Experiment No : 06

Aim : To study & implement classification of IP addresses.

Theory : An IP (Internet Protocol) address is a unique identifier for a node or host connection on an IP network. An IP address is a 32 bit binary number usually represented as 4 decimal values, each representing 8 bits, in the range 0 & 255 (known as octets) separated by decimal points. This is known as “dotted decimal” notation.

Example: 140.179.220.200

10001100 . 10110011 . 11011100 . 11001000

Every IP address consists of 2 parts, one identifying the network & one identifying the node. The class of the address & subnet mask determines which part belongs to the network address & which part belongs to the node address.

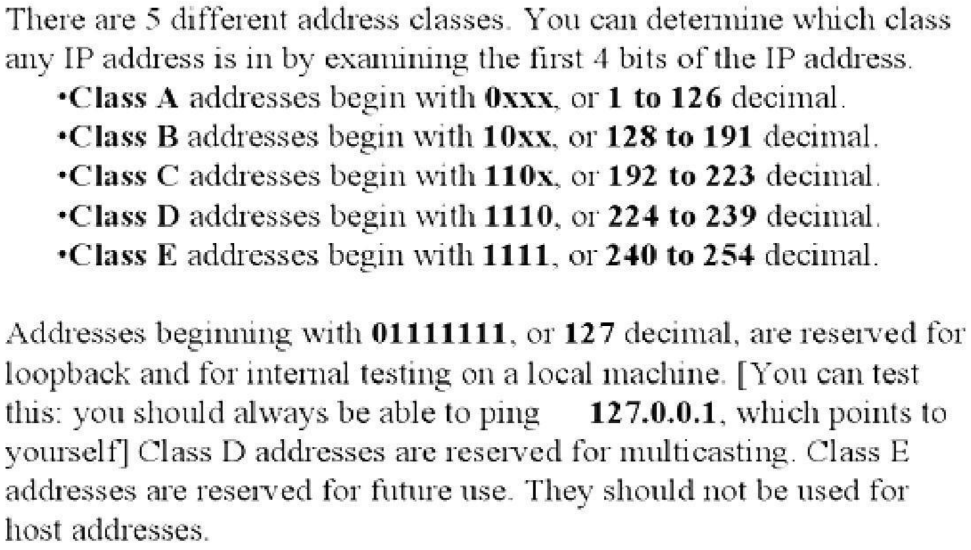
The network address uniquely identifies each network. Every machine on the same network shares that network address as a part of its IP address.

In the IP address 130.57.30.56, for example, the 130.57. is the network address.

The node address is assigned to, & uniquely identifies, each machine on a network. The part of the address must be unique because it identifies a particular machine-an individual, as opposed to a network, which is a group. This number can also be referred to as a host address. In the sample IP address 130.57.30.56, & .30.56 is the node address.

The designers of the Internet decided to create classes of networks based on network size. For a small number of networks possessing a large number of nodes, they created the rank class A network. At the other extreme is the class C network, reserved for the numerous networks with a small number of nodes. The class distinction for networks is between very large & very small is predictably called a class B network.

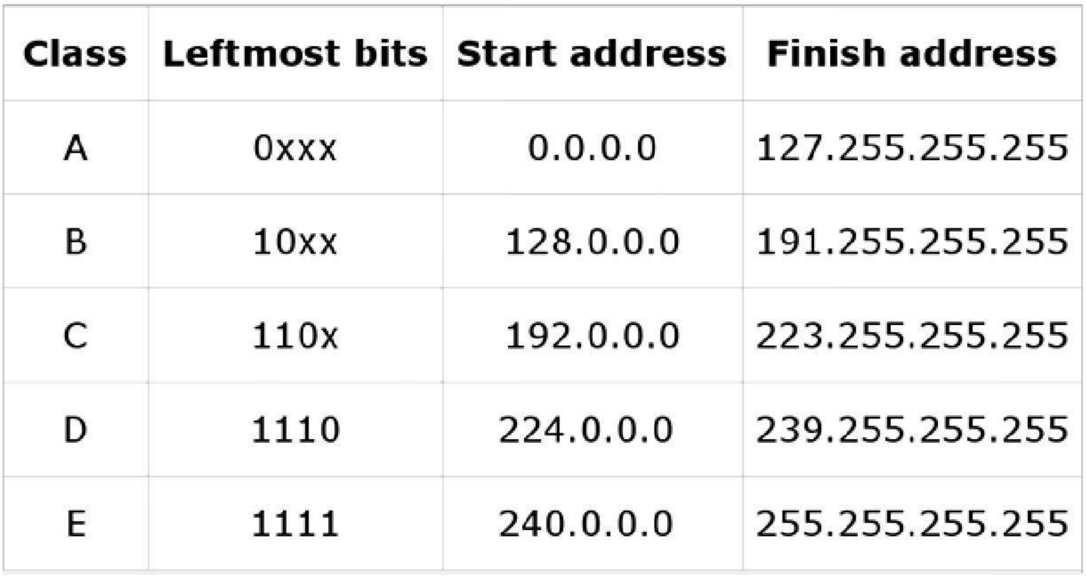
How one could subdivide an IP address into a network & node address is determined by the class designation of one’s network



There are 5 different address classes. You can determine which class any IP address is in by examining the first 4 bits of the IP address.

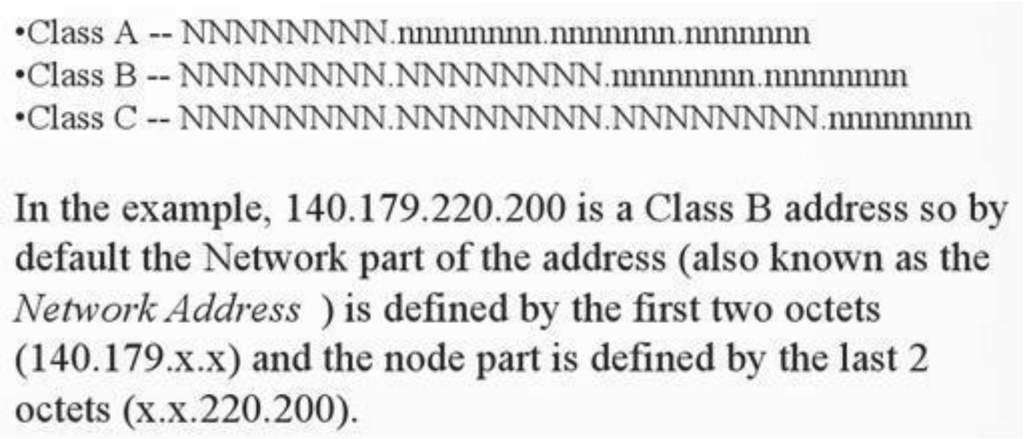
- **Class A** addresses begin with **0xxx**, or **1 to 126** decimal.
- **Class B** addresses begin with **10xx**, or **128 to 191** decimal.
- **Class C** addresses begin with **110x**, or **192 to 223** decimal.
- **Class D** addresses begin with **1110**, or **224 to 239** decimal.
- **Class E** addresses begin with **1111**, or **240 to 254** decimal.

Addresses beginning with **01111111**, or **127** decimal, are reserved for loopback and for internal testing on a local machine. [You can test this: you should always be able to ping **127.0.0.1**, which points to yourself] Class D addresses are reserved for multicasting. Class E addresses are reserved for future use. They should not be used for host addresses.



Class	Leftmost bits	Start address	Finish address
A	0xxx	0.0.0.0	127.255.255.255
B	10xx	128.0.0.0	191.255.255.255
C	110x	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

Now we can see how the Class determines, by default, which part of the IP address belongs to the network (N) & which part belongs to the node (n)

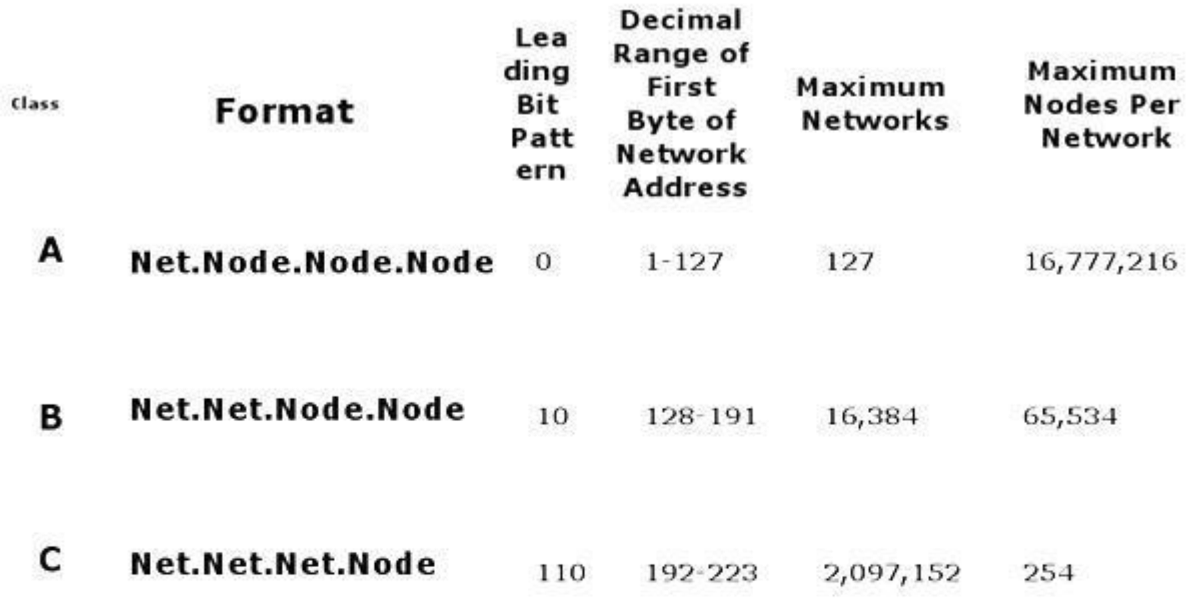


•Class A -- NNNNNNNN.nnnnnnnn.nnnnnnn.nnnnnnn
•Class B -- NNNNNNNN.NNNNNNNN.nnnnnnnn.nnnnnnnn
•Class C -- NNNNNNNN.NNNNNNNN.NNNNNNNN.nnnnnnnn

In the example, 140.179.220.200 is a Class B address so by default the Network part of the address (also known as the *Network Address*) is defined by the first two octets (140.179.x.x) and the node part is defined by the last 2 octets (x.x.220.200).

In our example, 140.179.0.0 specifies the network address for 140.179.220.200. when the node section is set to all “1”s, it specifies the broadcast that is sent to all hosts on the network.

140.179.255.255 specifies the example broadcast address. Note that this is true regardless of the length of the node section.



Class	Format	Leading Bit Pattern	Decimal Range of First Byte of Network Address	Maximum Networks	Maximum Nodes Per Network
A	Net.Node.Node.Node	0	1-127	127	16,777,216
B	Net.Net.Node.Node	10	128-191	16,384	65,534
C	Net.Net.Net.Node	110	192-223	2,097,152	254

Subnetting :

Subnetting an IP network can be done for a variety of reasons, including organization, use of different physical media (such as Ethernet, FDDI, WAN etc.), preservation of address space, & security. The most common reason is to control network traffic. In an Ethernet network, all nodes on a segment see all the packets transmitted by all the other nodes on the segment. Performance can be adversely affected under heavy traffic loads, due to collisions & resulting retransmissions. A router is used to connect IP networks to minimize the amount of traffic each segment must receive.

Subnet Masking

Applying a subnet mask to an IP address allows you to identify the network and node parts of the address. The network bits are represented by the 1s in the mask, and the node bits are represented by the 0s. Performing a bitwise logical AND operation between the IP address and the subnet mask results in the Network Address or Number.

For example, using our test IP address and the default Class B subnet mask, we get:

```
Class B IP Address
10001100.10110011.11110000.11001000     140.179.240.200
11111111.11111111.00000000.00000000     255.255.000.000
---------------------------------------------------    ----------------------
10001100.10110011.00000000.00000000     140.179.000.000
```

Network Address = 140.179.000.000

Default subnet masks:

- **Class A** –
  255.0.0.0
  11111111.00000000.00000000.00000000

- **Class B** –
  255.255.0.0
  11111111.11111111.00000000.00000000

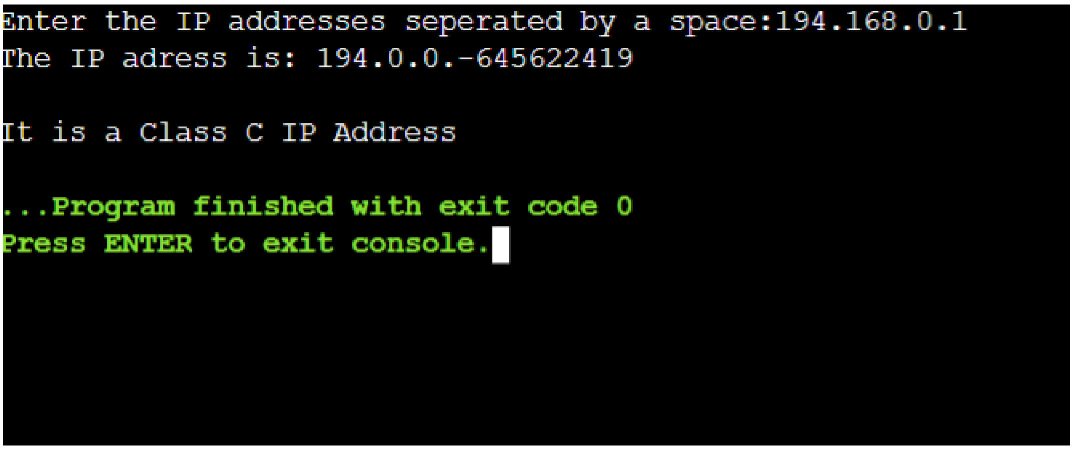
- **Class C** –
  255.255.255.0
  11111111.11111111.11111111.00000000

Code :

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
int a,b,c,d,i;
int x[8]={0};

cout<<"Enter the IP addresses seperated by a space:";
```

```
cin>>a>>b>>c>>d;

cout<<"The IP adress is: "<<a<<"."<<b<<"."<<c<<"."<<d<<endl<<endl;

for(i=0;i<8;i++)

{

x[7-i]=a%2;

a=a/2;

}

if(x[0]==0)

{

cout<<"It is a Class A IP Address";

}

else if(x[0]==1&&x[1]==0)

{

cout<<"It is a Class B IP Address";

}

else if(x[0]==1&&x[1]==1&&x[2]==0)

{

cout<<"It is a Class C IP Address";

}


else if(x[0]==1&&x[1]==1&&x[2]==1&&x[3]==0)

{

cout<<"It is a Class D IP Address";
}


else if(x[0]==1&&x[1]==1&&x[2]==1&&x[3]==1&&x[4]==1)
{
cout<<"It is a Class E IP Address";
}
getch();
}
```

Output :



```
Enter the IP addresses seperated by a space:194.168.0.1
The IP adress is: 194.0.0.-645622419

It is a Class C IP Address

...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION: - Hence we have successfully Study & implemented the classification of IP addresses.