Module 3: Network Layer

Logical Addressing - IPv4 Addresses

Logical Addressing - IPv4 Addresses

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet. IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. 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.

1. Address Space

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

IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet.

2. Notations

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

a. Binary Notation

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

01110101 10010101 00011101 00000010

b. Dotted-Decimal Notation

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

117.149.29.2

Figure 3.1 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255

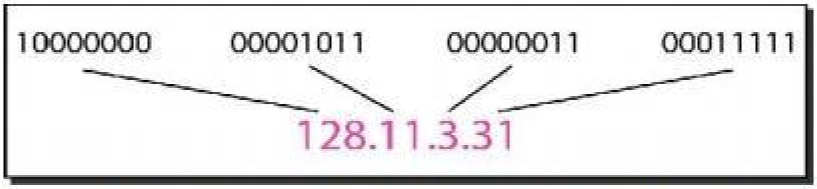


Figure 3.1 Dotted-decimal notation and binary notation for an IPv4 address

Example 3.1

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

a)10000001 00001011 00001011 11101111

b)11000001 10000011 00011011 11111111

Solution

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

a)129.11.11.239

b)193.131.27.255

Example 3.2

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

a)111.56.45.78

b)221.34.7.82

Solution

We replace each decimal number with its binary equivalent.

a)01101111 00111000 00101101 01001110

b)11011101 00100010 00000111 01010010

Example 3.3

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

a)111.56.045.78

b)221.34.7.8.20

c) 75.45.301.14

d)11100010.23.14.67

Solution

a)There must be no leading zero (045).

b)There can be no more than four numbers in an IPv4 address.

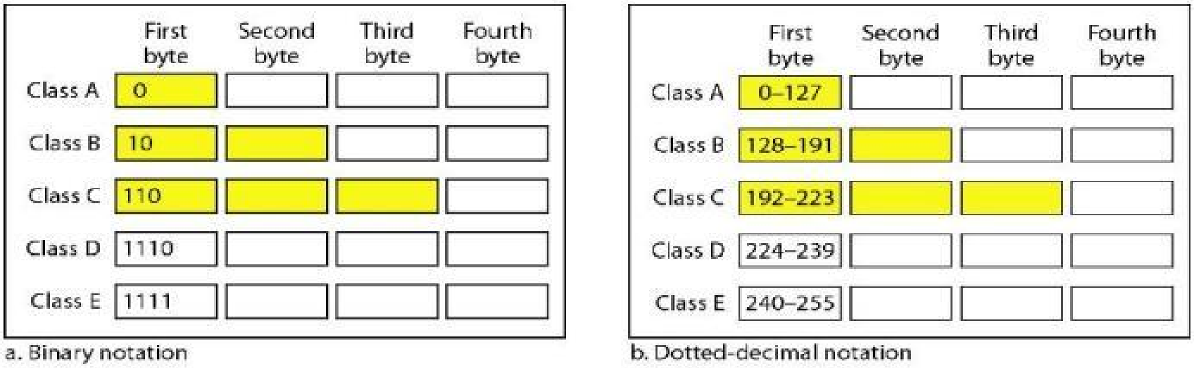
c) Each number needs to be less than or equal to 255 (301 is outside this range).

d)A mixture of binary notation and dotted-decimal notation is not allowed.

3. Classful Addressing

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

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class. Both methods are shown in Figure 3.2.



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

a. Binary notation

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

b. Dotted-decimal notation

Figure 3.2 Finding the classes in binary and dotted-decimal notation

Example 3.4

Find the class of each address.

a)00000001 00001011 00001011 11101111

b)11000001 10000011 00011011 11111111

c) 14.23.120.8

d)252.5.15.111

Solution

a)The first bit is O. This is a class A address.

b)The first 2 bits are 1; the third bit is O. This is a class C address.

c) The first byte is 14 (between 0 and 127); the class is A.

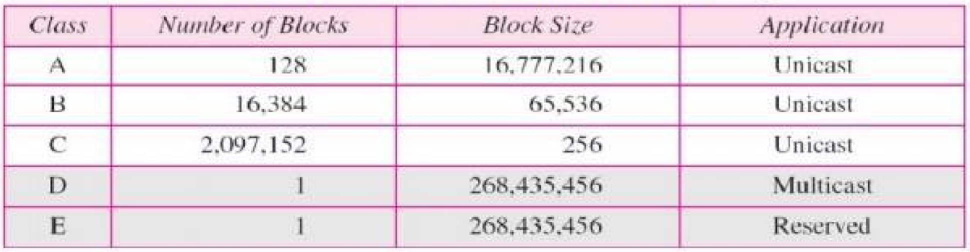
d)The first byte is 252 (between 240 and 255); the class is E.

3.1 Classes and Blocks

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in Table 3.1.

Class A addresses were designed for large organizations with a large number of attached hosts or routers. Class B addresses was 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. Class D addresses were designed for multicasting. The class E addresses were reserved for future use. In c1assfnl addressing, a large part of the available addresses were wasted

Table 3.1 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

3.2 Netid and Hostid

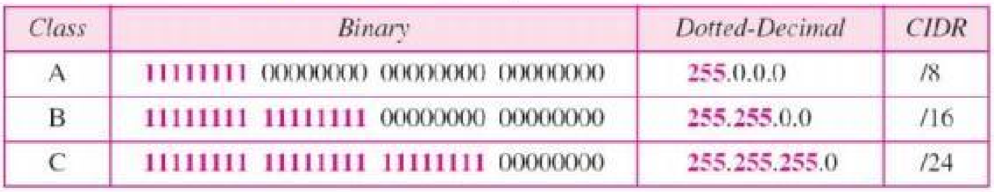
In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.

These parts are of varying lengths, depending on the class of the address. The netid is in color, the hostid is in white. Note that 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.

3.3 Mask

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous Is followed by contiguous as. The masks for classes A, B, and C are shown in Table 3.2. The concept does not apply to classes D and E

Table 3.2 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

The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid. The last column of Table 3.2 shows the mask in the form */n* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called slash notation orClassless Interdomain Routing (CIDR) notation. The notation is used in classless addressing,

3.4 Subnetting

If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets). Subnetting increases the number of Is in the mask.

3.5 Supernetting

In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a supemet. An organization can apply for a set of class C blocks instead of just one.

3.6 Address Depletion

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. Yet the number of devices on the Internet is much less than the 232 address space. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations. One solution that has alleviated the problem is the idea of classless addressing.

4. Classless Addressing

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

4.1 Address Blocks

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block 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. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

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

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

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

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

Example 3.5

Figure 3.3 shows a block of addresses, in both binary and dotted-decimal notation, granted to a small business that needs 16 addresses.

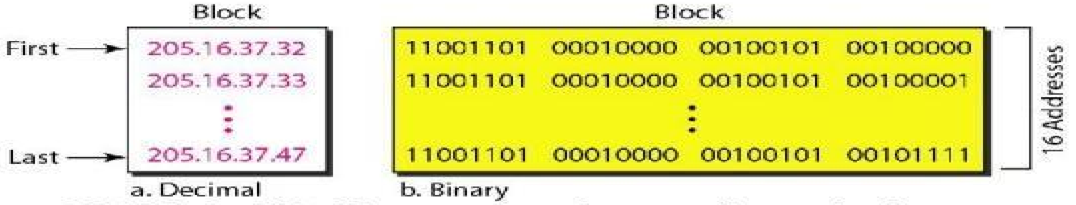


Figure 3.3 A block of16 addresses granted to a small organization

We can see that the restrictions are applied to this block. 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 by 16 results in 215,024,210.

4.2 Mask

A better way to define a block of addresses is to select any address in the block and the mask. As we discussed before, a mask is a 32-bit number in which the n leftmost bits are 1s and the 32 - n rightmost bits are 0s. However, in classless addressing the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of n preceded by a slash.

First Address: The first address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to 0s. The first address in the block can be found by setting the rightmost 32 - n bits to 0s.

Example 3.6

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 00100 I 11. If we set 32 - 28 rightmost bits to 0, we get 11001101 000100000100101 0010000 or 205.16.37.32.

Last Address: The last address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to Is. The last address in the block can be found by setting the rightmost 32 - n bits to Is.

Example 3.7

Find the last address for the block in Example 3.6.

Solution

The binary representation of the given address is 11001101 0001000000100101001001 11. If we set 32 - 28 rightmost bits to 1, we get 11001101 00010000 001001010010 1111 or 205.16.37.47.

Number of Addresses: The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula 232- n.

Example 3.8

Find the number of addresses in Example 3.6.

Solution

The value of n is 28, which means that number of addresses is 232- 28 or 216.

Example 3.9

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

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 Is; the result is 0 otherwise.

Address: 11001101 00010000 00100101 00100111

Mask: 11111111 11111111 11111111 11110000

First address: 11001101 00010000 00100101 00100000

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

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

Mask complement: 000000000 00000000 00000000 00001111

Number of addresses: 15 + 1 =16

4.3 Network Addresses

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

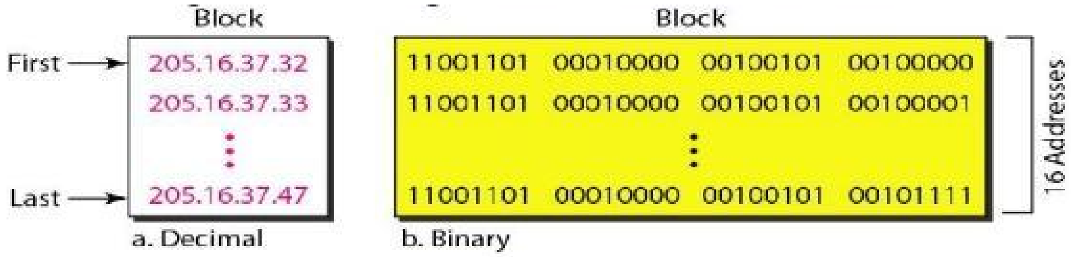


Figure 3.4 A network configuration for the block 205.16.37.32/28

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. We say directly or indirectly because we do not know the structure of the network to which the other side of the router is connected. 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.

4.4 Hierarchy

IP addresses, like other addresses or identifiers we encounter these days, have levels of hierarchy. For example, a telephone network in North America has three levels of hierarchy. The leftmost three digits define the area code, the next three digits define the exchange, the last four digits define the connection of the local loop to the central office. Figure 3.5 shows the structure of a hierarchical telephone number

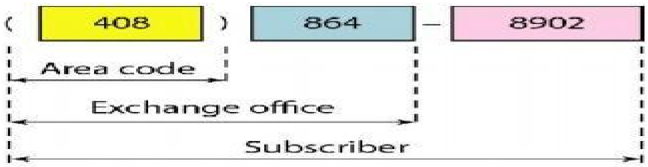


Figure 3.5 Hierarchy in a telephone network in North America

Two-Level Hierarchy: No Subnetting

An IP address can define only two levels of hierarchy when not subnetted. The n leftmost bits of the address x.y.z.t/n define the network (organization network); the 32 – n rightmost bits define the particular host (computer or router) to the network. The two common terms are prefix and suffix. The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix. Figure 3.6 shows the hierarchical structure of an IPv4 address.

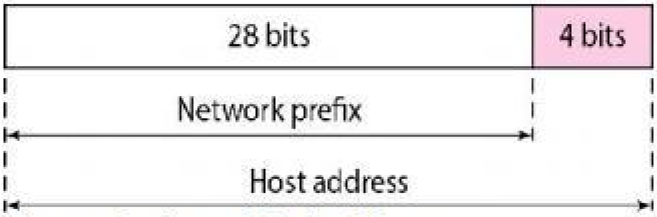


Figure 3.6 Two levels of hierarchy in an IPv4 address

The prefix is common to all addresses in the network; the suffix changes from one device to another. Each address in the block can be considered as a two-level hierarchical structure: the leftmost n bits (prefix) define the network; the rightmost 32 - n bits define the host.

Note that applying the mask of the network, /26 to any of the addresses gives us the network address 17.12.14.0/26. We leave this proof to the reader. We can say that through subnetting, we have three levels of hierarchy. Note that in our example, the subnet prefix length can differ for the subnets as shown in Figure 3.8.

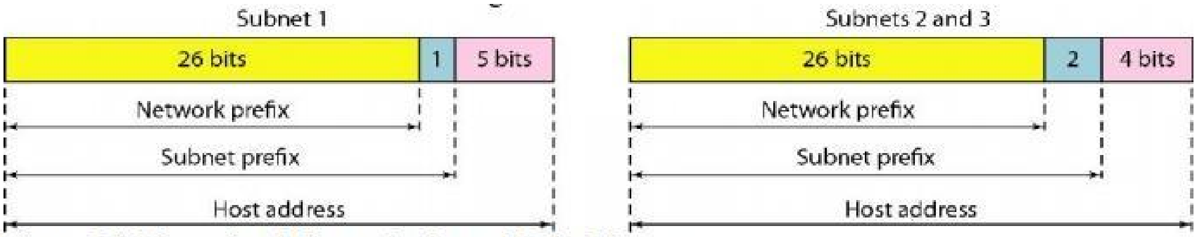


Figure 3.8 Three-level hierarchy in an IPv4 address

More Levels of Hierarchy

The structure of classless addressing does not restrict the number of hierarchical levels. An organization can divide the granted block of addresses into subblocks. Each subblock can in turn be divided into smaller subblocks. And so on. One example of this is seen in the ISPs. A national ISP can divide a granted large block into smaller blocks and assign each of them to a regional ISP. A regional ISP can divide the block received from the national ISP into smaller blocks and assign each one to a local ISP. A local ISP can divide the block received from the regional ISP into smaller blocks and assign each one to a different organization. Finally, an organization can divide the received block and make several subnets out of it.

5. Address Allocation

The next issue in classless addressing is address allocation. The ultimate responsibility of address allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP. Each ISP, in turn, divides its assigned block into smaller subblocks and grants the subblocks to its customers. In other words, an ISP receives one large block to be distributed to its Internet users. This is called address aggregation: many blocks of addresses are aggregated in one block and granted to one ISP.

Example 3.10

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

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

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

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

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

Solution

Figure 3.9 shows the situation.

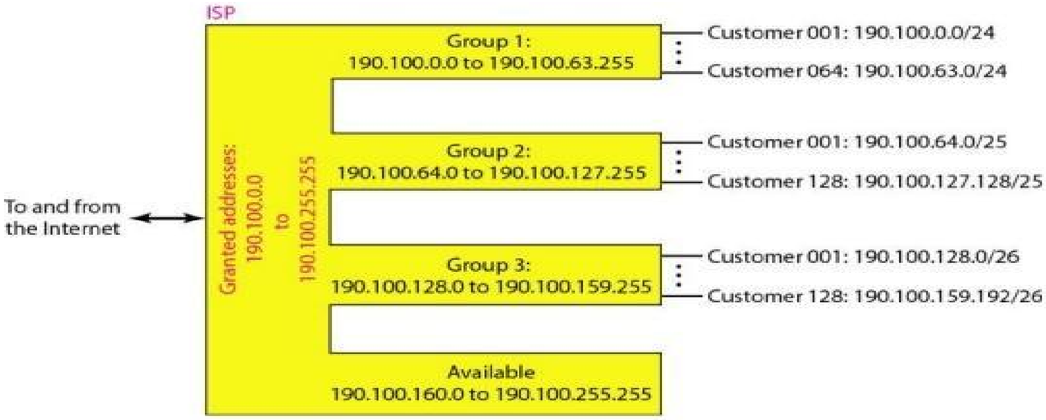

ISP
Group 1:
190.100.0.0 to 190.100.63.255
Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24
Group 2:
190.100.64.0 to 190.100.127.255
Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25
To and from
the Internet
Granted addresses:
190.100.0.0
to
190.100.255.255
Group 3:
190.100.128.0 to 190.100.159.255
Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26
Available
190.100.160.0 to 190.100.255.255

Figure 3.9 An example of address allocation and distribution by an ISP

1. Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2256) bits are needed to define each host. The prefix length is then 32 - 8 =24. The addresses are

1st Customer: 190.100.0.0/24 100.0.255/24

2nd Customer: 190.100.1.0/24190   190.100.1.255/24

64th Customer: 190.100.63.0/24   190.100.63.255/24

Total =64 X 256 =16,384

2. Group2

For this group, each customer needs 128 addresses. This means that 7 (10g2 128) bits are needed to define each host. The prefix length is then 32 - 7 =25. The addresses are

3. Group3

For this group, each customer needs 64 addresses. This means that 6 (log2 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   190.100.128.63/26

2nd Customer: 190.100.128.64/26   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

IPv4 (The Internet Protocol version 4)

IPv4:

The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term *best-effort* means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

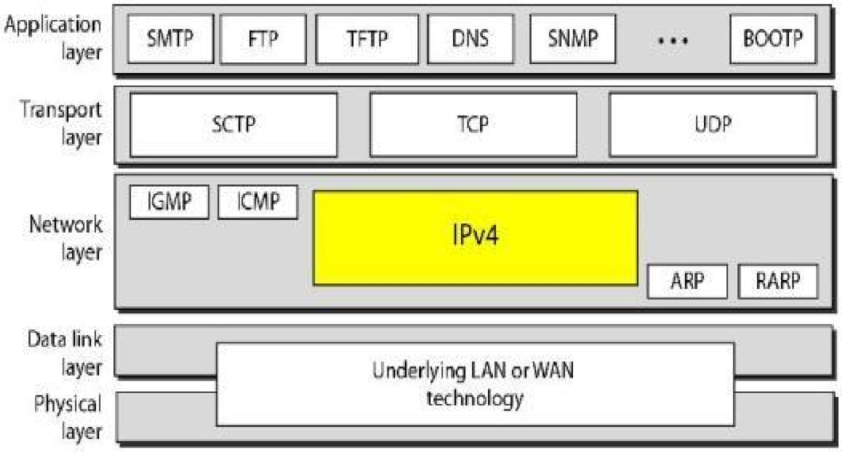


Figure 3.18 Position ofIPv4 in TCPIIP protocol suite

1. Datagram

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

- **Version (VER):** This 4-bit field defines the version of the IPv4 protocol. Currently theversion is 4. However, version 6 (or IPv6) may totally replace version 4 in the future. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4. All fields must be interpreted as specified in the fourth version of the protocol. If the machine is using some other version of IPv4, the datagram is discarded rather than interpreted incorrectly

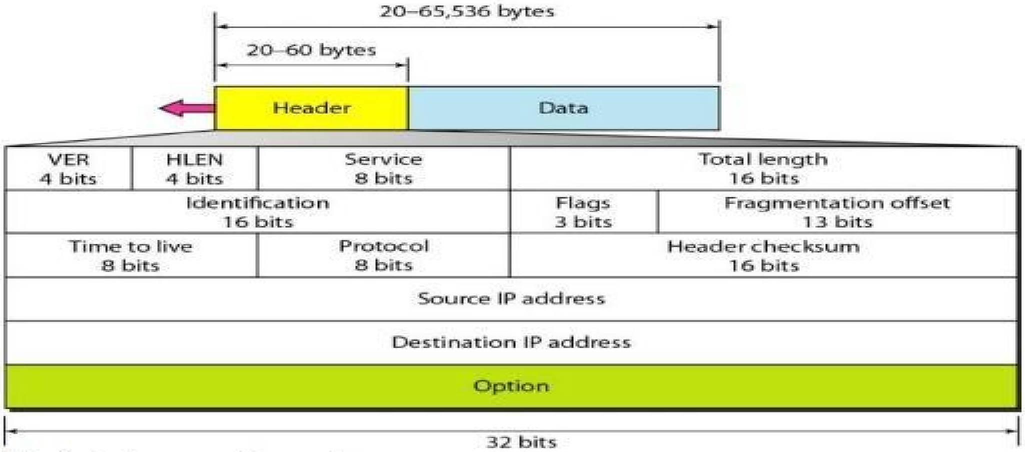


Figure 3.19 IPv4 datagram format

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

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

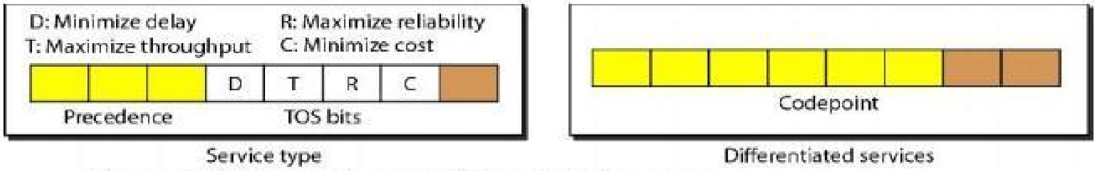


Figure 3.20 Service type or differentiated services

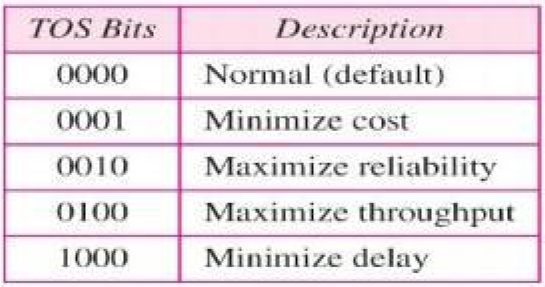
1. Service Type

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.

- **Precedence** is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary).The precedence defines the priority of the datagram in issues such as congestion.

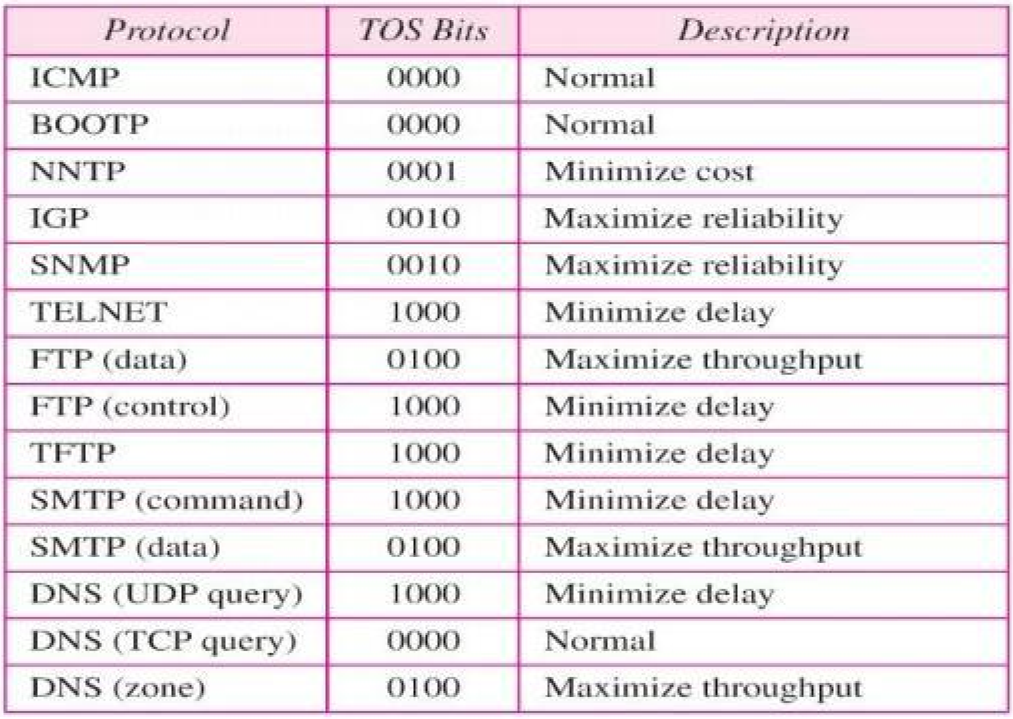
- **TOS bits** are a 4-bit subfield with each bit having a special meaning. Although abit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in Table 3.4. With only 1 bit set at a time, we can have five different types of services.

Table 3.6 Types of service



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

Table 3.7 Default types of service



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

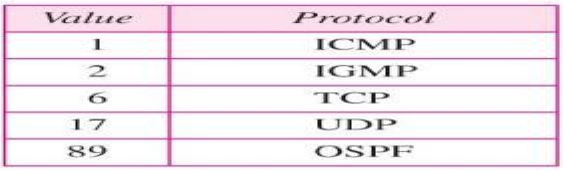
2. Differentiated Services

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

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

- When the 3 rightmost bits are not all 0s, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities. The first category contains 32 service types; the second and the third each contain 16. The first category (numbers 0, 2, 4, ... ,62) is assigned by the Internet authorities (IETF). The second category (3, 7, 11, 15, , 63) can be used by local authorities (organizations). The third category (1, 5, 9, .., 61) is temporary and can be used for experimental purposes

Table 3.8 Values for code points



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

- **Total length**. This is a In-bit field that defines the total length (header plus data) of theIPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4.

If the size of an IPv4 datagram is less than 46 bytes, some padding will be added to meet this requirement. In this case, when a machine de-capsulate the datagram, it needs to check the total length field to determine how much is really data and how much is padding.

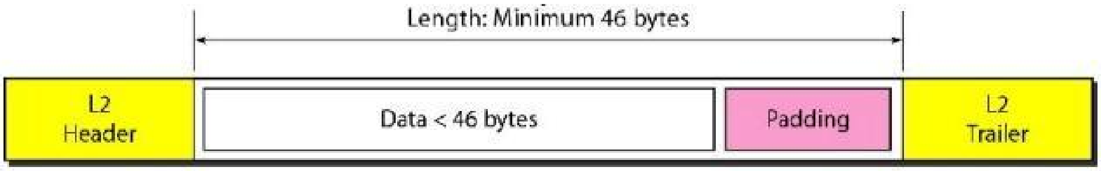


Figure 3.21 Encapsulation of a small datagram in an Ethernet frame

- **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. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zero.
- **Protocol:** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered. In other words, since the IPv4 protocol carries data from different other protocols.
- **Checksum:** The checksum concept and its calculation.
- **Source address:** This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Destination address:** This 32-bit field defines the IPv4 address of the destination

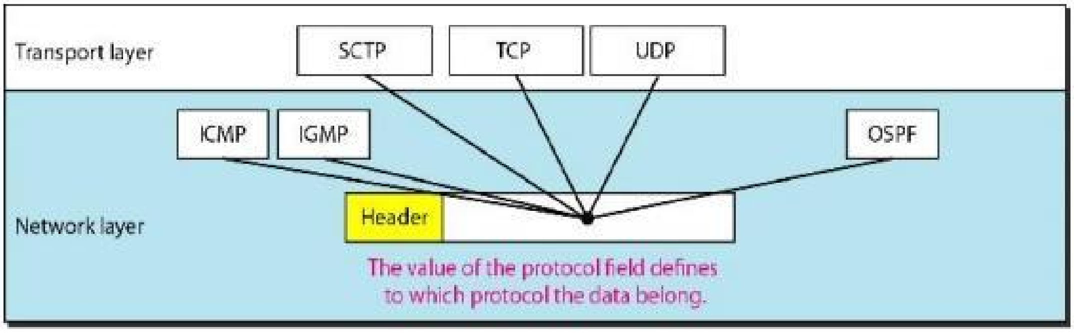

Transport layer
SCTP
TCP
UDP
ICMP
IGMP
OSPF
Network layer
Header
The value of the protocol field defines
to which protocol the data belong.

Figure 3.22 Protocol field and encapsulated data

Example 3.12

An IPv4 packet has arrived with the first 8 bits as shown: 01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length (2 x 4 =8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 3.13

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 x 4, or 32 bytes. The first 20 bytes are the base header; the next 12 bytes are the options.

Example 3.14

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 x 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40- 20).

Example 3.15

An IPv4 packet has arrived with the first few hexadecimal digits as shown. 0x45000028000100000102 ...

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes (16 hexadecimal digits). The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP..

Fragmentation

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

Checksum

The implementation of the checksum in the IPv4 packet follows the same principles. First, the value of the checksum field is set to O. Then the entire header is divided into 16-bit sections and added together. The result (sum) is complemented and inserted into the checksum field.

Example 3.16

Figure 3.23 shows an example of a checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented.

The result is inserted in the checksum field.

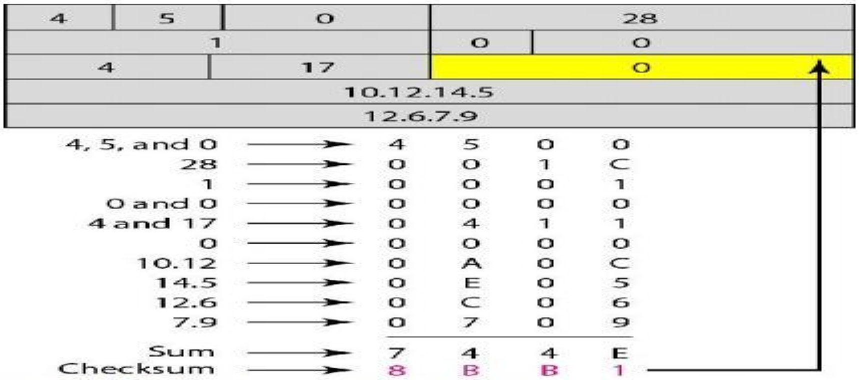


Figure 3.23 Example of checksum calculation in IPv4

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 and was discussed in the previous section. 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.

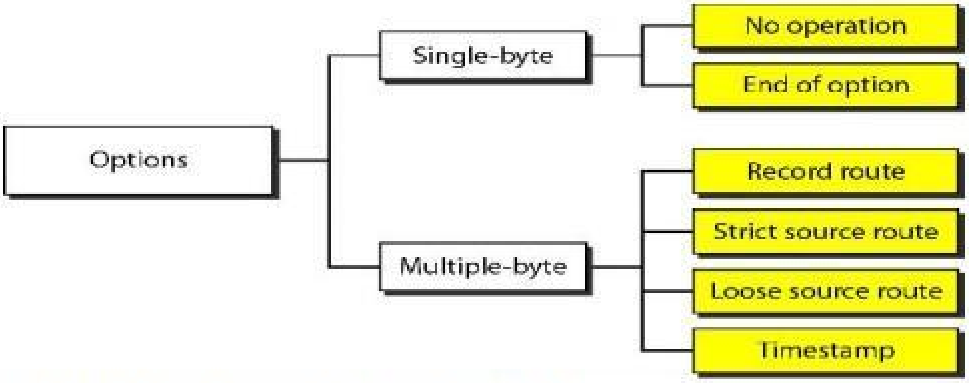


Figure 3.24 Taxonomy of options in IPv4

No Operation

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

End of Option

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

Record Route

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

Strict Source Route

A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Dictation of a route by the source can be useful for several purposes. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose.

Loose Source Route

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

Timestamp

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

IPv6 Addressing and IPv6 Header

IPv6 Addresses

Despite all short-term solutions, such as classless addressing, Dynamic Host Configuration Protocol (DHCP) and NAT, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself, such as lack of accommodation for real-time audio and video transmission, and encryption and authentication of data for some applications, have been the motivation for IPv6.

1. Structure

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

2. Hexadecimal Colon Notation

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

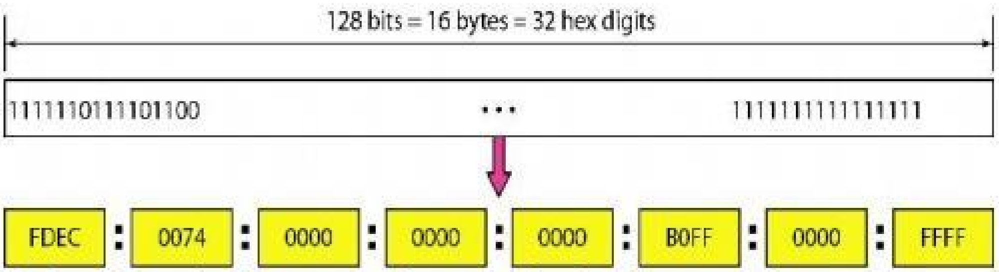


Figure 3.10 IPv6 address in binary and hexadecimal colon notation

3. Abbreviation

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

Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0.

Example 3.11

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

Solution

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

xxxx: xxxx: xxxx: xxxx: xxxx: xxxx: xxxx: xxxx

0: 15: : l: 12: 1213

This means that the original address is

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

4. Address Space

IPv6 has a much larger address space; 2128 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 designed such that no code is identical to the first part of any other code. In this way, there is no ambiguity; when an address is given, the type prefix can easily be determined.

5. Unicast Addresses

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. We discuss the second type here; the first type is left for future definition. The provider-based address is generally used by a normal host as a unicast address.

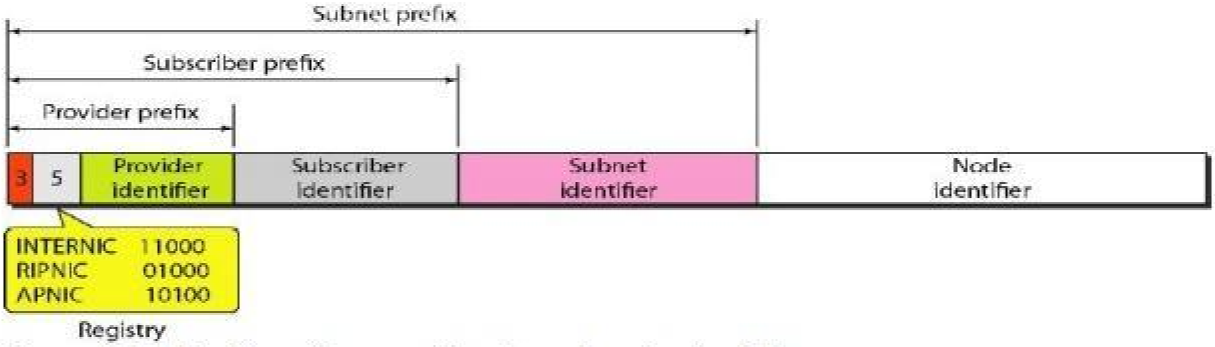


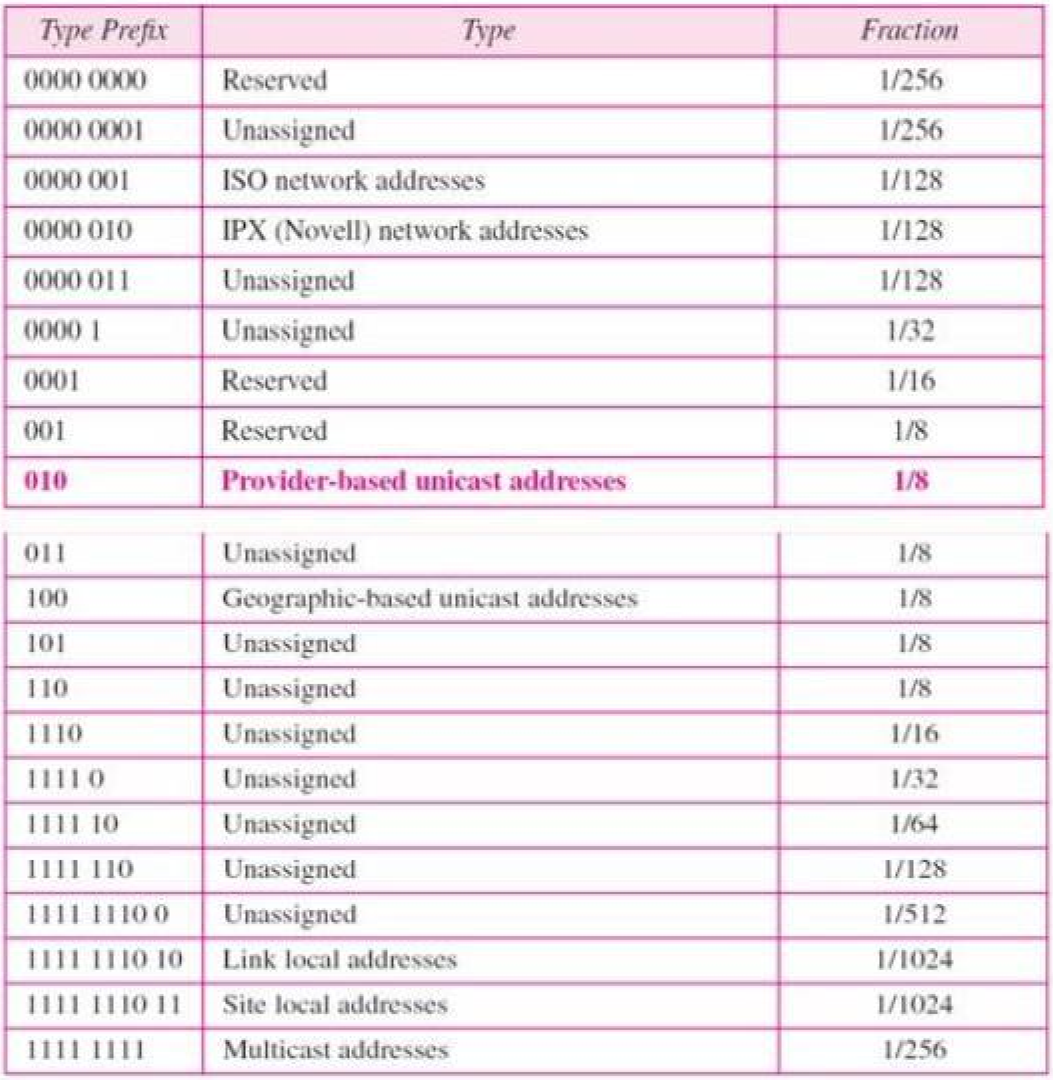
Figure 3.11 Prefixes for provider-based unicast address

Table 3.3 Type prefixes for 1Pv6 addresses

Fields for the provider-based address are as follows:

- **Type identifier:** This 3-bit field defines the address as a provider-based address.

- **Registry identifier.** This 5-bit field indicates the agency that has registered the address. Currently three registry centers have been defined. INTERNIC (code 11000) is the center for North America; RIPNIC (code 01000) is the center for European registration; and APNIC (code 10100) is for Asian and Pacific countries.



Type Prefix	*Type*	*Fraction*
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	**Provider-based unicast addresses**	**1/8**
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

· **Provider identifier.** This variable-length field identifies the provider for Internet access(such as an ISP). A 16-bit length is recommended for this field.

· **Subscriber identifier.** When an organization subscribes to the Internet through a provider, itis assigned subscriber identification. A 24-bit length is recommended for this field.

· **Subnet identifier.** Each subscriber can have many different subnetworks, and eachsubnetwork can have an identifier. The subnet identifier defines a specific subnetwork under the territory of the subscriber. A 32-bit length is recommended for this field.

· **Node identifier.** The last field defines the identity of the node connected to a subnet. Alength of 48 bits is recommended for this field to make it compatible with the 48-bit link (physical) address used by Ethernet. In the future, this link address will probably be the same as the node physical address.

6. Multicast Addresses

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

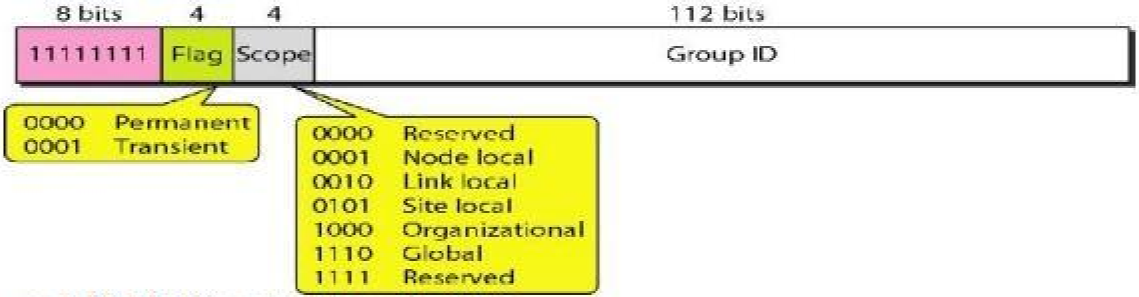


Figure 3.12 Multicast address in IPv6

The second field is a flag that defines the group address as either permanent or transient. A permanent group address is defined by the Internet authorities and can be accessed at all times. A transient group address, on the other hand, is used only temporarily. Systems engaged in a teleconference, for example, can use a transient group address. The third field defines the scope of the group address. Many different scopes have been defined.

7. Any cast Addresses

IPv6 also defines anycast addresses. An anycast address, like a multicast address, also defines a group of nodes. However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route). Although the definition of an anycast address is still debatable, one possible use is to assign an anycast address to all routers of an ISP that covers a large logical area in the Internet.

8. Reserved Addresses

Another category in the address space is the reserved address. These addresses start with eight Os (type prefix is 00000000). A few subcategories are defined in this category.

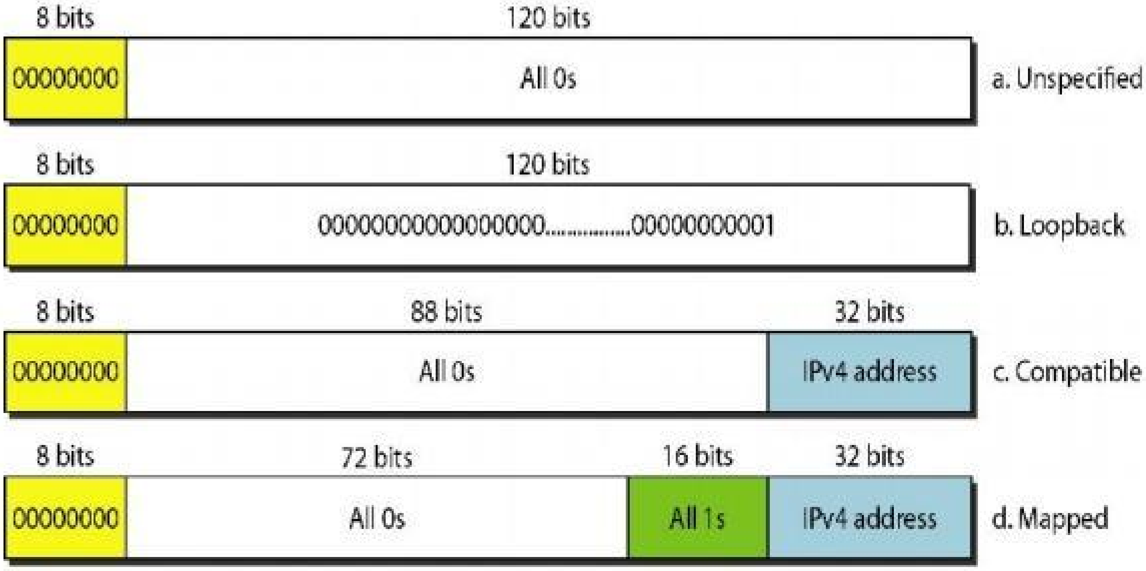


Figure 3.13 Reserved addresses in IPv6

An unspecified address is used when a host does not know its own address and sends an inquiry to find its address. A loopback address is used by a host to test itself without going into the network. A compatible address is used during the transition from IPv4 to IPv6.

9. 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 addresses are defined for this purpose.

IPv6 and its Header:

The network layer protocol in the TCPIIP protocol suite is currently IPv4 (Internetworking Protocol, version 4). IPv4 provides the host-to-host communication between systems in the Internet. Although IPv4 is well designed, data communication has evolved since the inception of IPv4 in the 1970s. IPv4 has some deficiencies that make it unsuitable for the fast-growing Internet.

Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet. The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.

1. Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

- Larger address space
- Better header format
- New options
- Allowance for extension
- Support for resource allocation
- Support for more security

2. Packet Format

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.

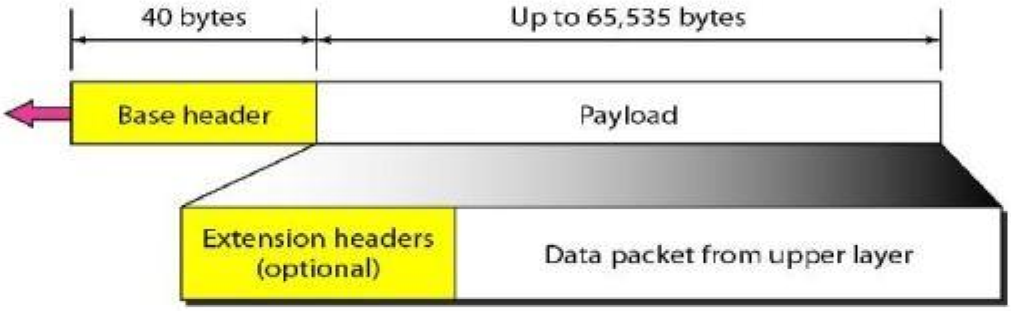


Figure 3.25 IPv6 datagram header and payload

a. Base Header

These fields are as follows:

- **Version:** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- **Priority**: The 4-bit priority field defines the priority of the packet with respect to traffic congestion.
- **Flow label**: The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
- **Payload length:** The 2-byte payload length field defines the length of the IP datagram excluding the base header.
- **Next header**: The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field.

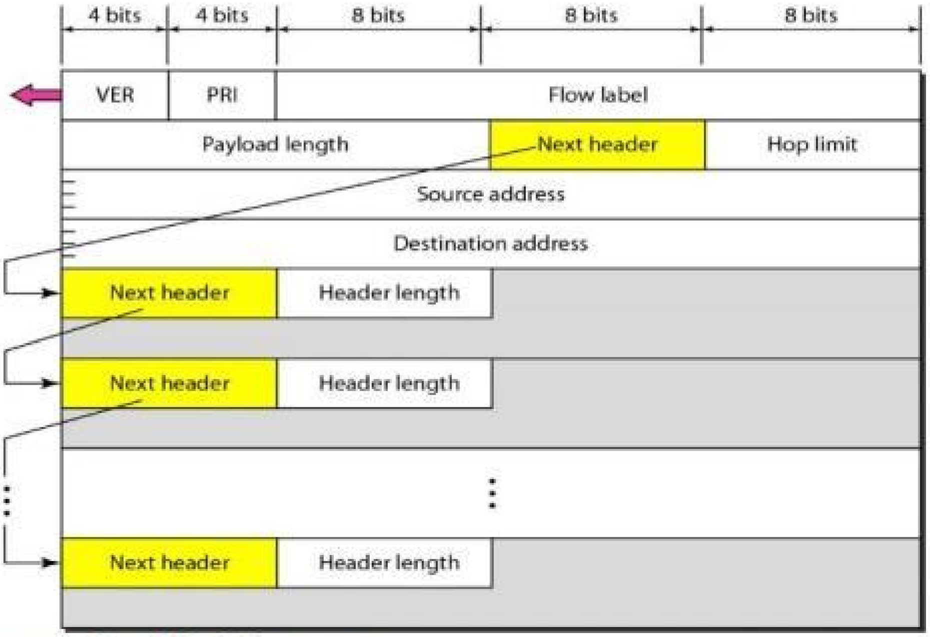


Figure 3.26 Format of an IPv6 datagram

- **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.
- **Destination address.** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.
- **Priority:** The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source. For example, if one of two consecutive datagrams must be discarded due to congestion, the datagram with the lower packet priority will be discarded. IPv6 divides traffic into two broad categories: congestion-controlled and non-congestion-controlled.

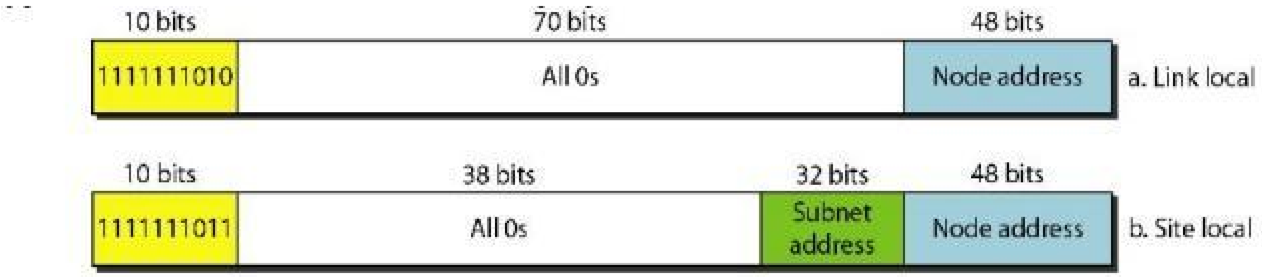


Figure 3.14 Local addresses in IPv6

Question: Elaborate IP Protocol and thus illustrate key differences in its V4 and V6?

What is IP?

An Internet Protocol address is also known as IP address. It is a numerical label which assigned to each device connected to a computer network which uses the IP for communication.

IP address act as an identifier for a specific machine on a particular network. The IP address is also called IP number and internet address. IP address specifies the technical format of the addressing and packets scheme. Most networks combine IP with a TCP (Transmission Control Protocol). It also allows developing a virtual connection between a destination and a source.

Introduction to IPv4 and IPv6

IPv4 and IPv6 are the internet protocols applied at the network layer and uses there best-effort, connectionless datagram delivery service model. IPv4 is the most widely used protocol right now whereas IPv6 is the next generation protocol for the internet.

Each IP datagram is handled independently from all others and it does not maintain any connection state information about related datagrams within the network elements (like routers). IP doesn't provide any reliability, it must be provided by the upper layers i.e Transport layer (E.g TCP, UDP).

What is IPv4?

IPv4 was the first version of IP. It was deployed for production in the ARPANET in 1983. Today it is most widely used IP version. It is used to identify devices on a network using an addressing system.

IPv4 is the fourth version of Internet protocol which uses 32 bit addressing and it allows 2^32 unique addresses i.e 4,294,967,296.

IPv4 address notation: 239.255.255.255, 255.255.255.0.

Till date, it is considered the primary Internet Protocol and carries 94% of Internet traffic.

What is IPv6?

It is the most recent version of the Internet Protocol. Internet Engineer Taskforce initiated it in early 1994. The design and development of that suite is now called IPv6.

IPv6 is a next-generation internet protocol which uses 128 bits addressing. Ipv6 allows 2^128 (340-undecillion) unique addresses i.e (34,000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000).

IPv6 addresses are denoted by eight groups of hexadecimal quartets separated by colons, E.g: 2001:cdba:0000:0000:0000:0000:3257:9652.

This new IP address version is being deployed to fulfill the need for more Internet addresses. It was aimed to resolve issues which are associated with IPv4. With 128-bit address space, it allows 340 undecillion unique address space. IPv6 also called IPng (Internet Protocol next generation).

Features of IPv4

- Connectionless Protocol
- Allow creating a simple virtual communication layer over diversified devices
- It requires less memory, and ease of remembering addresses
- Already supported protocol by millions of devices
- Offers video libraries and conferences

Features of IPv6

- Hierarchical addressing and routing infrastructure
- Stateful and Stateless configuration
- Support for quality of service (QoS)
- An ideal protocol for neighboring node interaction

Comparing Ipv4 and IPv6

IPv4:

- IPv4 is the **fourth version** of Internet protocol which uses 32 bit addressing and it allows 2^32 unique addresses i.e 4,294,967,296.
- The address notation for IPv4 is 239.255.255.255, 255.255.255.0.
- IPv4 has different class types: **A, B, C, D, and E**.
  - **Class A**, **Class B**, and **Class C** are the three classes of addresses used on IP networks in common practice.
  - **Class D** addresses are reserved for multicast.
  - **Class E** addresses are simply reserved, meaning they should not be used on IP networks.
- IPv4 addresses are categorized into three basic types:
  - **Unicast address:** A Unicast address acts as an identifier for a single interface
  - **Multicast address:** A Multicast address acts as an identifier for a group/set of interfaces that may belong to the different nodes.
  - **Broadcast address**: A Broadcast address, broadcasting a packet to an entire IPv4 subnet using the private IP address.
- **Address Resolution Protocol** (ARP): ARP is used by IPv4 to find a physical address, such as the MAC or link address, associated with an IPv4 address.
- **Domain Name System (DNS):** Applications accept hostnames and then use DNS to get an IP address, using socket API gethostbyname().
- Fragmentation of large packets occurs at both sender and destination nodes.
- **Internet Control Message Protocol(ICMP)** Used by IPv4 to communicate network information.
- **Maximum transmission unit (MTU)** Maximum transmission unit of a link is the maximum number of bytes that a particular link type, such as Ethernet or modem, supports. For IPv4, 576 is the minimum.
- Lack of security.

IPv6:

- IPv6 is a **next-generation internet protocol** which uses 128 bits addressing. It allows 2^128 (340-undecillion) unique addresses i.e (34,000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000).
- IPv6 addresses are denoted by eight groups of hexadecimal quartets separated by colons, E.g: 2001:cdba:0000:0000:0000:0000:3257:9652
  - The double colon (::) can be used once in the text form of an address to designate any number of 0 bits. For example,::ffff:10.120.78.40 is an **IPv4-mapped IPv6 address.**
- Ipv6 can be classified into three categories:
  - **Unicast addresses:** A Unicast address acts as an identifier for a single interface. An IPv6 packet sent to a Unicast address is delivered to the interface identified by that address.

  - **Multicast addresses:** A Multicast address acts as an identifier for a group/set of interfaces that may belong to the different nodes. An IPv6 packet delivered to a multicast address is delivered to the multiple interfaces.
  - **Anycast addresses:** Anycast addresses act as identifiers for a set of interfaces that may belong to the different nodes. An IPv6 packet destined for an Anycast address is delivered to one of the interfaces identified by the address.
- **Domain Name System (DNS):** Applications also accept IP addresses and then use DNS to get hostnames using APIs gethostbyaddr() or getaddrinfo ().
- Fragmentation can only occur at the source node, and reassembly is only done at the destination node.
- **Internet Control Message Protocol version 6 (ICMPv6)** provides some new attributes, other then attributes present in IPV4.
- **Maximum transmission unit (MTU):** Maximum transmission unit of a link is the maximum number of bytes that a particular link type, such as Ethernet or modem, supports. For IPv6, 1280 is the minimum.
- **Strong security** with inbuilt features like Encryption and Authentication.

Advantages of IPv6 over IPv4

- IPv6 provides increased address space.
- Reduced management requirement.
- IPv6 simplified the router’s task compared to IPv4.
- IPv6 allows for bigger payloads than what is allowed in IPv4.
- More efficient routing.
- IPv6 is more compatible with mobile networks than IPv4.
- Improved methods to change ISP.
- Better mobility support and multi-homing.
- Strong security with inbuilt features like Encryption and Authentication.

Difference between IPv4 and IPv6 Addresses

IPv4 & IPv6 are both IP addresses that are binary numbers. IPv4 is 32 bit binary number while IPv6 is 128 bit binary number address. IPv4 address are separated by periods while IPv6 address are separated by colons.

Both are used to identify machines connected to a network. In principle, they are the same, but they are different in how they work.

Basis for differences	IPv4	IPv6
Size of IP address	IPv4 is a 32-Bit IP Address.	IPv6 is 128 Bit IP Address.

Addressing method	IPv4 is a numeric address, and its binary bits are separated by a dot (.)	IPv6 is an alphanumeric address whose binary bits are separated by a colon (:). It also contains hexadecimal.
Number of header fields	12	8
Length of header filed	20	40
Checksum	Has checksum fields	Does not have checksum fields
Example	12.244.233.165	2001:0db8:0000:0000:0000:ff00:0042:7879
Type of Addresses	Unicast, broadcast, and multicast.	Unicast, multicast, and anycast.
Number of classes	IPv4 offers five different classes of IP Address. Class A to E.	lPv6 allows storing an unlimited number of IP Address.
Configuration	You have to configure a newly installed system before it can communicate with other systems.	In IPv6, the configuration is optional, depending upon on functions needed.
VLSM support	IPv4 support VLSM (Virtual Length Subnet Mask).	IPv6 does not offer support for VLSM.
Fragmentation	Fragmentation is done by sending and forwarding routes.	Fragmentation is done by the sender.
Routing Information Protocol (RIP)	RIP is a routing protocol supported by the routed daemon.	RIP does not support IPv6. It uses static routes.
Network Configuration	Networks need to be configured either manually or with DHCP. IPv4 had several overlays to handle Internet growth, which require more maintenance efforts.	IPv6 support autoconfiguration capabilities.
Best feature	Widespread use of NAT (Network address translation) devices which allows single NAT address can mask thousands of non-routable addresses, making end-to-end integrity achievable.	It allows direct addressing because of vast address Space.
Address Mask	Use for the designated network from host portion.	Not used.

SNMP	SNMP is a protocol used for system management.	SNMP does not support IPv6.
Mobility & Interoperability	Relatively constrained network topologies to which move restrict mobility and interoperability capabilities.	IPv6 provides interoperability and mobility capabilities which are embedded in network devices.
Security	Security is dependent on applications - IPv4 was not designed with security in mind.	IPSec(Internet Protocol Security) is built into the IPv6 protocol, usable with a proper key infrastructure.
Packet size	Packet size 576 bytes required, fragmentation optional	1208 bytes required without fragmentation
Packet fragmentation	Allows from routers and sending host	Sending hosts only
Packet header	Does not identify packet flow for QoS handling which includes checksum options.	Packet head contains Flow Label field that specifies packet flow for QoS handling
DNS records	Address (A) records, maps hostnames	Address (AAAA) records, maps hostnames
Address configuration	Manual or via DHCP	Stateless address autoconfiguration using Internet Control Message Protocol version 6 (ICMPv6) or DHCPv6
IP to MAC resolution	Broadcast ARP	Multicast Neighbour Solicitation
Local subnet Group management	Internet Group Management Protocol GMP)	Multicast Listener Discovery (MLD)
Optional Fields	Has Optional Fields	Does not have optional fields. But Extension headers are available.
IPSec	Internet Protocol Security (IPSec) concerning network security is optional	Internet Protocol Security (IPSec) Concerning network security is mandatory
Dynamic host configuration Server	Clients have approach DHCS (Dynamic Host Configuration server) whenever they want to connect to a network.	A Client does not have to approach any such server as they are given permanent addresses.
Mapping	Uses ARP(Address Resolution Protocol) to map to MAC address	Uses NDP(Neighbour Discovery Protocol) to map to MAC address
Combability with mobile devices	IPv4 address uses the dot-decimal notation. That's	IPv6 address is represented in hexadecimal, colon- separated notation. IPv6 is better suited to mobile networks.

why it is not suitable for
mobile networks.

IPv4 and IPv6 cannot communicate with other but can exist together on the same network. This is known as **Dual Stack.**

KEY DIFFERENCE

- IPv4 is 32-Bit IP address whereas IPv6 is a 128-Bit IP address.
- IPv4 is a numeric addressing method whereas IPv6 is an alphanumeric addressing method.
- IPv4 binary bits are separated by a dot(.) whereas IPv6 binary bits are separated by a colon(:).
- IPv4 offers 12 header fields whereas IPv6 offers 8 header fields.
- IPv4 supports broadcast whereas IPv6 doesn't support broadcast.
- IPv4 has checksum fields while IPv6 doesn't have checksum fields
- IPv4 supports VLSM (Virtual Length Subnet Mask) whereas IPv6 doesn't support VLSM.
- IPv4 uses ARP (Address Resolution Protocol) to map to MAC address whereas IPv6 uses NDP (Neighbour Discovery Protocol) to map to MAC address

Network Address Mapping

BRIEF INTRODUCTION TO DIFFERENT TYPES OF ADDRESSING IN NETWORKING:

Four levels of addresses are used in an internet employing the TCP/IP protocols: physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses.

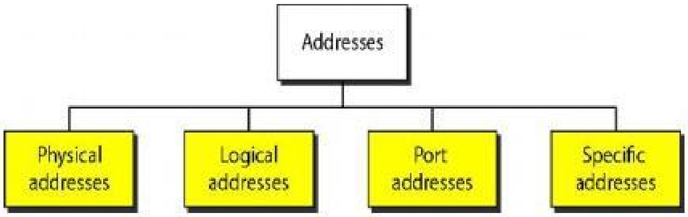


Figure.1.24 Addressing in TCP/IP

Each address is related to a specific layer in the TCP/IP architecture.

a. Physical Addresses:

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address.

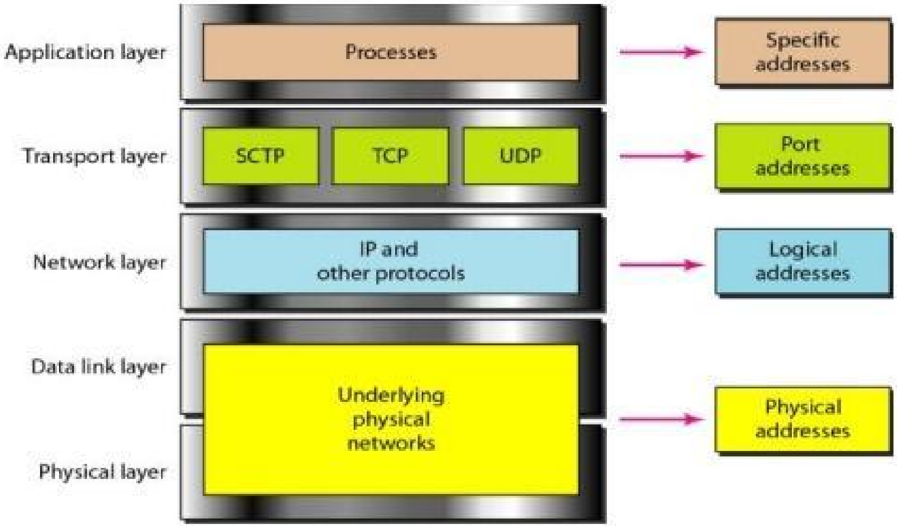


Figure.1.25 Relationship of layers and addresses in TCP/IP

The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a I-byte dynamic address that changes each time the station comes up.

b. Logical Addresses

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network.

The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet.

c. Port Addresses:

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

d. Specific Addresses:

Some applications have user-friendly addresses that are designed for that specific address.

Examples include the e-mail address and the Universal Resource Locator (URL). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

Address Mapping

An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host. The hosts and routers are recognized at the network level by their logical (IP) addresses. However, packets pass through physical networks to reach these hosts and routers. At the physical level, the hosts and routers are recognized by their physical addresses.

A physical address is a local address. Its jurisdiction is a local network. It must be unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware. An example of a physical address is the 48-bit MAC address in the Ethernet protocol, which is imprinted on the NIC installed in the host or router.

Limitations

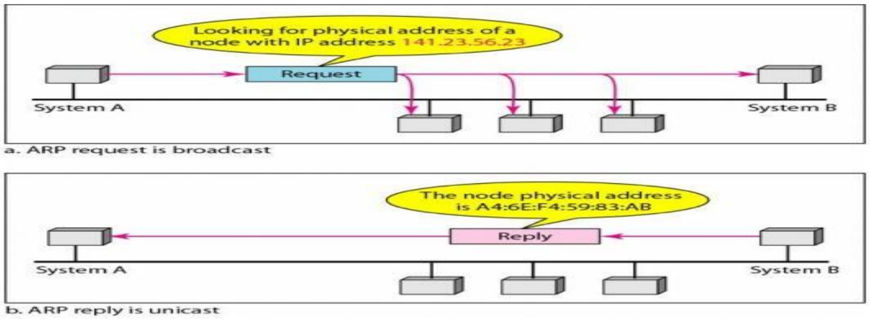
1. A machine could change its NIC, resulting in a new physical address.

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

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

To implement these changes, a static mapping table must be updated periodically. This overhead could affect network performance. In dynamic mapping each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one.

Mapping Logical to Physical Address: ARP


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B

a. ARP request is broadcast

b. ARP reply is unicast

Figure 3.27 ARP operation

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network

1. Cache Memory

Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B. It could have broadcast the IP packet itself. ARP can be useful if the ARP reply is cached because a system normally sends several packets to the same destination. A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

2. Packet Format

The fields are as follows:

🎬 **Hardware type**: This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network

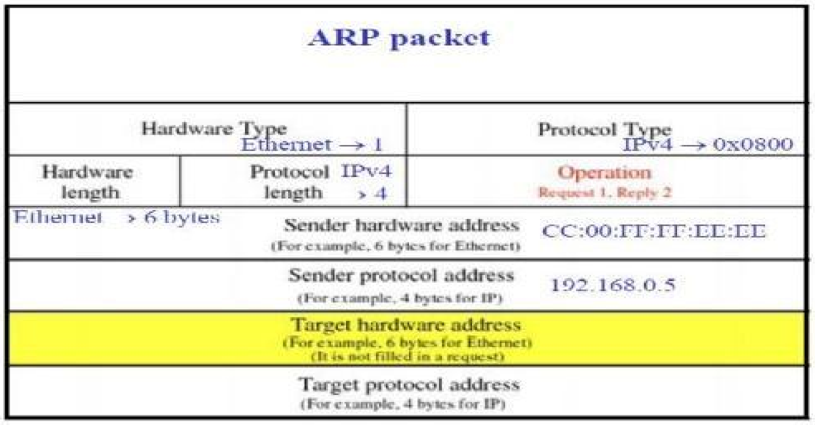


Figure 3.28 ARP Packet

Protocol type: This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.

Hardware length: This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.

Protocol length: This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.

Operation: This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).

Sender hardware address: This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.

Sender protocol address: This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.

Target hardware address: This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all Os because the sender does not know the physical address of the target.

Target protocol address: This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

3. Encapsulation

An ARP packet is encapsulated directly into a data link frame. For example, an ARP packet is encapsulated in an Ethernet frame. Note that the type field indicates that the data carried by the frame are an ARP packet

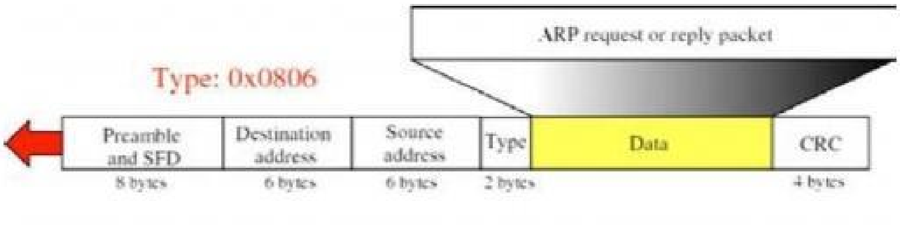


Figure 3.29 Encapsulation

Operation

The steps involved in an ARP process:

1. The sender knows the IP address of the target.

2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with 0s.

3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.

4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.

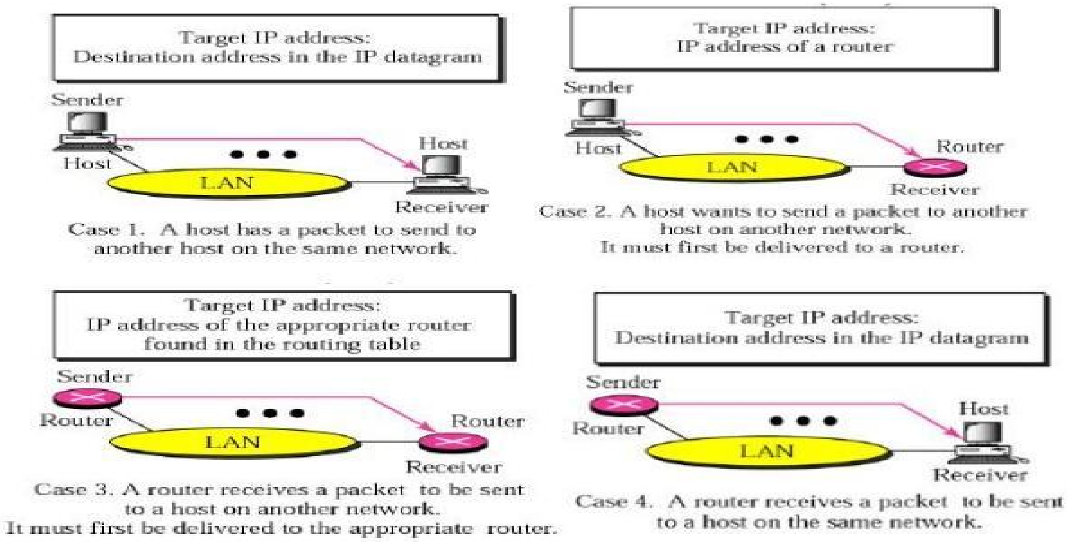
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.

6. The sender receives the reply message. It now knows the physical address of the target machine.

7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

4. Four Different Cases:

The following are four different cases in which the services of ARP can be used



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

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

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

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

Figure 3.30 Four cases using ARP

1. The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.

2. The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.

3. The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address that must be mapped to a physical address.

4. The sender is a router that has received a datagram destined for a host on the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

Example 3.17

A host with IP addresses 130.23.43.20 and physical address B2:34:55: 10:22: 10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB. The two hosts are on the same

Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

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

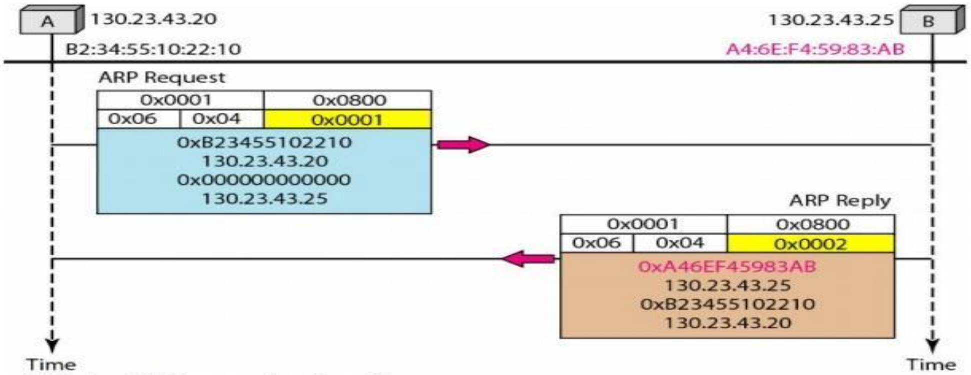


Figure 3.31 An ARP request and reply

5. Proxy ARP

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

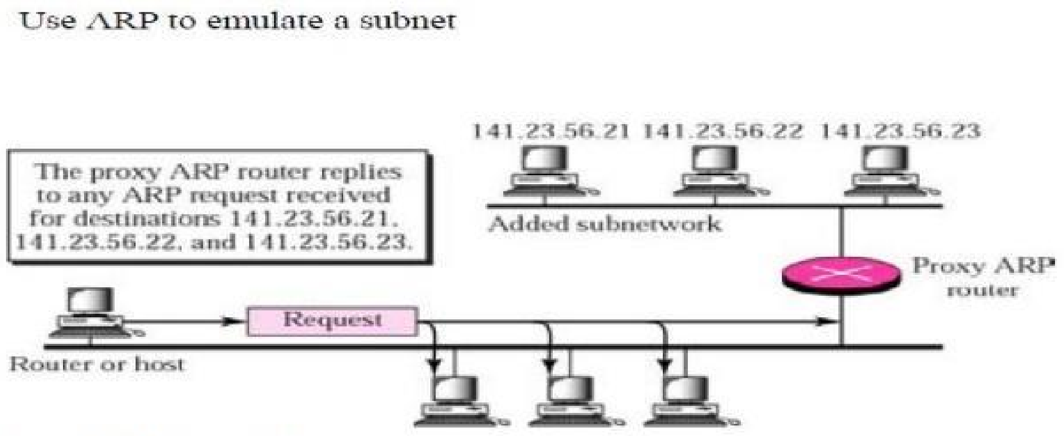

Use ARP to emulate a subnet
141.23.56.21
141.23.56.22
141.23.56.23
The proxy ARP router replies to any ARP request received for destinations 141.23.56.21, 141.23.56.22, and 141.23.56.23.
Added subnetwork
Proxy ARP router
Request
Router or host

Figure 3.32 Proxy ARP

However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses. One solution is to add a router running a proxy ARP. In this case, the router acts on behalf of all the hosts installed on the subnet. When it receives an ARP request with a target IP address that matches the address of one of its protégés (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address. When the router receives the IP packet, it sends the packet to the appropriate host. This may happen in two cases:

1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.

2. An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

Reverse Address Resolution Protocol (RARP)

Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.

However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.

The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.

There is a serious problem with RARP: Broadcasting is done at the data link layer. The physical broadcast address, all is in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete. Two protocols, BOOTP and DHCP, are replacing RARP

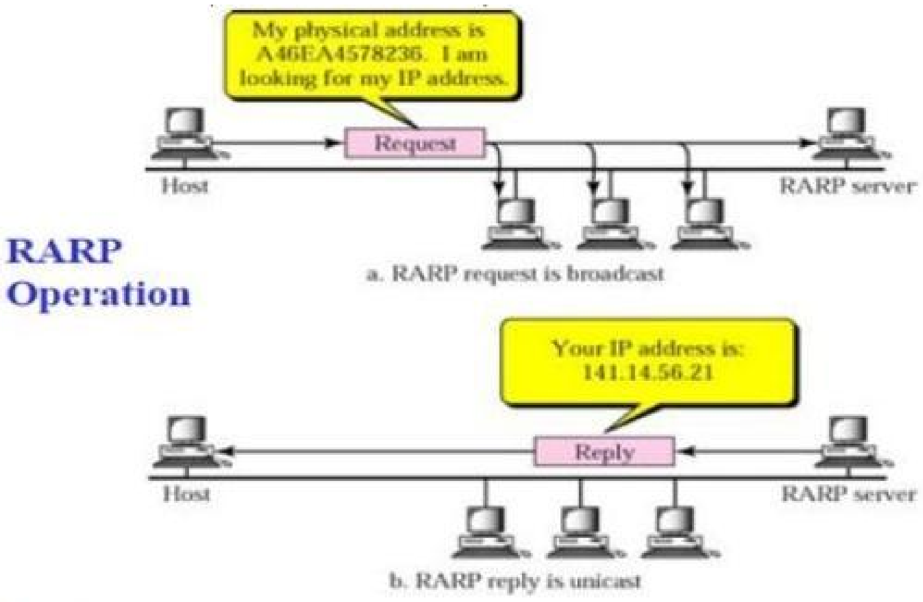
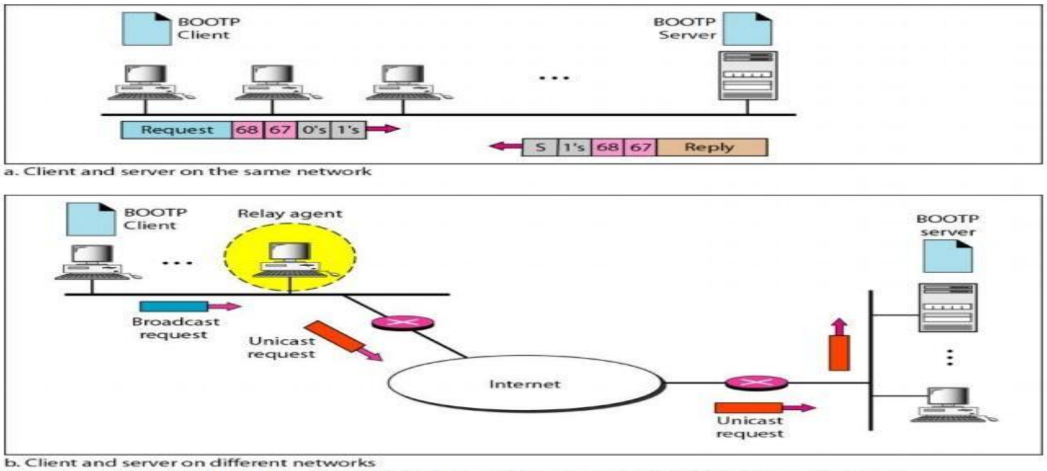

My physical address is A46EA4578236. I am looking for my IP address.
Request
Host
RARP server
RARP Operation
a. RARP request is broadcast
Your IP address is: 141.14.56.21
Reply
Host
RARP server
b. RARP reply is unicast

Figure 3.33 RARP Operation

Bootstrap Protocol (BOOTP)



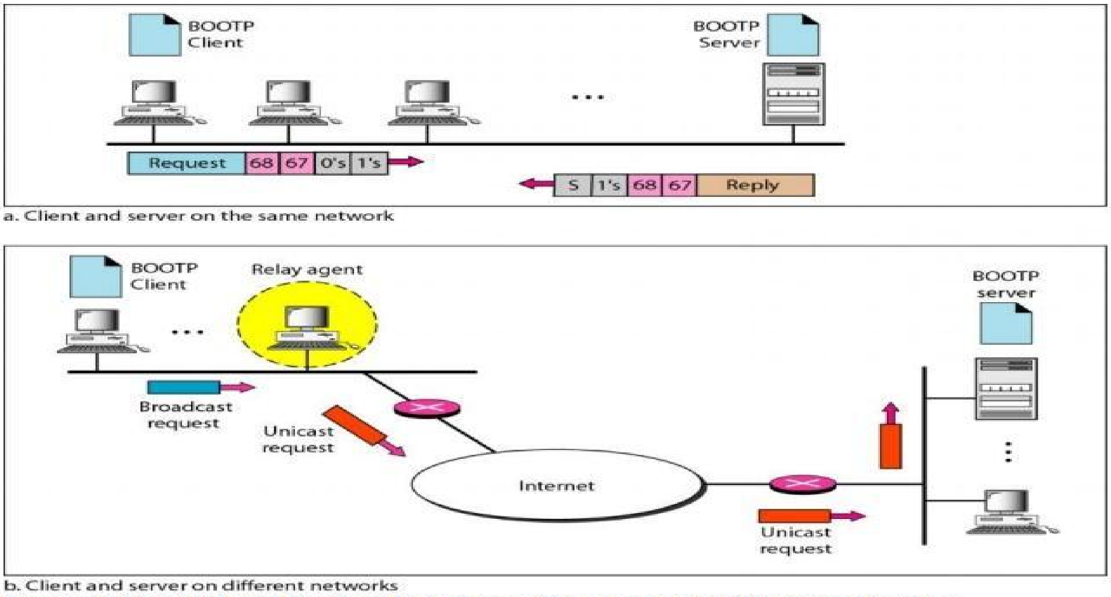
a. Client and server on the same network

b. Client and server on different networks

Figure 3.34 BOOTP client and server on the same and different networks

BOOTP

The Bootstrap Protocol (BOOTP) is a client/server protocol designed to provide physical address to logical address mapping. BOOTP is an application layer protocol. The administrator may put the client and the server on the same network or on different networks



a. Client and server on the same network

b. Client and server on different networks

Figure 3.34 BOOTP client and server on the same and different networks

One of the advantages of BOOTP over RARP is that the client and server are application-layer processes. As in other application-layer processes, a client can be in one network and the server in another, separated by several other networks. However, there is one problem that must be solved. The

BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay. The host in this case is called a relay agent. The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server. The packet, carrying a unicast destination address, is routed by any router and reaches the BOOTP server. The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the BOOTP client.

The Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) has been devised to provide static and dynamic address allocation that can be manual or automatic. DHCP provides static and dynamic address allocation that can be manual or automatic. Static Address Allocation In this capacity DHCP acts as BOOTP does. It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation: DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time. When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

Manual and Automatic Configuration:

One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes. DHCP, on the other hand, allows both manual and automatic configurations. Static addresses are created manually~ dynamic addresses are created automatically.

Internet Control Message Protocol (ICMP)

Not in syllabus but important otherwise

The IP provides unreliable and connectionless datagram delivery. It was designed this way to make efficient use of network resources. The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, it has two deficiencies: lack of error control and lack of assistance mechanisms.

The IP protocol also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive. And sometimes a network administrator needs information from another host or router. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

1. Types of Messages

ICMP messages are divided into two broad categories: error-reporting messages and query messages. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

2. Message Format

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As the first field, ICMP type, defines the type of the message. The code field specifies the reason for the particular message type. The last common field is the checksum field (to be discussed later in the chapter). The rest of the header is specific for each message type. The data section in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of the query.

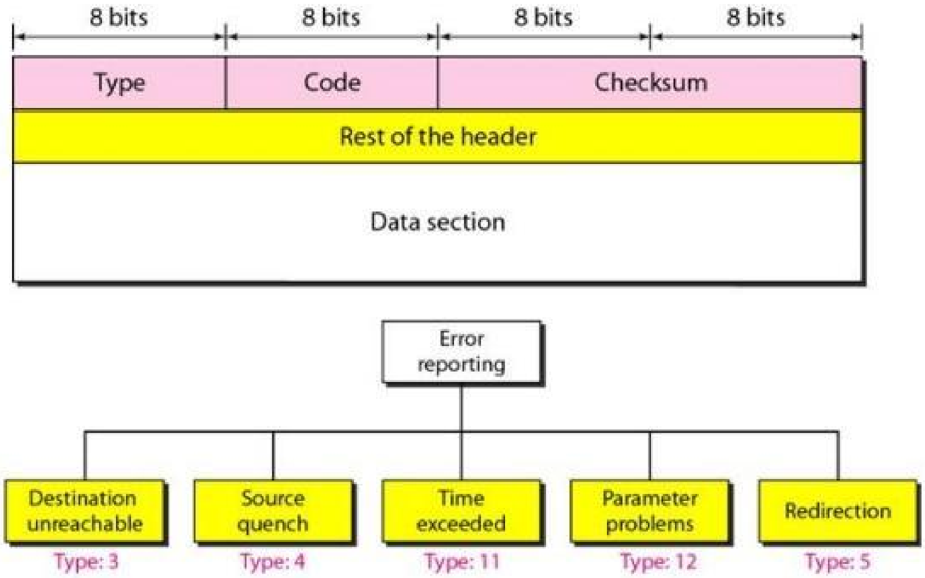


Figure 3.35 General format of ICMP messages

The Internet Group Management Protocol (IGMP)

Not in syllabus but important otherwise

The IP protocol can be involved in two types of communication: unicasting and multicasting. Unicasting is the communication between one sender and one receiver. It is a one-to-one communication. However, some processes sometimes need to send the same message to a large number of receivers simultaneously. This is called multicasting, which is a one-to-many communication. Multicasting has many applications. For example, multiple stockbrokers can simultaneously be informed of changes in a stock price, or travel agents can be informed of a plane cancellation. Some other applications include distance learning and video-on-demand.

The Internet Group Management Protocol (IGMP) is one of the necessary, but not sufficient (as we will see), protocols that is involved in multicasting. IGMP is a companion to the IP protocol.

1. Group Management

For multicasting in the Internet we need routers that are able to route multicast packets. The routing tables of these routers must be updated by using one of the multicasting routing protocols. IGMP is not a multicasting routing protocol; it is a protocol that manages group membership. In any network, there are one or more multicast routers that distribute multicast packets to hosts or other routers. The IGMP protocol gives the multicast routers information about the membership status of hosts (routers) connected to the network. A multicast router may receive thousands of multicast packets every day for different groups. If a router has no knowledge about the membership status of the hosts, it must broadcast all these packets. This creates a lot of traffic and consumes bandwidth. A better solution is to keep a list of groups in the network for which there is at least one loyal member. IGMP helps the multicast router create and update this list.

2. IGMP Messages

IOMP has gone through two versions. We discuss IGMPv2, the current version. IGMPv2 has three types of messages: the query, the membership report, and the leave report. There are two types of query messages: general and special.

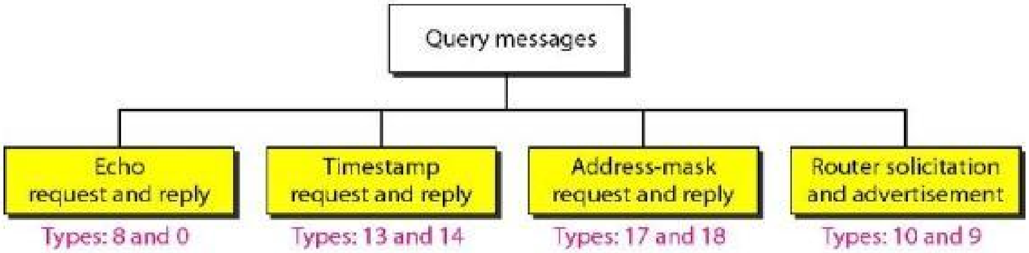


Figure 3.36 IGMP message types

3. Message Format

Figure 3.37 shows the format of an IOMP (version 2) message.

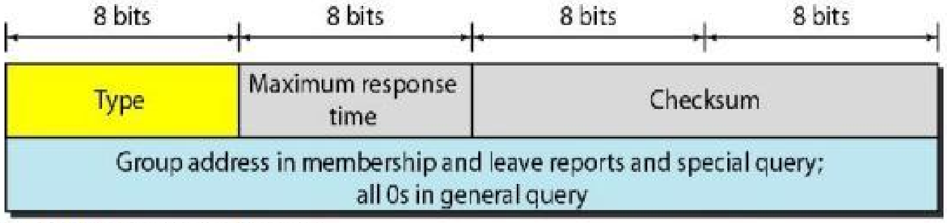


Figure 3.37 IGMP message format

- **Type.** This 8-bit field defines the type of message. The value of the type is shown in bothhexadecimal and binary notation.

- **Maximum Response Time.** This 8-bit field defines the amount of time in which a querymust be answered. The value is in tenths of a second; for example, if the value is 100, it means 10 s. The value is nonzero in the query message; it is set to zero in the other two message types.

- **Checksum**. This is a 16-bit field carrying the checksum. The checksum is calculated overthe 8-byte message

- **Group address.**The value of this field is 0 for a general query message. The value defines the group id (multicast address of the group) in the special query, the membership report, and the leave report messages.

4. IGMP Operation

IGMP operates locally. A multicast router connected to a network has a list of multicast addresses of the groups with at least one loyal member in that network. For each group, there is one router that has the duty of distributing the multicast packets destined for that group. This means that if there are three multicast routers connected to a network, their lists of group ids are mutually exclusive.

4.1 Joining a Group

A host or a router can join a group. A host maintains a list of processes that have membership in a group. When a process wants to join a new group, it sends its request to the host. The host adds the name of the process and the name of the requested group to its list. If this is the first entry for this particular group, the host sends a membership report message. If this is not the first entry, there is no need to send the membership report since the host is already a member of the group; it already receives multicast packets for this group. The protocol requires that the membership report be sent twice, one after the other within a few moments. In IGMP, a membership report is sent twice, one after the other.

4.2 Leaving a Group

When a host sees that no process is interested in a specific group, it sends a leave report. Similarly, when a router sees that none of the networks connected to its interfaces is interested in a specific group, it sends a leave report about that group. However, when a multicast router receives a leave report, it cannot immediately purge that group from its list because the report comes from just one host or router; there may be other hosts or routers that are still interested in that group. To make sure, the router sends a special query message and inserts the group id, or multicast address, related to the group. The router allows a specified time for any host or router to respond. If, during this time, no interest (membership report) is received, the router assumes that there are no loyal members in the network and purges the group from its list.

4.3 Monitoring Membership

A host or router can join a group by sending a membership report message. It can leave a group by sending a leave report message. However, sending these two types of reports is not enough. Consider the situation in which there is only one host interested in a group, but the host is shut down or removed from the system. The multicast router will never receive a leave report. How is this handled? The multicast router is responsible for monitoring all the hosts or routers in a LAN to see if they want to continue their membership in a group.

4.4 Delayed Response

To prevent unnecessary traffic, IGMP uses a delayed response strategy. When a host or router receives a query message, it does not respond immediately; it delays the response. Each host or router uses a random number to create a timer, which expires between 1 and l0s. The expiration time can be in steps of 1 s or less. A timer is set for each group in the list.

ICMPv6 (Internetworking Control Message Protocol)

Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP (ICMPv6). This new version follows the same strategy and purposes of version 4. ICMPv4 has been modified to make it more suitable for IPv6. In addition, some protocols that were independent in version 4 are now part of Internetworking Control Message Protocol (ICMPv6).

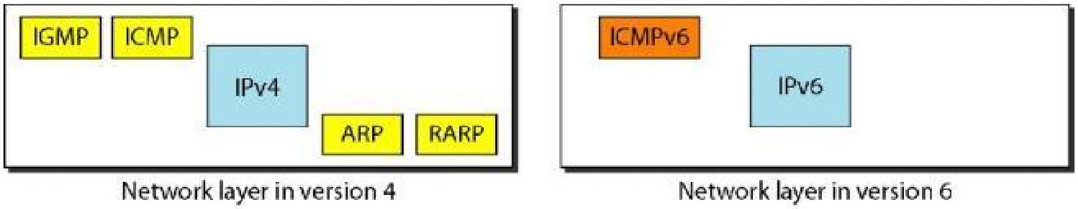


Figure 3.38 Comparison of network layers in version 4 and version 6

The ARP and IGMP protocols in version 4 are combined in ICMPv6. The RARP protocol is dropped from the suite because it was rarely used and BOOTP has the same functionality. Just as in ICMPv4, we divide the ICMP messages into two categories. However, each category has more types of messages than before.

Delivery of Packets in Network

The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet.

Direct Versus Indirect Delivery

The delivery of a packet to its final destination is accomplished by using two different methods of delivery, direct and indirect

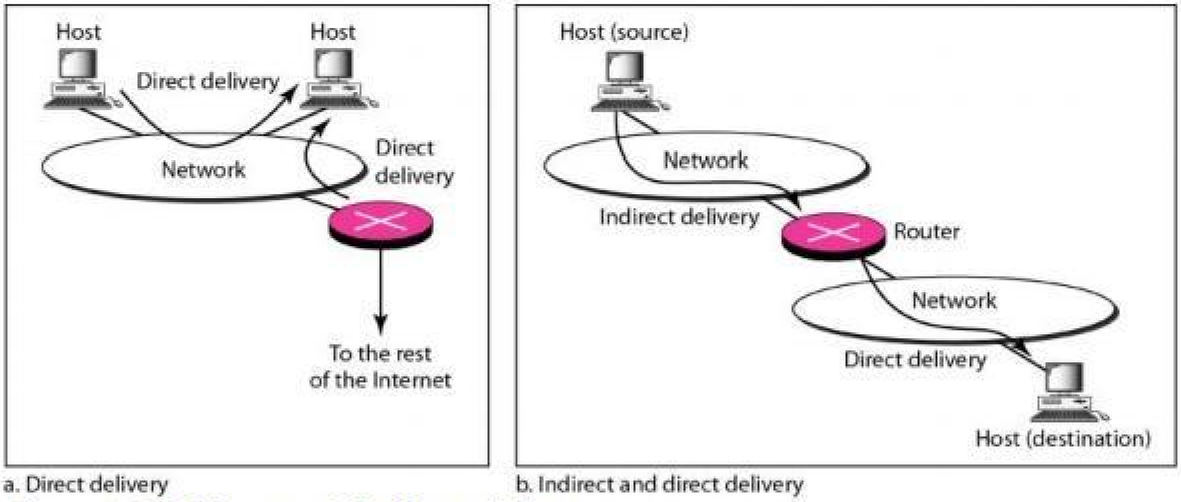


Figure 3.39 Direct and indirect delivery

Direct Delivery

In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host. The sender can easily determine if the delivery is direct. It can extract the network address of the destination (using the mask) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

Indirect Delivery

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination.

Forwarding Techniques

Forwarding

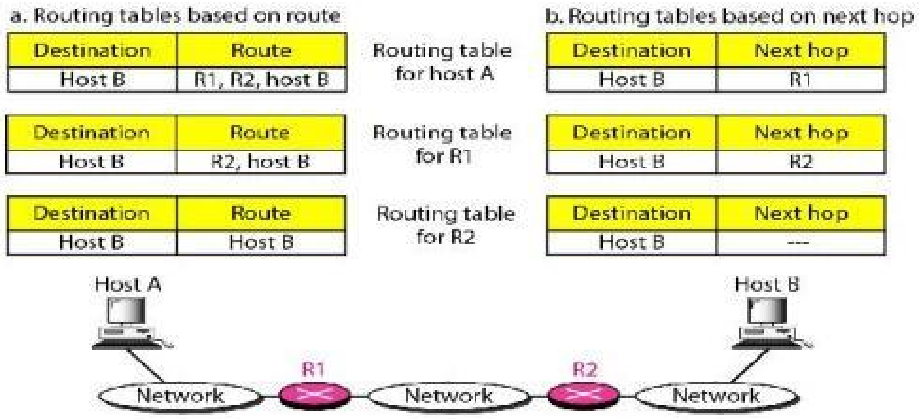
Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this 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.

Forwarding Techniques

Several techniques can make the size of the routing table manageable and also handle issues such as security.

a. 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). The entries of a routing table must be consistent with one another.



a. Routing tables based on route

Destination	Route
Host B	R1, R2, host B

Routing table for host A

Destination	Route
Host B	R2, host B

Routing table for R1

Destination	Route
Host B	Host B

Routing table for R2

b. Routing tables based on next hop

Destination	Next hop
Host B	R1

Destination	Next hop
Host B	R2

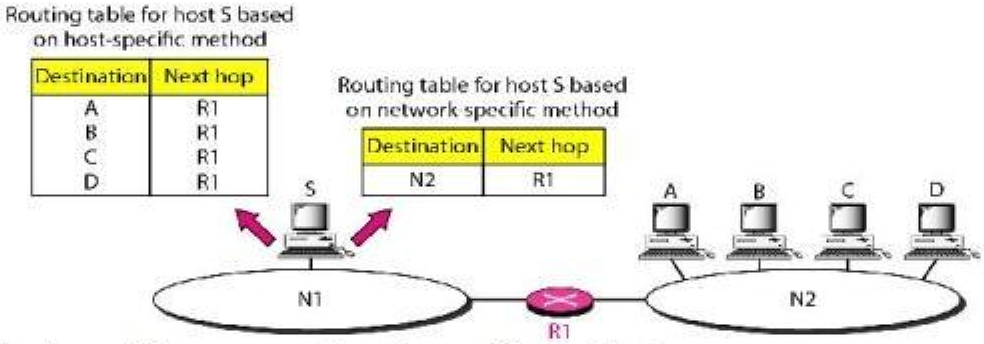
Destination	Next hop
Host B	---

Figure 3.40 Route method versus next-hop method

b. 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.

Host-specific routing is used for purposes such as checking the route or providing security measures



Routing table for host S based on host-specific method

Destination	Next hop
A	R1
B	R1
C	R1
D	R1

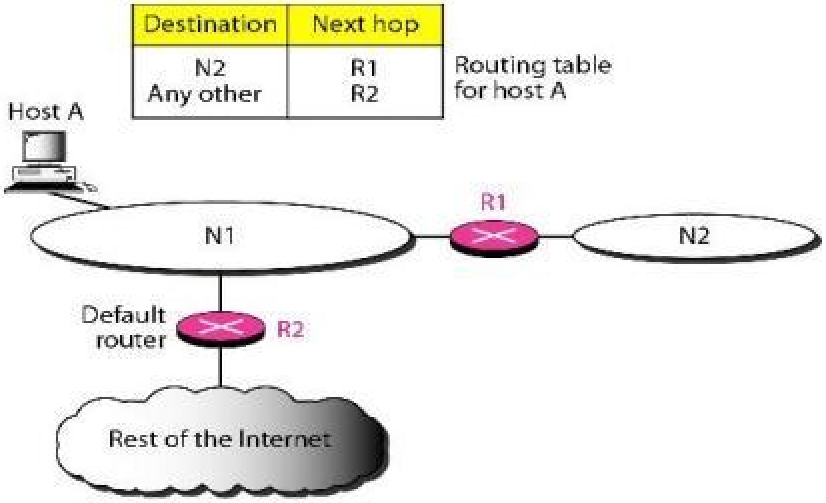
Routing table for host S based on network specific method

Destination	Next hop
N2	R1

Figure 3.41 Host-specific versus network-specific method

c. Default Method

Another technique to simplify routing is called the default method. Host A is connected to a network with two routers. Router R1 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 (normally defined as network address 0.0.0.0).



Destination	Next hop
N2	R1
Any other	R2

Routing table for host A

Figure 3.42 Default method

Example 3.18

Make a routing table for router R1, using the configuration in Figure 3.43

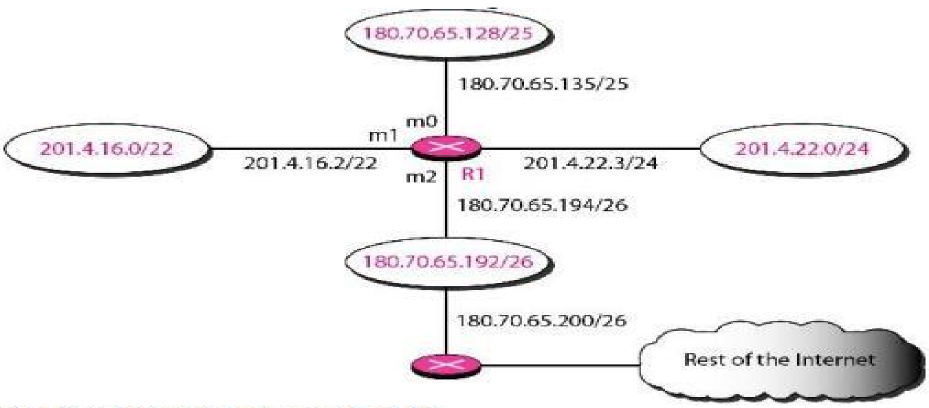
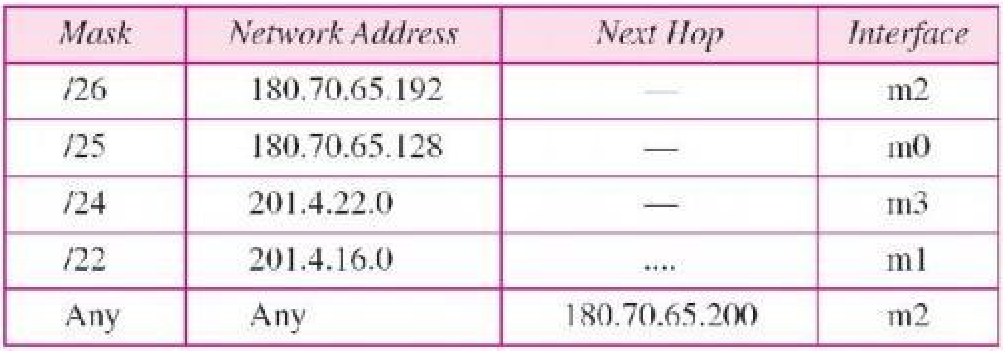


Figure 3.43 Configuration for Example 3.18
Solution

Table 3.9 Routing table for router R1 in Figure 3.43



Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	—	m2
/25	180.70.65.128	—	m0
/24	201.4.22.0	—	m3
/22	201.4.16.0	....	m1
Any	Any	180.70.65.200	m2

Example 3.19

Show the forwarding process if a packet arrives at R1 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

1. The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.

2. The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address (the destination address of the packet in this case) and the interface number m0 are passed to ARP for further processing.

Example 3.20

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

Solution

The router performs the following steps:

- The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 1).
- The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).
- The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address. The destination address of the packet and the interface number m3 are passed to ARP.

Example 3.21

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

Solution

This time all masks are applied, one by one, to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP. This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Unicast Routing Protocols

A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet. Today, an internet needs dynamic routing tables. The tables need to be updated as soon as there is a change in the internet. For instance, they need to be updated when a router is down, and they need to be updated whenever a better route has been found.

1. Optimization

A router receives a packet from a network and passes it to another network. A router is usually attached to several networks. One approach is to assign a cost for passing through a network. We call this cost a metric. However, the metric assigned to each network depends on the type of protocol. Some simple protocols, such as the Routing Information Protocol (RIP), treat all networks as equals. The cost of passing through a network is the same; it is one hop count. So if a packet passes through 10 networks to reach the destination, the total cost is 10 hop counts.

2. Intra- and Inter-domain Routing

An internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. An autonomous system (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intradomain routing. Routing between autonomous systems is referred to as interdomain routing

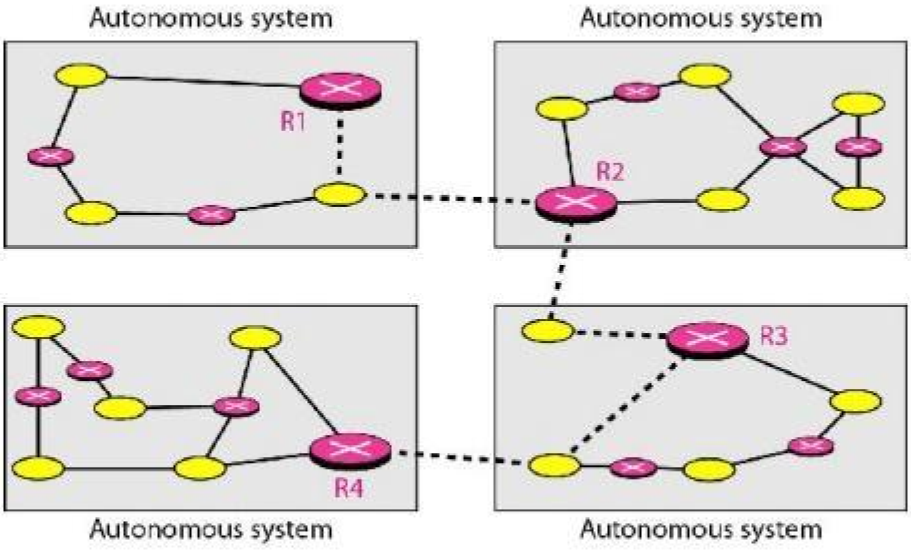


Figure 3.43 Autonomous systems

Several intra-domain and inter-domain routing protocols are in use.

O Two intra-domain routing protocols: Distance vector and link state.

O One inter-domain routing protocol: path vector.

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

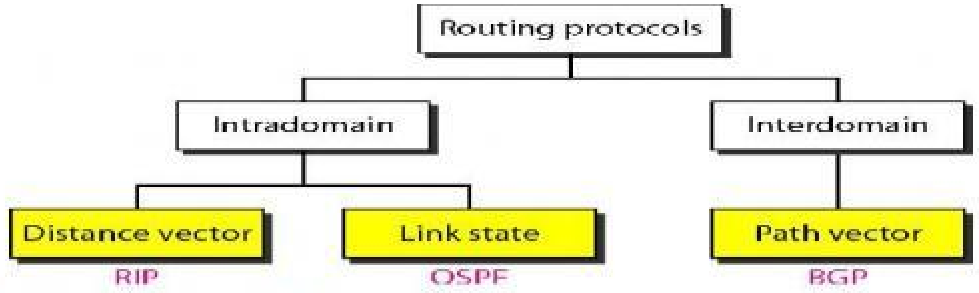
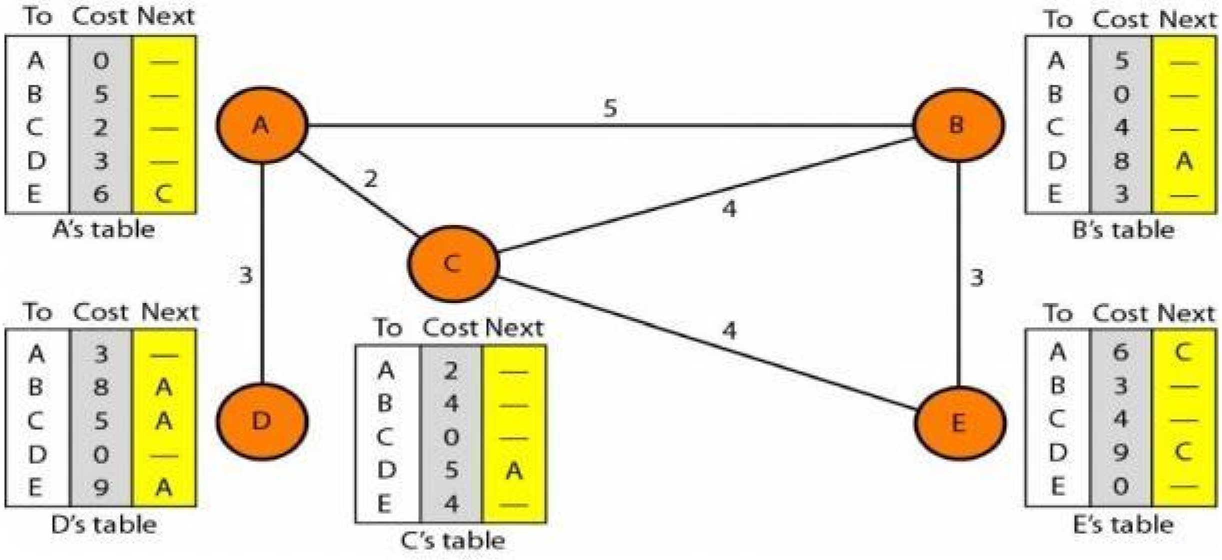


Figure 3.44 Popular routing protocols

3. Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).



To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's table

To	Cost	Next
A	5	—
B	0	—
C	4	—
D	8	A
E	3	—

B's table

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

D's table

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

C's table

To	Cost	Next
A	6	C
B	3	—
C	4	—
D	9	C
E	0	—

E's table

Figure 3.45 Distance vector routing tables

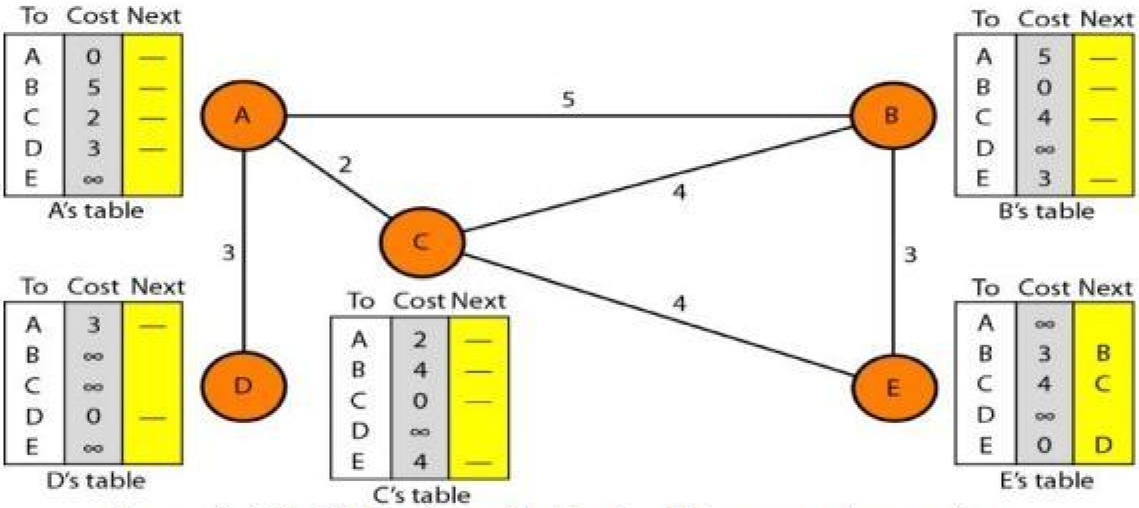
The table for node A shows how we can reach any node from this node. For example, our least cost to reach node E is 6. The route passes through C.

Initialization

The tables in Figure 3.45 are stable; each node knows how to reach any other node and the cost. At the beginning, however, this is not the case. Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.



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

A's table

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

B's table

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

D's table

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

C's table

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

E's table

Figure 3.46 Initialization of tables in distance vector routing

Updating

When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

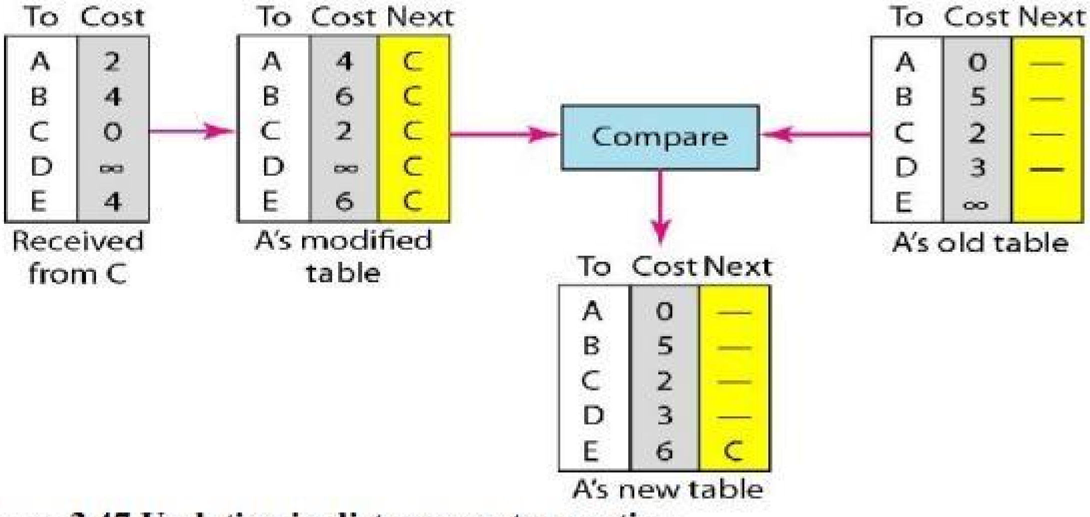
1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is *x* mi, and the distance between A and C is *y* mi, then the distance between A and that destination, via C, is $x + y$ mi.

2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.

3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a) If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b) If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3.



To	Cost
A	2
B	4
C	0
D	∞
E	4

Received from C

To	Cost	Next
A	4	C
B	6	C
C	2	C
D	∞	C
E	6	C

A's modified table

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

A's old table

To	Cost	Next
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

A's new table

Figure 3.47 Updating in distance vector routing

Two-Node Loop Instability

A problem with distance vector routing is instability, which means that a network using this protocol can become unstable. To understand the problem, let us look at the scenario depicted.

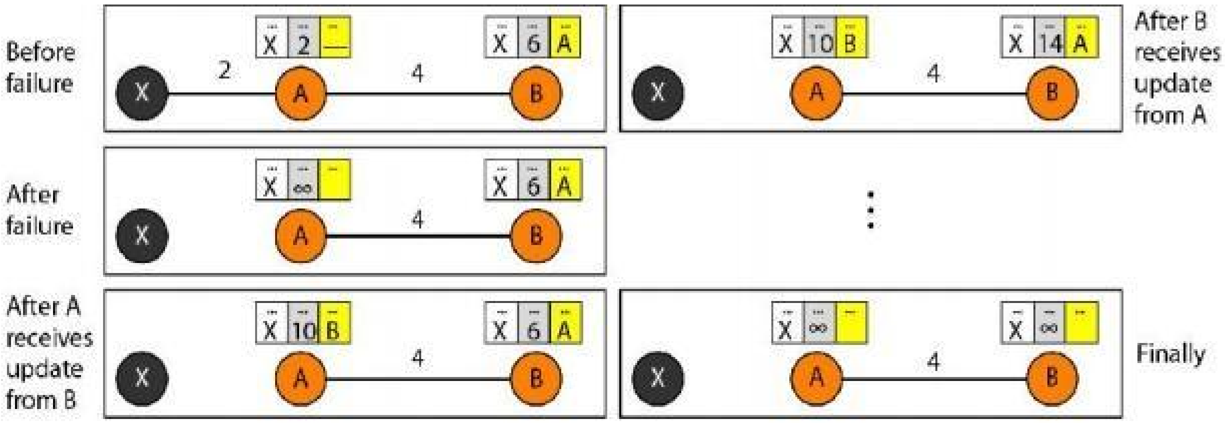


Figure 3.48 Two-node instability

Defining Infinity The first obvious solution is to redefine infinity to a smaller number, such as100. For our previous scenario, the system will be stable in less than 20 update s. As a matter of fact, most implementations of the distance vector protocol define the distance between each node to be I and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.

Split Horizon Another solution is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows). Taking information from node A, modifying it, and sending it back to node A creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X.

4. Link State Routing

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

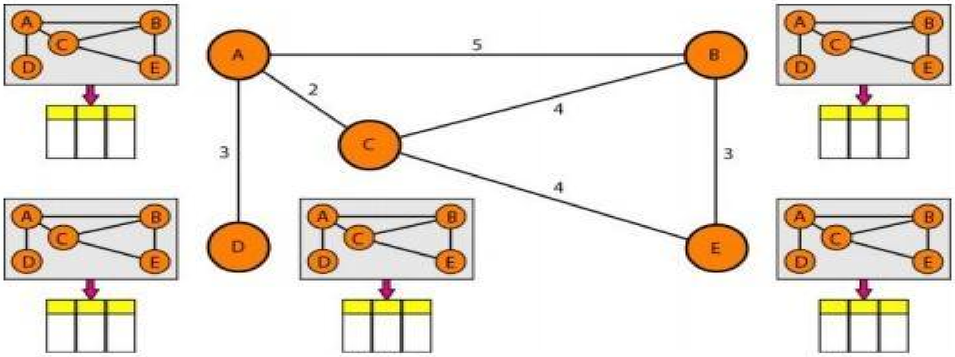


Figure 3.50 Concept of link state routing

Figure 3.50 Concept of link state routing

The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination

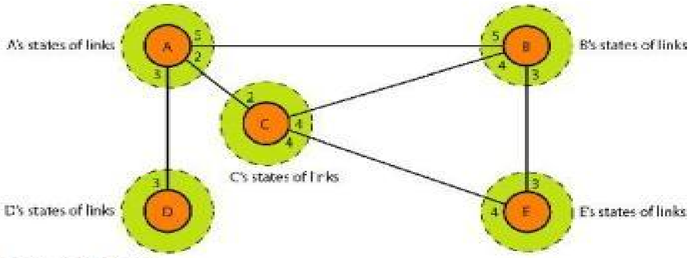


Figure 3.51 Link state knowledge

Building Routing Tables:

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

a) Creation of the states of the links by each node, called the link state packet (LSP).

b) Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.

c) Formation of a shortest path tree for each node.

d) Calculation of a routing table based on the shortest path tree.

Types of Links

In OSPF terminology, a connection is called a *link*. Four types of links have been defined: point-to-point, transient, stub, and virtual.

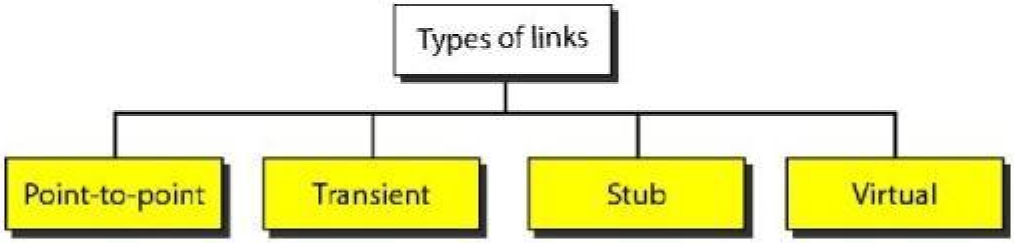


Figure 3.52 Types of links

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

5. Path Vector Routing

Distance vector and link state routing are both intra-domain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for inter-domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Path vector routing proved to be useful for inter-domain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node in each autonomous system that acts on behalf of the entire autonomous system.

Initialization

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

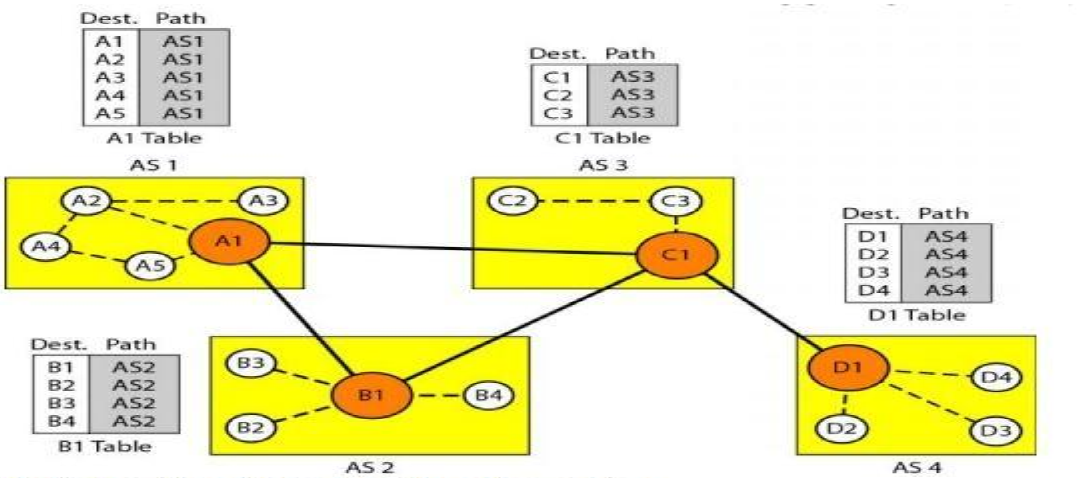


Figure 3.53 Initial routing tables in path vector routing

Figure 3.53 Initial routing tables in path vector routing

Node Al is the speaker node for AS1, B1 for AS2, C1 for AS3, and Dl for AS4. Node Al creates an initial table that shows Al to A5 are located in ASI and can be reached through it. Node B1 advertises that Bl to B4 are located in AS2 and can be reached through Bl. And so on

Multicast Routing Protocols

Not in syllabus but important otherwise

Unicast, Multicast, and Broadcast:

A message can be unicast, multicast, or broadcast.

1. Unicasting

In unicast communication, there is one source and one destination. The relationship between the source and the destination is one-to-one. In this type of communication, both the source and destination addresses, in the IP datagram, are the unicast addresses assigned to the hosts (or host interfaces, to be more exact).

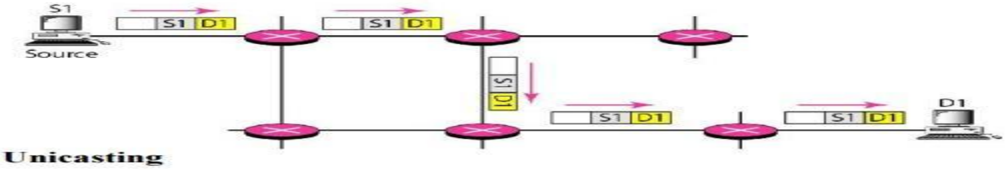


Figure 3.54 Unicasting

2. Multicasting

In multicast communication, there is one source and a group of destinations. The relationship is one-to-many. In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations. The group address identifies the members of the group.

A multicast packet starts from the source S1 and goes to all destinations that belong to group G1. In multicasting, when a router receives a packet, it may forward it through several of its interfaces.

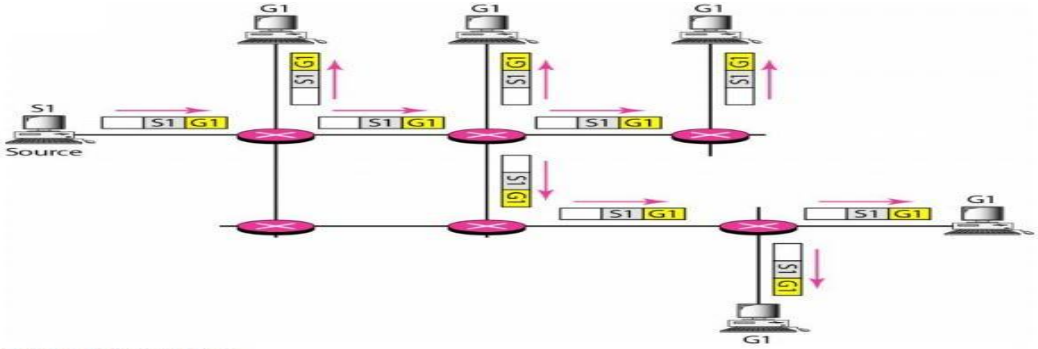


Figure 3.55 Multicasting

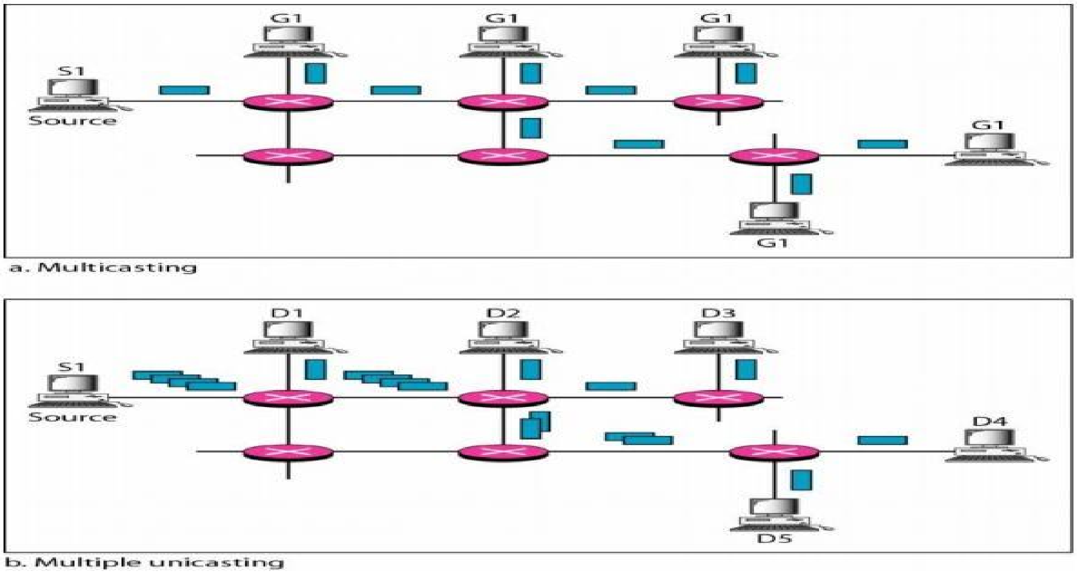
3. Broadcasting

In broadcast communication, the relationship between the source and the destination is one-to-all. There is only one source, but all the other hosts are the destinations. The Internet does not explicitly support broadcasting because of the huge amount of traffic it would create and because of the bandwidth it would need. Imagine the traffic generated in the Internet if one person wanted to send a message to everyone else connected to the Internet.

Multicasting versus Multiple Unicasting

Multicasting starts with one single packet from the source that is duplicated by the routers. The destination address in each packet is the same for all duplicates. Note that only one single copy of the packet travels between any two routers.

In multiple unicasting, several packets start from the source. If there are five destinations, for example, the source sends five packets, each with a different unicast destination address. Note that there may be multiple copies traveling between two routers.



a. Multicasting

b. Multiple unicasting

Figure 3.56 Multicasting versus multiple unicasting

4. Applications

Multicasting has many applications today such as access to distributed databases, information dissemination, teleconferencing, and distance learning.

5. Access to Distributed Databases

Most of the large databases today are distributed. That is, the information is stored in more than one location, usually at the time of production. The user who needs to access the database does not know the location of the information. A user's request is multicast to all the database locations, and the location that has the information responds.

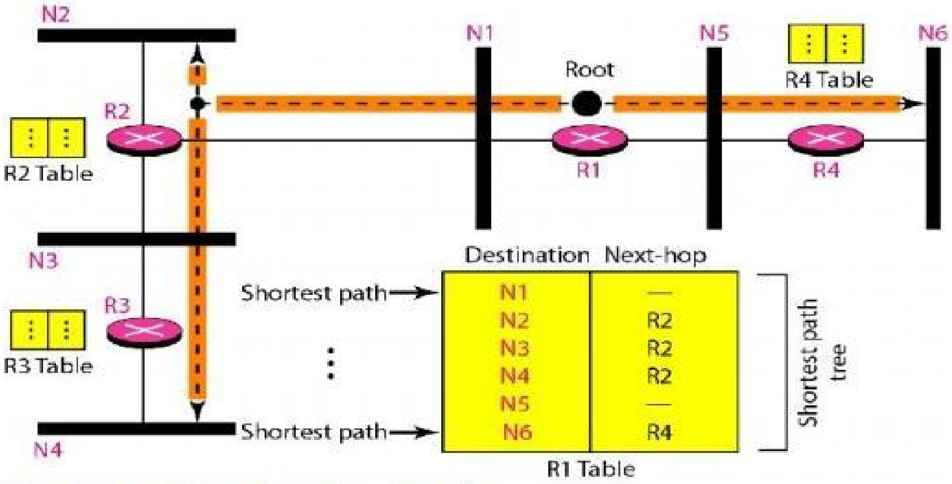
6. Information Dissemination

Businesses often need to send information to their customers. If the nature of the information is the same for each customer, it can be multicast. In this way a business can send one message that can reach many customers. For example, a software update can be sent to all purchasers of a particular software package.

7. Optimal Routing: Shortest Path Trees

The process of optimal inter domain routing eventually results in the finding of the shortest path tree. The root of the tree is the source, and the leaves are the potential destinations. The path from the root to each destination is the shortest path. However, the number of trees and the formation of the trees in unicast and multicast routing are different.

Unicast Routing: In unicast routing, when a router receives a packet to forward, it needs to find the shortest path to the destination of the packet. The router consults its routing table for that particular destination. The next-hop entry corresponding to the destination is the start of the shortest path. The router knows the shortest path for each destination, which means that the router has a shortest path tree to optimally reach all destinations. In other words, each line of the routing table is a shortest path; the whole routing table is a shortest path tree. In unicast routing, each router needs only one shortest path tree to forward a packet; however, each router has its own shortest path tree.



Destination	Next-hop
N1	—
N2	R2
N3	R2
N4	R2
N5	—
N6	R4

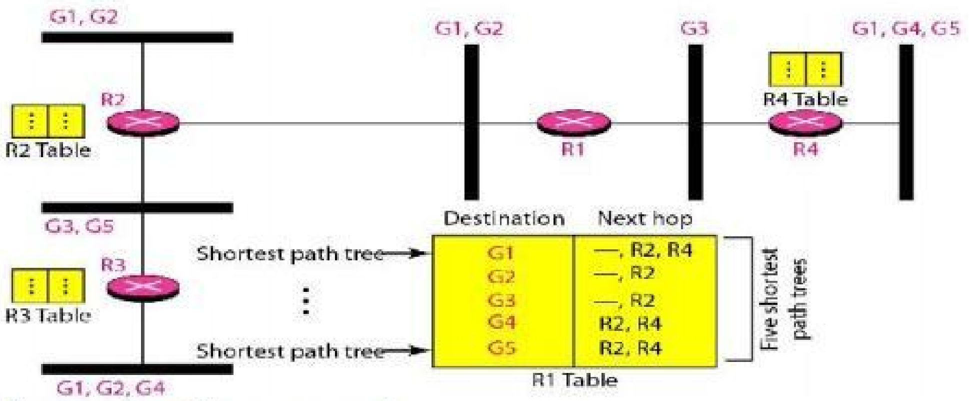
R1 Table

Figure 3.57 Shortest path tree in unicast routing

Multicast Routing: When a router receives a multicast packet, the situation is different from when it receives a unicast packet. A multicast packet may have destinations in more than one network. Forwarding of a single packet to members of a group requires a shortest path tree. If we have n groups, we may need *n* shortest path trees. We can imagine the complexity of multicast routing. Two approaches have been used to solve the problem: source-based trees and group-shared trees.

a. Source-Based Tree: In the source-based tree approach, each router needs to have one shortest path tree for each group. The shortest path tree for a group defines the next hop for each network that has loyal member(s) for that group. Five groups in the domain: GI, G2, G3, G4, and G5.

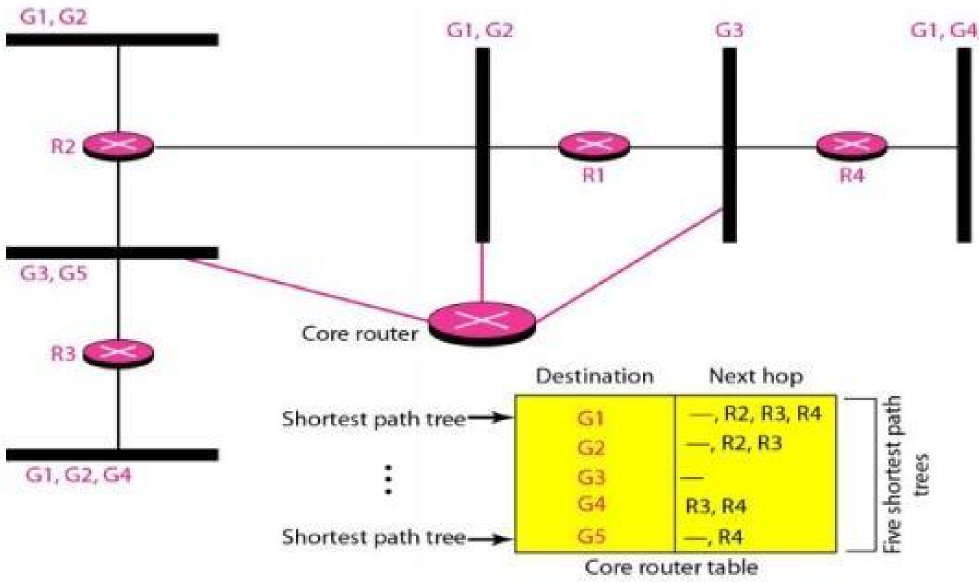
At the moment G1 has loyal members in four networks, G2 in three, G3 in two, G4 in two, and G5 in two. We have shown the names of the groups with loyal members on each network. There is one shortest path tree for each group; therefore there are five shortest path trees for five groups



Destination	Next hop
G1	—, R2, R4
G2	—, R2
G3	—, R2
G4	R2, R4
G5	R2, R4

Figure 3.58 Source-based tree approach

b. Group-Shared Tree:. In the group-shared tree approach, instead of each router having *m* shortest path trees, only one designated router, called the center core, or rendezvous router, takes the responsibility of distributing multicast traffic. The core has *m* shortest path trees in its routing table. The rest of the routers in the domain have none. If a router receives a multicast packet, it encapsulates the packet in a unicast packet and sends it to the core router. The core router removes the multicast packet from its capsule, and consults its routing table to route the packet.



Destination	Next hop
G1	—, R2, R3, R4
G2	—, R2, R3
G3	—
G4	R3, R4
G5	—, R4

Figure 3.59 Group-shared tree approach

8. Routing Protocols

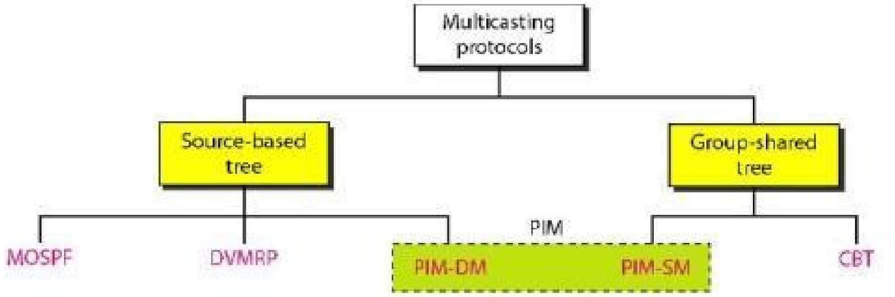


Figure 3.60 Taxonomy of common multicast protocols

a. Multicast Link State Routing: MOSPF

Multicast link state routing uses the source-based tree approach. links. For multicast routing, a node needs to revise the interpretation of *state.* A node advertises every group which has any loyal member on the link. Here the meaning of state is "what groups are active on this link." The information about the group comes from IGMP. Each router running IGMP solicits the hosts on the link to find out the membership status.

MOSPF Multicast Open Shortest Path First (MOSPF) protocol is an extension of theOSPF protocol that uses multicast link state routing to create source-based trees. The protocol requires a new link state update packet to associate the unicast address of a host with the group address or addresses the host is sponsoring. This packet is called the group-membership LSA.

b. Multicast Distance Vector: DVMRP

Multicast Distance Vector Routing Unicast distance vector routing is very simple; extending itto support multicast routing is complicated. Multicast routing does not allow a router to send its routing table to its neighbors. The idea is to create a table from scratch by using the information from the unicast distance vector tables.

Multicast distance vector routing uses source-based trees, but the router never actually makes a routing table. When a router receives a multicast packet, it forwards the packet as though it is consulting a routing table.

a) **Flooding**

b) Reverse Path Forwarding (RPF)

c) Reverse Path Broadcasting (RPB)

d) Reverse Path Multicasting (RPM)

DVMRP The Distance Vector Multicast Routing Protocol (DVMRP) is an implementation of multicast distance vector routing. It is a source-based routing protocol, based on RIP.

c. CBT

The Core-Based Tree (CBT) protocol is a group-shared protocol that uses a core as the root of the tree. The autonomous system is divided into regions, and a core (center router or rendezvous router) is chosen for each region.

The Core-Based Tree (CBT) is a group-shared tree, center-based protocol using one tree per group. One of the routers in the tree is called the core. A packet is sent from the source to members of the group following this procedure:

a) The source, which may or may not be part of the tree, encapsulates the multicast packet inside a unicast packet with the unicast destination address of the core and sends it to the core. This part of delivery is done using a unicast address; the only recipient is the core router.

b) The core decapsulates the unicast packet and forwards it to all interested intetfaces.

c) Each router that receives the multicast packet, in tum, forwards it to all interested interfaces.

PIM

Protocol Independent Multicast (PIM) is the name given to two independent multi cast routing protocols: Protocol Independent Multicast, Dense Mode (PIM-DM) and Protocol Independent Multicast, Sparse Mode (PIM-SM). Both protocols are unicast protocol-dependent, but the similarity ends here.

PIM-DM

PIM-DM is used when there is a possibility that each router is involved in multicasting (dense mode). In this environment, the use of a protocol that broadcasts the packet is justified because almost all routers are involved in the process. PIM-DM is a source-based tree routing protocol that uses RPF and pruning and grafting strategies for multicasting. Its operation is like that of DVMRP.

PIM-SM

PIM-SM is used when there is a slight possibility that each router is involved in multicasting (sparse mode). In this environment, the use of a protocol that broadcasts the packet is not justified; a protocol such as CBT that uses a group-shared tree is more appropriate. PIM-SM is used in a sparse multicast environment such as a WAN. PIM-SM is a group-shared tree routing protocol that has a rendezvous point (RP) as the source of the tree.