Batch: A1 Roll No.: 1711008

Experiment / assignment / tutorial No.__1_____

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Experiment No.:1

TITLE: IP classes and Implementation of Subnet mask concept.

AIM: To study IP classes and Implementation of Subnet mask concept.

An IP (Internet Protocol) address is a unique identifier for a node or host connection on an IP network. 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, and 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 that segment. Performance can be adversely affected under heavy traffic loads, due to collisions and the resulting retransmissions. A router is used to connect IP networks to minimize the amount of traffic each segment must receive.

This experiment enables student for identifying the class of the IP address and design particular subnets as per user requirements.

Expected Outcome of Experiment:

CO: Elaborate various network layer services and protocols in wired and wireless technology

Books/ Journals/ Websites referred:

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

Pre Lab/ Prior Concepts: IP Address, Classes, Subnet concept

New Concepts to be learned: Subnet mask calculation, Subnet address calculation

---

Stepwise-Procedure:

Applying a subnet mask to an IP address allows 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.

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

Additional bits can be added to the default subnet mask for a given Class to further subnet, or break down, a network. When a bitwise logical AND operation is performed between the subnet mask and IP address, the result defines the *Subnet Address* (also called the *Network Address* or *Network Number*). There are some restrictions on the subnet address. Node addresses of all "0"s and all "1"s are reserved for specifying the local network (when a host does not know its network address) and all hosts on the network (broadcast address), respectively. This also applies to subnets. A subnet address cannot be all "0"s or all "1"s. This also implies that a 1 bit subnet mask is not allowed. This restriction is required because older standards enforced this restriction. Recent standards that allow use of these subnets have superseded these standards, but many "legacy" devices do not support the newer standards. If you are operating in a controlled environment, such as a lab, you can safely use these restricted subnets.

CIDR -- **C**lassless **I**nter **D**omain **R**outing:
The "classful" system of allocating IP addresses can be very wasteful; Under supernetting, the classful subnet masks are extended so that a network address and subnet mask could, for example, specify multiple Class C subnets with one address.

For example, If about 1000 addresses are required, it could be possible to supernet 4 Class C networks together:

192.60.128.0 (11000000.00111100.10000000.00000000)Class C subnet address
192.60.129.0(11000000.00111100.10000001.00000000) Class C subnet address
192.60.130.0(11000000.00111100.10000010.00000000) Class C subnet address
192.60.131.0(11000000.00111100.10000011.00000000) Class C subnet address
---
192.60.128.0 (11000000.00111100.10000000.00000000) Supernetted subnet address
255.255.252.0 (11111111.11111111.11111100.00000000)Subnet Mask
192.60.131.255 (11000000.00111100.10000011.11111111) Broadcast address

In this example, the subnet 192.60.128.0 includes all the addresses from 192.60.128.0 to 192.60.131.255. In the binary representation of the subnet mask, the Network portion of the address is 22 bits long, and the host portion is 10 bits long.

Under CIDR, the subnet mask notation is reduced to simplified shorthand. Instead of spelling out the bits of the subnet mask, it is simply listed as the number of 1s bits that start the mask. In the above example, instead of writing the address and subnet mask as 192.60.128.0, Subnet Mask 255.255.252.0 .the network address would be written simply as: 192.60.128.0/22 Which indicates starting address of the network, and number of 1s bits (22) in the network portion of the address. Subnet mask in binary

11111111.11111111.11111100.00000000.

The use of a CIDR notated address is the same as for a Classful address. Classful addresses can easily be written in CIDR notation as Class A = /8, Class B = /16, and Class C = /24
To calculate the number of subnets or nodes,

No. of Nodes/ Subnets $=2^n-2$

Where n = number of bits in either field.

Multiplying the number of subnets by the number of nodes available per subnet gives you the total number of nodes available for your class and subnet mask. Also, note that although subnet masks with non-contiguous mask bits are allowed, they are not recommended.

Example:

10001100.10110011.11011100.11001000	140.179.220.200IP Address
11111111.11111111.**111**00000.00000000	255.255.**224**.000Subnet Mask
10001100.10110011.11000000.00000000	140.179.192.000Subnet Address
10001100.10110011.11011111.11111111	40.179.223.255 Broadcast Address

1. Program starts with taking IP address from user and the number of subnets from the user.
2. Then the calculation for subnet mask is done as specified in methodology.
3. Then with AND ing with subnet mask the subnet addresses are calculated.

IMPLEMENTATION: (printout of code)

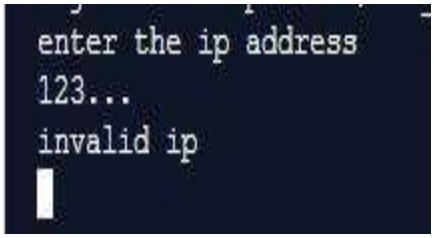
1)classless addressing

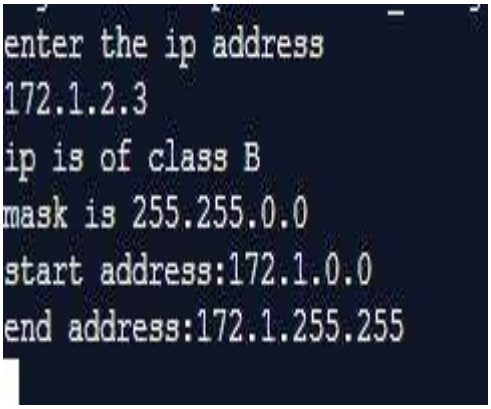
```
import java.util.*;
import java.io.*;


class Main {
  public static void main(String[] args)
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("enter the ip address");
    String input=sc.nextLine();
```

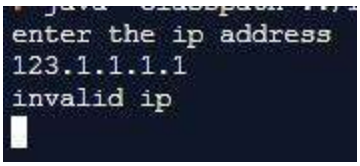
```
  String s1[]=input.split("\\.");
  try{
  if((s1.length>4||s1.length<4))
  System.out.println("invalid ip");
  else
  {
    for(int i=0;i<s1.length;i++)
    {
      if(Integer.parseInt(s1[i])>255)
      {System.out.println("invalid ip");
      System.exit(0);}
      else
      continue;
    }

  }
  }catch(Exception e)
 {
   System.out.println("invalid ip");
   System.exit(0);
 }
  char c=get_class(s1);
  get_first_last_addresss(s1,c);

  }
public static char get_class(String s[])
{
  try{
  int n=Integer.parseInt(s[0]);
  char ca;
  if(n>0&&n<=127)
  {
    System.out.println("ip is of class A");
    ca='A';
     return ca;
    }
  else if(n>127&&n<=191)
  {
    System.out.println("ip is of class B");
    ca='B';
     return ca;
    }
  else if(n>191&&n<=223)
  {
    System.out.println("ip is of class C");
    ca='C';
     return ca;
    }
    else if(n>223&&n<=239)
  {
    System.out.println("ip is of class D");
    ca='D';
     return ca;
```

```
      }
      else if(n>239&&n<=255)
     {
       System.out.println("ip is of class E");
       ca='E';
        return ca;
       }
        }catch(Exception e)
    {
      System.out.println("invalid ip");
      System.exit(0);
    }


     return 'S';

   }
   public static void get_first_last_addresss(String s[],char ca)
   {
     if(ca=='A')
     {
       System.out.println("mask is 255.0.0.0");
       System.out.println("start address:"+s[0]+".0.0.0");
       System.out.println("end address:"+s[0]+".255.255.255");

     }
     else if(ca=='B')
     {
       System.out.println("mask is 255.255.0.0");
       System.out.println("start address:"+s[0]+"."+s[1]+".0.0");
       System.out.println("end address:"+s[0]+"."+s[1]+".255.255");

     }
     else if(ca=='C'||ca=='D')
     {
       System.out.println("mask is 255.255.255.0");
       System.out.println("start address:"+s[0]+"."+s[1]+"."+s[2]+".0");
       System.out.println("end address:"+s[0]+"."+s[1]+"."+s[2]+"."+"255");

     }
     else
     {
       System.out.println("not possible");
     }

   }
}
```

Output







2)no of ip address as input

```
import java.util.*;

import javax.lang.model.util.ElementScanner6;
class Main {
  public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    System.out.println("enter ip address and no of ip address required");
    int i=0;
    //String in=sc.next();
    int ip=sc.nextInt();
    while(Math.pow(2,i)<ip)
    {
      i++;
    }
   int mask=32-i;

   String s="";
   int j1=0;
   for(int j=1;j<=32;j++)
   {
     if(j%8==0 && j!=0 &&j<=mask )
     {
       s=s+"1.";
       //j--;

     }
     else if(j%8==0 && j!=0 && j>mask)
     {
       s=s+"0.";
     }
```

```
   else if(j<=mask)
   {
     s=s+"1";
   }
   else
{
   s=s+"0";
}
}

 String s2[]=s.split("\\.");
 int arr1[]=new int[4];

 for(int i1=0;i1<s2.length;i1++)
 {
   arr1[i1]=Integer.parseInt(s2[i1],2);

 }

 System.out.println("enter any ip");
 String d_ip=sc.next();
 String s1[]=d_ip.split("\\.");
   try{
 if((s1.length>4||s1.length<4))
 System.out.println("invalid ip");
 else
 {
   for( i=0;i<s1.length;i++)
   {
     if(Integer.parseInt(s1[i])>255)
     {System.out.println("invalid ip");
     System.exit(0);}
     else
     continue;
   }

 }
 }catch(Exception e)
{
   System.out.println("invalid ip");
   System.exit(0);
}
int arr2[]=new int[4];
 for(int p=0;p<4;p++)
 {
   arr2[p]=Integer.parseInt(s1[p]);
 }
 String start="", end="";
 int q;
 System.out.println("first and last address are ");
 for(q=0;q<3;q++)
 {
```

```
        start=start+Integer.toString((arr1[q]&arr2[q]))+".";
        end=end+Integer.toString(((~arr1[q])|arr2[q])+256)+".";
     }
     start=start+Integer.toString((arr1[q]&arr2[q]));
        end=end+Integer.toString(((~arr1[q])|arr2[q])+256);
     System.out.println(start);
     System.out.println(end);
```

```
enter ip address and no of ip address required
1000
enter any ip
123.123.123.123
first and last address are
123.123.120.0
123.123.123.255
```

```
  }
}
```

3) subnetting

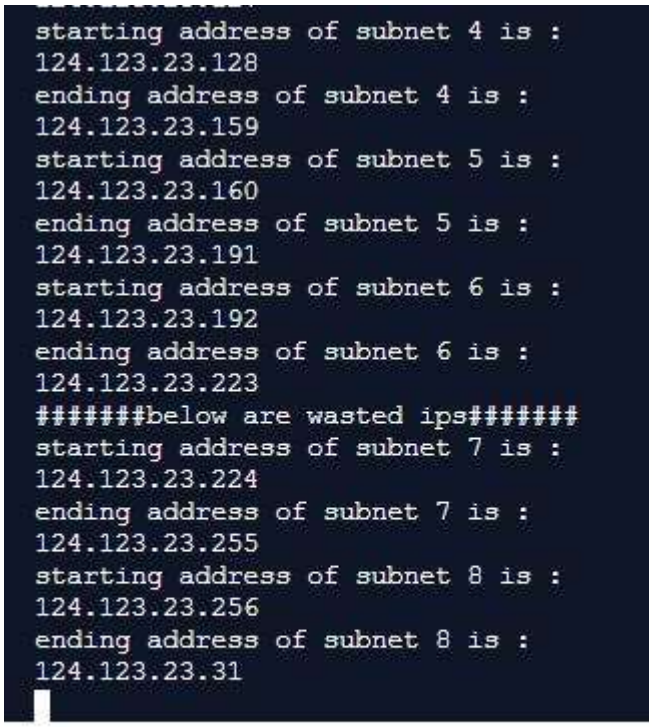
```
import java.util.*;

import javax.lang.model.util.ElementScanner6;
class Main {
  public static String cal_mask(int mask)
  {
    String s="";
    for(int j=1;j<=32;j++)
   {
     if(j%8==0 && j!=0 &&j<=mask )
     {
       s=s+"1.";
       //j--;

     }
     else if(j%8==0 && j!=0 && j>mask)
     {
       s=s+"0.";
     }

     else if(j<=mask)
     {
       s=s+"1";
     }
     else
   {
     s=s+"0";
   }
   }
   return s;
  }
  static boolean flag=true;
  public static int [] cal_start(int arr[])
  {
```

```
    {
      arr[3]=arr[3]+1;
      if(arr[3]>256)
      {
        arr[3]=0;
        arr[2]=arr[2]+1;
        return arr;
      }
      else
      {
        return arr;

      }



    }
  }
  public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    System.out.println("enter ip address: ");


    //String in=sc.next();
    String  ip=sc.next();
     System.out.println("enter number of subnets: ");
     int subs=sc.nextInt();
     System.out.println("enter the mask number: ");
     int m=sc.nextInt();
     int l1=0;
     int i;

    while(Math.pow(2,l1)<subs)
    {
      l1++;
    }

   int mask=m+l1;
   String ss=cal_mask(mask);


   int j1=0;

    String s2[]=ss.split("\\.");
    int arr1[]=new int[4];

    for(int i1=0;i1<s2.length;i1++)
    {
      arr1[i1]=Integer.parseInt(s2[i1],2);

    }
```

```
  String s1[]=ip.split("\\.");
    try{
  if((s1.length>4||s1.length<4))
  {System.out.println("invalid ip");
  System.exit(0);}
  else
  {
    for( i=0;i<s1.length;i++)
    {
      if(Integer.parseInt(s1[i])>255)
      {System.out.println("invalid ip");
      System.exit(0);}
      else
      continue;
    }

  }
  }catch(Exception e)
 {
   System.out.println("invalid ip");
   System.exit(0);
 }
 int arr2[]=new int[4];
  for(int p=0;p<4;p++)
  {
    arr2[p]=Integer.parseInt(s1[p]);
  }

  String start="", end="";
  int arr_end[]=new int[4];
  for(int k1=0;k1<4;k1++)
  {
    arr2[k1]=arr2[k1]&arr1[k1];

  }
for(int k=1;k<=Math.pow(2,l1);k++)
{
  System.out.println("starting address of subnet "+(k)+" is :");
   for(int ii=0;ii<4;ii++)
  {
    if(ii==3)
    System.out.print(arr2[ii]);
    else
    System.out.print(arr2[ii]+".");


  }
  System.out.println();


  for(int k1=0;k1<4;k1++)
  {
```

```
        arr_end[k1]=(~(arr1[k1])|arr2[k1])+256;

      }
      System.out.println("ending address of subnet "+k+" is :");

       for(int iii=0;iii<4;iii++)
      {
         if(iii==3)
        System.out.print(arr_end[iii]);
        else
        System.out.print(arr_end[iii]+".");

      }
      System.out.println();
      arr2=cal_start(arr_end);
      if(flag && k>=subs)
      {
        System.out.println("#######below are wasted ips#######");
        flag=false;

      }


    }
}
}
```

```
enter ip address:
124.123.23.34
enter number of subnets:
6
enter the mask number:
24
starting address of subnet 1 is :
124.123.23.32
ending address of subnet 1 is :
124.123.23.63
starting address of subnet 2 is :
124.123.23.64
ending address of subnet 2 is :
124.123.23.95
starting address of subnet 3 is :
```



CONCLUSION: there both classless and classfull addressing were studied sucessfully .

Post Lab Questions

1. Which of the following is private IP address?

   A. 12.0.0.1 B. 168.172.19.39

   C. 172.15.14.36 D. 192.168.24.43

   ANS D

2. Which class of IP address provides a maximum of only 254 host addresses per network ID?

   A. Class A

   B. Class B

   C. Class C

   D. Class D

   ANS: class c

3. What is the address range of a Class B network address in binary?

   A. 01xxxxxx

   B. 0xxxxxxx

C. 10xxxxxx

D. 110xxxxx

ANS: C

4. Which two statements describe the IP address 10.16.3.65/23?

   1.The subnet address is 10.16.3.0 255.255.254.0.

   2.The lowest host address in the subnet is 10.16.2.1 255.255.254.0.

   3.The last valid host address in the subnet is 10.16.2.254 255.255.254.0.

   4.The broadcast address of the subnet is 10.16.3.255 255.255.254.0.

   A. 1 and 3

   B. 2 and 4

   C. 1, 2 and 4

   D. 2, 3 and 4

   ANS:B

5. What is the maximum number of IP addresses that can be assigned to hosts on a local subnet that uses the 255.255.255.224 subnet mask?

   A. 14 B. 15

   C. 16 D. 30

   Ans: D

6. You need to subnet a network that has 5 subnets, each with at least 16 hosts. Which classful subnet mask would you use?

   A. 255.255.255.192 B. 255.255.255.224

   C. 255.255.255.240 D. 255.255.255.248

   ANS:B

7. You have a network that needs 29 subnets while maximizing the number of host addresses available on each subnet. How many bits must you borrow from the host field to provide the correct subnet mask?

   A. 2 B. 3

   C. 4 D. 5

   ANS: D

Date: _____ Signature of faculty in-charge