Appendix B

IPv4 Supplement

This appendix contains job aids and supplementary information that cover the following topics:

- IPv4 Addresses and Subnetting Job Aid
- Decimal-to-Binary Conversion Chart
- IPv4 Addressing Review
- IPv4 Access Lists
- IPv4 Address Planning
- Hierarchical Addressing Using Variable-Length Subnet Masks
- Route Summarization
- Classless Interdomain Routing

This Internet Protocol Version 4 (IPv4) supplement provides job aids and supplementary information intended for your use when working with IPv4 addresses.

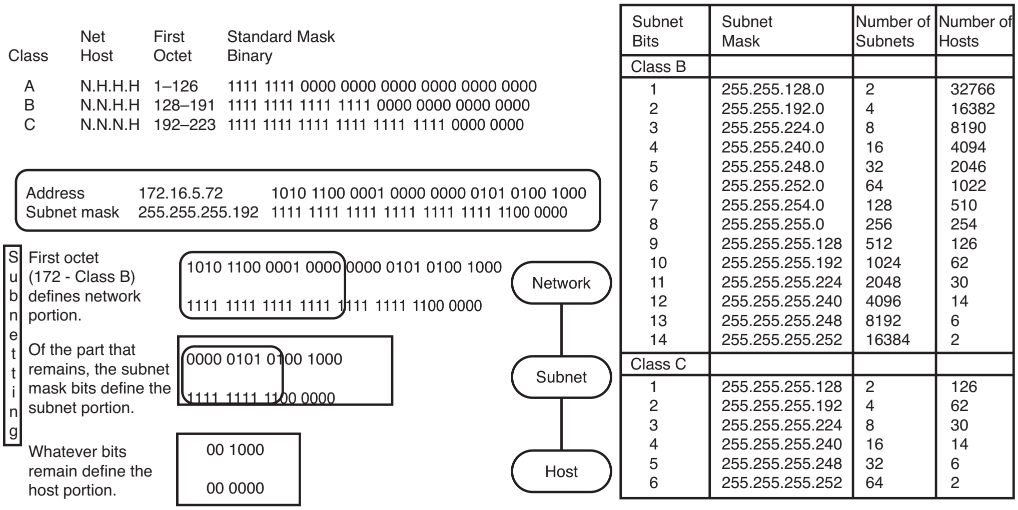
Note In this appendix, the term *IP* refers to IPv4.

This appendix includes an IP addressing and subnetting job aid and a decimal-to-binary conversion chart. The information in the sections "IPv4 Addressing Review" and "IPv4 Access Lists" should serve as a review of the fundamentals of IP addressing and of the concepts and configuration of access lists, respectively.

The remainder of the sections relate to IP address planning. Scalable, well-behaved networks are not accidental. They are the result of good network design and effective implementation planning. A key element for effective scalable network implementation is a well-conceived and scalable IP addressing plan, as described in the "IPv4 Address Planning" section. Variable-length subnet masking (VLSM), route summarization, and classless interdomain routing (CIDR) are then explored. VLSM allows the network administrator to subnet a previously subnetted address to make the best use of the available address space. Summarization and CIDR are advanced IP addressing techniques that keep the size of the routing tables from increasing as networks grow.

IPv4 Addresses and Subnetting Job Aid

Figure B-1 is a job aid to help you with various aspects of IP addressing, including how to distinguish address classes, the number of subnets and hosts available with various subnet masks, and how to interpret IP addresses.



Class	Net Host	First Octet	Standard Mask Binary
A	N.H.H.H	1–126	1111 1111 0000 0000 0000 0000 0000 0000
B	N.N.H.H	128–191	1111 1111 1111 1111 0000 0000 0000 0000
C	N.N.N.H	192–223	1111 1111 1111 1111 1111 1111 0000 0000

Address	172.16.5.72	1010 1100 0001 0000 0000 0101 0100 1000
Subnet mask	255.255.255.192	1111 1111 1111 1111 1111 1111 1100 0000

First octet (172 - Class B) defines network portion.

Of the part that remains, the subnet mask bits define the subnet portion.

Whatever bits remain define the host portion.

Subnet Bits	Subnet Mask	Number of Subnets	Number of Hosts
Class B			
1	255.255.128.0	2	32766
2	255.255.192.0	4	16382
3	255.255.224.0	8	8190
4	255.255.240.0	16	4094
5	255.255.248.0	32	2046
6	255.255.252.0	64	1022
7	255.255.254.0	128	510
8	255.255.255.0	256	254
9	255.255.255.128	512	126
10	255.255.255.192	1024	62
11	255.255.255.224	2048	30
12	255.255.255.240	4096	14
13	255.255.255.248	8192	6
14	255.255.255.252	16384	2
Class C			
1	255.255.255.128	2	126
2	255.255.255.192	4	62
3	255.255.255.224	8	30
4	255.255.255.240	16	14
5	255.255.255.248	32	6
6	255.255.255.252	64	2

Figure B-1 *IP Addresses and Subnetting Job Aid.*

Decimal-to-Binary Conversion Chart

Table B-1 can be used to convert from decimal to binary and from binary to decimal.

Table B-1 *Decimal-to-Binary Conversion Chart*

Decimal	Binary	Decimal	Binary	Decimal	Binary
0	00000000	28	00011100	56	00111000
1	00000001	29	00011101	57	00111001
2	00000010	30	00011110	58	00111010
3	00000011	31	00011111	59	00111011
4	00000100	32	00100000	60	00111100
5	00000101	33	00100001	61	00111101
6	00000110	34	00100010	62	00111110
7	00000111	35	00100011	63	00111111
8	00001000	36	00100100	64	01000000
9	00001001	37	00100101	65	01000001
10	00001010	38	00100110	66	01000010
11	00001011	39	00100111	67	01000011
12	00001100	40	00101000	68	01000100
13	00001101	41	00101001	69	01000101
14	00001110	42	00101010	70	01000110
15	00001111	43	00101011	71	01000111
16	00010000	44	00101100	72	01001000
17	00010001	45	00101101	73	01001001
18	00010010	46	00101110	74	01001010
19	00010011	47	00101111	75	01001011
20	00010100	48	00110000	76	01001100
21	00010101	49	00110001	77	01001101
22	00010110	50	00110010	78	01001110
23	00010111	51	00110011	79	01001111
24	00011000	52	00110100	80	01010000
25	00011001	53	00110101	81	01010001
26	00011010	54	00110110	82	01010010
27	00011011	55	00110111	83	01010011

continues

Table B-1 *Decimal-to-Binary Conversion Chart (continued)*

Decimal	Binary	Decimal	Binary	Decimal	Binary
84	01010100	112	01110000	140	10001100
85	01010101	113	01110001	141	10001101
86	01010110	114	01110010	142	10001110
87	01010111	115	01110011	143	10001111
88	01011000	116	01110100	144	10010000
89	01011001	117	01110101	145	10010001
90	01011010	118	01110110	146	10010010
91	01011011	119	01110111	147	10010011
92	01011100	120	01111000	148	10010100
93	01011101	121	01111001	149	10010101
94	01011110	122	01111010	150	10010110
95	01011111	123	01111011	151	10010111
96	01100000	124	01111100	152	10011000
97	01100001	125	01111101	153	10011001
98	01100010	126	01111110	154	10011010
99	01100011	127	01111111	155	10011011
100	01100100	128	10000000	156	10011100
101	01100101	129	10000001	157	10011101
102	01100110	130	10000010	158	10011110
103	01100111	131	10000011	159	10011111
104	01101000	132	10000100	160	10100000
105	01101001	133	10000101	161	10100001
106	01101010	134	10000110	162	10100010
107	01101011	135	10000111	163	10100011
108	01101100	136	10001000	164	10100100
109	01101101	137	10001001	165	10100101
110	01101110	138	10001010	166	10100110
111	01101111	139	10001011	167	10100111

Table B-1 *Decimal-to-Binary Conversion Chart*

Decimal	Binary	Decimal	Binary	Decimal	Binary
168	10101000	196	11000100	224	11100000
169	10101001	197	11000101	225	11100001
170	10101010	198	11000110	226	11100010
171	10101011	199	11000111	227	11100011
172	10101100	200	11001000	228	11100100
173	10101101	201	11001001	229	11100101
174	10101110	202	11001010	230	11100110
175	10101111	203	11001011	231	11100111
176	10110000	204	11001100	232	11101000
177	10110001	205	11001101	233	11101001
178	10110010	206	11001110	234	11101010
179	10110011	207	11001111	235	11101011
180	10110100	208	11010000	236	11101100
181	10110101	209	11010001	237	11101101
182	10110110	210	11010010	238	11101110
183	10110111	211	11010011	239	11101111
184	10111000	212	11010100	240	11110000
185	10111001	213	11010101	241	11110001
186	10111010	214	11010110	242	11110010
187	10111011	215	11010111	243	11110011
188	10111100	216	11011000	244	11110100
189	10111101	217	11011001	245	11110101
190	10111110	218	11011010	246	11110110
191	10111111	219	11011011	247	11110111
192	11000000	220	11011100	248	11111000
193	11000001	221	11011101	249	11111001
194	11000010	222	11011110	250	11111010
195	11000011	223	11011111	251	11111011

continues

Table B-1 *Decimal-to-Binary Conversion Chart (continued)*

Decimal	Binary	Decimal	Binary	Decimal	Binary
252	11111100	254	11111110		
253	11111101	255	11111111		

IPv4 Addressing Review

This section reviews the basics of IPv4 addresses:

- Converting IP addresses between decimal and binary
    - Determining an IP address class
    - Private addresses
    - Extending an IP classful address using subnet masks
    - Calculating a subnet mask
    - Calculating the networks for a subnet mask
    - Using prefixes to represent a subnet mask

Converting IP Addresses Between Decimal and Binary

An *IP address* is a 32-bit, two-level hierarchical number. It is hierarchical because the first portion of the address represents the network, and the second portion of the address represents the node (or host).

The 32 bits are grouped into 4 octets, with 8 bits per octet. The value of each octet ranges from 0 to 255 decimal, or 00000000 to 11111111 binary. IP addresses are usually written in dotted-decimal notation, which means that each octet is written in decimal notation and dots are placed between the octets. Figure B-2 shows how you convert an octet of an IP address in binary to decimal notation.

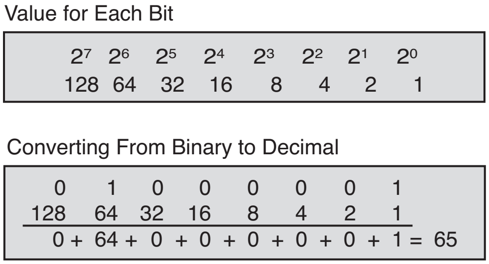


Figure B-2 *Converting an Octet of an IP Address from Binary to Decimal.*

It is important that you understand how this conversion is done because it is used when calculating subnet masks, a topic discussed later in this section.

Figure B-3 shows three examples of converting IP addresses between binary and decimal.

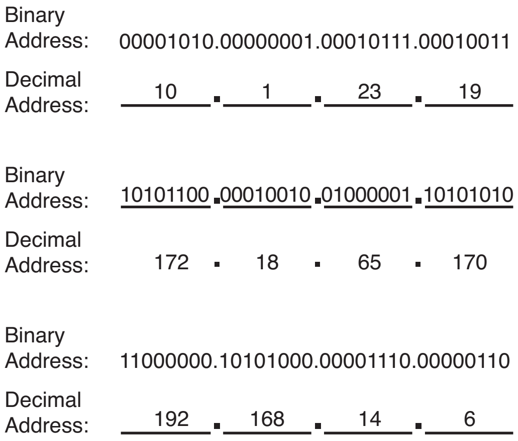

Binary Address: 00001010.00000001.00010111.00010011
Decimal Address: 10 . 1 . 23 . 19
Binary Address: 10101100 . 00010010 . 01000001 . 10101010
Decimal Address: 172 . 18 . 65 . 170
Binary Address: 11000000.10101000.00001110.00000110
Decimal Address: 192 . 168 . 14 . 6

Figure B-3 *Converting IP Addresses Between Binary and Decimal.*

Now that you understand the decimal-to-binary and binary-to-decimal conversion processes, use the following sections to review address classes and the uses of subnet masks.

Determining an IP Address Class

To accommodate large and small networks, the 32-bit IP addresses are segregated into Classes A through E. The first few bits of the first octet determine the class of an address. This then determines how many network bits and host bits are in the address. Figure B-4 illustrates the bits for Class A, B, and C addresses. Each address class allows for a certain number of network addresses and a certain number of host addresses within a network. Table B-2 shows the address range, the number of networks, and the number of hosts for each of the classes. (Note that Class D and E addresses are used for purposes other than addressing hosts.)

Using classes to denote which portion of the address represents the network number and which portion represents the node or host address is called classful addressing. Several issues must be addressed with classful addressing. First, the number of available Class A, B, and C addresses is finite. Another problem is that not all classes are useful for a midsize organization, as illustrated in Table B-2. As can be expected, the Class B range best accommodates a majority of today's organizational network topologies. Subnet masks, as described later in this appendix, in the "Extending an IP Classful Address Using Subnet Masks" section, were introduced to maximize the use of the IP addresses an organization receives, regardless of the class.

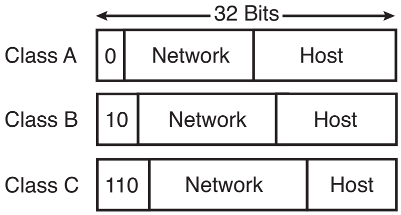


Figure B-4 *Determining an IP Address Class from the First Few Bits of an Address.*

Table B-2 *IP Address Classes*

Class	Address Range	Number of Networks	Number of Hosts
A[1]	1.0.0.0 to 126.0.0.0	126 (2^7 – 2 that are reserved)	16,777,214
B	128.0.0.0 to 191.255.0.0	16,386 (2^{14})	65,532
C	192.0.0.0 to 223.255.255.0	Approximately 2 million (2^{21})	254
D	224.0.0.0 to 239.255.255.255	Reserved for multicast addresses	—
E	240.0.0.0 to 254.255.255.255	Reserved for research	—

[1] The network 127.0.0.0 (any address starting with decimal 127) is reserved for loopback. Network 0.0.0.0 is also reserved and cannot be used to address devices.

Private Addresses

Requests For Comments (RFC) 1918, *Address Allocation for Private Internets*, has set aside the following IPv4 address space for private use:

- **Class A network—**10.0.0.0 to 10.255.255.255
- **Class B network—**172.16.0.0 to 172.31.255.255
- **Class C network—**192.168.0.0 to 192.168.255.255

Note RFCs are available at http://www.rfc-editor.org/rfcsearch.html.

Private addresses are reserved IPv4 addresses to be used only internally within a company's network. These private addresses are not to be used on the Internet, so they must be mapped to a company's external registered address when the company sends anything to a recipient on the Internet.

Note The examples in this book use only private addressing.

Extending an IP Classful Address Using Subnet Masks

RFC 950, *Internet Standard Subnetting Procedure*, was written to address the IP address shortage. It proposed a procedure, called *subnet masking*, for dividing Class A, B, and C addresses into smaller pieces, thereby increasing the number of possible networks.

A subnet mask is a 32-bit value that identifies which address bits represent network bits and which represent host bits. In other words, the router does not determine the network portion of the address by looking at the value of the first octet. Instead, it looks at the subnet mask that is associated with the address. In this way, subnet masks let you extend the usage of an IP address. This is one way of making an IP address a three-level hierarchy, as shown in Figure B-5.

To create a subnet mask for an address, use a binary 1 for each bit that you want to represent the network or subnet portion of the address, and use a binary 0 for each bit that you want to represent the node portion of the address. Note that the 1s in the mask are contiguous. The default subnet masks for Class A, B, and C addresses are as shown Table B-3.

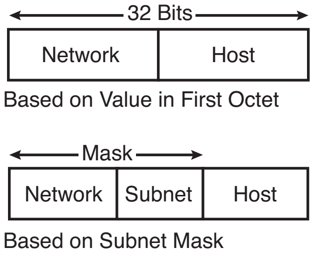


Figure B-5 *A Subnet Mask Determines How an IP Address Is Interpreted.*

Table B-3 *IP Address Default Subnet Masks*

Class	Default Mask in Binary	Default Mask in Decimal
A	11111111.00000000.00000000.00000000	255.0.0.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.11111111.11111111.00000000	255.255.255.0

Calculating a Subnet Mask

When contiguous 1s are added to the default mask, making the all-1s field in the mask longer, the definition of the network part of an IP address is extended to include subnets. However, adding bits to the network part of an address decreases the number of bits in the host part. Thus, creating additional networks (subnets) is done at the expense of the number of host devices that can occupy each network segment.

The number of subnetworks created is calculated by the formula 2^s, where *s* is the number of bits by which the default mask was extended.

Note Subnet 0 (where all the subnet bits are 0) must be explicitly allowed using the **ip subnet-zero** global configuration command in Cisco IOS releases earlier than 12.0. In Cisco IOS Release 12.0 and later, this command is enabled by default.

The number of hosts available is calculated by the formula $2^h - 2$, where h is the number of bits in the host portion. The two addresses subtracted in this host formula are for the addresses with all 0s and all 1s in the host field. In the host field, the all-0s bit pattern is reserved as the subnet identifier (sometimes called *the wire*), and the all-1s bit pattern is reserved as a directed broadcast address, to reach all hosts on that subnet.

Because subnet masks extend the number of network addresses you can use by using bits from the host portion, you do not want to randomly decide how many additional bits to use for the network portion. Instead, you want to do some research to determine how many network addresses you need to derive from your given IP address. For example, suppose you have the IP address 172.16.0.0, and you want to configure the network shown in Figure B-6. To establish your subnet mask, do the following:

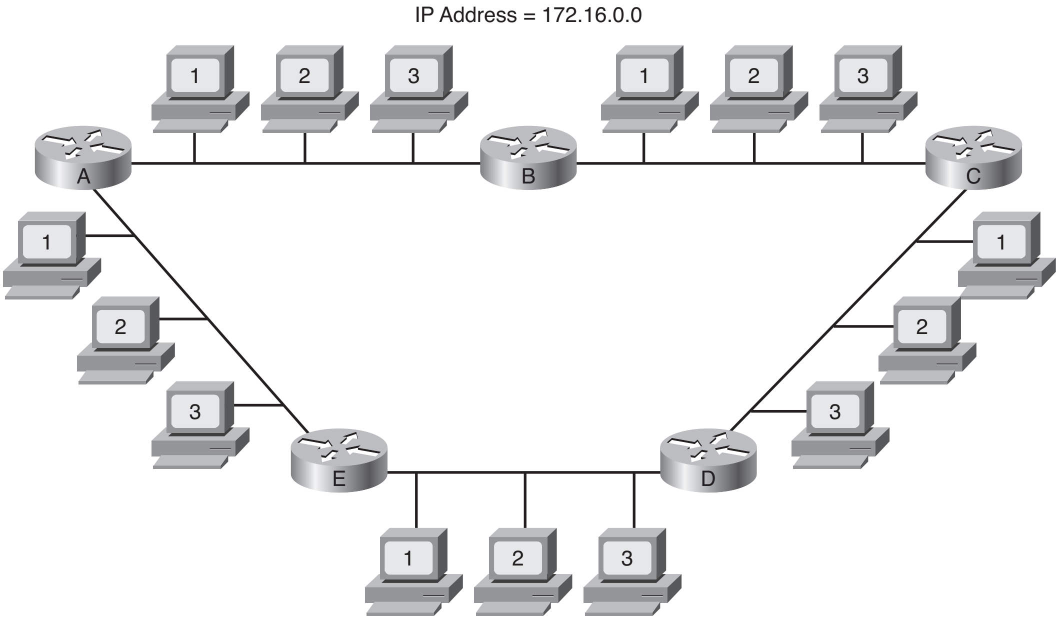


Figure B-6 *Network Used in the Subnet Mask Example.*

Step 1. Determine the number of networks (subnets) needed. Figure B-6, for example, has five networks.

Step 2. Determine how many nodes per subnet must be defined. This example has five nodes (two routers and three workstations) on each subnet.

Step 3. Determine future network and node requirements. For example, assume 100 percent growth.

Step 4. Given the information gathered in Steps 1 to 3, determine the total number of subnets required. For this example, ten subnets are required. See the earlier section “IPv4 Addresses and Subnetting Job Aid” to select the appropriate subnet mask value that can accommodate 10 networks.

No mask accommodates exactly 10 subnets. Depending on your network growth trends, you might select 4 subnet bits, resulting in a subnet mask of 255.255.240.0. The binary representation of this subnet mask is as follows:

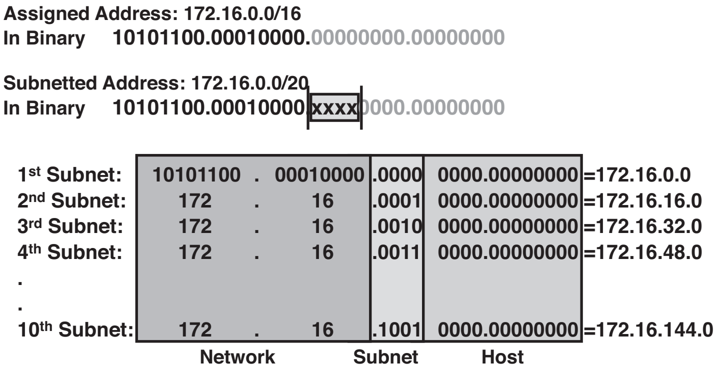
11111111.11111111.11110000.00000000

The additional 4 subnet bits would result in $2^s = 2^4 = 16$ subnets.

Calculating the Networks for a Subnet Mask

See Figure B-6. After you identify your subnet mask, you must calculate the ten subnetted network addresses to use with 172.16.0.0 255.255.240.0. One way to do this is as follows:

Step 1. Write the subnetted address in binary format, as shown at the top of Figure B-7. If necessary, use the decimal-to-binary conversion chart provided in Table B-1.



Assigned Address: 172.16.0.0/16
In Binary 10101100.00010000.00000000.00000000

Subnetted Address: 172.16.0.0/20
In Binary 10101100.00010000.xxxx0000.00000000

	Network			Subnet	Host	
1st Subnet:	10101100	.	00010000	.0000	0000.00000000	=172.16.0.0
2nd Subnet:	172	.	16	.0001	0000.00000000	=172.16.16.0
3rd Subnet:	172	.	16	.0010	0000.00000000	=172.16.32.0
4th Subnet:	172	.	16	.0011	0000.00000000	=172.16.48.0
.						
.						
10th Subnet:	172	.	16	.1001	0000.00000000	=172.16.144.0

Figure B-7 *Calculating the Subnets Shown in Figure B-6.*

Step 2. On the binary address, draw a line between the 16th and 17th bits, as shown in Figure B-7. This is the transition point between the network bits and the subnet bits. Then draw a line between the 20th and 21st bits. This is the transition point between the subnet bits and the host bits, and is the transition point between 1s and 0s in the subnet mask. Now you can focus on the target subnet bits.

Step 3. Historically, it was recommended that you begin choosing subnets from highest (from the far left bit) to lowest, so that you could leave bits available in case you need more host bits later on. However, this strategy does not allow you to adequately summarize subnet addresses, so the present recommendation is to choose subnets from lowest to highest (right to left).

When you calculate the subnet address, all host bits are set to 0. Therefore, for the first subnet, the subnet bits are 0000, and the rest of this third octet (all host bits) is 0000. To convert back to decimal, it is important to note that you must always convert an entire octet, 8 bits.

If necessary, use the decimal-to-binary conversion chart provided in Table B-1, and locate this first number. The third octet of the first subnet number is 00000000, or decimal 0. Do not forget the other 8 host bits in the fourth octet. This fourth octet is also 00000000, or decimal 0.

Step 4. (Optional) List each subnet in binary form to reduce the number of errors. This way, you will not forget where you left off in your subnet address selection.

Step 5. Calculate the second-lowest subnet number. In this case, it is 0001. When combined with the next 4 bits (the host bits) of 0000, this is binary 00010000, or decimal 16. Again, don't forget the other 8 host bits in the fourth octet. This fourth octet is again 00000000, or decimal 0.

Step 6. Continue calculating subnet numbers until you have as many as you need—in this case, 10 subnets, as shown in Figure B-7.

Using Prefixes to Represent a Subnet Mask

As discussed, subnet masks identify the number of bits in an address that represent the network, subnet, and host portions of the address. Another way of indicating this information is to use a *prefix*. A prefix is a slash (/) followed by a numeric value that is the number of bits in the network and subnet portion of the address. In other words, it is the number of contiguous 1s in the subnet mask. For example, assume you are using a subnet mask of 255.255.255.0. The binary representation of this mask is 11111111.11111111.11111111.00000000, which is 24 1s followed by eight 0s. Thus, the prefix is /24, for the 24 bits of network and subnet information, the number of 1s in the mask.

Table B-4 shows some examples of the different ways you can represent a prefix and subnet mask.

Table B-4 *Representing Subnet Masks.*

IP Address/Prefix	Subnet Mask in Decimal	Subnet Mask in Binary
192.168.112.0/21	255.255.248.0	11111111.11111111.11111000.00000000
172.16.0.0/16	255.255.0.0	11111111.11111111.00000000.00000000
10.1.1.0/27	255.255.255.224	11111111.11111111.11111111.11100000

It is important to know how to write subnet masks and prefixes because Cisco routers use both, as shown in Example B-1. You will typically be asked to input a subnet mask when configuring an IP address, but the output generated using **show** commands typically displays an IP address with a prefix.

Example B-1 *Examples of Subnet Mask and Prefix Use on Cisco Routers*

```
p1r3#show run
<Output Omitted>
interface Ethernet0
 ip address 10.64.4.1 255.255.255.0
!
interface Serial0
 ip address 10.1.3.2 255.255.255.0
<Output Omitted>

p1r3#show interface ethernet0
Ethernet0 is administratively down, line protocol is down
  Hardware is Lance, address is 00e0.b05a.d504 (bia 00e0.b05a.d504)
  Internet address is 10.64.4.1/24
<Output Omitted>

p1r3#show interface serial0
Serial0 is down, line protocol is down
  Hardware is HD64570
  Internet address is 10.1.3.2/24
<Output Omitted>
```

IPv4 Access Lists

This section reviews IPv4 access lists. It includes the following topics:

- IP access list overview
- IP standard access lists
- IP extended access lists
- Restricting virtual terminal access
- Verifying access list configuration

IP Access List Overview

Packet filtering helps control packet movement through the network, as shown in Figure B-8. Such control can help limit network traffic and restrict network use by certain users or devices. To permit packets to cross or deny packets from crossing specified router

Table B-5 *IP Access List Numbers*

Type of Access List	Range of Access List Numbers
IP standard	1 to 99 or from 1300 to 1999
IP extended	100 to 199 or from 2000 to 2699

interfaces, Cisco provides access lists. An IP access list is a sequential collection of permit and deny conditions that apply to IP addresses or upper-layer IP protocols. IP access lists identify traffic, and can be used for many applications, including filtering packets coming into or going out of an interface, or restricting packets to and from virtual terminal lines.

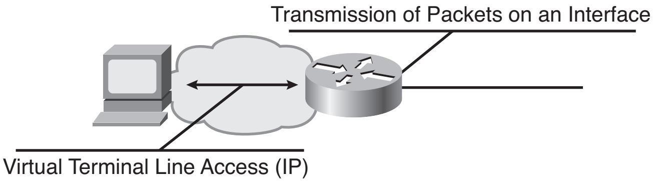

Transmission of Packets on an Interface
Virtual Terminal Line Access (IP)

Figure B-8 *Access Lists Control Packet Movement Through a Network.*

Table B-5 shows the available types of IP access lists on a Cisco router and their access list numbers. Named access lists are also available for IP.

This section covers IP standard and extended access lists. For information on other types of access lists, see the technical documentation on the Cisco website at http://www.cisco.com.

Warning Cisco IOS Release 10.3 introduced substantial additions to IP access lists. These extensions are backward compatible. Migrating from older releases to the Cisco IOS Release 10.3 or a later image will convert your access lists automatically. However, earlier releases are not upwardly compatible with these changes. Therefore, if you save an access list with the Cisco IOS Release 10.3 or a later image and then use older software, the resulting access list will not be interpreted correctly. This incompatibility can cause security problems. Save your old configuration file before booting Cisco IOS Release 10.3 (or later) images in case you need to revert to an earlier version.

IP Standard Access Lists

Standard access lists permit or deny packets based only on the packet's source IP address, as shown in Figure B-9. The access list number range for standard IP access lists is 1 to 99 or from 1300 to 1999. Standard access lists are easier to configure than their more robust counterparts, extended access lists, but do not provide the granularity available with extended access lists.

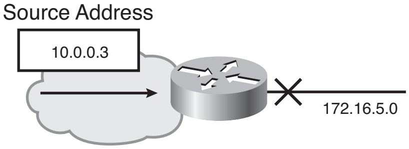


Figure B-9 *Standard IP Access Lists Filter Based Only on the Source Address.*

A standard access list is a sequential collection of permit and deny conditions that apply to source IP addresses. The router tests addresses against the conditions in an access list one by one. The first match determines whether the router permits or denies the packet. Because the router stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the router rejects the packet.

Figure B-10 shows the processing of inbound standard access lists. After receiving a packet, the router checks the packet's source address against the access list. If the access list permits the address, the router exits the access list and continues to process the packet. If the access list rejects the address, the router discards the packet and returns an Internet Control Message Protocol (ICMP) administratively prohibited message.

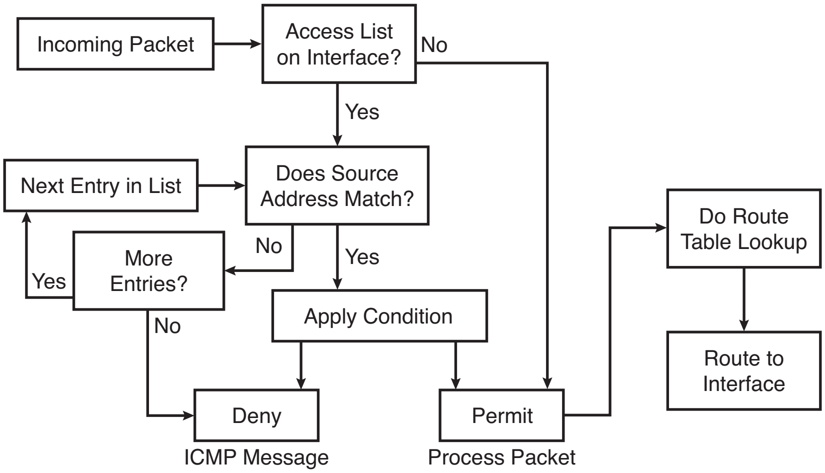


Figure B-10 *Inbound Standard IP Access List Processing.*

Note that the action taken if no more entries are found in the access list is to deny the packet. This illustrates an important rule to remember when creating access lists: The last entry in an access list is known as an *implicit* **deny any**; all traffic not explicitly permitted is implicitly denied. For example, consider what will happen if you create a list that just denies traffic that you do not want to let into your network, and you configure this on an interface. If you forget about this rule, *all* of your traffic is denied—the traffic explicitly denied by your list, and the rest of the traffic that is implicitly denied because the access list is applied to the interface.

Another important point to remember when configuring access lists is that order is important. Make sure that you list the entries in order, from specific to general. For example, if you want to deny a specific host address and permit all other addresses, make sure that your entry about the specific host appears first.

Figure B-11 illustrates the processing of outbound standard IP access lists. After receiving and routing a packet to a controlled interface, the router checks the packet's source address against the access list. If the access list permits the address, the router sends the packet. If the access list denies the address, the router discards the packet and returns an ICMP administratively prohibited message.

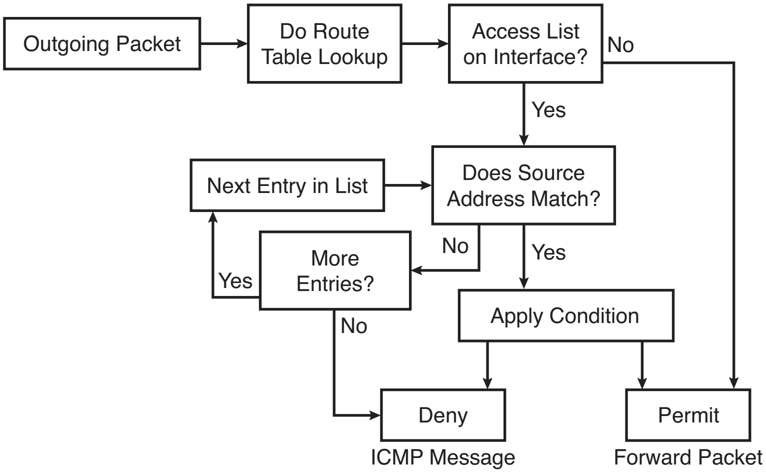


Figure B-11 *Outbound Standard IP Access List Processing.*

Wildcard Masks

Both standard and extended IP access lists use a wildcard mask. Like an IP address, a *wildcard mask* is a 32-bit quantity written in dotted-decimal format. The wildcard mask tells the router which bits of the address to use in comparisons:

- Address bits corresponding to wildcard mask bits set to 1 are ignored in comparisons.
- Address bits corresponding to wildcard mask bits set to 0 are used in comparisons.

An alternative way to think of the wildcard mask is as follows. If a 0 bit appears in the wildcard mask, the corresponding bit location in the access list address and the same bit location in the packet address must match (both must be 0 or both must be 1). If a 1 bit appears in the wildcard mask, the corresponding bit location in the packet matches (whether it is 0 or 1), and that bit location in the access list address is ignored. For this reason, bits set to 1 in the wildcard mask are sometimes called *don't care bits*.

Remember that the order of the access list statements is important because the access list is not processed further after a match is found.

Wildcard Masks The concept of a wildcard mask is similar to the wildcard character used in DOS-based computers. For example, to delete all files on your computer that begin with the letter *f*, you would enter this:

delete f*.*

The * character is the wildcard. Any files that start with f, followed by any other characters, and then a dot, and then any other characters, are deleted.

Instead of using wildcard characters, routers use wildcard masks to implement this concept.

Examples of addresses and wildcard masks, and what they match, are shown in Table B-6.

Access List Configuration Tasks

Whether you are creating a standard or extended access list, you need to complete the following two tasks:

Step 1. Create an access list in global configuration mode by specifying an access list number and access conditions.

Define a standard IP access list using a source address and wildcard, as shown later in this section.

Define an extended access list using source and destination addresses, and optional protocol-type information for finer granularity of control, as discussed in the "IP Extended Access Lists" section, later in this appendix.

Step 2. Apply the access list in interface configuration mode to interfaces (or in line configuration mode to terminal lines).

After creating an access list, you can apply it to one or more interfaces. Access lists can be applied either outbound or inbound on interfaces.

Table B-6 *Access List Wildcard Mask Examples*

Address	Wildcard Mask	What It Matches
0.0.0.0	255.255.255.255	Any address
172.16.0.0/16	0.0.255.255	Any host on network 172.16.0.0
172.16.7.11/16	0.0.0.0	Host address 172.16.7.11
255.255.255.255	0.0.0.0	Local broadcast address 255.255.255.255
172.16.8.0/21	0.0.7.255	Any host on subnet 172.16.8.0/21

Table B-7 *Standard IP* **access-list** *Command Description*

Parameter	Description
access-list-number	Identifies the list to which the entry belongs. A number from 1 to 99 or from 1300 to 1999.
permit | **deny**	Indicates whether this entry allows or blocks traffic from the specified address.
s*ource*	Identifies the source IP address.
source-wildcard	(Optional) Identifies which bits in the address field must match. A 1 in any bit position indicates don't care bits, and a 0 in any bit position indicates that the bit must strictly match. If this field is omitted, the wildcard mask 0.0.0.0 is assumed.
any	Use this keyword as an abbreviation for a source and source wildcard of 0.0.0.0 255.255.255.255.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. Exercise caution when using this keyword, because it consumes CPU cycles. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

IP Standard Access List Configuration

Use the **access-list** *access-list-number* {**permit** | **deny**} {*source* [*source-wildcard*] | **any**} [**log**] global configuration command to create an entry in a standard access list, as detailed in Table B-7.

When a packet does not match any of the configured lines in an access list, the packet is denied by default because there is an invisible line at the end of the access list that is equivalent to **deny any**. (**deny any** is the same as denying an address of 0.0.0.0 with a wildcard mask of 255.255.255.255.)

The keyword **host** can also be used in an access list. It causes the address that immediately follows it to be treated as if it were specified with a mask of 0.0.0.0. For example, configuring **host 10.1.1.1** in an access list is equivalent to configuring **10.1.1.1 0.0.0.0**.

Use the **ip access-group** *access-list-number* {**in** | **out**} interface configuration command to link an existing access list to an interface, as shown in Table B-8. Each interface can have both an inbound and an outbound IP access list.

Table B-8 **ip access-group** *Command Description*

Parameter	Description
access-list-number	Indicates the number of the access list to be linked to this interface.
in | **out**	Processes packets arriving on or leaving from this interface. The default is **out**.

Eliminate the entire list by entering the **no access-list** *access-list-number* global configuration command. Remove an access list from an interface with the **no ip access-group** *access-list-number* {**in** | **out**} interface configuration command.

Implicit Wildcard Masks

Implicit, or default, wildcard masks reduce typing and simplify configuration, but you must take care when relying on the default mask.

The access list line shown in Example B-2 is an example of a specific host configuration. For standard access lists, if no wildcard mask is specified, the wildcard mask is assumed to be 0.0.0.0. The implicit mask makes it easier to enter a large number of individual addresses.

Example B-2 *Standard Access List Using the Default Wildcard Mask*

```
access-list 1 permit 172.16.5.17
```

Example B-3 shows common errors found in access list lines.

Example B-3 *Common Errors Found in Access Lists*

```
access-list 1 permit 0.0.0.0
access-list 2 permit 172.16.0.0
access-list 3 deny any
access-list 3 deny 0.0.0.0 255.255.255.255
```

The first list in Example B-3—**permit 0.0.0.0**—would exactly match the address 0.0.0.0 and then permit it. Because you would never receive a packet from 0.0.0.0, this list would prevent all traffic from getting through (because of the implicit **deny any** at the end of the list).

The second list in Example B-3—**permit 172.16.0.0**—is probably a configuration error. The intention was probably 172.16.0.0 0.0.255.255. The exact address 172.16.0.0 refers to the network and would never be assigned to a host. As a result, nothing would get through with this list, again because of the implicit **deny any** at the end of the list. To filter networks or subnets, use an explicit wildcard mask.

The next two lines in Example B-3—**deny any** and **deny 0.0.0.0 255.255.255.255**—are unnecessary to configure because they duplicate the function of the implicit deny that occurs when a packet fails to match all the configured lines in an access list. Although they are not necessary, you might want to add one of these entries for record-keeping purposes.

Configuration Principles

The following general principles help ensure that the access lists you create have the intended results:

- Top-down processing
  - Organize your access list so that more specific references in a network or subnet appear before more general ones.
  - Place more frequently occurring conditions before less-frequent conditions.
- Implicit **deny any**
  - Unless you end your access list with an explicit **permit any**, it denies all traffic that fails to match any of the access list lines by default.
- New lines added to the end by default
  - Subsequent additions are always added to the end of the access list by default.
  - Cisco IOS Release 12.2(14)S introduced a feature called IP Access List Entry Sequence Numbering that enables network administrators to apply sequence numbers to **permit** or **deny** statements in a named IP access list and also reorder, add, or remove such statements. Before this feature, network administrators could only add access list entries to the end of an access list (which is the case for numbered access lists), meaning that if statements need to be added anywhere except the end of the access list, the entire access list must be reconfigured. You can selectively add or remove lines when using numbered access lists only by editing the numbered access list as though it is a named access list with the name equal to the number and using the sequence numbers that are automatically assigned to the lines.
- An undefined access list equals **permit any**
  - If you apply an access list with the **ip access-group** command to an interface before any access list lines have been created, the result is **permit any**. However, the list is live, so if you enter only one line, it goes from a **permit any** to a **deny** *most* (because of the implicit **deny any**) as soon as you press **Enter**. For this reason, you should create your access list before applying it to an interface.

Standard Access List Example

Figure B-12 shows a sample network, and Example B-4 shows the configuration on Router X in that figure.

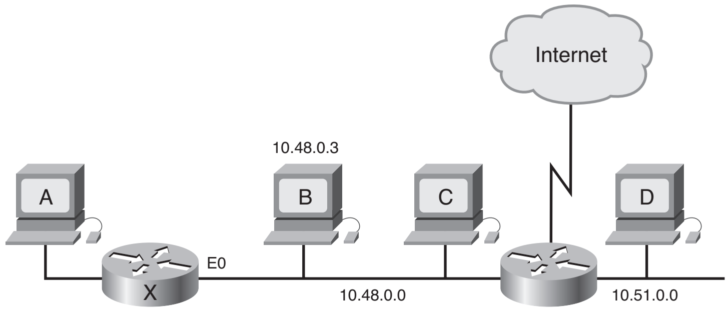


Figure B-12 *Network Used for the Standard IP Access List Example.*

Example B-4 *Standard Access List Configuration of Router X in Figure B-12*

```
Router(config)#access-list 2 permit 10.48.0.3
Router(config)#access-list 2 deny 10.48.0.0 0.0.255.255
Router(config)#access-list 2 permit 10.0.0.0 0.255.255.255
Router(config)#)!(Note: all other access implicitly denied)
Router(config)#interface ethernet 0
Router(config-if)#ip access-group 2 in
```

Consider which devices can communicate with Host A in this example:

- Host B can communicate with Host A. It is permitted by the first line of the access list, which uses an implicit host mask.
- Host C cannot communicate with Host A. Host C is in the subnet that is denied by the second line in the access list.
- Host D can communicate with Host A. Host D is on a subnet that is explicitly permitted by the third line of the access list.
- Users on the Internet cannot communicate with Host A. Users outside this network are not explicitly permitted, so they are denied by default with the implicit **deny any** at the end of the access list.

Location of Standard Access Lists

Access list location can be more of an art than a science. Consider the network in Figure B-13 and the access list configuration in Example B-5 to illustrate some general guidelines. If the policy goal is to deny Host Z access to Host V on another network, and not to change any other access policy, determine on which interface of which router this access list should be configured.

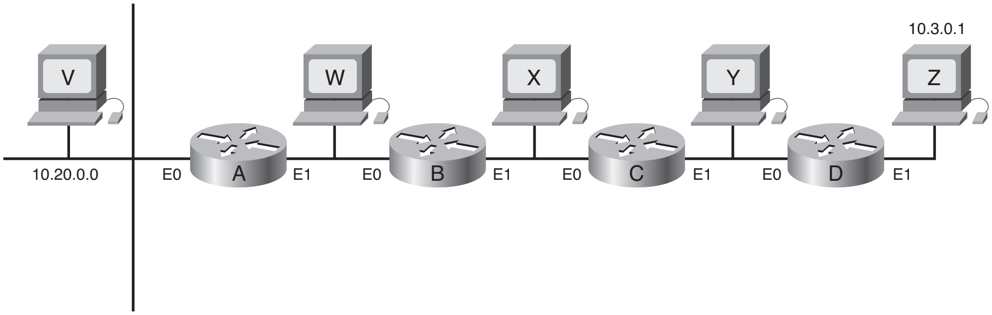


Figure B-13 *Location of the Standard IP Access List Example.*

Example B-5 *Standard Access List to Be Configured on a Router in Figure B-13*

```
access-list 3 deny 10.3.0.1
access-list 3 permit any
```

The access list should be placed on Router A because a standard access list can specify only a source address. No hosts beyond the point in the path where the traffic is denied can connect.

The access list could be configured as an outbound list on E0 of Router A. However, it would most likely be configured as an inbound list on E1 so that packets to be denied would not have to be routed through Router A first.

Consider the effect of placing the access list on other routers:

- **Router B—**Host Z could not connect with Host W (and Host V).
- **Router C—**Host Z could not connect with Hosts W and X (and Host V).
- **Router D—**Host Z could not connect with Hosts W, X, and Y (and Host V).

Therefore, for standard access lists, the rule is to place them as close to the *destination* as possible to exercise the most control. Note, however, that this means that traffic is routed through the network, only to be denied close to its destination.

IP Extended Access Lists

Standard access lists offer quick configuration and low overhead in limiting traffic based on source addresses in a network. *Extended access lists* provide a higher degree of control by enabling filtering based on the source and destination addresses, transport layer protocol, and application port number. These features make it possible to limit traffic based on the uses of the network.

Extended Access List Processing

As shown in Figure B-14, every condition tested in a line of an extended access list must match for the line of the access list to match and for the permit or deny condition to be applied. As soon as one parameter or condition fails, the next line in the access list is compared.

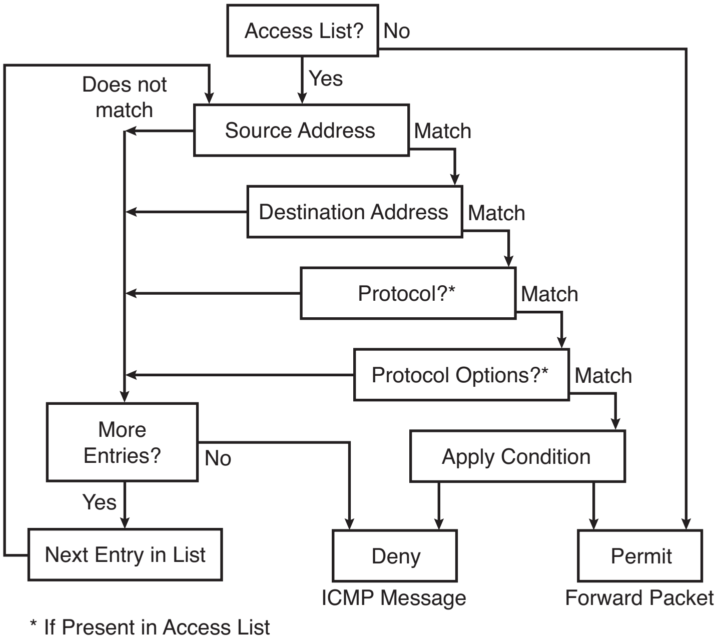


Figure B-14 *Extended IP Access List Processing Flow.*

The extended access list checks source address, destination address, and protocol. Depending on the configured protocol, more protocol-dependent options might be tested. For example, a TCP port might be checked, which allows routers to filter at the application layer.

Extended IP Access List Configuration

Use the **access-list** *access-list-number* {**permit** | **deny**} *protocol* {*source source-wildcard* | **any**} {*destination destination-wildcard* | **any**} [*protocol-specific-options*] [**log**] global configuration command to create an entry in an extended-traffic filter list. Table B-9 describes this command.

The wildcard masks in an extended access list operate the same way as they do in standard access lists, but note that they are not optional in extended access lists.. The keyword **any** in either the source or the destination position matches any address and is equivalent to configuring an address of 0.0.0.0 with a wildcard mask of 255.255.255.255. Example B-6 shows an example of an extended access list.

Table B-9 ip access-group *Command Description*

Parameter	Description
access-list-number	Identifies the list to which the entry belongs (a number from 100 to 199 or from 2000 to 2699).
permit | deny	Indicates whether this entry allows or blocks traffic.
protocol	**ip**, **tcp**, **udp**, **icmp**, **igmp**, **gre**, **eigrp**, **ospf**, **nos**, **ipinip**, **pim**, or a number from 0 to 255. To match any Internet protocol, use the keyword **ip**. As shown later in this section, some protocols allow more options that are supported by an alternative syntax for this command.
source and *destination*	Identifies the source and destination IP addresses.
source-wildcard and *destination-wildcard*	Identifies which bits in the address field must match. A 1 in any bit position indicates don't care bits, and a 0 in any bit position indicates that the bit must strictly match.
any	Use this keyword as an abbreviation for a source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
log	(Optional) Causes informational logging messages about a packet that matches the entry to be sent to the console. Exercise caution when using this keyword, because it consumes CPU cycles. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Example B-6 *Use of the Keyword* any

```
access-list 101 permit ip  0.0.0.0  255.255.255.255  0.0.0.0  255.255.255.255
! (alternative configuration)
access-list 101 permit ip any any
```

The keyword **host** can be used in either the source or the destination position. It causes the address that immediately follows it to be treated as if it were specified with a mask of 0.0.0.0. Example B-7 shows an example.

Example B-7 *Use of the Keyword* host

```
access-list 101 permit ip  0.0.0.0  255.255.255.255  172.16.5.17  0.0.0.0
! (alternative configuration)
access-list 101 permit ip any host 172.16.5.17
```

Use the **access-list** *access-list-number* {**permit** | **deny**} **icmp** {*source source-wildcard* | **any**} {*destination destination-wildcard* | *any*} [*icmp-type* [*icmp-code*] | *icmp-message*] global configuration command to filter ICMP traffic. The protocol keyword **icmp** indicates that an alternative syntax is being used for this command and that protocol-specific options are available, as described in Table B-10.

Cisco IOS Release 10.3 and later versions provide symbolic names that make configuring and reading complex access lists easier. With symbolic names, it is no longer critical to understand the meaning of the ICMP message type and code (for example, message 8 and message 0 can be used to filter the **ping** command). Instead, the configuration can use symbolic names, as shown in Table B-11. For example, the **echo** and **echo-reply** symbolic names can be used to filter the **ping** command. (You can use the Cisco IOS context-sensitive help feature by entering **?** when entering the **access-list** command to verify the available names and proper command syntax.)

Use the **access-list** *access-list-number* {**permit** | **deny**} **tcp** {*source source-wildcard* | **any**} [*operator source-port* | *source-port*] {*destination destination-wildcard* | **any**} [*operator*

Table B-10 *Extended IP* **access-list icmp** *Command Description*

Parameter	**Description**
access-list-number	Identifies the list to which the entry belongs (a number from 100 to 199 or from 2000 to 2699).
permit | **deny**	Indicates whether this entry allows or blocks traffic.
source and *destination*	Identifies the source and destination IP addresses.
source-wildcard and *destination-wildcard*	Identifies which bits in the address field must match. A 1 in any bit position indicates don't care bits, and a 0 in any bit position indicates that the bit must strictly match.
any	Use this keyword as an abbreviation for a source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
icmp-type	(Optional) Packets can be filtered by ICMP message type. The type is a number from 0 to 255.
icmp-code	(Optional) Packets that have been filtered by ICMP message type can also be filtered by ICMP message code. The code is a number from 0 to 255.
icmp-message	(Optional) Packets can be filtered by a symbolic name representing an ICMP message type or a combination of ICMP message type and ICMP message code. These names are listed in Table B-11.

Table B-11 *ICMP Message and Type Names*

Administratively-prohibited	Dod-host-prohibited
Alternate-address	Dod-net-prohibited
Conversion-error	Echo
Host-precedence-unreachable	Net-tos-redirect
Host-redirect	Net-tos-unreachable
Host-tos-redirect	Net-unreachable
Host-tos-unreachable	Network-unknown
Host-unknown	No-room-for-option
Host-unreachable	Option-missing
Information-reply	Packet-too-big
Information-request	Parameter-problem
Mask-reply	Port-unreachable
Mask-request	Precedence-unreachable
Mobile-redirect	Protocol-unreachable
Net-redirect	Reassembly-timeout

destination-port | *destination-port*] [**established**] global configuration command to filter TCP traffic. The protocol keyword **tcp** indicates that an alternative syntax is being used for this command and that protocol-specific options are available, as described in Table B-12.

established Keyword in Extended Access Lists When a TCP session is started between two devices, the first segment that is sent has the synchronize (SYN) code bit set but does not have the acknowledge (ACK) code bit set in the segment header, because it is not acknowledging any other segments. All subsequent segments sent do have the ACK code bit set, because they are acknowledging previous segments sent by the other device. This is how a router can distinguish between a segment from a device that is attempting to *start* a TCP session and a segment of an ongoing *established* session. The RST code bit is set when an established session is being terminated.

When you configure the **established** keyword in a TCP extended access list, it indicates that that access list statement should match only TCP segments in which the ACK or RST code bit is set. In other words, only segments that are part of an established session are matched. Segments that are attempting to start a session do not match the access list statement.

Table B-12 *Extended IP* **access-list tcp** *Command Description*

Parameter	Description
access-list-number	Identifies the list to which the entry belongs (a number from 100 to 199 or from 2000 to 2699).
permit | **deny**	Indicates whether this entry allows or blocks traffic.
source and *destination*	Identifies the source and destination IP addresses.
source-wildcard and *destination-wildcard*	Identifies which bits in the address field must match. A 1 in any bit position indicates don't care bits, and a 0 in any bit position indicates that the bit must strictly match.
any	Use this keyword as an abbreviation for a source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
operator	(Optional) A qualifying condition. Can be **lt**, **gt**, **eq**, or **neq**.
source-port and *destination-port*	(Optional) A decimal number from 0 to 65535 or a name that represents a TCP port number.
established	(Optional) A match occurs if the TCP segment has the ACK or RST bits set. Use this if you want a Telnet or other activity to be established in one direction only.

Table B-13 lists TCP port names that can be used instead of port numbers. You can find the port numbers corresponding to these protocols by entering a **?** in place of a port number or by looking at the port numbers on http://www.iana.org/assignments/port-numbers.

Table B-13 *TCP Port Names*

Bgp	Hostname
Chargen	Irc
Daytime	Klogin
Discard	Kshell
Domain	Lpd
Echo	Nntp
Finger	Pop2
Ftp control	Pop3
Ftp-data	Smtp
Gopher	Sunrpc

Other port numbers can be found at http://www.iana.org/assignments/port-numbers. A partial list of the assigned TCP port numbers is shown in Table B-14.

Use the **access-list** *access-list-number* {**permit** | **deny**} **udp** {*source source-wildcard* | **any**} [*operator source-port* | *source-port*] {*destination destination-wildcard* | **any**} [*operator destination-port* | *destination-port*] global configuration command to filter User Datagram Protocol (UDP) traffic. The protocol keyword **udp** indicates that an alternative syntax is being used for this command and that protocol-specific options are available, as described in Table B-15.

Table B-14 *Some Reserved TCP Port Numbers*

Port Number (Decimal)	Keyword	Description
7	ECHO	Echo
9	DISCARD	Discard
13	DAYTIME	Daytime
19	CHARGEN	Character generator
20	FTP-DATA	File Transfer Protocol (data)
21	FTP-CONTROL	File Transfer Protocol
23	TELNET	Terminal connection
25	SMTP	Simple Mail Transfer Protocol
37	TIME	Time of day
43	WHOIS	Who is
53	DOMAIN	Domain name server
79	FINGER	Finger
80	WWW	World Wide Web HTTP
101	HOSTNAME	NIC hostname server

Table B-15 *Extended IP* **access-list udp** *Command Description*

Parameter	Description
access-list-number	Identifies the list to which the entry belongs (a number from 100 to 199 or from 2000 to 2699).
permit | **deny**	Indicates whether this entry allows or blocks traffic.
source and *destination*	Identifies the source and destination IP addresses.

Table B-15 *Extended IP* **access-list udp** *Command Description (continued)*

source-wildcard and *destination-wildcard*	Identifies which bits in the address field must match. A 1 in any bit position indicates don't care bits, and a 0 in any bit position indicates that the bit must strictly match.
any	Use this keyword as an abbreviation for a source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.
operator	(Optional) A qualifying condition. Can be lt, gt, eq, or neq.
source-port and *destination-port*	(Optional) A decimal number from 0 to 65535 or a name that represents a UDP port number.

Table B-16 lists UDP port names that can be used instead of port numbers. You can find port numbers corresponding to these protocols by entering a **?** in place of a port number or by looking at http://www.iana.org/assignments/port-numbers.

Other port numbers can be found at http://www.iana.org/assignments/port-numbers. A partial list of the assigned UDP port numbers is shown in Table B-17.

Table B-16 *UDP Port Names*

Biff	Nameserver	Syslog
Bootpc	NetBios-dgm	Tacacs-ds
Bootps	NetBios-ns	Talk
Discard	Ntp	Tftp
Dns	Rip	Time
Dnsix	Snmp	Whois
Echo	Snmptrap	Xdmcp
Mobile-ip	Sunrpc	

Table B-17 *Some Reserved UDP Port Numbers*

Port Number (Decimal)	Keyword	Description
7	ECHO	Echo
9	DISCARD	Discard
37	TIME	Time of day
42	NAMESERVER	Host name server
43	WHOIS	Who is
53	DNS	Domain name server
67	BOOTPS	Bootstrap protocol server

continues

Table B-17 *Some Reserved UDP Port Numbers (continued)*

Port Number (Decimal)	Keyword	Description
68	BOOTPC	Bootstrap protocol client
69	TFTP	Trivial File Transfer Protocol
123	NTP	Network Time Protocol
137	NetBios-ns	NetBIOS name service
138	NetBios-dgm	NetBIOS datagram service
161	SNMP	SNMP
162	SNMPTrap	SNMP traps
520	RIP	RIP

Extended Access List Examples

In Figure B-15, Router A's interface Ethernet 1 is part of a Class B subnet with the address 172.22.3.0, Router A's interface Serial 0 is connected to the Internet, and the e-mail server's address is 172.22.1.2. The access list configuration applied to Router A is shown in Example B-8.

Example B-8 *Configuration on Router A in Figure B-15*

```
access-list 104 permit tcp any 172.22.0.0 0.0.255.255 established
access-list 104 permit tcp any host 172.22.1.2 eq smtp
access-list 104 permit udp any any eq dns
access-list 104 permit icmp any any echo
access-list 104 permit icmp any any echo-reply
!
interface serial 0
ip access-group 104 in
```

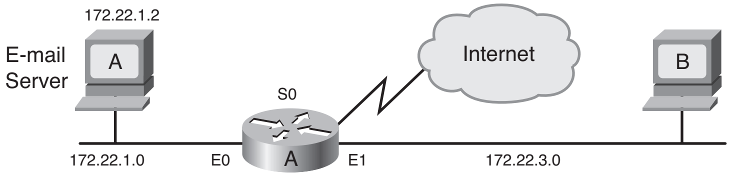


Figure B-15 *Network Used for the Extended IP Access List Example.*

In Example B-8, access list 104 is applied inbound on Router A's Serial 0 interface. The keyword **established** is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP segment has the ACK or RST bits set, which indicate that the packet belongs to an existing connection. If the session is not already established (the ACK bit is not set and the SYN bit is set), this means that someone on the Internet is attempting to initialize a session, in which case the packet is denied. This configuration also permits Simple Mail Transfer Protocol (SMTP) traffic from any address to the e-mail server. UDP domain name server packets and ICMP echo and echo-reply packets are also permitted from any address to any other address.

Another example is shown in Figure B-16. Example B-9 shows the access list configuration applied to Router A.

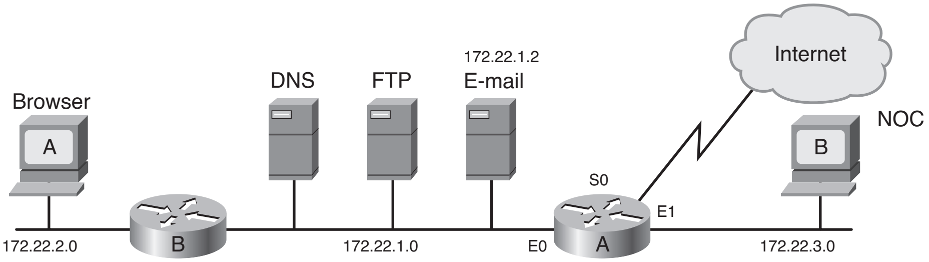


Figure B-16 *Extended IP Access List Example with Many Servers.*

Example B-9 *Configuration on Router A in Figure B-16*

```
access-list 118 permit tcp any 172.22.0.0  0.0.255.255 eq www established
access-list 118 permit tcp any host 172.22.1.2 eq smtp
access-list 118 permit udp any any eq dns
access-list 118 permit udp 172.22.3.0  0.0.0.255 172.22.1.0 0.0.0.255 eq snmp

access-list 118 deny icmp any 172.22.0.0  0.0.255.255 echo
access-list 118 permit icmp any any echo-reply
!
interface ethernet 0
ip access-group 118 out
```

In Example B-9, access list 118 is applied outbound on Router A's Ethernet 0 interface. With the configuration shown in Example B-9, *replies* to queries from the Client A browser (or any other host on the corporate network) to the Internet are allowed back into the corporate network (because they are established sessions). Browser queries *from* external sources are not explicitly allowed and are discarded by the implicit **deny any** at the end of the access list.

The access list in Example B-9 also allows e-mail (SMTP) to be delivered exclusively to the mail server. The name server is permitted to resolve DNS requests. The 172.22.1.0 subnet is controlled by the network management group located at the NOC server (Client B), so network-management queries (Simple Network Management Protocol [SNMP]) will be allowed to reach these devices in the server farm. Attempts to ping the corporate network from the outside or from subnet 172.22.3.0 will fail because the access list blocks the echo requests. However, replies to echo requests generated from within the corporate network are allowed to reenter the network.

Location of Extended Access Lists

Because extended access lists can filter on more than a source address, location is no longer the constraint it was when considering the location of a standard access list. Policy decisions and goals are frequently the driving forces behind extended access list placement.

If your goal is to minimize traffic congestion and maximize performance, you might want to push the access lists close to the source to minimize cross-network traffic and administratively prohibited ICMP messages. If your goal is to maintain tight control over access lists as part of your network security strategy, you might want them to be more centrally located. Notice how changing network goals affects access list configuration.

Here are some things to consider when placing extended access lists:

- Minimize distance traveled by traffic that will be denied (and ICMP unreachable messages).
- Keep denied traffic off the backbone.
- Select the router that will have the CPU overhead from processing the access lists.
- Consider the number of interfaces affected.
- Consider access list management and security.
- Consider network growth impacts on access list maintenance.

Restricting Virtual Terminal Access

This section discusses how you can use standard access lists to limit virtual terminal access. Standard and extended access lists block packets from going *through* the router. They are not designed to block packets that originate within the router. For example, an outbound Telnet extended access list does not prevent router-initiated Telnet sessions by default.

For security purposes, users can be denied virtual terminal (vty) access to the router, or they can be permitted vty access to the router but denied access to destinations from that router. Restricting vty access is less of a traffic-control mechanism than one technique for increasing network security.

Vty access is accomplished using the Telnet or Secure Shell (SSH) protocol. There is only one type of vty access list.

How to Control vty Access

Just as a router has physical ports or interfaces such as Ethernet 0 and Ethernet 1, it also has virtual ports. These virtual ports are called virtual terminal lines. By default, there are five such virtual terminal lines on a router, numbered vty 0 to 4, as shown in Figure B-17.

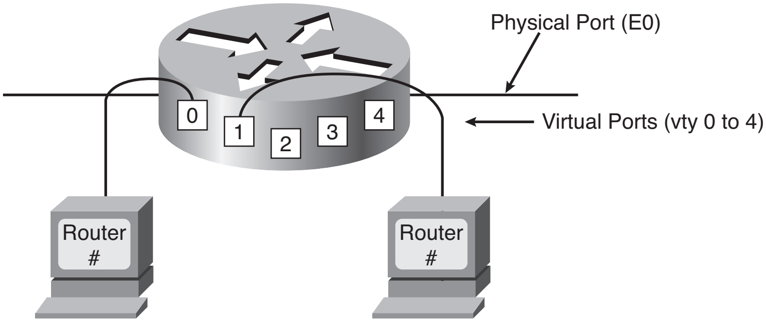


Figure B-17 *A Router Has Five Virtual Terminal Lines (Virtual Ports) by Default.*

You should set identical restrictions on all virtual terminal lines, because you cannot control on which virtual terminal line a user will connect.

Virtual Terminal Line Access Configuration

Use the **line vty** {*vty-number* | *vty-range*} global configuration command to place the router in line configuration mode, as described in Table B-18.

Use the **access-class** *access-list-number* {**in** | **out**} line configuration command to link an existing access list to a terminal line or range of lines, as described in Table B-19.

Table B-18 **line vty** *Command Description*

Parameter	Description
vty-number	Indicates the number of the vty line to be configured
vty-range	Indicates the range of vty lines to which the configuration applies

Table B-19 **access-class** *Command Description*

Parameter	Description
access-list-number	Indicates the number of the standard access list to be linked to a terminal line. This is a decimal number from 1 to 99 or from 1300 to 1999.
in	Prevents the router from receiving incoming connections from the addresses defined in the access list.
out	Prevents someone from initiating a Telnet to the addresses defined in the access list.

Note When you use the **out** keyword in the **access-class** command, the addresses in the specified standard access list are actually treated as *destination* addresses, rather than as source addresses.

In Example B-10, any device on network 192.168.55.0 is permitted to establish a virtual terminal session (for example, a Telnet session) with the router. Of course, the user must know the appropriate passwords for entering user mode and privileged mode.

Example B-10 *Configuration to Restrict Telnet Access to a Router*

```
access-list 12 permit 192.168.55.0 0.0.0.255
!
line vty 0 4
access-class 12 in
```

Notice that in this example, identical restrictions have been set on all virtual terminal lines (0 to 4), because you cannot control on which virtual terminal line a user will connect. Note that the implicit **deny any** still applies to this alternative application of access lists.

Verifying Access List Configuration

Use the **show access-lists** [*access-list-number* | *name*] privileged EXEC command to display access lists from all protocols, as described in Table B-20. If no parameters are specified, all access lists are displayed.

Table B-20 **show access-lists** *Command Description*

Parameter	Description
access-list-number	(Optional) Number of the access list to display
name	(Optional) Name of the access list to display

The system counts how many packets match each line of an access list. The counters are displayed by the **show access-lists** command.

Example B-11 illustrates sample output from the **show access-lists** command. In this example, the first line of the access list has been matched three times, and the last line has been matched 629 times. The second line has not been matched.

Example B-11 *Output of the* **show access-lists** *Command*

```
 p1r1#show access-lists
Extended IP access list 100
    deny tcp host 10.1.1.2 host 10.1.1.1 eq telnet (3 matches)
    deny tcp host 10.1.2.2 host 10.1.2.1 eq telnet
    permit ip any any (629 matches)
```

Use the **show ip access-list** [*access-list-number* | *name*] EXEC command to display IP access lists, as described in Table B-21. If no parameters are specified, all IP access lists are displayed.

Use the **clear access-list counters** [*access-list-number* | *name*] EXEC command to clear the counters for the number of matches in an extended access list, as described in Table B-22. If no parameters are specified, the counters are cleared for all access lists.

Use the **show line** [*line-number*] EXEC command to display information about terminal lines. The *line-number* is optional and indicates the absolute line number of the line for which you want to list parameters. If a line number is not specified, all lines are displayed.

IPv4 Address Planning

A well-designed large-scale internetwork with an effective IP addressing plan has many benefits, as described in this section.

Benefits of an Optimized IP Addressing Plan

An optimized IP addressing plan uses hierarchical addressing.

Perhaps the best-known addressing hierarchy is the telephone network. The telephone network uses a hierarchical numbering scheme that includes country codes, area codes, and local exchange numbers. For example, if you are in San Jose, California, and you call someone else in San Jose, you dial the San Jose prefix, 528, and the person's four-digit line number. Upon seeing the number 528, the central office recognizes that the destination telephone is within its area, so it looks up the four-digit number and transfers the call.

Table B-21 **show ip access-list** *Command Description*

Parameter	**Description**
access-list-number	(Optional) Number of the IP access list to display
name	(Optional) Name of the IP access list to display

Table B-22 clear access-list counters *Command Description*

Parameter	**Description**
access-list-number	(Optional) Number of the access list for which to clear the counters
name	(Optional) Name of the access list for which to clear the counters

Note In many places in North America now, the area code must also be dialed for local calls. This is because of changes in the use of specific digits for area codes and local exchange numbers. The telephone network is suffering from *address exhaustion*, just like the IP network. Changes in how telephone numbers are used is one solution being implemented to solve this problem.

In another example (see Figure B-18), to call Aunt Judy in Alexandria, Virginia, from San Jose, you dial 1, and then the area code 703, and then the Alexandria prefix 555, and then Aunt Judy's local line number, 1212. The central office first sees the number 1, indicating a remote call, and then looks up the number 703. The central office immediately routes the call to a central office in Alexandria. The San Jose central office does not know exactly where 555-1212 is in Alexandria, nor does it have to. It needs to know only the area codes, which summarize the local telephone numbers within an area.

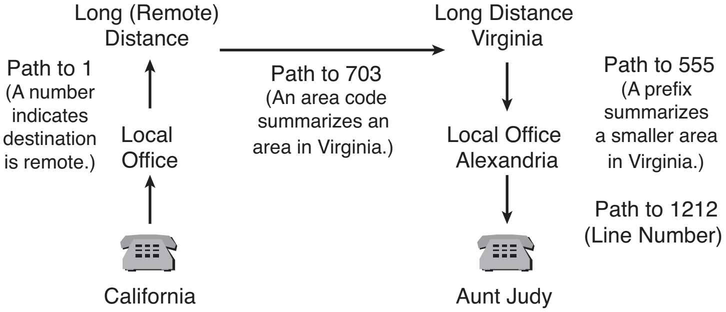


Figure B-18 *The Telephone Network Uses an Addressing Hierarchy.*

Note As you might have noticed, the telephone number used in this example is the number for international directory assistance. It is used for illustration purposes to ensure that Aunt Judy's personal number is not published.

If there were no hierarchical structure, every central office would need to have every telephone number worldwide in its locator table. Instead, the central offices have summary numbers, such as area codes and country codes. A summary number (address) represents a group of numbers. For example, an area code such as 408 is a summary number for the San Jose area. In other words, if you dial 1-408 from anywhere in the United States or Canada, followed by a seven-digit telephone number, the central office routes the call to a San Jose central office. Similarly, a routed network can employ a hierarchical addressing scheme to take advantage of those same benefits.

One of the benefits of hierarchical addressing is a reduced number of routing table entries. Whether it is with your Internet routers or your internal routers, you should try to keep your routing tables as small as possible by using route summarization.

Summarization (also called *aggregation*, *supernetting*, or *information hiding*) is not a new concept. When a router announces a route to a given network, the route is a summarization of all the host and device individual addresses that reside on that network. Route summarization is a way of having a single IP address represent a collection of IP addresses. This is most easily accomplished when you employ a hierarchical addressing plan. By summarizing routes, you can keep your routing table entries (on the routers that receive the summarized routes) manageable, which offers the following benefits:

- More efficient routing.
- A reduced number of CPU cycles when recalculating a routing table or sorting through the routing table entries to find a match.
- Reduced router memory requirements.
- Reduced bandwidth required to send the fewer, smaller routing updates.
- Faster convergence after a change in the network.
- Easier troubleshooting.
- Increased network stability. Because summarization limits the propagation of detailed routes, it also reduces the impact to the network when these detailed routes fail.

Another benefit of hierarchical addressing is the efficient allocation of addresses. Hierarchical addressing lets you take advantage of all possible addresses because you group them contiguously. With random address assignment, you might end up wasting groups of addresses because of addressing conflicts. For example, classful routing protocols (discussed in the later section “Implementing VLSM in a Scalable Network”) automatically create summary routes at a network boundary. Therefore, these protocols do not support discontiguous addressing (as you can see in Chapter 1, “Routing Services”), so some addresses would be unusable if not assigned contiguously.

Scalable Network Addressing Example

The network illustrated in Figure B-19 shows an example of scalable addressing. In this example, a U.S. national drugstore chain plans to have a retail outlet in every city in the country with a population greater than 10,000. Each of the 50 states has up to 100 stores, with 2 Ethernet LANs in each store, as follows:

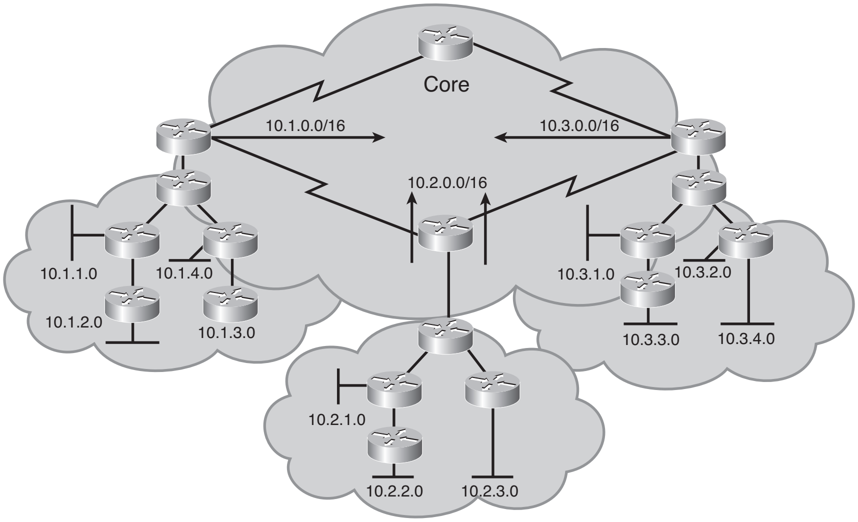


Figure B-19 *Scalable Addressing Allows Summarization.*

- One LAN is used to track customer prescriptions, pharmacy inventory, and reorder stock.
- The second LAN is used to stock the rest of the store and connect the cash registers to a corporatewide, instantaneous point-of-sale evaluation tool.

The total number of Ethernet LAN networks is 50 states * 100 stores per state * 2 LANs per store = 10,000. (An equal number of serial links interconnects these stores.)

Using a scalable design and creating 51 divisions (one for each state and one for the backbone interconnecting the divisions), the corporation can assign each division a block of IP addresses 10.*x*.0.0 /16. Each LAN is assigned a /24 subnet of network 10.0.0.0, and each division has 200 such subnets (two for each of the 100 stores). The network will have 10,000 subnets; without summarization, each of the 5000 routers will have all these networks in their routing tables. If each division router summarizes its block of networks 10.*x*.0.0 /16 at the entry point to the core network, any router in a division has only the 200 /24 subnets within that division, plus the 49 10.*x*.0.0 /16 summarizations that represent the other divisions, in its routing table. This results in a total of 249 networks in each IP routing table.

Nonscalable Network Addressing

In contrast to the previous example, if a hierarchical addressing plan is not used, summarization is not possible, as is the case in Figure B-20. Problems can occur in this network related to the frequency and size of routing table updates and how topology changes are processed in summarized and unsummarized networks. These problems are described next.

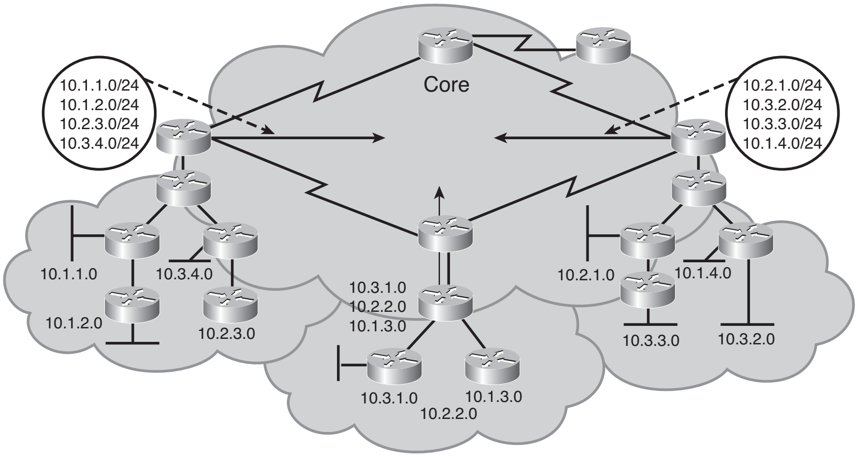


Figure B-20 *Nonscalable Addressing Results in Large Routing Tables.*

Update Size

Routing protocols such as the Routing Information Protocol (RIP), which sends a periodic update every 30 seconds, use valuable bandwidth to maintain a table without summarization. A single RIP update packet is limited to carrying 25 routes. Therefore, 10,000 routes means that RIP on every router must create and send 400 packets every 30 seconds. With summarized routes, the 249 routes means that only 10 packets need to be sent every 30 seconds.

Unsummarized Internetwork Topology Changes

A routing table with 10,000 entries constantly changes. To illustrate this constant change, consider the sample network with a router at each of 5000 different sites. A power outage occurs at site A, a backhoe digs a trench at site B, a newly hired system administrator begins work at site C, a Cisco IOS software upgrade is in progress at site D, and a newly added router is being installed at site E.

Every time a route changes, all the routing tables must be updated. For example, when using a routing protocol such as Open Shortest Path First (OSPF), an upgrade or topology change on the internetwork causes a shortest path first (SPF) calculation. The SPF calculations are large because each router needs to calculate all known pathways to each of the 10,000 networks. Each change a router receives requires time and CPU resources to process.

Summarized Network Topology Changes

In contrast to an unsummarized network, a summarized network responds efficiently to network changes. For example, in the sample drugstore network with 200 routes for each division, the routers within the division see all the subnets for that division. When a change occurs on one of the 200 routes in the division, all other routers in the division recalculate to reflect the topology change of those affected networks. However, the core router of that division passes a summarized /16 route and suppresses the /24 networks from advertisement to the core routers of other divisions. The summarized route is announced as long as any portion of the summarized block can be reached from that core router. The more specific routes are suppressed so that changes from this division are not propagated to other divisions.

In this scenario, each router has only 200 /24 networks, compared to the 10,000 /24 networks in an unsummarized environment. Obviously, the amount of CPU resources, memory, and bandwidth required for the 200 networks is less than the 10,000 networks. With summarization, each division hides more-specific information from the other divisions and passes only the summarized route that represents that overall division.

Hierarchical Addressing Using Variable-Length Subnet Masks

VLSM is a crucial component of an effective IP addressing plan for a scalable network. This section introduces VLSM, provides examples, and discusses methods of determining the best subnet mask for a given address requirement.

Network Mask

This section discusses the purpose of the network mask and its use within a network.

Use of the Network Mask

If a PC has an IP address of 192.168.1.67 with a mask of 255.255.255.240 (or a prefix length of /28), it uses this mask to determine the valid host addresses for devices on its local connection. These devices have the first 28 bits in their IP address in common (the range of these local devices is 192.168.1.65 through 192.168.1.78). If communication with any of these devices is necessary, the PC uses Address Resolution Protocol (ARP) to find the device's corresponding Media Access Control (MAC) address (assuming that it does not already have a destination MAC address for the IP address in its ARP table). If a PC needs to send information to an IP device that is not in the local range, the PC instead forwards the information to its default gateway. (The PC also uses ARP to discover the MAC address of the default gateway.)

A router behaves in a similar manner when it makes a routing decision. A packet arrives on the router and is passed to the routing table. The router compares the packet's destination IP address to the entries in the routing table. These entries have a prefix length associated with them.

The router uses the prefix length as the minimum number of destination address bits that must match to use the corresponding outbound interface that is associated with a network entry in the routing table.

Network Mask Example

Consider a scenario in which an IP packet with a destination address of 192.168.1.67 is sent to a router. Example B-12 shows the router's IP routing table.

Example B-12 *IP Routing Table for Network Mask Example*

```
192.168.1.0 is subnetted, 4 subnets
O 192.168.1.16/28 [110/1800] via 172.16.1.1, 00:05:17, Serial 0
C 192.168.1.32/28 is directly connected, Ethernet 0
O 192.168.1.64/28 [110/10] via 192.168.1.33, 00:05:17, Ethernet 0
O 192.168.1.80/28 [110/1800] via 172.16.2.1, 00:05:17, Serial 1
```

In this scenario, the router determines where to send a packet that is destined for 192.168.1.67 by looking at the routing table. The routing table has four entries for network 192.168.1.0. The router compares the destination address to each of the four entries for this network.

The destination address of 192.168.1.67 has the first three octets in common with all four entries in the routing table, but it is not clear by looking at the decimal representation which of those entries is the best match to route this packet. A router handles all packets in binary, not dotted-decimal, notation.

Following is the binary representation of the last octet for destination address 192.168.1.67 and the binary representation of the last octet for the four entries in the IP routing table. Because the prefix length is 28 and all four entries match at least the first 24 bits of 192.168.1, the router must find the routing table entry that matches the first 4 bits (bits 25 to 28) of the number 67. It is not important if the last 4 bits match (because they are host bits), so the target is 0100*xxxx*. The routing entry 64, which has a value of 0100 in the first 4 bits, is the only one that matches the requirement:

- 67: 01000011
- 16: 00010000
- 32: 00100000
- 64: 01000000
- 80: 01010000

The router therefore uses the 192.168.1.64 entry in the routing table and forwards this packet out of its Ethernet 0 interface to the next router (192.168.1.33).

Implementing VLSM in a Scalable Network

A major network (also known as a classful network) is a Class A, B, or C network.

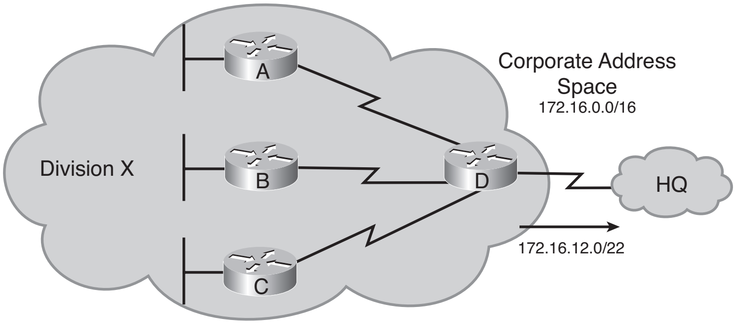
With classful routing, routing updates do not carry the subnet mask. Therefore, only one subnet mask can be used within a major network. This is known as fixed-length subnet masking (FLSM). An example of a classful routing protocol is RIP Version 1 (RIPv1).

With classless routing, routing updates do carry the subnet mask. Therefore, different masks may be used for different subnets within a major network. This is known as VLSM. Examples of classless routing protocols are RIP Version 2 (RIPv2), OSPF, Intermediate System-to-Intermediate System (IS-IS), and Enhanced Interior Gateway Routing Protocol (EIGRP).

Note Classful and classless routing protocols are discussed further in Chapter 1.

VLSM allows more than one subnet mask within a major network and enables the subnetting of a previously subnetted network address.

The network shown in Figure B-21 is used to illustrate how VLSM works.



172.16.12.0/22 has been assigned to Division X.
Range of Addresses: 172.16.12.0 to 171.16.15.255

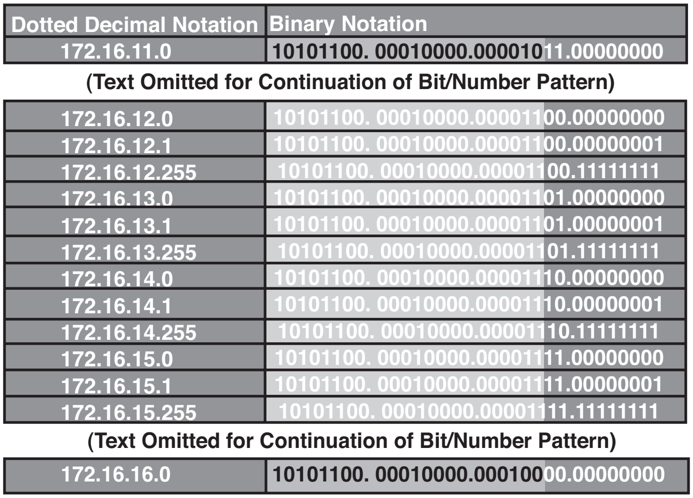
Figure B-21 *Network for the VLSM Example.*

The following are some characteristics that permit VLSMs to conserve IP addresses:

- **Efficient use of IP addresses—**Without the use of VLSMs, companies are locked into implementing a single subnet mask within an entire Class A, B, or C network number.
  - For example, suppose a network architect decides to use the 172.16.0.0/16 address space to design a corporate network. The architect determines that 64

blocks of addresses with up to 1022 hosts in each are required. Therefore, 10 host bits ($2^{10} - 2 = 1022$) and 6 subnet bits ($2^6 = 64$) are required for each block. The mask is therefore 255.255.252.0. The prefix is /22.

- The network architect assigns address block 172.16.12.0/22 to Division X, as shown in Figure B-21. The prefix mask of /22 indicates that all addresses within that range have the first 22 bits in common (when reading from left to right). The prefix mask provides Division X with a range of addresses from 172.16.12.0 through 172.16.15.255. The details of the range of addresses available to Division X are shown in the center block of Figure B-22. Within Division X, the networks are assigned addresses in this range, with varying subnet masks. Details of these address assignments are provided in the next section.



Dotted Decimal Notation	Binary Notation
172.16.11.0	10101100. 00010000.00001011.00000000

(Text Omitted for Continuation of Bit/Number Pattern)

172.16.12.0	10101100. 00010000.00001100.00000000
172.16.12.1	10101100. 00010000.00001100.00000001
172.16.12.255	10101100. 00010000.00001100.11111111
172.16.13.0	10101100. 00010000.00001101.00000000
172.16.13.1	10101100. 00010000.00001101.00000001
172.16.13.255	10101100. 00010000.00001101.11111111
172.16.14.0	10101100. 00010000.00001110.00000000
172.16.14.1	10101100. 00010000.00001110.00000001
172.16.14.255	10101100. 00010000.00001110.11111111
172.16.15.0	10101100. 00010000.00001111.00000000
172.16.15.1	10101100. 00010000.00001111.00000001
172.16.15.255	10101100. 00010000.00001111.11111111

(Text Omitted for Continuation of Bit/Number Pattern)

172.16.16.0	10101100. 00010000.00010000.00000000

Figure B-22 *Center Block Is Range of Addresses for VLSM for Division X in Figure B-21.*

- **Greater capability to use route summarization—**VLSMs allow for more hierarchical levels within an addressing plan and thus allow better route summarization within routing tables. For example, in Figure B-21, address 172.16.12.0/22 summarizes all the subnets that are further subnets of 172.16.12.0/22.
- **Reduced number of routing table entries—**In a hierarchical addressing plan, route summarization allows a single IP address to represent a collection of IP addresses. When VLSM is used in a hierarchical network, it allows summarized routes, which keeps routing table entries (on the routers that receive the summarized routes) manageable and provides the benefits described earlier in the "IPv4 Address Planning" section.
- Because of the reduced router requirements, it also might be possible to use some less-powerful (and therefore less-expensive) routers in the network.

The address 172.16.12.0/22 represents all the addresses that have the same first 22 bits as 172.16.12.0. Figure B-22 displays the binary representation of networks 172.16.11.0 through 172.16.16.0. Notice that 172.16.12.0 through 172.12.15.255 all have the first 22 bits in common, whereas 172.16.11.0 and 172.16.16.0 do not have the same first 22 bits. Therefore, the address 172.16.12.0/22 represents the range of addresses 172.16.12.0 through 172.16.15.255.

VLSM Calculation Example

You can best understand the design and implementation of a scalable IP address plan if you study a detailed example of how a VLSM network is laid out.

Figure B-23 shows a detailed view of the same Division X shown in Figure B-21.

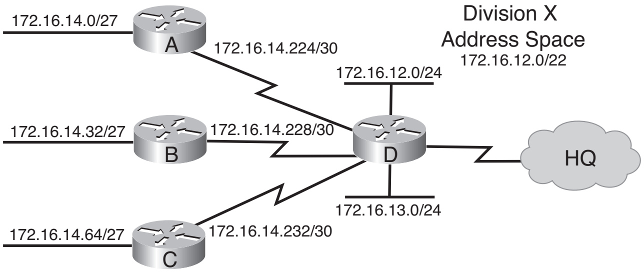


Figure B-23 *Detailed IP Addressing of Division X in Figure B-21.*

In Division X, the following exist:

- One virtual LAN (VLAN) on each of the 2 Ethernet ports of Router D, each with 200 users.
- Three remote sites, at Routers A, B, and C, each with a 24-port Cisco switch. The number of users at each remote site does not exceed 20.
- Three serial links to the remote sites. The serial links are point-to-point Frame Relay and require an address on each side.

VLSM allows you to further subnet the 172.16.12.0/22 address space, using variable masks, to accommodate the network requirements. For example, because point-to-point serial lines require only two host addresses, you can use a subnetted address that has only two host addresses and therefore does not waste scarce subnet numbers.

To start the VLSM process, determine the number of subnets necessary for the networks to which you need to assign IP addresses, and determine the number of hosts necessary per subnetwork. You can determine the number of hosts by checking corporate policy to see whether a limit is set per segment or VLAN, checking the physical number of ports on a switch, and checking the current size of the network or networks at other sites that fulfill the same role.

Note The decimal-to-binary conversion chart earlier in this appendix might be helpful when you are calculating VLSMs.

LAN Addresses

Because IP addresses are binary, they are used in blocks of powers of 2. A block of addresses contains 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, and so on addresses. Two addresses are lost each time you create a subnet: one for the network (wire) address and the other for the directed broadcast address.

The lowest address of the range, where the host bits are all 0s, is known as the network number or the wire address. The top of the address range, where the host bits are all 1s, is the directed broadcast address. The number of addresses in a block that can be assigned to devices is $2^b - 2$, where b is the number of host bits. For example, with 3 host bits, $2^3 - 2 = 8 - 2 = 6$ addresses can be assigned.

To determine the size of the block of addresses needed for a subnet, follow these steps:

Step 1. Calculate the maximum number of hosts on that subnet.

Step 2. Add 2 to that number for the broadcast and subnet numbers.

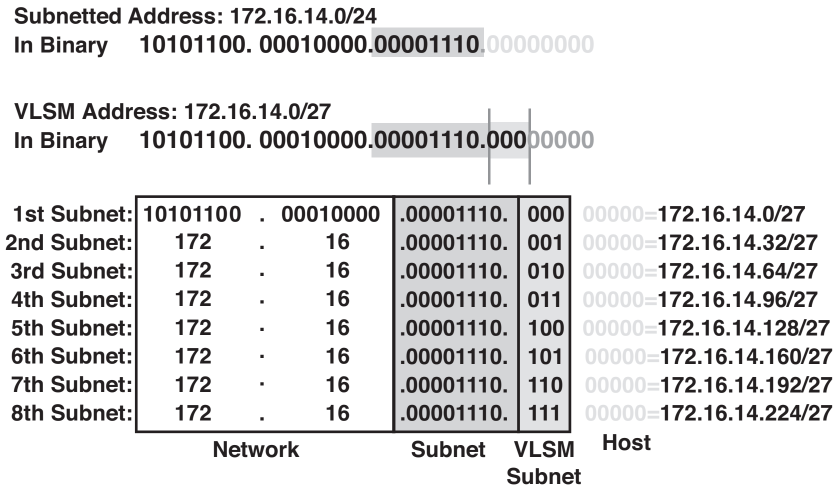
Step 3. Round up to the next higher power of 2.

In this example, the VLANs each have 200 users. Therefore, the number of addresses required is 200 + 2 = 202. Rounding up to the next power of 2 gives you 256. Thus, 8 ($2^8 = 256$) host bits are required for the VLANs. Therefore, the prefix is /24 (32 bits – 8 bits for the host = 24 bits). The network administrator subnets the 172.16.12.0/22 into four /24 subnets on Router D.

172.16.12.0/24 is assigned to VLAN 1, and 172.16.13.0/24 is assigned to VLAN 2. This leaves two /24 subnets, 172.16.14.0/24 and 172.16.15.0/24, to use for the switches at the three remote sites and the three serial point-to-point links.

The number of addresses required for the LANs at each remote site is 20 + 2 = 22. Rounding this up to the next power of 2 gives you 32. Thus, 5 host bits ($2^5 = 32$) are required to address the remote users at each site. Therefore, the prefix to use is /27 (32 bits – 5 bits for the host = 27).

You cannot use the 172.16.12.0/24 or 172.16.13.0/24 networks, because they are assigned to VLANs 1 and 2 on Router D. The process to further subnet 172.16.14.0/24 into /27 subnets is shown in Figure B-24. The first three subnets calculated in Figure B-24 are used on the LANs in Figure B-23.



Subnetted Address: 172.16.14.0/24
In Binary 10101100. 00010000.00001110.00000000

VLSM Address: 172.16.14.0/27
In Binary 10101100. 00010000.00001110.00000000

	Network			Subnet	VLSM Subnet	Host
1st Subnet:	10101100	.	00010000	.00001110.	000	00000=172.16.14.0/27
2nd Subnet:	172	.	16	.00001110.	001	00000=172.16.14.32/27
3rd Subnet:	172	.	16	.00001110.	010	00000=172.16.14.64/27
4th Subnet:	172	.	16	.00001110.	011	00000=172.16.14.96/27
5th Subnet:	172	.	16	.00001110.	100	00000=172.16.14.128/27
6th Subnet:	172	.	16	.00001110.	101	00000=172.16.14.160/27
7th Subnet:	172	.	16	.00001110.	110	00000=172.16.14.192/27
8th Subnet:	172	.	16	.00001110.	111	00000=172.16.14.224/27

Figure B-24 *Calculating Subnet Addresses for the LANs in Figure B-23.*

Serial Line Addresses

After you establish the addresses for the LANs at the remote sites, you must address the serial links between the remote sites and Router D. Because the serial links require two addresses, the number of addresses required is 2 + 2 = 4 (the 2 additional addresses are for the network number and the directed broadcast address).

Note Because only two devices exist on point-to-point links, a specification has been developed (as documented in RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*) to allow the use of only 1 host bit on such links, resulting in a /31 mask. The two addresses created—with the host bit equal to 0 and with the host bit equal to 1—are interpreted as the addresses of the interfaces on either end of the link rather than as the subnet address and the directed broadcast address. Support for /31 masks is provided on some Cisco devices running IOS Release 12.2 and later. You can find details about the support for this (and other features) on specific platforms and IOS releases at the Cisco feature navigator site (http://www.cisco.com/go/fn). In the example in this section, we do not assume the use of this feature.

In this case, there is no need to round up, because 4 is a power of 2. Therefore, 2 host bits will allow for two hosts per subnet. A network mask of /30 (32 bits – 2 host bits = 30 bits) is used. This prefix allows for only two hosts—just enough hosts for a point-to-point connection between a pair of routers.

To calculate the subnet addresses for the WAN links, further subnet one of the unused /27 subnets. In this example, 172.16.14.224/27 is further subnetted with a prefix of /30. The three additional subnet bits result in 2^3 = 8 subnets for the WAN links.

It is important to remember that only *unused* subnets should be further subnetted. In other words, if you use any addresses from a subnet, that subnet should not be further subnetted. In Figure B-23, three subnet numbers are used on the LANs. Another, as-yet-unused subnet, 172.16.14.224/27, is further subnetted for use on the WANs.

The WAN addresses derived from 172.16.14.224/27 are as follows. The shaded bits are the 3 additional subnet bits:

- 172.16.14.11100000 = 172.16.14.224/30
- 172.16.14.11100100 = 172.16.14.228/30
- 172.16.14.11101000 = 172.16.14.232/30
- 172.16.14.11101100 = 172.16.14.236/30
- 172.16.14.11110000 = 172.16.14.240/30
- 172.16.14.11110100 = 172.16.14.244/30
- 172.16.14.11111000 = 172.16.14.248/30
- 172.16.14.11111100 = 172.16.14.252/30

The first three of these subnets are used on the WANs shown in Figure B-23. The address information for the Router A to Router D link is as follows:

- **Network number—**172.16.14.224
- **Router A serial interface—**172.16.14.225
- **Router D serial interface—**172.16.14.226
- **Broadcast address—**172.16.14.227

The address information for the Router B to Router D link is as follows:

- **Network number—**172.16.14.228
- **Router B serial interface—**172.16.14.229
- **Router D serial interface—** 172.16.14.230
- **Broadcast address—**172.16.14.231

The address information for the Router C to Router D link is as follows:

- **Network number—**172.16.14.232
- **Router C serial interface—**172.16.14.233
- **Router D serial interface—**172.16.14.234

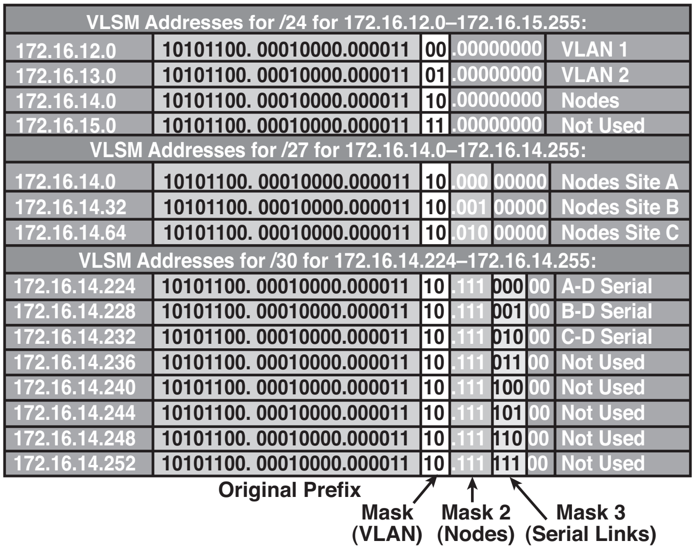
- **Broadcast address—**172.16.14.235

Note that to provide the most flexibility for future growth, the 172.16.14.224/27 subnet was selected for the WANs instead of using the next available subnet, 172.16.14.96/27. For example, if the company purchases more switches, the next IP segment could be assigned the 172.16.14.96/27 subnet, and the new remote site would be connected to Router D with the 172.16.14.236/30 serial subnet.

The 172.16.15.0/24 block could have been used for these /30 subnets, but only three subnets are currently needed, so a lot of the address space would be unused. The 172.16.15.0/24 block is now available to use on another LAN in the future.

Summary of Addresses Used in the VLSM Example

Figure B-25 summarizes the addresses, in binary, used in this example.



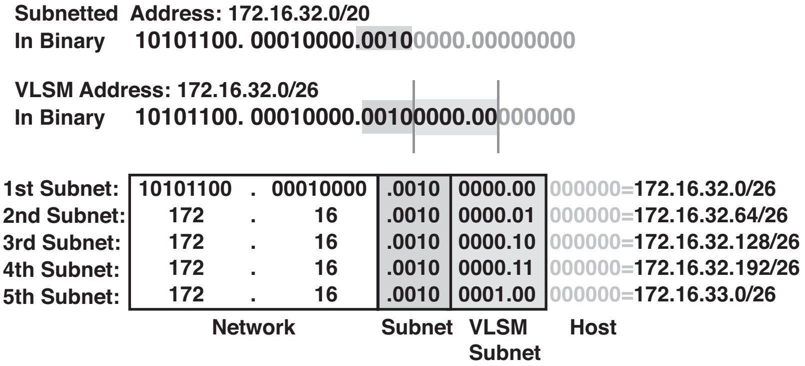
VLSM Addresses for /24 for 172.16.12.0–172.16.15.255:						
172.16.12.0	10101100. 00010000.000011	00	.00000000			VLAN 1
172.16.13.0	10101100. 00010000.000011	01	.00000000			VLAN 2
172.16.14.0	10101100. 00010000.000011	10	.00000000			Nodes
172.16.15.0	10101100. 00010000.000011	11	.00000000			Not Used
VLSM Addresses for /27 for 172.16.14.0–172.16.14.255:						
172.16.14.0	10101100. 00010000.000011	10	.000	00000		Nodes Site A
172.16.14.32	10101100. 00010000.000011	10	.001	00000		Nodes Site B
172.16.14.64	10101100. 00010000.000011	10	.010	00000		Nodes Site C
VLSM Addresses for /30 for 172.16.14.224–172.16.14.255:						
172.16.14.224	10101100. 00010000.000011	10	.111	000	00	A-D Serial
172.16.14.228	10101100. 00010000.000011	10	.111	001	00	B-D Serial
172.16.14.232	10101100. 00010000.000011	10	.111	010	00	C-D Serial
172.16.14.236	10101100. 00010000.000011	10	.111	011	00	Not Used
172.16.14.240	10101100. 00010000.000011	10	.111	100	00	Not Used
172.16.14.244	10101100. 00010000.000011	10	.111	101	00	Not Used
172.16.14.248	10101100. 00010000.000011	10	.111	110	00	Not Used
172.16.14.252	10101100. 00010000.000011	10	.111	111	00	Not Used

Figure B-25 *Binary Representation of the Addresses Used in Figure B-23.*

Another VLSM Example

This section illustrates another example of calculating VLSM addresses. In this example, you have a subnet address 172.16.32.0/20, and you need to assign addresses to a network that has 50 hosts. With this subnet address, however, you have $2^{12} - 2 = 4094$ host addresses, so you would be wasting more than 4000 IP addresses. With VLSM, you can further subnet the address 172.16.32.0/20 to give you more subnetwork addresses and fewer hosts per network, which would work better in this network topology. For example, if you subnet 172.16.32.0/20 to 172.16.32.0/26, you gain 64 (2^6) subnets, each of which can support 62 ($2^6 - 2$) hosts.

To further subnet 172.16.32.0/20 to 172.16.32.0/26, do the following, as illustrated in Figure B-26:



Subnetted Address: 172.16.32.0/20
In Binary 10101100. 00010000.00100000.00000000

VLSM Address: 172.16.32.0/26
In Binary 10101100. 00010000.00100000.00000000

	Network			Subnet	VLSM Subnet	Host
1st Subnet:	10101100	.	00010000	.0010	0000.00	000000=172.16.32.0/26
2nd Subnet:	172	.	16	.0010	0000.01	000000=172.16.32.64/26
3rd Subnet:	172	.	16	.0010	0000.10	000000=172.16.32.128/26
4th Subnet:	172	.	16	.0010	0000.11	000000=172.16.32.192/26
5th Subnet:	172	.	16	.0010	0001.00	000000=172.16.33.0/26

Figure B-26 *Further Subnetting a Subnetted Address.*

Step 1. Write 172.16.32.0 in binary.

Step 2. Draw a vertical line between the 20th and 21st bits, as shown in Figure B-26. This is the transition point between the original subnet bits and the VLSM subnet bits.

Step 3. Draw a vertical line between the 26th and 27th bits, as shown in Figure B-26. This is the transition point between the VLSM subnet bits and the host bits.

Step 4. Calculate the 64 subnet addresses using the bits between the two vertical lines, from lowest to highest. Figure B-26 shows the first five subnets available.

Route Summarization

As the result of corporate expansion and mergers, the number of subnets and network addresses in routing tables is increasing rapidly. This growth taxes CPU resources, memory, and bandwidth used to maintain the routing table. Route summarization and CIDR techniques can manage this corporate growth much like Internet growth has been managed. With a thorough understanding of route summarization and CIDR, you can implement a scalable network. This section describes summarization (CIDR is covered in the later section "Classless Interdomain Routing"). The relationship between summarization and VLSM is also examined. With VLSM, you break a block of addresses into smaller subnets. In route summarization, a group of subnets is rolled up into a summarized routing table entry.

Route Summarization Overview

In large internetworks, hundreds, or even thousands, of network addresses can exist. It is often problematic for routers to maintain this volume of routes in their routing tables. As mentioned in the "IPv4 Address Planning" section earlier, route summarization can reduce the number of routes that a router must maintain, because it is a method of representing a series of network numbers in a single summary address.

For example, in Figure B-27, Router D can either send four routing update entries or summarize the four addresses into a single network number. If Router D summarizes the information into a single network number entry, the following things happen:

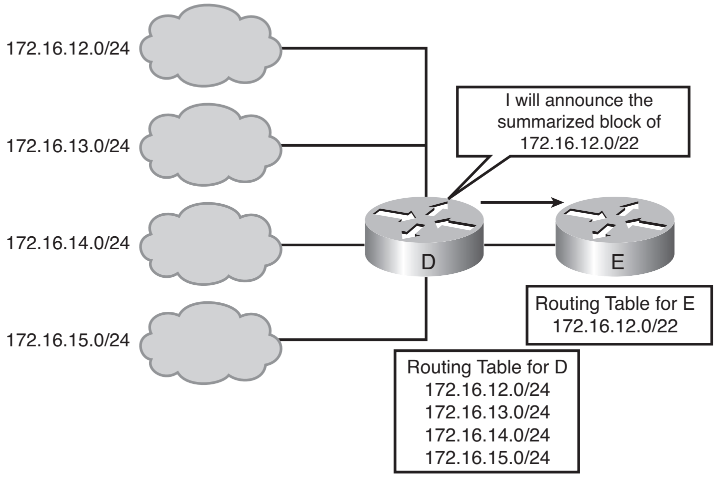


Figure B-27 *Routers Can Summarize to Reduce the Number of Routes.*

- Bandwidth is saved on the link between Routers D and E.
- Router E needs to maintain only one route and therefore saves memory.
- Router E also saves CPU resources because it evaluates packets against fewer entries in its routing table.

A summary route is announced by the summarizing router as long as at least one specific route in its routing table matches the summary route.

Another advantage of using route summarization in a large, complex network is that it can isolate topology changes from other routers. For example, in Figure B-27, if a specific subnet (such as 172.16.13.0/24) is *flapping* (going up and down rapidly), the summary route (172.16.12.0/22) does not change. Therefore, Router E does not need to continually modify its routing table as a result of this flapping activity.

Note *Flapping* is a common term used to describe intermittent interface or link failures.

Route summarization is possible only when a proper addressing plan is in place. Route summarization is most effective within a subnetted environment when the network addresses are in contiguous blocks in powers of 2. For example, 4, 16, or 512 addresses can be represented by a single routing entry because summary masks are binary masks—just like subnet masks—so summarization must take place on binary boundaries (powers of 2). If the number of network addresses is not contiguous or not a power of 2, you can divide the addresses into groups and try to summarize the groups separately.

Routing protocols summarize or aggregate routes based on shared network numbers within the network. Classless routing protocols (such as RIPv2, OSPF, IS-IS, and EIGRP) support route summarization based on subnet addresses, including VLSM addressing. Classful routing protocols (such as RIPv1) automatically summarize routes on the classful network boundary and do not support summarization on any other bit boundaries. Classless routing protocols support summarization on any bit boundary.

Note Summarization is described in RFC 1518, *An Architecture for IP Address Allocation with CIDR*.

Route Summarization Calculation Example

Router D in Figure B-27 has the following networks in its routing table:

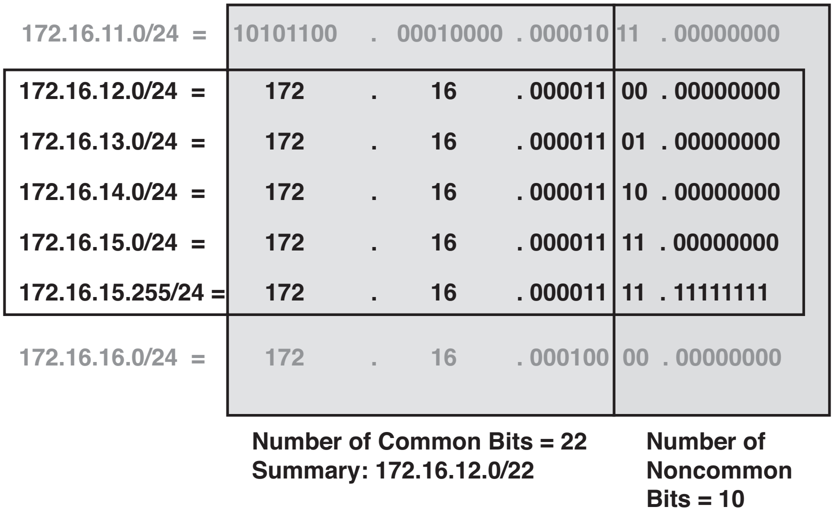
- 172.16.12.0/24
- 172.16.13.0/24
- 172.16.14.0/24
- 172.16.15.0/24

To determine the summary route on Router D, determine the number of highest-order (far left) bits that match in all the addresses. To calculate the summary route, follow these steps:

Step 1. Convert the addresses to binary format and align them in a list.

Step 2. Locate the bit where the common pattern of digits ends. (It might be helpful to draw a vertical line marking the last matching bit in the common pattern.)

Step 3. Count the number of common bits. The summary route number is represented by the first IP address in the block, followed by a slash, followed by the number of common bits. As Figure B-28 illustrates, the first 22 bits of the IP addresses from 172.16.12.0 through 172.16.15.255 are the same. Therefore, the best summary route is 172.16.12.0/22.



172.16.11.0/24 =	10101100	. 00010000	. 000010	11	. 00000000
172.16.12.0/24 =	172	. 16	. 000011	00	. 00000000
172.16.13.0/24 =	172	. 16	. 000011	01	. 00000000
172.16.14.0/24 =	172	. 16	. 000011	10	. 00000000
172.16.15.0/24 =	172	. 16	. 000011	11	. 00000000
172.16.15.255/24 =	172	. 16	. 000011	11	. 11111111
172.16.16.0/24 =	172	. 16	. 000100	00	. 00000000

Figure B-28 *Summarizing Within an Octet, for Router D in Figure B-27.*

Note In this network, the four subnets are contiguous, and the summary route covers all the addresses in the four subnets and only those addresses. Consider, for example, what would happen if 172.16.13.0/24 were not behind Router D, but instead were used elsewhere in the network, and only the other three subnets were behind Router D. The summary route 172.16.12.0/22 should no longer be used on Router D, because it includes 172.16.13.0/24 and might result in confusing routing tables. (However, this depends on how other routers in the network summarize. If the 172.16.13.0/24 route is propagated to all routers, they choose the route with the most bits that match the destination address and should route properly. This is further described in the section “Route Summarization Operation in Cisco Routers.”)

Note In Figure B-28, the subnets before and after the subnets to be summarized are also shown. Observe that they do not have the same first 22 bits in common and therefore are not covered by the 172.16.12.0/22 summary route.

Summarizing Addresses in a VLSM-Designed Network

A VLSM design allows for maximum use of IP addresses and more-efficient routing update communication when using hierarchical IP addressing. In Figure B-29, route summarization occurs at the following two levels:

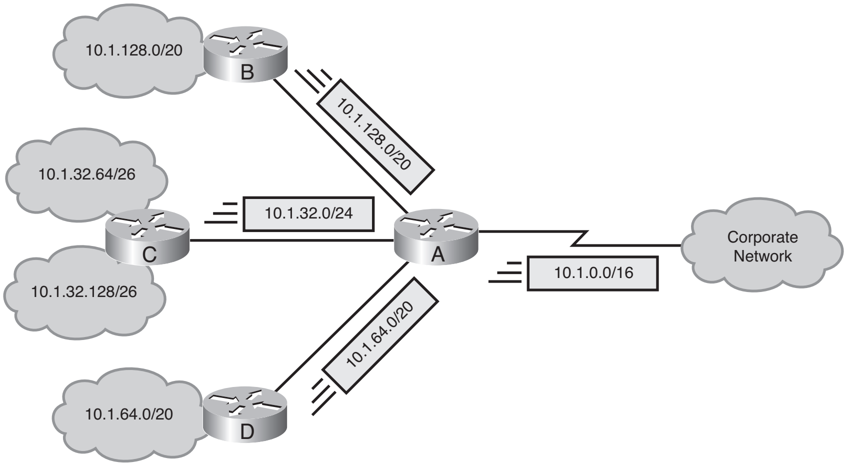


Figure B-29 *VLSM Addresses Can Be Summarized.*

- Router C summarizes two routing updates from networks 10.1.32.64/26 and 10.1.32.128/26 into a single update: 10.1.32.0/24.
- Router A receives three different routing updates. However, Router A summarizes them into a single routing update, 10.1.0.0/16, before propagating it to the corporate network.

Route Summarization Implementation

Route summarization reduces memory use on routers and routing protocol network traffic, because it results in fewer entries in the routing table (on the routers that receive the summarized routes). For summarization to work correctly, the following requirements must be met:

- Multiple IP addresses must share the same highest-order bits.
- Routing protocols must base their routing decisions on a 32-bit IP address and a prefix length that can be up to 32 bits.
- Routing updates must carry the prefix length (the subnet mask) along with the 32-bit IP address.

Route Summarization Operation in Cisco Routers

This section discusses generalities of how Cisco routers handle route summarization. Details about how route summarization operates with a specific protocol are discussed in the corresponding protocol chapter of this book.

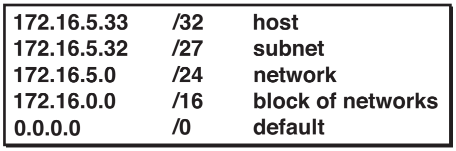
Cisco routers manage route summarization in two ways:

- **Sending route summaries—**Routing information advertised out an interface is automatically summarized at major (classful) network address boundaries by RIP and

EIGRP. Specifically, this automatic summarization occurs for routes whose classful network address differs from the major network address of the interface to which the advertisement is being sent. For OSPF and IS-IS, you must configure summarization.

Route summarization is not always a solution. You would not want to use route summarization if you needed to advertise all networks across a boundary, such as when you have discontiguous networks. When using EIGRP and RIPv2, you can disable this automatic summarization.

- **Selecting routes from route summaries—**If more than one entry in the routing table matches a particular destination, the longest prefix match in the routing table is used. Several routes might match one destination, but the longest matching prefix is used. For example, if a routing table has the paths shown in Figure B-30, packets addressed to destination 172.16.5.99 are routed through the 172.16.5.0/24 path, because that address has the longest match with the destination address.



172.16.5.33	/32	host
172.16.5.32	/27	subnet
172.16.5.0	/24	network
172.16.0.0	/16	block of networks
0.0.0.0	/0	default

Figure B-30 *Routers Use the Longest Match When Selecting a Route.*

Note When running classful protocols (for example, RIPv1), you must enable **ip classless** if you want the router to select a default route when it must route to an unknown subnet of a network for which it knows some subnets. See the "The **ip classless** Command" section in Chapter 1 for more details.

Note that by default (and for historical reasons) the routing table on Cisco routers acts in a classful manner, as described in the sidebar "The Routing Table Acts Classfully" in Chapter 1.

Route Summarization in IP Routing Protocols

Table B-23 summarizes the route summarization support available in the various IP routing protocols.

Classless Interdomain Routing

CIDR is a mechanism developed to help alleviate the problem of exhaustion of IP addresses and growth of routing tables. The idea behind CIDR is that blocks of multiple addresses (for example, blocks of Class C address) can be combined, or aggregated, to create a larger classless set of IP addresses, with more hosts allowed. Blocks of Class C network numbers are allocated to each network service provider. Organizations using the network service provider for Internet connectivity are allocated subsets of the service provider's address space as required. These multiple Class C addresses can then be sum-

Table B-23 *Routing Protocol Route Summarization Support*

Protocol	Automatic Summarization at Classful Network Boundary?	Capability to Turn Off Automatic Summarization?	Capability to Summarize at Other Than a Classful Network Boundary?
RIPv1	Yes	No	No
RIPv2	Yes	Yes	Yes
IGRP[1]	Yes	No	No
EIGRP	Yes	Yes	Yes
OSPF	No	—	Yes
IS-IS	No	—	Yes

[1] Interior Gateway Routing Protocol (IGRP) is no longer supported, as of Cisco IOS Release 12.3.

marized in routing tables, resulting in fewer route advertisements. (Note that the CIDR mechanism can be applied to blocks of Class A, B, and C addresses. It is not restricted to Class C.)

Note CIDR is described further in RFC 1518, *An Architecture for IP Address Allocation with CIDR*, and RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*. RFC 2050, *Internet Registry IP Allocation Guidelines*, specifies guidelines for the allocation of IP addresses.

Note that the difference between CIDR and route summarization is that route summarization is generally done within, or up to, a classful boundary, whereas CIDR combines several classful networks.

CIDR Example

Figure B-31 shows an example of CIDR and route summarization. The Class C network addresses 192.168.8.0/24 through 192.168.15.0/24 are being used and are being advertised to the Internet service provider (ISP) router. When the ISP router advertises the available networks, it can summarize these into one route instead of separately advertising the eight Class C networks. By advertising 192.168.8.0/21, the ISP router indicates that it can get to all destination addresses whose first 21 bits are the same as the first 21 bits of the address 192.168.8.0.

The mechanism used to calculate the summary route to advertise is the same as shown in the earlier “Route Summarization” section. The Class C network addresses 192.168.8.0/24 through 192.168.15.0/24 are being used and are being advertised to the ISP router. To summarize these addresses, find the common bits, as shown here (in bold):

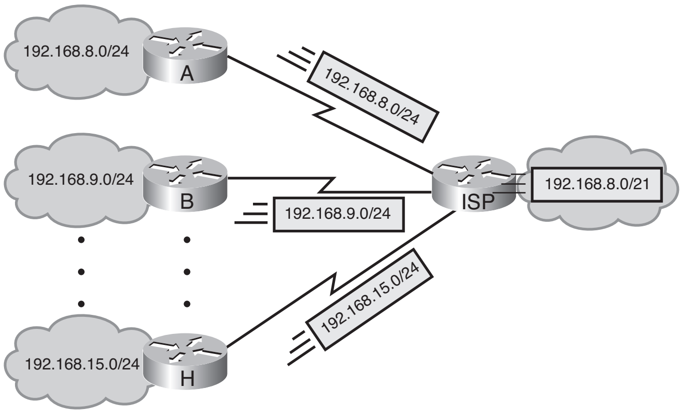


Figure B-31 *CIDR Allows a Router to Summarize Multiple Class C Addresses.*

192.168.8.0	192.168.**00001**000.00000000
192.168.9.0	192.168.**00001**001.00000000
192.168.10.0	192.168.**00001**010.00000000
. . .	
192.168.14.0	192.168.**00001**110.00000000
192.168.15.0	192.168.**00001**111.00000000

The route 192.168.00001*xxx.xxxxxxxx* or 192.168.8.0/21 (also written as 192.168.8.0 255.255.248.0) summarizes these eight routes.

In this example, the first octet is 192, which identifies the networks as Class C networks. Combining these Class C networks into a block of addresses with a mask of less than /24 (the default Class C network mask) indicates that CIDR, not route summarization, is being performed.

In this example, the eight separate 192.168.*x*.0 Class C networks that have the prefix /24 are combined into a single summarized block of 192.168.8.0/21. (At some other point in the network, this summarized block may be further combined into 192.168.0.0/16, and so on.)

Consider another example. A company that uses four Class B networks has the IP addresses 172.16.0.0/16 for Division A, 172.17.0.0/16 for Division B, 172.18.0.0/16 for Division C, and 172.19.0.0/16 for Division D. They can all be summarized as a single block: 172.16.0.0/14. This one entry represents the whole block of four Class B networks. This process is CIDR. The summarization goes beyond the Class B boundaries.