# Microsoft® Windows® Server 2003 TCP/IP Protocols and Services Technical Reference



| | |
|---|---|
| Author | Joseph Davies and Thomas Lee |
| Pages | 768 |
| Disk | 1 Companion CD(s) |
| Level | Int/Adv |
| Published | 02/26/2003 |
| ISBN | 0-7356-1291-9 |
| Price | ~~$49.99~~ |

To see this book's discounted price, select a reseller below.


# Chapter 6: Internet Protocol (IP) Addressing



---

# Chapter 6 Internet Protocol (IP) Addressing

To successfully administer and troubleshoot IP internetworks, it is important to understand all aspects of IP addressing. One of the most important aspects of TCP/IP network administration is the assignment of unique and proper IP addresses to all the nodes of an IP internetwork. Although the concept of IP address assignment is simple, the actual mechanics of efficient IP address allocation using subnetting techniques are somewhat complicated. Additionally, it is important to understand the role of IP broadcast and multicast traffic and how these addresses map to Network Interface Layer addresses such as Ethernet and Token Ring media access control (MAC) addresses.

## Types of IP Addresses

An IP address is a 32-bit logical address that can be one of the following types:

- **Unicast** A unicast IP address is assigned to a single network interface attached to an IP internetwork. Unicast IP addresses are used in one-to-one communications.
- **Broadcast** A broadcast IP address is designed to be processed by every IP node on the same network segment. Broadcast IP addresses are used in one-to-everyone communications.
- **Multicast** An IP multicast address is an address on which one or multiple nodes can be listening on the same or different network segments. IP multicast addresses are used in one-to-many communications.

## Expressing IP Addresses

The IP address is a 32-bit value that computers are adept at manipulating. Humans, however, do not think in binary mode, 32 bits at a time. Because most humans are trained in the use of decimal (base 10 numbering system) rather than binary (base 2 numbering system), it is common to express IP addresses in a decimal form.

The 32-bit IP address is divided from the high-order bit to the low-order bit into four 8-bit quantities called *octets*. IP addresses are normally written as four separate decimal octets delimited by a period (a dot). This is known as dotted decimal notation.

For example, the IP address 00001010000000011111000101000011 is subdivided into four octets:

00001010 00000001 11110001 01000011

Each octet is converted to a base 10 number and separated from the others by periods:

10.1.241.67

A generalized IP address is indicated with *w.x.y.z,* as Figure 6-1 shows.

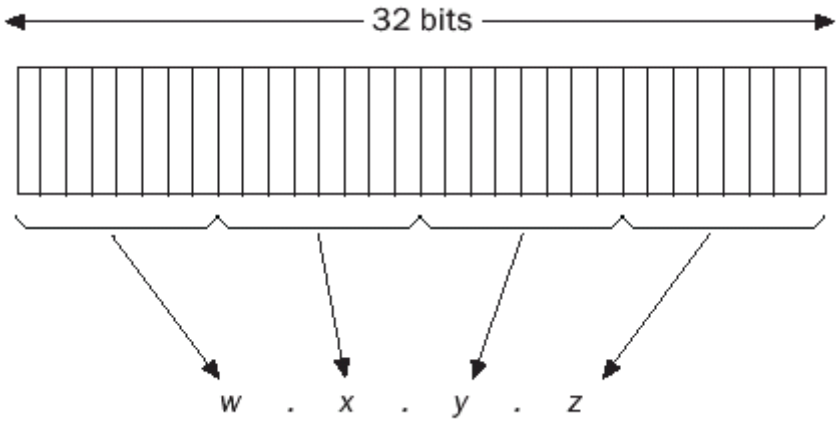


**Figure 6-1.** *The generalized IP address consisting of 32 bits expressed in dotted decimal notation.*

**Converting from Binary to Decimal**

*To convert a binary number to its decimal equivalent, add the numbers represented by the bit positions that are set to 1. Figure 6-2 shows an 8-bit number and the decimal value of each position.*

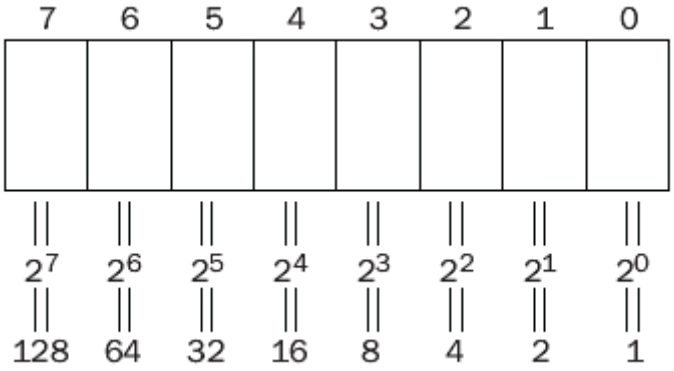


***Figure 6-2.*** *An 8-bit number showing bit positions and their decimal equivalents.*

For example, the 8-bit binary number 01000011 is 67 (64 + 2 + 1). The maximum number that can be expressed with an 8-bit number (11111111) is 255 (128 + 64 + 32 + 16 + 8 + 4 + 2 + 1).

## *Converting from Decimal to Binary*

To convert from decimal to binary, analyze the decimal number to see if it contains the quantities represented by the bit positions from the high-order bit to the low-order bit. Starting from the high-order bit quantity (128), if each quantity is present, the bit in that bit position is set to 1. For example, the decimal number 211 contains 128, 64, 16, 2, and 1. Therefore, 211 is 11010011 in binary notation.

## *IP Addresses in the IP Header*

IP addresses are used in the IP header's Source Address and Destination Address fields.

- The IP header's Source Address field is always either a unicast address or the special address 0.0.0.0. The unspecified IP address, 0.0.0.0, is used only when the IP node is not configured with an IP address and the node is attempting to obtain an address through a configuration protocol such as Dynamic Host Configuration Protocol (DHCP).
- The IP header's Destination Address field is a unicast address, multicast address, or broadcast address.

# Unicast IP Addresses

Each network interface on which TCP/IP is active must be identified by a unique, logical, unicast IP address. The unicast IP address is a logical address because it is an Internet Layer address that has no direct relation to the address being used at the Network Interface Layer. For example, the unicast IP address assigned to a host on an Ethernet network has no relation to the 48-bit MAC address used by the Ethernet network adapter.

The unicast IP address is an internetwork address for IP nodes that contains a network ID and a host ID.

- The network ID, or network address, identifies the nodes that are located on the same logical network. In most cases, a logical network is the same as a physical network segment with boundaries that are defined by IP routers. In some cases, multiple logical networks exist on the same physical network using a practice called *multinetting*. All nodes on the same logical network share the same network ID. If all nodes on the same logical network are not configured with the same network ID, routing or delivery problems occur. The network ID must be unique to the internetwork.
- The host ID, or host address, identifies a node within a network. A node is a router or host (a nonrouter interface such as a workstation, server, or other TCP/IP-based system). The host ID must be unique within each network segment.

**NOTE**

---

The term *network ID* applies to class-based network IDs, subnetted network IDs, and classless network IDs.

*Figure 6-3 is an example of a unicast IP address and its network ID and host ID portions.*

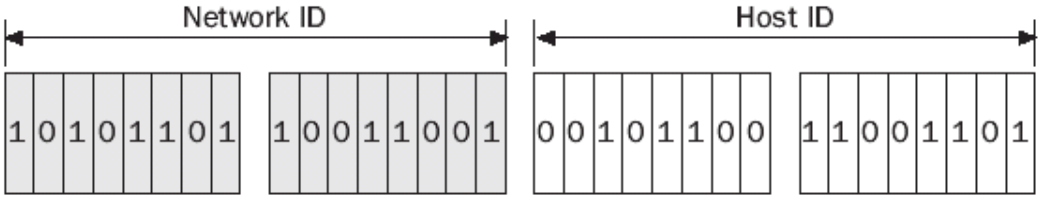


**Figure 6-3.** *The structure of an example IP address showing the network ID and host ID.*

**A History Lesson: IP Address Classes**

This section is called "A History Lesson" because modern networks are not based on the Internet address classes. Because of the Internet's recent rapid expansion, the Internet authorities saw clearly that the original class-based structure did not scale well to the size of a global internetwork. For example, if the Internet authorities were still handing out class-based addresses, there would be hundreds of thousands of routes in the routing tables of Internet backbone routers. To prevent this scaling problem, addressing on the modern Internet is classless. However, the understanding of Internet address classes is an important element in understanding IP addressing.

RFC 791 defined the unicast IP address in terms of address classes to create well-defined networks of various sizes. The design goal was to create the following:

- A small number of large networks (networks with a large amount of nodes)
- A moderate number of moderate-sized networks
- A large number of small networks

The result was the creation of address classes, subdivisions of the 32-bit IP address space defined by setting high-order bits and dividing the remaining bits into network ID and host ID.

**Class A**

Class A addresses are designed for networks with a large number of hosts. The high-order bit is set to 0. The first 8 bits (the first octet) are defined as the network ID; the last 24 bits (the last three octets) are defined as the host ID. Figure 6-4 illustrates the class A address.

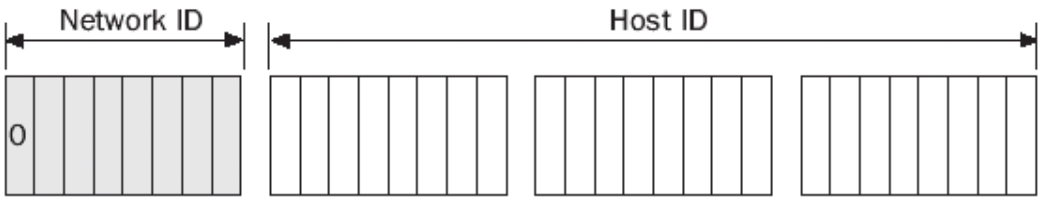


**Figure 6-4.** *The class A address showing the network ID and the host ID.*

**Class B**

Class B addresses are designed for moderate-sized networks with a moderate number of hosts. The two high-order bits are set to 10. The first 16 bits (the first two octets) are defined as the network ID; the last 16 bits (the last two octets) are defined as the host ID. Figure 6-5 illustrates the class B address.

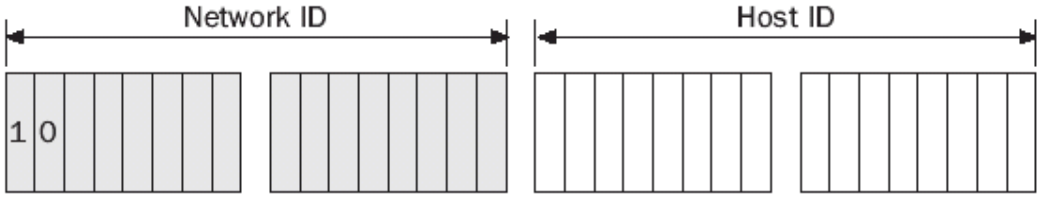


**Figure 6-5.** *The class B address showing the network ID and the host ID.*

**Class C**

Class C addresses are designed for small networks with a small number of hosts. The three high-order bits are set to 110. The first 24 bits (the first three octets) are defined as the network ID; the last 8 bits (the last three octets) are defined as the host ID. Figure 6-6 illustrates the class C address.

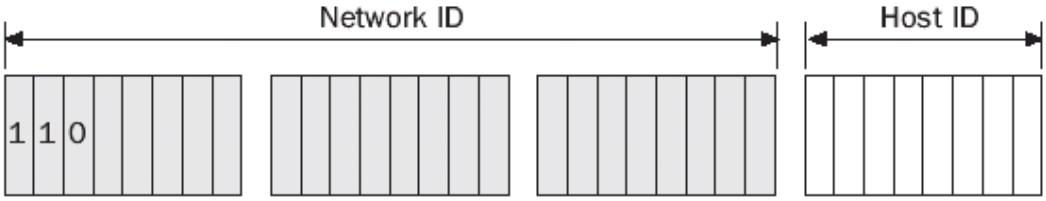


**Figure 6-6.** *The class C address showing the network ID and the host ID.*

**Additional Address Classes**

Class D and E addresses are defined, in addition to unicast address classes A, B, and C.

**Class D**

Class D addresses are for IP multicast addresses. The four high-order bits are set to binary 1110. The next 28 bits are used for individual IP multicast addresses. For more information on IP multicast addresses, see the section entitled "IP Multicast Addresses," later in this chapter. The Microsoft Windows Server 2003 family supports class D addresses for IP multicast traffic.

**Class E**

Class E addresses are experimental addresses reserved for future use. The five high-order bits in a class E address are set to 11110. The Windows Server 2003 family does not support the use of class E addresses.

## *Rules for Enumerating Network IDs*

When enumerating IP network IDs, the following rules apply:

- **The network ID cannot begin with 127 as the first octet.** All 127.*x.y.z* addresses are reserved as loopback addresses.
- **All the bits in the network ID cannot be set to 1.** Network IDs set to all 1s are reserved for broadcast addresses.
- **All the bits in the network ID cannot be set to 0.** Network IDs set to all 0s are reserved for indicating a host on the local network.
- **The network ID must be unique to the IP internetwork.**

*Table 6-1 lists the ranges of network IDs based on the IP address classes. Network IDs are expressed by setting all host bits to 0 and expressing the result in dotted decimal notation.*

***Table 6-1.*** *Address Class Ranges of Network Ids*

| Address Class | First Network ID | Last Network ID | Number of Networks |
|---|---|---|---|
| Class A | 1.0.0.0 | 126.0.0.0 | 126 |
| Class B | 128.0.0.0 | 191.255.0.0 | 16,384 |
| Class C | 192.0.0.0 | 223.255.255.0 | 2,097,152 |

**NOTE**

IP network IDs, even though expressed in dotted decimal notation, are not IP addresses assigned to network interfaces. The IP network ID is the network address that is common for all network interfaces attached to the same logical network.

## *Rules for Enumerating Host Ids*

When enumerating IP host IDs, the following rules apply:

- **All bits in the host ID cannot be set to 1.** Host IDs set to all 1s are reserved for broadcast addresses.
- **All the bits in the host ID cannot be set to 0.** Host IDs set to all 0s are reserved for the expression of IP network IDs.
- **The host ID must be unique to the network.**

Table 6-2 lists the ranges of host IDs based on the IP address classes.

***Table 6-2.*** *Address Class Ranges of Host Ids*

| Address Class | First Host ID | Last Host ID | Number of Hosts |
|---|---|---|---|
| Class A | *w*.0.0.1 | *w*.255.255.254 | 16,777,214 |
| Class B | *w.x*.0.1 | *w.x*.255.254 | 65,534 |
| Class C | *w.x.y*.1 | *w.x.y*.254 | 254 |

## *Subnets and the Subnet Mask*

Subnetting is designed to make more efficient use of a fixed address space, namely, an IP network ID. The network bits are fixed and the host bits are variable. Originally, the host bits were designed to indicate host IDs within an IP network ID. With subnetting, host ID bits can be used to express a combination of a subnetwork ID and a subnetwork host ID, thereby better utilizing the host bits.

Consider a class B network that has 65,534 possible hosts. A network segment of 65,534 hosts is technically possible but impractical because of the accumulation of broadcast traffic. All nodes on the same physical network segment belong to the same broadcast domain and share the same broadcast traffic. Because making all 65,534 hosts share the same broadcast traffic is not a practical configuration, most of the host IDs are not usable.

To create smaller broadcast domains and make better use of the host bits, RFC 950 defines a method of subdividing a network ID into subnetworks—subsets of the original class-based network—by using bits in the host ID portion of the original IP network ID. Each subnetwork, or subnet, is assigned a new subnetted network ID. Hosts on subnets are assigned host IDs from the remaining host bits in the subnetted network ID.

Although RFC 950 discusses subnetting in terms of class-based network IDs, subnetting is a general technique that can be used on classless network IDs or used recursively on subnetted network IDs. This is described in the section entitled "Variable-Length Subnetting," later in this chapter.

The proper subnetting of a network ID is transparent to the rest of the IP internetwork. For example, consider the class B network ID of 131.107.0.0 (shown in Figure 6-7), which is connected to the Internet. The class-based network ID was obtained from the InterNIC and is a fixed address space. Because this class B network ID represents an impractical broadcast domain, it is subnetted. However, in subnetting 131.107.0.0, we should not require any reconfiguration of the Internet routers.

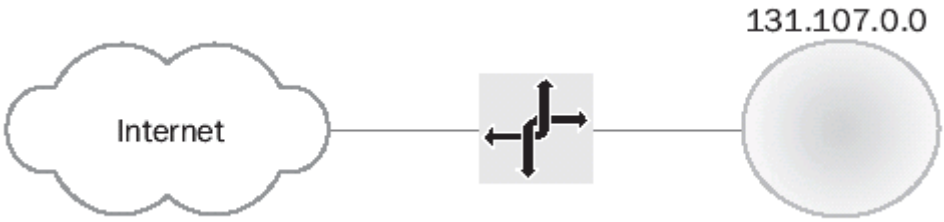


**Figure 6-7.** *The class B network 131.107.0.0 before subnetting.*

From an analysis of broadcast traffic, it is determined that there should be no more than 250 nodes on each broadcast domain. Therefore, network ID 131.107.0.0 is subnetted to look like a class C address by using the first 8 high-order host bits (the third octet represented by *y*) for the subnetted network ID. Note that before the subnetting, only the first two octets are considered the network ID. After the subnetting, the first three octets are considered the network ID. The new network IDs are 131.107.1.0, 131.107.2.0, and 131.107.3.0, as Figure 6-8 shows.

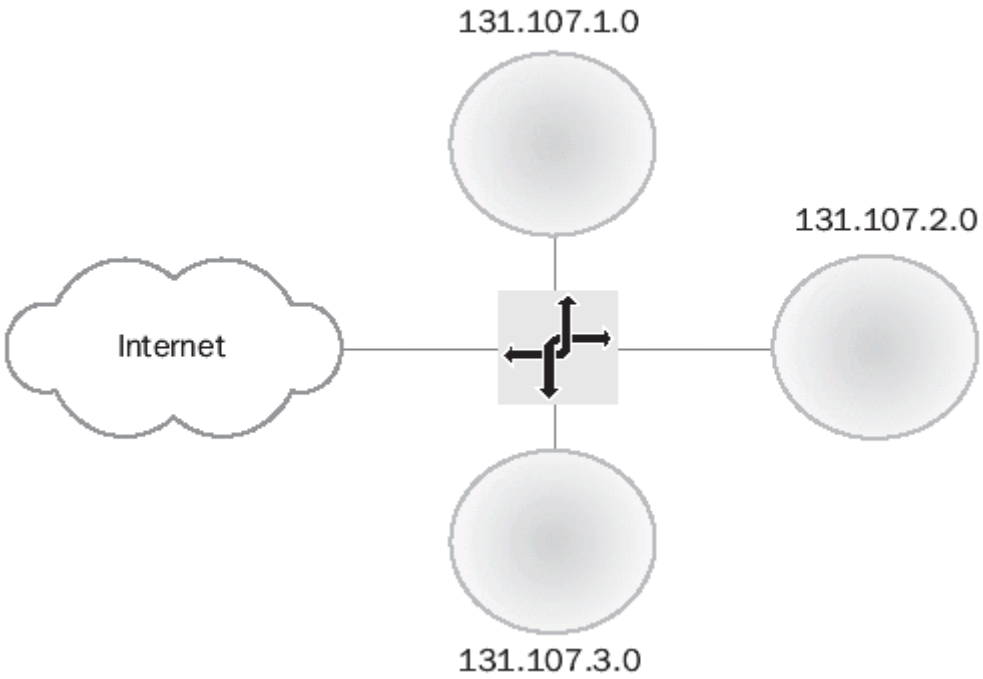


**Figure 6-8.** *The class B network 131.107.0.0 after subnetting.*

The IP router connected to the Internet has an interface on each of the subnets and is aware of the new subnetting scheme. The IP router forwards IP datagrams from the Internet to the host on the appropriate subnet. The Internet routers are completely unaware of the subnetting of 131.107.0.0. They still consider all IP addresses in the range of 131.107.0.0 through 131.107.255.255 to be reachable through the IP router's Internet interface.

**The Subnet Mask**

With subnetting, a host or router can no longer assume the network ID and host ID designations of the IP address classes. The node needs additional configuration to distinguish the network ID and host ID portions of an IP address, whether the network ID is class-based, classless, or a subnetted network ID.

RFC 950 defines the use of a bit mask to identify which bits in the IP address belong to the network ID and which belong to the host ID. This bit mask, called a *subnet mask* or *address mask*, is defined by the following:

- If the bit position corresponds to a bit in the network ID, it is set to 1.
- If the bit position corresponds to a bit in the host ID, it is set to 0.

Since the publication of RFC 950, TCP/IP nodes require a subnet mask to be configured for each IP address, even when class-based addressing is used. A default subnet mask corresponds to a class-based network ID. A custom subnet mask corresponds to either a classless network ID or a subnetted network ID. The subnet mask is the definitive piece of configuration information that allows the node to determine its own network ID.

**Subnet Masks in Dotted Decimal Representation**

Frequently, the subnet mask is expressed in dotted decimal notation. Although expressed in the same form as an IP address, the subnet mask is not an IP address. As an example of subnet masks in dotted decimal notation, default subnet masks are based on the IP address classes. Table 6-3 lists the default subnet masks for class A, B, and C network IDs in dotted decimal notation.

**Table 6-3.** *Dotted Decimal Notation for Default Subnet Masks*

| Address Class | Bits for Subnet Mask | Subnet Mask |
|---|---|---|
| Class A | 11111111 00000000 00000000 00000000 | 255.0.0.0 |
| Class B | 11111111 11111111 00000000 00000000 | 255.255.0.0 |
| Class C | 11111111 11111111 11111111 00000000 | 255.255.255.0 |

A custom subnet mask is used whenever you perform nonclassful addressing. In our earlier example, the classful network ID 131.107.0.0 is subnetted by using the third octet for subnets. The subnetted network ID 131.107.1.0 no longer uses the default subnet mask 255.255.0.0. To express the third octet as part of the network ID, the custom subnet mask 255.255.255.0 is used.

The subnetted network ID and its corresponding subnet mask are expressed in dotted decimal notation as 131.107.1.0, 255.255.255.0.

**Network Prefix Length Representation of Subnet Masks**

Although it is technically possible to subnet IP network IDs by choosing host bits in a noncontiguous fashion, it is impractical and mathematically challenging to enumerate the subnetted network IDs and the host IDs per subnet. For this reason, you must subnet by choosing host bits in a contiguous fashion from the high-order host bit.

Because the network ID bits are always contiguous starting from the highest order bit, an easier and more compact way of expressing the subnet mask is to indicate the number of network ID bits using network prefix notation, or Classless Inter-Domain Routing (CIDR) notation. Network prefix notation views the IP address in terms of the prefix (the network ID) and the suffix (the host ID). Network prefix notation is:

*/# of bits in the network ID*

Network prefix notation is commonly used with TCP/IP implementations other than the Windows Server 2003 family, and it is an important notation to understand looking forward to IP version 6 (IPv6).

*Table 6-4 lists the equivalent subnet mask in network prefix notation for the IP address classes.*

***Table 6-4.*** *Network Prefix Notation for Default Subnet Masks*

| Address Class | Bits for Subnet Mask | Network Prefix |
|---|---|---|
| Class A | 11111111 00000000 00000000 00000000 | /8 |
| Class B | 11111111 11111111 00000000 00000000 | /16 |
| Class C | 11111111 11111111 11111111 00000000 | /24 |

In our earlier example, the classful network ID 131.107.0.0, with the subnet mask of 255.255.0.0, is expressed in network prefix notation as 131.107.0.0/16. If 131.107.0.0 were subnetted by using the third octet to express subnets, a total of 24 contiguous bits would be used for the subnetted network ID. The subnetted network ID 131.107.1.0 and its corresponding subnet mask are expressed in network prefix notation as 131.107.1.0/24.

**Expressing Network IDs**

The fixed network ID bits and the subnet mask define the network ID. Therefore, network IDs must always be expressed by the combination of the network ID and a subnet mask. Expressing a network ID without its subnet mask is ambiguous. For example, for the network ID 10.16.0.0, which bits are used for the network ID? The first 16? The first 24? The first 12?

The following are examples of properly expressed network IDs:

- 192.168.45.0, 255.255.255.0
- 10.99.0.0/16

All hosts on the same logical network must be using the same network ID bits and the same subnet mask. For example, 131.107.0.0/16 is not the same as 131.107.0.0/24. For the network ID 131.107.0.0/16, the usable IP addresses range from 131.107.0.1 through 131.107.255.254. For the network ID 131.107.0.0/24, the usable IP addresses range from 131.107.0.1 through 131.107.0.254. Clearly, 131.107.0.0/16 and 131.107.0.0/24 do not represent the same group of hosts.

**Determining the Network ID**

In earlier examples, classful network IDs and subnetted network IDs all fell along octet boundaries where it was easy to determine the network ID and host ID portion of the IP address. However, real-world subnetting is not always done along octet boundaries. For example, some network administrators might determine that, for their situation, they need only three host bits for subnetting.

Because subnetting can occur along nonoctet boundaries, there must be a method of determining the network ID from an IP address with an arbitrary subnet mask. IP uses a method called a *bit-wise logical AND* to extract the network ID.

Recall how the subnet mask is defined: 1 is used to indicate a network ID bit and 0 is used to indicate a host ID bit. In a logical AND comparison, the result is 1 when the value of the two bits being compared is 1. Otherwise, the result is 0. This comparison is done for all 32 bits of the IP address and subnet mask. The result of the bit-wise logical AND of the IP address and the subnet mask is the network ID.

For example, what is the network ID of the IP node 131.107.164.26 with a subnet mask of 255.255.240.0? To obtain the result in binary notation, convert both the IP address and subnet mask to binary. Then perform the logical AND comparison for each bit.

**IP address** 10000011 01101011 10100100 00011010

**Subnet mask** 11111111 11111111 11110000 00000000

**Network ID** 10000011 01101011 10100000 00000000

The result of the bit-wise logical AND of the 32 bits of the IP address and the subnet mask is the network ID 131.107.160.0 with the subnet mask of 255.255.240.0.

Notice the following:

- The bits in the network ID portion of the IP address are copied directly to the result. A value of 1 in the network ID portion of the IP address becomes a 1 in the result. A value of 0 in the network ID portion of the IP address becomes a 0 in the result.
- All bits in the host ID portion of the IP address are set to 0. Because the subnet mask uses a 0 for host ID bit positions, the logical AND comparison always yields a 0.

Therefore, because the bits in the network ID are copied and the bits in the host ID are set to 0, the result must be the network ID.

# *How to Subnet*

The act of subnetting a network ID is a relatively complex procedure; although there are numerous subnet calculators available, the ability to subnet is a vital skill for any TCP/IP network administrator.

Subnetting is done in two basic steps:

1. Based on your design requirements, decide how many host bits you need for the proper balance between number of subnets and number of hosts per subnet.
2. Based on the number of host bits chosen, enumerate the subnetted network IDs, including the ranges of usable IP addresses for each subnetted network ID. The actual mechanics of defining the subnetted network IDs can be done in binary or decimal notation.

There are two methods for the second step of subnetting, the enumeration of the subnetted network Ids:

- The binary method, in which the individual bits of the subnetted network IDs are manipulated and converted to dotted decimal notation, can be used to subnet. However, this method does not scale well to large numbers of subnets. It is described here primarily to illustrate the subnetting process in its most fundamental form.
- The decimal method, in which subnetted network IDs are derived from calculations on decimal numbers, scales well to large numbers of subnets and lends itself well to spreadsheets and programming code.

**Step 1: Determining the Number of Host Bits**

To determine the number of host bits required for subnetting, perform an analysis of your internetwork. You should determine the following:

- **The number of subnets needed both now and in the future** Be sure to plan for expansion. Subnetting an existing network requires reassigning IP addresses to IP interfaces. Although DHCP can ease this burden, routers and other fixed-address types of hosts might need to be manually reconfigured. Subnetting is not something you want to do often.
- **The maximum number of hosts needed on each subnet** This number depends on how many hosts you want sharing the same broadcast traffic. In most cases, when choosing between more subnets and more hosts per subnet, the practical choice is to choose more subnets.

There is an inverse relationship between the number of subnets and the number of hosts per subnet that can be supported by a given subnetting scheme. As Figure 6-9 illustrates, when you choose more host bits, the number of subnets goes up, but the number of hosts per subnet goes down by approximately a factor of 2.

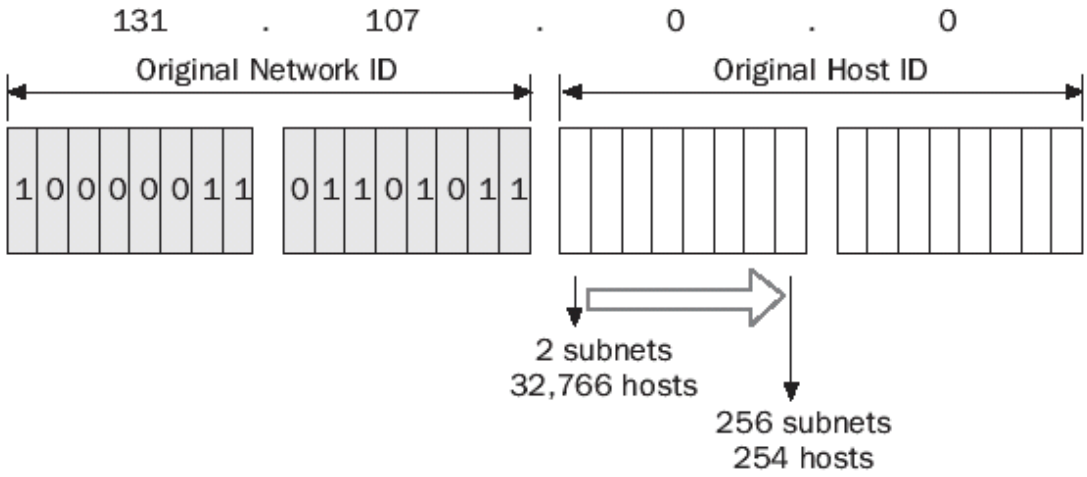


**Figure 6-9.** *The relationship between the number of subnets and hosts per subnet when subnetting the class B network ID 131.107.0.0.*

If we choose one host bit when subnetting the class B network ID 131.107.0.0, two subnets can be expressed, with 32,766 hosts per subnet. If we choose eight host bits, 256 subnets can be expressed with 254 hosts per subnet.

Determine how many subnets you need now, and plan for growth by estimating how many you will need in the next five years. Each physical network segment is a subnet. Point-to-point wide area network (WAN) connections such as leased lines might need subnetted network IDs, unless your routers support unnumbered connections. Non-broadcast multiple access (NBMA) WAN technologies such as frame relay need subnetted network IDs. Use additional host bits if the remaining host bits can express more hosts per subnet than you will need.

Subnetting always starts with a fixed address space in the form of a network ID. The network ID to be subnetted can be a classful network ID, a classless network ID (as allocated using CIDR), or a previously subnetted classful or classless network ID. The fixed address space contains a sequence of bits that are fixed (the network ID bits) and a sequence of bits that are variable (the host ID bits).

Based on your analysis of the desired number of subnets and number of hosts per subnet, a specific number of high-order host bits are converted from host bits into subnet bits. The combination of the original network ID bits and the converted host bits becomes the new subnetted network ID.

As you determine how many host bits you need, you determine the new subnet mask for your subnetted network IDs.

Tables 6-5, 6-6, and 6-7 list the subnetting of classful network IDs according to the requirement of a specific number of subnets. These tables can be useful when determining a subnetting scheme for a class-based network ID based on a required number of subnets and a desired number of hosts per subnet.

***Table 6-5.*** *Subnetting of a Class A Network ID*

| Required Number of Subnets | Number of Host Bits | Subnet Mask | Number of Hosts per Subnet |
|---|---|---|---|
| 1-2 | 1 | 255.128.0.0 or /9 | 8,388,606 |
| 3-4 | 2 | 255.192.0.0 or /10 | 4,194,302 |
| 5-8 | 3 | 255.224.0.0 or /11 | 2,097,150 |
| 9-16 | 4 | 255.240.0.0 or /12 | 1,048,574 |
| 17-32 | 5 | 255.248.0.0 or /13 | 524,286 |
| 33-64 | 6 | 255.252.0.0 or /14 | 262,142 |
| 65-128 | 7 | 255.254.0.0 or /15 | 131,070 |
| 129-256 | 8 | 255.255.0.0 or /16 | 65,534 |
| 257-512 | 9 | 255.255.128.0 or /17 | 32,766 |
| 513-1024 | 10 | 255.255.192.0 or /18 | 16,382 |
| 1025-2048 | 11 | 255.255.224.0 or /19 | 8190 |
| 2049-4096 | 12 | 255.255.240.0 or /20 | 4094 |
| 4097-8192 | 13 | 255.255.248.0 or /21 | 2046 |
| 8193-16,384 | 14 | 255.255.252.0 or /22 | 1022 |
| 16,385-32,768 | 15 | 255.255.254.0 or /23 | 510 |
| 32,769-65,536 | 16 | 255.255.255.0 or /24 | 254 |
| 65,537-131,072 | 17 | 255.255.255.128 or /25 | 126 |
| 131,073-262,144 | 18 | 255.255.255.192 or /26 | 62 |
| 262,145-524,288 | 19 | 255.255.255.224 or /27 | 30 |
| 524,289-1,048,576 | 20 | 255.255.255.240 or /28 | 14 |
| 1,048,577-2,097,152 | 21 | 255.255.255.248 or /29 | 6 |
| 2,097,153-4,194,304 | 22 | 255.255.255.252 or /30 | 2 |

**Table 6-6.** *Subnetting of a Class B Network ID*

| Required Number of Subnets | Number of Host Bits | Subnet Mask | Number of Hosts per Subnet |
|---|---|---|---|
| 1-2 | 1 | 255.255.128.0 or /17 | 32,766 |
| 3-4 | 2 | 255.255.192.0 or /18 | 16,382 |
| 5-8 | 3 | 255.255.224.0 or /19 | 8190 |
| 9-16 | 4 | 255.255.240.0 or /20 | 4094 |
| 17-32 | 5 | 255.255.248.0 or /21 | 2046 |
| 33-64 | 6 | 255.255.252.0 or /22 | 1022 |
| 65-128 | 7 | 255.255.254.0 or /23 | 510 |
| 129-256 | 8 | 255.255.255.0 or /24 | 254 |
| 257-512 | 9 | 255.255.255.128 or /25 | 126 |
| 513-1024 | 10 | 255.255.255.192 or /26 | 62 |
| 1025-2048 | 11 | 255.255.255.224 or /27 | 30 |
| 2049-4096 | 12 | 255.255.255.240 or /28 | 14 |
| 4097-8192 | 13 | 255.255.255.248 or /29 | 6 |
| 8193-16,384 | 14 | 255.255.255.252 or /30 | 2 |

**Table 6-7.** *Subnetting of a Class C Network ID*

| Required Number of Subnets | Number of Host Bits | Subnet Mask | Number of Hosts per Subnet |
|---|---|---|---|
| 1-2 | 1 | 255.255.255.128 or /25 | 126 |
| 3-4 | 2 | 255.255.255.192 or /26 | 62 |
| 5-8 | 3 | 255.255.255.224 or /27 | 30 |
| 9-16 | 4 | 255.255.255.240 or /28 | 14 |
| 17-32 | 5 | 255.255.255.248 or /29 | 6 |
| 33-64 | 6 | 255.255.255.252 or /30 | 2 |

**Step 2: Defining the Subnetted Network IDs (Binary Method)**

The technique presented here describes how to subnet an arbitrary network ID into subnets that yield both subnetted network IDs and their corresponding range of valid IP addresses using binary analysis. There are other techniques that might seem easier, but they are typically limited in scope. This technique works for any subnetting situation.

**Step 2a: Enumerating the Subnetted Network IDs (Binary)**

Create a three-column table with $2^n$ rows where *n* is the number of host bits chosen for the subnetting. The first column is used for the subnet number, the second column is for the binary representation of the subnetted network ID, and the third column is for the dotted decimal representation of the subnetted network ID.

For the binary representation for each entry in the table, the original network ID bits are fixed at their original values. The host bits chosen for subnetting, hereafter known as the subnet bits, are allowed to vary over all of their possible values, and the remaining host bits are set to 0.

The table's first entry is the subnet, defined by setting all the subnet bits to 0 (also called the all-zeros subnet). The result is converted to dotted decimal notation. This subnetted network ID does not appear to be different from the original network ID; but remember that a network ID is a combination of the dotted decimal notation and a subnet mask. With the new subnet mask, the subnetted network ID is clearly different from the original network ID.

In the following entries, treat the subnet bits as though they were distinct binary numbers. Increment the value within the subnet bits and convert the result of the entire 32-bit subnetted network ID to dotted decimal notation.

As an example of this technique, subnet the class B network ID 131.107.0.0 by using three bits of the classful host ID. The new subnet mask for the subnetted network IDs is 255.255.224.0, or /19. Based on using three host bits, create a table with eight entries ($8 = 2^3$). The first entry is the all-zeros subnet. The additional entries are increments of the binary number represented by the subnet bits (underlined). Table 6-8 lists the subnetted network IDs.

**Table 6-8.** ***A 3-Bit Subnetting of 131.107.0.0 (Binary)***

| Subnet | Binary Representation | Subnetted Network ID |
|---|---|---|
| 1 | 10000011.01101011.000<u>00000</u>.00000000 | 131.107.0.0/19 |
| 2 | 10000011.01101011.001<u>00000</u>.00000000 | 131.107.32.0/19 |
| 3 | 10000011.01101011.010<u>00000</u>.00000000 | 131.107.64.0/19 |
| 4 | 10000011.01101011.011<u>00000</u>.00000000 | 131.107.96.0/19 |
| 5 | 10000011.01101011.100<u>00000</u>.00000000 | 131.107.128.0/19 |
| 6 | 10000011.01101011.101<u>00000</u>.00000000 | 131.107.160.0/19 |
| 7 | 10000011.01101011.110<u>00000</u>.00000000 | 131.107.192.0/19 |
| 8 | 10000011.01101011.111<u>00000</u>.00000000 | 131.107.224.0/19 |

**Step 2b: Enumerating IP Address Ranges for Each Subnetted Network ID (Binary)**

For each subnetted network ID, the range of valid IP addresses must be determined as follows:

1. Create a three-column table with $2^n$ entries where *n* is the number of host bits chosen for the subnetting. The first column is used for the subnet number, the second column is for the binary representation of the first and last IP address in the range, and the third column is for the dotted decimal representation of the first and last IP address in the range. Alternately, you can extend the table created for enumerating the subnetted network IDs by adding two columns.
2. Express the first and last IP address in the range in binary notation. The first IP address is defined by setting the remaining host bits to 0, except for the last host bit. The last IP address is defined by setting the remaining host bits to 1, except for the last host bit.
3. Convert the binary representation of the first and last IP address to dotted decimal notation.
4. Repeat steps 2 and 3 until the table is complete.

To continue our example, Table 6-9 lists the enumeration of the range of valid IP addresses for the 3-bit subnetting of 131.107.0.0. The host bits are underlined.

**Table 6-9.** *Enumeration of IP Addresses for the 3-Bit Subnetting of 131.107.0.0 (Binary)*

| Subnet | Binary Representation | Range of IP Addresses |
|---|---|---|
| 1 | 10000011.01101011.000<u>00000</u>.<u>00000001</u> - 10000011.01101011.000<u>11111</u>.<u>11111110</u> | 131.107.0.1 - 131.107.31.254 |
| 2 | 10000011.01101011.001<u>00000</u>.<u>00000001</u> - 10000011.01101011.001<u>11111</u>.<u>11111110</u> | 131.107.32.1 - 131.107.63.254 |
| 3 | 10000011.01101011.010<u>00000</u>.<u>00000001</u> - 10000011.01101011.010<u>11111</u>.<u>11111110</u> | 131.107.64.1 - 131.107.95.254 |
| 4 | 10000011.01101011.011<u>00000</u>.<u>00000001</u> - 10000011.01101011.011<u>11111</u>.<u>11111110</u> | 131.107.96.1 - 131.107.127.254 |
| 5 | 10000011.01101011.100<u>00000</u>.<u>00000001</u> - 10000011.01101011.100<u>11111</u>.<u>11111110</u> | 131.107.128.1 - 131.107.159.254 |
| 6 | 10000011.01101011.101<u>00000</u>.<u>00000001</u> - 10000011.01101011.101<u>11111</u>.<u>11111110</u> | 131.107.160.1 - 131.107.191.254 |
| 7 | 10000011.01101011.110<u>00000</u>.<u>00000001</u> - 10000011.01101011.110<u>11111</u>.<u>11111110</u> | 131.107.192.1 - 131.107.223.254 |
| 8 | 10000011.01101011.111<u>00000</u>.<u>00000001</u> - 10000011.01101011.111<u>11111</u>.<u>11111110</u> | 131.107.224.1 - 131.107.255.254 |

**Step 2: Defining the Subnetted Network IDs (Decimal Method)**

The previous technique describes a subnetting technique using binary. Although this method works for any valid subnetting scheme, it does not scale well. For example, if you were performing a 10-bit subnetting, you would have 1024 entries in the table. Whereas programmers are adept at binary manipulation and can create programs to automate this process, nonprogrammers find it easier to work with decimal numbers. Therefore, the following technique treats the 32-bit network ID and IP address as a single decimal number to enumerate the subnetted network ID and its corresponding range of IP addresses. Either technique—binary or decimal—yields the same result.

**Step 2a: Enumerating the Subnetted Network IDs (Decimal)**

1. Create a three-column table with $2^n$ entries where *n* is the number of host bits chosen for the subnetting. The first column is used for the subnet number the second column is for the decimal representation of the subnetted network ID, and the third column is for the dotted decimal representation of the subnetted network ID.
2. Convert the original network ID from dotted decimal notation (*w.x.y.z*) to *N*, its decimal representation:

   $N = w \times 16777216 + x \times 65536 + y \times 256 + z$
3. Compute *I*, the increment value, based on *h*, the number of host bits remaining:

   $I = 2^h$
4. For the first table entry, the all-zeros subnet, the decimal representation of the subnetted network ID is *N*, and the subnetted network ID is *w.x.y.z,* with its new subnet mask.
5. For the decimal representation of the next table entry, add the increment *I* to the previous entry.
6. Convert the decimal representation of the subnetted network ID to dotted decimal notation (*W.X.Y.Z*) using the following formulas (where *s* is the decimal representation of the subnetted network ID):

   - $W = \text{int}(s/16777216)$
   - $X = \text{int}((s \bmod 16777216)/65536)$
   - $Y = \text{int}((s \bmod 65536)/256)$
   - $Z = s \bmod 256$

   In the formulas, *int ( )* denotes integer division and yields the integer multiple, and *mod ( )* denotes the modulus operator and yields the remainder after division.
7. Repeat steps 5 and 6 until the table is complete.

To compare the two techniques and verify that they will both yield the same result, let us perform a decimal 3-bit subnetting of 131.107.0.0.

Based on $n = 3$, we create a table with eight entries. The entry for Subnet 1 is the all-zeros subnet. N, the decimal representation of 131.107.0.0, is 2204827648 (131 ´ 16777216 + 107 ´ 65536). Because there are 13 remaining host bits, the increment value *I* is $2^{13}$, or 8192. Entries for Subnets 2 through 8 are incremented by 8192.

Table 6-10 lists the subnetted network IDs of 131.107.0.0.

**Table 6-10.** *A 3-Bit Subnetting of 131.107.0.0 (Decimal)*

| Subnet | Decimal Representation | Subnetted Network ID |
|---|---|---|
| 1 | 2204827648 | 131.107.0.0/19 |
| 2 | 2204835840 | 131.107.32.0/19 |
| 3 | 2204844032 | 131.107.64.0/19 |
| 4 | 2204852224 | 131.107.96.0/19 |
| 5 | 2204860416 | 131.107.128.0/19 |
| 6 | 2204868608 | 131.107.160.0/19 |
| 7 | 2204876800 | 131.107.192.0/19 |
| 8 | 2204884992 | 131.107.224.0/19 |

**Step 2b: Enumerating IP Address Ranges for Each Subnetted Network ID (Decimal)**

For each subnetted network ID, the range of valid IP addresses must be determined as follows:

1. Create a three-column table with $2^n$ entries where *n* is the number of host bits chosen for the subnetting. The first column is used for the subnet number, the second column is for the decimal representation of the first and last IP address in the range, and the third column is for the dotted decimal representation of the first and last IP address in the range. Alternately, you can extend the table created for enumerating the subnetted network IDs by adding two columns.
2. Compute the increment value *J* based on *h*, the number of host bits remaining:

   $J = 2^h - 2$
3. The decimal representation of the first IP address is $N + 1$, where *N* is the decimal representation of the subnetted network ID. The decimal representation of the last IP address is $N + J$.
4. Convert the decimal representation of the first and last IP address to dotted decimal notation (*W.X.Y.Z*) using the following formulas (where *s* is the decimal representation of the first or last IP address):

   - $W$ = int ($s$/16777216)
   - $X$ = int (($s$ mod 16777216)/65536)
   - $Y$ = int (($s$ mod 65536)/256)
   - $Z = s$ mod 256

   In the formulas, *int ( )* denotes integer division and yields the integer multiple, and *mod ( )* denotes the modulus operator and yields the remainder after division.
5. Repeat steps 3 and 4 until the table is complete.

To continue with our example, we enumerate the range of valid IP addresses for the 3-bit subnetting of 131.107.0.0. Compute the increment value $J = 2^{13} - 2 = 8190$. Table 6-11 lists the ranges of IP addresses for the eight subnetted network IDs.

**Table 6-11.** *Enumeration of IP Addresses for the 3-Bit Subnetting of 131.107.0.0 (Decimal)*

| Subnet | Decimal Representation | Range of IP Addresses |
|---|---|---|
| 1 | 2204827649 - 2204835838 | 131.107.0.1 - 131.107.31.254 |
| 2 | 2204835841 - 2204844030 | 131.107.32.1 - 131.107.63.254 |
| 3 | 2204844033 - 2204852222 | 131.107.64.1 - 131.107.95.254 |
| 4 | 2204852225 - 2204860414 | 131.107.96.1 - 131.107.127.254 |
| 5 | 2204860417 - 2204868606 | 131.107.128.1 - 131.107.159.254 |
| 6 | 2204868609 - 2204876798 | 131.107.160.1 - 131.107.131.107 |
| 7 | 2204876801 - 2204884990 | 131.107.192.1 - 131.107.223.254 |
| 8 | 2204884993 - 2204893182 | 131.107.224.1 - 131.107.255.254 |

#### All-Zeros and All-Ones Subnets

In the previous discussion's examples, we used the subnet where all the host bits were set to 0 (the all-zeros subnet) and the subnet where all the host bits were set to 1 (the all-ones subnet). The use of these subnets is somewhat controversial.

Originally, RFC 950 forbade the use of these subnets as valid subnets because:

- The all-zeros subnet caused problems for early routing protocols that did not use a subnet mask to distinguish a network ID. Therefore, 131.107.0.0/16 was the same network to the router as 131.107.0.0/19.
- The subnet broadcast address for the all-ones subnet uses the same address as a special broadcast address, called the *all-subnets-directed broadcast address*. An IP datagram for the all-subnets-directed broadcast was designed to be forwarded by routers to all classful network ID subnets. For more information on the all-subnets-directed broadcast address, see the section entitled "IP Broadcast Addresses," later in this chapter.

The restriction on the use of the all-zeros and all-ones subnets is part of the legacy of classful networks. The result of this restriction is that substantial portions of a fixed address space are unusable and wasted. For example, when performing a 3-bit subnetting of 131.107.0.0 and excluding the all-zeros and all-ones subnets, only six subnets are available. The range of IP addresses 131.107.0.1 through 131.107.31.254 for the all-zeros subnet and 131.107.224.1 through 131.107.255.254 for the all-ones subnet are unusable.

RFC 1812 now allows the use of all-zeros and all-ones subnets for classless environments for the following reasons:

- Classless environments use routing protocols that advertise the subnet mask with the network ID. Therefore, 131.107.0.0/16 is distinguishable from 131.107.0.0/19.
- The all-subnets-directed broadcast has no meaning in a classless environment.

Even though RFC 1812 now allows the use of these special subnets, there is no guarantee that all of your routers and hosts support them. It is a common default configuration for routers not to support one or the other special subnet and they must be instructed to do so. Verify that your routers and hosts support the all-zeros and all-ones subnets before using them. Hosts and routers running a member of the Windows Server 2003 family support the use of the all-zeros and all-ones subnets without additional configuration.

## *Variable-Length Subnetting*

The preceding discussion illustrates how a fixed network ID can be subdivided into equally sized subnets. The 3-bit subnetting of the classful network ID 131.107.0.0/16 produced eight equally sized subnets, each containing 8190 possible IP addresses. However, in the real world, network segments are not of equal sizes. Some network segments require more IP addresses than others. For example, a network segment containing hosts requires more IP addresses than a backbone network segment containing just a few routers. Point-to-point WAN connections require only two IP addresses.

If equally sized subnetting were done, it would have to be done based on the network segment that required the largest amount of hosts. All other network segments would have the same amount of IP addresses, some of which are unassigned or unusable.

To maximize the use of the fixed address space, subnetting is applied recursively to produce subnets of different sizes all derived from the same original network ID. This is known as variable-length subnetting. Differently sized subnets use different subnet masks, or variable-length subnet masks (VLSM).

Because all of the subnets are derived from the same network ID, if the subnets are contiguous, the routes for all the subnets can be summarized by advertising the original network ID. Contiguous subnets are subnets of the same network ID that are connected to each other.

When performing variable-length subnetting, care must be taken so that each subnet is unique, and with its subnet mask, can be distinguished from all other subnets of the original network ID. Variable-length subnetting requires a careful analysis of your network segments to determine how many of each sized network you require. Then, starting from your network ID, subnetting is performed as many times as needed to express as many subnets as desired with the proper sizes.

With variable-length subnetting, the subnetting technique is applied recursively: You subnet a previously subnetted network ID. When subnetting a previously subnetted network ID, the subnetted network ID bits are fixed and an appropriate number of remaining host bits is chosen for subnetting.

**Example of Variable-Length Subnetting**

To expand on our earlier example, let us continue subnetting the classful network ID of 131.107.0.0/16. After the 3-bit subnetting has been performed, the remaining addresses must be divided such that:

- Half of the addresses are reserved for future use
- Three subnets are allocated with up to 8190 IP addresses
- 31 subnets are allocated with up to 254 IP addresses
- 64 subnets are allocated with only 2 IP addresses

Recall that the 3-bit subnetting of 131.107.0.0/16 produced the eight subnets listed in Table 6-12.

**Table 6-12.** *The Eight Subnets for the 3-Bit Subnetting of 131.107.0.0/16*

| Subnet | Subnetted Network ID |
|---|---|
| 1 | 131.107.0.0/19 |
| 2 | 131.107.32.0/19 |
| 3 | 131.107.64.0/19 |
| 4 | 131.107.96.0/19 |
| 5 | 131.107.128.0/19 |
| 6 | 131.107.160.0/19 |
| 7 | 131.107.192.0/19 |
| 8 | 131.107.224.0/19 |

**Reserve Half of the IP Addresses for Future Use**

To reserve half of the addresses for future use, set aside the first four subnets (131.107.0.0/19, 131.107.32.0/19, 131.107.64.0/19, 131.107.96.0/19).

**Obtain Three Subnets with up to 8190 IP Addresses**

To obtain three subnets with up to 8190 IP addresses per subnet, choose the next three subnets (131.107.128.0/19, 131.107.160.0/19, 131.107.192.0/19). Each subnet has 13 host bits, for a total of 8190 IP addresses per subnet.

**Obtain 31 Subnets with up to 254 IP Addresses**

To obtain 31 subnets, each with up to 254 IP addresses, perform a 5-bit subnetting of 131.107.224.0/19. The result is 32 subnets (131.107.224.0/24, 131.107.225.0/24, 131.107.226.0/24 . . . 131.107.253.0/24, 131.107.254.0/24, 131.107.255.0/24). To fulfill the requirement, choose the first 31 subnets (131.107.224.0/24 to 131.107.254.0/24).

**Obtain 64 Subnets with only 2 IP Addresses**

To obtain 64 subnets with only 2 usable IP addresses, perform a 6-bit subnetting of 131.107.255.0/24. The result is 64 subnets (131.107.255.4/30, 131.107.255.8/30, 131.107.255.12/30 . . . 131.107.255.244/30, 131.107.255.248/30, 131.107.255.252/30).

Figure 6-10 shows the variable-length subnetting of 131.107.0.0/16.

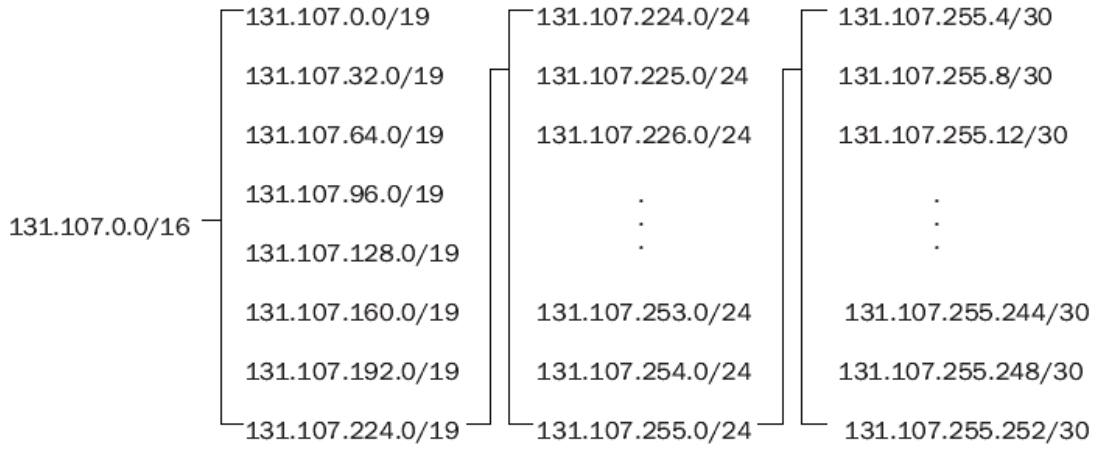


**Figure 6-10.** *The variable-length subnetting of 131.107.0.0/16 into subnets of different sizes.*

**Variable-Length Subnetting and Routing**

Variable-length subnetting requires routing protocols to advertise the subnet mask with the network ID. Routing Information Protocol (RIP) version 2, Open Shortest Path First (OSPF), and Border Gateway Protocol version 4 (BGP-v4) support variable-length subnetting environments, but RIP version 1 does not.

# *Supernetting and CIDR*

As the Internet grew suddenly from a collection of educational institutions and government agencies to a business-oriented, pervasive, global internetwork, great stress was placed on the IP address space. Assigning classful network IDs to organizations meant a quick, wasteful depletion of the Internet address space.

For example, numerous organizations worldwide require more than 254 IP addresses. Therefore, a single class C network ID is insufficient. A single class B network ID, however, provides sufficient IP addresses and enough host bits to implement subnetting within the organization's internal network. Although this is good for the organization, it is bad for the Internet IP address space. Consider the smaller organization that needs only 4000 IP addresses. Assigning a class B network with 65,534 possible IP addresses means that 61,534 IP addresses are unassigned and wasted.

Now, instead of an entire class B network ID, the InterNIC assigns a range of class C network IDs. For example, InterNIC assigns 16 class C network IDs to an organization needing 4000 IP addresses. Each class C network ID allows for 254 IP addresses. Therefore, 16 class C network IDs allow for 4064 IP addresses.

This technique minimizes the wasting of Internet IP addresses, but it introduces a new problem. If a single class B network ID is assigned, that single class B network ID becomes a single route in the routing tables of the Internet backbone routers. If 16 class C network IDs are assigned, 16 class C network IDs become 16 routes in the routing tables of the Internet backbone routers.

Extending this example to its ultimate limits, there are more than 2 million class C network IDs. After assigning them all, it is possible to have more than 2 million routes in the routing tables of the Internet backbone routers. Even with today's technology, it is difficult to build an IP router that can have a routing table with millions of entries, and forward IP datagrams at megabit or gigabit per second speeds.

To prevent this scaling problem from overwhelming Internet routers, a route aggregation technique called CIDR is used to express a range of class C network IDs as a single route. This is the method of address allocation that the modern Internet uses. CIDR solves the scaling problem by minimizing the total number of routes that must be stored in the routing tables of Internet routers.

*CIDR uses a supernetted subnet mask to express the range of class C network IDs. A supernetted subnet mask is less specific or contains fewer network ID bits than a classful subnet mask. In contrast, a subnetted subnet mask is more specific, or contains more network ID bits, than a classful subnet mask.*

**Views on CIDR Allocation**

The CIDR method of address allocation can be viewed in two different ways:

1. A range of class C network IDs
2. An address space in which multiple classful networks are combined into a single classless network

The latter perspective is more appropriate for today's Internet and for looking forward to IPv6.

**A Range of Class C Network IDs**

Viewed as a range of class C network IDs, our requirement is based on the number of class C network segments needed in our organization. The following requirements are for a range of class C network IDs to be expressible as a single route using a network ID and a subnet mask:

- The class C network IDs must be sequential.
- The number of allocated class C network IDs must be expressed as a power of 2.

*For example, Table 6-13 lists the range (or block) of eight class C network IDs, starting with network ID 223.1.184.0.*

***Table 6-13.*** *A Block of Eight Class C Network IDs Starting with 223.1.184.0*

| | | |
|---|---|---|
| Starting Network ID | 223.1.184.0 | 11011111 00000001 10111000 00000000 |
| Ending Network ID | 223.1.191.0 | 11011111 00000001 10111111 00000000 |

Notice that the first 21 bits (underlined) of the range of class C network IDs are the same. The last 3 bits of the third octet vary over all possible values from 000 through 111. This range of class C network IDs can be aggregated with the network ID and subnet mask listed in Table 6-14.

**Table 6-14.** *The Aggregated Block of Class C Network Ids*

| | |
|---|---|
| Network ID | 223.1.184.0 |
| Subnet Mask (binary) | 11111111 11111111 11111000 00000000 |
| Subnet Mask | 255.255.248.0 |
| Network Prefix Length | /21 |

A block of class-based network IDs, as allocated in this example, is known as a *CIDR block*.

Table 6-15 lists the number of class C network IDs and the supernetted subnet mask for a required number of hosts.

**Table 6-15.** *Supernetting and Class C Addresses*

| Required Hosts | Number of Class C Network IDs | Supernetted Subnet Mask |
|---|---|---|
| 2-254 | 1 | 255.255.255.0 or /24 |
| 255-508 | 2 | 255.255.254.0 or /23 |
| 509-1016 | 4 | 255.255.252.0 or /22 |
| 1017-2032 | 8 | 255.255.248.0 or /21 |
| 2033-4064 | 16 | 255.255.240.0 or /20 |
| 4065-8128 | 32 | 255.255.224.0 or /19 |
| 8129-16,256 | 64 | 255.255.192.0 or /18 |
| 16,257-32,512 | 128 | 255.255.128.0 or /17 |
| 32,513-65,024 | 256 | 255.255.0.0 or /16 |

**An Address Space**

From the perspective of an address space, CIDR blocks are no longer viewed as a range of class C network IDs. Even though the CIDR block is obtained from the class-defined range of class C network IDs, it does not necessarily represent a range of class C network IDs. Viewing the CIDR block as a range of class C network IDs implies that we will assign each class C network ID within the block to each of our networks.

In reality, we typically want to assign network IDs of various sizes to the networks of our intranet in a variable-length subnetting scheme. Now our requirement is based on the number of IP addresses required, rather than the number of class C networks in our organization.

For example, to assign 4000 IP addresses to an organization, determine the number of bits required to express 4000 IP addresses. Using powers of 2, 12 bits are needed to express 4094 IP addresses. Therefore, 12 bits are used for the host ID portion, and 20 bits for the network ID portion. The subnet mask indicates 20 bits of network ID. For example, starting from an unassigned portion of the IP address space, the InterNIC allocates the 223.1.176.0 network with the subnet mask of 255.255.240.0 (or 223.1.176.0/20) address space to the organization.

The allocated address space allows the assignment of the range of IP addresses from 223.1.176.1 through 223.1.191.254. However, it is unlikely that the organization will use all 4094 IP addresses on the same network segment. Rather, the organization can use variable-length subnetting and the 12 host bits to create a series of subnets containing the suitable number of appropriately sized subnets.

With CIDR, IP network IDs lose their classful heritage and become address spaces where certain bits are fixed (the network ID bits), and certain bits are variable (the host ID bits). Using variable-length subnetting techniques, the organization's needs should determine how to best utilize the host bits.

**CIDR and Routing**

CIDR, like variable-length subnetting, requires routing protocols to advertise the subnet mask with the network ID. RIP version 2, OSPF, and BGP-v4 support CIDR environments, but RIP version 1 does not.

## *Public and Private Addresses*

When deploying an IP addressing scheme in your organization, the main consideration is whether your intranet is connected to the Internet:

- If your organization is not connected to the Internet, it is technically possible to choose any IP network IDs—classful or classless—without concern for using overlapping addresses being used on the Internet. However, it is highly recommended that you choose a private address range.
- If your organization is connected to the Internet, it can be connected in one of two ways. If your organization uses a direct-routed connection using a router or firewall, you must use InterNIC-compliant addresses as allocated by the InterNIC or an Internet service provider (ISP). If your organization uses an indirect connection using a proxy server or a Network Address Translator (NAT), you must use addresses that do not overlap with addresses that do, or might, exist on the Internet.

Organizations connected to the Internet must choose between the use of public or private addresses.

**Public Addresses**

The InterNIC assigns public addresses that are within the public address space consisting of all of the possible unicast addresses on the Internet worldwide. Historically, the InterNIC assigned classful network IDs to organizations connecting to the Internet without regard to geographical location. Today, the InterNIC assigns CIDR blocks to ISPs based on geographical location; the ISPs then subdivide their assigned CIDR blocks to customers. Subdivision of the remaining class C address space based on geographical location was done to provide hierarchical routing and to minimize the number of routes in Internet backbone routers. Public addresses are guaranteed to be globally unique.

When an organization or an ISP is assigned a block of addresses in the public address space, a route exists in the Internet routers' routing tables so that the assigned public addresses are reachable through the ISP. Historically, a classful network ID was added to all of the Internet routers. Today, a route consisting of the range of assigned addresses is added to the routing tables of regional and ISP Internet routers.

One or more (network ID, mask) pairs summarize the range of public IP addresses assigned to an organization. These pairs become the routes in the ISP and Internet routers so that the IP addresses of the organization can be reached.

### Illegal or Overlapping Addresses

Organizations that are not connected to the Internet either directly or indirectly are free to choose any addressing scheme without regard to whether the addresses have been assigned to another ISP or organization. However, if that organization later decides to connect to the Internet, a new addressing scheme might be required.

The addresses assigned when the organization was not connected to the Internet might include public addresses that have been assigned to other organizations or ISPs by the InterNIC. If that is the case, these addresses are duplicates that conflict with assigned addresses. This is known as illegal, or overlapping, addressing. Internet traffic from hosts using illegal addresses is forwarded to the routers of the organization that was originally assigned those addresses. Therefore, organizations using illegal addressing are unreachable on the Internet.

For example, an organization that is not connected to the Internet decides to use the address space 207.46.130.0/24 for its intranet. As long as the organization does not connect to the Internet, the use of 207.46.130.0/24 is not an issue. If the organization then connects to the Internet using a direct routed connection, the use of 207.46.130.0/24 is illegal and no responses from hosts on the 207.46.130.0/24 network segment are received.

In this configuration, when a host sends traffic to an Internet location, it sends the traffic with the source IP address within the address space of 207.46.130.0/24. When the Internet host sends a response, it sends the response to the destination IP address within the address space of 207.46.130.0/24. InterNIC assigned Microsoft Corporation the address space 207.46.130.0/24, and a route exists in Internet routers to forward traffic with the destination IP address in this range to Microsoft Corporation's routers. Therefore, the responses to traffic sent by the hosts on the illegal address space 207.46.130.0/24 are forwarded to Microsoft Corporation's routers, and not to the routers of the organization using the illegal addresses.

**NOTE**

---

It is common practice among ISPs to discard IP packets sent from a customer site when the source IP address field is not set to a valid public address assigned to the customer. This is known as *ingress filtering,* which attempts to prevent the sending of traffic from hosts using illegal addresses and address spoofing (the sending of IP traffic from a source IP address that is not assigned to a host).

### Private Addresses

As the Internet experienced exponential growth, the demand for public IP addresses increased commensurately. Because each node on an organization's intranet required a globally unique public IP address, organizations requested enough IP addresses from the InterNIC to assign unique IP addresses to all of the nodes within their organizations.

However, when an analysis of IP addressing within organizations was done, the Internet authorities noticed that most organizations actually needed very few public addresses. The only hosts that required public IP addresses were those that communicated directly with systems on the Internet, such as Web servers, File Transfer Protocol (FTP) servers, e-mail servers, proxy servers, and firewalls. Most of the hosts within an organization's intranet obtained access to Internet resources through Application Layer gateways such as proxy servers and e-mail servers.

For hosts within the organization's intranet that do not require direct access to the Internet, a legal IP address space must be used. For this purpose, Internet authorities created the private address space, a subset of the Internet IP address space that can be used without conflict within an organization, for hosts that do not require a direct connection to the Internet.

The private and public address spaces are separate and do not overlap. The InterNIC never assigns private addresses—IP addresses within the private address space—to an organization or ISP. This also means that private IP addresses are not reachable on the Internet.

Because private addresses are not reachable on the Internet, hosts on an intranet with private addressing cannot be directly connected to the Internet. Rather, they must be indirectly connected to the Internet using a NAT or an Application Layer gateway such as a proxy server.

A NAT is a router that translates between private addresses and public addresses for Internet traffic. The proxy server receives a request from a host on the intranet for Internet resources. The proxy server then sends the request to the Internet resource and the response traffic is forwarded back to the requesting host. When the proxy server sends the request to the Internet resource, it uses public addressing. Both proxy servers and NATs have private addresses on their intranet interface and public addresses on their Internet interface.

**MORE INFO**

---

For more information on network address translation, see RFC 3022, which can be found in the \Rfc folder on the companion CD-ROM.

The following three address blocks define the private address space:

- **10.0.0.0/8** The 10.0.0.0/8 private network is an address space with 24 host bits that can be used for any subnetting scheme within the private organization.
- **172.16.0.0/12** The 172.16.0.0/12 private network is an address space with 20 host bits that can be used for any subnetting scheme within the private organization. From a classful perspective, the 172.16.0.0/12 private network ID is the range of 16 class B network IDs from 172.16.0.0/16 through 172.31.0.0/16.
- **192.168.0.0/16** The 192.168.0.0/16 private network is an address space with 16 host bits that can be used for any subnetting scheme within the private organization. From a classful perspective, the 192.168.0.0/16 private network ID is the range of 256 class C network IDs from 192.168.0.0/24 through 192.168.255.0/24.

**MORE INFO**

---

For more information on the public address space, see RFC 1918, which can be found in the \Rfc folder on the companion CD-ROM.

## *Automatic Private IP Addressing*

When you configure a computer running a member of the Windows Server 2003 family to obtain its IP address automatically and a DHCP server does not respond to the DHCPREQUEST and DHCPDISCOVER messages, TCP/IP for the Windows Server 2003 family configures itself using the Automatic Private IP Addressing (APIPA) feature (provided TCP/IP is not configured to use an alternate static address configuration). Using APIPA, TCP/IP for the Windows Server 2003 family randomly picks an IP address in the address space of 169.254.0.0/16. This address space has been reserved by the Internet Assigned Numbers Authority (IANA) and is not reachable on the Internet.

After choosing an IP address, TCP/IP for the Windows Server 2003 family sends a gratuitous Address Resolution Protocol (ARP) to check for IP address uniqueness. After receiving no response to the gratuitous ARP, TCP/IP for the Windows Server 2003 family is configured for the randomly chosen IP address and the subnet mask of 255.255.0.0. If a response to the gratuitous ARP is received, TCP/IP for the Windows Server 2003 family randomly chooses a new address in the 169.254.0.0/16 address space. After APIPA configuration, TCP/IP for the Windows Server 2003 family continues to send DHCPDISCOVER messages every five minutes. If a DHCP server responds, TCP/IP for the Windows Server 2003 family abandons the APIPA configuration and the DHCP-allocated address takes effect. For more information on gratuitous ARP, see Chapter 3, "Address Resolution Protocol (ARP)."

APIPA was designed to simplify the configuration of a single subnet small office/home office (SOHO) network that is not connected to the Internet or any other IP internetwork. With APIPA, all the computers on a single-subnet SOHO network configure themselves and are able to communicate without manually configuring TCP/IP or setting up a DHCP server.

APIPA does not provide automatic configuration of a default gateway, the IP address of a Domain Name System (DNS) server, a DNS domain name, the IP address of a Windows Internet Name Service (WINS) server, or NetBIOS node type. A single-subnet SOHO network does not need a default gateway, and broadcast NetBIOS name queries resolve names for communication between computers.

TCP/IP for the Windows Server 2003 family APIPA behavior is controlled by the following registry settings:

**IPAutoconfigurationEnabled**

**Key:**
**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters**
**and**

**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces*InterfaceGUID***

**Value type: REG_DWORD**

**Valid range: 0 - 1**

**Default: 1**
**Present by default: No**

IPAutoconfigurationEnabled either enables (when set to 1) or disables (when set to 0) APIPA-based IP address configuration either globally or per interface. The default is enabled both globally and per interface, and the setting for an interface overrides the global setting.

**IPAutoconfigurationSubnet**

**Key:**
**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters and**

**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces*InterfaceGUID***

**Value type: REG_SZ (String)**

**Valid range: A valid IP network ID expressed in dotted decimal notation.**

**Default: 169.254.0.0**
**Present by default: No**

IPAutoconfigurationSubnet specifies the IP network ID for the network prefix of APIPA-configured addresses. The default value is 169.254.0.0. IPAutoconfigurationSubnet can be specified globally or per interface, and the setting for an interface overrides the global setting.

**IPAutoconfigurationMask**

**Key:**
**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters and**

**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces*InterfaceGUID***

**Value type: REG_SZ (String)**

**Valid range: A valid subnet mask expressed in dotted decimal notation.**

**Default: 255.255.0.0**
**Present by default: No**

IPAutoconfigurationMask specifies the subnet mask for the network prefix of APIPA- configured addresses. The default value is 255.255.0.0. IPAutoconfigurationMask can be specified globally or per interface and the setting for an interface overrides the global setting.

**NOTE**

---

The network ID specified for the IPAutoconfigurationSubnet cannot be more specific than the subnet mask specified for the IPAutoconfigurationMask. In other words, the network ID cannot contain bits set to 1 when the correspond-ing bit in the mask is set to 0. An example of an incorrect network ID and subnet mask combination is the network ID 169.254.47.0 with the subnet mask of 255.255.0.0. The correct subnet mask for this network ID is 255.255.255.0.

---