# Internet Protocol Version4 (IPv4) & Subnetting Strategy

BY
Hayder Ayad Dawood
Mustafa Salam
Ali Adil Saeid
Department of Computer Science
University of Technology

# The Internet Protocol (IP)

- The Internet Protocol is the corner-stone of the TCP/IP architecture. All computers in the Internet understand IP.
- The main tasks of IP are:
  - The **addressing** of the computers, and the **fragmentation** of packets.

**There are two types of Internet Protocol:**

- **Internet Protocol version 4 (IPv4):** currently used version of Internet Protocol.

- **Internet Protocol version 6 (IPv6):** the upcoming replacement for IPv4. It contains some major improvements and new features.

# IPv4 Address:

- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.
  - Connectionless protocol
  - Fragments (divides) packets where necessary
  - Addressing via 32 bit Internet addresses

- However, it contains no functions for end-to-end message reliability or flow control. IP makes the 'best effort' to forward packets to the next destination, but **does not guarantee delivery because it is connectionless.**

# Cont…

**The IP address can be classify into two classes:**

- **Public address:**

This address considered as any valid address assigned to any user, and the organization who is responsible for registering IP ranges called Internet Service Providers (ISPs), and this address will be unique.

- **Private Address:**

Any number or address assigned to a device on a private TCP/IP Local Area Network that is accessible only within the Local Area Network.

# Internet Protocol Version 6 (IPv6)

- Maintains good features of IPv4, discards bad ones.

- Not compatible with IPv4

- Compatible with all other Internet protocols including TCP, UDP, ICMP, DNS, etc.

- Main features:

– Long addresses (128 bits) ⇒ supports billions of hosts.
– Simplified, fixed size header ⇒ routers can process packets faster.
– Support for authentication and privacy
– Better support for type of service.

# Internet addresses

Each network interface connected to the Internet has a unique address consisting of two parts:

- **Network address,** address of the network within the Internet (used by gateways for routing IP packets between networks).
- **Host address,** address of the computer within the network (used for delivering packets to a particular network interface within the network).

# Internet address format

- The 32-bit IP address is separated into four 8-bit octets, allowing each octet to have a value ranging from 0 to 255.

- Furthermore, the IP address is logically separated into two distinct components: the network ID and the host ID. The network ID is used to identify the subnet upon which the host resides. The host ID is used to identify the host itself within the given subnet.

> **IP addresses can be displayed in three typical formats:**

- Binary notation Binary notation is the format that systems on the network use to process the address. An example of binary notation is **1100000.10101000.0000001.01100100.**
- Hexadecimal notation Hexadecimal notation is the format typically used when identifying IPv6 addresses. An example of hexadecimal notation of an IPv4 address is **C0.A8.01.64**

# Cont…

- Dotted-decimal notation Dotted-decimal notation is the format that is typically used for displaying the IP address in a human-readable format. An example of dotted-decimal notation is **192.168.1.100**

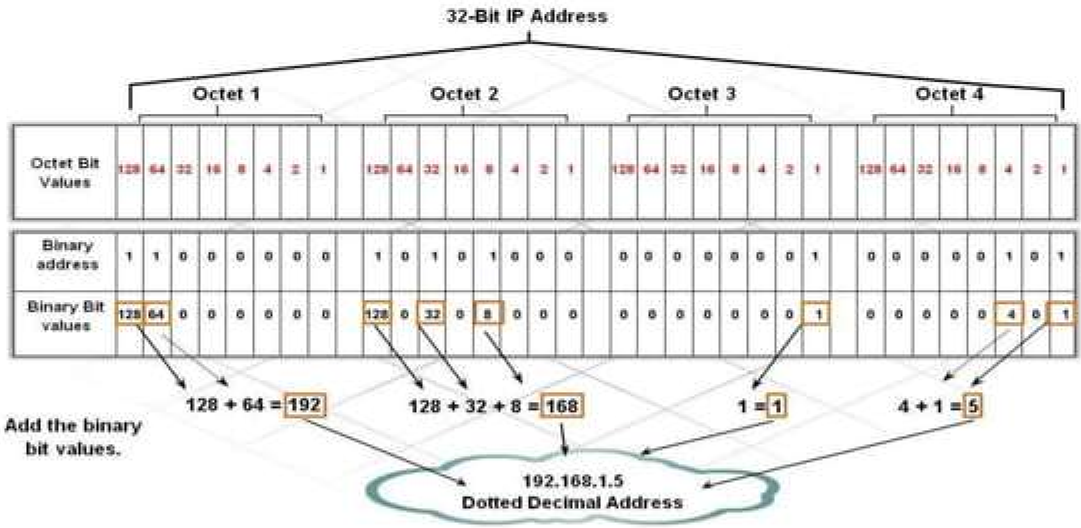



Figure 3: Structure of IP Address

# Classes of IP addresses

- Different networks have different sizes. Basically, there are many small networks and few large networks.

- To provide efficient use of 32-bit address space, IPv4 defined several **address classes** and associated address formats:

➢ **Class A**: allows 128 networks, 16 million hosts each.

The IP address start from 1.0.0.0 to 127.255.255.255, and the mask address is **255.0.0.0**

➢ **Class B:** allows 16,382 networks, 65,534 hosts each.

The IP address start from 128.0.0.0 to 191.255.255.255, and the mask address is **255.255.0.0**

# Classes of IP addresses

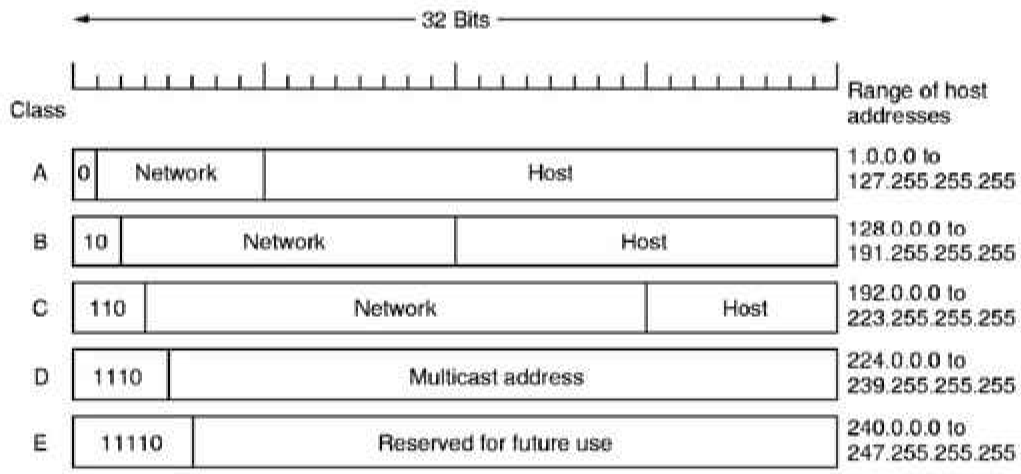
- **Class C:** allows 2 million networks, 254 hosts each.

  The IP address start from 192.0.0.0 to 223.255.255.255, and the mask address is **255.255.255.0**

- **Class D:** multicast networks The IP address start from 224.0.0.0 to 239.255.255.255.
- **Class E:** reserved for future use. From 240 to 255 and the 255.255.255.255 used for broadcast to all the subnet.

# Cont..

- One of the benefits of classful addresses is that they provide a hierarchy to the network through the use of the network ID. This translates into an efficient routing environment because it is easy for a router to determine what networks can be grouped together and treated as a single routing entry.



# Strategies to Conserve Addresses

- Several strategies have been developed and implemented to help the Internet community on how provides a good managing of IP addresses. These strategies help reduce the load on Internet routers and help administrators use globally unique IP addresses more efficiently. There are two common strategies, which are:

- Private Addressing
- Classless Inter-Domain Routing (CIDR)

# Private Address

- It means If the internetwork is limited to one organization, the IP addresses need only be unique within that organization. Only networks that interface with public networks such as the Internet need public addresses. Using public addresses on the outside and private addresses for inside networks is very effective.

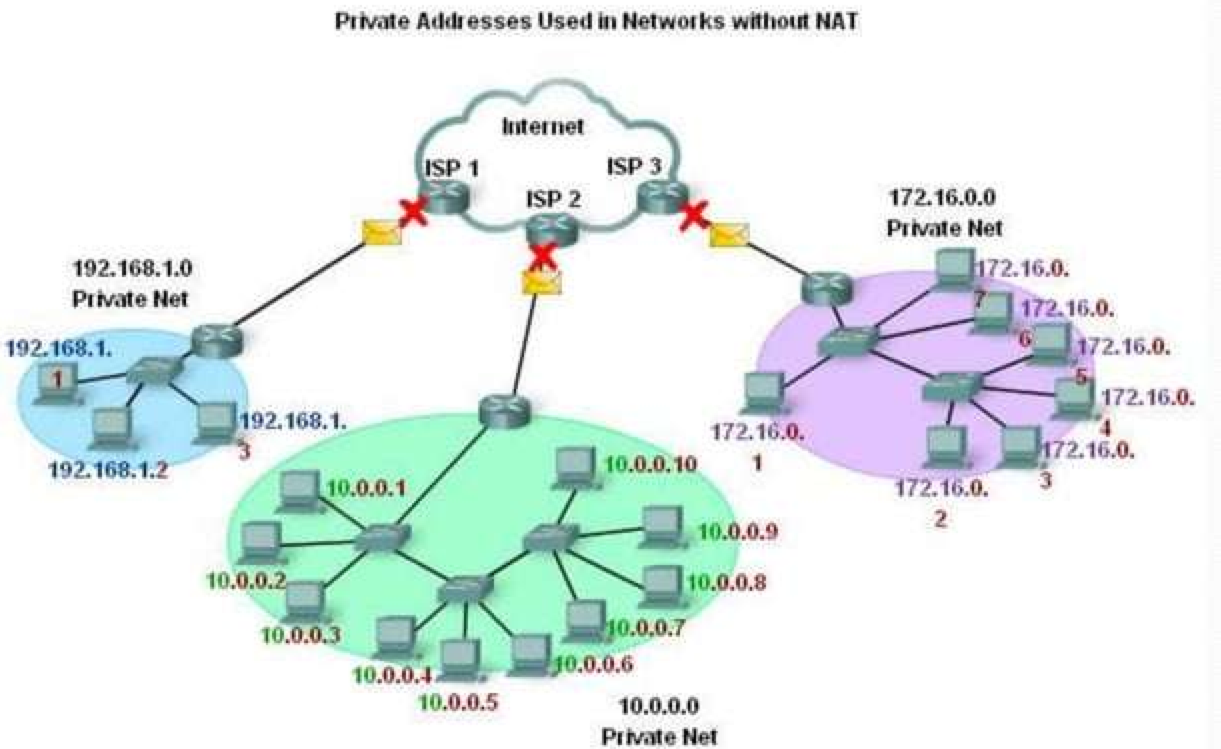
**Private Addresses:-**

RFC1918 designates three ranges of IP addresses as private:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

| Address Block | Classful Equivalent | Prefix Length | Number of Addresses |
|---|---|---|---|
| 10.0.0.0–10.255.255.255 | 1 Class A<br>256 Class B<br>65,536 Class C | /8 | 16,777,216 |
| 172.16.0.0–172.31.255.255 | 16 Class B<br>4,096 Class C | /12 | 1,048,576 |
| 192.168.0.0–192.168.255.255 | 1 Class B<br>256 Class C | /16 | 65,536 |

# Cont…



Private Addresses Used in Networks without NAT
Internet
ISP 1
ISP 2
ISP 3
192.168.1.0
Private Net
192.168.1.
1
192.168.1.2
192.168.1.
3
172.16.0.0
Private Net
172.16.0.
7
172.16.0.
6
172.16.0.
5
172.16.0.
4
172.16.0.
3
172.16.0.
2
172.16.0.
1
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7
10.0.0.8
10.0.0.9
10.0.0.10
10.0.0.0
Private Net

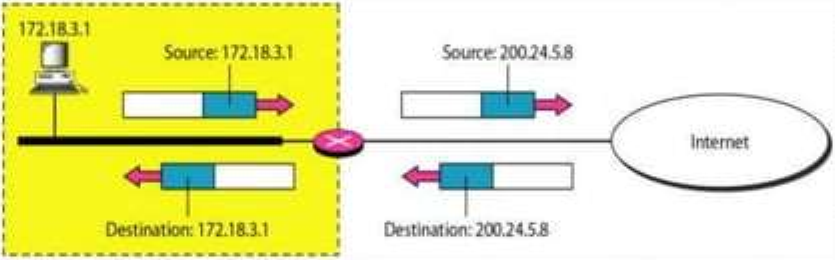
# Cont...

- There are two ways to convert the private address to public address:

1. **Network Address Translation (NAT).**

This technique has been used to convert the private address to public address, the **NAT** allowing us to access the internet and get services. The basic idea, is that technique used pool of public addresses and assign for each private address one public address. Thus, this way is inefficient due to the fact, that there are cost and delay associated with this operation. The table and the figure below show how the NAT make the mapping.

| Private Address | Public Address |
|---|---|
| 192.68.5.1 | 200.1.1.2 |
| 192.68.5.2 | 200.1.1.3 |
| 192.68.5.3 | 200.1.1.4 |



# Cont..

## 2. Port Address Translation (PAT).

It's another technique used to convert the private address to public. During PAT, each computer on LAN is translated to the same IP address (public), but with a different port number assignment. This way is much better than the NAT because we can use one public address to translate any private address, therefore we saved the cost. The table below shows the process of the PAT

| Private Address | Port address | Public Address | Port Number |
|---|---|---|---|
| 192.68.1.3 | 50133 | 200.0.1.2 | 1 |
| 192.68.1.5 | 63887 | 200.0.1.2 | 2 |

- The port address will be any random number in the allowed range, that the device created it when wants to access the internet.
- The packet will contain the port number that assign to the device that wants to access the internet and through this port number, the router when get the response message will translate it and make map this message to it is private address based on the port number.

# Addressing without Subnets

- If we have a class B with a Flat Network, the number of host will be more than $2^{16}$=65536 hosts,
- So the problem is here, that managing this network with this number of host is too tricky and the performance of this network will get down because of the heavy load. In other word, any single broadcast can slowdown the network.

- Therefore, the solution is the subnetting. Subnetting means divide or separate the single network into multiple networks that can reduce the loading from one network.
- The advantage of using subnetting is:-

  1. Reduce the traffic and the increase the performance.
  2. The smaller network can easier to manage.

# Subnetting

- As the number of distinct local networks grows, managing them become a serious headache. Every time a new network is installed the system administrator must contact NIC to get a new network number.

- The solution to the problem is to allow a network to be split into several independent parts for internal use but still act like a single network to the outside world. In the internet literature these parts are called **subnets**.

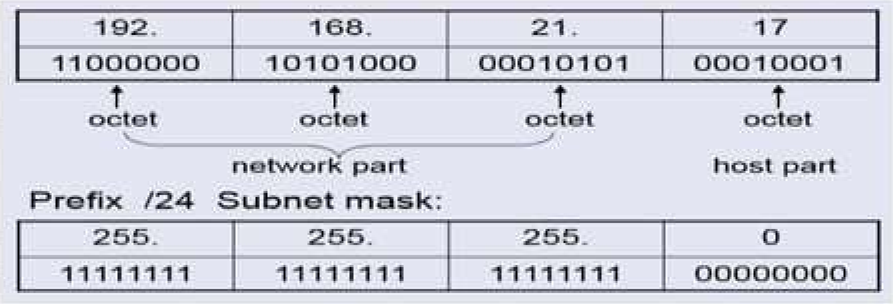
# Subnet masks:-

- A mask is a 32-bit binary number that is expressed in dotted decimal notation. By default, a mask contains two fields, the network field and the host field. These correspond to the network number and the locally administered part of the network address. When an administrator subnets, they are adjusting the way they view the IP address. Table 1: Default masks for classful addressing

| Address Class | Bits Used for Subnet Mask | Dotted Decimal Notation |
|---|---|---|
| Class A | 11111111 00000000 00000000 00000000 | 255.0.0.0 |
| Class B | 11111111 11111111 00000000 00000000 | 255.255.0.0 |
| Class C | 11111111 11111111 11111111 00000000 | 255.255.255.0 |

# Cont…

- Routers and hosts still assume class subnet masks by default:
- Class A /8 255.0.0.0
- Class B /16 255.255.0.0
- Class C /24 255.255.255.0
- 
- The figure below gives an example to class C mask address:




The first three octets represent the network part and the last octet represent the host part.

# Cont...

- There three important things that should be taken into our account when we thinking about subnetting:-

1. Network address – the first one
2. Broadcast address – the last one
3. Host addresses – everything in between

As well as, to find the number of hosts per subnet. We can use formal $2^x$ -2, where (x) is the number of unmasked bits (0's) .

# Cont...

- For example, in 11000000, the number of zeros gives us $2^6 - 2 = 62$ hosts. In this example, there are 62 hosts per subnet and we make subtract because the first IP address reserve for the network address and the last one for the network broadcast.

- While when we want to find number of networks, we can use this formal $2^y$

  Where Y represent the number of masked bits, (1's). For example, in 11000000, the number of ones gives us $2^2 = 4$

# Reserved and Restricted Addresses

- In any subnet, there are certain addresses that cannot be assigned to an individual device because they have a special purpose. The subnet address is the first address in a range that identifies the subnet. The broadcast address is the last address in the range, and all hosts on the subnet receive traffic if anything is sent to it.

- Assume that a subnet address is **172.31.9.0** with a mask of **255.255.255.0**. The subnet address is **172.31.9.0**, and the broadcast address is **172.31.9.255**.

# Classless Inter-Domain Routing (CIDR)

- Classless Inter Domain Routing (CIDR) is a method for assigning IP addresses without using the standard IP address classes like Class A, Class B or Class C. In CIDR , an IP address is represented as **A.B.C.D /n**, where "/n" is called the IP prefix or network prefix. The IP prefix identifies the number of significant bits used to identify a network.

- Example, 192.9.205.22 /18 means, the first 18 bits are used to represent the network and the remaining 14 bits are used to identify hosts.

- It's basically the method that ISPs (Internet Service Providers) use to allocate an amount of addresses to a company, a home—a customer. They provide addresses in a certain block size

- When you receive a block of addresses from an ISP, what you get will look something like this: 192.168.10.32/28. This is telling you what your subnet mask is. The slash notation (/) means how many bits are turned on (1s).

- The Class A default subnet mask, which is 255.0.0.0. This means that the first byte of the subnet mask is all ones (1s), or 11111111. When referring to a slash notation, you need to count all the 1s bits to figure out your mask. The 255.0.0.0 is considered a /8 because it has 8 bits that are 1s—that is, 8 bits that are turned on

# Subnetting Class C Addresses

- ✓In a Class C address, only 8 bits are available for defining the hosts
- ✓ that subnet bits start at the left and go to the right without skipping bits. This means that the only Class C subnet masks can be the following:

```
Binary       Decimal   CIDR
---------------------------------------------
10000000 = 128        /25
11000000 = 192        /26
11100000 = 224        /27
11110000 = 240        /28
11111000 = 248        /29
11111100 = 252        /30
```

## *The Fast Way Method: Subnetting a Class C Address*

start by using the second subnet mask available with a Class C address, which borrows 2 bits for subnetting     192 = 11000000

The 1s represent the subnet bits, and the 0's represent the host bits available in each subnet. 192 provides 2 bits for subnetting and 6 bits for defining the hosts in each subnet.

| |
|---|
| 00000000 = 0 |
| 01000000 = 64 |
| 10000000 = 128 |
| 11000000 = 192 |

**Example 255.255.255.192 (/26)**

Let's use the Class C subnet mask from the preceding example, 255.255.255.192, to see how much simpler this method is than writing out the binary numbers. We're going to subnet the network address 192.168.10.0 and subnet mask 255.255.255.192.

192.168.10.0 = Network address

255.255.255.192 = Subnet mask

Now, let's answer the big five:

- *How many subnets?* Since 192 is 2 bits on (11000000), the answer would be $2^2$.
- *How many hosts per subnet?* We have 6 host bits off (11000000), so the equation would be $2^6 - 2 = 62$ hosts.
- *What are the valid subnets?* 256 – 192 = 64. Remember, we start at zero and count in our block size, so our subnets are 0, 64, 128, and 192.

## Example 255.255.255.224 (/27)

This time, we'll subnet the network address 192.168.10.0 and subnet mask 255.255.255.224.

192.168.10.0 = Network address

255.255.255.224 = Subnet mask

- *How many subnets?* 224 is 11100000, so our equation would be $2^3 = 8$.
- *How many hosts?* $2^5 - 2 = 30$.
- *What are the valid subnets?* 256 – 224 = 32. We just start at zero and count to the subnet mask value in blocks (increments) of 32: 0, 32, 64, 96, 128, 160, 192, 224.

# Subnetting Class B Addresses

Class B subnet masks first. Notice that we have a lot more possible subnet masks than we do with a Class C network

| | | | |
|---|---|---|---|
| 255.255.128.0 | (/17) | 255.255.255.0 | (/24) |
| 255.255.192.0 | (/18) | 255.255.255.128 | (/25) |
| 255.255.224.0 | (/19) | 255.255.255.192 | (/26) |
| 255.255.240.0 | (/20) | 255.255.255.224 | (/27) |
| 255.255.248.0 | (/21) | 255.255.255.240 | (/28) |
| 255.255.252.0 | (/22) | 255.255.255.248 | (/29) |
| 255.255.254.0 | (/23) | 255.255.255.252 | (/30) |

We know the Class B network address has 16 bits available for host addressing. This means we can use up to 14 bits for subnetting (because we have to leave at least 2 bits for host addressing).

The process of subnetting a Class B network is pretty much the same as it is for a Class C, except that you just have more host bits. Use the same subnet numbers for the third octet with Class B that you used for the fourth octet with Class C, but add a zero to the network portion

Example 255.255.192.0 (/18)
172.16.0.0 = Network address
255.255.192.0 = Subnet mask

- *Subnets?* $2^2 = 4$.
- *Hosts?* $2^{14} - 2 = 16{,}382$ (6 bits in the third octet, and 8 in the fourth).
- *Valid subnets?* 256 – 192 = 64. 0, 64, 128, 192. Remember the subnetting is performed in the third octet, so the subnet numbers are really 0.0, 64.0, 128.0, and 192.0, as shown in the next table.

| | | | | |
|---|---|---|---|---|
| **Subnet** | 0.0 | 64.0 | 128.0 | 192.0 |
| **First host** | 0.1 | 64.1 | 128.1 | 192.1 |
| **Last host** | 63.254 | 127.254 | 191.254 | 255.254 |

**Example 255.255.240.0 (/20)**

172.16.0.0 = Network address

255.255.240.0 = Subnet mask

- *Subnets?* $2^4 = 16$.
- *Hosts?* $2^{12} - 2 = 4094$.
- *Valid subnets?* 256 – 240 = 0, 16, 32, 48, etc., up to 240. Notice that these are the same numbers as a Class C 240 mask.

The following table shows the first four subnets, valid hosts, and broadcast addresses in a Class B 255.255.240.0 mask:

| | | | | |
|---|---|---|---|---|
| **Subnet** | 0.0 | 16.0 | 32.0 | 48.0 |
| **First host** | 0.1 | 16.1 | 32.1 | 48.1 |
| **Last host** | 15.254 | 31.254 | 47.254 | 63.254 |

# Subnetting Class A Addresses

Class A subnetting is not performed any differently from Classes B and C, but there are 24 bits to play with instead of the 16 in a Class B address and the 8 in a Class C address. Let's start by listing all the Class A subnets.

```
255.128.0.0  (/9)          255.255.240.0  (/20)
255.192.0.0  (/10)         255.255.248.0  (/21)
255.224.0.0  (/11)         255.255.252.0  (/22)
255.240.0.0  (/12)         255.255.254.0  (/23)
255.248.0.0  (/13)         255.255.255.0  (/24)
255.252.0.0  (/14)         255.255.255.128  (/25)
255.254.0.0  (/15)         255.255.255.192  (/26)
255.255.0.0  (/16)         255.255.255.224  (/27)
255.255.128.0  (/17)       255.255.255.240  (/28)
255.255.192.0  (/18)       255.255.255.248  (/29)
255.255.224.0  (/19)       255.255.255.252  (/30)
```

You must leave at least 2 bits for defining hosts. And I hope you can see the pattern by now. Remember, we're going to do this the same way as a Class B or C subnet.

**Example** **255.255.240.0 (/20)**

255.255.240.0 gives us 12 bits of subnetting and leaves us 12 bits for host addressing.

- *Subnets?* $2^{12} = 4096$.
- *Hosts?* $2^{12} - 2 = 4094$.
- *Valid subnets?* $256 - 240 = 16$. The subnets in the second octet are a block size of 1 and the subnets in the third octet are 0, 16, 32, etc.

The following table shows some examples of the host ranges—the first three and the last subnets:

| | | | | | |
|---|---|---|---|---|---|
| Subnet | 10.0.0.0 | 10.0.16.0 | 10.0.32.0 | ... | 10.255.240.0 |
| First host | 10.0.0.1 | 10.0.16.1 | 10.0.32.1 | ... | 10.255.240.1 |
| Last host | 10.0.15.254 | 10.0.31.254 | 10.0.47.254 | ... | 10.255.255.254 |

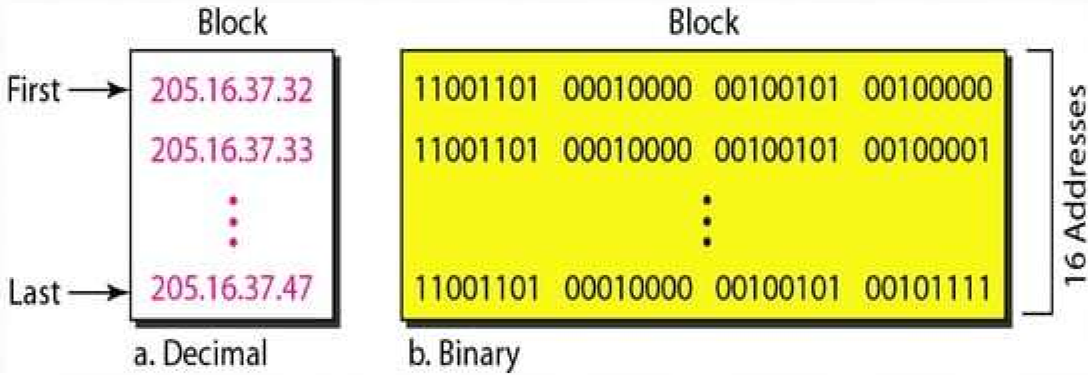
# Address Blocks

- To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks

- In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

  - The addresses in a block must be contiguous, one after another
  - The number of addresses in a block must be a power of 2 (1,2,8..)
  - The first address must be evenly divisible by the number of addresses

# Cont…




We can see that the restrictions are applied to this block. The addresses are contiguous. The number 4 of addresses is a power of 2 =16, and the first address is divisible by 16. The first address, when converted to a decimal number, is 3,440,387,360, which when divided by 16 results in 215,024,210

# Mask and Address Blocks

- In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.
- The first address in the block can be found by setting the rightmost $32 - n$ bits to 0s
- The last address in the block can be found by setting the rightmost $32 - n$ bits to 1s
- The number of addresses in the block can be found by using the formula $2^{32-n}$

# Mask and Address Blocks

- Example: 205.16.37.39/28
  - The binary representation is 1100110 00010000 00100101 00100111
  - If we set 32 − 28 rightmost bits to 0, we get 11001101 00010000 00100101 00100000

    → 205.16.37.32 (First address)
  - If we set 32 − 28 rightmost bits to 1, we get 11001101 00010000 00100101 00101111

    → 205.16.37.47 (Last address)
  - The value of n is 28, which means that number of addresses is $2^{32-28}$ or 16

# Cont…

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32-bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information.

Example 205.16.37.39/28 can be represented as 11111111 11111111 11111111 11110000 (twenty-eight 1's and four 0's). Find

a. The first address
b. The last address
c. The number of addresses

# Cont…

Solution

a. The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.

| | |
|---|---|
| Address: | 11001101 00010000 00100101 00100111 |
| Mask: | 11111111 11111111 11111111 11110000 |
| First address: | 11001101 00010000 00100101 00100000 |

b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.

| | |
|---|---|
| Address: | 11001101 00010000 00100101 00100111 |
| Mask complement: | 00000000 00000000 00000000 00001111 |
| Last address: | 11001101 00010000 *00100101* *00101111* |

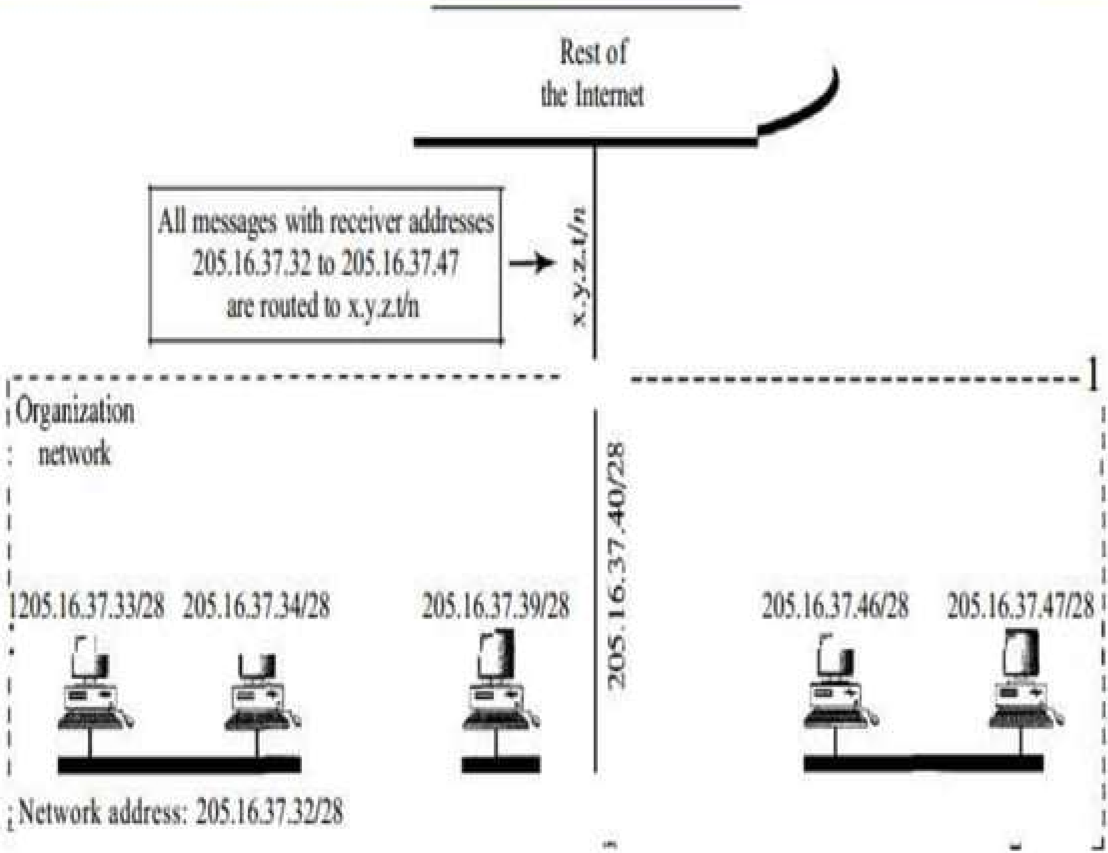
c. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

| | |
|---|---|
| Mask complement: | 000000000 00000000 00000000 00001111 |
| Number of addresses: | 15 + 1 = 16 |

# Network Addresses

A very important concept in IP addressing is the network address. When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet. The first address in the class, however, is normally (not always) treated as a special address. The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world.

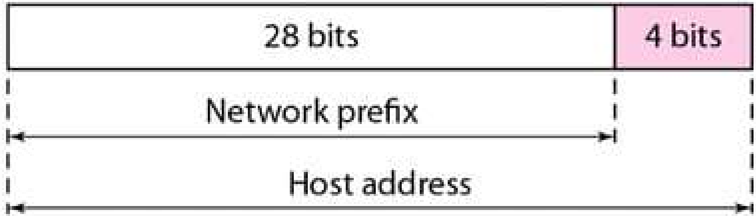
As well as, we will see that the first address is the one that is used by routers to direct the message sent to the organization from the outside the organization network is connected to the Internet via a router. The router has two addresses. One belongs to the granted block; the other belongs to the network that is at the other side of the router. We call the second address x.y.z.t/n because we do not know anything about the network it is connected to at the other side. All messages destined for addresses in the organization block (205.16.37.32 to 205.16.37.47) are sent, directly or indirectly, to x.y.z.t/n. We say directly or indirectly because we do not know the structure of the network to which the other side of the router is connected


Rest of
the Internet
All messages with receiver addresses
205.16.37.32 to 205.16.37.47
are routed to x.y.z.t/n
x.y.z.t/n
Organization
network
205.16.37.40/28
205.16.37.33/28
205.16.37.34/28
205.16.37.39/28
205.16.37.46/28
205.16.37.47/28
Network address: 205.16.37.32/28

# Two-Level Hierarchy: No Subnetting

➢ An IP address can define only two levels of hierarchy when <u>not subnetted</u>. The n left- most bits of the address x.y.z.tJn define the network (organization network); the 32 - n

➢Rightmost bits define the particular host (computer or router) to the network. The two common terms are prefix and suffix. The part of the address that defines the network is called the prefix



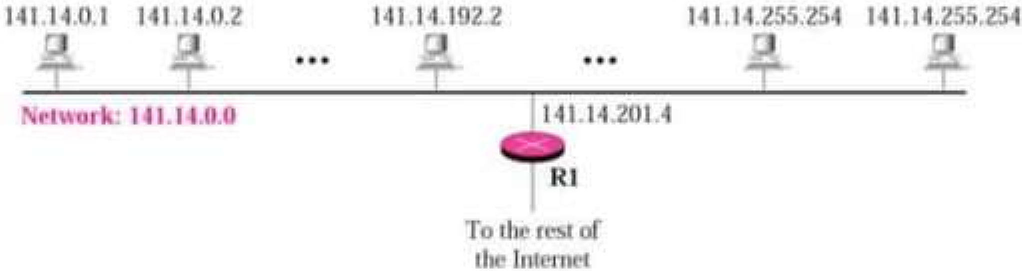
# Three-Levels of Hierarchy: Subnetting

An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets. The rest of the world still sees the organization as one entity.

However, internally there are several subnets. All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.

The organization, however, needs to create small sub-blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also has its own

# A network with two levels of hierarchy (not subnetted)

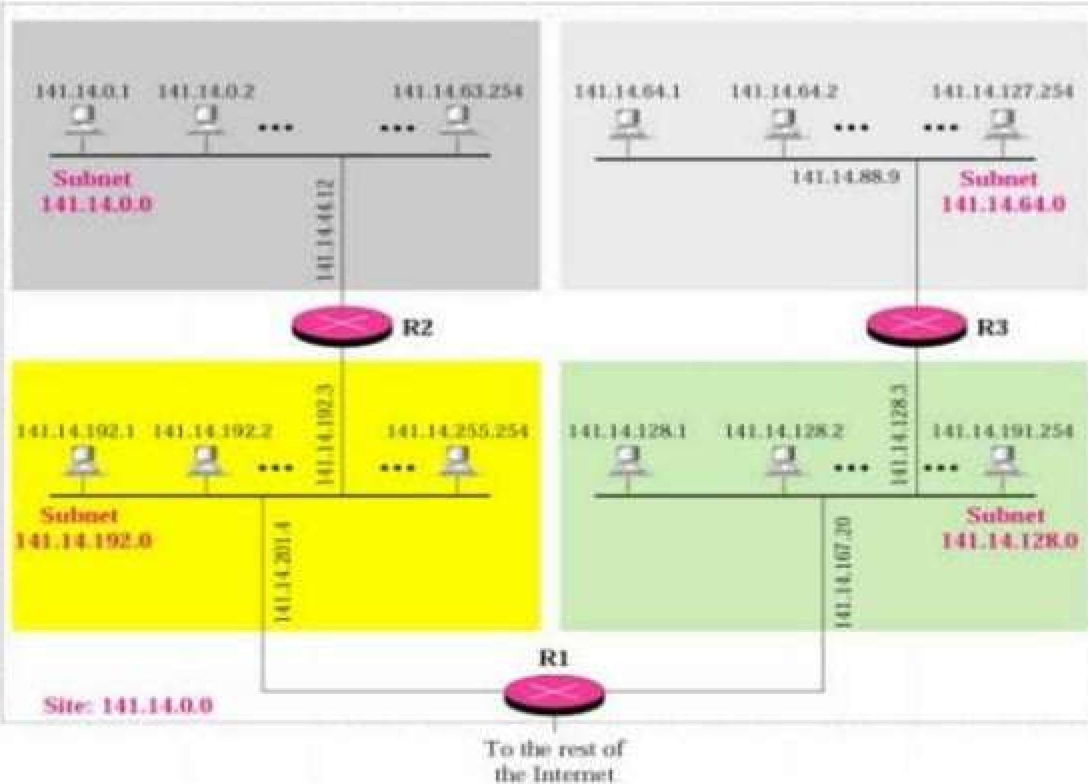



The network above (network 141.14.0.0) uses class B addressing, it has therefore 254x254 = 64516 hosts.

A LAN with 64516 hosts is too big. An additional level of hierarchy is required in order to brake the large number of hosts into several smaller groups. For example, we can brake the hosts into four groups (subnets):

Subnet 141.14.0.0 has hosts 141.14.0.1 ... 141.14.63.254
Subnet 141.14.64.0 has hosts 141.14.64.1 ... 141.14.127.254
Subnet 141.14.128.0 has hosts 141.14.128.1 ... 141.14.191.254
Subnet 141.14.192.0 has hosts 141.14.192.1 ... 141.14.255.254

# A network with three levels of hierarchy (subnetted)



## Example 1

**IP address: 130.45.34.56**
**Mask: 255.255.240.0 What is the subnet address?**

| | | | | |
|---|---|---|---|---|
| IP = | 10000010 | 00101101 | 00100010 | 00111000 |
| M = | 11111111 | 11111111 | 11110000 | 00000000 |
| && = | 10000010 | 00101101 | 00100000 | 00000000 |
| | 130 | 45 | 32 | 0 |

The subnetwork address is **130.45.32.0**.

## Finding the subnetwork address

*Straight Method*

Convert IP address into binary form,
AND with the mask,
convert to dot-decimal form

*Short-Cut Method:*

If the byte in the mask is 255, copy the byte in the address.

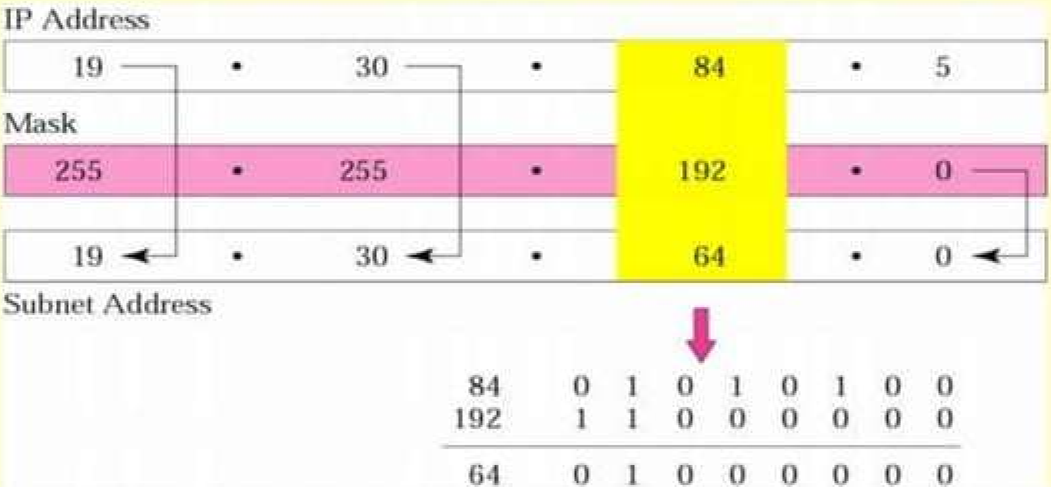
If the byte in the mask is 0, replace the byte in the address with 0.

If the byte in the mask is neither 255 nor 0, write the mask and the address in binary and apply the AND operation (as above).

# Example 2

**IP = 19.30.80.5**

**M = 255.255.192.0** **What is the subnet address?**

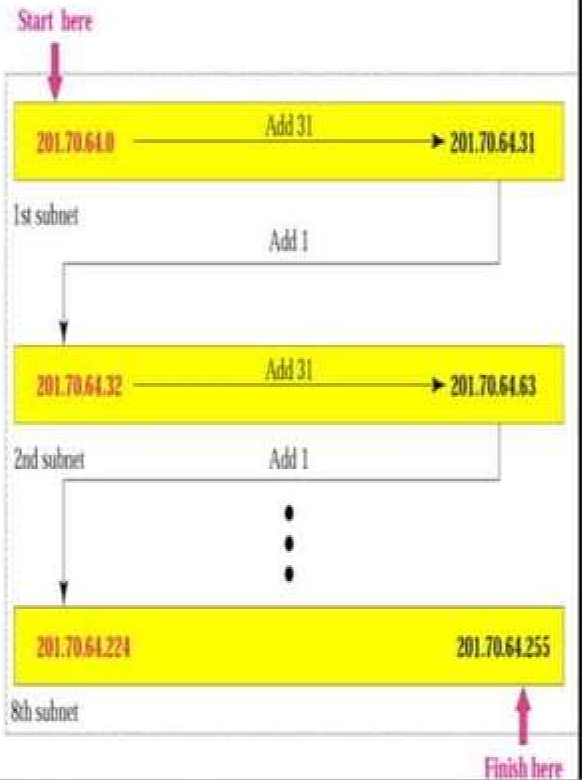


## Example 3

A company is granted the site address 201.70.64.0 (class C). The company needs six subnets. Design the subnets.

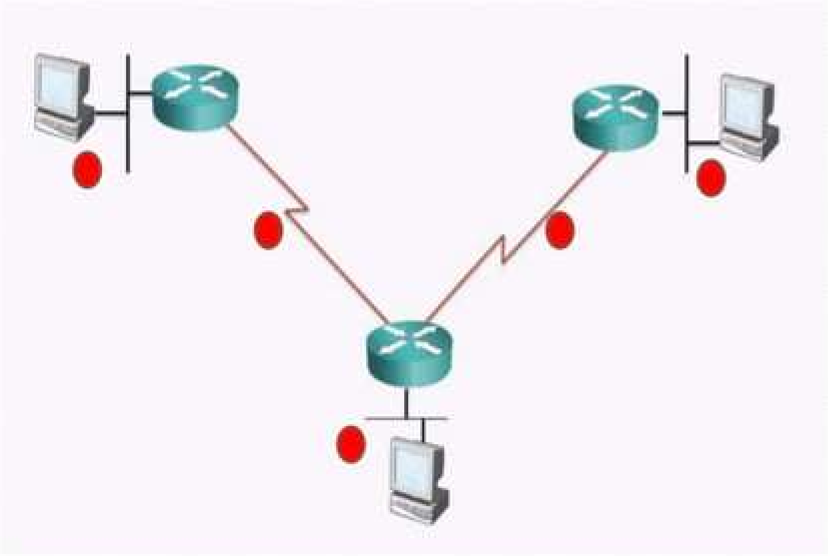
### Solution

- The number of 1s in the default mask is 24 (class C).
- The company needs six subnets. This number 6 is not a power of 2. The next number that is a power of 2 is 8 ($2^3$). We need 3 more 1s in the subnet mask. The total number of 1s in the subnet mask is 27 (24 + 3).
- The total number of 0s is 5 (32 - 27). The mask is
  11111111 11111111 11111111 11100000 or
  **255.255.255.224**
- The number of subnets is 8.
- The number of addresses in each subnet is $2^5$ (5 is the number of 0s) or 32.



# Example 4

An organization has purchased the Class C Address **216.21.5.0** and would like to use it to address the following **5 networks**.



# Cont…

**IP: 216.21.5.0**
**Subnet Mask: 255.255.255.0**

-----------------------------------------

**Networks number = 5**
**First: Convert the number of networks to binary.**

5 = 00000101

3 bits

**Second: Reserve bits in subnet mask and find your increment.**

**Subnet Mask: 255.255.255.0 = 11111111.11111111.11111111.00000000**
**11111111.11111111.11111111.11100000**

3 bits

5 bits

**No. of the networks= 2^number of 1's = 2^3 = 8] Networks**
**No. of the hosts= 2^number of 0's -2 = [2^5 = 32 – (2) =30] Hosts number**

**The increment value:-**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Bit Values in a Octet: | 128 | 64 | **32** | 16 | 8 | 4 | 2 | 1 | |
| | 1 | 1 | **1** | 0 | 0 | 0 | 0 | 0 | so our increment value is 32 |

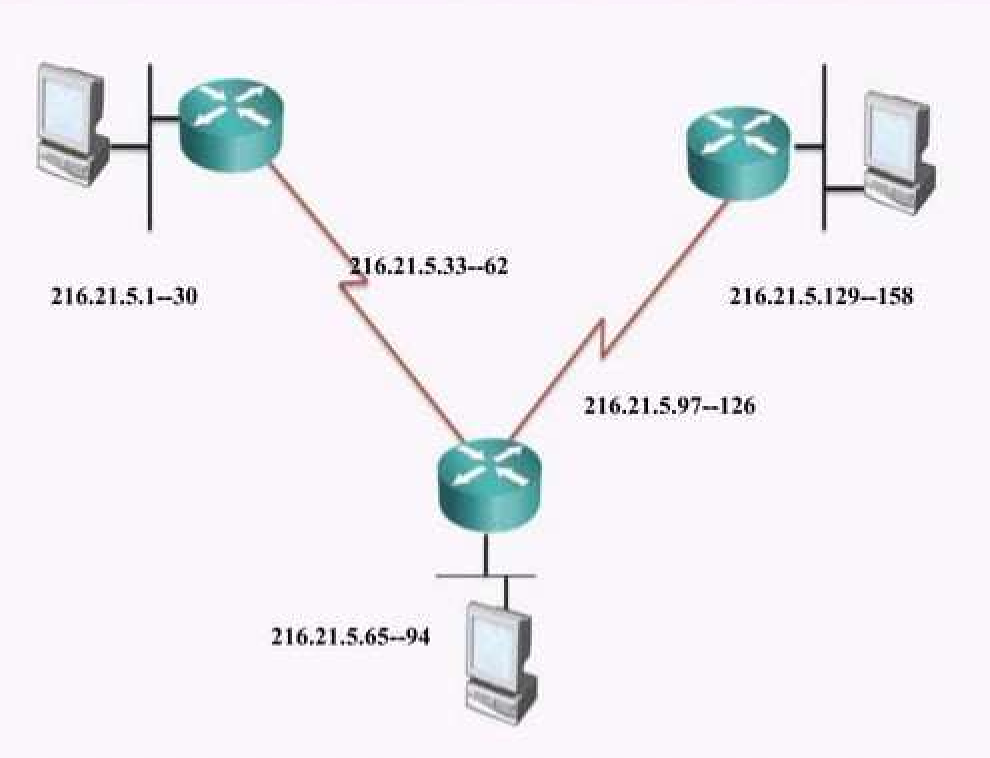
Lowest incrementing value

# Cont…

- Third: Use increment to find the network ranges.

| No. | Subnet address | First valid address | Last valid Address | Broadcast address |
|---|---|---|---|---|
| 1 | 216.21.5.0 | 216.21.5.1 | 216.21.5.30 | 216.21.5.31 |
| 2 | 216.21.5.32 | 216.21.5.33 | 216.21.5.62 | 216.21.5.63 |
| 3 | 216.21.5.64 | 216.21.5.65 | 216.21.5.94 | 216.21.5.95 |
| 4 | 216.21.5.96 | 216.21.5.97 | 216.21.5.126 | 216.21.5.127 |
| 5 | 216.21.5.128 | 216.21.5.129 | 216.21.5.158 | 216.21.5.159 |
| 6 | 216.21.5.160 | 216.21.5.161 | 216.21.5.190 | 216.21.5.191 |
| 7 | 216.21.5.192 | 216.21.5.193 | 216.21.5.222 | 216.21.5.223 |
| 8 | 216.21.5.224 | 216.21.5.225 | 216.21.5.254 | 216.21.5.255 |

# Cont…



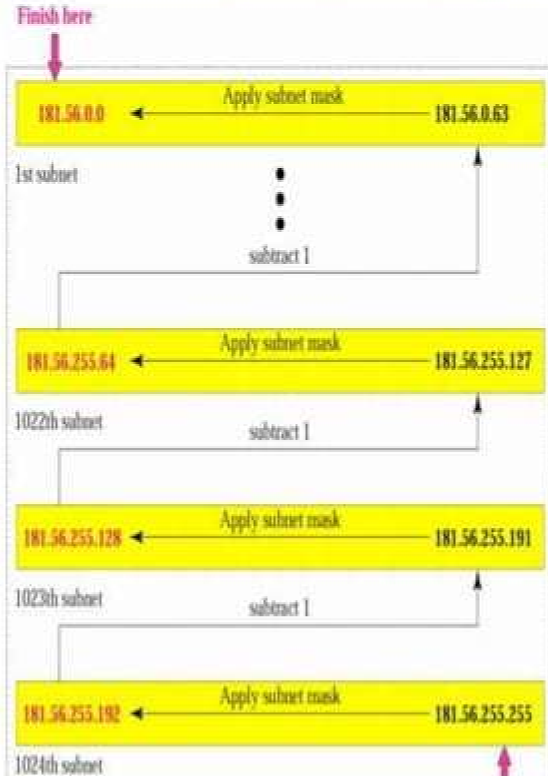
216.21.5.33--62
216.21.5.1--30
216.21.5.129--158
216.21.5.97--126
216.21.5.65--94

## Example 5

A company is granted the site address 181.56.0.0 (class B). The company needs 1000 subnets. Design the subnets.

***Solution***

- The company needs 1000 subnets. This number is not a power of 2. The next number that is a power of 2 is 1024 ($2^{10}$). We need 10 more 1s in the subnet mask.
- The total number of 1s in the subnet mask is 26 (16 + 10).
- The total number of 0s is 6 (32 - 26). The mask is
  11111111 11111111 11111111 11000000 or
  **255.255.255.192**
- The number of subnets is 1024.
- The number of addresses in each subnet is $2^6$ (6 is the number of 0s) or 64.



## Example 6

What is the network address if one of the addresses is 167.199.170.82/27?

**_Solution_**

The prefix length is 27, which means that we must keep the first 27 bits as is and change the remaining bits (5) to 0s. The 5 bits affect only the last byte. The last byte is 01010010. Changing the last 5 bits to 0s, we get 01000000 or 64. The network address is 167.199.170.64/27.

## Example 7

An ISP is granted a block of addresses starting with 190.100.0.0/16. The ISP needs to distribute these addresses to three groups of customers as follows:

1. The first group has 64 customers; each needs 256 addresses.

2. The second group has 128 customers; each needs 128 addresses.

3. The third group has 128 customers; each needs 64 addresses.

Design the subblocks and give the slash notation for each subblock. Find out how many addresses are still available after these allocations.

**Group 1**

For this group, each customer needs 256 addresses. This means the suffix length is 8 ($2^8 = 256$). The prefix length is then $32 - 8 = 24$.

01: 190.100.0.0/24 ➔ 190.100.0.255/24

02: 190.100.1.0/24 ➔ 190.100.1.255/24

..........................................

64: 190.100.63.0/24 ➔ 190.100.63.255/24

Total = 64 × 256 = 16,384

**Group 2**

For this group, each customer needs 128 addresses. This means the suffix length is 7 ($2^7 = 128$). The prefix length is then $32 - 7 = 25$. The addresses are:

**001**: 190.100.64.0/25 ➔ 190.100.64.127/25

**002**: 190.100.64.128/25 ➔ 190.100.64.255/25

**128**: 90.100.127.128/25 ➔ 190.100.127.255/25

Total = 128 × 128 = 16,384

**Group 3**

For this group, each customer needs 64 addresses. This means the suffix length is 6 ($2^6 = 64$). The prefix length is then $32 - 6 = 26$.

**001**:190.100.128.0/26 ➔ 190.100.128.63/26

**002**:190.100.128.64/26 ➔ 190.100.128.127/26

..............................

**128**:190.100.159.192/26 ➔ 190.100.159.255/26

Total = 128 × 64 = 8,192

***Number of granted address : 65,534***

***Number of allocated address : 40,960***

***Number of available address : 24,574***

# Example 8

- An organization has purchased the Class C Address **195.5.20.0** and would to create networks of **50** hosts each.

**IP: 195.5.20.0**
**Subnet Mask: 255.255.255.0**

---

**Hosts number = 50**

***First*: Convert the number of Hosts to binary.**

50 = 00110010

**6 bits**

***Second*: Reserve bits in subnet mask and find your increment.**

**Subnet Mask: 255.255.255.0 = 11111111.11111111.11111111.00000000**
**11111111.11111111.11111111.11000000**

**2 bits**

**[Increment → 2^2 = 4 ] Networks**
**[2^6 = 64 – (2)=62] Hosts number**

**6 bits**

**Or**

| Bit Values in a Octet: | 128 | **64** | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Lowest incrementing value

# Cont..

- Third: Use increment to find the network ranges.

| No | Subnet | First valid address | Last valid address | Broadcast Address |
|---|---|---|---|---|
| 1 | 195.5.20.0 | 195.5.20.1 | 195.5.20.62 | 195.5.20.63 |
| 2 | 195.5.20.64 | 195.5.20.65 | 195.5.20.126 | 195.5.20.127 |
| 3 | 195.5.20.128 | 195.5.20.129 | 195.5.20.190 | 195.5.20.191 |
| 4 | 195.5.20.192 | 195.5.20.193 | 195.5.20.254 | 195.5.20.255 |

# Example 9

- Example: An organization has purchased the **Class A** Address 10.0.0.0 and would to create networks of 100 hosts each.

**IP: 10.0.0.0**
**Subnet Mask: 255.0.0.0**

---

**Hosts number = 100**

***First*: Convert the number of Hosts to binary.**

100 = 01100100

**7 bits**

***Second*: Reserve bits in subnet mask and find your increment.**

**Subnet Mask: 255.0.0.0 = 11111111.00000000.00000000.00000000**
**11111111.11111111.11111111.10000000**

**17 bits**

**[Increment → 2^17 = 131072 ] Networks**
**[2^7 = 128 – (2)=126] Hosts number**

**7 bits**

**Or**

| Bit Values in a Octet: | **128** | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|
| | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Lowest incrementing value

# Cont…

| No | Subnet | First valid address | Last valid address | Broadcast Address |
|---|---|---|---|---|
| 1 | 10.0.0.0 | 10.0.0.1 | 10.0.0.126 | 10.0.0.127 |
| 2 | 10.0.0.128 | 10.0.0.129 | 10.0.0.254 | 10.0.0.255 |
| 3 | 10.0.1.0 | 10.0.1.1 | 10.0.1.126 | 10.0.1.127 |
| 4 | 10.0.1.128 | 10.0.1.129 | 10.0.1.254 | 10.0.1.255 |

Thanks
4
ur
Attention
!