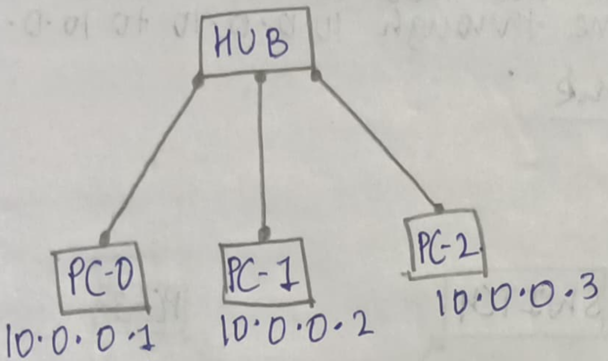
16/06/2023 Lab-01

Create a topology consisting of three or more devices connected with the help of a hub, switch and combined.

Aim: To understand the working of Hub, switch and hybrid network.

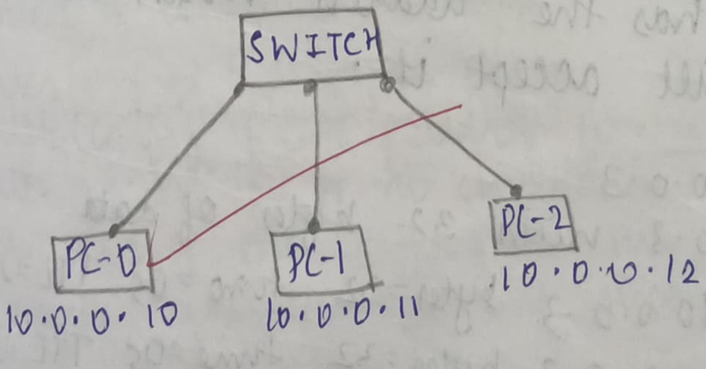
(i) Using Hub



Observation:

Whenever a source node sends data in the network, the hub recieves the data from the source and brodcasts it over the network i.e it sends the message to all the remaining nodes in the network and the node whose destination address matches with the message send recieves & accepts that data. The communication was carried between 10·0·0·1 → 10·0·0·3.

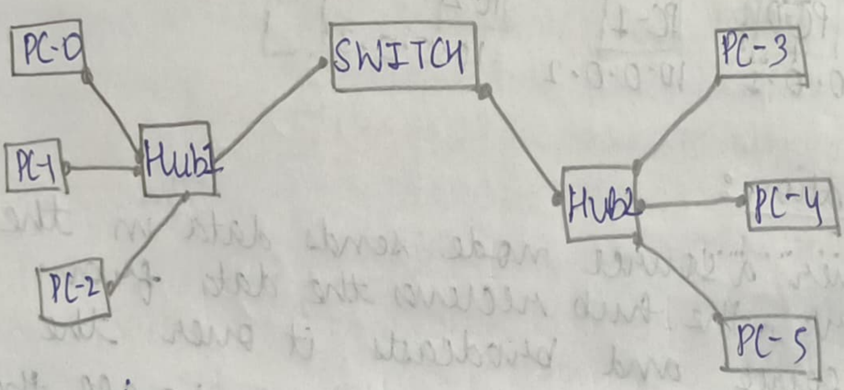
(ii) Using Switch



Observation:

Whenever a source node sends the data to other node, the switch first recieves the data. In the first time the switch sends it to all the other devices but when sent again it sends it to only the destination device which has the address match. The communication is done through 10.0.0.10 to 10.0.0.13

iii Hybrid network



Observation:

The PC-0 sends the message to the Hub-1 which recieves it and then broadcasts to PC-1, PC-2 and the switch. The switch then sends it to Hub-2 which recieves it and broadcasts it to all the devices PC-3, PC-4, PC-5. The PC-4 which is the destination device and has the address match with the message will accept it.

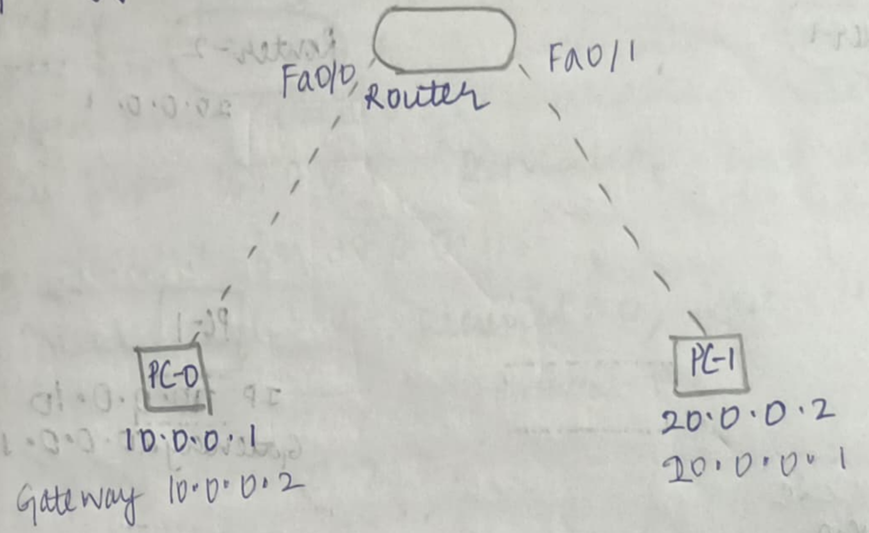
PC > ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data
Reply from 10.0.0.3 bytes=32 time=0s TTL=255
Reply from 10.0.0.3 bytes=32 time=0s TTL=255
Reply from 10.0.0.3 bytes=32 time=0s TTL=255
Reply from 10.0.0.3 bytes=32 time=0s TTL=255
Packets: Sent: 4 Recieved 4, Loss = 0

23/06/2023 Lab-02

Q Create a topology consisting of 2 devices connected with the help of a router

Aim : To check the functionality of a Router

Topology-1



Observation

The router Router-PT is connected to two end devices PC-0 & PC-1. First we configure the IP address of PC-0 & PC-1 as 10.0.0.1 and 20.0.0.2 since they belong to different network.

Then we configured the Router for each interface of PC-0 & PC-1 by writing a set of commands in the CLI window of the router

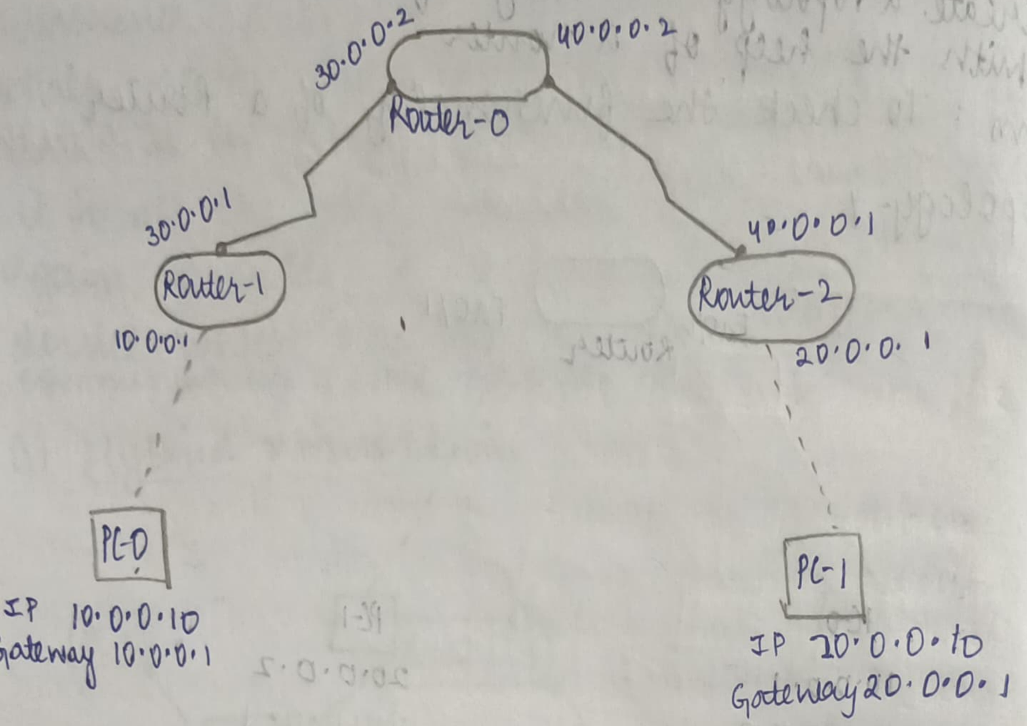
Router>enable
Router # configure terminal
Router(config) # interface Fa0/0
Router(config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown

After setting the interface we ping but it shows request timed out since we have not set the gateway.

After setting the gateway for each device as 10.0.0.2 and 20.0.0.1 respectively, the connection is made.

The 10.0.0.2 will be the gateway for PC-0

Topology -2



Observation

In this topology Router-1 is connected to PC-0 and Router 2 is connected to PC-1. Router-0 is connected to Router-1 & Router-2 afterwards. For PC-0 IP address is set as 10·0·0·10 and the gateway for the connection to Router-1 is set as 10·0·0·1

For PC-2, IP address is set as 20·0·0·10 and the gateway for the connection is set as 20·0·0·1
Then for the Router-0 we set the gateway for Router-1 & Router-2 connection by the commands in the CLI window and setting as 30·0·0·2 & 40·0·0·1 respectively

Ping Requests

PC > ping 20.0.0.10.

Pinging from 20.0.0.1: Destination with 32 bytes of data

Reply from 10.0.0.1 : Destination host unreachable
Reply from 10.0.0.1 : Destination host unreachable
Request timed out 10.0.0.1 : Destination host unreachable
Reply from 10.0.0.1 Destination host unreachable
Ping statistics for 20.0.0.10.
Packets: sent = 4 , Received = 0, Lost = 4 (100% Loss)

PC > ping 30.0.0.2 :

Pinging 30.0.0.2 with 32 bytes of data:
Request timed out
Request timed out
Request timed out
Request timed out

PC > ping 30.0.0.1.
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Packets: Send = 4 , Received = 4, Lost 0 (0% Loss)

30/6/23

9

30/6/2023

Lab-02 (Continued)

Aim: To depict static connection between 2 routers in a topology of routers

Procedure:

Go to CLI in Router configuration and enter the following commands "show ip route".
This command is used to display the IP routing table of a router. The router provides additional route information, including how the route was learned, how long the route has been in the table and which specific interface to use to get to a predefined destination

Setting of ip routes:

Router-1

Router > enable
Router # configure terminal
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router (config) # exit

Observation:

Router # show ip route

C 10.0.0.0 /8 is directly connected, FastEthernet 0/0
S 20.0.0.0 /8 [1/0] via 30.0.0.2
C 30.0.0.0 /8 is directly connected, serial 2/0.
S 40.0.0.0 /8 [1/0] via 30.0.0.2

Router-0

Router > enable
Router # configure terminal
Router (config) # ip route 20.0.0.0 255.0.0.0 40.0.0.1
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router (config) # exit

Observation

Router # show ip route

S 10·0·0·0/8 [1/0] via 30·0·0·1
S 20·0·0·0/8 [1/0] via 40·0·0·1
C 30·0·0·0/8 is directly connected, Serial 2/0
C 40·0·0·0/8 is directly connected, serial 3/0

Router-2

Router >enable
Router # configure terminal
Router (config) # ip route 10·0·0·0 255·0·0·0 40·0·0·2
Router (config) # ip route 30·0·0·0 255·0·0·0 40·0·0·2
Router (config) # exit

Observation:
S 10·0·0·0/8 [1/0] via 40·0·0·2
C 20·0·0·0/8 is directly connected, Fast Ethernet 0/0
C 40·0·0·0/8 is directly connected, Serial 3/0
S 30·0·0·0/8 [1/0] via 40·0·0·2

Ping commands

Ping 20·0·0·10
Pinging 20·0·0·10 with 32 bytes of data.
Reply from 20·0·0·10 bytes=32 time=2 ms TTL=125
Reply from 20·0·0·10 bytes=32 time=9 ms TTL=125
Reply from 20·0·0·10 bytes=32 time=6 ms TTL=125
Reply from 20·0·0·10 bytes=32 time=11 ms TTL=125

Ping statistics for 20·0·0·10:
Packets sent = 4, Received = 4, Lost = 0 (0% Loss)

Ping 30·0·0·1
Pinging 30·0·0·1 with 32 bytes of data.
Reply from 30·0·0·1 bytes = 32 time = 0 TTL = 225
Reply from 30·0·0·1 bytes = 32 time = 0 TTL = 225
Reply from 30·0·0·1 bytes = 32 time = 0 TTL = 225
Reply from 30·0·0·1 bytes = 32 time = 2 TTL = 225
Packets sent = 4, Received = 4 Lost = 0 (0% Loss

Ping 10·0·0·10
Pinging 10·0·0·10 with 32 bytes of data
Reply from 10·0·0·10 bytes =32 time = TTL = 255
Reply from 10·0·0·10 bytes = 32 time = TTL = 255
Reply from 10·0·0·10 bytes = 32 time = TTL = 255
Reply from 10·0·0·10 bytes = 32 time = TTL = 255

Packets sent = 4 Received = 4 Lost = 0

Outcome

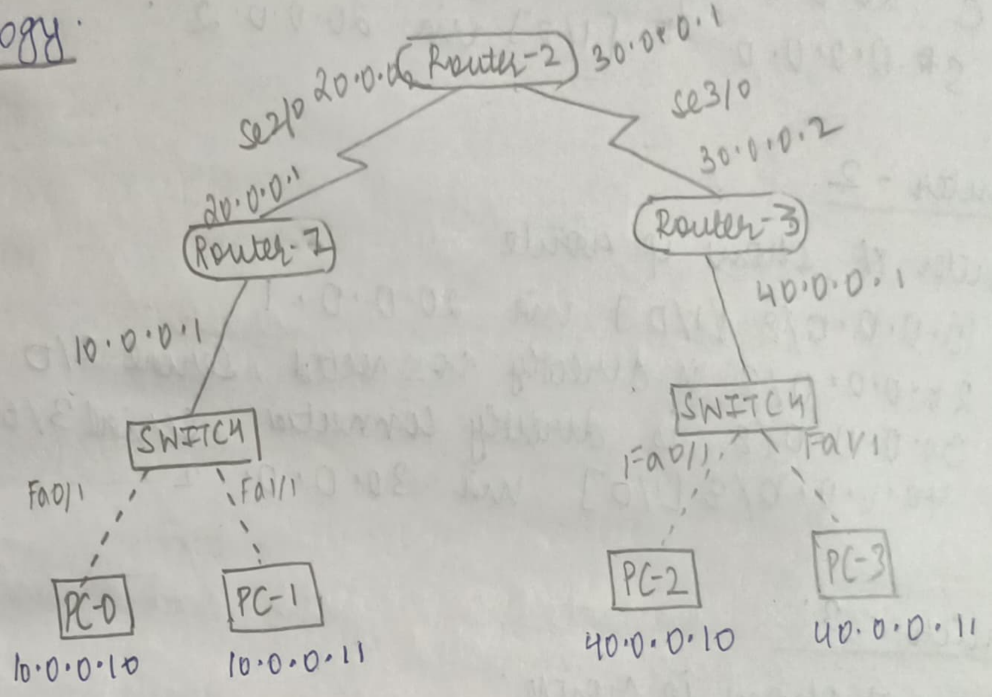
To understand how we can put static route to different routers in a connection of multiple routers.

30/6/2023

Lab-03

<u>Aim</u>: To create a topology of routers in order to depict default and static connection

<u>Topology</u>:



<u>Procedure</u>:

Using the "ip route" command configure Default route to Router 1 & Router 3.

<u>CLI commands</u>:

<u>Router-1</u>.

Router (config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

<u>Router-2</u>

Router (config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

<u>Router-3</u>

Router(Config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

Observation :

Router-1

Router # show ip route

C 10.0.0.0 /8 is directly connected , Fast Ethernet 0/0

C 20.0.0.0/8 is directly connected , Serial 2/0

S* 0.0.0.0 /0 [1/0] via 20.0.0.2 .

Router - 2

Router # show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected , Serial 2/0

C 30.0.0.0/8 is directly connected , Serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0. 2

Router - 3 :

Router # show ip route

C 30.0.0.0/8 is directly connected , Serial 2/0

C 40.0.0.0 /8 is directly connected, Fast Ethernet

S* 0.0.0.0 /0 [1/0] via 30.0.0.1

Ping Commands .

pc > ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data :

Reply from 40.0.0.10 bytes = 32 time = 9 ms TTL = 125

Reply from 40.0.0.10 bytes = 32 time = 2 ms TTL = 125

Reply from 40.0.0.10 bytes = 32 time = 3 ms TTL = 125

Reply from 40.0.0.10 bytes = 32 time = 17 m TTL 125

Packets sent = 4 Received = 4 Lost = 0

Outcome:

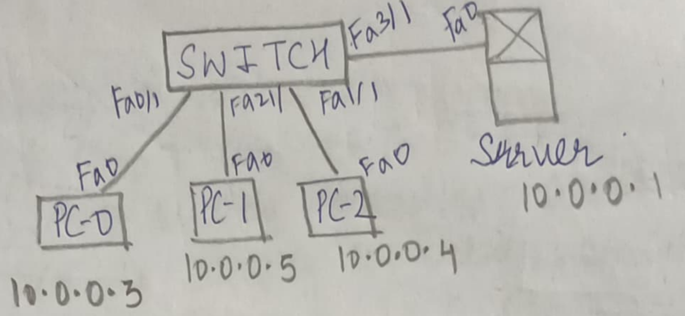
TO learn about how we can configure the ip route of routers to default and static respectively.

14/7/2023

Lab-04

Aim: To configure DHCP within a lan, outside a LAN and three networks

Topology-1



Procedure:

1.) Three PC are connected to a switch and a general server is also connected.

2.) The server ip address is set to 10.0.0.1

3.) In the server services DHCP is switched on, now it serves as DHCP server.

4.) For the pool: server pool, starting ip address is set to 10.0.0.2.

5.) Now, for each ~~server~~ PC; set it as DHCP.

6.) All PC's are given IP Address Dynamically.

Observation:

IP address for all PC's is set automatically.

Result:

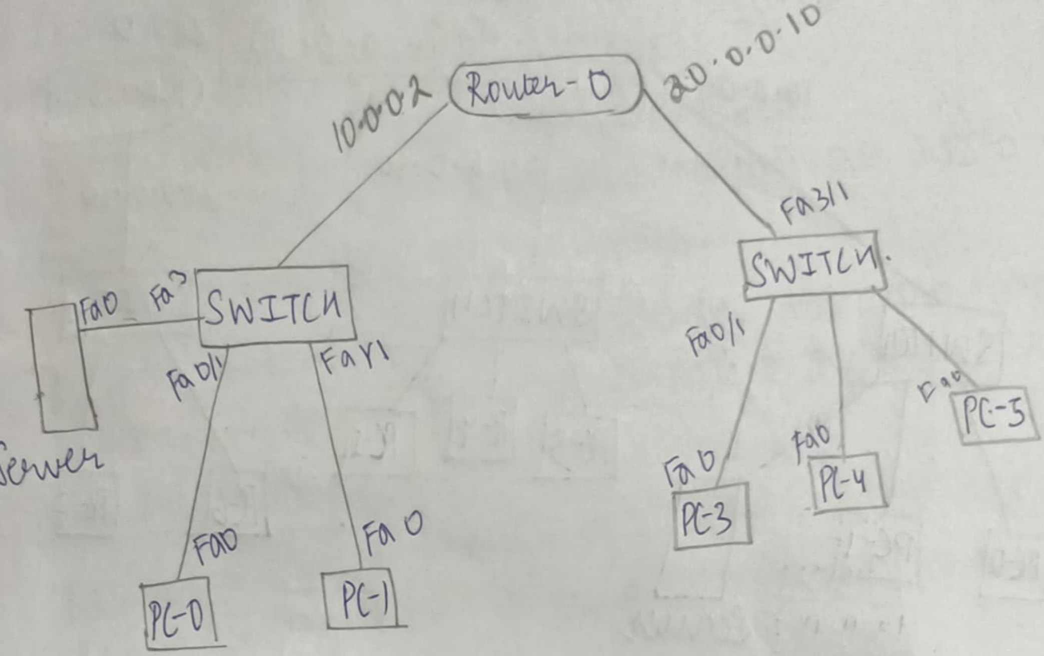
IP address of

PC-0 → 10.0.0.3

PC-1 → 10.0.0.5

PC-2 → 10.0.0.4

Topology - 2



Procedure:

1.) The server ip address is set to 10.0.0.1
2.) Another server pool: serverpool 1 and give gateway as 10.0.0.2.
3.) To the router set ip address for Fa0/1 & Fa3/1 as 10.0.0.2 and 20.0.0.10.
4.) Add the ip helper address command:
#(config)interface Fa 3/1
(config) ip helper - address 10.0.0.1

5. For server pool
Gateway: 10.0."0.2.
Start ip addr: 20.0.0.2

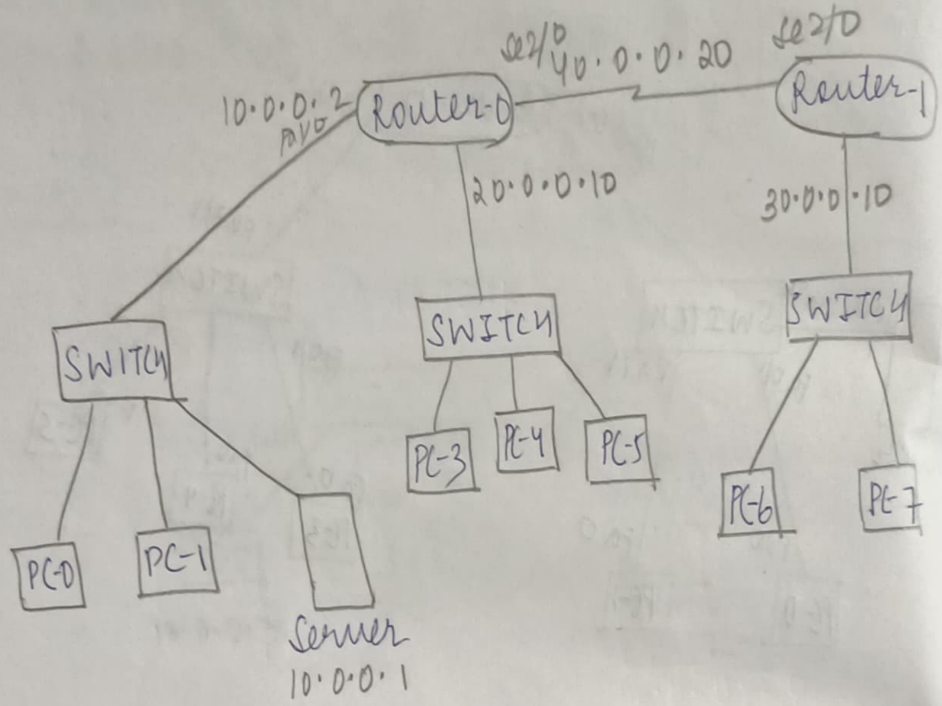
Observation

All PC's in LAN-2 will get ip address automati-caley.

Result:

PC-3 : 20.0.0.2
PC-4 : 20.0.0.4
PC-5 : 20.0.0.3

Topology - 3



Procedure:

1.) To the previous topology add another network with the Router-1, switch and 2PC's (PC-6 & PC-7)

2.) Add another server pool : Server pool 2 with gateway 10.0.0.2 and starting ip address 30.0.0.1

3.) For all PC in network 30.-.-.. make it as DHCP.

4.) For the Router-0 and Router-1, set the interface ip address, and ip route

5.) Give ip-helper address respectively for Router-0 and Router-1.

6.) All PC's in network 3 are assigned ip address dynamically.

CLI commands

Router-1

Router(config)# interface Fa0/0
Router(config-if)# ip address 30.0.0.10 255.0.0.0
Router(config-if)# no shutdown
Router(config)# interface se2/0
Router(config-if)# ip address 40.0.0.20 255.0.0.0
Router(config-if)# no shutdown
Router(config)# ip route 10.0.0.0 255.0.0.0 40.0.0.10
Router(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.10
Router(config)# exit
Router(config)# interface ~~30.0.~~ Fa0/0
Router(config-if)# ip helper-address 10.0.0.1
Router(config)# exit

Observation:

All PC's in network-3 i.e PC-6 & PC-7 are configured dynamically.

Result:

PC-6: 30.0.0.2.
PC-7: 30.0.0.3

Outcome

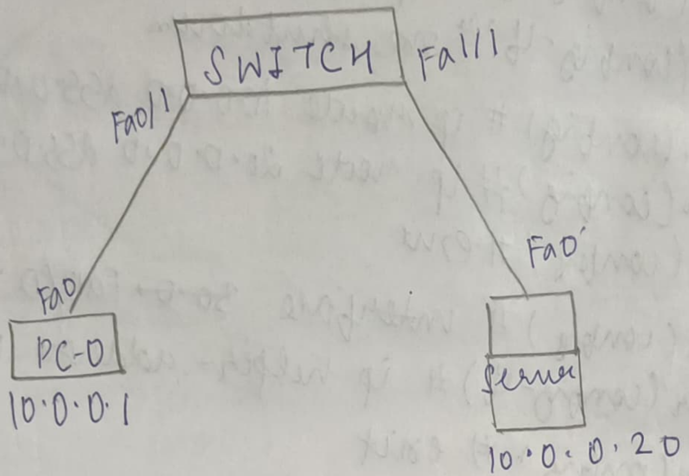
The DHCP helps manage allocation of ip address to end users. The device wanting to access a network gets an ip address allocated dynamically to it by the server in the neighbour network.

21/7/2023

Lab-05

Aim:
Configure web server, DNS within a LAN.

Topology:



Procedure:

1.) Create a topology as shown using a PC, Server, and a switch.

2.) Set the ip address as 10.0.0.1 and 10.0.0.2 for PC and server respectively.

3.) In the server, under DNS service create new rushil.com website with URL 10.0.0.2 and add under HTTP, modify the index.html file and add name and usn as:
<h1><center> RUSHIL BINDROO</center><h1>
<h1> USN: 1BM21CS172 </h1>

4.) In PC-0 go to desktop → web browser and type rushil.com. You'll be able to see the website with entered name & USN.

Result:

Web Browser

URL: http://rushil.com.

Cisco Packet Tracer

RUSHIL BINDRDO

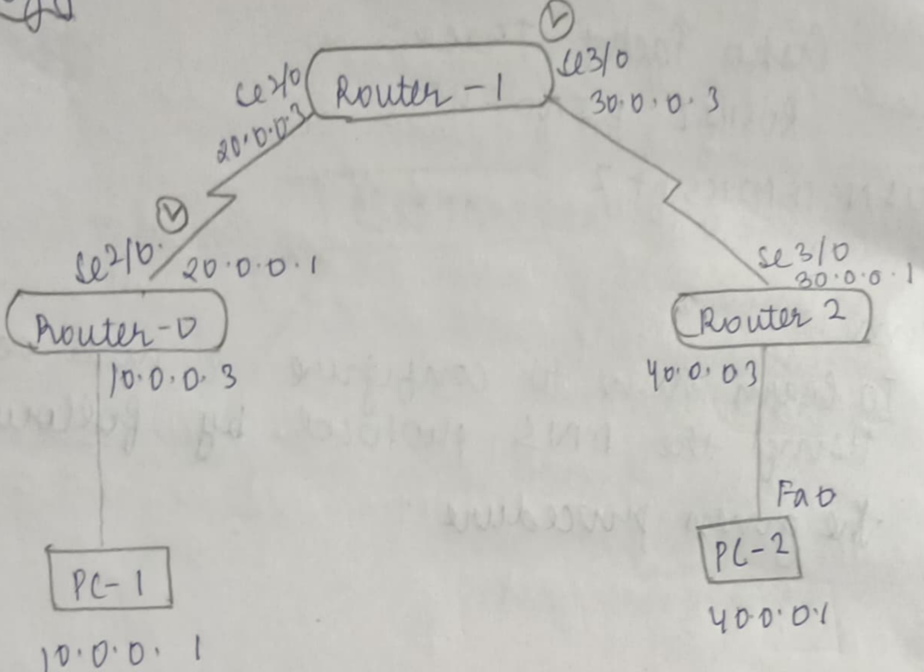
USN: 1BM21CS172

Outcome:

To learn how to configure a web server using the DNS protocol by following the given procedure.

Aim: To configure RIP routing protocol in Routers.

Topology:



Procedure:

1.) Create a topology as shown above using 2PC's and 3 routers

2.) Configure the ip addresses of 2PC's as 10.0.0.1 and 40.0.0.1 for PC-1 and PC-2 respectively. Set the gateways as 10.0.0.3 and 40.0.0.3

3.) Configure the routers (Router-0)

Router > enable

Router # config terminal

Router (config) # interface Fa0/0

Router (config-if) # ip address 10.0.0.2 255.0.0.0

Router (config-if) # no shutdown.

Router (config) # ip interface Serial 2/0

Router (config-if) # ip address 20.0.0.1 255.0.0.0

Router (config-if) # no shutdown

Same - command for Router-1 & Router-2

→ For router 0,

Router (config)# interface Se 2/0
Router (config-if)# encapsulation PPP
Router (config-if)# no shutdown
Router (config-if)# exit

Repeat this for Router 1 interfaces Se 2/0 & Se 3/0
for router 2 Se 3/0

→ For router 0 (Se 2/0) and router 1 (Se 3/0)
Router (config)# interface Se 2/0
Router (config-if)# clock rate 64000
Router (config-if)# no shutdown
Router (config-if)# exit

For all the 3 routers, repeat this step:

> Router rip
> # network < network-1 ID>
> network < network 2 - ID>
> exit

Ping command

PC> ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1 bytes = 32 time = 9 ms TTL=125
Reply from 40.0.0.1 bytes = 32 time = 2 ms TTL=125
Reply from 40.0.0.1 bytes = 32 time = 3 ms TTL=125
Reply from 40.0.0.1 bytes = 32 time = 12 ms TTL=125

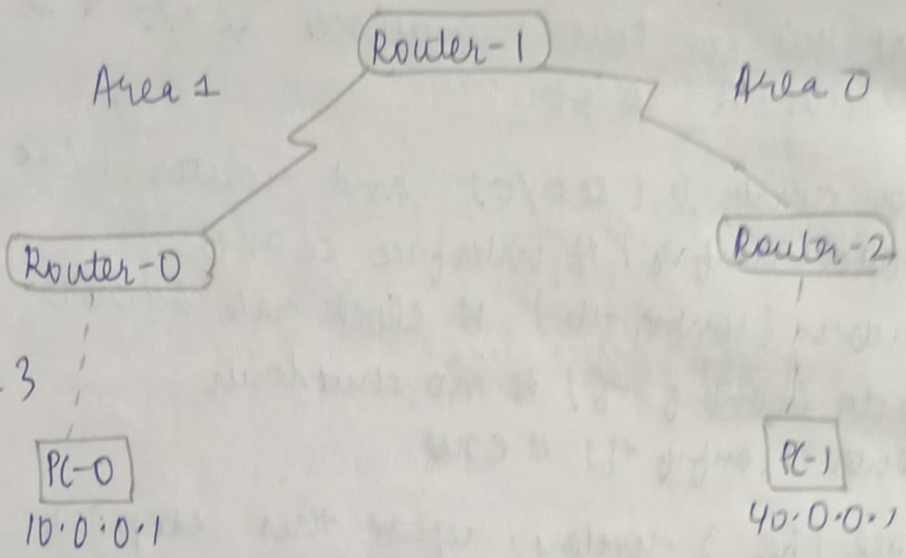
Outcome: To learn to configure routers using RIP protocol using the commands learned and that RIP is an active routing protocol.

04/08/2023

Lab-06

Aim: To demonstrate OSPF protocol in routers

Topology:



Procedure:

1.) Create the above topology with 3 routers and 2 PC's and set the ip address of the 2PC's

2.) Now provide the ip address of the router interfaces as well as the "encapsulation ppp" command at the router - router interface.

3.) For Router-0 & Router-1 in network 20930 provide clock rate 64000.

4.) For each router enable ip routing by configuring OSPF in all routers and provide the router-id.

5.) Create dedicated loopback interface for all the routers.

6.) Create a virtual link to connect area 0 to area 3.

CLI commands

1) Configuring ip address to all router - interfaces

Router-1

Router > enable.
Router # configure terminal.
Router (config) # interface se 2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # encapsulation PPP.
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # interface se 3/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # encapsulation PPP.
Router (config-if) clock rate 64000
Router (config-if) # no shutdown.

2.) Enabling ip routing by configuring ospf in all routers.

Router-0

Router (config) # router ospf 1.
Router (config-router) # router-id 1.1.1.1
Router (config-router) # network 10.0.0.0 0.255.255.255 area3
Router (config-router) # network 20.0.0.0 0.255.255.255 area1
Router (config-router) # exit.

Router-1

Router (config) # router ospf 1
Router (config-router) # router-id 2.2.2.2
Router (config-router) # network 20.0.0.0 0.255.255.255 area1
Router (config-router) # network 30.0.0.0 0.255.255.255 area 0
Router (config-router) # exit

Router-2

Router(config)# router ospf 1
Router(config-router)# router-id 3.3.3.3
Router(config-router)# network 30.0.0.0 0.255.255.255 area 0
Router(config-router)# network 40.0.0.0 0.255.255.255 area 2
Router(config-router)# exit

3.) Creating dedicated loopback interface for all routers.

Router-0

Router(config)# interface se2/0
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.252 255.255.0.0
Router(config-if)# no shutdown.

Router-1

Router(config)# interface se2/0
Router(config-if)# interface loop back 0
Router(config-if)# ip add 172.16.1.253 255.255.0.0
Router(config-if)# no shutdown

Router-2

Router(config)# interface se2/0
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.254 255.255.0.0
Router(config-if)# no shutdown.

4.) Create virtual link between Router-0 & Router-2.

Router-0
Router(config)# router ospf 1

Router(config-router)# area 1 virtual-link 2.2.2.2
Router(config-router)# exit

Router-1

Router(config)# router ospf 1
Router(config-router)# area 1 virtual-link 1.1.1.1
Router(config-router)# exit

Ping Command

PC > ping 40.0.0.10
Packets: sent = 4, Received = 0, lost = 4.

PC > ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10 bytes = 32 time = 9 ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 6 ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 8 ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 7 ms TTL=125
Reply from 40.0.0.10 bytes

Ping statistics for 40.0.0.10:
Packets: sent = 4, Received = 4, lost = 0.

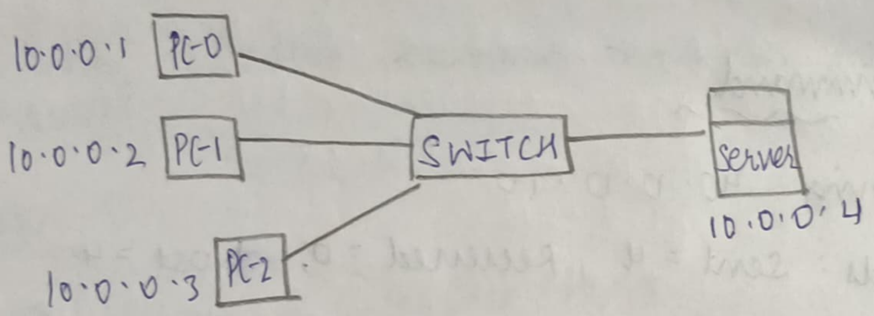
Outcome:
OSPF stands for open shortest path first is a link layer / application layer protocol which is used to find the best route for packet transfer and involves creating a virtual link.

11/08/2023

Lab-07

Aim: To demonstrate ARP protocol in routers.

Topology:



Procedure:

1.) Create the topology as shown above with 3 PC's, 1 server and 1 switch.

2.) Assign ip address to all PC's and server in the network.

3.) Go to simulation mode and open the ARP tables for all 3 PC's and the server.

4.) Now send PDU from PC-0 to server via the switch and note the incoming and outgoing PDU details at the PC, switch and the server.

5. When the PDU reaches the server the ARP table of server is updated with the "mac" address of PC-0.

6. Now send the PDU back from server to the PC-0.

7. When the PDU reaches the PC-0 the ARP table of PC-0 is updated with the

MAC address of the server

8.) Similarly send PDU from all the PC's to the server and back in order to update the ARP table.

9.) Use the command " arp - a " in order to view the 'arp table'.

CLI command & Observation

10.0.0.1

PC > arp - a

Internet Address	Physical Address	Type
10.0.0.4	0001.0409.0936	Dynamic

10.0.0.4

PC > arp - a

Internet Address	Physical Address	Type
10.0.0.1	00e0.b04b.2ac8	Dynamic
10.0.0.2	0090.217e.17d6	Dynamic
10.0.0.3	0060.3e17.5ed6	Dynamic

Outcome:

To learn about how the Address Resolution Protocol maps the IP address to the MAC address.

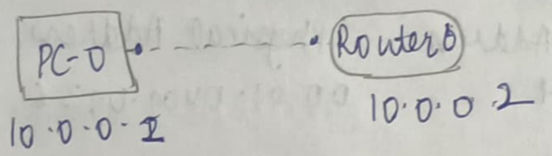
18/08/2023

Lab-08

A) Telnet

Aim: Create a topology in order to depict Telnet

Topology:



Procedure

1.) Create the topology as shown above with 1PC and 1 router.

2.) In the CLI of the router process the following commands.

CLI Commands

Router > enable

Router # config t

Router (config) # hostname R_1

R_1 (config) # enable secret P1

R_1 (config) # interface Fa 0/0

R1 (config-if) # ip address 10.0.0.1 255.0.0.0

R_1 (config-if) # no shut

R1(config-if) # line vty 0 5
R1(config-line) # login
% login disabled on line 132, unless
'password' is set.
R1(config-line) # password PO
R1(config-line) # exit
R1(config) # exit

R1 # wr

Observation

PC> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32
Ping statistics for 10.0.0.1
Packets: sent=4, Received=4, lost 0

PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access Verification
Password:
R1> enable
Password:
R1 # show ip route
Codes: C
D
N,
E,

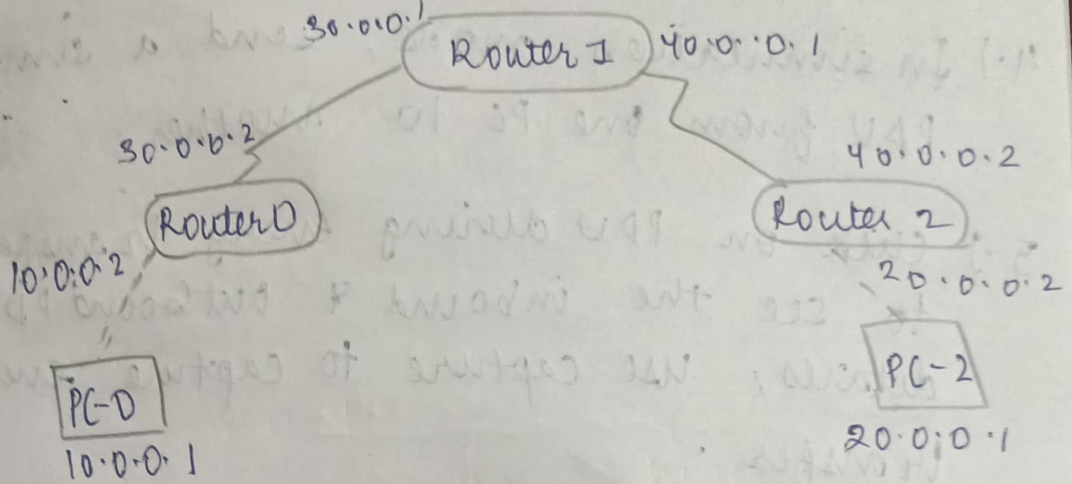
Outcome:

To understand how TELNET works in a LAN and how password is set and access is done.

B) TTL

Aim: To demonstrate TTL / life of a packet.

Topology:



Procedure:

1.) Create a topology as shown above with 2PC's and 3 routers.

2.) Configure the IP address as 10.0.0.1 and 20.0.0.1 for PC0 & PC1 respectively.

3.) Configure the IP address for routers and static / default routing.

Router 0:

Router > enable

Router # config t

Router (config) # interface Fa 0/0

Router (config-if) # ip address 10.0.0.2 255.0.0.0

Router (config-if) # no shut.

Router (config-if) # exit.

Router (config) # interface Se 2/0.

Router (config-if) # ip address 30.0.0.2 255.0.0.0

Router (config-if) # no shut.

Router (config-if) # exit.

Router(config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config) # end.

Similarly configure for router 1 and router 2.

4.) In simulation mode, send a simple PDU from one PC to another.

5.) Click on PDU during every transfer to see the inbound & outbound PDU details, use capture to capture every transfer.

Observation

1.) Sending PDU from PC-1 to PC-2

PDU details:

At device 10.0.0.1

TTL: 255

At Router-0

TTL: 254

At Router-1

TTL: 253

At Router-2

TTL: 252

) sending PDU back from PC-2 to PC-1

At device 20.0.0.1

TTL : 128

At Router-2

TTL : 127

At Router-1

TTL : 126

At Router-0

TTL : 125.

At PC-1

TTL : 125

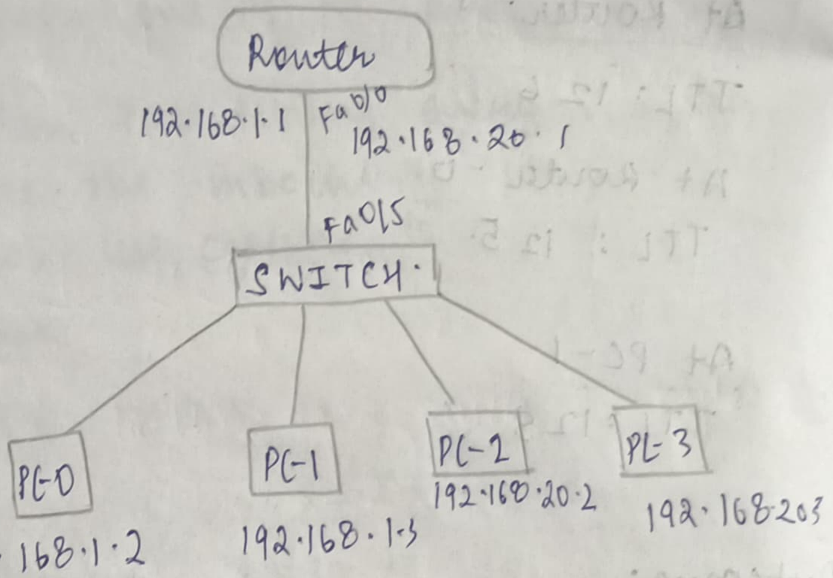
Outcome:

TTL (Time-to live) is a value in IP packet that tells a network router when the packet has been in the network too long and should be discarded. The TTL value instructs a network router when a packet should be discarded.

C) <u>VLAN</u>

Aim: To construct a VLAN and make the PC's communicate among a VLAN.

Topology -



Procedure:

1.) Create a topology as given above using 4PC's, 1 switch (switch-2960) and a router (router-1841)

2.) Configure the IP addresses for PC's as 192.168.1.2 & 192.168.1.3 for PC0 & PC1 and 192.168.20.2 & 192.168.20.3 for PC2 & PC3 respectively

3. Configure the IP address for router using following commands

Router > enable

Router # config t

Router(config)# interface Fa 0/0
Router (config-if)# ip address 192.168.1.1 255.255.0.0
Router (config-if)# no shut
Router (config-if)# exit.
Router (config

4.) set the gateway as 192.168.1.1 for PC0 and PC1 and 192.168.20.1 for PC2 and PC3 respectively.

5.) In the switch, go to VLAN database and create/add new VLAN database by name VLAN-new.

6.) Now, go to interface Fa0/5 in the switch and make it trunk, In VLAN everything needs to be selected. This allows different VLAN's to send frames over a single link called trunk.

7) Now, go to router and select VLAN database. Enter the number & name of VLAN created before, go to CLI in the router and give the following commands:

Router (VLAN)# exit
APPLY completed
Exiting ...
Router# config t
Router (config)# interface Fa0/0.1
Router (config-subif)# encapsulation dot1q 2.

Router (config-sub if)# ip address 192.168.20.1
255.255.255.0

Router (config-sub if)# no shut

Router (config-sub if)# exit

8.) In the switch for Fa 0/3 & Fa 0/4
select VLAN and number as no
given for VLAN while creating.
Now, ping from PC 0 to PC 3.

~~PC~~ Observation:

PC> ping 192.168.20.2.
pinging 192.168.20.2 with 32 bytes of data
Reply from 192.168.20.2 bytes = 32 time=4ms TTL
Reply from 192.168.20.2 bytes = 32 time=0ms
Reply from 192.168.20.2 bytes = 32 time=3ms
Reply from 192.168.20.2 bytes = 32 time=1ms

ping statistics for 192.168.20.2
packets: Sent = 4, Received = 4, Lost = 0

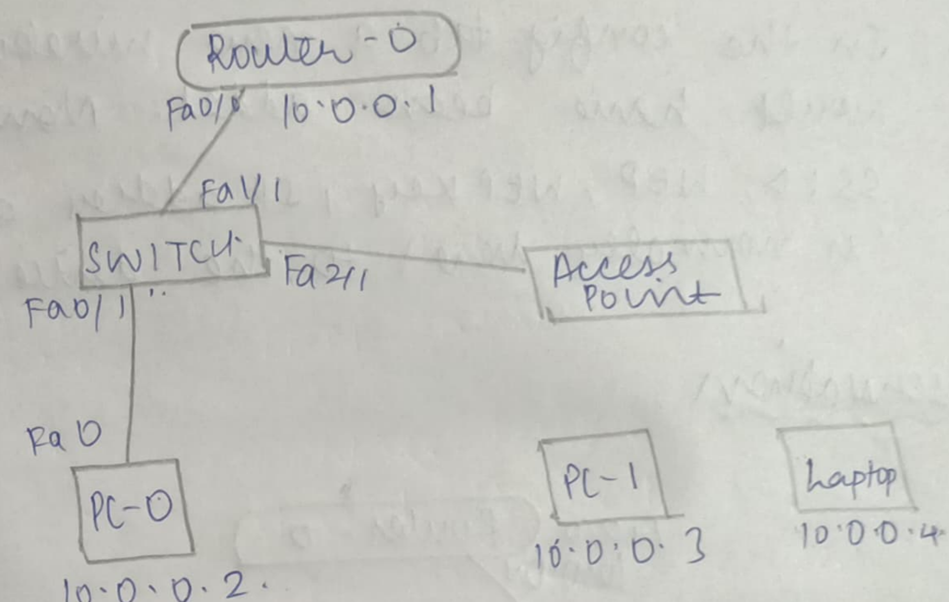
Outcome:

A virtual local area network (VLAN) is a virtualized connection that connects multiple devices and network nodes from multiple LAN's into 1 logical network

D) WLAN

Aim : To construct a WLAN and make the nodes communicate wirelessly.

Topology:



L=127

TTL=127

TTL=127

TTL=127

Procedure:

1.) Create the topology as shown above with PC's, switch, router, Access point and laptop.

2.) Configure PC0 and router.

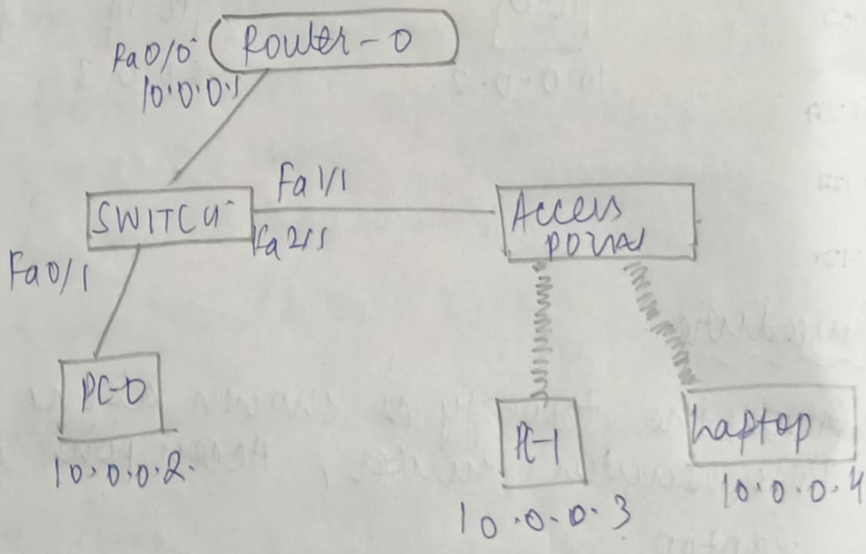
3.) Configure the Access point, go to port 1 and give the SSID name.

4.) Select WEP and give any 10 digit Hex key (9876543210). Configure PC1 and laptop with wireless standards.

5.) Switch off the device. Drag the existing PT-HOST-NM-1AM to the component listed in the LHS.
Drag WMP300N wireless interface to the empty port. Switch on the device.

6.) In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and Gateway (as normally done) to the device

Observation



PC0

PC> ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data.
Reply from 10.0.0.3: bytes=32 time=47ms TTL
Reply from 10.0.0.3: bytes=32 time=32ms TTL
Reply from 10.0.0.3: bytes=32 time=35ms TTL
Reply from 10.0.0.3, bytes=32 time=3ms TTL
Packets: sent=4, Received=4 lost=0

Outcome:

A WLAN stands for wireless LAN is a group of devices that form a network based on radio transmissions rather than wired connections and how we can configure it in a topology.

21/09/2023

Lab-09. (Cycle-2)

Exp: 13

Aim: Write a program for error detection using CRC-CCIT (16 bits)

```
#include<stdio.h>
#include<string.h>
# define N strlen(gen_poly)
char data [28]
char check_value [28]
char gen_poly [16];
int data_length i,j;

void XOR() {
    for(j=1; j<N; j++)
        check_value[j] = ((check_value[j] == gen_poly[j])?0
}

void reciever() {
    printf("Enter the recieved data");
    scanf("%s", data);
    printf("Data recieved %s", data);
    crc();
    for(i=0; (i<N+1) && check_value[i]='1'); i++);
        if (i< N-1)
            printf("\n Error detection \n");
    else
        printf("\n No error detected");
}
```

```
void crc() {
  for (i=0; i<N; i++)
    check_value [i] = data[i];

  do {
    if check_value [0] == '1')
      XOR();
    for (j=0; j< N-1; j++)
      check_value [j] = check_value [j+1];
    check_value [j] = data [i++);
  } while ((i< data length +N - 1))

}

int main()
{
  printf("Enter data to be transmitted");
  scanf("%s", data);
  printf("Enter the generating polynomial");
  scanf("%s", gen_poly);
  data_length = strlen (data)
  for (i= data_length, i< 17; i++)
      data [i] = '0'
  printf("\n");
  printf("Data : %s", data);
  printf("\n")
  crc();
  printf("CRC value : %s", check_value)
```

```
for (i = data_length, i < 17; i++)
    data [i] = check value[ i - data_length]

  printf ("\n")
printf ("Final data : %s", data).
receiver()
  return 0;
}
```

Output

Enter data to be transmitted : 1011011111
Enter the generating poly : 10111
Data padded with zeroes : 10110.11111000 00000
CRC : 00010
Final data to be sent : 101101111100010
Enter the received data : 101101111100100

Data received : 101101111100100
Error detected :

Experiment - 14

Aim: Write a program for congestion control using Leaky Bucket Algorithm.

```
# include <stdio.h>
int main() {
    int incoming, outgoing, buck_size, n, store=0.
    printf ("Enter bucket size, Outgoing rate & no of i/p)
    scanf("%d %d %d", & buck_size, & outgoing, &n);

    while(n != 0) {:
        printf ("Enter the incoming packet size : ");
        scanf("%d", & incoming)
        printf ("Incoming Packet size : %d", incoming).
        if (incoming <= (buck_size - store)) {
            store += incoming
            printf ("Bucket Buffer size %d out of %d",
                    store, buck_size);

        }
        else {
            printf ( Dropped %d no. of packets", incoming-(buck_size
                                                        -store
            printf ("Bucket buffer size %d out of %d", store,
                                                    buck_size
            store = buck_size
        }
        store = store - outgoing
```

```
store = store - outgoing
printf("After outgoing %d left out of %d in
        buffer", store, buck-size)

    n--;
  }
}
```

Output

Enter bucket size, outgoing rate and no of inputs:

10
2
4

Enter the incoming packet size : 4
Incoming packet size = 4
Bucket buffer size 4 out of 10

After outgoing 2 packets left out of 10 in buffer

Enter the incoming packet size = 8
Incoming packet size = 8
Bucket buffer size 10 out of 10
After outgoing 8 packets left in buffer
Incoming packet size = 6
Dropped 4 no. of packets

Bucket buffer size : 8 out of 10
After outgoing 8 packets left out of 10 in buffer

03/09/2023

Lab-10

Experiment 15

Aim: Using TCP/IP sockets, write a server-client program to make client sending the file name and the server to send back the contents of the requested file if present.

ClientTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket = connect((serverName, serverPort))
sentence = input("\n Enter file name")
clientSocket = send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n From Server")
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

```
while (1):
    print ("Server ready to recieve\n");
    connection socket, addr = server socket. accept();
    sentence = Connection Socket. recv(1024). decode().
    file = open (sentence, "r")
    l = file. read (1024)
    connectionSocket . sent (l. encode())
    print ("\n Sent contents of " + sentence + " to " + addr
    file. close ()
connect. socket. close ()
```

Output

Client

Enter file name: file.txt

Repely from server:

TCP IP Program

Server

Sent contents of: file.txt

Experiment - 16

Aim: Using UDP sockets, write client server program to make client send file name and server send back the contents of the requested file.

Client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file name: ");
clientSocket.sendto(bytes(sentence, "UTF-8"),
                    (serverName, serverPort))

filecontents, addr = clientSocket.recvfrom(2048)
print("\n Reply from server")
print(filecontents.decode("UTF-8"))

clientSocket.close()
```

Server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF-INET, SOCK-DGRAM)
serverSocket.bind(("127.0.0.1", 12000))
print("Server ready")
```

```
while(1):
sententence, addr = server socket . recv from (2048)
sentence = sentence . decode ("UTF-8")
 file = open (sentence, "r")
 con = file . read (2048)

 server Socket . send to (bytes (con, "UTF-8"), addr)
 print ("\n Sent contents of ' ", end = ' ' )
   print (sentence)
  file . close ()
```

Output

Client

Enter file name : file1.txt

reply from server

UDP socket prgm

Server

sent contents of : file1.txt