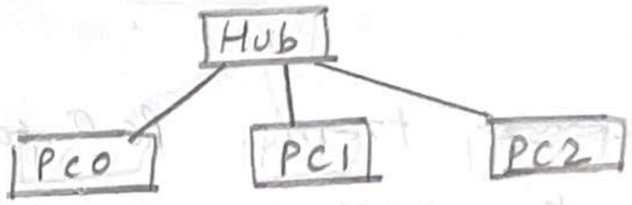
Experiment - 1

Create a topology consisting of 3 or more devices, connected with the help of a hub.



Place one hub and 3 end devices

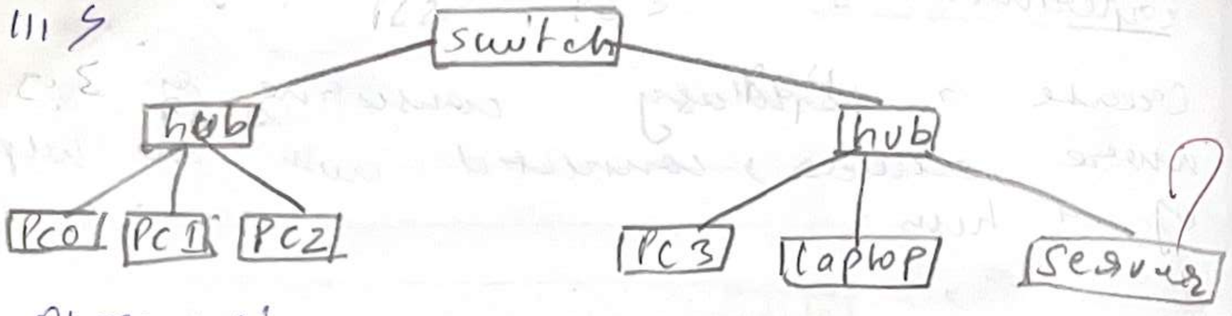
Observation:

Whenever a source node sends data in a network, hub receives the data and sends the data to all the remaining nodes. The node with matching destination address accepts the data

(ii)

Observation:

Here the difference is that when the source nodes sends data to other node then switch only sends to nodes whose address match.



Observation

Sending Message from PC 0 to PC 3

i) PC0 sends the message > Hub 1 recieves the message and transmits to PC 1, PC 2 both of which rejects the message as destination address does not match,

ii) Then switch transmits the message to the Hub 2, then Hub 2 sends to the PC 3, ~~and~~ ~~PC3~~ laptop and server.

iii) PC 3 accepts the message and acknowledges BACK.

laptop and server rejects the message.

Date: 30/9/23

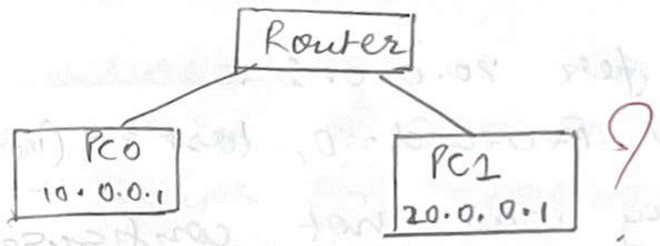
(3)

Experiment - 2

Aim: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

i)

Topology:



Procedure:

Steps 1: Place one router (Router - PT) & 2 end devices.

2: Connect the end devices to the router with appropriate cable.

3: Set IP address and default gateway to each end device.

4: Configure the router using CLI.

5: Put IP address same as the respective gateway address of the desktop.

6: Select PC0 & open the command prompt and ping the PC1 using its ip address.

Command prompt Output:

In PC1,

ping 20.0.0.2

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out
Request timed out
Request timed out
Request timed out

Ping statistics for 20.0.0.2:
Packets: Sent = 4, Recieved = 0, lost = 4 (100% loss)

- since gateway was not configured.

ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

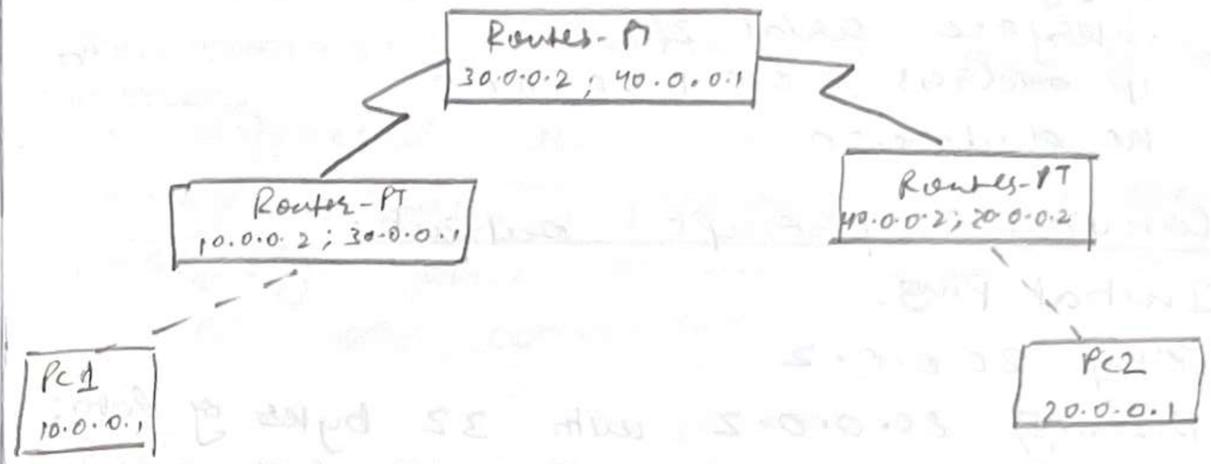
Reply from 20.0.0.1: bytes = 32 time = 0 ms TTL = 127
Reply from 20.0.0.1: bytes = 32 time = 0m TTL = 127
Reply from 20.0.0.1: bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.1: bytes = 32 time = 0ms TTL = 127

Ping statistics for 20.0.0.1:
Packets: Sent = 4, Recieved = 4, lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Observation:

Each data packets sent across the network contains address information that a router can use to determine that if the source and destination are on the same network.

11) Topology:



Procedure:

Step 1: Place 2 PC's on workspace and 3 router on workspace.

Step 2: Connect 1 PC to one router and another PC to other router and connect these 2 router to third routers.

Step 3: Set the IP address of 1st PC as 10.0.0.2 and gateway to 10.0.0.2. Set the IP address of 2nd PC as 20.0.0.1 and gateway to 20.0.0.2

Pinging PC2 from PC1 shows destination host unreachable.

Step 4: In CLI of PC1

```
> enable
config terminal
interface fastethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
```

Step 5: Adding static routes to router using:

>enable
config terminal
interface serial 2/0
ip address 30.0.0.1 255.0.0.0
no shutdown.

Command prompt output:

Initial Ping:

ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable
Reply from 10.0.0.2 : Destination host unreachable

Ping statistics for 20.0.0.2:
Packets: Sent = 4, Recieved = 0, lost = 4 (100% loss)

~~Since, IP address~~

Observation:

Since, IP address 20.0.0.2 is not directly connected to router 1, so manually 20.0.0.2 needs to be connected to router 1.

After Adding static Routes to every router:

Command prompt output.

ping 10.10.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2 : bytes = 32 time = 0m TTL = 255
Reply from 10.0.0.2 : bytes = 32 time = 0m TTL = 255
Reply from 10.0.0.2 : bytes = 32 time = 0m TTL = 255
Reply from 10.0.0.2 : bytes = 32 time = 0m TTL = 255
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Recieved = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Observation:

* As expected after adding ~~routes~~ static routes, packets can be sent successfully since all the routers and devices are connected.

ping 30.0.0.2

(18)

Lab-3

Static Routing

Aim: Configure default route, static route to the Routers.

Same Topology

Procedure:

Adding Static Routes

1. To router 1 for networks 20.0.0.0 and 40.0.0.0.

Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.1
Routes(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1

2. To router 3 for networks 10.0.0.0 and 20.0.0.0

Routes(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.2
Routes(config)# ip route 20.0.0.0 255.0.0.0 40.0.0.2

3. To router 2 for networks 10.0.0.0 and 30.0.0.0

Routes(config)# ip route 30.0.0.0 255.0.0.0 40.0.0.1
Routes(config)# ip route 10.0.0.0 255.0.0.0 40.0.0.1

Observation:

The ip routes have been added to each routers, which can be seen by running 'show ip route' command.

Router-1

show ip route
C 10.0.0.0/8 is directly connected, Fast Ethernet 0/0
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, serial 2/0
S 40.0.0.0/8 [1/0] via 30.0.0.1

Routes 2

show ip route
S 10.0.0.0/8 [1/0] via 40.0.0.1
C 20.0.0.0/8 is directly connected, Fast Ethernet 0/0
S 30.0.0.0/8 [1/0] via 40.0.0.1
C 40.0.0.0/8 is directly connected, Serial 2/0

Routes-3

show ip route
S 10.0.0.0/8 [1/0] via 30.0.0.2
S 20.0.0.0/8 [1/0] via 40.0.0.2
C 30.0.0.0/8 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 3/0

Output

The ping request to all networks are successful

From PC0

→ Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data.

Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125

Ping statistics for 20.0.0.1:
Packets: Sent=4, Recieved=4, lost=0 [0% loss]

→ ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2: bytes=32 time=7ms TTL=253
Reply from 40.0.0.2: bytes=32 time=7ms TTL=253
Reply from 40.0.0.2: bytes=32 time=6ms TTL=251
Reply from 40.0.0.2: bytes=32 time=6ms TTL=253

Ping statistics for 40.0.0.2
Packets: Sent=4, Recieved=4, lost=0 [0% loss]

RT-0

From PC1

→ Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Reply from 30.0.0.2 : bytes=32 time=2ms TTL=253
Reply from 30.0.0.2 : bytes=32 time=19ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253

Ping statistics for 30.0.0.2:
Packets = Sent=4, Recieved=4, lost=0 (0% loss)

→ Ping 10.0.0.1

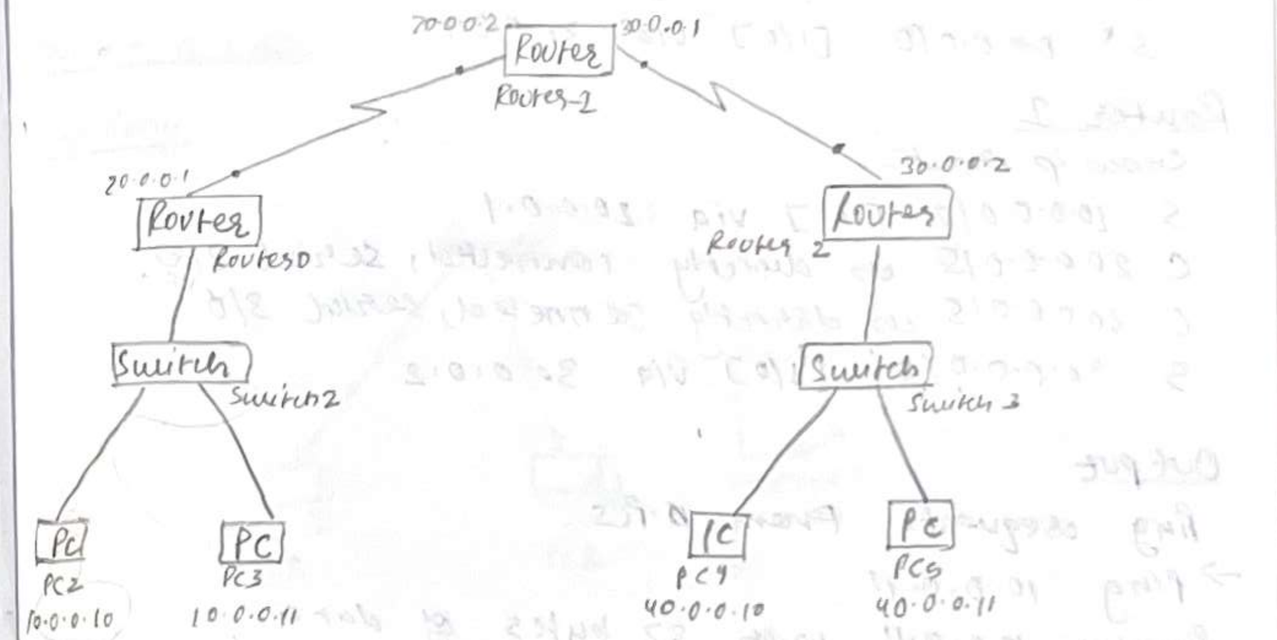
Pinging 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 : bytes=32 time=7ms TTL=125
Reply from 10.0.0.1 : bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125

Ping statistics for 10.0.0.1:
Packets: Sent=4, Recieved=4, lost=0 (0% loss)

Default Routing

Topology



Procedure

- Configuring Default routes to Router 0 and Router 2

Router 0
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1

- Configuring two static routes to Router 1

Router 1
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

Observation

The default routes to routes 0 and routes 2 and the static routes to routes 1 have been added.

Router 0

show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 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

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

Router 1

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

Output

Ping requests: From PCs

→ Ping 10.0.0.11

Pinging 10.0.0.11 with 32 bytes of data

Reply from 10.0.0.11: bytes=32 time=10ms TTL=125

Reply from 10.0.0.11: bytes=32 time=6ms TTL=125

Reply from 10.0.0.11: bytes=32 time=8ms TTL=125

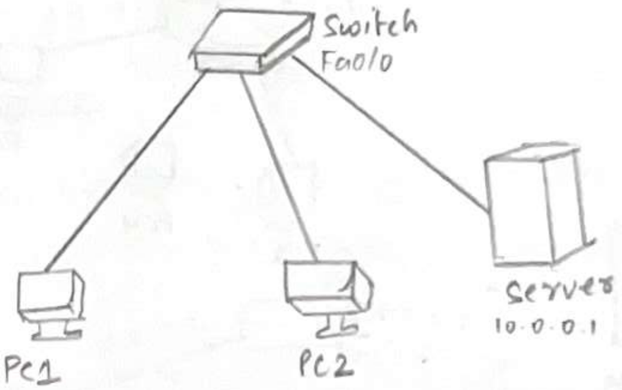
Reply from 10.0.0.11: bytes=32 time=5ms TTL=125

Lab-4

Aim: Configure DHCP within a LAN and outside LAN

Within a LAN

Topology:



Procedure:

1) Two PC's are connected to a switch with a server.
2) The server ip address is set to 10.0.0.1 and it is static
3) Then, in the server we go to services and turn the DHCP server to on.
5) We can give the pool name to server pool.
6) Then we set the maximum number of users for eg: 512
7) The we set the start IP address to start from 10.0.0.10.

Observation:

IP addresses of all the PC's connected are set automatically

Result:

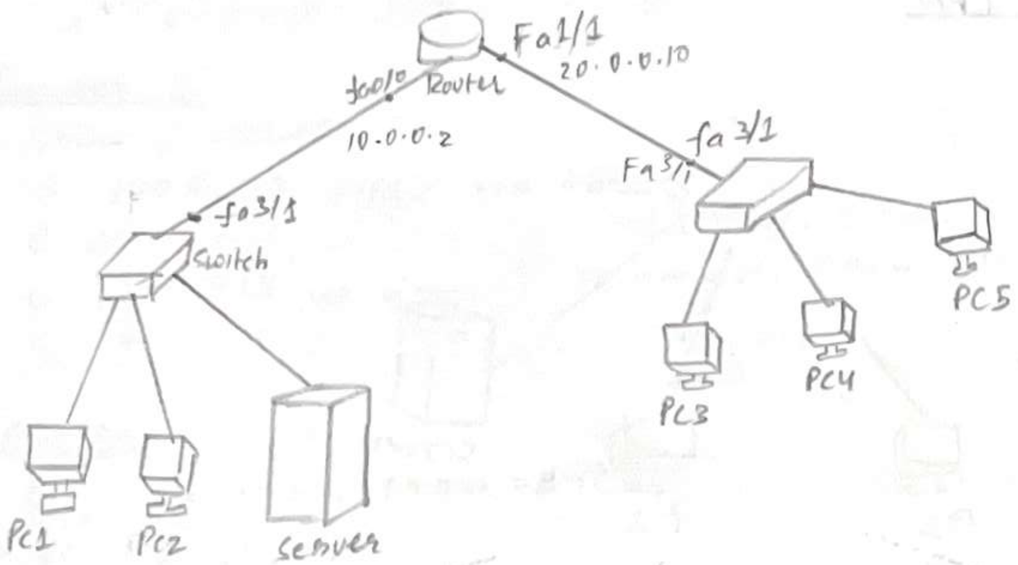
IP address:

PC1 : 10.0.0.3

PC2 : 10.0.0.2

Outside a LAN

Topology:



Procedure:

1) ~~If~~ Then another switch is connected through a router where 3 PC devices are connected.

2) Here, one server itself is used to assign IP address of the PC devices in other network.

Configuring Routes:

~~router~~ #interface fa0/0
#ip address 10.0.0.0 255.0.0.0
#no shutdown
#exit.

For other network
#interface fa1/0
#ip address 20.0.0.0 255.0.0.0
#no shutdown
#exit

3) The we have to add another server.
We can set the name as server pool 2.
We set the Default gateway as 20.0.0.1
Again we set the max user as 512.

Observation:

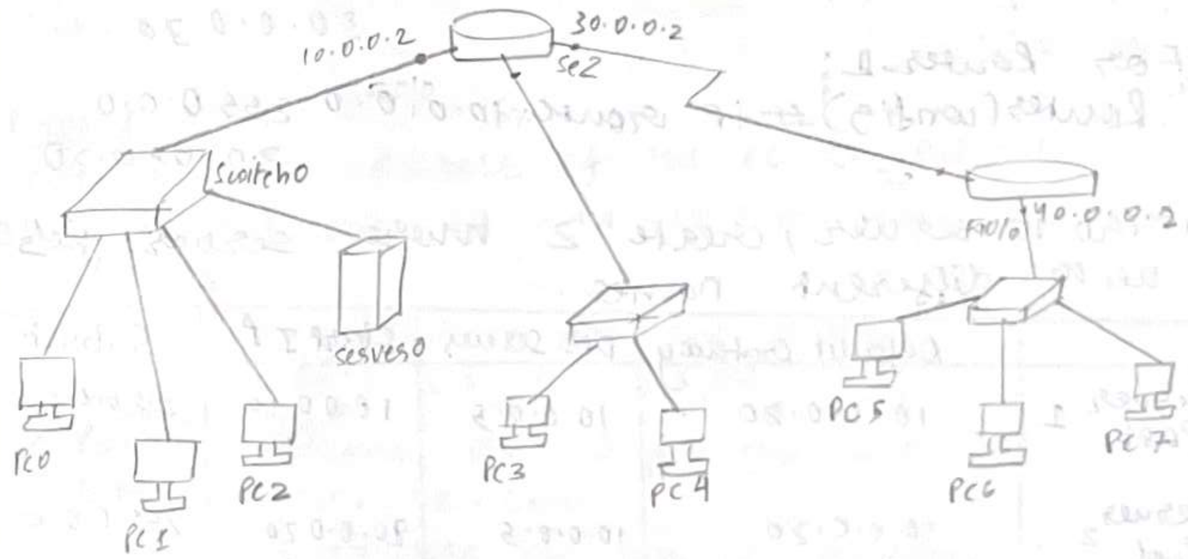
IP addresses of the PC's of other network is automatically allotted.

Result:

IP address:
PC3: 20.0.0.2
PC4: 20.0.0.3
PC5: 20.0.0.4

Outside LAN

Topology:



Procedure:

1) To the previous topology we add another network with the router-1, switch and 2 PC's.

2) Config the ip address for ports Fa 4/0 and Fa 0/0 of R0.

Router > enable
config terminal
interface Fa4/0
ip address 10.0.0.20 255.0.0.0
ip helper-address 10.0.0.5
No shutdown
exit

Similarly do for Fa1/0.

~~Ftp~~
Ip helper address is the address of server0.

4) We turn on the DHCP for the 3 newly added PC's.

5) Then router 1 and router 0 is connected Router 1 is connected to switch 2 and 3 PC's. (PC5, PC6, PC7)

6) Config ip address of router 1 for interface serial 2/0 and Fa0/0 with 30.0.0.30 and 40.0.0.20 respectively.

7) Config ip address of router 0 for serial 2/0 as 30.0.0.20.

8) Perform static routing for Routes 0:

Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.30

For Router 1:

Routes(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.20

9) Go to server, create 2 more server pools with different names.

	Default Gateway	DNS Server	Start IP	Subnet
Server Pool 1	10.0.0.20	10.0.0.5	10.0.0.10	255.0.0.0
Server Pool 2	10.0.0.20	10.0.0.5	20.0.0.20	255.0.0.0
server pool 3	10.0.0.20	10.0.0.5	40.0.0.10	255.0.0.0

10) Switch IP configuration of PC5, PC6, PC7 to DHCP.

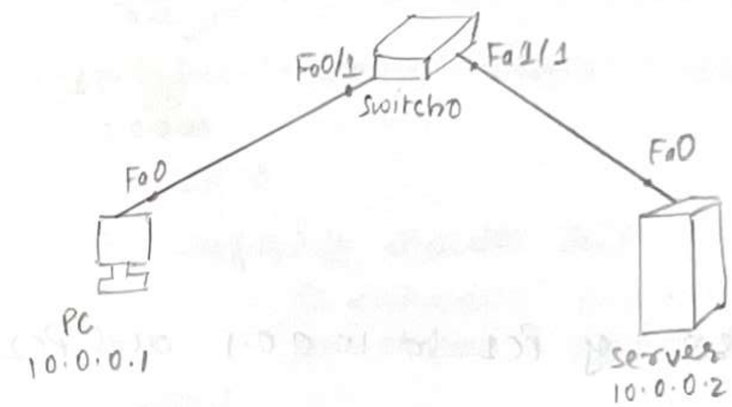
Obs. Observation

IP Addresses are set automatically using DHCP protocol by the server.

Lab-5

Aim: Configure web server, DNS with a lan and RIP routing Protocol in Routes

TOPOLOGY:



Procedure:

1) Set the ip address of the PC to 10.0.0.1

2) Then set the ip address of the server to 10.0.0.2

3) Then inside server we ~~set~~ go to services

4) We set DNS service to ON

5) Put the Name as BMSCE for the site for eg: BMSCE bmscecse.com and the address as the ip address of the server for eg: 10.0.0.2 and add it.

6) Then we switch to PC and ~~set~~ Inside web browser we can either put the name as the ip address of the site that we just saved.

7) DNS -> Then in index.html we can modify the HTML elements.

<h> RYAN THAPA </h>
<h1> 1BM21CS174 </h1>

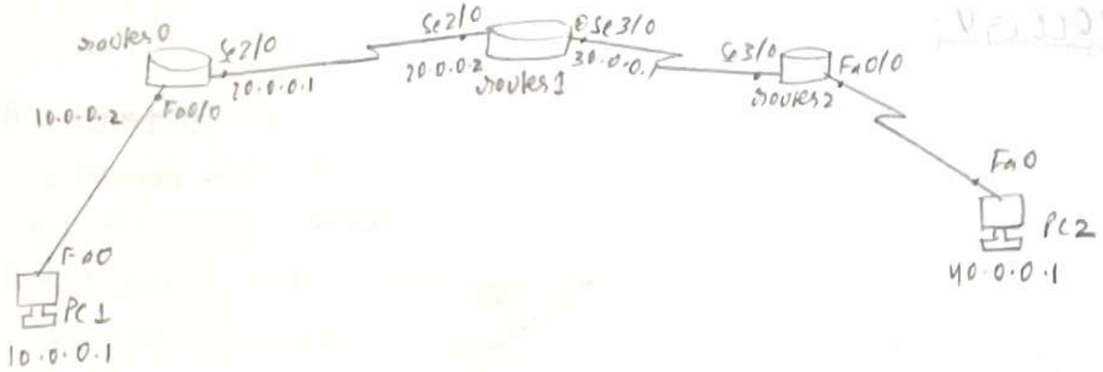
We save the html file.

Observation / Result:

When we access the site through PC, we should see our site with Name and USN.

RIP routing Protocol in Routers

Topology:



Procedure:

1) Set the ip address of PC1 to 10.0.0.1 and PC2 to 40.0.0.1

2) We add 3 router with router 0 connected to PC1 and router 2 to PC2

3) In CLI of router 0 well configure the IP address for the interface Fa0/0.

```
Router# configure terminal
Router(config)# interface Fa0/0
        # ip address 10.0.0.2  255.0.0.0
        # no shutdown
```

Similarly we'll set for router 2 for the interface Fa0/0 with ip address 40.0.0.1

4) To set the routing protocol

Router 0 CLI

```
Router(config)#interface Se2/0
        # ip address 20.0.0.1  255.0.0.0
        # encapsulation ppp
        # clock rate 64000
        # no shutdown
```

Router 1 CLI

```
Router(config)# interface Se2/0
        # ip address 20.0.0.2  255.0.0.0
        # encapsulation ppp
        # no shutdown
```

We notice that we don't put clock rate for the above interface.

similarly we set for Router 2 for interface Se3/0 without clock rate and ip address to 30.0.0.2

For router 1, with interface Se 3/0 we set the ip address to 30.0.0.1 with clock rate 64000 as we did for Router 0.

5) After the above steps we set the router's rip.

For Routers 0

```
Router(config)# routes rip
        #network 10.0.0.0
        #network 20.0.0.0
```

For Routers 1

```
        #routes rip
        #network 20.0.0.0
        #network 30.0.0.0
```

For Routers 2

```
        # routes rip
        # network 30.0.0.0
        # network 40.0.0.0
```

Observation:

After the rip routes have been set when we run the show ip route command we can see which routes is connect directly and indirectly.

Routers 0

```
Router > show ip route
C   10.0.0.0/8 is directly connected, FastEthernet 0/0
C   20.0.0.0/8 is variably subnetted, 2 subnets, 2
    masks
C   20.0.0.2/32 is directly connected, Serial 2/0
R   30.0.0.0/8 [120/1] via 20.0.0.2, Serial 2/0
R   40.0.0.0/8 [120/2] via 20.0.0.2, Serial 2/0.
```

We can see from topology that Routes 0 is directly connected through network 10.0.0.0 and 20.0.0.0 but 30.0.0.0 and 40.0.0.0 are indirectly connected via 20.0.0.2.

Routes 1:

Routes# show ip route.

C 20.0.0.0/8 is directly connected, Serial 2/0
C 30.0.0.0/8 is directly connected, Serial 3/0
R 10.0.0.0/8 [120/1] via 20.0.0.1
R 40.0.0.0/8 [120/1] via 30.0.0.2, Serial 3/0

Routes 2:

Routes# show ip route

R 10.0.0.0/8 [120/2] via 30.0.0.1, Serial 3/0
R 20.0.0.0/8 [120/1] via 30.0.0.1, Serial 3/0
C 30.0.0.1/32 is directly connected, Serial 3/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0

Result:

CMD:

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=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=6ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent=4, Recieved=4, Lost=0 (0% loss)

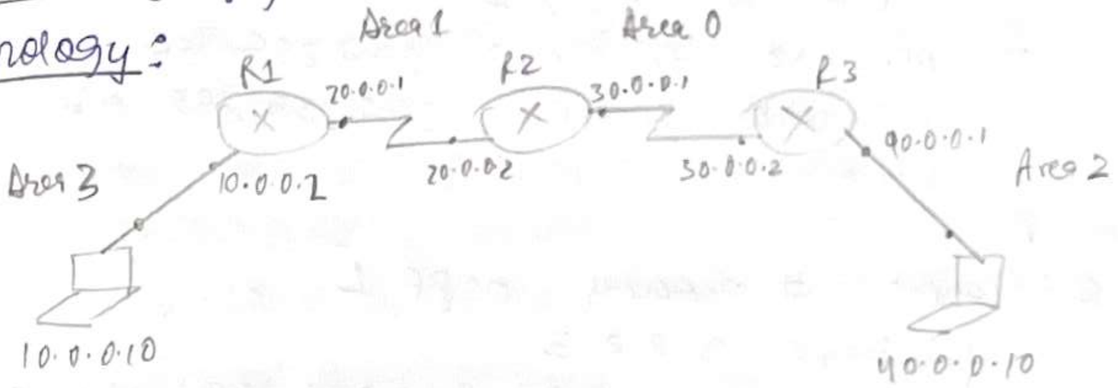
Observation:

We can see that we are able to successfully ping from PC0 to PC1.

Experiment - 6

Aim: Configure OSPF routing protocol

Topology:



Procedure:

1) Create topology using 3 routers and 2 PC's
2) Configure ip address to all interfaces

In R1

interface fa0/0
ip address 10.0.0.1 255.0.0.0
no shutdown
exit
interface se 1/0
ip address 20.0.0.1 255.0.0.0
encapsulation PPP
clock rate 64000
no shutdown
exit

Encapsulation PPP for all interfaces
Clock rate 64000 for interfaces with clock.

3) Enable rip routing in all routing

In Routes R1,

R1 (config) # routes ospf 1
R1 (config-routes) # routes 10.0.0.0 255.255.255.255 area 2

rel
routes id 1.1.1.1
network 10.0.0.0 255.255.255.255 area 3
network 20.0.0.0 255.255.255.255 area 1
exit.

In R2,

R2(config) # router ospf 1
router-id 2.2.2.2
network 20.0.0.0 255.255.255.255 area 1
network 30.0.0.0 255.255.255.255 area 0
exit

In R3,

R3(config) # router ospf 1
router-id 3.3.3.3
network 30.0.0.0 255.255.255.255 area 0
network 40.0.0.0 255.255.255.255 area 2
exit

Routes id identifies the router

4) Checking routing table of R2

Routes # show ip route

Gateway of last resort is not set

C 10.0.0.0/8 is directly connectly
C 20.0.0.0/8 is directly connected
O IA 40.0.0.0/8 via 20.0.0.2
O IA 30.0.0.0/8 via 20.0.0.2

Here R2 knows Area 0 and network 20.0.0.0 connected to R2 from R1, so R1 learns networks through this network.

~~R5 config~~

5) Configure loopback address to the routers

R1(config-if) # interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shutdown

R2(config-if) # interface loopback 0
add 172.16.1.253 255.255.0.0
no shutdown

R3(config-if) # interface loopback 0
ip address 172.16.1.254 255.255.0.0
no shutdown

6) Checking routing table of R3

R3 # show ip route

Gateway of last resort is not set

O IA 20.0.0.0/8 via 30.0.0.1
C 40.0.0.0/8 is directly connected
C 30.0.0.0/8 is directly connected.
C 172.16.0.0/16 is directly connected, Loopback0

R3 doesn't know about area 3 so we create virtual link.

6. Create virtual link between R1, R2 by this we create a virtual link to connect area 3 to area 0.

R1(config) # router ospf 1
area 1 virtual-link 2.2.2.2

R2(config-router) # area 1 virtual-link 1.1.1.1

7) R2 and R3 get updates about Area 3.

Check routing table of 3

R3# show ip route

Gateway of last resort not set

O IA 20.0.0.0/8 via 30.0.0.1
C 40.0.0.0/8 is directly connected
O IA 10.0.0.0/8 via 30.0.0.2
C 30.0.0.0/8 is directly connected, Serial 1/0

Observation / Result:

ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10 : bytes: 32 time=10ms TTL=125
Reply from 40.0.0.10 : bytes: 32 time=10ms TTL=125
Reply from 40.0.0.10 : byts : 32 time=0ms TTL=125
Reply from 40.0.0.10 : bytes: 32 time=0ms TTL=125

Ping statistics for 40.0.0.10 :

Packets: Sent = 4, Recieved = 4, lost = 0 (0% loss)

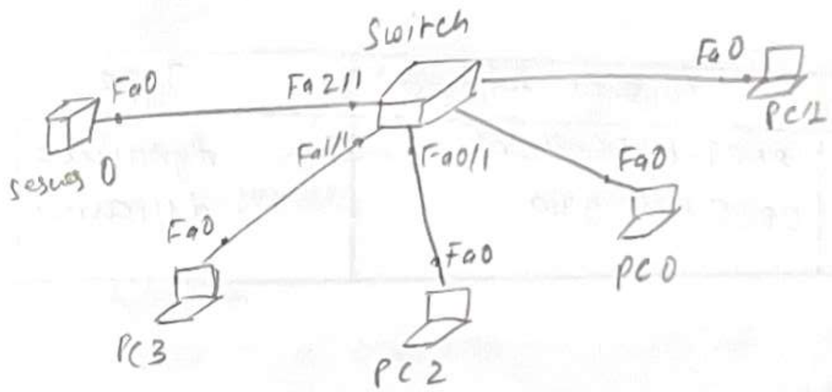
Approx round trip times in ms

Minimum = 6ms, Maximum = 11ms, Average = 8ms

Experiment - 7

Aim: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology:



Procedure:

1) Create topology of 4 PC's and a server
2) IP address is assigned to all.
3) Connect them through a switch
4) Use 'inspect tool' to click on PC to see the ARP table.
5) Command in CLI for the same is arp -a.
6) Initially ARP table is empty.
7) Also in CLI of switch, the command - show mac address table can be given on every transaction to see how the switch learns from transactions and build address table.
8) Use capture button to see step by step changes

Observation / Result

ARP Table for PC1

IP Address	Hardware - Address	IIf
10.0.0.1	0001.635C.43B0	Fa0
10.0.0.3	0000.0C51.041A	Fa0

PC > arp -a

Internet address	Physical Address	Type
10.0.0.1	0001.635C.43b0	dynamic
10.0.0.3	0000.0C51.041a	dynamic

Ping

ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data:

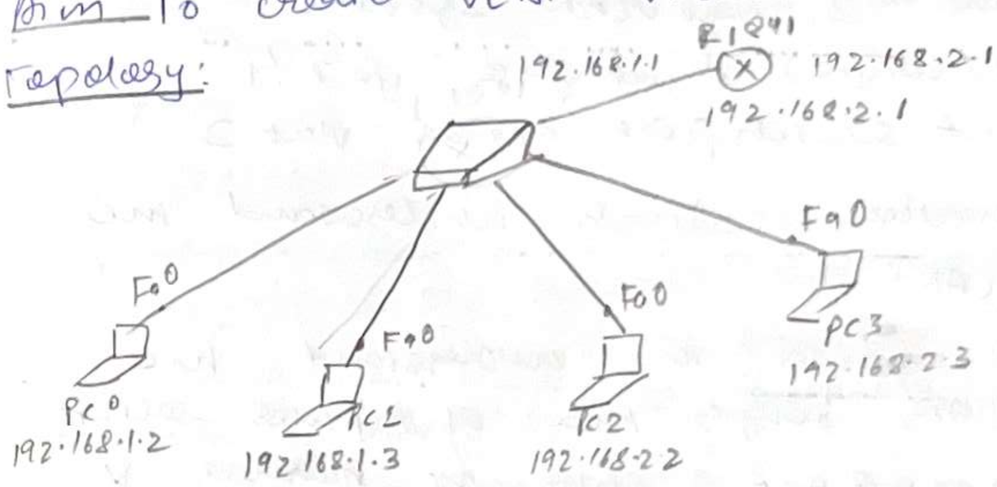
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL=128

Packets sent=4 Recieved=4 lost=0 (0% loss)

Experiment 8

Aim: To create VLAN - Virtual LAN

Topology:



Procedure

1) Create a topology using 4 PC's, a switch and cross the 1941 routes.

2) In switch, go to config tab and select VLAN database

3) Give any VLAN number, say 2. Include any name, say NEWVLAN.

switch (config) # vlan 2
name NEWVLAN
exit
interface Fa 4/1

4) Select the interface fast ethernet 4/1 and make it the trunk.

VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk. This is done by adding an additional header information called tag to the Ethernet frame.

The process of adding this small header is called VLAN tagging.

5) Look into the interfaces of all switches which are 2 New VLAN systems

switch(config) # interface fa 2/1
switchport access vlan 2

This makes switch understand me NEWVLAN.

6) For routers to understand me NEWLLAN, configs two of routers select VLAN DATABASE, enter the number & name of the VLAN created.

Router CLI:
Router(vlan) # exit
APPLY completed
Exiting...
config terminal
interface fa0/0.1
Router(config-subif) # encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit

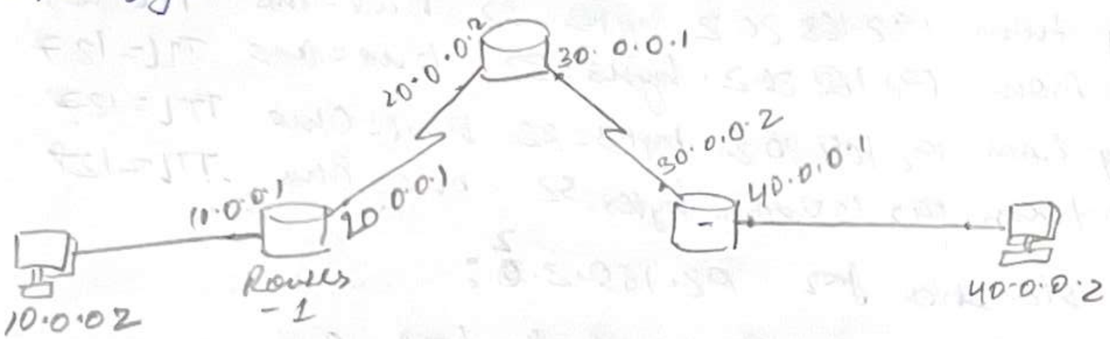
Output:
Ping from PC-1 to:
ping 192.168.1.3
pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=0ms TTL=120
Reply from 192.168.1.3: bytes=32 time=0ms TTL=120
Reply from 192.168.1.3: bytes=32 time=0ms TTL=120
Reply from 192.168.1.3: bytes=32 time=0ms TTL=120

Ping statistics for 192.168.20.2:
Packets: Sent=4, Recieved=4, lost=0

Ping from PC 1 to PC-2
ping 192.168.2.2
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.2.2:
Packets: Sent=4, Recieved=4, lost=0

Aim: To demonstrate the TTL life of a packet

Topology:



~~Step~~

Procedure:

Step 1: Create topology with 2 PC and 3 router as shown above.

Step 2: Configure the IP address as 10.0.0.2 and 40.0.0.2 from PC1 and PC2

Step 3: Configure the IP address of routers and the static and ~~of~~ default routes.

Router 1 - Configuration:

```
Router(config)# interface fa0/0
                ip add 10.0.0.1 255.0.0.0
                no shut

Router(config)# interface S0/0
              # ip route 0.0.0.0 0.0.0.0 20.0.0.2
```

Similarly configure the IP address of routers 2 and 3

Configuring static route: for routes 2

```
# ip route 10.0.0.0 255.0.0.0 20.0.0.1
  ip route 40.0.0.0 255.0.0.0 30.0.0.2
```

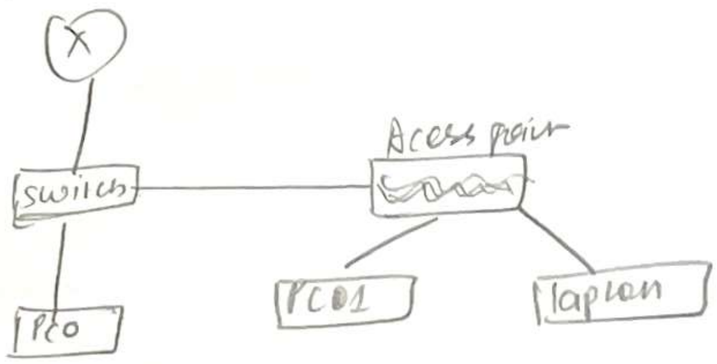
Output:

sending a PDU from PC1 to PC2

We can see the TTL value at the PDU table change

AIM: To construct a WLAN and make the nodes communicated wirelessly

Topology:



Procedure:

① Pick the end devices, switch, access point and routes.

② Assign IP address to all device

③ Config the router
enable
config-t
interface fastethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shutdown

④ Click on the access port go to port → port address give an SSID assign a key

⑤ Click on PC2
→ Goto Physical → switch off the PC
→ replace the ethernet cable with
→ My Link Sys – WMP350 - -
→ Switch on the PC and the same for the laptop as well.

⑥ Click on PC2 → go the desktop
→ Wireless network
→ WLAN will be seen → click on that & enter pass
→ Click on connect.

⑦ Click on PC2 → go to desktop
→ Wireless network → WLAN
→ Password →
Repeat for laptop

⑧ send a PDU from PC0 to laptop and to see the result.

Result

Click on PC0
→ Ping 10.0.0.2
Reply from 10.0.0.2 bytes-32 time-0ms TTL=225
packets : sends = 4, Received = 4, lost = 0

→ ping 10.0.0.4
Reply from 10.0.0.4 bytes=32 time=26 ms TTL=200
Packet sent = 4, Recieved = 4, lost = 0 (0% loss

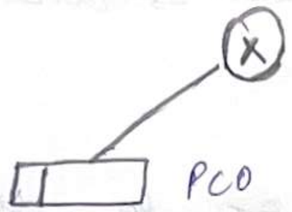
Observation

In the access post port is for wired connection and port for wireless connection

If the physical configuration of the devices is not changed then wireless connection can not be established.

Aim: To Demonstrate TELNET in single CAN

Topology:



Procedure:

1) Select the end device and router and connect.
2) Assign IP address to PC0 - 10.0.0.1
3) Configure the router with ip address
   Then
   - # enable secret pass
   # interface fastethernet 0/0
   # ip address 10.0.0.2 255.0.0.0
   # no shut
   # line vty 0 5
   # login
   # password P0
   # exit
4) click on PC0
   PC> ping 10.0.0.2

Reply from 10.0.0.2: bytes=32 time=0ms TTL=2
" " " " " "
" " " " " "
" " " " " "

PC> telnet 10.0.0.2
   User Access Verification
   password: P0
Router> enable
password: pass P0
Router # show IP route

C: 10.0.0.0/8 is directly connected, FastEthernet 0/0.

Result:
The routers were accessed by the PC0 successfully.

Observation:
For the remote login to successfully the password must be entered correctly and there are two different password for login and for giving commands.

Write a program for error detecting code using CRC- CCITT (16-bits)

```
#include <stdio.h>
#include <string.h>
#define CRC_Poly 0x1021

unsigned short calculate CRC(const char
        *data, int length) {
  unsigned short crc = 0xFFFF
  for(int i=0; i<length; i++) {
    crc ^= (unsigned short) data[i]<<8;
    for (int j=0; j<8; j++) {
        if (crc & 0x8000)
        crc = (crc <<1) ^ crc_poly;
      else
        crc <<=1;
    }
  }
  return crc;
}
int main() {
  char data[100];
  printf("Enter data :");
  scanf("%s", data);
  int data_length = strlen(data);
unsigned short checksum = calculate CRC
                (data, data length);
printf("Calculate CRC : 0x%.04X\n",
        checksum);
unsigned short recieved checksum;
printf("Enter recieved CRC:");
scanf("%hx", &recieved checksum);
if (recieved checksum == checksum) {
  printf("Data is error-free\n");
}
else {
  printf("Data contains errors\n");
}
return 0;
}
```

Output

Enter data : $x^{10} + x^{12} + x^5 + 1$

Calculated CRC : 0x FF8A

Enter recieved CRC : 0x FF8A

Data is error free.

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

- Server TCP.PY

```
From socket import *
server name = "127.0.0.1"
Server port = 12000
server socket = socket (AF_INET)
                        SOCK_STREAM )
server socket. bind ( server name, server_port)
server socket. listen (1)
while 1
    print "The server is ready to receive";
    connection socket; addr = server socket.accept()
    sentence = Connection socket. recv (1024)
        decode
    d = file. read (1024)
    connection socket sent (1. encodes)
    print f "\n sent contents of ' + sentence "
    file close ()
    connection socket. Close ()
```

Client TCP.PY

```
From socket import *
Server Name = 12000
client socket = socket (AF_INET, Sock_Stream)
client socket. connect (" Enter Name");
client socket = Input (" Enter file name");
file content = client socket. recv (1024) decode
print f ("\n from server \n")
print ( file contents)
client Socket. close()
```

Output

```
The server is ready to receive
Client TCP.py
Enter file name : d:\Ryan Mapa\CN\TCP\Server.TCP.py
from server
from socket = "import *
Server Name = "127.0.0.1"
Server Port = 12000
Server socket = socket (AF_INET, SOCK_STREAM
Server Socket. bind (( Server Name, Server Port))
Server Socket. listen (1)
while d :
    print ("The server is ready to
    received ")
Connection Socket, addr = Server Socket. Accept ()
Sentence = Connection Socket. recv (1024). d
                                    decode()
file = open (sentence, 'r')
d = file. read (1024)
connect socket. send ((encoded))
print f ("\n Sent Content of ' + sentence)
file. close()
connection Socket. close()
```

the server is ready to receive
Sent contents of server TCP.py
The server is ready to receive

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

Server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
print('\n Sent contents of ', sentence, ':')
print(sentence)
# for i in sentence:
    # print(str(i), end="")
file.close()
```

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))
file contents, serverAddress = clientSocket.recvfrom(2048)
print('\n Reply from Server:\n')
print(file contents.decode("utf-8"))
# for i in file contents:
  # print(str(i), end="")
ClientSocket.close()
ClientSocket.exit()
```

Output

```
The server is ready to receive
 Client UDP.py
Enter file name: d:\Kyamtaya\CN\UDP\server UDP.p
Reply from Server:
from socket import *
serverPort = 12000
 serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
while 1:
    print("The server is ready to receive")
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    l = file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"),
                                clientAddress)
print('\n Sent contents of', end=' ')
print(sentence)
# for i in sentence:
  # print(str(i), end="")
file.close()
```

Title: Write a program for congestion control using leaky bucket Algorithm

```
# int main ()
    int no_of_queries, storage, output_pkt_size
  int input_pkt_size, bucket_size, size_left;
  storage = 0;
  no_of_queries = 4;
  Bucket_size = 10,
  Input_pkt_size = 4;
  Out_pkt_size = 1;
  for (int i=0; i< no_of_queries; i++)
  size_left = bucket_size - storage;
  if (input_pkt_size <= size_left)
     storage += input_pkt_size;
  } else {
    printf ("Packet loss = %d\n", input_pktsize);
  }
  printf ("Buffer size = %d out of bucket
   size %d\n", storage, bucket_size);
  storage -= output_pkt_size;
  }
  return 0;
}
```

Output

Buffer size = 4 out of bucket size = 10

" " = 3 " " " = 10

" " = 2 " " " = 10

" " = 1 " " " = 10