Computer Networking with TCP/IP

Module 3

Network Layer Protocols

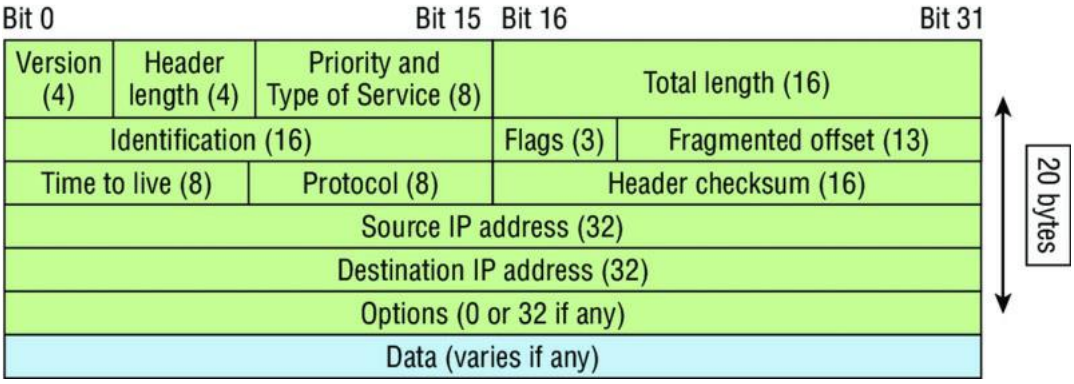
The Three important Network layer protocols are:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)

Internet Protocol (IP)

Internet Protocol (IP) essentially is the Internet layer. The other protocols found here merely exist to support it.

IP header



Bit 0		Bit 15	Bit 16	Bit 31
Version (4)	Header length (4)	Priority and Type of Service (8)	Total length (16)	
Identification (16)			Flags (3)	Fragmented offset (13)
Time to live (8)		Protocol (8)	Header checksum (16)	
Source IP address (32)				
Destination IP address (32)				
Options (0 or 32 if any)				
Data (varies if any)				

IPv4 Addresses

The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address. **An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet**; an IP address is the address of the interface.

IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. However, if a device has two connections to the Internet, via two networks, it has two IPv4 addresses. **The IPv4 addresses are universal** in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

A protocol like IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).

Notation

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

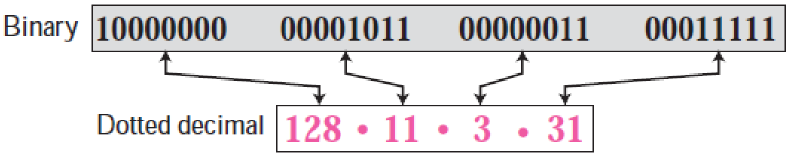
Binary Notation: Base 2

In binary notation, an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces is usually inserted between each octet (8 bits).

01110101 10010101 00011101 11101010

Dotted-Decimal Notation: Base 256

To make the IPv4 address more compact and easier to read, an IPv4 address is usually written in decimal form with a decimal point (dot) separating the bytes. This format is referred to as dotted-decimal notation.



Hexadecimal Notation: Base 16

We sometimes see an IPv4 address in hexadecimal notation. Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits. This notation is often used in network programming.

Range of Addresses

We often need to deal with a range of addresses instead of one single address. We sometimes need to find the number of addresses in a range if the first and last address is given. Other times, we need to find the last address if the first address and the number of addresses in the range are given. In this case, we can perform subtraction or addition operations in the corresponding base (2, 256, or 16). Alternatively, we can covert the addresses to decimal values (base 10) and perform operations in this base.

Example 5.5

Find the number of addresses in a range if the first address is 146.102.29.0 and the last address is 146.102.32.255.

Solution

We can subtract the first address from the last address in base 256 (see Appendix B). The result is 0.0.3.255 in this base. To find the number of addresses in the range (in decimal), we convert this number to base 10 and add 1 to the result.

The Hierarchical IP Addressing Scheme

The 32-bit IP address is a structured or hierarchical address, as opposed to a flat or nonhierarchical address. Although type of addressing scheme could have been used, hierarchical addressing was chosen for a good reason. The advantage of this scheme is that it can handle many addresses. The disadvantage of the flat addressing scheme, and the reason it's not used for IP addressing, relates to routing. If every address were unique, all routers on the Internet would need to store the address of each machine on the Internet. This would make efficient routing impossible, even if only a fraction of the possible addresses were used!

The solution to this problem is to use a two- or three-level hierarchical addressing scheme that is structured by network and host or by network, subnet, and host.

Network Addressing

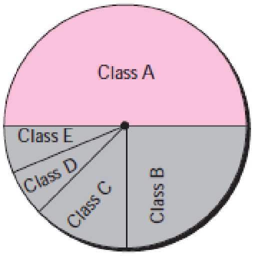
The network address (which can also be called the network number) uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. For example, in the IP address 172.16.30.56, 172.16 is the network address.

Classful and Classless Addressing

IP addresses, when started a few decades ago, used the concept of classes. This architecture is called **classful addressing**. In the mid-1990s, a new architecture, called **classless addressing**,

Classful Addressing

In classful addressing, the IP address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the whole address space. Figure 5.5 shows the class occupation of the address space.



Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

	8 bits	8 bits	8 bits	8 bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

Recognizing Classes

	Octet 1	Octet 2	Octet 3	Octet 4
Class A	0........			
Class B	10......			
Class C	110.....			
Class D	1110....			
Class E	1111....			

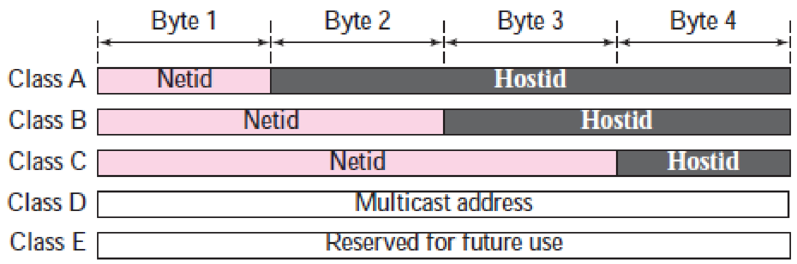
Binary notation

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–299			
Class E	240–255			

Dotted-decimal notation

Netid and Hostid

In classful addressing, an IP address in classes A, B, and C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address.



Class A

Since only 1 byte in class A defines the netid and the leftmost bit should be 0, the next 7 bits can be changed to find the number of blocks in this class. Therefore, class A is divided into 27 = 128 blocks that can be assigned to 128 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 16,777,216 addresses.

Class B

Since 2 bytes in class B define the class and the two leftmost bits should be 10 (fixed), the next 14 bits can be changed to find the number of blocks in this class. Therefore, class B is divided into 214 = 16,384 blocks that can be assigned to 16,384 organizations (the number is less because some blocks were reserved as special blocks). However, each block in this class contains 65,536 addresses.

Class	HOB	NET ID Bits	Host ID Bits	No of Networks	Host Per Network	Start Address	End Address
Class A	0	8	24	2^{7}=128	2^{24}=16,777,216	0.0.0.0	127.255.255.255
Class B	10	16	16	2^{14}=16,384	2^{16}=65,536	128.0.0.0	191.255.255.255
Class C	110	24	8	2^{21}=2,097,152	2^{8}=256	192.0.0.0	223.255.255.255
Class D	1110	-	-	-	-	224.0.0.0	239.255.255.255
Class E	1111	-	-	-	-	240.0.0.0	255.255.255.255

Class C

Since 3 bytes in class C define the class and the three leftmost bits should be 110 (fixed), the next 21 bits can be changed to find the number of blocks in this class. Therefore, class C is divided into 221 = 2,097,152 blocks, in which each block contains 256 addresses, that can be assigned to 2,097,152 organizations

Class D

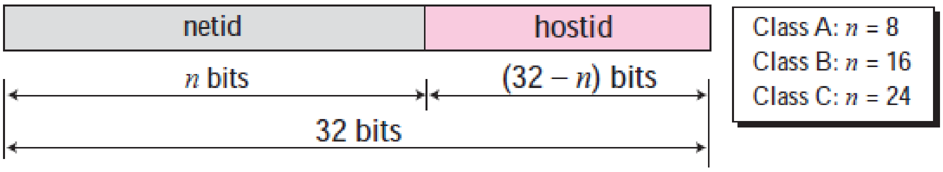
There is just one block of class D addresses. It is designed for multicasting, as we will see in a later section. Each address in this class is used to define one group of hosts on the Internet.

Class E

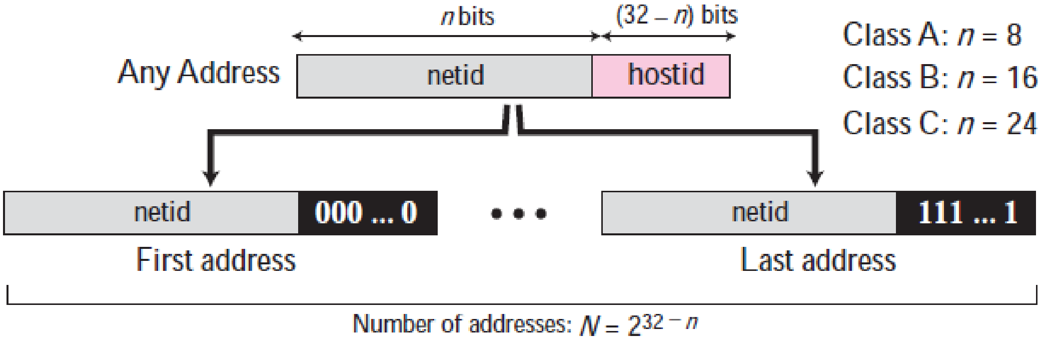
There is just one block of class E addresses. It was designed for use as reserved addresses.

Two-Level Addressing

The whole purpose of IPv4 addressing is to define a destination for an Internet packet (at the network layer). If n bits in the class defines the net, then 32 − n bits define the host. However, the value of n depends on the class the block belongs to. The value of n can be 8, 16 or 24 corresponding to classes A, B, and C respectively.



1. The number of addresses in the block, N, can be found using N = 232−n.
2. To find the first address, we keep the n leftmost bits and set the (32− n) rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the (32 − n) rightmost bits all to 1s.

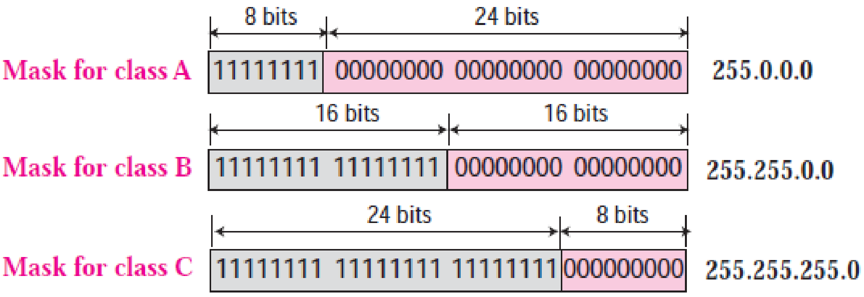


Network Address

The above three examples show that, given any address, we can find all information about the block. The first address, network address, is particularly important because it is used in routing a packet to its destination network.

Network Mask

The methods we described previously for extracting the network address are mostly used to show the concept. The routers in the Internet normally use an algorithm to extract the network address from the destination address of a packet. To do this, we need a network mask.



Three-Level Addressing

To reach a host on the Internet, we must first reach the network and then the host. It soon became clear that we need more than two hierarchical levels, for two reasons. **First,** an organization that was granted a block in class A or B needed to divide its large network into several subnetworks for better security and management. **Second**, since the blocks in class A and B were almost depleted and the blocks in class C were smaller than the needs of most organizations, an organization that has been granted a block in class A or B could divide the block into smaller subblocks and share them with other organizations.

Subnetting

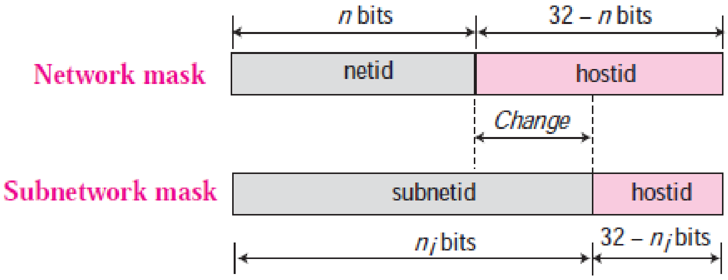
The idea of splitting a block to smaller blocks is referred to as subnetting. In subnetting, a network is divided into several smaller subnetworks (subnets) with each subnetwork having its own subnetwork address.

Subnet Address

When a network is subnetted, the first address in the subnet is the identifier of the subnet and is used by the router to route the packets destined for that subnetwork.

Subnet Mask

The network mask is used when a network is not subnetted. When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork. A subnetwork has subnetid and hosted.

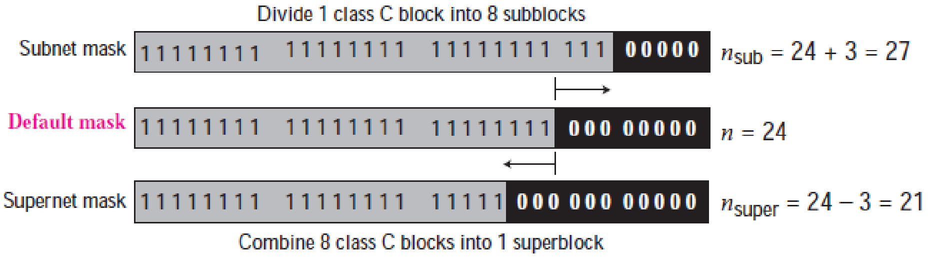


Supernetting

Several sub networks are combined to create a supernetwork.

Supernet Mask

A supernet mask is the reverse of a subnet mask.



CLASSLESS ADDRESSING

Subnetting and supernetting in classful addressing did not really solve the address depletion problem and made the distribution of addresses and the routing process more difficult.

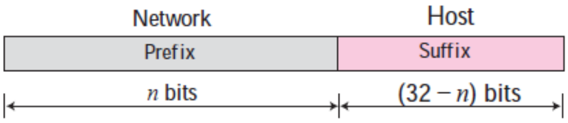
The short-term solution still uses IPv4 addresses, but it is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion.

Variable-Length Blocks

In classless addressing, the whole address space is divided into variable length blocks. Theoretically, we can have a block of 20, 21, 22, . . . , 232 addresses.

Two-Level Addressing

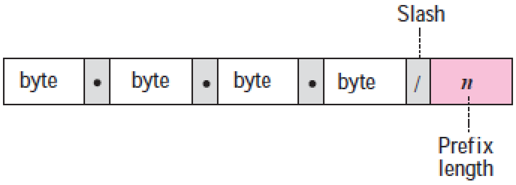
In classful addressing, two-level addressing was provided by dividing an address into netid and hostid. In classless addressing, the prefix defines the network, and the suffix defines the host.



In classless addressing, the value of n is referred to as prefix length; the value of 32 − n is referred to as suffix length.

Slash Notation (CIDR – Classless Inter Domain Routing)

The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block.



The slash notation is formally referred to as classless interdomain routing or CIDR (pronounced cider) notation. The slash notation is formally referred to as classless interdomain routing or CIDR (pronounced cider) notation.

Network Mask

The idea of network mask in classless addressing is the same as the one in classful addressing.

Extracting Block Information

An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address.

- The number of addresses in the block can be found as: $N = 2^{32-n}$
- The first address (network address) in the block can be found by ANDing the address with the network mask: First address = (any address) AND (network mask)
- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

Block Allocation

The next issue in classless addressing is block allocation. How are the blocks allocated? The ultimate responsibility of block allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN). For the proper operation of the CIDR, three restrictions need to be applied to the allocated block.

- The number of requested addresses, N, needs to be a power of 2. This is needed to provide an integer value for the prefix length, n. The number of addresses can be 1, 2, 4, 8, 16, and so on.
- The value of prefix length can be found from the number of addresses in the block. Since $N = 2^{32-n}$, then $n = \log_2 (2^{32}/N) = 32 - \log_2 N$. That is the reason why N needs to be a power of 2.
- The requested block needs to be allocated where there are a contiguous number of unallocated addresses in the address space. However, there is a restriction on choosing the beginning addresses of the block. The beginning address needs to be divisible by the number

of addresses in the block. To see this restriction, we can show that the beginning address can be calculated as X × 2^{n-32} in which X is the decimal value of the prefix. In other words, the beginning address is X × N.

Example

An ISP has requested a block of 1000 addresses. The following block is granted.

- Since 1000 is not a power of 2, 1024 addresses are granted (1024 = 210).
- The prefix length for the block is calculated as n = 32 − $\log_2$1024 = 22.
- The beginning address is chosen as 18.14.12.0 (which is divisible by 1024).

The granted block is 18.14.12.0/22. The first address is 18.14.12.0/22 and the last address is 18.14.15.255/22.

Finding Information about Each Subnetwork

After designing the subnetworks, the information about each subnetwork, such as first and last address, can be found using the process we described to find the information about each network in the Internet.

Example

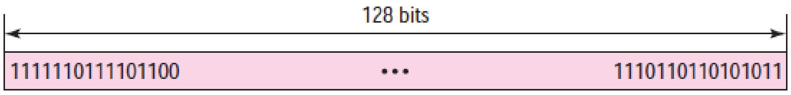
An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

Address Aggregation

One of the advantages of CIDR architecture is address aggregation. ICANN assigns a large block of addresses to an ISP. Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers; many blocks of addresses are aggregated in one block and granted to one ISP.

IPv6 Addresses

An IPv6 address is 128 bits or 16 bytes (octet) long. The address length in IPv6 is four times of the length address in IPv4.



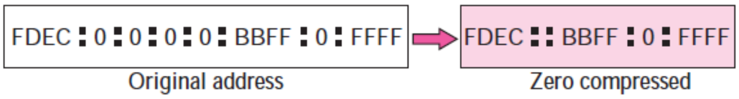
Colon Hexadecimal Notation

To make addresses more readable, IPv6 specifies colon hexadecimal notation (or colon hex for short). In this notation, 128 bits are divided into eight sections, each 2 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.

FDEC : BA98 : 7654 : 3210 : ADBF : BBFF : 2922 : FFFF

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.

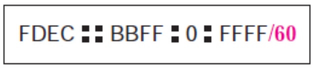
Zero compression, can be applied to colon hex notation if there are consecutive sections consisting of zeros only. We can remove all the zeros altogether and replace them with a double semicolon.



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 compressed.

CIDR Notation

IPv6 allows classless addressing and CIDR notation. For example, Figure 26.4 shows how we can define a prefix of 60 bits using CIDR.



FDEC::BBFF:0:FFFF/60

Example

Show abbreviations for the following addresses:

a. 0000:0000:FFFF:0000:0000:0000:0000:0000

b. 1234:2346:0000:0000:0000:0000:0000:1111

c. 0000:0001:0000:0000:0000:0000:1200:1000

d. 0000:0000:0000:0000:0000:FFFF:24.123.12.6

Solution

a. 0:0:FFFF::

b. 1234:2346::1111

c. 0:1::1200:1000

d. ::FFFF:24.123.12.6

Address Space

The address space of IPv6 contains 2128 addresses as shown below. This address space is 296 times of the IPv4 address—definitely no address depletion.

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.

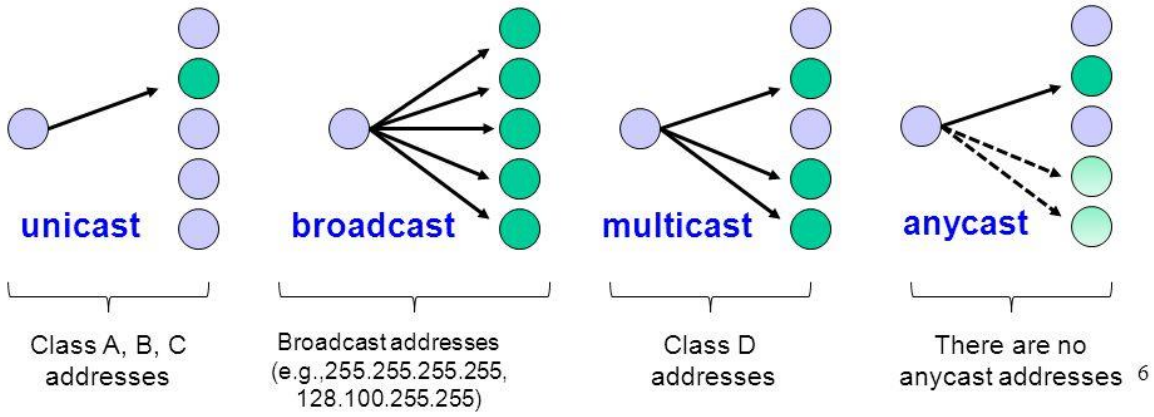
Multicast Address

A multicast address also defines a group of computers. However, there is a difference between anycasting and multicasting. In multicasting, each member of the group receives a copy.

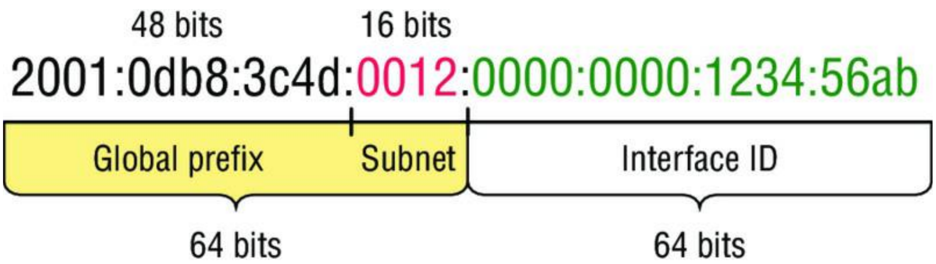
Broadcasting and Multicasting

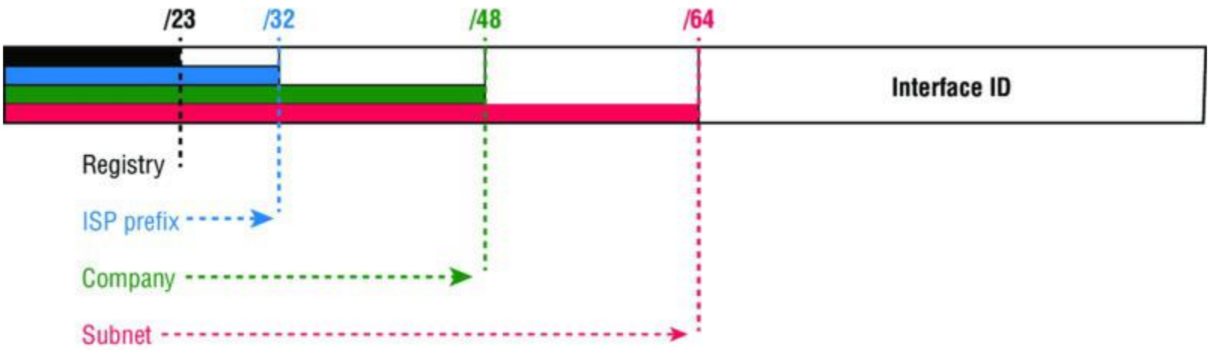
It is interesting that IPv6 does not define broadcasting,

- Supported by IPv4
  - one-to-one (unicast)
  - one-to-all (broadcast)
  - one-to-many (multicast)
- Not supported by IPv4:
  - one-to-any (anycast)


unicast
broadcast
multicast
anycast
Class A, B, C addresses
Broadcast addresses (e.g.,255.255.255.255, 128.100.255.255)
Class D addresses
There are no anycast addresses
6

ADDRESS SPACE ALLOCATION



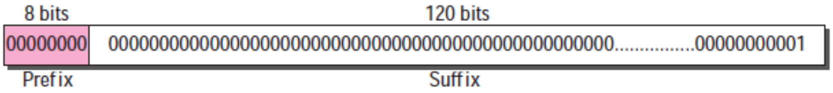


IPv4 Compatible Addresses

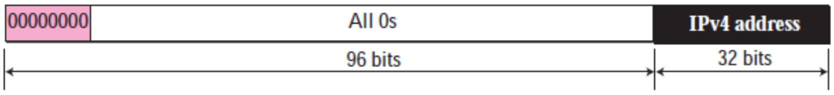
Addresses that use the prefix (00000000) are reserved, but part of it is used to define some IPv4 compatible addresses. This block occupies 1/256 of the total address space, which means that there are 2120 addresses in this block. In CIDR notation, this block can be defined as 0000::/8.

Unspecified Address The unspecified address is a subblock containing only one single address, which is defined by letting all suffix bits to 0s. In other words, the entire address consists of zeros. CIDR notation for this one-address subblock is ::/128.

Loopback Address This is an address used by a host to test itself without going into the network. The loopback address as shown below consists of the prefix 00000000 followed by 119 0s and one 1. The CIDR notation for this one-address single block is ::1/128.



Embedded IPv4 Addresses As we will see in Chapter 27, 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. The CIDR notation for this subblock is ::/96.

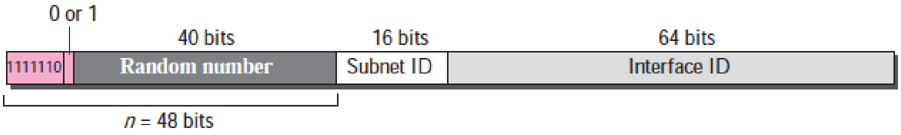


Global Unicast Block

This is the main block used for unicast communication between hosts in the Internet.

Unique Local Unicast Block

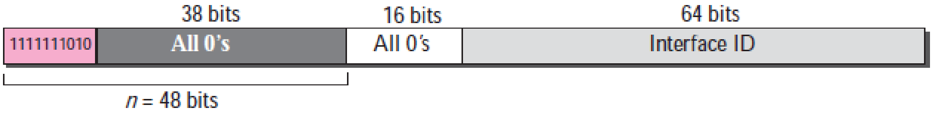
IPv6 uses two large blocks for private addressing: one at the site level and one at the link level.



A subblock in a unique local unicast block can be privately created and used by a site.

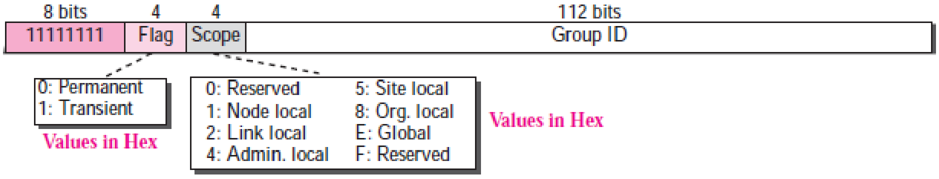
Link Local Block

The second block designed for private addresses is link local block. A subblock in this block can be used as a private address in a network. This type of address has the block identifier 1111 1110 10. The next 54 bits are set to zero. The last 64 bits can be changed to define the interface for each computer.



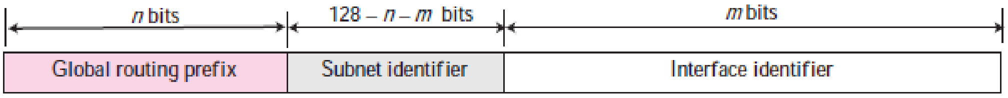
Multicast Block

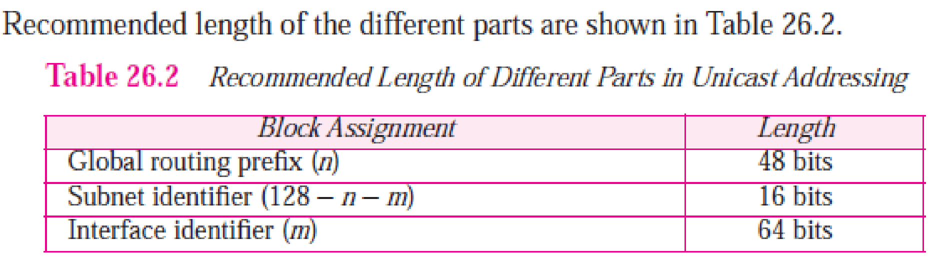
In IPv6 a large block of addresses are assigned for multicasting. All these addresses use the prefix 11111111.



GLOBAL UNICAST ADDRESSES

This block in the address space that is used for unicast (one-to-one) communication between two hosts in the Internet is called global unicast address block. CIDR notation for the block is 2000::/3.





Recommended length of the different parts are shown in Table 26.2.

Table 26.2 *Recommended Length of Different Parts in Unicast Addressing*

Block Assignment	*Length*
Global routing prefix (*n*)	48 bits
Subnet identifier (128 – *n* – *m*)	16 bits
Interface identifier (*m*)	64 bits

Global Routing Prefix

The first 48 bits of a global unicast address are called global routing prefix.

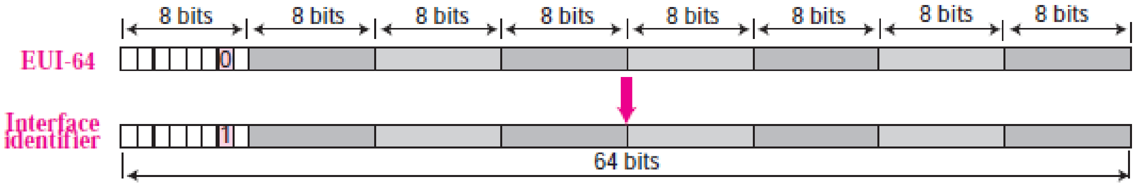
Subnet Identifier

The next 16 bits defines a subnet in an organization. This means that an organization can have up to 216 = 6553 subnets, which is more than enough.

Interface Identifier

The last 64 bits define the interface identifier. Two common physical addressing scheme can be considered for this purpose: the 64-bit extended unique identifier (EUI-64) defined by IEEE and the 48-bit physical 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.



The additional 16 bits are defined as 15 ones followed by one zero, or FFFE

Method –

1. Split the 6 Byte 48Bit (12Hex Digit) MAC address in two halves
2. Insert FFFE in between the two making the interface id as 16 Hex digits (64Bits)
3. Invert the 7th Bit of the interface id.

Example

Find the interface identifier if the Ethernet physical address is (F5-A9-23-14-7A-D2)16 using the format we defined for Ethernet addresses.

Solution

We only need to change the seventh bit of the first octet from 0 to 1, insert two octet FFFE16 and change the format to colon hex notation. The result is F7A9:23FF:FE14:7AD2 in colon hex.

0 <-> 2   4 <-> 6   8 <-> A   C <-> E | 1 <-> 3   5 <-> 7   9<-> B   D <-> F

Why IPV6

❑ **Larger address space**. An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296 times) 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 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.

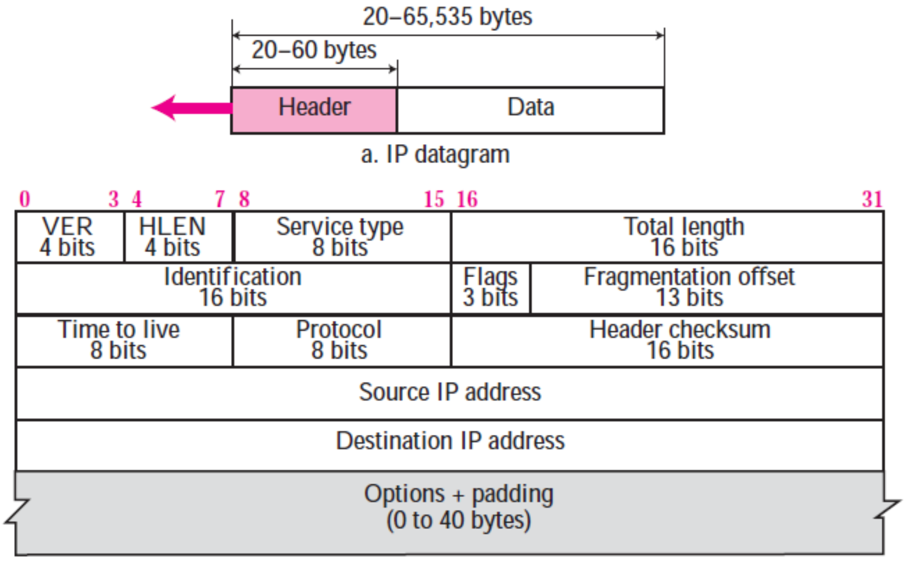
❑ **Support for more security**. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Internet Protocol (IPv4)

IP is an unreliable and connectionless datagram protocol—a best-effort delivery service. The term best-effort means that IP packets can be corrupted, lost, arrive out of order, or delayed and may create congestion for the network.

DATAGRAMS

Packets in the network (internet) layer are called datagrams. Figure 7.2 shows the IP datagram format. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length.



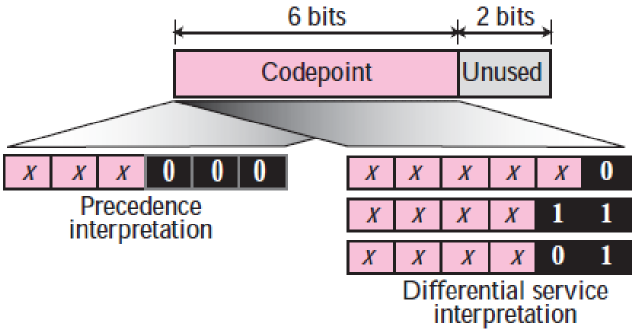
a. IP datagram

b. Header format

Version (VER). This 4-bit field defines the version of the IP protocol.

Header length (HLEN). This 4-bit field defines the total length of the datagram header in 4-byte words.

Service type. In the original design of IP header, this field was referred to as type of service (TOS).



Total length. This is a 16-bit field that defines the total length (header plus data) of the IP datagram in bytes.

Identification. This field is used in fragmentation (discussed in the next section).

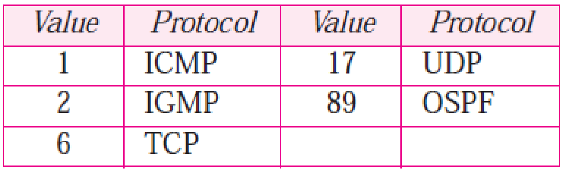
Flags. This field is used in fragmentation (discussed in the next section).

Fragmentation offset. This field is used in fragmentation (discussed in the next section).

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

Protocol. This 8-bit field defines the higher-level protocol that uses the services of the IP layer.

Table 7.2 *Protocols*



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

Checksum. It is used to detect corruption in the header of IPv4 packets

Source address. This 32-bit field defines the IP address of the source.

Destination address. This 32-bit field defines the IP address of the destination.

Example 1

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

01000010

The receiver discards the packet. Why?

Solution

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

Example 2

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

Solution

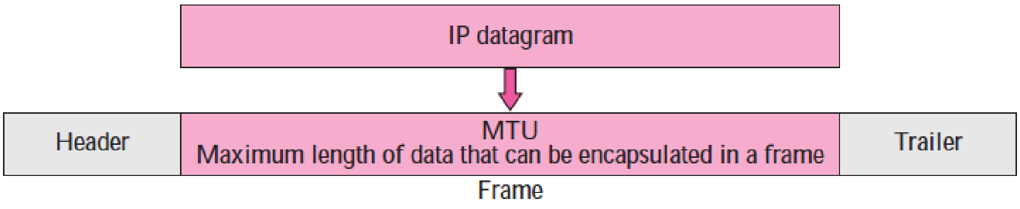
The HLEN value is 8, which means the total number of bytes in the header is 8 × 4 or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

FRAGMENTATION

A datagram can travel through different networks. Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.

Maximum Transfer Unit (MTU)

When a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size, which is defined by the restrictions imposed by the hardware and software used in the network.



What is Fragmentation

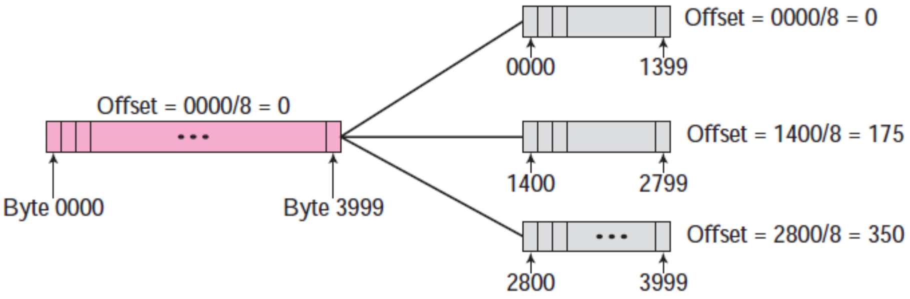
To make the IP protocol independent of the physical network, the designers decided to make the maximum length of the IP datagram equal to 65,535 bytes. This makes transmission more efficient if we use a protocol with an MTU of this size. However, for other physical networks, we must divide the datagram to make it possible to pass through these networks. This is called fragmentation.

Identification 16 bits	Flag 3 bits	Fragmentation offset – 13 bits

Identification. This 16-bit field identifies a datagram originating from the source host.

Flags. This is a three-bit field. **The first bit** is reserved (not used). **The second bit** is called the do not fragment bit. If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram through any available physical network, it discards the datagram and sends an ICMP error message to the source host. If its value is 0, the datagram can be fragmented if necessary. **The third bit** is called the more fragment bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment.

Fragmentation offset. This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure 7.8 shows a datagram with a data size of 4000 bytes fragmented into three fragments. The bytes in the original datagram are numbered 0 to 3999. The first fragment carries bytes 0 to 1399. The offset for this datagram is 0/8 = 0. The second fragment carries bytes 1400 to 2799; the offset value for this fragment is 1400/8 = 175. Finally, the third fragment carries bytes 2800 to 3999. The offset value for this fragment is 2800/8 = 350.



Method to find the details

a. The first fragment has an offset field value of zero.

b. Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.

c. Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.

d. Continue the process. The last fragment has a more bit value of 0.

Example

A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?

Solution

To find the number of the first byte, we multiply the offset value by 8. This means that the first byte number is 800. We cannot determine the number of the last byte unless we know the length of the data.

Example 2.

A datagram of size 3000 Bytes (20 Byte + 2980 IP payload) has reached a router and must be forwarded to the next the destination with MTU of 500 Bytes.

a. How many Fragments will be generated?
b. What is the MF, Offset, Total Length? (MF means more fragment bit)

Assume that the DF flag was not set

Assume that no optional fields of the IP header are in use (i.e. IP header is 20 bytes)

The original datagram was 3000 bytes, subtracting 20 bytes for header, that leaves 2980 bytes of data.

Assume the ID of the original packet is 'x'

With an MTU of 500 bytes, 500 - 20 = 480 bytes of data may be transmitted in each packet

Therefore, ceiling (2980 / 480) = 7 packets are needed to carry the data.

The packets will have the following characteristics (NOTE: offset is measured in 8 byte blocks.

Packet 1: ID=x, Total_len=500, MF=1, Frag_offset=0

Packet 2: ID=x, Total_len=500, MF=1, Frag_offset=60

Packet 3: ID=x, Total_len=500, MF=1, Frag_offset=120

Packet 4: ID=x, Total_len=500, MF=1, Frag_offset=180

Packet 5: ID=x, Total_len=500, MF=1, Frag_offset=240

Packet 6: ID=x, Total_len=500, MF=1, Frag_offset=300

Packet 7: ID=x, Total_len=120, MF=0, Frag_offset=360

Example

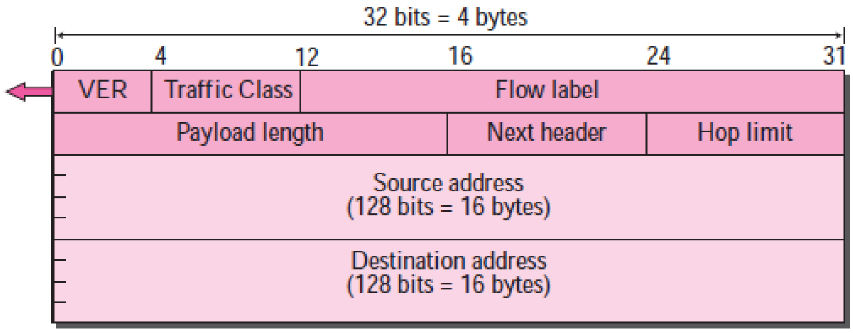
A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte?

Solution

The first byte number is 100 × 8 = 800. The total length is 100 bytes and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte number is 800, the last byte number must be 879.

IPv6- Packet format.

Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



Base Header

Base Header is having 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.

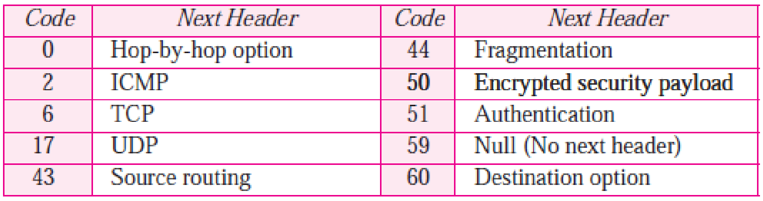
❑ Traffic Class. This 8-bit field is used to distinguish different payloads with different delivery requirements. It replaces the service class field in IPv4.

❑ 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 base header.

❑ Next header. The next header is an 8-bit field defining the header that follows the base header in the datagram.

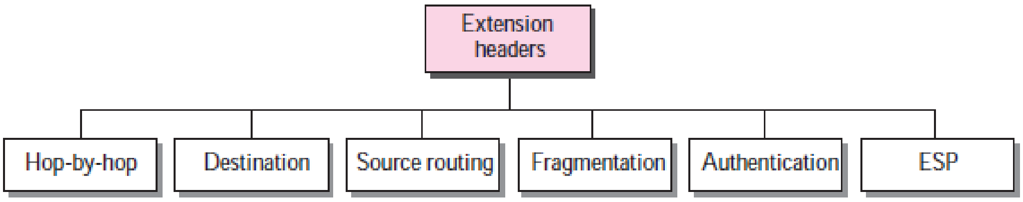
Table 27.1 *Next Header Codes*



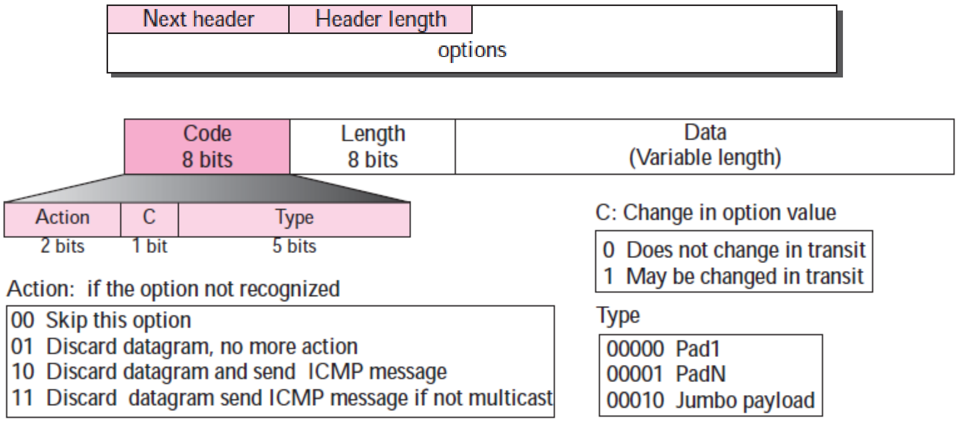
Code	*Next Header*	*Code*	*Next Header*
0	Hop-by-hop option	44	Fragmentation
2	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option

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.



Hop-by-hop option header format


Next header
Header length
options
Code 8 bits
Length 8 bits
Data (Variable length)
Action
C
Type
2 bits
1 bit
5 bits
C: Change in option value
0 Does not change in transit
1 May be changed in transit
Action: if the option not recognized
00 Skip this option
01 Discard datagram, no more action
10 Discard datagram and send ICMP message
11 Discard datagram send ICMP message if not multicast
Type
00000 Pad1
00001 PadN
00010 Jumbo payload

Comparison between IPv4 and IPv6 Headers

The following shows the comparison between IPv4 and IPv6 headers.

❑ The header length field is eliminated in IPv6 because the length of the header is fixed in this version.

❑ The service type field is eliminated in IPv6. The traffic class and flow label fields together take over the function of the service type field.

❑ The total length field is eliminated in IPv6 and replaced by the payload length field.

❑ The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.

❑ The TTL field is called hop limit in IPv6.

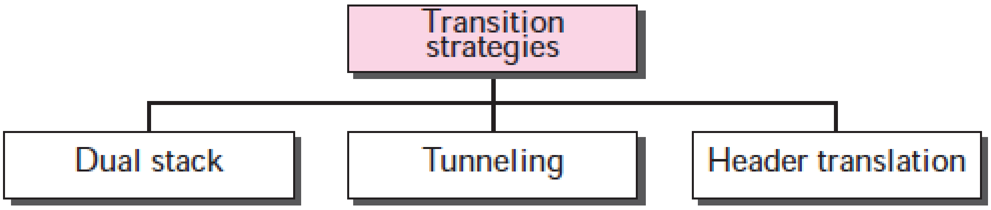
❑ The protocol field is replaced by the next header field.

❑ The header checksum is eliminated because the checksum is provided by upper layer protocols; it is therefore not needed at this level.

❑ The option fields in IPv4 are implemented as extension headers in IPv6.

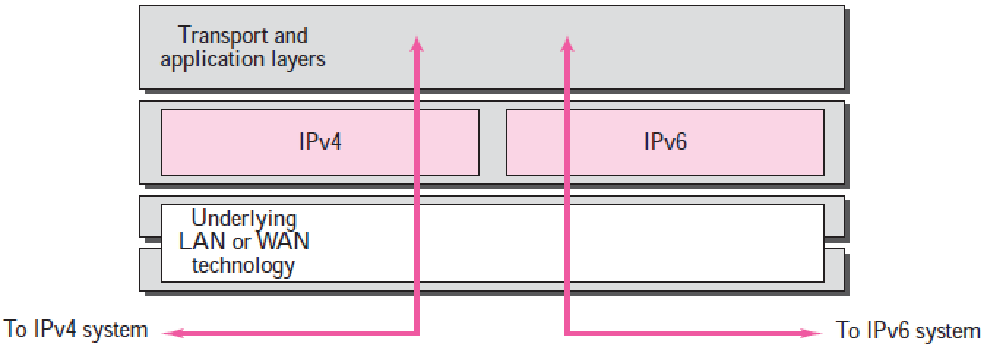
TRANSITION FROM IPv4 TO IPv6

It will take a considerable amount of time before every system on the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems. Three strategies have been devised by the IETF to help the transition.



Dual Stack

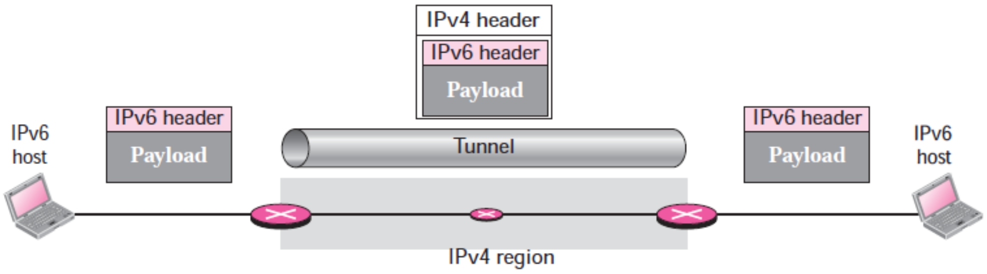
It is recommended that all hosts, before migrating completely to version 6, have a dual stack of protocols. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.


Transport and application layers
IPv4
IPv6
Underlying LAN or WAN technology
To IPv4 system
To IPv6 system

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

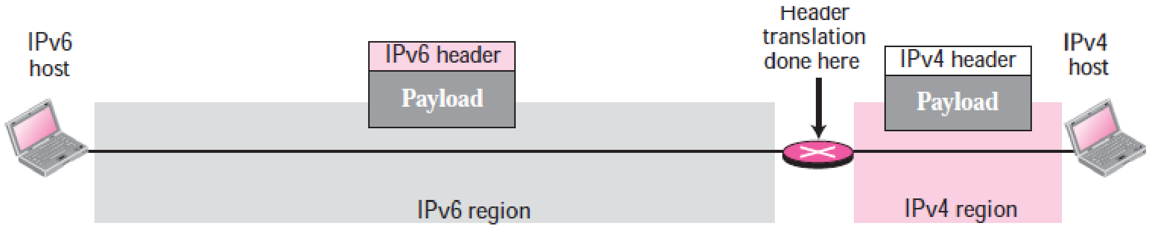
Tunneling is a strategy used when two computers using IPv6 want to communicate with each other, and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So, the IPv6 packet is encapsulated in an IPv4 packet when it enters the region,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



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 be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header.



❑ The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.

❑ The value of the IPv6 priority field is discarded.

❑ The type of service field in IPv4 is set to zero.

❑ The checksum for IPv4 is calculated and inserted in the corresponding field.

❑ The IPv6 flow label is ignored.

❑ Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.

❑ The length of IPv4 header is calculated and inserted into the corresponding field.

❑ The total length of the IPv4 packet is calculated and inserted in the corresponding field.

Address Mapping Protocols:

Before the IP protocol can deliver a packet from a source host to the destination host, it needs to know how to deliver it to the next hop first. An IP packet can consult its routing table, to find the IP address of the next hop. But since IP uses the services of the data link layer, it needs to know the physical address of the next hop. This can be done using a protocol, called Address Resolution Protocol (ARP).

ADDRESS MAPPING

An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host.

The delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done using either static or dynamic mapping.

Static Mapping

Static mapping means creating a table that associates a logical address with a physical address. This table is stored in each machine on the network. Each machine that knows. This has some limitations because physical addresses may change in the

following ways:

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

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

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

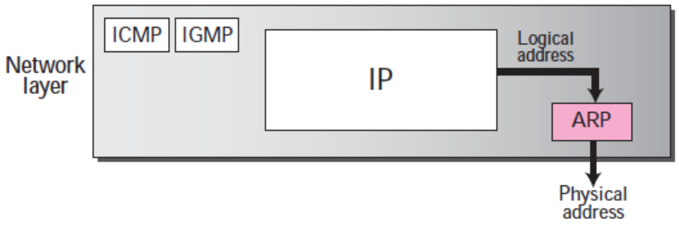
To implement these changes, a static mapping table must be updated periodically. This overhead could affect network performance.

Dynamic Mapping

In dynamic mapping, each time a machine knows the logical address of another machine, it can use a protocol to find the physical address. Two protocols have been designed to perform dynamic mapping: Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP). ARP maps a logical address to a physical address; RARP maps a physical address to a logical address.

THE ARP PROTOCOL

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. A mapping corresponds a logical address to a physical address.



Anytime a host, or a router, needs to find the physical address of another host or router on its network, it sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

ARP defines a protocol that includes the ARP Request, which is a message that makes the simple request “if this is your IP address, please reply with your MAC address.” ARP also defines the ARP Reply message, which indeed lists both the original IP address and the matching MAC address.

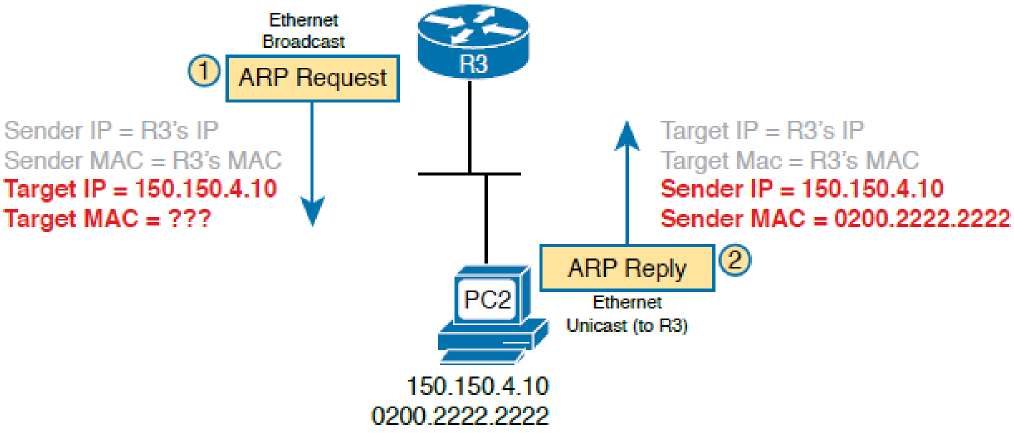
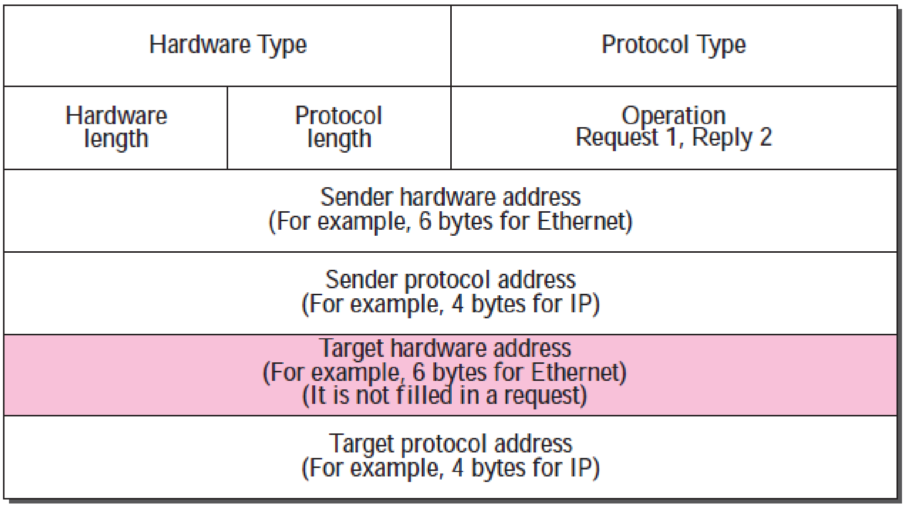


Figure 3-15 *Sample ARP Process*

ARP packet

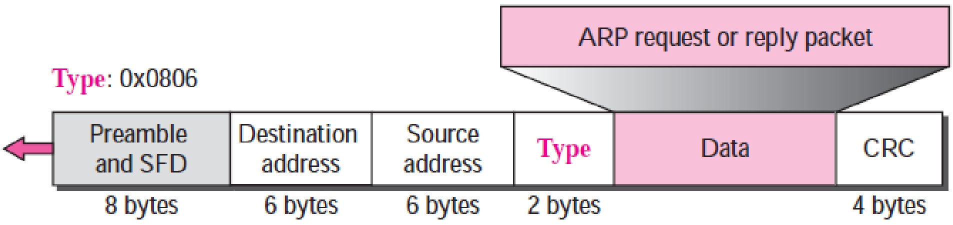


<table>
<tr><td colspan="2">Hardware Type</td><td>Protocol Type</td></tr>
<tr><td>Hardware length</td><td>Protocol length</td><td>Operation
Request 1, Reply 2</td></tr>
<tr><td colspan="3">Sender hardware address
(For example, 6 bytes for Ethernet)</td></tr>
<tr><td colspan="3">Sender protocol address
(For example, 4 bytes for IP)</td></tr>
<tr><td colspan="3">Target hardware address
(For example, 6 bytes for Ethernet)
(It is not filled in a request)</td></tr>
<tr><td colspan="3">Target protocol address
(For example, 4 bytes for IP)</td></tr>
</table>

- **Hardware type.** This is a 16-bit field defining the type of the network on which ARP is running.
- **Protocol type**. This is a 16-bit field defining the protocol.
- **Hardware length**. This is an 8-bit field defining the length of the physical address in bytes.
- **Protocol length**. This is an 8-bit field defining the length of the logical address in bytes.
- **Operation.** This is a 16-bit field defining the type of packet. Two packet types are
- defined: ARP request (1), ARP reply (2).
- **Sender hardware address.** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- **Sender protocol address**. This is a variable-length field defining the logical address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address**. This is a variable-length field defining the physical address of the target.
- **Target protocol address.** This is a variable-length field defining the logical address of the target. For the IPv4 protocol, this field is 4 bytes long.

An ARP request is broadcast; an ARP reply is unicast.

Encapsulation of ARP packet



These are seven steps involved in an ARP process:

1. The sender knows the IP address of the target. We will see how the sender obtains this shortly.

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

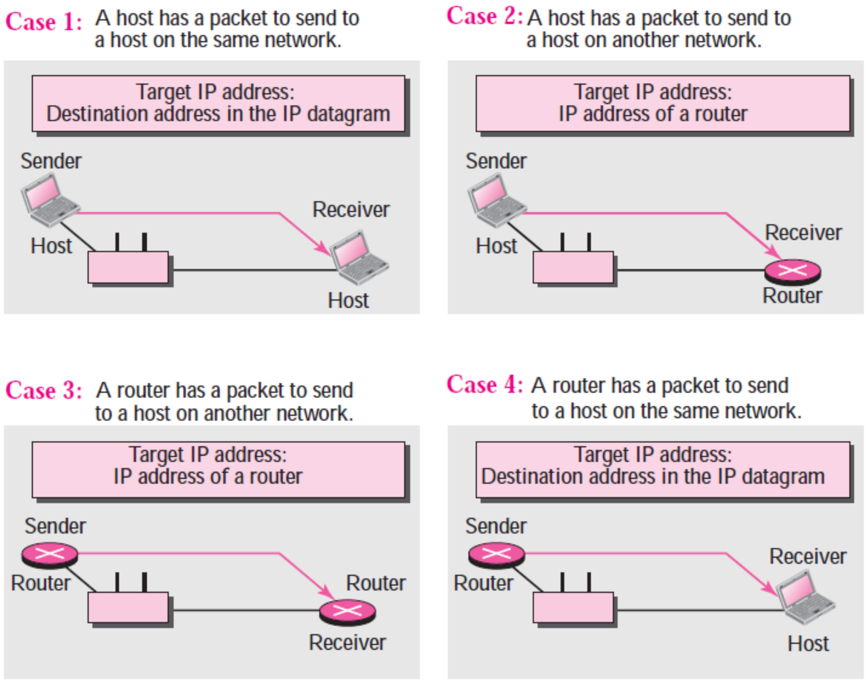
3. The message is passed to the data link layer where it is encapsulated in a frame using the physical address of the sender as the source address and the physical broadcast address as the destination address.

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

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

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

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



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

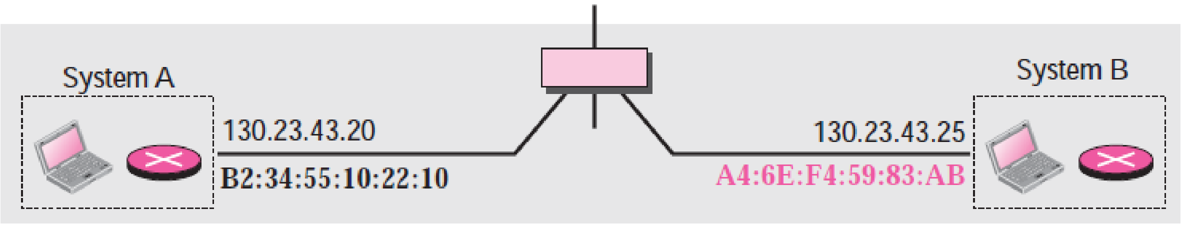
Case 2: A host has a packet to send to a host on another network.

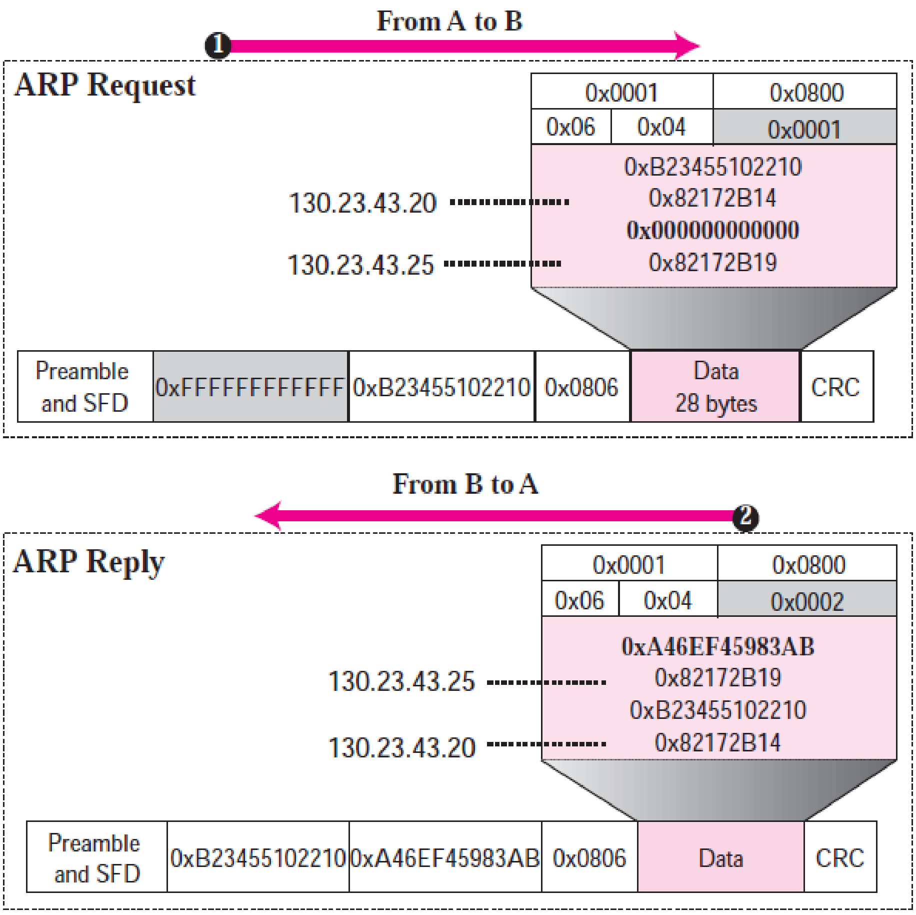
Case 3: A router has a packet to send to a host on another network.

Case 4: A router has a packet to send to a host on the same network.

Example

A host with IP address 130.23.43.20 and physical address B2:34:55:10:22:10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB (which is unknown to the first host). The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.





DHCP Dynamic Host Configuration Protocol

Introduction

Each computer that uses the TCP/IP protocol suite needs to know its IP address. In other words, four pieces of information are normally needed:

1. The IP address of the computer

2. The subnet mask of the computer

3. The IP address of a router

4. The IP address of a name server

These four pieces of information can be stored in a configuration file and accessed by the computer during the bootstrap process.

DHCP

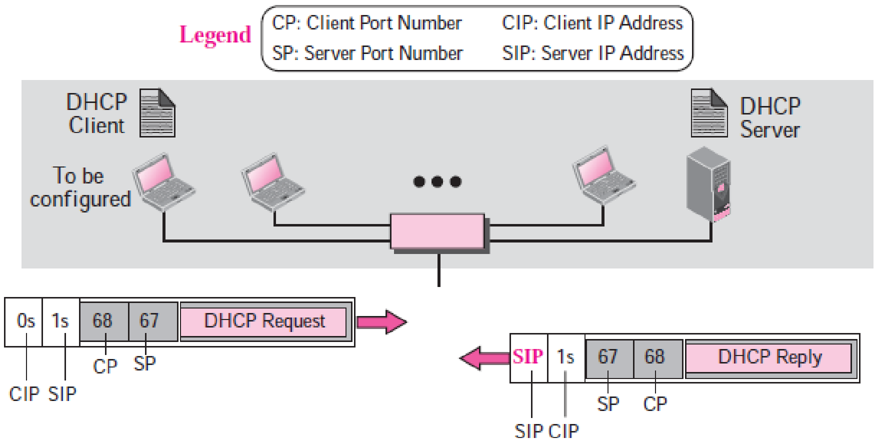
The Dynamic Host Configuration Protocol (DHCP) is a client/server protocol designed to provide the four pieces of information for a diskless computer or a computer that is booted for the first time.

DHCP OPERATION

The DHCP client and server can either be on the same network or on different networks.

Same Network

Although the practice is not very common, the administrator may put the client and the server on the same network.

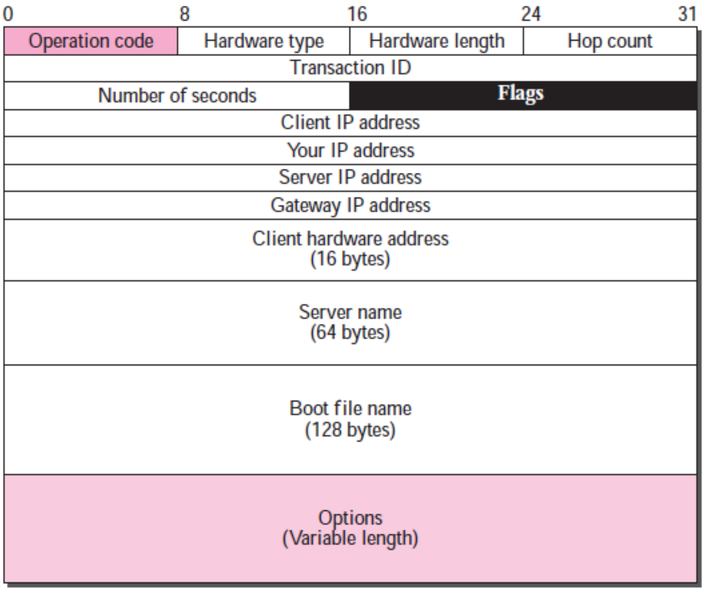


DHCP Operation

1. The DHCP server issues a passive open command on UDP port number 67 and waits for a client.

2. A booted client issues an active open command on port number. The message is encapsulated in a UDP user datagram, using the destination port number 67 and the source port number 68. The UDP user datagram, in turn, is encapsulated in an IP datagram. The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client uses all 0s as the source address and all 1s as the destination address.

3. The server responds with either a broadcast or a unicast message using UDP source port number 67 and destination port number 68. The response can be unicast because the server knows the IP address of the client. It also knows the physical address of the client, which means it does not need the services of ARP for logical to physical address mapping. However, some systems do not allow the bypassing of ARP, resulting in the use of the broadcast address.



Packet Format

❑ Operation code. This 8-bit field defines the type of DHCP packet: request (1) or reply (2).

❑ Hardware type. This is an 8-bit field defining the type of physical network. Each type of network has been assigned an integer. For example, for Ethernet the value is 1.

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

❑ Hop count. This is an 8-bit field defining the maximum number of hops the packet can travel.

❑ Transaction ID. This is a 4-byte field carrying an integer. The transaction identification is set by the client and is used to match a reply with the request. The server returns the same value in its reply.

❑ Number of seconds. This is a 16-bit field that indicates the number of seconds elapsed since the time the client started to boot.

❑ Flag. This is a 16-bit field in which only the leftmost bit is used, and the rest of the bits should be set to 0s. A leftmost bit specifies a forced broadcast reply (instead of unicast) from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client. Since the client does not know its IP address, it may discard the packet. However, if the IP datagram is broadcast, every host will receive and process the broadcast message.

❑ Client IP address. This is a 4-byte field that contains the client IP address. If the client does not have this information, this field has a value of 0.

❑ Your IP address. This is a 4-byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.

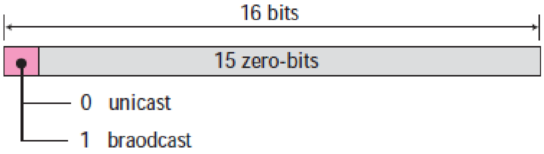
❑ Server IP address. This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.

❑ Gateway IP address. This is a 4-byte field containing the IP address of a router. It is filled by the server in a reply message.

❑ Client hardware address. This is the physical address of the client. Although the server can retrieve this address from the frame sent by the client, it is more efficient if the address is supplied explicitly by the client in the request message.

❑ Server name. This is a 64-byte field that is optionally filled by the server in a reply packet.

Flag format



❑ Client IP address. This is a 4-byte field that contains the client IP address. If the client does not have this information, this field has a value of 0.

❑ Your IP address. This is a 4-byte field that contains the client IP address. It is filled by the server (in the reply message) at the request of the client.

❑ Server IP address. This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.

❑ Gateway IP address. This is a 4-byte field containing the IP address of a router. It is filled by the server in a reply message.

❑ Client hardware address. This is the physical address of the client. Although the server can retrieve this address from the frame sent by the client, it is more efficient if the address is supplied explicitly by the client in the request message.

❑ Server name. This is a 64-byte field that is optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the domain name of the server. If the server does not want to fill this field with data, the server must fill it with all 0s.

❑ Boot filename. This is a 128-byte field that can be optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the full pathname of the boot file. The client can use this path to retrieve other booting information. If the server does not want to fill this field with data, the server must fill it with all 0s.

❑ Options. This is a 64-byte field with a dual purpose. It can carry either additional information (such as the network mask or default router address) or some specific vendor information. The field is used only in a reply message. The server uses a number, called a magic cookie, in the format of an IP address with the value of 99.130.83.99. When the client finishes reading the message, it looks for this magic cookie. If present, the next 60 bytes are options. An option is composed of three fields: a

1-byte tag field, a 1-byte length field, and a variable-length value field. The length field defines the length of the value field, not the whole option.

CONFIGURATION

The DHCP has been devised to provide static and dynamic address allocation.

Static Address Allocation

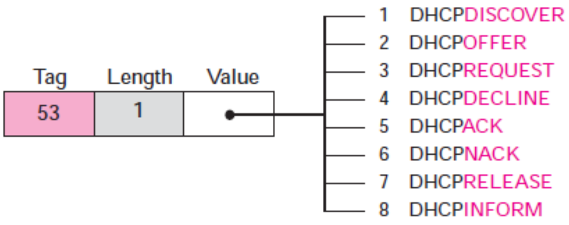
In this capacity, a DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation

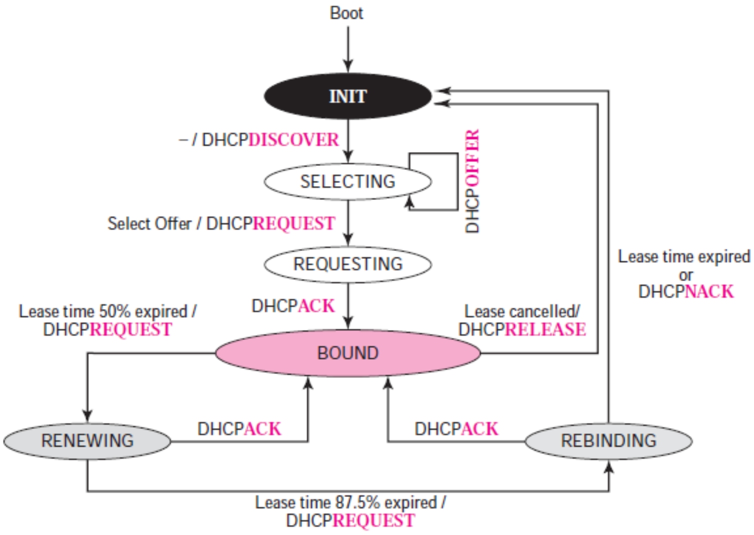
DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific period of time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the choice to agree or disagree with the renewal. If the server disagrees, the client stops using the address.



DISCOVER, OFFER, REQUEST, ACK (DORA).



Boot
INIT
– / DHCPDISCOVER
SELECTING
DHCPOFFER
Select Offer / DHCPREQUEST
REQUESTING
DHCPACK
Lease time 50% expired / DHCPREQUEST
Lease cancelled/ DHCPRELEASE
Lease time expired or DHCPNACK
BOUND
RENEWING
DHCPACK
DHCPACK
REBINDING
Lease time 87.5% expired / DHCPREQUEST

INIT State When the DHCP client first starts, it is in the INIT state (initializing state). The client broadcasts a DHCPDISCOVER message using port 67.

SELECTING State After sending the DHCPDISCOVER message, the client goes to the selecting state. Those servers that can provide this type of service respond with a DHCPOFFER message. In these messages, the servers offer an IP address.

REQUESTING State The client remains in the requesting state until it receives a DHCPACK message from the server that creates the binding between the client physical address and its IP address.

BOUND State In this state, the client can use the IP address until the lease expires.

RENEWING State The client remains in the renewing state until one of two events happens. It can receive a DHCPACK, which renews the lease agreement. if a DHCPACK is not received, and 87.5 percent of the lease time expires, the client goes to the rebinding state.

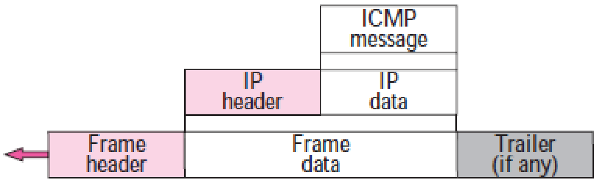
REBINDING State The client remains in the rebinding state until one of three events happens. If the client receives a DHCPNACK or the lease expires, it goes back to the initializing state and tries to get another IP address.

Error Reporting protocol

ICMP-Introduction

The IP protocol has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies.

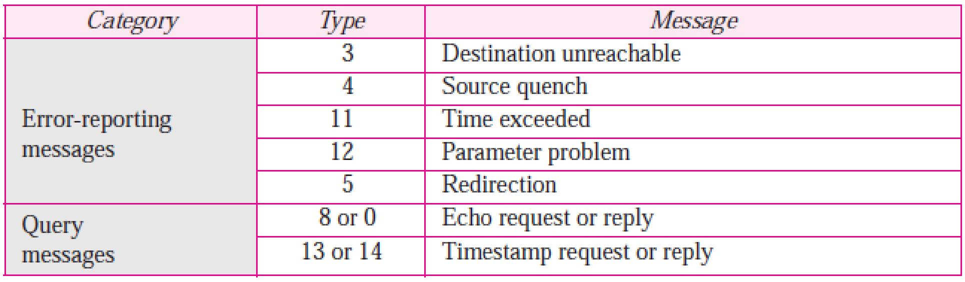
ICMP itself is a network layer protocol. However, its messages are not passed directly to the data link layer as would be expected. Instead, the messages are first encapsulated inside IP datagrams.



MESSAGES

ICMP messages are divided into two broad categories: **error-reporting messages** and **query messages.** The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.

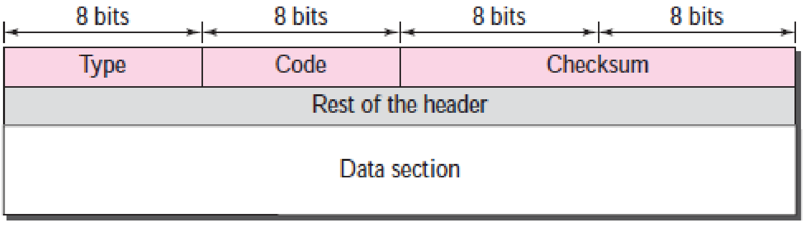
Table 9.1 *ICMP messages*



Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

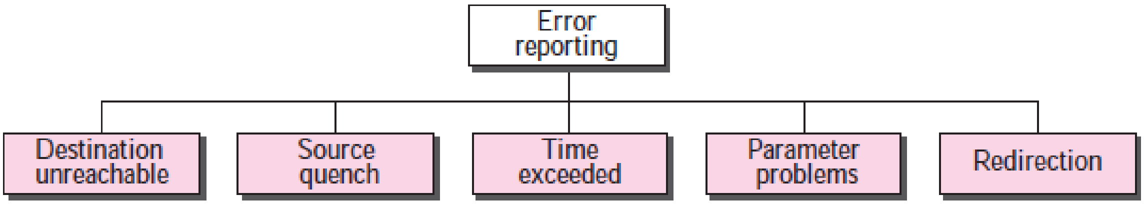
Message Format

An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all.



Error Reporting Messages

One of the main responsibilities of ICMP is to report errors. Although technology has produced increasingly reliable transmission media, errors still exist and must be handled. IP, is an unreliable protocol. This means that error checking and error control are not a concern of IP. ICMP was designed, in part, to compensate for this shortcoming. However, ICMP does not correct errors, it simply reports them. Error correction is left to the higher-level protocols.



- No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- No ICMP error message will be generated for a datagram having a multicast address.
- No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

1. Destination Unreachable

When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram.

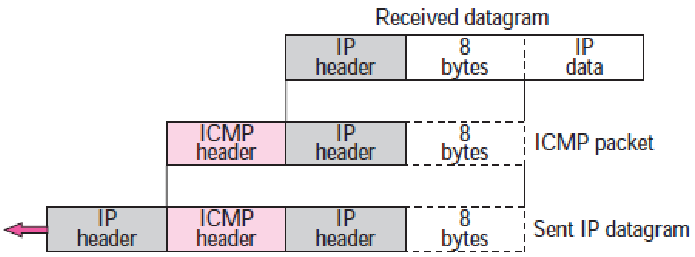
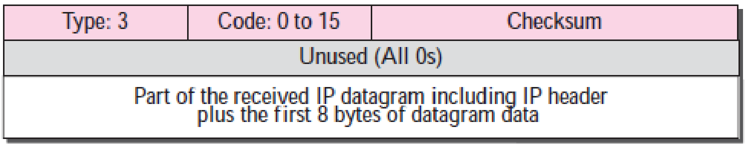


Figure 9.6 *Destination-unreachable format*



Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Destination-unreachable messages with codes 2 or 3 can be created only by the destination host. Other destination-unreachable messages can be created only by routers.

2. Source Quench

The IP protocol is a connectionless protocol. There is no communication between the source host, which produces the datagram, the routers, which forward it, and the destination host, which processes it. One of the ramifications of this absence of communication is the lack of flow control and congestion control.

3. Time Exceeded

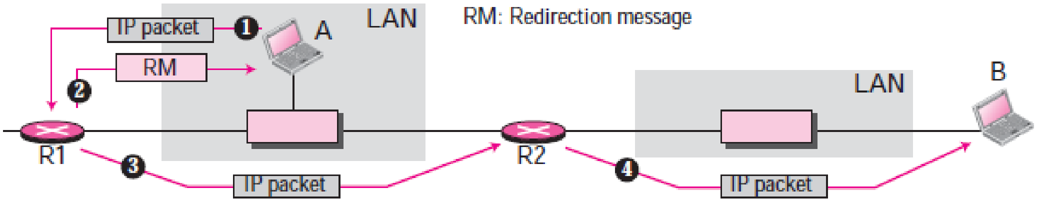
The time-exceeded message is generated in two cases: Whenever a router decrements a datagram with a time-to-live value to zero, it discards the datagram and sends a time-exceeded message to the original source.

4. Parameter Problem

Any ambiguity in the header part of a datagram can create serious problems as the datagram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram,

5. Redirection

When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router. The same is true if the sender is a host. Both routers and hosts then must have a routing table to find the address of the router or the next router.



Query Messages

In addition to error reporting, ICMP can also diagnose some network problems. This is accomplished through the query messages.

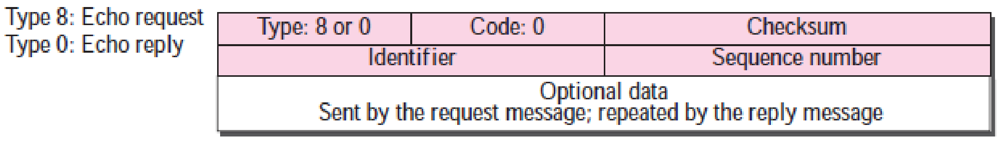
1. Echo Request and Reply

The echo-request and echo-reply messages are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems. An echo-request message can be sent by a host or router. An echo-reply message is sent by the host or router that receives an echo-request message.

2. Timestamp Request and Reply

Two machines (hosts or routers) can use the timestamp-request and timestamp-reply messages to determine the round-trip time needed for an IP datagram to travel between them. It can also be used to synchronize the clocks in two machines.

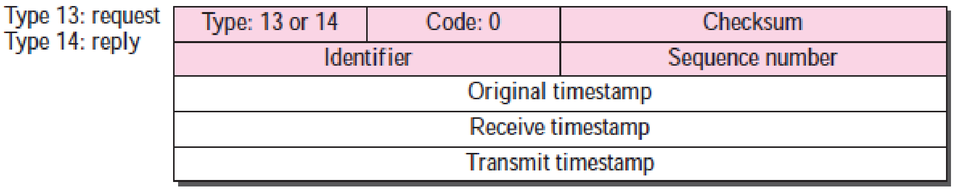
Figure 9.12 *Echo-request and echo-reply messages*



Type 8: Echo request
Type 0: Echo reply

Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

Figure 9.13 *Timestamp-request and timestamp-reply message format*

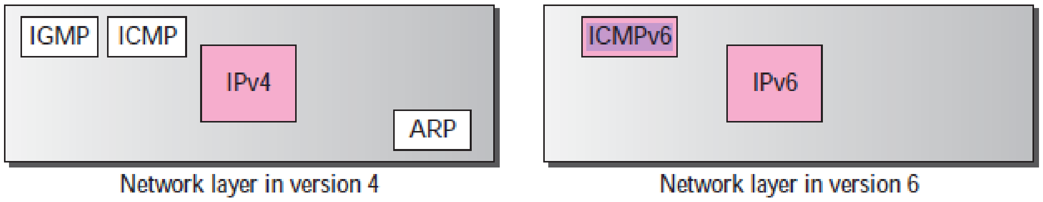


Type 13: request
Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

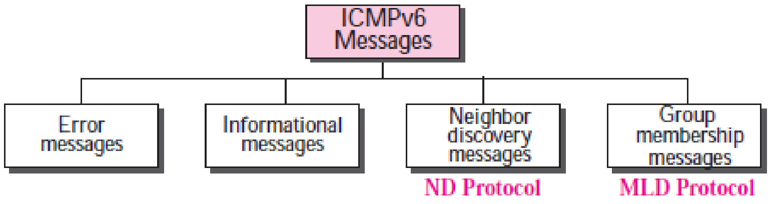
Internet Control Message Protocol version 6 (ICMPv6),

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, like ICMPv4, is message-oriented; it uses messages to report errors, to 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.

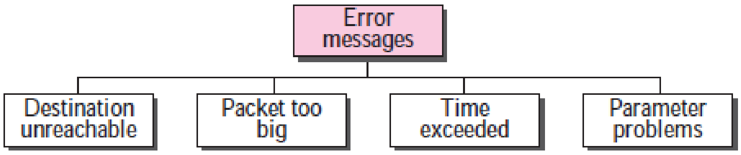
Figure 28.2 *Taxonomy of ICMPv6 messages*



ERROR MESSAGES

Four types of errors are handled: destination unreachable, packet too big, time exceeded, and parameter problems

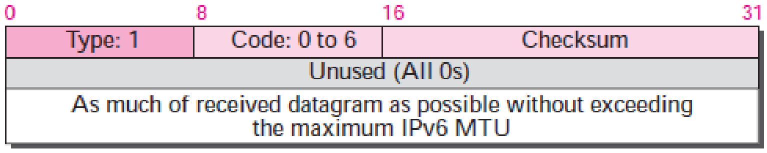
Figure 28.3 *Error-reporting messages*



Destination-Unreachable Message

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.

Figure 28.4 *Destination-unreachable message*



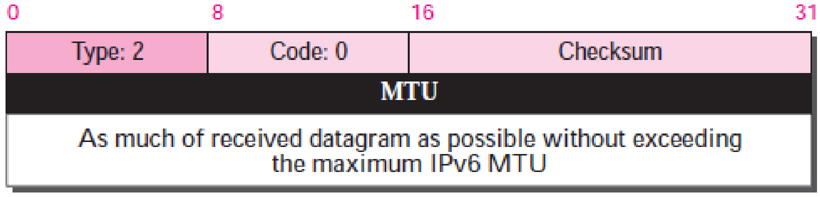
Type: 1	Code: 0 to 6	Checksum
Unused (All 0s)		
As much of received datagram as possible without exceeding the maximum IPv6 MTU		

- Code 0. No path to destination.
- Code 1. Communication with the destination is administratively prohibited.
- Code 2. Beyond the scope of source address.
- Code 3. Destination address is unreachable.
- Code 4. Port unreachable.
- Code 5. Source address failed (filtering policy).
- Code 6. Reject route to destination.

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.

Figure 28.5 *Packet-too-big message*

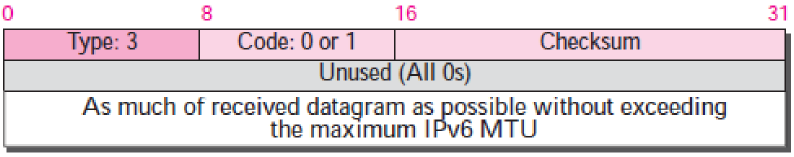


Type: 2	Code: 0	Checksum
MTU		
As much of received datagram as possible without exceeding the maximum IPv6 MTU		

Time-Exceeded Message

As we discussed in Chapter 9, 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.

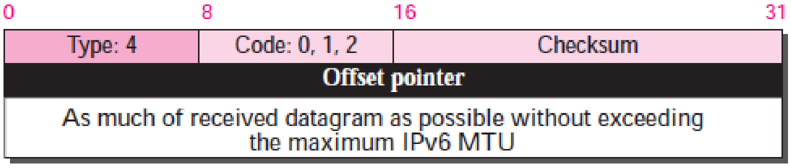
Figure 28.6 *Time-exceeded message*


0
8
16
31
Type: 3
Code: 0 or 1
Checksum
Unused (All 0s)
As much of received datagram as possible without exceeding the maximum IPv6 MTU

Parameter-Problem Message

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.

Figure 28.7 *Parameter-problem message*


0
8
16
31
Type: 4
Code: 0, 1, 2
Checksum
Offset pointer
As much of received datagram as possible without exceeding the maximum IPv6 MTU

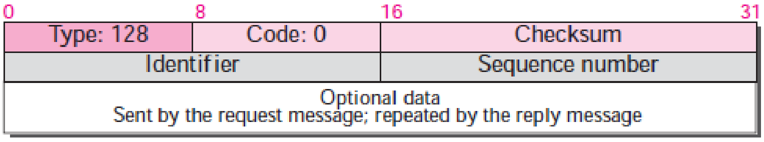
INFORMATIONAL MESSAGES

Two of the ICMPv6 messages can be categorized as informational messages: echo request and echo reply messages.

Echo-Request Message

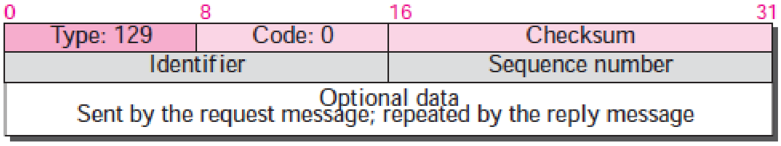
The idea and format of the echo-request message is the same as the one in version 4. The only difference is the value for the type.

Figure 28.8 *Echo-request message*



Echo-Reply Message

The idea and format of the echo-reply message is the same as the one in version 4. The only difference is the value for the type.



NEIGHBOR-DISCOVERY MESSAGES

Several messages in the ICMPv6 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 the neighbor.

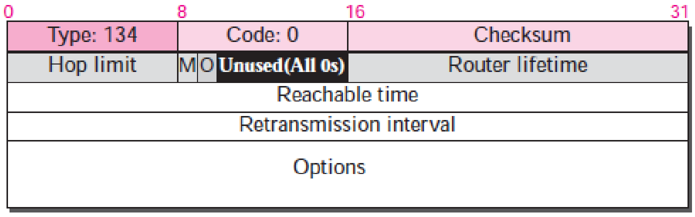
Router-Solicitation Message

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 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. shows the format of the router-advertisement message.

Figure 28.11 *Router-advertisement message*



- Hop Limit. This 8-bit field limits the number of hops that the requestor should use as the hop limit in its IPv6 datagram.
- M. This 1-bit field is the “manage address configuration” field. When this bit is set to 1, the host needs to use the administration configuration.
- O. This 1-bit field is the “other address configuration” field. When this bit is set to 1, the host needs to use the appropriate protocol for configuration.
- Router Lifetime. This 16-bit field defines the lifetime (in units of seconds) of the router as the default router. When the value of this field is 0, it means that the router is not a default router.
- Reachable Time. This 32-bit field defines the time (in units of seconds) that the router is reachable.
- Retransmission Interval. This 32-bit field defines the retransmission interval (in units of seconds).
- Option. Some possible options are the link layer address of the link from which the message is sent, the MTU of the link, and address prefix information.

Neighbor-Solicitation Message

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 use a unicast response.

Neighbor-Advertisement Message

The neighbor-advertisement message is sent in response to the neighbor-solicitation message. This is equivalent to the ARP reply message in IPv4.

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.

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.

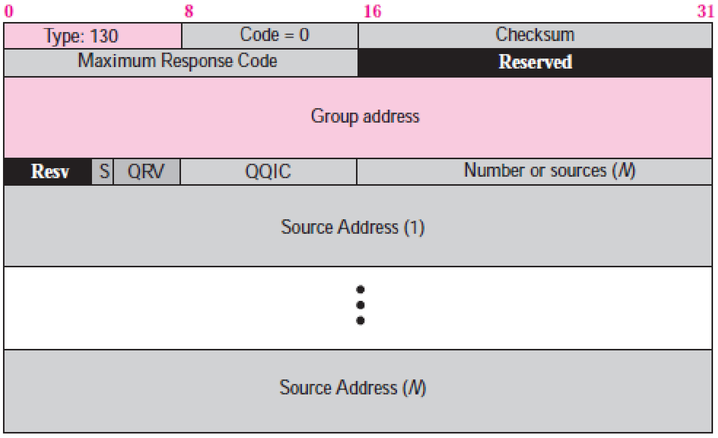
GROUP MEMBERSHIP MESSAGES

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 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.

Figure 28.17 *Membership-query message format*



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.

Membership-Report Message

Note that 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.

Functionality

MDLv2 protocol behaves in the same way as IGMPv3.

Calculation of Maximum Response Time

As we mentioned, the size of the Max Resp Code in MLV2 is twice the size of the same field in IGMPv3. For this reason, the calculation of maximum response time is slightly different in this protocol.

Calculation of Query Interval

The calculation of query interval follows the same process as the calculation of maximum response delay; it is calculated from the value of the QQIC field.

Routing Protocols

INTRODUCTION

An internet is a combination of networks connected by routers. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.

Cost or Metric

One approach is to assign a cost for passing through a network. We call this cost a metric. High cost can be thought of as something bad; low cost can be thought of something good.

Static versus Dynamic Routing Tables

A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet.

Routing Protocol

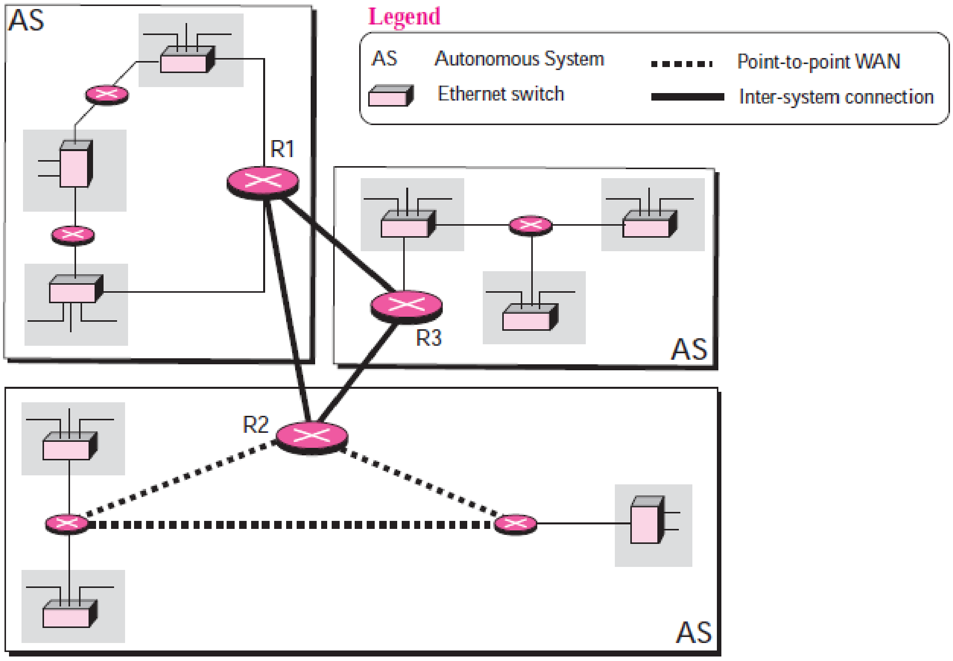
Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that lets routers in the internet inform each other of changes. It allows routers to share whatever they know about the internet or their neighborhood.

Routing protocols can be either an interior protocol or an exterior protocol. An interior protocol handles intradomain routing; an exterior protocol handles interdomain routing. We start the next section with defining these terms.

INTRA- AND INTER-DOMAIN ROUTING

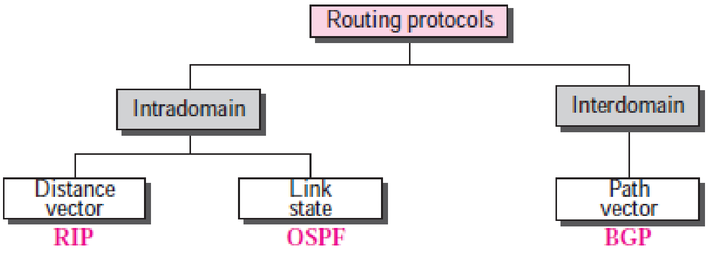
An internet is divided into **autonomous systems**. An **autonomous system** (AS) is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as **intra-domain routing**. Routing between autonomous systems is referred to as **inter-domain routing**. Each autonomous system can choose one or more intradomain routing protocols to handle routing inside the autonomous system. However, only one interdomain routing protocol handles routing between autonomous systems.

Figure 11.1 *Autonomous systems*



Routing Information Protocol (RIP) is the implementation of the distance vector protocol. Open Shortest Path First (OSPF) is the implementation of the link state protocol. Border Gateway Protocol (BGP) is the implementation of the path vector protocol. RIP and OSPF are interior routing protocols; BGP is an exterior routing protocol.

Figure 11.2 *Popular routing protocols*



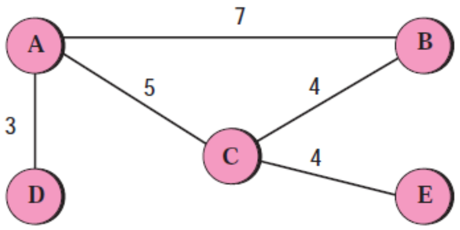
DISTANCE VECTOR ROUTING

We first discuss distance vector routing. This method sees an AS, with all routers and networks, as a graph, **a set of nodes** and lines (edges) connecting the nodes. A router can normally be represented by a node and a network by a link connecting two nodes, although other representations are also possible. The graph theory used an algorithm called Bellman-Ford (also called Ford-Fulkerson) for a while to find the shortest path between nodes in a graph given the distance between nodes.

Bellman-Ford Algorithm

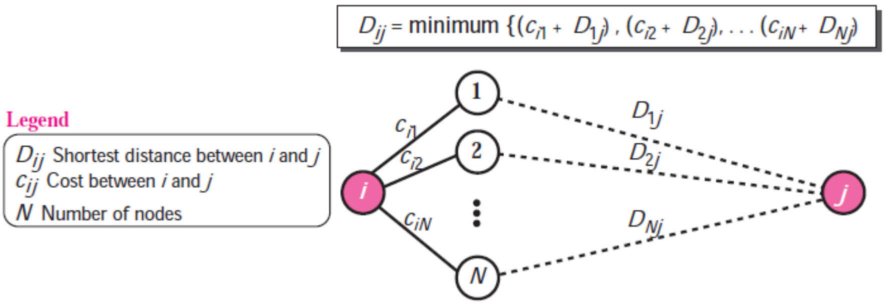
If we know the cost between each pair of nodes, we can use the algorithm to find the least cost (shortest path) between any two nodes. shows a map with nodes and lines. The cost of each line is given over the line; the algorithm can find the least cost between any two nodes.

Figure 11.3 *A graph for the Bellman-Ford algorithm*



The algorithm is based on the fact that if all neighbors of node *i* know the shortest distance to node *j*, then the shortest distance between node *i* and *j* can be found by adding the distance between node i and each neighbor to the neighbor's shortest distance to node *j* and then select the minimum.

Figure 11.4 *The fact behind Bellman-Ford algorithm*


D_{ij} = minimum $\{(c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \ldots (c_{iN} + D_{Nj})\}$
Legend
D_{ij} Shortest distance between i and j
c_{ij} Cost between i and j
N Number of nodes
c_{i1}
c_{i2}
c_{iN}
1
2
N
i
j
D_{1j}
D_{2j}
D_{Nj}

We create a shortest distance table (vector) for each node using the following steps:

1. The shortest distance and the cost between a node and itself is initialized to 0.
2. The shortest distance between a node and any other node is set to infinity. The cost between a node and any other node should be given (can be infinity if the nodes are not connected).
3. The algorithm repeats as shown in Figure 11.4 until there is no more change in the shortest distance vector.

Distance Vector Routing Algorithm

The Bellman-Ford algorithm can be very well applied to a map of roads between cities because we can have all the initial information about each node at the same place. We can enter this information into the computer and let the computer hold the intermediate results and create the final vectors for each node to be printed. In other words, the algorithm is designed to create the result synchronously.

1. In distance vector routing, the cost is normally hop counts (how many networks are passed before reaching the destination). So the cost between any two neighbors is set to 1.
2. Each router needs to update its routing table asynchronously, whenever it has received some information from its neighbors. In other words, each router executes part of the whole algorithm in the Bellman-Ford algorithm. Processing is distributive.
3. After a router has updated its routing table, it should send the result to its neighbors so that they can also update their routing table.
4. Each router should keep at least three pieces of information for each route: destination network, the cost, and the next hop. We refer to the whole routing table as Table, to the row i in the table as $Table_i$, to the three columns in row i as $Table_i$. dest, $Table_i$. cost, and $Table_i$. next.
5. We refer to information about each route received from a neighbor as R (record), which has only two pieces of information: R.dest and R.cost. The next hop is not included in the received record because it is the source address of the sender.

Count to Infinity

A problem with distance vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly. For a routing protocol to work properly, if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance vector routing, this takes some time. The problem is referred to **as count to infinity**.

Two-Node Loop

One example of count to infinity is the two-node loop problem.

Defining Infinity The first obvious solution is to redefine infinity to a smaller number, such as 16. For our previous scenario, the system will be stable in fewer updates.

Split Horizon Another solution is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface.

Split Horizon and Poison Reverse Using the split horizon strategy has one drawback. Normally, the Distance Vector Protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: “Do not use this value; what I know about this route comes from you.”

Three-Node Instability

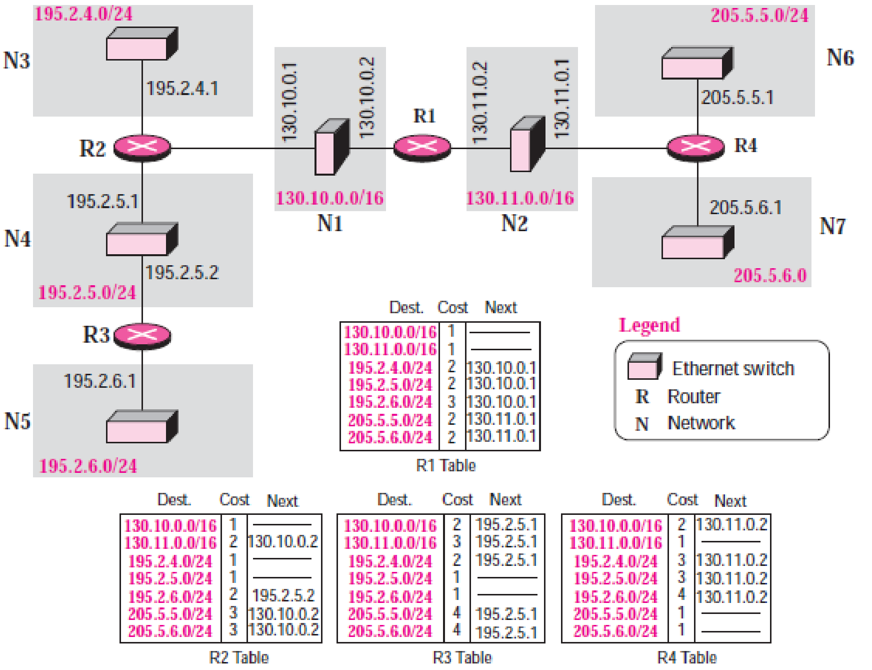
The two-node instability can be avoided using split horizon combined with poison reverse. However, if the instability is between three nodes, stability cannot be guaranteed.

RIP (Routing Information Protocol)

The Routing Information Protocol (RIP) is an intradomain (interior) routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:

1. In an autonomous system, we are dealing with routers and networks (links), what was described as a node.
2. The destination in a routing table is a network, which means the first column defines a network address.
3. The metric used by RIP is very simple; the distance is defined as the number of links (networks) that have to be used to reach the destination. For this reason, the metric in RIP is called a hop count.
4. Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
5. The next node column defines the address of the router to which the packet is to be sent to reach its destination.

Figure 11.10 *Example of a domain using RIP*



Dest.	Cost	Next
130.10.0.0/16	1	———
130.11.0.0/16	1	———
195.2.4.0/24	2	130.10.0.1
195.2.5.0/24	2	130.10.0.1
195.2.6.0/24	3	130.10.0.1
205.5.5.0/24	2	130.11.0.1
205.5.6.0/24	2	130.11.0.1

R1 Table

Dest.	Cost	Next
130.10.0.0/16	1	———
130.11.0.0/16	2	130.10.0.2
195.2.4.0/24	1	———
195.2.5.0/24	1	———
195.2.6.0/24	2	195.2.5.2
205.5.5.0/24	3	130.10.0.2
205.5.6.0/24	3	130.10.0.2

R2 Table

Dest.	Cost	Next
130.10.0.0/16	2	195.2.5.1
130.11.0.0/16	3	195.2.5.1
195.2.4.0/24	2	195.2.5.1
195.2.5.0/24	1	———
195.2.6.0/24	1	———
205.5.5.0/24	4	195.2.5.1
205.5.6.0/24	4	195.2.5.1

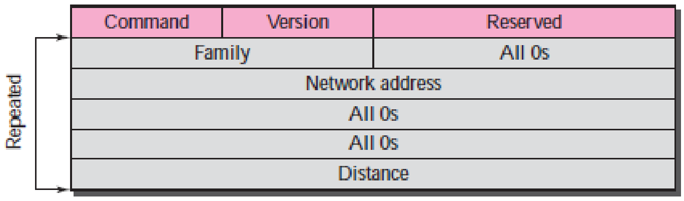
R3 Table

Dest.	Cost	Next
130.10.0.0/16	2	130.11.0.2
130.11.0.0/16	1	———
195.2.4.0/24	3	130.11.0.2
195.2.5.0/24	3	130.11.0.2
195.2.6.0/24	4	130.11.0.2
205.5.5.0/24	1	———
205.5.6.0/24	1	———

R4 Table

RIP Message Format

Figure 11.11 *RIP message format*



- **Command**. This 8-bit field specifies the type of message: request (1) or response (2).
- **Version**. This 8-bit field defines the version. In this book we use version 1, but at the end of this section, we give some new features of version 2.
- **Family**. This 16-bit field defines the family of the protocol used. For TCP/IP the value is 2.
- **Network address**. The address field defines the address of the destination network. RIP has allocated 14 bytes for this field to be applicable to any protocol. However, IP currently uses only 4 bytes. The rest of the address is filled with 0s.
- **Distance.** This 32-bit field defines

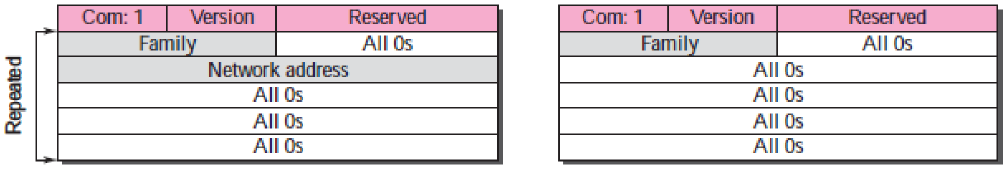
Requests and Responses

RIP has two types of messages: request and response.

Request

A request message is sent by a router that has just come up or by a router that has some time-out entries. A request can ask about specific entries or all entries.

Figure 11.12 *Request messages*



a. Request for some

b. Request for all

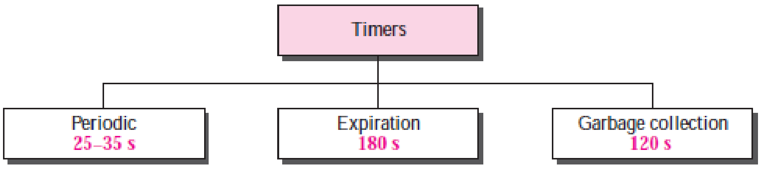
Response

A response can be either solicited or unsolicited. A solicited response is sent only in answer to a request. It contains information about the destination specified in the corresponding request. An unsolicited response, on the other hand, is sent periodically, every 30 seconds or when there is a change in the routing table. The response is sometimes called an update packet.

Timers in RIP

RIP uses three timers to support its operation. The periodic timer controls the sending of messages, the expiration timer governs the validity of a route, and the garbage collection timer advertises the failure of a route.

Figure 11.14 *RIP timers*



Periodic Timer

The periodic timer controls the advertising of regular update messages. Although the protocol specifies that this timer must be set to 30 s, the working model uses a random number between 25 and 35 s. This is to prevent any possible synchronization and therefore overload on an internet if routers update simultaneously.

Expiration Timer

The expiration timer governs the validity of a route. When a router receives update information for a route, the expiration timer is set to 180 s for that particular route. Every time a new update for the route is received, the timer is reset. In normal situations this occurs every 30 s. However, if there is a problem on an internet and no update is received within the allotted 180 s, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable. Every route has its own expiration timer.

Garbage Collection Timer

When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16. At the same time, a timer called the garbage collection timer is set to 120 s for that route.

RIP Version 2

RIP version 2 was designed to overcome some of the shortcomings of version 1. The designers of version 2 have not augmented the length of the message for each entry. They have only replaced those fields in version 1 that were filled with 0s for the TCP/IP protocol with some new fields.

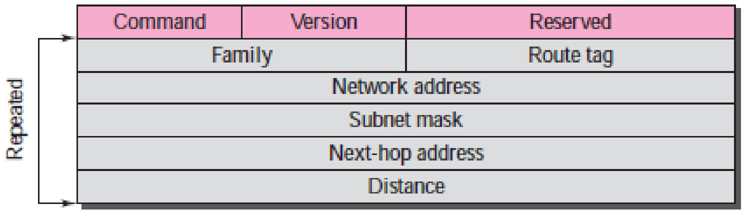
Message Format

Route tag. This field carries information such as the autonomous system number. It can be used to enable RIP to receive information from an interdomain routing protocol.

Subnet mask. This is a 4-byte field that carries the subnet mask (or prefix). This means that RIP2 supports classless addressing and CIDR.

Next-hop address. This field shows the address of the next hop..

Figure 11.15 *RIP version 2 format*



Classless Addressing

Probably the most important difference between the two versions of RIP is classful versus classless addressing. RIPv1 uses classful addressing.

Authentication

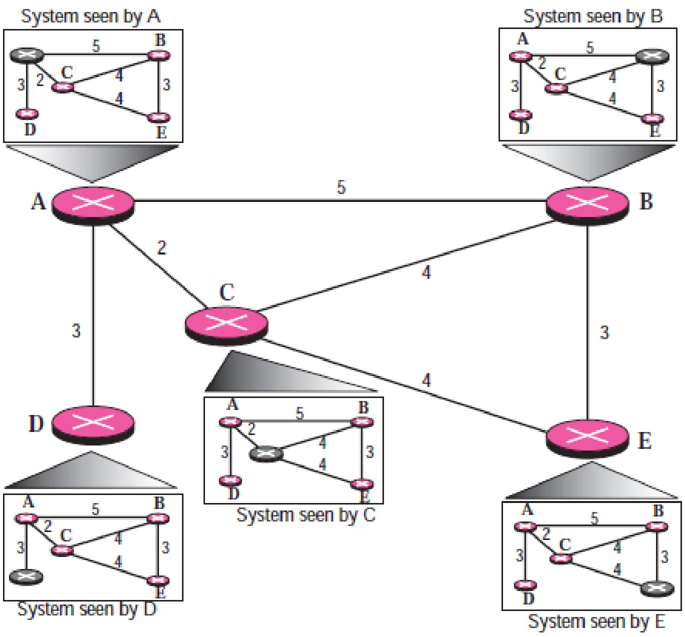
Authentication is added to protect the message against unauthorized advertisement. No new fields are added to the packet; instead, the first entry of the message is set aside for authentication information.

RIP uses the services of UDP on well-known port 520.

LINK STATE ROUTING

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

Figure 11.17 *Concept of link state routing*



Building Routing Tables

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

1. Creation of the states of the links by each node, called the link state packet or LSP.
2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Creation of Link State Packet (LSP)

A link state packet (LSP) can carry a large amount of information. We assume that it carries a minimum amount of data: the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time. LSPs are generated on two occasions:

1. **When there is a change in the topology of the domain**. Triggering of LSP dissemination is the main way of quickly informing any node in the domain to update its topology.
2. **On a periodic basis**. The period in this case is much longer compared to distance vector routing. As a matter of fact, there is no actual need for this type of LSP dissemination. It is done to ensure that old information is removed from the domain. The timer set for periodic dissemination is normally in the range of 60 minutes or 2 hours based on the implementation. A longer period ensures that flooding does not create too much traffic on the network.

Flooding of LSPs

After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:

1. The creating node sends a copy of the LSP out of each interface.
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
    a. It discards the old LSP and keeps the new one.
    b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Formation of Shortest Path Tree: Dijkstra Algorithm

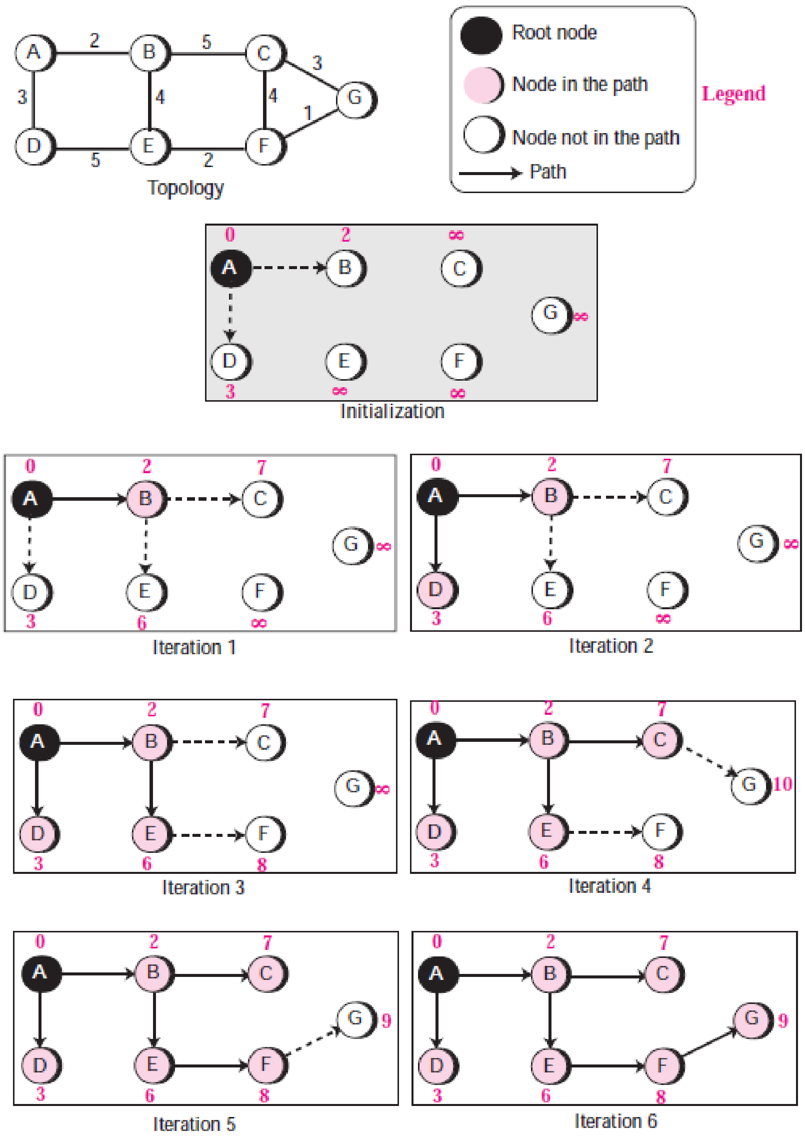
After receiving all LSPs, each node will have a copy of the whole topology. However, the topology is not sufficient to find the shortest path to every other node; a shortest path tree is needed.

A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route. A shortest path tree is a tree in which the path between the root and every other node is the shortest. What we need for each node is a shortest path tree with that node as the root. The Dijkstra algorithm is used to create a shortest path tree from a given graph. The algorithm uses the following

steps:

1. **Initialization**: Select the node as the root of the tree and add it to the path. Set the shortest distances for all the root's neighbors to the cost between the root and those neighbors. Set the shortest distance of the root to zero.
2. **Iteration**: Repeat the following two steps until all nodes are added to the path:
    a. **Adding the next node to the path**: Search the nodes not in the path. Select the one with minimum shortest distance and add it to the path.
    b. **Updating**: Update the shortest distance for all remaining nodes using the shortest distance of the node just moved to the path in step 2.

Figure 11.19 *Forming shortest path three for router A in a graph*



Topology

Initialization

Iteration 1

Iteration 2

Iteration 3

Iteration 4

Iteration 5

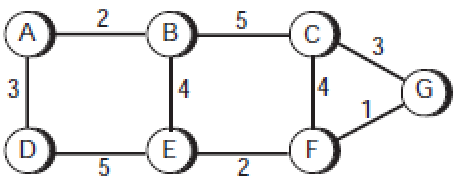
Iteration 6

Calculation of Routing Table from Shortest Path Tree

Each node uses the shortest path tree found in the previous discussion to construct its routing table. The routing table shows the cost of reaching each node from the root.

Table 11.4 *Routing Table for Node A*

Destination	*Cost*	*Next Router*
A	0	—
B	2	—
C	7	B
D	3	—
E	6	B
F	8	B
G	9	B



Topology

OSPF (Open Shortest Path First)

The Open Shortest Path First (OSPF) protocol is an intradomain routing protocol based on link state routing. Its domain is also an autonomous system.

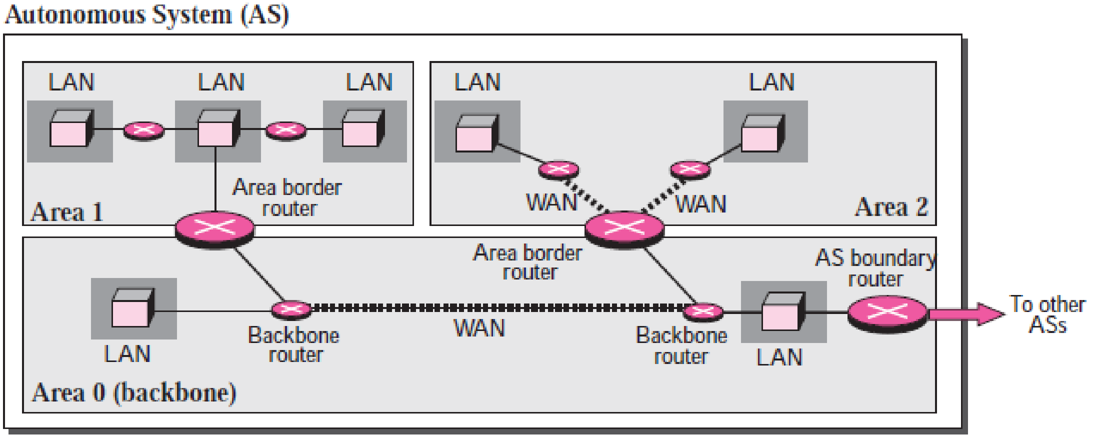
Areas

To handle routing efficiently and in a timely manner, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be divided into many different areas. All networks inside an area must be connected.

Routers inside an area flood the area with routing information. At the border of an area, special routers called **area border routers** summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the backbone; all of the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. This does not mean that the routers within areas cannot be connected to each other, however. The routers inside the backbone are called the **backbone routers**.

If, because of some problem, the connectivity between a backbone and an area is broken, **a virtual link** between routers must be created by the administration to allow continuity of the functions of the backbone as the primary area.

Figure 11.21 *Areas in an autonomous system*



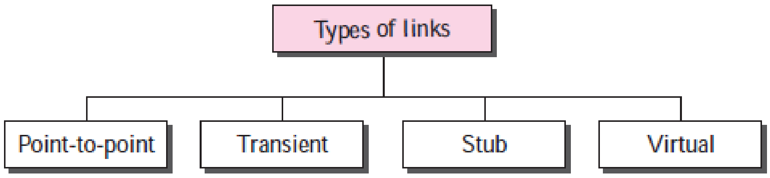
Metric

The OSPF protocol allows the administrator to assign a cost, called the metric, to each route. The metric can be based on a type of service (minimum delay, maximum throughput, and so on). As a matter of fact, a router can have multiple routing tables, each based on a different type of service.

Types of Links

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

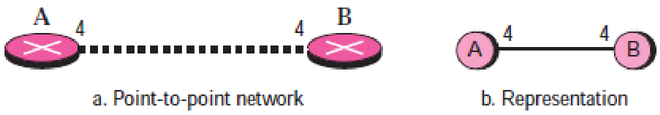
Figure 11.22 *Types of links*



Point-to-Point Link

A point-to-point link connects two routers without any other host or router in between. In other words, the purpose of the link (network) is just to connect the two routers. The metrics, which are usually the same, are shown at the two ends, one for each direction. In other words, each router has only one neighbor at the other side of the link.

Figure 11.23 *Point-to-point link*

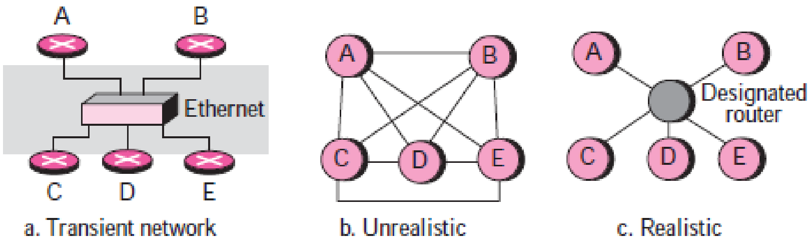


Transient Link

A transient link is a network with several routers attached to it. The data can enter through any of the routers and leave through any router. All LANs and some WANs with two or more routers are of this type. In this case, each router has many neighbors.

Router A has routers B, C, D, and E as neighbors. Router B has routers A, C, D, and E as neighbors. If we want to show the neighborhood relationship in this situation,

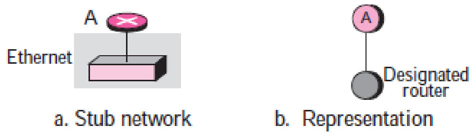
Figure 11.24 *Transient link*



Stub Link

A stub link is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router. This is a special case of the transient network. We can show this situation using the router as a node and using the designated router for the network. However, the link is only one directional, from the router to the network.

Figure 11.25 *Stub link*



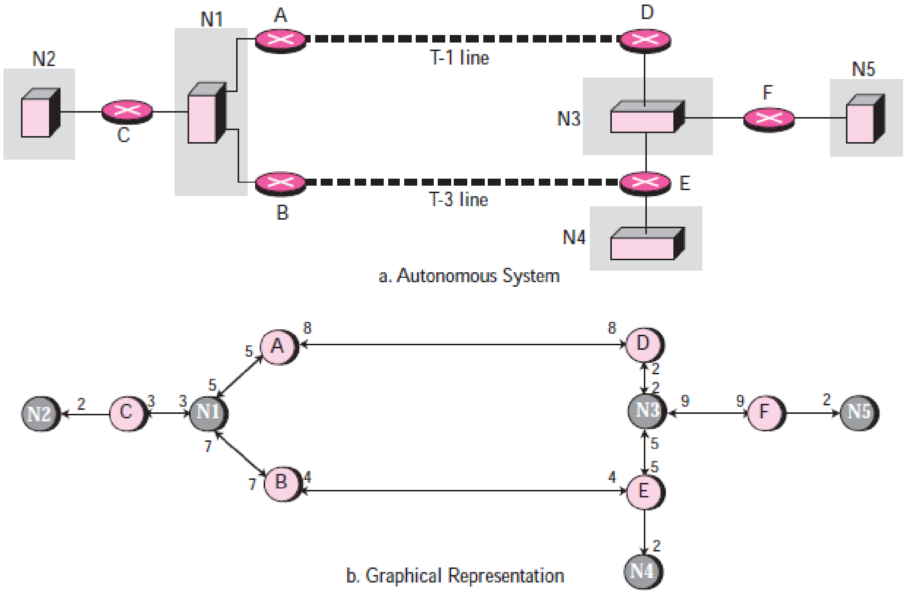
a. Stub network b. Representation

Virtual Link

When the link between two routers is broken, the administration may create a virtual link between them using a longer path that probably goes through several routers.

Graphical Representation

Figure 11.26 *Example of an AS and its graphical representation in OSPF*



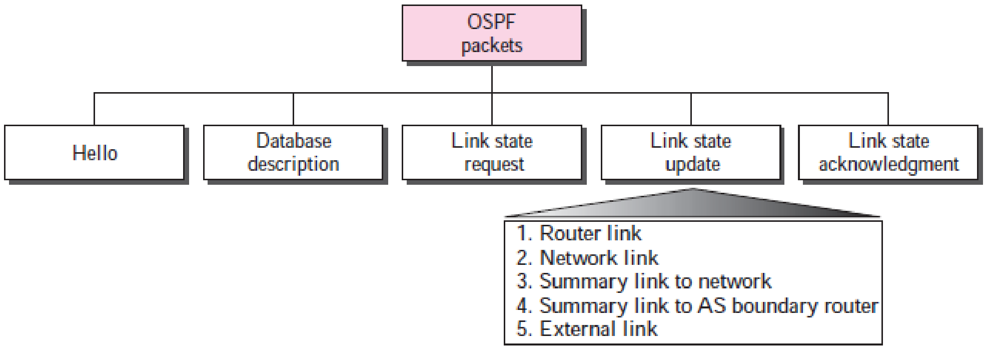
a. Autonomous System

b. Graphical Representation

OSPF Packets

OSPF uses five different types of packets: hello, database description, link state request, link state update, and link state acknowledgment. The most important one is the link state update that itself has five different kinds.

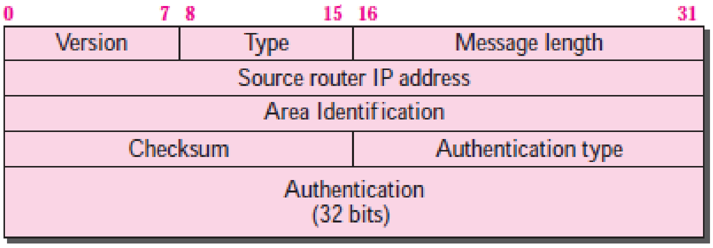
Figure 11.27 *Types of OSPF packets*



Common Header

All OSPF packets have the same common header (see Figure 11.28). Before studying the different types of packets, let us talk about this common header.

Figure 11.28 *OSPF common header*



Version. This 8-bit field defines the version of the OSPF protocol. It is currently version 2.

Type. This 8-bit field defines the type of the packet. As we said before, we have five types, with values 1 to 5 defining the types.

Message length. This 16-bit field defines the length of the total message including the header.

Source router IP address. This 32-bit field defines the IP address of the router that sends the packet.

Area identification. This 32-bit field defines the area within which the routing takes place.

Checksum. This field is used for error detection on the entire packet excluding the authentication type and authentication data field.

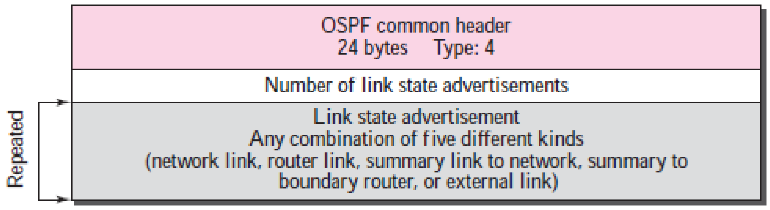
Authentication type. This 16-bit field defines the authentication protocol used in this area. At this time, two types of authentications are defined: 0 for none and 1 for password.

Authentication. This 64-bit field is the actual value of the authentication data. In the future, when more authentication types are defined, this field will contain the result of the authentication calculation. For now, if the authentication type is 0, this field is filled with 0s. If the type is 1, this field carries an eight-character password.

Link State Update Packet

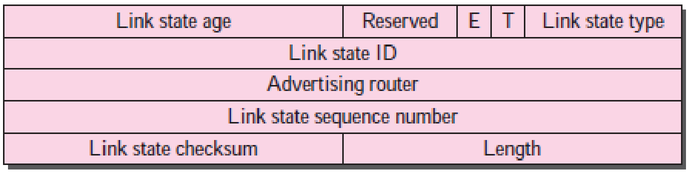
We first discuss the link state update packet, the heart of the OSPF operation. It is used by a router to advertise the states of its links. The general format of the link state update packet is shown below.

Figure 11.29 *Link state update packet*


OSPF common header
24 bytes Type: 4
Number of link state advertisements
Link state advertisement
Any combination of five different kinds
(network link, router link, summary link to network, summary to boundary router, or external link)
Repeated

Each update packet may contain several different LSAs. All five kinds have the same general header.

Figure 11.30 *LSA general header*



Link state age. This field indicates the number of seconds elapsed since this message was first generated. Recall that this type of message goes from router to router (flooding). When a router creates the message, the value of this field is 0. When each successive router forwards this message, it estimates the transit time and adds it to the cumulative value of this field.

E flag. If this 1-bit flag is set to 1, it means that the area is a stub area. A stub area is an area that is connected to the backbone area by only one path.

T flag. If this 1-bit flag is set to 1, it means that the router can handle multiple types of service.

Link state type. This field defines the LSA type. As we discussed before, there are five different advertisement types: router link (1), network link (2), summary link to network (3), summary link to AS boundary router (4), and external link (5).

Link state ID. The value of this field depends on the type of link. For type 1 (router link), it is the IP address of the router. For type 2 (network link), it is the IP address of the designated router. For type 3 (summary link to network), it is the address of the network. For type 4 (summary link to AS boundary router), it is the IP address of the AS boundary router. For type 5 (external link), it is the address of the external network.

Advertising router. This is the IP address of the router advertising this message.

Link state sequence number. This is a sequence number assigned to each link state update message.

Link state checksum. This is not the usual checksum. Instead, the value of this field is calculated using Fletcher's checksum (see Appendix C), which is based on the whole packet except for the age field.

Length. This defines the length of the whole packet in bytes.

Router Link LSA

A router link defines the links of a true router. A true router uses this advertisement to announce information about all of its links and what is at the other side of the link (neighbors). See Figure 11.31 for a depiction of a router link.

The router link LSA advertises all of the links of a router (true router). The format of the router link packet is shown in Figure 11.32. The fields of the router link LSA are as follows:

Link ID. The value of this field depends on the type of link. Table 11.5 shows the different link identifications based on link type.

Link data. This field gives additional information about the link. Again, the value depends on the type of the link (see Table 11.5).

Figure 11.31 *Router link*

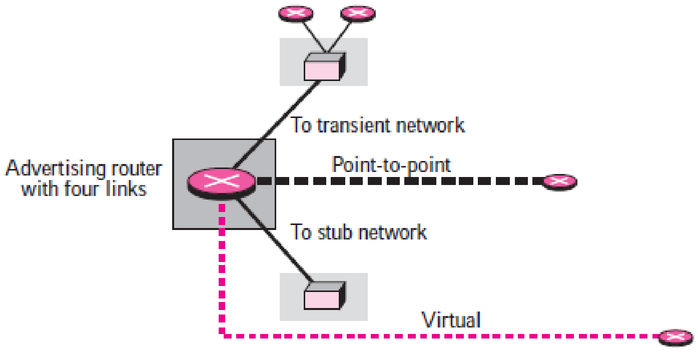


Figure 11.32 *Router link LSA*

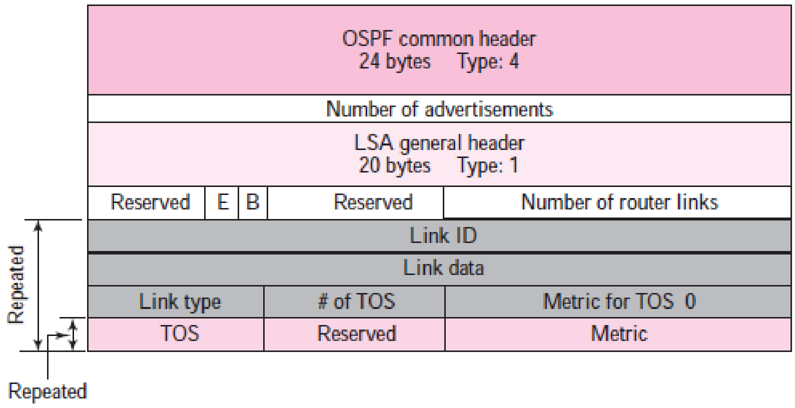


Table 11.5 *Link Types, Link Identification, and Link Data*

Link Type	*Link Identification*	*Link Data*
Type 1: Point-to-point	Address of neighbor router	Interface number
Type 2: Transient	Address of designated router	Router address
Type 3: Stub	Network address	Network mask
Type 4: Virtual	Address of neighbor router	Router address

Link type. Four different types of links are defined based on the type of network to which the router is connected (see Table 11.5).

Number of types of service (TOS). This field defines the number of types of services announced for each link.

Metric for TOS 0. This field defines the metric for the default type of service (TOS 0).

TOS. This field defines the type of service.

Metric. This field defines the metric for the corresponding TOS.

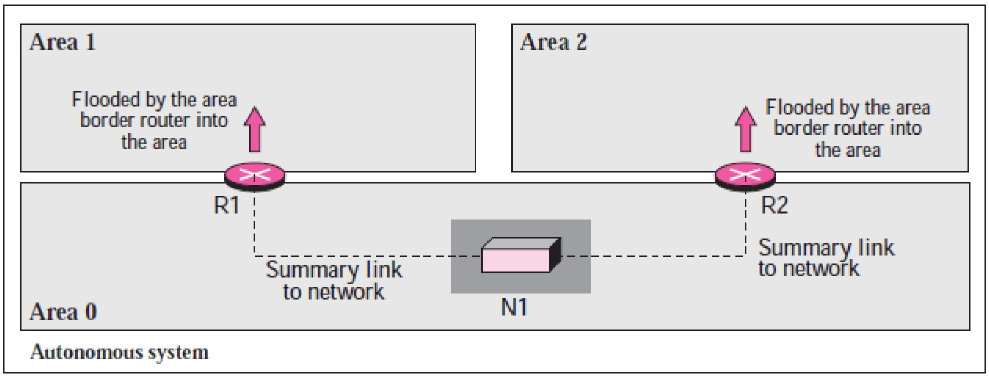
Network Link LSA

A network link defines the links of a network. A designated router, on behalf of the transient network, distributes this type of LSP packet. The packet announces the existence of all of the routers connected to the network.

Summary Link to Network LSA

Router link and network link advertisements flood the area with information about the router links and network links inside an area. But a router must also know about the networks outside its area; the area border routers can provide this information. An area border router is active in more than one area. It receives router link and network link advertisements, and, as we will see, creates a routing table for each area.

Figure 11.40 *Summary link to network*


Area 1
Flooded by the area border router into the area
Area 2
Flooded by the area border router into the area
R1
R2
Summary link to network
Summary link to network
N1
Area 0
Autonomous system

PATH VECTOR ROUTING

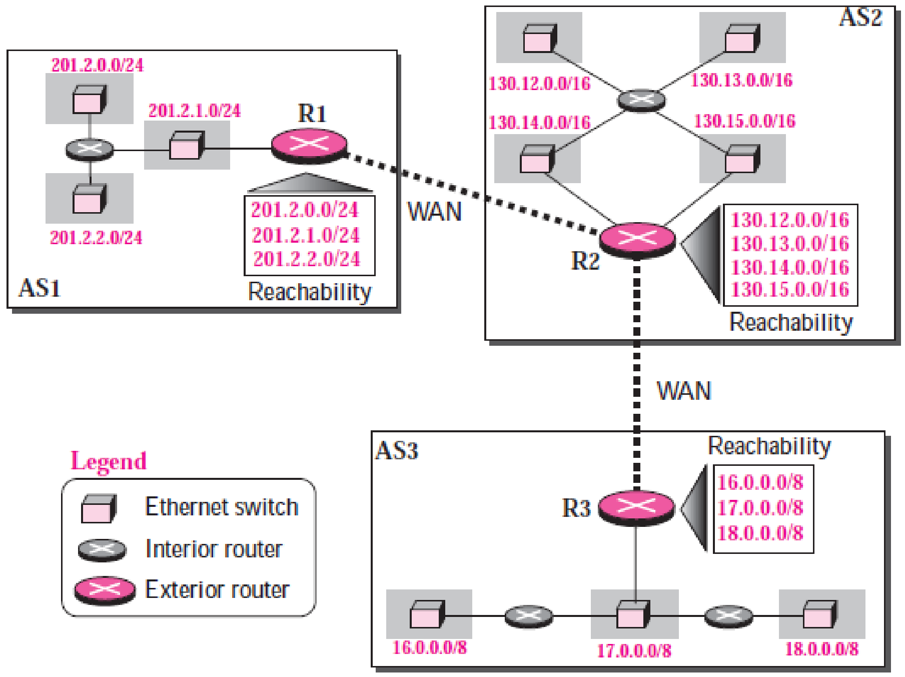
Distance vector and link state routing are both interior routing protocols. They can be used inside an autonomous system as intra-domain or intra-AS (as sometimes are called), but not between autonomous systems. Both of these routing protocols become intractable when the domain of operation becomes large.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

Path vector routing is exterior routing protocol proved to be useful for interdomain or inter-AS routing as it is sometimes called. In distance vector routing, a router has a list of networks that can be reached in the same AS with the corresponding cost (number of hops). In path vector routing, a router has a list of networks that can be reached with the path (list of ASs to pass) to reach each one. In other words, the domain of operation of the distance vector routing is a single AS; the domain of operation of the path vector routing is the whole Internet. The distance vector routing tells us the distance to each network; the path vector routing tells us the path.

Reachability

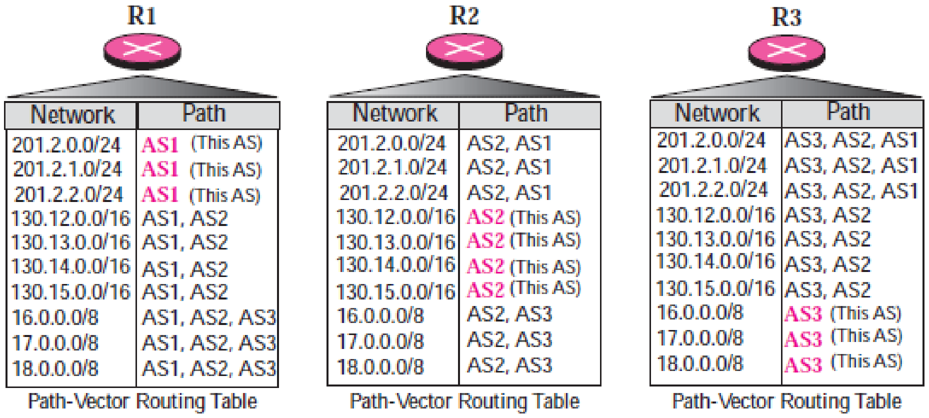
To be able to provide information to other ASs, each AS must have at least one path vector routing that collects reachability information about each network in that AS. The information collected in this case only means which network, identified by its network address (CIDR prefix), exists (can be reached in this AS). In other words, the AS needs to have a list of existing networks in its territory.

Each distance vector (exterior router) has created a list which shows which network is reachable in that AS.



Routing Tables

A path vector routing table for each router can be created if ASs share their reachability list with each other. In Figure 11.50, router R1 in AS1 can send its reachability list to router R2. Router R2, after combining its reachability list, can send the result to both R1 and R3. Router R3 can send its reachability list to R2, which in turn improves its routing table, and so on.



R1

Network	Path
201.2.0.0/24	AS1 (This AS)
201.2.1.0/24	AS1 (This AS)
201.2.2.0/24	AS1 (This AS)
130.12.0.0/16	AS1, AS2
130.13.0.0/16	AS1, AS2
130.14.0.0/16	AS1, AS2
130.15.0.0/16	AS1, AS2
16.0.0.0/8	AS1, AS2, AS3
17.0.0.0/8	AS1, AS2, AS3
18.0.0.0/8	AS1, AS2, AS3

Path-Vector Routing Table

R2

Network	Path
201.2.0.0/24	AS2, AS1
201.2.1.0/24	AS2, AS1
201.2.2.0/24	AS2, AS1
130.12.0.0/16	AS2 (This AS)
130.13.0.0/16	AS2 (This AS)
130.14.0.0/16	AS2 (This AS)
130.15.0.0/16	AS2 (This AS)
16.0.0.0/8	AS2, AS3
17.0.0.0/8	AS2, AS3
18.0.0.0/8	AS2, AS3

Path-Vector Routing Table

R3

Network	Path
201.2.0.0/24	AS3, AS2, AS1
201.2.1.0/24	AS3, AS2, AS1
201.2.2.0/24	AS3, AS2, AS1
130.12.0.0/16	AS3, AS2
130.13.0.0/16	AS3, AS2
130.14.0.0/16	AS3, AS2
130.15.0.0/16	AS3, AS2
16.0.0.0/8	AS3 (This AS)
17.0.0.0/8	AS3 (This AS)
18.0.0.0/8	AS3 (This AS)

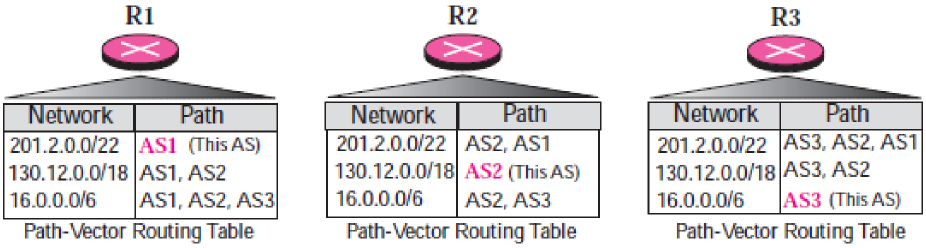
Path-Vector Routing Table

Loop Prevention

The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a reachability information, it checks to see if its autonomous system is in the path list to any destination. If it is, looping is involved and that network-path pair is discarded.

Aggregation

The path vector routing protocols normally support CIDR notation and the aggregation of addresses (if possible). This helps to make the path vector routing table simpler and exchange between routers faster.



Network	Path
201.2.0.0/22	AS1 (This AS)
130.12.0.0/18	AS1, AS2
16.0.0.0/6	AS1, AS2, AS3

Path-Vector Routing Table

Network	Path
201.2.0.0/22	AS2, AS1
130.12.0.0/18	AS2 (This AS)
16.0.0.0/6	AS2, AS3

Path-Vector Routing Table

Network	Path
201.2.0.0/22	AS3, AS2, AS1
130.12.0.0/18	AS3, AS2
16.0.0.0/6	AS3 (This AS)

Path-Vector Routing Table

Policy Routing

Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

BGP

Border Gateway Protocol (BGP) is an interdomain routing protocol using path vector routing.

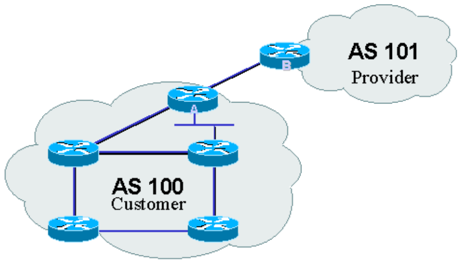
Types of Autonomous Systems

We can divide autonomous systems into three categories: stub, multihomed, and transit.

Stub AS

A stub AS has only one connection to another AS. The interdomain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink.

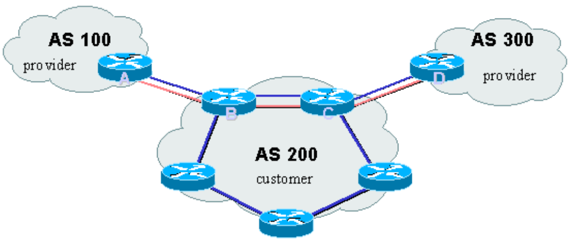
Stub AS



Multihomed AS

A multihomed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS. It can send data traffic to more than one AS, but there is no transient traffic. It does not allow data coming from one AS and going to another AS to pass through.

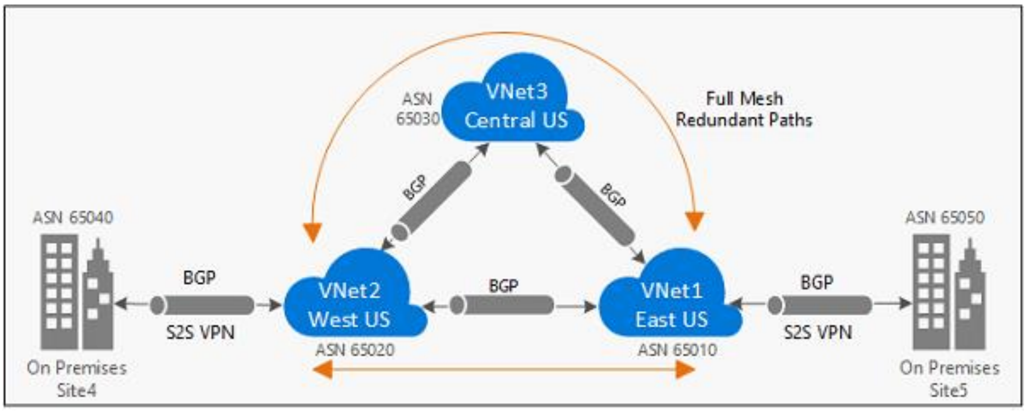
Multi-homed AS



- More details on multihoming coming up…

Transit AS

A transit AS is a multihomed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs (Internet backbones).



CIDR

BGP uses classless interdomain routing addresses. In other words, BGP uses a prefix to define a destination address. The address and the number of bits (prefix length) are used in updating messages.

Path Attributes

Attributes are divided into two broad categories: well-known and optional. A well-known attribute is one that every BGP router must recognize. An optional attribute is one that needs not be recognized by every router.

Well-known attributes are themselves divided into two categories: mandatory and discretionary. A well-known mandatory attribute is one that must appear in the description of a route. A well-known discretionary attribute is one that must be recognized by each router, but is not required to be included in every update message.

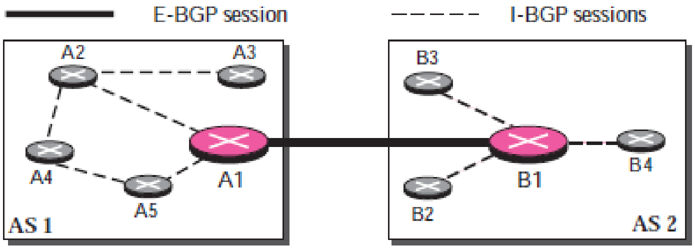
The optional attributes can also be subdivided into two categories: transitive and nontransitive. An optional transitive attribute is one that must be passed to the next router by the router that has not implemented this attribute. An optional nontransitive attribute is one that must be discarded if the receiving router has not implemented it.

BGP Sessions

The exchange of routing information between two routers using BGP takes place in a session. A session is a connection that is established between two BGP routers only for the sake of exchanging routing information. To create a reliable environment, BGP uses the services of TCP. BGP sessions are sometimes referred to as semipermanent connections.

External and Internal BGP

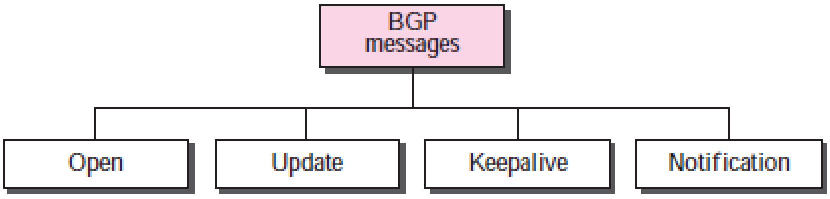
If we want to be precise, BGP can have two types of sessions: external BGP (E-BGP) and internal BGP (I-BGP) sessions. The E-BGP session is used to exchange information between two



speaker nodes belonging to two different autonomous systems. The IBGP session, on the other hand, is used to exchange routing information between two routers inside an autonomous system.

Types of Packets

BGP uses four different types of messages: open, update, keepalive, and notification.



Packet Format

All BGP packets share the same common header. Before studying the different types of packets, let us talk about this common header. The fields of this header are as follows:

Marker. The 16-byte marker field is reserved for authentication.

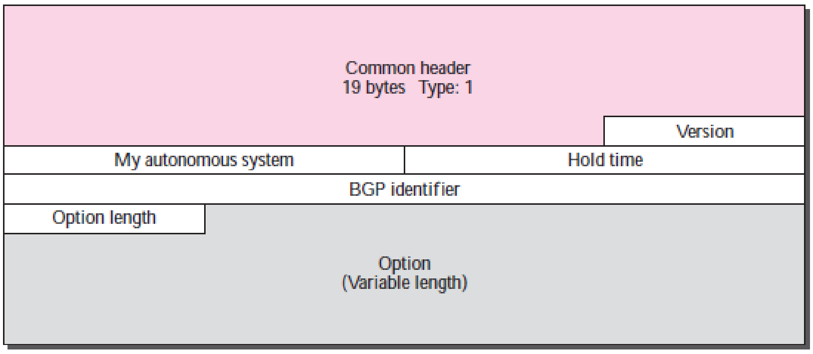
Length. This 2-byte field defines the length of the total message including the header.

Type. This 1-byte field defines the type of the packet. As we said before, we have four types, and the values 1 to 4 define those types.

Open Message

To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message. If the neighbor accepts the neighborhood relationship, it responds with a keepalive message, which means that a relationship has been established between the two routers.

Figure 11.56 *Open message*



Version. This 1-byte field defines the version of BGP. The current version is 4.

My autonomous system. This 2-byte field defines the autonomous system number.

Hold time. This 2-byte field defines the maximum number of seconds that can elapse until one of the parties receives a keepalive or update message from the other.

BGP identifier. This 4-byte field defines the router that sends the open message. The router usually uses one of its IP addresses (because it is unique) for this purpose.

Option length. The open message may contain some option parameters. In this case, this 1-byte field defines the length of the total option parameters.

Option parameters. If the value of the option parameter length is not zero, it means that there are some option parameters. Each option parameter itself has two subfields: the length of the parameter and the parameter value.

Update Message

The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, announce a route to a new destination, or both. Note that BGP can withdraw several destinations that were advertised before, but it can only advertise one new destination in a single update message.

Unfeasible routes length. This 2-byte field defines the length of the next field.

Withdrawn routes. This field lists all the routes that must be deleted from the previously advertised list.

Path attributes length. This 2-byte field defines the length of the next field.

Path attributes. This field defines the attributes of the path (route) to the network whose reachability is being announced in this message.

Network layer reachability information (NLRI). This field defines the network that is actually advertised by this message.

Keepalive Message

The routers (called peers in BGP parlance) running the BGP protocols exchange keepalive messages regularly (before their hold time expires) to tell each other that they are alive.

Notification Message

A notification message is sent by a router whenever an error condition is detected or a router wants to close the connection.

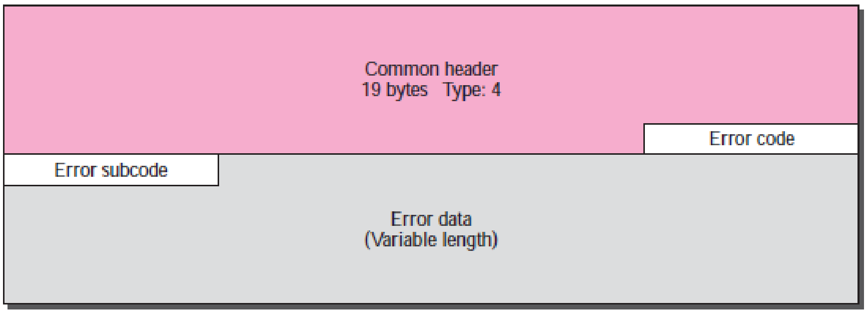
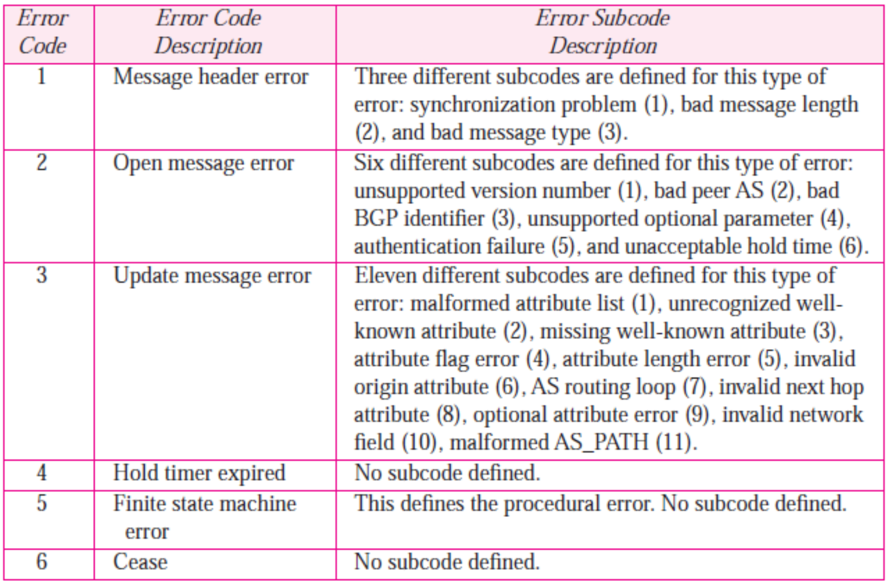


Table 11.6 *Error Codes*



Error Code	*Error Code Description*	*Error Subcode Description*
1	Message header error	Three different subcodes are defined for this type of error: synchronization problem (1), bad message length (2), and bad message type (3).
2	Open message error	Six different subcodes are defined for this type of error: unsupported version number (1), bad peer AS (2), bad BGP identifier (3), unsupported optional parameter (4), authentication failure (5), and unacceptable hold time (6).
3	Update message error	Eleven different subcodes are defined for this type of error: malformed attribute list (1), unrecognized well-known attribute (2), missing well-known attribute (3), attribute flag error (4), attribute length error (5), invalid origin attribute (6), AS routing loop (7), invalid next hop attribute (8), optional attribute error (9), invalid network field (10), malformed AS_PATH (11).
4	Hold timer expired	No subcode defined.
5	Finite state machine error	This defines the procedural error. No subcode defined.
6	Cease	No subcode defined.

Encapsulation

BGP messages are encapsulated in TCP segments using the well-known port 179. This means that there is no need for error control and flow control.