CHAPTER 18

TCP/IP

The **Transmission Control Protocol/Internetworking Protocol (TCP/IP)** is a set of protocols, or a protocol suite, that defines how all transmissions are exchanged across the Internet. Named after its two most popular protocols, TCP/IP has been in active use for many years and has demonstrated its effectiveness on a worldwide scale.

18.1 OVERVIEW OF TCP/IP

In 1969, a project was funded by the **Advanced Research Project Agency (ARPA),** an arm of the U.S. Department of Defense. ARPA established a packet-switching network of computers linked by point-to-point leased lines called **Advanced Research Project Agency Network (ARPANET)** that provided a basis for early research into networking. The conventions developed by ARPA to specify how individual computers could communicate across that network became TCP/IP.

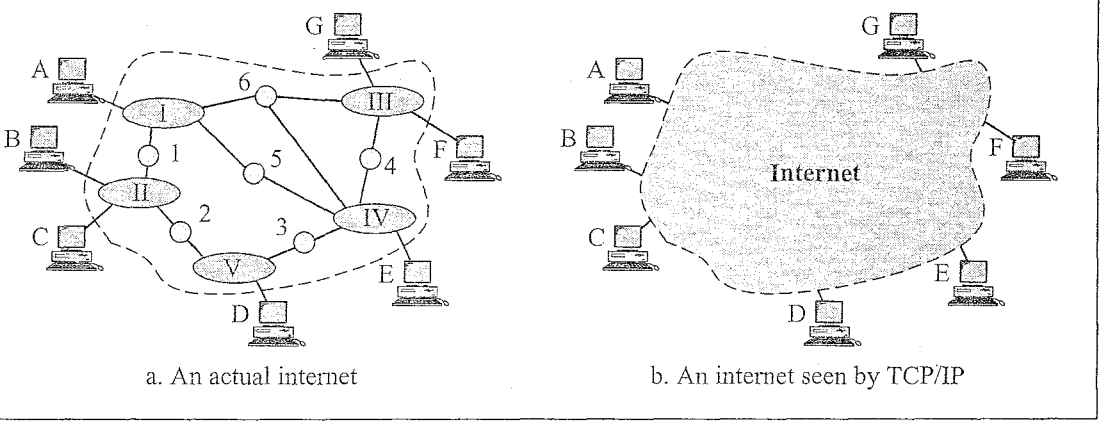
As networking possibilities grew to include other types of links and devices, ARPA adapted TCP/IP to the demands of the new technology. As involvement in TCP/IP grew, the scope of ARPANET expanded until it became the backbone of an internetwork today referred to as the Internet.

TCP/IP and the Internet

TCP/IP and the concept of internetworking developed together, each shaping the growth of the other. Before moving more deeply into the protocols, however, we need to understand how TCP/IP relates to the physical entity of any internet it serves.

An internet under TCP/IP operates like a single network connecting many computers of any size and type. Internally, an internet (or, more specifically, the Internet) is an interconnection of independent physical networks (such as LANs) linked together by internetworking devices. Figure 18.1 shows the topology of a possible internet. In this example, the letters A, B, C, and so on represent hosts. A **host** in TCP/IP is a computer. The solid circles in the figure, numbered 1, 2, 3, and so on, are routers or gateways. The larger ovals containing roman numerals (I, II, III, etc.) represent separate physical networks.

Figure 18.1 *An internet according to TCP/IP*



To TCP/IP, the same internet appears quite differently (see again Figure 18.1). TCP/IP considers all interconnected physical networks to be one huge network. It considers all of the hosts to be connected to this larger logical network rather than to their individual physical networks.

TCP/IP and OSI

Transmission Control Protocol (TCP) was developed before the OSI model. Therefore, the layers in the TCP/IP protocol do not match exactly with those in the OSI model. The TCP/IP protocol is made up of five layers: physical, data link, network, transport, and application. The application layer in TCP/IP can be equated to the combination of session, presentation, and application layers of the OSI model.

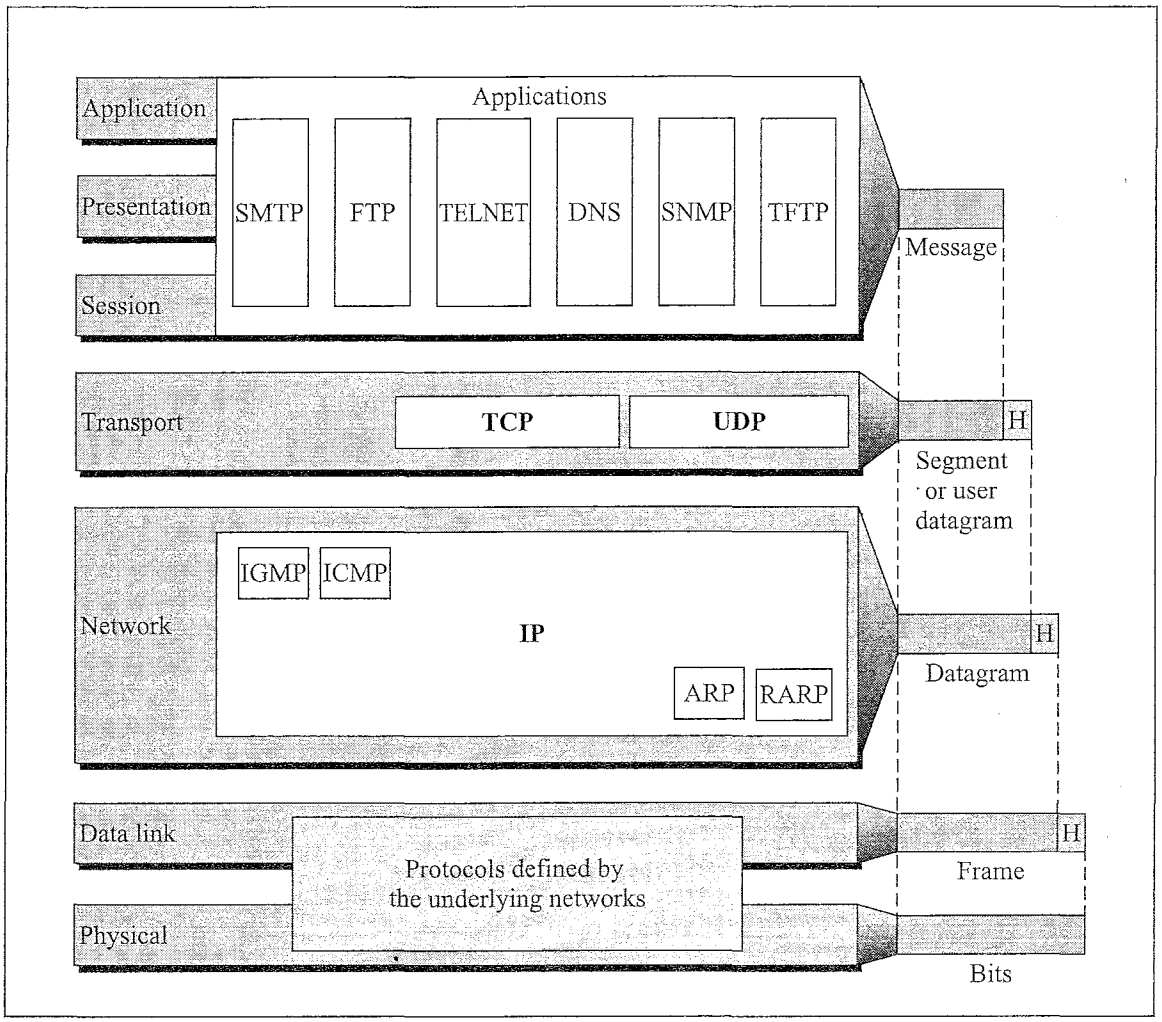
At the transport layer, TCP/IP defines two protocols: TCP and **User Datagram Protocol (UDP).** At the network layer, the main protocol defined by TCP/IP is **Internetworking Protocol (IP),** although there are some other protocols that support data movement in this layer.

At the physical and data link layers, TCP/IP does not define any specific protocol. It supports all of the standard and proprietary protocols discussed earlier in this book. A network in a TCP/IP internetwork can be a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN).

Encapsulation

Figure 18.2 shows the **encapsulation** of data units at different layers of the TCP/IP protocol suite. The data unit created at the application layer is called a *message.* TCP or UDP creates a data unit that is called either a **segment** or a **user datagram.** The IP layer in turn will create a data unit called a **datagram.** The movement of the datagram across the Internet is the responsibility of the TCP/IP protocol. However, to be able to move physically from one network to another, the datagram must be encapsulated in a frame in the data link layer of the underlying network and finally transmitted as signals along the transmission media.

Figure 18.2 *TCP/IP and OSI model*



18.2 NETWORK LAYER

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the internetwork protocol (IP). IP, in turn, contains four supporting protocols: ARP, RARP, ICMP, and IGMP. Each of these protocols is described later in this chapter.

Internetwork Protocol (IP)

IP is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless datagram protocol—a best-effort delivery service. The term *best-effort* means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. As we have seen in previous chapters, transmissions along physical networks can be destroyed for a number of reasons. Noise can cause bit errors during transmission across a medium; a congested router may discard a datagram if it is unable to relay it before a time limit runs out; routing quirks can end in looping and the ultimate destruction of a datagram; and disabled links may leave no usable path to the destination.

If reliability is important, IP must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage. An example of a situation similar to pairing IP with a protocol that contains reliability functions is a self-addressed, stamped postcard included in a letter mailed through the post office. When the letter is delivered, the receiver mails the postcard back to the sender to indicate success. If the sender never receives the postcard, he or she assumes the letter was lost and sends out another copy.

IP transports data in packets called datagrams (described below), each of which is transported separately. Datagrams may travel along different routes and may arrive out of sequence or duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive. Because it is a connectionless service, IP does not create virtual circuits for delivery. There is no call setup to alert the receiver to an incoming transmission.

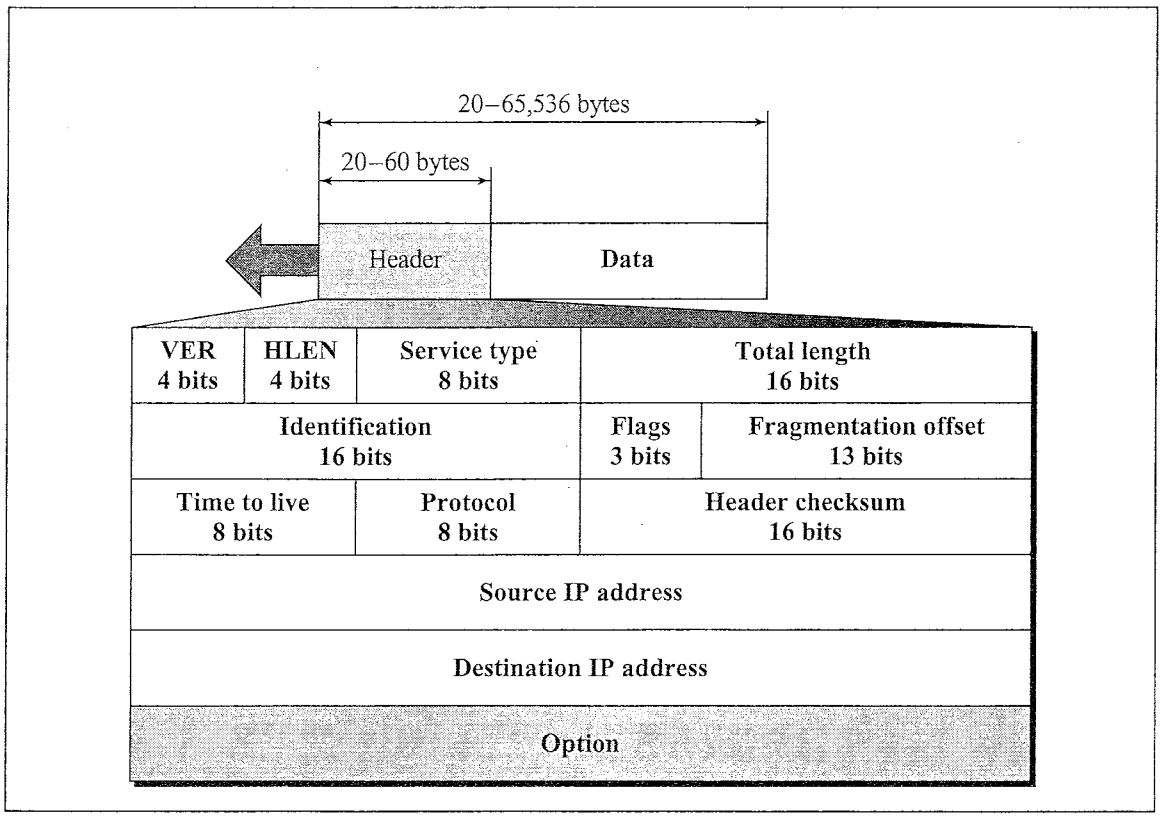
The limited functionality of IP should not be considered a weakness, however. IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency.

Datagram

Packets in the IP layer are called datagrams. Figure 18.3 shows the **IP datagram** format. A datagram is a variable-length packet consisting of two parts: header and data. The header can be from 20 to 60 bytes and contains information essential to routing and delivery. It is customary in TCP/IP to show the header in 4-byte sections. A brief description of each field is in order.

- **Version.** The first field defines the version number of the IP. The current version is 4 (**IPv4**), with a binary value of 0100.
- **Header length (HLEN).** The HLEN field defines the length of the header in multiples of 4 bytes. The 4 bits can represent a number between 0 and 15, which, when multiplied by 4, gives a maximum of 60 bytes.
- **Service type.** The service type field defines how the datagram should be handled. It includes bits that define the priority of the datagram. It also contains bits that specify the type of service the sender desires such as the level of throughput, reliability, and delay.
- **Total length.** The total length field defines the total length of the IP datagram. It is a 2-byte field (16 bits).
- **Identification.** The identification field is used in **fragmentation.** A datagram, when passing through different networks, may be divided into fragments to match the network frame size. When this happens, each fragment is identified with a sequence number in this field.
- **Flags.** The bits in the flags field deal with fragmentation (the datagram can or cannot be fragmented; can be the first, middle, or last fragment; etc.).

Figure 18.3 *IP datagram*



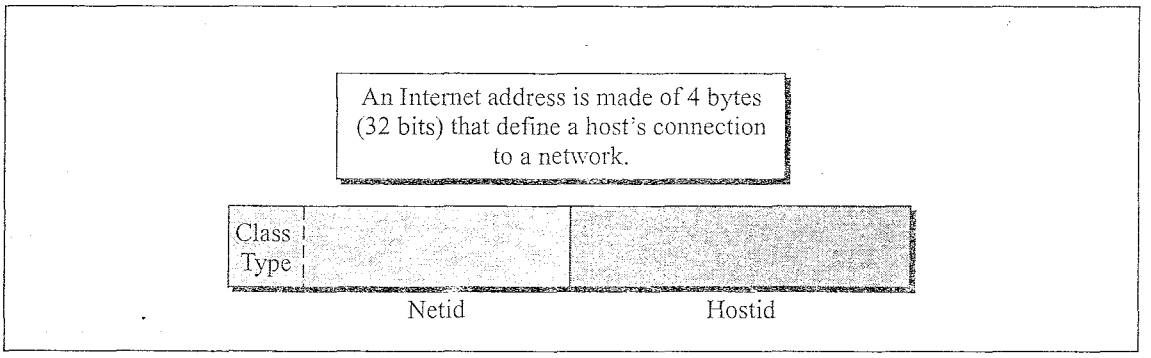
- **Fragmentation offset.** The fragmentation offset is a pointer that shows the offset of the data in the original datagram (if it is fragmented).
- **Time to live.** The time-to-live field defines the number of hops a datagram can travel before it is discarded. The source host, when it creates the datagram, sets this field to an initial value. Then, as the datagram travels through the Internet, router by router, each router decrements this value by 1. If this value becomes 0 before the datagram reaches its final destination, the datagram is discarded. This prevents a datagram from going back and forth forever between routers.
- **Protocol.** The protocol field defines which upper-layer protocol data are encapsulated in the datagram (TCP, UDP, ICMP, etc.).
- **Header checksum.** This is a 16-bit field used to check the integrity of the header, not the rest of the packet.
- **Source address.** The source address field is a 4-byte (32-bit) Internet address. It identifies the original source of the datagram.
- **Destination address.** The destination address field is a 4-byte (32-bit) Internet address. It identifies the final destination of the datagram.
- **Options.** The options field gives more functionality to the IP datagram. It can carry fields that control routing, timing, management, and alignment.

18.3 ADDRESSING

In addition to the physical addresses (contained on NICs) that identify individual devices, the Internet requires an additional addressing convention: an address that identifies the connection of a host to its network.

Each **Internet address** consists of 4 bytes (32 bits), defining three fields: class type, netid, and hostid. These parts are of varying lengths, depending on the class of the address (see Figure 18.4).

Figure 18.4 *Internet address*



Classes

There are currently five different **classes of address.** The different classes are designed to cover the needs of different types of organizations. For example, class A addresses are numerically the lowest. They use only 1 byte to identify class type and netid, and leave 3 bytes available for hostid numbers. This division means that class A networks can accommodate far more hosts than can class B or class C networks, which provide 2- and 1-byte hostid fields, respectively. Currently both class A and class B are full. Addresses are available in class C only.

Class D is reserved for **multicast addresses. Multicasting** allows copies of a datagram to be passed to a select group of hosts rather than to an individual host. It is similar to **broadcasting,** but, where broadcasting requires that a packet be passed to all possible destinations, multicasting allows transmission to a selected subset. Class E addresses are reserved for future use. Figure 18.5 shows the structure of each **IP address class.**

Example 1

What is the class of each of the following addresses?

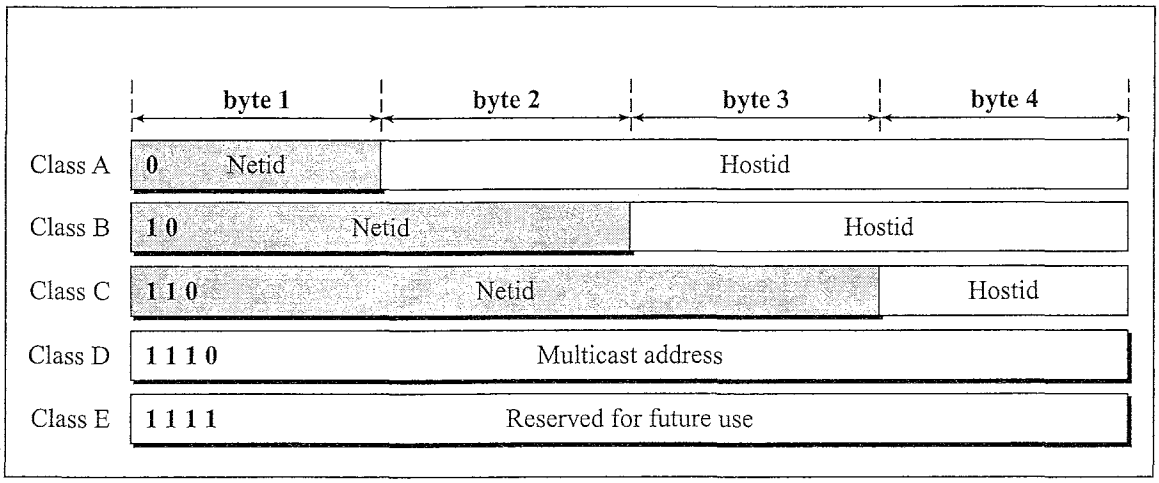
a. 10011101 10001111 11111100 11001111
b. 11011101 10001111 11111100 11001111
c. 01111011 10001111 11111100 11001111
d. 11101011 10001111 11111100 11001111
e. 11110101 10001111 11111100 11001111

Solution

The first bits define the class:

a. Class B
b. Class C
c. Class A
d. Class D
e. Class E

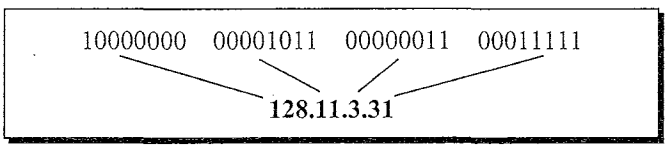
Figure 18.5 *Internet classes*



Dotted-Decimal Notation

To make the 32-bit form shorter and easier to read, Internet addresses are usually written in decimal form with decimal points separating the bytes—**dotted-decimal notation.** Figure 18.6 shows the bit pattern and decimal format of a possible address.

Figure 18.6 *IP addresses in decimal notation*



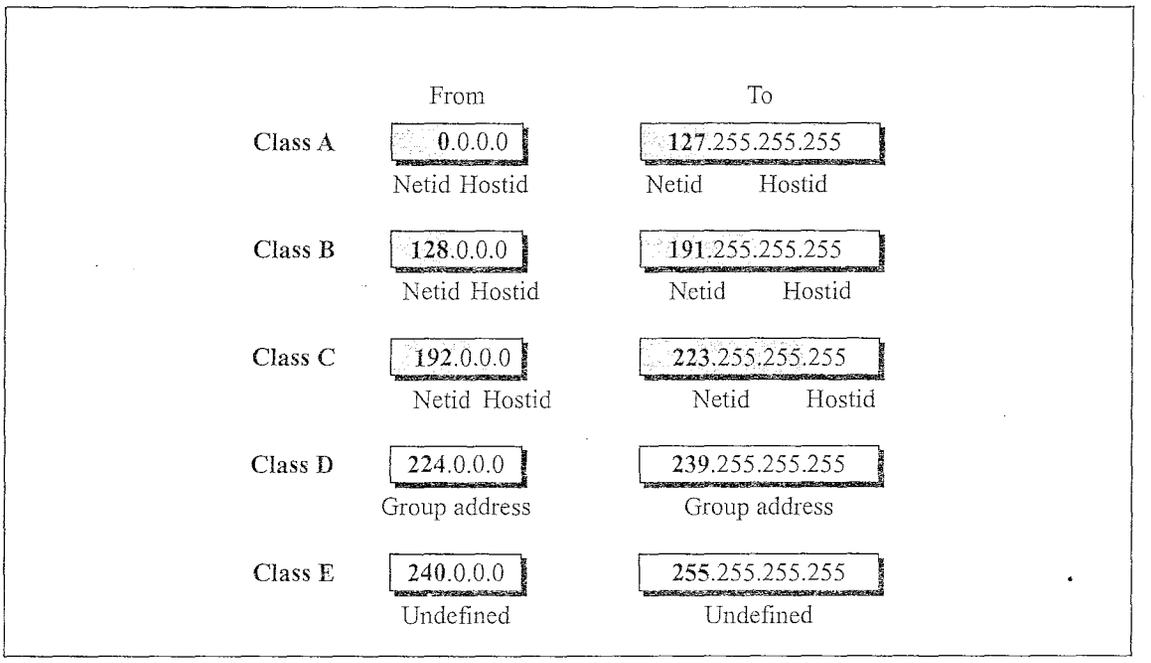
Looking at the first byte of an address in decimal form allows us to determine at a glance to which class a particular address belongs (see Figure 18.7).

Example 2

Write each of following in dotted-decimal notation:

a. 10011101 10001111 11111100 11001111
b. 11011101 10001111 11111101 00001111
c. 01011101 00011111 00000001 11110101

Figure 18.7 *Class ranges of Internet addresses*



d. 11111101 10001010 00001111 00111111
e. 11111110 10000001 01111110 00000001

Solution

Each byte is converted to a decimal number between 0 and 255.

a. 157.143.252.207
b. 221.143.253.15
c. 93.31.1.245
d. 253.138.15.63
e. 254.129.126.1

Example 3

Find the class of each address:

a. 4.23.145.90
b. 227.34.78.7
c. 246.7.3.8
d. 129.6.8.4
e. 198.76.9.23

Solution

The first number defines the class.

a. Class A
b. Class D
c. Class E
d. Class B
e. Class C

Example 4

Find the netid and the hostid for each address:

a. 4.23.145.90
b. 227.34.78.7
c. 246.7.3.8
d. 129.6.8.4
e. 198.76.9.23

Solution

First find the class and then find the netid and hostid.

a. Class A, netid: 4 hostid: 23.145.90
b. Class D, no hostid or netid
c. Class E, no hostid or netid
d. Class B, netid: 129.6 hostid: 8.4
e. Class C, netid: 198.76.9 hostid: 23

Example 5

Find the network address for each address:

a. 4.23.145.90
b. 227.34.78.7
c. 246.7.3.8
d. 129.6.8.4
e. 198.76.9.23

Solution

First find the class and then find the network address.

a. Class A, network address: 4.0.0.0
b. Class D, no network address
c. Class E, no network address
d. Class B, network address: 129.6.0.0
e. Class C, network address: 198.76.9.0

Nodes with More Than One Address

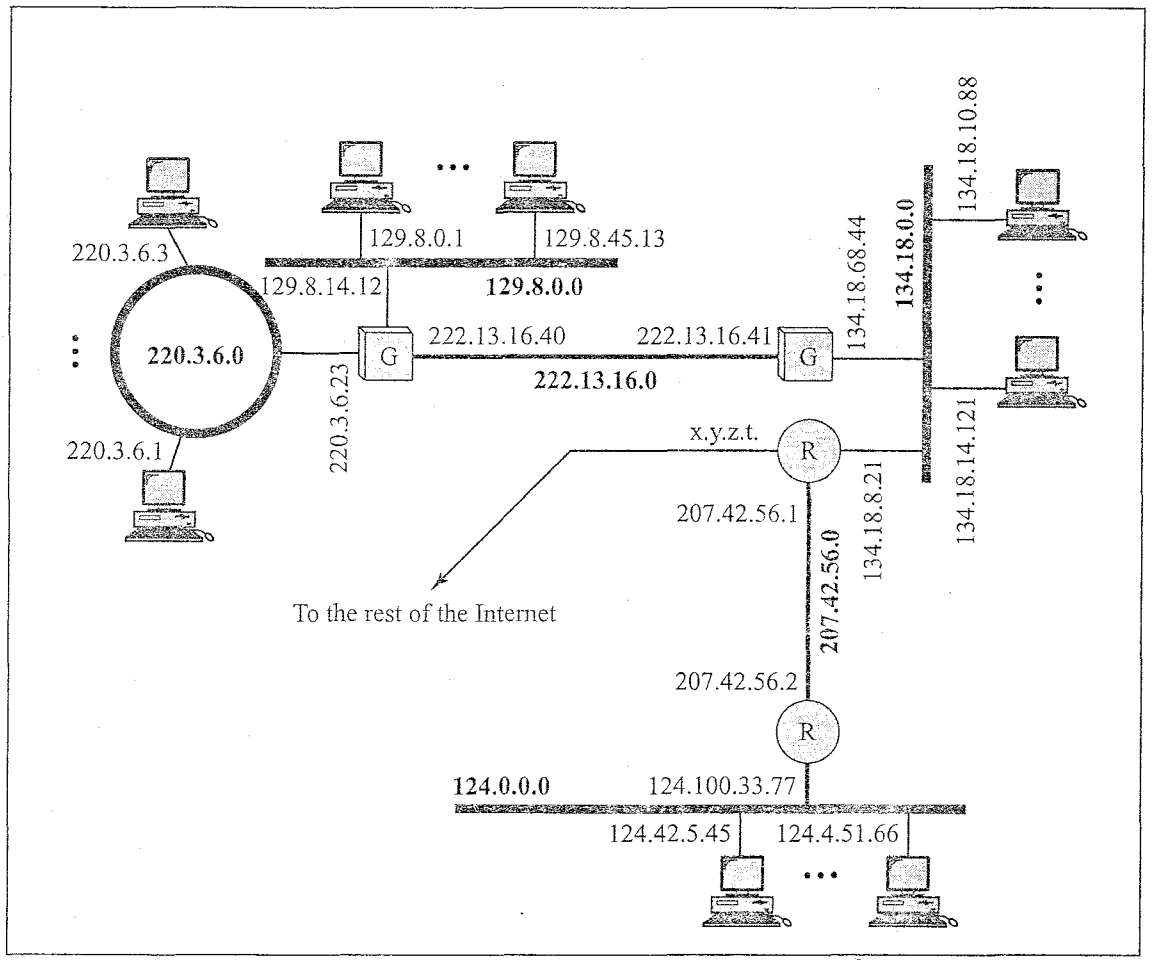
As we have said, an internet address defines the node's connection to its network. It follows, therefore, that any device connected to more than one network (e.g., any router) must have more than one internet address. In fact, a device has a different address for each network connected to it.

A Sample Internet

An internet address specifies both the network to which a host belongs (**netid**) and the host itself (**hostid**). Figure 18.8 shows a portion of the Internet made up of LANs (three Ethernets and a Token Ring). Routers are indicated by circles containing Rs. Gateways

are indicated by boxes containing Gs. Each has a separate address for each of its connected networks. The figure also shows the network addresses **in bold.** A network address is the netid with the hostid part set to 0s. The network addresses in the figure are 129.8.0.0 (class B), 124.0.0.0 (class A), 134.18.0.0 (class B), and 220.3.6.0 (class C).

Figure 18.8 *Network and host addresses*

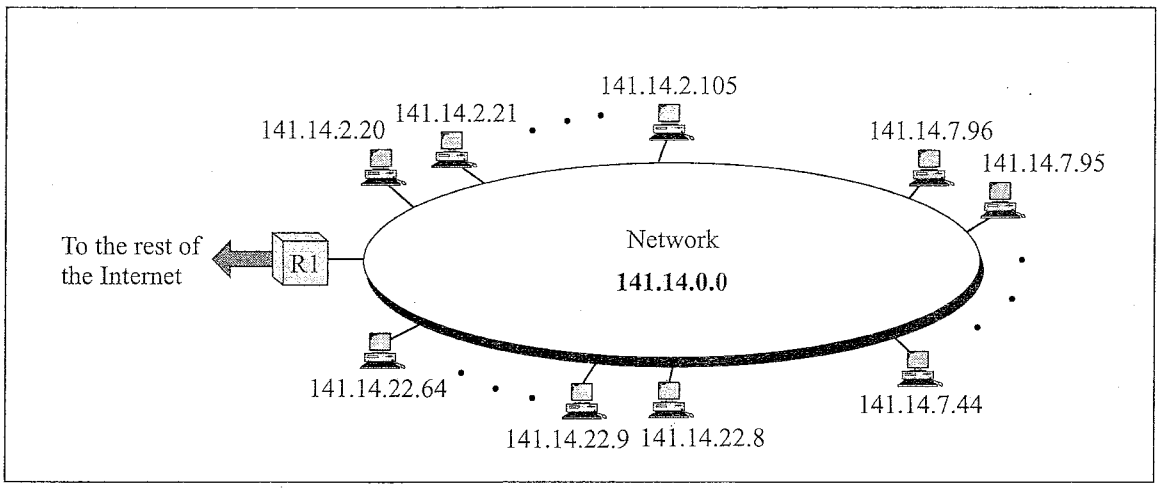


18.4 SUBNETTING

As we previously discussed, an **IP address** is 32 bits long. One portion of the address indicates a network (netid), and the other portion indicates the host (or router) on the network (hostid). This means that there is a sense of hierarchy in IP addressing. To reach a host on the Internet, we must first reach the network using the first portion of the address (netid). Then we must reach the host itself using the second portion (hostid). In other words, classes A, B, and C in IP addressing are designed with two levels of hierarchy.

However, in many cases, these two levels of hierarchy are not enough. For example, imagine an organization with a class B address. The organization has two-level hierarchical addressing, but it cannot have more than one physical network (see Figure 18.9).

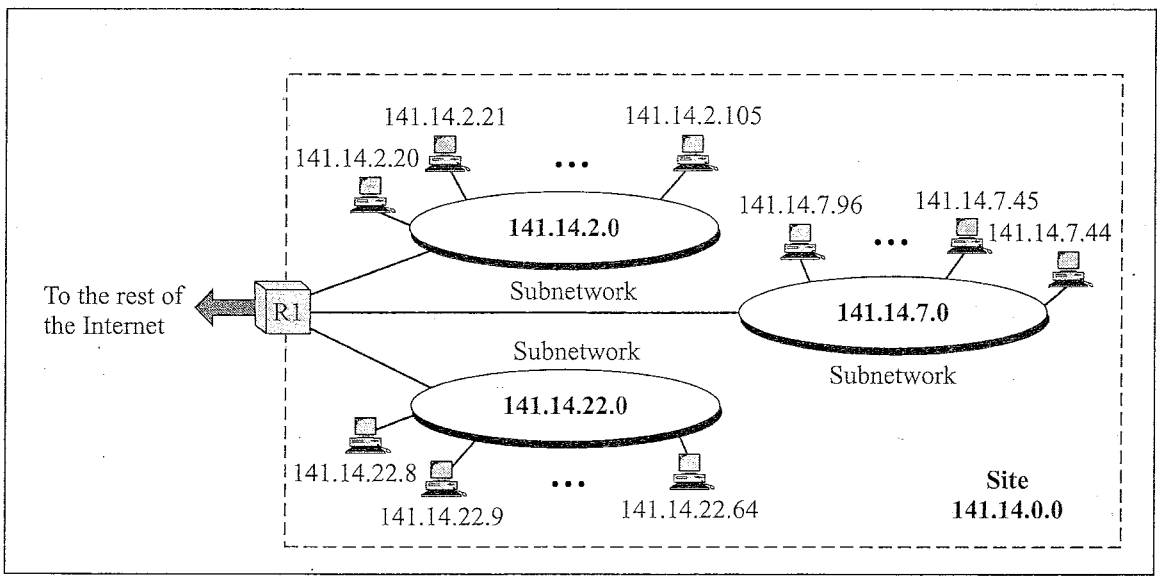
Figure 18.9 *A network with two levels of hierarchy (not subnetted)*



With this scheme, the organization is limited to two levels of hierarchy. The hosts cannot be organized into groups, and all of the hosts are at the same level. The organization has one network with many hosts.

One solution to this problem is **subnetting,** the further division of a network into smaller networks called **subnetworks.** For example, Figure 18.10 shows the network in Figure 18.9 divided into three subnetworks.

Figure 18.10 *A network with three levels of hierarchy (subnetted)*



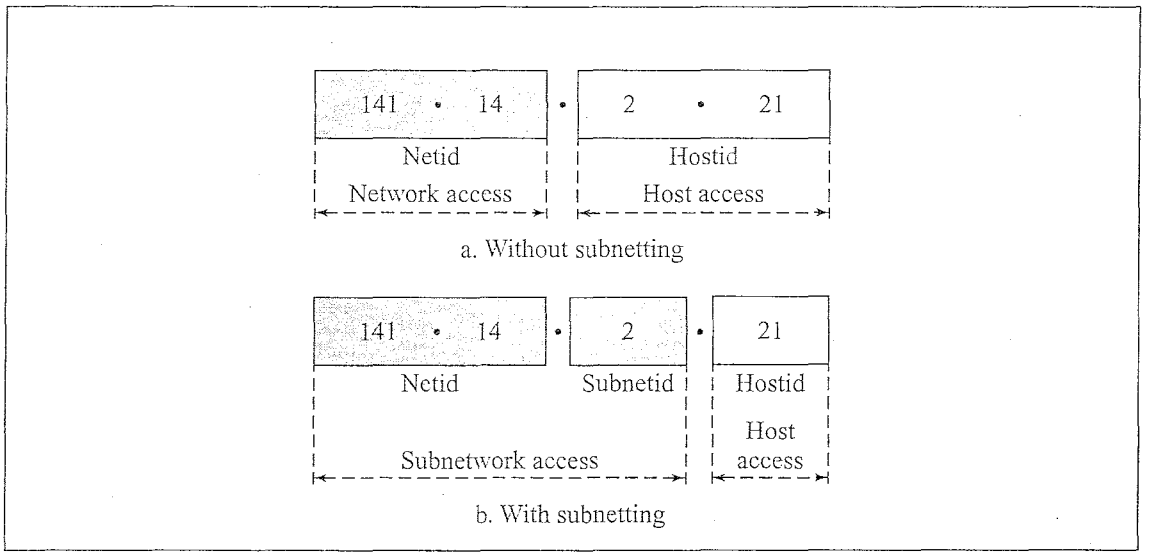
In this example, the rest of the Internet is not aware that the network is divided into three physical subnetworks: the three subnetworks still appear as a single network to the rest of the Internet. A packet destined for host 141.14.2.21 still reaches router R1. The destination address of the IP datagram is still a class B address where 141.14 defines the netid and 2.21 defines the hostid.

However, when the packet arrives at router R1, the interpretation of the IP address changes. Router R1 knows that the network 141.14 is physically divided into three subnetworks. It knows that the last two octets define two things: subnetid and hostid. Therefore, 2.21 must be interpreted as subnetid 2 and hostid 21. The router R1 uses the first two octets (141.14) as the netid, the third octet (2) as the subnetid, and the fourth octet (21) as the hostid.

Three Levels of Hierarchy

Adding subnetworks creates an intermediate level of hierarchy in the IP addressing system. Now we have three levels: netid, subnetid, and hostid. The netid is the first level; it defines the site. The second level is the **subnetid;** it defines the physical subnetwork. The hostid is the third level; it defines the connection of the host to the subnetwork (see Figure 18.11).

Figure 18.11 *Addresses in a network with and without subnetting*



The routing of an IP datagram now involves three steps: delivery to the site, delivery to the subnetwork, and delivery to the host.

Masking

Masking is a process that extracts the address of the physical network from an IP address. Masking can be done whether we have subnetting or not. If we have not subnetted the network, masking extracts the network address from an IP address. If we have subnetted, masking extracts the **subnetwork address** from an IP address (see Figure 18.12).

Masks Without Subnetting

To be compatible, routers use a mask even if there is no subnetting. The masks for networks that are not subnetted are defined in Table 18.1.

Figure 18.12 *Masking*

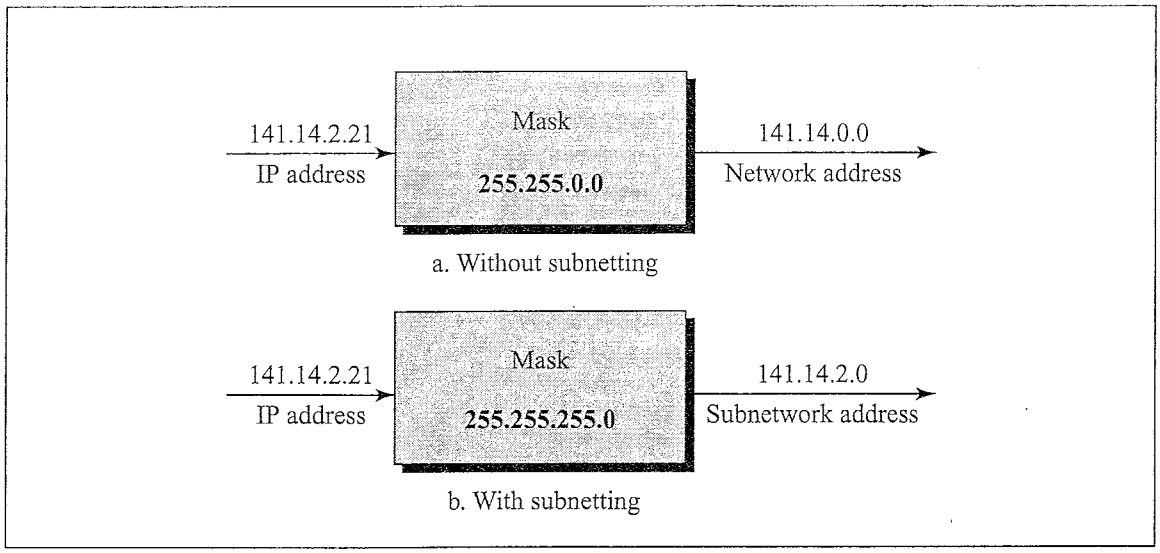


Table 18.1 *Mask for unsubnetted networks*

Class	*Mask*	*Address (Example)*	*Network Address (Example)*
A	255.0.0.0	15.32.56.7	15.0.0.0
B	255.255.0.0	135.67.13.9	135.67.0.0
C	255.255.255.0	201.34.12.72	201.34.12.0
D	N/A	N/A	N/A
E	N/A	N/A	N/A

Masks with Subnetting

When there is subnetting, the mask can vary. Table 18.2 shows some examples of masks used for subnetting.

Table 18.2 *Masks for subnetted networks*

Class	*Mask*	*Address (Example)*	*Network Address (Example)*
A	255.255.0.0	15.32.56.7	15.32.0.0
B	255.255.255.0	135.67.13.9	135.67.13.0
C	255.255.255.192	201.34.12.72	201.34.12.64
D	N/A	N/A	N/A
E	N/A	N/A	N/A

Finding the Subnetwork Address

To find the subnetwork address, apply the mask to the IP address.

Boundary-Level Masking

If the masking is at the boundary level (the mask numbers are either 255 or 0), finding the subnetwork address is very easy. Follow these two rules:

1. The bytes in the IP address that correspond to 255 in the mask will be repeated in the subnetwork address.
2. The bytes in the IP address that correspond to 0 in the mask will change to 0 in the subnetwork address.

Example 6

The following shows how to get the subnetwork address from an IP address:

IP address	45	.	23	.	21	.	8
Mask	255	.	255	.	0	.	0
Subnetwork address	45	.	23	.	0	.	0

Example 7

The following shows how to get the subnetwork address from an IP address:

IP address	173	.	23	.	21	.	8
Mask	255	.	255	.	255	.	0
Subnetwork address	173	.	23	.	21	.	0

Nonboundary-Level Masking

If the masking is not at the boundary level (the mask numbers are not just 255 or 0), finding the subnetwork address involves using the bit-wise AND operator. Follow these three rules:

1. The bytes in the IP address that correspond to 255 in the mask will be repeated in the subnetwork address.
2. The bytes in the IP address that correspond to 0 in the mask will change to 0 in the subnetwork address.
3. For other bytes, use the bit-wise AND operator.

Example 8

The following shows how to get the subnetwork address from an IP address:

IP address	45	.	123	.	21	.	8
Mask	255	.	192	.	0	.	0
Subnetwork address	45	.	64	.	0	.	0

As you can see, three bytes are easy to determine. However, the second byte needs the *bit-wise AND* operation. The bit-wise AND operation is very simple. If two bits are both 1s, the result is 1; otherwise, the result is 0.

123	0 1 1 1 1 0 1 1
192	1 1 0 0 0 0 0 0
64	0 1 0 0 0 0 0 0

Example 9

The following shows how to get the subnetwork address from an IP address:

IP address	213	23	47	37
Mask	255	255	255	240
Subnetwork address	213	23	47	32

As you can see, three bytes are easy to determine. However, the fourth byte needs the bit-wise AND operation.

37	0 0 1 0 0 1 0 1
240	1 1 1 1 0 0 0 0
32	0 0 1 0 0 0 0 0

18.5 OTHER PROTOCOLS IN THE NETWORK LAYER

TCP/IP supports four other protocols in the network layer: ARP, RARP, ICMP, and IGMP.

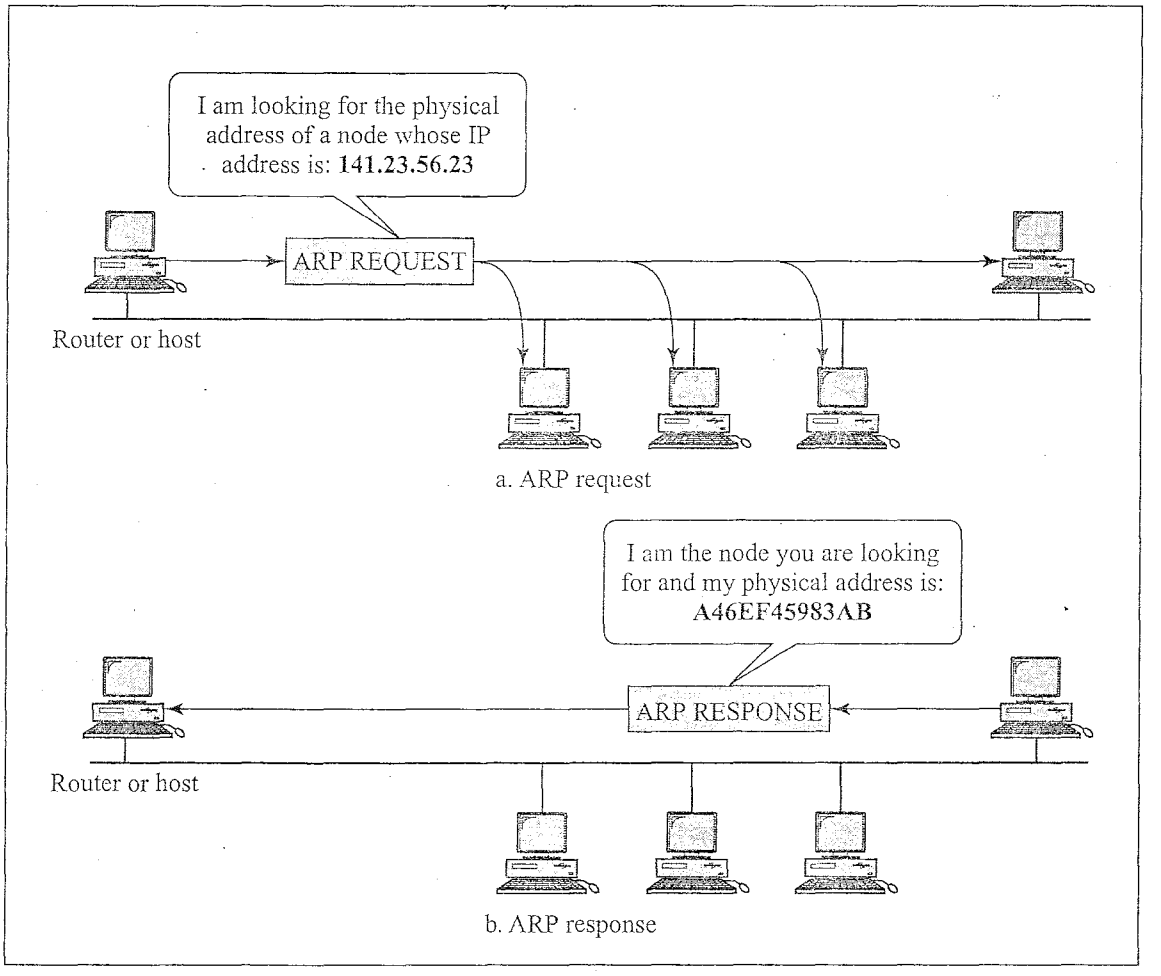
Address Resolution Protocol (ARP)

The **address resolution protocol (ARP)** associates an IP address with the physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address usually imprinted on the network interface card (NIC).

Physical addresses have local jurisdiction. The IP addresses, on the other hand, have universal jurisdiction. ARP is used to find the physical address of the node when its Internet address is known.

Anytime a host, or a router, needs to find the physical address of another host on its network, it formats an ARP query packet that includes the IP address and broadcasts it over the network (see Figure 18.13). Every host on the network receives and processes the ARP packet, but only the intended recipient recognizes its internet address and sends back its physical address. The host holding the datagram adds the address of the target host both to its cache memory and to the datagram header, then sends the datagram on its way.

Figure 18.13 *ARP*



Reverse Address Resolution Protocol (RARP)

The **reverse address resolution protocol (RARP)** allows a host to discover its internet address when it knows only its physical address. The question here is, Why do we need RARP? A host is supposed to have its internet address stored on its hard disk!

Answer: True. But what if the host is a diskless computer? Or what if the computer is being connected to the network for the first time (when it is being booted)? Or what if you get a new computer but decide to keep the old NIC?

RARP works much like ARP. The host wishing to retrieve its internet address broadcasts a RARP query packet that contains its physical address to every host on its

physical network. A server on the network recognizes the RARP packet and returns the host's internet address.

Internet Control Message Protocol (ICMP)

The **Internet Control Message Protocol (ICMP)** is a mechanism used by hosts and routers to send notification of datagram problems back to the sender.

As we saw above, IP is essentially an unreliable and connectionless protocol. ICMP, however, allows IP to inform a sender if a datagram is undeliverable. A datagram travels from router to router until it reaches one that can deliver it to its final destination. If a router is unable to route or deliver the datagram because of unusual conditions (disabled links, or the device is on fire) or because of network congestion, ICMP allows it to inform the original source.

ICMP uses echo test/reply to test whether a destination is reachable and responding. It also handles both control and error messages, but its sole function is to report problems, not correct them. Responsibility for correction lies with the sender.

Note that a datagram carries only the addresses of the original sender and the final destination. It does not know the addresses of the previous router(s) that passed it along. For this reason, ICMP can send messages only to the source, not to an intermediate router.

Internet Group Message Protocol (IGMP)

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 trip cancellation. Some other applications include distance learning and video-on-demand.

IP addressing supports multicasting. All 32-bit IP addresses that start with 1110 (class D) are multicast addresses. With 28 bits remaining for the group address, more than 250 million addresses are available for assignment. Some of these addresses are permanently assigned.

The **Internet Group Message Protocol (IGMP)** has been designed to help a multicast router identify the hosts in a LAN that are members of a multicast group. It is a companion to the IP protocol.

18.6 TRANSPORT LAYER

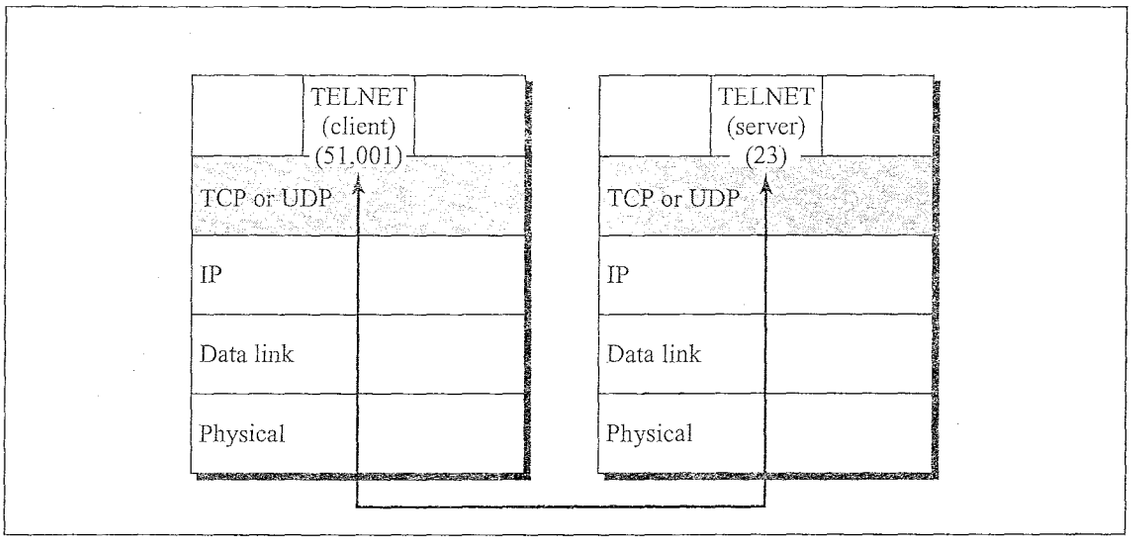
The transport layer is represented in TCP/IP by two protocols: TCP and UDP. Of these, UDP is the simpler; it provides nonsequenced transport functionality when reliability and security are less important than size and speed. Most applications, however, require reliable end-to-end delivery and so make use of TCP.

The IP delivers a datagram from a source host to a destination host, making it a host-to-host protocol. Today's operating systems, however, support multiuser and multiprocessing environments. An executing program is called a process. A host receiving a datagram may be running several different concurrent processes, any one of which is a possible destination for the transmission. In fact, although we have been talking about hosts sending messages to other hosts over a network, it is actually a source process that is sending a message to a destination process.

The transport protocols of the TCP/IP suite define a set of conceptual connections to individual processes called protocol ports or, more simply, ports. A protocol port is a destination point (usually a buffer) for storing data for use by a particular process. The interface between processes and their corresponding ports is provided by the operating system of the host.

The IP is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another. TCP/IP's transport level protocols are port-to-port protocols that work on top of the IP protocols to deliver the packet from the originating port to the IP services at the start of a transmission, and from the IP services to the destination port at the end (see Figure 18.14).

Figure 18.14 *Port addresses*



Each port is defined by a positive integer address carried in the header of a transport layer packet. An IP datagram uses the host's 32-bit internet address. A frame at the transport level uses the process **port address** of 16 bits, enough to allow the support of up to 65,536 (0 to 65,535) ports.

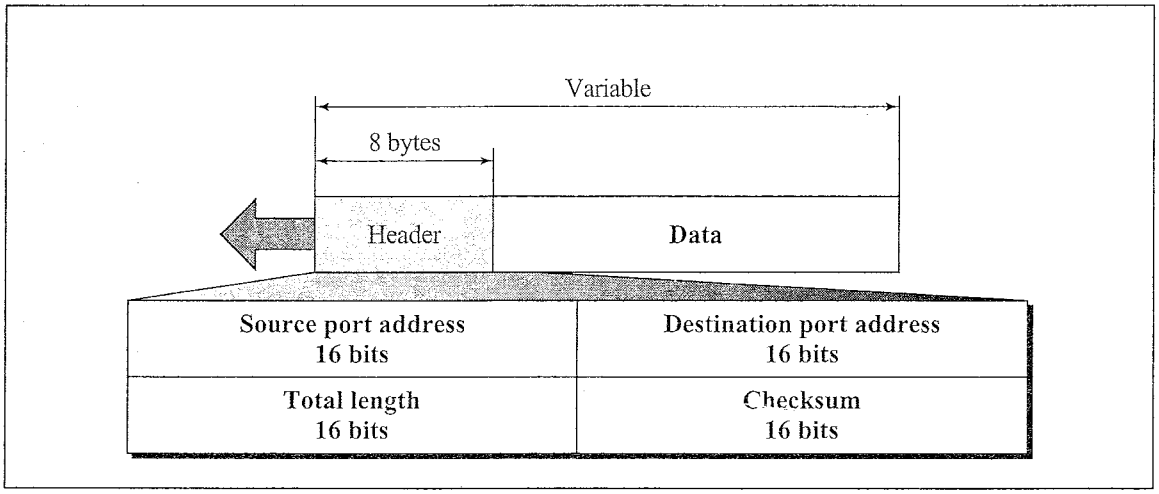
User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is the simpler of the two standard TCP/IP transport protocols. It is an end-to-end transport level protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer. The

packet produced by the UDP is called a user datagram (see Figure 18.15). A brief description of its fields is in order.

- **Source port address.** The source port address is the address of the application program that has created the message.
- **Destination port address.** The destination port address is the address of the application program that will receive the message.
- **Total length.** The total length field defines the total length of the user datagram in bytes.
- **Checksum.** The checksum is a 16-bit field used in error detection.

Figure 18.15 *UDP datagram format*



UDP provides only the basic functions needed for end-to-end delivery of a transmission. It does not provide any sequencing or reordering functions and cannot specify the damaged packet when reporting an error (for which it must be paired with ICMP). UDP can discover that an error has occurred; ICMP can then inform the sender that a user datagram has been damaged and discarded. Neither, however, has the ability to specify which packet has been lost. UDP contains only a checksum; it does not contain an ID or sequencing number for a particular data segment.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) provides full transport layer services to applications. TCP is a reliable stream transport port-to-port protocol. The term *stream,* in this context, means connection-oriented: a connection must be established between both ends of a transmission before either may transmit data. By creating this connection, TCP generates a virtual circuit between sender and receiver that is active for the duration of a transmission. (Connections for the duration of an entire exchange are different, and are handled by session functions in individual applications.) TCP begins each transmission by alerting the receiver that datagrams are on their way (connection establishment) and ends each transmission with a connection termination. In this way, the receiver knows to expect the entire transmission rather than a single packet.

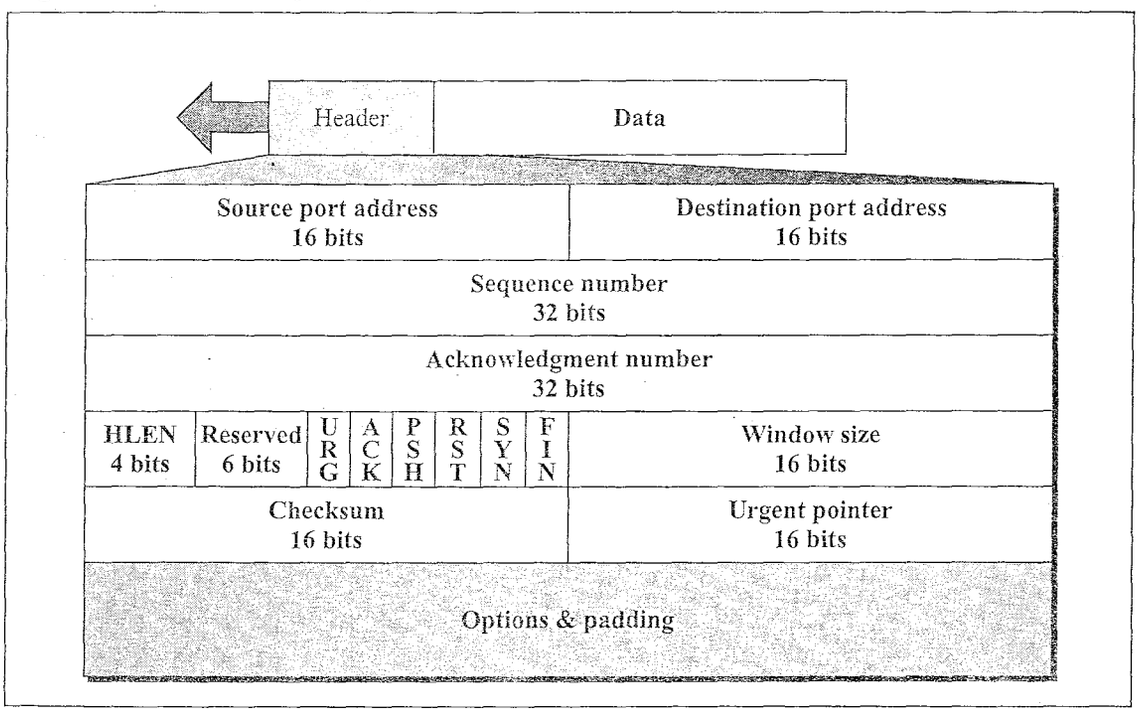
IP and UDP treat multiple datagrams belonging to a single transmission as entirely separate units, unrelated to each other. The arrival of each datagram at the destination is therefore a separate event, unexpected by the receiver. TCP, on the other hand, as a connection-oriented service, is responsible for the reliable delivery of the entire stream of bits contained in the message originally generated by the sending application. Reliability is ensured by provision for error detection and retransmission of damaged frames; all segments must be received and acknowledged before the transmission is considered complete and the virtual circuit is discarded.

At the sending end of each transmission, TCP divides long transmissions into smaller data units and packages each into a unit called a segment. Each segment includes a sequencing number for reordering after receipt, together with an acknowledgment ID number and a window-size field for sliding window ARQ. Segments are carried across network links inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.

The TCP Segment

The scope of the services provided by TCP requires that the segment header be extensive (see Figure 18.16). A comparison of the TCP segment format with that of a UDP user datagram shows the differences between the two protocols. TCP provides a comprehensive range of reliability functions but sacrifices speed (connections must be established, acknowledgments waited for, etc.). Because of its smaller frame size, UDP is much faster than TCP, but at the expense of reliability. A brief description of each field is in order.

Figure 18.16 *TCP segment format*



- **Source port address.** The source port address defines the application program in the source computer.

- **Destination port address.** The destination port address defines the application program in the destination computer.
- **Sequence number.** A stream of data from the application program may be divided into two or more TCP segments. The sequence number field shows the position of the data in the original data stream.
- **Acknowledgment number.** The 32-bit acknowledgment number is used to acknowledge the receipt of data from the other communicating device. This number is valid only if the ACK bit in the control field (explained later) is set. In this case, it defines the byte sequence number that is next expected.
- **Header length (HLEN).** The four-bit HLEN field indicates the number of 32-bit (4-byte) words in the TCP header. The 4 bits can define a number up to 15. This is multiplied by 4 to give the total number of bytes in the header. Therefore, the size of the header can be a maximum of 60 bytes (4×15). Since the minimum required size of the header is 20 bytes, 40 bytes are thus available for the options section.
- **Reserved.** A 6-bit field is reserved for future use.
- **Control.** Each bit of the 6-bit control field functions individually and independently. A bit can either define the use of a segment or serve as a validity check for other fields. The urgent bit, when set, validates the urgent pointer field. Both this bit and the pointer indicate that the data in the segment are urgent. The ACK bit, when set, validates the acknowledgment number field. Both are used together and have different functions, depending on the segment type. The PSH bit is used to inform the sender that a higher throughput is needed. If possible, data must be pushed through paths with higher throughput. The reset bit is used to reset the connection when there is confusion in the sequence numbers. The SYN bit is used for sequence number synchronization in three types of segments: connection request, connection confirmation (with the ACK bit set), and confirmation acknowledgment (with the ACK bit set). The FIN bit is used in connection termination in three types of segments: termination request, termination confirmation (with the ACK bit set), and acknowledgment of termination confirmation (with the ACK bit set).
- **Window size.** The window is a 16-bit field that defines the size of the sliding window.
- **Checksum.** The checksum is a 16-bit field used in error detection.
- **Urgent pointer.** This is the last required field in the header. Its value is valid only if the URG bit in the control field is set. In this case, the sender is informing the receiver that there are **urgent data** in the data portion of the segment. This pointer defines the end of urgent data and the start of normal data.
- **Options and padding.** The remainder of the TCP header defines the optional fields. They are used to convey additional information to the receiver or for alignment purposes.

18.7 NEXT GENERATION: IPv6 AND ICMPv6

The network layer protocol in the TCP/IP 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, including the following:

- IPv4 has a two-level address structure (netid and hostid) categorized into five classes (A, B, C, D, and E). The use of address space is inefficient. For instance, when an organization is granted a class A address, 16 million addresses from the address space are assigned for the organization's exclusive use. If an organization is granted a class C address, on the other hand, only 256 addresses are assigned to this organization, which may not be a sufficient number. Also, millions of addresses are wasted in classes D and E. This method of addressing has depleted the address space of IPv4, and soon there will not be any addresses left to assign to any new system that wants to be connected to the Internet. Although the subnetting and supernetting strategies have alleviated some of the addressing problems, subnetting and supernetting make routing more complicated.
- 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.
- The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.

To overcome these deficiencies, **IPv6 (Internetworking Protocol, version 6)**, also known as **IPng (Internetworking Protocol, next generation)** was proposed and is now a standard. In IPv6, the protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP addresses were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMP protocol. Routing protocols, such as RIP and OSPF, were also slightly modified to accommodate these changes. Communication experts predict that IPv6 and its related protocols will soon replace the current IP version. In this chapter we talk first about IPv6. Then we discuss ICMPv6.

IPv6

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

- **Larger address space.** An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a four-fold increase in the address space.
- **Better header format.** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options.** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension.** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation.** In IPv6, the type-of-service field has been removed, but a mechanism (called **flow label**) has been added to enable the source

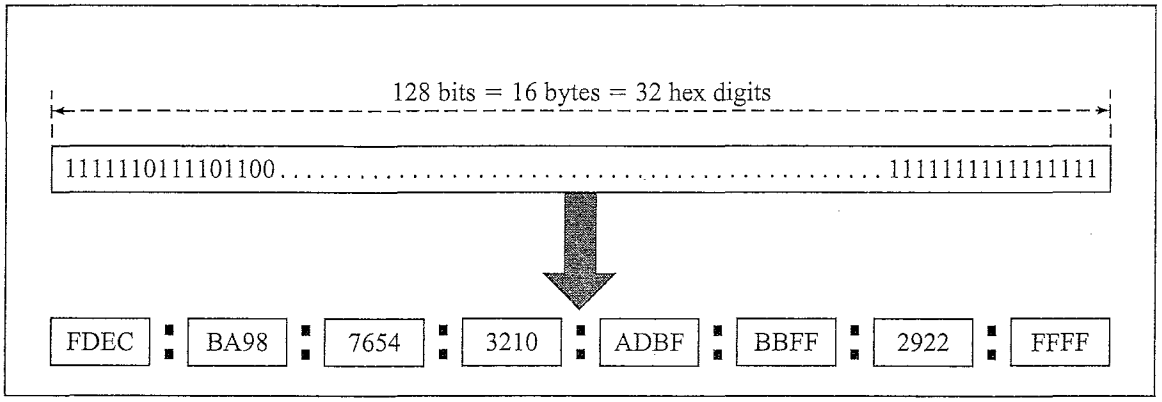
to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.

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

IPv6 Addresses

An IPv6 address consists of 16 bytes (octets), making it 128 bits long (see Figure 18.17).

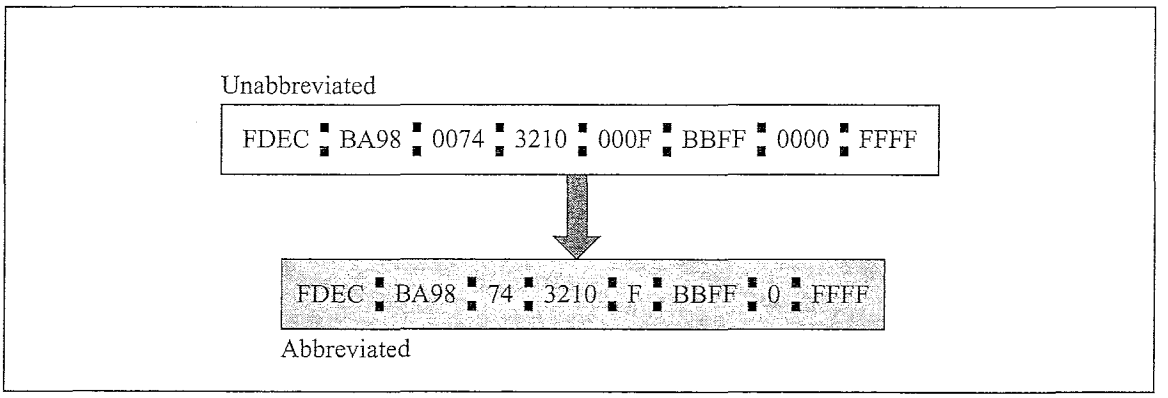
Figure 18.17 *IPv6 address*



Hexadecimal Colon Notation To make addresses more readable, IPv6 addresses are written in **hexadecimal colon notation.** In this notation, 128 bits are divided into eight sections, each two bytes in length. Two bytes in hexadecimal notation require four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.

- **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. For an example, see Figure 18.18.

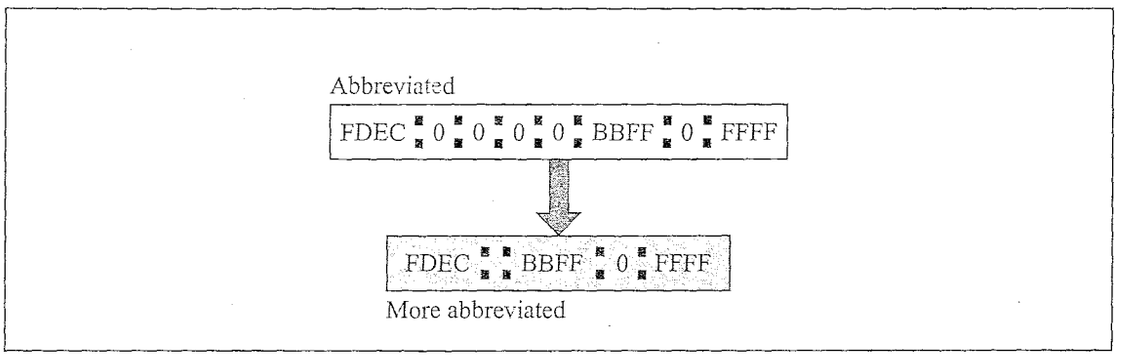
Figure 18.18 *Abbreviated address*



Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0. Note that 3210 cannot be abbreviated. Further abbreviations are possible if there

are consecutive sections consisting of zeros only. We can remove the zeros altogether and replace them with a double semicolon. Figure 18.19 shows the concept.

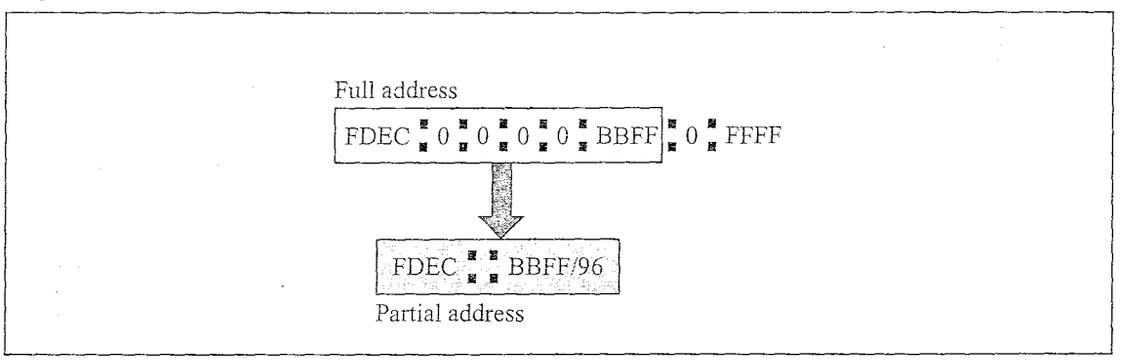
Figure 18.19 *Abbreviated address with consecutive zeros*



Note that this type of abbreviation is allowed only once per address. If there are two runs of zero sections, only one of them can be abbreviated. Reexpansion of the abbreviated address is very simple: align the unabbreviated portions and insert zeros to get the original expanded address.

Sometimes we need to refer to only part of the address, not all of it. To do so, place a slash after the digits you wish to keep, and follow it with the number of digits kept. For example, Figure 18.20 shows how the first six sections can be written in a shortened form.

Figure 18.20 *Partial address*



Categories of Addresses IPv6 defines three types of addresses: unicast, anycast, and multicast.

- **Unicast addresses.** A **unicast address** defines a single host. The packet sent to a unicast address should be delivered to that specific host.
- **Anycast addresses.** An **anycast address** defines a group of hosts whose addresses have the same prefix. For example, all computers connected to the same physical network share the same prefix address. A packet sent to an anycast address should be delivered to exactly one of the members of the group—the closest or most easily accessible.
- **Multicast addresses.** A multicast address defines a group of computers that may or may not share the same prefix and may or may not be connected to the same

physical network. A packet sent to a multicast address should be delivered to each member of the set.

Address Space Assignment The **address space** has many different purposes. The designers of the IP addresses divided the address space into two parts, with the first part called the **type prefix.** This variable-length prefix defines the purpose of the address. The codes are 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. Figure 18.21 shows the IPv6 address format.

Figure 18.21 *Address structure*

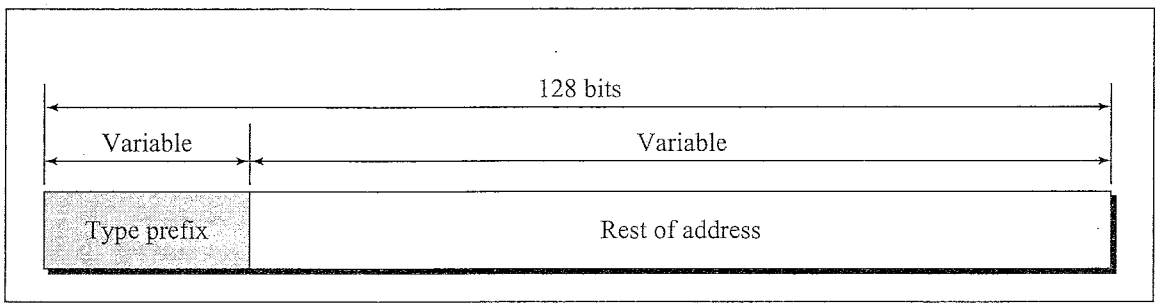


Table 18.3 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space.

Table 18.3 *Type prefixes for IPv6 addresses*

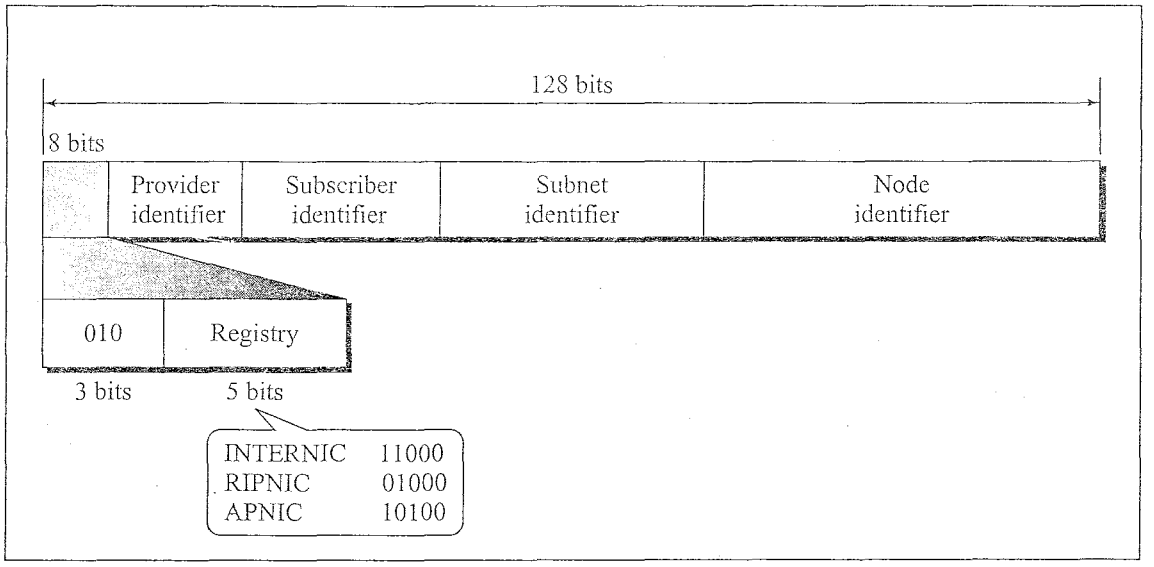
Type Prefix	*Type*	*Fraction*
0000 0000	Reserved	1/256
0000 0001	Reserved	1/256
0000 001	NSAP (Network Service Access Point)	1/128
0000 010	IPX (Novell)	1/128
0000 011	Reserved	1/128
0000 100	Reserved	1/128
0000 101	Reserved	1/128
0000 110	Reserved	1/128
0000 111	Reserved	1/128
0001	Reserved	1/16
001	Reserved	1/8
010	**Provider-based unicast addresses**	**1/8**
011	Reserved	1/8
100	Geographic unicast addresses	1/8
101	Reserved	1/8
110	Reserved	1/8

Table 18.3 *Type prefixes for IPv6 addresses (continued)*

Type Prefix	*Type*	*Fraction*
1110	Reserved	1/16
1111 0	Reserved	1/32
1111 10	Reserved	1/64
1111 110	Reserved	1/128
1111 1110 0	Reserved	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-based unicast addresses.** The provider-based address is generally used by a normal host as a unicast address. The address format is shown in Figure 18.22.

Figure 18.22 *Provider-based address*



Fields for the provider-based addresses are as follows:

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

b. **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.

c. **Provider identifier.** This variable-length field identifies the provider for Internet access. A 16-bit length is recommended for this field.

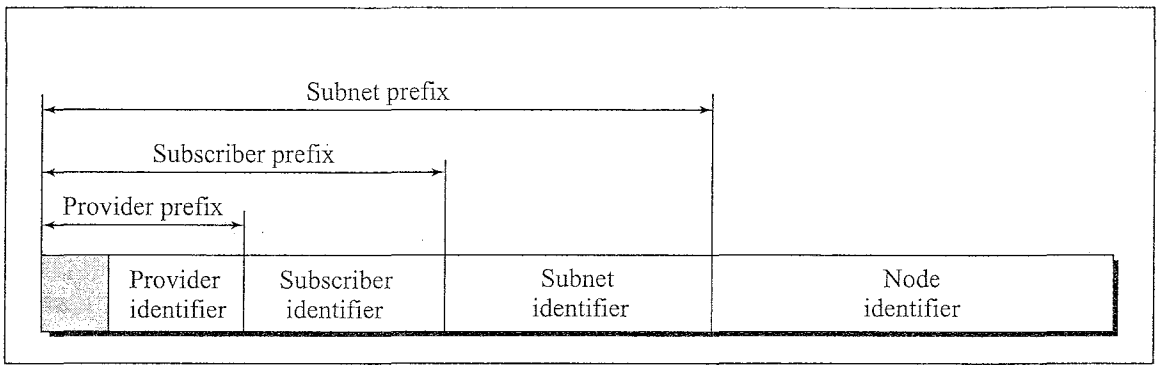
d. **Subscriber identifier.** When an organization subscribes to the Internet through a provider, it is assigned a subscriber identification. A 24-bit length is recommended for this field.

e. **Subnet identifier.** Each subscriber can have many different subnetworks and each network can have different identifiers. The subnet identifier defines a specific network under the territory of the subscriber. A 32-bit length is recommended for this field.

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

We can think of a provider-based address as a hierarchical identity having several prefixes. As shown in Figure 18.23, each prefix defines a level of hierarchy. The type prefix defines the type, the registry prefix uniquely defines the registry level, the provider prefix uniquely defines a provider, the subscriber prefix uniquely defines a subscriber, and the subnet prefix uniquely defines a subnet.

Figure 18.23 *Address hierarchy*



- **Other addresses.** Other address types are used for purposes that are beyond the scope of this book. For further information, see *TCP/IP Protocol Suite* by Behrouz Forouzan.

IPv6 Packet Format

The IPv6 packet is shown in Figure 18.24. 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 usually contain up to 65,535 bytes of information.

Base Header Figure 18.25 shows the base header with its eight fields. 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. We will discuss this field later.

Figure 18.24 *IPv6 datagram*

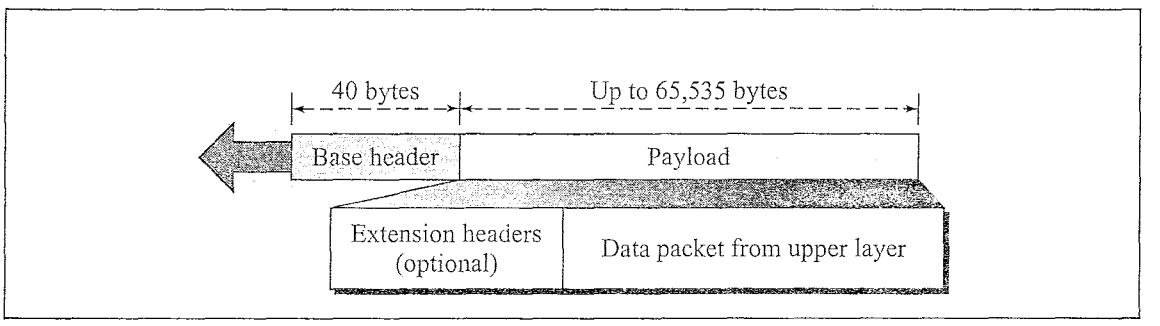
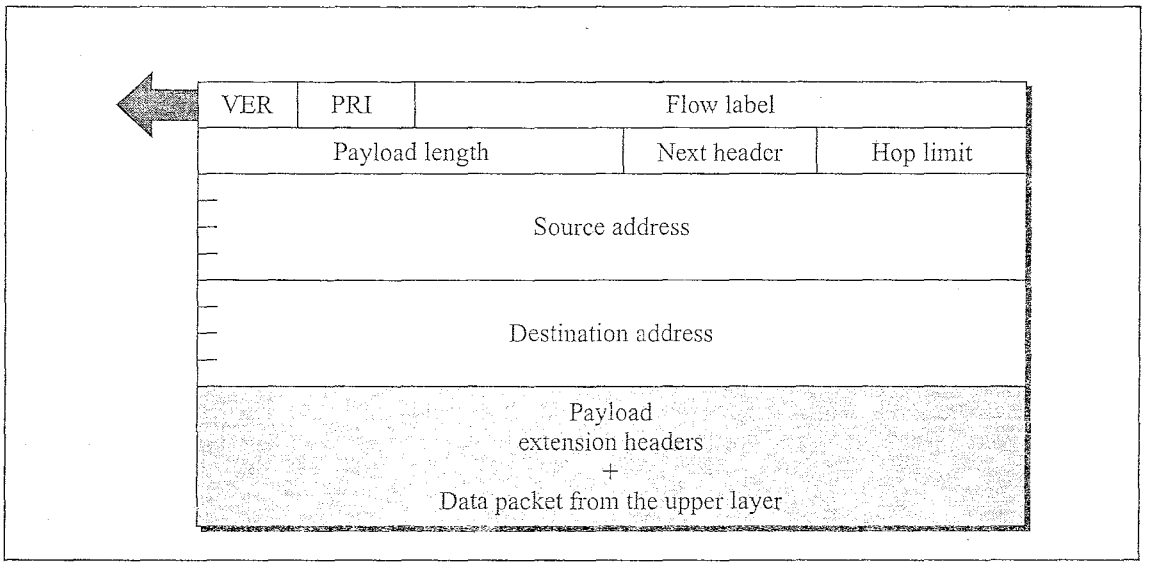


Figure 18.25 *Format of an IPv6 datagram*



- **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. We will discuss this field later.
- **Payload length.** This 2-byte payload length field defines the total 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 for an upper-layer protocol such as UDP or TCP. Each extension header also contains this field. Table 18.4 shows the values of next headers. Note that this field in version 4 is called the *protocol*.

Table 18.4 *Next header codes*

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing

Table 18.4 *Next header codes (continued)*

Code	*Next Header*
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

- **Hop limit.** The 8-bit **hop limit** field serves the same purpose as the TTL field in IPv4.
- **Source address.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address.** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

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 priority will be discarded. IPv6 divides traffic into two broad categories: congestion-controlled and noncongestion-controlled.

- **Congestion-Controlled Traffic.** If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as **congestion-controlled traffic.** For example, the TCP protocol, which uses the sliding window protocol, can easily respond to the traffic. In congestion-controlled traffic, it is understood that packets may arrive delayed or even be lost or received out of order. Congestion-controlled data are assigned priorities from 0 to 7, as listed in Table 18.5. A priority of 0 is the lowest; a priority of 7 is the highest.

Table 18.5 *Priorities for congestion-controlled traffic*

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

The priority descriptions are as follows:

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

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

c. **Unattended data traffic.** If the user is not waiting (attending) for the data to be received, the packet will be given priority 2. Email belongs to this group. A user initiates an email message to another user, but the receiver does not know that an email will arrive soon. In addition, an email is usually stored before it is forwarded. A little bit of delay is of little consequence.

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

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

f. **Control traffic.** Control traffic has been given the highest priority (7) in this category. Routing protocols such as OSPF and RIP and management protocols such as SNMP use this priority.

- **Noncongestion-Controlled Traffic.** This refers to a type of traffic that expects minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. In other words, the source does not adapt itself to congestion. Real-time audio and video are good examples of this type of traffic.

  Priority numbers from 8 to 15 are assigned to **noncongestion-controlled traffic.** Although there are not yet any particular standard assignments for this type of data, the priorities are usually assigned based on how much the quality of received data can be affected by discarding some packets. Data containing less redundancy (such as low-fidelity audio or video) can be given a higher priority (15). Data containing more redundancy (such as high-fidelity audio or video) should be given lower priority (8) (see Table 18.6).

Table 18.6 *Priorities for noncongestion-controlled traffic*

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

Flow Label A sequence of packets, sent from a particular source to a particular destination, that needs special handling by routers is called a *flow* of packets. The combination of the source address and the value of the *flow label* uniquely defines a flow of packets.

To a router, a flow is a sequence of packets that share the same characteristics, such as traveling the same path, using the same resources, having the same kind of security,

and so on. A router that supports the handling of flow labels has a flow label table. The table has an entry for each active flow label; each entry defines the services required by the corresponding flow label. When the router receives a packet, it consults its flow label table to find the corresponding entry for the flow label value defined in the packet. It then provides the packet with the services mentioned in the entry. However, note that the flow label itself does not provide the information for the entries of the flow label table; that information is provided by other means such as the hop-by-hop options or other protocols.

In its simplest form, a flow label can be used to speed up the processing of a packet by a router. When a router receives a packet, instead of consulting the routing table and going through a routing algorithm to define the address of the next hop, it can easily look in a flow label table for the next hop.

In its more sophisticated form, a flow label can be used to support the transmission of real-time audio and video. Real-time audio or video, particularly in digital form, requires resources such as high bandwidth, large buffers, long processing time, and so on. A process can make a reservation for these resources beforehand to guarantee that real-time data will not be delayed due to a lack of resources. The use of real-time data and the reservation of these resources requires other protocols such as Real Time Protocol (RTP) and Resource Reservation Protocol (RSVP) in addition to IPv6.

To allow the effective use of flow labels, three rules have been defined:

1. The flow label is assigned to a packet by the source host. The label is a random number between 1 and $2^{24} - 1$. A source must not reuse a flow label for a new flow while the existing flow is still alive.
2. If a host does not support the flow label, it sets this field to zero. If a router does not support the flow label, it simply ignores it.
3. All packets belonging to the same flow should have the same source, same destination, same priority, and same options.

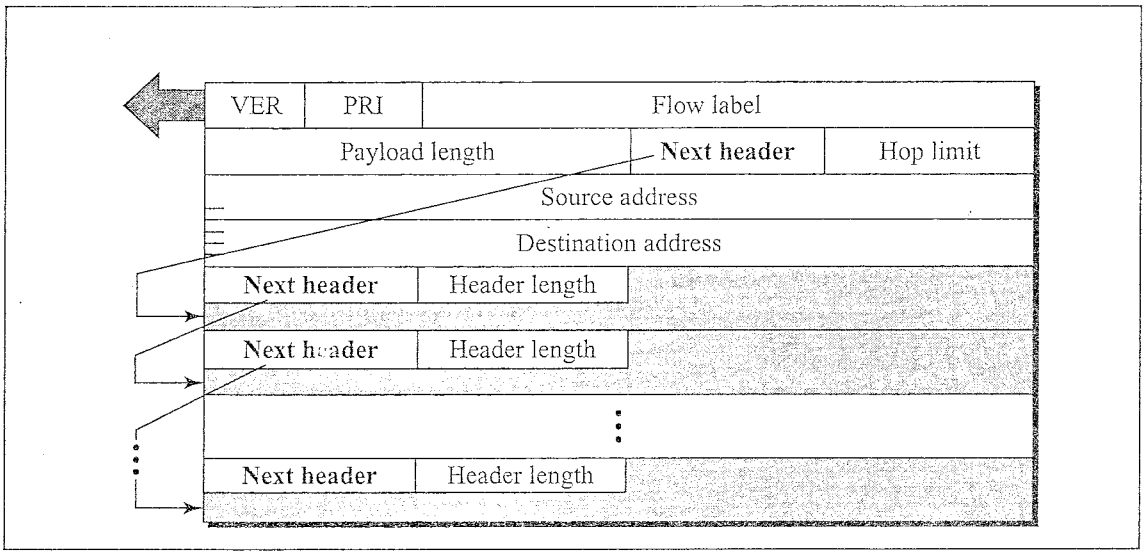
Comparison Table 18.7 compares IPv4 and IPv6 headers.

Table 18.7 *Comparison between IPv4 and IPv6 packet header*

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

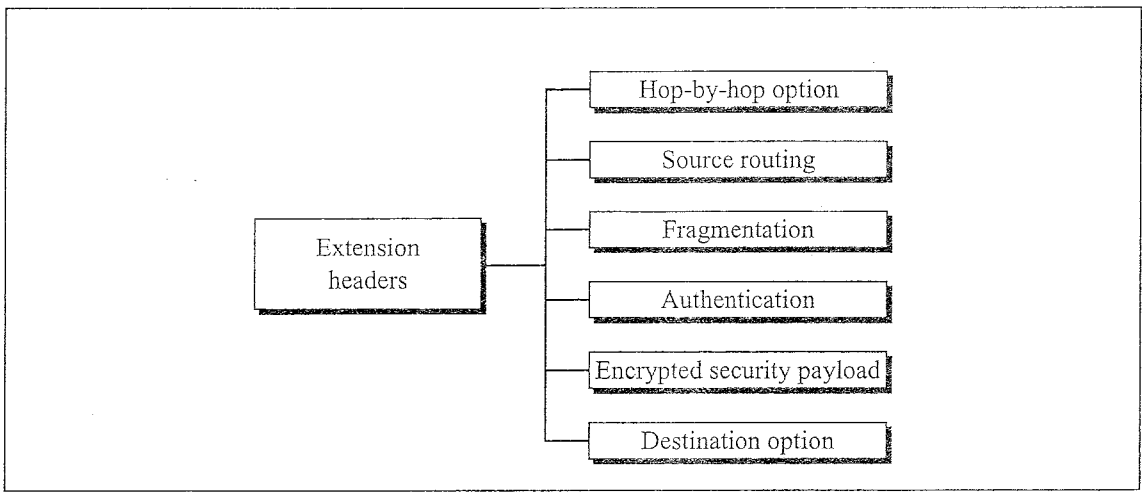
Extension Headers The length of the base header is fixed at 40 bytes. However, to give more functionality to the IP datagram, the base header can be followed by up to six extension headers. Many of these headers are options in IPv4. Figure 18.26 shows the extension header format.

Figure 18.26 *Extension header format*



Six types of extension headers have been defined: hop-by-hop option, source routing, fragmentation, authentication, encrypted security payload, and destination option (see Figure 18.27).

Figure 18.27 *Extension header types*

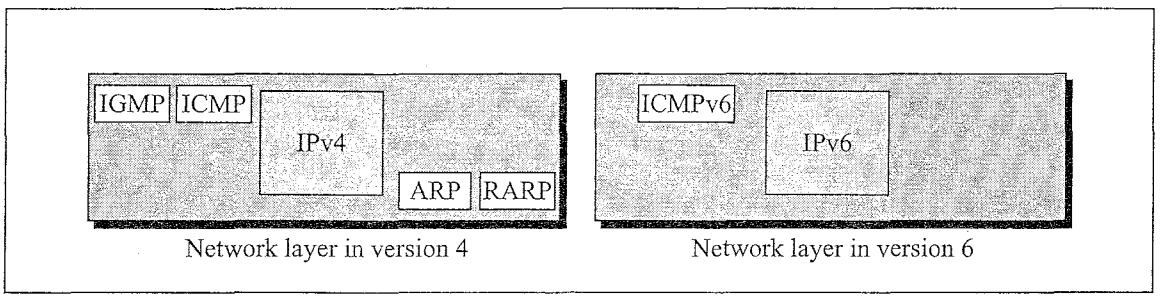


ICMPv6

Another protocol that has been modified in version 6 is ICMP (ICMPv6). This new version follows the same strategy and purposes of version 4, but 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 ICMPv6. Figure 18.28 compares the network layers of version 4 and version 6.

Figure 18.28 *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 is not used often.

Figure 18.29 shows two broad categories of ICMP messages: error-reporting and query.

Figure 18.29 *Categories of ICMP messages*

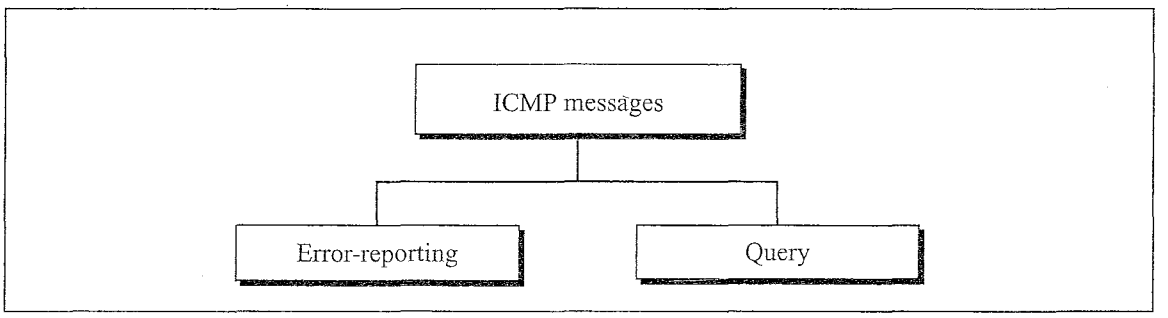


Figure 18.30 shows the five different error-reporting messages: destination unreachable, packet too big, time exceeded, parameter problems, and redirection.

Figure 18.30 *Types of error-reporting messages*

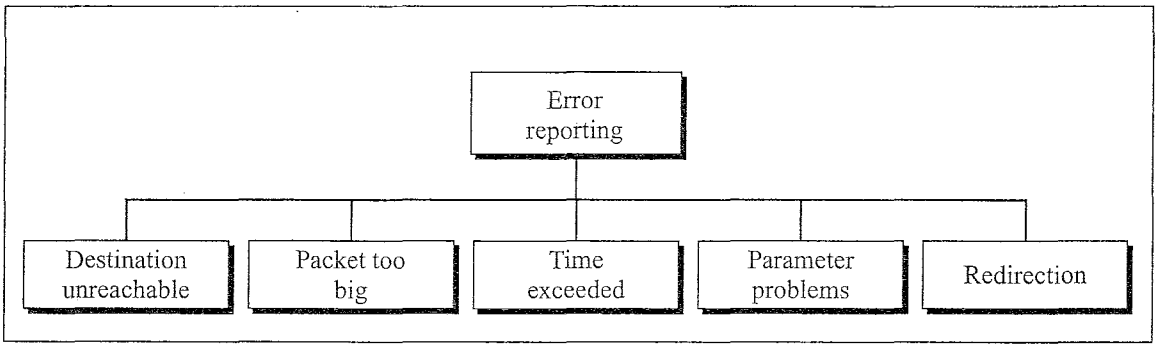
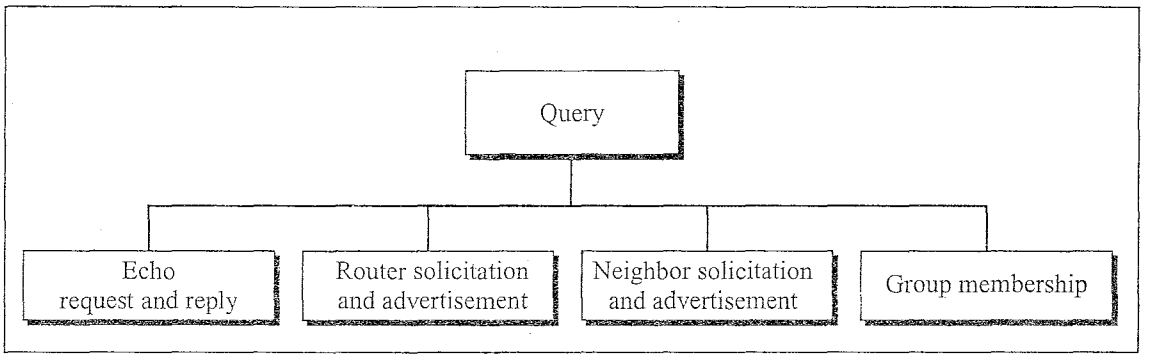


Figure 18.31 shows the four different query messages: echo request and reply, router solicitation and advertisement, neighbor solicitation and advertisement, and group membership.

Figure 18.31 *Types of query messages*



18.8 KEY TERMS

abbreviation
address resolution protocol (ARP)
address space
Advanced Research Project Agency (ARPA)
Advanced Research Project Agency Network (ARPANET)
anycast address
base header
broadcasting
class of address
congestion-controlled traffic
datagram
dotted-decimal notation
encapsulation
extension header
flow label
fragmentation
hexadecimal colon notation
hop limit
host
hostid
Internet address
Internet Control Message Protocol (ICMP)
Internet Group Message Protocol (IGMP)
Internetworking Control Message Protocol, version 6 (ICMPv6)
Internetworking Protocol (IP)
Internetworking Protocol, next generation (IPng)
Internetworking Protocol, version 6 (IPv6)
IP address
IP address class
IP datagram
IPv4
masking
multicast address
multicasting
netid
next header
node identifier
noncongestion-controlled traffic
options
port address
provider identifier
provider-based unicast address
registry identifier
reverse address resolution protocol (RARP)
segment
subnet identifier
subnetid
subnetting
subnetwork
subnetwork address
subscriber identifier
Transmission Control Protocol (TCP)
Transmission Control Protocol/ Internetworking Protocol (TCP/IP)
type identifier

type prefix
unicast address
urgent data
user datagram
User Datagram Protocol (UDP)

18.9 SUMMARY

- Transmission Control Protocol/Internetworking Protocol (TCP/IP) is a set of rules and procedures that govern the exchange of messages in an internetwork.
- TCP/IP was originally developed as a protocol for networks that wanted to be connected to ARPANET, a U.S. Department of Defense project. ARPANET is now known as the Internet.
- TCP/IP is a five-layer protocol suite with four bottom layers that match the OSI model fairly closely. The highest level, the application layer, corresponds to OSI's top three layers.
- The Internetwork Protocol (IP) is defined at the network layer. IP is unreliable and connectionless.
- The IP packet, called the datagram, consists of a variable header and a variable data field.
- An internet address (better known as the IP address) uniquely defines the connection of a host to its network.
- The 4-byte IP address is usually written in dotted decimal notation.
- Subnetting allows an additional level of hierarchy in IP addressing.
- The address resolution protocol (ARP) finds the physical address of a device if its IP address is known.
- The reverse address resolution protocol (RARP) will find a host's IP address from its physical address.
- The internet control message protocol (ICMP) handles control and error messages in the IP layer.
- There are two protocols at the transport level:
  a. User Datagram Protocol (UDP).
  b. Transmission Control Protocol (TCP).
- A protocol port is a source or destination point of an executing program in the application layer.
- UDP is unreliable and connectionless. UDP communication is port-to-port. The UDP packet is called a user datagram.
- TCP is reliable and connection-oriented. TCP communication is port-to-port. The packet is called a segment.
- IPv6, the latest proposed version of the Internet Protocol, has a 128-bit address space, a revised header format, new options, an allowance for extension, support for resource allocation, and increased security measures.
- IPv6 uses hexadecimal colon notation with abbreviation methods available.
- There are three types of addresses: unicast, anycast, and multicast.

- An IP datagram is composed of a base header and a payload.
- The 40-byte base header consists of the version, priority, flow label, payload length, next header, hop limit, source address, and destination address fields.
- The priority field is a measure of the importance of a datagram.
- The flow label identifies the special-handling needs of a sequence of packets.
- A payload consists of optional extension headers and data from an upper layer.
- Extension headers add functionality to the IPv6 datagram.
- ICMPv6, like version 4, reports errors, handles group memberships, updates specific router and host tables, and checks the viability of a host.
- The five error-reporting messages deal with unreachable destinations, packets that are too big, expired timers for fragments and hop counts, header problems, and inefficient routing.
- Query messages are in the form of a response and a reply.

18.10 PRACTICE SET

Multiple-Choice Questions

1. Which OSI layer corresponds to the TCP-UDP layer?
   a. physical
   b. data link
   c. network
   d. transport
2. Which OSI layer corresponds to the IP layer?
   a. physical
   b. data link
   c. network
   d. transport
3. Which OSI layer(s) correspond to TCP/IP's application layer?
   a. application
   b. presentation
   c. session
   d. all of the above
4. Which of the following is true about the IP address?
   a. It's divided into exactly two classes.
   b. It contains a fixed-length hostid.
   c. It was established as a user-friendly interface.
   d. It is 32 bits long.

5. Which IP address class has few hosts per network?
   a. A
   b. B
   c. C
   d. D
6. The purpose of ARP on a network is to find the _____ given the _____.
   a. Internet address, domain name
   b. Internet address, netid
   c. Internet address, station address
   d. station address, Internet address
7. Which of the following apply to UDP?
   a. is unreliable and connectionless
   b. contains destination and source port addresses
   c. reports certain errors
   d. all of the above
8. Which of the following applies(y) to both UDP and TCP?
   a. transport layer protocols
   b. port-to-port communication
   c. services of IP layer used
   d. all of the above
9. Which of the following is a class A host address?
   a. 128.4.5.6
   b. 117.4.5.1
   c. 117.0.0.0
   d. 117.8.0.0
10. Which of the following is a class B host address?
    a. 230.0.0.0
    b. 130.4.5.6
    c. 230.0.0.0
    d. 30.4.5.6
11. Which of the following is a class C host address?
    a. 230.0.0.0
    b. 130.4.5.6
    c. 200.1.2.3
    d. 30.4.5.6
12. TCP/IP's _____ layer corresponds to the OSI model's top three layers.
    a. application
    b. presentation
    c. session
    d. transport

13. When a host knows its physical address but not its IP address, it can use _____.
    a. ICMP
    b. IGMP
    c. ARP
    d. RARP
14. This transport layer protocol requires acknowledgment.
    a. UDP
    b. TCP
    c. FTP
    d. NVT
15. Which of the following is the default mask for the address 198.0.46.201?
    a. 255.0.0.0
    b. 255.255.0.0
    c. 255.255.255.0
    d. 255.255.255.255
16. Which of the following is the default mask for the address 98.0.46.201?
    a. 255.0.0.0
    b. 255.255.0.0
    c. 255.255.255.0
    d. 255.255.255.255
17. Which of the following is the default mask for the address 190.0.46.201?
    a. 255.0.0.0
    b. 255.255.0.0
    c. 255.255.255.0
    d. 255.255.255.255
18. For a maximum number of hops, set the hop limit field to decimal _____.
    a. 16
    b. 15
    c. 42
    d. 0
19. In IPv6, a datagram with a priority of _____ will be discarded before a datagram with a priority of 12.
    a. 11
    b. 7
    c. 0
    d. any of the above
20. The maximum size for an IPv6 datagram is _____ bytes.
    a. 65,535
    b. 65,575

c. 2^{32}

d. $2^{32} + 40$

21. Which of the following types of ICMP messages need to be encapsulated into an IP datagram?
    a. neighbor solicitation
    b. echo response
    c. redirection
    d. all of the above

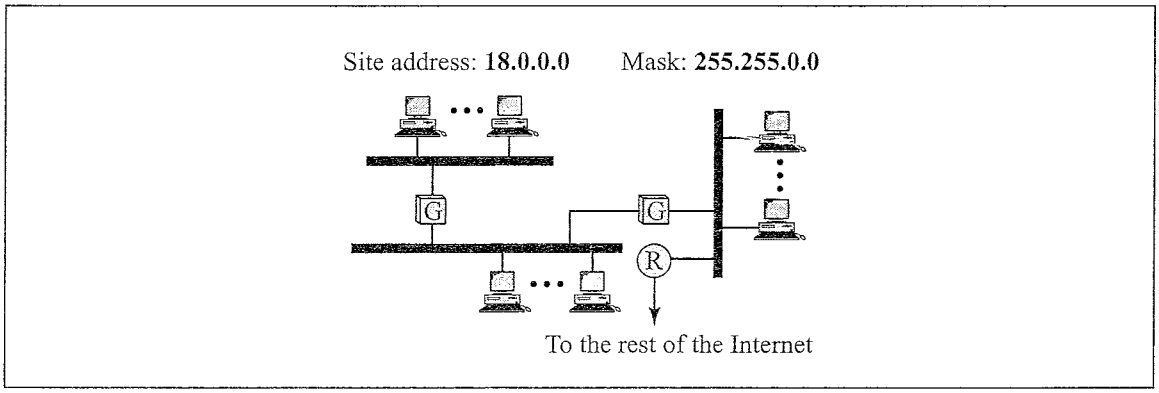
Exercises

22. Show by calculation how many networks (not hosts) each IP address class (A, B, and C only) can have.
23. Show by calculation how many hosts per network each IP address class (A, B, and C only) can have.
24. Change the following IP addresses from dotted-decimal notation to binary notation:
    a. 114.34.2.8
    b. 129.14.6.8
    c. 208.34.54.12
    d. 238.34.2.1
    e. 241.34.2.8
25. Change the following IP addresses from binary notation to dotted-decimal notation:
    a. 01111111 11110000 01100111 01111101
    b. 10101111 11000000 11110000 00011101
    c. 11011111 10110000 00011111 01011101
    d. 11101111 11110111 11000111 00011101
    e. 11110111 11110011 10000111 11011101
26. Find the class of the following IP addresses:
    a. 208.34.54.12
    b. 238.34.2.1
    c. 114.34.2.8
    d. 129.14.6.8
    e. 241.34.2.8
27. Find the class of the following IP addresses:
    a. 11110111 11110011 10000111 11011101
    b. 10101111 11000000 11110000 00011101
    c. 11011111 10110000 00011111 01011101
    d. 11101111 11110111 11000111 00011101
    e. 01111111 11110000 01100111 01111101

28. Find the netid and the hostid of the following IP addresses:
    a. 114.34.2.8
    b. 19.34.21.5
    c. 171.34.14.8
    d. 190.12.67.9
    e. 220.34.8.9
    f. 205.23.67.8
29. Find the network address of the following IP addresses:
    a. 23.67.12.1
    b. 126.23.4.0
    c. 190.12.67.9
    d. 220.34.8.9
    e. 237.34.8.2
    f. 240.34.2.8
    g. 247.23.4.78
30. Write the following masks in binary notation:
    a. 255.255.255.0
    b. 255.255.0.0
    c. 255.255.224.0
    d. 255.255.255.240
31. Write the following masks in dotted-decimal notation:
    a. 11111111111111111111111111111000
    b. 11111111111111111111111111100000
    c. 11111111111111111111100000000000
32. Show in binary format each of the following masks used in class B networks.
    a. 255.255.192.0
    b. 255.255.0.0
    c. 255.255.224.0
    d. 255.255. 255.0
33. Show in binary format each of the following masks used in class C networks.
    a. 255.255.255.192
    b. 255.255.255.224
    c. 255.255.255.240
    d. 255.255. 255.0
34. What is the maximum number of subnets in class A networks using the following masks?
    a. 255.255.192.0
    b. 255.192.0.0
    c. 255.255.224.0
    d. 255.255.255.0

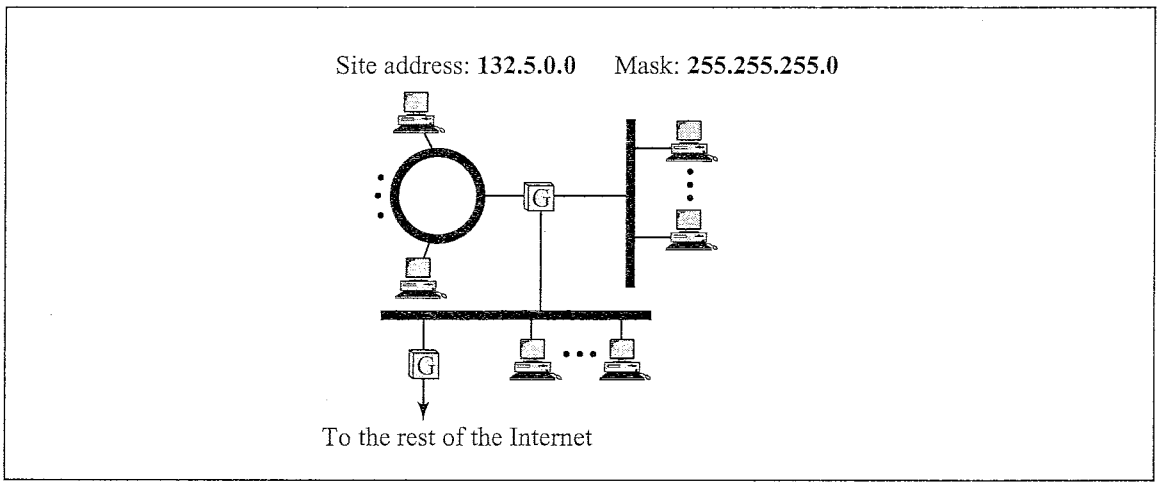
35. What is the maximum number of subnets in class C networks using the following masks?
    a. 255.255.255.192
    b. 255.255.255.224
    c. 255.255.255.240
    d. 255.255.255.0
36. Find the subnetwork address for the following:
    IP address: 125.34.12.56 Mask: 255.255.0.0
37. Find the subnetwork address for the following:
    IP address: 120.14.22.16 Mask: 255.255.128.0
38. Find the subnetwork address and host address for the following:
    IP address: 200.34.22.156 Mask: 255.255.255.240
39. Figure 18.32 shows a site with a given network address and mask. The administration has divided the site into several subnetworks. Choose appropriate subnetwork addresses, host addresses, and router addresses.

Figure 18.32 *Site for Exercise 39*



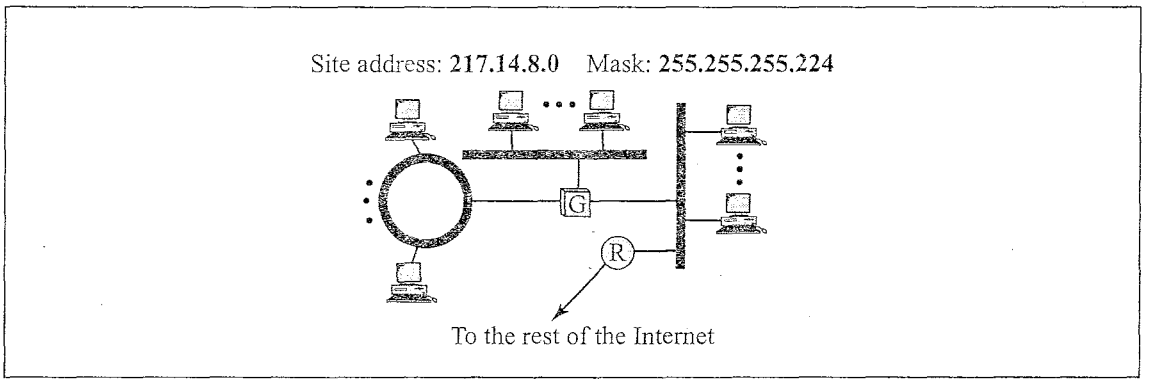
40. Figure 18.33 shows a site with a given network address and mask. The administration has divided the site into several subnetworks. Choose appropriate subnetwork addresses, host addresses, and router addresses.

Figure 18.33 *Site for Exercise 40*



41. Figure 18.34 shows a site with a given network address and mask. The administrator has divided the site into several subnetworks. Choose appropriate subnetwork addresses, host addresses, and router addresses.

Figure 18.34 *Site for Exercise 41*



42. Show the shortest form of the following addresses:
    a. 2340:1ABC:119A:A000:0000:0000:0000:0000
    b. 0000:00AA:0000:0000:0000:0000:119A:A231
    c. 2340:0000:0000:0000:0000:119A:A001:0000
    d. 0000:0000:0000:2340:0000:0000:0000:0000
43. Show the original (unabbreviated) form of the following addresses:
    a. 0::0
    b. 0:AA::0
    c. 0:1234::3
    d. 123::1:2
44. What is the type of each of the following addresses:
    a. FE80::12
    b. FEC0::24A2
    c. 4821::14:22
    d. 54EF::A234:2
45. Show the provider prefix (in hexadecimal colon notation) of an address assigned to a subscriber if it is registered in the USA with the provider identification ABC1.
46. Show in hexadecimal colon notation the IPv6 address compatible to the IPv4 address 129.6.12.34.
47. Show in hexadecimal colon notation the IPv6 address mapped to the IPv4 address 129.6.12.34.
48. Show in hexadecimal colon notation the link local address in which the node identifier is 0::123/48.
49. Show in hexadecimal colon notation the site local address in which the node identifier is 0::123/48.
50. Show in hexadecimal colon notation the permanent multicast address used in a link local scope.

51. What are the possible first two bytes for a multicast address?
52. An IPv6 packet consists of the base header and a TCP segment. The length of data is 320 bytes. Show the packet and enter a value for each field.
53. An IPv6 packet consists of a base header and a TCP segment. The length of data is 128,000 bytes (jumbo payload). Show the packet and enter a value for each field.
54. How many more addresses are available with IPv6 than IPv4?
55. In designing the IPv4-mapped address, why didn't the designers just prepend 96 1s to the IPv4 address?