S.No	Date	Title	Page No.	Sign
1	13.7.24	study of various network commands used in Linux & windows	1	
2	27.7.24	Study of Network cables	8	
3	6.8.24	Experiments on CISCO Packet Tracer (Simulation Tool	10	
4	10.8.24	Setting up and Configuring LAN using a switch and Ethernet cable	13	
5.	13.8.24	Experiment on Packet Capture Tool: wireshark	17	
6.	10.9.24	Hamming Code		
7.	11.9.24	sliding window protocol		
8a	9-11-24	Simulate VLAN configuration using CISCO packet Tracer		
8b	9-11-24	~~The~~ Design & configure virtual LAN		
9	9-11-24	Implementation of subnetting in CISCO PACKET TRACER		
10a	9-11-24	Internetworking with routers in CISCO packet tracer		
10b	9-11-24	Design & configure an internetwork		
11a	9-11-24	Simulate Static Routing Protocol		
11b	9-11-24	~~Read~~ Simulate RIP using CISCO packet Tracer		
12a	9-11-24	Implement Echoclient Server using TCP/UDP Socket		
12b	9-11-24			
13	9-11-24	Implement a chat program using socket programming		
14	9-11-24	Implement Ping program		
15	9-11-24	write a code using RAW Analyze various types of packets		

CS19541 – Computer Networks Lab Manual

Exp No 1

Date 7/13/24

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

BASIC NETWORKING COMMANDS:

i) arp -a:

Interface: 172.16.75.34 --- 0x13

Internet Address	Physical Address	Type
172.16.72.1	2C-5A-1C-CF-BC-41	dynamic
224.0.0.2	01-00-5E-00-00-02	static
224.0.0.251	01-00-5E-00-00-FB	static
224.0.0.252	01-00-5E-00-00-FC	static

ii) hostname:

DESTOP-8D

iii) ipconfig /all: Windows IP configuration

Ethernet adapter Ethernet 3:

Media State . . . . . : Media disconnected

connection-specific DNS Suffix . :

Wireless LAN Adapter Local Area Connection* 13:

iv) nbtstat -a:

Displays protocol statistics and current TCP/IP connections using NBT (NETBIOS over TCP/IP).

NBTSTAT [[-a RemoteName] [-A IP Address] [-c] [-n]

v) netstat: Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49676	DESTOP-8D:49677	ESTABLISHED
TCP	127.0.0.1:49677	DESTOP-8D:49676	ESTABLISHED

iv) nslookup:

Default Server: UnKnown
Address: 192.16.7.1

> google.com
Server: UnKnown
Address: 192.16.7.1

v) pathping: usage: pathping [-g host-list] [-h maximum hops] [-i address] [-n] [-p period] [-q num queries] [-w timeout] [-4] [-6] target-name

vi) ping:

i) # ping hostname → ping DESKTOP-8D
Pinging DESKTOP-8D [fe80::8408:8062:da8b:e018
with 32 bytes of data:

ii) # ping ip address → ping 172.16.75.34
Pinging 172.16.75.34 with 32 bytes of data: Reply from 172.16.75.34:

iii) # ping fully qualified domain name
→ ping www.facebook: Pinging star-mini.c10r.facebook.com [157.240.219.35] with 32 bytes of data:

vii) Route:
Manipulates network routing tables.
ROUTE [-f][-p] [-4|-6] command [destination]
[MASK netmask] [gateway] [METRIC metric] [IF interface]
-f clears the routing table

SOME IMPORTANT LINUX NEWORKING COMMANDS

1) ip:- ip [OPTIONS] OBJECT {COMMAND | help}
ip [-force] -batch filename

ip address show
where OBJECT := {link | address | addrlabel | route | rule | neigh | ntable | tunnel | vrf}
OPTIONS = {-V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] | -h[uman-readable] | -iec | -f[amily] {inet |

[illegible] [illegible] [illegible]

ip address show

mtu 65536 qdisc noqueue state

UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

ip address add 192.168.1.254/24 dev enp503

Operation not permitted RTNETLINK answers: File exists

ip address del 192.168.1.254/24 dev enp503

deletes the address specified

ip ~~address~~ link set eth0 up

sets up eth0

ip link set eth0 down

tears down eth0

ip link set eth0 promiscon

sets up ~~eth~~ eth0 promisc on

ip route add 192.168.1.254 dev eth0

adds route to eth0

ip route delete 192.168.1.0/24 via 192.168.1.254

deletes route to eth0

ip route get 10.10.1.4

gets the address 10.10.1.4

i) ifconfig

~~mtu 1500~~

enp0s3: flags=4163< UP, BROADCAST, RUNNING,
MULTICAST > mtu 1500

inet 172.16.8.206 netmask 255.255.252.0

broadcast 172.16.11.255

iii) mtr

My traceroute [V0.87] Tue Jul 16

localhost.localdomain (::)

Keys: Help Display mode Restart statistics order of fields

::1

Packets		Pings				
Loss%	Snt	Last	Avg	Best	Wrst	StDev
0%	124	0.1	0.1	0.1	0.2	0.0

mtr -google.com

localhost.localdomain (0.0.0.0)

Keys: Help Display mode Restart statistics order of fields qu

host

172.16.8.1

static-47.229.249.49-tataidc.co.in

142.250.171.162

142.251.227.213

142.250.228.81

maa05s12-in-f14-1e100.net

mtr - 8 google.com

No DNS support. Sorry.

mtr - b google.com

My traceroute [v0.85]
localhost. localdomain (0.0.0.0)
Keys: Help Display Mode Restart statistics
order of fields quit

Host

1. HP.16.8.1
2. Static-41.229.249.49-tataidc.co.in (49.249.229
3. 142.250.171.162
4. 142.251.227.217
5. 216.5.4.197
6. maa05s10-in-f14.1e100.net (216.58.200.142

mtr -c 10 google.com

localhost.localdomain (0.0.0.0)
Keys: Help Display Mode Restart statistics
order of fields quit

Host

1. 172.16.18.1
2. static-41.229.249-47-tataidc.co.in
3. 142.250.171.162
4. 142.251.227.217
5. 216.5.4.197
6. maa05s10-in-f14.1e100.net

iv) tcpdump

dnf install -y tcpdump

Last metadata expiration check: 0:00:39 ago on
Tue 28 Jul 2020 08:16:48 AM IST.
Package tcpdump-14:4.9.0-2.fc26.i686
already installed, skipping.
Dependencies resolved.
Nothing to do.
Complete!

tcpdump -D

1. enp50 [Up, Running]
2. any (Pseudo-device that captures on all interfaces)
3. lo [Up, Running]
4. wlp350 [Up]

tcpdump -i ~~eth0~~ lo

tcpdump: verbose output suppressed, use -v
or -vv for full protocol decode listening
on lo, link-type EN10MB (Ethernet),
capture size 262144 bytes

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i lo dst 8.8.8.8

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i lo port 53

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i lo host 8.8.8.8 and port 53

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i lo dst 8.8.8.8

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes.

tcpdump -i lo port 53

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i lo host 8.8.8.8 and port 53

tcpdump: verbose output suppressed use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

tcpdump -i eth0 -c 10 port 443

tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes

ping

PING google.com (142.250.182.14) 56(84) bytes of data.
64 bytes from maa05s18-in-f14.1e100.net
time=3.33ms
64 bytes from maa05s18-in-f14.1e100.net
time=3.16ms
64 bytes from maa05s18-in-f14.1e100.net
time=3.15ms

-c 10 google.com

host 10 times

Configuring ethernet connection by using nmcli

1. To create connection profile

NAME UUID TYPE

New 802-3-ethernet connection 7fefdf34-d53-457c-94e4-

Type

802-3-ethernet Device

[student@localhost]

802-ethernet 2. ~]$ nmcli connection add con-name

dest ifname enp

type 802-3-ethernet

connection dest' (619 45158-39990-4186-94

added -eff4164

successfully

3. [student@ localhost ~]$ nmcli connection show

NAME UUID TYPE DEVICE

new 802-3-ethernet connection 7fefdf34-d53-457c-94e4- 802-3-ethernet enp2s0

802-ethernet 93dc82k-260-4444-27e7-762546061

dest 619457 5e-3990-44586-9a3a-eff41

RESULT:-

Thus the study of various network commands used in Linux & windows

Exp 8: STUDY OF DIFFERENT TYPES OF NETWORK CABLE

Date : 21.1.25

a. understand different types of network cables

Different type of cable used in networking are

1. Unshielded Twisted Pair (UTP) cable.
2. Shielded Twisted Pair (STP) cable.
3. coaxial cable
4. Fibre optic cable.

cable type	category	Max Data Transmission	Advantages/ Disadvantages	Application
UTP	category 3	10bps	advantages • cheaper in cost • Easy to install as they have a smaller overall diameter Disadvantage • More prone to EMI interference & noise	10 Base - T ethernet
	category 5	upto 100 Mbps		Fast ethernet gigabit,
	category 5e	1 Gbps		Fast ethernet, gigabit
	category 6	10 Gbps	Advantages • Shielded • Faster than UTP • less susceptible to noise	gigabit ethernet 10 GBase (100m

Coaxial cable	RG-6 RG-59 RG-11	10-100 mbps	Advantages • High bandwidth • Immune to interference Disadvantages • limited distance • cost • size is bulky	Used in Television signal internet
fibre optics cable	Single mode multi mode	100 Gbps	Advantages • Highspeed • High bdw • High security Disadvantages • expensive • Requires skills installers	Maximum distance of fibre optics cable around 100 m

b. Make your own ethernet / cross-over cable / straight cable

Tools and parts needed:

• Ethernet cabling : CAT5e is certified for gigabit support, but CAT5 cabling works as well just over shorter distances.

• A crimping tool : This is an all-in-one networking tool used down the pins the plug & strip & cut the shielding off the cables.

• Optional two plug shields

Step 1: To start construction of cable, begin by threading shields onto the cable.

Step 2: Next, strip approximately 1.5 cm from both ends the of cable shielding. Crimping tool has a round area to complete this task.

Step 3: After you will need to untangle the wires, there should be four twisted pairs referring back to the sheet, arrange them top to bottom.

Step 4: Once the order is correct, bunch them together in a line and if there are any stick and farther than others strip them back to create an even level. The difficult aspect is placing these into the RJ45 without messaging up the order. To, do so, hold so on.

Step 5: Next push the cable right in the notch, at the end of the plug needs to be over the cable shielding.

Step 6: After the wires are successfully securely sitting inside the plug, insert into crimping tool.

Step 7: Lastly, repeat for the other end using diagram B.

Straight cable :- All the wires are in the same order on both ends of the cable

Cross cable :- The transmit and receive wires are swapped on each end

2. which type of cable is used to connect a router / switch to your PC ?

Straight cable is used to connect PC to a computer

3. which type of cable is used to connect two PCs.

• Cross cables are used for PC-PC connection

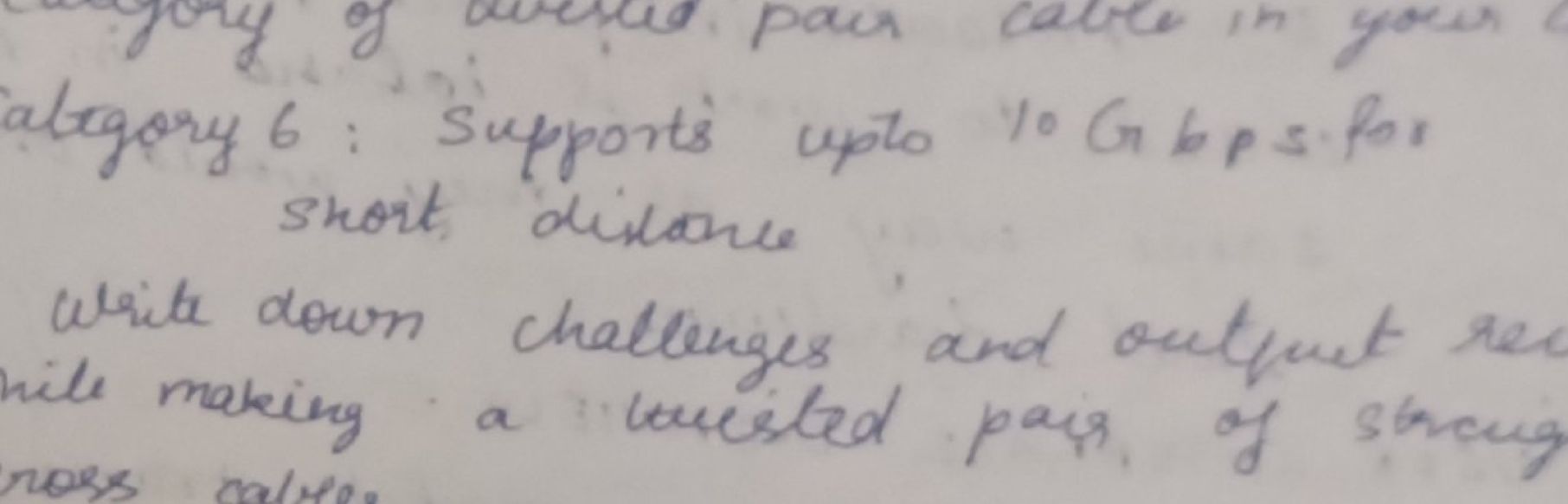
4. Category of twisted pair cable in your lab.

Category 6 : Supports upto 10 Gbps for short distance

5. Write down challenges and output received while making a twisted pair of straight / cross cables.

challenges faced :- crimping issues, wire order, testing.

output received :- Successfully made cable



RESULT:- Thus the networks cables are studied and their different types have been verified

Practical 6

Student observations

1. What is the difference between straight cable and cross cable?

Straight cable :- All the wires are in the same order on both ends of the cable

Cross cable :- The transmit and receive wires are swapped on each end

2. Which type of cable is used to connect a router / switch to your PC?

Straight cable is used to connect PC to a controller

3. Which type of cable is used to connect two PCs

• cross cables are used for PC - PC connection

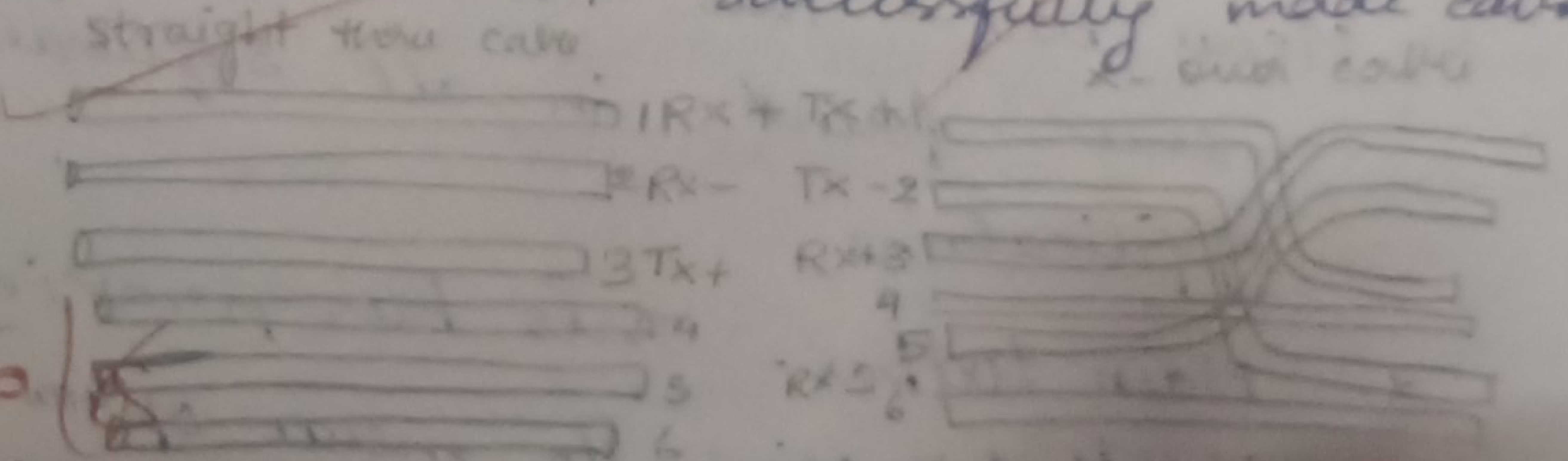
4. Category of twisted pair cable in your lab.

Category 6 : Supports upto 10 Gbps for short distance

5. Write down challenges and output received while making a twisted pair of straight / cross cables.

challenges faced :- crimping issues, wire order, testing

output received :- successfully made cable



[illegible]

RESULT :- Thus the network cables ~~are studied~~ and their different types have been verified

EXP 3

AIM : TO STUDY THE PACKET TRACER AND USE ITS INTERFACE COMPONENT

DATE : 6.8.24

c) To understand environment of CISCO packet tracer to design simple network.

1. It allows you to model complex systems out the need for dedicated equipment.

2. It helps you to practice your network configuration and trouble shooting skills

3. It is available for both Linux and windows desktop environment.

4. Protocols in packet tracer are coded to work & behave in the same way as they would on real hardware

USER INTERFACE OVERVIEW :-

The layout of packet tracer is divided into several components

1. Menu bar - This is a common menu found in all Software applications

2. Main toolbar - This bar provides shortcut wins to menu option.

that are currently activated in the network.

3. Logical / Physical workspace - These tabs allows you to toggle between the logical & physical workspace.

4. Workspace - This is area wherein topologies are related & simulation are displayed

5. common toolbar - This toolbar provides control for manipulating topologies such as select, move, layout, place, delete.

6. Real-time / Simulation mode - These tabs are used to toggle between real & simulation model.

7. Network component box - This component contains all network & end devices available with packet tracer.

8. User created packet box - User can create highly customized packets to test their topology from this and the results are displayed as a list.

6 Design how HUB & switch are forwarding the data & write the difference & observation of the behaviour of the switch and HUB.

Practical Observation

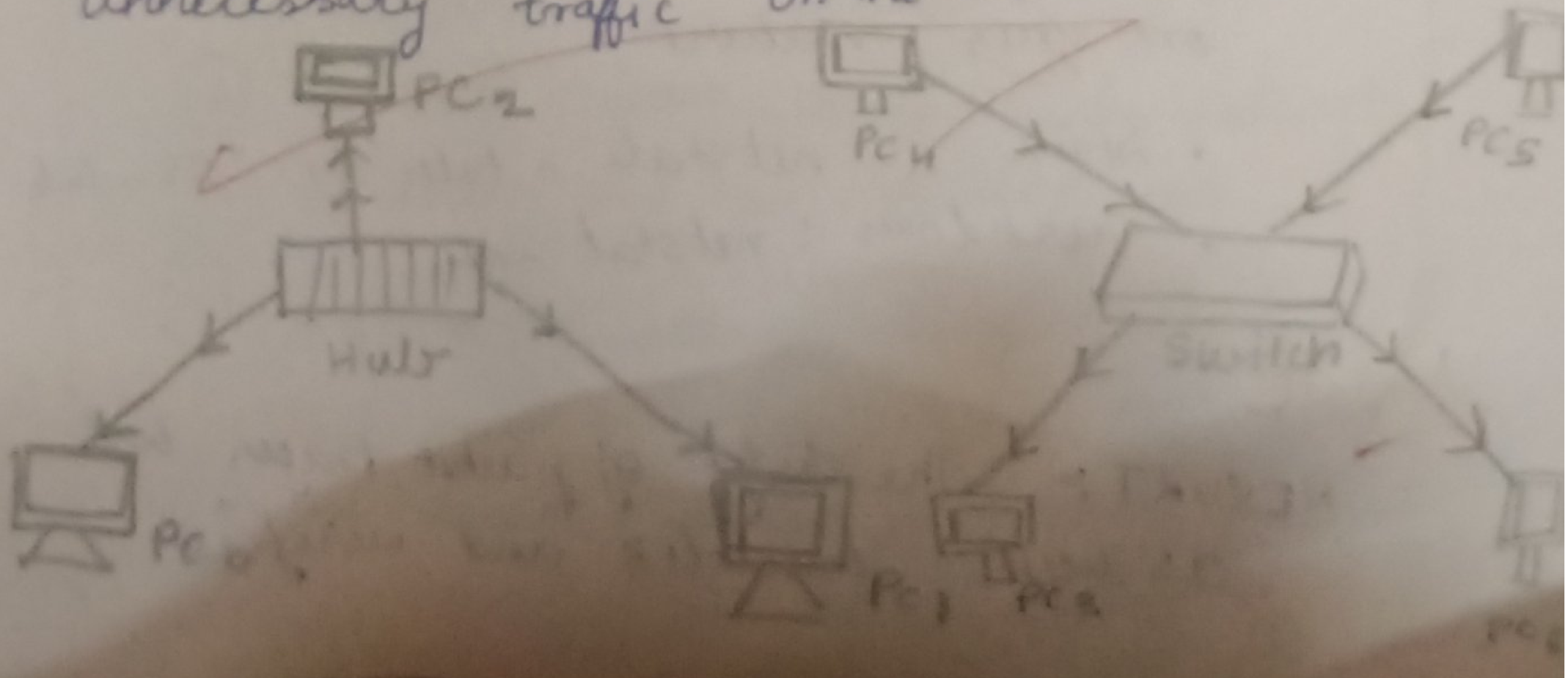
From your observation write down behaviour of switch and HUB in terms of forwarding the packets received by them

HUB

- Broadcasting :- A hub is a basic networking device that operates at the physical layer (layer 1) of the OSI model
- Efficiency : Less efficient, as it increases unnecessary traffic.

SWITCH :-

- Packet Forwarding : Uses MAC addresses to forward packets only to intended recipients
- Efficiency : More efficient as it reduces unnecessary traffic on the network



Student information : practical 1.

1. To find the reachability of a host machine from your device
command : ping <hostname - or - IP address>

2. To get the details of hops taken by a packet to reach M's destination
Command : traceroute <host name - or - IP address>
(on linux) or tracert <host name - or - IP_address>
(on windows)

3. To display the IP configuration of your machine
- command (Linux) : ifconfig or ipa
- windows : ipconfig

4. To display the TCP port status on your machine
- Linux : 'netstate - tuln' or 'ss - tuln'
- windows : netstat -an

20/8

RESULT :- The study of packet tracer tool as been executed and verified

Date: 10.9.24 Practical - 14

Aim

Setup and configure a LAN (Local area network) using a switch and ethernet cable & on your desk

What is LAN?

A local Area Network (LAN) refers to a network that connects devices within a limited area such as office building, school or home. It enables users to resources including data, printers and internet access

How to setup LAN

(1) - plan and design of an appropriate network topology taking into account network requirements and equipment location

(2) - take 4 computers, a switch with 8, 16 or 24 ports which is sufficient for network of these sizes and 4 Ethernet cables.

(3) Connect your computers to network Switch via an Ethernet cable

(4) Assign IP address to your PCs log on to the client computer as administrator or as owner

↳ log on to the client computer as administrator or as owner

↳ Right click local area connection ethernet → go to properties → select Internet protocol (TCP IP v4)

↳ click on properties → select the following ip address option and assign ip address

5) configure a network switch :

→ Connect your computer to the switch to switch's web interface, you will need to connect to your computer to the switch using an ethernet cable.

↳ log in to the web interface

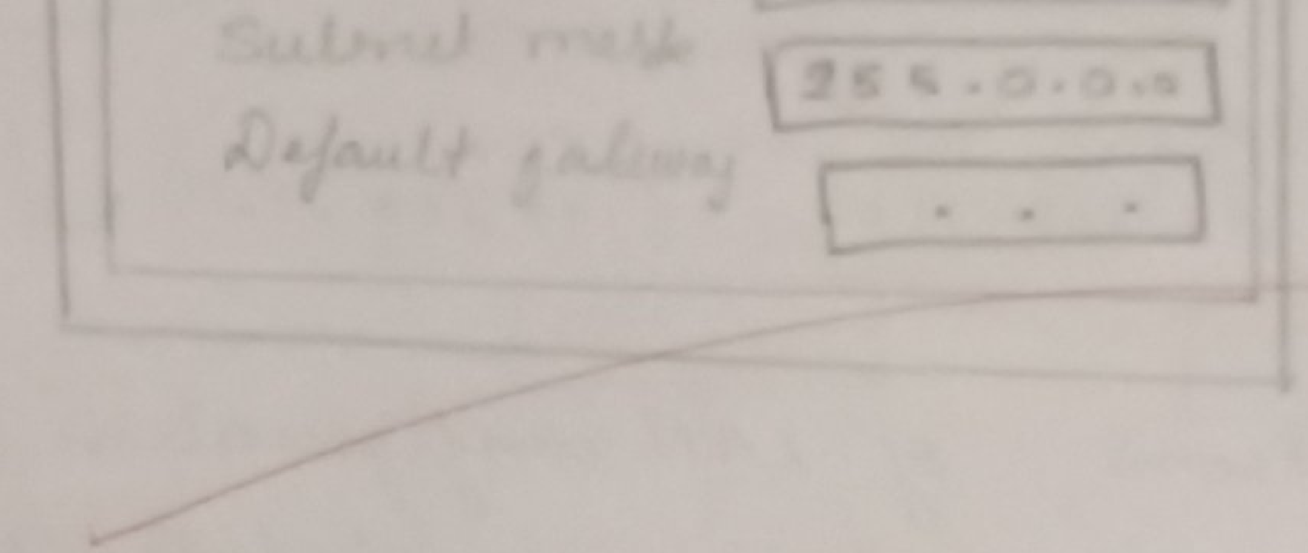
↳ configure basic settings

↳ assign IP address as 10.1.1.5, subnet mask 255.0.0.0

6) check the connectivity between switch and other machine by using ping command in the command prompt of device

7) select a folder go to properties → click sharing tab → share it with everyone on the same LAN

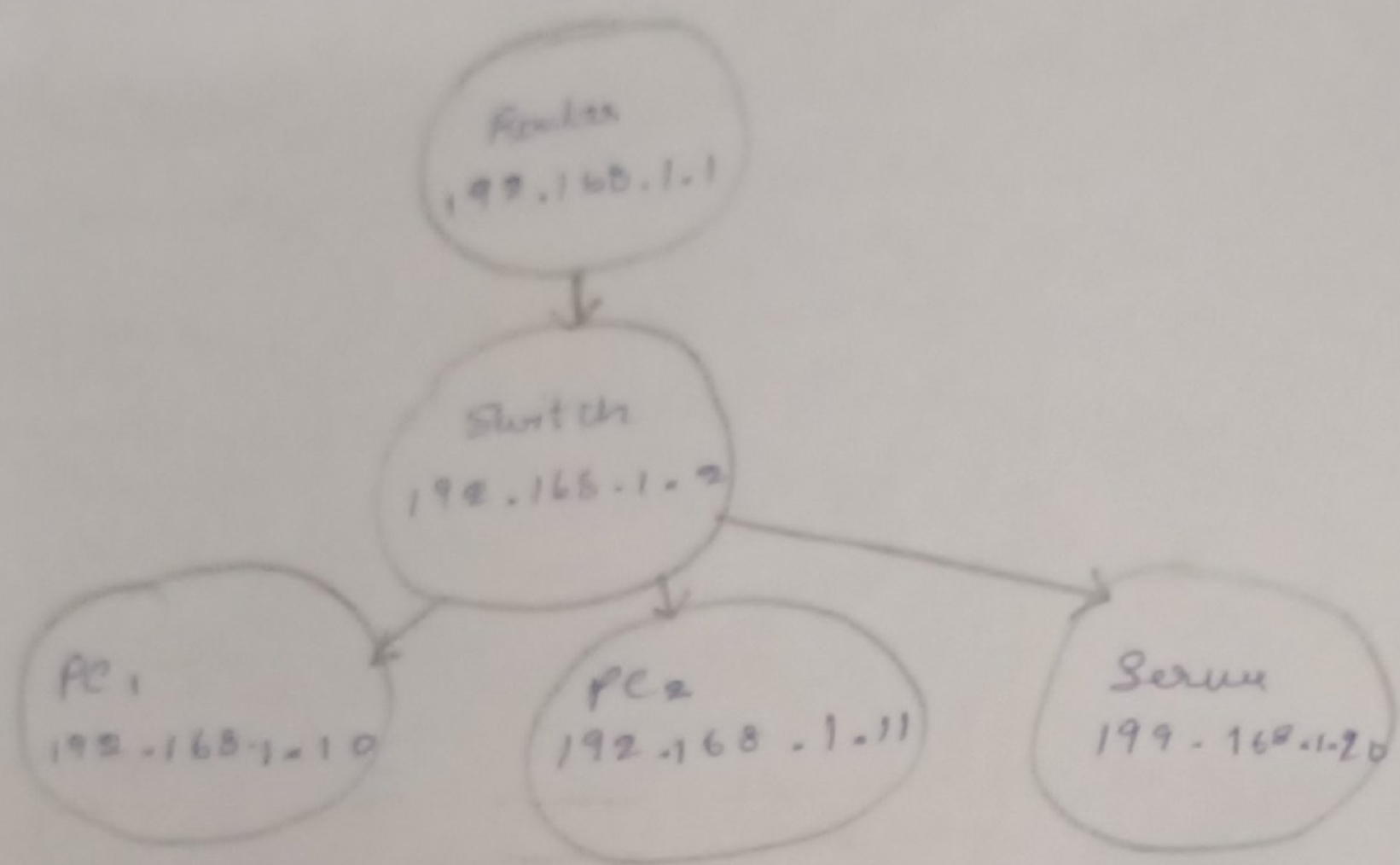
8) → my to access the shared folder & from other computers of the network



RESULT:- The experiment for setup and configuring a LAN has been verified.

Student Observation 14

Draw a neat diagram of new LAN in the configuration observation task. write IP configuration of each of every device. write an outcome and challenges faced while configuration the LAN.



outcome of LAN configuration

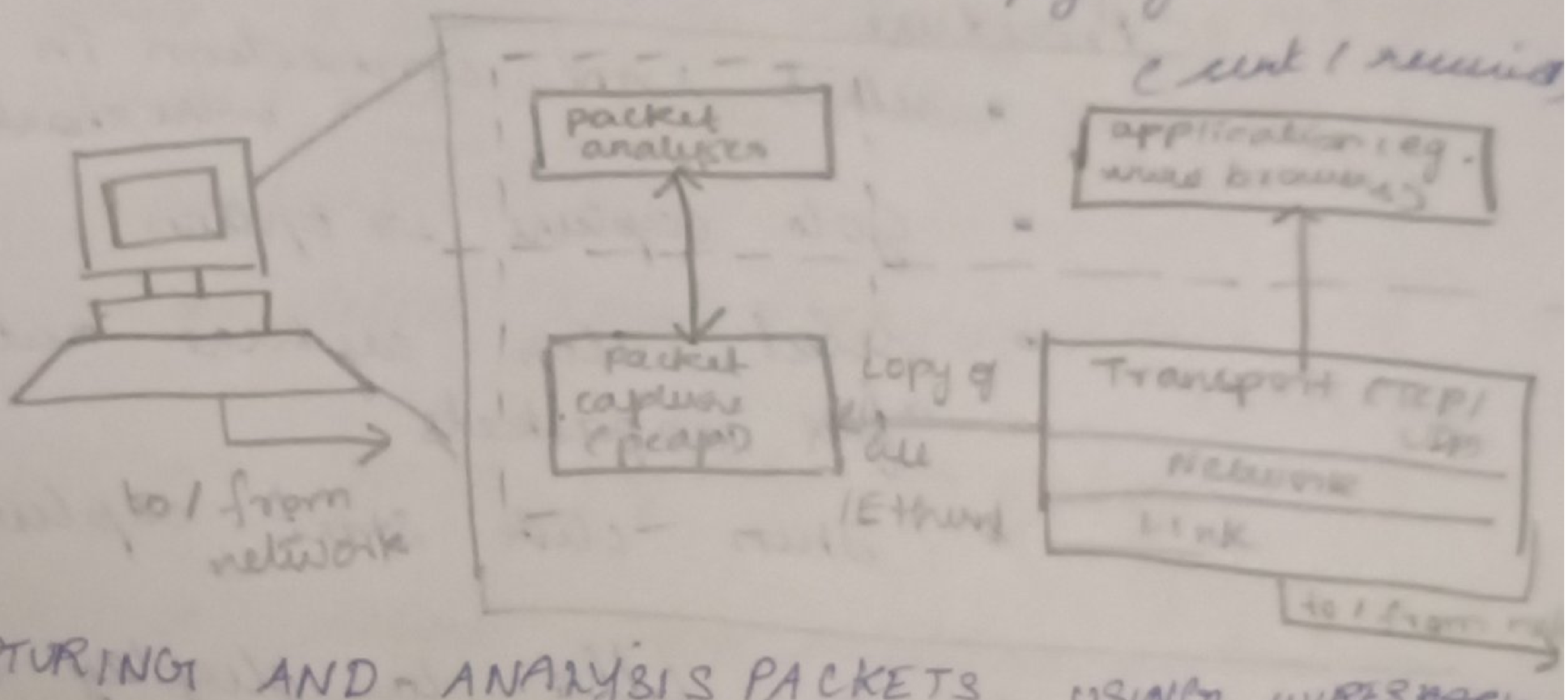
- Successful communication between all connected devices
- Proper IP assignment, ensuring no IP conflicts
- Verified connectivity using tools like 'ping' to test the connection between devices

~~Internet access~~

challenges faced :- IP address conflicts, Connectivity issues, Subnetting errors, Firewall settings.

Packet Sniffer - Sniffs messages being sent / received from / by your computer

- Store the display the contents of various protocol fields in the messages
- Passive program
  - never sends packets itself
  - no packets addressed to it
  - receives a copy of all packets sent / received



CAPTURING AND ANALYSIS PACKETS USING WIRESHARK TOOL

To filter capture, view, packets in wireshark tools. Capture 100 packets from the Ethernet : IEEE 802.3 LAN Interface and save it.

edure

elect Local Area Connection in

wireshark

Go to capture → option

Select Stop capture automatically

after 100 packets

Then click Start capture

Save the packets

create to filter only TCP/UDP

packets, inspect the packets and

provide flow graph.

Procedure

- select LAN connection in wireshark
- goto capture → option
- select Stop capture automatically after 100 packets
- Then click Start capture

2. Create a filter to display only ARP packets and inspect the packets

- go to capture → option
- Select stop capture automatically after 100 packets
- Then click Start capture
- Search ARP packets in search bar.
- Save the packets

3. Create a filter to display only DNS packets and provide the flow graph.

Procedure :-

- Go to capture → option
- Select stop capture automatically after 100 packets
- Then click Start capture
- Search DNS packets in search bar.
- To see flow graph click Statistics
- Save the packets.

4. Create a filter to display only HTTP packets and inspect the packets

Procedure

- Select Local Area Connection in wireshark
- Go to capture → option
- Select stop capture automatically after 100 packets
- Then click Start capture.
- Save the packets

5. Create a filter to display only IPI ICMP packets.

Proadure .. • Go to capture
- Then click Start capture
- Save the packets

8. create a filter to display only DHCP packets and inspect the p

Procedure

- Select Local Area Connection in wireshark
- go to capture → option
- Then click start capture
- Save the packets

20/8

RESULT :- The experiment on packet capture tool : wireshark as been verified.

Student Observation :-

1) What is promiscuous mode ?

This promiscuous mode is a configuration for a network interface that allows it to capture all packets on the network segment it is connected.

2) Does ARP packets have a transport layer header ? Explain.

No, ARP packets do not have a transport layer header because it operates at data link layer of the OSI model.

3) Which transport layer protocol is used by DNS ?

DNS primarily uses UDP as it's transport protocol but it also uses TCP for layer responses

4) What is the port number used by protocol ?

The default port number used by HTTP is 80. For HTTPS the secure version of default port is 443.

5) What is a broadcast IP address ?

It is a special address used to send all devices on a specific network or subnet.

P.NO:6
DATE: 31-08-24

AIM: write a program to implement error detection and correction using HAMMING CODE concept.

Error correction at Datalink layer:

Hamming code is a set of error-correction codes that can be used to detect the errors that can occur when the data is transmitted from the sender to the receiver. It is a technique developed by R.W Hamming for error correction.

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. Apply hamming code concept on the binary data and add redundant bits to it.
3. Save this output in a file called channel

Create a receiver program with below features

1. Receiver program should read
2. Apply hamming code on the binary
3. If there is an error, display the position of the error

4. Else remove the redundant bits and convert the binary data to ascii and display the output.

Student observation :-

```
import java.util.*;
public class HammingCode {
    public static void generatehammingcode
            ( int[] data, int [] code) {

        code [2] = data [0];
        code [4] = data [1];
        code [5] = data [2];
        code [6] = data [3];

    int code [0] = data [0] ^ data [1] ^ data [3];
        code [1] = data [0] ^ data [2] ^ data [3];
        code [3] = data [1] ^ data [2] ^ data [3];
    }
    public static void detectAndCorrectError (int
                                              [] code)
        int p1 = code [0] ^ code [2] ^ code [4] ^
                                          code [6];
        int p2 = code [1] ^ code [2] ^ code [5] ^
                                          code [6];
        int p4 = code [3] ^ code [4] ^ code [5]
                                          ^ code [6];
    int errorPosition = p1 * 1 + p2 * 2
                                + p4 * 4;
    if (errorPosition == 0)
            System.out.println ("No error detected");
    else
        System.out.println (" Error detected at position: "
                                          + errorPosition);
    code [errorPosition - 1] = code [errorPosition - 1] ^ 1;
```

~~Input:~~

System.out.println("corrected code:")
for (int bit : code)
{ System.out.print(bit);
}
~~System.out.println~~

}
}

~~output:~~

```
public static void main(String [] args) {
    Scanner scanner = new Scanner(System.in);
    int [] data = new int [4]
    int [] code = new int [7];
System.out.println("Enter 4 data bits (0 or 1):");
for(int i=0; i<4; i++) {
        data[i]=scanner.nextInt();

}
generatehammingcode(data, code);
system.out.println("Generated hamming code");
for(int bit: code) { System.out.print(bit); }
system.out.print("Enter position of error:");
    int errorbit = scanner.nextInt();
    if(errorbit != 0) { code[errorbit-1] = code[errorbit-1];
        system.out.println("Code with error introduced:")
    for(int bit: code) { System.out.println(bit);
}}
detectAndCorrectError(code)
```

RESULT:- The experiment to ~~verify~~ write hamming code is executed.

Enter the text to be corrected: Deepthi

Text to Binary : 01000100 1101 0001 11 0 101 0 1110 01
00

Redudant bit : 7

Redudant bit position : 1 2 4 8 16 32 64

Parity bits : parity (1:1), parity (2:1),
parity (4:1), parity (8:1), parity (16:1)
parity (32:0), parity (64:1)

Sender output: 11011001 00100111010001 11
01010110100

Introduced error at position: 2

Binary with error : 10011001001 0011 0100
1110101011011100

Error detected at position: 2

Error corrected at position : 2

Binary after error correction : 11011001001
1110101011011 0001000010

Decoded text : Deepthi

RESULT: Experiment for learning code is verified

EXP.NO:7

DATE: 31-8-24

AIM:- WRITE A PROGRAM IMPLEMENT FLOW CONTROL AT DATA LINK LAYER USING SLIDING WINDOW PROTOCOL

Program should achieve at least below given requirements. You can make it a bidirectional program wherein reciver is sending it's data frames with acknowledgement (Piggybacking)

Create a sender program with following features:-

1. Input Window size from the user.
2. Input a Text message from user.
3. consider 1 character per frame.
4. Create a frame with following fields

[Frame no. Data]

5. Send the frames.
6. Wait for acknowledgement from Reciver
7. Reciver_Buffer file is called
8. check ACK field for acknowledgement number.
9. If the acknowledgement number is as expected, send new set of frames accordingly.

create a reciever file with following features

1. Reader a file called Sender_Buffer
2. Check the Frame no.

```
System.out.print("Enter the window size: ");
int windowSize = scanner.nextInt();
scanner.nextLine();
System.out.print("Enter message: ")
String message = scanner.nextLine();
int frameCount = message.length();
int nextFrameToSend = 0;
int ackExpected = 0;
while (ackExpected < frameCount) {
  { int framesSent = 0;
      for (int i = nextFrameToSend; i< nextFrameT
          + windowSize && i < frameCount; i++)
                                          Sen
      sendFrame(i, message.charAt(i));
      framesSent ++;
  }
nextFrameToSend += framesSent;
if (ackReceived) {
      ackExpected ++;
  } else {
      System.out.println("Time out
or NACK received. Resending frames starting
from " + ackExpected);
      nextFrameToSend = ackExpected;
}}
System.out.println("All frames sent successfully");
}
```

```
public static void (String args[])throws IOException{
    while (true) {
        boolean frameReceived = false;
    try ( Buffered reader= new BufferReader (new
                    FileReader (SENDER-BUFFER)
    String line;
    while ((line = reader.readline ()) != null) {
        int frameNo= Integer.parseInt (line.
                        indexOf(';'));
      if (frameNo == nextFrameExpected) {
    System.out.println ("Frame" + frameNo+
                    " received correctly.");
       nextFrameExpected ++;
       sendResponse ( "ACK"+ frameNo);
       frameReceived = true;
     } else { system.out.println ("Frame"+ fram
              + "out of order. Expected :"+
    nextFrameExpected ); sendResponse ("NACK"+
    nextFrameExpected );
    }    break;
} }
if ( !frameReceived ) { system.out.println ("No
        }             new frames received");
  try { Thread.sleep (2000); } catch (Interrupted
  Exception e) { e.printStackTrace (); } } }
  private static void sendResponse (String response)
throws IOException { System.out.println ("sending"
                        + response);
    [illegible] writer = new BufferedWriter
                        ((new FileWriter (RECEI
                            _BUFFER
                        + [illegible]
```

Sending Frames OUTPUT

Sent Fr.
Enter window size: 3
Enter a message to send: deepthi

Sending frames :-
sent frame 0 : d
sent frame 1 : e
sent frame 2 : e
waiting for ack
ack received
sent frame : p
sent frame : t
sent frame : h
waiting for ack
ACK not received
resending frames
sent frame : p
sent frame : t
sent frame : h
waiting for ack
ack received
sent frame : i

Receiving frame :-
Received frame 0 : d [OK]
Received frame 1 : e [OK]
Received frame 2 : e [OK]
Received frame 3 : p [OK]
Received frame 4 : t [OK]
Received frame 5 : h [error]

- Resending frames --
Received frame 3 : p [OK]
Received frame 4 : t [OK]
Received frame 5 : h [OK]
Received frame 6 : i [OK]

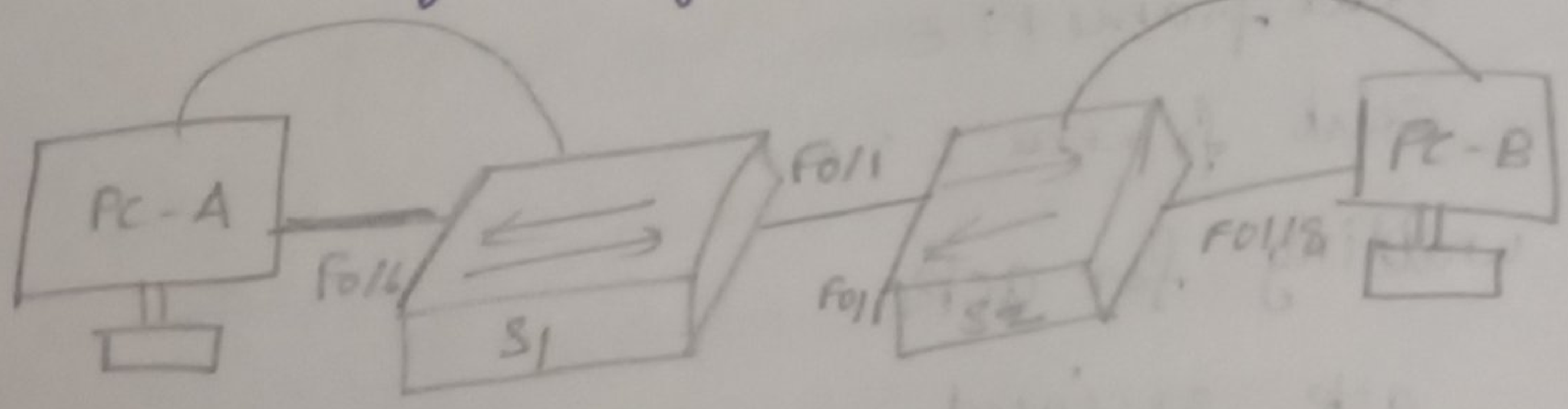
5/10

RESULT :- Experiment for sliding window protocol

Practical 8

AIM:- a) Stimulate Virtual LAN configuration using CISCO packet tracer Simulation.

Packet Tracer - Configure VLANS and Trunking - Physical Mode Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 1	192.168.1.11	255.255.255.0	N/A
S2	VLAN 1	192.168.1.12	255.255.255.0	N/A
PC-A	NIC	192.168.10.3	255.255.255.0	PC-
PC-B	NIC	192.168.10.4	255.255.255.0	192

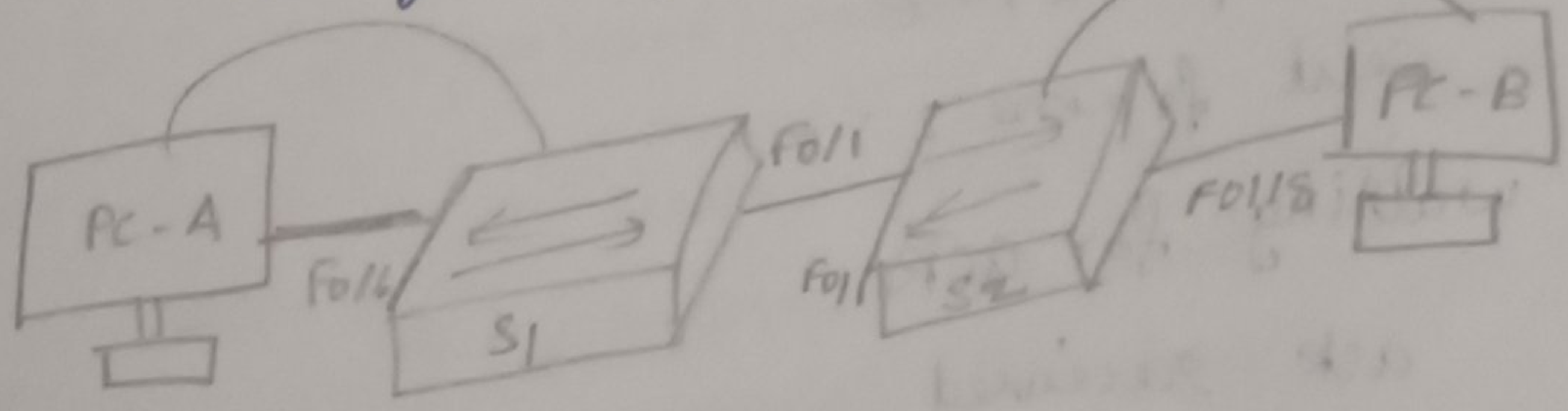
Build Network & Configure Basic Device Settings

Step 1 :- Build the network as shown in the topology

Practical 8

AIM:- a) Stimulate Virtual LAN configuration using CISCO packet tracer Simulation.

Packet Tracer - Configure VLANS and Trunking - Physical Mode Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 1	192.168.1.11	255.255.255.0	N/A
S2	VLAN 1	192.168.1.12	255.255.255.0	N/A
PC-A	NIC	192.168.10.3	255.255.255.0	PC-2
PC-B	NIC	192.168.10.4	255.255.255.0	192.

Build Network & Configure Basic Device Settings

Step 1 :- Build the network as shown in the topology

1) Assign PC-A to the operation VLAN.

S1(config)# interface fo1/6

S1(config-if)# switchport mode access

S1(config-if)# switchport access vlan 10

2) From VLAN 1, remove the management IP address & configure it on VLAN 99

b) Issue the show vlan brief command and verify that the VLANs are assigned to the correct interfaces.

d) Assign PC-B to the operations VLAN on S2

e) From VLAN 1, remove the management IP address and configure it on VLAN 99 according to the addressing table

PART 3 - Maintain VLAN Port assignments and VLAN Database

a) Add VLAN 30 to interface Fo1/24 without issuing the global VLAN command.

S1(config)# interface fo1/24

b) Verify that the new VLAN is displayed in the VLAN table

c) Use the no vlan 30 command to remove VLAN 30 from VLAN database

d) Use the no vlan 30 command to remove VLAN 30 from VLAN database → S1(config)# no vlan 30
S1(config)# end

Step 4 - Configure an 802.1Q Trunk Between the Switches

You will configure interface Fo 1/1 to use the Dynamic Trunking Protocol (DTP) to allow it to negotiate the trunk mode.

Step1: Use DTP to initate trunking on Fo 1/1.

Step2: Manually config trunk interface Fo/1

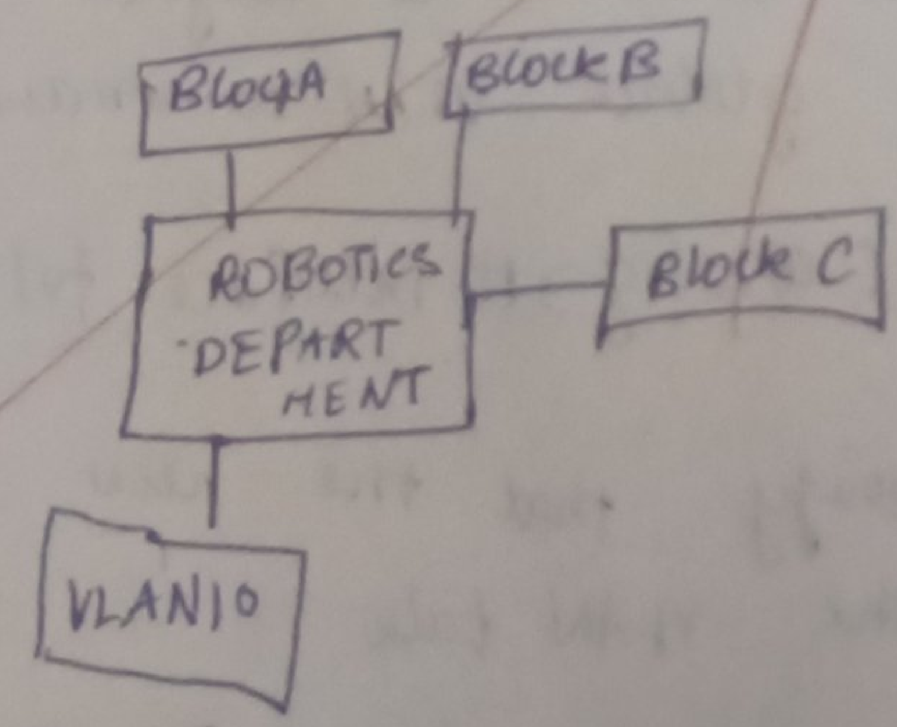
Student Observation

Part a) Draw and label the VLAN

- Network Topology :-

i) There are three different blocks where the faculty are located

ii) We will connect all blocks to a central switch

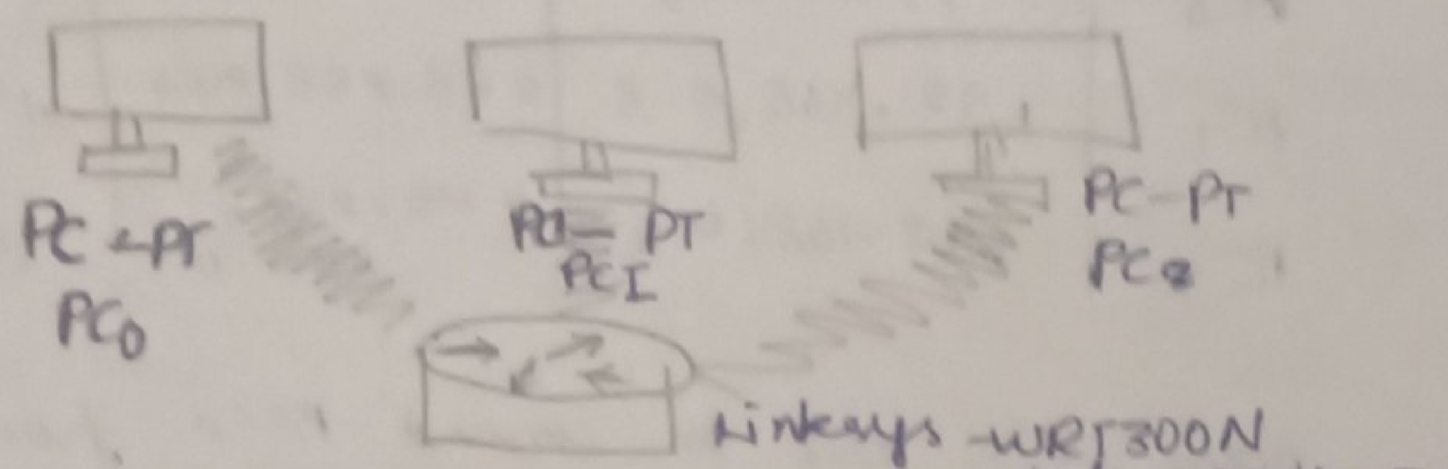


RESULT: Simulating VLAN configuration using CISCO packet tracer is executed and verified

Practical 8 - b

AIM :- b) Configuration of wireless LAN using CISCO Packet Tracer.

Design a topology with three PCs connected from Linksys wireless routers



Perform following configuration:-

- configure static IP on PC and wireless Router
- set SSID to MotherNetwork
- Set IP address of router to 192.168.0.1 PC0 to 192.168.0.3 and PC2 to 192.168.0.4
- Secure your network by configuring WAP key on Router
- Connect PC by using WAP key.

Step 1:- Click on wireless router

- select administration tab from top menu, set username & password to admin & click on Save settings.
- Next click on wireless tab and set default SSID to MotherNetwork
- Set Key1 to 0123456789

- Repeat same process on PC_1 & PC_2

Student Observation

c) what is a SSID of a wireless router ?

SSID Service set Identifier is the name of a wireless network. it is the unique identifier that allows devices to connect to a specific WIFI - network among the many that might be available in the same area.

d) What is a security key in a wireless router ?

A Security key is a password that secures wireless network. It is required for a device to connect to the WIFI network, ensuring that only authorised users can access the network.

- WEP - Wired equivalent privacy
- WPA - Wi-Fi Protected Access
- WPA2 (Wi-Fi Protected Access 2)

e) Configure a simple wireless LAN using a real access point

- Connect to the access point.
- Log in to the access point
- Configure the SSID
- set the security tape
- Channel Selection
- save the configuration
- Test the WLAN connection

Result :- Configuration of VLAN using cisco packet tracer is executed & verified

implementation of SUBNETTING
ACKET TRACER simulator.

classless IP subnetting is a techn
that allows for more efficient use
IP addresses by allowing for subnet m
that are not just the default ma
for each IP class

CREATING A NETWORK TOPOLOGY:-
The first step in implementing class
IP subnetting is to create a network
topology in Packet Tracer.

ADDING THE DEVICES:-
Once we have created our network
topology in Packet Tracer. To create
a network topology in Packet Trac
select the "New" button in the top
left corner

SUBNETTING:- To subnet the network
address 192.168.1.0/24 to provide
enough space for at least 5 addres

STUDENT OBSERVATION:-

a) write down your understanding of subnetting

Subnetting is the process of dividing a large network into smaller, more manageable subnetworks or subnets.

b) what is the advantage of implementing subnetting within a network?

Implementing subnetting within a network provides several advantages

i) Efficient IP address management
ii) Reduced Network Traffic
iii) Enhanced security
iv) improved network performance

c) Find out where subnetting is implemented in your college

i) check the network configuration
ii) analyse the IP address & subnet mask
→ if the subnet mask is not 255.255.255.0 (which indicates a standard class C network), subnetting is likely in use
iii) contact the network administrator
iv) document the subnets

RESULT:- The experiment for implementation of subnetting in CISCO Packet Tracer Simulator is executed & verified

Practical 10

AIM:- a) Internetworking with routers in CISCO PACKET TRACER simulator

b) Design and config a simple internetworking using a router.

In this network, a router and 2 PCs are used. Computers are connected with routers using a copper straight-through cable. After forming the network, to check network connectivity a simple PDU is transferred from PC_0 to PC_1

Procedure :-

Step-1 [configuring Router 1]:

1. Select the router and open CLI.
2. Press ENTER to start configuring Router 1
3. Type enable to activate the priviledged mode.

Router 1 Command line Interface

```
Router > enable
Router # config t
Enter configuration commands, one per line.
End with CNTL/Z
Router (config) # interface FastEthernet 0/0
Router (config-if) # ip address 192.168.10.1
                              255.255.255.0
Router (config-if) # no shutdown
Router (config-if) #
```

PC configuration table

Device name	IP address	Subnet mask	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.20.2	255.255.255.0	192.168.20.1

Designed network topology :-

i) Sending PDU from PC0 to PC1

ii) Acknowledgement from PC1 to PC0

9/11

RESULT:- The experiment for internetworking with routers in CISCO PACKET TRACER simulator

PC configuration table

Device name	IP address	Subnet mask	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.20.2	255.255.255.0	192.168.20.1

Designed network topology :-

i) Sending PDU from PC0 to PC1

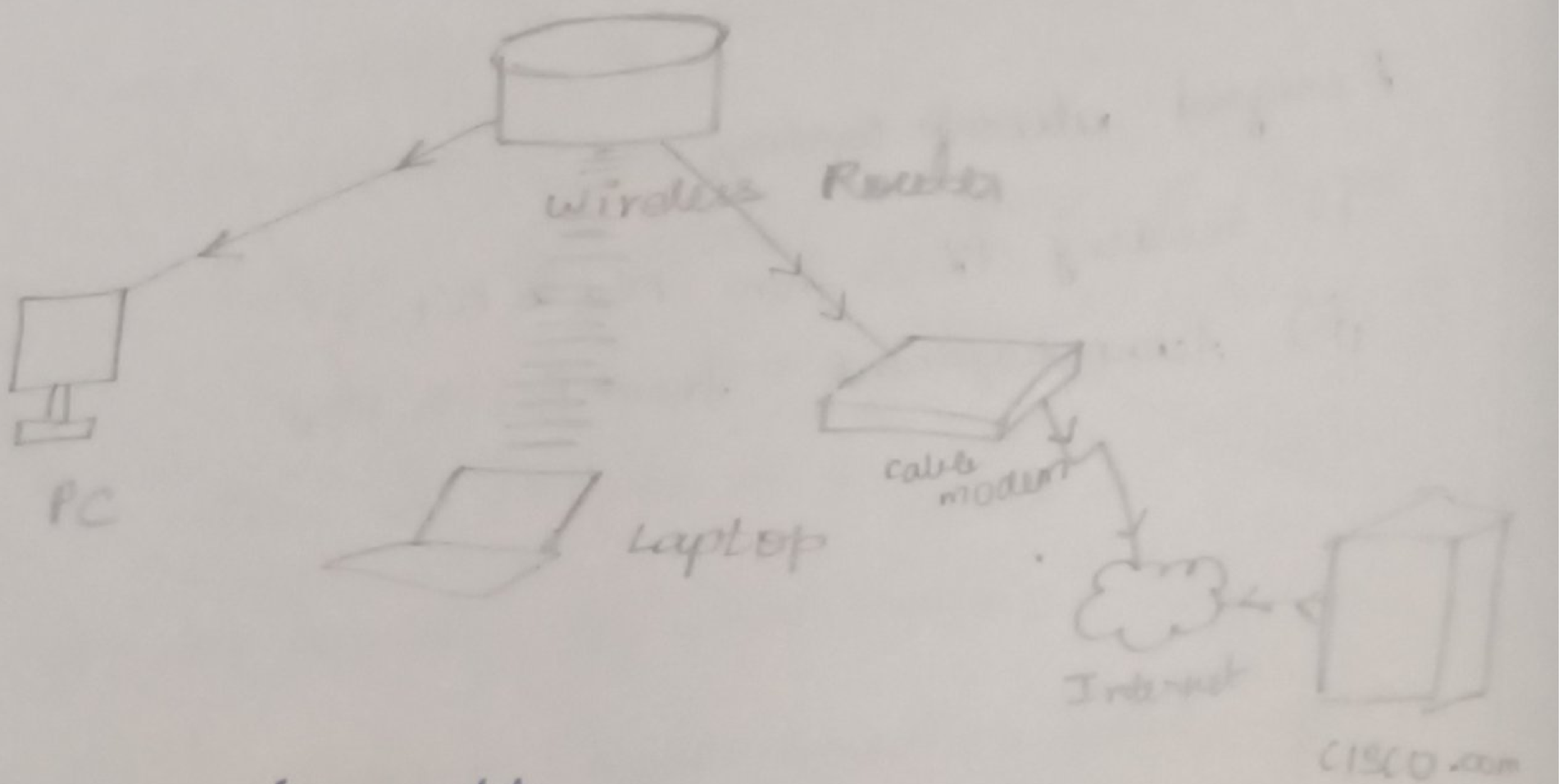
ii) Acknowledgement from PC1 to PC0

9/11

RESULT:- The experiment for internetworking with routers in CISCO PACKET TRACER Simulator

PRACTICAL 10

AIM:- b) Design and configure an internetwork using wireless router, DHCP server and internet cloud.



Addressing table

Device	Interface	IP address	subnet mask	Default gateway
PC	Ethernet 0	DHCP		192.168.0.1
wireless Router	LAN	192.168.0.1	255.255.255.0	
wireless Router	Internet	DHCP		
CISCO.COM Server	Ethernet 0	200.67.220.220	255.255.255.0	
Laptop	wireless 0	DHCP		

Objectives

Part 1: Build a simple network in logical topology workspace

Part 2: Configure the network devices

Part 3: Test connectivity between network devices

Step 3 : Configure the PC

a. Configure the PC for the wired network

click on the PC icon on the Packet Tracer logical workspace & select the Desktop tab & the IP configuration icon

Step 4: Configure the internet cloud

a) Install network modules if necessary

b) Identify the From and To ports

c) Identify the type of provider

Step 6:- Configure the cisco.com server

a) Configure the cisco.com server as a DHCP server

b) Configure the Cisco.com server as a DNS server to provide domain name to IPV4 address resolution

c) Configure the Cisco.com server Global settings

d) Configure the cisco.com server FastEthernet 0 Interface Settings

Part 1: Refresh the IPv4 settings on the PC

a) Verify that the PC is receiving IPv4 configuration information from DHCP

b) Test connectivity to the Cisco.com server from the PC

Student Observation:

1. Key features of the configuring wireless Router and DHCP Server:

Wireless Router:

- SSI configuration: Set the network name for easy identification.
- Security Settings: Enable WPA2 or WPA3 encryption for secure connections.
- Access control: Keep to Manage who can connect to the network by MAC filtering

DHCP Server:

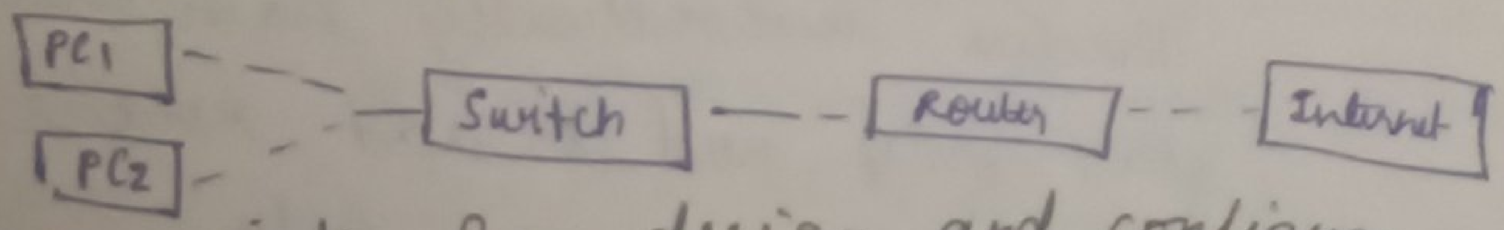
- Dynamic IP Assignment: Automatically assigns IP addresses to devices to the network
- Lease Time Management configures how long an IP address is reserved for a device

2. Significance of DHCP Server in Internetworking:
→ Simplified Reduce the need for manual IP configuration. Dynamic IP allocation → efficiently manages IP addresses in a network

3. Design and configuration of Inter-network

Inter-network Design → Components: Switch, Router, and Ethernet cables connecting devices

Topology example

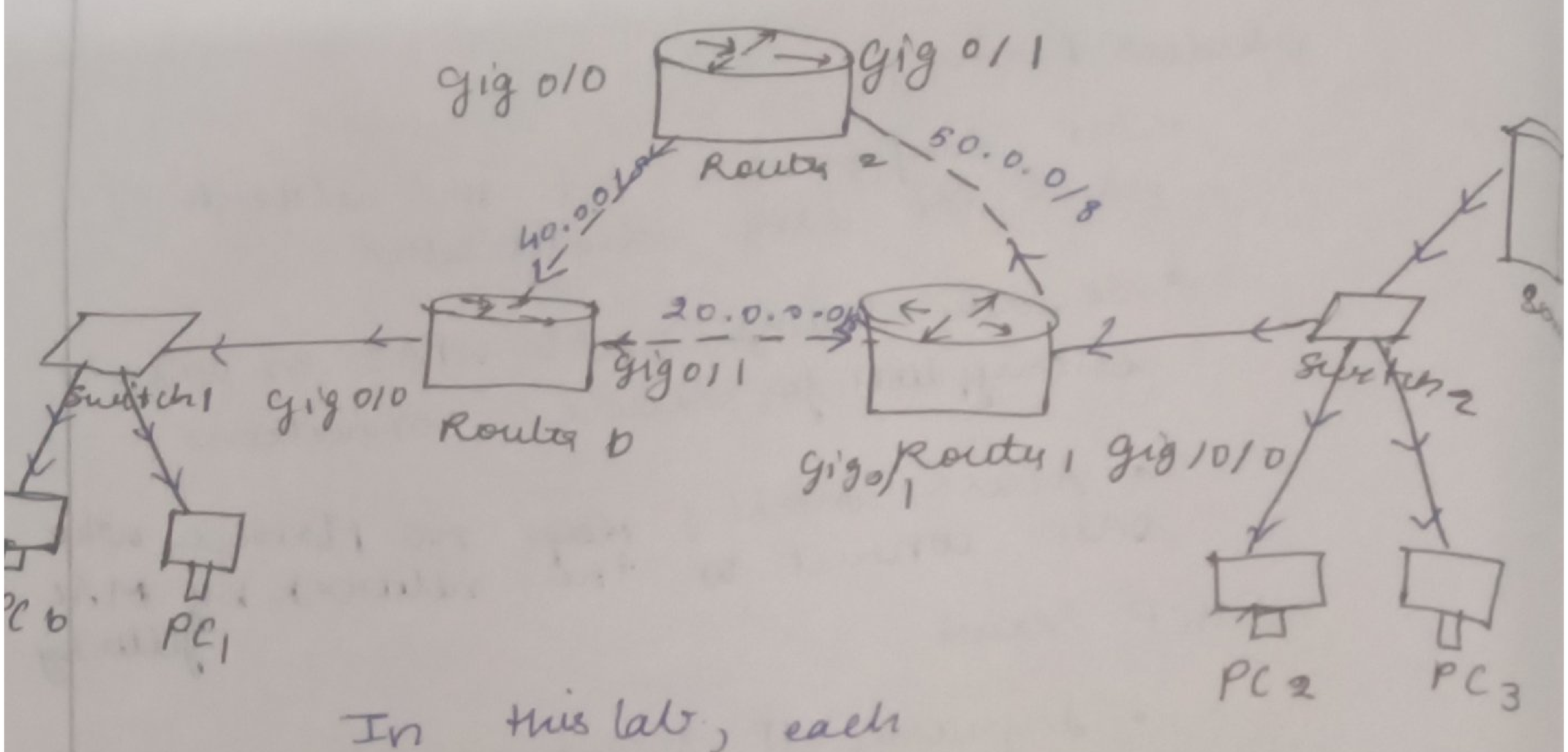


RESULT:- Experiment for design and configure an internetwork using wireless router, DHCP server is executed and verified

Practical 11 (a)

AIM:- a) Simulate Static Routing Configuration using CISCO Packet Tracer

Setting up a practice lab



In this lab, each network has two routes to reach. We will configure one route as the main route and another route as the backup route. If the link bandwidth of all routes is the same, we use the route that has the least no. of routers as the main route.

Creating, adding, verifying static routes

Routers automatically learn their connected networks we only need to add routers for the networks that are not available on the router's interfaces

- Note down the route you want to delete
- Use 'the 'no ip route' command to delete the route

backup route → main route, when you delete the main route

RESULT :- Simulating Static Routing configuration using CISCO packet tracer is executed and verified

Practical 11 (b)

AIM :- b) Simulate RIP using Cisco Packet Tracer

Initial IP configuration

Device	Interface	IP configuration	connected with
PC0	fast Ethernet	10.0.0.2/8	Router 0's Fa 0/1
Router 0	Fa0/1	10.0.0.1/8	PC0's Fast Ethernet
Router 0	S0/0/1	192.168.1.254/30	Router 2's S0/0/1
Router 0	S0/0/0	192.168.1.249/30	Router 1's S0/0/0
Router 1	S0/0/0	192.168.1.250/30	Router 0's S0/0/1
Router 1	S0/0/1	192.168.1.246/30	Router 2's S0/0/0
Router 2	S0/0/0	192.168.1.245/30	Router 1's S0/0/1
Router 2	S/0/1	192.168.1.253/30	Router 0's S0/0/0
Router 2	Fa0/1	20.0.0.1/30	PC 1's Fast Ethernet
PC 1	Fast ethernet	20.0.0.2/30	Router 2's Fa0/1

Assign IP address to PCs

Double click PCs and click Desktop menu ;ion and click IP configuration. Assign IPaddress referring the above table

Assign IPaddress to interfaces of routers

Double click Router0 & click CLI and press enter key to access the command prompt of Router0

Following commands are used to access the global configuration mode

Router > enable

Router # configure terminal

Enter configuration commands, one per line.

End with CNTL / Z

Router (config) #

The following commands will assign IP address on FastEthernet 0/0.

Router (config) # interface fastEthernet 0/0

Router (config-if) # ip address 10.0.0.1 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # end

Router (config) #.

interface fastEthernet 0/0 command is used to enter in interface mode

ip address 10.0.0.1 255.0.0.0 command will assign IP address to interface

no shutdown command will bring interface up.

exit command is used to return in global configuration mode.

Serial interface needs two additional parameters clock rate and bandwidth. Every serial cable has two ends DTE & DCE

Route 1

PC 0 [Source / destination -10.0.0.2] <==> Router 0 [

Fast Ethernet 0/1 - 10.0.0.1] <==> Router 0

[Serial 0/0/1 - 192.168.1.254] <==> Router 2

[Serial 0/0/1 - 192.168.1.253] <==> Router 2

[Fast Ethernet 0/0 - 20.0.0.1] <==> PC1 [

Destination / source - 20.0.0.2]

Route 2

PC0 [Source / destination -10.0.0.2] <==>

Router 0 [FastEthernet 0/1 -10.0.0.1] <==>

Router 0

[serial 0/0/0 - 192.168.1.249] <==> Router 1

[serial [0/0/0 - 192.168.1.250] <==> Router

RIP will automatically reroute the traffic.

Use tracert command.

9/11.

RESULT:- The experiment for simulating RIP using CISCO packet tracer is executed & verified.

Practical 12(a)

AIM :- a) Implement echo client server using TCP/UDP Sockets

Algorithm :-

TCP Echo Sever :-

1. Create a socket using socket() and specify it as a TCP socket.
2. Bind the socket to a specified IP address and port
3. Put the socket in listening mode to wait for incoming connections -
4. Repeat
   - Accept a connection from a client
   - while connected :
     - Receive data from client.
     - If data is received, send it back (echo)
     - If no data is received, break the loop & close the connection
5. Continue listening for clients indefinitely

TCP Echo Client

1. Create a TCP socket
2. Connect the socket to the server's IP address & port
3. Send a message to the server
4. Wait for the server to echo the message back
5. Display the received
6. Close the connection

UDP Echo Client - Server

UDP Echo Server

1. Create a UDP Socket
2. Bind the socket to a specified IP address and port
3. Repeat:
   - wait to receive a message from client
   - send the receive message back to the client's address
4. Continue listening for incoming messages indefinitely

UDP Echo client Pseudocode

1. Create a UDP socket
2. send a message to servers IP address & port
3. wait for server to echo message back
4. display received message
5. close the connection.

Input :- ~~client~~ "Hello TCP server"

Output :- connected by ('127.0.0.1', some_port)
received from client: Hello, TCP Server!
Disconnected from ('127.0.0.1', some_port)

4/9/11

RESULT :- The experiment for implementing echo client server using TCP/UDP sockets are executed

code

Server.py

```
import socket
server_socket = socket.socket(socket.AF_INET
                              socket.SOC
                              STREA
server_socket.bind(('localhost',
                    12345))
server_socket.listen()
print("TCP server listening on port 12345.
while True:
    client_socket, address = server_socket.accep
    print(f"connected to {address}")
    data = client_socket.recv(1024)
    if data:
        client_socket.sendall(data)
client_socket.close
```

client.py

```
import socket
client_socket = socket.socket(socket.AF_I
                              socket.SOCK_STR
client_socket.connect(('localhost', 12345))
client_socket.sendall(b"Hello, server")
data = client_socket.recv(1024)
print("Received from server:", data.de
client_socket.close()
```

Practical 12(b)

AIM :- b) Implement chat client server using TCP/UDP sockets

Algorithm :- TCP Chat Server Pseudocode

1. Create a TCP socket
2. Bind the socket to a specified IP address and port
3. Put the socket in listening mode to wait for incoming connections
4. Accept a connection from client
5. While connected - • start a loop to listen for message from client
   - If a message is received, print it and prompt the server user to send a reply
   - Send the reply to the client
6. When the client disconnects, close the connection
7. Continue to listen for new connections indefinitely.

TCP chat client Pseudocode

1. Create a TCP socket
2. Connect the socket to the server's IP address & port
3. While connected:
   - Send a message to the server
   - wait for server's reply & print it
4. Continue the chat until the user decides to disconnect by closing the socket

UDP chat ~~client~~ Server Pseudocode

1. Create a UDP socket
2. Bind the socket to a specified IP address and port
3. While connected: • Receive a message and the client address
   - Print the message received from client
   - Prompt the server user to send a reply
   - Send the reply to the client's address
4. Continue the chat until the user decides to exit by breaking the loop

UDP chat client Pseudocode

1. Create a UDP socket.
2. While connected:
   - Send a message to the server's IP address & port
   - Receive and print the server's reply
3. Continue the chat until the user decides to disconnected by closing the socket

2. Input : Hello Server !
(client)

output :
(server)

client : Hello, server!

Input : Hi client ! How can I help?
(server)

Output : How can I help you ?
(client)

Code :- TCP chat server :-

```
import socket
import threading

def handle_client(client_socket):
    while True:
        message = client_socket.recv(1024)
        if message:
            print(f"client: {message}")
            client_socket.send(f"Server: {message}".encode())

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen()
```

TCP chat client :-

```
import socket
import threading

def receive_messages(client_socket):
    while True:
        message = client_socket.recv(1024).decode()
        if message:
            print(message)

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))

thread = threading.Thread(target=receive_messages, args=(client_socket,))
thread.start()
```

RESULT :- The experiment for implementing chat client server using TCP/IP sockets is verified and executed.

Practical 13

AIM:- Implement your own ping program

Algorithm:-

1. Define the target: Accept the IP address or hostname as input

2. Set Up the Socket
   - Create a raw socket using ICMP (which requires administrator / root privileges)
   - Set a timeout for the socket to handle cases where no response is received.

3. Create and Send ICMP Echo Request:
   - Construct an ICMP packet with the appropriate header (eg. ICMP type, code, checksum)
   - Send the ICMP packet to the target IP

4. Receive ICMP Echo Reply:
   - Wait for ICMP reply within the defined timeout period
   - Record the round-trip time (RTT) based on the time taken for the packet to travel to the destination and back

5. Display Results:
   - Print the RTT if echo is reply is received
   - If no reply is received, display a timed message

6. Repeat: Repeat steps 3-5 multiple times for more results

7. Compute Summary Statistics:- After all pings are completed, calculate minimum, maximum & average RTT for the session

input :- ping ("google.com")

output :- Pinging google.com [8.8.8.8] with
Reply from 8.8.8.8: time = 15.32ms
Reply from 8.8.8.8: time = 14.67ms
Reply from 8.8.8.8: time = 15.12ms
Reply from 8.8.8.8 time = 14.99ms
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0%
rtt min/avg/max = 14.67/14.99/15

code :- Server.py

```
import socket
def start_server(host='127.0.0.1'
                 port=1234
    with socket.socket(socket.AF_IN
                       socket.SOCK_DGRA
    as s:
        s.bind((host, port))
        print(f"UDP server running o
                                {host}:{
        while True:
            data, addr = s.recvfrom(1024)
            print(f"Received message from
{data.decode()}")
            s.sendto(b'Pong', addr)
if __name__ == "__main__":
    start_server()
```

client.py :-

```
import socket
import time
def ping_server(host='127.0.0.1', port=12345)
    with socket.socket(socket.AF_INET, sock
                       sock_
    as s:
        try: s.settimeout(2)
             start = time.time()
             s.sendto(b'Ping', (host, port))
             data, addr = s.recvfrom(1024)
             end = time.time()
    except socket.timeout: print("Request timed ou
if __name__ == "__main__":
    ping_server()
```

13/9/11

RESULT:- Implement your own ping program for, experiment as been tested and verified and executed.

Practical 14

AIM :- Write a code using RAW sockets to implement packet sniffing.

Algorithm :- 1. Set up the socket :- • create a raw socket with AF_PACKET family for packet Capturing
- Specify the protocol to capture-relevant packets
- Bind the socket to the network interface you wish to monitor

2. Capture & process packets :- • continuously read data from socket to capture packets
- For each captured packet, parse the headers to extract useful information.

3. Display Packet information :- • Extract source and destination IP addresses, port no.s.
- Print out the packet details for each captured packets

4. Repeat :- Continue till manually stopped

Input :- ping google.com

output :- Ethernet Frame:
Desination MAC : 00:1A:2B:3C:4D:5E,
MAC : 00:5E:4D:3C:2B:1A, Protocol : 8
IPv4 Packet : Version : 4, Header length : 20,
TTL : 64
Protocol : 1, Source IP : 192.168.1.0,
Target IP : 8.8.8.8
ICMP Packet:
Type 8 (Echo request), Code : 0

Ethernet frame :
Destination MAC : 00:5E:4D:3C:2B:1A
Source MAC : 00:1A:2B:3C:4D:5E, Protocol 8
IPv4 Packet : Version 4, Header length : 20,
TTL : 64, Protocol : 1, Source IP : 8.8.8.8,
Target IP : 192.168.1.10
ICMP packet :

Type 0 (Echo Reply) , code: 0

code

```
from scapy.all import sniff
from scapy.layers.inet import IP

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("-" * 80)

def main():
    sniff(prn=packet_callback, filter="ip", store=0)

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

13/11

RESULT:- The experiment for using RAW Sockets to implement packet sniffing is executed & verified

Practical 15: Using Webalizer for web log Analysis

AIM: To analyse the different types of web logs using webalizer tool

Procedure:-

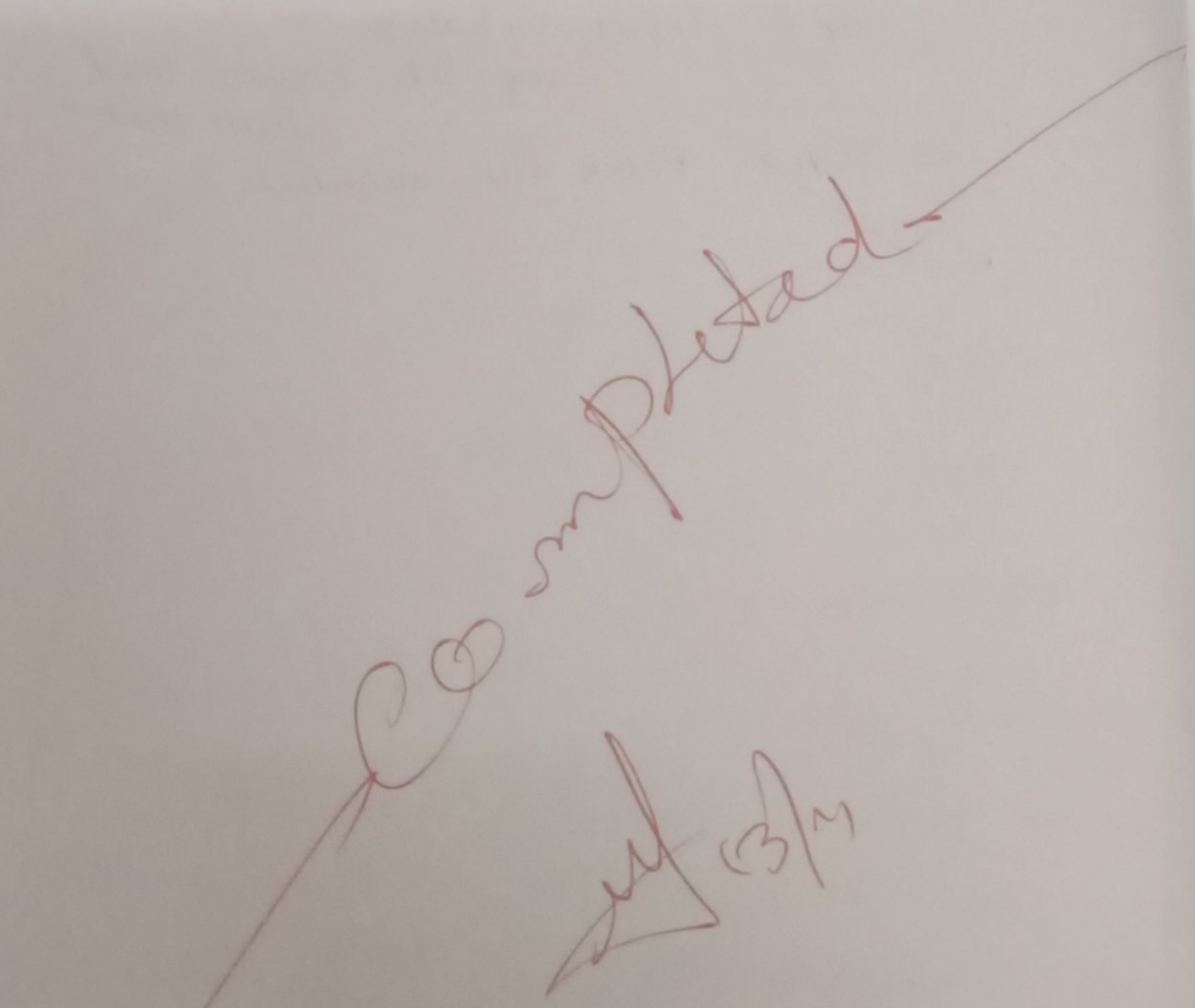
Step 1: Run webalizer windows version

Step 2: Input webalizer ~~w~~ log file (down load from web)

Step 3: Press run webalizer

13/4

RESULT:- Experiment for using webalizer for web log analysis is executed & verified


Completed
3/4