AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
(IOIT)
ADDING VALUE TO ENGINEERING
An Autonomous Institute Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognised by Govt. of Maharashtra
Accredited by NAAC with "A+" Grade | NBA - 5 UG Programmes

All India Shri Shivaji Memorial Society's

Institute of Information Technology Pune

Department of Computer Engineering

LABORATORY MANUAL

COMPUTER NETWORK LABORATORY

SY BTech Computer Engineering

SEMESTER-II

Subject Code: COPCC408

TEACHING SCHEME	**EXAMINATION**
SCHEME	
Lectures: 3 Hrs/Week	Theory:60 Marks
Practical: 2 Hrs/Week	In sem:40 Marks
	Practical:50 Marks

INDEX

Sr. No	Title	Page No	Date of Conduction	Date of Submission	Sign
1.	Bridge the gap (Basic Network commands)				
2.	Setup a wired LAN using Layer 2 Switch. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility				
3.	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.				
4.	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.				
5.	Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.				
6.	Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.				
7.	Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP				
8.	Write a program using TCP socket for wired network for following a. Say hello to each other b. File transfer c. Calculator **OR** Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.				
9.	Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.				
10.	Content Beyond Syllabus: Windows File sharing				

Assignment No.	1
Title	**Bridge the gap (Basic Network commands)**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 01

TITLE: Bridge the gap (Basic Network commands)

OBJECTIVES: To get familiar with different network commands

PROBLEM STATEMENT: Execute different network commands link ping, ipconfig, NetStat, ARP, Hostname, Tracert

OUTCOME: Students will be able to,

1. illustrate various network commands.

THEORY-CONCEPT:

The operating system consists of various built-in, command-line networking utilities that are used for network troubleshooting. We will see various networking commands which are most essentials for every network administrator.

These commands are as follow-

1. Ping : Ping is used to testing a network host capacity to interact with another host. Just enter the command Ping, followed by the target host's name or IP address. The ping utilities seem to be the most common network tool. For Example: If an Internet connection is not in the office, for instance, the ping utility is used to determine if the problem exists in the office or the Internet provider's network. The following shows an image of how ping tools to obtain the locally connected router's connectivity status.

2. **NetStat :** Netstat is a Common TCP – IP networking command-line method present in most Windows, Linux, UNIX, and other operating systems. The netstat provides the statistics and information in the use of the current TCP-IP Connection network about the protocol. There are various options a user can use with the Netstat command.

Options are as follows-

- -a: This will display all connection and ports
- -b: Shows the executable involved in each connection or hearing port
- -e: This protocol will combine with the -sand display the ethernet statistics
- -n: This will display the address and the port number in the form of numerical
- -o: It will display the ID of each connection for the ownership process.
- -r: It will display the routing table
- -v: When used in combination with -b, the link or hearing port sequence for every executable is

shown.

3. Ip Config: The command IP config will display basic details about the device's IP address configuration. Just type IP config in the Windows prompt and the IP, subnet mask and default gateway that the current device will be presented. If you have to see full information, then type on command prompt config-all and then you will see full information. There are also choices to assist you in resolving DNS and DHCP issues.

4. Hostname : To communicate with each and other, the computer needs a unique address. A hostname can be alphabetic or alphanumeric and contain specific symbols used specifically to define a specific node or device in the network. For example, a hostname should have a domain name (TLD) of the top-level and a distance between one and 63 characters when used in a domain name system (DNS) or on the Internet.

5. Tracert: The tracert command is a Command Prompt command which is used to get the network packet being sent and received and the number of hops required for that packet to reach to target. This command can also be referred to as a traceroute. It provides several details about the path that a packet takes from the source to the specified destination. The tracert command is available for the Command Prompt in all Windows operating systems. The syntax for Tracert Command

tracert [-d] [-h MaxHops] [-w TimeOut] target

6. Lookup: The Nslookup, which stands for name server lookup command, is a network utility command used to obtain information about internet servers. It provides name server information for the DNS (Domain Name System), i.e., the default DNS server's name and IP Address. *The syntax for Nslookup is as follows.*

Nslookup or

Nslookup [domain_name]

7. ARP : ARP stands for Address Resolution Protocol. Although network communications can readily be thought of as an IP address, the packet delivery depends ultimately on the media access control (MAC). This is where the protocol for address resolution comes into effect. You can add the remote host IP address, which is an arp -a command, in case you have issues to communicate with a given host. The ARP command provides information like Address, Flags, Mask, IFace, Hardware Type, Hardware Address, etc.

CONCLUSION:

In this experiment we have learned how to execute different network commands.

Assignment No.	2
Title	**Setup and Test LAN Network**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 02

Title: Setup a wired LAN

PROBLEM STATEMENT

Setup a wired LAN using Layer 2 Switch. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility

OBJECTIVES:

1. To understand the structure and working of various networks including the interconnecting devices used in them.
2. To get hands on experience of making and testing cables

TYPES OF NETWORK

Common examples of area network types

are: LAN - Local Area Network

WLAN - Wireless Local Area Network

WAN - Wide Area Network

MAN - Metropolitan Area Network

SAN - Storage Area Network, System Area Network, Server Area Network, or sometimes Small Area Network

CAN - Campus Area Network, Controller Area Network, or sometimes Cluster Area Network

PAN - Personal Area Network

DAN - Desk Area Network

LAN - Local Area Network

A LAN connects network devices over a relatively short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs (perhaps one per room), and occasionally a LAN will span a group of nearby buildings.

MAN-Metropolitan Area Network

A network spanning a physical area larger than a LAN but smaller than a WAN, such as a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.

WAN - Wide Area Network

As the term implies, a WAN spans a large physical distance. The Internet is the largest WAN, panning the Earth. A WAN is a geographically-dispersed collection of LANs. A network device

called a router connects LANs to a WAN. In IP networking, the router maintains both a LAN address and a WAN address.

TYPES OF CABLES

Cable is the medium through which information usually moves from one network device to another. There are several types of cable which are commonly used with LANs. In some cases, a network will utilize only one type of cable, other networks will use a variety of cable types. The type of cable chosen for a network is related to the network's topology, protocol, and size. Understanding the characteristics of different types of cable and how they relate to other aspects of a network is necessary for the development of a successful network.

The following sections discuss the types of cables used in networks and other related topics.

- Unshielded Twisted Pair (UTP) Cable
- Shielded Twisted Pair (STP) Cable
- Coaxial Cable
- Fiber Optic Cable
- Cable Installation Guides
- Wireless LANs

Unshielded Twisted Pair (UTP) Cable

Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks

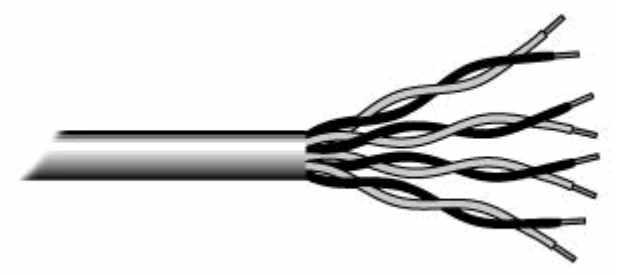


Fig.1. Unshielded twisted pair

The quality of UTP may vary from telephone-grade wire to extremely high-speed cable. The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other electrical devices. The tighter the twisting, the higher the supported transmission rate and

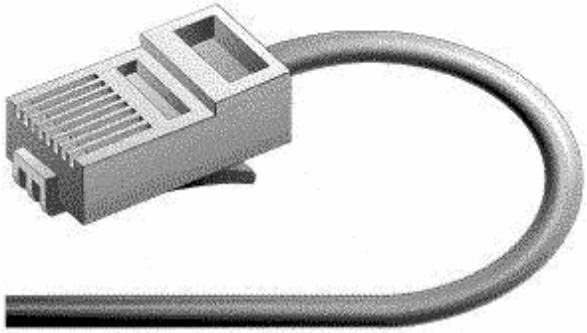
the greater the cost per foot. The EIA/TIA (Electronic Industry Association/Telecommunication Industry Association) has established standards of UTP and rated six categories of wire (additional categories are emerging).

Category	Speed	Use
1	1 Mbps	Voice Only (Telephone Wire)
2	4 Mbps	LocalTalk & Telephone (Rarely used)
3	16 Mbps	10BaseT Ethernet
4	20 Mbps	Token Ring (Rarely used)
	100 Mbps (2 pair)	100BaseT Ethernet
5	1000 Mbps (4 pair)	Gigabit Ethernet
5e	1,000 Mbps	Gigabit Ethernet
6	10,000 Mbps	Gigabit Ethernet

Unshielded Twisted Pair Connector

The standard connector for unshielded twisted pair cabling is an RJ-45 connector. This is a plastic connector that looks like a large telephone-style connector (See fig. 2). A slot allows the RJ-45 to be inserted only one way. RJ stands for Registered Jack, implying that the connector follows a standard borrowed from the telephone industry. This standard designates which wire goes with each pin inside the connector.

Fig. 2. RJ-45 connector



Shielded Twisted Pair (STP) Cable

Although UTP cable is the least expensive cable, it may be susceptible to radio and electrical frequency interference (it should not be too close to electric motors, fluorescent lights, etc.). If you must place cable in environments with lots of potential interference, or if you must place cable in extremely sensitive environments that may be susceptible to the electrical current in the UTP, shielded

twisted pair may be the solution. Shielded cables can also help to extend the maximum distance of the cables.

Shielded twisted pair cable is available in three different configurations:

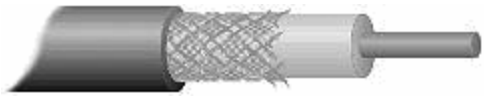
1. Each pair of wires is individually shielded with foil.
2. There is a foil or braid shield inside the jacket covering all wires (as a group).

3. There is a shield around each individual pair, as well as around the entire group of wires (referred to as double shield twisted pair).

Coaxial Cable

Coaxial cabling has a single copper conductor at its center. A plastic layer provides insulation between the center conductor and a braided metal shield (See fig. 3). The metal shield helps to block any outside interference.

Fig. 3. Coaxial cable



Although coaxial cabling is difficult to install, it is highly resistant to signal interference. In addition, it can support greater cable lengths between network devices than twisted pair cable. The two types of coaxial

Thin coaxial cable is also referred to as thinnet. 10Base2 refers to the specifications for thin coaxial cable carrying Ethernet signals. The 2 refers to the approximate maximum segment length being 200 meters. In actual fact the maximum segment length is 185 meters. Thin coaxial cable has been popular in Thick coaxial cable is also referred to as thicknet. 10Base5 refers to the specifications for thick coaxial cable carrying Ethernet signals. The 5 refers to the maximum segment length being 500 meters. Thick coaxial cable has an extra protective plastic cover that helps keep moisture away from the center conductor. This makes thick coaxial a great choice when running longer lengths in a linear bus network. One disadvantage of thick coaxial is that it does.

Coaxial Cable Connectors

The most common type of connector used with coaxial cables is the Bayone-Neill-Concelman (BNC) connector (See fig. 4). Different types of adapters are available for BNC connectors, including a T-connector, barrel connector, and terminator. Connectors on the cable are the weakest points in any network. To help avoid problems with your network, always use the BNC connectors that crimp.

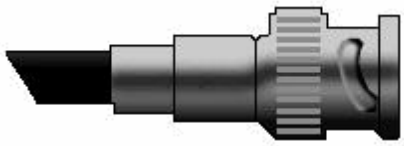


Fig. 4. BNC connector

Fiber Optic Cable

Fiber optic cabling consists of a center glass core surrounded by several layers of protective materials (See fig. 5). It transmits light rather than electronic signals eliminating the problem of electrical interference. This makes it ideal for certain environments that contain a large amount of electrical interference. It has also made it the standard for connecting networks between

Fiber optic cable has the ability to transmit signals over much longer distances than coaxial and twisted pair. It also has the capability to carry information at vastly greater speeds. This capacity broadens communication possibilities to include services such as video conferencing and interactive services. The cost of fiber optic cabling is comparable to copper cabling; however, it is

The center core of fiber cables is made from glass or plastic fibers (see fig 5). A plastic coating then cushions the fiber center, and kevlar fibers help to strengthen the cables and prevent breakage. The outer insulating jacket made of teflon or PVC.

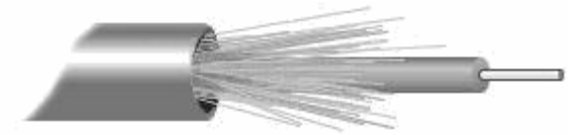


Fig. 5. Fiber optic cable

There are two common types of fiber cables -- single mode and multimode. Multimode cable has a larger diameter; however, both cables provide high bandwidth at high speeds. Single mode can provide more distance, but it is more expensive.

NETWORK TOPOLOGY

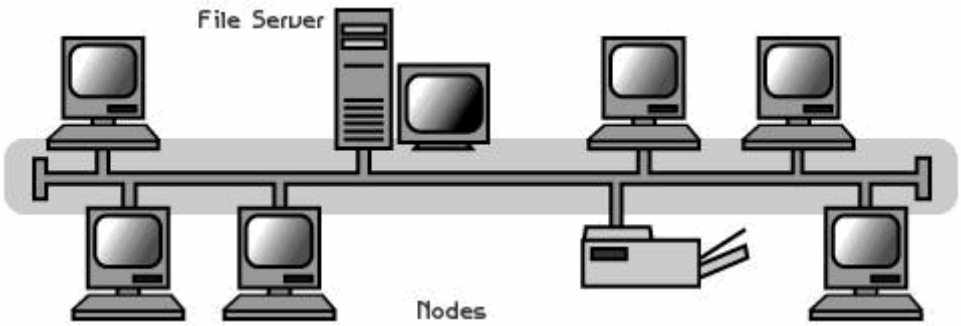
The study of network topology recognizes seven basic topologies: [3]

- Point-to-point topology
- Bus (point-to-multipoint) topology
- Star topology
- Ring topology
- Tree topology
- Mesh topology
- Hybrid topology

This classification is based on the interconnection between computers — be it physical or logical.

The physical topology of a network is determined by the capabilities of the network access devices and media, the level of control or fault tolerance desired, and the cost associated with cabling or telecommunications circuits.

Bus



Bus network topology

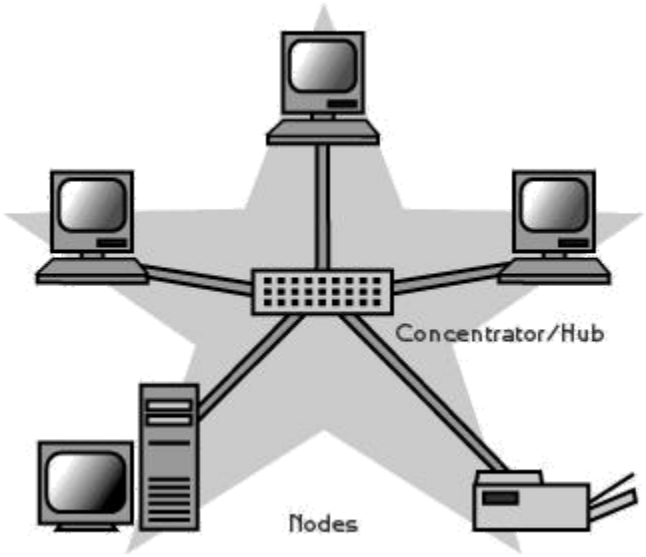
In local area networks where bus topology is used, each machine is connected to a single cable. Each computer or server is connected to the single bus cable through some kind of connector. A terminator is required at each end of the bus cable to prevent the signal from bouncing back and forth on the bus cable. A signal from the source travels in both directions to all machines connected on the bus cable until it finds the MAC address or IP address on the network that is the intended recipient. If the machine address does not match the intended address for the data, the machine ignores the data.

Alternatively, if the data does match the machine address, the data is accepted. Since the bus topology consists of only one wire, it is rather inexpensive to implement when compared to other topologies. However, the low cost of implementing the technology is offset by the high cost of managing the network. Additionally, since only one cable is utilized, it can be the single point of failure. If the network cable breaks, the entire network will be down.

Star

A star topology is designed with each node (file server, workstations, and peripherals) connected

directly to a central network hub, switch, or concentrator (See fig. 2).



Data on a star network passes through the hub, switch, or concentrator before continuing to its destination. The hub, switch, or concentrator manages and controls all functions of the network. It also acts as a repeater for the data flow. This configuration is common with twisted pair cable; however, it can also be used with coaxial cable or fiber optic cable.

Advantages of a Star Topology

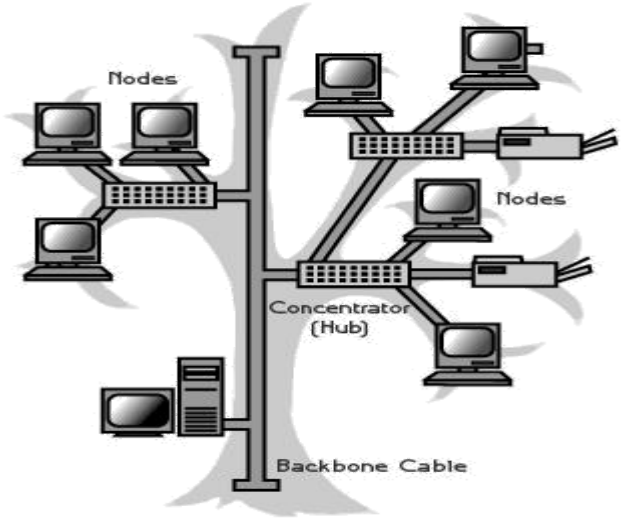
- Easy to install and wire.
- No disruptions to the network when connecting or removing devices.
- Easy to detect faults and to remove parts.

Disadvantages of a Star Topology

- Requires more cable length than a linear topology.
- If the hub, switch, or concentrator fails, nodes attached are disabled.
- More expensive than linear bus topologies because of the cost of the hubs, etc.

Tree or Expanded Star

A tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable (See fig.). Tree topologies allow for the expansion of an existing network, and enable schools to configure a network to meet their needs.



Advantages of a Tree Topology

- º Point-to-point wiring for individual segments.
- º Supported by several hardware and software venders.

Disadvantages of a Tree Topology

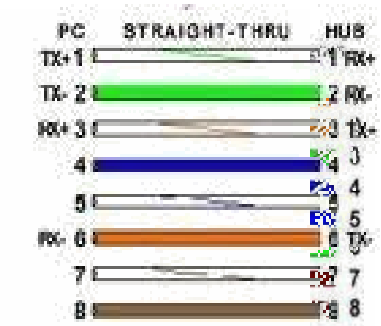
- º Overall length of each segment is limited by the type of cabling used.
- º If the backbone line breaks, the entire segment goes down.
- º More difficult to configure and wire than other topologies.

PROCEDURE

There are generally three main types of networking cables: straight-through, crossover, and rollover cables. Each cable type has a distinct use, and should not be used in place of another. So how do you know which cable to use for what you need?

The Purpose of Straight-Through Cables

Straight-through cables get their name from how they are made. Out of the 8 pins that exist on both ends of an Ethernet cable, each pin connects to the same pin on the opposite side. Review the diagram below for a visual example:



Notice how each wire corresponds to the same pin. This kind of wiring diagram is part of the 568A standard. The 568B standard achieves the same thing, but through different wiring. It is generally accepted to use the 568A standard as pictured, since it allows compatibility with certain telephone hardware- while 568**B doesn't.**

Straight-through cables are primarily used for connecting unlike devices. A straight-through cable is typically used in the following situations:

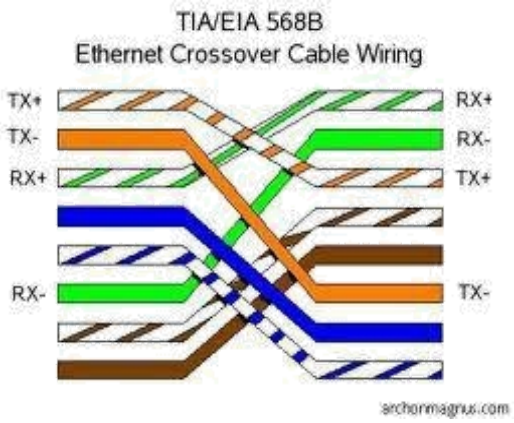
Use a straight-through cable when:

- Connecting a router to a hub
- Connecting a computer to a swtich
- Connecting a LAN port to a switch, hub, or computer

Note that some devices such as routers will have advanced circuitry, which enables them to use both crossover and straight-through cables. In general, however, straight-through cables will not connect a co**mputer and router because they are not "unlike devices."**

The purpose of Crossover Cables

Crossover cables are very similar to straight-through cables, except that they have pairs of wires that crisscross. This allows for two devices to communicate at the same time. Unlike straight-through cables, we use crossover cables to connect like devices. A visual example can be seen below:



Notice how all we did was switch the orange-white and green-white wires, and then the orange and green wires. This will enable like devices to communicate. Crossover cables are typically used in the following situations:

Use a crossover cable when:

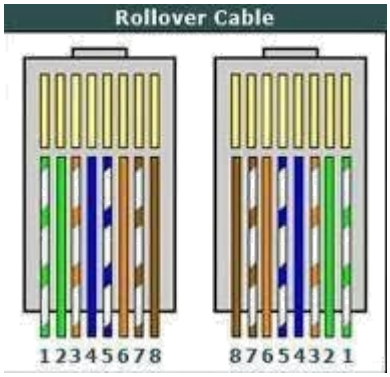
- Connecting a computer to a router
- Connecting a computer to a computer
- Connecting a router to a router
- Connecting a switch to a switch
- Connecting a hub to a hub

While the rule of thumb is to use crossover cables with like devices, some devices do not follow standards. Others provide support for both types of cables. However, there is still something that both crossover and straight-**through cables can't do.**

The Purpose of Rollover Cables

Rollover cables, like other cabling types, got their name from how they are wired. Rollover cables essentially have one end of the cable wired exactly opposite from the other. This **essentially "rolls over" the wires**- but why would we need to do such a thing? Rollover **cables, also called Yost cables, usually connect a device to a router or switch's console port.**

This allows a programmer to make a connection to the router or switch, and program it as needed. A visual example can be seen below:



Notice that each wire is simply "rolled over." These types of cables are generally not used very much, so are usually colored differently from other types of cables.

APPLICATION

1. Most networks in the real world use the above-mentioned devices to create networks.

CONCLUSION

Thus, we have successfully understand the structure and working of various networks including hands on experience of making and testing cables.

Assignment No.	3
Title	**Study of Hamming and CRC**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 03

Tittle: To demonstrate error detection and correction using Hamming Codes or CRC

Objectives : To implement error detection and correction techniques

Problem Statement: Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

Outcome: Demonstrate Hamming Codes or CRC with example.

Software Requirements: C and wireshark

Operating System: Open source (Ubuntu)

THEORY:

CRC:

A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents.

On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption. CRCs can be used for errorcorrection, see bitfilters.

CRCs are so called because the *check* (data verification) value is a *redundancy* (it expands the message without adding information) and the algorithm is based on *cyclic* codes.

CRCs are popular because they are simple to implement in binary hardware, easy to analyze mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function.

The CRC was invented by W. Wesley Peterson in 1961; the 32-bit CRC function of Ethernet and many other standards is the work of several researchers and was publishedin 1975.

CRCs are based on the theory of cyclic error-correcting codes. The use of systematic cyclic codes, which encode messages by adding a fixed-length check value, for the purpose of error detection in communication networks, was first proposed by W. Wesley Peterson in 1961. Cyclic codes are not only simple to

implement but have the benefit of being particularly well suited for the detection of burst errors: contiguous sequences of erroneous data symbols in messages.

This is important because burst errors are common transmission errors in many communication channels, including magnetic and optical storage devices. Typically an *n*- bit CRC applied to a data block of arbitrary length will detect any single error burst not

longer than *n* bits and will detect a fraction $1/(1 - 2^{-n})$ of all longer error bursts.

Specification of a CRC code requires definition of a so-called generator polynomial. This polynomial becomes the divisor in a polynomial long division, which takes the message as the dividend and in which the quotient is discarded and the remainder becomes the result.

The important caveat is that the polynomial coefficients are calculated according to the arithmetic of a finite field, so the addition operation can always be performed bitwise- parallel (there is no carry between digits). The length of the remainder is always less than the length of the generator polynomial, which therefore determines how long the result can be.

In practice, all commonly used CRCs employ the Galois field of two elements, GF(2). The two elements are usually called 0 and 1, comfortably matching computer architecture.

A CRC is called an *n*-bit CRC when its check value is *n* bits long. For a given *n*, multiple CRCs are possible, each with a different polynomial. Such a polynomial has highest degree *n*, which means it has *n* + 1 terms.

In other words, the polynomial has a length of *n* + 1; its encoding requires *n* + 1 bits. Note that most polynomial specifications either drop the MSB or LSB, since they are always 1. The CRC and associated polynomial typically have a name of the form CRC-*n*-XXX as in the table below.

The simplest error-detection system, the parity bit, is in fact a trivial 1-bit CRC: it uses the generator polynomial *x* + 1 (two terms), and has the name CRC-1.

Application

A CRC-enabled device calculates a short, fixed-length binary sequence, known as the *check value* or *CRC*, for each block of data to be sent or stored and appends it to the data, forming a *codeword*.

When a codeword is received or read, the device either compares its check value with one freshly calculated from the data block, or equivalently, performs a CRC on the whole codeword and compares the resulting check value with an expected *residue* constant.

If the CRC check values do not match, then the block contains a data error.

The device may take corrective action, such as rereading the block or requesting that it be sent again. Otherwise, the data is assumed to be error-free (though, with some small

probability, it may contain undetected errors; this is the fundamental nature of error- checking).

Data integrity

CRCs are specifically designed to protect against common types of errors on communication channels, where they can provide quick and reasonable assurance of the integrity of messages delivered. However, they are not suitable for protecting against intentional alteration of data.

Firstly, as there is no authentication, an attacker can edit a message and recompute the CRC without the substitution being detected. When stored alongside the data, CRCs and cryptographic hash functions by themselves do not protect against *intentional* modification of data.

Any application that requires protection against such attacks must use cryptographic authentication mechanisms, such as message authentication codes or digital signatures (which are commonly based on cryptographic hash functions).

Secondly, unlike cryptographic hash functions, CRC is an easily reversible function, which makes it unsuitable for use in digital signatures.

Thirdly, CRC is a linear function with a property that ; as a result, even if the CRC is encrypted with a stream cipher that uses XOR as its combining operation (or mode of block cipher which effectively turns it into a stream cipher, such as OFB or CFB), both the message and the associated CRC can be manipulated without knowledge of the encryption key; this was one of the well-known design flaws of the Wired Equivalent Privacy (WEP) protocol.[5]

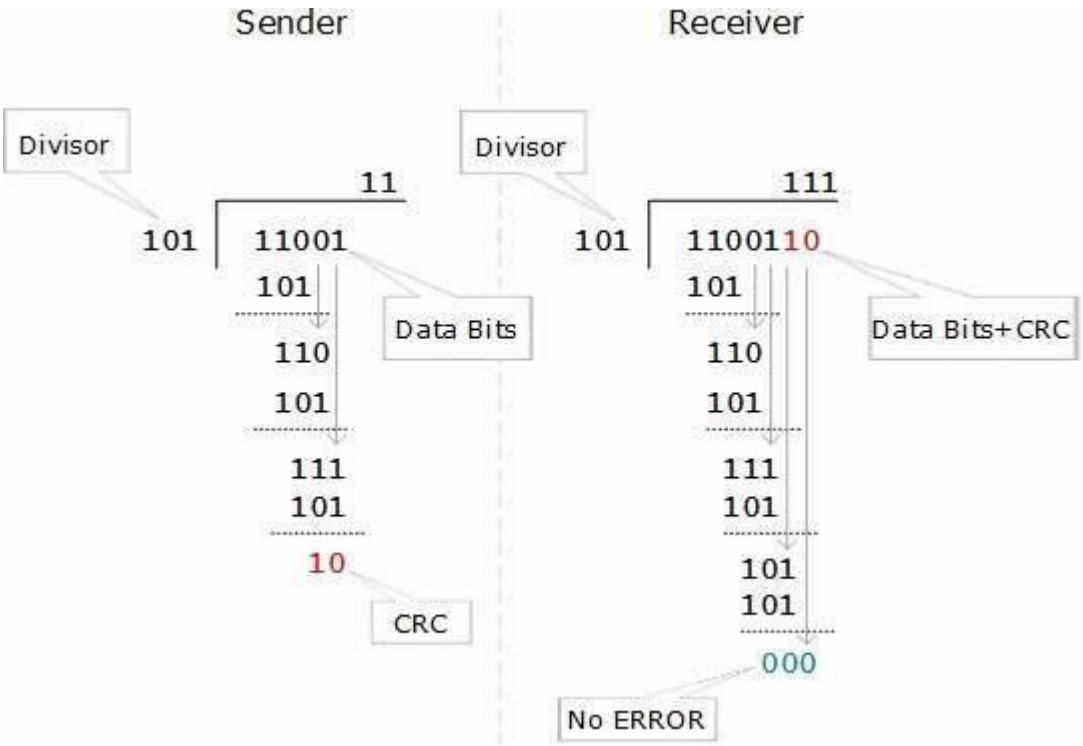
To compute an n-bit binary CRC, line the bits representing the input in a row, and position the $(n + 1)$-bit pattern representing the CRC's divisor (called a

"polynomial") underneath the left-hand end of the row.

Example:

Cyclic Redundancy Check: CRC

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as codewords.



At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

Hamming code

- Hamming codes are a family of linear error-correcting codes that generalizethe Hamming(7,4)-code
- Invented by Richard Hamming in 1950

Hamming codes can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors.

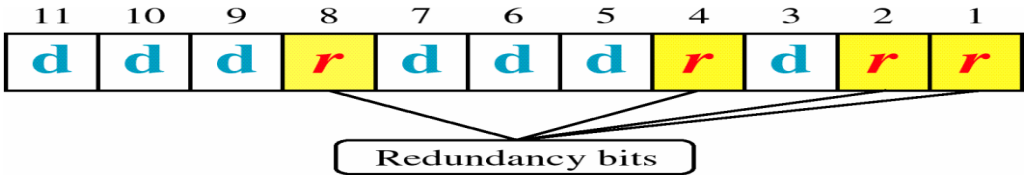
General algorithm

- The following general algorithm generates a single-error correcting (SEC) codefor any number of bits.

- Number the bits starting from 1: bit 1, 2, 3, 4, 5, etc.

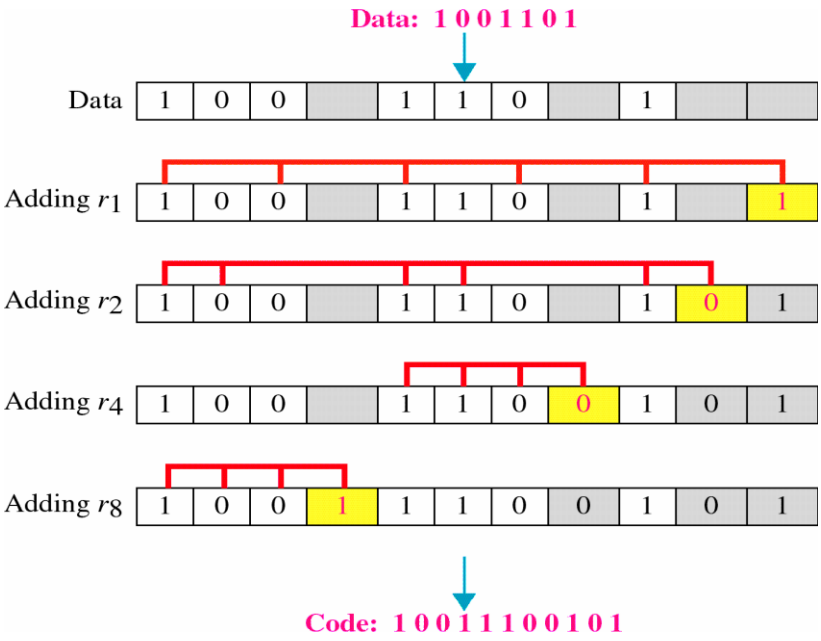
- Write the bit numbers in binary: 1, 10, 11, 100, 101, etc.

- All bit positions that are powers of two (have only one 1 bit in the binary form oftheir position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)
- All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.
- Each data bit is included in a unique set of 2 or more parity bits, as determinedby the binary form of its bit position.
- Each data bit is included in a unique set of 2 or more parity bits, as determinedby the binary form of its bit position.
    - Parity bit 1 covers all bit positions which have the least significant bit set:bit 1 (the parity bit itself), 3, 5, 7, 9, etc.
    - Parity bit 2 covers all bit positions which have the second least significantbit set: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc.
    - Parity bit 4 covers all bit positions which have the third least significantbit set: bits 4–7, 12–15, 20–23, etc.
    - Parity bit 8 covers all bit positions which have the fourth least significantbit set: bits 8–15, 24–31, 40–47, etc.
    - In general each parity bit covers all bits where the bitwise AND of theparity position and the bit position is non-zero.

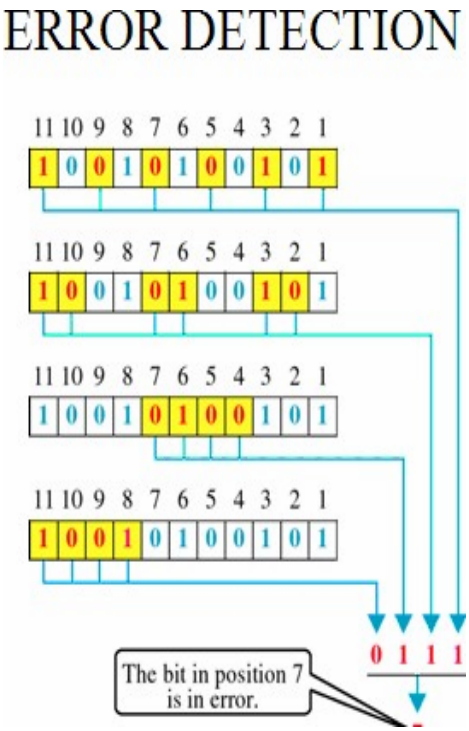


Example

Error detection



Error correction



Conclusion: Hence we have implemented CRC and Hamming code.

Assignment No.	4
Title	**Study of Go back N and Selective Repeat Modes of Sliding Window Protocol.**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 04

Title: Implementation of sliding window protocol(Go back N and Selective Repeat**)**

Objectives: To demonstrate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode .

Problem Statement:

Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.

Outcomes:

Demonstrate Go back N and Selective Repeat Modes and also captured packets using Wireshark Packet Analyzer Tool for peer to peer mode.

Theory:

The basic idea of sliding window protocol is that both sender and receiver keep a ``window'' of acknowledgment. The sender keeps the value of expected acknowledgment; while the receiver keeps the value of expected receiving frame. When it receives an acknowledgment from the receiver, the sender advances the window. When it receives the expected frame, the receiver advances the window.

In transmit flow control, sliding window is a variable-duration window that allows a sender to transmit a specified number of data units before an acknowledgement is received or before a specified event occurs.

Flow Control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data should not be allowed to overwhelm the receiver. Receiver should also be able to inform the transmitter before its limits (this limit may be amount of memory used to store the incoming data or the processing power at the receiver end) are reached and the sender must send fewer frames. Hence, Flow control refers to the set of procedures used to restrict the amount of data the transmitter can send before waiting for acknowledgment.

There are two methods developed for flow control namely Stop-and-wait and Sliding-window Sliding window algorithms, used by TCP, permit multiple data packets to be in simultaneous transit, making more efficient use of network bandwidth.

Sliding Window Protocol:

With the use of multiple frames for a single message, the stop-and-wait protocol does not perform well. Only one frame at a time can be in transit. Efficiency can be greatly improved by

allowing multiple frames to be in transit at the same time. Efficiency can also be improved by making use of the full-duplex line. To keep track of the frames, sender station sends sequentially numbered frames. Since the sequence number to be used occupies a field in the frame, it should be of limited size. If the header of the frame allows k bits, the sequence numbers range from 0 to 2k – 1. Sender maintains a list of sequence numbers that it is allowed to send (sender window).

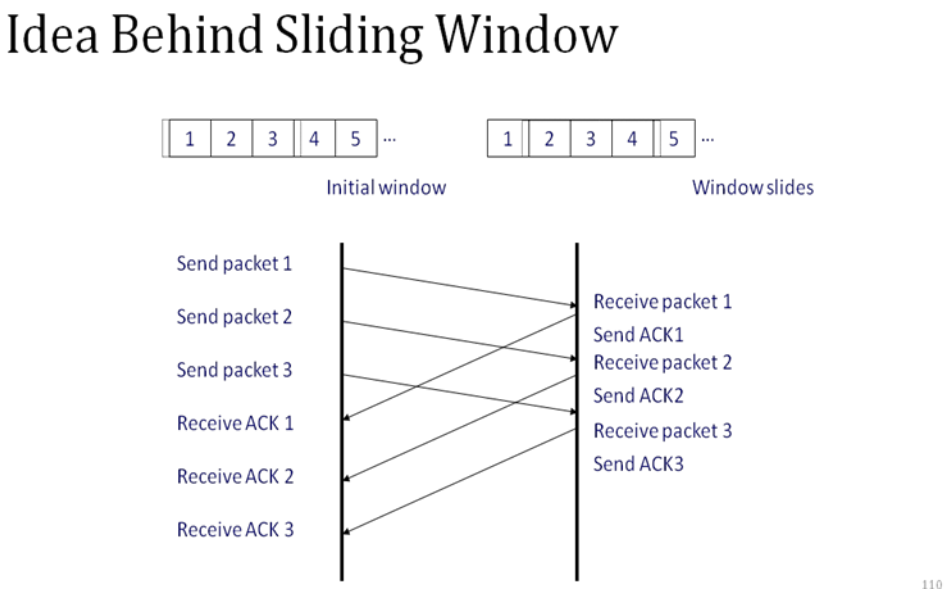
The size of the sender's window is at most 2k – 1. The sender is provided with a buffer equal to the window size. Receiver also maintains a window of size 2k – 1. The receiver acknowledges a frame by sending an ACK frame that includes the sequence number of the next frame expected. This also explicitly announces that it is prepared to receive the next N frames, beginning with the number specified. This scheme can be used to acknowledge multiple frames. It could receive frames 2, 3, 4 but withhold ACK until frame 4 has arrived. By returning an ACK with sequence number 5, it acknowledges frames 2, 3, 4 in one go. The receiver needs a buffer of size 1.

Sliding window algorithm is a method of flow control for network data transfers. TCP, the Internet's stream transfer protocol, uses a sliding window algorithm.

A sliding window algorithm places a buffer between the application program and the network data flow. For TCP, the buffer is typically in the operating system kernel, but this is more of an implementation detail than a hard-and-fast requirement.

Data received from the network is stored in the buffer, from where the application can read at its own pace. As the application reads data, buffer space is freed up to accept more input from the network. The window is the amount of data that can be "read ahead" - the size of the buffer, less the amount of valid data stored in it. Window announcements are used to inform the remote host of the current window size.

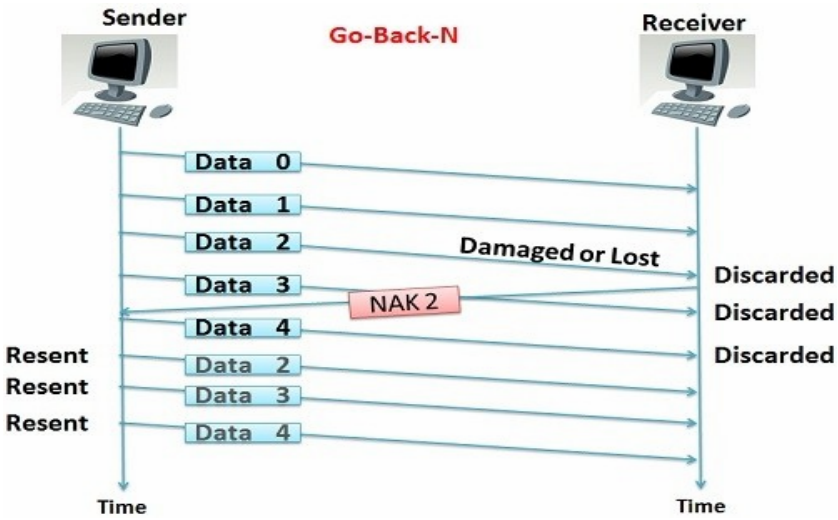
> An example of a sliding window in packet transmission is one in which, after the sender fails to receive an acknowledgement for the first transmitted packet, the sender "slides" the window, i.e. resets the window, and sends a second packet. This process is repeated for the specified number of times before the sender interrupts transmission. Sliding window is sometimes (loosely) called *acknowledgement delay period*.



Go-Back-N Protocol and "Selective Repeat Protocol" are the sliding window protocols. The sliding window protocol is primarily an error control protocol, i.e. it is a method of error detection and error correction. The basic difference between go-back-n protocol and selective repeat protocol is that the "go-back-n protocol" retransmits all the frames that lie after the frame which is damaged or lost. The "selective repeat protocol" retransmits only that frame which is damaged or lost.

Go back N ARQ

In the Go-Back-N Protocol, the sequence numbers are modulo 1!", where *m* is the size of the sequence number field in bits.

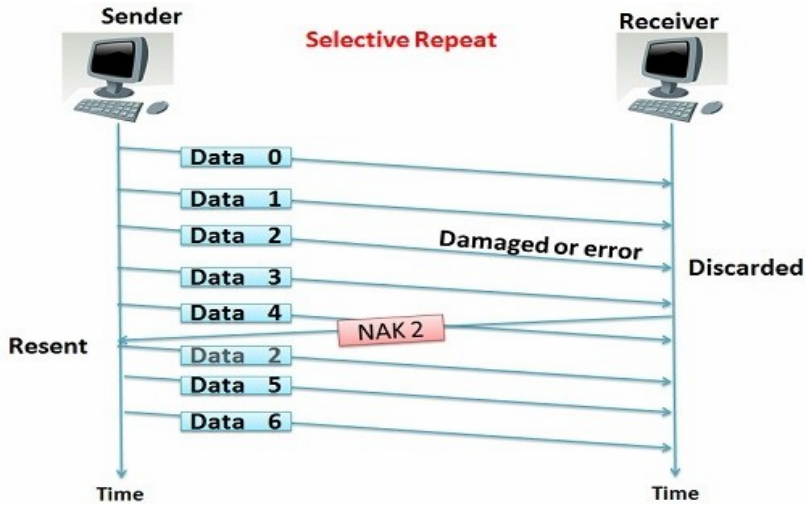


Selective Repeat ARQ

Go-Back-N ARQ simplifies the process at the receiver site. The receiver keeps track of only one variable, and there is no need to buffer out-of-order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link. In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is another mechanism that does not

resend *N* frames when just one frame is damaged;only the damaged frame is resent. This mechanism is called Selective RepeatARQ.

Key Differences Between Go-Back-N and Selective Repeat



1. Go-Back-N protocol is design to retransmit all the frames that are arrived after the damaged or a lost frame. On the other hand, Selective Repeat protocol retransmits only that frame that is damaged or lost.
2. If the error rate is high i.e. more frames are being damaged and then retransmitting all the frames that arrived after a damaged frame waste the lots of bandwidth. On the other hand, selective repeat protocol re-transmits only damaged frame hence, minimum bandwidth is wasted.
3. All the frames after the damaged frame are discarded and the retransmitted frames arrive in a sequence from a damaged frame onwards, so, there is less headache of sorting the frames hence it is less complex. On the other hand only damaged or suspected frame is retransmitted so, extra logic has to be applied for sorting hence, it is more complicated.
4. Go-Back-N has a window size of N-1 and selective repeat have a window size <=(N+1)/2.
5. Neither sender nor receiver need the sorting algorithm in Go-Back-N whereas, receiver must be able to sort the as it has to maintain the sequence.
6. In Go-Back-N receiver discards all the frames after the damaged frame hence, it don't need to store any frames. Selective repeat protocol does not discard the frames arrived after the damaged frame instead it stores those frames till the damaged frame arrives successfully and is sorted in a proper sequence.
7. In selective repeat NAK frame refers to the damaged frame number and in Go-Back-N, NAK frame refers to the next frame expected.
8. Generally the Go-Back-N is more is use due to its less complex nature instead of Selective Repeat protocol.

Conclusion: Hence we have implemented of sliding window protocol(Go back N and Selective Repeat).

Assignment No.	5
Title	**Different types of topologies by using a packet tracer tool**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 05

TITLE: Implementation of different types of topologies and types of transmission media by using a packet tracer tool.

OBJECTIVES: 1. To learn different types of topologies and transmission media

2. To learn packet tracer tool

PROBLEM STATEMENT: Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool

OUTCOME: Students will be able to,

1. get familiar with packet tracer tool
2. implement different topologies using packet tracer tool
3. understand different transmission media

THEORY-CONCEPT:

Network topology is the geometric representation of relationship of all the links connecting the devices or nodes. Network topology represent in two ways one is physical topology that define the way in which a network is physically laid out and other one is logical topology that defines how data actually flow through the network.

Cisco Packet Tracer (CPT) is multi-tasking network simulation software to perform and analyze various network activities such as implementation of different topologies, select optimum path based on various routing algorithms, create DNS and DHCP server, sub netting, analyze various network configuration and troubleshooting commands. In order to start communication between end user devices and to design a network, we need to select appropriate networking devices like routers, switches, hubs and make physical Connection by connection cables to serial and fast Ethernet ports from the component list of packet tracer. Networking devices are costly so it is better to perform first on packet tracer to understand the concept and behavior of networking.

Implementation of Bus Topology in Cisco Packet Tracer

A bus topology is a network in which nodes are directly linked with a common half-duplex link. A host on a bus topology is called a station. In a bus network, every station will accept all network packets, and these packets generated by each station have equal information priority. A bus network includes a single network segment and collision domain.

Steps to Configure and Setup Bus Topology in Cisco Packet Tracer :

Step 1: First, open the cisco packet tracer desktop and select the devices given below:

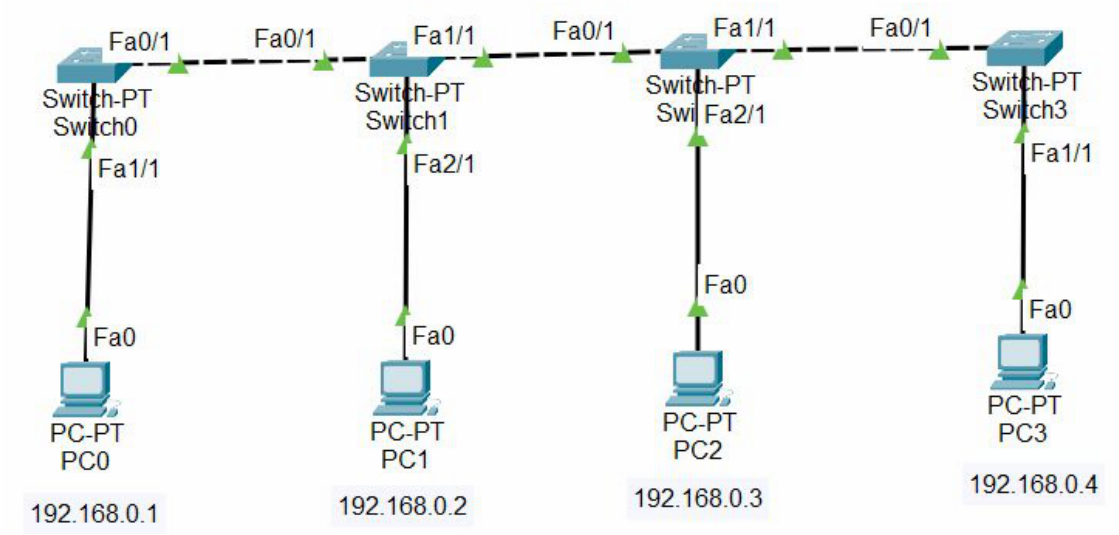
S.NO	Device	Model-Name
1.	PC	PC
2.	Switc	PT-Switch

	h	

IP Addressing Table

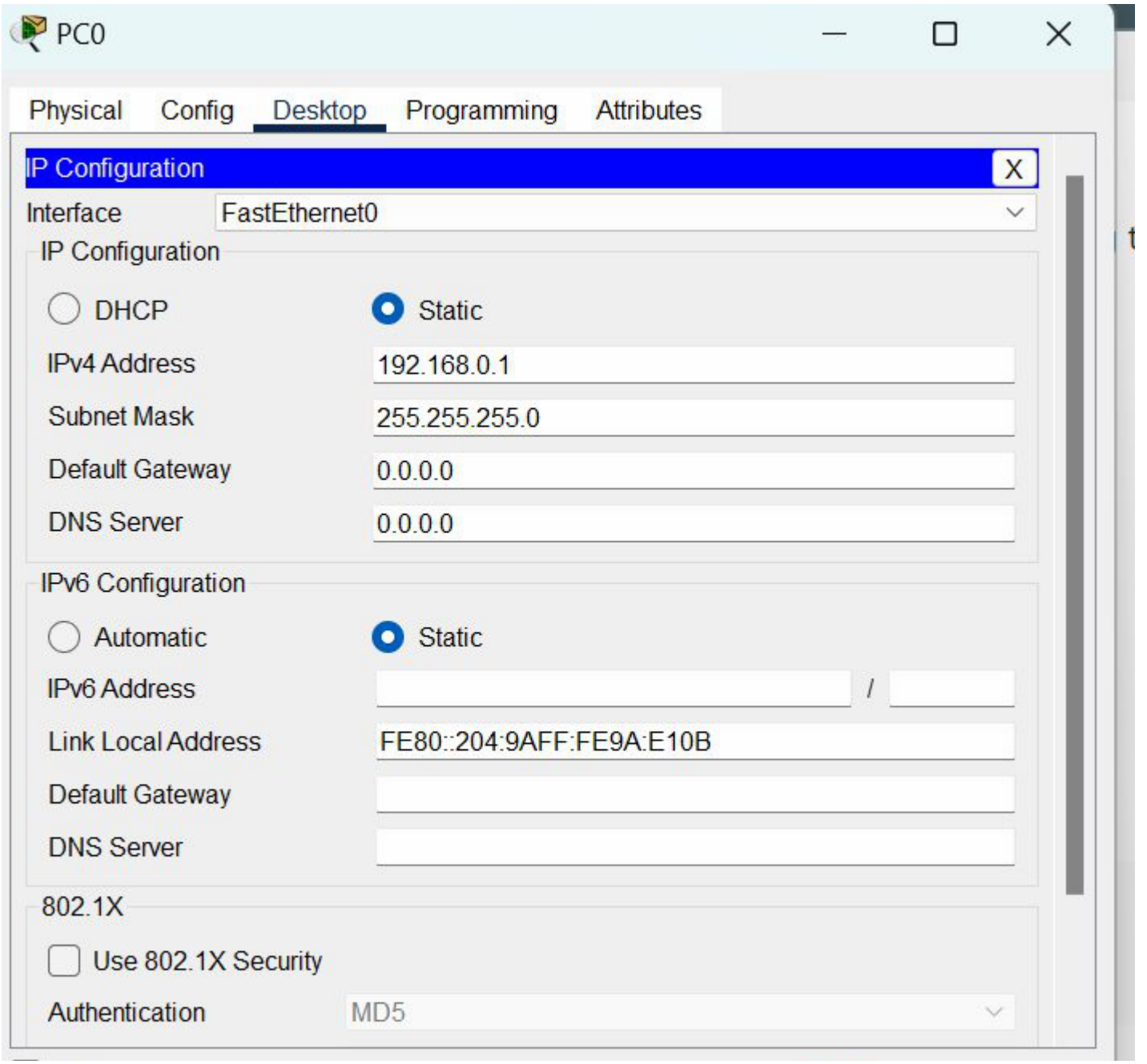
S.NO	Device	IPv4 Address	Subnet Mask
	pc0	192.168.0.1	255.255.255.0
	pc1	192.168.0.2	255.255.255.0
	pc2	192.168.0.3	255.255.255.0
	pc3	192.168.0.4	255.255.255.0

- Then, create a network topology as shown below image:
- Use an Automatic connecting cable to connect the devices with others.



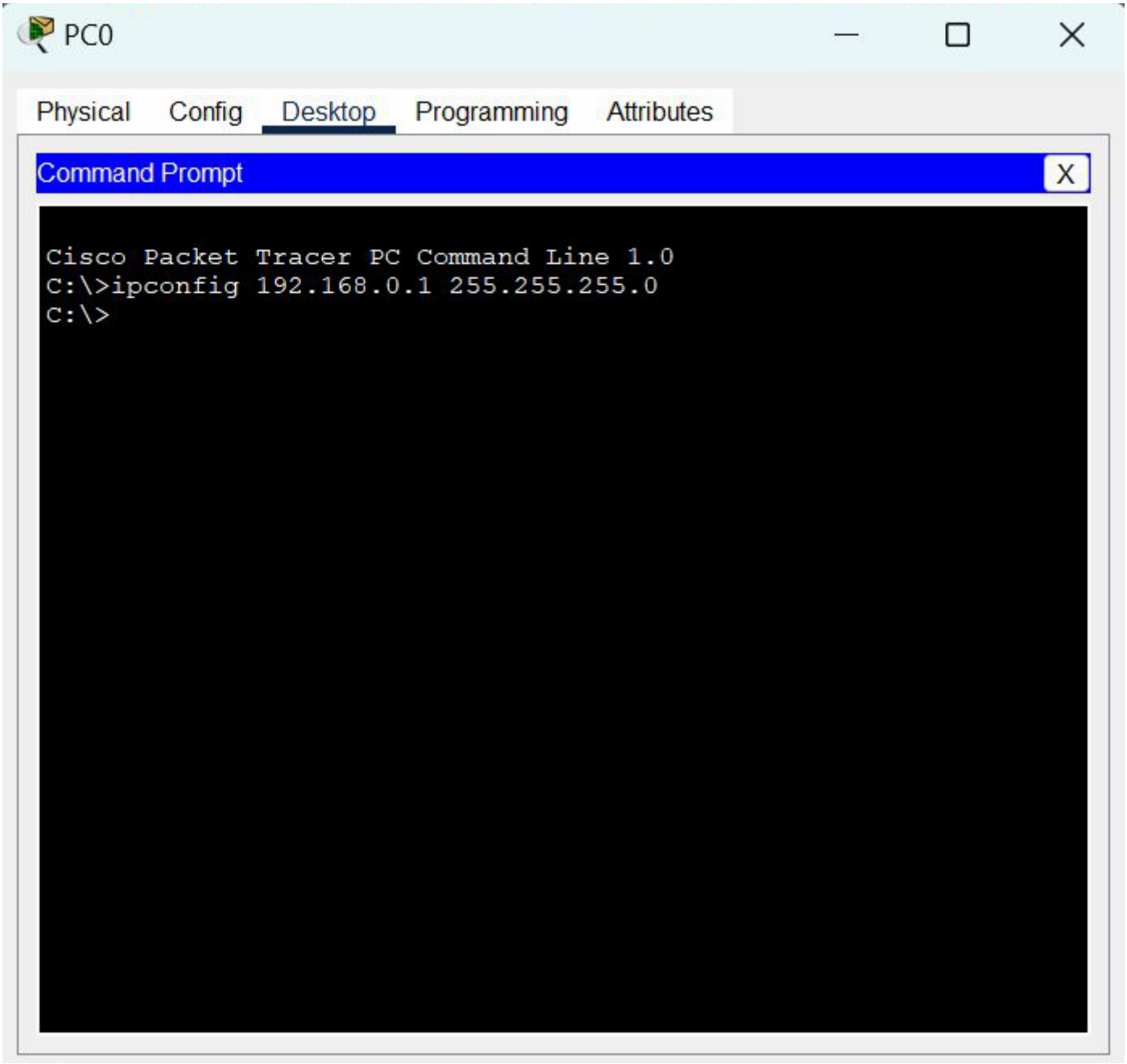
Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.



- Assigning an IP address using the ipconfig command, or we can also assign an IP address with the help of a command.
- Go to the command terminal of the PC.
- Then, type ipconfig <IPv4 address><subnet mask><default gateway>(if needed)

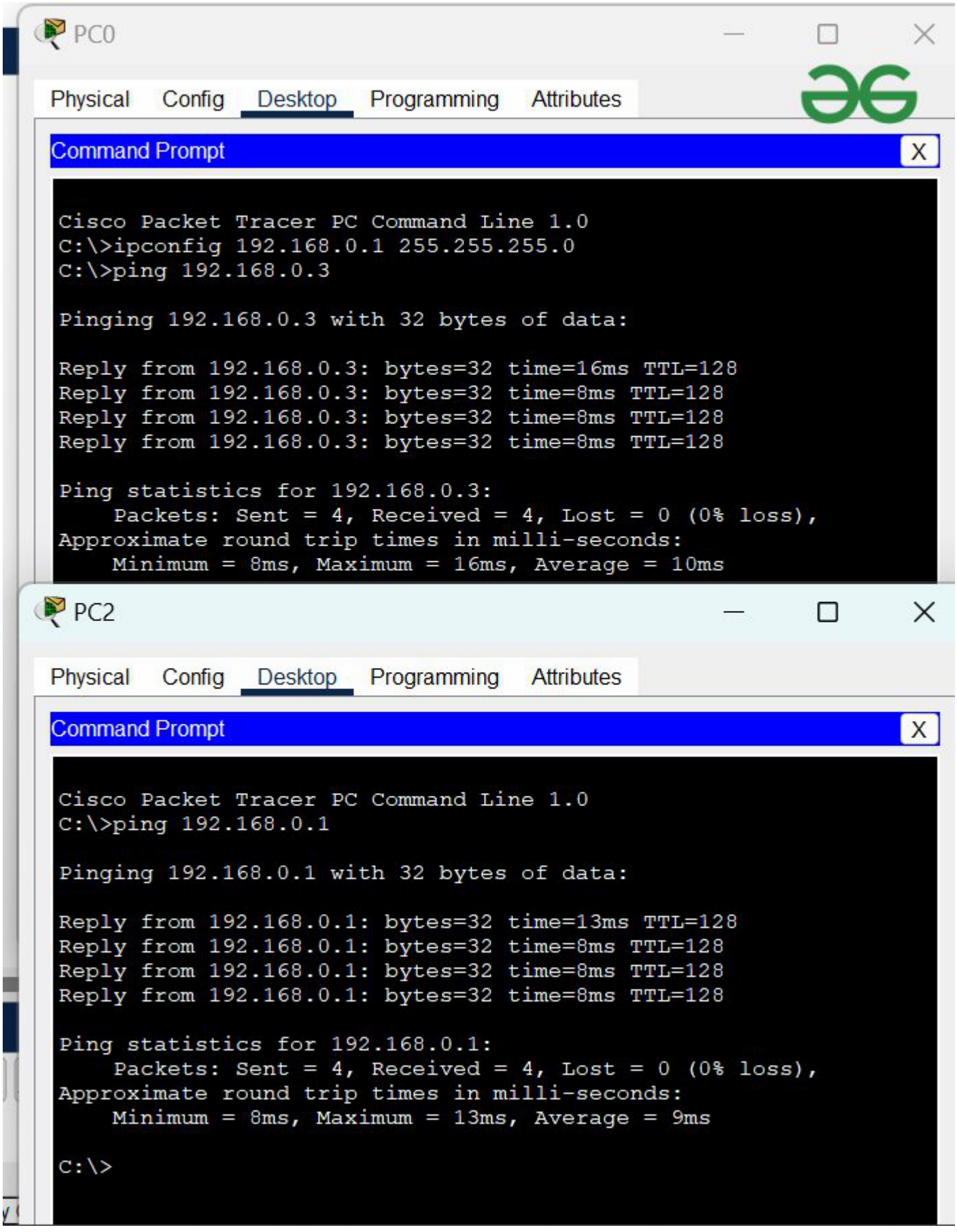
  Example: ipconfig 192.168.0.1 255.255.255.0



- Repeat the same procedure with other PCs to configure them thoroughly.

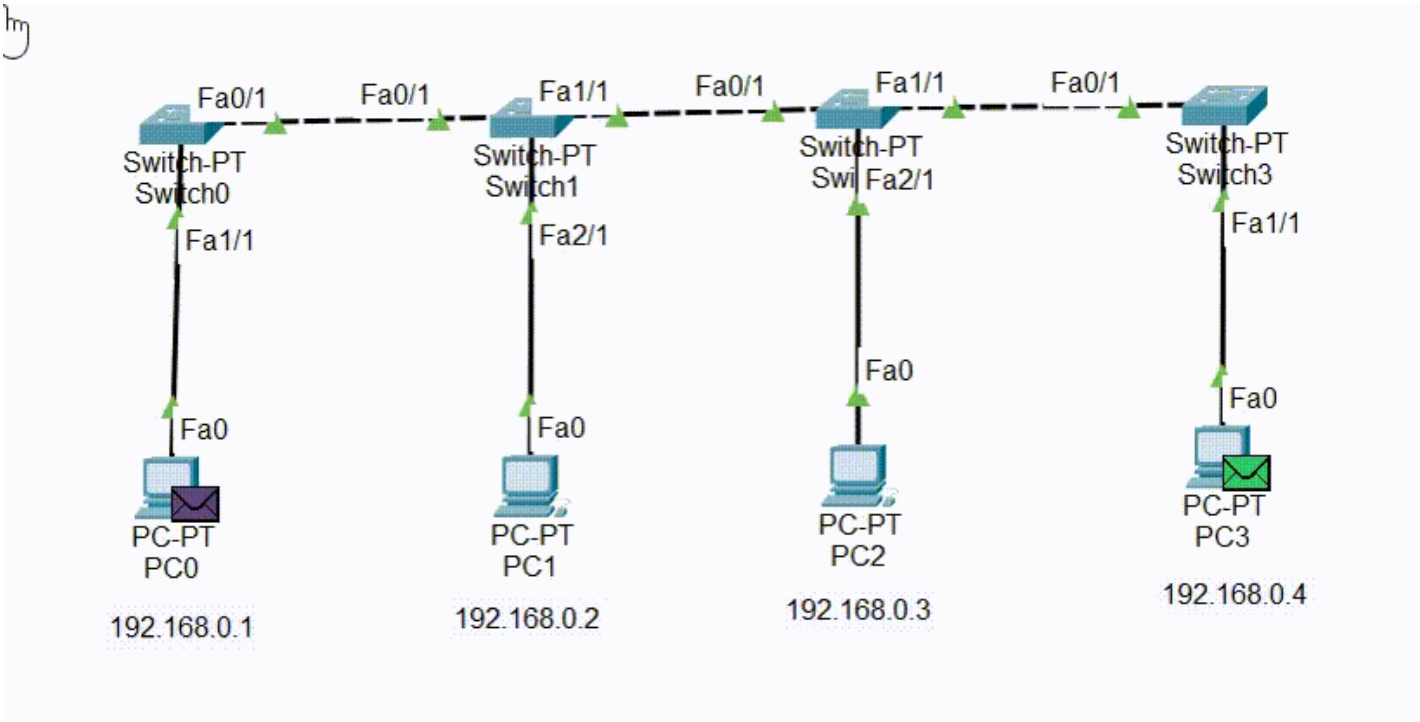
Step 3: Verify the connection by pinging the IP address of any host in PC0.

- Use the ping command to verify the connection.
- As we can see we are getting replies from a targeted node on both PCs.
- Hence the connection is verified.



Simulation Result:

A simulation of the experiment is given below we have sent two PDU packets one targeted from PC0 to PC2 and another targeted from PC3 to PC1.



Implementing Star Topology using Cisco Packet Tracer

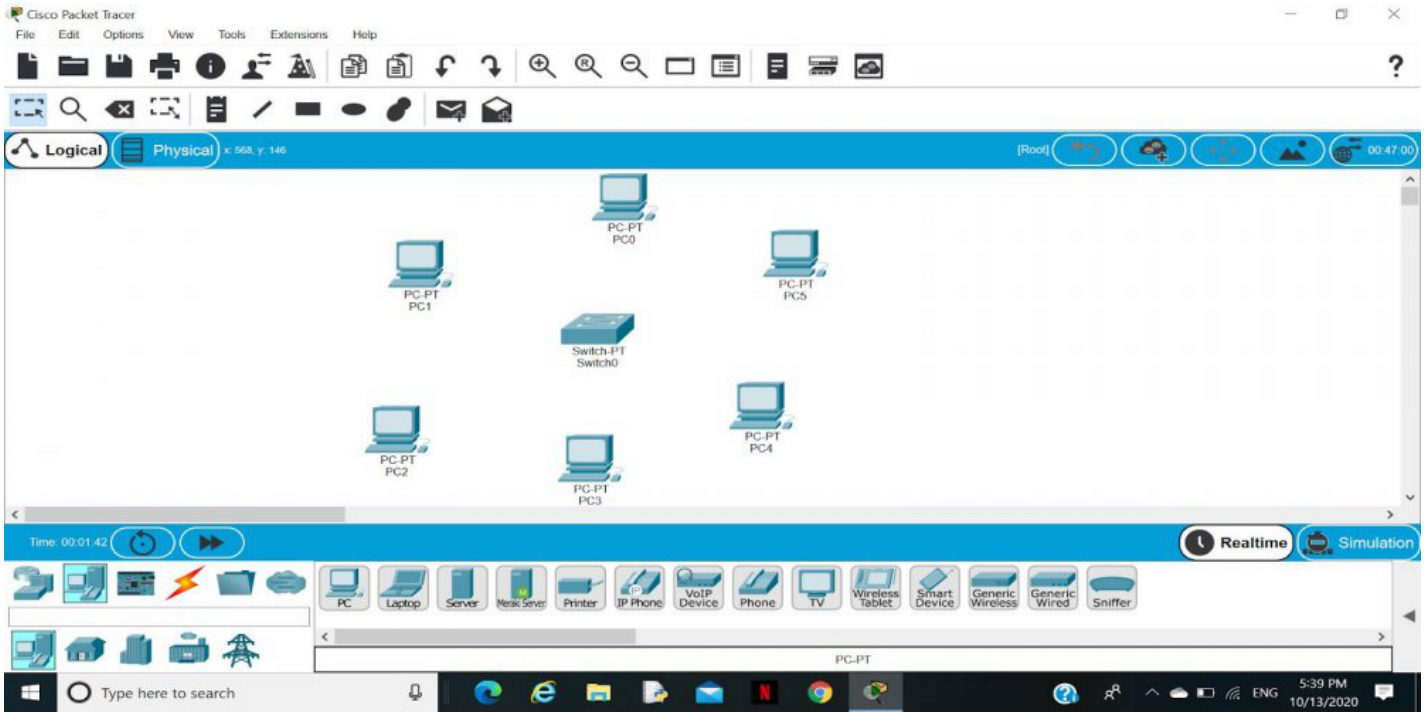
A star topology for a Local Area Network (LAN) is one in which each node is connected to a central connection point, such as a hub or switch. Whenever a node tries to connect with another node then the transmission of the message must be happening with the help of the central node. The best part of star topology is the addition and removal of the node in the network but too many nodes can cause suffering to the network.

In this article, we will discuss How to Implement Star Topology using Cisco Packet Tracer.

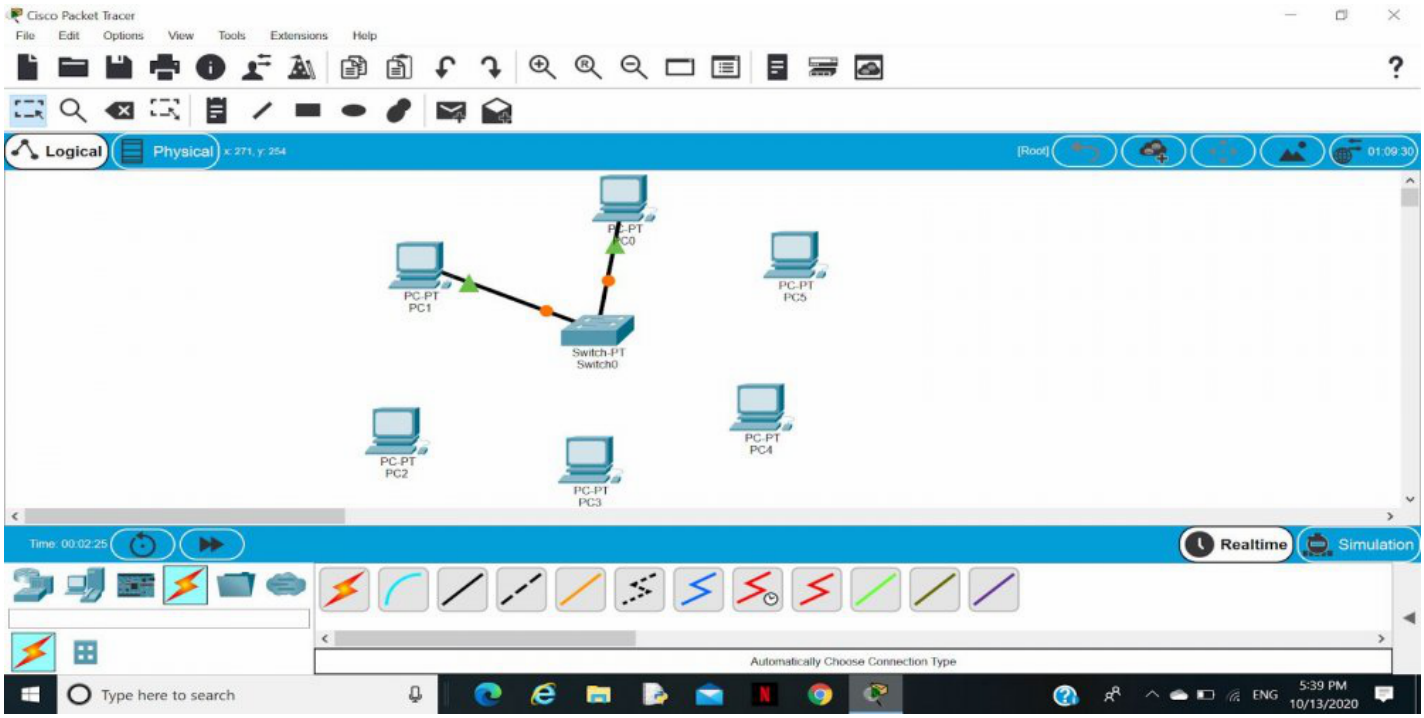
A Cisco packet tracer is a simulation tool that is used for understanding the networks. The best part of the Cisco packet tracer is its visualization you can see the actual flow of the message and understand the workflow of the network devices. Implementation of Star Topology using Cisco Packet Tracer is done using Switch.

Steps Implementing Star Topology using Cisco Packet Tracer:

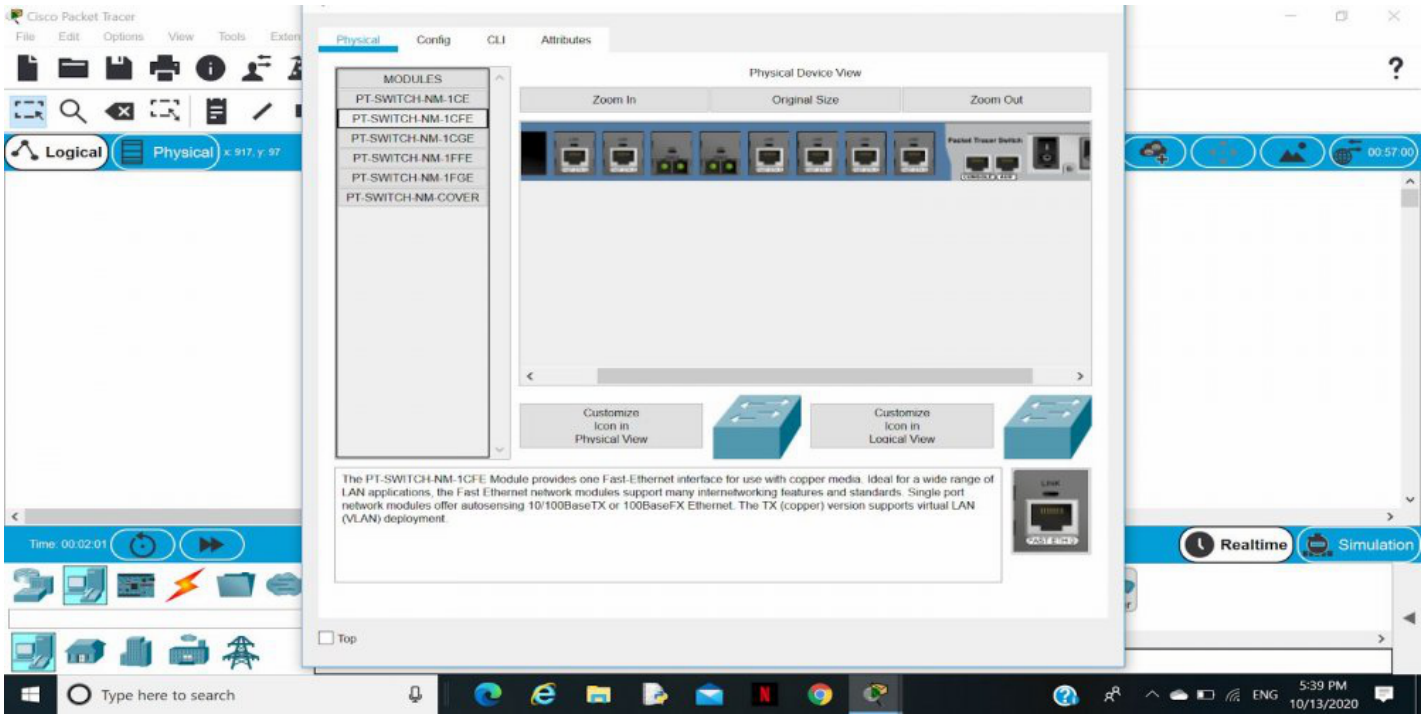
Step 1: We have taken a switch and linked it to six end devices.



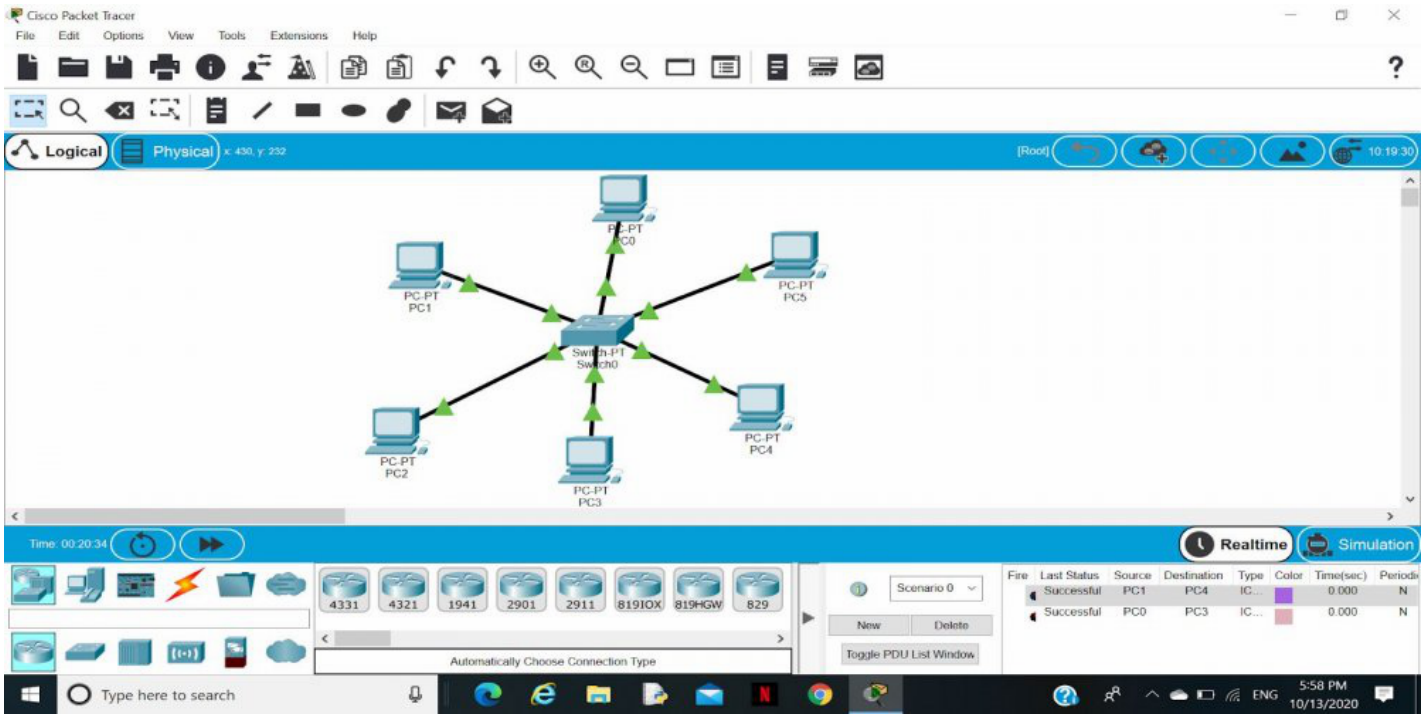
Step 2: Link every device with the switch.



Step 3: Provide the IP address to each device.



Step 4: Transfer message from one device to another and check the Table for Validation.



Now to check whether the connections are correct or not try to ping any device and the image below is doing the same.

To do ping one terminal of one device and run the following command:

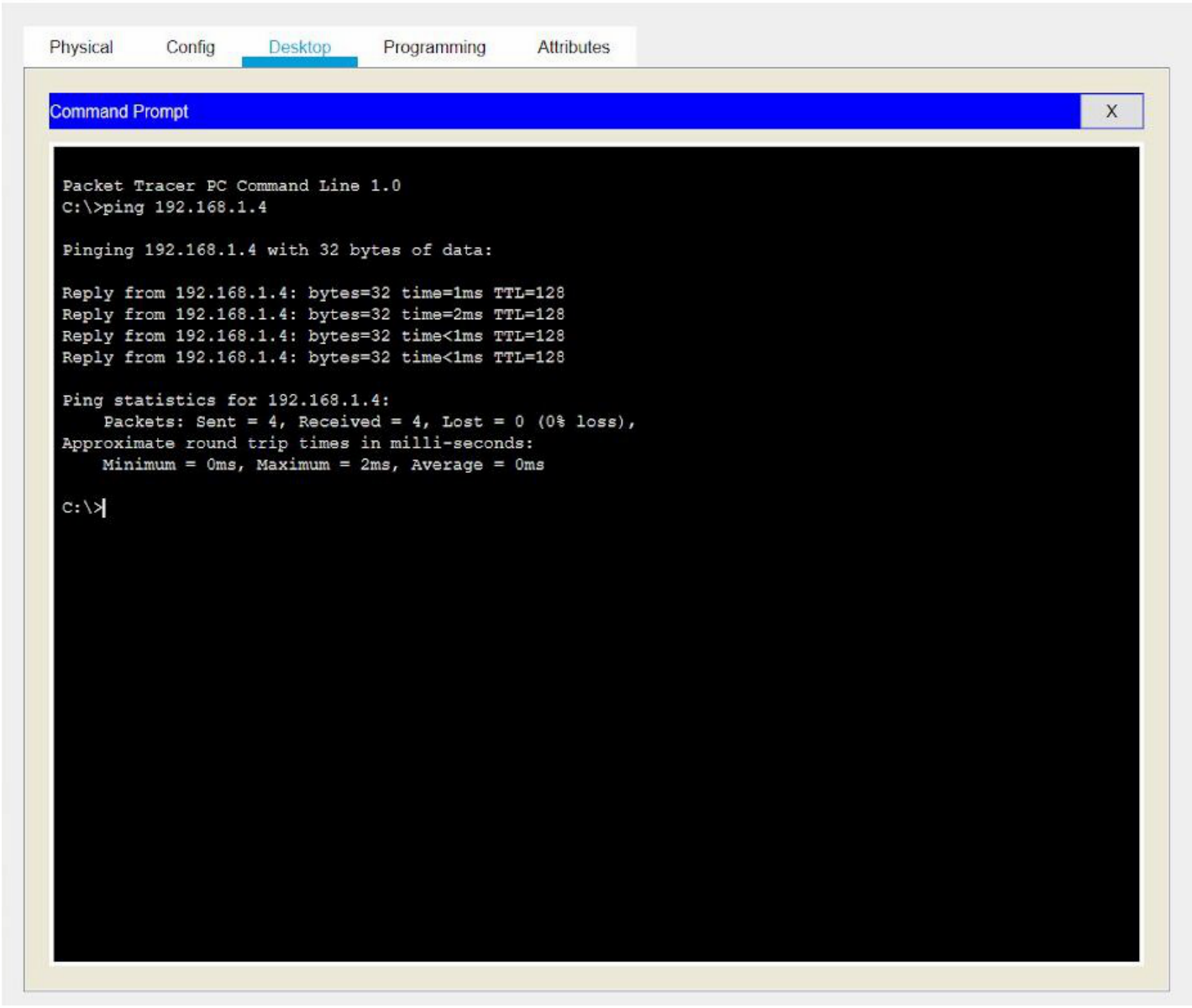
Command:

"ping ip_address_of _any_device"

Example:

ping 192.168.1.4

Note:If the connections are correct
then you will receive the response.



Implementing RING Topology using Cisco Packet Tracer

Ring topology is a kind of arrangement of the network in which every device is linked with two other devices. This makes a circular ring of interconnected devices which gives it its name. Data is usually transmitted in one direction along the ring, known as a unidirectional ring. The data is delivered from one device to the next until it reaches the decided destination. In a bidirectional ring, data can travel in either direction.

To learn about Ring Topology refer to the Advantages and Disadvantages of ring topology article.

Steps to Configure and Setup Ring Topology in Cisco Packet Tracer :

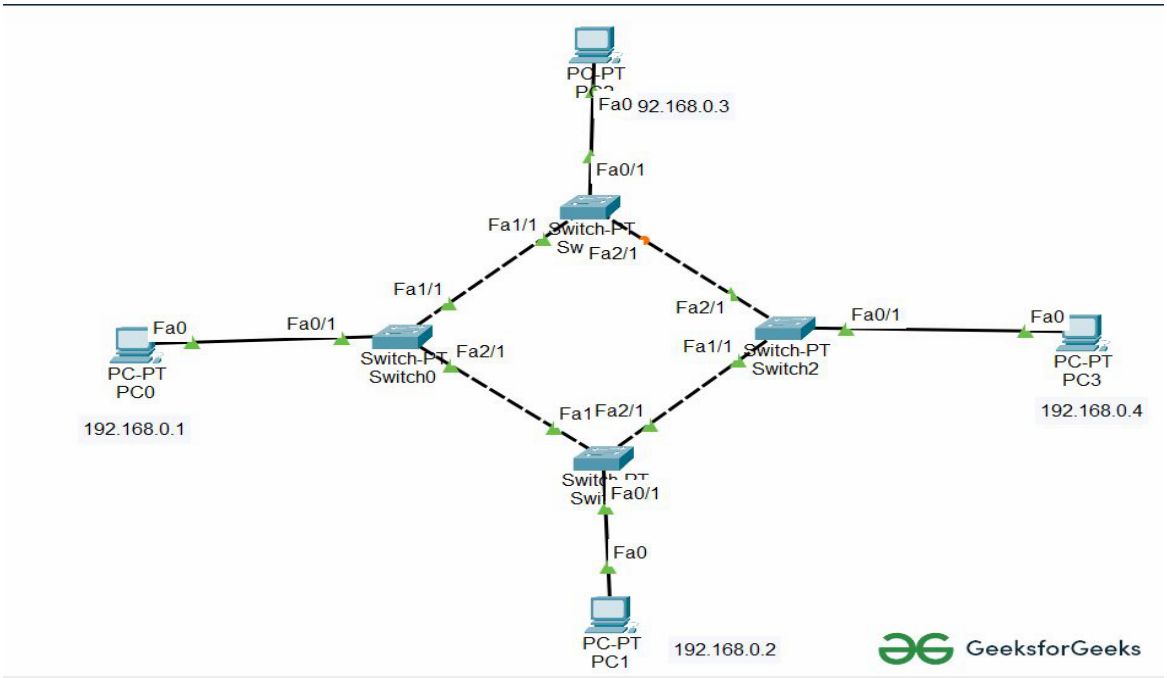
Step 1: First, open the cisco packet tracer desktop and select the devices given below:

S.NO	Device	Model Name
1.	PC	PC
2.	Switch	PT-Switch

IP Addressing Table

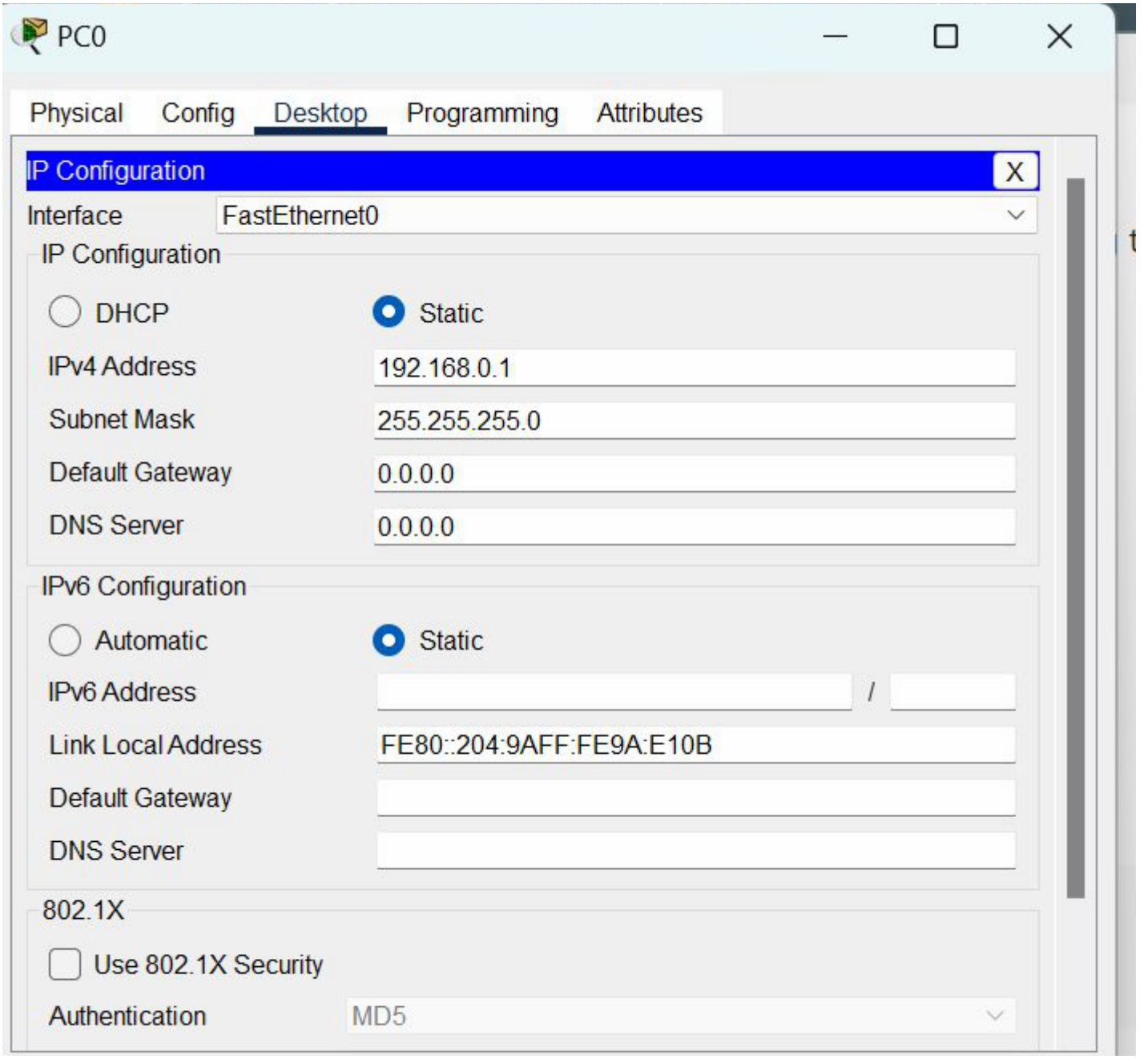
S.NO	Device	IPv4 Address	Subnet Mask
1.	**pc0**	192.168.0.1	255.255.255.0
2.	**pc1**	192.168.0.2	255.255.255.0
3.	**pc2**	192.168.0.3	255.255.255.0
4.	**pc3**	192.168.0.4	255.255.255.0

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.



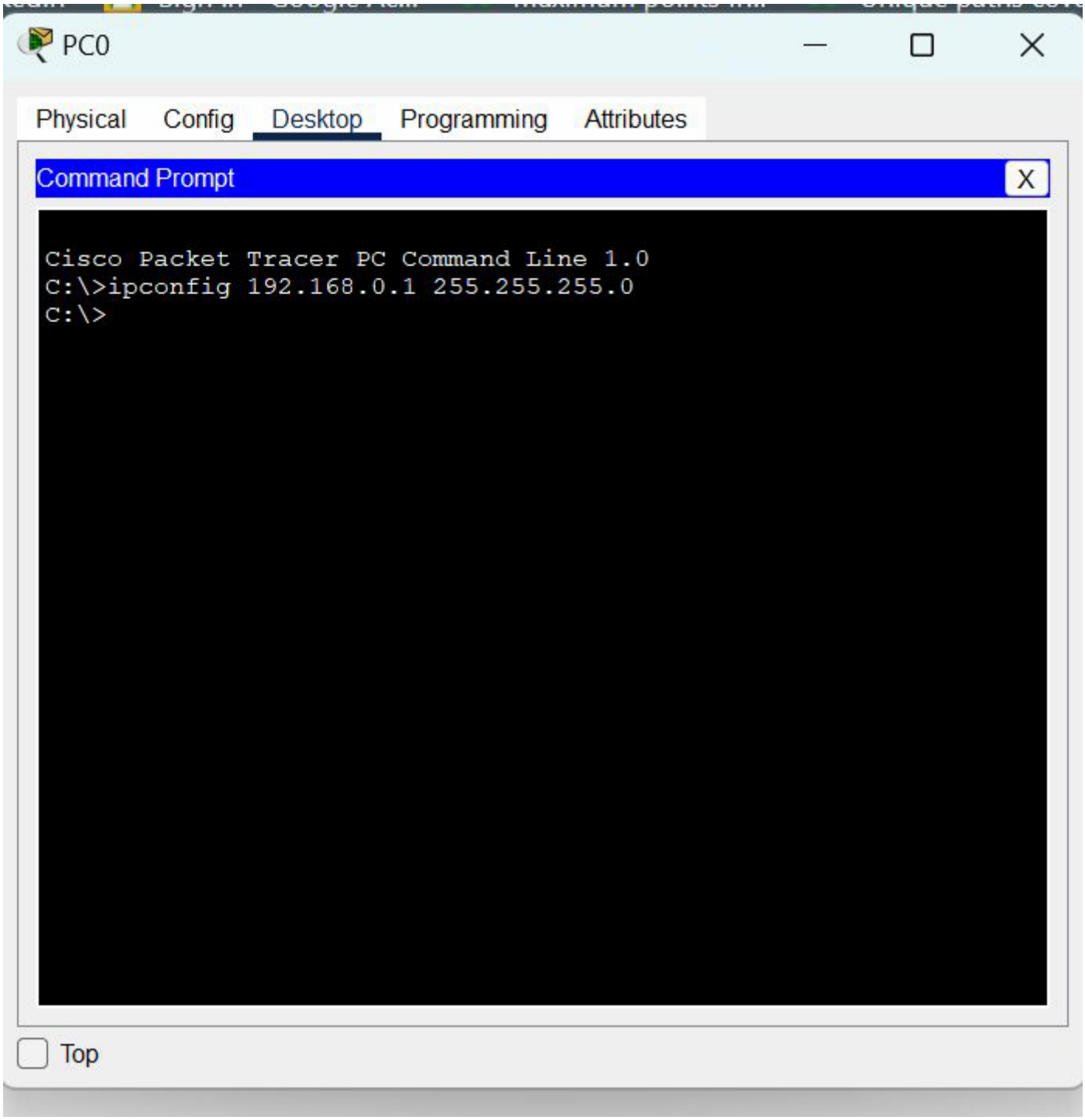
Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.



- Assigning IP address using the ipconfig command, or we can also assign an IP address with the help of a command.
- Go to the command terminal of the PC.
- Then, type ipconfig <IPv4 address><subnet mask><default gateway>(if needed)

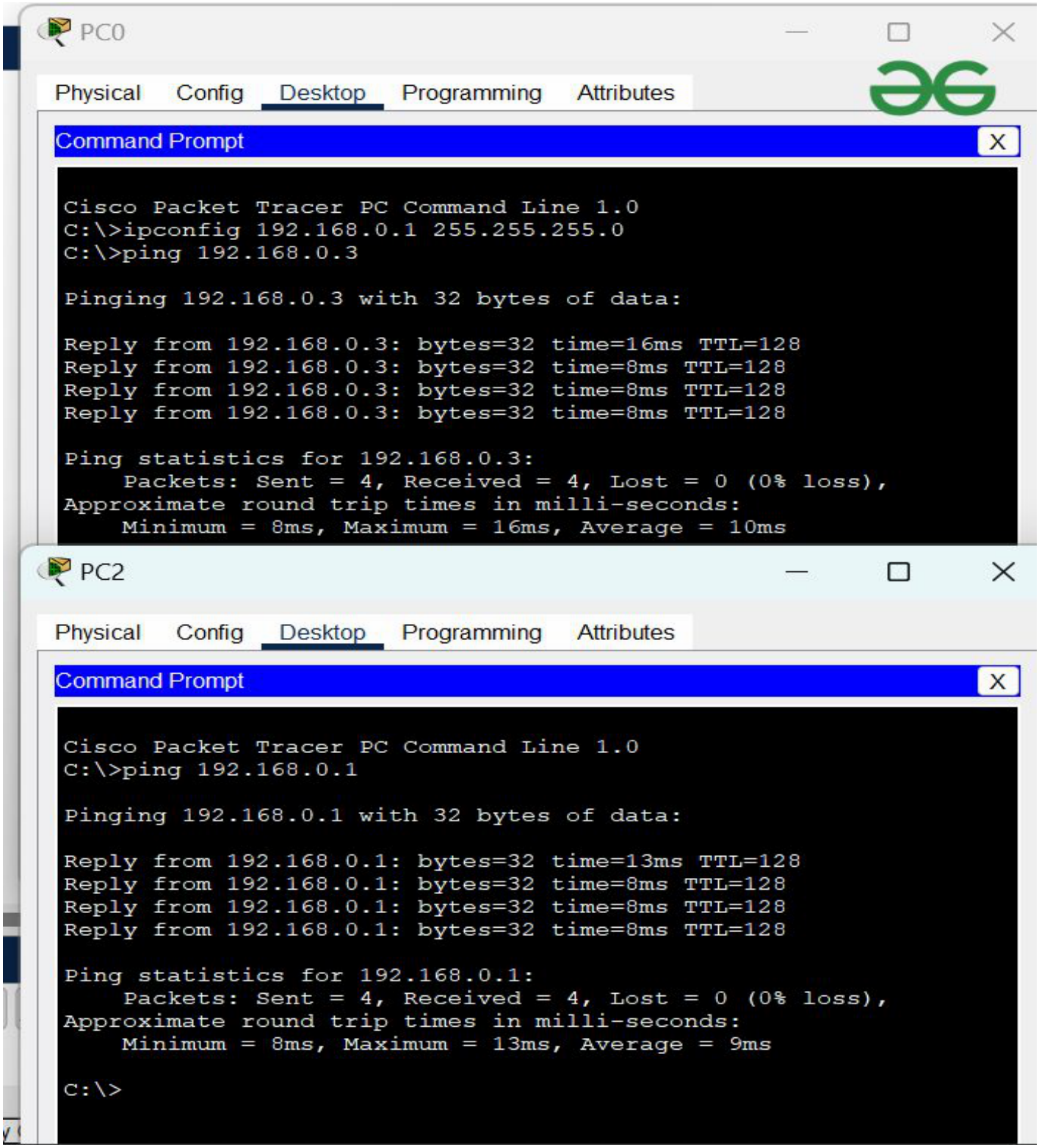
Example: ipconfig 192.168.0.1 255.255.255.0



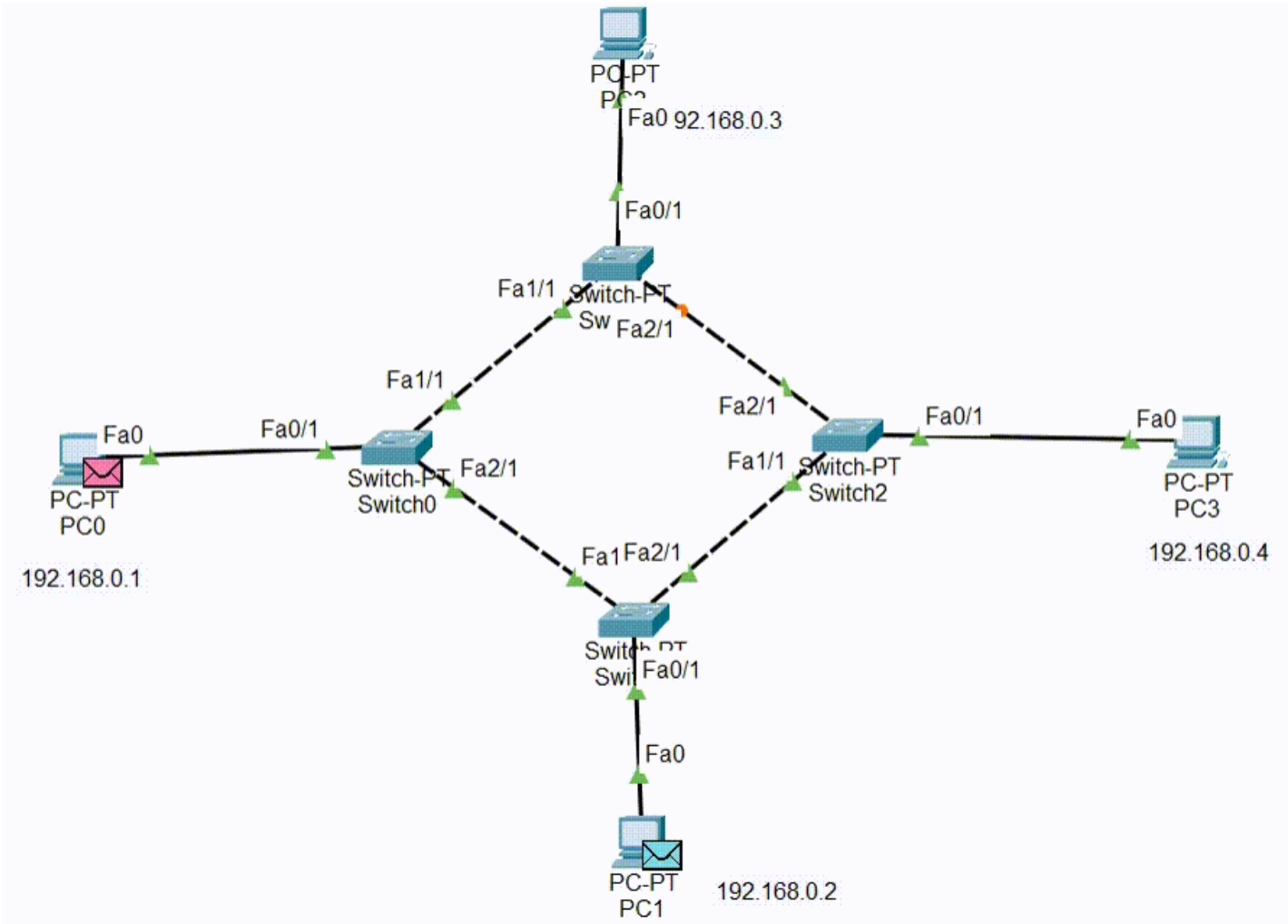
- Repeat the same procedure with other PCs to configure them thoroughly.

Step 3: Verify the connection by pinging the IP address of any host in PC0.

- Use the ping command to verify the connection.
- As we can see we are getting replies from a targeted node on both PCs.
- Hence the connection is verified.



- A simulation of the experiment is given below we have sent two PDU packets one targeted from PC0 to PC2 and another targeted from PC1 to PC3.



CONCLUSION: We have learned how to implement different topologies using different transmission media in packet tracer tool.

Assignment No.	6
Title	**Implementation of link state /Distance vector routing protocol**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 06

AIM: Find suitable path for transmission using link state /Distance vector routing protocol

OBJECTIVE: 1. To learn basic concept of protocol

2.To learn link state/Distance vector routing protocol.

PROBLEM STATEMENT: Write a program to implement link state routing protocol to find a suitable path for transmission.

OUTCOME: Students will be able to

1. Implement link state routing protocol

THEORY-CONCEPT:

Routing algorithm is a part of network layer software which is responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagram internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new established route is being set up. The latter case is sometimes called session routing, because a rout remains in force for an entire user session (e.g., login session at a terminal or a file). Routing algorithms can be grouped into two major classes: adaptive and non-adaptive.

Non-adaptive algorithms do not base their routing decisions on measurement or estimates of current traffic and topology. Instead, the choice of route to use to get from I to J (for all I and J) is compute in advance, offline, and downloaded to the routers when the network ids booted. This procedure is sometime called static routing.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get information (e.g., locally, from adjacent routers, or from all routers), when they change the routes, and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

Two algorithms in particular, distance vector routing and link state routing are the most popular. The Link State Routing Algorithm is an interior protocol used by every router to share information or knowledge about the rest of the routers on the network. The link state routing algorithm is distributed by which every router computes its routing table. With the knowledge of the network topology, a router can make its routing table. Now, for developing the routing table, a router uses a shortest path computation algorithm like Dijkstra's algorithm along with the knowledge

of the topology. The routing table created by each router is exchanged with the rest of the routers present in the network, which helps in faster and more reliable delivery of data.

A router does not send its entire routing table with the rest of the routers in the inter-network. It only sends the information of its neighbors. A router broadcasts this information and contains information about all of its directly connected routers and the connection cost.

In link state routing, each router shares its knowledge of its neighborhood with every other router in the internet work.

(i) Knowledge about Neighborhood: Instead of sending its entire routing table a router sends info about its neighborhood only.

(ii) To all Routers: each router sends this information to every other router on the internet work not just to its neighbor .It does so by a process called flooding. (iii)Information sharing when there is a change: Each router sends out information about the neighbors when there is change.

PROCEDURE:

The Dijkstra algorithm follows four steps to discover what is called the shortest path tree(routing table) for each router:

The algorithm begins to build the tree by identifying its

roots. The root router's trees the router itself.

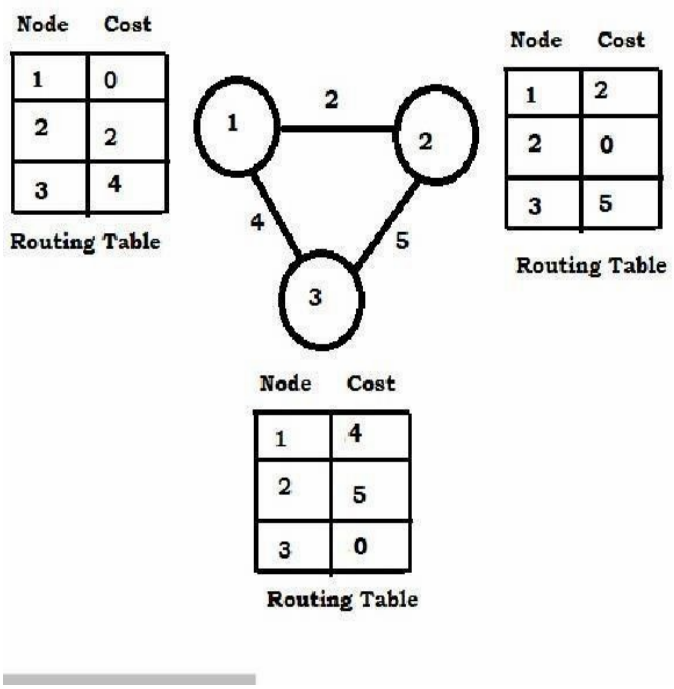
The algorithm then attaches all nodes that can be reached from the root.

The algorithm compares the tree's temporary arcs and identifies the arc with the lowest cumulative cost.

This arc and the node to which it connects are now a permanent part of the shortest path tree. The algorithm examines the database and identifies every node that can be reached from its chosen node. These nodes and their arcs are added temporarily to the tree.

Distance Vector Routing Algorithm using Bellman Ford's Algorithm

OBJECTIVE: Obtain Routing table at each node using distance vector routing algorithm for agiven subnet.



PROGRAM LOGIC:

Distance Vector Routing Algorithms calculate a best route to reach a destination based solely on distance. E.g. RIP. RIP calculates the reach ability based on hop count. It' s different from link state algorithms which consider some other factors like bandwidth and other metrics to reach a destination. Distance vector routing algorithms are not preferable for complex networks and take longer to converge.

PROCEDURE: Go to debug -> run or press CTRL + F9 to run the program.

SOURCE CODE:

```
#include<stdio.h>

struct node

{

   unsigned dist[20];

   unsigned from[20];

}rt[10];

int main()

{

   int costmat[20][20];

   int nodes,i,j,k,count=0;
```

```
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);//Enter the nodes
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
       for(j=0;j<nodes;j++)
       {
          scanf("%d",&costmat[i][j]);
          costmat[i][i]=0;
          rt[i].dist[j]=costmat[i][j];//initialise the distance equal to cost matrix
          rt[i].from[j]=j;
       }
    }
       do
       {
          count=0;
          for(i=0;i<nodes;i++)//We choose arbitary vertex k and we calculate the direct distance
from the node i to k using the cost matrix
          //and add the distance from k to node j
          for(j=0;j<nodes;j++)
          for(k=0;k<nodes;k++)
             if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
             {//We calculate the minimum distance
                rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                rt[i].from[j]=k;
                count++;
```

```
            }
        }while(count!=0);
        for(i=0;i<nodes;i++)

        {
            printf("\n\n For router %d\n",i+1);
            for(j=0;j<nodes;j++)
            {
                printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
            }
        }
    printf("\n\n");
    getch();
}
```

Output:

```
Enter the number of nodes :
3
Enter the cost matrix :
0 2 7
2 0 1
7 1 0
For router 1
node 1 via 1 Distance 0
node 2 via 2 Distance 2
node 3 via 3 Distance 3
For router 2
node 1 via 1 Distance 2
node 2 via 2 Distance 0
node 3 via 3 Distance 1
For router 3
```

node 1 via 1 Distance 3

node 2 via 2 Distance 1

node 3 via 3 Distance 0

CONCLUSION: Thus, we have successfully implemented Distance vector or link state protocol algorithm

Assignment No.	7
Title	**Use packet Tracer tool for configuration of 3 router network for protocol RIP/OSPF/BGP**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 07

TITLE: Configuration of 3 router networks using RIP/OSPF/BGP in Packet Tracer

OBJECTIVES: 1. To configure router.

2. To understand routing protocols.

PROBLEM STATEMENT: Use packet Tracer tool for configuration of 3 router networks using one of the following protocols RIP/OSPF/BGP.

OUTCOME: Students will be able to,

1. Configure router networks using Packet Tracer.
2. Understand routing protocols.

THEORY-CONCEPT:

RIP Routing Configuration Using 3 Routers in Cisco Packet Tracer

Routing Information Protocol (RIP) is an active routing protocol that operates hop count as a routing metric to find the most suitable route between the source and the destination network. It is a distance-vector routing protocol that has an AD value of 120 and works on the Network layer of the OSI model.

Steps to Configure and Verify Three Router Connections in Cisco Packet Tracer using RIP Routing:

Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

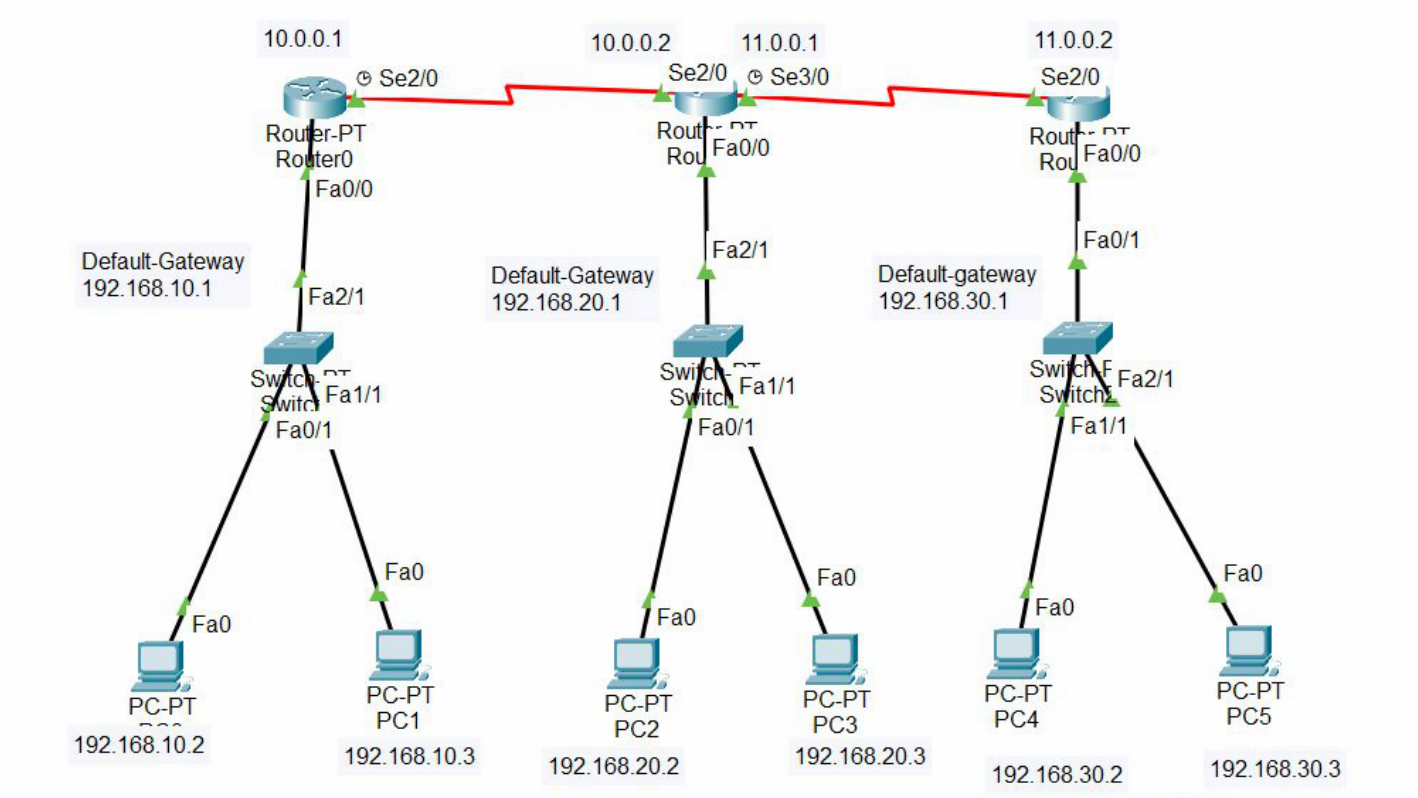
S.NO	Device	Model Name	Qty.
1.	PC	PC	6
2.	Switch	PT-Switch	3
3.	Router	PT-router	3

IP Addressing Table:

S.NO	Device	IPv4 Address	Subnet mask	Default Gateway
1.	PC0	192.168.10.2	255.255.255.0	192.168.10.1
2.	PC1	192.168.10.3	255.255.255.0	192.168.10.1
3.	PC2	192.168.20.2	255.255.255.0	192.168.20.1
4.	PC3	192.168.20.3	255.255.255.0	192.168.20.1

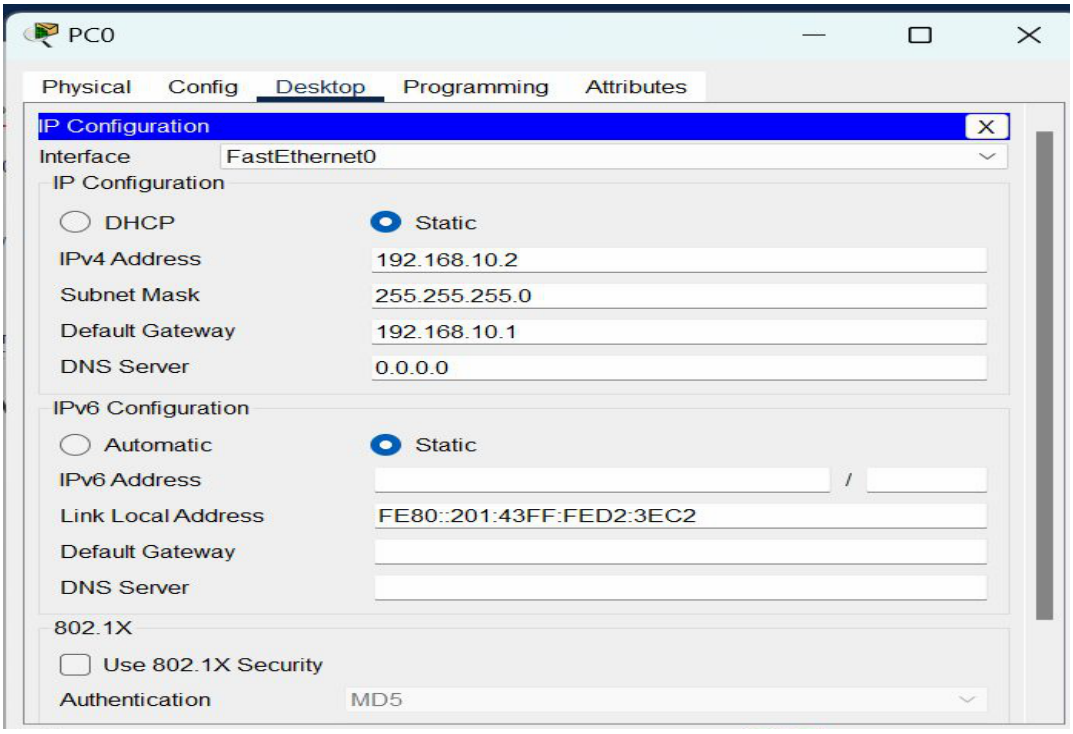
5.	PC4	192.168.30.2	255.255.255.0	192.168.30.1
6.	PC5	192.168.30.3	255.255.255.0	192.168.30.1

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.



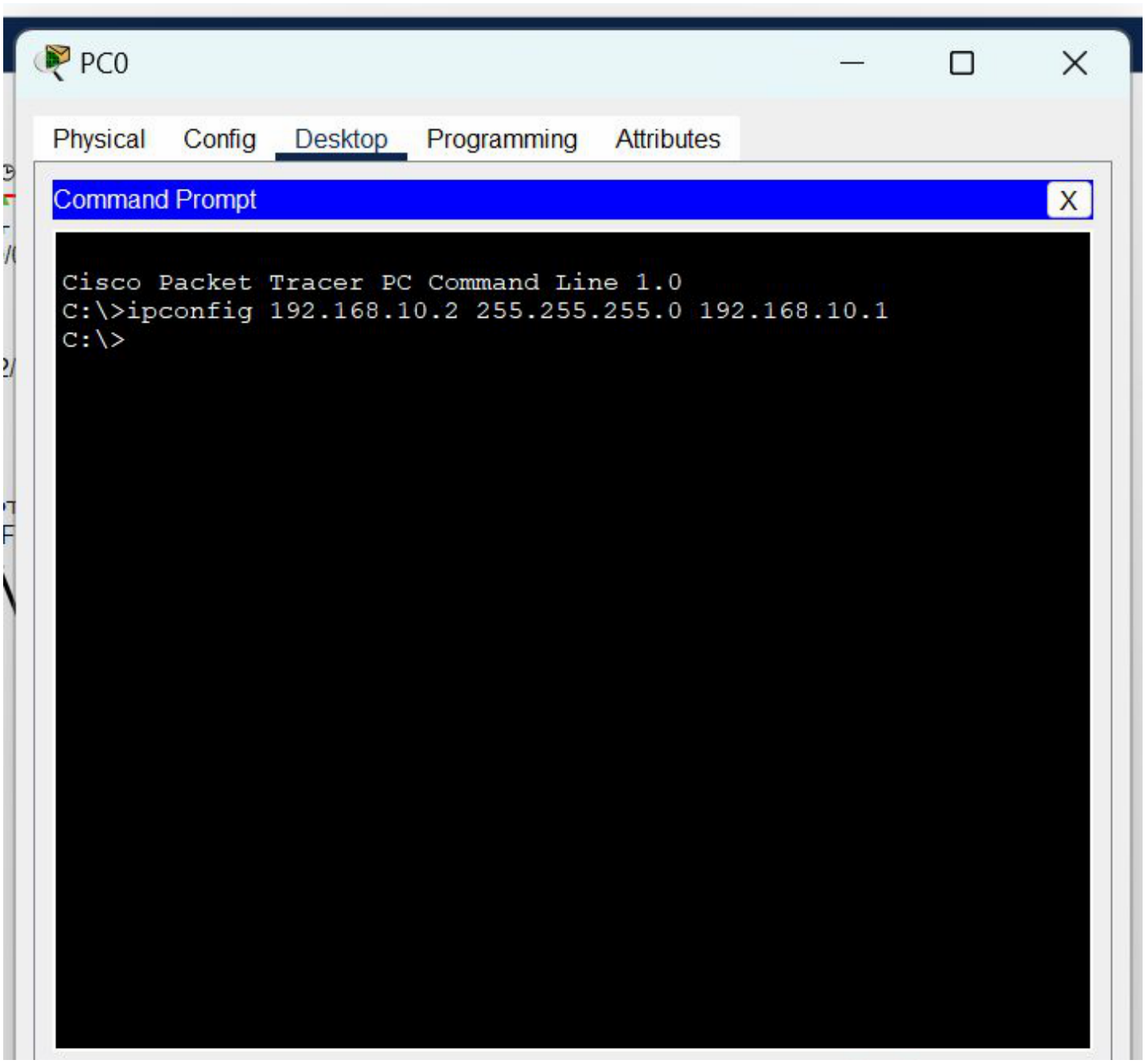
Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.
-



- Assigning an IP address using the ipconfig command, or we can also assign an IP address with the help of a command.
- Go to the command terminal of the PC.
- Then, type iPConfig <IPv4 address><subnet mask><default gateway>(if needed)

Example: iPConfig 192.168.10.2 255.255.255.0 192.168.10.1



- Repeat the same procedure with other PCs to configure them thoroughly.

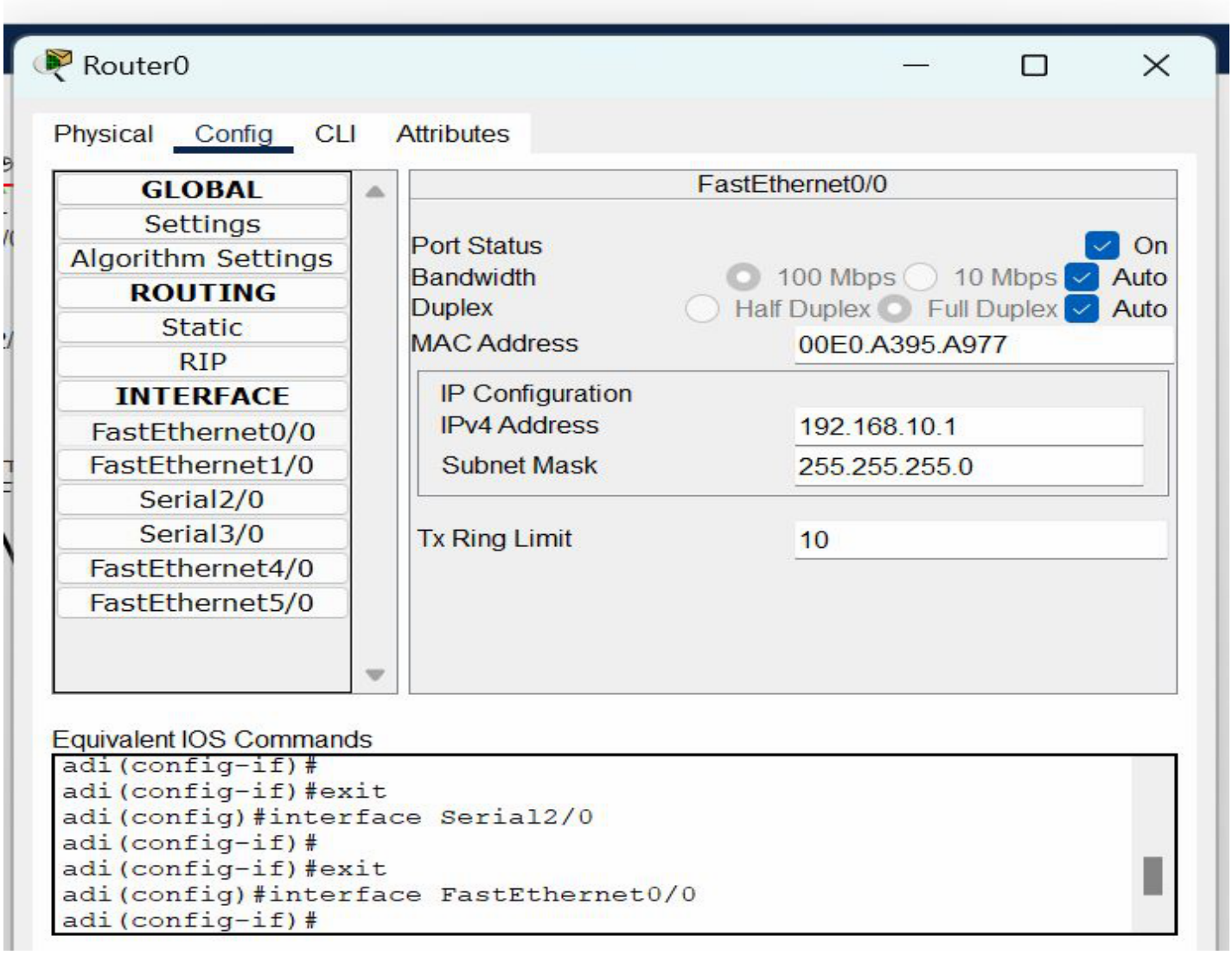
Step 3: Configure router with IP address and Subnet mask.

IP Addressing Table Router:

S.NO	Device	Interface	IPv4 Address	Subnet mask

1.	router 0	FastEthernet0/0	192.168.10.1	255.255.255.0
		Serial2/0	10.0.0.1	255.0.0.0
2.	router 1	FastEthernet0/0	192.168.20.1	255.255.255.0
		Serial2/0	10.0.0.2	255.0.0.0
		Serial3/0	11.0.0.1	255.0.0.0
3.	router 2	FastEthernet0/0	192.168.30.1	255.255.255.0
		Serial2/0	11.0.0.2	255.0.0.0

- To assign an IP address in router0, click on router0.
- Then, go to config and then Interfaces.
- Make sure to turn on the ports.
- Then, configure the IP address in FastEthernet and serial ports according to IP addressing Table.
- Fill IPv4 address and subnet mask.



- Repeat the same procedure with other routers to configure them thoroughly.

Step 4: After configuring all of the devices we need to assign the routes to the routers.

To assign RIP routes to the particular router:

- First, click on router0 then Go to CLI.
- Then type the commands and IP information given below.

CLI command : router rip
CLI command : network <network id>

RIP Routes for Router0 are given below:

Router(config)#router rip
Router(config-router)#network 192.168.10.0
Router(config-router)#network 10.0.0.0

RIP Routes for Router1 are given below:

Router(config)#router rip
Router(config-router)#network 192.168.20.0
Router(config-router)#network 10.0.0.0
Router(config-router)#network 11.0.0.0

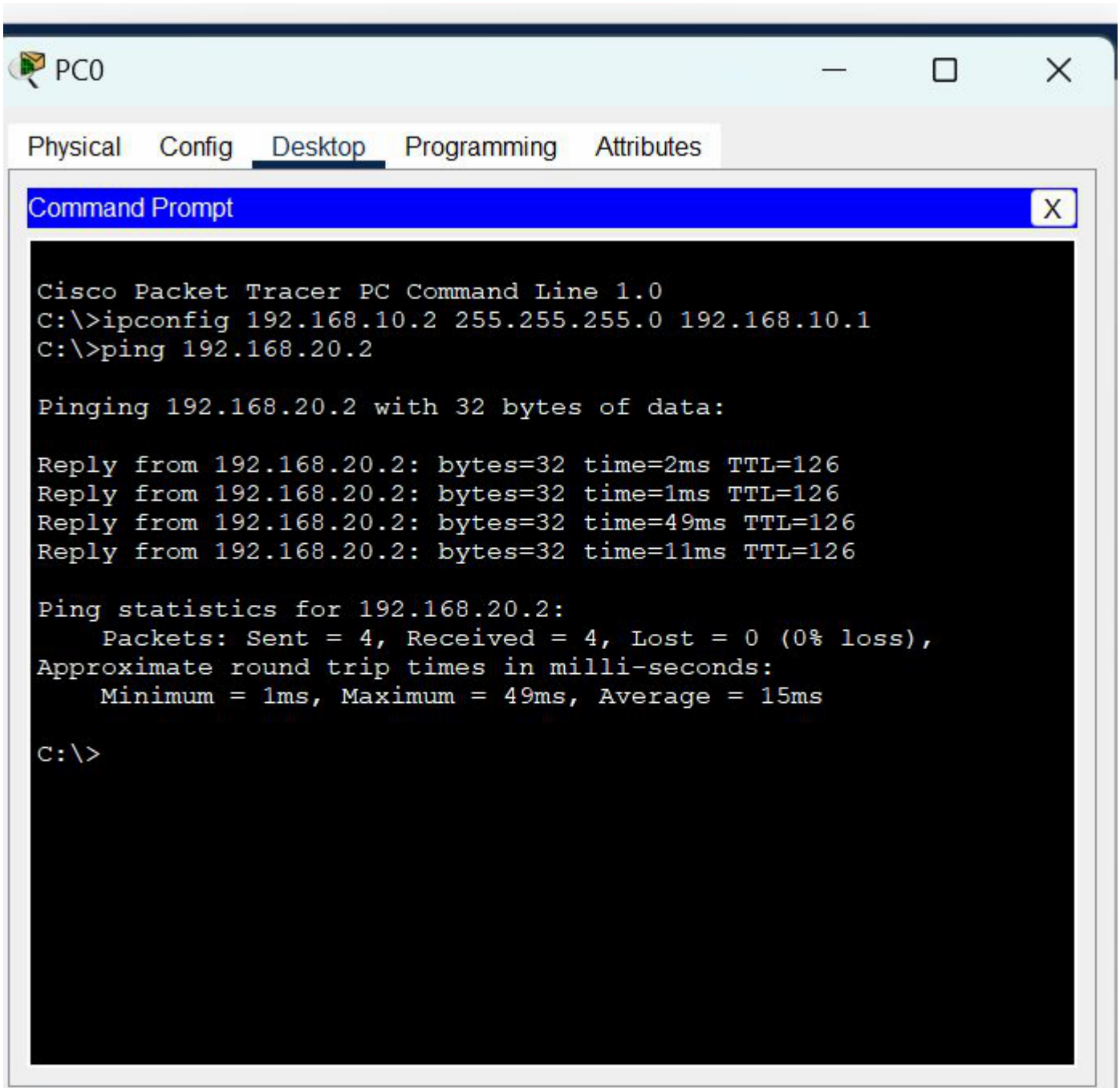
RIP Routes for Router2 are given below:

Router(config)#router rip
Router(config-router)#network 192.168.30.0
Router(config-router)#network 11.0.0.0

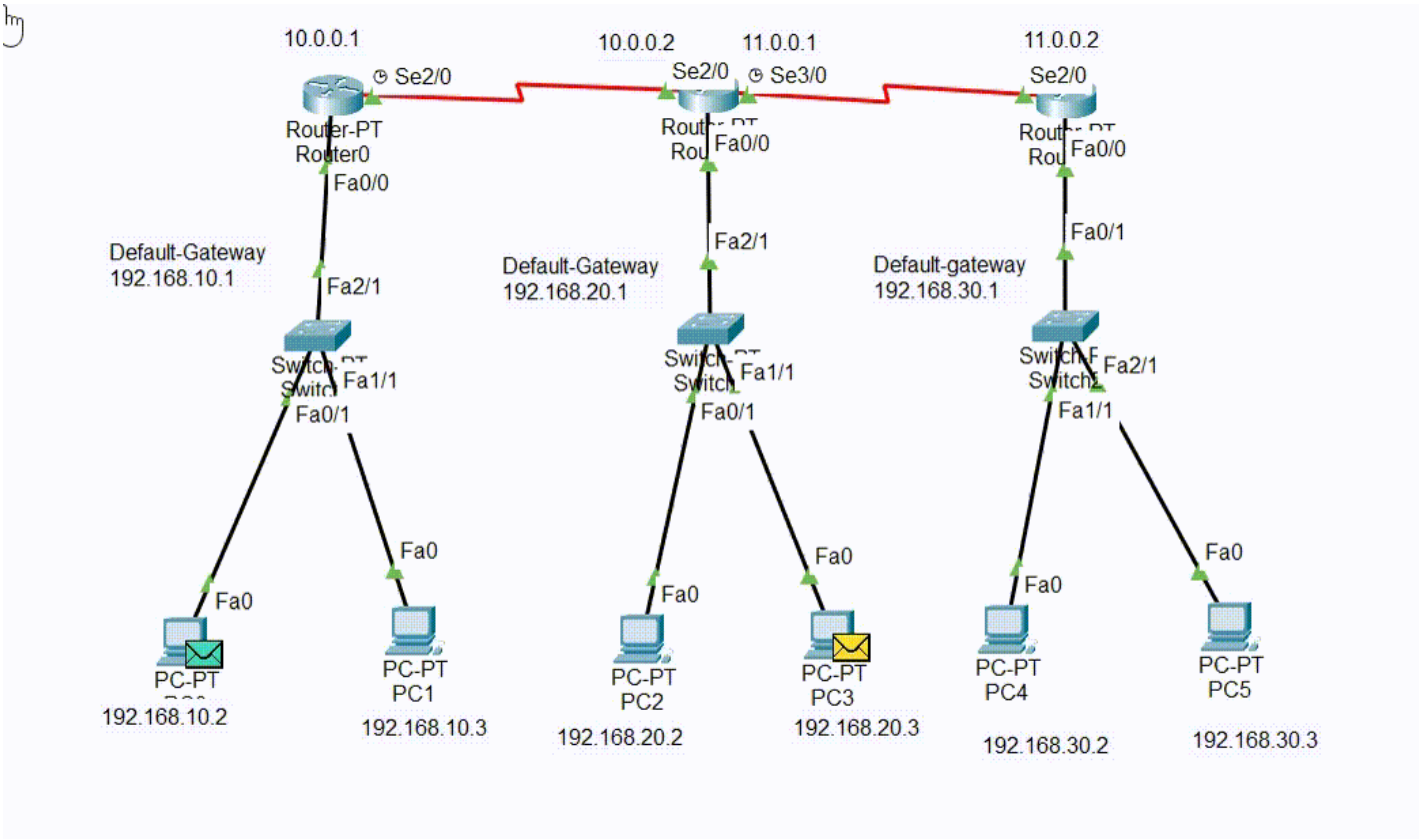
Step 5: Verifying the network by pinging the IP address of any PC.

- We will use the ping command to do so.
- First, click on PC0 then Go to the command prompt.
- Then type ping <IP address of targeted node>.
- As we can see in the below image we are getting replies which means the connection is working properly.

Example : ping 192.168.20.2



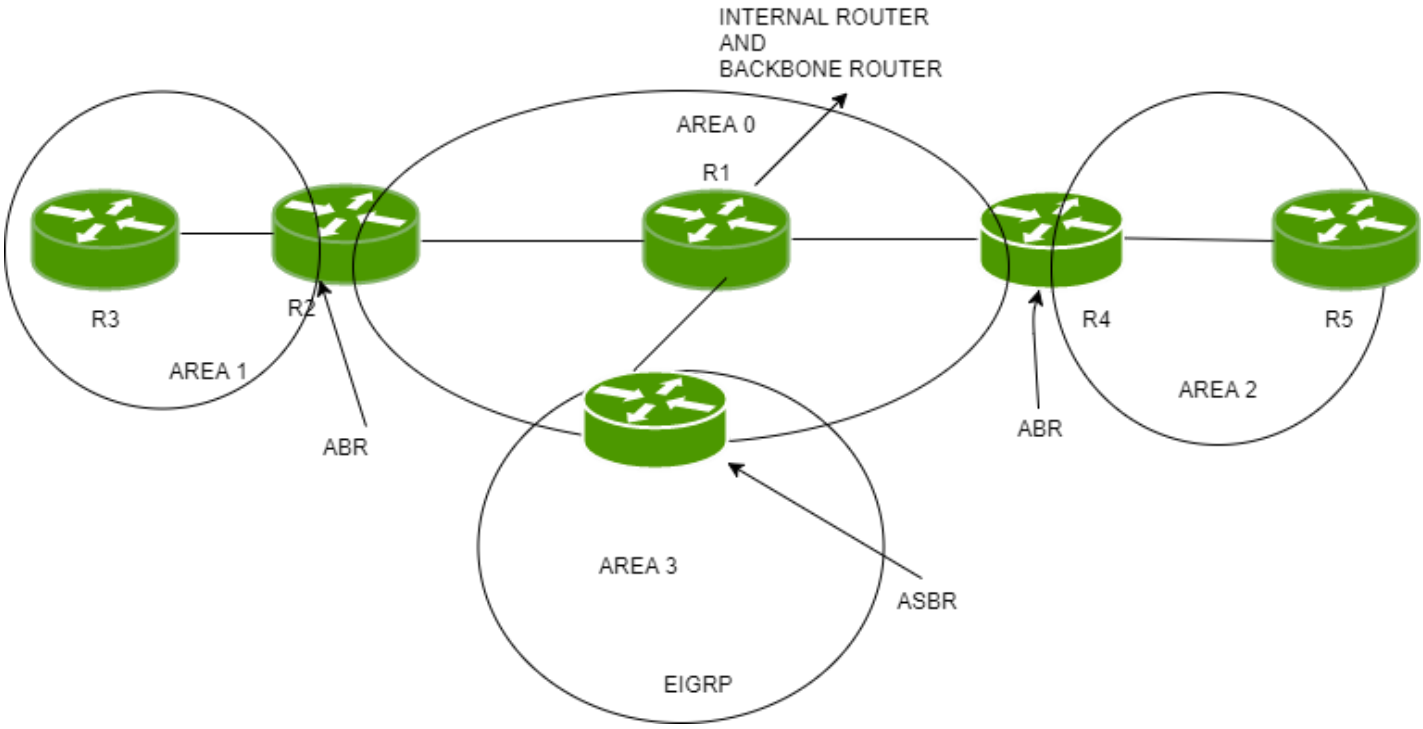
- A simulation of the experiment is given below we are sending PDU from PC0 to PC2 and PC3 to PC5:



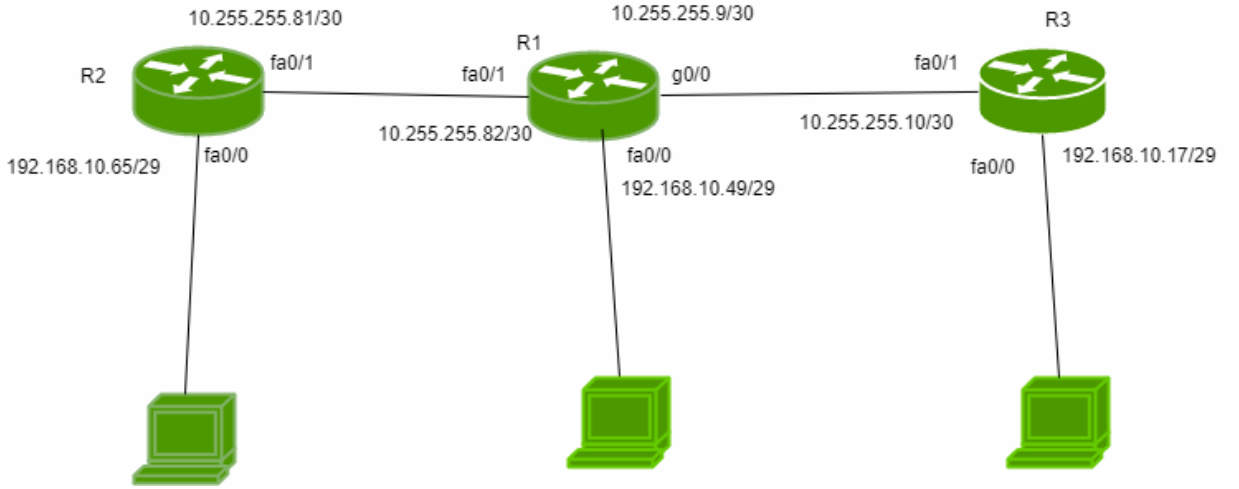
Open shortest path first (OSPF) router roles and configuration

Open shortest path first (OSPF) is a link-state routing protocol which is used to find the best path between the source and the destination router using its own SPF algorithm.

Open shortest path first (OSPF) router roles –
An area is a group of contiguous networks and routers. Routers belonging to same area shares a common topology table and area I'd. The area I'd is associated with router's interface as a router can belong to more than one area. There are some roles of router in OSPF:



1. **Backbone router –** The area 0 is known as backbone area and the routers in area 0 are known as backbone routers. If the routers exists partially in the area 0then also it is a backbone router.
2. **Internal router –** An internal router is a router which have all of its interfaces in a single area.
3. **Area Boundary Router (ABR) –** The router which connects backbone area with another area is called Area Boundary Router. It belongs to more than one area. The ABRs, therefore, maintain multiple link-state databases that describe both the backbone topology and the topology of the other areas.
4. **Area Summary Border Router (ASBR) –** When an OSPF router is connected to a different protocol like EIGRP or Border Gateway Protocol, or any other routing protocol then it is known as AS. The router which connects two different AS (in which one of the interfaces is operating OSPF) is known as Area Summary Border Router. These routers perform redistribution. ASBRs run both OSPF and another routing protocol, such as RIP or BGP. ASBRs advertise the exchanged external routing information throughout their AS.
5. **Configuration –**



There is a small topology in which there are 3 routers namely R1, R2, R3 are connected. R1 is connected to networks 10.255.255.80/30 (interface fa0/1), 192.168.10.48/29 (interface fa0/0) and 10.255.255.8/30 (interface gi0/0)
Note – In the figure, IP addresses are written with their respected interfaces but as have to advertise networks therefore, you have to write network I'd. R2 is connected to networks 192.168.10.64/29 (interface fa0/0), 10.255.255.80/30 (interface fa0/1). R3 is connected to networks 10.255.255.8/30 (int fa0/1), 192.168.10.16/29 (int fa0/0).

Now, configuring OSPF for R1.

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.10.48 0.0.0.7 area 1
R1(config-router)#network 10.255.255.80 0.0.0.3 area 1
R1(config-router)#network 10.255.255.8 0.0.0.3 area 1
```

Here, 1 is the OSPF instance or process I'd. It can be same or different (doesn't matter). You have use wildcard mask here and area used is 1.
Now, configuring R2

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.10.64 0.0.0.7 area 1
R2(config-router)#network 10.255.255.80 0.0.0.3 area 1
```

Similarly, for R3

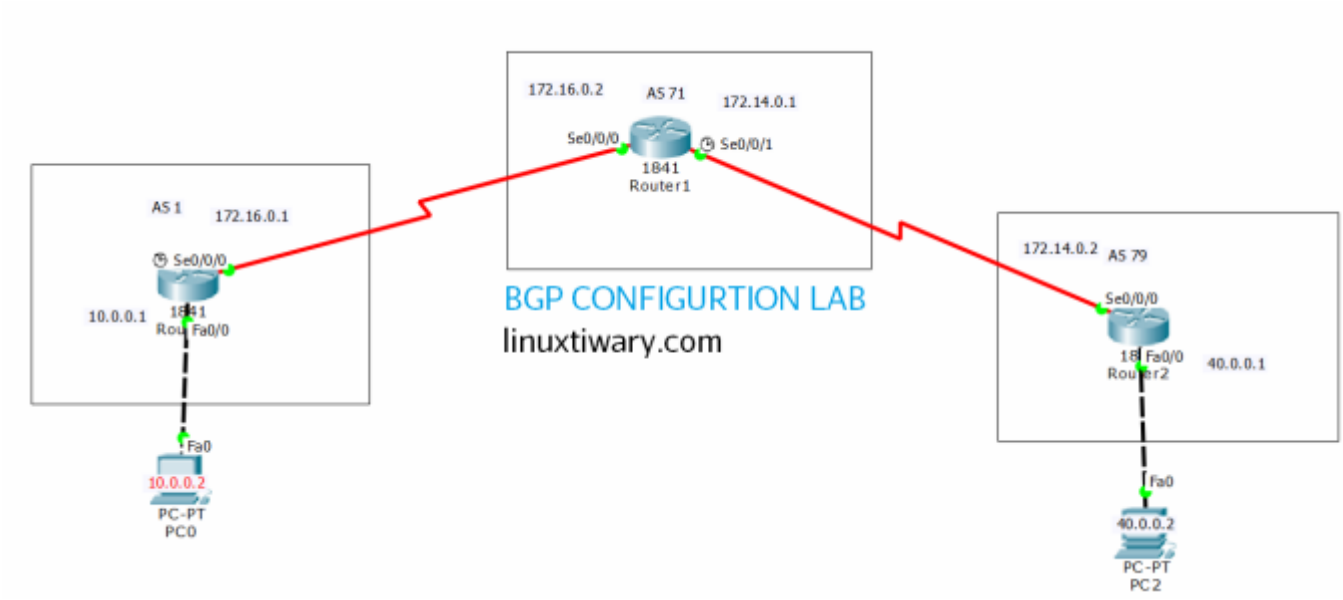
```
R3(config)#router ospf 1
R3(config-router)#network 192.168.10.16 0.0.0.7 area 1
R3(config-router)#network 10.255.255.8 0.0.0.3 area 1
```

You can check the configuration by typing command

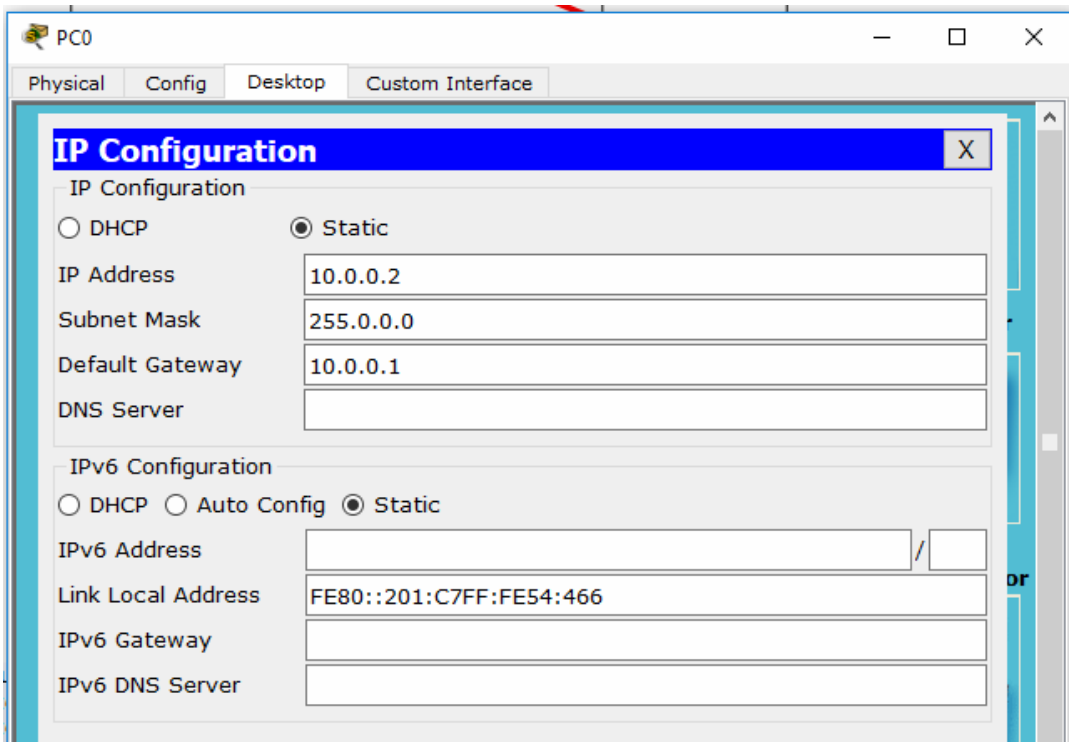
```
R3#show ip protocols
```

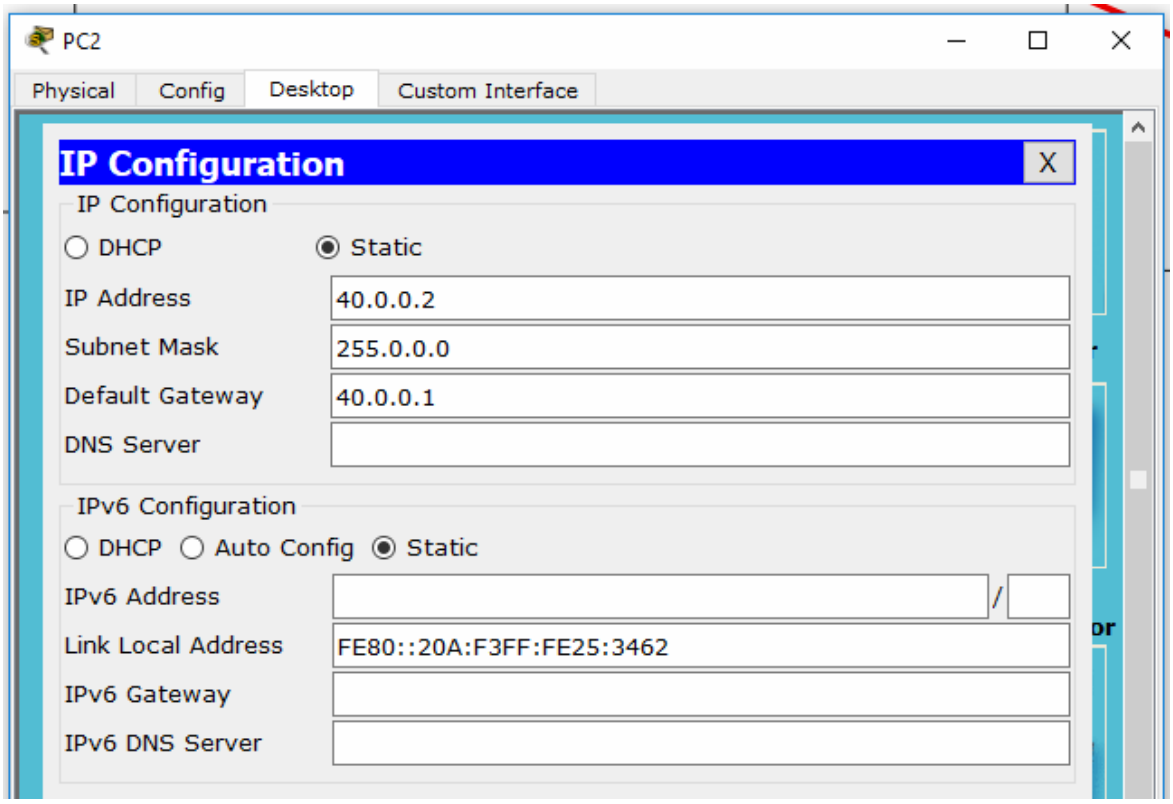
BGP configuration lab using cisco packet tracer
is an example lab showing bgp configuration.Here in This Lab I have taken 3 AS they are 1,71 and 79 respectively

Step 1:Draw BGP Topology Diagram.



Step 2: Assign ip address on each device as mentioned in Diagram.





Step 3: bgp configuration on Router R1:

```
R1(config)#router bgp 1
R1(config-router)#neighbor 172.16.0.2 remote-as 71
R1(config-router)#network 10.0.0.0 mask 255.0.0.0
R1(config-router)#exit
R1(config)#do write
Building configuration...[OK]
R1(config)#
```

Step 4: bgp configuration on Router R2:

```
R2(config)#router bgp 71
R2(config-router)#neighbor 172.16.0.1 remote-as 1
R2(config-router)#neighbor 172.14.0.2 remote-as 79
R2(config-router)#network 40.0.0.0 mask 255.0.0.0
R2(config-router)#exit
R2(config)#do write
Building configuration...[OK]
R2(config)#
```

Step 5: bgp configuration on Router R3:

```
R3(config)#router bgp 79
R3(config-router)#neighbor 172.14.0.1 remote-as 71
R3(config-router)#network 40.0.0.0 mask 255.0.0.0
R3(config-router)#exit
```

R3(config)#do write
Building configuration...[OK
R3(config)#

Step 6: bgp configuration Testing and troubleshooting.

For bgp testing we will ping both pc and check the network communication.

Now I am on PC2:

PC>ipconfig

PC>ping 10.0.0.2

Step 7: check bgp route on router R1:

R1#show ip route

Step 8: Check whether bgp protocols configure or not on Routre R1:

R1#show ip protocols

Step 9: Show BGP Status

R1#show ip bgp summary

Conclusion: We have successfully implemented RIP/OSPF/BGP on packet tracer.

Assignment No.	8
Title	**Write a program using TCP socket or UDP socket**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 08

Problem Definition: Write a program using TCP socket for wired network for following

a. Say Hello to Each other
b. File transfer
c. Calculator (Arithmetic)
d. Calculator (Trigonometry)

OR

Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machine

Objectives:

- Set up connection TCP between two nodes .
- a. Say Hello to Each other
- b. File transfer
- c. Calculator (Arithmetic)
- d. Calculator (Trigonometry)

Theory:

Socket Programming:

The Berkeley socket interface, an API, allows communications between hosts or between processes on one computer, using the concept of a socket. It can work with many different I/O devices and drivers, although support for these depends on the operating system implementation. This interface implementation is implicit for TCP/IP, and it is therefore one of the fundamental technologies underlying the Internet. It was first developed at the University of California, Berkeley for use on Unix systems. All modern operating systems now have some implementation of the Berkeley socket interface, as it has become the standard interface for connecting to the Internet. Programmers can make the socket interfaces accessible at three different levels, most powerfully and fundamentally at the RAW socket level. Very few applications need the degree of control over outgoing communications that this provides, so RAW sockets support was intended

to be available only on computers used for developing Internet related technologies.

TCP

TCP provides the concept of a connection. A process creates a TCP socket by calling the socket() function with the parameters PF_INET or PF_INET6 and SOCK_STREAM.

Server

Setting up a simple TCP server involves the following steps: Creating a TCP socket, with a call to socket().

Binding the socket to the listen port, with a call to bind(). Before calling bind(), aprogrammer must declare a sockaddr_in structure, clear it (with bzero() ormemset()), and the sin_family (AF_INET or AF_INET6), and fill its sin_port (the listening port, in network byte order) fields. Converting a short int to networkbyte order can be done by calling the

function htons() (host to network short).

Preparing the socket to listen for connections (making it a listening socket), with acall to listen().

Accepting incoming connections, via a call to accept(). This blocks until anincoming connection is received, and then returns a socket descriptor for theaccepted connection. The initial descriptor remains a listening descriptor, andaccept() can be called again at

any time with this socket, until it is closed.

Communicating with the remote host, which can be done through send() andrecv(). Eventually closing each socket that was opened, once it is no longer needed, using close().

Note that if there were any calls to fork(), each process must close the sockets it knew about (the kernel keeps track of how many processes have a descriptor open), and two processes should not use the same socket at once.

Client:

Setting up a TCP client involves the following steps:

1. Creating a TCP socket, with a call to socket().
2. Connecting to the server with the use of connect, passing a sockaddr_in structurewith the sin_family set to AF_INET or AF_INET6, sin_port set to the port theendpoint is listening(in network byte order), and sin_addr set to the IPv4 or IPv6address of the listening

   server (also in network byte order.)

1. Communicating with the server by send()ing and recv()ing.Terminating the connectionand cleaning up with a call to close(). Again, if therewere any calls to fork(), each processmust close() the socket.

Functions:

1. **socket():**

   socket() creates an endpoint for communication and returns a descriptor. socket() takesthree arguments:

- domain, which specifies the protocol family of the created socket. For example:
- PF_INET for network protocol IPv4 or
- PF_INET6 for IPv6).
- type, one of:
- SOCK_STREAM (reliable stream-oriented service)
- SOCK_DGRAM (datagram service)
- SOCK_SEQPACKET (reliable sequenced packet service), or
- SOCK_RAW (raw protocols atop the network layer).protocol

usually set to 0 to represent the default transport protocol for the specified domain and type values (TCP for PF_INET or PF_INET6 andSOCK_STREAM, UDP for those PF_ values andSOCK_DGRAM), but whichcan also explicitly specify a protocol.

The function returns -1 if an error occurred. Otherwise, it returns an integer representing the newly-assigned descriptor.

Prototype:

int socket(int domain, int type, int protocol);connect():

connect() returns an integer representing the error code: 0 represents success, while -1 represents an error. Certain types of sockets are connectionless, most commonly user datagram protocol sockets. For these sockets, connect takes on a special meaning: the default target for sending and receiving data gets set to the given address, allowing the use of functions such as send() and recv() on connectionless sockets.

Prototype:

int connect(intsockfd, conststructsockaddr *serv_addr, socklen_taddrlen);bind():

bind() assigns a socket an address. When a socket is created using socket(), it is given an address family, but not assigned an address. Before a socket may accept incoming connections, it must bebound. bind() takes three arguments:

➢ sockfd, a descriptor representing the socket to perform the bind onmy_addr, a pointer to a sockaddr structure representing the address to bind to.

➢ addrlen, a socklen_t field representing the length of the sockaddr structure.It returns 0 on success and -1 if an error occurs.

Prototype:

int bind(intsockfd, structsockaddr *my_addr, socklen_taddrlen);listen()

listen() prepares a bound socket to accept incoming connections. This function is only applicable

to the SOCK_STREAM and SOCK_SEQPACKET socket types. It takes two arguments:

- ➢ sockfd, a valid socket descriptor.

backlog, an integer representing the number of pending connections that can bequeued up at any one time. The operating system usually places a cap on this value.

Once a connection is accepted, it is dequeued. On success, 0 is returned. If an error occurs, -1 is returned.

Prototype:

int listen(intsockfd, int backlog);accept()

Programmers use accept() to accept a connection request from a remote host. It takes the following arguments:

- ➢ sockfd, the descriptor of the listening socket to accept the connection from.
- ➢ cliaddr, a pointer to the sockaddr structure that accept() should put the client'saddress information into.
- ➢ addrlen, a pointer to the socklen_t integer that will indicate to accept() how largethe sockaddr structure pointed to by cliaddr is. When accept() returns, the ocklen_t integer then indicates how many bytes of the cliaddr structure wereactually used.
- ➢ The function returns a socket corresponding to the accepted connection, or -1 if an erroroccurs.

Prototype:

int accept(intsockfd, structsockaddr *cliaddr, socklen_t *addrlen);

Blocking vs. nonblocking

Berkeley sockets can operate in one of two modes: blocking or non-blocking. A blocking socket will not "return" until it has sent (or received) all the data specified for the operation. This may cause problems if a socket continues to listen: a program may hang as the socket waits for data that may never arrive. A socket is typically set to blocking or nonblocking mode using the fcntl() or ioctl() functions.

Cleaning up

The system will not release the resources allocated by the socket() call until a close() call occurs. This is especially important if the connect() call fails and may be retried. Each call to socket() must have a matching call to close() in all possible execution paths.

Algorithm:

Server Program

1. Open the Server Socket:

ServerSocket server = new ServerSocket(PORT);

2. Wait for the Client Request:

   Socket client = server.accept();

3. Create I/O streams for communicating to the client

   DataInputStream is = new DataInputStream(client.getInputStream());

DataOutputStreamos = new DataOutputStream(client.getOutputStream());

4. Perform communication with client

   Receive from client: String line =

   is.readLine();Send to client:

   os.writeBytes("Hello\n")

5. Close socket:

client.close();

Client Program

1) Create a Socket Object:

   Socket client = new Socket(server, port_id);

2) Create I/O streams for communicating with the server.

   is = new DataInputStream(client.getInputStream());

   os = new DataOutputStream(client.getOutputStream());

3) Perform I/O or communication with the server:

   Receive data from the server: String line =

   is.readLine();Send data to the server:

   os.writeBytes("Hello\n");

4) Close the socket when done:

5) client.close();

Questions:

1. What is a socket?
2. What is the difference between a connection-less and a connection-orientedcommunication system? How are they implemented?
3. Which of them is more reliable?
4. Why is the port number required?
5. How is the socket programming in linux different from that in windows?

UDP Socket Programming:

- UDP is normally used for real-time applications that cannot tolerate uneven delay between sections of a received message.

Algorithm:

Server Program

1. Open the Server Socket:
   ServerSocket server = new ServerSocket(PORT);
2. Wait for the Client Request:
   Socket client = server.accept();
3. Create I/O streams for communicating to the clientDataInputStream is =
   newDataInputStream(client.getInputStream());
   DataOutputStreamos = new DataOutputStream(client.getOutputStream());
4. Perform communication with client
   Receive from client: String line = is.readLine();
   Send to client: os.writeBytes("Hello\n")
5. Close socket:
   client.close();

Client Program

1. Create a Socket Object:
   Socket client = new Socket(server, port_id);
2. Create I/O streams for communicating with the server.
   is = new DataInputStream(client.getInputStream());
   os = new DataOutputStream(client.getOutputStream());
3. Perform I/O or communication with the server:
   Receive data from the server: String line =
   is.readLine();Send data to the server:
   os.writeBytes("Hello\n");
4. Close the socket when done:

5. client.close();

CONCLUSION

Thus we have successfully implemented the socket programming for TCP and UDP using C++/Java.

Assignment No.	9
Title	**Write a program for DNS lookup**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 09

Problem Definition : Write a program for DNS lookup. Given an IP address input, it should return URL and vice versa**.**

Objective:

- ➢ To get the host name and IP address.
- ➢ Map the host name with IP address and Vice-versa

Theory:

Need for DNS:

To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name. When the Internet was small, mapping was done using a host file. The host file had only two columns: name and address. Every host could store the host file on its disk and update it periodically from a master host file. When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

Today, however, it is impossible to have one single host file to relate every address with a name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there is a change. One solution would be to store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping. But we know that this would create a huge amount of traffic on the Internet. Another solution, the one used today, is to divide this huge amount of information into smaller parts and store each part on a different computer. In this method, the host that needs mapping can contact the closest computer holding the needed information. This method is used by the Domain Name System (DNS).

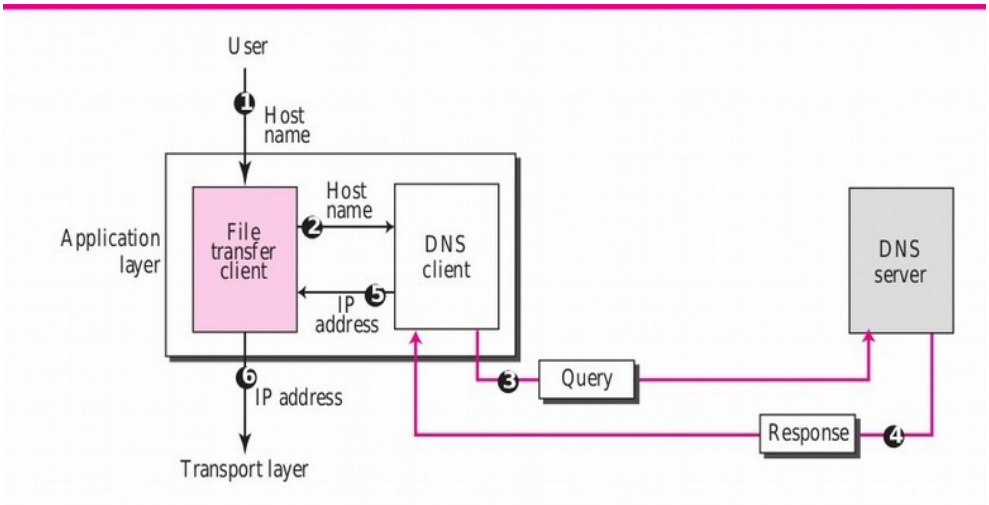


Figure 1: How TCP/IP uses a DNS client and a DNS server to map a name to an address; the reverse mapping is similar.

In Figure 1 , a user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as forouzan.com. However, the TCP/IP suite needs the IP address of the file transfer server to make the connection. The following six steps map the host name to an IP address.

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. We know that each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

NAME SPACE:

To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. In other words, the names must be unique because the addresses are unique.

A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

Flat Name Space:

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it

has no meaning. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space;

In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

Domain Name Space:

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127 (see Figure 2).

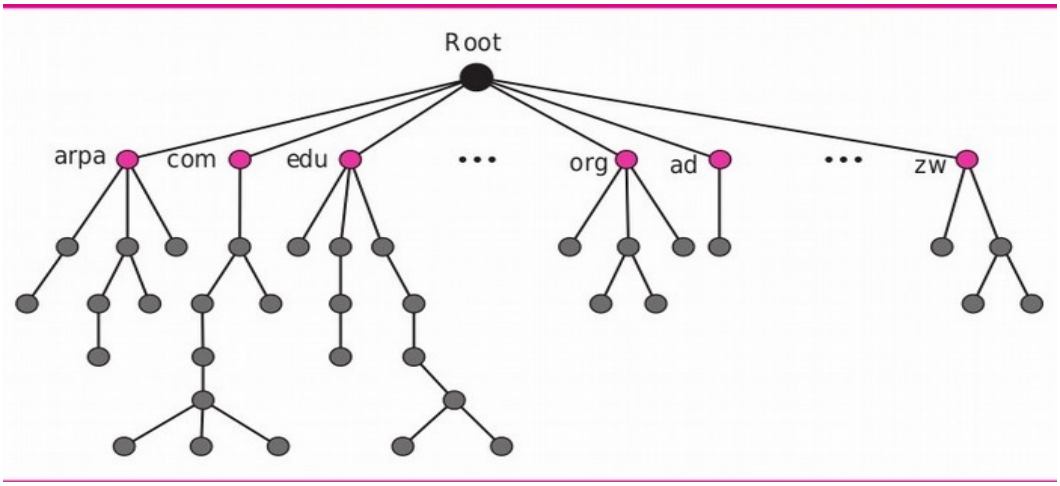


Figure 2 : Domain Name Space

Label:

Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string). DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name:

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the

label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

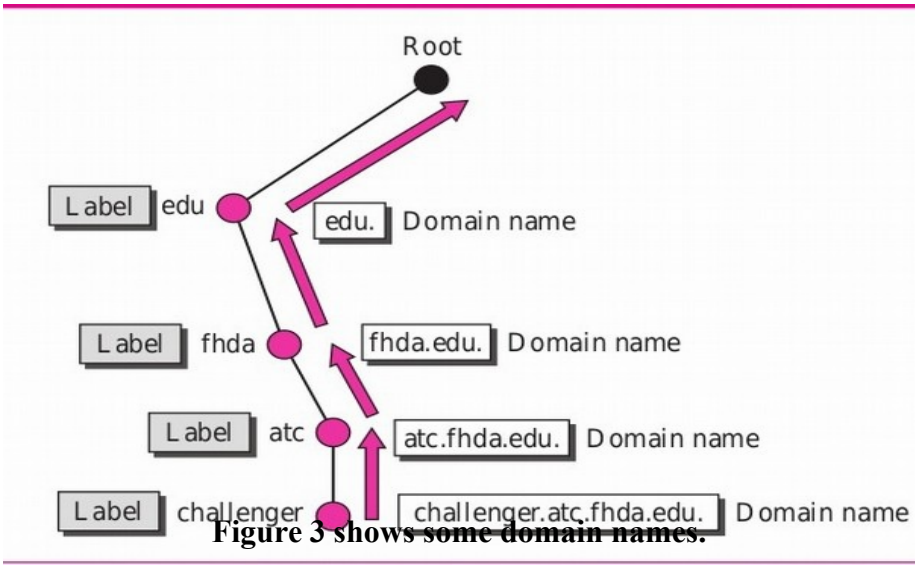


Figure 3 shows some domain names.

Domain:

A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. Figure 4 shows some domains. Note that a domain may itself be divided into domains (or subdomains as they are sometimes called).

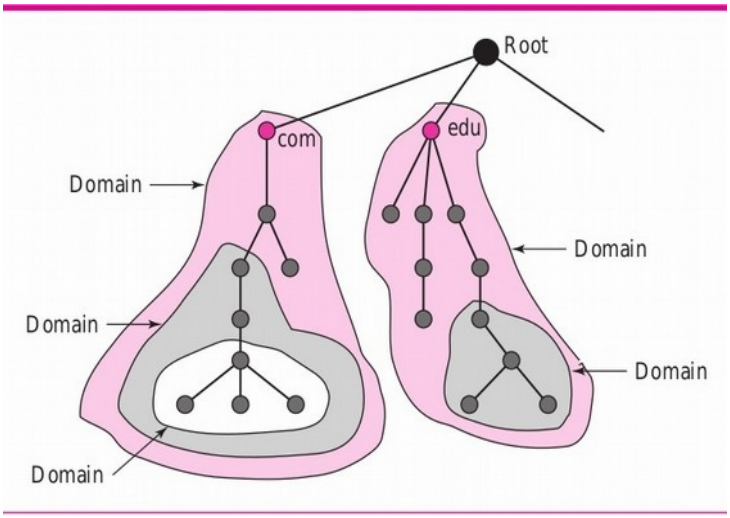


Figure 4 Domain

RESOLUTION:
Mapping a name to an address or an address to a name is called name-address resolution.

Resolver:
DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information. After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.

Mapping Names to Addresses:
Most of the time, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping. If the domain name is from the generic domains section, the resolver receives a domain name such as "chal.atc.fhda.edu.". The query is sent by the resolver to the local

DNS server for resolution:
If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly. If the domain name is from the country domains section, the resolver receives a domain name such as "ch.fhda.cu.ca.us.". The procedure is the same.Mapping Addresses to Names A client can send an IP address to a server to be mapped to a domain name. As mentioned before, this is called a PTR query. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and two labels, in-addr and arpa, are appended to create a domain acceptable by the inverse domain section. For example, if the resolver receives the IP address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending. The domain name sent is "121.45.34.132.in-addr.arpa.", which is received by the local DNS and resolved.

Recursive Resolution:
The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it

responds; otherwise, it sends the query to yet another server. When the query is finally resolved,the response travels back until it finally reaches the requesting client.

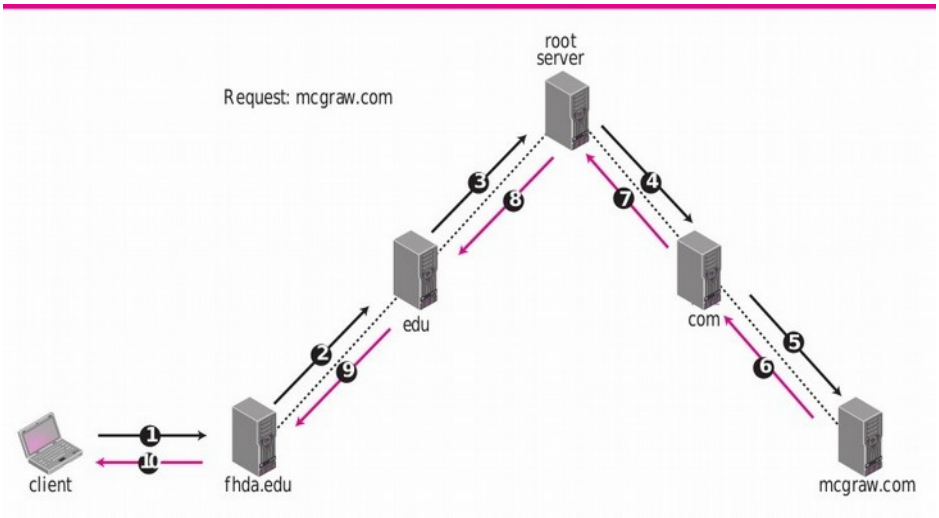


Figure 5: Recursive resolution.

Algorithm:

1. **Give the input as website address e.g.www.google.com.**
2. Get address, if first character is a digit then assume is an address
3. Convert address from string representation to byte array
4. Get Internet Address of this host address
5. Get Internet Address of this host name
6. Show the Internet Address as name/address
7. Print Host Name and Host Address
8. Get the default initial Directory Context
9. Get the DNS records for Address
10. Get an enumeration of the attributes and print them out#

javac DNSLookup.java
#java DNSLookup google.com
google.com/64.233.167.99
-- DNS INFORMATION --
A: 64.233.187.99, 72.14.207.99, 64.233.167.99
NS: ns4.google.com., ns1.google.com., ns2.google.com., ns3.google.com. MX:
10 smtp4.google.com., 10 smtp1.google.com., 10 smtp2.google.com., 10

Conclusion:

Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintaina directory of domain names and translate them to Internet Protocol (IP) addresses. Thus we have implemented DNS lookup

Assignment No.	10
Title	**Content Beyond Syllabus: Windows File sharing**
Subject	Computer Network Lab
Class	T.E. (Comp)
Roll No.	

Experiment Number: 10

Content Beyond Syllabus

AIM: Windows File sharing

OBJECTIVE: 1. To learn how to share folder from one computer to another.

PROBLEM STATEMENT: To share a folder from a computer and access the shared folder from another computer.

OUTCOME: Students will able to

1. illustrate windows file sharing.

THEORY-CONCEPT:

Computer Network is a connection of two or more devices that are connected through a medium in order to exchange information. With the help of a Computer Network, you can easily send or receive data to or from a computing device. It's a common situation that you have several computers near each other and you want to transfer files between them. You don't have to pull out a USB drive, nor do you have to send them over email. So, there are faster, easier ways.

This is easier than it was in the past, as you don't have to mess with any complicated Windows networking settings. There are lots of ways to share files, but we'll cover some of the best.

Windows Homegroup

Assuming the computers are using Windows 7 or Windows 8, a Windows Homegroup is one of the easiest ways to share files between them. Windows home networking has been extremely complicated to configure in the past, but Homegroup is easy to set up. Just create a Homegroup from the Homegroup option within Windows Explorer (File Explorer on Windows 8) and you'll get a password. Enter that password on nearby computers and they can join your Homegroup. They'll then have access to your shared files when they're on the same network — you can select the libraries you want to share while creating a Homegroup.

Procedure

1. The aim is to Share the files between Two Pc's.
2. To perform the experiment, follow the below steps
3. Click on the first Pc
4. Follow the step to share a folder from the given location

5. Close the Pc by clicking on the black bar
6. Click on second Pc
7. follow the step to access the files
8. Close the Pc
9. If all the steps are performed correctly then Experiment is successful
10. Else Need to do it again

CONCLUSION:

We have successfully studied windows file sharing.