Assignment No. 5

Problem Definition:

Write a program to demonstrate subnetting and find the subnet masks. (Use JAVA/PYTHON)

1. Prerequisite:

1. Network Layer: Roles, Protocols
2. Java Programming Syntax

2. Learning Objectives:

- Students will able to understand IP Addressing and Subnetting

3. Theory

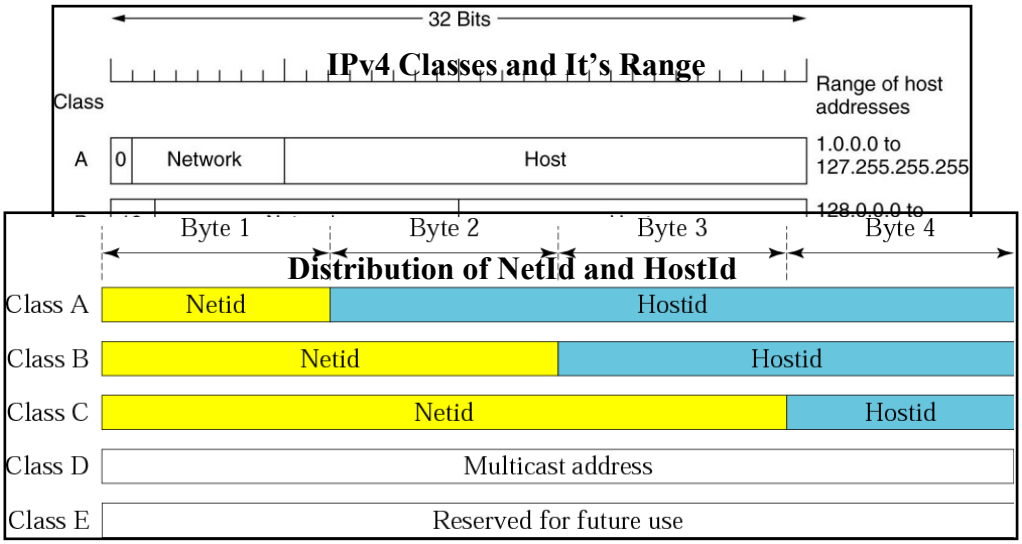
Introduction to IPv4

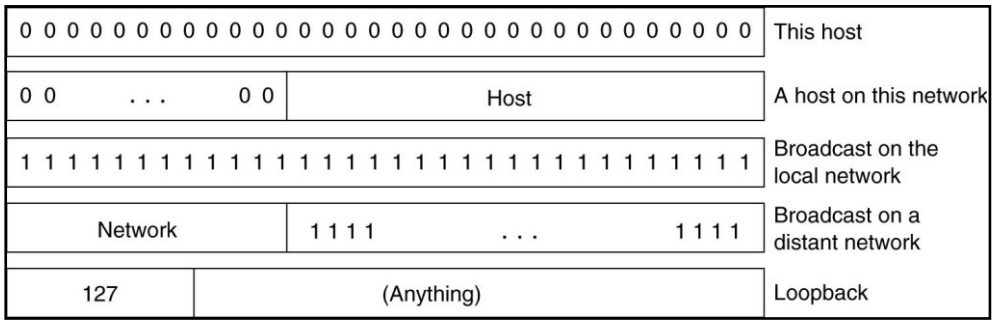
The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address. An IP address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. IP addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address. The address space of IPv4 is 232 or 4,294,967,296.

Network classes

Internet addresses are allocated by the Inter NIC, the organization that administers the Internet. These IP addresses are divided into classes. The most common of these are classes A, B, and C. Classes D and E exist, but are not generally used by end users. Each of the address classes has a different default subnet mask. You can identify the class of an IP address by looking at its first octet. Following are the ranges of Class A, B, and C Internet addresses, each with an example address:

- Class A networks use a default subnet mask of 255.0.0.0 and have 0-127 as their first octet. The address 10.52.36.11 is a class A address. Its first octet is 10, which is between 1 and 126, inclusive.
- Class B networks use a default subnet mask of 255.255.0.0 and have 128-191 as their first octet. The address 172.16.52.63 is a class B address. Its first octet is 172, which is between 128 and 191, inclusive.
- Class C networks use a default subnet mask of 255.255.255.0 and have 192-223 as their first octet. The address 192.168.123.132 is a class C address. Its first octet is 192, which is between 192 and 223, inclusive.





Special IP Address

Addresses for Private Networks

Class	*Netids*	*Blocks*
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

The network address is the beginning address of each block. It can be found by applying the default mask to any of the addresses in the block (including itself). It retains the netid of the block and sets the hostid to zero.

Table 1: Default masks

Class	*Mask in binary*	*Mask in dotted-decimal*
A	**11111111** 00000000 00000000 00000000	**255.**0.0.0
B	**11111111 11111111** 00000000 00000000	**255.255.**0.0
C	**11111111 11111111 11111111** 00000000	**255.255.255.**0

Need of Subnetting

Specifically, the network addresses available for assignment to organizations are close to depletion. This is coupled with the ever-increasing demand for addresses from organizations that want connection to the Internet.

There are 4 of the major reasons for subnetting or segmenting network?

1. To divide a large network into smaller segments to reduce traffic and speed up the sections of your network.
2. To connect networks across geographical areas.
3. To connect different topologies such as Ethernet, Token Ring, and FDDI together via routers.
4. To avoid physical limitations such as maximum cable lengths or exceeding the

maximum number of computers on a segment.

In this section we briefly discuss solution: Subnetting. A Class A, B, or C TCP/IP network can be further divided, or subnetted, by a system administrator.

Example

A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into 20 separate subnets.

Step 1) **Determine the number of subnets and convert to binary**

- In this example, the binary representation of 20 = 00010100.

Step 2) **Reserve required bits in subnet mask and find incremental value**

- The binary value of 20 subnets tells us that we need at least 5 network bits to satisfy this requirement (since you cannot get the number 20 with any less than 5 bits – 10100)

- Our original subnet mask is 255.255.255.0 (Class C subnet)

- The full binary representation of the subnet mask is as follows: 255.255.255.0 = 11111111.11111111.11111111.00000000

- We must "convert" 5 of the client bits (0) to network bits (1) in order to satisfy the requirements: New Mask = 11111111.11111111.11111111.11111000

- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks – 255.255.255.248 - Our increment bit is the last possible network bit, converted back to a binary number:
New Mask = 11111111.11111111.11111111.1111(1)000 – bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number 8

Step 3) **Use increment to find network ranges**

- Start with your given network address and add your increment to the subnetted octet:209.50.1.0 209.50.1.8 209.50.1.16 …etc

- You can now fill in your end ranges, which is the last possible IP address before you startthe next range 209.50.1.0 – 209.50.1.7 209.50.1.8 – 209.50.1.15 209.50.1.16 – 209.50.1.23
…etc

- You can then assign these ranges to your networks. Remember the first and last addressfrom each range (network / broadcast IP) is unusable.

Conclusion:

Hence we have studied IP Addressing and Subnetting.

Program-

```
import java.io.*;
public class Subnet {
    public static void main(String[] args) throws IOException {
        System.out.println("ENTER IP:");
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String ip = br.readLine();
        String checkclass = ip.substring(0, 3);
        int cc = Integer.parseInt(checkclass);
        String mask = null;
        if (cc > 0) {
            if (cc <= 127) {
                mask = "255.0.0.0";
                System.out.println("Class A IP Address");
                System.out.println("SUBNET MASK:\n" + mask);
            }
            if (cc >= 128 && cc <= 191) {
                mask = "255.255.0.0";
                System.out.println("Class B IP Address");
                System.out.println("SUBNET MASK:\n" + mask);
            }
            if (cc >= 192 && cc <= 223) {
                mask = "255.255.255.0";
                System.out.println("Class C IP Address");
                System.out.println("SUBNET MASK:\n" + mask);
            }
            if (cc >= 224 && cc <= 239) {
                mask = "255.0.0.0";
                System.out.println("Class D IP Address Used for multicasting");
            }
            if (cc >= 240 && cc <= 254) {
                mask = "255.0.0.0";
```

```
            System.out.println("Class E IP Address Experimental Use");
        }
    }
    String networkAddr = "";
    String lastAddr = "";
    String[] ipAddrParts = ip.split("\\.");
    String[] maskParts = mask.split("\\.");
    for (int i = 0; i < 4; i++) {
        int x = Integer.parseInt(ipAddrParts[i]);
        int y = Integer.parseInt(maskParts[i]);
        int z = x & y;
        networkAddr += z + ".";
        int w = z | (y ^ 255);
        lastAddr += w + ".";
    }
    System.out.println("First IP of block: " + networkAddr);
    System.out.println("Last IP of block: " + lastAddr);
  }
}
```

Output-