AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
[I.O.I.T.]
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

Department of Artificial Intelligence & Data Science

Lab Manual

Computer Network Laboratory
(ADPCC508)

Prepared by
Ms. S. D. Kadu
Mrs. S. S. Sheikh

TY AI&DS

Semester V

Academic Year 2024-25

Department of Artificial Intelligence & Data Science

Vision

To be amongst the top 5% of Artificial Intelligence & Data Science programs for catering to the changing needs of the industry and society.

Mission

1. To impart AI&DS skills emphasizing interdisciplinary design, development and analysis.
2. To foster employability, entrepreneurial and research opportunities in providing intelligent solutions for the needs of industry and society.
3. To promote career development with ethical responsibility.

Program Education Objectives(PEOs)

PEO1: Graduates will be able to analyze, formulate and function efficiently in a multi-disciplinary context to address industrial problems.

PEO2: Graduates will be able to work collaboratively with professionalism and ethical responsibilities to provide innovative solutions to society.

PEO3: Graduates will excel in their careers by adapting to new technologies.

Program Specific Outcomes (PSOs)

PSO1 Problem Solving and Programming Skills: Graduates will be able to apply programming skill to identify, modify and test algorithms that apply intelligence to make realistic decisions in problem solving.

PSO2 Professional Skills: Graduates will be able to collect, analyze, interpret and visualize data to solve problems in agriculture, automation, finance and medical domains.

Program Outcomes (POs)

Graduates will be able to

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. **[Engineering knowledge]**
2. Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. **[Problem analysis]**
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. **[Design/development of solutions]**
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. **[Conduct investigations of complex problems]**
5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations. **[Modern tool usage]**
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. **[The engineer and society]**
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. **[Environment and sustainability]**
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. **[Ethics]**
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. **[Individual and team work]**
10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. **[Communication]**
11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. **[Project management and finance]**
12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. **[Life-long learning]**

Course Objectives:

1. To establish communication among the computing nodes in wired technologies.
2. To learn network , transport and application layer protocols in a simulated networking environment.
3. To Configure the computing nodes with understanding of protocols and topologies.
4. To learn and understand socket programming in network
5. To understand network simulation tool: packet tracer and Wireshark.

Course Outcomes:

On completion of the course, learner will be able to–

1. illustrate communication among different nodes using protocols and topologies.
2. implement network, transport and application layer protocols in a simulated networking environment.
3. implement the concept of socket programming in TCP and UDP.
4. apply network simulation tools: packet tracer and Wireshark.

INDEX

Sr. No.	Title of Experiment	CO	PO	PSO
1.	Bridge the gap (Basic Networking command)			
2.	Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.	CO1, CO4	1,2,3,5, 9	PSO2
3.	Write a program to demonstrate Sub-netting and find subnet masks.	CO1, CO4	1,2,3,5, 9	PSO2
4.	Use packet Tracer tool for configuration of 3 router networks using one of the following protocols RIP/OSPF/BGP.	CO2	1,2,3,5, 9	PSO1
5.	Write a program to implement link state routing protocol to find a suitable path for transmission.	CO2	1,2,3,5, 9	PSO1
6.	Socket Programming using python.	CO4	1,2,3,5, 9	PSO1
7.	Write a program using TCP socket for wired network for following File transfer	CO2, CO3	1,2,3,5, 9	PSO2
8.	Study and Analyze the performance of FTP protocol using Packet tracer tool.	CO2, CO3	1,2,3,5, 9	PSO2
9.	Write a program for DNS lookup. Given an IP address input, it should return URL and vice versa.	CO2, CO3	1,2,3,5, 9	PSO2
10.	To share a folder from a computer and access the shared folder from another computer.	CO3, CO4	1,2,3,5, 9	PSO2

Bridge the Gap -1

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: 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.

DESIGNING OF TOPOLOGY

1. Bus Topology In local area network, it is a single network cable runs in the building or campus and all nodes are connected along with this communication line with two endpoints

called the bus or backbone. In other words, it is a multipoint data communication circuit that is easily control data flow between the computers because this configuration allows all stations to receive every transmission over the network. For bus topology we build network using three generic pc which are serially connected with three switches using copper straight through cable and switches are interconnected using copper cross over cable shown in fig1.

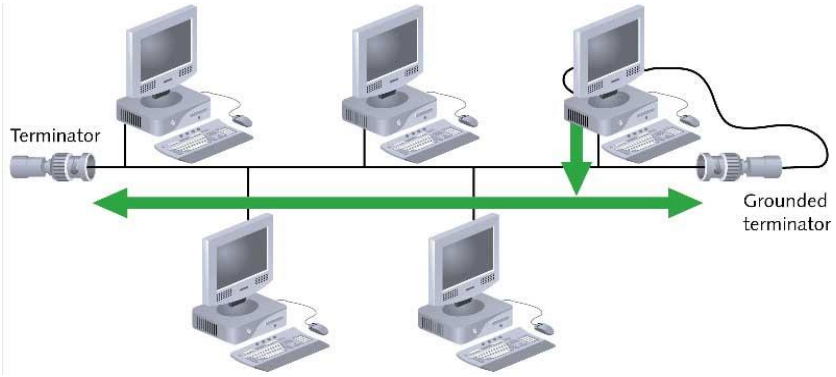


Fig1. Bus topology

2. Star Topology, all the cables run from the computers to a central location where they are all connected by a device called a hub. It is a concentrated network, where the end points are directly reachable from a central location when network is expanded. Ethernet 10 base T is a popular network based on the star topology. For star topology we build network using five generic pc which are centrally connected to single switch 2950-24 using copper straight through cable shown in fig2.

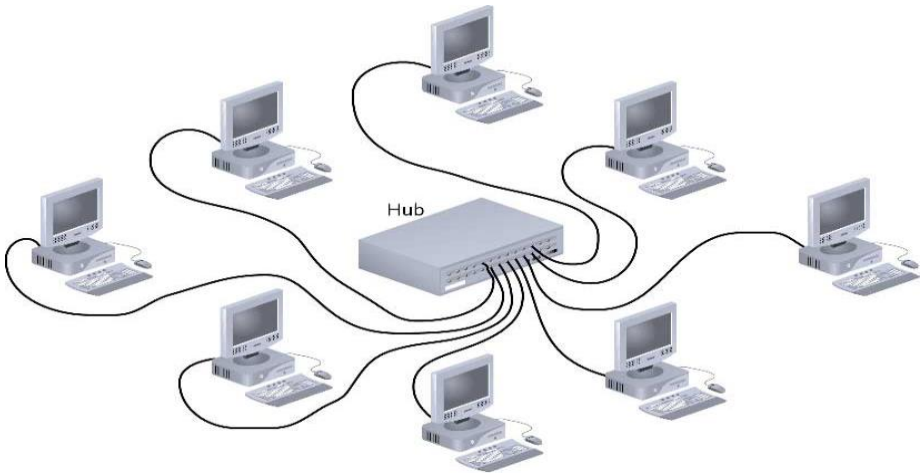


Fig 2. Star Topology

3. Mesh Topology, every device has a dedicated point to point link to every other device. The term dedicated stand for link carries traffic only between two devices it connects. It is a well-connected topology; As shown in fig3. , in this every node has a connection to every other node in the network. The cable requirements are high and it can include multiple topologies. Failure in one of the computers does not cause the network to break down, as they have alternative paths to other computers star topology, all the cables run from the computers to a central location.

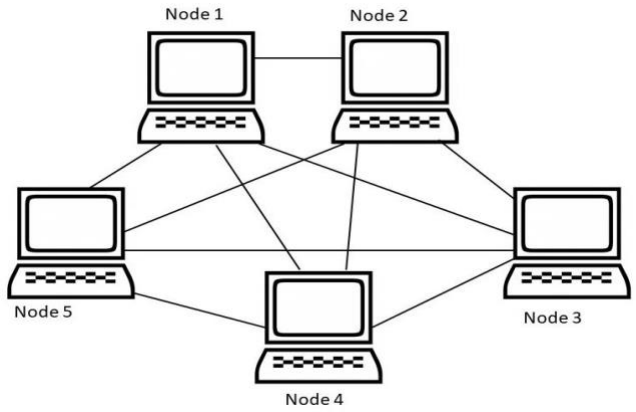


Fig 3. Mesh Topology

TRANSMISSION MEDIA

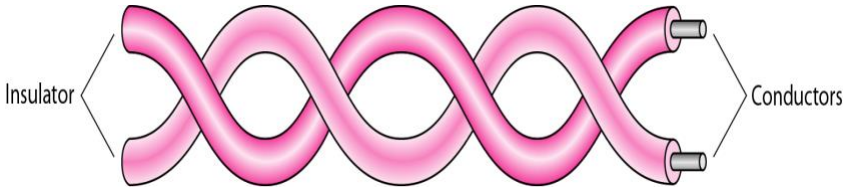
- Transmission media is a communication channel that carries the information from the sender to the receiver. Data is transmitted through the electromagnetic signals. The main functionality of the transmission media is to carry the information in the form of bits through LAN (Local Area Network). It is a physical path between transmitter and receiver in data communication. In a copper-based network, the bits in the form of electrical signals. In a fiber-based network, the bits in the form of light pulses. In OSI(Open System Interconnection) phase, transmission media supports the Layer 1. Therefore, it is considered to be as a Layer 1 component. The electrical signals can be sent through the copper wire, fiber optics, atmosphere, water, and vacuum. The characteristics and quality of data transmission are determined by the characteristics of medium and signal. Transmission media is of two types are wired media and wireless media. In wired media, medium characteristics are more important whereas, in wireless media, signal characteristics are more important. Different transmission media have different properties such as bandwidth, delay, cost and

ease of installation and maintenance. The transmission media is available in the lowest layer of the OSI reference model, i.e., Physical layer.Transmission Media is broadly classified into the following types:

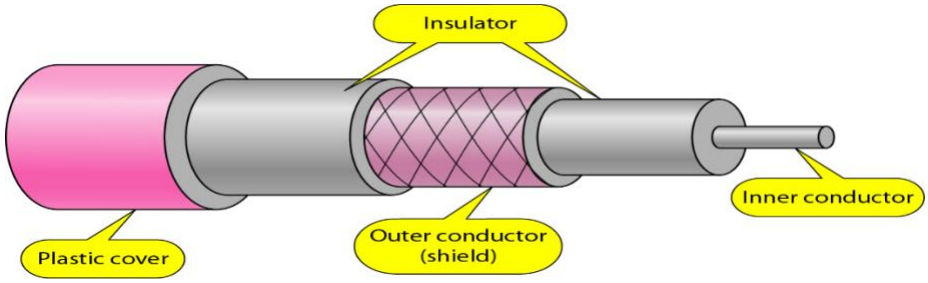
1. Guided
2. Unguided

1. Guided Transmission media: Transmission capacity depends on the distance and on whether the medium is point-to-point or multipoint. There are three types of guided transmission media.

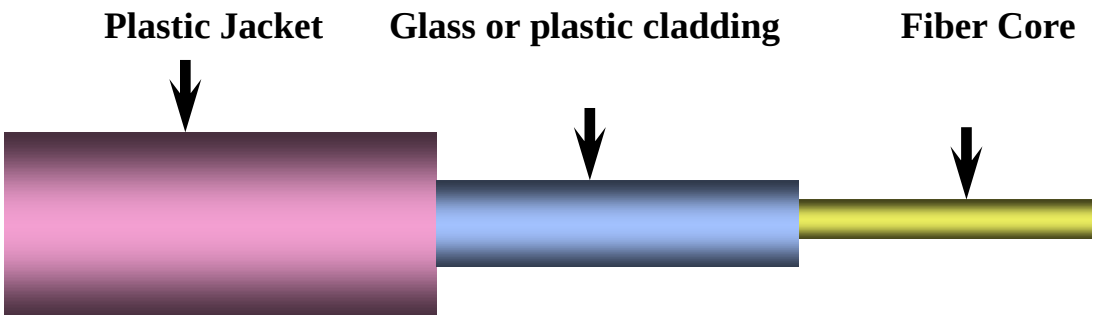
a) Twisted pair wires



b) Coaxial cables



c) Optical fiber



2. Unguided transmission Media: It is also referred to as Wireless or Unbounded transmission media. No physical medium is required for the transmission of electromagnetic signals.

   Features:

   - The signal is broadcasted through air
   - Less Secure
   - Used for larger distances

   There are 3 types of Signals transmitted through unguided media:

Radio waves – These are easy to generate and can penetrate through buildings. The sending and receiving antennas need not be aligned. Frequency Range:3KHz – 1GHz. AM and FM radios and cordless phones use Radio waves for transmission.

Microwaves – It is a line-of-sight transmission i.e., the sending and receiving antennas need to be properly aligned with each other. The distance covered by the signal is directly proportional to the height of the antenna. Frequency Range:1GHz – 300GHz. These are majorly used for mobile phone communication and television distribution.

Infrared – Infrared waves are used for very short distance communication. They cannot penetrate through obstacles. This prevents interference between systems. Frequency Range:300GHz – 400THz. It is used in TV remotes, wireless mouse, keyboard, printer, etc.

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

Assignment No: 3

TITLE: Implement subnetting and find the subnet masks.

OBJECTIVES: 1. To learn network address classes

1. To understand the concept of subnetting

PROBLEM STATEMENT: Write a program to demonstrate Sub-netting and find subnet masks.

OUTCOME: Students will be able to

1. memorize network address classes
2. implement subnetting and subnet masks

THEORY-CONCEPT:

SUBNETTING

Subnetting is a process of dividing large network into the smaller networks based on layer 3 IP address. Every computer on network has an IP address that represent its location on network. Two version of IP addresses are available IPv4 and IPv6. In this article we will perform subnetting on IPv4.

IPv4

IP addresses are displayed in dotted decimal notation, and appear as four numbers separated by dots. Each number of an IP address is made from eight individual bits known as octet. Each octet can create number value from 0 to 255. An IP address would be 32 bits long in binary divided into the two components, network component and host component. Network component is used to identify the network that the packet is intended for, and host component is used to identify the individual host on network. IP addresses are broken into the two components:

Network component: - Defines network segment of device.

Host component: - Defines the specific device on a particular network segment

IP Classes in decimal notation

1. Class A addresses range from 1-126
2. Class B addresses range from 128-191
3. Class C addresses range from 192-223
4. Class D addresses range from 224-239

5. Class E addresses range from 240-254

- 0 [Zero] is reserved and represents all IP addresses.
- 127 is a reserved address and is used for testing, like a loop back on an interface.
- 255 is a reserved address and is used for broadcasting purposes

SUBNET MASK

Subnet mask is a 32 bits long address used to distinguish between network address and host address in IP address. Subnet mask is always used with IP address. Subnet mask has only one purpose, to identify which part of an IP address is network address and which part is host address.

For example, how will we figure out network partition and host partition from IP address 192.168.1.10. Here we need subnet mask to get details about network address and host address.

- In decimal notation subnet mask value 1 to 255 represent network address and value 0 [Zero] represent host address.
- In binary notation subnet mask **ON** bit [1] represent network address while **OFF** bit [0]represent host address.

In decimal notation

IP address	192.168.1.10
Subnet mask	255.255.255.0

Network address is 192.168.1 and host address is 10.

In binary notation

IP address	11000000.10101000.00000001.00001010
Subnet mask	11111111.11111111.11111111.00000000

Network address is 11000000.10101000.00000001 and host address is 00001010

The following table shows IP class associated with default subnet

Table 1

IP Class	Default Subnet	Network bits	Host bits	Total hosts	Valid hosts
A	255.0.0.0	First 8 bits	Last 24 bits	16, 777, 216	16, 777, 214
B	255.255.0.0	First 16 bits	Last 16 bits	65,536	65,534
C	255.255.255.0	First 24 bits	Last 8 bits	256	254

Network ID

First address of subnet is called network ID. This address is used to identify one segment or broadcast domain from all the other segments in the network.

Block Size

Block size is the size of subnet including network address, hosts addresses and broadcast address.

Broadcast ID

There are two types of broadcast, direct broadcast and full broadcast.

Direct broadcast or local broadcast is the last address of subnet and can be heard by all hostsin subnet.

Full broadcast is the last address of IP classes and can be heard by all IP hosts in network. Full broadcast address is 255.255.255.255

The main difference between direct broadcast and full broadcast is that routers will not propagate local broadcasts between segments, but they will propagate directed broadcasts.

Host Addresses

All address between the network address and the directed broadcast address is called host address for the subnet. You can assign host addresses to any IP devices such as PCs, servers, routers, and switches.

Single class C IP range can fulfill this requirement, still you have to purchase 2 class C IP range, one for each network. Single class C range provides 256 total addresses and we need only 30 addresses, this will waste 226 addresses. These unused addresses would make additional route advertisements slowing down the network.

With subnetting you only need to purchase single range of class C. You can configure router to

take first 26 bits instead of default 24 bits as network bits. In this case we would extend default boundary of subnet mask and borrow 2 host bits to create networks. By taking two bits from the host range and counting them as network bits, we can create two new subnets, and assign hosts them. As long as the two new network bits match in the address, they belong to the same network. You can change either of the two bits, and you would be in a new subnet.

Default subnet mask

Class	**Subnet Mask**	**Format**
A	255.0.0.0	Network.Host.Host.Host
B	255.255.0.0	Network.Network.Host.Host
C	255.255.255.0	Network.Network.Network.Host

ADVANTAGE OF SUBNETTING

- Subnetting breaks large network in smaller networks and smaller networks are easier to manage.
- Subnetting reduces network traffic by removing collision and broadcast traffic, that overall improve performance.
- Subnetting allows you to apply network security polices at the interconnection between subnets.
- Subnetting allows you to save money by reducing requirement for IP range.

CONCLUSION: We have successfully implemented the subnetting and subnet mask program.

Assignment No: 4

TITLE: Configuration of 3 router networks using RIP 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,

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

THEORY-CONCEPT:

Routing Information Protocol (RIP):

The Routing Information Protocol (RIP) is one of the oldest distance-vector routing protocols which employs the hop count as a routing metric. RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination. The largest number of hops allowed for RIP is 15, which limits the size of networks that RIP can support.

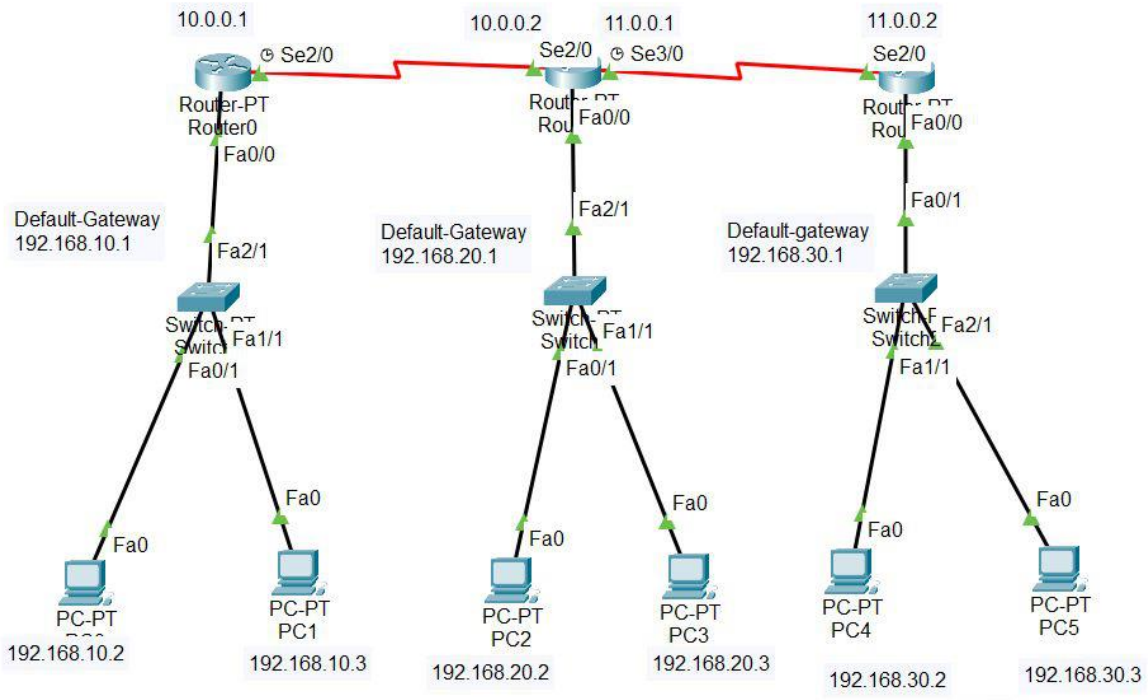
Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

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

IP Addressing Table:

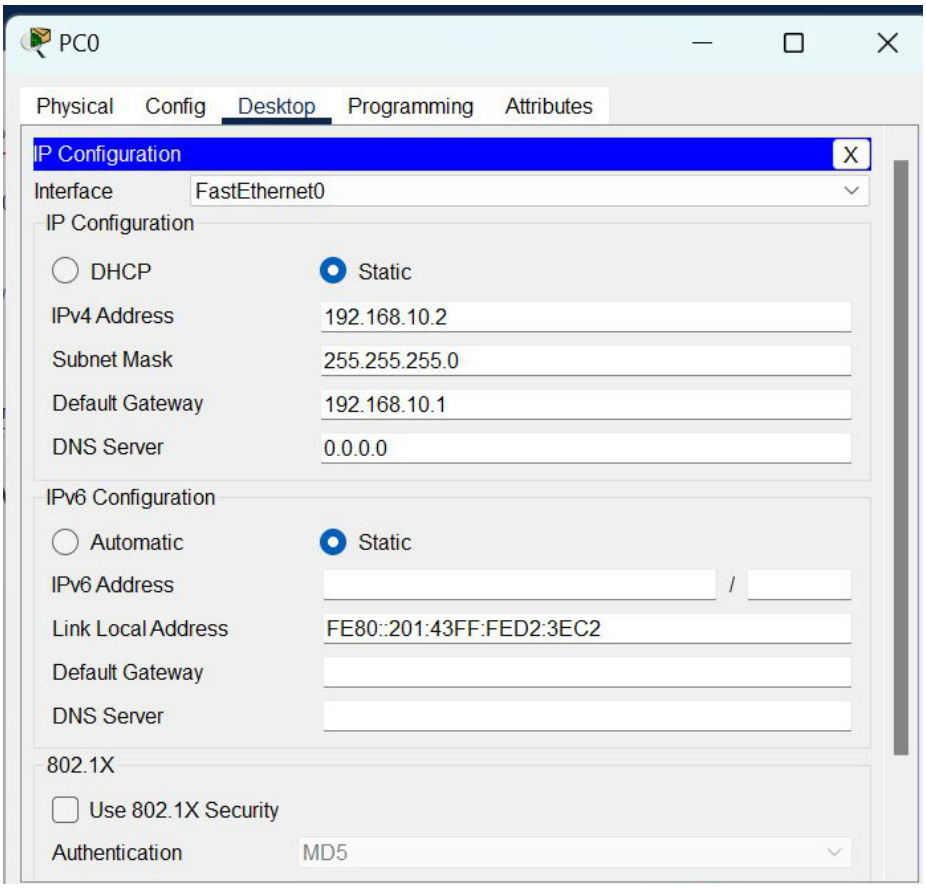
S.NO	Device	IPv4 Address	Subnet mask	Default Gateway
1.	PC0	192.168.10.2	255.255.255.0	192.168.10.1
2.	PC1	192.168.10.3	255.255.255.0	192.168.10.1
3.	PC2	192.168.20.2	255.255.255.0	192.168.20.1
4.	PC3	192.168.20.3	255.255.255.0	192.168.20.1
5.	PC4	192.168.30.2	255.255.255.0	192.168.30.1
6.	PC5	192.168.30.3	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.

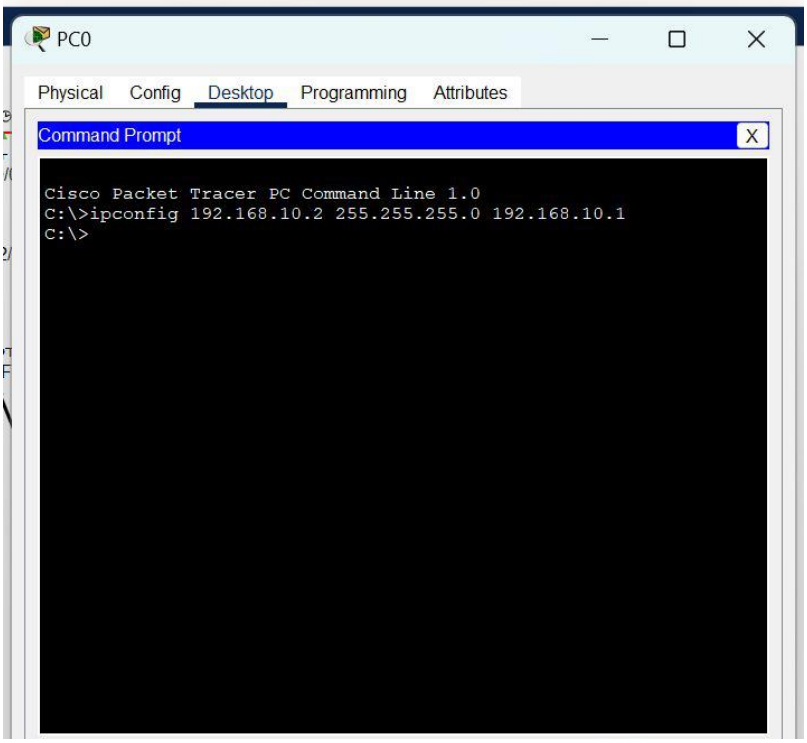


Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- 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.

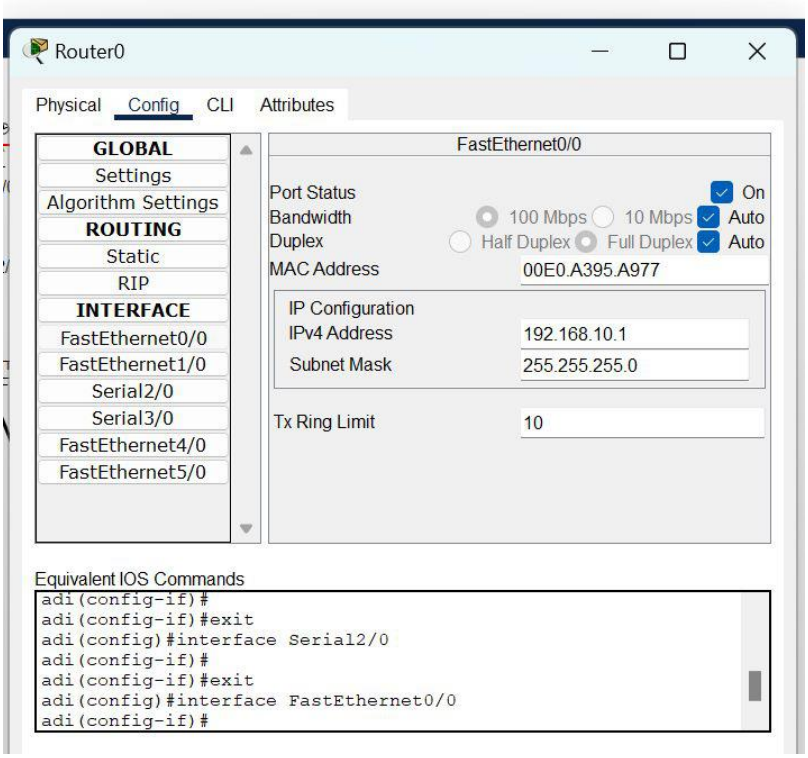
Step 3: Configure router with IP address and Subnet mask.

IP Addressing Table Router:

S.NO	Device	Interface	IPv4 Address	Subnet mask
1.	router0	FastEthernet0/0	192.168.10.1	255.255.255.0
		Serial2/0	10.0.0.1	255.0.0.0
2.	router1	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.	router2	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 : network <network id>

- RIP Routes for Router0 are given below:

Router(config)#network 192.168.10.0
Router(config)#network 10.0.0.0

- RIP Routes for Router1 are given below:

Router(config)#network 192.168.20.0
Router(config)#network 10.0.0.0
Router(config)#network 11.0.0.0

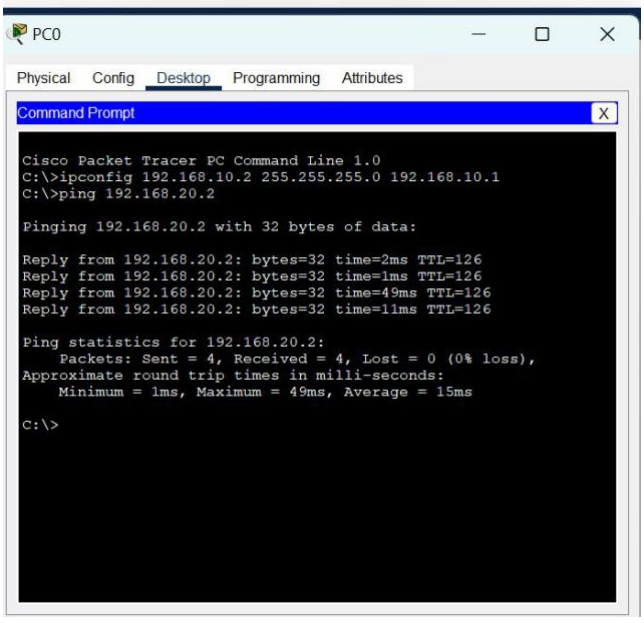
- RIP Routes for Router2 are given below:

Router(config)#network 192.168.30.0
Router(config)#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



CONCLUSION: We have learned to configure 3 router networks using RIP in Cisco Packet Tracer tool.

Assignment No:5

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.

Nonadaptive 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.

CONCLUSION: Thus, we have successfully implemented link state protocol algorithm.

Assignment No:06

AIM: Socket Programming using Python.

OBJECTIVE: 1. To learn socket programming.

2. To classify client server communication

PROBLEM STATEMENT: To study the Socket Programming.

OUTCOME: Students will able to

1. Apply Socket programming for client server communication.

THEORY-CONCEPT:

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server.

They are the real backbones behind web browsing. In simpler terms, there is a server and a client.

Socket programming is started by importing the socket library and making a simple socket.

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Here we made a socket instance and passed it two parameters. The first parameter is AF_INET and the second one is SOCK_STREAM. AF_INET refers to the address-family ipv4. The SOCK_STREAM means connection-oriented TCP protocol.

A simple server-client program:

Server: A server has a bind() method which binds it to a specific IP and port so that it can listen to incoming requests on that IP and port. A server has a listen() method which puts the server into listening mode. This allows the server to listen to incoming connections. And last a server has an accept() and close() method. The accept method initiates a connection with the client and the close method closes the connection with the client.

- First of all, we import socket which is necessary.
- Then we made a socket object and reserved a port on our pc.
- After that, we bound our server to the specified port. Passing an empty string means that the server can listen to incoming connections from other computers as well. If we would have passed 127.0.0.1 then it would have listened to only those calls made within the local computer.

- After that we put the server into listening mode.5 here means that 5 connections are kept waiting if the server is busy and if a 6th socket tries to connect then the connection is refused.
- At last, we make a while loop and start to accept all incoming connections and close those connections after a thank you message to all connected sockets.
  - **Client:**
    Now we need something with which a server can interact. We could telnet to the server like this just to know that our server is working. Type these commands in the terminal:

```
# start the server
$ python server.py

# keep the above terminal open
# now open another terminal and type:

$ telnet localhost 12345
```

If 'telnet' is not recognized. On windows search windows features and turn on the "telnet client" feature.

- First of all, we make a socket object.
- Then we connect to localhost on port 12345 (the port on which our server runs) and lastly, we receive data from the server and close the connection.
- Now save this file as client.py and run it from the terminal after starting the server script.

CONCLUSION: In this experiment we implemented client server communication with help of socket.

Assignment No:7

AIM: TCP socket for wired network.

OBJECTIVE: 1. To learn TCP socket for wired network

PROBLEM STATEMENT: Write a program using TCP socket for wired network for following

a. File transfer

OUTCOME: Students will able to

1. implement socket programming using TCP.

THEORY-CONCEPT:

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.

Socket: Sockets allow communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor. A file descriptor is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else.

Socket Types

There are four types sockets available to the users. The first two are most commonly used and the last two are rarely used. Processes are presumed to communicate only between sockets of the same type but there is no restriction that prevents communication between sockets of different types.

- Stream Sockets − Delivery in a networked environment is guaranteed. If you send through the stream socket three items "A, B, C", they will arrive in the same order − "A, B, C". These sockets use TCP (Transmission Control Protocol) for data transmission. If delivery is impossible, the sender receives an error indicator. Data records do not have any boundaries.

- Datagram Sockets − Delivery in a networked environment is not guaranteed. They're connectionless because you don't need to have an open connection as in Stream Sockets − you build a packet with the destination information and send it out. They use UDP (User Datagram Protocol).

- Raw Sockets − These provide users access to the underlying communication protocols, which support socket abstractions. These sockets are normally datagram oriented, though their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not intended for the general user; they have been provided mainly for those interested in developing new communication protocols, or for gaining access to some of the more cryptic facilities of an existing protocol.

- Sequenced Packet Sockets − They are similar to a stream socket, with the exception that record boundaries are preserved. This interface is provided only as a part of the Network Systems (NS) socket abstraction, and is very important in most serious NS applications. Sequenced-packet sockets allow the user to manipulate the Sequence Packet Protocol (SPP) or Internet Datagram Protocol (IDP) headers on a packet or a group of packets, either by writing a prototype header along with whatever data is to be sent, or by specifying a default header to be used with all

outgoing data, and allows the user to receive the headers on incoming packets.

Stages for server

1. Socket creation:

   int sockfd = socket(domain, type, protocol)

   sockfd: socket descriptor, an integer (like a file-handle)

   domain: integer, communication domain e.g., AF_INET (IPv4 protocol) , AF_INET6 (IPv6 protocol)

   type: communication type

   SOCK_STREAM: TCP(reliable, connection oriented)

   SOCK_DGRAM: UDP(unreliable, connectionless)

   protocol: Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)

2. Bind:

   int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

   After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure). In the example code, we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address

3. Listen:

   int listen(int sockfd, int backlog);

   It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

4. Accept:

   int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);

   It extracts the first connection request on the queue of pending connections for the listening

socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.

Stages for Client

1. Socket connection: Exactly same as that of server's socket creation

2. Connect:

int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server's address and port is specified in addr.

CONCLUSION:

We have successfully implemented the TCP socket programming

Assignment No:8

AIM: Study and Analyze the performance of FTP

OBJECTIVE: 1. Learn FTP

PROBLEM STATEMENT: Study and Analyze the performance of FTP protocol using Packet tracer tool.

OUTCOME: Students will able to

1. analyze the performance of HTTP, HTTPS and FTP using packet tracer tool.

THEORY-CONCEPT:

HTTP:

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

There are three basic features that make HTTP a simple but powerful protocol:

1. **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and

after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.

2. **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
3. **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:

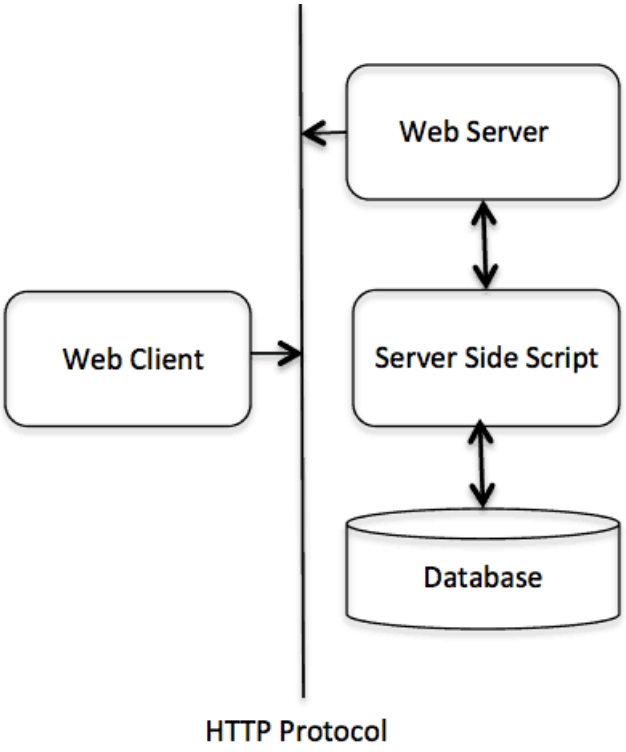


Fig: Basic architecture of a web application

The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

HTTPS:

HTTPS is an abbreviation of Hypertext Transfer Protocol Secure. It is a secure extension or version of HTTP. This protocol is mainly used for providing security to the data sent between a website and the web browser. It is widely used on the internet and used for secure communications. This protocol uses the 443 port number for communicating the data. This protocol is also called HTTP over SSL because the HTTPS communication protocols are encrypted using the SSL (Secure Socket Layer).

By default, it is supported by various web browsers. Those websites which need login credentials should use the HTTPS protocol for sending the data.

Advantages of HTTPS

- The main advantage of HTTPS is that it provides high security to users.
- Data and information are protected. So, it ensures data protection.
- SSL technology in HTTPS protects the data from third-party or hackers. And this technology builds trust for the users who are using it.
- It helps users by performing banking transactions.

Disadvantages of HTTPS

- The big disadvantage of HTTPS is that users need to purchase the SSL certificate.

- The speed of accessing the website is slow because there are various complexities in communication.
- Users need to update all their internal links.

The **File Transfer Protocol (FTP)** is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP employs a **client-server** architecture whereby the client machine has an **FTP client** installed and establishes a connection to an **FTP server** running on a remote machine. After the connection has been established and the user is successfully authenticated, the data transfer phase can begin.

Let's now do FTP configuration in Packet Tracer:

1.Build the network topology.

2. Configure static IP addresses on the Laptop and the server.

Laptop: **IP address:** 192.168.1.1 Subnet **Mask:** 255.255.255.0

Server: **IP address:** 192.168.1.2 Subnet **Mask:** 255.255.255.0

3. Now try using an **FTP client** built in the Laptop to send files to an **FTP server** configured in the Server.

From the Laptop's command prompt, FTP the server using the server IP address by typing:

```
ftp  192.168.1.2
```

Provide the **username**(cisco) and **password**(cisco) [which are the defaults] for ftp login.

You are now in the FTP prompt .

PC0 has an **FTP client** which can be used to read, write, delete and rename files present in the FTP server.

The **FTP server** can be used to read and write configuration files as well as IOS images. Additionally, the FTP server also supports file operations such rename, delete and listing directory.

With that in mind, we can do something extra. So let's do this:

Create a file in the Laptop then **upload** it to the server using **FTP**.

To do this, open the **Text Editor** in the Laptop, create a file and give it your name of choice.

Type any text in the editor then **save** your file. e.g. myFile.txt.

Now upload the file from the Laptop to the server using FTP. (An FTP connection has to be started first. But this is what we've done in step 3)

So to do an FTP upload, we'll type:

```
put MyFile.txt
```

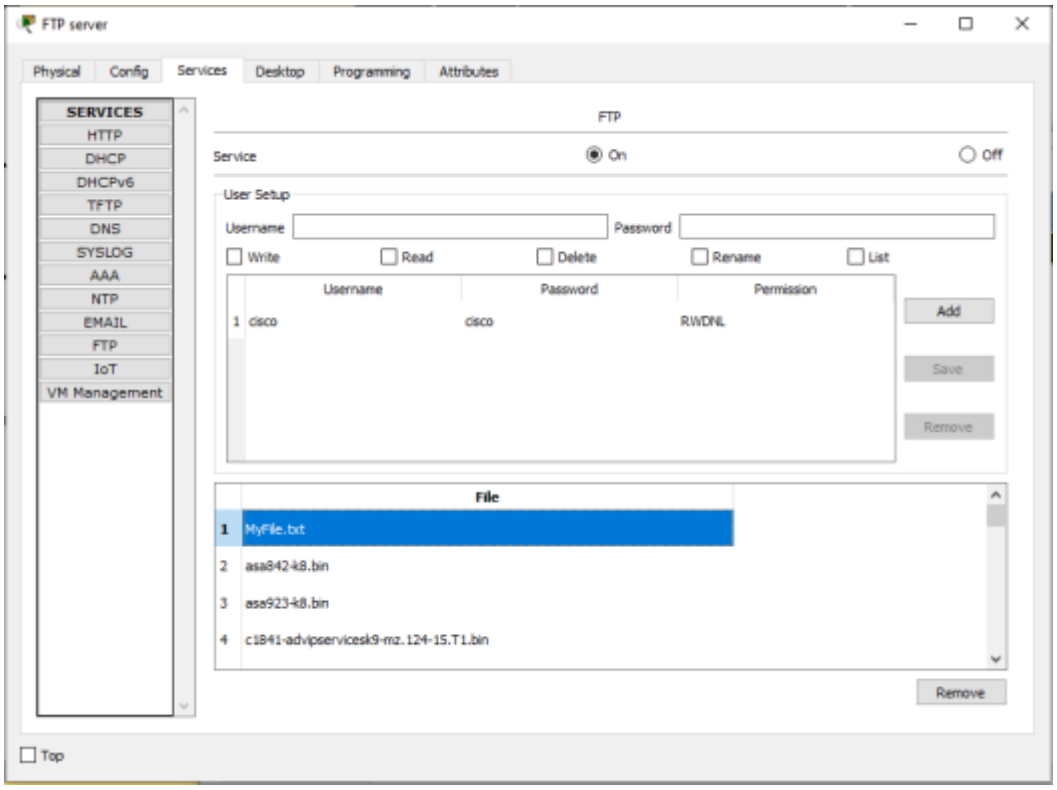
```
ftp>
ftp>put MyFile.txt

Writing file MyFile.txt to 192.168.1.2:
File transfer in progress...

[Transfer complete - 47 bytes]

47 bytes copied in 0.023 secs (2043 bytes/sec)
ftp>
```

Once file upload is successful, go to the Server **FTP directory** to verify if the file sent has been received . To do this, go to **Server-> Services->FTP**. Here look for MyFile.txt sent from the laptop.



CONCLUSION :

We have successfully analyzed the performance of HTTP, HTTPS and FTP.

Assignment No: 09

AIM: Program for DNS lookup.

OBJECTIVE: 1. To learn DNS concept.

2.To get the host name and IP address.

3.To map the host name with IP address and Vice-versa

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

OUTCOME: Students will able to

1. summarize DNS lookup.

THEORY-CONCEPT:

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).

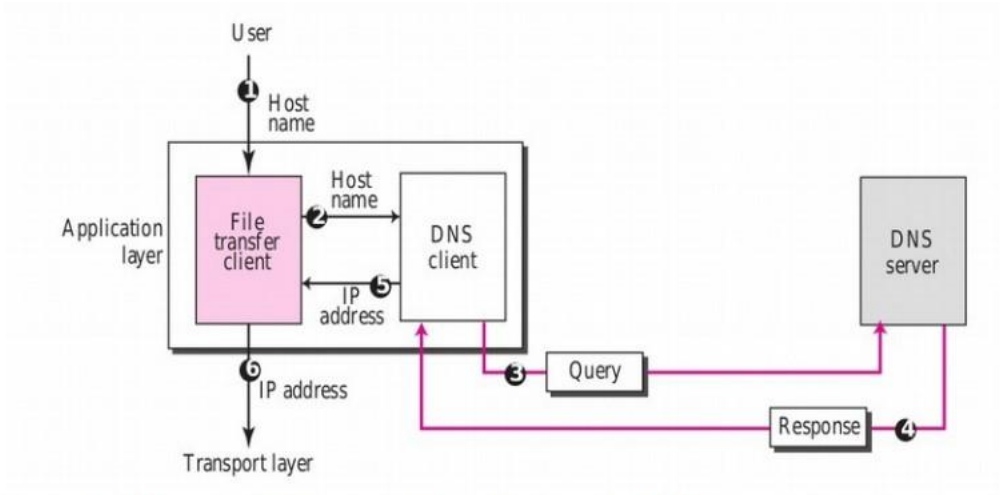


Figure 1: How TCP/IP uses a DNS client and a DNS server to map a name to an address; the reverse mapping is similar.

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)

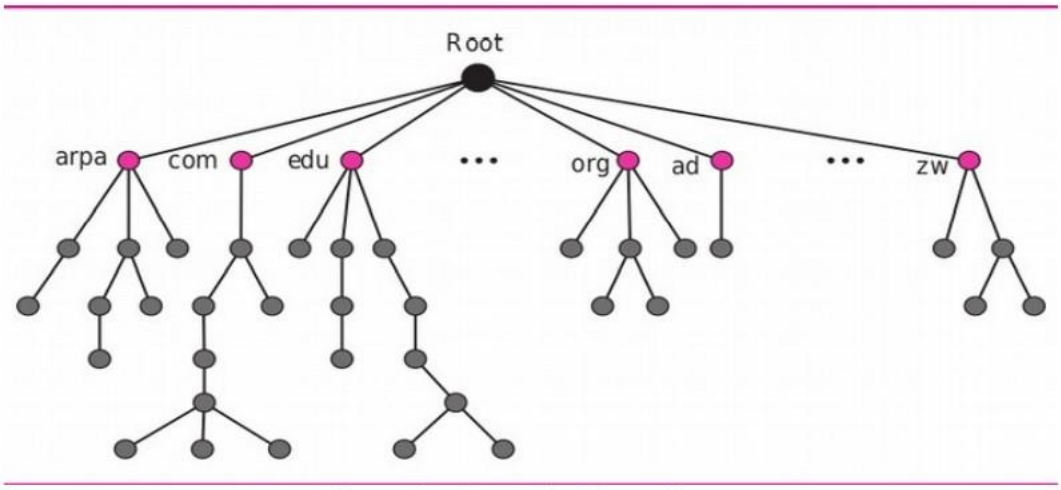


Figure 2 : Domain Name Space

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.

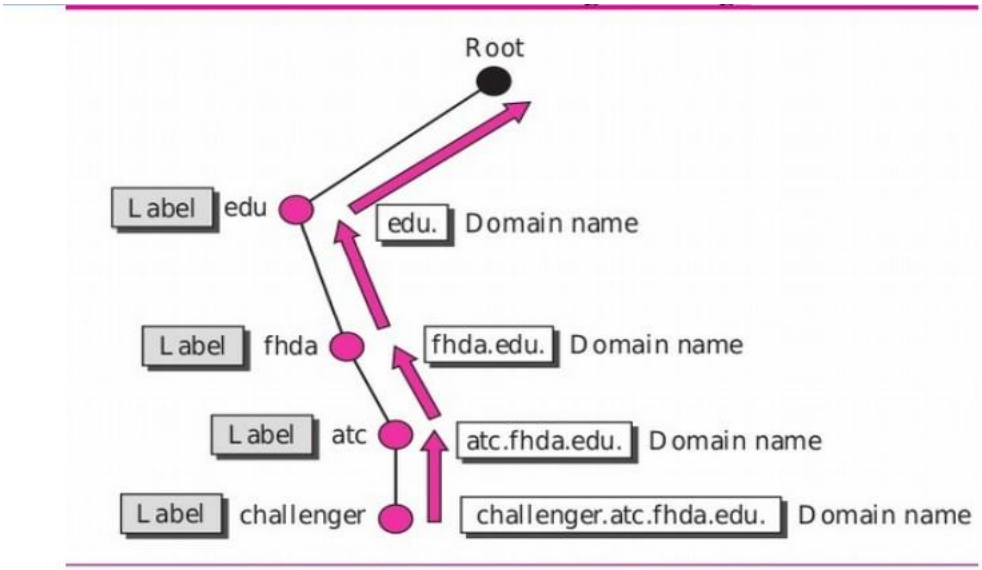


Figure 3 shows some domain names.

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).

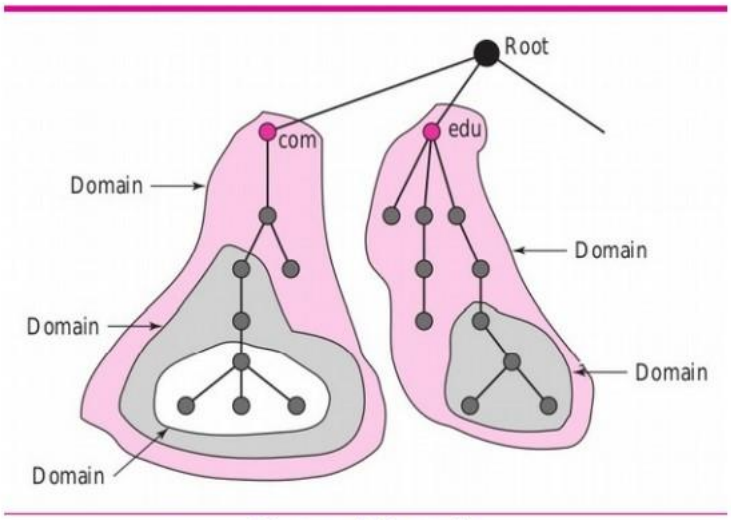


Figure 4 Domain

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

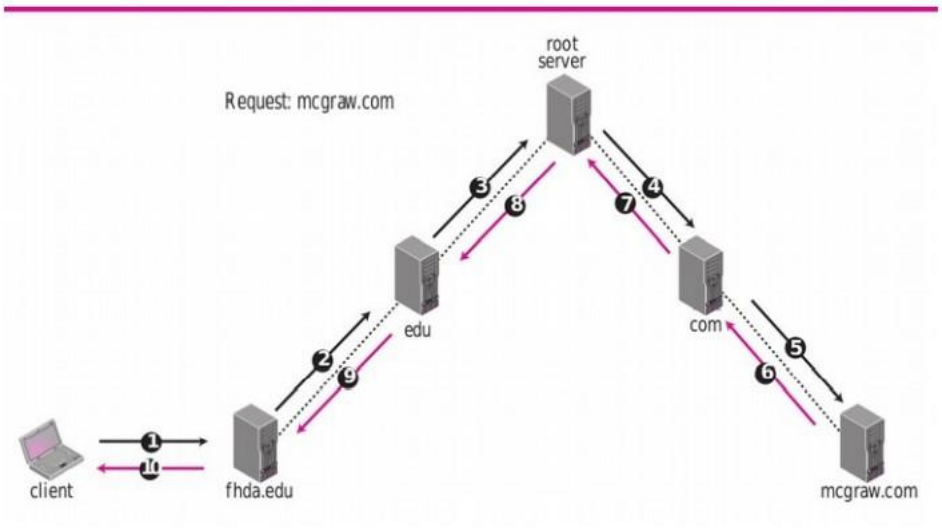


Figure 5: Recursive resolution.

CONCLUSION:

We have successfully performed DNS lookup program.

Content Beyond Syllabus -10

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.