Day-10

# Subnetting

→ Let's see these IPv4 address classes, so we can then understand the needs for classLESS IPv4 Addressing.

| Class | First octet (Binary) | first octet range (Decimal) | |
|---|---|---|---|
| A | 0xxxxxxx | 0-127 | 0.0.0.0 ~ 127.255.255.255 |
| B | 10xxxxxx | 128-191 | 128.0.0.0 ~ 191.255.255.255 |
| C | 110xxxxx | 192-223 | 192.0.0.0 ~ 223.225.255.255 |
| D | 1110xxxx | 224-239 | 214.0.0.0 ~ 239.255.255.255 |
| E | 1111xxxx | 240-255 | 240.0.0.0 ~ 255.255.255.255 |

Only class A, B & C addresses can be assigned to a device as an IP address. as classes D & E have special purposes I mentioned in the IPv4 addressing Topic.

| Class | Prefix |
|---|---|
| A | 18 |
| B | 116 |
| C | 124 |

→ So, how does a company get their own network to use? Well, IP addresses are assigned to companies or organizations by a non-profit American corporation called IANA, the

The Internet Assigned Numbers Authority.

→ The IANA assigned IPv4 address & Networks to companies based on their size.

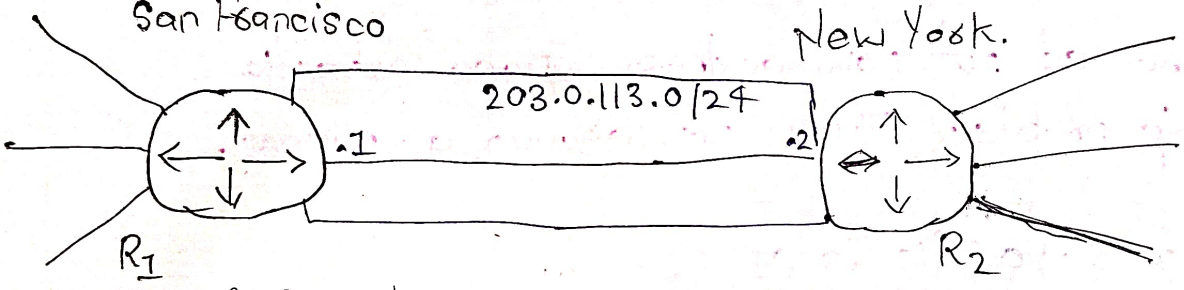
- for Example :→ large.
a very small company might receive a class A or class B network, while a small company might receive a class C network.

However this system led to many wasted IP addresses, so multiple methods of improving this system has been created. Let me give a Example of how this strict system of addresses can waste IP addresses

→ So, here are two routers. R1 has three network connected to it, R2 has three network connected here, each of these network will have a few switches, with many end hosts such as PCs & servers. connect to these switches.

→ However, there is one more network here. That's this network connecting two routers. This is known as 'point-to-point' networking, meaning that its a network connecting two points in this case R1 & R2.

So, because it is a point to point network we don't need a large address block, so let use a class C network, 203.0.113.0/24 Because this a class C network there are 256 address




256 addresses

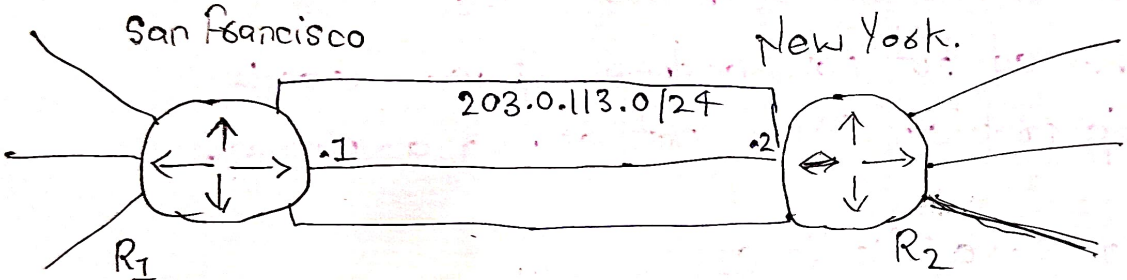
- -1 network address (203.0.113.0)
- -1 broadcast address (203.0.113.255)
- -1 R1's address (203.0.113.1)
- -1 R2's address (203.0.113.2)

= 252 addresses WASTED

Clearly this is not a ideal system.

→ However, there is one more network here. That's this network connecting two routers. This is known as 'point-to-point' networking, meaning that its a network connecting two points in this case R1 & R2.

So, because it is a point to point network we don't need a large address block, so let use a class C network 203.0.113.0/24 Because this a class C network there are 256 address




256 addresses

-1 network address (203.0.113.0)
-1 broadcast address (203.0.113.255)
-1 R1's address (203.0.113.1)
-1 R2's address (203.0.113.2)

= 252 addresses WASTED

Clearly this is not a ideal system.

# CIDR (Classless Inter-Domain Routing)

→ When the Internet was first created, the creators did not predict that the Internet would become as large as it is today.

This resulted in wasted address space like the example I showed you (there are many such more Ex).

> "The Total IPv4 address space include over 4 billion addresses, and that seemed like a huge number of addresses when IPv4 was created but now address space exhastion is a big problem, there not enough addresses".

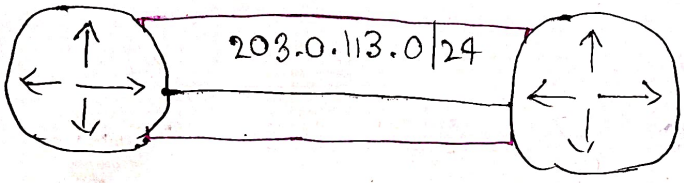
One way to solve this is CIDR.

→ The IETF (Internet Engineering Task force) introduced CIDR In 1993 to replace the 'classful' addressing system.

- With CIDR, the requisments of
  - Class A = /8
  - class B = /16
  - class C = /24
  - .... were removed.
- This allowed larger network to be split into smaller network, allowing greater efficiency.
- These smaller network are called 'subnetwork' or 'subnets'.

→ Let's look at an example of splitting a larger network into a smaller network so, we understand how it works?

The same point-to-point network we looked at before Previously, it was assigned the 203.0.113.0/24 network space, but that resulted in lots of wasted addresses.




11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 00000000
255 . 255 . 255 . 0

network address, broadcast address

$2^8 - 2 = 254$ usable addresses.

number of host bits

So, we have 254 usable addresses, but we only need two one for R1 & one for R2.

Howeves, CIDR allows us to assign diffesent psefix length. it doesn't have to be /24. Let's get some psactice calculating the numbess of hosts with diffesent Psefix.

## CIDR Psactice

How many usable addsess ase these in each netwosk?

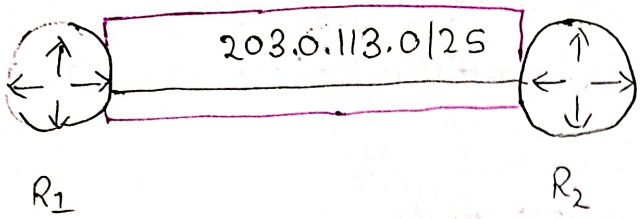
- 203.0113.0/25
  203.0.113.0/26
  203.0.113.0/27
  203.0.113.0/28
  203.0.113.0/29
  203.0.113.0/30
  203.0.113.0/31
  203.0.113.0/32

$2^n - 2$ = usuable addsesses
$n$ = numbes of host bits.

→ i put /31 & /32 in sed because they'se a little bit special.

Let's calculate how many usable addsess ase in each netwosk?

## CIDR(/25)



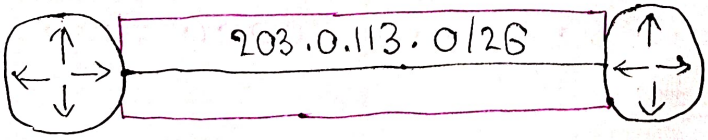

Notice that the network portion of the address has extend into the first bit of the last octet ~~& the mask~~

11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 10000000
255 . 255 . 255 . 128

$2^7-2 = 126$ usable address, but we want only two (2) address one for $R_1$ & one for $R_2$ so we will be wasting 124 addresses. That better than wasting 252 addresses with a /24 prefix length but its still wasteful.

## CIDR(/26)




11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

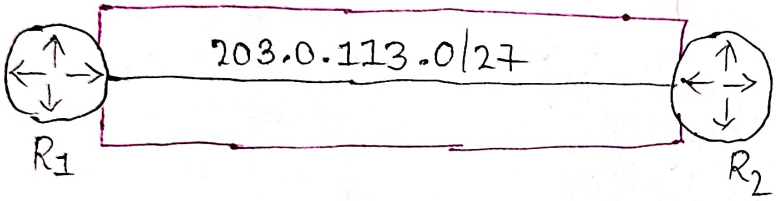
11111111 . 11111111 . 11111111 . 11000000
255 . 255 . 255 . 192

Two bits of the last octet are now part of the network portion.

$2^6 - 2 = 62$ usable address in this network.
↳ host bits

If we were to use a /26 network mask for the 203.0.113.0 network we would wasting 60 addresses.

## CIDR (/27)




11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11100000
255 . 255 . 255 . 224

$2^5 - 2 = 30$ usable addresses.
↳ host bits.

## CIDR (/28)

11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11110000
255 . 255 . 255 . 240

$2^4 - 2 = 14$ usable addresses.
↳ host bits.

## CIDR(129)

11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11111000
255 . 255 . 255 . 248

$2^3 - 2 = 6$ usable addresses.
↳ host bits

After give $R_1$ & $R_2$ addresses 4 wasted addresses.

## CIDR(130)

11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11111100
255 . 255 . 255 . 252

These are only Two host bits, means 2 usable addresses. So this is perfect

$2^2 - 2 = 2$ usable addresses.
↳ host bit

4 Total address, that's the network address & the broadcast address, R1 Address & R2 address, 0 wasted addresses.

Before moving on to /31 & /32 Let me clasify a little bit. So instead of 203.0.113.0/24 we will use 203.0.113.0/30 which is a subnet of that larger class C network.

203.0.113.0/30

include the address range of = 203.0.113.0 - 203.0.113.3

(203.0.113.0) 11001011.00000000.01110001.00000000
(203.0.113.1) 11001011.00000000.01110001.00000001
(203.0.113.2) 11001011.00000000.01110001.00000010
(203.0.113.3) 11001011.00000000.01110001.00000011

2 usable addresses which are assigned to $R_1$ & $R_2$

So, we took up 4 addresses with this subnet what about the other addresses in the 203.0.113.0/24?

→ The remaining addresses in the 203.0.113.0/24 address block (203.0.113.4 - 203.0.113.255) are now available to be used in other subnets!

that's the magic of subnetting instead of using 203.0113.0/24 and wasting 252 addresses We can use /30 and waste 0 addresses(no) Os, perhaps there is another way to make this even more efficient? Lets look into it.

# CIDR(/31)

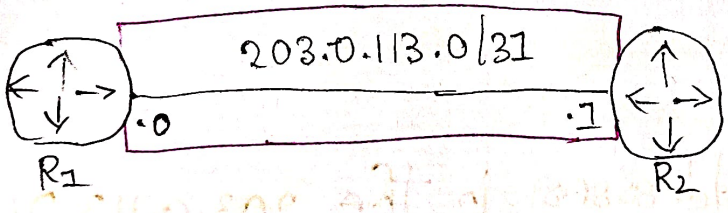
11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11111110
255 . 255 . 255 . 254

There is only 1 host bit, so means 0 usable address.

$2^1 - 2 = 0$ usable addresses that we can assign to device.

So, you used to not abe able to use /31 network prefixes because of this. However for a point-to-point connection like this it usually is possible to use /31 mask. Let's check it out.




$R_1$ = 203.0.113.0

$R_2$ = 203.0.113.1

The, 203.0.113.0/31 network consists of the addresses.

= (203.0.113.0 - 203.0.113.1) which is actually ony 2 address.

(203.0.113.0) 11001011 . 00000000 . 01110001 . 00000000

(203.0.113.1) 11001011 . 00000000 . 01110001 . 00000001

Normally this would be a problem, because it leaves no usable addresses after subtracting the network & broadcast addresses, but for point-to-point network like this # dedicated connection like this b/w two routers, there is actually no need for a network address or a broadcast address so, we can break the rules in this case & assign the only two addresses in this network to our routers.

NOTE:- If you try this configuration on cisco routers you'll get a warning like this.

Reminding you to make sure that this is a Point-to-point link, but it is totally valid configuration.

```
Router(config-if)#ip address 203.0.113.0 255.255.255.254
% Warning: use /31 mask on non point-to-point interface cautiously
Router(config-if)#
```

→ The remaining addresses in the 203.0.113.0/24 address block which are (203.0.113.2 - 203.0.113.25
are now available to be used in other network.

Pepole still do use /30 for a point-to-point connections at times, but /31 mask are totally valid & more efficient than /30.

So, I recommed this method.

# CIDR(132)

11001011 . 00000000 . 01110001 . 00000000
203 . 0 . 113 . 0

11111111 . 11111111 . 11111111 . 11111111
255 . 255 . 255 . 255

$2^0 - 2 = -1$ usable address
host bits

clearly the formula doesn't work in this case You won't be able to use a /32 mask in this case, and you will probably never use /32 mask configure an actual interface. However there are some, example when you want to uses for a /32 mask for create a static route not to a network, but just to one specific host, you can use a /32 mask to specify that exact host.

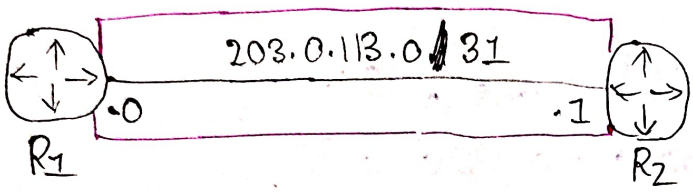
## CIDR Notation

| Dotted Decimal | CIDR Notation |
|---|---|
| 255.255.255.128 | /25 |
| 255.255.255.192 | /26 |
| 255.255.255.224 | /27 |
| 255.255.255.240 | /28 |
| 255.255.255.248 | /29 |
| 255.255.255.252 | /30 |
| 255.255.255.254 | /31 |
| 255.255.255.255 | /32 |

A simple chart showing the dotted decimal subnet mask & their equivalent in CIDR notation.

That's right the way of writing a prefix with a slash followed by the prefix length like /25, /26, etc is called CIDR Notation, because it was introduced with the CIDR System

## Subnetting

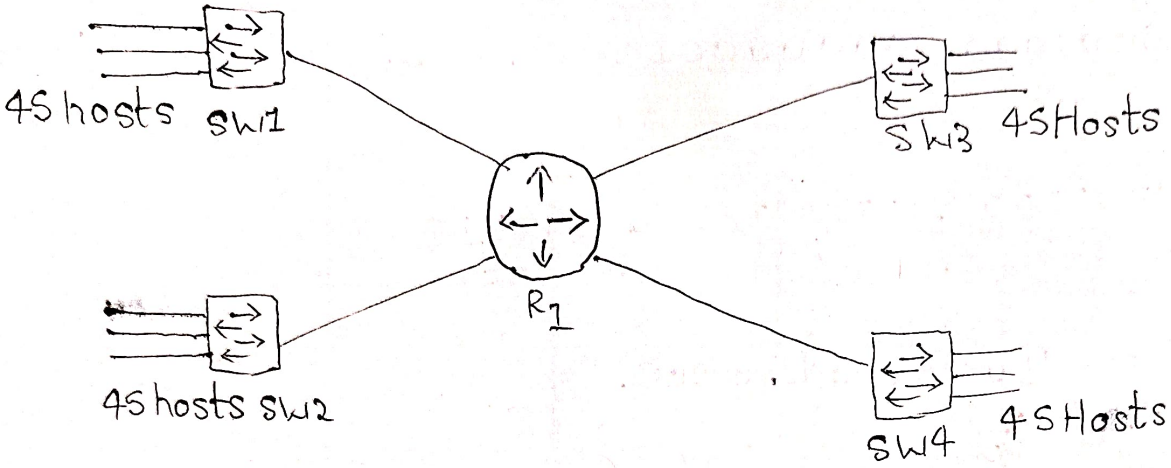



→ I spend alot of time one just one Example, but I hope you can see the use of subnetting dividing a larger network into smaller network called subnet. Instead of using whole 203.0.113.0/24 network for the point-to-point connection we can use a /30 subnet and use only 4 addresses or even better use a /31 subnet & use only 2 addresses.

One more Example:

Here's a scenario. There are 4 networks connected to R1 with many hosts connected to each switch These 45 hosts per network R1 needs an IP address in each network so its address is included in the range You have received the 192.168.1.0/24 network,

Divide the 192.168.1.0/24 network into four subnets that can accommodate the numbers of hosts required.




first off, are there enough addresses in 192.168.1.0/24 network in the first place?

So, we need 45 hosts per network, including R1, but also remember that each network has a network & broadcast address so that's plus 2

so we need 47 addresses per subnet

47 X 4 = 188, so there's no problem in terms of the number of hosts.

192.168.1.0/24 is a class c network so, these are 256 addresses, so we will be able to assign 4 subnets to accomodate all hosts, no problem.

Okay let's see how we can calculate the subnets we need to make. We need four equal sized subnets with enough room for at least 45 hosts.

Here, I've written out 192.168.1.0 with a /30 mask, 255.255.255.252.

I skiped /32 & /31 because they are point-to-point.

(/30)

11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

11111111 . 11111111 . 11111111 . 11111100
255 . 255 . 255 . 252

$2^2 - 2$ = 2 usable addresses.
↳ 2 host bit

(/28)

11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

11111111 . 11111111 . 11111111 . 11110000
255 . 255 . 255 . 240

$2^4 - 2$ = 14 usable addresses.
4 host bit

(/26)

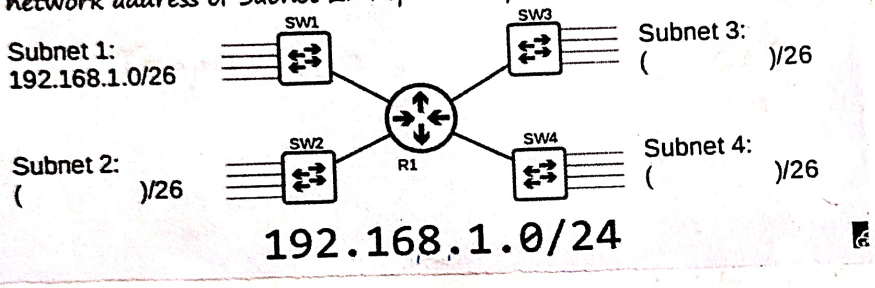
11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

11111111 . 11111111 . 11111111 . 11000000
255 . 255 . 255 . 192

$2^6 - 2$ = 62
host bit

/27 doesn't provide enough address space.
/26 provide more than we need, but we have to go with /26. unfortunately we can't always make subnets have exactly the number of addresses you want. some unused address space. That's actually fine since its good to have some room for growth anyway.

The first subnet (Subnet 1) is 192.168.1.0/26. What are the remaining subnets?
HINT: Find the broadcast address of Subnet 1. The next address is the network address of Subnet 2. Repeat the process for Subnets 3 and 4.




## Subnet 1

Subnet 1: 192.168.1.0/26

11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

To find the broadcast address for this subnet which is highest address in the subnet's address range, set all of the bits in the host portion to 1.

11000000 . 10101000 . 00000001 . 00111111
192 . 168 . 1 . 63

that's the broadcast address.

So, the address range for subnet 1 is:

(192.168.1.0 - 192.168.163)

The network address of subnet 2 will be 1 higher than the broadcast address.

(Subnet 2)

Subnet 2: 192.168.1.64/26

Here it is in binary notice that we changed one of the bits we borrowed from the host postion to a 1.

11000000 . 10101000 . 00000001 . 01000000
192 . 168 . 1 . 64

Next broadcast addresses set all to 1.

11000000 . 10101000 . 00000001 . 01111111
192 . 168 . 1 . 127

So, the range of subnet 2 is:

(192.168.1.64 - 192.168.1.127)

Add 1 to the broadcast address and we'll get the network address for subnet 3.

(Subnet 3)

Subnet 3: 192.168.1.128/26

Here is the network address in binary, again notice we changed one of the borrowed purple bits to 1 but the host bits are all 0.

11000000 . 10101000 . 00000001 . 10000000
192 . 168 . 1 . 128

Next broadcast address set host bit all to 1.

11000000 . 10101000 . 00000001 . 10111111
192 . 168 . 1 . 191

So, the range of subnet 3 is:
(192.168.1.128 – 192.168.1.191).

Add 1 to broadcast address and we ge network address for subnet 4.

(Subnet 4)

Subnet 4: 192.168.1.192/26.

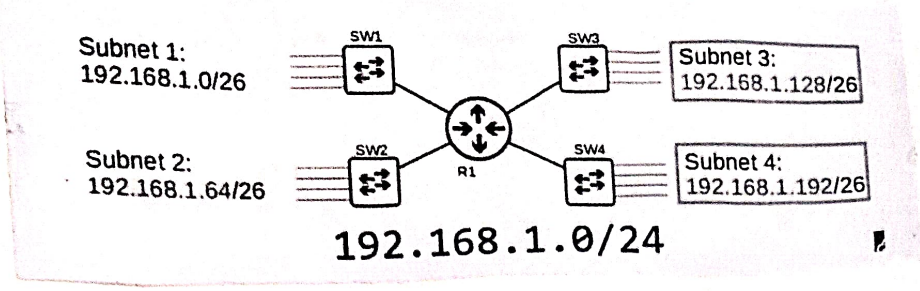
Here is the network address in binary, this time the borowed bits are all 1, so this is our last subnet, we dont have any room for more.

11000000 . 10101000 . 00000001 . 11000000
192 . 168 . 1 . 192

Chage host bit to 1. and here the broadcast addsesses.

11000000 . 10101000 . 00000001 . 11111111
192 . 168 . 1 . 255

Subnet 4 range is:
(192.168.1.192 - 192.168.1.255)




## Subnetting Tricks

192.168.1.0/26

11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

So, we found /26 subnet mask is appropriate that because there are 6 host bits which allow 62 hosts.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NETWORK PORTION | | HOST PORTION | | | | | |

NOTICE that LAST bit of the network portion is 64, this means that to find the next subnet, we just to add 64
Add 64 & we get

192.168.1.64 which is the network address of subnet 2

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NETWORK | | HOST | | | | | |

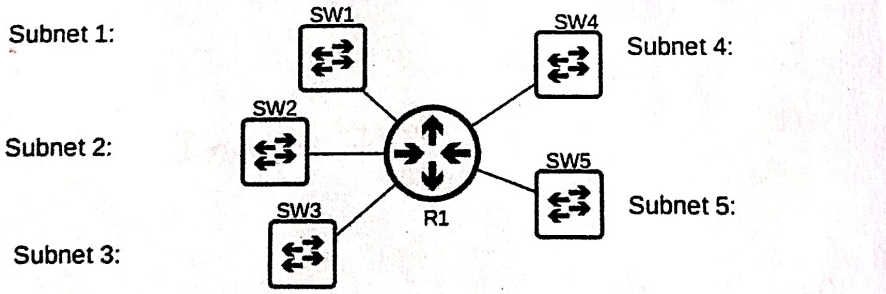
Add 64 again we get
192.168.1.128 which is the network address of subnet 3.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| NETWORK PORTION | | HOST PORTION | | | | | |

Add 64 again & we get
192.168.1.192 which is the network address of subnet 4.

So, as you see by adding 64 each time we were able to find the network addresses of each subnet.

Now, Another similar Exercise




**192.168.255.0/24**

*Divide the* **192.168.255.0/24** *network into five subnets of equal size. Identify the five subnets.*

In this case, the number of hosts in each subnet hasn't been specified so let's make 5 subnet that are as large as they can be

so, all we have to do to solve this problem is find how many bits we have to "borrow" from the host portion

(/24)

11000000 . 10101000 . 11111111 . 00000000
192 . 168 . 255 . 0

Borrowing 0 bits = can't make any subnets

(/25)

11000000 . 10101000 . 11111111 . 00000000
192 . 168 . 255 . 0

Borrowing 1 bit = can make 2 subnets

Why is that?

Well all of the orginal network bits the blue bits cannot be changed
That is the network we received.
However the purple bit, the bit we borrowed from the host portion, we can changed & it can be either 0, 1 or

If its 0 we have the 192.168.255.0/25 network
If its 1 we have the 192.168.255.128/25 network

The formula for the number of subnets:

$2^X$ = number of subnets

X = (number of borrowed bits)

the number of host in subnet:

* $2^n - 2$ = number of hosts
  $n$ = number of host bit.

-1 = network address
-1 = broadcast address

→ Anyway, we need 5 subnet, so borrowing 1 bit isnt enough (/26)

Borrowing 2 bit = can make 4 subnet

still not enough we need 5 subnet.

(/27)

11000000 . 10101000 . 11111111 . 00000000
192 . 168 . 255 . 0

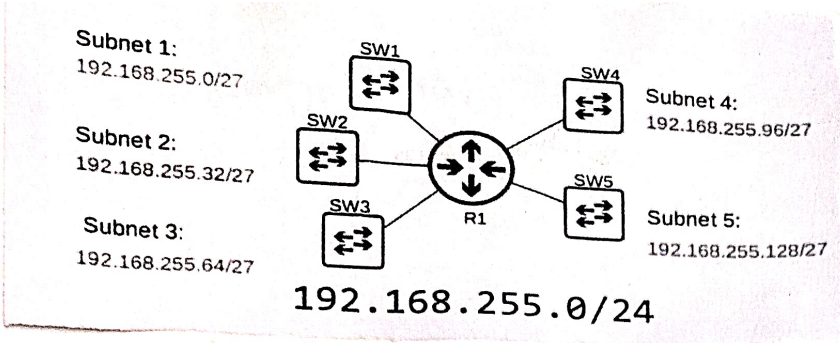
Borrowing 3 bit = can make 8 subnet

So, this is our answer. its more subnet than we need, we need 5 but if we borrow only 2 bits and use a /28 mask we wont have enough

So, our first subnet will be: 192.168.255.0/27

| 11000000 | 10101000 | 11111111 | 128 64 32 | 16 8 4 2 1 |
|---|---|---|---|---|
| | | | 0 0 0 | 0 0 0 0 0 |
| | | | NETWORK Position | Host Position. |

So, using the trick i tell you, should be able to calculate the other subnets now

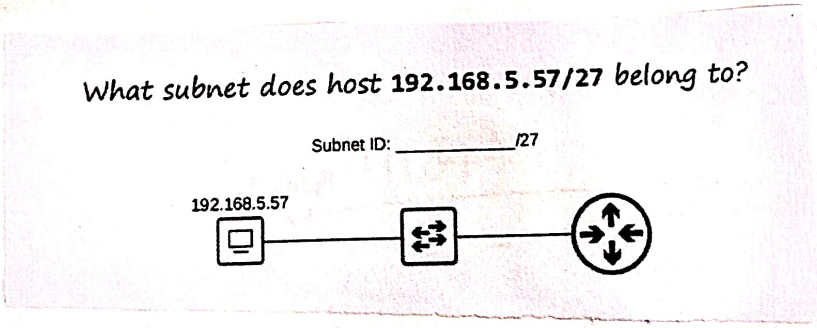



Okay we made up 5 subnet. as ourequirment but /27 Prefix length allow upto 8 subnet to be made from the address range

the remaining subnets are:

Subnet 6: 192.168.255.160/27
Subnet 7: 192.168.255.192/27
Subnet 8: 192.168.255.224/27

## Identify the Subnet




So, we have the IP address of a host, 192.168.5.57 & don't know the network address of the subnet. Let see how we can figure out it.

Here, 192.168.5.57 in binary

11000000 . 10101000 . 00000101 . 00111001
192 . 168 . 5 . 57

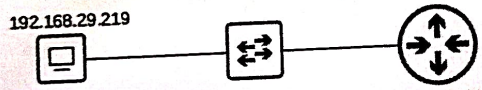
However it's /27 so, let's show the borrowed bits these three purple bits are borrowed & added to the network portion.
Now to find network address we simply need to change all of the host bits to 0

(128 64 32)
11000000 . 10101000 . 00000101 . 00100000
192 . 168 . 5 . 32

So, that's the answer :- 192.168.5.32/27

What subnet does host **192.168.29.219/29** belong to?

Subnet ID: _____/29




Here 192.168.29.219 in binary

11000000 . 10101000 . 00011101 . 11011011
192 . 168 . 29 . 219

its /29 so, let for finding network address simply change all of the host bit to 0.

11000000 . 10101000 . 00011101 . 11011000
192 . 168 . 29 . 216

So, host 192.168.29.219/29 blongs to the subnet 192.168.29.216/29.

# Subnet/Hosts (class c)

| Prefix Length | Number of Subnets | Number of Hosts |
|---|---|---|
| /25 | 2 | 126 |
| /26 | 4 | 62 |
| /27 | 8 | 30 |
| /28 | 16 | 14 |
| /29 | 32 | 6 |
| /30 | 64 | 2 |
| /31 | 128 | 0(2) |
| /32 | 256 | 0(1) |

→ Take NOTE of /31, the number of hosts is 0. its because there is only a single host bit, so there are only two Possible addresse. which are taken by the Network & broadcast address.

For point-to-point connection you can actually use a /31 & assign those two address to each end of the connection & have no network or broadcast addresses.

/32 technically uses all bits for the network address, allowing no hosts, but you can assign a /32 mask to identify a specific host when writing routes & such, & in some other special case.

# Subnetting Class B Networks

In class B there are many more host bits, & therefore many more possible subnets that can be created with a class B network than a class C. The process of subnetting is Exactly same of Class A, B & C networks.

Let's see the Example:-

You have been given the 172.16.0.0/16 network. You are asked to create 80 subnets for your company various LAN's What prefix length should you use?

(/16)

10101100 . 00010000 . 00000000 . 00000000
172 . 16 . 0 . 0

Borrowing 0 bits = can't make any subnet.

(/23)

10101100 . 00010000 . 00000000 . 00000000
172 . 16 . 0 . 0

Borrowing 7 bits = 128 bits

Subnet Mask

11111111 . 11111111 . 11111110 . 00000000
255 . 255 . 254 . 0

So, this is correct answer we should use a /23 prefix length so we can create the 80 subnets we need

128 subnet is more than we need but /22 only allow for 64 which is not enough. I won't show you all 80 subnets of course, but lets look some of the subnets

```
10101100.00010000.00000000.00000000
   172   .   16   .    0    .    0
10101100.00010000.00000010.00000000
   172   .   16   .    2    .    0
10101100.00010000.00000100.00000000
   172   .   16   .    4    .    0
10101100.00010000.00000110.00000000
   172   .   16   .    6    .    0
10101100.00010000.00001000.00000000
   172   .   16   .    8    .    0
```

Example 2: You have been given the 172.18.0.0/16 network your company requires 250 subnets with the same numbers of hosts per subnet. What prefix length should you use?

/24

10101100 . 00010010 . 00000000 . 00000000
172 . 18 . 0 . 0

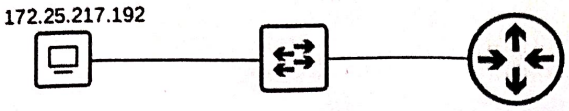
Borrowing 8 bits = 256 ~~hosts~~ subnet

8 host bit = 254 hosts per subnet

# Identify the Subnet

What subnet does host **172.25.217.192/21** belong to?

Subnet ID: _____/21




```
10101100.00011001.11011001.11000000
   172    .   25    .   217   .   192
          ↓
10101100.00011001.11011000.00000000
   172    .   25    .   216   .   0
```

# Subnets/Hosts (class B)

| Prefix Length | Number of Subnets | Number of Hosts |
|---|---|---|
| /17 | 2 | 32766 |
| /18 | 4 | 16382 |
| /19 | 8 | 8190 |
| /20 | 16 | 4094 |
| /21 | 32 | 2044 |
| /22 | 64 | 1022 |
| /23 | 128 | 510 |
| /24 | 256 | 254 |

| Prefix Length | Number of Subnets | Number of Hosts |
|---|---|---|
| /25 | 512 | 126 |
| /26 | 1024 | 62 |
| /27 | 2048 | 30 |
| /28 | 4096 | 14 |
| /29 | 8192 | 6 |
| /30 | 16384 | 2 |
| /31 | 32768 | 0 (2) |
| /32 | 65536 | 0 (1) |

Okay, let's go subnetting class A networks.

| Class | Leading bits | Size of *network number* bit field | Size of *rest* bit field | Number of networks | Addresses per network |
|---|---|---|---|---|---|
| Class A | 0 | 8 | 24 | 128 ($2^7$) | 16,777,216 ($2^{24}$) |
| Class B | 10 | 16 | 16 | 16,384 ($2^{14}$) | 65,536 ($2^{16}$) |
| Class C | 110 | 24 | 8 | 2,097,152 ($2^{21}$) | 256 ($2^8$) |

There are 24 host bits that we can borrow from meaning lots of room to make subnets.

## Subnetting Class A Networks

Que:- You have been given the 10.0.0.0/8 network. You must create 2000 subnets which will be distributed to various enterprises.

What is the prefix length mus you use?

How many host addresses (usable addresses) will be in each subnets?

(18)

00001010 . 00000000 . 00000000 . 00000000
10 . 0 . 0 . 0

Borrowing 0 bits = can't make any subnet.

Subnetmask:

11111111 . 00000000 . 00000000 . 00000000
255 . 0 . 0 . 0

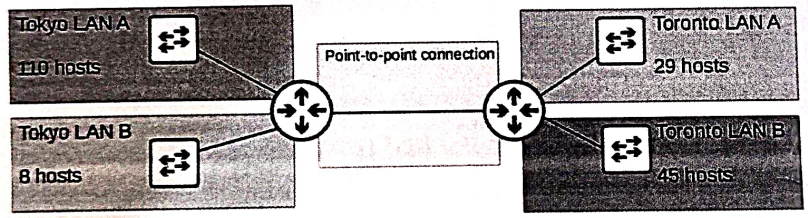
# Variable-Length Subnet Masks (VLSM)

→ Until now, we have practiced subnetting used FLSM (Fixed-Length Subnet Mask).

This mean that all of subnet use the same prefix length (i.e subnetting a class C network into 4 subnet using /26.)

→ However, VLSM (Variable-Length subnet Mask) is the process of creating subnet of different sizes, to make your use of network addresses more efficient.

→ VLSM is more complicated than FLSM, but its easy if you follow the step correctly.




**192.168.1.0/24**

We are assigned the 192.168.1.0/24 network, & must divide it into 5 subnets to provide IP addresses for all hosts in the enterprise network. If we were toy to this with fixed-length subnet mask, we would need to borrow 3 bits to make enough subnet. that would leave 5 host bit left 5 host bit provide only 30 host address. so thats not enough addresses for Tokyo LAN A & Toronto LanB

Subnet size to each LAN, which allow us to make sure each LAN has enough addresses available.

So, What are the steps to subnet using VLSM?

1) Assign the largest subnet at the start of the address space.
2) Assign the second-largest subnet after it
3) Repeat the process until all the subnets have been assigned. from largest to smallest.

That's means we will assign subnets in this order.

① Tokyo LAN A 110 hosts.
② Toronto LAN B 45 hosts.
③ Toronto LAN A 29 hosts.
④ Tokyo LAN B 8 hosts.
⑤ Point to Point connection.

So, lets do Tokyo LAN A :-

Network address: 192.168.1.0/25
Broadcast address: 192.168.1.127/25
first usable address: 192.168.1.1/25
Last " " : 192.168.1.126/25
Total no. of usable host addresses: 126

I decided to use a/25 prefix length
Well, a/25 prefix length 7 host bits
which means a total 128 address, so
126 usable host address.

So,

$2^{What?}$ = at least 2000

Remember, each bit you borrow double the number of subnet you can make.

2, 4, 8, 16, 32, 64, 128, 256, 512

1024, 2048

↳ 11 borrows bits

$2^{11} = 2048$

So, let's borrowing 11 bit that allow us to create the required 2000 subnets

So, we use /19 prefix length.

How many host will be in each subnets?

Well these are 13 host bit remaining

So that means 8190 host per subnets.

13 host bit = $2^{13} - 2$ = 8190 host per subnet

(/19)

00001010 . 00 000000 . 000 00000 . 00000000

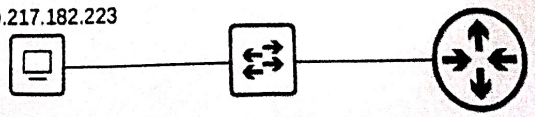
10 . 0 . 0 . 0

Subnet mask:

11111111 . 11111111 . 11100000 . 00000000

255 . 255 . 224 . 0

PC1 has an IP address of **10.217.182.223/11.**
Identify the following for PC1's subnet:
1) Network address:
2) Broadcast address:
3) First usable address:
4) Last usable address:
5) Number of host (usable) addresses:



00001010 . 11011101 . 10110110 . 11011111
10 . 217 . 182 . 223

↓
To find network address
change all of the host bit to 0.

00001010 . 11000000 . 00000000 . 00000000
10 . 192 . 0 . 0

↓
Add 1 to network address
to get first usable address

00001010 . 11000000 . 00000000 . 00000001
10 . 192 . 0 . 1

↓
Change all host bit to 1
& you get broadcast address.

00001010 . 11011111 . 11111111 . 11111111
10 . 223 . 255 . 255

↓
Subtract (1) from broadcast address
& you get last usable address.

00001010 . 11011111 . 11111111 . 11111110
10 . 223 . 255 . 254

↓
To determine the number of
host address count the number
of host bit

00001010 . 11000000 . 00000000 . 00000000
10 . 192 . 0 . 0

21 host bit = $2^{21} - 2 = 2,097,150$ host per subnet.

We need 110, so /25 the correct prefix length.

/25

11000000 . 10101000 . 00000001 . 00000000
192 . 168 . 1 . 0

Network address is 192.168.1.0/25

↓ convert all host bit to 1 & get the broadcast address.

11000000 . 10101000 . 00000001 . 01111111
192 . 168 . 1 . 127

Broadcast address: 192.168.1.127

So, we now have our first subnet 192.168.1.0/25 for Tokyo LAN A ~~That~~ /25 subnet uses up half of the address space of the 192.168.1.0/24 network, but that's no problem.

Using VLSM we ~~have~~ can assign smaller subnets to these other LANs, you'll see that there is enough space left.

Toronto LAN B.

So, 192.168.1.127 = broadcast address of Tokyo LAN A If we add one to that we will get the network address of the next subnet which will be used for Toronto LAN B. Therefore 192.168.1.128 is the network address of Toronto LAN B. What prefix length?

Network address: 192.168.1.128/ ?? → 26
Broadcast " : 192.168.1.191/26
first usable address: 192.168.1.129/26
Last usable address: 192.168.1.190/26
Total no. of usable host addresses: 60

What Prefix length?
To accommodate 45 hosts we will use a /26 prefix length that leaves 6 host bits, which allows for 62 host address & 60 usable address.

/26

11000000 . 10101000 . 00000001 . 10000000
192 . 168 . 1 . 128

↓ convert all host bit to 1 for broad cast address

11000000 . 10101000 . 00000001 . 10111111
192 . 168 . 1 . 191

These two subnet takeup three quarters of the address space, but that's no problem there is still room for more, smaller subnet.

Toronto LAN A

192.168.1.191 = broadcast address of Toronto LAN B
So, 192.168.1.192 = network address of Toronto LAN A.
What is the prefix length:

Network Address: 192.168.1.192/??→27
Broadcast " : 192.168.1.223/27
first usable address: 192.168.1.193/27
Last " " : 192.168.1.222/27
Total no. of usable host address: 30

Toronto LAN A required 29 host, so we should use a /27 prefix length which leaves 5 host bit & 30 host address means 28 usable.

11000000 . 10101000 . 00000001 . 11000000
192 . 168 . 1 . 192

↓ → Broadcast address.

11000000 . 10101000 . 00000001 . 11011111
192 . 168 . 1 . 223

Tokyo LAN B:

192.168.1.223 = broadcast address of Toronto LAN A.
192.168.1.224 = network address of Tokyo LAN B.

Network Address: 192.168.1.224/28
Broadcast Addres: 192.168.1.239/28
first usable Addres: 192.168.1.225/28
Last " " : 192.168.1.238/28
Total no. of usable host address: 14

So, we used address space all the way up to 192.168.1.239
These not a lot of space left but that's no problem.
Oneshwar:
Point to Point connection only required 2 address.

192.168.1.239 = broadcast address of Tokyo LAN B
192.168.1.240 = network address of point to point.

What prefix length?

So, as I mentioned earlier in the first subnetting, it is possible to use a /31 prefix length for a subnet required only two hosts.

However, for the CCNA test if you are asked what prefix length to use for a subnet that requires two hosts, I recommend NOT using a /31.

Instead, what other prefix length allows for 2 hosts?

So, a /30 prefix length allow for 2 hosts

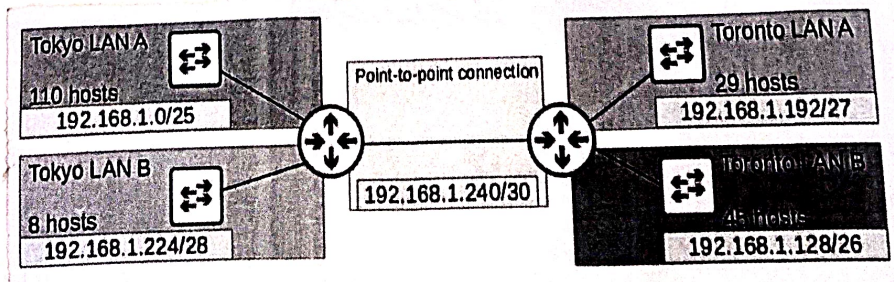
/30

11000000 . 10101000 . 00000001 . 11110000
192 . 168 . 1 . 240 → Network add...

↓ → broadcast address

11000000 . 10101000 . 00000001 . 11110011
192 . 168 . 1 . 243

N/A: 192.168.1.240/30
B/A 192.168.1.243/30
FUA: 192.168.1.241/30
LUA: 192.168.1.242/30
Total no. of usable host addres: 2




192.168.1.0/24