INDEX

NAME: Anand.S STD.: CSE SEC.: A ROLL NO.: 026 SUB.: CN - LAB

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	13/7/24	Study of Various Network Commands	9	
2.	16/7/24	Study of Network Cables	10	
3.	27/7/24	Experiments on CISCO PACKET TRACER	10	
4.	17/8/24	LAN using Switches and HUB		
5.	17/8/24	Wireshark		
6.	20/9/24	Hamming Code		
7.	24/9/24	Sliding Window		
8.	4/10/24	LAN config Using CISCO		
9.	8/10/24	Subnetting in CISCO		
10.	15/10/24	Internetworking in CISCO		
11.	18/10/24	Routing at Network layer		
12.	22/10/24	End-End communication at TC		
13.	25/10/24	Ping Program		
14.	29/10/24	Packet Sniffing		
15.	5/11/24	Types of Server - Webalizer Tool		

Date	Title	Page No.	Teacher's Sign / Remarks
4/10/24	VLAN		
4/10/24	WLAN		
15/10/24	Simple Network Router		
15/10/24	DHCP server		
18/10/24	Static Routing		
18/10/24	RIP		
22/10/24	TCP/UDP server		
22/10/24	CHAT Program		

completed

23/11/24

netstat: displays statistics about active TCP/ IP connections, routing
network connections, routing tables and interface statistics.

OUTPUT:

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49674	3ca52znvmj:65001	ESTABLISHED
TCP	127.0.0.1:49675	3ca52znvmj:49670	ESTABLISHED
TCP	127.0.0.1:49676	3ca52znvmj:49675	ESTABLISHED

nslookup: Tool used to perform DNS lookups in Linux, displaying data such as IP addresses, MX records and NS servers of a domain.

OUTPUT:

Server: dns.google.com

Address: 8.8.8.8

Non-authoritative answer:

Name: google.com

Addresses: 2404:6800:4007:81b:200e

142.250.182.78

Pathping: Combines Ping and Tracert, tracing the route to a destination and testing each router along the way to gather data loss statistics.

OUTPUT:

Usage: Pathping [-g host-list] [-h maximum_hops] [-i address] [-n]
[-p period] [-q num-queries] [-w timeout]
[-4] [-6] target_name

Ping: test connectivity between two nodes using ICMP (Internet Control Message Protocol) and can be used with a hostname, IP address or fully qualified domain name

OUTPUT:

Ping statistics for 142.250.182.78:

Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss),

Approximate round trip times in milli-seconds:

Minimum = 34ms, Maximum

Route: shows / manipulates the IP routing table and is used to [illegible] static routes to specific hosts or networks via an interface

OUTPUT:

ROUTE [-f] [-p] [-4] [-6] command [destination]
[MASK netmask] [gateway] [METRIC metric] [IF Interface]

SOME IMPORTANT LINUX COMMANDS

1. ip: Essential for administrators, used to show address information, manipulation routing, and display network devices, interfaces and tunnels

Command Syntax: ip <OPTIONS> <OBJECT> <COMMAND>

a) Show IP addresses assigned to an interface: 'ip address show'

OUTPUT

ens33: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 00:0c:29:e6:03:46 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 192.168.209.130/24 brd 192.168.209.255 scope global dynamic noprefixroute ens33
inet6 fe80::20c:29ff:fee6:346/64 scope link
valid_lft forever preferred_lft forever.

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

c) Delete an IP to an interface: 'ip address del 192.168.209.130/24 dev ens33

d) Bring an interface online: ~~'ip address'~~ 'ip link set ens33 up'

e) Bring an interface offline: 'ip link set ens33 ~~off~~ down'

f) Enable promiscuous mode for an interface: 'ip link set ens33 promisc

~~g) Add a default route via a local gateway: 'ip route add default via 192.168.209.255 dev ens33'~~

g) Display the route taken for a specific IP: 'ip route get 10.10.1.4'

OUTPUT

10.10.1.4 via 192.168.209.2 dev ens33 src 192.168.209.130 uid 0
cache

2. ifconfig: Simple for configuring and troubleshooting, has been replaced by the 'ip' command.

OUTPUT:

ens33: flags = 4419 <UP, BROADCAST, RUNNING, PROMISC, MULTICAST> mtu 1500

inet 192.168.209.130 netmask 255.255.255.0 broadcast 192.168.209.255

inet6 fe80::20c:29ff:fe26:346 prefixlen 64 scopeid 0x20<link>

ether 00:0c:29:26:03:46 txqueuelen 1000 (Ethernet)

RX packets 93541 bytes 141079603 (141.0 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 25871 bytes 1595187 (1.5 MB)

TX errors 0 dropped 0 overruns 0 carriers 0 collisions 0

3. mtr: combines the functionality of ping and traceroute, providing detailed statistics about each hop, including response time and packet loss, to diagnose network issues.

Command Syntax: mtr <options> hostname/IP

a) Show statistics including each hop with time and loss%.

mtr google.com

OUTPUT:

Cucumber-Nova (192.168.209.130) → google.com (142.250.182.78)

2024-07-30 03:46:59

HOST	PACKETS Loss%	Snt	Last	Avg	Best	PING Wrst	StDev
1. _gateway	0.0	261	0.7	0.6	0.3	10.7	1.6
2. maa05s20-in-f14.1e100.net	0.0	294	6.3	7.1	5.2	269.1	17.7

b) Show numeric IP addresses instead of hostnames:

`mtr -n google.com`

c) Show both numeric and IP addresses and hostnames:

`mtr -b google.com`

d) Set the number of pings to send:

`mtr -c 10 gogle.com`

4. tcpdump: designed for capturing and displaying packets.

a) Install 'tcpdump': `sudo apt-get install tcpdump`

OUTPUT

Reading package lists ... Done

Reading dependency tree ... Done

Building dependency tree ... Done

Reading state information ... Done

tcpdump is already the newest version (4.99.4-3ubuntu4.1)

tcpdump set to manually installed

0 upgraded 0 newly installed 0 to remove and 0 not upgraded.

b) List all available interfaces for capturing: `tcpdump -D`

OUTPUT:

1. ens33 [Up, running, Connected]
2. lo [Up, Running, Loopback]

c) Capture traffic on `ens33`: `tcpdump -i ens33`

OUTPUT:

12 packets ~~received~~ captured

12 packets received by filter

0 packets dropped by kernel

5. ping: Verifies IP-level connectivity by sending ICMP Echo messages and displaying Echo Reply messages with Round-trip messages.

OUTPUT

ping google.com

--- google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3006 ms

rtt min/avg/max/mdev = 10.364/30.444/44.225/12.518 ms

CONFIGURING AN ETHERNET CONNECTION USING nmcli

If you connect a host to the network over Ethernet, you can manage the connection's settings on the command line by using the nmcli utility.

PROCEDURE

1. List connection profiles: `nmcli connection show`
2. Add a new Ethernet connection (or skip if modifying existing):
   `nmcli connection add con-name "AAA" ifname ens33 type ethernet`
3. Optionally rename the connection profile:
   `nmcli connection modify 'Wired Connection 1'`
4. Display current settings:
   `nmcli connection show`
5. Configure IPV4 settings:
   `nmcli connection modify 'Wired Connection 1' ipv4.method auto`
6. Configure IPV6 settings:
   `nmcli connection modify 'Wired Connection 1' ipv6.method auto`
7. Activate the Profile:
   `nmcli connection up`

VERIFICATION:

1. Display IP settings:

`ip address show ens33`

2. Display IPV4 default gateway

- `ip route show default`

3. Display IPV6 default gateway

`ip -6 route show default`

4. Display DNS settings:

`cat /etc/resolv.conf`

5. Verify connectivity with 'ping':

`ping google.com`

Student Observation:

1. Which command is used to find the reachability of a host machine from your device? - 'ping'

2. Which command will be give the details of hops taken by a packet to reach its destination? - 'traceroute'

3. Which command displays the ip configuration of your machi
- 'ifconfig'

4. Which command displays the TCP port status in your machi
- 'netstat'

5. Write the modify the ip configuration in a Linux machin
- `nmcli connection modify "Wired connection 1".`

6/8

RESULT:

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

Study of different types of Network Cables

EX.NO: 2 DATE: 16/7/24

AIM: Study of different types of Network Cables

a) Understand different types of network cables:

Different type of cable used in networking are:

1. Unshielded Twisted Pair (UTP) Cable
2. Shielded Twisted Pair (STP) Cable
3. Coaxial Cable
4. Fibre Optic Cable

Cable Type	Category	Maximum Data Transmission	Advantages / DisAdvantages	Application / Use	Image
UTP	Category 3	10 bps	Advantages: - cheaper in cost - Easy to install as they have a small overall diameter DisAdvantages: - More Prone to Electromagnetic interference and noise (EMI)	10 Base -T Ethernet	
	Category 5	Up to 100 Mbps		Fast Ethernet, Gigabit Ethernet	
	Category 5e	1 Gbps		Fast Ethernet, Gigabit Ethernet	
STP	Category 6/6a	10 Gbps	Advantages: - Shielded - Faster than UTP - Less Susceptible to noise and interference DisAdvantage - Expensive - Greater	Gigabit Ethernet, 10G Ethernet (55 m) Widely used in data centers	
SSTP	Category 7	10 Gbps		Gigabit Ethernet 10G Ethernet	

Coaxial Cable	RG-6 RG-59 RG-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	Single Mode Multi Mode	100 Gbps	Advantages - High Speed - High Bandwidth - High Security - Long Distance Disadvantages: - Expensive - Requires skilled installers	Maximum distance of fibre optics cable is around 100 metres

b) Make ~~[illegible]~~ own Ethernet Cross-Over Cable / Straight Cable

Tools and parts needed:

- Ethernet Cabling: CAT 5e is certified for gigabit supp but CAT 5 cabling works as well, just over short distances

- A Crimping Tool: This is an all-in-one networking too shaped to push down the pins in plug and strip and cut the shield off the cables.

- Two RJ45 plugs

- Optional two plug shields.

Step 1: To [illegible], begin by stripping shields onto the cable

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

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

Step 4: Once the Order is correct, bunch them together in a line, and if there are any that stick out further than others, strip them back to create an even level. The difficult aspect is placing these into the RJ45 plug without messing up the order. To do so, hold the plug with the clip side facing away from you and have the gold pins facing towards you.

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. Simply strip the cables back a little more.

Step 6: After the wires are securely sitting inside the plug, insert it into the crimping tool and push down. It should be placed correctly, but pushing too hard can crack the fragile plastic plug.

Step 7: Lastly Repeat for the other end Using Diagram B (to make a crossover cables) / Using diagram A (to make a Straight through cable).

To test it, plug it in and attempt to connect two devices directly.

Student Observation

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

Cross cables have crossed wiring for connecting similar devices, while straight cables ~~have~~ parallel wiring for connecting different devices

2. Which type of cable is used to connect two PC? (Straight / Cross Cable)

Cross cable

3. Which type of cable is used to connect a router / switch to your PC? (Straight / Cross cable)

Straight Cable

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

The category is typically Cat5e or Cat6.

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

Making a twisted pair cable involves arranging the wires in the correct order, facing challenges with precise cutting and crimping and ensure reliable network connections upon testing.

RESULT:

Thus the cable connection is done and executed successfully

Experiments on CISCO PACKET

(Simulation Tool)

Ex.NO:3 DATE: 27/7/24

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 existing network design / simulation and modelling tool.

1. It allows you to model complex system 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 devices.

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 a real hardware

INSTALLING PACKET TRACER:

To download Packet Tracer, go to https://www.netcad.com and log in with your Cisco Networking Academy credentials; then click on the Packet Tracer graphic and download the package appropriate for your operating system. (Can be pkt used to download in your laptop).

Windows

Installation in windows is pretty simple and straight forward; the setup comes in a single file named PacketTracer – Setup 6.0.1.exe. Open this file to begin the setup wizard, accept the license agreement choose a location and start the installation.

Linux

Linux users with an Ubuntu / Debian distribution should download the file for Ubuntu, and those using Fedora / Redhat / CentOS must download the file for Fedora. Grant executable permission to this file by using chmod and execute it to begin the installation.

Chmod +X PacketTracer601_i386_installer – rpm.bin
./ PacketTracer601_i386_installer – rpm.bin

USER INTERFACE OVERVIEW:

The layout of the Packet Tracer is divided into several components. The components of the Packet Tracer interface are as follows:

1. Menu bar - This is a common menu found in all software applications 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 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. Workspace - This is the area where topologies are created and simulations

5. Common tools bar - This toolbar provides controls for manipulating topologies, such as select, move layout, place note, delete, inspect, resize shape 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 of the network and end devices available with Packet Tracer and is further divided into two areas

i) Area 7a: Device-Type selection box - Contains Device Categories

ii) Area 7b: Device-Specific Selection box - When a device category is selected, this selection box displays the different device models within that category

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

b) Analyse the behaviour of network devices using CISCO PACKET TRACER Simulation.

1. From the network component box, click and drag-and-drop the below 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. Here, the default gateway and DNS server information is not needed as these are only two end devices in the network.

4. Click on the PDU (message icon) from the common toolbox, Drag and drop it on one of PC (source machine) and then drop it on another PC (destination machine) connected to the HUB.

5. Observe the flow of PDU from source PC to destination PC by selecting the Realtime mode of simulation.

6. Repeat step 3 and step 5 for the PCs connected to the switch

7. Observe how HUB and switch are forwarding the PDU and Write your observation and conclusion about the behaviours of switch and HUB.

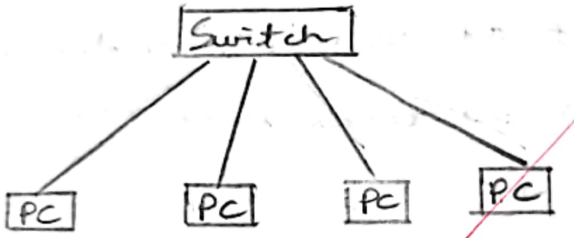
Student Observation

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

A HUB broadcasts packets to all ports, while a switch forwards packets only to the destination port based on MAC address

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

A star topology is implemented where each PC is connected to a central switch or hub.



6/6

ESULT

Thus the Packet T

Setup and Configure

Using a Switch and Ethernet cables in your lab

EX.NO: 4 DATE:

AIM: Setup and configure a LAN (Local Area Network) using a switch and Ethernet cables in your lab.

What is LAN?

A 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 connects devices to promote collaboration and transfer information between users, such as computers, printers, servers and switches. A local area network (LAN) switch serves as the primary connecting device, managing and directing communications within the local network. Each connected device on a LAN switch can communicate directly with each other, allowing for fast and secure data transfer.

How to set up a LAN

Step 1: Plan and Design an appropriate network topology taking into account network requirements and equipments location.

Step 2: You can take 4 computers, a switch with 8, 16 or 24 ports which is sufficient for networks of these sizes, and 4 ethernet cables.

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

Step 4: Assign IP address to your PCs

1. Log On on the client computer as Administrator or Owner
2. Click Network and Internet connections
3. Right click Local Area Connection / Ethernet

Similarly assign IP address to all the PCs connected to switch.

Step 5: configure a network switch.

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

2. Log in to the web interface: Open a web-browser and enter the IP address of the switch in the address bar. This should bring up the login page for the switch's web interface. Enter the username and password to log in.

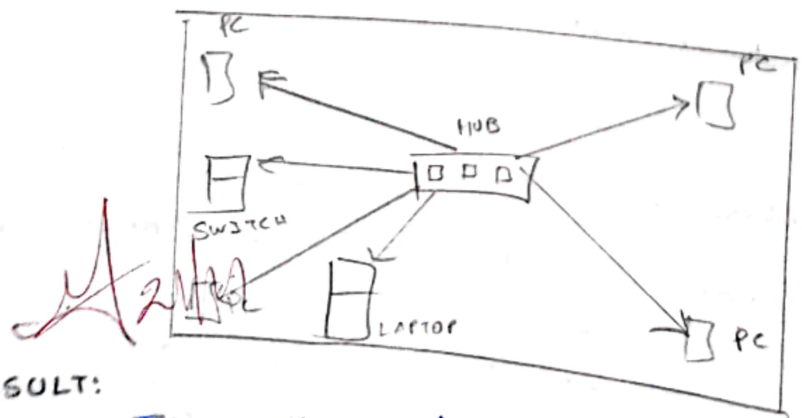
3. Configure the basic settings: Once you're logged in you will be able to configure basic settings for the switch.

4. Assign IP address as: 10.1.15, 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.

Step 7: Select a folder → goto 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



RESULT:

EX NO: 5 DATE:

AIM: Experiments on Packet capture tool : Wireshark.

PACKET SNIFFER: - Sniff Message Sent / Receive
- Store & Display the content Various protocol

STRUCTURE DIAGNOSTIC TOOLS:
- TCP dump
- WIRESHARK

WIRESHARK:

A network analysis tool formerly known as Ethereal, Capture packet in real time & display them human readable format Used to inspect suspicious program network. Capture Network, decode packet protocol, Analyze problem

CAPTURING AND ANALYSING PACKETS USING WIRESHARK TOOL:

To filter, capture, view, packet in wireshark tool.

Capture 100 packets from the ethernet : IEEE 802.3 LAN Interface and save it.

PROCEDURE:

- Select Local area connection wireshark
- Go to capture -> option
- Select stop capture automatically after 100 packet
- Then click start capture
- Save the packets

① Create a filter to display only TCP/UDP packet, inspect the packet & Provide flow graph

PROCEDURE:

- Select Local Area connection in wire Shark.
- Go to capture -> Option
- Select stop capture automatically after 100 packets.

- Then click
- Save the packets search TCP packet
- To see flow graph click statistics - flow graph.

2) Create a filter to Display only TCP/UDP packets inspect the packets and provide the flow graphs.

3) Create a filter to Display only TCP packets inspect the packets.

4) Create a filter to display only ARP packets to inspect the packets.

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

6) Create a filter to display only HTTP packets to inspect the packets.

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

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

Student observation:

1) What is promiscuous Mode?

This promiscuous mode is a configuration for a network interface that allows is to capture all packets on the Network segment it is connected for not just the packets intended in it.

2) Does ARP packets have to transport layer header? Explain

No, ARP packets do not have a transport layer because it operates at the data link layer of the OSI Model. It is used to map IP addresses to MAC Addresses.

3) Which Transport layer protocol is used by DNS?

DNS primarily uses UDP as its transport layer protocol but it also uses TCP for larger resources.

4) What is the port Number used by HTTP protocol?

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

5) What is a Broadcast IP Address?

It is a special used to sent packets to all devices on a specific network or subnet Mask.

23/11

Result:

Thus the experiment on wireshark using packet sniffer has been implemented and captured successfully

Ex. No: 6. Implement error detection correction using Hamming code concept.

Aim:

Write a program to implement error and correction using HAMMING code concept. Make a test run to inspect data stream and verify error correction feature.

Error correction at data link layer:

Hamming code is set of error correction codes this can be used to detect and correct the error that can occur when the data is transmitted from the sender to the receiver, It is a technique developed by P.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 its.

3. save this output in a file called channel

```
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 i in range(0, len(
                                        binary),8)]
    return ''.join([chr(int(char, 2)) for char in
                                        chars])
def calc_redundant_bits(m):
    r=0
    while(2**r < m+r+1):
        r+=1
    return r
def pos_redundant_bits(data, r)
    j=0
    k=0
    m=len(data)
    res=''
    for i in range(
```

```
        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
        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("error detected at position:{res}")
            data = list(data)
            if res<=n:
                data[res-1] = '0' if data[res-1]=='1'
        print(f"error corrected at position:
                {res}").
        else:
            print("error position out of range
                No correction performed")
        else:
            print("No error detected")
            return data
```

```
    for i in range(1, len(data)+r):
        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
    data = list(data)
    data[position-1] = '0' if data[position-1] == '1' else '1'
    print(f"Introduced error at position: {position}")
    return ''.join(data)
```

Sender program.

```
def sender(text):
    binary_data = text_to_binary(text)
    m = len(binary_data)
    print(f"Sender output (binary with redunant bits)")
    return arr,

def receiver(data):
    r = calc_redunant_bits(len(data))
    corrected_data = detect_and_correct(data, r)
    original_data = remove_redunant_bits(corrected_data, r)
    print(f"decoded text: {ascii output}")
```

main program:

```
if __name__ == "__main__":
```

result:

The program

Ex.No:7 Sliding window protocol. 24/9/24

Aim:

Write a program to implement flow control at data link layer using sliding window protocol simulate the flow of frames from or node to another.

Procedure:

1) The sliding window protocol controls the flow of data between the sender and receiver with a fixed window size n.

2) The sender can send up to N frames without waiting for Acknowledgement.

3) After sending, the widow "slider" forward as acknowledgement are received.

4) The receiver process frames in sequence and acknowledges them.

5) Lost or corrupted frames use retransmitted, ensuring reliable and order delivery.

```
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):
    print("\n--Sending Frames--")
    for i in range(window_size):
        if i<len(frames and not frames[i].
        if random.random()<0.2:
            print("received Frame{
            frame[i].frame_no}: {frames[i].
            data}.data}[ERROR]")
            frames[i].acknowledgement = false
        else:
            print("Received Frame {frames[i].
            data}[ERROR]").
```

```
def sliding_window_protocol():
    window_size = int(input("enter window size:"))
    message = input("Enter a message to used:")
    frames = [frames(i, message[i]) for i in
        range(len(message))]
    base = 0
    while base < len(frames):
        send_frames(frames[base:], window_size)
        time.sleep(2)
        receive_frame(frames[base:], window_size)
    while base < len(frames) and frame[base]
    acknowledged:
        base += 1
    if base < len(frames):
        print("\n Resending in acknowledged
                frames... \n")
        time.sleep(2)
        print("\n Resending in acknowledged
    frames... \n")
        time.sleep(2)
        print("\n frames send and
    unacknowledged")
if __name__ == "__main__":
        sliding_window_protocol()
```

Output:

```
Enter window size: 5
Enter a message to send: HELPP
---- Sending Frames ---
Sent Frame 0: H
Sent Frame 1: E
Sent Frame 2: L
Sent Frame 3: P
Sent Frame 4: P
```

Frame sent, waiting for acknowledgement

--- Receiving Frames ---

Received frame 0: H [Received]
Received frame 1: E [Received]
Received frame 2: L [Received]
Received frame 3: P [Received]
Received frame 4: P [Error].

--- Resending frames ---

Received frames 4: O [Received].

All frames are sent and acknowledgement.

Result:

Thus the code for flow control acknowledgement is executed successfully.

Ex. NO: 8(a)
Date: 1/10.

Aim:
To simulate a virtual LAN (VLAN) configuration using CISCO packet tracer, follow these step-by-step instruction

Step 1: Set up the topology.
1. Open cisco packet tracker
2. Drag and Drop switches (CISCO 2960) and PCs into the workspace.
3. Use copper straight ~~th~~ through cables to connect the PC's to the switch using their Fast ethernet ports.

Step 2: Assign IP addresses to PC's
1. Click on each PC, go to the Desktop tab and click IP configuration.
2. Assign IP addresses to each PC.

- PC1 - IP. ~~199~~ 192.168.1.10 Subnet Mask: 255.255.255.0
- PC2 - IP - 192.168.2.10 Subnet Mask: 255.255.255.0

Step 3: VLAN configuration on the switch:
1) Click on the switch and open the CLI (Command Line Interface)
2. Enter the following commands to create VLAN's and assign the two ports to different VLAN's

Create VLAN's.
Switch > enable.
Switch # configure terminal
Switch # VLAN 10.
Switch (config-vlan)# name.
VLAN-10.
Switch (config-VLAN)# name
Switch (config)# VLAN 20.
VLAN -20.

Assign switch ports to VLANS:

For PC1 connected to Fast ethernet 0(1), assign it to VLAN 10:

```
switch (config) # interface.
fast ethernet 0/1
switch (config-if) # switchport
mode access
switch (config-if) # switchport.
access VLAN 10
switch (config) # exit.
```

For PC2 connected to fast Ethernet 0(2) assign it to VLAN 20:

```
switch(config) # interface
fast ethernet 0/2.
switch(config-if) # switch-port.
mode access.
switch (config-if) # switchport.
access vlan 20.
switch (config-if) # exit.
```

Step 4: verify VLAN configuration

1) Run the following command to confirm VLAN's are properly configured.

Switch # show VLAN Brief.

Step: 5 Test connectivity

1) Go to PC1: Open the command prompt and Ping PC2's IP address (192.168.2.10)

2) Since PC1 and PC2 are in different VLANS; they should not be to communicate with each other. (you will receive timeouts).

Step 6: Save the configuration.

Finally save the configuration using.

Switch # copy running-config startup config.

b) Configuration of wireless LAN using a CISCO packet tracers.

Block A Block B

Robotics DEPT Block L

VLAN 10.

1) Click network devices Down click wireless devices select wireless Router in centre
2) Add 8 PC's click End Devices.
3) Click router GUI tab click given IP addresses as (192.168.0.1) click Disabled & save setting.
4) Come up again click wireless, wireless settings change Network name (SSID to home network. Save settings)
5) come up, click wireless settings, Security Mode change to WEP give key 1 as 0123456789, save
6) close tab.
7) Click PC(1) desktop, IP configuration
   IP address : 192.168.0.5,
   Subnet Mask : 255.255.255.0.
   default gateway : 192.168.0.1 ; close tab,
8) Go to Next PC(2) desktop · IP configuration
   IP address : 192.168.0.6.
   Subnet Mask : 255.255.255.0.
   Default gateway : 192.168.0.1 close tabs.
9) Go to next PC(5) desktop. IP configuration.
   IP address : 192.168.0.7.
   Subnet Mask : 255.255.255.0
   Default gateway : 192.168.0.1 close tabs.
10) Again click PC(1) PC wireless → error will come, close it.
11) On top, go to Physical tab, Turn off the pc by clicking the red Button.

12) Now turn on the PC, by clicking the red. Button. Go to desktop tab on the top, PC wireless, connect, wireless mode, connect WEP key (5) 0123456789, connect now when you will the link information it will show you have successfully connected close tab.

13) Now PC(2) again PC wireless (Repeat step 10/11/12)

14) PC(3) =) (Repeat step (10, 11, 12).)

15) to check the connectivity whether all is connected click any PC command prompt, any IP address given during this process. Ping 192.168.0.6 Enter.

OUTPUT -

a)

VLAN	NAME	Status	Ports
1	default	activ	Fa 0/6, Fa 0/6 Fa 0/7, Fa 0/8. Fa 0/24 Gib 0/1, Gig 0/2.
10	Marketing	active	Fa 0/1, Fa 0/2:
20	sales	active	Fa 0/3, Fa 0/4.
1002	fad 1. default	activ	
1003	token. ring default	active	

b) Ping 192.168.0.6

Pinging 192.168.0.6 with 32 bytes of data:

Request timed out.

Ping statistics for 192.168.0.6.

Packets: Sent = 4 Received = 0 Lost = 4 (100% loss)

Result:

Thus to simulate VLAN configuration using CISCO packet simulator and configuration of wireless LAN using CISCO packet tracer is done executed successfully.

Student observation:

1) What is SSID of a wireless router?

The SSID (Service Set Identifier) is the name of a wifi-network that allows devices to identify and connect to it. It's a ~~~~ unique label ~~identify~~ ~~and to~~ for each wireless network, helping distinguish core from another when multiple networks and available. SSIDs can be upto 32 characters long and are broadcasted by default, making them visible to nearby devices. When your connect to wifi, you select the SSID of the network you want to join

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

A security key in a wireless router is the password used to protect a wifi network. It ensures only authorized users can connect and encrypts data sent over the network. Common types includes WPA2(WPA3 ~~to~~ keys which are secure and the older, less secure WEP key. When you join a WIFI-network you must enter this security key for access.

3) Configure a simple wirelessLAN in your LAB using a real access point and write down the configurations in your Notebook.

1) Setup access point (AP):

- Power up the AP and connect your computer to it via Ethernet.

2) Access AP configuration:

- In a Browser, enter the AP's default and login with default credentials.

3) Configure SSID:

Set you network name.
Enable SSID broadcast

4) Set wireless security:

- Choose WPA 2-PSK or WPA 3 for security.

• Set a password.

5) configure wireless channel:
• Set the channel to Auto or manually select a channel.
• choose the band 2.4GHz or 5GHz.

6) Enable DHCP:
• Enable DHCP if requested to assign IP address

7) Save settings:
and restart the AP

8) test connection: by connecting devices to the SSID using the password.

Notebook Summary:
1) SSID: lab-WLAN
2) Security: WPA2-PSK,
3) key: Lab@12345
4) Channel: Auto,
5) Band: 2.4GHz or 5GHz,
6) DHCP: Enabled.

Result:
The experiment for implementation of subnetting in CISCO packet tracer simulator is executed

Ex. No: 7.
Date: 9/10/24.

Aim:
Implementation of SUBNETTING IN CISCO PACKET TRACER simulation.

Steps:
1) Create a network topology,
* Open a CISCO packet tracer,
* Click on New > Network > Generic to create a Blank topology.

2) Add Devices:
- Add the following devices,
  - 2 routers (R1, R2)
  - 2 switches (S1, S2)
  - 10 PCs (5 for each subnet)
- connect the devices.
  - use the appropriate cables to connect
    - R1 to S1.
    - R1 to R2
    - S2 to R2.

3) Subnetting configuration:
- Network address: 192.168.1.0/24
- Subnet Mask: /27 (provides 8 subnets hosts addresses each

4) IP addressing scheme:
* Router R1
  - Gigabit Ethernet 0/0: 192.168.1.1
  - Gigabit Ethernet: 0/1: 192.168.2.2.

Switch S1:
- Fast Ethernet 0/1: 192.168.1.0/27.

PCs
- PC1 192.168.3.11
- PC2 192.168.3.12
- PC3 192.168.3.13
- PC4 192.168.3.14
- PC5 192.168.3.15.

Switch S2:
Fast ethernet 0/1: 192.168.1.0/24.
PCs PC1: 192.168.3.11
PC2: 192.168.3.12
PC3: 192.168.3.13
PC4: 192.168.3.14
PC5: 192.168.3.15.

1) Configuring the devices:
- Open CLI on Router R1 and router enable.
- configure terminate
- Interface Gigabit ethernet 0/0.
- IP address 192.168.2.1 → 255.255.255.224
not shutDown.
- Exit
- Switch configuration:
- open CLI on Switch S1 & enter enable.
configure terminal.
Interface fastethernet 0/1
Interface mode acces.
Switchport mode access.
Exit
- Pc configuration:
- Right click on each Pc and select config.

Enter:
- IP address subnet mask (255.255.255.224)
default gateway (Router ID)

by testing the networks:
- Open the command prompt on each PC
- Use the ping command to check connectivity between PC's and the router Ping 192.168.1.x (for PCs in the first subnet) Ping (192.168.2.x (for pc's in the second subnet)

4) Conclusion:
If all pings are successful your subnetting and network configuration in Cisco packet tracer is functioning correctly.

Student Observation:

1) Write down your understanding of Subnetting.

Subnetting is the pratice of dividing a large Network into smaller, manageable sub-Networks to improve performance and security. It uses a subnet Mask to define the Network and host portions of an IP Address.

2) What is the advantage of Implementing subnetting within a Network?

1) Improved Performance: Reduces Broadcast traffic and congestion.

2) Enhanced Security: Isolates sensitive data and control access.

3) Find out whether subnetting is Implemented in your college. If yes, draw and list down.

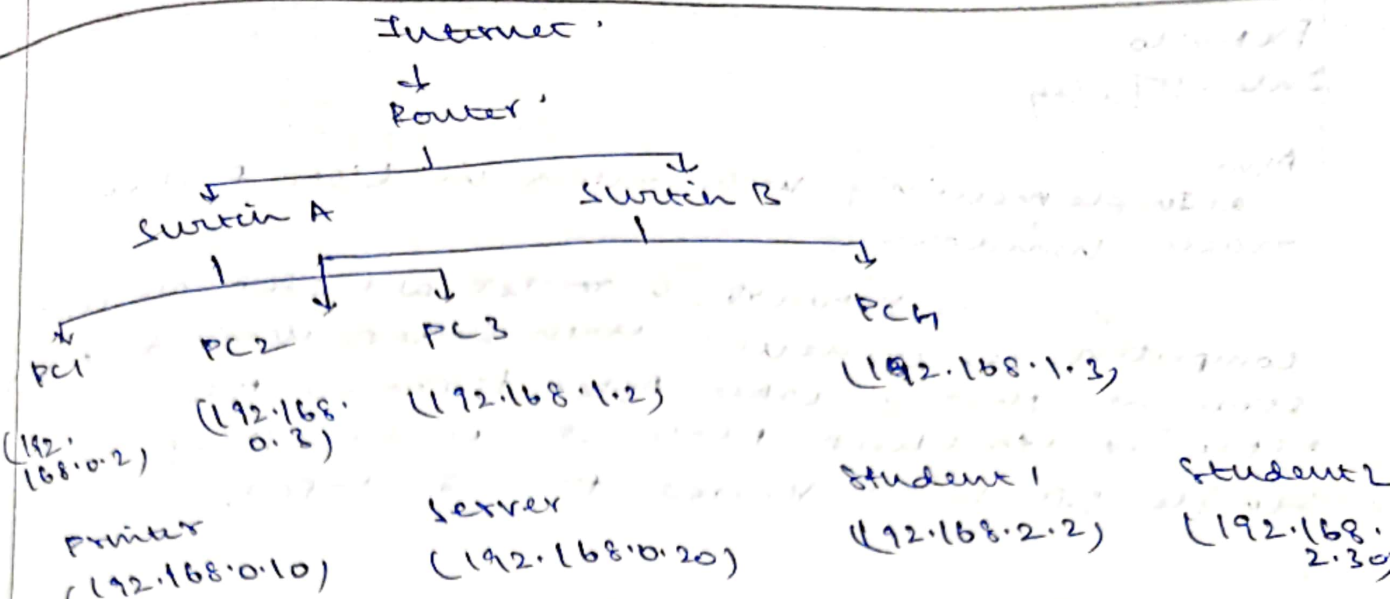
Subnets used with IP Address.

Main Network:

- Network address: 192.168.0.0/22
- Total Address: 1024

Subnet used:

Subnet Name	Subnet Address	Subnet Mask	IP Router	Total host
Academic departments	192.168.0.0 /24	255.255.255.0	192.168.0.1 - 192.168.0.255	254
Administration	192.168.1.0/24	255.255.255.0	192.168.1.1 - 192.168.1.255	254
Student Housing	192.168.2.0/24	255.255.255.0	192.168.2.1 - 192.168.2.255	254
Library Services	192.168.3.0/24	255.255.255.0	192.168.3.1 - 192.168.3.255	254

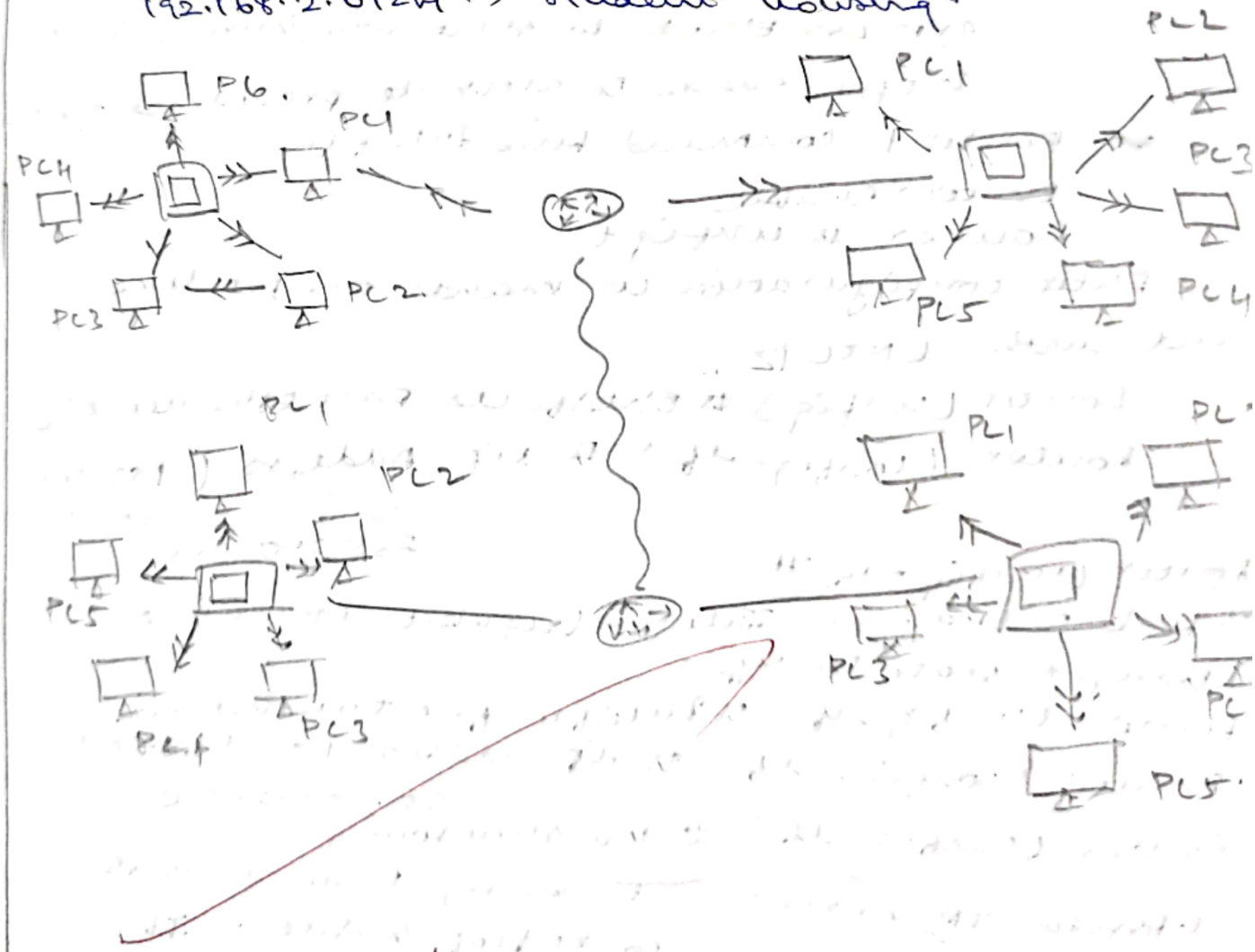


subnets:

192.168.0.0/24 - Academic department

192.168.1.0/24 - Administration

192.168.2.0/24 → Student housing.



Result:

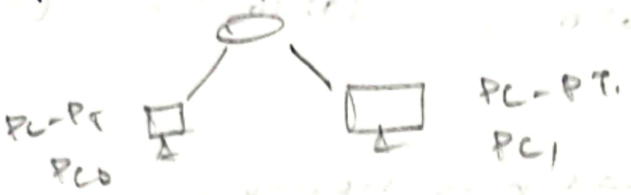
Thus the implementation of subnetting in Cisco packet tracer simulator is done and executed successfully.

4/2/24

Ex.No: 10.
Date: 15/10/24

Aim:
a) Implementing with routers in Cisco Packet tracer window

In this network, a router and 2 PC's are used. Computer are connected with routers using a copper straight-through cable. After forming the Network, to check network connectivity a simple PDU is transferred from PC to PC1



Procedure:

Step 1: 1) Select router and open CLI,
2) Press ENTER to start configuring router
3) Type enable to activate privileged mode

Router | command line interface
Router > enable
Router # config t.
Enter configuration commands one per line and with CNTL/Z,
Router (config) # Interface Fast Ethernet 0/0.
Router (config-if) # ip Address (192.168.10.1) 255.255.255.0.
Router (config-if) #
%. Link 0- changed: Interface Fast Ethernet 0/0 changed state to up.
Router (config-if) # Interface fast Ethernet 0/1.
Router (config-if) # ip address 192.168.20.1. 255.255.255.0.
Router (config-if) # no-shutdown.

1. Assign IP addresses to every PC in network
2. Select the PC, go to desktop & select IP configuration and assign an IP address default gateway subnet mask.
3. Assign the default gateway of PC as 192.168.10.1.

Step 3: Connecting PC's with routers.

1. Connect Fast ethernet 0 port of PC0 with Fast ethernet 0/0 port of Router using a copper straight through cable.
2. Connect Fast ethernet 0 port of PC1 with fast ethernet 0/1 port of router 1 using a copper straight through cable.

Router configuration table:

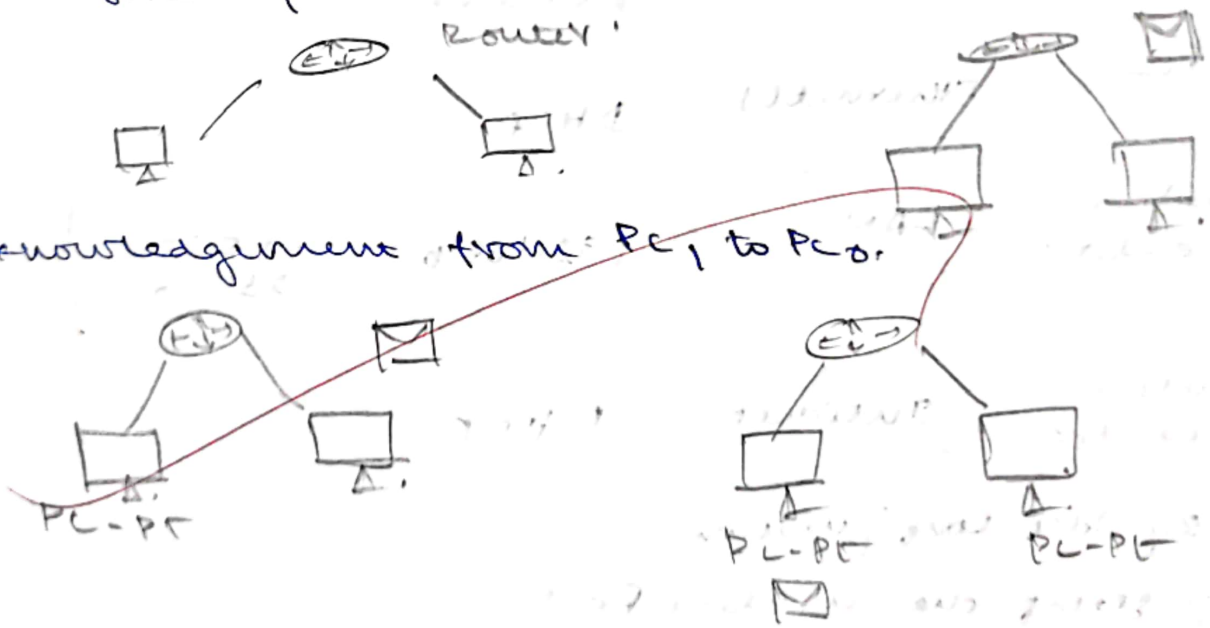
Device Name	IP address Fast Ethernet 0/0	Subnet Mask	IP Address Fast Ethernet	Subnet Mask
Router1	192.168.60.1	255.255.255.0	192.168.20.1	255.255.255.0

PC configuration table:

Device Name	IP address	Subnet Mask	Gateway
PC0	192.168.60.2	255.255.255.0	192.168.60.1
PC1	192.168.20.2	255.255.255.0	192.168.20.1

Designed Network Topology:

Sending a PDU from PC0 to PC1:



Acknowledgement from PC1 to PC0.

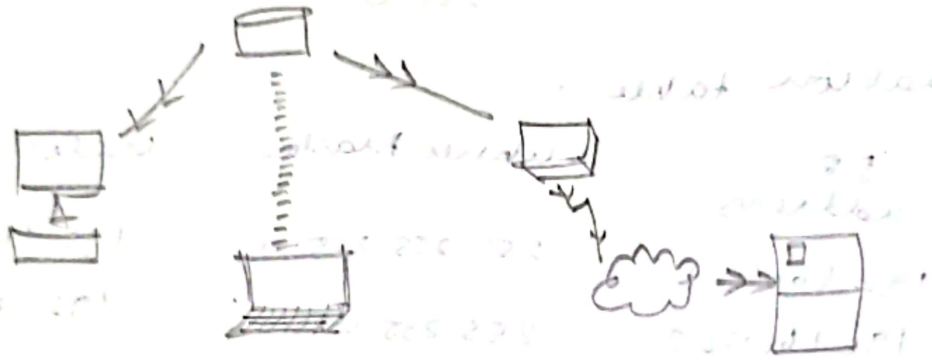
Practical:

Design and configure an interworking using wireless router, DHCP server and internet cloud.

Step 1:

component required.

- Wireless Router
- DHCP server
- Internet cloud.
- End Devices.
- Ethernet cables.



Addressing table:

Device	Interface	IP Address	Subnet Mask	Default Gateway
PC	Ethernet()	DHCP.		
Wireless Router.	LAN	192.168.0.1	255.255.255.0.	
Wireless Router	Internet	DHCP		

Step By step configuration:

Step 1: Setup the wireless Router.

- Physical setup,
  - Connect the Internet source (ICP) to the router WAN port.
  - Connect any wired devices via LAN ports on the Router.
- Access Router Admin Panner,
  - Connect a device to the
  - Open a web B

*Go to the wireless settings
* Set the SSID (network name) and password for the WIFI network.
* Choose the encryption method.
Step 2: configure the DHCP server:
* Most modern routers have a built in DHCP server that assigns IP address to devices
* Enable DHCP under the LA settings
* Set the IP address range.
* Specify the least time for IP address
* Internet cloud setup:
The Internet cloud represent your connection to the wider internet. If you are simulating the network you can use cloud. devices in simulation software like USB Packtet on GNS3

GNS3:
* For a real world setup, make sure your router is connected to the ISP's modules or directly to the Internet, Port
. verify the WAN combination type in the network settings

Step 4: configure devices:
* For wireless devices, connect them to the wireless network using the SSID and password you set earlier.
* the devices should receive an IP address automatically from the DHCP server.

Step 5: Ping test: from a connected device, open the command prompt or terminal and ping an external IP to verify internet connectivity.
Check IP address: run ipconfig (Windows) to check if the device has received a correct IP address from the DHCP server

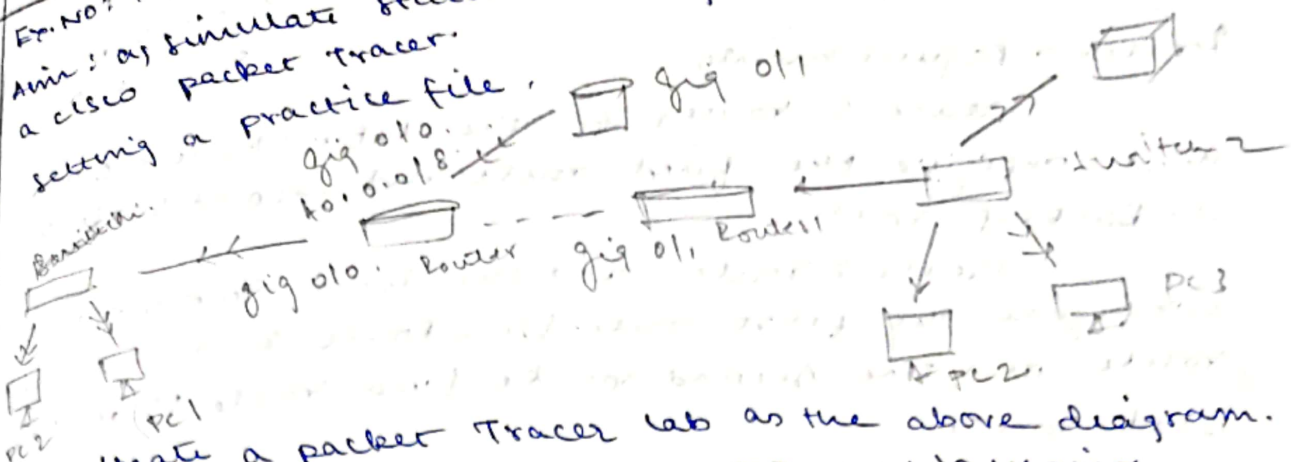
Step 6: Port forwarding if certain services need to be accessed from the Internet, you may need to configure port forwarding.

18/10/24

Ex. No: 11(A)

Aim: To simulate Static Routing configuration using a cisco packet tracer.

Setting a practice file.



Create a packet Tracer lab as the above diagram.

Packet tracer lab with Initial IP configuration

If we specify 2 routes to reach the same destination, the router automatically selects the best route and another router that has the least No. of routes that has the least No. of routes. If you manually wants to select a route that the router should add to the routing table you have to set the AD value of the route lower than the routes.

```
# ip route    30.0.0.0.255.0.0.0.20.0.0.2.10
# ip route    30.0.0.0.255.0.0.0.40.0.0.2.20.
```

If the first router fails the router automatically add the second route to the routing table.

Routers automatically learn their connected Networks, we only need to add routes for that Network that are Not Available on the interface For Example, Network 10.0.0.0/8, 20.0.0.0/8 and 40.0.0.0/8 are directly connected to Router 0. Thus we dont Need to configure routes for these Network. Network 20.0.0.0/8 & Network 50.0.0.0/8. are Not Available on Router 0.

Router	Available Networks.	Networks Available
Router 0	10.0.0.0/8 20.0.0.0/8 40.0.0.0/8	30.0.0.0/2, 50.0.0.0/8.
Router 1	20.0.0.0/8 30.0.0.0/8	10.0.0.0/8

Router 2 — 40.0.0.0/8, 50.0.0.0/8, 10.0.0.0/8, 20.0.0.0/8, 30.0.0.0/8.

Router 0 Requirements:

Create 2 routes for the Network 30.0.0.0/8 and configure the first route and second route as a Backup route.

Create 2 routes and the route 30.0.0.0/8 and configure the first route (via Route 2) as the main route and the second route (via route 1) as the Back route.

Router > enable.
Router # configure terminal
Router (config)# ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router (config)# ip route 30.0.0.0 255.0.0.0 40.0.0.2
Router (config)# ip route 30.0.0.0 100.255.255.255.255.20.0.0.22
Router (config)# ip route 50.0.0.255.0.0.0.40.0.0.
Router (config)# ip route 50.0.0.0 255-0.0.0.20.0.0.2.
Router (config)# exit
Router # show ip route static

Router 1 requirements

Create 2 routes for Network 10.0.0.0/8 and configure the first route as the main route and the main route and the second route as Backup route.

Create 2 routes for Network 40.0.0.0/8 and configure the first route (via - Router 0) as the Main route and second route as a Backup route.

Router 2 requirement

Create static routers for Network 10.0.0.0/8 and network 30.0.0.0/8 and verify the router adds both Router do routing table.

verifying static routing:

By sending ping requests to PC Networks

These routes via router and via

packet tracer lab of the above example.

Deleting a static route:

To delete a static route.

To delete a static route, use the following steps.

Use the show ip route static command to print all static routes.

Move down the router you want to delete.

Backup route → Main route → when you delete the main route.

Result: Simulating static Routing configuration using CISCO packet Tracer is executed and verified.

Scanned with OKEN Scanner

Date

Simulate RIP using CISCO packet tracer.

Aim:

to simulate RIP using CISCO Packet tracer.

Steps:

* Drag and Drop at least 2-3 Routers onto the workspace.
* connect the routers using appropriate cables
* Add PCs to each connected Network to the Routers.

Configure the ROUTERS:

Click on the Router 0, go to CLI tab enter the commands.

```
Router >enable
Router # configure terminal.
Router (config) # interface fast Ethernet 0/0
Router (config-if) # no shutdown
Router (config-if) # exit
```

Repeat the above configuration for all router interfaces connecting to other routers or networks.

Enable RIP on the Routers:

enter the following commands.

```
Router 0 (config) # router rip.
Router 0 (config-router) # Network 10.0.0.0
Router 0 (config-router) # network 192.168.1.0.
Router 0 (config-router) # Network 192.168.1.24.
```

Router 1:

```
Router 1 (config) # router rip
Router 1 (config-router) # Network 192.168.0.0
Router 1 (config-router) # network 192.168.1.248.
```

Router 2

```
Router 2 (config) # router rip.
Router (config-) # Network 20.0.0.0.
Router 2 (config-router) # Network 192.168.1.2.
Router 2 (config-router) # network
```

Ex: 12-1a)

Implement echo client server using TCP/UDP sockets.

TCP echo client server Algorithm

TCP server Algorithm:

1) Create a TCP socket
2) Bind the socket into a local address and port.
3) Listen for incoming client connections
4) Accept a client connection
5) Loop
6) Close the connection.

TCP-Client Algorithm:

1) Create a TCP socket.
2) Connect to the server using specified address and port.
3) Send a message to server.
4) Receive the message.
5) Loop.
6) Close the connection

TCP-Client Algorithm:

1) Create a TCP socket.
2) Connect to the server using specified address and port.
3) Send a message to server.
4) Receive the message from the server.
5) Display the received message.
6) Close the socket.

tcp-server.py:

```
import socket
def tcp_server():
    sever_socket () = socket.socket (Socket.
        AF_INET, socket.sock_stream).
    server_socket.bind ("localhost",
    server_socket.
```

```
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(AF_INET,
                                  socket.SOCK_STREAM)
    client.socket.connect(("local host"),12345))

try:
    message = input("Enter a message to send")
    client_socket.sendall(message.encode())
    data = client.socket.recv(1024)

finally:
    client_socket.closed()
if __name__=="__main__":
    tcp client()
```

Output:

```
>>>python tcp_client.py
Enter a message to send: Hi, I am Anand
received from server: Hi, I am Anand.
```

4231/7

Result:

Thus the program to implement echo client server using TCP is

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 a connected clients start a new process

3. For each connected clients start a new checking for new Message

4. Keep running the process till the server stops.

chat_client

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

2. Start a process by creating a to listen to Messages.

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).
            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.connect((host, port))
    print("connected to
```

client_socket.send(message.encode("UTF-8

chat-server.py:

```
import socket
import threading
def handle_client(client_socket)
while True:
    try: message = client_socket.recv(102
    If not message:
        break
while True:
    client_socket.addr = server.accept()
    print(f"New connect from {addr}")
    start() args = (client socket)
if __name__ == "__main__":
```

output:

```
>python chat-server.py
chat server started on 127.0.0.1:12345.
New connection from ('127.0.0.1', 57220)
>python chat-client.py
You: Anand
You: Server-Received.
```

23/11

Result:

Thus the program to Implement the client server using TCP is executed successfully

Practical - 12:

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.AF_INET, socket.
    socket.SOCK_DGRAM) as s:
        s.bind((HOST, PORT))
        print(f"UDP server (HOST='127.0.0.1',
          PORT=12345)
        while True:
            data, addr = s.recv from (1024)
            print(f"Received message from
{addr}: {data.decode()}")
            s.send to (b'pong', addr)
if __name__ == "__main__":
        start_server()
```

client.py.

```
import socket.
def start_server(HOST='127.0.0.1', PORT=
                                    12345)
try:
    s.bind((HOST
    print(f"UDP server running on
                        {host}:{port}")
while True:
    data addr = recv
```

```
        s.sendto (b'pong', addr)
    except socket.timeout:
        print ("Request timed out")
```

output:

Server,

```
>python ping-server.py
UDP Server running on 127.0.0.1:12345
Received message from (127.0.0.1, 5734)
```

Client:

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

23/4

Result: Thus the program to Implement ping program is executed successfully.

EX.NO:14

Date: 29/10/24

Aim:

write a code using RAW sockets to implement Packet sniffing.

Algorithm:

- Create a raw socket
- Continuously capture incoming packets using recfrom()
- Parse and display information like the source and destinated IP address and protocol.
- Close the socket after the capture process

Code:

```
from scap.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP
if IP in packet[IP]
    proto = ip_layer.protocol
    src_ip = ip_layer.src
    dest_ip = ip_layer.dst
    protocol_name = " "
    if protocol == 1:
        protocol_name = "TCP";
    else if protocol == 6:
        protocol_name = "IP";
    else
        protocol_name = "Unknown protocol"
def main():
    sniff(prn = packet_callback, filter = "ip",
          (Score = 0)
    if __name__ == "__main__":
        main();
```

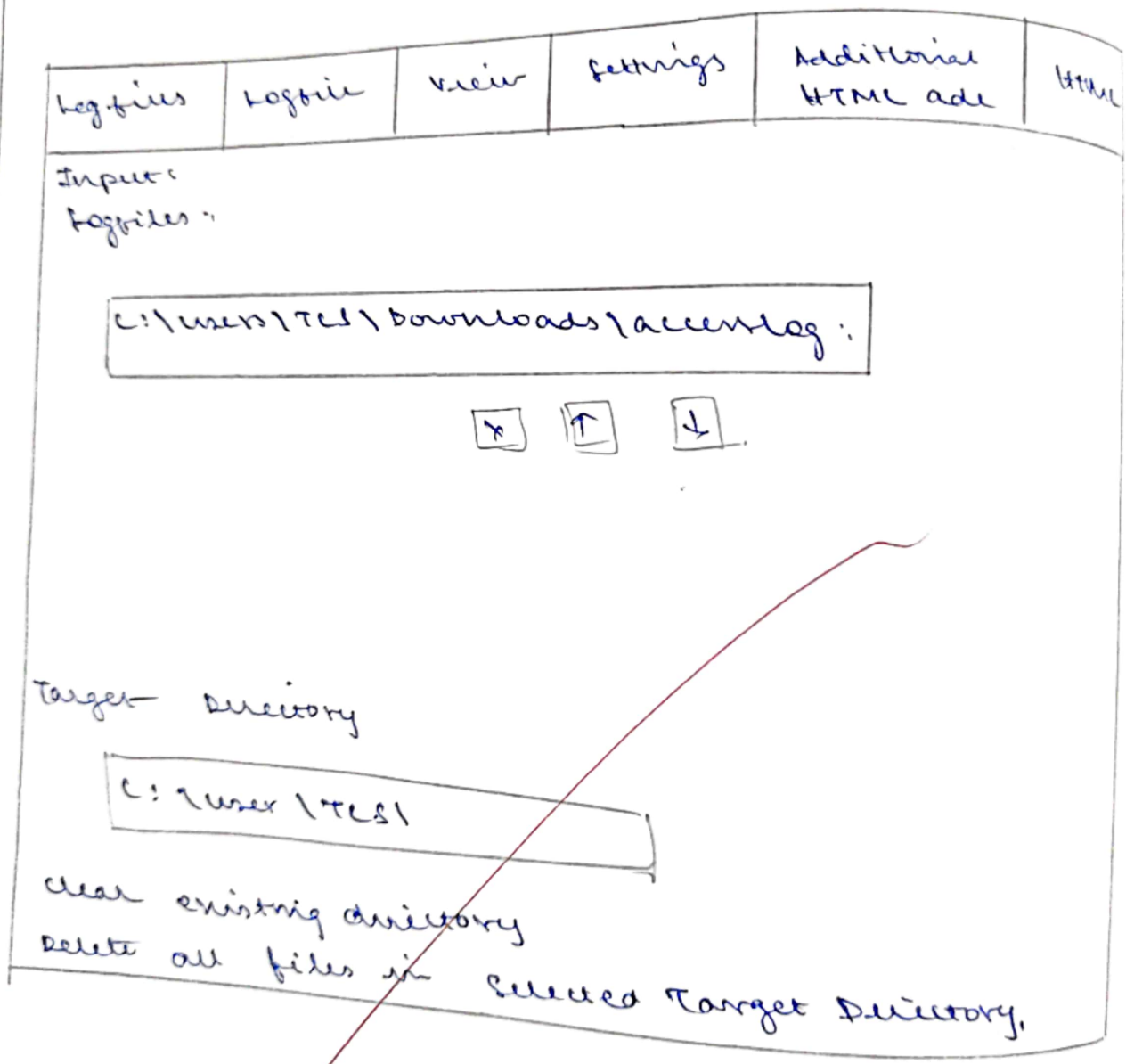
5/4/24

Ex No: 15

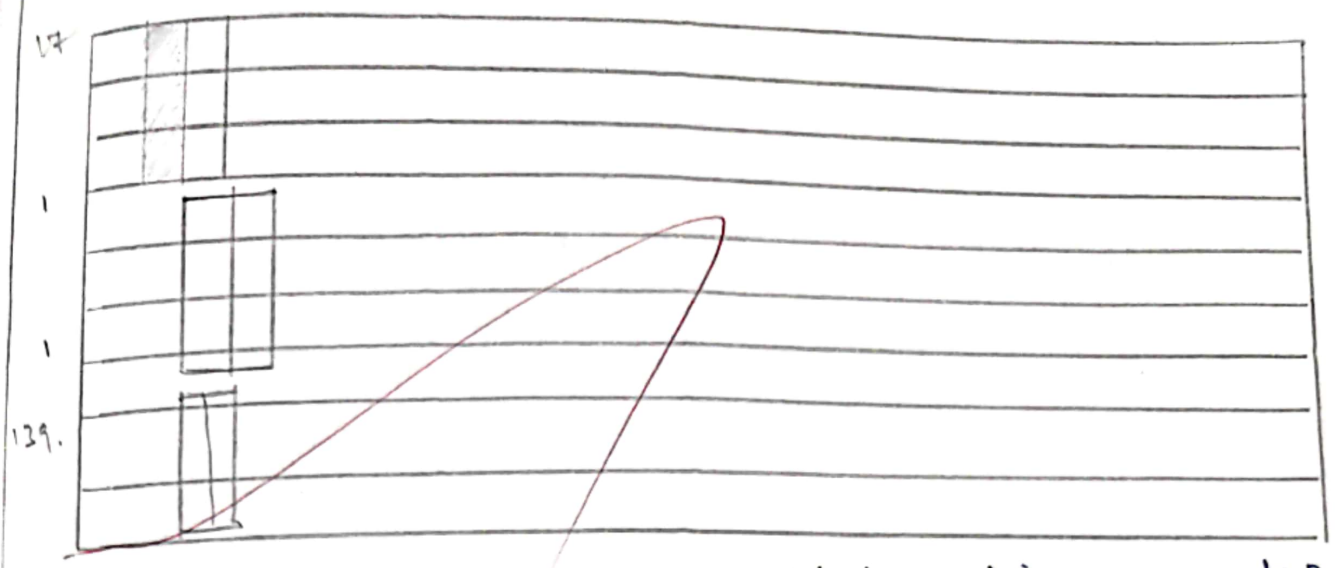
Aim to analyse the different types of web logs using Webalizer tool.

Procedure:

S1: Run webalizer windows version
S2: Input web log file
S3: Press run webalizer.



Daily usage for November 2024.



Days	Hits	Files	Pages	Visits	Sites	KBytes
3	67.10	651000	110001	110001	116000	13910080

Result: Thus the procedure to analyse the

Copyright