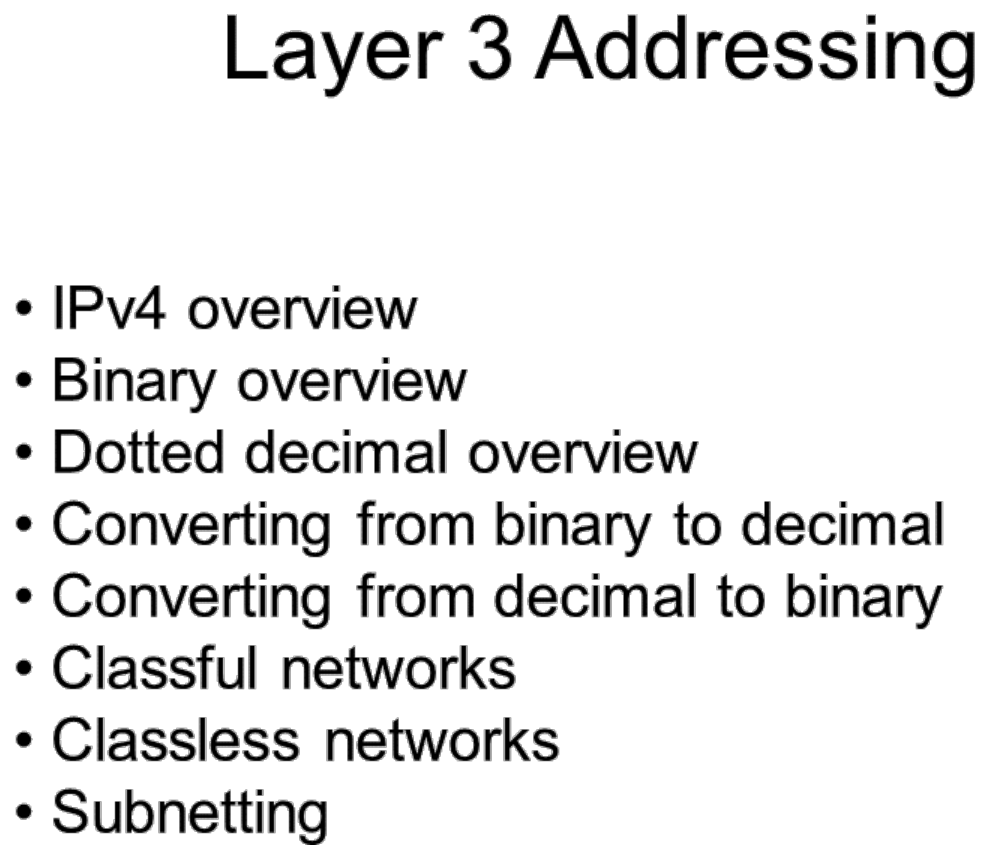
CCNA Curriculum v4 | Courseware



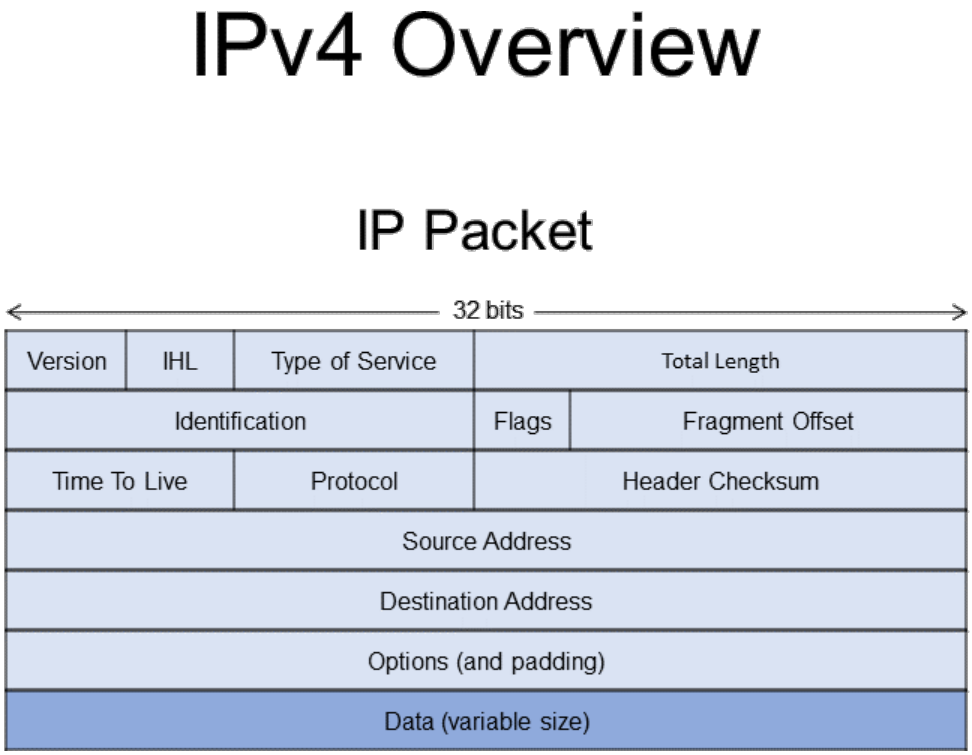
Layer 3 Addressing

Layer 3 addresses are assigned to devices that use Internet Protocol (IP) in order to exchange data on a network. IP is a routed protocol. Routed protocols, which are also called routable protocols, define the way that information is packaged and sent from one network device to another. Routed protocols do not determine the path from the source address to the destination address; that function is handled by routing protocols. Routed protocols simply address the packet of information and allow the routing protocols to determine the path from the source address to the destination address.

Like Layer 2 addressing, the purpose of a Layer 3 address is to pinpoint the location of a particular device on the network. However, in contrast to Layer 2 addresses, Layer 3 addresses are not built into a piece of hardware. Instead, Layer 3 addresses are typically assigned by a network administrator to a particular device. Because these addresses are not tied to a particular piece of hardware, the addresses can be reused when a device is upgraded or replaced with another device. There are two general categories of IP addresses: public IP addresses and private IP addresses. Public IP addresses are globally unique addresses that are assigned by the Internet Assigned Numbers Authority (IANA) to large companies and to Internet service providers (ISPs). Private IP addresses are locally unique

addresses that are not intended to be routable across public networks, such as the Internet; instead they are intended to be used only on internal networks. Although there is nothing that would inherently prevent private IP addresses from being routed on the Internet, most providers filter private address ranges at their borders.

There are two versions of IP used in networking today: IP version 4 (IPv4) and IP version 6 (IPv6). IPv6 was developed to address the issue of IPv4 nearing its limit on the number of available addresses. IPv4 will be the focus of the material presented in this section, unless otherwise specified.



IPv4 Overview

IP is a routed protocol and a logical addressing method that operates at the Network layer of the OSI model. IPv4 supports unicast, multicast, and broadcast addressing of packets.

A basic IPv4 header without options is 20 octets in length; 20 octets is equal to 20 bytes, or 160 bits. An IP header contains the following fields:

Version – 4-bit field that specifies the IP version, such as IPv4 or IPv6

IP Header Length (IHL) – 4-bit field that specifies the number of 32-bit sections, or "words," in the IP header

Type of Service – 8-bit field that indicates the importance of a packet by specifying the quality of service desired

Total Length – 16-bit field that indicates the number of bytes, or octets, in the entire IP packet

Identification – 16-bit field that identifies the current packet, which is used to re-assemble packet fragments

Flags – 3-bit field that is used to control fragmentation

Fragment Offset – 13-bit field that indicates the position of the data in a fragmented packet so that the packet can be re-assembled correctly

Time To Live (TTL) – 8-bit field that acts as a countdown timer; when the timer reaches zero, the packet should be discarded

Protocol – 8-bit field that specifies the upper-layer protocol that should process the packet after IP processing is complete

Header Checksum – 16-bit field that is used to verify the integrity of the IP header to ensure that it was not modified or corrupted in transit

Source Address – 32-bit field that specifies the source IP address

Destination Address – 32-bit field that specifies the destination IP address

Options – variable-length field that specifies other assorted IP options, such as security and source routing parameters

Data is part of the IP packet but is not part of the header.

Because an IP address is a 32-bit number, the range of possible values is ultimately limited to 2^{32} unique addresses. Within that predefined range of possible addresses, some IP address ranges are set aside for specific purposes. The following are some of those specific purposes:

Loopback addresses – All addresses in the 127.0.0.0/8 network are reserved for testing purposes. For example, you can use the **ping 127.0.0.1** command to verify that Transmission Control Protocol/ Internet Protocol (TCP/IP) is operating correctly on your computer or router. After you issue the **ping 127.0.0.1** command, your computer or router should receive Echo Reply packets if TCP/IP is operating correctly.

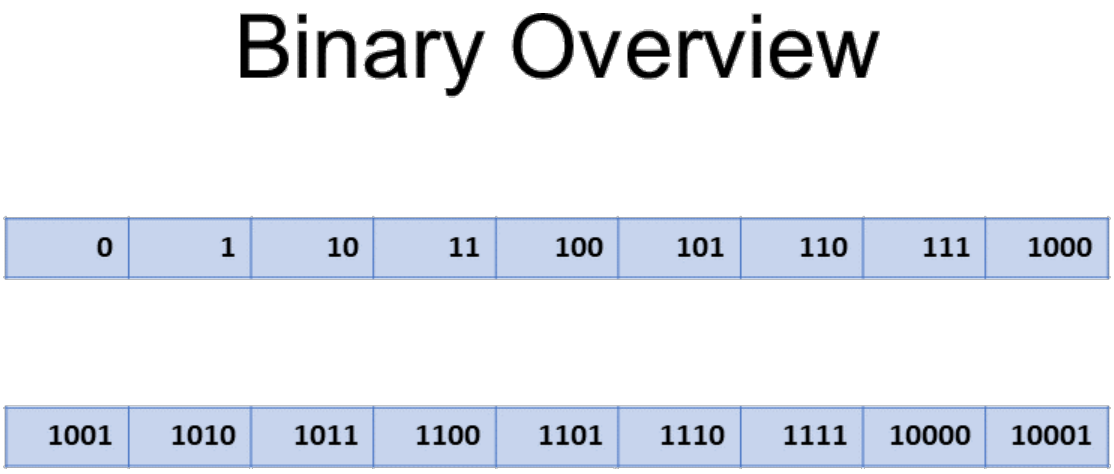
Private IP addresses – 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 are not routable across the Internet; they are used only on internal networks. Network Address Translation (NAT) can be used to translate private IP addresses to public IP addresses so that internal devices can communicate on an external network, such as the Internet.

Automatic Private IP Addresses (APIPA) – 169.254.0.0/16 addresses are link-local private addresses that are randomly generated by a client when the client cannot dynamically obtain an IP address. Request for Comments (RFC) 3927 specifies that APIPA addresses be within the 169.254.0.0 through 169.254.255.255 address range; however, the first 256 and last 256 addresses from this range

are reserved and must not be assigned to hosts.

Multicast addresses – 224.0.0.0/4 addresses are used to send a single stream of data to multiple devices simultaneously, thereby conserving bandwidth. The 224.0.0.1 multicast address is the All Hosts address, which is used to send multicast information to all hosts on a subnet.

Global broadcast address – 255.255.255.255 is used to send data to every computer on a network or subnetwork.



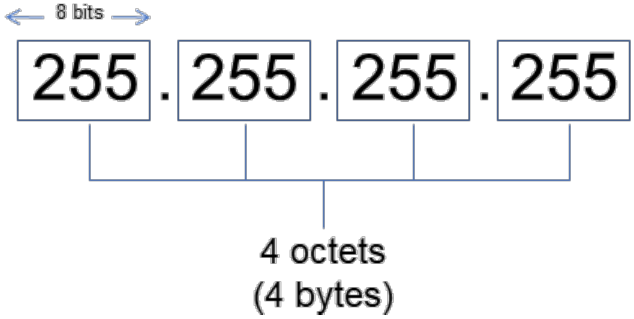
Binary Overview

A binary system is a base-2 system, meaning that every possible number can be composed of only two different character choices; the binary number system used in computers consists of 0s and 1s. Counting in base 2 (binary) is really no different from counting in base 8 (octal), base 10 (decimal), or base 16 (hexadecimal) except for how many character choices you have available to you. Once a position has exhausted all available character choices, another position is added to the right and the new position begins with the first character choice. Binary numbers are read digit by digit, unlike the decimal system where the numbers are read as a representation of the value and position of the characters. For example, 1110 in binary is read one-one-one-zero, whereas that same number in decimal would be read one thousand one hundred and ten.

Dotted Decimal Overview

binary	0	1	10	11	100	101	110	111
decimal	0	1	2	3	4	5	6	7

binary	1000	1001	1010	1011	1100	1101	1110	1111
decimal	8	9	10	11	12	13	14	15



Dotted Decimal Overview

Dotted decimal notation is a way of expressing a binary number as a group of decimal numbers. Because an IP address is 32 bits long, it can easily be divided into four equally sized groups called octets. Each octet contains 8 binary digits, which is a sufficient number of bits to represent any decimal value from 0 through 255.

Converting from Binary to Decimal

	BIT WEIGHT								DECIMAL VALUE
	128	64	32	16	8	4	2	1	
1st OCTET	1	1	1	1	1	1	1	1	255
2nd OCTET	1	1	1	1	1	1	1	1	255
3rd OCTET	1	1	1	1	1	1	1	1	255
4th OCTET	1	1	1	1	1	1	1	1	255

Converting from Binary to Decimal

Each digit in a binary octet corresponds to a decimal value. The leftmost bit in each octet has a decimal value of 128, and the rightmost bit has a decimal value of 1. Because a binary system represents numbers as factors of 2, the decimal value of each bit is halved for each position that you move from the leftmost bit to the rightmost bit. For example, the bit next to the leftmost bit has a decimal value of 64, and the bit to its right has a decimal value of 32. Conversely, if you moved from the rightmost bit to the leftmost bit, the values would double. Thus the bit next to the rightmost bit has a decimal value of 2, and the bit to its left has a decimal value of 4.

You convert an IP address from binary to decimal one octet at a time. Starting from the leftmost octet, the decimal value for each octet is computed by adding up the bit weight for any bit containing a 1 within the octet. For example, the binary number 11111111 is calculated by the following method:

128 (1 x 128)
+64 (1 x 64)
+32 (1 x 32)
+16 (1 x 16)
+ 8 (1 x 8)
+ 4 (1 x 4)
+ 2 (1 x 2)
+ 1 (1 x 1)

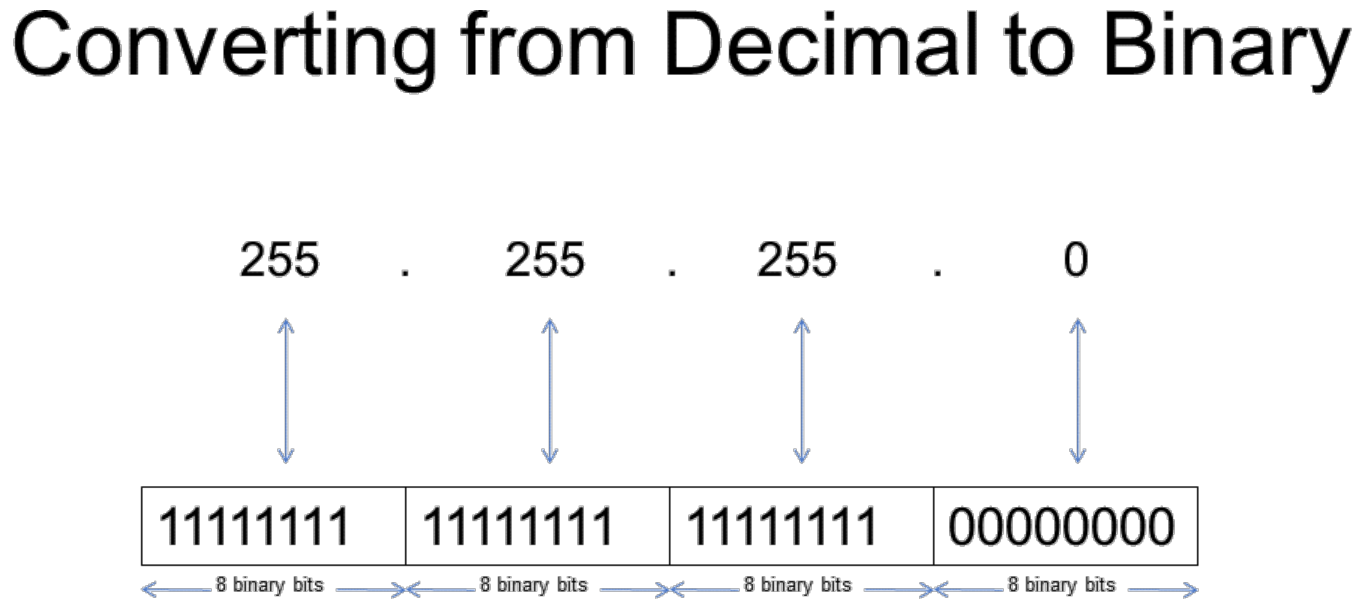
255

Knowing the decimal value of each of the binary digits is the key to converting from binary to decimal. If you see a 1 for a binary place value, you add the corresponding decimal equivalent for that place value. If you see a 0 for a binary place value, you do not add anything. To convert the address 00001010.10101110.00100101.11010100 to decimal, start with the first octet:

0 (0 x 128)
+ 0 (0 x 64)
+ 0 (0 x 32)
+ 0 (0 x 16)
+ 8 (1 x 8)
+ 0 (0 x 4)
+ 2 (1 x 2)
+ 0 (0 x 1)

10

The second octet, 10101110, converts to 128 + 32 + 8 + 4 + 2 = 174; the third octet, 00100101, converts to 32 + 4 + 1 = 37; and the fourth octet, 11010100, converts to 128 + 64 + 16 + 4 = 212. Therefore, the entire binary address 00001010.10101110.00100101.11010100 converts to 10.174.37.212.



Converting from Decimal to Binary

Converting from decimal to binary can be done in two ways: decimal numbers can be converted to binary by using descending powers of 2 with subtraction or by using short division by 2 with remainders.

When using descending powers of 2 with subtraction, you first need to list the powers of 2 from the base 2 table in reverse order:

128

64

32

16

8

4

2

1

To convert a decimal number to binary, begin by asking whether the highest possible power of 2 will go into the decimal number. If the answer to this question is yes, put a 1 in that column; if the answer

is no, put a 0 in that column. If you answered yes, you would subtract the corresponding power of 2 from your original number and take the product to the next lowest power of 2 and repeat the process through each layer of the table. If you answered no, you would put a 0 in the column and take your original number to the next power of 2 in the table. To convert the decimal number 192 to binary by using the descending-power-of-2-with-subtraction method, begin by asking whether 128 will go into 192. The answer is yes, so you should put a 1 in that column:

128 (1)

64

32

16

8

4

2

1

You should then subtract 128 from 192, for a value of 64. Now, repeat this process throughout the remaining powers of 2 in the table. The next power of 2 is 64. The value of 64 is divisible by 64. Therefore, you should put a 1 in that column:

128 (1)

64 (1)

32

16

8

4

2

1

Next, subtract 64 from 64, for a value of 0. If you continue to repeat the process down each layer of the table, you will see that none of the rest of the powers of 2 will go into 0; therefore, put a 0 in each of the columns:

128 (1)

64 (1)

32 (0)

16 (0)

8 (0)

4 (0)

2 (0)

1 (0)

Once you have completed this process, the binary result is found in the digits you have marked in each column; the number 192 in decimal converts to 11000000 in binary.

Using the short-division-by-2-with-remainders method, you would start with the decimal number, divide it by 2, and note the remainder. Then you would take the product of the division and divide that number by 2, again noting the remainder, and continue this process to 0. The binary number is composed of the remainders in reverse order. Using this method to convert 192 in decimal to binary looks like this:

192 ÷ 2 = 96 with a remainder of 0

96 ÷ 2 = 48 with a remainder of 0

48 ÷ 2 = 24 with a remainder of 0

24 ÷ 2 = 12 with a remainder of 0

12 ÷ 2 = 6 with a remainder of 0

6 ÷ 2 = 3 with a remainder of 0

3 ÷ 2 = 1 with a remainder of 1

1 ÷ 2 = 0 with a remainder of 1

Since you have reached 0, your division sequence is complete and the binary number can be read as the remainders in reverse order, or 11000000.

Classful Networks

Class	Purpose	High-order binary bit	Decimal IP range 1st octet	Default mask
Class A	Commercial use	0	0 – 127*	255.0.0.0
Class B	Commercial use	10	128 – 191	255.255.0.0
Class C	Commercial use	110	192 – 223	255.255.255.0
Class D	Multicasting	1110	224 – 239	
Class E	Experimental	1111	240 – 255	

*Class A first octets of 0 and 127 are reserved

Classful Networks

Classful network IP addresses are divided into Class A, Class B, Class C, Class D, and Class E addresses. Classes A, B, and C are available for commercial use. Class D addresses are used for multicast traffic. Class E addresses are reserved for experimental purposes only.

The IP address consists of two parts: a network portion and a host portion. The host and network portions of an IP address in a classful network are divided at octet boundaries, as seen in the default network mask of each class. The network portion of a Class A IP address is in the first octet, the network portion of a Class B IP address is in the first and second octets, and the network portion of a Class C IP address is in the first, second, and third octets.

It is easy to determine the IP address class of a binary IP address by looking at the high-order bits, which are the first few bits of the binary address. Class A addresses start with 0, Class B addresses start with 10, Class C addresses start with 110, Class D addresses start with 1110, and Class E addresses start with 1111. When IP addresses are in decimal notation, you can identify the class of an IP address by the number in the first octet:

Class A – first octet ranges from 0 through 127

Class B – first octet ranges from 128 through 191

Class C – first octet ranges from 192 through 223

Class D – first octet ranges from 224 through 239

Class E – first octet ranges from 240 through 255

Addresses are also divided into public addresses and private addresses. Most IP addresses are public addresses, which are used to communicate over public networks, such as the Internet. Private addresses are not routable over public networks and are typically used to address devices on internal networks. NAT is required in order to route private IP addresses over the Internet.

The following are the valid IP addresses in each of the classes available for commercial use as defined by RFC 1918:

Class A – 10.0.0.0 through 10.255.255.255

Class B – 172.16.0.0 through 172.31.255.255

Class C – 192.168.0.0 through 192.168.255.255

Sample classless subnetting from a single /16 network of 65,534 usable hosts

Binary Mask (1 = network bit, 0 = host bit)	Subnet Mask	CIDR	Networks	Usable Hosts per Network
11111111.11111111.10000000.00000000	255.255.128.0	/17	2	32,766
11111111.11111111.11000000.00000000	255.255.192.0	/18	4	16,382
11111111.11111111.11100000.00000000	255.255.224.0	/19	8	8,190
11111111.11111111.11110000.00000000	255.255.240.0	/20	16	4,094
11111111.11111111.11111000.00000000	255.255.248.0	/21	32	2,046
11111111.11111111.11111100.00000000	255.255.252.0	/22	64	1,022
11111111.11111111.11111110.00000000	255.255.254.0	/23	128	510

Classless Networks

The host and network portions of an IP address in a classless network are not divided at the octet boundaries. The entire IP address is viewed as a 32-bit stream, which is arbitrarily divided according to the requirements of the network. This allows network engineers and administrators to combine and divide networks using variable-length subnet masks (VLSMs) without regard to classful network boundaries, thus making more efficient use of IP address space. This is accomplished with the use of Classless Inter-Domain Routing (CIDR) and results in the formation of address blocks.

The CIDR number used for notation indicates the number of bits that belong to the network portion of a 32-bit IP address, appears after the last octet, and is preceded by a slash before the number. The address 10.1.1.0/18 is an example, where the /18 is the CIDR notation. In this example, 18 bits of the

IP address belong to the network portion and the remaining bits of the IP address belong to the host portion. Because classless networks are not limited to the 8 bits of an octet, the slash notation could be any number from 1 through 31. In contrast, because classful networks are limited to the boundaries of the octets, the slash notation of a classful network will always be /8, /16, or /24.

To calculate how many bits are in the network portion of an address when a subnet mask is used, convert the mask to binary and count how many digits are set to a value of 1. It is possible to determine the number of bits that belong to the network portion of and IP address from the subnet mask 255.255.252.0.

Each of the four dotted decimal numbers in an IP address or subnet mask represents an octet of eight binary digits. Each of the binary digits has a decimal value. A value of 1 for the first binary digit is equal to a decimal value of 128, a value of 1 in the second binary digit is equal to a decimal value of 64, and so on. Setting all eight binary digits equal to 1 is equal to a decimal value of 255, as calculated below:

128 (1 x 128)
+64 (1 x 64)
+32 (1 x 32)
+16 (1 x 16)
+ 8 (1 x 8)
+ 4 (1 x 4)
+ 2 (1 x 2)
+ 1 (1 x 1)

255

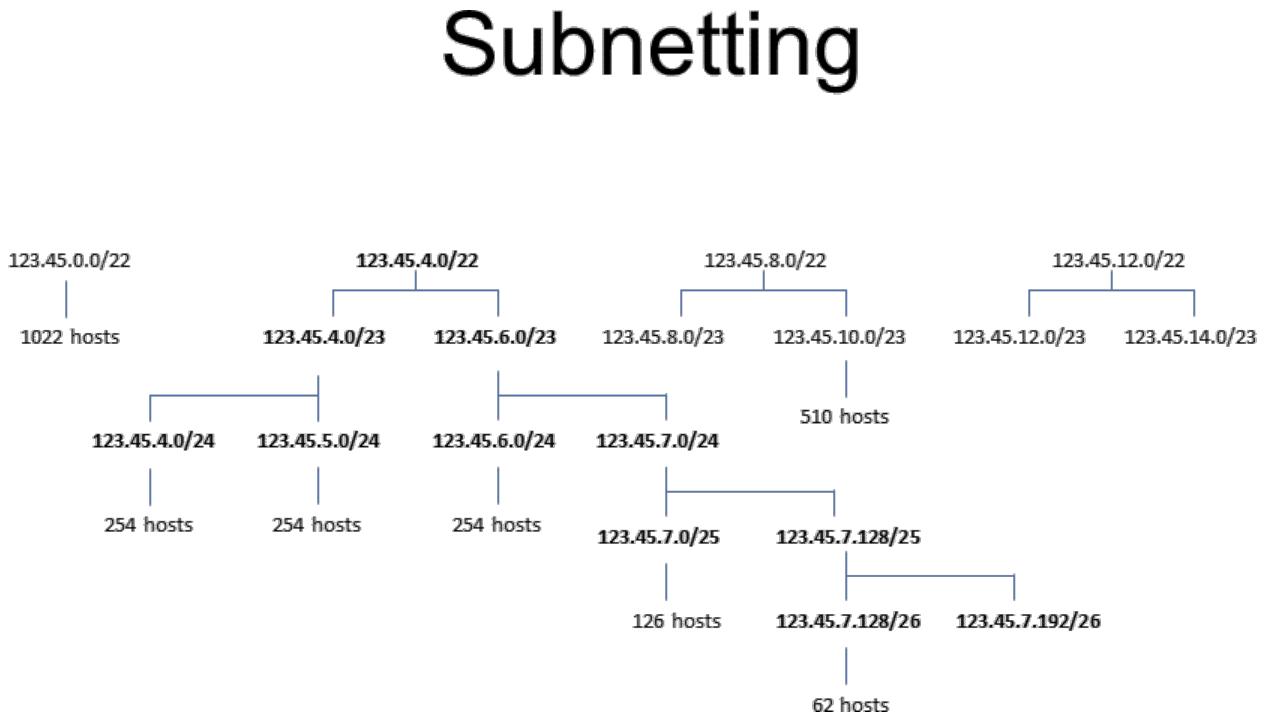
Therefore, each dotted decimal value of 255 is equal to a binary value of 11111111.

In the 255.255.252.0 subnet mask, the third octet, 252, is equal to the binary value 11111100. The last octet, 0, is equal to 00000000. Therefore, the 255.255.252.0 subnet mask is equal to 11111111.11111111.11111100.00000000.

All the digits that are set to a value of 1 indicate the bits that are used for the network portion of an IP address, and all the digits that are set to a value of 0 indicate the bits that are used for the host portion of an IP address. Since the number used for CIDR notation indicates the number of bits that are used for the network portion of the address, you can calculate this number by counting the number of 1s in the binary subnet mask value. The binary subnet mask value 11111111.11111111.11111100.00000000 contains 22 digits equal to 1; therefore, the CIDR subnet mask is /22.

A hierarchical VLSM structure can be used in combination with route summarization to maximize the

use of IP addresses and minimize routing update traffic on a router. Routers summarize networks by grouping subnet addresses into larger subnet addresses. Therefore, a hierarchical VLSM design makes the summarization of routes from that design more efficient. Route summarization is discussed more in the Subnetting section of this module.



Subnetting

Subnetting is a technique used to divide a network into smaller subnetworks. All devices on a network that belong to a given subnet have a common number in the network portion of their IP address and a unique number in the host portion.

To determine the subnetwork address range of a given IP address/subnet mask combination, you must first identify the *interesting octet* within the subnet mask. The interesting octet is the first octet that contains a decimal value other than 255 or 0.

The following example uses the /22 CIDR subnet mask. The subnet mask will need to be converted from CIDR notation to dotted decimal notation in order for you to calculate networks and hosts.

To convert /22 from CIDR notation to dotted decimal notation, begin at the left and set the first 22 bits to a value of **1**. These bits identify the network portion of the IP address. The remaining 10 bits will be set to **0**.

/22 = 11111111.11111111.11111100.00000000

As noted in the Converting from Binary to Decimal section, binary bit weight increases in significance from right to left, with the leftmost bit in each octet worth a decimal value of 128 and the rightmost bit worth a decimal value of 1. The decimal value for each octet is computed by adding up the bit weight

for any bit containing a 1 within the octet. The following exhibit displays how to calculate the decimal value of the subnet mask octets based on the binary value assigned to each bit:

	BIT WEIGHT								DECIMAL VALUE
	128	64	32	16	8	4	2	1	
1st OCTET	1	1	1	1	1	1	1	1	255
2nd OCTET	1	1	1	1	1	1	1	1	255
3rd OCTET	1	1	1	1	1	1	0	0	252
4th OCTET	0	0	0	0	0	0	0	0	0

/22 = 11111111.11111111.11111100.00000000 = 255.255.252.0

Now that the subnet mask is in a dotted decimal notation, the interesting octet is easily identified because the third octet has a decimal value of 252, not 255 or 0.

Once the interesting octet has been identified, the network numbers are determined by the weight of the least significant bit, or the rightmost bit, in the interesting octet that is set to **1**. Another way to determine the network numbers is to simply subtract the decimal value of the interesting octet from 256. The difference between 256 and 252 is 4; therefore, the networks will be arranged in multiples of 4.

	BIT WEIGHT							
	128	64	32	16	8	4	2	1
3rd OCTET	1	1	1	1	1	1	0	0

A partial list of the available networks using the 255.255.252.0 subnet mask in this scenario contains:

48.25.0.0

48.25.4.0

48.25.8.0

48.25.16.0

48.25.24.0

...and so on.

The total number of hosts can be determined by the number of bits equal to 0 in the binary subnet mask. You can then calculate the number of hosts for a given subnetwork by using the formula $2^n - 2$, where n is the number of bits equal to 0 in the subnet mask.

/22 = 11111111.11111111.11111100.00000000

There are 10 bits equal to 0 in the /22 subnet mask. Using the $2^n - 2$ formula, you can see that 1,022 hosts are available for each subnetwork when a subnet mask of /22 is applied: 2^{10} = 1,024, and 1,024 – 2 = 1,022. You must subtract 2 from the number of available hosts because the first address is the subnetwork address and the last address is the broadcast address; neither of those addresses can be assigned to an interface. Taking one bit away from the subnet mask doubles the size of the subnetwork; a 21-bit subnet mask allocates 2,046 host addresses, a 20-bit subnet mask allocates 4,094 host addresses, a 19-bit subnet mask allocates 8,190 host addresses, and so on.

Although it is important to learn how to convert CIDR subnet masks to dotted decimal subnet masks, the following list displays the most commonly used subnet mask conversions:

/8 = 255.0.0.0

/16 = 255.255.0.0

/17 = 255.255.128.0

/18 = 255.255.192.0

/19 = 255.255.224.0

/20 = 255.255.240.0

/21 = 255.255.248.0

/22 = 255.255.252.0

/23 = 255.255.254.0

/24 = 255.255.255.0

/25 = 255.255.255.128

/26 = 255.255.255.192

/27 = 255.255.255.224

/28 = 255.255.255.240

/29 = 255.255.255.248

/30 = 255.255.255.252

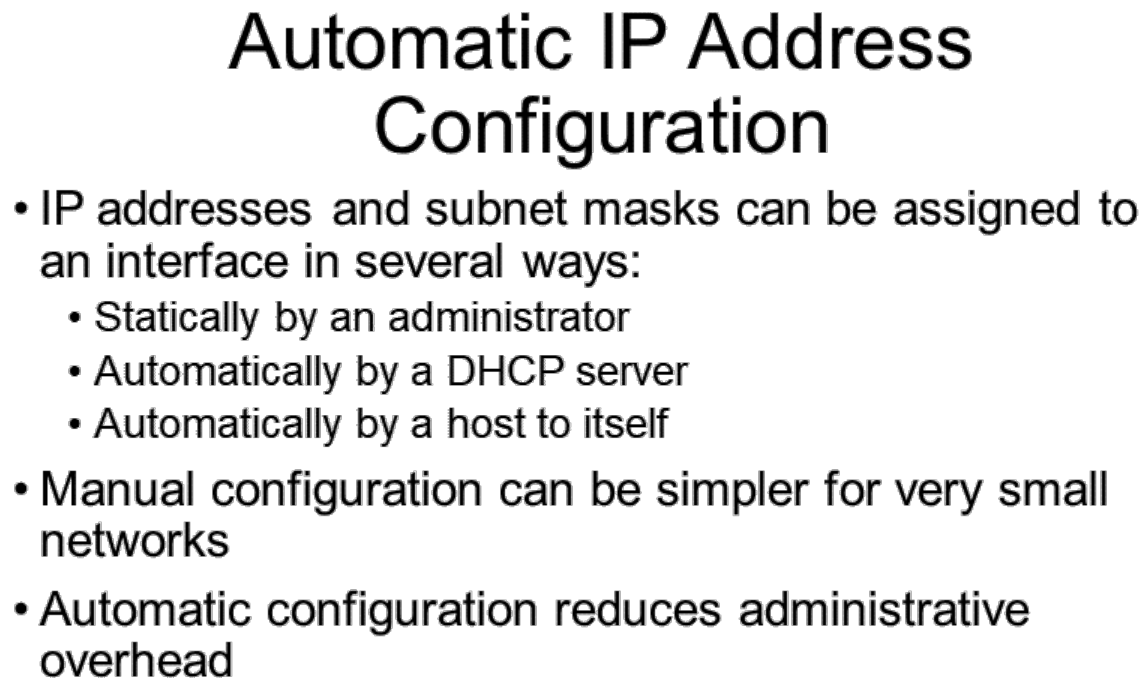
Subnetting and Route Summarization

Subnetting and route summarization can work together so that a network can efficiently use

registered IP addresses and so that the routers on that network are able to minimize routing table information and routing update traffic. Route summarization, which is also known as supernetting, enables a router to advertise multiple contiguous subnets as a single, larger subnet.

The way route summarization works is basically the reverse of how subnetting works; instead of separating an existing subnet into several smaller subnets, summarization combines several smaller subnets into one larger subnet. This enables routers on the network to maintain a single summarized route in their routing tables. Therefore, fewer routes are advertised by the routers, which reduces the amount of bandwidth required by routing update traffic.

Route summarization is most efficient when the subnets can be summarized within a single subnet boundary and are contiguous, meaning that all of the subnets are consecutive. For example, if a router is connected to the 123.45.67.64/30 subnet, the 123.45.67.68/30 subnet, the 123.45.67.72/30 subnet, and the 123.45.67.76/30 subnet, all the subnets configured on that router are contiguous. Therefore, the router could summarize all four /30 subnets and advertise the route to those subnets as a single 123.45.67.64/28 subnet. The 123.45.67.64/28 subnet contains the IP address range from 123.45.67.64 through 123.45.67.79, which summarizes all four /30 networks.



Automatic IP Address Configuration

IPv4 addresses and subnet masks can be manually configured by an administrator, automatically assigned by a Dynamic Host Configuration Protocol (DHCP) server, or automatically assigned to a host by itself. Very small networks can be easily managed by an administrator who manually allocates IP addresses to devices as they are needed. However, a DHCP configuration makes sense for networks in which the manual assignment of IP addresses could produce significant administrative overhead.

As previously mentioned, APIPA addresses are IPv4 addresses that a host can automatically assign to itself when no manual configuration exists and no DHCP server is available for automatic configuration.