INDEX

Name G.Vigneshwaran Class CSE-D

Sl No.	Date	Particulars	Marks	Signature of Teacher /Remarks
1	14/7/25	Study of various N.work commands used in Linux and Windows		24/8/25 9
2	24/7/25	Study of different types of network cable		31/8/25 2
3	24/7/25	Study of packet tracer installation and user interface overview		
4	28.8.25	setup & configure ALAL		
6	14/8/25	Hamming code		18/9/25 6
7	14/9/25	sliding Windows		
8	11/9/25	NMAP to discover		
9	18/9/25	Implementation of subnet using cisco		
10	18/9/25	Interworking with routers		
11	22/9/25	Simulate RIP routing		
12	25/9/25	Client-server using TCP		
13	29/9/25	Ping Program		
14	6/10/25	Packet switching using RAW		

14/7/21

Practical - 1

AIM:- Study of various networks commands used in Linux and Windows.

Basic Networking commands

1) arp - a - shows the IP and MAC addresses of devices on your network

O/P: Interface : 172.16.75.78 ---0x5

172.16.72.1	7C-5a-1c-cf-bC-41	dynamic
244.0.0.2	ff-ff-ff-ff-ff-ff	static
239.192.152.143	01-00-5C-40-9B-BF	static

2) hostname - Displays name of your system

O/P: K503-78

3) IP config / all = Displays detailed TCP/IP configuration

O/P: Windows IP config

Hostname : K503-78

Node type : hybrid

ethernet adapter

Media state : disconnected

4) nbstat -a - displays NetBIOS name resolution statistics, useful for diagnosing name resolution.

Output

NETBIOS Remote machine Name table

Name	Type	Status
KALI3	<00>	Registered

5) netstat shows network statistics, such as active connections, routing tables and interface info

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address		
tcp	0	0	localhost:22	*:*	LISTEN

6) nslookup - Resolves domain names to IP addresses

Server: Unknown
Address: 192.168.1.1)
Non-authoritative answer.
Name: google.com
Addresses: 142.250.72.238

7) Pathping ~~Combines ping and tracert~~

~~Combines ping and tracert~~

Pathping is unique to windows OS and it is basically a combination of ping & tracert commands.

Pathping [-g host-list] [-h maximum_hops]
[-i address] [-a] [-p period] [-q num_queries]
[-w timeout] [-4] [-6] target_name

8) Ping: command which is the best way to test connectivity between two nodes.

Pinging google.com [142.250.77.46] with 32 bytes of data

9) Route: Displays or modifies the kernel IP routing table

route Print

Active Routes

Network Destination	Network mask	Gateway	Interface	Metric
192.168.1.0	255.255.255.0	on-link	192.168.1.5	281

Linux commands:

1. ip : It is one of the basic commands every administrator will need in daily work.

a) ip address show

172.16.0.1.254

b) ip address add 192.168.1.254/24 dev enp503

c) ip address del 192.168.1.254/24 dev enp503

d) ip link set eth0 up

2) ifconfig: It is a stable and many sysadmin tool built for configuring and troubleshooting networks. It has been

eth0 : flags = 4163 <UP, BROADCAST, RUNNING, MULTICAST mtu 1500

inet 192.168.1.5 netmask 255.255.255.0

broadcast 19.2.168.1.255

3) mtr : It is a program with a cc
interface that serves as a netwo
and troubleshooting tool.

a) mtr google.com

Host : myhost.local Loss %. Snt Last Avg Best Wrst

1. |-- 192.168.1.1 0.0%. 10 0.8 0.7 0.5 1.0 0.2

) mtr -g google.com

Host Loss % Snt Last Avg Best Wrst stdev

gateway.home 0.0%. 10 0.7 0.8 0.6 1.2 0.2

) tcpdump

12:17:03.123456 IP 192.168.1.10.55432 > 142.5

Flags [S] seq 123456789

) Ping : It is a tool that verifies IP-level
connectivity to another TCP/IP computer

Ping google.com

64 bytes from seq 02 ttl 27 - in (216.58.204.774)

icmp - seq=1 ttl = 52 time = 10.7 ms

Li... connection show

NAME UUID TYPE DEVICE

Wired connection 1 a5bb490-d20 ethernet ens160
-3668

2) nmcli connection add con-name if name type
connection ethernet.

3) nmcli connection modify "wired connection"
used for rename

4) nmcli connection show
IPV4.method : auto
IPV6.method : auto

5) nmcli connection modify "Wired connection"
IPV4.method auto

6) nmcli connection modify "Wired connection"
ipb.method auto

7) nmcli connection up method "LAN"

[redacted]

Student observation

1) which command is used to find the reachability of a host machine from your device.

=> Ping <hostname or IP> ping google.com

2) which command will be give the details of hops taken by a packet to reach it destination

=> mtr <hostname>
mtr google.com

3) which command displays the IP configuration of your machine?

on linux : ip address show

on windows : ipconfig / all

4) which command displays the TCP port status?

netstat
netstat -

5) write the command to modify the the IP configuration in a Linux machine

do add : ip address add 192.168.1.254/24 dev enp0s3/16

do del : ip address del 192.168.1.254/24 dev enp0s3/16

Result

Therefore the above Linux and Windows commands are executed successfully.

21/8/20

Practical - 2

Aim: Study of different types of network cables.

(a) Understand different types of network cable.

Different types of cables used in networking cable

1. Unshielded twisted pair (UTP) cable
2. Shielded twisted pair (STP) cable
3. Coaxial cable
4. Fibre optic cable

Cable type	Category	Maximum data transmission	Adv/ Disadv	Applications / use
UTP	Category 3 Category 5 Category 5e	10 bps UP TO 100 Mbps 1 Gbps	Adv · Cheap in cost · Easy to install Disadvantages · More prone to EMI	10 Base T Ethernet Fast Ethernet Gigabit Ethernet Fast Ethernet Gigabit
STP	Category 6, 6a	10 Gbps	Adv · Shielded · Faster than UTP	Gigabit Ethernet 10 G Ethernet (55m)
SSTP	Category 7	10 Gbps	· Less susceptible to noise Disadv · Expensive · Inflexible	Gigabit Ethernet 10 G Ethernet (100m)
Coaxial cable	RG-6 RG-59 RG-11	10 - 100 Mbps	Adv · High bandwidth · Immune to interference · Versatile Disadv · Limited distance · Cost · Susceptibility	Speed of sound transmission television network High speed internet connections
Fibre optics cable	Singlemode Multimode	100 Gbps	Adv · High speed · High bandwidth · High security · Long distance Disadv · Expensive · Requires skilled installers	· Maximum distance of fibre optic cable is around 100 meters

Student observation

1) What is the difference between cross cable & straight cable?

Ans:- Cross cable connects similar devices while straight cable connects different devices.

2) Which type of cable is used to connect 2 PC's?

Ans: cross cable

3) Which type of cable is used to connect a router / switch & your PC?

Ans:- straight cable.

4) Find out the category of twisted pair cable used in your to connect the PC to network socket.

Ans: Cat 5e or cat 6.

5) Write down understanding, challenges faced and output received while making a twisted pair cross / straight cable.

Ans:- making cables required careful pin alignment, challenge was crimping properly, but the output was working correctly.

Result: Different types of Network cable studied successfully.

31/10/25

4/7/25

5) Study of packet tracer tool Installation and user Interface overview

EXNO 6

Aim: To study the packet tracer tool Installation and user Interface

(a) Cisco packet tracer has been successfully installed

(b) Analyse the behaviour of network devices using cisco packet tracer simulation.

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

(a) 4 Generic PCs and one HUB

(b) 4 Generic PC 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 the cable, the link LED should glow in green.

(c) Similarly connect 4 PCs to the switch using copper straight-through cable

3. click on the PCs connected to hub, go to the desktop tab, click on IP configuration, and enter an IP address and subnet mask.

Click on the PDU from the comman tool bar

(a) Drag and drop it on one of PC and then drop it on another PC connected to the HUB

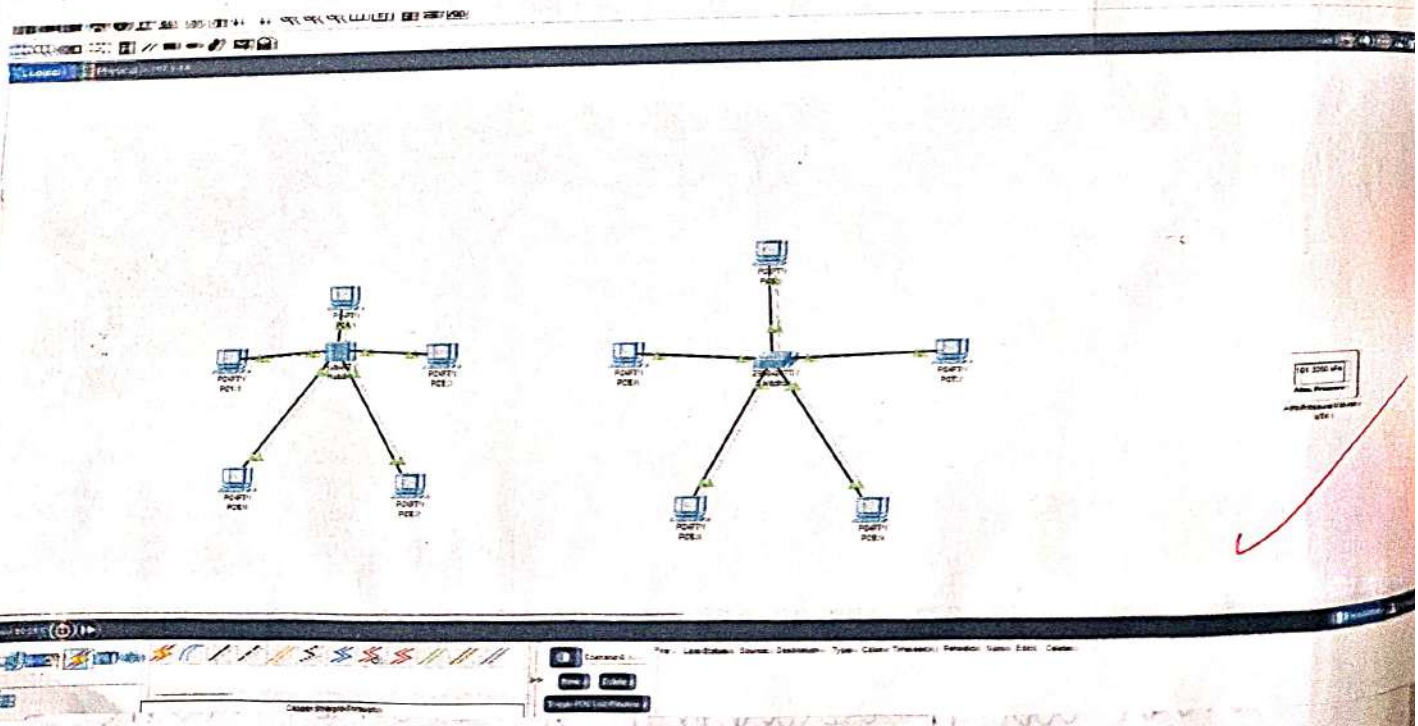
4) observe the flow of PDU from source PC to destination PC by selecting the Realtime mode of Simulation

5) Repeat steps 3 to step 5 for the PC'S connected to the switch

6) observe how HUB and switch are forwarding the PDU and about the behaviour of switch and HUB

17/25, 10:15 PM

Screenshot 2025-07-31 090324.png



Student observation:

a) From your observation write down the behaviour of switch and HUBS in terms of forwarding the PACKET received by them.

A switch forward packets only to the specific device (PORT) based on MAC address, while a hub broadcasts packets to all connected devices.

(b) Find out the network topology implemented in your college and draw and label that topology in your observation book.

The network topology commonly used in college is star topology, where all devices are connected a central switch or hub.

15/10/16

Result:

The packet trace tool installation and user interface overview is studied.

Experiment - 5

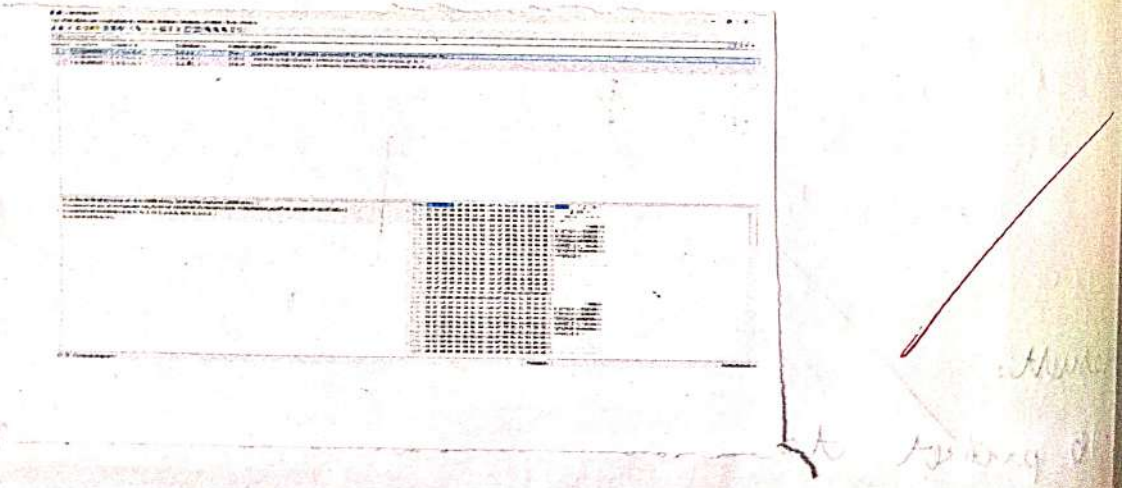
Aim:

Experiments on packet capture tool: wireshark.

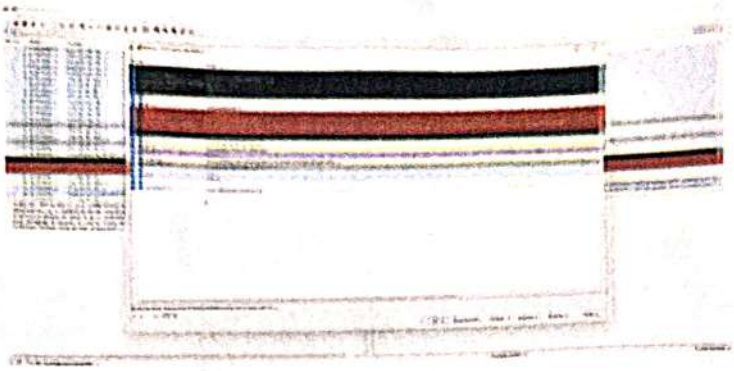
Capturing packets:

After downloading and installing wireshark, launch it and double click the name of a network interface.

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



To view exactly what the colour codes mean. click view colouring rules.



Capturing & analysing packets using wireshark tool

1. Filter TCP/UDP packets

-> select local area connection in wireshark capture -> option.

-> select stop capture automatically after 100 packets

. The click stop capture

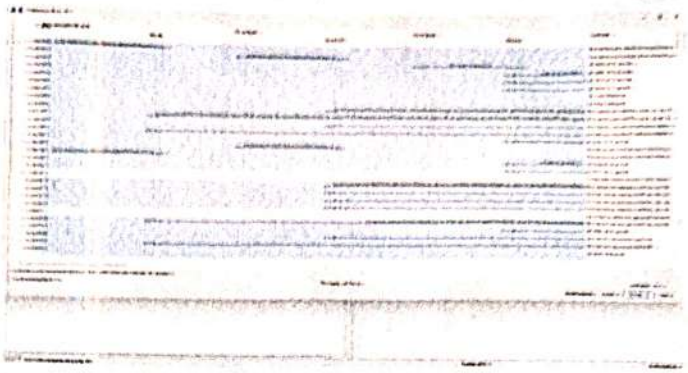
. Search TCP packets in search bar

. To see flow graph click statistics -> flow graph

. save the packets

5) Filter to display IP / SCHP packets
- search ICMP / IP in search bar

6) Filter no display only DHCP packets.



Solution observation

1) What is promiscuous mode?
It is a network interface mode in which a network card capture all the network packets regardless of their destination MAC address

2) Does ARP header have transport layer header
No ARP is a part of network layer

3) Which transport layer protocol is used by DNS?
DNS uses both: UDP and TCP

4) What is the port number used by HTTP protocol?
Port 80

5) What is broadcast IP address?
It is used to send data to all host on specific network segments.

Result:

Exp no: 6 experiment-6 Hamming code

Date : 25.8.26

Aim : 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.

code :

```
#include <stdio.h>
#include <math.h>
int calc_parity (int data[], int size, int
    int parity = 0;
    for (int i=0; i< size; i++){
        if (i & (i<<p)){
            parity ^= data[i];
        }
    }
    return parity;
}
for (i=0; ; i<r; i++)
{
    int parity_bit = pow(2, i);
    int parity=0;
    for (j=1; j<=n; j++) {
        if (j & parity_bit)
            parity ^= code[j];
    }
    code[parity_bit] = parity
}
```

```
for (i = n; i >= 1; i--)
printf("%d", code[i]);
printf("\n");
}
void detectAndCorrect()
{
unit n = m + r;
int i, j, errorPos
for (i = 0; i < r; i++)
{
int parityPos = pow(2, i);
int parity = 0;
for(j = 1; j <= n; j++)
{
if (j & parityPos)
parity ^= code[j];
}
if (parity != 0)
errorRes += parityPos;
}
if (counter == 0) {
printf("No error");
}
else if (errorPos <= n) {
printf("Error at Position: %d", errorPos);
code[errorPos] ^= 1;
printf("corrected code:");
for(i = n; i >= 1; i--)
Printf("\n");
}
use a
```

```
printf("Multiple bit errors");
    }
}
int main()
{ int i, pos;
  scanf("%d", &m);
  if (m<1 || m>1) {
    printf("Invalid");
    return 0;
  }
for (i=0; i<m; i++) {
    printf("Enter bit %d:", i+1);
    scanf("%d", &data[i]);
  }
  generateHammingCode();
printf("Position to enter error");
scanf("%d", &pos);
if(pos!=0 && pos<=m+r)
  {
    code[POS] ^= 1;
    for(i=m+r; i>=1; i--)
      printf("%d", code[i]);
    printf("\n");
  }
  detectAndCorrect();
    return 0;
}
```

Sample I/P. O/P

Enter no. of data bits: 4

Enter bit 1: 1

Enter bit 2: 0

Enter bit 3: 1

Enter bit 4: 1

Generated Hamming code: 1100110

Enter position for error: 3

Code after error: 1100010

Error detected at position: 3

Corrected code: 1100110

Result:

Hence the hamming code concept has been implented successfully.

14.8.25 Experiment - 7

Aim:

write a program to implement flow control using sliding window protocol.

Code:

```
#include <stdio.h>
#include <stdl
#include <string.h>
#include <unistd.h>
#include -
#define MAX 50
int main()
{ char message[MAX];
  int window = size;
  printf("Enter Message");
  scanf("%s", message);
  scanf("%d", &window-size);
  int length= strlen(message);
  int base=0;
  int next= 0;
  srand(time(0));
  while(base <length)
  { while(next< base+window-size && next<length)
    {
      printf("[5]sending frame : d ->%c\n",
      next, message[next]);
      sleep(1);
    }
```

```
int error_frame = base + (rand() % (next - base));
int error = (rand() % 4 == 0);
if (error)
{
    Printf("[R] Error! NACK for frame %d", error_frame);
    Sleep(1);
    printf("[S] Retransmitting");
    next = error_frame;
}
else
{
    Printf("[R] ACK received for frame %d -> %d,"
    base, next - 1)
    base = next;
    }
    Sleep(2);
}
Printf("All frames transmitted successfully!");
return 0;
}
```

Sample Input / output

Enter Message : Hello

Window size : 2

sending frame 0 -> H

1 -> e

ACK 0 -> 1

sending frame 2 -> l

3 -> l

NACK for frame 3

Retransmitting

Sending Frame 3 -> l

ACK 2 -> 3

sending frame 4 -> 0

NACK for frame 4

~~sending~~

All frame transmitted Successfully

Result:

Hence the sliding window protocol has been implemented successfully.

11.09.25

Experiment - 8

Aim: To discover Live hosts using Nmap Scans on the tryHackMe platform room

This experiment outlines the process that Nmap takes before port scanning to find which systems are online. This stage is critical since attempting to pre-scan offline systems will merely waste time and create unneeded network noise.

The following is the information that will be covered in an attempt to discover live hosts:

There will be two scanners introduced

1. arp-scan
2. nmap

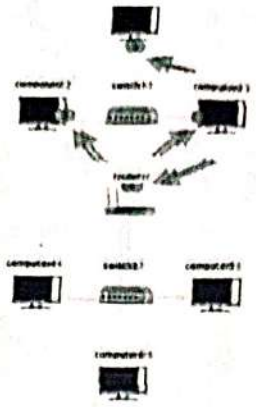
Nmap (network Mapper) - It is a well-known tool for mapping network. locating live hosts and detecting running services

Nmap's Scripting engine can be used to extend its capabilities such as fingerprinting services and exploiting flaws

A Nmap scan usually goes through steps, as in figure below:

1. Enumerate targets
2. Discover Live hosts
3. Reverse DNS
4. Scan port
5. Detect versions
6. Detect OS
7. Trace Route
8. Scripts
9. Write output

output



Result:

Thus we learnt how ARP, ICMP, TCP and UDP can detect the hosts by completing the tryHackMe room.

15.09.25 Experiment - 9 Implementation of Sub w cisco packet tracer

Aim:

To implement subnetting w cisco packet tracer

Steps:

1. Creating a network topology:

· Create a network topology. Select "New", it "Network" & generic. This will create a blank network topology.

2. Adding the devices

· to add a device, select device from bottom left corner and drag it into the topology.

· Add PCs, switches and routers as required

· Connect the devices using cable from one port to another devices port

3. Subnetting

· To subnet the address of 192.168.1.0/24 Cannot 127 mask.

· This will give us 4 subnets with 30 hi

Ex: PC-1 - 192.168.1.11 Switch1 Router

PC1 - 192.168.2.11 Switch2 192.168.2.1

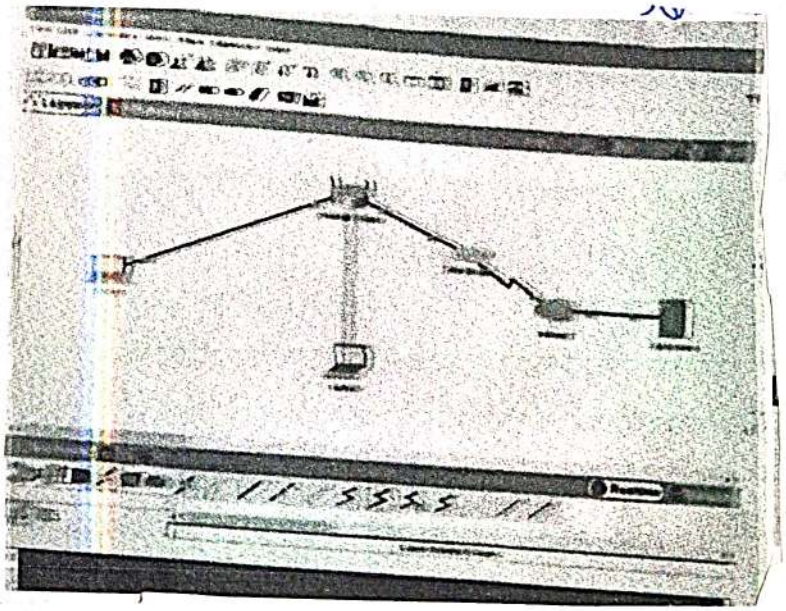
1. Configuring the devices

- open cli on the router
- Enter

enable
configure terminal
interface fast ethernet 0 1 0
ip address { IP address } { subnet mask }
no shut down
exit

- Replace { IP address } { subnet mask } with your desired address.

Output



Result:

This experiment was practiced successfully and Subnetting was implemented

26.9.25 Experiment - 10 - a

Routers in cisco packet tracer

Aim: Interworking with routers in CISCO PACKET Tracer.

Design and configure a simple intranetwork using a router.

On this network, a router and 2 PCs are used. computers connected with routers using copper straight - through cable. After forming the network, to network connectivity a simple, PDU is transferred from PC0 to PC1.

Procedure

Step - 1 (Configuring Router):

1. Select the router and open CLI.
2. Press Enter to start configuring Routers
3. Type enable to activate the privileged mode.

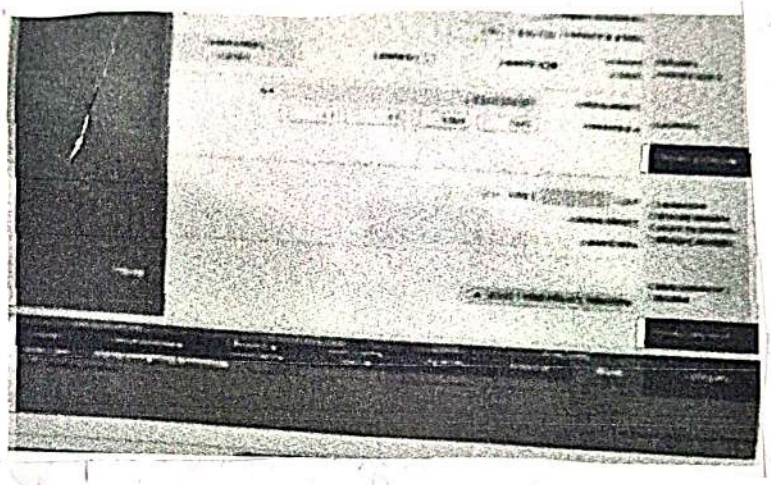
Step - 2 (Configuring PC):

1. Assign IP addresses to every PC in the network

· Select the PC, go to the desktop and select IP configuration and assign an IP address, default gateway, subnet mask.

3. Assign the default gateway of PC0 as 192.168.10.1

4. Assign the default gateway PC1 as 192.168.20.1,



Result:

Interconnecting of routers was completed Successfully.

30.9.25 Experiment 10 - b

Aim: Design and configure and interconnect using wireless router, DHCP server and internet cloud

Part 1 :- Build a simple network the logical topology workspace

Step 1 : Launch packet Tracer

Step 2 : Build the topology

(a) Add network devices to the workspace using the device selection box, add the network devices to the workspace.

(b) change display names of the network devices to change the display names of the network devices logical workspace. Then click on the config table in the device configuration window.

(c) Add the physical cabling between devices on the workspace

Part 2: configure the network

Step 1: configure the wireless router

(a) create the wireless network on the wireless router.

(b) click on the save settings

Step 2: configure the laptop

a) configure the laptop to access the wireless network.

step 3: configure the PC

a) configure the PC, for the wired network.

Step 4: configure the internet cloud

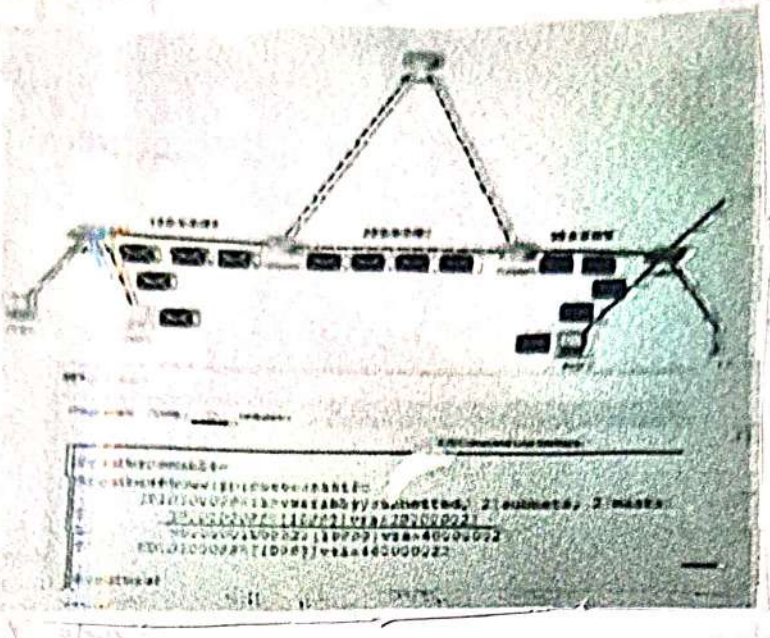
(a) Install network modules if necessary

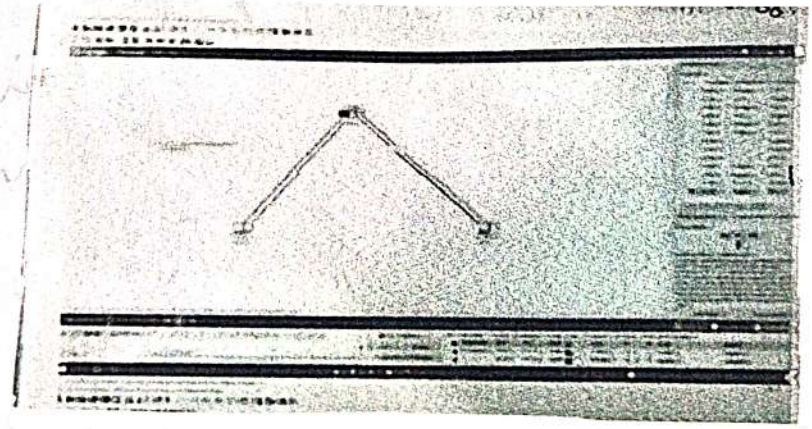
b) Identify the form and to ports

C. Identify the type of provider

Step 5: configure the cisco.com server

(a) configure the cisco.com server a DHCP server.





Result:

Design be configuration of inter networking using router, DHCP was created successfully

16/10/25

12.9.25 Practical -11 static Routing configuration

Aim: (a) Simulate static Routing Configuration using Cisco Packet tracer.

static routes are the routes you manually add to the routing table. The process of adding static routes to the routing table is known as static routing.

Let's take a packet tracer example to understand how to use static routing to a create and add a static route to a practical lab:

create a packet tracer lab as shown in the following image or download the following pre-created lab and load it on packet tracer.

Router C) Requirements:

Create two routes for network 30.0.0.0 and configure the first route as the main route and the second route.

- create two route for the host 30.0.0.100, and configure the first route (Via-route 1) as the main route and the second route (via-Router 1) as a backup route.
- Create two routes for network 50.0.0.1 8 and configure the first route (Via-Router 2) as the main route and the second route (Via-Router) as backup route.
- Verify the router adds only main router to the routing table.

Verifying Static routing

on-Router (1), we configured two routes for network 30.0.0.0/8. There route are Via. Router and Via Router. We set first route as the backup note.

We can verify this configuration in two ways.

Aim: b) Simulate RIP using CISCO Packet Tracer

Assign IP address to PC's

Double click PC's and click Desktop menu then and click IP configuration. Assign IP address referring the above table.

Assign IP address to interface of router

Double click Router() and click CLI and Press enter key to access the command prompt of Router()

We need to configure IP address and other parameters or interfaces before we could actually use them for routing. Interface mode is used to assign IP address and other parameters. Interface mode can be accessed from global configuration mode.

Result

01.10.25 Practical - 12 - TCP/UDP packets

Aim: (a) Implement echo client server using TCP/UDP Socket

Code:

TCP Server:

```
import socket
HOST = '127.0.0.1'
PORT = 65432
with socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((HOST, PORT))
    s.listen()
print(f'TCP server listening on {HOST}:{PORT}')
conn, addr = s.accept()
with conn:
    print:
        print(f"connected by {addr}")
    while True:
        data = conn.recv(1024
        if not data:
            break
print(f"Received {data.decode()}")
conn.sendall(data)
```

TCP client

```
import socket
Host = '127.0.0.1'
port = 5432
with socket(socket.AF_INET,
socket.SOCK_STREAM)
s.connect((HOST,PORT))
while TRUE:
    message = input("You:")
    if message.lower() = 'quit':
        break
s.sendall(message.encode())
    data = s.recv(1024)
print('Echo: {data.decode()}')
```

Input: Client types something

Server output: Client: Ping

Client output: You: Ping server: ACK: Ping

Result:

Implementation of echo client server using TCP/IP sockets was completed successfully

07.10.25 Practical 12-b · Chat client server.

Aim b) Implement chat client server using TCP, UDP sockets.

Code:

UDP Server

```
import socket
Host = '127.0.0.1'
Port = 65433
with socket.socket(socket.AF_INET, socket.SOCK_DGRAM):
    s.bind((Host, PORT))
    print(f"UDP Server listening on {Host}:{PORT}");
    while True:
        data, addr = s.recvfrom(1024)
        print(f"Received from {addr}: {data.decode()}")
        s.send(data, addr)
```

UDP client:

```
import socket
Host = '127.0.0.1'
Port = 65433
```

```
with socket.socket(socket.AF_INET,
    socket_stream.programs:
while True:
    message = input("You:")
    if message.lower() == 'exit':
        break
    s.sendto(message.encode(), (HOST, PORT))
    data server = s.recvfrom(1024)
    print(f"Echo: {data.decode()}")
```

Output.

```
PS python TCP_server.py
server output:
TCP server listening on 127.0.0.1:65432
Client output
Connected to server 127.0.0.1:65432.
type 'exit' to quit
```

Result:

Implementation of chat client & server using TCP/UDP sockets was executed Successfully.

13a - Ping program 29/9/25

```
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.settimeout(2)
        start = time.time()
        s.sendto(b'Ping', (host, port))
        data, end = time.time()
        print(f"Received {data.decode()}
from {addr} in {end-start:.2f}
seconds")
    except socket.timeout:
        print("Request timed out")
```

Experiment 13b

```
import socket

def start_server(host='127.0.0.1',
    port=12345):
with socket.socket(socket.AF_INET,
 socket.SOCK_DGRAM) as s:
    s.bind((host, port))
    print(f"UDP server running on
    {host}:{port}")
    while True:
    data, addr = s.recvfrom
     s.sendto(b'pong', addr)

if __name__ == "__main__":
    start server()
```

Result:

Thus a ping program has been successfully executed using python..

15/10/26

Expt 14 -

```
from scapy.all import sniff

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 == "TCP":
    elif protocol == 17
    protocol_name = "Unknown Protocol"

    Print(f"Protocol: {protocol name}")
    Print(f"Source IP: {src_ip}")
    Print("-" * 50)
```

Result:

Thus packet sniffing has been successfully implemented using Raw sockets