SIMULATION AND WORKING OF

INTERNET PROTOCOL ADDRESSING

AND SUBNETTING

Report Submitted to the SASTRA Deemed to be

University as the requirement for the course

CSE302: COMPUTER NETWORKS

Submitted by

MUPPA CHINMAI RAM NAGA SAI PRASAD

(Reg. No.:122003158, CSE)

FERBRUARY 2021

SCHOOL OF COMPUTING THANJAVUR,

TAMIL NADU, INDIA – 613 401

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF COMPUTING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**Simulation and Working of Internet Protocol Addressing and Subnetting**" submitted as a requirement for the course, **CSE302: COMPUTER NETWORKS** for B.Tech. is a bonafide record of the work done by **Shri Muppa Chinmai Ram Naga Sai Prasad (Reg. No.122003158, Bachelor of Technology -Computer Science Engineering**) during the academic year 2020-21, in the School of Computing.

Project Based Work *Viva voc*e held on __________________

Examiner 1 **Examiner 2**

List of Figures

Figure No.	Title	Page No.
1	IPv4 or IPv6	31
2	Output for Class-A Address	41
3	Output for Class-B Address	41
4	Output for Class-C Address	42
5	Networks without Subnetting	44
6	Configuring Router0	45
7	Configuring Router1	46
8	Configuring Router0 Serial Port	46
9	Configuring Router1 Serial Port	47
10	Configuring Router0 IP Route	47
11	Configuring Router1 IP Route	48
12	Networks after Subnetting	49

List of Tables

Table No.	Table Name	Page No.
1	No. of Hosts and Network Bits for each Address Class	7
2	IPv4 Address Types	10 - 11
3	IPv4 VS IPv6	16
4	Class-A Subnets	19 - 20
5	Class-B Subnets	21
6	Class-C Subnets	22
7	Table showing different department subnet masks and notations	23
8	Subnetting Vs Supernetting	28

Abbreviations

IP	Internet Protocol
PC	Portable Computer
IPv4	Internet Protocol Version4
IPv6	Internet Protocol Version6
CISCO	Commercial & Industrial Security Corporation
IPNG	Internet Protocol Next Generation
IETF	Internet Engineering Task Force
ARPA	Advanced Research Project Agency
LAN	Local Area Networks
MAN	Metropolitan Area Networks
WAN	Wide Area Networks
SAGE	Semi-Automatic Ground Environment
IANA	Internet Assigned Numbers Authority
CIDR	Classless Inter Domain Routing
NAT	Network Address Translation
QoS	Quality of Service
RFC	Request For Comments
PPP	Point to Point Protocol
DHCP	Dynamic Host Configuration Protocol
TCP	Transmission Control Protocol
OSI	Open Systems Interconnections
MTU	Minimum Transmission Unit
VLSM	Variable Length Subnet Mask
FLSM	Fixed Length Subnet Mask
Y2K	Year 2000
CPU	Central Processing Unit
ISP	Internet Service Provider
CLI	Command Line Interface

PROBLEM STATEMENT

Identify different classes of IP Addresses and the size of the addresses.

Understand about the Internet Protocol Version 4 and addressing and their working.

Understand about different classes of IP Addresses such as Class-A, class-B, class-C, class-D and class-E IP Addresses.

Understand the different types of Internet Protocol Addresses and their advantages and disadvantages.

To understand about the concept of Subnetting and identify different classes of subnets such as the Class A, Class B, Class C, Class D and Class E subnets.

Understand the differences between Subnetting and Supernetting.

Understand about the Classless Inter Domain Network (CIDR) Notation.

OBJECTIVE

Find about the type of IP Addresses, such as IPv4 and IPv6. Develop a code to classify the addresses based on their range, which allows computers to send and receive information.

Develop a code for Internet Protocol Subnetting in any programming language.

Simulate the Subnetting concept with the help of CISCO Packet Tracer.

Use the Simulation to demonstrate it, which ensures that traffic destined for a device within a subnet, which reduces congestion and improves network performance, speed and administration.

Finally, point out the loop holes in the entire work and try to solve the problem of these loopholes in the future course of work.

ABSTRACT

IP Address is a number that is related to a particular computer or network. Current Internet protocol IPv4 is replaced with IPv6. The process of splitting a network into two or more networks is known as the Subnetting.

For systems to find one another in a distributed environment, nodes are provided with specific addresses that unambiguously identify the actual network the system is on and unambiguously identify the system to explicit networking.

When the two identifiers are mixed, the result's a universally unique address. This address is called as IP Address or just as IP. It may be a code created of numbers divided by three dots that identifies a selected PC on the Internet. These addresses are literally 32 bit binary numbers.

The later generation Internet Protocol, at the start called IP Next Generation (IPNG), and then further as IPv6, has been developed by the Internet Engineering Task Force (IETF) to switch the present web protocol (also referred to as IPv4), which provides 2128 potential number of locations.

To facilitate the combining of IPv6 into present networks, many transition mechanisms are planned by the IETF IPNG Transition working party.

This work inspects and through empirical observation evaluates two transition mechanisms, particularly IPv6 to IPv4 tunneling, and dual-stack process, as they associate to the working of Internet Protocol Version 6.

The main objective of this paper is to check and analyze IPv4 and IPv6 networks, examine their properties and header formats.

This paper conjointly makes an attempt to stipulate the key deployment problems and security-related challenges that are being encountered and managed with during the migration method.

We can protect our network and improve the working of the network with the assistance of Subnetting.

Keywords: Internet Protocol Address, IPv4, IPv6, Subnet, Tunneling, Network

Table of Contents

Title	Page No.
Bonafide Certificate	ii
List of Figures	iii
List of Tables	iv
Abbreviations	v
Problem Statement and Objective	vi
Abstract	vii
1. Introduction	1
2. Proposed Work	29
3. Code Snippets	30
4. Simulation	43
5. Results	50
6. Conclusion and Future Plans	51
7. References	52

INTRODUCTION

1.1. Computer Network:

If two nodes connected by a single link, then it is called as a Network. A computer network can be considered as a cluster of systems of systems and alternative computing hardware devices that are attached by communication channels, to provide communication and resource-sharing among a large range of users. Networks are divided by their characteristics.

One of the first examples of computer network was a network of information exchanging computers that served as a part of the United States Military Semi-Automatic Ground Environment. The network that became the idea for the Internet was ARPANET. It was developed by the US Advanced Research Projects Agency i.e. ARPA. It’s this network that emerged to what we now call as the Internet.

Networks are helpful for:

- Resource Division
- Connecting multiple computers to send and receive data
- Remote Area – connectivity (Very long distance)
- Accuracy – transmitted, received data
- Easy access of information
- Enables communication via e-mail, videos etc.

Topology of network:

- LAN
- MAN
- WAN
- INTRANET
- INTERNET

★ INTERNET = LAN + MAN + WAN + INTRANET

Communication Model:

Parameters:

- Sender
- Receiver
- Channel
- Data
- Protocol i.e. Set of Rules for Data Transmission

1.2. What is a Network?

It is said that a network in the world of computers is a set of interconnected hosts that can be wired or wireless, via some shared media.

A network of computers helps its hosts to share and exchange data and Data related to the media. The Network may be a LAN across the office or MAN across a city or WAN which is distributed across cities.

A computer network may be as easy as two PCs connecting to a computer network together by a unique copper wire or it can be developed to the complexity that is attached to one another, known as the Internet.

A Network comprises huge number of components to achieve its final target of sharing the data. Nature of components in Computer Network is given below:

- **Hosts:** It is said that the hosts are located at the end of the network. A Host is a medium of data and other host will be the target. Among hosts, data flows end to end. A host can be a server, a database server or a system.

```
  N1------------------------ (Path / Link) -----------------------N2
(Host)                                                          (Host/End System)
```

- **Media:** It may be a copper cable, co-axial cable or copper cable if wired. It can be free-to-air radio frequency or a special wireless band if wireless. It's easy to use wireless frequencies. Remote sites are also interconnected.

- **Hub:** It is used in a multi-port to connect hosts in a segment LAN. It is also a multi-port repeater.
  Because of low throughputs, hubs are now seldom used. The hub operates on the OSI Model Layer-1 (Physical Layer).

- **Switch:** A switch is a bridge for multi-ports and is used for linking hosts in a segment LAN.
  Switches are much quicker than hubs and operate on the Wire Velocity.
  The switch operates on Second Layer i.e. data Link Layer, but third layer i.e. Network Layer switches are also usable.

  Various Switching Mechanisms:

  1. Circuit(Telephone):
     Circuit already established
     Voice Conversation on Telephone
     Static-fixed path

  2. Path(Between Computers-Data)
     Path on Demand Basis
     According to Network Traffic
     Dynamic

- **Router:** Third Layer i.e. Network Layer device that produces data routing choices directed for some data or information is called Router.

- **Firewall:** It is software or a blend of hardware and software used to secure information about users from unknown users on the Internet/Network.

A numerical address is an IP Address (Internet Protocol Address) attributed to each device such as Computer, Printer etc. It takes part in a network of machines using the Internet Communication Protocol. Two main functions are served by the IP Address. They are:

1. Network or Host Recognition
2. Addressing of Location

“A name shows what we explore. An address shows where it is. A path shows how it is to get over there”.

The fast Internet explosion and the presence of high levels of Speed networks for wireless and broadband have contributed in order to deplete IPv4. IP protocol was established far more than three decades before, with roughly an address space of billion does not cater the current needs. The Internet Assigned Numbers Authority (IANA) assigned the final chunk of IP addresses to the Regional Internet Registries declaring end of IPv4 addresses on Feb 3, 2011.

The depletion of addresses has presented a serious problem for internet network development. Temporary solutions such as CIDR (classless inter domain routing), PPP/DHCP (address sharing) and NAT (translation of network addresses) does not appear to help taking into account the number of devices that are frequently accessing the internet every day. As the protocol was created a far time ago, the mobility, protection and QoS (Quality of Service) features are also maintained by extra protocols that are not possible to implement into the protocol.

1.3. IP Address

We recognize that an Internet Protocol address (IP address) is an IP address. IP Addresses comes under the Network Layer. Each system such as machine, printer participating in a computer network using the Internet Protocol for communication is assigned a numerical mark. There are 5 Internet Protocol Address Classes. Blocks of "Class A” (2^{24} addresses, nearly 16.7 million). The blocks "Class B" (2^{16} addresses, i.e. 65536) and "C" (2^{8} addresses, i.e. 256) were rendered open to smaller networks. Early architecture of networks is restricted only to these three sizes. They are used in LAN and WAN.

Different classes of IP Addresses are of classes A, B, C, D and E.

1. **Class A Address:**

It always sets the first bit of the first octet to zero. Therefore, the first octet ranges from 1 to 127

00000001 – **0**1111111 (1 – 127)

Class A addresses only contain IP addresses that start from 1.x.x.x to 126.x.x.x only. The 127.x.x.x Internet Protocol range is reserved for IP addresses with loopback. Default "Class A" IP address subnet mask is 255.0.0.0, which means that 126 (2^7-2 networks) and 16777214(2^{24}-2) can be used for Class A addressing. "Class A" IP Address has the format:

NNNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

2. **Class B Address:**

It is a class of IP Address in which first two bits in the first octet set to 10. Therefore, the first octet ranges from 128 – 191.

10000000 – **10**111111 (128 – 191)

Class B IP Address contains IP Addresses that start from 128.0.x.x to 191.255.x.x. Default "Class B" IP Address subnet mask is 255.255.x.x. 16384(2^{14}) Networks and 65534(2^{16} -2) Host Addresses are given by Class B IP Address.

"Class B" Subnet Mask has the format:

NNNNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

3. Class C Address:

It is a class of IP Address in which first three bits of first octet initialized to 110. Therefore, the first octet ranges from 192 – 223.

11000000 – **110**111111 (192 – 223)

Class C Addresses contains IP Addresses that start from 192.0.0.x to 223.255.255.x. Default “Class C” IP Address subnet mask is 255.255.255.x. 2097152 (2^{21}) Network and 254 (2^{8}-2) Host Addresses are given by Class C IP Address.

“Class C” Subnet Mask has the format:

NNNNNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

Where;

‘N’ indicates the Network bit,

“H” indicates the Host bit.

4. Class D Address:

It is a class of IP Address in which first four bits of first octet are initialized to 1110. Therefore, the first octet ranges from 224 – 239.

11100000 – **1110**1111

Class D Addresses contains IP Addresses that start from 224.0.0.0 to 239.255.255.255. This class of IP Address is taken for Multi-Casting. In the case of Multi-Casting, information is not directed for a specific host. So, bringing host addresses from IP Addresses is not needed.

“Class D” Addresses have no Subnet Mask.

5. **Class E Address:**

This class of IP Address is used for experimental purposes, Research & Development. Class E Addresses contains IP Addresses that start from 240.0.0.0 to 255.255.255.254.

“Class E” IP Addresses also does not contain any Subnet Mask.

Table showing the Number of Host and Network bits for each class of Address:

Address Class	Number of Host Bits	Number of Network Bits
A	24	8
B	16	16
C	8	24

No. of Hosts and Network Bits for each Address Class

Priority bits for each class:

1. Class A = 0
2. Class B = 10
3. Class C = 110

1.4. TYPES OF IP ADDRESS:

IP Addresses are mainly divided into two types.

They are:

- Public IP Address
- Private IP Address

1. **Public IP:**

A system on the Internet is recognized by its IP Address. Each Computer is assigned a Public IP Address, which attaches to the Internet where every IP is distinct. In this scenario, there cannot be two computers with all over the Internet with the same public IP address.

This method of addressing makes it possible for the computers to find each other and exchange information.

The user does not have any power over the public IP Address.

The public IP Address is provided by the Internet Service Provider (ISP) to the system. A Public Internet Protocol Address may be static or dynamic.

A static IP address for the public does not alter and is used for the hosting of web pages or services mainly on the Web.

A dynamic IP Address on the other hand is selected from a bunch of Existing Addresses and varies each time a user switches to the Internet. Many users will have a dynamic IP allocated to their system that goes off when the machine is disconnected from the Site. When it is again connected, system receives a new IP.

2. **Private IP:**

When the IP Address falls inside one of the private reserved IP Address range Networks such as LAN, it is deemed to be a private IP Address.

The Authority for Internet Assigned Numbers (IANA) has the following three blocks of IP address space have been reserved for Local Networks or private networks:

They are:

- 10.0.0.0 – 10.255.255.255 (No. of Addresses: 16,777,216)
- 172.16.0.0 – 172.31.255.255 (No. of Addresses: 1,048,576)
- 192.168.0.0 – 192.168.255.255 (No. of Addresses: 65,536)

The private IP addresses used to number the computers that are used in a private network, such as house, school and company LANs in hotels and airports, which make it possible to connect with each of the computers in the network.

1.5 IPv4

One of the major TCP/IP protocols is the Internet Protocol. The protocol operates on the network layer of OSI model and the TCP/IP model's Internet layer. This protocol is also responsible for defining hosts. The Internet Protocol Model IPv4 is established upon their logical addresses and to route information between the basic addresses.

IP provides a method for uniquely distinguishing hosts by a system of Internet Protocol Addressing method. IP uses the best distribution effort, i.e. it does not guarantee the delivery of packets to the target host, but to reach the destination, it will do its best.

Most widely used form of address is IPv4 Address.

Version 4 of the Internet Protocol uses logical 32-bit addresses. The Internet Protocol is a layer 3 (OSI) protocol that takes data from Layer-4 (Transport) segments and divides them into packets. The IP packet wraps the obtained data unit from the layer above, and connects details to its own header.

Version 4 (IPv4) of the Internet Protocol is the fourth version of Protocol on the Internet (IP). It is among one of the fundamental protocols of Internet methods for standards-based internetworking, and was introduced for manufacturing in the 1983 ARPANET. Much Internet traffic is still redirected, despite the continuous implementation of a successor protocol, Internet Protocol Version 6 (IPv6). In the IETF publication RFC 791 (September 1981) IPv4 is defined, replacing an earlier description (RFC 760, January 1980). For use on packet-switched networks, IPv4 is a connectionless protocol. It works on the best efforts of model distribution, in that it does not certify delivery, nor this ensures correct sequencing or avoidance of false service. 32-bit (four-byte) addresses are used for IPv4, which restricts the address space to 2^{32} addresses. The production of IPv6 in the 1990s was stimulated by this restriction, and it was commercially deployed since

2006. The limited address space suffered fatigue on Feb 3, 2011 due to the demand of expanding internet.

Having been considerably delayed by classy design of the network, Classless Inter-Domain Routing, Translation of addresses and network addresses (NAT).

Private Network special address blocks of approximately 18 million and multi-cast addresses of approximately 270 million are reserved by IPv4.

Address range of IPv4 addresses is from 0.0.0.0 to 255.255.255.255 i.e. from A to E IP Class types. The value of any segment or byte ranges from 0 and 255 (both included).

Example: Command prompt> ping 127.0.0.1 (Loop Back Address)- It Gives our computer details.

Net ID: Network Address

Host ID: Hosts Connected to Network (Number of Hosts –ex: 100 systems)

IPv4 Address Types:

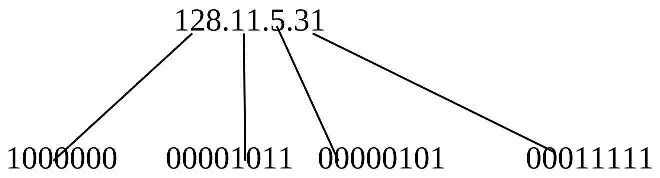
The below table shows different IPv4 Address Types and their purposes with an example:

IPv4 Address Type	Function	Example
Uni-cast	Delivers packet to a single host	192.0.0.1
Multi-cast	Delivers to a group of hosts	224.0.0.1
Broadcast	Delivers to each host	192.168.1.255

Loopback	Delivers to itself	127.0.0.1
Unspecified	Not known Network	0.0.0.0
All-host broadcast	Broadcasts to each host on local- link	255.255.255.255
Directed Broadcast	Broadcast to a certain network	192.168.2.255

<u>IPv4 Address Types</u>

Dotted Decimal Representation of IPv4:



1.6. 127.0.0.1:

127.0.0.1 is often referred to as the 'localhost' loopback Internet protocol address.

This address is used to establish an IP connection to the same machine or to the device that the end-user is operating.

The most common practice is to create a connection using the address 127.0.0.1; however, using any IP address in the 127.x.x.x range would operate in the same or similar way.

The loopback construction provides the ability to validate or build the IP stack on the system to a computer or to a device capable of networking.

The establishment of a network link to the loopback address 127.0.0.1 is done in the same way as the establishment of one on the network with any remote machine or unit. The primary distinction is that the link restrains from using the hardware of the local network interface.

To test software, computer administrators and software developers usually make use of 127.0.0.1.

255.0.0.1 Subnet Mask would usually be allocated while creating an IPv4 link with 127.0.0.1.

If a packet addressed to the loopback IP address is received by any public switch, router, or gateway, it is required to drop the packet without typing the information.

As a consequence, if a data packet is sent beyond the localhost, it won't mistakenly arrive at a device that will attempt to address it.

This feature of the loopback helps to preserve network security.

Advantages for 127.0.0.1:

Sending a ping request to 127.0.0.1 is a common technique to check that the networking equipment, operating system, and TCP/IP implementation of a device are functioning correctly.

Administrators or device users may solve network communication problems based on the outcomes of the test.

By using the 127.0.0.1 address, advantages of loop back address is to deceive computer security or computer students into trying to crack, test, or test network speed.

The user scans the stored Hosts file for domain name resolution. In the Hosts log, the 127.0.0.1 Internet Protocol address is often identified on computers attributed to the address "localhost." In order to stop the end user from seeking licit information security assistance with malware infection, computer malware is often used to attribute licit websites to the local host.

IP addresses for their intended recipients are found in the messages created by TCP/IP application software. As a special IP address, TCP/IP recognizes the 127.0.0.1 Address.

Before sending data to the physical network, the protocol tests each data. It then immediately re-routes any data to the collecting end of the TCP/IP stack with a target of 127.0.0.1.

Addresses that start from 127.0.0.0 up to 127.255.255.255 are reserved by IPv4, while 127.0.0.1 is used as the loopback address in most of the scenarios.

Not any one of the private IP address ranges specified in IPv4 belongs to 127.0.0.1 and 127.0.0.0 Network locations.

Often, people who research computer networking confuse 127.0.0.1 with the address 0.0.0.0.0. Although both have unique IPv4 definitions, 0.0.0.0 does not have any functionality for loopback.

1.7. IPv6

Version 6 of Internet Protocol, also known as IPNG (IP next Generation) is the latest Internet edition of the IP. IPv6 arrives with an address scheme of 128 bits, enough to cover nearly every linked computer with a global unique address on earth. IPv6 utilizes 128 bit addresses with 2^{128} address space addresses.

For every computer and user, such a large address space allows them to connect to the Internet in the world. It also helps in removing the usage of NAT in IPv6 and boosts connectivity, flexibility and reliability in the network. IPv6 had to support larger space for emails, protection in the protocol and multimedia transmission in real time. IPSec support, unlike in IPv4 where it was optional, has become a mandatory requirement in IPv6. Identification of payload (used in QoS) has been replaced by the field of Flow Mark Field in packet of IPv6.

The fragmentation concept has been eliminated. Extension headers in IPv6 have been substituted for the checksum and options. In addition, IPv6 does not need manual configuration or DHCP because the device is involved in automatic "stateless" configuration, one of IPv6's design objectives.

Finally, the size of the packet header was also increased from 20 bytes in IPv4 to 40 bytes in IPv6. The Address combinations that are possible for IPv6 are 340 Undecillion.

A. Header Design of IPv6:

1. The header length has been increased from 20 to 40 bytes.
2. For address, IPv6 has 16 bytes i.e. 128 bits.
3. Header Fields have been decreased from 12 to 8.
4. No option fields are present in IPv6.
5. For greater functionalities, Extension Headers are used.

B. Extension Headers of IPv6:

IPv6 Extension Header Fields are given below:

1. *Hop by Hop Operation:* This option is taken when the data is transferred to all routers by the source visited by a datagram. So far, only 3 options are defined: Pad-1, Pad-n, Jumbo payload.

For alignment purposes, Pad-1 option having 1 byte length is designed. Pad-n option is same as pad-1, except that it is used for alignment purposes when 2 or more bytes are used. If the payload
length is more than 65,535 bytes, then Jumbo payload is referred.

2. *Fragmentation:* In IPv6, only fragments can be rendered from the original source. A source locates the smallest MTU value supported on the path by any network with the help of path MTU Discovery process.

3. *Authentication:* This header carries out the message sender's validation and guarantees that data integrity is maintained.

4. *Encrypted Security Protocol:* This header provides anonymity and protects against eavesdropping.

5. *Source Routing:* It includes the idea of a strict source route and, as in IPv4, and a loose source route. The source uses the strict source route for a predetermined route for the datagram as it passes across the internet. For a particular form of service, such as minimum delay or max throughput, the sender may make a choice of path. It can also select a more convenient and more efficient route for the purpose of the sender.

6. *Destination Option:* It is used when the source just transfers the data to the intended destination.
   The routers in between are not allowed to access this data.

C. Transition from IPv4 to IPv6:

The transition between today's IPv4 Internet and the IPv6 Internet, the Internet of the future, is going to be a long process when both protocols exist concurrently.
A framework for maintaining stepwise and independent change is necessary to migrate to IPv6 services. The smooth coexistence of IPv4 and IPv6 nodes during the transition time must be supported by such a mechanism.
To facilitate the smooth transition from IPv4 to IPv6 services, IETF has created the Ngtrans Community.
It is possible to classify the different transformation methods broadly into three groups, namely dual stack, tunneling and translation mechanisms.

1.8. IPv4 VS IPv6

The following table shows the difference between IPv4 and IPv6 Addressing:-

Property	IPv4	IPv6
Address	4bytes (32 bits)	16 bytes (128 bits)
Packet Size	576 bytes	1280 bytes
Fragmentation of Packets	Fragmentation is Optional Routers and sending hosts.	Packet Size is 1280 bytes without fragmentation.
Header of Packet	Does not recognize flow of packet for QoS Handling. Check Sum Dependent. Options till 40 bytes are included.	Contains the Flow Label area that defines QoS handling packet flow. Not depends on a check Sum. For Optional Data, Expansion headers are used.
DNS Records	Records of Address (A) and maps names of hosts.	Records of Address (AAAA) and maps names of hosts.
Configuration of Address	Manual or via DHCP	SLAAC (Stateless address auto configuration) using ICMPv6(Internet Control Message Protocol version 6)
Format of Address	Dotted Decimal 192.168.120.2	Hexadecimal Notation 2002:0BD8:0324:AB00: 0134:4678:8901:ABCD
IPSec	Optional	Mandatory
Checksum Header	Yes	No
Broadcast and Multicast	Yes	No

IPv4 VS IPv6

1.9. SUBNETTING

Each Internet Protocol class has its own default subnet mask that needs a prefixed number of networks and a prefixed number of hosts per network for that Internet Protocol Class. The versatility of having fewer hosts per network or more networks per IP class is not supported by classic IP addressing. Initially, subnets were planned to fix the lack of IP addresses over the Internet.

The versatility of borrowing parts of the host portion of the IP address and using them as a network in the network, called a subnet, is provided by CIDR. One single “Class A” IP address can be used to provide smaller sub networks that has improved network management by using subnetting.

Dividing the network into a variety of subnets offers the following advantages:

- Reduces the traffic of network by reducing the amount of broadcasts.
- Helps exceed the limitations of a LAN.
  Ex: the maximum number of permissible hosts.
- Allows users to access a work network from their own work network homes. No need to unlock the complete network.
- The subnets can be further broken into sub-subnets.

The Variable Length Subnet Mask (VLSM) is a method of splitting an IP space, without wasting IP addresses, into subnets of different sizes.

When we do sub-netting, all sub-networks have the same number of hosts, known as FLSM (Fixed length subnet mask).

Subnetting aims to build a computer network that is fast, efficient, and resilient. The traffic flowing through them requires more efficient routes as networks grow wider and more complex.

If all network traffic traveled through the grid using the same path at the same time, bottlenecks and congestion will result in slow and not efficient backlogs.

SUBNET MASK:

The 32-bit Internet Protocol address contains about the host and its network information. It is very important that both are differentiated. Routers use the Subnet Mask to do this, which is as long as the IP address is the size of the network address.

The Subnet Mask may have a length of 32 bits. If the binary IP address is ANDed with its Subnet Mask, the network address returns the result.

For example, say 192.168.0.0 is the IP address and 255.255.255.0 is the subnet mask, and then:

IP	192.168.0.0	11000000	10101000	00000000	00000000
Mask	255.255.255.0	11111111	11111111	11111111	00000000
Network	192.168.0.0	11000000	10101000	00000000	00000000
			Final Result		

The subnet mask thus helps to extract the network ID Number and the Host Number from the Internet Protocol address.

It can now be found that the network is 192.168.0.0. The host on the above network is 192.168.0.0. Both may be same or they may differ.

We always reserve an Internet Protocol address to recognize the subnet and another one to mark the subnet address of the broadcast.

A Way to find Subnet Mask on Windows Computer:

Go to the Run box (Windows Key + R) and cmd to open the Command Prompt to find the subnet mask for your Windows device. You can input the ipconfig /all command here and press the Enter key. The subnet mask will be listed in the output under "Ethernet Adapters - Local Area Connection" as one of the parameters.

Alternatively, open the Network & Internet Control Panel -> Network & Sharing Center-> Local Area Connection, and then click the Details button.

Here, along with other details, such as the default gateway and DNS servers, you will see the IPv4 subnet mask.

Types of Subnet Classes:

1. **Class A Subnets:**

In this class, the satrting octet only is used as a network identifier and the remaining three octets are used to allocate hosts**.**

Bits from the host component are borrowed to build more subnet in Class A, and the subnet mask is altered accordingly.

For instance, if one MSB (Most Significant Bit) is taken from second-octet host bits and added to the network address, two subnets are formed with 2^{23}-2 subnet hosts.

To represent Subnetting, the Subnet mask is altered accordingly. A list of some possible combinations of "Class A" Subnets is given below:

Network	Mask of Subnet	Borrowed Bits	Subnets	Hosts per Subnet
8	255.0.0.0	Zero	1	16777214
9	255.128.0.0	One	2	8388606
10	255.192.0.0	Two	4	4194302
11	255.224.0.0	Three	8	2097150
12	255.240.0.0	Four	16	1048574

13	255.248.0.0	Five	32	524286
14	255.252.0.0	Six	64	262142
15	255.254.0.0	Seven	128	131070
16	255.255.0.0	Eight	256	65534
17	255.255.128.0	Nine	512	32766
18	255.255.192.0	Ten	1024	16382
19	255.255.224.0	Eleven	2048	8190
20	255.255.240.0	Twelve	4096	4094
21	255.255.248.0	Thirteen	8192	2046
22	255.255.252.0	Fourteen	16384	1022
23	255.255.254.0	Fifteen	32768	510
24	255.255.255.0	Sixteen	65536	254
25	255.255.255.128	Seventeen	131072	126
26	255.255.255.192	Eighteen	262144	62
27	255.255.255.224	Nineteen	524288	30
28	255.255.255.240	Twenty	1048576	14
29	255.255.255.248	Twenty One	2097152	6
30	255.255.255.252	Twenty Two	4194304	2

Class-A Subnets

The very first and last IP address of a subnet is also in the event of subnetting, used for both the subnet number and the subnet Broadcast. Since hosts cannot be allocated these two Internet Protocol addresses, sub-netting cannot be enforced as Network Bits by using more than 30 bits, in which less than two hosts per subnet are sustained.

2. Class B Subnets:

By default, 14 bits are used as network bits that provide (2^{14}) 16384 Networks and (2^{16} -2) 65534 Hosts, using Classful Networking.

It is possible to subnet Class B IP addresses the same way as Class A addresses by borrowing bits from host bits.

All possible combinations of Class B subnets are given below:

16	255.255.0.0	Eight	256	65534
17	255.255.128.0	Nine	512	32766
18	255.255.192.0	Ten	1024	16382
19	255.255.224.0	Eleven	2048	8190
20	255.255.240.0	Twelve	4096	4094
21	255.255.248.0	Thirteen	8192	2046
22	255.255.252.0	Fourteen	16384	1022
23	255.255.254.0	Fifteen	32768	510
24	255.255.255.0	Sixteen	65536	254
25	255.255.255.128	Seventeen	131072	126
26	255.255.255.192	Eighteen	262144	62
27	255.255.255.224	Nineteen	524288	30
28	255.255.255.240	Twenty	1048576	14
29	255.255.255.248	Twenty One	2097152	6
30	255.255.255.252	Twenty Two	4194304	2

Class-B Subnets

3. Class C Subnets:

As it can only have 254 hosts in a network, Class C IP addresses are typically allocated to a very limited network size.

Here is a list of all possible combinations of Class B IP address subnets:

Network	Mask of Subnet	Borrowed Bits	Subnets	Hosts per Subnet
24	255.255.255.0	Sixteen	65536	254
25	255.255.255.128	Seventeen	131072	126
26	255.255.255.192	Eighteen	262144	62
27	255.255.255.224	Nineteen	524288	30
28	255.255.255.240	Twenty	1048576	14
29	255.255.255.248	Twenty One	2097152	6
30	255.255.255.252	Twenty Two	4194304	2

Class-C Subnets

ISP’s can face a situation in which, according to customer requirements, they need to assign IP subnets of various sizes.

One customer may request 3 IP addresses from a Class C subnet, and another may request 10 IPs.

It is not feasible for an ISP to break the IP addresses into fixed size.

Instead, he may want to subnet the subnets in a way such that minimal wastage occurs.

An administrator, for example, has a network of 192.168.1.0/24. Suffix /24 (pronounced "slash 24") specify the number of bits used for the address of the network.

Example:

In this case, there are three separate departments with the administrator having varying number of hosts.

There are 100 computers in the Transactions department, 50 computers in the buying department, 25 computers in accounts and 5 computers in management.

The subnets in CIDR have a fixed scale. The administrator will not meet all the specifications of the network using the same technique.

As mentioned in the example, the following procedure demonstrates how VLSM can be used to assign department-wise IP addresses.

STAGES:

The different stages involved in the procedure are as follows:

1. **Stage-1:**

Create a list of potential subnets.

Mask of Subnet	Notation of Slash	Hosts per Subnet
255.255.255.0	/24	254
255.255.255.128	/25	126
255.255.255.192	/26	62
255.255.255.224	/27	30
255.255.255.240	/28	14
255.255.255.248	/29	6
255.255.255.252	/30	2

Table showing different department subnet masks and notations

2. **Stage-2:**

In descending order, sort the IP criteria i.e. from Highest to Lowest:

They are given below:

For Transactions – 100

For Buying – 100

For Accounts – 100

For Management– 100

3. **Stage-3:**

Assign the maximum IPs range to the peak requirement. So let's allocate the Transactions department the IP 192.168.1.0 /25 (255.255.255.128).

This subnet of IP with 192.168.1.0 Network Number has 126 no. of correct Host IP Addresses that meet the Transactions department's requirements.

There are 10000000 as the last octet in the subnet mask used by this subnet.

4. **Stage-4:**

Assign the next maximum IPs range 192.168.1.128 /26 (255.255.255.192) to the Buying Department.

This subnet of IP with 192.168.1.128 Network Number has 62 no. of correct Host IP Addresses that meet the Buying department's requirements.

In the final octet, the subnet mask used has 11000000.

5. **Stage-5:**

Assign the next maximum IPs range 192.168.1.192 /27 (255.255.255.224) to the Accounts Department.

This subnet of IP with 192.168.1.192 Network Number has 30 no. of correct Host IP Addresses that meet the Accounts department's requirements.

In the last octet, the subnet mask used has 11100000.

6. Stage-6:

Assign the next maximum IPs range 192.168.1.224 /29 (255.255.255.248) to the Management Department.

This subnet of IP with 192.168.1.224 Network Number has 6 no. of correct Host IP Addresses that meet the Management department's requirements. In the last octet, the subnet mask used has 11111000.

The administrator should subnet the IP subnet in such a way that the least amount of IP addresses is wasted by using VLSM.

The administrator, in this case, is still left with plenty of IP addresses even after assigning IPs to every department, which was not possible if he used CIDR.

1.10. CIDR:

It stands for Classless Inter Domain Routing.

As a standard scheme for routing network traffic across the Internet, CIDR was developed in the 1990s.

Why should we use CIDR?

Internet routers handled network traffic based on the class of IP addresses before CIDR technology was created.

In this scheme, for the purposes of routing, the importance of an IP address specifies its sub-network.

CIDR is a substitute to conventional Internet Protocol Subnetting that orders the IP Addresses, independently of the importance of the addresses themselves, into sub-networks.

As it essentially enables several subnets to be clustered together for network routing, CIDR is also known as Supernetting.

Notation of CIDR:

Using a combination of an IP address and its associated network mask, CIDR specifies an IP address range.

In the following style, CIDR notation is used:

xxx. xxx. xxx. xxx /n

Here; ‘n’ denotes the no. of leftmost or ‘1’ bits present in the mask.

1.11. SUPERNETTING:

The opposite of Subnetting is Supernetting.

The method by which a bunch of contiguous subnet networks are summarized back into a single wide network is known as Supernetting.

It is also called as the summarization of routes and grouping of routes.

We shift mask bits to the left for Supernetting, instead of shifting mask bits to the right of the default mask for subnetting. The size block of each network must be equal and should be in the 2^n format.

With Subnetting, at the cost of the host address, we generate more network addresses.

We produce more host addresses with Supernetting at the cost of network addresses.

Supernetting is about counting on orders of 2, i.e. 2, 4, 8, 16, etc., much like Subnetting. The starting network id should be precisely divisible by the supernet's entire size.

You need to make sure that when you create a supernet, it covers just the networks you want to combine and not more.

You must use routing protocols such as EIGRP, OSPF, and BGP that support classless route ads in order to make the most of Supernetting. With supernetting, a classical routing protocol such as RIPv1 cannot be used effectively.

Why is Supernetting performed?

Supernetting is performed specifically for the routing tables to be optimized.

The summary of all known networks is a routing table.

In order to find a new path and to find the good destination path, routers share routing tables.

Supernetting was used to make the process of routing easy.

Routers can share all routes from routing tables as they are without Supernetting. With Supernetting, before delivering, it will summarize them.

Benefits of Supernetting:

The following benefits are provided by Supernetting.

- It decreases the size of updates for routing.
- It offers a clearer network summary.
- It limits the usage of memory and CPU resources.
- It reduces the time needed to restore the routing tables.
- Regulate and decrease network traffic.
- Helpful in addressing the question of missing IP addresses.

Disadvantages of Supernetting:

The following are the disadvantages of Supernetting.

- When integrated, it does not cover various network areas.
- Every networks must be of the same class and each IPs should be adjacent.

1.12. SUBNETTING Vs SUPERNETTING

The following table shows the differences between Subnetting and Supernetting:

SUBNETTING	SUPERNETTING
The method for breaking the network into sub-networks is Subnetting.	The method of merging the tiny networks is called as Supernetting.
Network addresses bits are increased during subnetting.	In Supernetting, the bits of host addresses are improved.
The mask bits are pushed to the right in the subnetting process.	The mask bits are pushed to the left during the Supernetting process
Subnetting is implemented through subnet masking of variable lengths.	Classless Inter Domain Routing (CIDR) is used to implement Supernetting.
Address depletion is decreased or eliminated during subnetting.	It is mainly used to simplify the process of routing.
Subnets are developed by enlarging the Network prefix.	By reducing the Network prefix, Huge no. of prefixes can be summarized with a single prefix.
It is expensive.	Not much expensive when compared to Subnetting.
Instead of using it across the network, network protection can be easily implemented between subnets.	In the same class, the entire network should exist.

Subnetting Vs Supernetting

Ultimately, both approaches are helpful in improving the availability of IP Addresses and to minimize IP Address depletion.

PROPOSED WORK

Initially, understand what is meant by computer networking and study about various components of networking such as Router, Hub and Switch. Study about topology of networks and the communication model of networking.

Then, understand the concept of Internet Protocol Addressing and various classes of IP Addresses. Then, identify which class a given IP Address belongs to by taking the required conditions. Find about what are private and public Internet Protocol Addresses.

Learn about the Internet Protocol Version-4 Addressing and how it is represented. Find about various types of IPv4 Addresses. Find about what is Internet Protocol Version-6 Addressing and learn about its header structure and extension headers and get a basic idea of transition mechanisms for IPv4 to IPv6 Addressing. Then, find about how IPv6 is different from the traditional IPv4 Addressing.

Then, understand about the concept of Subnetting and the benefits of doing Subnetting. Understand about what is subnet mask, Network Field and Host Field.

Understand about Variable and Fixed Length Subnetting. Then, learn about various types of subnet classes. Find about what is meant by Classless Inter Domain Routing. Then, get basic information about what is meant by Supernetting and its uses.

Analyze the advantages and disadvantages of Supernetting.

Find the various differences between Subnetting and Supernetting.

Ultimately, develop a code for the concept of Subnetting and implement it and get the output of Broadcast number, network number, Possible Host Address Ranges and what class the given IP Address belongs to and so on.

Simulate the concept of Subnetting by taking IP Address of any of the classes using the CISCO Packet Tracer.

CODE SNIPPETS

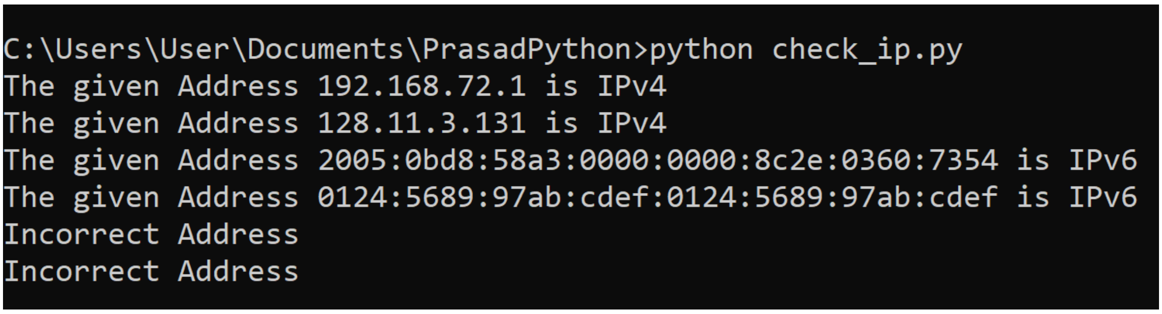
1). Python Code to check whether the given IP Address belongs to type of IPv4 or it belongs to IPv6:

CODE:

```
class Address(object):

    # Check the type of Address
    def check_IP(self, IP_Add):

        # Find whether the Address is IPv4 or not
        def check_IPv4(x):
            try:

                return str(int(x)) == x and 0<=int(x)<=255
            except:

                return False

        # Find whether the Address is IPv6 or not
        def check_IPv6(x):

            if len(x) > 4:
                return False

            try :

                return int(x, 16) >= 0 and x[0] != '-'

            except:

                return False

        if IP_Add.count(".") == 3 and all(check_IPv4(i) for i in IP_Add.split(".")):

            return "The given Address " + IP_Add + " is IPv4"
        if IP_Add.count(":") == 7 and all(check_IPv6(i) for i in IP_Add.split(":")):
            return "The given Address " + IP_Add + " is IPv6"

        return "Incorrect Address"



a = Address()

print(a.check_IP("192.168.72.1"))

print(a.check_IP("128.11.3.131"))

print(a.check_IP("2005:0bd8:58a3:0000:0000:8c2e:0360:7354"))

print(a.check_IP("0124:5689:97ab:cdef:0124:5689:97ab:cdef"))

print(a.check_IP("256.32.553.3"))

print(a.check_IP("256.32:553.3"))
```

Output:

The output of the above program is given below:



```
C:\Users\User\Documents\PrasadPython>python check_ip.py
The given Address 192.168.72.1 is IPv4
The given Address 128.11.3.131 is IPv4
The given Address 2005:0bd8:58a3:0000:0000:8c2e:0360:7354 is IPv6
The given Address 0124:5689:97ab:cdef:0124:5689:97ab:cdef is IPv6
Incorrect Address
Incorrect Address
```

<u>IPv4 or IPv6</u>

2). Python code for Subnetting:

The below code takes the inputs as IP Address and a subnet mask.

It gives the information about number of hosts per subnet , number of mask bits, wildcard mask, Network address, Broadcast address, and Maximum number of subnets, IP Address range and the list of generated IP’s in the given IP Address range.

CODE:

```
from ipaddress import IPv4Address, IPv4Network

import random

import sys

def Int_to_Bin(integer):
    binary = '.'.join([bin(int(x)+256)[3:] for x in integer.split('.')])
    return binary


def subnet_calculator():
    classA = IPv4Network(("10.0.0.0", "255.0.0.0"))
    classB = IPv4Network(("172.16.0.0", "255.240.0.0"))
    classC = IPv4Network(("192.168.0.0", "255.255.0.0"))
```

```
try:
    while True:
        enter_ip = input("\nEnter the IP address: ")

        octet_ip = enter_ip.split(".")

        int_oct_ip1 = [int(i) for i in octet_ip]

        if int(octet_ip[0])>=1 and int(octet_ip[0])<=126:
            print("Class A")
        elif int(octet_ip[0])>=128 and int(octet_ip[0])<=191:
            print("Class B")
        elif int(octet_ip[0])>=192 and int(octet_ip[0])<=223:
            print("Class C")
        elif int(octet_ip[0])>=224 and int(octet_ip[0])<=239:
            print("Class D")
        else:
            print("Class E")



        if (len(int_oct_ip1) == 4) and (int_oct_ip1[0] != 127) and (int_oct_ip1[0] != 169) \
           and (0 <= int_oct_ip1[1] <= 255) and \
           (0 <= int_oct_ip1[2] <=255) and \
           (0 <= int_oct_ip1[3] <= 255):
            break
        else:
```

```
        print("Wrong IP, retry \n")
        continue

# Enter Mask Inputs
masks = [0, 128, 192, 224, 240, 248, 252, 254, 255]

while True:
    # subnet-mask as input
    input_subnet = input("\nEnter the Subnet Mask: ")

    # Verify the subnet mask
    oct_sub_net = [int(j) for j in input_subnet.split(".")]

    # display oct_sub_net
    if (len(oct_sub_net) == 4) and \
            (oct_sub_net[0] == 255) and \
            (oct_sub_net[1] in masks) and \
            (oct_sub_net[2] in masks) and \
            (oct_sub_net[3] in masks) and \
            (oct_sub_net[0] >= oct_sub_net[1] >= oct_sub_net[2] >= oct_sub_net[3]):
        break
    # Otherwise
    else:
        print("Wrong subnet mask, retry\n")
        continue
        # Changing IP and subnet to binary
        ip_in_binary = []
        # Change each IP octet to binary
        ip_in_binary_octets = [bin(i).split("b")[1] for i in int_oct_ip1]
        # Convert each binary octet of 8 bit length by adding zeros
        for i in range(0,len(ip_in_binary_octets)):
            if len(ip_in_binary_octets[i]) < 8:
                # Find the added bin
                padded_bin = ip_in_binary_octets[i].zfill(8)
                # Append the padded bin to binary IP
                ip_in_binary.append(padded_bin)
            #  If length is greater than 8
            else:
                ip_in_binary.append(ip_in_binary_octets[i])

# merge the binary octets
        ip_bin_mask = "".join(ip_in_binary)
        # display ip_bin_mask
        sub_in_binary = []
        # convert each subnet octet to binary
        sub_binary_octet = [bin(i).split("b")[1] for i in oct_sub_net]



# make each binary octet of 8 bit length by adding zeros
        for i in sub_binary_octet:
            if len(i) < 8:
```

```
        sub_added = i.zfill(8)

        sub_in_binary.append(sub_added)

    # If length is greater than 8
    else:
        sub_in_binary.append(i)

# display sub_in_binary
sub_binary_mask = "".join(sub_in_binary)

# Compute number of hosts
nil_zeros = sub_binary_mask.count("0")
nil_ones = 32 - nil_zeros
nil_hosts = abs(2 ** nil_zeros - 2)

# Computing Wildcard Mask
wild_mask = []

for i in oct_sub_net:
    wild_bit = 255 - i
    # Compute wild_mask
    wild_mask.append(wild_bit)
wildcard = ".".join([str(i) for i in wild_mask])

# Computing the Network and Broadcast Address
network_add_binary = ip_bin_mask[:nil_ones] + "0" * nil_zeros
```

```
    broadcast_add_bin = ip_bin_mask[:nil_ones] + "1" * nil_zeros
    network_add_binary_octet = []

    broadcast_binoct = []

    [network_add_binary_octet.append(i) for i in [network_add_binary[j:j+8] \
                       for j in range(0, len(network_add_binary), 8)]]

    [broadcast_binoct.append(i) for i in [broadcast_add_bin[j:j+8] \
                   for j in range(0,len(broadcast_add_bin),8)]]

    network_add_dec_final = ".".join([str(int(i,2)) for i in network_add_binary_octet])
    broadcast_add_dec_final = ".".join([str(int(i,2)) for i in broadcast_binoct])

    # Compute the host IP Range
    begin_ip_host = network_add_binary_octet[0:3] + \
    [(bin(int(network_add_binary_octet[3],2)+1).split("b")[1].zfill(8))]

    first_ip = ".".join([str(int(i,2)) for i in begin_ip_host])
    final_ip_host = broadcast_binoct[0:3] + \
[bin(int(broadcast_binoct[3],2)-1).split("b")[1].zfill(8)]
    last_ip = ".".join([str(int(i,2)) for i in final_ip_host])
    # display the input IP Address
    print("\nThe entered ip address is: " + enter_ip + "-->" + Int_to_Bin(enter_ip))
    # display the input subnet mask
    print("The entered subnet mask is: " + input_subnet + "-->" + \
Int_to_Bin(input_subnet))
```

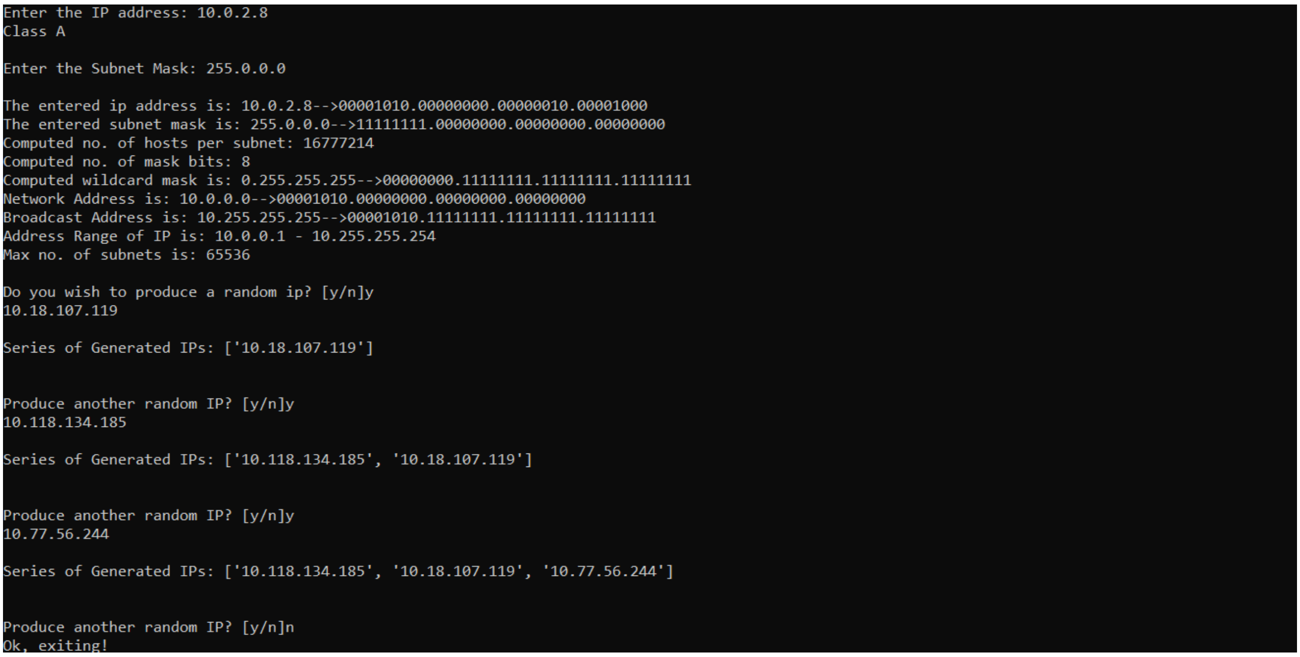
```
        # display the number of hosts per subnet
        print("Computed no. of hosts per subnet: {0}".format(str(nil_hosts)))
        # display the number of mask bits
        print("Computed no. of mask bits: {0}".format(str(nil_ones)))


        # display the wildcard mask
        print("Computed wildcard mask is: {0}".format(wildcard) + "-->" + \
Int_to_Bin(wildcard))
        # display the network address
        print("Network Address is: {0}".format(network_add_dec_final) + "-->" + \
Int_to_Bin(network_add_dec_final))


        # display the broadcast address
        print("Broadcast Address is: {0}".format(broadcast_add_dec_final) + "-->"+ \
Int_to_Bin(broadcast_add_dec_final))
        # display the IP Address range
        print("Address Range of IP is: {0} - {1}".format(first_ip, last_ip))
        # display maximum number of subnets
        print("Max no. of subnets is: " + str(2**abs(24 - nil_ones)))


        series_ip = []
        print("")
        # ask to generate a random ip in the range
        if input("Do you wish to produce a random ip? [y/n]") == 'y':
            while True:
                arbitrary_ip = []
                # Verify if the octet bit is equal in first and last host address
```

```
    # If equal, add. or generate random IP
    for i in range(0,len(begin_ip_host)):
        for j in range(0,len(final_ip_host)):
            if i == j:
                if begin_ip_host[i] == final_ip_host[j]:

                    arbitrary_ip.append(int(begin_ip_host[i],2))
                else:
                    arbitrary_ip.append(random.randint(int(begin_ip_host[i],2), \
int(final_ip_host[j],2)))

    arbitrary_ip_final = ".".join(str(i) for i in arbitrary_ip)

    # Check if generated IP has already been printed. If so, compute again till
#unique IP is obtained
    if arbitrary_ip_final in series_ip:
        # If all IPs in the host range are used, exit
        if len(series_ip) == nil_hosts:
            print("All IPs in the range consumed, leaving\n")
            break
        continue

    # If length of series IP not equal to nil_hosts
    else:
        print(arbitrary_ip_final + '\n')
    series_ip.append(arbitrary_ip_final)
    print("Series of Generated IPs:" , sorted(series_ip) ,'\n')
```

```
            if input("\nProduce another random IP? [y/n]") == 'y':
                continue
            # If you don’t want to produce random IP
            else:
                print("Ok, exiting!")
                break



    except KeyboardInterrupt:
        print("Stopped by User, leaving\n")
    except ValueError:
        print("Appears to have typed an wrong value, leaving\n")


# Calling the function
if __name__ == '__main__':
    subnet_calculator()
# End of Code
```

Output:

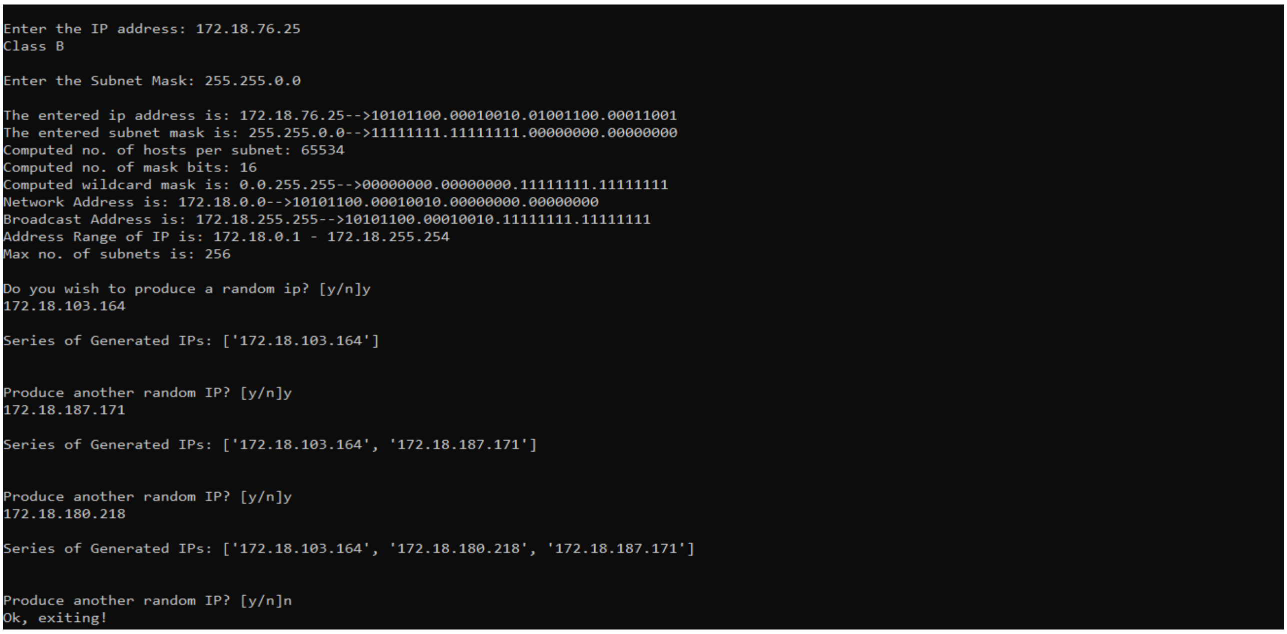
The output for the class A IP Address is given below:



```
Enter the IP address: 10.0.2.8
Class A

Enter the Subnet Mask: 255.0.0.0

The entered ip address is: 10.0.2.8-->00001010.00000000.00000010.00001000
The entered subnet mask is: 255.0.0.0-->11111111.00000000.00000000.00000000
Computed no. of hosts per subnet: 16777214
Computed no. of mask bits: 8
Computed wildcard mask is: 0.255.255.255-->00000000.11111111.11111111.11111111
Network Address is: 10.0.0.0-->00001010.00000000.00000000.00000000
Broadcast Address is: 10.255.255.255-->00001010.11111111.11111111.11111111
Address Range of IP is: 10.0.0.1 - 10.255.255.254
Max no. of subnets is: 65536

Do you wish to produce a random ip? [y/n]y
10.18.107.119

Series of Generated IPs: ['10.18.107.119']


Produce another random IP? [y/n]y
10.118.134.185

Series of Generated IPs: ['10.118.134.185', '10.18.107.119']


Produce another random IP? [y/n]y
10.77.56.244

Series of Generated IPs: ['10.118.134.185', '10.18.107.119', '10.77.56.244']


Produce another random IP? [y/n]n
Ok, exiting!
```

Output for Class-A Address

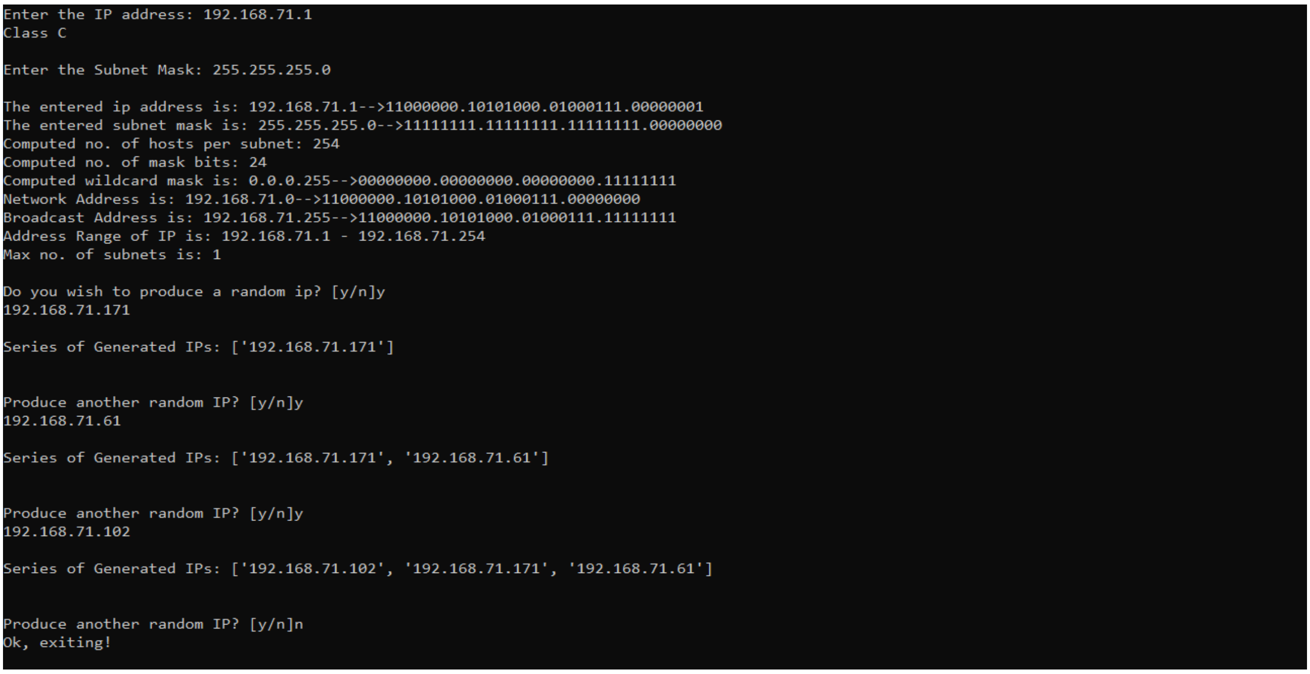
The output for the class B IP Address is given below:



```
Enter the IP address: 172.18.76.25
Class B

Enter the Subnet Mask: 255.255.0.0

The entered ip address is: 172.18.76.25-->10101100.00010010.01001100.00011001
The entered subnet mask is: 255.255.0.0-->11111111.11111111.00000000.00000000
Computed no. of hosts per subnet: 65534
Computed no. of mask bits: 16
Computed wildcard mask is: 0.0.255.255-->00000000.00000000.11111111.11111111
Network Address is: 172.18.0.0-->10101100.00010010.00000000.00000000
Broadcast Address is: 172.18.255.255-->10101100.00010010.11111111.11111111
Address Range of IP is: 172.18.0.1 - 172.18.255.254
Max no. of subnets is: 256

Do you wish to produce a random ip? [y/n]y
172.18.103.164

Series of Generated IPs: ['172.18.103.164']


Produce another random IP? [y/n]y
172.18.187.171

Series of Generated IPs: ['172.18.103.164', '172.18.187.171']


Produce another random IP? [y/n]y
172.18.180.218

Series of Generated IPs: ['172.18.103.164', '172.18.180.218', '172.18.187.171']


Produce another random IP? [y/n]n
Ok, exiting!
```

Output for Class-B Address

The output for the class C IP Address is given below:



```
Enter the IP address: 192.168.71.1
Class C

Enter the Subnet Mask: 255.255.255.0

The entered ip address is: 192.168.71.1-->11000000.10101000.01000111.00000001
The entered subnet mask is: 255.255.255.0-->11111111.11111111.11111111.00000000
Computed no. of hosts per subnet: 254
Computed no. of mask bits: 24
Computed wildcard mask is: 0.0.0.255-->00000000.00000000.00000000.11111111
Network Address is: 192.168.71.0-->11000000.10101000.01000111.00000000
Broadcast Address is: 192.168.71.255-->11000000.10101000.01000111.11111111
Address Range of IP is: 192.168.71.1 - 192.168.71.254
Max no. of subnets is: 1

Do you wish to produce a random ip? [y/n]y
192.168.71.171

Series of Generated IPs: ['192.168.71.171']


Produce another random IP? [y/n]y
192.168.71.61

Series of Generated IPs: ['192.168.71.171', '192.168.71.61']


Produce another random IP? [y/n]y
192.168.71.102

Series of Generated IPs: ['192.168.71.102', '192.168.71.171', '192.168.71.61']


Produce another random IP? [y/n]n
Ok, exiting!
```

Output for Class-C Address

Simulation

Let us consider a class C Network with Network Id say 192.168.1.0 and with subnet mask of 255.255.255.0

After Subnetting, suppose 3 bits are taken from Host Id, then the Network Id has 27 bits.

(24 + 3 = 27). Where; 24 are fixed network Id bits with the class C.

The subnet mask becomes 255.255.255.224 (224 = 11100000).

When three bits are taken, No. of networks that become possible will be 2^3 = 8.

Here, first 000 sub network is generally not used. Since, it denotes Network Id.

Second sub network is 001 i.e. 00100000(8-bit). Decimal = 32. It is also not used because there are all zeros in the host Id part.

So, IP Address will start from 192.168.1.33. In each subnet network 32 (2^5) hosts are possible.

First and last hosts are not used because they represent the network Id and Broadcast Id respectively.

So, first network ranges from 192.168.1.33 to 192.168.1.63.

192.168.1.64 represents network Id for second network.

So, second network starts at 192.168.1.65.

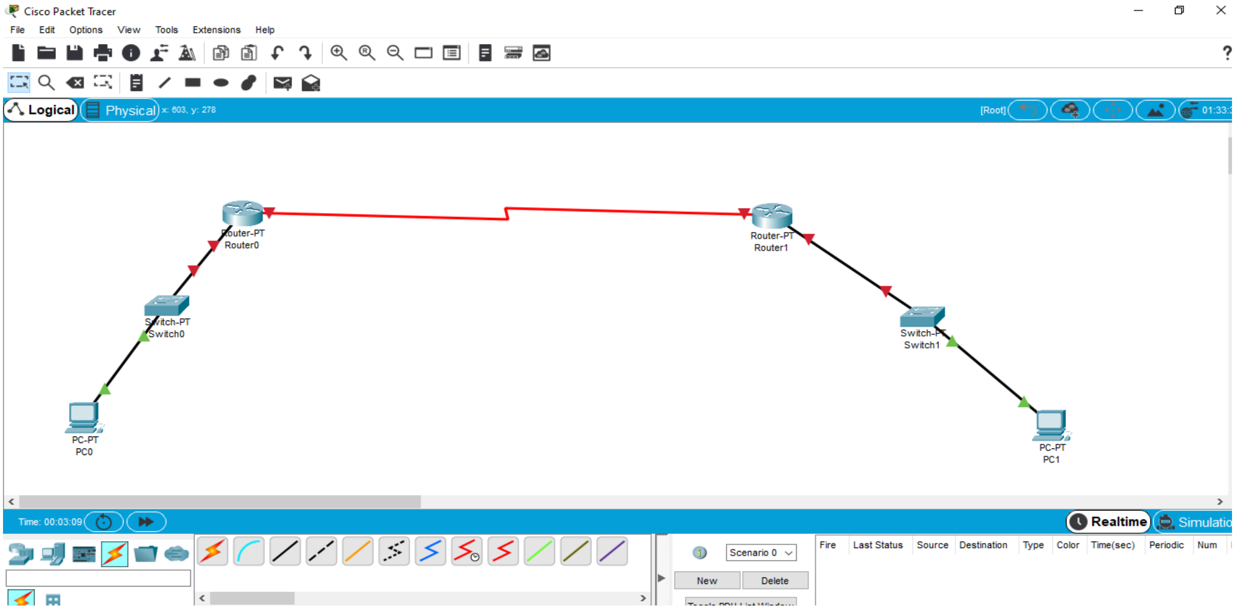
Using the CISCO Packet Tracer, we do the simulation and set up the network structure correctly with the help of Subnetting concept.

We will connect two networks with the help of third network (Between Routers).

Simulation Process:

Initially, the two networks each have a router, switch and a PC. Then, routers of both the networks will be connected.

It is shown as below:



<u>Networks without Subnetting</u>

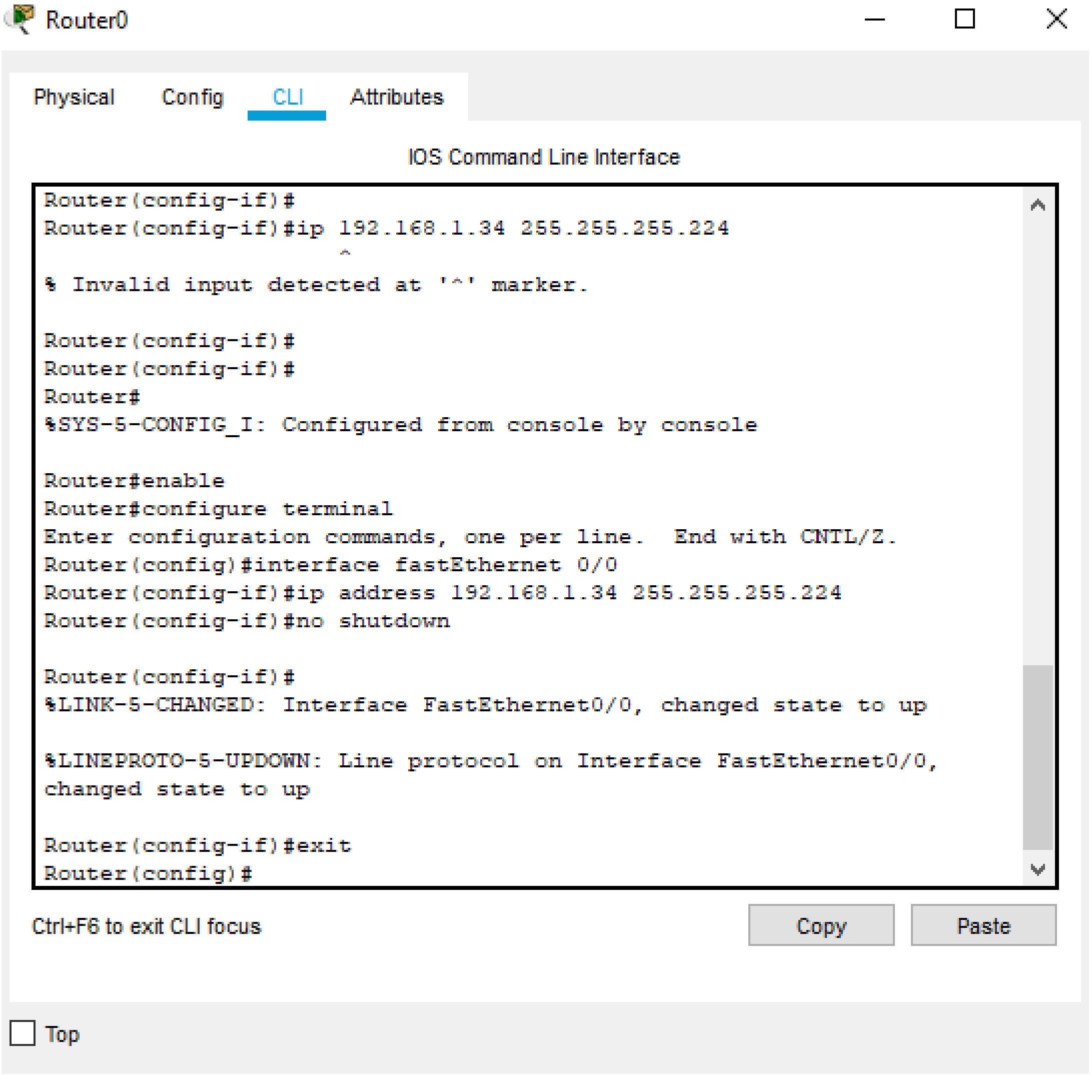
For PC0, we assign the IP Address as 192.168.1.33, subnet mask as 255.255.255.224(default).

Default gateway is the IP Address of the router i.e. 192.168.1.34. We configured our PC0.

Router can be configured in two ways:

1. Directly using config option.
2. Using CLI.

First, we configure Router0 using CLI. It is given below:



```
Router(config-if)#
Router(config-if)#ip 192.168.1.34 255.255.255.224
                    ^
% Invalid input detected at '^' marker.

Router(config-if)#
Router(config-if)#
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 192.168.1.34 255.255.255.224
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up

Router(config-if)#exit
Router(config)#
```

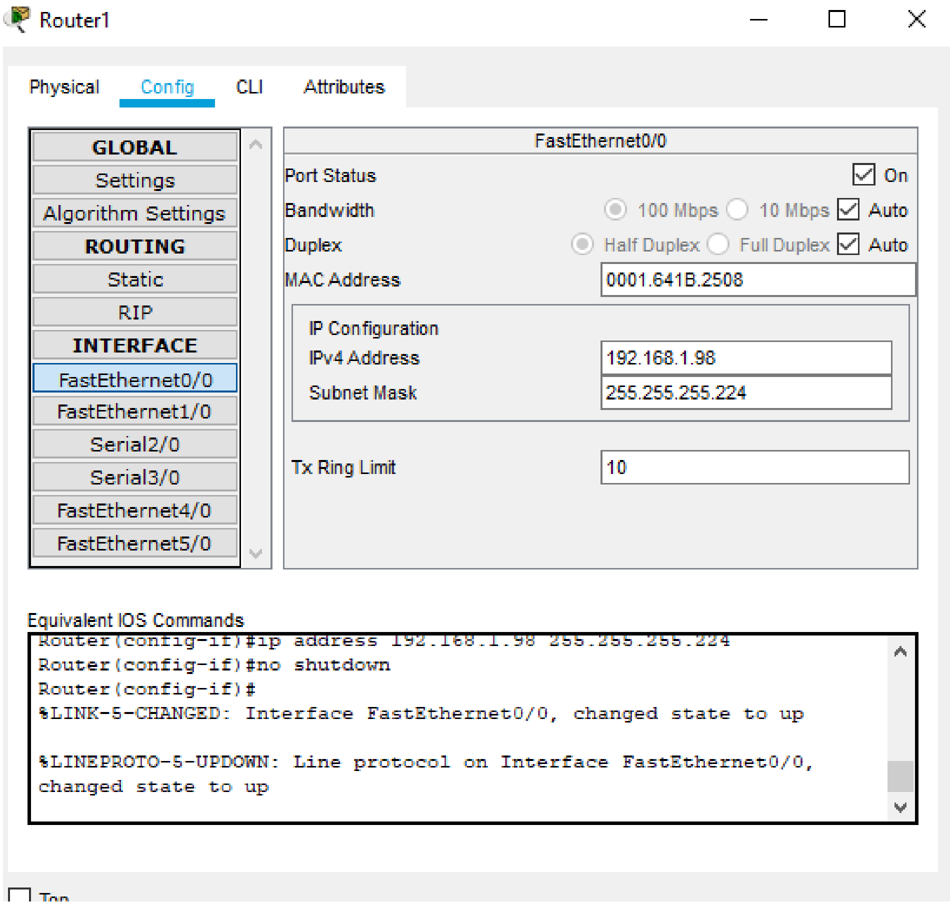
Configuring Router0

Next, For PC1 we assign IP Address as 192.168.1.97, subnet mask as 255.255.255.224 and default gateway as 192.168.1.98.

So, PC1 gets configured.

Then, we configure the Router1 using Config option.

It is given below:



FastEthernet0/0

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ● Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0001.641B.2508

IP Configuration

IPv4 Address 192.168.1.98

Subnet Mask 255.255.255.224

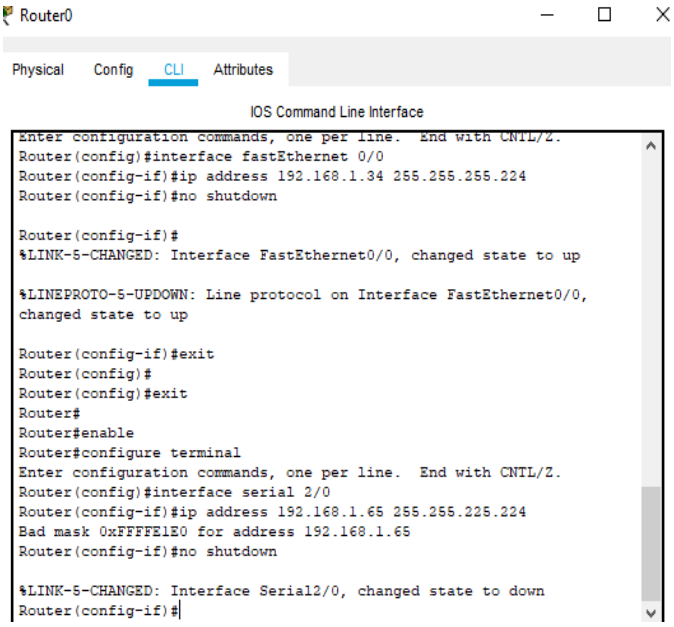
Tx Ring Limit 10

Equivalent IOS Commands

```
Router(config-if)#ip address 192.168.1.98 255.255.255.224
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
```

Configuring Router1

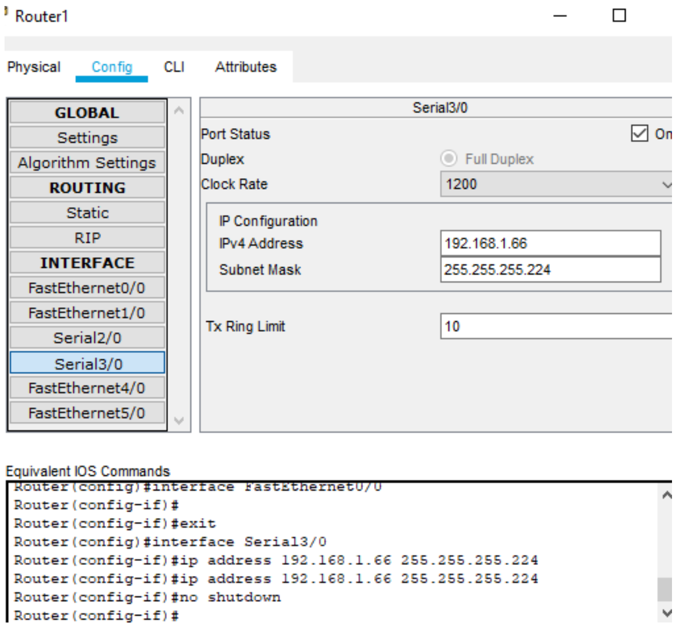
Now, configure the serial ports through CLI and Config options.



IOS Command Line Interface

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 192.168.1.34 255.255.255.224
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up

Router(config-if)#exit
Router(config)#
Router(config)#exit
Router#
Router#enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 2/0
Router(config-if)#ip address 192.168.1.65 255.255.225.224
Bad mask 0xFFFFE1E0 for address 192.168.1.65
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
```

Configuring Router0 Serial Port



Serial3/0

Port Status ☑ On

Duplex ○ Full Duplex

Clock Rate 1200

IP Configuration

IPv4 Address 192.168.1.66

Subnet Mask 255.255.255.224

Tx Ring Limit 10

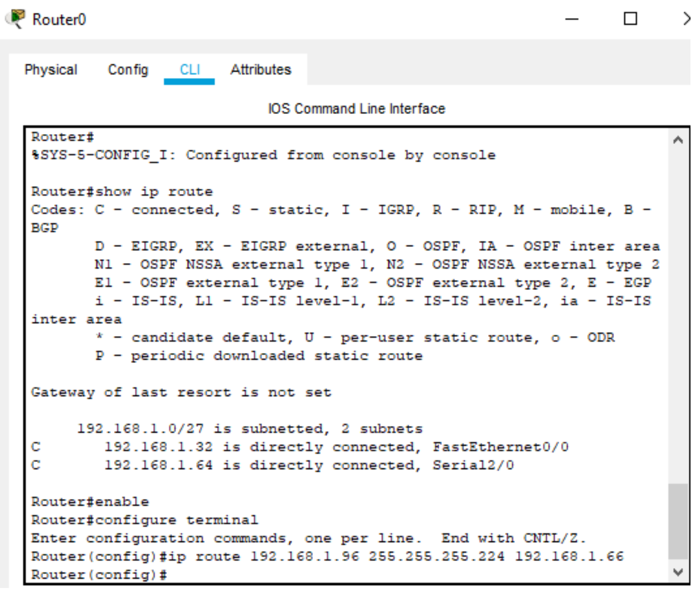
Equivalent IOS Commands

```
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 192.168.1.66 255.255.255.224
Router(config-if)#ip address 192.168.1.66 255.255.255.224
Router(config-if)#no shutdown
Router(config-if)#
```

Configuring Router1 Serial Port

The, we check by sending packets whether they are successfully sent or not. We can’t send the packets from PC0 to PC1.

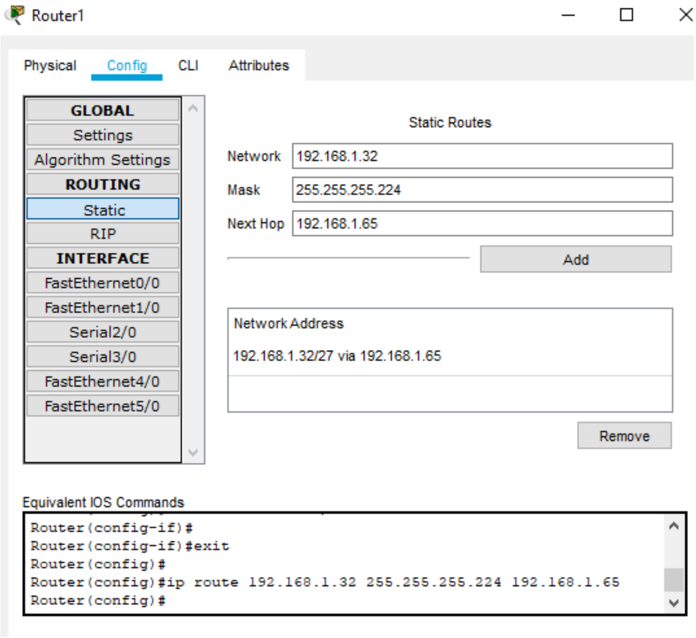
Router0 does not know about 192.168.1.96 network. Since, it is not configured yet. We can resolve this by:



```
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.1.0/27 is subnetted, 2 subnets
C       192.168.1.32 is directly connected, FastEthernet0/0
C       192.168.1.64 is directly connected, Serial2/0

Router#enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 192.168.1.96 255.255.255.224 192.168.1.66
Router(config)#
```

Configuring Router0 IP route

Similarly, configure Router1. We configure the Router1 through Config Option instead of CLI. Here, we need select the static menu and we need to input the Network, Mask and the Next Hop of the Router. Then, Click on the Add Button. We see that the IP route has been set up. Then, go to the setting and click on save option to save the inputs.



Network 192.168.1.32

Mask 255.255.255.224

Next Hop 192.168.1.65

Network Address

192.168.1.32/27 via 192.168.1.65

Equivalent IOS Commands

```
Router(config-if)#
Router(config-if)#exit
Router(config)#
Router(config)#ip route 192.168.1.32 255.255.255.224 192.168.1.65
Router(config)#
```

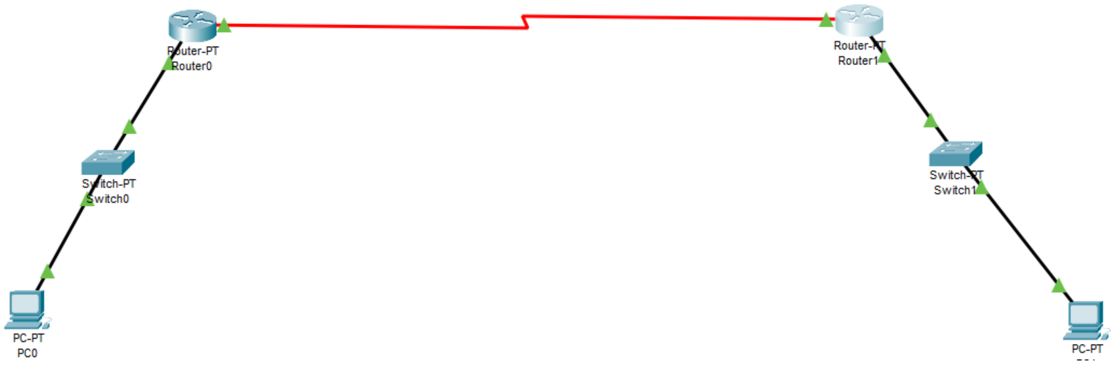
Configuring Router1 IP route

The Next Hop is the IP Address of router to which we are connecting the respective router.

We can verify whether the packets are receiving or not.

The final output after configuring all the networks is given below:

Final Output:



Networks after Subnetting

Finally, we have done the simulation for the given IP Address and checked the output.

All the networks are working properly.

RESULTS

First program is used to find the type of IP Address, whether it is IPv4 or IPv6.

We enter the IP Address in the format of string.

IPv4 address is given in the format of decimal dotted format. IPv4 Address consists of four groups.

We convert the first part of the IP Address to the integer format and check whether it exists in between 0 to 255.

Otherwise, return False and check whether the given address is IPv6 or not.

IPv6 Address is given in the format of hexadecimal format. It consists of eight groups.

First, we check whether the first part of the address length is greater than four or not. If it is greater than four, return False.

Then we check the hexadecimal format to determine whether the address is of type IPv6 or not. If it is IPv6, then print the output as the address is IPv6.

If the address is not either IPv4 or IPv6, then print the output as the address is not either IPv4 or IPv6.

Second program is used to show the concept of Subnetting.

Here, first we identified the class of IP Address whether it is class A, class B or class C.

Then, we enter the respective subnet mask for that class and we print the given IP and Subnet mask in binary format. Then, we find the first starting address of that IP Address range. It becomes the host address.

The last address of that IP Address range becomes the Broadcast Address for that IP Address.

By subtracting the subnet mask of the given IP Address from 255.255.255.255, we get the wildcard address for the given Internet Protocol Address.

We then found the number of hosts per subnet, maximum subnets.

Also, we can display the list of IP Addresses in the given IP Address Rang one by one.

Finally, we simulated Subnetting concept with the help of CISCO Packet Tracer.

CONCLUSION AND FUTURE PLANS

Although a 32-bit IP address provides an incredibly large address numbers. The class structures A, B and C use Host ID and Network ID assignments effectively.

Subnetting enhances the scenario by allowing stronger network-host split assignments, thus enhancing efficiency and large-scale network maintainability.

We are able to use Subnetting and Supernetting to reduce the cost and to protect our network.

In this paper, a major problem that has been overlooked is security. We consider that every other node trusts each another node. But, it is not the problem and the following circumstances can arise:

- For nodes which do not exist, a node demands IP addresses.
- In this manner, a node will obtain all the IP addresses that deny membership in the network to others.
- Without following the protocol given, a node assigns IP addresses to other nodes.
- This may contribute to problems with IP addresses that may be hard to fix.
- A node chosen gives other nodes the wrong data.
- The process of synchronization in our protocol depends on reliable broadcasting.
- Since, not any such broadcast happen in a distributed mobile world, one can inquire questioning the protocol strength.

IPv4 to IPv6 migration is often compared to Y2K Crisis, which needs time and resource investment. Companies have yet to identify the exhaustion of IPv4 numbers as a troubling problem, and are not prepared to suspend the necessary investment in the future.

The possibility of inadequate time and expense will be present in the future. The cost of IPv6 migration may be an issue.

The costs present involve numbering networks again and concurrently running two protocol stacks such as IPv4 and IPv6, updating to suitable software and hardware, training personnel, and testing implementations of networks.

IPv6, however, provides major advantages and features required by the modern, stable internet. The migration method could be the only feasible solution in the long run, given the number of problems in the current network.

Finally, there are many advantages of using IP Address Subnetting and Supernetting. There may be few issues such as protection of network and system which must be looked into in the future.

References

[1]. About 127.0.0.1: http://www.tech-faq.com/127-0-0-1.html

[2]. Subnetting-Techopedia: https://www.techopedia.com/definition/28328/subnetting

[3]. Private and Public IP Addresses: https://help.keenetic.com/hc/en-us/articles/213965789-What-is-the-difference-between-a-public-and-private-IP-address-

[4]. Private and Public IP Differences: https://www.gohacking.com/private-and-public-ip-addresses/

[5]. Differences between IPV4 & IPV6,By Ali, AmerNizar Abu: https://www.ijcsi.org/papers/IJCSI-9-3-1-314-317.pdf

[6]. Introduction to IPv4: https://en.wikipedia.org/wiki/IPv4#:~:text=Internet%20Protocol%20version%204%20(IPv4,the%20ARPANET%20in%20January%201983.

[7]. Introduction to IPv6: https://en.wikipedia.org/wiki/IPv6

[8]. Intro to Subnetting: https://www.pluralsight.com/blog/it-ops/simplify-routing-how-to-organize-your-network-into-smaller-subnets

[9]. Subnetting Vs Supernetting: https://vivadifferences.com/5-difference-between-subnetting-and-supernetting/

[10]. IP Address - Subnetting and Supernetting:

https://www.ijettcs.org/Volume4Issue5(2)/IJETTCS-2015-10-20-38.pdf