Boson

Subnetting Study Guide

by Boson Software, LLC

An octet is a binary number of 8 bits, with the lowest possible number being 00000000 in binary, which converts to 0 in dotted decimal format. The highest possible number is 11111111 in binary, which converts to 255 in dotted decimal format.

These examples make use of the "all zeros" and "all ones" subnets as per RFC 1878.

Table 1

Address Class Summary			
Class	**Number of Subnets**	**Number of Hosts Per Subnet**	**Range of Network IDs (First Octet)**
Class A	128	16,777,214	1 – 126
Class B	16,386	65,534	128 – 191
Class C	2,097,154	254	192 – 223

The First-Octet Rule: *High-Order Bits Determine the Address Class*

The decimal number in the first octet, or the high-order bits, represents the network ID for the address, and that network ID remains the same for all hosts on that network segment. High-order bits, which are also called the most significant bits, are those in the left-most portion of the binary address. The remaining octets act as host IDs. Address classes are categorized based on the decimal number in the first octet, as seen in Table 1, the **Address Class Summary** table.

Table 2

Class	Binary Start	Binary End	First Octet	Most Significant Bits	Special Note
Class A	00000000	01111111	0 – 127*	0	Assignable
Class B	10000000	10111111	128 – 191	10	Assignable
Class C	11000000	11011111	192 – 223	110	Assignable
Class D	11101111	11101111	224 – 239	1110	Multicast
Class E	11110000	11111111	240 – 255	11110	InterNic

*Addresses beginning with 0 are used for the network.
*Addresses beginning with 127 are used for loopback testing.

Table 3

Class	Total Number of Subnets Per Class	Equation	Subnets
Class A	**01111111**.00000000.00000000.00000000	$2^7 = 128-2 = 126$*	Net.H.H.H
Class B	1**0**1**11111.11111111**.00000000.00000000	$2^{14} = 16,386$	Net.Net.H.H
Class C	**11011111.11111111.11111111**.00000000	$2^{21} = 2,097,154$	Net.Net.Net.H

*Addresses beginning with 0 are used for the network.
*Addresses beginning with 127 are used for loopback testing.

Table 4

Class	Total Number of Hosts Per Subnet	Equation	Hosts
Class A	00000000.**11111111.11111111.11111111**	$2^{24} - 2 = 16,777,214$	*x*.255.255.255
Class B	00000000.00000000.**11111111.11111111**	$2^{16} - 2 = 65,534$	*x.y*.255.255
Class C	00000000.00000000.00000000.**11111111**	$2^8 - 2 = 254$	*x.y.z*.255

Table 5

Reserved Address Space *RFC 1166 and 1918 = Private Address Space (Internal Use Only)		
Netblock	Special Use	Reference
10.*x.x.x*	Private	RFC 1918
127.*x.x.x*	Loopback	Diagnostics
172.(16 – 31).*x.x*	Private	RFC 1918
192.0.0.*x*	Reserved	JBP
192.0.1.*x*	Backbone – Test – C	RH6
192.0.2.*x*	Internet – Test – C	JBP
192.0.(3 – 255).*x*	Unassigned	NIC
192.1.(0 – 1).*x*	Backbone Local Nets	SGC
192.1.3.*x*	Backbone Apollo Nets	SGC
192.168.*x.x*	Private	RFC 1918

Converting Decimal into Binary

1. Create a chart with eight columns.
2. In the first row, enter **1** in the far right column. Moving to the left, double the preceding value to enter **2** in the next column, **4** in the next column, **8** in the next column, etc.
3. Continue in that manner until all columns in the first row contain a value. The column on the far left should contain the value **128**. These values are called weighted values.
4. Beginning with the left-most weighted value, subtract each weighted value from the decimal being converted to binary until you reach the largest weighted value that can be subtracted without yielding a negative result. Enter a **1** as the binary value for that weighted value, and enter a 0 as the binary values for the weighted values that yielded negative results.
5. After reaching the largest weighted value that can be subtracted from the decimal without yielding a negative result, subtract the next weighted value from the remainder of the previous weighted value, not from the decimal. If a subsequent weighted value yields a negative result, enter a **0** as the binary value and subtract the next weighted value from the remainder yielded by the last successful subtraction (**Note**: See Weighted Values 2 and 1 in Table 6 below for an example).
6. Table 6 below illustrates the conversion of the decimal 29 into binary:

Table 6

	Most Significant Bit							Least Significant Bit
Weighted Values	128	64	32	16	8	4	2	1
Subtraction from Decimal	29 – 128 = **–99**	29 – 64 = **–35**	29 – 32 = **–3**	29 – 16 = 13	13 – 8 = 5	5 – 4 = 1	1 – 2 = **–1**	1 – 1 = 0
Binary	0	0	0	1	1	1	0	1

7. The decimal 29 converts into binary as `00011101`.

Converting Binary into Decimal

1. Create a chart with eight columns.
2. In the first row, enter **1** in the far right column. Moving to the left, double the preceding value to enter **2** in the next column, **4** in the next column, **8** in the next column, etc.
3. Continue in that manner until all columns in the first row contain a value. The column on the far left should contain the value **128**. These values are called weighted values.
4. Enter the binary value in the chart below the weighted values so that there is one **0** or one **1** below each weighted value.
5. Add together all weighted values that have **1**s below them.
6. Table 7 below illustrates the conversion of the binary `11000011` into decimal:

Table 7

	Most Significant Bit						Least Significant Bit
Weighted Values	**128**	**64**	32	16	8	4	**2**	**1**
Binary	1	1	0	0	0	0	1	1

7. The binary `11000011` converts into the decimal 195 (128 + 64 + 2 + 1 = 195).

Subnetting

Subnetting allows a large network block to be logically subdivided into multiple smaller networks, or subnets. The use of multiple smaller networks allows the use of varying physical networks, such as Ethernet or Token Ring, which could not otherwise be combined. Additionally, the smaller subnets can improve the speed of traffic and permit easier management.

Subnet masks provide the logical segmentation required by routers to be able to address logical subnets. A default gateway IP address on a router interface allows clients to access networks outside their local subnets. The number of bits used in a subnet mask determines the number of subnets available.

Things to Remember

- The incremental value is the IP host address starting point for the subnet.
- Each incremental value represents another logical subnet.
- Routers only use the network ID and the subnet broadcast.
- Clients only use the IP address on the clients' local subnets and the clients' subnet broadcasts.
- Each bit represents a power of 2.

Table 8

Binary Mask	Decimal Mask	Binary Hosts	Bits Used	Number of Subnets	Valid Increments
00000000	0	HHHHHHHH	0	Net ID	Not a subnet
10000000	128	NHHHHHHH	1	2	128
11000000	192	NNHHHHHH	2	4	64
11100000	224	NNNHHHHH	3	8	32
11110000	240	NNNNHHHH	4	16	16
11111000	248	NNNNNHHH	5	32	8
11111100	252	NNNNNNHH	6	64	4
11111110	254	NNNNNNNH	7	128	2
11111111	255	NNNNNNNN	8	256	1

Legend: Red represents network bit(s), whereas green represents host bit(s).

Logical Addressing

All subnets can be categorized into seven ranges with three classes each (Class A, Class B, and Class C).

Table 9

Class A						
Maximum Number of Subnets	Calculation for Host ID	Maximum Number of Hosts Per Subnet	Subnet Mask	Subnet Bits Required	Host Bits Available	Range
0	$2^{24} - 2$	16,777,214	255.0.0.0	0	24	Begin Class A
2	$2^{23} - 2$	8,388,606	255.128.0.0	1	23	Range 1/8
4	$2^{22} - 2$	4,194,302	255.192.0.0	2	22	Range 2/8
8	$2^{21} - 2$	2,097,150	255.224.0.0	3	21	Range 3/8
16	$2^{20} - 2$	1,048,574	255.240.0.0	4	20	Range 4/8
32	$2^{19} - 2$	524,286	255.248.0.0	5	19	Range 5/8
64	$2^{18} - 2$	262,142	255.252.0.0	6	18	Range 6/8
128	$2^{17} - 2$	131,070	255.254.0.0	7	17	Range 7/8

Table 10

Class B						
Maximum Number of Subnets	Calculation for Host ID	Maximum Number of Hosts Per Subnet	Subnet Mask	Subnet Bits Required	Host Bits Available	Range
0	$2^{16} - 2$	65,534	255.255.0.0	8	16	Class A Range 8/8 and Begin Class B
2	$2^{15} - 2$	32,766	255.255.128.0	9	15	Range 1/8
4	$2^{14} - 2$	16,382	255.255.192.0	10	14	Range 2/8
8	$2^{13} - 2$	8,190	255.255.224.0	11	13	Range 3/8
16	$2^{12} - 2$	4,094	255.255.240.0	12	12	Range 4/8
32	$2^{11} - 2$	2,046	255.255.248.0	13	11	Range 5/8
64	$2^{10} - 2$	1,022	255.255.252.0	14	10	Range 6/8
128	$2^{9} - 2$	510	255.255.254.0	15	9	Range 7/8

Table 11

Class C						
Maximum Number of Subnets	Calculation for Host ID	Maximum Number of Hosts Per Subnet	Subnet Mask	Subnet Bits Required	Host Bits Available	Range
0	$2^{8} - 2$	254	255.255.255.0	16	8	Class B Range 8/8 and Begin Class C
2	$2^{7} - 2$	126	255.255.255.128	17	7	Range 1/6
4	$2^{6} - 2$	62	255.255.255.192	18	6	Range 2/6
8	$2^{5} - 2$	30	255.255.255.224	19	5	Range 3/6
16	$2^{4} - 2$	14	255.255.255.240	20	4	Range 4/6
32	$2^{3} - 2$	6	255.255.255.248	21	3	Range 5/6
64	$2^{2} - 2$	2	255.255.255.252	22	2	Range 6/6

Things to Remember

- The octets that contain only 1s or only 0s do not need additional calculation.
- If there are only 1s, the decimal value is 255; if there are only 0s, the decimal value is 0.
- Calculation is only required for the octet that has been broken at the bit boundary.

Class C Example Breakdown

Table 12

Subnet	Octet	Range	Incremental Value	Description
Network	00/000000	0	64	net ID (goes in routing table)
	00/000001	1		first host ID in Subnet 0
	00/111110	62		last host ID in Subnet 0
	00/111111	63		broadcast for only Subnet 0
Subnet 1 of 4	01/000000	64	64	net ID (goes in routing table)
	01/000001	65		first host ID in Subnet 1
	01/111110	126		last host ID in Subnet 1
	01/111111	127		broadcast for only Subnet 1
Subnet 2 of 4	10/000000	128	64	net ID (goes in routing table)
	10/000001	129		first host ID in Subnet 2
	10/111110	190		last host ID in Subnet 2
	10/111111	191		broadcast for only Subnet 2
Broadcast Subnet	11/000000	192	64	net ID (goes in routing table
	11/000001	193		first host ID in All 1s subnet
	11/111110	254		last host ID in All 1s subnet
	11/111111	255		local wire (all subnets) broadcast
Network	000/00000	0	32	net ID (goes in routing table)
	000/00001	1		first host ID in Subnet 0
	000/11110	30		last host ID in Subnet 0
	000/11111	31		broadcast for only Subnet 0
Subnet 1 of 8	001/00000	32	32	net ID (goes in routing table)
	001/00001	33		first host ID in Subnet 1
	001/11110	62		last host ID in Subnet 1
	001/11111	63		broadcast for only Subnet 1
Subnet 2 of 8	010/00000	64	32	net ID (goes in routing table)
	010/00001	65		first host ID in Subnet 2
	010/11110	94		last host ID in Subnet 2
	010/11111	95		broadcast for only Subnet 2
Subnet 3 of 8	011/00000	96	32	net ID (goes in routing table)
	011/00001	97		first host ID in Subnet 3

Subnet	Octet	Range	Incremental Value	Description
	011/11110	126		last host ID in Subnet 3
	011/11111	127		broadcast for only Subnet 3
Subnet 4 of 8	100/00000	128	32	net ID (goes in routing table)
	100/00001	129		first host ID in Subnet 4
	100/11110	158		last host ID in Subnet 4
	100/11111	159		broadcast for only Subnet 4
Subnet 5 of 8	101/00000	160	32	net ID (goes in routing table)
	101/00001	161		first host ID in Subnet 5
	101/11110	190		last host ID in Subnet 5
	101/11111	191		broadcast for only Subnet 5
Subnet 6 of 8	110/00000	192	32	net ID (goes in routing table)
	110/00001	193		first host ID in Subnet 6
	110/11110	222		last host ID in Subnet 6
	110/11111	223		broadcast for only Subnet 6
Broadcast Subnet	111/00000	224	32	net ID (goes in routing table)
	111/00001	225		first host ID in All 1s subnet
	111/11110	251		last host ID in All 1s subnt
	111/11111	255		local wire (all subnets) broadcast

Power of 2 Shortcuts for Calculating Subnets

Networks Gained from Using Host Bits

Where *n* is equal to the number of borrowed host bits used for subnets, 2^n = the total number of networks gained.

Hosts Available after Subnetting

Where *h* is equal to the number of host bits remaining after subnetting, $2^h - 2$ = the total number of hosts available.

Example

You need to subnet the Class C network address 193.5.15.0 to obtain 5 additional networks, and you need to find out how many hosts will be available after subnetting.

There are 8 host bits for a Class C address. Beginning with borrowing 1 host bit, calculate the number of networks that would be gained until you find a number of bits that yields at least five networks.

- 1 host bit: $2^1 = 2$ networks
- 2 host bits: $2^2 = 4$ networks
- 3 host bits: $2^3 = 8$ networks

From the calculations, you can see that borrowing 2 host bits will only yield 4networks; you would not obtain the 5 networks that you need. However, borrowing 3 host bits will yield 8 networks, which will provide the 5 networks that you need.

Of the 8 host bits for a Class C address, you have allocated 3 of those for subnetting, thus leaving 5 host bits available for their original purpose. As a result, $2^5 - 2$, or 30, hosts can be supported on each subnet.

Class A

(24 bits available for subnetting)

Table 13

Bits Required (*n*)	Subnets (2^2)	Hosts Per Subnet ($2^h - 2$)	Slash Notation	Mask	Subnet Slice
1	$2^1 = 2$	$2^{23} - 2 = 8,388,606$	/9	255.128.0.0	Subnet 1/8
2	$2^2 = 4$	$2^{22} - 2 = 4,194,302$	/10	255.192.0.0	Subnet 2/8
3	$2^3 = 8$	$2^{21} - 2 = 2,097,150$	/11	255.224.0.0	Subnet 3/8
4	$2^4 = 16$	$2^{20} - 2 = 1,048,574$	/12	255.240.0.0	Subnet 4/8
5	$2^5 = 32$	$2^{19} - 2 = 524,286$	/13	255.248.0.0	Subnet 5/8
6	$2^6 = 64$	$2^{18} - 2 = 262,142$	/14	255.252.0.0	Subnet 6/8
7	$2^7 = 128$	$2^{17} - 2 = 131,070$	/15	255.254.0.0	Subnet 7/8
8	$2^8 = 256$	$2^{16} - 2 = 65,534$	/16	255.255.0.0	Subnet 8/8

Class B

(16 bits available for subnetting)

Table 14

Bits Required (n)	Subnets (2^2)	Hosts Per Subnet (2^h –2)	Slash Notation	Mask	Subnet Slice
1	$2^1 = 2$	$2^{15} - 2 = 32{,}766$	/17	255.255.128.0	Subnet 1/8
2	$2^2 = 4$	$2^{14} - 2 = 16{,}382$	/18	255.255.192.0	Subnet 2/8
3	$2^3 = 8$	$2^{13} - 2 = 8{,}190$	/19	255.255.224.0	Subnet 3/8
4	$2^4 = 16$	$2^{12} - 2 = 4{,}094$	/20	255.255.240.0	Subnet 4/8
5	$2^5 = 32$	$2^{11} - 2 = 2{,}046$	/21	255.255.248.0	Subnet 5/8
6	$2^6 = 64$	$2^{10} - 2 = 1{,}022$	/22	255.255.252.0	Subnet 6/8
7	$2^7 = 128$	$2^9 - 2 = 510$	/23	255.255.254.0	Subnet 7/8
8	$2^8 = 256$	$2^8 - 2 = 254$	/24	255.255.255.0	Subnet 8/8

Class C

(8 bits available for subnetting)

Table 15

Bits Required (n)	Subnets (2^2)	Hosts Per Subnet (2^h –2)	Slash Notation	Mask	Subnet Slice
1	$2^1 = 2$	$2^7 - 2 = 126$	/25	255.255.255.128	Subnet 1/6
2	$2^2 = 4$	$2^6 - 2 = 62$	/26	255.255.255.192	Subnet 2/6
3	$2^3 = 8$	$2^5 - 2 = 30$	/27	255.255.255.224	Subnet 3/6
4	$2^4 = 16$	$2^4 - 2 = 14$	/28	255.255.255.240	Subnet 4/6
5	$2^5 = 32$	$2^3 - 2 = 6$	/29	255.255.255.248	Subnet 5/6
6	$2^6 = 64$	$2^2 - 2 = 2$	/30	255.255.255.252	Subnet 6/6

Practice

You have the address 132.7.0.0. You need 5 subnets of equal size with 1,500 hosts each. Find the following information in the given order:

1. the number of host bits to obtain for the required number of subnets
2. the numbers of hosts available per subnet after subnetting
3. the decimal value of the new subnet mask
4. the incremental value of the subnets
5. the first host, broadcast, and last host of each subnet

---

1. Using powers of 2, determine the number of host bits to obtain for 5 subnets.
   - The first octet is 132, which is within the Class B range (128 – 191). Recall that each octet has 8 bits.
   - The first two octets in a Class B address may not be used for subnetting, so there are two octets, or 16 bits, available.
   - Beginning at the left side of the available bits and moving toward the right, remove one bit at a time from the host portion of the binary and calculate the number of networks gained with each bit.

     2^1 = 2
     2^2 = 4 (Not enough networks; keep trying.)
     2^3 = 8 (3 bits will provide the 5 subnets and leave 1 available for future expansion.)
   - Enter those 3 bits into a weighted values chart, as shown in Table 16 below.

Table 16

	Most Significant Bit						Least Significant Bit	Hosts
Weighted Values	128	64	32	16	8	4	2	1	
Binary	1	1	1	0	0	0	0	0	

2. Using powers of 2, determine how many hosts will be available after subnetting.
   - Recall that there are 16 host bits available for a Class B address. You will use 3 of those for subnetting, thus leaving 13 host bits available to use for hosts.

     $2^{13} - 2$ = 8,190 hosts available per subnet

     **If you prefer to have extra subnets available for future expansion over having extra host capacity, you can use additional host bits now.*

3. Using the weighted values chart created in step 1, determine the new decimal value of the subnet mask.
   - Add together each weighted value with a 1 below it.

     128 + 64 + 32 = 224
   - Since this is a Class B address and the first two octets are reserved, the default subnet mask is 255.255.0.0.
   - The 3 host bits were removed from the third octet, so the new decimal mask is 255.255.224.0, or /19 in slash notation (8 bits + 8 bits + 3 bits = 19).

4. Using the decimal mask or the last subnet bit's weighted value, determine the incremental value of the subnets.
   - Subtract the decimal value calculated in step 3 from 256 to obtain the incremental value.

     256 – 224 = 32

     OR
   - Look at Table 16 from step 1, and find the last weighted value toward the right with a 1 below it. That value, 32, is the incremental value.

5. Use the incremental value to find the first host IP, broadcast, and last host IP for each subnet
   - Begin with the incremental value, which is 32 in this scenario, and add 1 to obtain the first host IP for subnet 1.

     32 + 1 = 33 (first host IP for subnet 1)
   - Add the current subnet's incremental value with the incremental value and subtract 1 to obtain the broadcast for subnet 1.

     32 + 32 – 1 = 63
   - Subtract 1 from the subnet 1's broadcast to obtain the last host IP for subnet 1.

     63 – 1 = 62
   - Repeat this process for each subsequent subnet, as shown in Table 17 below.

Table 17

Range	Subnet	Incremental Value (subnet ID used by routing table)	First Host IP	Broadcast	Last Host IP
0 × 32	0	0	1	31	30
1 × 32	1	32	33	63	62
2 × 32	2	64	65	95	94
3 × 32	3	96	97	127	126
4 × 32	4	128	129	159	158
5 × 32	5	160	161	191	190
6 × 32	6	192	193	223	222
7 × 32	7	224	225	255	254

Printable Study Cards

Bit Conversions		
Bits	Binary	Decimal Mask
1	10000000	128
2	11000000	192
3	11100000	224
4	11110000	240
5	11111000	248
6	11111100	252
7	11111110	254
8	11111111	255

Boson

Bit Conversions (Class C)		
Bits	Subnets	Hosts
1	$2^1 = 2$	$2^7 - 2 = 126$
2	$2^2 = 4$	$2^6 - 2 = 62$
3	$2^3 = 8$	$2^5 - 2 = 30$
4	$2^4 = 16$	$2^4 - 2 = 14$
5	$2^5 = 32$	$2^3 - 2 = 6$
6	$2^6 = 64$	$2^2 - 2 = 2$
7	$2^7 = 128$	$2^1 - 1 = 1$*
8	$2^8 = 256$	$2^0 - 1 = 0$*

Boson

Incremental Value Per Mask	
Decimal Mask	Incremental Value
128	128
192	64
224	32
240	16
248	8
252	4
254	2
255	1

Boson

Address Class Ranges		
Class	Range	Network ID
A	1 – 126	0
B	128 – 191	10
C	192 – 223	110

Boson

Calculating Binary and Decimal Masks

	Most Significant Bit							Least Significant Bit
Weighted Values	128	64	32	16	8	4	2	1
Binary	1	1	1	1	1	1	1	1

Boson

The table above depicts the decimal mask 255 in binary. The equation is 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.