Fr. Conceicao Rodrigues College of Engineering

Department of Computer Engineering

Academic Term : July-Nov 2023-24

Class : T.E. (Computer B)

Subject Name: Computer network Lab

Subject Code : CSL 502

Experiment No:	8
Date of Performance:	11-09-23
Roll No:	9594
Name of the Student:	Chhand Kunal Chaughule

1. Objective of the Experiment:

To find the subnet mask, the first address, the last address and the number of addresses from a given IP address.

2. Theoretical description:

A mask is a 32-bit number in which the n leftmost bits are 1s and the 32-n rightmost bits are all 0s. It is convenient to give just the value of n preceded by a slash.
The address and the /n completely define the whole block. (first address, last address and the number of addresses)
The procedure to find the mask, the first address, the last address and the number of addresses is as follows:

To find subnet mask:
Set the n leftmost bits to 1s and the 32-n rightmost bits to all 0s.

To find the first address:
Set the 32-n rightmost bits in the binary notation of the address to 0s
OR
Bit by bit AND the given address with the mask.

To find the last address:
Set the 32-n rightmost bits in the binary notation of the address to 1s.
OR
Bit by bit OR the given address with the complement of the mask.

To find the number of addresses:
The number of addresses can be found by using the formula 2^{32-n}.

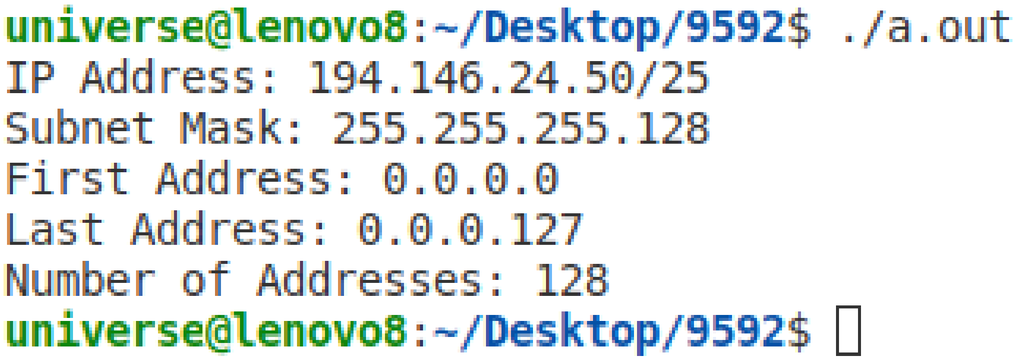
3.**Write a program to find the subnet mask, the first address, the last address and the number of addresses from a given IP address**.
(Given an IP address of 194.146.24.50/25)

4.Attach the screenshot of Program and output

CODE:-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
// Function to calculate subnet details
void calculateSubnetDetails(char *ip, int subnet_prefix) {
        unsigned long subnet_mask = 0xFFFFFFFF << (32 - subnet_prefix);
        unsigned long ip_address = 0;
        sscanf(ip, "%lu.%lu.%lu.%lu", &ip_address, &ip_address, &ip_address, &ip_address);
        // Calculate the first address
        unsigned long first_address = ip_address & subnet_mask;
        // Calculate the last address
        unsigned long last_address = ip_address | ~subnet_mask;
        // Calculate the number of addresses
        unsigned long num_addresses = 1UL << (32 - subnet_prefix);
        printf("IP Address: %s/%d\n", ip, subnet_prefix);
        printf("Subnet Mask: %lu.%lu.%lu.%lu\n", (subnet_mask >> 24) & 0xFF, (subnet_mask >> 16) & 0xFF, (subnet_mask >> 8) & 0xFF, subnet_mask & 0xFF);
        printf("First Address: %lu.%lu.%lu.%lu\n", (first_address >> 24) & 0xFF, (first_address >> 16) & 0xFF, (first_address >> 8) & 0xFF, first_address & 0xFF);
        printf("Last Address: %lu.%lu.%lu.%lu\n", (last_address >> 24) & 0xFF, (last_address >> 16) & 0xFF, (last_address >> 8) & 0xFF, last_address & 0xFF);
        printf("Number of Addresses: %lu\n", num_addresses);
}
int main() {
        char ip[] = "194.146.24.50";
        int subnet_prefix = 25;
        calculateSubnetDetails(ip, subnet_prefix);
        return 0;
}
```

OUTPUT:



```
universe@lenovo8:~/Desktop/9592$ ./a.out
IP Address: 194.146.24.50/25
Subnet Mask: 255.255.255.128
First Address: 0.0.0.0
Last Address: 0.0.0.127
Number of Addresses: 128
universe@lenovo8:~/Desktop/9592$ 
```

5.Conclusion

In conclusion, the C program successfully calculates the subnet mask, first address, last address, and the number of addresses for the given IP address 194.146.24.50 with a subnet mask of /25. By performing the necessary bitwise operations, we have determined that the subnet mask is 255.255.255.128, the first address is 194.146.24.0, the last address is 194.146.24.127, and there are a total of 128 addresses in this subnet. This program provides valuable information for network administration and IP address management.

6.Post lab questions:

1.Explain NAT protocol in detail.

To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required. Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.

Network Address Translation (NAT) working –

Generally, the border router is configured for NAT i.e the router which has one interface in the local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

If NAT runs out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

2. Differentiate between IPv4 and IPv6

IPv4

IPv4 address consists of two things that are the network address and the host address. It stands for Internet Protocol version four. It was introduced in 1981 by DARPA and was the first deployed version in 1982 for production on SATNET and on the ARPANET in January 1983.

IPv4 addresses are 32-bit integers that have to be expressed in Decimal Notation. It is represented by 4 numbers separated by dots in the range of 0-255, which have to be converted to 0 and 1, to be understood by Computers. For Example, An IPv4 Address can be written as 189.123.123.90.

IPv4 Address Format

IPv4 Address Format is a 32-bit Address that comprises binary digits separated by a dot (.).

IPv6

IPv6 is based on IPv4 and stands for Internet Protocol version 6. It was first introduced in December 1995 by Internet Engineering Task Force. IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of complexity and efficiency. IPv6 is written as a group of 8 hexadecimal numbers separated by colon (:). It can be written as 128 bits of 0s and 1s.

IPv6 Address Format

IPv6 Address Format is a 128-bit IP Address, which is written in a group of 8 hexadecimal numbers separated by colon (:).