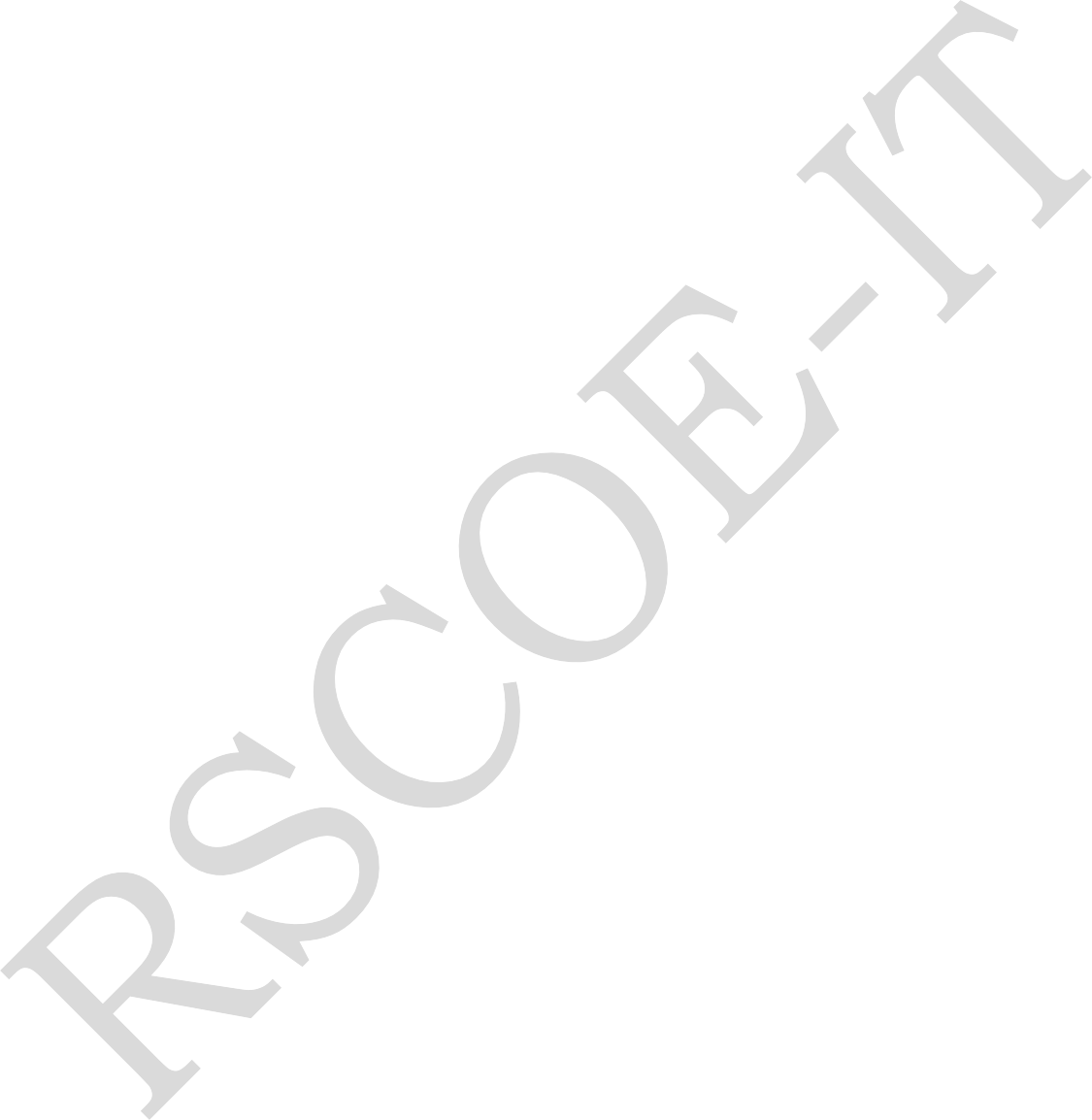
|  |  |  |
| --- | --- | --- |
| D:\College Data\formats\college icons\jspmlogo.gif | **JSPM’s**  **RAJARSHI SHAHU COLLEGE OF ENGINEERING**  **TATHAWADE, PUNE-33**  **(An Autonomous Institute Affiliated to SavitribaiPhule Pune University,Pune)** | D:\College Data\formats\college icons\rscoelogo.gif |

## LAB MANUAL



**OF**

**Computer Networks and Applications**

**(IT-2111)**

**S.Y.B.Tech(2020)**

**INFORMATION TECHNOLOGY**

**Authors: 1) Dr. Maya S. BembdeCreation Date: -December 2021**

**LastUpdated: -July2022**

**Version: -**

**1**

**© JSPM Group of Institutes, Pune. All Rights Reserved. All the information in this Course Manual is confidential. Participants shall refrain from copying, distributing, misusing/ or disclosing the content to any third parties any circumstances whatsoever.**

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | | | |
| **Sr.**  **No.** |  | **Topic** | **Page. No.** |
| **1** | | **How to Use This Manual…………….** | **4** |
| **2** | | **Laboratory Details……………………** | **5** |
|  | 2.1 | Laboratory Objective | **8** |
|  | 2.2 | Laboratory Equipment | **8** |
|  | 2.3 | Laboratory Plan | **12** |
|  | 2.4 | Experiment No.1 | **13** |
|  |  | Practical Session Plan | **13** |
|  |  | Aim, theory, procedure, observations, calculations etc | **14-22** |
|  | 2.5 | Experiment No.2. | **23** |
|  |  | Practical Session Plan | **23** |
|  |  | Aim, theory, procedure, observations, calculations etc | **24-42** |
|  | 2.6 | Experiment No.3 | **43** |
|  |  | Practical Session Plan | **43** |
|  |  | Aim, theory, procedure, observations, calculations etc | **44-47** |
|  | 2.7 | Experiment No.4 | **48** |
|  |  | Practical Session Plan | **48** |
|  |  | Aim, theory, procedure, observations, calculations etc | **49-54** |
|  | 2.8 | Experiment No.5 | **55** |
|  |  | Practical Session Plan | **55** |
|  |  | Aim, theory, procedure, observations, calculations etc | **56-60** |
|  | 2.9 | Experiment No.6 | **61** |
|  |  | Practical Session Plan | **61** |
|  |  | Aim, theory, procedure, observations, calculations etc | **62-73** |
|  | 2.10 | Experiment No.7 | **74** |
|  |  | Practical Session Plan | **74** |
|  |  | Aim, theory, procedure, observations, calculations etc | **75-78** |
|  | 2.11 | Experiment No.8 | **79** |
|  |  | Practical Session Plan | **79** |
|  |  | Aim, theory, procedure, observations, calculations etc | **80-88** |
|  | 2.12 | Experiment No.8 | **89** |
|  |  | Practical Session Plan | **89** |
|  |  | Aim, theory, procedure, observations, calculations etc | **89-94** |
|  | 2.13 | Experiment No.8 | **95** |
|  |  | Practical Session Plan | **95** |
|  |  | Aim, theory, procedure, observations, calculations etc | **96-107** |

1. **How to Use ThisManual**

This manual assumes that the facilitators are aware of collaborative learning methodologies.

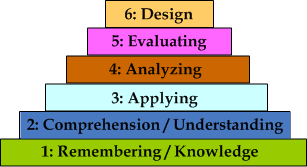
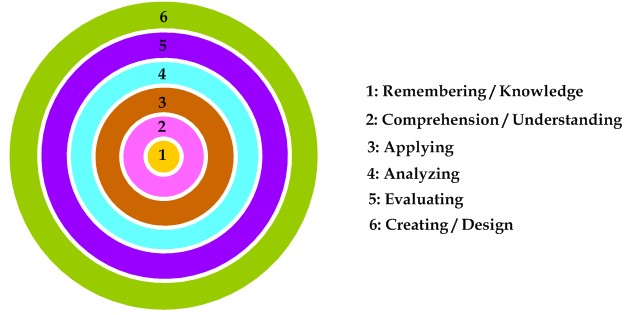
This manual will provide a tool to facilitate the session on Digital Communication modules in collaborative learning environment.

The facilitator is expected to refer this manual before the session.

## Icon of Graduate Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **K**  Applying Knowledge | **A**  Problem Analysis | **D**  Design &Development | **I**  Investigation of problems |
| **M**  Modern Tool Usage | **E**  Engineer & Society | **E**  Environment Sustainability | **T**  Ethics |
| **T**  Individual & Team work | **O**  Communication | **M**  Project  Management & Finance | **I**  Life Long Learning |

**Disk Approach- Digital Blooms Taxonomy**



This Manual uses icons as visual cues to the interactivities during the session.

|  |  |
| --- | --- |
|  | **Blooms Taxonomy** |
| DB11.bmp | Remembering |
| DB13.bmp | Understanding |
| DB12.bmp | Applying |
| DB14.bmp | Analyzing |
| DB15.bmp | Evaluating |
| DB16.bmp | Creating |

|  |  |
| --- | --- |
|  | This icon is used to indicate instructions for faculties. |
|  | This icon is used to indicate a statement to be made by faculty. |
|  | This icon is used to indicate a list of additional resources. |
|  | This icon indicates an activity to be conducted. |
|  | This icon indicates questions to be asked by faculty. |

## VISION of Institute

To satisfy the aspirations of youth force, who want to lead the nation towards prosperity through techno economic development.

## MISSION of Institute

To provide, nurture and maintain an environment of high academic excellence, research and entrepreneurship for all aspiring students, which will prepare them to face global challenges maintaining high ethical and moral standards.

## VISION of Department

To create quality information technology professionals through superior academic environment.

## MISSION of Department

To incorporate the IT fundamentals in students to be successful in their career. To motivate the students for higher studies, research and entrepreneurship. To provide IT services to society.

## PSO’s of Department

PSO1: Demonstrate the ability to apply discrete principles of mathematics along with programming paradigms to expedite solution building in the IT domain.

PSO2: Apply computational techniques using core aspects of network and system programming to deliver secured applications in the arena of analytics and computing.

PSO3: Demonstrate project management skills to handle multidisciplinary complex tasks proficiently and utilize these skills for entrepreneurship.

**PROGRAM OUTCOMES (POs)**

**Engineering Graduates will be able to:**

**PO 1: Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO 2. Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO 3. Design/Development of Solutions:** 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.

**PO 4. Conduct investigations of complex problems:** 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.

**PO 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO 6. The engineer and society:** 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.

**PO 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainabledevelopment.

**PO 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO 10. Communication:** 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.

**PO 11. Project management and finance:** 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.

**PO 12. Life-long learning:** 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.

## Course Prerequisites:Basic knowledge of Computer Programming, Basic knowledge of Computer Organization, Fundamentals of Data Communication.

## Course Objectives:

## To familiarize students with basic concepts and types of networks.

## To understand ISO/OSI model and TCP/IP model

## To understand data link, network layer of ISO/OSI Model

## To understand transport and application layer ISO/OSI Model

## To study Wireless network.

## Course Outcomes

**CO1:** Define Computer Networks and describe how different types of computer networks are designed using concept of layered approach.

**CO2:** Understand design issues of Data Link Layer such as frame formation, error control and MAC protocols. Compare and demonstrate different flow control mechanisms.

**CO3:** Apply concepts of IP addresses and routing algorithms to design subnets and to construct routing tables respectively.

**CO4:** Analyse functionalities of TCP and UDP protocols and demonstrate socket programming to establish Client-Server Communication.

**CO5:** Understand the working and security aspects of different protocols of Application layer.

**CO6:** Understand the key technological components of the Wireless Network and its applications.

## LaboratoryObjectives

**Upon completion of Computer Networks and Application Lab student should able to:**

1. Distinguish between different types of transmission media in terms of its characteristics, cost and use in different types of network designs
2. Utilize different Linux commands and TCP/IP utilities based on its use.
3. Implement a small network using Hub, switch and Router using Cisco Packet Tracer.
4. Implement Client/Server architecture and Star, Bus and Ringnetwork topology. Assign IP addressing and subnet masks to the devices in the network using Cisco Packet Tracer.
5. Implement Sliding Window protocol to achieve flow control.
6. Implement TCP sockets and UDP sockets using socket programming to study connection-oriented and connectionless services.
7. Implement concept of subnetting and supernetting using network simulator.
8. Implement Distance Vector/ Link State Routing algorithm.

## Laboratory Equipmentlist

**‘**

|  |  |
| --- | --- |
| **Sr No.** | **Other Networking devices** |
| **1** | **Network Interface card** |
| **2** | **Twisted Pair Cables** |
| **3** | **Hub** |
| **4** | **Switch** |
| **5** | **Router** |
| **6** | **Crimping Tool** |
| **7** | **Cable Tester** |

**Software Requirement:**

**1. Windows /Linux operating System**

**2. Cisco Packet Tracer**

* 1. **LaboratoryPlan**

|  |  |  |
| --- | --- | --- |
| **Exp. No.** | **Title of Experiment** | **Session** |
| **1.** | Study on Networks Cabling (Guided and Unguided) | III |
| **2.** | Explore and Study of TCP/IP utilities and Network Commands on Linux.   1. Ping 2. Tracert/Traceroute/Tracepath 3. ipconfig / ifconfig 4. NSlookup 5. Hostname 6. Arp 7. Whois 8. Finger 9. Netstat 10. Port Scan / nmap 11. Route | IV |
| **3.** | Implementation of small network using Hub, switch and Router using Cisco Packet Tracer. | V |
| **4.** | Using simulator build a small network (P2P, Client/Server) implement network topology (Star, Bus, Ring) and IP addressing and subnet masking. | VI |
| **5.** | To write a C program to perform sliding window. | VII |
| **6.** | Socket Programming in C/C++ a) TCP Client, TCP Server b) UDP Client, UDP Server. | VIII |
| **7.** | Using any simulator configure: 1.Subnetting of a given network 2. Super netting of given network. | IX |
| **8.** | Simulation of Distance Vector/ Link State Routing algorithm. | X |
| **9.** | Setup a wired LAN using Layer 2 Switch and then IP switch ofminimum four computers. | XI |
| **10.** | Configure RIP/OSPF/BGP using packet Tracer. | XII |
| End semester assessment | |  |
|  | Mock Practical Examination |  |

**EXPERIMENT NO.1**

**Study on Networks Cabling (Guided and Unguided)**

# Practical Session Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 100 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 20 | Results and conclusions | Keywords | Lists, Facilitates | Listens,  Participates, Discusses | intrapersonal,  Comprehension |

**Title: Study on Networks Cabling (Guided and Unguided)**

**Objectives:** Distinguish between different types of transmission media in terms of its characteristics, cost and use in different types of network designs

**Aim/ Problem Statement**: To study different types of transmission media used in wired and wireless networks.

## Apparatus: Twisted pair cable, RJ45 connector, Crimping tool, cable tester

## Theory: What is Transmission media? In data communication terminology, a transmission medium is a physical path between the transmitter and the receiver i.e. it is the channel through which data is sent from one place to another. Transmission Media is broadly classified into the following types:

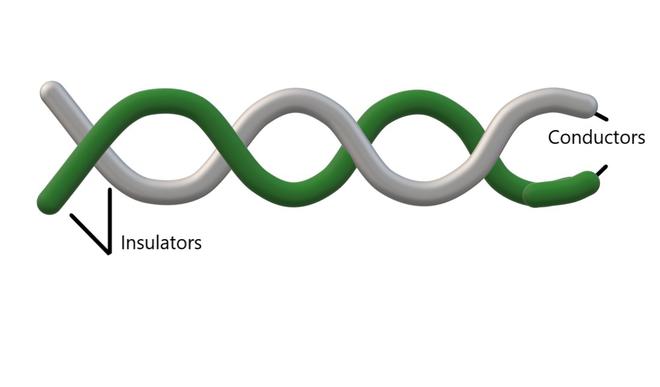
## Transmission Media - javatpoint

**1. Guided Media:**It is also referred to as Wired or Bounded transmission media. Signals being transmitted are directed and confined in a narrow pathway by using physical links.   
Features:

* High Speed
* Secure
* Used for comparatively shorter distances

There are 3 major types of Guided Media:

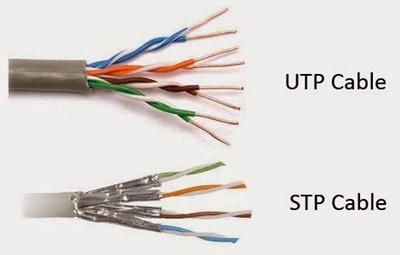
**(i) Twisted Pair Cable –**   
It consists of 2 separately insulated conductor wires wound about each other. Generally, several such pairs are bundled together in a protective sheath. They are the most widely used Transmission Media.

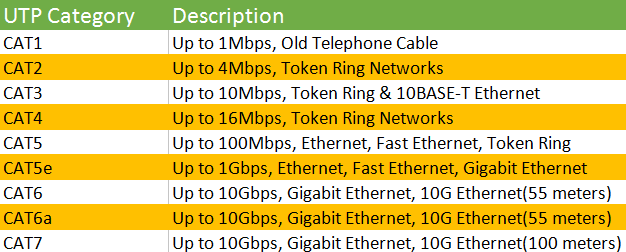


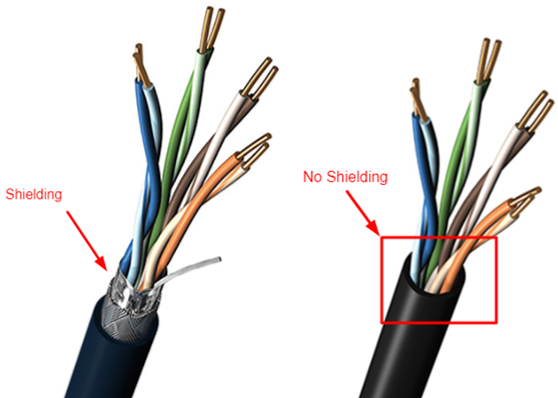
Twisted Pair is of two types:

* **Unshielded Twisted Pair (UTP):**

UTP consists of two insulated copper wires twisted around one another. This type of cable has the ability to block interference and does not depend on a physical shield for this purpose. It is used for telephonic applications.





**Advantages of UTP cable:**

⇢ Least expensive

⇢ Easy to install

⇢ High-speed capacity

⇢ Susceptible to external interference

⇢ Lower capacity and performance in comparison to STP

⇢ Short distance transmission due to attenuation

* **Shielded Twisted Pair (STP):**

This type of cable consists of a special jacket (a copper braid covering or a foil shield) to block external interference. It is used in fast-data-rate Ethernet and in voice and data channels of telephone lines.

RJ45 connector

Crimping tool and LAN Tester



**Advantages:**

**⇢**Better performance at a higher data rate in comparison to UTP

⇢ Eliminates crosstalk

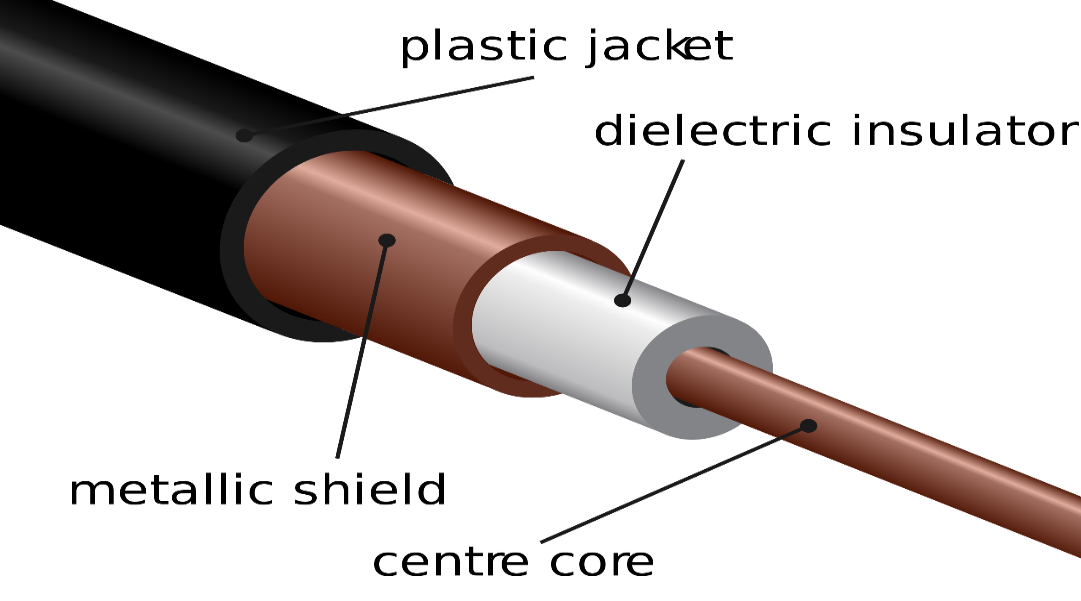
⇢Comparatively faster

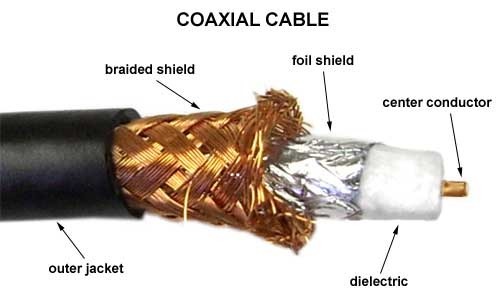
⇢Comparatively difficult to install and manufacture

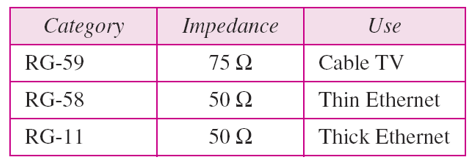
⇢More expensive

⇢ Bulky

**(ii) Coaxial Cable –**    
It has an outer plastic covering containing an insulation layer made of PVC or Teflon and 2 parallel conductors each having a separate insulated protection cover. The coaxial cable transmits information in two modes: Baseband mode(dedicated cable bandwidth) and Broadband mode(cable bandwidth is split into separate ranges). Cable TVs and analog television networks widely use Coaxial cables.





**Categories of Coaxial cable:**

**Connectors used:**



**Advantages:**

* High Bandwidth
* Better noise Immunity
* Easy to install and expand
* Inexpensive

**Disadvantages:**

* Single cable failure can disrupt the entire network

**(iii) Optical Fiber Cable**  
It uses the concept of reflection of light through a core made up of glass or plastic. The core is surrounded by a less dense glass or plastic covering called the cladding. It is used for the transmission of large volumes of data.

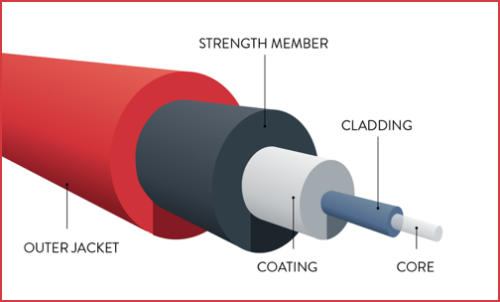
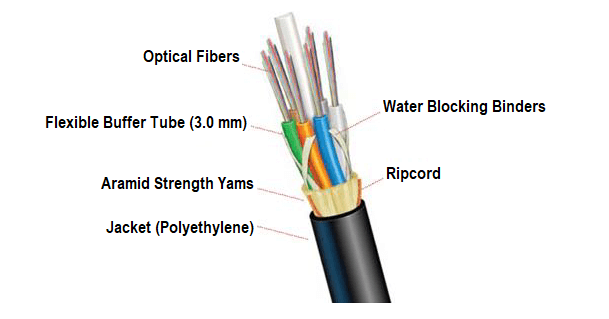
The cable can be unidirectional or bidirectional. The WDM (Wavelength Division Multiplexer) supports two modes, namely unidirectional and bidirectional mode.

**Advantages:**

* Increased capacity and bandwidth
* Lightweight
* Less signal attenuation
* Immunity to electromagnetic interference
* Resistance to corrosive materials

**Disadvantages:**

* Difficult to install and maintain
* High cost
* Fragile



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

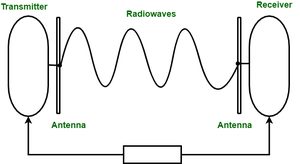
Features:

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

There are 3 types of Signals transmitted through unguided media:

* 1. **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.

****

**Further Categorized as (i) Terrestrial and (ii) Satellite.**

**(ii) 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.

**(iii) 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:** Thus we have understood transmission media in detail.

Upon completion of Experiment students will be able to:

1. Distinguish different types of transmission media.
2. Identify transmission media to be used in a specific type of network design.

# EXPERIMENT NO.2

## Explore and Study of TCP/IP utilities and Network Commands on Linux.

## Ping

## Tracert/Traceroute/Tracepath

## ipconfig / ifconfig

## NSlookup

## Hostname

## Arp

## Whois

## Finger

## Netstat

## Port Scan / nmap

## Route

## Practical Session Plan.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time (min)** | | | **Content** | | **Learning Aid / Methodology** | | **Faculty Approach** | | **Typical Student Activity** | | **Skill /**  **Competency Developed** |
| 20 | | | Explanation of Experiment | | Chalk & Talk , Presentation | | Introduces, Facilitates, Explains | | Listens | | Knowledge, Communication, intrapersonal, Application |
| 100 | | | Explanation and demonstration of each command :  Syntax  Use  Inputs  Expected output | | Paper work | | Introduces, Facilitates, Explains | | Discuss | | Knowledge, Communication, intrapersonal, Application |
| 90 | | | Implementation of commands | | Keyboard, mouse, computers. | | Introduces, Facilitates, Explains | | Discuss, participates | | Knowledge, Communication, intrapersonal, Application |
| 20 | Checking of output | | Keyboard, mouse,  Computers. | | Introduces,  Facilitates, Explains | | Discuss,  participates | | Knowledge,  Communication, intrapersonal, Application | | |
| 10 | Results and conclusions | | Keywords | | Lists, Facilitates | | Listens, Participates, Discusses | | Knowledge, Communication, intrapersonal, Comprehension | | |

**Title:** Explore and Study of TCP/IP utilities and Network Commands on Linux.

**Objectives:** Utilize different Linux commands and TCP/IP utilities based on its use

**Aim/ Problem Statement**:To explore and study TCP/IP utilities and basic network commands

## Theory:Following commands are studied in detail

## Ping: is one of the most basic yet useful network commands to utilize in the command prompt application. It tells you whether your computer can reach some destination IP address or domain name, and if it can, how long it takes data to travel there and back again. The ping command is one of the most often used networking utilities for detecting devices on a network and for troubleshooting network problems. When you ping a device you send that device a short message, which it then sends back (the echo). The general format is ping hostname or ping IPaddress.

## Synatx:

## pingwww.google.com

## ping 216.58.208.68

## Output:

## C:\>ping example.microsoft.com

## pinging example.microsoft.com [192.168.239.132] with 32 bytes of data:

## Reply from 192.168.239.132: bytes=32 time=101ms TTL=124

## Reply from 192.168.239.132: bytes=32 time=100ms TTL=124

## Reply from 192.168.239.132: bytes=32 time=120ms TTL=124

## Reply from 192.168.239.132: bytes=32 time=120ms TTL=124

* + 1. **Tracert/Traceroute/Tracepath**

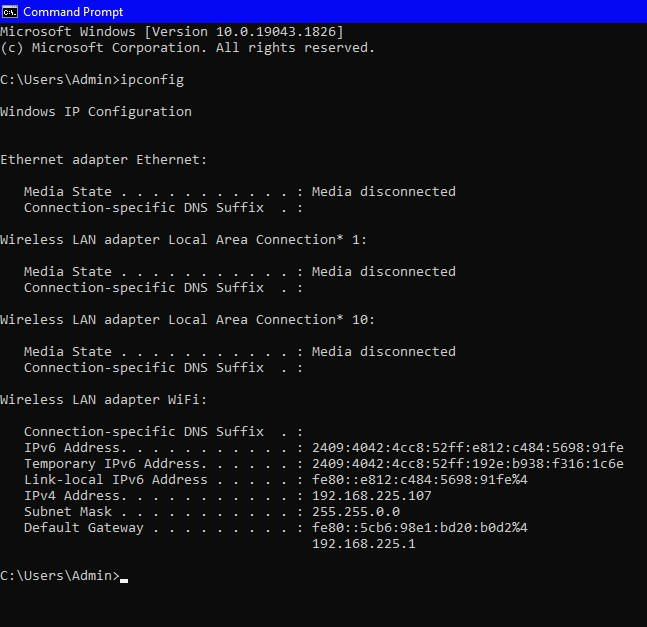
## Tracertstands for Trace Route. Likepingit sends out a data packet as a way to troubleshoot any network issues you might have, but it instead tracks the route of the packet as it hops from server to server.

* + 1. ipconfig / ifconfig

The “**ifconfig**” command is used for displaying current network configuration information, setting up an ip address, netmask, or broadcast address to a network interface, creating an alias for the network interface, setting up hardware address, and enable or disable network interfaces.

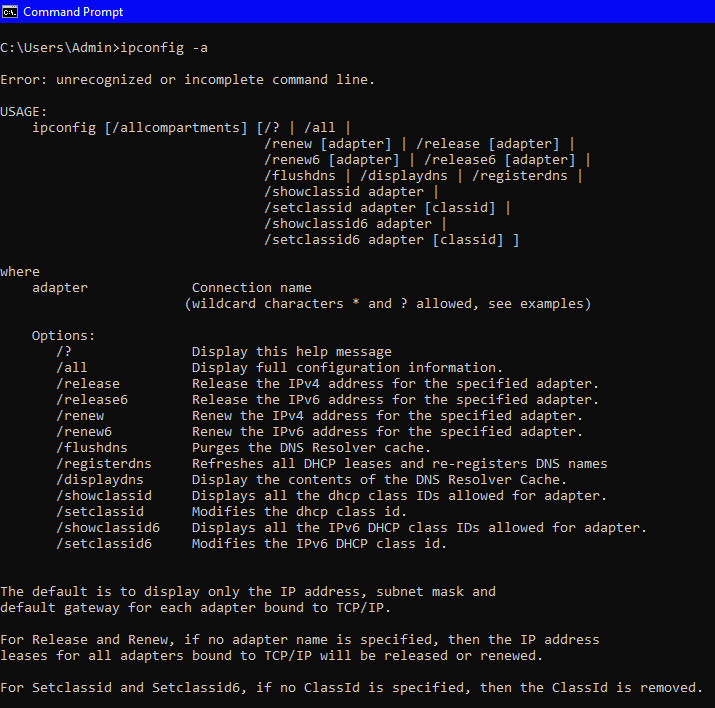
* 1. **View All Network Interface Settings:**

The “**ifconfig**” command with no arguments will display all the active interfaces details. The **ifconfig** command is also used to check the assigned IP address of a server.



* 1. **Display Information of All Network Interfaces :**

The following ifconfig command with the -a argument will display information of all active or inactive network interfaces on the server. It displays the results for eth0, lo, sit0 and tun0.

****

* + 1. NSlookup

It is used for querying the Domain Name System (DNS) to obtain domain name or IP address mapping information. The main use of nslookup is for troubleshooting DNS related problems.

Nslookup can be use in interactive and non-interactive mode. To use in interactive mode type nslookup at the command line and hit return. You should get an nslookup command prompt.

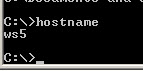


Then you can provide IP address or domain name accordingly



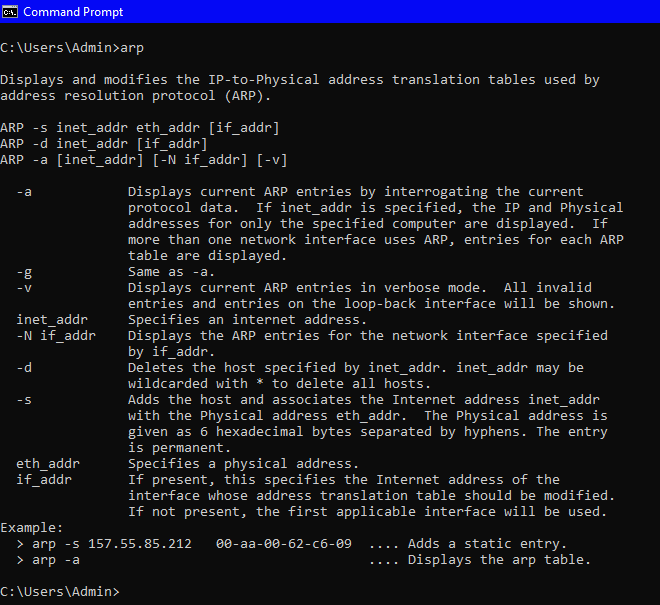
* + 1. **Hostname**

A very simple command that displays the host name of your machine. This is much quicker than going to the control panel>system route.

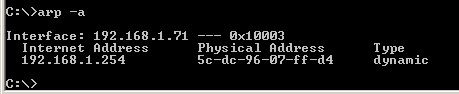


* + 1. **Arp**

This is used for showing the address resolution cache. This command must be used with a command line switch. Type arp at the command line to see all available options.



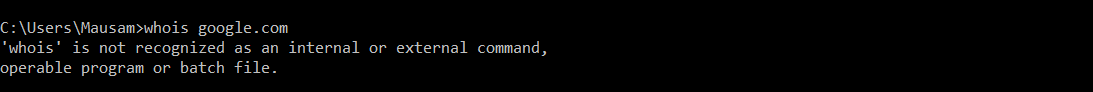
arp -a is the most common.



* + 1. **Whois**

It is used to find out the information about a domain, such as the owner of the domain, the owner’s contact information, and the name servers that the domain is using.It allows you to perform lookup of owner information of a website by querying databases that store the registered users of a domain or IP address.

But if you try to run the same command on a Windows machine, you will face the following error:

 >whois youtube.com  
'whois' is not recognized as an internal or external command,  
operable program or batch file.

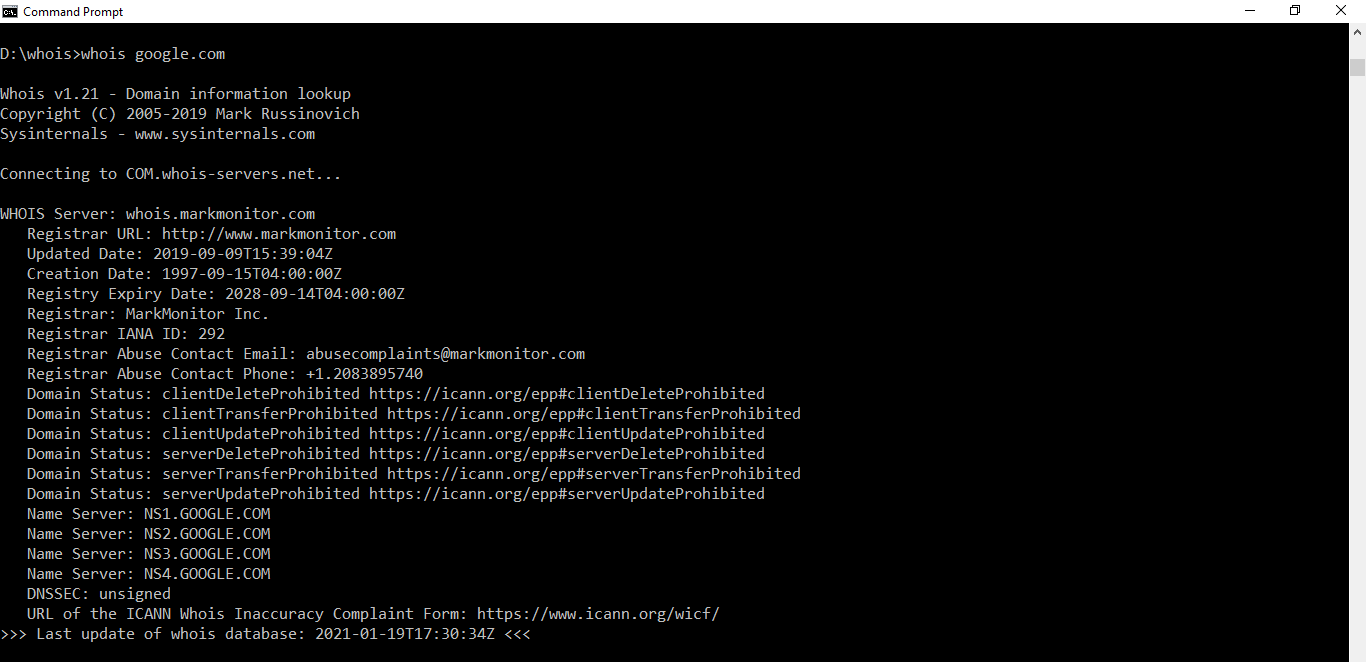
In order to resolve the ‘whois’ not recognized, we need to manually install a Whois program.

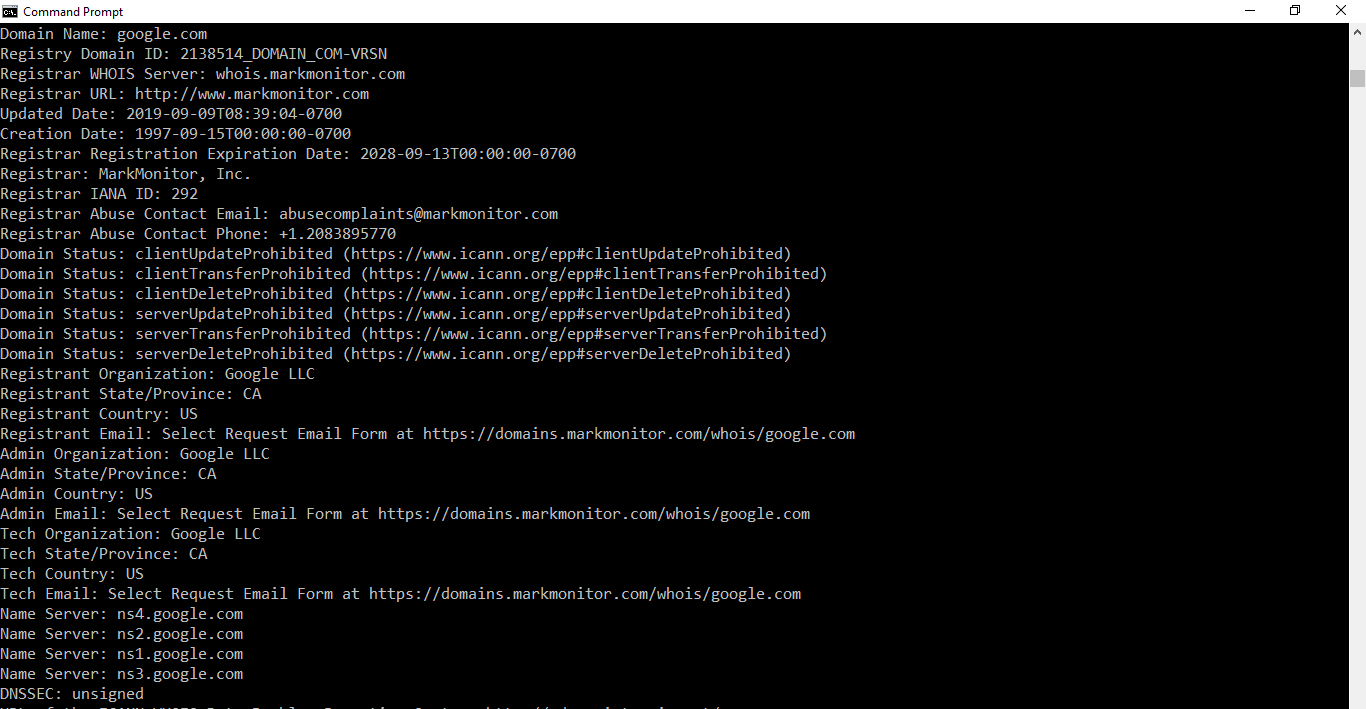
1. [Download Whois Program](https://docs.microsoft.com/en-us/sysinternals/downloads/whois) from Microsoft’s site.
2. Create a folder in your computer(eg. whois) and Extract the content of the downloaded zip file to your created folder.
3. example — D:\whois
4. You will find whois.exe and whois64.exe under your extracted location. In my case it is D:\whois\whois.exe and D:\whois\whois64.exe
5. Open command prompt.
6. Navigate to the directory where you extracted the whois.exe. In my case I will type

>cd D:\whois

7. Run ‘whois’ command now and it should work. Example —

D:\whois>whois google.com





If you want to run the ‘whois’ command from anywhere then you can add the directory (in my case D:\whois) to the system PATH environment variable.Use following steps:

* Adding the your ‘whois’ directory path to Windows system PATH
* Open System in Control Panel — -> Go to Advanced System Settings — -> Click on Environment Variables —> Define a new System Variable with PATH=D:\whois
* Close and reopen your command prompt for allowing the changes to take place. Now, you will be able to run the WHOIS command from any path inside command prompt.

1. **Finger**

It displays information about the user or user on the specified remote computer (usually a UNIX computer) running the finger service or daemon. The remote computer specifies the format and output for displaying user information. If used without parameters, the finger command will show the help section.

**Syntax:**

finger [-l] [] [@] [.]

**Parameters:**

Parameter Description-l Displays user information in long list format. User Specifies the user you want to know about. If you omit the User parameter , the finger command displays information about all users on the specified computer. @ Specify the remote computer to run the finger service, where you are searching for user information. You can specify the computer name or IP address. /? Show help at the command prompt.

Note

• You can specify multiple User @ Host parameters.

• You must precede the parameters to give finger a dash (-) rather than a slash (/).

• This command is only available if the Internet Protocol (TCP / IP) protocol is installed as a component in the properties of the network adapter in Network Connections.

• Windows Server 2003 does not provide finger service.

**For example**

* To display information for user1 on users.microsoft.com, enter:

finger user1@users.microsoft.com

* To display information for all users on users.microsoft.com, enter:

finger @users.microsoft.com

1. **Netstat**

It means network statistics, is a Command Prompt command used to display very detailed information about how your computer is communicating with other computers or network devices.

Specifically, it can show details about individual network connections, overall and protocol-specific networking statistics, and much more, all of which could help troubleshoot certain kinds of networking issues.

**Syntax:**

netstat [-a] [-b] [-e] [-f] [-n] [-o] [-p protocol] [-r] [-s] [-t] [-x] [-y] [time\_interval] [/?]

| **Netstat Command List** | |
| --- | --- |
| **Option** | **Explanation** |
| **netstat** | **Execute the netstat command alone to show a relatively simple list of all active**[**TCP**](https://www.lifewire.com/transmission-control-protocol-and-internet-protocol-816255)**connections which, for each one, will show the local**[**IP address**](https://www.lifewire.com/what-is-an-ip-address-2625920)**(your computer), the foreign IP address (the other computer or network device), along with their respective port numbers, as well as the TCP state.** |
| **-a** | **This switch displays active TCP connections, TCP connections with the listening state, as well as UDP ports that are being listened to.** |
| **-b** | **This netstat switch is very similar to the -o switch listed below, but instead of displaying the PID, will display the process's actual file name. Using -b over -o might seem like it's saving you a step or two but using it can sometimes greatly extend the time it takes netstat to fully execute.** |
| **-e** | **Use this switch with the netstat command to show statistics about your network connection. This data includes bytes, unicast packets, non-unicast packets, discards, errors, and unknown protocols received and sent since the connection was established.** |
| **-f** | **The -f switch will force the netstat command to display the**[**Fully Qualified Domain Name**](https://www.lifewire.com/what-does-fqdn-mean-2625883)**(FQDN) for each foreign IP addresses when possible.** |
| **-n** | **Use the -n switch to prevent netstat from attempting to determine**[**host names**](https://www.lifewire.com/what-is-a-hostname-2625906)**for foreign IP addresses. Depending on your current network connections, using this switch could considerably reduce the time it takes for netstat to fully execute.** |
| **-o** | **A handy option for many troubleshooting tasks, the -o switch displays the process identifier (PID) associated with each displayed connection. See the example below for more about using netstat -o.** |
| **-p** | **Use the -p switch to show connections or statistics only for a particular *protocol*. You can not define more than one *protocol* at once, nor can you execute netstat with -p without defining a *protocol*.** |
| ***protocol*** | **When specifying a *protocol* with the -p option, you can use tcp, udp, tcpv6, or udpv6. If you use -s with -p to view statistics by protocol, you can use icmp, ip, icmpv6, or ipv6 in addition to the first four I mentioned.** |
| **-r** | **Execute netstat with -r to show the IP routing table. This is the same as using the route command to execute route print.** |
| **-s** | **The -s option can be used with the netstat command to show detailed statistics by protocol. You can limit the statistics shown to a particular protocol by using the -soption and specifying that *protocol*, but be sure to use -s before -p *protocol* when using the switches together.** |
| **-t** | **Use the -t switch to show the current TCP chimney offload state in place of the typically displayed TCP state.** |
| **-x** | **Use the -x option to show all NetworkDirect listeners, connections, and shared endpoints.** |
| **-y** | **The -y switch can be used to show the TCP connection template for all connection. You cannot use -y with any other netstat option.** |
| ***time\_interval*** | **This is the time, in seconds, that you'd like the netstat command to re-execute automatically, stopping only when you use**[**Ctrl-C**](https://www.lifewire.com/what-is-ctrl-c-used-for-2625834)**to end the loop.** |
| **/?** | **Use the**[**help switch**](https://www.lifewire.com/help-switch-2625896)**to show details about the netstat command's several options.** |

* 1. **Show Active TCP Connections:**
* netstat -f

It shows all active TCP connections.

Active Connections

 Proto Local Address Foreign Address State

 TCP 127.0.0.1:5357 VM-Windows-7:49229 TIME\_WAIT

 TCP 127.0.0.1:49225 VM-Windows-7:12080 TIME\_WAIT

 TCP 192.168.1.14:49194 75.125.212.75:http CLOSE\_WAIT

 TCP 192.168.1.14:49196 a795sm.avast.com:http CLOSE\_WAIT

 TCP 192.168.1.14:49197 a795sm.avast.com:http CLOSE\_WAIT

 TCP 192.168.1.14:49230 TIM-PC:wsd TIME\_WAIT

 TCP 192.168.1.14:49231 TIM-PC:icslap ESTABLISHED

 TCP 192.168.1.14:49232 TIM-PC:netbios-ssn TIME\_WAIT

 TCP 192.168.1.14:49233 TIM-PC:netbios-ssn TIME\_WAIT

 TCP [::1]:2869 VM-Windows-7:49226 ESTABLISHED

* 1. **Show Specific Connections Only:**

netstat -0 | findstr 28604

netstat command to show only the connections that are using a specific PID, *28604*in this example.

A similar command could be used to filter out the connections witha *CLOSE\_WAIT* state, by replacing the PID with *ESTABLISHED*.

1. Port Scan / nmap

Nmap is a free, open source and multi-platform network security scanner used for network discovery and security auditing. Amongst other things, it allows you to create a network inventory, managing service upgrade schedules, monitor host or service uptime and scan for open ports and services on a host.

Nmap can be extremely useful for helping you get to the root of the problem you are investigating, verify firewall rules or validate your routing tables are configured correctly.

To get started, download and install Nmap from the nmap.org website and then launch a command prompt.

Typing nmap [hostname] or nmap [ip\_address] will initiate a default scan. A default scan uses 1000 common TCP ports and has Host Discovery enabled.

Host Discovery performs a check to see if the host is online. In a large IP range, this is useful for identifying only active or interesting hosts, rather than scanning every single port on every single IP in the range (a lot of which may not even be there).

When the scan is complete, you should see an Nmap scan report similar to the one shown in the image above. This confirms Nmap is installed and operating correctly.

You will notice the information returned is PORT | STATE | SERVICE. Before we take a deeper dive into the commands, it would be valuable to know what the different ‘STATES’ mean.

STATE Description

Open The target port actively responds to TCP/UDP/SCTP requests.

Closed The target port is active but not listening.

Filtered A firewall or packet filtering device is preventing the port state being returned.

Unfiltered The target port is reachable but Nmap cannot determine if it is open or closed.

Open/Filtered Nmap cannot determine if the target port is open or filtered.

Closed/Filtered Nmap cannot determine if the target port is closed or filtered.

Let us now look at some commands we can use for scanning open ports.

Nmap Port Scanning Commands

The “–open” parameter

In any of the commands below, you can specify the “–open” parameter in your Nmap command to have Nmap only show you ports with an “Open” state.

nmap –open [ip\_address]

Scanning a single port

nmap -p 80 [ip\_address]

This command will initiate a default scan against the target host and look for port 80.

Scanning a specific range of ports

nmap -p 1-200 [ip\_address]

This command will initiate a default scan against the target host and look for portsbetween the range of 1-200.

Scanning the entire port range

nmap -p- [ip\_address]

This command will initiate a scan against the target host looking for all ports (1-65535).

Scanning the top 100 ports (fast scan)

nmap -F [ip\_address]

This command will initiate a fast scan against the target host looking only for the top 100common TCP ports.

Scanning multiple TCP/UDP ports

nmap -p U:53,67-68,T:21-25,80,135 [ip\_address]

This command will initiate a scan against the target host looking only for specified UDP and TCP ports.

Scanning for specific service names

nmap -p http,ssh,msrpc,microsoft-ds [ip\_address]

This command will initiate a scan against the target host looking for ports associated with specified service names.

TCP SYN scan (default)

nmap -sS [ip\_address]

This command will initiate a TCP SYN scan against the target host. A TCP SYN scan sends a SYN packet to the target host and waits for a response. If it receives an ACK packet back, this indicates the port is open. If an RST packet is received, this indicates the port is closed. If no response is received after multiple transmissions, the port is considered filtered (a device or application between the source and the target is filtering the packets).

TCP connect scan

nmap -sT [ip\_address]

This command will initiate a TCP connect scan against the target host. A TCP connect scan is the default scan performed if a TCP SYN scan is not possible. This type of scan requests that the underlying operating system try to connect with the target host/port using the ‘connect’ system call.

UDP port scan

nmap -sU [ip\_address]

This command will initiate a UDP port scan against the target host. A UDP scan sends a UDP packet to the target port(s). If a response is received, the port is classified as Open. If no response is received after multiple transmissions, the port is classified as open/filtered.

SCTP INIT scan

nmap -sY [ip\_address]

This command will initiate an SCTP INIT scan against the target host. An SCTP INIT scan is similar to the TCP SYN scan but specific to the SCTP protocol. An INIT chunk is sent to the target port(s). If an INIT-ACK chunk is received back, the port is classified as open. If an ABORT chunk is received, the port is classified as closed. If no response is received after multiple transmissions, the port is classified as filtered.

Zenmap

The Nmap installation package comes with a front-end GUI for Nmap called Zenmap, used to control Nmap from a user interface rather than a command-line.

One of the key benefits of using the GUI front-end version is the ability to save scanning profiles. You can configure a profile to include customized scanning options, scan targets, ping options, scripts, and performance options.

You may wish to create a new profile before initiating a scan. To do this, go to Profile > New Profile or Command.

A default list of scan profiles is included and the description and options for each can be reviewed from Profile > Edit Selected Profile.

To kick off a scan, enter the target to be scanned and choose a scan profile before clicking ‘Scan’.

When the scan is complete, the results will be displayed in the ‘Nmap Output’ tab with a further breakdown available in the Ports/Hosts, Topology, Host Details, and Scans tabs.

Go to the Ports/Hosts tab for a detailed list of all the open ports found on the target host.

Zenmap saves a history of your scans and allows you to compare two scans side-by-side. To do this, go to Tools > Compare Results. This is useful for eyeballing whether two hosts have the same list of open ports.

1. **Route command**

The route command is used to manually configure the routes in a routing ta Route is an external command that is available for the following Microsoft operating systems as route.executable such as Windows 10.

>route PRINT

rout ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1 METRIC 3 IF2  
 ^destination ^mask ^gateway ^metric ^interface

When IF is not given, it tries to find the best interface for the gateway.

>route PRINT

>route PRINT 157\* .... Only prints those matching 157\*

>route DELETE 157.0.0.0

>route PRINT

One way to use this would be as follows: You can't ping the server that you are connecting to, but you know the IP address to be 127.16.16.10.

>route PRINT

Interface List  
0x1 ........................... MS TCP Loopback interface  
0x2 ...00 14 a4 c3 44 20 ...... Xircom CardBus Ethernet 10/100 Adapter  
0x3 ...00 b0 d0 43 55 a5 ...... 3Com EtherLink PCI  
0x4 ...00 01 b0 8f 8f80 ......NdisWan Adapter

Active Routes:  
Network Destination Netmask Gateway Interface Metric  
0.0.0.0 0.0.0.0 127.16.8.14 127.16.8.14 1  
127.0.0.0 255.0.0.0 127.0.0.1 127.0.0.1 1  
127.16.0.0 255.255.0.0 127.16.8.14 127.16.8.14 1  
127.16.8.14 255.255.255.255 127.0.0.1 127.0.0.1 1  
192.168.50.0 255.255.255.0 192.168.50.65 192.168.50.65 2  
192.168.50.65 255.255.255.255 127.0.0.1 127.0.0.1 1  
192.168.50.255 255.255.255.255 192.168.50.65 192.168.50.65 1  
224.0.0.0 224.0.0.0 127.16.8.14 127.16.8.14 1  
224.0.0.0 224.0.0.0 192.168.50.65 192.168.50.65 1  
255.255.255.255 255.255.255.255 192.168.50.65 192.168.50.65 1

Notice that no gateway for the current IP goes to 255.255.255.0, so it must be added. Now enter the following command.

>route ADD 127.16.0.0 MASK 255.255.255.0 <your current IP from winntcfg or winipcfg> METRIC 1

Next, enter the following command:

>route print

Active Routes:  
Network Destination Netmask Gateway Interface Metric  
0.0.0.0 0.0.0.0 127.16.8.14 127.16.8.14 1  
127.0.0.0 255.0.0.0 127.0.0.1 127.0.0.1 1  
127.16.0.0 255.255.0.0 127.16.8.14 127.16.8.14 1  
\*\* 127.16.0.0 255.255.255.0 127.16.8.14 127.16.8.14 1  
127.16.8.14 255.255.255.255 127.0.0.1 127.0.0.1 1  
192.168.50.0 255.255.255.0 192.168.50.65 192.168.50.65 2  
192.168.50.65 255.255.255.255 127.0.0.1 127.0.0.1 1  
192.168.50.255 255.255.255.255 192.168.50.65 192.168.50.65 1  
224.0.0.0 224.0.0.0 127.16.8.14 127.16.8.14 1  
224.0.0.0 224.0.0.0 192.168.50.65 192.168.50.65 1  
255.255.255.255 255.255.255.255 192.168.50.65 192.168.50.65 1

Notice the \*\* IP address gives me the default gateway.

## Use the route Command

You can use the **route** command to view, add and delete routes on a Microsoft Windows NT server that runs Cisco ICM. You can use these options with the **route** command:

**route** [-f] [-p] [command [destination] [mask subnetmask] [gateway] [metric costmetric]]

### 

### Command Options

This section explains each of the options that you can use with the **route** command.

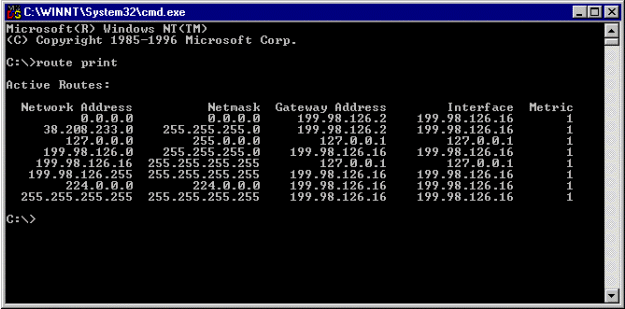
* The **-f** option clears the routing tables of all gateway entries. If you use the **-f** option in conjunction with one of the commands, the tables are cleared before you run the command.
* By default, routes are not preserved when you restart the system. Use the **-p** option with the **add** command to make a route persistent. Use the **-p** option with the **print** command to view the list of registered persistent routes.
* The **command** option specifies one of the six commands in this table:

|  |  |
| --- | --- |
| **Command** | **Definition** |
| **print** | Prints a route |
| **add** | Adds a route |
| **delete** | Deletes a route |
| **change** | Modifies an existing route |
| **destination** | Specifies the computer to send command |
| **mask subnetmask** | Specifies a subnet mask to be associated with this route entry (default subnet mask is 255.255.255.255) |

* The **destination** specifies the network destination of the route. The destination can be an IP network address, an IP address for a host route, or a default route.
* A **netmask** is a 32-bit mask that you can use to divide an IP address into subnets and specify the available hosts in the network. If you do not specify a netmask the default value 255.255.255.255 applies.
* The **gateway** option specifies the default gateway. All symbolic names used for the destination or gateway are looked up in the network and computer name database files NETWORKS and HOSTS. If the command is **print** or **delete**, you can use wildcards for the destination and gateway, or you can omit the gateway.
* The **metric** option assigns an integer cost metric (that ranges from 1 to 9999) which you can use to calculate the fastest, most reliable, and least expensive routes.

**"IF"**specifies the interface index for the interface over which the destination is reachable. If you do not specify **IF**, an attempt is made to find the best interface for a given gateway.

Here is an example of the **route** command:

[](https://www.cisco.com/c/dam/en/us/support/docs/voice-unified-communications/unified-intelligent-contact-management-enterprise/20524-route-command.gif)

### 

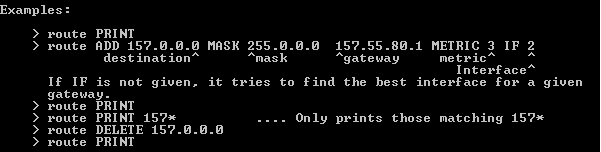
### **Examples:**

In order to view the entire contents of the IP routing table, issue the **route print** command.

In order to add a persistent route to the destination 10.19.0.0 with the subnet mask of 255.255.0.0 and the next hop address of 10.10.0.1, issue the **route -p add 10.19.0.0 mask 255.255.0.0 10.10.0.1** command.

In order to view the routes in the IP routing table that begin with "172.", issue the **route print 172.\*** command.

In order to delete all routes in the IP routing table that begin with "172.", issue the **route delete 172.\*** command.

[](https://www.cisco.com/c/dam/en/us/support/docs/voice-unified-communications/unified-intelligent-contact-management-enterprise/20524-route-command2.gif)

**Conclusion:**We successfully demonstrated basic network commands and TCP/IP utilities.

Upon completion of Experiment students will be able to:

1. Scan the network
2. Trubleshoot the network

# EXPERIMENT NO.3

**Implementation of small network using Hub, switch and Router using Cisco Packet Tracer.**

**Practical Session Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 20 | Testing of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates,  Discussed | Knowledge, Communication,  intrapersonal, Comprehension |

**Title:** Implementation of small network using Hub, switch and Router using Cisco Packet Tracer.

**Objectives:** • Students will able to understand working of switching devices and differences between those.

**Aim/ Problem Statement**: To study and implement the working of following networking devices:

1) Switch

2) Hub

3) Router

**Theory:**

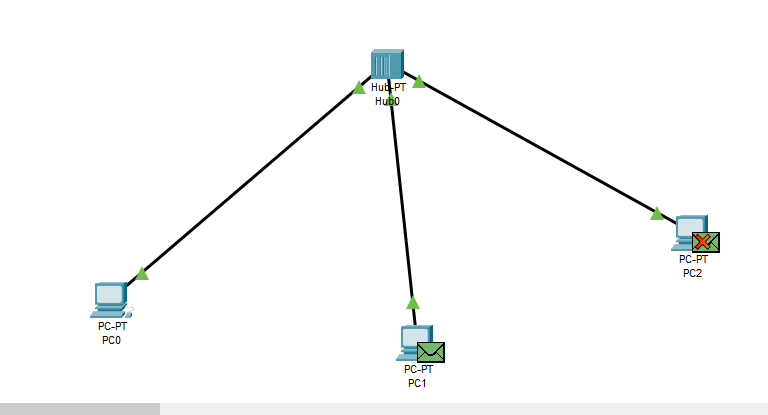
**1) Switch:**

A switch is a network device. Operating at Layer 2 i.e. Data Link layer of the OSI model. Switch connects devices in a network and uses data packet switching to send and receive over the network. A switch has many ports, to which computers are plugged in. However, when a data frame arrives at any port of a network der switch, it examines the destination address and sends the frame to the corresponding device (s). Thus, it Supports unicast and multicast communications.

**2) Hub:**

Hubs are networking devices operating at Physical Layer of the OSI model that are used to connect in a network. A hub is less sophisticated than a switch, the latter of which can isolate data transmissions to specific devices.They are generally used to multiple devices connect computers in a LAN.A hub has many ports in it. A computer which intends to ` be connected to the network is plugged in to one of these ports.When data a frame arrives at a port, it is broadcast to every other port, without considering whether it is destined for a particular destination or not.

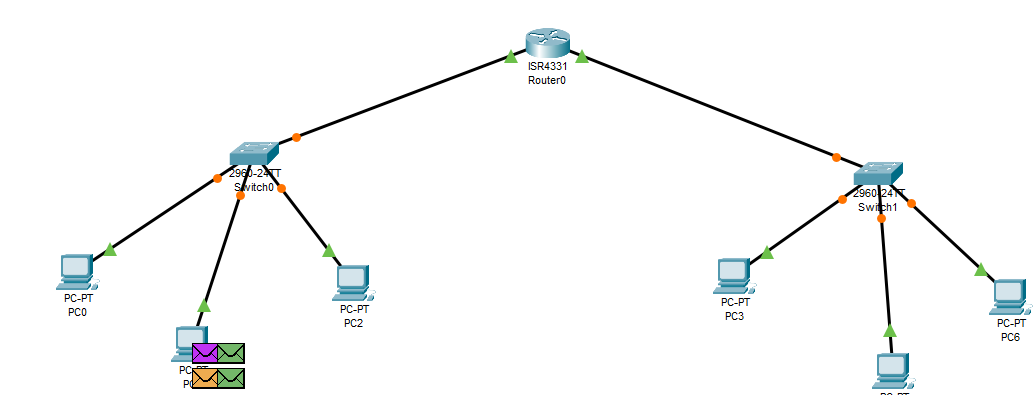
****

****

**3) Router:**

Routers are networking devices operating at layer 3 or a Network layer of the OSI model. These are responsible for receiving, analyzing and forwarding data packets among the connected networks.

* When a data packet arrives, the router inspects the destination address, consults its routing tables to decide the optimal route and then transfers the packet along this route.
* Router transfers data in the form of IP packets. In order to transmit data, it uses IP and address mentioned in the field of the IP packet.
* A router is more expensive than other networking devices like hubs, switches, etc.

****

* **Difference Between Hub and Switch:**

|  |  |
| --- | --- |
| **HUB** | **SWITCH** |
| 1) The main objective of a Hub is to transmit the signal to a port, which will respond to where the signal was received. | 1) A switch allows you to set up and terminate connections as needed. |
| 2) Hubs operate at the Physical Layer | 2) Switches function at the Data Link Layer. |
| 3) Hubs use broadcast type transmission. | 3) Switches use unicast, multicast as well as broadcast type transmission. |
| 4) Hub can have maximum 4 ports. | 4) Switch can have 24 to 28 ports. |
| 5) There is only one collision domain in a Hub. | 5) In a Switch, each port has its own collision domain. |
| 6) Hubs do not provide packet filtering. | 6) Switches provide packet filtering. |
| 7) Hub uses half duplex transmission mode. | 7) Switch uses full duplex transmission mode. |

**Conclusion:**

We have successfully implemented and studied various networking devices like Switch, Hub and Router.

Upon completion of Experiment students will be able to:

1. Understand working of Hub, Switch and Router
2. Students will be able to design networks using these devices.

# EXPERIMENT NO. 4

**Using simulator build a small network (P2P, Client/Server) implement network topology (Star, Bus, Ring) and IP addressing and subnet masking.**

**Practical Session Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 20 | Testing of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates,  Discusses | Knowledge, Communication,  intrapersonal,Comprehension |

**Title:** Using simulator build a small network (P2P, Client/Server) implement network topology (Star, Bus, Ring) and IP addressing and subnet masking.

**Objectives:** • Implementation of network topology, IP addressing and subnet masking to build a small networks using a simulator.

**Aim/ Problem Statement**: Using simulator build a small network (P2P, Client/Server) implement network topology (Star, Bus, Ring) and IP addressing and subnet masking.Study Client/server topology and different topologies.

**Theory:**

A network is a collection of end devices and networks devices connected to each other for communication and sharing.

1) Star

2) Bus

3) Ring

1. **Star Topology:**

Star topology is a [network topology](https://www.techopedia.com/definition/5538/network-topology) in which each network component is physically connected to a central node such as a router, hub or switch. In a star topology, the central hub acts like a server and the connecting nodes act like clients. When the central node receives a [packet](https://www.techopedia.com/definition/5380/packet) from a connecting node, it can pass the packet on to other nodes in the network. A star topology is also known as a star network.Star networks require a [point-to-point connection](https://computernetworktopology.com/point-to-point-topology/) between the central node and connecting devices. To improve communication between the devices on the network, the central node can provide signal reconditioning and amplification services.

**Advantages of Star Topology**

* It is very reliable – if one cable or device fails then all the others will still work
* It is high-performing as no data collisions can occur
* Less expensive because each device only need one I/O port and wishes to be connected with hub with one link.
* Easier to put in
* Robust in nature
* Easy fault detection because the link are often easily identified.
* No disruptions to the network when connecting or removing devices.
* Each device requires just one port i.e. to attach to the hub.
* If N devices are connected to every other in star, then the amount of cables required to attach them is N. So, it’s easy to line up.

**Disadvantages of Star Topology**

* Requires more cable than a linear bus.
* If the connecting network device (network switch) fails, nodes attached are disabled and can’t participate in network communication.
* More expensive than linear bus topology due to the value of the connecting devices (network switches)
* If hub goes down everything goes down, none of the devices can work without hub.
* Hub requires more resources and regular maintenance because it’s the central system of star.
* Extra hardware is required (hubs or switches) which adds to cost
* Performance is predicated on the one concentrator i.e. hub.

**2) Bus Topology:**

Bus topology is a network type in which every computer and network device is connected to a single cable. It transmits the data from one end to another in a single direction. No bi-directional feature is in bus topology. It is a multi-point connection and a non-robust topology because if the backbone fails the topology crashes. In Bus Topology, various MAC (Media Access Control) protocols are followed by LAN Ethernet connections like TDMA, Pure Aloha, CDMA, Slotted Aloha, etc.

**Advantages of bus topology:**

* If N devices are connected to each other in a bus topology, then the number of cables required to connect them is 1, which is known as backbone cable, and N drop lines are required.
* The cost of the cable is less compared to other topologies, but it is used to build small networks.

**Disadvantages of bus topology:**

* If the common cable fails, then the whole system will crash down.
* If the network traffic is heavy, it increases collisions in the network. To avoid this, various protocols are used in the MAC layer known as Pure Aloha, Slotted Aloha, CSMA/CD, etc.
* Security is very low.

**3) Ring Topology:**

In this topology, it forms a ring connecting devices with exactly two neighboring devices.A number of repeaters are used for Ring topology with a large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the network.The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called Dual Ring Topology. In-Ring Topology, the Token Ring Passing protocol is used by the workstations to transmit the data.

**Advantages of ring topology:**

* The possibility of collision is minimum in this type of topology.
* Cheap to install and expand.

**Disadvantages off ring topology:**

* Troubleshooting is difficult in this topology.
* The addition of stations in between or removal of stations can disturb the whole topology.
* Less secure

**Output with sub-netting:**



**4) Peer to peer network:**

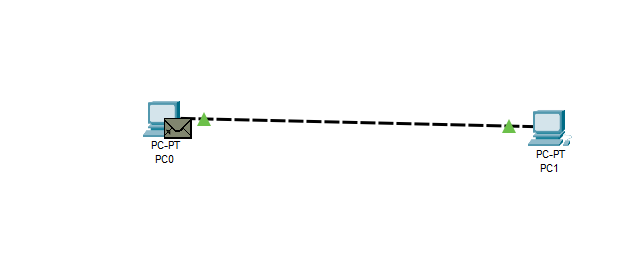
A peer to peer network is a simple network of computers. It first came into existence in the late 1970s. Here each computer acts as a node for file sharing within the formed network. Here each node acts as a server and thus there is no central server to the network. This allows the sharing of a huge amount of data. The tasks are equally divided amongst the nodes. Each node connected in the network shares an equal workload. For the network to stop working, all the nodes need to individually stop working. This is because each node works independently.

**Advantages of P2P Network:**

* Network is easy to maintain because each node is independent of each other.
* Since each node acts as a server, therefore the cost of the central server is saved.
* Adding, deleting and repairing nodes in this network is easy.

**Disadvantages of P2P Network:**

* Because of no central server, data is always vulnerable to get lost because of no backup.
* It becomes difficult to secure the complete network because each node is independent



**5) Client-Server Network:**

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

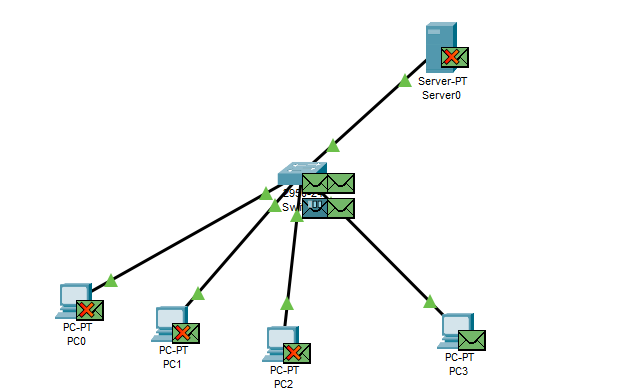
* **Client:** When we talk the word Client, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a Client is a computer (Host) i.e. capable of receiving information or using a particular service from the service providers (Servers).
* **Servers:** Similarly, when we talk the word Servers, It mean a person or medium that serves something. Similarly in this digital world a Server is a remote computer which provides information (data) or access to particular services.

**Advantages of Client-Server network:**

* Centralized system with all data in a single place.
* Cost efficient requires less maintenance cost and Data recovery is possible.
* The capacity of the Client and Servers can be changed separately.

**Disadvantages of Client-Server network:**

* Clients are prone to viruses, Trojans and worms if present in the Server or uploaded into the Server.
* Server are prone to Denial of Service (DOS) attacks.
* Data packets may be spoofed or modified during transmission.
* Phishing or capturing login credentials or other useful information of the user are common and MITM (Man in the Middle) attacks are common.



**What is sub-netting?**

Sub-netting is a technique that is used to divide the individual physical network into a smaller size called sub-networks. These sub-networks are called a subnet. An internal address is made up of a combination of the small networks segment and host segment. A sub network is designed by accepting the bits from the IP address host portion; then, they are uses to assign a number of small-sized sub-networks in the original network.

**What is super-netting?**

Super-netting is the process that is used to combine several sub networks into a single network. Its process is inverse of the sub-netting process. In super-netting, mask bits are moved towards the left of the default mask; network bits are converted into hosts bits. Super-netting is also called router summarization and aggregation. It creates a more number of host addresses at the expense of network addresses. The Internet service provider performs the super-netting process to achieve the most efficient IP address allocation.

**Conclusion:**

We have successfully simulated a small network using topologies and assigned suitable subnet mask and IP address to it.

Upon completion of Experiment students will be able to:

* 1. Understand Client/ Server architecture
  2. Design different network topologies

# EXPERIMENT NO.5

**To write a C program to perform sliding window.**

**Practical Session Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 20 | Testing of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates,  Discusses | Knowledge, Communication,  intrapersonal, Comprehension |

**Title:** To write a C program to perform sliding window.

**Objectives:** • Students will able to understand Go back N and Selective Repeat Modes of Sliding Window Protocol

**Aim/ Problem Statement**: Write program to implement Siding Window protocol

**Theory:**Data-link layer is responsible for implementation of point-to-point flow and error control mechanism.

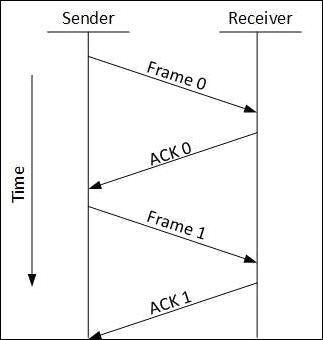
**Flow Control**

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.

Two types of mechanisms can be deployed to control the flow:

1. **Stop and Wait**

This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.



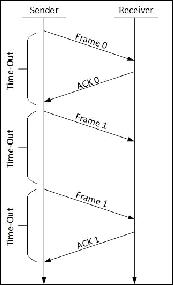
**2. Sliding Window**

In this flow control mechanism, both sender and receiver agree on the number of data-frames after which the acknowledgement should be sent. As we learnt, stop and wait flow control mechanism wastes resources, this protocol tries to make use of underlying resources as much as possible.

**Error Control**

When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted. In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.In such case, both sender and receiver are equipped with some protocols which helps them to detect transit errors such as loss of data-frame. Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Requirements for error control mechanism:

* **Error detection** - The sender and receiver, either both or any, must ascertain thatthere is some error in the transit.
* **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
* **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, itsends a NACK back to the sender and the sender must retransmit the correct frame.
* **Retransmission:** The sender maintains a clock and sets a timeout period. If anacknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it’s acknowledgement is lost in transit.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

1. **Stop-and-wait ARQ**

The following transition may occur in Stop-and-Wait ARQ:

* + The sender maintains a timeout counter.
  + When a frame is sent, the sender starts the timeout counter.
  + If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
  + If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
  + If a negative acknowledgement is received, the sender retransmits the frame.

1. **Go-Back-N ARQ**

Stop and wait ARQ mechanism does not utilize the resources at their best. When the acknowledgement is received, the sender sits idle and does nothing. In Go-Back-N ARQ method, both sender and receiver maintain a window. The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame’s sequence number.

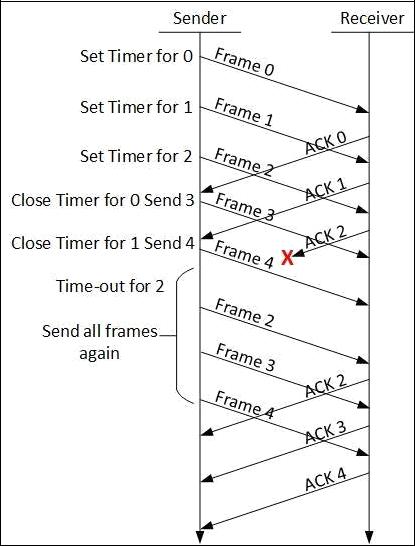
When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement. If all frames are positively acknowledged, the sender sends next set of frames. If sender finds that it has received NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

**3. Selective Repeat ARQ**

In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.

In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.

The sender in this case, sends only packet for which NACK is received.



#include<stdio.h>

#include<stdlib.h>

int main()

{

srand(time(0));

intrandNo = rand();

int count=0;

printf( "%d", randNo);

intwsize, frameNum;

printf(" Enter Frame Number : ");

scanf("%d", &frameNum);

printf(" Enter window size : ");

scanf("%d", &wsize);

//for(inti=0; i<frameNum; i++){

for(int j=1; j<=wsize; j++){

printf("sending frame %d \n", j);

}

for(inti=1; i<=wsize; i++){

if(i==rand()%wsize+1){

printf("ackonowledgemengtreceviewd %d\n",i);

count++;

printf("sending next frame %d \n",wsize+count);

//break;

}

else{

printf("ackon. notreceview %d\n", i);

printf("Resending farme %d\n",i);

}

}

return 0;

}

**Conclusion:** Hence we have studied Go back N and Selective Repeat Modes of Sliding Window Protocol

**Expected Input:**Enter Window size: 5

Enter number of frames: 10

**Expected Output/Results:**

**EXPERIMENT NO.6**

**Socket Programming in C/C++ a) TCP Client , TCP Server b) UDP Client , UDP Server.**

**Practical Session Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 20 | Testing of program | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Discuss, participates | Knowledge, Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates, | Listens, Participates,  Discusses | Knowledge, Communication,  intrapersonal, Comprehension |

**Title:** Socket Programming in C/C++ a) TCP Client , TCP Server b) UDP Client , UDP Server.

**Objectives:** • Students will able to understand socket programming.

Students will able to design networking applications using TCP protocol.

Students will able to design networking applications using UDP protocol.

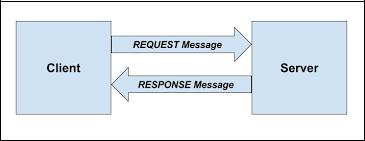
**Aim/ Problem Statement**: Write a program to implement TCP and UDP sockets using socket programming

**Theory:**

**Client-Server Model**

Network applications can be divided into two process: a Client and a Server, with a

communication link joining the two processes.



Normally, from Client side it is one-one connection. From the Server Side, it is many-one

Connection. The standard model for network applications is the Client-Server model. A

Server is a process that is waiting to be contacted by a Client process so that server can do something for the client. Typical BSD Sockets applications consist of two separate application level processes; one process (the client) requests a connection and the other process (the server) accepts it.

**Client-Server Model Using TCP**

TCP Clients sends request to server and server will receives the request and response with acknowledgement. Every time client communicates with server and receive response from it. Algorithm to create client and server process is as below.

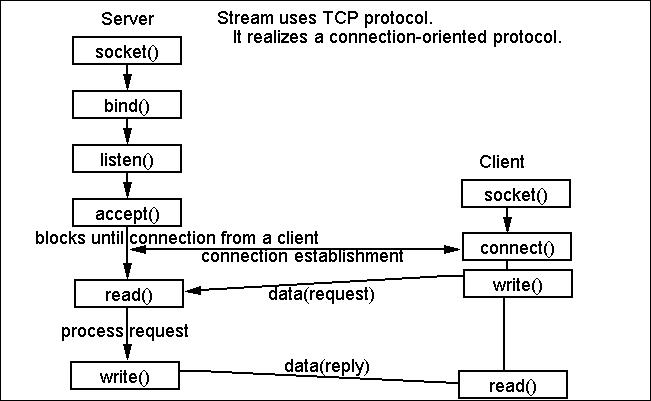
**ALGORITHM:**

**Server**

1. Create a server socket and bind it to port
2. Listen for new connection and when a connection arrives, accept it
3. Read Client's message and display it
4. Get a message from user and send it to client
5. Close the server socket
6. Stop

**Client**

1. Create a client socket and connect it to the server‟s port number
2. Get a message from user and send it to server
3. Read server's response and display it
4. Close the client socket
5. Stop

**Socket functions for TCP client/server in Connection-oriented Scenario**

**Elementary Socket System Calls**

**1. socket() socket() System Call:** Creates an end point for communication and returns adescriptor.

**#include <sys/socket.h>**

**#include <sys/types.h>**

**int socket ( int Address Family, intType,intProtocol);**

Return Values: Upon successful completion, the socket subroutine returns an

integer (the socket descriptor). It returns -1 on error.

|  |  |  |
| --- | --- | --- |
| **2. bind( )** | **bind( ) System call:** Binds a name to a socket. |  |
|  | **Description:** The **bind** subroutine assigns a Name parameter to an unnamed | |
|  | socket. It assigns a local protocol address to a socket. |  |
|  | **#include <sys/socket.h>** |  |
|  | **int bind (intsockfd, structsockaddr \*myaddr, intaddrlen);** |  |
|  |  |  |

Return Values: Upon successful completion, the bind subroutine returns a value of 0. Otherwise, it returns a value of -1 to the calling program.

1. **connect() connect() System call:** The connect function is used by a TCP client toestablish a connection with a TCP server.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **#include <sys/socket.h>** | | |  |  |  |
|  | **int connect(intsockfd, structsockaddr \*servaddr, intaddrlen);** | | | | |  |
|  | Return Values: Upon successful completion, the connect subroutine returns a | | | | |  |
|  |  |  |  | |  |  |
|  | value of 0. Otherwise, it returns a value of -1 to the calling program. | | | | |  |
| **4. listen()** | **listen() System call:** This system call is used by a connection-oriented server | | | | |
|  | to indicate that it is willing to receive connections. | | | | |
|  | **#include <sys/socket.h>** | | |  |
|  | **int listen (intsockfd, int backlog);** | | |  |
|  | Return values: Returns 0 if OK, -1 on error | | |  |
| **5. accept()** | **accept() System call:** The actual connection from some client process is | | | | |
|  | waited for by having the server execute the accept system call. | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **#include <sys/socket.h>** | | |  |  |  |
|  | **int accept (intsockfd, structsockaddr \*cliaddr, int \*addrlen);** | | | | |  |
|  | Return Values: This system call returns up to three values: an integer return | | | | |  |
|  |  | |  | |  |  |
|  | code that is either a new socket descriptor or an error indication, the protocol | | | | |  |
|  | address of the client process (through the cliaddr pointer), and the size of this | | | | |  |
|  | address (through the addrlen pointer). | | |  |  |  |
|  |  | | |  | |  |
|  |  | | |  | |  |
| **6. read()** | **read() and write() System call:** - read/write | | | from a file descriptor | |  |
| **and write()** | **#include <unistd.h>** | | |  |  |  |
|  | **ssize\_t read(int***fd***, void \****buf***, size\_t***count***);** | | |  |  |  |
|  | **ssize\_t write(int***fd***, const void \****buf***, size\_t***count***);** | | | | |  |
|  | Return Values: On success, the number of bytes read or written is returned | | | | |  |
|  |  |  | | |  |  |
|  | (zero indicates nothing was written/read). On error, -1 is returned. | | | | |  |
|  |  | | | | |  |
| **7. Send( ),** | S**end( ),sendto( ),recv( ) and recvfrom( ) system calls:** | | | | |  |
| **sendto( ),** | These system calls are similar to the standard read and write functions, but one | | | | |  |
| **recv( )** | additional argument is required. | | |  |  |  |
|  |  |  |  |  |  |
|  |  | | |  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **and** | **#include <sys/socket.h>** |  |  |
|  | **recvfrom( )** | **int send(intsockfd, char \*buff, intnbytes, int flags);** | | |
|  |  | **intsendto(intsockfd, char void \*buff, intnbytes, int flags, structsockaddr** | | |
|  |  | **\*to, intaddrlen);** |  |  |
|  |  | **intrecv(intsockfd, char \*buff, intnbytes, int flags);** | | |
|  |  | **intrecvfrom(intsockfd, char \*buff, intnbytes, int flags, structsockaddr** | | |
|  |  | **\*from, int \*addrlen);** |  |  |

The first three arguments, sockfd, buff and nbytes are the same as the first three arguments to read and write. The flags argument is either 0 or is formed by logically OR'ing one or more of the constants.

Return Values: All four system calls return the length of the data that was written or read as the value of the function. Otherwise it returns, -1 on error.

**8. Close( ) Close( ) system call:** The normal Unix close function is also used to close asocket and terminate a TCP connection.

**#include <unistd.h>**

**int close (intsockfd);**

## TCP Client.c:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

void error(const char \*msg)

{

perror(msg);

exit(0);

}

int main(intargc, char \*argv[])

{

intsockfd, portno, n;

structsockaddr\_inserv\_addr;

structhostent \*server;

char buffer[256];

if (argc< 3) {

fprintf(stderr,"usage %s hostname port\n", argv[0]);

exit(0);

}

portno = atoi(argv[2]);

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd< 0)

error("ERROR opening socket");

server = gethostbyname(argv[1]);

if (server == NULL) {

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr,

(char \*)&serv\_addr.sin\_addr.s\_addr,

server->h\_length);

serv\_addr.sin\_port = htons(portno);

if (connect(sockfd,(structsockaddr \*) &serv\_addr,sizeof(serv\_addr)) < 0)

error("ERROR connecting");

printf("Please enter the message: ");

bzero(buffer,256);

fgets(buffer,255,stdin);

n = write(sockfd,buffer,strlen(buffer));

if (n < 0)

error("ERROR writing to socket");

bzero(buffer,256);

n = read(sockfd,buffer,255);

if (n < 0)

error("ERROR reading from socket");

printf("%s\n",buffer);

close(sockfd);

return 0;

}

client cheerios 51717

## TCP Server.c

/\* A simple server in the internet domain using TCP

The port number is passed as an argument \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

void error(const char \*msg)

{

perror(msg);

exit(1);

}

int main(intargc, char \*argv[])

{

intsockfd, newsockfd, portno;

socklen\_tclilen;

char buffer[256];

structsockaddr\_inserv\_addr, cli\_addr;

int n;

if (argc< 2) {

fprintf(stderr,"ERROR, no port provided\n");

exit(1);

}

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd< 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(argv[1]);

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = INADDR\_ANY;

serv\_addr.sin\_port = htons(portno);

if (bind(sockfd, (structsockaddr \*) &serv\_addr,

sizeof(serv\_addr)) < 0)

error("ERROR on binding");

listen(sockfd,5);

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd,

(structsockaddr \*) &cli\_addr,

&clilen);

if (newsockfd< 0)

error("ERROR on accept");

bzero(buffer,256);

n = read(newsockfd,buffer,255);

if (n < 0) error("ERROR reading from socket");

printf("Here is the message: %s\n",buffer);

n = write(newsockfd,"I got your message",18);

if (n < 0) error("ERROR writing to socket");

close(newsockfd);

close(sockfd);

return 0;

}

server 51717

**Server-Client file transfer using UDP**

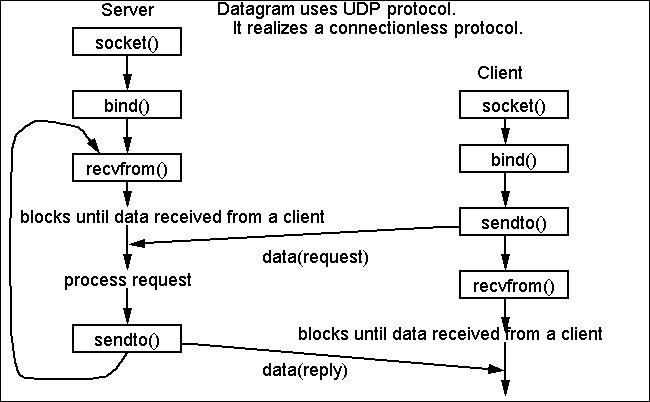
**Server:**

* Include appropriate header files.
* Create a UDP Socket using socket() system call.
* Fill in the socket address structure (with server information)
* Specify the port where the service will be defined to be used by client.
* Bind the address and port using bind() system call
* Receive a file name of text, audio or video from the Client using recvfrom() system call.
* Sends file to client using sendto() system call.
* Close the server socket
* Stop

**Client:**

* Include appropriate header files.
* Create a UDP Socket.
* Fill in the socket address structure (with server information)
* Specify the port of the Server, where it is providing service
* Send a file name to the server using sendto() system call.
* Receive a file from the Server using recvfrom() system call.
* Close the client socket
* Stop

**Socket functions for UDP client/server in Connectionless Scenario**



#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<arpa/inet.h>

int main()

{

int port =1234;

intsockfd;

structsockaddr\_insi\_me,si\_other;

char buffer[1024];

socklen\_taddr\_size;

sockfd= socket(AF\_INET, SOCK\_DGRAM,0);

memset(&si\_me,'\0',sizeof(si\_me));

si\_me.sin\_family= AF\_INET;

si\_me.sin\_port=htons(port);

si\_me.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

bind(sockfd,(structsockaddr\*)&si\_me,sizeof(si\_me));

addr\_size=sizeof(si\_other);

recvfrom(sockfd, buffer,1024,0,(structsockaddr\*)&si\_other,&addr\_size);

printf("[+]Data Received: %s", buffer);

return0;

}

## UDP client.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main()

{

int port = 1234;

intsockfd;

structsockaddr\_inserverAddr;

char buffer[1024];

socklen\_taddr\_size;

sockfd = socket(PF\_INET, SOCK\_DGRAM, 0);

memset(&serverAddr, '\0', sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(port);

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

strcpy(buffer, "Hello Server\n");

sendto(sockfd, buffer, 1024, 0, (structsockaddr\*)&serverAddr, sizeof(serverAddr));

printf("[+]Data Send: %s", buffer);

return 0;

**Conclusion:** Thus we have implemented socket programming to create TCP and UDP sockets.

## Upon completion of Experiment students will be able to:

## Implement TCP socket to transfer message from one machine to another machine

## Implement UDP socket to transfer message from one machine to another machine

**EXPERIMENT NO.7**

**Using any simulator configure: 1.Subnetting of a given network 2. Super netting of given network.**

**Practical Session Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm  and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of polygon filling program. | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20 | Testing of  program | Keyboard, mouse,  computers. | Introduces,  Facilitates, Explains | Discuss,  participates | Knowledge,  Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates, Discusses | Knowledge, Communication, intrapersonal, Comprehension |

**Title:** Using any simulator configure: 1.Subnetting of a given network 2. Super netting of given network.

**Objectives:** To implement sub-netting

**Aim/ Problem Statement**: Using any simulator configure: 1.Subnetting of a given network 2. Super netting of given network.

**Theory:**

Design subnet works for an institute with following configuration:

IT Block= 136 Hosts

Science Block= 95 Hosts

Arts Block= 51 Hosts

PG Block= 24 Hosts

===Solution:

**1) IT Block- 136 Hosts**

28=**256**

Total Bits of IP address=**32**

32-8= **24**

Subnet mask= 24bits

**192.168.1.0/24**

IP: **192.168.1.0**

Subnet Mask: **255.255.255.0**

First host IP: **192.168.1.1**

Last host IP: **192.168.1.254**

Router Conf. IP: 192.168.1.95 (**any IP between first and last host IP**)

**2) Science Block- 95 Hosts**

27=**128**

Total Bits of IP address=**32**

32-7= **25**

Subnet mask= 25bits

**192.168.2.0/25**

IP: **192.168.2.0**

Subnet Mask: **255.255.255.128**

First host IP: **192.168.2.1**

Last host IP: **192.168.2.126**

Router Conf. IP: 192.168.2.95 (**any IP between first and last host IP**)

**3) Arts Block- 51 Hosts**

26=**64**

Total Bits of IP address=**32**

32-6= **26**

Subnet mask= 26bits

**192.168.1.0/26**

IP: **192.168.2.128**

Subnet Mask: **255.255.255.192**

First host IP: **192.168.2.129**

Last host IP: **192.168.2.190**

Router Conf. IP: 192.168.2.185 (**any IP between first and last host IP**)

**4) PG Block- 24 Hosts**

25=**32**

Total Bits of IP address=**32**

32-5= **27**

Subnet mask= 27bits

**192.168.2.0/27**

IP: **192.168.2.192**

Subnet Mask: **255.255.255.224**

First host IP: **192.168.1.143**

Last host IP: **192.168.1.222**

Router Conf. IP: 192.168.2.215 (**any IP between first and last host IP**)

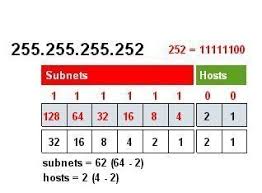
Configure the all the department IP and subnet-mask.



**Apply a Sub-netting:**

Sub-netting works by applying the concept of extended network addresses to individual computer (and another network device) addresses. An extended network address includes both a network address and additional bits that represent the subnet number.

Together, these two data elements support a two-level addressing scheme recognized by standard implementations of IP. The network address and subnet number, when combined with the host address, support a three-level scheme.



Consider the following real-world example: A small business plans to use the 192.168.1.0 network for its internal (intranet) hosts. The human resources department wants their computers to be on a restricted part of this network because they store payroll information and other sensitive employee data. But because this is a Class C network, the default subnet mask of 255.255.255.0 allows all computers on the network to be peers (to send messages directly to each other) by default.

The first four bits of 192.168.1.0:

1100

This places the network in the Class C range and also fixes the length of the network address at 24 bits. To subnet this network, more than 24 bits must be set to **1** on the left side of the subnet mask.

For every additional bit set to **1** in the mask, another bit becomes available in the subnet number to index additional subnets. A two-bit subnet number can support up to four subnets, a three-bit number supports up to eight subnets, and so on.

**Conclusion:**

We have successfully implemented and learned about concept of sub-netting.

**EXPERIMENT NO.8**

Simulation of Distance Vector/ Link State Routing algorithm.

**Practical Session Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time (min)** | **Content** | **Learning Aid / Methodology** | **Faculty Approach** | **Typical Student Activity** | **Skill /**  **Competency Developed** |
| 20 | Explanation of Experiment | Chalk & Talk , Presentation | Introduces, Facilitates, Explains | Listens | Knowledge, Communication, intrapersonal, Application |
| 10 | Preparation of Algorithm  and logic development for program | Paper work | Introduces, Facilitates, Explains | Discuss | Knowledge, Communication, intrapersonal, Application |
| 60 | Coding of polygon filling program. | Keyboard, mouse, computers. | Introduces, Facilitates, Explains | Discuss, participates | Knowledge, Communication, intrapersonal, Application |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 20 | Testing of  program | Keyboard, mouse,  computers. | Introduces,  Facilitates, Explains | Discuss,  participates | Knowledge,  Communication, intrapersonal, Application |
| 10 | Results and conclusions | Keywords | Lists, Facilitates | Listens, Participates, Discusses | Knowledge, Communication, intrapersonal, Comprehension |

**Title: Simulation of Distance Vector/ Link State Routing algorithm.**

**Objectives:** To simulate Distance vector Routing protocol

**Aim/ Problem Statement**: Write a program to implement Distance vector routing protocol

## Theory:

## A distance-vector routing (DVR) protocol requires that a router inform its neighbors of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as Bellman-Ford algorithm).

**Bellman Ford Basics –** Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances,based on a chosen metric, are computed using information from the neighbors’ distance vectors.

Information kept by DV router -

* Each router has an ID
* Associated with each link connected to a router,
* There is a link cost (static or dynamic).
* Intermediate hops

Distance Vector Table Initialization -

* Distance to itself = 0
* Distance to ALL other routers = infinity number.

**Distance Vector Algorithm –**

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
   * It receives a distance vector from a neighbor containing different information than before.
   * It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination

Dx(y) = Estimate of least cost from x to y

C(x,v) = Node x knows cost to each neighbor v

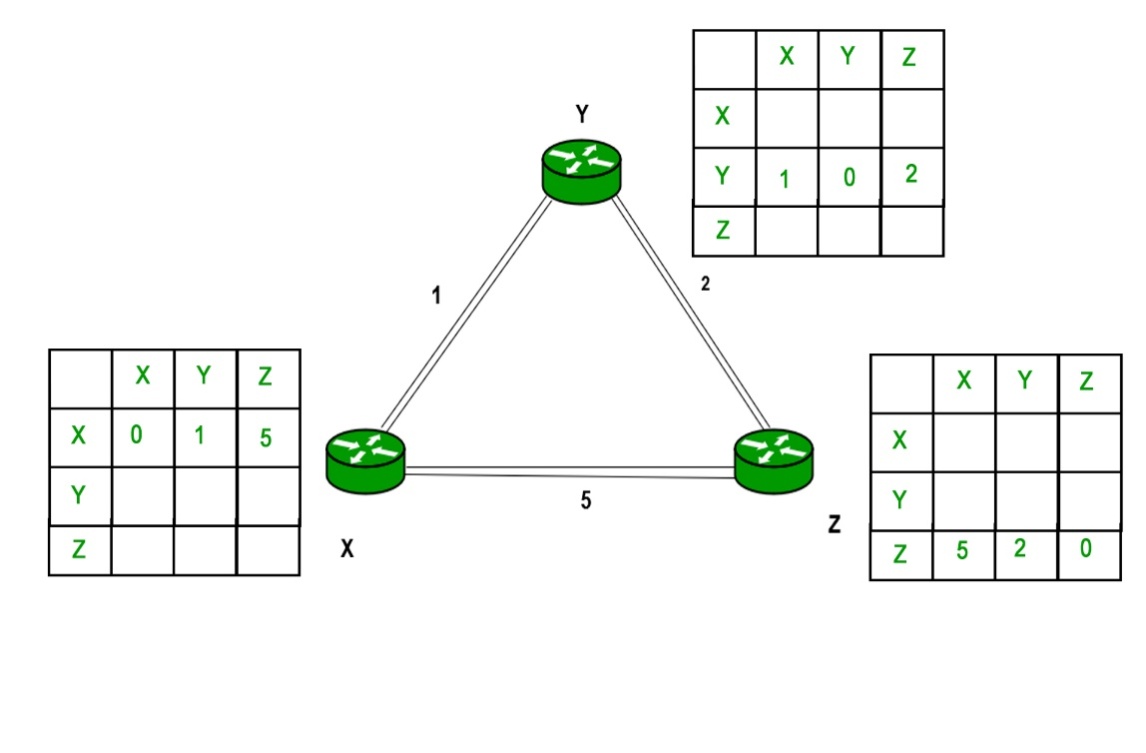
Dx= [Dx(y): y ∈ N ] = Node x maintains distance vector

Node x also maintains its neighbors' distance vectors

– For each neighbor v, x maintains Dv = [Dv(y): y ∈N ]

**Note –**

* From time-to-time, each node sends its own distance vector estimate to neighbors.
* When a node x receives new DV estimate from any neighbor v, it saves v’s distance vector and it updates its own DV using B-F equation:
* Dx(y) = min { C(x,v) + Dv(y), Dx(y) } for each node y ∈ N



**Example –** Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.

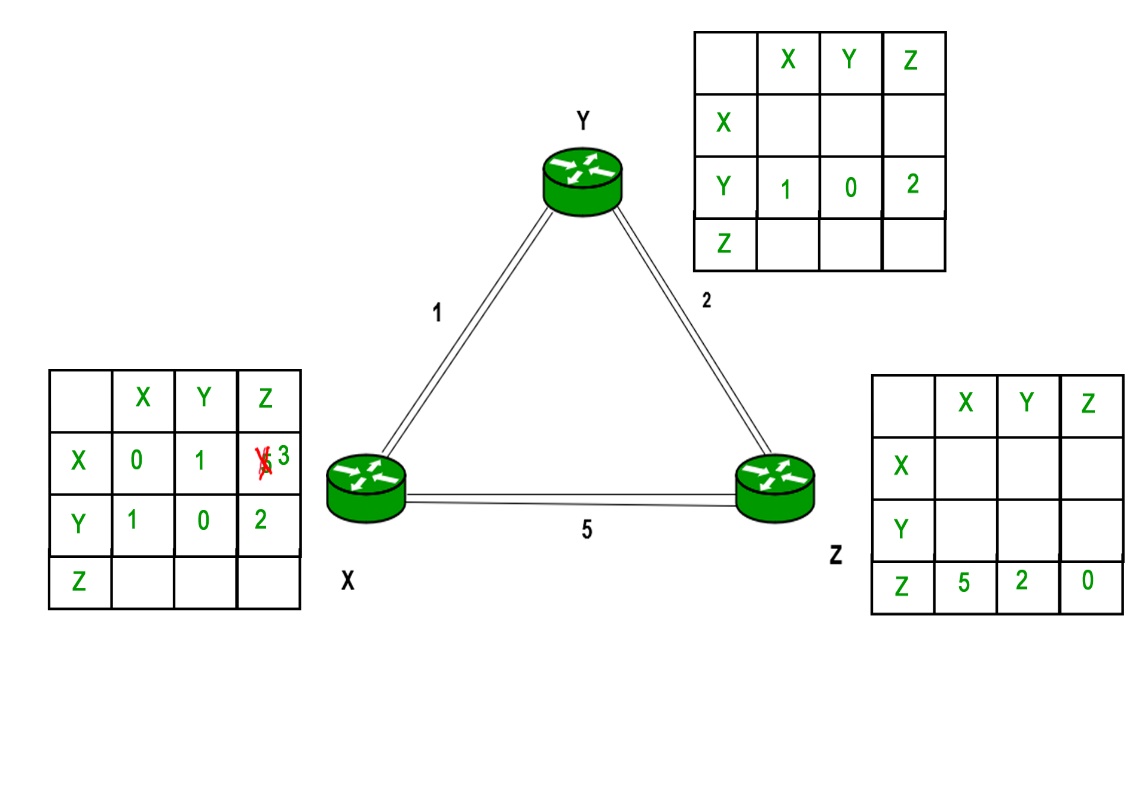
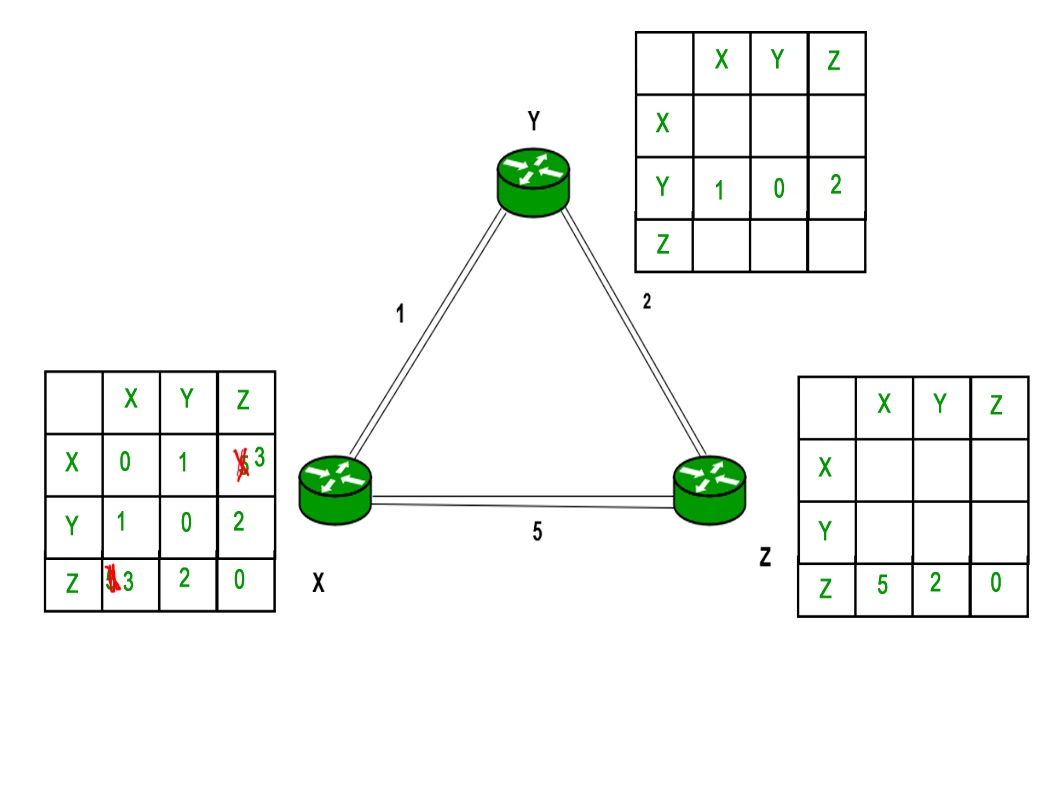
Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

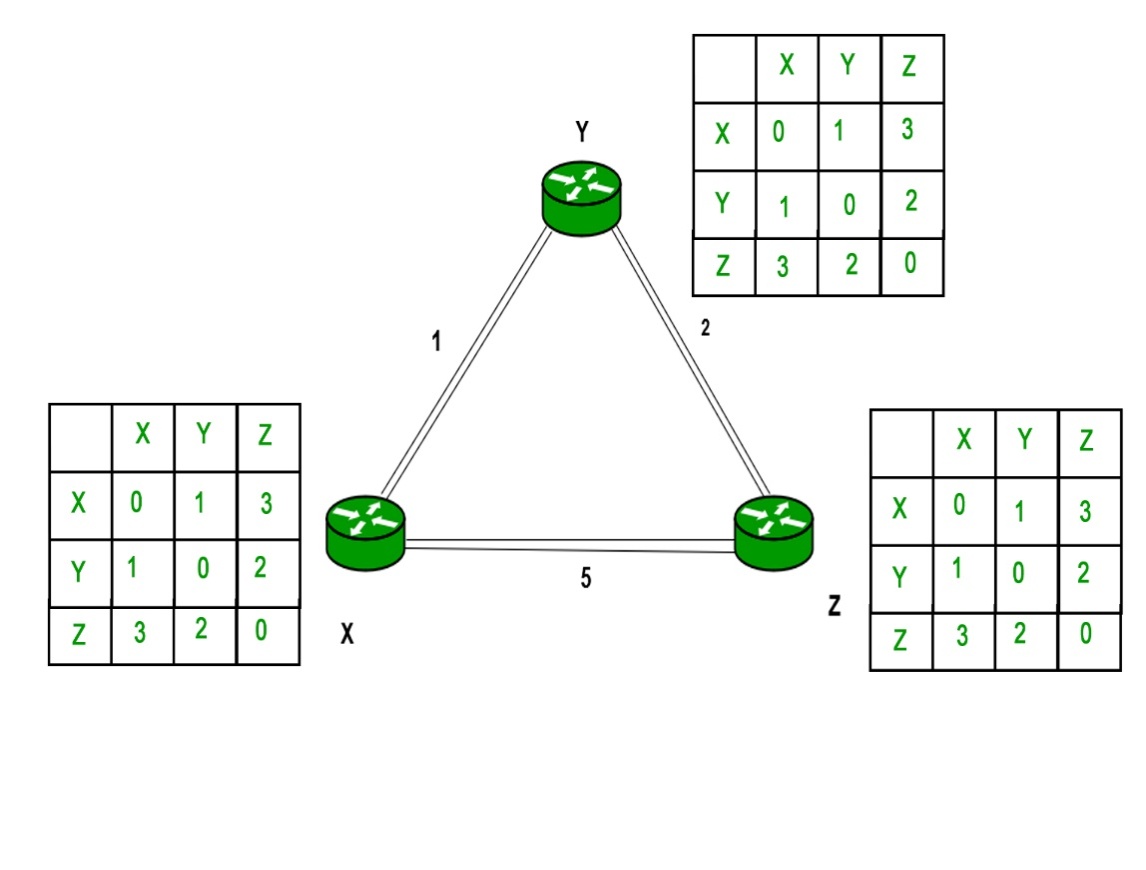
Dx(y) = min { C(x,v) + Dv(y)} for each node y ∈ N

As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.  
  
  
Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

Dx(y) = min { C(x,v) + Dv(y)} for each node y ∈ N

As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.

  
Similarly for Z also –  


Finally the routing table for all –  
  
  
**Advantages of Distance Vector routing –**

* It is simpler to configure and maintain than link state routing.

**Disadvantages of Distance Vector routing –**

* + It is slower to converge than link state.
  + It is at risk from the count-to-infinity problem.
  + It creates more traffic than link state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
  + For larger networks, distance vector routing results in larger routing tables than link state since each router must know about all other routers. This can also lead to congestion on WAN links.

**Note –** Distance Vector routing uses UDP(User datagram protocol) for transportation.

#include<stdio.h>

#define MAX 100

#define TEMP 0

#define PERM 1

#define infinity 9999

#define NIL -1

voidfindPath(int source, int destination);

voiddvrp(int source);

intminTEMP();

voidcreateGraph();

voiddisplayGraph();

intnumNodes;

intadjMatrix[MAX][MAX];

int predecessor[MAX];

intpathLength[MAX];

int status[MAX];

int main(){

int source, destination;

createGraph();

printf("Enter the Source Vertex : ");

scanf("%d", &source);

dvrp(source);

for(inti=0; i<numNodes; i++){

if(i != source){

findPath(source, i);

}

}

return 0;

}

voiddvrp(int source){

int current;

for(inti=0; i<numNodes; i++){

predecessor[i] = NIL;

pathLength[i] = infinity;

status[i] = TEMP;

}

pathLength[source] = 0;

while(1){

current = minTEMP();

if(current == NIL)

return;

status[current] = PERM;

for(inti=0; i<numNodes; i++){

if(adjMatrix[current][i] != 0 && status[i] == TEMP){

if(pathLength[current] + adjMatrix[current][i] <pathLength[i]){

predecessor[i] = current;

pathLength[i] = pathLength[current] + adjMatrix[current][i];

}

}

}

// displayGraph();

}

}

intminTEMP(){

int min = infinity;

int k = NIL;

for(inti=0; i<numNodes; i++){

if(status[i] == TEMP &&pathLength[i] < min){

min = pathLength[i];

k = i;

}

}

return k;

}

voidfindPath(int source, int destination){

intdest = destination;

int u;

int path[MAX];

intshortDist = 0;

int count = 0;

while(destination != source){

count++;

path[count] = destination;

u = predecessor[destination];

shortDist += adjMatrix[u][destination];

destination = u;

}

count++;

path[count] = source;

printf("\nShortest Path from %d to %d is : ", source, dest);

for(inti = count; i>=1; i--){

printf("%d ", path[i]);

}

printf("\nShortest Distance is : %d \n", shortDist);

}

voidcreateGraph(){

intnumEdges, edgeWeight;

printf("Enter Number of Vertices : ");

scanf("%d", &numNodes);

numEdges = numNodes \* (numNodes - 1);

int origin, destination;

for(inti = 0; i<= numEdges; i++){

printf("\nEnter origin and destination (-1 -1 to quit) : ");

scanf("%d %d", &origin, &destination);

if(origin == -1 && destination == -1)

break;

printf("Enter weight of edge : ");

scanf("%d", &edgeWeight);

if(origin < 0 || destination < 0 || origin >= numEdges || destination >= numEdges){

printf("\nInvalidOrign or Destination...");

i--;

}

else

adjMatrix[origin][destination] = edgeWeight;

}

displayGraph();

}

voiddisplayGraph(){

printf("\nGraph \n");

for(inti = 0; i<numNodes; i++){

for(int j = 0; j <numNodes; j++)

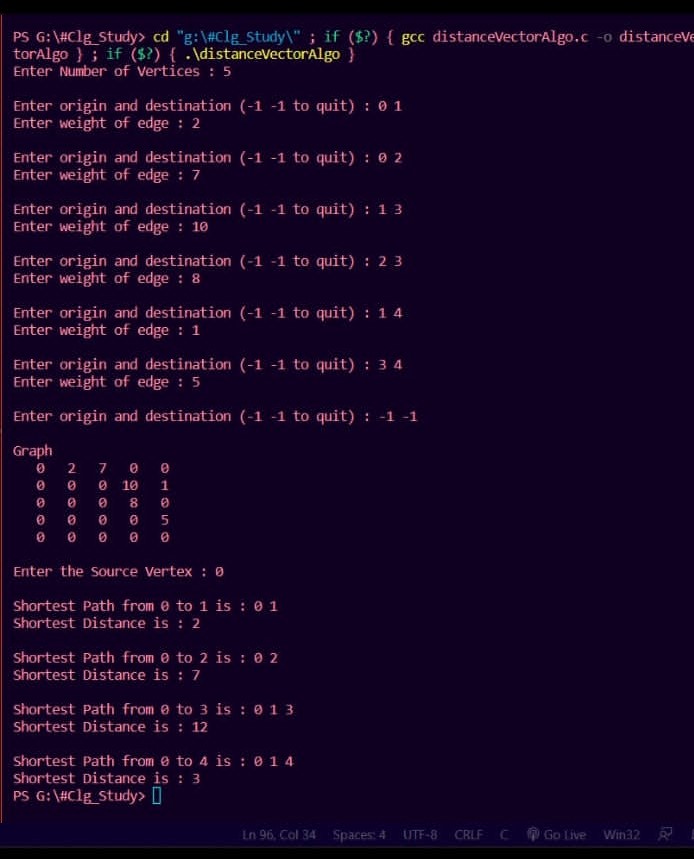
printf("%4d", adjMatrix[i][j]);

printf("\n");

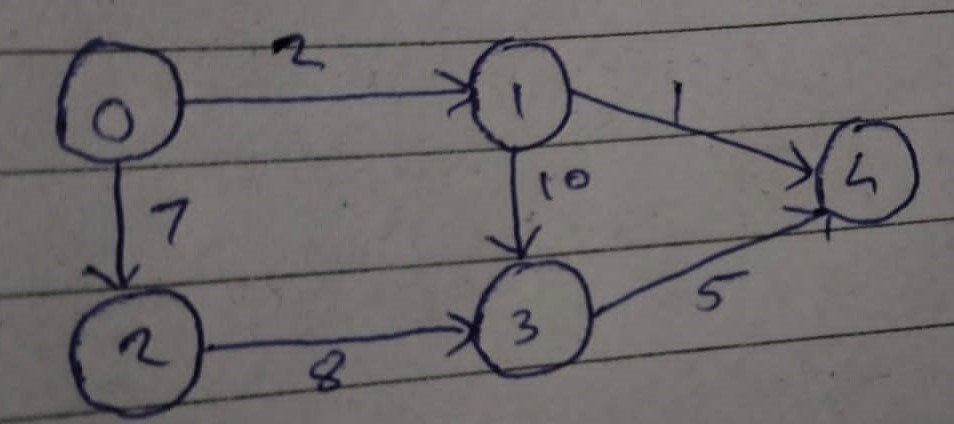
}

printf("\n");

}

**Input:**

**Output:**

****

**Conclusion:** Thus we have successfully implemented simulation of Distance Vector Routing Algorithm

## Upon completion of Experiment students will be able to :

1. Understand and implement simulation of Distance Vector routing algorithm.

# EXPERIMENT NO. 9

**Problem Definition:**

Setup a wired LAN using Layer 2 Switch and then IP switch ofminimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool.

**1. Components:**

RJ-45 connector, Crimping Tool, Twisted pair Cable(Cat6), Line Tester, HTTP Server (Apache) with Website pages of your Institute, Four Client Nodes with Wi-Fi Support, Wireshark Protocol Analyzer tool on all nodes, Layer-II Switch, Layer-III IP Switch, Wi-Fi Access Point.

**2. Prerequisite:**

1. Networking Components: Switch, Router, etc.
2. Linux Command: Ping
3. Wireshark Tool
4. IP Addressing
5. **Learning Objectives:**
   * Students will able to setup wired and Wi-Fi network
   * Learn to setup wired and Wi-Fi office/organization network
6. **Theory**

**Cable Preparation**

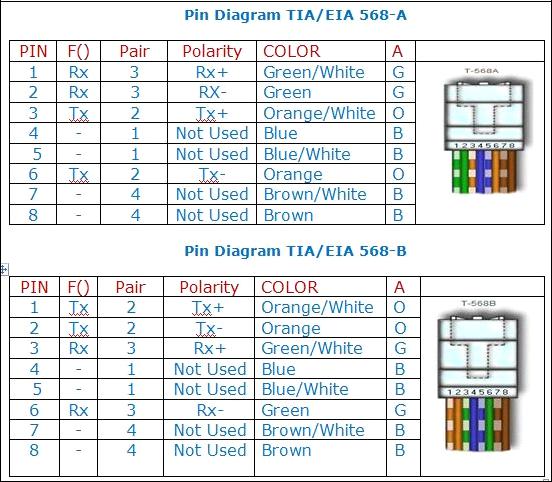
The cable will be constructed using either TIA/EIA T568A or T568B standards for Ethernet, which determines the color wire to be used on each pin.

**Straight-through patch cables** are normally used to connect a host directly to a hub orswitch or to a wall plate in an office area. With a straight-through cable, the color of wire used by pin 1 on one end is the same color used by pin 1 on the other cable end, and similarly for the remaining seven pins.

With a **crossover cable** the second and third pairs on the RJ-45 connector at one end of the cable are reversed at the other end. The pin-outs for the cable are the T568A standard on one

end and the T568B standard on the other end. Crossover cables are normally used to connect hubs and switches or can be used to directly connect two hosts to create a simple network.

**TIA/EIA 568A and 568B Wiring Standards**



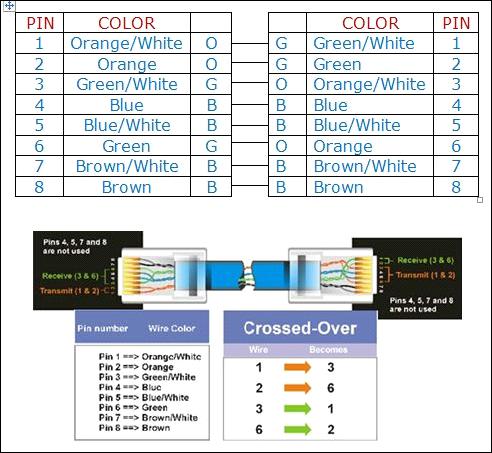
**Prepare and test an Ethernet straight-through and Crossover patch cable**

**Step 1: Obtain and prepare the cable**

* Determine the length of cable required. This could be the distance from a computer to a switch or between a device and an RJ-45 outlet jack.
* Using wire strippers, remove 5.08 cm (2 in.) of the cable jacket from both ends of the cable.

**Pin Diagram TIA/EIA 568-B for Straight-Through Cabling**

**Pin Diagram EIA/TIA 568B for Cross-Over Cabling**



**Step 2: Prepare and insert the wires**

* Determine which wiring standard will be used. Circle the standard. [T568A | T568B] and locate the correct table or figure from the “Wire Diagrams” based on the wiring standard used.
* Spread the cable pairs and arrange them roughly in the desired order based on the standard chosen.
* Untwist a short length of the pairs and arrange them in the exact order needed by the standard moving left to right starting with pin 1.
* It is very important to untwist as little as possible. The twists are important because they provide noise cancellation
* Straighten and flatten the wires between your thumb and forefinger. Ensure the cable wires are still in the correct order as the standard.
* Cut the cable in a straight line to within 1.25 to 1.9 cm (1/2 to 3/4 in.) from the edge of the cable jacket. If it is longer than this, the cable will be susceptible to crosstalk (the interference of bits from one wire with an adjacent wire).
* The key (the prong that sticks out from the RJ-45 connector) should be on the underside pointing downward when inserting the wires. Ensure the wires are in order from left to right starting with pin 1. Insert the wires firmly into the RJ-45 connector until all wires are pushed as far as possible into the connector

**Step 3: Inspect, crimp, and re-inspect**

* Visually inspect the cable and ensure the right color codes are connected to the correct pin numbers.
* Visually inspect the end of the connector. The eight wires should be pressed firmly against the end ofthe RJ-45 connector. Some of the cable jacket should be inside the first portion of the connector. This provides strain relief for the cable. If the cable jacket is not far

inside the connector, it mayeventually cause the cable to fail.

* If everything is correctly aligned and inserted properly, place the RJ-45 connector and cable into the

crimper. The crimper will push two plungers down on the RJ-45 connector.

* Visually re-inspect the connector. If improperly installed, cut the end off and repeat the process.

**Step 4: Terminate the other cable end**

* Use the previously described steps to attach an RJ-45 connector to the other end of the cable.
* Visually re-inspect the connector. If improperly installed, cut the end off and repeat the process.

**Step 5: Test the cable**

* Use the cable to connect a PC to a network.
* Visually check the LED status lights on the NIC card. If they are on (usually green or amber) the cable

is functional.

* On the PC, open the command prompt.
* Type ifconfig
* Write down the default gateway IP address.
* Or you can use line tester to test the prepared cable.

**Network Devices**

**1. Repeater:**

Functioning at Physical Layer. A repeater is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances. Repeater have two ports ,so cannot be use to connect for more than two devices.

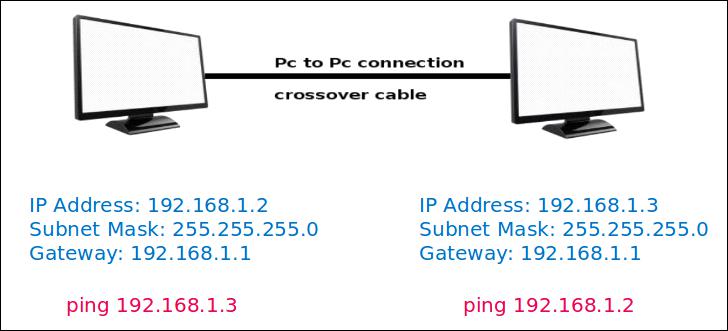
**2. Hub:**

An Ethernet hub, active hub, network hub, repeater hub, hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

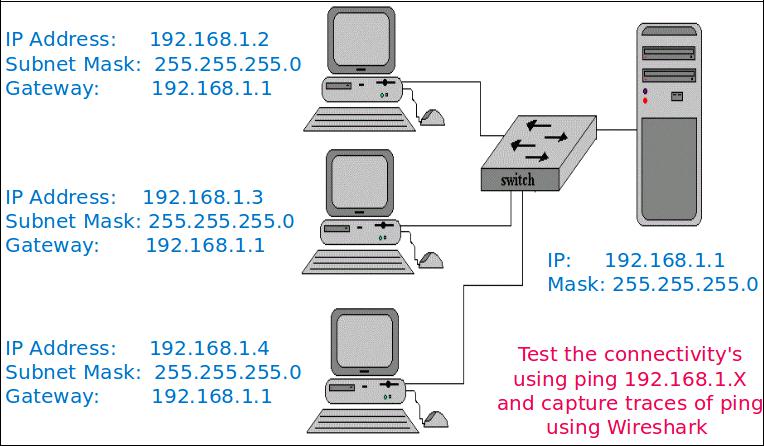
**3. Switch:**

A network switch or switching hub is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches.

**4. Bridge:**



**Setting Up LAN using Straight-Through Cable**



**Figure 1**

Connect four machines using Straight-Through cable to switch and router and then configure all using ip address, subnet mask and gateway address as shown in figure. Ping all the machines and capture ICMP packets in Wireshark tool.

1. **Testing Web Server over LAN**
   * Installation of Web Server – Apache2 or Tomcat7
   * Install the server – sudo apt-get install apache
   * Start web server - /etc/init.d/apache2 start
   * Create the web page and store in /var/www/http
   * Access the web pages from client machines 1/2/3

Test the web server by accessing web pages stored on server and capture the traces of http ,tcp, ip and Ethernet-II using Wireshark.

**Conclusion:**

Hence we have designed wired and wireless LAN using crossover and straight-through cable, and captured the ICMP, HTTP packets in Wireshark.

**EXPERIMENT: 10**

**Problem Definition:**

Configure RIP/OSPF/BGP using packet Tracer.

**1. Prerequisite:**

1. Protocols: RIP, OSPF, BGP
2. Packet Tracer
3. **Learning Objectives:**
   * Students will able to configure protocols like RIP, OSPF, BGP using Packet Tracer.
4. **Theory**

**Routing Protocols**

Routing protocols maintains routing tables where routing table contains a route to every destination network .

**Dynamic Routing Protocols**

There are three types of it as follows:

1. Routing Information Protocol (RIP)
2. Open Shortest Path First (OSPF)
3. Border Gateway Protocol (BGP)

RIP and OSPF are Interior Gateway Protocols (IGPs); they are designed to operate in a single autonomous system (AS). (An AS is a group of networks administered by the same authority). BGP is an Exterior Gateway Protocol (EGP), which allows routers in different autonomous systems to exchange routes. Because BGP routers must regulate traffic between networks controlled by organizations with different policies.

**How Routing Protocols Work**

A router constructs its routing table using the information it receives from other routers. The router changes its routing table in response to routing updates that provide additional information or notification that conditions in the network have changed (for example, a link has failed). This responsiveness explains why using a routing protocol is often called dynamic routing.

The protocol must dictate parameters such as the following:

* **How routers compute a route’s metric and select the best route for their routing table:**

Routing protocols can have a relatively complicated system for calculating a route’s metric.

So that you can select the best routing protocol (or protocols) for your network environment. If necessary, you can change which routes are chosen by altering the default metrics that a protocol assigns certain routes.

* **What information routers include in routing updates:** With some routing protocols,routers exchange their entire routing tables. With other routing protocols, routers exchange only portions of the routing table.
* **Which routers and router interfaces send and receive updates:** Most protocols specifythat when routers receive an update on an interface, they do not send the same update from that interface. This common sense rule minimizes overhead.
* **When routers send and receive updates and hellos:** To lower overhead and conservebandwidth, you can alter how often routers send certain messages.
  1. **Routing Information Protocol (RIP)**

RIP is one of the oldest dynamic routing protocols on the Internet that is still in use.RIP is an intradomain routing protocol that uses a distance vector approach to determine the paths between routers. RIP minimizes the number of hops on each path, where each point-to-point link or LAN constitutes a hop. Each RIP-enabled router periodically sends the content of its routing table to all its neighboring routers in an update message. For each routing table entry, the router sends the destination (host IP address or network IP address and associated prefix) and the distance to the destination measured in hops. When a router receives an update message from a neighboring router, it updates its own routing table

**Configuring RIP on CISCO ROUTER**

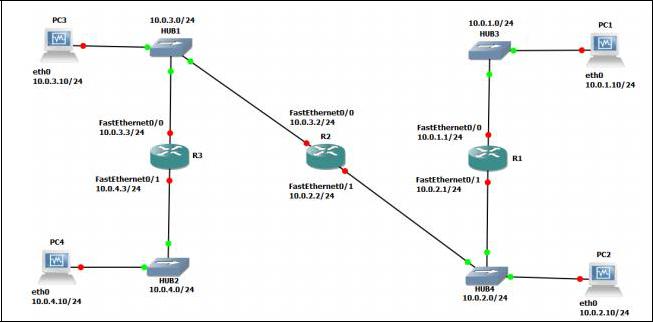


Fig. 1 Network topology

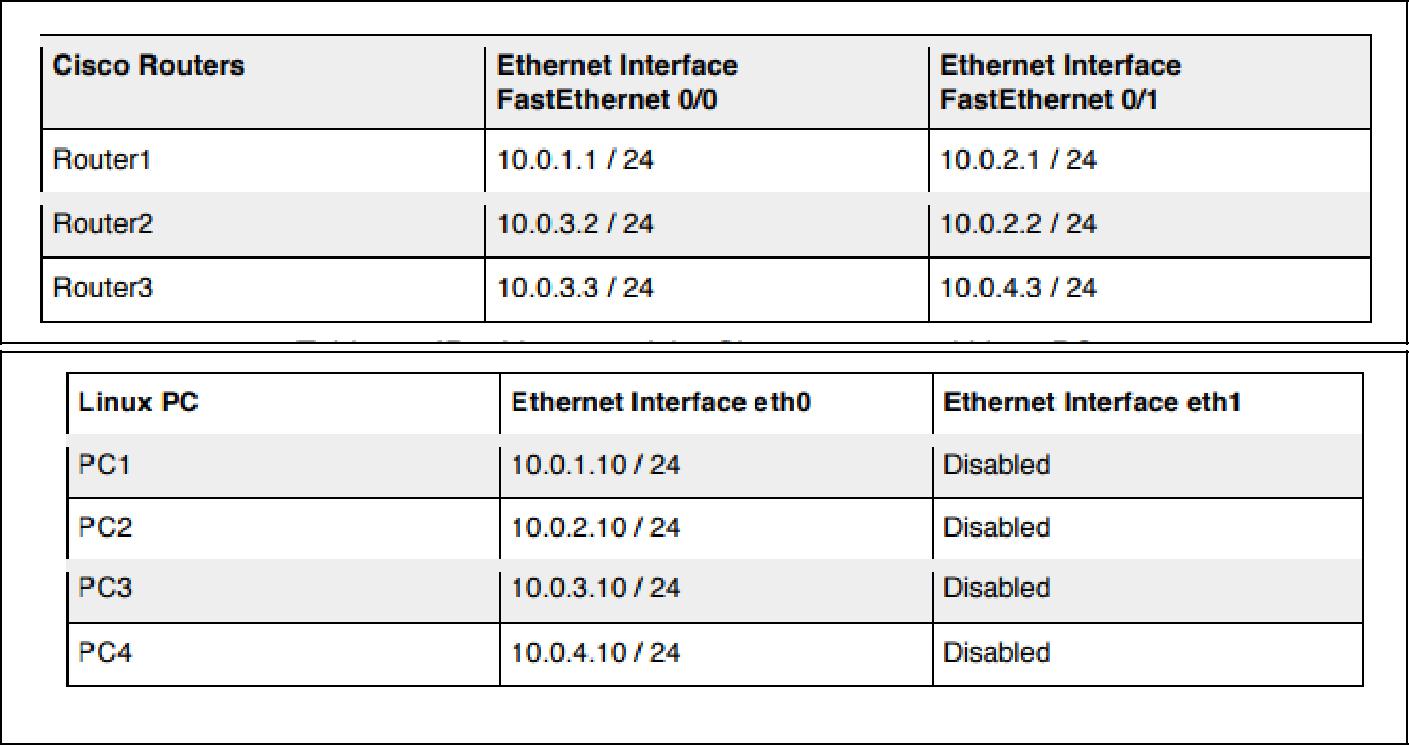
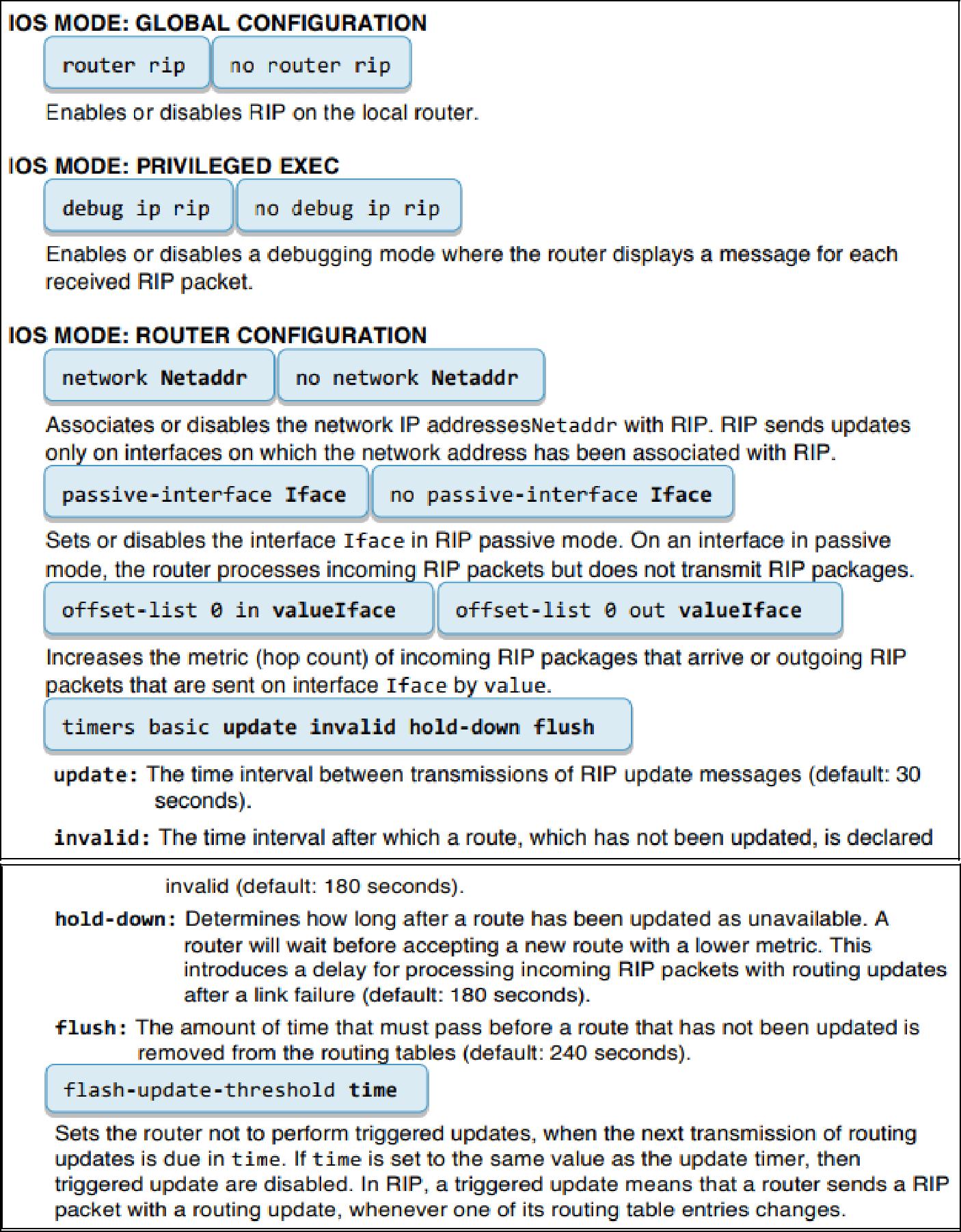


Table 1: IP addresses of the Cisco routers and Linux PCs

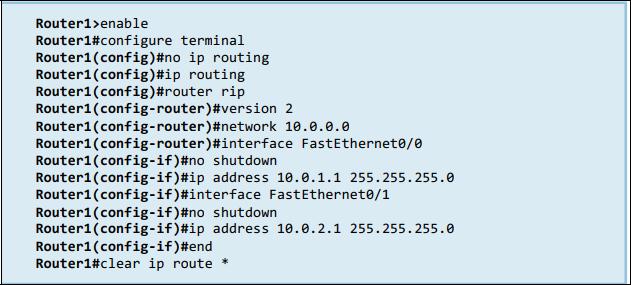
Above Figures describe the network configuration

**Exercise 1. Configuring RIP on Cisco routers**

In this exercise, you will configure all the routers to run RIP. After the configuration, all the routers should be able to ping all the other routers. Following is a brief overview of the basic commands used to configure RIP on a Cisco router. Make sure you type in the command in the correct command mode (note the prompt).



1. Connect the PCs and the Cisco Routers as shown in Figure1. The PCs and routers are connected with Ethernet hubs.
2. Start Routers by clicking the right button and select Start; then, open a terminal by clicking the right button and select Console.
3. On Router1, Router2, and Router3, configure the IP addresses as shown in Table 1, and enable the routing protocol RIP. The commands to set up Router 1 are as follows:



1. After you have configured the routers, check the routing table at each router with the show ip route command. Each router should have four entries in the routing table: two entries for directly connected networks and two other entries for remote networks that were added by RIP.
2. From each router, issue a ping command to the IP address of interfaces FastEthernet0/0 and FastEthernet0/1 on all remote routers.

**2.** **Open Shortest Path First (OSPF)**

OSPF is a link state routing protocol, in which each router sends information on the cost metric of its network interfaces to all other routers in the network, The information about the interfaces is sent in messages that are called link state advertisements (LSAs). LSAs are disseminated using flooding; that is, a router sends its LSAs to all its neighbors, which, in turn, forward the LSAs to their neighbors and so on. However, each LSA is forwarded only once. Each router maintains a link state database of all received LSAs, which provides the router with complete information about the topology of the network, Routers use their link state databases to run a shortest-path algorithm that computes the shortest paths in the network.

The network configuration is shown in Figure 2 and Table 2. Note that PC1-4 are set up as routers.

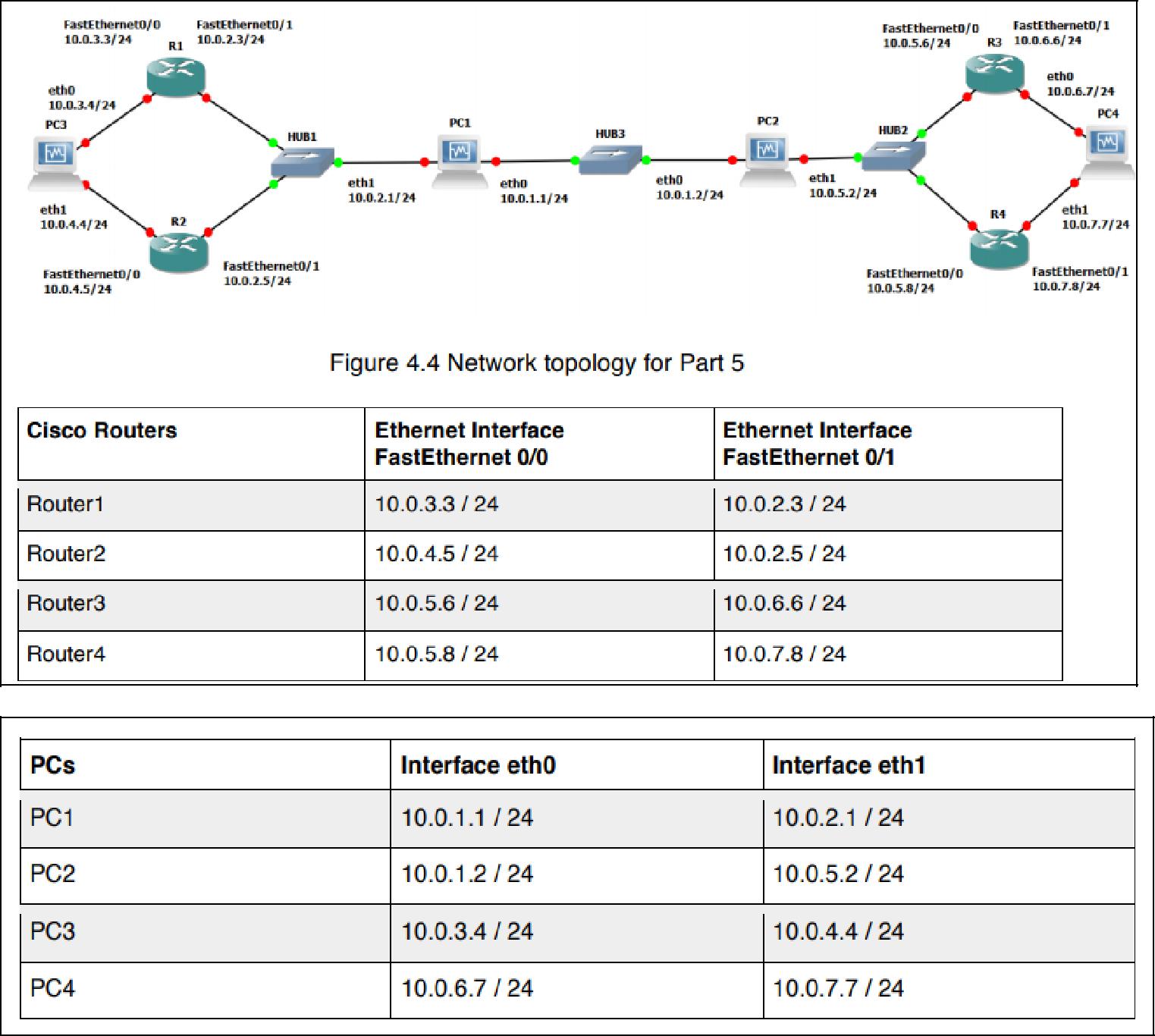
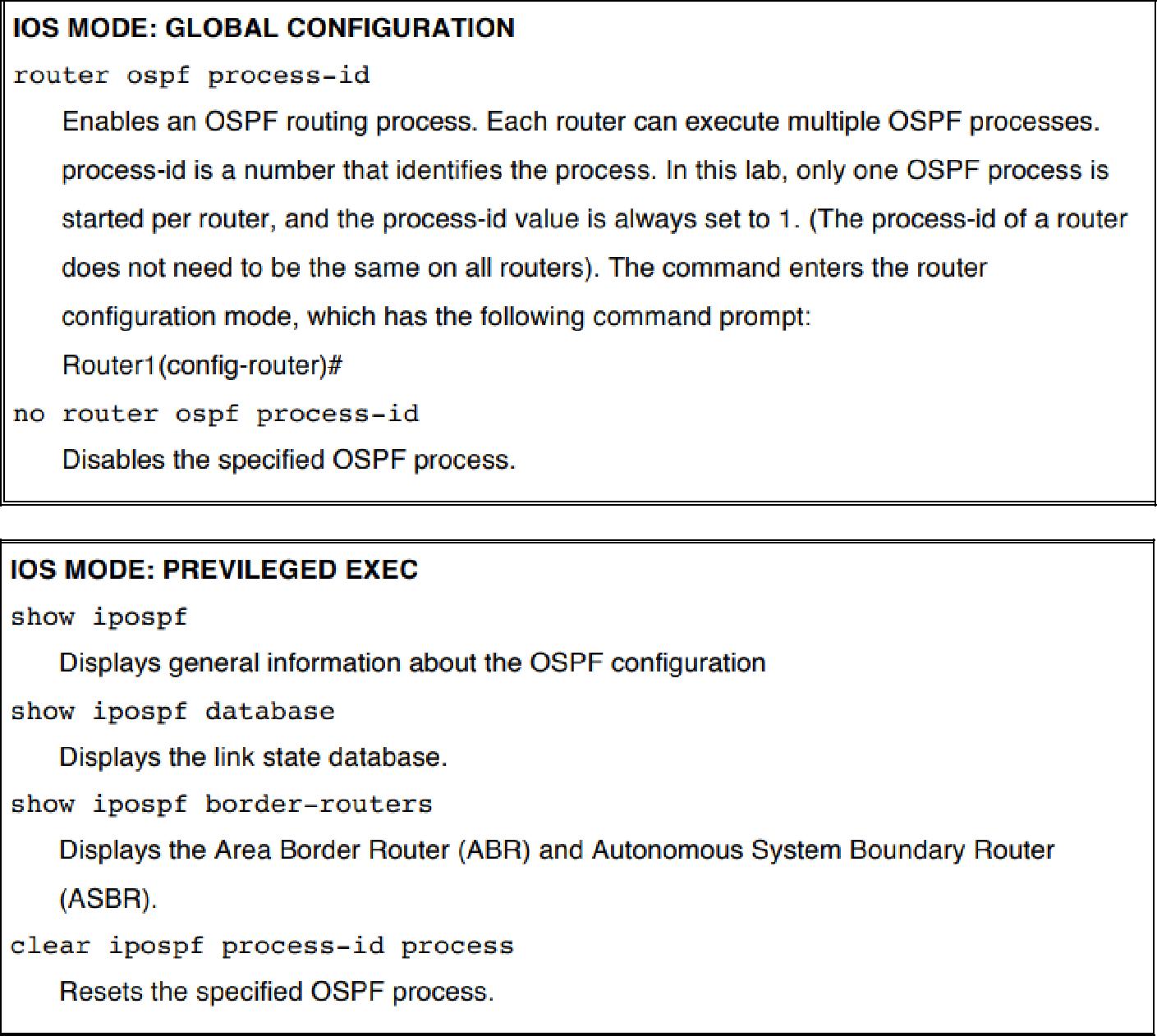
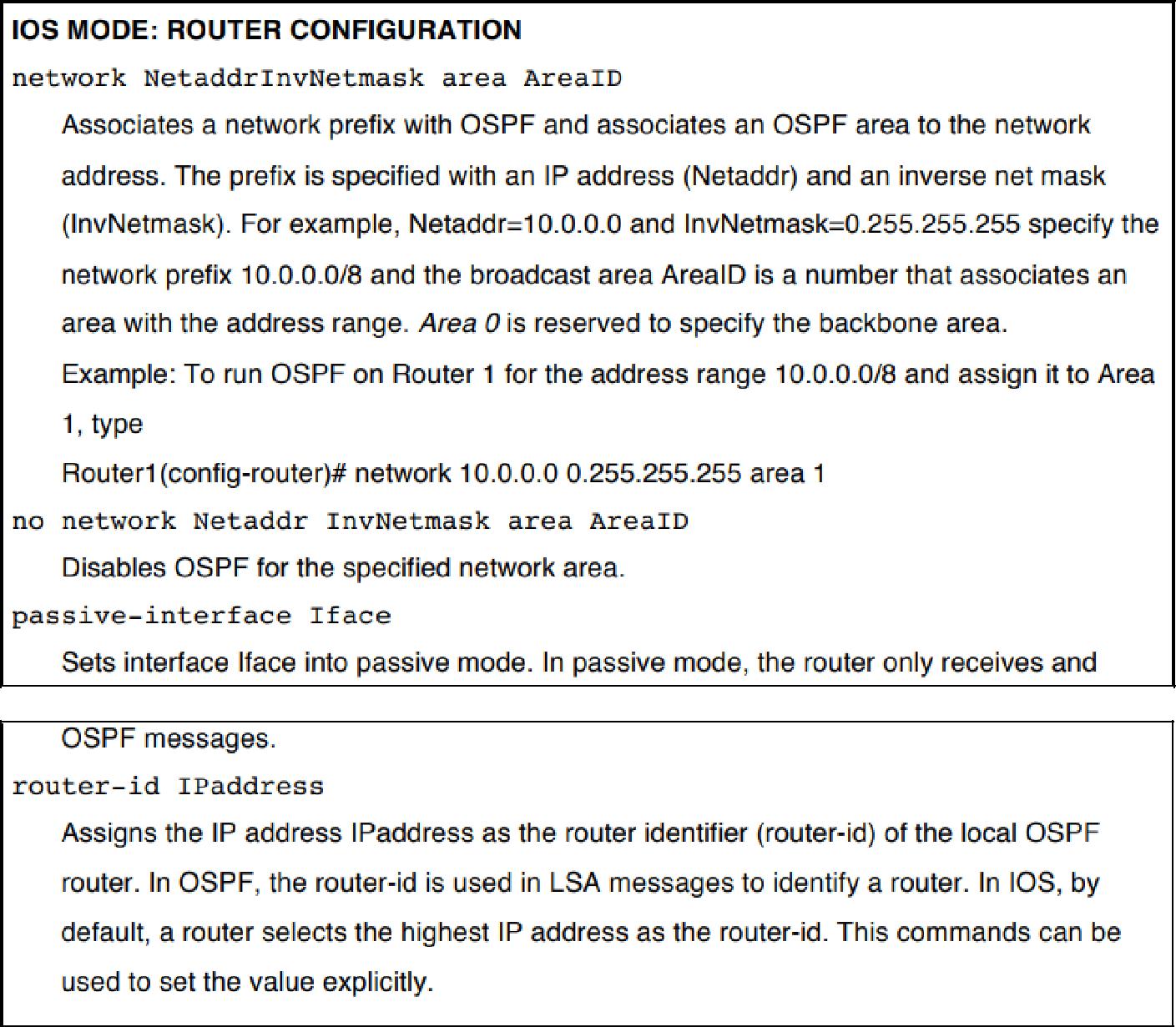


Table 2: . IP addresses of the routers and PCs

**Configuring OSPF on Cisco routers**

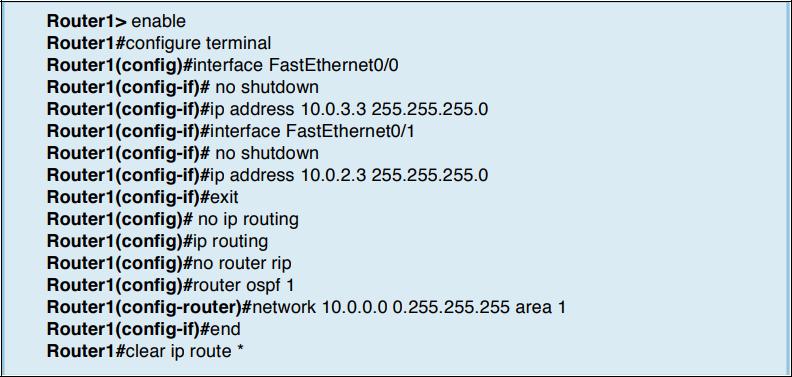
In this exercise, you configure OSPF on the Cisco routers. A brief description of the basic IOS commands used to configure OSPF on a Cisco router follows. As usual, each command must be issued in a particular IOS command mode.





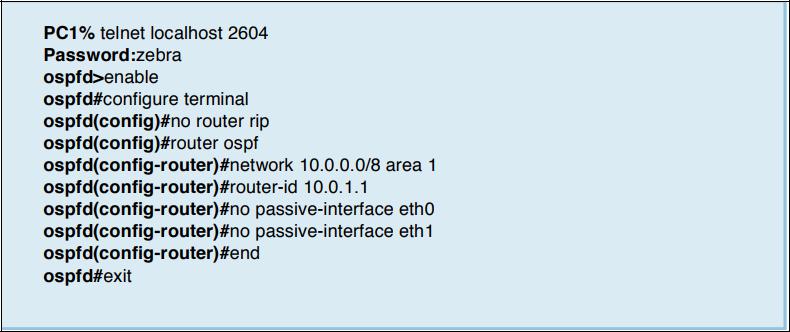
1. Connect the routers as shown in Figure 2.
2. Configure the Cisco routers to run OSPF.

The following commands are used to configure Router1:



These commands configure the IP addresses of the routers, disable RIP, and enable OSPF for Area 1 and network 10.0.0.0/8. Since no router-id is specified, the highest IP address of Router1, 10.0.3.3, is used as the router-id. The router-id can be verified by issuing the command show ip OSPF.

1. Set up the PCs as OSPF routers. Refer to Figure 2 for the connections and to Table 2 for the IP addresses. Use the following set of commands.
   1. Now configure the PC's similar to the way you configured the routers



5. Enable ip\_forwarding on all the PCs.

**Observing convergence of OSPF**

In comparison to the distance vector protocol RIP, the link state routing protocol OSPF

quickly adapts to changes in the network topology. In this exercise, you observe the interactions of OSPF after a change to the network topology.

1. On PC1, start to capture traffic with Wireshark on interface FastEthernet0/0. Set a filter to display only OSPF packets.
2. From PC3, run a trace command to PC4. Confirm from the output and Figure 2 whether the path from PC3 to PC4 includes Router3 or Router4.
3. Issue a ping command from PC3 to PC4 (10.0.7.7). Do not terminate the ping command until this exercise is completed.
4. If the path from PC3 to IP address 10.0.7.7 from Step 2 included Router3, then disconnect the Ethernet cable of FastEthernet0/1 interface of Router3. Otherwise, disconnect the Ethernet cable of FastEthernet0/1 interface of Router4. When the Ethernet cable is disconnected, the ping command on PC3 will show that IP address 10.0.7.7 is not reachable.
5. Now OSPF updates the routing tables. Use the Wireshark window on PC1 to observe the transmitted OSPF messages:

* How quickly are OSPF messages sent after the cable is disconnected?
* How many OSPF messages are sent?
* Which type of OSPF packet is used for flooding link state information?
* Describe the flooding of LSAs to all routers.
* Which type of encapsulation is used for OSPF packets (TCP, UDP, or other)?
* What is the destination address of OSPF packets?

1. Wait until the ping command is successful again, that is, ICMP Echo Reply messages arrive at PC3. This happens when the routing tables have been updated.
2. Stop the ping command and save the ping statics output.

* Count the number of lost packets and calculate the time it took OSPF to update the routing tables. (The ping command issues an ICMP Echo Request message approximately once every second.)

1. Issue another trace command from PC3 to IP address 10.0.7.7 By now, the output should show the new route to PC4.
2. Save the link state database on all Cisco routers to a file, and verify that all routers indeed have the same link state database.

• Compare the output of the command “show ip OSPF database” from the Cisco routers

10. Stop Wireshark on PC1, and save the different types of OSPF packets captured by

Wireshark. Save one copy of each type of OSPF packet that you observed. a) Pick a single link state advertisement packet captured by Wireshark, and describe how to interpret the information contained in the link state advertisement.

**3. Border Gateway Protocol (BGP)**

This provides some exposure to the inter domain Border Gateway Protocol (BGP), which determines paths between autonomous systems on the Internet.

BGP uses a path vector algorithm, where routers exchange full path information of a route. An important feature of BGP is that it can define routing policies, which can be used by a network to specify which type of traffic it is willing to process. The network configuration for this part is shown in Figure 3, and the IP configuration information is given in table 3. the network has three autonomous systems with AS numbers 100, 200 and 300. PC4, is used to capture the BGP packets transmitted between the ASs.

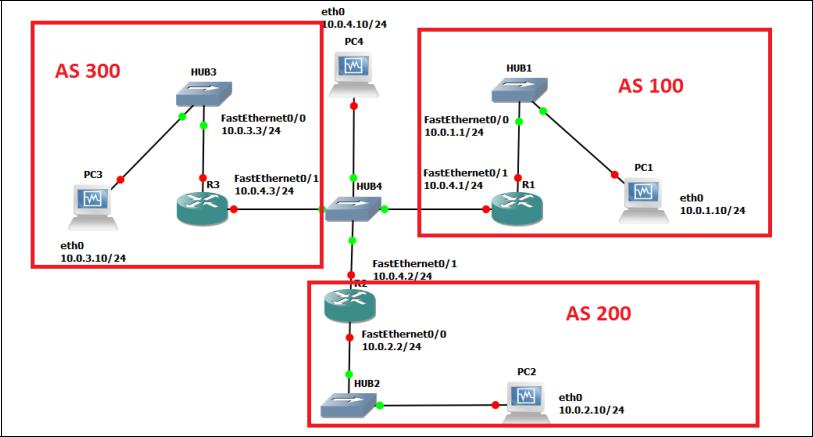
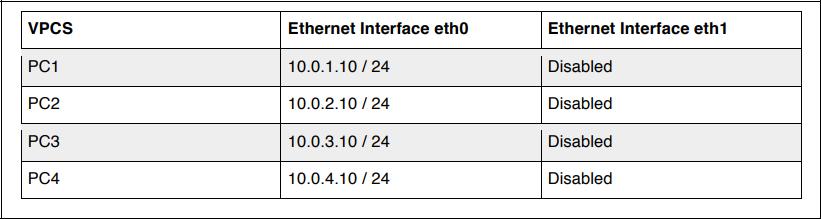


Figure 3 Network topology



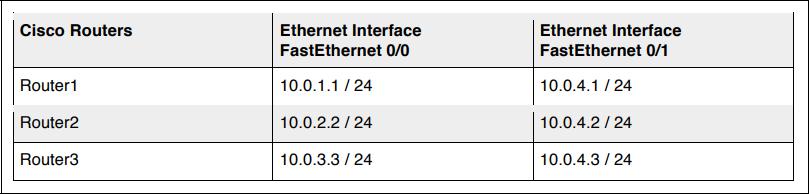


Table 3 IP addresses of the routers and PCs

**Basic BGP configuration**

Here, you configure the Cisco routers as BGP routers and you assign routers to autonomous systems. The configuration is completed when you can issue ping commands between any two PCs Next we summarize the Cisco IOS commands that are used to enable BGP

1. Disable all RIP or OSPF processes that are running on the Cisco routers. Use the following commands:

Router1# no router ospf 1

Router1# no router rip

1. Disable all RIP or OSPF processes running on the Linux PCs using the following command. For PC1, on the console at the prompt type:

PC1% /etc/init.d/quagga stop

1. Assign the IP addresses to Ethernet interface eth0 of each PC as indicated in Table 3
2. Disable eth1 on the Linux PCs using the following command as shown in Table 3. For PC1, on the console at the prompt type:

PC1% ifconfig eth1 down

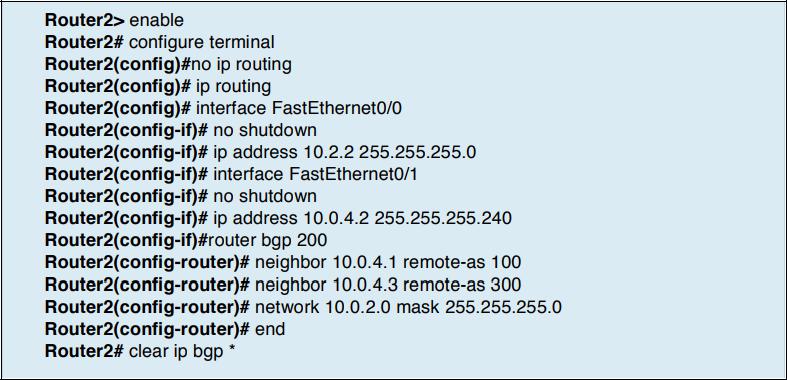
1. Add a default gateway to PC1, PC2, and PC3 as follows:

PC1% route add default gw 10.0.1.1/24

PC2% route add default gw 10.0.2.2/24

PC3% route add default gw 10.0.3.3/24

1. Start Wireshark on PC4 and set a display filter to capture only BGP packets.
2. Configure the Cisco routers to run BGP with the autonomous system numbers shown in Figure 3. The routers must know the AS number of their neighbors. Following is the configuration for Router2. Router 2 is in AS 200 and neighbors are AS 100 and AS 300.



1. On PC1, issue a ping command to PC3. The command succeeds when BGP has converged.
2. Once the routing tables have converged, you see all the other AS entries in the BGP routing table. On each Cisco router, save the output of the following commands:

Router1# show ip route

Router1# show ipbgp

Router1# show ipbgp paths

* Describe the different types of BGP messages that you observe in the Wireshark window on

PC4.

* Notice that BGP transmits messages over TCP connections. What is a reason that BGP uses

TCP to transmit its messages?

* What is the IP address of the next-hop attribute for AS 100 on Router 2?
* What are the BGP peers in this topology?

1. Stop the Wireshark traffic capture on PC4 and save the BGP packets captured by Wireshark.
   1. Use the output to provide answers to the questions in Step 7.
   2. Which BGP message(s) contain(s) the AS-PATH information? Use a BGP message

to illustrate your answer.

c) Use the saved output to provide a brief explanation of how the routers find the proper path between the autonomous systems.

**BGP convergence**

Disconnect one of the links between two BGP peers and observe how the BGP protocol reconfigures the paths.

1. After previous Exercise, save the output of the command show ip BGP neighbors on Router.2. Pay attention to the neighbor AS information.
2. On PC4, run Wireshark and set a display filter for BGP. Observe the flow of BGP packets between the autonomous systems.
3. On all routers, change the keep alive timer to 10 seconds and the hold time timer to 30 seconds. This speeds up the convergence time by a factor of 6 as compared to the default values. The following are the commands for Router2:

Router2# configure terminal

Router2(config)#router bgp 200

Router2(config-router)# timers bgp 10 30

Router2(config-router)#end

Router2#clear ipbgp \*

4. Disconnect the cable of interface FastEthernet0/1 on Router1.

* + From the output you saved, describe how the BGP routers learn that a link is down. (Hint:

Look at the BGP State field)

* Which BGP messages indicate that there is a link problem? Use a BGP message to answer the question.
  1. Use the command show ip BGP neighbors on Router2 and Router3 to obtain the neighbor information. Save the output.

1. Wait until BGP converges. Save the routing tables on Router2 and Router3. What can you say?
2. Stop the Wireshark traffic captured on PC4 and save the Wireshark BGP packets.
3. From the output you saved, describe how the BGP routers learn that a link is down. (Hint: Look at the BGP State field)
4. Which BGP messages indicate that there is a link problem? Use a BGP message to answer the question.

**Conclusion:**

Hence we have configured RIP, OSPF and BGP using packet tracer.