MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Problem Statement:

Write a program to implement a subnet calculator

Name :Soham R Shirke

Roll No:047

Division: A3

Assignment:

1. IPv4 addresses and different classes

IP stands for **Internet Protocol** and **v4** stands for **Version Four** (IPv4)

IP (version 4) addresses are 32-bit integers that can be expressed in hexadecimal notation. For example, **192.0. 2.146** is a valid IPv4 address

IP is a part of the TCP/IP (Transmission Control Protocol/Internet Protocol) suite, where IP is the principal set of rules for communication on the Internet

Classes:

In the IPv4 IP address space, there are five classes: A, B, C, D and E. Each class has a specific range of IP addresses (and ultimately dictates the number of devices you can have on your network).

Class A Public

IP address belonging to class A are assigned to the networks that contain a large number of hosts.

- The network ID is 8 bits long.
- The host ID is 24 bits long.

Class B Public

IP address belonging to class B are assigned to the networks that ranges from medium-sized to large-sized networks.

- The network ID is 16 bits long.
- The host ID is 16 bits long.

Class C Public

- IP address belonging to class C are assigned to small-sized networks.
    - The network ID is 24 bits long.
    - The host ID is 8 bits long.

Class D Public

IP address belonging to class D are reserved for multi-casting. The higher order bits of the first octet of IP addresses belonging to class D are always set to 111

Class E Public

IP addresses belonging to class E are reserved for experimental and research purposes. IP addresses of class E ranges from 240.0.0.0 – 255.255.255.254. This class doesn't have any sub-net mask

2. Classful and Classless addressing

Classful routing protocols do not send subnet mask information with their routing updates. A router running a classful routing protocol will react in one of two ways when receiving a route:

• If the router has a directly connected interface belonging to the same major network, it will apply the same subnet mask as that interface.

• If the router does not have any interfaces belonging to the same major network, it will apply the classful subnet mask to the route.

Classless routing protocols do send the subnet mask with their updates.

Thus, Variable Length Subnet Masks (VLSMs) are allowed when using classless routing protocols.

3. Need of subnetting

The goal of subnetting is to create a fast, efficient, and resilient computer network. As networks become larger and more complex, the traffic traveling through them needs more efficient routes. If all network traffic was traveling across the system at the same time using the same route, bottlenecks and congestion would occur resulting in sluggish and inefficient backlogs.

Creating a subnet allows you to limit the number of routers that network traffic has to pass through. An engineer will effectively create smaller mini-routes within a larger network to allow traffic to travel the shortest distance possible.

4. Example of subnetting

5. Executable code with screenshots of the output window.

```
#include <iostream>
#include <math.h>
using namespace
std;

int calc_subnetmask(int nb, int x)
{
int t = abs(x - nb), k = 7, ans
= 0; while (t != 0)
{
if (k == -
1) k = 7;
```

```
ans += pow(2,
k); k--;

t--;

}

return ans;

}

int main()

{

cout << "<<----------SUBNET CALCULATOR   >>
"<<endl<<endl<<endl;


int ip[4] =
{0}; int nb,x;

cout << "Enter IP address (Use space instead of '.'): "
<< endl; for (int i = 0; i < 4; i++)

{

cin >> ip[i];

}

cout << endl;

cout << "Enter CIDR
length: "; cin >> nb;
```

```
cout << "\nIP Address :
"; for (int i = 0; i < 4;
i++)

{

if (i == 3)

{

cout << ip[i] << " / " << nb <<
endl; break;

}

cout << ip[i] << ".";

}

if (ip[0] >= 0 && ip[0] <= 127)

{

cout << "Class A" <<
endl; x = 8;

}

else if (ip[0] >= 128 && ip[0] <= 191)

{

cout << "Class B" << endl;

;
```

```
x = 16;
}
else if (ip[0] >= 192 && ip[0] <= 223)
{
cout << "Class C" <<
endl; x = 24;
}
else
{
cout << "Enter valid IP.   " << endl;
}
if (nb != 24 && nb != 16 && nb != 8)
{
int machines, n;
cout << "It is classless addressing\n";
int subnet_mask =
calc_subnetmask(nb, x); cout <<
"Subnet mask : " << subnet_mask; int
no_of_networks = pow(2, abs(x - nb));
cout << endl
<< "Number of networks : " << no_of_networks;
```

```
}

else

{

cout << "It is classful addressing\n";

}

return 0;

}
```