10/7/25

Ex:1

Aim

Study of various networking commands used in linux and windows.

Basic Networking Commands

1) arp-a

Interface: 172.16.75.15 --- 0x6

Internet Address	Physical Address	Type
172.16.72.1	75-59-1c-cf-be-41	dynamic

2) Hostname

DESKTOP-HSA DBUG

3) ip config/ all

Window IP configuration

Hostname . . . DESKTOP-HSA DBUG

primary DNS Suffix

4) nbtstat -a

Displays protocol statistics and current TCP/IP connection using NBT

5) netstat

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.1:49672	Desktop-DBUG	Establishment

6) nslook up

Default Server: unknown

Address : 172.16.78.1

7) Pathping

Usage: Pathping [-g host-list] [-h maximum-hops)

[-i address] [-n] [-p period] [-q num-queries)

[-w time out)

8) Route

Command is used to show manipulate the IP routing

Linux Commands

1) ip: Setting up new systems and assigning IPs to trouble Shooting existing system

ip <Options> [object] <commands>

a) ip address show

O/P

1. lo: <LoopBack, UP, Lower-UP>

2. @SP31 $: <NO-carrier, Broadcast, Multicast>

3. wlp2s0: <BROADCAST, MULTICAST, UP, Lower-UP>

b) Add an IP address

ip address add 192.168.1.254/24 dev eps3146

RTNETLDNK answers: File exists

c) To delete an IP

ip address del 192:168.1.254/24 dev enps31f6

d) Alter the status of interface (online)

ip link set esp31f6UP

e) Alter the status of interface (online)

ip line set enps31f6 down

f) Delete the route

ip route delete 192.168.1.0124 via 192.168.1.254

g) Display the route

ip route get 10.101.4

2) ifconfig: It is a staple in many sys admin's tool belt for configuring and trouble shooting networks.

O/P

enps31f6: flag: 4099 <UP, BROADCAST, MUITICAST>

ether 20:88:1086 sd:d4 txquelen 1000

Rx Packets 0 byts 0 (00B)

Rx errors 0 dropped 0 overrun 0 frame 0

TX Packets 0 bytes (0.0B)

3) mtr: (Matt's traceout) is a program with a command line interface that serve as a network diagonastic and troubleshooting tool

Syntax: mtr [options] <hostname/IP

a) The basic mtr command shows you the statistics

mtr google.com

O/P

My traceout [v0.95]

fedora [178.16.75.86] → google.com

Host	Packets			pings		
	loss%	snt	last	Avg	Best	Worst
1. gateway	0.0%	219	2.8	5.6	1.8	88.5
				18.0	4.6	53.8
2.142.250	0.0	261	7.8			

b) Show numeric IP address - mtr -g google.com

c) Show the numeric IP address

mtr -b google.com

d) mtr -c 10: google.com

4) tcpdump: for capturing and displaying packets

For install : dnf install -y tcpdump

a) tcpdump -i enp53s1f6

O/P: dropped pkts to tcpdump

b) tcpdump -i enp53s1f6 -c 10 host 8.8.8.8

O/P: dropped pkts to tcpdump

c) tcpdump -i enp53s1f6 port 53

To capture traffic to and from numbers

O/P: dropped pkts to tcpdump

5) Ping: It is used to trouble shoot connectivity, reachability and name resolution.

Ping google.com

O/P: PING google.com (142.253.221.238) 56(34) bytes of data

from kedero (192.168.8.296) icmp_seq=1

Destination host unreachable

Host unreachable

Result 31/7/25

Various network commands used in linux and windows has be executed successfully.

Ex:2

24/7/25

Aim:

Study of different types of Network cables

Cable Type	Category	Maximum Data Transmission	Adv / Disadvantage	Application
	Category 3	10 bps	Adv * cheaper in cost	10 Base-T Ethernet
UTP	Category 5	Upto to 100 Mbps	* Easy to install as they have a smaller overall diameter Dis Adv * More prone to Electromagnetic interference and noise	Fast Ethernet, Gigabit Ethernet Fast Ethernet, Gigabit Ethernet
STP	Category 6, 6a	10Gbps	Adv * Shielded * Faster than UTP	Gigabit Ethernet, 10G Ethernet
SSTP	Category 7	10Gbps	Dis adv * Expensive * Greater installation effort	Gigabit Ethernet
Fibre optics Cable	Single mode Multi Mode	100Gbps	Adv * High speed * High bandwidth Dis Adv * Expensive * Requires Skilled Installers	Maximum distance of fibre optics Cable is around 100 meters

Coaxial Cable	RG-6 RG-59 RG-11	10-100 Mbps	* High Bandwidth * Low loss bandwidth * versatile * easy * cost * size & bulky	Speed of Signal is slow Television Network High Speed.

Student observation

1. A straight cable connects different devices, while a cross connects similar devices
2. A cross cable is used to connect two pcs.
3. A straight cable is used to connect a router / switch to your pc.
4. The category of twisted pair cable used in the lab is typically cat5e or cat6.
5. Making a twisted pair cable requires precise wiring. common challenges include ensuring proper pin alignment.

10/10

Result

Successfully studied different types of Network.

31/8/25

Ex:3

Aim:

To study the packet tracker tool Installation and user Interface overview

Installation Packet Tracker

Windows

Installation in windows; the setup comes in a single file named Packettracer_setup 6.0.1.exe. open this file and start the installation.

Linux

Linux users with an ubuntu dist and those using fedora, Grant executable permission by using chmod and execute it to begin the installation.

Steps:

1. From the network component box, click and drag and drop the below components;

   a. 4 Generic PCs and one HUB

   b. 4 Generic PCs and one switch

2. Click on connections

   a. Click on copper - Straight Through cable

   b. Select one of the PC and connect it to HUB using cable

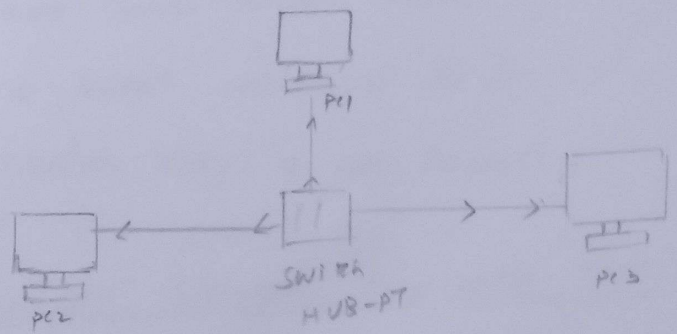
   c. Similarly connect 4 PCs to the Switch

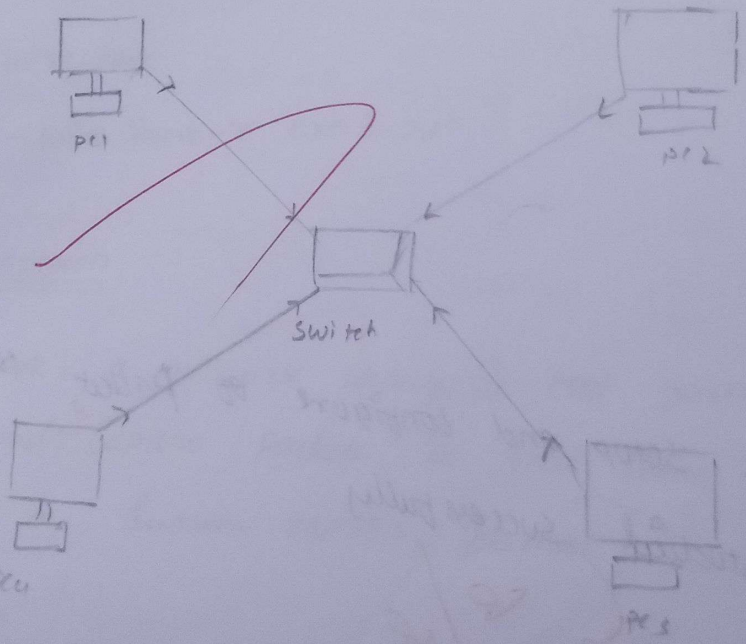
3. Click on the PCs Connect to hub, go to Desktop tab click on IP configuration and enter an IP address and Subnet mask.

4. Observe the flow of PDU from source PC to destination PC by selecting the realtime mode of simulation

5. Repeat Step #3 to Step #5 for the PCs connected to the Switch

6. Observe How HUB and Switch are forwarding the PDU and write your observation and conclusion abt the behaviors of switch and HUB.





Student Observation

a) A switch forwards packets only to the specific port of the destination device, minimizing unnecessary traffic.

b) Start topology all devices connected to a central switch, sometimes combined into a star- BUS hybrid topology – multiple star segments connected by backbone links

Result

Thus setup and configure of Packet tracer tool studied successfully

8/8/25

Aim

Experiments on Packet Capture tool : Wireshark

Packet Sniffer

* Sniffs message being sent / received from by your computer
* Store and display the contents of the various protocol
* Passive program:
  - never sends packet itself
  - no packets addressed to it
  - receives copy of all packets

Packet Sniffer Structure Diagnostic Tools

* TCP dump
  - Eg: tcpdump -env host 10.129.41.2 -w exe 3.out
* Wireshark
  - Wireshark -r exe 3.out

Description

Wireshark, a network analysis tool formerly known as the read capture packets in real time and display them in human-redable format

What we can do with wireshark:

* Capture network traffic
* Decode packet protocol using dissectors
* Define filters - capture and display
* Watch smart statistics
* Interactively browse the traffic

Capturing Packets

After downloading and installing wireshark, launch it and double click the name of a network interface under capture to start capturing packets.

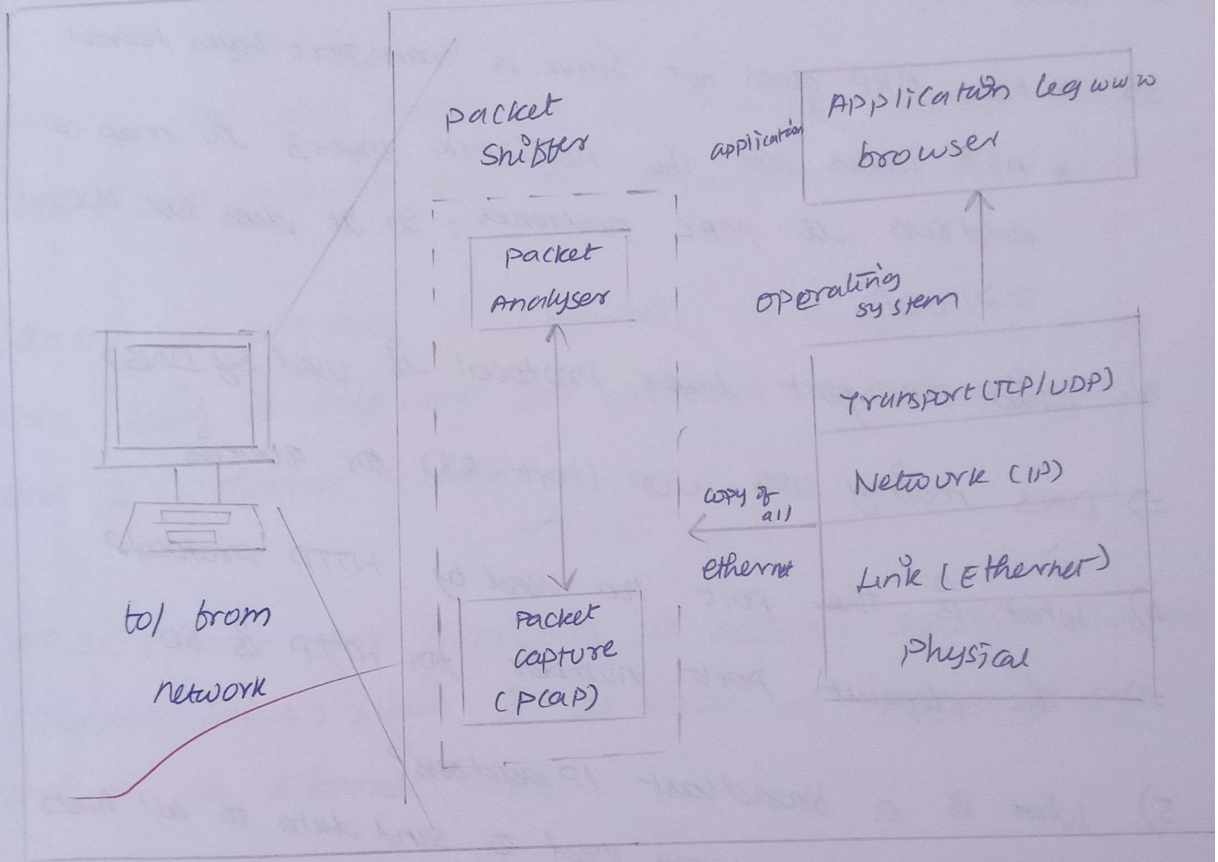
As soon as you click the interface name you'll see the packets start to appear in real time

The "Packet list" Pane

the packet list pane displays all the packets in the current capture file. The "packet list" pane each line in the packet list corresponds to one packet in the capture file.

The "Packet details" Pane

The packet details pane shows the current packet in a more detailed form. This pane shows the protocols and protocol field of the packet selected in the packet list pane.



Student Observation

1) What is promiscuous mode?

A mode of operation of a network interface card (NIC), where it captures all packets on the network not just those add addressed to it.

2) Does ARP packets have transport layer headers?

=> * NO, ARP does not have a transport layer header

* ARP works at the network Layer3 to map IP address to MAC addresses, So it does not use TCP/UDP

3) Which transport layer protocol is used by DNS?

=> DNS mainly uses UDP (Port 53) for queries

4) What is the port no used by HTTP protocol?

=> The default port number for HTTP is 80

5) What is a broadcast IP address?

=> A special IP address used to send data to all hosts in a network.

Result

Thus the experiment is completed successfully

28/8/25

9/10

18/25

Aim

To setup and configure a LAN [Local area Network] using a switch and Ethernet cables in your lab.

How to set up a LAN

Step 1: plan and design an appropriate network topology taking into account network requirements and location

Step 2: You can take 4 computers, a switch 8, 16 or 24 ports which is sufficient for network of these sizes and 4 Ethernet cables.

Step 3: Connect your computers to network switch via an Ethernet cable, which is as simple as plugging one end of the Ethernet cable

Step 4: Assign IP address to your PCs

1. Log on to the client computer as Administator
2. Click Network and Internet Connections
3. Right click Local Area Connection → Go to properties → Select Internet protocol → select up address option and assign IP address

Step 5: Configure a network switch

1. Connect your computer to the switch: To access the switch's web interface.

2. Log in to the web Interface: open a web browser and enter the IP address of the switch in the address bar.

3. configure basic settings: once you're logged in, you will be able to configure basic

4. Assign IP address as: 10.1.1.5, Subnet mask 255.0.0.0

Step 6: Check the connectivity between switch and other machine by using ping command

Step 7: Select a folder → go to properties → click sharing tab → Share it with everyone on the same LAN

Step 8: Try to access the shared folder from others computers of the network.

Student Observation

Write the outcome and challenges faced while configuration the LAN.

Outcome of configuring the LAN:

1) LAN was set up successfully
2) All devices like computers & printers are connected and could share files
3) Internet was working on all designs
4) IP addresses were set correctly
5) The Network was tested and worked well challenges Faced

1) Some devices had same IP addresses, causing connection error
2) Some cables were faulty or not connected properly
3) Setting on routers or switches were sometimes wrong

Result 12/09/25 10/10

Thus a LAN was setup and configured succesfully

10/9/05

Aim

To write a program to implement error detection and correction using Hamming code, concept. Make a test run to input data stream and verify error correction feature.

Create Sender program with below features

1) Input to sender file should be a text of any length. program should convert the text to binary.

2) Applying hamming code concept on the binary data and add redundant bits to it.

3) save this o/p in a file called channel.

Create Receiver program with below features

1) Receiver program should read the input from channel file.

2) Apply hamming code on the binary data to check for errors

3) If there is an error, display the position of the error

4) Else remove the redudant bits and the convert binary data to ascii and display the o/p.

Student observation

Sender.py

```
string = input ("Enter String")
S = ''.join (format (ord (i), '08b') for i in string)
rb = 0
for i in range (len(S)):
    if (2** i >= len(S) + i+1):
    rb=i
    break
m = len(S) + rb
l=[]
pos=[]
c=0
S= S[::-1]
for i in range (rb):
   pos. append (2**i)
for i in range (m):
    if (i+1) in pos:
      l. insert (i, 'P')
    else:
      l. insert (i, int (S[c]))
      c+=1
print ("parity bit positions :", pos)
Print ("Initial encoded data with 'P'", l)
```

```
for i in range (len(l)):
    if (2**i >= len(l)+i+1):
        rb=i
        break
for i in range (rb):
    pos.append (2**i)
print (pos)
q=[]
for i in l [::-1]:
    if (i=="1" or i=="1"):
        q.append (1)
    else:
        q.append(0)
for p in pos:
    count =0
    i =p-1
    while i<len(l):
        count += q[i:i+p].count(1)
for i in range (len (c [::-1])):
    change += c[i] *2**i
print ("Error", change)
print ("corrected code", q)
```

19/9/25

Result

Thus implementation of error detection and correction using Hamming code was executed successfully.

Aim

Write a program to implement flow control at data link layer using sliding window protocol.

Sender program:

1. Input window size from the user
2. Input a Text message from the user
3. Consider 1 character per Time
4. Reader a file called Reciever_Buffer

Receiver program

1. Reader a file called Sender-Buffer
2. check the frame no
3. If the Frame no. are as expected, with the appropriate ACK.

Student observation

Sliding.py

```
from re import x
import time
import os
os.system ('clear')
SB = open ("sender-Buffer.txt", "a+")
RB = open ("Reciever_Buffer.txt", "r+")
```

```
SB.truncate(0)
RB.truncate(0)
ws = int(input("Enter window size"))
s= list(s)
if (ws < len(s)):
  for i in range (0, len(s), ws):
    p= s[i:i+ws]
    y = s[i+ws:i+ws+ws]
    print (str(p))
    time.sleep(ws)
    print ("sending →", str(y))
    x=0
  while (x< ws):
    time.sleep(2)
  if (len(p)>x):
    print (p[x])
  time.sleep(1)
  if (len(y)>x):
    print (y[x])
    SB.write (y[x])
  x+=1
else:
  print ("The window size is too large.")
```

Result

Thus the program is implemented successfully

9/9/25 10/10

EX.NO:8

NMAP Live HOST DISCOVERY

Aim:

This experiment outlines the process that Nmap takes before Port-scanning to find which systems are online.

1) ARP scan : This scan uses ARP request to discover like host

2) ICMP scan : This scan uses ICMP requests to identify live hosts

3) TCP / UDP Ping scan : This scan uses sends packet to TCP ports and UDP ports to determine live hosts

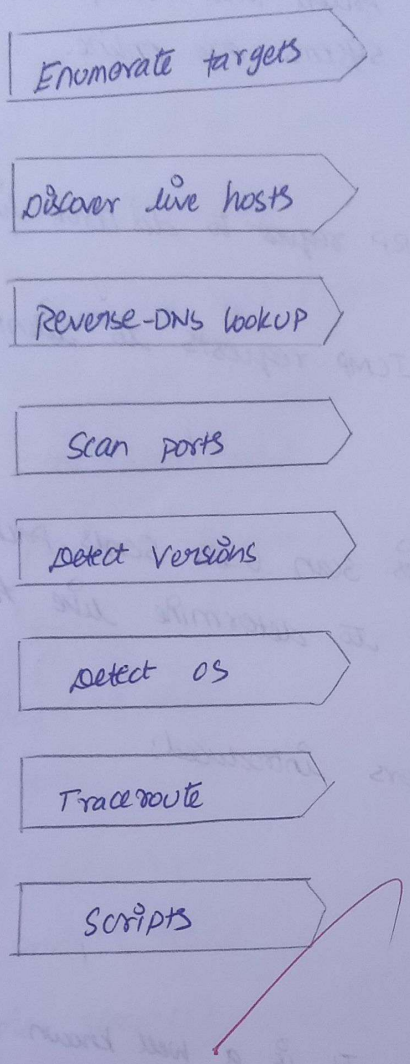
There will be two scanners introduced:

1) arp-scan
2) nmap scan

Nmap (Network Mapper) - It is a well known tool for mapping networks, locating live hosts, and detecting running services

Nmap's scripting engine can be used to extend its capabilities, such as finger printing services and appearing flaws

The Scan typically follow the steps represented in the image below.



Result

Thus the experiment was completed successfully

b) AIM:

Configure a wireless LAN using Cisco packet Tracer with three PCs connected to a Linksys wireless Router

Procedure:

1) Router configuration

* Access the Linksys wireless router
* Under Administration, set username
* Go to Wireless tab, set SSID to mother network
* under Wireless security, select WEP mode
* Set router IP address to 192.168.0.1

2) PC configuration

* Assign static IP addresses on each PC:

↳ PC0: IP - 192.168.0.2, Subnet - 255.255.255.0
Gateway - 192.168.0.1

↳ PC1: IP - 192.168.0.3, Subnet - 255.255.255.0
Gateway - 192.168.0.1

↳ PC2: IP - 192.168.0.4, Subnet - 255.255.255.0
Gateway - 192.168.0.1

3) Connect PCs to Wireless Network:

* On each PC, go to Dekstop > PC Wireless
* Refresh "to default Mother network on channel with WEP Security
* verify Connection and active Wireless card

Student observation

a) What is SSID of a Wireless router?

=> SSID [Service Set Identifier] is the name of Wireless network that identifies the router's Wireless Signal

b) What is a Security key in Wireless router?

=> A Security key is a password used to protect the Wireless network from unauthorized access.

c) Configure a simple Wireless LAN in your lab using a real access point and write down the config

=> Set the access point's SSID, configure Security mode, with a password, assign IP addresses to devices.

Result

Thus the configuration of Wireless LAN using CISCO packet tracer is implemented.

Ex: 9

Subnetting in Cisco Packet Tracer

Aim:

Implementation of Subnetting in cisco packet tracer Simulator

Concept

Classless IP Subnetting allows dividing a network into smaller Subnets for efficient IP address utilization. Instead of fixed classful masks, Subnetting uses variable-length subnet masks.

Steps

1) Create Topology : Add routers, Switches and PCs in Cisco Packet Tracer

2) Subnetting : eg) 192.168.1.0.124 Subnetted into 27 gives 8 subnets with 30 hosts each

3) Assign IPs : Allocate Subnet IPs to routers, switches and PC's

4) Configure devices : Router → Configure Interfaces with IP
Switch → set access mode on ports

5) Test : use ping to check connectivity among devices

15/7 9/10

Result

Thus Subnetting in CISCO Packet tracer is implemented Successfully.

Ex:10

Internetworking with routers in cisco packet tracer simulator

Aim:

Internetworking with routers in cisco packet tracer, design a simple network with 1 router and using cisco packet tracer

procedure

1) configure Router

* Go to CLI → enable → config it
* Set IP for Fast Ethernet 0/0 : 192.168.10.0.1 255.255.255.0 → no shutdown
* set IP for Fast Ethernet : 192.168.20.1

2) configure PCs

* PC0 : IP 192.168.10.2
* PC1 : IP 192.168.20.2

3) Connected devices

* PC0 → Router Fast Ethernet 0/0
* PC1 → Router Fast Ethernet 0/1

4) Test Connectivity

* ping from PC0 to PC1 to verify network connection

4) Aim :

Configure a network with a wireless router, DHCP server and internet connection

Procedure :

1) * Add Wireless router, PC, laptop, Cabel Modem, Internet Cloud
* connect using copper and coxial cabels as required

2) Configure Wireless Router
* set SSID : Home Network
* Enable DHCP & Set DNS : 208.67.220.220

3) Configure laptop & PC
* laptop : Install wireless module → connect to Home network
* PC : Enable DHCP → Get IP automatically from router

4) Configure cisco.com server
* Enable DHCP & DNS services with IP : 208.67.220.2

verify connectivity
* Refresh IP on PC
* ping disco.com to test internet Connection

Student observation

1) Write down the key features of configuring wireless routes

=> Wireless Routes : Provides wifi, sets SSID, Connect devices

2) What is the significance of DHCP server in inter networking

=> * Simplifies IP assignment & avoids conflicts

* Ensures smooth network communication

3) Design & configure an Inter-network

=> * Router : 192. 168.1.1

* PC0 : 192. 168.1.2, Gateway : 192.168.1.1

* PC1 : 192. 168.1.3, Gateway : 192.168.1.1

15/1 x 9/10

Result

Internetworking with routers in CISCO packet tracer simulator is done successfully

Aim:

Static Routing configuration, to simulate static routing configuration using Cisco Packet Tracer & Understand how to manually add, verify and manage static routes in a network.

Procedure:

1) open Cisco packet tracer & create 3 routers with PCs
2) Connect router and PCs using proper cabels
3) Assign IP address to all interface based on the given network Table
4) Configure static routes on each router
5) verify routing table to check main and backup router
6) Test Connectivity using ping & tracer b/w PCs
7) Simulate link failure by removing a connection and check backup route
8) delete static route if needed using CLI router

AIM:

To Simulate using routing Information protocol (RIP) with the help of cisco packet tracer

Procedure:

1) Create a network using 3 routers & PCs

2) Connect all devices & assign IP address as per table

3) Enable all interfaces of routes

4) Enable RIP protocol on all Routes

5) Check using Routing tables entries learned thru RIP

6) Verify network connectivity by using ping

Result

Simulate Static Routing configuration is done Successfully.

EX:12

a) Aim

Implement echo client server using TCP / UDP Sockets

Client

```
import socket;
import time;

def ping_server(host="127.0.0.1', port=12345)
    with socket.socket(socket_AF_INET, socket.
        sock_DGRAM as s:

try:
    s.sendto(b,'Hello', (host, port))
except s.timeout:
    print("Request timed out")

ping_server()
```

Result

The message hello is successfully sent to the

5) Aim

Implement Chat client Server using TCP/ UDP Sockets

Algorithm

```
import Socket;
def start-server (host = '127.0.0.1', port=12345)
    with socket.socket (Socket.AN-INET)
                        (SOCK-DGRAM) as S:
        S.bind ((host, Port))
        print (f" UDP server is running)
        While true:
            data . addr : S. recvfrom (124)
start-server()
```

O/P

UDP server running on 127.0.0.1 : 12345

Result

The chat client server using TCP/ UDP Socket is implement successfully

Implement your own ping program

Aim:

Implement your own ping program

Algorithm:

```
import socket
import time

def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.
    try:
        s.settimeout(2)
        start = time.time()
        s.sendto(b'Ping', (host, port))
        data, addr = s.recvfrom(1024)
        end = time.time()
        print(f"Receive {data.decode()} from
        in {end-start:.2f} seconds")
except socket.timeout:
    print("Request timed out")

if __name__ == "__main__":
    ping_server()
```

b)

```
import socket

def start-server (host='127.0.0.1', port=12345)
    with socket.socket (socket.AF_INET,
                        socket_DGRAM) as s:

s.bind ((host, port))

print (f "UDP server running on {host}:{port}
    {data.decode ()}")

while True:

    data, addr = s.recfrom (1024)
    print (f "received message from {addr}:
        {data.decode ()}")
    s.sendto (b'Pong', addr)

start-server();
```

O/P

UDP server running on 127.0.0.1:12345

Result

Hence the code executed successfully

2/10 x 10/10

CODE Using Raw Sockets to Implement Packet Snuffing

Aim:

To write a code using raw sockets to implement packet snuffing.

Algorithm:

```
from scapy.all import sniff
from scapy.layers.inet import TCP, IP, UDP

def packet_callback(packet):
    if IP in packet:
        ip_layer = packet[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst

    protocol_name = " "
    if protocol == 1:
        protocol_name = "ICMP"
    elif protocol == 6:
        protocol_name = "TCP"
    elif protocol == 17:
        protocol_name = "UDP"
    else:
        protocol_name = "unknown protocol"
```

```
print (f" protocol : {protocol-name}")
print (f" source IP : {src-ip}")
print (f" Destination IP : {dst-ip}")
print ("-" ,50)
```

Output :

Protocol : TCP

Source IP : 192.168.1.5

Destination IP : 192.217.14.206

protocol : UDP

Destination IP : 8.8.8.8

Result

Hence the code executed successfully