Computer Networks-Lab

Instructor: Engr. Khuram Shahzad

CL30001 – Computer Networks-Lab

AUGUST 10, 2022

NATIONAL UNIVERSTIY OF COMPUTER AND EMERGING SCIENCES, FAST- PESHAWAR CAMPUS
Department of Computer Science & Software Engineering

Computer Networks - Lab 13 & 14

OBJECTIVES

After these Lab students shall be able to perform

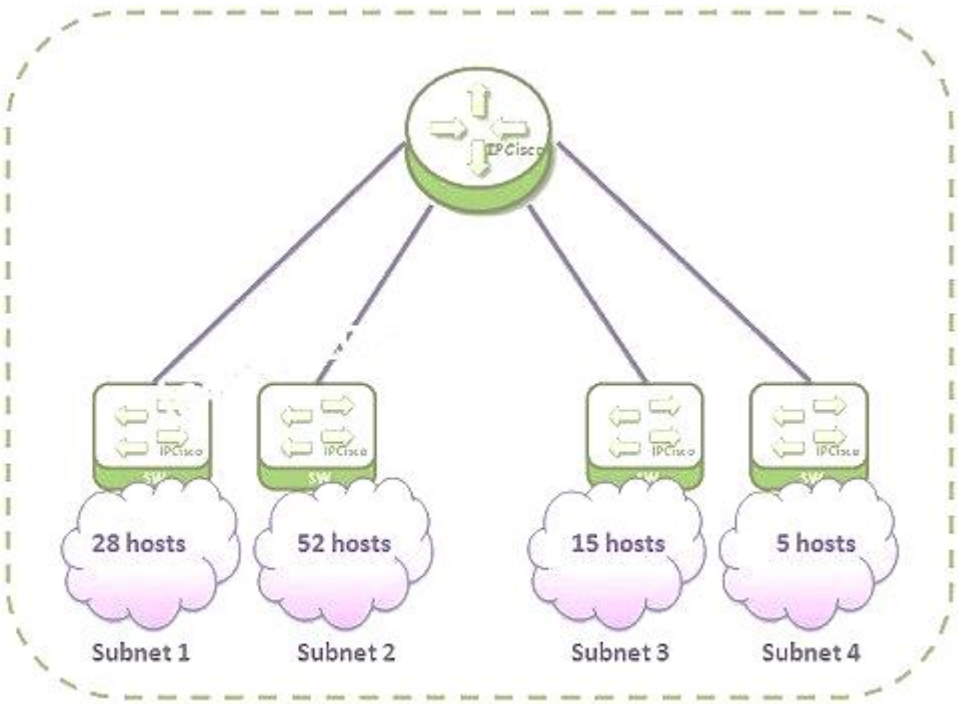
- **Practical Implementation Subnetting in Cisco Packet Tracer**
- **IP Subnetting**
    - **CIDR versus VLSM**
    - **Special Subnets**
- **Subnetting Examples**
- **VLSM Examples**
- **Paper Question solution**

PRE-LAB READING ASSIGNMENT

Remember the delivered lecture carefully.

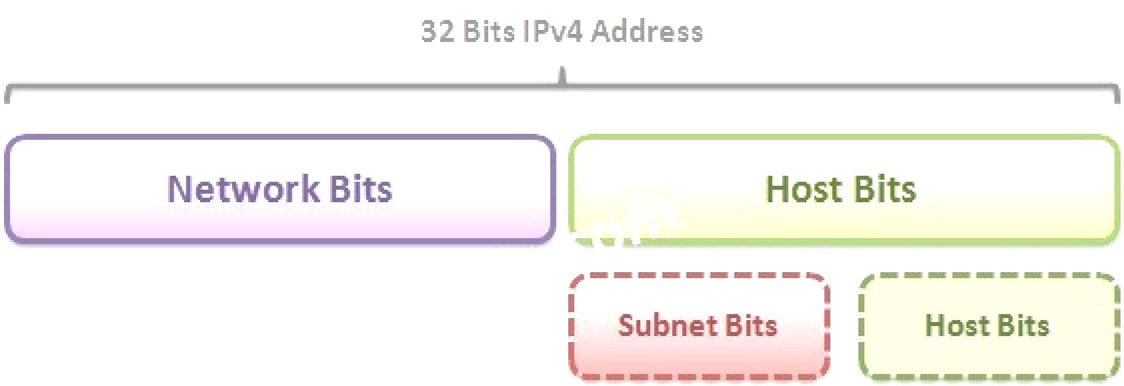
Table of Contents

IP Subnetting and Subnetting Examples



IP Subnetting

As we talked about before, there are two parts in an **IP Address**. One for them is Network part and the another is Host part. With **IP Subnetting**, we are adding one more part. This is **"Subnet Part"**. From the Host part, we borrow some bits and we will use this part for Subnet. In this lesson, we will learn Subnetting with **Subnetting Examples**.



As a basic defining, **Subnetting** is dividing the network into smaller network groups and by doing this, using the **IP Address** Block more efficient.

For Subnetting, **Subnet Masks** are used. Subnets masks are 32 bit addresses like IP Addresses. Subnet Masks are used with IP Addresses. The 1s represents the network parts, and 0s represents the host parts.

We can show Subnet Masks with four octets like **IP addresses** (255.255.255.0) or we can show it like /X . Here, for the 255.255.255.0 Subnet Mask, we can use /24. This means that the first 24 bit is full of 1s and it is network part.

Special Subnets

In Subnetting some Subnet Masks are used specifically sometimes. These are /24, /30, /31/ and /32.

● /24 **is the Subnet Mask that is usually used in the local networks by default.**
● /32 **is the Subnet Mask used generally on Loopback and System interfaces.**
● /31 **is the Subnet Mask used on point-to-point links.**
● /30 **is also widely used in Service Provider Networks for point-to-point connections.**

Loopback Interface is the "**virtual**" interfaces. There can be many Loopback interfaces in a Router. Loopback Interfaces are used for its "always up and never physically down" characteristics generally. We give these Loopback Interfaces a /32 **Loopback IP address**.

There is also a **System Address** that is used on Alcatel-Lucent Service Routers. This is a specific loopback address that provide to reach the router's itself. This address is very important for ALU routers. It is used in many protocol configurations. System addresses are /32 **IP addresses**.

Now, let's practice what we have learned with **Subnetting Examples**.

Subnetting Examples

In this part, we will see four different **Subnetting Examples**. With these **Subnetting Examples**, you will learn this lesson very well.

IP Subnetting Examples: Example 1

In the first one of the **Subnetting Examples**, we will use, 192.168.5.85 /24 Address. Let's determine the network and host part of this address. This is the first example, so we are starting with an easy example.

IP Address : **192.168.5.85**
Subnet Mask : **255.255.255.0**

For this example, firstly we will convert this decimal numbers to the binary equals. As you can see below, the 1s in the Subnet Mask will show the number of bits that network part has. And the 0s will show the host part bits.

IP Address : **11000000. 10101000.00000101.01010101**
Subnet Mask : **11111111. 11111111. 11111111.00000000**

So, here, the first 24 bits (First 3 octets) are network bits and the last 8 bits (Last octet) are the host bits.

For this IP and Subnet Mask, to determine the Network Address of this IP address, we will use "**AND**" operation between IP Address and Subnet Mask in binary mode.

IP Add: **11000000. 10101000.00000101.01010101**
SubM : **11111111. 11111111. 11111111.00000000**
AND : **11000000. 10101000.00000101.00000000**

When we use AND operation with this binary numbers, as you can see, the last octet will be multiple with zero **(AND is Multiplication)**. So the result of this multiplication will be **192.168.5.0**. Here, the first three octets will be same as IP address and the last octet will be full of 0s.

For this example our broadcast address will be **192.168.5.255**. AS you can see, all the host bits are full of 1s for broadcast address. The other addresses in the middle through **192.168.5.1** to **192.168.5.254** are host addresses.

IP Subnetting Examples: Example 2

In the second one of **Subnetting Examples**, we will do a little more complex example. This time our IP address will be 10.128.240.50/30.

IP Address : **10.128.240.50**
Sunet Mask : **255.255.255.252**

Here, there is a challenge in front of us. As you can see, we have seen the /30 and write 255.255.255.252. How can we do this? Let's see bit by bit.

/30 means that the subnet mask has 30 bits 1s and 2 bits 0s. Remember the total Subnet Mask is 32 bits. So in binary mode our Subnet Mask is:

11111111.11111111.11111111.11111100 (First 30 bits are 1s and 2 bits are 0s)

And the decimal equal of this Subnet Mask is : **255.255.255.252**

Now, let's determine the network, broadcast and host addresses of this prefix. An **IP address** with Subnet Mask called Prefix. So, we will write the binary equals of IP address and Subnet and use AND again.

IP Add : **00001010.10000000.11110000.00110010**
SubM : **11111111.11111111.11111111.11111100**
AND : **00001010.10000000.11110000.00110000**

The result of **AND** operation is the Network Address. This is **00001010.10000000.11110000.00110000** in binary. The decimal value of this is 10.128.240.48.

Here, the last two bits are host bits and the other bits are network bits. When we set all the host bits with 1s, we will find the Broadcast Address. This is **00001010.10000000.11110000.00110011** in binary. The decimal value is **10.128.240.51.**

The middle addresses can be used for hosts. These addresses are **10.128.240.49** and **10.128.240.50.**

Network Address : **10.128.240.48**
Host Addresses : **10.128.240.49 and 10.128.240.50**
Broadcast Address : **10.128.240.51**

/30 addresses are generally used in **Service Provider Networks**. So, you can work with /30 too much in the future.

Subnetting Example 3

Now, let's do a comparison example and see the benefits of Subnetting.

Think about 172.16.100.0/24 prefix and 172.16.100.0/28 prefix. As you can see, the only difference is Subnet Mask. In the first prefix, first 24 bits is network bits and the last 8 bits (32-24) are the host bits. In the second prefix, first 28 bits are network bits and the last 4 bits (32-28) are host bits.

Let's firstly talk about the first Prefix. We will write the **IP Address** and the Subnet Mask of this Prefix in binary format:

172.16.100.0 = **10101100.00010000.01100100.00000000**
255.255.255.0 =**11111111.11111111.11111111.00000000**

When we use **AND** operation, our network address is **10101100.00010000.01100100.00000000** (the same as the IP showed in prefix by change). This is 172.16.100.0 .

And the Broadcast address is 172.16.100.255. The other 254 IP addresses are host IP addresses.

172.16.100.1
172.16.100.2
….
172.16.100.254

For the first prefix, we have only one network that has 254 hosts.

Now, let's check the second prefix. 172.16.100.0 /28.

172.16.100.0 = 10101100.00010000.01100100.00000000
255.255.255.240 =11111111.11111111.11111111.11110000

When we use **AND** operation here, our network address will be the same, in binary 10101100.00010000.01100100.00000000 or in decimal 172.16.100.0. But our Broadcast address will change, because our host bits are only the last 4 bits anymore.

```
10101100.00010000.01100100.00000000 network address (172.16.100.0)
10101100.00010000.01100100.00001111 broadcast address (172.16.100.15)
```

This is for the first network. We divide the network by using higher Subnet Mask. So, let's look at the other networks:

```
10101100.00010000.01100100.00010000
10101100.00010000.01100100.00100000
10101100.00010000.01100100.00110000
10101100.00010000.01100100.01000000
….
10101100.00010000.01100100.11110000
```

As you can see, we have **16 networks**. We have divide a Prefix, into smaller 16 different Prefix. Each of these Prefix has **14** host address, **1** broadcast address and **1** network address.

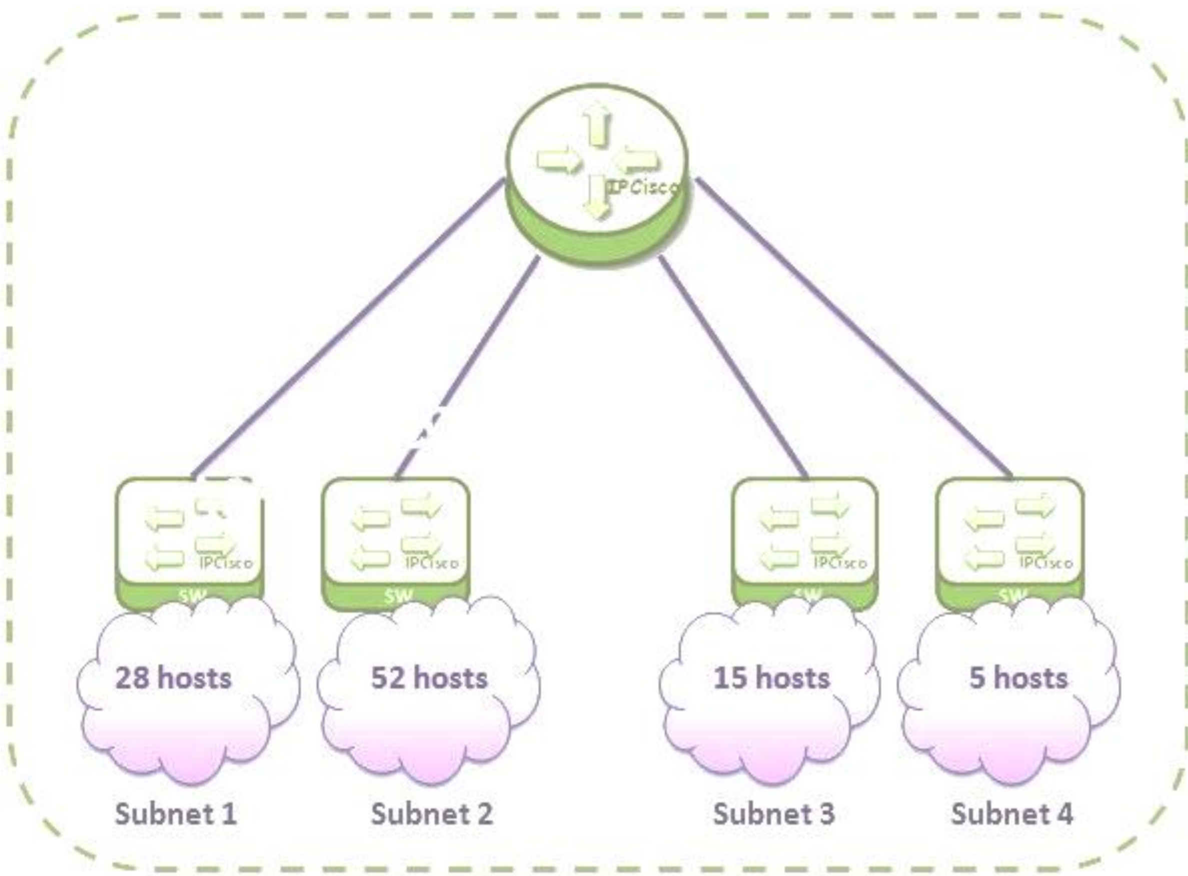
So, if we use a given address with a higher Subnet Mask value like given in the second example, we will have more networks. In other words, we can divide the network into smaller pieces. So, we will not waste the **IP Addresses**. Smaller networks that has few hosts do not need more addresses. With Subnetting, using a small network with few host addresses is a way of best practice of a network engineer.

Before the usage of an IP Prefix, it is better to check your needs for now and for the future. How many subnets and hosts you need and you will need in the future? According to these needs, you can determine Subnetting and divide your IP Prefix into smaller parts.

Subnetting Examples, Example 4

In the last of these **Subnetting Examples**, we will see our network's needs and according to these needs, we will determine our IP Address Prefixes.

We will use the below topology. And we have given **192.168.1.0/24** IP Address.



As you can see, in this topology, there are four subnets and each subnets host address need is also given.

Subnet 1 = 28 hosts
Subnet 2 = 52 hosts
Subnet 3 = 15 host
Subnet 4 = 5 hosts

To overcome this Subnetting issue, firstly we determine the host bits for each subnet.

For the first subnet; we need 5 host bits. With 5 bits we can have 2^5=32 addresses. This means that there are 32-2 usable host addresses.

For the second subnet; we need 6 host bits. With 6 bits we can have 2^6=64 addresses. This means that there are 64-2 usable host addresses.

For the third subnet; we need 5 host bits. With 5 bits we can have 2^5=32 addresses. You can think that we can use 2^4=16 address. But we cannot. Because, one of the address is used for broadcast address and the other is for network address. This means that there are 14 usable addresses.

For the fourth subnet; we need 3 host bits. With 3 bits we can have 2^3=8 addresses. This means that there are 8-2 usable host addresses.

Now let's pick it up.

For first subnet, our Subnet Mask will be /27 (27 network bits and 5 host bits. 5+27=32)
For second subnet, our Subnet Mask will be /26 (26 network bits and 6 host bits. 6+26=32)
For third subnet, our Subnet Mask will be /27 (27 network bits and 5 host bits. 5+27=32)
For fourth subnet, our Subnet Mask will be /29 (29 network bits and 3 host bits. 3+29=32)

Here, the router interfaces will also need **IP address**. So, for each subnet, one IP address will be go to the Router interface.

Remember, we have given an IP address 192.168.1.0/24. Let's divide this Prefix according to the above values.

Let's begin with the biggest network. If we use /26 with 192.168.1.0 like 192.168.1.0/26, then we will have 4 subnets. The given Subnet was 24 and our new subnet is 26. 26-24=2 and 2^2=4 subnets.

```
192.168.1.0/26
192.168.1.64/26
192.168.1.128/26
192.168.1.192/26
```

We will use the first one for the Subnet 2. (192.168.1.0/26)

Now, for the first and third subnet, let's use the second block (192.168.1.64/26) and divide it again. If we divide it by borrowing a bit again, then we will have two subnets.

```
192.168.1.64/27
192.168.1.96/27
```

We can use these two Prefixes for first and second subnet.

And lastly, for the small subnet, we can use the above third block(192.168.10.128/26). We will divide it again. Because we need only 5 host address.

When we divide again, we will have the below small subnets, and we can use the first one for our fourth and last Subnet.

```
192.168.1.128/29
192.168.1.136/29
192.168.1.144/29
```

```
192.168.1.152/29
192.168.1.160/29
192.168.1.168/29
192.168.1.176/29
192.168.1.184/29
```

As you can see, with this Subnetting, we have used our IP Block very efficiently. The unused remaining blocks can be used in the future. What are these remaining blocks let's remember. The remaining blocks are the last block of **/26 subnets** and the last 7 block of **/29 subnets**.

At the end our subnets will be like below:

```
192.168.1.0/26
192.168.1.64/27
192.168.1.96/27
192.168.1.128/29
```

This is **VLSM (Variable Length Subnet Mask)**. Its meaning is using subnet of subnets or dividing a network into smaller network with using different subnet mask

Class A network subnetting

Available Subnet – 16.0.0.0/8

Required Subnets – 700

__Process__

Convert subnet mark to Binary

/8 = 255.0.0.0 = 11111111.00000000.00000000.00000000

All ones represents the network   zeroes represents the hosts

---

__Find an increment__

2^n = subnets (n = numbers of bits required)

To find out required bit multiply 2x2 until you get required subnets number or bigger number

In this scenario we need 700 networks and we are getting number 1024 by 2x2x2x2x2x2x2x2x2x2 = 1024

2^10 = 1024 subnets

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int f1/0
Router(config-if)#ex
Router(config)#int f0/0
Router(config-if)#ip add 58.65.153.1 255.255.255.224
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#ex
Router(config)#ip dhcp pool net1
Router(dhcp-config)#network 58.65.153.0 255.255.255.224
Router(dhcp-config)#default-router 58.65.153.1
Router(dhcp-config)#dns-server 58.65.153.1
Router(dhcp-config)#ex
Router(config)#
Router(config)#int f2/0
Router(config-if)#ip add 58.65.153.33 255.255.255.224
Router(config-if)#no shut
```

Count all the ones to get new subnet mask

Our new subnet mask is /18 = 255.11111111.11000000.00000000

All Ones -255   128+64=192

= 255.255.192.0

128 64 32 16 8 4 2 1

1 1 0 0 0 0 0 0   64 is the smallest network bit

We will find our network ranges using 64 as a increment

Network ranges

16.0.0.0----------------- 16.0.63.255
+64 → (Increment)

16.0.64.0---------------- 16.0.127.255
+64

16.0.128.0--------------- 16.0.191.155
+64

16.0.192.0--------------- 16.0.255.255
+64

16.1.0.0------------------ 16.1.63.255
+64

16.1.64.0----------------- 16.1.127.255
+64

16.1.128.0--------------- 16.1.191.255
+64

16.1.192.0----------------16.1.255.255   and so on

Class A Subnetting Example

Available Subnet – 15.0.0.0/8

Required Hosts – 8

Process

Convert subnet mark to Binary

/8 = 255.0.0.0 = 11111111.00000000.00000000.00000000

All ones represents the networks zeroes represents the hosts

Find an increment

2^n-2 = Hosts (n = numbers of bits required)

(We have to subtract 2. We cannot use first and last ip of the subnet range. First ip

will be used as subnet ID and the last will be used as a broadcast address)

To find out required bits, multiply 2x2 until you get bigger number by 2 than required Hosts

(We need bigger number because we will get less host ips after subtracting 2. For example: if we need 64 hosts and we are getting exact number 64 then that means we have 64 ips but we have to subtract 2 for the subnet id and broadcast address. 64-2=62, we will get 62 hosts instead of 64. Bigger number will give us more ips to use)

In this scenario we need 8 Hosts and we are getting bigger number than 8 by 2x2x2x2 = 16,

2^4= 16 - 2 = 14 hosts per network

We need to save four bits to get 8 hosts

Save 4 host bits and convert rest to network bits

11111111.11111111.11111111.1111<u>0000</u>

8 8 8 4

Four host bits has been saved and others are converted to network bits

Count all the ones to get new subnet mask

Our new subnet mark is/28 = 255.11111111.11111111.<u>1111</u>0000

(255) (255)

128+64+32+16=240

= 255.255.255.240

128 64 32 <u>16</u> 8 4 2 1

1 1 1 <u>1</u> 0 0 0 0 16 is the smallest network bit

We will find our subnet ranges using 16 as a increment

<u>Network ranges</u>

15.0.0.0———————————— 15.0.0.15

15.0.0.16———————————— 15.0.0.31

15.0.0.32—————— —————— 15.0.0.47

15.0.0.48———————————— 15.0.0.64 and so on

First address – Network ID Last address – Broadcast address

Subnetting Class B Network

Available Subnet – 160.10.0.0/16

Required Subnets – 200

Process

Convert subnet mask to Binary

/16 = 255.255.0.0 = 11111111.11111111.00000000.00000000

All ones represents the network   zeroes represents the hosts

---

Find an increment

2^n = subnets (n = numbers of bits required)

To find out required bits multiply 2x2 until you get required subnets number or bigger number

In this scenario we need 200 networks and we are getting number 256 by
2x2x2x2x2x2x2x2 = 256

2^8 = 256 subnets

We need to convert eight bits to get 200 networks

We have to change hosts bits into network bits to get more subnets

11111111.11111111.11111111.00000000
8 8 8

Eight hosts bits are now network bits

Count all the ones to find out new subnet mask

Our new subnet mask is /24 = 255.255.<u>11111111</u>.00000000

↓

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

= 255.255.255.0

128 64 32 16 8 4 2 <u>1</u>

1 1 1 1 1 0 0 <u>0</u> 1 is the smallest network bit

→ We will find our network ranges using 1 as a increment

<u>Network ranges</u>

160.10.0.0 - 160.10.0.255
+1 → (Increment)

160.10.1.0 - 160.10.1.255
+1

160.10.2.0 - 160.10.2.255
+1

160.10.3.0 - 160.10.3.255
+1

160.10.4.0 - 160.10.4.255
+1

160.10.5.0 - 160.10.5.255
+1

160.10.6.0 - 160.10.6.255
+1

160.10.7.0 - 160.10.7.255 and so on

2x2x2x2x2x2x2 = 128

2^7 = 128 subnets

We need to convert seven bits to get 128 subnets

We have to change hosts bits into network bits to get more networks

11111111.11111111.11111110.00000000

8 8 7

seven hosts bits are now network bits

Count all the ones to get new subnet mask

Our new subnet mask is /23 = 255.255.11111110.00000000

128+64+32+16+8+4+2=254

= 255.255.254.0

128 64 32 16 8 4 2 1

1 1 1 1 1 0 0 0 2 is the smallest network bit

We will find our subnet ranges using 2 as a increment

Class B Subnetting

Available Subnet – 140.168.0.0/16

Required Hosts – 400

Process

Convert subnet mark to Binary

/16 = 255.255.0.0 = 11111111.11111111.00000000.00000000

All ones represents the networks   zeroes represents the hosts

---

Find an increment

2^n-2 = Hosts (n = numbers of bits required)

(We have to subtract 2. We cannot use first and last ip of the subnet range. First ip

will be used as subnet ID and the last will be used as a broadcast address)

To find out required bits, multiply 2x2 until you get bigger number by 2 than required Hosts

(We need bigger number because we will get less host ips after subtracting 2. For example: if we need 64 hosts and we are getting exact number 64 then that means we have 64 ips but we have to subtract 2 for the subnet id and broadcast address. 64-2=62, we will get 62 hosts instead of 64. Bigger number will give us more ips to use)

In this scenario we need 400 Hosts and we are getting number 512 by 2x2x2x2x2x2x2x2x2 = 512,

2^9 = 512 - 2 = 510 hosts

We need to save nine bits to get 400 hosts

Save 9 host bits and convert rest to network bits

11111111.11111111.1111111<u>0.00000</u> <u>000</u>

8 8 7

Nine host bits has been saved and remaining seven are converted to network bits.

Count all the ones to get new subnet mask

Our new subnet mark is/23 = 255.255.<u>1111111</u>0.00000000

128+64+32+16+8+4+2=254

= 255.255.254.0

128	64	32	16	8	4	<u>2</u>	1
1	1	1	1	1	0	<u>0</u>	0

2 is the smallest network bit

We will find our subnet ranges using 2 as a increment

<u>Network ranges</u>

<u>Subnet ID</u> *<u>Broadcast Address</u>*

140.168.0.0——————— 140.168.1.255

140.168.2.0——————— 140.168.3.255

140.168.4.0———————- 140.168.5.255

140.168.6.0———————- 140.168.7.255 and so on

Class C Subnetting Example

Available Subnet - 215.37.1.0/24

Required subnets – 25

<u>Process</u>

Convert subnet mask to Binary

/24 = 255.255.255.0 = 11111111.11111111.11111111.00000000

↓ ↓

All ones represents the network zeroes represents the hosts

Find an increment

2^n = subnets (n = numbers of bits required)

To find out required bits, multiply 2x2 until you get required subnets number or bigger number

In this scenario we need 25 subnets and we are getting number 32 by 2x2x2x2x2 = 32,

2^5 = 32 subnets

We need to convert five bits to get 25 subnets

We have to change hosts bits into network bits to get more subnets

11111111.11111111.11111111. <u>11111</u>000

8 8 8 5

Five hosts bits are now network bits

Count all the ones to get new subnet mask

Our new subnet mask is /29 = 255.255.255.<u>11111</u>000

128+64+32+16+8=248

= 255.255.255.248

128	64	32	16	<u>8</u>	4	2	1
1	1	1	1	<u>1</u>	0	0	0

8 is the smallest network bit

We will find our network ranges using 8 as a increment

Network ranges

215.37.1.0------------------ 215.37.1.7
+8 → (Increment)

215.37.1.8------------------ 215.37.1.15
+8

215.37.1.16----------------- 215.37.1.31
+8

215.37.1.32----------------- 215.37.1.39
+8

215.37.1.40----------------- 215.37.1.47
+8

215.37.1.48----------------- 215.37.1.55
+8

215.37.1.56----------------- 215.37.1.63
+8

215.37.1.64----------------- 215.37.1.71 and so on