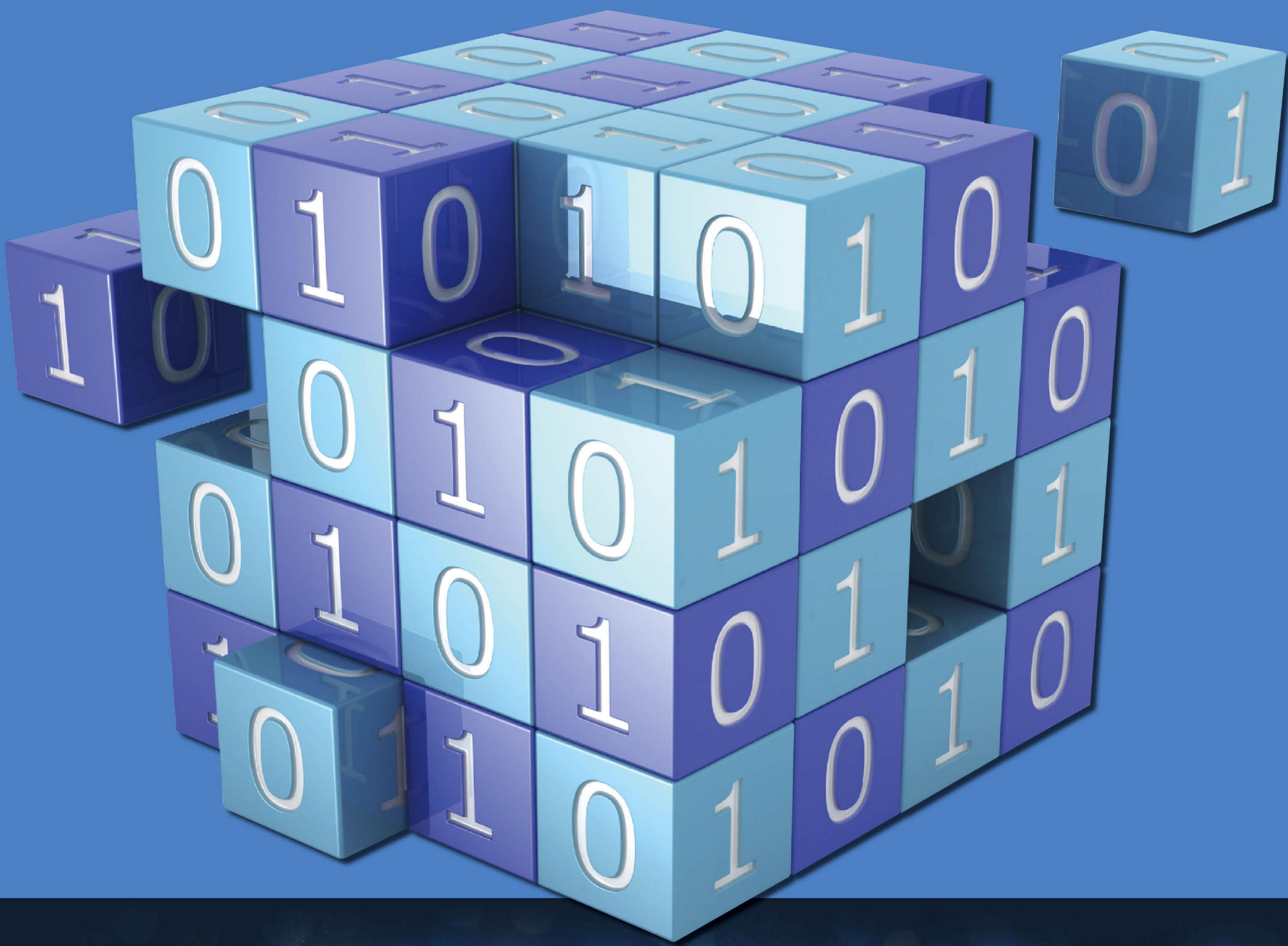
how to master SUBNETTING



René Molenaar

Introduction

Binary and hexadecimal numbers are a complete mystery for many of us. Often we don't find it really interesting because on the internet there are plenty of "subnet" or "binary" calculators where you can easily calculate from decimal to binary to hexadecimal or the other way around, without knowing how the exact calculation works.

This is no problem when you are not configuring or designing networks on a daily basis, but it will be a problem as soon as you take a networking exam, so it's best to know how to do these calculations off the top of your head.

Another advantage you will have is once you have mastered the art of binary calculations you can immediately "see" how big a network is and what the subnet mask is when people start throwing numbers at you.

One of the things I do in life is work as a Cisco Certified System Instructor (CCSI) and I noticed many people have trouble finding out what the subnet mask is, how many hosts are in a subnet, how to do summarization and so they fail at passing exams like CCNA or CCNP. Anyone working with networks on a professional level should be able to do binary calculations if you ask me.

This book will teach you how to calculate subnets and subnet masks, how to calculate the numbers of hosts available etc. for class A,B and C networks. And the best part: **You will be able to do this off the top of your head, no need to write stuff down!** Once you have mastered the tricks in this book you will wonder why you ever had difficulty solving subnetting questions :)

Enjoy reading my book and good luck mastering your binary and subnetting skills!

René Molenaar

P.S. There are 10 types of people in the world: Those that understand ***binary***, and those who don't!

P.P.S. If you have any questions or comments about this book, please let me know:

E-mail: info@gns3vault.com
Website: gns3vault.com
Facebook: facebook.com/gns3vault
Twitter: twitter.com/gns3vault
Youtube: youtube.com/gns3vault

Index

1. Binary Basics

Before we start calculating subnets and talk about IP addressing, let's first check out some basics of binary calculations. We are all used to work with decimal numbers where we count from 1 till 10. This is easy because we have 10 fingers so we don't have to count off the top of our head.

In the binary system, we only work with 0 or 1.

0 = Off
1 = On

Bits	128	64	32	16	8	4	2	1

The bit on the far left side is called the most significant bit (MSB) because this bit has the highest value. The bit on the far right side is called the least significant bit (LSB) because this one has the lowest value.

So how do we convert decimal numbers into binary? Let me show you an example:

If we want the decimal number "0" in binary this means we leave all the bits "off".

Bits	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0

Let's take the number 178 and turn it into binary, just start at the left and see which bits "fit in" to make this number. 128 + 32 + 16 + 2 = 178.

Bits	128	64	32	16	8	4	2	1
178	1	0	1	1	0	0	1	0

Just one more! Let's turn 255 into binary. 128 + 64 + 32 + 16 + 8 + 4 + 2 +1 = 255

Bits	128	64	32	16	8	4	2	1
255	1	1	1	1	1	1	1	1

As you can see 255 is the highest decimal number you can create when you have 8 bits to play with.

As you can see, whenever you add a bit, the decimal value doubles. For example: 2,4,8,16,32,64,128,256,512,1024,2048 and so on. This is called the "powers of 2".

This is a good moment to create your own "cheat sheet" . Take a piece of paper and write down the 8 bits for yourself.

Exercise 1:

See if you can solve the following decimal to binary calculations:

Bits	128	64	32	16	8	4	2	1
12								
54								
187								
192								
44								
147								

Now try to do it the other way around and calculate from binary to decimal:

Bits	128	64	32	16	8	4	2	1
	1	1	0	0	1	0	1	0
	0	0	1	1	1	0	0	1
	0	1	0	1	0	1	0	1
	1	1	1	1	0	0	1	1
	0	0	1	0	0	0	0	1
	1	0	0	0	0	1	1	1

The appendix of this book will show you the answers.

2. Welcome to Subnetting

Before we start calculating subnets, the first thing we need to do is take a look at what subnets and IP addresses are.

An IP address is a numeric value that you configure on every device in a network, think about computers, laptops, servers but also networking equipment like routers, firewalls and switches. The IP address identifies every device with a "unique" number. Devices within the same IP subnet are able to communicate without using a router.

Let's take a look at some of the terminology you might encounter when we talk about IP addresses:

IP Terminology:

Bit(s)	A bit has 2 possible values, 1 or 0. (on or off)
Byte	A byte is 8 bits.
Octet	An octet is just like a byte 8 bits, you often see byte or octet both being used.
Nibble	A nibble is 4 bits, we'll talk about this more in the Hexadecimal chapter.
Network address	When we talk about routing, the network address is important. Routers use the network address to send IP packets to the right destination. 192.168.1.0 with subnet mask 255.255.255.0 is an example of a network address.
Subnet	A subnet is a network that you split up in multiple smaller subnetworks.
Broadcast address	The broadcast address is being used by applications and computers to send information to all devices within a subnet, 192.168.1.255 with subnet mask 255.255.255.0 is an example of a broadcast address.

Hierarchical IP addressing:

IP addresses are 32 bits, divided in 4 "blocks" also known as 4 bytes or octets. Every byte has 8 bits. 4x8 = 32 bits.

There are many ways to write down an IP address:

Decimal:	192.168.1.1
Binary:	11000000.10101000.0000001.0000001
Hexadecimal:	C0.A8.01.01

Decimal is what we are used to work with, as this is the way you normally configure an IP address in operating systems like Microsoft Windows, Linux or most networking equipment. Hexadecimal you won't see often but for example you might encounter this in the windows registry.

IP addresses are hierarchical unlike non-hierarchical addresses like MAC-addresses. This has some advantages, you can use a lot of IP addresses (with 32 bits the biggest number you can create is 4,3 billion or to be precise 4,294,967,296). The advantage of having a hierarchical model is needed for routing, imagine that every router on the planet would need to know every IP address on the planet. Routing wouldn't be very efficient that way…

A better solution is a hierarchical model where we use "network", "subnet" and "hosts".

Try to compare this to phone numbers:

0031	This is the country code for The Netherlands
013	This is the city code for Tilburg
1234567	This is a single number for a customer.

The complete phone number is 0031-013-1234567.

IP addresses use a similar hierarchical structure.

Network addresses:

Every subnet has 1 network address!

The network address is a unique identification of the network. Every device within the same subnet shares this network address in its IP address, for example:

192.168.100.1
192.168.100.2
192.168.100.3

192.168.100. is the network address and .1, .2 and .3 are host addresses. The IP address will tell you in what subnet they are located. The network address has to be the same for all the hosts; the host part has to be unique. When the Internet was invented they created different "classes" of networks each with a different size. At this moment there are 3 classes that are important to us:

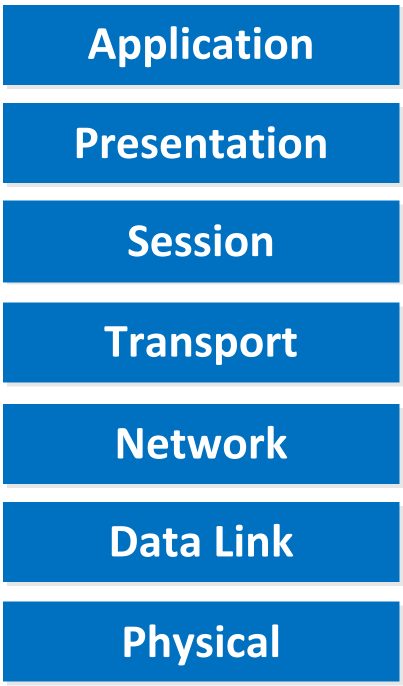
	8 bits	8 bits	8 bits	8 bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

Broadcast addresses:

Every subnet has 1 broadcast address!

When we talk about broadcasts in IP world, we talk about layer3 broadcasts. In case you have no idea what I'm talking about…take a look at the OSI model:

OSI Model



The OSI model describes a layered approach of a network, getting into the details of all the different layers of the OSI model is outside the scope of the book, but to get an understanding of broadcasts it's important to look at layer 2 and layer 3.

There's a layer2 and layer3 broadcast, and there's a big difference between them. When we look at a LAN (Local Area Network) we are probably using Ethernet. MAC addresses are used to uniquely identify a network device, for example: 00:50:56:c0:00:08 is a MAC address that uniquely identifies my computer. On a LAN it's possible to send a layer 2 broadcast so that all computers on the LAN segment will receive this message (Ethernet Frame). The destination MAC address would be FF:FF:FF:FF:FF:FF (when you read the hexadecimal chapter you'll see that FF:FF:FF:FF:FF:FF is a string with only 1's in binary).

Now let's take a look at a layer 3 broadcast. Layer 3 is where we talk about IP addressing, and we can also send a broadcast. For example take the 192.168.1.0 network.

192.168.1.255 for this subnet is the broadcast address, this means when we send an IP packet to 192.168.1.255 that all hosts on this subnet will receive this packet. Pretty neat right? Some old applications might still use this form of communication.

Class A:

Back to our network addresses, let's take a look at Class A. The first bit always has to be a 0. This leaves us 7 bits to "play" with. The lowest value you can create by changing all bits to "0" is 0. By changing all 7 bits to "1" you get 127.

Bits	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0
127	0	1	1	1	1	1	1	1

64 + 32 + 16 + 8 + 4 + 2 + 1 = 127.

As you can see the Class A range is between 0. and 127.

Class B:

For a class B network the first bit has to be a 1. The second bit has to be a 0.

Bits	128	64	32	16	8	4	2	1
128	1	0	0	0	0	0	0	0
191	1	0	1	1	1	1	1	1

128 + 32 + 16 + 8 + 4 + 2 +1 = 191

As you can see class B networks always start with 128. and the last network is 191.

Class C:

For a class C network the first bit has to be a 1, the second bit a 1 and the third a 0.

Bits	128	64	32	16	8	4	2	1
192	1	1	0	0	0	0	0	0
223	1	1	0	1	1	1	1	1

128 + 64 = 192

128 + 64 + 16 + 8 + 4 + 2 + 1 = 223

As you can see Class B networks start at 192. and the last network is 223.

Class D and E:

There is also a class D for multicast traffic which starts at 224. and ends at 239. Class E is for "research usage". We are not going to use these classes for our binary calculations.

Class A Addressing:

A class A network has 1 byte reserved for the network address which means the other 3 bytes are left for hosts. This means we have a couple of networks and every network can have a lot of hosts (how to determine how many hosts each network has we will see later!).

Byte	Byte	Byte	Byte
Network	**Hosts**	**Hosts**	**Hosts**

If we look at the IP address 53.21.43.63 then "53" is the network address and "21.43.63" is the host address, all machines on this subnet will have the "53" as network address.

Byte	Byte	Byte	Byte
Network	**Hosts**	**Hosts**	**Hosts**
53.	**21.**	**43.**	**63**

Class B Addressing:

A class B network has 2 bytes reserved for the network address which means the other 2 bytes are left for hosts. This means we have even more networks but less hosts per network compared to class A.

Byte	Byte	Byte	Byte
Network	**Network**	**Hosts**	**Hosts**

For example, 172.16.100.68, the network address is 172.16. and the host address is 100.68.

Byte	Byte	Byte	Byte
Network	**Network**	**Hosts**	**Hosts**
172.	**16.**	**100.**	**68**

Class C Addressing:

A class C network has 3 bytes reserved for the network address which means the other byte is left for hosts. Now we have a lot of networks but only a few hosts per network.

Byte	Byte	Byte	Byte
Network	**Network**	**Network**	**Hosts**

Another example, 192.168.200.53, the network address is 192.168.200. and the host address is .53.

Byte	Byte	Byte	Byte
Network	**Network**	**Network**	**Hosts**
192.	**168.**	**200.**	**53**

Private IP addressing

There is a difference between public and private IP addresses. The people who invented the IP addressing scheme decided it would be a good idea to have a range of networks that are not routable on the internet. Now this isn't entirely true, I should say "should not be routed on the internet". It's up to the service providers to filter these networks.

If every device on the planet would require a unique IP address then we would have already run out of address space by now. Instead, there are some private ranges you can use for your internal networks and these are not accessible from the internet. Now perhaps you are wondering why you are able to access the internet from your home computer?

The answer to this question is that you have 1 public IP address that you got from your internet provider, and all your home computers have private IP addresses. Your router runs NAT (Network address Translation) and makes sure all private IP addresses will be translated to your single public IP address. This way all computers can access the internet by using a single private IP address! (and we can all browse/surf the internet all day long…)

These are the Private IP address ranges:

Class A:	10.0.0.0 – 10.255.255.255
Class B:	172.16.0.0 – 172.31.255.255
Class C:	192.168.0.0 – 192.168.255.255

If you made it through this chapter and you understand everything….very good! When in doubt please reread this chapter since it's important you understand everything before continuing, since we are going to start calculating subnets…ready? Let's go!

3. Subnetting: The beginning

Let's take a Class C network and take a good look at it, so we can play around with binary numbers.

For example: 192.168.1.0

In binary it looks like this:

	192	168	1	0
IP address	**11000000**	10101000	00000001	00000000

In the previous chapter I explained that a class C network consists of 3 bytes for the network part, and one byte for hosts:

Byte	Byte	Byte	Byte
Network	**Network**	**Network**	**Hosts**
192.	**168.**	**1.**	**0**

Now the question is…how does a network device know which part is the network-part, and which side is the host-part? Is it because it's a Class C network? Is it some secret rule that everyone just knows about?

The answer is no, we use something called a subnet mask! For this network, it would be the following subnet mask:

255.255.255.0

Now what does this subnet mask exactly do? The word "mask" might tell you that it must mean that it's hiding something…but that is not the case, and to show you the answer we have to look at some binary numbers:

IP address (decimal)	192	168	1	0
IP address (binary)	11000000	10101000	00000001	00000000
Subnet mask (decimal)	**255**	**255**	**255**	**0**
Subnet mask (binary)	11111111	11111111	11111111	00000000

The subnet mask will specify which part of the IP address is the network-part and which part is the host-part. The 1 means it's the network-part, the 0 means the host-part.

To clarify this let me just take the binary numbers, the subnet mask tells you the first 24 bits are the network-address and the 8 bits that are left we can use for hosts.

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

For our 192.168.1.0 example this means 24 bits are reserved for network and 8 bits are reserved for hosts.

Let's write down those 8 host-bits:

	128	64	32	16	8	4	2	1

What's the highest value you can create with 8 bits? Let's have a look:

128 + 64 + 32 + 16 + 8 + 4 + 2 +1 = 255

	128	64	32	16	8	4	2	1
255	1	1	1	1	1	1	1	1

Cool! So now we know that with 8 bits the highest value we can create is 255, does this mean we can have 255 hosts in this network? The answer is no because for every network there are 2 addresses we can't use:

1) Network address: this is the address where all the host bits are set to 0.

IP address	192	168	1	0
	11000000	10101000	00000001	00000000

2) Broadcast address: this is the address where all the host bits are set to 1.

IP address	192	168	1	255
	11000000	10101000	00000001	11111111

*The network address has all hosts bits set to **0**!*
*The broadcast address has all host bits set to **1**!*

Alright so let's take 255 – 2 = 253. Does this mean we can have a maximum of 253 hosts on our network?

The answer is still no! I messed with your head because the highest value you can create with 8 bits is not 255 but 256. Why? Because you can also use a value of "0".

Does this make your head spin? Let's take a look at our 192.168.1.0 network in binary:

IP address	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000
Subnet mask	**255**	**255**	**255**	**0**
	11111111	11111111	11111111	00000000

Network	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000

Broadcast	**192**	**168**	**1**	**255**
	11000000	10101000	00000001	11111111

The network address has all host bits set 0, so in decimal this is 0.
The broadcast address has all host bits set to 1, so in decimal this is 255.

This means everything in between we can use for hosts, 1 – 254 so that's 254 valid IP addresses we can use to configure hosts!

Don't start counting at "1", but start counting at "0". The "0" is a valid number.

Great! So now you have seen what a network looks like in binary, what the subnet mask does, what the network and broadcast addresses are and that we can fit in 254 hosts in this Class C network.

Now let's say I don't want to have a single network where I can fit In 254 hosts, but I want to have 2 networks? Is this possible? It sure is! Basically what we are doing is taking a Class C network and chop it in 2 pieces, and this is what we call subnetting. Let's take a look at it in binary:

IP address	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000
Subnet mask	**255**	**255**	**255**	**0**
	11111111	11111111	11111111	00000000

If we want to create more subnets we need to borrow bits from the host-part. For every bit you borrow you can double the number of subnets, by borrowing 1 bit we create 2 subnets out of this single network. There are 8 host-bits so if we steal one to create more subnets this means we have only 7 bits left for hosts.

What will the new subnet mask be? Let's take a look at it in binary:

Subnet mask	255	255	255	128
	11111111	11111111	11111111	10000000

The first 24 bits are the same so we only have to look at the 4th octet, let's write down those bits:

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0

Calculate it back to decimal and you'll have 128. The subnet mask will be 255.255.255.128.

The second question is, how "big" are these 2 subnets and how many hosts can we fit in?

128	64	32	16	8	4	2	1
N/A	1	1	1	1	1	1	1

We have 7 bits left so let's do the binary to decimal calculation:

64 + 32 + 16 + 8 + 4 + 2 +1 = 127.

Don't forget about the 0! Because we can use the 0 the highest value we can create with 7 bits is 128.

Our original class C network has now been divided in 2 subnets with a size of 128 each. So what will the network addresses of the 2 new subnets be? Let's work this example out in binary:

Subnet #1:

By applying the new subnet mask we only have **7 host bits** to play with.

192.168.1.0
255.255.255.128

IP address	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000
Subnet mask	**255**	**255**	**255**	**128**
	11111111	11111111	11111111	10000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.0

Network	192	168	1	0
	11000000	10101000	00000001	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.1

Network	192	168	1	1
	11000000	10101000	00000001	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.126

Network	192	168	1	126
	11000000	10101000	00000001	01111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.127

Broadcast	192	168	1	127
	11000000	10101000	00000001	01111111

Subnet #2:

The first subnet ended at 192.168.1.127 so we just continue with the next subnet at 192.168.1.128:

192.168.1.128
255.255.255.128

IP address	192	168	1	128
	11000000	10101000	00000001	10000000
Subnet mask	**255**	**255**	**255**	**128**
	11111111	11111111	11111111	10000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.128

Network	192	168	1	128
	11000000	10101000	00000001	10000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.129

Network	192	168	1	129
	11000000	10101000	00000001	10000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.254

Network	192	168	1	254
	11000000	10101000	00000001	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.255

Broadcast	192	168	1	255
	11000000	10101000	00000001	11111111

That's it! That's the first network we just subnetted in 2 subnets and we found out what the network and broadcast addresses are, and what IP addresses we can use for hosts.

Let me show you another one, we take the same Class C 192.168.1.0 network but now we want to have 4 subnets. For every host-bit we borrow we can double the number of subnets we can create, so by borrowing 2 host bits we can create 4 subnets.

Every "host-bit" you "borrow" doubles the number of subnets you can create.

What will the new subnet mask be? Let's take a look at it in binary:

Subnet mask	255	255	255	192
	11111111	11111111	11111111	11000000

Calculate it from binary to decimal: 128+64 = 192.

The new subnet mask will be 255.255.255.192

Subnet #1:

By applying the new subnet mask we only have **6 host bits** to play with.

192.168.1.0
255.255.255.192

IP address	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000
Subnet mask	**255**	**255**	**255**	**192**
	11111111	11111111	11111111	11000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.0

Network	**192**	**168**	**1**	**0**
	11000000	10101000	00000001	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.1

Network	**192**	**168**	**1**	**1**
	11000000	10101000	00000001	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.62

Network	**192**	**168**	**1**	**62**
	11000000	10101000	00000001	00111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.63

Broadcast	**192**	**168**	**1**	**63**
	11000000	10101000	00000001	00111111

Subnet #2:

The first subnet ended at 192.168.1.63 so we just continue with the next subnet at 192.168.1.64:

192.168.1.64
255.255.255.192

IP address	192	168	1	64
	11000000	10101000	00000001	01000000
Subnet mask	**255**	**255**	**255**	**192**
	11111111	11111111	11111111	11000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.64

Network	192	168	1	64
	11000000	10101000	00000001	01000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.65

Network	192	168	1	65
	11000000	10101000	00000001	01000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.126

Network	192	168	1	126
	11000000	10101000	00000001	01111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.127

Broadcast	192	168	1	127
	11000000	10101000	00000001	01111111

Subnet #3:

The second subnet ended at 192.168.1.127 so we just continue with the next subnet at 192.168.1.128:

192.168.1.128
255.255.255.192

IP address	192	168	1	128
	11000000	10101000	00000001	10000000
Subnet mask	**255**	**255**	**255**	**192**
	11111111	11111111	11111111	10000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.128

Network	192	168	1	128
	11000000	10101000	00000001	10000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.129

Network	192	168	1	129
	11000000	10101000	00000001	10000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.190

Network	192	168	1	190
	11000000	10101000	00000001	10111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.191

Broadcast	192	168	1	191
	11000000	10101000	00000001	10111111

Subnet #4:

The second subnet ended at 192.168.1.191 so we just continue with the next subnet at 192.168.1.192:

192.168.1.192
255.255.255.192

IP address	192	168	1	192
	11000000	10101000	00000001	11000000
Subnet mask	**255**	**255**	**255**	**192**
	11111111	11111111	11111111	11000000

Network address:

The network address has all host bits set to 0, so the network address will be: 192.168.1.192

Network	192	168	1	192
	11000000	10101000	00000001	11000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 192.168.1.193

Network	192	168	1	193
	11000000	10101000	00000001	11000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 192.168.1.254

Network	192	168	1	254
	11000000	10101000	00000001	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 192.168.1.255

Broadcast	192	168	1	255
	11000000	10101000	00000001	11111111

There we go! We just chopped down our 192.168.1.0 class C network into 4 subnets! If you understand everything up to this point...great job! Does this look like a lot of work? Honestly...yes it is!

I promise you to show you some tricks to calculate Class C,B and even A subnets without touching any binary numbers….and even better, you don't have to write stuff down you can do it off the top of your head.

The reason I don't show you this right away is that you need to understand what is happening "under the engine" before you can apply the fast tricks.

Exercise 2:

Now it's time for you to calculate some subnets, see if you can solve the following questions:

1. Take the 192.168.1.0 Class C network and create 8 subnets out of it. Write down the following information:
   a. The first 2 subnets.
   b. The network addresses.
   c. The broadcast addresses.
   d. The usable host IP addresses.
2. Take the 192.168.1.0 Class C network and create 16 subnets out of it. Write down the following information:
   a. The first 2 subnets.
   b. The network addresses.
   c. The broadcast addresses.
   d. The usable host IP addresses.

The appendix of this book will show you the answers.
Okay so we have played enough with Class C networks, let's try a Class B network. You'll see that it's exactly the same thing.

Let's take the 172.16.100.0 Class B network with subnet mask 255.255.0.0 and create 2 subnets out of it:

IP address	**172**	**16**	**100**	**0**
	11000000	00010000	01100100	00000000
Subnet mask	**255**	**255**	**0**	**0**
	11111111	11111111	00000000	00000000

If we want to create more subnets we need to borrow bits from the host-part. For every bit you borrow you can double the number of subnets, by borrowing 1 bit we create 2 subnets out of this single network. Now the difference with a Class C network is that we have more host-bits to play with, that's all.

What will the new subnet mask be? Let's take a look at it in binary:

Subnet mask	255	255	128	0
	11111111	11111111	10000000	00000000

As you can see the net subnet mask will be 255.255.128.0 and we have 7+8 = 15 host bits left to play with.

How "big" are these 2 subnets? Well we have 15 bits so let's take a look:

	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
N/A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 +1 = 32767.

Don't forget about the 0! So the highest value you can create with 15 bits is **32768**.

If you want to know to know how many usable host IP addresses you have, you take 32768 – 2 (because of the network and broadcast address).

32768 – 2 = 32766 usable host IP addresses. That's a lot of computers/laptops/servers!

A much faster way to calculate this is by using the "powers of 2" that I explained earlier:

2 to the power of 15 (or 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2) = 32768.

32768 minus 2 (network + broadcast address) = 32766.

Does this make sense to you? Good! My promise is still standing…I will show you how to solve these subnetting questions without touching any binary, you just need to make sure you understand the math that is going on first.

Let's calculate what the subnets look like.

<u>Subnet #1:</u>

By applying the new subnet mask we only have **15 host bits** to play with.

172.16.0.0
255.255.128.0

IP address	172	16	0	0
	10101100	0001000	00000000	00000000
Subnet mask	**255**	**255**	**128**	**0**
	11111111	11111111	10000000	00000000

Network address:

The network address has all host bits set to 0, so the network address will be: 172.16.0.0

Network	172	16	0	0
	10101100	0001000	00000000	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 172.16.0.1

Network	172	16	0	1
	10101100	0001000	00000000	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 172.16.127.254

Network	172	16	127	254
	10101100	0001000	01111111	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 172.16.127.255

Network	172	16	127	255
	10101100	0001000	01111111	11111111

<u>Subnet #2:</u>

The first subnet ended at 172.16.127.255 so we just continue with the next subnet at 172.16.128.0:

172.16.128.0
255.255.128.0

IP address	172	16	128	0
	10101100	0001000	10000000	00000000
Subnet mask	**255**	**255**	**128**	**0**
	11111111	11111111	10000000	00000000

Network address:

The network address has all host bits set to 0, so the network address will be: 172.16.128.0

Network	172	16	128	0
	10101100	0001000	10000000	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 172.16.128.1

Network	172	16	128	1
	10101100	0001000	10000000	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 172.16.255.254

Network	172	16	255	254
	10101100	0001000	11111111	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 172.16.255.255

Network	172	16	255	0
	10101100	0001000	11111111	11111111

Alright so we just subnetted this 172.16.0.0 class B network into 2 subnets, you are doing the exact same thing but now you have more bits to play with…

Exercise 3:

Now see if you can solve these questions:

1. Take the 172.16.0.0 Class B network and create 4 subnets out of it. Write down the following information:
    a. The first 3 subnets.
    b. The network addresses
    c. The broadcast addresses
    d. The usable host IP addresses.
2. Take the 172.16.0.0 Class B network and create 128 subnets out of it. Write down the following information:
    a. The first 4 subnets
    b. The network addresses
    c. The broadcast addresses
    d. The usable host IP addresses.

The appendix of this book will show you the answers.

So subnetting a class B network wasn't that hard right? Let's try a Class A network and see what happens:

Let's take the 10.0.0.0 Class A network with subnet mask 255.0.0.0 and create at least 12 subnets out of it:

IP address	10	0	0	0
	00001010	00000000	00000000	00000000
Subnet mask	**255**	**0**	**0**	**0**
	11111111	00000000	00000000	00000000

If we want to create more subnets we need to borrow bits from the host-part. For every bit you borrow you can double the number of subnets (remember the "powers of 2" ?), by borrowing 4 bits we can create 16 subnets out of this single network. 3 bits would not be enough because we can only create 8 subnets then.

What will the new subnet mask be? Let's take a look at it in binary:

Subnet mask	255	240	0	0
	11111111	11110000	00000000	00000000

As you can see the subnet mask will be 255.240.0.0 and we have 4+8+8 = 20 host bits left to play with.

How "big" are these 16 subnets? Well we have 20 bits so let's just use the "powers of 2" to solve this question:

2 to the power of 20 = 1.048.576

If you want to know to know how many usable host IP addresses you have, you take 1.048.576– 2 (because of the network and broadcast address).

1.048.576– 2 = 1.048.574 usable host IP addresses. That's lots and lots of computers/laptops/servers!

Let's calculate what the subnets look like, I'm not going to do all of them, just 3 of them. By now you should be familiar what the math looks like.

Subnet #1:

By applying the new subnet mask we only have **19 host bits** to play with.

10.0.0.0
255.240.0.0

IP address	10	0	0	0
	00001010	00000000	00000000	00000000
Subnet mask	**255**	**240**	**0**	**0**
	11111111	11110000	00000000	00000000

Network address:

The network address has all host bits set to 0, so the network address will be: 10.0.0.0

Network	10	0	0	0
	00001010	00000000	00000000	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 10.0.0.1

Network	10	0	0	1
	00001010	00000000	00000000	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 10.15.255.254

Network	10	15	255	254
	00001010	00001111	11111111	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 10.15.255.255

Network	10	15	255	255
	00001010	00001111	11111111	11111111

Subnet #2:

The broadcast address of Subnet #1 was 10.15.255.255 so our next subnet starts at 10.16.0.0

10.16.0.0
255.240.0.0

IP address	10	16	0	0
	00001010	00001000	00000000	00000000
Subnet mask	**255**	**240**	**0**	**0**
	11111111	11110000	00000000	00000000

Network address:

The network address has all host bits set to 0, so the network address will be: 10.16.0.0

Network	10	16	0	0
	00001010	00010000	00000000	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 10.16.0.1

Network	10	16	0	1
	00001010	00010000	00000000	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 10.15.255.254

Network	10	31	255	254
	00001010	00011111	11111111	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 10.31.255.255

Network	10	31	255	255
	00001010	00011111	11111111	11111111

Subnet #3:

The broadcast address of Subnet #2 was 10.31.255.255 so our next subnet starts at 10.16.0.0

10.32.0.0
255.240.0.0

IP address	10	32	0	0
	00001010	00010000	00000000	00000000
Subnet mask	255	240	0	0
	11111111	11110000	00000000	00000000

Network address:

The network address has all host bits set to 0, so the network address will be: 10.32.0.0

Network	10	32	0	0
	00001010	00010000	00000000	00000000

First usable host IP address:

The first usable host IP address is the one that comes after the network address, so this will be: 10.32.0.1.

Network	10	32	0	1
	00001010	00010000	00000000	00000001

Last usable host IP address:

The last IP address we can use for a host is the one before the broadcast address, so this will be: 10.47.255.254

Network	10	47	255	254
	00001010	00101111	11111111	11111110

Broadcast address:

The broadcast address has all host bits set to 1 so the broadcast address we get is: 10.47.255.255

Network	10	47	255	255
	00001010	00101111	11111111	11111111

Alright so that's subnetting a Class A network! I showed you how to do all of this in binary and by now you should have a good understanding how it works "under the engine". In the next chapter I'll show you how to do subnetting a whole lot faster, and even off the top of your head!

4. Subnetting: The Fast Way

You have probably seen enough binary numbers now, so let's work some more with decimal numbers. We can do subnetting just by working with decimal numbers.

As you have seen in the binary examples, the rule of "powers of 2" is very useful. By taking an extra bit the decimal value doubles every time:

- For every host bit you borrow the number of subnets you can create doubles.
- Every host bit left doubles the size of the subnet.

Instead of thinking/working in binary, we'll start thinking in **"blocks"**.

Take this 192.168.1.0 network with subnet mask 255.255.255.0 as an example:

We know because the subnet mask is 255.255.255.0 we have 8 bits left, and with 8 bits the highest "number" we can create is 256.

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.

Don't forget about the 0! The 0 is being used so the highest value you can create is 256.

Visualize this as a block:

We want to subnet our 192.168.1.0 network, so we'll chop our "block" in 2 pieces.

When we chop this block in 2, this is what we get:

So now we created 2 subnets out of our Class C network, the next questions are:

- What are the network addresses?
- What are the broadcast addresses?
- What is the subnet mask?
- What are the usable host IP addresses?

The network addresses we can write down, they are both blocks of "128", we'll start at 192.168.1.0 and the 2nd subnet will be 192.168.1.128. From .0 - .127 = "128".

Subnet #1: Network: 192.168.1.0

Subnet #2: Network: 192.168.1.128

The second question is, what are the broadcast addresses? Well we know that the broadcast address is the last address within a subnet, so we can just write those down now we know the network addresses:

Subnet #1:	Network:	192.168.1.0
	Broadcast:	192.168.1.127
Subnet #2:	Network:	192.168.1.128
	Broadcast:	192.168.1.255

The third question, what is the subnet mask? To solve this question I'll teach you a new trick.

Take "256" minus "block size" will give you the subnet mask:

256 – 128 = 128.

The subnet mask will be 255.255.255.128

This is a trick to remember, I would write it down on your cheat sheet.

One question left; what are the usable host IP addresses?

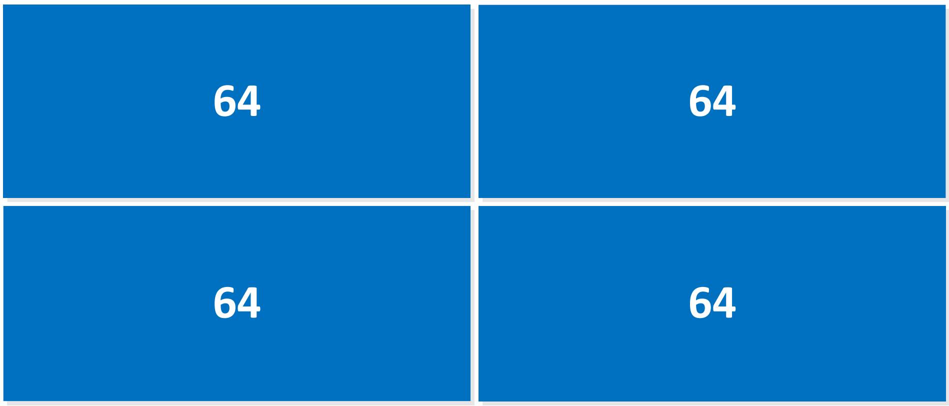
- The first usable host IP address comes after the network address.
- The last usable host IP address comes before the broadcast address.
- Everything in between is a usable host IP address.

Subnet #1:	Network:	192.168.1.0
	First Host:	192.168.1.1
	Last Host:	192.168.1.126
	Broadcast:	192.168.1.127
Subnet #2:	Network:	192.168.1.128
	First Host:	192.168.1.129
	Last Host:	192.168.1.254
	Broadcast:	192.168.1.255

That was a lot faster right? We just subnetted this Class C network, calculated the network address, broadcast address and the usable host IP addresses.

Let's try one more!

We'll take the 192.168.1.0 Class C network but now we'll chop it into 4 pieces, so we get 4 "blocks".



We have the same set of questions to answer:

- What are the network addresses?
- What are the broadcast addresses?
- What is the subnet mask?
- What are the usable host IP addresses?

Let's write down the networks, all "blocks" of 64:

Subnet #1: Network: 192.168.1.0

Subnet #2: Network: 192.168.1.64

Subnet #3: Network: 192.168.1.128

Subnet #4: Network: 192.168.1.192

Now we know the networks we can write down the broadcast addresses:

Subnet #1:	Network:	192.168.1.0
	Broadcast:	192.168.1.63
Subnet #2:	Network:	192.168.1.64
	Broadcast:	192.168.1.127
Subnet #3:	Network:	192.168.1.128
	Broadcast:	192.168.1.191
Subnet #4:	Network:	192.168.1.192
	Broadcast:	192.168.1.255

What is the subnet mask?

Take "256" minus "block size" will give you the subnet mask:

256 – 64 = 192.

The subnet mask will be 255.255.255.192

I hope you enjoyed reading the sample chapters of “How to Master Subnetting”. If you want to read the full version you can click on the link below.

Click on the picture below to get the full version: