S.NO	DATE	EXPERIMENT		FACULTY SIGNATURE
1.	13/7/24	Study of various Network Command	9	
2.	27/7/24	Study of different types of network cables	10	
3.	6/8/24	Study the packet tracer tool, Installation and user Interface overview	10	
4.	17/8/24	Setup and configure LAN using a switch and ethernet cable	9	
5.	17/8/24	wireshark tool		
6.	10/9/2024	Hamming code		
7.	10/9/2024	Sliding window protocol		
8.	12/9/2024	a) ~~Cisco packet tracer~~ VLAN		
8.	12/9/2024	b) Wireless LAN using CISCO packet tracer		
9.	20/9/2024	Implementing of subnetting in cisco packet tracer		
10.	29/9/2024	a) Internetworking with routes b) Design and configure an Internetwork DHCP		
11.	5/10/24	a) Stimulate static routting b) Stimulate RIP using CISCO packet		

12	10/10/24	a) Implement echo client server. b) Implement chat client server.	
13.	21/10/24	Implement own ping program	
14	25/10/24	Code using raw socket Packet sniffing	
15	1/11/24	Types of weblog using webalizer tool	

completed

19/11.

Expt Study of Various Network commands used in Linux and Windows

Basic Networking Commands

1. arp -a:

Output

Internet Address	Physical Address	Type
172.16.72.1	7c-5a-1c-cf-be-41	dynamic
172.16.73.97	2a-3c-13-0a-cc-17	dynamic
239.225.255.251	01-00-5e-7f-ff-fb	static

2. hostname:

Output

Desktop-C01BHTD

3. ipconfig /all

output

Windows IP configuration

Host name DESKTOP-C01BHTO
Node type . . . . . Hybrid
IP Routing enabled . . . . . NO

Ethernet adapter Ethernet 3:

Media State . . . . . Media disconnected
Description . . . . . Intel(R) Ethernet Connection
Physical Address 20-88-10-86-8D-3B
DHCP Enabled . . . . . Yes

Wireless LAN adapter Local Area Connection* 13:

Media State ..... Media disconnected

Physical address 4E-82-A9-77-BD-A3

DHCP Enabled ..... Yes

Autoconfiguration ..... Yes

4. nbtstat -a

Output

NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n]
[-r] [-R] [-RR] [-s] [-S] [Interval]]

5. netstat

Output

Active Connections

Proto	Local Address	Foreign Address	State
TCP	172.16.75.50:51666	123:http	ESTABLISHED
TCP	172.16.75.50:51692	maa05s28-in:https	ESTABLISHED
TCP	172.16.75.50:51693	172.16.72.1:domain	TIME_WAIT
TCP	172.16.75.50:51694	172.16.72.1:domain	TIME_WAIT

6. nslookup

Output

www.google.com

Server: Unknown

Address: 172.16.72.1

7. Pathping

Output

[-g host-list] [-h maximum_hops] [-i address] [-n]

[-p period] [-q num_queries] [-w timeout]

[-4] [-6] target_name

options:

-g host-list	Loose source route along host-list
-h maximum_hops	Maximum number of hops
-i address	Use the specified source address

8 ping

Output

[-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] [-R] [-S srcaddr] [-c compartment]
[-p] [-6] target_name

options

-t	Ping the specified host until stopped
-a	Resolve address to hostname
-S	srcaddr Source address to use

9. Route

Output

ROUTE [-f] [-p] [-4|-6] command [destination]
[mask netmask] [gateway] [metric metric] [IF Interface]

-f	clears the routing tables of all gateway entries
-4	Force using IPV4

Command

PRINT	prints a route
ADD	adds a route
DELETE	delete a route
CHANGES	modifies an existing route

Linux Commands

1. ip

a) ip address show

1. lo: <LOOPBACK> mtu 65536 no queue state DOWN group default glen 100

2. enp8 20: <BROADCAST, MULTICAST-UP> mtu 1500 qdex fq-model state UP group default glen 100

ip address add 192.168.1.254/24 dal ep280 (enp203) (assign an Ip to an Pnterface

ip address del 192.168.1.254/24 dev lo [enp803] (after the status of Pnterface by pinging etho online)

2. If config

ens 160: flags = 4113<UP, BROADCAST, RUNNING, MUTICAST> mtr 1500
Pnet 192.168.22.128 netmask 255.255.255.0
broadcast 192.168.22.255

3. mtr

mtr <options> hostname IIP

7. # mtr google.com

keys: Helps display mode restart statistic order of fields quit

Host	Packets			pings			
	Loss	Sent	Last	Avg	Best	Worst	StDev
-gateway	0.0%	163	1.2	1.0	0.3	6.1	0.6

4. tcp dump:

def install -y tcpdump
to install tcpdump

tcpdump -D

1. ens140 [Up, Running, Connected]
2. any (Pseudo-device that captures on all interfaces)

tcpdump -i eth0
dropped privs to tcpdump
tcpdump: verbose output suppressed on all interface -v[v] for full protocol decode

tcpdump -i eth0 -c 10 host 8.8.8.8
dropped privis to tcpdump
tcpdump: verbose output suppressed, use -v[v] for full protocol decode listening on 10 unit. type

tcpdump -i eth0 host 88.8.8 and port 53
this is for specific host

5. ping

usage ping [-t] [-a] [-n count] [-l size] [-f]
[-i][L] [-v TOS] [-r count] [-s count] [-j host-list]
[-k host-list]] [-w timeout] [-R]
[-S srcaddr] [-c compartment]
[-p] [-4] [-6] target_name.

ping google.com

PING google.com (216.58.200.142) 56(84)
bytes of data 64 bytes from maa05s10-in-f14.
1e100.net (216.58.20142)

ping -c 10 www.google.com

PING google.com (142.250.193.100) 56(84)
bytes of data --google.com ping statistics--
10 pckts transmitted, 0 received, + 10 errors
100% packet loss, time 9197ms pipe 3

Configuring an ethernet connection

1. # nmcli connection show
NAME UUID TYPE DEVICE
wired connection 95e6410-cc20 ethernet enp1s0
2. # nmcli connection show
3. # nmcli connection modify "wired connection
4. # nmcli connection modify ipv4 method
5. # nmcli connection up intnet_LAN
activate the profile.

RESULT

Thus the various networking commands in Linux and Windows were executed successfully.

PRATICAL - 2

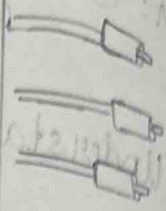
AIM: Study of different types of Network cables.

a) Understand different types of network cable

1) Unshielded twisted pair (UTP) cable
2) Schielded twisted pair (STP) cable
3) Coaxial Cable
4) Fibre optic cable

Cable Type	Category	Maximum data Transmission	Advantage / disadvantage	Application (or) use	Image
UTP	Category 3 Category 5 Category 5e	10 bps Upto 100 mbps 1 Gbps.	Adv:- Cheaper Easy to install they have smaller overall diameter Dis:. More prone to (EMI)	10-base-T Ethernet Fast ethernet, Gigabit ether	
STP SSTP	Category 6, 6a Category 7	10Gbps 10 Gbps	Adv:- Shielded • Faster than UTP • Less susceptible Dis: • Expensive • Greater installation effort	Gigabit ethernet, 10G Ethernet Widely used in data centres Gigabit Ethernet 10G Ethernet	
Coaxial cable	RG-6 RG-59 RG-11	10-1000Mbps	Adv • High bandwidth • Immune to interference Dis: • Limited distance • Cost • Size is bulky	Speed of signal is 500 m Television network High speed Internet connections	

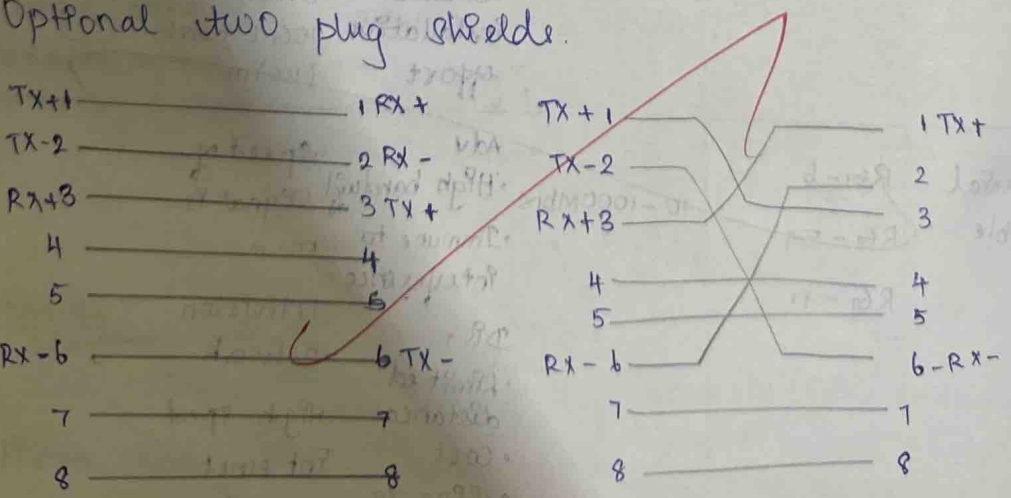
fibre optics cable	single mode multi mode	100 Gbps	Adv • High speed • High bandwidth • High security Dis • Expensive • required skilled	Maximum distance of fibre optics cable is around 100 meters

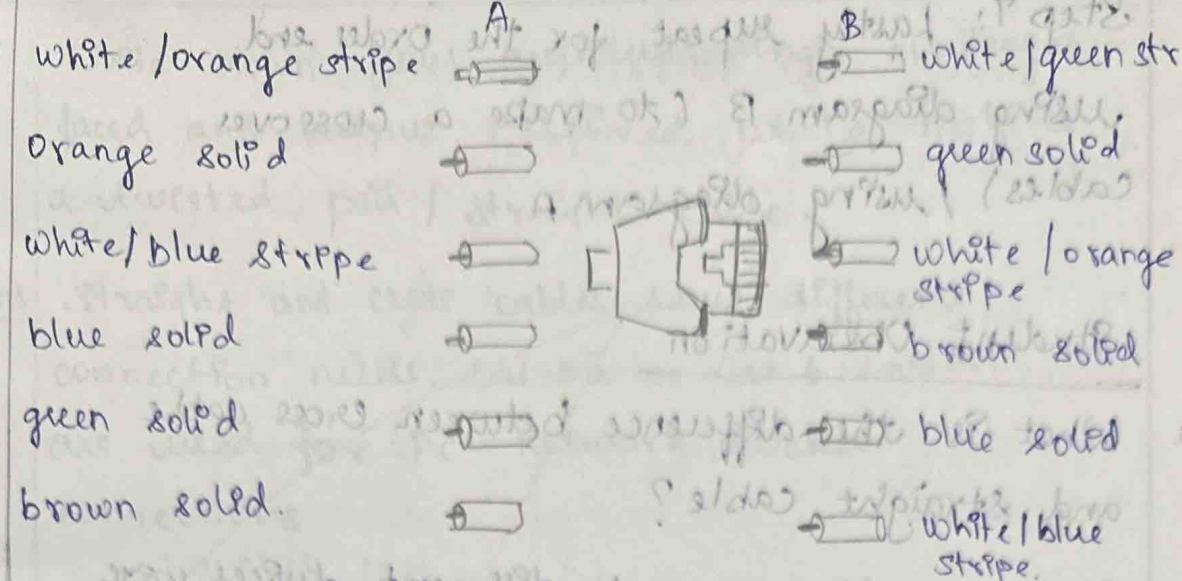


b. Make Your Own Ethernet Cross-Over cable / Straight cable

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 shaped to push down the pins in the plug and strip and cut the shielding off the cables.
- Two RJ45 plugs
- Optional two plug shields.





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

Step 2: Next strip approximately 1.5cm of cable shielding from both ends. The crimping tool has a round area

Step 3: After, you will need to untangle the wire. there should be four twisted pairs.

Step 4: Once the order is correct, bunch them together in a line, and if there are any that stick out further than others

Step 5: Next, push the cable right in. The notch at the end of the plug needs to be just over the cable shielding. and if it isn't, that means that you stripped off too much shielding.

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 order end using diagram B (to make a crossover cables) / using diagram A

Student Observation

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

Ans Straight cables connect different devices with identical wiring on both ends while cross cables connect similar devices with reversed wiring at one end

2. Which type of cable is used to connect two PC?

Ans Cross cable is type used to connect two PC

3. Which type cable is used to connect a router / switch to your PC

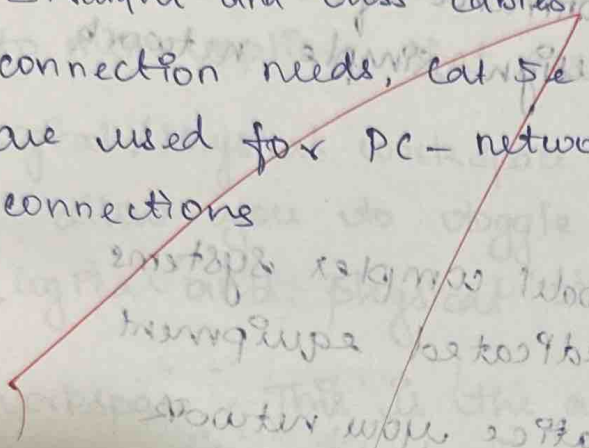
Ans Straight cable is used to connect a router / switch

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

Ans Typically category 5e (cat 5e) or category 6 (cat 6)

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

Ans Straight and cross cables serve different connection needs, Cat 5e or Cat 6 cables are used for PC – network socket connections



6/8

RESULT

Thus the cable connection is done and executed successfully

PRATICAL-3

To study the Packet tracer tool Installation and user Interface Overview

To understand environment of CISCO PACKET TRACER to design simple network

Introduction

It allows you to model complex systems without need for dedicated equipment

It helps you to practice your network configuration

It is available for both Linux & Windows

Protocols in Packet Tracer are coded to work and behave in the same way as they would on real hardware

Installing Packet Tracer

To download Packet Tracer, go to https://www.netacad.com and log in with your Cisco Networking Academy credentials

USER INTERFACE OVERVIEW

1. Menu bar - This is common trend found in all software applications
2. Main toolbar - This bar provides shortcut icons to menu options that are commonly accessed
3. Logical / Physical workspace tabs - These tabs allow you to toggle between the logical and physical work areas.
4. Workspace - This is the area where topologies are created and simulations are displayed
5. Common tools bar - It controls for manipulating topologies
6. Real time - These tabs are used to toggle between the real and simulation modes
7. Network component box - It contains all of the network and end devices available
8. User created packet box - User can create highly created customized packet to test their topology from this area

Analyse the behaviour of network devices using CISCO Packet TRACER Simulator

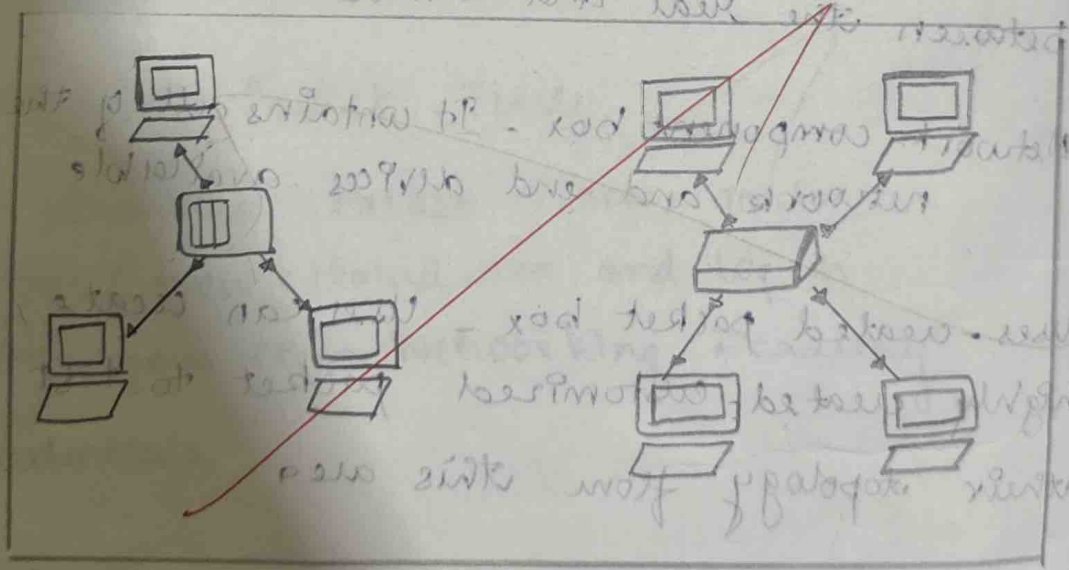
1. From the network component box, click and drag and drop the below components
   a. 4 Generic PCs and one HUB
   b. 4 Generic PCs and one switch

2. Click on Connections.

a. Click on Copper Straight - Through Cable

b. Select one of the PC and connect it to HUB using the cable. The link LED should glow in green

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



3. Click on the PCs connected to hub, go to the desktop tab, click on IP configuation and enter an IP address and subnet mask. Here, the default gateway and DNS server information is not needed as there are only two end devices on the network

PC0	PC1
IP configuration	IP configuration
IP configuration	IP configuration
IP address	IP address
Subnet mask	Subnet mask
Default Gateway	Default Gateway
DNS server	DNS server

4. Observe the flow of PDU from the 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 behaviors of switch and HUB

Student Observation

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

Switch forwards packets to the specific device based on the MAC addresses, while a hub broadcasts packets to all connected devices

Find out the network topology implemented in your college.

The network topology implemented in our college is star topology.

RESULT

Thus the packet tracer tool is installed and executed successfully

17/8/24 Practical - 4

Setup and configure a LAN (Local area network) using a switch and Ethernet cables in your lab

What is LAN?

A Local area Network (LAN) refers to a network that connects 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 connect devices to promote collaboration and transfer information between users such as computers, printers, servers. A local area network switch serves as the primary connecting devices.

How to step up a LAN

Step 1: Plan and Design an appropriate network topology taking into account network requirement

Step 2 :- You can take 4 computers, a switch with 8, 16, 24 ports

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

Step 4 :- Assign IP address to your PCs

1. Log on the client computer as administrator
2. Click Network and internet
3. Right click local area connection → Go to properties → Set internet protocal → click on properties → Select ip address option

Step 5 :- Configure a network switch

1. Connect your computer to switch: To access the switch's web interface, you will need to connect your computer.
2. Log in to the web interface: Open a web browser and enter the IP address of the switch in the address bar.

3. Configure basic settings: Once you're logged in, you will be able to configure basic settings for the switch

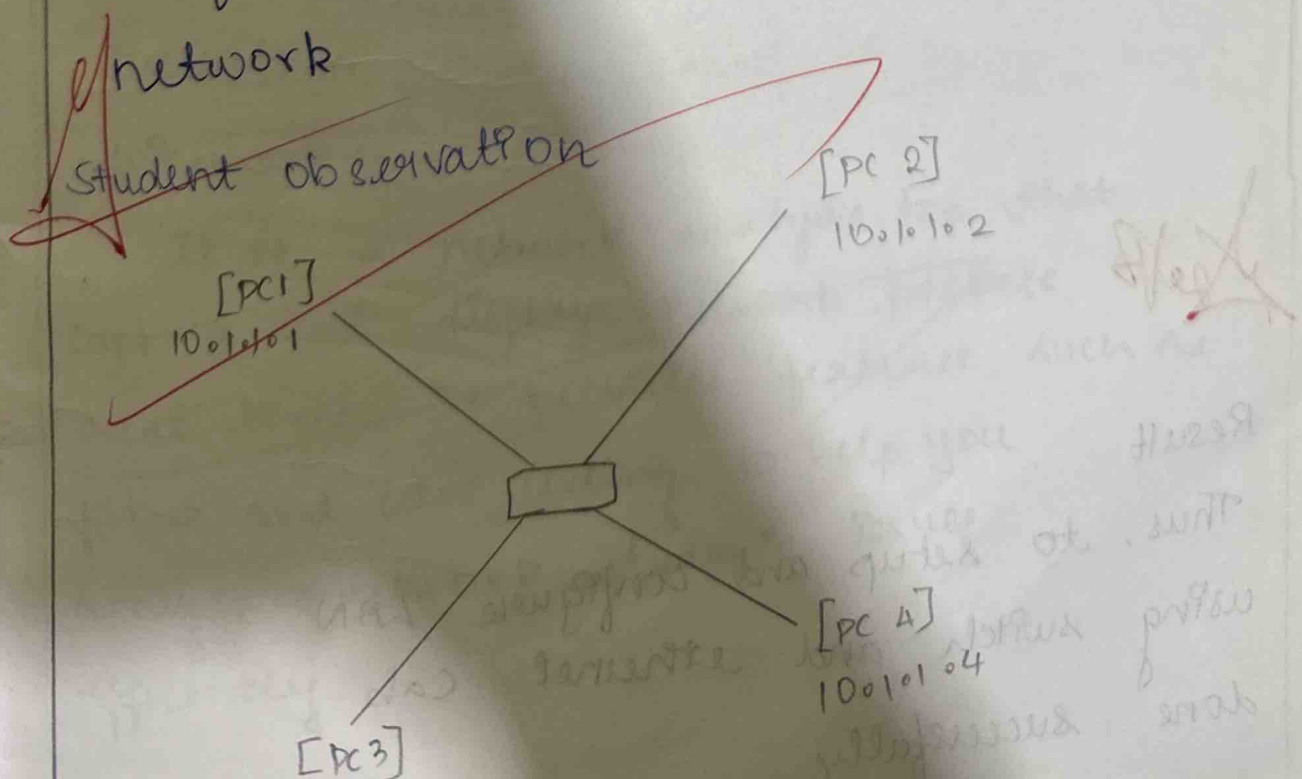
4. Assign IP address as 10.1.1.5, subnet mask mask 255.0.0.0

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



Lan was successfully setup and all devices could communicate with each other using their assigned IP addresses shared resources like folders were accessible from all connected PCs.

Challenges Faced:

Ensuring PC has a unique IP address to avoid conflicts.

Result

Thus, to setup and configure LAN using switch and ethernet cab done successfully

Exp: 5 17/8/24

Experiments on packet capture tool wireshark

AIM

Experiments on packet capture tool wireshark

Packet sniffer:

Monitors network traffic sent to and from your computer captures and display the details of various protocol fields within the data packets

Diagonstic tools:

Tcp dump

Ex: Tcpdump - enx host 10.129.41.2 -10

wireshark

Ex. wireshark - r ex e- 8.out

Description

wireshark:

It is a network analysis too that captures and displays network packets in real time. It provides features such as filters and color coding to help you analyse network traffic issues effectively

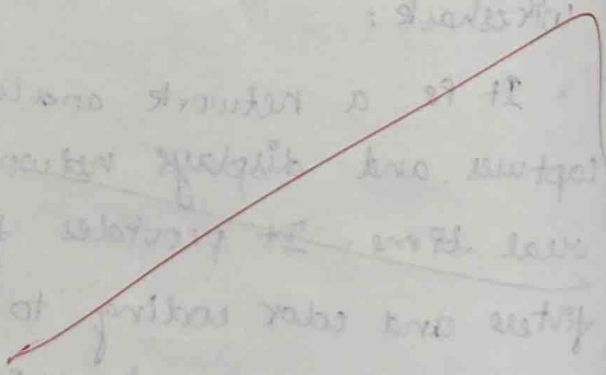
What can we do with wireshark

capture network traffic
decode various packet protocols
apply filters to capture and display specific data.
Monitor statistics and analyze problems

Users:
Network administrators
Security engineers
Developers

Getting wireshark
For windows - download from official website
For linux - Available in package repository

Capturing packets
- Launch wireshark
- Double click the network interface under capture to start capturing packets.



Wireshark Interface overview:

1) stop capturing traffic
2) packet list pane
3) Packet details pane
   shows detailed information of selected packet
4) Packet bytes pane:
   shows selected packets data in hexdump

color coding

Light purple - TCP traffic
Light blue - UDP traffic

Filtering packets

Apply filters to focus on specific network traffic

use analyse > display filters to pick or save filters, see the docs for more Info.

Capturing and analysing packets using wireshark tool

To filter, view capture packets, capture 100 packets from the ethernet.

Procedure

Select LAN goto capture → option
Select stop captures after 100 packets

then check start capture
Save packets

Create a filter to display only TCP (UDP packets inspect the packets and provide flow graph

Select LAN, goto capture → option
Select stop capture after 100 packets
click start capture
Save the packets

Create a filter to display only ARP packets and Inspect the packet

Go to capture → option
Select stop capture after 100 packets
then click start capture
Search ARP packets in search

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

procedure

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

Create a filter to display only HTTP packets and inspect the packets procedure

Select local area connection
Go to capture → option
Select stop capture after 100 packets
Search HTTP packets
Save packets

Create a filter to display only IP/ICMP packets and inspect the packets

Procedure
→ Select local area connection
→ Go to capture → option
→ select stop capture
→ Then click start capture
→ Search ICMP/IP packets
→ Save packets

Create a filter to display only DHCP packets and inspect the packets

Procedure
Select local area connection in Wireshark
Go to capture → option
Select stop capture
Then click start capture
Search DHCP packets
Save packets.

Student observation

1. What is promiscuous mode?

A It allows a network interface to capture all packets on the network regardless of destination

2. Does ARP packets has transport layer header? Explain.

A ARP packets do not have a transport layer header

3. Which transport layer protocol is used by DNS?

A DNS uses UDP, and occasionally TCP on port 53

4. What is port number used by http protocol?

A The port numbers used by http uses portocol 80

5. What is a broadcast ip address?

A An IP address used to send data to all devices on a subnet

Result

Thus the experiment on packet capture tool wireshark has been successfully

10/9/24

Pratical -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 codes that can be used to detect and correct the errors that can occur when the data

Create sender program with below featu

1. Input to sender file should be a text
2. Apply hamming code concept on the binary data and add redundant bits
3. Save this output

Create a receiver program

1. Receiver program should read the input
2. Apply hamming code on the binary data
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.

10/2/24 Pratical -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 codes that can be used to detect and correct the errors that can occur when the data

Create Sender program with below feature

1. Input to sender file should be a text
2. Apply hamming code concept on the binary data and add redundant bits
3. Save this output

Create a receiver program

1. Receiver program should read the input
2. Apply hamming code on the binary data
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.

Code

```
def text_to_binary(text): (convert to binary)
    return ''.join(format(ord(char), '08b') for char
def binary_to_text(binary):
    chars = [binary[i:i+8] for i in range(0, len(binary
    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(data)
    res = ''
    for i in range(1, m + r + 1):
        if(i == 2**j):
            res = res + '0'
            j += 1
        else:
            res = res + data[k]
            k += 1
    return res
```

```
def calc_parity_bits(arr, r):
    n = len(arr)
    arr = list(arr)
    for i in range(r):
        parity = 0
        position = 2 ** i
        for j in range(1, n+1):
            if j & position:
                parity ^= int(arr[j-1])
        arr[position-1] = str(parity)
    return ''.join(arr)

def detect_and_correct(data, r):
    n = len(data)
    res = 0
    for i in range(r):
        parity = 0
        position = 2 ** i
        for j in range(1, n+1):
            if j & position:
                parity ^= int(data[j-1])
        if parity != 0:
            res += position
    if res != 0:
        print(f"Error detected at position: {res}")
        data = list(data)
        if res <= n:
            data[res-1] = '0' if data[res-1] == '1' else '1'
            print(f"Error corrected at position: {res}")
```

```
else:
print("Error position out of range.
      No correction performed.")
corrected_data = ''.join(data)
return corrected_data

else:
  print("No error detected")
  return data

def redundan_remove_bits(data, r)
  p = 0
  original_data = ''
  for p in range(1, len(data)+1):
    if i == 2**j:
      j += 1

    else:
      original_data += data[i-1]
    return original_data

def introduce_error(data, position):
  if pos < 1 or pos > len(data):
    print("Error")
    return data
  data = list(data)
  data[pos-1] = '0' if data[pos-1] == '1' else '1'
  return ''.join(data)
```

```
def sender(text):
    binary_data = text_to_binary(t)
    m = len(binary_data)
    r = calc_redu_bits(m)
    arr = pos_red_bit(bin_data, r)
    ans = calc_parity_bits(ans, r)
    print(f"Sender output {ans}")
    return ans

def receiver(data):
    r = calc_red_bit(len(data))
    corr_dat = dect_cor(data, r)
    original_data = rem_r_b(cor_data, r)
    ascii_output = bina_to_text
    print(" f decoded)

pf_name_ == "_main_":
    input_text = input(" text to be encoded:")
    channel_data = sender(input_text)
    corrupted_data = introduce_error(channel_data, 2)
    receiver(corrupted_data)
```

Output: Enter the string: bhargavi

Original data bit: 01100010010000110000101110010011001111010000101110100101110100

No. of redundant bit: 7, [1, 2, 4, 8, 16, 32, 64]. Done

Encode bits with redundant bit: 11001100001001111010000110000101011001001101101000010110111000010100

Enter the bit to be changed: 7

Updated encoded bits: 11001110001001111010000101000010101001100111

detected error at: 7

corrected encoded bits: 11001100010011110100001100001010000010111001001101101000010110

RESULT

Thus, hamming code has been executed and verified successfully

10/9/24

Pratical-7

AIM: Write a program to implement flow control at data link layer using SLIDING WINDOW PROTOCOL. Simulate flow of frames from one node to another

Create a sender program with 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. Send the frames
6. Wait for the acknowledgement from receiver
7. Reader a file called receiver-buffer.
8. Checker ACK field called receiv. for the acknowledgement number
9. If the acknowledgement number is as expected, send new set of frames accordingly else if NACK is received, resend the frames accordingly.

Create a receiver file with following features

1. Reader a file called Sender_Buffer
2. Check the frame no
3. If the frame no. are as expected, write the appropriate ACK no. in the receiver_Buffer file
Else write NACK no. in the receiver_Buffer file

Code

```
import random

class Frame:
def __init__(self, frame_no, data):
    self.frame_no = frame_no
    self.data = data
    self.ack = False

def send_frames(frames, win_siz):
    print("\n--Sending Frames ---")
for i in range(window_size):
if i < len(fram) and not frames[i].ack:
    print(f"Sent {frames[i].frame_no}: {frames[i].data}")
print("Frames sent, waiting for acknowledgment")

def receive_frames(frames, win_size):
for i in range(win_siz):
if i < len(frames) and not frames[i].ack:
if random.random() < 0.2:
    print(f"{frames[i].frame_no}: {frames[i].data}")
    [i].ack = False
```

```
    frames[i].ack = True

def sliding_win_protocol():
    win_size = int(input("Enter window size:"))
    message = input("Enter a message to send:")
    frames = [Frame(i, message[i]) for i in
              range(len(message))]
    base = 0
    while base < len(frames):
        send_frames(frames[base:], win_size)
        time.sleep(2)
        receive_frame(frames[base:], win_size)
        while base < len(frames) and frame[base].ack
            base += 1
        if base < len(frames):
            print("Resending unacknowledged frames")
            time.sleep(2)
    print("All frames sent and acknowledged!")
if __name__ == "__main__":
    sliding_window_protocol
```

Output

```
Enter window size: 5
Enter a message to send: bhargavi
Sending frames
Sent frame 0: b
Sent frame 1: h
Sent frame 2: a
Sent frame 3: r
```

Frames sent, waiting for acknowledgments

Receiving Frames

Received frame 0 : b [error]
Received frame 1 : h [error]
Received frame 2 : a [error]
Received frame 3 : r [OK]
Received frame 4 : g [OK]

Resending unacknowledged frames

Sending frames

Sent frame : b
Sent frame : h
Sent frame : a

Frames sent, waiting for acknowledgments

Received frame 0 : b [~~error~~ OK]
Received frame 1 : h [~~error~~ OK]
Received frame 2 : a [OK]

All frames sent and acknowledged

Result

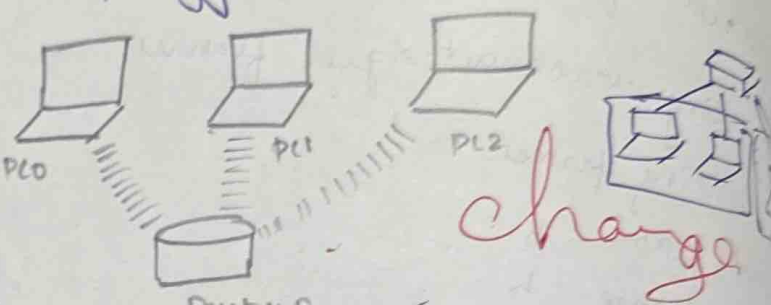
Thus, the program for sliding window has been executed successfully

12/9/24 Pratical 8

AIM

a) Stimulate Virtual LAN configuration using cisco packet Tracer configuration simulation

Packet Tracer - configure VLANs and Trunking physical mode Topology



Addressing Table

Device	Interface	IP address	Subnet Mask	Default G
S1	VLAN 1	192.168.1.11	255.255.255.0	N/A
S2	VLAN 2	192.168.1.12	255.255.255.0	N/A
PC-A	NIC	192.168.10.3	255.255.255.0	192.168
PC-B	NIC	192.168.10.4	255.255.255.0	192.1

Part-A

Stimulate VLAN configuration using cisco Packet Tracer

Part 1: Build the network and configure basic device Settings

1. Build the network

* Add two switches (S1 and S2) and two as per the topology

* Connect devices with copper straight-through cables.

* Use console cables to connect PC-A to S1 and PC-B to S2

2. Configure Basic Switch Settings:

* Device name: Assign S1 to Switch and S2 to Switch 2

* Password: set class as the encypted password, cisco for console and encrypt are passwords

* IP address

S1: 192.168.1.11 /24

S2: 192.168.10.12 /24

Shutdown unused interfaces

Set clock and save configuration

3. Configure PCs:

* PC-A: 192.168.10.3/24

Gateway: 192.168.10.1

* PC-B: 192.168.10.4/24

Gateway: 192.168.10.1

4. Test connectivity:

* Ping tests:

→ PC-A ↔ PC-B : No

→ PC-A ↔ S1 : Yes

→ PC-B ↔ S2 : Yes

→ S1 ↔ S2 : Yes

Part 2: Create VLANs and assign ports

1. Create VLANS

* S1 & S2:

VLAN 10, name operations

VLAN 20, name paying lot.

VLAN 99, name management

* Verify with show VLAN brief

2. Assign VLANs to switch ports:

* PC-A to VLAN 10:

interface 10/6

Switch port mode access

Switch port access VLAN 10

* Remove IP from VLAN, assign to VLAN 99

interface VLAN

no ip address

interface VLAN 99

PC-B to VLAN 10

Verify with show VLAN brief

port 3 : Maintain VLAN Database

1. change port assignments:

* Assign interface fo/11-24 to VLAN 99

interface range fo/11-24
switch port access fo/11-24

* Reassign fo/11 and fo/21 to VLAN 10
* Remove VLAN assignment from fo/24 with no switch port access VLAN

2. Remove VLAN from Database

* Assign and remove VLAN 30 from to 12

interface fo/24
Switch port access VLAN 30
no VLAN 30

Part-4 Configure 802

1. DTP Trunk on Fo/1:

* Set fo/1 to dynamic desirable mode:

interface fo/1
Switch port mode dynamic desirable

* Verify with show interface menu

2. Manual Trunk configuration

* Set Fo/1 as trunk and change native VLAN to 1000:

interface fo/1
switch port mode trunk
switch port trunk native
VLAN 1000

Part B: VLAN Design for robotics department

* Scenario: 10 faculty 13 blocks, same logical VLAN
* Solution:

→ VLAN creation: Create a a single VLAN for the robotics department are 8 blocks. V10

* Switch configuration: Assign port in each block to VLAN 10 for faculty pcs

This configuration ensures that are robotics faculty are in the same VLAN, regardless of physical location

Student Observation

2) Show the ip configuration for each device

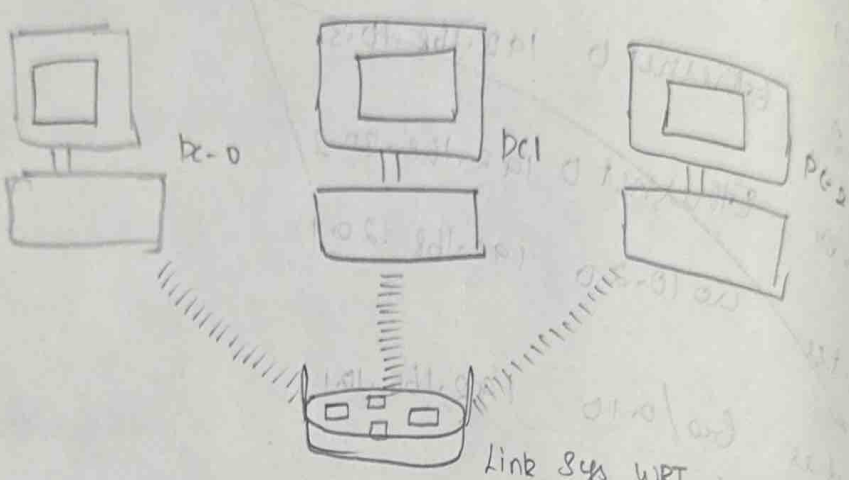
Device	Interface	IP address	Subnet mask	Default Gateway
PC1	Ethernet 0	192.168.10.2	255.255.255.0	
PC2	Ethernet 0	192.168.10.3		
Server	Ethernet 0	192.168.20.2		
Router	Go 10.20	192.168.20.1		
Router	Go/0.10	192.168.10.1		

Result 19/11.

Thus the virtual LAN configuration using cisco packet tracer configuration was executed successfully

Practical : 8

Aim : b) Configuration of wireless LAN using CISCO Packet tracer



1) Router Configuration

- Access the administration tab on the wireless router
- Set the username and password to admin
- Navigate to wireless and change SSID to other network
- Set security mode to WEP & configkey

2) PC configuration

Set static Ip addresses for each PC

- PC0 : IP: 192.168.0.2 Subnet : 255.255.255.0
- PC1 : IP: 192.168.0.3 Subnet 255.255.255.0
- PC2 : IP: 192.168.0.4 Subnet 255.255.255.0

3) Connecting PCs to wireless network

- For each PC, go to desktop tab, select PC wireless click connect
- Refresh list connect to other network

4) Verification

- Ensure all PCs shows as connected to wireless network with active PC cards. Repeat the connection

Student observation

What is SSID of a wireless router?

A The SSID - Service Set Identifier is a name of wireless network that allows devices to identify and connection to it.

What is a Security key in wireless router?

A The Security key is a code used to authenticate user and encrypt data transmitted over the wireless network ensuring service access

Configure a simple wireless LAN in your lab using real access point and the ~~cisco~~ configuration

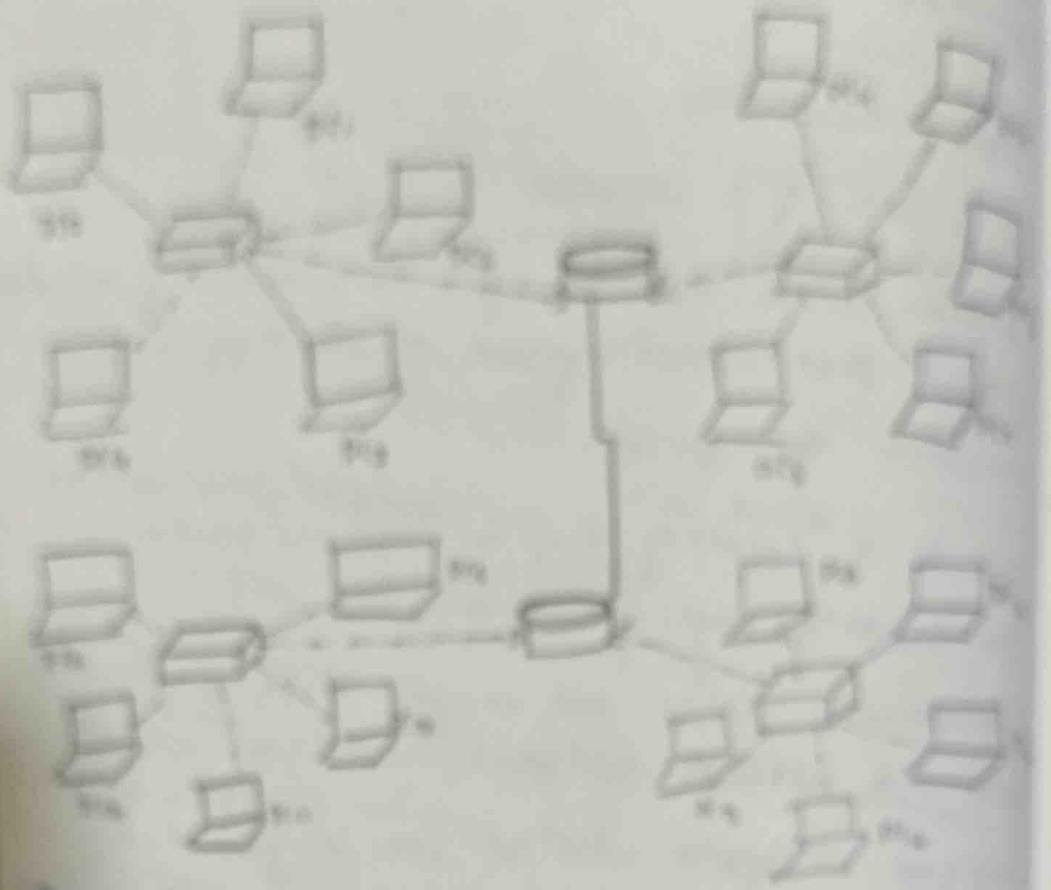
A To configure

- Connect to the access point
- Access the configuration page
- Login, set SSID choose Security type
- Set security key, setting

19/11

Result

The configuration of wireless LAN using CISCO Packet Tracer successfully executed.



classless IP subnetting allows for efficient use of IP address by enabling variable length subnet masks, which helps divide an IP address space into smaller subnet for better management

Create a network Topology

- open packet tracer, select "new", then "Network" & "Generic" to star a blank topology

Add Devices:

- Drag and drop routers, switches, PCs from devices
- Connect devices using cable tool

Subnetting

- For the network address 192.168.1.0/24 use /27 subnet mask to provide at least 5 addresses for end device the switches & routers.
- The setup creates 8 subnet, each with usable host address

Network configuration

Router configuration:

use the CLI to set up the router Interface

fast ethernet 0/0: connect to switch

fast ethernet 0/1: connect PC

Switch configuration:

Access CLI

```
enable
configure terminal
interface fast ethernet 0/0
switchport mode access
exit
interface ethernet 0/2
switch port mode access
```

Pc configeration

- IP address
- Subnet mask
- Default gateway
- DNS Server

Ensure Ip address & subnet mask match subnet of router fast ethernet 0/1

Gigabit Ethernet configuration

use the CLI to configure the Gigabit ethernet 0/0 interface with desired IP addrs and subnet mask then enable the interface

Student observation

a) Write down your understanding subnetting

Subnetting is a process of diving a larger network into smaller manageable sub networks.

b) What is advantage of implementing subnetting within a network?

- Improved network
- Efficient IP address management
- Enhanced security

c) find out whether subnetting is implemented your college

Subnet1: 192.168.0.0/24

Subnet2: 192.168.1.0/24

Subnet3: 192.168.2.0/24

Result

Implemented Subnet in cisco packet tracer simulation successfully & executed

29/9/24 Pratical - 10

Aim: a) Internetworking with routers in CISCO PACKET TRACER Simulator

Configure the network

Configuring Router1

1) Open router CLI
2) Activate privileged mode by typing enable
3) enter global configuration mode with confict
4) Configure fast ethernet 0/0:
   - interface fast ethernet 0/0
   - ip address 192.168.10.1 255.255.255.0
   - no shutdown
5) Configure fast ethernet 0/1:

Configuring pcs

Assign ip address to each PC:

- PC0: IP address : 192.168.10.2
  Subnet : 255.255.255.0
  Gateway : 192.168.10.1

Connecting PCs to routers

1) PC0 to router1
2) PC1 to router 2

Router Configuration Table

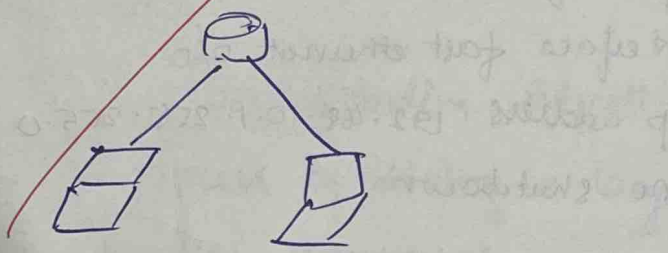
Device Name	IP address	Subnet mask
Router1	192.168.10.1	255.255.255.0
	192.168.20.1	255.255.255.0

PC configuration table

Device name	IP address	Subnetmask	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.10.2	25.255.255.0	192.168.20.1

Testing:

To verify connectivity perform PDU transfer from PC0 - PC1 if successfully the network is correctly configured.



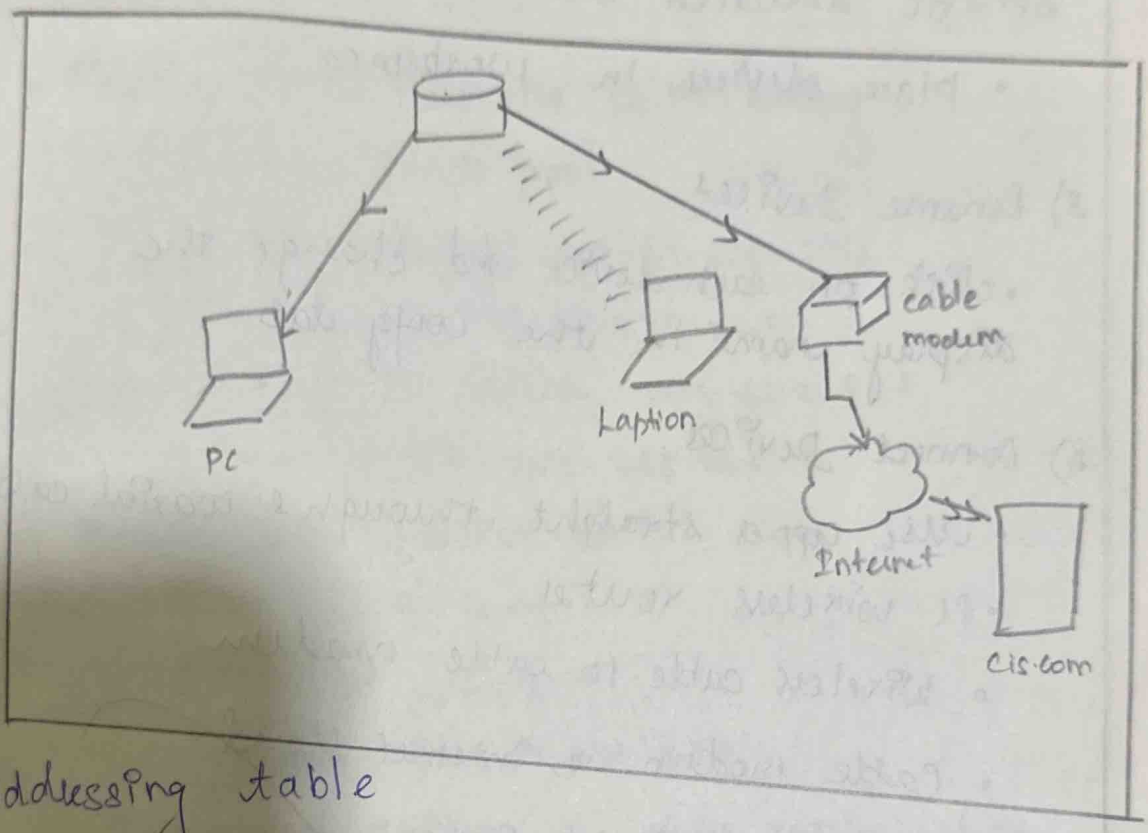
19/11

Result:-

Thus, the packet tracer simulator has been executed successfully

29/9/24 Praticcal-10

Aim b) Design and configure an internetwork using wireless router, DHCP server and internet



Addressing table

Device	Interface	Ipaddress	Subnet mask	Default Gateway
Pc	ethernet()	DHCP		192.168.0.1
Wireless router	LAN	192.168.0.1	255.255.255.0	
Wireless router	Internet	DHCP		
Ciscocen server	ethernet	208.67.220220		
Laptop	wireless	DHCP		

Part 1: Build a simple Network

1) Launch Packet Tracer

2) Add Devices
   - Select devices from device type & device specific selection boxes
   - Place devices in workspace

3) Rename Devices
   - click on each device and change the display name in the config tab

4) Connect Devices
   - Use copper straight through & coaxial cable
   - PC wireless router
   - Wireless cable to cable modem
   - Cable modem to internet cloud
   - Internet cloud to Server

Part 2: Configure Network devices

1) Wireless router
   - Set SSID to flow network
   - enable DHCP and set DNS server 208.67.222.222

2) Laptop
   - Replace ethernet module with wireless WPC 300N
   - Connect to "home network" wirelessly

3) PC
   - Set to receive DHCP configuration for an IPV4 address

4) Internet Cloud

- Install necessary network module
- Set up coaxial and ethernet connections in config tab

5) Cisco Server

- Enable DHCP service with setting
  - Pool name: DHCP pool
  - Default Gateway: 208.67.220.210
  - DNS Server: 208.67.220.220
  - Starting IP address: 208.67.220.1
  - Subnet mask: 255.255.255.0
- Enable DNS Service with
  - Name: cisco.com
  - Type: A record
  - Address: 208.67.220.220.
- Set Global setting for static configuration
- Configure fast ethernet with a static IP address: 208.67.220.67-220.220

Part 3: Verify Connectivity

Refresh IPv4 setting on PC

- Use ip config /release and ip config to obtain IP address from DHCP

Test Connectivity

Ping the cisco server from PC using the command ping cisco.com. Verify successfully

Result: Thus,
Design and configure of internetwork using wireless router, DHCP Server and designed successfully

9/4

Student Observation

① Write down the key features of configuring wireless router and DHCP server.

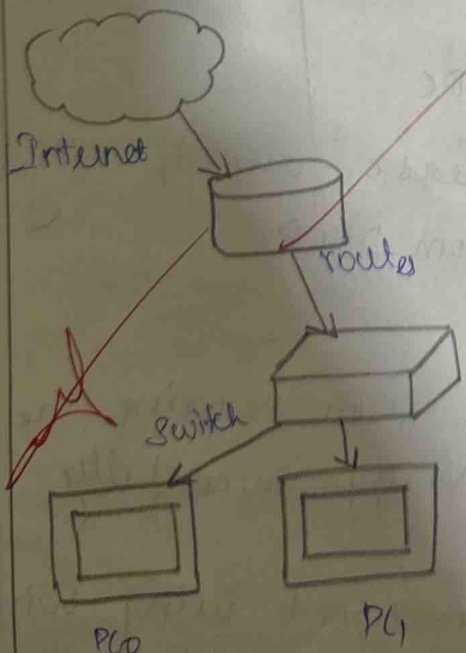
A Wireless router configuration:

- SSID setup: Create a unique network name
- Security Protocol: choose WPA3 or WPA2
- Channel Selection: Set optimal channels to reduce interference
- Guest Network: Configure a separate network for visitors
- Firmware Update: Regularly update for security

② What is the significance of DHCP server in networking?

A Automatic IP assignment, reduces conflicts, ease of management, Dynamic Update.

③ Design and configure an internetwork you can using switches, router and ethernet cable.



Router: IP: 192.168.1.1

Switch

PC0: IP: 192.168.1.2

Subnet: 255.255.255.0

Default: 192.168.1.1

PC1: IP: 192.168.1.3

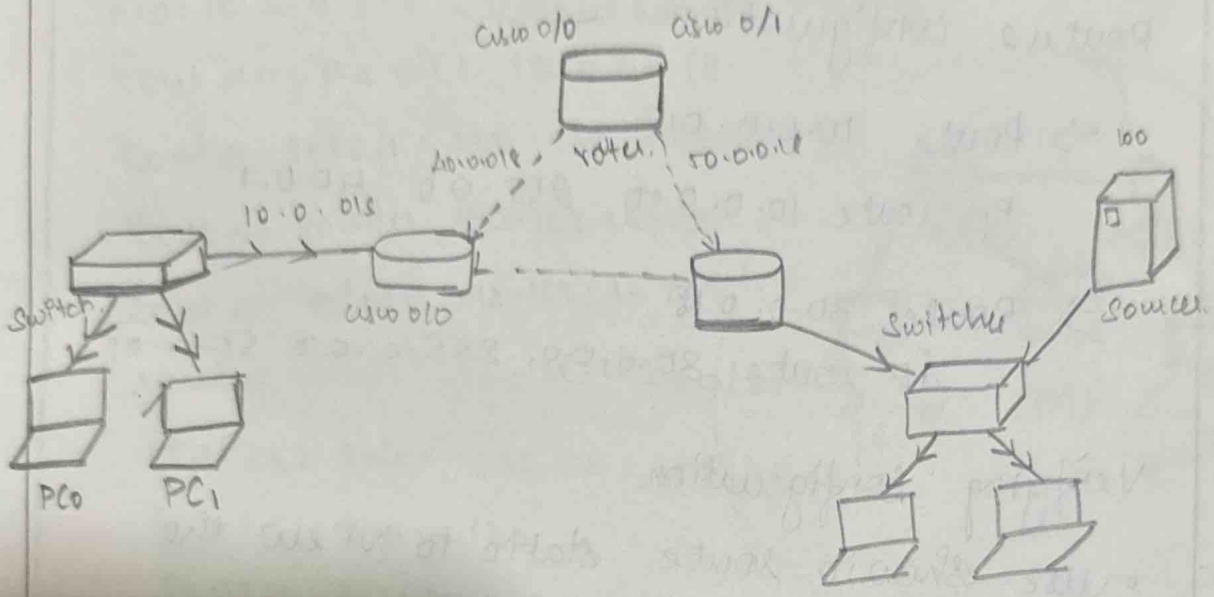
Subnet mask: 255.255.255.0

Default: 192.168.1.1

Studies.

Pratical - 11

a) Aim: Stimulate Static Routing Configuration using Cisco packet tracer.



Lab setup

Network design create a lab with router and network as per specified requirement

Connection Network

- Router 0: 10.0.0.1, 40.0.0.1, 30.0.0.1, 50.0.0.1
- Router 1: 20.0.0.1, 50.0.0.1, 10.0.0.1, 40.0.0.1
- Router 2: 40.0.0.1, 50.0.0.1, 20.0.0.1, 30.0.0.1

Router Configuration.

→ Router for 30.0.0.1

- Main: ip route 30.0.0.0 255.0.0.0 20.0.0.1
- Backup: ip route 30.0.0.0 255.0.0.0 40.0.0.2

→ Host route for 30.0.0.100

- Main: ip route 30.0.0.100 255.255.40.0.0.2
- Backup: ip route 30.0.0.100 255.255.20.0.0.2

→ Route 50.0.0.0

- Main: ip route 50.0.0.0 255.255.255.40.0.0.2
- Backup: ip route 50.0.0.0 255.0.0 20.0.0.2 20

→ Route 40.0.0.0 18

main: ip route 40.0.0.0 255.0.0.0 20.0.1.10

backup: ip route 40.0.0.0 255.0.0.0 50.0.0.20

Router2 configuration

→ Route 10.0.0.0 18

ip route 10.0.0.0 255.0.0 40.0.1

→ Route 30.0.0 18

ip route 30.0.0.0 255.0.0.0 50.0.0.2

Verifying configuration

- use show ip route static to view the routing table
- Testing routes:
  - ↳ Ping or use trace from PCs in connected network to verify routing paths

Simulating Route failure

. To test backup route, disconnected the primary route and recheck connectivity to the target network

Deleting static routes

1) Use show IP address to view all routes
2) Identify the route to delete
3) Use no ip route [destination][mask] [nexthop] to remove the route

Result: Thus, cisco packet has been executed successfully

Slot 24 Pratical-11

b) Aim: Simulate RIP using cisco packet Tracer

network Configuration

- Devices and Ip Configuration

PC0: 10.0.0.2/8 (connected to router)

Router0 Fa0/1: 10.0.0.1/8

Router1 s0/0/1: 192.168.1.254/30

Router0 s0/0/0: 192.168.249/30

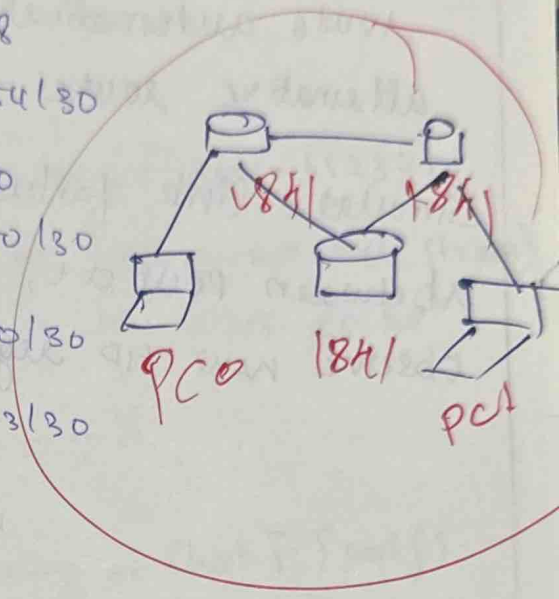
Router1 s0/0/0: 192.168.1.250/30

router s0/0/1: 192.168.1.249/30

Router2 s0/0/1: 192.168.1.253/30

PC1: 20.0.0.2/30

Router 2: 20.0.0.1/30



Steps to Configure

Assign IP address to routers.

↳ Access each router

↳ Use configure terminal to enter global configuration mode

Configure serial Interface

↳ Set clock rate and bandwidth for DCE ends of serial connections

Implement RIP protocol

↳ Enable RIP on each router with router rip

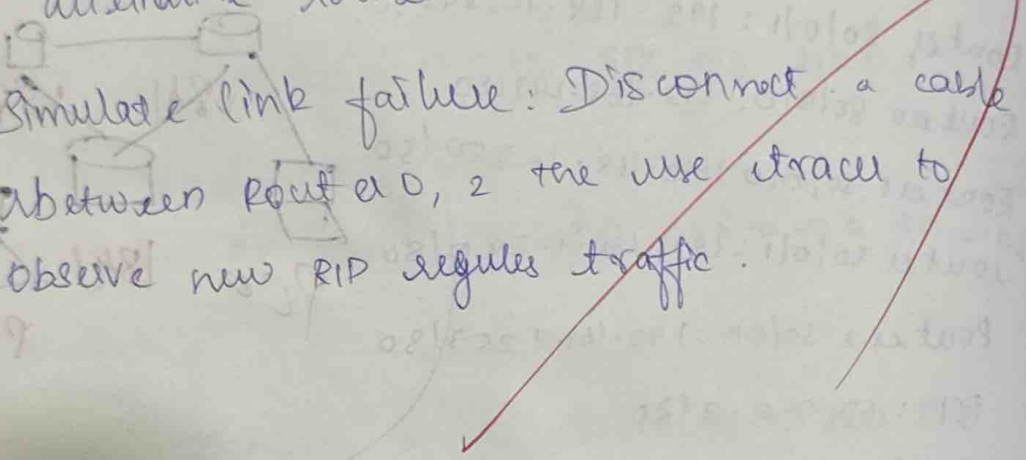
↳ Specify network to addressing using network commericial.

Verification

ping test: use the ping command from PC to test connectivity to PC0

Route management: RIP dynamically manages routes automatically switching functi. to alternative routes

Simulate link failure: Disconnect a cable between Router 0, 2 then use tracert to observe new RIP reguler traffic.



Result:

Thus, the cisco packet tracer p network configuration has been executed successfully

10/10/24

Pratical - 12

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

Algorithm

TCP Echo Server

```
import socket
def tcp_echo_server(host='127.0.0.1', port=65432):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                                        as server_socket:

        Server_socket.bind((host, port))
        Server_socket.listen()
        print(f'Tcp echo server listening on {host}:{port}')
        while True:
            conn, addr = server_socket.accept()
            with conn:
                print(f'Connected {addr}')
                while True:
                    data = conn.recv(1024)
                    if not data:
                        break
                    conn.sendall(data)

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

TCP client echo:

```
import socket
def tcp_echo_client(host='127.0.0.1', port=65432):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                                    as client_socket:
        client_socket.connect((host, port))
        message = input("enter message:")
        client_socket.sendall(message(.encode())
        data = client_socket.recv(1024)
        print(f"Received from sever:{data.decode()}")

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

output

Enter message to echo : hello

Received message from sever: hello

Server	Client
connection established:(127.0.0.1)	connection established to sever at ('local host', 12
Received message: hello	Received from Sever: hello
Echo message back	

Result:-

Thus, client Server using TCP/UDP Socket has been executed successfully

10/10/24 Pratical-12

b) Implement chat client server using TCP/UDP Socket

Algorithm

- Create a Socket using TCP (Sock_Stream)
- Connect to Server's IP address
- Take input from the user
- Print received a response from source
- Print servers response on client side

TCP Server code

```
import socket
Server_host = '127.0.0.1'
Server_port : 12345
Server_Socket = Socket.socket ( socket.AF_INET)
Server_socket.bind (( Server_host, server_port))

while True:
    message = client_socket.run (1024).decode()
    if message.lower() = 'quit'
    print ("Client has ended the chat")
    break
    print (f" Client freamen: )
    response = Input (" Server...")
    client_Socket.Send (respond.encode())
    if response.lower(): 'quit'
        print (" Server has ended the chat")
        break

client_socket.close()
Server_socker.close()
```

client:

```
import socket
server_host = '127.0.0.1'
server_port = 12345
client_socket = socket.socket(socket.AF_INET
client_socket.connect((server_host, server_port))
while True:
    message = input("Client")
    client_socket.send(message.encode())
    if message.lower() == 'quit':
        print("Client ended chat")
        break
    response = client_socket.recv(1024).decode()
    print(f"Server: {response}")
client_socket.close()
```

Output

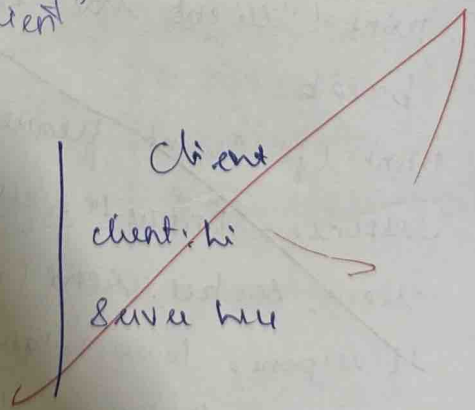
Enter message server = "Helo server"

Reply = "Hello client"

Server

client: hi

Server: hii



Result

Thus the implementation of chat client server using TCP was successfully and verified.

21/10/24

Pratical - 13

AIM : Implement your own ping program

ALGORITHM :

- Open a raw socket to send ICMP req
- Create the ICMP Echo payment request packet including a header and data
- Send packet send the ICMP request to target host
- calculate the time
- Show response

CODE

```
import os
import socket
import struct
import time
import select

ICMP_Echo_request = 8
ICMP_Echo_Reply = 0

def checksum(data)
    sum = 0
    countto = (len(data)/2)*2
    count = 0
    while count < count_to:
    this_val = data[count+1] 255 + data[count]
    sum = sum + this_val

def create_packet(id)
header = struct.pack('bbHHh', ICMP_Echo_request, 0, id, 1)
```

```
checksum_val = checksum(header)
header = struct.pack(ICMP_echo_response, 0, ch
def ping(host):
try
ICMP_socket = socket.socket(socket.AF_IN
dest_addr = socket.gethostbyname(host)
packet_id = os.getpid()
packet = create_packet(packet(dest_addr,
start_time = time.time()
if ready[0]
    rev_packet, addr = icmp_socket.recv(1024, endian
if __name__ == "__main__":
    host = input("Enter host to ping:")
    ping(host)
```

Output

Enter host to ping = www.google.com

Reply: time = 21.45ms

Result

Thus, the ping program has been executed successfully.

4/9/11

25/10/24 Practical - 14

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

Algorithm

1. Create a raw socket to capture packets, specifying the protocol
2. Use the socket to continuously receive data packets.
3. For each packet, decode and analyze the IP and transport layer headers
4. Extract relevant information such as source and destination IP addresses, protocol and port number, then print or store data.

Code

```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP
from scapy.config import conf
conf.L3socket = conf.L3socket6

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
```

```
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}")
printf(f"Source IP: {src_ip}")
Print(f"Destination IP: {dst_ip}")
print("-" * 50)

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

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

Output

Protocol: TCP

Source IP : 140.82.112.25

Destination IP: 192.168.1.1

Protocol : UDP

Source IP : 140.82.112.25

Destination IP: 224.0.0.251

RESULT

Thus, packet sniffing is implemented and executed successfully

[-h] [-b] target_name

11/12/19

Pratical - 15

Aim: To analyze the different types of web logs using webalizer tool

Procedure

Step1: Run webalizer windows version

Step2: Input web log file

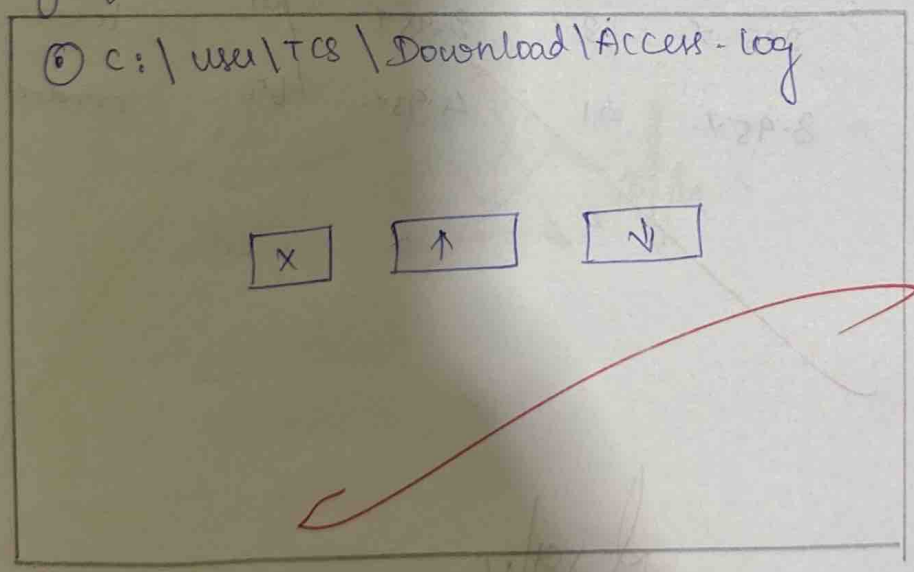
Step3: Press run webalizer

Input

Choose logfiles view setting additional HTML code hide graph ignore/include graph config file.

Input:

log files

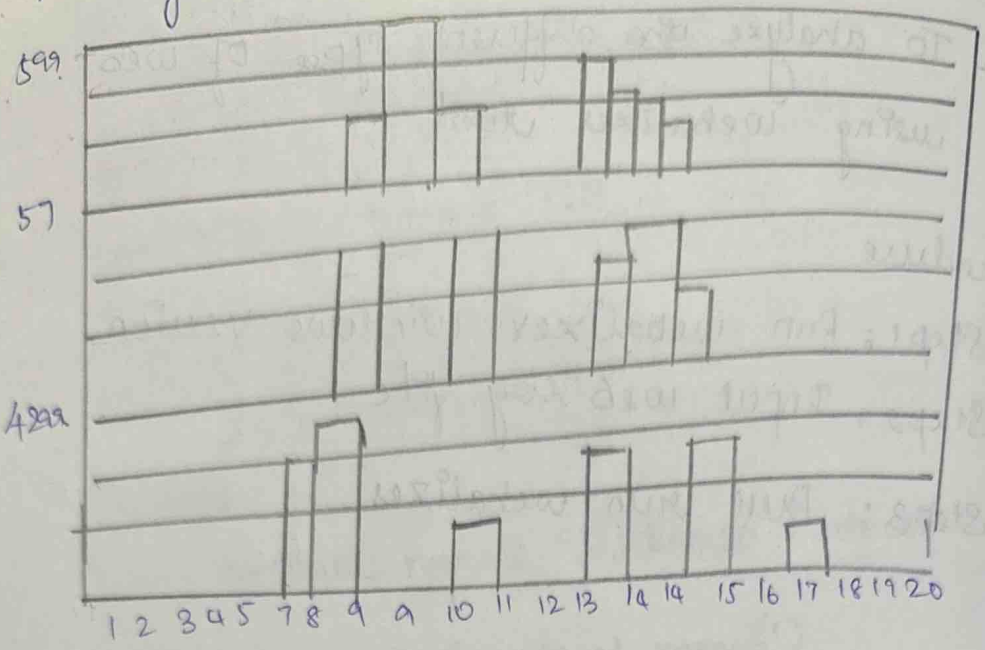


Target directory

C:\use\TCS1

clear existing directory

Monthly Statistics.



Ip address

1.	100	6.047 %	88	7.047 %	62	Showtable.net
2.	72	4.66 %	71	5.66 %	51	Search.net
3.	47	3.04 %	27	3.84 %	52	outlook.com
4.	44	2.95 %	81	3.95 %	91	hexamut.com
5.	35	2.05 %	63	2.95 %	87	bc.ca
6.	29	3.95 %	41	4.93 %	69	pandc.com

19/11

Result: Thus the different types of weblogs are successfully analyzed with Webalizer tools and output is verified.


completed
19/11/