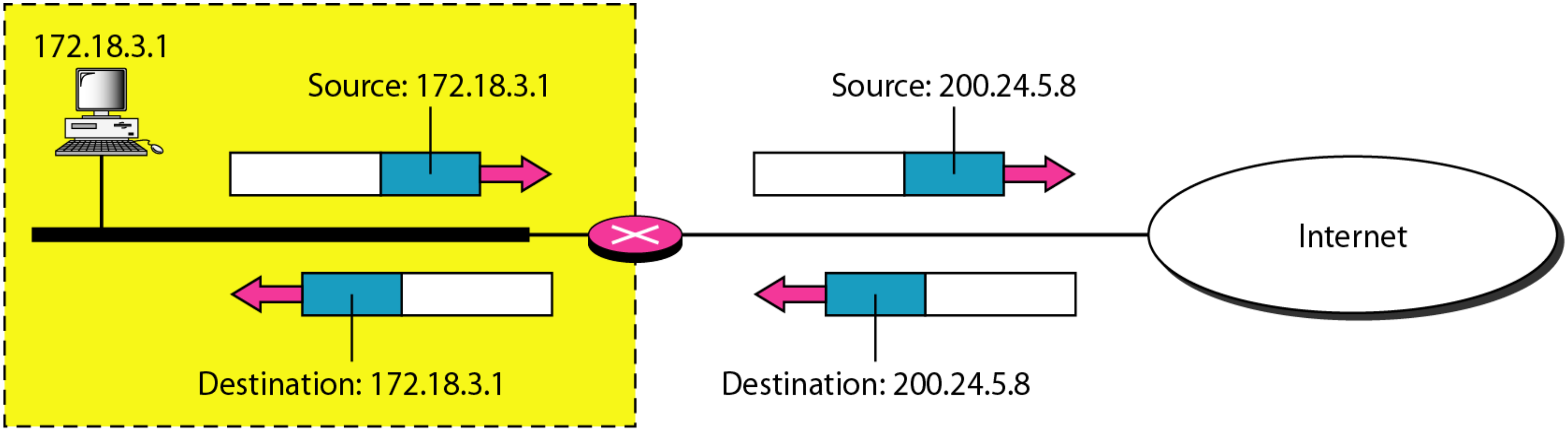
- Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private networks.
- They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these 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.

Figure 19.10 *A NAT implementation*



- 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 transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Figure 19.11 *Address translation*



Site using private addresses

- 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 19.12 *NAT address translation*

200.24.8.5 is public address of Nat router which connect to the internet

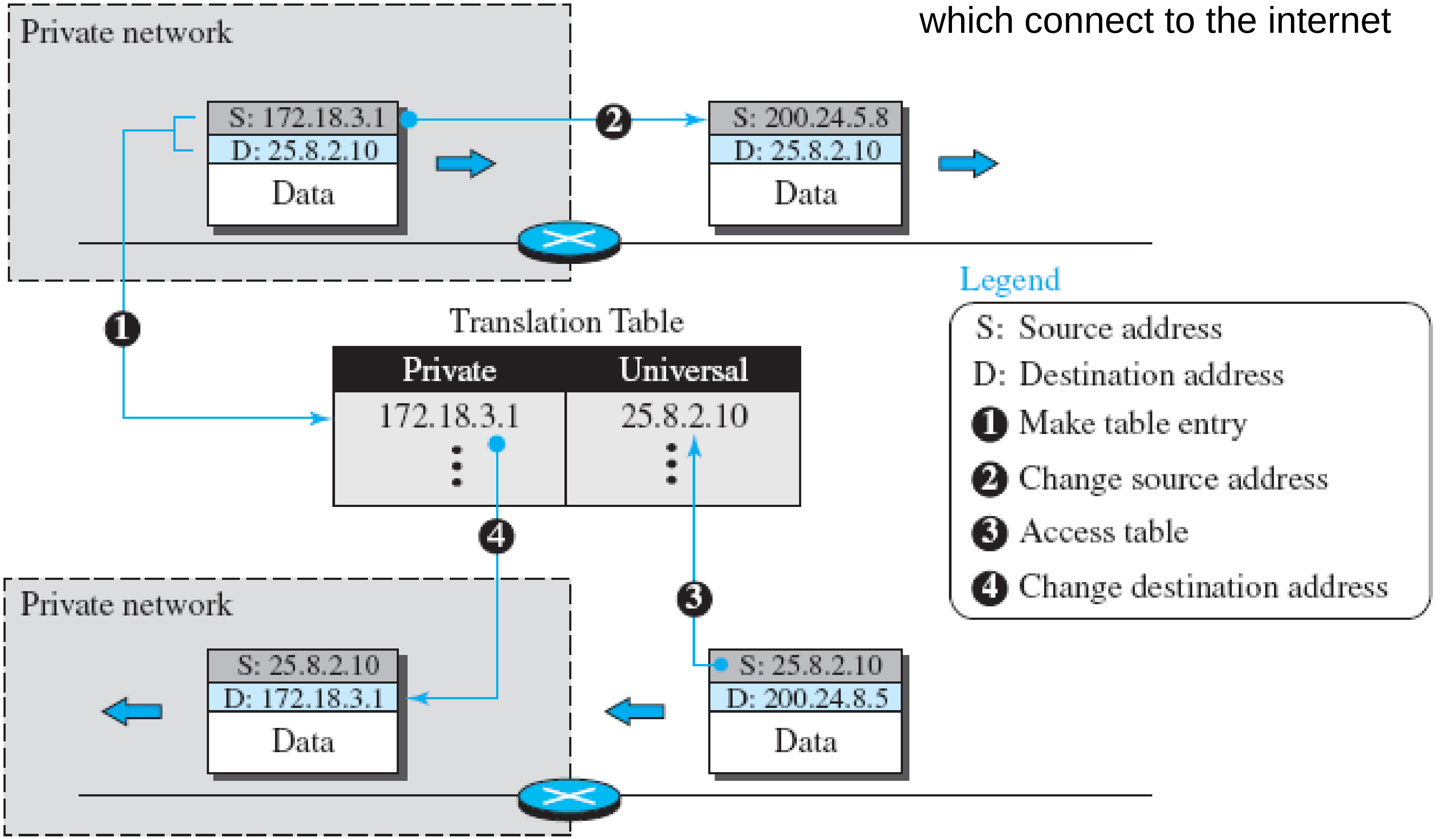


Figure 19.12 *NAT address translation*

- Translating the source addresses for outgoing packets is straightforward.
- **But how does the NAT router know the destination address for a packet coming from the Internet?**
- There may be tens or hundreds of private IP addresses, each belonging to one specific host. The problem is solved if the NAT router has a translation table
- A translation table has only two columns: the private' address and the external address (destination address of the packet).
- When the router translates the source address of the outgoing packet, it also makes note of the destination address-where the packet is going.
- When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet. Figure 19.12 shows the idea.
- Note that the addresses that are changed (translated) are shown in color.

Using a Pool of IP Addresses

- Since the NAT router has only one global address, only one private network host can access the same external host.
- To remove this restriction, the NAT router uses a pool of global addresses.
- For example, instead of using only one global address (200.24.5.8), the NAT router can use four addresses (200.24.5.8, 200.24.5.9, 200.24.5.10, and 200.24.5.11).
- In this case, four private network hosts can communicate with the same external host at the same time because each pair of addresses defines a connection. However, there are still some drawbacks.
- In this example, no more than four connections can be made to the same destination. Also, **no private-network host can access two external server programs (e.g, HTTP and FTP) at the same time.**

Using both IP and port

- For example, suppose two hosts with addresses 172.18.3.1 and 172.18.3.2 inside a private network need to access the HTTP server on external host 25.8.3.2.
- If the translation table has five columns, instead of two, that include the source and destination port numbers of the transport layer protocol, the ambiguity is eliminated.

- **When the response from HTTP comes back, the combination of source address (25.8.3.2) and destination port number (1400) defines the-private network host to which the response should be directed.** Note also that for this translation to work, the temporary port numbers (1400 and 1401) must be unique.

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
. . .	. . .	. . .	. . .	. . .

19-2 IPv6 ADDRESSES

- The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4.
- An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.
- 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.
- An IPv6 address is 128-bits long.

19-2 IPv6 ADDRESSES

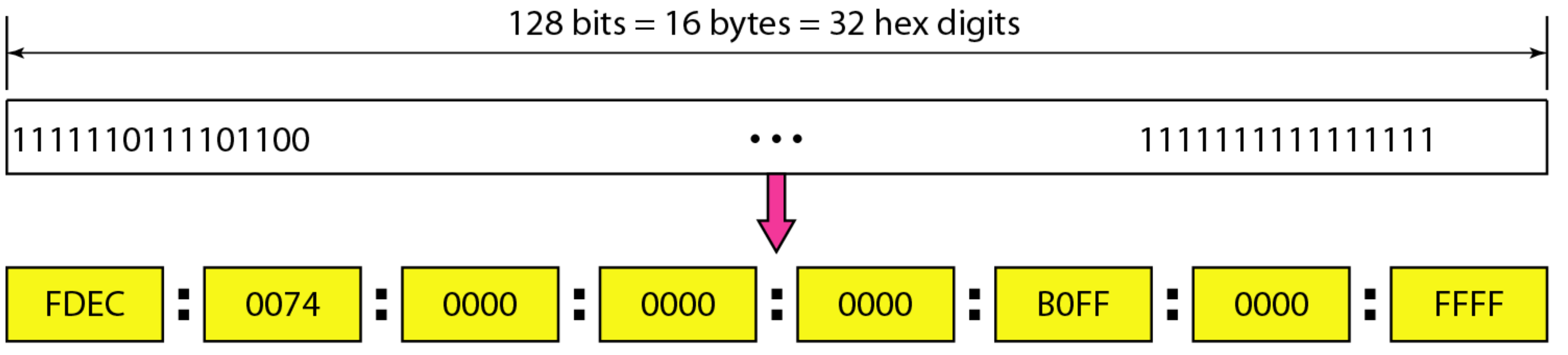
Representation

- A computer normally stores the address in binary, but it is clear that 128 bits cannot easily be handled by humans.
- Several notations have been proposed to represent IPv6 addresses when they are handled by humans. The following shows two of these notations: binary and colon hexadecimal.

Binary (128 bits)	1111111011110110	...	1111111100000000
Colon Hexadecimal	FEF6:BA98:7654:3210:ADEF:BBFF:2922:FF00		

- Binary notation is used when the addresses are stored in a computer. The colon hexadecimal notation (or colon hex for short) divides the address into eight sections, each made of four hexadecimal digits separated by colons.

IPv6 address in binary and hexadecimal colon notation



19-2 IPv6 ADDRESSES

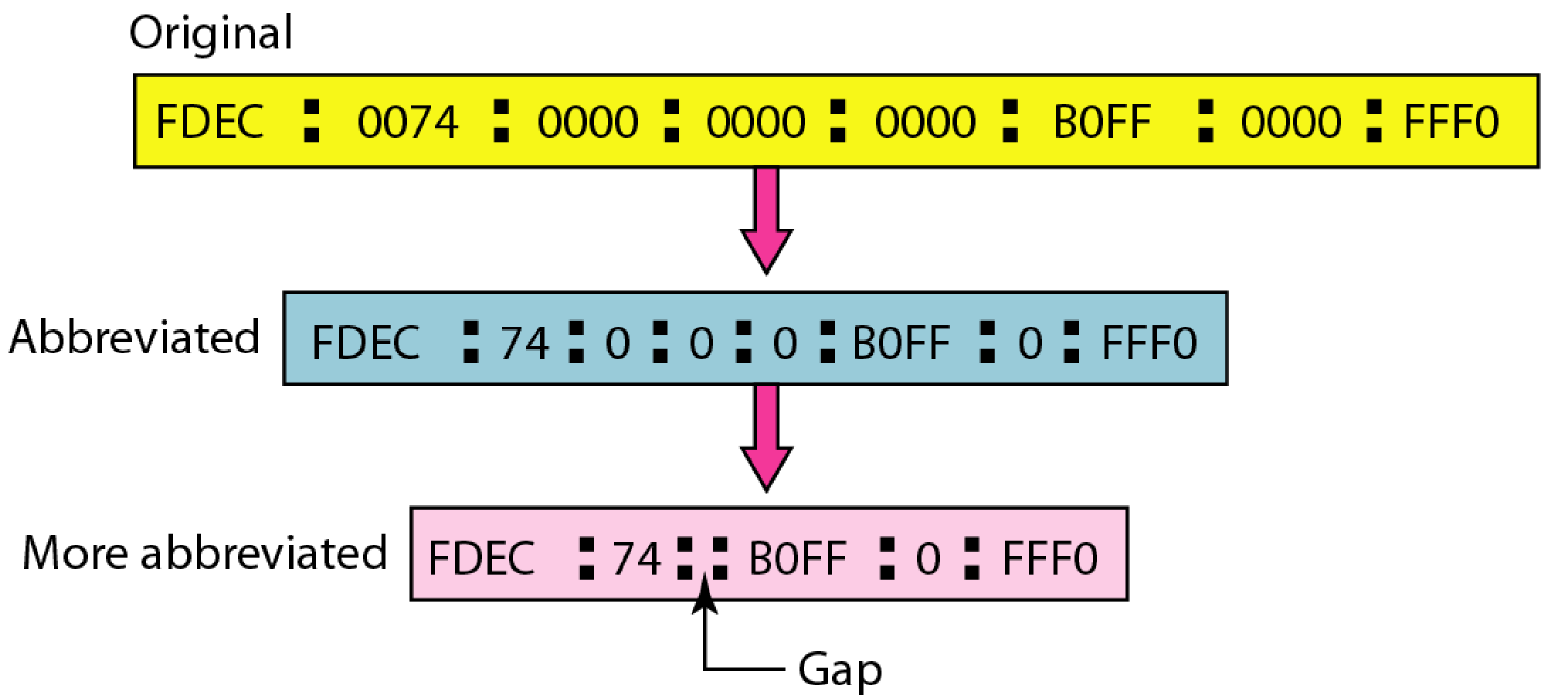
Abbreviation

- Although an IPv6 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 can be omitted. Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0.
- Note that 3210 cannot be abbreviated. Further abbreviation, often called **zero compression, can be applied to colon hex notation if there are consecutive sections consisting of zeros only.**
- We can remove all the zeros and replace them with a **double semicolon.**

FDEC:0:0:0:0:BBFF:0:FFFF → FDEC::BBFF:0:FFFF

- Note that this type of abbreviation is allowed only once per address. If there is more than one run of zero sections, only one of them can be compressed.

Abbreviated IPv6 addresses



19-2 IPv6 ADDRESSES

Address Space

- IPv6 has a much larger address space; 2^{128} addresses are available. The designers of IPv6 divided the address into several categories.
- **A few leftmost bits, called the type prefix,** in each address define its category. **The type prefix is variable in length**, but it is unique form other code.
- Table 19.5 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space. (d what fraction of the total address space each represents.)

19-2 IPv6 ADDRESSES

Table 19.5 *Type prefixes for 1Pv6 addresses*

Type Prefix	*Type*	*Fraction*
00000000	Reserved	1/256
00000001	Unassigned	1/256
0000001	ISO network addresses	1/128
0000010	IPX (Novell) network addresses	1/128
0000011	Unassigned	1/128
00001	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	Provider-based unicast addresses	1/8

19-2 IPv6 ADDRESSES

Table 19.5 *Type prefixes for IPv6 addresses (continued)*

Type Prefix	*Type*	*Fraction*
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1116
11110	Unassigned	1132
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
11111110 a	Unassigned	1/512
1111 111010	Link local addresses	111024
1111 1110 11	Site local addresses	1/1024
11111111	Multicast addresses	1/256

19-2 IPv6 ADDRESSES

<u>Unicast Addresses</u>

- 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.

<u>Multicast Addresses</u>

- 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.

<u>Anycast Addresses</u>

- 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

19-2 IPv6 ADDRESSES

Reserved Addresses

- Another category in the address space is the reserved address. These addresses start with eight 0s (type prefix is 00000000).

Local Addresses

- 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. Two types of Logical addresses are there: **Link Local and Site Local.**

Need for Network Layer

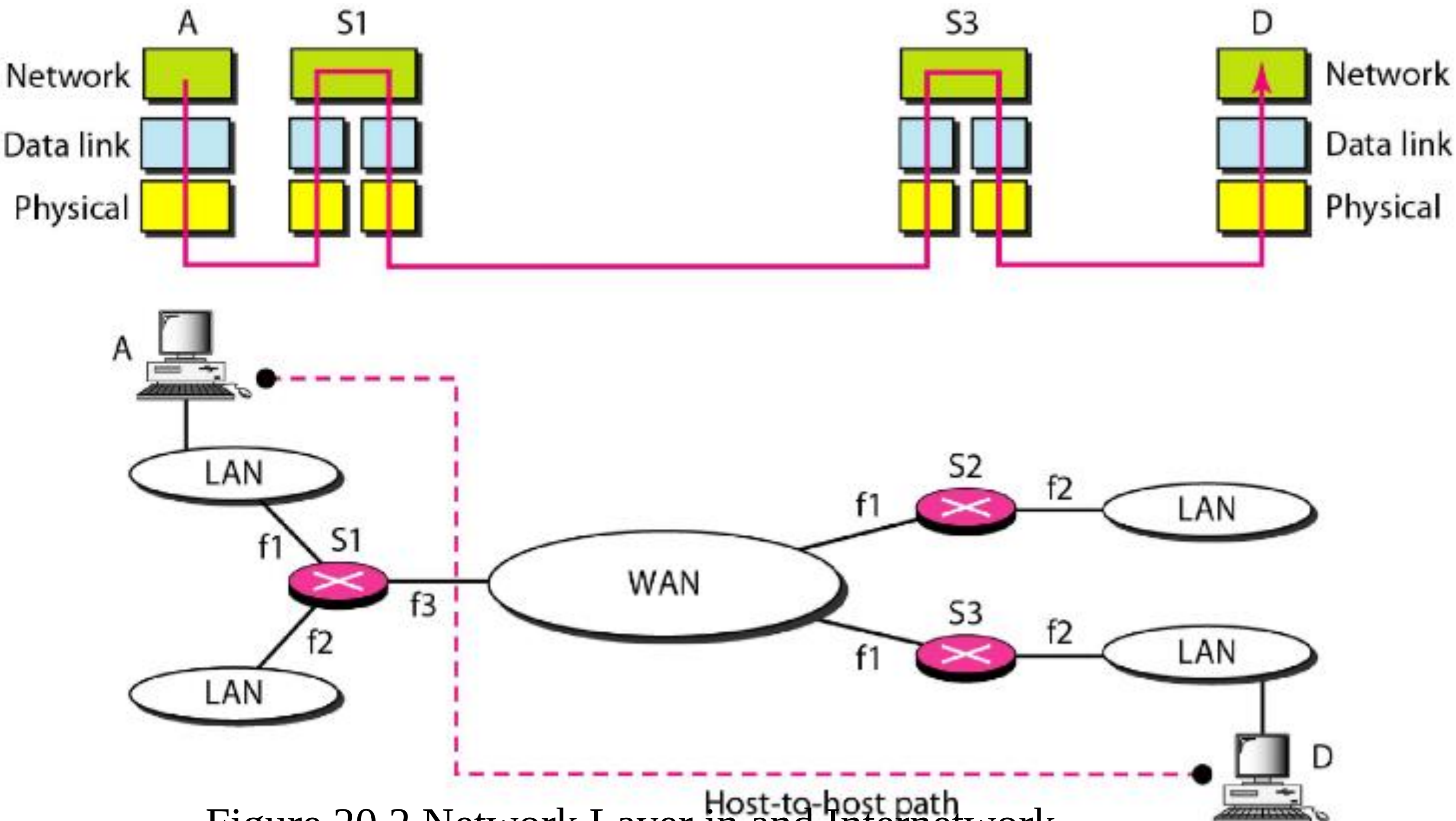


Figure 20.2 Network Layer in and Internetwork

Network Layer

- The network layer at the source is responsible for creating a packet from the data coming from another protocol (such as a transport layer protocol or a routing protocol).
- The header of the packet contains, among other information, the logical addresses of the source and destination.
- **The network layer is responsible for checking its routing table** to find the routing information .If the packet is too large, the packet is fragmented

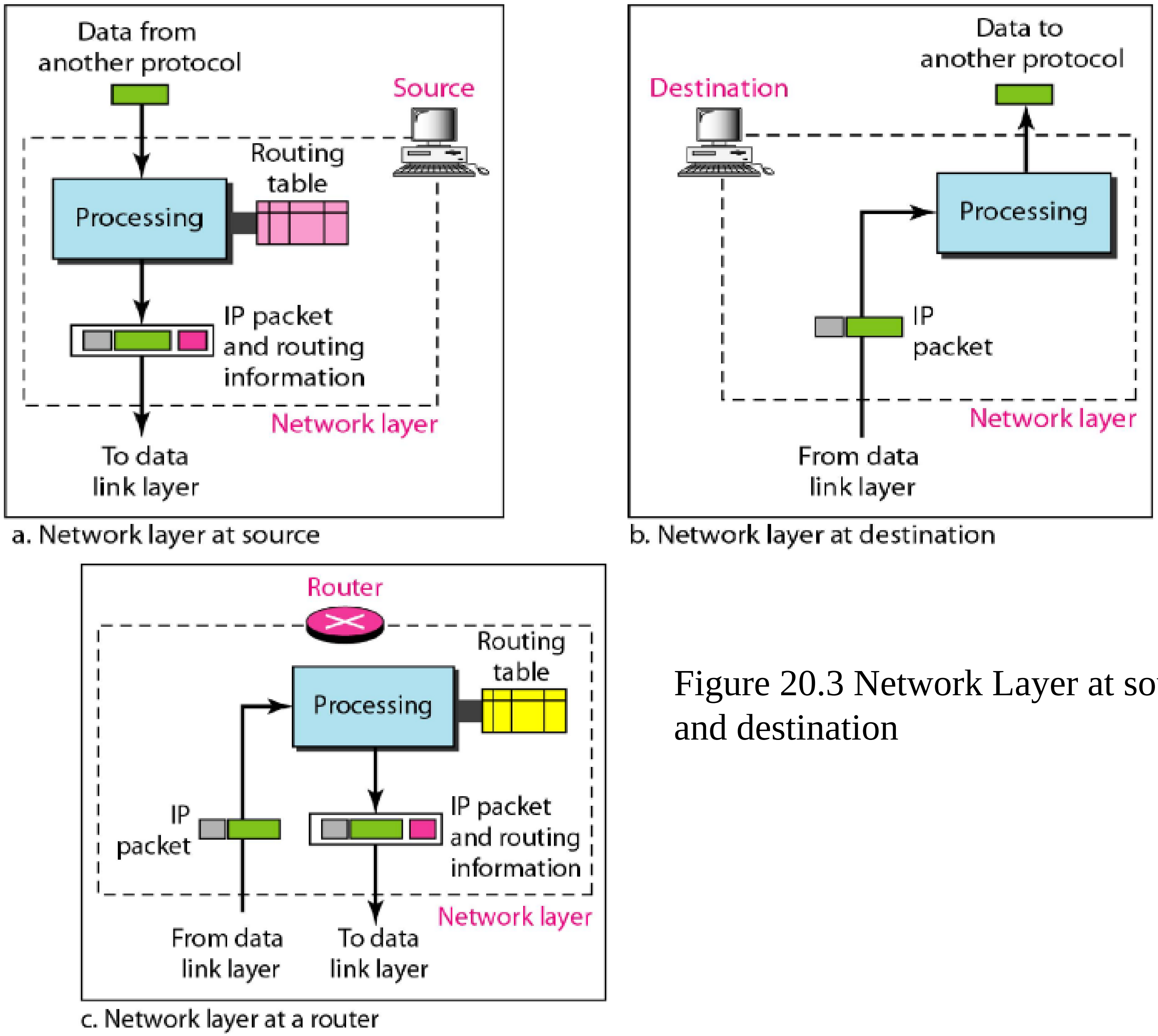


Figure 20.3 Network Layer at source, router and destination

Need for Network Layer

- The network layer at the switch or router is responsible for routing the packet.
- When a packet arrives, the router or switch consults its routing table and finds the interface from which the packet must be sent.
- The packet, after some changes in the header, with the routing information is passed to the data link layer again.
- The network layer at the destination is responsible for address verification; it makes sure that the destination address on the packet is the same as the address of the host.
- If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.

Internet as a Datagram Network

- Switching can be divided into three broad categories:
- Switching is transfer packet thorough network switch.
- circuit switching, packet switching, and message switching.

- Packet switching uses either the virtual circuit approach or the datagram approach.

- The Internet has chosen the datagram approach to switching in the network layer.
- It uses the universal addresses defined in the network layer to route packets from the source to the destination.

Internet as a Connectionless Network(Data Gram)

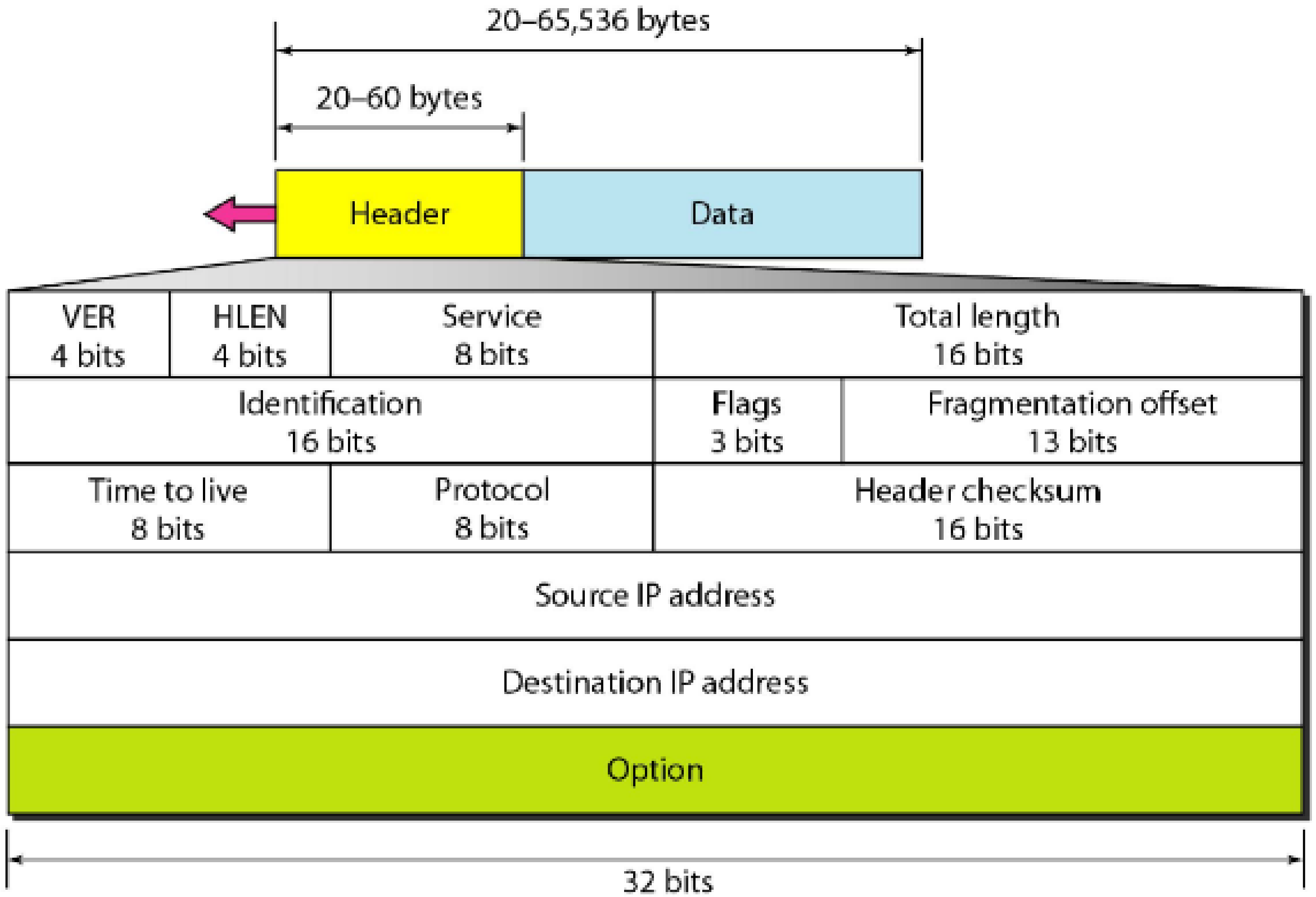
- Delivery of a packet can be accomplished by using either a **connection-oriented or a connectionless network service.**
- **Connection-oriented service:** the source first makes a connection with the destination before sending a packet. In this case, there is a relationship between packets. They are sent on the same path in sequential order.
- When all packets of a message have been delivered, the connection is terminated.
- The decision about the **route** of a sequence of packets with the same source and destination addresses can be made **only once,** when the connection is established.
- Switches do not recalculate the route for each individual packet. This type of service is used in a virtual-circuit approach.

Internet as a Connectionless Network(datagram)

- **In conneectionless service,** the network layer protocol treats each packet independently, with each packet having no relationship to any other packet.
- The packets in a message **may or may not travel the same path** to their destination.
- This type of service is used in the datagram approach to packet switching. The Internet has chosen this type of service at the network layer.
- The reason is that the Internet is made of so many heterogeneous networks that it is almost impossible to create a connection from the source to the destination without knowing the nature of the networks in advance.

Datagram Format(Ipv4)

- Packets in the IPv4 layer are called datagrams.
- Following **Figure** shows the IPv4 datagram format.



Datagram

- 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.
- header in 4-byte sections as follows:

- Version (VER): This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPv6) may totally replace version 4 in the future.

- Header length (HLEN): 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 size, the value of this field is 15 (15 x 4 = 60).
- Services: this 8-bit field. This field, previously called service type, is now called differentiated services

Datagram

- 1. 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.
- 3 bit precedence bit 0 to 111 define the priority of datagram. If router is congested and need to discard some datagram . Datagram with the lowerst priority discarded first.

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

Datagram

- Application programs can request a specific type of service. The defaults for some applications are shown in Table 20.2.

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

Table 20.2 Default types ofservice

Datagram header Conti..

Total length :This is a In-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes

Identification - This field is used in fragmentation

Flags. This field is used in fragmentation

Fragmentation offset: This field is used in fragmentation

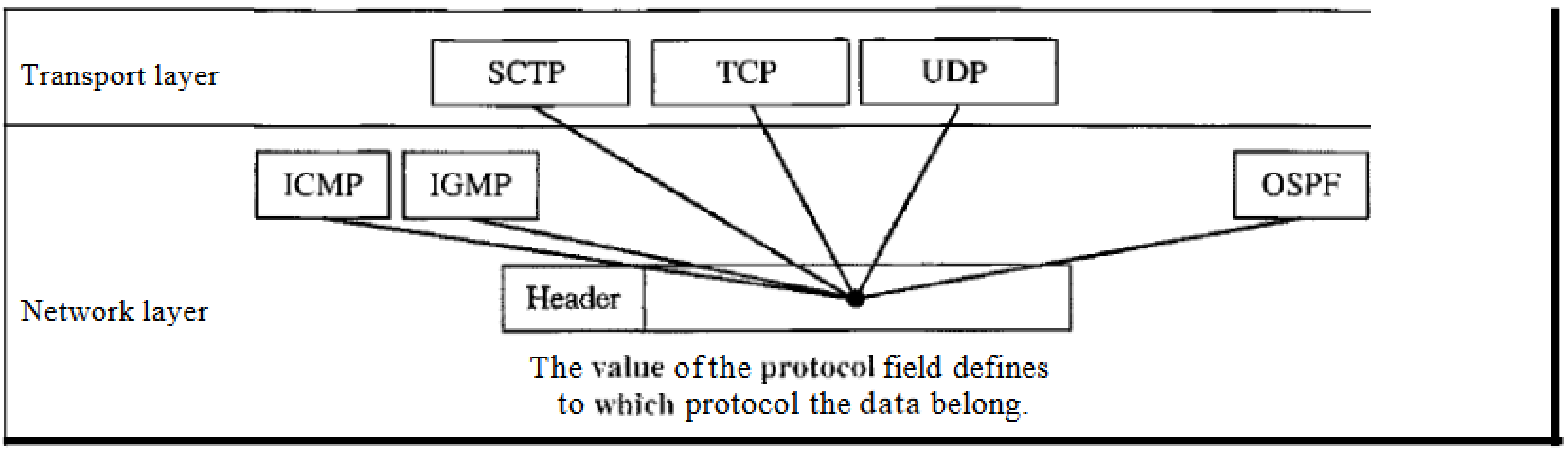
Time to live: A datagram has a limited lifetime in its travel through an internet.

decremented by each visited router. The datagram was discarded when the value became zero

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

Checksum : its calculation

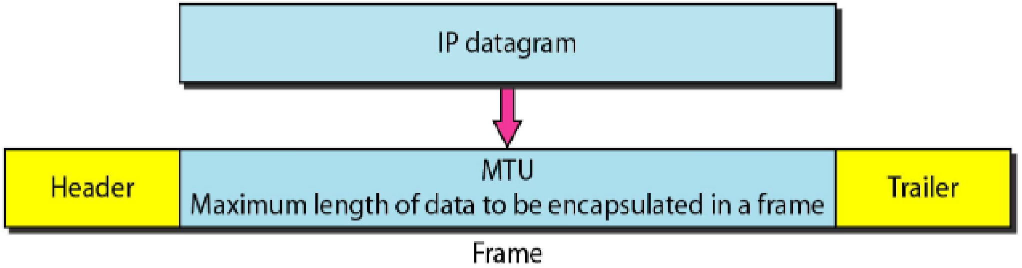


Fragmentation

- 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.
- For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Fragmentation

- Maximum Transfer Unit (MTU)
- Each data link layer protocol has its own frame format in most protocols. One field is the maximum size of the data field, means when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size.



Maximum transfer unit (MTU)

MTUs for some networks

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

Fragmentation

- Maximum Transfer Unit (MTU)
- To make the IPv4 protocol independent of the physical network, the designers decided to make the maximum length of the IPv4 datagram equal to 65,535 bytes.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed.
- A datagram can be fragmented several times before it reaches the final destination.

When a datagram is fragmented, required parts of the header must be copied by all fragments.

- The host or router that fragments a datagram must change the values of three fields: **flags, fragmentation offset, and total length**.
- The rest of the fields must be copied. Of course, the value of the checksum must be recalculated regardless of fragmentation.

- Flags This is a 3-bit field. The first bit is reserved.
  - The second bit is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram.
  - 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.
- Figure 20.11 shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = O.
- The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175.
- Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 =350.

Checksum

- The implementation of the checksum in the IPv4 packet follows the same principles as we have seen before.
- 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 the checksum field.
- The checksum in the IPv4 packet covers only the header, not the data.

- **First**, all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole packet. Therefore, the checksum for the IPv4 datagram does not have to check the encapsulated data.
- **Second**, the header of the IPv4 packet changes with each visited router, but the data do not. So the checksum includes only the part that has changed.
- If the data were included, each router must recalculate the checksum for the whole packet, which means an increase in processing time.

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.

IPv6 protocol

Advantages

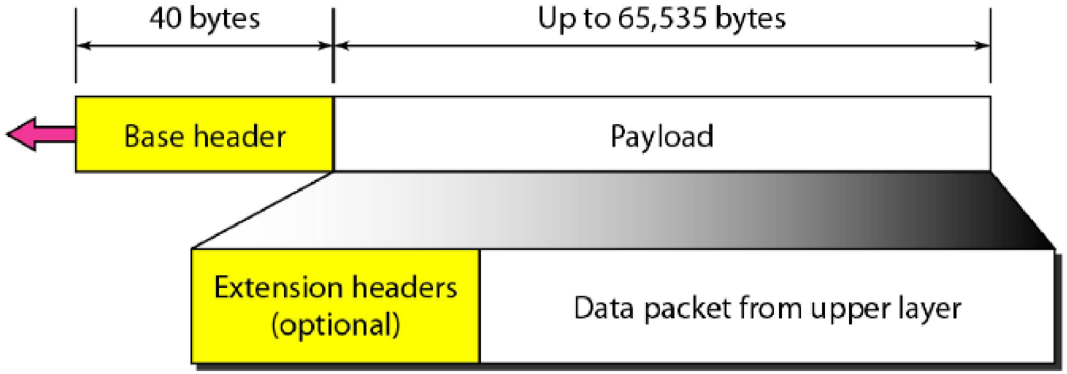
- Larger address space: An IPv6 address is 128 bits long, a huge (2^{96}) increase in the address space
- Better header format: IPv6 uses a new header format with options separated from the base header and inserted, when needed,
- New options. IPv6 has new options to allow for additional functionalities.
- Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- Support for resource allocation: 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

<u>Packet Format</u>

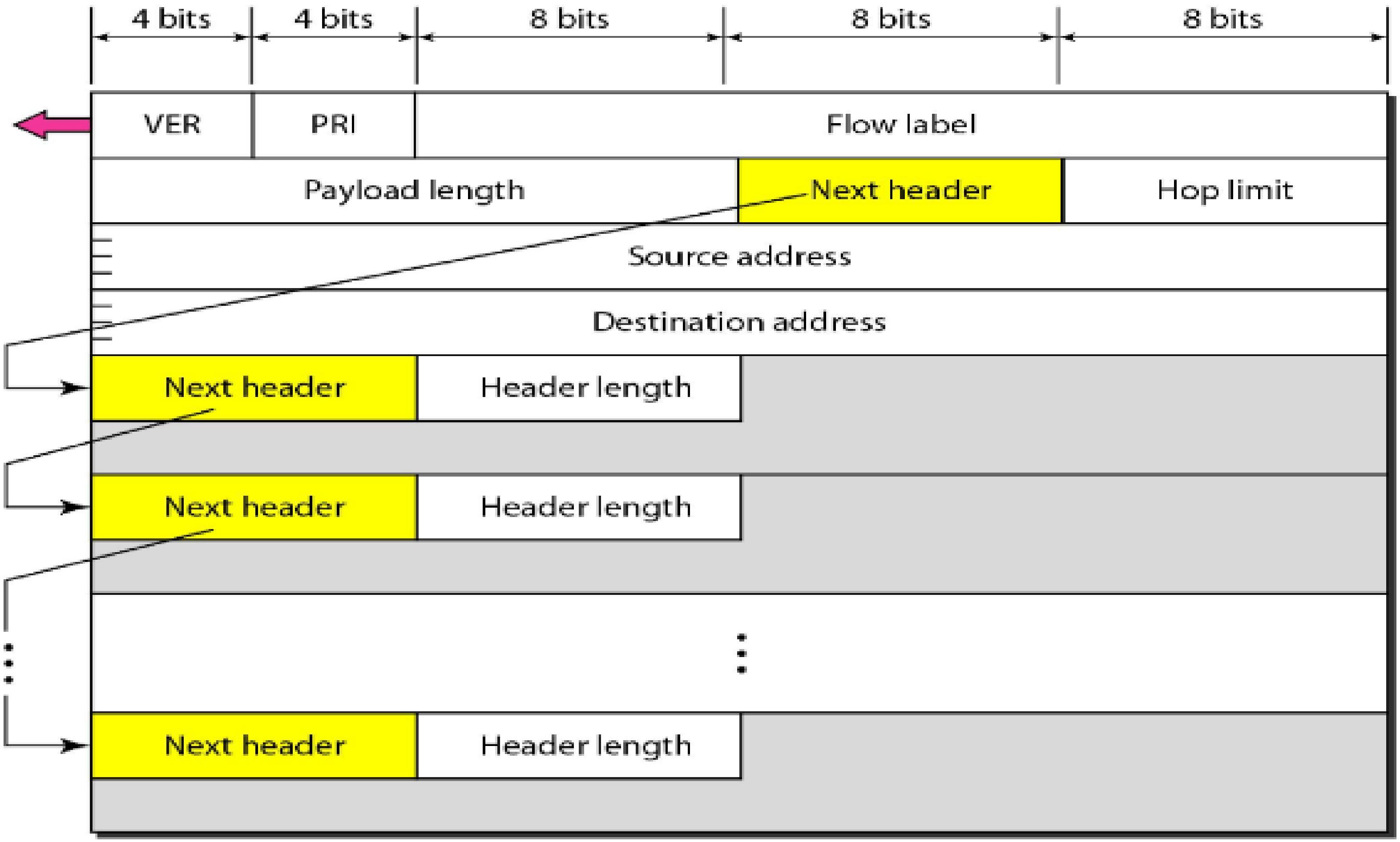
- Each packet is composed of a mandatory base header followed by the payload.
- The payload consists of two parts: optional extension headers and data from an upper layer.
- The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.
- **Base Header**
- Figure shows the base header with its eight fields. These fields are as follows:
- <u>Version</u>. This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- <u>Priority.</u> The 4-bit priority field defines the priority of the packet with respect to traffic congestion. We will discuss this field later.

IPv6 Header


40 bytes
Up to 65,535 bytes
Base header
Payload
Extension headers
(optional)
Data packet from upper layer

IPv6 Header

Figure 20.16 Format ofan IPv6 datagram



IPv6 Header

- Flow label. The flow label is a 3-byte (24-bit) field that is designed to provide handling for a particular flow of data.
- Payload length. The 2-byte payload length field defines the length of the IP datagram
- 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
- Hop limit. This 8-bit hop limit field serves the same purpose as the TIL 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.

IPv6

- Destination address: The destination address field is a 16-byte (128-bit)
- **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 noncongestion-controlled.

IPv6 header format

- <u>Flow Label</u>
- 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 a router, a flow is a sequence of packets that share the same characteristics, such as traveling the same path, using the same resources, having the same kind of security, and so on.
- A router that supports the handling of flow labels has a flow label table.

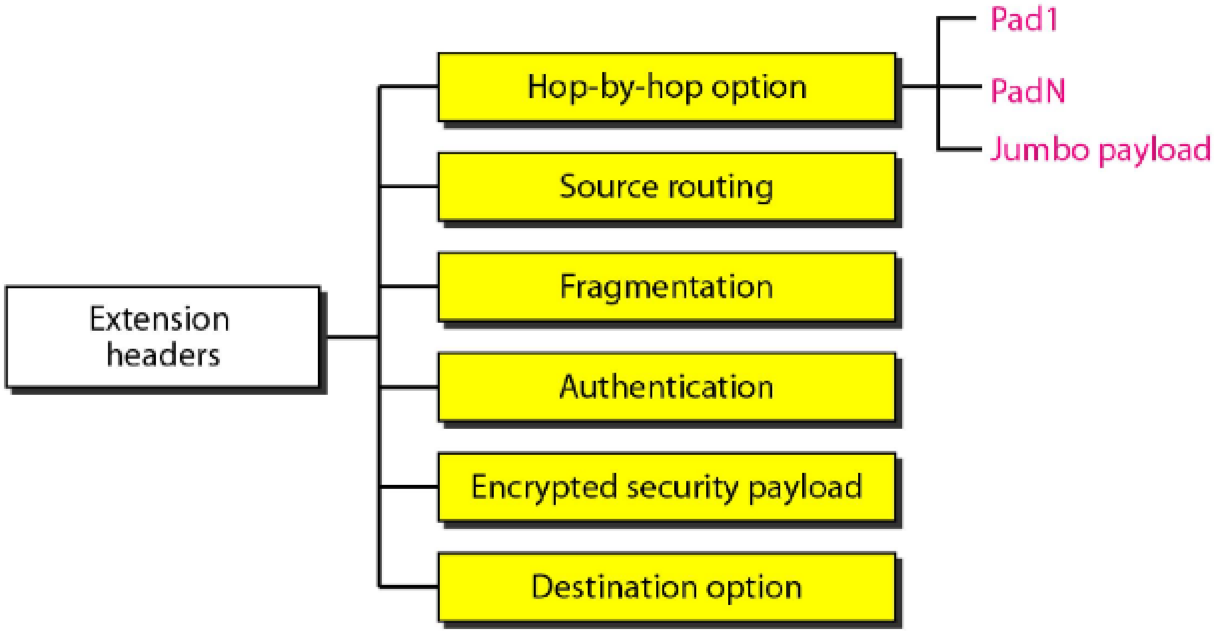
IPv6

- 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.

IPv6 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. Six types of extension headers have been defined, as shown in Figure 20.17.

Figure 20.17 Extension header types



IPV6 Forwarding Techniques

- Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table.
- a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.
- However, this simple solution is impossible today in an internetwork such as the Internet because the number of entries needed in the routing table would make table lookups inefficient.
- Several techniques can make the size of the routing table manageable and also handle issues such as security.

Forwarding Techniques(IPv6)

- Next-Hop Method Versus Route Method
- One technique to reduce the contents of a routing table is called the next-hop method.
- In this technique, the routing table holds only the address of the next hop instead of information about the complete route (route method).
- Figure shows how routing tables can be simplified by using this technique.

a. Routing tables based on route / b. Routing tables based on next hop

	a. Destination	a. Route	b. Destination	b. Next hop
Routing table for host A	Host B	R1, R2, host B	Host B	R1
Routing table for R1	Host B	R2, host B	Host B	R2
Routing table for R2	Host B	Host B	Host B	---

Forwarding Techniques

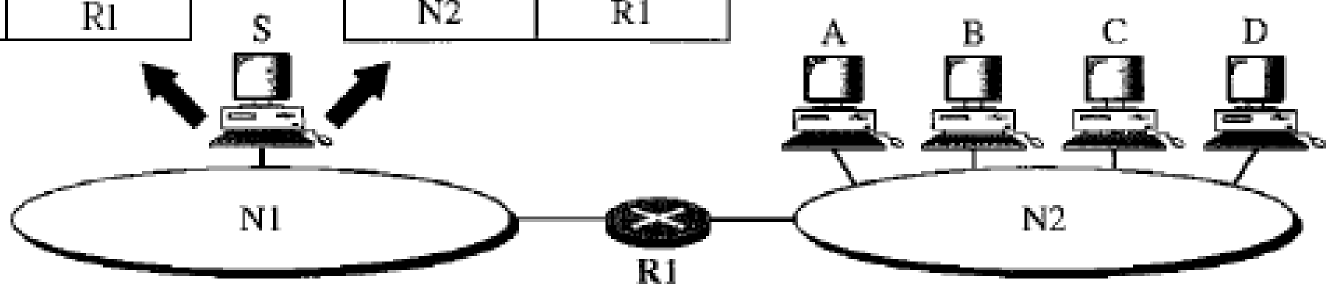
Figure 22.3 *Host-specific versus network-specific method*

Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

Routing table for host S based on network-specific method

Destination	Next hop
N2	R1



Forwarding technique

- Network-Specific Method Versus Host-Specific Method
- A second technique to reduce the routing table and simplify the searching process is called the network-specific method.
- Here, instead of having an entry for every destination host connected to the same physical network (host-specific method), we have only one entry that defines the address of the destination network itself.
- In other words, we treat all hosts connected to the same network as one single entity

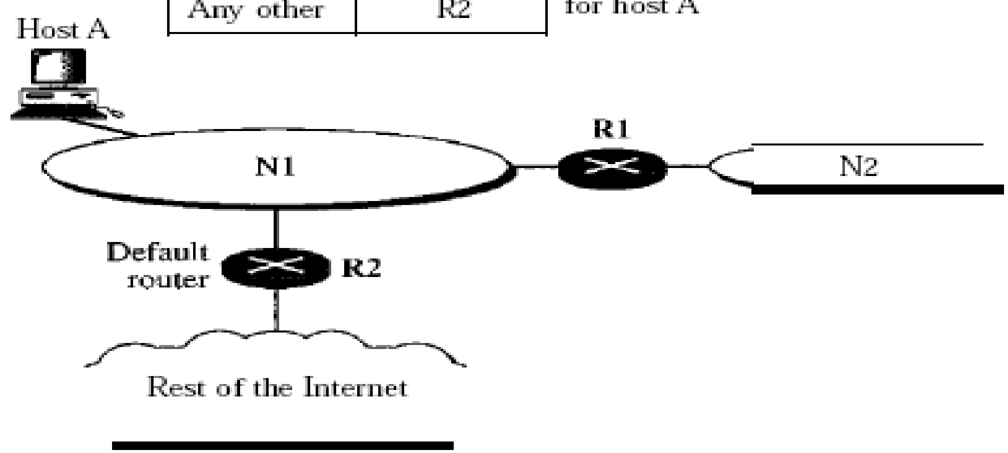
Forwarding Techniques

- <u>Default Method</u>
- Another technique to simplify routing is called the default method. In Figure 22.4 host A is connected to a network with two routers. Router Rl routes the packets to hosts connected to network N2.
- However, for the rest of the Internet, router R2 is used. So instead of listing all networks in the entire Internet, host A can just have one entry called the default.

Figure 22.4 *Default method*

Destination	Next hop
N2	R1
Any other	R2

Routing table for host A



Routing Table(IPv6)

- A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. The routing table can be either static or dynamic.
- Static Routing Table
- A static routing table contains information entered manually. The administrator enters the route for each destination into the table.
- When a table is created, it cannot update. automatically when there is a change in the Internet. The table must be manually altered by the administrator.
- A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting.

Routing Table

- Dynamic Routing Table
- A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP.
- Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically.
- The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets

Routing Table

- *Format*
- As mentioned previously, a routing table for classless addressing has a minimum of four columns.
- We should be aware that the number of columns is vendor-dependent, and not all columns can be found in all routers.
- Figure 22.10 shows some common fields in today's routers.

Figure 22.10 *Common fields in a routing table*

Mask	Network address	Next-hop address	Interlace		Reference count	Use

Routing Table

- ***Format***
- Mask. This field defines the mask applied for the entry.
- Network address. This field defines the network address to which the packet is finally delivered. this field defines the address of the destination host.
- Next-hop address. This field defines the address of the next-hop router to which the packet is delivered.
- Interface. This field shows the name of the interface.
- Flags. This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).

Routing Table

- *Format*
- Reference count. This field gives the number of users of this route at the moment.
- For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.
- Use. This field shows the number of packets transmitted through this router for the corresponding destination.