Computer Networks (3150710)

GUJARAT TECHNOLOGICAL UNIVERSITY

C.K.PITHAWALA COLLEGE OF ENGINEERING AND TECHNOLOGY

Computer Engineering Department

Computer Network (3150710)

Index

Sr no.	Problem Statement	COs	Date	Grade	Sign
1.	Study of different network devices in detail.	CO1			
2.	Study of different types of network cables and practically implement the cross-wired cable and straight through cable using clamping tool.	CO1			
3.	Study of basic network command and Network configuration commands	CO1			
4.	Implement different LAN topologies using Network Simulator	CO1			
5.	Implement error detection methods using simple parity check and 2D parity check.	CO5			
6.	Implement error detection methods using CRC (Cyclic Redundancy Check).	CO5			
7.	Implement error correction method using Hamming Code.	CO5			
8.	Implement Bit Stuffing.	CO5			
9.	Implement the concept of VLAN using packet tracer	CO4			
10.	Implement the concept of static routing.	CO4			

11.	Implement the concept of dynamic routing (RIP, OSPF, BGP).	CO4			
12.	Packet capture and header analysis by wire-shark (TCP,UDP,IP)	CO3			

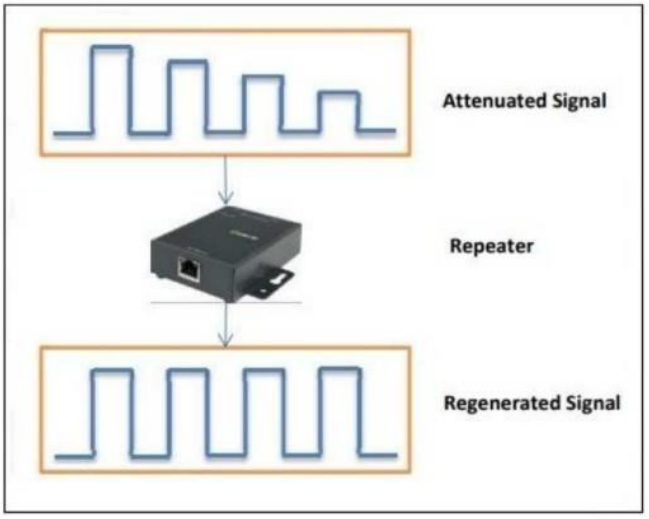
Practical 1

Aim →

Study of different network devices in detail.

Repeaters: ➔

- Repeaters are network devices operating at the physical layer of the OSI model that amplifies or regenerate an incoming signal before retransmitting it.
- They are incorporated into networks to expand their coverage area. They are also known as signal boosters.
- Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data
- A repeater receives a signal and, before it becomes too weak or corrupted, regenerates the original bit pattern.
- The repeater then sends the refreshed signal.
- A repeater can extend the physical length of a LAN.
- The location of a repeater on a link is vital. A repeater must be placed so that a signal reaches it before any noise changes the meaning of any of its bits.
- If the corrupted bit travels much farther, however, accumulated noise can change its meaning completely
- At that point, the original voltage is not recoverable, and the error needs to be corrected.
- A repeater placed on the line before the legibility of the signal becomes lost can still read the signal well enough to determine the intended voltages and replicate them in their original form.



Types of Repeaters: →

- According to the types of signals that they regenerate, repeaters can be classified into two categories −
  - Analog Repeaters − They can only amplify the analog signal.
  - Digital Repeaters − They can reconstruct a distorted signal.

- According to the types of networks that they connect, repeaters can be categorized into two types −
  - Wired Repeaters − They are used in wired LANs.
  - Wireless Repeaters − They are used in wireless LANs and cellular networks.

- According to the domain of LANs they connect, repeaters can be divided into two categories −
  - Local Repeaters − They connect LAN segments separated by small distances.
  - Remote Repeaters − They connect LANs that are far from each other.

Advantages of Repeaters: →

- Repeaters are simple to install and can easily extend the length of the coverage area of networks.
- They are cost-effective.
- Repeaters don’t require any processing overhead. The only time they need to be investigated is in case of degradation of performance.

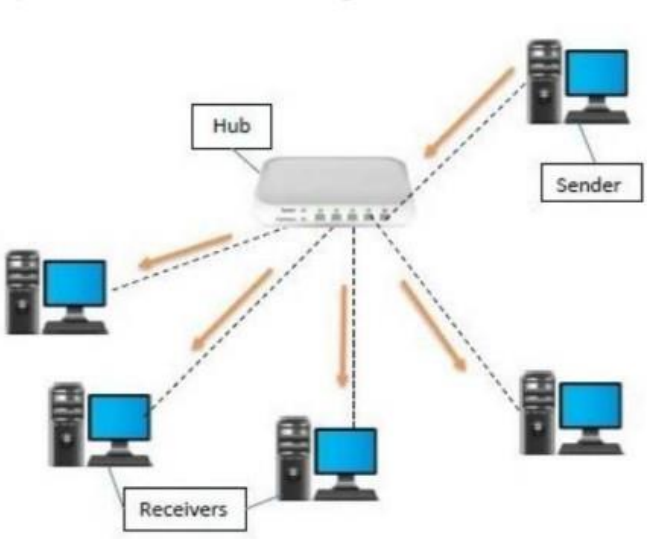
- They can connect signals using different types of cables.

Disadvantages of Repeaters: →

- Repeaters cannot connect dissimilar networks.
- They cannot differentiate between actual signal and noise ● They cannot reduce network traffic or congestion.
- Most networks have limitations on the number of repeaters that can be deployed.

Hub: →

- A hub is a physical layer networking device that is used to connect multiple devices in a network. They are generally used to connect computers in a LAN.
- A hub has many ports in it. A computer that intends to be connected to the network is plugged into one of these ports
- When a data frame arrives at a port, it is broadcast to every ot her port, without considering whether it is destined for a particular destination or not.



Types of Hubs:

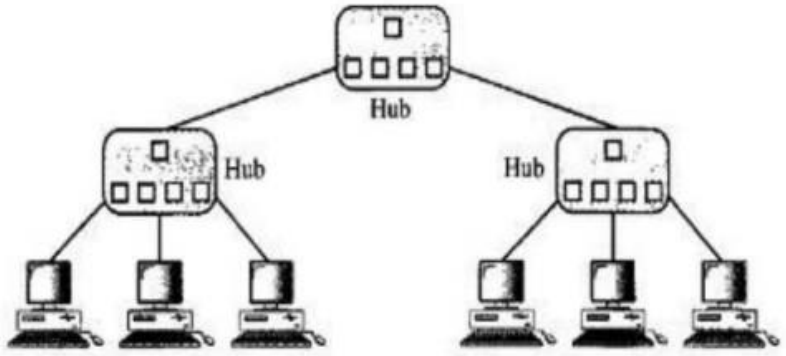
- *Passive Hubs*
  - A passive hub is just a connector. It connects the wires coming from different branches. In a star-topology Ethernet LAN, a passive hub is just a point where the signals coming from different stations collide; the hub is the collision point.

- This type of hub is part of the media; its location in the Internet model is below the physical layer.

- *Active Hubs*
  - An active hub is actually a multipart repeater. It is normally used to create connections between stations in a physical star topology.
  - However, hubs can also be used to create multiple levels of hierarchy, as shown in Figure. The hierarchical use of hubs removes the length limitation of 10Base-T (100 m).

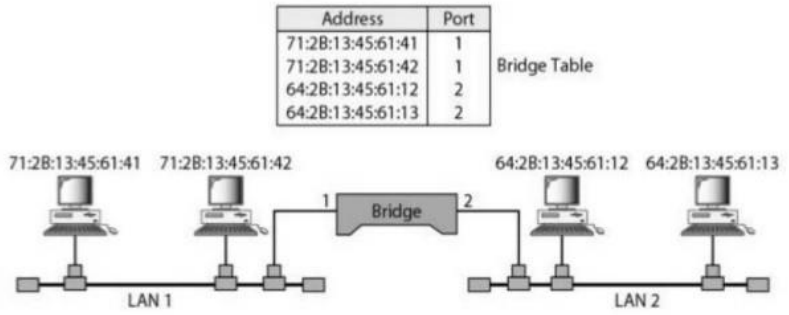
Intelligent Hubs:

- Intelligent hubs are active hubs that provide additional network management facilities. They can perform a variety of functions of more intelligent network devices like network management, switching, providing flexible data rates etc.



Bridges →

- A bridge operates in the physical layer as well as in the data link layer. It can regenerate the signal that it receives and as a data link layer device, it can check the physical addresses of the source and destination contained in the frame.
- The major difference between the bridge and the repeater is that the bridge and the repeater is that the bridge has a filtering capability.
- That means it can check the destination address of a frame and decide if the frame should be forwarded or dropped
- If the frame is forwarded, then the bridge should specify the port over which it should be forwarded.



Types of Bridges: →

- Transparent Bridges
  - These are the bridges in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

- Source Routing Bridges
  - In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The host can discover frames by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to the destination.

Router →

- Routers are networking devices operating at layer 3 or a network layer of the OSI model.
- They are responsible for receiving, analyzing, and forwarding data packets among the connected computer 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.
- A router is a three-layer device that routes packets based on their logical addresses (host-to-host addressing).
- A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route.
- The routing tables are normally dynamic and are updated using routing protocols. Data is grouped into packets or blocks of data.

- Each packet has a physical device address as well as a logical network address. The network address allows routers to calculate the optimal path to a workstation or computer.
- The router consults the routing table to determine the optimal route through which the data packets can be sent
- A routing table typically contains the following entities − ○ IP addresses and subnet mask of the nodes in the network.
    - ○ IP addresses of the routers in the network
    - ○ Interface information among the network devices and channels

Types of Routers →

- Wireless Router →

    They provide WiFi connection WiFi devices like laptops, smartphones, etc. They can also provide standard Ethernet routing. For indoor connections, the range is 150 feet while it's 300 feet for outdoor connections.

- Broadband Routers →

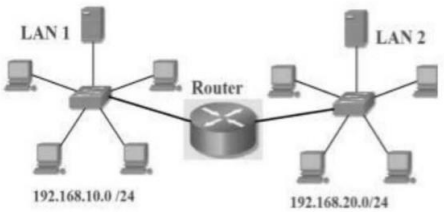
    They are used to connect to the Internet through telephone and to use voice over Internet Protocol (VoIP) technology for providing high-speed Internet access. They are configured and provided by the Internet Service Provider.

- Core Routers →

    They can route data packets within a given network, but cannot route the packets between the networks. They help to link all devices within a network thus forming the backbone of the network. It is used by ISP and communication interfaces.

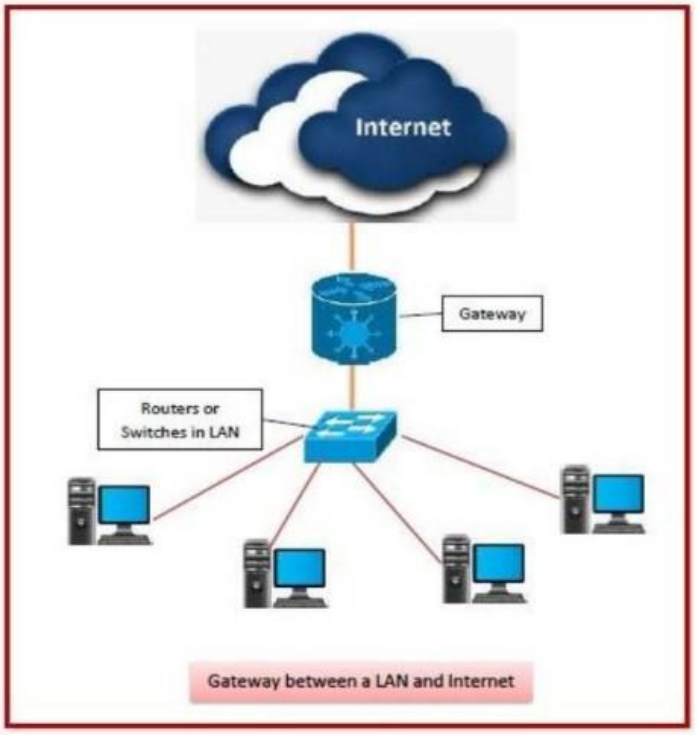
- Edge Routers →

    They are low-capacity routers placed at the periphery of the networks. They connect the internal network to the external networks and are suitable for transferring data packets across networks. They use the Border Gateway Protocol (BGP) for connectivity. There are two types of edge routers, subscriber edge routers and label edge routers.



Gateway →

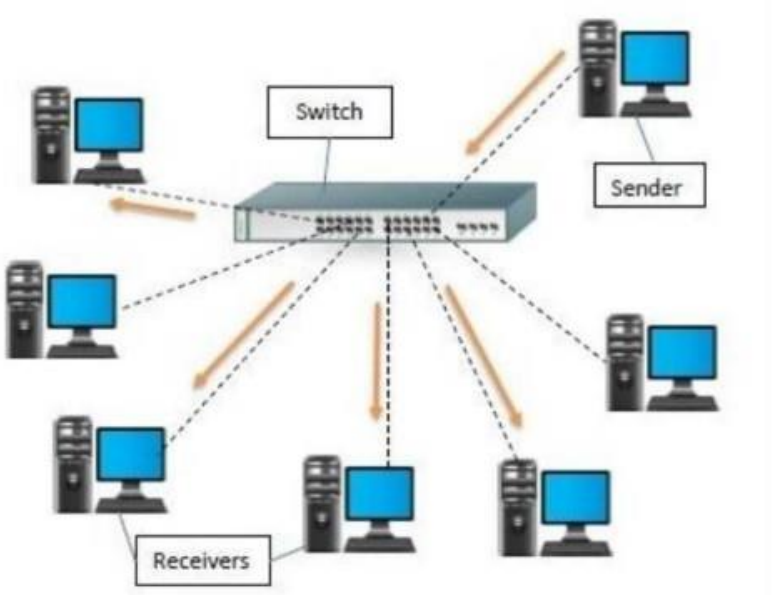
- A gateway is a network node that forms a passage between two networks operating with different transmission protocols.
- The most common type of gateway, the network gateway operates at layer 3, i.e. network layer of the OSI (open systems interconnection) model.
- However, depending upon the functionality, a gateway can operate at any of the seven layers of the OSI model.
- It acts as the entry-exit point for a network since all traffic that flows across the networks should pass through the gateway.
- Only the internal traffic between the nodes of a LAN does not pass through the gateway.



- Gateway is located at the boundary of a network and manages all data that inflows or outflows from that network.
- A gateway operates as a protocol converter, providing compatibility between the different protocols used in the two different networks.
- A gateway is generally implemented as a node with multiple NICs (network interface cards) connected to different networks. However, it can also be configured using the software.

Switch→

- A switch is a data link layer networking device that connects devices in a network and uses packet switching to send and receive data over the network.
- Like a hub, a switch also has many ports, into which computers are plugged.
- However, when a data frame arrives at any port of a network switch, it examines the destination address and sends the frame to the corresponding device(s).



Types of Switches →

- Unmanaged Switch
  - These are inexpensive switches commonly used in home networks and small businesses. They can be set up by simply plugging into the network, after which they instantly start operating. When more devices need to be added, more switches are simply added by this plug-and-play method. They are referred to as u-managed since they do not require to be configured or monitored.

- Managed Switch →
  - These are costly switches that are used in organizations with large and complex networks since they can be customized to augment the functionalities of a standard switch. The augmented features may be QoS (Quality of Service) like higher security levels, better

precision control, and complete network management. Despite their cost, they are preferred in growing organizations due to their scalability and flexibility. A simple Network Management Protocol (SNMP) is used for configuring managed switches.

- LAN Switch →
  - Local Area Network (LAN) switches to connect devices in the internal LAN of an organization. They are also referred to as Ethernet switches or data switches. These switches are particularly helpful in reducing network congestion or bottlenecks. They allocate bandwidth in a manner so that there is no overlapping of data packets in a network.

- PoE Switch →
  - Power over Ethernet (PoE) switches are used in PoE Gigabit Ethernets. PoE technology combines data and power transmission over the same cable so that devices connected to it can receive both electricity as well as data over the same line. PoE switches offer greater flexibility and simplify the cabling connections.

Sign:- _____ Date:- _____

Practical 2

Aim →

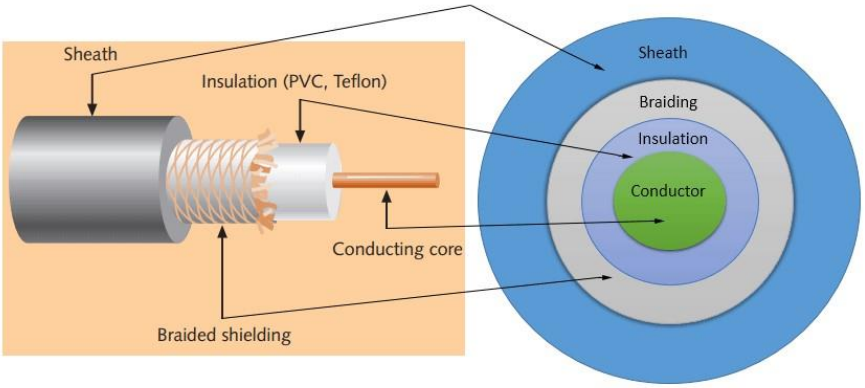
Study different types of network cables and practically implement the cross-wired cable and straight-through cable using a clamping tool.

→ To connect two or more computers or networking devices in a network, network cables are used. There are three types of network cables; coaxial, twisted-pair, and fiber-optic.

Coaxial cable

This cable contains a conductor, insulator, braiding, and sheath. The sheath covers the braiding, the braiding covers the insulation, and the insulation covers the conductor.

The following image shows these components.



Sheath

This is the outer layer of the coaxial cable. It protects the cable from physical damage.

Braided shield

This shield protects signals from external interference and noise. This shield is built from the same metal that is used to build the core.

Insulation

protects the core. It also keeps the core separate from the braided shield. Since both the core and the braided shield use the same metal, without this layer, they will touch each other and create a short circuit in the wire.

Conductor

The conductor carries electromagnetic signals. Based on the conductor a coaxial cable can be categorized into two types; single-core coaxial cable and multi-core coaxial cable.

A **single-core** coaxial cable uses a single central metal (usually copper) conductor, while a **multi-core** coaxial cable uses multiple thin strands of metal wires. The following image shows both types of cable.

Specifications of coaxial cables

have been in use for the last four decades. During these years, based on several factors such as the thickness of the sheath, the metal of the conductor, and the material used in insulation, hundreds of specifications have been created to specify the characteristics of coaxial cables.

Twisted-pair cables

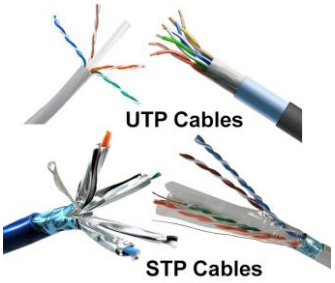
The twisted-pair cable was primarily developed for computer networks. This cable is also known as an **Ethernet cable**. Almost all modern LAN computer networks use this cable.

This cable consists of color-coded pairs of insulated copper wires. Every two wires are twisted around each other to form a pair. Usually, there are four pairs. Each pair has one solid color and one striped color wire. Solid colors are blue, brown, green, and orange. In stripped color, the solid color is mixed with the white color.

Based on how pairs are stripped in the plastic sheath, there are two types of twisted-pair cable; UTP and STP.

In the **UTP (*Unshielded twisted-pair*) cable**, all pairs are wrapped in a single plastic sheath.

In the **STP (*Shielded twisted-pair*) cable**, each pair is wrapped with an additional metal shield, then all pairs are wrapped in a single outer plastic sheath.



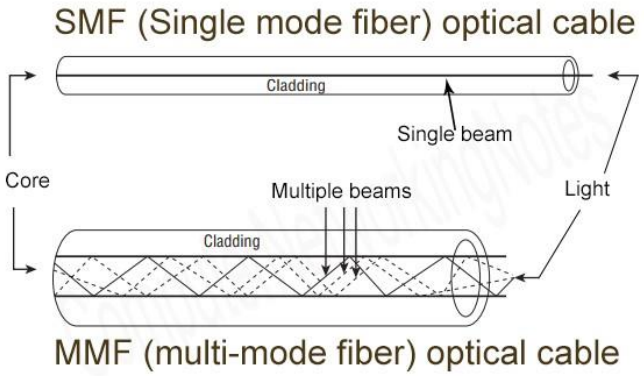
Fiber optic cable

This cable consists of a core, cladding, buffer, and jacket. The core is made from thin strands of glass or plastic that can carry data over a long distance. The core is wrapped in the cladding; the cladding is wrapped in the buffer, and the buffer is wrapped in the jacket.

- Core carries the data signals in the form of light.
- Cladding reflects light back to the core.
- Buffer protects the light from leaking.
- The jacket protects the cable from physical damage.

Fiber optic cable is completely immune to EMI and RFI. This cable can transmit data over a long distance at the highest speed. It can transmit data up to 40 kilometers at the speed of 100Gbps.

Fiber optic uses light to send data. It reflects light from one endpoint to another. Based on how many beams of light are transmitted at a given time, there are two types of fiber optic cable; SMF and MMF.



SMF (Single-mode fiber) optical cable

This cable carries only a single beam of light. This is more reliable and supports much higher bandwidth and longer distances than the MMF cable. This cable uses a laser as the light source and transmits 1300 or 1550 nano-meter wavelengths of light.

MMF (multi-mode fiber) optical cable

This cable carries multiple beams of light. Because of multiple beams, this cable carries much more data than the SMF cable. This cable is used for shorter distances. This cable uses an LED as the light source and transmits 850 or 1300 nano-meter wavelengths of light.

Procedure for the cross-wired cable and straight-through cable using a clamping tool.

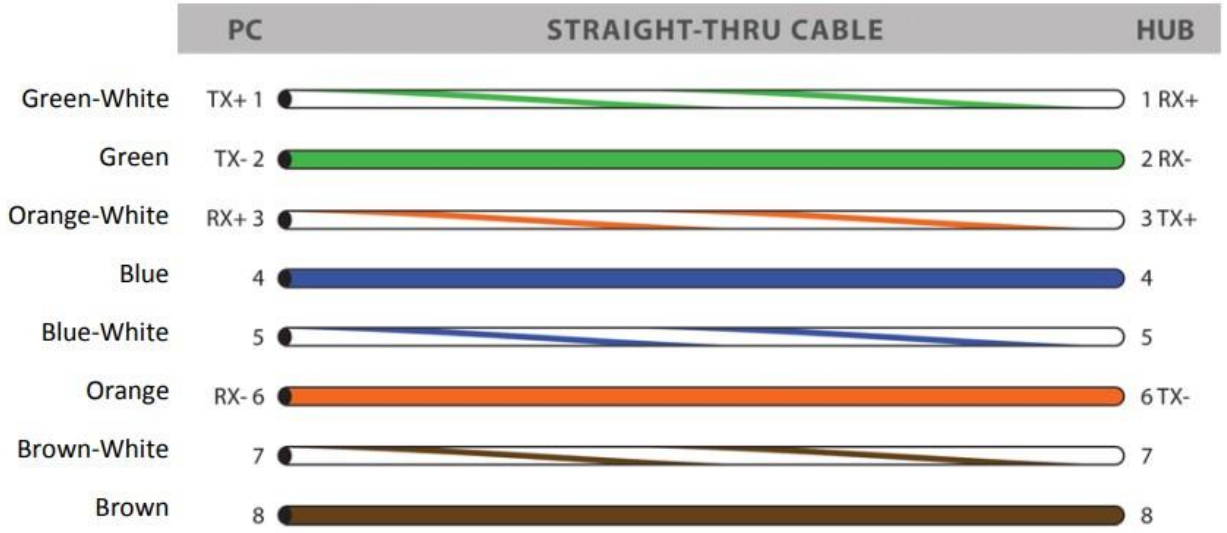
1. . Start by stripping off about 2 inches of the plastic jacket off the end of the cable. Be very careful at this point, as to not nick or cut into the wires, which are inside. Doing so could alter the characteristics of your cable, or even worse render is useless. Check the wires, one more time for nicks or cuts. If there are any, just whack the whole end off, and start over.

2. Spread the wires apart, but be sure to hold onto the base of the jacket with your other hand. You do not want the wires to become untwisted down inside the jacket. Category 5 cable must only have 1/2 of an inch of

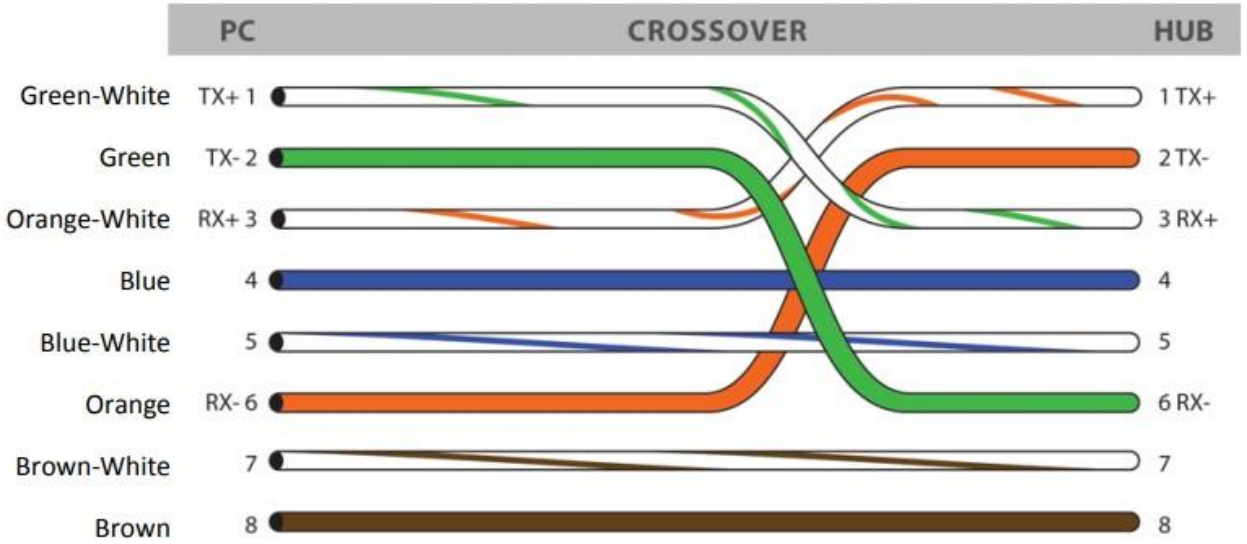
   'untwisted' wire at the end; otherwise, it will be 'out of spec'. At this point, you obviously have ALOT more than 1/2 of an inch of un-twisted wire.

3. You have 2 end jacks, which must be installed on your cable. If you are using a pre-made cable, with one of the ends whacked off, you only have one end to install - the crossed-over end. Below are two diagrams, which show how you need to arrange the cables for each type of cable end. Decide at this point which ends you are making and examine the associated picture below.

➢ **Diagram shows you how to prepare straight through wired connection**



➢ **Diagram shows you how to prepare Cross wired connection**



Sign:- _____ Date :- _____

Practical 3

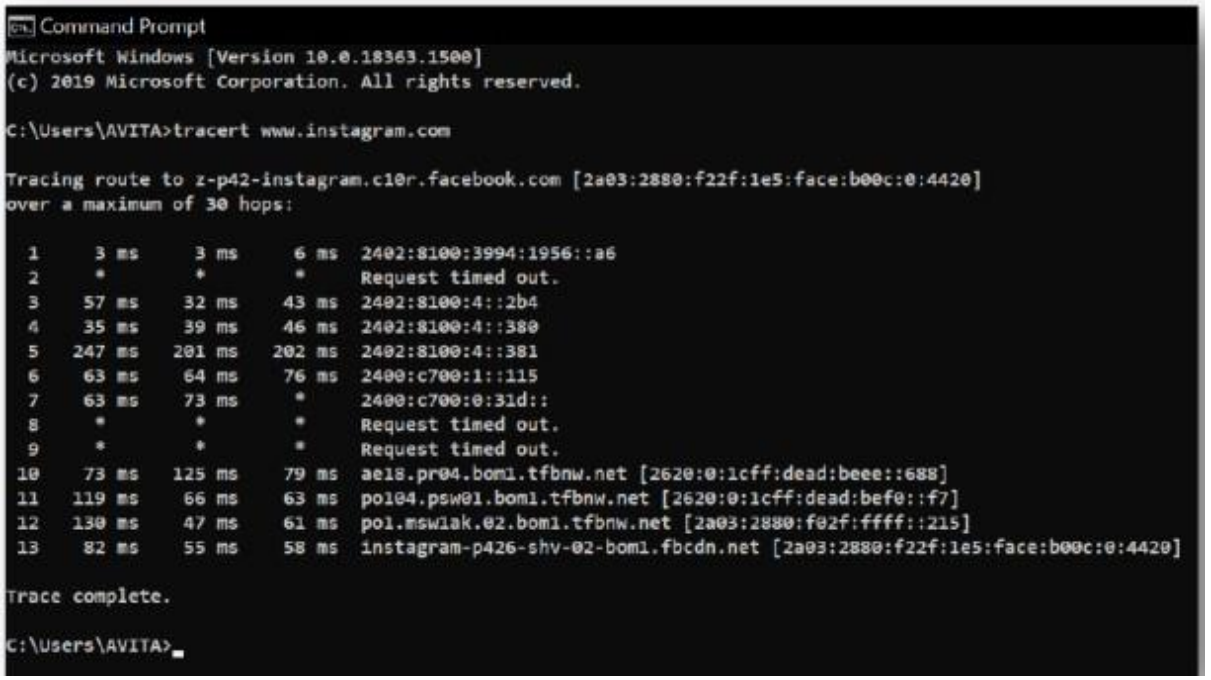
Aim →

Study basic network commands and network configuration commands.

1. → Tracert Command

The tracert command is a Command Prompt command that's used to show several details about the path that a packet takes from the computer or device you're on to whatever destination you specify.

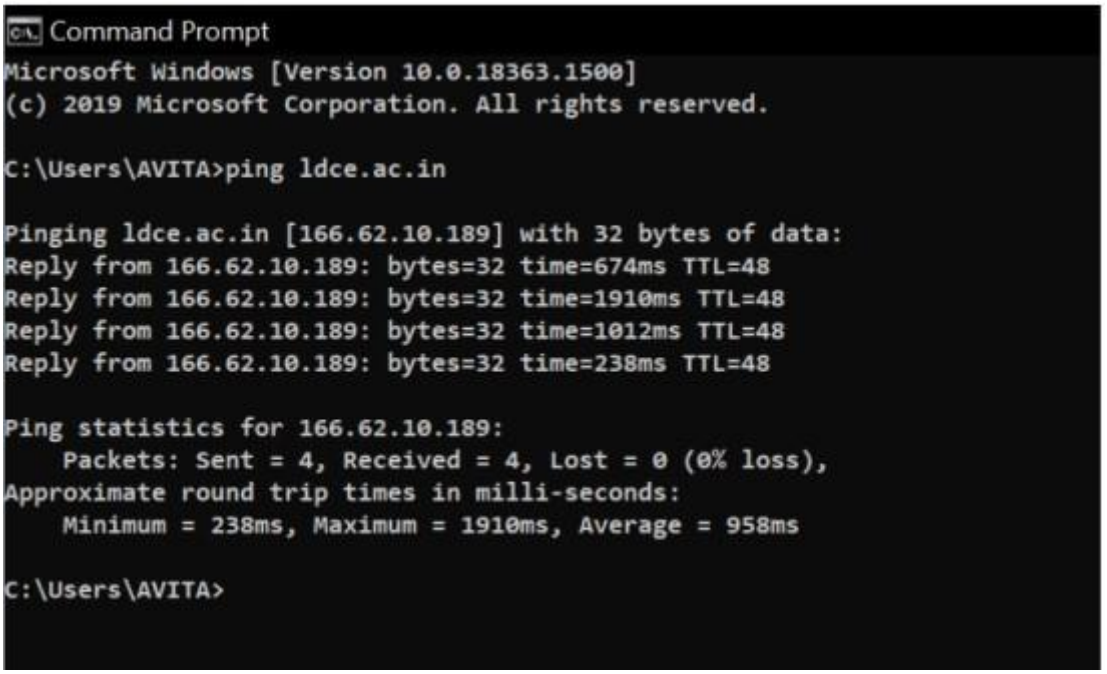
Tracert Command Option tracert –d :- This option prevents tracert from resolving IP addresses hostnames to , often resulting in much faster results.



```
Command Prompt
Microsoft Windows [Version 10.0.18363.1500]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\AVITA>tracert www.instagram.com

Tracing route to z-p42-instagram.c10r.facebook.com [2a03:2880:f22f:1e5:face:b00c:0:4420]
over a maximum of 30 hops:

  1     3 ms     3 ms     6 ms  2402:8100:3994:1956::a6
  2     *        *        *     Request timed out.
  3    57 ms    32 ms    43 ms  2402:8100:4::2b4
  4    35 ms    39 ms    46 ms  2402:8100:4::380
  5   247 ms   201 ms   202 ms  2402:8100:4::381
  6    63 ms    64 ms    76 ms  2400:c700:1::115
  7    63 ms    73 ms     *     2400:c700:0:31d::
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10    73 ms   125 ms    79 ms  ae18.pr04.bom1.tfbnw.net [2620:0:1cff:dead:beee::688]
 11   119 ms    66 ms    63 ms  po104.psw01.bom1.tfbnw.net [2620:0:1cff:dead:bef0::f7]
 12   130 ms    47 ms    61 ms  po1.mswiak.02.bom1.tfbnw.net [2a03:2880:f02f:ffff::215]
 13    82 ms    55 ms    58 ms  instagram-p426-shv-02-bom1.fbcdn.net [2a03:2880:f22f:1e5:face:b00c:0:4420]

Trace complete.

C:\Users\AVITA>_
```

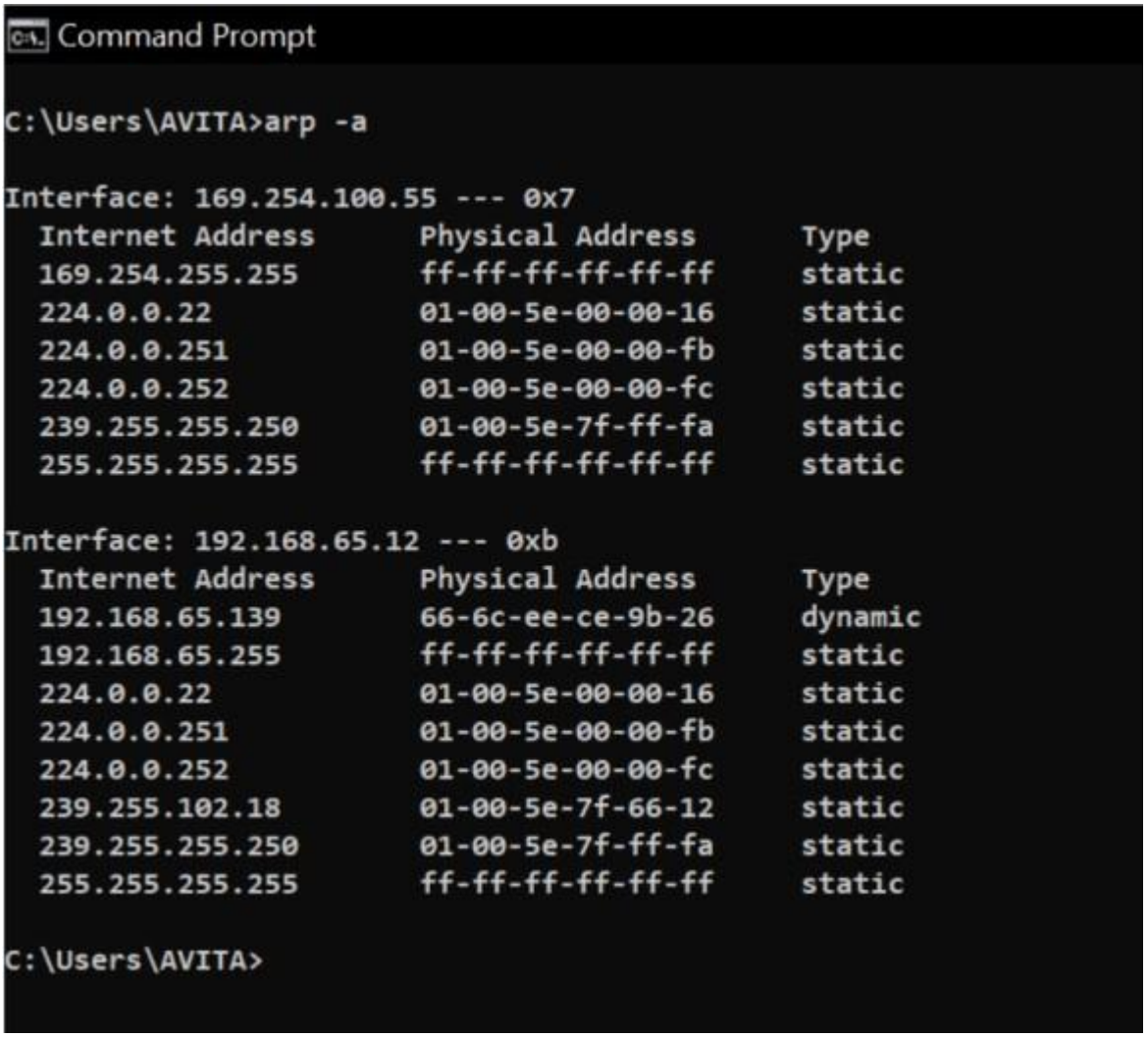
Ping Command →

Ping is a command-line utility, available on virtually any operating system with network connectivity, that acts as a test to see if a networked device is reachable. The ping command sends a request over the network to a specific device. A successful ping results in a response from the computer that was pinged back to the originating computer.



ARP Command

ARP stands for “Address Resolution Protocol “and is a protocol for mapping an IP address to a physical MAC address on a local area network. Basically, ARP is a program used by a computer system to find another computer’s MAC address based on its IP address. Communication between two computers on the same broadcast domain means a local area network. First, the client checks its ARP cache. is an ARP cache table of IP addresses with their corresponding MAC addresses.



Netstat Command

The netstat command, meaning, is a command network statistics prompt command used to display detailed information very about how your computer is communicating with other computers or network devices.

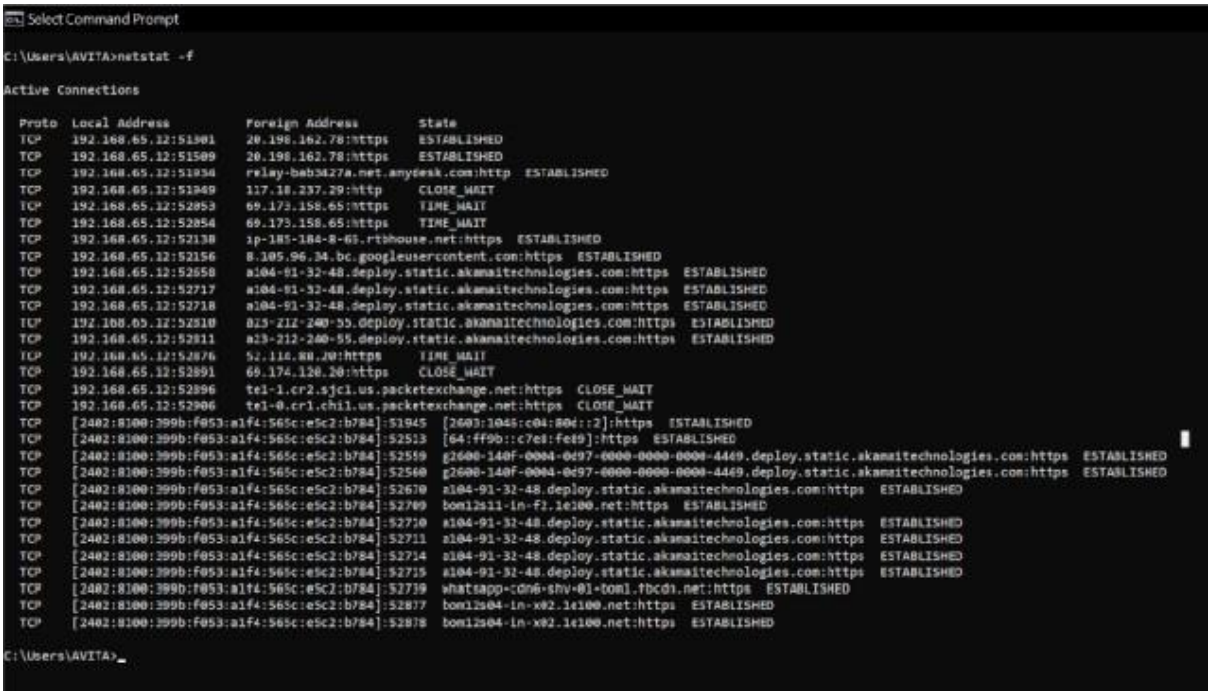
Specifically, the netstat command can show details about individual network connections, overall and -specific network protocol ng statistics, and much more, all of which could help troubleshoot certain kinds of networking issues.

Netstat Command Option 1. -f netstat –f:-

The switch will force the netstat command to display the Fully Qualified Domain Name (FQDN) for each foreign IP address when possible.

2. netstat –o:-

A handy option for many troubleshooting tasks, the switch displays the process identifier (PID) -o associated with each displayed connection. See the example below for more about using.



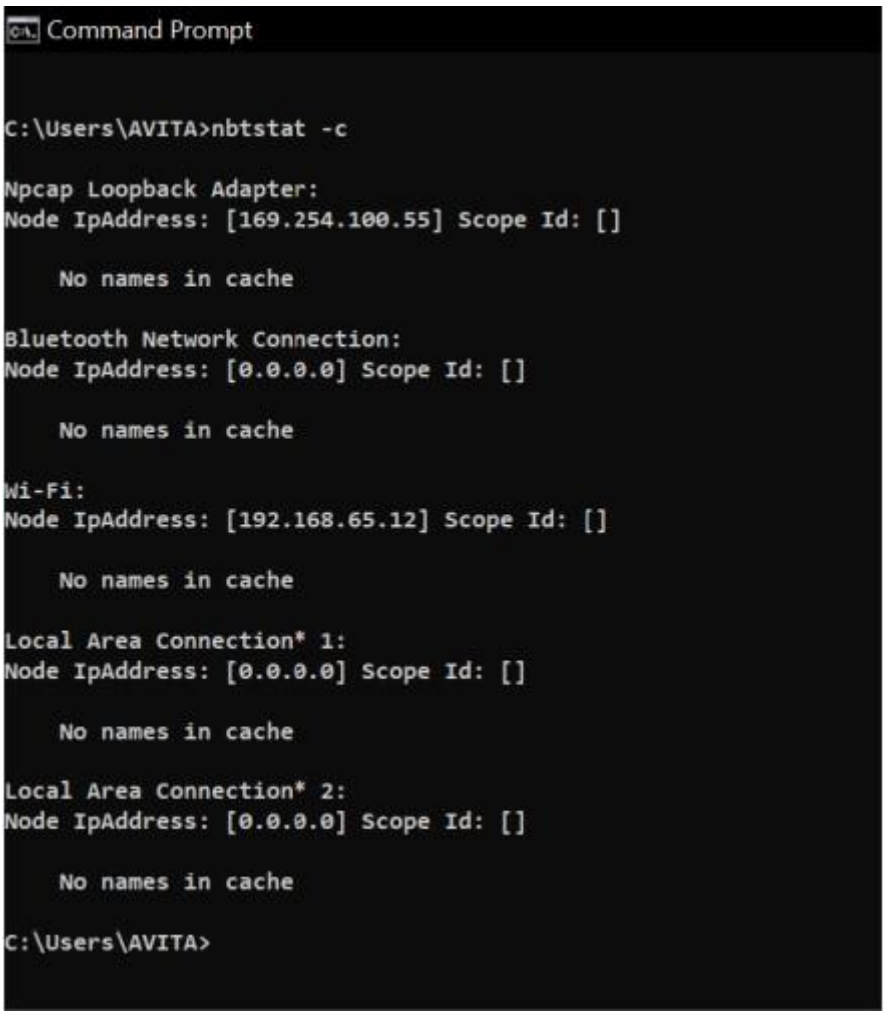
Nbstat Command

Nbtstat is a utility that displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP), which helps troubleshoot NetBIOS name resolution issues. Normally, name resolution is performed when NetBIOS over TCP/IP is functioning correctly. It does this through a local cache lookup, WINS or DNS server query or through LMHOSTS or hosts file lookup.

Netstat Command Option

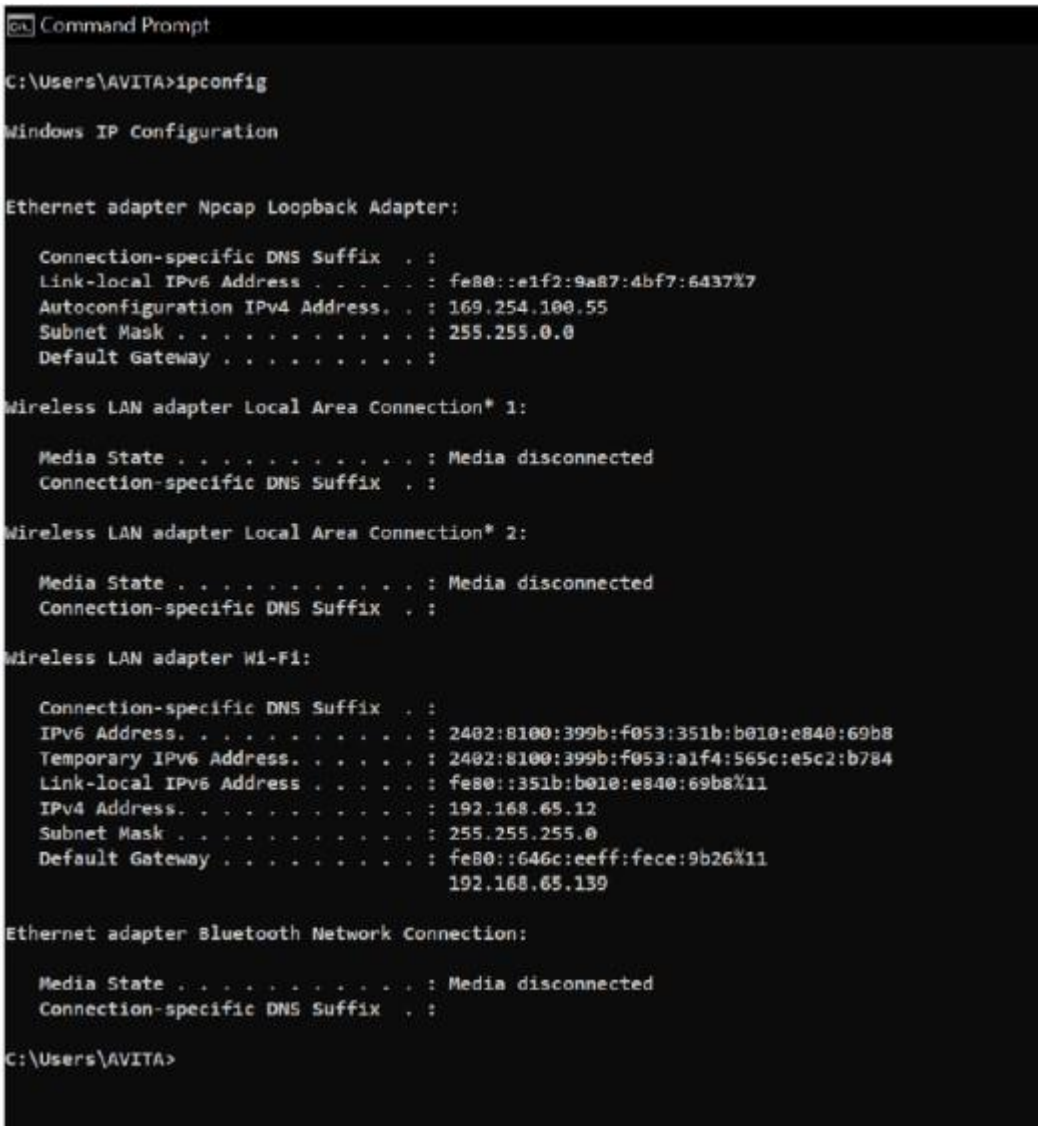
1. nbtstat –c :-

Lists NBT emote machine names and ’s cache of r their IP addresses.



ipconfig Command

IPconfig is a console application designed to run from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer. It also allows some control over your network adapters, IP addresses (DHCP-assigned specifically), and even your DNS cache.

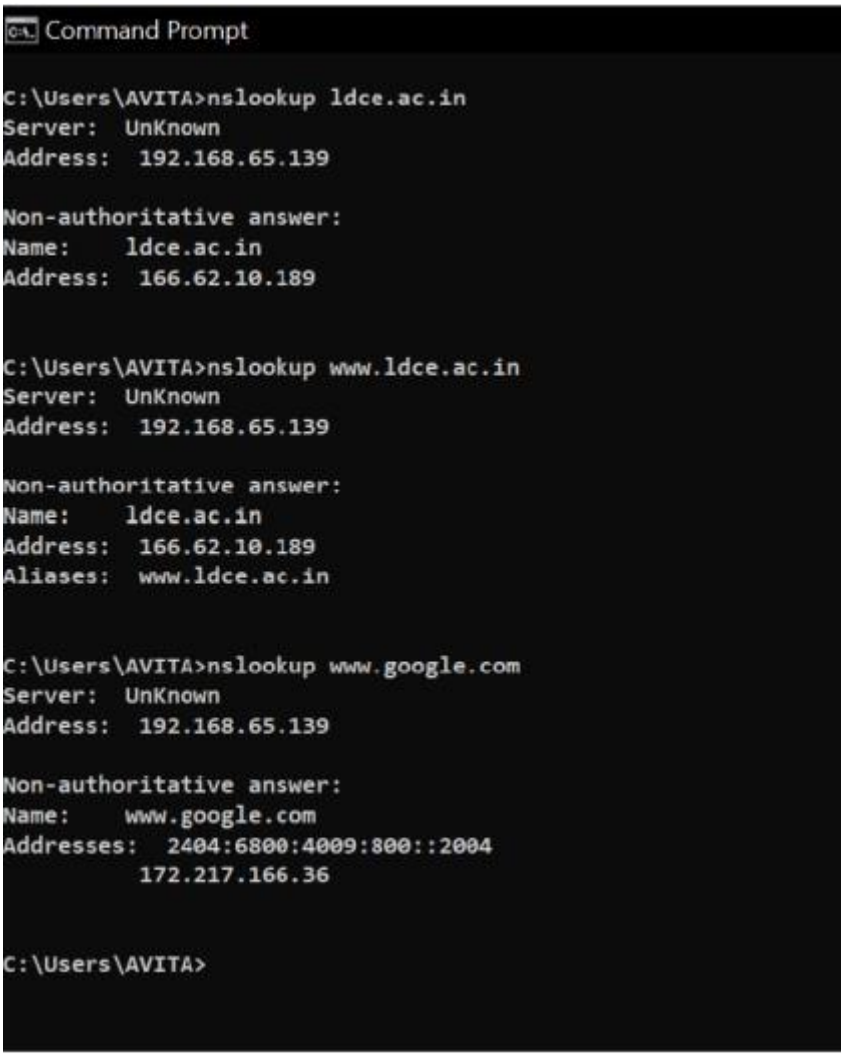


Winipcfg command

winipcfg, which stands for Windows IP Configuration, is one of the utility programs that Microsoft included in their operating systems starting from Windows 95. This utility is used in providing specific information about the computer's TCP/IP settings and configurations, such as IP and DNS addresses.

Nslookup Command

Nslookup is the name of a program that lets an Internet server administrator or any computer user enter a hostname (for example, "whatis.com") and find out the corresponding IP address or domain name system(DNS) record. The user can also enter a command for it to do a reverse DNS lookup and find the hostname for an IP address that is specified.



Sign:- _____ Date :- _____

Practical 4

Aim →

Implement different LAN topologies using Network Simulator.

Bus Topology

Bus topology is a network type in which every computer and network device is connected to a single cable. When it has exactly two endpoints, then it is called Linear Bus topology.

Features of Bus Topology

- It transmits data only in one direction
- Every device is connected to a single cable

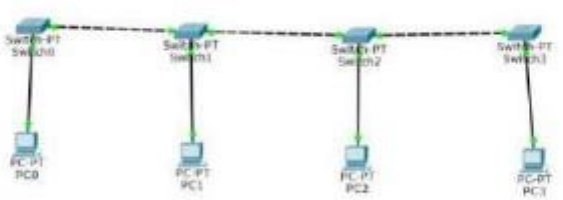
Advantages of Bus Topology ●

It is cost effective.

- Cable required is least compared to other network topology.
- Used in small networks.
- It is easy to understand.
- Easy to expand joining two cables together.

Disadvantages of Bus Topology

- Cables fail then the whole network fails.
- If network traffic is heavy or nodes are more the performance of the network decreases.
- Cable has a limited length.
- It is slower than the ring topology.



Ring Topology

It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first. Exactly two neighbors for each device.

Features of Ring Topology

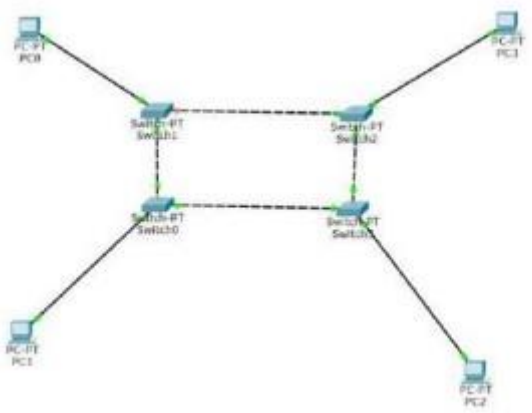
- A number of repeaters are used for Ring topology with large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node. Hence to prevent data loss repeaters are used in the 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 Dual Ring Topology, two ring networks are formed, and data flow is in opposite directions in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
- In Dual Ring Topology, two ring networks are formed, and data flow is in opposite directions in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.

Advantages

- Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
- Cheap to install and expand

Disadvantages

- Troubleshooting is difficult in ring topology.
- Adding or deleting the computers disturbs the network activity.
- Failure of one computer disturbs the whole network.



Star Topology

In this type of topology all the computers are connected to a single hub through a cable. This hub is the central node and all others nodes are connected to the central node.

Features

- Every node has its own dedicated connection to the hub.
- Hub acts as a repeater for data flow.
- Can be used with twisted pair, Optical Fibre, or coaxial cable.

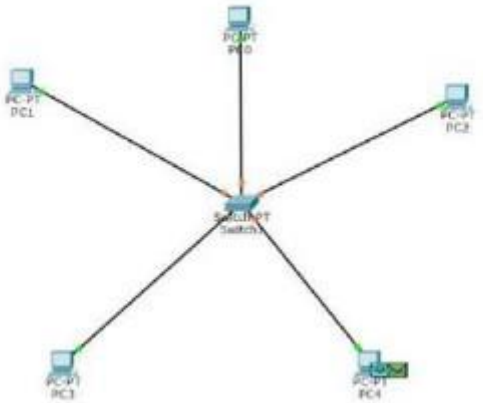
Advantages

- Fast performance with few nodes and low network traffic.
- Hub can be upgraded easily.
- Easy to troubleshoot
- Easy to set up and modify
- Only that node is affected which has failed, rest of the nodes can work smoothly.

Disadvantages

- Cost of installation is high.
- Expensive to use.
- If the hub fails then the whole network is stopped because all the nodes depend on the hub.
- Performance is based on the hub that is it depends on its capacity.

Implementation



Mesh Topology

It is a point-to-point connection to other nodes or devices. All the network nodes are connected to each other. Mesh has n(n1)/2 physical channels to link n devices.

Types of Mesh Topology

1. **Partial Mesh Topology** → In this topology some of the systems are connected in the same fashion as mesh topology but some devices are only connected to two or three devices.
2. **Full Mesh Topology** → Each and every node or device is connected to each other.

Features

- Fully connected
- Robust
- Not Flexible,

Advantages

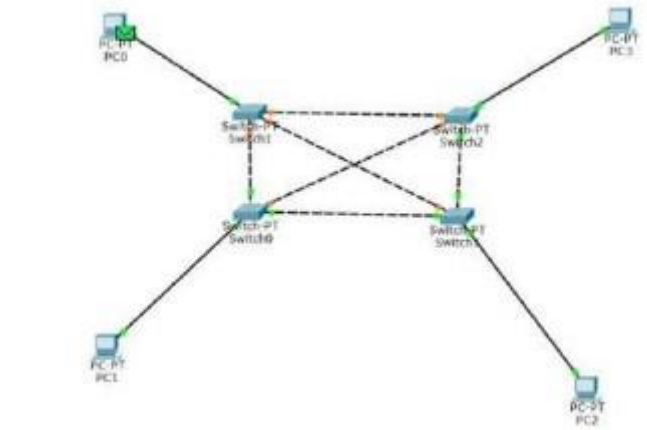
- Each connection can carry its own data load.
- It is robust.
- Fault is diagnosed easily.
- Provides security and privacy.

Disadvantages

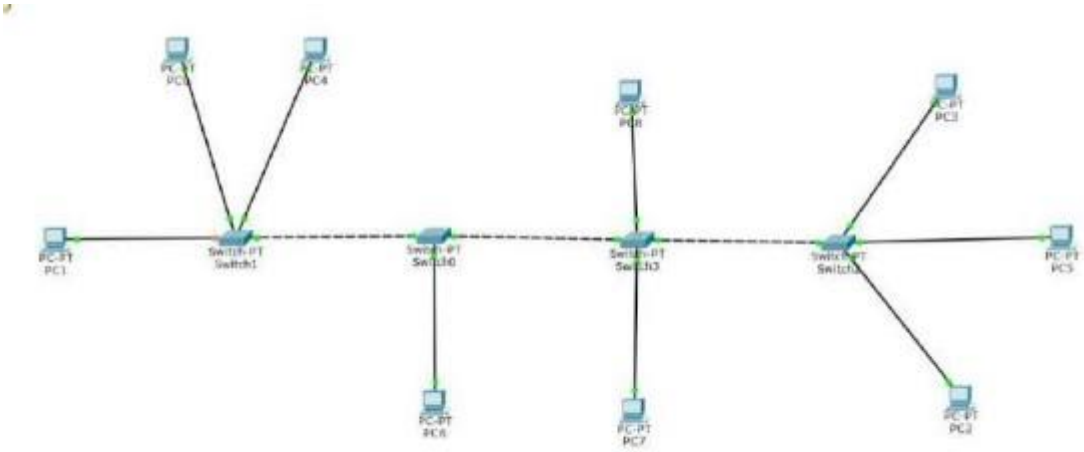
- Installation and configuration are difficult.

- Cabling cost is more.
- Bulk wiring is required.

Implementation of Mesh Topology



Implementation of Hybrid Topology



Sign:- _____ Date :- _____

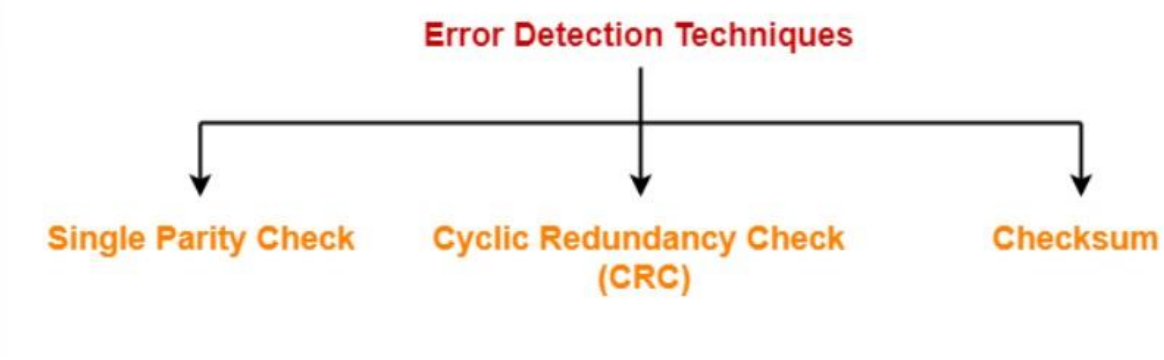
Practical 5

Aim →

Implement error detection methods using simple parity check and 2D parity check.

Error Detection Methods:-

Some popular error detection methods are-



1. Single Parity Check
2. Cyclic Redundancy Check (CRC)

Single Parity Check →

In this technique→

- One extra bit called a **parity bit** is sent along with the original data bits.
- Parity bit helps to check if any error occurred in the data during the transmission.

Steps Involved →

Step 1

At Sender Side,

- Total number of 1's in the data unit to be transmitted is counted.
- The total number of 1's in the data unit is made even in case of even parity.

- The total number of 1's in the data unit is made odd in case of odd parity.
- This is done by adding an extra bit called a **parity bit**.

Step 2

- The newly formed code word (Original data + parity bit) is transmitted to the receiver.

Step 3

At the receiver side,

- Receiver receives the transmitted code word.
- The total number of 1's in the received code word is counted.

Then, the following cases are possible-

- If a total number of 1's is even and even parity is used, then the receiver assumes that no error occurred.
- If a total number of 1's is even and odd parity is used, then the receiver assumes that the error occurred.
- If a total number of 1's is odd and odd parity is used, then the receiver assumes that no error occurred.
- If a total number of 1's is odd and even parity is used, then the receiver
  assumes that the error occurred.

Parity Check Example →

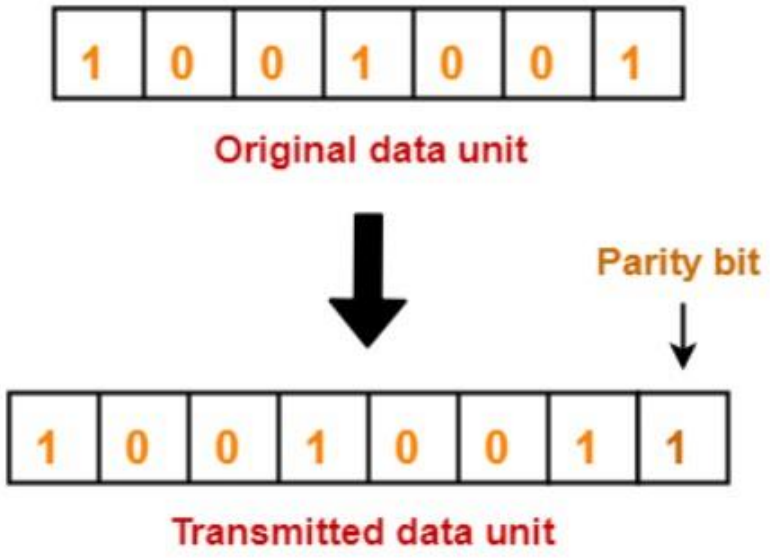
Consider the data unit to be transmitted is 1001001 and even parity is used.

Then,

On Sender Side-

- Total number of 1's in the data unit is counted.
- a Total number of 1's in the data unit = 3.
- Clearly, even parity is used and a total number of 1's is odd.
- So, parity bit = 1 is added to the data unit to make a total number of 1's even.

- Then, the code word 10010011 is transmitted to the receiver.



At Receiver Side-

- After receiving the code word, a total number of 1's in the code word is counted.
- Consider receiver receives the correct code word = 10010011.
- Even parity is used and a total number of 1's is even.
- So, the receiver assumes that no error occurred in the data during the transmission.

Advantage-

- This technique is guaranteed to detect an odd number of bit errors (one, three, five, and so on).
- If an odd number of bits flip during transmission, then the receiver can detect it by counting the number of 1s.

Limitation-

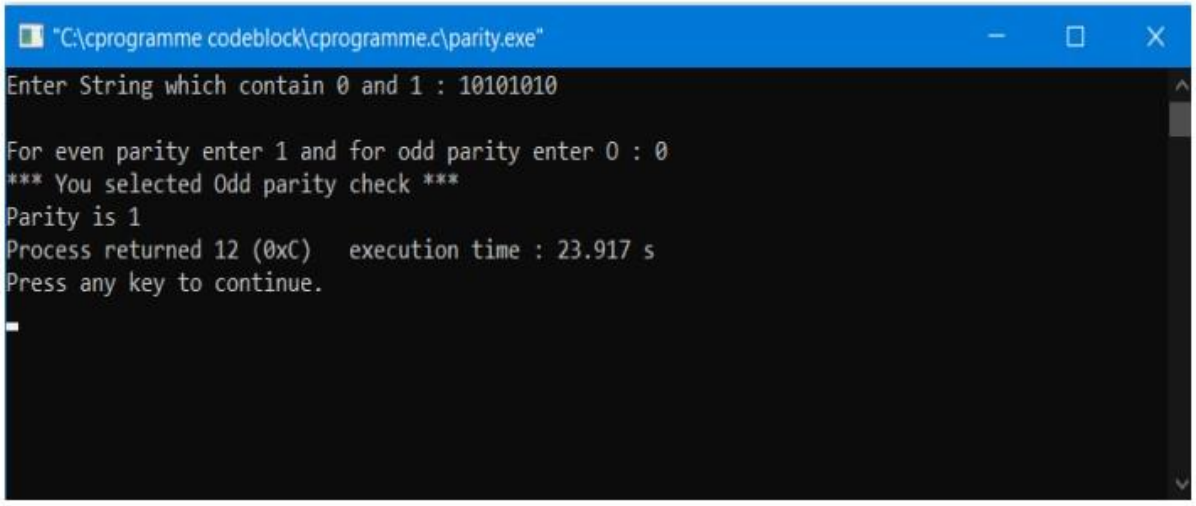
- This technique can not detect an even number of bit errors (two, four, six, and so on).
- If an even number of bits flips during transmission, then the receiver can not catch the error.

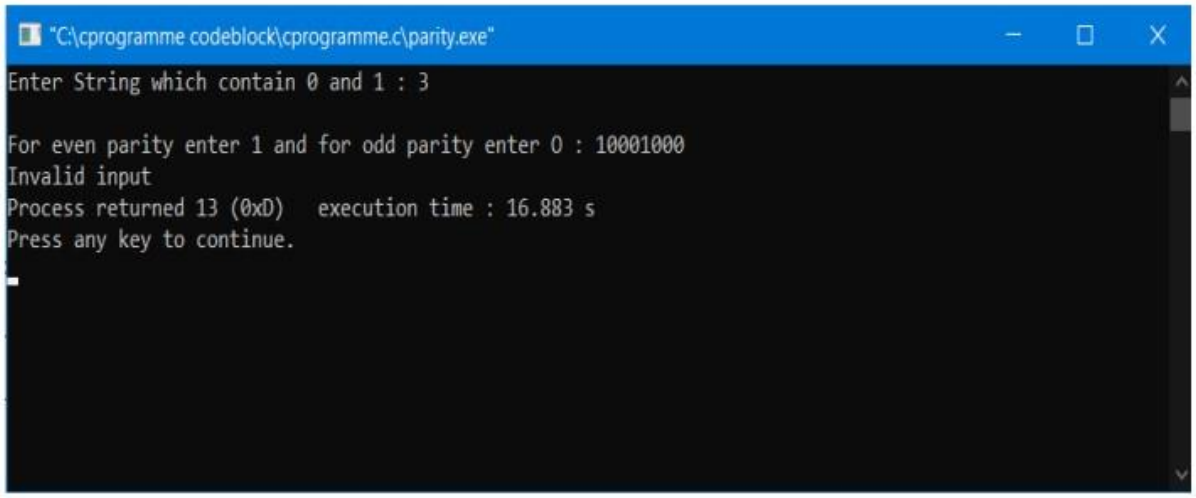
Code→

```
#include

<stdio.h> void

main() {

int num,r,flag=0,p; printf("Enter String which

contain 0 and 1 : "); scanf("%d",&num);

printf("\nFor even parity enter 1 and for odd parity enter 0 :

"); scanf("%d",&p); while(num!=0)

{

r=num%10;

if(r==1) flag+=1;

num=num/10;
} if(p==1) { printf("*** You selected Even parity

check ***"); if(flag%2==0)

printf("\nParity is 0");

else printf("\nParity

is 1");

} else if(p==0) { printf("*** You selected Odd

parity check ***"); if(flag%2==0)

printf("\nParity is 1");

else printf("\nParity

is 0");
```

```
}

else      printf("Invalid

input");

}
```

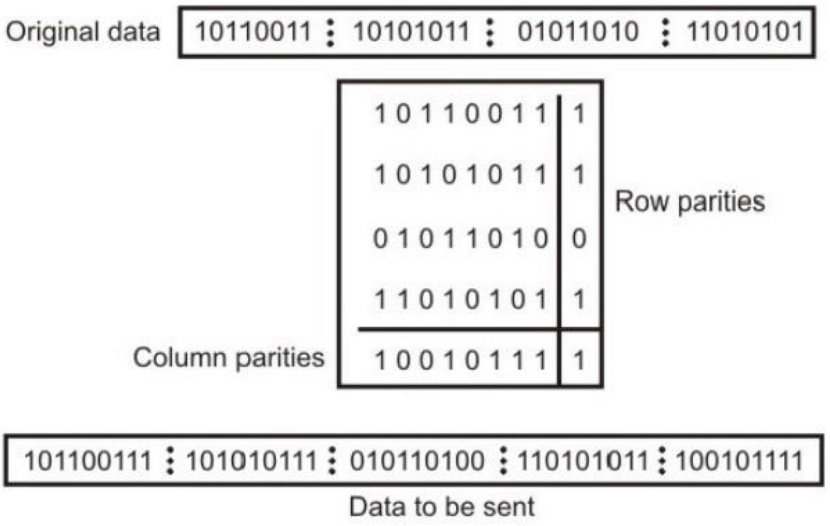
Output →





TWO DIMENSION PARITY CHECK

Performance can be improved by using two-dimensional parity check, which organizes the block of bits in the form of a table. Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns then both are sent along with the data. At the receiving end, these are compared with the parity bits calculated on the received data.



Two-dimension Parity Checking

Two- Dimension Parity Checking increases the likelihood of detecting burst errors. As we have shown in Fig. that a 2-D Parity check of n bits can detect a burst error of n bits. A burst error of more than n bits is also detected by 2-D Parity check with a high-probability. There is, however, one pattern of error that remains elusive. If two bits in one data unit are damaged and two bits in exactlythe same position in another data unit are also damaged, the 2-D Parity check checker will not detect an error.

For example, if two data units: are 11001100 and 10101100. The first and second last bits in each of them are changed, making the data units 01001110 and 00101110, the error cannot be detected by 2-D Parity check.

Code →

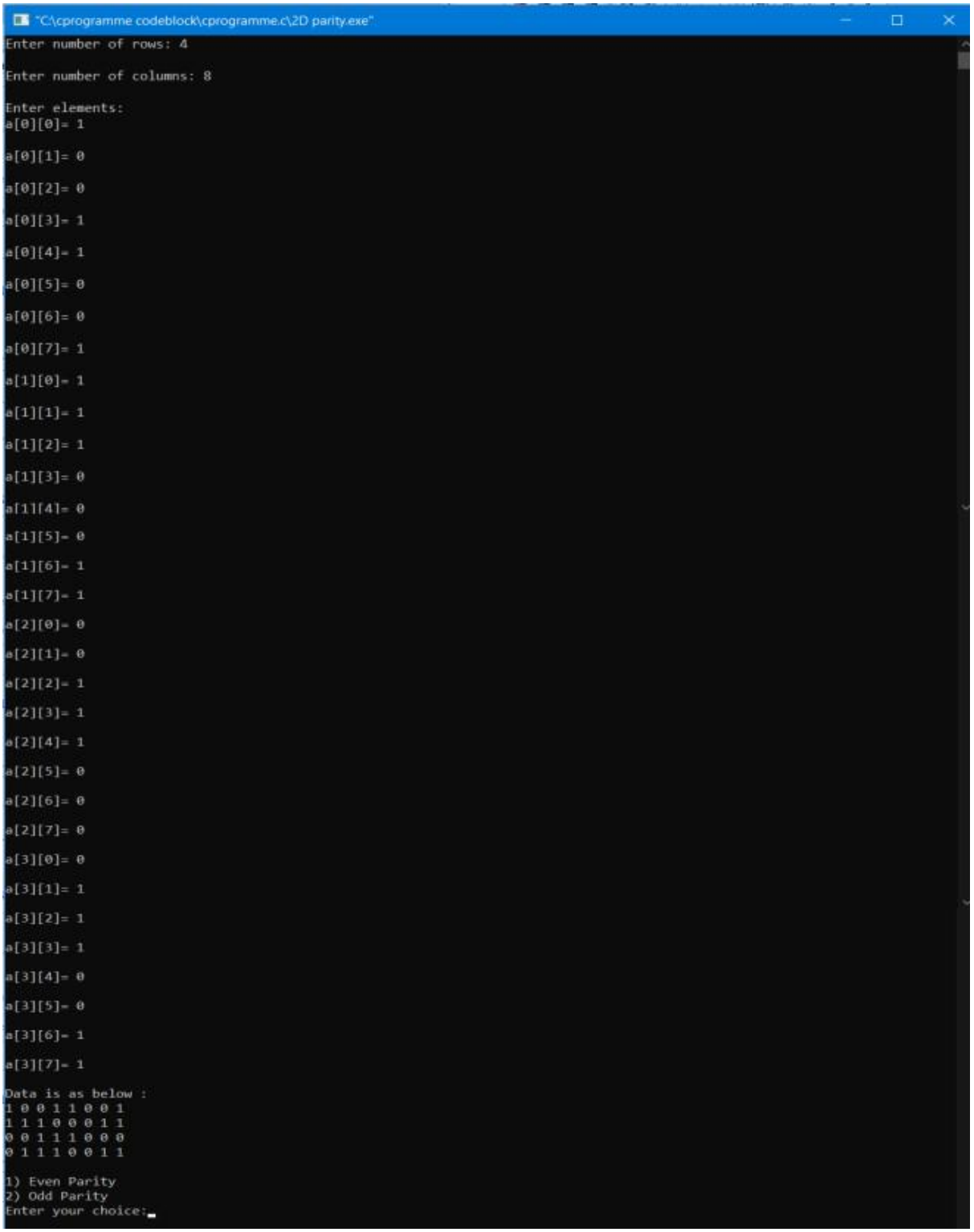
```
#include
<stdio.h> void
main() {
int
a[100][100],m,n,p,rflag=0,cflag=0,i,j;
printf("Enter number of rows: "); s
canf("%d",&m); printf("\nEnter number
```

```
of columns: "); scanf("%d",&n);
printf("\nEnter elements: ");
for(i=0;i<m;i++)
{ for(j=0;j<n;j++)
{
printf("\na[%d][%d]= ",i,j); scanf("%d",&a[i][j]);
}
}
printf("\nData is as below :\n");
for(i=0;i<m;i++)
{ for(j=0;j<n;j++)
{
printf("%d ",a[i][j]);
}
printf("\n");
} printf("\n1) Even Parity \n2) Odd Parity\nEnter your
choice:"); scanf("%d", &p); if (p == 1)
{
for ( i = 0; i < m; i++)
{ rflag=0; for (j = 0;
j < n; j++)
{
if (a[i][j] == 1)
{
rflag++; }
}
if (rflag % 2 == 0)
{
}
else
{
}
Computer Network (3150710)
}
a[i][j] = 0; a[i][j] = 1; for
(j = 0; j < (n+1); j++)
{ cflag=0; for (i = 0; i
< m; i++)
```

```
{
if (a[i][j] == 1)
{
cflag++;
}
}
if (cflag % 2 == 0)
{
a[i][j] = 0;
}
else
{
a[i][j] = 1;
}
}
}
else
{
for (i = 0; i < m; i++)
{
rflag=0; for (j = 0; j
< n; j++)
{
if (a[i][j] == 1)
{
rflag++;
}
}
if (rflag % 2 == 0)
{
a[i][j] = 1;
}
else
{
a[i][j] = 0;
}
}
for ( j = 0; j < (n+1); j++)
```

```
{ cflag=0; for (i = 0; i
< m; i++)
Computer Network (3150710)
{
if (a[i][j] == 1)
{
cflag++;
}
}
if (cflag % 2 == 0)
{
a[i][j] = 1;
}
else
{
a[i][j] = 0;
}
}
}
printf("\nData after parity is as below
:\n"); for(i=0;i<(m+1);i++) {
for(j=0;j<(n+1);j++)
{
printf("%d ",a[i][j]);
}
printf("\n");
}
}
```

Output →


"C:\cprogramme codeblock\cprogramme.c\2D parity.exe"
Enter number of rows: 4
Enter number of columns: 8
Enter elements:
a[0][0]= 1
a[0][1]= 0
a[0][2]= 0
a[0][3]= 1
a[0][4]= 1
a[0][5]= 0
a[0][6]= 0
a[0][7]= 1
a[1][0]= 1
a[1][1]= 1
a[1][2]= 1
a[1][3]= 0
a[1][4]= 0
a[1][5]= 0
a[1][6]= 1
a[1][7]= 1
a[2][0]= 0
a[2][1]= 0
a[2][2]= 1
a[2][3]= 1
a[2][4]= 1
a[2][5]= 0
a[2][6]= 0
a[2][7]= 0
a[3][0]= 0
a[3][1]= 1
a[3][2]= 1
a[3][3]= 1
a[3][4]= 0
a[3][5]= 0
a[3][6]= 1
a[3][7]= 1
Data is as below :
1 0 0 1 1 0 0 1
1 1 1 0 0 0 1 1
0 0 1 1 1 0 0 0
0 1 1 1 0 0 1 1
1) Even Parity
2) Odd Parity
Enter your choice:

```
1) Even Parity
2) Odd Parity
Enter your choice:1

Data after parity is as below :
1 0 0 1 1 0 0 1 0
1 1 1 0 0 0 1 1 0
0 0 1 1 1 0 0 0 0
0 1 1 1 0 0 1 1 0
0 0 1 1 0 0 0 1 0

Process returned 4 (0x4)   execution time : 78.075 s
Press any key to continue.
```

Sign:- _____ Date :- _____

Practical 6

Aim →

Implement error detection methods using CRC.

Cyclic Redundancy Check →

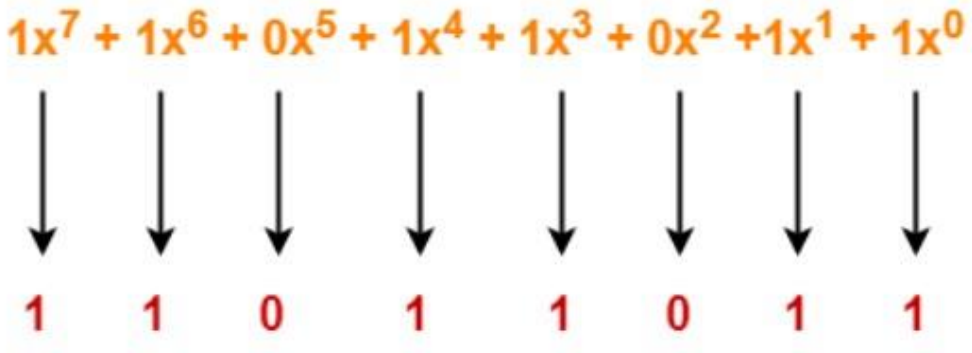
- CRC generator is an algebraic polynomial represented as a bit pattern.
- Bit pattern is obtained from the CRC generator using the following rule-

The power of each term gives the position of the bit and the coefficient gives the value of the bit.

Example-

Consider the CRC generator is $x^7 + x^6 + x^4 + x^3 + x + 1$.

The corresponding binary pattern is obtained as-



Thus, for the given CRC generator, the corresponding binary pattern is 11011011.

Properties Of CRC Generator-

The algebraic polynomial chosen as a CRC generator should have at least

the following properties**Rule-01:**

- It should not be divisible by x.
- This condition guarantees that all the burst errors of length equal to the length of the polynomial are detected.

Rule-02:

- It should be divisible by x+1.
- This condition guarantees that all the burst errors affecting an odd number of bits are detected.

Important Notes-

If the CRC generator is chosen according to the above rules, then-

- CRC can detect all single-bit errors
- CRC can detect all double-bit errors provided the divisor contains at least three logic 1's.
- CRC can detect any odd number of errors provided the divisor is a factor of x+1.
- CRC can detect all burst errors of length less than the degree of the polynomial.
- CRC can detect most of the larger burst errors with a high probability.

Steps Involved-

Error detection using the CRC technique involves the following steps-

Step-01: Calculation Of CRC At Sender Side-

At the sender side,

- A string of n 0's is appended to the data unit to be transmitted.
- Here, n is one less than the number of bits in the CRC generator.
- Binary division is performed of the resultant string with the CRC generator.
- After division, the remainder so obtained is called as **CRC**.

- It may be noted that CRC also consists of n bits.

Step-02: Appending CRC To Data Unit-

At the sender side,

- The CRC is obtained after the binary division.
- The string of n 0's appended to the data unit earlier is replaced by the CRC remainder.

Step-03: Transmission To Receiver-

- The newly formed code word (Original data + CRC) is transmitted to the receiver.

Step-04: Checking at Receiver Side-

At the receiver side,

- The transmitted code word is received.
- The received code word is divided with the same CRC generator.
- On division, the remainder so obtained is checked.

The following two cases are possible-

Case-01: Remainder = 0

If the remainder is zero,

- Receiver assumes that no error occurred in the data during the transmission.
- Receiver accepts the data.

Case-02: Remainder ≠ 0

If the remainder is non-zero,

- Receiver assumes that some error occurred in the data during the transmission.
- Receiver rejects the data and asks the sender for retransmission.

PRACTICE PROBLEMS BASED ON CYCLIC REDUNDANCY CHECK (CRC)-

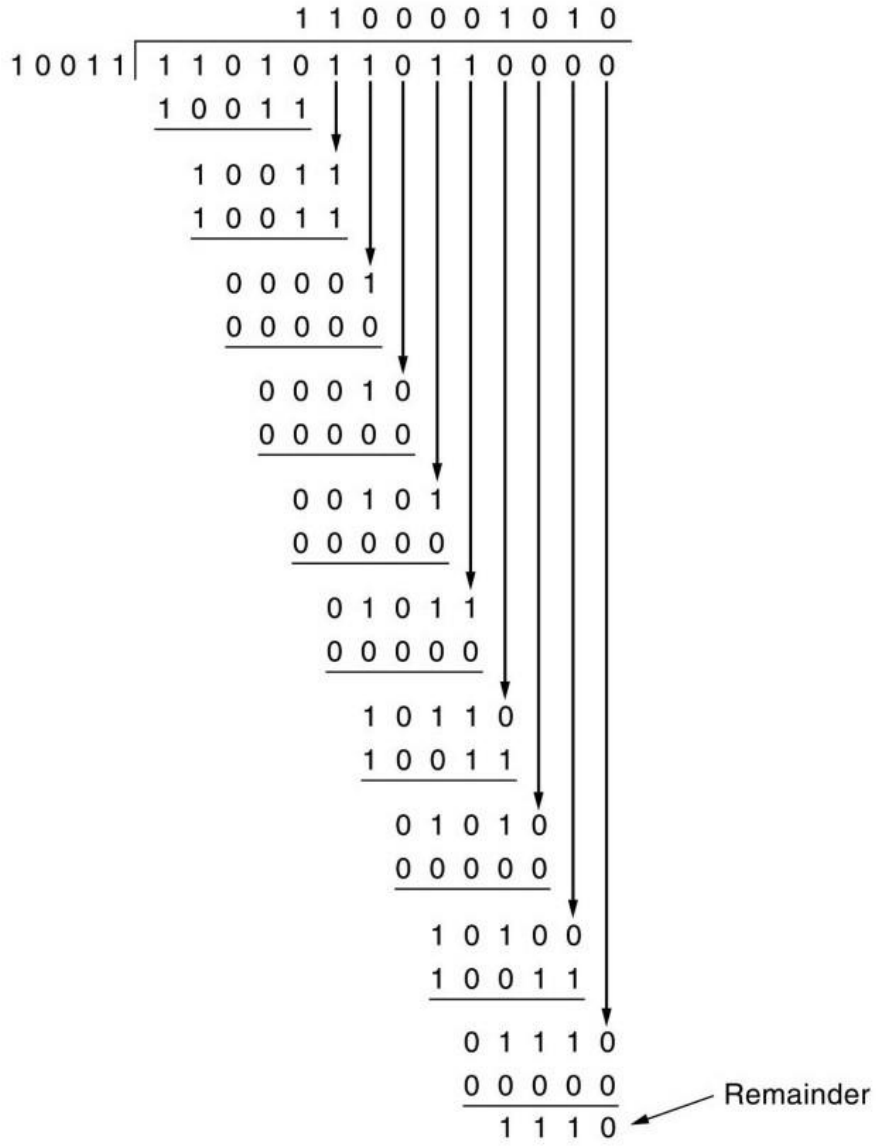
Problem-01:

A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x^4+x+1. What is the actual bit string transmitted?

Solution-

- The generator polynomial $G(x) = x^4 + x + 1$ is encoded as 10011.
- Clearly, the generator polynomial consists of 5 bits.
- So, a string of 4 zeroes is appended to the bit stream to be transmitted.
- The resulting bit stream is 11010110110000.

Now, the binary division is performed as-



Code →

```
#include <stdio.h>
#include <string.h> void
main()
{ int i,j,keylen,msglen; char input[100],
key[30],temp[30],quot[100],rem[30],key1[30]; printf("Enter
Frame: "); gets(input); printf("Enter Generator: "); gets(key);
keylen=strlen(key); sglen=strlen(input); strcpy(key1,key); for
(i=0;i<keylen-1;i++)
{
input[msglen+i]='0';
```

```
} for (i=0;i<keylen;i++)
temp[i]=input[i]; for (i=0;i<msglen;i++)
{
quot[i]=temp[0]; if(quot[i]=='0')
for (j=0;j<keylen;j++) key[j]='0';
Computer Network (3150710) else
for (j=0;j<keylen;j++) key[j]=key1[j];
for (j=keylen-1;j>0;j--)
{
if(temp[j]==key[j])
rem[j-1]='0'; else
} rem[j-1]='1'; rem[keylen-
1]=input[i+keylen];
strcpy(temp,rem);
} strcpy(rem,temp); printf("\nQuotient is "); for
(i=0;i<msglen;i++) printf("%c",quot[i]);
printf("\nRemainder is "); for (i=0;i<keylen-
1;i++) printf("%c",rem[i]);
}
```

Output:

```
Enter Frame: 1101011111
Enter Generator: 10011
Quotient is 11000011
```

Sign:- _____ Date :- _____

Practical 7

Aim →

Implement error correction method using Hamming Code.

Hamming Code

Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction.

In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs

recalculations to detect errors and find the bit position that has an error.

Encoding a message by Hamming Code

The procedure used by the sender to encode the message encompasses the following steps −

- Step 1 − Calculation of the number of redundant bits.
- Step 2 − Positioning the redundant bits.
- Step 3 − Calculate the values of each redundant bit.

Once the redundant bits are embedded within the message, this is sent to the user.

Step 1 − Calculation of the number of redundant bits.

If the message contains *mm*number of data bits, *rr*number of redundant bits are added to it so that *mr* is able to indicate at least (*m* + *r*+ 1) different states. Here, (*m* + *r*) indicates the location of an error in each of (m + r) bit positions and one additional state indicates no error. Since *rr* bits can indicate 2rr states, 2rr must be at least equal to (*m* + *r* + 1). Thus the following equation should hold $2r \geq m+r+1$

Step 2 – Positioning the redundant bits.

The *r* redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc. They are referred in the rest of this text as *r1* (at position 1), *r2* (at position 2), *r3* (at position 4), *r4* (at position 8) and so on.

Step 3 – Calculating the values of each redundant bit.

The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

- Even Parity − Here the total number of bits in the message is made even.
- Odd Parity − Here the total number of bits in the message is made odd.

Each redundant bit, ri, is calculated as the parity, generally even parity, based upon its bit position. It covers all bit positions whose binary representation includes a 1 in the ith position except the position of ri. Thus −

- r1 is the parity bit for all data bits in positions whose binary representation includes a 1 in the least significant position excluding 1 (3, 5, 7, 9, 11 and so on)

- r2 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 2 from right except 2 (3,
  6, 7, 10, 11 and so on)
- r3 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 3 from right except 4 (5-7, 12-15, 20-23 and so on)

Decoding a message in Hamming Code

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps for recalculation are −

- Step 1 − Calculation of the number of redundant bits.
- Step 2 − Positioning the redundant bits. ● Step 3 − Parity checking.
- Step 4 − Error detection and correction

Step 1 − Calculation of the number of redundant bits

Using the same formula as in encoding, the number of redundant bits are ascertained.

$2r \geq m + r + 1$ where *m* is the number of data bits and *r* is the number of redundant bits.

Step 2 − Positioning the redundant bits

The *r* redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc.

Step 3 – Parity checking

Parity bits are calculated based upon the data bits and the redundant bits using

the same rule as during generation of c1,c2 ,c3 ,c4 etc. Thus
c1 = parity(1, 3, 5, 7, 9, 11 and so on)

c2 = parity(2, 3, 6, 7, 10, 11 and so

on) c3 = parity(4-7, 12-15, 20-23 and

so on)

Step 4 – Error detection and correction

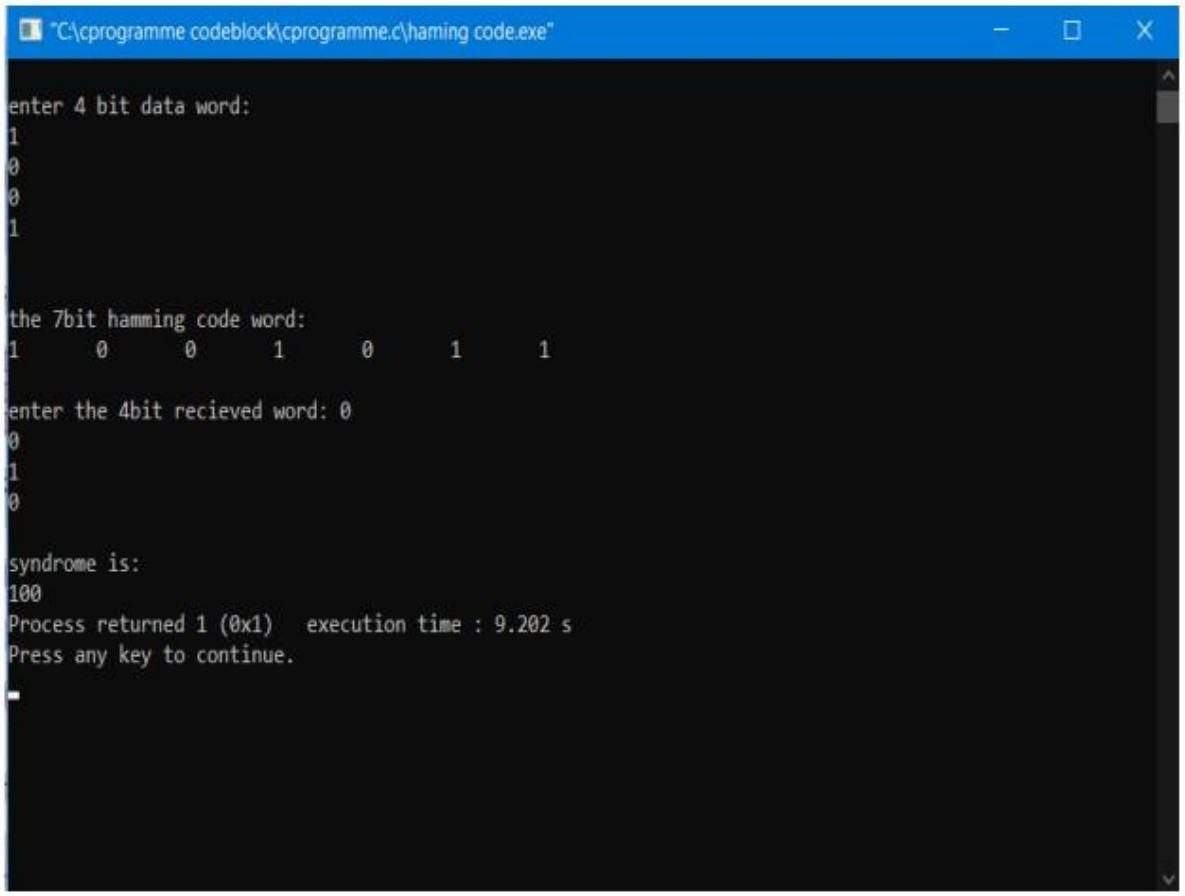
The decimal equivalent of the parity bits binary values is calculated. If it is 0, there is no error. Otherwise, the decimal value gives the bit position which has an error. For example, if c1c2c3c4 = 1001, it implies that the data bit at position 9, the decimal equivalent of 1001, has an error. The bit is flipped to get the correct message.

Code →

```
#include<stdio.h>
int main()
{
int a[4],b[4],r[3], s[3],i,q[3];
printf("\nenter 4 bit data word:\n");
for(i=3;i>=0;i--)
{ scanf("%d",&a[i]);
} r[0]=(a[2]+a[1]+a[0])%2;
r[1]=(a[2]+a[1]+a[3])%2;
r[2]=(a[0]+a[1]+a[3])%2; printf("\n\nthe
7bit hamming code word: \n");
for(i=3;i>=0;i--)
{
printf("%d\t",a[i]);
}
```

```
for(i=2;i>=0;i--)
{
printf("%d\t",r[i]);
}
printf("\n"); printf("\nenter the 4bit
recieved word: "); for(i=3;i>=0;i--)
scanf("%d",&b[i]); //calculating
syndrome bits s[0]=
(b[2]+b[1]+b[0]+r[0])%2; s[1]=
(b[3]+b[2]+b[1]+r[1])%2; s[2]=
(b[0]+b[1]+b[3]+r[2])%2;
printf("\nsyndrome is: \n"); for(i=2;
i>=0;i--)
{
printf("%d",s[i]);
} if((s[2]==0) && (s[1]==0) && (s[0]==0)) printf("\n RECIEVED WORD IS
ERROR FREE\n"); if((s[2]==1) && (s[1]==0) && (s[0]==1))
printf("\nrecieved word is error with error in 1bit(b0) position from
right\n"); if((s[2]==1) && (s[1]==1) && (s[0]==1)) printf("\nrecieved word
is error with error in 2bit(b1) position from right\n"); if((s[2]==0) &&
(s[1]==1) && (s[0]==1)) printf("\nrecieved word is error with error in
3bit(b2) position from right\n") if((s[2]==1) && (s[1]==1) && (s[0]==0))
printf("\nrecieved word is error with error in 4bit(b3) position from
right\n"); return(1);
}//End of Hamming code program
```

Output →



Sign:- _____ Date :- _____

PRACTICAL: 8

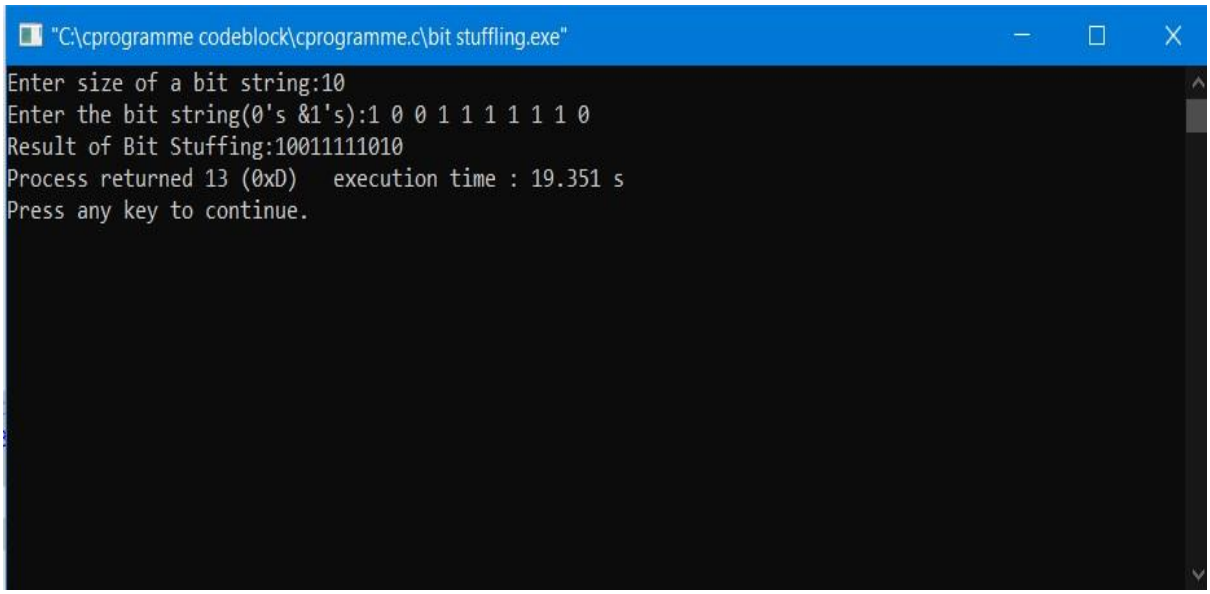
AIM: Implement Bit Stuffing.

Code:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

void main()
{
 int x[50], y[50], n;
 int i, j, k, count = 1;

 printf("Enter size of a bit string:");
 scanf("%d", &n);

 printf("Enter the bit string(0's &1's):");
 for (i = 0; i < n; i++)
 {
  scanf("%d", &x[i]);
 }

 i = 0;
 j = 0;

 while (i < n)
 {
  if (x[i] == 1)
  {
   y[j] = x[i];

   //count is less than 5 as 0 is inserted after every
   //5 consecutive 1's
    for (k = i + 1; x[k] == 1 && k < n && count < 5; k++)
    {
     j++;
     y[j] = x[k];
     count++;

     if (count == 5)
     {
      j++;
      y[j] = 0;
     }

     i = k;
```

```
    }
   }
   else
   {
    y[j] = x[i];
   }

   i++;
   j++;
  }

 //Displaying final result
  printf("Result of Bit Stuffing:");
  for (i = 0; i < j; i++)
  {
   printf("%d", y[i]);
  }

  getch();
}
```

Output:



Sign:- _____ **Date :-** _____

PRACTICAL: 9

AIM: Implement the concept of VLAN using Packet Tracer Tool.

Software: Cisco Packet Tracer

Theory:

Virtual LAN (VLAN) is a concept in which we can divide the devices logically on layer 2 (data link layer). Generally, layer 3 devices divide the broadcast domain but the broadcast domain can be divided by switches using the concept of VLAN.

A broadcast domain is a network segment in which if a device broadcasts a packet then all the devices in the same broadcast domain will receive it. The devices in the same broadcast domain will receive all the broadcast packets but it is limited to switches only as routers don't forward out the broadcast packet. To forward out the packets to different VLAN (from one VLAN to another) or broadcast domains, inter Vlan routing is needed. Through VLAN, differentsmall-size sub-networks are created which are comparatively easy to handle.

VLAN ranges:

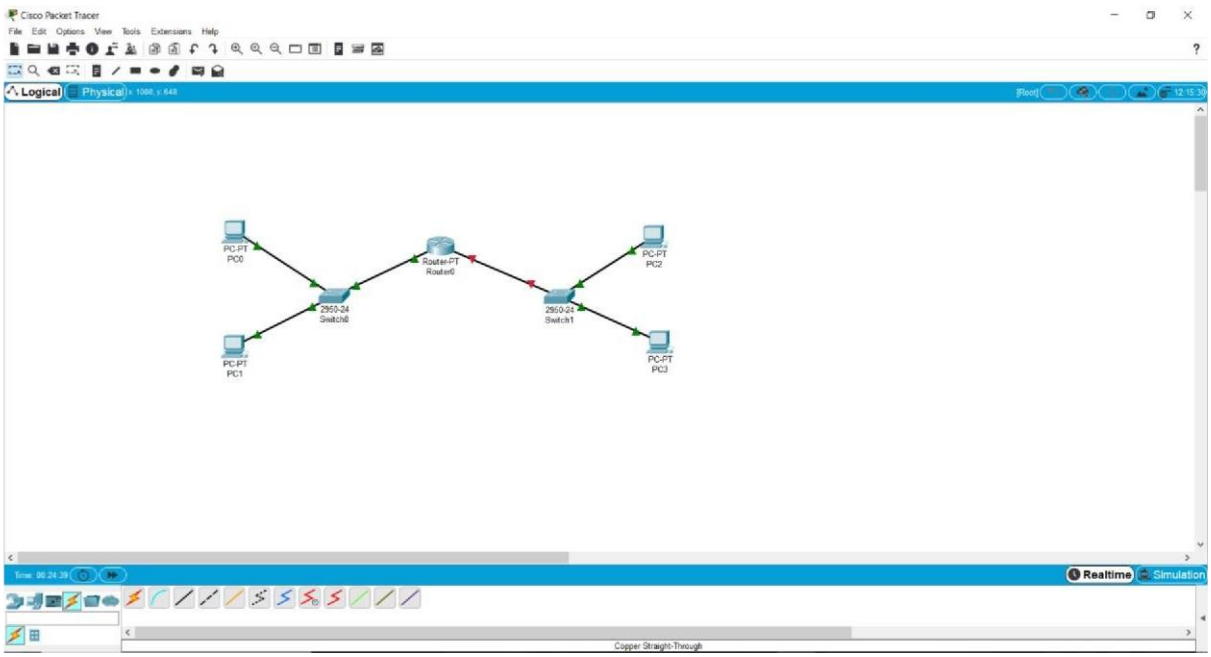
- **VLAN 0, 4095:** These are reserved VLAN which cannot be seenor used.
- **VLAN 1:** It is the default VLAN of switches. By default, all switch ports are in VLAN. This VLAN can't be deleted or edited but can be used.
- **VLAN 2-1001:** This is a normal VLAN range. We can create, edit and delete these VLAN.
- **VLAN 1002-1005:** These are CISCO defaults for FDDI and token rings. These VLAN can't be deleted.

Creating VLAN steps:

- Select 4 'end systems' from the Left hand side bottom of the screen visible to you and place these end systems into the area above.
- Select 2 switches(2950-24 configuration) from the pan of the 'Network Devices' and place them into the configuration area.
- Select one 'PT-Router' from the pan of the 'Network Devices' and place it into the middle of the switches.
- From the pan of 'connections', select the copper straight-through cable and place it first onto the PC0 side and select FastEthernet0 and place another end of the cable onto the switch0 side and select FastEthernet0/1.
- Same way, do it for the other 3 end systems as:
- 1. PC1: FastEthernet0 to Switch0 : FastEthernet0/2
- 2. PC2: FastEthernet0 to Switch1 : FastEthernet0/1
- 3. PC3: FastEthernet0 to Switch1 : FastEthernet0/2
- From the pan of 'connections', select the copper straight-through cable and place it first onto the Switch0 side and select FastEthernet0/3 and place another end of the cable onto the Router0 side

and select FastEthernet0/0.

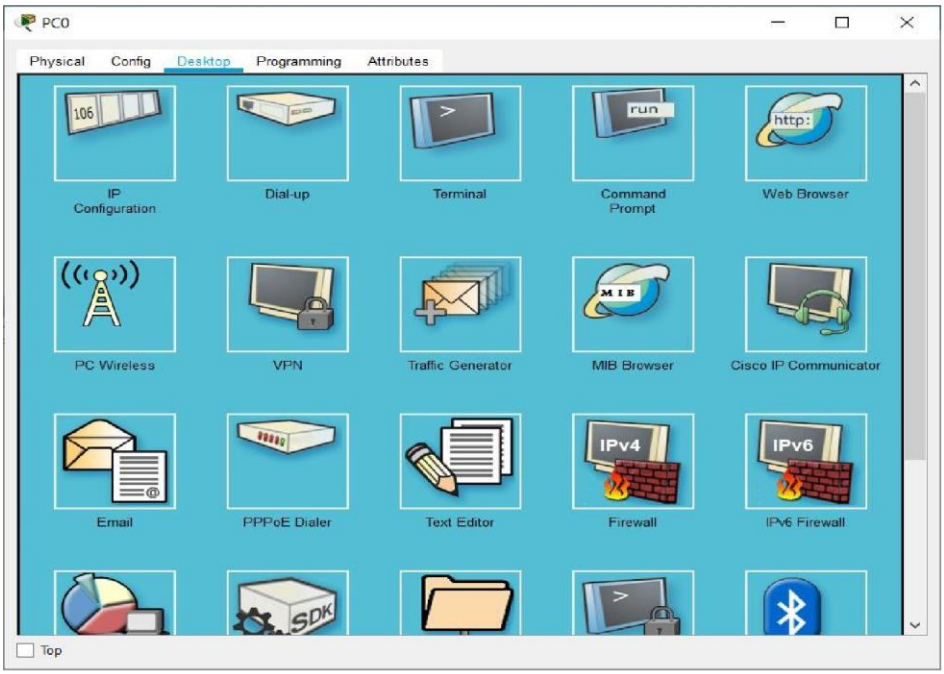
- Same way, do it for Switch1:
- 1. Switch1 : FastEthernet0/3 to Router0 : FastEthernet1/0
- The configuration willbe:



- Next, assign IP addresses as per the following range:

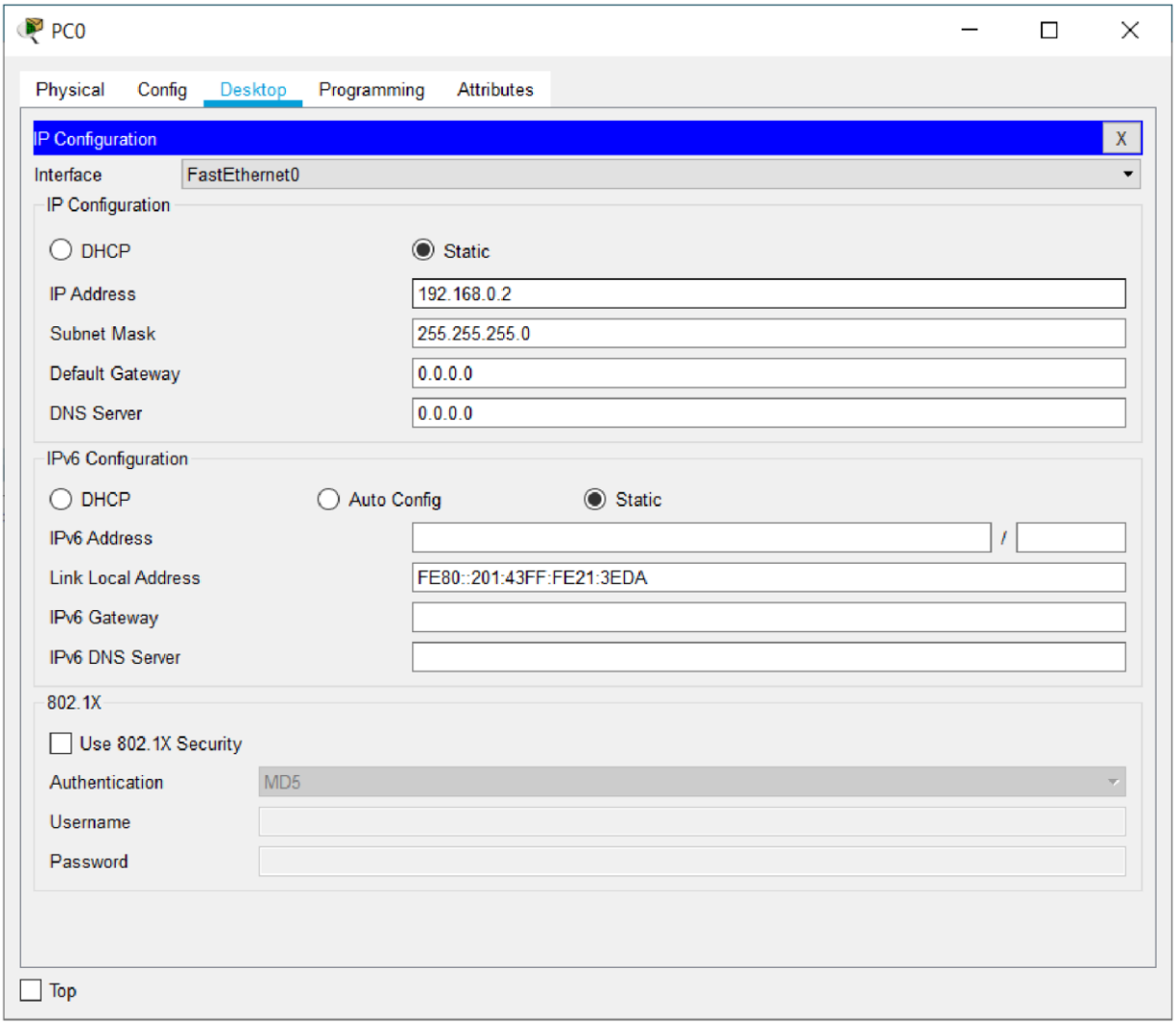
1. Leftside Subnetrange: 192.168.0.X(2End Systems & aswitch)
2. Right side Subnet range: 192.168.10.X(2 End Systems & a switch)

Now. Click on PC0 and the following window will pop up:



- Go to Desktop pan-> IP Configuration
- Assign IP Address & subnet mask and then close the pop-up window:

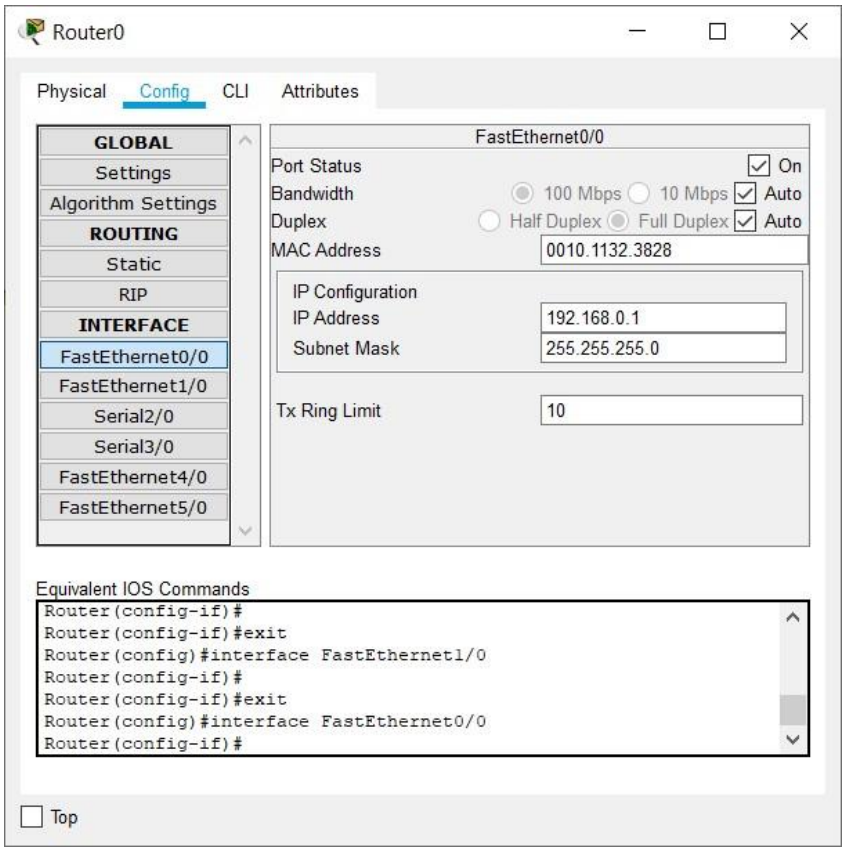
1. IP Address: 192.168.0.2
2. Subnet Mask: 255.255.255.0



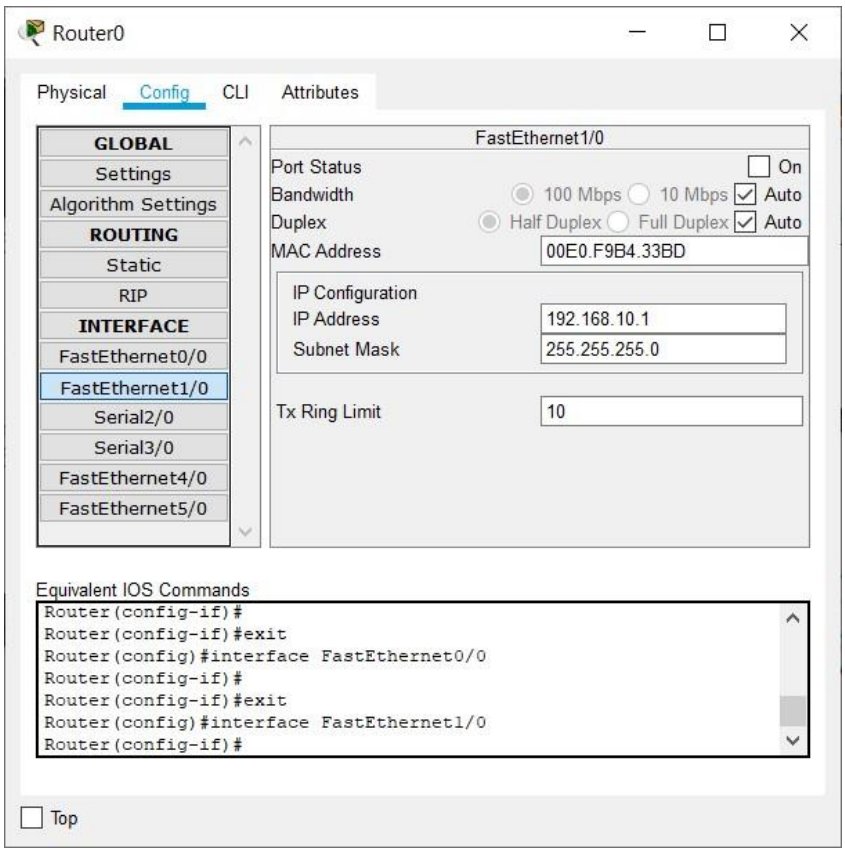
Same way, do it for PC1, PC2 & PC3.

End System	IP Address	Subnet Mask
PC1	192.168.0.3	255.255.255.0
PC2	192.168.10.2	255.255.255.0
PC3	192.168.10.3	255.255.255.0

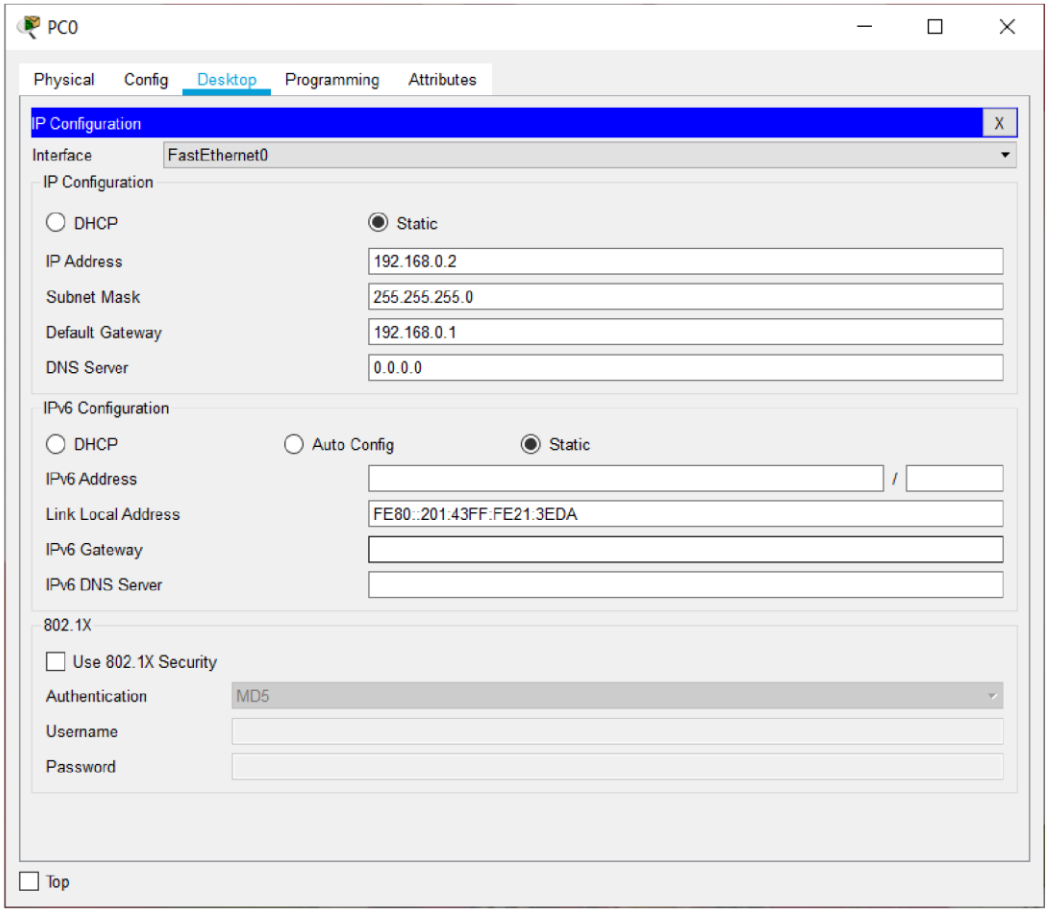
- Then, click on Router0 and go to config tab and then click on FastEthernet0/0.
- Update the values as per the following screenshot:
  1. Check the Port status to On.



- Same way, do it for FastEthernet1/0:



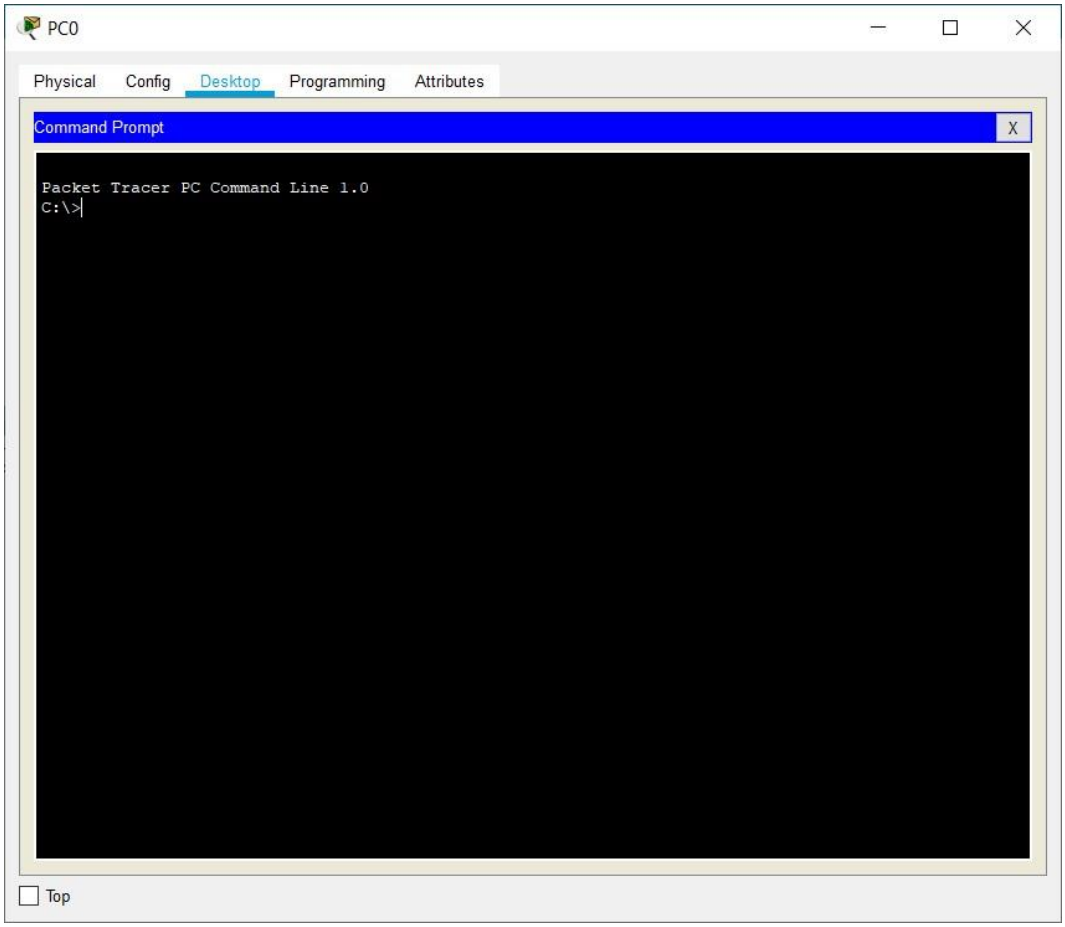
- Then, click on PC0 and go to Desktop-> IP Configuration & set the Default Gateway to: 192.168.0.1



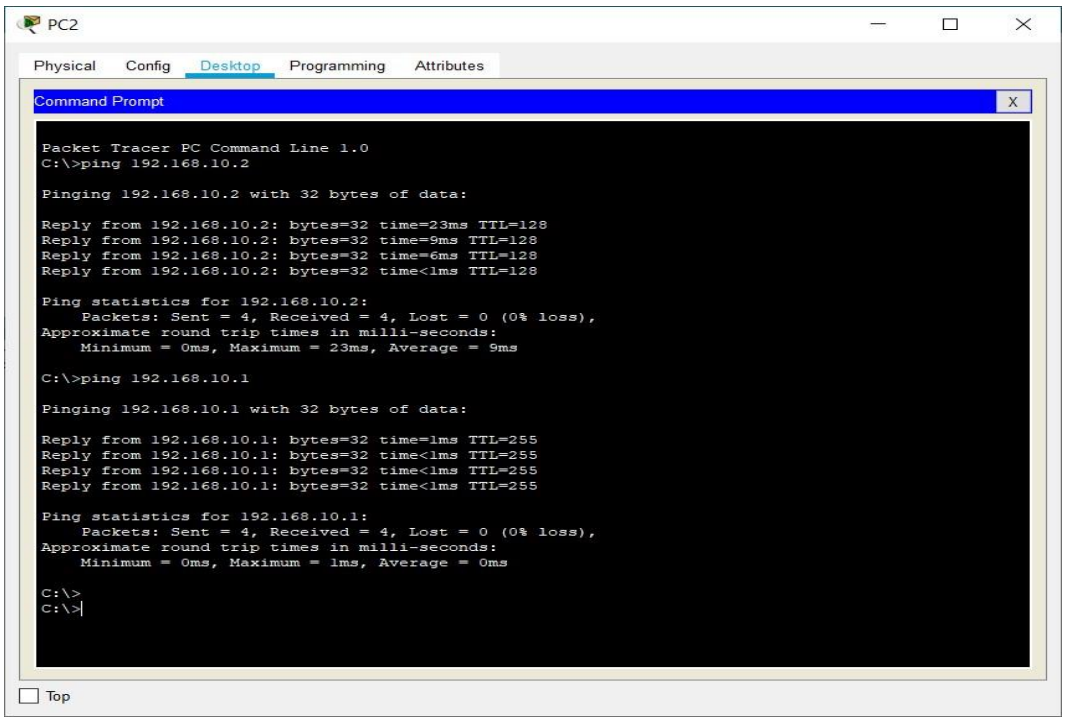
- Same way, do it for PC1, PC2 & PC3.

End System	Default Gateway
192.168.0.3	192.168.0.1
192.168.10.2	192.168.10.1
192.168.10.3	192.168.10.1

- Now, open any one end-system, e.g., PC1 and go to Desktop-> Command Prompt; the following window will pop up:



- Then, type: ping 192.168.10.2(PC2) at the command prompt just opened.
- Following is the output of the ping command, to confirm the proper VLAN setup.



- **Output:**

Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time=13ms TTL=255

Reply from 192.168.10.1: bytes=32 time<1ms TTL=255 Reply from 192.168.10.1: bytes=32 time<1ms TTL=255 Reply from 192.168.10.1: bytes=32 time<1ms TTL=255

Ping statistics for192.168.10.1:
Packets: Sent=4, Received=4, Lost=0(0% loss), Approximate round trip times in milli-seconds: Minimum=0ms, Maximum=13ms, Average=3ms

PRACTICAL: 10

AIM: Implement the concept of Link State(LS) Routing Algorithm using c/c++/Java.

Explanation:

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

The three keys to understand the Link State Routing algorithm:

- **Knowledge about the neighborhood:**

Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other
routers.

- **Flooding:**

Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.

- **Information sharing:**

A router sends the information to every other router only when the change occurs in the information.

Link State Routing has two phases:

Reliable Flooding:

o Initial state: Each node knows the cost of its neighbors.
o Final state: Each node knows the entire graph.

Route Calculation:

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.

- The Dijkstra's algorithm is an iterative, and it has the property that after k^{th} iteration of the algorithm, the least cost paths are well known for k destination nodes.

Some notations:

- c(i , j): Link cost from node i to node j. If i and j nodes are not directly linked, then c(i , j) = ∞.

- D(v): It defines the cost of the path from source code to destination v that has the least cost currently.

- P(v): It defines the previous node (neighbor of v) along with current least cost path from source to v.

- N: It is the total number of nodes available in the network.

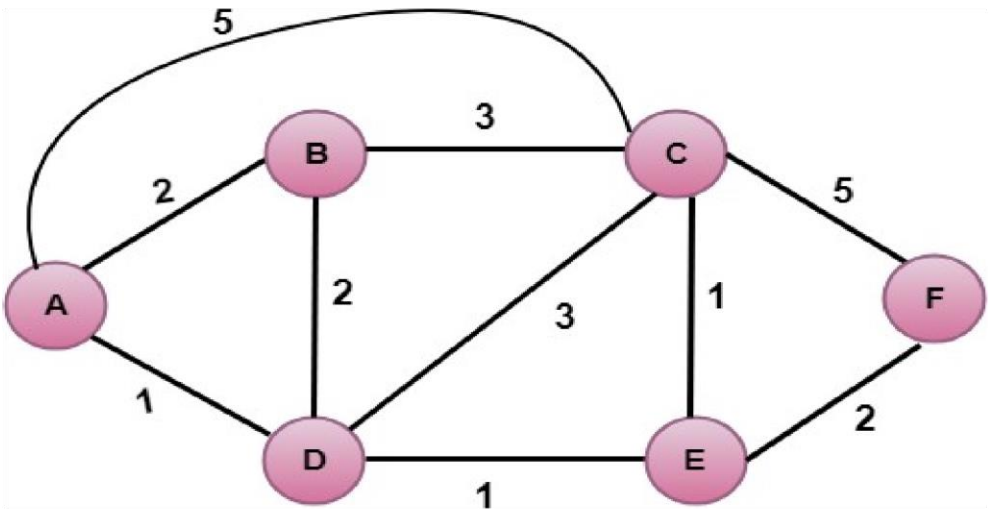
Algorithm:

Initialization
N = {A} // A is a root node.
for all nodes v
if v adjacent to A
then D(v) = c(A,v)
else D(v) = infinityloop
find w not in N such that D(w) is a minimum.
Add w to N
Update D(v) for all v adjacent to w and not in N:

D(v) = min(D(v) , D(w) + c(w,v)) -->Until all nodes in N

In the above algorithm, an initialization step is followed by the loop. The number of times the loop is executed is equal to the total number of nodes available in the network.

Let's understand through an example:



In the above figure, source vertex is A.

Step 1:

The first step is an initialization step. The currently known least cost path from A to its directly attached neighbors, B, C, D are 2,5,1 respectively. The cost from A to B is set to 2, from A to D is set to 1 and from A to C is set to 5. The cost from A to E and F are set to infinity as they are not directly linked to A.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞

Step 2:

In the above table, we observe that vertex D contains the least cost path in step 1. Therefore, it is added in N. Now, we need to determine a least-cost path through D vertex.

a) Calculating shortest path from A to B

v = B, w = D
D(B) = min(D(B) , D(D) + c(D,B))

= min(2, 1+2)
= min(2, 3)

The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

b) Calculating shortest path from A to C

v = C, w = D
D(C) = min(D(C) , D(D) + c(D,C))

= min(5, 1+3)
= min(5, 4)

The minimum value is 4. Therefore, the currently shortest path from A to C is 4.

c) Calculating shortest path from A to E

v = E, w = D
D(E) = min(D(E) , D(D) + c(D,E))

= min(∞, 1+1)
= min(∞, 2)

The minimum value is 2. Therefore, the currently shortest path from A to E is 2.

Note: The vertex D has no direct link to vertex E. Therefore, the value of D(F) is infinity.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞

Step 3:

In the above table, we observe that both E and B have the least cost path in step 2. Let's consider the E vertex. Now, we determine the least cost path of remaining vertices through E.

a) Calculating the shortest path from A to B.

v = B, w = E
D(B) = min(D(B) , D(E) + c(E,B))

= min(2 , 2+ ∞)
= min(2, ∞)

The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

b) Calculating the shortest path from A to C.

v = C, w = E
D(C) = min(D(C) , D(E) + c(E,C))

= min(4 , 2+1)
= min(4,3)

The minimum value is 3. Therefore, the currently shortest path from A to C is 3.

c) Calculating the shortest path from A to F.

v = F, w = E
D(F) = min(D(F) , D(E) + c(E,F))

= min(∞ , 2+2)
= min(∞ ,4)

The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E

Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E
4	ADEB		3,E			4,E
5	ADEBC					4,E

Final table:

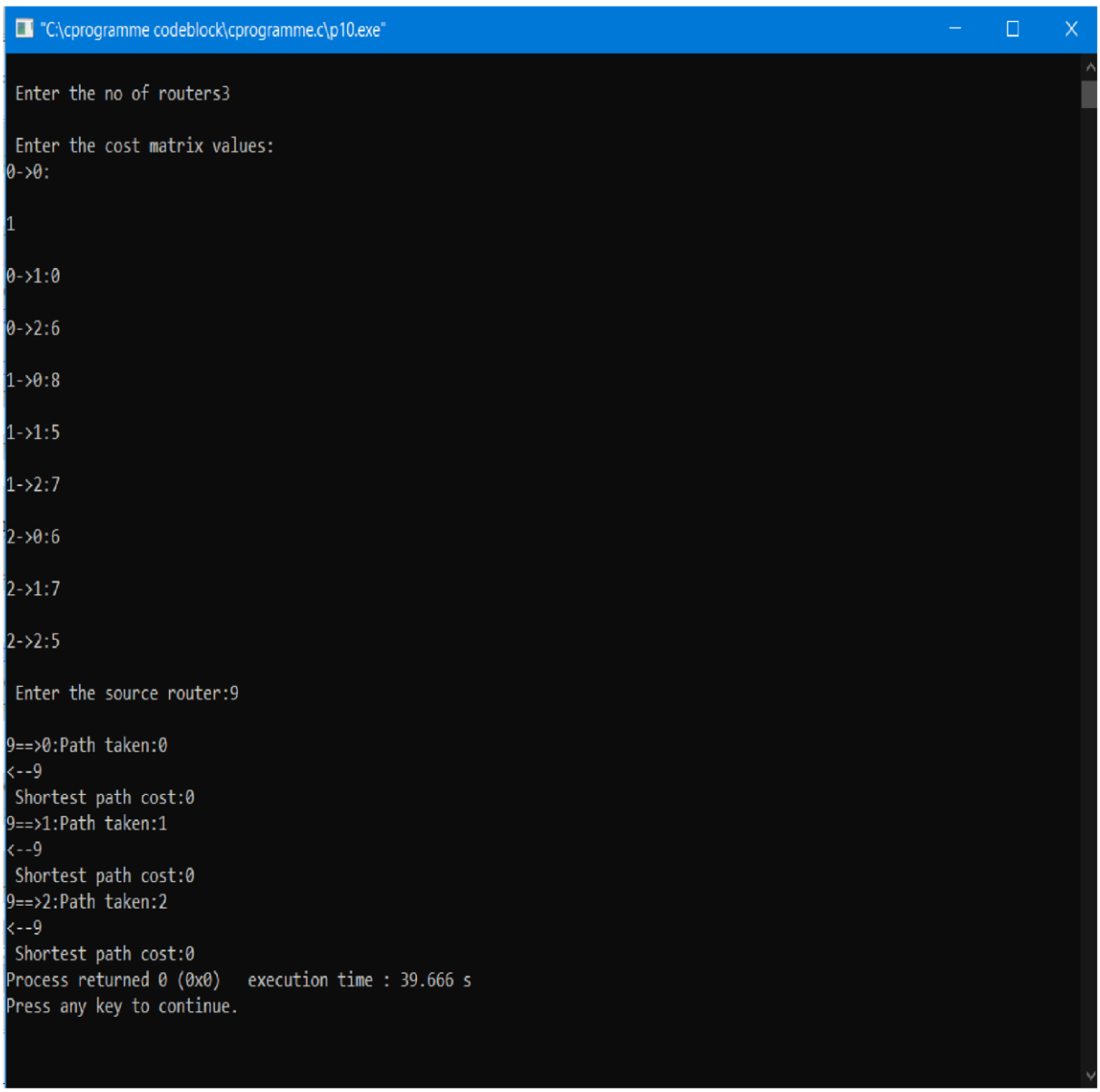
Step	N	D(B),P(B)	D(C),P(C)	D(D),P(D)	D(E),P(E)	D(F),P(F)
1	A	2,A	5,A	1,A	∞	∞
2	AD	2,A	4,D		2,D	∞
3	ADE	2,A	3,E			4,E
4	ADEB		3,E			4,E
5	ADEBC					4,E
6	ADEBCF					

Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
int count, src_router, i, j, k, w, v, min;
int cost_matrix[100][100], dist[100],
last[100]; int flag[100];
printf("\n Enter the no of routers");
scanf("%d", &count);
printf("\n Enter the cost matrix values:");
for (i = 0; i < count; i++)
{
for (j = 0; j < count; j++)
{
printf("\n%d->%d:", i, j);
scanf("%d", &cost_matrix[i][j]);
if (cost_matrix[i][j] < 0)
cost_matrix[i][j] = 1000;
}
}
printf("\n Enter the source router:");
scanf("%d", &src_router);
for (v = 0; v < count; v++)
{
flag[v] = 0;
last[v] = src_router;
dist[v] = cost_matrix[src_router][v];
}
flag[src_router] = 1;
for (i = 0; i < count; i++)
{
min = 1000;
for (w = 0; w < count; w++)
{
if (!flag[w])
if (dist[w] < min)
{
v = w;
min = dist[w];
}
}
flag[v] = 1;
for (w = 0; w < count; w++)
{
if (!flag[w])
if (min + cost_matrix[v][w] < dist[w])
{
dist[w] = min + cost_matrix[v][w]; last[w] = v;
```

```
}
}
}
for (i = 0; i < count; i++)
{
printf("\n%d==>%d:Path taken:%d", src_router, i, i); w = i;
while (w != src_router)
{
printf("\n<--%d", last[w]); w = last[w];
}
printf("\n Shortest path cost:%d", dist[i]);
}
return 0;
}
```

Output:



Sign:- _____ **Date :-** _____

PRACTICAL: 11

AIM: Implement the concept of Distance Vector(DV) Routing Algorithm using c/c++/Java.

Explanation:

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 –

- A router transmits its distance vector to each of its neighbors in a routing packet.
- Each router receives and saves the most recently received distance vector from each of its neighbors.
- 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.

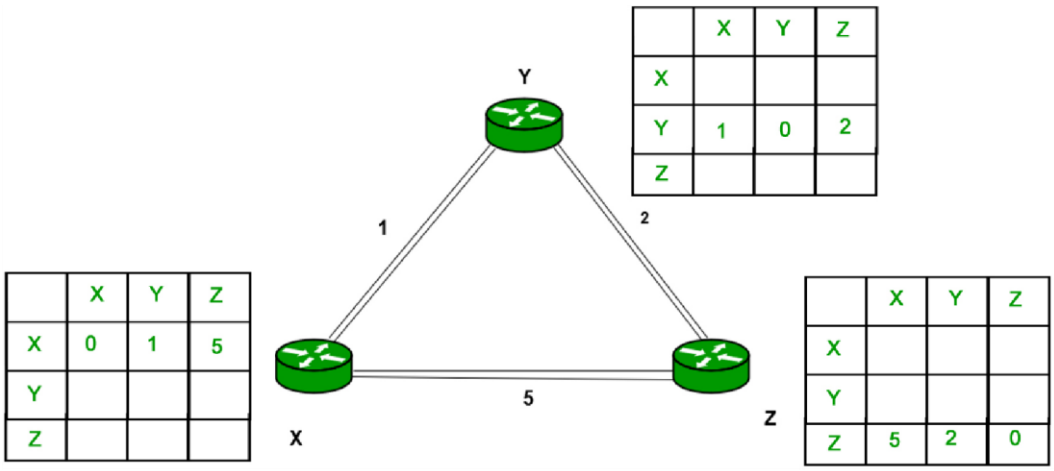
The DV calculation is based on minimizing the cost to each destination

- 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 $\in$ N] = Node x maintains distance vector
- Node x also maintains its neighbors' distance vectors
- For each neighbour v, x maintains Dv = [Dv(y): y $\in$ N]

Note:

- From time-to-time, each node sends its own distance vector estimate to neighbors.

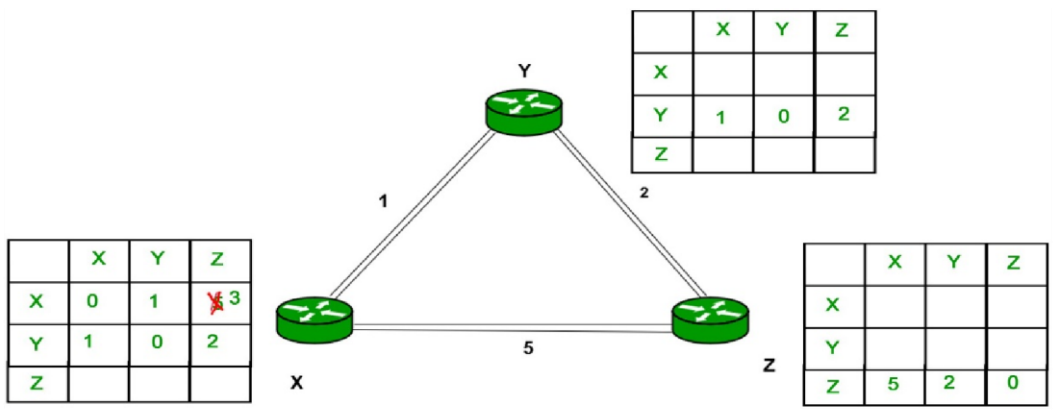
- When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using **Bellman - Ford** equation:
  Dx(y) = min{ C(x,v) + Dv(y), Dx(y) } for each node y ∈ N
  Example – Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.



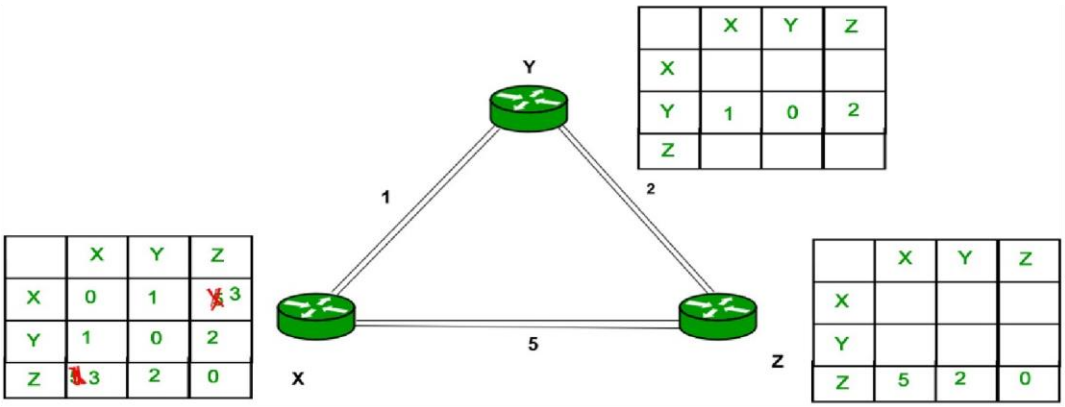
Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

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

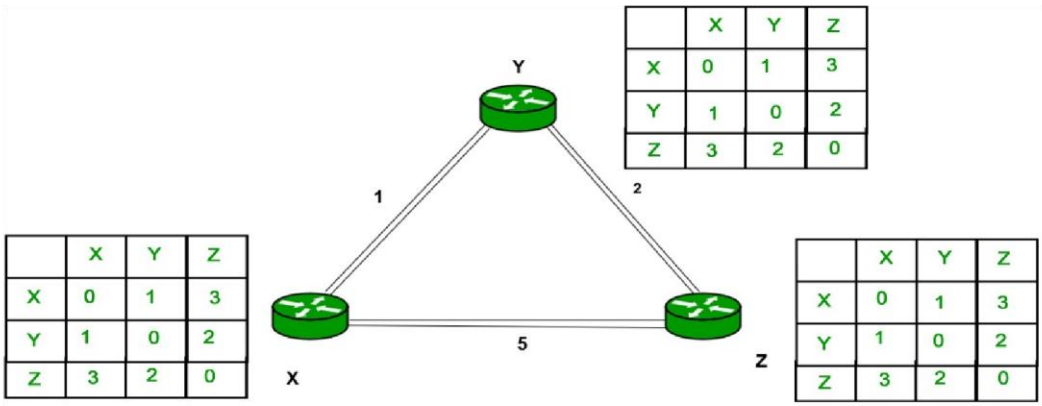
As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.



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.

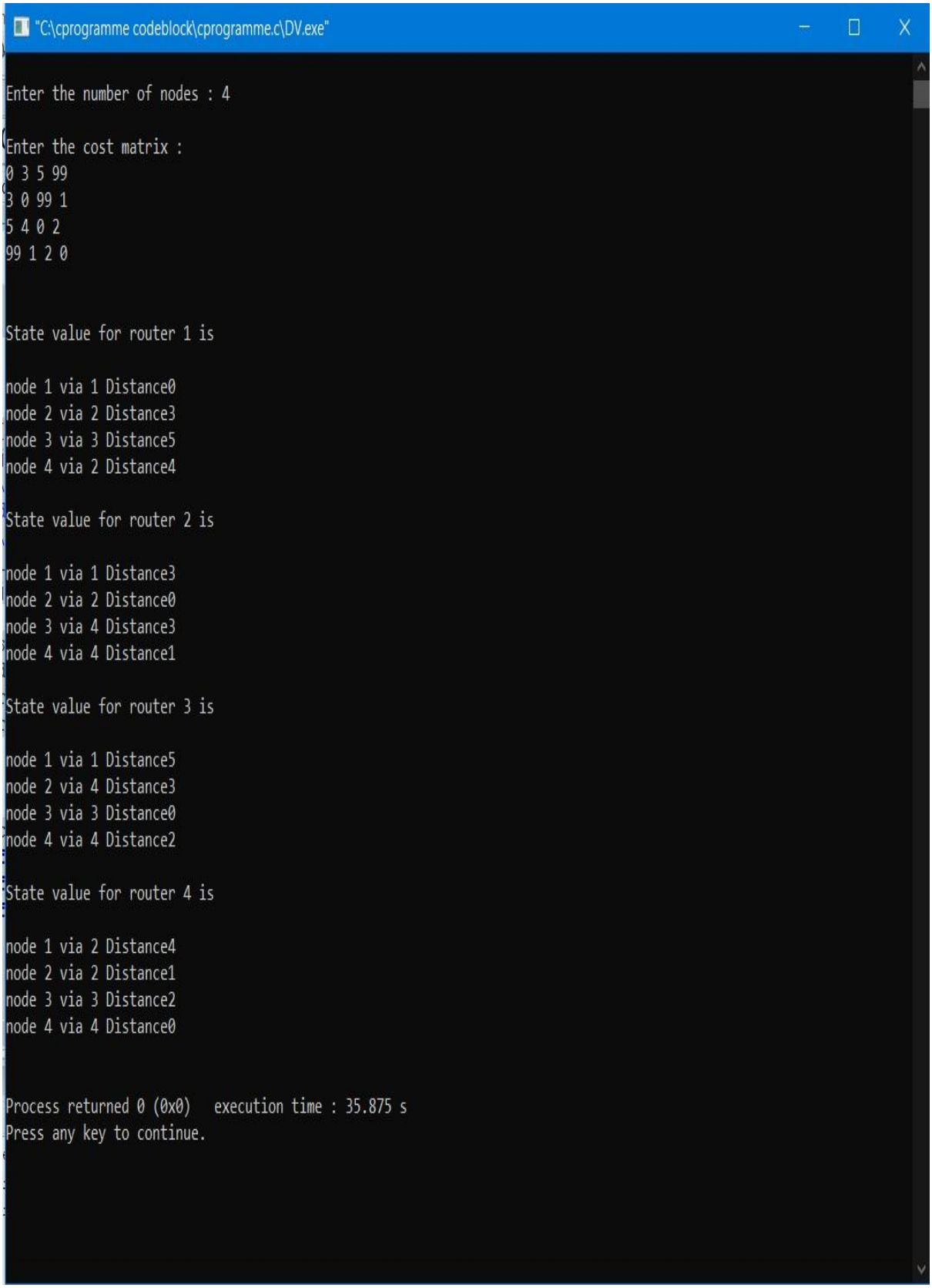
Disadvantages of Distance Vector routing –

- 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.

Code:

```
#include<stdio.h>
struct node
{
	unsigned dist[20];
	unsigned from[20];
}rt[10];
int main()
{
	int dmat[20][20];
	int n,i,j,k,count=0;
	printf("\nEnter the number of nodes : ");
	scanf("%d",&n);
	printf("\nEnter the cost matrix :\n");
	for(i=0;i<n;i++)
		for(j=0;j<n;j++)
		{
			scanf("%d",&dmat[i][j]);
			dmat[i][i]=0;
			rt[i].dist[j]=dmat[i][j];
			rt[i].from[j]=j;
		}
		do
		{
			count=0;
			for(i=0;i<n;i++)
			for(j=0;j<n;j++)
			for(k=0;k<n;k++)
				if(rt[i].dist[j]>dmat[i][k]+rt[k].dist[j])
				{
					rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
					rt[i].from[j]=k;
					count++;
				}
		}while(count!=0);
		for(i=0;i<n;i++)
		{
			printf("\n\nState value for router %d is \n",i+1);
			for(j=0;j<n;j++)
			{
				printf("\t\nnode %d via %d
Distance%d",j+1,rt[i].from[j]+1,rt[i].dist[j]);
			}
		}
	printf("\n\n");
}
```

Outtput:



Sign:- _____ **Date :-** _____

PRACTICAL: 12

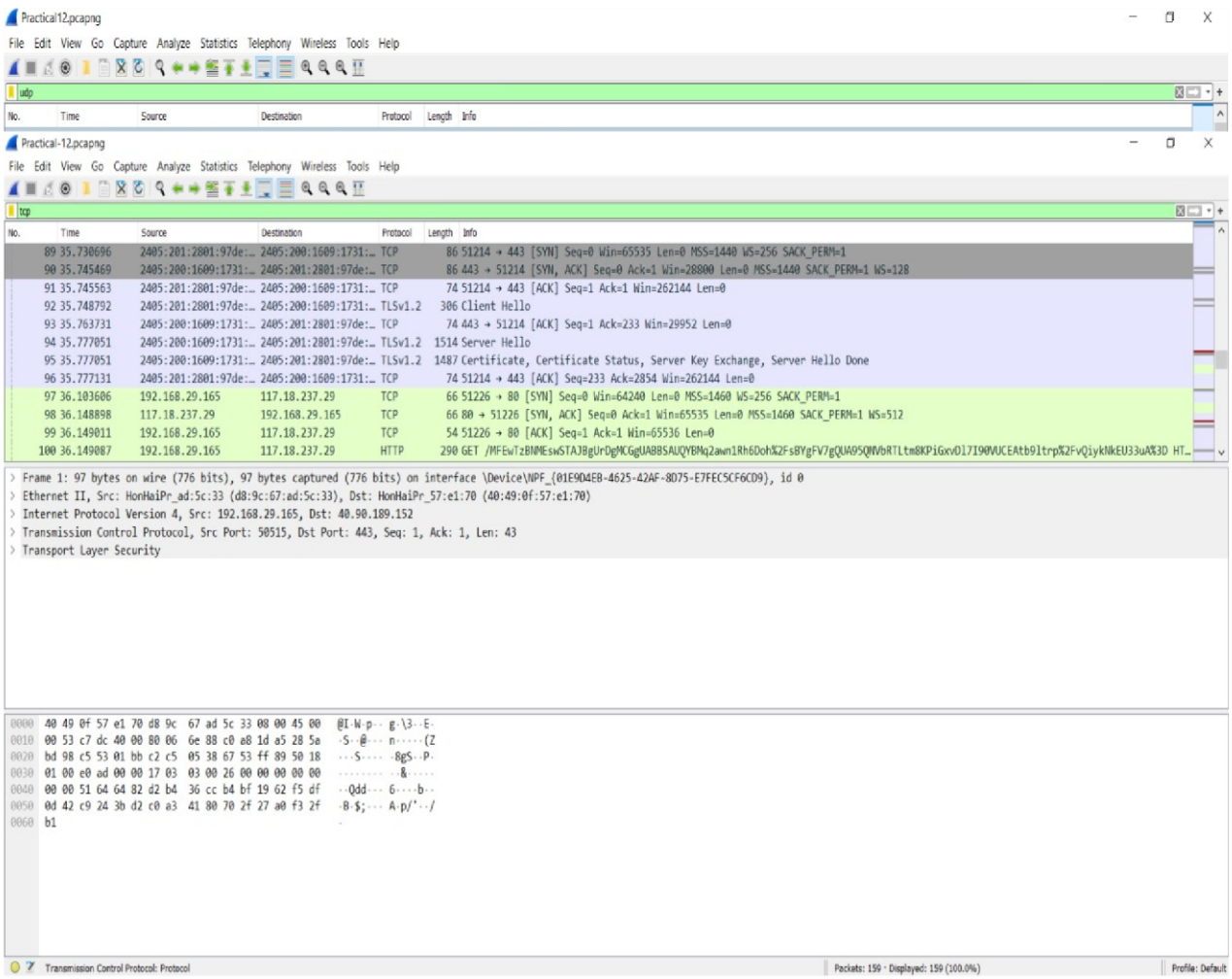
AIM: Packet capture and header analysis by wire-shark (TCP,UDP,IP).

Software: Packet Analyzer – wireshark

Introduction:

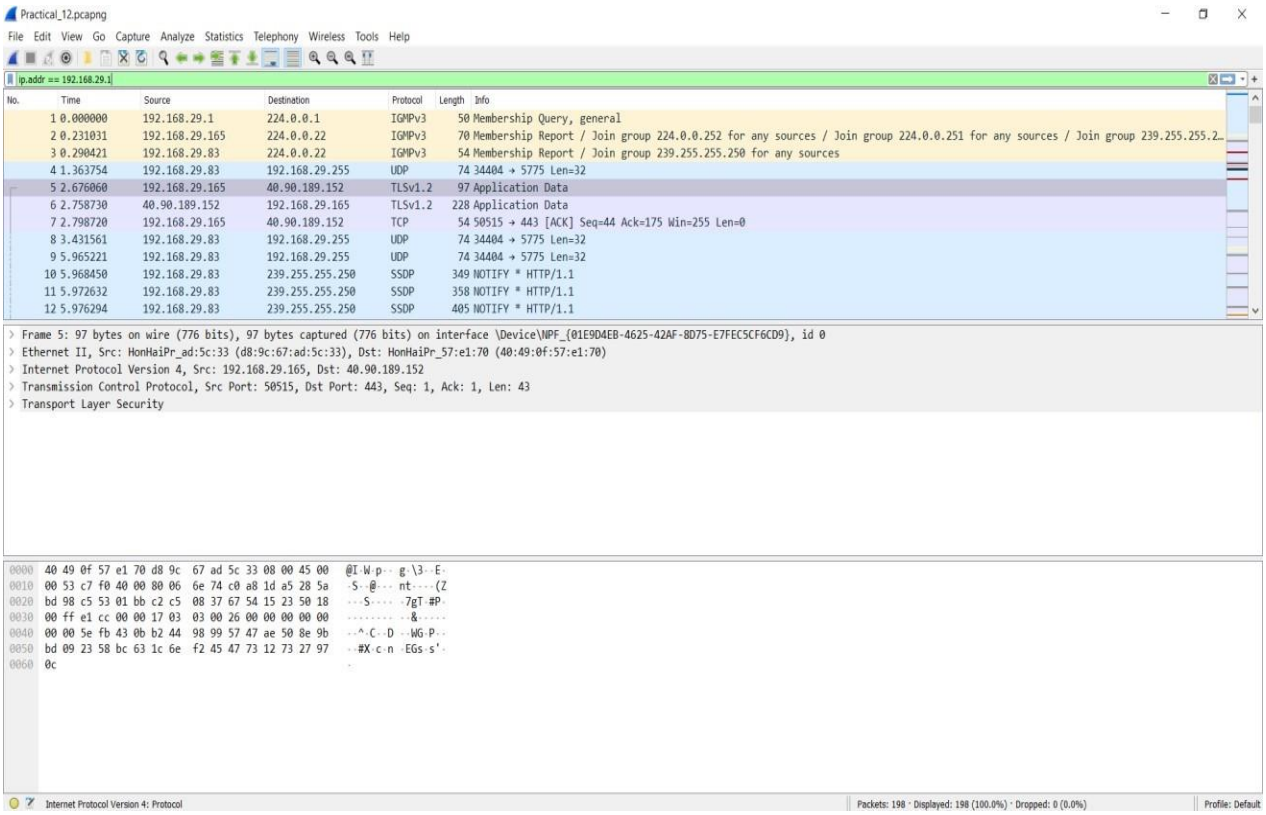
Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer.

- **UDP Packet Capture:**



- **TCP Packet Capture:**

IP Packet Capture:



Sign:- _____ **Date :-** _____