CHAPTER 22

Next Generation IP

The address depletion of IPv4 and other shortcomings of this protocol prompted a new version of IP in the early 1990s. The new version, which is called ***Internet Protocol version 6 (IPv6)*** or ***IP new generation (IPng)*** was a proposal to augment the address space of IPv4 and at the same time redesign the format of the IP packet and revise some auxiliary protocols such as ICMP. It is interesting to know that IPv5 was a proposal, based on the OSI model, that never materialized.

The following lists the main changes in the IPv6 protocol: larger address space, better header format, new options, allowance for extension, support for resource allocation, and support for more security. The implementation of theses changes made it necessary to create a new version of the ICMP protocol, ICMPv6.

This chapter is made of four sections:

- The first section discusses the addressing mechanism in the new generation of the Internet. The section first describes the representation and address space. It then shows the allocation in the address space. The section finally explains autoconfiguration and renumbering, which makes it easy for a host to move from one network to another.
- The second section discusses IPv6 protocol. First the new packet format is described. The section then shows how use of extension headers can replace the options in version 4.
- The third section discusses ICMPv6. The section describes how the new protocol replaces several auxiliary protocols in version 4. The section also divides the messages in this protocol into four categories and describes them.
- The fourth section briefly shows how transition can be made from the current version to the new one smoothly. The section explains three strategies that need to be followed for this smooth transition.

22.1 IPv6 ADDRESSING

The main reason for migration from IPv4 to IPv6 is the small size of the address space in IPv4. In this section, we show how the huge address space of IPv6 prevents address depletion in the future. We also discuss how the new addressing responds to some problems in the IPv4 addressing mechanism. An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4.

22.1.1 Representation

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

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

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

Abbreviation

Although an IPv6 address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section can be omitted. Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0. Note that 3210 cannot be abbreviated. Further abbreviation, often called ***zero compression,*** can be applied to colon hex notation if there are consecutive sections consisting of zeros only. We can remove all the zeros and replace them with a double semicolon.

FDEC:0:0:0:0:BBFF:0:FFFF ⟶ **FDEC::BBFF:0:FFFF**

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

Mixed Notation

Sometimes we see a mixed representation of an IPv6 address: colon hex and dotted-decimal notation. This is appropriate during the transition period in which an IPv4 address is embedded in an IPv6 address (as the rightmost 32 bits). We can use the colon hex notation for the leftmost six sections and four-byte dotted-decimal notation instead of the rightmost two sections. However, this happens when all or most of the leftmost sections of the IPv6 address are 0s. For example, the address (::130.24.24.18) is a legitimate address in IPv6, in which the zero compression shows that all 96 leftmost bits of the address are zeros.

CIDR Notation

As we will see shortly, IPv6 uses hierarchical addressing. For this reason, IPv6 allows slash or CIDR notation. For example, the following shows how we can define a prefix of 60 bits using CIDR. We will later show how an IPv6 address is divided into a prefix and a suffix.

FDEC::BBFF:0:FFFF/60

22.1.2 Address Space

The address space of IPv6 contains 2^{128} addresses. This address space is 2^{96} times the IPv4 address—definitely no address depletion—as shown, the size of the space is

340, 282, 366, 920, 938, 463, 374, 607, 431, 768, 211, 456.

To give some idea about the number of addresses, we assume that only 1/64 (almost 2 percent) of the addresses in the space can be assigned to the people on planet Earth and the rest are reserved for special purposes. We also assume that the number of people on the earth is soon to be 2^{34} (more than 16 billion). Each person can have 2^{88} addresses to use. Address depletion in this version is impossible.

Three Address Types

In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast.

Unicast Address

A unicast address defines a single interface (computer or router). The packet sent to a unicast address will be routed to the intended recipient.

Anycast Address

An **anycast address** defines a group of computers that all share a single address. A packet with an anycast address is delivered to only one member of the group, the most reachable one. An anycast communication is used, for example, when there are several servers that can respond to an inquiry. The request is sent to the one that is most reachable. The hardware and software generate only one copy of the request; the copy reaches only one of the servers. IPv6 does not designate a block for anycasting; the addresses are assigned from the unicast block.

Multicast Address

A multicast address also defines a group of computers. However, there is a difference between anycasting and multicasting. In anycasting, only one copy of the packet is sent to one of the members of the group; in multicasting each member of the group receives a copy. As we will see shortly, IPv6 has designated a block for multicasting from which the same address is assigned to the members of the group. It is interesting that IPv6 does not define broadcasting, even in a limited version. IPv6 considers broadcasting as a special case of multicasting.

22.1.3 Address Space Allocation

Like the address space of IPv4, the address space of IPv6 is divided into several blocks of varying size and each block is allocated for a special purpose. Most of the blocks are still unassigned and have been set aside for future use. Table 22.1 shows only the assigned blocks. In this table, the last column shows the fraction each block occupies in the whole address space.

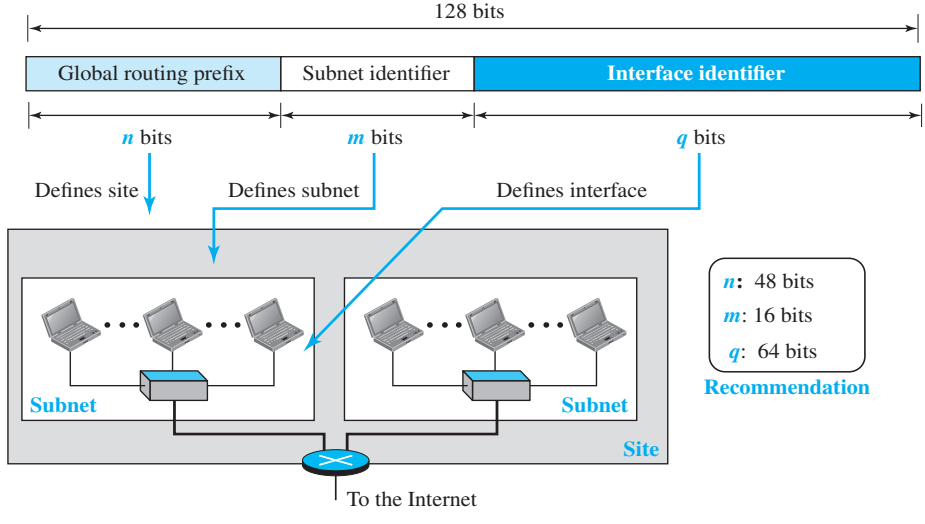
Table 22.1 *Prefixes for assigned IPv6 addresses*

Block prefix	*CIDR*	*Block assignment*	*Fraction*
0000 0000	0000::/8	Special addresses	1/256
001	**2000::/3**	**Global unicast**	**1/8**
1111 110	FC00::/7	Unique local unicast	1/128
1111 1110 10	FE80::/10	Link local addresses	1/1024
1111 1111	FF00::/8	Multicast addresses	1/256

Global Unicast Addresses

The block in the address space that is used for unicast (one-to-one) communication between two hosts in the Internet is called the *global unicast address block*. CIDR for the block is 2000::**/3**, which means that the three leftmost bits are the same for all addresses in this block (001). The size of this block is 2^{125} bits, which is more than enough for Internet expansion for many years to come. An address in this block is divided into three parts: *global routing prefix* (*n* bits), *subnet identifier* (*m* bits), and *interface identifier* (*q* bits), as shown in Figure 22.1. The figure also shows the recommended length for each part.

Figure 22.1 *Global unicast address*



The global routing prefix is used to route the packet through the Internet to the organization site, such as the ISP that owns the block. Since the first three bits in this part are fixed (001), the rest of the 45 bits can be defined for up to 2^{45} sites (a private organization or an ISP). The global routers in the Internet route a packet to its destination site based on the value of n. The next m bits (16 bits based on recommendation) define a subnet in an organization. This means that an organization can have up to $2^{16} = 65{,}536$ subnets, which is more than enough.

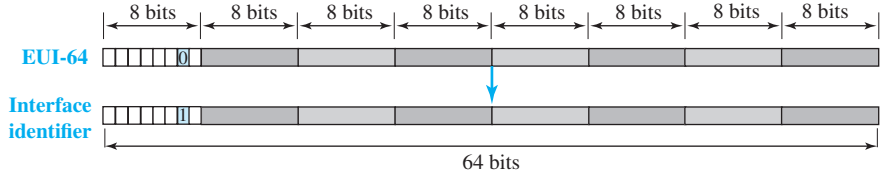
The last q bits (64 bits based on recommendation) define the interface identifier. The interface identifier is similar to hostid in IPv4 addressing, although the term *interface identifier* is a better choice because, as we discussed earlier, the host identifier actually defines the interface, not the host. If the host is moved from one interface to another, its IP address needs to be changed.

In IPv4 addressing, there is not a specific relation between the hostid (at the IP level) and link-layer address (at the data-link layer) because the link-layer address is normally much longer than the hostid. The IPv6 addressing allows this relationship. A link-layer address whose length is less than 64 bits can be embedded as the whole or part of the interface identifier, eliminating the mapping process. Two common link-layer addressing schemes can be considered for this purpose: the 64-bit extended unique identifier (EUI-64) defined by IEEE and the 48-bit link-layer address defined by Ethernet.

Mapping EUI-64

To map a 64-bit physical address, the global/local bit of this format needs to be changed from 0 to 1 (local to global) to define an interface address, as shown in Figure 22.2.

Figure 22.2 *Mapping for EUI-64*



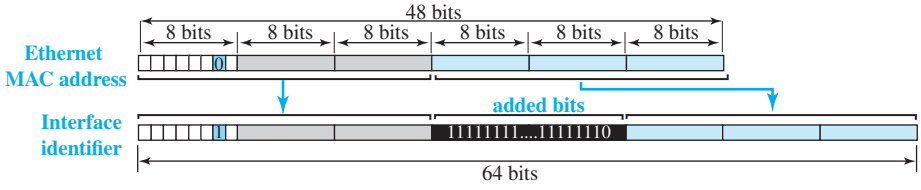
Mapping Ethernet MAC Address

Mapping a 48-bit Ethernet address into a 64-bit interface identifier is more involved. We need to change the local/global bit to 1 and insert an additional 16 bits. The additional 16 bits are defined as 15 ones followed by one zero, or $FFFE_{16}$. Figure 22.3 shows the mapping.

Example 22.1

An organization is assigned the block 2000:1456:2474/48. What is the CIDR notation for the blocks in the first and second subnets in this organization?

Figure 22.3 *Mapping for Ethernet MAC*



Solution

Theoretically, the first and second subnets should use the blocks with subnet identifier 0001_{16} and 0002_{16}. This means that the blocks are 2000:1456:2474:0000/64 and 2000:1456:2474:0001/64.

Example 22.2

Using the format we defined for Ethernet addresses, find the interface identifier if the physical address in the EUI is $(\textbf{F5-A9-23-EF-07-14-7A-D2})_{16}$.

Solution

We only need to change the seventh bit of the first octet from 0 to 1 and change the format to colon hex notation. The result is **F7A9:23EF:0714:7AD2.**

Example 22.3

Using the format we defined for Ethernet addresses, find the interface identifier if the Ethernet physical address is $(\textbf{F5-A9-23-14-7A-D2})_{16}$.

Solution

We only need to change the seventh bit of the first octet from 0 to 1, insert two octets $FFFE_{16}$ and change the format to colon hex notation. The result is **F7A9:23FF:FE14:7AD2** in colon hex.

Example 22.4

An organization is assigned the block 2000:1456:2474/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is $(\textbf{F5-A9-23-14-7A-D2})_{16}$?

Solution

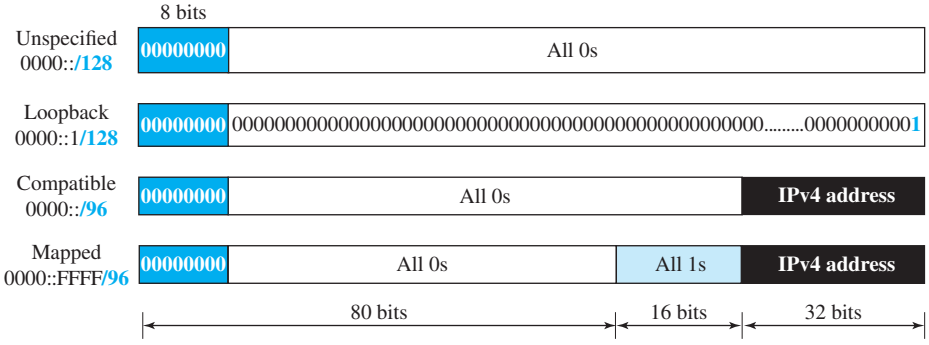
The interface identifier for this interface is **F7A9:23FF:FE14:7AD2** (see Example 22.3). If we append this identifier to the global prefix and the subnet identifier, we get:

2000:1456:2474:0003:F7A9:23FF:FE14:7AD2/128

Special Addresses

After discussing the global unicast block, let us discuss the characteristics and purposes of assigned and reserved blocks in the first row of Table 22.1. Addresses that use the prefix (**0000::/8**) are reserved, but part of this block is used to define some special addresses. Figure 22.4 shows the assigned addresses in this block.

Figure 22.4 *Special addresses*



The unspecified address is a subblock containing only one address, which is used during bootstrap when a host does not know its own address and wants to send an inquiry to find it (see DHCP section).

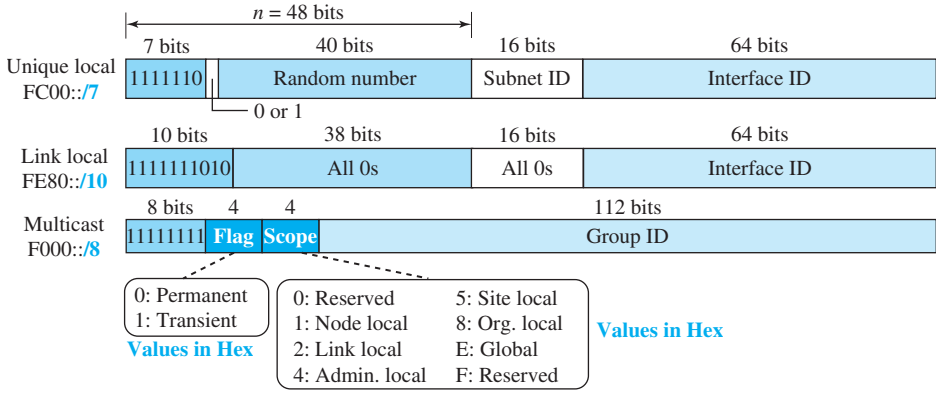
The loopback address also consists of one address. We discussed loopback addresses for IPv4 before. In IPv4 the block is made of a range of addresses; in IPv6, the block has only a single address in it.

As we will see later, during the transition from IPv4 to IPv6, hosts can use their IPv4 addresses embedded in IPv6 addresses. Two formats have been designed for this purpose: compatible and mapped. A **compatible address** is an address of 96 bits of zero followed by 32 bits of IPv4 address. It is used when a computer using IPv6 wants to send a message to another computer using IPv6. A **mapped address** is used when a computer already migrated to version 6 wants to send an address to a computer still using version 4. A very interesting point about mapped and compatible addresses is that they are designed such that, when calculating the checksum, one can use either the embedded address or the total address because extra 0s or 1s in multiples of 16 do not have any effect in checksum calculation. This is important for UDP and TCP, which use a pseudoheader to calculate the checksum, because the checksum calculation is not affected if the address of the packet is changed from IPv6 to IPv4 by a router.

Other Assigned Blocks

IPv6 uses two large blocks for private addressing and one large block for multicasting, as shown in Figure 22.5. A subblock in a **unique local unicast block** can be privately created and used by a site. The packet carrying this type of address as the destination address is not expected to be routed. This type of address has the identifier 1111 110, the next bit can be 0 or 1 to define how the address is selected (locally or by an authority). The next 40 bits are selected by the site using a randomly generated number of length 40 bits. This means that the total of 48 bits defines a subblock that looks like a global unicast address. The 40-bit random number makes the probability of duplication of the address extremely small. Note the similarity between the format of these addresses and the global unicast. The second block, designed for private addresses, is the **link local block.** A subblock in this block can be used as

Figure 22.5 *Unique local unicast block*



a private address in a network. This type of address has the block identifier 1111111010. The next 54 bits are set to zero. The last 64 bits can be changed to define the interface for each computer. Note the similarity between the format of these addresses and the global unicast address.

We discussed multicast addresses of IPv4 earlier in the chapter. Multicast addresses are used to define a group of hosts instead of just one. In IPv6 a large block of addresses are assigned for multicasting. All these addresses use the prefix 11111111. 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, as shown in the figure.

22.1.4 Autoconfiguration

One of the interesting features of IPv6 addressing is the **autoconfiguration** of hosts. As we discussed in IPv4, the host and routers are originally configured manually by the network manager. However, the Dynamic Host Configuration Protocol, DHCP, can be used to allocate an IPv4 address to a host that joins the network. In IPv6, DHCP protocol can still be used to allocate an IPv6 address to a host, but a host can also configure itself.

When a host in IPv6 joins a network, it can configure itself using the following process:

1. The host first creates a **link local address** for itself. This is done by taking the 10-bit link local prefix (1111 1110 10), adding 54 zeros, and adding the 64-bit interface identifier, which any host knows how to generate from its interface card. The result is a 128-bit link local address.
2. The host then tests to see if this link local address is unique and not used by other hosts. Since the 64-bit interface identifier is supposed to be unique, the link local

address generated is unique with a high probability. However, to be sure, the host sends a *neighbor solicitation message* (see Chapter 28) and waits for a *neighbor advertisement message.* If any host in the subnet is using this link local address, the process fails and the host cannot autoconfigure itself; it needs to use other means such as DHCP for this purpose.

3. If the uniqueness of the link local address is passed, the host stores this address as its link local address (for private communication), but it still needs a global unicast address. The host then sends a *router solicitation message* (discussed later in the chapter) to a local router. If there is a router running on the network, the host receives a *router advertisement message* that includes the global unicast prefix and the subnet prefix that the host needs to add to its interface identifier to generate its global unicast address. If the router cannot help the host with the configuration, it informs the host in the *router advertisement message* (by setting a flag). The host then needs to use other means for configuration.

Example 22.5

Assume a host with Ethernet address $(\textbf{F5-A9-23-11-9B-E2})_{16}$ has joined the network. What would be its global unicast address if the global unicast prefix of the organization is 3A21:1216:2165 and the subnet identifier is A245:1232?

Solution

The host first creates its interface identifier as **F7A9:23FF:FE11:9BE2** using the Ethernet address read from its card. The host then creates its link local address as:

FE80::F7A9:23FF:FE11:9BE2

Assuming that this address is unique, the host sends a router solicitation message and receives the router advertisement message that announces the combination of global unicast prefix and the subnet identifier as 3A21:1216:2165:A245:1232. The host then appends its interface identifier to this prefix to find and store its global unicast address as:

3A21:1216:2165:A245:1232:F7A9:23FF:FE11:9BE2

22.1.5 Renumbering

To allow sites to change the service provider, **renumbering** of the address prefix (*n*) was built into IPv6 addressing. As we discussed before, each site is given a prefix by the service provider to which it is connected. If the site changes the provider, the address prefix needs to be changed. A router to which the site is connected can advertise a new prefix and let the site use the old prefix for a short time before disabling it. In other words, during the transition period, a site has two prefixes. The main problem in using the renumbering mechanism is the support of the DNS, which needs to propagate the new addressing associated with a domain name. A new protocol for DNS, called Next Generation DNS, is under study to provide support for this mechanism.

22.2 THE IPv6 PROTOCOL

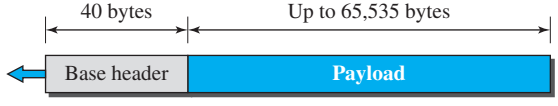
The change of the IPv6 address size requires the change in the IPv4 packet format. The designer of IPv6 decided to implement remedies for other shortcomings now that a change is inevitable. The following shows other changes implemented in the protocol in addition to changing address size and format.

- ***Better header format.*** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- ***New options.*** IPv6 has new options to allow for additional functionalities.
- ***Allowance for extension.*** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- ***Support for resource allocation.*** In IPv6, the type-of-service field has been removed, but two new fields, traffic class and flow label, have been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- ***Support for more security.*** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

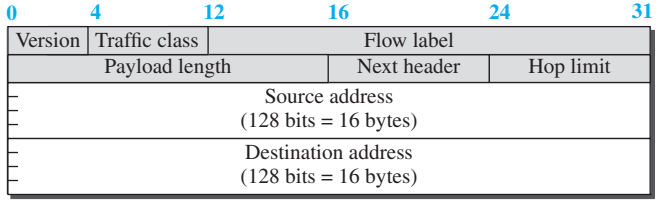
22.2.1 Packet Format

The IPv6 packet is shown in Figure 22.6. Each packet is composed of a base header followed by the payload. The base header occupies 40 bytes, whereas payload can be up to 65,535 bytes of information. The description of fields follows.

Figure 22.6 *IPv6 datagram*



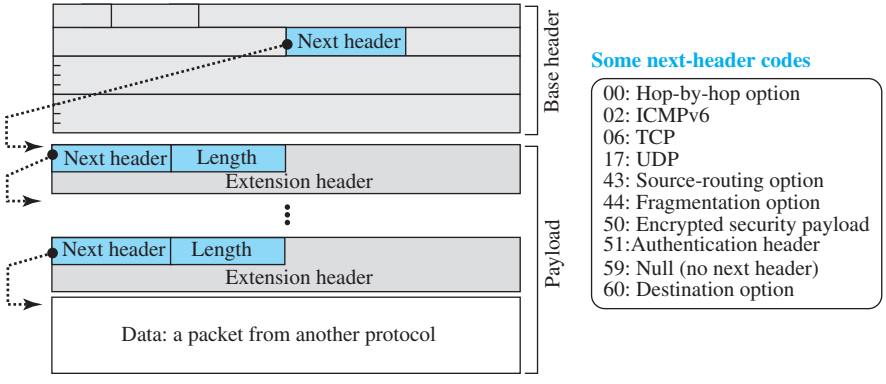
a. IPv6 packet



b. Base header

- ***Version.*** The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.
- ***Traffic class.*** The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements. It replaces the *type-of-service* field in IPv4.
- ***Flow label.*** The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data. We will discuss this field later.
- ***Payload length.*** The 2-byte payload length field defines the length of the IP datagram excluding the header. Note that IPv4 defines two fields related to the length: header length and total length. In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined.
- ***Next header.*** The **next header** is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram. This field is similar to the protocol field in IPv4, but we talk more about it when we discuss the payload.
- ***Hop limit.*** The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- ***Source and destination addresses.*** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.
- ***Payload.*** Compared to IPv4, the payload field in IPv6 has a different format and meaning, as shown in Figure 22.7.

Figure 22.7 *Payload in an IPv6 datagram*



The payload in IPv6 means a combination of zero or more extension headers (options) followed by the data from other protocols (UDP, TCP, and so on). In IPv6, options, which are part of the header in IPv4, are designed as extension headers. The payload can have as many extension headers as required by the situation. Each extension header has two mandatory fields, next header and the length,

followed by information related to the particular option. Note that each next header field value (code) defines the type of the next header (hop-by-hop option, source-routing option, . . .); the last next header field defines the protocol (UDP, TCP, . . .) that is carried by the datagram.

Concept of Flow and Priority in IPv6

The IP protocol was originally designed as a connectionless protocol. However, the tendency is to use the IP protocol as a connection-oriented protocol. The MPLS technology described earlier allows us to encapsulate an IPv4 packet in an MPLS header using a label field. In version 6, the flow label has been directly added to the format of the IPv6 datagram to allow us to use IPv6 as a connection-oriented protocol.

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; the 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 forwarding 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 require other protocols such as Real-Time Transport Protocol (RTP) and Resource Reservation Protocol (RSVP) in addition to IPv6 (see Chapter 28).

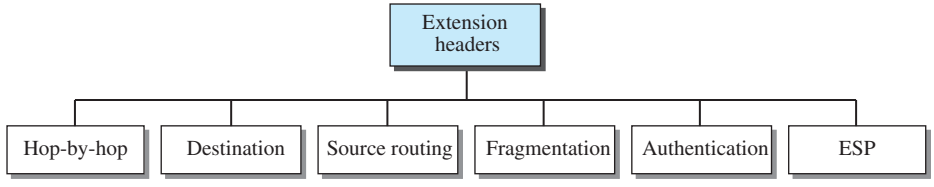
Fragmentation and Reassembly

There are still fragmentation and reassembly of datagrams in the IPv6 protocol, but there is a major difference in this respect. IPv6 datagrams can be fragmented only by the source, not by the routers; the reassembly takes place at the destination. The fragmentation of packets at routers is not allowed to speed up the processing of packets in the router. The fragmentation of a packet in a router needs a lot of processing. The packet needs to be fragmented, all fields related to the fragmentation need to be recalculated. In IPv6, the source can check the size of the packet and make the decision to fragment the packet or not. When a router receives the packet, it can check the size of the packet and drop it if the size is larger than allowed by the MTU of the network ahead. The router then sends a packet-too-big ICMPv6 error message (discussed later) to inform the source.

22.2.2 Extension Header

An IPv6 packet is made of a base header and some extension headers. The length of the base header is fixed at 40 bytes. However, to give more functionality to the IP datagram, the base header can be followed by up to six **extension headers.** Many of these headers are options in IPv4. Six types of extension headers have been defined. These are hop-by-hop option, source routing, fragmentation, authentication, encrypted security payload, and destination option (see Figure 22.8).

Figure 22.8 *Extension header types*



We briefly describe the extension headers in this section, but the complete description is posted at the book website.

Complete descriptions of extension headers are posted on the book website under Extra Materials for Chapter 22.

Hop-by-Hop Option

The *hop-by-hop option* is used when the source needs to pass information to all routers visited by the datagram. For example, perhaps routers must be informed about certain management, debugging, or control functions. Or, if the length of the datagram is more than the usual 65,535 bytes, routers must have this information. So far, only three hop-by-hop options have been defined: Pad1, PadN, and jumbo payload.

- ***Pad1.*** This option is 1 byte long and is designed for alignment purposes. Some options need to start at a specific bit of the 32-bit word. If an option falls short of this requirement by exactly one byte, Pad1 is added.
- ***PadN.*** PadN is similar in concept to Pad1. The difference is that PadN is used when 2 or more bytes are needed for alignment.
- ***Jumbo payload.*** Recall that the length of the payload in the IP datagram can be a maximum of 65,535 bytes. However, if for any reason a longer payload is required, we can use the jumbo payload option to define this longer length.

Destination Option

The **destination option** is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information. The format of the destination option is the same as the hop-by-hop option. So far, only the Pad1 and PadN options have been defined.

Source Routing

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

Fragmentation

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

If the source does not use a Path MTU Discovery technique, it fragments the datagram to a size of 1280 bytes or smaller. This is the minimum size of MTU required for each network connected to the Internet.

Authentication

The **authentication** extension header has a dual purpose: it validates the message sender and ensures the integrity of data. The former is needed so the receiver can be sure that a message is from the genuine sender and not from an imposter. The latter is needed to check that the data is not altered in transition by some hacker. We discuss more about authentication in Chapters 31 and 32.

Encrypted Security Payload

The **encrypted security payload (ESP)** is an extension that provides confidentiality and guards against eavesdropping. Again, we discuss providing more confidentiality for IP packets in Chapter 32.

Comparison of Options between IPv4 and IPv6

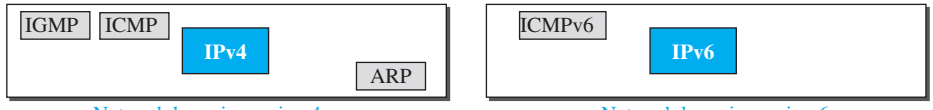
The following shows a quick comparison between the options used in IPv4 and the options used in IPv6 (as extension headers).

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

22.3 THE ICMPv6 PROTOCOL

Another protocol that has been modified in version 6 of the TCP/IP protocol suite is ICMP. This new version, Internet Control Message Protocol version 6 (ICMPv6), follows the same strategy and purposes of version 4. ICMPv6, however, is more complicated than ICMPv4: some protocols that were independent in version 4 are now part of ICMPv6 and some new messages have been added to make it more useful. Figure 22.9 compares the network layer of version 4 to that of version 6. The ICMP, ARP (discussed in Chapter 9), and IGMP protocols in version 4 (Chapter 21) are combined into one single protocol, ICMPv6.

Figure 22.9 *Comparison of network layer in version 4 and version 6*

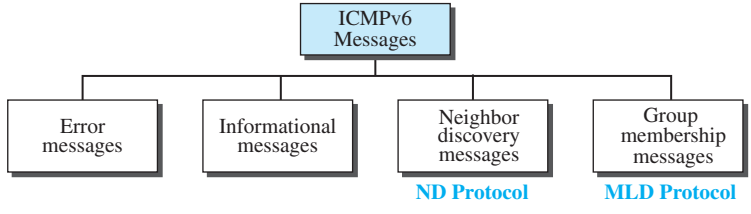


We briefly describe the ICMPv6 in this section, but the complete description is posted at the book website.

Complete descriptions of ICMPv6 messages are posted on the book website under Extra Materials for Chapter 22.

We can divide the messages in ICMPv6 into four groups: error-reporting messages, informational messages, neighbor-discovery messages, and group-membership messages, as shown in Figure 22.10.

Figure 22.10 *Categories of ICMPv6 messages*



22.3.1 Error-Reporting Messages

As we saw in our discussion of version 4, one of the main responsibilities of ICMPv6 is to report errors. Four types of errors are handled: destination unreachable, packet too big, time exceeded, and parameter problems. Note that the source-quenched message, which is used to control congestion in version 4, is eliminated in this version because the

priority and flow label fields in IPv6 are supposed to take care of congestion. The redirection message has moved from the error-reporting category to the neighbor-discovery category, so we discuss it as part of the neighbor-discovery messages.

ICMPv6 forms an error packet, which is then encapsulated in an IPv6 datagram. This is delivered to the original source of the failed datagram.

Destination-Unreachable Message

The concept of the destination unreachable message is the same as described for ICMPv4. When a router cannot forward a datagram or a host cannot deliver the content of the datagram to the upper layer protocol, the router or the host discards the datagram and sends a *destination-unreachable* error message to the source host.

Packet-Too-Big Message

This is a new type of message added to version 6. Since IPv6 does not fragment at the router, if a router receives a datagram that is larger than the maximum transmission unit (MTU) size of the network through which the datagram should pass, two things happen. First, the router discards the datagram. Second, an ICMP error packet—a *packet-too-big message*—is sent to the source.

Time-Exceeded Message

A *time-exceeded* error message is generated in two cases: when the *time to live* value becomes zero and when not all fragments of a datagram have arrived in the time limit. The format of the *time-exceeded* message in version 6 is similar to the one in version 4. The only difference is that the type value has changed to 3.

Parameter-Problem Message

Any ambiguity in the header of the datagram can create serious problems as the datagram travels through the Internet. If a router or the destination host discovers any ambiguous or missing value in any field, it discards the datagram and sends a *parameter-problem* message to the source. The message in ICMPv6 is similar to its version 4 counterpart.

22.3.2 Informational Messages

Two of the ICMPv6 messages can be categorized as informational messages: echo request and echo reply messages. The echo-request and echo-reply messages are designed to check whether two devices in the Internet can communicate with each other. A host or router can send an echo-request message to another host; the receiving computer or router can reply using the echo-reply message.

Echo-Request Message

The idea and format of the echo-request message is the same as the one in version 4.

Echo-Reply Message

The idea and format of the echo-reply message is the same as the one in version 4.

22.3.3 Neighbor-Discovery Messages

Several messages in ICMPv4 have been redefined in ICMPv6 to handle the issue of neighbor discovery. Some new messages have also been added to provide extension. The most important issue is the definition of two new protocols that clearly define the functionality of these group messages: the *Neighbor-Discovery (ND) protocol* and the *Inverse-Neighbor-Discovery (IND) protocol.* These two protocols are used by nodes (hosts or routers) on the same link (network) for three main purposes:

1. Hosts use the ND protocol to find routers in the neighborhood that will forward packets for them.
2. Nodes use the ND protocol to find the link-layer addresses of neighbors (nodes attached to the same network).
3. Nodes use the IND protocol to find the IPv6 addresses of neighbors.

Router-Solicitation Message

The idea behind the *router-solicitation* message is the same as in version 4. A host uses the router-solicitation message to find a router in the network that can forward an IPv6 datagram for the host. The only option that is so far defined for this message is the inclusion of the physical (data-link layer) address of the host to make the response easier for the router.

Router-Advertisement Message

The *router-advertisement* message is sent by a router in response to a router solicitation message.

Neighbor-Solicitation Message

As previously mentioned, the network layer in version 4 contains an independent protocol called Address Resolution Protocol (ARP). In version 6, this protocol is eliminated, and its duties are included in ICMPv6. The neighbor solicitation message has the same duty as the ARP request message. This message is sent when a host or router has a message to send to a neighbor. The sender knows the IP address of the receiver, but needs the data-link address of the receiver. The data-link address is needed for the IP datagram to be encapsulated in a frame. The only option announces the sender data-link address for the convenience of the receiver. The receiver can use the sender data-link address to send a unicast response.

Neighbor-Advertisement Message

The *neighbor-advertisement* message is sent in response to the neighbor-solicitation message.

Redirection Message

The purpose of the redirection message is the same as described for version 4. However, the format of the packet now accommodates the size of the IP address in version 6. Also, an option is added to let the host know the physical address of the target router.

Inverse-Neighbor-Solicitation Message

The *inverse-neighbor-solicitation* message is sent by a node that knows the link-layer address of a neighbor, but not the neighbor's IP address. The message is encapsulated in an IPv6 datagram using an all-node multicast address. The sender must send the following two pieces of information in the option field: its link-layer address and the link-layer address of the target node. The sender can also include its IP address and the MTU value for the link.

Inverse-Neighbor-Advertisement Message

The *inverse-neighbor-advertisement* message is sent in response to the inverse-neighbor-discovery message. The sender of this message must include the link-layer address of the sender and the link-layer address of the target node in the option section.

22.3.4 Group Membership Messages

The management of multicast delivery handling in IPv4 is given to the IGMPv3 protocol. In IPv6, this responsibility is given to the *Multicast Listener Delivery* protocol. MLDv1 is the counterpart to IGMPv2; MLDv2 is the counterpart to IGMPv3. The material discussed in this section is taken from RFC 3810. The idea is the same as we discussed in IGMPv3, but the sizes and formats of the messages have been changed to fit the larger multicast address size in IPv6. Like IGMPv3, MLDv2 has two types of messages: *membership-query message* and *membership-report message*. The first type can be divided into three subtypes: *general*, *group-specific*, and *group-and-source specific*.

Membership-Query Message

A membership-query message is sent by a router to find active group members in the network.

The fields are almost the same as the ones in IGMPv3 except that the size of the multicast address and the source address has been changed from 32 bits to 128 bits. Another noticeable change in the field size is in the *maximum response code* field, in which the size has been changed from 8 bits to 16 bits. We will discuss this field shortly. Also note that the format of the first 8 bytes matches the format for other ICMPv6 packets because MLDv2 is considered to be part of ICMPv6.

Membership-Report Message

The format of the membership report in MLDv2 is exactly the same as the one in IGMPv3 except that the sizes of the fields are changed because of the address size. In particular, the record type is the same as the one defined for IGMPv3 (types 1 to 6).

22.4 TRANSITION FROM IPv4 TO IPv6

Although we have a new version of the IP protocol, how can we make the transition to stop using IPv4 and start using IPv6? The first solution that comes to mind is to define a transition day on which every host or router should stop using the old version and

start using the new version. However, this is not practical; because of the huge number of systems in the Internet, the transition from IPv4 to IPv6 cannot happen suddenly. It will take a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

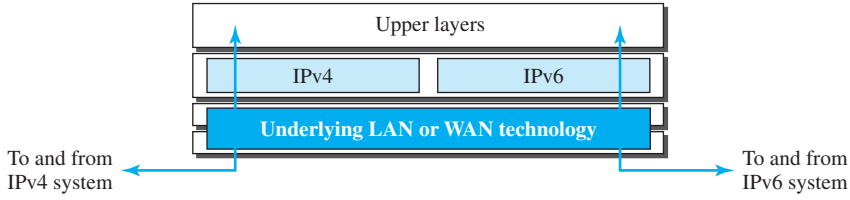
22.4.1 Strategies

Three strategies have been devised for transition: dual stack, tunneling, and header translation. One or all of these three strategies can be implemented during the transition period.

Dual Stack

It is recommended that all hosts, before migrating completely to version 6, have a **dual stack** of protocols during the transition. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6. See Figure 22.11 for the layout of a dual-stack configuration.

Figure 22.11 *Dual stack*



To determine which version to use when sending a packet to a destination, the source host queries the DNS. If the DNS returns an IPv4 address, the source host sends an IPv4 packet. If the DNS returns an IPv6 address, the source host sends an IPv6 packet.

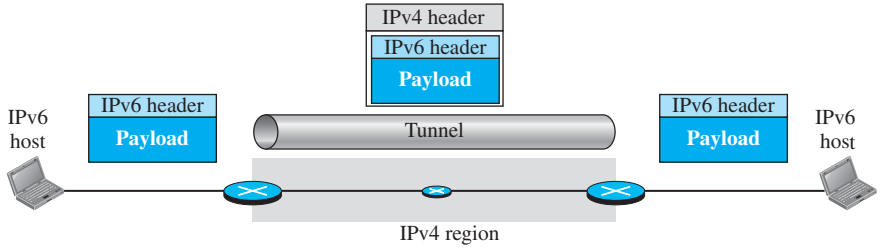
Tunneling

Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet enters a tunnel at one end and emerges at the other end. To make it clear that the IPv4 packet is carrying an IPv6 packet as data, the protocol value is set to 41. Tunneling is shown in Figure 22.12.

Header Translation

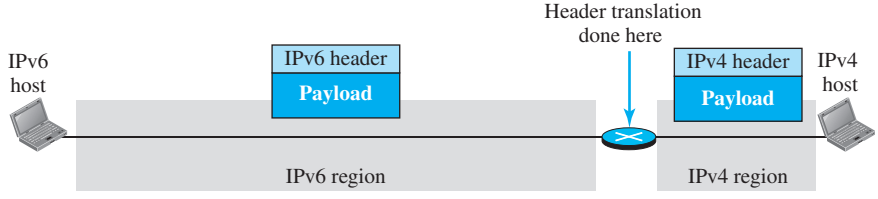
Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4. The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver. In this case, the header format must

Figure 22.12 *Tunneling strategy*



be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header (see Figure 22.13).

Figure 22.13 *Header translation strategy*



22.4.2 Use of IP Addresses

During the transition a host may need to use two addresses, IPv4 and IPv6. When the transition is complete, IPv4 addresses should disappear. The DNS servers (see Chapter 26) need to be ready to map a host name to either address type during the transition, but the IPv4 directory will disappear after all hosts in the world have migrated to IPv6.

22.5 END-CHAPTER MATERIALS

22.5.1 Recommended Reading

Books

Several books give thorough coverage of materials discussed in this chapter. We recommend [Com 06], [Tan 03], [Koz 05], [Ste 95], [GW 04], [Per 00], [Kes 02], [Moy 98], [WZ 01], and [Los 04].

RFCs

IPv6 addressing is discussed in RFCs 2375, 2526, 3513, 3587, 3789, and 4291. IPv6 protocol is discussed in RFCs 2460, 2461, and 2462. ICMPv6 is discussed in RFCs 2461, 2894, 3122, 3810, 4443, and 4620.

22.5.2 Key Terms

anycast address
authentication
autoconfiguration
colon hexadecimal notation
compatible address
destination option
dual stack
encrypted security payload (ESP)
extension header
fragmentation
header translation
Internet Protocol version 6 (IPv6)
IP new generation (IPng)
link local address
link local block
mapped address
next header
Path MTU Discovery technique
renumbering
tunneling
unique local unicast block
zero compression

22.5.3 Summary

IPv6 has a 128-bit address space. Addresses are presented using hexadecimal colon notation with abbreviation methods available. In IPv6, a destination address can belong to one of three categories: unicast, anycast, and multicast. The address space of IPv6 is divided into several blocks of varying size and each block is allocated for a special purpose. The most important block is the one with prefix 001, which is used for global unicast addressing. Two interesting features of IPv6 addressing are autoconfiguration and numbering.

An IPv6 datagram is composed of a base header and a payload. A payload consists of optional extension headers and data from an upper layer. Extension headers add functionality to the IPv6 datagram.

ICMPv6, like ICMPv4, is message-oriented; it uses messages to report errors, get information, probe a neighbor, or manage multicast communication. However, a few other protocols are added to ICMPv6 to define the functionality and interpretation of the messages.

Three strategies used to handle the transition from version 4 to version 6 are dual stack, tunneling, and header translation.

22.6 PRACTICE SET

22.6.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

22.6.2 Questions

Q22-1. Explain the advantages of IPv6 when compared to IPv4.

Q22-2. Explain the use of the flow field in IPv6. What is the potential application of this field?

Q22-3. Distinguish between compatible and mapped addresses and explain their applications.

Q22-4. List three protocols in the IPv4 network layer that are combined into a single protocol in IPv6.

Q22-5. What is the purpose of including the IP header and the first 8 bytes of datagram data in the error-reporting ICMP messages?

Q22-6. If you are assigned an IPv6 address by your ISP for your personal computer at home, what should be the first (leftmost) three bits of this address?

Q22-7. Find the size of the global unicast block from Table 22.1.

Q22-8. Find the size of the special address block from Table 22.1.

Q22-9. Find the size of the unique local unicast block from Table 22.1.

Q22-10. Find the size of the multicast block from Table 22.1.

Q22-11. Explain the benefit of autoconfiguration.

Q22-12. Explain the benefit of renumbering.

Q22-13. Which field in the IPv6 packet is responsible for multiplexing and demultiplexing?

Q22-14. Assume a datagram carries no option. Do we still need a value for the next header field in Figure 22.7?

Q22-15. Which message in version 6 replaces the ARP request message in version 4? Which replaces the ARP reply message?

Q22-16. Which messages in version 6 replace the IGMPv6 messages in version 4?

Q22-17. In which transition strategy do we need to encapsulate IPv6 packets in the IPv4 packets?

Q22-18. In which transition strategy do we need to have both IPv4 and IPv6 in the path?

22.6.3 Problems

P22-1. Compare and contrast the IPv4 header with the IPv6 header. Create a table to compare each field.

P22-2. Show the unabbreviated colon hex notation for the following IPv6 addresses:

- **a.** An address with 64 0s followed by 32 two-bit (01)s.
- **b.** An address with 64 0s followed by 32 two-bit (10)s.
- **c.** An address with 64 two-bit (01)s.
- **d.** An address with 32 four-bit (0111)s.

P22-3. Show abbreviations for the following addresses:

- **a.** 0000:FFFF:FFFF:0000:0000:0000:0000:0000
- **b.** 1234:2346:3456:0000:0000:0000:0000:FFFF
- **c.** 0000:0001:0000:0000:0000:FFFF:1200:1000
- **d.** 0000:0000:0000:0000:FFFF:FFFF:24.123.12.6

P22-4. Decompress the following addresses and show the complete unabbreviated IPv6 address:

a. ::2222 **b.** 1111:: **c.** B:A:CC::1234:A

P22-5. Show the original (unabbreviated) form of the following IPv6 addresses:

a. ::2 **b.** 0:23::0 **c.** 0:A::3

P22-6. What is the corresponding block or subblock associated with each of the following IPv6 addresses, based on Table 22.1:

a. FE80::12**/10** **b.** FD23::**/7** **c.** 32::**/3**

P22-7. An organization is assigned the block 2000:1234:1423/48. What is the CIDR for the blocks in the first and second subnets in this organization?

P22-8. Find the interface identifier if the physical address of the EUI is $(F5\text{-}A9\text{-}23\text{-}AA\text{-}07\text{-}14\text{-}7A\text{-}23)_{16}$ using the format we defined for Ethernet addresses.

P22-9. Find the interface identifier if the Ethernet physical address is $(F5\text{-}A9\text{-}23\text{-}12\text{-}7A\text{-}B2)_{16}$ using the format we defined for Ethernet addresses.

P22-10. An organization is assigned the block 2000:1110:1287/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is $(F5\text{-}A9\text{-}23\text{-}14\text{-}7A\text{-}D2)_{16}$.

P22-11. Using the CIDR notation, show the IPv6 address compatible to the IPv4 address 129.6.12.34.

P22-12. Using the CIDR notation, show the IPv6 address mapped to the IPv4 address 129.6.12.34.

P22-13. Using the CIDR notation, show the IPv6 loopback address.

P22-14. Using the CIDR notation, show the link local address in which the node identifier is 0::123/48.

P22-15. Using the CIDR notation, show the site local address in which the node identifier is 0::123/48.

P22-16. 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.

P22-17. 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.

P22-18. Which ICMP messages contain part of the IP datagram? Why is this needed?

P22-19. Make a table to compare and contrast error-reporting messages in ICMPv6 with error-reporting messages in ICMPv4.

P22-20. Make a table to compare and contrast informational messages in ICMPv6 with informational messages in ICMPv4.

P22-21. Make a table to compare and contrast neighbor-discovery messages in ICMPv6 with the corresponding messages in version 4.

P22-22. Make a table to compare and contrast inverse neighbor-discovery messages in ICMPv6 with the corresponding messages in version 4.

P22-23. Make a table to compare and contrast group-membership messages in ICMPv6 with the corresponding messages in version 4.