INDEX

Name A. Ashwin Subject CN

Standard Section Roll No.

School / College

S. No.	Date	Title	Page No.	Teacher's Sign
1	13/7/24	Basic network commands		
2	27/7/24	Study of types of cable		
3.	30/7/24	Study of packet tracker tool		
4.		configure b/w LAN & Switch Ethernet cable		
5.		Packet capture wireshark		
6.		Error detection and correction using hamming code		
7.		flow chase control using sliding window protocol		
8.		a) Virtual LAN configuration b) Simulate of wireless lan		
9.		Implement subnetting		
10.		a) Internet working with subnet b) DHCP		
11.		a) Static routing configuration b) simulate RIP using CISCO		
12.		a) Echo client server using TCP/UDP b) chat client-server using TCP/UDP		
13.		Implement ping program		
14		Implement snmp in RAW socket		
15		WEBALIZER		

Exp: 1

Date: 15/7/24.

Practical - 1.

Basic Network commands.

Aim:

Study of various network commands used Linux and windows

Basics Network commands:

arp -a: Arp is a short form of a address resolution protocal, It will show IP address

Host name: This will display the name of a computer.

Ipconfig/all - Display detailed configuration info about TCP/IP connection.

nbtstat -a - Solve problem with NetBIOS name resolution

netstat :- (network statistics) displays a variety of statistics about computer active TCP/IP connections.

nslookup: is a tool used to perform DNS lookups in Linux.

Pathping: pathping is unique to windows and Basically a combination of the ping and Tracert commands.

Route: It is used to show IP routing table.

Experiment 1: Study of various Network commands used in Linux and Windows.

Basic Networking command:

1. arp -a:

Output: Interface: 172.16.75.54 ... 0x12

Interface Address	physical Address	Type
172.16.72.1	7c-59-1c-cf-be-91	dynamic
172.16.72.133	4c-ac-a3-65-97-f5	dynamic

Hostname:

Desktop - COIBHTD

nbstat:

Active connections

Proto	Connection	Forgin Address	State
TCP	127.0.0.1:4978	DESKTOP-COIBHTD: 49679	ESTABLISHED

nslookup:

Default server: unknown

Address: 172.16.472.1

Pathping:

usage: pathping [-g host-list] [-h Maximum-hops] [-address] [-h] [-P period] [-q num-queries]

Route: ROUTE [-f] [-p] [-4/-6] command [destination] [MASK netmask] [gateway] [METRIC Metric] [IF Interface]

-6 Force using IPV6

1. ip - Basic command

shows Address Information, Manipulation routing, plus display Network Various Devices, Interface Tunnels.

ip address show::

enp250: < BROADCAST, MULTICAST, UP, LOWER UP>
1500q disc fq-code State up group default
gron 1000 link/ether 50:99:4C:35:0C:f5 brd
ff:ff:ff:ff:ff: inet 172.16.3.92/22 brd 172.16.11.255
Scope global enp280.

Valid-lft forever preferred-lft forever
inseat6 fe80::id125:ecf6:d561:1d1a/65 Scope
link Valid-lft forever preferred lft forever.

ip <options> <object> <command>

2. ifconfig

enp280: flags = 4163 < UP, BROPCAST, RUNNING,
MULTICAST > mtu 1500 inet 172.16.3.92 netmask
255.255.252.0 broadcast 172.16.11.25

Here, are the common use cases to the ip command

a) To show the IP addresses assigned to an Interface on your server:

lo: <LOOPBACK, PROMISC, UP, LOWER-UP> mtu 65536
q disc noqueue state UNKNOWN group Default
glen 1000 inet 127.0.0.118 slope host lo

Valid-lft forever preferred-lft forever.

b) [root@server n] ip address add 192.168.1:254 124
dev enp255.

These the assignment of an IP an Interface is added successfully.

c) [root@server n] ip address del 192.168.1.254 124 dev.

Thus the deletion of an IP on an interface is deleted successfully.

d) [root@server n] ip link set up

The status of an Interface is altered by enabling promise online.

e) [root@server n] # ip link set down.

The status of an interface is allowed by bringing the interface offline.

f) [root@server n] # ip link set promisc on

The status of an interface is allowed by bring the interface offline.

g) [root@user n] # ip route add default via 192.168.1.254 dev.

A default route is added for all addresses via the local gateway 192.168.1.254 sthan can be reached on the device used.

h) [root@Server n] # ip route add 192.168.1.0124 via 192.168.1254

A route is added to 192.168.1.0124 via the gateway at 192.168.1.254.

i) [root@server] # route add 192.168.1.0124 dev

A route is added to 192.168.1.0124 that can be reached on the device used.

j) [root@server n] #ip aroute del 192.168.1.0124 via 192.168.1.254.

The route is deleted for the route 192.168.1.0.0124 the gateway at 192.168.1.254.

k) [root@server n] # ip route get 10.10.1.4.

10.10.1.4 via 172.16.8.1 dev eth250

2) ip config.

enp2s0: flags = 4163 <UP, BROADCAST,
RUNNING MULTICAS > mtu 1500
inet 172.16.8.98 netmask 255.255.252.0
broadcast 172.16.11.255
inet6 fe80::b1bb:3129:5126:6676
prefixlen 64 scopeid 0x20<link>
ether 50:9a:4c:34:d4:cc txqueuelen 1000
(Ethernet)

RX packets 331201 bytes 206265688 (196.7 MiB)
RX errors 0 dropped 72 overruns 0 frame 0
TX packets 05914 bytes 4452470 (4.2 MiB)

3) mtr

a) mtr google.com

localhost. localDomain (0.0.0.0)
key: Helps: display mode restart order
of field c init.

Host

1) 172.16.8.1
2) static.ic - 41.229.429.49 - tatatd.co.in
3) 14250.171.162

Packets.

Loss %	Snt	Last	Pings Avg	Best	Wrst	StdDev
0.0%	163	0.2	0.2	0.1	1.1	0.0

b) mtr -b google.com

Local host. local admin keys Help Display
mode Restart order of fields quit. Host-

1. 172.16.8.1
2) static 41.229.429.49 - static.com
3) 142.250.171.102

	Packets			Pings		
Loss%	Snt	Last	Avg	Best	Wrst	StDev
0.0%	10	0.2	0.2	0.1	0.1	0.0

d) mtr -c google.com

Local host localdomain

keyf help displayed restart statistic of just Host

1) 172.16.18.1

2) 142.250.171.102

4) tcp dump.

last Meta data expiration check: 1:26:24 ago on Tue 23 Jul 2024 08:13:55 AM EST

Package tcpdump: 14:4.9:0-2.fc28.i686 is already installed, skipping Dependencies resolved Nothing to do. Complete!

Result:

Thus the study of various Networks connections used in Unix and windows is done executed successfully.

Study of types of cable

:: 2

27/7/24

TX.

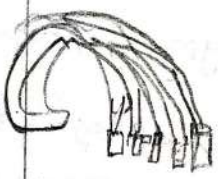
Study of different type of network cables

a) understand different types of network cables.

different type of cables used in networking are:

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

cable type	category	Maximum Data transmission	Advantages / Disadvantages	Applications	Image
UTP	category 3	10 bps upto	Advantages * cheaper in cost. * Easy to install as they have a small overall diameter. Disadvantages. * More prone to (EMI) Electromagnetic interference noise.	10 Box Ethernet	
	category 5	100 Mbps		fast Ethernet, Gigabit Ethernet	
	category 5e	1Gbps		Fast Ethernet gigabit Ethernet	
STP	category 6, 6a	10 Gbps	Advantages: * Shielded * Faster than UTP * less susceptible to noise and interference.	Gigabit Ethernet 10G Ethernet (55m) widely used in data centers	
SSTP	Category	10 Gbps	Disadvantage * Expensive * Greater Installation effort	Gigabit Ethernet 10G Ethernet (100m)	

Coaxial cable	RG-6 RG-59 RG-11	10 - 100 Mbps	Advantages: * High bandwidth * Immune to interference * Low loss bandwidth * Less tail Disadvantage * Limited * distance * cost * size is bulky		
Fibre optics cable.	Single mode multi mode	100 Gbps	Advantage * High speed * High bandwidth * High security * Long security Disadvantage * Expensive * Requires skilled installers	Maximum distance of fibre optics cable is around 100 meters	

b) Make your own Ethernet cross-over cable/ straight cable

Tools and parts needed

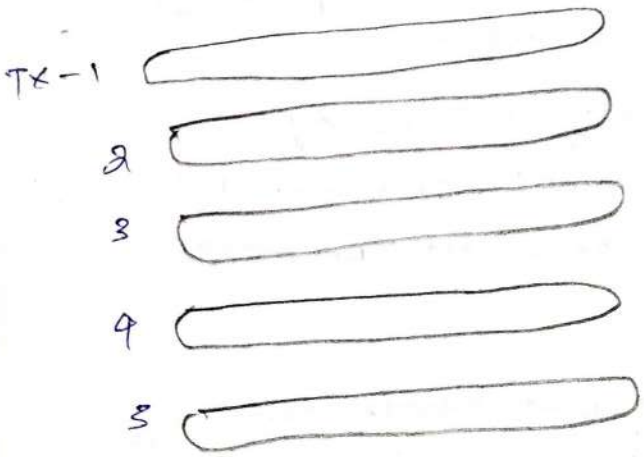
* Ethernet cabling CAT5e is certified for gigabit support but CAT5 cabling work as well, just over shorter distance.

* A crimping tool. This is an all in one networking tool shaped to push down the pins in the plug and strip and cut the shielding off the cables.

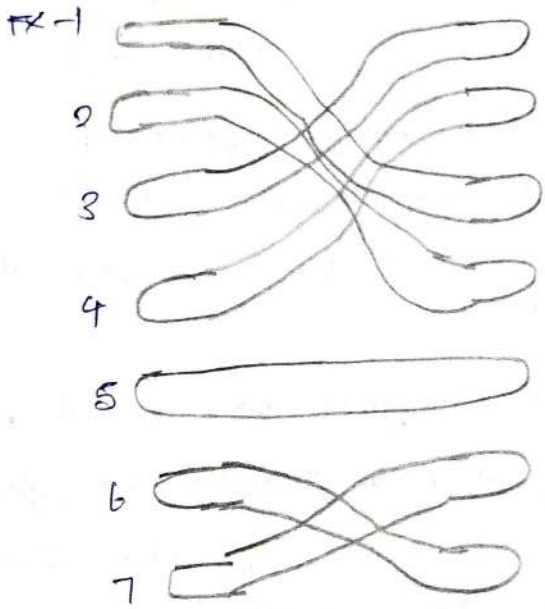
*. Two RJ45 plugs

* optional two plug shields.

Straight thelf cable.

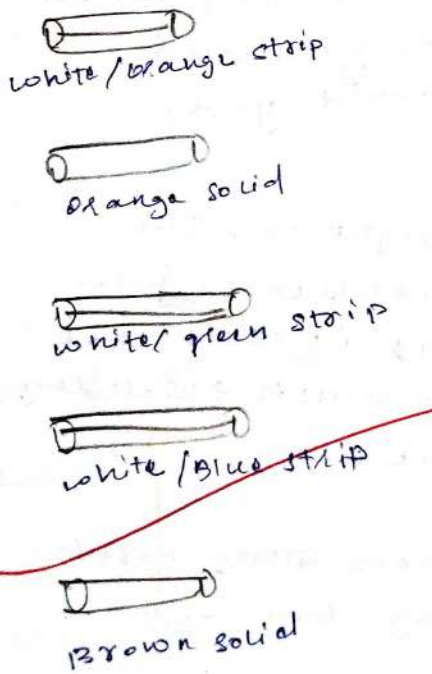


X-over cable.

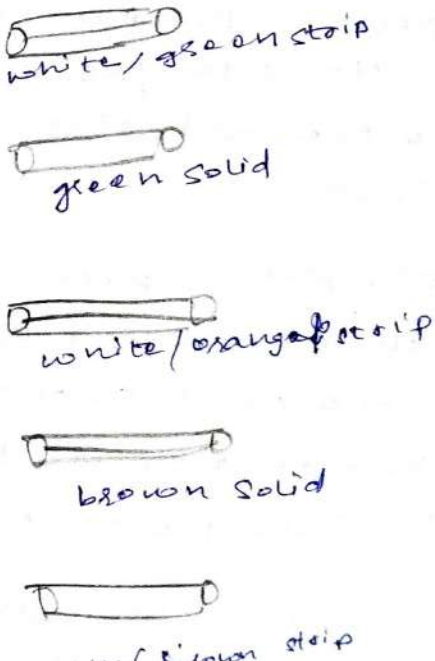


Difference b/w crossover cable and straight cable straight through network cable. Both side should be A cross over cable: one side A, one side B.

A



B



Step1: To start construction of the device, begin by threading sheilds onto the cable.

Step2: Next, strip approximately 1.5cm of cable shield from both the ends. The crimping tool has a round area to complete this task.

step3: 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 other in B.

step4: once the order is correct, bunch them together in a wire, and if there are any that stick out farther than others, snip them back to create an even level. The difficult aspect is placing these into RJ45 plug with the clip side facing away from you and have the gold pins facing toward you, as shown.

step5: 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 cable back a little more.

step6: After the wires are securely sitting inside the plug, insert it into the crimping tool and push down.

Step7: Lastly, repeat for the other end using diagram (B) using diagram (A)

Result:

Thus study of various type of cable is executed and verified

EXP. NO: 3 study of packet tracker tool

DATE: 30/7/24.

Aim:

To study the packet tracer tool Installation and user Interface.

To Understand Environmental of CISCO PACKET TRACER to design simple network.

Introduction:

A simulator, as the name suggest, simulates network device and it Environment.

1. It allows you to model complex system without the need for dedicated equipment

2. It helps you to pratice your Network configuration and troubleshooting skills via computer

3. It is avilable for both the linux and window desktop Environment.

4. protocols in packet Tracker are coded to work and behave in the same way as they would on real hardware.

Installing packet Tracker:

To download packet tracer, go to http://www.netacad.com and log in with your cisco networking Academy credentials;

Windows:

Instalcation in windows is pretty simple and straight forward; the setup comes in a single file named. to begin the setup wizard, accept the license agreement, choose a

LINUX:

Linux users with ubuntu / Debian distribution should download the file for fedora.
for ubuntu, and those using Fedora / Redhat / centos must download the file for Fedora.
Grant executable permission chmod +x packet tracer601_i386-installer-8pm.bin

./packetTracer601-i386_installer-8pm.bin

User interface overview:

The layout of packet tracer is divided into several compound. The components of packet tracer interface are follows:

1. Menu bar - This is a common menu in all software application.

2. Main toolbar - This bar provides shortcut icons to menu open, save, zoom, undo & redo and on right hand side.

3. Logical/physical workspace - The tabs allow you to toggle b/w the logical & physical area.

4. Workspace - This is the area where topologies are created & simulations are displayed.

5. Common tools bar - This toolbar provides for manipulation topologies, such as move layout.

6. Real time/Simulation tabs - these tabs are used to toggle b/w real & simulation.

7. Network component box - This component contains all of network & device available with packet tracks and further divided into two areas. Area Device - Specific selection box - when a device is selected within that category.

8. User created packet box: user can create highly customized packet to test their topology from this area and displayed.

d) Analyse the behaviour of network device using cisco packet.

1) From the network component box, click & drag & drop below component.

a) 4 generic pc is & HUB

b) 4 generic pc's & one switch

2) Click on connections.

a) Click on copper straight - through cable

b) Select one of the pc's & connect it to HUB using the Pc's & connect it to should glow in green indicating that

c) Click a HUB

3) Click on the Pcs command to hub, go to the tab, click on IP config, & enter IP address a subnet mask, then the default gateway & DNS server on-needed as these are only two end devices network

a) Drag & drop it one of PC (source machine) & then drop it connected to HUB

4) Observe the flow of PDU from source Pc to destination Pc by selecting the realtime mode of simulation.

5) Repeat the step (3) and step (5) to pc's

6) observe how HUB of switch forward & write observation & conclusion of switch & HUB.

Result:

Thus study of packet tracer tool installation face is done & executed successfully.

Azoln

28

Practical - 1

Student observation.

1. which command is used to find the reachability of a host machine from your device?

command to find the reachability of a host machine from your device.

- ping <hostname or IP address>

2. which command will be give the details of hops taken by a packet to reach its destination?

command to get details of hops taken by a packet to reach its destination

- traceroute <hostname or IP address> (Linux / unix)

3. which command displays the TCP port status in your machine

- netstat -tuln
- ss -tuln

4. which command display the IP configuration of your machine?

- ifconfig (Linux/unix, older version)
- ip addr (Linux/unix newer version)
- ipconfig (windows)

5. write the modify ip configuration in a Linux machine.

Temprorary changes.

using ifconfig (old version)

using ip (modern method)

sudo ip addr add 192.168.1.10/24 dev eth0

sudo ip route add default via 192.168.1.1

Practical - 2.

Student observation:

(1) What is the difference between cross cable and straight cable?

Straight cable:

- All the wires are in same order on both ends of the cable.
- used to connect different types of device

cross cable:

- The transmit and recive wires are swapped on 2nd end.
- used to connect similar device directly.

(2) which type of cable is connect a router or switch to your pc?

- used to connect a pc to a router or switch.

(3) which type of cable is used to connect two pc cross cable.

- used for direct pc to pc connections without a hub, switch or router.

(4) Find out the category of pair cable used in your lab to connect the pc to the network socket

category 6: Support up to 10GBPS short distance

(5) write down your understanding, challenges faced and output received while making a twisted pair cross/straight cable.

understandings.

straight cable:

→ used to connect diff. types of Network device

→ Both ends have the same wiring order.

cross cable

→ used to connect similar types of Network directly

→ one end follows TIA/EIA-568-B and the other follows TIA/EIA-568-A standard

Challenges faced.

- Crimping issues
- wire order
- testing

output received.

- successfully made cable

Practical - 3.

Student observation:

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

HUB:

Broadcasting:

→ A hub is basic networking device that operates at the the physical layer of the OSI model.

→ when hub receives a packet on one of its ports it broadcasts the packet to all other ports,

→ This behaviour results in all devices connected to the HUB receiving the packets, even if they are not intended recipient.

Collission Domain:

→ All devices connected to HUB are in same collision domain.

SWITCH:

· Intelligent forwarding:

→ A switch operates ~~receives~~ at the data link layer of OSI model.

→ when a switch receives a packet it examines the MAC address of the destination device

→ It forwards the packet only to the port associated the destination MAC address, thus reducing unnecessary traffic and improving Network efficiency

Collision Domain:

→ Each port on a switch represent a separate collision Domain

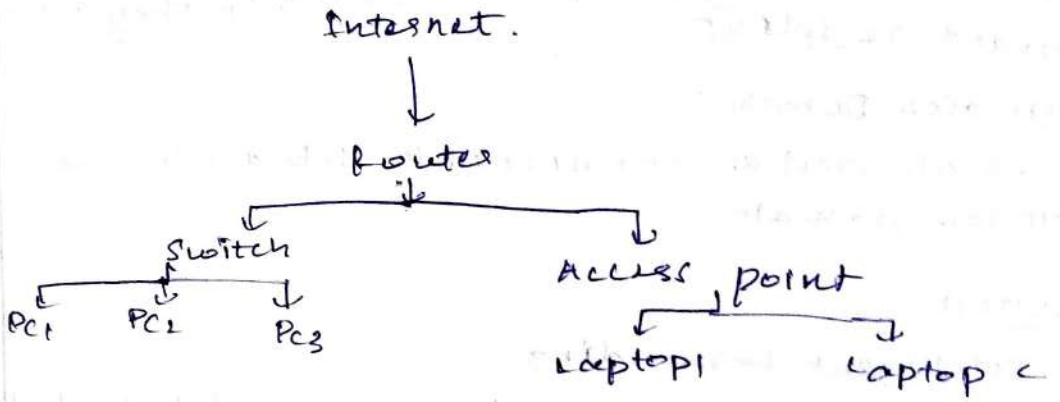
→ This Insolation significaty reduces the likelihood of collision compared to a hub. A packet is only forwarded to the Intended recipient

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

- Star topology

→ All devices are connected to a central switch or hub.

→ This is one of the most common and widely used topologies and efficiency



EX.NO:4
Date: 11/8/24

Aim:
setup and configure a LAN (Local Area Network) using a switch and ethernet cables in your lab.

What is LAN?
A Local Area Network 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.

How to set up LAN.
① → plan & Design an appropriate network topology taking into account network requirement and equipment location.

②-Take 4 computers, a switch 8, 16, or 24 ports which is sufficient for network of this size and 4 ethernet cables.

③- connect your computer to network switch via an Ethernet cable.

④ - Assign IP Address to your PCs
↳ log on to the client computers as admin
↳ click Network and Internet connections
↳ Right click local Area Connection Ethernet → Go to properties → select Internet protocol (TCP/IPv4)
↳ click on properties → select us the following ip address option and assign ip address

⑤- Configuration Network switch
↳ connect your computer to the switch to switch web Interface, you will need to connect your computer to the switch using as ethernet cable.
→ Log in to the web - Interface
→ configure basic settings
→ Assign IP Address as 10.1.1.5;

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

① - Select a folder → go to properties → click sharing lab → share it with everyone on the same lan.

② Try to access the shared folder from other computers of the network.

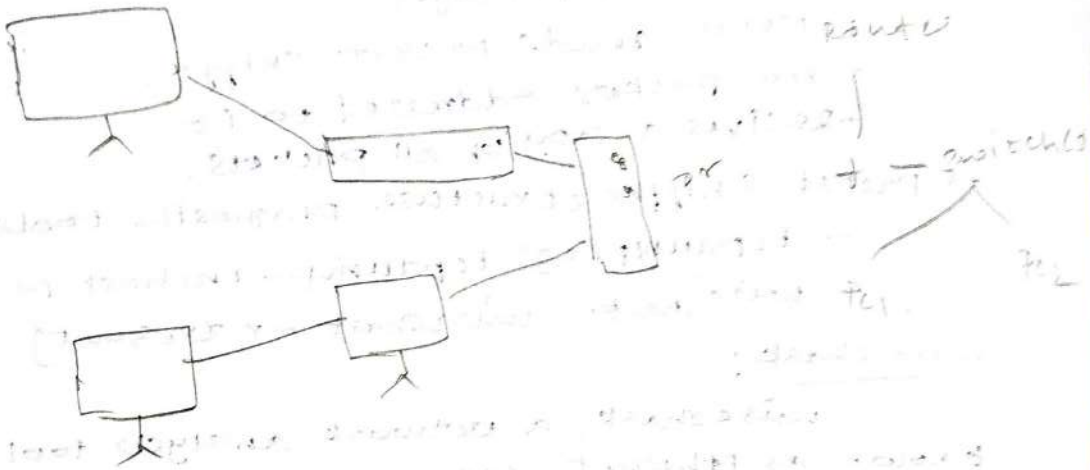
Result:

The experiment for setup and configuring a LAN been verified.

Student observation:

Draw a neat Diagram of LAN in the configuration observation book, that you have Implemented in your lab. write the IP config of each & every device with the outcome & challenge faced with configuration the LAN

write IP configuration of each & every device.



outcome:

LAN was successfully setup and all devices would communicate with each other using their assigned IP Address. Shared resources like folders were accessible.

challenges faced.

→ Measuring each pc has a unique Ip Address to avoid conflicts.

→ Initial difficulty accessing the switch web due to incorrect IP address.

Ex.No: 5 Packet capturing tool: wireshark
Date: 7/8/24.

Aim.

Experiments on packet capture tool - wireshark

Packet sniffer

→ sniffs messages being sent / received from/ by your computer.

→ store and display the contents on the various protocal fields in manages.

- nerves reads packet sniffs.
- no packets addressed to it
- Recives a copy of all packets.

→ Packet sniffer structure Diagnostic tools

- tcpdump (eg: tcpdump -nr host 10.129.41.2)
- wireshark (wireshark -r 2x23out)

wireshark:

wireshark, a Network analysis tool formally known as Ethereal, capture packet in real time and display them in human readable format.

→ what we can do — capture Network traffic
decode packets protocols using dissectors.
- Analyse problems.

→ used for - people learn network protocol.
Network administrators: troubleshoot Network problems.

Getting wireshark: wireshark can be download for windows or macos from the official website.

capturing Packets: After Downloading and installing wireshark launch it and double click capture to start capturing packets.

The "packet Details" Panel: shows all packets in current capture.

The "packet details" panel: shows the current packet in a more detailed

The "packet Bytes" Panel: shows

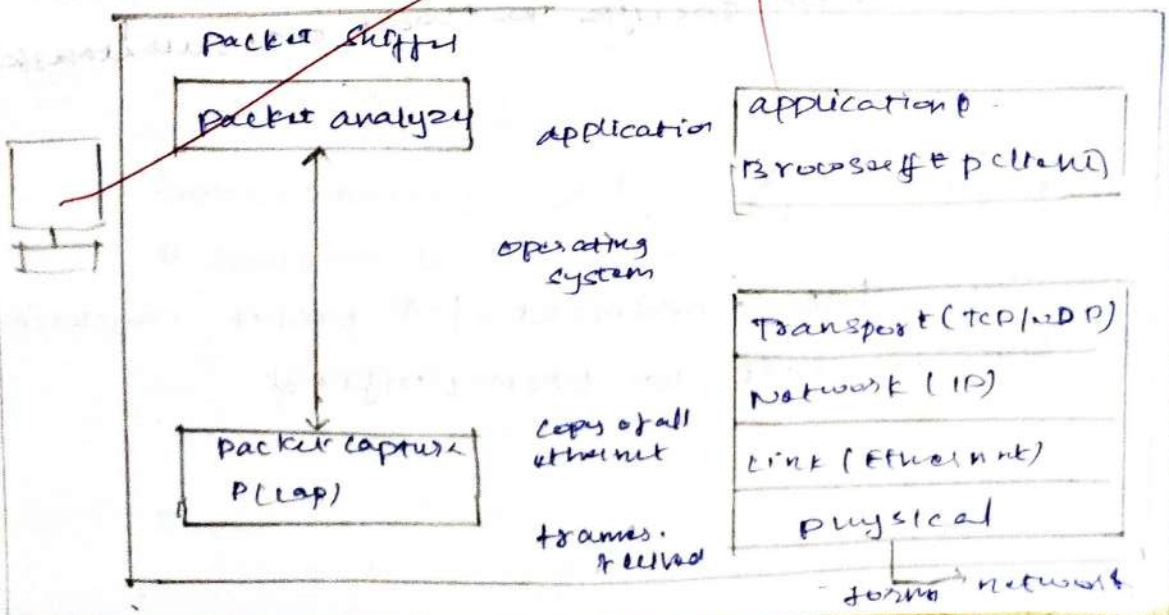
colouring, small capture, Filtering packet. Inspecting packets, Flow graph; gives a better understanding of what we see.

Capturing and Analysing packets using wireshark tools.

Procedure:

→ select LAN in 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~~ inspect the packets and provide the flow graph
2. create a filter to Display only ARP packet and Inspect packets.
3. create a filter to Display only DNS packet and provide a flow graph
4. creat a filter to display only HTTP packets and Inspect the packets.
5. create a Filter to display only IP/ICMP packet and inspect the packets.
6. create filter to display only DHCP and inspect the packets.



student observation.

① what is promiscuous mode? ~~is a configuration~~

Promiscuous mode is a configuration for a Network interface that allows its to capture all packets on the Network segment it is connected to, Not Just the packets

② Does ARP packets have a 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 is used to map IP addresses to MAC address.

③ which transport layer protocol is used by DNS

DNS primary uses UDP as its transport layer protocol but it also uses TCP for larger responses.

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 Default port in 443.

⑤ what is a Broadcast IP Address?

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

Result!

Thus the experiment for packet capturing tool, wireshark has been verified

Ex: no: 6
Date: 12/9/2024

Hamming code

Aim:
write a program to implement error detection & correction using hamming code concept make a test run to input data stream & verify correction

→ Error Detection & Data link layer:
Hamming code is a set of error-correction codes that can be used to detect & correct & errors that occur when data is transmitted from sender

→ Create sender program with below features.
- → Input → text to any length
- → convert to binary
- → apply hamming code concept on binary data & add redundant bits to it

→ Create reciver program with below feature
- → should read input from channel file
- → Apply hamming code the binary data to check the errors
- → else remove redundent bits &

Code

```
convert binary data to ascii
import numpy as np
# function to convert text to binary
def text_to_binary(text):
    return ''.join(format(ord(char), '08b')
                   for char in text)

# function to calculate redundant needed
for error correction.
```

```
def binary_to_text(binary):
    chas = [binary[i:i+8] for i in range
            (0, len(binary), 8)]
# function to insert redendant bits int the data
    def pos_redundant_bits(data_s)
        j = 0
        k = 0
        m = len(data)
        res = ""
# adding redendant bits at position that are
  process of 2
        for i in range (1, m, +r, +1)
            if (i == 2**j):
                res = res + 0
                j += 1
        return res
# function do calculate parity bits
        def detect_and_correct(encoded_bits):
            r=0
            while (2**r) <= len(encoded_bits)
            r += 1
            error_pos = 0
            for i in range (r)
            Parity_pos = 2**i
            Parity=0
            for j in range(parity_pos-1, len(encoded
                                    Parity_pos)
            for k in range(parity_pos):
                if j+k < len(encoded_bits)
                Parity ^= int(encoded_bits[j+k])
            if parity != 0:
            return error_pos error_pos += parity_pos.
```

```
def main()
    input_str = input("Enter the string to encode:")
    data_bits = string_to_bits(input_str)
    print("original data bits: {}".format(data_bits))

    r = calculate_redundant_bits(len(data_bits))
    print("number of redundant bits needed = {}".format(r))
    encoded_bits = insert_redundant_bits(data_bits, r)
    print("Encoded bits with redundant bits: {}".format(encoded_bits))

    user_input = input("Enter the bit position (1-based index) to change (or 0 to skip): ")

    if user_input != '0':
        bit_position = int(user_input) - 1
        if 0 <= bit_position < len(encoded_bits):
            encoded_bits_list = list(encoded_bits)
            encoded_bits_list[bit_position] = '1' if encoded_bits_list[bit_position] == '0' else '0'
            encoded_bits = ''.join(encoded_bits_list)
            print("Bit at position {} changed. Updated encoded bits: {}".format(bit_position+1, encoded_bits))
        else:
            print("Invalid bit position entered. No change made.")

    detected_error_pos = detect_and_correct(encoded_bits)
    if detected_error_pos:
        print("Detected error at position: {}".format(detected_error_pos))
        encoded_bits_list = list(encoded_bits)
        encoded_bits_list[detected_error_pos-1] = '1'
```

```
if encoded_bits_list[detected_error_pos-1] == '0'
    else '0'
encoded_bits = ''.join(encoded_bits_list)
    print("corrected encoded bits = {}".format
                                (encoded_bits))
else:
    print("No error detected.")
corrected_data_bits = [encoded_bits[i] for i
    in range(len(encoded_bits)) if (i+1)&(i)!=0]
corrected_data_bits = ''.join(corrected_data_bits)
print("corrected data bits: {}".format(corrected_
                                data_bits))
    print("original message: {}".format(original_
                                message))
if __name__ == "__main__":
    main()
```

Result:

This the framing code program has been implemented successfully.

Ex.No: 7
7/10/2024

Sliding window

Aim:

write a program to implement flow control of data Link layer using SLIDING WINDOW PROTOCOL. Simulate the flow of frames from one node to Another.

Create a sender program with following features.

1. Input window size from user
2. Input a text message from user.
3. Consider 1 character per frame.
4. Create a frame with following fields.
5. Send the frames.
6. Wait for the Acknowledgement from the reciver.
7. Reader a file called Recives - Buffer.
8. check Ack field for Acknowledgement from the ~~receiver~~ number.
9. If the Acknowledement number is as expected, send new set of frames accordingly, else if Nack is recived, resend the frames accordingly.

Create a Receiver file with following feature.

1. Read a file called Sender - Buffer
2. check the frame No
3. If the frame No. are as expected, write the appropriate Ack No. in the Recives - Buffer file Else write NACK No. in the Recives - Buffer file.

Student observation:

```
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]
                                    acknowledged.
                print("sent Frame {frames[i].frame_
                        no}: {frames[i] data}")
            print("Frames sent, waiting for
                                    Acknowledgements")
    def receive_frames(frames, window size):
        print("\n-- Receiving Frames --")
        for i in range(window_size):
            if i < len(frames) and not frames[i]
                if random.random() < 0.2
                    print(f"Recived frame {frames[i].frame
                            {frames[i]:data} {ERROR}")  no}
                    frames[i].acknowledged = False
                else
                    print(f"Recived frame {frames[i].frame
                            {frames[i].data}. {ok}")
                    frames[i].acknowledged = True
    def sliding_window_protocol:
        window_size = int(input("Enter window size:"))
        message = input("Enter message to send!")
        frames = [frames(i, message[i]) for i in
```

```
base=0
while base < len(frames):
    send_frames(frames[base], window_size)
    time.sleep(2)
    receive_frames(base:), window_size)
    Acknowledged.
        base += 1
    if base < len(frames)
    print("\n Resending acknowledged frames", time.
                                        sleep(2))
    print("\n All frames sent & unacknowledged).
if ~~frame~~ __name__ == "__main__":
        sliding_window_protocol()
```

output:

```
Enter window size: 5
Enter message to sent: ACTHWI-
    - Sending frames

    Sent frame 0: A
    Sent frame 1: S
    Sent frame 2: H
    Sent frame 3: W
    Sent frame 4: I
frames sent, waiting for acknowledgement
    - Recving frames -
Received frame 0: A (Received)
Received frame 1: S (Received)
Received frame 2: H (Received)
Received frame 3: W (Received)
Received frame 4: I (Received)

    Resending frame
Received frame: 0: A (Received
All the frames are sent & acknowledged.
```

Ex. No: 8
21/10/21

A

Pac

P

Result:

Thus the code for flow control (sliding protocol) is executed successfully.

EX. NO: 8
2/10/21

VIRTUAL LAN

Aim:

Simulate Virtual LAN config. using Cisco Packet Tracer Simulation

Post1:

Build the network & config basic device setting

1. Network Setup

* Add two switches S1 & S2 along with PC (PC-A & PC-B) as shown in the topology

* connect devices using copper straight through cables for network connection & console for console access

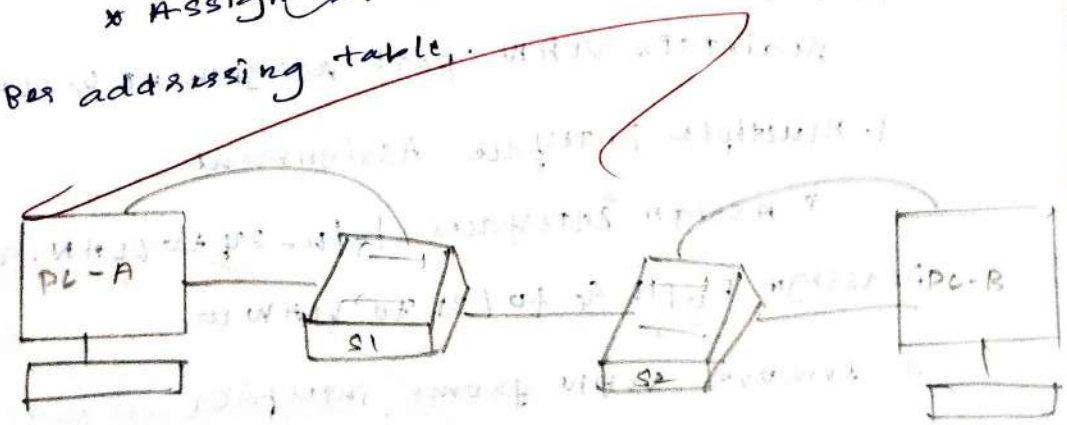
2. Basic config on switches (S1 & S2)

* Access each switch via PC terminal the
* Enter config. mode & set device config.

(CLI mode, console, & vty ethernet)

3. PC Configuration

* Assign IP address to PC-A & PC-B as per addressing table.



Device	Interface	IP address	Subnet	def gateway
S1	LAN1	192.168.1.11	255.255.255.0	N/A
S2	LAN2	192.168.1.12	255.255.255.0	N/A
Pc-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
Pc-B	NIC	192.168.10.4	255.255.255.0	192.168.10.1

4. testing connectivity

* ping blw PC-A, PC-B, S1, & S2, verify the responses to check network connectivity.

Part-2

Create VLAN's & Assign switch port

1. VLAN creation

* Create VLAN's in S1 & S2 using the VLAN commands & assign names.

2. Assign port to VLANS

* Assign Pc-A to VLAN 10 on S1 & set the management IP on VLAN 99 Remove VLAN1 IP

* verify VLAN assignment using show VLAN brief & show IP interface brief.

Part-3

Maintain VLAN port Assignment & VLAN DB

1. Multiple interface Assignment

* Assign interfaces fa0/11-24 to VLAN 99 on S1 Reassign Fa0/11 & fa0/21 to VLAN 10

2. Remove VLAN from interface

* Remove VLAN 99 from fa0/24, then verify the default VLAN is reassigned

3. VLAN removal

* Create VLAN 50 directly on fa0/24, then delete VLAN 50 from database & observe what happens.

Part-4

Configure on fo2. I@ trunk

Blw the switches.

1. DTP tranlang

* Configure fo1, on S1 to initiate trunking using dynamic trunking protocal (DTP) with command switch port mode

* check the states changes & verify on S2

2. manual Trunk configure

* Disable DTP on Po1, & set it to

* manual trunk mode

S1 (config) # interface fo11

S1 (config) # switchput mode trunk

* change native VLAN to 1000

S1 (config) # interface fo11

S9 (config-if) # switch port trunk native vlan 1000

Result:

Thus the virtual LAN has been executed successfully.

Ex: 8(b)
2/10/21

~~Virtual LAN~~ Wireless LAN

Aim:

Simulate virtual lan config. using cisco Packet tracer simulation

Aim:

configuration of wireless LAN using cisco Packet tracer:

Design a topology with this is connected from wireless router

Perform following configuration:

* configure static IP on Pc & wirelessrouter
* Set SSID to mother.
* Set IP address of router to 192.168.0.1, Pc0 to 192.168.0.2, Pc1 to 192.168.0.3 and PC2 to 192.168.0.4.
* Secure your network by configuring wap key on routes.
* connect pc by using WAP key on Routes
* connect Pc by using WAP key

To complete these tasks follow these steps by step instruction.

Step: 1 - click on wireless router,

* Select administration tab from to menu, Set username and password to admin and click on the save setting

b)

* Next click on wireless tab and set default SSID to MothersNetwork.
* Now select wireless security and change security mode to WEP
* Set Key1 to 0123456789.
* Again go in the end of page and click on Save setting
* Now we have completed all given task on wireless router. Now configure the static IP on all three pc
* Double click on Pc select Desktop tab click on IP configuration select static IP and set IP as given below

PC	IP	Subnetmask	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 it's its time to connect Pc's for from wireless routes. To do so click Pc select Desktop click on pc wireless
* click on connected tab and click on Refresh button.
* It will ask for WAP key fwrite 0123456789 and click connect
* It will connect you with wirelessrouter
* Repeat same process on PC1 and PC2

Student observation:

c) what is SSID of a wireless router?

* It is the unique name assigned to a wireless network, allowing devices to identify and connect to it

* It helps distinguish different network in the same area and can be customized by users to make their network more recognizable.

d) what is security key in wireless router?

* A security key is the password used to protect a Wi-Fi network from unauthorized access

* common types include WEP, WPA and WPA2 with WPA 2 being the most secure.

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

1) connect the access point (AP)

Plug in the AP and connect it to your computer via wi-fi or ethernet

2) Login to AP

open to AP a browser, type the PPS IP address (eg: 192.168.1.1) and log in with the username and password

3) set the SSID

Name your network Eg. Lab_WLAN.

4) set security.

* choose WPA2 - PASSONE for security

* set password. security123

5) save setting

* save the changes and restart the AP.

Result:

Thus the wireless LAN has been Executed successfully

EX: 9 Subnetting

Date: 9/10/24

AIM:

Implementation of SUBNETTING IS CISCO PACKET TRACKER simulation.

Steps:

1) Create a network topology

* open a cisco packet tracker
* click on new network symbol to create a black topology

2) Add devices.

* Add the following devices
  * 2 routers (R1, R2)
  * 2 switches (S1, S2)
  * 10 pcs (5 for each subnet)
* connect devices
  * uses 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: 127 (provides 8 subnets host addresses each

4) IP addressing scheme:

* Routers R1
  * Gigabit Ethernet 0/0: 192.168.1.1
  * Gigabit Ethernet 0/1: 192.168.2.1

switch S1

* Fast ethernet 0/0: 192.168.1.0/27

PC's

- PC1 192.168.2.11
- PC2 192.168.2.12
- PC3 192.168.3.13
- PC4 192.168.3.14
- PC5 192.168.3.15

Switch S2:

Fast ethernet 0/1 : 192.168.1.0/27

PC's

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

5) Configusating the devices

- Open CLI on Routes R1 and Router enable
- configuse terminal
- IP address 192.168.2. → 255.255.255.224

No shutdown

- Exit
- Switch configusation
  - open CLI on switch S1 & enter enable

Configuse terminal
Interface fastethernet 0/1
Interface mode access
switchport mode access
Exit:
Interface fast ethernet 0/2
switchport mode access
exit

- Pc configusation:
  - Right - click on each Pc and select config

Enter:

. IP address, subnet mask (255.255.255.240) default gateway (Router (ID))

b) Testing the Network:

* open the command prompt on each pc

* Test the ping command prompt on to check connectivity between pc is and the router ping 192.168.1.x (for Pc's in the first subnet)

ping 192.168.3.x (for pc's in the second subnet)

4) Conclusion:

If all pings are successful your Subnetting and network configuration in Cisco packet tracker is functioning correctly

Student observation:

1) write down your understanding of subnetting.

Subnetting is the pratice of dividing a lasger 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 securities: Isolates sinsitive data and control access.

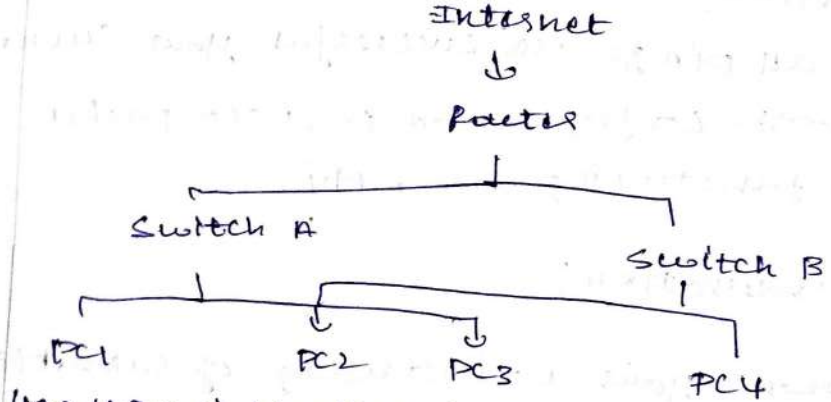
3) Find out whether subnetting is implemented in your college. If yes, draw and list down Subnets used with IP address.

1) main network

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

Subnet used

	Subnet name	Subnet Address	Subnet mask	IP Range	Max Hosts
1)	Academic department	192.168.0.0/24	255.255.255.0	192.168.0.1 - 192.168.0.255	254
2)	Administration	192.168.1.0/24	255.255.255.0	192.168.1.1 - 192.168.1.255	254
3)	Student Housing	192.168.2.0/24	255.255.255.0	192.168.2.1 - 192.168.2.255	254
4.	Library Services.	192.168.3.0/24	255.255.255.0	192.168.3.1 - 192.168.3.255	254



Printers Servers Student 1 Student 2

(192.168.0.1) (192.168.0.20) (192.168.2.2) (192.168.2.3)

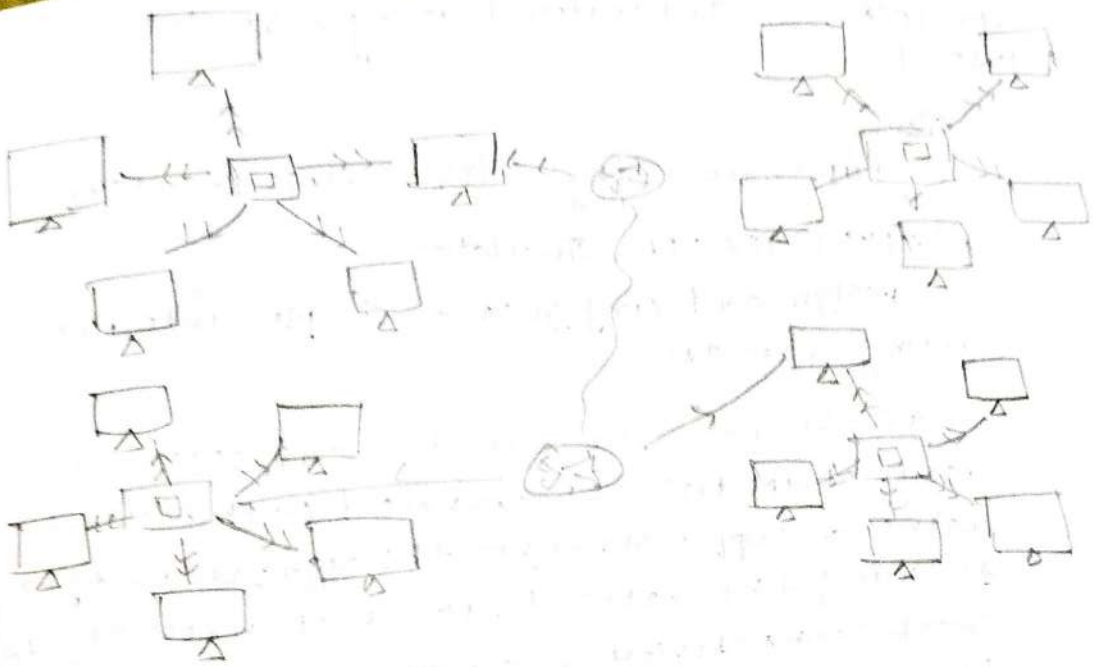
Subnets:

192.168.0.0/24 → Academic department

192.168.1.0/24 → Administration

192.168.2.0/24 → Student Housing

	Host
255	254
255.	254
255	254
55	254



Result:

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

EX.No: 10(A) Internetwork using a router
22/10/2024

Aim:

Internetworking with routers in CISCO PACKET TRACER Simulator.

Design and configure a simple internetwork using a router.

In this network, a router and 2PCs are used. Computers are connected with routers using a copper straight-through cable - After forming the network, to check network, to ~~check~~ connectivity a simple 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)
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 PC1 C10 to the desktop and select IP configuration and assign an IP address, Default gateway, Subnet mask.
3) Assign the default gateway of PC0 as 192.168.10.1
4) Assign the default gateway of PC1 as 192.168.20.1

Step-3 (connecting PCs with Router)

1) connect FastEthernet0 port of PC0 with Fast Ethernet 0/0 of Router1 using a copper straight-through cable.

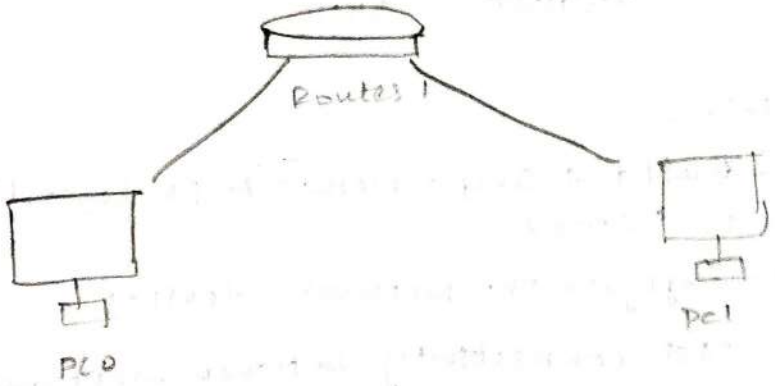
3) Connect FastEthernet 0 port of PC1 with FastEthernet 0/1 port of Router1 using a copper straight-through cable.

Routers Configuration Table.

Device name	IP address FastEthernet	Subnet mask	IP address FastEthernet 0/1	Subnet mask
Router1	192.168.10.1	255.255.255.0	192.168.20.1	255.255.255.0

Pc configuration table:

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



Ready

Result:

Thus the interconnecting with routers in cisco has be implemented successfully.

Ex.No: 10B Internetwork using wireless router
28/10/2024 DHCP servers and internet cloud

Aim:

Design and configure an internetwork using wireless routes, DHCP ~~Server~~ servers and internet cloud

Addressing table.

Device	Interface	IP address	Subnet mask	Default gateway
PC	Ethernet0	DHCP		
wireless router	LAN	192.168.0.1	255.255.255.0	
wireless routes	Internet	DHCP		
CISCO-lom server	Ethernet0	209.67.220.220	255.253 255.0	
Laptop	wireless0	DHCP		

objectives:

Part1: Build a simple Network in logical topology workspace

Part2: configure the Network devices

Part3: Test connectivity between Network devices

Part4: Save the file and close packet tracer.

Part1: Build a simple network in logical topology workspace.

Step1: Launch packet tracer

step2: Build the topology

Part2: configure the network devices

Step1

Configure the wireless router.

a) create the wireless network on the wireless routes change all the necessary setting.

b) save the settings

step2
configure the laptop

a) configure the laptop to access the wireless network

step3:
configure the PC

a) configure the PC for the wireless network

step4:
configure the internet cloud

a) Install - network modules if necessary
b) Identify the from and to ports
c) Identify the type of provides.

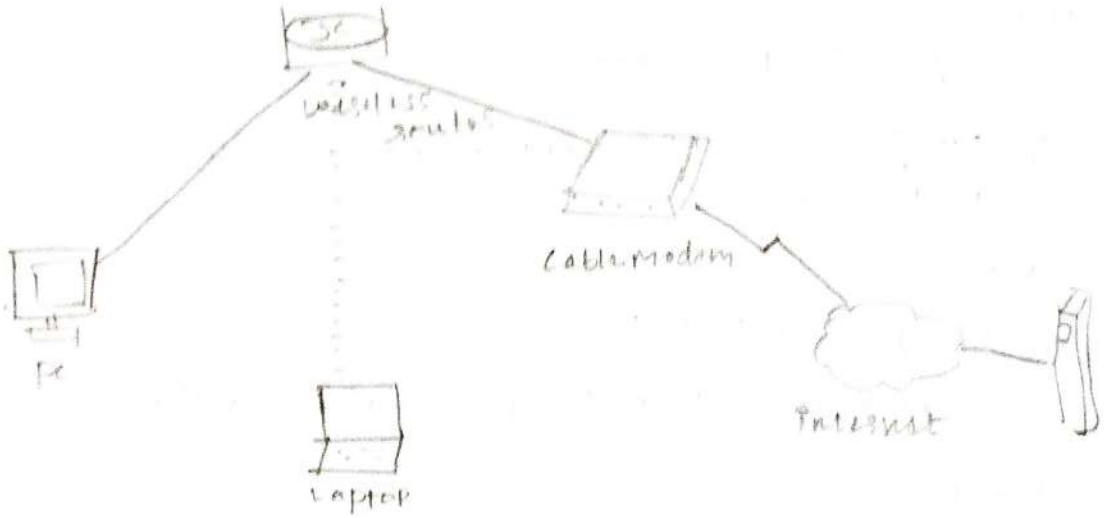
step 5:
configure the cisco.com servers.

a) configure the cisco.com servers as DHCP servers
b) configure the cisco.com servers as DNS server to provide domain name to IPv4 address resolution.
c) configure the cisco.com servers global settings.

part 3: Verify connectivity

step1.
Refresh the IPv6 settings on the PC

a) Verify that the PC is receiving IPv4 configuration from DHCP
b) Test connectivity to the cisco.com servers from the PC



student observation:

1) Key features of configuring wireless routers and DHCP server:

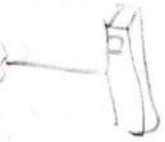
* wireless router configuration: set the SSID (network name), security type (WPA2 or WPA3) and encryption method to secure network access. Configure the router's IP address, subnet mask and default gateway to align with network's addressing scheme.

* DHCP server configuration: Enable IP DHCP server to automatically assign IP addresses to devices on network. ~~Set IP~~.

2) What is the significance of DHCP servers in internetworking.

* Automated IP assignment. DHCP simplifies IP address management by dynamically assigning IP addresses to devices. reducing the risk of IP conflicts and manual errors.

* Efficient resource allocation: It supports efficient allocation of network resources by managing configuration, such as subnet masks, gateways and DNS servers, for seamless device connectivity access different network segments.



Result:

Thus the internetworking using wireless routers, DHCP servers and internet cloud is implemented successfully.

Ex: 11[a]
11/11/2024

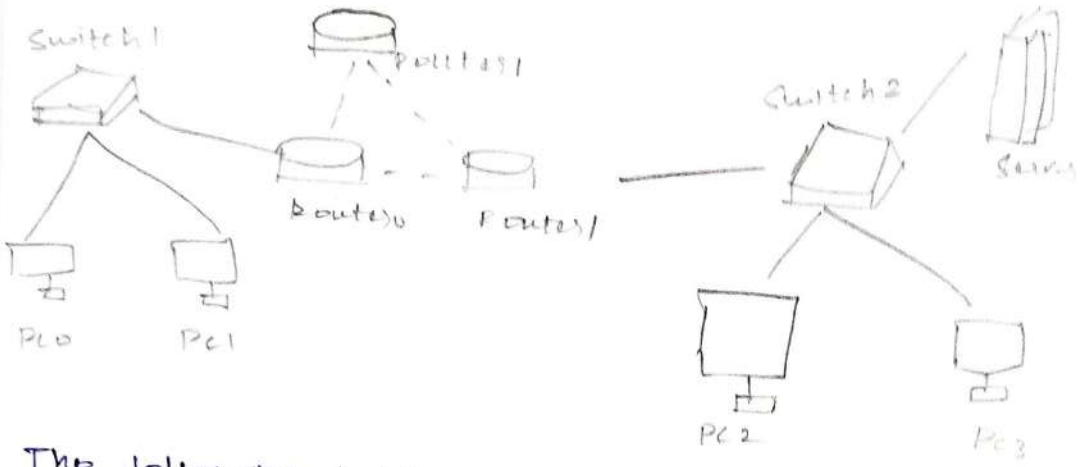
Static routing

Aim:

Simulate static routing configuration using cisco packet tracer.

Setting up a pratical lab.

Create a packet tracer lab as shown in following Image.



The following table lists the connected network of each routes.

Routes	Available network on local interfaces	Network available on other routers interfaces
Router0	10.0.0.0.18, 20.0.0.0.18 40.0.0.18	30.0.0.018. 50.0.00.18
Router1	20.0.0.018 20.0.0.018 50.0.0 -18	10.0.0.018. 20.0.0.018 20.0.0.018
Router2	40.0.0.0181 50.0.0.018	10 0.0.018 40.0.00.18

Router 0 configuration

enable

configure terminal

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

IP route 30.0.0.100 255.255.255.255 40.0.0.2 10

IP route 30.0.0.100 255.255.255.255 20.0.0.2 20

IP route 50.0.0.0 255.0.0.0 40.0.0.2 10

IP route 50.0.0.0 255.0.0.0 20.0.0.2 20

exit

Router 1 configuration

enable

configure terminal

IP route 10.0.0.0 255.0.0.0 20.0.0.1 10

IP route 10.0.0.0 255.0.0.0 50.0.0.1 20

IP route 40.0.0.0 255.0.0.0 20.0.0.1 10

IP route 40.0.0.0 255.0.0.0 50.0.0.1 20

Router 2 configuration

enable

configure terminal

ip route 10.0.0.0 255.0.0.0 40.0.0.1

IP route 30.0.0.0 255.0.0.0 50.0.0.2

Exit

Verifying static routing.

traceroute command sends ping requests to destination host and track the path they take to reach the destination.

Deleting a static route

* show ip route static command is used to print all static routes.

* Note down the route you want to delete

* Use the 'no ip route' command to delete the route.

Result:

Thus the static routing configuration has been executed successfully.

Ex: 11 (b) PIP using cisco packet tracer
11/11/2024

Aim:
simulate PIP using cisco packet tracer

Initial IP configuration

Device	Interface	IP configuration	Connected with
PC0	FastEthernet	10.0.0.2/8	Router 0's Fa0/1
Router 0	Fa0/1	10.0.0.1/8	Router 2's Gig0/1
Router 0	Gig0/1	192.168.1.254/30	PC0's fast Ethernet
Router 0	Se0/1/0	192.168.1.247/30	Router 1's Se0/1/0
Router 1	Se0/1/0	192.168.1.230/30	Router 1's Se0/1/0
Router 1	Se0/1/1	192.168.1.246/30	Router 0's Se0/1/0
Router 2	Se0/1/0	192.168.1.245/30	Router's Se0/1/0
Router 2	Se0/1/0	192.168.1.253/30	Router 1's Se0/1/1
Router 2	Fa0/0	20.0.0.1/30	PC1's Fast Ethernet
PC1	FastEthernet	20.0.0.2/30	Router 2's Fa0/1

Assign IP address to PCs

Double click PCs and click desktop menu item and click IP configuration. assign IP address referring the above table.

Assign IP address to interfaces of routers.

Assign the IP address for all the interfaces on the routers as the data given in the IP configuration table.

- set the clock rate for DCE and not for the DTE end.
- show controllers interface gives whether the interface is DCE or DTE.

Configure RIP routing protocol

Router 0

```
router rip
network 10.0.0.0
network 192.168.1.252
network 192.168.1.248
```

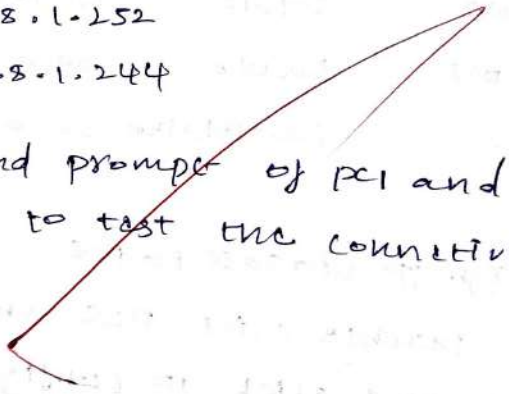
Router 1

```
router rip
network 192.168.1.244
network 192.168.1.248
```

Router 2

```
router rip
network 10.0.0.0
network 192.168.1.252
network 192.168.1.244
```

Access the command prompt of PC1 and use ping command to test the connectivity from PC0



Result:

Thus the simulation of RIP using cisco Packet tracer has been implemented successfully.

Ex: 1(a) Echo client server using TCP/UDP sockets

Aim

Implementing echo client server using TCP/UDP sockets

Algorithm:

1) TCP echo client server algorithm

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

TCP client Algorithm:

1) Create TCP socket
2) connect to server using specified address & port
3) Send message to server
4) Receive the echoed message from the server
5) Display received message
6) close the socket.

Program:

tcp_server.py

```
import socket
def tcp_server():
    server_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_STREAM)
    server_socket.bind(("localhost", 12345))
    server_socket.listen(1)
```

```
Print(f"connected to {client_address}")
try
  while True
    data = connection.recv(1035)
if data:
    print(f"Received: {data.decode()}")
  else:
    break
finally:
    connection.close()
if __name__ == "__main__"
    tcp_server()
tcp_client.py
    import socket
  def tcp_client():
    client_socket = socket.socket(socket.AF_INE
                                  socket.SOCK_STREAM
  client.socket.connect(("localhost", 12345))
try:
  message = input("Enter a message to send")
  client_socket.send all(message.encode())
  data = client.socket.recv(1033)
    Print(f"Received from server: {data.
                                  decode()}")
finally:
    client socket.close()
if __name__ == "__main__"
        tcp_client()
```

output:

Terminal: 1

TCP_server.py

PS C:\Education\Semus\CN\lab\python -u "c:\Education\CN\lab\EXP2\tcp_server.py"

TCP Server is waiting for a connection...

connected to ('127.0.0.1', 53356)

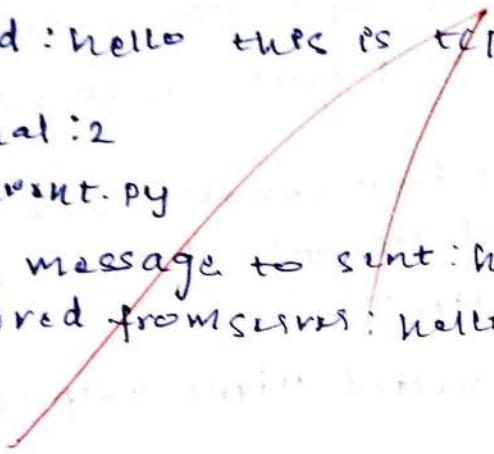
Received: hello this is tcp client

Terminal: 2

TCP_client.py

Enter message to sent: hello this is tcp client

Received from server: hello this is tcp client.



Result:

Thus the program to echo client server using TCP/UDP Socket is executed successfully.

Ex: 12(b) Chat program using socket programming
6/11/24

Aim:

To implement the chat client server using TCP/UDP sockets.

Algorithm:

Chat server.

1) Start the server by creating a socket, bind to specific address and port, listen for incoming connections.

2) When a new client connects add client a list of connected clients. Start a new process to talk to the clients.

3) For each connected client keep checking new message.

4) If a client disconnects, remove that client from the list & stop talking to that client

5) Keep running the process till the server stops.

Chat client.

1) Connect to server by creating a socket and connect it to server

2) Start a process to listen to message from the server.

3) Keep asking for new message

4) Keep running ctrl the user decided to quit.

chat-server.py.

```
import socket
import threading
def handle_client(client_socket):
```

```
while True:
    try:
        message = client_socket.recv(1033).decode
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 occured: {e}")
    break.
client_socket.close()
def start_server():
    server = socket.socket(socket.AF_INET, socket.
                                SOCK_STREAM)
server.bind(('127.0.0.1', 12345))
server.listen(5)
print("chatserver started on 127.0.0.1:12345")
client_handler = threading.Thread(target=
        handle_client, args=(client_socket,))
client_handler.start()

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

```
chat_client.py
import socket
import threading
def receive_messages(client_socket):
while True:
    try
        message = client_socket.recv(1033).decode(
                                'utf-8')
if message:
    print(f"Server: {message}")
except Exception as e:
    print(f"An error occured: {e}")
```

```
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 the chat server.")

    threading.Thread(target=receive_messages,
                     args=(client_socket)).daemon = True
                                          start()
while True:
    message = input("You:")
    client_socket.send(message.encode('utf-8'))

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

output:

Terminal : 1

CHAT_SERVER.py

```
Chat server started on 127.0.0.1:12345
New connection from ('127.0.0.1', 53680)
Received message from client: hellooo this is
Type your message to client: hello this is chat
```

Terminal : 2

CHAT_CLIENT.py

```
Connected to the chat server
You: hello this is chat client
You = Server = hellothis is chat server
```

Result:

Thus the program to implement chat program using socket programming is executed successfully.

Ex: 15
6/11/24

ping program

Aim:
implement your own ping program

Algorithm.

ping - client. py

1) socket creation
2) then set a timeout of 2 second to ensure if no response is received it will stop waiting and print "request time out"
3) send a 'ping' msg to specified host & port
4) It listens for a response & calculates the time disstance b/w sending and receiving the packet

ping. server. py

1) Initialise UDP socket
2) BIND to IP address & port
3) Listen for incoming message
4) Receive data
5) Send response.

Program;

ping-server. py

```
import socket
def start_server(host = '127.0.0.1', port = 12345)
    with socket. socket (AF-INET, socket-sock
        s. bind ((host, port))                DGRAM)
        print (f" UDP server running on {host}
                                    {port} ")
while true:
    data, addr = s. recvfrom (1023)
    print (f" Received message from {addr}"
            {data. decode()}") ; send to
                    (b' pong', addr)
if - name == "
    start-server()
```

ping client. py

```
import socket
import time
def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
as s:
    try:
        s.settimeout(2)
        start = time.time()
        s.sendto(b'ping', (host, port))
        data, addr = s.recvfrom(1024)
        end = time.time()
        print(f"Received {data.decode()} from
                {addr} in {end-start:.2f} seconds")
    except socket.timeout:
        print("Request timed out").
if __name__=="__main__":
    ping_server()
```

output.

Terminal1

```
ping_server.py
UDP Server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 51044):
                ping
```

Terminal 2:

```
ping_client.py
Received pong from ('127.0.0.1', 12345) in
                0.00 seconds
```

Result:

Thus the ping program has been executed successfully.

EX. NO:14 Packet Sniffing

Aim:

write a code using RAW sockets to implement packet sniffing.

Algorithm.

1) Install python and scapy

2) Create a program open text editor and create a file in notpad called packetsniffer & code to capture

3) Set up packet trackers by check if packet has IP layer, identify the packets protocol such as TCP, UPP, ICMP.

4) Capture Network packets

5) Run the packet sniffer by using command prompt.

6) Generate Network traffic by running the program.

Program:

```
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
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst
    Protocol name = ""
    if protocol name == 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__":
```

output:

```
Protocol : UDP
Source IP : 172.16.33.84
Destination IP : 244.0.0.251

Protocol : UDP
Source IP: 172.16.33.84
Destination IP: 244.0.0.251
Protocol : UDP
Source IP: 172.16.32.182
Destination IP: 8.8.8.8
```

Result:

Thus the packet sniffing program has been executed successfully.

Ex: 5
7/11/24

webalizes.

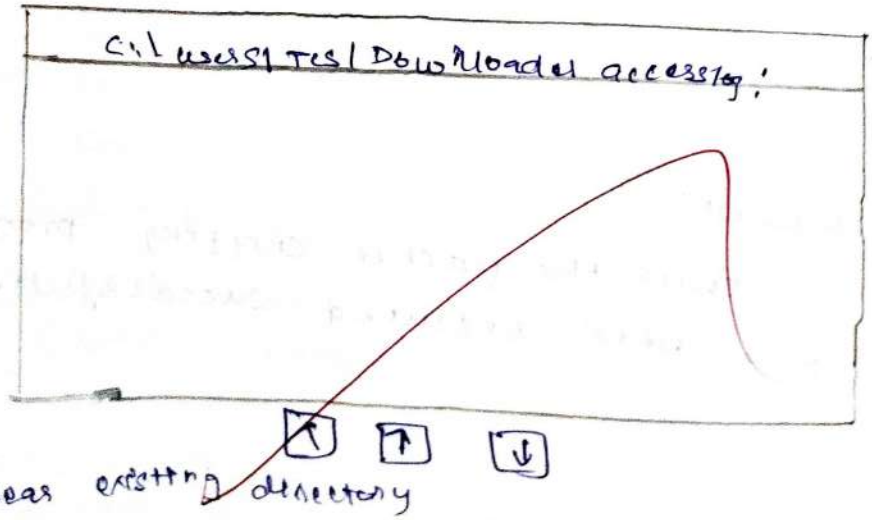
Aim:
To analyse the different types of weblogs using webalizes tool.

Algorithm:

* At first you must have installed xampp and hosted any website previously
* In the webalizes.config update the destination path to store the weblogs
* After updating it open command prompt
* Then open the xampp folder in the command prompt and execute the webalizes-exe file.
* the result is stored in the updated destination path
* check the various logs in the index.html file.

Logbilog	Logbible	View	Setting	Additional HTML cod	HTML

Input:
Logbilog



clear existing directory
Delete all files in selected Target Directory

Result:
Tw
differ
ps ex

Result:

Thus the procedure to analysis the different type of web logs using apache log is executed successfully.