CHAPTER SEVEN

7. Introduction to ip addressing and sub-netting

7.1. General

In the networking and communications area, a protocol is the formal specification that defines the procedures that must be followed when transmitting or receiving data. Protocols define the format, timing, sequence, and error checking used on the network.

In plain English, the above means that if you have 2 or more devices e.g. computers which want to communicate, then they need a common "Protocol" which is a set of rules that guide the computers on how and when to talk to each other.

There are hundreds of protocols out there and it is impossible to list them all here, but instead we have included some of the most popular protocols around so you can read up on them and learn more about them by your own. Network protocols provide what are called "link services." These protocols handle addressing and routing information, error checking, and retransmission requests. Network protocols also define rules for communicating in a particular networking environment such as Ethernet or Token Ring.

- Protocols prepare data in a linear fashion.
- Protocol in one layer performs a certain set of operations on data.
- The data is then passed to the next layer where another protocol performs a different set of operations.

- At the destination, the protocols undo the construction of the packet that was done on the source side, in reverse order.

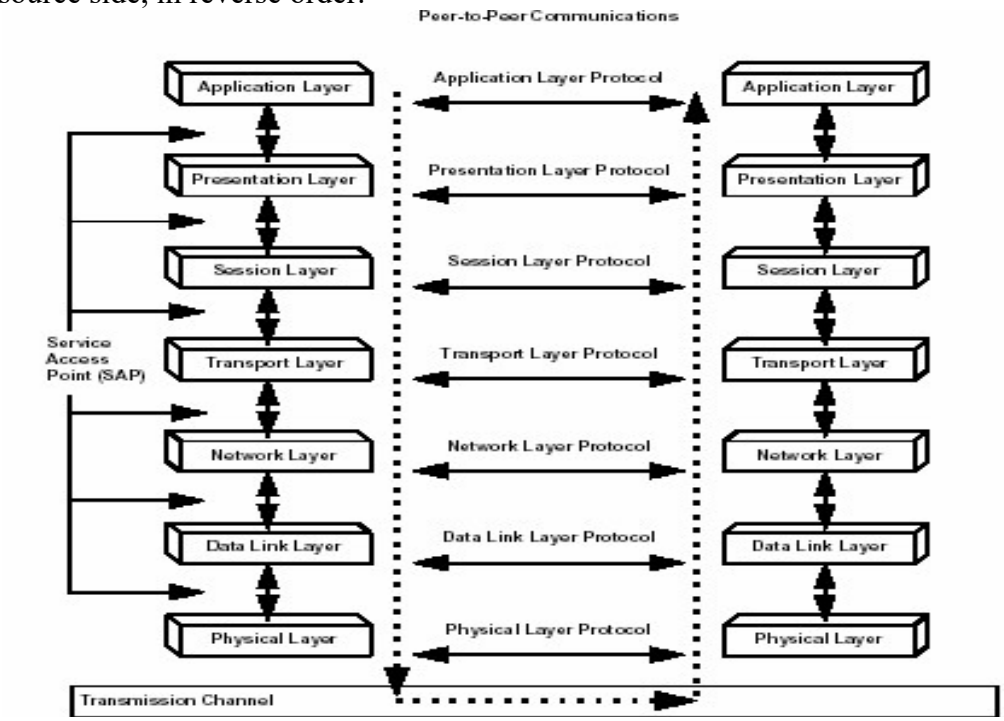


Figure 7.1 Peer-to-Peer Communications

Service access point (SAP) The interface between each of the seven layers in the OSI protocol stack that has connection points, similar to addresses, used for communication between layers.

Common/popular protocols

Table 7.1 Descriptions of Common protocols

OSI layer	Protocol	Description
Application Layer Protocol	**SMTP (Simple Mail Transfer Protocol)**	An Internet protocol for transferring e-mail.
	FTP (File Transfer Protocol)	An Internet files transfer protocol.
	SNMP (Simple Network Management Protocol)	An Internet protocol for monitoring networks and network components
	Telnet	An Internet protocol for logging on to remote hosts and processing data locally.
Transport Layer Protocol	**NetBEUI (NetBIOS extended user interface)**	Establishes communication sessions between computers (NetBIOS) and provides the underlying data transport

		services (NetBEUI). Protocols that are commonly used for Microsoft-based, peer-to-peer networks are NetBEUI and NetBIOS(**network basic input/output system)).** Note: NetBIOS is a Session layer protocol.
	TCP	The TCP/IP protocol for guaranteed delivery of sequenced data
Network layer protocol	Sequenced Packet Exchange (SPX)	Part of Novell's IPX/SPX protocol suite for sequenced data.
	IP	The TCP/IP protocol for packet-forwarding routing. Detailed explanation about IP is given below.
	Internetwork Packet Exchange (IPX)	NetWare's protocol for packet forwarding and routing

7.2. Introduction to the Internet Protocol

Perhaps one of the most important and well known protocols is the Internet Protocol (IP). IP gives us the ability to uniquely identify each computer/device in a network or on the Internet.

When a computer is connected to a network or the Internet, it is assigned a unique IP address. If you're connecting to the Internet, chances are you're given an IP automatically by your ISP (**I**nternet **S**ervice **P**rovider), if you're connecting to your LAN then you're either given the IP automatically or you manually configure the workstation with an assigned IP.

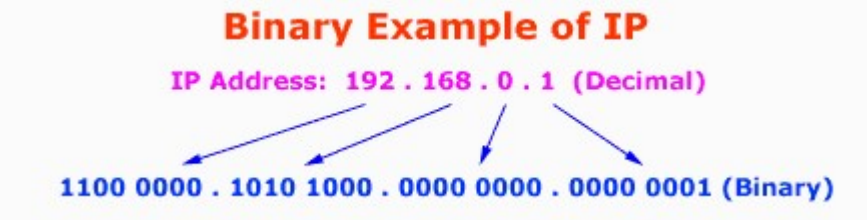
If you really want to understand how network communications work, especially when it comes to an IP network, like the Internet, a full understandation of IP is critical. DNS, FTP, SNMP, SMTP, HTTP and a lot of other protocols and services rely heavily on the IP protocol in order to function correctly, so you can immediately see that IP is more than just an IP Address on your workstation.

IP is one of the most important protocols in the networking world.

7.3. Network Math

The following figure shows:-

- **How an IP address looks like and**
- **How IP address is understood by a computer**



Understanding the conversion between Decimal and Binary

Algorithm 1

To convert a number form base 2 (Bin) to base 10 (Dec)

1) **Get the place value of each digit**
2) **Find the sum of the place values**

Algorithm 2

To convert a number from base 10 to base 2

1) **Divide the integer part successively by 2**
2) **Accumulate the remainders bottom up**

Exercise:

COBVERT

a) $(192)_{DEC}$ to $(\ )_{BIN}$
b) $(168)_{DEC}$ to $(\ )_{BIN}$
c) $(11000000)_{BIN}$ to $(\ )_{DEC}$
d) $(10101000)_{BIN}$ to $(\ )_{DEC}$

Notice: using n bits, we can represent 2^n decimal numbers

Example 1

Using 2 bits how many decimal numbers can be represented

Solution:

2^n, n=2 =>, $2^2 = 4$

Bin	Dec
00	0
01	1
10	2
11	3

Example 2

Using four bits

2^n, n=4, => 2^4= 16, decimal numbers can be represented

BINARY				DECIMAL
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3

0	**1**	**0**	**0**	4
0	**1**	**0**	**1**	5
0	**1**	**1**	**0**	6
0	**1**	**1**	**1**	7
1	**0**	**0**	**0**	8
1	**0**	**0**	**1**	9
1	**0**	**1**	**0**	10
1	**0**	**1**	**1**	11
1	**1**	**0**	**0**	12
1	**1**	**0**	**1**	13
1	**1**	**1**	**0**	14
1	**1**	**1**	**1**	15

Logical operation (operation on bits)

The And Logic

Let A and B are two digital inputs

THRTH TABLE

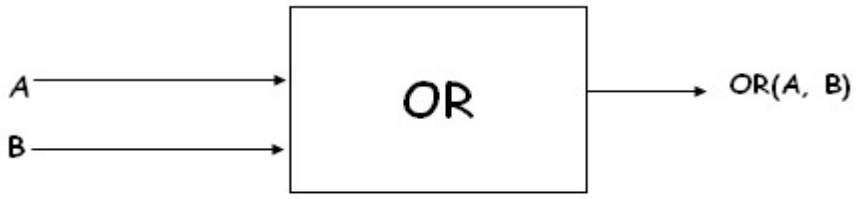
Possible number of unique input combinations = $\mathbf{2^n}$, **n=2,** $\mathbf{2^2 = 4}$

A	B	AND(A,B)
0	**0**	**0**

0	**1**	**0**
1	**0**	**0**
1	**1**	**1**

THE OR LOGIC

Let A and B are two digital inputs



THRTH TABLE

Possible number of unique input combinations = $\mathbf{2^n}$**, n=2,** $\mathbf{2^2 = 4}$

A	B	OR(A,B)
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

7.4. Internet Protocol Classes, Network & Host ID

There are certain values that an IP Address can take and these have been defined by the IEEE committee.

A simple IP Address is a lot more than just a number. It tells us the network that the workstation is part of and the node ID.

When the IEEE committee sat down to sort out the range of numbers that were going to be used by all computers, they came out with 5 different ranges or, as we call them, "Classes" of IP Addresses and when someone applies for IP Addresses they are given a certain range within a specific "Class" depending on the size of their network.

The 5 different Classes of IP address defined by the IEEE committee is given below

CLASS A	**0XXXXXXX**	**XXXXXXXX**	**XXXXXXXX**	**XXXXXXXX**
CLASS B	**10XXXXXX**	**XXXXXXXX**	**XXXXXXXX**	**XXXXXXXX**
CLASS C	**110XXXXX**	**XXXXXXXX**	**XXXXXXXX**	**XXXXXXXX**
CLASS D	**1110XXXX**	**XXXXXXXX**	**XXXXXXXX**	**XXXXXXXX**
CLASS E	**11110XXX**	**XXXXXXXX**	**XXXXXXXX**	**XXXXXXXX**

Note: "X" stands for don't care values (i.e. either "o" or " 1")

The first 3 classes (A, B and C) are used to identify workstations, routers, switches and other devices whereas the last 2 Classes (D and E) are reserved for special use.

An IP Address consists of 32 Bits, which means it's 4 bytes long. The first octet (first 8 Bits or first byte) of an IP Address is enough for us to determine the Class to which it belongs. And, depending on the Class to which the IP Address belongs, we can determine which portion of the IP Address is the **Network ID** and which is the **Node ID.**

For example, if the first octet of an IP Address is "168" then, using the above table, it falls within the 128-191 range, which makes it a Class B IP Address.

Understanding the Classes

Companies are assigned different IP ranges within these classes, depending on the size of their network. For instance, if a company required 1000 IP Addresses it would probably be assigned a range that falls within a Class B network rather than a Class A or C.

- The **Class A** IP Addresses were designed for large networks,
- **Class B** for medium size networks and
- **Class C** for smaller networks.

Introducing Network ID and Node ID concepts

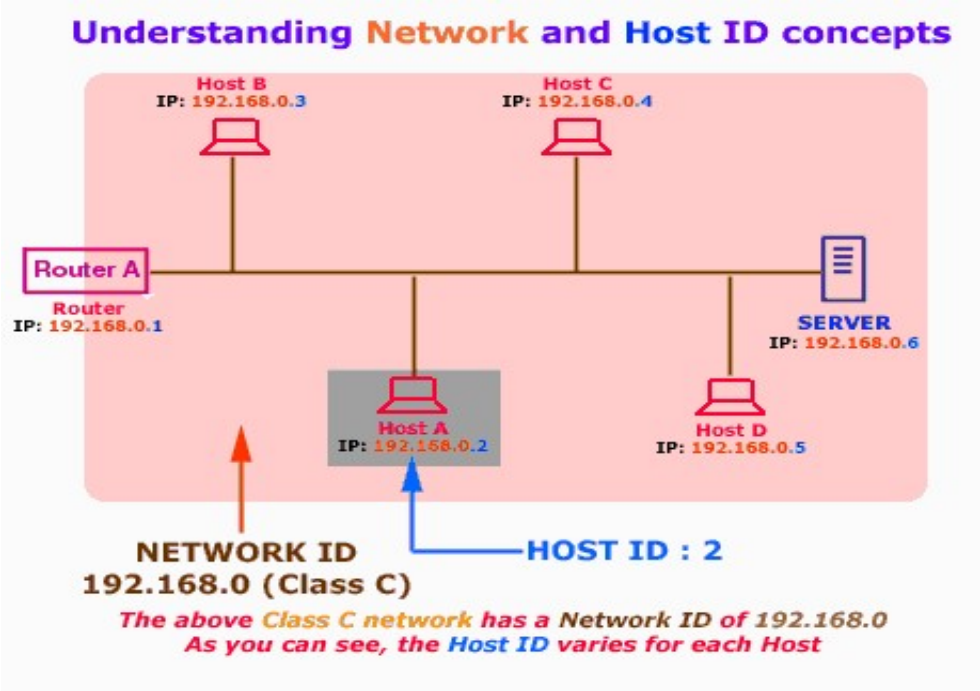
IP Address gives us 2 pieces of valuable information:

1) It tells us which network the device is part of (Network ID).

2) It identifies that unique device within the network (Node ID).

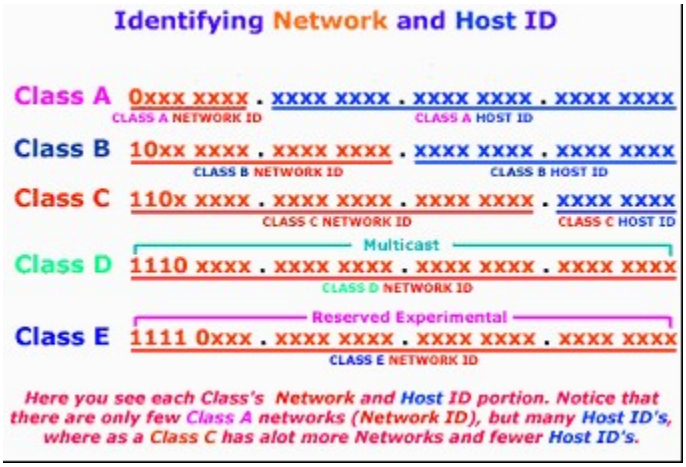
The Network ID tells us which network a particular computer belongs to and the Node ID identifies that computer from all the rest that reside in the same network.

The picture below gives you a small example to help you understand the concept:



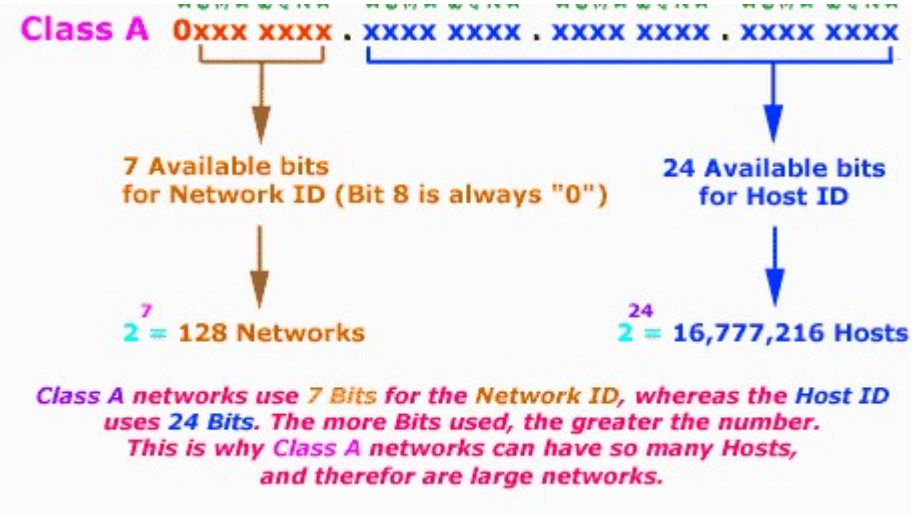
THE NETWORK AND NODE ID OF EACH CLASS

The table below shows you (in binary) how the Network ID and Node ID changes depending on the Class:

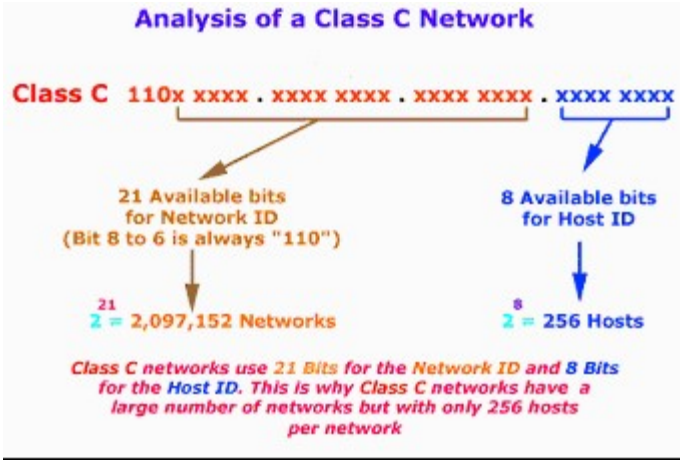


Analysis of each class

1. Class A network

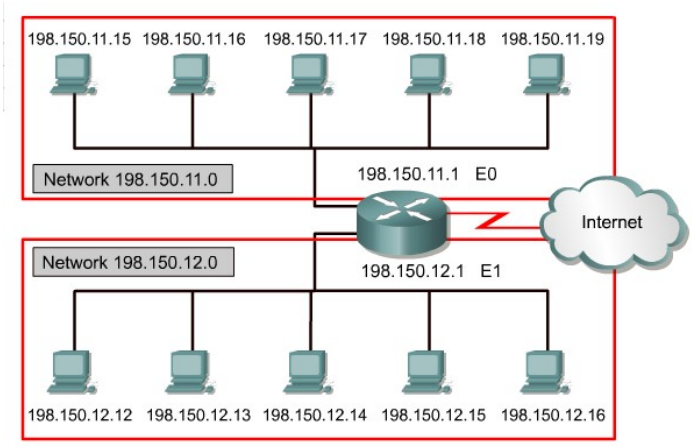


2. Class C network

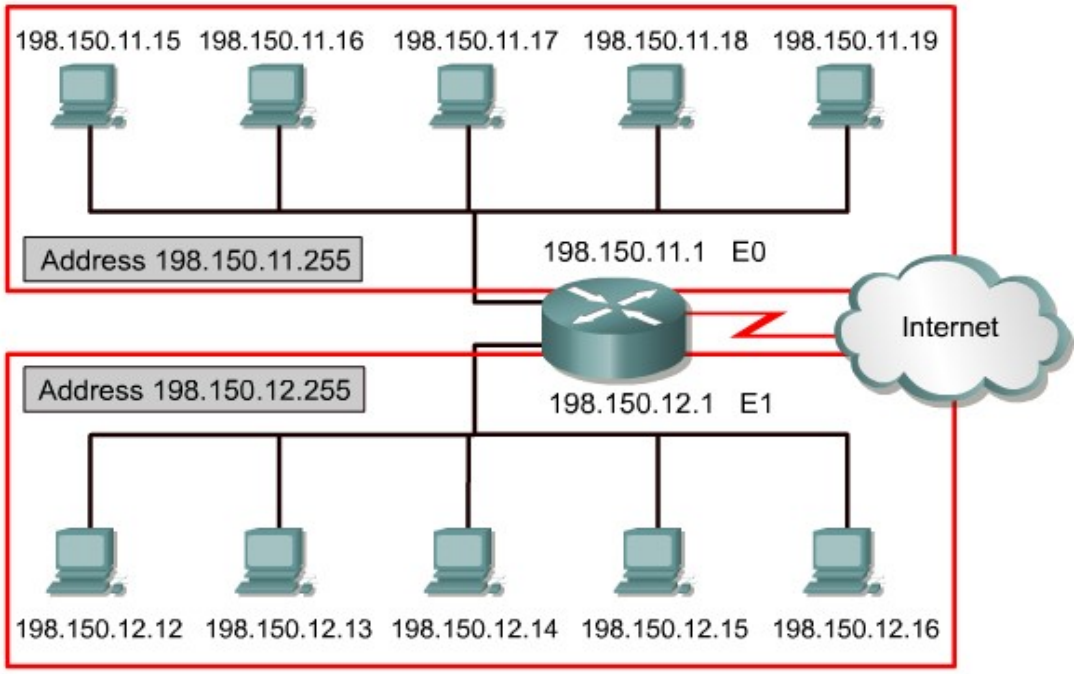


IMPORTANT NOTE: It is imperative that every network, regardless of Class and size, has a Network Address (first IP Address e.g 192.168.0.0 for Class C network) and a Broadcast Address (last IP Address e.g 192.168.0.255 for Class C network), which **cannot** be used. So when calculating available IP Addresses in a

Network address network, always remember to subtract 2 from the number of IP Addresses within that network.



Broadcast address



Exercise**: do the same thing for Class B network.**

7.5. subnetting

What is Subnetting?

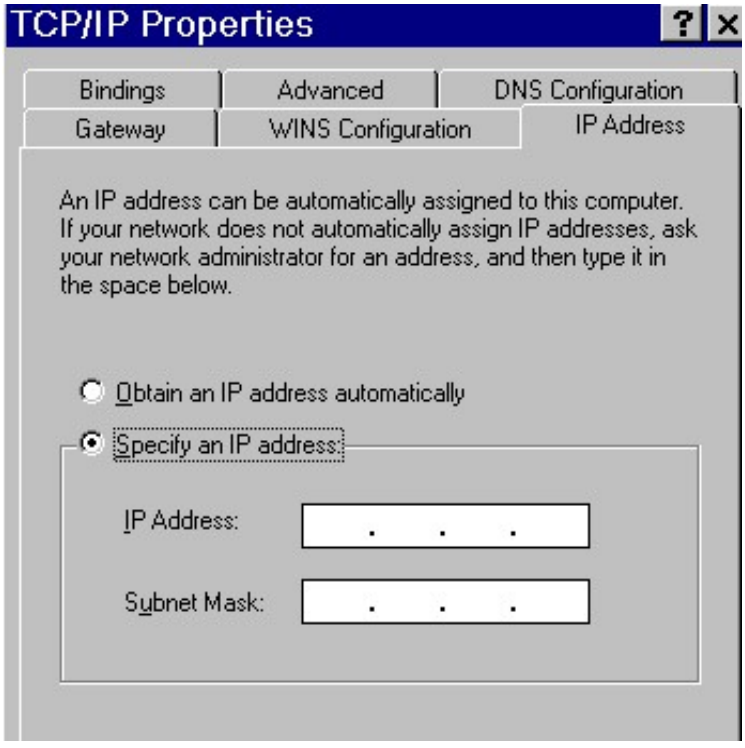
When we Subnet a network, we basically split it into smaller networks. For example, when a set of IP Addresses is given to a company, they might want to "partition" that one network into smaller ones, one for each department. This way, their Technical department and Management department can each have a small network of their own. By subnetting the network we can partition it to as many smaller networks as we need and this also helps reduce traffic and hides the complexity of the network.

By default, all type of Classes (A, B and C) have a subnet mask, we call it the "Default Subnet mask". You need to have one because:

1) All computers need the subnet mask field filled when configuring IP

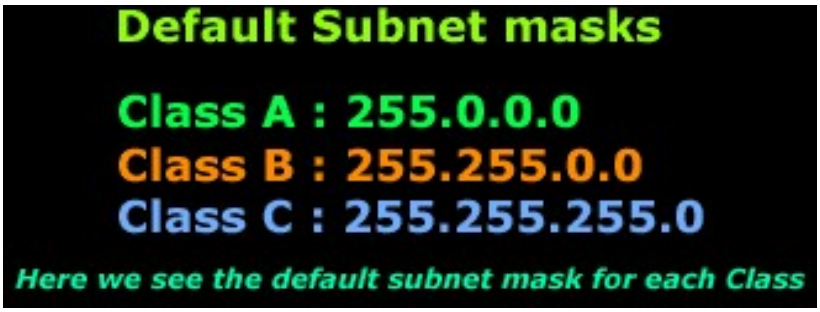
2) You need to set some logical boundaries in your network

2) You should at least enter the default subnet mask for the Class you're using



Subnet mask is what determines the Network ID and Host ID portion of an IP Address.

The table below shows clearly the subnet mask that applies for each network Class.



Slash Format

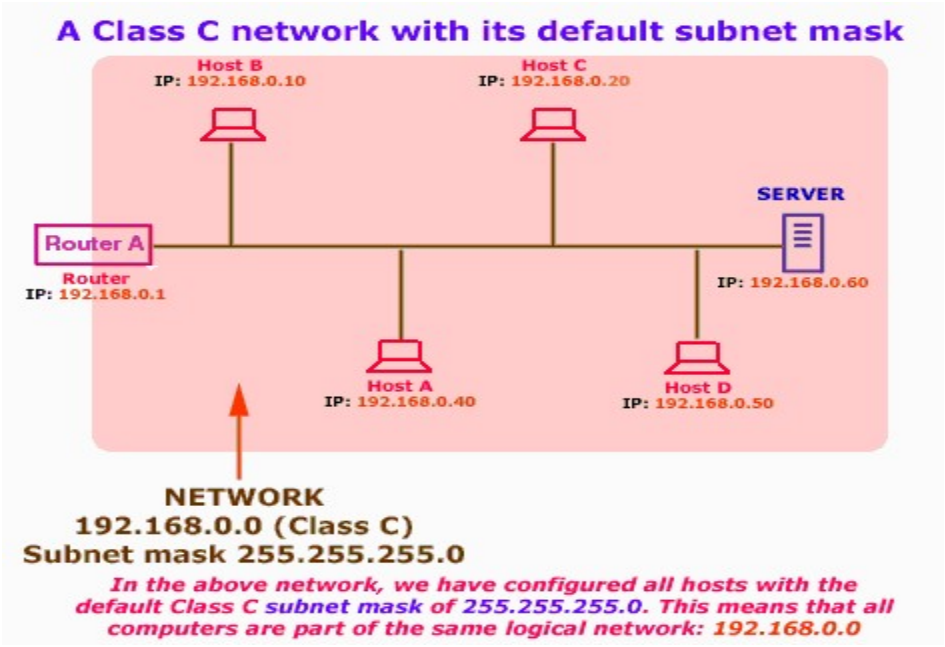
Class A **11111111.00000000.00000000.00000000 /8**

Class B **11111111.11111111.00000000.00000000 /16**

Class C **11111111.11111111.11111111.00000000 /24**

Note that the default subnet masks have been set by the IEEE committee.

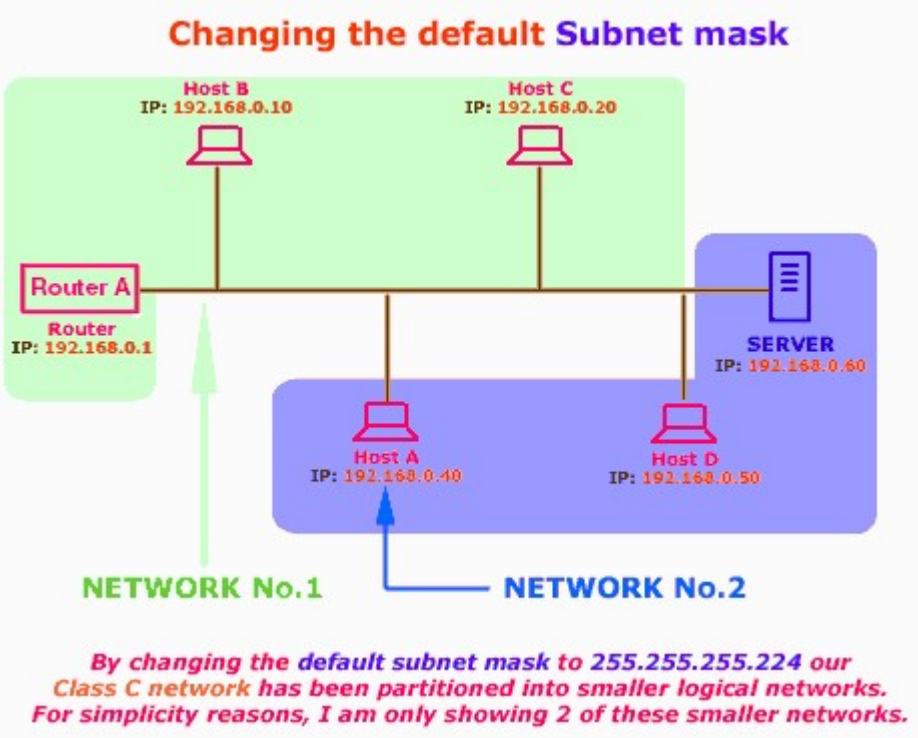
The picture below shows our example network (192.168.0.0). All computers here have been configured with the default Class C subnet mask (255.255.255.0):



Because of the subnet mask we used, all these computers are part of the one network. This also means that any one of these hosts (computers, router and server) can communicate with each other.

If we now wanted to partition this network into smaller segments, then we would need to change the subnet mask appropriately so we can get the desired result. Let's say we needed to change the subnet mask from 255.255.255.0 to 255.255.255.224 on each configured host.

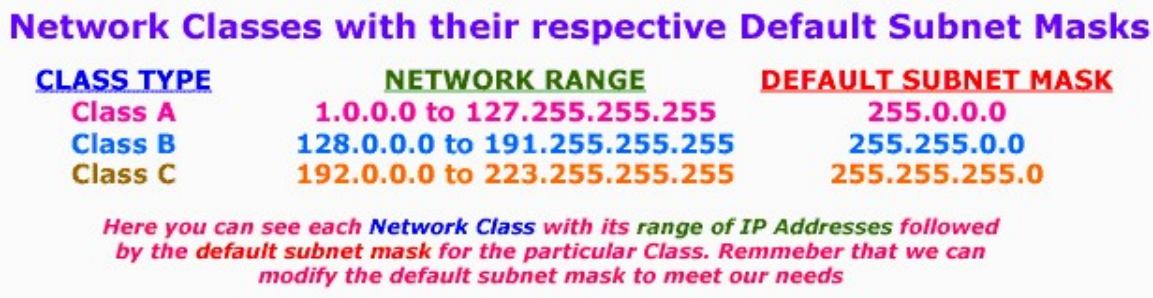
The picture below shows us how the computers will see the network once the subnet mask has changed:



In reality, we have just created 8 networks from the one large network we had, but here to keep things simple for now and showing only 2 of these smaller networks.

Default Subnet masks of each Class

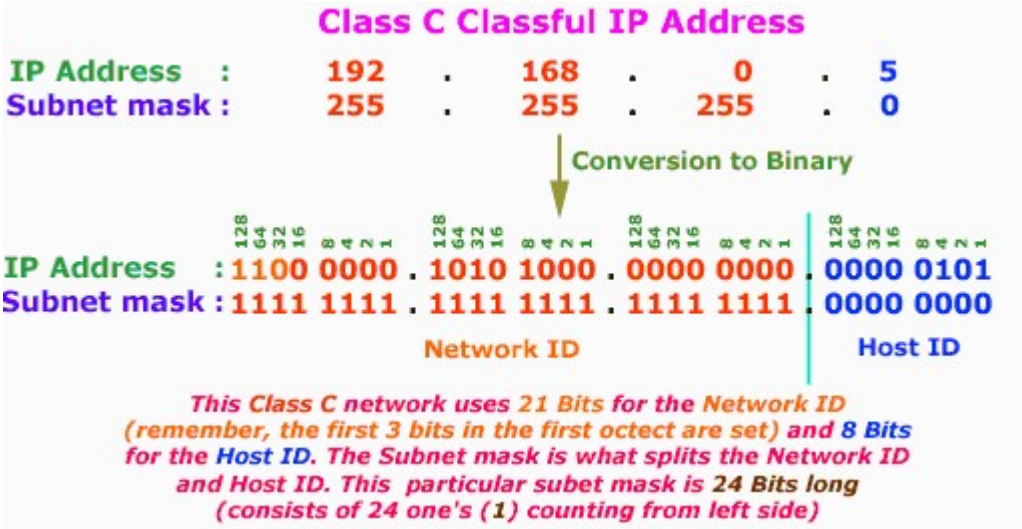
The picture below shows our 3 Network Classes with their respective default subnet mask:



The Effect of a Subnet Mask on an IP Address

We know that an IP Address consists of two parts, 1) The Network ID and 2) The Host ID. This rule applies for all IP Addresses that use the default subnet mask and we call them Classful IP Addresses.

Example: Class C classful IP address



NOTE:

All Class C Classful IP Addresses have a 24 bit subnet mask (255.255.255.0).

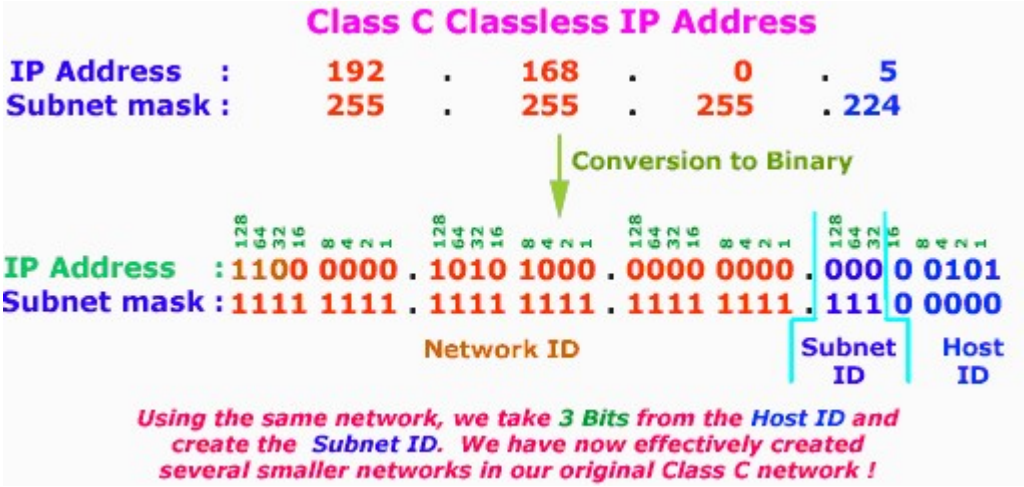
All Class B Classful IP Addresses have a 16 bit subnet mask (255.255.0.0).

All Class A Classful IP Addresses have an 8 bit subnet mask (255.0.0.0).

On the other hand, the use of an IP Address with a subnet mask other than the default results in the standard Host bits (the Bits used to indentify the HOST ID) being divided in to two parts: a Subnet ID and Host ID. These type of IP Addresses are called **Classless IP**

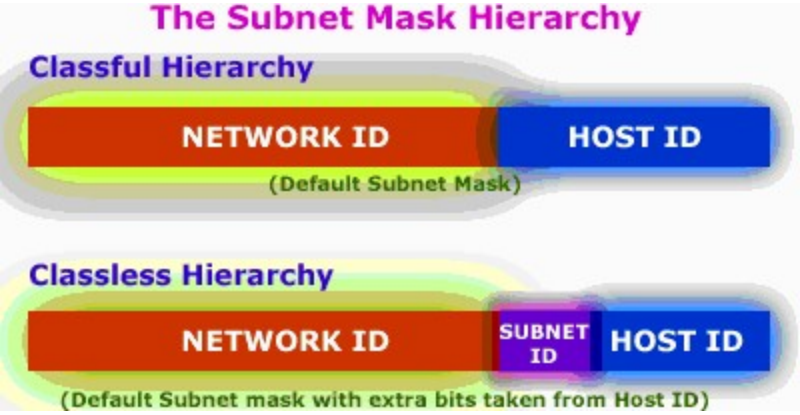
Addresses.

In order to understand what a "Classless IP Address" is without getting confused, we are going to take the same IP Address as above, and make it a Classless IP Address by changing the default subnet mask:



Looking at the picture above you will now notice that we have a Subnet ID, something that didn't exist before. As the picture explains, we have borrowed 3 bits from the Host ID and used them to create a Subnet ID. Effectively we partitioned our Class C network into smaller networks.

The picture below shows us both examples:



<u>Given an IP address, find its network ID , host ID and Subnet ID?</u>

Converting the IP address 10.34.23.134 to binary would result in:
00001010.00100010.00010111.10000110

Performing a Boolean AND of the IP address 10.34.23.134 and the subnet mask 255.0.0.0 produces the network address of this host:

00001010.00100010.00010111.10000110
11111111.00000000.00000000.00000000
00001010.00000000.00000000.00000000

Converting the result to dotted decimal, 10.0.0.0 is the network portion of the IP address, when using the 255.0.0.0 mask.

Any device, or gateway, connecting n networks/subnetworks has n distinct IP addresses, one for each network / subnetwork that it interconnects.

By extending the mask to be 255.255.255.224 (/27), you have taken three bits from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device since host ids of all zeros or all ones are not allowed (it is very important to remember this). So, with this in mind, the following subnets have been created.

- 204.15.5.0 /27 Host Address Range 1 To 30
- 204.15.5.32 /27 Host Address Range 33 To 62
- 204.15.5.64 /27 Host Address Range 65 To 94
- 204.15.5.96 /27 Host Address Range 96 To 126
- 204.15.5.128 /27 Host Address Range 129 To 158
- 204.15.5.160 /27 Host Address Range 161 To 190
- 204.15.5.192 /27 Host Address Range 193 To 222
- 204.15.5.224 /27 Host Address Range 225 To 254

Note: There are two ways to denote the above masks. First, since you are using three bits more than the "natural" Class C mask, you can denote these addresses as having a 3-bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. Using this method, one of the above networks can be described with the notation prefix/length. For example, 204.15.5.32/27 denotes the network 204.15.5.32 255.255.255.224.

Example 1

If you have network 172.16.0.0 ,then you know that its natural mask is 255.255.0.0 or 172.16.0.0/16. Extending the mask to anything beyond 255.255.0.0 means you are subnetting. You can quickly see that you have the ability to create a lot more subnets than with the Class C network above. If you use a mask of 255.255.248.0 (/21), how many subnets and hosts per subnet does this allow for?

172.16.0.0 - 10101100.00010000.00000000.00000000
255.255.248.0 - 11111111.11111111.11111000.00000000

You are using five bits from the original host bits for subnets. This will allow you to have 32 subnets (2^5). After using the five bits for subnetting, you are left with 11 bits for host addresses.

This will allow each subnet so have 2048 host addresses (2^{11}), 2046 of which could be assigned to devices.

Example 2

Determine to which subnet each address belongs.

Device A: 172.16.17.30/20
Device B: 172.16.28.15/20
Solution

Determining the Subnet for Device A:

172.16.17.30 - 10101100.00010000.00010001.00011110
255.255.240.0 - 11111111.11111111. 11110000.00000000
Subnet = 10101100.00010000.00010000.00000000 = 172.16.16.0, by performing AND logic

Device A belongs to subnet 172.16.16.0.

Determining the Subnet for Device B:

172.16.28.15 - 10101100.00010000.00011110. 00001111
255.255.240.0 - 11111111.11111111. 11110000.00000000
Subnet = 10101100.00010000.00010000.00000000 = 172.16.16.0

Device B belongs to subnet 172.16.16.0.

So from the above determinations, DeviceA and DeviceB have addresses that are part of the same subnet.

Example 3

Given the Class C network of 204.15.5.0/24, subnet the network in order to create the following network with the host requirements shown.

Net A: must support 14 hosts
Net B: must support 28 hosts
Net C: must support 2 hosts
Net D: must support 7 hosts
Net E: must support 28 host

You are required to create five subnets. The largest subnet must support 28 host addresses. Is this possible with a Class C network? and if so, then how?

You can start by looking at the subnet requirement. In order to create the five needed subnets you would need to use three bits from the Class C host bits.

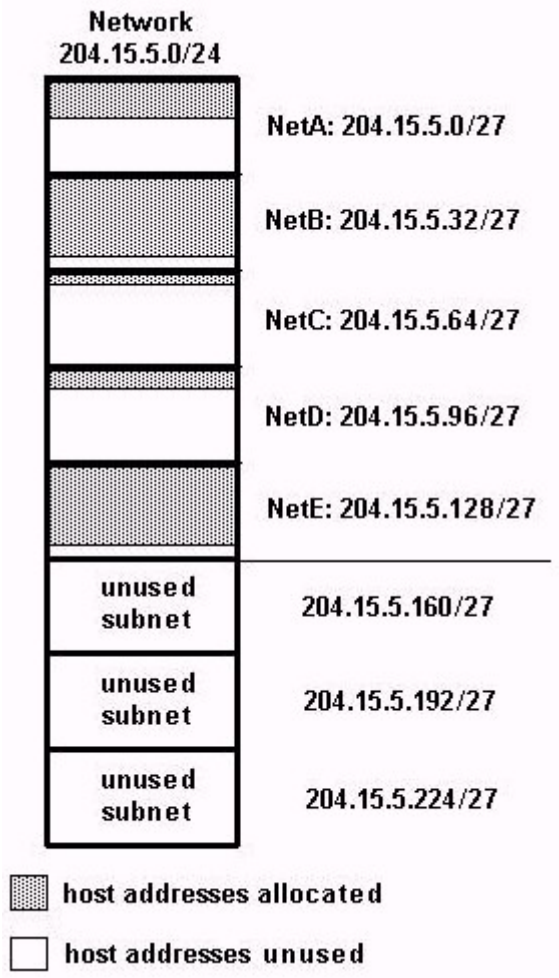
Since you need three subnet bits, that leaves you with five bits for the host portion of the address. How many hosts will this support? $2^5 = 32$ (30 usable). This meets the requirement.

Therefore you have determined that it is possible to create the above network with a Class C network. An example of how you might assign the subnetworks is as follows:

Net A: 204.15.5.0/27 Host Address Range 1 To 30
Net B: 204.15.5.32/27 Host Address Range 33 To 62
Net C: 204.15.5.64/27 Host Address Range 65 To 94
Net D: 204.15.5.96/27 Host Address Range 97 To 126
Net E: 204.15.5.128/27 Host Address Range 129 To 158

VARIABLE LENGTH SUBNET MASKS (VLSM)

In all of the previous examples of subnetting you will notice that the same subnet mask was applied for all the subnets. This means that each subnet has the same number of available host addresses. You may need this in some cases, but, in most cases, having the same subnet mask for all subnets ends up wasting address space. For example, in the second exercise above, a class C network was split into eight equal-size subnets; however, each subnet did not utilize all the available host addresses, resulting in wasted address space. This can be visualized as follows:



Looking at the above graphic, you can see that of the subnets that are being used, NetA, NetC, and NetD have a lot of unused host address space. This may have been a deliberate design accounting for future growth, but in many cases this is just wasted address space due to the fact that the same subnet mask is being used for all the subnets.

Variable Length Subnet Masks (VLSM) allows you to use different masks for each subnet, thereby using address space efficiently.

Example 3: VLSM

Given the following requirements, develop a **SUBNETTING** scheme using VLSM:

Net A: must support 14 hosts
Net B: must support 28 hosts
Net C: must support 2 hosts
Net D: must support 7 hosts
Net E: must support 28 host

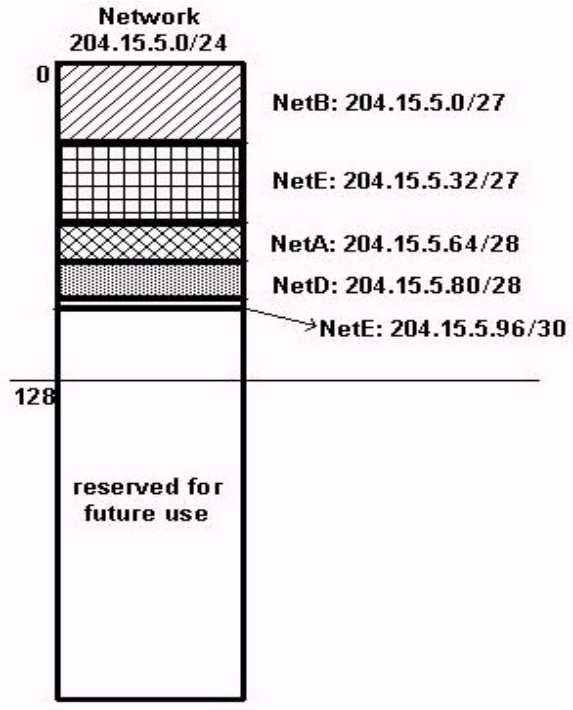
Determine what mask will allow the required number of hosts.

Net A: requires a /28 (255.255.255.240) mask to support 14 hosts
Net B: requires a /27 (255.255.255.224) mask to support 28 hosts
Net C: requires a /30 (255.255.255.250) mask to support 2 hosts
Net D: requires a /28 (255.255.255.240) mask to support 7 hosts
Net E: requires a /27 (255.255.255.224 mask to support 28 hosts

The Easiest Way To Assign The Subnets Is To Start Assigning The Largest First. Thus, You Can Assign In The Following Manner.

Net B: 204.15.5.0/27 Host Address Range 1 To 30
Net E: 204.15.5.32/27 Host Address Range 33 To 62
Net A: 204.15.5.64/28 Host Address Range 65 To 78
Net D: 204.15.5.80/28 Host Address Range 81 To 94
Net C: 204.15.5.96/30 Host Address Range 97 To 98

This can be graphically represented as follows.



From the above graphic you can see how using VLSM helped save more than half of the address space.