Computer Networks - LAB MANUAL

Aim:- Study of various network commands used in Linux and window

Basic network commands:

arp -a : Interface : 172.16.75.54 --- 0x12

Internet Address	Physical Address	Type
172.16.72.1	7c-50-1c-cf-be-41	dynamic
172.16.72.133	4c-0e-03-65-97-f3	dynamic

hostname:

DESKTOP - C01BH7D

nbtstat -a:-

NBTSTAT [[-a Remote name] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [interval] -a (adapter status) Lists the remote machine name table given ip name

Net Stat:-

Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49673	Desktop-C01BH70:49674	

Nslookup:

Default Server: 172.16.8.1

Address: 172.16.8.1 #53

Pinging usage: Ping [-g host-list] [-n maximum hops] [-r address] [-h] [-p Period] [-q num-queries] [-w timeout] [-4] [-6] target-name

Ans:-

Kernel IP routing table
Destination gateway genmask flag metric Ref
Default gateway 0.0.0.0 UG 100 0

9. To show the IP address assigned to an interface on your server

a. [root@server ~]# IP address show

1: lo: <Loop Back, UP, lower-UP> mtu 65536 qdisc noqueue state UNknown group default qlen 1000
link/loop back 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever

a [root@server ~]# ip address and 192.168.1.234/24

To assign an IP to an interface for example,

a. [root@server ~]# ip address del 192.168.1.234,21 dev enp0s3

To delete an IP on an interface.

d. [root@server ~]# ip link set enp0s3 up

After the status of the interface is, bringing the interface enp250 online

e [root@server ~]# ip link set enp0s3 down After the status of the interface by bringing the interface enp250

f. [root@server ~]# ip link set echo promisc

. After the status of the interface by enabling promisc mode for enp2s0

g. [root@server ~]# ip route add default via 192.168.1.254 dev enp2s0

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 192.168.1.0/24 via 192.168.1.254

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

[root@server ~]# ip route 192.168.1.0/24 dev echo

To add a route to 192.168.1.0/24 that can be reached on device echo.

j. [root@server ~]# ip route . delete 192.168.1.0/24 via 192.168.1.256

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

dev ens160 via 192.165.1.254

Ping 192.168.22.128 netmask 255.255.255.0
broadcast 192.168.22.255

---

3. mtr

mtr <option> <hostname ip/fq>

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

Key: Help Display mode restart Statistics order of fields

	Packets		Pings		
Host	Loss% Snt Last		Avg	Best	Wrst StDev
1.-gateway	0.0% 163 1.2		1.0	0.3	6.1 0.6

b. [root@server ~]# mtr -g google.com

-f, --filename file — read hostname(s) from file
-4 — use IPv4 only
-6 — use IPv6 only
-u, --udp — use UDP instead of ICMP echo

c. [root@server ~]# mtr -b google.com

Keys: Help Display mode Restart Statistics Order of fields quit

	Packets		Ping		
Host	Loss% Snt Last		Avg	Best	Wrst StDev
1.-gateway	0.0% 391 1.1		1.1	02	102 0.8

(d) [root@server ~]# mtr -c10 google.com

Key: Help- Display mode Restart Statistics order of fields quit

	Packets			Pings		
Host	Loss%	Snt	Last	Avg	Best	Wrst StDev
gateway	0.0%	1	0.7	7	0	0.4 1.3 0.3

---

4. Tcpdump

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

To install - y tcpdump

[root@server ~]# tcpdump -D

1. ens160 [Up, Running, Connected]

2. any (l) or do - Drie that captures on all interfaces) [up, Running]

[root @server ~]# tcpdump -i eth0

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

[root @server ~]# tcpdump -i eth0 -c 10

10 packets captured
20 packets received by filter
0 packets dropped by kernel

[root @server ~]# tcpdump -i eth0 -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, linux
bytes (ethernet) snapshot length 262144 bytes

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

dropped privs to tcpdump
tcpdump: verbose output suppressed, use -v[v].
for full protocol decode listening on lo,

[root @server ~]# tcpdump -i eth0 net 10.1.0.0 mask 255.255.255.0

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

[root@server ~]# tcpdump -i eth0 net 10.1.0.0/24

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

[root @server ~]# tcpdump -i eth0 net 10.1.0.0/24

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

[root@server ~]# tcpdump -i ethO port 53
to capture only DNS port 53 traffic

[root@server ~]# tcpdump -i ethO host 8.8.8.8
and port 53
this is for a specific host

[root@server ~]# tcpdump -i ethO -c 10 host
www.google.com and port 443
To capture only HTTPS traffic

[root@server ~]# tcpdump -i ethO port not 53 and
not 25
To capture all port except port 80 and 25

Ping
-Usage: ping [-D] [-a] [-T count] [-i interval] [-b]
[-I intf] [-v TOS] [-r count] [-s count] [-T timestamp_opt]
[-k shost- list] [-w deadline] [-e] [-S sndbuf]
[-l (preload)] [-p] [-4] [-6] destination

[root@server ~]# ping google.com
PING google.com (216.58.200.142) 56(84) bytes
of Data 64 bytes from maa03s... 10... 1e100.net
(216.58.200.45): icmp_seq=1 ...

[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, 10 received, 0% errors, 100%
packet loss, ... time 9197ms pipe 3

nmcli:

"Vmware VMXNET3"
ethernet (vmxnet3), 00:0C:29:69:E0:FF, hw
ip4 default
inet4 192.168.22.128/24
inet4 192.168.22.0/24 metric 100

nmcli connection show

NAME	UUID	TYPE	DEVICE
wired connection 1	05e6b490-1c20-3668-8f8b-63140 27f3f75	ethernet	ens150

nmcli connection modify "Wired connection 1"
Here, "wired connection 1" is the name of the connection

Display the current settings of the connection profile
nmcli connection show
connection.interface-name: ens150
connection.autoconnect: yes
ipv4.method: auto
ipv6.method: auto

Verification
ip address show IP setting of the NIC:
ens150: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc fq_codel
state UP group default qlen 1000
link/ether 52:54:00:17:b8:b6 brd ff:ff:ff:ff:ff:ff
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute ens150
valid_lft forever preferred_lft forever
inet6 2000:1:db8:1ddc:160 scope global noprefixroute
valid_lft forever preferred_lft forever

Ftcat /etc/resolv.conf

Search poxpiglo.com

nameserver 192.0.2.200

If multiple connection profiles are activated, the order of nameserver entries depend on the profile priority values in those profiles and the connection type.

nmcli

"vmxnet3 VMXNET3"
ethernet (vmxnet3), 00:0c:29:61:f0:ff, hw
ip4 default
inet4 192.168.22.128/24
inet6 192.168.22.0/24 noprefixroute

nmcli connection show

NAME	UUID	Type	Device
Wired Connection	2a0b0c8-7276-3116 -b790-0b29021b0c9	ethernet	ens160

nmcli connection show

connection interface-name: ens160
connection autoconnect: yes
ipv4 method: auto
ipv6 method: auto

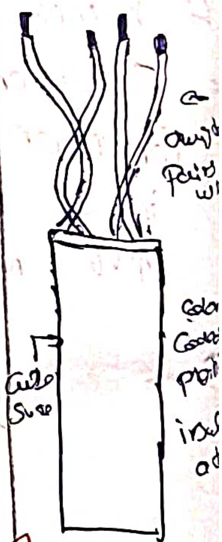
29/8

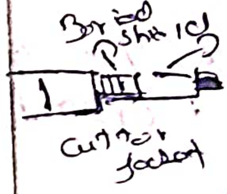
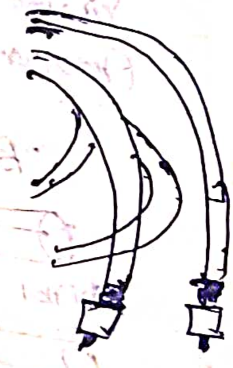
Result:
Thus the study of various Network commands used in Linux and windows is done and executed successfully

Exp no :- 2
Date :-

Aim :-
Study of Different types of Network Cable

(a) Understand different types of network cable
Different types 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	Maximum Data transmission	Adv/ Disadv	Appli use	image
UTP	Category3 Category5	10bps up to mbps	Adv:- • Cheaper cost • Easy to install • having a smaller overall diameter Disadv:-	10 base Ether fast Ethernet gigabit Ethernet	
	Category 5e	1Gps	• More prone to (EMI) Electro magnetic interference and noise	for Ethernet gigabit Ethernet	
STP	Category 6,6a	10Gbps	Adv:- • Shielded • faster than UTP • Less susceptible to noise and interface	gigabit Ethernet 10 G Ethernet (55 m) Ruggedized in data centers	

SSTP	Category	1 Gbps	Disadv:- • Expensive • greater • installation effort Adv:- • high bandwidth • immune to interference low loss bandwidth • versatile	gigabit Ethernet 10 Ethernet (100m) Speed of signal is 500m to television network & high speed internet connection	
Coaxial cable	RG-6 RG-59 RG-11...	10-100 Mbps	Adv:- • high bandwidth immune to interference low loss bandwidth • versatile Disadv:- • Limited distance • cost • size is bulky	Speed of 500m ~~television~~ television network high speed internet connection	
fibre optics cable	Single mode Multi mode	100 Gbps	Adv:- • high speed • high band • high security • long distance Disadv • Expensive • requires skilled installer	Maximum distance of fibre optics cable is around 100km	

(b) Make your own Ethernet (RJ45) - over cable Straight-through ~~cable~~ cable

Tools and parts needed

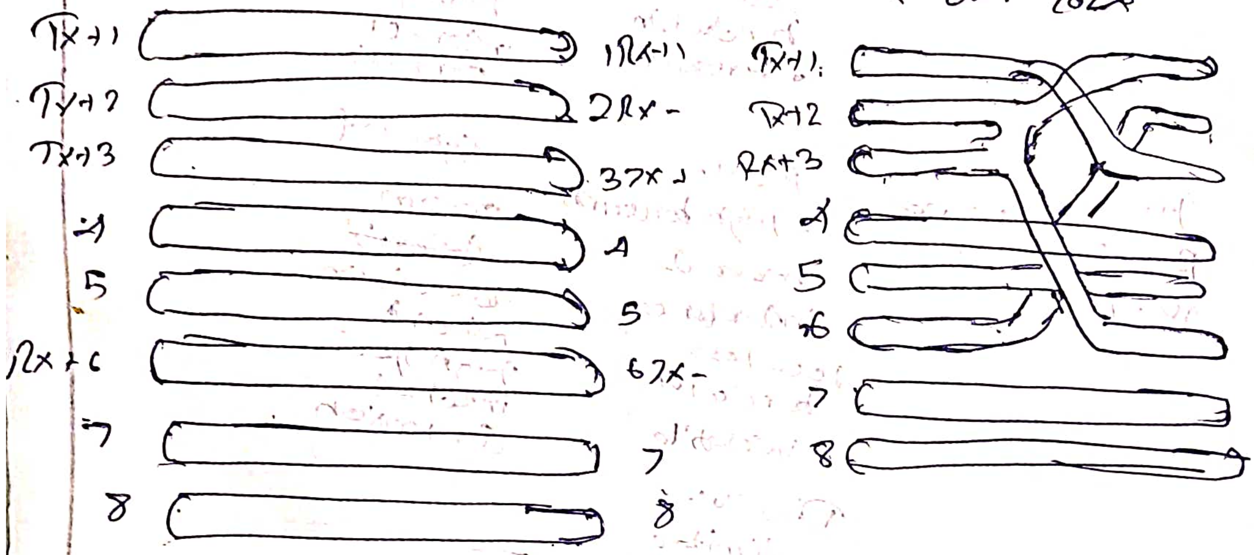
• Ethernet cabling CAT 5E is certified for gigabit support but CAT 5 cabling works as well, just over shorter Distance

• A crimping tool this is an install pin on removing dead shaped to push down the pins in the plug and strip and cut the shielding off the cable

• Two RJ45 plugs
• optional two plug ~~shields~~ shields

Straight thru Cable

X-over Cable



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

A

- White orange stripe
- Orange strong
- White orange stripe
- blue solid
- White blue stripe
- Green solid
- White brown stripe

B

- stripe
- Green solid
- Orange stripe
- White brown stripe
- brown solid
- orange solid
- blue solid
- white/blue stripe

Step 1:- To start construction of the device, beginning thread the shields onto the cable

Step 2:- Next, strip approximately 1.5cm of cable sheilding 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" Reforming back to the sleeve, array them from top to bottom. One end should be in arrangement of and order in B

Step 4:- once the order is locked, bunch them together in a line and use the one that sticks out further than others, snip them back to create an even level. difficult aspect is placing these into the RJ45 plug without messing up the order. To do so, hold the plug with the clip side facing towards you, and those the gold pins facing toward you, as shown.

Step 5:- Next, push the cable right into the reach at the end of the plug. Needs to be visible, and now, and you stripped off too much shielding - simply strip the cable 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 the study of various network devices and process successfully.

Ex NO 3

Date

Aim :-

To study the packet tracer installation and user interface overview

1) To understand environment of cisco packet tracer to Design Simple network

introduction

A Simulator or more specifically Simulated networks device and its environment Packet tracer is a operating network design Simulation & modelling tool

(1) It allows you to model complex systems without the need for dedicated equipment

(2) It helps you to practice your network configuration and trouble shooting skills via computer or an android or ios based mobile device

3) It is available for both the linux and windows Desktop environment

4) Protocol 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.netacad.com and log in with your cisco networking academy credentials; then click on the Packet tracer graphic and download the package appropriate for your operating system (can be used to download in your laptop)

Windows:

installation in windows is pretty simple and straightforward; the setup comes in a single file named PacketTracer-Setup 6.0.1.exe. open this file to begin the setup wizard, accept the license agreement, choose a location, and start the installation

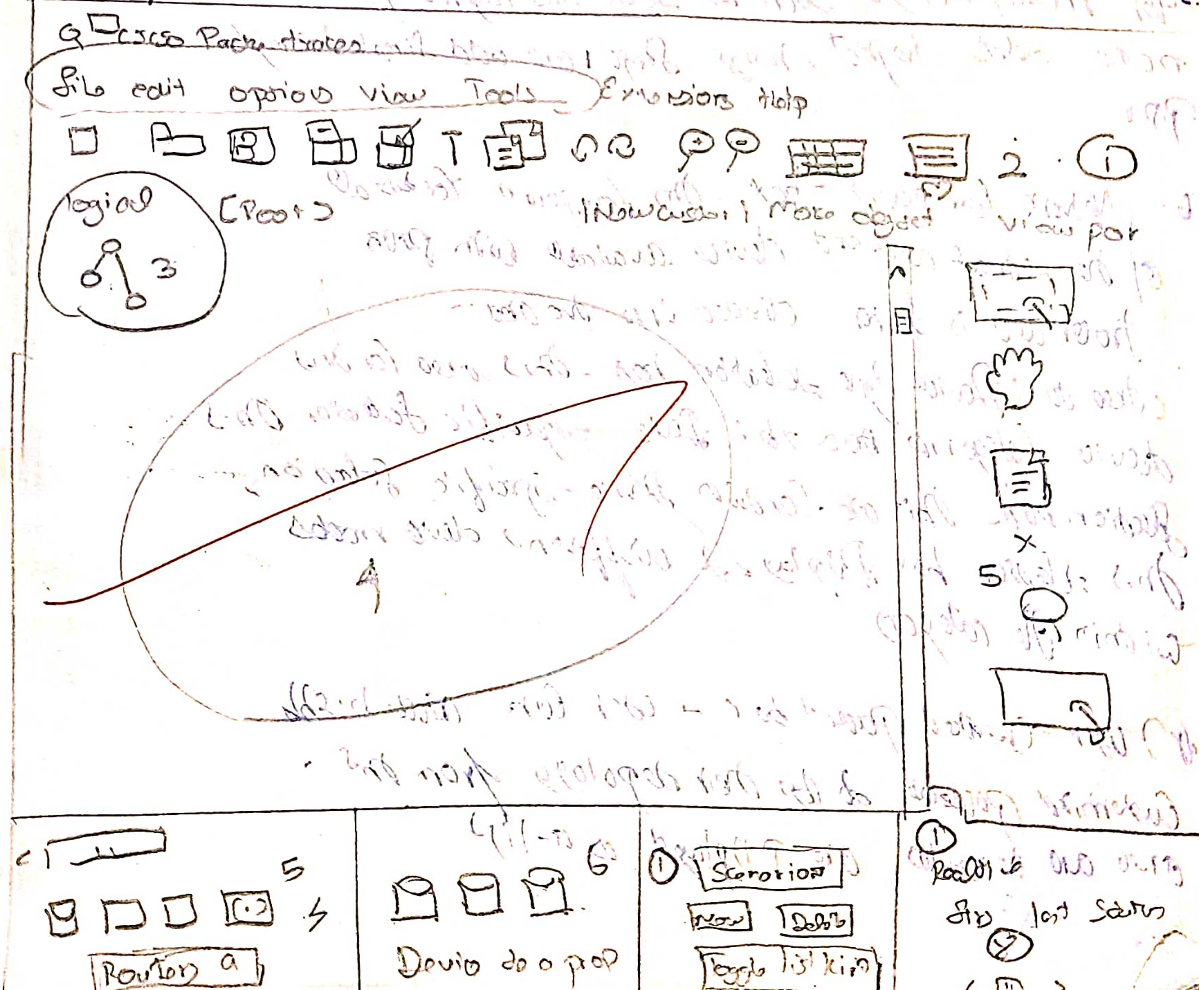
Linux:

Linux users with an ubuntu/debian distribution should download the file for ubuntu and those using fedora/Redhat/Centos must download the file for fedora grant executable permission to this file by using chmod, and execute it to begin the installation

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

./PacketTracer601-i386-installer-rpm.bin

User interface overview



1. menu bar - This is a common menu found in all software application it is used to open, save, print, change preferences and so on

2. main toolbar - This bar provides shortcut - icons to common options that are commonly accessed such as open, save, zoom, undo and redo, and on the right - hand side is an icon for entering radar information for the current network

3. Logical / Physical workspace tab - Here tabs allow you to toggle between the logical and physical workspaces

4. workspace - This is the area where topologies are created and simulations are Displayed

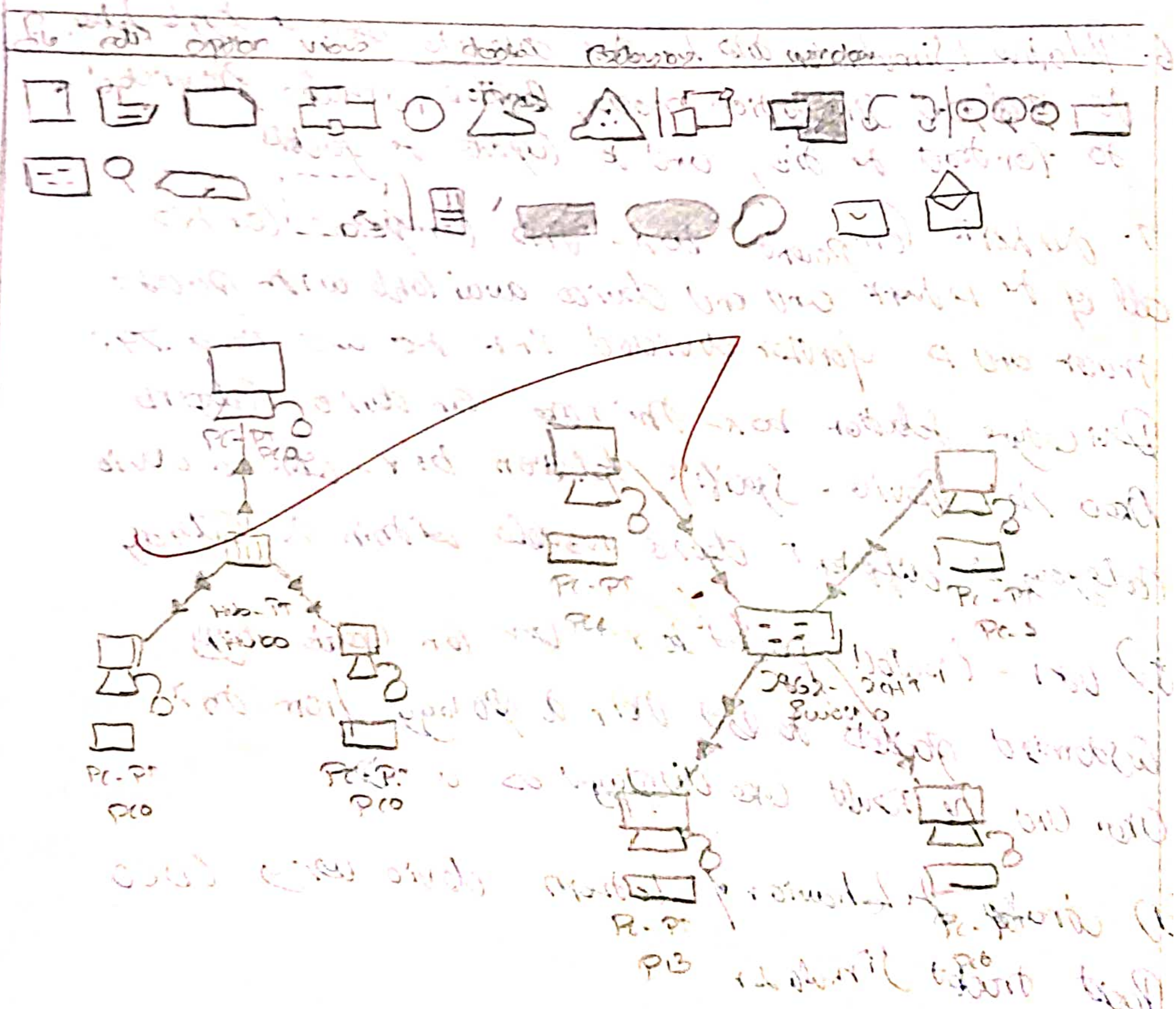
5. Common tools bar - This toolbar provides controls for manipulating it such as select mode, layout mode, delete, inspect, large shape, and add simple / complex PDU

6. Network component - box - The component box contains all of the network and end devices available with each floor and is further divided into two areas:
Area A: Device type selection box - This area contains device categories Area B: Device - specific selection box - This area contains Device - specific selection. This selection box Displays the different device models within the category

7) User - created packet box - Lets you create highly customized packets to be sent to topology from this area and the results are Displayed as a list

(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. 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 PCs and connect it to the Hub using the cable
4. Link LED should turn green indicating that the link is up
Similarly connect remaining PCs to the HUB
5. Similarly connect 4 PCs to the switch using copper straight-through cable



Q3. Click on the PCs connected to hub, go to the Desktop & then click on IP configuration, and enter the IP address and subnet mask. Here the default gateway and DNS server information is not needed as there are only two end devices in network.

Click on PDU (msg icon) from common tool bar.

a. Drag and Drop it over PC (Source machine) and then Drop it over PC (connected others)

4. Obs. the flow of PDU from Source PC to destination PC by selecting the Realtime mode of simulation

5. Repeat Step 1 to 3 for PCs connected to switch

Obs - how HUB and Switch are forwarding PDU and write your observation and conclusion about switch & HUB

Result:- The Study of Packet tracer tool and interface installation has been done successful

Ex2:- Student obs

• which command is used to find the reachability of a host/device from

(1) What is the diff betw cross cable and straight cable?

Ans: Straight cable

the wiring of both the end of the cable is identical
It used for connecting diff type of device

Cross cable

• the transmit and receive wires are crossed on an end of the cable
It is used for connecting similar device

(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 to a PC

Ex3 Student observation

(a) Behaviour of switch and hub in forwarding packets
A hub broadcasts packets to all devices, leading to more collision and network congestion
A switch forwards packets only to the specific device they're intended for, reducing traffic and improving efficiency

(b) Network topology implemented in our college
obs: Look at how devices like computer switches routers and hub are interconnected

Common network topology

Star
Bus
Mesh

Student obs:-
Practical 1

(1) Ping <host-name or IP address>

(2) traceroute <hostname or IP address>

(3) If config or IP address show

(4) netstat -toln

(5) if config eth0 192.168.1.100 netmask
255.255.255.0 up

Ex no: 6
Date

Set up and Configure LAN using Switch & ethernet cable

Aim:-

Setup and configure a LAN Local Area Network using a switch and ethernet cable in your lab

What is a LAN?

A Local Area Network (LAN) refers to a network that connects devices within a limited area, such as office building School or close. It enables users to share resources including data, printers and internet access

How to setup LAN

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

Step 2: You can take 4 computers, a switch with 8, 16, or 24 ports which is sufficient for network of this size and 4 ethernet cables

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

Step 4: Assign IP address to your PCs

1. Log on to the client computers as Administrator or as User
2. Click start and Internet Connection
3. Right click local Area Connection / Ethernet => go properties => Internet Protocol (TCP/IPv4)

click on properties -> select to go follow 1 IP address on Alor using IP address

Logon to the client computer an Administrator or an Owner

-> click network and sharing connection / connect -> go to properties -> select Internet Protocol (TCP) IPv4 -> click on prop.

Step:- Similarly assign IP address to all nodes connected to switch

PC1 - IP add 10.1.1.1 Subnet mask 255.0.0.0
PC2 - IP add 10.1.1.2 Sm 255.0.0.0
PC3 - IP ad 10.1.1.3 sm 255.0.0.0
PC4 - IP add 10.1.1.4 Sm 255.0.00.0

Step:- Configure

1. Connect your computer to the switch :- To access switch's web interface

2. Log in to the web interface

3. Assign IP address an 10.1.1.5 and subnet mask 255.0.0.0

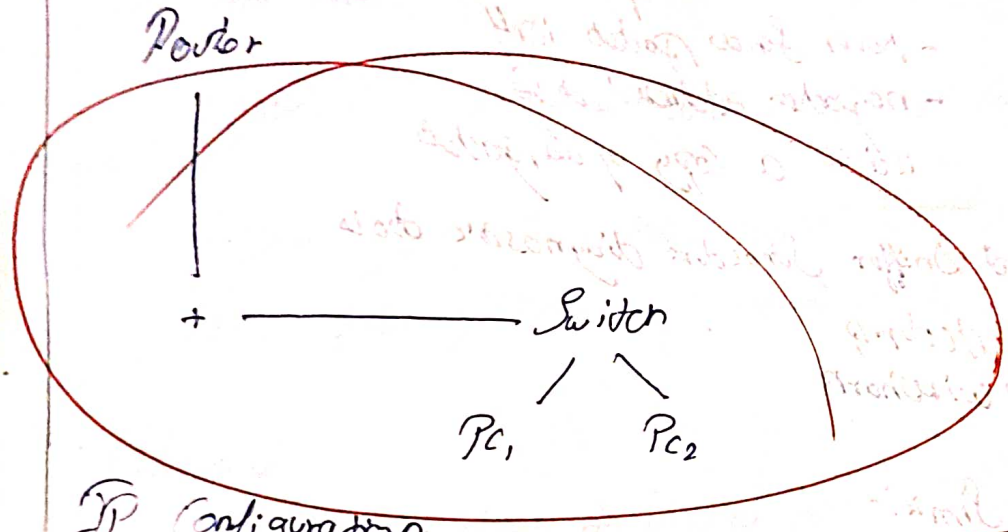
Step:- Check the connectivity between switch and other machine by using ping command in the command prompt

Step:- Select a folder -> go to prop -> click sharing tab -> share, with everyone in the same LAN

Step 6:- Try to ping the Should Yobot gun other computer of another LAN

STUDENT OBSERVATION:-

Draw a neat diagram of the LAN in the configuration obs book that you have implemented in your lab. write IP configuration of each and every device. write the outcome and challenges faced while configuring the LAN



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

Incomplete

Result:- Thus the Setups and Configuration of LAN executed Successfully

Practical - 5

Aim:-

Experiment on packet capture tool wireshark

Packet Sniffer:-

- Access and Display the contents of various protocol fields in messages
  - Passive Receives packets itself
  - Not packet addressed to it
  - Receives a copy of all packets

Packet Sniffer Structure diagnostic tools
- tcpdump
- wireshark

Wireshark:-

A network analyzer tool formerly known as Ethereal, Captures packet in real time and display them in human-readable format.

→ What we can do:- • Capture network traffic
• Decode packets
Protocols using display
• Analyse problems.

→ used for:-

Prop: learn network Protocol
interface network administration
Trouble Shoot network problems.

Getting wireshark:-

It can be download for windows or macos from its official website

Capture packets:-

After downloading & installing wireshark choose it & double click the name of an device interface under capture to start capturing packets

"packet list": display all packets in current capture

"packet bytes": show data of current packet, colour coding, filtering packet, flow graph

Procedure:-

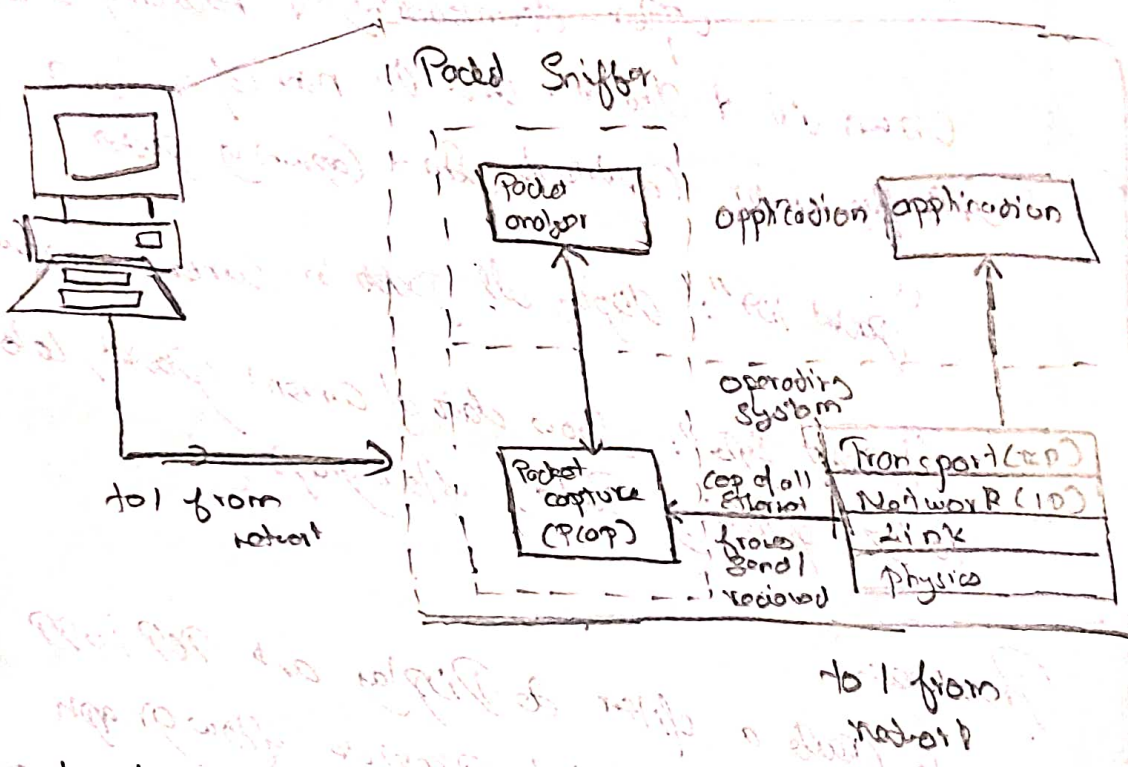
1. Create a filter to display all TCP/UDP packets, inspect the packet & provide flow graph
2. Create a filter to display only ARP packets and inspect packets
3. Create a filter to display only DNS packets and provide flow graph.
4. ~~Create a filter to display only HTTP packets~~ and inspect packets.
5. Create a filter to display only IP/ICMP packets and inspect the packets
6. Create a filter to display only DHCP and inspect the packets

Student Observation

1. What is Protocol Analyzer?

Packet Sniffer - Short Diagnostic Tools

- Tcpdump - Eg: tcpdump elohost 10.1294.2
- Wireshark - wireshark -r exe-2.cap



Wireshark:-

- Function
- Features
- Uses
- Installation

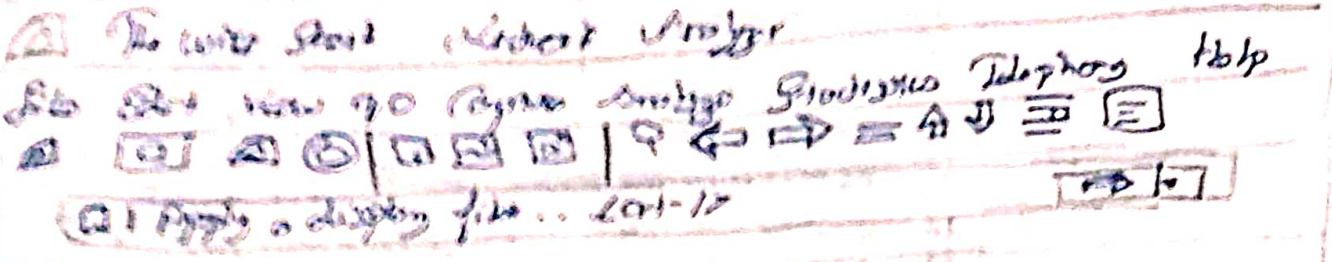
Incomplete

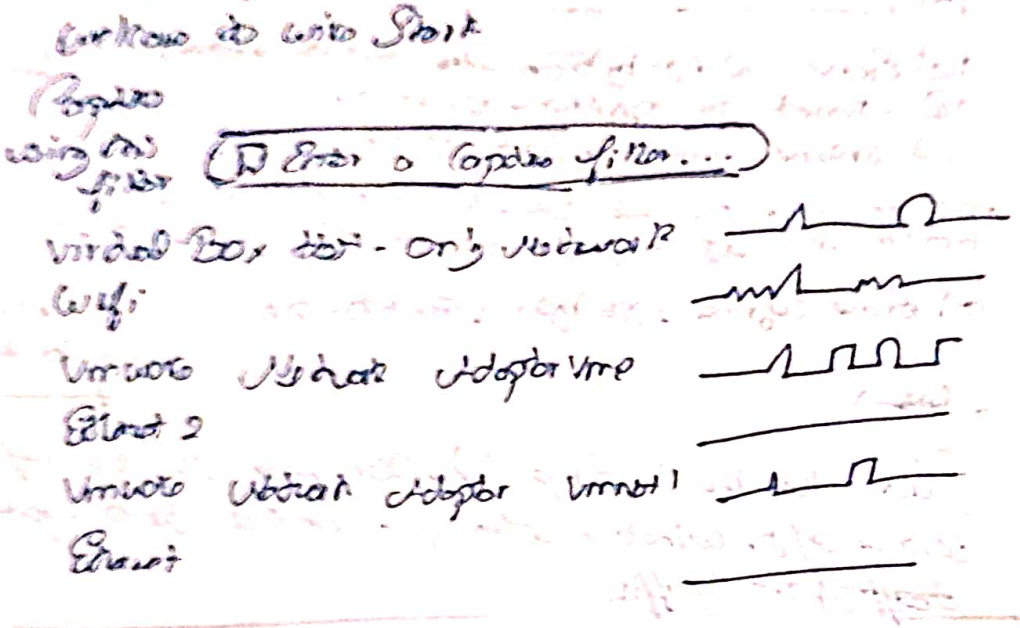
Result:-

Thus the experiment on Packet Analyzer tool, Wireshark has been ~~created~~ created successfully

rsolution

- Wireshark : official website
- Linux : Ratoy impossible

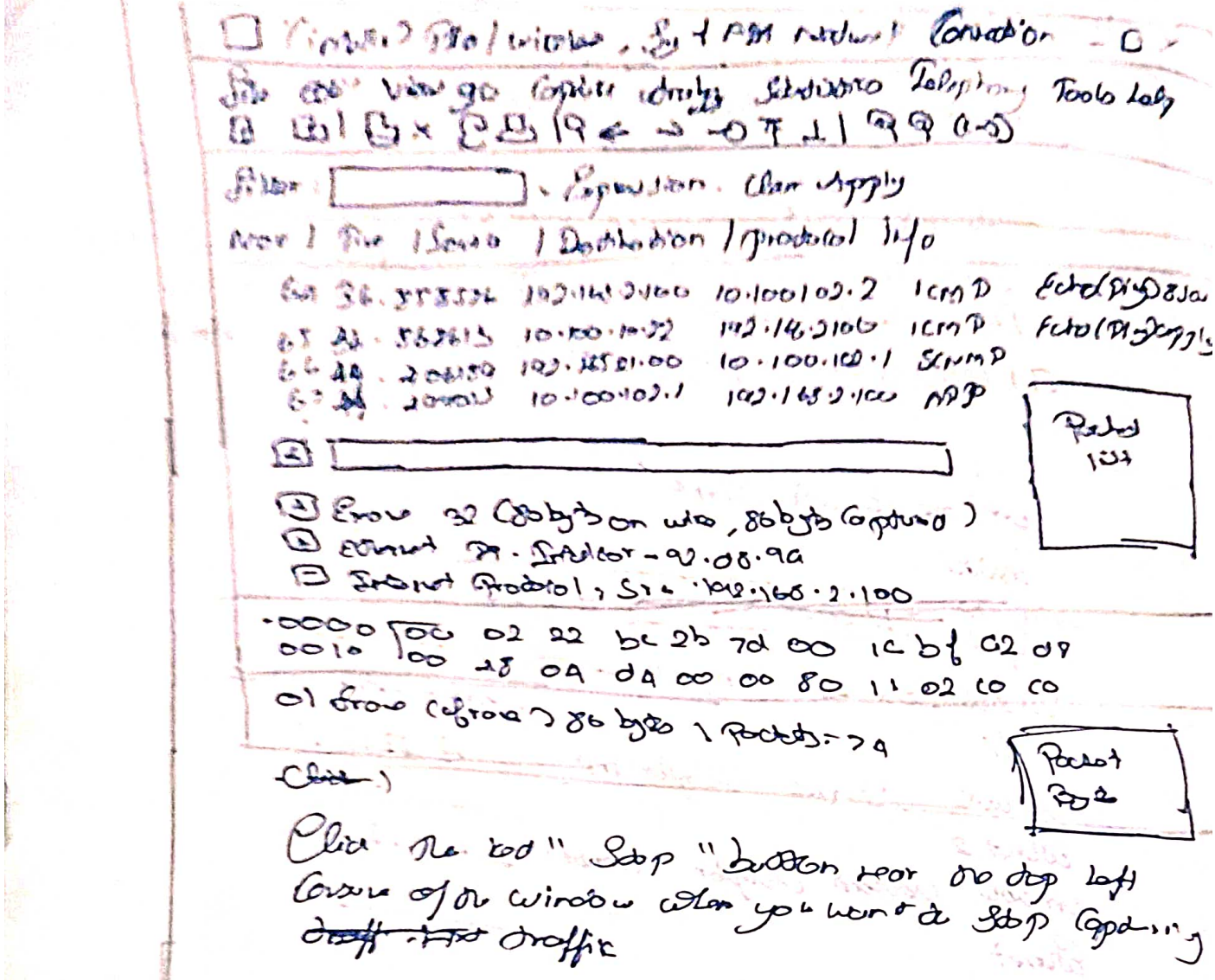




Usage :- Start capturing by selecting Network interface, promiscuous mode capture all packets

Click on top "Stop" button near to capture to left of the window close of the window when you want to stop capturing traffic

Packets list pane :-
- Display all packets in the capture file
- Selecting a wire shark detailed view in the other panes



Click the red "Stop" button near the top left corner of the window when you want to stop capturing traffic

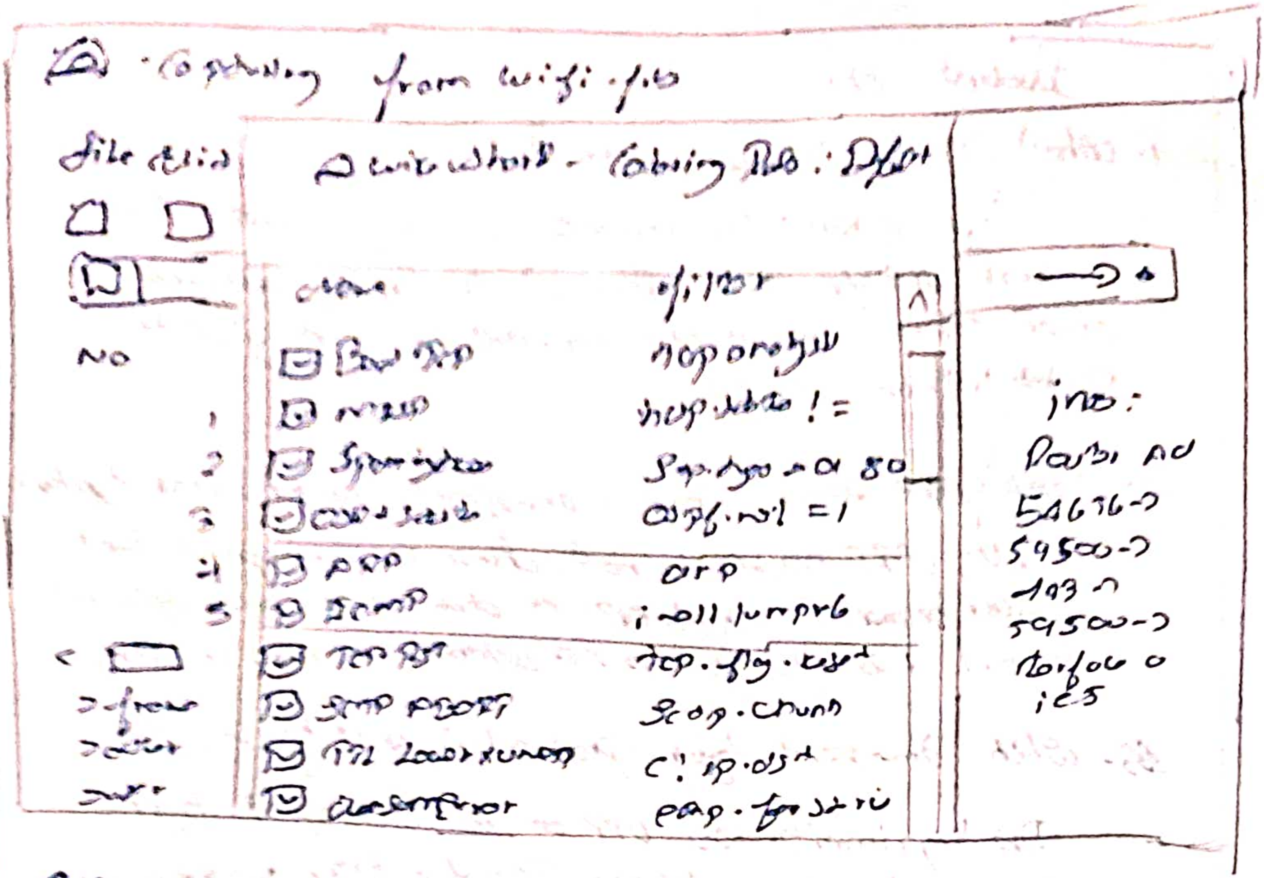
Color Coding

- Light Purple - TCP traffic
- Light Blue - UDP traffic
- Black/Red - Packets with errors
- Customization - View or modify color rules under View > Coloring Rules

Sample

→ Use Sample file to practice in wireshark Open via File > Open

→ Save your captures with file > Save for later review

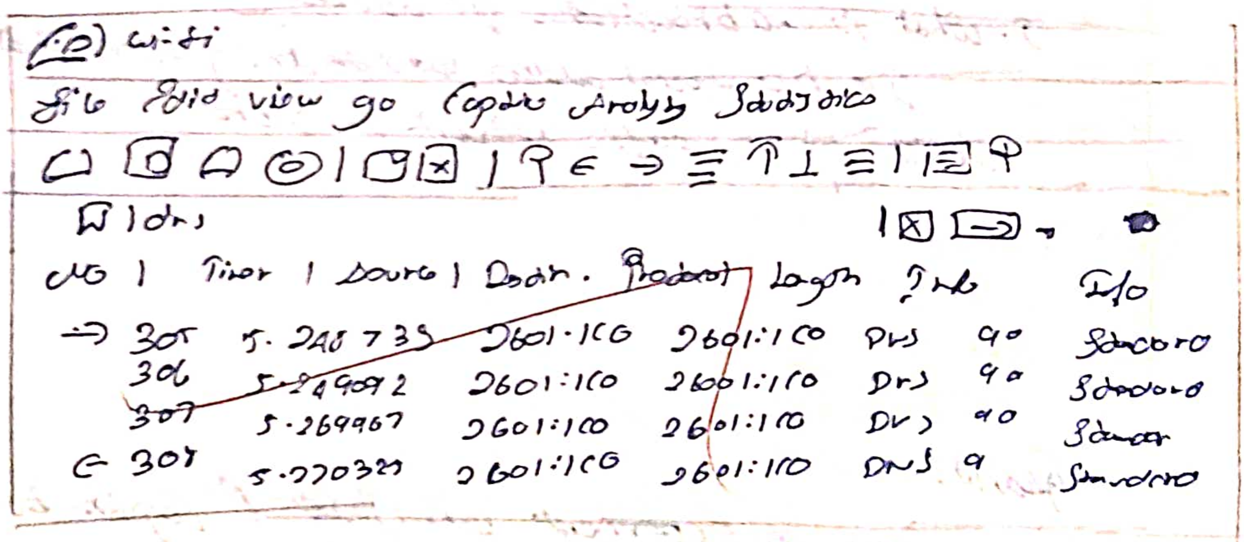


Filtering Packets:-

-> Apply filter to focus on specific network traffic

-> Use display filters to isolate traffic for analysis

Type a filter, press Enter, e.g. 'dns' for DNS packet wireshark
- auto complete



Use Analyze -> Display filters to pick a saved filter. Select a packet for more info. Right click a packet, choose follow TCP stream to see the full conversation. Use follow for other protocols too.

Student Obs:-

1. What is promiscuous mode?

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

2. Does ARP packets have a transport layer? Explain

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

3. What transport layer protocol is used by DNS?

DNS primarily uses UDP as its transport layer protocol but it also uses TCP for large queries

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

The default port number used by HTTP protocol is 80 for HTTP, the secure version of HTTP, the default port is 443

5. What is a broadcast IP address?

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

Result:-

Thus the experiment on Packet Capture tool is done Successfully

6
Error-

Hamming code

Aim:- Write a program to implement error Detection using hamming code concept. Make a test run to input data stream & verify error detection process

Error correction at data link layer:-

It is one of technique that can be used to detect & correct the error that occur when data is transmitted from sender to receiver

Code of sender program with below specifications

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

Code Receiver program with below specifications

1. Receiver program should read input from file
2. Apply hamming code on binary data to check error
3. If error, Display position of error
4. Also remove redundant bits from binary data & ASCII & display output

elif base < len (frames):
print ("\nResending unacknowledged frame... \n")
time.sleep(1) # Simulate delay
before resending

print ("In all frames sent are Acknowledged !")

if __name__ == "__main__":
sliding_window_protocol()

Code:-

```
import numpy as np
def text_to_binary (text):
 return ''.join (format (ord (char), '08b') for char in
                                                  text)

def binary_to_text (binary):
chars = [binary[i:i+8] for i in range (0, len (binary), 8)]
 return ''.join ([chr (int (char, 2)) for char in chars])

def calc_redundant_bits (m):
 r=0
 while (2**r < m+r+1):
   r+=1
 return r

def pos_redundant_bits (data, r):
 j=0
 k=0
 m=len(data)
 res=''
 print ("placing redundant bits at
 position:", end="")
```

```
for i in range (1, m + n + 1):
 if i == 2 ** j:
   res += '0'
  print(i, end = " ")
   j += 1
  else:
   res += data[k]
   k += 1
  print()
  return res

def calc_parity_bits (arr, r):
n = len(arr)
arr = list(arr)
parity_output = "Parity Bits: "
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)

   parity_output += f" (Position(position): (parity))"
  print (parity_output.strip())
  return "".join(arr)

def detect_and_correct (data, r):
 n = len (data)
 res = 0
 for i in range(r)
  parity = 0
  position = 2 ** i
```

```
for i in rang (1, r+1):
  if j & position:
  Parity += int (data [j-1])
  if parity != 0:
    ro += position

if ro! =0
Print (f" Error detected at position: {ro}")
data = list (data)
if ro <= n:
data [ro-1] = '0' if data [ro-1] == '1' else '1'
Print (f" Error Corrected at position: {ro}")
else:
  Print ("Error position out of rang. No Correction
Performed.")
  Corrected_data = ''.join (data)
  return Corrected_data
  else:
  Print ("No error detected.")
  return data

Def remove_redundant_bits (data, r)
 j=0
 original_data= ''
 for i in rang (1, len (data)+1):
 if i == 2**j:
  j+=1
 else:
   original_data += data [i-1]
 return original_data
```

```
def introduce_error(data, position):
    if position < 0 or position >= len(data):
        print("Error: Position is out of bounds.")
        return data
    data = list(data)
    data[position] = '0' if data[position] == '1' else '1'
    print(f"Introduced error at position {position}")
    return ''.join(data)

def sender(text):
    binary_data = text_to_binary(text)
    arr = len(binary_data)
    r = calculate_redundant_bits(arr)
    all = pos_redundant_bit(binary_data, r)
    out = calc_parity_bit(arr, r)
    print(f"Sender output (binary with redundant bits): {out}")
    return out

def receiver(data):
    r = calc_redundant_bit(len(data))
    print(f"Binary with error: {data}")
    corrected_data = detect_and_correct_data(r)
    original_data = remove_redundant_bits(corrected_data, r)
    ascii_output = binary_to_text(original_data)
    print(f"Decoded text: {ascii_output}")

if __name__ == "__main__":
    input_text = input("Enter characters to encode: ")
    channel_data = sender(input_text)
    corrupted_data = introduce_error(channel_data)
    receiver(corrupted_data)
```

Output:- 0100001101101000011000010110111

(iii) Enter the text to be encoded: chandu
placing redundant bits at positions: 1 2 4 8 16 32
parity Bits [position 1: 1] [position 2: 1]
[position 4: 0] [position 8: 1] [position 16: 0]
[position 32: 0]
Final Output (binary with redundant bits):
01001101001101100100000110000100111001100
000111001001110101

Binary with error: 0000110100110110010000011000
11000010101011011001000111001
001001110101

Error Detected at position: 2
Error correct at position: 2

Binary after error correction: 11001101001011001
000011000010101011001100100011001
001001110101

Decoded text: chandu

Result:- Thus the above program has executed successfully

7: Practical

Aim: Write a program to implement flow control at data link layer using SLIDING WINDOW Protocol. Simulate the flow of frames from one node to other

Program should adhere to the below given requirements. You can make it a bidirectional program where in receiver is ready its data frame with acknowledgement (Piggybacking)

Create a sender program with following features
1. Input window size from user
2. Input a text message from user
3. Convert 1 character per frame
4. Create a frame with following fields (frame no, Data)
5. Send the frame
6. wait for an acknowledgement from the receiver
7. Resend a file called Receiver - Buffer
8. Check ACK field for the Acknowledgement
9. If the Acknowledgement number is expected Get frame Accordingly else if resend it

Create a receiver file with following features
1. Reader a file called Sender - Buffer.
2. Check the frame no
3. If the frame no. also expected, write in appropriate ACK no in the receiver - Buffer file. Else write NACK no in the Receiver Buffer file.

Code:-

```
import time
import random
class frame:
    def __init__(self,frame_no,data):
        self.frame_no = frame_no
        self.data = data
        self.acknowledge = False

def send_frames(frames,window_size):
    print("\n--Sending frames---\n")
    for i in range(window_size):
        if i<len(frames) and not frames[i].acknowledge:
            print("Sent frame [{frames[i].frame_no}]:
            {frame[i].code}\n")
    print("Frames sent, waiting for acknowledgements..\n")

def receive_frames(frames,window_size):
    print("\n--Receiving frames--\n")
    for i in range(window_size):
        if i<len(frames) and not frames[i].acknowledge
            if random.random()<0.2:
                print(f"\nReceived frame [{frame[i].frame_no}]:
                {frames[i].data} [LOST]")
                frames[i].acknowledge = False
            else:
                print(f"\n Received frame [{frames[i].frame_no}]:
                    {frame[i].data} [OK]")
                frames[i].acknowledge = True
```

```
def Sliding window protocol():
    window_size = int(input("Enter window size: "))
    message = input("Enter a message to send: ")
    frames = [frame(i, message[i]) for i in range(len(message))]
    base = 0
    while base < len(frames):
        send_frames(frames[base:], window_size)
        time.sleep(2)
        receive_frames(frames[base:], window_size)
        while base < len(frames) and frames[base].
        acknowledged:
            base += 1
        if base < len(frames):
            print("\n\nResending unacknowledged frames.\n")
            time.sleep(2)
    print("\nAll frames sent and acknowledged!")

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

OUTPUT:

```
Send frame 3:
Enter window size: 3
Enter a message to send: Chandra

--- Sending frames ---

Sent frame 0: C
Sent frame 1: h
Sent frame 2: a
```

frames sent, waiting for acknowledgment...

---Receiving frames---
Received frame 0: C [OK]
Received frame 1: h [OK]
Received frame 2: O [OK]

Resending unacknowledged frames...

---Sending frames---
Sent frame 3: n [OK]
Sent frame 4: d [OK]
Sent frame 5: r [Error]
Frames sent, waiting for acknowledgments...

Resending unacknowledged frames...
---Sending frames---
Sent frame 5: d
Sent frame 6: U

Frames sent, waiting for acknowledgment...

---Receiving frames---
Received frame 5: r [OK]
Received frame 6: U [OK]

All frames sent and acknowledged!

Result:

Thus the above program is executed Successfully

Ex: 8

Aim: Switch Virtual LAN Configuration using Cisco packet tracer Simulation

Device	Interface	IP address	Subnet mask	Default gateway
S1	VLAN 1	192.168.1.11	255.255.255.0	N/A
S2	VLAN 2	192.168.1.12	255.255.255.0	N/A
R-A	WK	192.168.10.3	255.255.255.0	192.168.0.1
R-B	WIC	192.168.10.6	255.255.255.0	192.168.0.5

Part 1:- Build the Network and Configure Basic Device Setting

Part 2:- Create VLANs and Assign Switch Ports

Part 3:- Maintain VLAN Port assignment and the VLAN Database

Part 4:- Configure an 802.1Q trunk between Switches

Background / Scenario

Modern Switches use VLANs to enhance network performance by Dividing large layer 2 broadcast Domains into Smaller ones and enhancing security by controlling which communication VLANs Simplify network Design and meet Organizational goals

VLAN trunk Span VLAN across multiple Devices, enabling traffic from Different VLANs to travel over a single link while maintaining VLAN segmentation

In this process, prior Physical mode activity you will create VLAN on the switches assign them to Switch accessports and verify functionality. Then you'll create VLAN trunk to enable communication between hosts in the same VLAN, regardless of which switch they are connected to

Instruction:

Part 1:- Build Network & Configure Basic Setting

(1) Network setup.

- Drag Switches (S1, S2) to the Packet Tracer PCs (PC-A, PC-B) to the table.
- Use copper Straight - through cables for Connectivity and Console cable from PCs to Switches

(2) Switch Configuration:

- In the Desktop tab of each PC Console into each Switch via the PC terminal Enable Privileged Exec mode and Configure Device name, Password (Privileged Console by Password Encryption, banner, assign VLAN for Security, shut Unused Interface and set to Clear
- Save Configuration

(3) PC Configuration.

- Assign IPs to PC-A and PC-B, PC-D from the Addressing table

All 3rd Grounding
Ping between Device (PC-A to PC-B, PC-A to S1, PC-B to S2 and S1 only)

Part 2:- VLAN Creation of Port Assignment
(1) Create VLANs.
* Define VLAN (operation, parking lot, management) on both Switches
* Use a show vlan brief command to verify VLAN Creation
(2) Assign VLANs
* Assign PC-A to VLAN 10 (operation) on S1 and PC-B to VLAN 10 on S2
* Remove management IP from VLAN 1, assign it to VLAN 99 on both Switches
* Test Connectivity with ping.
(3) Verification
* Use the show vlan brief and show ip interface brief Commands to verify Status

Part 3:- Maintain VLAN Port Assignment and Database
Step 1:- Assign VLAN to multiple interface on S1, assign interface f0/11-24 to VLAN 99
S1 (config) interface f0/11-21
S1 (config-if range) # Switchport mode access
S1 (config-if range) # Switchport Access Vlan 99
* Reassign f0/11 and f0/21 to VLAN 10.

Step 2: Remove vlan assigned

* Remove vlan 99 from fa 1/0/3
  S1(config)# interface fa 1/0/3
  S1(config-if)# no switchport access

Step 3: Remove vlan assigned

* Remove vlan 99 from fa 1/0/3
  S1 (config)# interface fa 1/0/3
  S1 (config-if)# no switchport access

* Remove vlan 30:
  S1(config)# no vlan 30

Point 4: Configure Security and Remove Switches

Step 1: Use CTP to initiate trunking.

* Set fa 1/0/5 to negotiate trunk mode:
  S1(config)# interface fa 1/0/5
  S1(config-if)# switchport mode dynamic Desirable

* Verify VLAN traffic over trunk and issue show interface trunk

Step 2: Manually configure trunk

* Force trunking on fa 1/0/5 for both switches
* S1(config)# switchport mode trunk

* Change native VLAN to 1000:
  S1 (config)# interface fa 1/0/5
  S1 (config-if)# switchport trunk native vlan 1000

Student observation

(i) What is SSID of a wireless router

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

(ii) What is Security Key in wireless router?

- Security Key is the password used to protect a wifi network from unauthorized access
- Common types include WEP, WPA, however with WPA2 is height to most secure

(c) Configure a simple wireless LAN in your lab using a local access point and write down the Configuration in your note book

(1) Connect to access point (AP)

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

(2) Login to AP

Open a browser type the AP's IP address and log in with the username and password

(3) get the SSID

Name your network e.g. Lab-WLAN

(4) Set Security

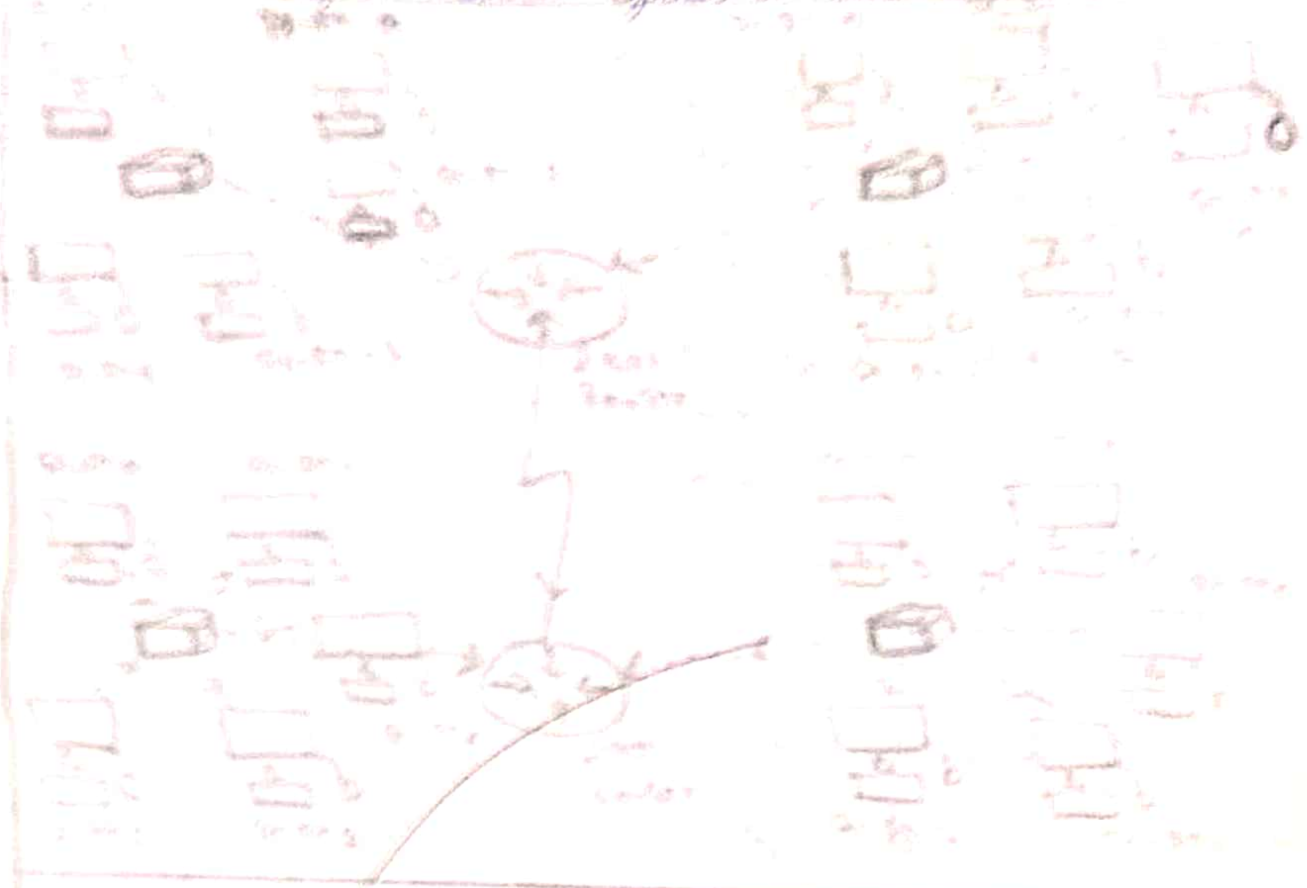
- Choose WPA2-password for security
- Set password: Duke123

(5) Save setting

Exp :

Aim: Simulate Class IP Subnetting using Cisco Packet Tracer.

Description: Class IP Subnetting efficiently divides a network by allowing flexible subnet masks, enabling the division of IP address spaces to smaller subnets.



Network Topology Creation:

* Created a new blank network topology
* Added routers, switches and PCs

2. Device Added

* Router R1
    * GigabitEthernet 0/0 : 192.168.1.1
    * GigabitEthernet 0/1 : 192.168.2.1

Switch S1

- Port connected to PC1: 192.168.1.0/27
- Port connected to PC2: 192.168.2.0/27

- PC on Subnet 192.168.1.0/27
  - 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

PC on Subnet 192.168.2.0/27:

- PC1: 192.168.2.11
- PC2: 192.168.2.12
- PC3: 192.168.2.13
- PC4: 192.168.2.14
- PC5: 192.168.2.15

3. Configuration Steps
   - Router Configuration
     - Configured interfaces with IP address and subnet mask using CLI
   - Switch Configuration
     - Set ports to access mode
   - PC Configuration
     - Assigned IP address, Subnet masks, and default gateways
4. Testing:
   - Pinging between PC and router confirmed both connectivity successful

Student discussion

(1) Write down your understanding of Subnetting

Ans:- Understanding of Subnetting: Subnetting divides a large IP address space into smaller subnets (subnets) for better management and organization of Devices

(2) What is one of imp Subnetting with example

Ans:- Efficient IP Management
Improved Performance
Enhanced Security
Simplified Management

(3) Find out whether Subnetting is implemented in your college If yes, draw and list down the subnets used in IP addressing

Ans:- Faculty network 192.168.10.128
Student network 192.168.200/24
Admin network 192.168.30.0/24
Lab network 192.168.40.0/24

Result:- Hence

Subnetting was successfully implemented, allowing efficient IP address management and networking among Devices

Ex 10

Date

Aim:- To do working with router in Cisco Packet Tracer Simulator

In this we have a router and 2 PC do use computers are connected with router using a Copper Straight-Through cable. After performing the network do check network is working PDU is transferred from PC0 to PC1

Procedure

Step 1. (Configuration Router)

1. Select the Router and open CLI
2. Press Enter to start configuration Router

Router > Command line interface

Router > enable

Router > Config terminal

Enter Config command one per line

Router (config)# interface fast ethernet 0/0

Router (config-if)# IP address

Router(config-if)# no shutdown
Router(config)#
Router(config-if)# interface fastethernet
Router(config-if)# no shutdown

(5).

1. assign Routerno to any PC in network
2. Select the PC, goto the Desktop and Select IP configuration and assign and Routers default gateway, Subnet mask
3. assign the default gateway of PC as 192.168.0.1

(3).

1. Connect fastethernet 0/1 port of PC with fastethernet 0/0 Port of Router 1 using a copper Straight Through cable
2. Connect fastethernet 0/1 port of PC, with fastethernet 0/1 Port of Router for 1 using a copper Straight Through cable

Router configuration table

Device name	Router fastethernet 0/0	Subnet mask	Router fastethernet 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

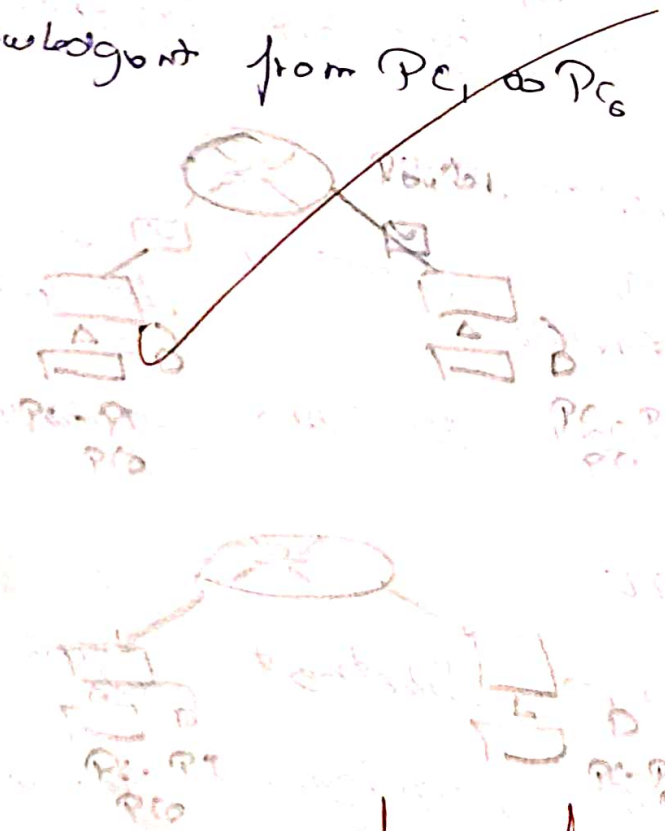
Device name	IP address	Subnet mask	gateway
PC0	192.168.10.2	255.255.255.0	192.168.10.6
PC1	192.168.20.2	255.255.255.0	192.168.20.1

Design wired topology

Simulate Design wireless topology

Ping a PC from Node PC1

Acknowledgement from PC_1 to PC_6



Result :- Do introducing with routes in cisco Packet Tracer Simulator is Done, and executed Successfully.

Exp 13
Date

Aim:- Design and configure a wireless network using wireless router DHCP server and internet cloud

Address table

Device	Interface	IP address	Subnet mask	Default gateway
PC	Ethernet	DHCP		192.168.3
Wireless Router	LAN	192.168.0.1	255.255.255.0	
Wireless Router	Internet	DHCP		
Laptop	Wireless	DHCP		

Part 1 - Build a simple network

1. Launch Packet tracer: open the packet tracer software
2. add devices
   * Add network devices (PC, wireless, router, cable connecting, internet cloud, Cisco (on (drive)) to the work space
   * Use the Device selection box to select and drag devices to the logical topology
3. Rename devices: change the display name of the devices using the config tab in the device windows
4. Add cabling

Use copper straight-through cable to connect the PC to a switch [illegible] the wireless router [illegible] and the external [illegible] to the cisco.com server

[illegible] console cable to connect [illegible] to the internet

Physical | Config | GUI | Attributes

Global Setting Algo Setting Interface Internet LAN Wireless	Global Setting Display name [Wireless Router0]

Part: Configure Network Devices

1. Configure Wireless Router
Set the SSID to "HomeNetwork" in the wireless tab in the setup tab enable DHCP and set the DNS server
IP to 208.67.220.220

Configure laptop
- Remove the existing module and replace it with a wireless module
- Connect the laptop to the router
- Use the command prompt to check IP settings with the ipconfig /all command

Configure cisco.com server
Setup the server as DHCP and DNS server
Pool name: DHCP pool
DNS 208.67.220.220

Security IP : 208.67.220.1

Configure the fastethernet 0 interface with added IPv4 208.67.220.220

Wireless - N Broadband router

Wireless | Setup | Wireless | Security | Access Restriction | Apps | Admin

Basic wireless setting

Network mode	mixed ▽
Network name (SSID)	Linksys ▽
Radio Band	Auto ▽
Standard channel	1 - 2.412 GHz
SSID Broadcast	Enable / Disabled

Help

Part 5 Verify Connectivity

1 - Refresh PC IP setting

In the PC's Command prompt use the Commands IP Config / release and IP Config / renew to refresh the IP Configuration. Verify that the PC has received an IP in the 192.168.0.xxx

Physical | Config | Desktop | Programming | Attributes

Command prompt

IP address 0.0.0.0

Subnet mask 0.0.0.0

Default gateway 0.0.0.0

DNS server 0.0.0.0

C:\>

C:\>

C:\>

C:\>

C:\>

C:\>

C:\> IP Config / renew

IP address 192.168.0.101

Subnet mask 255.255.255.0

Default gateway . . . 192.168.0.1
IPv4 Address . . . 208.67.220.220

C:\>

C:\>

C:\>

C:\>

C:\>

C:\>

C:\>

C:\>ping 208.67.220.220

Pinging 208.67.220.220 with 32 bytes of data:

Reply from 208.67.220.220 bytes = 32 time=13ms TTL

Reply from 208.67.220.220 bytes = 32 time=10ms TTL

Ping statistics for 208.67.220220

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

Approximate round trip times in milliseconds

Minimum = 13ms, Maximum = 10ms, Avg = 3ms

C:\>

Part a: Save and close Packet Tracer

Once anything is setup and verified save the configuration and close Packet Tracer

Lab observation :-

1. Key features of configuring wireless router and DHCP wireless router.

SSID : name of wireless network eg "home network"

Channel setting: Select the appropriate wireless channel to minimize interference

IP addressing : Configure the router internal IP (Default gateway of the network)

DHCP server :

- Enable DHCP : Activate the DHCP service to manage IP address assignment

2. Significance of DHCP server in network loading

The DHCP server simplifies network management by automatically assigning IP address subnet mask gateway and DNS server to devices. This dynamic allocation avoids conflicts manual IP configuration reducing speed and administrative workload in large network

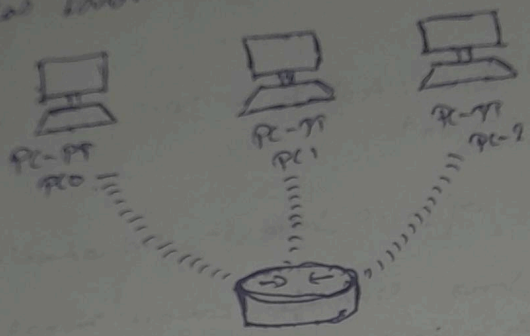
Result :- Thus the above program executed successfully

8

Wireless LAN

Aim:
Configuration of wireless LAN using Cisco Packet Tracer

Design a topology with the PCs connected from Linksys wireless router



Perform following Configuration

- Configure Static IP on PC and Wireless Router
- Set SSID as MohitNetwork
- Set IP address of router as 192.168.0.1 ; PC0 as 192.168.0.2 PC1 as 192.168.0.1 and PC2 as 192.168.0.4
- Save your Network by Configuring WAP key on Router
- Connect PC by using WAP key.

To complete above tasks follow these steps by step instruction

Step 1 :- Click on wireless Router.

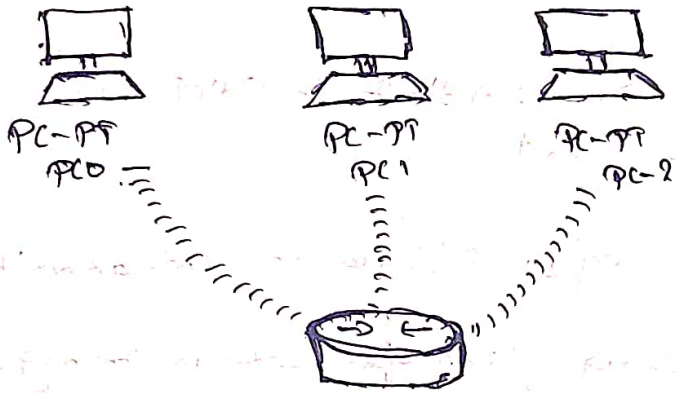
- Select administration tab from drop menu and enter username and password as admin

8b

Wireless LAN

Aim:
Configuration of wireless LAN using Cisco Packet Tracer

Design a topology with three PC connected from Linksys wireless router



Perform following Configuration

- Configure Static IP on PC and wireless Router
- Set SSID as MCANetwork
- Set IP address of router as 192.168.0.1 ; PC0 as 192.168.0.2 PC1 as 192.168.0.1 and PC2 as 192.168.0.4
- Secure your Network by Configuring WAP key on Router
- Connect PC by using WAP key.

To complete above tasks follow these steps by step instruction

Step1 :- Click on wireless Router

- Select administration tab from top menu username and password as admin

and Click on Save Setting

* Next click on wireless tab and Set Default SSID as Mother Network

* Now Select wireless security and change security Mode as WEP.

* Set key 1 as 0123456789

* Again go in the end of page and click Save Setting

* Now we have completed all given task on wireless router. Now Configure the Static IP on all three PCs

* Double click on Pc Select Desktop tab click on IP Configuration Select IP and Set IP as given below

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

* Now its time to connect PC's from wireless router. To do so click PC Select Desktop click on PC wireless

* Click on connect tab and click on Refresh button

* It will ask for WEP key 0123456789 and click connect

* It will connect you with wireless Router

* Repeat same process on PC1 and PC2

Learning Objs

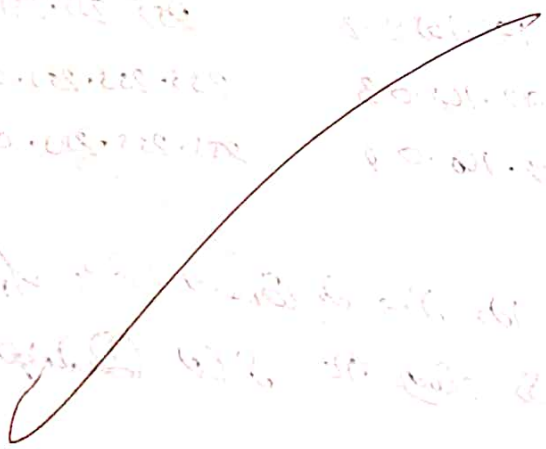
(1) What is ossid of a wireless Routing.

* To help Distinguish different networks in the Same area and can be customized by users make their own network more recognizable

(2) What is Security Key in wireless router?

* A Security key is the password used to protect a wireless network from unauthorized access

* Common types include WEP, WPA & WPA2 being the most Secure



Result:- Thus the wireless lan has created Successfully

11 Q

11/11/2024

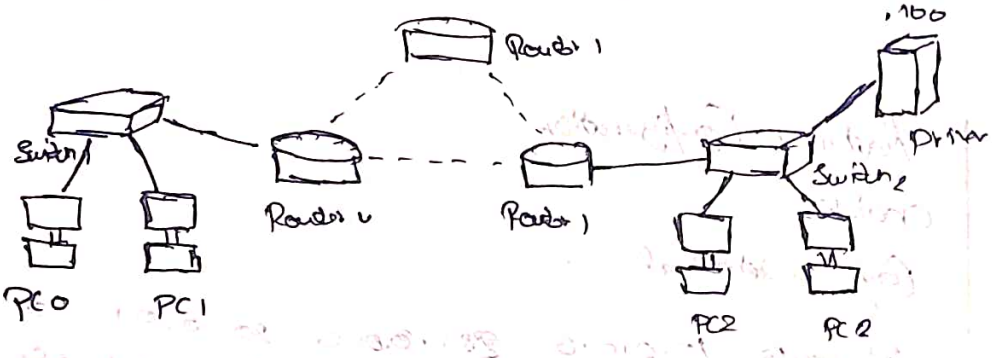
Static routing

Aim:-

Simulate Static Routing Configuration using Cisco Router Packet Tracer

Setting up a Practice lab

Create a packet tracer lab as shown in following image



The following table list the connected networks of each router

Router	Available network on level configuration	Network available on other routers interface
Router 0	10.0.0.16 20.0.0.16 40.0.0.16	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 Configuration

enable

Configure terminal (terminal)

ip route 20.0.0.0 255.0.0.0 20.0.0.2 10
ip route 90.0.0.0 255.0.0.0 40.0.0.2 20
ip route 20.0.0.100 255.255.255.255 40.0.0.2 10
ip route 90.0.0.100 255.255.255.255 20.0.0.2 20
ip route 30.0.0.0 205.0.0.0 40.0.0.2 10
ip route 850.000.0 255.0.0.0 20.0.2 20

Router 1 Configuration

enable

Configure terminal

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

Router 2 Configuration

enable

Configure terminal

ip router 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

Verifying static routing

Traceroute Command sends ping request to destination host and tracks the path they take to reach the destination

- Defines a static route
- Show ip route static command used to print all static routes.

+ Note Pour the rout from you want to Delete
+ Use the 'no ip route' Command to Delete the route

Result:-

Thus the static routing Configuration has been implemented Successfully

Ex No
Date

PPP using Cisco packet tracer

Aim:
Simulate PPP using Cisco Packet tracer

Initial IP Configuration

Device	Interface	IP Configuration	Connected with
PC	FastEthernet	10.0.0.2/8	Router0
Router0	Serial	192.168.1.254/30	Router1
Router0	Serial	192.168.1.249/30	Router2
Router1	Serial	192.168.1.200/30	Router3
Router1	Serial	192.168.1.240/30	Router2
Router2	Gig0/0	172.168.240.1	Router0
Router2	Serial	192.168.1253/30	Router3
Router2	Serial	20.0.0.1/8	PC1
C1	FastEthernet	20.0.0.2/8	Router2

Assign IP address to PCs

Double click PC and click Desktop then click IP Configuration assign IP address as per the above table

Assign IP address to interface of routers

Assign the IP address for all the interface of the routers as the Date given in the IP Configuration table

- Set the clock rate for DTE and not for the DTE end
- Show Controller interface gives whether the interface is DCE or DTE

05/10/23

- Configure RIP routing protocol

Router 0

router rip

network 10.0.0.0

network 192.168.1.252

network 192.168.1.208

Router 1

router rip

network 192.168.1.216

network 192.168.1.248

Router 2

router rip

network 10.0.0.0

network 192.168.1.252

network 192.168.1.244

- Access the Command Prompt of PC1 and use Ping Command to test the Connectivity from PC0

Output

Result:-

Thus the Simulation of RIP using Cisco packet tracer has been implemented Successfully

Exp. No:

Echo client server using TCP loop
Socket

Aim:

Implement Echo Client Server using TCP loop Socket

Algorithm:

(1) Create a TCP Socket
(2) Bind Socket to local address port
(3) Listen for incoming client connection
(4) Accept a client connect
(5) Loop
    - Receive Data from Client
    - If data is Received, send it back to client
    - Else, break loop
(6) Close the connection

TCP client algorithm

(1) Create TCP Socket
(2) Connect to send 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

```
Tcp_Serve_Tcp
import socket
def tcp_server()
Server_socket = Socket_Socket (Socket: AF_INET, Socket, Socket
                                                    STREAM)
Server_socket. bind ("localhost"12340))
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. recv (1024)
  if Data
      Print f"(Received. {Data. encode()}")
  else:
      Break
finally:
    Connection. Close()
  if name = "main__"
   tcp _Server()
tcp_Client()
     Client_Socket: Socket. Socket (Socket. AF_INET,
                              Socket. Sock, STREAM)
 Client. Socket. Connect (("local host" 12,345))
try:
 Message = input ("Enter a message to Send")
 Client_Socket. Send all (message. encode())
```

```
data = client_socket.recv(1024)
print("Received from server", data.decode())
finally:
    client_socket.close()
if __name__ == "__main__":
    tcp_client()
```

Output:-

Terminal:1

TCP_server.py

PS C:\Education\3rd\CN\Lab\Python_0> & "C:\Education\3rd\CN\Lab\env\Scripts\python.exe" "c:\Education\3rd\CN\Lab\env\TCP_server.py"

TCP server is waiting for a connection...

Connection to ('127.0.0.1', 53356)

Received: Hello this is TCP client

Terminal:2

TCP_client.py

Enter msg to send: Hello this is TCP Client

Received from server: Hello. this is TCP client

Result:-

Thus 2 program to echo client-server using TCP/UDP socket is executed successfully.

Ex 12 b

Chat Program using Socket Programming

Aim:-

To implement a chat client server using TCP/IP

Procedure

Chat Server.

1) Start the server by creating a socket, bind it to specific address and port, listen for incoming connections
2) Accept client connection and add it to list of connected clients start a new process to handle the client
3) For each connected client loop continuously receive messages
4) If a client disconnects, remove that client from the list & stop handling the client
5) Keep running the process indefinitely

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("Message received".encode('utf-8'))
```

```
except Exception as e:
    print(f"An error occurred: {e}")
    break

client.send(chat)

def start_server():
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(('127.0.0.1', 12345))
    server.listen(5)
    print(f"Chat server 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_client,
                         args=(client_socket,))
        client_handler.start()
if __name__ == "__main__":
    start_server()
```

```
chat_client.py
import socket
import threading
def receive_messages(client_socket):
    while True:
        try:
            message = client_socket.recv(1024).decode('utf-8')
            if message:
                print(f"\r{message}")
        except Exception as e:
            print(f"An error occurred: {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.")
  threading.Thread(target = receive_messages, args (client_socket
                      daemon = True).start()

  while true
     message = input ("You...")
     client_socket.send(message.encode('utf-8'))

  if name_ = "main_":
     start_client()
```

Output

Terminal: 1

chat_server.py

Chat server started on 127.0.0.1:12345

New connection from (127.0.0.1, 53020)

Received message from client: hello this is chatclient

Enter your message to client: hello this is chatserver

Terminal: 2

chat_client.py

Connected to the chat server

You: hello this is chat client

You: server: hello this is a chat server

Result:

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

Ex 13
6/11/2021

Ping Pong Program

Aim:

Implement Your own Ping Program

Algorithm

Ping_client.py

(1) Socket Creation

(2) Set Seta timeout of a second to ensure if no response is received it will stop waiting and print "Request" terminal

(3) Send a 'ping' msg to specific host & port

(4) If it waits for a response Calculate & the difference b/w sending to packet

Ping_Server.py

(1) Initialize UDP Socket

(2) Bind to IP address & port

(3) Listen for incoming messages

(4) Receive Data

(5) Send Response

Program

Ping_Server.py

import start_server (chost = '127.0.0.1', port=12300)

Def ... (AF_INET) socket.SOCK_DGRAM),

s.bind ((chost, port))

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

While true:

Data, addr = s.recvfrom(1024)

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

Ping_client.py:

```
import socket
import sys

def ping_server(host='127.0.0.1', port=12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        try:
            s.settimeout(3)
            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 1:

Ping_Server.py

UDP Server running on 127.0.0.1 : 12345

Received ping from ('127.0.0.1', 54321) : ping

Terminal 2:-

Ping_client.py

Reply from (127.0.0.1, 12345) in 0.00 seconds

Result:-

Thus the Ping Program has been Executed Successfully

Ex 11

Packet Sniffing

Aim:

Write a code using Raw socket to implement Packet Sniffing

Algorithm:

(1) Install Python and Scapy

(2) Create a program to open text editor and Create a file in notepad called packet_sniffer.py & code to capture and analyse the packets

(3) Set up packet handler by check if packet has IP layer, identify the packet protocol such as TCP, UDP, ICMP.

(4) Capture network packets

(5) Run the packet sniffer by using command

(6) generate network traffic by running the program

Program

```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP
Def packet_callback(packet)
if IP in packet_callback (packet)
if ip_layer = packet (IP)
Protocol = IP_layer.proto
src_IP_layer_src
dst_IP_IP layer.Dst
Protocol name = ''
if protocol == 1
```

```
        Protocol = "Simple"
    elif protocol == 1
    Protocol_name = 'ICP'
    elif protocol == 17
        Protocol_name = UDP
else
Protocol_name = 'Unknown Protocol'
Print ("Protocol = {Protocol_name}")
Print ("Source_IP = {Src_IP}")
Print ("Destination IP = {Dst_IP}")
Print ("=" * 30)
Def main()
```

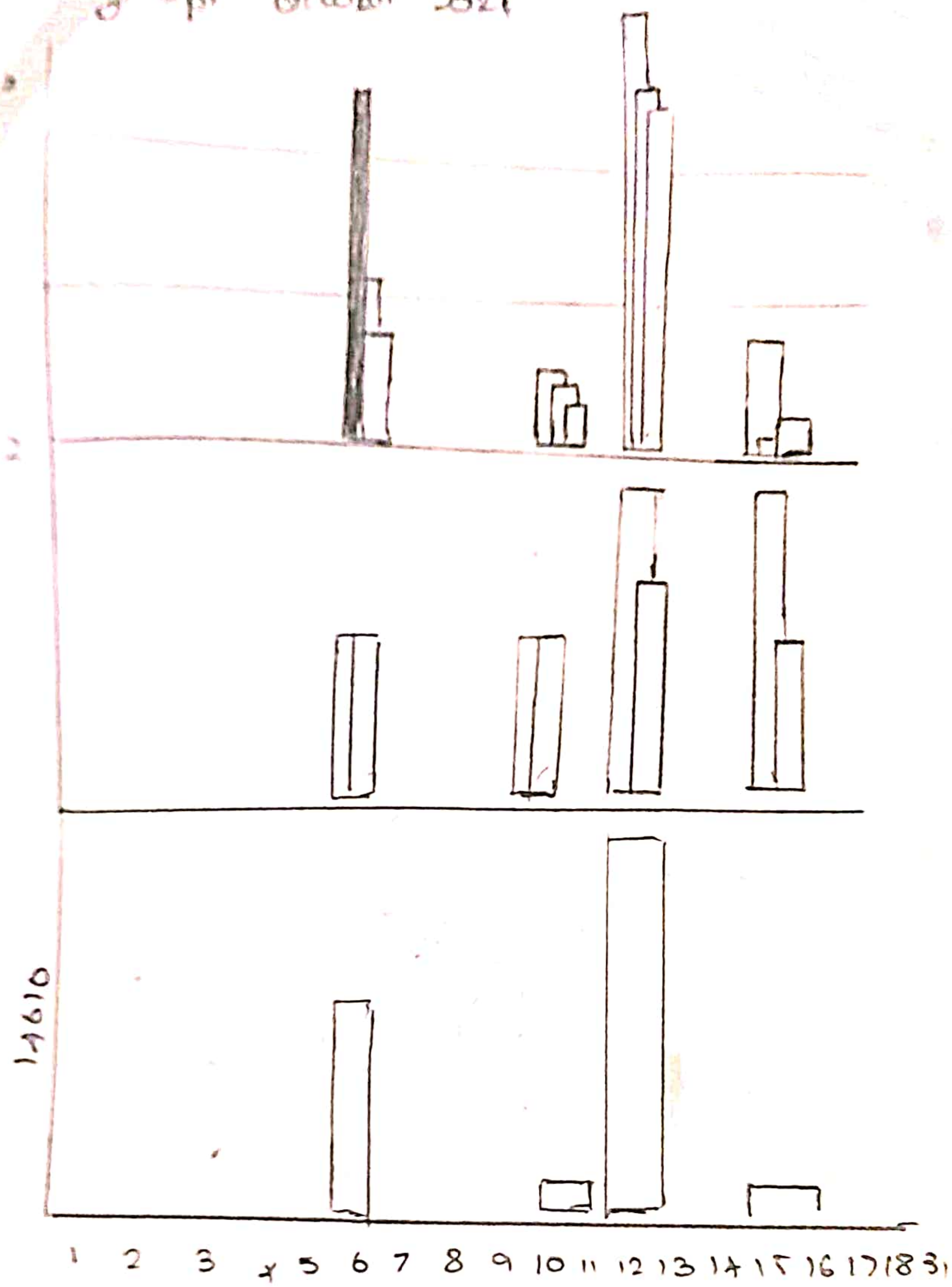
Output:-

Protocol = UDP
Source IP = 192.168.38.20
Destination IP = 240.0.0.251
Protocol = UDP
Source IP: 192.168.32.36
Destination IP: 240.0.0.251
Protocol = UDP
Source IP: 192.168.37.82
Destination IP: 8.8.8.8

Result:- Hence

Thus a packet Sniffing program has been executed Successfully

... upto october 2021



Result:

Thus the different types of weblogs has been analysed using webalizer.