INDEX

NAME: Ashwin.V STD.: V SEC.: A ROLL NO.: 518 SUB.: COMPUTER NETWORK

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	13/07/24	Network Commands	1	
2.	20/07/24	Different Types of Network Cables	9	
3.	30/7/24	Experiment on Cisco Packet Tracer	11	
4.	10/8/24	Set up and configuration a lan with a switch and network cable	13	
5	13/8/24	Experiment on packet capture tools - wireshark	13	
6	3-9-24	Error Correction at Data Link Layer	20	
7.	3-9-24	Flow Control at Data Link Layer	30	
8 a)	04-10-24	~~CISCO~~ Packet tracer Simulation	35	
9.	29-10-24	Subnetting In CISCO	40	
10a)	29-10-24	Interhetworking with Routers	43	
10b)	30-10-24	Configure an Interhetwork using Wireless Router	49	
11a)	04-11-24	Simulate Static Routing Configuration using CISCO	54	
11b)	04-11-24	Simulate RIP using CISCO	56	
12a)	04-11-24	Echo Client Server using TCP/UDP	60	
12b)	04-11-24	Implement Chat Client Server using TCP/UDP	63	
13)	04-11-24	Implement your own ping program	67	
14)	04-11-24	Packet Sniffing	70	
15	4-11-24	Wireshark.		
8a)	04-10-24	Simulate virtual lan configuration using CISCO Packet Tracer	35	
8b	04-10-24	Configure of wireless LAN using CISCO Packet Tracer	37	

completed

21/11

Exp: 1 Study of Basic Networking Commands

Date: 13/07/24

Used in Linux and Windows

AIM:

Study of Various Network commands used in Linux and Windows.

BASIC NETWORKING COMMANDS

arp -a : ARP or address resolution protocal reveals the IP address of your computer along with the IP address and MAC address of your router.

Output:

Interface : 172.16.75.14 ... 0x12

Internet Address	Physical Address	Type
172.16.72.1	7C-5a-1C-CF-bE-41	dynami
172.16.72.133	4C-ac-a3-63-97F3	dynami
172.16.72.195	2a-3C-13-9a-CC-17	dynami

hostname: simplest of all TCP/IP commands. It si display the name of your computer.

Output:

DESKTOP-C01BH7O

IPConfig /all: This command display detailed Configeration Information about your TCP/IP Connect Including the router gateway. DNS, DHCP Setti and the type of Ethernet.

OUTPUT

Windows IP Configuration

Host Name . . . . . : Desktop-COIBITTO

Primary DNS Suffix . . . :

Node Type . . . . . : Hybrid

IP Routing Enabled . . . . : No

WINS Proxy Enabled . . . : No

Netstat -a: This command help to solve problems with NetBios name resolution (Nbt stands for NetBios over TCP/IP)

Output:

Displays Protocol Statistics and Current TCP/IP Connection using NBT (NetBios over TP/P)

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

-a (adapter Status) List the remote machine's name table given its name

-A (adapter Stus) List the remote machine name table given its IP

-c (cache) List NBT cache of remote [machine]

-n (names) names and their IP adders lists Local NetBios names.

netstat: Netstat is a command-line tool that displays Statistics about a computer active TCP/IP Connections including incomming and outgoing Connection, routing tables and Interface Statistics.

output:

Interface list

19... 18 ce 51 7f dbf8 -- Microsoft Wifi Virtual Adapter

14.. 12 ce 51 7f dbf8... Microsoft Virtual Adapter

17...1C ce 51 7f dbf8 -- Realtek RTL8852BE WiFi 6 802.11

nslookup: nslookup is a linux tool for DNS lookups & showing details like IP address, MX records for email, servers and NS records for name services. It has interactive & non interactive modes.

output:

Server: Unknown

Address: 2401:4900:56:9:2800

Non-authoritative answer

Name: www.google.com

Address: 2401:6800:4009:81a::2004

142.250.195.132.

Pathping:. Pathping is a windows command that combines Ping and tracert. It traces the route to a destination address and tests each route along the way for data loss rates.

output:

usage: Pathping [-g host-list] [-h maximum_hops] [-i

[-P Period] [-q num-queries] [-w timeo

[-4] [-6] target_name

Ping: Packet Internet Groper) command is the best way to test connectivity between two nodes. Ping use ICMP (Internet Control Message protocol) to communicate to other devices.

(1) # Ping hostname (Ping localhost)

(2) # Ping IPaddress (Ping 4.2.2.)

(3) # Ping fully qualified domain name (Ping

Output:

Options:

-t Ping the specified host until stopped to see statistics and continue - type Control

-a Resolve addresses to hostnames.

Route: The 'route' command in windows is used to displays and manipulates the IP routing table. It allows setting up static routing to specific hosts or network via a desired network interface.

Output:

Manipulates network routing tables.

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

-f Clears the routing tables of all gateway entries if this is used in conjuction with one of the commands

-p When used with ADD command, make a route persistant across boots of the system

-4 Force using IPv4

-6 Force using IPv6

some important

IP: Syntax:

[Root

: lo (Loopback

State unk

00:00:00:0

[Root

[Root

[Root

[Root

[Root

[Root

[Root

[Root

[Root

[Root

[Root

Some important linux

1) IP: Syntax: IP <options> <object> <command>

a) [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

b) [root@server~] # IP address add 192.168.1.254/24 dev enp203

c) [root@server~] # IP address del 192.168.1.254/24 dev enp203

d) [root@server~] # IP link set up

e) [root@server~] # IP link set ethodown

f) [root@server~] # IP link set eth0 promisc on

g) [root@server~] # IP route add default via 192.168.1.254

h) [root@server~] # IP route add 192.168.1.0/24 via 192.168.1.

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

j) [root@server~] # IP route delete 192.168.1.0/24 via

k) [root@server~] # IP route 10.10.1.4

1. Ifconfig

lo: flags=73<UP, loopback, running> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 4 bytes 240 (240.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4 bytes 240 (240.0 B)
TX errors 0 dropped overruns 0 carrier 0
collisions 0

3) MTR

MTR <options> hostname / IP

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

keys: Help display mode Restart statistics order of fields

Host	Loss %	Snt	Last	Avg	Best	Wrst	St
gateway	0.0%	163	1.2	1.6	0.3	6.1	

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

-F --Filename read hostname (s) from a file
-4 use IPv4 only
-6 use IPv6 only
b) --udp use UDP instead of ICMP echo

[Root@server ~] #
keys: help Display M
Host Loss
gateway 0.0%
192.168.22.2)
[Root@server ~]
keys: help Disp
Host Loss
gateway 0.0
4) tcp dump
[Root@ser
To th
[Root@ser
1. echo/40 [
2. any (PS
[UP, R
[Root@s
dropped
tcpdump
-v [v] ..

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

keys: help Display Mode Restart statistics order of fi

Host	Loss%	Snt	Last	Avg	Best	Wrst	St
Gateway (192.168.22.2)	0.0%	391	1.1	1.1	0.2	10.2	0.8

d) [Root@server ~] # mtr -C 10 google.com

keys: help Display mode Restart statistics order of fields

Host	Loss%	Snt	Last	Avg	Best	Wrst	Sta
Gateway	0.0%	4	0.7	7	0.4	1.3	0.3

4) tcp dump

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

To install -y tcpdump

[Root@server ~] # tcpdump -D

1. eth0/4a [UP, Running, Connected]
2. any (Pseudo-device 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

output:

Interface list

19... 18 ce 51 7f dbf8 .. Microsoft wifi Virtual Adapter

14.. 12 ce 51 7f dbf8 ... Microsoft Virtual Adapter

17...1C ce 51 7f dbf8 .. Realtek RTL8852BE WiFi 6 802.

Nslookup: Nslookup is a linux tool for DNS lookups showing details like IP address, MX records for email, servers and NS records for name services. It has interactive and non interactive modes.

output:

Server: Unknown

Address: 2401:4900:56:9::2800

Non-authoritative answer

Name: www.google.com

Address: 2401:6800:4009:81a::2004

142.250.195.132

Pathping: Pathping is a windows command that combines ping and tracert. It traces the route to a destination address and tests each router along the way for data loss rates.

output:

usage: Pathping [-g host-list] [-h maximum_hops] [-i address]
[-P Period] [-q num-queries] [-w timeout]
[-4] [-6] target_name

Ping: Packet Internet Groper) command is the best way to test connectivity between two nodes. Ping use ICM (Internet Control Message protocol) to communicate to other devices.

(i) # Ping hostname (Ping localhost)

(2) # Ping IPaddress (Ping 4.2.2.)

(3) # Ping fully qualified domain name (Ping

Output:

Options:

-t Ping the specified host until stopped to see statistics and continues - type Control-

-a Resolve addresses to hostnames.

Route: The 'route' command in Windows is used to displays and manipulates the IP routing table. It allows setting up static routing to specific hosts or network via a desired network interface.

Output:

Manipulates network routing tables.

Route [-f] [-p] [-4 | -6] command [destination] [Mask netmask] [gateway] [METRIC metric] [IF

-f Clears the routing tables of all gateway Interfaces if this is used in conjunction with one of the commands

-p When used with ADD command, make a route persistent across boots of the system

-4 Force using IPv4

-6 Force using IPv6

Some important linux

1) IP: Syntax: IP <Options> <Object> <command>

a) [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

b) [root@server~] # IP address add 192.168.1.254/24 dev enp203

c) [root@server~] # IP address del 192.168.1.254/24 dev enp203

d) [root@server~] # IP link set up

e) [root@server~] # IP link set eth0 down

f) [root@server~] # IP link set eth0 promisc on

g) [root@server~] # IP route add default via 192.168.1.254 dev

h) [root@server~] # IP route add 192.168.1.0/24 via 192.168.1.254

i) [root@server~] # IP route add 192.168.1.0/24 dev et

j) [root@server~] # IP route delete 192.168.1.0/24 via 192

k) [root@server~] # IP route 10.10.1.4

2. Ifconfig

lo: flags=73<UP, loopback, running> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 4 bytes 240 (240.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4 bytes 240 (240.0 B)
TX errors 0 dropped overruns 0 carrier 0
collisions 0

3) MTR

MTR <options> hostname/IP

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

keys: Help display mode Restart statistics order of fields

Host	Loss %	Snt	Last	Avg	Best	Wrst	St
gateway	0.0%	163	1.2	1.9	0.3	6.1	

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

-F --Filename read hostname(s) from a file
-4 use IPv4 only
-6 use IPv6 only
-u, --udp use UDP instead of ICMP echo

[Root@server ~] #
keys: Help Display M
Host Loss%
gateway 0.0%
192.168.22.2)
[Root@server ~]
keys: Help Disp
Host Loss
gateway 0.0
4) tcpdump
[Root@server
To th
[Root@ser
1. echo/40 [U
2. any (PS
[UP, Ru
[Root@se
dropped
tcpdump
-v [v]...

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

keys: help DisplayMode Restart Statistics order of fi

Host	Loss%	Snt	Last	Avg	Best	Wrst	St
gateway (192.168.22.2)	0.0%	391	1.1	1.1	0.2	10.2	0.8

d) [Root@server ~] # mtr -C -b google.com

keys: help DisplayMode Restart Statistics order of fields

Host	Loss%	Snt	Last	Avg	Best	Wrst	Sta
gateway	0.0%	4	0.7	7	0.4	1.3	0.3

4) tcp dump

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

To install -y tcpdump

[Root@server ~] # tcpdump -D

1. eno1/4o [UP, Running, Connected]

2. any (Pseudo-device 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 -c10

10 Packets captured

20 Packets received by filter

0 Packets dropped by kernel

[root@server ~] # tcpdump -p eth0 -c10 not arp

dropped privs to tcpdump

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo, link-type EN10MB (Ethernet) snapshot length 262144 bytes

[root@server ~] # tcpdump -l eth0 port 80 and host 8.8.8.8

dropped privs to tcpdump

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo,

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

dropped privs to tcpdump

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on lo,

[root@server ~] # tcpdump -i eth0 net 0.0.0.0 mask 255.255.255.0

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

[Root@Server1] # tcpdump -i eth0 net 10.0.0.0/24

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

[Root@Server1] # tcpdump -i eth0 Port 53

To capture only DNS Port 53 traffic

[Root@Server1] # tcpdump -i eth0 host 8.8.8.8 and Port 53

This is for a specific host

[Root@Server1] # tcpdump -i eth0 -c 10 host WWW.google.com and port-443

To capture only HTTPS traffic

[Root@Server1] # tcpdump -i eth0 Port not 53 and not 25.

To capture all port except Port 80 and 25

Ping:

usage: Ping [-t] [-a] [-n count] [-l size] [-f] [-i ttl] [-v tos] [-r count] [-s count] [[-j host-list] [-k host-list] [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p] [-4] [-6] target_name

[root@ serverx] # ping google.com

Ping google.com (216-58-200-142) 56(84) bytes of

64 bytes from maos5 co-in-f14.1e100.net

(216.58.200.142): icmp_seq=46tt

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

Ping google.com (142.250.199.100) 56(84) bytes

data -- google.com Ping Statistics

10 Packets transmitted or received 4 (0 errors

100% Packet loss, time 9197ms PIPE3.

nmcli Connection Show

Name	UUID	Type	Device
RecA4	H04a98ood-ff4 88ea-80b7das	802-11 wireless	--
TLFL3-84	14846364591aa40 -83fc-93467	802-11 wireless	--
VIVO71	978(84cf(456) -908-704860	802-11 wirele	--

nmcli connection modify "wired connection" IPv4 method into

nmcli connection UP RECA4.

Password or encryption key are required to access the wireless Network 'REC A4'

Warning: Password for "802-11-wireless-security" Not given in 'Password-file and nmcli

Cannot ask without '--ask' option
Connection Successfully activated (D-Bus active path
org/freedesktop/NetworkManager/ActiveConnection/2)

Student observation

1) Which command is used to find the reachable of the host

Ans: 'Ping'

2) Which command will give the details of hops taken by a Packet to reach the destination

Ans: 'Trace Route'

3) Which command is used to display the IP configuration of your Machine

Ans: 'IPconfig'

4) Which command display the TCP port Status in your mac

Ans: 'net Stat'

5) Write the steps to modify the IP Configuration in Linux

: nmcli connection modify "Wired Connection"

Results

Thus the Study of various network commands used in Linux and windows executed successfully.

Exno:2 Study of different types of Network Cable

Date: 20/7/24

Aim:

Study of different types of network cables

a) Understand different types of network cables

Different types of cables used in networking are:

i) Unshielded twisted Pair (UTP)

ii) Shielded twisted Pair (STP)

iii) Coaxial Cable

iv) Fibre optic Cable

Cable Type	Category	Maximum data rate	Advantage/ Disadvantage	Appli / Use	Image
UTP	Category 3	10 PS	Advantage: • Cheaper in cost • Easy to install	10 base T Ethernet	
	Category 5	UPTo 100 MBPS	Disadvantage • More prone to (EMI)	Fast Ethernet	
	Category 5e	1 GBPS		gigabit Ethernet	
STP	Category	10 GBPS	Advantage: • Shielded • Faster than UTP • less Susceptible to noise and Interface	gigabit Ethernet 10G Ethernet (55m) widely used in data center	

Category 7

RG-6
RG-59
RG-11

Fibre Optic Cable

STP	Category 7	10Gbps	Disadvantage • Expensive • Complex installation effort	Gigabit Ethernet 10 Gb Ethernet (100m)
Coaxial Cable	RG-6 RG-59 RG-11		Advantage • High bandwidth • Immune to interference • Low loss bandwidth • Versatile	Speed of Signal 100 Mbps Television network High speed Internet Connection
Fibre Optic Cable	Single mode Multi Mode	100Gbps	Advantage - High speed - High bandwidth - High Security - Long distance Disadvantage - Expensive - Require Skilled Installer	Maximum distance of fibre optic cable is around 100 meter

b) Make your own Ethernet Cross over Cable Straight Cable.

Tools and Parts needed

- Ethernet Cabling - CAT5e is certified for gigabit support, but CAT 5 cabling works as well, just over shorter distance

* A crimping tool: this is an all in one hel
tool shapes to push down the pins in the
and strip and cut the shielding of the cables

Step1: To start construction of the device, begins
threading shield on to cable

Step2: next strip approximately 65cm of cable
from both the ends the crimping tools need
round area to complete this task.

Step3: After you will need to untangle the wires
there should be four "Twisted Pair" orderings
back to the sheet, arrange them from top to
bottom and should be in arrangement A and
other in A

Step4: Once the order is current, bunch them
in a line and if there are any that stray
out further than other, snip them back to
create an even level. The difficult aspect
is placing these into the RJ45 plug without
with clip side placing away from you and
the gold pins facing towards you as shown.

Step5: Next push the cable right in. the hole
at the end of the plug needs to be just over
the cable shielding and if it isn't that means
that you stripped off too much shielding simply
simply the cables back a little more

Step6: After the
the plug needs
pass down

Step7: Lastly,
diagram (B) co
To test it, plug
directly

Student observed

b) What is th
cross cables
device while S

Which type
Ans: cross cab

3. Which type of
Ans: Straight cabl

4. Find out t
due to conn
Ans: The catego

5. Making a tw
the correct
Result:

Thus t
successfully

Step6: After the wires are securly sitting insert the Plug insert it into the crimping tool and Pass down

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

To test it, Plug it and attempt to connect two devices directly

Student observation

1) What is the different between cross and Straight cable.

Cross cables have crossed wiring for connecting similar device while Straight cable are parallel wiring for unlike

2. Which type of cable used to connect two PCs?

Ans. cross cable

3. Which type of cable used to connect a router/Switch to PC

Ans: Straight cable

4. Find out the Category of twisted pair cable used in your lab to connect the PC

Ans. The category can be typically cat 5e or cat 6

5. Making a twisted pair cable involves arranging the wires in the correct order, facing challenges with precise wiring.

Result:

Thus the cable connection is done analysed successfully.

Ex:3 Experiments on Cisco Packet Tracer (Simulation Tool)

Date: 30/7/24

Aim

To Study the Packet tracer tool installation and user interface overview

To understand environment of CISCO PACKET TRACER to design simple network

INTRODUCTION:

A Simulator as the name suggests, simulate network devices and its environment. Packet Tracer is an exciting network design, simulation and 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 troubleshooting skills via computer or an Android or ios based mobile device

3. It is available for both linux and windows desktop environment.

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

Installing

To down
can and
Credentia
and down
operatin
Laptop):

Windows
Install
Storage
named
file t
agreem

Linux

Shou
work
fil
to
to

Installing Packet Tracer:

To download Packet Tracer go to http://www.netcad. can and log in with your Cisco Networking Academy Credentials; then click on the Packet Tracer 7 data 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 licence agreement, choose a location and start the installa

Linux:

Linux users with an Ubuntu/Debian distribution should download the file for Ubuntu and those using fedora/Redhat/Cent OS must download the file for fedora. Grant executable Permission to this file be using unmod and execute it to begin the installation.

Command to Packet Tracer601 - 1836 Installer 64bit.exe
/ Packet Tracer 1836 Installer 32bit.exe

1. Menu bar - This is a common menu found in all software applications. It is used to open, save, print, change preference and so on

2. Main toolbar - This bar provides shortcut icons to menu option that are commonly accessed such as open, save, zoom, undo and redo.

3. Logical and physical workspace table: These tabs allow you to toggle between the logical and physical work areas

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

5. Common toolbar: This toolbar provides controls for manipulating topologies such as select, move layout, places note, delete, inspect, resize, reshape

6. Realtime / simulation tabs: These tabs are used to toggle between the realtime and simulation modes. Buttons are also provided to control to the time, and to capture the packets.

To Network ...
all of the ...
with Pack...
into two...
box - Thi...
The: Devic...
category ...
the dif...
8. User ...
Custom ...
from ...
as ...
a) Anal...
CISCO...
b) Pr...
dr...
2)

To Network Component box - This Component contains all of the network and end devices available with Packet Tracer and is further divided into two areas: Area 1: Device type selection box - This area contains device categories Area 2: Device-Specific Selection box - When a device category is selected, this selection box displays the different device models within the category.

3. User Created packetbox - User can create highly customized Packets to test their topology from this area and the recieved are displayed as a list.

4) Analyze the behaviour of network device with CISCO PACKET TRACER Simulation

1) From the network component box click and drag and drop the below components:

a. A Generic PCs and One HUB
b. A Generic PCs and One Switch

2) Click on Connections.

a. Click on Copper Straight Through Cable
b. Select one of the PC and connect it to hub

using the cable. The link LED should flash green indicating that the link is up & working.

Similarly connect remaining 3 PCs to the HB

5. Similarly connect 4 PCs to the Switch using copper straight-through cable

3. Click on the PCs connected to now go to the Desktop tab, click on IP Configuration and enter an IP address and Subnet mask. Here the default gateway and DNS server information is not needed as there are only two end devices in the map

Click on the PDU (envelope icon) from the common tool bar,

a. Drag and drop it on one of PC (source machine) then drop it on another PC (destination machine) connected to the HB

4. Observe the flow of PDU from source PC to destination PC by selection the Real time mode of simulation.

5. Repeat Step #3 to Step #5 for the PCs connected to the Switch

6. Observe how Hub and Switch are forwarding the PDU and write your observation and conclusion about the behaviour of Switch and Hub.

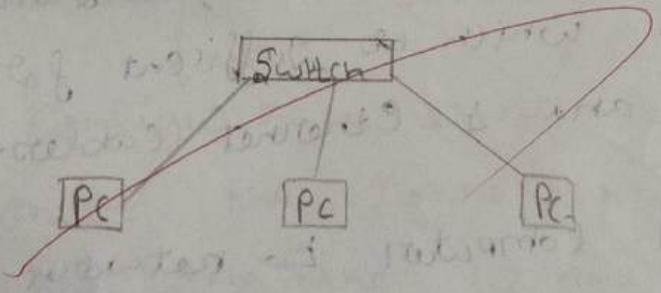
Students Observation:

As from your observation write down the behaviour of Switch and Hub in terms of forwarding the packet received by them

A Hub broadcasts packets to all ports, while a Switch forwards packets only to the destination host based on MAC address.

5) Find out the network topology implemented in your college and draw and table that topology in your observation.

A Star topology is implemented where each PC is connected to a central Switch or hub.



4/2/18

Result:

Thus the Packet tracer tool is used and executed successfully.

Ex: 4
Date: 10/8/24

Aim:
Setup and configure a LAN (Local Area Network) using a Switch and Ethernet Cables in your Lab

What is a LAN?

A Local Area Network (LAN) refers to a Network that connects device within a limited area, such as office building, school, or home. It enables users to share resource including data, Printer and Internet access.

How to set up a LAN?

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

Step 2: You can take 4 computers, a Switch with 8, 16 or 24 Ports which is sufficient for network of these sizes and 4 Ethernet cables.

Step 3: Connect your Computer to network Switch Via an Ethernet cable is simple as plugging one end of the Ethernet Cable into your Computer and the other end into your network Switch.

Step 4: Assign
1. Log on
or as ow
2. Click N
3. Right
to Proper
Click on
address

Similarly
to Swit
PC1 - IP a
PC2 - IP a
PC3 - IP a
PC4 - IP a

Step: 5

Step: 6

Step: 7
Shar
Sam

Step4: Assign IP address to your PCs

1. Log on to the client computers as Administrator or as owner.
2. Click Network and Internet Connections
3. Right click Local Area Connection/Ethernet -> to Properties -> Select Internet - Protocol (TCP/IP) click on Properties -> Select use the following IP address option and assign IP address

Similarly assign IP address to all the PCs connected to Switch

PC1 - IP address: 10.1.1.1, Subnet mask 255.0.0.0

PC2 - IP address - 10.1.1.2, Subnet mask 255.0.0.0

PC3 - IP address 10.1.1.3, Subnet mask 255.0.0.0

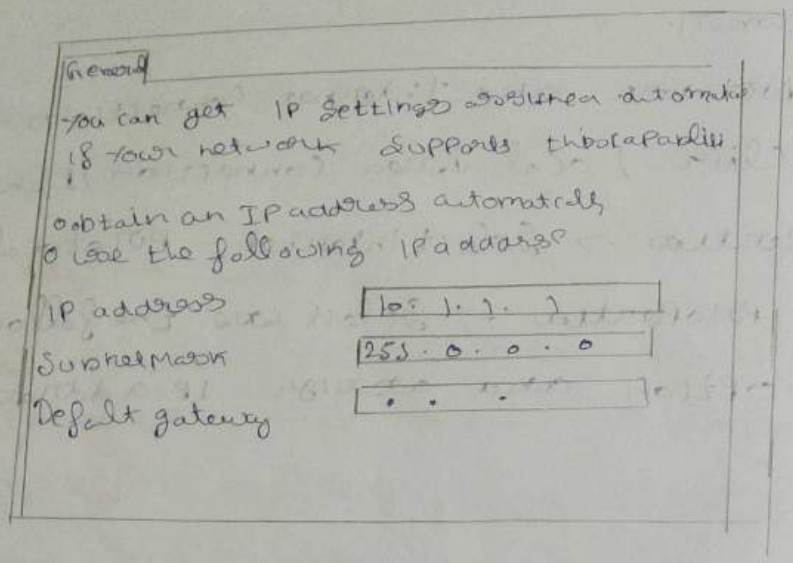
PC4 - IP address 10.1.1.4, Subnet mask 255.0.0.0

Step:5 Configure a network Switch

Step:6 Check the connectivity b/w Switch and other machine using Ping command in the command Prompt of the device

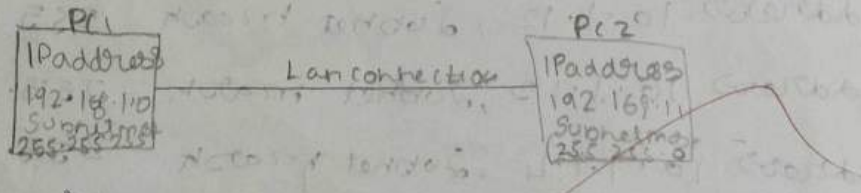
Step:7 Selet a folder -> go to properties -> Click Sharing tab -> Share it with everyone on the same Lan.

Step 8 Try to access the Shared folder from other computers to the network.



Student observation

Draw a neat diagram of LAN Configuration observation book. That you have Implemented in your lab write the IP Configuration.



* Advance Manage Share File Show the file between both device

* Now Search the file in Secondary device

Result:

Thus the LAN Connection has been established Successfully between two device

Ex: 5
Date: 13/3/24

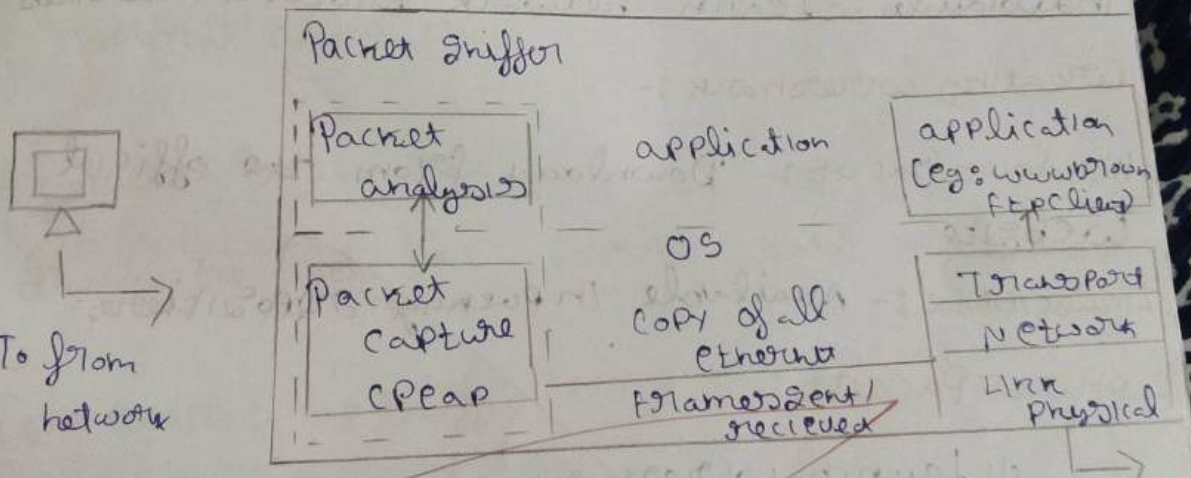
Aim:

Experiments on Packet capture tool: wireshark

Packet Sniffer: Captures messages sent / received by your computer

Function: Stores and displays protocol fields in messages

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



Wireshark:-

Wireshark is a network analysis tool that captures and displays packets in real time. It feature filter color coding and detailed description capabilities to analysis traffic and troubleshoot issue

Features:-

* Capture network traffic
* Decode Packet Protocales with dissecting
* Define captures and displays filters
* Monitor Smart statistics
* Analysis network problem
* Browse traffic interactively.

Uses:-

network administrator :- Troubleshoot network issues.

network Security engineers :- examine security problems

Developers :- Debug protocal Implementations

Individuals :- Learn network protocal Internals

Creating wireshark :-

Windows/MacOS :- Download from the official website

Linux/UNIX :- Available in Package repositories.

Captures Packets :-

1) Launch wireshark

2) Double click the network interface under Capture to start capturing packets.

Samples:-

- Use Sample files to practice in wireshark
  Open file via → open
- Save your captures with File → Save for later review.

Filtering Packets:-

→ Apply filters to focus on specific network traffic

→ Close 'others' apps to save traffic for analysis

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

Use analysis → displays filters to pitch or save filters. See the docs for more info.

Right click a packet choose follow > TCP Stream to see the full conversation use follow for other protocol tool.

Capturing and Analysis Packets using wireshark tool:-

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

Procedure:-

i) Select local area connection
ii) Go to capture → option
iii) Select Stop captures automatically after
iv) then Start capture

(v) Save the Packets

Create a filter to display only TCP/UDP Packets Inspect the Packets and Provide flow graph

(i) Select local area Connection
(ii) Go to Capture → option
(iii) Select Stop Capture after 100 Packets
(v) Click Start Capture.

2) Create a filter to display of ARP Packets and Inspect the Packet

(i) Go to Capture → option
(ii) Select Stop Capture after 100 Packets
(iii) then click Start Capture
(iv) Search ARP Packets in Search Bar
(v) Save the Packets.

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

(i) Go to Capture → option
(ii) Select Stop Capture after 100 Packets
(iii) click Start capture
(iv) Search DNS Packets
(v) See flow graph by Statistics → flow graph
(vi) Save the Packets.

4) Create a filter to display only HTTP Pa

PROCEDURE :-

i) Go to Capture → option
ii) Select Stop after 100 Packets
iii) Click Start Capture
iv) Search HTTP Packets
v) Save the packets

5) Create a filter to display only IP/TCP Packets and Inspect the Packets.

Procedure :-

i) Select local area connection
ii) Go to Capture → option
iii) Select Stop Capture after 100 Pa
iv) Click Start Capture
v) Search IMP / IP Packets
vi) Save the Packets

6) Create a filter to display only DHCP Packets and Inspect the Packets.

Procedure :-

i) Go to Capture → option
ii) Select Stop after 100 Packets
iii) Click Start Capture
iv) Search DHCP Packets

v) Save the Packets

Student observation

1) What is Promiscuous mode?

* This promiscuous mode is a configuration for a network interface that allows it to capture all packets on the network segment it is connected

2) Does ARP Packets have a Transport?

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

3. Which transport layer Protocol is used by DNS?

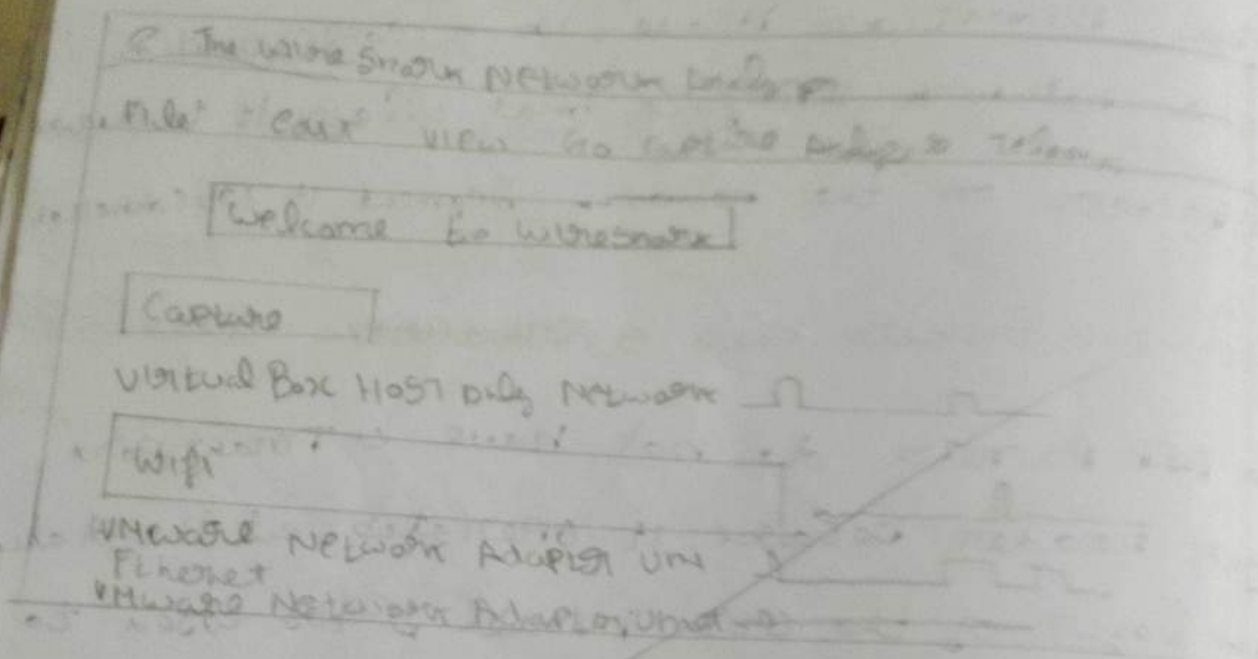
* DNS primarily uses UDP as its transport layer protocol but it also uses TCP for larger responses.

4. What is the port number used by HTTP Proto.

The default Port number used by HTTP Protocol is 80 for HTTPS (the secure version of HTTP, the default port is 443

5. What is broadcast IP address.

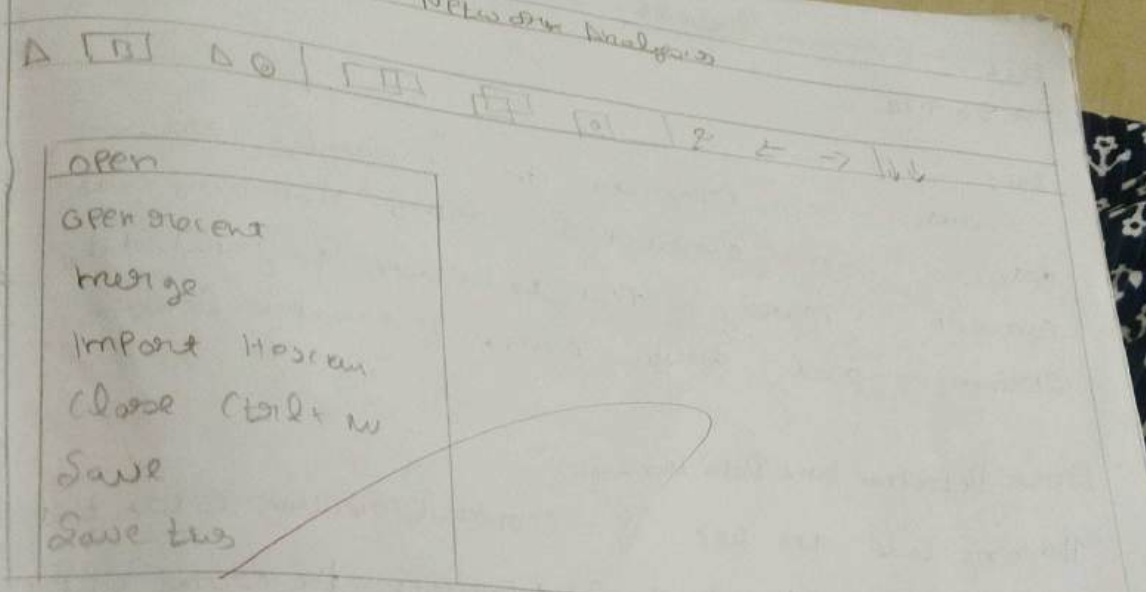
It is a special address used to send Packets to all devices on a specific network or Subnet.



Result:

to Sen
pecific

△ The Wireshark Network Analyzer



4/2018

Result:

Thus the Experiment on Packets Capture tool: Wireshark is verified Successfully

Practical 6

Ex: 6

Date: 30-9-24

Aim:

Write a program to implement error detection and correction using Hamming Code Concept. Make a test to run to input data stream and verify error correction feature

Error Detection and Data Link Layer

Hamming Code is set of error-correction codes that can be used to detect and correct the errors that occur when the data is transmitted from the sender to the receiver. It is a technique developed by R. Hamming for error correction

Create a Sender Program with below Features

* Input to Sender file should be a text of any length. Program should convert the text to binary
* Apply hamming code concept on the binary data and redundant bit to it

Create Receiver program with below Features

* Receiver program should read input from file
* Apply hamming code on binary data error
* If error, display by position of error

Program:

def string
return
def calc
r1 = 0
while (2
r1 +
return
def inser
m = len
r = calc
encoded
j = 0
for i
Pari
Pari
for
Par
for
if

Program:

```
def string_to_bits(s):
    return ''.join(format(ord(c), '08b') for c in s)

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

def insert_redundant_bits(data_bits):
    m = len(data_bits)
    r = calculate_redundant_bits(m)
    encoded_bits = ['0'] * (m + r)
    j = 0
    for i in range(r):
        parity_pos = 2 ** i
        parity = 0
        for j in range(parity_pos - 1, len(encoded_bits),
        parity_pos):
            for k in range(parity_pos):
                if j + k < len(encoded_bits):
                    parity ^= int(encoded_bits[j + k])
        encoded_bits[parity_pos - 1] = str(parity)
```

```
    return error_pos

def main():
    input_str = input("Enter the string to be
    data_bits = string_to_bits(input_str)
print("Original data bits: {}".format(data
n = calculate_redundant_bits(len(data_bits))
print("Number of redundant bits needed: {}".
encoded_bits = insert_redundant_bits(data
print("Encoded bits with redundant bits:
    {encoded_bits}")
if 0 <= bit_position < len(encoded_bits):
    encoded_bits_list = list(encoded_bits)
    encoded_bits_list[bit_position] = '0' else
    encoded_bits = ''.join(encoded_bits_list
    print("Bit at position {} changed.
    .: {}".format(bit_position + 1, encoded
else
    print("Invalid bit position entered. No
detected_error_pos = detect_and_correct(enc
if detected_error_pos:
```

output

Enter
original
Number
Encode
no
Enter bit
Bit at
1 o
Dete
Cor
Corr
Orig
Rea
Corr

```
encoded_bits_list = list(encode_bits)
encoded_bits)) if (i+1) & (i) != 0)
corrected_data_bits = ''.join(corrected_data_bits)
print ("corrected data bits: {}".format(corrected_da
original_message = ''.join(chr(int((corrected_data
print ("original message: {}".format(original_message))

if _name_ == "_main_":
    main()
```

output

Enter the String to encode :ASHWIN
original data bits: 0100000101001100100001110100011.
Number of redundant bits needed: 6
Encoded bits with redundant bits:

1101100110100101001101000100001011001101000

Enter bit position (1-base index) to change (or 0 to skip): 7
Bit at position 2 changed. updated encoded bits:

10010010001001001101010001000010101110100100

Detected error position at position: 2
Corrected encoded bits:

110110010001010011010000101011010010010101

Corrected data bits 010000 10101110090010100100
Original message : ASHWIN

Result: Thus the error detection and correction using hamming executed successfully.

Ex.No: 7

Date: 3-9-24

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 another.

Procedure:

* Initialize Frames: Creates frames for each character in the message
* Send Frames: Send a batch of frames determined by the window size.
* Wait for Acknowledgement: Simulates a delay and then recieve acknowledgement.
* Process Recived frames: Check if frames are acknowledged or have error.
* Slide window: Update the window base to next acknowledged frames.
* Resend Unacknowledged frames: Resend frames that were acknowledged
* Repeat: Continue until all frames are acknowledged

```
import time
import random
class Frame
def __init__
self.frame
self.data
self.ackno
def send_f
print("In
for i in ra
if i < le
print(
print("
def recei
print(
for i
if i
if r
prin
{ f
```

```
import time
import random
class Frame:
    def __init__(self, frame_no, data):
        self.frame_no = frame_no
        self.data = data
        self.acknowledged = false
def send_frames(frames, window_size):
    print("\n-- Sending Frames ---")
for i in range(window_size):
    if i < len(frames) and not frames[i].acknowledged:
    print(f"Send Frame {frames[i].frame_no}: {frames[i]
print(" Frames Sent, waiting for acknowledgement... \n")
def recives_frames(frames, window_size):
print("\n -- Reciving Frames ---")
for i in range(window_size):
if i < len(frames) and not frames[i].acknowledged
    if random.random() < 0.2:
print(f"Recived Frames {frames[i].frame_no}:
    {frames[i].data} [ERROR]")
```

```
            frames[i].acknowledged = False
        else:
            print(f"Received Frame {frames[i].frame_no}")
            frames[i].acknowledged = True

def sliding_window_protocal():
    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):
    ~~frames~~ print("\nResending unacknowledged fra
    time.sleep(2)

print("\nAll frames send and Acknowledged"); Result:

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

Output

Enter window size:
[i]. Enter a message

ent frame 0: n
Frame sent, wa
ent Frame 1: e
Frame sent, w
ent Frame 2:
Frame sent, w

Received Frame
Received Frame
Received Frame
Received Fra
Sent Frame
Frame Sen
Resving F
Received Fr

Succes

Output

```
Enter window size: 4
Enter a message to send: hello

Sent frame 0: h
Frame sent, waiting for acknowledgment
Sent frame 1: e
Frame sent, waiting for acknowledgment
Sent frame 2: l
Frame sent, waiting for acknowledgment

Received frame 0: h [ok]
Received frame 1: e [ok]
Received frame 2: l [ok]
Received frame 3: l [ok]

Sent frame 4: o
Frame sent, waiting for acknowledgment

Resending frame
Received frame 4: o [ok]
```

21/11

Result:

Thus the above Program was executed successfully.

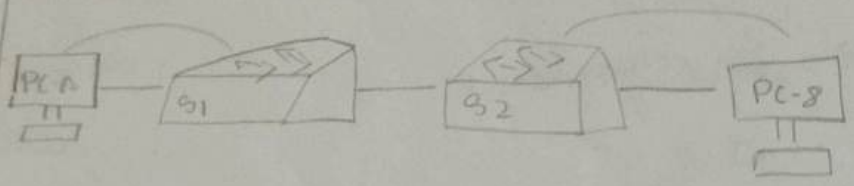
Practical 8A

Ex:8A

Date: 04-10-24

Aim:

Simulate 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 1	192-168-1-12	255-255-256.0	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

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Create VLANs and Assign Switch Ports

Part 3: Maintain VLAN Port

Part 4: Configure an 802.1Q Trunk between the

Switches.

algorithm is

Modern Swi

performance

remains

ecurity

Simplify

goals.

LAN Tra

enabling

travel

VLAN

n the

You w

Ports

a

betw

of

Switches.

Background / Creation

Modern Switches use VLANS to enhance network performance by dividing large layer 2 broadcast domains into smaller ones and enhancing security by controlling host communication. VLANS simplify network design to meet organization goals.

VLAN Trunks span VLANS across multiple devices enabling traffic from different VLANS to travel over a single link while maintaining VLAN segmentation.

In the Packet Tracer Physical mode activity you will create VLANS on two switches assign ports and verify functionality. Then you'll create a VLAN Trunk to enable communication between hosts in the same VLAN, regardless of which switch they are connected to.

Instructions:

Part 1: Build Network & Configure Basic

1) Network Setup:

* Drag Switches (S1, S2) to the Rack and (PC-A, PC-B) to the table

* Use Copper Straight-through cables for and Console cables from PCs to Switch

2) Switch Configuration.

* In the desktop tab of each PC console, each Switch via the PC terminal. Enter Privileged EXE mode and configure:

* Device name, Password (Privileged, Console, vty), Password encryption, banner message

* VLAN 1 IP for testing, Shut unused interfaces and Set the clock

* Save Configuration

3) PC Configuration

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

4. Test Connectivity.

* Ping between devices (PC-A to PC-B, PC-A to S1, PC-B to S2 and S1 to S2)

Part 2: VLAN Creation & Port Assignment

1) Create VLAN:

* Define VLANS (Operation, Parking, Lot, Management native) on both Switches.

* Use the Show VLAN brief command to verify VLAN creation.

2) Assign VLANS:

* Assign PC-A to VLAN 10 (Operations) on S1 and PC-B to VLAN 10 on S2

* Remove mangement 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 command to verify status

Part 3: maintain VLAN port Assignment database

Step 1: Assign VLAN to multiple interfaces

* on S1 assign interface Fa0/1-24 to VLAN

S1(config) Interface range fa0/1-24

S1(config-if-range) # Switchport mode access

S1(config if-range) # Switchport access vlan 99

* Reassign Fa0/11 and F21 to VLAN 10

Step 2: Remove VLAN assignment

* Remove VLAN 99 from Fa0/24

S1(config) # Interface fa0/24

S1(config-if) # no Switchport access vlan

Step 3. Remove VLAN from Database

* Add VLAN 30 to Fa0/24 without global con

S1(config) #Interface fa0/24

S1(config) # Switch port access Vlan 30

* Remove VLAN 30:

S1(config) # no Vlan 30

Part 4: Configure 802.1Q Trunk Between Switches.

Step 1: use DTP to Initiate Trunking

Set Fa0/1 on S1

S1(config) # Interface

S1(config-if) # S

verify VLAN Trunk

Interface trunk

Step 2: manually

force Trunking

S1(config) #Int

S1(config-if) #

change native

S1(config) # In

S1(config-if) #

Results:

Thus

* Set fall on S1 to negative Trunk mode:

S1(config) # Interface fall .

S1(config-if) # Switchport mode dynamic desirable

* Verify VLAN Traffic over Trunk and Issue Show Interface trunk

Step2: Manually configure Trunk.

- Force Trunking on fall for both Switches.

S1(config) # Interface fall

S1(config-if) # Switchport mode trunk

Change native VLAN to 1000:

S1(config) # Interface fall

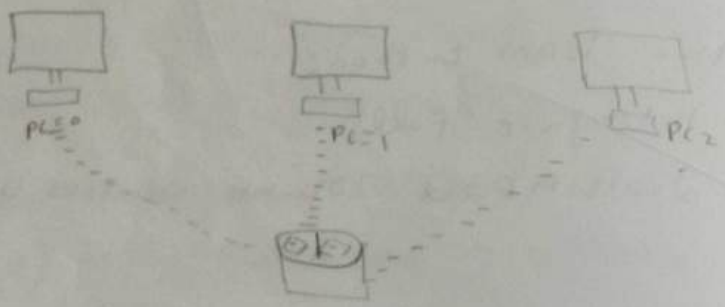
S1(config-if) # Switchport Trunk native vlan 1000

Result:

Thus the above code executed Successfully.

Aim:
Configuration of wireless LAN using Cisco packet Tracer.

Design a topology with three PCs connected to Linksys wireless Routers.



Perform following configuration:

* Configure State IP on PC and wireless IP on
* Set SSID to Motherhetwork
* Set IP address of Router to 192.168.1 PC0 Ok IP 192.168.0.2 PC1 to 192.168.0.3 and PC2 to IP ad 192.168.0.4
* Secure your network by configuring WAP key On Routes
* Connect PC by using Wap key

PC
PC0
PC1
PC2

To complete these tasks follow these step by step instructions:-

* Select administration tab from top menu. Set username and password to admin and click on save setting
* Next click on window tab and set default SSID to MotherNetwork
* Now select wireless security and change security mode to wep
* Set key 1 to 0123456789
* Again goto the end of page and click on Save Setting
* Now we have completed all give tasks on wireless routers. Now configure the static IP on all three PC's
* Double click on PC select Desktop tab click on IP configuration Select static IP and set IP as given below

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	192.169.0.4	255.255.255.0	192.168.0.1

* Now its time to connect PC6 from wireless Router. To do So click PC Select - Desktop on PC wireless
* click on connect tab and click on Refresh button
* It will ask for wap key Insert 0123456789 and click connect
* It will connect you with wireless Router
* Repeat Same process on PC4 and PC2

Student observation

(1) What is SSID of a wireless Router

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

(2) What is security key in wireless Router

* A security key is the password used to protect a wi-fi network from unauthorized access
* Common types include WEP, WPA and WPA2

with WPA2 is high the most secure

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

i) Connect the 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 APs IP address and log in with the username and password

3) Set the SSID

Name your network (e.g: Lab-WLAN

4) Set Security

* Choose WPA2-Personal for Security
* Set password : Secure123.

5. Save Setting

Aim:

Implementation of Subnetting in Cisco Packet Tracer Simulator

Classless IP Subnetting allows more efficient

Ex:9
Date: 29-10-24
Aim:
Implementation of
TRACER Simulator

Classless IP Subn
use of IP addres
that aren't
This enables
smaller Subne
space into
networks are
STEP for Imp
1) Create a
click "New"
topology
2) Add devi
the devices
by dragg
3) Subnetting
Subnet m
address

Result

Thus the Program executed successfully. and outp

Verified

Ex:9
Date: 29-10-24

Aim:

Implementation of Subnetting in CISCO PACKET TRACER Simulator

Classless IP Subnetting allows more efficient use of IP addresses by using Subnet masks that aren't restricted to default class masks. This enables dividing an IP address space into smaller subnets, ideal when limited IP addresses space into smaller sub are available but multiple networks are needed

STEP for Implementation:

1) Create a network topology in Packet Tracer click "New" -> "Network" -> "Generic" to start a blank topology

2) Add devices: Drag routers, Switches and PCs from the device list into the topology. Connect devices by dragging a cable between ports

3) Subnetting: for the network 192.168.1.0/24, use a /26 subnet mask to create subnets, each with 30 host address, This allows enough space for end devices

Switches and a router

Router R1

Gigabit Ethernet 0/0 : 192 : 168 : 1.1

Gig bit Ethernet 011 . 192 . 168 . 2.1

Switch S12

Fast Ethernet 011 . 192.168 . 1 . 0/27

PC . 1 : 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 012 . 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

Router R2 :

Fast Ethernet 0/0 : 192 . 168 . 3.1

Fast Ethernet 011 : 192 . 168 . 4.1

Switch S2 :

Fast Ethernet 011 : 192 . 168 . 3.0/27

PC1 : 192 . 168 . 3 . 11

PC2 : 192 . 168 . 3 . 12

PC3 : 192 . 168 . 3 . 13

PC4 : 192 . 168 . 3 . 1

PC5 : 192 . 168 . 3 . 1

Fast Ethernet

PC1 : 192 . 168

PC2 : 192 . 168

PC3 = 192 . 168 .

PC4 : 192 . 168

PC5 : 192 . 1

Configuring t

Router Conf

1) Righ clic

2) Enter t

Configure t

Interface F

Ip address

no shuta

* Fast Et

Fast Et

PC_1: 192.168.3.11

PC_2: 192.168.3.12

PC_3: 192.168.3.13

PC_4: 192.168.3.14

PC_5: 192.168.3.15

Fast Ethernet 0/2: 192.168.4.0/27

PC_1: 192.168.4.11

PC_2: 192.168.4.12

PC_3: 192.168.4.13

PC_4: 192.168.4.14

PC_5: 192.168.4.15

Configuring the devices:

Router Configuration

1) Right click the Router and Select "CLI"

2) Enter the following commands enable

Configure terminal

Interface Fast Ethernet 0/0

Ip address {IP address} {Subnet mask}

no Shutdown

* Fast Ethernet 0/0 connects to the Switch and

Fast Ethert connect

Switch Configuration

1) Right click the Switch and Select "CLI"
2) Enter the following commands
enable
configure terminal
Interface Fast Ethernet 0/1
Switch port mode Access
exit
Interface Fast Ethernet 0/2
Switch Mode access
exit

* This configure the Switch ports to operate in access mode for PC's

PC Configuration

1) Right click each PC and Select "Config"
2) Enter IP addr Subnet mask, default gateway and DNS with the same Subnet as the Fast Ethernet 0/1.

Gigabit Ethernet Configuration

1) Right click the Router and Select "CLI"
2) Enter the following commands.

Enable
configure terminal
Interface gigabit Ethernet 0/0
IP address {IPaddress} {Subnet Mask}

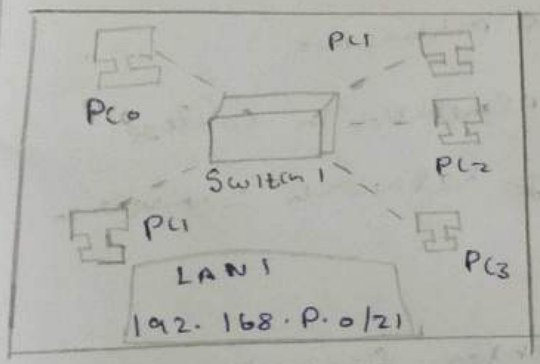
This Config
with an IP
now that
we can te
on each P
can also
o/w the

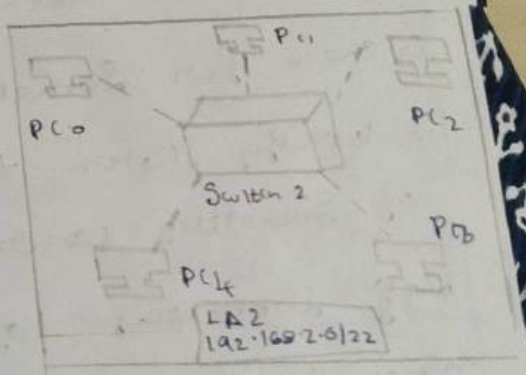
PC0
Sw
PC1
LAN
192.1

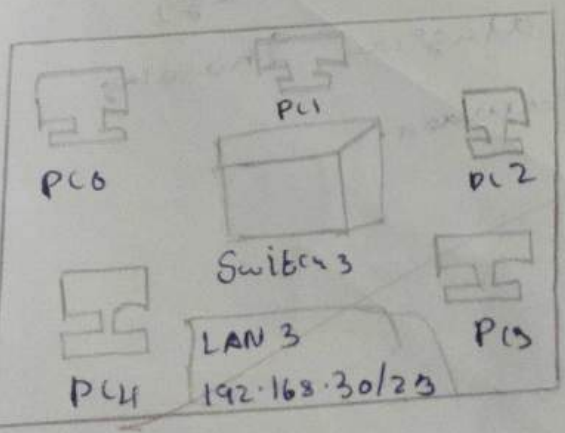
PC0
PC4

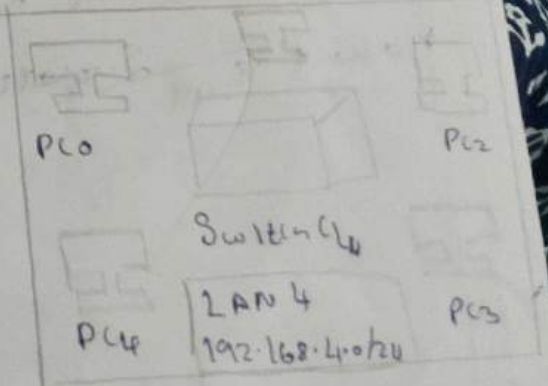
* This configures the Gigabit Ethernet interface with an IP and Subnet mask.

Now that our network topology is configured we can test the network. Open a Command Prompt on each PC and try to ping the other PC. We can also use the "ping" command to test connectivity b/w the Router and PCs









a) Understanding of Subnetting

* It is the process of dividing a large network smaller, more manageable subnetworks.
* It helps organize network traffic, efficiently and security by reducing broadcast domains

b) Advantages of implmating Subnetting

* Improved network performance
* Enhanced Security and management

c) check for Subnetting Implmetation in College

* Verify with your college network administrator of Subnetting (if used

* If Subnetting is implmenton, list the VLANs and their Corresponding IP address ranges then, draw a network diagram showing how the Subnetting are divided.

ing a large network
Subnetworks.
traffic efficiently in
broadcast domain

etting
ance
anagment

etation in college
twork campus

, list the above
ress range
ran showing
.

Result:

Thus the above program executed successfully

Practical 10A

Exc: 10A
Date: 29-10-24

Aim:
a) Internetworking with Router in CISCO PACKET TRACER Simulator

Procedure:

* Place 2 PCs and a 2811 Router as shown in our manual
* Connect PC1 with Router and PC2 Router using copper through cable
* Click on Router Go to Config. Select Fast Ethernet, write IPV4 address as "192.168.10.1" click Fin
* Go to CL I type the following codes
Router (config-if)# no Shutdown
Router # exit
* Select Fast Ethernet 0/1 write IPV4 address as "192.168.20.1" click Enter
* Go to CLI type the following codes:
Router (config-if)# no Shutdown
Router # exit
* Click on PC0. Go to Desktop -> IP configuration and the below details

IPV4 Address: 192.168.10.2
Default Gateway: 192.168.10.1

Subnet mask:
. click Enter
. Click on PC1
IPV4 Address
Default Gate
Subnet mask
. click Ente

Res

Subnet mask : 255.255.255.0

- Click Enter

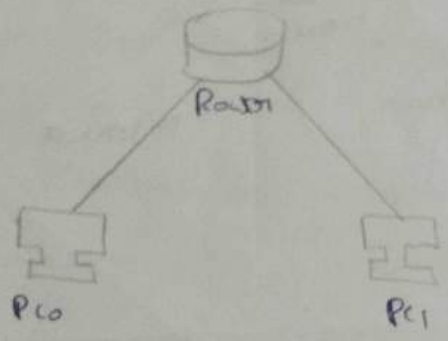
- Click on PC1 goto Desktop -> IP configuration

IPv4 Address : 192.168.20.2

Default Gateway : 192.168.10.1

Subnet mask : 255.255.255.0

- Click Enter



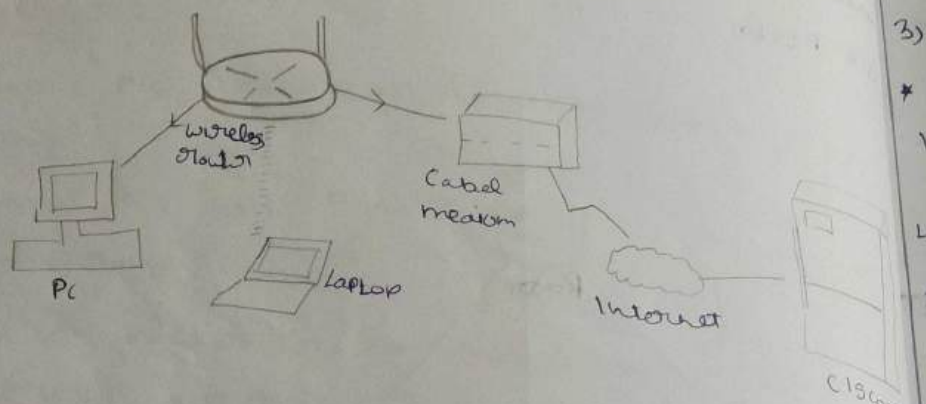
Result:

Thus the above program executed successfully

Practical-10B

Exc: 10B
Date: 30-10-24

Aim: Design and configure and internetwork using router, DHCP server and Internet cloud.



Addressing Table

Device	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		
Cisco.com Server	Ethernet0	208.67.220.220	255.255.0	
Laptop	Wireless0	DHCP		

Part1: Build a Simple Network:

1) Launch Packet Tracer

2) Add Devices
* Select device
selection boxes
* Place device

3) Rename Devic
* Click on ea
in the config

4) Connect Devi
* use copper
* PC wireless
* Wireless
* Cable m
* Internet

Gigabit Ethe

use the
Interface
then cabl

Testing

Open Co

3) Add Devices

* Select device from Device Type & Device Specific Selection boxes
* Place devices in workspace.

3) Rename Devices:

* Click on each device and change the display name in the config tab.

4) Connect Devices:

* Use Copper Straight Through Coaxial Cable Connect
* PC wireless router
* wireless router to cable modem
* Cable modem to Internet cloud
* Internet cloud to Cisco.com server

Gigabit Ethernet Configuration

use the CLI to configure the Gigabit Ethernet o/p interface with desired IP address and subnet mask then enable the interface.

Testing the Network

Open Command prompt and Ping PC.

Student observation

a) Write down your understanding subnetting

A Subnetting is process of dividing a large network smaller manageable subnetwork (subnets) Each subnet operate within a defined range of IP address

b) What is advantage of implementing subnetting a network.

A Improved network. Efficient IP address management Enhanced security. Simplified Network management

c) Find out whether subnetting is implemented in your if yes, draw and list down the subnet used

Subnet 1: 192.168.0.0/24 (192:168.0.1 - 192.168.0.254)
Subnet 2: 192.168.1.0/24 (192:168.1.1 - 192.168.254)
Subnet 3: 192.168.2.0/24 (192.168.2.1 - 192.168.2.254).

Result:

Implementing Subnet in Cisco Packet Tracer Simulator successfully executed.

Ex-No: 11a
Date 04-11-24

Aim: Simulate Static Packet Tracer

10.0.0.5

Pc0 Pc1

Lab Setup:

Connection

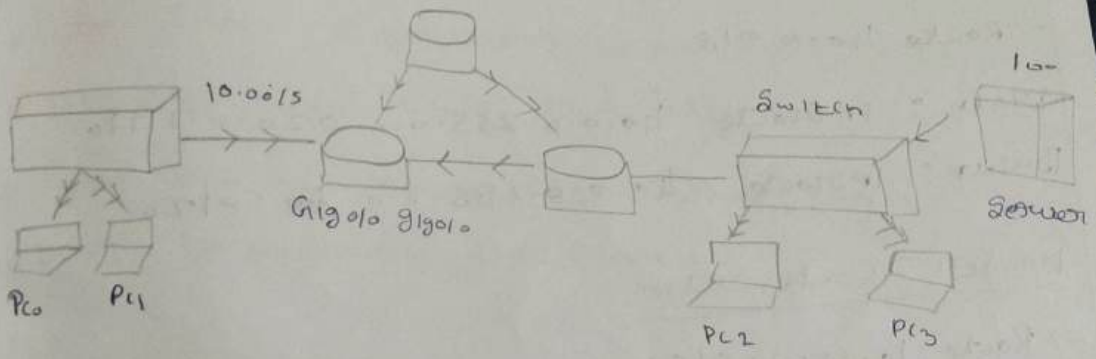
* Router 0: 10
* Router 1: 2
* Router 2: 4

Router Con

-> Router
-> Mains
-> Backup
-> Host

Ex.No: 11a
Date 04-11-24

Aim:
Simulate Static Routing Configuration using Cisco Packet Tracer



Lab Setup:

Connection Network:

* Router 0: 10.0.0.0/8, 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, 10.0.0.0/8, 40.0.0.0
* Router 2: 4.0.0.0, 50.0.0.0/8, 20.0.0.0/8, 30.0.0.0

Router Configuration

→ Router for 30.0.0.0/8

→ Main: IP Route 30.0.0.0 255.0.0.0 20.0.2.0

→ Backup: IP Route 30.0.0.0 255.0.0.0 40.0.0.0

→ Host Route 30.0.0.100/8:

*Main: IP route 30.0.0.100 255.255.255.40.0.0.22.0

*Backup: IP route 30.0.0.100 255.255.255 40.0.0.2

→ Route 10.0.0.0/8

main: IP route 10.0.0.255.0.0.0 20.0.0.11.0

Backup: IP route 10.0.0.255.0.0.0 50.0.0.12.0

→ Route 40.0.0/8

main: IP route 40.0.0 255.0.0 020.0.1 11.0

Backup: IP route 40.0.0.255.0.0 [illegible].0/20

Router 2 Configuration

→ Route 10.0.0.0/8

IP route 10.0.0.0255.0.0.0 40.0.0.1

→ Route 50.0.0.0/8

IP route 30.0.0 255.0.0.0 30.0.0.2

Verifying configuration

* Use Show IP route static to view the Routing table

* Testing routers.

↳ Ping where tracer from PCs in connected network to verify routing Path

Simulating

* To test route and network.

Deleting

* use

* Identify

* use

steps

Simulating route failure:

* To test backup routes disconnect the primary route and recheck connectivity to the target network.

Deleting static routes

* use show ip address to view all routes
* Identify the route to delete
* use no ip route [destination] [mask] [next hop] to remove the route

Result: Thus Simulation Static routing configuration using Cisco Packet tracer successfully implemented.

Practical 11B

Ex: 11b

Date: 04-11-24

Aim:

Simulate RIP using Cisco Packet Tracer

network configuration

Device and IP configuration

Pc 0: 10.0.0.2 /8

Router 0 : Fa0/1: 10.0.0.1 /8

Router 0 S0/0/1: 192.168.1.254 /30

Router 1 S0/0/0: 192.168.1.250 /30

Router 1 S0/0/1: 192.168.1.246 /30

Router 2 S0/0/1: 192.168.1.253 /30

Pc1 : 20.0.0.2 /30

Router 2 Fa0/1: 20.0.0.1 /30

Step to configure

Assign IP address to Router:

-> Access each Router in CLI

-> use configure technical to enter global config mode

Configure serial Interface

-> Set clock rate and band width for all the ends of serial connections

Implement RIP

-> Enable RIP

-> Specify network

Commands

Verification:

Ping test

Route Man

automatic

goes do

Simulate

Router

now

Result

Implement RIP Protocol

→ Enable RIP on each Routers with Router RIP
→ Specify network to advertise using network command

Verification:

Ping test: Use the ping command from PC to Router Connectivity to PC

Route Mangement: RIP dynamically manages Routers automatically switching to alternative Route if Router goes down

Simulate Link failure: Disconnect a cable b/w Router 0, Router 2, then use tracer to observe how RIP Responce traffic.

Result: Thus the Simulation RIP using Cisco Packet Tracer Implemented Successfully.

Practical 12A

Exc: 12a

Date: 04-11-24

Aim:

To Implement echo client server using TCP/IP Sockets

a) TCP, echo Client-Server algorithm

TCP server algorithm

1) Create a TCP socket
2) Bind the socket to a local address
3) Listen for incoming client connection
4) Accept a client connection
5) Loop
6) close the connection

TCP-Client Algorithm:

1. Create a TCP Socket
2) Connect to the server using specified address and port
3) Send a message to server
4) Recive the message from the server
5) Display the recived message
6) close the Socket

tcp_server.py:

```
Import socket
def TCP_Server():
    Server_Socket = Socket.Socket(Socket.AF_INET,
```

Server_Socket
server_Socket
Print("TCP
Connection.(cli
Print(f'Con
try:
while Tru
data =
if data
print(
else
break;
finally:
Connec
if -nan
tcp_
tcp_clien
Import
def tcp_
client
Clien
try:

```
server_socket.bind(("localhost", 12345))
server_socket.listen(1)
print("TCP server is waiting for a connection")
connection, client_address = server_socket.accept()
print(f"Connected to {client_address}")
try:
    while True:
        data = connection.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 sed")
client_socket . sendall (message . encode())
data = client . socket . recv (1024)
print (f"Recieved from server: {data . decode}")

Finally:
    client . socket . close ()

if __name__ == "__main__"
    tcp_client ()

Output
Client
PSC:\user\Ashan\onedrive\Deskt\CN7CaEx12-a
PSC:\user\Ashwin\onedrive\Desktod CN\Ex12a>python\to
TCP server is wait for a connection
Connected to ('127'.0.0.1' 57900
Recieved: hi Iam ashwin
PSC: users\Ashwin\onedrive\Desktop\CN\Ex12a>
Server
PSC: \Users\Ashwin\onedrive\Desktop\CN7CaEx12-a
PS C: \user\Ashwin\onedrive\desktop\CN\Ex12-a>python
Enter a message to sena: hi Iam ashwin
Recieved from server: hi Iam ashwin

Result:
The program for using echo client server using TCP has been executed success

Ex: 12B
Date: 4-11-24

Aim:
b) Implement Ch

ALORITHM:

Chat server

1) Start the se
* Create a s
* Bind it to
phone number
* Listem fo

2) Accept
* When a
· Ad
· St

3) Recieve
* For ea
* keep
* When

Practical 12 B

Exc : 12B

Date : 4-11-24

Aim :

b) Implement chat client server using TCP/IP Sockets

ALGORITHM :

Chat Server

1) Start the Server :

* Create a Socket (like a phone line).
* Bind it to a specific address and Port (Set your phone number)
* Listen for Incomming connection (wait for calls)

2) ~~Alle~~ Accept Connection :

* When a new client connects :
  - Add the client to list of connected clients
  - Start a new process to talk to this client

3. Recive Messages :

* For each connected client
* Keep checking for new message
* When a message is recived, show it on the Server Screen

4. Handle Disconnections:

* If a client disconnects
  * Remove that client from the list
  * Stop talking to that client

5. Keep Running

* Repeat the process until you stop the server

Chat client

1) Connect to the server:

* Create a socket
* Connect it to the server's address and port

2) Recieve Messages:

* Start a process to listen for messages fr... server
  * Whenever a message arrives, show it on ... screen.

3) Send Messages:

* Keep asking the user for new messages.
  * If the user types a message, send it to ser...
  * If the user types 'exit' disconnect from t... server and close the client.

4. Keep Running:

* Repeat the process until the user desires to ...

import socket
import threa...
def recieve...
while true:
try:
message =
decode (utf-...
if mes...
print (f...
except
print...
brea...
def start...
client...
soc...
host...
port...
clien...
print...
threadin...
clo...
while T...
me...
Clie...
if -h...

```
import socket
import threading
def recieve_message (client_socket)
while true:
  try:
  message = client_socket.reco(1024).
decode (UTF-8)
    if message:
  print (f"server : {message}");
  except Exception as e:
   print (f" an error occured : {e}")
   break.
def start client ():
  client_socket = socket.Socket (socket.AFINET, Socket
   SOCK_STREAM)
  host = '127.0.0.1'
  Port = 12345
  client = socket.connect (host, Port))
 print ("connected to chat server")
threading.thread (target.recieve : message, arg = (client
 daemon = true).start ()
while True:
   message = input ("yes")
  client_socket.send (message.encode ('UTF-8')
if -name- = "-main--"
        start_client ()
```

Chat Server.py :-

```
Import Socket
Import threading
def handle_client (client_socket):
while True:
 try:
message = client_socket.recv(1024).
decode ('utf-8')
  if not message:
    break
Print(f"Recived message from client {mess
client_socket.close()
 except Exception are:
 Print ff" An error has occured")
   break
 client.socket.close()
def start_server ():
   server = socket.socket (socket. AFINET,
Socket_Socket.Sock_STREM)
   server.bind (('127.0.0.1", 12345))
   server.listen (5)
Print ("Chart server has started on
127.0.0.1:12345")
 While true:
client_socket addr = server.accept()
Print (f"New Connection from
{addry")
```

Client_na
thread (tar
Client
if -- name
Sta

Output

PS C:\Users
Chat Ser
NewCome
Recived m
Type yo

PS C: \U
PS C: \U
Connect
You:
You:

Result:
T
Server
Execu

```
client_handler = threading
Thread (target: handle args= (client_socket)
    client_handler: start()
if __name__ = "__main__":
    start_server().
```

Output

```
PS C:\Users\Ashwin\OneDrive\Desktop\CN\Ex\2b\Py>
Chat Server Started on 127.0.0.1:12345
New connection from ('127.0.0.1' 58000)
Received message from client: Hello I am Ashwin
Type your message to client: Clear Hi

PS C:\User\Ashwin\OneDrive\Desktop\LN
PS C:\Users\Ashwin\OneDrive\CN\Ex\2-b
Connected to the Chat Server
You: Hello I am Ashwin
Your: Server: Clear Hi
```

Result:

Thus the Implementation of Chat client Server using TCP/UDP Socket has been Successfully Executed.

Practical-13

Ex: 13

Date: 04-11-24

Aim:

Implement your own ping program.

Algorithm:

Ping_client.py

1) Socket Creation

2) Then set a timeout of 2 second to ensure that if no responses is recieved it will stop waiting and recieve prime request time.

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

4) It listens for a response and calculate time difference between the sending and recieving the packet.

Ping server.py:-

1) Initlize the UDP socket

2) Bind to IP address and port

3) Listen for incoming messages

4) Recieves data

5) Send response

Code:

Ping_server.py:-

import sock
def start
s.bind (
print (f'
{host}'
while True
data, a
print (f
{addr}:
s.sendto
if_name
sta
Ping_cli
import
import
def ping
with s
try:
s.set
start
s.sen
data

```
import socket
def start_server(host='127.0.0.1', port=12345):
    s.bind((host, port))
    print(f"UDP server running on
    {host}:{port}")
while True
    data, addr = s.recv from (1024)
    print(f'Recived 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='127.0.0.0' port=12345):
with socket.socket(socket.AFINET, Socket.SOCKDG
try:
s.settim out (2)
start = time.time()
s.sendto(b'ping', (host, port))
data, addr = s.recvfrom(1024)
```

```
        end = time.time()
        print(f"Received {data.decode()} from {addr}")
        print(f"Round-trip time: {end - start:.2f} seconds")
    except socket.timeout:
        print("Request timed out")

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

OUTPUT:

```
>Python ping_server.py
UDP server running on 127.0.0.1 12345
Received message from ('127.0.0.1', 57134):
>Python ping_client.py
Received Ping from ('127.0.0.1', 12345) in 0.
```

Result: The Program to create and implement Ping message has been successfully executed.

Ex: 14
Date: 04-11-24

Aim: write

Packet Sn

Algorithm

1) Install

2) Create a file call to

3) Set Packet proto

4) Capt

5) Run

6) Cr the

Practical 14

Exc: 14
Date: 04-11-24

Aim:
Write a code using RAW Sockets to implement Packet Sniffing.

Algorithm:

1) Install Python and Scapy
2) Create a Program Open text editor create a file in notepad called packt Sniffing.py and code to capture and analysis the network packets
3) Set up Packet tracer by check of the Packet has IP layers identify the packet overhead 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:
        ip_layer = packet[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst
    protocol_name = ""
    if protocol == 1:
        protocol_name = "ICMP"
    elif protocol == 6:
        protocol_name = "TCP"
    elif protocol == 17:
        protocol_name = "UDP"
    else:
        protocol_name = "Unknown Protocol"
    print(f"Protocol: {protocol_name}")
    print(f"Source IP {src_ip}")
    print(f"Destination IP: {dst_ip}")
    print("-"*50)

def main():
    sniff(prn=packet_callback, filter="IP",
    store=0)
if __name__=="__main__":
    main()
```

Output :-

```
>Python packet_
  Protocol: TCP
Source IP : 232
destination IP: 29
```

Result :-

The

is done

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

Output:-

```
>Python Packet_Sniffer.py
  Protocol: TCP
Source IP : 232.15.215.227
destination IP: 292.163.109.73
```

Result:

The Program to implement Packet Sniffing is done successfully.

Practical-15

Exc: 15

Date: 04-11-24

Aim:

To analyzes the different types of Web log using Weblizer Pool.

Procedure:

Step1: Run weblizer window version

Step2: Input Web file (download from Web)

Step3: Press Run weblizer.

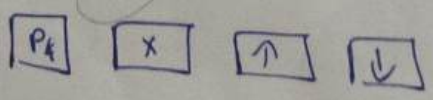
Input:

Choose Logfiles View Setting Additional HTML(log, hits/graph | ignore | include graph Config file

Input:

Log files:

① C:\User\Tess\Download\Access.log



Target directory:

C:\User\Tess

☐ Clear Existing directory

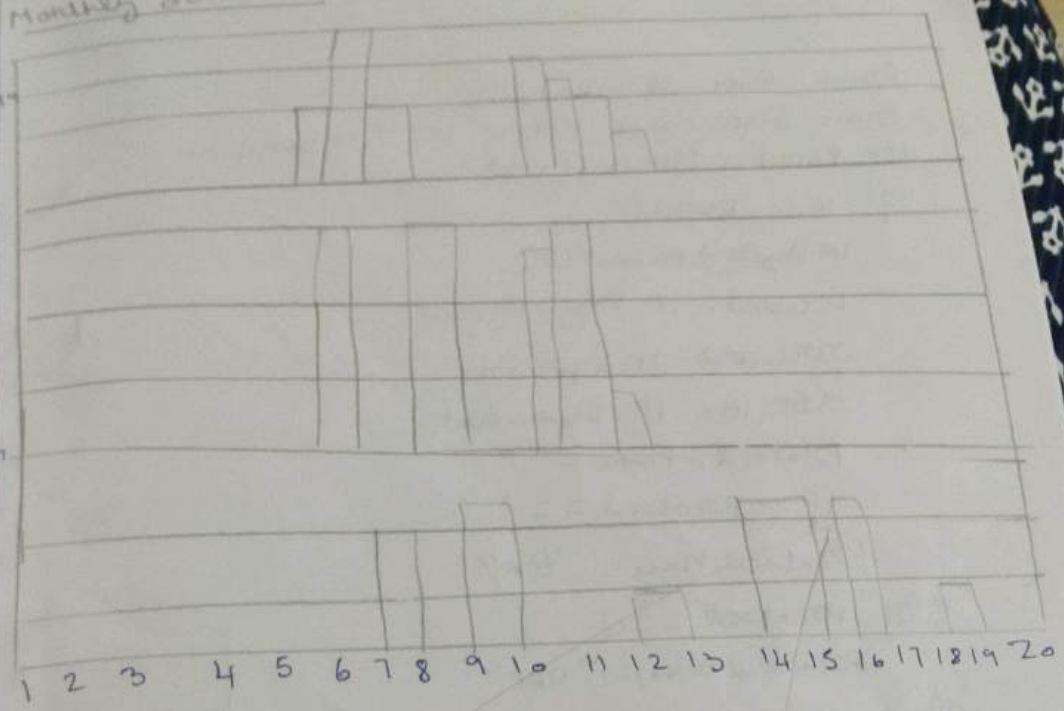
Monthly statistics

1 2 3

1.	100
2.	72
3.	47
4.	44
5	3
6.	2

Results

Monthly Statistics



IP Address

1.	100	6.47.1.	83	7.47.1.	62	Snowtrib.ca
2.	72	4.66.1.	11	5.66.1.	51	Search.cat
3.	47	3.04.1.	27	3.04.1.	52	outcome.ca
4.	44	2.9.1.	81	3.95.1.	91	heavent.ca
5	35	2.5.1.	63	2.95.1.	51	oc.ca
6.	29	3.95.1.	41	4.95.1.	64	nohaoct.ca

Result: Three different types of webpage one successfully analyzed with weblizer tool and output verified.

Program 8:

```
From Scapy.all import Sniff
From Scapy.inet import IP, TCP, UDP, ICMP
def Packet_Callback(packet)
if IP in Packet:
    IP layer = Packet[IP]
    Protocol = IP_layer.Protocol
    Src_IP = IP_layer.Src
    dst_IP = IP_layer.dst
    Protocol_name = ""
    if protocol == ""
    Protocol name = "tcp"
elif protocol == 6
    Protocol name = "UDP"
else
    Protocol_name = "UnknownProtocol"
Print("f" Protocol: {Protocol_name}")
Print("f" Source IP: {Src_IP}")
Print(f" Destination IP: {dst_IP}")
Print("_" *50)
def main()
    .Sniff(prn = Packet_Callback, filter="ip", sto
if __name__ == "__main__":
    main()
```

Output:

Thus the Program executed successfully.

, OOP, ICMP

completed

Ruth.

ter OOPs, Stor