VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (22CS4PCCON)

Submitted by

Sonal (1BM22CS286)

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 **Sonal(1BM22CS286),** 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.

Dr. Shashikala Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

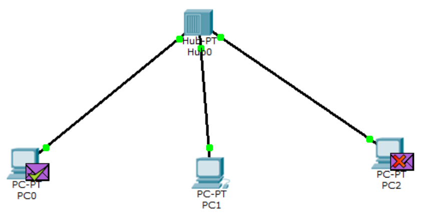
Github Link: sonalkolekar/CN

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:



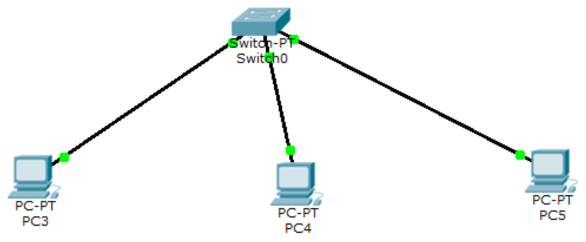
i. Select 3 end devices and a hub.
ii. Connect all the devices to the hub with a straight through cable.
iii. For each device do the configuration to set IP address for fast Ethernet.
iv. Add simple PDU, select source and destination and simulate the experiment.

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:



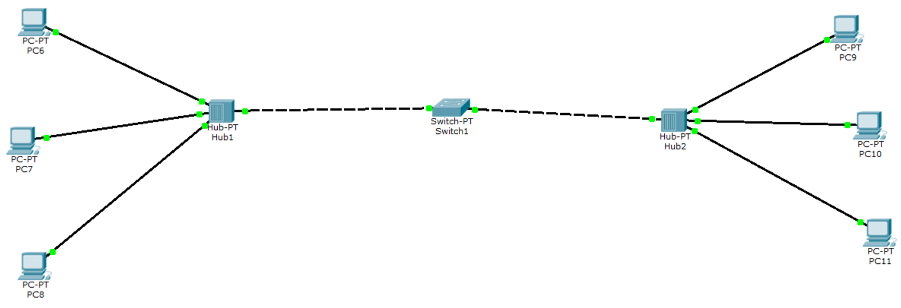
i. Select 4 end devices and a switch.
ii. Connect all the devices to the switch with straight through cable.
iii. For each device do the configuration. Set IP address for fast ethernet.
iv. Add simple PDU, select source and destination and simulate the experiment.

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.

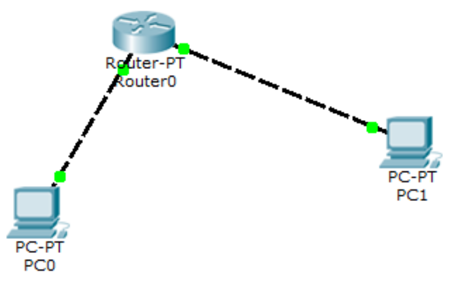
Observation:

- 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.

<u>Program 2</u>

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

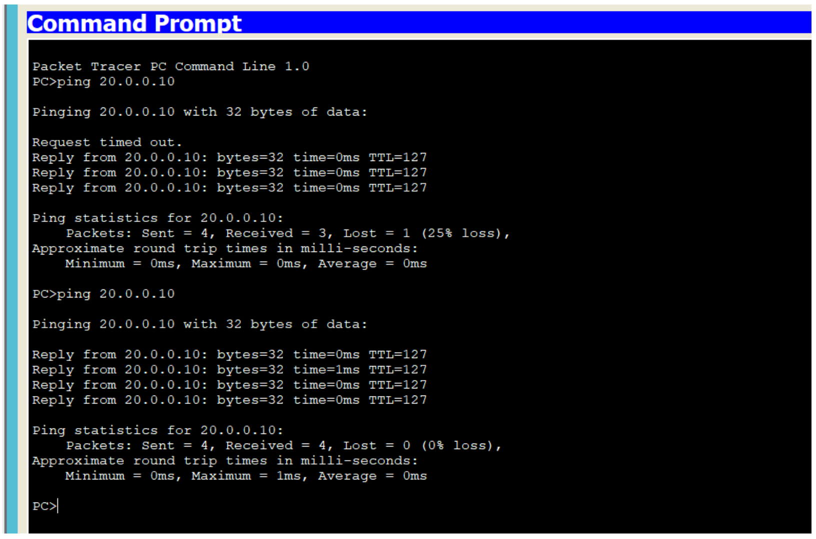
  PC 0 and router are successfully connected

  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 :



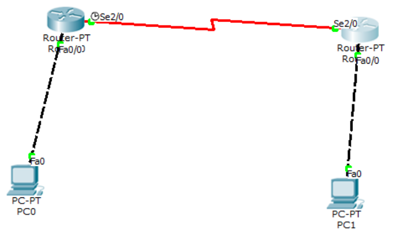
```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=1ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

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

Program 3

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

Procedure along with the topology:



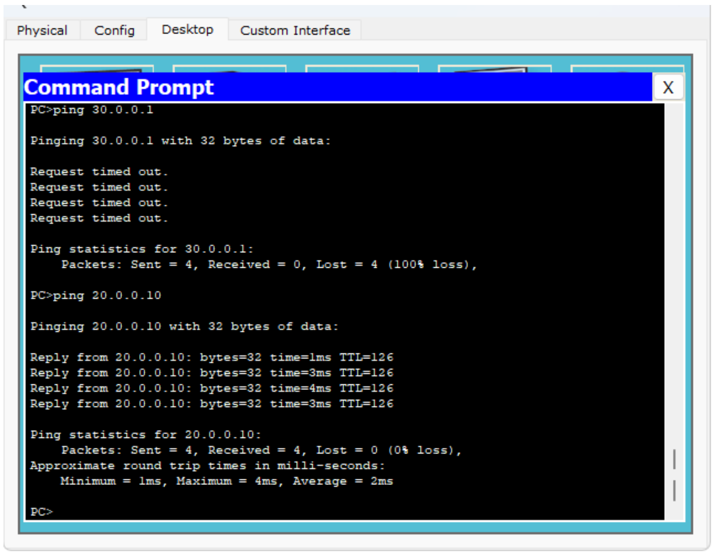
i. Select a generic router R1.
ii. Connect an end device PC1 to router R1 through parallel connection - fastethernet 010.
iii. Configure PC1 with ip address 10.0.0.10 and gateway 10.0.0.2
iv. Similarly select another generic router R2 and connect an end device PC2 - fastethernet 010.
v. Configure PC2 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

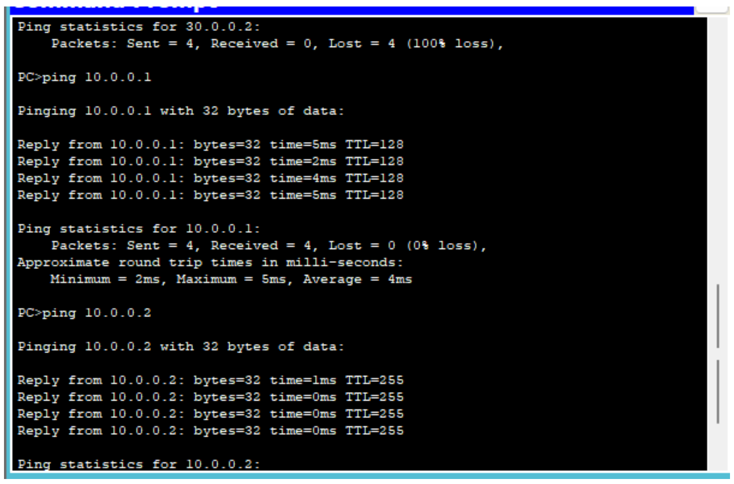
    "Interface fastethernet 1/0, changed state to up"
viii. Hence the connection between Router and end devices is established.
ix. Now connect router R1 with router R2 using serial cable (specially connected)
x. To setup connection b/w routers again
    - Select router R1 and go to CLI
    - Router(config)# 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”

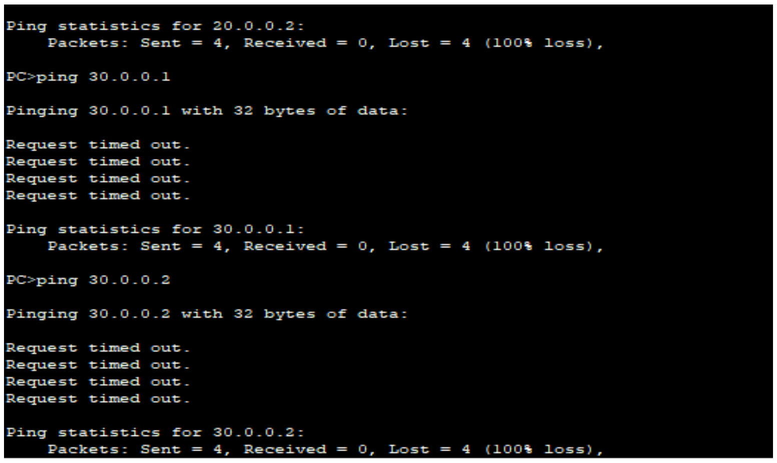
Screen shots/ output :



```
PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data:

Reply from 20.0.0.10: bytes=32 time=1ms TTL=126
Reply from 20.0.0.10: bytes=32 time=3ms TTL=126
Reply from 20.0.0.10: bytes=32 time=4ms TTL=126
Reply from 20.0.0.10: bytes=32 time=3ms TTL=126

Ping statistics for 20.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 4ms, Average = 2ms

PC>
```



```
Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=5ms TTL=128
Reply from 10.0.0.1: bytes=32 time=2ms TTL=128
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128
Reply from 10.0.0.1: bytes=32 time=5ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 5ms, Average = 4ms

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
```



```
Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

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

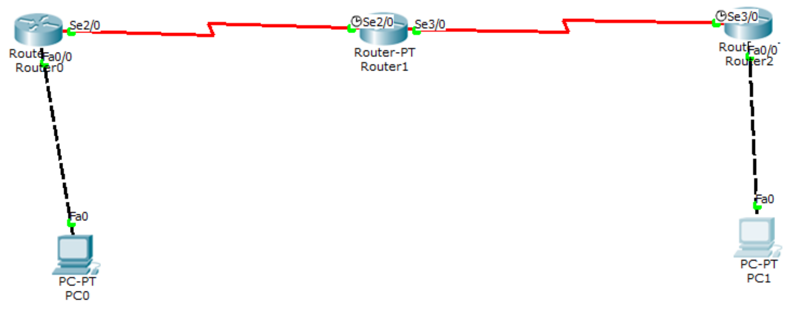
So now check if all the connections are done by going to each router's CLI and typing show ip route.

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

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

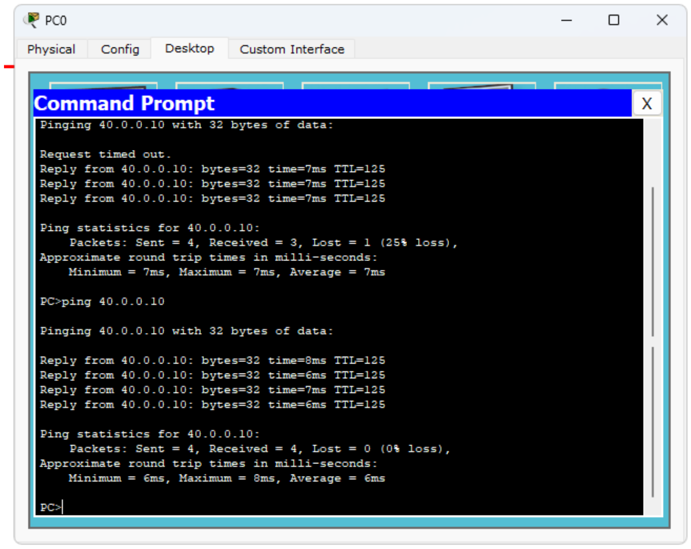
xviii. So we will go to Router 0

- Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

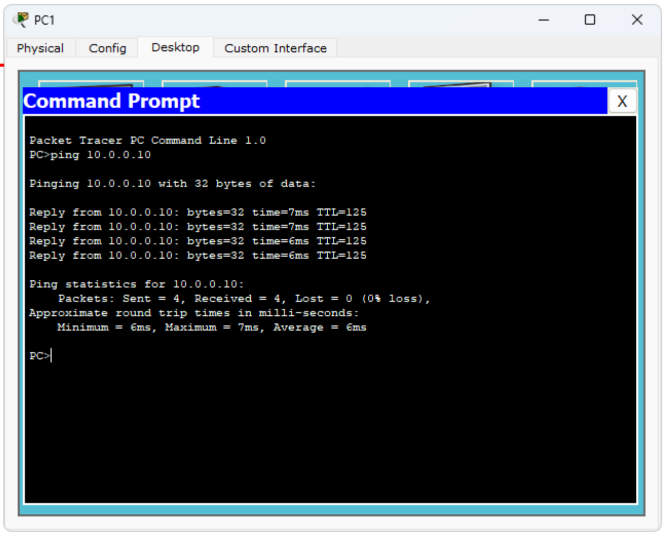
xix. Then Router 2

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

Screen shots/ output :



```
Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 7ms, Average = 7ms

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 8ms, Average = 6ms

PC>
```



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 7ms, Average = 6ms

PC>
```

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:

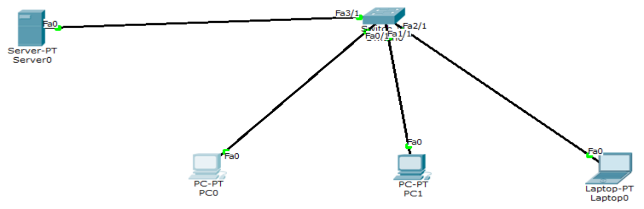
- PC> ping 10.0.0.10
- Pinging 10.0.0.10 with 32 bytes Packets: Sent = 4, Received = 4, Lost = 0

<u>Program 5</u>

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

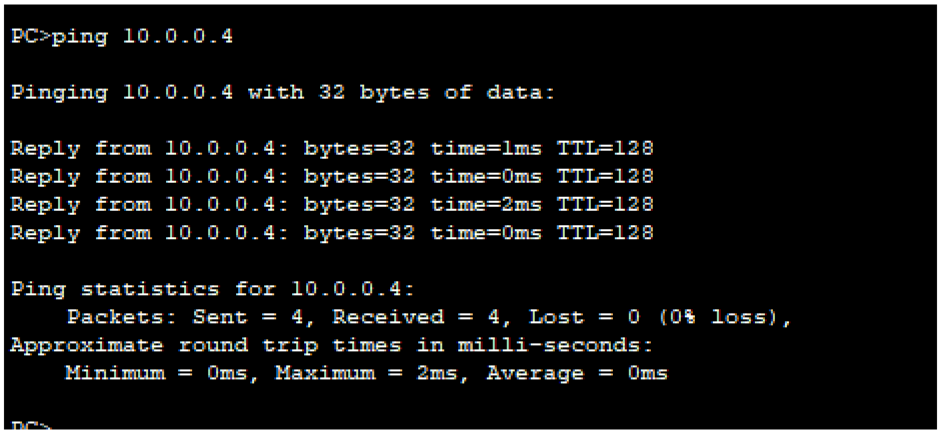
a) Within LAN

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 :



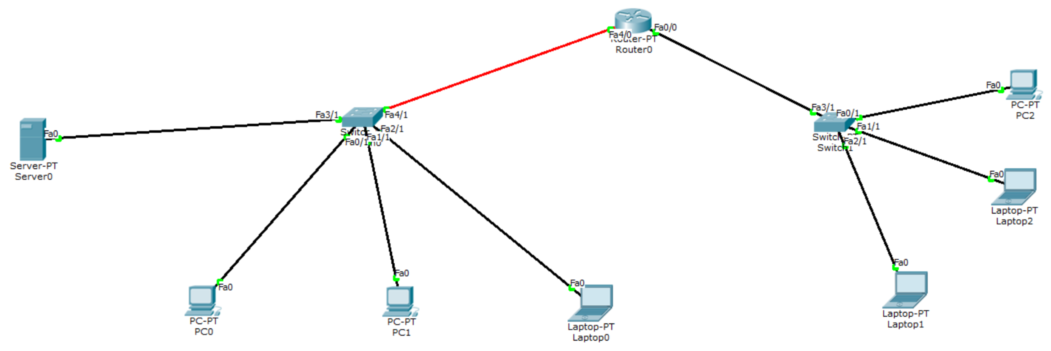
```
PC>ping 10.0.0.4

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)

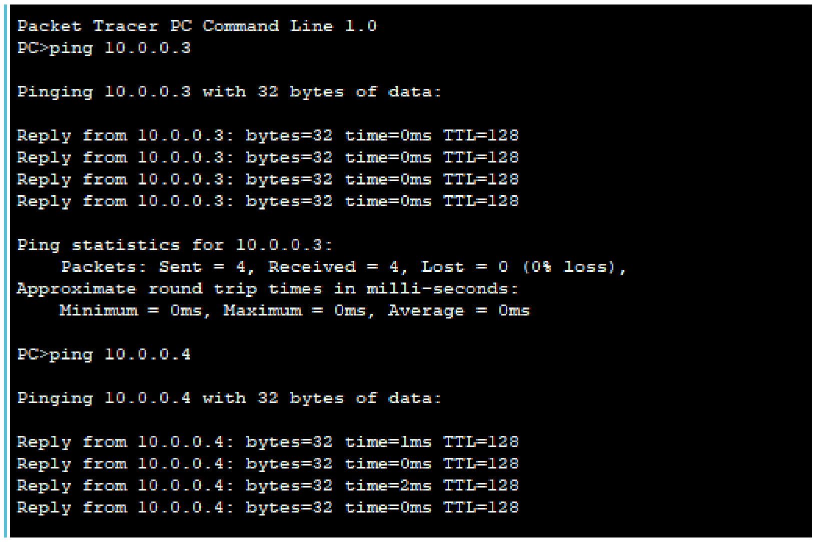
b) Outside LAN

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 :

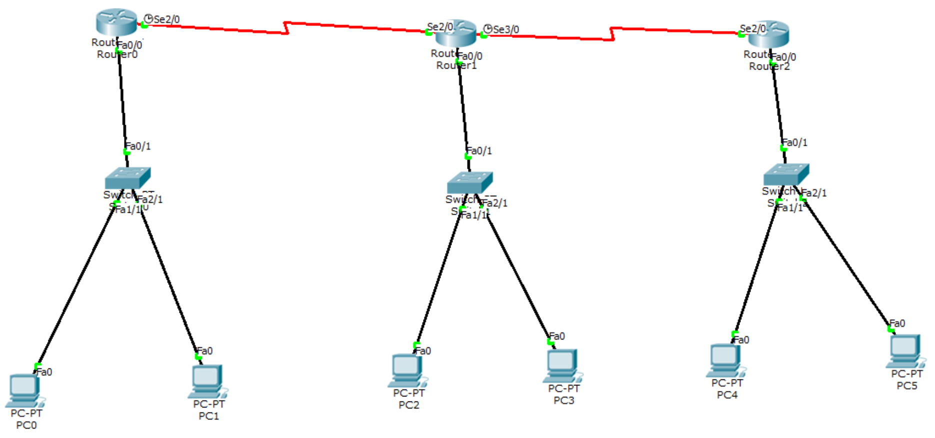

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

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.

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 ………….
  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

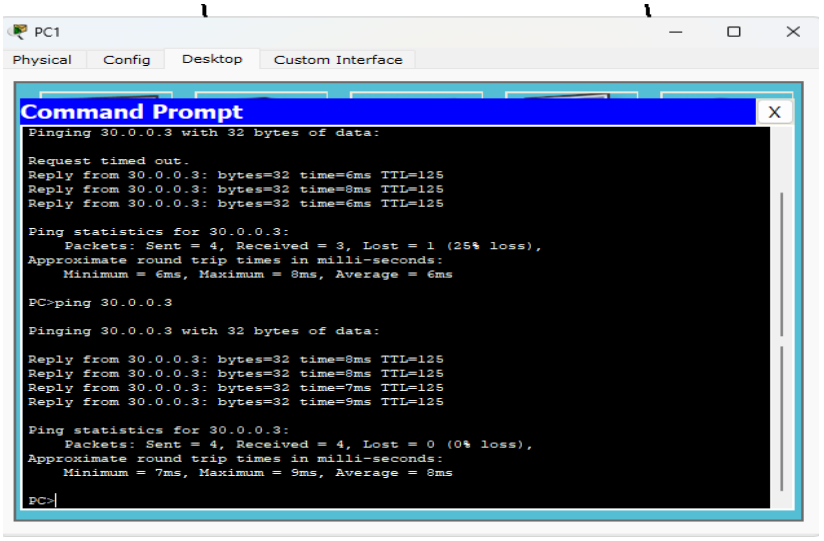
R
C . . .
C
C . . .

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
  C . . .
  C
  C . . .

Screen shots/ output :



```
Pinging 30.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.3: bytes=32 time=6ms TTL=125
Reply from 30.0.0.3: bytes=32 time=8ms TTL=125
Reply from 30.0.0.3: bytes=32 time=6ms TTL=125

Ping statistics for 30.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 8ms, Average = 6ms

PC>ping 30.0.0.3

Pinging 30.0.0.3 with 32 bytes of data:

Reply from 30.0.0.3: bytes=32 time=8ms TTL=125
Reply from 30.0.0.3: bytes=32 time=8ms TTL=125
Reply from 30.0.0.3: bytes=32 time=7ms TTL=125
Reply from 30.0.0.3: bytes=32 time=9ms TTL=125

Ping statistics for 30.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 9ms, Average = 8ms

PC>
```

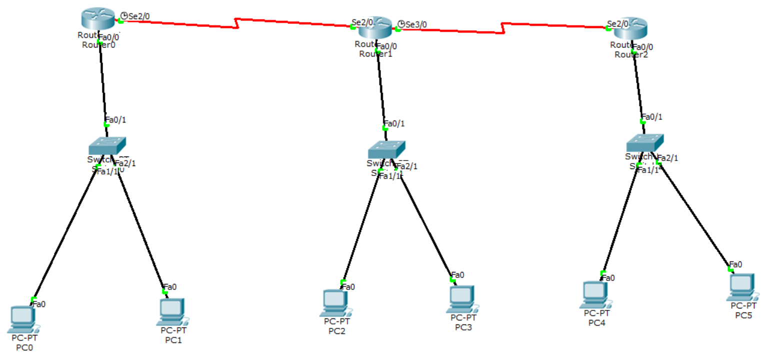
Observation:

From PC1 we can ping to PC5
 PC1> ping 30.0.0.3
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

Program 7

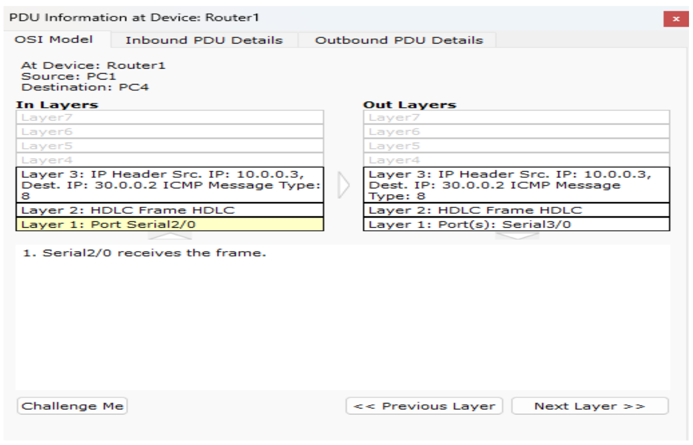
Aim of the program:
Demonstrate the TTL/ Life of a Packet:

Procedure along with the topology:



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

Screen shots/ output :



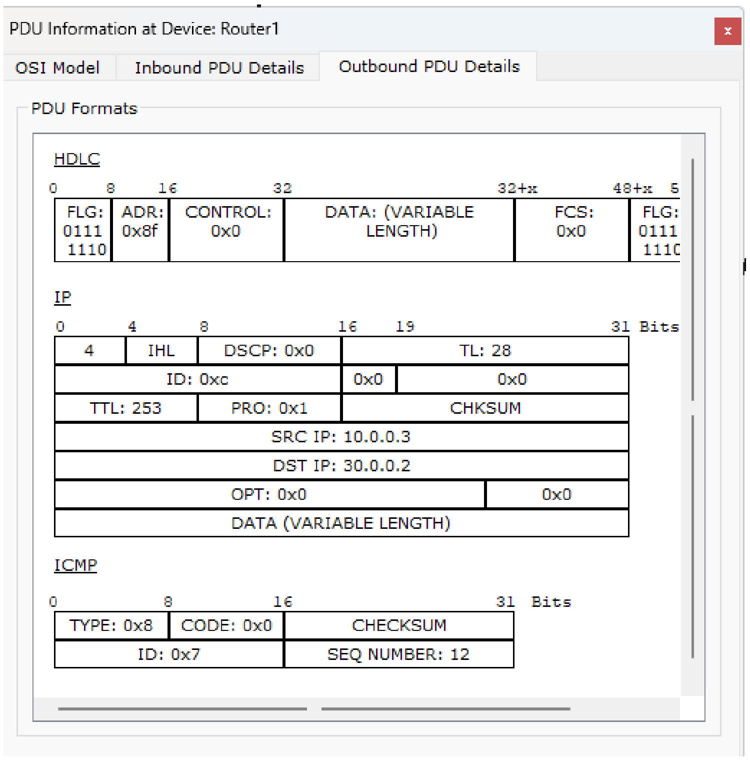
PDU Information at Device: Router1

At Device: Router1
Source: PC1
Destination: PC4

In Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.3, Dest. IP: 30.0.0.2 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port Serial2/0

Out Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.3, Dest. IP: 30.0.0.2 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial3/0

1. Serial2/0 receives the frame.



PDU Information at Device: Router1

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

HDLC

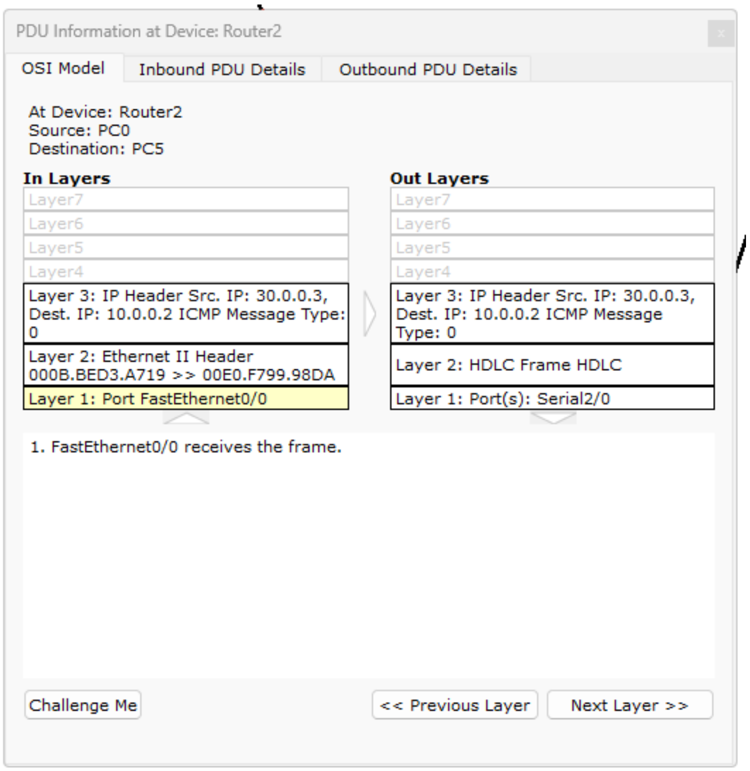
0	8	16	32	32+x	48+x	5
FLG: 0111 1110	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 0111 111	

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	



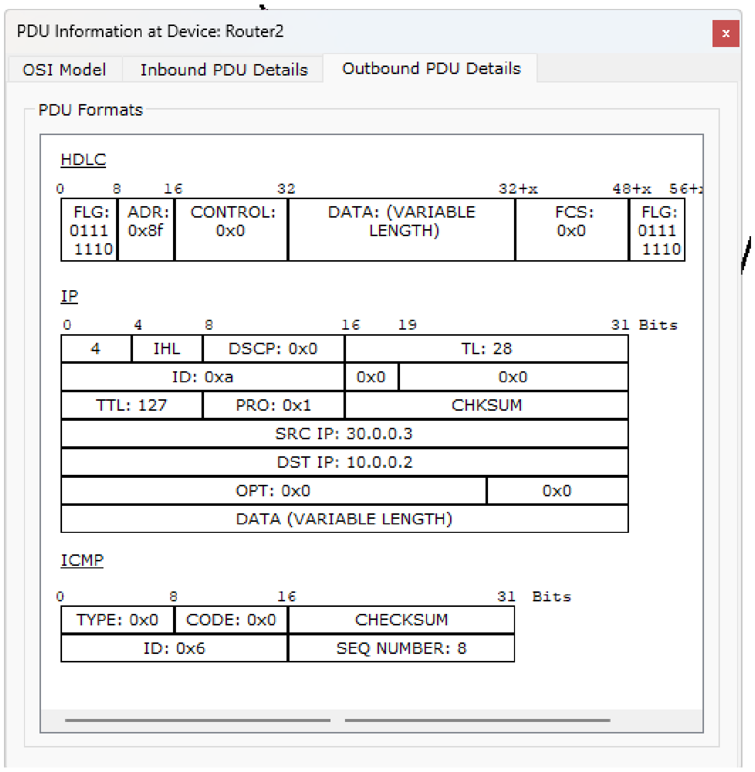
PDU Information at Device: Router2

OSI Model | Inbound PDU Details | Outbound PDU Details

At Device: Router2
Source: PC0
Destination: PC5

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 30.0.0.3, Dest. IP: 10.0.0.2 ICMP Message Type: 0	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 2: HDLC Frame HDLC
Layer 1: Port FastEthernet0/0	Layer 1: Port(s): Serial2/0

1. FastEthernet0/0 receives the frame.



PDU Information at Device: Router2

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

HDLC

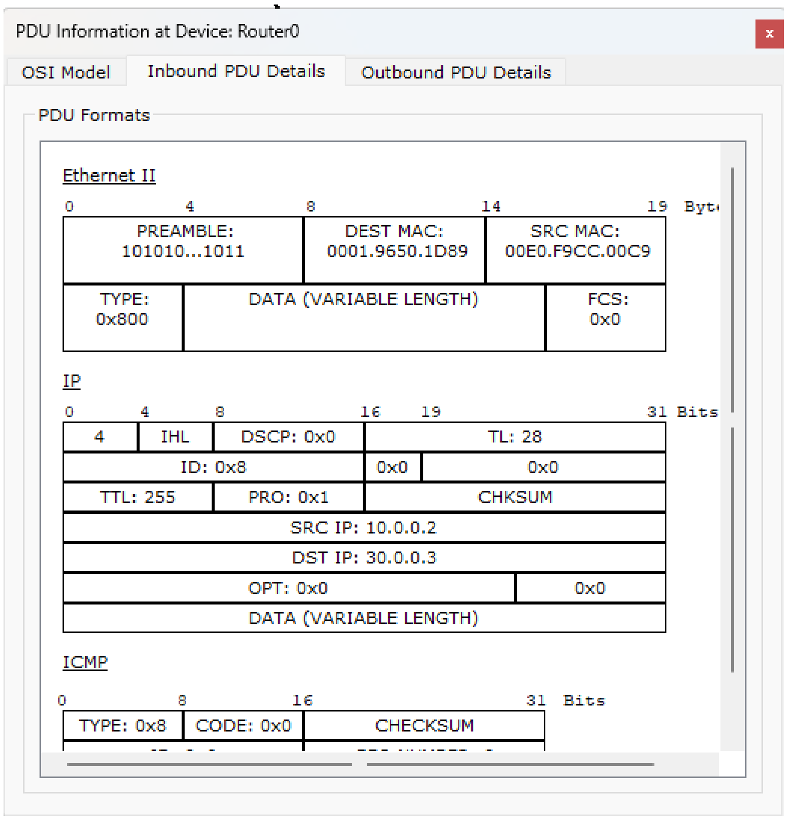
```
0      8     16           32                 32+x       48+x  56+x
FLG: 0111 1110 | ADR: 0x8f | CONTROL: 0x0 | DATA: (VARIABLE LENGTH) | FCS: 0x0 | FLG: 0111 1110
```

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

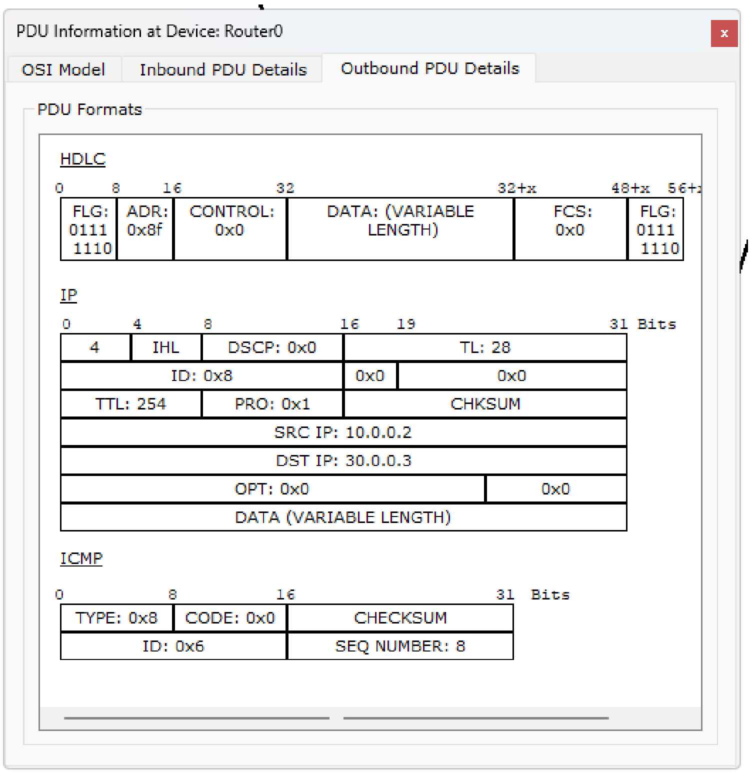
```
0           4          8          14              19  Byt
PREAMBLE: 101010...1011 | DEST MAC: 0001.9650.1D89 | SRC MAC: 00E0.F9CC.00C9
TYPE: 0x800 | DATA (VARIABLE LENGTH) | FCS: 0x0
```

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
```


PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

HDLC

FLG: 0111 1110	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 0111 1110

IP

4	IHL	DSCP: 0x0	TL: 28
ID: 0x8		0x0	0x0
TTL: 254	PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.2			
DST IP: 30.0.0.3			
OPT: 0x0			0x0
DATA (VARIABLE LENGTH)			

ICMP

TYPE: 0x8	CODE: 0x0	CHECKSUM
ID: 0x6		SEQ NUMBER: 8

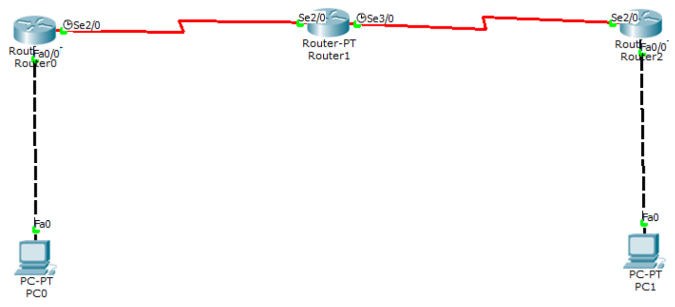
Observations:

- 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.

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

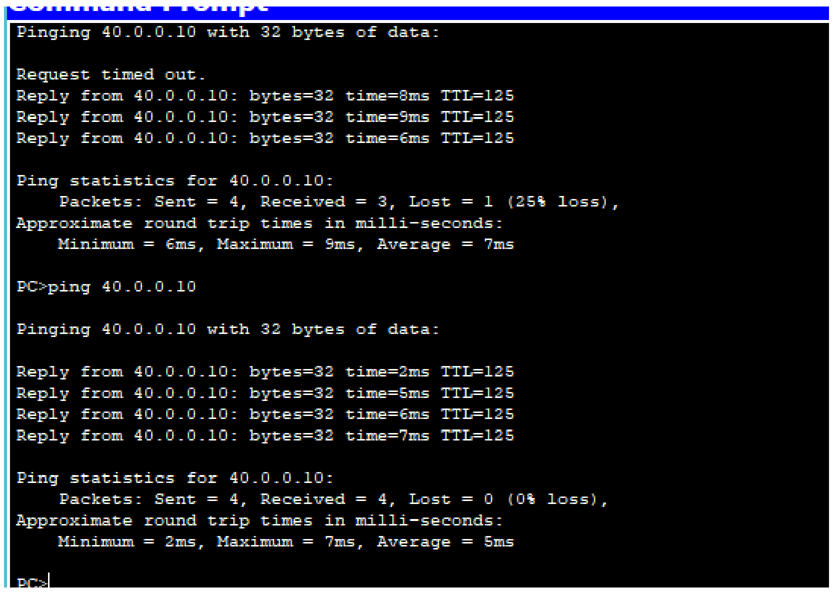
xviii. On Router R3

xix. R3#show ip route

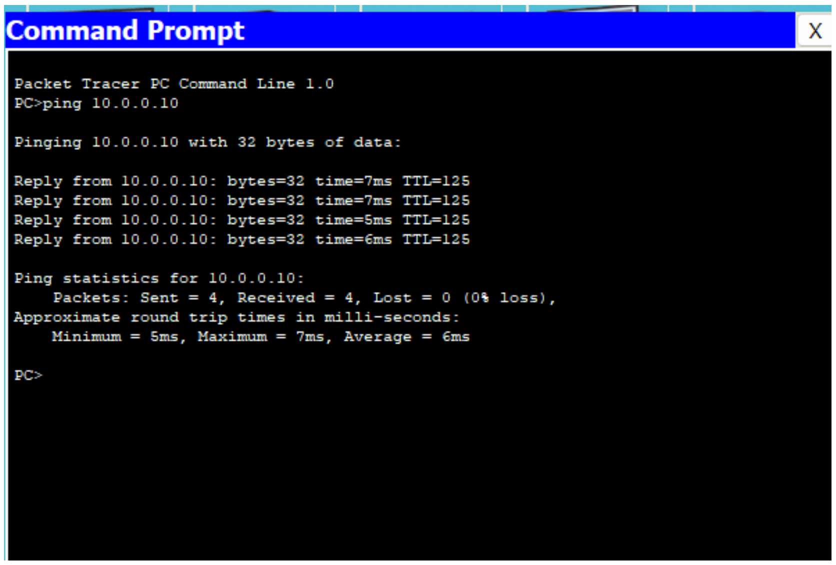
```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
Gateway of last resort is not set

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 :



```
Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 9ms, Average = 7ms

PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 5ms

PC>
```



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=5ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 7ms, Average = 6ms

PC>
```

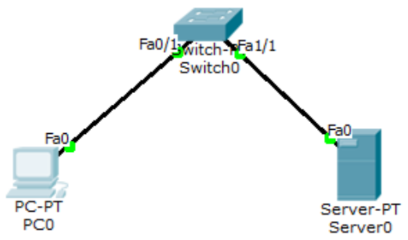
Observations:

- 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)

Program 9

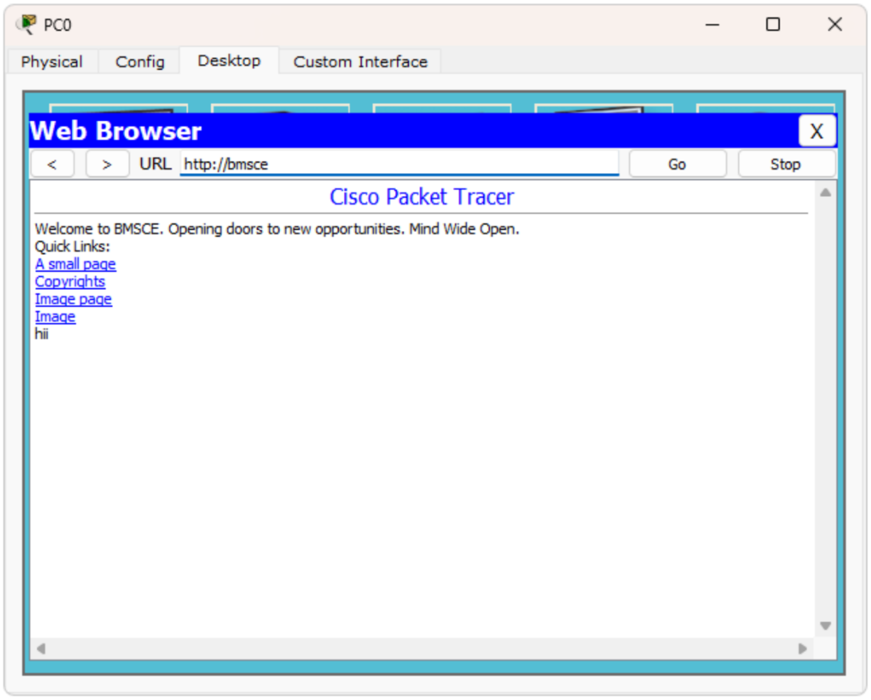
Aim of the program:
Configure Web Server, DNS within a LAN.

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 :



http://bmsce

Cisco Packet Tracer

Welcome to BMSCE. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image
hii

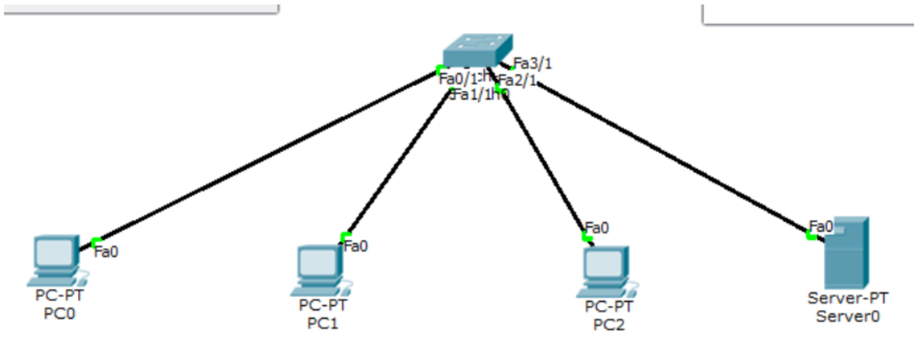
Observations:

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

Program 10

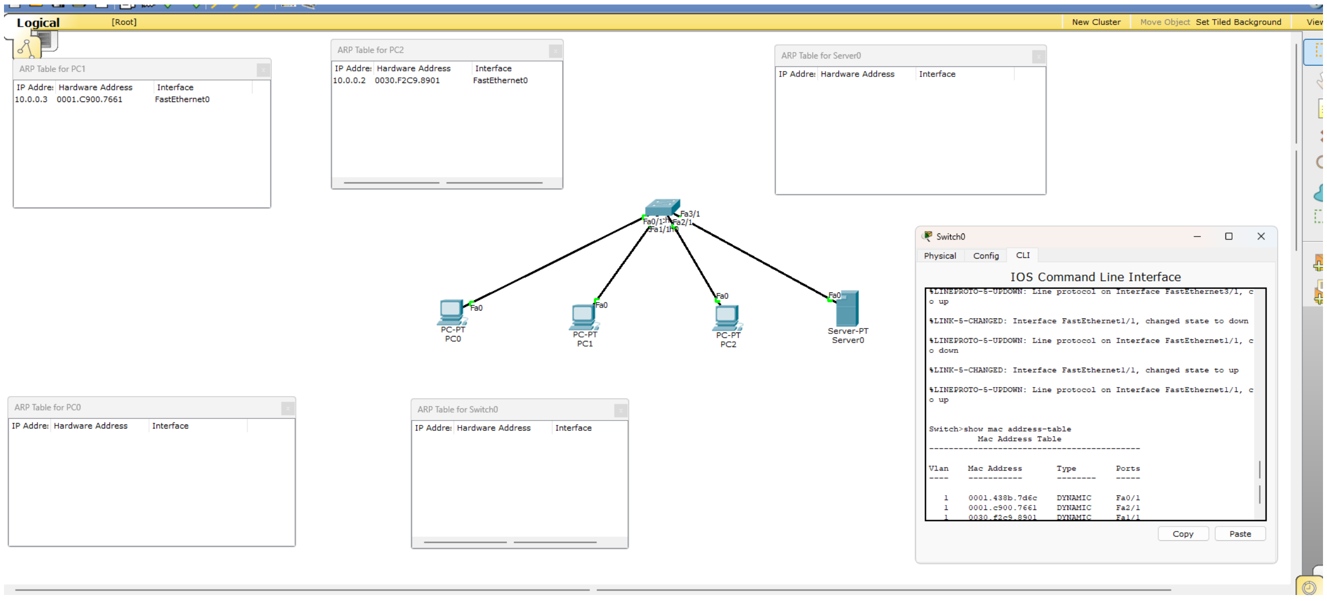
Aim of the program:
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

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 :



ARP Table for PC1

IP Addre:	Hardware Address	Interface
10.0.0.3	0001.C900.7661	FastEthernet0

ARP Table for PC2

IP Addre:	Hardware Address	Interface
10.0.0.2	0030.F2C9.8901	FastEthernet0

ARP Table for Server0

IP Addre:	Hardware Address	Interface

ARP Table for PC0

IP Addre:	Hardware Address	Interface

ARP Table for Switch0

IP Addre:	Hardware Address	Interface

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, c
o up

%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, c
o down

%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, c
o up


Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.438b.7d6c    DYNAMIC     Fa0/1
   1    0001.c900.7661    DYNAMIC     Fa2/1
   1    0030.f2c9.8901    DYNAMIC     Fa1/1
```

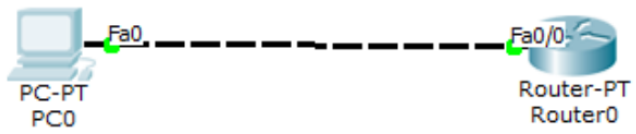
Observations:

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

Program 11

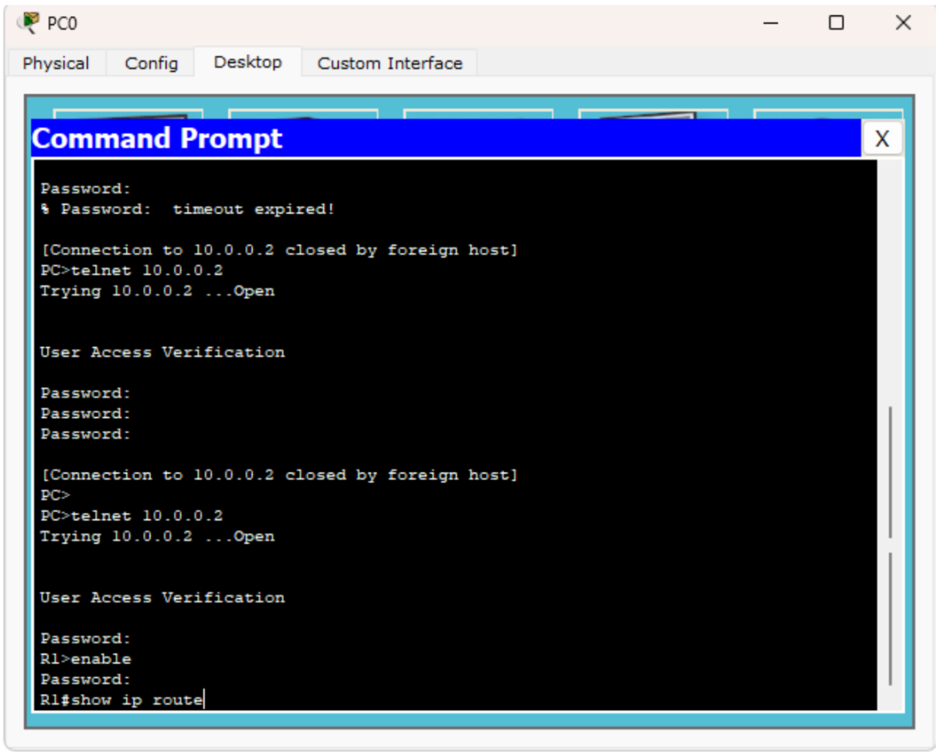
Aim of the program:
To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

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 :



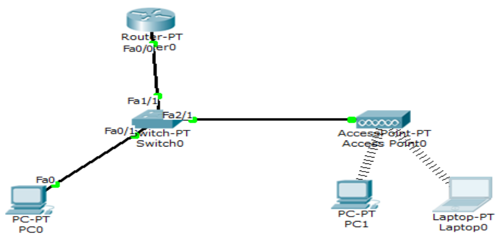
```
Password:
% Password:  timeout expired!

[Connection to 10.0.0.2 closed by foreign host]
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
Password:
Password:

[Connection to 10.0.0.2 closed by foreign host]
PC>
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
R1>enable
Password:
R1#show ip route
```

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

<u>Program 12</u>

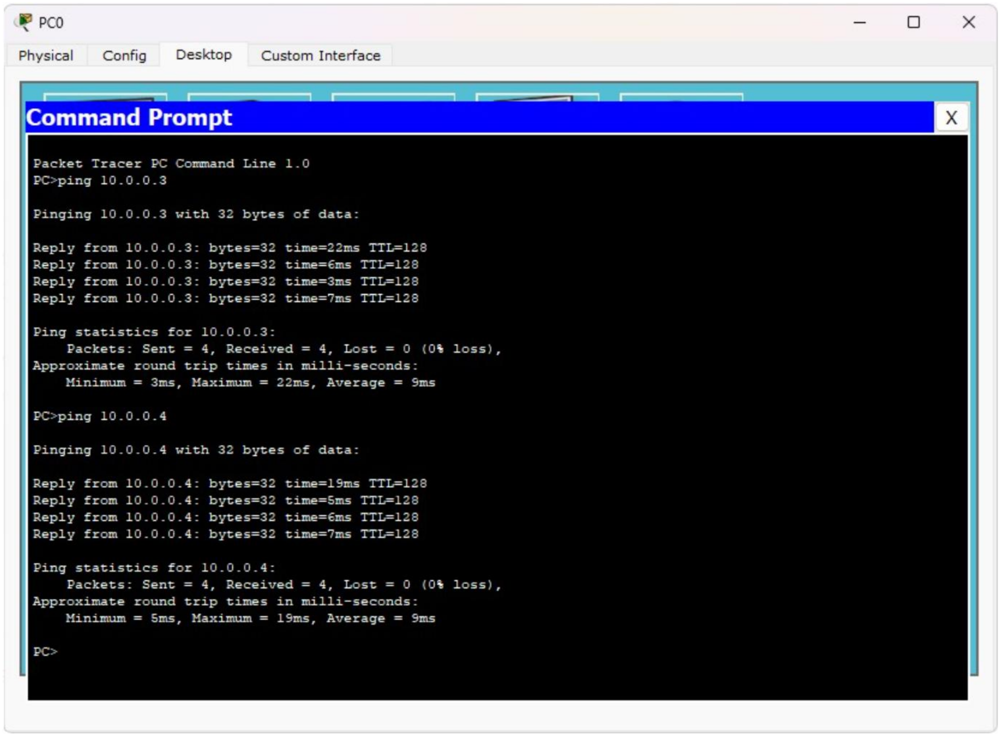
Aim of the program:
To construct a WLAN and make the nodes communicate wirelessly

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 :



```
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=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms 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 = 3ms, Maximum = 22ms, Average = 9ms

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=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms 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 = 5ms, Maximum = 19ms, Average = 9ms

PC>
```

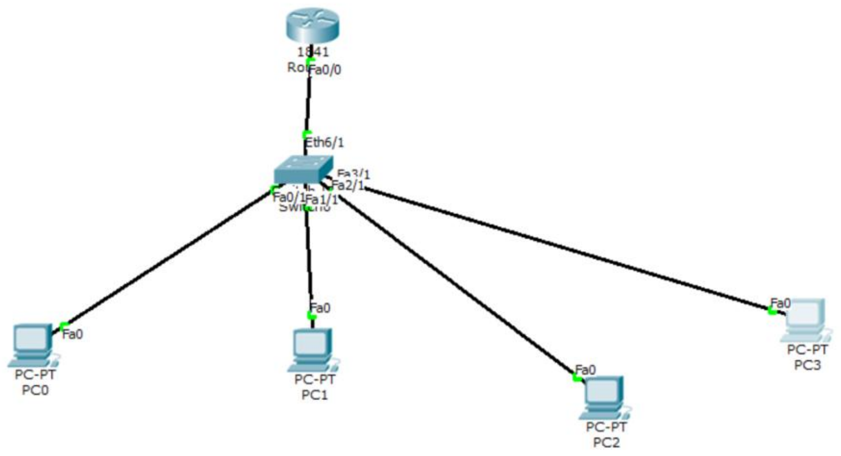
Observations:

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

<u>Program 13</u>

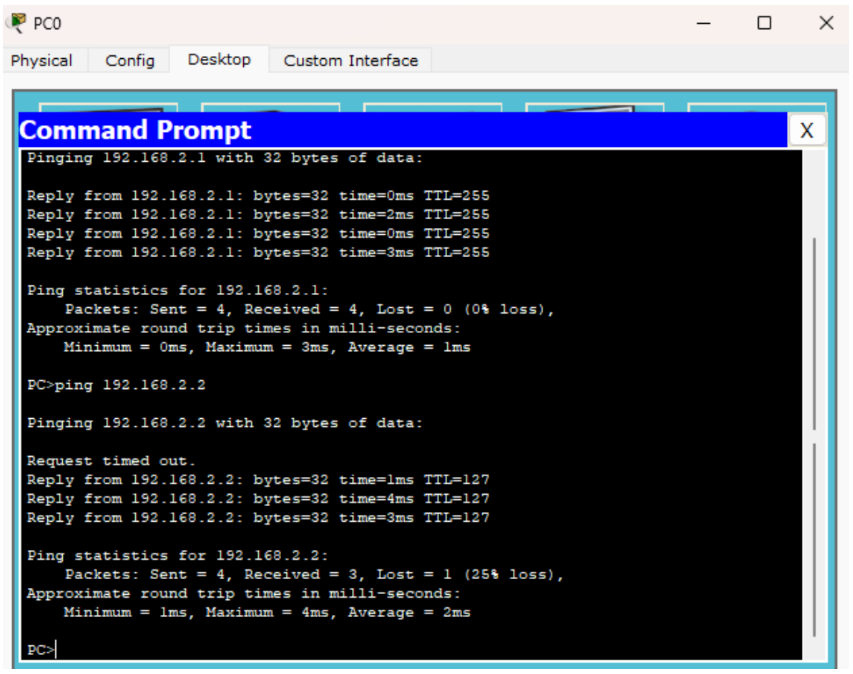
Aim of the program:
To construct a VLAN and make the PC's communicate among a VLAN

Procedure along with the topology:

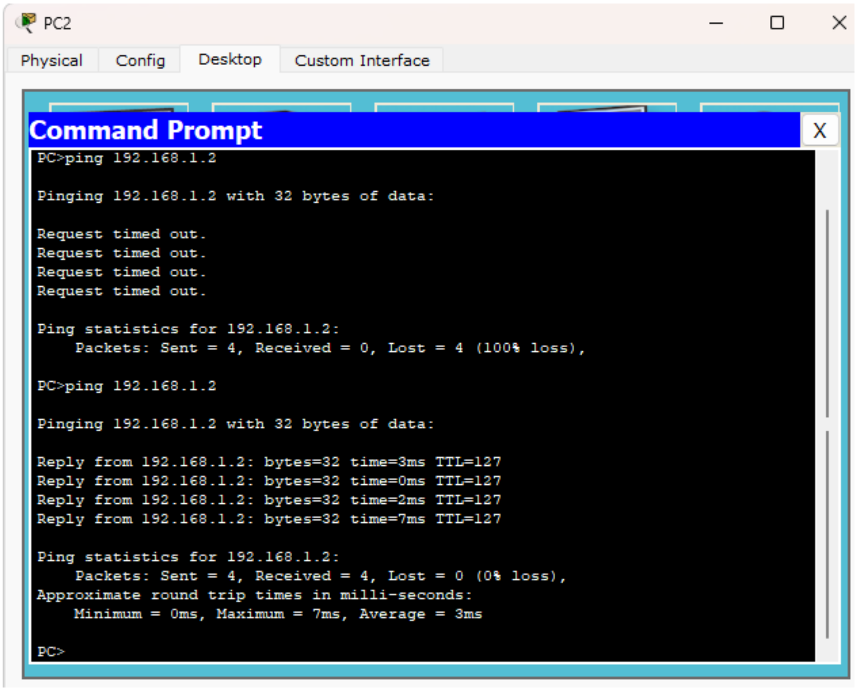


i. Create the topology seen above using router 1841
ii. Assign IPs as shown in the topology
iii. Go to the switch, choose VLAN database to configure the VLAN
iv. Give VLAN number and VLAN name add it
v. Select the interface i.e. fastethernet0/1 (near the switch from router) and make it trunk. (VLAN trunking 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

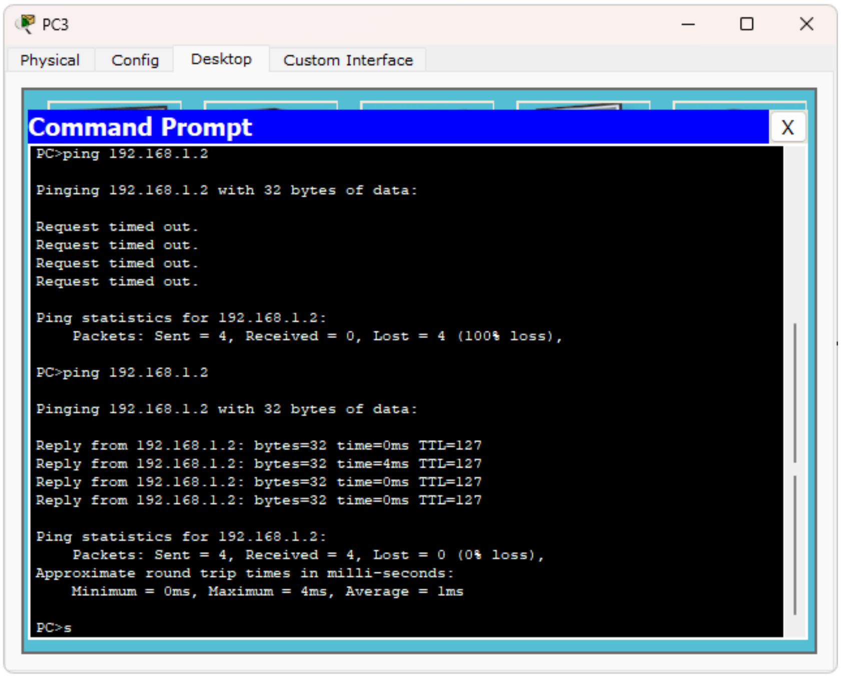
Screen shots/ output :



```
Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time=0ms TTL=255
Reply from 192.168.2.1: bytes=32 time=2ms TTL=255
Reply from 192.168.2.1: bytes=32 time=0ms TTL=255
Reply from 192.168.2.1: bytes=32 time=3ms TTL=255

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127
Reply from 192.168.2.2: bytes=32 time=3ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 4ms, Average = 2ms

PC>
```



```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=3ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=2ms TTL=127
Reply from 192.168.1.2: bytes=32 time=7ms TTL=127

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 7ms, Average = 3ms

PC>
```



```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=4ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>s
```

Observations:

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

PART B PROGRAMS

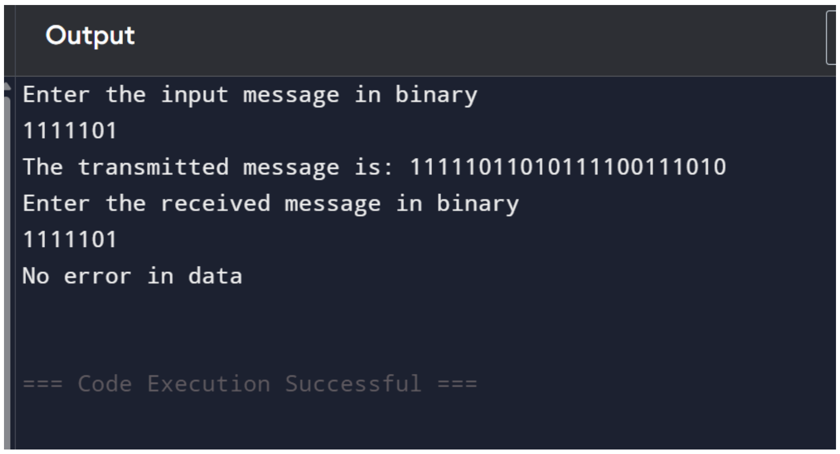
<u>Program 14</u>

Aim of the program

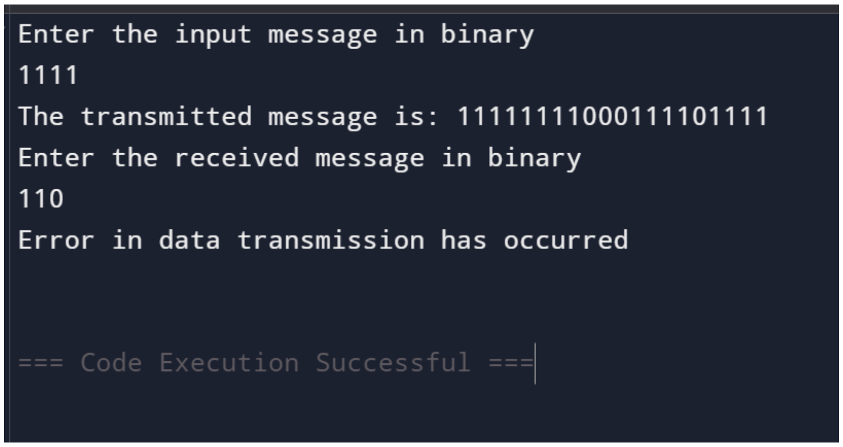
Write a program for error detecting code using CRC-CCITT (16-bits).

```
#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



```
Enter the input message in binary
1111101
The transmitted message is: 11111011010111100111010
Enter the received message in binary
1111101
No error in data


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



```
Enter the input message in binary
1111
The transmitted message is: 11111111000111101111
Enter the received message in binary
110
Error in data transmission has occurred


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

Program 15

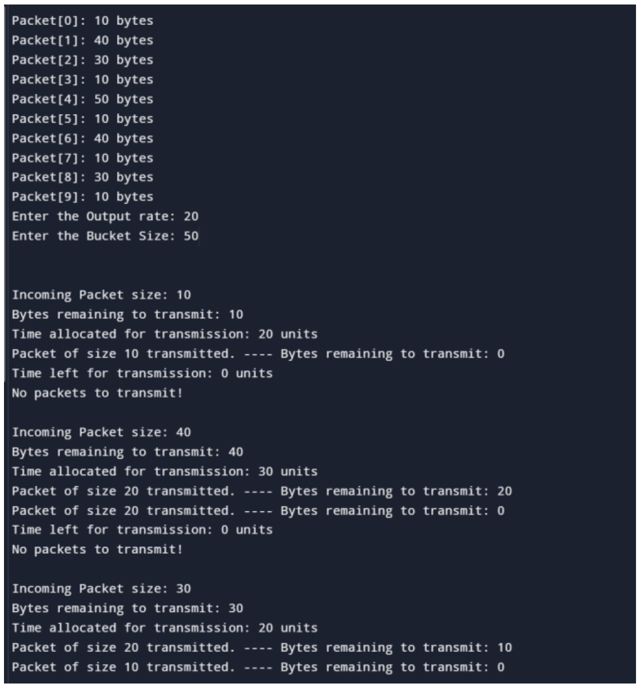
Aim of the program
Write a program for congestion control using Leaky bucket algorithm.

```
#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() {
```

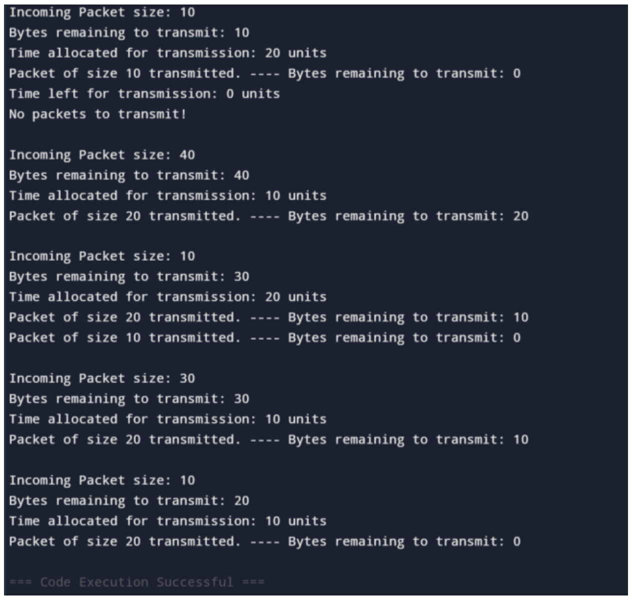
```
int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm = 0, p_time, op;

// 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 ===
```

Program 16

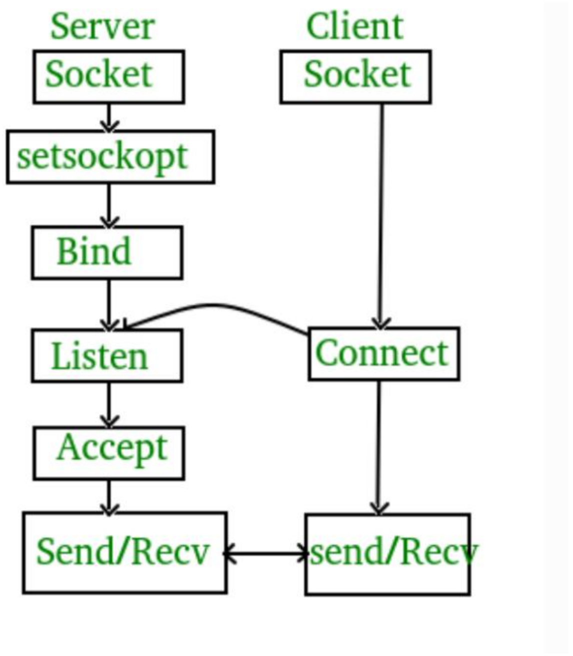
Aim of the program
Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

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;
}
```



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.

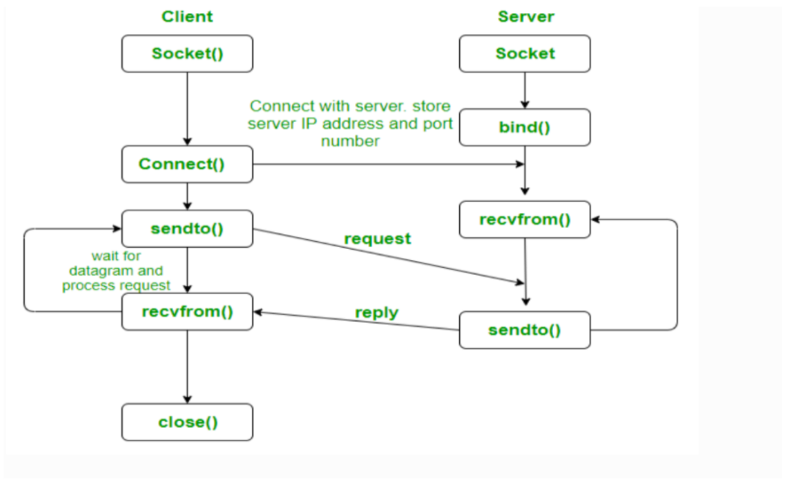
Server program

```
#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);
```


Client
Server
Socket()
Socket
Connect with server. store server IP address and port number
bind()
Connect()
recvfrom()
sendto()
request
wait for datagram and process request
recvfrom()
reply
sendto()
close()