INDEX

Name Agish Std CSE Sec A

Roll No 016 Subject Computer Networks School/College REC

S.No.	Date	Title	Page No.	Remarks
1	13/7/24	Study of various network command	8	
2	27/7/24	study of network cables	9	
3	30/7/24	EXP on cisco packet	10	
4	17/8/24	Setup and configure LAN using a switch and ethernet cable		
5	20/8/24	EXP on packet capture tool wireshark.		
6	27/8/24	Error detection and correction using hamming code		
7		Flow control using sliding window protocol		
8.		Virtual LAN configuration		
9.		Implement Subnetting		
10.		Internetworking with routers		
11.		static routing configuration		
12.		A) Echo client-server using TCP/UDP		
		B) chat client-server using TCP/UDP		
13		Implement ping program		
14		Implement sniffing in RAW sockets		
15		Analyse type of weblogs		

completed

20/11

Exp:1

AIM:

Study of various network commands used in linux and windows

Basic networking commands:

arp -a :- ARP is short form of address resolution protocol, it will show the IP address of your computer along with the IP address and MAC address of your router.

output:

Interface: 172.16.75.13 --- 0xf

Internet address	physical address	Type
172.16.72.1	7C-5a-1c-cf-be-41	dynamic
172.16.73.97	2a-3c-13-0a-cc-17	dynamic
172.16.75.7	4C-82-a9-77-ff-de	dynamic
172.16.75.14	4C-82-a9-77-ff-c1	dynamic
172.16.75.16	4C-82-a9-77-ff-f5	dynamic

hostname: This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

Output

Desktop-C01BH7D

ipconfig /all :- This command displays detailed configuration information about your TCP/IP connection including router, Gateway, Dns, DHCP and type of ethernet adapter in your system.

output:

```
Windows IP configuration
Hostname . . . . . : DESKTOP-C01BH7D
primary Dns suffix . . . :
Node type . . . . . : Hybrid
IP Routing enabled . . . : No
WINS proxy enabled . . . : No
```

Nbtstat -a: This command helps to solve and displays protocol statistics and current TCP/IP connections using NBT.

output:

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

-a (adapter status) Lists the remote machine name table given its name.

-A (adapter status) List the remote machine name table given its IP address.

-n (names) Lists local netBIOS names

nslookup: (name server lookup) is a tool used to perform DNS lookups in linux.

eg:- nslookup www.google.com

output:

Server: unknown
Address: 172.16.72.1
Non-authoritative answer:
Name: www.google.com
Address: 2404:6800:4007:812::2004

netstat: (network statistics) net stat display a variety of statistics about a computers active TCP/IP connections.
eg:- netstat -r

output:

Interface list
14...20 88 10 87 9f ab....Intel(R) Ethernet connection (17) I219-LM
9...4e 82 a9 77 ff ad.... Microsoft wi-fi direct virtual adapter #5
8...42 82 a9 77 ff ad.... Realtek RTL8852BE wifi 6802.11ax PCIe adapter

Pathping: pathping is unique to windows and is basically a combination of the ping and tracert commands.

output:

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

Ping (packet internet command is the best way to test connectivity between two nodes.

1. # ping hostname (ping localhost)
2. # ping ip address (ping 4.2.2.2)
3. # ping fully qualified domain name (ping www.facebook.com)

output:

options (# ping localhost):-
Pinging laptop-6VTH MA56 [::1] with 32 bytes of data
Reply from ::1: time 1ms
Reply from ::1: time 1ms

(# ping 4.2.2.2):-
Pinging 4.2.2.2 with 32 bytes of data
Reply from 4.2.2.2 bytes = 32 time=259ms
Reply from 4.2.2.2 bytes = 32 time=266ms
TTL=48

Route: The 'route' command in windows is used to display and manipulate the IP routing table.

output:

Manipulates network routing tables.
Route [-f] [-p] [-4|-6] command [destination] [mask netmask] [gateway] [metric metric] [IF interface]
-f clears the routing tables of all gateway entries. If this is used in conjuction with one of the command

the tables are cleared prior to running the command.

-H Force using IPv4

-6 Force using IPv6

Some important linux commands

Ip: The Ip command can show address information, manipulate routing, plus display network various devices. Interfaces and tunnels

ip <options> <object> <command>

a) To show the Ip addresses assigned to an interface on your server.

ip address show

enp2s0: <Broadcast, multicast, up, up>
mtu 1500 qdisc noqueue state
Down group default qun 1000.
link / ether 50:9a:4C:35:11:44 H:H:H:H:H:H
inet 172.16.8.107122 bnd 172.16.11.255
scope global enp2s0 valid

b) To assign an ip to an interface for example enp2s0

ip address add 192.168.1.254/24 dev enp2s0

c) To delete an ip on interface
ip address del 192.168.1.254/24 dev enp2s0

d) Alter the states of the interface by bringing the interface online.

ip link set enp2s0 up

e) Alter the status of the interface by bringing the interface offline.

ip link set enp2s0 down

f) Alter the status of the interface by enabling promiscuous mode for the interface.

ip link set enp250 promise on

g) Add a default route (for all addresses) via the local gateway 192.168.1.254 that can be reached on device enp250

ip route add default via 192.168.1.254 dev enp250

h) Add a route to 192.168.1.0/24 via the gateway at 192.168.1.254

ip route add 192.168.1.0/24 via 192.168.1.254

i) Add a route to 192.168.1.0/24 that can be reached on device enp250.

ip route add 192.168.1.0/24 dev enp250

j) Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254

ip route delete 192.168.1.0/24 via 192.168.1.254

k) Display the route taken for ip 10.10.1.4

- ip route get 10.10.1.4

O/P

10.10.1.4 via 192.168.1.254 dev ens160
cwd 0 cache.

2. ipconfig

The ipconfig command is a staple in many sysadmin's tool but for configuring and troubleshooting networks. It has since been replaced "ip".

output

Rx errors 0 drop 0 overrun 0 frames.

Tx packets 0 bytes 0 (0.0 B)

Tx errors 0 dropped 0 overruns 0 carries 0 collisions 0

3) mtr

mtr is traceroute is a command line tool for network diagnostics, combining the functionality of ping and traceroute It shows the route from a computer to a specified host and provides statistics like response time and packet loss for each hop.

mtr <options> hostname / ip

a) Basic mtr command that shows you the statistics # mtr google.com

localhost. local domain (0.0.0.0)

Host	Packets Loss %	Snt	Last	Avg	Pings Best	Wrst	Stdev
1. 172.168.1	0.0%	42	0.4	0.4	0.1	0.1	0.0
2. stcrc 41.229.249.49	0.0%	42	0.2	0.2	0.2	0.3	0.0
3. 142.250.171.162	0.0%	42	6.0	5.0	12	7.7	1.1

b) Show numeric ip addresses.

mtr -n google.com

No GTK support sorry

c) Show numeric ip address and hostnames

mtr -b google.com

localhost.localdomain (0.0.0.0)

Host	Packets Loss %	Snt	Last	Avg	Pings Best	Wrst	St dev
1. 172.16.8.1	0.0%	150	0.2	1.1	0.1	11.6	2.4
2. 142.250.171.112	0.0%	150	7.3	1.5	6.3	8.5	16.4
3. 142.251.227.217	1.4%	150	8.0	6.2	4.7	100.1	16.4

d) Set the number of pings

mtr -c 10 google.com

localhost.localdomain (0.0.0.0)

Host	Packets Loss %	Snt	Last	Avg	Pings Best	Wrst	St Dev
1. 172.16.8.1	0.0%	1	0.2	1.1	0.1	11.6	2.1
2. 142.250.171.162	0.0%	1	1.2	1.5	6.3	40.1	2.4
3. 142.251.227.217	44.1	1	2.4	6.2	4.7	50.2	6.4

4) tcpdump

It is designed for capturing and displaying packets

a. # dnf install -y tcpdump

Last metadata expiration check 0:02:19 ago on Sat 27 July 2024 12:07:01 PM IST.

Package

Dependencies resolved

Nothing to do. complete

b. $ tcpdump can use. You will need to use sudo or have root access in this case.

output:

ens160 [up, Running, connected]

any [psuedo device that all interfaces]

lo [up, Running, Loop back]

If you want to capture traffic on eth0, you want to capture traffic in eth0, you can initiate that with tcpdump -i eth0

tcpdump -i eth0

output:

dropped privs to tcpdump

tcpdump: verbose output, use -v [v]

link-type, Ethernet, Snapshot length 262144 bytes.

tcpdump -i eth0 -c 10

Capture traffic to and from one host you can filter out traffic coming from and going to 8.8.8.8 use the command

tcp dump -i eth0 -c 10 host 8.8.8.8

dropped privs to tcpdump

EN10MB (Ethernet) snapshot length 262144 bytes.

tcpdump i eth0 dst host 8.8.8.8

capture traffic to and from a network

dropped privs to tcpdump.

full protocol decode listening on

ens160, link-type EN10MB, snapshot.

5) ping:

ping is a tool that verifies IP connectivity by sending ICMP echo request messages with round trip times. reachability and name resolution.

Ping google.com

PING google.com (142.250.195.174) 56(84) bytes of data 64 bytes from maa03s41 in-f14.1e100.net icmp-seq=1 ttl=128 time=4.53 ms

64 bytes from maa35s41-in-f14.1e100 net icmp seq=2 ttl=128 time=9.67 ms

10 packets transmitted, 10 received, 0% packet loss, time 9016

rtt min/avg/max/mdev = 4.087/6.280/11.96/2.45 ms.

configuring an ethernet connection by using nmcli

if you connect a host to the network over ethernet, you using nmcli utility.

Procedure:

1) list the network manager connection profiles nmcli connection show

NAME	UUID	TYPE	DEVICE
wired connection1	5ec9a722-1def-35d4-5719-6692c0b2c9cbb	Ethernet	ens160

2) nmcli connection add con-name <connection-name> if name <device-name> type ethernet.

3) i) nmcli connection add con-name "my connection" if name ens160 type ethernet

ii) connection 'My connection - LAN'
Activate the profile

Result :-

Thus the network commands used in linux and windows and studied

Student observation

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

The ping command in networking is used to test the reachability of a host an internet

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

Trace router command

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

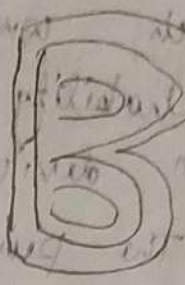
ipconfig command

4. which command displays the TCP port status in your machine?

netstat command

5. Write the modify the ip config in a Linux machine?

1. use the ipconfig command followed by the name of your network interface and the new i/p address to be changed on your computer.
2. change the ip address in the file ./. etc / system file / network scripts / ifcfg etho, and restart the system.



white / orange stripe

orange solid

white / green stripe

blue solid

white / blue stripe

green solid

white / brown stripe

brown solid

white / green stripe

green solid

white / orange stripe

white / brown stripe

brown

orange solid

blue stripe

Exp: 2

AIM:-

Study of different types of network cables

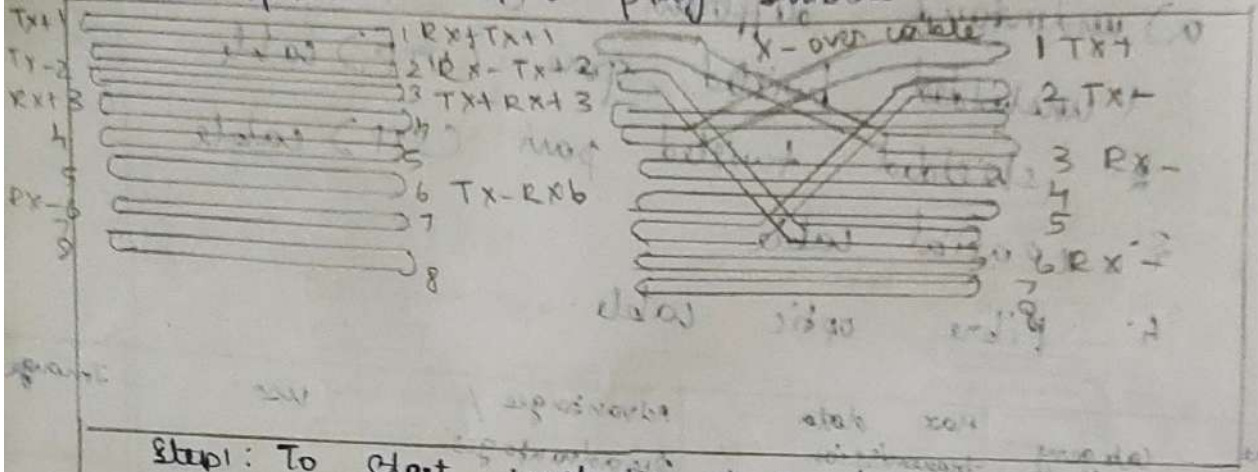
a) understand different types of cables

1. unshilded twisted pair (UTP) cable
2. shielded twisted pair (STP) cable
3. coaxial cable
4. Fibre optic cable.

cable type	category	Max data transmission	Advantages / Disadvantages	use	Image
UTP	category 3 5 6	10 bPS upto 100mbps 1 GTB ps	Advantages * Cheaper in cost * easy to install Disadvantage more EMI	* Fast ethernet gigabit ethernet	
STP FSTP	category 5 category 7	10 GbPS	A * shielded * faster than utp * less prone to noise D * effort	gigabit ethernet 10 g ethernet (55 m) gigabit ethernet	
coaxial cable	RG-6 RG-59 RG-11	10-100 Mbps	A * High bandwidth * Immune to interfe D * size * cost	Speed of signal is 500 m Television network high speed connected	
fibre optic	Single mode multi mode	100 Gbps	Advantages * High Speed * High Secure D * cost high	Max distance of fibre optic cable around 100 m	

b) Make your own ethernet crossover cable straight cable tools and parts needed.

* Ethernet cabling: CAT5e is certified for gigabit support but CAT5 cabling works as well, just over shorter distances.
* Two RJ45 plugs
* optional two plug shields.



Step 1: To start construction of the device, begin by threading shields on the cable.

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

Step 3:- After you will need to untangle the wire there must be four twisted pairs refering back to the sheet arrange them from top to bottom. one end should be in arrangement A & B.

Step 4: Once the order is correct, bunch them together in a line and if the order is correct, create a level without mess.

Step 5. Next push the cable right in. The notch at the end of the plug needs to be just other the cable shielding and if it exists.

Step 6: After the wires are securely sitting inside the plug insert it into crimping tool and push down.

Step 7: Last repeat for the others.

Result:-

Thus different types of network cables are understood and studied.

Exp: 3

Aim:

To study the packet tracer tool install and user interface over view.

c) To understand environment of cisco packet tracker to design simple network.

Introduction:

A simulator, as the name suggests stimulates network devices and its environment. packet trace is an exciting.

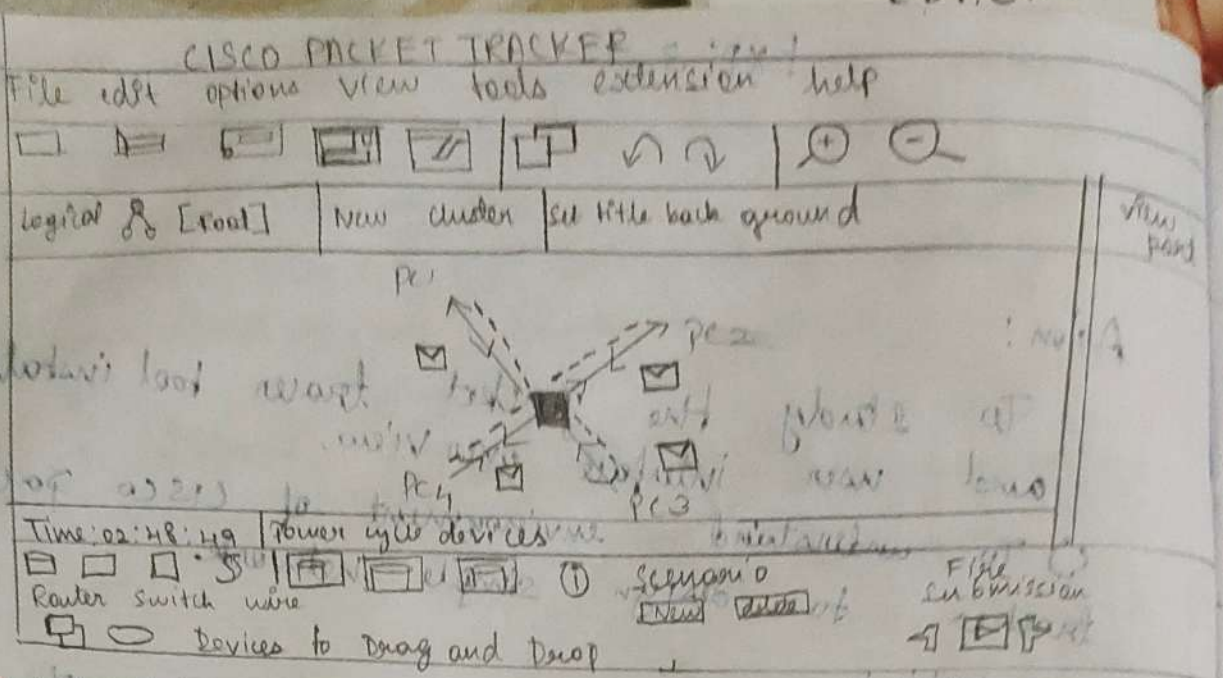
1. It allows you to model complex system without the need for dedicate equipment
2. It helps you to practice your network based mobile device
3. It is available for both the Linux and windows desktop environment.
4. protocols in packet tracer are coded to work in same way.

Installing packet tracer:

To download packet tracer, go to https://www.netacad.com and login with your cisco networking academy credentials. Then click on the packet tracer graph and download the package appropriate for your operating system.

windows

Install in window is simple and straight forward. location and start the installation.



1. Menubar - This is a common found in all software application it is used to open, save, print, change and soon.
2. Main tool bar - This bar provides shortcut icons to menu options are accessed.
3. logical / physical workspace tabs - These tabs allows you to toggle.
4. workspace - This toolbar provides control for manipulating topologies, such as select, move layout, place, inspect etc.
5. common tool bar - provides control for manipulating topology.
6. Sketch user created packet box. users can create highly customized packets to test their topology.
7. Real time / simulation tabs - These are used to toggle between them.

d) Analyse the behaviour of network devices using cisco packet tracker.

1. From the network component box click and drag and drop.
   a. 4 generic pcs and 1 hub
   b. 4 generic pcs and 1 switch

2. Click on connections:-
   a. Click on copper straight - cable
   b. Select one of the pc and connect to hub using the cable.
   c. Similarly connect 4 pcs to the switch using copper straight through cable.

3. Click on pcs connected to hub, go to desktop tab, click on IP config and enter an ip address and mask. Here the default gateway is in the network.

Click on the PDU from the tool bar.

4. Drag and drop it on one of PC.
   Observe the flow of PDU from the source PC.

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

9/8

Result:-
Thus the exp on cisco packets are executed.

Student observation:-

EX: 2

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

Straight cable

* The wiring of both the ends of the cable is identical.
* used for connecting different types of devices.
eg pc to switch / router.

cross cable

* The transmiter and receive wires are called one end of user.
* used for connecting similar devices.
Ex: pc to pc.

2) which type of cable is used to connect two pc?

A cross cable is used to connect two pc's directly.

3) which type of cable is used to connect a router / switch to your pc?

A straight cable is used to connect a router or switch to a pc.

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

You need to physically inspect the ethernet cable connected to your pc. The cable typically has its category printed along the length of cable.

Ex. 3 Student observation

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

HUB

Broadcasts: sends incoming packets to all connected devices, regardless of the destination collision domain leading to potential data.

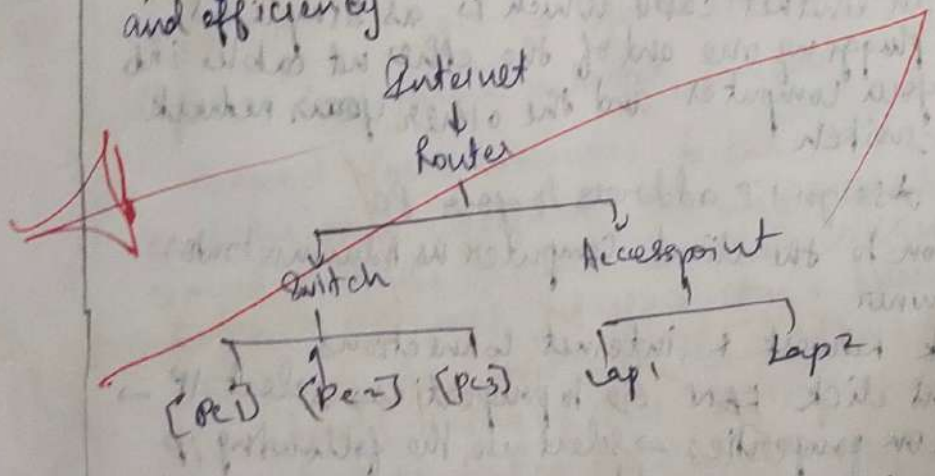
Switch:

Unicasts: when a switch receives a packet on one of its ports it checks the destination MAC address and forwarding the packets received by them only to the port associated with the MAC address

2. Find out the network topology implemented in your college and draw and label that topology in your note

Star topology

All devices are connected to a central switch or hub. This is one of the most common and widely used topologies in modern networks due to its simplicity and efficiency



Exp: 4

AIM:

Setup and Configure a LAN using a switch & ethernet cables in your Lab

What is a LAN?

A LAN connects devices within a limited area like an office or school allowing users to share resources such as data printers and internet access. A LAN switch acts as a central device managing & directing communication between connected devices for fast and secure data transfer

How to setup LAN?

Step 1: Plan & design an appropriate network topology taking into account network requirements and equipment location

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

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

Step 4: Assign IP address to your PC

Logon to the client computer as Administrator or as owner

Click network & internet connections

Right click LAN Go to properties select IP → Click on properties → Select use the following IP & assign the IP address

Internet protocol version 4 (TCP/IPV4) properties

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

○ Obtain IP address automatically
○ Use the following IP

IP address [10.1.1.1]
Subnet mask [255.0.0.0]
Default gateway [. . .]

○ Obtain server address automatically
○ Use the following DNS server address
Preferred DNS server [. . .]
Alternate DNS server [. . .]

☐ Validate settings upon exit

[Advanced...]

Similarly assign IP address to all PCs connected to switch

PC1 - IP : 10.1.1.1 Subnet mask 255.0.0.0
PC2 - IP : 10.1.1.1 Subnet mask 255.0.0.0
PC3 - IP : 10.1.1.3 Sub net mask 255.0.0.0
PC4 - IP : 10.1.1.4 Subnet mask 255.0.0.0

Step 5: Configure a network switch

Connect your computers to the switch to access the switch's web interface. You will need to connect your computer to the switch using an ethernet cable. Login to the web interface open a browser and enter the IP of the switch in the address bar This should bring up the login page for the switch's web interface Enter the username & password to login Configure basic settings once you're logged in you will be able to configure basic settings for the switch Assign IP as 10.1.1.1 subnet mask 255.0.0.0

Step 6: Check the connectivity between switch & other machine using ping

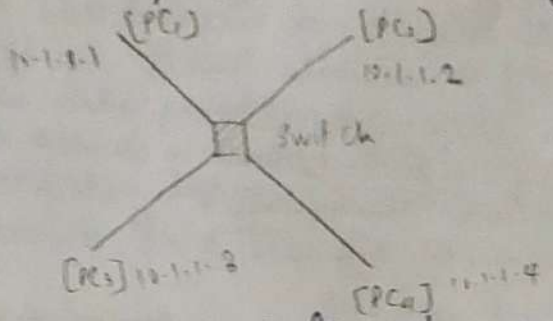
Step 7: Select a folder go to prop → sharing tab share it with everyone on the same LAN

Step 8: Try to access the share folder from other computers of the network

Result: Thus the above experiment on LAN configure executed successfully.

Student observation

Draw a net diagram of LAN in the configuration observation book that you have implemented in your lab. Write the ip configuration of each & every device. Write the outcome and challenges faced while configuring the LAN



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

Challenges Faced

ensuring each PC has unique IP to ensure conflict

Initially difficult accessing the switches web interface due to incorrect IP address entry or login credentials

Ensuring proper cable connections to avoid loose connections that could lead to network cables

Exp :5

Aim:

Experiments on packet capture tool wireshark

Packet sniffer

Monitors network traffic sent to and from your computers

captures & display the details of various protocol fields within the data packet

Operates in passive mode

Never transmit packet itself

Does not receive packets directly addressed to it

Obtains copies of all packets

Diagnostic tools:

Tcp dump

Ex: Tcpdump -enx host 10.124.41.2 -w exe 3-out

wireshark

Ex: wireshark -r exe-3.out

Description:

Wireshark:

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

What can we do with wireshark

Capture network traffic

Decode various packet protocols

Apply filters to capture & display specific data

Monitor statistics & analyse problems

Interactively explore network traffic

Users

Network Administrators

Security Engineers

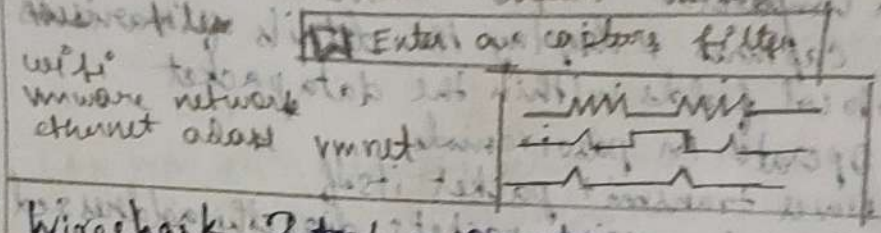
Developers

Learners

Capturing packets

launch wireshark

Double click the network interface under capture to start capturing packets



Wireshark Interface overview

1. Stop capturing traffic

Click red stop button near left corner

2. Packet list pane

Displays all packets in a current capture file

3. packet details pane

Shows detailed information of selected packet. Displays protocol & its fields in a format called tree

4. packet bytes pane

Shows the selected packets data in a hexdump style shows the data of the current packet colour coding

Sample

Use sample files to practice in wireshark

open file → open

save your capture with file > save for later review

Packet list

No	Time	source	Destination	protocol	Info
64	56.85	192.168.2.100	10.100.10.2	ICMP	echo (ping)
65	86.86	10.100.10.2	192.168.2.100	ICMP	echo (ping)
66	44.40	192.168.2.100	[illegible]	[illegible]	[illegible] request

Packet details

Frame 32 (82 byte on wire 86 bytes captured)

Ethernet II src intel (00 92 d [illegible] 94)

user datagram protocol src port [illegible] (1024)

source port: 1024 [illegible] Destination port: snmp (161)

Packet bytes

0000 00 0c 2c bc 2f [illegible] d8 [illegible]

0010 00 48 02 d4 00 06 11 02 60 [illegible] 02 dd

Filtering packets

Apply filters to focus on specific network traffic
use other apps to isolate traffic for analysis
Type a filter enter eg:- dns
use analyse -> display filters to pick or save filters see the docs for more

Create a filter to display only DNS packets & provide flow graph

Go to capture -> option
Select stop capture after 100 packets
Click start capture
Search DNS packets
See flow graph by statistics -> flow graph
Save the packets

Create a filter to display only http packets

Go to capture -> option
Select stop after 100 packets
Click start capture
Search http packets
Save the packets

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

Select LAN, go to capture option

Select stop capture after 100 packets

Click start capture

Search ICMP/IP packets in search bar

save the packets

Capturing & analysing packets using wireshark tool

To filter, view capture packets, Capture 100 packet from the ethernet

Select LAN, go to Capture → option

Select stop captures automatically after 100 packets

Then click start capture

Save the packets.

Info

click a packet, choose follow TCP stream to see the full conversation.

we follow for other protocols.

capturing & analysing packets using wireshark tool. To filter, view capture 100 packets from ethernet

Procedure:-

Select LAN go to capture → option

select stop captures automatically after 100 packets then click start capture

Save the packets.

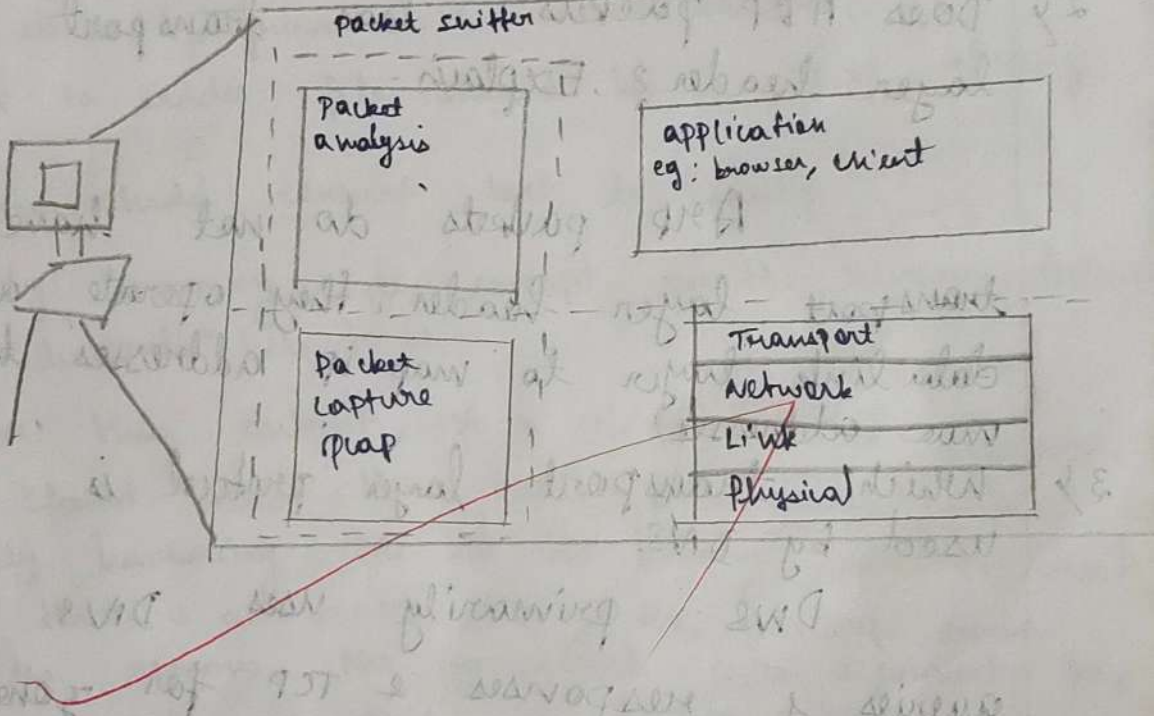
Create a filter to display only TCP/UDP packets.

Inspects the packets & provide flow graph

select LAN, go to capture → option

select stop capture after 100 packets.

save the packets.



Result:-

Thus the experiments on packet capture tool wireshark is executed successfully.

Ex 5, Student observation:-

1) What is promiscuous mode:

A network interface in promiscuous mode captures all traffic on the network segment regardless of destination address allowing for comprehensive sniffing.

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

ARP packets do not have a transport layer header. They operate at the data link layer to map IP addresses to MAC addresses.

3) Which transport layer protocol is used by DNS

DNS primarily uses DNS for queries & responses & TCP for zone transfer.

4) What is the port number used by HTTP protocol.

The HTTP protocol uses port 80 by default communication.

Exp: 6

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

Receiver:-

Create sender program with below features:-

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

1. It should read the input from channel file.
2. Apply hamming code on the binary data to check
3. If there is an error, display the pos of error.
4. Else remove the redundant bits & convert the binary data to ascii & display output.

code:-

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

```
# fn to calculate redundant bits
def calc_red_bits(m)
    r=0
    while (2**r < m+r+1)
        r+=1
    return r
# fn to insert redundant bits into data
def pos_red_bits(data, r)
    j=0
    k=0
    m=len(data)
    res=''
# Adding redundant bits at pos that are powers of
for i in range(1, m+r+1):
    if i==2**j
        j+=1
    else:
        res=res+data[k]
        k+=1
    return res
# fn to calculate parity bits
def cal_parity_bits(arr, r)
    n=len(arr)
    arr=list(arr)
    for i in range(r)
        p=0
        pos=2**i;
    for j in range(i, n+1):
        if j & pos:
parity=int(arr[j-1])
    arr[pos-1]=str(p)
    return ''.join(arr)
```

```
# fn to detect & correct errors
def detect_correct(data, r):
    n = len(data)
    res = 0
# calculate parity bits
    for i in range(r):
        parity = 0
        pos = 2**i
        for j in range(i, n+1):
            if j & pos:
                parity ^= int(data[j-1])
        if parity != 0:
            res += pos
if res != 0:
    print("error at pos: {res}")
    data = list(data)
# correct the error
    if res <= n:
        data[res-1] = '0' if data[res-1] == '1' else '1'
        print(f"Error corrected at pos: {res}")
    else:
        print("error pos out of range. No correction")
    corrected_data = ''.join(data)
    return corrected_data
else:
    print("No error detected")
    return data
# fn to remove redundant bits
def remove_bits(data, r):
    j = 0
    original_data = ''
    for i in range(1, len(data)+1):
        if i == 2**j:
```

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

# fn to introduce an error (data, position)
if position < 1 or position > len(data):
    print("Error: pos is out of range")
    return data
data = list(data)
# flip the bit at specified pos
data[position-1] = '0'
    if data[position-1] == '1' else '1'
print(f"Introduced error at pos: {position}")
return ''.join(data)

# sender program
def sender(text):
    binary_data = text_to_binary(text)
    m = len(binary_data)
    r = calc_red_bits(m)
    arr = pos_redundant_bits(binary_data, r)
    arr = calc_parity_bits(arr, r)
print(f"Sender output (binary with redundant
        bits): {arr}")
return arr

# receiver program:
def receiver(data):
    r = calc_red_bits(len(data))
corrected_data = detect_correct(data, r)
ascii_output = binary_to_text(original_data)
print(f"decoded text: {ascii_op}")

# Main program:
if name == "main":
    text = input("Enter the text")
```

channel data = sender (input text)
corrupted data = introduce_error (channel data, 2)
receiver (corrupted_data)

Output:-

Enter the text to be encoded: ak
Sender message in binary : 01100001011010 11
no. of parity bits : 5
parity bits / redundant bits for sent message:
P1 : 1
P2 : 0
P4 : 1
P8 : 1
P16 : 1

The hamming code for even parity
: 10011101000 10111 01011

Enter pos to change the bit: 6
Revived message in binary: 01100001011010 1)
Decoded message at receiver side : ak

Result:- Thus the above code (hamming code) executed and verified.

9/11

good

Exp-7

AIM:-

Write a program to Implement flow control at data link layer using sliding window protocol stimulate the flow of frames from one node to another.

Create a sender program with following features

1. Input window size from the size
2. Input a text message from the user
3. Consider 1 character per frame.
4. Create a frame with following fields.
5. Send the frames.
6. Wait for the acknowledgement from the receivers.
7. Read a file called receiver buffer.
8. Check ACK field for the acknowledgement
9. If the no is as expected, send new set of frames accordingly. Else if NACK is received resend the frames accordingly.

Create a receiver file with following features.

1. 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 receiver-buffer-file Else write NACK no in the receiver buffer file.

code:-

```
import time
import random

class frame:
    def __init__(self, frame_no, data):
        self.frame_no = frame_no
```

```
self.data = data
self.acknowledged = False
def send_frames(frames, window_size):
print("Sending frames")
for i in range(window_size):
if i < len(frames) and not frames[i].acknowledg
print(f"Sent frame {frames[i].frame_no}
{frames[i].data}")
print("frames sent, waiting for acknowledgement")
def receiver_frames(frames, window_size):
print("Receiving frames"):
for i in range(window_size):
if i < len(frames) and not frames[i].acknowl
if random.random() < 0.7:
print(f"received frame {frames[i].frames_no}
{frames[i].data} [ERROR]")
frames[i].acknowledged = True.
def sliding_window_protocol():
window_size = int(input("Enter window size:"))
Message = input("Enter a message to send:")
frames = [Frame(i, message[i])
for i in range(len(message))]
base = 0
while base < len(frames):
send_frames(frames[base:], window_size)
time.sleep(2)
receive_frames(frames[base:], window_size)
while base < len(frames) and frames[base]
acknowledged
base += 1
if base < len(frames):
print("Resending ask acknowledged frames"):
time.sleep(2)
print("All frames sent a acknowledged")
```

if - name == "main":
Sliding - window - protocol()

Output:- Enter window size: 5
Enter window a message to send : GBISH

Sending frames	Receiving frames
sent frame 0: A	Received frame 0: A [OK]
sent frame 1: G	Received frame 1: G [OK]
Sent frame 2: I	Received frame 2: I [OK]
Sent frame 3: S	Received frame 3: S [OK]
Sent frame 4: H	Received frame 4 : H [OK]

~~Frame Sent 5: y~~
sending, wait for ack
All frames are sent and acknowledged

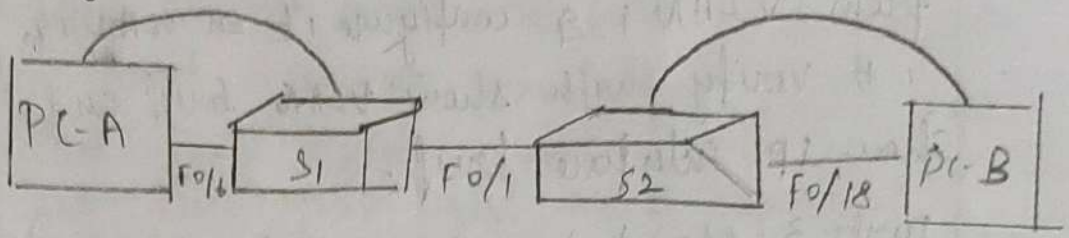
19/11.

Result:- Thus the sliding window protocol is executed successfully.

Exp:8A
8/10/2024

8A -

AIM - (a) Simulate virtual LAN configuration using CISCO packet tracer simulation.



Device	Interface	IP Address	Subnet mask	Default
S1	VLAN-1	192-168-11	255.255.255.0	N/A
S2	VLAN-1	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

Part-1: Build the network and configure basic device settings.

Step-1: Build the network

Objective:- Connect the device as shown in the topology.

steps:-

* Drag switches S1 & S2 to the rack
* Drag PC-A and PC-B to the table & power them on.
* connect the devices with copper straight through cables.

Step 2:- Assign VLANS to switch interfaces.

Objective: Assign ports to VLANS.

Steps:-

* Assign PC-A to VLAN 10 (operations)
* Remove the management ip address from VLAN 1 & configure it on VLAN 99.

steps

* Assign PC-A to VLAN 10.
* Remove the management IP address from VLAN 1 & configure it on VLAN 99.
* Verify with show VLAN brief and show IP interface brief.

Part-3: Maintain VLAN port assignment and the VLAN database.

Step 1: Assign VLAN to multiple interface

objective: Assign multiple interfaces to a VLAN.

steps * Assign VLAN 99 to interfaces F0/11-24 on S1.

step 2: Remove VLAN assignment from interface.

objective:- Remove VLAN assignment from an interface.

steps:-

* Use the no switch port access VLAN command to remove VLAN assignment from F0/24.

Part-4:- configure an 802.1Q Trunk between switches.

Step-1 : Use DTP to initiate trunking.

Objective: configure dynamic trunking protocol.

Step - 2 :- configure basic switch settings

Objective :- configure both switches.

Steps :-

* Use the terminal in each pc to console the switch & enter privileged EXEC mode.
* Set the device name for each switch
* set the privileged exec password to class.
* Set the console password and enable login
* Encrypt plaintext password.

step-3: configure pc hosts

Objective: Assign ip address to PCA and PCB from addressing table.

Steps :- In ip configuration input the ip address for pcs test connectivity.

Step 4 :- Test connectivity.

objective :- Test pings between devices.

close configuration window.

Part 2 - create VLANS on both switches

step1 - create VLANS.

Objective: create VLANS on both switches.

Steps :- use the VLAN command on S1 and S2 to create VLANS operations, parking, management & Native.

Step 2 : Assign VLAN to switch interfaces

objective : Assign ports to VLANS

step

Assign PCA to VLAN 10 (operations)

Remove the management ip address from VLANS.

Part 4: Configure an 802.1Q trunk between switches.

Step1: Use DTP to initiate trunking.

Objective:- Configure dynamic tracking protocol DTP on interface f0/1 to negotiate a trunk between S_1 and S_2.

Steps:- Set the trunk mode using switch port mode dynamic desirable on S_1.

Verify using show interfaces trunk & ensure trunking is enabled between S_1 & S_2

Questions:-

1. Can S_1 ping S_2?
   Yes if trunking is successfully configured S_1 can ping S_2.
2. Can PC-A ping PC-B?
   Yes, if VLANs are properly configured + trunking is enabled PC-A can ping PC-B

Step 2:- Use the command switch port mode trunk to force trunking FOR on both switch.

Reflection questions:-

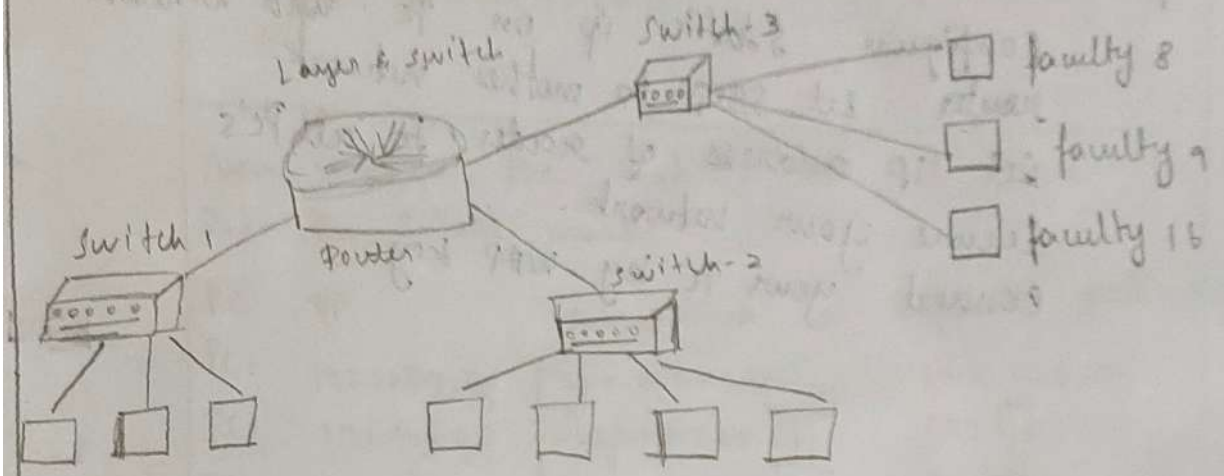
1. What is needed to allow hosts on VLAN to communicate to host on VLAN 99?
   We need a 3 layer device such as a router or a 3 layer switch the inter VLAN routing configured.
2. What are the primary benefits that an organisation can receive through effective use of VLANS?

* Improved network segmentation.
* Enhanced security by isolating traffic.

* Reduced broad cast traffic
* Better network management & flexibility

Student observation:-

a) Draw and label the VLAN for 10 faculty in robotics department, sitting in 3 different blocks.



b) Show the ip configuration for each device

faculty-1 192.168.10.1/24 faculty-2 192.168.10.2/24
faculty-3 192.168.10.3/24 faculty-4 192.168.10.4/24
faculty-5 : 192.168.10.5/24 faculty-6 192.168.10.6/24
faculty-7 : 192.168.10.7/24 faculty-8 : 192.168.10.8/24
faculty-9 192.168.10.9/24 faculty-10 : 192.168.10.10/24

c) Write the commands for VLAN configuration in switch.

switch (config) # Vlan 10
switch (config-vlan) # name Robotics-VLAN
switch (config-vlan) # exit
switch (config) # interface range for 0/1-10
switch (config-if-range) # switchport mode access
switch (config-if-range) # switch port access vlan 10
switch (config-if) # exit.

Result:- Thus the above configuration for Virtual LAN is executed.

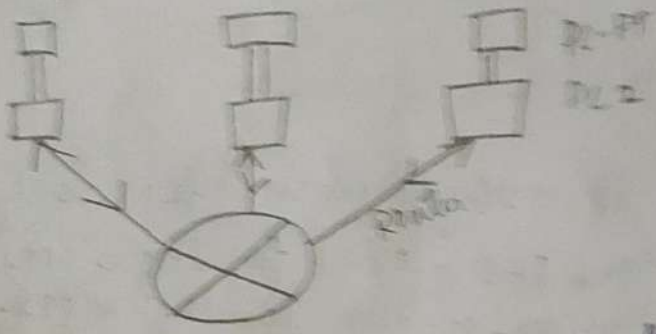
21/9/24
Ex. No: 8b

Aim:-

To design a topology with three PCs connected from linksys wireless router

procedure:-

configure static IP on PC and wireless router set SSID to router network
Set IP address of router to all PCs
Secure your network.
Connect your PC by WEP key



Step 1: Click on wireless Router

Step setup	wireless	Security	Access restriction	Administration
Management	Router pass: admin			
Router Access	Router pass: admin			

Setup	wireless	security	Access restriction	Administration
Security mode	Disabled Disabled WEP			

set Key 1

Setup wireless security | Access Restriction | Administration

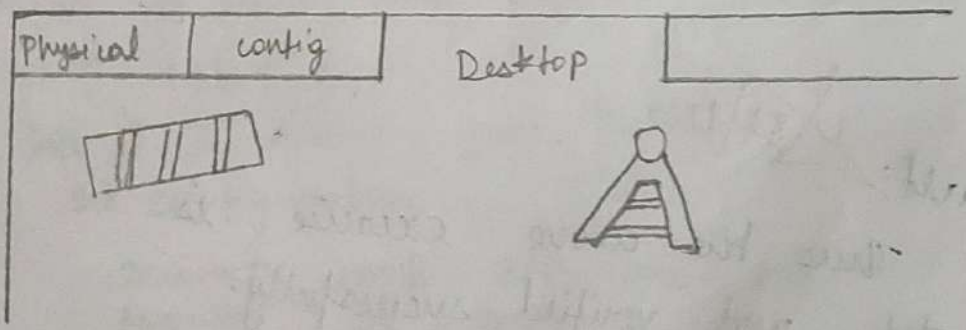
security mode wep
Encryption
Pass phrase
Key 1 : 0123456789

Now configure the static ipon all three PCs and set the subnet mask.

PC	IP	Subnet mask	Default gateway
PC0	192.168.0.2	255.255.2556	192.168.0.1
PC1	192.168.0.3	255.255.255.0	192.168.0.1
PC2	192.168.0.4	255.255.255.0	192.168.0.1

Now its time to connect PCs from wireless router Goclick PC Select desktop click on PC wireless

click Pc wireless

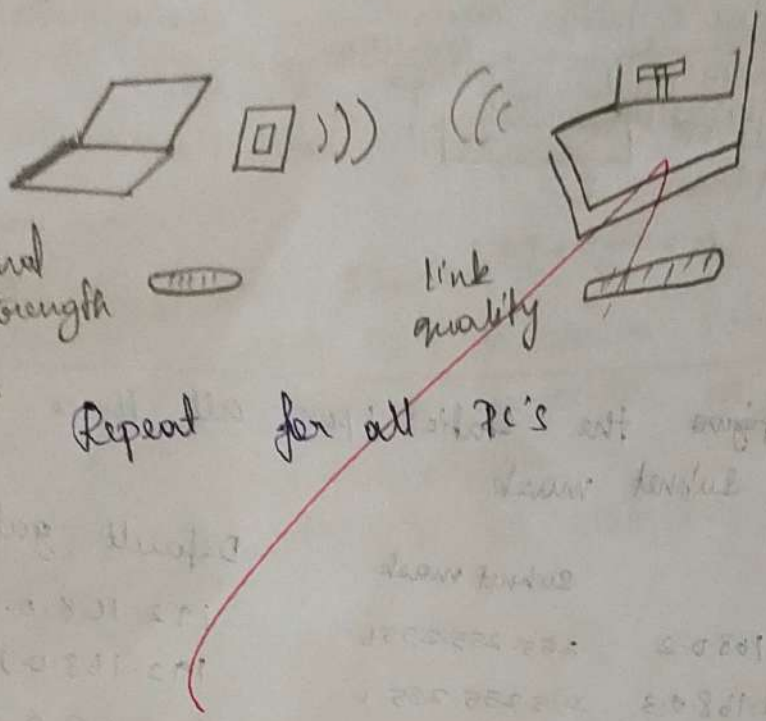


Click on connect tab and click on refresh button

NEP key
Security WEP
WEP: 64 bit
WEPKEY 1 : 0123456789

cancel cannot

Click on connect button to connect network. It will ask for WAP key



Repeat for all PC's

Result:

Thus the above exercise is executed and verified successfully.

Student observation:-

c) What is SSID a wireless router.

The SSIDC Service set identifies of wireless router is the network name that identifies a wifi network. It allows devices to connect to it.

d) what is a security key in wireless router?

A security key in a wireless router is a password used to protect a wifi network ensuring that only authorized users can connect common types include WEP, WPA, WPA2 keys.

e) configure a simple wireless LAN in your LAB using a real access point & write down the configurations in your network.

configuration:-

Access point setup: connect the access point to power and network then access.

eg: 192.168.1.1

set SSD name your network

set configuration.

security mode WPA2-PSK

password: set a key.

Save settings:- Apply the changes and save the access point.

ex:

SSID: cab-wifi

security key: cabsecure DB

security mode: WPA2-PSK

IP range 192.168.1.100 to 192.168.1.150

channel 6

Mode 802.11 b/g/n

Ex: 9

Aim:-

Implementation of subnetting in cisco packet tracer simulators.

classless ip subnetting is a technique that allows for more efficient use of ip address by allowing for subnet masks that are not just the default masks for each ip class. This means that we can divides are ip address space into smaller subnets which can be usful when we leave limited no of ip address but need to create multiful network.

Creating a network topology

The first step in implementing classless ip subnetting is to create a network topology in packet tracer select new button in the topleft corner the select 'network' generic

Adding the devices.

Once we have created our network topology we can add devices to it. Here we will be adding routers, switches, PCs To add a device select the device from the bottom left corner and drag it onto the network topology.

subnetting:-

To subnet the network address of 192.168.1.0/24 to provide enough space for atleast 5 addresses for end devices, the switch & the router we can use a 2^3 subnet mask. This will give us a subnet with 30 host address each.

The ip addressing for the network shown in the topology can be as followed.

Router R1:
switch S1:

Fast
PC1: 192.168.1.11
PC2: 192.168.1.12
PC3: 192.168.1.13
PC4: 192.168.1.14
PC4: 192.168.1.15

Fast
ethernet 0/2
192.68.2.0/27
PC1: 192.168.2.11
PC2: 192.168.2.12
PC3: 192.168.2.13
PC4: 192.168.2.14
PC5: 192.168.2.15

Router R2
switch S2

Fast
ethernet 0/1
192.68.3.0/27
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

Fast
ethernet
192.168.4.0/27
PC1: 192.168.4.11
PC2: 192.168.4.12
PC3: 192.168.4.13
PC4: 192.168.4.14
PC5: 192.168.4.15

configure the devices:-

Router configuration:-

Access the CLI: Right click on the router and select 'CLI' to open the command line interface.

configure interface:-

* Enter enable & configure terminal to login configuration mode.

Fast ethernet 0/0:

Enter: Interface ethernet 0/0

Set ip: ip address {Ip address} {Subnet mask}

Active: no shutdown.

Exit: Interface configuration: exit.

Fast ethernet 0/1:

* Repeat the above steps. connecting this interface to one of the PCs.

configure gigabitethernet:

Use interface gigabitethernet 0/0, set ip and subnet. activate with no shutdown & exit.

Switch configuration

Enter enable and configure terminal.
Enter interface fast ethernet 0/1 then switchport mode access, exit.
Repeat for fast ethernet 0/2 for connecting to the second PC

Pc configuration:-

* Must be in the same subnet as the router is fastethernet 0/1 interface.
* Set to the ip of the router interface connected to the PCs.
* Enter DNS detail are reached.

Testing the network:-

A successful ping indicates proper PC-to-PC communication.

This ensures router connectivity with PCs.

Result:- Thus the above connection and packet tracer was executed successfully.

Student observation:-

a) Write down your understanding of subnetting.

Subnetting is the process of dividing a larger network into smaller, manageable sub-networks. Each subnet operates with its own ip address range. helping to organize & manage network traffic.

b) What is the advantage of implementing subnetting with a network.

Efficient IP management.
improved network performance.
enhanced security.

c) Find out whether subnetting is implemented in your college.

college Subnetting:

Subnet 1 (Admin) - 192.168.1.0/24
Subnet 2 (library) - 192.168.2.0/24
Subnet 3 (labs) - 192.168.3.0/24

Ex No. 10.a)

Internetworking with routers in cisco packet tracer

Aim:- To create a simple network with a router connecting two PCs using a copper straight through cable. Then testing connectivity by sending a PDU from PC0 to PC1

Steps

1) Routers configuration:-

* Open CH, enter privileged mode (enable)
* Enter global configuration.
* configure fast ethernet 0/0:-
  Set Ip: 192.168.10.1 255.255.255.0
  Bring interface up (no shutdown)
* configure fastethernet 0/1:-
  Set ip: 192.168.20.1 255.255.255.0
  Bring interface up (no shutdown)

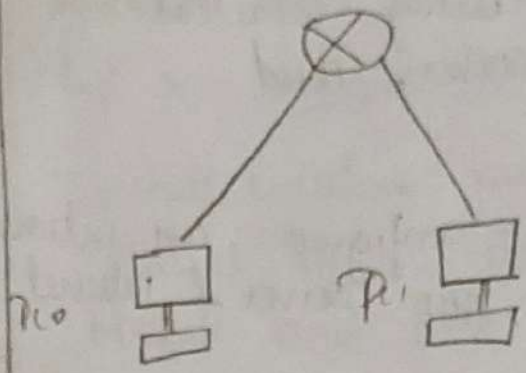
PC configuration:

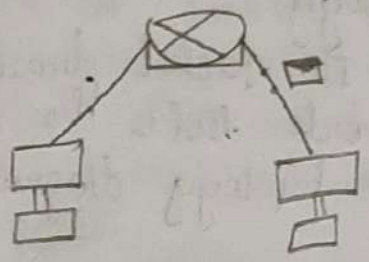
PC0: ip address: 192.168.10.2
Subnet mask: 255.255.255.0
Default gateway: 192.168.10.1

PC1: ip address: 192.168.20.2
Subnet mask: 255.255.255.0
Default gateway: 192.168.20.1

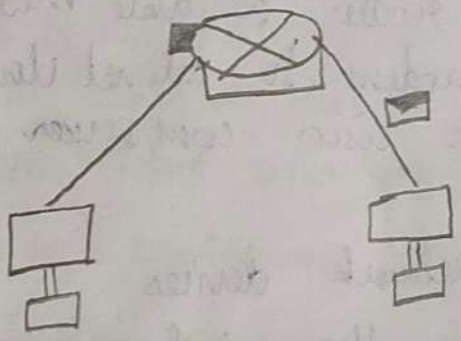
cable connections:

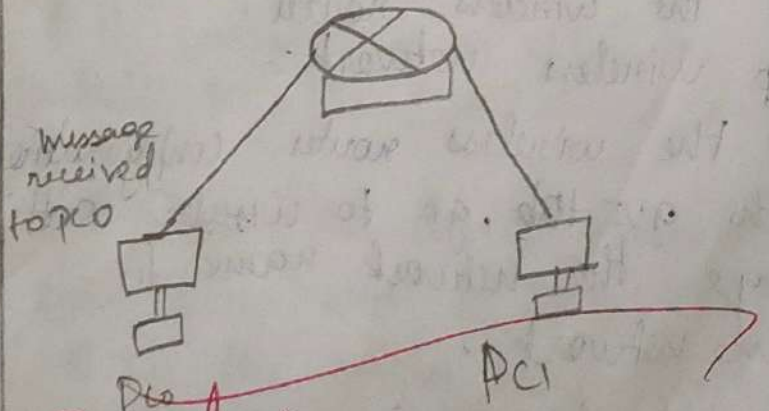
Send a PDU from PC0 to PC1 to view
Verify functionality.





Acknowledgement from Pc1 to Pc0





Result:- Thus the internet working with routers in cisco packet tracer is executed sucusfuly

Internetwork using wireless router, DHCP server, cloud

Aim:- To design and configure an internetwork using wireless router, DHCP server & internet cloud.

Steps:

launch packet tracer.
Build the topology.
* Add devices:- In packet tracer, select & place each network device the workspace according to the topology diagram.
* Rename device.
* connect devices with cables.
  - Pc to wireless router.
  - wireless router to cable modem
  - cable modem to internet cloud.
  - cloud to cisco.com server.

configure the network devices.
Step 1:- configure the wireless router.
setup wireless network:

Open the wireless router.
setup wireless network:-
Open the wireless router configuration.
in the GUI tab, go to wireless sitting change the network name to "Home network".

setup internet connection:
* In the setup tab ensure DHCP server is enabled.
* Set the DNS server IP address to 208.67.220.220

* click save settings

step 2 configure the laptop

Install wireless module

Insert the wireless WPC 300N Module and turn the laptop back on.

connect to wireless network:

In the desktop tab, under PC wireless settings, under connect select "Homenetwork" from the list of networks & connect.

step 3:- configure the PC

i) Enable DHCP

ii) Verify IP address

open command prompt on the PC ipconfig all to confirm the PC received an IP address in the 192.168.0.x range.

step 4:- configure the internet cloud.

① Install network modules (if needed)

* click the internet cloud icon, go to the physical tab.

* if missing power off the device and install PT-Cloud-NM-1CX and PT-cloud-NM-1CF modules.

* power the device back on.

set from and to ports:-

* In the config tab, select cable under connections.

* set from port as coaxial and to port as Ethernet.

step 5 configure the cisco.com server
select DHCP from the services in the left pane.

Click on the turn the DCHP service on
pool name: DHCP pool
Default gateway: 208.67.220.220
DNS server: 208.67.220.220
starting IP address: 208.67.220.1
Max. No. of users: 50

a) click add to add the pool.

b) configure the cisco.com server as a DNS server to provide domain name to IPV4 address

while still in the services tab select DNS server to provide domain name to left pane:

Click on to turn the DNS service on
Name: cisco.com
Type: A Record
Address: 208.67.220.220

Click Add to add the DNS service setting

configure cisco.com server global settings:-

Go the config tab, select settings configure as follows:
Select: static
Gateway: 208.67.220.1
DNS server: 208.67.220.220

Select : static
IP address : 202.67.220.220
Subnet mask : 255.255.255.0

Verify connectivity.

Refresh IPV4 settings on PC.

open Desktop > command prompt

Run ipconfig /release and ipconfig /renew to confirm the IP is in the 192.168.0.x range

From the PC's configure prompt issue ping cisco.com to verify connection.

19/11

Result:-

Thus all the connections are given and executed successfully.

11.a)

Aim:- Simulate static routing configuration using cisco packet tracer.

1. Setting up lab
   * Open cisco.
   * Create a network topology with 3 routers, each connected to different network.

2. Ip address configuration.
   * Assign Ip to routers.
   Router 0 connected network: 10.0.0.0/8, 20.0.0.0/8, 40.0.0.0
   Router 1 and Router 2 will have other network.

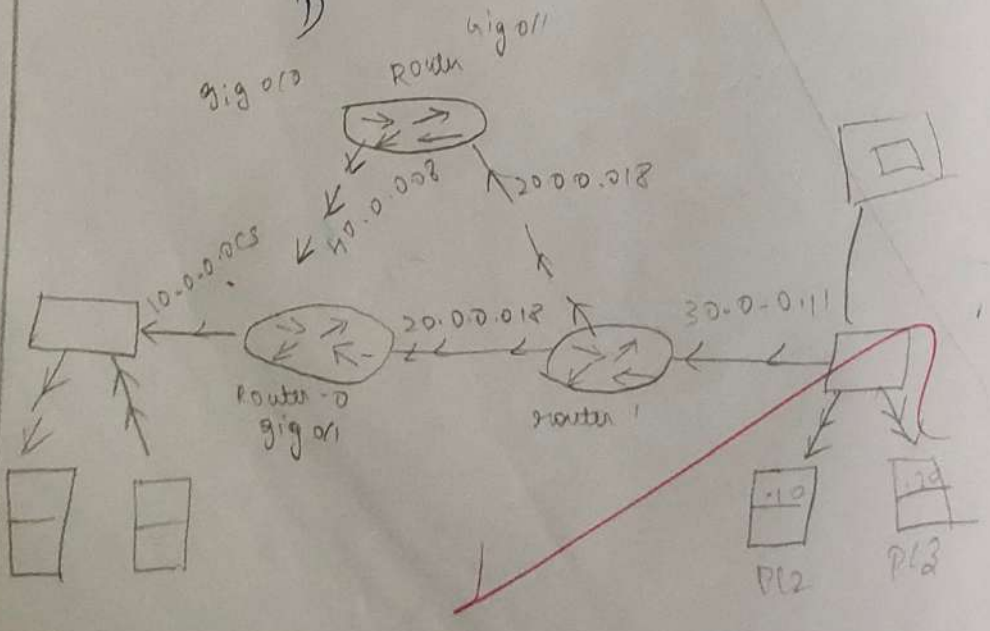
3. Static routing configuration.
   In router 0 add static route for network not directly connected to it like 30.0.0.0/8 and 50.0.0/8. ran commands.

Router # Ip route 30.0.0.0 255.0.0.20.0.0.8
Router (config) # Ip route 50.0.0.0 255.00.0 40.0.0

The AD value is 10 for the route and 20 for backup route.

4. Verification:
   To verify the static router added use the command.
   Main route fails the backup route will automatically be added.

D



The following table lists the connected networks of each routers.

Router	Available networks on local interface	Networks on other router
Router 0	10.0.0.0/8 20.00.0/8 40.0.0.0/8	30.0.0.0/8 50.0.0.0/8
Router 1	20.0.0.0/8 30.0.0.0/8 50.0.0.0/8	10.0.0.0/8 40.0.0.0/8
Router 2	40.0.0.0/8 50.0.0.0/8	10.0.0.0/8, 20.0.0.0/8 30.0.0.0/8

Router# configure terminal.

Router (config)# ip route 30.0.0.0 255.0.0.0 20.0.0.0

Router (config)# ip route 30.0.0.100 255.255.255.255 40.0.0.0/8

Router (config)# exit.

Router # show ip route static

30.0.0.0/8 is variably subnet, 2 subnet masks.

S 30.0.0.0/8 [10/0] via 20.0.0.2

S 30.0.0/32 [10/0] via 40.0.0.2

S 50.0.0/8 [10/0] via 40.0.0.2

Router#

Router 1 requirements

Create 2 routes for network 10.0.0.0/8 and configure first route (via router 0) as main route and second route (via-routers) as backup route.

Create 2 routes for network 40.0.0.0/8 and configure first value (via-router 2) as backup route

Router (config)# 10.0.0.0 255.0.0.0 20.0.0.1 10

Router (config)# 10.0.0.0 255.0.0.0 50.0.0.1 20

Router(config)# Ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router (config)# Ip route 40.0.0.0 2555.0.0.0 50.0.0.2
Router (config)# exit.

Router 2 configuration.

Router> enable
Router# configure terminal
Router (config)# Ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router (config)# Ip route 30.0.0.0 2550.0.0 50.0.0.2
Router (config)# exit

Verifying static routing.

1. Using tracer command.
   PC in network 10.0.0.0/8 run target command to 30.0.0.0/8

2. Check routing table
   In router 0, run command.
   router # show Ip route
   shows which route is currently for 30.0.0.1

output:-

1. Router # show IP route static
2. Router (config) # no ip route 30.0.0.0 255.0.0.0 <next hop ip>

19/4

Result:- Thus simulating static routing config using cisco packet is executed successfully.

11.b)

Aim: Simulate RIP using Cisco packet tracer

Initial IP configuration:

Device	Interface	I/P config	connected to
P/c	fastethernet	10.0.0.2/8	router 0 - fe
R0	fe 0/1	10.0.0.1/8	Pc0 fe
R0	fa 0/1	192.162.1.254/30	router 2 - se0/1
R0	se 0/1	192.162.1.250/30	router 1 - se0/1
R0		192.168.1.246/30	router 2 - se0/1
R1	se0/0/0	192.168.1.249/30	router 2 - se0/1
R1	se0/0/0	192.168.1.245/30	router 1 - se0/1
R2	se0/0/0	192.168.1.253/30	router se0/1
R2	se0/0/1	20.0.0.1/30	Pc1 fe
R2	fe 0/1	20.0.0.2/30	R2 - se0/1
Pc1	fe		

Assign Ip address to Pc

* Select Pc goto desktop → Ip config and assign Ip.

2. Assign Ip to router.
   * Open CLI in router
   * Run the command
   Router > enable
   Router # config me terminal

Assign Ip se and serial interface
Router (config) # interface fastethernet 0/0
Router (config) # ip address [ip] [subnet mask]
Router (config) # no shutdown
Router # exit

3. Router specific configuration.

Router 0: 10.0.0.1	192.168.1.249	192.168.1.253
Router 2: 20.0.0.1	192.168.1.245	192.168.1.253

4. Enable RIP routing protocol.
enter global config mode.
router (config)# router rip.
router (config-router)# network.

5. Verify connectivity.
Use ping command from PC to test connection.
This process. The IP's ~~interface~~ and enable the RIP protocol for routing between network

Results:- Thus simulation of RIP using cisco packet tracer is executed.

10.a) Echo using TCP client

Aim: Implement echo client server using TCP Sockets

1. TCP echo client-server algorithm

TCP Server algorithm

1. Create a TCP socket
2. Bind the socket to local address
3. Listen for incoming client connection
4. Accept a client connect
5. loop
6. Close the connection

TCP Server.py

```
import socket
def tcp_server():
    server_socket = socket.socket(socket.AF_INET,
                                  socket.SOCK_
    server_socket.bind(("local host", 12345))
    server_socket.listen(1)
print("TCP server is waiting for a connection")
print(F" connected to {client_address}");
try: while true.
data = connection.recv(1024)
    if data
    print("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_INET,
                              socket.sock_stream).
client.socket.connect(host,port))
message = input(" enter message :")
client_socket.sendall(message.encode)
data = client.socket.recv(1024)
print(f" Received from server {data.
  decode()}')
if __name__ == "__main__":
    tcp_echo_client().
```

output: Server.

Enter message to echo : hello world
received from server : hello world.

client ?

UDP server :-

```
import socket
def start_server():
server_sock = socket.socket(socket.AF_INET, sock_
                            DGRAM)
server_address = ('localhost ' 12345)
print(f" server listening on {server_adres}"
  while true:
    message, client_address = server_socket.recv
from(4096)
    printf(f"received message : {message.
decode()} from {client_address).
if __name__ == "__main__":
start_server()
```

Output:-

TCP Server is listening on 127.0.0.1:12345
connected by ('127.0.0.1', 11888)
Received: Hello
connected by ('127.0.0.1', 12272)
Received: Hello world

Enter message to send: Hello
Received from server: Hello
Enter message to send: Hello world
Received from server: Hello world

Result:- Thus the TCP/UDP socket is implemented successfully

Ex 12.b) Aim:-

TCP server:

```
import socket
def start_tcp_server(host='127.0.0.1', port=65432):
server_socket = socket
```

chat client.py:-

```
import socket
import threading
def receive_message(client_socket):
    while true:
    try:
        message = client_socket.recv(1024);
    if message:
        print(f"server: {message}")
except exceptions:-
    print(f" an error occured: {e}")
    break
def start_client():
client_socket = (
    )
host='127.0.0.1'
port = 12345
client_socket.connect(host, port))
print(" connected to ['server")
message = client_socket.recv(1024).decode
if message:
    print(f"server: {message}")
    exception exe
        print(f"an error occured - {e}")
    break
```

```
def start_client()
    client_socket = socket.socket(socket.AF_
                                  INET,
                                  socket.SOCK_STREAM)

host = '127.0.0.1'
port = 12345
client_socket.connect((host, port))
print("Connected to chat server")
threading.Thread(target=receive_message
                 args=(client_socket,))
daemon=True).start()

while True:
    message = input("You: ")
    client_socket.send(message.encode
if __name__ == "__main__"
    start_client()
```

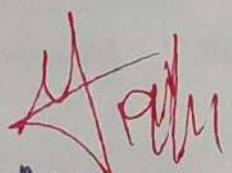
chat_server.py

```
import socket
import threading
def handle_client(client_socket)
while True
    try: message = client_socket.recv(1024)
    if not message
        break
    print("Received message from client")
except Exception as e:
    print("An error has occured")
    break
client_socket.close()
def start_server()
    server_socket = socket.socket
```

OUTPUT

```
>python chat server.py
 Chat server started on 127.0.0.1:12345
new connection from (127.0.0.1,57226)
received from client: Agish
Type from message to client: Hello
```

```
>python chat_client.py
 Connected to that server
   You : Agish
   You : Sever: hello
```



RESULT:

Thus the implementation of chat client server using TCP/UDP socket has been successfully executed & verified

Aim: implement your own ping own ping program:

Algorithm:-

ping-client.pg

1. Socket created
2. Then set a timeout of 2 second to ensure that if no response is received a it will stop waiting and print "Request timeout.
3. Send a ping msg to specified host and port.
4. It listens for a response and calculates the time difference between sending and receiving packet.

program:

```
import socket
def start_server(host='127.0.0.1', port=12345)
with socket.socket(AF=INET, Socket.socket)
print("Upp. server running on {host}:{port}")
while true:
    data addr = S.recv from (1024)
print(f"Received msg from {addr}
S.sendto(b'ping', addr).
if __name__=="__main__":
    start=server.
```

ping-client.py

```
import socket
import time
```

```
def ping_server(host='127.0.0.1', port=12345):
    with socket(socket.AF_INET,
    socket.SOCK_DGRAM):

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} {end-start, 3+3 seconds}
except socket.timeout:
    print("Request timed out")
if __name__ == "__main__":
    ping_server()
```

output:

```
> python ping_server.py
UDP Server running on 127.00.1.12345
Received message from ('127.0.0.1', 6734)
ping.

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

19/11

Result:- Thus the above program is verified and executed successfully.

Exp: 14

Aim:- Write a code using raw socket to implement packet sniffing.

Program:-

```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, TCMP
def packet_callback(packets):
    if IP in packet:
        ip_layer = packet[IP]
        protocol = ip_layer.protocol
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst
protocol_name = ""
if protocol == 1:
    Protocol name = "ICMP"
    elif protocol == 8:
        protocol_name = "TCP"
        protocol_name = "Unknown protocol".
print(f"protocol {prot_name}")
print(f"Source IP: {src_ip}")
print(f"Destination IP: {dst_ip}")
print("_" * 50)
def main():
    sniff(prn = packet_callback, filter = "ip",
          store = 0)
if __name__ == "__main__":
    main()
```

output:-

```
> python packet
```

Result: Thus the above program is verified and executed successfully.

Exp. 15

AIM:

To analyse the different types of weblogs using utilize tool

ALGORITHM:

1. Run webalizer windows version
2. Input weblog file
3. press run webalizer

Steps:- 1. open xampp and check weather there is an already visited website if not host a website.

2. Go to browser and type "localhost/project name" to run the project.

3. Then go to xampp > webalizer > weblizer.exe

4. The open xampp > apache > logs > access and check if the file is not empty.

5. If it is empty properly run the website else continue.

6. Open terminal in open weblize. Enter weblizer.exe.

7. Then go to xampp > htdocs > weblizer > index.html

8. This file is output.

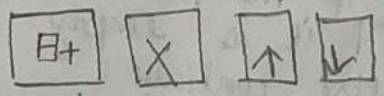
RESULT:

Thus the experiment for using webalizer for weblog analysis executed verified

Input :-

logfiles:

① C:\Users\TCS\Downloads\access-log

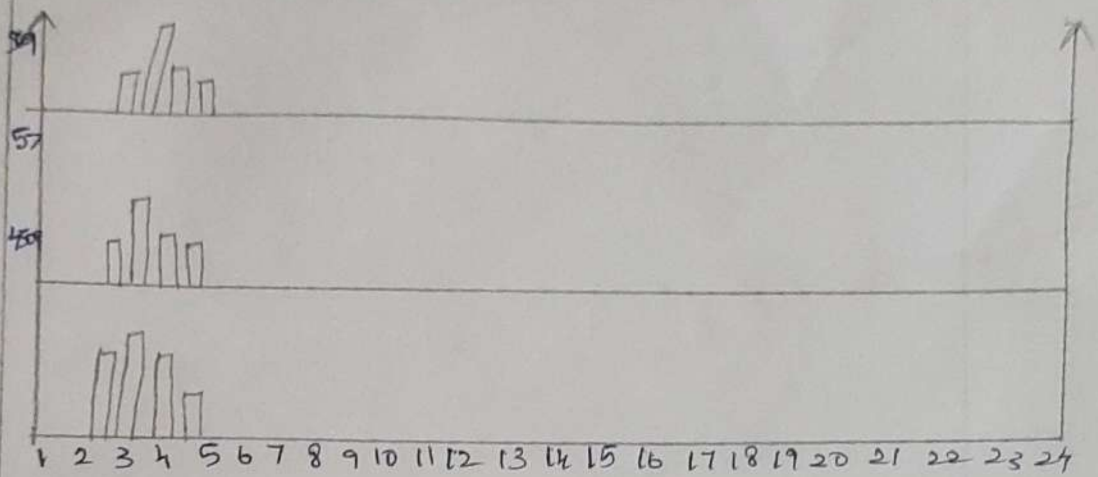


C:\Users\TCS\

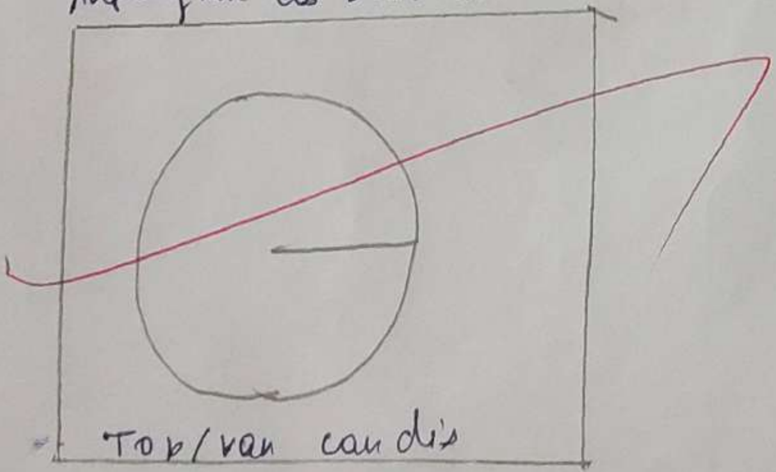
☐ clear existing directory

output:-

Entry pages	first page	errors	host	reference	src page	sis	(->)	(->)



Sum - Document Abfragin (2.B num / sarp, lent) Aufragam as down in mrut main 2001



Some step webalize

20/4

Result:- Thus the different types of web logs using utilize tool is analyzed.

completed

20/4