Suffix	Hosts	32-Borrowed = CIDR	2^Borrowed = Hosts	Binary => dec = Suffix
.255	1	/32	0	...11111111
.254	2	/31	1	...11111110
.252	4	/30	2	...11111100
.248	8	/29	3	...11111000
.240	16	/28	4	...11110000
.224	32	/27	5	...11100000
.192	64	/26	6	...11000000
.128	128	/25	7	...10000000
Classful / Classless				

CIDR notation	Network Mask	Available sub-networks	Available Hosts per network	Total usable hosts
/24	255.255.255.0	1	256	254
/25	255.255.255.128	2	128	126
/26	255.255.255.192	4	64	62
/27	255.255.255.224	8	32	30
/28	255.255.255.240	16	16	14
/29	255.255.255.248	32	8	6
/30	255.255.255.252	64	4	2
/31	255.255.255.254	128	2	2 *

How to Subnet a Network

How to use this paper

- ♦ Absolute Beginner: Read all Sections 1-4
- ♦ Need a quick review: Read Sections 2-4
- ♦ Just need a little help: Read Section 4

Part I: For the IP Beginner

IP Network Addresses
To understand network IP addressing, let's take a look at postal addresses. Every building must have its own unique address in order for mail to be delivered. An address consists of different parts such as the street, number, and city. In a network, every device must have its own unique IP address. That is, every network device (printer, server, router, etc.) must be identified with a separate IP address.

Many Devices or Hosts
Using the postal example, think about an apartment building. In order for the mail to arrive at the right apartment, each apartment must have its own unique identifier (apartment number) in addition to the street name and address.

With IP addresses, an organization is assigned a unique IP network, such as **192.168.1.0**, but a single IP address must be assigned for each network device. For example, let's assume that ABC company has 5 devices in one of their buildings that need IP addresses, and that ABC is assigned the network address 192.168.1.0. The IP addresses could be assigned as follows:

Device 1 (router): 192.168.1.1
Device 2 (office printer): 192.168.1.2
Device 3 (Ms. Chung's laptop): 192.168.1.3
Device 4 (Receiptionist's computer): 192.168.1.4
Device 5 (company server): 192.168.1.5

Hosts .1, .2, .3, .4, and .5 are all on the 192.168.1 network, just as apartments 1, 2, 3, 4, and 5 might be located in the apartment building at 123 First Street.

When you see an IP address, you will always see another number associated with it that looks something like one of the following:

- 255.255.255.0
- 255.255.0.0
- 255.0.0.0

This number is called the **subnet mask**. A subnet mask is used to show which portion of the IP address identifies the network and which portion identifies a specific host on the network. This may seem unnecessary at first since the first three numbers of an IP address starting from the left (such as 192.168.1) always identify the network, and the last number (such as .1) always identifies an individual device in networks like the one in our example, which is called a standard Class C network. However, this is not true for other networks. That's why subnet masks are used to differentiate the network portion of the address from the host portion. The following table shows the subnet mask 255.255.255.0, which is always the subnet mask for a standard Class C network, applied to one of our example addresses. Notice how the mask reveals which portion of the string 192.168.1.1 is the individual host address.

NOTE: Class C addresses have a number from 192 to 223 in the first octet.

Network portion			Host portion
192	168	1	1
255	255	255	0

The number 255 indicates that the corresponding section of the address is part of the network address. The 0 indicates that the corresponding section is the host portion of the address.

This document and the Subnet game refer only to Class C networks. Once you understand IP addressing for Class C networks, it will be much easier for you to understand IP addressing for any network.

Part II: Binary Numbers

To understand network addresses, we should take a moment to consider the binary number system. Since all electronic devices only understand binary numbers, all network addresses are actually made up of binary digits called **bits**. An IP address consists of 32 bits, broken into four parts called octets. Each octet equals 8 bits. Our sample network address of 192.168.1.0, looks like this in binary.

11000000	10101000	00000001	00000000
192	168	1	0

But how does 11000000 equate to 192? How does 10101000 equate to 168? Here's how . . . . .

A bit can have only two possible values: **on**, which is represented by a **1**, or **off**, which is represented by a **0**.. To represent the decimal number 0, as in the last octet above, all 8 bits are turned off. When turned on, each bit has a value. Let's take a look at one 8-bit octet , since each octet contains a distinct number. The values are as follow:

Bit	1	1	1	1	1	1	1	1
Bit value	128	64	32	16	8	4	2	1

The right-most bit has a value of 1, the next bit has a value of 2, the next bit has a value of 4, and so on as shown in the chart above. The decimal value of an octet is the sum of the bit values. So if all bits are on, the value of the octet is 255, or 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

The following table shows that 11000000 equals 192 in decimal. The bit with a value of 128 and the bit with a value of 64 are turned on, so we add 128 + 64 and get 192.

Bit	1	1	0	0	0	0	0	0
Bit value	**128**	**64**	32	16	8	4	2	1

This table shows that 10101000 equals 168 in decimal. The bits with values of 128, 32, and 8 are turned on, and their sum equals 168.

Bit	1	0	1	0	1	0	0	0
Bit value	**128**	64	**32**	16	**8**	4	2	1

Part III: From IP Addressing to IP Subnetting

Subnetting can be useful in a variety of ways, including simplifying network administration, enabling you to use different physical media such as Ethernet and FDDI, and adding a layer of security to your network. The most common use of subnetting is to control network traffic.

Subnetting is done by borrowing host bits and using them as network bits. To begin, let's look at our ABC company network address (192.168.1.0) and its subnet mask (255.255.255.0) as expressed in binary. Notice that the address bits that have corresponding mask bits set to 1 represent the network address. Address bits that have corresponding mask bits set to 0 represent the individual host address.

Network address	11000000	10101000	00000001	00000000
Subnet mask	11111111	11111111	11111111	00000000

With this address, the bits from octets 1, 2, and 3 are used to identify the network portion of the address. However, we could subnet our network by borrowing bits from the fourth octet. To do so, we must take bits consecutively from left to right. In the following table, we borrow the bit with a value of 128..

Network address	11000000	10101000	00000001	00000000
Subnet mask	11111111	11111111	11111111	**1**0000000

This changes our subnet mask. Instead of 255.255.255.0, it is now 255.255.255.128.

The more host bits you use for subnets, the more subnets you have available. However, as more subnets are created, the less host addresses are available per subnet. In the following table, we borrow both the 128 and the 64 bit. We now have only 6 bits left for host addresses, and our mask is now 255.255.255.192.

Network address	11000000	10101000	00000001	00**000000**
Subnet mask	11111111	11111111	11111111	**11**000000

Notice also that we use the fourth octet to subnet a Class C network. Because this document and the Subnet game refer only to Class C networks, the remainder of this document will focus only on the fourth octet. Once you understand IP subnetting for Class C networks, it will be much easier for you to understand IP subnetting for any network.

So, back to our fourth octet, exactly what network and host addresses can we create by borrowing these two bits and thus applying this .192 mask?

Network address	[redacted]	[redacted]	[redacted]	00**000000**
Subnet mask	[redacted]	[redacted]	[redacted]	**11**000000

Let's look first at the subnets. We have two digits with which to create subnets.

Network address	[redacted]	[redacted]	[redacted]	**00**000000
Subnet mask	[redacted]	[redacted]	[redacted]	11000000

Looking at the bit value chart, it's easy to see that those two digits can be . . .

00, which equals 0

Bit	**0**	**0**	0	0	0	0	0	0
Bit value	128	64	32	16	8	4	2	1

01, which equals 64

Bit	**0**	**1**	0	0	0	0	0	0
Bit value	128	**64**	32	16	8	4	2	1

10, which equals 128

Bit	**1**	**0**	0	0	0	0	0	0
Bit value	**128**	64	32	16	8	4	2	1

or 11, which equals 192

Bit	**1**	**1**	0	0	0	0	0	0
Bit value	**128**	**64**	32	16	8	4	2	1

So we have created the following subnets:
192.168.1.**0**
192.168.1.**64**
192.168.1.**128**
192.168.1.**192**
That means that these numbers (0, 64, 128, and 192) can no longer be host addresses because they are now subnet addresses. Each subnet, like any network, must have a broadcast address as well. The broadcast address is the last address on the network, so on our first subnet, 192.168.1.0, available host addresses are 192.168.1.**1** through 192.168.1.**62.** The 192.168.1.**63** is the broadcast address and 192.168.1.**64** is the address of the next subnet. Remember that before we subnetted our network, we could use all 8 bits of the 4th octet for host addresses. That gave us 254 host addresses. We now have 62 for each of our 4 subnets, or 248 total.

The following table shows the complete results of borrowing two host bits to subnet our Class C network.

Subnet	Network address	Host addresses	Broadcast address
Subnet mask: 255.255.255.**192**			
First subnet	192.168.162.**0**	192.168.162.**1** - 192.168.162.**62**	192.168.162.**63**
Second subnet	192.168.162.**64**	192.168.162.**65** - 192.168.162.**126**	192.168.162.**127**
Third subnet	192.168.162.**128**	192.168.162.**129** - 192.168.162.**190**	192.168.162.**191**
Fourth subnet	192.168.162.**192**	192.168.162.**193** - 192.168.162.**254**	192.168.162.**255**

<u>Part IV: All You Really Need</u>

Subnetting can seems pretty complicated, but here's all you really need in order to easily and rapidly subnet a network, a little chart we sometimes call the Happy Chart. You'll soon see why.

Mask	128	192	224	240	248	252	254	255
Bit	1	1	1	1	1	1	1	1
Bit value	128	64	32	16	8	4	2	1

Assume you own the 199.1.2.0 network. You need to create 16 subnets and you will need no more than 12 hosts on each subnet. Complete the following steps:

1. Find out how many bits you need to borrow by counting by powers of two (starting with 2^1) from the left-most bit until you reach the number of subnets you need. Draw an imaginary line to the right of the last bit you borrow. The subnet mask you need is the one to the left of the line. To make sure you have the right amount of hosts for each network, you can count by powers of two (starting with 2^1) from the right-most bit until you reach the number of hosts you need.

	2	**4**	**8**	**16**					
Mask	128	192	224	**240**		248	252	254	255
Bit	**1**	**1**	**1**	**1**		1	1	1	1
Bit value	128	64	32	16		8	4	2	1

2. Obtain the network addresses by starting with the 0 network, which is always the first (199.1.2.**0**) and adding the bit value that corresponds to the mask. For this example, that tells us the second network is 199.1.2.16. Continue to increment by this bit value to obtain all the network addresses.

199.1.2.**0**
199.1.2.**16**
199.1.2.**32**
199.1.2.**48**
199.1.2.**64**
199.1.2.**80**
199.1.2.**96 . . . and so on**

Mask	128	192	224	**240**		248	252	254	255
Bit	**1**	**1**	**1**	**1**		1	1	1	1
Bit value	128	64	32	**16**		8	4	2	1

3. Once you see the network addresses, it's easy to determine the broadcast addresses and the valid host addresses for each subnet. For example, the last address on the 199.1.2.0 network has to be 199.1.2.**15** because 199.1.2.16 is the next network address. Since it's the last address on the network, 199.1.2.**15** is the broadcast address. All addresses between 199.1.2.**0** and 199.1.2.**15** are the host addresses for the network.

Let's try another example. Assume you own the 200.20.2.0 network. You want to create 2 subnets, but you need up to 125 hosts on at least one of the subnets. Complete the following steps:

1. Find out how many bits you need to borrow by counting the bits by powers of two (starting with 2^1) from the left-most bit until you reach the maximum number of subnets you need. Draw an imaginary line to the right of the last bit you borrow. The subnet mask you need is the one to the left of the line.

	2								
Mask	**128**		192	224	240	248	252	254	255
Bit	**1**		1	1	1	1	1	1	1
Bit value	128		64	32	16	8	4	2	1

2. Obtain the network addresses by starting with the 0 network, which is always the first (200.20.2.**0**) and adding the bit value that corresponds to the mask. For this example, that tells us the second network (and last in this case) is 200.20.2.**128**. The reason it

has to be the last subnet is that you would get 200.20.2.256 as the next network address if you incremented again by 128. An octet cannot contain a number greater than 255.

200.20.2.**0**
200.20.2.**128**

Mask	**128**		192	224	240	248	252	254	255
Bit	1		1	1	1	1	1	1	1
Bit value	**128**		64	32	16	8	4	2	1

3. Once you see the network addresses, it's easy to determine the broadcast addresses and the valid host addresses for each subnet. For this example, the last address on the 200.20.2.**0** network has to be 200.20.2.**127** because 200.20.2.128 is the next network address. Since it's the last address on the network, 200.20.2.**127** is the broadcast address. All addresses between 200.20.2.**0** and 200.20.2.**127** are the host addresses for the network.

Let's do that example again with just a little modification. Again assume you own the 200.20.2.0 network. You want to subnet your network as much as possible, but you need up to **125** hosts on at least one of the subnets. Complete the following steps:

1. **Note: This is the only steps we will change. We are changing it to show how you can count by powers of two from right to left to ensure that you get the desired number of hosts.** .. . Find out how many bits you need to borrow by counting the bits by powers of two (starting with 2^1) from the right-most bit until you reach the maximum number of hosts you need on any subnet.. Draw an imaginary line to the left of the last bit you borrow. The subnet mask you need is still the one to the left of the line.

			128	**64**	**32**	**16**	**8**	**4**	**2**
Mask	**128**		192	224	240	248	252	254	255
Bit	**1**		1	1	1	1	1	1	1
Bit value	128		64	32	16	8	4	2	1

2. Obtain the network addresses by starting with the 0 network, which is always the first (200.20.2.**0**) and adding the bit value that corresponds to the mask. For this example, that tells us the second network (and last in this case) is 200.20.2.**128**. The reason it has to be the last subnet is that you would get 200.20.2.256 as the next network address if you incremented again by 128. An octet cannot contain a number greater than 255.

200.20.2.**0**
200.20.2.**128**

Mask	**128**		192	224	240	248	252	254	255
Bit	1		1	1	1	1	1	1	1
Bit value	**128**		64	32	16	8	4	2	1

3. Once you see the network addresses, it's easy to determine the broadcast addresses and the valid host addresses for each subnet. For this example, the last address on the 200.20.2.**0** network has to be 200.20.2.**127** because 200.20.2.128 is the next network address. Since it's the last address on the network, 200.20.2.**127** is the broadcast address. All addresses between 200.20.2.**0** and 200.20.2.**127** are the host addresses for the network.

Now you try it! (Answers found at the end of this paper)

Here are some examples for you to try. Use the Happy Chart to help you.

Example 1

XYZ Company would like to subnet its network so that there are five separate subnets. They will need 25 computers in each subnet. Complete each of the following:

NOTE: If you create more than five subnets, list the extra ones too.

Subnet	Network address	Host addresses	Broadcast address
Subnet mask: 255.255.255._____			
First subnet	192.168.162.___	192.168.162.___ - 192.168.162.___	192.168.162.___
Second subnet	192.168.162.___	192.168.162.___ - 192.168.162.___	192.168.162.___
Third subnet	192.168.162.___	192.168.162.___ - 192.168.162.___	192.168.162.___
Fourth subnet	192.168.162.___	192.168.162.___ - 192.168.162.___	192.168.162.___
Fifth subnet	192.168.162.___	192.168.162.___ - 192.168.162.___	192.168.162.___
Sixth subnet ?			
?			

Note: The term "Class C" is used in this document to help facilitate the understanding of IP addressing and subnetting. IP address class terminology is rarely used in the industry anymore because of the introduction of classless interdomain routing (CIDR).

Example 2
Company ABC would like to subnet its network (219.7.9.0) so that there are 32 separate subnets. They will need 6 hosts in each subnet. Complete the following table:

NOTE: Because there are so many subnets, don't write them all out (unless you just want to). If you can do the first ten and know what the last one is, you get the idea.

Subnet	Network address	Host addresses	Broadcast address
Subnet mask: 255.255.255.___			
First subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Second subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Third subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Fourth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Fifth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Sixth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Seventh subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Eighth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Ninth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
Tenth subnet	219.7.9. __	219.7.9. __ - 219.7.9. __	219.7.9. __
.			
.			
Thirty-second subnet	219.7.9. ___	219.7.9. ___ - 219.7.9. ___	219.7.9. ___

Example 3
The Acme Company would like to subnet its network (195.5.5.0) so that there are 50 separate subnets. They will need only 2 hosts in each subnet. Complete each of the following:

NOTE: Because there are so many subnets, you don't need to write them all out. If you can fill in the information required below (the subnet mask, the addresses for the first few subnets, and the total number of subnets created), you obviously get the idea.

Subnet	Network address	Host addresses	Broadcast address
Subnet mask: 255.255.255. ___			
First subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Second subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Third subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Fourth subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Fifth subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Sixth subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
Seventh subnet	195.5.5. __	195.5.5. __ - 195.5.5. __	195.5.5. __
.			
.			

How many subnets are actually created with this subnet mask you used? __

Table 1-7 Variable-length Subnet IDs

Subnet Number	Subnet ID (Binary)	Subnet Mask	Hosts per Subnet	Example Subnet Address
1	0	255.255.255.128	126	208.147.66.0/25
2	10	255.255.255.192	62	208.147.66.128/26
3	110	255.255.255.224	30	208.147.66.192/27
4	1110	255.255.255.240	14	208.147.66.224/28
5	11110	255.255.255.248	6	208.147.66.240/29
6	111110	255.255.255.252	2	208.147.66.248/30
7	111111	255.255.255.252	2	208.147.66.252/30

Maximizing Available Address Space

In Table 1-7, notice that the seventh and final subnet listed is the same size as the sixth and is distinguished by an all-1s subnet ID instead of by the trailing 0 used with the other subnet IDs. As an alternative to using the maximum seven subnets presented, you could define the all-1s subnet ID at any level in the table to replace all the subnets listed below that subnet. For example, you could define a subnet ID of 11 to replace subnets 3 through 7 listed in the table.

Exam Tip Just about everyone considers VLSMs confusing. If you see a question on VLSMs on the 70-642 exam, and you very well might, it will probably be the toughest question you will face on the whole test. To handle such questions, first try to eliminate incorrect answer choices whose subnet masks do not match the appropriate incremental pattern. Then, try to eliminate answer choices whose address ranges do not properly correspond to the pattern of 1s with a single trailing 0. You might need to perform decimal-to-binary conversions to get the answer correct. Most of all, though, make sure you don't spend too much time on a VLSM question. Eliminate what you can, and if you don't have an answer within 3 minutes or so, take your best guess and move on.

PRACTICE Learning to Work with Address Blocks

In this practice, you perform exercises that help solidify your understanding of address blocks, subnet masks, and host capacity.

◆ Exercise 1 Choosing an Appropriate Subnet Mask

You are adding a new server to each of the following subnets. Given the addresses of the existing computers on that subnet, determine which subnet mask you should assign the new server.

1. Which subnet mask would you assign to the new server?

Subnet 1:Existing Computers
10.2.12.1
10.2.41.23
10.2.41.100
10.2.41.101

Answer Choices:

A. 255.0.0.0 (/8)
B. 255.255.0.0 (/16)
C. 255.255.255.0 (/24)

[redacted]

2. Which subnet mask would you assign to the new server?

Subnet 2: Existing Computers
192.168.34.1
192.168.34.55
192.168.34.223
192.168.34.5

Answer Choices:

A. 255.0.0.0 (/8)
B. 255.255.0.0 (/16)
C. 255.255.255.0 (/24)

[redacted]

◆ Exercise 2 Converting Subnet Masks to Dotted-Decimal Notation

Convert the following subnet masks in slash notation to dotted-decimal by using your familiarity with the /16 subnet mask, the /24 subnet mask, and the nine possible subnet mask octet values. Write the final answer in each space provided.

Slash Notation	Dotted-decimal
/18	
/28	
/21	
/30	

Slash Notation	Dotted-decimal
/19	
/26	
/22	
/27	
/17	
/20	
/29	
/23	
/25	

Answer:

Slash Notation	Dotted-decimal
/18	[redacted]
/28	
/21	
/30	
/19	
/26	
/22	
/27	
/17	
/20	
/29	
/23	
/25	

◆ Exercise 3 Converting Subnet Masks to Slash Notation

Using your familiarity with 255.255.0.0, 255.255.255.0, and with the nine possible values in a subnet mask octet, convert the following subnet masks in dotted-decimal notation to slash notation. Write the final answer in each space provided.

Dotted-decimal	Slash Notation
255.255.240.0	
255.255.255.248	

Dotted-decimal	Slash Notation
255.255.192.0	
255.255.255.128	
255.255.248.0	
255.255.255.224	
255.255.252.0	
255.255.128.0	
255.255.255.252	
255.255.224.0	
255.255.254.0	
255.255.255.192	
255.255.255.240	

Answer:

Dotted-decimal	Slash Notation
255.255.240.0	[redacted]
255.255.255.248	[redacted]
255.255.192.0	[redacted]
255.255.255.128	[redacted]
255.255.248.0	[redacted]
255.255.255.224	[redacted]
255.255.252.0	[redacted]
255.255.128.0	[redacted]
255.255.255.252	[redacted]
255.255.224.0	[redacted]
255.255.254.0	[redacted]
255.255.255.192	[redacted]
255.255.255.240	[redacted]

◆ Exercise 4 Determining the Host Capacity of Networks

For each of the given address blocks below, determine the number of hosts that can be supported. Use either the halving-and-doubling or subtract-from-256 technique, as appropriate. Write down the answer in the space provided in the right column. (Hint: remember to subtract two from the total number of addresses to determine the number of supported hosts.)

Address Block	Number of Supported Hosts
131.107.16.0/20	
10.10.128.0 Subnet mask: 255.255.254.0	
206.73.118.0/26	
192.168.23.64 Subnet mask: 255.255.255.224	
131.107.0.0 Subnet mask: 255.255.255.0	
206.73.118.24/29	
10.4.32.0/21	
172.16.12.0/22	
192.168.1.32 Subnet mask: 255.255.255.128	
131.107.100.48/28	
206.73.118.12 Subnet mask: 255.255.255.252	
10.12.200.128/25	
192.168.0.0 Subnet mask: 255.255.248.0	
172.20.43.0/24	
131.107.32.0 Subnet mask 255.255.255.240	
10.200.48.0 Subnet mask: 255.255.240.0	
192.168.244.0/23	
10.0.0.0 /30	
172.31.3.24 Subnet mask: 255.255.255.248	
206.73.118.32/27	
131.107.8.0 Subnet mask: 255.255.252.0	
192.168.0.64 Subnet mask: 255.255.255.192	

Answer:

Address Block	Number of Supported Hosts
131.107.16.0/20	[redacted]
10.10.128.0 Subnet mask: 255.255.254.0	[redacted]
206.73.118.0/26	[redacted]
192.168.23.64 Subnet mask: 255.255.255.224	[redacted]
131.107.0.0 Subnet mask: 255.255.255.0	[redacted]
206.73.118.24/29	[redacted]
10.4.32.0/21	[redacted]
172.16.12.0/22	[redacted]
192.168.1.32 Subnet mask: 255.255.255.128	[redacted]
131.107.100.48/28	[redacted]
206.73.118.12 Subnet mask: 255.255.255.252	[redacted]
10.12.200.128/25	[redacted]
192.168.0.0 Subnet mask: 255.255.248.0	[redacted]
172.20.43.0/24	[redacted]
131.107.32.0 Subnet mask 255.255.255.240	[redacted]
10.200.48.0 Subnet mask: 255.255.240.0	[redacted]
192.168.244.0/23	[redacted]
10.0.0.0 /30	[redacted]
172.31.3.24 Subnet mask: 255.255.255.248	[redacted]
206.73.118.32/27	[redacted]
131.107.8.0 Subnet mask: 255.255.252.0	[redacted]
192.168.0.64 Subnet mask: 255.255.255.192	[redacted]

◆ Exercise 5 Determining Network Size Requirements in Slash Notation Terms

Each of the values in the left column of the table below refers to a number of computers that a given network must support. In the corresponding space in the right column, specify with a subnet mask in slash notation the smallest network address size that will accommodate those computers.

The first row is provided as an example.

(Hint: remember to add two to the number of hosts in order to determine the number of addresses needed.)

Number of Network Hosts	Subnet Mask (/n)
18	/27
125	
400	
127	
650	
7	
2000	
4	
3500	
20	
32	

Answer:

Number of Network Hosts	Subnet Mask (/n)
125	[redacted]
400	[redacted]
127	[redacted]
650	[redacted]
7	[redacted]
2000	[redacted]
4	[redacted]
3500	[redacted]
20	[redacted]
32	[redacted]

◆ Exercise 6 Determining Network Size Requirements in Terms of a Dotted-Decimal Subnet Mask

Each of the values in the left column of the table below refers to a number of computers that a given network must support. In the corresponding space in the right column, specify with a subnet mask in dotted-decimal notation the smallest network size that will accommodate those computers.

The first row is provided as an example.

(Hint: remember to add two to the number of hosts in order to determine the number of addresses needed. Then, use the halving-and-doubling or subtract-from-256 technique.)

Number of Network Hosts	Subnet Mask (w.x.y.z)
100	255.255.255.128
63	
1022	
6	
1100	
12	
150	
2500	
20	
300	
35	

Answer:

Number of Network Hosts	Subnet Mask (w.x.y.z)
63	[redacted]
1022	[redacted]
6	[redacted]
1100	[redacted]
12	[redacted]
150	[redacted]
2500	[redacted]
20	[redacted]

SUBNETTING PRACTICE EXERCISES

By Pang-Chieh Chou, http://www.hal-pc.org/~pchou

As a former teacher, I believe in the value of practice exercises to apply the concepts. Answers to the exercises are at the end of each section.

I. Conversion between binary and decimal forms of IP addresses

a. Convert the binary IP address into decimal form.
b. Identify the IP address as Class A, Class B, or Class C.

1. 01011011 . 11111110 . 00000000 . 01010011
2. 11110001 . 10101010 . 00100000 . 11010000
3. 10010001 . 00000100 . 11001111 . 11111100

[redacted]

In the following exercises, you may use the Windows calculator, which can be accessed as follows: Start > Programs > Accessories > Calculator. Then set the View menu to Scientific. Notice that if you enter a decimal number and then click Bin, the binary form is displayed. Clicking back on Dec, the decimal form is displayed.

a. Convert the IP addresses from decimal to binary form.
b. Classify the IP address as Class A, Class B, or Class C.

4. 20.54.255.1
5. 220.255.100.64
6. 160.1.254.130

Answers:

[redacted]

II. Determining Whether Two IP Addresses in Binary Form are on the Same Subnet

In each of the following exercises, you are to use the subnet mask to determine whether the two IP addresses belong to the same subnet or different subnets.

7)

01100110 . 10110010 . 00011100 . 00000001 IP Address 1

01100110 . 10100010 . 00011100 . 00000011 IP Address 2

11111111 . 11100000 . 00000000 . 00000000 Subnet Mask 1

8)

01100110 . 10110010 . 00011100 . 00000001 IP Address 1

01100110 . 10100010 . 00011100 . 00000011 IP Address 2

11111111 . 00000000 . 00000000 . 00000000 Subnet Mask 2

9)

01100110 . 10110010 . 00011100 . 00000001 IP Address 1

01100110 . 10100010 . 00011100 . 00000011 IP Address 2

11111111 . 11111110 . 00000000 . 00000000 Subnet Mask 3

10)

11001000 . 01000100 . 10010100 . 10110100 IP Address 3

11001000 . 01000100 . 10010100 . 10101000 IP Address 4

11111111 . 11111111 . 11111111 . 11100000 Subnet Mask 4

11)

11001000 . 01000100 . 10010100 . 10110100 IP Address 3

11001000 . 01000100 . 10010100 . 01100010 IP Address 5

11111111 . 11111111 . 11111111 . 11100000 Subnet Mask 4

[redacted]

III. Determining the appropriate subnet mask based on Class address and number of subnets

In the following exercises, you may use the chart on the last page of the subnetting tutorial. You are to answer the following question for each:

You have a Class ______ IP network ID. You wish to have _____ subnets. What should your subnet mask be if you wish to have the maximum possible number of hosts on each subnet ?

12. A ; 50
13. A ; 12
14. B ; 200
15. B ; 4
16. C ; 20
17. C ; 9

Answers:

[redacted]

IV. Finding the range of IP addresses for a subnet

For each of exercise, find the ranges of IP addresses for all subnets.

18. Network: 53.x.y.z ; Subnet Mask: 255.240.0.0
19. Network: 26.x.y.z ; Subnet Mask: 255.224.0.0
20. Network: 147.10.y.z ; Subnet Mask: 255.255.248.0
21. Network: 132.0.y.z ; Subnet Mask: 255.255.192.0
22. Network: 201.100.56.z ; Subnet Mask: 255.255.255.224

[redacted]

[redacted]

V. Determining whether two IP addresses in decimal form belong to the same subnet based on subnet mask.

Determine whether the two IP addresses belong to the same subnet or different subnets.

23)

136. 15. 40. 12 IP address 1

136. 15. 30. 4 IP address 2

255.255.248.0 Subnet Mask

24)

110. 100. 50. 4 IP address 1

110. 118. 50. 3 IP address 2

255.240.0.0 Subnet Mask

25)

196. 26. 100. 45 IP address 1

196. 26. 100. 35 IP address 2

255.255.255.224 Subnet Mask

[redacted]

Back to Subnetting Simplified Tutorial