Answers to Appendix C Extra Practice Problems

This document contains the answers and explanations for the problems posed in Appendix C, “Extra Practice,” of the book. The problems in Appendix C are organized based on the chapters in the book that explain the processes and math formulas used to solve the problems. This document uses that same organization, for easier reference.

Answers to Chapter 1 Practice Problems

Like most types of math problems, practicing the process can help. This section lists several practice problems for converting numbers between different formats. The four types of problems related to Chapter 1, and posed in this appendix, are as follows:

- Converting an 8-bit binary number to a decimal number
- Converting a decimal number to an 8-bit binary number
- Converting an 8-bit binary number to a hexadecimal number
- Converting a hexadecimal number to an 8-bit binary number

Answers to Converting an 8-bit Binary Number to a Decimal Number

Table W-1 lists the answers to the binary-to-decimal conversion problems listed in Table C-2 of Appendix C of the book. Following that, the answers are explained in more detail.

Table W-1 Answers to Binary-to-Decimal Conversion Problems

Problem Number	Binary	Decimal Value
1	01010101	85
2	10101010	170
3	11100011	227
4	11110000	240
5	00001111	15
6	11000000	192

The process for converting from an 8-bit binary number to decimal, as covered in Chapter 1 of the book, is relatively straightforward. It is repeated here:

How To

Step 1 Write the powers of 2, in decimal, in the top row of a table, similar to Table W-2.

Step 2 On the second line, write the binary number that is to be converted, lining up each binary digit under the powers of 2.

Step 3 Multiply each pair of numbers (the numbers in the same column) together.

Step 4 Add the eight products from Step 3.

Table W-2 shows the details of the process for Problem 1.

Table W-2 Answer to Problem 1: Converting 01010101 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	0	1	0	1	0	1	0	1
Product of two numbers in same column	**0**	**64**	**0**	**16**	**0**	**4**	**0**	**1**
Sum of all products	85							

The first step in the four-step conversion process is always to write the powers of 2, as shown in the top row of Table W-2. Then, in Step 2, the binary number is recorded in the second row. At that point, the last two steps are simple arithmetic: you multiply the two numbers in each column together (Step 3) and then add those eight numbers together (Step 4). In this case, the answer is 85.

Tables W-3 through W-7 show the details of the conversion for binary-to-decimal Problems 2 through 6.

Table W-3 Answer to Problem 2: Converting 10101010 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	1	0	1	0	1	0	1	0
Product of two numbers in same column	**128**	**0**	**32**	**0**	**8**	**0**	**2**	**0**
Sum of all products	170							

Table W-4 Answer to Problem 3: Converting 11100011 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	1	1	1	0	0	0	1	1
Product of two numbers in same column	**128**	**64**	**32**	**0**	**0**	**0**	**2**	**1**
Sum of all products	227							

Table W-5 Answer to Problem 4: Converting 11110000 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	1	1	1	1	0	0	0	0
Product of two numbers in same column	**128**	**64**	**32**	**16**	**0**	**0**	**0**	**0**
Sum of all products	240							

Table W-6 Answer to Problem 5: Converting 00001111 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	0	0	0	0	1	1	1	1
Product of two numbers in same column	**0**	**0**	**0**	**0**	**8**	**4**	**2**	**1**
Sum of all products	15							

Table W-7 Answer to Problem 6: Converting 11000000 to Decimal

Value Associated with That Digit or Column	128	64	32	16	8	4	2	1
The number itself	1	1	0	0	0	0	0	0
Product of two numbers in same column	**128**	**64**	**0**	**0**	**0**	**0**	**0**	**0**
Sum of all products	192							

Answers to Converting a Decimal Number to an 8-bit Binary Number

Table W-8 lists the answers to the decimal-to-binary conversion problems listed in Table C-3 of Appendix C of the book. Following that, the answers are explained in more detail.

Table W-8 Answers to Decimal-to-Binary Conversion Problems

Problem Number	Decimal Value	Binary
1	255	11111111
2	128	10000000
3	200	11001000
4	9	00001001
5	100	01100100
6	248	11111000

Chapter 1 of the book covers the process of converting from decimal to 8-bit binary numbers. The process listed in Chapter 1 is repeated here for reference, after which the answers are provided.

Begin by writing down eight powers of 2, starting with 128 on the left, and ending with 1 on the right, similar to Table W-9. The blank lines below the powers of 2 are placeholders in which you will record the binary digits.

Table W-9 Table for Converting Decimal to 8-bit Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	___	___	___	___	___	___	___	___

Beginning with 128 and moving toward 1, follow these steps eight times, once for each power of 2:

Step 1 If the decimal number is greater than or equal to the power of 2, do the following:

How To

A. Record a 1 as the binary digit under the power of 2.

B. Subtract the power of 2 from the decimal number, which results in a number called the "remainder."

C. Use the remainder for the next step/power of 2.

Step 2 If the decimal number is less than the power of 2, do the following:

A. Record a 0 as the binary digit under the power of 2.

B. Move to the next power of 2 and use the same remainder as in this step.

Answer to Problem 1: Converting 255 to Binary

Table W-10 shows the results of the preceding process for the first problem, which is to convert decimal 255 to 8-bit binary. 255 is a popular number with IP addressing, so many people simply memorize that 255 is 11111111 in binary.

Table W-10 Answer to Problem 1: Converting 255 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_1_	_1_	_1_	_1_	_1_	_1_	_1_	_1_

The decision points at each step are as follows:

Step 1 The decimal number (255) is greater than 128, so a 1 was recorded in the 128 column, and 255 – 128 = 127.

Step 2 The remaining decimal number (127) is greater than or equal to 64, so a 1 was recorded in the 64 column, and 127 – 64 = 63.

Step 3 The remaining decimal number (63) is greater than or equal to 32, so a 1 was recorded in the 32 column, and 63 – 32 = 31.

Step 4 The remaining decimal number (31) is greater than or equal to 16, so a 1 was recorded in the 16 column, and 31 – 16 = 15.

Step 5 The remaining decimal number (15) is greater than or equal to 8, so a 1 was recorded in the 8 column, and 15 – 8 = 7.

Step 6 The remaining decimal number (7) is greater than or equal to 4, so a 1 was recorded in the 4 column, and 7 – 4 = 3.

Step 7 The remaining decimal number (3) is greater than or equal to 2, so a 1 was recorded in the 2 column, and 3 – 2 = 1.

Step 8 The remaining decimal number (1) is greater than or equal to 1, so a 1 was recorded in the 1 column. (The final bit of math, 1 – 1 = 0, isn't really needed at this point.)

Answer to Problem 2: Converting 128 to Binary

Table W-11 shows the answer for decimal-to-binary conversion Problem 2.

Table W-11 Answer to Problem 2: Converting 128 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_1_	_0_	_0_	_0_	_0_	_0_	_0_	_0_

The decision points at each step are as follows:

Step 1 The decimal number (128) is greater than or equal to 128, so a 1 was recorded in the 128 column, and 128 – 128 = 0.

Step 2 The remaining decimal number (0) is not greater than or equal to 64, so a 0 was recorded in the 64 column.

Step 3 The remaining decimal number (0) is not greater than or equal to 32, so a 0 was recorded in the 32 column.

Step 4 The remaining decimal number (0) is not greater than or equal to 16, so a 0 was recorded in the 16 column.

Step 5 The remaining decimal number (0) is not greater than or equal to 8, so a 0 was recorded in the 8 column.

Step 6 The remaining decimal number (0) is not greater than or equal to 4, so a 0 was recorded in the 4 column.

Step 7 The remaining decimal number (0) is not greater than or equal to 2, so a 0 was recorded in the 2 column.

Step 8 The remaining decimal number (0) is not greater than or equal to 1, so a 0 was recorded in the 1 column.

Answer to Problem 3: Converting 200 to Binary

Table W-12 shows the answer for decimal-to-binary conversion Problem 3.

Table W-12 Answer to Problem 3: Converting 200 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_1_	_1_	_0_	_0_	_1_	_0_	_0_	_0_

The decision points at each step are as follows:

Step 1 The decimal number (200) is greater than or equal to 128, so a 1 was recorded in the 128 column, and 200 – 128 = 72.

Step 2 The remaining decimal number (72) is greater than or equal to 64, so a 1 was recorded in the 64 column, and 72 – 64 = 8.

Step 3 The remaining decimal number (8) is not greater than or equal to 32, so a 0 was recorded in the 32 column.

Step 4 The remaining decimal number (8) is not greater than or equal to 16, so a 0 was recorded in the 16 column.

Step 5 The remaining decimal number (8) is greater than or equal to 8, so a 1 was recorded in the 8 column, and 8 – 8 = 0.

Step 6 The remaining decimal number (0) is not greater than or equal to 4, so a 0 was recorded in the 4 column.

Step 7 The remaining decimal number (0) is not greater than or equal to 2, so a 0 was recorded in the 2 column.

Step 8 The remaining decimal number (0) is not greater than or equal to 1, so a 0 was recorded in the 1 column.

Answer to Problem 4: Converting 9 to Binary

Table W-13 shows the answer for decimal-to-binary conversion Problem 4.

Table W-13 Answer to Problem 4: Converting 9 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_0_	_0_	_0_	_0_	_1_	_0_	_0_	_1_

The decision points at each step are as follows:

Step 1 The decimal number (9) is not greater than or equal to 128, so a 0 was recorded in the 128 column.

Step 2 The remaining decimal number (9) is not greater than or equal to 64, so a 0 was recorded in the 64 column.

Step 3 The remaining decimal number (9) is not greater than or equal to 32, so a 0 was recorded in the 32 column.

Step 4 The remaining decimal number (9) is not greater than or equal to 16, so a 0 was recorded in the 16 column.

Step 5 The remaining decimal number (9) is greater than or equal to 8, so a 1 was recorded in the 8 column, and 9 – 8 = 1.

Step 6 The remaining decimal number (1) is not greater than or equal to 4, so a 0 was recorded in the 4 column.

Step 7 The remaining decimal number (1) is not greater than or equal to 2, so a 0 was recorded in the 2 column.

Step 8 The remaining decimal number (1) is greater than or equal to 1, so a 1 was recorded in the 1 column. (The final bit of math, 1 – 1 = 0, isn't really needed at this point.)

Answer to Problem 5: Converting 100 to Binary

Table W-14 shows the answer for decimal-to-binary conversion Problem 5.

Table W-14 Answer to Problem 5: Converting 100 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_0_	_1_	_1_	_0_	_0_	_1_	_0_	_0_

The decision points at each step are as follows:

Step 1 The decimal number (100) is not greater than or equal to 128, so a 0 was recorded in the 128 column.

Step 2 The remaining decimal number (100) is greater than or equal to 64, so a 1 was recorded in the 64 column, and 100 – 64 = 36.

Step 3 The remaining decimal number (36) is greater than or equal to 32, so a 1 was recorded in the 32 column, and 36 – 32 = 4.

Step 4 The remaining decimal number (4) is not greater than or equal to 16, so a 0 was recorded in the 16 column.

Step 5 The remaining decimal number (4) is not greater than or equal to 8, so a 0 was recorded in the 8 column.

Step 6 The remaining decimal number (4) is greater than or equal to 4, so a 1 was recorded in the 4 column, and 4 – 4 = 0.

Step 7 The remaining decimal number (0) is not greater than or equal to 2, so a 0 was recorded in the 2 column.

Step 8 The remaining decimal number (0) is not greater than or equal to 1, so a 0 was recorded in the 1 column.

Answer to Problem 6: Converting 248 to Binary

Table W-15 shows the answer for decimal-to-binary conversion Problem 6.

Table W-15 Answer to Problem 6: Converting 248 to Binary

Power of 2	128	64	32	16	8	4	2	1
Binary Digits	_1_	_1_	_1_	_1_	_1_	_0_	_0_	_0_

The decision points at each step are as follows:

Step 1 The decimal number (248) is greater than 128, so a 1 was recorded in the 128 column, and 248 – 128 = 120.

Step 2 The remaining decimal number (120) is greater than or equal to 64, so a 1 was recorded in the 64 column, and 120 – 64 = 56.

Step 3 The remaining decimal number (56) is greater than or equal to 32, so a 1 was recorded in the 32 column, and 56 – 32 = 24.

Step 4 The remaining decimal number (24) is greater than or equal to 16, so a 1 was recorded in the 16 column, and 24 – 16 = 8.

Step 5 The remaining decimal number (8) is greater than or equal to 8, so a 1 was recorded in the 8 column, and 8 – 8 = 0.

Step 6 The remaining decimal number (0) is not greater than or equal to 4, so a 0 was recorded in the 4 column.

Step 7 The remaining decimal number (0) is not greater than or equal to 2, so a 0 was recorded in the 2 column.

Step 8 The remaining decimal number (0) is not greater than or equal to 1, so a 0 was recorded in the 1 column.

Answers to Converting an 8-bit Binary Number to a Hexadecimal Number

Table W-16 lists the answers to the binary-to-hexadecimal conversion problems listed in Table C-4 of Appendix C of the book. Following that, the answers are explained in more detail.

Table W-16 Answers to Binary-to-Hexadecimal Conversion Problems

Problem Number	8-bit Binary Number	2-digit Hex Number
1	01010101	55
2	10101010	AA
3	11100011	E3
4	11110000	F0
5	00001111	0F
6	11000000	C0

The process to convert between binary and hex, and vice versa, is relatively simple if you use a conversion chart, as presented in Chapter 1. Table W-17 lists all 16 hexadecimal digits and the 4-digit binary numbers they represent.

Table W-17 Binary and Hexadecimal Conversion Chart

Hexadecimal Digit	**Binary Equivalent**	**Decimal Equivalent**
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

To perform the conversion, take each set of 4 binary digits and look up the corresponding hexadecimal values. For example, 01010101 has two sets of 4 binary digits: 0101 and 0101. Each set of 4 bits corresponds to hex digit 5, so the hex value is 55. Table W-18 breaks down the logic used to find the answers to all six binary-to-hexadecimal conversion problems listed in Table C-4 of Appendix C.

Table W-18 Answers to Binary-to-Hexadecimal Conversion Problems

Problem Number	8-bit Binary Number	First Set of 4 Binary Digits	Hex Value	Second Set of 4 Binary Digits	Hex Value	Combined 2-digit Hex Value
1	01010101	0101	5	0101	5	55
2	10101010	1010	A	1010	A	AA
3	11100011	1110	E	0011	3	E3
4	11110000	1111	F	0000	0	F0
5	00001111	0000	0	1111	F	0F
6	11000000	1100	C	0000	0	C0

Answers to Converting a Hexadecimal Number to an 8-bit Binary Number

Table W-19 lists the answers to the hexadecimal-to-binary conversion problems listed in Table C-5 of Appendix C of the book. Following that, the answers are explained in more detail.

Table W-19 Answers to Hexadecimal-to-Binary Conversion Problems

Problem Number	2-digit Hex Number	8-bit Binary Number
1	F0	11110000
2	0E	00001110
3	AB	10101011
4	07	00000111
5	7A	01111010
6	AA	10101010

The process to convert between hex and binary is relatively simple if you use a conversion chart, as presented in Chapter 1. Table W-20 provides a chart with all 16 hexadecimal digits and the 4-digit binary numbers they represent.

Table W-20 Binary and Hexadecimal Conversion Chart

Hexadecimal Digit	Binary Equivalent	Decimal Equivalent
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

To perform the conversion, take each hexadecimal digit and substitute the 4-bit binary code from the conversion chart. For example, F0 has 2 hex digits. The F corresponds to binary 1111, and the 0 corresponds to binary 0000. Table W-21 breaks down the logic used to find the answers to all six hexadecimal-to-binary conversion problems listed in Appendix C.

Table W-21 Answers to Hexadecimal-to-Binary Conversion Problems

Problem Number	Hex Value	First Hex Digit	Binary Value	Second Hex Digit	Binary Value	Combined 8-Digit Binary Value
1	F0	F	1111	0	0000	11110000
2	0E	0	0000	E	1110	00001110
3	AB	A	1010	B	1011	10101011
4	07	0	0000	7	0111	00000111
5	7A	7	0111	A	1010	01111010
6	AA	A	1010	A	1010	10101010

Answers to Chapter 2 Practice Problems

You can determine the answers to the data transfer time problems listed in Appendix C in the book with some simple math. Setting up the problem is the only tricky part. To set up the problem, you need to do the following:

- Find the constraining link's bandwidth. Figure C-1 in Appendix C shows three WAN links, with one of those links being the slowest (constraining) link bandwidth in each problem.
- Convert the file sizes from megabytes to megabits. The problems list the file sizes in megabytes, and the formula T = S/BW requires the unit of bits instead of bytes for the file size (S).

Table W-22 shows the answers to the three data transfer time problems, pointing out the constraining link, that link's bandwidth, the file size in bits, and the result of the T = S/BW calculation.

Table W-22 Answers to Data Transfer Time Problems—Theoretical

Problem Number	Constraining Link	Constraining Bandwidth (bps)	File Size (Bits)	Transfer Time (Seconds)
1	C	512,000	80,000,000	156.25
2	A or C	256,000	40,000,000	156.25
3	A	64,000	1,600,000,000	25,000

To find the estimated data transfer time using a link's effective throughput, you follow the same rationale as when applying the theoretical maximum. Although the logic works the same as with the previous calculation, in this case, the constraining link's bandwidth is now reduced based on the assumption that other traffic will be competing with this one particular file transfer. Table W-23 lists the answers.

Table W-23 Answers to Data Transfer Time Problems Using Estimated Throughput

Problem Number	Constraining Link	Constraining Bandwidth (bps)	Throughput	File Size (Bits)	Transfer Time (Seconds)
1	C	512,000	256,000	80,000,000	312.5
2	A or C	256,000	85,333	40,000,000	469
3	A	64,000	16,000	1,600,000,000	100,000

Answers to Chapter 8 Practice Problems

The key to determining the devices in a collision domain and a broadcast domain is to remember what devices break LANs into separate collision and broadcast domains, and which devices do not. Use the following as a guideline:

- Devices that use Layer 1 forwarding processes—repeaters and hubs—do not separate LANs into different collision domains or broadcast domains.
- Devices that use Layer 2 forwarding processes—bridges and switches—do separate LANs into different collision domains, but do not separate LANs into separate broadcast domains.
- Devices that use Layer 3 forwarding processes—routers—separate LANS into different collision domains and different broadcast domains.

As for determining which links support full duplex, when LANs are built solely with UTP cabling and fiber-optic cabling, full duplex can be used as long as the cable does not connect to a hub or repeater.

Figures W-1 and W-2 show the same network topologies as shown in Figures C-2 and C-3, respectively, in Appendix C. In this case, the collision domains are enclosed with a circle that uses a dotted line style, and the broadcast domains are enclosed by circles that use a solid line style. Also, all links that can use full duplex are shown simply as thicker lines.

Figure W-1 Answers to Problem 1 for Identifying Collision Domains, Broadcast Domains, and Full-Duplex Links

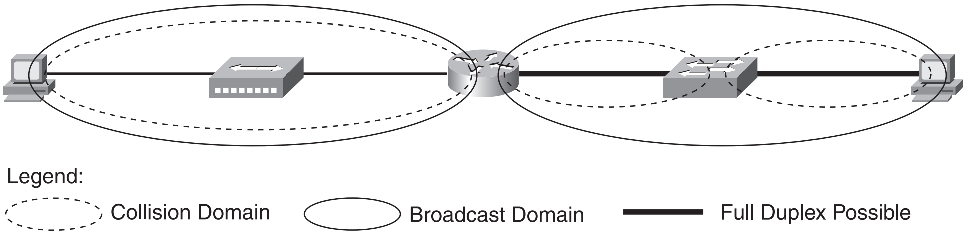
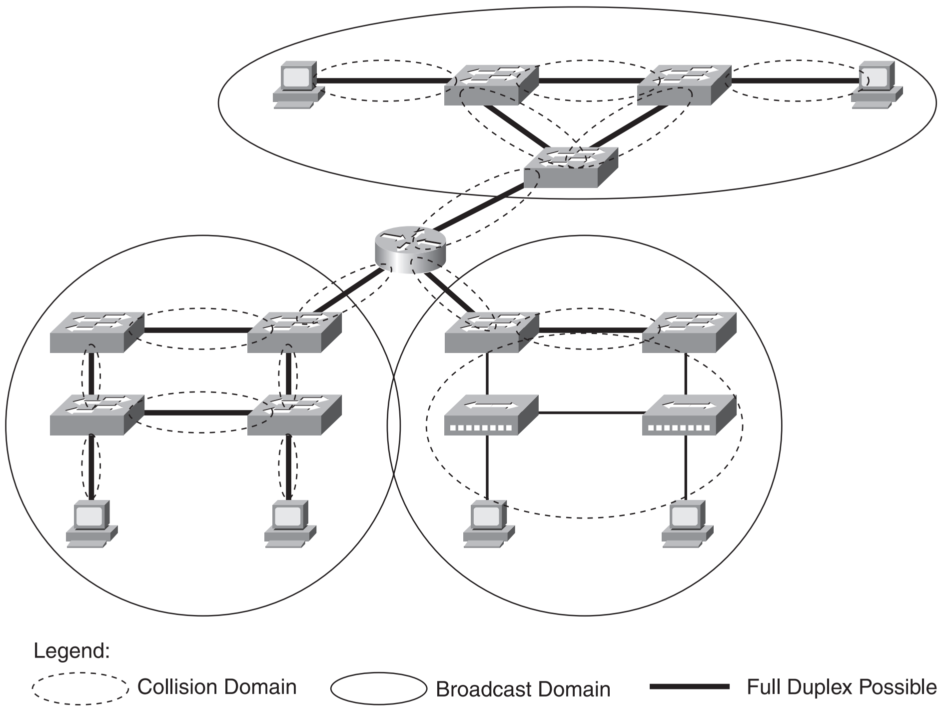


Figure W-2 Answers to Problem 2 for Identifying Collision Domains, Broadcast Domains, and Full-Duplex Links



Answers to Chapter 9 Practice Problems

Appendix C of the book includes three main types of practice problems related to Chapter 9:

- Dissecting the structure of an unsubnetted IP address
- Determining whether an IP address is a private address and whether it can legally be assigned to a host as an IP address, assuming no subnetting is used
- Dissecting the structure of a subnetted address that uses a simple mask

Answers to Dissecting Unsubnetted IP Addresses

Table W-24 lists the answers to the problems posed in Table C-8 of Appendix C in the book. Following Table W-24, the text explains the answers.

Table W-24 Answers to Dissecting Unsubnetted IP Addresses

Problem Number	IP Address	Class	Network Number	Number of Octets in the Network Part	Number of Octets in the Host Part
1	10.1.1.1	A	10.0.0.0	1	3
2	200.20.200.200	C	200.20.200.0	3	1
3	128.28.2.2	B	128.28.0.0	2	2
4	191.240.1.1	B	191.240.0.0	2	2
5	224.1.1.1	D	—	—	—
6	100.1.1.1	A	100.0.0.0	1	3
7	192.191.2.2	C	192.191.2.0	3	1
8	150.1.4.4	B	150.150.0.0	2	2

To dissect an unsubnetted IP address, you first need to be able to quickly determine whether an address is a Class A, B, or C address. When no subnetting is used, these addresses have a set size for the network and host parts of the address, as shown in Table W-25, and as covered in Chapter 9 in the book.

Table W-25 Classful IP Address Classes Based on the First Octet

Range of Values in First Octet (Inclusive)	Class	Length of Network Part	Length of Host Part
1–126	A	1 octet	3 octets
128–191	B	2 octets	2 octets
192–223	C	3 octets	1 octet

The only other fact needed to solve the problems in Table C-8 (repeated in this document, with answers, as Table W-24) is to remember that a network number has all decimal 0s in the host octets. For example, Problem 1 lists IP address 10.1.1.1. From the first octet's value (10), and Table W-25, it is clear that 10.1.1.1 is in a Class A network. By definition, then, with no subnetting, the address has a 1-octet network part and a 3-octet host part. The network number would be the same as the IP address (10.1.1.1), except that all the host octets would instead be 0s. In this case, the host octets are the last 3 octets, so the network number is 10.0.0.0.

The following list explains how the answers for Problems 2 through 7 were found:

Problem 2 IP address 200.20.200.200 is a Class C address, because the first octet (200) is between 192 and 223. That means it has 3 octets in the network part and 1 octet in the host part. So, the network number is the same as the IP address, except that the last octet is 0, making the network number 200.20.200.0.

Problem 3 IP address 128.28.2.2 is a Class B address, because the first octet (128) is between 128 and 191, inclusive. That means it has 2 octets in the network part, and 2 octets in the host part. So, the network number is the same as the IP address, except that the last 2 octets are 0, making the network number 128.28.0.0.

Problem 4 IP address 191.240.1.1 is a Class B address, because the first octet (191) is between 128 and 191, inclusive. That means it has 2 octets in the network part, and 2 octets in the host part. So, the network number is the same as the IP address, except that the last 2 octets are 0, making the network number 191.240.0.0.

Problem 5 IP address 224.1.1.1 is a Class D address, because the first octet (224) is between 224 and 239, inclusive. As such, this address does not reside in any specific Class A, B, or C network.

Problem 6 IP address 100.1.1.1 is a Class A address, because the first octet (100) is between 1 and 126, inclusive. That means it has 1 octet in the network part, and 3 octets in the host part. So, the network number is the same as the IP address, except that the last 3 octets are 0, making the network number 100.0.0.0.

Problem 7 IP address 192.191.2.2 is a Class C address, because the first octet (192) is between 192 and 223, inclusive. That means it has 3 octets in the network part, and 1 octet in the host part. So, the network number is the same as the IP address, except that the last octet is 0, making the network number 192.191.2.0.

Problem 8 IP address 150.1.4.4 is a Class B address, because the first octet (150) is between 128 and 191, inclusive. That means it has 2 octets in the network part, and 2 octets in the host part. So, the network number is the same as the IP address, except that the last 2 octets are 0, making the network number 150.1.0.0.

Answers to Determining Whether Addresses Are Private and Valid

Table W-26 lists the answers to the problems posed in Table C-9 of Appendix C in the book. Following Table W-26, the text explains the answers.

Table W-26 Answers to Determining Whether Addresses Are Valid and Private and Identifying Their Address Classes

Problem Number	IP Address	Class of Address	Assignable to an IP Host?	Part of a Private Network?
1	192.168.0.0	C	No	Yes
2	128.1.255.255	B	No	No
3	239.1.1.1	D	No	No

Table W-26 Answers to Determining Whether Addresses Are Valid and Private and Identifying Their Address Classes *(continued)*

Problem Number	IP Address	Class of Address	Assignable to an IP Host?	Part of a Private Network?
4	9.1.1.1	A	Yes	No
5	172.32.0.255	B	Yes	No
6	193.1.1.1	C	Yes	No
7	223.223.223.0	C	No	No
8	245.1.1.1	E	No	No

The problems listed in Table C-9 of Appendix C require that you think about the class of the address (A, B, C, D, or E), whether the address falls into an RFC 1918 private IP address range, and whether the address is reserved (meaning that the address cannot be used by a host as its IP address). Table W-27 lists the ranges of the first octets of IP addresses and the class implied by those values.

Table W-27 IPv4 Address Classes A Through E

Class	First Octet	Purpose
A	1–126	Unicast IP addresses
B	128–191	Unicast IP addresses
C	192–223	Unicast IP addresses
D	224–239	Multicast IP addresses
E	240–255	Experimental use

Table W-28 lists the range of private IP addresses per RFC 1918.

Table W-28 Private IP Networks per RFC 1918

Class	Range of Network Numbers	Total Number of Networks
A	10.0.0.0	1
B	172.16.0.0–172.31.0.0	16
C	192.168.0.0–192.168.255.0	256

To answer the questions in Table C-9 of Appendix C, you need the information in Tables W-27 and W-28 and the ability to recognize whether a dotted-decimal number is a network number or a broadcast address, neither of which can be assigned to a host to use as an IP address. The following two statements summarize how to recognize a network number or network broadcast address:

- Network numbers have the same value in the network octets as do all the assignable IP addresses in the network, but they have decimal 0s in the host octets
- Network broadcast addresses have the same value in the network octets as do all the assignable IP addresses in the network, but they have decimal 255s in the host octets

Armed with these facts, you can solve the problems. The following list outlines the logic used to solve each of the eight problems listed in Table C-9 of Appendix C:

Problem 1 192.168.0.0 is a Class C network number that is also in the range of private addresses listed in Table W-28. Being a network number, it cannot be assigned to a host to be used as that host's IP address.

Problem 2 128.1.255.255 is Class B network 128.1.0.0's network broadcast address, because it has 255s in the 2 host octets. The first octet (128) is not in the range of private addresses listed in Table W-28.

Problem 3 239.1.1.1 is a Class D multicast address, meaning that it cannot be assigned to any host. Also, it cannot possibly be a private address, because all private addresses are either Class A, B, or C addresses.

Problem 4 9.1.1.1 is a Class A address that is neither the network number (9.0.0.0) nor the network broadcast address (9.255.255.255), so it can be assigned for use by a host. The first octet (9) is not in the range of private addresses listed in Table W-28, so it is not a private address.

Problem 5 172.32.0.255 is an assignable address in Class B network 172.32.0.0. 172.32.0.0's network broadcast address is 172.32.255.255, with assignable addresses between 172.32.0.1 and 172.32.255.254, making 172.32.0.255 a valid IP address for use by a host. Also, 172.32 is just outside the range of Class B private addresses, as shown in Table W-28, so it is not a private address.

Problem 6 193.1.1.1 is a Class C address that is neither the network number (193.1.1.0) nor the network broadcast address (193.1.1.255), so it can be assigned for use by a host. The network (193.1.1.0) is not in the range of private addresses listed in Table W-28, so it is not a private address.

Problem 7 223.223.223.0 is a Class C network number, so it cannot be assigned to a host to be used as that host's IP address. It is not in the range of private addresses in Table W-28.

Problem 8 245.1.1.1 is a Class E experimental address, meaning that it cannot be assigned to any host. Also, it cannot possibly be a private address, because all private addresses are either Class A, B, or C addresses.

Answers to Dissecting a Subnetted Address That Uses a Simple Mask

Table W-29 shows the answers to the problems listed in Table C-10 of Appendix C. Following Table W-29, the text explains the answers.

Table W-29 Answers to Dissecting Subnetted IP Addresses

Problem Number	IP Address	Mask	Class	Subnet or Network Number	Number of Octets in the Network Part	Number of Octets in the Subnet Part	Number of Octets in the Host Part
1	10.1.1.1	255.255.0.0	A	10.1.0.0	1	1	2
2	200.20.200.200	255.255.255.0	C	200.20.200.0	3	0	1
3	128.28.2.2	255.255.255.0	B	128.28.2.0	2	1	1
4	191.240.1.1	255.255.255.0	B	191.240.1.0	2	1	1
5	99.1.1.1	255.255.255.0	A	99.1.1.0	1	2	1
6	100.1.1.1	255.255.0.0	A	100.1.0.0	1	1	2
7	192.1.2.2	255.255.255.0	C	192.1.2.0	3	0	1
8	150.1.4.4	255.255.255.0	B	150.1.4.0	2	1	1

The key to filling in the last three columns of Table W-28 is to determine the class of address (A, B, or C) and look at the number of 0s in the mask. With simple subnet masks—masks that have only decimal 255s and 0s in them—the number of decimal 0s in the mask defines the number of host octets at the end of the address. Also, the number of network octets is determined by the class, with Class A being 1 octet of network, Class B being 2 octets of network, and Class C being 3 octets of network. And, of course, the three parts of the address must add up to 4 octets.

For example, in Problem 1, 10.1.1.1 is a Class A address, meaning it has a 1-octet network part. The mask is 255.255.0.0, meaning the subnetting scheme implies that the last 2 octets are the host octets in this case. That leaves 1 octet for the subnet part of the address.

To find the subnet number, you write down the same number as the IP address, but substitute a 0 for the host octets. In the case of 10.1.1.1, mask 255.255.0.0, you copy down 10.1 and substitute 0s for the last 2 octets (the host octets), for a subnet number of 10.1.0.0.

The following list provides commentary on Problems 2 through 8 from Table W-28:

Problem 2 200.20.200.200 is a Class C address, so it has 3 network octets. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving no subnet octets. In effect, no subnetting is used for this problem. The network number would be 200.20.200.0, which is the same number as 200.20.200.200, but with the 1 host octet changed to be 0.

Problem 3 128.28.2.2 is a Class B address, so it has 2 network octets. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving 1 subnet octet. The subnet number would be 128.28.2.0, which is the same number as 128.28.2.2, but with the 1 host octet changed to be 0.

Problem 4 191.240.1.1 is a Class B address, so it has 2 network octets. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving 1 subnet octet. The subnet number would be 191.240.1.0, which is the same number as 191.240.1.1, but with the 1 host octet changed to be 0.

Problem 5 99.1.1.1 is a Class A address, so it has 1 network octet. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving 2 subnet octets. The subnet number would be 99.1.1.0, which is the same number as 99.1.1.1, but with the 1 host octet changed to be 0.

Problem 6 100.1.1.1 is a Class A address, so it has 1 network octet. With mask 255.255.0.0, it has 2 host octets (the last 2 octets), leaving 1 subnet octet. The subnet number would be 100.1.0.0, which is the same number as 100.1.1.1, but with the 2 host octets changed to be 0.

Problem 7 192.1.2.2 is a Class C address, so it has 3 network octets. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving no subnet octets. In effect, no subnetting is used for this problem. The network number would be 192.1.2.0, which is the same number as 192.1.2.2, but with the 1 host octet changed to be 0.

Problem 8 150.1.4.4 is a Class B address, so it has 2 network octets. With mask 255.255.255.0, it has 1 host octet (the last octet), leaving 1 subnet octet. The subnet number would be 150.1.4.0, which is the same number as 150.1.4.4, but with the 1 host octet changed to be 0.

Answers to Chapter 10 Practice Problems

Appendix C lists six different types of problems that can be solved based on Chapter 10 of the book. This section provides the answers to those problems. The types of problems are as follows:

- Determining the number of required subnets for an internetwork, based on a diagram of the internetwork
- Determining the minimum number of subnet and host bits required in the structure of an IP address to support the maximum number of required subnets and hosts per subnet
- Determining the subnet mask(s) that support the stated maximum number of subnets and hosts per subnet
- For a chosen IP network and mask, determining all the subnets of that network

- For a chosen IP network and mask, determining the subnet broadcast address for each subnet and the range of assignable addresses in each subnet
- For a given IP address and mask, determining the resident subnet

Answers to Determining the Number of Required Subnets, Based on a Network Diagram

The key to determining the number of required subnets is to remember the following two rules:

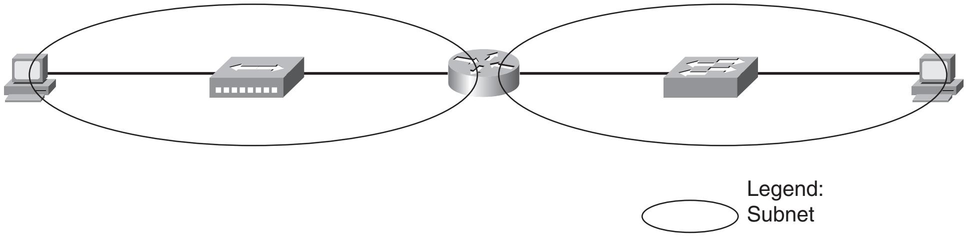
- On LANs, each broadcast domain needs one subnet.
- Each point-to-point WAN link also needs one subnet.

Although each of these rules has some exceptions, they are true for the level of IP and subnetting covered in all the semesters of the Cisco Networking Academy Program CCNA curriculum. Finding each point-to-point link in a diagram is easy, so the only potentially tricky part is to identify each LAN broadcast domain.

Figures W-3 through W-6 show the answers to the four problems posed in Figures C-4 through C-7 in Appendix C. These figures are the same as the figures in Appendix C, but in these figures each required subnet is circled.

Routers separate LANs into different broadcast domains, but hubs, switches, repeaters, and bridges do not. So, in Figure W-3, two different sets of LAN links are separated from each other by the one router, creating two broadcast domains, which in turn then requires two separate subnets.

Figure W-3 Answer to Problem 1 for Determining the Number of Required Subnets



Although Figure W-4 is much more detailed than Figure W-3, the same basic principles apply. The groups of four LAN switches do not create separate broadcast domains, but the one router does. Three groups of LAN devices and links are separated from the others by the one router. So, there are three broadcast domains and a need for three subnets.

Figure W-5 most closely matches a figure seen several times throughout the book, with one main site at the top and three remote sites at the bottom of the figure. So, four LAN broadcast domains exist, one at each site, requiring four subnets. Also, three WAN point-to-point links exist, requiring three more subnets, for a total of seven required subnets.

Figure W-4 Answer to Problem 2 for Determining the Number of Required Subnets

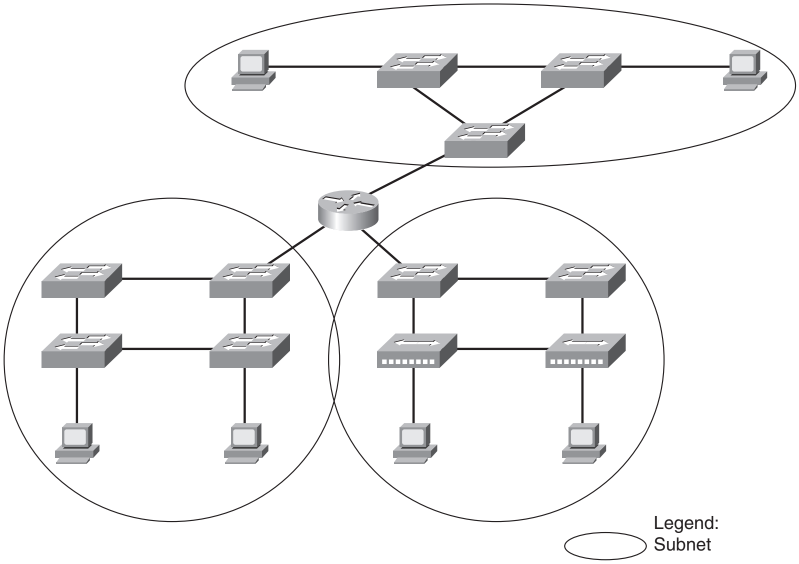


Figure W-5 Answer to Problem 3 for Determining the Number of Required Subnets

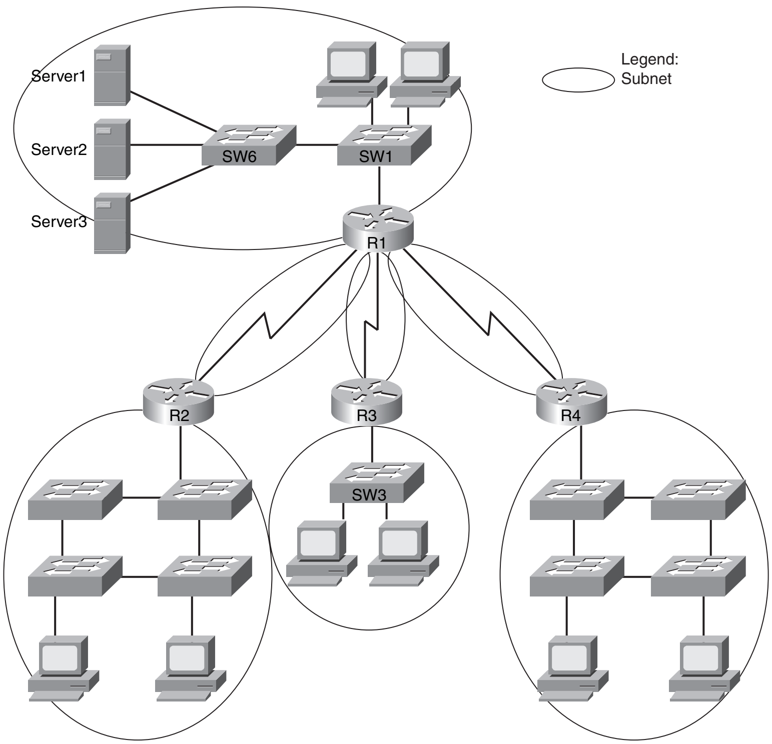
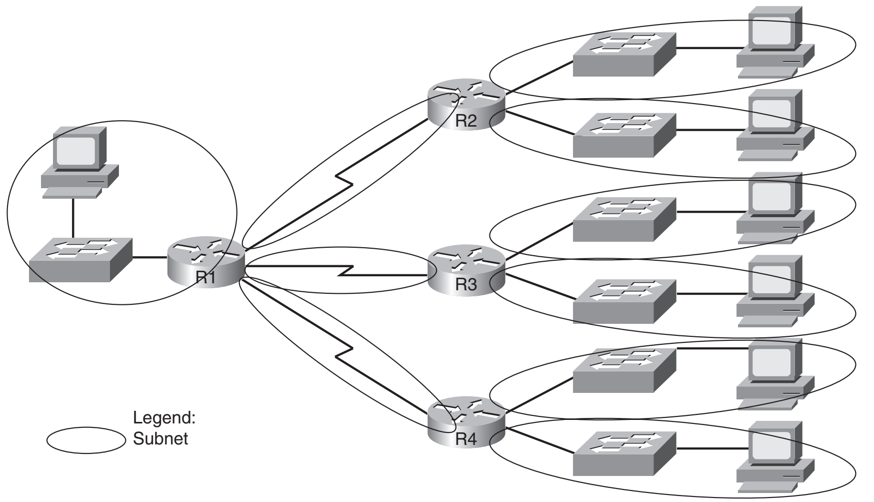


Figure W-6 has three routers with two LAN interfaces plus a WAN interface. R2, R3, and R4 each have two LAN interfaces connected to two different LAN switches. So, each of these three routers connects to two LAN broadcast domains, requiring six subnets. R1 on the left connects to a single LAN broadcast domain, requiring a single subnet. Finally, the three WAN point-to-point links each require a subnet, so for this problem, ten subnets are required.

Figure W-6 Answer to Problem 4 for Determining the Number of Required Subnets



Answers to Determining the Required Number of Subnet and Host Bits

Table W-30 lists the answers for the problems stated in Table C-11 of Appendix C. Following Table W-30, the text explains the answers.

Table W-30 Answers to Determining the Number of Subnet and Host Bits Required

Problem Number	Classful Network Subnets Being Subnetted	Maximum Number of of Hosts Needed	Maximum Number per Subnet Needed	Minimum Number of Required Subnet Bits	Minimum Number of Required Host Bits
1	10.0.0.0	4000	4000	12	12
2	128.1.0.0	200	200	8	8
3	192.168.2.0	10	10	4	4
4	172.30.0.0	200	100	8	7
5	200.1.1.0	50	2	6	2
6	150.1.0.0	255	200	9	8

The minimum number of subnet bits required is *s*, where *s* is the smallest number that makes the following formula true:

$2^s - 2 =>$ the number of required subnets

Similarly, the minimum number of host bits required is *h*, where *h* is the smallest number that makes the following formula true:

$2^h - 2 =>$ the number of required hosts per subnet

As a result, determining the answer to each problem relates specifically to either calculating or memorizing the powers of 2. Table W-31 lists the powers of 2 through 2^{15}.

Table W-31 The Powers of 2, 2^0 Through 2^{15}

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16,384	32,768

With the information in Table W-31, all you need to do is look at the requirements for the number of subnets and hosts and do some quick comparisons to find the number of required subnet and host bits. The following list describes the logic used to determine the answers in Table W-30.

Problem 1 To create 4000 subnets, 11 subnet bits are not enough, because $2^{11} - 2$ is only 2046. However, 12 subnet bits would be enough, because $2^{12} - 2$ is 4094, which is larger than 4000. To support 4000 hosts per subnet, the same math applies, with 11 host bits not being enough, but 12 host bits supporting 4094 hosts per subnet, because $2^{12} - 2$ is 4094.

Problem 2 To create 200 subnets, 7 subnet bits are not enough, because $2^7 - 2$ is only 126. However, 8 subnet bits would be enough, because $2^8 - 2$ is 254, which is larger than 200. The same math applies to the process of deciding the number of host bits needed to support 200 hosts per subnet.

Problem 3 To create 10 subnets, 3 subnet bits are not enough, because $2^3 - 2$ is only 6. However, 4 subnet bits would be enough, because $2^4 - 2$ is 14, which is larger than 10. The same math applies to the process of deciding the number of host bits needed to support 10 hosts per subnet.

Problem 4 To create 200 subnets, 7 subnet bits are not enough, because $2^7 - 2$ is only 126. However, 8 subnet bits would be enough, because $2^8 - 2$ is 254, which is larger than 200. To support 100 hosts per subnet, 6 host bits are not enough, because $2^6 - 2$ is only 62. However, 7 host bits would be enough, because $2^7 - 2$ is 126, which is larger than 100.

Problem 5 To create 50 subnets, 5 subnet bits are not enough, because $2^5 - 2$ is only 30. However, 6 subnet bits would be enough, because $2^6 - 2$ is 62, which is larger than 50. To support 2 hosts per subnet, 1 host bit is not enough, because $2^1 - 2$ is 0. However, 2 host bits would be enough, because $2^2 - 2$ is 2, which provides the exact number of hosts needed per subnet.

Problem 6 To create 255 subnets, 8 subnet bits are not enough, because $2^8 - 2$ is only 254. However, 9 subnet bits would be enough, because $2^9 - 2$ is 510, which is larger than 255. To create 200 hosts per subnet, 7 host bits are not enough, because $2^7 - 2$ is only 126. However, 8 host bits would be enough, because $2^8 - 2$ is 254, which is larger than 200. In this case, with a Class B network, the requirements cannot be met, because the network part is 16 bits long per the Class B network rules, and the subnet and host parts need a total of 17 bits, making the network, subnet, and host parts 33 bits long—and the address is only 32 bits long.

Answers to Determining the Subnet Mask

Table W-32 shows the answers to the problems posed in Table C-12 of Appendix C. Following Table W-32, the text explains the answers.

Table W-32 Answers to Determining the Subnet Mask, Given a Set of Requirements

Problem Number	Classful Network Being Subnetted	Number of Required Subnet Bits	Number of Required Host Bits	Only Possible Mask, or Mask That Supports the Maximum Number of Subnets	Mask That Supports the Maximum Number of Hosts per Subnet (as Needed)
1	10.0.0.0	12	12	255.255.240.0	—
2	128.1.0.0	10	6	255.255.255.192	—
3	200.1.1.0	5	3	255.255.255.248	—
4	192.168.2.0	3	4	255.255.255.240	255.255.255.224
5	172.30.0.0	7	7	255.255.255.128	255.255.254.0
6	150.1.0.0	4	4	255.255.255.240	255.255.240.0

Chapter 10 describes two processes to find the correct subnet masks: one using binary math and one that works in some cases but uses decimal math. The next section explains the answers to all six problems using binary math, and the section following that explains four of the problems using decimal math.

Determining the Answer Using Binary Math

The network, subnet, and host parts of an IPv4 address must total 32 bits. In some cases, the design requirements may exceed 32 bits and thus may not be allowed. For instance, in a Class B network, the network part is always 16 bits long. In that case, if the number of subnets and hosts per subnet specified in the design would require more than 16 combined subnet and host bits, the design requirements simply cannot be met. For problems like those in Table C-12 of Appendix C, first make sure that the required minimum size of the network, subnet, and host parts of the address add up to 32 bits or less, to ensure that the design requirements can be met. For all six problems in Table C-12, the requirements can be met.

In cases in which the number of bits in the network, subnet, and host parts totals 32 bits, one and only one subnet mask meets the requirements. That is true of the first three problems in Table C-12 and Table W-32. For these problems, you can use the following algorithm, as listed in Chapter 10 of the book:

How To

Step 1 Write down 8, 16, or 24 binary 1s, depending on whether the network being subnetted is a Class A, B, or C network, respectively. These bits represent the network bits.

Step 2 Going left to right, write down an additional *s* binary 1s, with *s* being the number of subnet bits.

Step 3 Write down binary 0s for the rest of the bits; these represent the host bits.

Step 4 Convert this 32-bit binary number, *8 bits at a time*, back to decimal.

Answer to Problem 1: Determining the Subnet Mask

For Problem 1, Class A network 10 is used, so Step 1 requires that the mask have an octet of binary 1s. For Step 2, 12 more binary 1s are needed, because the requirements state a need for 12 subnet bits. For Step 3, the rest of the bits are binary 0s. Finally, the resulting 32-bit number must be converted to decimal, 8 bits at a time. The only tricky part of that process is to remember the 8-bits-at-a-time part, especially for the third octet, because it has some subnet bits and some host bits. The resulting mask is 255.255.240.0. The following outlines the first three steps in the process:

Step 1	Step 2	Step 3
11111111	**11111111 1111**	1111 00000000

Answer to Problem 2: Determining the Subnet Mask

For Problem 2, Class B network 128.1.0.0 is used, so Step 1 requires that the mask have two octets of binary 1s. For Step 2, 10 more binary 1s are needed, because the requirements state a need for 10 subnet bits. For Step 3, the rest of the bits are binary 0s. Finally, the resulting 32-bit number must be converted to decimal, 8 bits at a time. Again, remember the 8-bits-at-a-time part, especially for the fourth octet, because it has some subnet bits and some host bits. The resulting mask is 255.255.255.192. The following outlines the first three steps in the process:

Step 1	Step 2	Step 3
11111111 11111111	**11111111 11**	000000

Answer to Problem 3: Determining the Subnet Mask

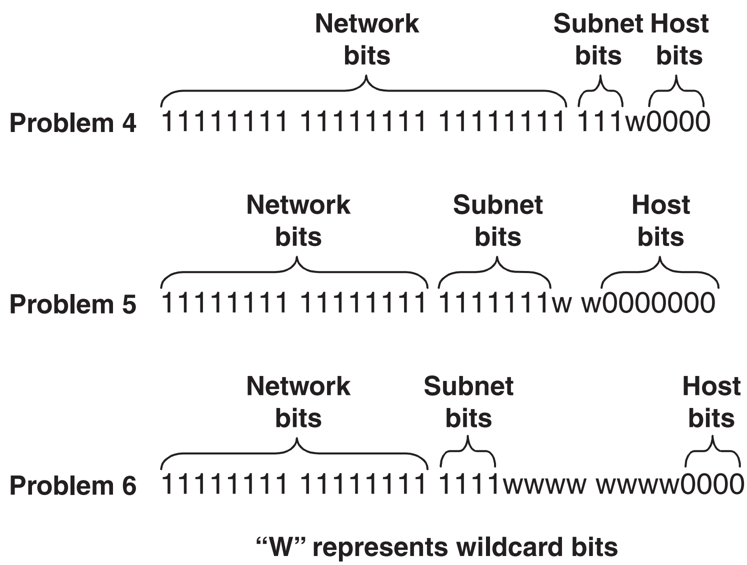
For Problem 3, Class C network 200.1.1.0 is used, so Step 1 requires that the mask have 3 octets of binary 1s. For Step 2, 5 more binary 1s are needed, because the requirements state a need for 5 subnet bits. For Step 3, the rest of the bits are binary 0s. Finally, the resulting 32-bit number must be converted to decimal, 8 bits at a time. The resulting mask is 255.255.255.248. The following outlines the first three steps in the process:

Step 1	Step 2	Step 3
11111111 11111111 11111111	**11111**	000

Finding Multiple Subnet Masks for Problems 4, 5, and 6

The first three problems had only one valid answer, but Problems 4, 5, and 6 have at least two answers. As indicated in the description of Figure 10-20 in Chapter 10, if the required number of bits in the combined network, subnet, and host fields adds up to less than 32, multiple masks meet the requirements. In those cases, you can examine the structure of the mask and determine which bits must be network, subnet, and host bits, and then list the remaining bits as wildcard bits. These wildcard bits can be either 0 or 1, within restrictions. Figure W-7 shows the format of the possible masks for Problems 4, 5, and 6, with w representing wildcard bits.

Figure W-7 Finding the Subnet Mask: Answers for Problems 4, 5, and 6



To find the possible masks, you simply substitute 0s or 1s for the wildcard bits. The more 0s used in the wildcard bits, the larger the host part of the addresses, and the more hosts supported. The more 1s used in the wildcard bits, the more subnet bits, and the larger the number of supported subnets. However, these wildcard bits of the mask must be assigned using the following rule:

> A subnet mask must begin with a number of consecutive 1s, and when a single digit is listed as binary 0 (going left to right), all the remaining digits must be 0s.

To complete Problems 4, 5, and 6, you need to find the mask that supports the maximum number of subnets while still meeting the requirements. To do so, set all the wildcard bits to 1s, making the subnet part as large as it can be. The next paragraphs explain the specific answers for Problems 4, 5, and 6.

For Problem 4, the answer that gives the largest number of subnets would be 255.255.255.240. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 1s. The following line shows the mask in binary, with the one bold digit being the one wildcard bit in this case:

11111111 11111111 11111111 111 **1** 0000

Also for Problem 4, the subnet mask that gives the largest number of hosts per subnet would be 255.255.255.224. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 0s, as shown in the following binary number:

11111111 11111111 11111111 111 **0** 0000

For Problem 5, the answer that gives the largest number of subnets would be 255.255.255.128. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 1s. The following line shows the mask in binary, with the bold digits being the wildcard bits in this case:

11111111 11111111 1111111 **1 1** 0000000

Also for Problem 5, the subnet mask that gives the largest number of hosts per subnet would be 255.255.254.0. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 0s, as shown in the following binary number:

11111111 11111111 1111111 **0 0** 0000000

For Problem 6, there are 8 wildcard bits in the mask, so many different masks are supported. The answer that gives the largest number of subnets would be 255.255.255.240. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 1s. The following line shows the mask in binary, with the eight bold digits being the wildcards bit in this case:

11111111 11111111 1111 **1111 1111** 0000

Also for Problem 6, the subnet mask that gives the largest number of hosts per subnet would be 255.255.240.0. To find the answer, you would use the mask represented in Figure W-x8 and set all the wildcard bits to 0s, as shown in the following binary number:

11111111 11111111 1111 **0000 0000** 0000

Determining the Answer Using a Decimal Shortcut

The decimal shortcut for finding the subnet mask, as described in Chapter 10, assumes that the subnet part of the address is 8 bits or less in length. The process is repeated here:

How To

Step 1 Write down the default Class A, B, or C mask (255.0.0.0, 255.255.0.0, or 255.255.255.0, respectively).

Step 2 Identify the interesting octet of the default mask. The *interesting* octet is the first octet of value 0, when reading left to right (in other words, octet 2 for Class A, octet 3 for Class B, and octet 4 for Class C). This is the octet holding all the subnet bits, given the assumption of 8 or fewer subnet bits.

Step 3 Using a subnet mask shortcut table like Table W-33, replace the interesting octet's 0 value with the correct value from the table.

Chapter 10 explains how you can derive the numbers in Table W-33, or you can simply just memorize the numbers.

Table W-33 Subnet Mask Shortcut Chart

Mask Value	128	192	224	240	248	252	254	255
Number of Subnet Bits	1	2	3	4	5	6	7	8

Problems 3 through 6 in Table C-12 (from the book's Appendix C) and Table W-32 require only 8 or fewer subnet bits. So, the decimal shortcut can be used to find the mask–specifically, the mask with the least number of subnet bits. The following list explains how each of these four problems can be solved using the decimal shortcut:

Problem 3 200.1.1.0 is a Class C network, so the default mask is 255.255.255.0. The interesting octet for this shortcut process would be the fourth octet. With a need for 5 subnet bits, as listed in Table W-32, the mask value in the interesting octet would be 248. So, the resulting mask would be 255.255.255.248.

Problem 4 192.168.2.0 is a Class C network, so the default mask is 255.255.255.0. The interesting octet for this shortcut process would be the fourth octet. With a need for 3 subnet bits, as listed in Table W-32, the mask value in the interesting octet would be 224. So, the resulting mask would be 255.255.255.224.

Problem 5 172.30.0.0 is a Class B network, so the default mask is 255.255.0.0. The interesting octet for this shortcut process would be the third octet. With a need for 7 subnet bits, as listed in Table W-32, the mask value in the interesting octet would be 254. So, the resulting mask would be 255.255.254.0.

Problem 6 150.1.0.0 is a Class B network, so the default mask is 255.255.0.0. The interesting octet for this shortcut process would be the third octet. With a need for 4 subnet bits, as listed in Table W-32, the mask value in the interesting octet would be 240. So, the resulting mask would be 255.255.240.0.

Answers to Determining the Subnet Numbers of a Single Network When Given a Subnet Mask

The following list provides the answers to the three problems listed in Table C-13 in Appendix C:

Problem 1 10.0.0.0 is the network number and the zero subnet number. The next several subnets are 10.1.0.0, 10.2.0.0, 10.3.0.0, 10.4.0.0, and so on, with the second octet incrementing by 1 for each new subnet number. Continuing that pattern, 10.253.0.0 and 10.254.0.0 are also subnets, with 10.255.0.0 being the broadcast subnet.

Problem 2 172.16.0.0 is the network number and the zero subnet number. The next several subnets are 172.16.4.0, 172.16.8.0, 172.16.12.0, 172.16.16.0, and so on, with the third octet incrementing by 4 for each new subnet number. Continuing that pattern, 172.16.244.0 and 172.16.248.0 are also subnets, with 10.252.0.0 being the broadcast subnet.

Problem 3 192.168.8.0 is the network number and the zero subnet number. The next two subnets are 192.168.8.64 and 192.168.8.128, with 192.168.8.192 being the broadcast subnet.

The next section shows how to get the answers using the binary process described in Chapter 10 of the book. After that, the decimal shortcut to find the answers to each question is described.

Finding the Answers Using Binary Processes

Explaining the binary process to find the subnets of a network when given a mask takes several pages. You should review Chapter 10's explanation of the process before doing the problems, assuming you want to practice doing the problems with the binary process. For convenience, the main algorithm is copied here from Chapter 10:

How To

Step 1 Calculate 2^s, where s is the number of subnet bits.

Step 2 Write down the binary version of the classful IP network number.

Step 3 Note that this number is the zero subnet and should not be used.

Step 4 Draw two vertical lines down the length of the page, separating the network and host parts of the 32-bit binary number.

Step 5 Copy down the network and host bits on multiple lines until you have a total of s lines of binary, with all but the first one (the zero subnet) not having any bits listed in the subnet part.

Step 6 Starting with the second row, add 1 in binary to the previous subnet field. This causes each successive subnet field to have a different value. Do this through the end of the list, which should end with a subnet field of all binary 1s.

Step 7 Note that the last entry—the one with all binary 1s in the subnet field—is the broadcast subnet and is reserved.

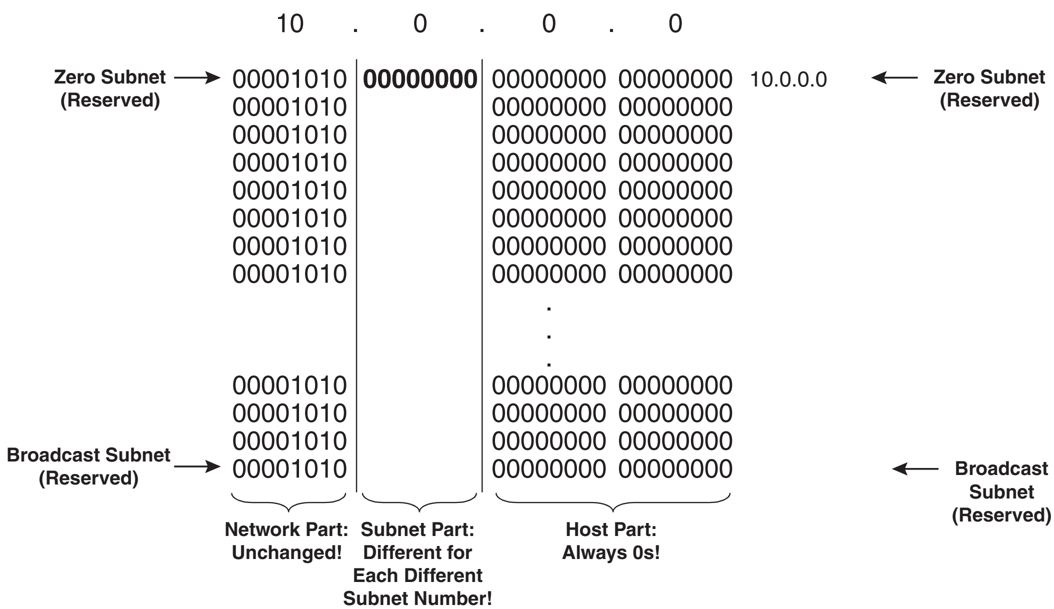
Step 8 Convert the 32-bit numbers, an octet at a time, back to decimal to have the subnet numbers in decimal.

Most of the steps set up the problem. Most of the work occurs in Steps 6 and 8.

Answer to Problem 1: Determining the Subnets of Network 10.0.0.0 Using Binary

For Problem 1 from Table C-13 in Appendix C, Figure W-8 shows the setup for the problem, essentially completing through Step 5 in the list.

Figure W-8 Problem 1 for Determining the Subnets of a Network: Through Step 5



After getting the problem organized using the first five steps, you repeat Step 6, adding binary 1 to the subnet field in the preceding numbers. For Step 7, the process stops when the subnet field grows to all binary 1s. Figure W-9 shows the results, with some of the middle subnets not shown just to save a little space.

After it is completed, you can easily find the first and last subnet numbers, which are the zero subnet and broadcast subnet, respectively.

Figure W-9 Problem 1 for Determining the Subnets of a Network: Completed

	10	. 0 .	0 . 0		
Zero Subnet (Reserved) →	00001010	**00000000**	00000000 00000000	10.0.0.0	← **Zero Subnet (Reserved)**
	00001010	**00000001**	00000000 00000000	10.1.0.0	
	00001010	**00000010**	00000000 00000000	10.2.0.0	
	00001010	**00000011**	00000000 00000000	10.3.0.0	
	00001010	**00000100**	00000000 00000000	10.4.0.0	
	00001010	**00000101**	00000000 00000000	10.5.0.0	
	00001010	**00000110**	00000000 00000000	10.6.0.0	
	00001010	**00000111**	00000000 00000000	10.7.0.0	
			. . .		
	00001010	**11111100**	00000000 00000000	10.252.0.0	
	00001010	**11111101**	00000000 00000000	10.253.0.0	
	00001010	**11111110**	00000000 00000000	10.254.0.0	
Broadcast Subnet (Reserved) →	00001010	**11111111**	00000000 00000000	10.255.0.0	← **Broadcast Subnet (Reserved)**
	Network Part: Unchanged!	**Subnet Part: Different for Each Different Subnet Number!**	**Host Part: Always 0s!**		

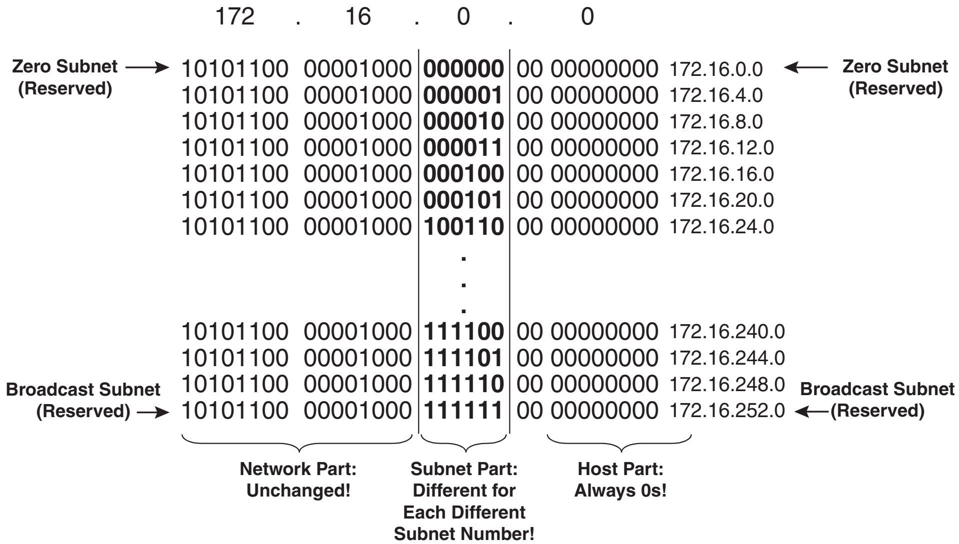
Answer to Problem 2: Determining the Subnets of Network 172.16.0.0 Using Binary

For Problem 2, Figure W-10 shows through Step 5, and Figure W-11 shows the completed process.

Figure W-10 Problem 2 for Determining the Subnets of a Network: Through Step 5

	172 . 16	. 0	. 0		
Zero Subnet (reserved) →	10101100 00001000	**000000**	00 00000000	172.16.0.0	← **Zero Subnet (reserved)**
	10101100 00001000		00 00000000	172.16.4.0	
	10101100 00001000		00 00000000	172.16.8.0	
	10101100 00001000		00 00000000	172.16.12.0	
	10101100 00001000		00 00000000	172.16.16.0	
	10101100 00001000		00 00000000	172.16.20.0	
	10101100 00001000		00 00000000	172.16.24.0	
		. . .			
	10101100 00001000		00 00000000	172.16.240.0	
	10101100 00001000		00 00000000	172.16.244.0	
	10101100 00001000		00 00000000	172.16.248.0	
Broadcast subnet (reserved) →	10101100 00001000		00 00000000	172.16.252.0	← **Broadcast subnet (reserved)**
	Network Part: Unchanged!	**Subnet Part: Different for Each Different Subnet Number!**	**Host Part: Always 0s!**		

Figure W-11 Problem 2 for Determining the Subnets of a Network: Completed



For Problem 2, as usual, the network number (172.16.0.0) is the same numerical value as the zero subnet. The broadcast subnet is 172.16.252.0, with the rest of the subnets being the ones recommended for use (at least until later Networking Academy CCNA courses, in which classless routing is introduced).

Answer to Problem 3: Determining the Subnets of Network 192.168.8.0 Using Binary

For Problem 3, Figure W-12 shows through Step 5, and Figure W-13 shows the completed process.

Figure W-12 Problem 3 for Determining the Subnets of a Network: Through Step 5

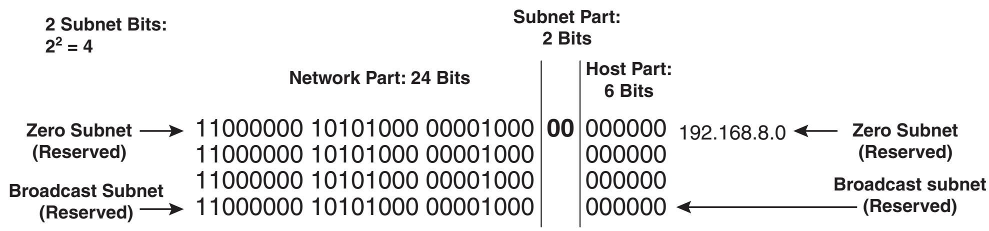
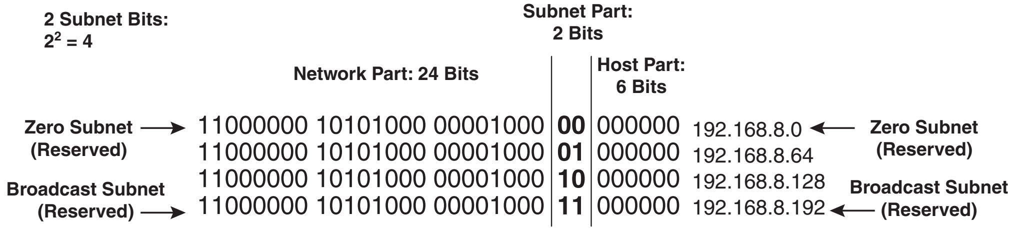


Figure W-13 Problem 3 for Determining the Subnets of a Network: Completed



For Problem 3, as usual, the network number (192.168.8.0) is the same numerical value as the zero subnet. The broadcast subnet is 192.168.8.192.0. The only other two subnets are 192.168.8.64 and 192.168.8.128.

Finding the Answers Using Decimal Shortcuts

Chapter 10 describes a shortcut to find the subnet numbers. The shortcut assumes 8 or fewer subnet bits, which is true for all three problems in this section. If it is not fresh in your memory, you may want to review the shortcut process described in Chapter 10. For convenience, the core steps of the process are listed here:

How To

Step 1 Create a table like Table W-34, with the number of empty rows being 2^S, one row for every numerical subnet. (If that table is too big to draw, just leave some lines for the first several subnets and the last several subnets.)

Step 2 Write down the zero subnet in the first empty row's Subnet column.

Step 3 Write down the network-wide broadcast address in the last row's Broadcast Address column.

Step 4 Decide which octet holds all the subnet bits. (Remember, this process assumes at most 8 subnet bits.) For Class A networks, this will be the second octet; for Class B networks, the third octet; and for Class C networks, the fourth octet.

Step 5 To find the subnet numbers, take the previous subnet number, and add the increment to the octet where the subnet bits sit.

You can use Table W-34 for the process that is described in Chapter 10 and repeated in the preceding list. You do not need to draw the table when working a problem, but it is useful for organizing the information, particularly when you also need to find the range of valid IP addresses and broadcast address for each subnet.

Table W-34 Subnetting Shortcut Table

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address

Answer to Problem 1: Determining the Subnets of Network 10.0.0.0 Using Decimal

For Problem 1 in Table C-13 of Appendix C, Table W-35 shows the setup of the subnetting shortcut table through Step 4 in the process. Table W-36 shows the table with column 1 completed. The key values used by the process are as follows:

- The interesting octet is the second octet.
- The increment—the value added to the interesting octet to find the next subnet number—is 256 – 255 = 1.

Table W-35 Problem 1 Through Step 4

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
10.0.0.0			
			10.255.255.255

Table W-36 shows the consecutive subnet numbers, with the interesting octet, the second octet, incrementing by 1 in each case.

Table W-36 Problem 1 Completed

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
10.0.0.0			
10.1.0.0			
10.2.0.0			
10.3.0.0			
Omitting many values…			
10.253.0.0			
10.254.0.0			
10.255.0.0			10.255.255.255

Answer to Problem 2: Determining the Subnets of Network 172.16.0.0 Using Decimal

For Problem 2 in Table C-13 of Appendix C, Table W-37 shows the setup of the subnetting shortcut table through Step 4 in the process. Table W-38 shows the table with column 1 completed. The key values used by the process are as follows:

- The interesting octet is the third octet.
- The increment—the value added to the interesting octet to find the next subnet number—is 256 – 252 = 4.

Table W-37 Problem 2 Through Step 4

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
172.16.0.0			
			172.16.255.255

Table W-38 shows the consecutive subnet numbers, with the interesting octet, the third octet, incrementing by 4 in each case.

Table W-38 Problem 2 Completed

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
172.16.0.0			
172.16.4.0			
172.16.8.0			
172.16.12.0			
Omitting many values…			
172.16.244.0			
172.16.248.0			
172.16.252.0			172.16.255.255

Answer to Problem 3: Determining the Subnets of Network 192.168.8.0 Using Decimal

For Problem 3 in Table C-13 of Appendix C, Table W-39 shows the setup of the subnetting shortcut table through Step 4 in the process. Table W-40 shows the table with column 1 completed. The key values used by the process are as follows:

- The interesting octet is the fourth octet.
- The increment—the value added to the interesting octet to find the next subnet number—is 256 – 192 = 64.

Table W-39 Problem 3 Through Step 4

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
192.168.8.0			
			192.168.8.255

Table W-40 shows the consecutive subnet numbers, with the interesting octet, the fourth octet, incrementing by 64 in each case.

Table W-40 Problem 3 Completed

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
192.168.8.0			
192.168.8.64			
192.168.8.128			
192.168.8.192			192.168.8.255

Answers to Determining the Subnet Broadcast Addresses and Range of Assignable Addresses

Table W-41, Tables W-42, and W-43 list the answers to the problems in Table C-14 in Appendix C. Following those tables, two sections describe how to find the answers: one section covers the binary processes, and the other section covers the decimal shortcuts.

Table W-41 Answers to Problem 1

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
10.0.0.0	10.0.0.1	10.0.255.254	10.0.255.255
10.1.0.0	10.1.0.1	10.1.255.254	10.1.255.255
10.2.0.0	10.2.0.1	10.2.255.254	10.2.255.255
10.3.0.0	10.3.0.1	10.3.255.254	10.3.255.255
Omitting many values…			
10.253.0.0	10.253.0.1	10.253.255.254	10.253.255.255
10.254.0.0	10.254.0.1	10.254.255.254	10.254.255.255
10.255.0.0	10.255.0.1	10.255.255.254	10.255.255.255

Table W-42 Answers to Problem 2

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
172.16.0.0	172.16.0.1	172.16.3.254	172.16.3.255
172.16.4.0	172.16.4.1	172.16.7.254	172.16.7.255
172.16.8.0	172.16.8.1	172.16.11.254	172.16.11.255
172.16.12.0	172.16.12.1	172.16.15.254	172.16.15.255
Omitting many values…			
172.16.244.0	172.16.244.1	172.16.247.254	172.16.247.255
172.16.248.0	172.16.248.1	172.16.251.254	172.16.251.255
172.16.252.0	172.16.252.1	172.16.255.254	172.16.255.255

Table W-43 Answers to Problem 3

Subnet	Lowest IP Address	Highest IP Address	Broadcast Address
192.168.8.0	192.168.8.1	192.168.8.62	192.168.8.63
192.168.8.64	192.168.8.65	192.168.8.126	192.168.8.127
192.168.8.128	192.168.8.129	192.168.8.190	192.168.8.191
192.168.8.192	192.168.8.193	192.168.8.254	192.168.8.255

Finding the Answers Using Binary

Chapter 10 describes how to find the range of assignable addresses in each subnet. To find the range, you first must find the broadcast address for each subnet. The process to find the broadcast addresses, after the subnet numbers have been found, is as follows:

How To

Step 1 In each binary subnet number, change all the host bits to binary 1.

Step 2 Convert these new numbers back to decimal, 8 bits at a time (even if an octet is partly subnet and partly host).

The previous set of problems asked you to find all the subnet numbers. If you used the binary process as described in Chapter 10 of the book, you should already have the subnet numbers listed, along with the host part noted. If not, the binary subnet numbers for all three problems are shown in Figure W-x10, Figures W-x12, and W-x14, respectively.

To find the subnet broadcast addresses for Problem 1, you must change the last 16 bits of the subnet numbers to binary 1s. The following list shows the binary and decimal values of the first four subnets for Problem 1, with the host bits shown in bold:

00001010 00000000 **11111111 11111111**—10.0.255.255

00001010 00000001 **11111111 11111111**—10.1.255.255

00001010 00000010 **11111111 11111111**—10.2.255.255

00001010 00000011 **11111111 11111111**—10.3.255.255

To find the subnet broadcast addresses for Problem 2, you must change the last 10 bits of the subnet numbers to binary 1s. The following list shows the binary and decimal values of the first four subnets for Problem 1, with the host bits shown in bold:

10101100 00010000 000000**11 11111111**—172.16.3.255

10101100 00010000 000001**11 11111111**—172.16.7.255

10101100 00010000 000010**11 11111111**—172.16.11.255

10101100 00010000 000011**11 11111111**—172.16.15.255

Two parts of this process may pose a particular challenge. First, the host part of the address is the last 10 bits, which includes the entire fourth octet and the last 2 bits of the third octet. To find a subnet's broadcast address, the host bits in the subnet number must be changed to binary 1s. Then, to convert the value back to decimal, the conversion must always convert 8 bits at a time back to decimal. Sometimes people learning about subnetting see that an octet has some subnet bits and some host bits, like octet 3 in this case, and think that the conversion process should keep the bits separate. However, whenever converting a 32-bit address, subnet, or broadcast address back to its dotted-decimal form, the conversion always happens 8 bits at a time.

To find the subnet broadcast addresses for Problem 3, you must change the last 6 bits of the subnet numbers to binary 1s. The following list shows the binary and decimal values of subnets for Problem 1, with the host bits shown in bold:

11000000 10101000 00001000 00**111111**—192.168.8.63

11000000 10101000 00001000 01**111111**—192.168.8.127

11000000 10101000 00001000 10**111111**—192.168.8.191

11000000 10101000 00001000 11**111111**—192.168.8.255

The decimal versions of the broadcast addresses in this problem sometimes confuse people because they simply do not look like they are correct. Some broadcast addresses have 255s in some of the last octets, but others do not. The only requirement is that all subnet broadcast addresses must have all binary 1s in the host part of the number. As can be seen with these answers, just because a dotted-decimal number has something besides 255 in the last octet does not mean that the number isn't a subnet broadcast address.

After finding the subnet number and subnet broadcast address for each subnet, finding the range of assignable IP addresses in each subnet is simple:

How To

Step 1 To find the smallest of the assignable IP addresses in a subnet, add 1 to the fourth octet of the subnet number.

Step 2 To find the largest of the assignable IP addresses in a subnet, subtract 1 from the fourth octet of the subnet broadcast address.

Finding the Answers Using a Decimal Shortcut

Chapter 10 in the book describes a decimal-only shortcut with which to find the subnet numbers and their subnet broadcast addresses. The previous section asked you to find the subnet numbers. Tables W-36, Table W-38, and W-40 list the subnet numbers. To find each subnet's broadcast address, after you build tables like W-36, W-38, and W-40, you need to use only the following bit of logic, beginning with the last subnet's broadcast address listed in the lower-right corner of the tables:

> To find the broadcast addresses, take the later broadcast address, and subtract the increment from the octet where the subnet bits sit.

For example, with Problem 1, the broadcast subnet's broadcast address is 10.255.255.255. The second octet is the interesting octet in this case, because all 8 subnet bits sit inside the second octet. In this case, the increment is 1 because the mask's second octet is 255, and 256 – 255 = 1. So, to find the next-lower subnet's broadcast address, subtract 1 from 10.255.255.255's second octet, giving a value of 10.254.255.255. Repeat the process, yielding 10.253.255.255, 10.252.255.255, 10.251.255.255, and so on. The last few subnet broadcast addresses found—the broadcast addresses for the numerically smallest subnets—are 10.2.255.255, 10.1.255.255, and the broadcast address for the zero subnet, 10.0.255.255. Table W-41 earlier in this section shows the values.

For Problem 2, the broadcast subnet's broadcast address is 172.16.255.255. The third octet is the interesting octet in this case, because all 6 subnet bits sit inside the third octet. In this case, the increment is 4, because the mask's third octet is 252, and 256 – 252 = 4. To find the next-lower subnet's broadcast address, subtract 4 from 172.16.255.255's third octet, giving a value of 172.16.251.255. Repeat the process, yielding 172.16.247.255, 172.16.243.255, 172.16.239.255, and so on. The last few subnet broadcast addresses found—the broadcast addresses for the numerically smallest subnets—are 172.16.11.255, 172.16.7.255, and the broadcast address for the zero subnet, 172.16.3.255. Table W-42 earlier in this section shows the values.

Finally, for Problem 3, the broadcast subnet's broadcast address is 192.168.8.255. The fourth octet is the interesting octet in this case, because both subnet bits sit inside the fourth octet. In this case, the increment is 64, because the mask's fourth octet is 192, and 256 – 192 = 64. So, to find the next-lower subnet's broadcast address, subtract 64 from 192.168.8.255's fourth octet, giving a value of 192.168.8.191. Repeat the process, yielding 192.168.8.127. The final subnet broadcast address is the next one calculated, 192.168.8.63. Table W-43 earlier in this section shows the values.

For all problems, when you know the subnet number and subnet broadcast addresses, the following simple two-step process finds the range of assignable addresses:

How To

Step 1 To find the smallest of the assignable IP addresses in a subnet, add 1 to the fourth octet of the subnet number.

Step 2 To find the largest of the assignable IP addresses in a subnet, subtract 1 from the fourth octet of the subnet broadcast address.

Answers to Finding the Resident Subnet

Table W-44 lists the answers to the problems from Table C-15 in Appendix C. Following Table W-44, the text explains the answers, using both binary math and decimal shortcuts.

Table W-44 Answers to Finding the Resident Subnet

Problem Number	IP Address	Mask	Resident Subnet
1	10.1.2.3	255.255.0.0	10.1.0.0
2	172.16.100.2	255.255.254.0	172.16.100.0
3	192.168.8.201	255.255.255.252	192.168.8.200
4	192.168.19.177	255.255.255.240	192.168.19.176
5	172.30.200.200	255.255.248.0	172.30.200.0
6	200.1.1.180	255.255.255.192	200.1.1.128

Finding the Answers Using Binary

To find the resident subnet in binary, you can use a simple procedure to perform a bitwise Boolean AND on the IP address and mask:

How To

Step 1 Convert the IP address and mask to binary, writing the IP address first, and the subnet mask directly below it.

Step 2 Perform a bitwise Boolean AND of the two numbers.

Step 3 Convert the resulting 32-bit number, 8 bits at a time, back to decimal.

You can use the math covered in Chapter 1 for the decimal and binary conversion processes,

or you can use the conversion table in Appendix B of the book. The math for the conversion process is not shown here. However, the binary IP addresses, masks, and results of the Boolean AND are shown in upcoming tables.

Table W-45 shows the work for Problem 1, which is for IP address 10.1.2.3, mask 255.255.0.0.

Table W-45 Answer to Problem 1: Finding the Resident Subnet 10.1.2.3

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 10.1.2.3 (Step 1)	00001010	00000001	00000010	00000011
Mask 255.255.0.0 (Step 1)	11111111	11111111	00000000	00000000
AND Result—Subnet Number (Step 2)	00001010	00000001	00000000	00000000
Decimal Subnet Number (Step 3)	10.1.0.0			

Table W-46 shows the work for Problem 2, which is for IP address 172.16.100.2, mask 255.255.254.0.

Table W-46 Answer to Problem 2: Finding the Resident Subnet 172.16.100.2

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 172.16.100.2 (Step 1)	10101100	00010000	01100100	00000010
Mask 255.255.254.0 (Step 1)	11111111	11111111	11111110	00000000
AND Result—Subnet Number (Step 2)	10101100	00010000	01100100	00000000
Decimal Subnet Number (Step 3)	172.16.100.0			

Table W-47 shows the work for Problem 3, which is for IP address 192.168.8.201, mask 255.255.255.252.

Table W-47 Answer to Problem 3: Finding the Resident Subnet of 192.168.8.201

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 192.168.8.201 (Step 1)	11000000	10101000	00001000	11001001
Mask 255.255.255.252 (Step 1)	11111111	11111111	11111111	11111100
AND Result—Subnet Number (Step 2)	11000000	10101000	00001000	11001000
Decimal Subnet Number (Step 3)	192.168.8.200			

Table W-48 shows the work for Problem 4, which is for IP address 192.168.19.177, mask 255.255.255.240.

Table W-48 Answer to Problem 4: Finding the Resident Subnet 192.168.19.177

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 192.168.19.177 (Step 1)	11000000	10101000	00010011	10110001
Mask 255.255.255.240 (Step 1)	11111111	11111111	11111111	11110000
AND Result—Subnet Number (Step 2)	11000000	10101000	00010011	10110000
Decimal Subnet Number (Step 3)	192.168.19.176			

Table W-49 shows the work for Problem 5, which is for IP address 172.30.200.200, mask 255.255.248.0.

Table W-49 Answer to Problem 5: Finding the Resident Subnet 172.30.200.200

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 172.30.200.200 (Step 1)	10101100	00011110	11001000	11001000
Mask 255.255.248.0 (Step 1)	11111111	11111111	11111000	00000000
AND Result—Subnet Number (Step 2)	10101100	00011110	11001000	00000000
Decimal Subnet Number (Step 3)	172.30.200.0			

Table W-50 shows the work for Problem 6, which is for IP address 200.1.1.180, mask 255.255.255.192.

Table W-50 Answer to Problem 6: Finding the Resident Subnet 200.1.1.180

	First Octet	Second Octet	Third Octet	Fourth Octet
IP Address 200.1.1.180 (Step 1)	11001000	00000001	00000001	10110100
Mask 255.255.255.192 (Step 1)	11111111	11111111	11111111	11000000
AND Result—Subnet Number (Step 2)	11001000	00000001	00000001	10000000
Decimal Subnet Number (Step 3)	200.1.1.128			

Finding the Answers Using a Decimal Shortcut

The decimal shortcut to find the resident subnet is part of the same shortcut process to find all subnets of the same Class A, B, or C network in which the address resides. As covered in Chapter 10, the process is as follows:

How To

Step 1 Write down the Class A, B, or C network number in which the address resides.

Step 2 Using the process defined in Chapter 10, discover all subnet numbers of the Class A, B, or C network, using the stated subnet mask.

Step 3 Compare the IP address to the subnet numbers, finding the subnet number closest to the IP address, but still less than the IP address. That is the resident subnet.

The following sections explain this process for each of the six problems.

Answer to Problem 1: Finding the Resident Subnet for 10.1.2.3, Mask 255.255.0.0

IP address 10.1.2.3 is in Class A network 10.0.0.0. Using the shortcut process to find the possible subnets of network 10.0.0.0, using mask 255.255.0.0, the interesting octet is the second octet, and the increment is 256 – 255 = 1 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

10.0.0.0 (zero subnet)

10.1.0.0

10.2.0.0

10.3.0.0

and so on…

10.254.0.0

10.255.0.0 (broadcast subnet)

IP address 10.1.2.3 is between subnets 10.1.0.0 and 10.2.0.0. Because 10.1.0.0 is the smaller of the two subnet numbers, it is the resident subnet.

Answer to Problem 2: Finding the Resident Subnet for 172.16.100.2, Mask 255.255.254.0

IP address 172.16.100.2 is in Class B network 172.16.0.0. Using the shortcut process to find the possible subnets of network 172.16.0.0, using mask 255.255.254.0, the interesting octet is the third octet, and the increment is 256 – 254 = 2 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

172.16.0.0 (zero subnet)

172.16.2.0

172.16.4.0

172.16.6.0

and so on…

172.16.96.0

172.16.98.0

172.16.100.0

172.16.102.0

and so on…

172.16.252.0

172.16.254.0 (broadcast subnet)

IP address 172.16.100.2 is between subnets 172.16.100.0 and 172.16.102.0. Because 172.16.100.0 is the smaller of the two subnet numbers, it is the resident subnet.

Answer to Problem 3: Finding the Resident Subnet for 192.168.8.201, Mask 255.255.255.252

IP address 192.168.8.201 is in Class C network 192.168.8.0. Using the shortcut process to find the possible subnets of network 192.168.8.0, using mask 255.255.255.252, the interesting octet is the fourth octet, and the increment is 256 – 252 = 4 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

192.168.8.0 (zero subnet)

192.168.8.4

192.168.8.8

192.168.8.12

and so on…

192.168.8.196

192.168.8.200

192.168.8.204

192.168.8.208

and so on…

192.168.8.248

192.168.8.252 (broadcast subnet)

IP address 192.168.8.201 is between subnets 192.168.8.200 and 192.168.8.204. Because 192.168.8.200 is the smaller of the two subnet numbers, it is the resident subnet.

Answer to Problem 4: Finding the Resident Subnet for 192.168.19.177, Mask 255.255.255.240

IP address 192.168.19.177 is in Class C network 192.168.190. Using the shortcut process to find the possible subnets of network 192.168.19.0, mask 255.255.255.240, the interesting octet is the fourth octet, and the increment is 256 – 240 = 16 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

192.168.19.0 (zero subnet)

192.168.19.16

192.168.19.32

192.168.19.48

192.168.19.64

192.168.19.80

192.168.19.96

192.168.19.112

192.168.19.128

192.168.19.144

192.168.19.160

192.168.19.176

192.168.19.192

192.168.19.208

192.168.19.224

192.168.19.240 (broadcast subnet)

IP address 192.168.19.177 is between subnets 192.168.19.176 and 192.168.19.192. Because 192.168.19.176 is the smaller of the two subnet numbers, it is the resident subnet.

Answer to Problem 5: Finding the Resident Subnet for 172.30.200.200, Mask 255.255.248.0

IP address 172.30.200.200 is in Class B network 172.30.0.0. Using the shortcut process to find the possible subnets of network 172.30.0.0, using mask 255.255.248.0, the interesting octet is the third octet, and the increment is 256 – 248 = 8 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

172.30.0.0 (zero subnet)

172.30.8.0

172.30.16.0

and so on…

172.30.184.0

172.30.192.0

172.30.200.0

172.30.208.0

172.30.216.0

and so on…

172.30.240.0

172.30.248.0 (broadcast subnet)

IP address 172.30.200.200 is between subnets 172.30.200.0 and 172.30.208.0. Because 172.30.200.0 is the smaller of the two subnet numbers, it is the resident subnet.

Answer to Problem 6: Finding the Resident Subnet for 200.1.1.180, Mask 255.255.255.192

IP address 200.1.1.180 is in Class C network 200.1.1.0. Using the shortcut process to find the possible subnets of network 200.1.1.0, using mask 255.255.255.192, the interesting octet is the fourth octet, and the increment is 256 – 192 = 64 (256 minus the mask's value in the interesting octet). So, the subnet numbers are as follows:

200.1.1.0 (zero subnet)

200.1.1.64

200.1.1.128

200.1.1.192 (broadcast subnet)

IP address 200.1.1.180 is between subnets 200.1.1.128 and 200.1.1.192. Because 200.1.1.128 is the smaller of the two subnet numbers, it is the resident subnet.