INDEX

NAME: L. P. Dhana Sree STD.: B.E SEC.: CSE A ROLL NO.: 220701061 SUB.: CN Obser

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	13.7.24	Study of various commands used in linux and windows	9	
2.	27.7.24	Study of network cables	10	
3)	30.7.24	Equipments on Cisco packet Tracer	10	
4		Setup and configure of LAN		
5.		Experimentation: Packet Capture tool :- Wireshark.		
6.		Hamming code in C.		
7.		Sliding Window.		
8(a)		VLAN configuration		
8(b)		Wireless configuration.		
9.		Subnetting Cisco Packet Tracer		
10(a)		Internetworking with Routers		
10(b)		Internetwork using DHCP server and Internet cloud.		
11(a)		Static Routing Protocol using Cisco		
11(b)		RIP using Cisco		

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
13.		Ping Program		
14.		Packet Sniffing		
15.		Webalizer tool.		

completed

12/11

Teacher's Sign / Remarks

Date : 13-07-2024

Ex no: 01

AIM:-

Study of various network commands used in Linux and Windows

Basic Network Commands:

1) arp -a :- Interface : 172.16.75.54 .... 0x12

Internet address	Physical address	Type
172.16.72.1	7c-5a-1c-cf-bc-41	dynamic
172.16.72.133	4c-ac-a3-65-97-f3	dynamic

2) Hostnames

DESKTOP-- C01BHT0

3) nbtstat -a

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

-a [adapter status] lists the network machines name table given its name

4) netstat :-

Active connections:

Proto	Local address	Foreign address	Status
TCP	127.0.0.1 49678	DESKTOP: C01BHT0: 49679	Established

5) nslookup :-

Default server: unknown

Address :- 172.16.72.1

pathping:- usage:- pathping [-g host list]
[-h maximum hops] [-i address] [-n] [-p period]
[-q num queries] [-w timeout] [-4] [-6]
<target_name options:-4 Force using IPV4.

Route:- Route [-f] [-p] [-4|6] command
[destination] [Mask netmask] [gateway]
[METRIC metric] [IF interface]
-6 force using IPV6.

Some important Linux Networking Commands:

a) ip:-
ip <OPTIONS> <object> <command>

(b) [root@server ~]# ip address show
1: lo: <LOOPBACK, UP, LOWER-UP> mtu 65536
qdisc sequence state UNKNOWN group
default qlen 1000
link/Loopback, 00:00:00:00:00:00 brd 00:00:00
inet 127.0.0.1/8 scope host lo..

(c) [root@server ~]# ip address add 192.168.1.254/24
dev enp2s03
To assign an IP to an interface

(d) [root@server ~]# ip link set lo up
To alter the status of the interface by
bringing the interface enp2s0 online

(e) [root@server ~]# ip link set lo down
To alter the status of the interface
by bringing the interface enp2s0

f [root@server ~]# ip link set lo promisc on

To alter the status of the interface by enabling promiscuous mode for lo.

g [root@server ~]# ip route add default via 192.168.1.254/21 dev lo

To add a default route for all address via the local gateway 192.168.1.254 that can be reached on device enp2s0

h) [root@server ~]# ip route add 192.168.1.0.124 via 192.168.1.254

To add a route to 192.168.1.0/24 via gateway at 192.168.1.254

i) [root@server ~]# ip route add 192.168.1.0/24 dev lo

To add a route to 192.168.1.0.124 that can be searched on device enp2s0.

j) [root@server ~]# ip route delete 192.168.1.0/24 via 192.168.1.254.

To delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254.

k) Display:-> [root@server ~]# ip route get 10.10.1.4

10.10.1.4 dev enp2s0 src 192.168.1.25 uid 1000 cache.

2. ifconfig:-

ens160: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500

inet 192.168.22.128 netmask : 255.255.255.0 broadcast 192.168.22.255

3) mtr:-

mtr <options> hostname | IP

a) [root@server ~]# mtr google.com.

local.localdomain (0.0.0.0)

keys: help.displaymode restart order of fields quit

Host

1) 172.16.8.1

2) Statistic - 41229.429.49 - tataid.c.co.in

3) 14250.171.162

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

b) mtr g - google.com.

No GTK support.

c) mtr -c google.com.

Localhost localdomain

keys. help displaymode restart statistic OF quit

Host :

d)

4) tcpdump:-

[root@server~]# dnf install -y tcpdump

to install -y tcpdump.

[root@server ~] # tcpdump -D

1) lo [up, Running, connected]

2] any [pseudo-device that captures on all interfaces] up running)

[root@server ~]# tcpdump -i lo

dropped privs to tcpdump tcdump: verbose output suppressed use -v[v] ... for full protocol decode.

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

10 packets captured

20 packets received by filter.

0 packets dropped by kernel

[root@server ~] # tcpdump -i lo -c 10 host 8.8.8.8

dropped privs to tcpdump.

tcpdump: verbose output suppressed, use -v[v]

for full protocol decode listening on lo (lo) type EN10MB(ethernet) snap shot length 262144 bytes

[root@server ~]# tcpdump -i eth0 sec host 8.8.8.8.

dropped privis to tcpdump.

tcpdump verbose output suppressed use -v [v]

for full protocol decode listening on lo.

[root@server ~]# tcpdump -i lo ist host 8.8.8.8.

dropped privis to tcpdump

tcpdump verbose output suppressed use v[-v].

for full protocol decode listening on lo

[root@server ~]# tcpdump -i lo net 10.1.0.0.

usage 255.255.255.0.

To capture traffic and to form a specific network using the command

[root@server ~]# tcpdump -i lo net 10.1.0.0/24

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

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

To capture only DNS port 53 traffic

[root@server ~]# tcpdump -i lo host 8.8.8.8 and port 53.

This is for a spicific host

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

To capture only HTTPS traffic.

[root@server ~]# tcpdump -i lo port root 53 and rot 25.

To capture all port except port 80 and 25.

Ping:-

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

[root@server ~]# ping google.com

PING google.com (216.58.200.142) (56) (84) bytes of data 64 bytes from maa05s10-in-f14.1e100.net (216.58.200.142): icmp.seq=46tt

[root@server ~]# ping -c 10 google.com

PING google.com (142.250.193.100) 56 (84) bytes of data -- google.com ping statistics --

10 packets transmitted, 0 received, +10 errors, 100% packet loss, time 9197ms pipe 3.

nmcli configuration

1) nmcli connection show:-

Name	UUID	Type	Device
wi	f8ac812f-cb23-4995	802-3-ethernet	enp2s0

2) nmcli connection add con name RECA4 ifname type ethernet:-

Successfully added.

3) ip address show enp2s0:-

enp2s0 :<BROADCAST, MULTICAST, UP LOWER-UP> state up, group default qlen 1000

4) ip route show default

default via 172.16.8.1 dev enp2s0 proto static metric 100 scope link src 172.16.9.155

Result :-

Thus the study of various commands in Linux and windows are studied

Ex : 1 Student observation

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

The ping command is a networking utility used to test the reachability of a host on an Internet protocol network.

2) Which command will be given the details of a hops taken by a packet to reach its destination?

Traceroute shows the sequence of routers or network nodes that data packets pass through to reach out their destination.

3) Which commands displays the ip configuration of your machine.

On windows :-

1) Command prompt (cmd) : ipconfig

2) Powershell :- get - NetIPaddress

4) Which command displays the TCP port status in your machine

On windows macOs and Linux:

netstat -an

5) Write the modify the ip configuration in a Linux machine :-

1) Use the "ifconfig" command followed by the name of your network interface and the new IP address to be changed on your computer

2) Use the "ip addr add" command followed by the new IP address and subnet mask.

3) Change the ip address in the file "/etc/sysconfig network-scripts ifcfg eth0" restart the system

Ex.No:2
Date: 20-7-24

AIM:-
Study of different types of Network Cables.

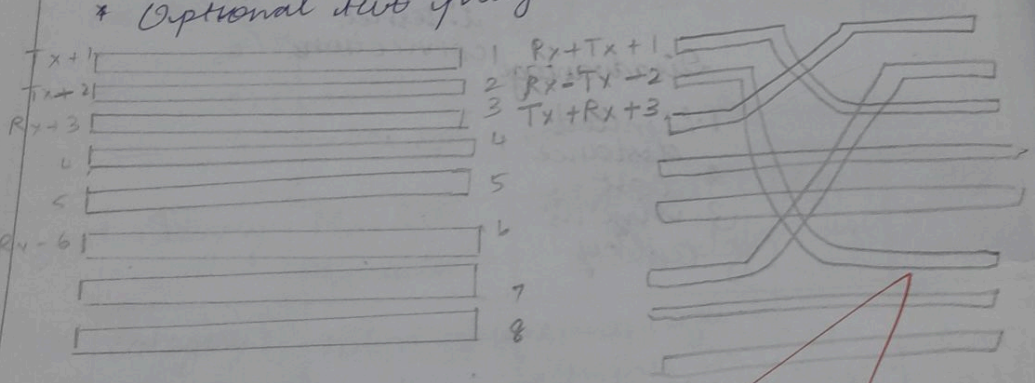
1a) Understand different types of network cables:
Different type of cables used in networking are
1. unshielded twisted pair (UTP) cable
2) Shielded Twisted pair (STP) cable
3) Coaxial cable
4) Fibre optic cable

Cable Type	Category	Max data Transmission	Advantages/ Disadvantages	Appl/ use	Image
UTP	category 3 and 5	10bps upto 100 Mbps	Advantages: * Cheaper in cost * Easy to install Disadvantages: * More prone to Electro magnetic interface and noise	10 Base - TE thernet. Fast Ethernet gigabit	
	category 5e	1 Gbps			
STP	Category 6, 6a	10 Gbps	Advantages: Shilded Faster Thum UTP Less susceptible	Gigabit Ethernet (55 m) Widely used	

SSTP	Category 7	10 Gbps	Disadvantage: Expensive Greater installation effort	Gigabit Ethernet (100m	
Coaxial Cable	RG-6 RG-59 RG-11	10-100 Mbps	Advantage: 1. High Bandwidth 2. Immune to interference 3. Versatile Disadvantage 1. Limited distance 2. Cost 3 size is bulky	speed of signal is 500m Television network High speed internet connections	
Fibre optic cable	single mode Multi mode	100 Gbps	Advantages: * High speed * High Band width 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 - CAT5 e is certified for gigabit support but CAT-5 cabling works as well, just over shorter distances
* A crimping tool. This is an all-in one networking tool shaped to push down the shielding off the cables
* Two RJ45 plugs
* Optional two plug shields.



Difference between crossover cable and straight cable.

Straight through network cable: both sides should be A cross over cable: One side A, one side B

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

Step:2.
Next, strip approximately 1.5cm of cable shielding from both ends.

Step:3 After you will need to untangle the wires. There should be four "twisted pairs". referencing back to the sheet, arrange them from top to Bottom. One end should be in arrangement A and other in B.

Step:-4
Once the order is correct, bunch them together in a wire and if there are any that stick out farther than others, strip them to back to create an even level. The difficult aspect is placing these into that RJ45 plug without messing up the order.

Step:5. Next, push the cable right in the notch at the end of the plug needs to be just over the cable shielding, and if off too much shielding

Step:-6 After the wires are securely sitting inside the plug and insert and push down

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

Result:-
Thus the different types of network

Ex: 2 student observation

1) What is the difference between cross and straight cable

Straight cable	cross cable
1) The wiring of both ends of the cable is identical	1) The transmit and receive wires are crossed on end of cable
2) It is used for connecting different devices	2) It is used for connecting similar device
eg:- PC to switch/router	eg:- PC to PC

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

A cross cable is used to connect two PC 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 out the category of twisted pair cable used in your LAN to connect the PC to network socket?

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

CAT5: supports upto 100Mbps

CAT5C: supports upto 96ps

CAT6:- Supports upto 1096ps for shorter distances

5) Write down your misunderstanding, challenges faced and output received while making twisted pair cross:-

understanding: Making a twisted pair

Ex no:- 3

Aim:-

To study the packet tracer tool installation and user interface overview

a) To understand environment of CISCO packet tracer to design simple network

a)

A simulator, as the name suggests, without the need for dedicated equipment. Packet Tracer is an exciting network design, simulation and modelling tool.

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

2) It helps us you to practice your network configuration and troubleshooting skills via computer or an android or iOS based mobile device.

3) It is available for both the Linux and Windows desktop environment.

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

Installing Packet tracer :-

To download Packet tracer, go to https://www.netcad.com and log in with your cisco Networking Academy credentials. Then click on the Packet Tracer graphic and download the package appropriate ...ating system.

Linux :-

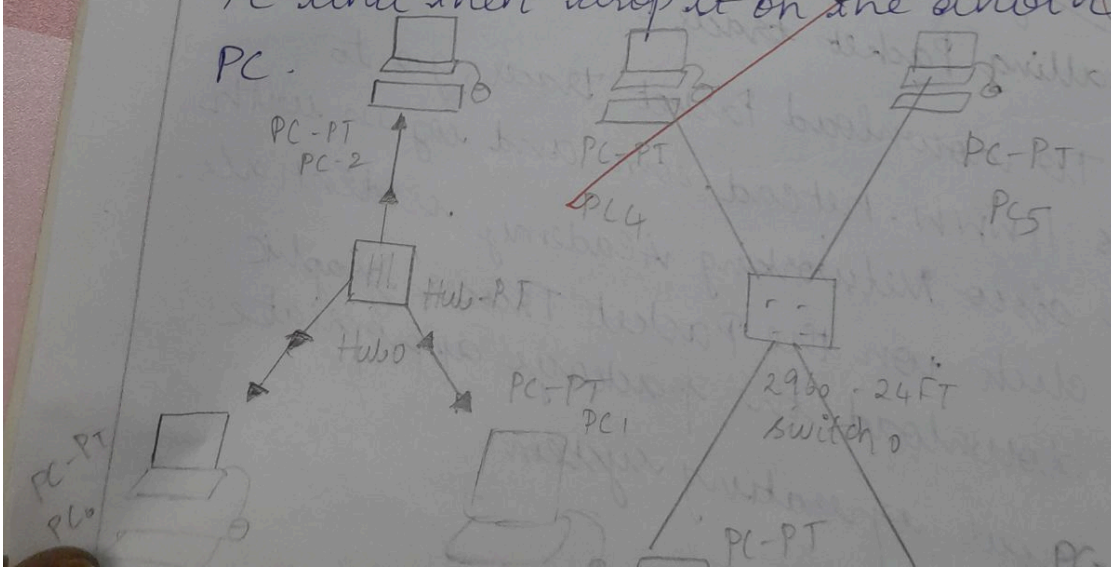
Linux users with an ubuntu/debian distribution should download the file for ubuntu and those using fedora/ubuntu download the file for fedora. Grant executable permission to this file by using chmod and execute it begin the installation

chmod + x PacketTracer601-i386-installer rpm.bin */PacketTracer601-i386-installer-rpm.bin.

Click on the PC's connected to hub, go to the desktop tab, click on IP configuration, out enter an IP address and subnet mask. Here default gateway and DNS server information not need as there are only two end devices in network.

Click on PDU message icon from the common tool bar.

a) Drag and drop it one of the PC and then drop it on the another PC.



1. Menu bar :- This is a common menu found in all software applications, it is used to open, save, zoom, undo and redo and entering network.

2) Main toolbar :- This bar provides shortcut icons to menu options such as zoom, open etc and on the right-hand side is an icon for entering network information for the current network

3) Logical/physical workspace tabs :-
These tabs allow you to toggle between the logical and physical work are as.

4) Workspace :- This is the area where topologies are created and simulations are displayed.

5) common tools bar :- This toolbar provides control for manipulating topologies such as select, move layout, place, note, delete inspect resize, shape and add simple/complex PDU.

6) Real time/simulation tabs :- These tabs are used to toggle between the real and simulation modes. Buttons are also provided to control the time and capture the packets

d) Analyse the behaviour of network devices using CISCO PACKET TRACER simulator :-

1. From the network component box/ click and drag and drop the below components :-

a) Generic PCs and one HUB
b) A Generic PCs and one switch.

2) Click on connections
a) Click on copper straight - through cable
b) select one of the PC and connect it to HUB using the cable. The link LED should glow in green, indicating that the link is up. similarly connect remaining 3 PCs to the HUB.

3) Click on the PC connected to Hub, go to the desktop tab, click on IP configuration and enter an IP address and subnet mask. Here the default gateway and DNS server information is not needed as there are only two end devices in map

click on the PDU from the comment tool bar.
a) Drag and drop it on one of the PC.
(and then drop it on another PC.

) Repeat step #3 to step #5.

) Observe the Hubs and switches and how it is forwarding the PDU write your observation and conclusion

Results:
Thus the overview of the Packet tracer tool installation and use

Student observation- Practical-3

a) Behaviour of switch and hub in forwarding Packets.

switch:-

Packet forwarding:- Uses Mac addresses to forward packets only to the intended recipient.

efficiency:- More efficient, as it reduces unnecessary traffic on the network.

Collision domains:- Each port on switch is its own collision domain, reducing collisions and improving performance

Hub:

Packet Forwarding:- Broadcast packets to all connected devices regardless of destination.

Efficiency:- Less efficient as it increases unnecessary traffic

b) Network topology implemented in our college:

observation:- Look at how devices like computer switches, routers and hubs are interconnected.

drawing:- draw the topology in your observation book based on your finding.

Common network topology:-

star:- All devices are connected to central hub.

Bus:- All devices share a single communication line.

Mesh:- Devices are interconnected with multiple paths between them.

Exercise:4
Date:

Aim:-
To setup and configure a LAN 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 office building, school or home. It allows users to share resources like data printers and internet access. The LAN switch acts as a central device, managing and directing communication between connected devices, ensuring fast and secure data transfer.

How to setup a LAN:-

1) Plan and Design: Choose the appropriate network topology based on your requirement and equipment placement.

2) Gather Equipment: Use 4 computers, a switch and 4 Ethernet Cables

3) Connect devices:-
Connect each computer to the switch using ethernet cables. Simply plug one end of each cable into a computer and other end into switch.

4) Assign IP address:-
* Log into each computer as an administrator
* Go to network and internet connections
* Right click on Local area connection / ethernet select Properties

* Choose IP (TCP/IPV4) and then click properties
* Select the "use following IP address and enter a suitable IP address and enter a suitable IP address for each device.

PC1: IP: 10.1.1.1, subnet mask: 255.0.0.0,
PC2: IP: 10.1.1.2, subnet mask: 255.0.0.0
PC3: IP: 10.1.1.3; subnet mask: 255.0.0.0
PC4: IP: 10.1.1.4, subnet mask: 255.0.0.0.

Configure ~~to~~ the computer network:-

1) Connect computer to switch:- Use an ethernet cable to connect your computer to a switch.
2) Log into the web interface: Open a browser enter the switch's IP address and log in with username and password.
3) Configure basic settings: set the IP address for the switch 10.1.1.5 and subnet mask to 255.0.0.0

Check connectivity:-

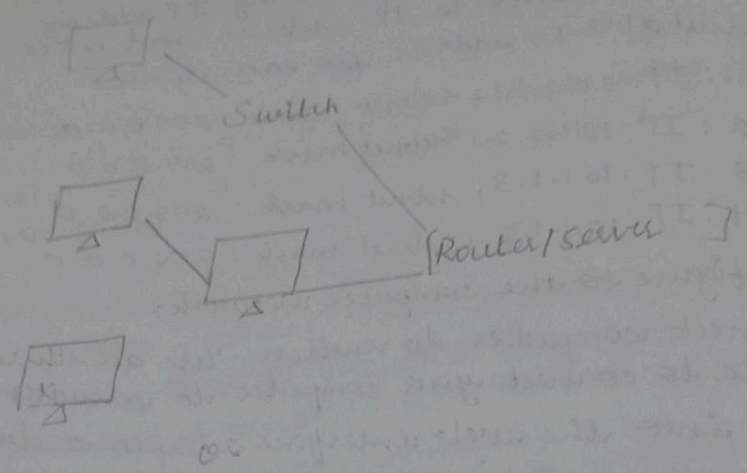
Use the ping command in the command prompt to ensure connectivity between switch and other devices.

Share a folder:-

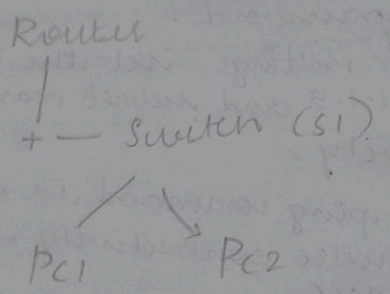
* select the folder, go to properties and click the sharing tab.
* Share the folder with everyone on the same LAN

Access the shared folder.

Try accessing the shared folder from other components on the network.



or



IP configuration:

PC1: IP: 10.1.1.1, subnet mask: 255.0.0.0

PC2: IP: 10.1.1.2, subnet mask, 255.0.0.0

PC3: IP: 10.1.1.3, subnet mask, 255.0.0.0

PC4: IP: 10.1.1.4, subnet mask, 255.0.0.0

20/8

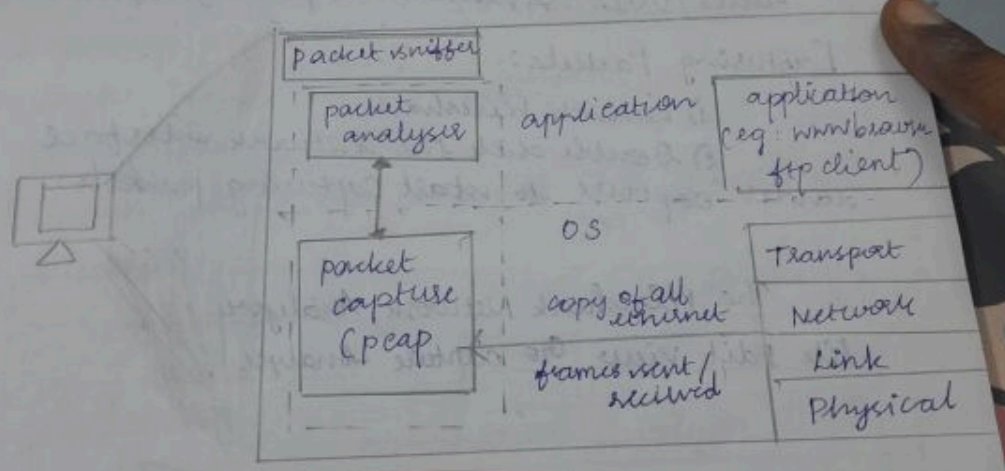
Result:

Thus the setup and configure of LAN is done successfully

Exercise: 5
Date:

Aim:-
Experiments on Packet capture tool: Wireshark.

Packet sniffer: captures messages sent/received by your computer.
Functions: Stores and displays protocol fields in messages
* Passive: Does not send packets or received packets addressed to it. only receives copies of all packets.



Wireshark:-
Wireshark is a network analysis tool that captures and displays packets in real time. It features filters coloring and detailed description capabilities to analyse traffic and troubleshoot issues.

Features:-
* Capture network traffic
* Decode packet protocols with

* Define captures and displays filters
* Monitor smart statistics
* Analyse network problems
* Browse traffic interactively.

Users:-
network administrators - Troubleshoot network issues
network security engineers - Examine security problems
Developers - Debug protocol implementations
Individuals: Learn network protocol internals

Getting Wireshark:-
Windows/macOS:- Download from the official website
Linux/UNIX:- Available in package repositories

Capturing Packets:-
1) Launch Wireshark
2) Double click the network interface under capture to start capturing packets

The Wireshark Network Analyzer
File Edit View Go capture Analyze

Welcome to Wireshark

Capture
using this filter...
Enter a capture filter

VirtualBox Host-only network
Wifi
VmWare Network Adapter VmNe
Ethernet 2
Ethernet -

Wireshark interface Overview:-

1) stop capturing traffic
   click red 'Stop' Button near left corner

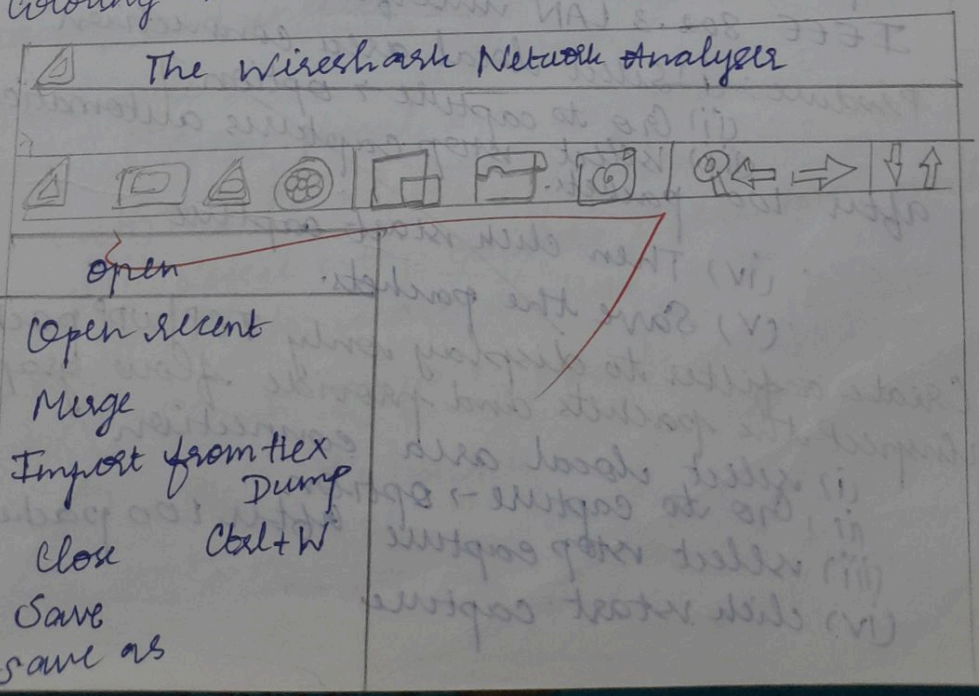
2) "Packet List" Pane:
   * Displays all packets in current capture file
   * Each line corresponds to one packet
   * selecting a line shows more details in 'Packet details' and 'Packet Bytes' panes

3) The "Packet Details" Pane:
   * Shows details information of selected packet.
   * Displays protocols and protocol fields in a format called tree.

4) The "packet Bytes" Pane:-
   Shows the selected packet's data in hexdump style

Light Purple:- TCP traffic
Light Blue:- UDP traffic
Black:- Packets with errors (eg:- out-of-order delivery)

To view or customize color codes:- Go to view > Coloring rules.



Sample:-

* Use sample files to practice in wireshark open file via → open
* Save your captures with File → Save for later review.

Filtering Packets:-

→ Apply filters to focus on specific network traffic.

→ Close other apps to isolate traffic for analysis

Type a filter, press Enter. eg :- 'dns' for DNS packets wireshark auto complete,

Use analyze & display filters to pick or save filters. See the docs for more info. Right click a packet, choose follow > TCP stream to see the full conservation. Use follow for other protocol too.

Capturing and Analysis Packets using Wireshark tool:-

To filter, view capture packets, capture 100 packets, from the ethernet, IEEE 802.3 LAN interface and save it

Produre:- (i) Select local area connection

(ii) Go to capture → option

(iii) Select stop captures automatically after 100 packets.

(iv) Then click start capture

(v) Save the packets.

i) Create a filter to display only TCP/UDP packets, inspect the packets and provide flow graph

(i) select local area connection.

(ii) Go to capture → option.

(iii) select stop capture after 100 packets

(iv) click start

save the packets

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

(i) Go to capture → option

(ii) Select stop capture after 100 packets

(iii) Then click start capture

(iv) Search ARP packets in search bar.

(v) Save the packets.

3) Create a filter to display only DNS Packets and provide flow graph

(i) Go to capture → option

(ii) Select stop capture after 100 packets

(iii) Click start capture.

(iv) Search DNS packets.

(v) See flow graph by statistics → flow graph

(vi) Save the packets.

4) Create a filter to display only HTTP packets

PROCEDURE :-

(i) Go to capture → option

(ii) select stop after 100 packets

(iii) click start capture.

(iv) Search HTTP packets.

(v) Save the packets.

5) Create a filter to display only IP | TCP packets and inspect the packets.

Procedure :-

(i) Select local area connection

(ii) Go to capture → option

(iii) Select stop capture after 100 packets.

(iv) Click start capture.

(v) Search ICMP | IP packets

(vi) Save the packets.

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

Procedure:-

(i) Go to capture → option
(ii) Select Stop after 100 packets
(iii) Click start capture.
(iv) Search DHCP packets.
(v) Save the packets.

Student observation:-

1) What is promiscuous mode?

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

2) Does ARP packets have a transport?

No. ARP packets do not have a transport layer header because it operates at the data link layer of the OSI model & is used to map IP addresses to MAC addresses.

3) Which transport layer protocol is used by DNS?

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

4) What is the Port number used by HTTP protocol?

The default port number

port is 443.

5. What is broadcast IP addresses

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

Result:-

Thus the wireshark is executed and done successfully.

Ex no: 6
Date:

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

Error correction at Data link layer:-
Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is transmitted from sender to receiver.

Create sender program with below features;
→ Input → text to any length
→ convert text to binary.
→ Apply Hamming code concept on the binary data and add redundant bits to it
→ Save this output in a file called channel.

Create receiver program with below features.
→ should read input from channel file
→ Apply hamming code on the binary data to check for errors.
→ Else remove the redundant bits and convert binary data to ascii.

Code:-

```
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 bits needed
for error correction.

def binary_to_text(binary):
    chars = [binary[i:i+8] for i in range(0,
len(binary),8)]
    return ''.join([chr(int(c,8) ...
```

```
# Function to calculate redundant bits needed
for error detection.
def calc_redundant_bits(m):
    r=0
    while(2**r<m+r+1):
        r+=1
        return r
# Function to insert redundant bits into the
data
    def pos_redundant_bits(data,r):
        j=0
        k=0
        m=len(data)
        res=''
# Adding redundant bits at positions that are
powers of 2.
        for i in range(1,m+r+1):
            if i==2**j:
                res=res+'0'
                j+=1
            else:
                res=res+data[k]
                k+=1
        return res
# Function to calculate Parity bits.
    def calc_parity_bits(arr,r):
        n=len(arr)
        arr=list(arr)
        for i in range(r):
            parity=0
            position=2**i
            for j in range(1,n+1):
                if j&position:
                    parity^=int(arr[j-1])
            arr[position-1]=str(parity).
```

```
    return ''.join(cars)
# Function to detect and correct errors.
def detect_and_correct(data, r):
    n = len(data)
    res = 0
# Calculate parity bits
    for i in range(r):
        parity = 0
        position = 2**i
        for j in range(1, n+1):
            if j & position:
                parity ^= int(data[j-1])
        if (parity != 0):
            res += position
    if res != 0
        print(f"error detected at position: {res}")
        data = list(data)
        # correct the error
        if res <= n:
            data[res-1] = '0' if data[res-1] == '1' else '1'
            print(f"error corrected at position: {res}")
        else:
            print("error position out of range. No correction performed.")
        corrected_data = ''.join(data)
        return corrected_data
    else:
        print("No error detected.")
        return data
```

```
# Function to remove redundant bits
def remove_redundant_bits(data, r):
    j=0
    original data =''
    for i in range(1, len(data)+1):
        if i == 2**j:
            j+=1
        else:
            original_data += data[i-1]
    return original_data.

# Function to introduce an error in the data
def introduce_error(data, position):
    if position<1 or position>len(data):
        print("error position is out of range.")
        return data
    data = list(data)
    # flip the bit at the specified position

    (1-based index)
    . data[position-1] = '0' if data[position-1]
    == '1' else '1'
    print(f"Introduced error at position:
{position}")
    return ''.join(data)

# Sender program
def sender(text):
    binary-data = text_to_binary(text)
    m = len(binary-data)
    r = calc_redundant_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 ass
# Receiver program.
def receiver (data) :
    r = calc - redundant - bits (len (data))
    corrected - data = detect - and - correct
(data, r)
# Redundant bits Removal
original - data = remove - redundant - bits
(corrected - data, r)
ascii - output = binary - to - text
(original - data)
print (f "decoded text : {ascii - output}")

# Main program :
if __name__ = "__main__"
```

Result: The program for hamming code is executed successfully.

Output:

Enter the text to be encoded: Dharasree.

Redundant bits : 7

Placing redundant bits at positions : 1 2 4 8 16 32 64

Parity Bits : [Position 64 : 0]

Sender output [binary with redundant bits) :

1001110101000110010000110000101010110011

1000010111001101110010001100101011001

Introduced error at position : 2

Redundant bit : 7.

Binary with error : 1100110101000110010000111

0000101011100110000101110011011100100011

1001010110010 1.

error detected position : 2

error corrected position : 2

Binary after error detection.

1001110101000110010000101010111001100000

1011100110111001000110010101100101

Decode :- Dharasree.

Ex no:7 Sliding Window Protocol Date: 14/9/24

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

Procedure:

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

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

3. After sending, the window "slides" forward as acknowledgements are received.

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

5) Lost or corrupted frames are retransmitted, ensuring reliable and ordered delivery.

```
import time
import random
class Frame:
    def__init__(self, frame_no, data):
        self.frame_no = frame_no
        self.data = data
        self.acknowledged = False
def send_frames(frames, window_size):
    print("\n --Sending Frames ---")
    for i in range(window_size):
        if i < len(frames) and not frames[i].
acknowledged:
            print(f"Sent Frame {frames[i].frame_no
}: {frames[i].data}")
    print("Frames sent, waiting for acknowledgeme
...\n")
def receive_frames(frames, window_size):
    print("\n -- Receiving Frames --")
    for i in range(window_size):
        if i < len(frames) and not frames[i]
.acknowledged:
            if random.random() < 0.2:
                print(f"Received Frame {
frames[i].frame_no}: {frames[i].data} [ERROR]")
                frames[i].acknowledged = False
            else
                print(f"Received Frame {frames
[i].frame_no}: {frames[i].data} [OK]")
                frames[i].acknowledged = True
```

```
def sliding_window_protocol():
    window_size=int(input("Enter windowsize :"))
    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("\n Resending unacknowledged
              frames....\n")
            time.sleep(2)
    print("\n All frames sent and unacknowledged!")

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

Output:

```
Enter Window Size : 5
Enter a message to send : DHANA
    ---- Sending Frames ---
    Sent Frame 0 : D
    Sent Frame 1 : H
    Sent Frame 2 : A
    Sent Frame 3 : N
    Sent Frame 4 : A
```

Frames sent, waiting for acknowledgement

-- Receiving Frames - - -

Received Frame 0 : D [Received]

Received Frame 1 : H [Received]

Received Frame 2 : A [Received]

Received Frame 3 : N [Received]

Received Frame 4 : A [ERROR]

-- Resending Frames - - -

Received Frame 4 : A [Received]

All frames are sent and acknowledged

5/11.

Result:-

Thus the code for Flow control (sliding control) is executed successfully

Ex No: 8 (a)
Date:

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

Step 1: Setup the topology:-

1. Open cisco packet tracer
2. Drag and Drop switches (Cisco 2969) and 2 PCs into the Workspace.
3) Use copper Straight-through cables to connect the PC's to the switch using their Fast ethernet ports.

Step-2) Assign IP addresses to PC's

1. Click on each PC, go to the Desktop tab and click IP configuration.
2. Assign IP addresses to each PC:-
   - PC1 - IP - 192.168.1.10. subnet mash: 255.255.255.0
   - PC2 - IP - 192.168.2.10. subnet mash: 255.255.255.0

Step: 3 VLAN Configuration on the switch:-

1. Click on the switch and open the CLI (command Line interface).
2. Enter the following commands to create VLAN's and assign the two ports to different VLAN's

Create VLAN's:-

Switch> enable

Switch # configure terminal

Switch # VLAN 10

Switch(config-vlan)# name

VLAN-10

Switch(config-VLAN) # name

Switch (config) # VLAN 20

VLAN-20

Switch(config - VLAN)#exit

Assign switch Ports to VLANs:-

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

Switch(config) # interface fast ethernet 0/1

Switch (config-if) # switch Port mode access.

Switch (config-if) # Switch Port access vlan 10

Switch(config-if) # exit

For PC2 Connected to Fast ethernet 0(2) assign Pt to VLAN 20:

Switch (config) # interface fast ethernet 0/2

Switch(config-if) # switch-port mode access

Switch (config-if) # switchport access vlan 20.

Switch (config-if) # exit.

Step: 4 Verify VLAN configuration:-

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

Switch # show VLAN brief.

Step: 5 Test connectivity :-

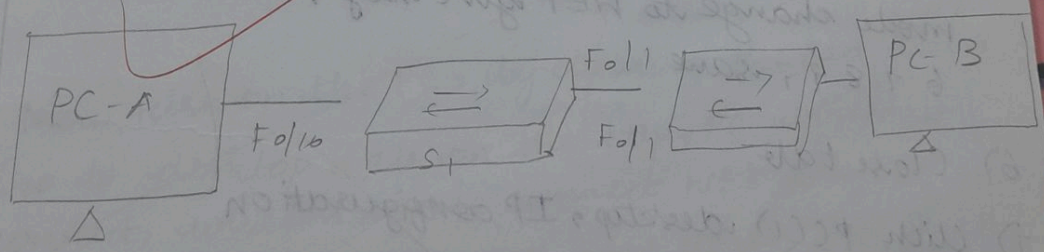
1) Go to PC1 . open the command prompt. and Ping PC2's IP address (192.168.2.10).

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

Step: 6 save the configuration.

Finally save the configuration using.

Switch # copy running-config startup config.

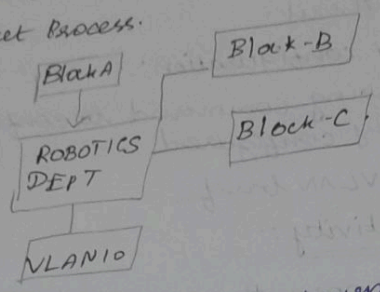


Packet Tracer - Configure - VLAN's Trunking

- Physical Mode Topology

b) Configuration of wireless LAN using a CISCO

Packet Process.



1) Click network devices down Click wireless devices select wireless Router in center

2) Add B PC'S

click End devices

3) Click router GUI tab check given up IP address is (192.168.0.1) Click disabled & save settings.

4) Come up again click wireless, wireless settings change network name (SSID to home network save settings.

5) Come up, click wireless settings, security mode change to WEP give key 1 as 0 1 2 3 4 5 6 7 8 9, save

6) Close tab

7) Click PC(1) desktop, IP configuration

IP address : 192.168.0.5

Subnet mask : 255.255.255.0

default gateway : 192.168.0.1 close tab.

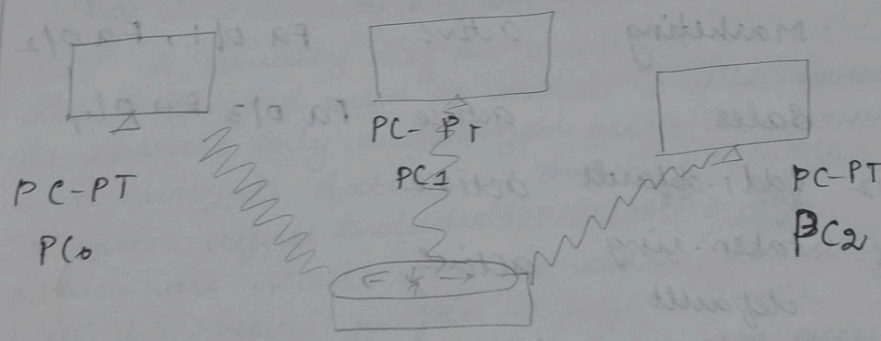
8) Go to next PC(2) desktop, IP configuration

IP address: 192.168.0.6
Subnet mask : 255.255.255.0
Default gateway: 192.168.0.1 close tab.

9) Go to next PC(s) desktop, IP configuration
IP address:- 192.168.0.7.
Subnet mask :-255.255.255.0.
Default gateway: 192.168.0.1 close tab.

10) Again click PC(1) PC wireless → error will come, Close it

11) On top, go to physical tab, Turn off the PC by clicking the red Button



12) Now turn on the PC, by clicking the red Button. Go to desktop tab on the top, PC wireless, connect, wireless mode, connect WEP key 1=> 0123456789. connect now when you click the link information it will show you have successfully connected. close tab

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

(4) PC(3) → (Repeat step (10, 11, 12))

14) to check the connectivity whether all is connected click any PC command prompt any IP address given during this process.

Ping 192.168.0.6 enter

OUTPUT:

a)

VLAN	NAME	status	Ports
1	default	active	Fa 0/6, Fa 0/6, Fa 0/7, Fa 0/8, Fa 0/24, Gig 0/1, Gig 0/2
10	Marketing	active	Fa 0/1, Fa 0/2
20	Sales	active	Fa 0/3 Fa 0/4
1002	fddi-default	active	
1003	token-ring default	active	

b) Ping 192.168.0.6

Pinging 192.168.0.6 with 32 bytes of data

Request timed out.

Ping Statistics for 192.168.0.6

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

Result:-

Thus to simulate virtual LAN configuration using CISCO packet simulator and configuration of wireless LAN using CISCO Packet Tracer is done and executed

Student observation

1) What is SSID of a wireless router?

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

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

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

3) Configure a simple Wirless LAN in your LAB using a real access point and write down the configurations in your notebook.

1) Setup access point (AP):

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

2) Access AP configuration:

- In a browser, enter the AP's default and log in with default credentials

3) Configure SSID:

Set your network name

Enable SSID broadcast

4) Set Wireless security:
- Choose WPA2-PSK or WPA3 for security
- Set a password

5) Configure wireless channel:
- Set the channel to Auto or manually select a channel
- Choose the band 2.4GHz or 5GHz

6) Enable DHCP:
- Enable DHCP if required to assign IP address

7) Save Settings :- and restart the AP

8) Test Connection. by connecting devices to the SSID using the password.

Notebook Summary:

1) SSID: Lab_WLAN
2) Security: WPA2-PSK
3) Key: Lab@12345
4) Channel: Auto
5) Band: 2.4GHz or 5GHz
6) DHCP: Enabled

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

Ex : 9
Date: 9/10/24

Aim:
Implementation of SUBNETTING IS CISCO PACKET TRACER simulation.

Steps:
1) Create a network topology
2) • Open a Cisco Packet Tracer
• Click on New > Network > Generic to create a blank topology.

2) Add devices
• add the following devices.
• 2 routers (R1, R2)
• 2 switches (S1, S2)
• 10 PCs (5 for each subnet)

• Connect the devices.
• Use the appropriate cables to connect
• R1 to S1
• R1 to R2
• S2 to R2

3) Subnetting configuration:
• Network address: 192.168.1.0/24
• Subnet mask: 127 (provides ~~and~~ 8 subnets host addresses each

4) IP addressing scheme:
• Router R1
• Gigabit Ethernet 0/0: 192.168.1.1
• Gigabit Ethernet 0/1: 192.168.2.2

Switch S1

• Fast ethernet 0/1: 192.168.1.0/27.

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

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) Configuring the devices:

- open CLI on Router R1 and router enable
- Configure terminal.
- Interface Gigabit ethernet 0/0
- IP address 192.168.2.1 255.255.255.224

No Shutdown

- Exit

- Switch configuration
- open CLI on switch S1 & enter enable

configure terminal

Interface fastethernet 0/1

Interface mode access

Switchport mode access

interface fast ethernet 0/2
switchport mode access
exit

- PC configuration:
  - Right-click on each PC and select config

Enter:

- IP address, subnet mask (255.255.255.224) default gateway (Router (1D)

6) Testing the Networks:

- Open the command prompt on each PC
- List the ping command to check connectivity between PC's 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)

7) Conclusion:

If all pings are successful your subnetting and network configuration in CISCO packet tracer is functioning correctly.

Student observation:

1) Write down your understanding of subnetting.

Subnetting is the practice of dividing a larger network into smaller, manageable sub-networks to improve performance and security. It uses a subnet mask to define the network and host portions of an IP address.

2) What is the advantage of implementing subnetting within a network?

1) Improved performance: Reduces Broadcast traffic and congestion.

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

3) Find out whether subnetting is implemented in your college. If Yes, draw and list down subnets used with ip address

1) Main network:
- Network address: 192.168.0.0/22
- Total address: 1024

Subnet used:

Subnet name	Subnet Address	Subnet Mask	IP Router	Total host
1) Academic departments	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

Internet
↓
Router

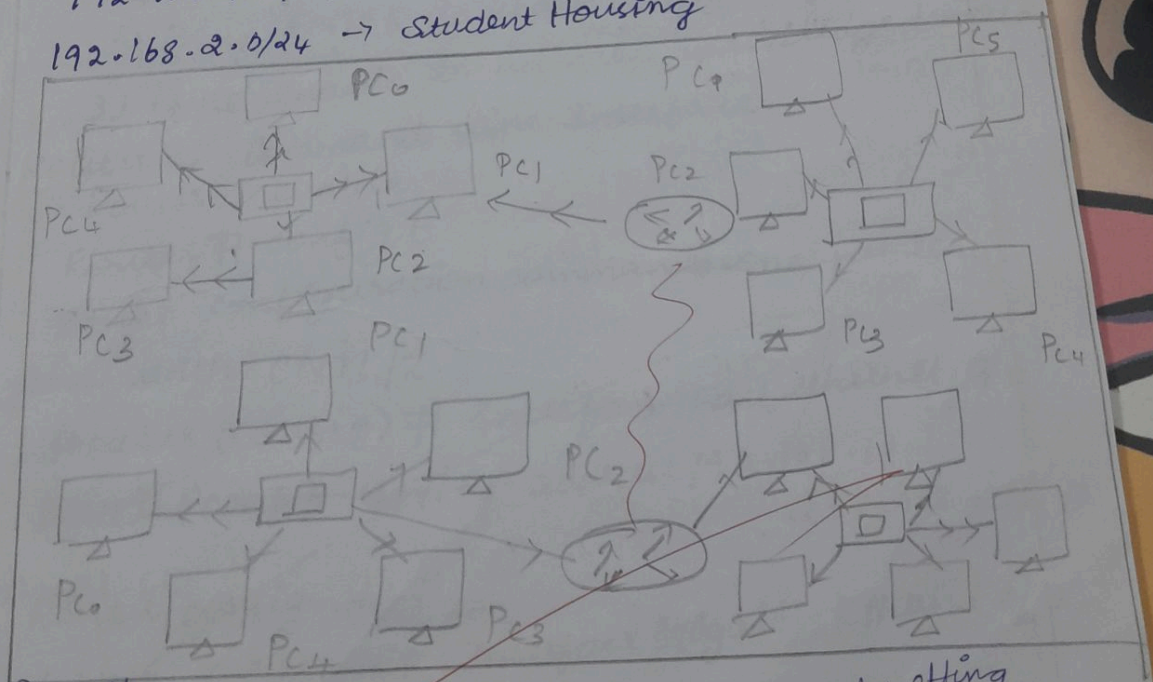
Switch A — Switch B

PC1 (192.168.0.2) | PC2 (192.168.0.3) | PC3 (192.168.1.2) | PC4 (192.168.1.3)

Printer (192.168.0.10) | Server (192.168.0.20) | Student 1 (192.168.2.2) | Student 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



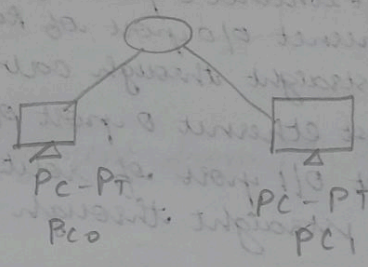
Result :

Thus the implementation of subnetting in CISCO Packet Tracer simulator is done and executed successfully

18/11

Exp: 10
Date:
a) Internetworking with routers in CISCO Packet tracer simulator

Aim: In this network, a router and 2 PC's are used. Computers are connected with routers using a copper straight-through cable. After forming the network, to check network connectivity a simple PDU is transferred from PC0 to PC1



Procedure:

step: 1

1) select router and open CLI
2) Press ENTER to start configuring router 1.
3) Type enable to activate priviledged mode

Router 1 Command line Interface

Router > enable

Router # config t

Enter configuration commands one per line, end with CNTL/Z.

Router (config) # interface Fast ethernet 0/0

Router (config-if) # ip address: 192.168.10.1 255.255.255.0

Router (config-if) #

%. LINK-5-CHANGED: Interface Gigabit Ethernet 0/0 changed state to up

Router (config-if) # Interface Fast ethernet 0/1

Router (config-if) # ip address 192.168.20.1 255.255.255.0

Router (config-if) # no-shutdown

1. Assign IP addresses to every PC in network
2. select the PC, go to desktop & select I.P configuration and assign an IP address default gateway subnet mask
3) Assign the default gateway of PC0 as 192.168.10.
4) Assign the default gateway of PC1 as 192.168.20.1

Step: 3 (Connecting PC's with router)

1. Connect Fast ethernet 0 port of Port 0 of PC0 with Fast ethernet 0/0 port of Routest using a copper straight through cable

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

Router configuration table

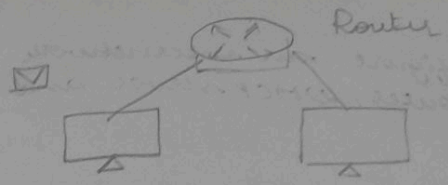
Device name	IP address Fast ethernet 0/0	Subnet mask	IP addr Fast ethernet 0/1	Subnet mask
Router 1	192.168.10.1	255.255.255.0	192.168.20.1	255.255.255.0

PC configuration table:

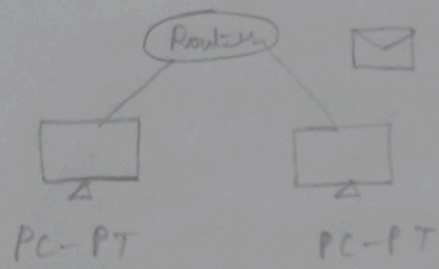
Device Name	IP address	Subnet Mask	Gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	192.168.20.2	255.255.255.0	192.168.20.1

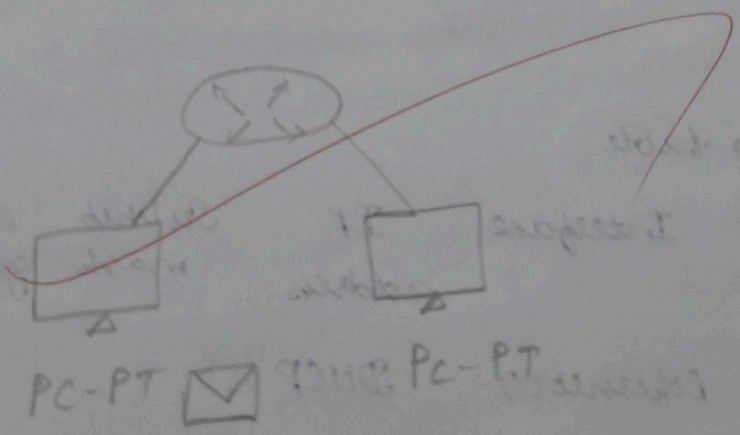
Designed network Topology:

Sending a PDU from PC0 to PC1



Acknowledgement from PC1 to PC0:



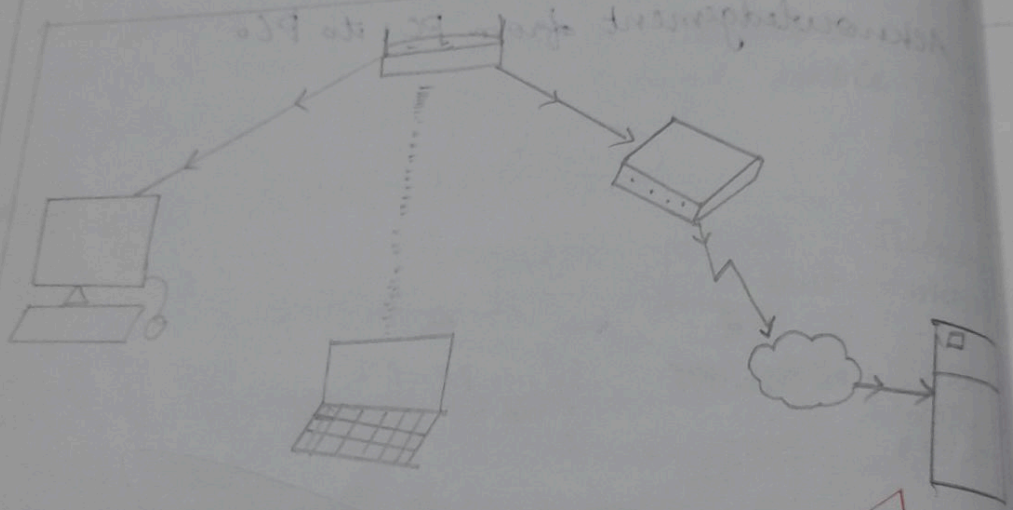


Practical-10

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

Step:1
Components Required

- Wireless Router
- DHCP Server
- Internet Cloud
- End devices.
- Ethernet cables.



Addressing table:

Device	Interface	IP address	Subnet mask	default gateway
PC	Ethernet(7)	DHCP		
Wireless router	LAN	192.168.0.1	255.255.255.0	
Wireless router	Internet	DHCP		

Step by step configuration:

Step: 1 Setup the wireless Router

- Physical Setup:
  - Connect the internet source (ICP) to the router WAN port
  - Connect any wired devices via LAN ports on the Router.
  - Access Router Admin Panel
    - Connect a device to the router
    - Open a web browser and type the default IP address of the router.
    - Login using the default credentials.

Enable wireless Networking.

* Go to the wireless settings
* Set the SSID (network name) and password for the Wifi network.
* Choose the encryption method, (WPA2 is recommended).

Step: 2 Configure the DHCP server.

* Most modern routers have a built-in DHCP server that assigns IP address to devices.
* Enable DHCP under the LAN settings
* Set the IP address range (for example, from 192.168.1.2 for 192.168.1.100).
* Specify the lease time for IP address
* Optionally, configure DHCP reservations for specific devices if needed

Step: 3 Internet Cloud Setup:

The internet Cloud represents your connection to the wider internet. If you

are simulating the network, you can use Cloud devices in simulation software like Cisco Packet Tracer or GNS3.

- For a real-world setup, make sure your router is connected to the ISP modem or directly to the Internet provider.
- Verify the WAN connection type in the network settings.

Step: 4 Configure devices.

- For wireless devices (eg: Laptop, phones). connect them to the wireless network using the SSID and password you set earlier.
- The devices should receive an IP address automatically from the DHCP server.

Step: 5

Ping test: from a connected device, open the command prompt or terminal and ping an external IP to verify internet connectivity.

- Check IP address: Run ipconfig / ifconfig to check if the device has received a correct IP address from the DHCP server.

Step: 6

Port forwarding: If certain services need to be accessed from the internet, you may need to

Considerations for the network Design:

Security: Ensure the Wifi network is secured with WPA2 encryption and use a strong password.

Backup: In larger networks, a dedicated DHCP server might be needed for reliability

Scalability:- Depending on the number of devices, switches or additional access points might be required to extend the network.

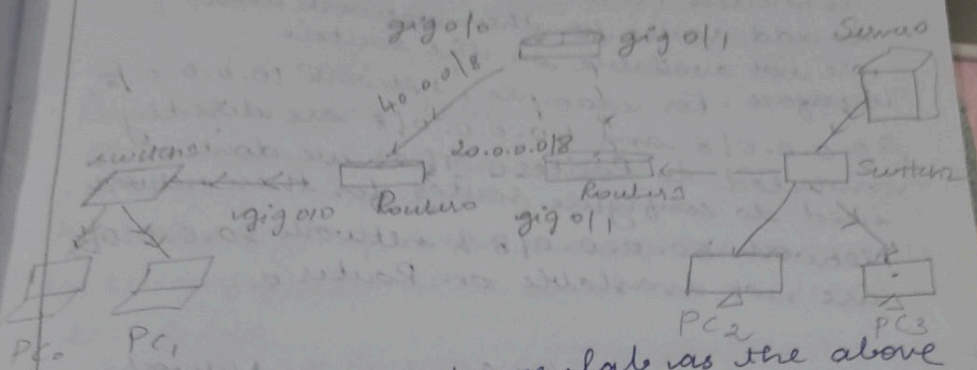
Result:

Thus the design and configuration of an internetwork using wireless router is executed successfully.

Practical 11 (a)

Aim a) Simulate static Routing configuration using a Cisco Packet Tracer.

Setting a practice lab.



Create a packet tracer lab as the above diagram. Packet tracer lab with initial IP configuration.

If we specify 2 routes to reach the same destination, the router automatically selects the best route and another router that has the least number of routes.

If you manually wants to select a route that the router should add to the routing table you have to set the AD value of the route lower than the routes

ip route 30.0.0.0.255.0.0.0 20.0.0.2.10

ip route 30.0.0.0.255.0.0.0 40.0.0.2.20

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

Routers automatically learn their connected networks, we only need to add routes for their network that are not available on the router's interface. For example, network 10.0.0.0/8, 20.0.0.0/8 and 40.0.0.0/8 are directly connected to Router 0. Thus we don't need to configure routes for these network. Network 30.0.0.0/8 & network 50.0.0.0/8 are not available on Router 0.

Router	Available networks on local interfaces	Networks available on other router's interface
Router 0	10.0.0.0/8 20.0.0.0/8 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 0 requirements:

Create 2 routes for the network 30.0.0.0/8 and configure the first route and second route as a backup route

Create 2 routes and the host 30.0.0.100/8 and configure the first route (via Route 2) as the main route and the second route (via route 1) as the back route.

Router > enable
Router# configure terminal
Router (config)# ip route 30.0.0.0.255.0.0.0.20.0.0.2.10
Router(config)# ip route 30.0.0.0.255.0.0.0.40.0.0.2.20
Router(config)# ip route 30.0.0.100.255.255.255.255.20.0.0.2.20
Router (config)# ip route 50.0.0.255.0.0.0 40.0.0.2.10
Router(config)# ip route 50.0.0.0 255.0.0.0.20.0.0.2.20
Router (config)# exit
Router # show ip route static.

Router 1 requirements:
create 2 routes for network 10.0.0.0/8 and configure the first route as the main route and the second route as backup route.

create 2 routes for network 40.0.0.0/8 and configure the first route (via Router 0) as the main route and second route as a backup route.

Router 2 requirement:
Create static routes for network 10.0.0.0/8 and network 30.0.0.0/8 and verify the router adds both routes to routing table.

Verifying static route: By sending ping request to network.

There routes via router 1 and via router 2. We set the first route as the main route and second route as the Backup route.

The following link provides the configured packet tracer lab of the above example.

Deleting a static route

To delete a static route.

To delete a static route, use the following steps.

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

Note down the router you want to delete

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

Result: Simulating Static Routing configuration using CISCO Packet Tracer is executed and verified.

Ex:-11b

Date

SIMULATE RIP USING CISCO PACKET TRACER.

Aim:-

To simulate RIP using CISCO Packet Tracer

Steps:-

* Drag and Drop at least 2-3 routes onto the workspace.
* Connect the routers using appropriate cables
* Add PCs to each connected network to the routers

Configure the Routers:

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

Router > enable
Router # configure terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit

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

Enable RIP on the Routers:

enter the following commands.

Router 0 (config) # router rip
Router 0 (config-router) # network 10.0.0.0

Enable RIP on the Routers:

enter the following commands,

Router 0 (config) # router rip

Router 0 (config-router) # network 10.0.0.0

Router 0 (config-router) # network 192.168.1.252

Router 0 (config-router) # network 192.168.1.248

Router: 1

Router 1 (config) # router rip

Router 1 (config-router) # network 192.168.1.252

Router 1 (config-router) # network 192.168.1.248

Router: 2

Router 2 (config) # router rip

Router 2 (config-router) # network 20.0.0.0

Router 2 (config-router) # network 192.168.1.252

Router 2 (config-router) # network 192.168.1.248

Verify the RIP configuration:-

Use the following command to check if RIP is configured and routes are being advertised.

Show ip route

Show ip protocols

Test connectivity. Ping from one PC to another across the network to ensure data can travel through the configured routes

Result:-

Thus the simulation of static routing configuration using Packet Tr...

Ex:12a Date:-

Implement echo client server using TCP/UDP sockets

1) TCP echo-client-server algorithm:

TCP server algorithm:

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

TCP-client Algorithm:

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

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 ("TCP server is waiting for
a connection").
connection.client_address = server_socket.
accept()
```

```
print(f"connected to {client address}")
try:
    while True:
        data = connection.recx(1024).
        if data:
            print(f"Received : {data.decode
('3')
        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(('localhost',
12345))

    try:
        message = input("Enter a message
to send");

    client_socket.sendall(message.encode())
    data = client.socket.recx(1024)
    print(f"Received from server:
    {data.decode}
```

```
finally:
        client.socket.close()
if __name__ = "__main__":
        tcp_client()
```

Output :-

```
> python tcp_client.py
Enter a msg to send : Hi, this is Dhanasree
Received from Server : Hi, this is Dhanasree
> python tcp_server.py
TCP server is waiting for a connection
Connected to ('127.0.0.0.1', 56983)
Received :- Hi, this is Dhanasree
```

4/4

Result :-

The program for using echo client server using TCP has been executed successfully.

Practical-12 (b)

Aim :-
To implement the chatclient server using TCP/UDP server.

Algorithm :-

Chat - server

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

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

3) For each connected clients start a new thread checking for new messages

4) If a client disconnects remove that client from the list or stop talking to that client.

5) keep running the process till the server stops.

Chat - client :-

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

2) Start a process by creating a to listen to messages from server

3) Keep asking for the new messages

4) Keep running till the user decides to quit.

Code :-
import
import
def
wh

decode

def

AEI

Code:-

```
import socket
import threading
def receive_message(client_socket):
    while true:
        try:
            message = client_socket.recv(1024).
decode('utf-8').
            if message:
                print(f"Server: {message}").
        except Exception as e:
            print(f"an error occured: {e}")
            break.

def start_client():
    client_socket = socket.socket(socket.
AFINET, 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("Yes:")
        client_socket.send(message.
encode('Utf-8').
if __name__ == "__main__":
        start_client()
```

Chat server.py:-

```
import socket
import threading
def handle_client(client_socket):
while True:
    try:
        message = client_socket.recv(1024).
decode('utf-8').
    if not message:
        break
    print(f"Received message from client
{message}")
        client_socket.send(response from
client.encode('utf-8').
    except Exception as e:
        print(f"An error has occurred.")
        break
client.socket.close()

def start_server():
    server = socket.socket(socket.AFINET,
socket.socket.SOCK_STREAM)
    server.bind(('127.0.0.1',12345))
    server.listen(5).
    print("Chat server has started on
127.0.0.1:12345")
    while true:
        client_socket, addr = server.accept()
        print(f"New connection from
{addr}")
        client_handler = threading.
Thread(target=handle, args=(client_socket))
        client_handler.start()
```

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

Output:-

```
> python tcp_clientchat.py.
You:- Hi
You:- Goodmorning
> python tcp-serverchat.py.
The message is done successfully:- "Hi"
the message is done successfully:- "Good morning.
```

Result:-

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

Practical-13

Aim:-

Implement your own ping program

Algorithm:

ping-client.py:

1) socket creation

2) then set a timeout of 2 second to ensure that if no response is received it will stop waiting and receive prints request time out.

3) Sends a 'ping' msg to specified host and port

4. It listens for a response and calculates time difference between the sending and receiving the packet.

Ping server.py:-

1) Initialise the UDP socket.

2) Bind to IP address and PORT

3) Listen for incoming messages

4) Receive data

5) Send Response.

Code:

ping-server.py:-

```
import socket
def start_server(host='127.0.0.1',
port=12345)
    s.bind(host, port))
    print(f"UDP server running on.
{host}:{port}")
while True:
    data, addr=s.recvfrom(1024)
    Print(f"Received message
{addr}: {data
```

```
    s.sendto(b'pong', 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(socket.AFINET,
socket.SOCK_DGRAM).
    ass:
        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_client()
```

Output:

```
>python ping_server.py
UDP server running on 127.0.0.1:12345
Received message from ('127.0.0.1', 5734):
>python ping client.py
Received pong from ('127.0.0.1', 12345) in
0.00 sec.
```

12/11

Result: The program to create and implement ping message has been successfully executed.

Practical-14

Aim:
Write a code using RAW sockets to implement packet sniffing.

Algorithm:
1) Install python and scapy
2) Create a program open text editor create a file in notepad called packet sniffing.py and code to capture and analyse the network packets
3) set up packet tracer by check of the packet has IP layer, identify the packet overload protocol such as TCP, UDP, ICMP.
4) capture network packets.
5) Run the packet sniffer by using command.
6) Generator 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[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst
```

```
    protocol_name = " "
    if protocol == 1:
        protocol_name = "ICMP"
    elif protocol == 6:
        protocol_name = "TCP"
    elif protocol == 17:
        protocol_name = "UDP"
    else:
        protocol_name = "Unknown protocol"
    print(f"Protocol : {protocol_name}")
    print(f"Source-IP : {src_ip}")
    print(f"Destination-IP : {dst_ip}")
    print("-"*50)
def main():
    sniff(prn=packet_callback, filter="ip",
store=0)
if __name__ == "__main__":
        main()
```

Output:-

```
> python packet_sniffer.py
protocol : TCP
Source IP : 23215.215.227
destination IP : 292.168.109.73
```

14/11

Result:-

The program to implement packet sniffing is done successfully

Practical -15

Aim: To analyse the different types of weblogs using webalizer tool

Algorithm:

Step: 1 Run webalizer windows version

Step: 2 Input web log file, download from web

Step: 3 Press run webalizer

10/11

Result:- Thus the different type of weblogs using webalizer tool is alone successfully

nes of

Version
nload.

completed

12/11