INDEX

NAME Arun Mohan STD CSE SEC A ROLL NO [illegible]

S.No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	16/7/24	Study of various Network		
2.	30/7/24	Study of Network cables		
3.	06/8/24	Experiments on CISCO Packet tracer		
4.	09/10/24	setup and configure a LAN using Ethernet		
5	16/08/24	Experiments on Product Capture using Wireshark		
6.	20/9/24	Hamming code		
7	23/9/24	Sliding window		
8	2/10/24	Virtual LAN config		
9	8/10/24	subnetting in CISCO		
10	15/10/24	Internetworking with routers in CISCO		
11.	12/10/24	Routing at network layer		
12	22/10/24	End-End Communication at Transport layer		
13	25/10/24	Implement your own Ping		
14	29/10/24	Write a code using RAW sockets to implement		
15	5/11/24	Analyze Various Types of [illegible]		

S.No	Date	Title	Sign
8a)	18/10/24	Virtual LAN configuration using cisco packet server	
b)	25/10/24	Configure of wireless LAN using cisco packet tracer	
9		Implementation of Subnetting in cisco tracer	
10a)	1/11/24	Internetwork using a Router	
b)	28/10/24	Internetwork using wirless Router DHCP server and Internet cloud	
11a)	4/11/24	Static Routing	
b)	4/11/24	RIP using cisco packet tracer	
12.a)	6/11/24	Echo client server using TCP/IP socket	
b)	6/11/24	chat program using socket programming	
13.	6/11/24	Ping Program	
14.	9/11/24	Packet Sniffing	
15.	11/11/24	Nebalizer	

completed

21/11

Date :- 12-07-2024

Ex No:- 01 Study of Various Network

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
172.16.42.1	7c-5a-1c-cf-bc-41	dynamic
172.16.72.133	4c-ac-a2-6f-9f-f3	dynamic

2) Hostname :-
DESTOP-- COIEH TO

3) nbtstat -a :
NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [Interval]
-a [adapter status] lists the network netbios name table given its name.

4) netstat :
Active connections
Proto LocalAddress Foreign address status
TCP 127.0.0.1:49674 Desktop Established
COIENTD:49674

5) nslookup :
Default server : unknown

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
6 Force using IPv6

Route :- Route [-f] [-p] [-4|6] command
[destination] [Mask network] [gateway]
[METRIC metric] [if Interface]

Some Important Linux Networking commands

1) ip :- ip <options> <object> <command>

2) [root@server ~] # ip address show

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

3) [root@server ~] # ip link set up
To alter the status of the Interface by bringing Interface enp2s0 on the

4) [root@server ~] # ip link set to down
To alter the status of the Interface by bringing the Interface enp2s0.

5) [root@server ~] # ip link set to promise on
To alter the status of the Interface by enabling promiscuous

6) [root@server ~] # ip route add default via 192.168.1.254 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.

7) [root@server ~] # ip route add 192.168.1.4 via 192.168.1.254
To add a route to 192.168.10.24 via gateway at 192.168.1.254.

8) [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

9) [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

10) Display -> [root@server ~] # ip route
get 10.10.1.4
uid 1000
cache

2. ifconfig:

~~lnp tbo flags . 19.~~

Output:

enp2s0 : flags = 4163 <UP, Broadcast, Running
> 1680 1500

inet 172.16.8.90 netmask 255.255.252.0

broadcast - 172.16.11.255

Rx packets 170123 bytes 17652931 (26.5 MB)

Tx packets 4339 bytes 1136044 (1.1 MB)

Rx errors 0 dropped 0 overruns 0

carrier collisions.

wlp3s0 : flags = 4099 <UP, broadcast,

multicast > 1500

ether ee:fb:34:8f:9d:84

Rx packets 0 bytes 0

Rx errors 0 dropped

TX packets 0 packets 0 bytes

Tx errors.

3) mtr

mtr google.com

(a)

Host	Loss%	Snt	Avg	Best	Worst
1. 172.16.8.1	0.0%	46	0.2	0.2	0.3
2. 142.250.171.362	0.0%	62	0.7	5.5	2.5
3. 142.251.229.217	0.0%	79	3.1	1.2	9.6

(b)

mtr google.com

Host	Loss%	Snt	Avg	Best	Worst
1. 172.16.8.4	0.0%	30	0.2	0.2	0.4
2. 142.250.171.162	0.0%	42	2.6	2.5	39

d) 3.142.251.52.207 0.0% 57 3.1 2.2

4) tcpdump:

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

to install - y tcpdump

[root@server ~] # tcpdump - D

1) lbo [up, Running, connected]

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

[root@server ~] # tcpdump -nn

dropped privs to tcpdump tcpdump

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 -c10 8.8.8.8

dropped privs to tcpdump

to full protocol decode listening on
lbt type EN10MB (ethernet) snapshot
length 262144 bytes.

[root@ server N] # tcpdump -i eth0 sec host
8888

dropped privs to tcpdump
tcpdump restore output

3. lo [up, running]
4. wlp3s0 [up
5. bluetooth [adapter number 0]
6. usbmon0 (All USB buses)

tcp dump -i Pato

(Q4) No such device exist

5) ping # ping google.com

(Q5)

PING google.com (142.250.77.142) 56(84)
64 bytes from maa05s16-in-f14.1e100.net
(142.250.77.14)

; icmp_seq=1 ttl=120 time=2.39 ms

64 bytes from maa05s16-in-f14.1e100.net (142
.250.77.14)

; icmp_seq=2 : ttl=120 time=2.67

64 bytes from maa05s16-in-f14.1e100.net (142.250
.77.142

: icmp_seq= 3 : ttl = 120 time = 2.64 m

< stopped ping: google.com

configuring an ethernet connection by
using nmcli

If you connect a host to the network
over ethernet, you can manage the
connections setting on the command
line by using nmcli

Procedure

1) list the network manager connection profiles
nmcli connection show

(Q6)

NAME	UUID	TYPE
wired connection	asa9uuo0-ebbe-3995-lock	ethernet
lo	exa-deb1-20ja-a7e1-99ee-	

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

3) optional Rename the connection profile
nmcli connection modify "wired connection"
"wired connection" is the name of connection

Display the current setting of connection profile # nmcli connection show

connection interface-name: Apso
connection autoconnect: yes
ipv4.method: auto
ipv6 method: auto

configure the ipv4 settings
To use DHCP, enter
nmcli connection modify "wired connection" method auto
to set a static ipv4 address, network mask default gateway, DNS servers & search
nmcli connection modify "wired connection ipv4.address 192.02.254 ipv4.dns 192.0.2.20 ipv4 example.

config the ipv6 settings
to use stateless address autoconfig (SLAAC) enter
nmcli connection modifies "wired connection method auto

Activate the profile
nmcli connection modifies "wired connection method auto.

Verification:

1) Display the IP setting of NIC's
ip address show enp5s20
enp5s0 2: <broadcast, multicast, up, lower up> mtu 1500 qdisc fq_codel state up group default qlen 1000

2) Display the ipv4 default gateway
ip route show default
default via 192.0.2.254 dev enp5s0 proto static metric 102.

3) Display the ipv6 default gateway
ip-6 route show default
default via 2001:db8:1:: dev enp5s0 proto static metric 102 pref

4) Display the DNS settings
cat /etc/resolv.conf
search example.com
name server 192.0.2.200
name server 2001:db8::1::ffff

5) Use the ping utility to verify that can send packets
ping <host_name> or <ip-addr>

4) Display the current setting of connection profile # nmcli connection show

connection interface-name; Apso
connection autoconnect yes
ipv4.method: auto
ipv6.method: auto

5) configure the ipv4 settings
To use DHCP, enter
nmcli connection modify "wired connection"
method auto
to set a static ipv4 address, network mask default gateway, DNS servers & search
nmcli connection modify "wired connection"
ipv4.address 192.02.254 ipv4.dns 192.0.2.20
ipv4 example.

6) config the ipv6 settings
to use stateless address autoconfig (SLAAC)
enter
nmcli connection modifies "wired connection"
method auto.

7) Activate the profile
nmcli connection modifies "wired connection"
method auto.

verification:

1) Display the IP setting of NLC's
ip address show enp1s0
enp1s0 2: broadcast, multicast, up, lowerup
mtu 1500 qdisc fq_code. state up group default
len 1000

2) Display the ipv4 default gateway
ip route show default
default via 192.0.2.254 dev. enp1s0 proto static
102.

3) Display the ipv6 default gateway
ip-6 route show default
default via 2001: db8:1: : fe dev enp1s0 proto
static metric 102 pref

4) Display the DNS settings
cat /etc (resolve_conf)
search example.com
name server 192.0.2.200
name server 2001: db8 : 1 : : ffff

5) Use the ping utility to verify that the host can send packets
ping <host_name : or_ ip_address>

Trouble shooting :

- Verify that the network cable is plugged into the host or sub.
- check whether the link failure exists ones on this host and host connect to sum
- Verify that network cable & network interface are working as experted perform hardware diagonstics steps and replace defeat cables and network interface and
- If configuration on disk doesn't match then config. on the device, start & restart manager creates on it

student observation :

1) What command is used to find the reachability of a host machine from your device
   (x) ping
2) Which command will give the details of taken by packet to reach its destination
   (x) trace route

3) Which commands displays the Tcp port status in your machine)
   (x) net state -tae
4) Which command displays & config your machine
   (x) ip config
5) write & modify the ip config in where machine give any answer it.
   (x) Edit the network config file & the use ifdown & 'ipv4' or 'ip' command to apply changes,

Result :

Thus study of various network command is executed & verified successfully.

Ex: No:2

Date: 20.7.24

Aim:-

Study of different types of Network Cables

1a) understand different type of network Cables

1. unshield 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	Application	Image
UTP	Category 3 and 5	10 Mbps upto 100 Mbps	Advantages * cheaper cost * Easy to install Disadvantages * More to electromagnetic interference and noise	i.e. Base-Ethernet Fast-Ethernet gigabit	
	Category 5e	1 Gbps			
STP	Category 6, 6a	10 Gbps	Advantages shielded for Than UTP less noise	Light Ethernet (10M) widely used in data center	
Coaxial Cable	RG-6 RG-59 RG-11	10-100 Mbps	Advantages 1. High bandwidth 2. Immune to interference Disadvantage * Limited distance 2. cost	speed of signal is 200 Mbps for network high speed internet	
Fibre optic cable	Single mode multi mode	100 Gbps	Advantages * High speed * High Bandwidth Disadvantages * Expensive * Require skilled	Maximum distance fibre optic cable is around 100 meters	

b) Make your own Ethernet cross-over (cable), straight cable

Tools and parts needed.

* Ethernet cabling CAT5e is certified for gigabit support, but CAT5 cabling works as well, just over shorter distances.
* A crimping tool. This is an all in one networking tool shaped to push down the pins in the plug and strip and cut the shielding off the cables.
* Two RJ45 plugs
* Optional two plug shields

straight thru cable X-over cable

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

step 1:- To start construction of the cables, begin by threading shields onto the cables

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

step 3:- After, you will need to untangle the wires; there should be four "twisted pairs". Referencing back to the sheet, arrange them from top to bottom. One end should be in arrangement A and 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, snip them back to create an even level. The difficult aspect is placing. The into the RJ45 plug without messing up the order. To do so, hold the plug and have the gold pins facing forward you, as shown.

steps:-

Next, push the cable right in. The notch at the end of the plug needs to be just over the cable sheathing, and if it isn't, that means that you stripped off too much sheathing. Simply strip the cables back a little more.

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

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

RESULT:-

Thus study of various type of cables is executed & verified successfully.

student observation.

1) What is the RJ45 cross cable & straight cable.

straight cable	cross cable
The wiring of both ends of cable is identical	The transmit & receive wires are reversed @ end of cable
It is used for connection types of devices	It is used for connection similar devices

2) Which type of cable used to connect two PC? (straight / cross cable)

(A) cross cable is used to connect two PC directly.

3) Which type of cable is used to connect a router or switch to PC

(A) A straight cable is used to connect a router or switch on PC

4) Find out the category of twisted pairs table in your lab to connect PC

(A) cat 5e, cat 6, cat 6a

5) Write down your understanding, challenges faced, or received while making a twisted pair.

(A) creating a twisted pair cable requires arranging wires according to specific sol. challenges to specifics

crimping:- output a functional cable that provides proper connectivity b/w devices

STUDY OF PACKET TRACKER TOOL

EXPT NO: 3

30-7-24

AIM:- To study the packet tracker tool installation & user Interface overview.

INTRODUCTION:-

1. It allows you to make complex system without for dedicated equipment.

2. It helps you practice your network configuration & trouble shooting skills via computer

3. It is available for windows desktop

4. Protocols in packets tracker coded to work & behave in real hardware.

INSTALLING PACKET TRACER:

WINDOWS:-

Installation in windows is pretty simple in straight the setup comes in single file name packettracer open setup file to begin the setup wizard choose location and start installation.

Linux:- It is ubuntu / kubuntu distribution should download the file for ubuntu, and those using fedora users must download the file for fedora

grant executable access channel

To packet tracker 601 - i386 - Installer - npn.bin

, packet tracker 601 - i386 - Installer - npn-bin

User Interface Overview:-

The layout of Packet Tracer is divided into several components. The components of the Packet Tracker Interface are follows

1. Menu Bar:- This is a common menu found in all software applications, it is used to open, save, print change preferences and soon.

2. Main Toolbar:- This bar provides shortcut icons to mean options to menu options that are commonly accessed such as open, save, zoom, undo.

3. Logical / Physical workspace bar:- This tab allows you to toggle between the logical and physical workspace

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

5. common tool Bar :- This toolbar provides controls for manipulating topologies, such as select, move layout, place note, delete, inspect, resize shape.

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.

7. Network component box :- This compound contains all of the network and each devices available. The area contains 7.1.1: Device specific sub

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

d) Analyse the behaviour of network devices using (Cisco Packet Tracer simulator)

1. From the network & compound box, click and drag and drop the components
   a) 4 Generic Pcs and one Hub
   b) 4 Generic Pcs and one Switch

2. click on connections
   a) click on copper straight - Through cable
   b) select one of the Pc and connect it to HUB using cable.
   c) similarly connect 4Pcs to the switch using copper straight through cable.

3. click on the Pcs connected to hub, go the Desktop tab, click on IP configuration, and enter an IP address and subnet mask.

• click on PDU (message Icon) from common tool Bar, a) Drag and drop it one of Pc (source machine) and then drop it on another Pc connected to HUB.

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

5. Repeat step # 3 to 5 for Pcs connec to switch.

6. Observe how HUB and switch are forwarding the PDU and write your observation and conclusion about swit

Ex:2 student observation

which command is used to find the reachability of a host machine from.

1) what is the difference between cross cable and straight cable?

Ans

Straight cable	Cross cable
* The wiring of the ends of the cable is identical * It is used for connecting different types of devices	* The Transmit and receive wires are crossed on one end of the cable * It is used for connecting similar devices

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

A cross cable is used to connect two PCs directly.

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

A cross cable is used to connect a router / switch a PC

4) Find out the category of twisted pair cable used in your lab to connect PC to router?

Cat 5e, Cat 6 or Cat 6a.

RESULT

The study of Packet tracer and user interface installation has been done successfully.

5) Write down your understanding, language used and output received while making a twisted pair cross/ straight cable.

Creating a twisted pair cable requires arranging wires according to specific standards (T568A / T568B).

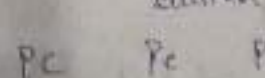
Q:2 Student observation

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

→ A Hub broadcast packets to all ports while a switch forwards packets only to the destination. Port based on the MAC address

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

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



EXNO:- 04 Setup and Configuration of a
DATE:- 9/8/24 LAN using Switch and Ethernet cable

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

What is a LAN?

A Local Area Network (LAN) refers to a network that connect devices within an limited area, such as office building, school or house. It enables users to share resources, including data, printers and Internet access.

How to set up a LAN?

1. Plan and design an appropriate network topology taking into account network requirements and equipment location.

2. You can take 4 computers, a switch with 8, 16 or 24 ports which is sufficient or for network of these sizes

3. Connect your computer to network switch ethernet cable. One to your computer other end to another.

4. Assign IP address to your PCs

* login to client comp as Administrator or as owner
* click network and connection.
* Right click local area connection ethernet. go to properties → select internet protocol (TCP/IPV4) → select the following ip address option and assign ip address.

Networking sharing

Internet Protocol version 4 (TCP/IPv4) Prop

General

You can get IP settings assigned automatically if your network supports this capability.

* Obtain an IP address automatically
* Use the fall IP address

IP address 10.1.1.1

Subnet mask 255.00.0

Default gateway

* Obtain DNS server address automatica
* Use the fall DNS server address

Preferred DNS server...

Alternate DNS server...

validate setting upon exit

Advanced

Similarly assign IP address to all the PCs connected to switch

PC_1 - IP address : 10.1.1.1 / subnet mask 255.0.0.0

PC_2 - IP address : 10.1.1.2, subnet mask 255.0.0.0

PC_3 - IP address : 10.1.1.3, subnet mask 255.0.0.0

PC_4 - IP address : 10.1.1.4, subnet mask 255.0.0.0

5. configure a network switch

6. Check the connectivity b/w switch and other machine using ping command in the command prompt of the device

7. Try to access the shared folder from other computers to network.

Result:

LAN was successfully setup using switch and Ethernet cable.

Exp No: 5 Packet Capturing tool : Wireshark

Date:

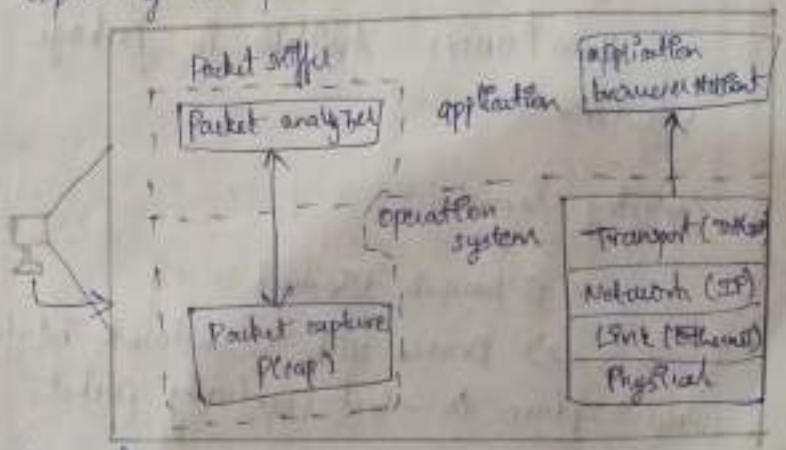
Aim:

Experiments on Packets capture tool : Wireshark

Packet sniffer captures messages in sent / received by your computer

Functions : stores and displays protocol fields in message.

* Passive : Does not send packets or receives packets addressed to it only makes copies of all packets



Wireshark:

Wireshark is a network analysis tool that captures and displays packets in real time. It features filter color coding and deep packet description capabilities.

Features:

* Capture network traffic
* Decode packet protocols with dissectors

* Analysis network problem
* Browse traffic interactively

Uses:

network administrators Troubleshoot network issues.

network security engineers. Examine security problems.

Getting Wireshark:

Windows / macOs : Download from the official website

Linux / UNIX: Available in package repositories

Capturing Recorder

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

WSJ:

VmWare Network Adapter VmNet

Ethernet - 2

Ethernet -

Sample:

Use sample files to practice in wireshark open file via -> open.

Altering Packets

Apply filter to focus on specific network traffic use other apps to isolate traffic for analysis Type a filter enter eg; dns

use analyze > display filter to pick or save filter see the docs for more

create a file to display only DNS packets + provide flows graph.

Go to capture -> option
select stop capture after 100 packets
click start capture
save packets

create a filter to display only http packets

Go to capture -> option
select stop after 100 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
search start capture
search ICMP/IP packets in search bar
Save the packets

capturing + analysing packets using wireshark tool To filter, view packets from the ethernet

select LAN, go to capture -> option
select stop capture automatically after
Then click start capture
Save the packets

capturing & analysing packets using wireshark tool to filter, view capture too packets from ethernet

Procedure:

Select LAN go to capture -> option
Select stop captures automatically after 100 packets then check start capture
save the packets

create a filter to display only TCP/UDP packets

Inspect the packets & provide flow graph select LAN, go to capture -> option
SAVE the packets

Ex 5 student observation

1) What is promiscous mode?

A network interface in promiscous mode captures all traffic on the network segment regardless of destination address.

2) Does ARP packets has transport layer header? Explain

Arp packets do not have transport layer header. They operate at datalink layer to map ip address to mac addr.

3) What transport layer protocal is used by DNS

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

4) What is the port number used by HTTP protocal

The HTTP_protocol uses port 80 by default communication.

RESULT

Thus the experiment on packet capture tool wireshark is executed successfully

Expt 6 Hamming Code

AIM:

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

Receiver:

Create sender programs 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.
4. Save this output in a file called channel
5. Apply hamming code on the binary data to check
6. Else remove the redundant & convert the binary data to ascii & display the 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)
def binary_to_text(binary)
chars=[binary[i:i+8] for i in range(0,len(
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
def pos_red_bits(data,r)
    j=0
    k=0
    m=len(data)
    res
for i in range(1, m+r+1)
    if i==2**j
    j+=1
else: res=res+data[k]
    k+=1 return res
```

```
def cal_parity_bits(arr,r):
    n = len(arr)
    arr = list(arr)
    for p in range(r):
        p = 0
        pos = 2**p;
    for j in range(p,n+1)
        if j+1 == pos:
parity = int(arr[j-1])
    arr[pos-1] = str(p)
    return join(arr)

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)
    parity += int(data[j-1])
if parity != 0
        res += position
if res != 0
print("error at pos:",res)
    data = list(data)
if res < n
data[res-1] = '0' if data[res-1] == '1' else '1'
print("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:
def remove bits(data, r)
    j=0
    original data =''
for i in range(i< len(data)+1):
    if i == 2**j-1:
    j += 1
else
    original data += data[i-1]
    return original data
if position < 1 or position > len(data)
print("Error pos is out of range")
return data.
```

```
data[position-1] = '0'
if data[position-1] == '1' else '1'
print("Introduced error at pos: {position}")
return pos, data

def sender(text):
    binary_data = text_binary(text)
    ph = len(binary_data)
    r = calc_red_bits(ph)
    arr = pos_redundant_bits(binary_data, r)
    arr = calc_parity_bits(arr, r)
print("{} 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("{} decoded text: {ascii_op}")

channel_data = sender(input_text)
receiver(corrupted_data)
```

Output:-

Enter the text to be encoded. of
sender message in binary : 01110000101001

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
: 100111010000101110101

Enter pos to change the bit : 4
Received msg in binary : 011000010111010
Decoded msg : ak

RESULT Thus the above code has been executed

Exp No: 7 Sliding Window
4/10/24

AIM:

Write a program to implement flow control at data link layer using sliding window protocol from one node to other.

Create a sender program will followings:

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

Create a receiver file:

1. Read a file named sender buffer
2. check the frame no
3. If the frame no are as excepted write the approx Ack. 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.acknowledge = False
    def send_frames(frames, window_size):
        print("sending frames")
        for i in range(window, size):
            if i < len(frames):
                print("sent frame", frames[i].frame_no,
                      frames[i].data)
    def receiver_frames(frames, window, size):
        for i in range(window_size):
            if random.random() > 0.1:
```

```
frames[i].acknowledge = True

def sliding_window_protocol():
    window_size = int(input("Enter window size"))
    message = input("Enter an message to send:")
    frames = [Frame(i, message[i])
    for i in range(len(message))]
    base = 0
while base < len(frames)
    send_frames(frames[base:], windowsize)
while base < len(frames) and frames[base]
ack base += 1
    if base < len(frames):
        print("Resending ack ack frames")
    time.sleep(2)
if __name__ == "__main__":
    sliding_window_protocol()
```

Output:-

```
Enter window size = 3
Enter a message to send: hello
--- sending Frames ---
Sent frame 0=h
Frames sent, waiting for acknowledgeme
sent frame 1=e
Frames sent, waiting for acknowledgem
Sent frame 2: l
Frames sent, waiting for acknowledgement
sent frame 3: l
Frames sent, waiting for acknowledge
sent frame 4: o
Frames sent, waiting for acknowledgem
--- Receiving Frames ---
Received frame 0: h [OK]
Received frame 1: e [OK]
Received frame 2: l [OK]
Received frame 3: l [OK]
Received frame 4: o [OK]
```

RESULT:

Thus the sliding window protocol has been executed successfully.

Exp: 8A

Virtual LAN

AIM - (a) Simulate - Virtual LAN configuration using Cisco packet tracer Simulation.

Device	Interface	IP address	subnetmask	Default
S1	VLAN-1	192.168.1	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

obj: connect the device as shown in the topology

Steps:

* Drag switch S1 & S2 to the rack
* Drag PC-A and PC-B to the table & power them on
* Connect the devices with copper

step-2: Assign VLANs to switch Interfaces

objective: Assign ports to VLANs

* Assign PC-A to VLAN 10
* Remove the management IP addr from VLAN 6 configuration.

steps:

* Assign PC-A to VLAN 10
* Remove the management ip addr
* Verify with show VLAN brief and show ip interface

Part-3 Maintain VLAN port assignment and the VLAN addr

step 1: Assign multiple interfaces to a VLAN

step 2: Remove VLAN assignment from an Interface

steps: Use the no switch port access VLAN command to remove VLAN assignment

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

step1 : Use DTP to initiate trunking.

steps : configure basic switch settings

obj : configure both switches

steps :

* use the terminal on each pc to console the switch & enter privileged EXEC mode.
* set the device name for each switch
* set the privileged exec password
* set the console password and enable
* Encrypt plaintext password.

steps : configure pc vests

obj : Assign ip addr to PCA and PC3

steps : In ip config input the ip address for PC3 for connectivity.

step4 :- Test connectivity

obj : Test pings between devices close configuration window.

step1 - Assign PCA to VLAN10 (operations) Remove the management ip address from VLAN1

ports : configure an 802.1q trunk between switches.

step1 : Use DTP to initiate trunking

obj : configure dynamic tracking.

verify using show interfaces trunk & ensuring is enabled between S1 & S2.

Questions :

1. can S1 ping S2 ?

Yes if trunking is successfully configure

S1 can ping S2

2. Can PC-A ping PC-B?

Yes, if VLANs are properly configured -trunking is enabled PC-A can ping PC-B

Reflection questions :

1. What is needed to allow host on VLAN to communicate to host on VLAN. on?

We need a 3 layer devices such as a router or a 2layer switch the VLAN routing configuration.

2. What are the primary benefits that an organization can receive through effective?

* Improved network segmentation

Student demonstration:

a) Draw and label the VLAN for 10 faculty in robotics department, staffing 5 in 3 different

b) Show the IP configuration for each device

faculty - 1 : 192.168.10.1/24 faculty - 3 : 192.168.10.3/24

faculty - 2 : 192.168.10.2/24 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

c) Write the commands

switch (config) # vlan 10

switch (config-vlan) # name Robotics-Vlan

switch (config-vlan) # exit

switch (config-if-range) # switch port access

switch (config-if) # exit

Result:

Thus the above circuit is executed successfully.

EXNO: 8b

AIM:

To design a topology with three PCs connected from Packet tracer wireless routers

Procedure:

Configure static IP on PC and wireless router set SSID to router network set IP address of router to all PC's connect your PC by using Key

Step 1: click on wireless Router

Step 2: wireless security Admin Setup

Management Router pass: admin

Router Access Router pass: admin

Setup wireless security Access restriction

Security mode Disabled

Enabled

WEP

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

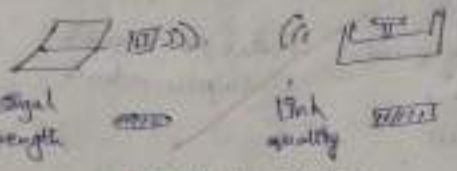
Pc	IP	Subnet mask	Default gateway
PC0	192.168.0.2	255.255.255.0	192.168.0.1
PC1	192.168.0.3	255.255.255.0	192.168.0.1
PC2	192.168.0.4	255.255.255.0	192.168.0.1

Click PC wireless

click on connect tab and click on refresh button.

WEP Key
Security WEP
WEP: 0x1234567890
KEY1: 0123456789

Cancel connect



Repeat for all PCs

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

Student observation:

1) What is SSID a wireless router

The SSID can be set identified of wireless router. It allows devices to connect to it

2) What is a security key in wireless router

A security key in a wireless router is a password used to protect a wifi network. WPA2 keys

3) Configure a simple wireless LAN in Java CAS using a real point & write down the configuration in your network.

Access point setup, connect the access point to power and network the access

eg: 192.168.1.1

set SSID name and configure

Set security mode WPA2-PSK

Password: set a key.

Ex:14
11/07/24

Subnetting

AIM:

Implementation of subnetting in cisco packet tracer simulator

Classless ip subnetting is a technique that allows for more efficient use of ip address by allowing for subnet masks that are just the default masks. when we have limited no of ip addr but need to create multiple network

Creating a network topology:

Once we have created our network topology we can add devices to it. Here we will be adding routers, switches, PCs to add a devices select the devices topology:

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
- PC5: 192.168.1.15

Fast ethernet 0/2
192.68.2.0/24
- 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
192.68.20/27
- 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

Configure the devices:-

Router configuration

configuration interface

Fast ethernet 0/0:

Enter: Interface ethernet 0/0

Active: no shutdown

Exit: Interface configuration: exit

Fast ethernet 0/1:

* Repeat the above steps.

Configure gigabit ethernet:-

Switch configuration:-

Enter vlan1 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 fa fastethernet 0/1 interface

* Set to the fastethernet 0/1 interface

* Enter once default or reached.

Testing the network:-

A successful ping indicates proper pc-to-pc communication

RESULT:-

Thus the above connection and packet trace was executed succes

Student declaration

a) Write down your understanding of subnetting.

Subnetting is the process of dividing a larger network into smaller manageable sub networks. Each subnet operation.

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

Ex:10 Internetwork using a router

Aim:

Internetworking with routers in cisco

Packet Tracer Simulator

Design and configure a simple internetwork using a router

In this network, a router and 2 pcs are used. Computers are connected with routers using a copper straight-through cable. After forming the network, to check network connectivity a simple ping transfer from PC0 to PC1

Procedure:

step1 (configuring router 0)

1) Select the router and open CLI

2) Press Enter to start configuring router

3) Type enable to activate the privileged mode

step2 (configuring PCs)

1) Assign IP address to every PC in the network

2) Select the Pc, to the desktop and Select Ip configuration and assign an Ip address, Default gateway, subnet Mask

steps Connecting Pc with Router

1) Connect Fast Ethernet 0 port of Pc0 with Fast Ethernet 0/0 port of Router using a copper straight-through cable

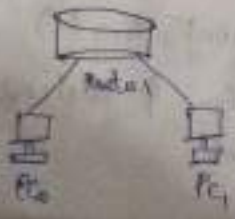
2) Connect Fast-Ethernet0 port of Pc1 with Fast Ethernet 0/0 port of Router

Router configuration Table

Device name	Ipaddr Fast-Ethernet	Subnet mask	Ipaddr Fast Ethernet	Subnet mask
Router	192.168.10.1	255.255.255.0	192.168.20.1	255.255.255.0

Pc configuration Table

Device name	Ipaddr	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



RESULT

Thus the Internetworking with router in cisco has be Implemented successfully.

Ex:10.b Interntwork using wireless routers
2/10/24 DHCP server and Internet cloud

Aim:-
Design and configure an Internetwork using wireless router, DHCP, server and Internet cloud

Address table:-

Device	Port/Interface	IP address	Subnet mask	Default gateway
Pc	Ethernet0	DHCP		192.168.0.1
wireless router	LAN	192.168.0.1	255.255.255.0	
wireless Router	Internet	DHCP		
Classroom server	Ethernet0	200.100.50.130	255.255.255.0	
laptop	wireless0	DHCP		

Algorithm:-
Part 1:- Build a simple network in logical Topology workspace
Part 2:- configure the network devices
Part 3:- Test connectivity between different devices

Part-4 : Save the file and close packet tracker

Part 2:- configure the network devices

Step 1:-
configure the wireless router
a) Create the wireless network on the wireless router. change all the nessary
b) Save the settings

Step 2:-
configure the laptop
a) configure the laptop to access the wireless network

Step 3
a) configure the Pc for the wired network

Step 4:- configure the Internet cloud
a) Install network modules if nessary
b) Identify the from and to ports
c) Identify the type of provides

step 5:-

configure the Cisco.com server.

a) configure the Cisco.com server.

b) configure the Cisco.com server global settings

Part-3 verify connectivity

step 1:- Refresh the IPv4 settings on the PC

Test connectivity to the Cisco.com server from the PC

student observation:-

1) Key features of configuration wireless router and DHCP server?

Ans:- *Wireless router configuration set the SSID (network name), security type (WPA2 or WPA3) and method to secure network access.

* DHCP server configuration:- Enable the DHCP server to automatically assign IP address to devices on network.

*

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

* Automated IP assignment. DHCP simplifies IP address management by dynamically assign IP addr to devices reducing the risk of IP conflicts and manage errors

Result:- Thus the internetworking using wireless router, DHCP server and Internet cloud is implemented successfully.

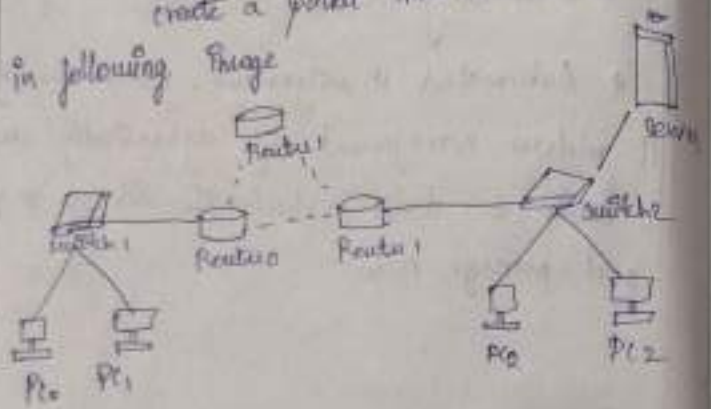
Ex No 11a
Routing

Static Routing

Aim: Simulate static routing configuration using cisco packet tracer

setting up a practice lab

create a packet tracer lab as shown in following image



The following table lists the connected networks

Router	Available network on local Interfaces	network available on other router's
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, 50.0.0.0/8, 60.0.0.0/8	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 configuration

enable configure terminal

ip route 30.0.0.0 255.0.0.0 20.0.0.2 10
ip route 30.0.0.0 255.0.0.0 40.0.0.2 20
ip route 30.0.0.100 255.255.255.255 40.0.0.2 10
ip route 30.0.0.100 255.255.255.255 20.0.0.2 20
ip route 50.0.0.0 255.0.0.0 20.0.0.2 10

exit

Router 1 configuration:

enable configure terminal

ip route 10.0.0.0 255.0.0.0 20.0.0.1 10
ip route 10.0.0.0 255.0.0.0 50.0.0.1 20
ip route 40.0.0.0 255.0.0.0 20.0.0.1 10
ip route 40.0.0.0 255.0.0.0 50.0.0.1 20

exit

Router 2 configuration

enable
configure terminal

ip route 10.0.0.0 255.0.0.0 40.0.0.1
ip route 30.0.0.0 255.0.0.0 50.0.0.2

exit

Verifying static routing

tracert command sends ping requests to destination host and tracks the path they take to reach.

RESULT:

Thus the static routing configuration has been implemented successfully.

Ex: 11b
14/11/24

RIP using cisco packet tracer

AIM:

Simulate RIP using cisco packet tracer

Partial IP configuration

Devices	Interface	IP configuration	connected with
Pc0	Fast Ethernet	100.0.2/8	Router0's Fa0/1
Router0	Fa0/1	100.0.1/8	Pc0's fast Ethernet
Router0	Se0/0/1	192.168.1.254/30	Router2's se0/0/1
Router0	Se0/0/0	192.168.1.249/30	Router1's se0/0/0
Router1	Se0/0/0	192.168.1.250/30	Router0's se0/0/0
Router1	Se0/0/1	192.168.1.246/30	Router2's se0/0/0
Router2	Se0/0/0	192.168.1.245/30	Router1's se0/0/1
Router2	Fa0/1	20.0.0.1/30	Pc1's Fast Ethernet
Pc1	Fast Ethernet	20.0.0.2/30	Router2's fa0/1

Assign IP address to interfaces of routers

* set the clock rate for DCE and not for the DTE end

* show controllers interface gives whether the interface is DCE or DTE

Configure RIP routing protocol

Router
router rip
network rip
Network 10.0.0.0
network 192.168.1.252
network 192.168.1.248

Router1
router rip
Network 192.168.1.244
Network 192.168.1.248

Router2
router rip
network 10.0.0.0
network 192.168.1.252
Network 192.168.1.144

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

RESULT

Thus the simulation of RIP using (Two

Ex no: 6/11/24 Echo client server using TCP/UDP sockets

AIM: Implement echo client server using TCP/UDP sockets

Algorithm (server):

1) Create a TCP socket
2) Bind socket to local address port
3) Listen for incoming client connections
4) Accept a client connect
5) Loop
   * receive data from client
   * else break loop
6) Close the connection.

Algorithm (client)

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

Program:

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("The server is waiting for a connection")
    connection, client_address = server_socket.accept()
    print(f"Connected to {client_address}")
    try:
        while True:
            data = connection.recv(1024)
            if data:
                print(f"Received: {data.decode()}")
            else:
                break
    finally:
        connection.close()
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.recv(1024)
        print(f"Received from server: {data.decode()}")
    finally:
        client_socket.close()
if __name__ == "__main__":
    tcp_client()
```

Output:

Terminal 1:

Tcp_server.py

```
PS C:\Education\semester\cn lab\python -u "C:\Education\semester\cn lab\tcp_server.py"
Tcp server is waiting for a connection...
connected to ('127.0.0.1', 53356)
Received: hello this is tcp client
```

Terminal 2

TCP client ??

Enter message to send hello this is tcp client received from server hello this is tcp client

Result:

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

Aim: Chat program using socket programming

To implement the chat client server using TCP/UDP server

Algorithm

chat server

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

2) When a new client connects and client a list of clients start a new process to talk too the clients

3) For each connected client keep checking new messages.

4) keep running the process till the server stop

chat client

1) connect to server by creating a socket

2) start a process to listen to message from the server

3) Keep running till the user decides to quit

chat_server.py

```
import socket
import threading
def handle_client(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode
            if not message:
                break
            print(f"Received message from client: {message}")
            client_socket.send(response.encode('utf-8'))
        client_socket.close()
def start_server():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(('127.0.0.1', 12345))
    server.listen(5)
    print("Chat server started on 127.0.0.1:12345")
    while True:
        client_socket, addr = server.accept()
        print(f"New connection from {addr}")
        client_handler.start()
```

chat_client.py

```
import socket
import threading
def receive_message(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).de
            if message:
                print(f"server: {message}")
        except Exception as e:
            print(f"An error occured: {e}")
            break
def start_client():
    client_socket = socket.socket(socket.AF_
INET, socket.SOCK_STREAM)
    host = '127.0.0.1'
    port = 12345
    client_socket.connect((host, port))
    print("Connected to the chat server!")
if __name__ == "__main__":
    start_client()
```

Output:

Terminal-1

chat_server.py

chat server started on 127.0.0.1:12345

New connection from ('127.0.0.1', 53686)

Received message from client: hello this is chat client

Type your msmge to client - hello this is chat server

Terminal:2

chat_client.py

connected to the chat server

You: hello this is chat client

You: server: hello this is chat server.

Result

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

Ex No: 13 Ping Program

Aim:

Implement your own ping program

Algorithm:

ping-client.py

1) socket creation

2) Then set a timeout of 1 second to ensure if no response is received it will stop waiting and print "Request point"

3) send a 'ping' msg to specified host post

Ping_server.py

1) Initialize UDP socket

2) Bind to IP address & port

4) Reverse data

5) send response

Program:

ping_server.py

import socket

def start_server(host='127.0.0.1', port=12345)

with socket.socket(AF_INET, SOCK_DGRAM)

s.bind((host, port))

print(f"UDP server running on {host}:{port}")

```
while True:
    data, addr = s.recvfrom(1024)
    print(f"Received message from {addr}: {data.decode()}")
    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.AF_INET, socket.SOCK_DGRAM) as s:
    try:
        s.settimeout(5)
        start = time.time()
        s.sendto(b'ping', (host, port))
        data, addr = s.recvfrom(1024)
        end = time.time()
        print(f"Received {data.decode()} from {addr} in {end-start:.2f} seconds")
    except socket.timeout:
        print("Request timed out")

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

Output:

Terminal:

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

Terminal:

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

RESULT:

Thus the ping programs has been executed successfully.

Ex: No: 7th
Date:

Packet Sniffing

AIM:

write a code using RAW sockets to implement packet sniffing.

Algorithm

1) Install python and scapy
2) Create a program open text editor and create a file in notepad called.
3) setup packet trace by check if packet has IP layer, identify the packets
4) Run the packet sniffer by using command
5) Run the packet sniffer by using command

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

output:

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

RESULT:

Thus the packet sniffing program has been executed successfully

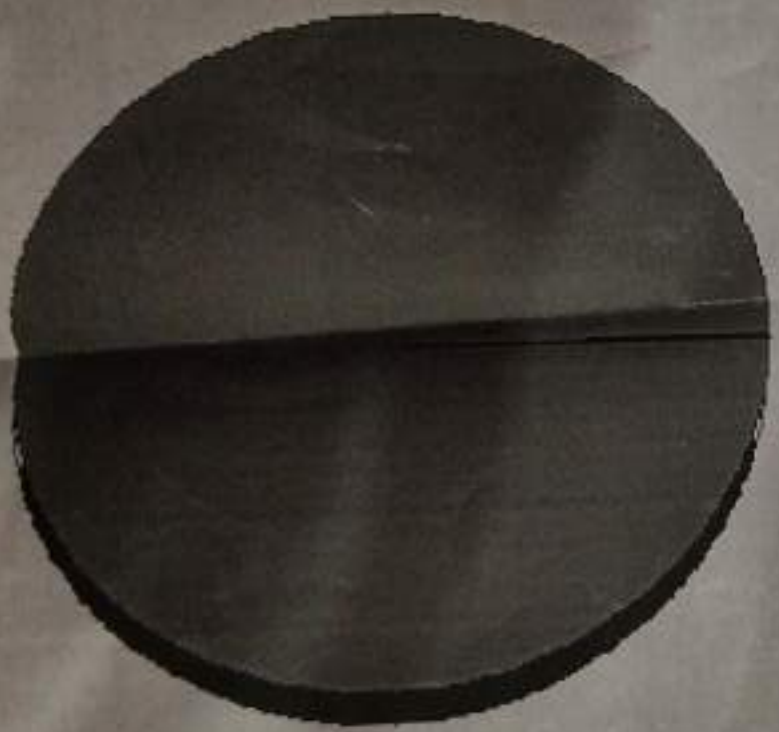
Ex: 15 Weblalyzer
7/11/24

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

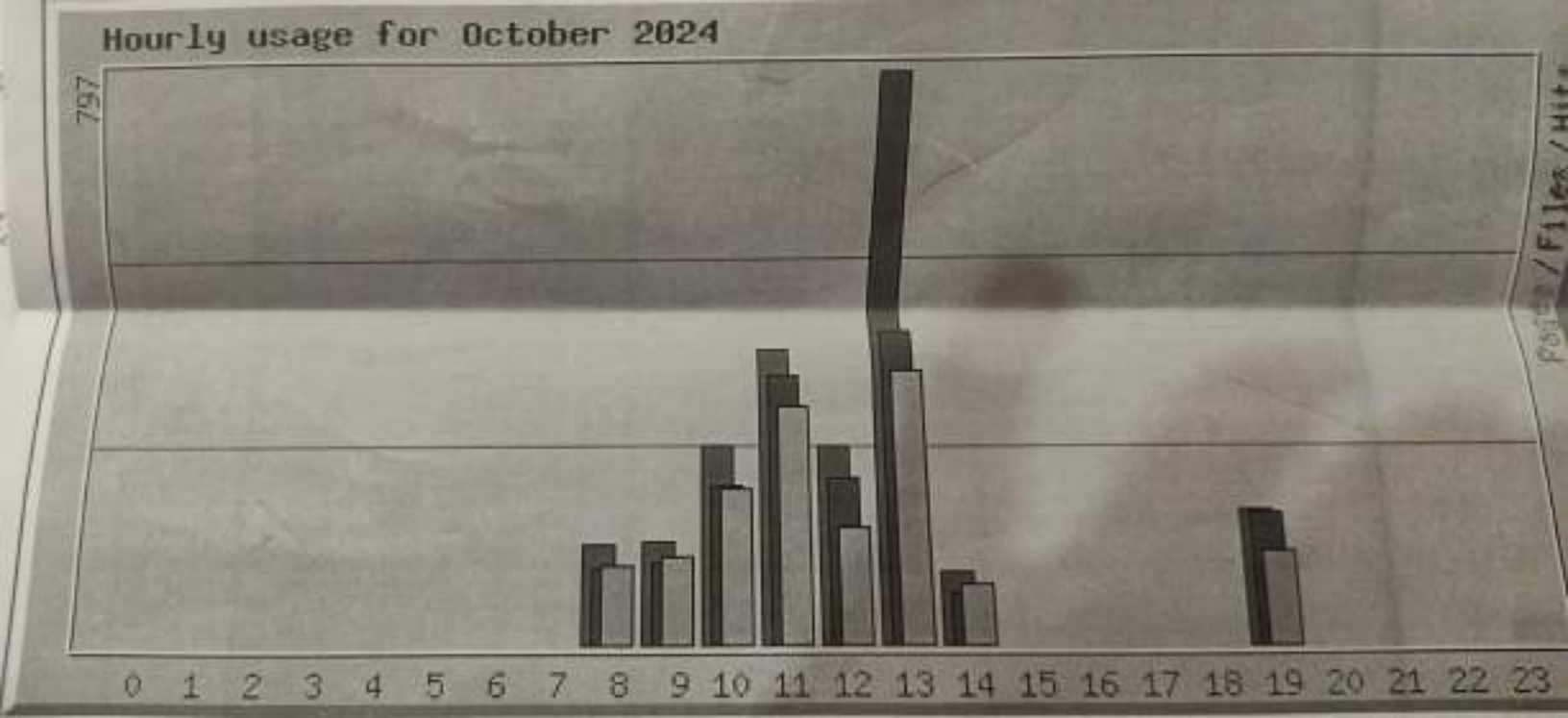
Algorithm:
* At first you must have installed Xampp and hosted
* In the web alizer. config update the distination path to store the weblogs
* After updating it open command prompt
* The result is stored in the updated destination path
* check the various logs in the index.html file.

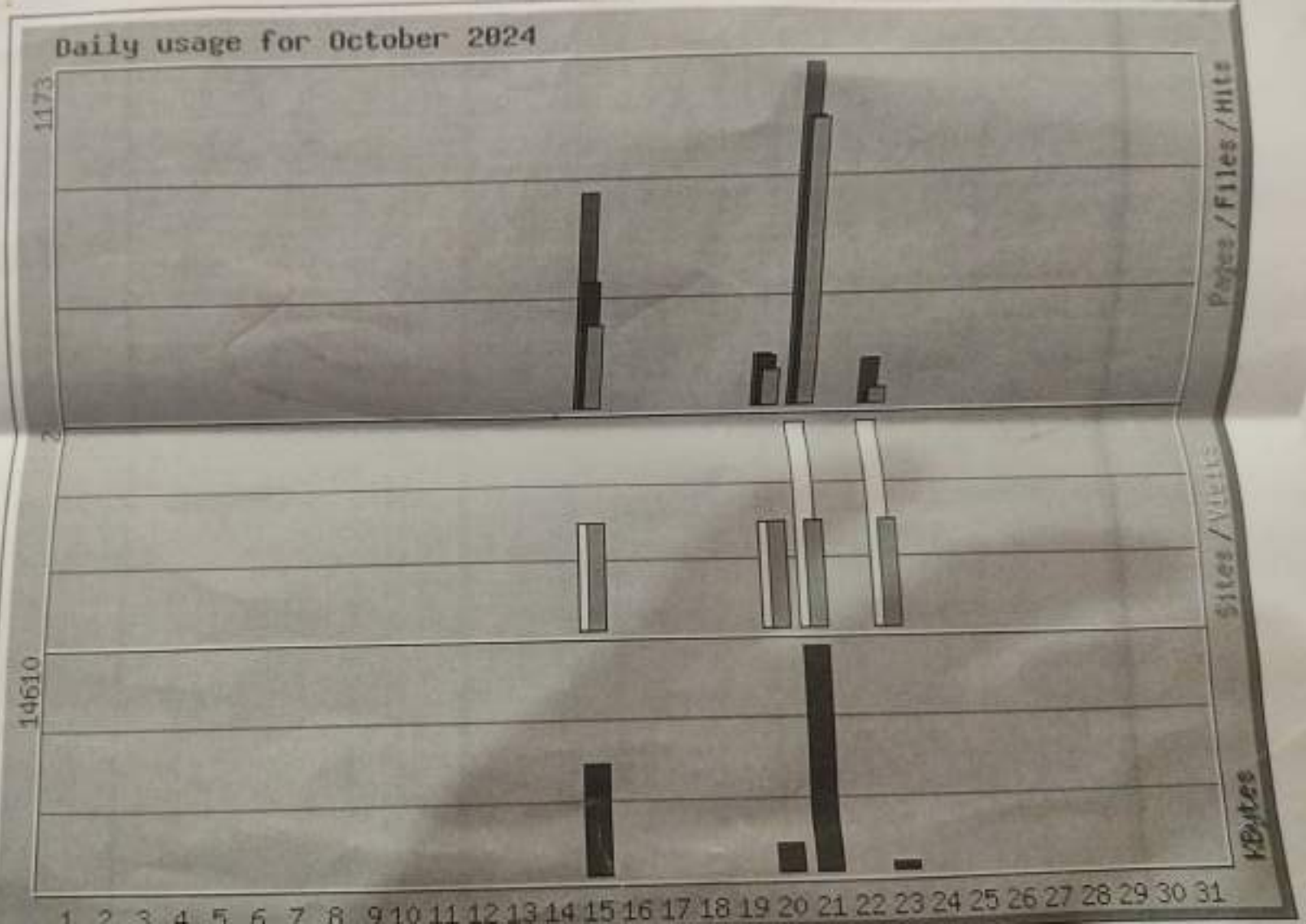
RESULT
Thus the different types of weblogs has been analysis using webalizer tool.

Usage by Country for October 2024



Unresolved/Unknown (100%)


Hourly usage for October 2024
797
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23


Daily usage for October 2024
1173
2
14610
Pages / Files / Hits
Sites / Visits
KBytes
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31