VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (22CS4PCCON)

Submitted by

Chandrakala K M (1BM23CS403)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Academic Year 2024-25 (odd)

B.M.S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled " Computer Network (22CS4PCCON)" carried out by **Chandrakala K M (1BM23CS403),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Spoorthi D M Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

CYCLE - 1

Github Link: https://github.com/Chandrakala8050/computer-network-Lab-.git

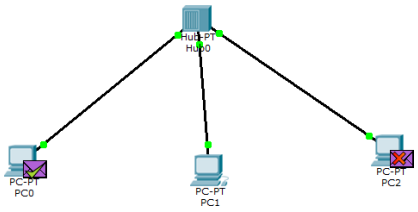
CYCLE - 2

Program 1

Aim of the program:
Simulate sending a simple PDU from source to destination using hub and switch as connecting devices

a) Using Hub

Procedure along with the topology:



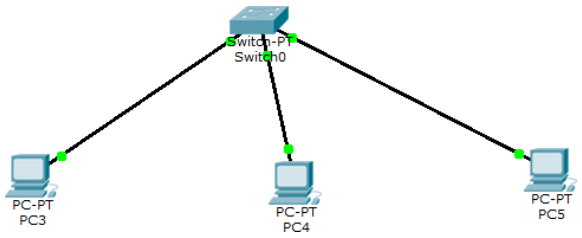
i. Select 3 end devices and a hub.
ii. Connect all the devices to the hub with a straight through cable.
iii. For each device do the configuration to set IP address for fast Ethernet.
iv. Add simple PDU, select source and destination and simulate the experiment.

Observation:

- Source - PCO, Destination - PC2
- Message is sent from PCO to hub.
- The hub sends the message to both PC1 and PC2 devices. PC1 rejects while PC2 accepts the message.
- PC2 sends feedback to the hub, hub sends it to PCO to PC1. PCO accepts while PC1 rejects the message.

b) Using Switch

Procedure along with the topology:



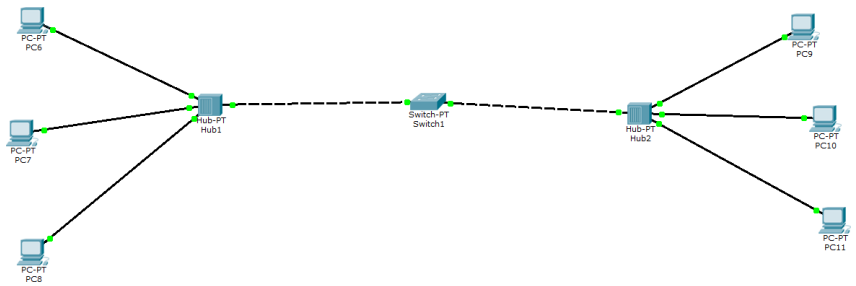
i. Select 4 end devices and a switch.
ii. Connect all the devices to the switch with straight through cable.
iii. For each device do the configuration. Set IP address for fast ethernet.
iv. Add simple PDU, select source and destination and simulate the experiment.

Observation:

- Source: PCO , Destination: PC2
- Message is sent from PCO to switch.
- Switch sends the message to PC2.
- Feedback is sent from to switch which then sends it to PCO.

c) Using switch and hub

Procedure along with the topology:



i. Select 1 switch, 2 hubs and 6 end devices.
ii. Connect first 3 devices to Hub 0 and other 3 devices to Hub 1, connect the two hubs to the switch.
iii. Do the configuration (set IP addresses) for each device.
iv. Add simple PDU, select source and destination and simulate the experiment.

Observation:

- Source: PCO , Destination: PC4
- Message is sent from PCO to Hub 0.
- Hub 0 sends message to PC1, PC2 and switch
- Switch sends to Hub 1, Hub 1 sends to switch, PC3, PC4 and PC5. PC4 accepts while others reject.
- Similarly, feedback is sent by PC4 to the PCO.

Observation Book:

LAB-1

25/09/24

①

Cisco Packet Tracer:

Packet Tracer is a medium fidelity, networks-capable. Simulation-based learning environment for networking novices to design, configure & troubleshoot Computer Network at a CCNA level of complexity.

Connections:

1) Console: It can be made Console Connections can be made between PCs & routers or Switches.

2) Copper Straight Through: This Cable type is the Standard Ethernet media for connecting b/w devices that operates at different OSI layers.

3) Copper Cross over: This Cable type is the Ethernet media for Connecting b/w devices that operates at the same OSI layer

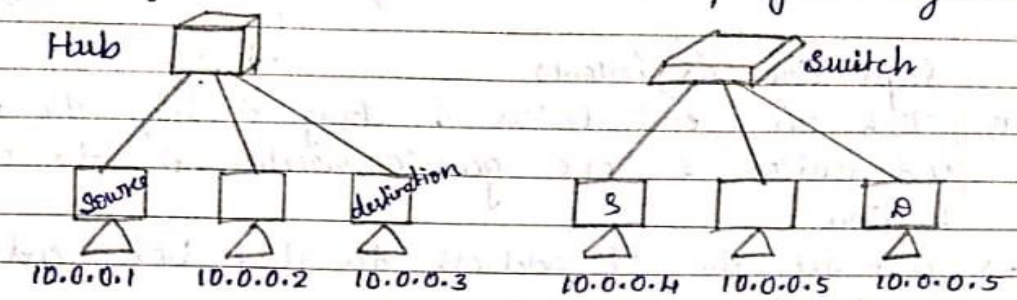
4) Fiber: Fiber media is used to make connections b/w fiber ports.

5) Phone: Phone line Connections can only be made b/w devices with modem ports.

6) Coaxial: Coaxial media is used to make Connections b/w Coaxial ports such as a Cable modem Connected to a packet tracer Cloud.

7) Octal: The 8-port asynchronous Cable provides the high density Connector on one end & eight RJ-45 plugs on the other.

1) Aim: Create a topology & Simulate sending a simple PDU from source to destination using hub & switch as connecting devices and demonstrate ping message.



Steps are as follows

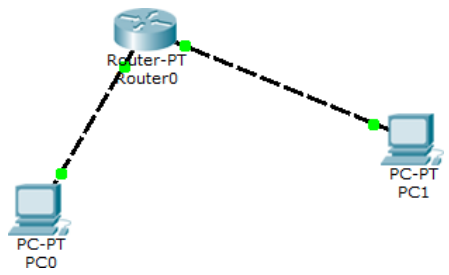
1) Setup the topology
   - Connect PC1, PC2 & PC3 to Hub
   - Connect PC4, PC5 & PC6 to Switch
2) Configure the IP addresses
3) Connect the copper straight through cable to the hub & switch.
4) Send a Ping from PC1 to PC3 in hub & Send a ping from PC4 to PC6 in Switch.
5) Observe the transmission Process

	Hub, 4/12 ports	Switch multi port & b/w 4 & 48 port.
1)	It is a broadcast device	It is a point to point device.
2)	It operates at physical layer	It operates at datalink layer
3)	It is not an intelligent device	It is an intelligent device
4)	It simply broadcasts the incoming packet	It uses the switching table to find the correct destination.
5)	It can be used as a repeater	It cannot be used as a repeater
6)	Not very costly	very costly.
7)	Not a Sophisticated device	It is a Sophisticated device.
8)	Transmission mode is Half duplex.	transmission mode is half / full duplex.

Program 2

Aim of the program:
Configure IP address to routers in packet tracer.

Procedure along with the topology:



Step 1

i. Place 2 generic PCs and one generic router.
ii. Connect both PCs to the router's fast ethernet ports using copper crossover wire.

Step 2

i. Select PC1 config -> Fast Ethernet 0
ii. Set IP address as 10.0.0.10 and default gateway 10.0.0.1.
iii. Similarly for PC 2 IP 20.0.0.10 and default gateway 20.0.0.1.

Step 3

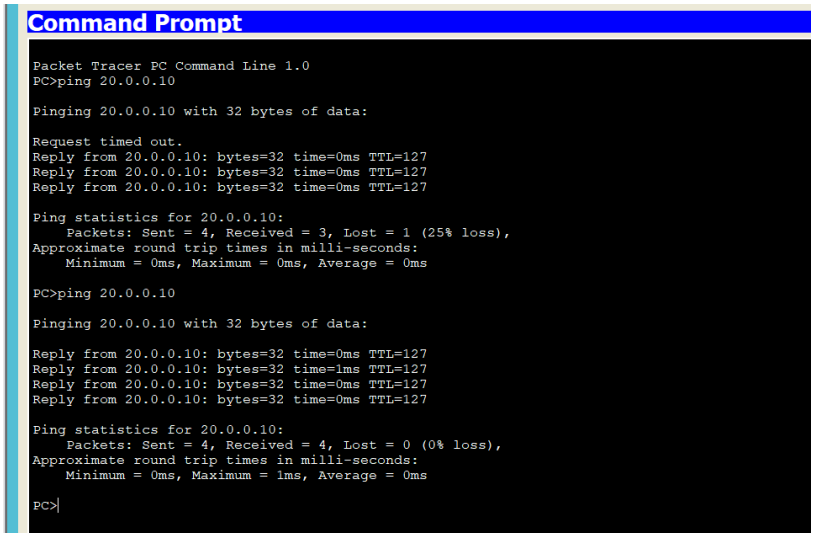
i. Select router and go to CLI
ii. Execute the following commands
    - >enable
    - #config terminal

- # interface Fast Ethernet 0/0
- # ip address 10.0.0.1 255.0.0.0
- # no shutdown
- # exit
  PC 0 and router are successfully connected

  Now run for PC 1
- interface Fast Ethernet 1/0
- # ip address 20.0.0.1 255.0.0.0
- # no shutdown
  Successfully connected

Step 4: Select the PC 0 and open the command prompt.
Step 5: Ping the PC 1 using the following command: ping 20.0.0.10
Step 6: Observe the output.

Screen shots/ output :



Observation:

PC 0 succesfully pings with PC1 with 32 bytes of data.

Observation Book:

LAB-02

9/10/24

(3)

Aim:-

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

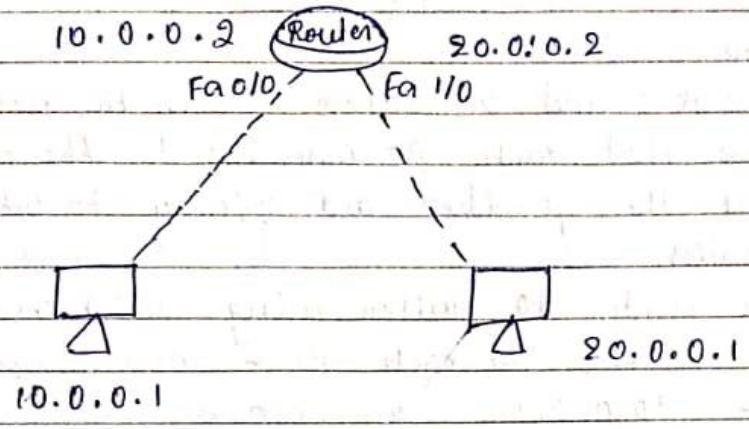
Steps are as follows

1) Click on end devices & drag & drop the 02 end devices & one generic router in the routes section.
2) then set the IP address to that both end devices & also set the gateway.
3) In router click on config tab then click on fastEthernet 0/0 then set the IP address & subnet mask do the same thing for fastEthernet 1/0 also.
4) In Router, click on CLI tab then run the below commands
   1) enable
   2) Config terminal
   3) interface fastEthernet 0/0
   4) ip address 10.0.0.1 255.0.0.0
   5) no shutdown
   6) exit

* enable
* show ip route : It will show the number of connected LAN's.

5) Next click on End devices, then click on Desktop → Command prompt → run ping command.

Topology: Router



Show ip Route:

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, FastEthernet 1/0.

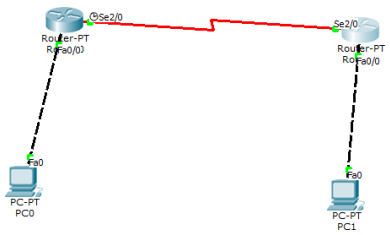
Observation:

→ we observed how ping command will be work &
→ how the Show ip Route command works.
→ how the Routes will work & what is the function of Routers.

Program 3

Aim of the program:
Configure default route, static route to the Router.(Part 1)

Procedure along with the topology:



i. Select a generic router R1.
ii. Connect an end device PC1 to router R1 through parallel connection - fastethernet 010.
iii. Configure PC1 with ip address 10.0.0.10 and gateway 10.0.0.2
iv. Similarly select another generic router R2 and connect an end device PC2 - fastethernet 010.
v. Configure PC2 with ip address 20.0.0.10 and gateway 20.0.0.2.
vi. Now, select router R1, go to CLI and execute the following.
- Router> enable
- Router# config terminal
- Router(config)# interface fastethernet 0/0
- Router(config-if)# ip address 10.0.0.1 255.255.255.0
- Router(config-if)# no shutdown

"Interface fastethernet 0/0, changed state to up"

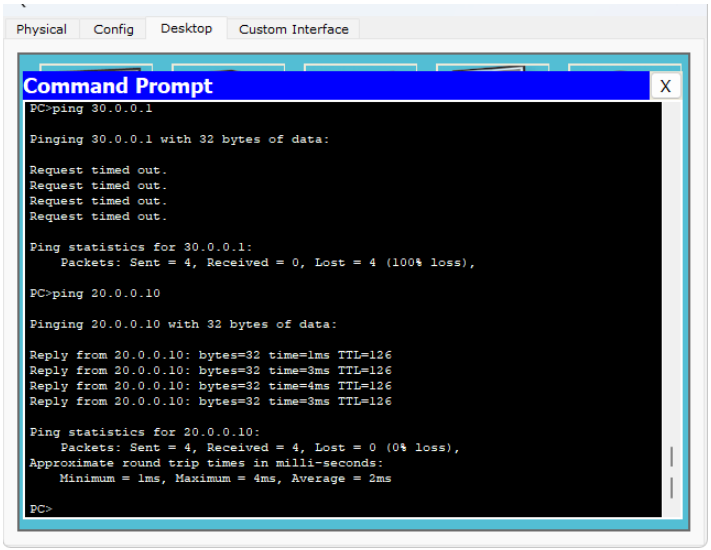
vii. Similarly select routes R2 goto (1) and execute the same
- Router> enable
- Router# config terminal
- Router(config)# interface fastethernet 1/0
- Router(config-if)# ip address 20.0.0.2 255.255.255.0
- Router(config-if)# no shutdown

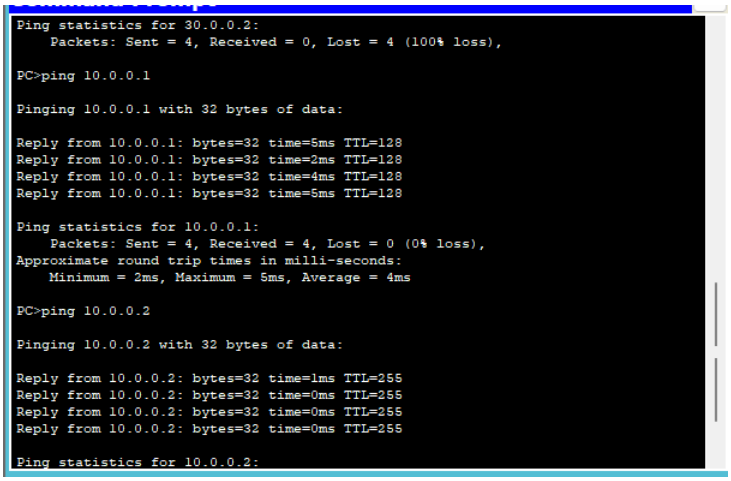
"Interface fastethernet 1/0, changed state to up"

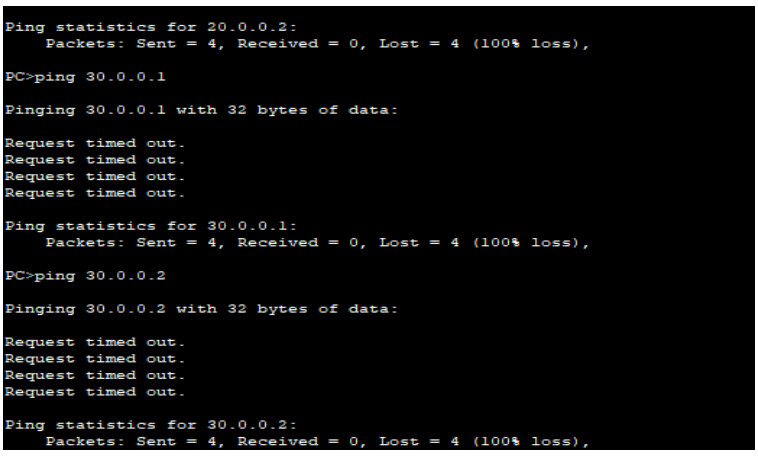
viii. Hence the connection between Router and end devices is established.
ix. Now connect router R1 with router R2 using serial cable (specially connected)
x. To setup connection b/w routers again
- Select router R1 and go to CLI
- Router(config)# interface serial 2/0
- Router(config-if)# ip address 30.0.0.1 255.0.0.0
- Router(config-if)# no shutdown
- Select router R2 and go to CLI
- Router(config)# interface serial 3/0
- Router(config-if)# ip address 30.0.0.2 255.0.0.0
- Router(config-if)# no shutdown

"Interface serial 2/0 changed state to up"

Screen shots/ output :







Observations:

- After setting up the above mentioned topology ,
- Now try to ping PC1 with PC0
- Open command prompt for PC1 and type -> ping 20.0.0.1
- Destination net host unreachable
- Packets sent = 4, received = 0, lost = 4, loss = 100%

It is also observed that the end system PC0 was only pinged with router R0 only

In order to establish static connection ,go to router 0's CLI and type the following commands

- Router> enable
- Router# config terminal
- Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.2

Then go to router 2's CLI and type the following commands

- Router> enable
- Router# config terminal
- Router(config)# ip route 10.0.0.0 255.0.0.0 30.0.0.1

So now check if all the connections are done by going to each router's CLI and typing show ip route.

- After this again go to PC1 and now type in CP
- ping 20.0.0.10
- Packets: sent = 4, Received = 4, Lost = 0

Observation Book:

LAB-03

(5)

AIM:

3. Configure default route, static route, to the Router

Procedure:

1) Select 2 PC's and 2 routers from the tool bar. make sure that each PC connected to the each other.
2) then set the ip address and gateway to both the end devices.
3) Connect router to router using serial connection
4) Assign IP for the each PC & also assign subnet
   PC1 → 10.0.0.1 , 255.0.0.0
   PC2 → 20.0.0.1 , 255.0.0.0
5) Assign gateway for the each PC
   PC1 → 10.0.0.2
   PC2 → 20.0.0.2

In CLI

→ Follow these commands for Router 1

1 enable
2 Config terminal
3 interface fastethernet 0/0
4 ip address 10.0.0.2 255.0.0.0
5 no shut
6 exit

For assign IP for router 1

1. interface serial2/0
2. ip address 30.0.0.1 255.0.0.0
3. no shutdown

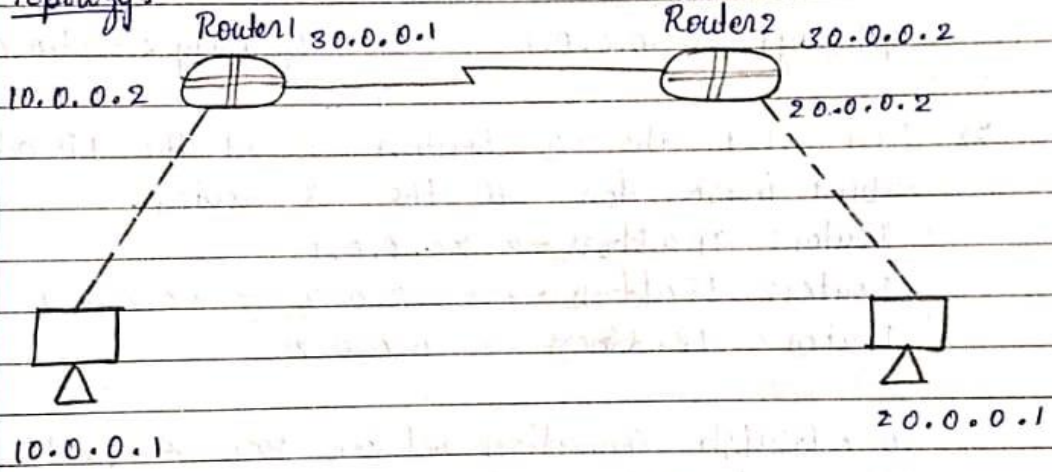
→ Follow these commands for Router 2

1) enable
2) Config terminal

5) interface FastEthernet 1/0
4. ip address 20.0.0.2 255.0.0.0
5. no Shoutdown

For assign IP for router 2 ⇒
1. interface Serial 2/0
2. ip address 30.0.0.2 255.0.0.0
3. no Shutdown

Topology:



Observation:
If we are establish connection from one PC from a networks to another PC over another network. we get the output as Destination host unreachable. & it only establish connection within a networks

To overcome this proble we use the Static route. Now the connection will establish between PC0 & PC1 over a different networks

Program 4

Aim of the program:
Configure default route, static route to the Router.(Part 2)

Procedure along with the topology:

i. Select a generic router R0
ii. Connect an end device PC0 to router R0 through parallel connection fast-ethernet 0/0
iii. Configure PC0 with ip address 10.0.0.10 and gateway 10.0.0.1
iv. Select another router R1
v. There is a serial connection from router R0 to R1 i.e Se2/0
vi. Select another router R2 and there is a serial connection from router R1 to R2 i.e Se3/0
vii. From Router R2 there connect a PC1 through parallel connection fa0/0
viii. Configure PC1 with ip address 40.0.0.10 and gateway 40.0.0.1
ix. Now go to Router 0 and CLI and type

- Router> enable
- Router# config terminal
- Router(config)# interface fastethernet0/0
- Router(config-if)# ip address 10.0.0.1 255.0.0.0
- Router(config-if)# no shutdown

"Interface FastEthernet 0/0 changed state to up"

x. Similarly go to Router 1 and CLI and type

- Router> enable
- Router# config terminal
- Router(config)# interface serial2/0
- Router(config-if)# ip address 20.0.0.2 255.0.0.0
- Router(config-if)# no shutdown

"Interface Serial 2/0 changed state to up"

xi. So now we have to do for serial3/0

- Router> enable
- Router# config terminal
- Router(config)# interface serial3/0
- Router(config-if)# ip address 30.0.0.1 255.0.0.0

- Router(config-if)# no shutdown

"Interface Serial 3/0 changed state to up"

xii. Now go to Router 2 and CLI and type

- Router> enable
- Router# config terminal
- Router(config)# interface fastethernet0/0
- Router(config-if)# ip address 40.0.0.1 255.0.0.0
- Router(config-if)# no shutdown

"Interface FastEthernet 0/0 changed state to up"

xiii. So now we will do for serial 3/0

- Router(config)# interface serial3/0
- Router(config-if)# ip address 30.0.0.2 255.0.0.0
- Router(config-if)# no shutdown

xiv. Now we will set static Route

xv. So for that we will go to Router 1 CLI and type

- Router# show ip route
- C 20.0.0.0/8 is directly connected, Serial2/0
- C 30.0.0.0/8 is directly connected, Serial3/0

xvi. So now again type

- Router(config)# terminal
- Router(config)# ip address 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
- Router(config)# exit
- Router# show ip route
- S 10.0.0.0/8 via 20.0.0.1
- C 20.0.0.0/8 is directly connected, Serial2/0
- C 30.0.0.0/8 is directly connected, Serial3/0
- S 40.0.0.0/8 via 30.0.0.1

xvii. Now we have to set default Route

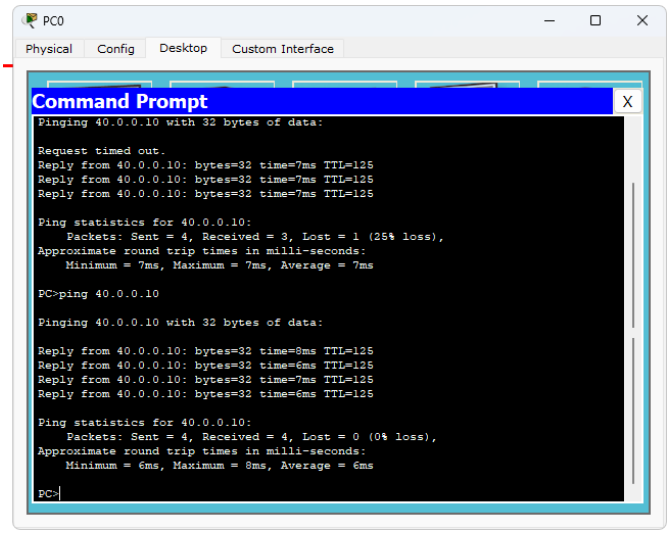
xviii. So we will go to Router 0

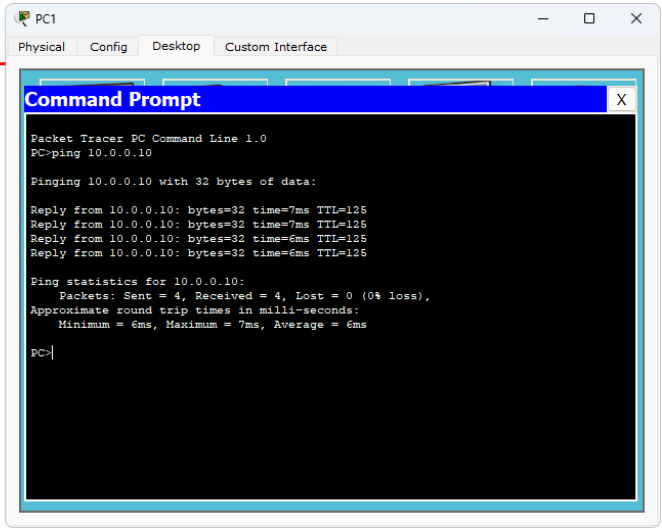
- Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

xix. Then Router 2

- Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1"

Screen shots/ output :





Observation:

So now we will go to PC0 and in the command prompt type:

- PC> ping 40.0.0.10
- Pinging 40.0.0.10 with 32 bytes of data
- Ping statistics for 40.0.0.10
- Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

Now to ping PC1 we have to go to command prompt and type:

- PC> ping 10.0.0.10
- Pinging 10.0.0.10 with 32 bytes Packets: Sent = 4, Received = 4, Lost = 0

Observation Book:

LAB - 04

23/10/21

(7)

AIM:
Configure default route, static route to the Router

Procedure:

1) Select Two PC's & Set the IP address for both the PC's & also Set the Gateway to that PC's.
   PC1 IP address → 10.0.0.10
   PC2 IP address → 40.0.0.10
   Gateway 1 → 10.0.0.1 Gateway 2 → 40.0.0.1

2) Then Select the 3 Routers & Set the IP address & Subnet mask for all the 3 routers.
   Router 1 IP address → 20.0.0.1
   Router 2 IP address → 20.0.0.2 & 30.0.0.1
   Router 3 IP address → 30.0.0.2

To establish connection between PC0 & router 0 Follow these Commands in router 0 CLI →

1. enable
2. Config terminal
3. interface FastEthernet 0/0
4. ip address 10.0.0.1 255.0.0.0
5. no Shutdown

To establish Connection between PC0 & router 1 Follow these Commands in router 1 CLI →

1) enable
2) Config terminal
3) interface Serial 3/0
4) ip address 20.0.0.2 255.0.0.0

5) no shut
6) exit

To establish connection between router0 & router1 follow these commands in the router0 CLI →

1) enable.
2) Config terminal
3) interface Serial 3/0
4) ip address 20.0.0.1 255.0.0.0
5) no shut
6) exit

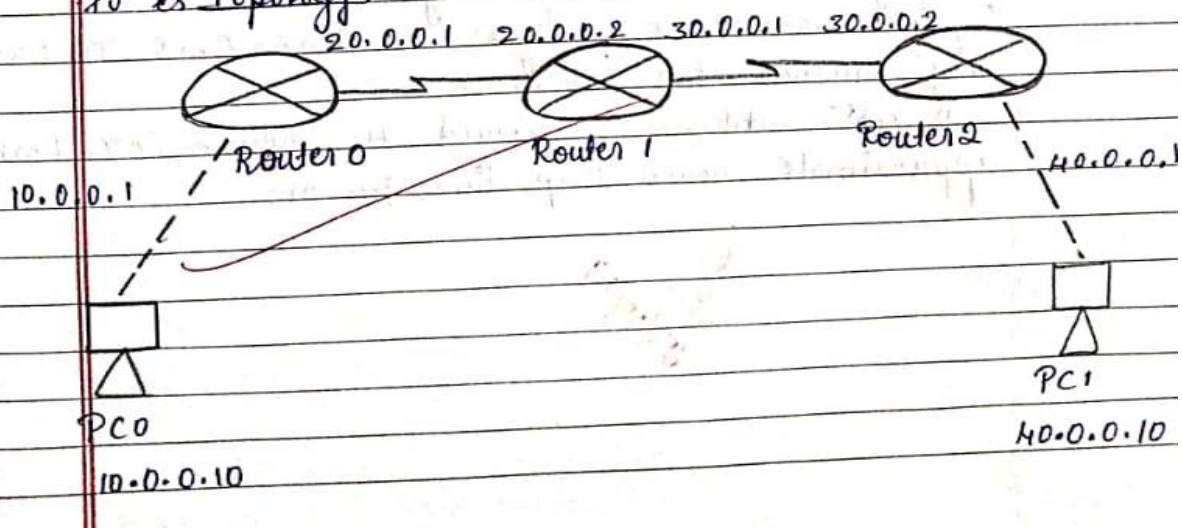
* To establish connection between router0 & router1, router2.
follow these Command in router1

1) ip route 10.0.0.0 255.0.0.0 20.0.0.1
2) ip route 40.0.0.0 255.0.0.0 30.0.0.2

To establish Connection between PC0 & router1 follow these commands in router0 & router1

ip route 0.0.0.0 0.0.0.0 20.0.0.2
ip route 0.0.0.0 0.0.0.0 30.0.0.1

To es Topology:



Observation: we will get to know how to connect 3 Routers, & how to connect the Static route.

→ Connection establishes from router0.
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

→ Connection establishes from router1
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial 3/0
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

→ Connection establish from router 2
C 30.0.0.0/8 is directly connected, Serial 3/0
C 40.0.0.0/8 is directly connected, FastEthernet 1/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

? output for the Ping Command:-
→ 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 = 255
ping statistics for 20.0.0.1
packets: Sent = 4, Recieved = 4 host = 0 (0% loss)
approximate round trip times in ms.

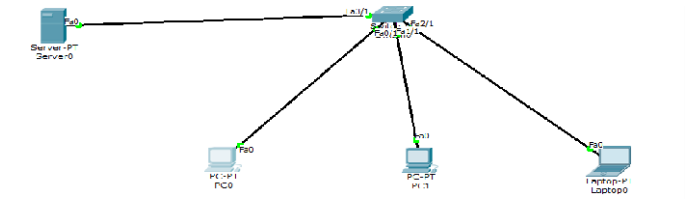
23/10/24

Program 5

Aim of the program:
Configure DHCP within a LAN and outside LAN

a) Within LAN

Procedure along with the topology:



i. Select generic server-PT
ii. Select generic PC0, PC1, laptop-PT laptop0, Switch, laptop-1, laptop-2, laptop-3 and a router-0.
iii. All are connected used Automatically choose Connection Type.
iv. Click on server-PT and give the
v. IP address: 10.0.0.1
vi. Subnet Mask: 255.0.0.0
vii. Default Gateway: 10.0.0.10
viii. Select DHCP and set service as on. Set the pool Name as switch one, default gateway as 10.0.0.1, maximum number of users as 100 and start IP address as 10.0.0.3. Click Add.
ix. Click on each PC, navigate to IP configuration and select DHCP.
x. Ping from one device to another

Screen shots/ output :

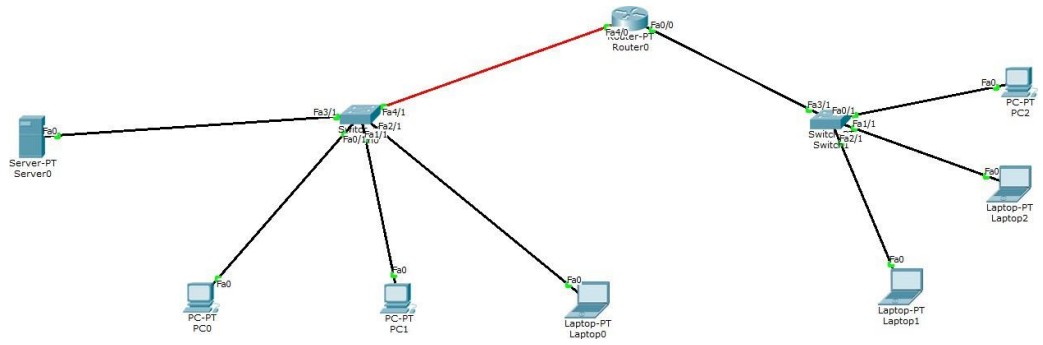
```
PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=2ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Observation:

Pinging 10.0.0.4 with 32 bytes of data
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Packets: Sent = 4, Received = 4, Loss = 0 (0% loss)

b) Outside LAN

Procedure along with the topology:



i. Select server, set IP Address as 10.0.0.2, default gateway as 10.0.0.0, DNS server as 0.0.0.0
ii. Navigate into config, select DHCP, modify default gateway as 10.0.0.1, start IP address as 10.0.0.0 (switch 1)
iii. Name a new pool as switch two, set start IP as 20.0.0.3, default gateway as 20.0.0.1, maximum number of users as 100, set gateway in setting as 10.10.0.1
iv. Select router, navigate to CLI and enter the following commands:
    - enable
    - config terminal
    - interface fastethernet 0/0
    - ip address 10.0.0.1 255.0.0.0
    - ip helper address 10.0.0.2
    - no shutdown
    - exit
v. Repeat the same

Screen shots/ output :

```
Packet Tracer PC Command Line 1.0
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=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

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=2ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
```

Observation:
The ip address is set for all end devices.

Observation Book:

LAb-05

Bafna Gold

Date: Page:

10

Configure DHCP

Q. Configure DHCP within LAN & outside LAN.

Aim: The aim of this experiment is to automate the assignment of IP addresses to devices on the network. DHCP allows devices to join the network & receive a unique IP address & other necessary configuration.

Procedure:

1) Select 6 PC's & Select 1 Server & 1 router, drag & drop all the de network devices.

2) Connect the network devices using cables.
for Router → Fiber Cable & Copper Straight through
for PC's → Copper Straight through
for Server → Copper Straight through

3) Next for one LAN Set the DHCP (Dynamic host Configuration protocol) IP address & do the same thing to another LAN also.

4) In Server → Select Services → then select DHCP → Create 2 Pools.

Pool 1 : Switch 01	Pool 2: Switch 2
Pool name: Switch1	pool name: Switch 2
Default gateway: 10.0.0.1	Default gateway: 20.0.0.1
DNS Server: 10.0.0.0	DNS Server: 20.0.0.0
IP address: 10.0.0.3	IP address: 20.0.0.3
Subnet mask: 255.0.0.0	Subnet mask 255.0.0.0
max users: 100	max User: 100

5) In Router part, Click on Router → CLI → Run the below Commands

→

⇒ LAN 01

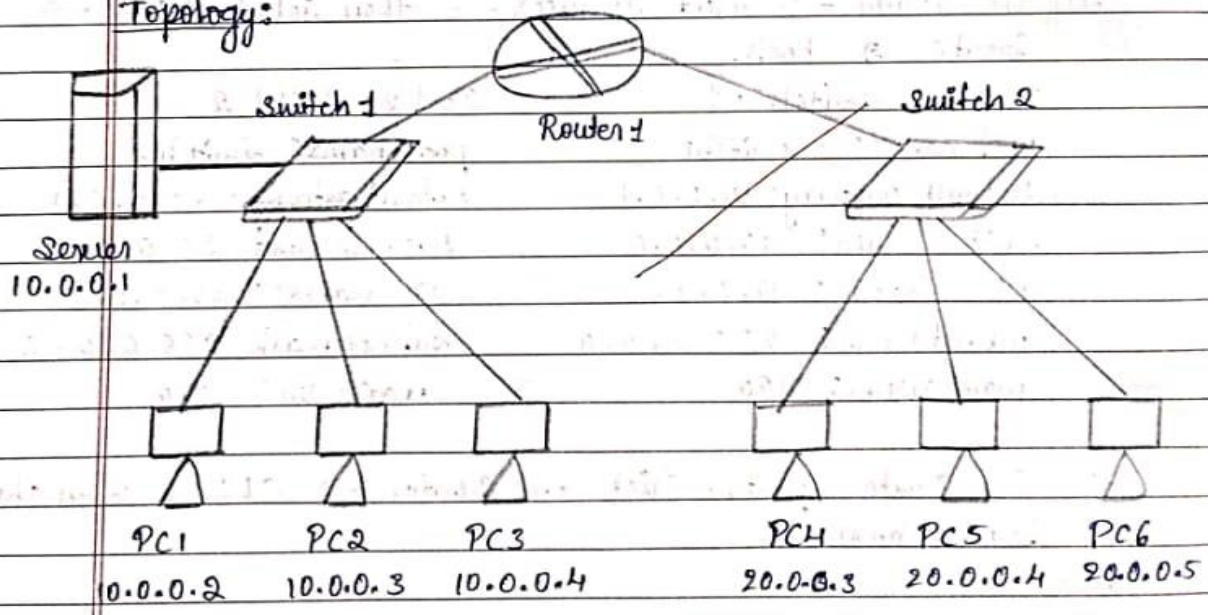
```
enable
config terminal
interface FastEthernet 0/0
ip address 20.0.0.1 255.0.0.0
ip helper-address 20.0.0.2
no shutdown
exit
```

⇒ LAN 02

```
enable
Config terminal
interface FastEthernet 4/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shutdown
exit
```

Now all connections are over now both the LAN's are connected each other

Topology:



Bafna Gold
Date: Page:

Observation: DHCP automates the assignment of IP address & other network configuration details to devices on a network. It works by leasing IP addresses for a specific duration & can provide additional settings like DNS & gateway information.

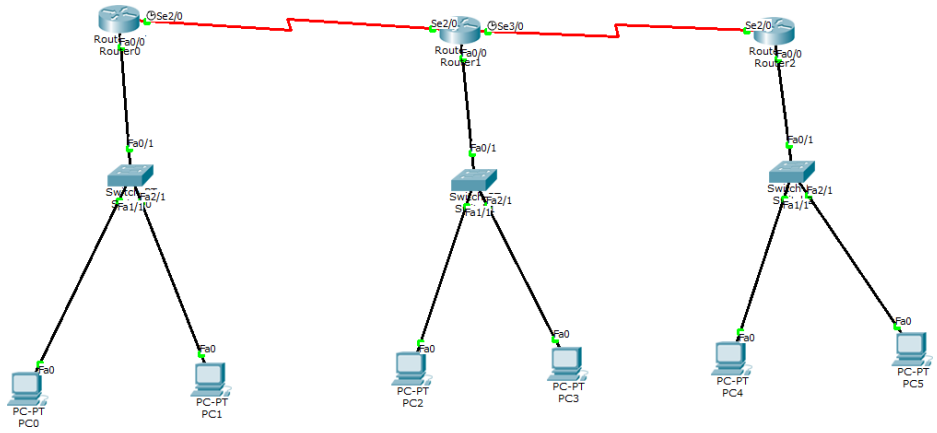
Output:

Packet 4 : Sent = 4 Received = 4 Lost = 0 (0% Loss),
approximate round trip times in ms.

Program 6

Aim of the program:
Configure RIP routing Protocol in Routers

Procedure along with the topology:



i. Configure the routers, switches and generic PCs
ii. For the PCs, navigate to config interface and fastethernet 0 and set the IP address for all six PCs
iii. Similarly for all six PCs, navigate to config, global settings and set gateway as the IP address of the router it is connected to
iv. In router 0, navigate to config, fast ethernet 0/0 and add IP address (10.0.0.1)
v. Then go to serial 3/0 and update IP address as 40.0.0.1
vi. Go to CLI and enter command no shut
vii. In Router 1, Go to
- Interface fast ethernet 0/0, set Ip address as 20.0.0.1.
- Go to serial 2/0 and set IP address to 40.0.0.2
- In CLI enter no shut.
- Go to serial 3/0 and set Ip address as 50.0.0.1
- In CLI enter no shut.

viii. In Router 2 Go to Interface
- fast ethernet 0/0, set IP address as 30.0.0.1
- Go to serial 2/0 and set IP address as 50.0.0.2

ix. If connections are not active after this, in all routers check the fast ethernet 0/0 and enter no shut in the CLI again. Now all the connections are active (green).

x. So now go to router 0 and in CLI type

- Router(config)# router rip
- Router(config-route)# network 10.0.0.0
- Router(config-route)# network 40.0.0.0
- Router(config-route)# exit
- Router(config)# exit
- Router# show ip route

  C ……..

  R ……..

  C …….

xi. Now go to router 1 and in CLI type

- Router(config)# router rip
- Router(config-route)# network 40.0.0.0
- Router(config-route)# network 50.0.0.0
- Router(config-route)# network 20.0.0.0

xii. Router# show ip route

R

C . . .

C

C . . .

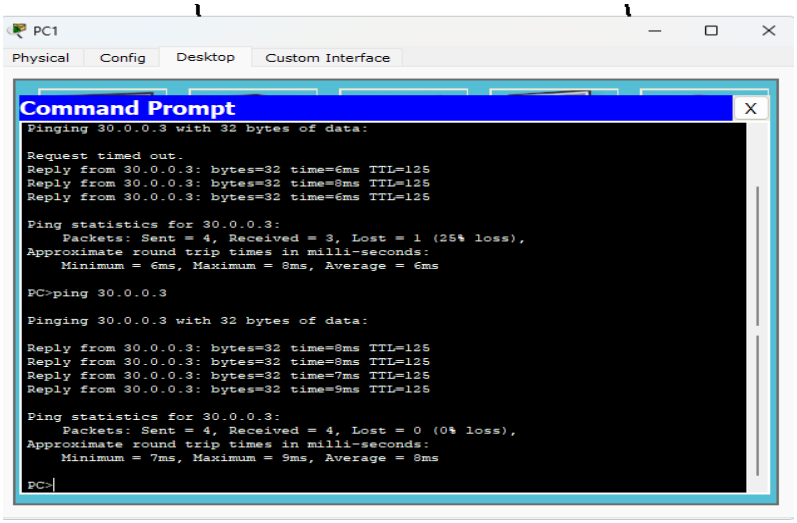
xiii. Now go to router 2 and type

- Router(config)# router rip
- Router(config-router)# network 50.0.0.0
- Router(config-router)# network 30.0.0.0
- Router# show ip route

  R . . .

  C

Screen shots/ output :



Observation:

From PC1 we can ping to PC5

PC1> ping 30.0.0.3

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

Observation Book:

LAB - 06

AIM: Configure routing information protocol in Router

Procedure:

1) Select the 3 Routers & drag on the Screen. Then Select 3 Switches & drag on the Screen then Connect the Same end devices to that Switches using Copper Straight through Cable.
2) Then Connect all the Switches to 3 different routers using Copper-Straight-through Cable. & also Connect the router to router using Serial DCE Cable.
   Router to Router → Serial Connection.
   Switch to Router → FastEthernet Connection.
3) Next Set the IP address to all the end devices with gateway.

PC0 → 10.0.0.2	PC1 → 10.0.0.3
PC2 → 20.0.0.2	PC3 → 20.0.0.3
PC4 → 30.0.0.2	PC5 → 30.0.0.3

4) Go to router 0, To Establish the Connection between Switch 0 & router follow these Commands.
   1) enable
   2) Config terminal
   3) interface FastEthernet 0/0
   4) ip address 10.0.0.1 255.0.0.0
   5) no Shut
   6) exit

To establish the Connection between router0 to router1

1) enable
2) Config terminal
3) interface Serial 2/0
4) ip address 40.0.0.1 255.0.0.0
5) ~~no Shut~~
5) no Shut
6) exit

Rafna Gold
Date: Page:

5) Go to router 1, For router 1 to router 0.
interface Serial 2/0
ip address 40.0.0.2 255.0.0.0

For router 1 to router 2
interface Serial 3/0.
ip address 50.0.0.1 255.0.0.0

For router 1 to Switch 1
interface FastEthernet 0/0
ip address 20.0.0.0.1 255.0.0.0.

6) Go to router 2, For Router 2 to router 1
interface serial 3/0
ip address 50.0.0.2 255.0.0.0

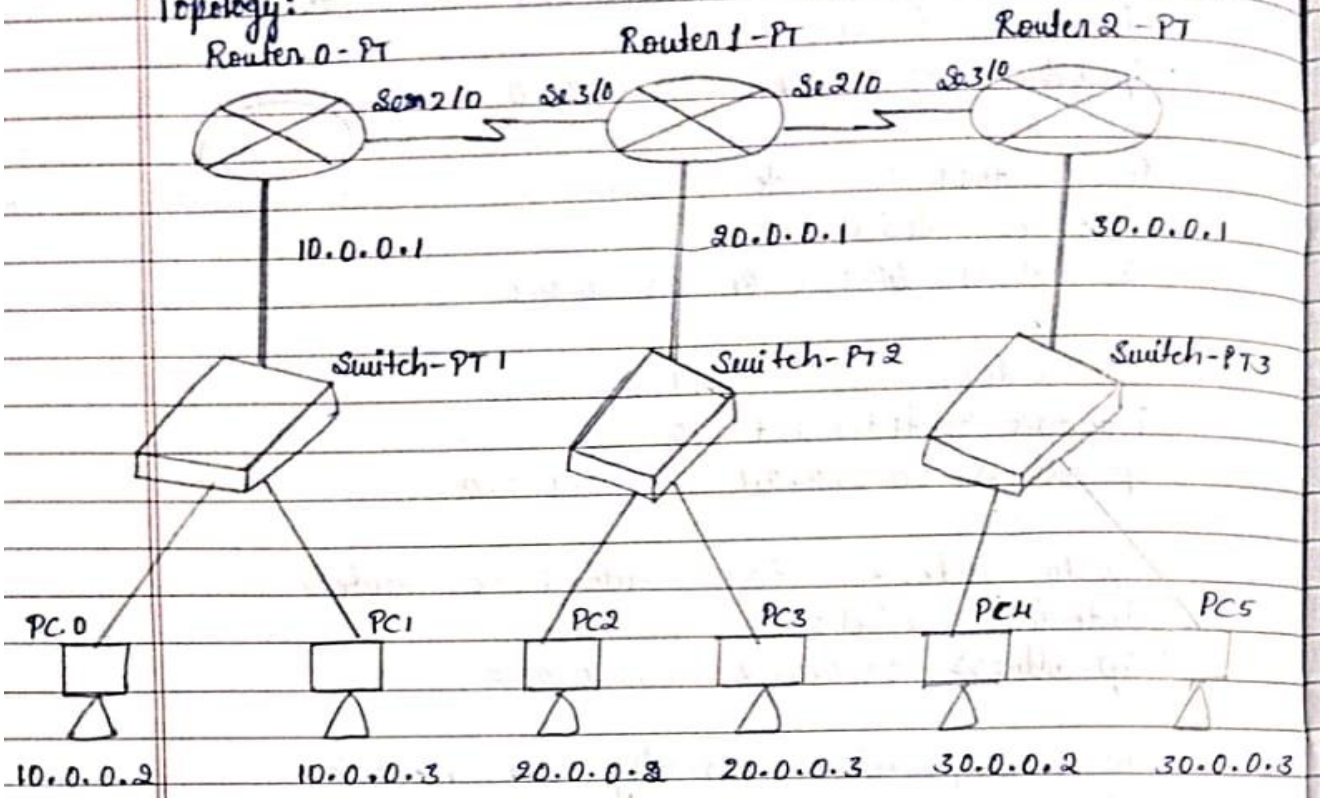
To send packet sucessfully over network
For router 0

1) Config terminal
router rip
network 10.0.0.0
networks 40.0.0.0
Show ip route
LAN1

router rip
network 40.0.0.0
network 50.0.0.0
network 20.0.0.0
Show ip router
LAN2

router rip
network
50.0.0.0
network
30.0.0.0
Show ip route
LAN3

Topology:



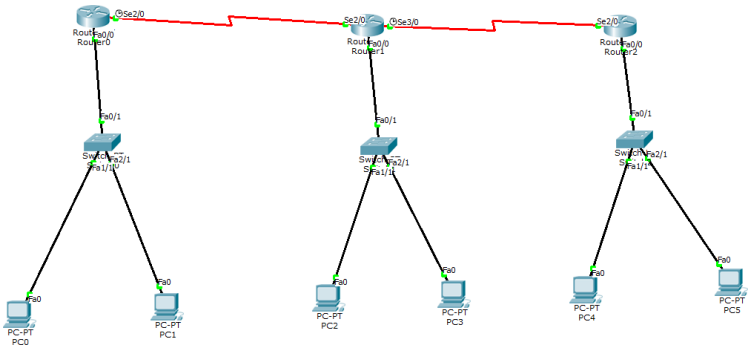
A

Observation: Connection established sucessfully over a network & packets are sent fnorm one to anothes PC over a network.

Program 7

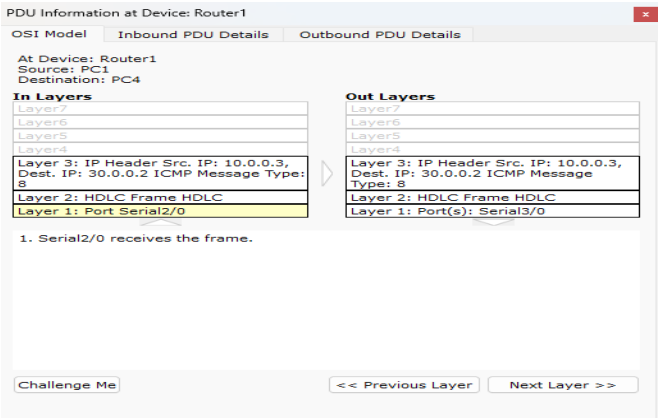
Aim of the program:
Demonstrate the TTL/ Life of a Packet:

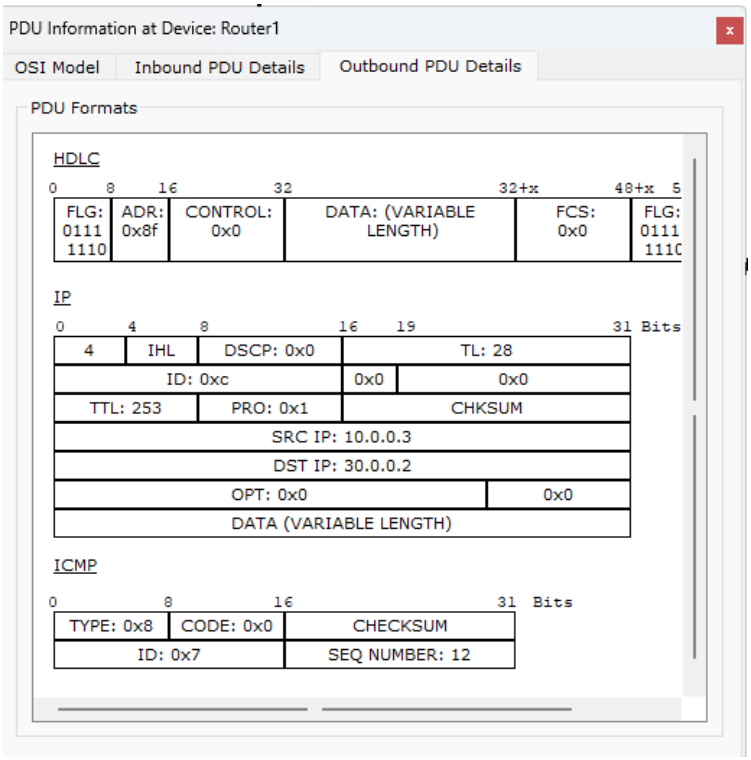
Procedure along with the topology:

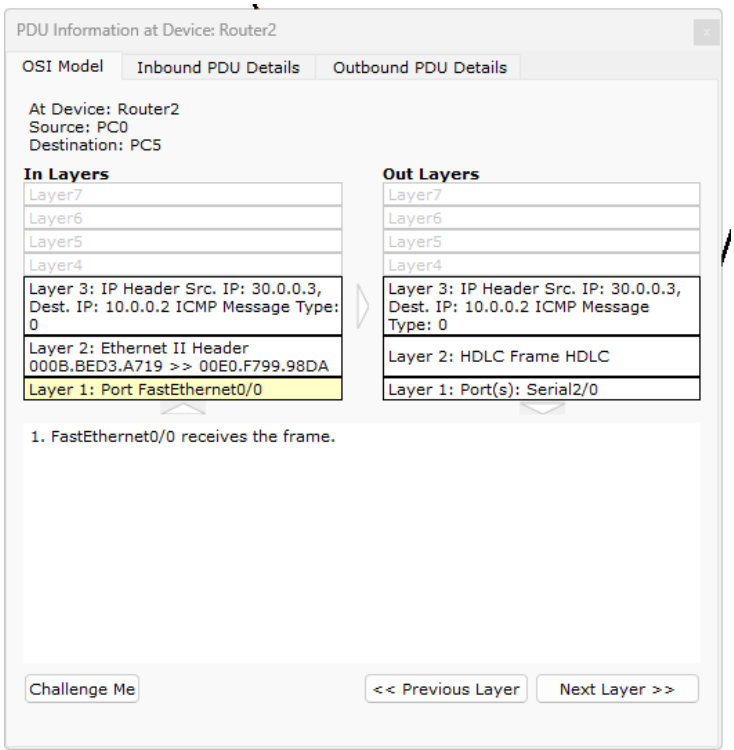


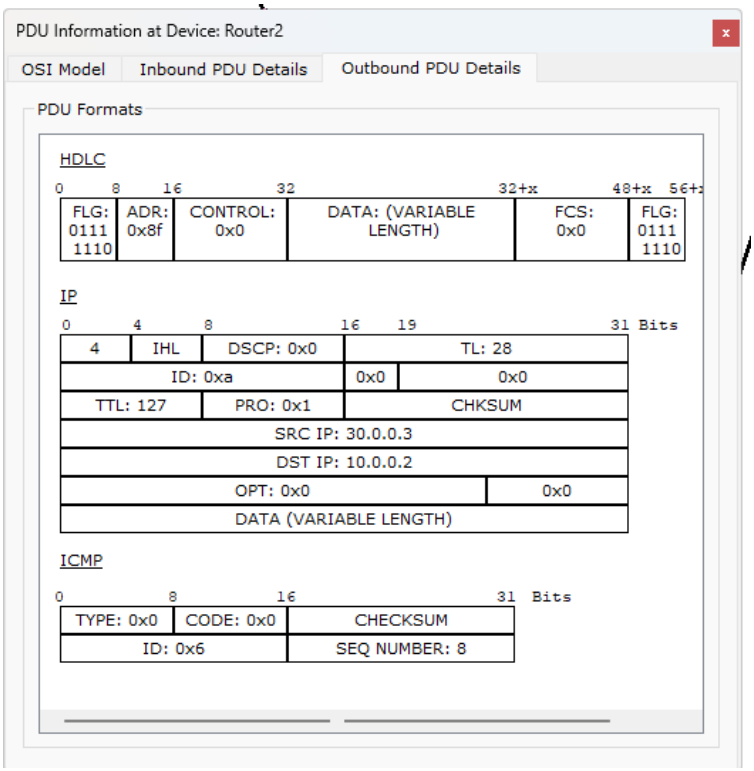
i. Add a simple PDU across the PCs of different network.
ii. Consider PC0 to PC5

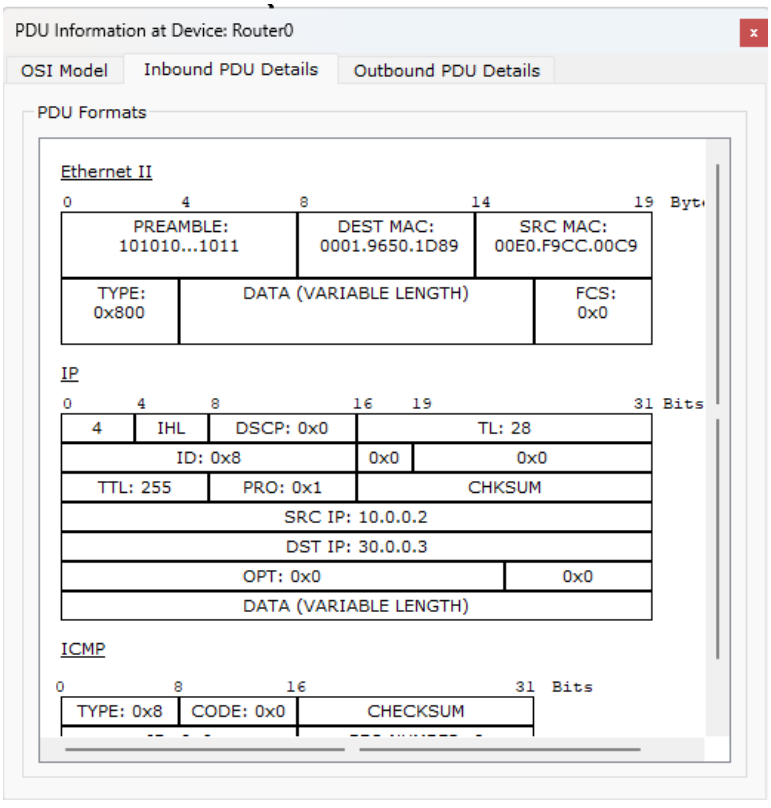
Screen shots/ output :

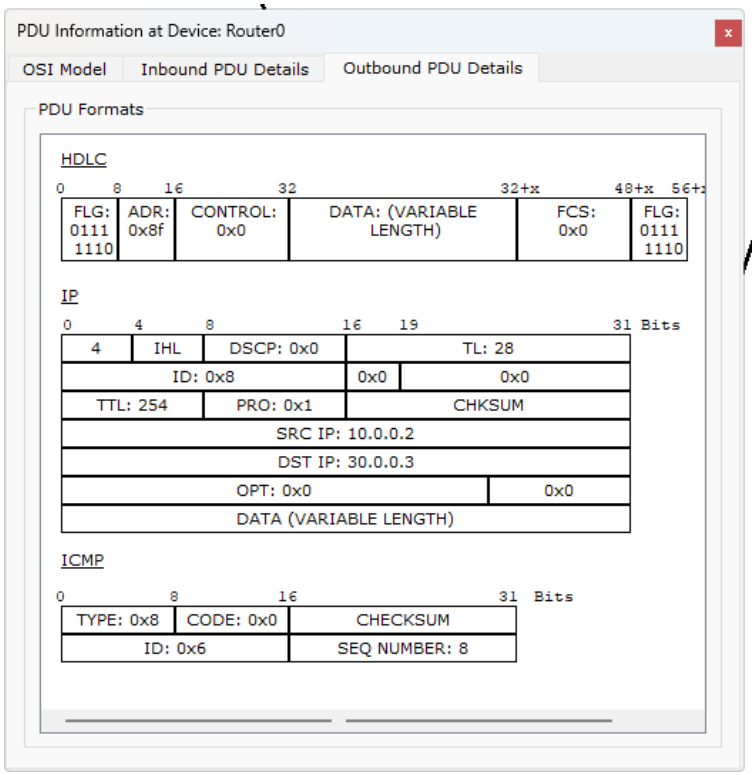



PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 5
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 111C
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xc
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.3
DST IP: 30.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x7
SEQ NUMBER: 12


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Router2
Source: PC0
Destination: PC5
In Layers
Out Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 30.0.0.3, Dest. IP: 10.0.0.2 ICMP Message Type: 0
Layer 2: Ethernet II Header 000B.BED3.A719 >> 00E0.F799.98DA
Layer 1: Port FastEthernet0/0
Layer 3: IP Header Src. IP: 30.0.0.3, Dest. IP: 10.0.0.2 ICMP Message Type: 0
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial2/0
1. FastEthernet0/0 receives the frame.
Challenge Me
<< Previous Layer
Next Layer >>


PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 127
PRO: 0x1
CHKSUM
SRC IP: 30.0.0.3
DST IP: 10.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x6
SEQ NUMBER: 8


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0001.9650.1D89
SRC MAC: 00E0.F9CC.00C9
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x8
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 30.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM



Observations:

- While Auto capture and observing the TTL across each PC, it was observed as follows:
- PDU Information at Device: PC1 TTL: 225
- PDU Information at Device: Router1 TTL: 254
- PDU Information at Device: Router2 TTL: 253
- Cisco packet tracer has the maximum TTL as 225.
- It is observed that the TTL document on the message is being passed stop by stop (router to router).
- The figure of the OSI model of switch demonstrates the flow of packets in 2 layers while 3 layers in the router. The TTL reaches zero once all the packets are received.

Bafna Gold
Date: Page:

Aim: demonstrate TTL / life of a packet.

Steps

1) Select a packet
2) Transfer it from one pc to another over a network i.e. PC0 to PC1
3) while transforming packets pause the packets (Click on Auto / Capture play)
4) Then click on packet then you can able to view inbound & outbound PDU.
5) It will reduces the TTL when we are pause the Auto / Capture play.

Observation: when we are sending the packet over through one LAN to another LAN it will be reduces by 1 in TTL.

20/11

Program 8

Aim of the program:
Configure OSPF routing protocol

Procedure along with the topology:

i. Create topology like below i have given
ii. Configure ip address to all interfaces
iii. In Router RI,
- Rl(config)#interface fastethernet 2/0
- R 1(config-if)#ip address 10.0.0.1 255.0.0.0
- Rl(config-if)#no shutdown
- Rl(config-if)#exit

- Rl(config)#interface serial 1/0
- Rl(config-if)#ip address 20.0.0.1 255.0.0.0
- Rl(config-if)#encapsulation ppp
- Rl(config-if)#clock rate 64000
- Rl(config-if)#no shutdown
- Rl (config-if)#exit

iv. In Router R2
- R2(config)#interface serial 1/0
- R2(config-if)#ip address 20.0.0.2 255.0.0.0
- R2(config-if)#encapsulation ppp
- R2(config-if)#no shutdown
- R2(config-if)#exit

- R2(config)#interface serial 1/1
- R2(config-if)#ip address 30.0.0.1 255.0.0.0
- R2(config-if)#encapsulation ppp

- R2(config-if)#clock rate 64000
- R2(config-if)#no shutdown
- R2(config-if)#exit

v. In Router R3
- R3(config)#interface serial 1/0
- R3(config-if)#ip address 30.0.0.2 255.0.0.0
- R3(config-if)#encapsulation ppp
- R3(config-if)#no shutdown
- R3(config-if)#exit

- R3(config)#interface fastethernet 2/0
- R3(config-if)#ip address 40.0.0.1 255.0.0.0
- R3(config-if)#no shutdown
- R3(config-if)#exit

vi. Configure OSPF

vii. On Router R1:
- R1(config)#router ospf 1
- R1(config-router)#router-id 1.1.1.1
- R1(config-router)#network 10.0.0.0 0.255.255.255 area 3
- R1(config-router)#network 20.0.0.0 0.255.255.255 area 1
- R1(config-router)#exit

viii. On Router R2:
- R2(config)#router ospf 1
- R2(config-router)#router-id 2.2.2.2
- R2(config-router)#network 20.0.0.0 0.255.255.255 area 1
- R2(config-router)#network 30.0.0.0 0.255.255.255 area 0
- R2(config-router)#exit

ix. On Router R3
- R3(config)#router ospf 1
- R3(config-router)#router-id 3.3.3.3
- R3(config-router)#network 30.0.0.0 0.255.255.255 area 0
- R3(config-router)#network 40.0.0.0 0.255.255.255 area 2
- R3(config-router)#exit

x. Configure Loopback Interfaces

xi. On Router R1:
- R1(config)#interface loopback 0
- R1(config-if)#ip address 172.16.1.252 255.255.0.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

xii. On Router R2:
- R2(config)#interface loopback 0
- R2(config-if)#ip address 172.16.1.253 255.255.0.0
- R2(config-if)#no shutdown
- R2(config-if)#exit

xiii. On Router R3:
- R3(config)#interface loopback 0
- R3(config-if)#ip address 172.16.1.254 255.255.0.0
- R3(config-if)#no shutdown
- R3(config-if)#exit

xiv. Configure Virtual Link

xv. On Router R1:
- R1(config)#router ospf 1
- R1(config-router)#area 1 virtual-link 2.2.2.2
- R1(config-router)#exit

xvi. On Router R2:
- R2(config)#router ospf 1
- R2(config-router)#area 1 virtual-link 1.1.1.1
- R2(config-router)#exit

xvii. Verify Routing Table

xviii. On Router R3

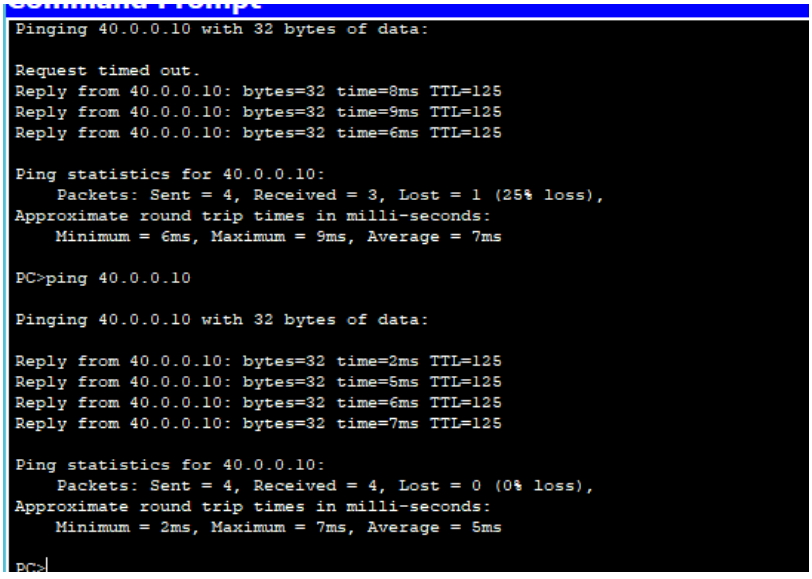
xix. R3#show ip route

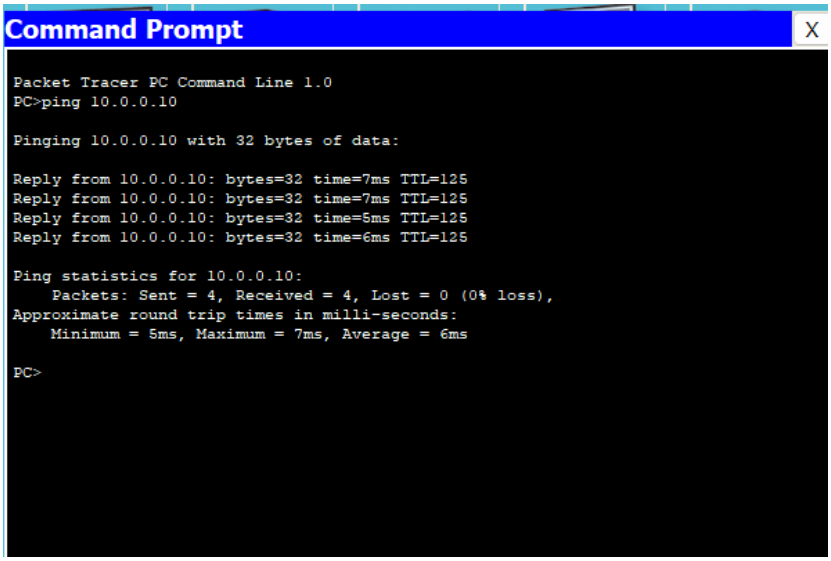
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
Gateway of last resort is not set

O 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:56, Serial1/0
C 40.0.0.0/8 is directly connected, FastEthernet2/0
O 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:56, Serial1/0
C 30.0.0.0/8 is directly connected, Serial1/0

xx. Check connectivity between host l 0.0.0. l 0 to 40.0.0.10

Screen shots/ output :





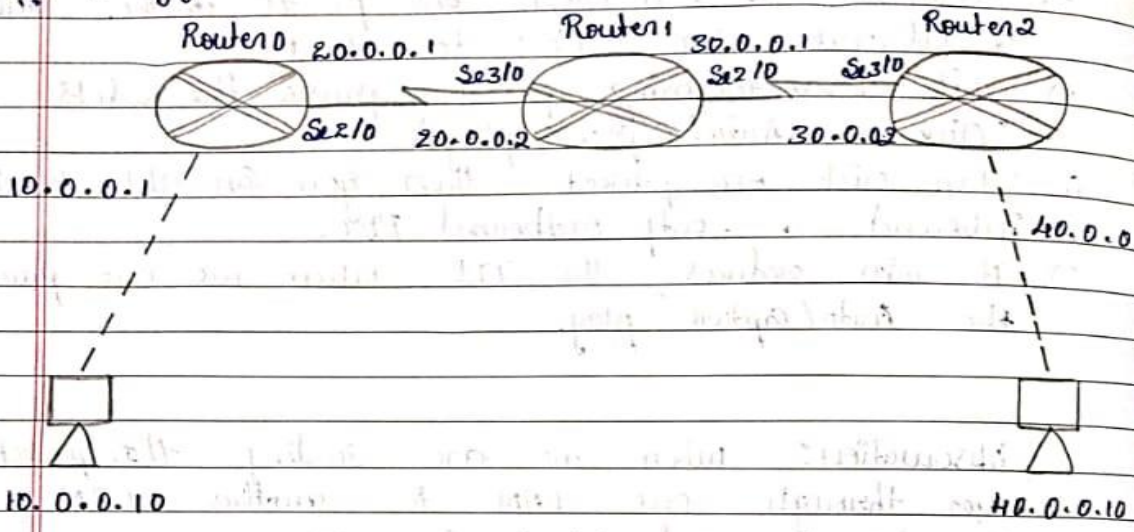
Observations:

- Check connectivity between host 10.0.0.10 to 40.0.0.10
- PC> ping 40.0.0.10 (In CP of PC0)
- Packets: sent = 4, received = 4, lost = 0 (0% loss)
- PC> ping 10.0.0.10 (In CP of PC4)
- Packets: sent = 4, received = 4, lost = 0 (0% loss)

LAB-07

AIM: How to configure OSPF Routing Protocol & Connect Areas

Topology:



Step1: Create a topology like above I have given
Step2: Configure ip address to all interfaces.

In Router R1
interface fastethernet 2/0
ip address 10.0.0.1 255.0.0.0
no shutdown
interface Serial 1/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shutdown

In Router R2
interface Serial 1/0
ip address 20.0.0.2 255.0.0.0

encapsulation ppp
no shutdown
interface Serial 1/1
ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shutdown

In Router R3
Interface Serial 1/0
ip address 30.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
interface fastethernet 2/0
ip address 40.0.0.1 255.0.0.0
no shutdown

Step3 : Now, Enable ip routing by ~~conflicting~~ Configuring ospf routing protocol in all routers

In Router 1
router ospf 1
router id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 3
network 20.0.0.0 0.255.255.255 area 1

~~In~~ In Router 2
router ospf 1
router-id 2.2.2.2
network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0

In Router R3
router ospf 1
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2

Step 4: Now check routing table of R1
In Router1
→ show ip route

Step 5: Now check Routing table of R3.
→ show ip route.

Step 6: Create virtual link between R1, R2 by this we create a virtual link to connect area 3 to area 0.
In Router 1
router ospf 1
area 1 virtual-link 2.2.2.2

In Router 2
router ospf 1
area 1 virtual-link 1.1.1.1

Step 7: R2 & R3 get updates about Area 3 Now, check routing table of R3.
In Router R3
→ show ip route.

Step 8: Check Connectivity between host 10.0.0.10 to 40.0.0.10.

Bafna Gold
Date: Page:

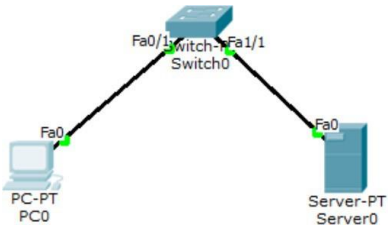
Observation: observation about Configuring the OSPF routing protocol & connecting areas typically focus on its hierarchical structure, performance & scalability.

02/12/24

Program 9

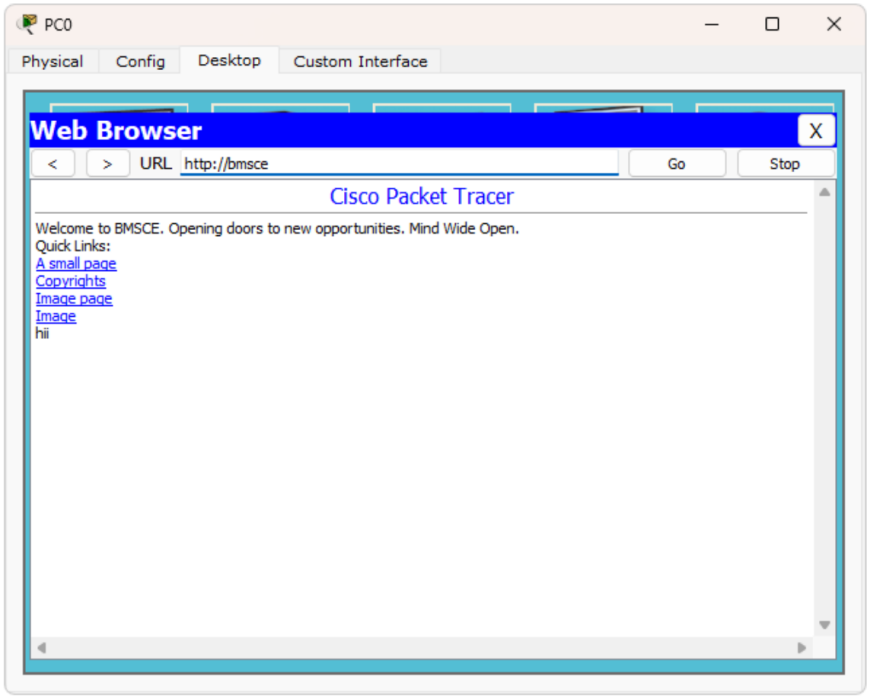
Aim of the program:
Configure Web Server, DNS within a LAN.

Procedure along with the topology:



i. Create the above topology using PC, server, and switch.
ii. Set the IP address of PC normally.
iii. To set the IP of the server, go to config then select static method, set IP, and make sure port status is on.
iv. Ping server from PC (ping is successful, 0% loss).
v. Now go to PC -> Desktop -> web server.
vi. In the virtual browser type in the ip address of server and click.
vii. Now web page is visible.

Screen shots/ output :



Observations:

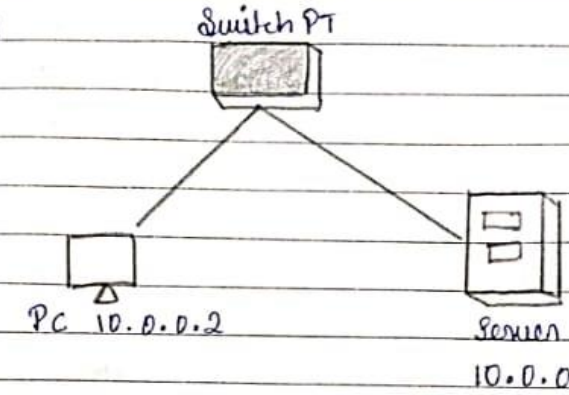
- Successfully accessed the server's web page from PC (web server) by entering ip address.
- Created simple LAN with PC, switch and server.

Bafna Gold
Date: Page:

To construct Configure web server, DNS within LAN.

AIM: To Configure DNS Server

Topology:



Procedure:

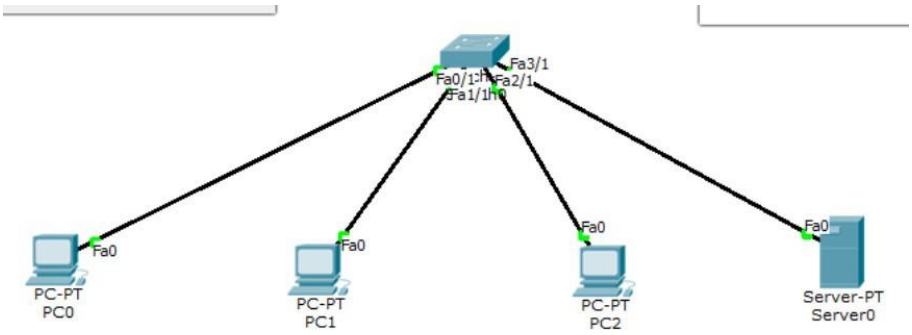
* Select the PC, Switch & Server & Connect them as shown in the topology.
* Assign IP address to PC and Server.
  Note: No IP address is assigned to switch
* Go to DNS Services in the Server & turn it on write name & address & press add.
* In Services select HTTP change on edit the file.
* After editing the file, Select the PC & go to desktop in the PC. In the desktop go to web browser & there write the domain name to get result.

Observation:

* Connections is Success-fully established.
* The domain name System given IP address with domain name.
* When entered domain name the Counter of the specific IP address comes from Server.

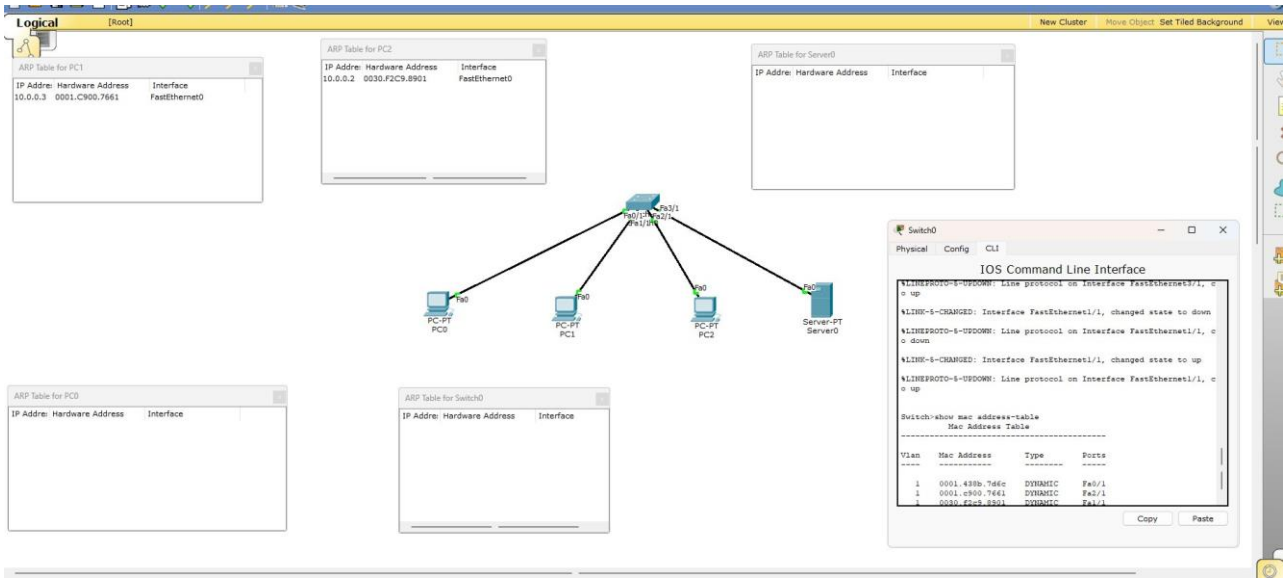
Program 10

Aim of the program:
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
Procedure along with the topology:



i. Create a topology of 4 PCs and a server and switch.
ii. Assign IP to all.
iii. Connect them through a switch.
iv. Use the inspect tool to click on a PC to see the ARP table (command in CLI for same is arp -a).
v. Initially ARP table is empty (Arp in CLI of both, this command show mac address table can be given on every transaction to see how the switch learns from transaction and build the address table)
vi. Use the capture in the simulation panel to go step by step so that changes in ARP can be clearly noted

Screen shots/ output :



Observations:

Switch as well the nodes update the ARP table as and when new communication starts.

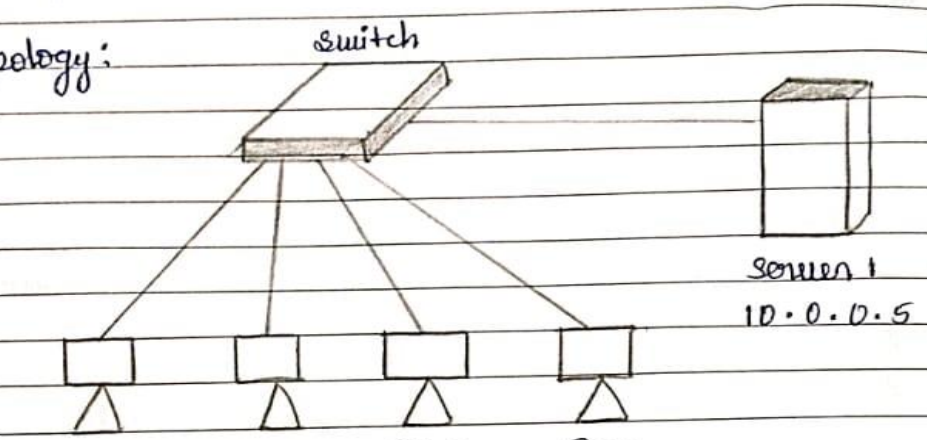
LAB-08

18/12/2024

1) To Construct the Simple LAN & understand the concept and operation of Address Resolution Protocol.

AIM: Construct the Simple LAN & understand the Concept & operation of Address Resolution Protocol.

Topology:



ARP tables:

ARP table for PC0

IP Address	Hardware Address	Interface
10.0.0.4	0009.7CB2A2....	FastEthernet 0

ARP table for PC1

IP address	Hardware Address	Interface
10.0.0.1	00E0.8FBC.76...	FastEthernet 0
10.0.0.5	0000.0CAB.B0--	FastEthernet 0

ARP table for PC2

IP address	Hardware Address	Interface
10.0.0.4	0009.7CB2.A25A	FastEthernet0

Bafna Gold
Date: Page:

7) 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.

8) hook into the interfaces of the switches with the 2 NEWVLAN systems.
This makes the switch understand NEWVLAN.
Next the router is to understand the NEWVLAN

9) Config tab of router select VLAN DATABASE enter the number & name of the Vlan created.

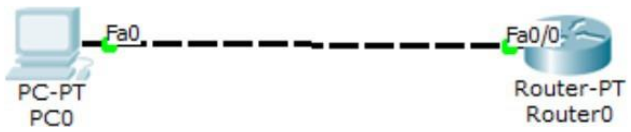
Go to CLI, & Run the following
→ Config t
→ interface fastEthernet 0/0.1
→ encapsulation dot1q 2
→ ip address 192.168.2.1 255.255.255.0
→ no shut
→ exit

Observation:
1) VLANs are used to logically segment a network, devices in the same VLAN to communicate directly while isolating traffic from other VCAN'S.
2) Switches must be configured to create VLAN'S, assign VCAN IDS, & allocate ports to specific VLAN'S.

Program 11

Aim of the program:
To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Procedure along with the topology:

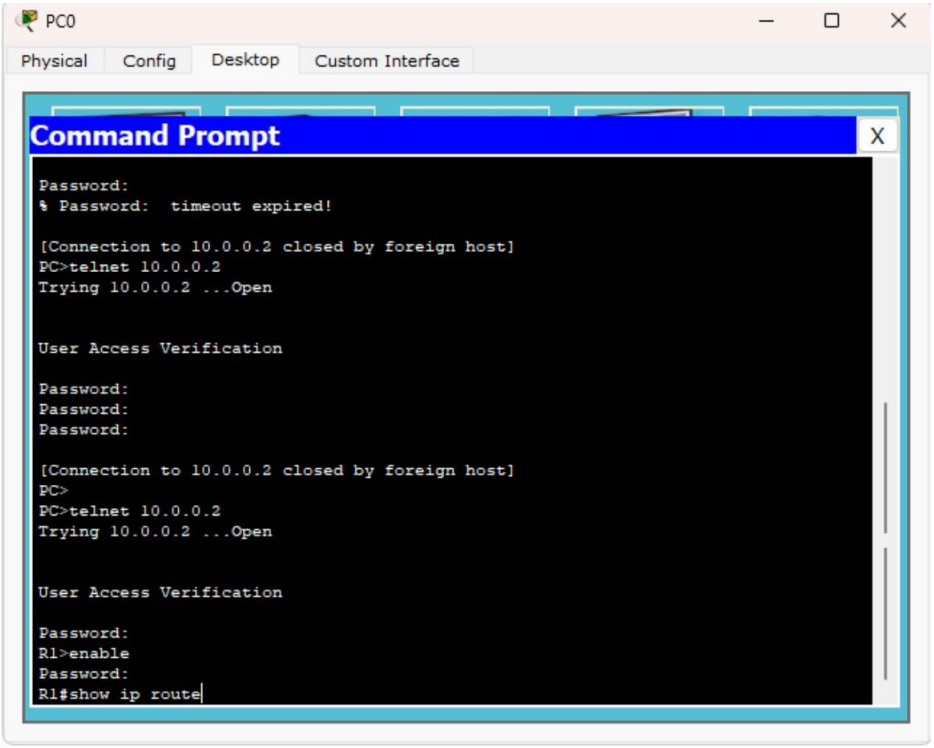


i. Make simple topology shown above.
ii. Commands in router
   - enable # config terminal # hostname
   - R1 # enable secret Pl
   - # interface fastethernet 0/0
   - # ip address 10.0.0.1 255.0.0.0
   - # no shut
   - # line vty 0 5
   - # login
   - #password PO
   - #exit
   - >exit
   - wr --> to save changes

iii. commands in pc
   - ping 10.0.0.1
   - ping results:- (0% loss)

iv. (password for user access verification is PO) ,(password for enable is P1)
v. Accessing router CLI from PC
   - user access verification
   - password:
   - r1> enable
   - password:
   - r1#show ip route

Screen shots/ output :



Observations:

The admin in PC is able to run commands as run in router CLI and see the result from PC.

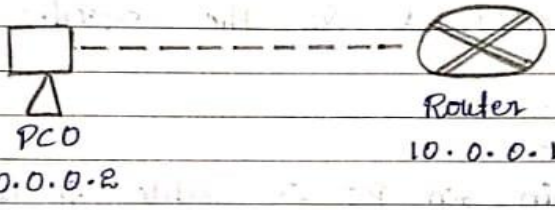
Observation Book:

Bafna Gold

4) AIM: To understand the operation of TELNET by accessing the routers in Servers room from a PC in IT office.

It provides the CLI for the communication with a remote device or server. Sometimes employed for remote management but also for initial device setups like network hardware. Telnet stands for Teletype Network but it can also be used as a verb. to telnet is to establish a connection using the telnet protocol.

Topology:



Steps are as follows:

Commands in Router

→ enable
→ config t
→ hostname R1
→ enable secret p1
→ interface fastethernet 0/0
→ ip address 10.0.0.1 255.0.0.0
→ no shut
→ line vty 0 5 --to allow virtual terminal access for 6 users.
→ login
→ password p0
→ exit
→ exit

→ wr - to save changes in router

Commands in PC
In Command prompt
Ping 10.0.0.1
Ping results seen
Password for user Access verification is p0

password for enable is p1
Accessing routing CLI from PC
The admin in PC is able to run Commands as run in router CLI & see the results from PC.

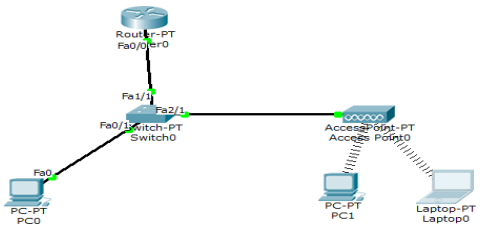
Observation:
The admin in PC is able to run Commands as run in router CLI & see the result from PC.

31/12/24

Program 12

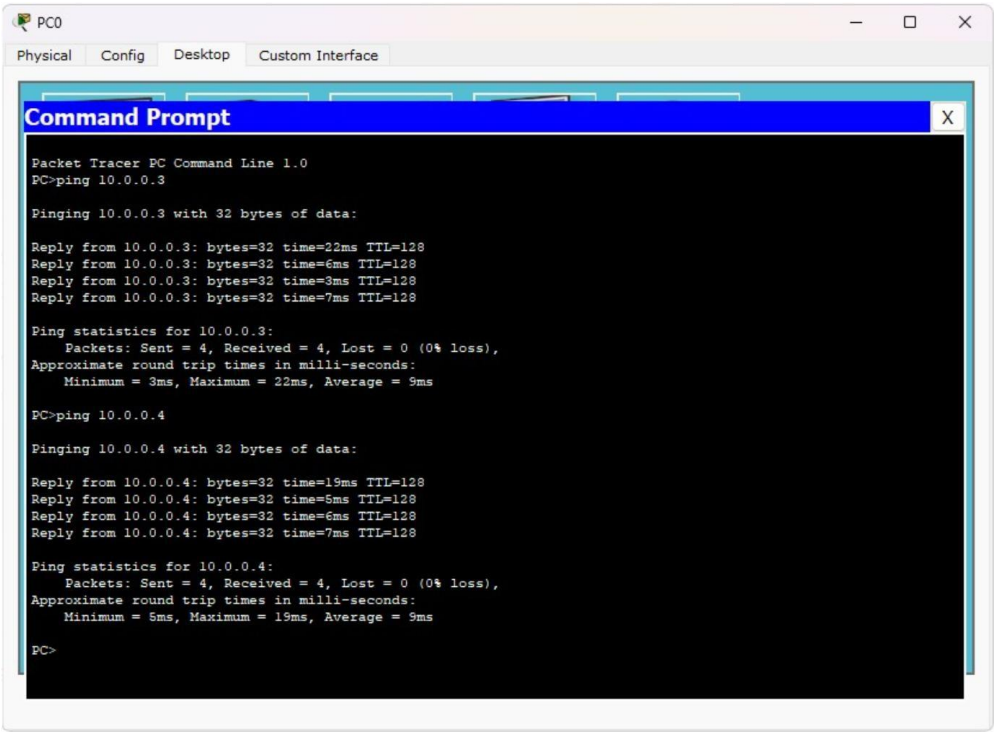
Aim of the program:
To construct a WLAN and make the nodes communicate wirelessly

Procedure along with the topology:



i. Construct the above topology.
ii. Configure PC and Router as normally done.
iii. Configure access point - Port1 -> SSID Name - any name (WLANhere)
iv. Select WEP and give any 10 digit hex key (1234567890 here)
v. Configure PC1 and laptop with wireless standards
vi. In PC1 Switch off the device, drag the existing PT-HOST-NM-1AM to the component listed in LHS.
vii. Drag WMP300N wireless interface to the empty port. Switch on PC
viii. 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
ix. Do similar in laptop.
x. Ping from every device to every other to check the result

Screen shots/ output :



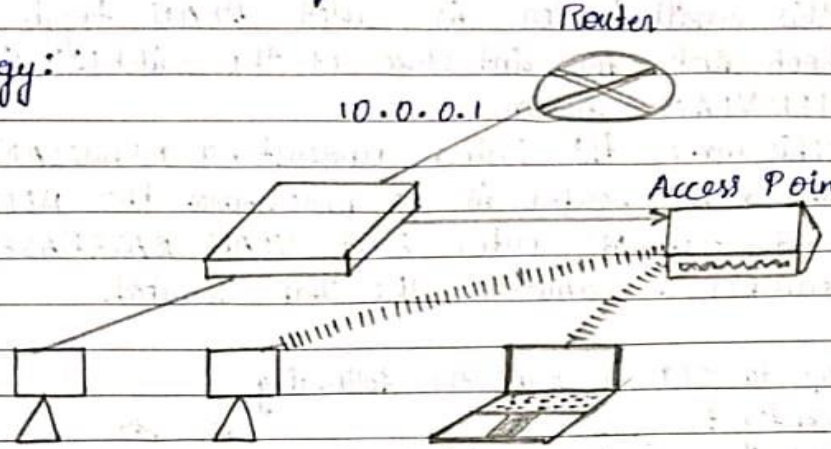
Observations:

- Device could connect to WLAN as long as they are in the network range
- Signal strength decreased with increase in distance
- Ping is successful

Observation Book:

5) AIM: To Construct a WLAN & make the nodes Communicate wirelessly.

Topology:



Steps are as follows:

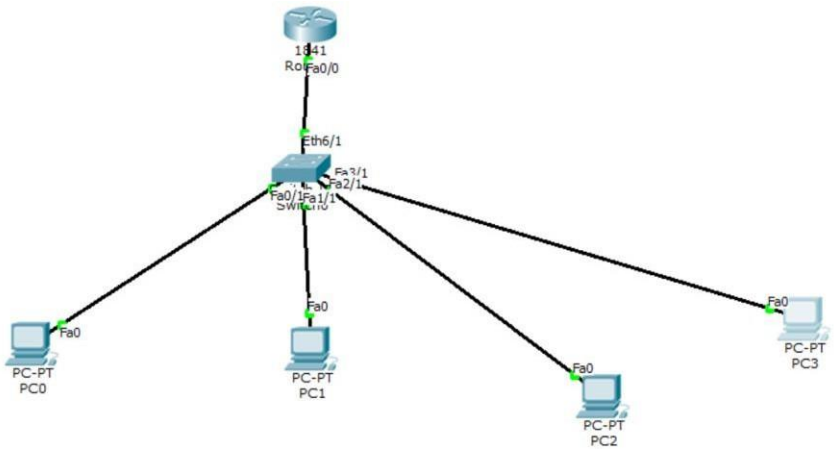
1) Construct the above topology
2) Configure PC3 & the Router1 as it is normally done.
3) Configure Access Point 1 - Port1 → SSID Name - any name (WLAN here).
4) Select WEP and give any 10 digit hex key - 1234567890
5) Configuring PC4 & laptop with wireless standards.
6) Switch off the device. drag the existing PT-HOST-NM-1AM to the Component listed in the LHS. Drag WM P300N wireless interface to the empty port. Switch on the device.
7) In the Configure tab a new wireless interface would have been added. Now Configure SSID, WEP key, IP address and Gateway to the device.

Observation: Ping from every device to every ~~other~~ device & see the results.

Program 13

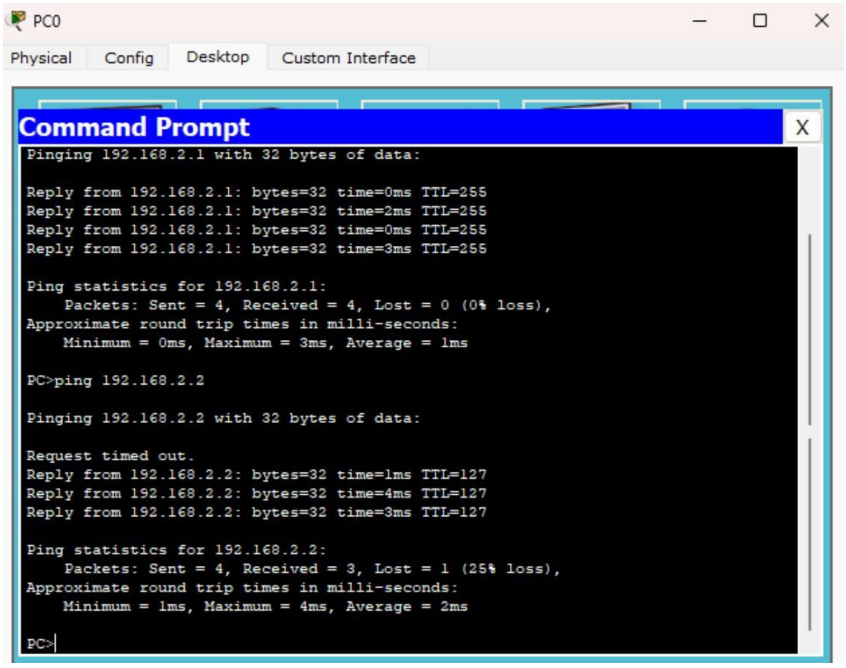
Aim of the program:
To construct a VLAN and make the PC's communicate among a VLAN

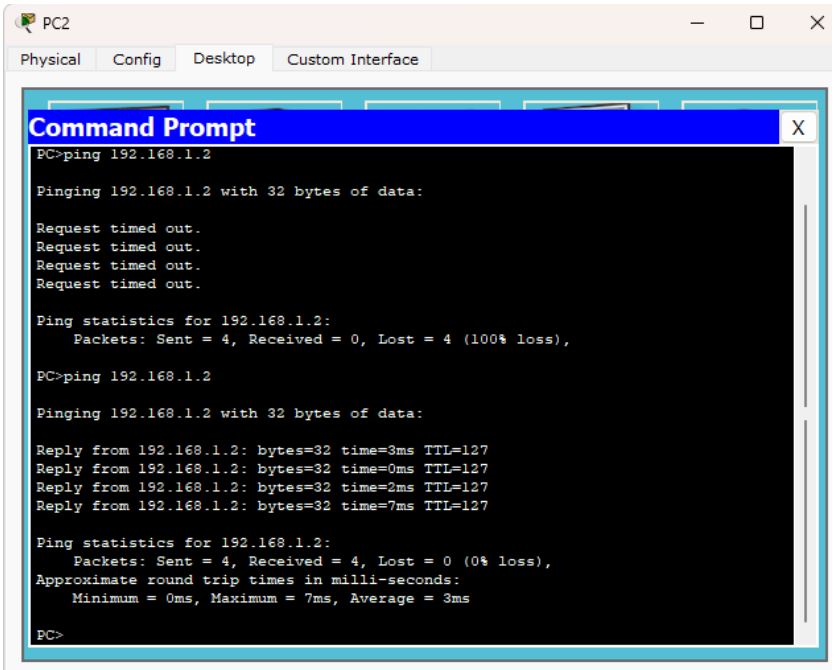
Procedure along with the topology:

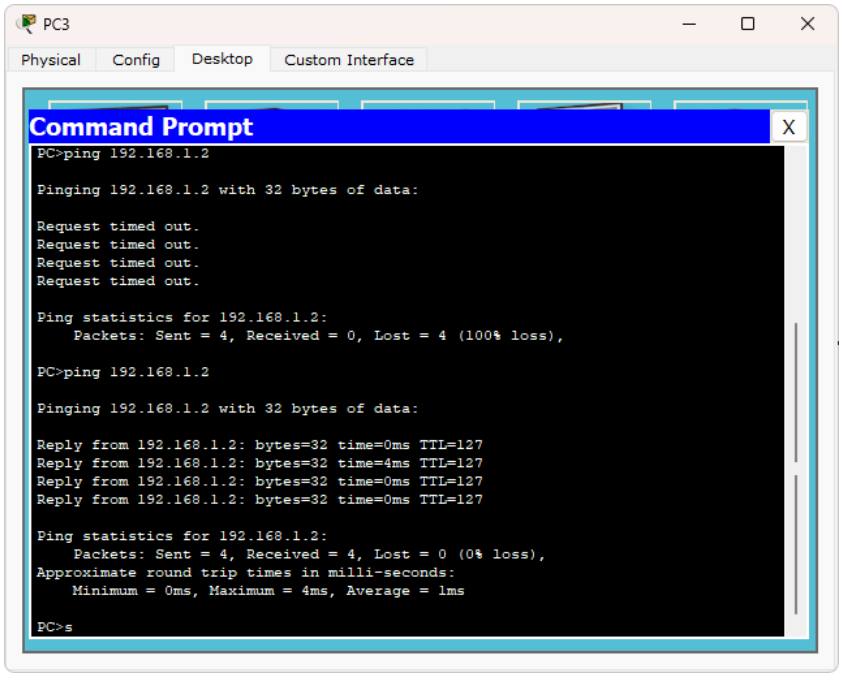


i. Create the topology seen above using router 1841
ii. Assign IPs as shown in the topology
iii. Go to the switch, choose VLAN database to configure the VLAN
iv. Give VLAN number and VLAN name add it
v. Select the interface i.e. fastethernet0/1 (near the switch from router) and make it trunk. (VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk)
vi. To make router understand VLAN,
    - Go to config tab of router, select VLAN database, enter the number and name of VLAN created.
    - Go to CLI
    - Router(vlan)#exit
    - Router#config t
    - Router(config)#interface fastethernet0/1
    - Router(config-subif)# #encapsulation dot1q 2
    - #ip address 192.168.21 255.255.255.0
    - #no shut
    - #exit
    - Router(config)#exit

Screen shots/ output :







Observations:

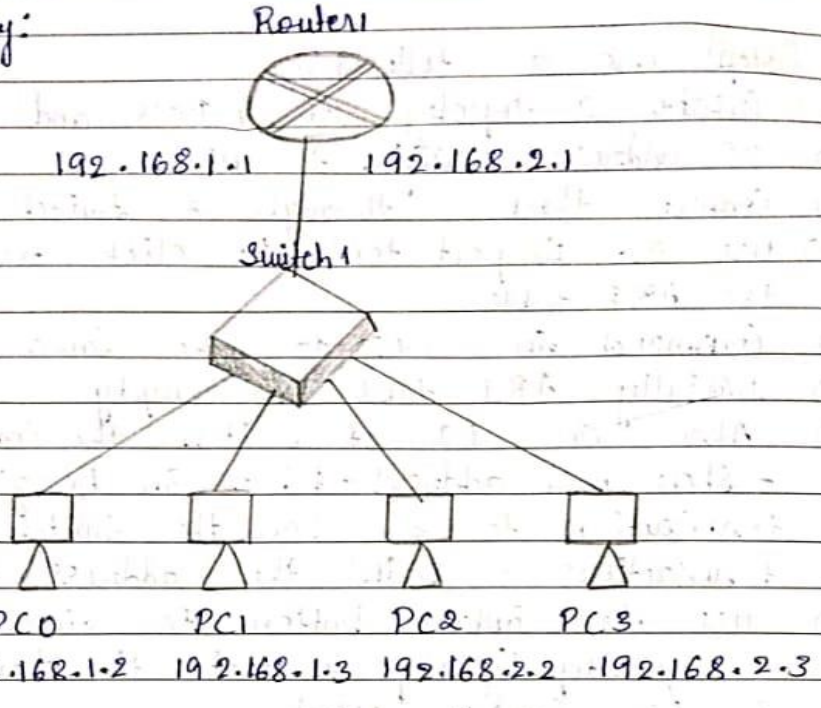
- Proper trunk configuration is established to make VLAN work properly
- Ping from one VLAN to another works properly

Observation Book:

LAB-

2) AIM: To Construct a VLAN and make the PC's communicate among a VLAN

Topology:



Steps:

1) To create a new VLAN, we use a class C type addresses.
2) Create a topology as seen below choose the 1841 router.
3) In the switch, go to config tab and select VLAN database.
4) Give any VLAN number say 2 here include any name. Say Add.
5) Select the interface is fastethernet 4/1 (near the switch from router) and make it the trunk.
6) VLAN trunking allows switches to forwards frames from different VLAN'S over a single link called trunk.

Bafna Gold
Date: Page:

→ 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.

8) hook into the interfaces of the switches with the 2 NEWVLAN systems.

This makes the switch understand NEWVLAN.

Next the router is to understand the NEWVLAN

9) Config tab of router select VLAN DATABASE enter the number & name of the Vlan created.

Go to CLI, & Run the following

→ Config t
→ interface fastEthernet 0/0.1
→ encapsulation dot1q 2
→ ip address 192.168.2.1 255.255.255.0
→ no shut
→ exit

Observation:

1) VLANS are used to logically segment a network, devices in the same VLAN to communicate directly while isolating traffic from other VLAN'S.
2) Switches must be configured to create VLAN'S, assign VLAN IDS, & allocate ports to specific VLAN'S.

CYCLE – 2 PROGRAMS

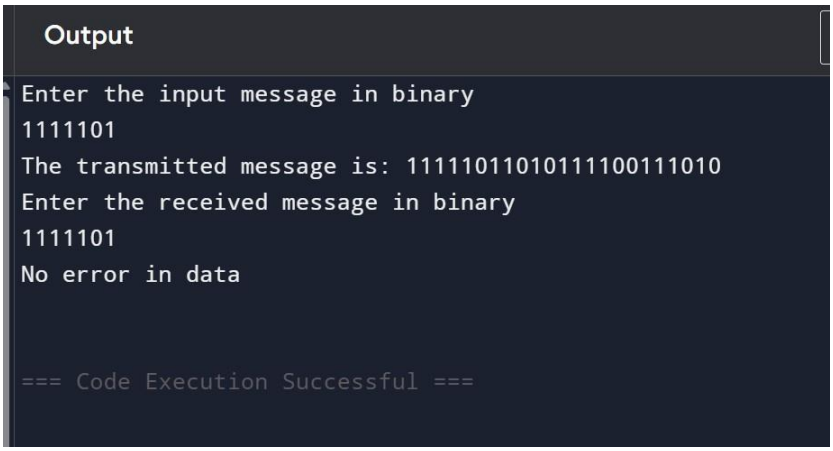
Program 14

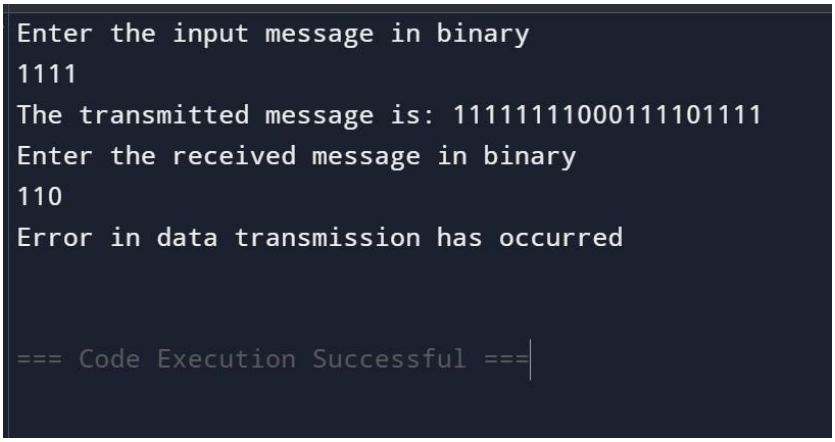
Aim of the program

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

```
#include <iostream>
#include <string.h>
using namespace std;
int crc(char *ip, char *op, char *poly, int mode)
{
strcpy(op, ip);
if (mode) {
for (int i = 1; i < strlen(poly); i++)
strcat(op, "0");
}
/* Perform XOR on the msg with the selected polynomial */
for (int i = 0; i < strlen(ip); i++) {
if (op[i] == '1') {
for (int j = 0; j < strlen(poly); j++) {
if (op[i + j] == poly[j])
op[i + j] = '0';
else
op[i + j] = '1';
}
}
}
/* check for errors. return 0 if error detected */
for (int i = 0; i < strlen(op); i++)
if (op[i] == '1')
return 0;
return 1;
}
int main()
{
char ip[50], op[50], recv[50];
/* x 16 + x12 + x5 + 1 */
char poly[] = "10001000000100001";
cout << "Enter the input message in binary"<< endl;
cin >> ip;
crc(ip, op, poly, 1);
cout << "The transmitted message is: " << ip << op + strlen(ip) << endl;
cout << "Enter the received message in binary" << endl;
cin >> recv;
if (crc(recv, op, poly, 0))
cout << "No error in data" << endl;
else

cout << "Error in data transmission has occurred" << endl;
return 0;}
```

Output





Program 15

Aim of the program

Write a program for congestion control using Leaky bucket algorithm.

```
#include <iostream>
#include <cstdlib>
#include <unistd.h>
using namespace std;

#define NOF_PACKETS 10

// Function to generate random numbers within a range
int generateRandom(int range) {
    int rn = (rand() % range);
    return rn == 0 ? 1 : rn; // Ensure non-zero values
}

int main() {
```

```
int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm = 0, p_time, op;

// Generate random packet sizes
for (i = 0; i < NOF_PACKETS; ++i)
    packet_sz[i] = generateRandom(6) * 10;

// Display packet sizes
for (i = 0; i < NOF_PACKETS; ++i)
    cout << "\nPacket[" << i << "]: " << packet_sz[i] << " bytes";

// Input: Output rate and bucket size
cout << "\nEnter the Output rate: ";
cin >> o_rate;
cout << "Enter the Bucket Size: ";
cin >> b_size;

// Process each packet
for (i = 0; i < NOF_PACKETS; ++i) {
    // Check if the packet size exceeds bucket size
    if ((packet_sz[i] + p_sz_rm) > b_size) {
        if (packet_sz[i] > b_size) {
            // Packet size exceeds bucket capacity
            cout << "\n\nIncoming packet size (" << packet_sz[i] << " bytes) exceeds bucket capacity ("
                 << b_size << " bytes) - PACKET REJECTED!";
        } else {
            // Bucket overflow
            cout << "\n\nBucket capacity exceeded - PACKETS REJECTED!";
        }
    } else {
        // Accept packet into the bucket
        p_sz_rm += packet_sz[i];
        cout << "\n\nIncoming Packet size: " << packet_sz[i];
        cout << "\nBytes remaining to transmit: " << p_sz_rm;

        // Simulate transmission
        p_time = generateRandom(4) * 10;
        cout << "\nTime allocated for transmission: " << p_time << " units";

        for (clk = 10; clk <= p_time; clk += 10) {
            sleep(1); // Simulate time delay
            if (p_sz_rm > 0) {
                if (p_sz_rm <= o_rate) {
                    op = p_sz_rm;
                    p_sz_rm = 0;
                } else {
                    op = o_rate;
                    p_sz_rm -= o_rate;
                }
                cout << "\nPacket of size " << op << " transmitted.";
                cout << "-----Bytes remaining to transmit: " << p_sz_rm;
            } else {
                cout << "\nTime left for transmission: " << (p_time - clk) << " units";
```

```
                cout << "\nNo packets to transmit!";
            }
        }
    }
  }
  return 0;
}
```

Output

```
Packet[0]: 10 bytes
Packet[1]: 40 bytes
Packet[2]: 30 bytes
Packet[3]: 10 bytes
Packet[4]: 50 bytes
Packet[5]: 10 bytes
Packet[6]: 40 bytes
Packet[7]: 10 bytes
Packet[8]: 30 bytes
Packet[9]: 10 bytes
Enter the Output rate: 20
Enter the Bucket Size: 50


Incoming Packet size: 10
Bytes remaining to transmit: 10
Time allocated for transmission: 20 units
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 40
Bytes remaining to transmit: 40
Time allocated for transmission: 30 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 20
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 30
Bytes remaining to transmit: 30
Time allocated for transmission: 20 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
```

```
Incoming Packet size: 10
Bytes remaining to transmit: 10
Time allocated for transmission: 20 units
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 40
Bytes remaining to transmit: 40
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 20

Incoming Packet size: 10
Bytes remaining to transmit: 30
Time allocated for transmission: 20 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0

Incoming Packet size: 30
Bytes remaining to transmit: 30
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10

Incoming Packet size: 10
Bytes remaining to transmit: 20
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 0

=== Code Execution Successful ===
```

Observation Book:

Bafna Gold
Date: Page:

leakybucket

1) write a program for congestion control using leaky bucket algorithm.

program : lbucket.cc

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define NOF_PACKETS 5
/*
int rand (int a) {
int rn = (random() % 10) % a;
return rn == 0 ? 1 : rn;
}
*/
/*
#include <stdlib.h>
   long int random(void);
*/
int main() {
int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate,
p_sz_rm = 0, p_sz, p_time, op;
for (i = 0; i <= NOF_PACKETS; ++i)
   packet_sz[i] = random() % 100;
for (i=0; i <NOF_PACKETS; ++i)
   printf("\n packet[%d]: %d bytes \t", i, packet_sz
[i]);
printf("\n Enter the Output rate:");
Scanf("%d", &o_rate);
printf("Enter the bucket size:");
Scanf("%d", &b_size);
for (i=0; i <NOF_PACKETS; ++i)
```

```
{if (packet_sz[i] + p_sz_rm) > b_size)
   if (packet_sz[i] > b_size)
     /* Compare the packet size with bucket size */
        printf ("\n\n Incoming packet size (%d bytes) is
Greater than bucket capacity (%d bytes) - PACKET
REJECTED ", packet_sz[i], b_size);
   else:
        printf ("\n\n Bucket capacity exceeded - PACKETS
REJECTED!!");
else {
p_sz_rm += packet_sz[i];
printf ("\n\n Incoming packet size: %d ", packet_sz[i]);
printf ("\n Bytes remaining to transmit: %d", p_sz_rm);
// p_time = random() * 10;
// printf ("\n Time left for transmission: %d units",
p_time);
// for (clk = 10; clk <= p_time; clk += 10)
while (p_sz_rm > 0) {
Sleep (1);
if (p_sz_rm) {
 if (p_sz_rm <= o_rate) /* packet size remaining
Comparing with output rate */
     op = p_sz_rm, p_sz_rm = 0;
else
     op = o_rate, p_sz_rm -= o_rate;
     printf ("\n Packet of size %d Transmitted", op);
     printf ("--- Bytes Remaining to Transmit: %d",
p_sz_rm);
}
else {
printf ("\n No packets to transmit!!"); }}}}}
```

Program 16

Aim of the program

Using TCP/IP 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.

Client Side

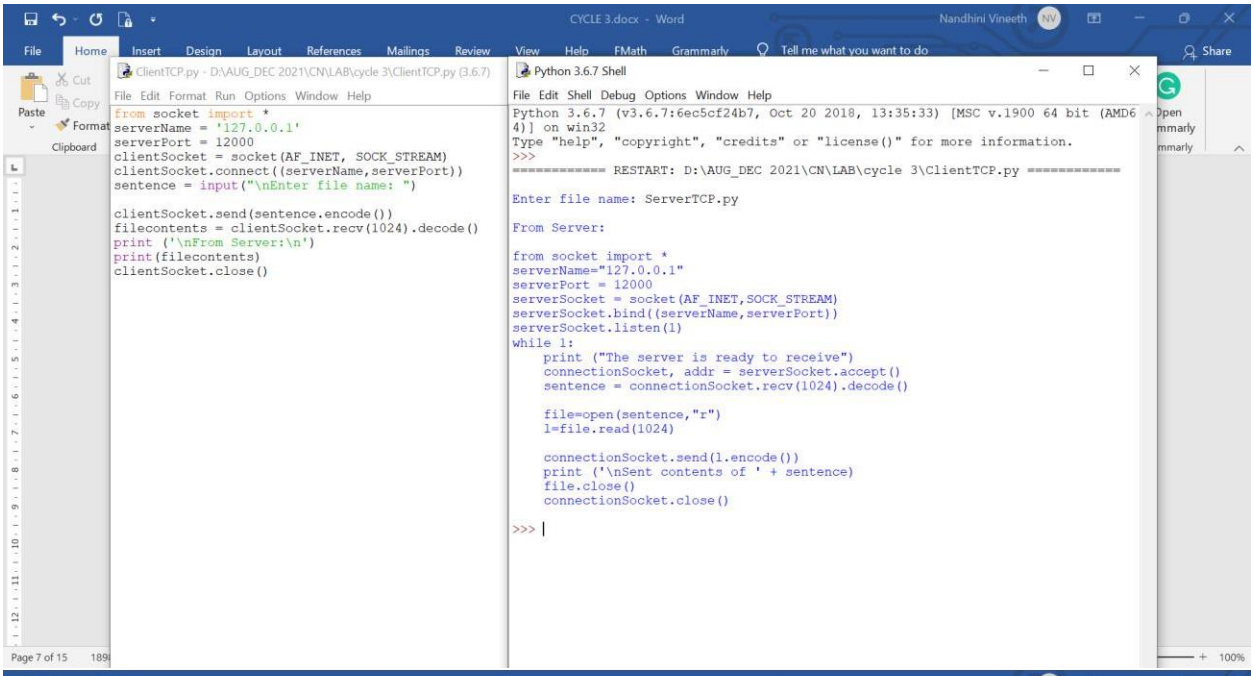
```
#include <unistd.h>
int main()
{
int soc, n;
char buffer[1024], fname[50];
struct sockaddr_in addr;
/* socket creates an endpoint for communication and returns a file descriptor */
soc = socket(PF_INET, SOCK_STREAM, 0);
/*
* sockaddr_in is used for ip manipulation
* we define the port and IP for the connection.
*/
addr.sin_family = AF_INET;
addr.sin_port = htons(7891);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");

/* keep trying to esatablish connection with server */
while(connect(soc, (struct sockaddr *) &addr, sizeof(addr))) ;
printf("\nClient is connected to Server");
printf("\nEnter file name: ");
scanf("%s", fname);
/* send the filename to the server */
send(soc, fname, sizeof(fname), 0);
printf("\nRecieved response\n");
/* keep printing any data received from the server */
while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
printf("%s", buffer);
return 0;
}
```

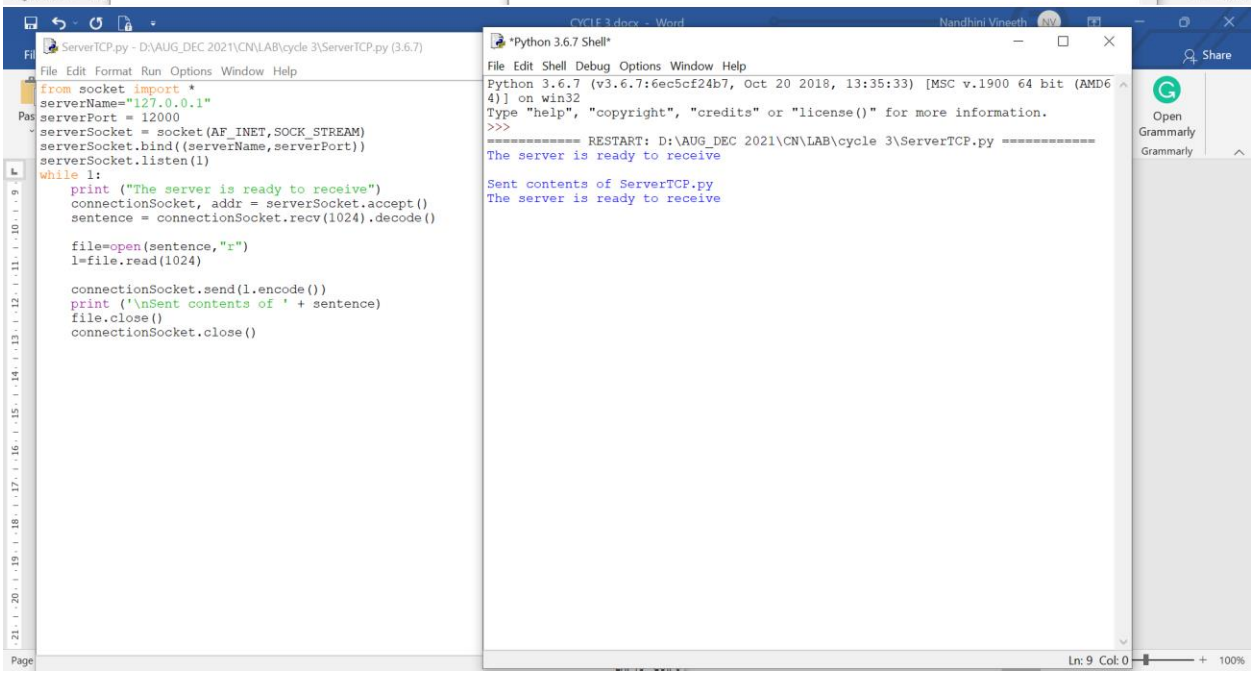
Server Side

```
#include <stdio.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{
int welcome, new_soc, fd, n;
char buffer[1024], fname[50];
struct sockaddr_in addr;
welcome = socket(PF_INET, SOCK_STREAM, 0);
```

```
addr.sin_family = AF_INET;
addr.sin_port = htons(7891);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
bind(welcome, (struct sockaddr *) &addr, sizeof(addr));
printf("\nServer is Online");
/* listen for connections from the socket */
listen(welcome, 5);
/* accept a connection, we get a file descriptor */
new_soc = accept(welcome, NULL, NULL);

/* receive the filename */
recv(new_soc, fname, 50, 0);
printf("\nRequesting for file: %s\n", fname);
/* open the file and send its contents */
fd = open(fname, O_RDONLY);
if (fd < 0)
send(new_soc, "\nFile not found\n", 15, 0);
else
while ((n = read(fd, buffer, sizeof(buffer))) > 0)
send(new_soc, buffer, n, 0);
printf("\nRequest sent\n");
close(fd);
return 0;
}
```

OUTPUT:


CYCLE 3.docx - Word
Nandhini Vineeth
ClientTCP.py - D:\AUG_DEC 2021\CN\LAB\cycle 3\ClientTCP.py (3.6.7)
File Edit Format Run Options Window Help
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
Python 3.6.7 Shell
File Edit Shell Debug Options Window Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\AUG_DEC 2021\CN\LAB\cycle 3\ClientTCP.py ==========
Enter file name: ServerTCP.py
From Server:
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 ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
>>>


CYCLE 3.docx - Word
Nandhini Vineeth
ServerTCP.py - D:\AUG_DEC 2021\CN\LAB\cycle 3\ServerTCP.py (3.6.7)
File Edit Format Run Options Window Help
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 ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,"r")
l=file.read(1024)
connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close()
connectionSocket.close()
Python 3.6.7 Shell
File Edit Shell Debug Options Window Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\AUG_DEC 2021\CN\LAB\cycle 3\ServerTCP.py ==========
The server is ready to receive
Sent contents of ServerTCP.py
The server is ready to receive
Ln: 9 Col: 0

Observation Book:

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

Client program.

```
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("file \n From Server:\n")
print(fileContents)
ClientSocket.close()
```

Server.py program

```
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("The Server is ready to receive")
    ConnectionSocket, addr = ServerSocket.accept()
```

```
    Sentence = ConnectionSocket.recv(1024).decode()
    file = open(Sentence,"r")
file = open(Sentence,"r")
l = file.read(1024)
ConnectionSocket.send(l.encode())
print('\n Sent Contents of' +Sentence)
file.close()
ConnectionSocket.close()
```

Output:

Server → The server is ready to receive
Sent Contents of Server.py.
Client → Enter file name: Server.py

From Server:

```
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("The Server is ready to receive")
    ConnectionSocket, addr = ServerSocket.accept()
    Sentence = connectionSocket.recv(1024).decode()
    file = open(Sentence, "r")
    l = file.read(1024)
    ConnectionSocket.send(l.encode())
    print('\n Sent Contents of:' + Sentence)
    file.close()
    ConnectionSocket.close()
```

Program 17

Aim of the program

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 program

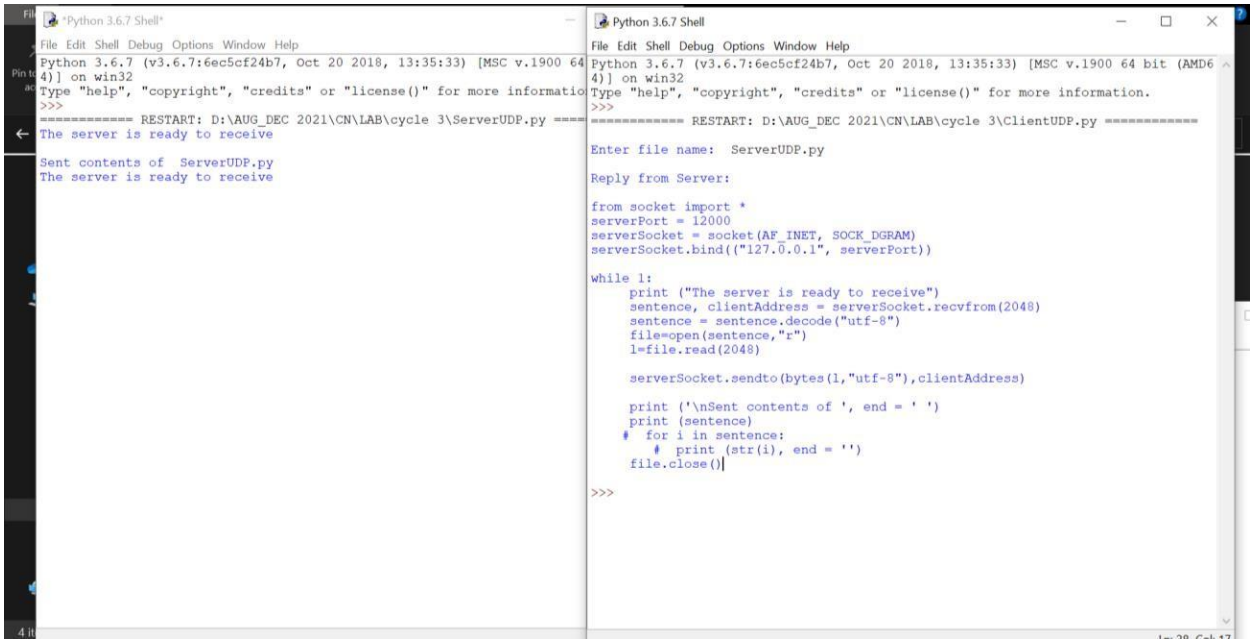
```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000
// Driver code
int main()
{
char buffer[100];
char *message = "Hello Client";
int listenfd, len;
struct sockaddr_in servaddr, cliaddr;
bzero(&servaddr, sizeof(servaddr));
// Create a UDP Socket
listenfd = socket(AF_INET, SOCK_DGRAM, 0);
servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
servaddr.sin_port = htons(PORT);
servaddr.sin_family = AF_INET;
// bind server address to socket descriptor
bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
//receive the datagram
len = sizeof(cliaddr);
int n = recvfrom(listenfd, buffer, sizeof(buffer),

0, (struct sockaddr*)&cliaddr,&len); //receive message from server

buffer[n] = '\0';
puts(buffer);
// send the response
sendto(listenfd, message, MAXLINE, 0,
(struct sockaddr*)&cliaddr, sizeof(cliaddr));

}
```

client driver program

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
```

```
#include <sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>
#include<stdlib.h>
#define PORT 5000
#define MAXLINE 1000
// Driver code
int main()
{

char buffer[100];
char *message = "Hello Server";
int sockfd, n;
struct sockaddr_in servaddr;
// clear servaddr
bzero(&servaddr, sizeof(servaddr));
servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
servaddr.sin_port = htons(PORT);
servaddr.sin_family = AF_INET;
// create datagram socket
sockfd = socket(AF_INET, SOCK_DGRAM, 0);
// connect to server
if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
{
printf("\n Error : Connect Failed \n");
exit(0);
}
// request to send datagram
// no need to specify server address in sendto
// connect stores the peers IP and port
sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));
// waiting for response
recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)NULL, NULL);
puts(buffer);
// close the descriptor
close(sockfd);
```

OUTPUT:


Python 3.6.7 Shell
File Edit Shell Debug Options Window Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64
4)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
========== RESTART: D:\AUG_DEC 2021\CN\LAB\cycle 3\ServerUDP.py ====
The server is ready to receive
Sent contents of ServerUDP.py
The server is ready to receive
Python 3.6.7 Shell
File Edit Shell Debug Options Window Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\AUG_DEC 2021\CN\LAB\cycle 3\ClientUDP.py ==========
Enter file name: ServerUDP.py
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 ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
>>>

Observation Book:

Q. Using UDP Sockets, write a Client-Server program to make client sending the file name & the Server to send back the contents of the requested file if present.

Solution:

Client.py

```
from socket import *
ServerName = "127.0.0.1"
Server port = 12000
ClientSocket = Socket(AF_INET, SOCK_DGRAM)
Sentence = input("\n Enter file name:")

ClientSocket.sendto(bytes(Sentence, "utf-8"),(ServerName, ServerPort))
fileContents, ServerAddress = ClientSocket.recvfrom(2048)
print('\n Reply from Sever:\n')
print(fileContents.decode("utf-8"))
#for i in fileContents:
    #print(str(i), end='')
ClientSocket.close()
ClientSocket.close()
```

Server.py

```
import Socket
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")
con = file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"), clientAddress)
print('\n Sent Contents of', end = ' ')
print(Sentence)
# for i in Sentence:
  # print(str(i), end = '')
file.close()
```

Output:

```
Server --> The Server is ready to receive
Sent Contents of Server.py.
The Server is ready to receive.
```

```
Client --> Client.py
Enter file name: Server.py
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)
```