13/8/24

PAGE No.
DATE: / /

Pratical-1

Ex. no. 1 Date:- 13/7

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

Basic Networking Commands:

1) **arp -a:**

Interface: 172.16.75.45 --- 0xf

Internet Address	Physical address	Type
172.16.72.1	7c-5a-1c-cf-6e-41	dynamic
172.16.72.133	4c-ae-a3-65-97-F3	dynamic
172.16.79.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	Static
239.255.255.251	01-00-5e-7f-ff-fb	Static

2) **hostname:**

C:\USERS\lenovo > hostname

DESKTOP-C01BH7D

3) **ipconfig/all:**

Wireless LAN adapter Wi-Fi 3:

Connection-specific DNS suffix:
Link-local IPv6 address . . . . . : fe80::a890:cbeb:9dbb:d229%15
IPv4 address . . . . . : 172.16.75.45
Subnet Mask . . . . . : 255.255.248.0
Default Gateway . . . . . : 172.16.72.1

PAGE No.

DATE : / /

nbstat -a

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49672	Laptop-b:49673	Established
TCP	127.0.0.1:49673	Laptop-b:49672	Established
TCP	127.0.0.1:49674		
TCP	127.0.0.1:49677		

netstat

tcp 0 0 localhost.localdo:47074 maa05s23.1e100.net:https ESTABLISHED

Proto	Refcnt	Flags	Type	State	I-node	Path
unix	3	[ ]	DGRAM		3394	/run/systemd/notify
unix	3	[ ]	STREAM	CONNECTED	10243	

Proto destination source State PSH DUD SUD IMN

Proto destination source Channel

nslookup

> www.google.com

Server : 172.16.8.1

Address : 142.250.193.164

Non-authoritative answer :-

Name : www.google.com

address :- 142.250.193.164

Name : www.google.com

address : 2404:6800:4007:821::2004

> :

PAGE No.
DATE : / /

Route:-

destination	gateway	Genmask	Flags	Metric	Ref	Use
default	gateway	0.0.0.0	UG	100	0	0
172.16.8.0	0.0.0.0	255.255.252.0	U	100	0	0

Ping:-

usage : ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface] [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] [hop1 ...] destination

usage : Ping -6 [-aAbBdDfhLnOqRUvV64] [-c count] [-i interval] [-I interface] [-l preload] [-m mark] [-M pmtudisc_option] [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] destination.

PAGE No.

DATE: / /

Linux networking Commands

1. Ip

(a) #IP address show

1: lo: <LOOPBACK, UP, LOWER_UP> mtu 65536 qdisc noqueue state unknown group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever

2: enp2s0: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000 link/ether 50:9a:4c:35:11:4d brd ff:ff:ff:ff:ff:ff
inet 172.16.8.107/22 brd 172.16.11.255 scope global enp2s0
valid_lft forever preferred_lft forever
inet6 fe80::2d5:2c44:dfad:49ab/64 scope link
valid_lft forever preferred_lft forever

3: wlp3s0: <NO-CARRIER, BROADCAST, MULTICAST, UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
link/ether 92:99:a9:16:5d:04 brd ff:ff:ff:ff:ff:ff

PAGE No.
DATE : / /

2. Iplowfig

ens 160: flags = 4163<UP, BROADCAST, RUNNING, MULTICAST>
mtr 1500
inet 192.168.22.128 netmask 255.255.255.0
broadcast 192.168.22.255

3. MTR

mtr <options> hostname · IP

a. [root@server] # mtr google.com
keys: Help pmplay mode restart statistics order of fields

	Host	Packets Loss %	Snt	Last	Pings Avg	Best	Wrst	StDev
1.	gateway	0.0%	163	1.2	1.0	0.3	6.1	0.6

b. [root@server] # mtr -q google.com
-F .. Filename FILE read hostname(s) from a file
-4 use IPv4 only
-6 use IPv6 only
-4, --nap use UDP instead of ICMP echo

c. [root@server] # mtr -b google.com
keys: Help Display mode Restart statistics Order of fields quit

	Host	Packets Loss %	Snt	Last	Pings Avg	Best	Wrst	StDev
1.	gateway (192.168.22.2)	0.0%	391	1.1	1.1	0.2	10.2	0.8

d. [root@server ~] # mtr -c 10 google.com
Keys: Help Display mode Restart statistics Order of fields quit

	Host	Packets Loss %	Snt	Last	Pings Avg	Best	Wrst	StDev
1.	gateway	0.0%	4	0.1	7	0.1	1.3	0.3

PAGE No.
DATE : / /

4. tcpdump

[root@server] # dnf install -y tcpdump

To install -y tcpdump

[root@server ~] # tcpdump -D

1. Ens 40 [Up, Running, Connected]
2. any (Linux - device that captures on all interfaces) [up, running]

[root@server ~] # tcpdump : ethio
dropped : not shown output suppressed, use -v[v] for full protocol decode

[root@server ~] # tcpdump -i ethio -c 10

10 packets captured
20 packets received by filter
0 packets dropped by kernel

[root@Server ~] # tcpdump -i ethio -c 10 host 8.8.8.8
dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]

[root@Server ~] # tcpdump -i ethio src host 8.8.8.8

dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v]
for full protocol decode listening to 10

PAGE NO.
DATE : / /

[root@server ~]# tcpdump -i ethx host x.x.x.x

dropped priv. to tcpdump
tcpdump: verbose output suppressed, use -v[v] for full protocol decode on lo,

[root@server ~]# tcpdump -i ethx net 10.0.0.0 mask 255.255.255.0

To capture traffic and to form a specific network using this command.

(or)

[root@server ~]# tcpdump -i ethx net 10.0.0.0/24

~~to cap~~

Capture traffic to and form port numbers

Capture dns port 53 traffic:

[root@server ~]# tcpdump -i eth0 port 53

For specific host,

[root@server ~]# tcpdump -i eth0 -c 10 host www.google.com and port 443

To capture only HTTPS traffic,

[root@server ~]# tcpdump -i eth0 port 13

to capture all port except port 80 and 21,

[root@server ~]# tcpdump -i eth0 port not 53 and not 25

PAGE No.

DATE : / /

Nemli

1. # nemli connection show

Name	UUID	type	device
Wired Connection 1	a5eb6490-cc20-3b5f-8bf8-0314a27f8f75	Ethernet	ens20

2. # nemli connection add con-name wired connection 1 ifname ens20 type ethernet

3. # nemli connection modify 'Wired Connection 2"
→ Here, "wired connection 1" is the name of the connection

4. Display the current settings of the connection profile:
nemli connection show

Connection.interface-name: enp2s0
Connection.autoconnect: yes
IPV4 method: auto
IPV6 method: auto

Result:
Thus the study of various network Commands used in Linux and windows is done and executed successfully.

PAGE No.
DATE : / /

5. To set a static IPv4 address, network mask, default gateway, DNS servers, and search domain, enter:

nmcli connection modify "wired connection 1" ipv4.method manual ipv4.addresses 192.0.2.1/24 IPV4.gateway 192.0.2.254 ipv4.dns 192.0.2.200 IP4.dns-search www.google.com.

7. # nmcli connection up "wired-LAN"

→ Activating the Profile.

2014

Result :-

Thus the study of various network commands used in Linux and windows done executed successfully.

PAGE No.
DATE: / /

Practical - 2

Ex no: 2
Date: 24/7

Aim:-
Study of different types of network cables.

(a) understand different types of network cable:
Different type of cables used in networking are:
1. unshielded Twisted Pair (UTP) Cable
2. Shielded twisted Pair (STP) Cable
3. Coaxial Cable
4. Fibre Optic Cable

Cable type	Category	Maximum Data Transmission	Advantages/ disadvantages	Appl/ use	Image
UTP	Category 3 Category 5	10 bps/ up to 100 mbps	Advantages: • cheaper in cost • Easy to install as they have a small overall diameter. Disadvantages: • more prone to EM? Electro magnetic interference and noise	10 Base TBEthernet Fast Ethernet Gigabit Ethernet	
	Category 5e	1 Gbps			
	Category 6, 6a	• 10 Gbps	Advantages:- • Shielded • Faster than • UTP to Less susceptible to noise and interference	Gigabit Ethernet 10G (STm)	

PAGE No.
DATE: / /

			disadvantages:	
SSTP	Category 7	10Gbps	. Expensive . Greater installation effort	gigabit Ethernet, 10G Ethernet (100m)
Coaxial Cable	54-6 54-9 54-11	10-100 Mbps	advantages: . High bandwidth . Immune to interference . Low loss bandwidth . Versatile disadvantages: . Limited distance . cost . Size is bulky	Speed of signal is 500m television network High speed internet Connections
Fibre Optics Cable	Single mode Multimode	100Gbps	Advantages . High speed . High bandwidth . High security . long distance Disadvantages: . Expensive . Requires skilled installers	Maximum distance of fibre optics Cable is around 100m

PAGE No.

DATE : / /

b) Make your own Ethernet (cross-over cable) Straight cable.

- Tools and parts needed.

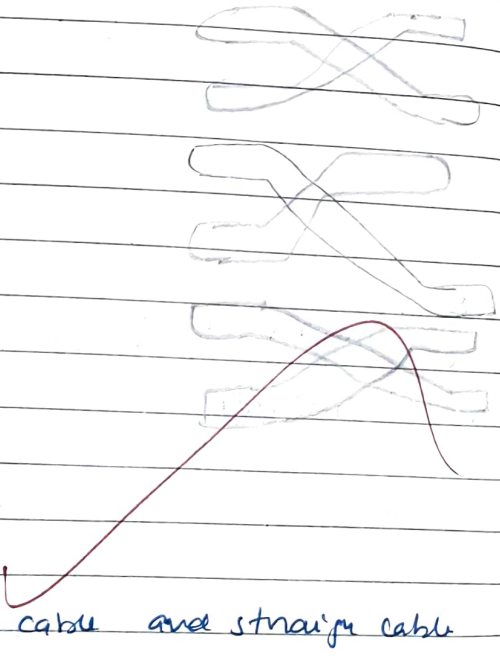
Ethernet cabling. CAT 5C is certified for gigabit support, but CAT 5 cabling works as well, just over shorter distances.

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

- Two RJ45 plugs
- Optional two plug shields.

Straight through cable

X-over cable



Difference between crossover cable and straight cable

Straight cable:

Straight through network cable: both ends should be a cross over cable; one side B, one side B

PAGE No.
DATE : / /

Step 1:- To start the Construction of the device, begin by threading shrink onto the cable.

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

Step 3:- After, you will need to untangle the wires; there should be four twisted pairs. Referring back to the sheet, arrange them from top to bottom one and should them from top to bottom. One arrangement A and other in B.

Step 4:- Once the order is correct, bunch them together in a wire, and if there are any that stick out further than others, snip them back to create an even level. The different part is placing them 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 towards you, as shown.

Step 5:- Now push the cable right in 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; simply snip the cable back A little more.

Step 6:-

Step 7:- Lastly, repeat for the other end using diagram(A) and (B)

Result:-
Thus the different Network cables were

PAGE No.
DATE : / /

Exno:- 3
Date:- 6/8/24

Aim:- To study the packet tracer tool Installation and User Interface Overview

a) 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 model complex systems without the need for dedicated equipment.
2. It helps you to practice your network configuration and troubleshooting skills via computer or an android or ios based mobile device.
3. It is available for both the Linux and windows desktop environments.
4. Protocols in packet tracer are coded to work and behave in the same way as they would on real hardware.

PAGE No.

DATE: / /

Installing Packet Tracer

To download Packet tracer, visit netacad.com and log in with your Cisco Networking Academy credentials. Once logged in, click on the packet tracer graphic and download the version that matches your operating system.

For Windows:

- Download the setup file named "PacketTracer_setup6.0.1.exe".
- Run the file, accept the license agreement, choose an installation location, and complete the setup.

For Linux (Ubuntu/Debian):

- Download the appropriate file for your distribution

Fedora/Redhat/CentOS must download should the file for ubuntu. Grant executable programs permission to the file by using chmod, and execute it to began the installation.

Chmod +x PacketTracer601_i386-installer-rpm.bin
./PacketTracer601_i386-installer-rpm.bin

PAGE NO.

DATE

User Interface Overview:-

The layout of packet tracer is divided into various components. The components of the packet tracer interface are as follows: mention the numbers with explanation.

1. Menu bar - This is a common menu found in all software application. It is used to open, save, print, change preferences, and so on.
2. Main toolbar - This bar provides shortcut icons to menu options that are commonly accessed such as open, save, zoom, undo and redo, and on the right hand side is an icon for entering network information for the current network.
3. Logical / physical workspace tabs - These tabs allow you to toggle between the logical and physical work areas.
4. Common Work space - This is the area where topologies are created and simulation are displayed.
5. Common tools bar - This toolbar provides controls for manipulating topologies, such as select, move layout, place note, delete, inspect, resize, change, and add simple / complex PDU.
6. Real Time / Simulation Tabs - These tabs are used to toggle between the real and simulation modes. Buttons are also provided to control the time, and to capture the packets.
7. Network Component box - This component contains all the network and end devices available for the packet tracer and is further divided into two areas: area for device type selection box - This area contains device categories, device type selection contains devices category is selected.

PAGE No.
DATE: / /

, this section box displays the different device models within category.

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

d) Analyze the behavior 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, indicating that the link is up, similarly connect remaining 3 PCs to the HUB.
   (c) Similarly connect 4 PCs to the switch using copper straight through cable.

3. Click on the PCs connected to hub, go to the desktop tab, click on IP configuration and Enter an IP address and Subnet mask. Leave the default gateway and DNS server information is not needed as there are only two end devices in the network.

4. Observe the flow of PDU from source pc to destination pc by selecting the Realtime mode

PAGE No.
DATE: / /

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

(a)

Switch

- packet forwarding.
- learning process.
- Broadcasting.

Hub

20/11

RESULT:-

Thus the program for Installing of the cisco packet tracer in your window was successfully installed.

PAGE No.
DATE : / /

Exno: 4
Date : 10/8/24

Aim:- Set up and configure a LAN (local area Network) using a switch and Ethernet cable in your lab.

What is LAN?

→ A local area network connect devices within a limited area, like an office, school or home, enabling resource sharing such as data, printer, and internet access. A LAN switch is the central device that manages and directs communication within the network, allowing connected devices to communicate directly for fast and secure data transfer.

How to setup a LAN

Step 1:- plan and design an appropriate network topology taking into account network requirement and equipment location

Step 2:- You can take 4 computers, a switch with 8, 16 or 24 ports which is sufficient for network of three users and 4 Ethernet cable.

Step 3:- Connect your computers to network switch via an Ethernet cable, which is as simple as plugging one end of the Ethernet cable into your computer and the other end into your network switch.

Step 4:- Assign IP address to your PCs

PAGE NO.
DATE / /

1. Log on to the client computer as Administrator or as Owner.

2. Click Network and Internet Connection.

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 IP address.

Step 4:- Similarly assign IP address to all the PCs connected to switch.

PC1 - IP add 10.1.1.1 Subnet mask 255.0.0.0

PC2 - IP add 10.1.1.2 SM 255.0.0.0

PC3 - IP add 10.1.1.3 SM 255.0.0.0

PC4 - IP add 10.1.1.4 SM 255.0.0.0

Step 5 : Configure a switch

1- connect your ~~connection~~ computer to the switch : To access the switch's web interface.

2. Log in to the web interface

3. Configure the basic settings

4. Assign IP address as 10.1.1.5 and Subnet mask 255.0.0.0.

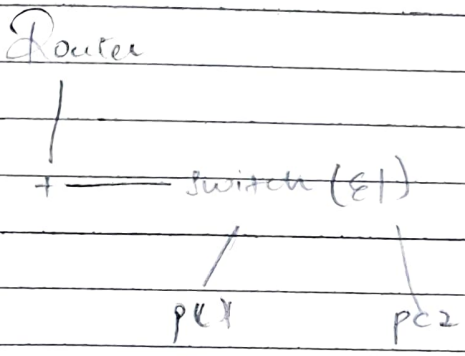
Step 6:- Check the connectivity between switch and other machine by using ping command in the command prompt of the device.

PAGE NO.
DATE : / /

Step 7 :- Select a folder → go to properties → click Sharing tab → Share it, with everyone in the same LAN.

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

Student observation :-



IP configuration

PC1 : IP : 10.1.1.1 Subnet name : 255.0.0.0

PC2 : IP 10.1.1.2 Subnet : 255.0.0.0

Result :-

The LAN setup was configured.

Page No.
Date: / /

Experiment 5 Date: 13/8/24

Aim:

Experiments on packet capture tool: wireshark.

Packet sniffer:

i) Easily strategize, setup and measure a given by Computer.

ii) Store and display the contents of the various protocol fields in messages.

- Need saves packets only
- No packets addressed to it
- Receives a copy of all packets

Packets sniffer structure diagnostic tools.

i) Tcpdump (eg: tcpdump) - enrhust 10.129.41.2 → web)

ii) Wireshark (wireshark → enrhust)

Wireshark

Wireshark is a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format.

→ What we can do - Capture network traffic
→ Decode packets protocol using dissectors
- Analyze problems

→ Used for - People: Kavi, network protocol internals networks, administrators: trouble shoot network problems.

Getting wireshark - Wireshark can be downloaded to windows or macOS from its official website.

Capturing Packets: After downloading and installing wireshark, launch it and double-click the name of a network interface under Capture to start capturing packets.

The "Packet List" Pane: display all packets in current capture the packet number, time, source, destination, protocol and more details.

The "Packet Bytes" pane: shows data of current packet: color coding, domain captures, filtering packets, inspecting packets, flow graph, giving a better understanding of what will be.

Capturing and analyzing packets using wireshark tool.

Procedure:

Select local area connection in wireshark
Go to capture → option.
Select "Stop capture" automatically after 100 reputable then click start capture
Save the packets.

1. Create a filter to display only TCP/UDP packets and provide the follow graph.
2. Create a filter to display only ARP packets and export the packets.
3. Create a filter to display only DNS packets and provide the follow graph.

PAGE No.

DATE : / /

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

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

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

Result:

The Experiment on wireshark is executed successfully.

PAGE No.
DATE : / /

Student observation :

1. Promiscuous mode is a configuration network interface card (NIC) or network device that allows it to capture and process all the network segment to which its allows rather than only packets addressed to it.

2. No, ARP packets do not have a transport layer header. ARP is a protocol used to map IP addresses to physical MAC addresses on a local network. It's part of the network & OSI model.

3. DNS (domain Name System) uses both UDP and TCP as its transport layer protocol but it also uses TCP for larger responses.

4. The default port number used by HTTP protocol is 80. for HTTPS, the secure version of HTTP the default port is 443

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

PAGE No.
DATE: / /

Experiment 6 Date:- 16/9/24

Hamming code

Aim:- Write a program to implement error detection using hamming code concept. Make a test run to input data stream and verify error detection feature.

Error Correction at data link layer:

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

Create Sender program with below features:

1. Input to sender file should be a text of any length. Program should convert the text to binary.
2. Apply hamming code concept on the binary data and add redundant bits to it.
3. Save this output in a file called channel.

Create receiver program with below features:

1. Receiver program should read the input from channel file.
2. Apply hamming code on binary data - check the error.

PAGE No.

DATE : / /

3. If there is a error, display position of user.

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

Student observation:

Code:

```
import numpy as np.
def text_to_binary (text):
    return ''.join format(ord(char), '08b' for
    char in text)

def binary_to_text (binary):
    chars = [binary[i:i+8] for 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: r<m+r+1);
        r+=1;
        return r.

def pos_redundant_bits (data, r)
        j:=0
        k:=0
        m:= len(data)
        res = ''.
```

PAGE No.
DATE : / /

```
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):                    # calculating parity bits
    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
```

PAGE NO.
DATE : / /

```
if res != 0:
    print(f"Error detected at position {data
    = len(data)
    if res == n:
        data[res-1] = '0' if data[res-1]
                    else '1'.
        print(f"Error port corrected at
                    Position")
    else:
        print("Error position out of
        range for correction performed")

Corrected_data = ''.join(data)
return corrected_data
else:
    print("No error detected").
    return data.

def remove_redundant_bits(data, r)
        j=0
        Original_data=''
    for i 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 position<1 or position>len(data):
    print("Error position is out of
        range")
    return data.
```

PAGE No.
DATE / /

```
    data = list(data)
    data[position-1] = '0' if data[position-1] == '1' else '1'
    print("Introduced error position:", position)
    return ''.join(data)

# Sender program
def sender(text):
    binary_data = text_to_binary(text)
    m = len(binary_data)
    r = calc_redundant_bits(binary_data, r)
    arr = calc_parity_bits(arr, r)
    print("Sender output (binary with
          redundant bits):", ''.join(arr))
    return arr

# receiver program
def receiver(data):
    n = calc_redundant_bits(len(data))
    corrected_data = detect_correct(data, r)
    original_data = remove_redundant_bits(corrected_data)
    ascii_output = binary_to_text(original_data)
    print("Decoded text:", ascii_output)

if __name__ == "__main__":
    input_text = input("Enter text to
        be encoded:")
    channel_data = sender(input_text)
    corrupted_data = introduce_bit()
    receiver(corrupted_data)
```

PAGE No.
DATE

P:

Enter the text to be encoded: hello
Sender O/P (Binary with redundant bits)
1000100111000011100101010111000111
Introduced error at position: 2
Error corrected at position: 2
Decoded text: hello
=== Code Execution Successful ===

20/11

Result:

Thus the program for finding the error using the hamming code was successfully executed.

PAGE No.
DATE: / /

Experiment - 7 Date:- 15/9/24

Sliding Window Protocol

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.

Code:-

```
import time
import random

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

def send_frames(frames, window_size, base):
    print("\n--- Sending frames ---")
    for i in range(window_size):
        if base + i < len(frames) and not frames[base + i].acknowledged:
            print(f"Sent Frame {frames[base + i].frame_no}: {frames[base + i].data}")
    print("Frames sent, waiting for acknowledgements...\n")

def receive_frames(frames, window_size, base):
    print("\n--- Receiving frames ---")
```

PAGE No.
DATE : / /

```
for i in range (window_size):
    if base+i < len(frames) and
    not frames[base+i].acknowledged:
        if random.random() < 0.2:
            print("Received Frame {frames
base+i].frame_no}: {frames[base+i].data}
{Error}")
            frames[base+i].acknowledged = False
        else:
            print("Received Frame {frame[ba
frame_no}: {frame[base+i].data}
[OK]")
            frames[base+i].acknowledged = True

def sliding_window_protocol():
    window_size = int(input("Enter wind
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, window_size, base
    time.sleep(2)

    receive_frames(frames, window_size,
                   base)

while base < len(frames) and frames[base
```

PAGE No.
DATE : / /

```
        acknowledged :
            base += 1

        if base < len(frames):
                print("\n Representing Unacknowledged
        frames ---- \n")
            time.sleep(2)

        print("\nAll frames sent and acknowledged!")


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

Output:

```
Enter window size : 2
Enter a message to send : abi

---- Sending Frames ----
Sent frame 0 : a
Sent frame 1 : b

Frames sent, waiting for acknowledgements ...

---- Receiving Frames ----
Received Frame 0 : a [Error]
Received Frame 1 : b [OK]
```

Result :-

Thus the program for sliding window protocol using python was successfully executed.

PAGE No.
DATE : / /

Experiment - 8 Date:- 20/9/24

Configuration of wireless LAN using cisco packet tracer

Aim:-

Simulate virtual LAN configuration using CISCO packet tracer simulation

Objective:-

Part 1: Build the network and configure Basic Device settings.

Part 2: Create VLANs and ~~design~~ switch Ports

Part 3: Maintain VLAN port assignments and the VLAN Database

Part 4: Configure an 802.1Q trunk between switches

Instructions

Part 1: Build the network and configure basic device settings

Steps

Step 1: Build the Network as shown in the Topology

(a) Click and Drag both switches S1 and S2 to the Rack

(b) Click and drag Both PCA and PC-B to the table and use the power button to turn other.

(c) Provide network connectivity.

(d) Connect Console cable from device

PAGE No.
DATE: / /

Step 2: Configure basic settings for each switch
(a) Enter configuration mode
(b) Enter the correct line on each PC
(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 vty
(g) encrypt the plain text Passwords
(h) Create a banner that warns anyone accessing the device
(i) Configure the IP address listed in the address table for VLAN 1 on the switch
(j) Shut down all interfaces that will not be used.
(k) Set the clock on each switch
(l) Close configuration window.

Step 3: Configure PC hosts

Step 4: Test connectivity

Part 2: Create VLANs and assign switch ports

Step 1: Create VLANs on the switches
(a) Create VLANs on S1
(b) Create the VLANs on S2
(c) Issue the show vlan brief command on S1 to view the list of VLANs on S1

Step 2: Assign VLANs to the correct switch interface
(a) Assign VLANs to the interface on S1.

PAGE NO.
DATE: / /

1) Assign VLANs to the interfaces
2) from VLAN 1 to remove the management IP address and configure it on VLAN 30 ?

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

(6) Issue the show VLAN brief command and verify that the VLAN are assigned to the correct interfaces

(7) assign PC-B to the operations VLAN on S2.

(8) from VLAN 1, remove the management IP address and configure it on VLAN

(9) use the show VLAN assigned to the correct interfaces.

Part 3: Maintain VLAN port Assignments VLAN Database

Step 1 : Assign a VLAN to multiple interfaces to continue configuring both network switches.

Step 2: Remove configuring both network switches a. add VLAN 30 to interface Fo/24 without issuing the global VLAN commands.

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

PAGE No
DATE / /

data base.

Part 4: Configure an 802.1Q trunk between the switches

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

Step 2: manually configure trunk interface F0/1

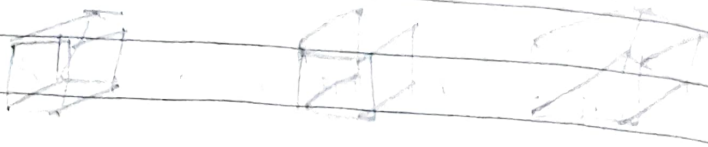
Result:-

Thus virtual LAN configuration using Cisco packet tracer is executed successfully.

PAGE No.
DATE: / /

Aim:-

(5) Configuration of wireless LAN using cisco packet tracer.



PC-PT PC-PT PC-PT

PC0 PC1 PC2

To complete this these tasks follow these steps by step instructions

1. Click on wirless router
- Select Administrator tab from top menu click on save settings.
- Next click on wireless tab and set default SSID 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 PC's from wireless router
- Click on connect button to connect mother network
- It will ask for WEP key insert 0123456789 and click connect

· If will connect you with wireless router and you can use internet.

· Repeat same process on pc1 and pc2

Student observation.

Student observation.

c) what is SSID of a wireless router

· SSID is the name of a wireless network. It is used to identify and differentiate one network from another. When you connect to a wi-fi network, you typically see a list of available SSID.

2014

Result:-

Thus configuration of wireless LAN using cisco packet tracer is executed successfully.

PAGE NO.
DATE: / /

Ex. 9

Date: 24/9/24

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 for each IP class.

Creating a Network Topology:

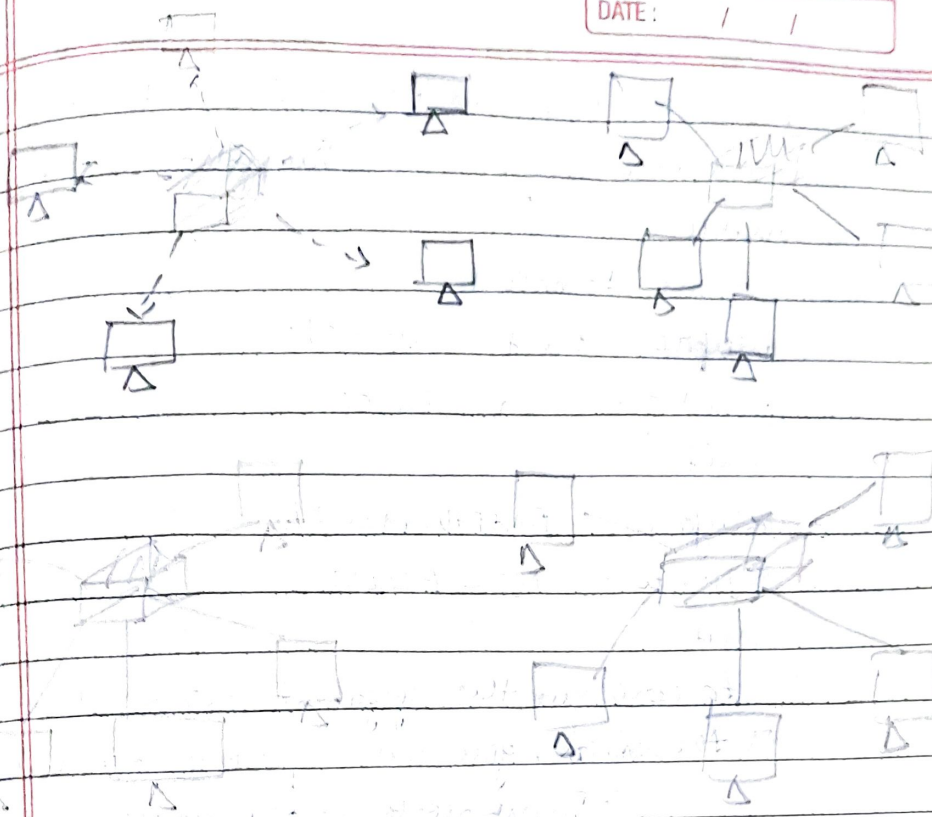
The first step in implementing classless IP subnetting is to create a network topology in Packet Tracer

Adding the devices.

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

Subnetting:-

To subnet the network address of 192.168.10.0/ to provide enough space for at least 5 addresses for end devices, the switch and the router we can use a /27 Subnet mask. This will give us 8 subnets with 30 host address each.



Configuring the devices:-

Now that we have added our devices and connected them, we can start configuring them. This will open the command line interface for the router. On the CLI, enter the following

- enable
- configure terminal
- ip address {IP address} {Subnet Mask}
- no shutdown
- exit
- interface FastEthernet 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

PAGE NO.
DATE: / /

click on the switch and select "CLI"
. in the CLI, enter the following commands
enable
Configure terminal
interface FastEthernet 0/1
Switchport mode access
exit
interface FastEthernet 0/2
Switchport mode Access
exit.

to configure the gigabit ethernet interface on the router, you can follow these steps

1. Right click on the router and select CLI
2. Enter the following commands.

enable
Configure terminal
interface gigabit Ethernet 0/0
ip address <ip address> <subnet mask>
no shutdown
exit

testing the network:

open a cmd 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 router and the pc.

Advantages of subnetting:

(a) Large networks into smaller more manageable networks can operate independently, while still being part of the large network. It can identify the default subnet mask to allocate IP address to different organizations and efficiency

(b) Improved network management

Breaking down a large network into subnets make it easier to manage.

Enhanced security

Subnets can select vulnerable areas, to assess certain address of the network.

A 20 Mk

Result:

Thus the program was successfully done.

practical - 10

Date :- 28/9/24

Aim:-

(a) Inter networking with router in cisco packet tracer simulator.

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

Procedure:-

Step 1 : (Configuring Router 1)

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

Step 2 :- (Configuring PCs)

1. Assign IP addresses to every PC in the network
2. Select the PC, GO TO the desktop and select IP configuration, subnet mask.
3. Assign the default gateway of PC0 as 192.168.10.1
4. Assign the default gateway of PC1 as 192.168.20.

Step 3 : (Configuring PCs with router)

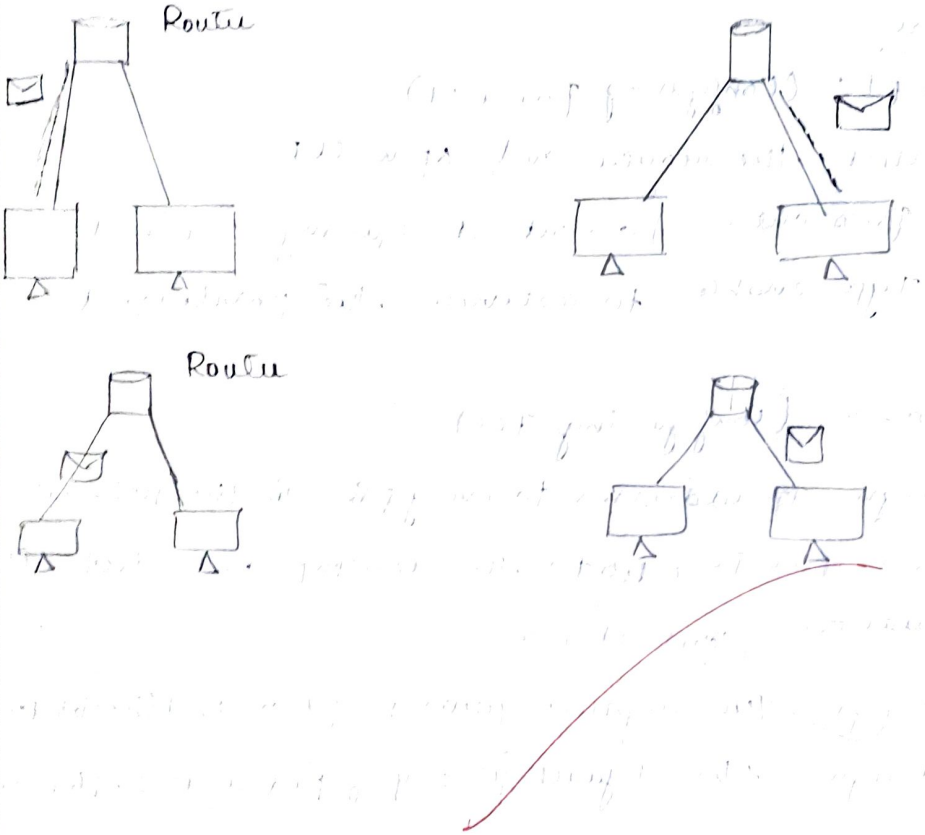
1. Fast Ethernet 0/0 port of Router 1
2. Fast Ethernet 0/1 port of Router 1

Router configuration tasks

Device Name	IP address	Subnet mask	Ip address FastEthernet 0/0	Subnet mask
Router 1	192.168.10.1	255.255.255.0	192.168.20.1	255.255.255.0

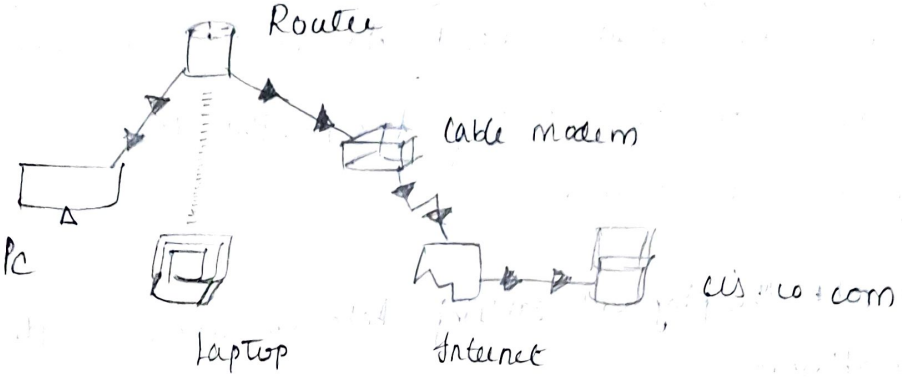
PC Configuration Table

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



Aim:-

b) Design and implementation an internetwork using wireless router, DHCP server and internet cloud.



Addressing Table:-

Device	Interface	IP address	Subnet mask	Default Gateway
PC	Ethernet 0	DHCP		192.168.0.1
Wireless Router	LAN	192.168.0.1	255.255.255.0	
Wireless Router	Internet	DHCP		
cis.co.com Server	Ethernet 0	208.67.220.2	255.255.255.0	
Laptop	Wireless	DHCP		

Objectives:-

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

Step 1:- Launch Packet Tracer

Step 2:- Build the topology

a. Add network devices to the workspace

To place a device on to 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 names 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 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 1 : configure the network devices

Step 1 : configure the wireless router

a. Create the wireless network on the wireless router.

b. Click on the save setting tab

Step 2 : configure the laptop

a. configure the laptop to access the wireless network

Step 3 : configure the PC

a. configure the PC for the wired network.

Step 4 : configure the Internet cloud.

a. Install network modules if necessary

b. Identify the form and TO ports

c. Identify the type of provider.

Step 5 : configure the cisco.com server

a. configure the cisco.com server 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 setting

d. Configure the cisco.com server global setting

part-3 : verify connectivity

Step1: Refresh the IPv4 setting on the PC.

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

b) test connectivity to the cisco.com server from the PC

Student observation

1. SSID Configuration: set up a unique network name (SSID) for your wireless network to allow device 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 either networks or devices.

- Frequency Band: Select the 2.4 GHz or 5 GHz band, depending on device compatible and coverage requirements.

DHCP server configuration:

- IP address range: define the IP address range that the DHCP server will assign to devices

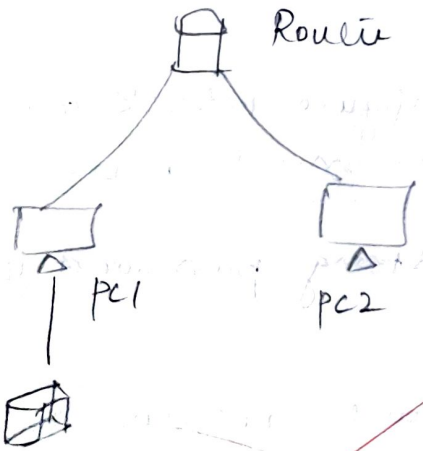
2. Significance of DHCP

- Automatic IP assignment: DHCP dynamically assigns IP addresses, reducing manual configuration and preventing IP conflicts.

Supports scalability: DHCP servers make it easy to add and manage multiple devices across large networks, as essential feature in growing environments

IP address

(i) Configure the DHCP server on the router
(ii) Connect the PCs to the switch
(iii) test connectivity



Result:-

Thus the program for intra networking & subnet routing using cisco packet tracer was successfully done.

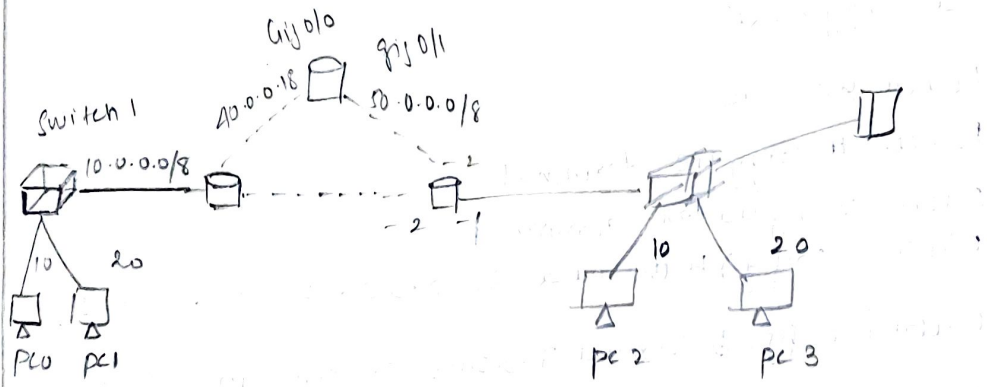
Ex. no : 11

Date :- 11/10/24

Aim :-

(a) Simulate static routing configurations using Cisco packet tracer.

Static Routes are the 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, verify static Routes

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

Router	Available Networks on local Interfaces	Networks available on other router interfaces
Router 0	10.0.0.0/8	30.0.0.0/8
	20.0.0.0/8	50.0.0.0/8
	40.0.0.0/8	
Router 1	20.0.0.0/8	10.0.0.0/8
	30.0.0.0/8	40.0.0.0/8
	50.0.0.0/8	
Router 2	40.0.0.0/8	10.0.0.0/8
	50.0.0.0/8	20.0.0.0/8
		30.0.0.0/8

Router 0 requirements:

- Create 2 routes for network 30.0.0.0/8 and configure the first route as the main route and the second route as a backup route
- Create 2 routes for the host 30.0.0.1/8012 and configure.
- Create 2 route for network 50.0.0.0/8 and configure

Router configuration

Router > enable
Router # configure terminal
Enter configuration command
Router (config) # ip route 30.0.0.0 255.0.0.0 20.0.0.021
Router (config) # ip route 30.0.0.0255.0.0.0 40.0.0.2 20
Router (config) # exit
Router # show ip route static
53.0.0.0/8 [10/0] via 20.0.0.2
53.0.0.0/82 [10/0] via 10.0.0.2
Router #

Router 1 requirements.

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

Router configuration:

Router > enable
Router # configure terminal
Enter configuration command, one per line
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1 10

Router (config) # ip route 10.0.0.0 255.0.0.0 50.0.0.1 20

S 10.0.0.0/8 [10/0] via 20.0.0.1

S 40.0.0.0/8 [10/0] 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 routes.

Router 2 configuration :-

Router > enable

Router # configure terminal

Enter configuration commands

Router (config) # ip route 10.0.0.0 255.0.0.0 40.0.0.0 10

Router (config) # ip route 20.0.0.0 255.0.0.0 50.0.0.0 20

Router (config) # exit

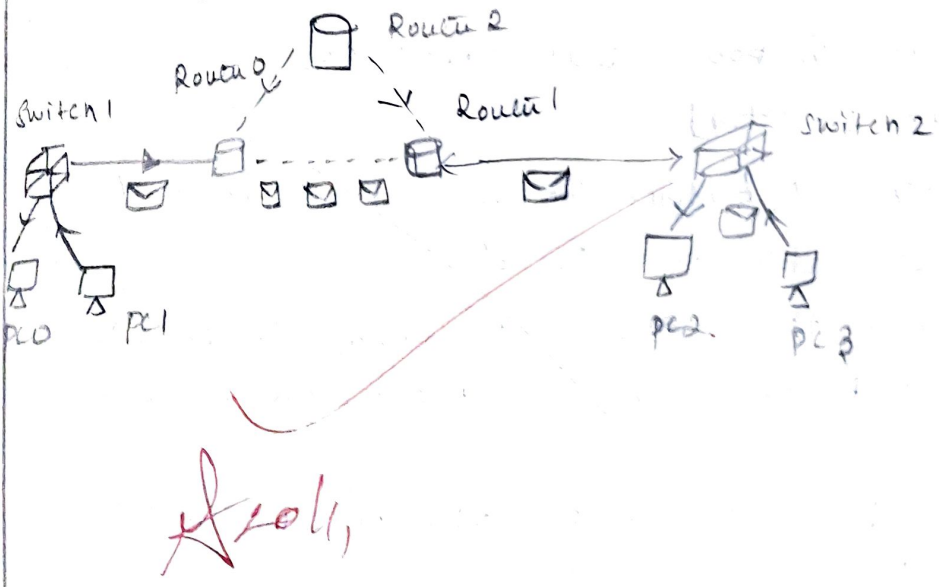
Router # show ip route static

S 10.0.0.0/8 [1/0] via 40.0.0.1

S 30.0.0.0/8 [1/0] via 50.0.0.2

Router #

Verifying static Routing



Result:-

Thus the connection simulate static routing. Configuration using cisco packet tracer is configured.

Aim:-

b) Simulate RIP using CISCO packet tracer.

Assign IP address to pcs

Assign IP address to interfaces of Routers

Configure IP address and other parameters on interfaces before we could actually use them for routing.

Router > enable

Router # configure terminal

Enter configuration commands

Router (config)#

Interface FastEthernet 0/0 command is used to enter in interface mode.

IP address 10.0.0.1 255.0.0.0 commands until assign IP address to interface

No. shutdown command will bring interface up we can use show controller interface up command for previllage mode to check the cable is end.

Router # show controllers serial 0/0/0

Interface serial 0/0.

Hardware is power QUICC MPC 860

[Output omitted]

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

Router # configure terminal Command is used to end up global configuration mode.

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

Router1 :-

Router>enable

Router# configure terminal

Enter configuration command

Router(config)# interface serial 0/0/0.

Router(config-if)# ip address (192.161.1.250.255.255.255.255)

Router(config-if)# no shutdown

Router(config-if)# exit

Router(config)# interface serial 0/0/1

Router(config-if)# ip address 192.168.7.245.255.255.255.255

Router(config-if)# clock rate 64000

Router(config-if)# bandwidth 64

Router(config-if)# no shutdown

Router(config-if)# exit

use same commonand to assign ipaddress to Router2.

Configure RIP Routing Protocol :

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

Router0

Router(config)# router rip

Router(config-router)# network 10.0.0.0

Router(config-router)# network 192.168.1.255

similarly configuring Router2 and Router3 with different ip address like network 192.161.1.2 and more

Result :-

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

Ex. no. 12 Date :- 5/10/24

Aim :-

Implement echo client server using tcp / udp sockets

TCP echo client - server algorithm

Server :

1. Create a tcp socket
2. Connect the socket to a local address and port
3. Listen for incoming client connection.
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 tcp_server() :
        Server . socket = Socket & Socket T: AF_INET ,
        Socket : Stream ,, Stream)

        Server .socket . bind (("localhost"
    12245)
  Server_socket . listen ()
```

```
print ("TCP server is waiting for connection")
    connect, 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.sendall(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 Roshan

Received from server : 'Hi, I am Roshan

> python tcp-server.py

TCP server is waiting for a connection

Connected to ('127.0.0.1', 56893)

Received : Hi, I am Roshan

Result:-

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

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, listen for incoming connections.
2. When a new client connects add client to a list of connected clients start a new process to talk to the client
3. For each connected clients start a new keep checking for new messages.
4. If a client disconnects remove that client from the list and ~~stop~~ talking to that client.
5. Keep running the process till the server stops.

Chat_client :

1. Connect to the server by creating a socket and connect it to a server address and port.
2. Start a process by creating a to listen to messages.
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")
            if message:
                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

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-8")
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.AFINET
    Socket.socket.SOCK_STREAM)
server.bind(('127.0.0.1', 12345))
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 stated on 127.0.0.1 : 12345
New connection from ('127.0.0.1', 57226)
Received from client : Roshen
type from message to client: Received
>python chat_client.py
    Connected to chat server
    You: chill
    You: server: Received
```

Result:-

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

pratical - 13

Date - 8/10/24

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 socket.socket (SOCKET AFINET
                        Socket.SOCK_DGRAM) as 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 ('y')}
        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, AFINET)
                        socket.SOCK_DRAM ) as s:
    try:
        s.bind ((host, port))
        print (f" UDP server running on
                {host}: {port}")
```

```
while true:
    data, addr = s.recvfrom(1024)
    print("UDP server running on {host}:{port}")
    s.sendto(b'pong', addr)

except socket.timeout:
    print("Request timed out")
```

Output:

```
>python my-server.py
UDP server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 8934)

>python ping_client
Received ping-from ('127.0.0.1', 12345) in
0.00 seconds
```

Result :-

Thus, the program to implement the program is executed successfully.

Ex: 14 Date :- 12/10/

Aim :-

Write a code using RAW Sockets to implement packet sniffing.

Algorithm :-

- Create a raw socket
- Continously capture incoming packets in a creation
- 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
    dst_ip = IP_layer.dst
    protocol_name = ""
    if protocol == 1
    protocol_name = "ICMP"
    elif protocol == 6:
        protocol_name = "TCP"
    elif protocol == 17:
        protocol_name = "UDP"
```

```
else:
    protocol_name = "unknown protocol"
print(f"protocol : {protocol_name}")
print(f"Source IP : {src_ip}")
print(f"Destination IP : {dst_ip}")
print("-"*50)

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
---
protocol : TCP
Source IP : 192.168.1.2
Destination IP : 172.217.14.206
---
```

20/4

Result:-

Thus, the code using RAW socket to implement packet sniffing is created successfully.

Ex. 15

Date :- 14/10/24

Aim :-

To analyze the difference types of web logs using Webalizer tool.

Procedure :-

S1. Run webaliser windows version

S2 : Input weblog file (down load from local)

S3 : press run webaliser

Choose Logfiles	Logfile	Views	Settings	Additional HTML code	HTML

Input :

Logfiles :

C:\Users\TCS\downloads\access.log

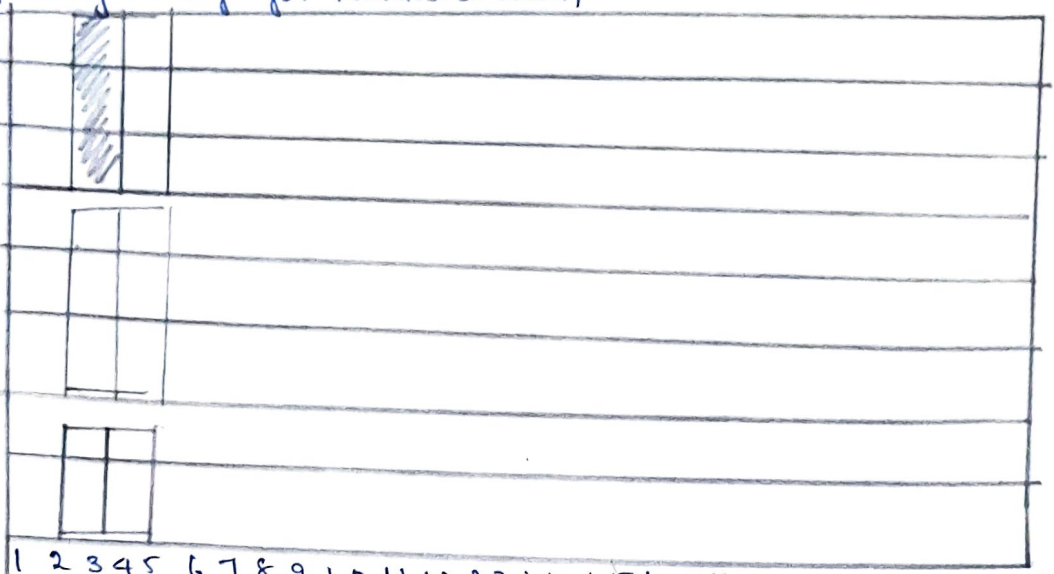
Target directory :

C:\User\TCS\

Clear existing directory

Delete all files in selected targeted directory

Daily usage for November 2024



Day	Hits	Files	Pages	Visits	Sites	Kbytes
3	67.100%	651000	6510000%	1100001.	11000.00%	139.60.00%

Monthly Statistics for October 2024		
Total hits		991
Total files		987
Total pages		971
Total visits		18
Total kbytes		3623
Total unique URLs		1
Total unique referrers		30
Total unique user agents		7
Hits per hour	Avg 2	Max 180
Hits per Day	49	247
pages per day	49	246
Files per Day	48	246
Visits per Day	0	1
Kbytes per Day	0	5
	181	901

2024

Result :-

Thus the procedure to analyse the different types of web logs using website log tool is executed successfully.