INDEX

NAME : K. Bavesh Karthik STD : ..... SEC ..... ROLL NO. 22071A1039

S.No.	Date	Title	Page No.	Teachers Sign / Remarks
1.	13/7/24	Study of various network command		
2.	27/7/24	Study of different types of network cables		
3.	6/8/24	Study the packet tracer tool Installation and user interface overview		
4.	9/8/24	Setup & Configure of LAN using Stand & ethernet.		
5.	23/8/24	Experiment on packet capture tool : wireshark		
6.	6/9/24	Error detector & correction using Hamming code		
7.	18/10/24	Sliding window protocol		
8a)	20/10/24	configure VLAN		
8b)	20/10/24	Configure Wireless LAN		
9.	24/10/24	Implement of Subnetting		
10a)	26/10/24	Internet working with Packet VLAN		
10b)	29/10/24	Internet working using Wireless Routers		
11a)	29/10/24	Simulate static Routers configurations		
11b)	29/10/24	Simulate RIP		
12a)	29/10/24	Implement echo client server		
12b)	31/10/24	Implement chat/Peer to peer		
13a)	31/10/24	Implement class diagram		
14	31/11/24	Implement packet sniffing using Raw Sockets		
15.	31/11/24	Analyze web logs using Webalizer tool		

Completed

Study of Various Network commands Used in Linux and windows

Ex no: 1

Aim:

Study of various Network commands used in Linux and windows

Basic network commands:

arp a: display the IP address of you computer along with the IP and MAC address of your router

Output: Interface: 192.168.209.1 ... 05

Internet address	Physical address	Type
192.168.209.255	ff-ff-ff-ff-ff-ff	static
224.0.22		
224.0.0.251	01-00 5e-00-00-16	static
	01-00-5e-00-00-fb	static

host name:

Tcp/Ip command that display the name of your computer

output: LAPTOP-LMTOD152

ip config /all: display details Tcp/Ip configuration including Router, Gate way, DNS, DHCP and Ethernet adapter type

output: Windows Ip Configuration

Host Name . . . . . LAPTOP-1M TO D1B2

Primary DNS suffix . . . . .

Node type . . . . . Mixed

IP Routing Enabled . . . . : No

WINS proxy Enabled . . . . : No

Netset -a : helps solve problem with net BIOS

Name resolution (Net BIOS over Tcp/IP)

output: NBTSTAT [[-a Remote Name] [-A IP address]

[-c] [-n] [-r] [-R] [-RR] [-s] [-S] [Interval]

netstate : display statistics about active TCP/IP connections, including network connection / routing table and interface statistics

output: Active connections

Proto	Local address	Foreign address	State
TCP	127.0.0.1:49674	3ca52zh vm1:1600	ESTABLISHED
TCP	127.0.0.1:49675	3ca52zh vm1:49690	ESTABLISHED
TCP	127.0.0.1:49676	3ca52z-vm1:49675	ESTABLISHED

nslookup ! Tool used to perform DNS lookup in linux, displaying details such as IP address mx records and NS servers of a domain

output : server dns.google.com address: 8.8

Non-authoritative answer:

Name: google.com

Address: 2404:6800:4007:816::2008

142.250.182.78

Pathping:

Combines ping and Tracert, tracing the route to a destination, testing each other along the way to gather data loss statistics.

output:

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

ping: Test connecting between two nodes using ICMP (Internet control message protocol) and can be used with a hostname, IP address or fully qualifying domain name

Output:

ping statistics for 142.250.182.78:

Packets: sent = 4, received = 4, lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

minimum = 34ms, maximum = 39ms,

Average = 33ms

Route: Show, manipulates the IP routing table and is used to set up static routes to specific hosts or networks via an interface.

output:

Route [-f] [-p] [-4|-6] command [destination] [mask network] [gateway] [metric metric] [if interface]

Some Important Linux commands

1. ip: Essential for administrators; used to show address information manipulation routing and display network devices, interface and tunnel.

comment syntax: ip <options> <object of command>

a.) show Ip address assigned to an interface

ip address show

output:

ens33: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc pfifo_fast state up group default qlen 1000 link/ether 0e:0c:29:eb:03:46 brd ff:ff:ff:ff:ff:ff

altname enp2s1

inet 192.168.209.130/24 brd 192.209.255 scope global dynamic noprefixroute ens33

valid_lft forever preferred_lft forever.

b.) Assign an IP to an interface: ip address 192.168.209.130/24 dev ens33

c.) delete an Ip to an interface: ip addr del 192.168.209.130/24

d.) Bring an Interface online: ip link set ens33 up

e.) Bring an interface offline: dev ens33

f.) [root @ server ~] # ip link set ens33 up mode for an interface

g.) default via 192.168.1.254 dev route add

[root @ server ~] # ip link set promisc on the status of an interface is allowed by bring the interface offline

h.) [root @ server ~] # ip route add 192.168.1.0/24 via 192.168.1.254

A route is added to 192.168.1.0/24 via
gateway at 192.168.1.254

1) [root @ server] # route add 192.168.1.0/24
dev be reached on the device used
[root @ server N] # ip aroute del 192.168.1.0/24
the gateway at 192.168.1.254

[root @ Server N] # ip route del 192.168.1.0/24.
via 192.168.1.254

The route is deleted for the route
192.168.1.0.0/24 the gate way at
192.168.1.254 k) [root @ server # ip
route get 10.10.1.4 10.10.1.4 via
172.16.8.1 dev enp250 SJC 172.16.5.19
vid o cache

a) IP config.

enp250 flags=4163 <UP, BROADCAST, RUNNING MULTICAST> MTU 1500
inet 172.16.8.98 netmask 255.255.252.0
broadcast 172.16.11.255
inet6 fe80::b1bb:3609:5c:c076
prefixlen 64 scope id 0x20<link>
ether 80:99:4c:34:d4:cc txqueuelen 1000
(Ethernet)

R X Rackets 331201 Bytes 20626 3688

(196 · 7 · 1.6)

R X error odeopped 72 olesuns of rame 0.

Tx Rackets 05914 bytes 44524720 14 · 2 mib)

3.) mtr

d.) mtr gogle · com

local host : local Domain (0.0.0.0)

key it · lps : displaymode restart order

of find sqnit .

Host

1.) 172 · 16 · 8 · 1

2.) statistic - 41.229 · 429 · 49 - tatand co. in

3.) 14250 : 17 · 162 -

packets

loss% snt last pings best west stddev Avg

0.01% 163 0.2 0.2 0.1 1.1 0.0

5.) mtc · b · google · com

local host · local admin keys Helps

Displayed mode Restart order or fields

exit Host

1.) 172 · 16 · 8 · 1

2.) Static 41 · 229 · 429 · 49 - static coin

3.) 142 · 250 · 171 · 102

Packets pings

Loss% snt last Avg Best Wost stdev

0.0% 3 0.2 0.2 0.1 0.1 0.0

d) mtr -c google.com

local host local domain

keys help displayed restart statistic of

exit Host

1.) 172.16.18.1

2.) 142.250.171.102.

4.) tcp dump

Last meta data expiration check

1:26:24 ago on Thu 23 Jan 2024

05:13:33 AM is package tcpdump. 14:4:9

If 20 1650 is already installed, skipping

Indentiss Solved Nothing to do compute

Result:

Thus the Study of various Network

connections used in Linux and

Windows in done executed successfully.

Exp No: 2
Date: 22/7/24

Study of types of cable

Aim:

Study of different type of network cables

a.) understand different types of network cables different type of cables used in networking are

1.) Unshielded Twisted pair (UTP) cable

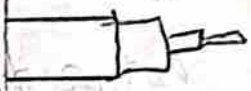
2.) shielded twisted pair (STP)

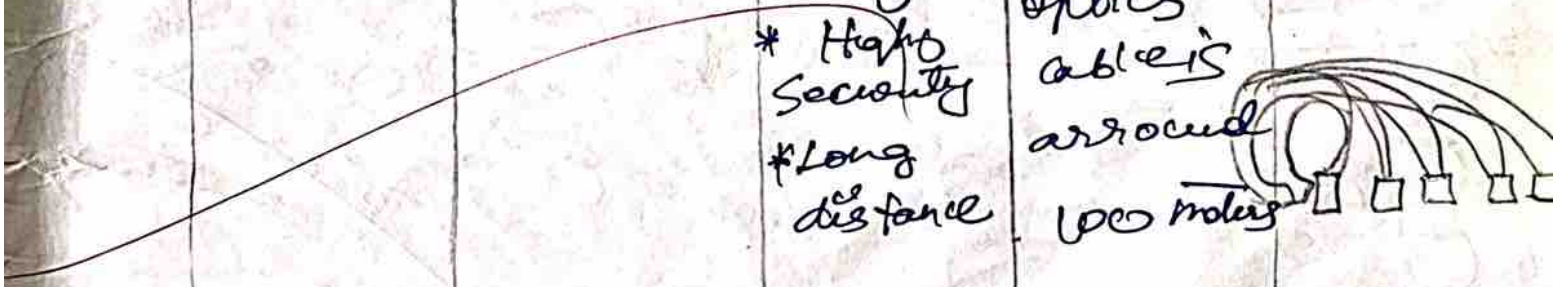
3.) Coaxial cable

4.) Fibre optic cable

cable Type	category	Maximum Data Transmission	Advantage Disadvantage	Application use	Image
UTP	category 3 category 5 category 5e	10bps upto 100 mbps 1 or bps	Advantage *Cheaper Incost *Easy to install as they have a smaller overal diameter Disadvantage more prone (EMI) Electro magnetic	10 Base Ethernet Fast Ethernet or gigabit Ethernet Fast Ethernet Gigabit Ethernet	
STP	category 6/6a	10 Gbps	Advantages Shielded	gigabit Ethernet 10 Ethernet (BSM)	
STP	Category	10 or bps	Faster than ust Less Sagnitt to no Beard Interference	Usually used in data centres	

STP	Category	10 Gbps	Disadvantage Expensive creates installation effort Expensive creates installation effort	Cat 6A cable Ethernet 10 Gbps (100 m)
Coaxial Cable	Rg-6 RG-59 RG-11	10-100 mbps	Advantages * High bandwidth * Immune to interference lowless bandwidth versatile	Used at Signal is 50 ohm Television network High speed internet connections
Fiber optics Cable	Single mode multimode	100 Gbps	Advantages *High speed *High bandwidth * High Security *Long distance Disadvantage * Expensive * requires Skilled Installer	maximum bandwidth of fiber optics cable is around 100 miles





b. Make your own Ethernet cross over cable

Straight cable

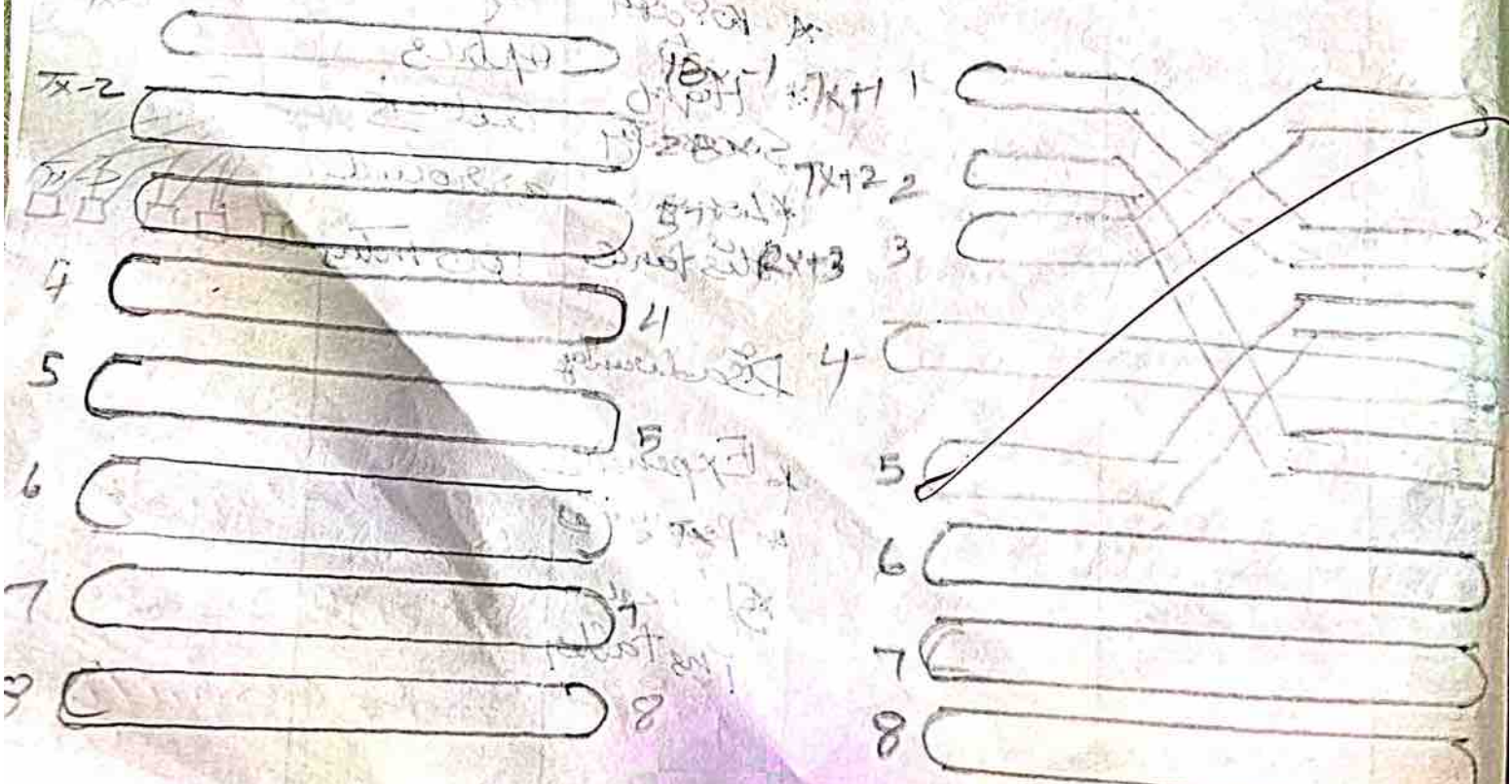
Tools and parts needed

Ethernet cables CAT5 is certified for gigabit support, but CAT5 cabling works as well just over shorter distance

A - crimping tool this is an all in one networking tool shaped to push down the pins in the plug and strip and cut the shielding all the cables

* Two RJ45 plugs
* optional two plug shields

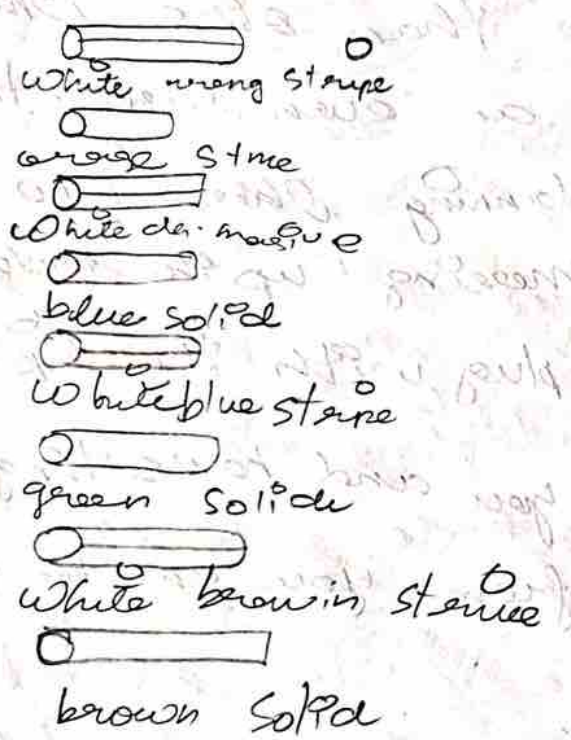
Straight through cables. X-over cable



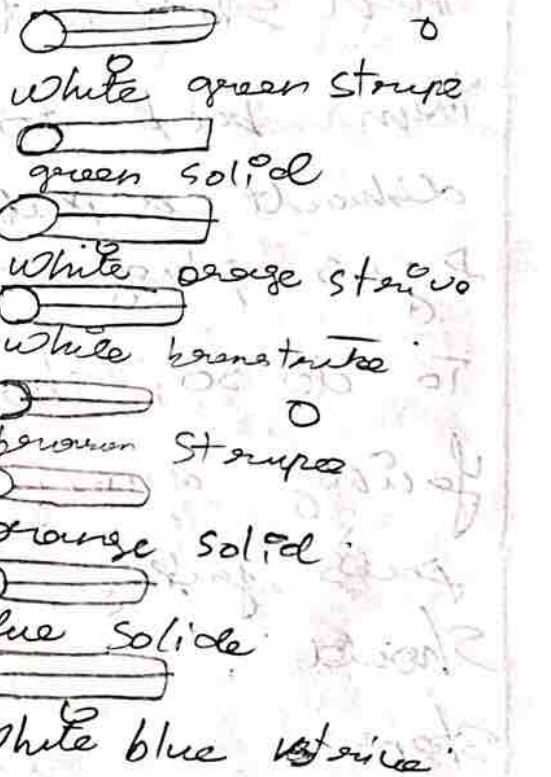
Difference between crossover cable and straight cable

Straight Through network cable. both side should be A cross over cable. one sid A one sid, B

A



B



Step1:-

To start construction at the device, begin by Threading Shields into the cable

Step2:

Next, Strip approximately 1.5cm at cable Scheduling from both the ends. the Crimping tool has a ranged area to complete this tiggs

Step3:

After, you will need to angle the cables there should be four twisted pairs there but

to the sheet, arrange them from top to bottom one and should be in arrangement A and other B.

Step4: Once the order is correct, push them together in a line and if there are any that stick out farther than others, slide them back to create an even level. The difficult aspect is planning these into the RJ45 plug without messing up the order. To do so, hold the plug with the clip side facing away from you and have the gold pins facing away from you and how it should.

Steps:

Next, push the cable right in the notch at the end of the plug needs in the be just, that over the cable shielding and it app too much shielding simply strip the cable back a little more.

Step 6:

After the wires are securely sitting inside the plug, insert it into the crimping tool and push down

step 7:-

Lastly, repeat for the other end using diagram (B) using diagram (A)

Result:-

Thus the different types of Network cable and crimping and clamping at cable is connected and Established the connection between the devices.

23/11

Exp No: 3

Experiments on CISCO Packet Tracer (Simulation Tool)

Aim:-

To study the packet tracer tool installation and user interface overview

c) To understand environment of CISCO PACKET TRACER TO design simple network

Introduction:-

A Simulator as the name suggests simulates network devices and its environment Packet Tracer is an exciting network design Simulation and modelling tool

1. It allows you to make complex system without the need for additional equipment
2. It helps you to practice your network configuration and trouble shooting skills. Wier computer or an Android or IOS bases mobile devices
3. It is available for both the linux and windows desktop environment

A Protocole in packet Tracer are added to work and behave in the same way as the would on real hardware

INSTALLING PACKET TRACER :-

To download packet tracer, go to :
https://www.netacad.com and log with your cisco networking Academy credentials then, click on the packet tracer graphic and download the packet appropriate for your operating System.

Windows :-

Installation in window is pretty simple and straight forward the setup comes in a single file named packettracer_setup_8.0.1.exe open this file to begin the setup wizard, accept the license agreement, choose a location, and start the installation.

Linux

Linux with an ubuntu Debian distribution should download the file for ubuntu, and those using Fedora / Redhat pantos must download the file for fedora Once executable permission to this file by using chmod, and execute it. to begin the installation

Chmod +x packet Tracer 601 - 1396 - installer - rpm .bin

Packet Tracer 601 - 1386 - installer - rpm .bin

1. menubar:-

This is a common menu found in all Software applications it is used to open, Save, print image rendername and do on.

2. Magic Toolbar:-

This bar provides shortcut icon to menu option that are commonly accessed on the engine - hand room, undo and redo and.

3. Logical physical workspace Tabs:-

There Tabs allow you to toggle between the logical and physical work areas.

4. Workspace:-

This is the area where topology are created and simulations are displayed

5. Common toolbar:-

This toolbar provides controls for manipulation topology. such as Select, move layout, place, note, edit Inspect design shape and add simple complex PDV

6. Real time/ Simulation table :-

There table are use to to gole between the real and simulation model Buttons are also provide to control the time and to capture the packets

7. Network component box :-

This components contain all of the network and end devices available with packet Tracer and is further divided into two area "Device type Selection box - This area contains device categories Area Tab:-

8. User - Created box :-

Users can create highly customized packets to test their topology From this area and the results are displayed as a list

d. Analyse the behaviours of network devices using CISCO packet TRACER Simulator :-

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 to hub using the cable. the link LED should glow in green, indicating that the link is up. Similarly connection remaining 3PCs to the hub

c. Similarly connect 4PCs to the switch using copper straight through cable.

3. Click on the PCs connection to hub goto the Desktop, click on the IP configuration and enter an IP address and subnet mask.

Click on the PDU (message icon) From the common tool bar,

a. Drag and drop it on one of PC (source machine) and then drop it on another PC (distination machine) connected to hub

4. observe the flow of PDU 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 about the behaviour of switch and Hub.

Student observation:-

a) From your observation the behaviour of switch and the hubs in the terms are forwading packet received.

b.) Network topology implemented in our college

a) A cross cable connect devices of the same type directly Eg [PtoPc] by swapping the transmit and receiver pairs

b) Cross cable type is used

c) which type of cable is used to connected a router / Switch to PC?

Straight cable is used

d: Find out the ~~category~~ of twisted pair cable used in laptop to PC? category of twisted pair cable is cat5e and cat6 are used

e: Write down the understanding, challenges faced and output received while making a twisted pair cable / Straight cable.

making twisted pair cable involves understanding pin configuration and ensuring proper ~~termination~~ include ~~soldering~~ measuring and ~~double~~ connection.

Result:-

Thus the different types of Network cable is studied and the Ethernet cable straight is connected successfully.

Ex no: 4

Date: 10.8.2024

AIM:-

Setup and configure a lan local area network using an switch and Ethernet cable in your lab.

What is lan:-

A local area network (LAN) refers to a network that connect devices within a limited area such as an office building school or home it enables users to share resources, including data printers and internet access. lan connects devices to promote collaboration and transfer information between users

How to setup a LAN:-

Step 1:

plan and Design an appropriate network topology taking into account network requirements and equipment location

Step 2:-

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

Step3:-

Connect your computer to network switch via a Ethernet cable, which is a sample as plugging one end of the Ethernet cable into your computer and the other end into your network switch.

Step4:-

Assign IP address to your PC

1. log on to the client computer as Administrator or as owner.
2. click Network and Internet connections.
3. Right click Local Area connection/ Ethernet -> Go to properties -> Select Internet Protocol (TCP/IPV4) -> click on properties -> Select use the Following ip address option and assign ipaddress

Step 5:-

Configure a network switch

1. Connect your computer to the switch : To access the switch web interface, you will need to connect your computer to the switch using an Ethernet cable
2. Assign IP Address as 10.1.15 Subnet mask

Step 6:-

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

Step 7:-

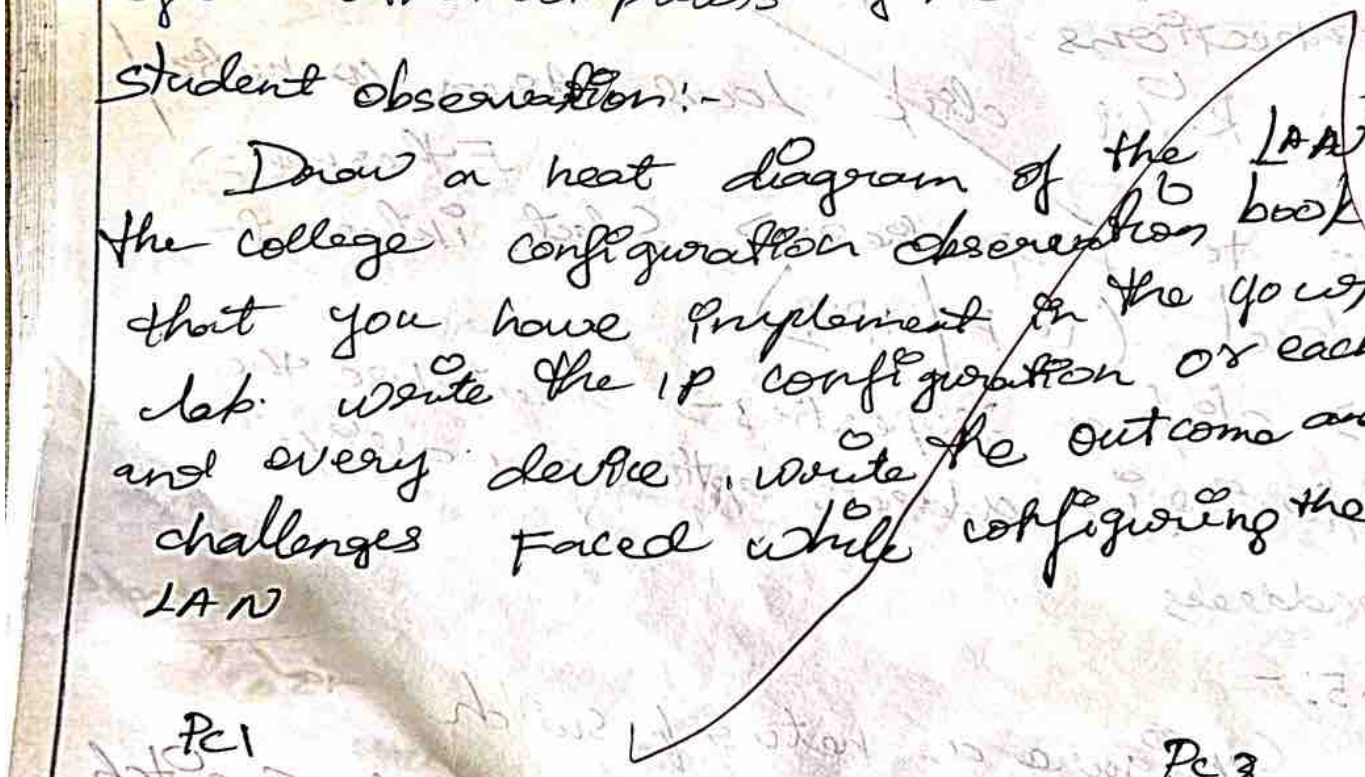
Select a Folder → go to properties → click sharing tab → share it with everyone on the same lan

Step 8:-

Try to access the shared folder from other computers of the network

Student observation:-

Draw a neat diagram of the LAN in the college configuration observation book that you have implement in the previous lab. write the IP configuration of each and every device, write the outcome and challenges faced while configuring the LAN



* Advance manage shared file between both device

* Now Search the file on Secondary device

Internet Protocol version 4 (TCP/IPv4) properties

General

You can get settings assigned automatically network support this capability otherwise you need to ask your network administrator for the appropriate IP setting

- Obtain IP address automatically
- use the following ip address

IP address 10.1.1

Subnet mask 255.000

Default gateway . .

- use the DNS server address

preferred DNS server . . .

Alternate DNS server . . .

ID Validate settings upon exit

Advanced

Result:-

Thus the lan connection has been successfully established between two devices

EXNO:3

Date

Aim:- Experiments on packets capture tool wireshark.

Packet shifter:-

-> Sniffs messages being sent/received from/by your computer.

-> Store and display the contents of the various protocol fields in manager

-> Never sends packets sniffs.

-> No packets Addressed to it

-> Receives a copy of all packets

-> Packets sniffs structure Diagnostic tool

-> uses (eg. tcpdump - enrost 10.129.41.2)

-> Wire share (wire shark is exact)

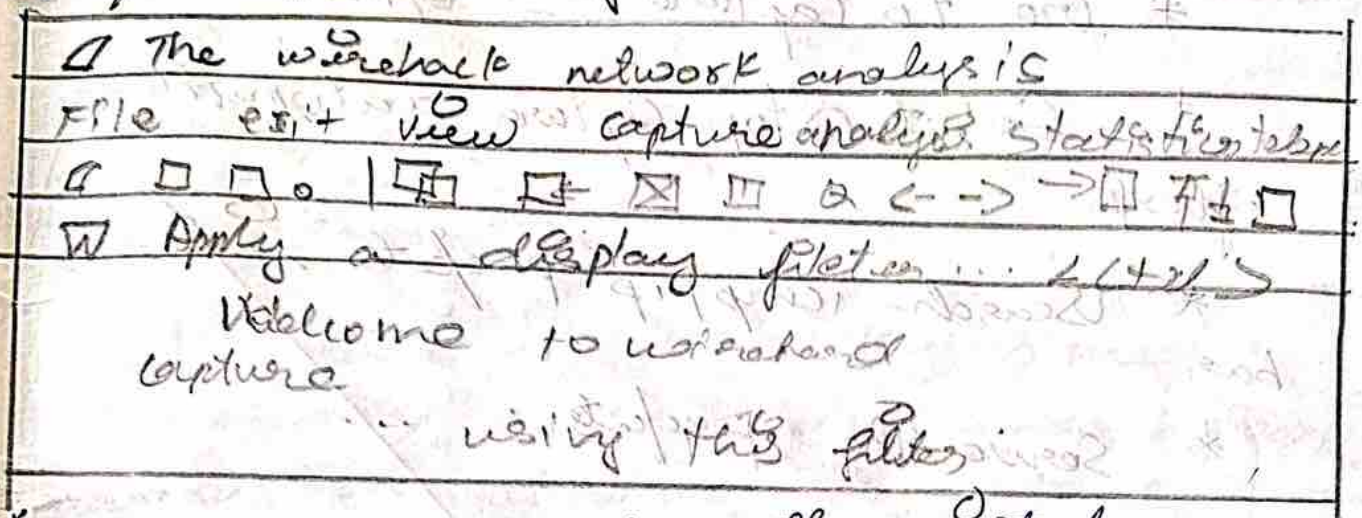
Wireshark:-

Wireshark a Network analysis tool formerly known as Ethereal, capture packet in real time and display them in human readable Format

Getting wireshark: wireshark can be download for various os macos from the official website capturing packets: After Downloading and installing wireshark

Capturing packets :-

After downloading an installing wireshark launch it and double-click the name of a network interface under Capture to start capturing packets that interface



As soon as you click the interface name, you will see the packets start to appear in real time wireshark capture each packet sent to or from your system

colour coding :-

You probably see packet highlight in a variety of different colour wireshark use colors to help you identify the types of traffic at a glance.

By default light purple is TCP traffic, light blue is UDP traffic and black identifies packets with errors - For example they could have been delivered out of order

5. Create a filter to display only ICMP packet and inspect the packets

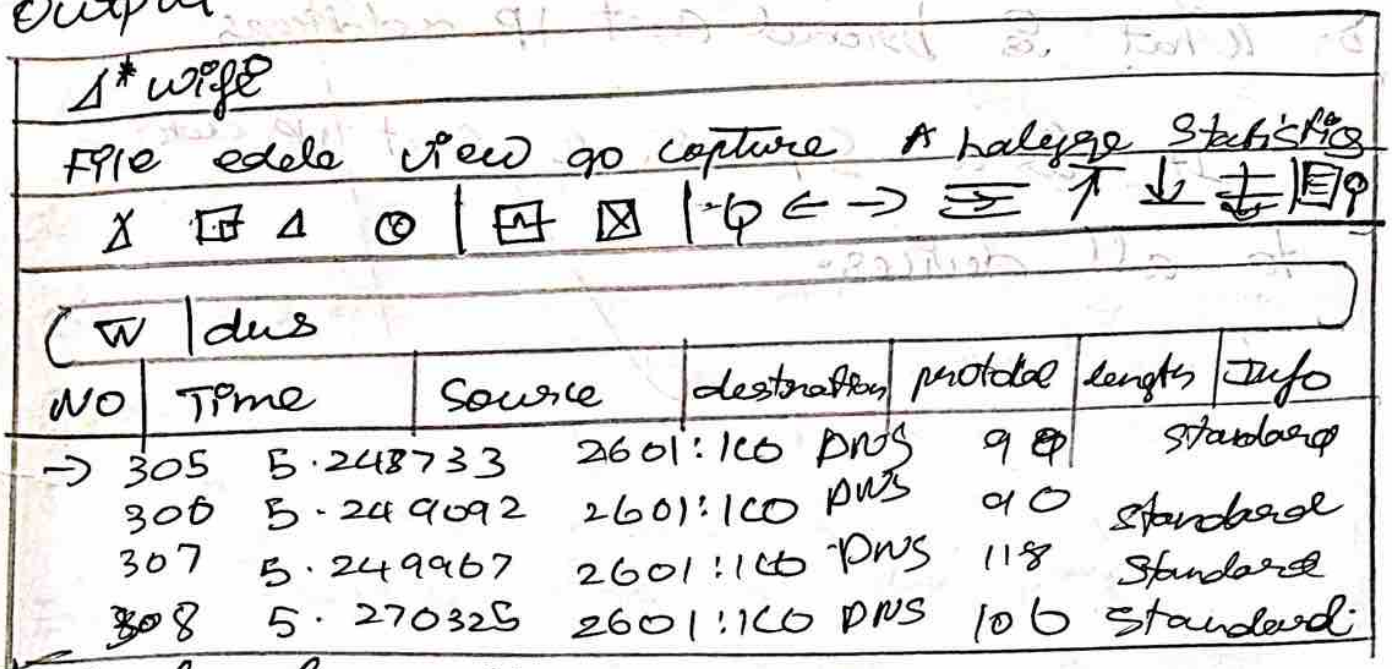
Procedure:

* Select local Area connection in Wireshark
* Go to Capture → option
* Select Stop Capture automatically after 60
* Search icmp/ip packets in search bar
* Serve the packets.

6. Create a filter to display only DHCP Packets and Inspect the packet procedure

* Select local Area Connection on Wireshark
* Go to Capture → option
* Then click start Capture
* Save the packets.

Output:



Student observation

1) What is promiscuous mode?

promiscuous mode is a computer networking feature that allows a device to read and capture all network traffic rather than just the traffic intended.

2) Does ARP packets have a transport?

No, ARP packet do not have a transport layer header because it operates at the data link layer at the OSI

3) Which transport layer protocal is used by DNS?

DNS primarily uses UTP as transport layer protected but it also uses Tcp for layer expensive

4) What is the port number used by HTTP protocol is for HTTPS the secure default port is 443

5. What is broad cast IP address

It is a Speacial cast IP address to all devices.

Result :-

Thus the experiment on packet capture tool : Wire Shark is verified successfully

Ex no:-6

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

Error correction at Data link layer:-

⇒ Hamming code is a set of error-correction code can be detect and correct the error that can occur when the data is transmitted From the sender to the receiver. It is a technique developed by Bw Hamming For error correction

Create Sender program with below feature

1. Input to Sender file should be a text of any length. program should correct the text to binary.
2. Apply hamming code concepts on the binary data redendant bits to it
3. Save this output in a File called channel

Create a receive program with below feature :-

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.

Also remove the redundant bits and convert the binary data to asci and display.

Student observation :-

```
def text_to_binary(text)
    return Join (Format (ord(char), '08b')
def binary_to_text (binary):
chars = [binary [i : i+8] for i in
range (0, len (binary), 8)]
return Join (Chr (int (char, 2)) for
char in chars])
def calc_redundant_bits(m):
    r=0
while (2**r < m + r + 1)
    r+ = 1
return r
```

```
def pos - Redundant - bits (data, r)
    j = 0
    k = 0
    m = len
res =
print (placing redundant bits at positions.
For i in range (1: m + r + 1):
    if i == 2 ** j:
        res = res + '0'
    print (i, end = " ")
    j += 1
else:
    res = res + data [k]
    k += 1
print ()
    return res
def calc - parity - bits (arr, r):
    n = len (arr)
    arr = list (arr)
    parity - bits = []
For q in ranges (r):
    parity = 0
    position = 2 **
    For s in range (1, n+1):
    if j & position:
        parity = in (arr [j-1])
arr [position -1] = str [parity - bits]:
```

print(f position[pos]: {bit}", end=")
print()
return join(arr)

def detect_and_correct(data, r):
r = cal_redundant_bits(len(data))
print(f"Binary errors
correction corrected data:")
ascii output = binary_to_text
(corrigicat_data)
print(f" Dec odd text {ascii_output}")

IF __name__ == "__main__":
input_text = input("Enter the text
encoded):
channel_data = sender(input_text)
corrupted_data = introduce_error
(channel_data, 2)
receiver(corrupted_data).

Output

Enter the text to be encoded:
: Bavesh Karthik.

01110110011 00001010010 11101000101010
00001101011011

Placing redundant bits at positions
1, 2, 4, 8, 16, 32, 64

Parity Bits: [Position 1:0] [Position 2:0]
[Position 4:1]
[Position 8:1] [Position 16:1] [Position 32:0]
[Position 64:1]

introduced error at position: 2
Binary with error: 010111110110011 0111
000001011 010010101110011011101110111 0000
101110010010111
error detection at position: 2
error correction at position: 2
Binary after error correction: 00011111
0110011100111000010110100101011 01 000
1010111 00 11 01110111011100000101110
00011 00000 1 0110 1110
Decode text . Bavashkarthik

Result:-
Thus the python program for hamming code to detect and correct has been executed successfully.

Exp NO: 7

Date :

AIM:-

Write a program to implement flow control at data link layer using Sliding window protocol Simulate the flow of frames from one node to another

Program should achieve at least below given requirements you can make it a bidirectional program wherein receiver is sending its data frame with acknowledgements (Piggybacking)

Create a Sender program with the following features:-

1. Input window size from the user
2. Input a Text Message from the user
3. Consider 1 character per Frame
4. Create a Frame with Following Fields
5. Sends the frames
6. Wait For the acknowledgement from the receiver.

7. Reader a file called receiver Buffer.
8. Check Ack field for the Acknowledgement number.

a. If the Acknowledgement number as expected Send new set of Frames accordingly.

[Cover Write the Sender Buffer file with new Frames]

```
import time
    import random
class Frame
    def init - (self - frame - no data)
    self frame no frame - no
    self data = data
    self acknowledge = False
def Sound - frame (frame, window - size)
    print ("[n - - Sender reader - -")
    for fn range (window - size):
if i< fn (frames) and not frames[i]
acknowledged:
    print ("Send frame & frames[i]
brand 3
& frames[i] data 3')
```

Print ("Frame Sent, Waiting for acknowledge.

def receiver_frame (frames, Sarom -sing)

print ("Cn-Receiving Frames --")

for i in rang (window_ring)

if i < len (frame) and not dream [i]

acknowledge:

if random random () < 0.1:

print (f" Received frame {frames [i] frame_no}: {frame [i] data} [Error]

frames [i] acknowledge = False

else

print (F" Received frame {frames [i] frame_no} {frame [i] data} (On}")

frames (i) acknowledged = True

def sliding_window_protocol ()

Window size = int (input ("Enter window size"))

```
message = input("Enter a message to send")
frames = [frames("" message "")for
in range(len message))]

base=0
while base < len(frames):
    send - frames(frames[base] window
                  -size)
    time.sleep(2)
    receive - frames(frames[base]
    window - size) what base < len(frame
    and frames[base]
    acknowledge:
base+=1
if base < len(frames):
print("\n resending unacknowledge
      frames ... \n")
time.sleep(2)
print("\n frames sent and
      acknowledge)"
if -name- == '-- main --:")
```

output

```
Enter window size: 6
Enter a message to user: BAVESH
..... Sending Frames ...
```

Sent frame 0: B
Sent frame 1: A
Sent frame 2: V
Sent frame 3: E
Sent Frame 4: S
Sent Frame 5: H

Frame Sent, waiting for acknowledgment
..... Receiving frames <- - - - -
Received Frame 0: B [Received]
Received frame 1: A [Received]
Received frame 2: V [Received]
Received frame 3: E [Received]
Received frame 4: S [Received]
Received frame 5: H [Error]
Resending frame
..... Resending frame
Received frames 4: H [Received]

Result:-
Thus the code for flow control Sliding control is executed Successfully.

Exp No:-8

Aim:-

a) Simulate Virtual LAN configuration using Cisco packet Tracer Simulation

objectives:

Part1: Build the network and configure Basic Device settings

Part2: Create VLANs and Assign switch ports

part3: Maintain VLAN port assignments and the VLAN database

Part4: Configure an 802.1Q Trunk between switches

Instructions:

Part 1:- Build the network configure Basic device setting

step 1: Build the network as shown in the topology

a. click and drag both switch S1 and S2 to the rack

b. click and drag both PC-A and PC-B to the table and use the power button to turn them on.

c. Provide network connectivity

d. connect console cables from devices.

Step 2: Configure basic settings for each switch

a. From the desktop tabon each PC, use the terminal to console, tabo each switch.

b. Enter configuration mode.

c. Assign a device name to each switch.

d. Assign class as the privileged EXEC encrypted password.

e. Assign cisco as the console.

f. Assign Cisco as the city.

g. Encrypt the plaintext passwords.

h. Create a banner that warns anyone accessing the device.

Configure the IP address listed in the address Table for VLAN 1 on the switch.

i. Shut down all interface that will not be used

k. Set the clock on each switch

l. Close configuration window

Step 3: configure Windows

Step 4: Test connectivity.

Part 2: Create VLANs and Assign Switch ports.

Step 1: Create VLANs on the Switch.

a.) Create the VLANs on S1

b.) Create the same VLANs on S2.

c.) Issue the show VLAN brief command to view the list of VLANs on S1

Step 2: Assign VLANs to the correct Switch Interfaces

a.) Assign VLANs to the interface on S1

1.) Assign PC-A to the operation VLAN

2.) From VLAN1 remove the management IP address and configure it on VLAN 99

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

c.) Issue the show ip interface brief command

d.) Assign PC-B to the operations VLANs on S2

e.) From VLAN1, remove the management IP address and configure it on VLAN 99.

Part 3: Maintain VLAN Port Assignments and the VLAN Database.

Step 1: Assign a VLAN to multiple interfaces From the Desktop tabon each PC, use Terminals to continue configuring both Network Switches.

Step 2: Remove a VLAN assignment from an Interface

Step 3: Remove a VLAN ID from the VLAN database a. Add VLAN 30 to interface F0/24 without issuing the global VLAN command.

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

c. Use the no VLAN 30 command to remove VLAN 30 from the VLAN database.

Part 4: Configure an 802.1Q Trunk between the switches.

Step 1: Use DTP to initiate trunking on S2 F0/1

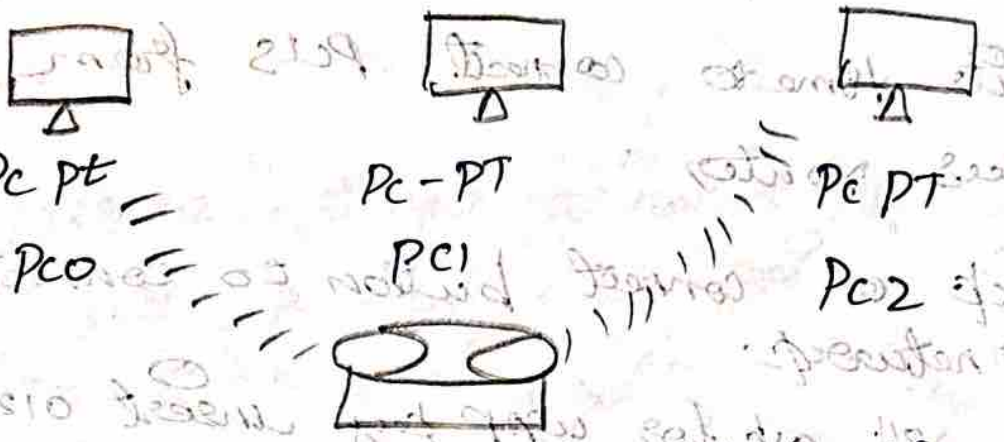
Step 2: Manually configure trunk interface F0/1

Result:-

Thus, Virtual LAN Configuration using CISCO PACKET TRACER is executed successfully.

Aim:-

b.) Configuration of Wireless LAN using cisco packet Tracer.



To complete these tasks follow these step by step instructions:-

Step1:- Click on wireless router.

* Select Administration tab from top menu / Set username and password to admin and click on save settings.

* Next click on wireless tab and set default. ESSID to Mother Network.

* Now select Wireless Security and change Security mode to WEP.

* Again go in the end of page and click on save settings.

PC	IP	Subnet mask	Default Gateway
PC0	192.168.0.2	255.255.255.0	192.168.0.1
PC1	192.168.0.3	255.255.255.0	192.168.0.1
PC2	192.168.0.4	255.255.255.0	192.168.0.1

Now its time to connect PCs from Wireless router

- Click on connect button to connect mother network.
- It will ask for wpa key insert 01234567 and click connect
- It will connect you with wireless router
- And PC1 card is active
- Repeat same process on PC1 and PC2

Student Observation:-

c) What is SSID of a Wireless router?

- SSID (Service Set Identifier) is the name of a wireless network. It is used to identify and differentiate one network from another. When you connect to a wifi network, you typically see a list of available SSIDs, which are the names of the wireless network being broadcast by nearby routers or access points

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

A security key is a password that is used to protect a wireless network. It ensures that only authorized users can connect to the network.

e.) configure a simple wireless LAN in your lab using a real access point and write down the configuration in your network.

1. Gather Equipment
   - Wireless router or access point
   - Device for configuration
2. Connect to the Access point.
   - plug in the router.
   - Connect your laptop to the router
3. Access the Routers configuration page.
4. Configure SSID and Security settings
5. Set up IP Address and DHCP settings.
6. Save and Reboot
7. Verification.

These steps will help you setup and secure a simple wireless LAN in a lab environment

Result:-

Thus, configuration of wireless LAN using Cisco Packet Tracer is executed successfully.

Exp No: 9

Aim:-

Implementation of Subnetting in CISCO Packet Tracer Simulator

Classless IP subnetting is a technique that allows for more efficient use of IP address by allowing for subnet mask that are not just the default masks for each IP class. This means that we can divide our IP address space into smaller subnets which can be useful when we have a limited number of IP address but need to create multiple networks.

Creating a network topology:-

The first step in implementing classless IP subnetting is to create a network topology in packet Tracer.

Adding the devices:-

Once we have created our network topology we can add devices to it. Here we will be adding routers switches and PCs.

Subnetting :-

To subnet the network address of 192.168.1.0/24 to provide enough space for at least 5 address for end devices, the switch, and the router, we can use a 127 subnet mask. This will give us 2 subnets with 30 hosts address each.

configuring the devices :-

. Now that we added our devices and connected them, we can start configuring them. This will open the command line interface (CLI) for the router. In the CLI, enter the following commands :

enable
configure terminal
interface fastEthernet 0/0
ip address {IP address} {subnet mask}
no shutdown
exit

interface Fast Ethernet 0/1
ip address {IP address} {subnet mask}
no shutdown
exit

Replace "{IP address}" and "{Subnet mask}" with your desired IP address and Subnet mask.

Next we will configure the switch.
Right click on the switch and select "CLI" In the CLI enter following commands:-

enable
configure terminal
interface fast Ethernet 0/1
switch port mode access
exit
interface Fast Ethernet 0/2
switch port mode access.
exit

Testing the network:-

Open a command prompt on each pc and try to ping the other pc. If the ping is successful then the network is functioning properly we can also use the "ping" command to test connectivity between the routers and the PCs.

Student Observation:-

a) Write down your understanding of Subnetting

Subnetting is the process of dividing a large network into smaller, more manageable subnetworks (subnets). Each subnet can operate independently, while still being part of the larger network. In subnetting we modify the default subnet mask to allocate Ip address to different network segments, improving network organization and efficiency.

b) Advantage of Implementing Subnetting :-

- Improved network management
- Enhaced Security
- Efficient Ip address utilization

• Reduced Network Traffic.

c) Subnetting Implementation in college

• Research and Mapping

To determine if Subnetting is in place at your college, consult with the network administration team they can provide insights on how the Ip address are segmented.

• Subnet diagram and IP address list.

• If Subnetting is implemented, you can create a visual representation (network diagram) and list the subnets with associated IP address for each department or section

Result:-

Thus the above program executed successfully.

Exp No: 10

Aim:-

a) Internetworking with routers in cisco packet Tracer Simulator.

In this network, 4 routers and 2 PCs are used computers and connected with routers using a copper straight through cable. After forming the network to check network connectivity a simple PDU is transfered from PC0 to PC1

Procedure:-

Step 1 (Configuring routers):

1. Select the routers and open CLI.
2. press Enter to start configuring routers.
3. Type enable to activate the privileged mode.

Step-2 (Configuring PCs):

1. Assign Ip address to every pc in the network
2. Select the PC. Go to the desktop and select Ip configuration and assign an Ip address Default gateways, Subnet mask.

3. Assign the the default gateway of PC0 as 192.168.10.1

4. Assign the default gateway of PC1 as 192.168.20.1

Step3 (Connecting PCs with Routers):

1. Fast Ethernet 0/0 port of Router1

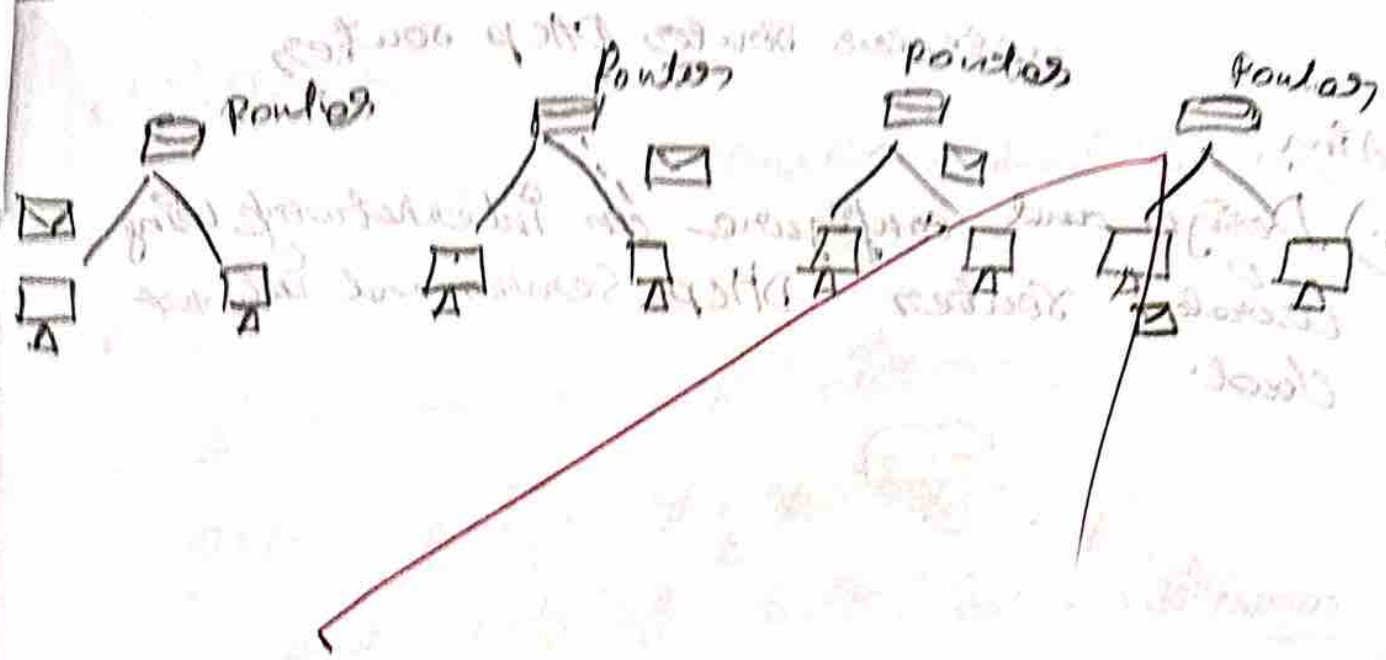
2. Fast Ethernet 0/1 port of Router1

Router configuration Table:-

Device Name	Ip address Fast Ethernet 0/0	Subnet mask	Ip addrs Fast Ethernet	Subn mask
Router1	192.168.10.1	255.255.255.0	192.168.20.1	255.25

PC configuration table:

Device	Ip address	Subnet mask	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.20.2	255.255.25.	192.168.20.1



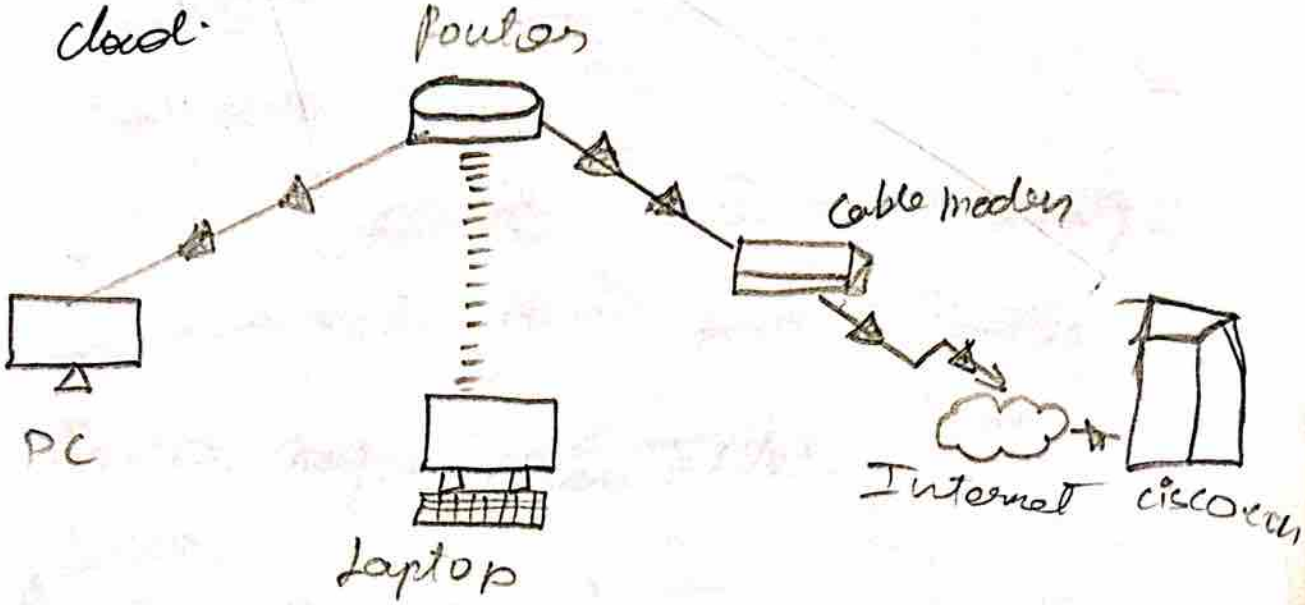
Result:

Thus the above program verified successfully using Cisco packet Tracer.

Wireless Routes DACp routes

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)
Wireless Router	LAN	19-168.0.1)	255.255.255.0	
Wireless Router	Internet	DACp		
Ciscocom Server	Ethernet 0	808.67.220-210	255.255.255.0	
Laptop	Wireless	DHCp		

objectives:-

Part 1: Build a simple Network in the logical Topology workspace.

step1: Launch packet Tracer.

step2: Build the topology

a. Add network devices to the workspace

To place a device onto the workspace, first choose a device type from the device type selection box

b. Change display names of the network devices to the workspace.

To change the display name of the network devices icon on the packet Tracer logical workspace then click on the config tab in the device configuration window.

c. Add the physical selection box, add the physical cabling between devices on the workspace.

using the device selection box, add the physical cabling between devices on the workspace.

The PC will need a copper straight through cable to connect to the wireless router.

Part 2: Configure the network Devices.

Step 1: Configure the wireless router

a.) Create the wireless network on the wireless router

b.) Click on the save settings tab.

Step 2: Configure the Laptop.

a.) Configure the laptop to access the wireless network

Step 3: Configure the PC

a.) Configure the IP for the wired network

Step 4: Configure the internet cloud

a.) Install network modules cloud

b.) Identify the form and IO ports

c.) Identify the type of provider.

Step 5: 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 Fast Ethernet Interface settings.

Part 3: Verify connectivity.

Step 1: Refresh the IPv4 setting 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. Write down the key feature of configuring wireless router and DHCP server.

Wireless router configuration

. SSID configuration: Set up a unique network name (SSID) for your wireless network to allow devices to identify and connect.

- Security settings:- Configure network security to protect against unauthorized access.
- Password: Set a strong password for connecting to the network.
- Channel Selection: Choose a wireless channel that minimizes interference from other networks or devices.
- Frequency Band: Select the 2.4 GHz or 5 GHz band, depending on device compatibility and coverage requirements.

DHCP Server configuration:-

- IP address range- Define the IP address range that the DHCP server will assign to devices.
- Subnet mask: Specify the subnet mask to define network boundaries.
- Lease time: Set the duration for which an IP address is assigned to a device.

2. Significance of DHCP Server in Internetworking.

- The Dynamic host configuration protocol (DHCP) server is crucial in internetworking because it simplifies and automates the process of assigning IP address to devices on a network.
- Automatic IP assignment: DHCP dynamically assigns IP address, reducing manual configuration and preventing IP conflicts.

3. Design and configuration of an Inter network in a lab.

Steps to design and configure an Inter-network:

1. Hardware requirements
   - one switch
   - one routes
   - Ethernet cables
2. Network layout
   - Router
   - Switch
   - Devices

3. Configuration steps:

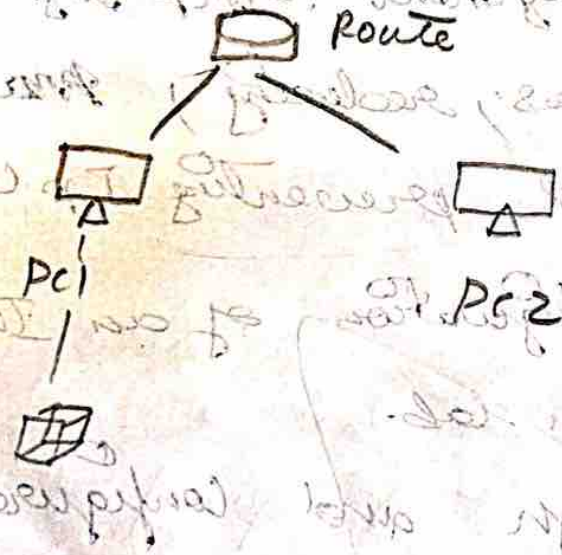
i) Connect the routes to switch

ii) Configure the Routers interface with Ipaddress.

iii) configure the DHCP server on the Routes

iv) connect PC# to the switch.

v) Test connectivity



Result:-

Thus the above program verified Succesfully using Cisco packe Tracer.

PRACTICAL - 11

AIM:-

a.) Simulate static routing configuration using CISCO Packet Tracer

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

Creating adding verifying static routes.

Routers automatically learn their connected networks, we only need to add routes for the network that are not available on the router interface.

Routers	Available networks on local interfaces	Networks available on the routers interface
Router 0	10.0.0.0/8 20.0.0.0/8 10.0.0.1/8	30.0.0.0/8 30.0.0.0/8
Router 1	20.0.0.1/8 30.0.0.0/8 30.0.0.0/8	10.0.0.0/8 40.0.0.0/8
Router 2	40.0.0.0/8 50.0.0.0/8	10.0.0.0/8, 10.0.0.0/8 30.0.0.0/8

Routers requirements

Create 2 routers for network 30.0.0.0/8 and configure the first route as the main route and the second route as a back up routes.

Result:-

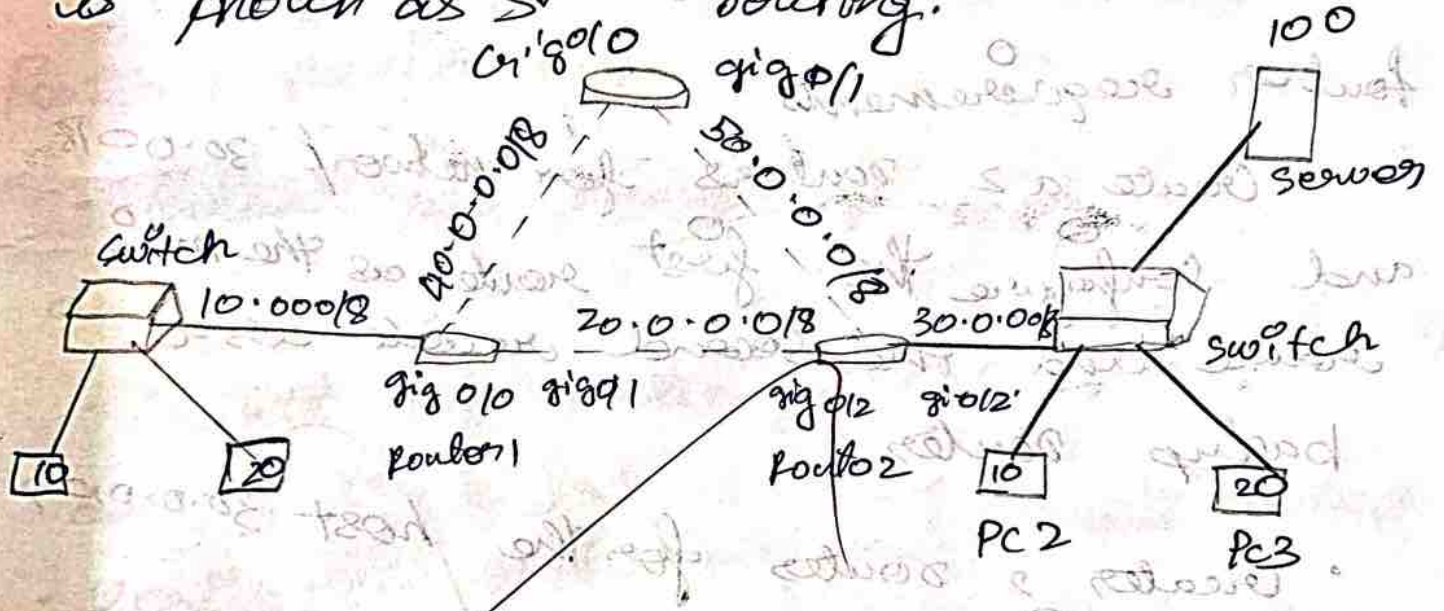
Thus the above program verified successfully.

PRACTICAL - 11

AIM:

a) Simulate Static routing configuration using Cisco packet Tracer.

Static routes are the routes you manually add to the router's table the process of adding static routes to the routing table is known as Static routing.



Creating adding verifying static routes

Routers automatically learn their connected networks we only need to add routes for the network that are not available on the routers interface

Router	Available networks on local interfaces	Network available on other interfaces
Router 0	10.0.0.0/8, 20.0.0.0/8, 40.0.0.0/8	30.0.0.0/8
Router 1	20.0.0.0/8, 30.0.0.0/8, 50.0.0.0/8	10.0.0.0/8, 40.0.0.0/8
Router 2	40.0.0.0/8, 50.0.0.0/8	10.0.0.0/8, 20.0.0.0/8, 30.0.0.0/8

Router requirements

Create a 2 routers for network 30.00/8 and configure the first route as the main route and the second router as a backup router

- Create 2 routers for the host 30.0.0.0/8 and configure
- Create 2 routers for network 50.00/8 and configure.

Router 0 configuration

Router> enable

Router# configure terminal

Enter configuration command

Router Config)# ip route 30.0.0.0 255.0.0.0 20.00.2.10

Router(Config)# ip route 30.0.0.0 255.0.0.0
10.0.0.2 20
Router(Config)# exit
Router# show ip route static
S 30.0.0.0/8 [1/0] via 20.0.0.2
S 30.0.0.10/32 [1/0] via 40.0.0.2
Router#

Router 1 requirements

- Create two routers for network 10.0.0.0/8 and configure
- Create two routers for network 40.0.0.0/8 and configure
- Verify the routers adds only main routers to the routing table.

Router 1 configuration

Router > enable
Router# configure terminal
Enter configuration command, one per line
Router(Config) # ip route 10.0.0.0 255.0.0.0
Router(config) # ip route 20.0.0.10 255.0.0.0

Routes #(config) # ext

Routes # show ip route static

S 3.0.0.0/8 [0/0] via 20.0.0.2

S 30.0.0.0/32 [0/0] via 40.0.0.2

Routes#

Routes | requirements

• Create two routers for network 10.0.0.0 and configure

• Create two routers for network 40.0.0.0 and configure

• Verify the router adds only main routers to the routing table.

Routes | configuration

Routes > enable

Routes # configure terminal

Enter configuration command, one per line

Routes (config) # ip routes 10.0.0.0 255.0.0.0 20.0.0.0

Routes (config) # ip routes 10.0.0.0 255.0.0.0 50.0.0.0

Routes (config) # int

S 10.0.0.0/8 [1/0] via 20.0.0.1

S 40.0.0.0/8 [1/0] via 20.0.0.1

Router #

Router 2 requirements

Create static routes for network 10.0.0.0/8 and network 30.0.0.0/8 and verify the routers.

Router 2 configuration

Router > enable

Router # configure terminal

Enter configuration commands

Router (config) # ip route 10.0.0.0 255.0.0.0 40.0.0.1

Router (config) # ip route 30.0.0.0 255.0.0.0 30.0.0.1

Router (config) # exit

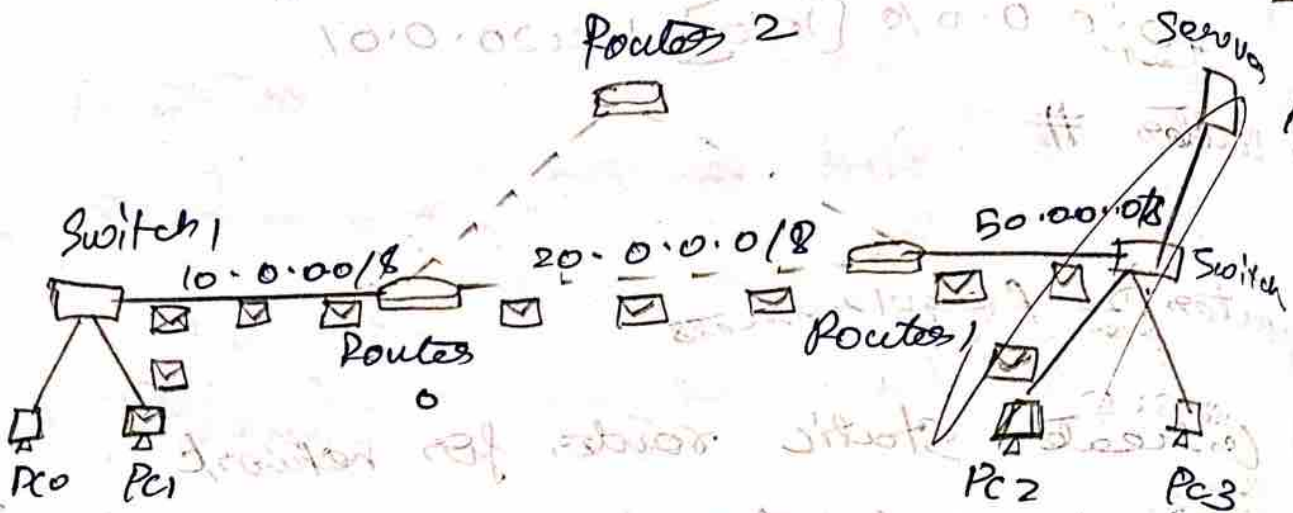
Router # show ip route static

S 10.0.0.0/8 [1/0] via 40.0.0.1

S 30.0.0.0/8 [1/0] via 30.0.0.2

Router #

Verifying Static Routing :



Result :-

Thus, the connection static routing configuration using CISCO PACKET TRACER is configured.

Aim:-

b.) Simulation RIP using CISCO PACKETTRACER

Assign Ip address to PCs.

Assign Ip address to interface to interface of routers

Router enable

Router # configure terminal

Enter configuration commands

Router (config) #

Interface fast Ethernet 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 shut down command will keeping interface up

we can use show controllers interface command for privilege mode to check the cable's end.

Router# show controllers Serial 0/0/0

Interface Serial 0/0

Hardware is power QUICC MPC860

DCE, V.35 clock rate 2000000

[Output omitted]

Fourth line of output confirms that DCE end of serial cable is attached

Router# configure terminal command is used to enter in global configuration mode

Router # interface Serial 0/0/0 command is used to enter in interface mode.

Router(config-if) #IP address 192.168.20.9 255.255.255
command assigns IP address to interface

Router (config if) # no shutdown
command brings interface up
Router (config-if)# exit.

Router,

Router > enable
Router # configure terminal
Enter configuration command
Router(config)# interface serial 0/0/0

Router (config-if) # no shut down

Router (config-if) # exit

Router (config -) # interface Serial 0/0/1

Router (config-if) # ip address 192.168.1.24
255.255.255.252

Router (config-if) # clock rate 64000

Router (config-if) # bandwidth 64

Router (config-if) # no shutdown

Router (config-if) # exit

ase same command to assign IP address to router 2

Configure RIP routing protocol

- Enable RIP routing protocol from global configuration mode.
- Test RIP routing protocol which network we want to advertise.

Router 0

Router(config) # router rip

Router 0 (config router) # network 10.0.0.0

Router 0 (config-router) # network

192.168.1.252

Similarly configure Routers 2 and Router 3 with different IP addresses like network 192.168.1.244 and more

RIP protocol automatically manage all routes for us, If one route goes down it automatically switches to another available.

RIP will automatically reroute the traffic we treat command again to see the magic of dynamic routing.

a.)

Result.

Thus, the connection to simulate RIP using Cisco packet tracer is configured.

Practical-12

a.)

AIM:-

Implement echo client server using TCP/UDP sockets.

TCP echo client - server algorithm.

Servers:-

1. Create a TCP socket.
2. connect the socket to a local address and port
3. Listen for incoming client connections
4. Accept a client connect
5. Loop.
   - Receive data from the client
   - If data is received, send it back to the client
   - else break the loop
6. Close connection

Client:

1. Create a TCP Socket
2. connect to the server using specified address and port
3. Send a message to server

4. Receive the echo message from the server.

5. Display the received message

6. Close Socket

TCP Server.py

```
import socket
def tcp_server():
    server_socket = socket.socket(socket.AF_INET,
                    socket.SOCK_STREAM)
    server_socket.bind(('localhost',
                    12345))
    server_socket.listen(1)
    print("TCP Server is waiting
    for connection...")
    client_address = server_socket.
    accept() print(f"connected to
                {client_address}")
try:
    while True:
        data = connection.recv(1024)
        if data:
            print(f"received: {data.decode()}")
```

```
else:
    break
finally
    connection.close()
```

tcp-client.py

```
import socket
def tcp-client():
    client_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_STREAM)
    client_socket.connect(("localhost", 12345))
    try
        message = input("Enter a message to send")
        client_socket.send(message.encode())
        data = client_socket.recv(1024)
        print(f"received from server: {data.
decode()}")
    finally:
        client_socket.close()
if __name__ == "__main__":
    tcp-client()
```

Output:

> python tcp_Client.py

Enter a message to send : Hi, I am
dabaresh.

Received from server : Hi, Iam dabaresh

> python tcp-Server.py.

tcpserver is waiting for a
connection connected to ('127.0.0.1', 63823)

Recived : Hi, I am dabaresh

Result:-

Thus the program to implement tcp
Clients Server using tcp is executed
succesfully

Practical - 12

b.)

Aim:-

To implement the chat client server using TCP/UDP server

Algorithm:

Chat server:

1. Start the server by creating a socket & bind to a specific address and port, then listen for incoming connections
2. When a new client connects add client to a list of a connected clients start a new loop process to talk to the clients
3. For each connected client start a new loop checking for new messages.
4. If a client disconnects remove that client from the list stop talking to that client
5. Keep running the process till the server stops chat client.

Chat Client:

1. Connect to the server by creating a socket and connect it to server address & port

2. Start a process by creating a to listen to messages from the server.

3. Keep asking for the new message

4. Keep running till the user decides to quit

chat-client.py

```
import socket
import threading
def receive_message(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode('utf-8')
            print(f"server: {message}")
        except Exception as e:
            print(f"an error occured: {e}")
            break
def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 12345
```

client = socket.connect(host, port))
print("connected to chat server")
threading.Thread(target=receive_message,
args=(client_socket), daemon=True)
start()
while True
message = input("You")
client_socket.send(message.encode
("utf-8"))
if __name__ == "main"
start_client()

chat_server.py
import socket
import threading
def handle_client(client_socket):
while True:
try:
message = client_socket.recv(1024)
decode("utf")
if not message
break
print(f"Received message from client
{message}")
client_socket.send(response.encode
("utf-8"))
except Exception as e:

```
print(f"An error has occured")
break
client_socket.close()
def start_server():
    server = socket.socket(socket.AF_INET,
    socket.socket.SOCK_STREAM
server.bind(('127.0.0.1', 1234.9))
server.listen(5)
print("chat server has started on
    127.0.0.1:12345")
while True:
    client_socket, addr = server.accept()
print(f"New connection from {addr}")
client_handle = threading.Thread(target
handle client.handle = start() args
=(client_socket,)
if __name__ == "__main__":
    start_server()
```

output:

```
>python chat-server.py
chat server started on 127.0.0.1:12345
New connection from ('127.0.0.1', 57226)
Received from client : Baresh
Type from message to client: Received
```

```
>python chat-client.py
connected to chat server
you : Chandru
you : server received.
```

23/11

Result

Thus, the program to implement the client server using TCP is executed successfully.

Practical-13

AIM:

Implement your own ping program.

ALGORITHM:-

- Open a raw socket to send ICMP request.
- Create the ICMP echo request packet including a header and data
- Send packet - send the ICMP request to target host
- Calculate the time
- Show response

Server script.py.

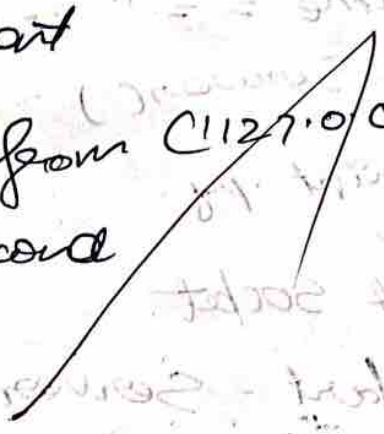
```
import socket
def start_server(host='127.0.0.1',
                 port=12345):
with s:
    s.bind(host,port)
print(f"UDP server running on
{host}:{port}")
while True:
    data,addr = s.recvfrom(1024)
print(f"Received message from
{addr}:
```

{data.decode()}')
S.sendto(b'pong', addr)
if __name__ == "__main__":
start_server()
Client script.py.
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"Up Server running on {host}:
{port}")
while true:
data, addr = s.recvfrom(1024)
print(f"received message from {addr}:
{data.decode()}")
s.sendto(b'pong', addr)
except socket.timeout
print("Request timedout")
output.
> python ping_server.py

UDP Server running on 127.0.0.1:12345
Received message from ("127.0.0.1", 57...)

Python ping-client

Received ping from ("127.0.0.1", 12345)
in 0.00 second



Result:-

Thus the program to implement ping program is executed successfully.

Practical-14

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

Algorithm:-
- create a raw socket
- continuously capture incoming packets using recvfrom()
- parse and display information like the source and destination IP address and protocol type
- close the socket after finishing the capture process.

Code:
```
from scapy.all import sniff
from scapy.layers.inet import IP TCP, UDP, ICMP
def packet_callback(packet):
    if IP in packet:
        ip_layer = packet[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dest_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: {dest_ip}")
print("-" * 50)
def main():
sniff(prn = packet_callback, filter
= "ip", store = 0)
if __name__ == "__main__":
main()
```

output:

protocol: TCP
Source IP: 192.168.1.2
Destination IP: 93.184.216.34

---

protocol: TCP
Source IP: 192.168.1.2

Destination : IP : 172 . 217 . 14 . 206

---

protocal : TCP

Source IP : 192 . 168 . 1 . 2

Destination IP : 172 . 217 . 14 . 206

3/11

Result :-

Thus, the code using raw Sockets to implement packet Sniffings is executed successfully.

Practical-13

AIM:-

To analyze the different types of web logs using webalizer tool.

Procedure:

Step 1 : Run Webalizer windows version

Step 2 : Input web log file (download from web)

Step 3 : press run webalizer

Source logfile view additional HTML code HTML

Input:
log file

C:|user|TCS|downloads|access log

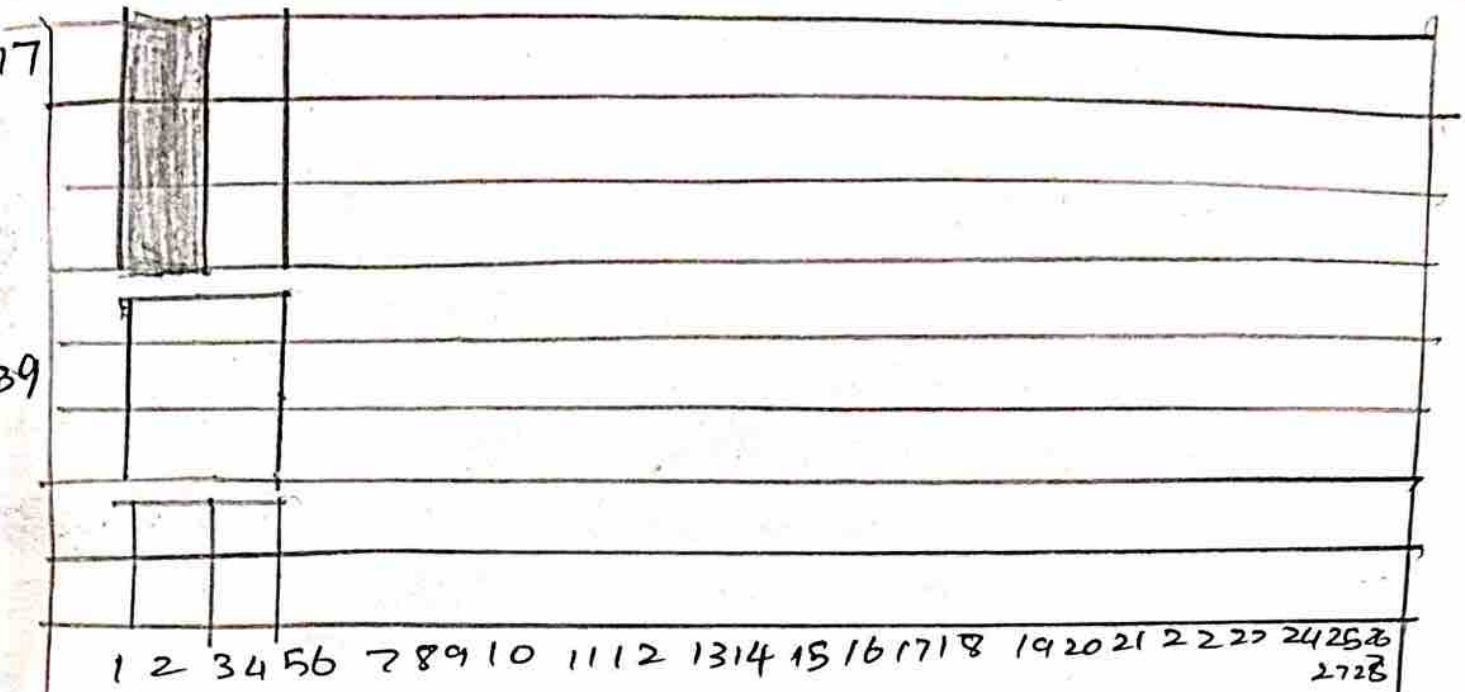
[X] [↑] [↑]

Target Directory

C:|user|TCS|

Clear existing directory
delete all files in selected Target directory

Daily usage for november 2024



Day	hits	Files	pages	visits	Sites	kbytes
3	67.100%	05100.00%	1100.00%		1100.00%	13910000%

Monthly statics for octoker 2024

Total Hits		aal
Total files		987
Total pages		971
Total visits		18
Total kBytes		3623
Total unique URLs		1
Total unique sfares		30
Total Hits per hour	Avg	7 max
Total Hits per day	2	180
Hits per day	44	247
Pages per day	49	245
Sites per day	49	245
UPSITS per day	0	1
KBytes per day	0	5
	121	906

Result

Thus the above program verified successfully.

completed

23/11