IP Addressing

Each TCP/IP host is identified by a logical IP address. The IP address is a network layer address and has no dependence on the data link layer address (such as a MAC address of a network interface card). A unique IP address is required for each host and network component that communicates using TCP/IP.

The IP address identifies a system's location on the network in the same way a street address identifies a house on a city block. Just as a street address must identify a unique residence, an IP address must be globally unique and have a uniform format.

Each IP address includes a network ID and a host ID.

- The *network ID* (also known as a *network address*) identifies the systems that are located on the same physical network bounded by IP routers. All systems on the same physical network must have the same network ID. The network ID must be unique to the internetwork.

- The *host ID* (also known as a *host address*) identifies a workstation, server, router, or other TCP/IP host within a network. The address for each host must be unique to the network ID.

Note The use of the term *network ID* refers to any IP network ID, whether it is class-based, a subnet, or a supernet.

An IP address is 32 bits long. Rather than working with 32 bits at a time, it is a common practice to segment the 32 bits of the IP address into four 8-bit fields called *octets*. Each octet is converted to a decimal number (the Base 10 numbering system) in the range 0-255 and separated by a period (a dot). This format is called *dotted decimal notation*. Table 10 provides an example of an IP address in binary and dotted decimal formats.

Table 10. Example of an IP address in binary and dotted decimal format

Binary Format	Dotted Decimal Notation
11000000 10101000 00000011 00011000	192.168.3.24

The notation *w.x.y.z* is used when referring to a generalized IP address and shown in Figure 3.

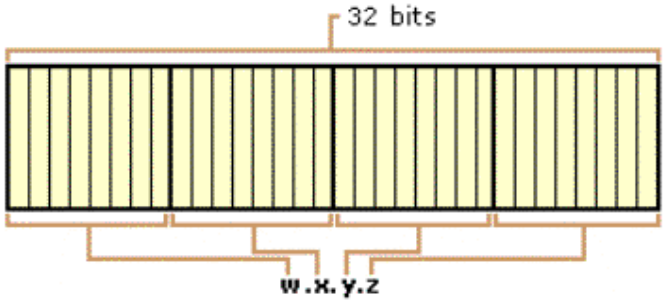

Figure 3. The IP address

Address Classes

The Internet community originally defined five address classes to accommodate networks of varying sizes. Microsoft TCP/IP supports class A, B, and C addresses assigned to hosts. The class of

address defines which bits are used for the network ID and which bits are used for the host ID. It also defines the possible number of networks and the number of hosts per network.

Class A

Class A addresses are assigned to networks with a very large number of hosts. The high-order bit in a class A address is always set to zero. The next seven bits (completing the first octet) complete the network ID. The remaining 24 bits (the last three octets) represent the host ID. This allows for 126 networks and 16,777,214 hosts per network. Figure 4 illustrates the structure of class A addresses.

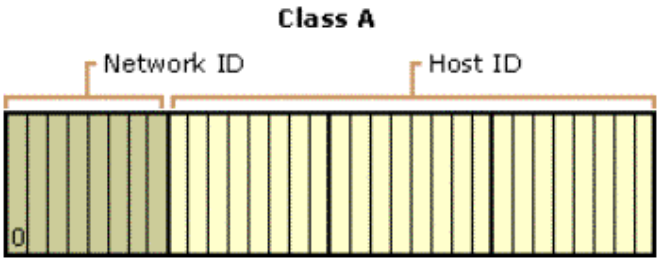

Figure 4. Class A IP addresses

Class B

Class B addresses are assigned to medium-sized to large-sized networks. The two high-order bits in a class B address are always set to binary 1 0. The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID. This allows for 16,384 networks and 65,534 hosts per network. Figure 5 illustrates the structure of class B addresses.

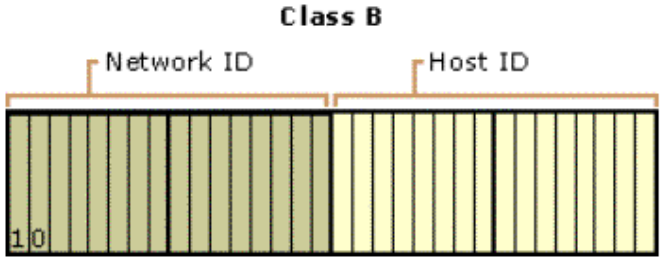

Figure 5. Class B IP addresses

Class C

Class C addresses are used for small networks. The three high-order bits in a class C address are always set to binary 1 1 0. The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID. This allows for 2,097,152 networks and 254 hosts per network. Figure 6 illustrates the structure of class C addresses.

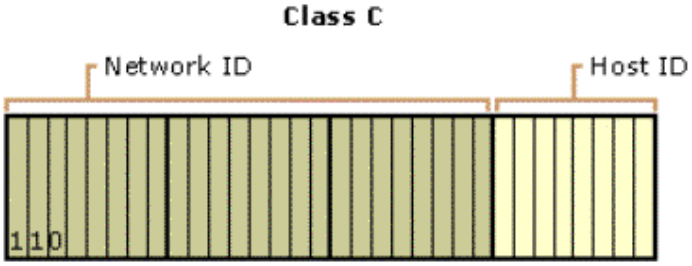

Figure 6. Class C IP addresses

Class D

Class D addresses are reserved for IP multicast addresses. The four high-order bits in a class D address are always set to binary 1 1 1 0. The remaining bits are for the address that interested hosts will recognize. Microsoft supports class D addresses for applications to multicast data to multicast-capable hosts on an internetwork.

Class E

Class E addresses are experimental addresses reserved for future use. The high-order bits in a class E address are set to 1 1 1 1.

Table 11 is a summary of address classes A, B, and C that can be used for host IP addresses.

Table 11. IP address class summary

Class	Value for w[1]	Network ID Portion	Host ID Portion	Available Networks	Hosts per Network
A	1–126	w	x.y.z	126	16,777,214
B	128–191	w.x	y.z	16,384	65,534
C	192–223	w.x.y	z	2,097,152	254

[1] The class A address 127.x.y.z is reserved for loopback testing and interprocess communication on the local computer.

Network ID guidelines

The network ID identifies the TCP/IP hosts located on the same physical network. All hosts on the same physical network must be assigned the same network ID to communicate with each other.

Follow these guidelines when assigning a network ID:

- The network address must be unique to the IP internetwork. If you plan on having a direct routed connection to the public Internet, the network ID must be unique to the Internet. If you do not plan on connecting to the public Internet, the local network ID must be unique to your private internetwork.
- The network ID cannot begin with the number 127. The number 127 in a class A address is reserved for internal loopback functions.
- All bits within the network ID cannot be set to 1. All 1s in the network ID are reserved for use as an IP broadcast address.
- All bits within the network ID cannot be set to 0. All 0s in the network ID are used to denote a specific host on the local network and will not be routed.

Table 12 lists the valid ranges of network IDs based on the IP address classes. To denote IP network IDs, the host bits are all set to 0. Note that even though expressed in dotted decimal notation, the network ID is not an IP address.

Table 12. Class ranges of network IDs

Address Class	First Network ID	Last Network ID
Class A	1.0.0.0	126.0.0.0
Class B	128.0.0.0	191.255.0.0
Class C	192.0.0.0	223.255.255.0

Host ID guidelines

The host ID identifies a TCP/IP host within a network. The combination of IP network ID and IP host ID is an IP address.

Follow these guidelines when assigning a host ID:

- The host ID must be unique to the network ID.
- All bits within the host ID cannot be set to 1, because this host ID is reserved as a broadcast address to send a packet to all hosts on a network.
- All bits in the host ID cannot be set to 0, because this host ID is reserved to denote the IP network ID.

Table 13 lists the valid ranges of host IDs based on the IP address classes.

Table 13. Class ranges of host IDs

Address Class	First Host ID	Last Host ID
Class A	w.0.0.1	w.255.255.254
Class B	w.x.0.1	w.x.255.254
Class C	w.x.y.1	w.x.y.254

Subnets and Subnet Masks

The Internet Address Classes were designed to accommodate three different scales of IP internetworks, where the 32 bits of the IP address are apportioned between network IDs and host IDs depending on how many networks and hosts per network are needed. However, consider the class A network ID, which has the possibility of over 16 million hosts on the same network. All the hosts on the same physical network bounded by IP routers share the same broadcast traffic; they are in the same broadcast domain. It is not practical to have 16 million nodes in the same broadcast domain. The result is that most of the 16 million host addresses are not assignable and are wasted. Even a class B network with 65 thousand hosts is impractical.

In an effort to create smaller broadcast domains and to better utilize the bits in the host ID, an IP network can be subdivided into smaller networks, each bounded by an IP router and assigned a new *subnetted network ID*, which is a subset of the original class-based network ID.

This creates *subnets*, subdivisions of an IP network, each with its own unique subnetted network ID. Subnetted network IDs are created by using bits from the host ID portion of the original class-based network ID.

Consider the example in Figure 7. The class B network of 139.12.0.0 can have up to 65,534 nodes. This is far too many nodes and, in fact, the current network is becoming saturated with broadcast traffic. The subnetting of network 139.12.0.0 should be done in such a way so that it does not impact or require the reconfiguration of the rest of the IP internetwork.

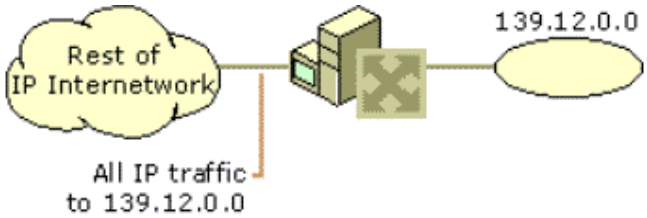

Figure 7. Network 139.12.0.0 before subnetting

Network 139.12.0.0 is subnetted by utilizing the first 8 host bits (the third octet) for the new subnetted network ID. When 139.12.0.0 is subnetted, as shown in Figure 8, separate networks with their own subnetted network IDs (139.12.1.0, 139.12.2.0, 139.12.3.0) are created. The router is aware of the separate subnetted network IDs and will route IP packets to the appropriate subnet.

Note that the rest of the IP internetwork still regards all the nodes on the three subnets as being on network 139.12.0.0. The other routers in the IP internetwork are unaware of the subnetting being done on network 139.12.0.0, and therefore require no reconfiguration.

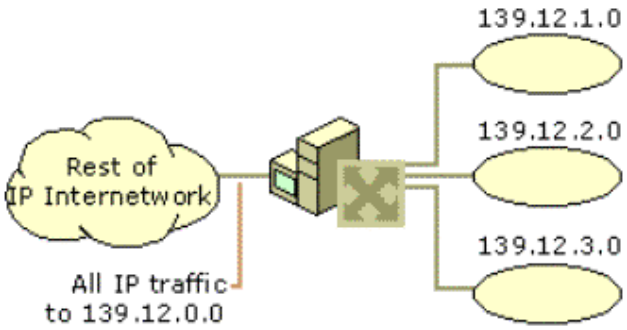

Figure 8. Network 139.12.0.0 after subnetting

A key element of subnetting is still missing. How does the router that is subdividing network 139.12.0.0 know how the network is being subdivided and which subnets are available on which router interfaces? To give the IP nodes this new level of awareness, the router must be told exactly how to discern the new subnetted network ID regardless of Internet Address Classes. To tell an IP node exactly how to extract a network ID, either class-based or subnetted, a *subnet mask* is used.

Subnet masks

With the advent of subnetting, one can no longer rely on the definition of the IP address classes to determine the network ID in the IP address. A new value is needed to define which part of the IP address is the network ID and which part is the host ID, regardless of whether class-based or subnetted network IDs are being used.

RFC 950 defines the use of a *subnet mask* (also referred to as an *address mask*) as a 32-bit value that is used to distinguish the network ID from the host ID in an arbitrary IP address. The bits of the subnet mask are defined as:

- All bits that correspond to the network ID are set to 1.
- All bits that correspond to the host ID are set to 0.

Each host on a TCP/IP network requires a subnet mask, even on a single-segment network. Either a *default subnet mask*, which is used when using class-based network IDs, or a *custom subnet mask*, which is used when subnetting or supernetting, is configured on each TCP/IP node.

Dotted decimal representation of subnet masks

Subnet masks are frequently expressed in dotted decimal notation. Once the bits are set for the network ID and host ID portion, the resulting 32-bit number is converted to dotted decimal notation. Note that even though it is expressed in dotted decimal notation, a subnet mask is not an IP address.

A default subnet mask is based on the IP address classes and is used on TCP/IP networks that are not divided into subnets. Table 14 lists the default subnet masks using the dotted decimal notation for the subnet mask.

Table 14. Default subnet masks in dotted decimal notation

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

Custom subnet masks are those that differ from the above default subnet masks when doing subnetting or supernetting. For example, 138.96.58.0 is an 8-bit subnetted class B network ID. Eight bits of the class-based host ID are being used to express subnetted network IDs. The subnet mask uses a total of 24 bits (255.255.255.0) to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in dotted decimal notation as:

```
138.96.58.0, 255.255.255.0
```

Network prefix length representation of subnet masks

Since the network ID bits must be always chosen in a contiguous fashion from the high-order bits, a shorthand way of expressing a subnet mask is to denote the number of bits that define the network ID as a network prefix using the network prefix notation: /<# of bits>. Table 15 lists the default subnet masks using the network prefix notation for the subnet mask.

Table 15. Default subnet masks in network prefix notation

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

For example, the class B network ID 138.96.0.0 with the subnet mask of 255.255.0.0 would be expressed in network prefix notation as 138.96.0.0/16.

As an example of a custom subnet mask, 138.96.58.0 is an 8-bit subnetted class B network ID. The subnet mask uses a total of 24 bits to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in network prefix notation as:

```
138.96.58.0/24
```

Note Since all hosts on the same network must use the same network ID, the ID must be defined by the same subnet mask. For example, 138.23.0.0/16 is not the same network ID as 138.23.0.0/24. The network ID 138.23.0.0/16 implies a range of valid host IP addresses from 138.23.0.1 to 138.23.255.254. The network ID 138.23.0.0/24 implies a range of valid host IP addresses from 138.23.0.1 to 138.23.0.254. Clearly, these network IDs do not represent the same range of IP addresses.

Determining the network ID

To extract the network ID from an arbitrary IP address using an arbitrary subnet mask, IP uses a mathematical operation called a *logical AND comparison*. In an AND comparison, the result of two items being compared is true only when both items being compared are true, otherwise, the result is false. Applying this principle to bits, the result is 1 when both bits being compared are 1; otherwise, the result is 0.

IP takes the 32-bit IP address and logically ANDs it with the 32-bit subnet mask. This operation is known as a *bit-wise logical AND*. The result of the bit-wise logical AND comparison of the IP address and the subnet mask is the network ID.

For example, what is the network ID of the IP node 129.56.189.41 with a subnet mask of 255.255.240.0?

To obtain the result, turn both numbers into their binary equivalents and line them up. Then perform the AND operation on each bit and write down the result.

```
10000001 00111000 10111101 00101001  IP Address
11111111 11111111 11110000 00000000  Subnet Mask
10000001 00111000 10110000 00000000  Network ID
```

The result of the bit-wise logical AND of the 32 bits of the IP address and the subnet mask is the network ID 129.56.176.0.

Subnetting

While the conceptual notion of subnetting by utilizing host bits is straightforward, the actual mechanics of subnetting are a bit more complicated. Subnetting is a three-step procedure:

1. Determine the number of host bits to be used for the subnetting.
2. Enumerate the new subnetted network IDs.
3. Enumerate the IP addresses for each new subnetted network ID.

Step 1: determining the number of host bits

The number of host bits being used for subnetting determines the possible number of subnets and hosts per subnet. Before you choose how many host bits, you should have a good idea of the number of subnets and hosts you will have in the future. Using more bits for the subnet mask than required will save you the time of reassigning IP addresses in the future.

The more host bits that are used, the more subnets (subnetted network IDs) you can have—but with fewer hosts. If you use too many host bits, you will allow for growth in the number of subnets but limit the growth in the number of hosts. If you use too few hosts, you will allow for growth in the number of hosts but limit the growth in the number of subnets.

For example, Figure 9 illustrates the subnetting of up to the first 8 host bits of a class B network ID. If we choose one host bit for subnetting, we obtain 2 subnetted network IDs with 16,382 hosts per subnetted network ID. If we choose 8 host bits for subnetting, we obtain 256 subnetted network IDs with 254 hosts per subnetted network ID.

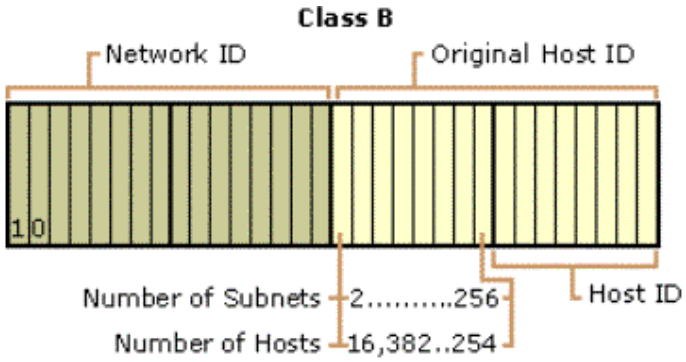

Figure 9. Subnetting a class B network ID

In practice, network administrators define a maximum number of nodes they want on a single network. Recall that all nodes on a single network share all the same broadcast traffic; they reside in the same broadcast domain. Therefore, growth in the amount of subnets is favored over growth in the amount of hosts per subnet.

Follow these guidelines to determine the number of host bits to use for subnetting:

1. Determine how many subnets you need now and will need in the future. Each physical network is a subnet. WAN connections may also count as subnets depending on whether your routers support unnumbered connections.

2. Use additional bits for the subnet mask if:
    - You will never require as many hosts per subnet as allowed by the remaining bits.
    - The number of subnets will increase in the future, requiring additional host bits.

To determine the desired subnetting scheme, you will start with an existing network ID to be subnetted. The network ID to be subnetted can be a class-based network ID, a subnetted network ID, or a supernet. The existing network ID will contain a series of network ID bits, which are fixed, and a series of host ID bits, which are variable. Based on your requirements for the number of subnets and the number of hosts per subnet, you will choose a specific number of host bits to be used for the subnetting.

Table 16 shows the subnetting of a class A network ID. Based on a required number of subnets and a maximum number of hosts per subnet, a subnetting scheme can be chosen.

Table 16. Subnetting 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
513-1,024	10	255.255.192.0 or /18	16,382
1,025-2,048	11	255.255.224.0 or /19	8,190
2,049-4,096	12	255.255.240.0 or /20	4,094
4,097-8,192	13	255.255.248.0 or /21	2,046
8,193-16,384	14	255.255.252.0 or /22	1,022
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 17 shows the subnetting of a class B network ID.

Table 17. Subnetting 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	8,190
9-16	4	255.255.240.0 or /20	4,094
17-32	5	255.255.248.0 or /21	2,046
33-64	6	255.255.252.0 or /22	1,022
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
513-1,024	10	255.255.255.192 or /26	62
1,025-2,048	11	255.255.255.224 or /27	30
2,049-4,096	12	255.255.255.240 or /28	14
4,097-8,192	13	255.255.255.248 or /29	6
8,193-16,384	14	255.255.255.252 or /30	2

Table 18 shows the subnetting of a class C network ID.

Table 18. Subnetting 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: enumerating subnetted network IDs

Based on the number of host bits you use for your subnetting, you must list the new subnetted network IDs. There are two main approaches:

- Binary—List all possible combinations of the host bits chosen for subnetting and convert each combination to dotted decimal notation.
- Decimal—Add a calculated increment value to each successive subnetted network ID and convert to dotted decimal notation.

Either method produces the same result—the enumerated list of subnetted network IDs.

Note There are a variety of documented shortcut techniques for subnetting. However, they only work under a specific set of constraints (for example, only up to 8 bits of a class-based network ID). The methods described below are designed to work for any subnetting situation (class-based, more than 8 bits, supernetting, variable-length subnetting).

Binary subnetting procedure

1. Based on *n*, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. The first column is the subnet number (starting with 1), the second column is the binary representation of the subnetted network ID, and the third column is the dotted decimal representation of the subnetted network ID.

   For each binary representation, the bits of the network ID being subnetted are fixed to their appropriate values and the remaining host bits are set to all 0s. The host bits chosen for subnetting will vary.

2. In the first table entry, set the subnet bits to all 0s and convert to dotted decimal notation. The original network ID is subnetted with its new subnet mask.

3. In the next table entry, increase the value within the subnet bits.

4. Convert the binary result to dotted decimal notation.

5. Repeat steps 3 and 4 until the table is complete.

As an example, a 3-bit subnet of the private network ID 192.168.0.0 is needed. The subnet mask for the new subnetted network IDs is 255.255.224.0 or /19. Based on $n = 3$, construct a table with 8 ($= 2^3$) entries. The entry for subnet 1 is the all-0s subnet. Additional entries in the table are successive increments of the subnet bits, as shown in Table 19. The host bits used for subnetting are underlined.

Table 19. Binary subnetting technique for network ID 192.168.0.0

Subnet	Binary Representation	Subnetted Network ID
1	11000000.10101000.<u>000</u>00000.00000000	192.168.0.0/19
2	11000000.10101000.<u>001</u>00000.00000000	192.168.32.0/19
3	11000000.10101000.<u>010</u>00000.00000000	192.168.64.0/19
4	11000000.10101000.<u>011</u>00000.00000000	192.168.96.0/19
5	11000000.10101000.<u>100</u>00000.00000000	192.168.128.0/19
6	11000000.10101000.<u>101</u>00000.00000000	192.168.160.0/19
7	11000000.10101000.<u>110</u>00000.00000000	192.168.192.0/19
8	11000000.10101000.<u>111</u>00000.00000000	192.168.224.0/19

Decimal subnetting procedure

1. Based on *n*, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. The first column is the subnet number (starting with 1), the second column is the

decimal (Base 10 numbering system) representation of the 32-bit subnetted network ID, and the third column is the dotted decimal representation of the subnetted network ID.

2. Convert the network ID (w.x.y.z) being subnetted from dotted decimal notation to N, a decimal representation of the 32-bit network ID.
3. `N = w*16777216 + x*65536 + y*256 + z`
4. Compute the increment value I based on *h*, the number of host bits remaining.
5. $I = 2^h$
6. In the first table entry, the decimal representation of the subnetted network ID is N and the subnetted network ID will be *w.x.y.z* with its new subnet mask.

7. In the next table entry, add I to the previous table entry's decimal representation.

8. Convert the decimal representation of the subnetted network ID to dotted decimal notation (*W.X.Y.Z*) through the following formula (where *s* is the decimal representation of the subnetted network ID):
9. `W = INT(s/16777216)`
10. `X = INT((s mod(16777216))/65536)`
11. `Y = INT((s mod(65536))/256)`
12. `Z = s mod(256)`

    INT( ) denotes integer division and mod( ) denotes the modulus, the remainder upon division.

13. Repeat steps 5 and 6 until the table is complete.

As an example, a 3-bit subnet of the private network ID 192.168.0.0 is needed. Based on n = 3, we construct a table with 8 entries. The entry for subnet 1 is the all-0s subnet. N, the decimal representation of 192.168.0.0, is 3232235520, the result of 192*16777216 + 168*65536. Since there are 13 host bits remaining, the increment I is 2^{13} = 8192. Additional entries in the table are successive increments of 8192, as shown in Table 20.

Table 20. Decimal subnetting technique for network ID 192.168.0.0

Subnet	Decimal Representation	Subnetted Network ID
1	3232235520	192.168.0.0/19
2	3232243712	192.168.32.0/19
3	3232251904	192.168.64.0/19
4	3232260096	192.168.96.0/19
5	3232268288	192.168.128.0/19
6	3232276480	192.168.160.0/19
7	3232284672	192.168.192.0/19
8	3232292864	192.168.224.0/19

The all-zeros and all-ones subnets

RFC 950 originally forbade the use of the subnetted network IDs where the bits being used for subnetting are set to all 0s (the *all-zeros subnet*) and all 1s (the *all-ones subnet*). The all-zeros subnet caused problems for early routing protocols and the all-ones subnet conflicts with a special broadcast address called the *all-subnets directed broadcast address*.

However, RFC 1812 now permits the use of the all-zeros and all-ones subnets in a Classless Interdomain Routing (CIDR)-compliant environment. CIDR-compliant environments use modern routing protocols which do not have a problem with the all-zeros subnet and the use of the all-subnets directed broadcast has been deprecated.

Before you use the all-zeros and all-ones subnets, verify that they are supported by your hosts and routers. Windows® supports the use of the all-zeros and all-ones subnets.

Step 3: enumerating IP addresses for each subnetted network ID

Based on the enumeration of the subnetted network IDs, you must now list the valid IP addresses for new subnetted network IDs. To list each IP address individually would be too tedious. Instead, we will enumerate the IP addresses for each subnetted network ID by defining the range of IP addresses (the first and the last) for each subnetted network ID. There are two main approaches:

- Binary—Write down the first and last IP address for each subnetted network ID and convert to dotted decimal notation.

- Decimal—Add values incrementally, corresponding to the first and last IP addresses for each subnetted network ID and convert to dotted decimal notation.

Either method produces the same result—the range of IP addresses for each subnetted network ID.

Binary procedure

1. Based on *n*, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. Alternately, add two columns to the previous table used for enumerating the subnetted network IDs. The first column is the subnet number (starting with 1), the second column is the binary representation of the first and last IP address for the subnetted network ID, and the third column is the dotted decimal representation of the first and last IP address of the subnetted network ID.

2. For each binary representation, the first IP address is the address where all the host bits are set to 0 except for the last host bit. The last IP address is the address where all the host bits are set to 1 except for the last host bit.

3. Convert the binary representation to dotted decimal notation.

4. Repeat steps 2 and 3 until the table is complete.

As an example, the range of IP addresses for the 3-bit subnetting of 192.168.0.0 is shown in Table 21. The bits used for subnetting are underlined.

Table 21. Binary enumeration of IP addresses

Subnet	Binary Representation	Range of IP Addresses
1	11000000.10101000.<u>000</u>00000.00000001 - 11000000.10101000.<u>000</u>11111.11111110	192.168.0.1 - 192.168.31.254
2	11000000.10101000.<u>001</u>00000.00000001 - 11000000.10101000.<u>001</u>11111.11111110	192.168.32.1 - 192.168.63.254
3	11000000.10101000.<u>010</u>00000.00000001 - 11000000.10101000.<u>010</u>11111.11111110	192.168.64.1 - 192.168.95.254
4	11000000.10101000.01100000.00000001	192.168.96.1

	11000000.10101000.01111111.11111110	192.168.127.254
5	11000000.10101000.10000000.00000001 - 11000000.10101000.10011111.11111110	192.168.128.1 - 192.168.159.254
6	11000000.10101000.10100000.00000001 - 11000000.10101000.10111111.11111110	192.168.160.1 - 192.168.191.254
7	11000000.10101000.11000000.00000001 - 11000000.10101000.11011111.11111110	192.168.192.1 - 192.168.223.254
8	11000000.10101000.11100000.00000001 - 11000000.10101000.11111111.11111110	192.168.224.1 - 192.168.255.254

Decimal procedure

1. Based on *n*, the number of host bits chosen for subnetting, create a 3-column table with 2^n entries. Alternately, add two columns to the previous table used for enumerating the subnetted network IDs. The first column is the subnet number (starting with 1), the second column is the decimal representation of the first and last IP address for the subnetted network ID, and the third column is the dotted decimal representation of the first and last IP address of the subnetted network ID.

2. Compute the increment value J based on *h*, the number of host bits remaining.
3. $J = 2^h - 2$
4. For each decimal representation, the first IP address is N + 1, where N is the decimal representation of the subnetted network ID. The last IP address is N + J.

5. Convert the decimal representation of the first and last IP addresses to dotted decimal notation (*W.X.Y.Z*) through the following formula (where *s* is the decimal representation of the first or last IP address):
6. $W = INT(s/16777216)$
7. $X = INT((s \bmod(16777216))/65536)$
8. $Y = INT((s \bmod(65536))/256)$
9. $Z = s \bmod(256)$

   INT( ) denotes integer division and mod( ) denotes the modulus, the remainder upon division.

10. Repeat steps 3 and 4 until the table is complete.

As an example, the range of IP addresses for the 3-bit subnetting of 192.168.0.0 is shown in Table 22. The increment J is $2^{13} - 2 = 8190$.

Table 22. Decimal enumeration of IP addresses

Subnet	Decimal Representation	Range of IP Addresses
1	3232235521 – 3232243710	192.168.0.1 - 192.168.31.254
2	3232243713 – 3232251902	192.168.32.1 - 192.168.63.254
3	3232251905 – 3232260094	192.168.64.1 - 192.168.95.254
4	3232260097 – 3232268286	192.168.96.1 - 192.168.127.254
5	3232268289 – 3232276478	192.168.128.1 - 192.168.159.254

6	3232276481 – 3232284670	192.168.160.1 - 192.168.191.254
7	3232284673 – 3232292862	192.168.192.1 - 192.168.223.254
8	3232292865 – 3232301054	192.168.224.1 - 192.168.255.254

Variable-Length Subnetting

One of the original uses for subnetting was to subdivide a class-based network ID into a series of equal-sized subnets. For example, a 4-bit subnetting of a class B network ID produced 16 equal-sized subnets (using the all-ones and all-zeros subnets). However, subnetting is a general method of utilizing host bits to express subnets and does not require equal-sized subnets.

Subnets of different size can exist within a class-based network ID. This is well-suited to real world environments, where networks of an organization contain different amounts of hosts, and different-sized subnets are needed to minimize the wasting of IP addresses. The creation and deployment of various-sized subnets of a network ID is known as *variable length subnetting* and uses *variable length subnet masks* (VLSM).

Variable length subnetting is a technique of allocating subnetted network IDs that use subnet masks of different sizes. However, all subnetted network IDs are unique and can be distinguished from each other by their corresponding subnet mask.

The mechanics of variable length subnetting are essentially that of performing subnetting on a previously subnetted network ID. When subnetting, the network ID bits are fixed and a certain amount of host bits are chosen to express subnets. With variable length subnetting, the network ID being subnetted has already been subnetted.

Variable-length subnetting example

For example, given the class-based network ID of 135.41.0.0/16, a required configuration is to reserve half of the addresses for future use, create 15 subnets with up to 2,000 hosts, and 8 subnets with up to 250 hosts.

Reserve half of the addresses for future use

To reserve half of the addresses for future use, a 1-bit subnetting of the class-based network ID of 135.41.0.0 is done, producing 2 subnets, 135.41.0.0/17 and 135.41.128.0/17. The subnet 135.41.0.0/17 is chosen as the portion of the addresses which are reserved for future use.

Table 23 shows one subnet with up to 32,766 hosts.

Table 23. Reserving half the addresses for future use

Subnet Number	Network ID (dotted decimal)	Network ID (network prefix)
1	135.41.0.0, 255.255.128.0	135.41.0.0/17

Fifteen subnets with up to 2,000 hosts

To achieve a requirement of 15 subnets with approximately 2,000 hosts, a 4-bit subnetting of the subnetted network ID of 135.41.128.0/17 is done. This produces 16 subnets (135.41.128.0/21, 135.41.136.0/21 . . . 135.41.240.0/21, 135.41.248.0/21), allowing up to 2,046 hosts per subnet. The

first 15 subnetted network IDs (135.41.128.0/21 to 135.41.240.0/21) are chosen as the network IDs, which fulfills the requirement.

Table 24 illustrates 15 subnets with up to 2,000 hosts.

Table 24. 15 Subnets with up to 2,046 hosts

Subnet Number	Network ID (dotted decimal)	Network ID (network prefix)
1	135.41.128.0, 255.255.248.0	135.41.128.0/21
2	135.41.136.0, 255.255.248.0	135.41.136.0/21
3	135.41.144.0, 255.255.248.0	135.41.144.0/21
4	135.41.152.0, 255.255.248.0	135.41.152.0/21
5	135.41.160.0, 255.255.248.0	135.41.160.0/21
6	135.41.168.0, 255.255.248.0	135.41.168.0/21
7	135.41.176.0, 255.255.248.0	135.41.176.0/21
8	135.41.184.0, 255.255.248.0	135.41.184.0/21
9	135.41.192.0, 255.255.248.0	135.41.192.0/21
10	135.41.200.0, 255.255.248.0	135.41.200.0/21
11	135.41.208.0, 255.255.248.0	135.41.208.0/21
12	135.41.216.0, 255.255.248.0	135.41.216.0/21
13	135.41.224.0, 255.255.248.0	135.41.224.0/21
14	135.41.232.0, 255.255.248.0	135.41.232.0/21
15	135.41.240.0, 255.255.248.0	135.41.240.0/21

8 subnets with up to 250 hosts

To achieve a requirement of 8 subnets with up to 250 hosts, a 3-bit subnetting of subnetted network ID of 135.41.248.0/21 is done, producing 8 subnets (135.41.248.0/24, 135.41.249.0/24 . . . 135.41.254.0/24, 135.41.255.0/24) and allowing up to 254 hosts per subnet. All 8 subnetted network IDs (135.41.248.0/24 to 135.41.255.0/24) are chosen as the network IDs, which fulfills the requirement.

Table 25 illustrates 8 subnets with approximately 250 hosts.

Table 25. 8 subnets with up to 254 hosts

Subnet Number	Network ID (dotted decimal)	Network ID (network prefix)
1	135.41.248.0, 255.255.255.0	135.41.248.0/24
2	135.41.249.0, 255.255.255.0	135.41.249.0/24

3	135.41.250.0, 255.255.255.0	135.41.250.0/24
4	135.41.251.0, 255.255.255.0	135.41.251.0/24
5	135.41.252.0, 255.255.255.0	135.41.252.0/24
6	135.41.253.0, 255.255.255.0	135.41.253.0/24
7	135.41.254.0, 255.255.255.0	135.41.254.0/24
8	135.41.255.0, 255.255.255.0	135.41.255.0/24

The variable length subnetting of 135.41.0.0/16 is shown graphically in Figure 10.

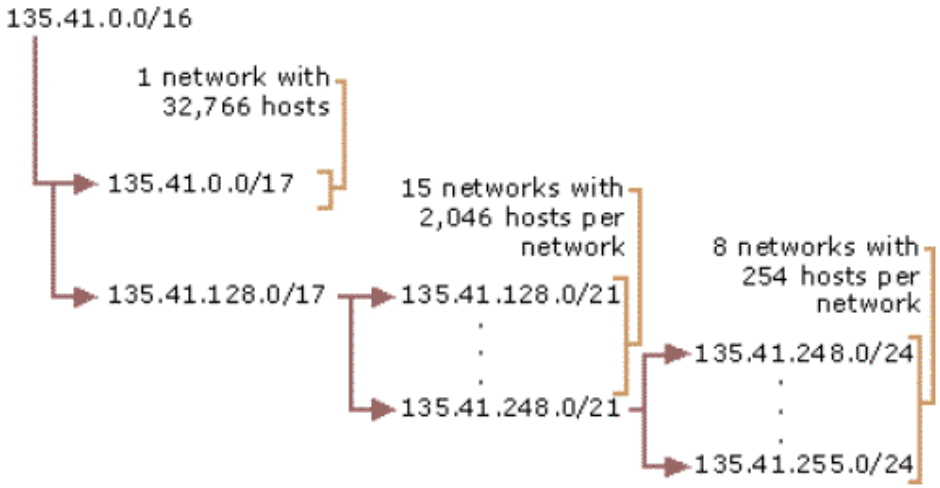

Figure 10. Variable-length subnetting of 135.41.0.0/16

Note In dynamic routing environments, variable length subnetting can only be deployed where the subnet mask is advertised along with the network ID. Routing Information Protocol (RIP) for IP version 1 does not support variable-length subnetting. RIP for IP version 2, Open Short Path First (OSPF), and BGPv4 all support variable length subnetting.

Supernetting and Classless Interdomain Routing

With the recent growth of the Internet, it became clear to the Internet authorities that the class B network IDs would soon be depleted. For most organizations, a class C network ID does not contain enough host IDs and a class B network ID has enough bits to provide a flexible subnetting scheme within the organization.

The Internet authorities devised a new method of assigning network IDs to prevent the depletion of class B network IDs. Rather than assigning a class B network ID, the Internet Network Information Center (InterNIC) assigns a range of class C network IDs that contain enough network and host IDs for the organization's needs. This is known as supernetting. For example, rather than allocating a class B network ID to an organization that has up to 2,000 hosts, the InterNIC allocates a range of 8 class C network IDs. Each class C network ID accommodates 254 hosts, for a total of 2,032 host IDs.

While this technique helps conserve class B network IDs, it creates a new problem. Using conventional routing techniques, the routers on the Internet now must have 8 class C network ID entries in their routing tables to route IP packets to the organization. To prevent Internet routers from becoming overwhelmed with routes, a technique called *Classless Interdomain Routing* (CIDR) is used

to collapse multiple network ID entries into a single entry corresponding to all of the class C network IDs allocated to that organization.

Conceptually, CIDR creates the routing table entry: {Starting Network ID, count}, where Starting Network ID is the first class C network ID and the count is the number of class C network IDs allocated. In practice, a supernetted subnet mask is used to convey the same information. To express the situation where 8 class C network IDs are allocated starting with Network ID 220.78.168.0:

Starting Network ID	220.78.168.0	100111100 01001110 10101000 00000000
Ending Network ID	220.78.175.0	100111100 01001110 10101111 00000000

Note that the first 21 bits (underlined) of all the above Class C network IDs are the same. The last three bits of the third octet vary from 000 to 111. The CIDR entry in the routing tables of the Internet routers becomes:

Network ID	Subnet Mask	Subnet Mask (binary)
220.78.168.0	255.255.248.0	1111111111 11111111 11111000 00000000

In network prefix notation, the CIDR entry is 220.78.168.0/21.

A block of addresses using CIDR is known as a *CIDR block*.

Note Since subnet masks are used to express the count, class-based network IDs must be allocated in groups corresponding to powers of two.

In order to support CIDR, routers must be able to exchange routing information in the form of {Network ID, Subnet Mask} pairs. RIP for IP version 2, OSPF, and BGPv4 are routing protocols that support CIDR. RIP for IP version 1 does not support CIDR.

The address space perspective

The use of CIDR to allocate addresses promotes a new perspective on IP network IDs. In the above example, the CIDR block {220.78.168.0, 255.255.248.0} can be thought of in two ways:

- A block of 8 class C network IDs.
- An address space in which 21 bits are fixed and 11 bits are assignable.

In the latter perspective, IP network IDs lose their class-based heritage and become separate IP address spaces, subsets of the original IP address space defined by the 32-bit IP address. Each IP network ID (class-based, subnetted, CIDR block) is an address space in which certain bits are fixed (the network ID bits) and certain bits are variable (the host bits). The host bits are assignable as host IDs or, using subnetting techniques, can be used in whatever manner best suits the needs of the organization.

Public and Private Addresses

If your intranet is not connected to the Internet, any IP addressing can be deployed. If direct (routed) or indirect (proxy or translator) connectivity to the Internet is desired, then there are two types of addresses employed on the Internet, public addresses and private addresses.

Public addresses

Public addresses are assigned by InterNIC and consist of class-based network IDs or blocks of CIDR-based addresses (called CIDR blocks) that are guaranteed to be globally unique to the Internet.

When the public addresses are assigned, routes are programmed into the routers of the Internet so that traffic to the assigned public addresses can reach their locations. Traffic to destination public addresses are reachable on the Internet.

For example, when an organization is assigned a CIDR block in the form of a network ID and subnet mask, that {network ID, subnet mask} pair also exists as a route in the routers of the Internet. IP packets destined to an address within the CIDR block are routed to the proper destination.

Illegal addresses

Private intranets that have no intent to connect to the Internet can choose any addresses they want, even public addresses that have been assigned by the InterNIC. If an organization later decides to connect to the Internet, its current address scheme may include addresses already assigned by the InterNIC to other organizations. These addresses would be duplicate or conflicting addresses and are known as *illegal addresses*. Connectivity from illegal addresses to Internet locations is not possible.

For example, a private organization chooses to use 207.46.130.0/24 as its intranet address space. The public address space 207.46.130.0/24 has been assigned to the Microsoft corporation and routes exist on the Internet routers to route all packets destined to IP addresses on 207.46.130.0/24 to Microsoft routers. As long as the private organization does not connect to the Internet, there is no problem, since the two address spaces are on separate IP internetworks. If the private organization then connected directly to the Internet and continued to use 207.46.130.0/24 as its address space, then any Internet response traffic to locations on the 207.46.130.0/24 network would be routed to Microsoft routers, not to the routers of the private organization.

Private addresses

Each IP node requires an IP address that is globally unique to the IP internetwork. In the case of the Internet, each IP node on a network connected to the Internet requires an IP address that is globally unique to the Internet. As the Internet grew, organizations connecting to the Internet required a public address for each node on their intranets. This requirement placed a huge demand on the pool of available public addresses.

When analyzing the addressing needs of organizations, the designers of the Internet noted that, for many organizations, most of the hosts on the organization's intranet did not require direct connectivity to Internet hosts. Those hosts that did require a specific set of Internet services, such as the World Wide Web access and e-mail, typically access the Internet services through application layer gateways, such as proxy servers and e-mail servers. The result is that most organizations only required a small amount of public addresses for those nodes (such as proxies, routers, firewalls, and translators) that were directly connected to the Internet.

For the hosts within the organization that do not require direct access to the Internet, IP addresses that do not duplicate already-assigned public addresses are required. To solve this addressing problem, the Internet designers reserved a portion of the IP address space and named this space the *private address space*. An IP address in the private address space is never assigned as a public address. IP addresses within the private address space are known as *private addresses*. Because the public and private address spaces do not overlap, private addresses never duplicate public addresses.

The private address space specified in RFC 1597 is defined by the following three address blocks:

- 10.0.0.0/8

The 10.0.0.0/8 private network is a class A network ID that allows the following range of valid IP addresses: 10.0.0.1 to 10.255.255.254. The 10.0.0.0/8 private network has 24 host bits which can be used for any subnetting scheme within the private organization.

- 172.16.0.0/12

  The 172.16.0.0/12 private network can be interpreted either as a block of 16 class B network IDs or as a 20-bit assignable address space (20 host bits) which can be used for any subnetting scheme within the private organization. The 172.16.0.0/12 private network allows the following range of valid IP addresses: 172.16.0.1 to 172.31.255.254.

- 192.168.0.0/16

  The 192.168.0.0/16 private network can be interpreted either as a block of 256 class C network IDs or as a 16-bit assignable address space (16 host bits), which can be used for any subnetting scheme within the private organization. The 192.168.0.0/16 private network allows the following range of valid IP addresses: 192.168.0.1 to 192.168.255.254.

The result of many organizations using private addresses is that the private address space is reused, helping to prevent the depletion of public addresses.

Since the IP addresses in the private address space will never be assigned by the InterNIC as public addresses, there will never exist routes in the Internet routers for private addresses. Traffic to destination private addresses are not reachable on the Internet. Therefore, Internet traffic from a host that has a private address must either send its requests to an application layer gateway (such as a proxy server), which has a valid public address, or have its private address translated into a valid public address by a *network address translator* (NAT) before it is sent on the Internet.

Well, that's it.

For more information on the TCP/IP protocol suite, see the following:

- Comer, Douglas, *Internetworking with TCP/IP, Vol 1*, 3rd Edition. Prentice Hall, 1996.
- Siyan, Karanjit S., *Inside TCP/IP*, 3rd Edition. New Riders Publishing, 1997.
- Stevens, W. Richard, *TCP/IP Illustrated, Volume 1, The Protocols*. Addison-Wesley, 1994.
- Thomas Lee, Joseph Davies, *Windows 2000 TCP/IP Protocols and Services Technical Reference.* Microsoft Press, 2000