Departmet of Computer Engineering

SY B. Tech.

LAB MANUAL

Computer Networks

Laboratory

Prepared By

Mr. Atul Pawar (Asst. Prof.)

Mrs. Swati Jaiswal (Asst. Prof.)

Mrs. Shruti Chaudhari (Asst. Prof.)

Examination Scheme: TW: 25 Marks **PR**: 25 Marks

Assignment No.	Assignment Title
1.	Study the college / organization network, networking devices and its working in detail. Study the college/organization Server functioning and security parameters.
2.	Setup a wired LAN using Layer 2 Switch and then IP switch of minimum four computers. It includes preparation of cable, testing of cable using line tester, configuration machine using IP addresses, testing using PING utility and preparing server to send file to client. Demonstrate the PING packets captured traces using Wireshark Packet Analyzer Tool.
3.	Write a program for error detection and correction for 7/8 bits ASCII codes using CRC.
4.	Write a program using TCP sockets for wired network to implement Peer to Peer Chat .
5.	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocolin peer-to-peer mode.
6.	Configure RIP/OSPF/BGP using packet Tracer.
7.	Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.
8.	Write a program to demonstrate subnetting and find the subnet masks.
9.	Installing and configure DHCP server.
10.	Write a program to simulate the behaviour of link state routing protocol to find suitable path for transmission.

Assignment No. 1

1. Problem Definition:

Study the college / organization network, networking devices and its working in detail. Study the college /organization Server functioning and security parameters.

2. Prerequisite:

Networking Components: Switch, Router, etc. Linux Command: Ping Addressing

3. Learning Objectives:

Students will able to understand college network and its server functioning.

4. Theory:

IP address definition An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network. An IP address is a string of numbers separated by periods. IP addresses are expressed as a set of four numbers — an example address might be 192.158.1.38. Each number in the set can range from 0 to 255. So, the full IP addressing range goes from 0.0.0.0 to 255.255.255.255. IP addresses are not random. They are mathematically produced and allocated by the Internet Assigned Numbers Authority (IANA), a division of the Internet Corporation for Assigned Names and Numbers (ICANN). Types of IP addresses Private IP addresses Every device that connects to your internet network has a private IP address. This includes computers, smartphones, and tablets but also any Bluetooth-enabled devices like speakers, printers, or smart TVs. Router generates private IP addresses that are unique identifiers for each device that differentiate them on the network. Public IP addresses A public IP address is the primary address associated with whole network. Public IP address is provided to your router by your ISP. Typically, ISPs have a large pool of IP addresses that they distribute to their customers. Public IP address is the address that all the devices outside your internet network will use to recognize your network. Public IP addresses come in two forms – dynamic and static. Dynamic IP addresses Dynamic IP addresses change automatically and regularly. ISPs buy a large pool of IP addresses and assign them automatically to their customers. Periodically, they re-assign them and put the older IP addresses back into the pool to be used for other customers. Static IP addresses In contrast to dynamic IP addresses, static addresses remain consistent. Once the network assigns an IP address, it remains the same. A DHCP Server is a network server that automatically provides and

assigns IP addresses, default gateways and other network parameters to client devices. It relies on the standard protocol known as Dynamic Host Configuration Protocol or DHCP to respond to broadcast queries by clients. A DHCP server automatically sends the required network parameters for clients to properly communicate on the network. DHCP servers usually assign each client with a unique dynamic IP address, which changes when the client's lease for that IP address has expired. The DHCP service brings three key values:

1) Operation tasks are reduced: the network administrator no longer needs to manually configure each client before it can use the network

2) The IP addressing plan is optimized: addresses no longer being used are freed up and made available to new clients connecting

3) User mobility is easily managed: the administrator doesn't need to manually reconfigure a client when its network access point changes. Client-Server Network A client server architecture is a computing model wherein the server hosts, delivers, and manages most of the resources and services requested by the client. It is also known as the networking computing model or client server network as all requests and services are delivered over a network. The client server architecture or model has another(other) system(s) connected over a network where resources are shared among the different computers. Peer-to-Peer Network Peer-to-peer network involves two or more computers that pool various peripheral resources such as printers and DVD players. Such shared resources are available on every computer in the network. In a peer-to-peer network, each computer behaves as the client as well as the server, and it communicates with the other computers directly.

Students should write in details : Private & public IP adrresses, PING,ifconfig commands, NAT.

Conclusion: Hence we have studied college network , networking devices and its working.

Expected Print Attachment : College network diagram

Assignment No. 2

Problem Definition:

Setup a wired LAN using Layer 2 Switch and then IP switch of minimum 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. Apparatus (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:

Networking Components: Switch, Router, etc.

Linux Command: Ping

Wireshark Tool

IP Addressing

3. Learning Objectives:

- Students will able to setup wired and Wi-Fi network
- Learn to setup wired and Wi-Fi office/organization network

4. 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 or switch 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.

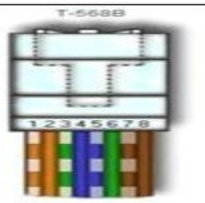
TIA/EIA 568A and 568B Wiring Standards

Pin Diagram TIA/EIA 568-A

PIN	F()	Pair	Polarity	COLOR	A
1	Rx	3	Rx+	Green/White	G
2	Rx	3	RX-	Green	G
3	Tx	2	Tx+	Orange/White	O
4	-	1	Not Used	Blue	B
5	-	1	Not Used	Blue/White	B
6	Tx	2	Tx-	Orange	O
7	-	4	Not Used	Brown/White	B
8	-	4	Not Used	Brown	B

Pin Diagram TIA/EIA 568-B

PIN	F()	Pair	Polarity	COLOR	A
1	Tx	2	Tx+	Orange/White	O
2	Tx	2	Tx-	Orange	O
3	Rx	3	Rx+	Green/White	G
4	-	1	Not Used	Blue	B
5	-	1	Not Used	Blue/White	B
6	Rx	3	Rx-	Green	G
7	-	4	Not Used	Brown/White	B
8	-	4	Not Used	Brown	B



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	COLOR	
1	Orange/White	O
2	Orange	O
3	Green/White	G
4	Blue	B
5	Blue/White	B
6	Green	G
7	Brown/White	B
8	Brown	B

	PIN	COLOR
O	1	Orange/White
O	2	Orange
G	3	Green/White
B	4	Blue
B	5	Blue/White
G	6	Green
B	7	Brown/White
B	8	Brown

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 of the 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 may eventually 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:

A network bridge connects multiple network segments at the data link layer (Layer 2) of the OSI model. In Ethernet networks, the term bridge formally means a device that behaves according to the IEEE 802.1D standard. A bridge and switch are very much alike; a switch being a bridge with numerous ports. Switch or Layer 2 switch is often used interchangeably with bridge. Bridges can

analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.

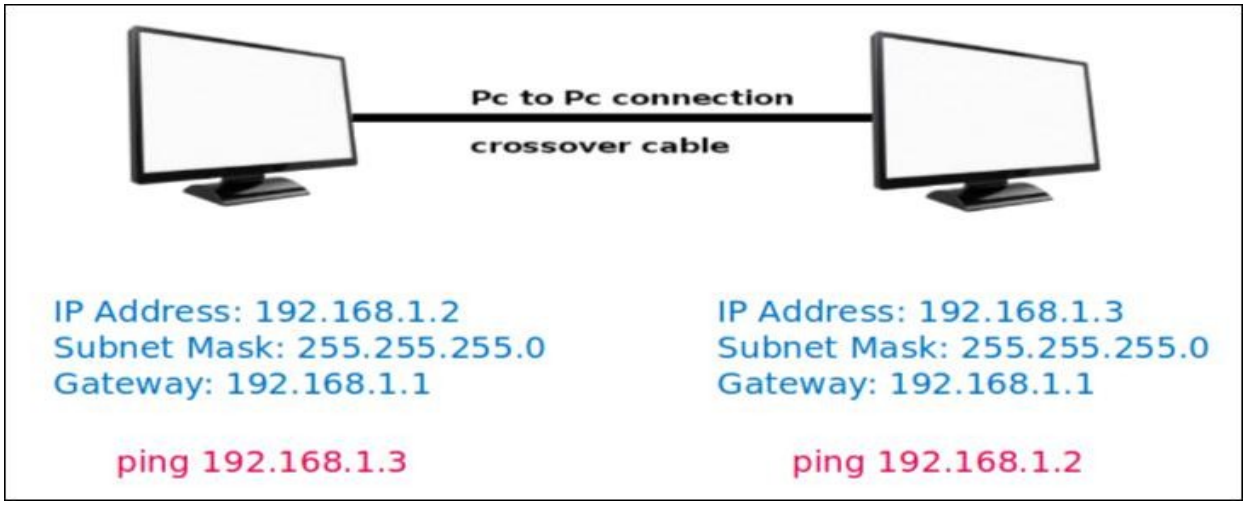
5. Router:

A router is an electronic device that interconnects two or more computer networks, and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.

6. Gate Way:

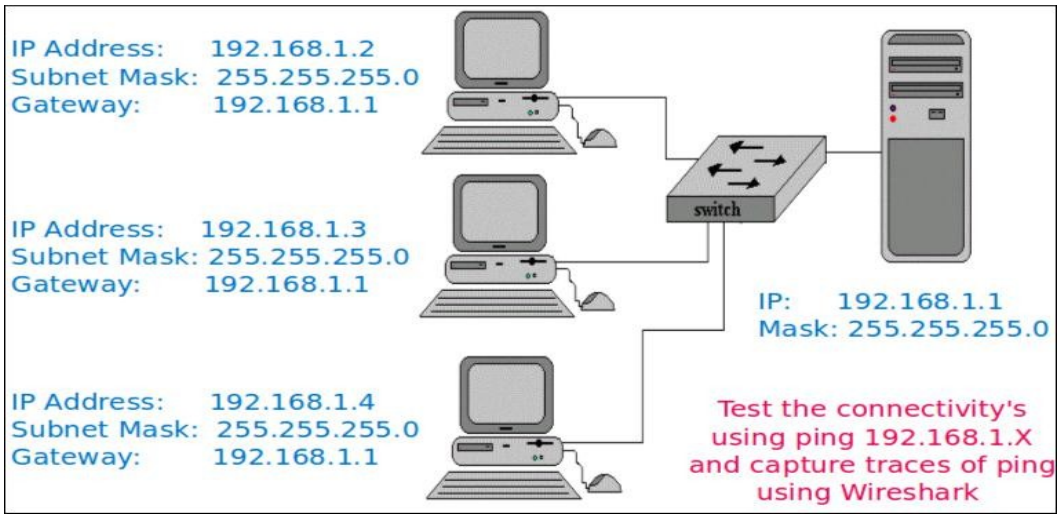
In a communications network, a network node equipped for interfacing with another network that uses different protocols. A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks. A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol.

Building and Testing of Wired Network:



Connect two machines using crossover cable and configure it using ip address, subnet mask and gateway address as shown in figure. Ping from both the machines and capture ICMP packets in Wireshark tool.

2. Setting Up LAN using Straight-Through Cable:



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.

3. 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.

Expected Print Attachment : Screen shot of Wireshark tool

Assignment No. 3

1. Problem Definition:

Write a program for error detection for 7/8 bits ASCII codes using CRC.

2. Prerequisite:

Basics of Error detection.

3. Learning Objectives:

To learn error detection for 7/8 bits ASCII codes using CRC.

4. Theory:

Error detection & Correction :

In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introducedduring transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases. Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory- less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts. Consequently, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst- error-detecting/correcting. Some codes can also be suitable for a mixture of random errors and burst errors.

CRC :

A Cyclic Redundancy Check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption. CRCs can be used for error correction, see bitfilters.

CRCs are so called because the check (data verification) value is a redundancy (it expands the message without adding information) and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyze mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the

check value has a fixed length, the function that generates it is occasionally used as a hash function.

Students should give Eg: Give any CRC example.

Conclusion:

Thus , we have studied the error detection technique for 7/8 bits ASCII codes using CRC.

Expected Print Attachment : CRC Program and it's output.

Assignment No. 4

Problem Definition:

Write a program using TCP sockets for wired network to implement Peer to Peer Chat.

1. Prerequisite:

1. Transport Layer: Roles, Protocols(TCP,UDP)
2. Java Programming Syntax
3. Socket Concept

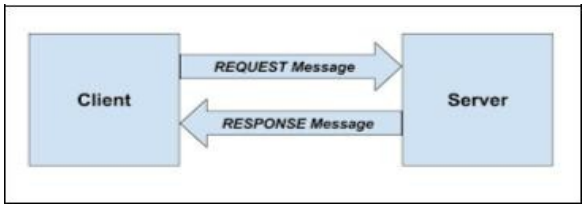
2. Learning Objectives:

- Students will able to understand socket programming.
- Students will able to design networking applications using TCP protocol.

3. 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:

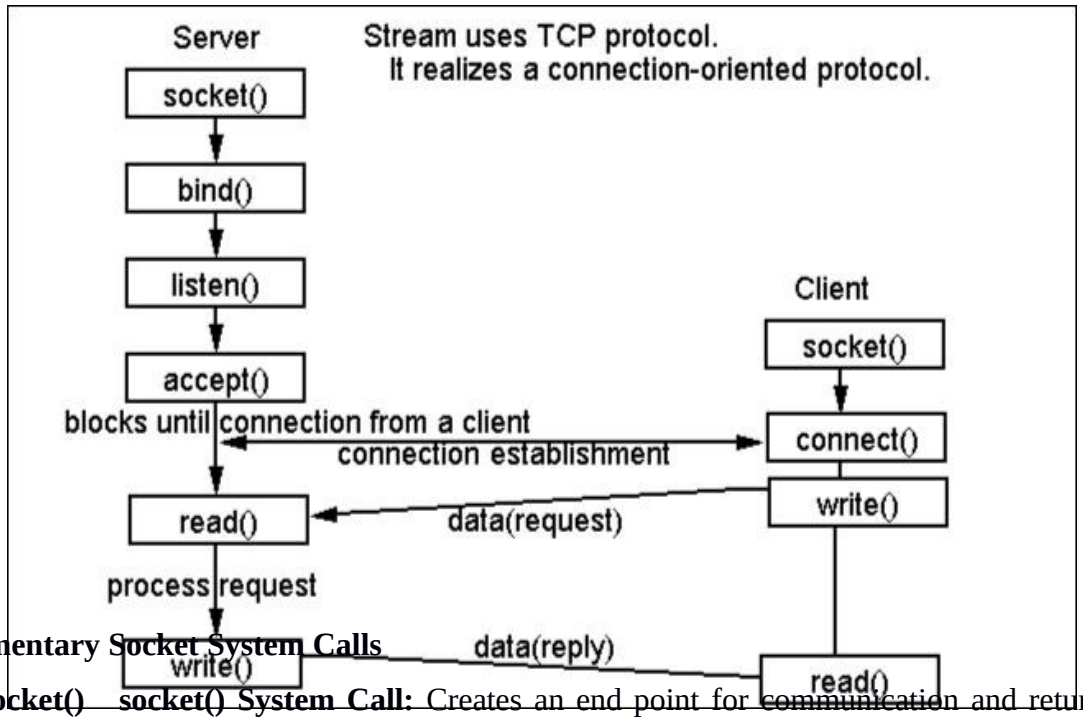
Server

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 a descriptor.

#include <sys/socket.h>

#include <sys/types.h>

int socket (int Address Family, int Type, int Protocol);

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 (int sockfd, struct sockaddr *myaddr, int addrlen);

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

3. **connect() connect() System call:** The connect function is used by a TCP client to establish a connection with a TCP server.

#include <sys/socket.h>

int connect(int sockfd, struct sockaddr *servaddr, int addrlen);

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 (int sockfd, 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 (int sockfd, struct sockaddr *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( ), sendto( ), recv( ) and recvfrom( )	**Send( ),sendto( ),recv( ) and recvfrom( ) system calls:** These system calls are similar to the standard read and write functions, but one additional argument is required.
and recvfrom( )	**#include <sys/socket.h>** **int send(int sockfd, char *buff, int nbytes, int flags);** **int sendto(int sockfd, char void *buff, int nbytes, int flags, struct sockaddr *to, int addrlen);** **int recv(int sockfd, char *buff, int nbytes, int flags);** **int recvfrom(int sockfd, char *buff, int nbytes, int flags, struct sockaddr *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 a socket and terminate a TCP connection. **#include <unistd.h>** **int close (int sockfd);**

Conclusion:

Hence we have studied TCP Socket Programming and implemented a program using TCP socket for wired network for chatting purpose.

Expected Print Attachment : Both Client and Server Programs and it's output.

Assignment No. 7

Problem Definition:

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

1. Prerequisite:

Data Link Layer: Roles, Protocols

Java Programming Syntax

Learning Objectives:

- Students will able to understand Go back N and Selective Repeat Modes of Sliding Window Protocol
-

Theory

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

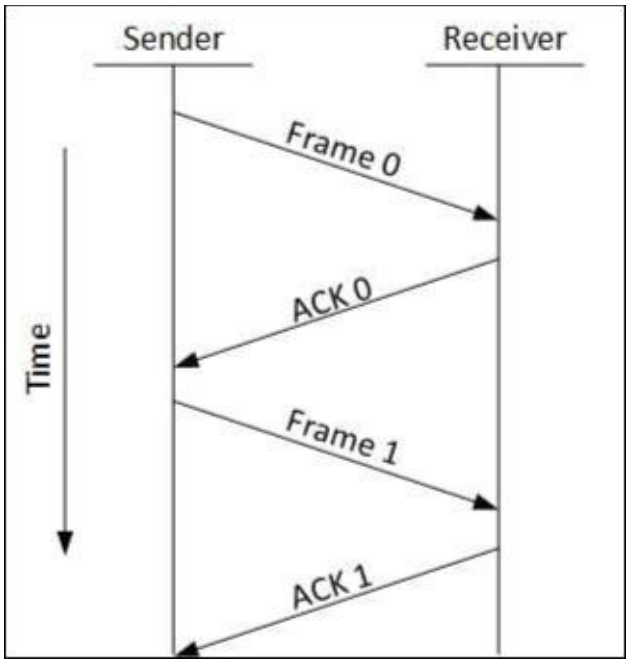
Flow Control

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:

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.

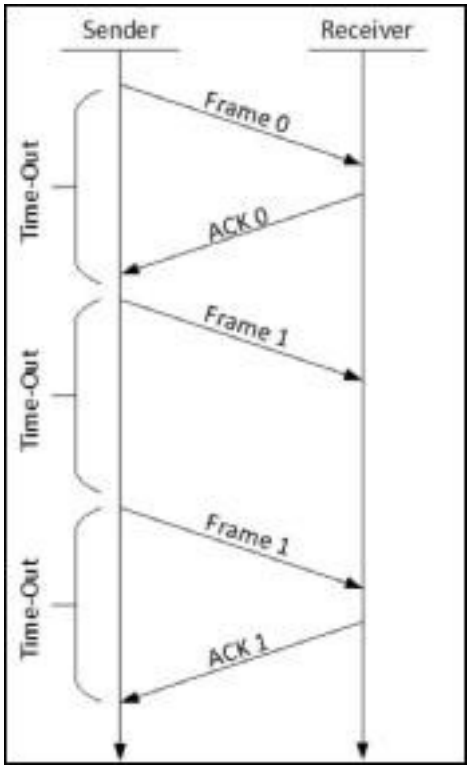
Requirements for error control mechanism:

6. **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
7. **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.

8. **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
9. **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement 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):

4. 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.

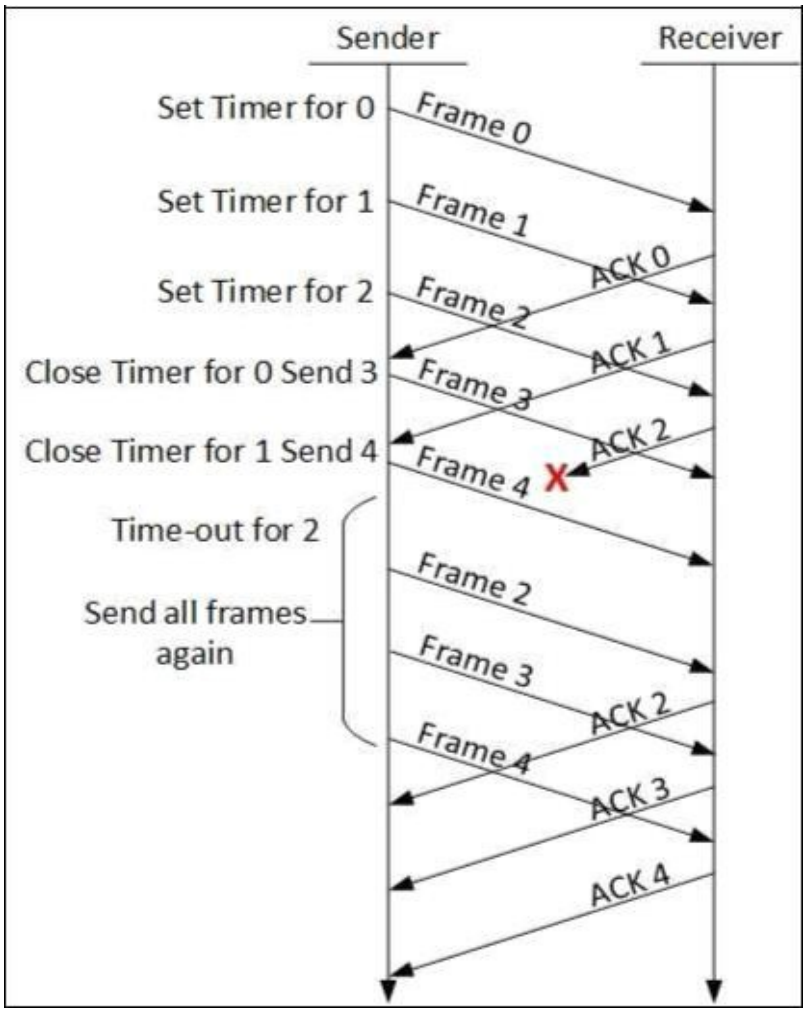
If acknowledgement of frame comes in time, the sender transmits the next frame in queue.

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.

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.

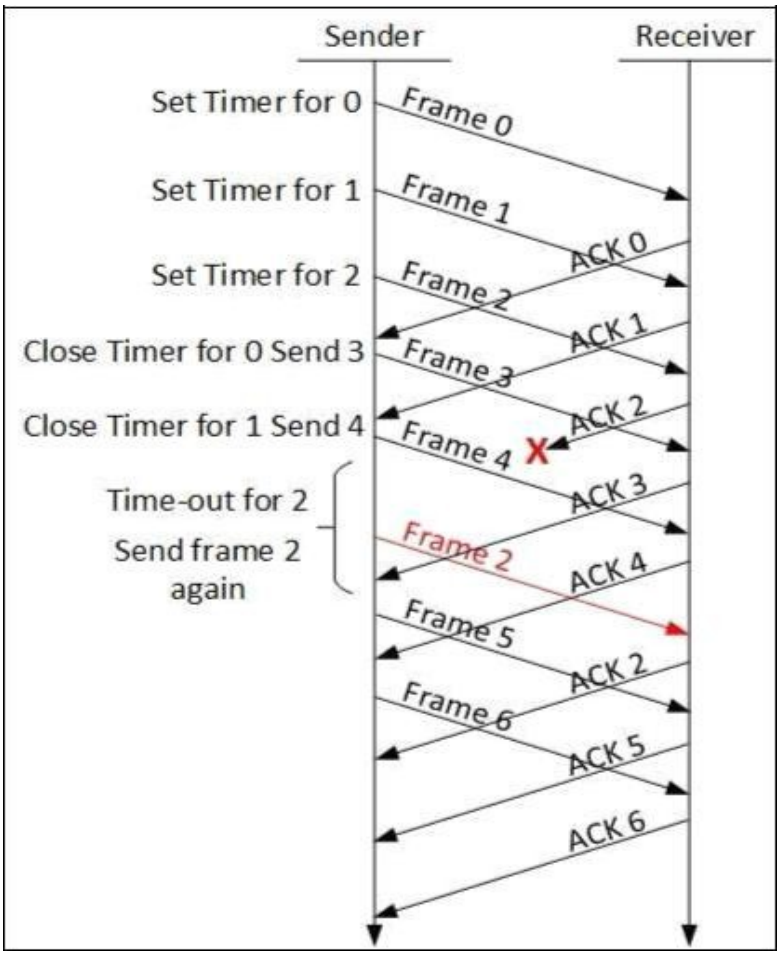


The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones. The receiving-window enables the receiver to receive multiple frames and acknowledge them. The receiver keeps track of incoming frame's sequence number.

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.

Testing

1. Run Wireshark tool
2. Run program
3. Capture packets in Wireshark

Conclusion:

Hence we have studied Go back N and Selective Repeat Modes of Sliding Window Protocol and captured packets in Wireshark tool.

Assignment No. 8

Problem Definition:

Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF/BGP.

1. Prerequisite:

1. Protocols: RIP, OSPF, BGP
2. Packet Tracer

2. Learning Objectives:

- Students will able to configure protocols like RIP, OSPF, BGP using Packet Tracer.

3. 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.

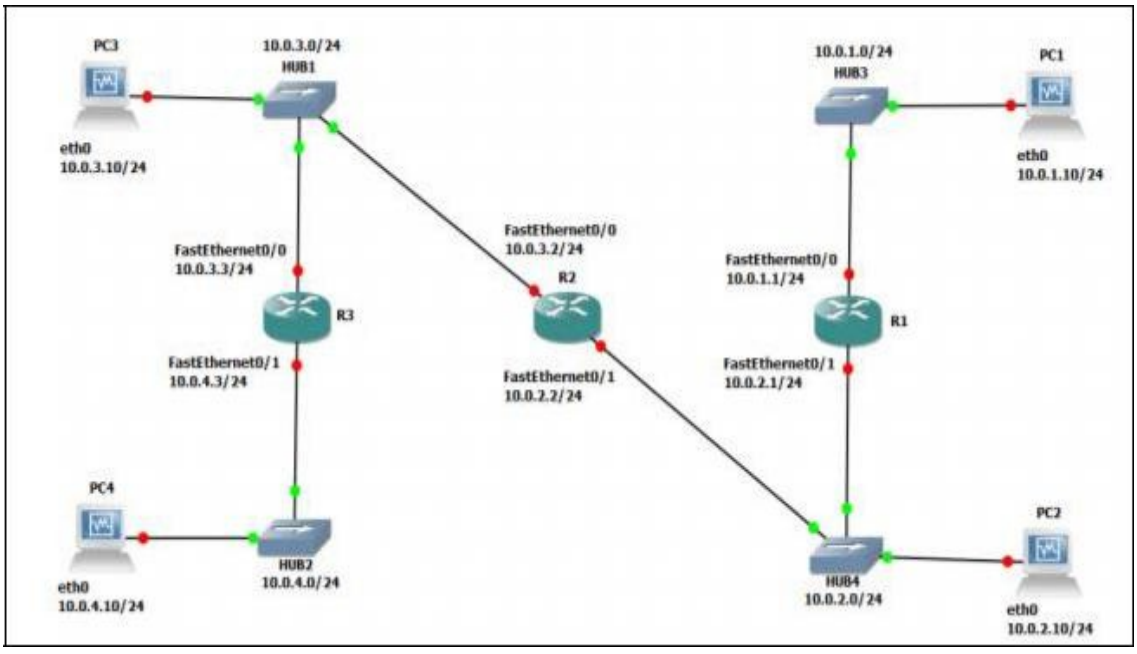
  So that you can select the best routing protocol (or protocols) for your network environment. If necessary, you can change which routes are chosen by altering the default metrics that a protocol assigns certain routes.

- **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 specify that 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 conserve bandwidth, 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.

Configuring RIP on CISCO ROUTER



Cisco Routers	Ethernet Interface FastEthernet 0/0	Ethernet Interface FastEthernet 0/1
Router1	10.0.1.1 / 24	10.0.2.1 / 24
Router2	10.0.3.2 / 24	10.0.2.2 / 24
Router3	10.0.3.3 / 24	10.0.4.3 / 24

Linux PC	Ethernet Interface eth0	Ethernet Interface eth1
PC1	10.0.1.10 / 24	Disabled
PC2	10.0.2.10 / 24	Disabled
PC3	10.0.3.10 / 24	Disabled
PC4	10.0.4.10 / 24	Disabled

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:

```
Router1>enable
Router1#configure terminal
Router1(config)#no ip routing
Router1(config)#ip routing
Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#network 10.0.0.0
Router1(config-router)#interface FastEthernet0/0
Router1(config-if)#no shutdown
Router1(config-if)#ip address 10.0.1.1 255.255.255.0
Router1(config-if)#interface FastEthernet0/1
Router1(config-if)#no shutdown
Router1(config-if)#ip address 10.0.2.1 255.255.255.0
Router1(config-if)#end
Router1#clear ip route *
```

4. 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.

5. From each router, issue a ping command to the IP address of interfaces FastEthernet0/0 and FastEthernet0/1 on all remote routers.

Conclusion:

Hence we have configured RIP using packet tracer.

Assignment No. 9

Problem Definition:

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

1. Prerequisite:

1. Application Layer: Roles, Protocols
2. Java Programming Syntax

2. Learning Objectives:

- Students will able to understand working of DNS Protocol

3. Theory

DNS

Domain Name System (DNS) is the default name resolution service used in a Microsoft Windows Server 2003 network. DNS is part of the Windows Server 2003 TCP/IP protocol suite and all TCP/IP network connections are, by default, configured with the IP address of at least one DNS server in order to perform name resolution on the network.

DNS Architecture

DNS architecture is a hierarchical distributed database and an associated set of protocols that define:

A mechanism for querying and updating the database.

A mechanism for replicating the information in the database among servers. A schema of the database.

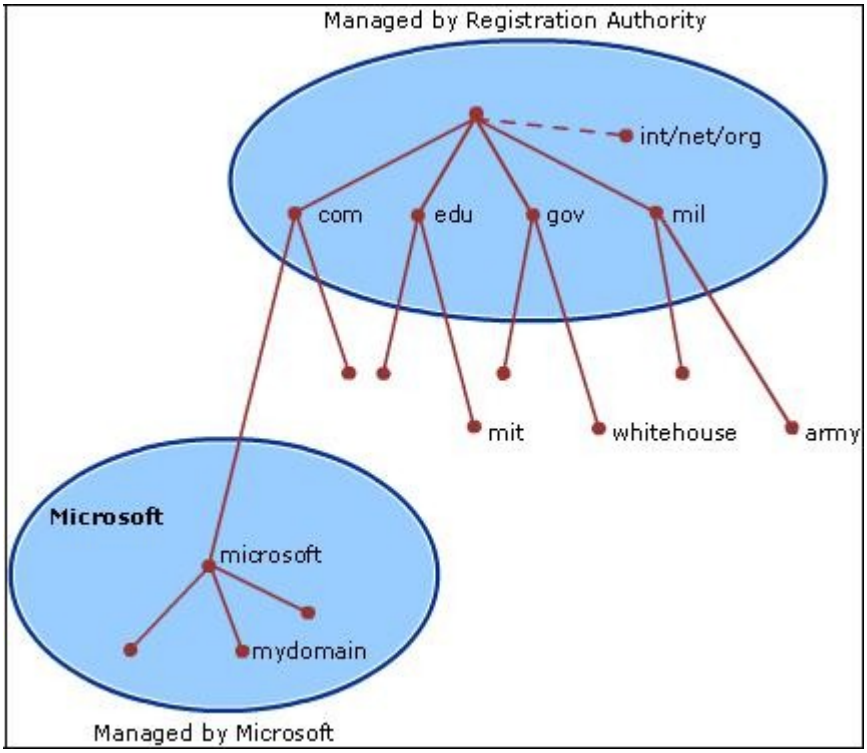
DNS Domain Names

The Domain Name System is implemented as a hierarchical and distributed database containing various types of data, including host names and domain names. The names in a DNS database form a hierarchical tree structure called the domain namespace. Domain names consist of individual labels separated by dots, for example: mydomain.microsoft.com.

A Fully Qualified Domain Name (FQDN) uniquely identifies the hosts position within the DNS hierarchical tree by specifying a list of names separated by dots in the path from the referenced host to the root. The next figure shows an example of a DNS tree with a host called mydomain

within the microsoft.com. domain. The FQDN for the host would be mydomain.microsoft.com.

DNS Domain Name Hierarchy



Types of DNS Domain Names

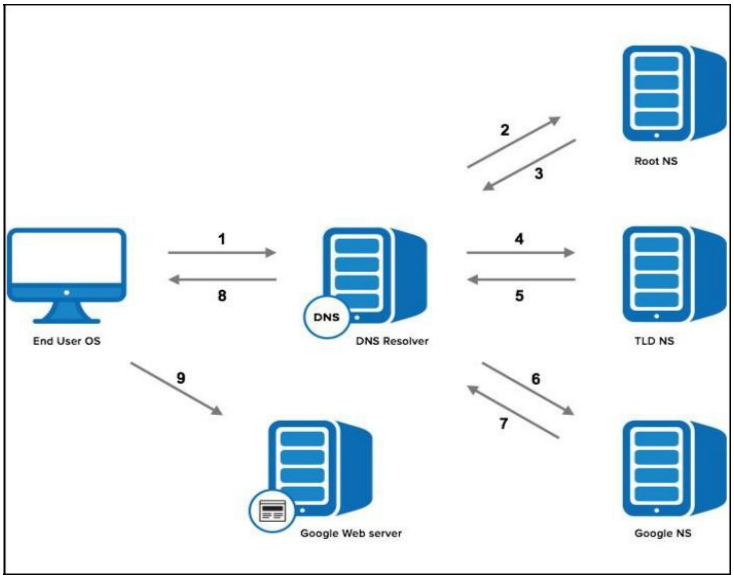
Name Type	Description	Example
Root domain	This is the top of the tree, representing an unnamed level; it is sometimes shown as two empty quotation marks (""), indicating a null value. When used in a DNS domain name, it is stated by a trailing period (.) to designate that the name is located at the root or highest level of the domain hierarchy. In this instance, the DNS domain name is considered to be complete and points to an exact location in the tree of names.	A single period (.) or a period used at the end of a name, such as "example.microsoft.com."
	Names stated this way are called fully	

	qualified domain names (FQDNs).	
Top level domain	Anameusedtoindicatea country/regionorthetypeof organization using a name.	".com", which indicates a name registered to a business for commercial use on the Internet.
Second level domain	Variable-length names registered to an individual or organization for use on the Internet. These names are always based upon an appropriate top-level domain, depending on the type of organization or geographic location where a name is used.	"microsoft.com.", which isthe second-level domain name registered to Microsoft by the Internet DNS domain name registrar.
Subdomain	Additional names that an organization can create that are derived from the registered second-level domain name. These include names added to grow the DNS tree of names in an organization and divide it into departments or geographic locations.	"example.microsoft.com.", which is a fictitious subdomain assigned byMicrosoft for use in documentation example names.
Host or resource name	Names that represent a leaf in the DNS tree of names and identify a specific resource. Typically, the leftmost label of a DNS domain name identifies a specific computer on the network. For example, if a name at this level is used in a host (A) RR, it is used to look up the IP address of computer based on its host name.	"host-a.example.microsoft.com.", where the first label ("host-a") is the DNS host name for a specific computer on the network.

Working of DNS Lookup

DNS is what translates your familiar domain name (www.google.com) into an IP address your

browser can use (173.194.33.174).



Before the page and any resource on the page is loaded, the DNS must be resolved so the browser can establish a TCP connection to make the HTTP request. In addition, for every external resource referenced by a URL, the DNS resolution must complete the same steps (per unique domain) before the request is made over HTTP. The DNS Resolution process starts when the user types a URL address on the browser and hits Enter. At this point, the browser asks the operating system for a specific page, in this case google.com.

Step 1: OS Recursive Query to DNS Resolver

Since the operating system doesn't know where "www.google.com" is, it queries a DNS resolver. The query the OS sends to the DNS Resolver has a special flag that tells it is a "recursive query." This means that the resolver must complete the recursion and the response must be either an IP address or an error.

Step 2: DNS Resolver Iterative Query to the Root Server

The resolver starts by querying one of the root DNS servers for the IP of "www.google.com." This query does not have the recursive flag and therefore is an "iterative query," meaning its response must be an address, the location of an authoritative name server, or an error. The root is represented in the hidden trailing "." at the end of the domain name. Typing this extra "." is not necessary as your browser automatically adds it.

Step 3: Root Server Response

These root servers hold the locations of all of the top level domains (TLDs) such as .com, .de, .io, and newer generic TLDs such as .camera.

The root doesn't have the IP info for "www.google.com," but it knows that .com might know, so it returns the location of the .com servers. The root responds with a list of the 13 locations of the .com gTLD servers, listed as NS or "name server" records.

Step 4: DNS Resolver Iterative Query to the TLD Server

Next the resolver queries one of the .com name servers for the location of google.com. Like the Root Servers, each of the TLDs have 4-13 clustered name servers existing in many locations. There are two types of TLDs: country codes (ccTLDs) run by government organizations, and generic (gTLDs). Every gTLD has a different commercial entity responsible for running these servers. In this case, we will be using the gTLD servers controlled by Verisign, who run the .com, .net, .edu, and .gov among gTLDs.

Step 5: TLD Server Response

Each TLD server holds a list of all of the authoritative name servers for each domain in the TLD. For example, each of the 13 .com gTLD servers has a list with all of the name servers for every single .com domain. The .com gTLD server does not have the IP addresses for google.com, but it knows the location of google.com's name servers. The .com gTLD server responds with a list of all of google.com's NS records. In this case Google has four name servers, "ns1.google.com" to "ns4.google.com."

Step 6: DNS Resolver Iterative Query to the Google.com NS

Finally, the DNS resolver queries one of Google's name server for the IP of "www.google.com."

Step 7: Google.com NS Response

This time the queried Name Server knows the IPs and responds with an A or AAAA address record (depending on the query type) for IPv4 and IPv6, respectively.

Step 8: DNS Resolver Response to OS

At this point the resolver has finished the recursion process and is able to respond to the end user's operating system with an IP address.

Step 9: Browser Starts TCP Handshake

At this point the operating system, now in possession of www.google.com's IP address,

provides the IP to the Application (browser), which initiates the TCP connection to start loading the page.

Conclusion:

Hence we have studied working of DNS protocol.

Assignment No. 10

Problem Definition:

Write a program to demonstrate subnetting and find the subnet masks.

1. Prerequisite:

1. Network Layer: Roles, Protocols
2. Java Programming Syntax

2. Learning Objectives:

Students will able to understand IP Addressing and Subnetting.

3. Theory

Introduction to Ipv4:

The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address. An IP address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. IP addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address. The address space of IPv4 is 232 or 4,294,967,296.

Network classes:

Internet addresses are allocated by the Inter NIC, the organization that administers the Internet. These IP addresses are divided into classes. The most common of these are classes A, B, and C. Classes D and E exist, but are not generally used by end users. Each of the address classes has a different default subnet mask. You can identify the class of an IP address by looking at its first octet. Following are the ranges of Class A, B, and C Internet addresses, each with an example address:

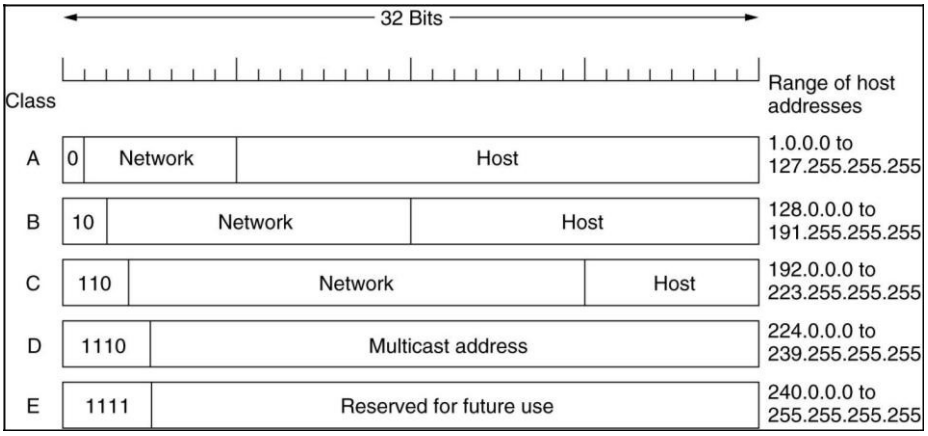
Class A networks use a default subnet mask of 255.0.0.0 and have 0-127 as their first octet. The address 10.52.36.11 is a class A address. Its first octet is 10, which is between 1 and 126, inclusive.

Class B networks use a default subnet mask of 255.255.0.0 and have 128-191 as their first octet. The address 172.16.52.63 is a class B address. Its first octet is 172, which is between 128 and 191, inclusive.

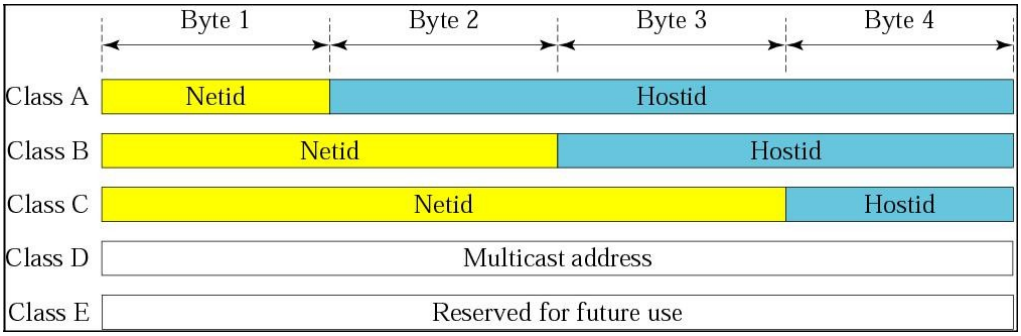
Class C networks use a default subnet mask of 255.255.255.0 and have 192-223 as their first octet. The address 192.168.123.132 is a class C address. Its first octet is 192, which is

between 192 and 223, inclusive.

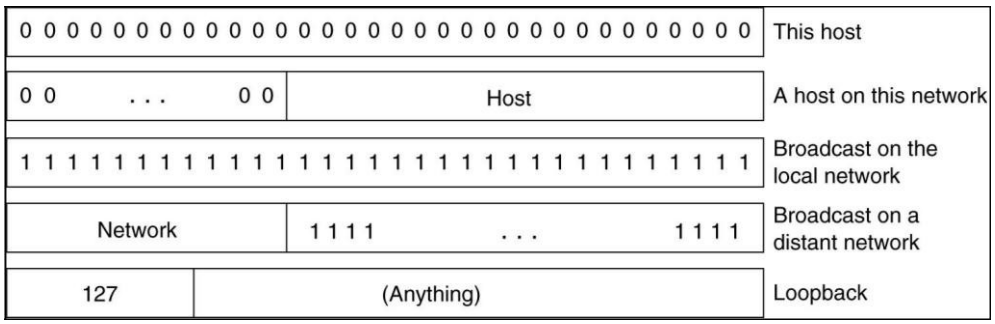
IPv4 Classes and It's Range



Distribution of NetId and HostId



Special IP Address



Addresses for Private Networks

Class	*Netids*	*Blocks*
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

The network address is the beginning address of each block. It can be found by applying the default mask to any of the addresses in the block (including itself). It retains the netid of the block and sets the hostid to zero.

Table 1: Default masks

Class	*Mask in binary*	*Mask in dotted-decimal*
A	**11111111** 00000000 00000000 00000000	**255.**0.0.0
B	**11111111 11111111** 00000000 00000000	**255.255.**0.0
C	**11111111 111111111 11111111** 00000000	**255.255.255.**0

Need of Subnetting

Specifically, the network addresses available for assignment to organizations are close to depletion. This is coupled with the ever-increasing demand for addresses from organizations that want connection to the Internet.

There are 4 of the major reasons for subnetting or segmenting network?

6. To divide a large network into smaller segments to reduce traffic and speed up the sections of your network.
7. To connect networks across geographical areas.
8. To connect different topologies such as Ethernet, Token Ring, and FDDI together via routers.
9. To avoid physical limitations such as maximum cable lengths or exceeding the maximum number of computers on a segment.

In this section we briefly discuss solution: Subnetting. A Class A, B, or C TCP/IP network can be further divided, or subnetted, by a system administrator.

Example:

A service provider has given you the Class C network range 209.50.1.0. Your company must break the network into 20 separate subnets.

Step 1) **Determine the number of subnets and convert to binary**

- In this example, the binary representation of 20 = 00010100.

Step 2) **Reserve required bits in subnet mask and find incremental value**

- The binary value of 20 subnets tells us that we need at least 5 network bits to satisfy this

requirement (since you cannot get the number 20 with any less than 5 bits – 10100)

1. Our original subnet mask is 255.255.255.0 (Class C subnet)
2. The full binary representation of the subnet mask is as follows: 255.255.255.0 = 11111111.11111111.11111111.00000000

- We must “convert” 5 of the client bits (0) to network bits (1) in order to satisfy the requirements: New Mask = 11111111.11111111.11111111.11111000
- If we convert the mask back to decimal, we now have the subnet mask that will be used on all the new networks – 255.255.255.248 - Our increment bit is the last possible network bit, converted back to a binary number:

  New Mask = 11111111.11111111.11111111.1111(1)000 – bit with the parenthesis is your increment bit. If you convert this bit to a decimal number, it becomes the number 8

Step 3) **Use increment to find network ranges**

8. Start with your given network address and add your increment to the subnetted octet: 209.50.1.0 209.50.1.8 209.50.1.16 …etc
1. You can now fill in your end ranges, which is the last possible IP address before you start the next range 209.50.1.0 – 209.50.1.7 209.50.1.8 – 209.50.1.15 209.50.1.16 – 209.50.1.23…etc
9. You can then assign these ranges to your networks. Remember the first and last address from each range (network / broadcast IP) is unusable.

Conclusion:

Hence we have studied IP Addressing and Subnetting.

Assignment No. 10

Problem Definition:

Write a program to simulate the behaviour of link state routing protocol to find suitable path for transmission.

1. Prerequisite:

1. Link State Routing Protocol
2. Distance vector routing protocol

2. Learning Objectives:

- Students will able to understand working of link state /Distance vector routing protocol.

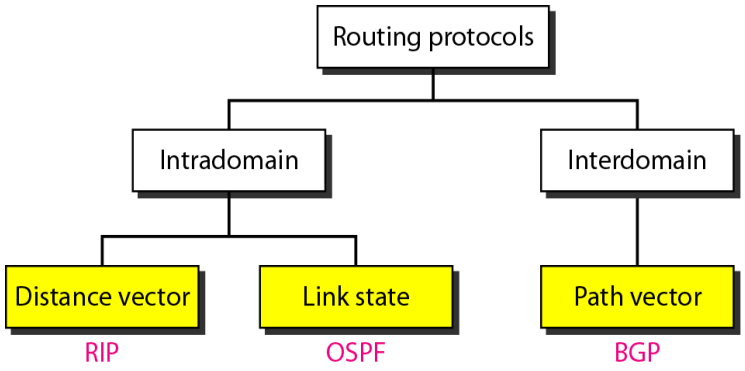
3. Theory

UNICAST ROUTING PROTOCOLS:

A routing table can be either static or dynamic. A static table is one with manual entries. A dynamic table, on the other hand, is one that is updated automatically when there is a change somewhere in the internet. Today, an internet needs dynamic routing tables. The tables need to be updated as soon as there is a change in the internet. For instance, they need to be updated when a router is down, and they need to be updated whenever a better route has been found.

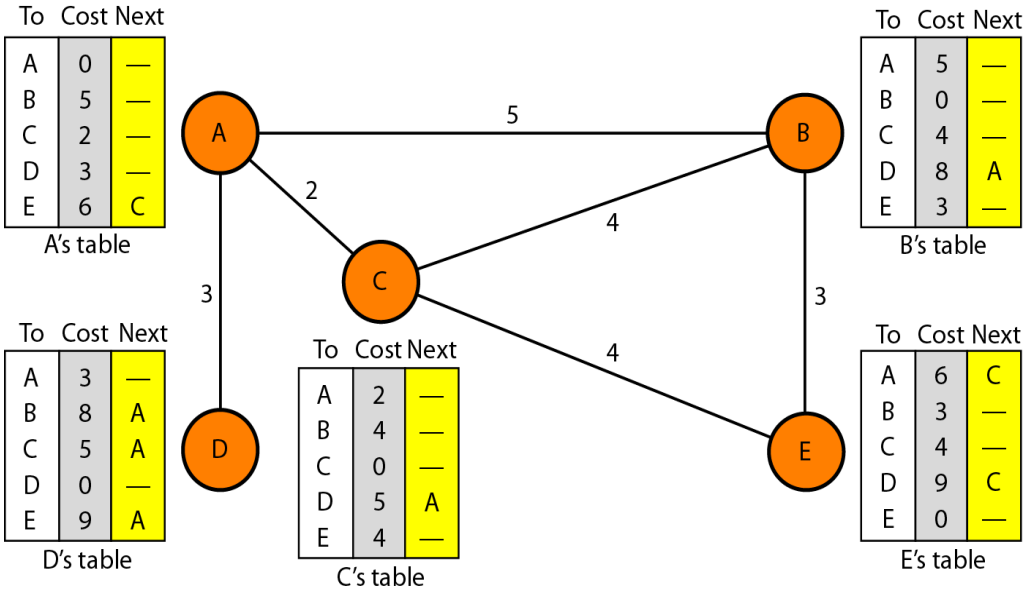
Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that lets routers in the internet inform each other of changes. It allows routers to share whatever they know about the internet or their neighborhood. The sharing of information allows a router in San Francisco to know about the failure of a network in Texas. The routing protocols also include procedures for combining information received from other routers.

Routing Information Protocol (RIP) is an implementation of the distance vector protocol. Open Shortest Path First (OSPF) is an implementation of the link state protocol. Border Gateway Protocol (BGP) is an implementation of the path vector protocol.



Distance Vector Routing:

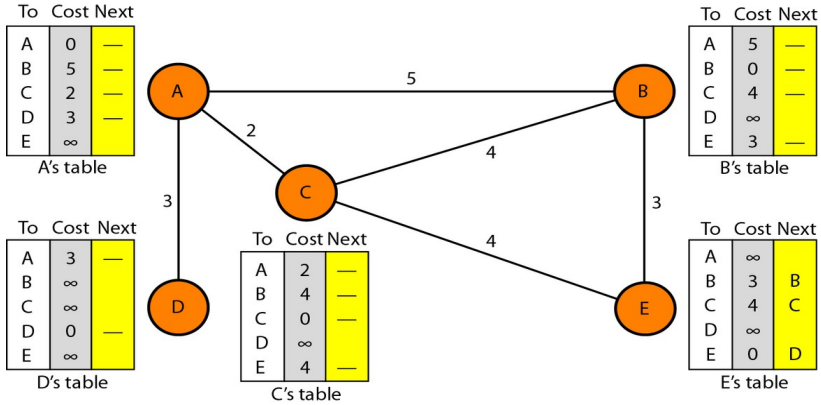
In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing). We can think of nodes as the cities in an area and the lines as the roads connecting them. A table can show a tourist the minimum distance between cities.



The table for node A shows how we can reach any node from this node. For example, our least cost to reach node E is 6. The route passes through C.

Initialization

The tables in Figure are stable; each node knows how to reach any other node and the cost. At the beginning, however, this is not the case. Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Next Figure shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).



Sharing:

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other. There is only one problem. How much of the table must be shared with each neighbor? A node is not aware of a neighbor's table. The best solution for each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard. However, the third column of a table (next stop) is not useful for the neighbor. When the neighbor receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table. A node therefore can send only the first two columns of its table to any neighbor. In other words, sharing here means sharing only the first two columns.

Updating :

When a node receives a two-column table from a neighbor, it needs to update its routing table.

Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is x mi, and the distance between A and C is y mi, then the distance between A and that destination, via C, is x + y mi.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

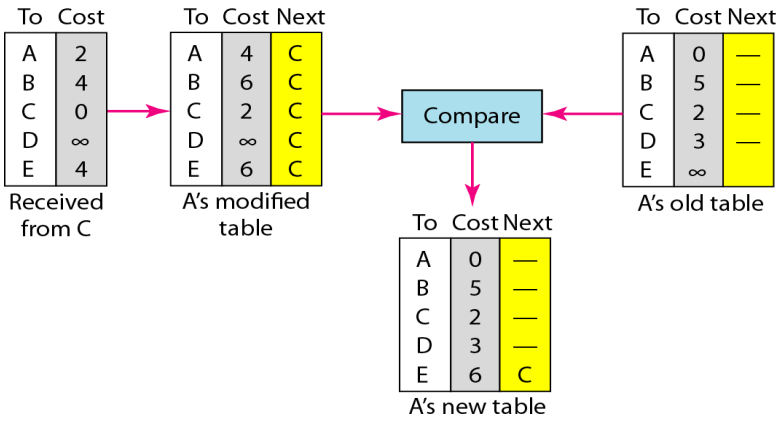
a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b. If the next-node entry is the same, the receiving node chooses the new row.

For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.

Next Figure shows how node A updates its routing table after receiving the partial table

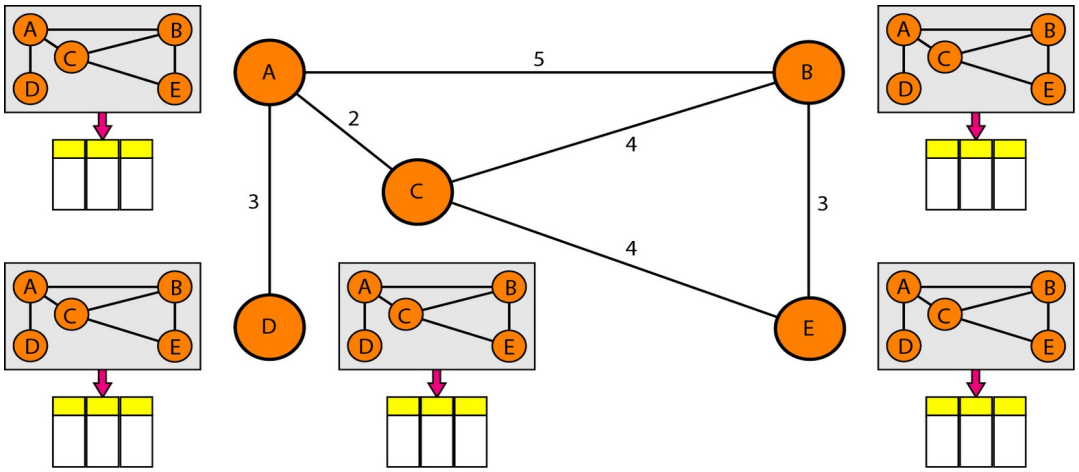
from node C.



Link State Routing :

Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.

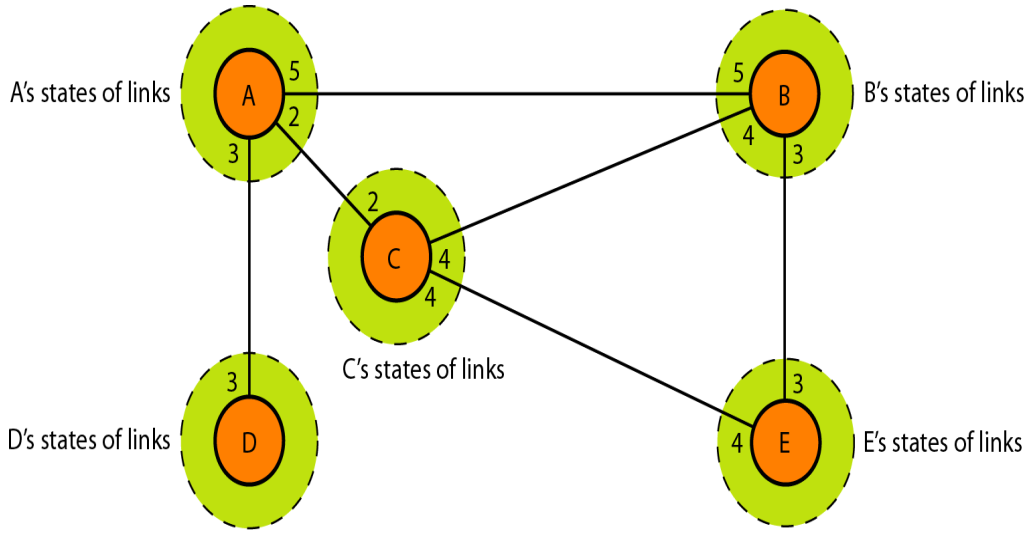
Next Figure shows the concept.



The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.

The topology must be dynamic, representing the latest state of each node and each link. If there are changes in any point in the network (a link is down, for example), the topology must be updated for each node. How can a common topology be dynamic and stored in each node? No node can know the topology at the beginning or after a change somewhere in the network. Link state

routing is based on the assumption that, although the global knowledge about the topology is not clear, each node has partial knowledge: it knows the state (type, condition, and cost) of its links. In other words, the whole topology can be compiled from the partial knowledge of each node. Next Figure shows the same domain as in previous figure, indicating the part of the knowledge belonging to each node.



Building Routing Tables:

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Conclusion:

Hence we have studied Link state and Distance vector routing protocol working in details.