VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

RACHANA R (1BM21CS155)

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

JUNE-SEPT
2023

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 "LAB COURSE **COMPUTER NETWORKS**" carriedout by **RACHANA R(1BM21CS155),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (22CS4PCCON)** work prescribed for the said degree.

Dr.Nandini Vineeth
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

INDEX

CYCLE 1

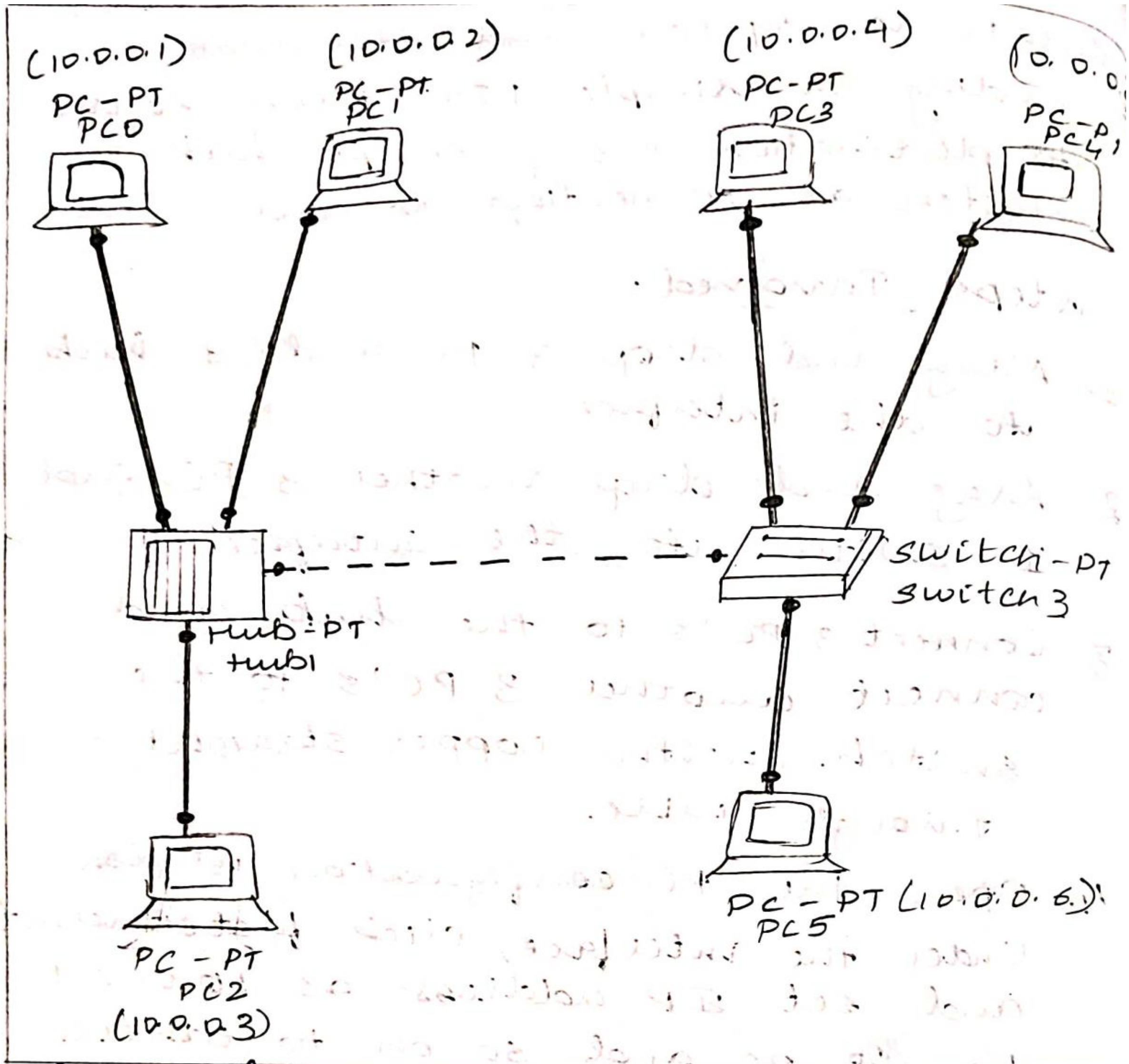
1.Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

Program - 01 16/6/23

Create a topology and stimulate sending a simple PDU from source to destination using simple hub & switch as connecting domains.

Steps Involved:

1. Drag and drop 3 PC and 1 hub to the interface.
2. Drag and drop another 3 PC and 1 switch to the interface.
3. Connect 3 PC's to the hub and connect another 3 PC's to the switch. with copper straight-through wire.
4. Open the PC configuration window Under the interface, click fastethernet and set IP address as 10.0.0.1 for 1st PC and so on to others.
5. ~~Send PDU~~ Turn on the switch, Send PDU from PC-1 ~~to~~ (10.0.0.1) to PC-2 (10.0..02) via hub.
6. Send PDU from PC-4 (10.0.0.4) to PC-5 (10.0.0.5) via switch.
7. Connect hub & switch, ~~con~~ send PDU from PC-1 (10.0.0.1) to PC-5 (10.0.0.5)



Command Prompt.

PC > ping 10.0.0.6

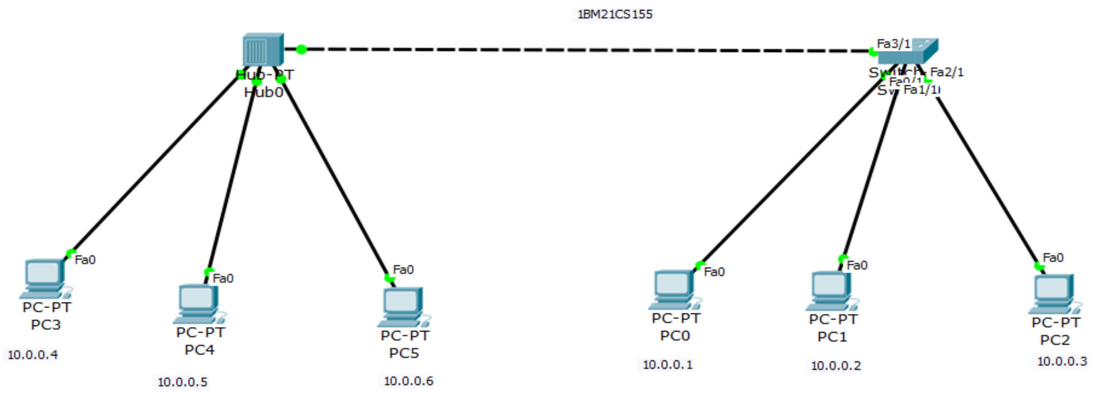
Replying 10.0.0.6 with 32 bytes of data:

10/10 Reply from 10.0.0.6 bytes = 32 time<6ms
Reply from 10.0.0.6 bytes = 32 time< 6ms
Reply from 10.0.0.6 bytes = 32 time = 6ms
Reply from 10.0.0.6 bytes = 32 time = 6 ms

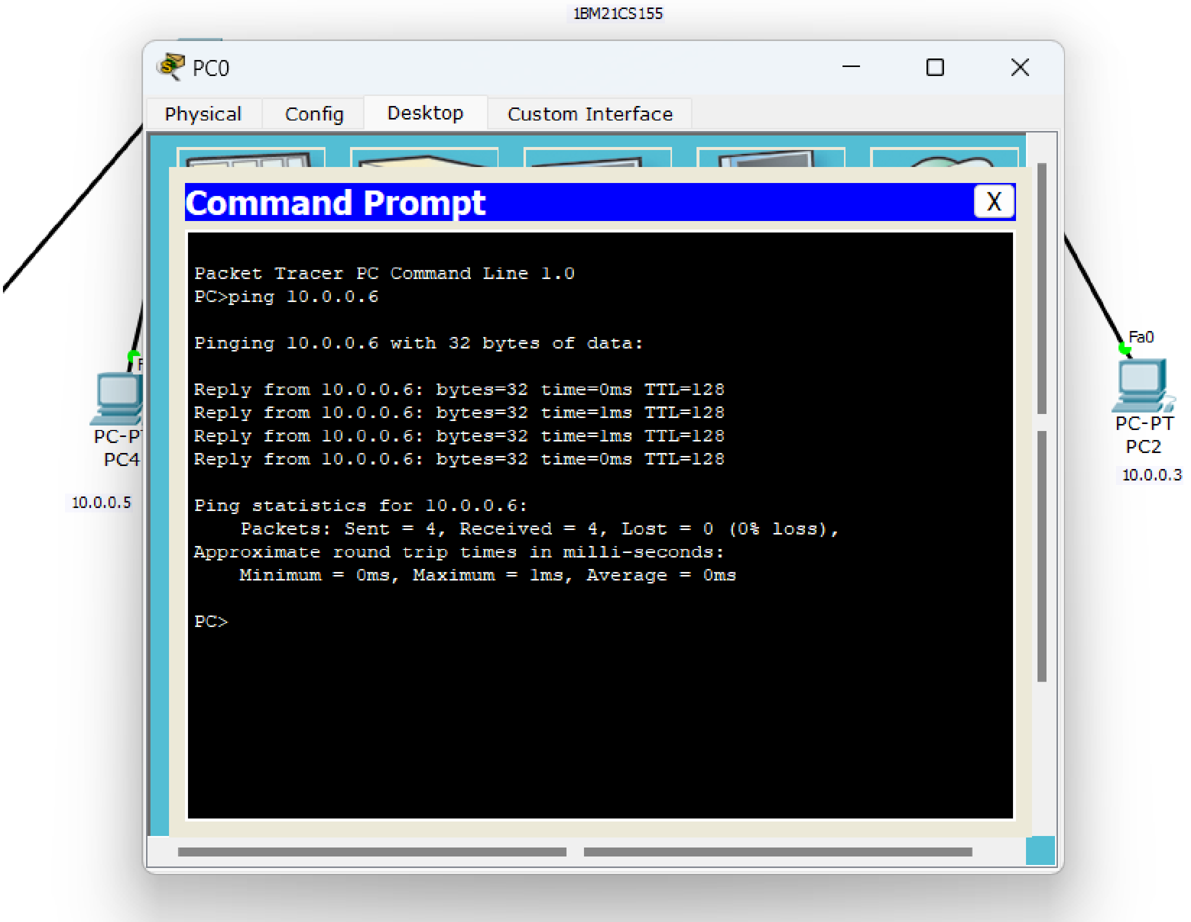
Ping statistics for 10.0.0.6.
Packets: sent 4, Received = 4, Lost = 0
Approx round trip times in ms:
min = 6ms, max = 6ms, avg: 6ms

16/8

TOPOLOGY:



OUTPUT:



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

Program - 02 28/6/23.

Create IP address to router in packet tracer. Explore the following message ping response, destination, request timed out reply.

Step 1: Place 4 PC's on workspace and 3 router on workspace

Step 2: Connect 2 PC's to a router & other 2 PC's to another & connect these ~~to~~ 2 router to another router.

Step 3: ~~Set~~ Set IP address of 1st PC as 10.0.0.1 & gateway to 10.0.0.2.

Set IP address of 2nd PC as 20.0.0.1 & gateway as 20.0.0.2.

Set IP address of 3rd PC as 30.0.0.1 & gateway as 30.0.0.2.

Set IP address of 4th PC as 40.0.0.1 & gateway as 40.0.0.2.

Step 4: In CLI of router 1:

Router > enable.

Router # config terminal

Router (config) # interface fast Ethernet 0/0.

Router (config-if) # ip address 10.0.0.3 255.0.0.0.

Router (config-if) # no shutdown.
Router (config-if) # exit.

- Repeat the same on router 2 & router 3.

step 5: Send a simple PDC from PC0 with ip-address 10.0.0.1 to PC1 with ip-address 20.0.0.1 & confirm how many packets sent by using ping command.

step6: Now if you ping from the PC with ip-address 10.0.0.1 as ping >40.0.0.1 the response will be: destination unreachable. Although it seemed there's a connection b/w 2 PC's indirectly via router but every router may not have information regarding every network present in topology so, those PC's cannot communicate. To eliminate this we should use static routing to reach every router manually

Step 7: We can do static routing for router 2 by following steps:

Router # config t
Router (config) # ip route 10.0.0.0
255.0.0.0 50.0.0.1
Router (config) # ip route 20.0.0.0
255.0.0.0 50.0.0.1

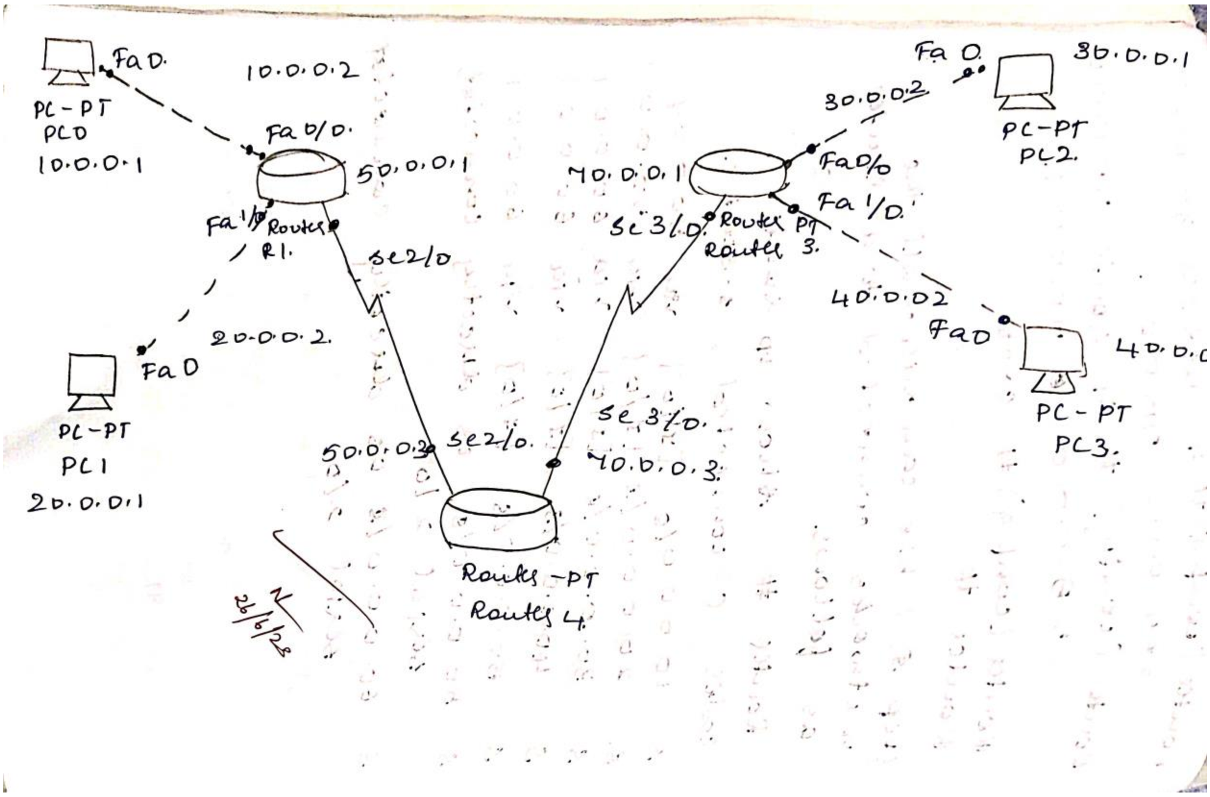
Router (config) # ip address 30.0.0.0
255.0.0.0 70.0.0.1.
Router (config) # ip route 40.0.0.0
255.0.0.0 70.0.0.1.

Router (config) # exit
Router #

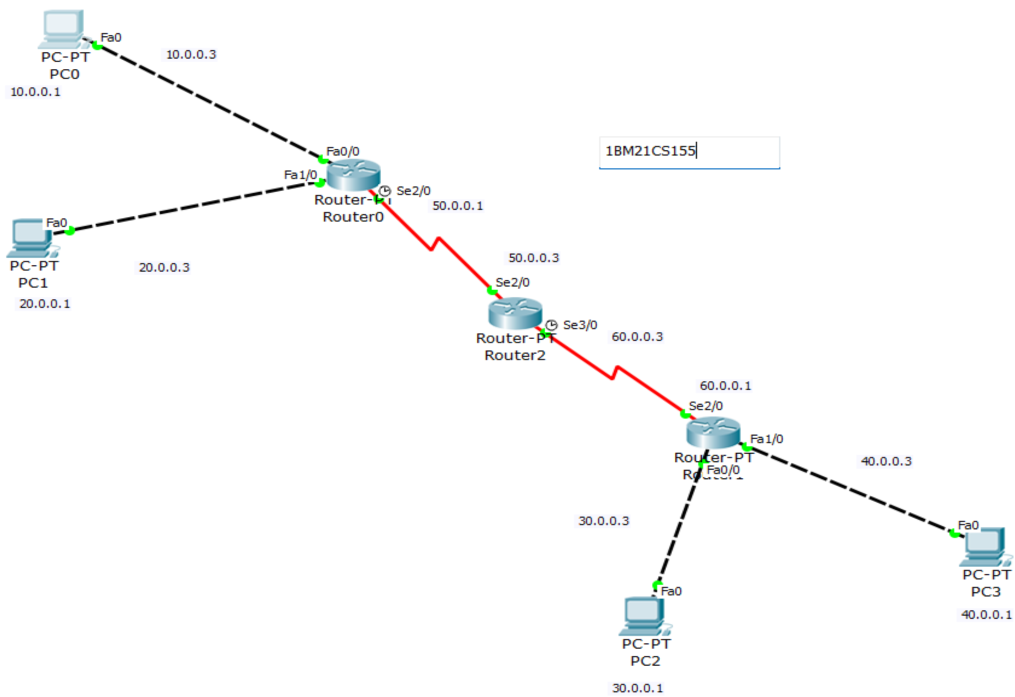
Step 8 : We can view all the networks connected to a router as follows.

Router # show ip route
codes C - connected , S - static.

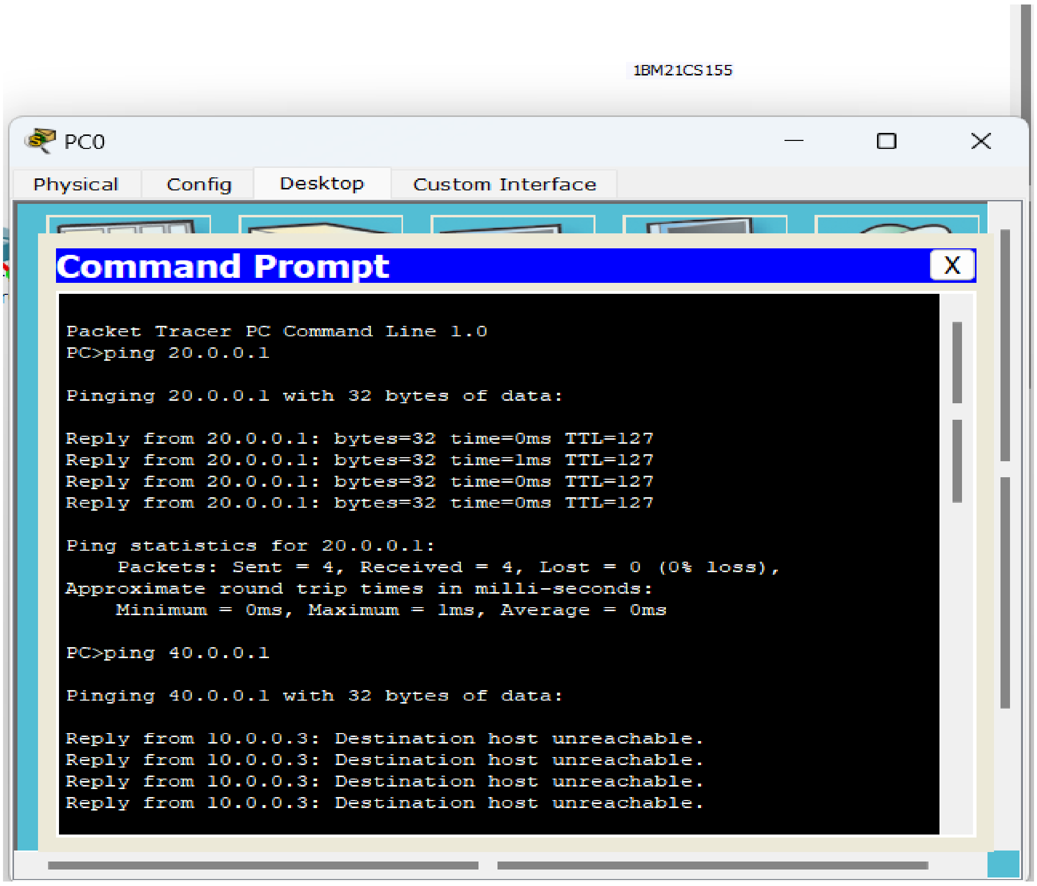
S 10.0.0.0/8 [1/0] via 50.0.0.1
S. 20.0.0.0/8 [1/0] via 50.0.0.1
S 30.0.0.0/8 [1/0] via 60.0.0.1
S 40.0.0.0/8 [1/0] via 60.0.0.1
S. 50.0.0.0/8 [1/0] via 60.0.0.1
S 50.0.0.0/8 is directly connected, serial 2/0.
S 60.0.0.0/8 is directly connected serial 2/0.

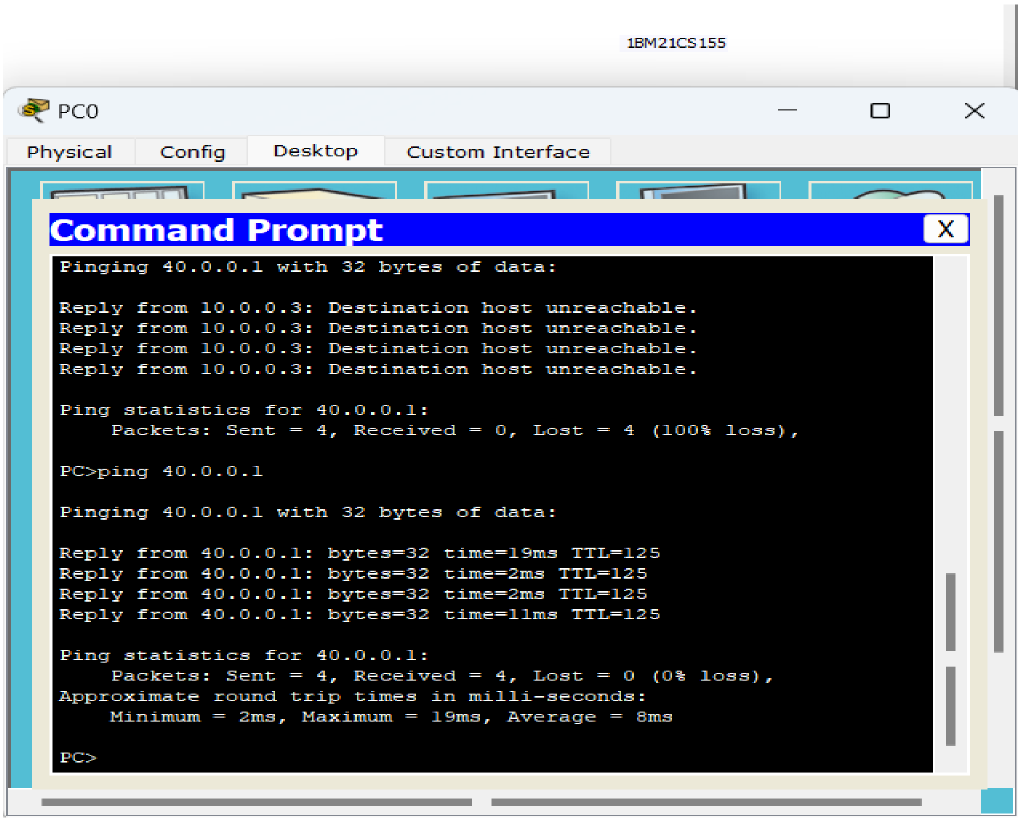

Fa 0
PC-PT
PC0
10.0.0.1
10.0.0.2
Fa 0/0
50.0.0.1
Fa 1/0
Router
R1
Se2/0
20.0.0.2
Fa 0
PC-PT
PC1
20.0.0.1
50.0.0.2
Se2/0
Se 3/0
70.0.0.3
Router-PT
Router 4
70.0.0.1
Se 3/0
Router-PT
Router 3
Fa 0/0
Fa 1/0
Fa 0
30.0.0.2
30.0.0.1
PC-PT
PC2
40.0.0.2
Fa0
PC-PT
PC3
26/6/23

TOPOLOGY


Fa0
PC-PT
PC0
10.0.0.1
10.0.0.3
Fa0/0
Fa1/0
Se2/0
Router0
50.0.0.1
Fa0
PC-PT
PC1
20.0.0.1
20.0.0.3
1BM21CS155
50.0.0.3
Se2/0
Se3/0
Router-PT
Router2
60.0.0.3
60.0.0.1
Se2/0
Fa1/0
Router-PT
Fa0/0
40.0.0.3
30.0.0.3
Fa0
PC-PT
PC3
40.0.0.1
Fa0
PC-PT
PC2
30.0.0.1

OUTPUT:


1BM21CS155
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=1ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.


1BM21CS155
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Reply from 10.0.0.3: Destination host unreachable.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=19ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=11ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 19ms, Average = 8ms
PC>

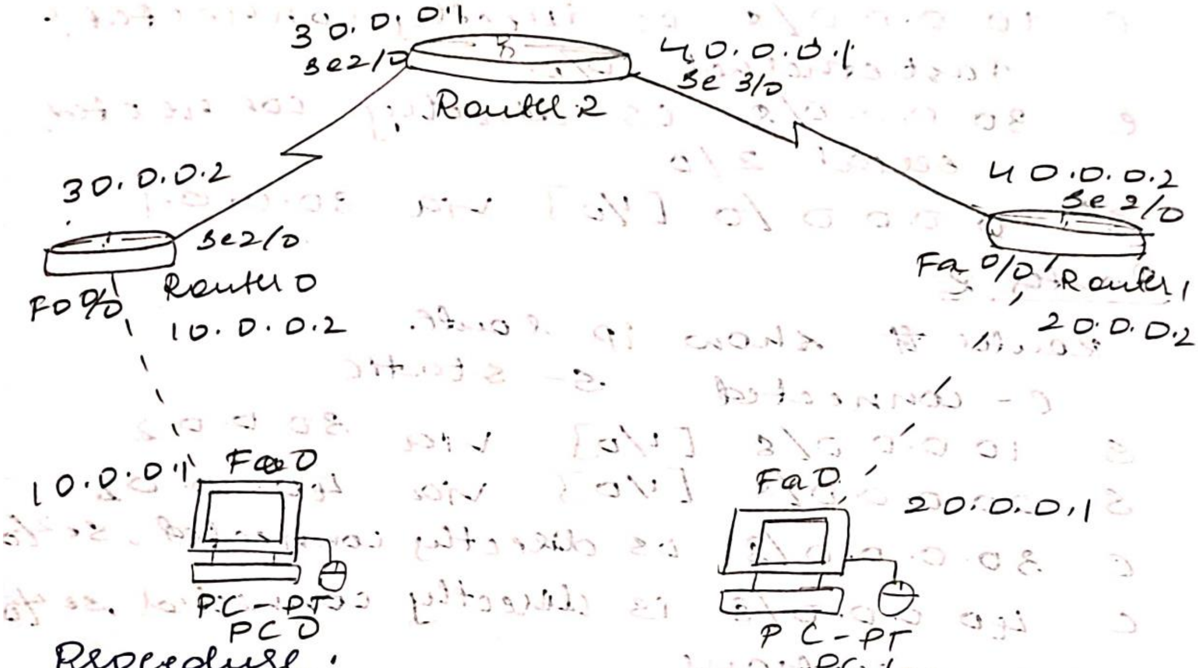
3.Configure default route, static route to the Router.

Program - 03 30/6/23

Default Routing and Static Routing.

Aim: To configure default route, static route to the router.

Topology:



Procedure:

Step1: Drag and drop 2 PC's & 3 router to the workspace & connect them as shown above.

Step2: Set IP address of 1st PC as 10.0.0.1 & 2nd PC as 20.0.0.1. Also set gateway of 2 PC as 10.0.0.2 & 20.0.0.2 respectively

Step3: Place different n/w IP address as 30.0.0.1 & 40.0.0.1 to left & right of Router-2 & start configuring the router interface for Router0.

Router 0.

Router # show ip route.

C- connected , S - static, * - candidate default.

Gateway of last route is 30.0.0.1 to network 0.0.0.0.

C 10.0.0.0/8 is directly connected, Fastethernet 0/0.

C 30.0.0.0/8 is directly connected, serial 2/0.

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

Router 2

Router # show ip route.

C- connected S- static

S 10.0.0.0/8 [1/0] via 30.0.0.2

S. 20.0.0.0/8 [1/0] via 40.0.0.2

C 30.0.0.0/8 is directly connected, se2/0

C 40.0.0.0/8 is directly connected, se2/0

ping operations

from PC0, ping PC1

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 : bytes =32 time = 2 ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time : 4ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time = 17 ms TTL = 125

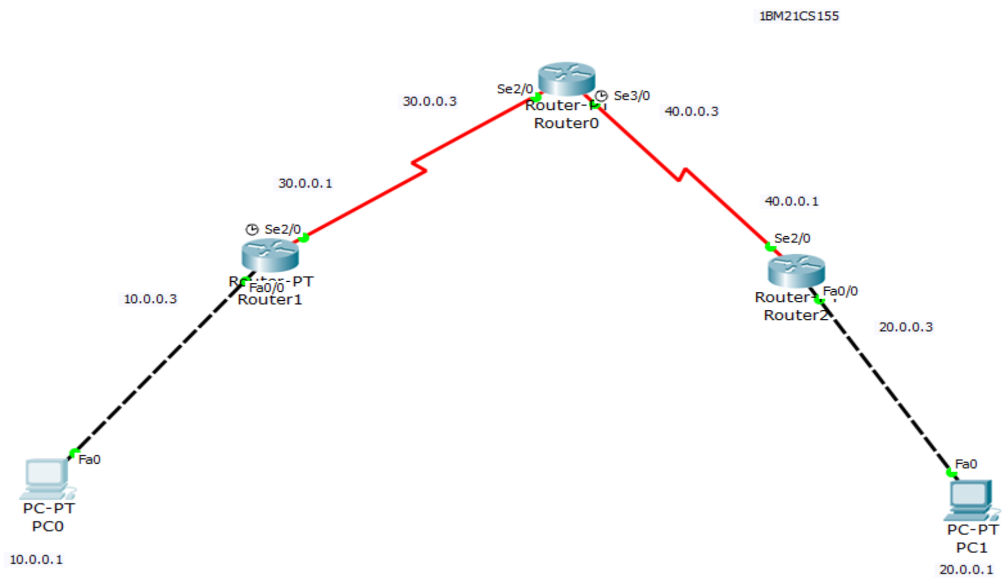
Reply from 20.0.0.1 : bytes = 32
time = 25 ms TTL = 125.

ping statistics for 20.0.0.1
packets : sent 4 , Received 4 , Lost = 0
Approx round trip time in ms.
min = 2ms max 25 ms Avg = 12ms

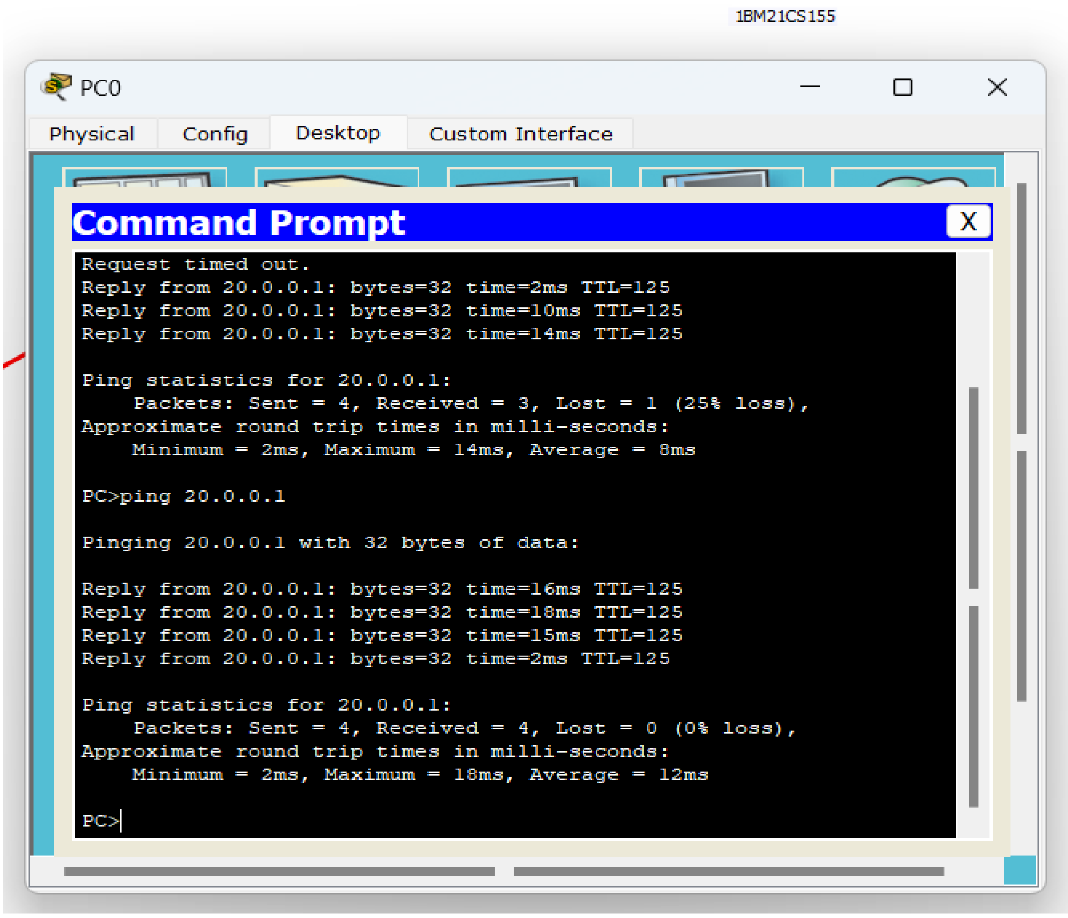
Commands ?
ip route ?

9/10
4/7/23

TOPOLOGY:



OUTPUT:



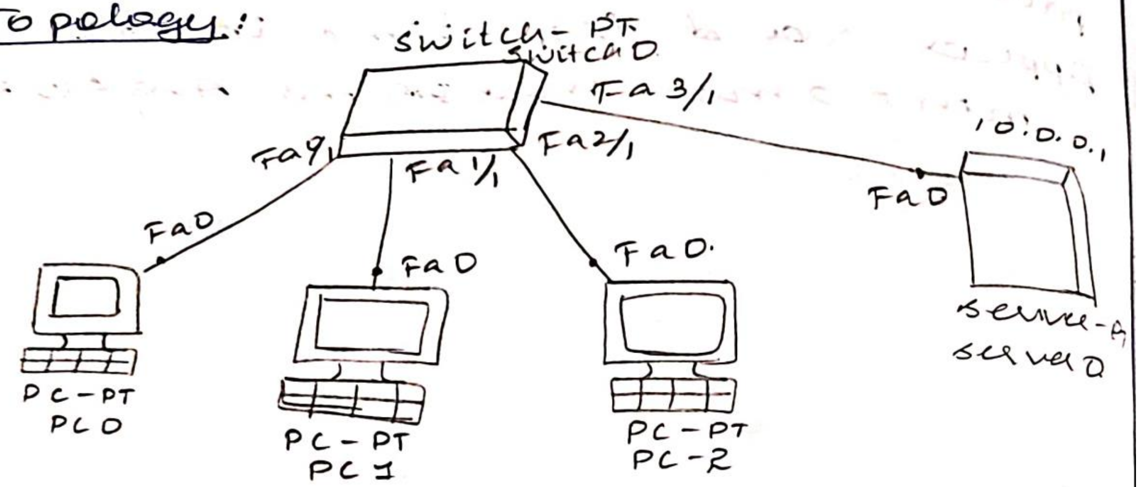
4.Configure DHCP within a LAN and outside LAN

Lab-05. 14/7/23

DHCP within LAN & outside LAN

Aim: To configure DHCP within LAN

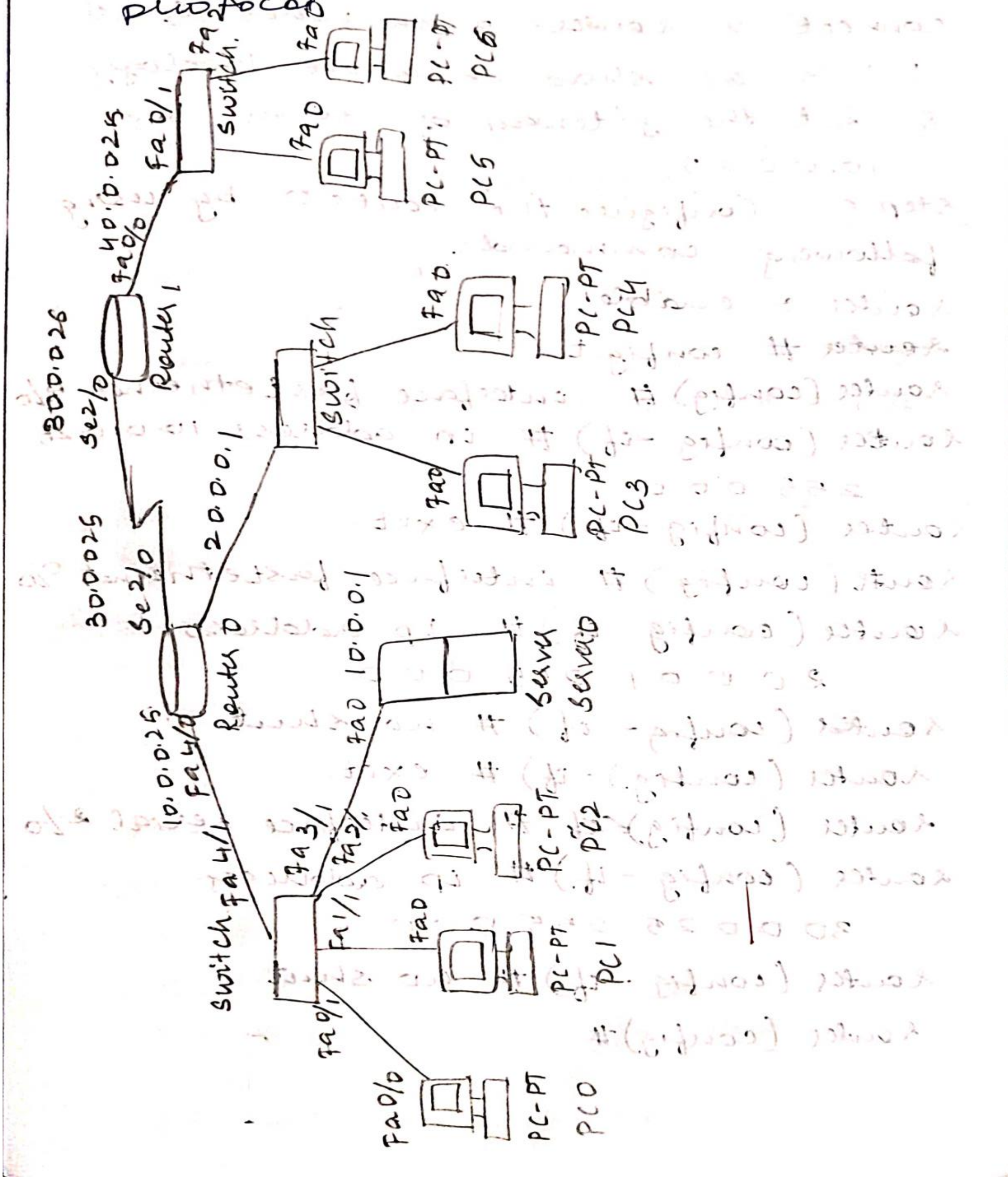
Topology:



Procedure:

1. Drag & drop 3 PC, 1 server & 1 switch to the workspace & connect them as shown in above topology
2. set the ip address of server as 10.0.0.1 and in server tab, turn on the DHCP & create a serverpool with start address as 10.0.0.2 and then save.
3. In all the PC's go to desktop → ip configuration & turn on the DHCP, the ip address will be assigned as 10.0.0.2, 10.0.0.3 etc,

These ip address are provided by the server through DHCP protocol



Procedure.

Step 4: To the previous topology, connect 2 routers, 2 switches & 5 PC's as show on above topology & set the gateway of server as 10.0.0.25.

Step 5: Configure the router 0 by using following commands.

router > enable.

router # config t.

router (config) # interface fastethernet 4/4

router. (config-if) # ip address 10.0.0.25 255.0.0.0.

router (config-if) # exit.

router (config) # interface fastethernet 0/0

router (config-if) # ip address 20.0.0.1 255.0.0.0.

router (config-if) # no shut.

router (config)-if) # exit.

router (config) ~~-if~~ # interface serial 2/0

router (config-if) # ip address 30.0.0.25 255.0.0.0

router (config-if) # no shut.

router (config)#

Step 6: Configure the router1 by the following commands.

router > enable
router # config t
router (config) # interface serial 2/0.
router (config-if) # ip address 30.0.0.26 255.0.0.0
router (config-if) # no shut.
router (config-if) # exit.
router (config) # interface fastethernet 0/0.
router (config-if) # ip address 40.0.0.25 255.0.0.0.
router (config-if) # no shut
router (config-if) # exit.

Step 7: Since router 0 knows only 10.0.0.0, 20.0.0.0 & 30.0.0.0 networks, we have to perform static routing to connect with 40.0.0.0 N/W by using following command.

router-0

router > enable
router # config t
router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.26
router (config) # ~~no~~ exit.

Step-8: For router 1 perform default routing by using following:

router 1

router > enable
router # config t
router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.25
router (config) # exit

Step-9: For router 0 & router 1 configure the ip helper address

router-0

router (config) # interface fastethernet 0/0
router (config-if) # ip helper-address 10.0.0.1
router (config-if) # exit

router 1

router (config) # interface fastethernet 0/0
router (config-if)# ip helper-address 10.0.0.1
router (config-if) # exit.

Step 10 create 2 more server pools on the server with starting address 20.0.0.2 & 40.0.0.2 & gateway 10.0.0.25

and check all the P.C's ip configuration window by setting DHCP the dynamic ip's will be assigned.

Observation

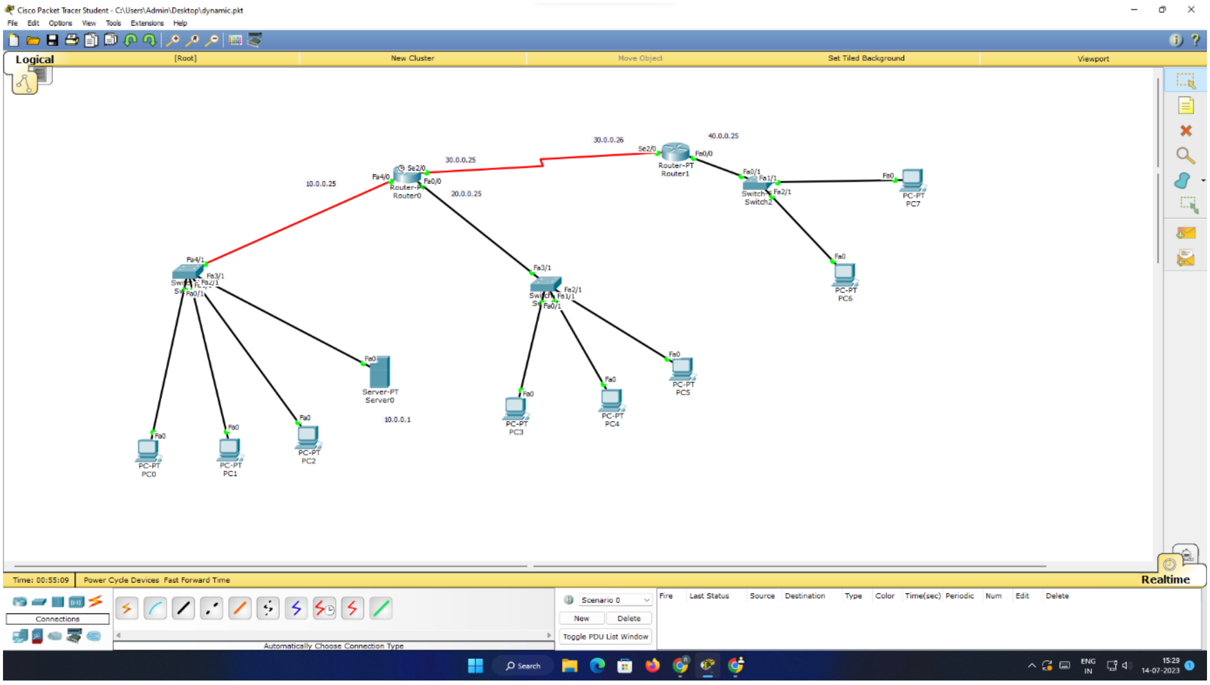
IP address will be assigned to PC's dynamically by the server through DHCP protocol.

Check def. gateway of pools.

17/7/23

10/10

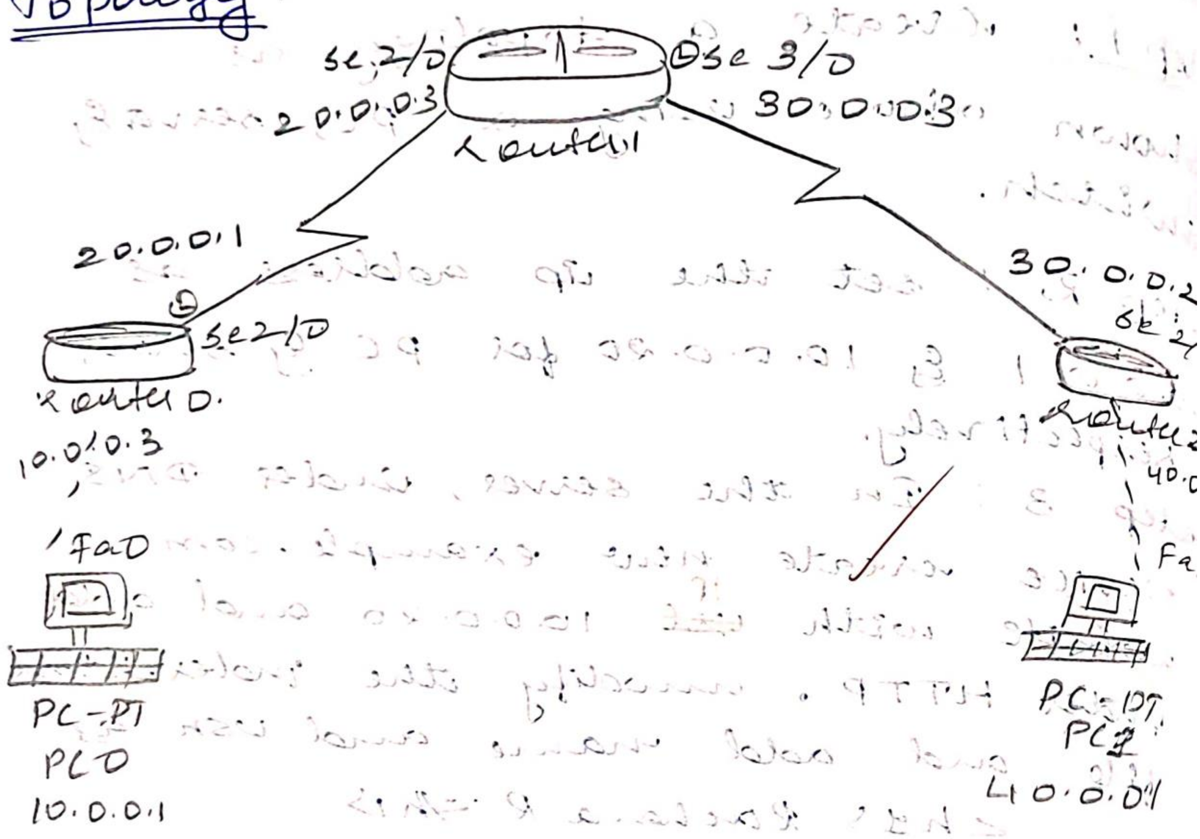
TOPOLOGY



5.Configure RIP routing Protocol in Routers.

Aim: To configure RIP routing protocol in Routers

Topology:



Procedure :

Step 1 : Create a topology using as shown above.

Step 2 : Configure the ip address of 2 PC as 10.0.0.1 & 40.0.0.1 for PC1 & PC2 respectively and set gateway as 10.0.0.3 & 40.0.0.3.

Step 3 : Plan the ip's to configure the routers for Router 0.

Router > enable

Router # config t

Router ~~#~~ (config) # interface fastethernet 0/0

Router (config - if) # ip address 10.0.0.3 255.0.0.0

router (config - if) # no shut.

router (config - if) # ip interface serial 2/0

router (config - if) # ip address 20.0.0.1 255.0.0.0

router (config - if) # no shut.

Similarly configure the ports of Router 1 & Router 2.

Step 4: For router-0

router (config) # interface serial 2/0
router (config-if)# ~~ip address~~ encapsulation PPP
router (config-if) # no shut
router (config-if) # exit
repeat this for router-1 interface → serial 2/0 & 3/0, router-2 → serial 2/0

step 5: For router 0 (serial 2/0) and router 1 (serial 3/0)

router (config) # interface serial 2/0
router (config) # clock rate 64000
router (config-if) # no shut
router (config) # exit

Step 6: For all the 3 routers, repeat this step.

Eg: Router - 0

router > enable
router # config t
router (config) # ~~interface~~ router rip
router (config-router) # network 10.0.0.0
router (config-router) # network 20.0.0.0

Similarly do this for Router 1 & 2 — no shut/ exit
then router # show ip route
This will result in saying that every router knows all 4 network in topology. Now you can ping from

PC1 to PC2

<u>Result</u>: In command prompt of PC1

PC> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data.

Reply from 40.0.0.1 : bytes = 32 t = 12 ms TTL = 12

Reply from 40.0.0.1 bytes = 32 t = 12 ms TTL = 12

Reply from 40.0.0.1 bytes = 32 t = 12 ms TTL = 125

Reply from 40.0.0.1 bytes = 32 t = 12 ms TTL = 125

ping statistics for 40.0.0.1

packets: sent = 4, received = 4, lost = 0 (0% loss)

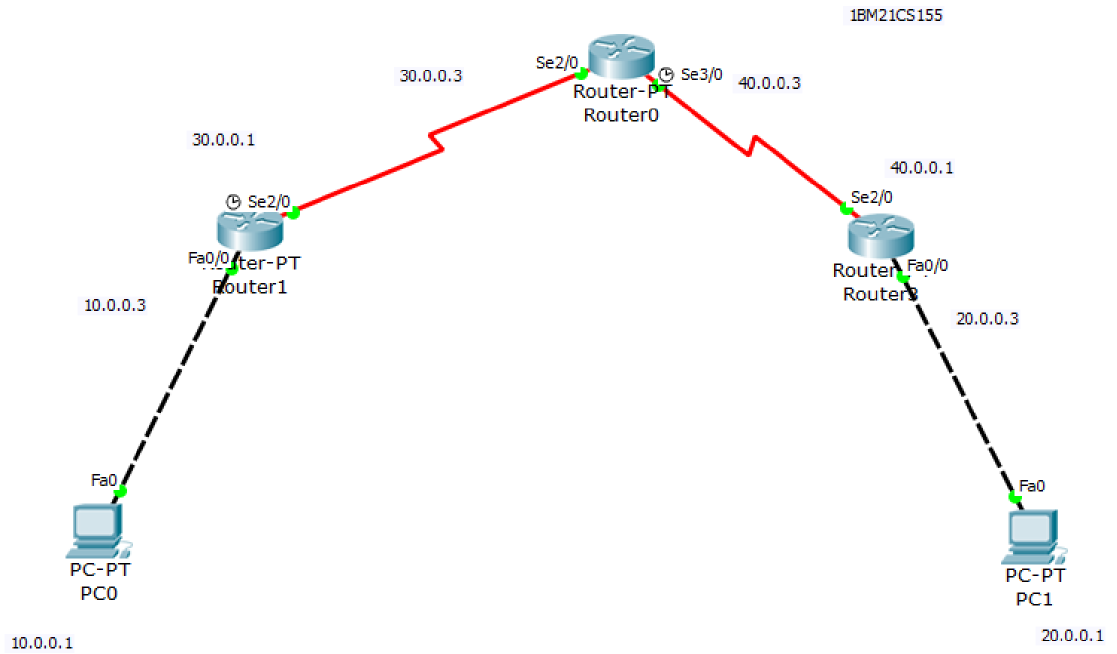
approx round trip times in millisecond

min = 2ms, max = 12ms, avg = 6 ms.

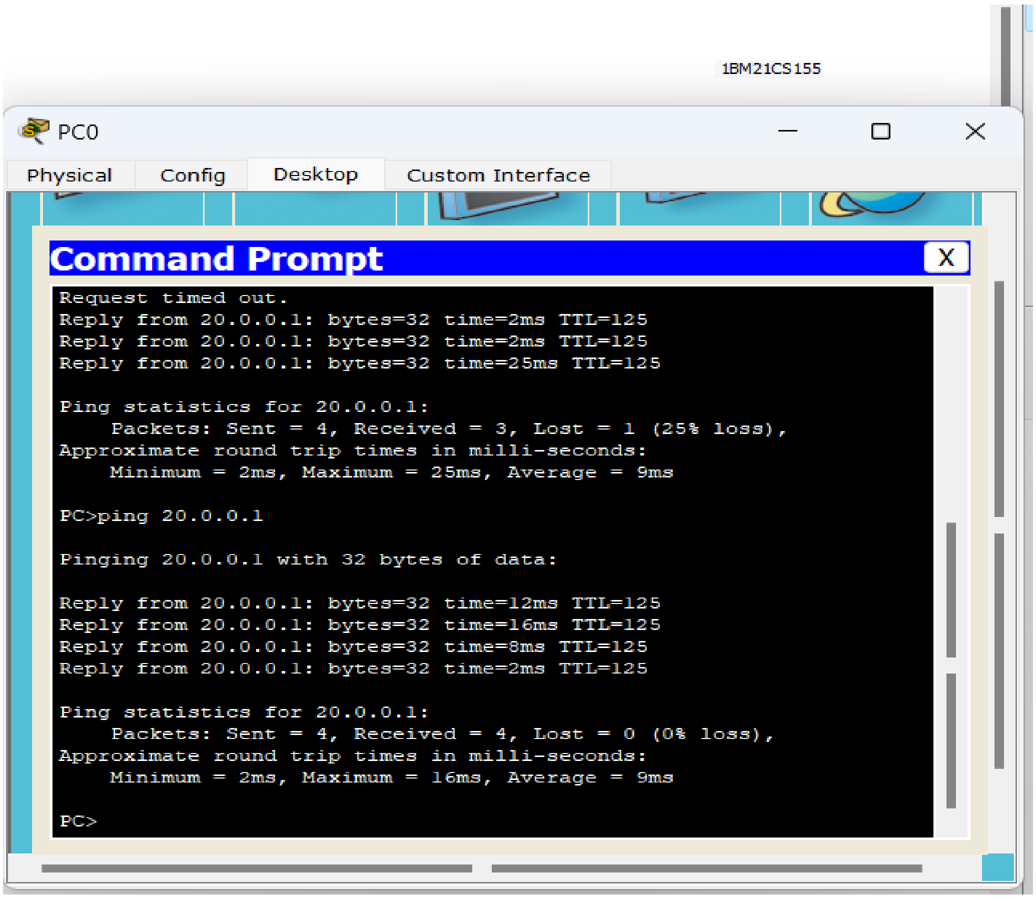
10/10

24/7/23

TOPOLOGY:



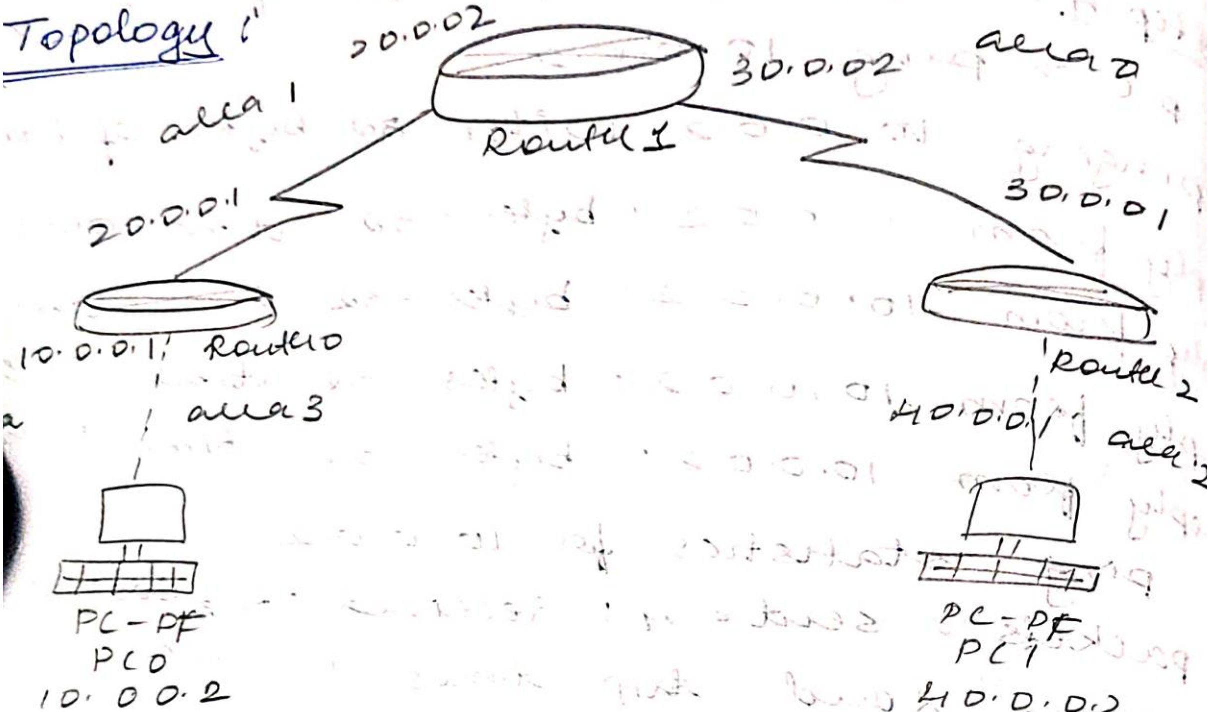
OUTPUT:



6. Configure OSPF routing protocol.

Aim: To configure OSPF routing protocol and connect areas

Topology :



Procedure

Step1: create a topology as shown using 2 PC & a router.

Step2: Configure ip address and gateway for PC as 10.0.0.2 & 10.0.0.1 for PC0 and 40.0.0.2 & 40.0.0.1 for PC1 resp.

Step 3: configure ip address to all router interface.

Router R0,

R0 (config)# interface fastethernet 0/0.

R0 (config-if)# ip address 10.0.0.1 255.0.0.0.

R0 (config-if)# no shut.

R0 (config)# interface serial 2/0.

R0 (config-if)# ip address 20.0.0.1 255.0.0.0

R0 (config-if)# encapsulation PPP.

R0 (config-if)# clock rate 64000.

R0 (config-if)# no shut.

R0 (config-if)# exit.

similarly, configure for R1 & R2.

Step 4: Now, enable ip routing by configuring OSPF routing protocol in all routers.

Router R0,

Router (config)# router ospf 1

Router (config-router)# router-id 1.1.1.1

router (config-router)# network 10.0.0.0 0.255.255.255. area 3.

router (config-router)# network 20.0.0.0 0.255.255.255 area 1

router (config-router)# exit.

lllly configure for R1 & R2

Step 5 : Now check routing table fo

Router # show ip route.

C - connected

O - ospf

C 10.0.0.0/8 is directly connected, Fa0/

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

O IA 40.0.0.0 /8 via 20.0.0.2 , 00:04:23
se 2/0.

O IA 30.0.0.0/8 via 20.0.0.2 , 00:07:29

Here R1 knows area 0 . Network 20.0.0.0 connected to R1 from R0, so R0 learns network through this network

Router (config) # router ospf 1
1 => process id (1 - 65535)

There must be one interface up to keep ospf process up. so its better to configure loopback address to keep OSPF process up.
so its better to configure loopback address to router. It is a virtual interface never goes down once we configured.

R0 (config-if) # interface loopback 0

R0 (config-if) # ip add 172.16.1.252
255.255.0.0

Step 6 :

Now check routing table for R3.

R3 # show ip route.

codes : O - OSPF   C : connected

O IA 20.0.0.0/8 via 30.0.0.2,
00:18:58 , se 2/0

C 40.0.0.0/8 directly connected,
fastethernet 0/0.

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

Here R3 doesn't known about the area 3 so we have to create virtual link b/w R0 & R1

Step 7 : create virtual link b/w R0, R1 by this we create a virtual link to connect area 3 & 0.

In R0

R0 (config)# router ospf 1

R0 (config-router)# area 1 virtual link 2.2.2.2

In R1

R1 (config)# router ospf 1

R1 (config-router)# area 1 virtual link 1.1.1.1

Step 8 : R1 & R2 get updates about area 3. Now check routing table for R2.

R2 # show ip route.

codes : O - ospf C - connected.

O IA 20.0.0.0/8 via 30.0.0.2 , 00:01:06, ...

C 40.0.0.0/8 is directly connected.
fastethernet 0/0.

O IA 10.0.0.0/8 via 30.0.0.2 , 00:01:56
se 2/0

Step 9 ping PC 1 from PC0,

Result

In PC0.

PC > ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2 : bytes = 32 time = 2ms
Reply from 40.0.0.2 : bytes = 32 time : 10ms
Reply from 40.0.0.2 : bytes = 32 time : 14ms
Reply from 40.0.0.2 : bytes = 32 time : 2ms

ping statistics for 40.0.0.2

packets : sent : 4 , received = 4, lost = 0

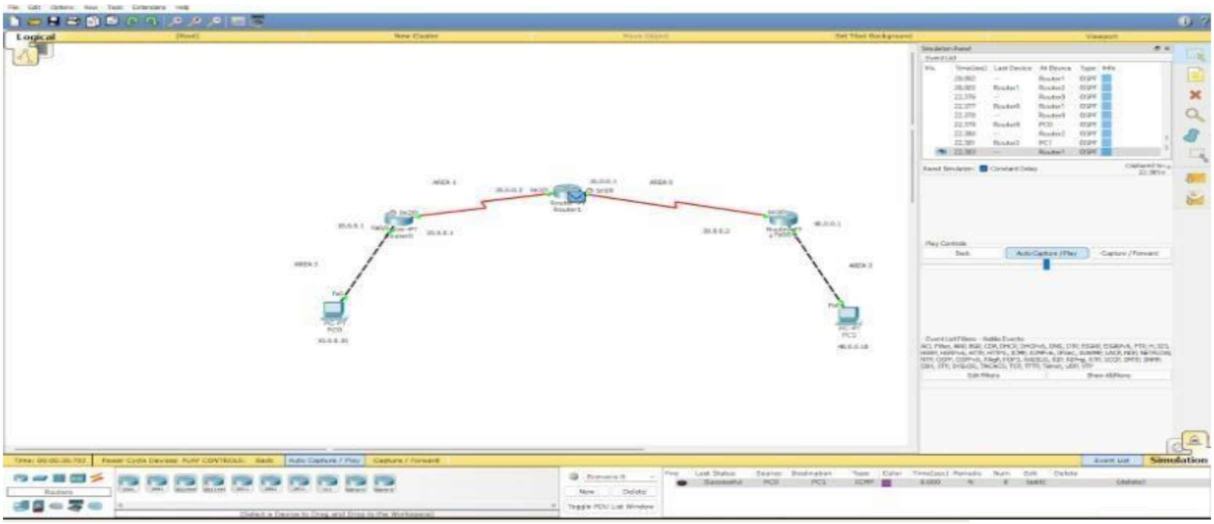
approx round trip times in ms.

min : 2ms , max = 14 ms , avg = 7ms.

10/10

28/8/23

TOPOLOGY:



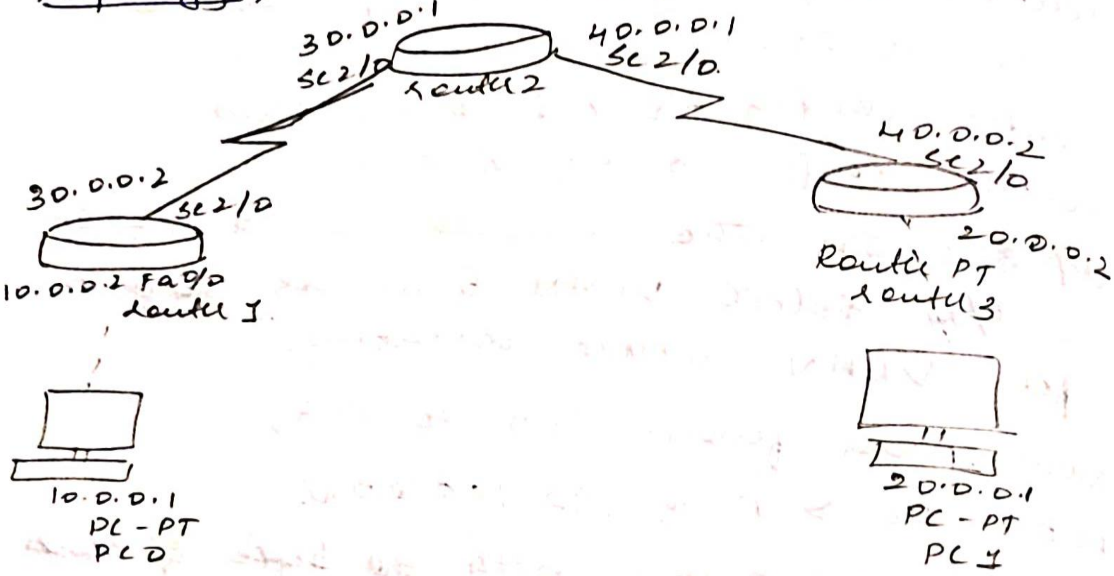
```
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 2ms, Average = 2ms

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=10ms TTL=125
Reply from 40.0.0.2: bytes=32 time=14ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 14ms, Average = 7ms
```

7. Demonstrate the TTL/ Life of a Packet.

Lab 7

Aim: To demostrate the TTL / life of a packet :

Topology :



Procedure :

Step1 : create a topology with 2 PC & 3 router as shown above

Step 2 : Configure the IP address as 10.0.0.1 & 20.0.0.1 for PC0 & PC1

Step3 : configure the IP add for routers and static /default routing

Router0 :

router # config t

router (config)# interface fastethernet 0/0

router (config-if)# ip address 10.0.0.2 255.0.0.0

router (config-if) # no shut.
router (config-if) # exit.
router (config-if) # interface serial 2/0
router (config-if) # ip address 30.0.0.2
255.0.0.0.
router (config-if) # no shut.
router (config-if) # exit.
router (config) # ip route 0.0.0.0 0.0.0.0
30.0.0.1.
router (config) # exit.
similarly, configure for router 1 & 2.

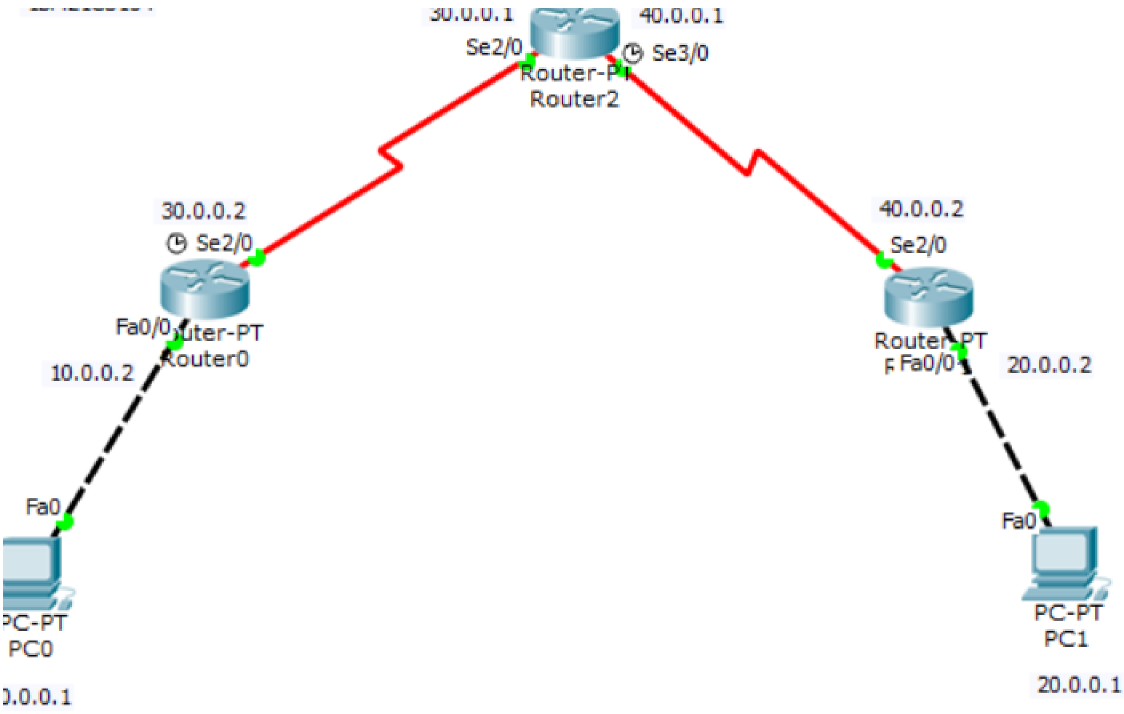
Step 4: In simulation mode, send a simple PDU from one PC to another.

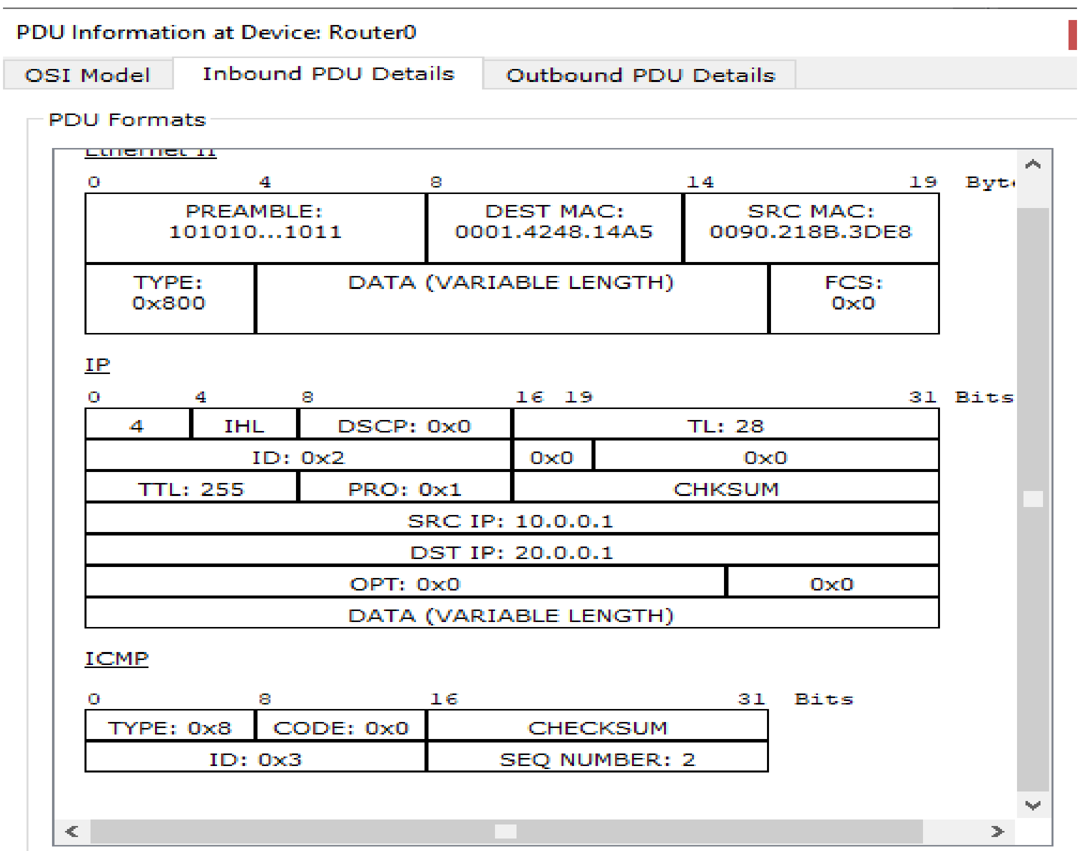
Step 5: click on PDU during every transfer to see the inbound & outbound PDU details, use capture buttons to capture everything.

Observation: 9/10

28/8/23

TOPOLOGY:

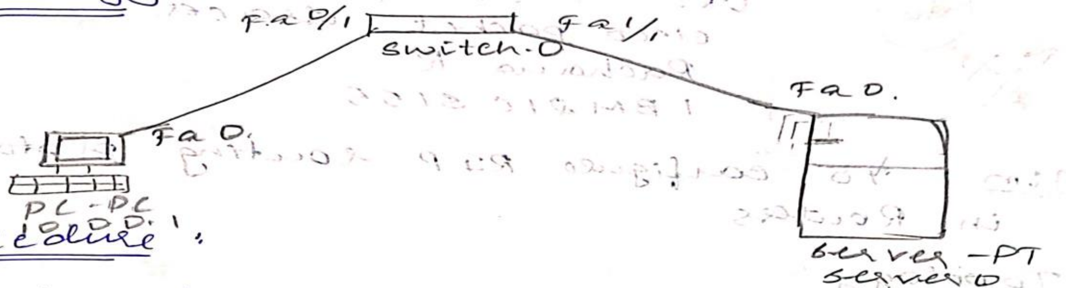

30.0.0.1
40.0.0.1
Se2/0
Se3/0
Router-PT
Router2
30.0.0.2
Se2/0
Fa0/0
Router0
10.0.0.2
40.0.0.2
Se2/0
Fa0/0
20.0.0.2
Fa0
PC0
Fa0
PC-PT
PC1
20.0.0.1


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
0 4 8 14 19
PREAMBLE: 101010...1011
DEST MAC: 0001.4248.14A5
SRC MAC: 0090.218B.3DE8
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x2
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 20.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x3
SEQ NUMBER: 2

8. Configure Web Server, DNS within a LAN.

Aim: To configure web server DNS, within a LAN.

Topology:



Procedure:

Step 1: Create a topology as shown above using a PC, server & switch.

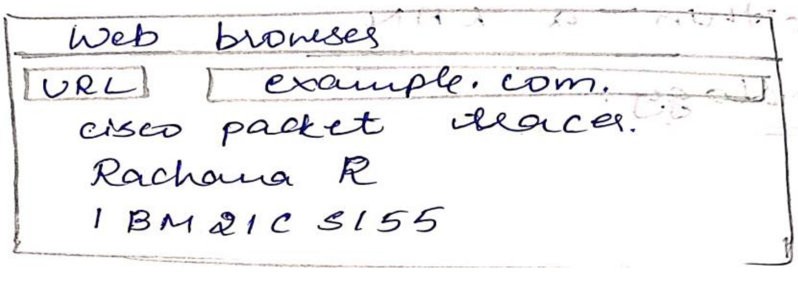
Step 2: set the ip address as 10.0.0.1 & 10.0.0.20 for PC & server respectively.

Step 3: In the server, under DNS service create new example.com website with ~~url~~ IP 10.0.0.20 and add under HTTP, modify the index.html file and add name and usn as

<h1> Rachana.R </h1>
<h2> 1BM21CS155 </h2>

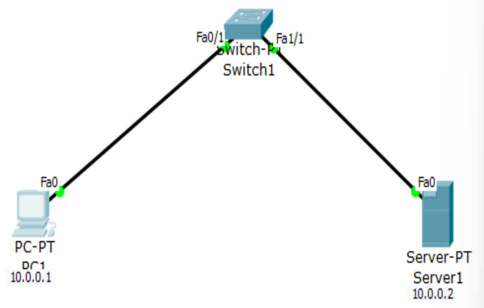
Step 4: In PC0, go to desktop → web browser and type example.com. You will be able to see the website with entered name & USN

Result

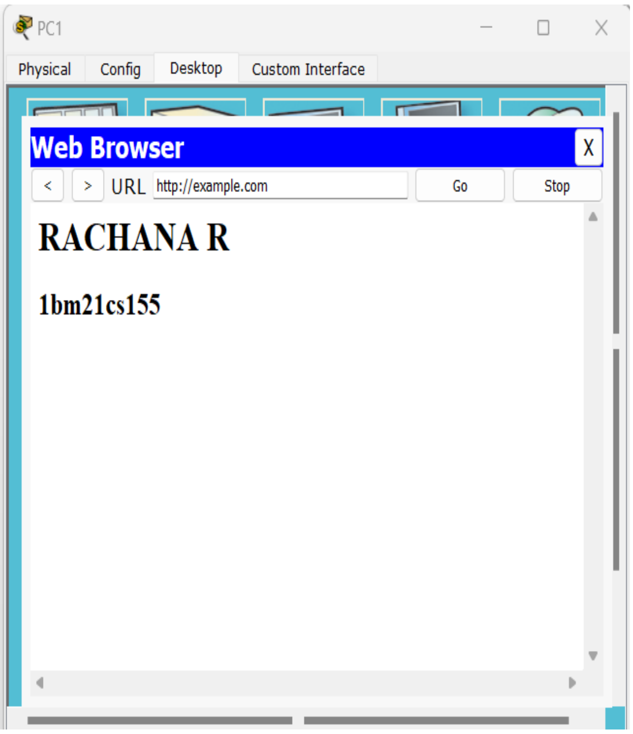


10/10
24/7/23

TOPOLOGY:



Fa0/1
Fa1/1
Switch-PT
Switch1
Fa0
PC-PT
PC1
10.0.0.1
Fa0
Server-PT
Server1
10.0.0.2



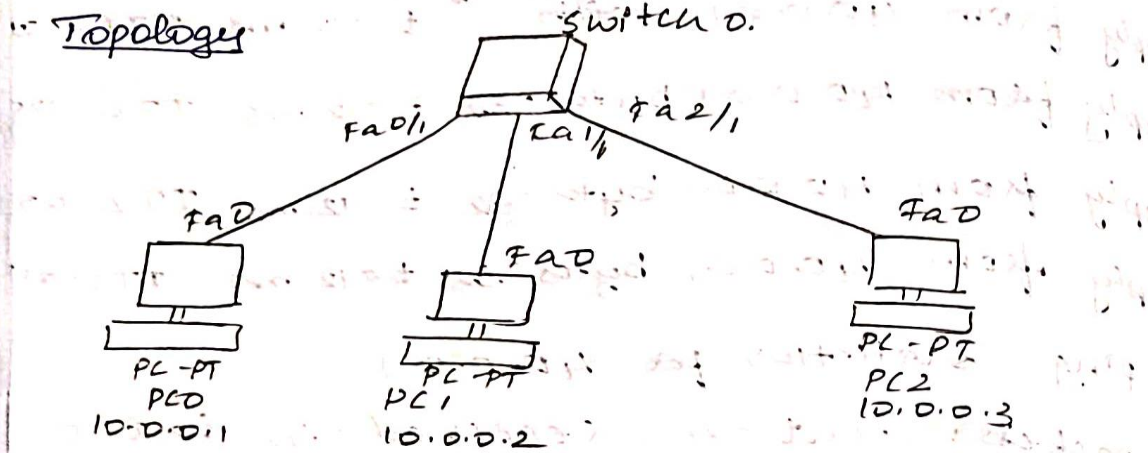
PC1
Physical
Config
Desktop
Custom Interface
Web Browser
X
URL
http://example.com
Go
Stop
RACHANA R
1bm21cs155

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

Lab 7

Aim: To construct a simple LAN and understand the concept of address resolution protocol (ARP)

Topology



Procedure:

Step 1: Drag & drop 3 PC & 1 switch to the workspace and connect them according to the topology.

Step 2: Configure the IP address of PC as 10.0.0.1, 10.0.0.2, 10.0.0.3 for PC0, PC1 & PC2

Step 3: Now in command prompt of PC0, ~~if~~ we run the command "arp -a" Initially arp table will be empty

Step 4: Also in CLI of switch, the command - show mac address table can be given on every transaction to see how the switch learns from transactions & build the address table.

Initally all tables are empty.

Step 5 : Now ping from PC0 to PC1

PC > ping 10.0.0.3.

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3 bytes = 32 times = 0ms.

Reply from 10.0.0.3 : bytes = 32 time = 0ms

Reply from 10.0.0.3 : bytes = 32 time = 0ms.

Reply from 10.0.0.3 : bytes = 32 time = 0ms.

ping statistics for 10.0.0.3.

packets : sent : 4, received = 4, lost = 0.

approx round-trip times in milli-seconds

min = 0ms , max = 0ms , avg = 0 ms.

Step 6 : Run "arp-a" command again in PC0.

PC > arp -a.

Internet address.	Physical address	Type
10.0.0.3.	0090.2176.1580	Dynamic.

Similarly ping PC2 from PC0 & re-run arp-a command.

~~PC~~ > arp-a

Internet add	Physical add	Type
10.0.0.3.	0090.2176.1580	dynamic
10.0.0.2	0060.5026.9350.	dynamic

PC > arp - d.

PC > arp - a

No arp entries found.

"arp - d" cmd is used to clear table.

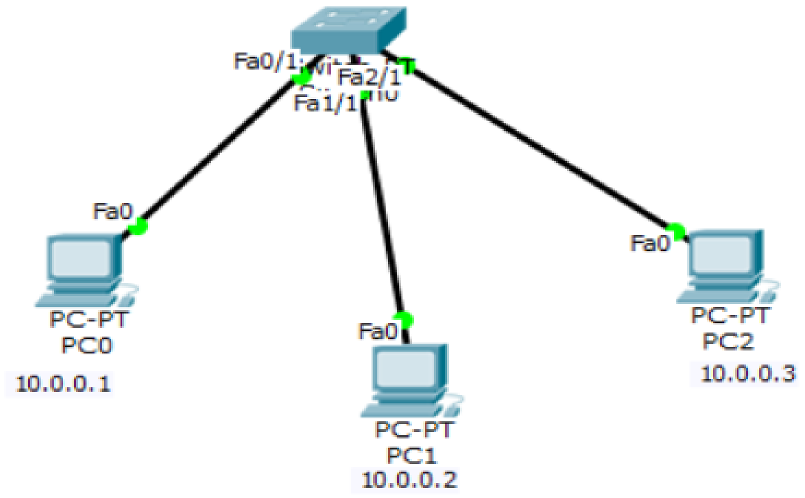
Observation:

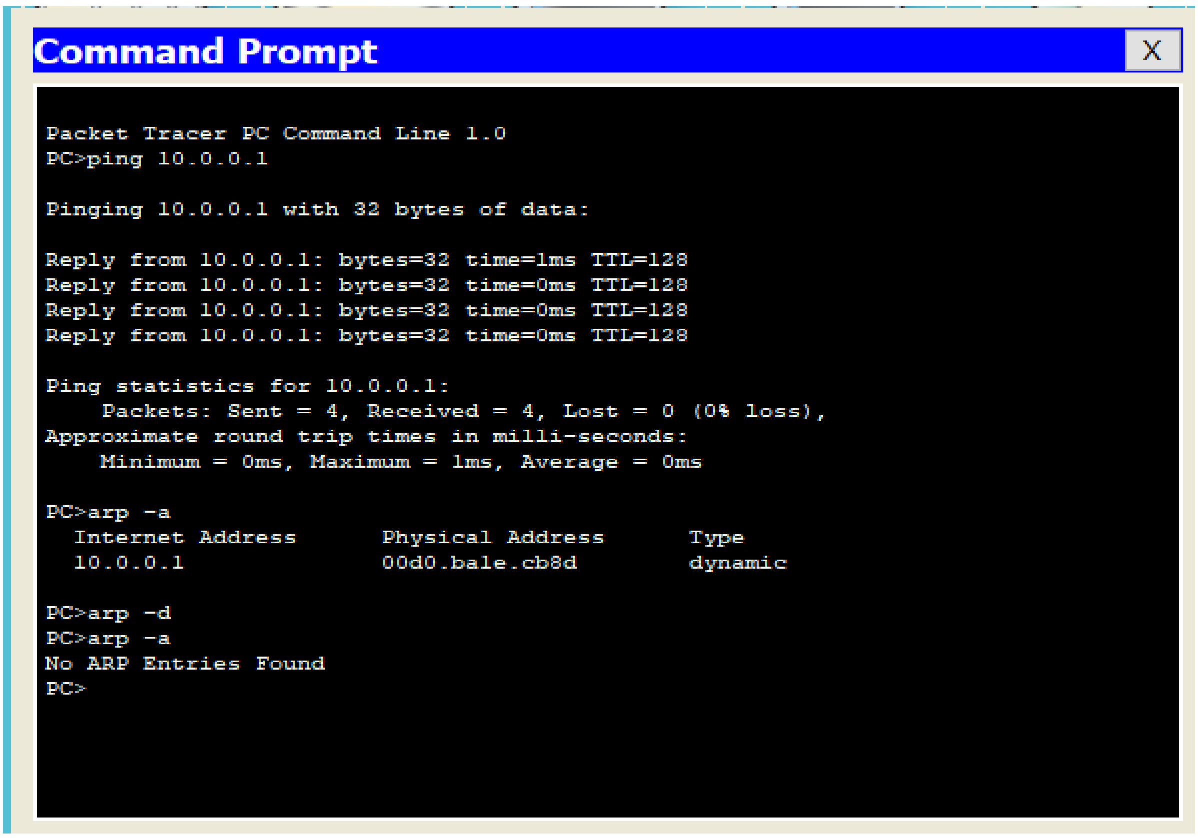
By using ARP protocol, Physical / MAC address of each device will get stored in table, ~~not~~ dynamically whenever there is a new transact, with the help of this table, switch performs forwarding of packet

10/10

28/8/23

TOPOLOGY:





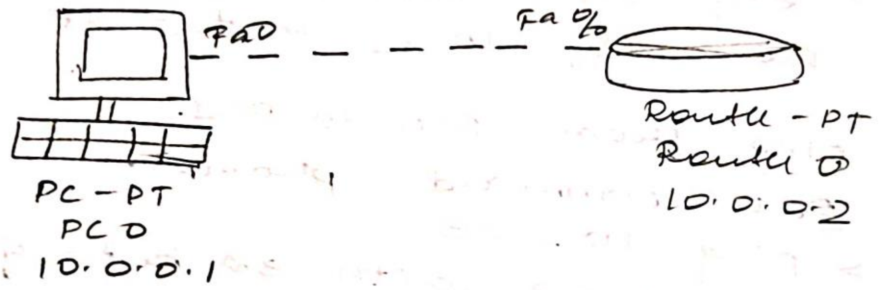

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=0ms TTL=128
Reply from 10.0.0.1: bytes=32 time=0ms TTL=128
Reply from 10.0.0.1: bytes=32 time=0ms 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 = 0ms, Maximum = 1ms, Average = 0ms

PC>arp -a
  Internet Address      Physical Address      Type
  10.0.0.1              00d0.ba1e.cb8d        dynamic

PC>arp -d
PC>arp -a
No ARP Entries Found
PC>
```

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

Lab 11

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

Topology



Procedure

Step 1: create a topology using 1 PC and 1 router as shown.

Step 2: set the ip address and gateway as 10.0.0.1 and 10.0.0.2 for the PC

Step 3: In the router, go to CLI

router > enable

router # config-t

router (config) # ~~hostname~~ hostname r1

r1 (config) # enable secret P1

r1 (config) # interface fastethernet 0/0

r1 (config-if) # ip address 10.0.0.2 255.0.0.0

r1 (config-if) # no shut

R1 (config -if) # line vty 0 5
R1 (config -line) # login.
% login disabled on line 132, until
'password' is set.
R1 (config - line) # password P0
R1 (config - line) # exit.
R1 # w.r.

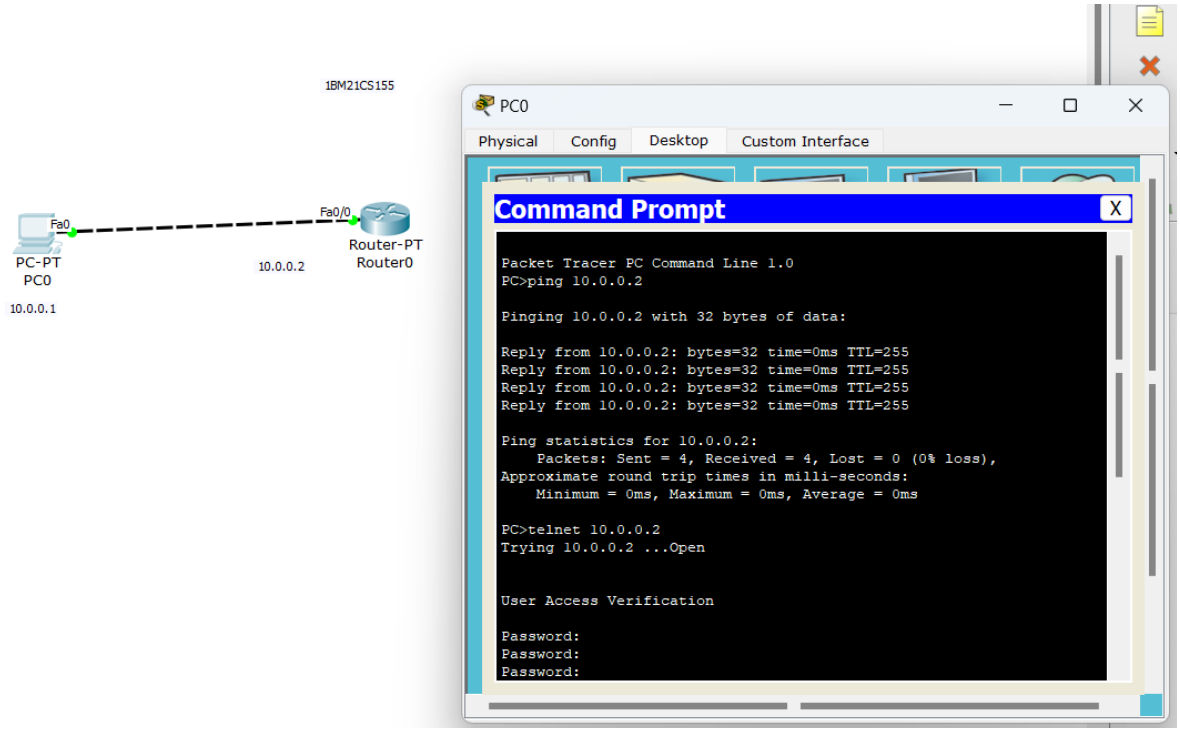
Step 4: In cmd prompt of PC,
PC > ping 10.0.0.2.
pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2 : bytes = 32 time < 0ms
Reply from 10.0.0.2 : bytes = 32 time = 0ms
Reply from 10.0.0.2 : bytes = 32 time = 0ms
Reply from 10.0.0.2 : bytes = 32 time = 0ms
ping statistics for 10.0.0.2
packets : sent = 4 : received = 4, lost = 0.
approx round trip times in ms.
min = 0 ms, max = 0ms, avg = 0ms
PC > telnet 10.0.0.2.
~~trying~~ 10.0.0.2 --- open
user access vertificam.
password: (P0)
R1 > enable
~~R1#~~ password : (P1)
R1 # show ip route
codes : C - connected.
C 10.0.0.0/8 is directly connected.

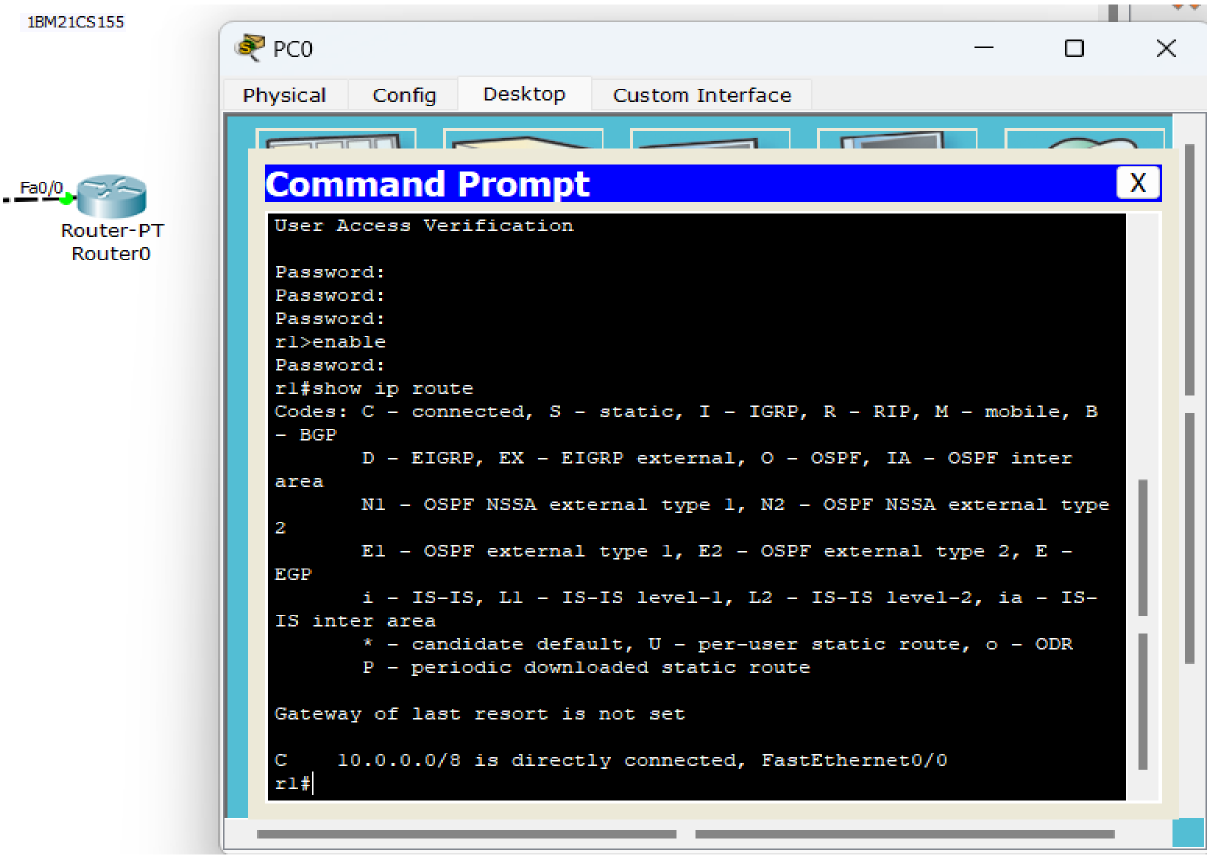
Observation

Using telnet protocol, we can access the router from the PC (which is connected to it).

Lab 12

TOPOLOGY:


1BM21CS155
Fa0
Fa0/0
PC-PT
PC0
10.0.0.1
10.0.0.2
Router-PT
Router0
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
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=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
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open
User Access Verification
Password:
Password:
Password:

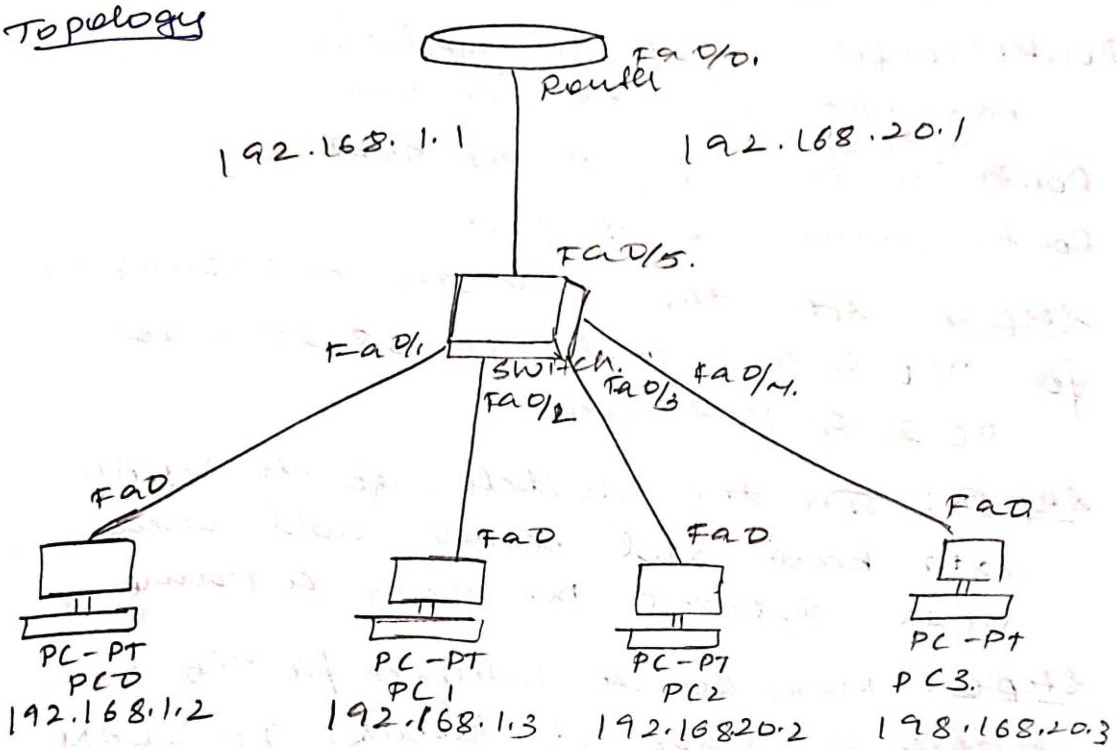

1BM21CS155
Fa0/0
Router-PT
Router0
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
User Access Verification
Password:
Password:
Password:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C   10.0.0.0/8 is directly connected, FastEthernet0/0
r1#

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

Lab 8

Aim: To construct a VLAN & make the PC communicate among VLAN.

Topology



Procedure:

Step1: create a topology as given above. Using 4 PC, 1 switch (switch-2960) & router (router-1841).

Step2: configure the IP address for PC as 192.168.1.2 & 192.168.1.3 for PC0 & PC1 & 192.168.20.2 & 192.168.20.3 for PC2 & PC3 resp.

Step3: Configure the ip address for router using following commands.

Router > enable
Router # config t
Router (config) # interface fastethernet
Router(config - if)# ip address
192 168.1.1 255.255.255.0
Router (config - if) # no shut
Router (config - if) # exit.

Step 4: Set the gateway as 192.168.1.1 for PC0 & PC1 and 192.168.20.1 for PC2 & PC3 resp.

Step 5: In the switch, go to VLAN data base and create add new VLAN database by giving a name.

Step 6: Now go to interface fa 0/5 in switch & make it trunk. In VLAN everything need to be selected.
This allows diff VLAN over single link called trunk.

Step 7 Go to router and select VLAN database. Enter the no & name of VLAN created before, goto CLI in the router & give the following cmd

Router (VLAN)# ~~Exist~~ exit.
Apply completed
exiting ---

Router (config)# interface fastethernet 0/1

router (config-subif) # encapsulation dot 1q 2

router (config- subif) # ip address 192.168.20.1 255.255.255.0

router (config-subif) # no shut.

router (config- subif) # exit.

Step 8: In the switch for fa 0/3 & fa 0/4 select VLAN & no as no given for VLAN while creating.

Now ping from PC0 to PC3,

PC0 PC > ping 192.168.20.2.

pinging 192.168.20.2 with 32 bytes of data.

Reply from 192.168.20.2 : bytes = 32 time <1ms

Reply from 192.168.20.2 : bytes = 32 time = 0ms

Reply from 192.168.20.2 : bytes = 32 time = 3ms

Reply from 192.168.20.2 : bytes = 32 time = 1ms

ping statistics for 192.168.20.2:

packets : sent 4 : received : 4 lost = 0.

approx round trip times in milli seconds

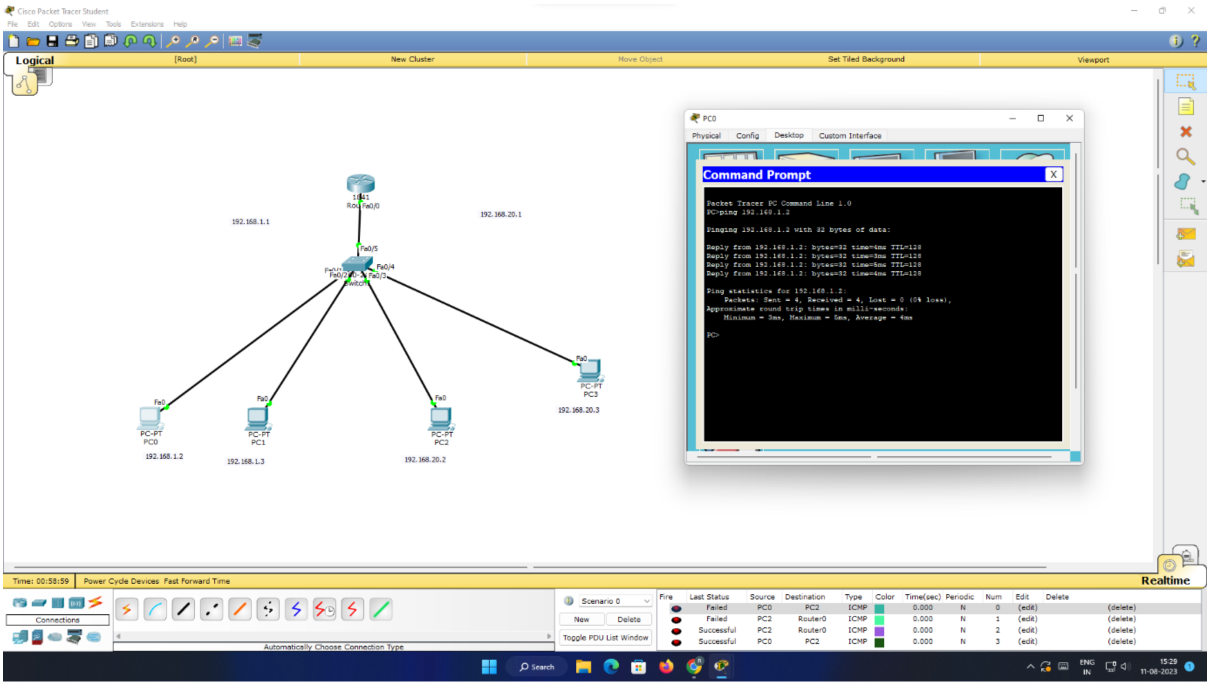
min = 0ms, max = 3ms avg = 2ms.

10/10

Observation?

N
28/5/25

TOPOLOGY:

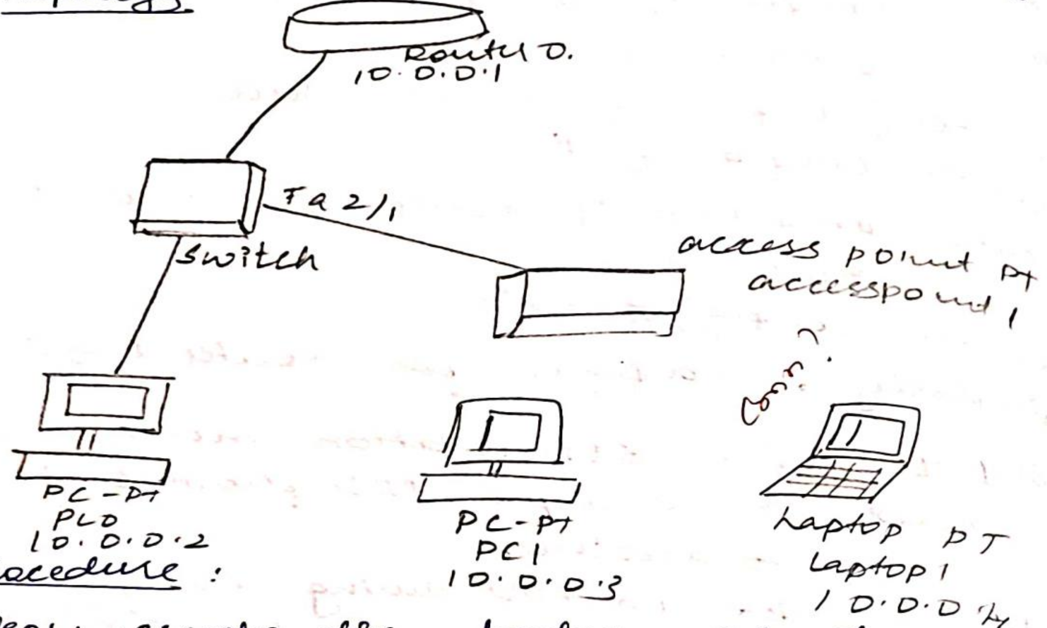

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=4ms TTL=128
Reply from 192.168.1.2: bytes=32 time=3ms TTL=128
Reply from 192.168.1.2: bytes=32 time=5ms TTL=128
Reply from 192.168.1.2: bytes=32 time=4ms TTL=128
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 3ms, Maximum = 5ms, Average = 4ms
PC>
192.168.1.1
192.168.20.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.1.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3
Realtime

12.To construct a WLAN and make the nodes communicate wirelessly.

Lab 10

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

Topology



Procedure :

Step 1 : create the topology as shown above with PC, switch router, access point and laptop.

Step 2 : Configure PC0 and router as normally done.

Step 3 : configure the access point 1, go to port 1 and give SSID name (any name)

Step 4 : Select WEP & give any 10 digit hex key (1234567890) Configure PC1 and laptop with wireless standards

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

Step6: 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.

Now ping from PC0 to PC1.

In PC0 command prompts

PC > ping 10.0.0.3.

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: bytes = 32 time = 47ms

Reply from 10.0.0.3: bytes = 32 time = 47ms

Reply from 10.0.0.3: bytes = 32 time = 47ms

Reply from 10.0.0.3: bytes = 32 time = 47ms

ping statistics for 10.0.0.3.

packets: sent: 4 received = 4, Lost = 0.

approx round - trip time in ms:

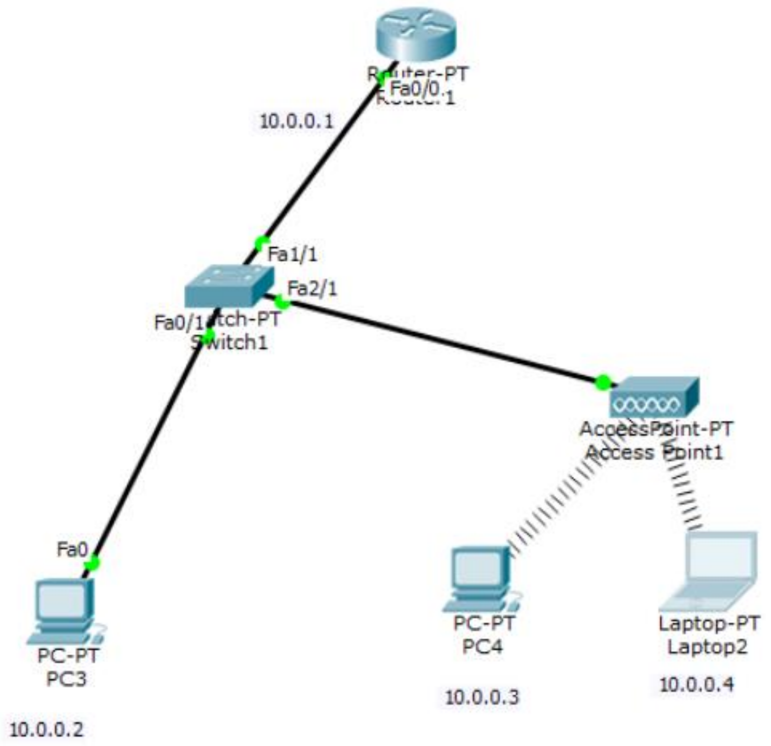
min = 35 max = 47ms, avg = 29ms

9/10

Observation

N
28/8/23

TOPOLOGY:



Router-PT
Fa0/0
Router1
10.0.0.1
Fa1/1
Fa2/1
Fa0/1
Switch-PT
Switch1
AccessPoint-PT
Access Point1
Fa0
PC-PT
PC3
10.0.0.2
PC-PT
PC4
10.0.0.3
Laptop-PT
Laptop2
10.0.0.4

CYCLE 2

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

CYCLE - 02

① Write a program for error detecting code using CRC - CCITT

```
#include <stdio.h>
char m[50], g[50], r[50], q[50],
   temp[50];
void crc (int n)
{     int i, j;
    for (i=0; i<n ; i++)
        temp[i] = m[i];
    for (i=0; i<16 ; i++)
        r[i] = m[i];
    for (i=0; i<n-16; i++) {
        if (r[0] == '1') {
            q[i] = '1';
            calram(); }
        else
            q[i] = '0';
            shift();
    }
    r[16] = m[17+i];
    r[17] = '\0';
    for (j=0; j<=17 ; j++)
        temp[j] = r[j];
    }q[n-16] = '\0';
}
```

```
void calcrem()
{
    int i,j;
    for (i=1; i<=16; i++)
    r[i-1] = ((int)temp[i]-48) ^
             ((int)g[i] - 48) +48;
}
void shift()
{
    int i;
    for (i=1; i<=16; i++)
      r[i-1] = r[i];
}
void caltrans(int n)
{
    int i, k=0
    for (i=n-16; i<n; i++)
     m[i] =((int)m[i] -48) ^ ((int) r[k++]
         - 48) +48);
     m[i] = '\0';
}
void main()
{
     int n, i=0;
     char ch, flag=0;
     printf("Enter the frame bits");
     while ((ch = getc(stdin))! ='\n')
      m[i++] = ch;
      n=i;
     for (i=0; i<16; i++)
      m[n++] = '0';
      m[n] = '\0';
```

```
printf ("message after appending 16
zeros : %s ", m);
for (i=0; i<=16 ; i++)
  g[i] = '0';
g[0] = g[4] = g[11] = g[16] = '1';
  g[17] = '\0';
printf ("generator : %s \n", g);
crc(n);
printf ("quotient : %s ", q);
 caltrans (n);
printf ("transmitted frame : %s ", m)
printf ("enter - transmitted frame :");
scanf ("%s", m);
printf (" crc checking \n");
 crc (n);
printf ("last remainder : %s ", r)
for (i=0; i<16; i++)
   if (r[i] != '0')
    flag = '1';
    else
      continue;
 if (flag == '1')
printf ("error ");
else
 printf (" frames are correct ")
}
```

O/P

Enter frame bits : 1011

Message after appending 16 zero :

1011 0000 0000 0000 0000

generator : 10001000000100001

quotient : 1011

transmitted : 1011 1011 0001 0110 1011

enter - transmitted frame

1011 1011 0001 0110 1011

last remainder 0000 0000 0000 0000

Received frame is correct.

10/10

28/8/23

2. Write a program for congestion control using Leaky bucket algorithm.

② WAP for congestion control using leaky bucket algorithm.

```
# include <stdio.h>
void main
{ int incoming, outgoing, bucketsize,
      n, store = 0;
  printf("enter bucket size, outgoing
    rate and no of I/P");
  scanf("%d %d %d", & bucket_size
      & outgoing, & n);
  while (n != 0)
  {
   printf("enter the incoming packet");
   scanf("%d", & incoming);
   printf("Incoming packet size
     %d \n", incoming);
   if (incoming <=(bucket_size))
   {
     store += incoming;
  printf("Bucket buffer size %d
     out of %d \n", store, bucketsize);
  }
   else {
  printf("Dropped %d no of packets \n",
    incoming - (bucket-size));
  printf("Bucket buffer size %d out
       of %d \n", store, bucket_size);
  store = bucket-size;
  }
```

```
store = store - outgoing;
printf ("After outgoing %d packets left
    out %d in buffer \n", store
        bucket_size);
    n--;
  }
}
```

O/P

Enter bucket size, outgoing rate and
  no of I/P
    20 10 2

Enter the incoming packet size =30
 Incoming packet size =30
 Dropped 10 no of packets
 Bucket buffer size 0 out of 20
After outgoing 10 packet left out
 20 in buffer.

Enter the incoming packet size=10
 Incoming packet size =10
 Bucket buffer size 10 out of 20
After outgoing 10 packets left out
 20 in buffer.

10/10

28/8/23

3. 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.

③ 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.

Server

1. A server has a bind() method which binds to specific IP and port so that it can listen to incoming request on that IP port and port.
2. A server has a listen() method which puts the server into listening mode. This allows server to the listen to incoming connections.
3. Server has accept() and close() method

accept() - initiates a connection with client

close() - closes the connection with the client.

Step 1: Open idle, in that in file, open new file and write the following code and save as ~~server.py~~ serverTCP.py

```
from socket import*
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,
               SOCKET_STREAM)
```

```
serverSocket.bind((serverName,
  serverPort))
serverSocket.listen(1)
while 1:
  print ("The server is ready to receive
  connectionSocket, addr = serverSocket.
                              accept()
  sentence = connectionSocket.recv(102
                              decode)

  file = open(sentence, "r")
  1 = file.read (1024)
  connectionSocket.send (1.encode())
  print('\n sent contents of' + setence)
  file.close()
  connectionSocket.close()
```

Step 2: Run the file server.py

O/P => The server is ready to receive

This shows that server is working

Client

Step 1: Make a socket object.

Step 2: Establish a connection with server and lastly we will receive data from the server and close the function

Step 3:

Open idle and open new file and write the following code and save as "client.py"

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name :")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()

print ("\n From server : \n")
print (filecontents)
clientSocket.close()
```

Run the file client.py

client o/p

O/P is enter file name . server.py

~~when the client request~~

server O/P.

The server is ready to receive

sent the contents of server.py

The server is ready to receive.

10/10

1/9/23

OUTPUT:

```
*IDLE Shell 3.10.8*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:/Users/Admin/Desktop/server1.py ==================
The server is ready to receive

Sent contents of server1.py
The server is ready to receive
```

```
IDLE Shell 3.10.8
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clent.py ===

Enter file name: server1.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()
```

4. 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.

(4) Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of req file if present.

Here, like in TCP/IP, we create socket object and bind it to specified port and server will be continously listening when the client sends request is responds accordinly

serverUDP.py

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

clientUDP.py

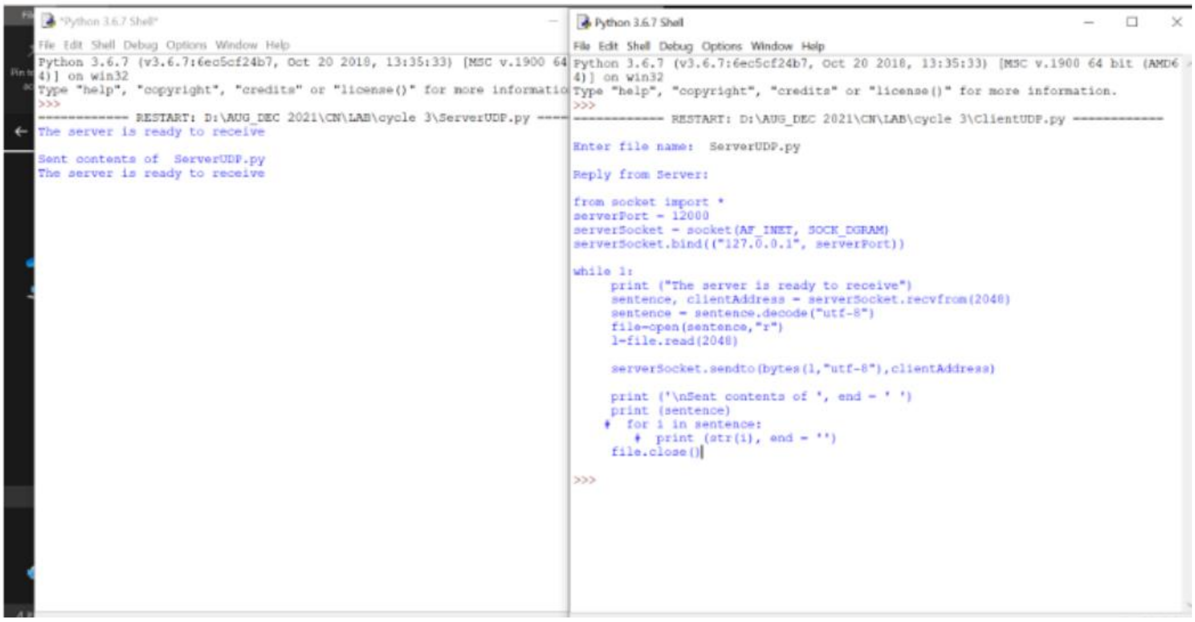
from socket. import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket (AF_INET,
SOCK_DGRAM)
sentence = input ("\n Enter file name:")
clientSocket . sendto (bytes (sentence,
"utf-8"), (serverName, serverPort))
file contents , server Address = clientSocket.
recvfrom (2048)
print ("\n Reply from server : \n")
print (file contents. decode ("utf-8"))
~~# for i in file contents:~~
~~# print (str(i), end = '')~~
~~clientSocket~~

O/p. (client) -
Enter the file name : serverUDP . py.

O/p (server)
The server is ready to receive
sent contents of serverUDP.py.

10/10
1/9/23

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()
>>>