VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

CHARAN G (1BM22CS078)

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

Sep 2024-Jan 2025

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 NETWORKS**" carried out by **Charan G (1BM22CS078)** who is 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 year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

CYCLE 1

INDEX

CYCLE 2

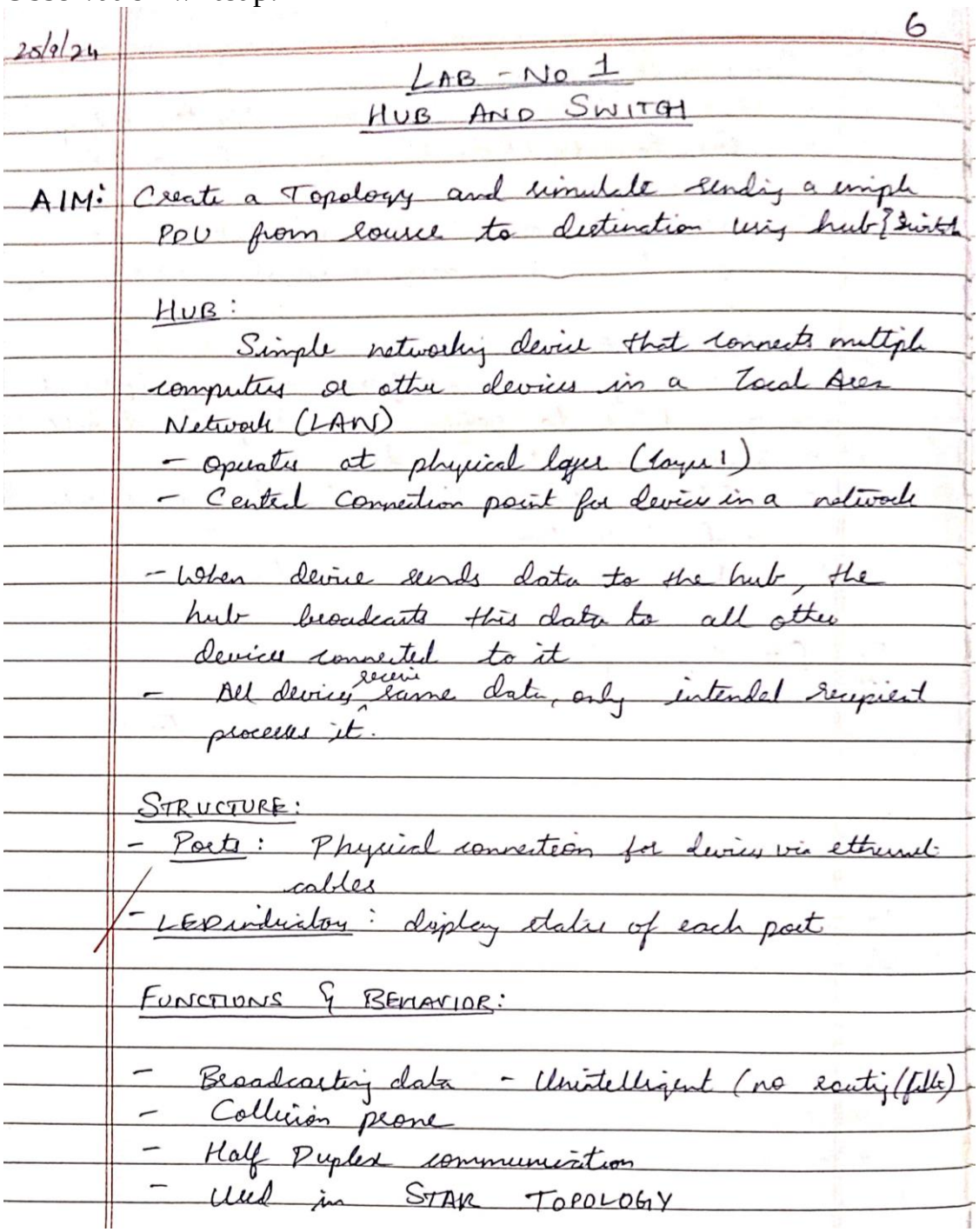
Cycle -1

Experiment 1: Hubs and Switches

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

A. HUBS:

Observation writeup:



28/9/24 6

LAB - NO 1

HUB AND SWITCH

AIM: Create a Topology and simulate sending a simple PDU from source to destination using hub & switch

HUB:

Simple networking device that connects multiple computers or other devices in a Local Area Network (LAN)

- Operates at physical layer (layer 1)
- Central connection point for devices in a network

- When device sends data to the hub, the hub broadcasts this data to all other devices connected to it
- All devices receive same data, only intended recipient processes it.

STRUCTURE:

- Ports: Physical connection for devices via ethernet cables
- LED indicators: display status of each port

FUNCTIONS & BEHAVIOR:

- Broadcasting data - Unintelligent (no security/filter)
- Collision prone
- Half Duplex communication
- Used in STAR TOPOLOGY

Bafna Gold
Date: Page:

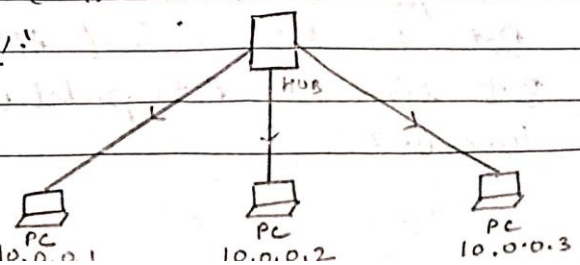
CREATING THE NETWORK: PROCEDURE

1.] **ADD DEVICES:**
- Select End Devices → PC
- Add multiple PC's, here lets say '3'
- Select Network Device → Hubs → Hub
- Now we add Hub to the workspace
- We have added 4 devices: one network device, three end devices

2.] **CONNECT DEVICES:**
- Select Connections → Copper Straight-Through as this is for connected devices of different types
- Click on PC and then click on Hub to form a connection:
  Click PC → Fast Ethernet 0
  Click Hub → Fast Ethernet 0 [0-5 : 6 ports]
- Connect all three PC's to the Hub

3. **CONFIGURE DEVICES:**
- click on PC → Config → Fast Ethernet 0
- Enter the IPv4 Address: 10.0.0.1
  Then click on subnet Mask to auto generate the address 255.0.0.0, then close window
- Immediately name the device with its assigned IP address: 10.0.0.1 by double clicking on the PC
- Repeat these steps to configure all the PC's and the hub

TOPOLOGY:



4. PING: TEST CONNECTIVITY:

- click on any one of the devices, say 10.0.0.1
  click 10.0.0.1 PC → Desktop → Command prompt
- Type 'ping 'ip-address''
  replace ip-address with actual address
  say, ping 10.0.0.3
- This is done to check if both the devices are connected / reachable
- On pinging, it was observed that both the devices were connected : Use simulation

5. OBSERVATIONS:

- Go to simulation mode at the right bottom corner
- Select PC 10.0.0.1 → Desktop → Comand Prompt
- ping 10.0.0.3 - enter the command
- A message icon is popped on the PC 10.0.0.1 from where the ping command was run
- In the simulation panel at the right side, there are simulation play buttons
- clicked on play buttons
- The data transfer was simulated

① Data packet ✉ was sent from PC 10.0.0.1 to the hub.
② The hub received it, then broadcasted the packet to the rest two PCs
③ Both the PCs received the data packets
④ But since the data packet was pinged to 10.0.0.3 :

Bafna Gold
Date: Page:

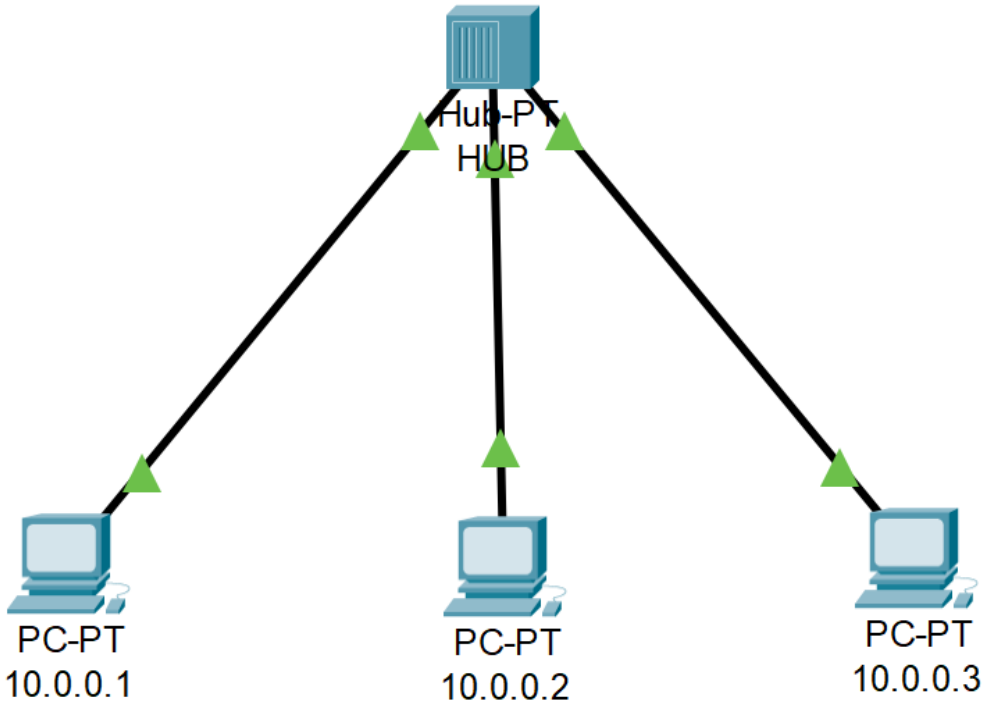
- The PC 10.0.0.2 on receiving showed/displayed blinking red cross mark : ✗
  Indicates → The data packet was not for them
- The PC 10.0.0.3 displayed nothing, indicating the data transfer was correct

⑤ PC 10.0.0.3 then replied with 32 bytes of data : sent the data packet to the hub
⑥ The hub received it, then again broadcasted it to the other two PC's
⑦ Both the PC's received the data packets
⑧ PC 10.0.0.1 showed blinking tick symbol in green : ✓, indicating the right device, the PC 10.0.0.2 displayed cross symbol ✗

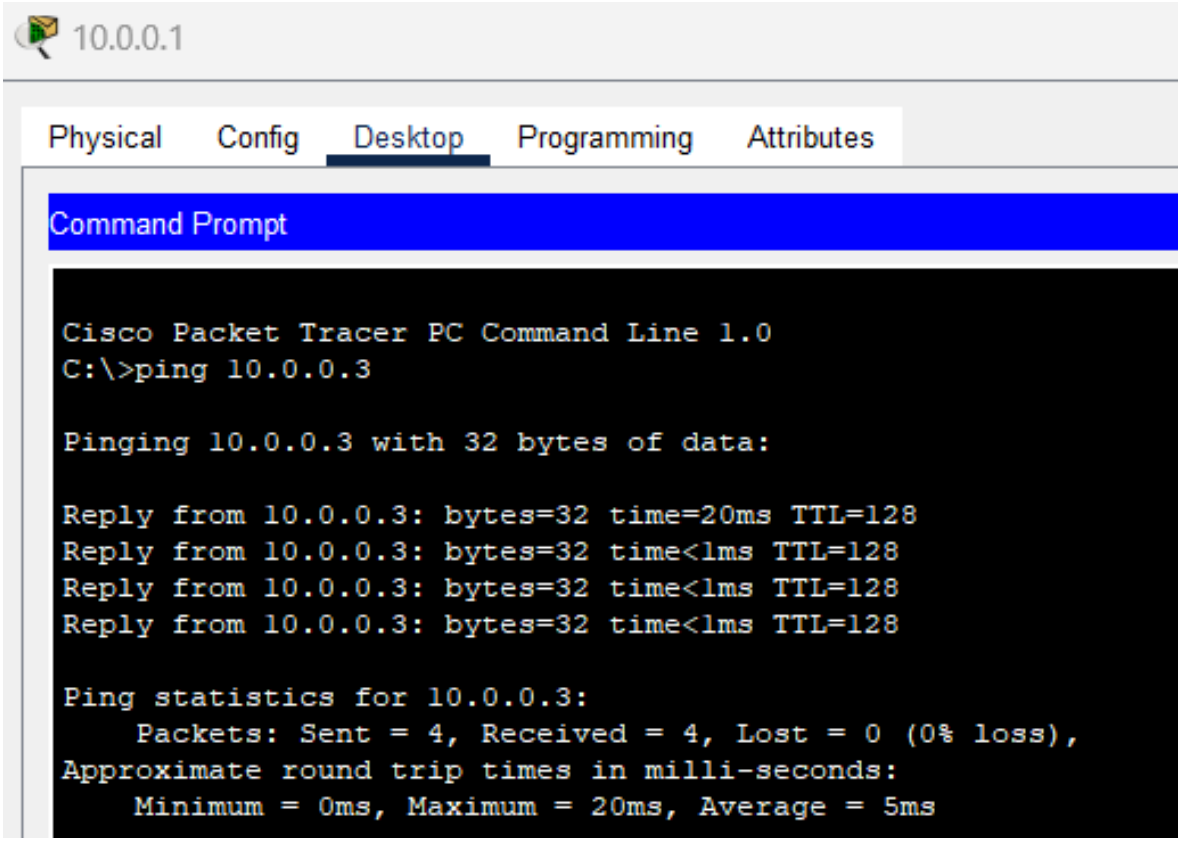
- The data packet icon ✉ changed colors for each cycle indicating different messages

CONCLUSION:
- Hub is an unintelligent Broadcasting device
- It is a half Duplex communication
- Collisions :
  - Added 6 devices, tried sending two packets from different devices at same time : collision occured 💥 : fire symbol
  - Hubs are collisions prone : they cannot handle collisions

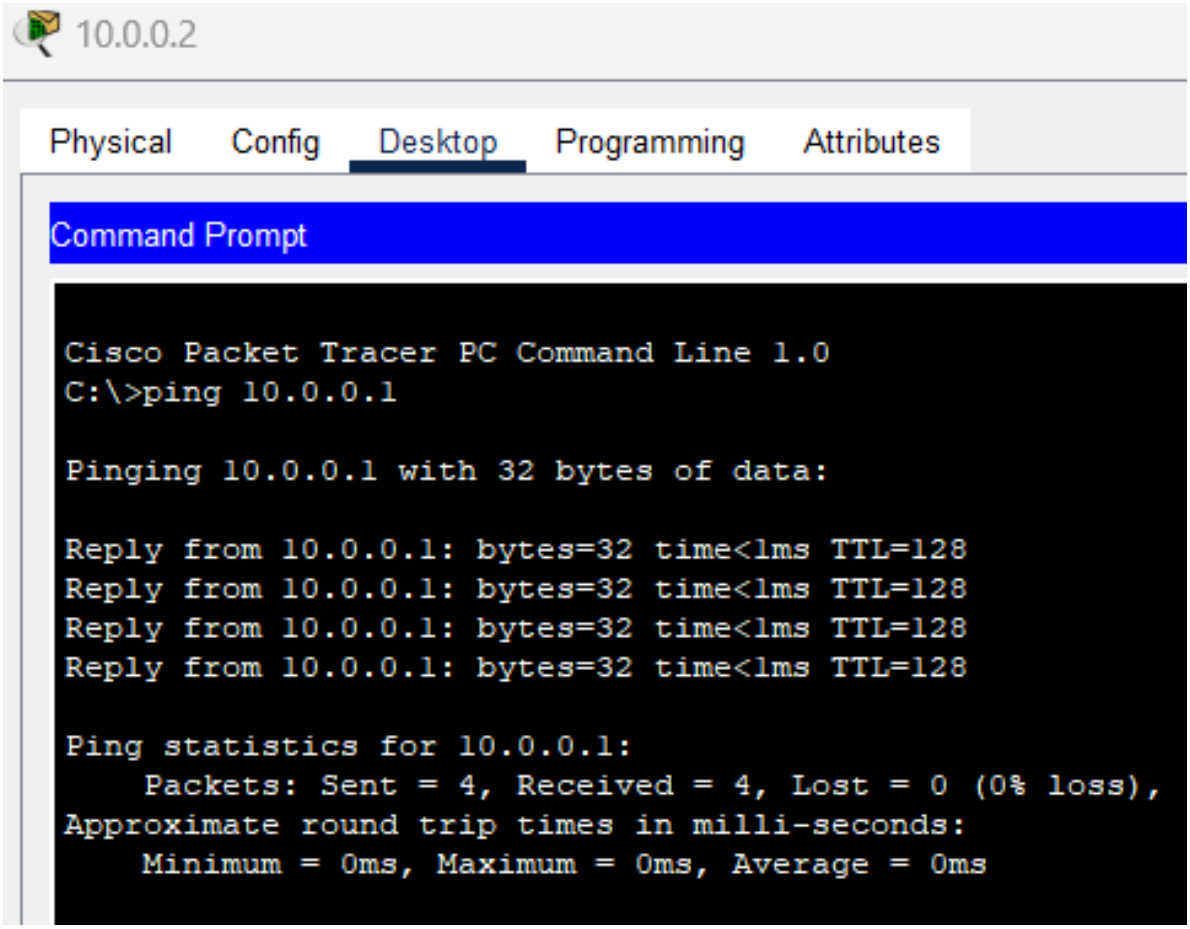
Screenshot of the topology:



Screenshot(s) of the output:



```
C:\>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=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

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 = 1ms, Average = 0ms
```



B. SWITCHES:

Observation writeup:

10

25/9/24

SWITCH

AIM: Switch: How switches work & its comparison to hubs

Network device that connects multiple devices within a local area network (LAN)

- Intelligently forwards data to the correct destination based on the MAC address of device
- Operates as Data Link Layer

STRUCTURE:

Ports: Connection for devices via cables

Mac Address Table: Maps Mac address of connected devices to corresponding switch port

CREATING THE NETWORK: PROCEDURE

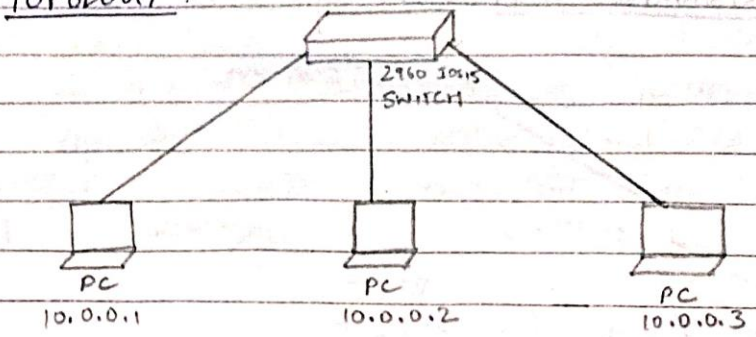
1) ADD DEVICES:
- Select End devices → PC → add 3 PCs
- Select Network devices → Switches → 2960 IOS15 add one switch

2) CONNECT DEVICES:
- Select Connections ⚡ → Bold Line
- Connect end devices to the switch
- Initially the connection shows 'Orange' color at switch end
  : Indicates Listening & Learning States (MAC Address)
- Then the color turns 'Green' indicating forwarding state : Spanning Tree Protocol

3) CONFIGURE DEVICES:
- Assign IP address to the devices
- Click device → Config → FastEthernet → IP address
  ↓
  Subnetmask

Bafna Gold
Date: Page:

TOPOLOGY:



4. PING: TEST CONNECTIVITY:
- click PC 10.0.0.1 → Desktop → Cmd Prompt
- Enter ping 10.0.0.3 to ping that device

5. Observations:
- Data pack was sent to switch from 10.0.0.1
- The switch then processed it, and sent it to its destination 10.0.0.3 only
- The PC 10.0.0.3 replied to the switch
- The switch processed it, then sent the packet back to 10.0.0.1 PC
- Then a green tick mark '✓' was blinking at PC 10.0.0.1 showing success

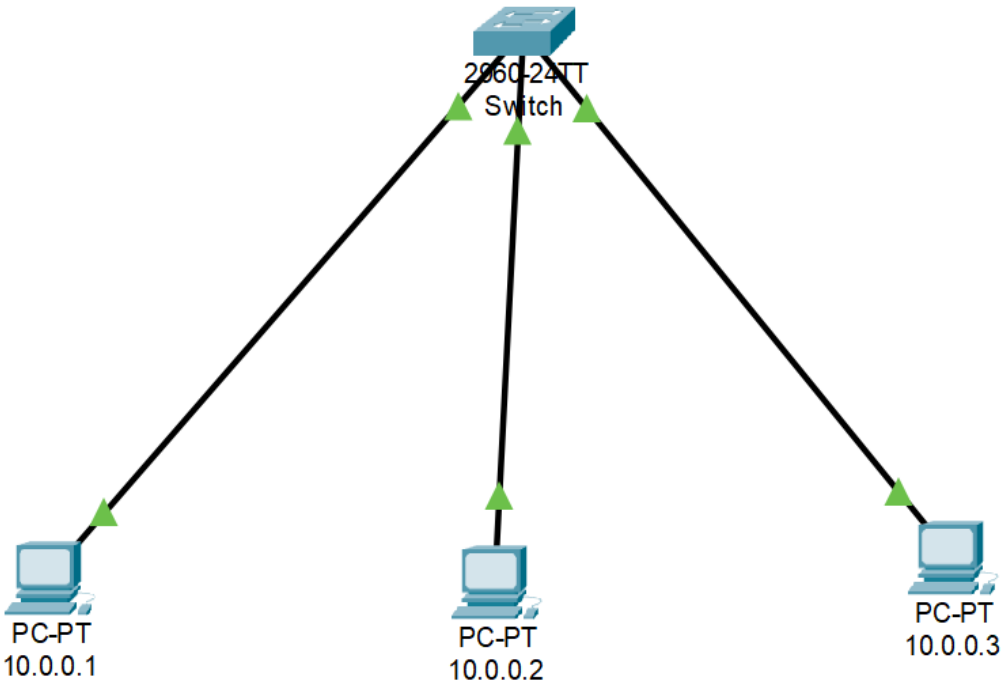
CONCLUSION: OBSERVATIONS:
- Switch is an intelligent networking device
- It stores MAC addresses of the devices
- It doesn't broadcast data, but it unicasts data
- It only sends data to its destination
- It is collision resistant: handles collisions

Difference :

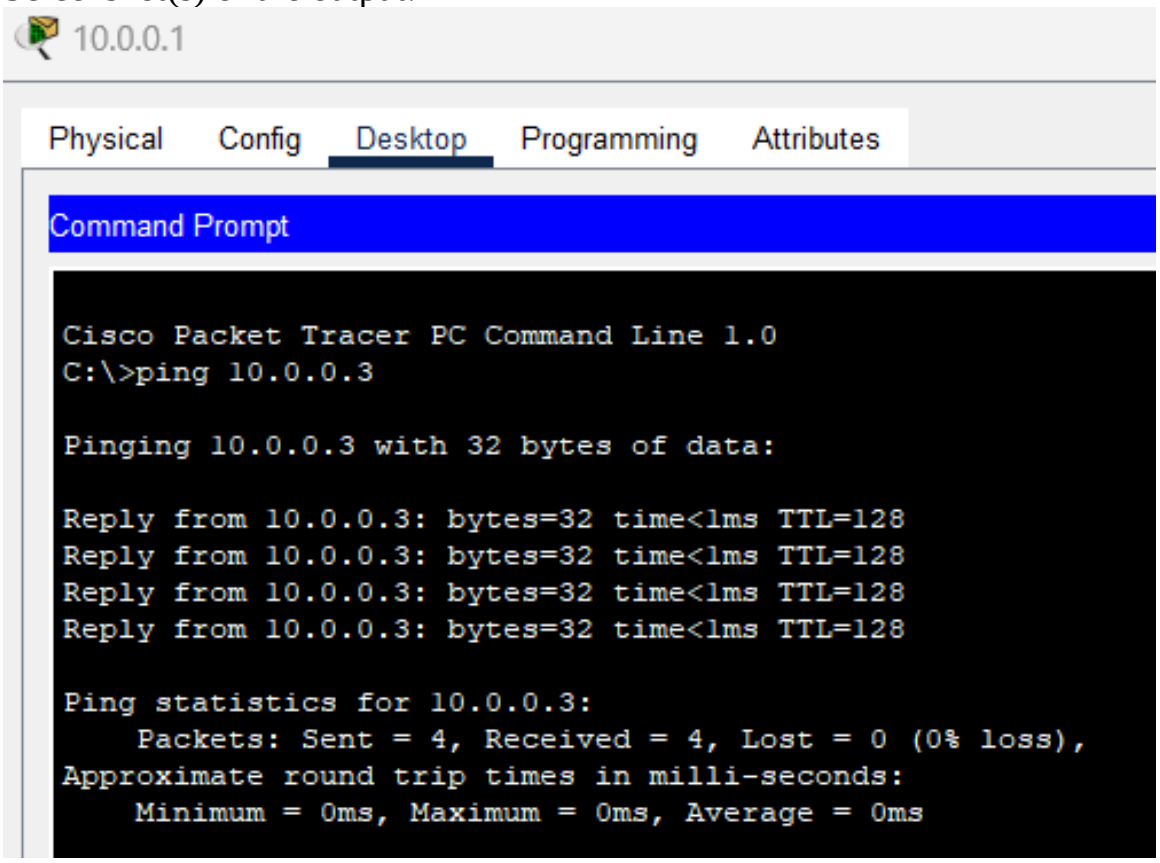
Switch	Hub
- Forwards data only to intended device	Broadcasts data to all connected devices
- Operates at layer 2 (Data link)	Operates at layer 1 (Physical link)
- Uses MAC address to forward data	- Doesn't use MAC address
- More efficient	Less efficient
- Dedicated bandwidth per port	Shared bandwidth across all ports
- Each port has its own collision domain	All devices share the same collision domain
- Supports full duplex communication	Only supports half duplex communication
- More expensive	Cheaper

25/9/24

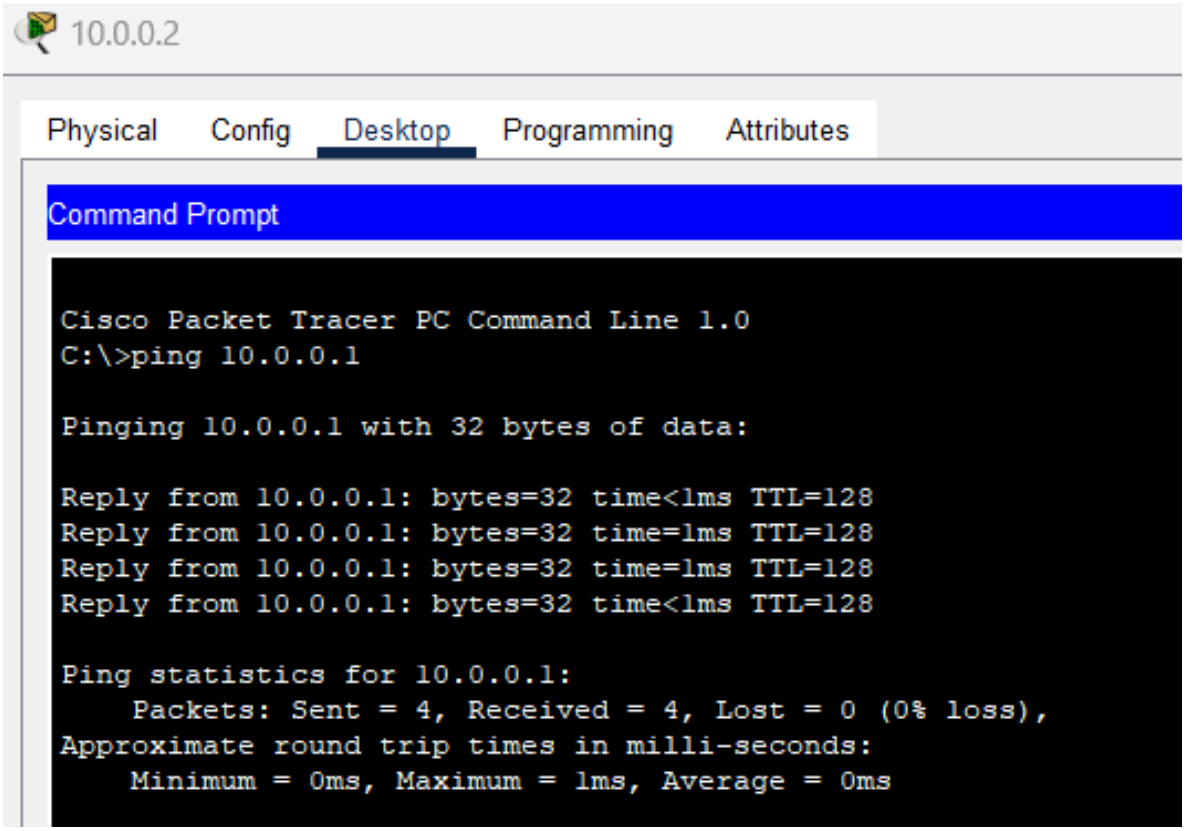
Screenshot of the topology:



Screenshot(s) of the output:



```
C:\>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=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128

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 = 1ms, Average = 0ms
```

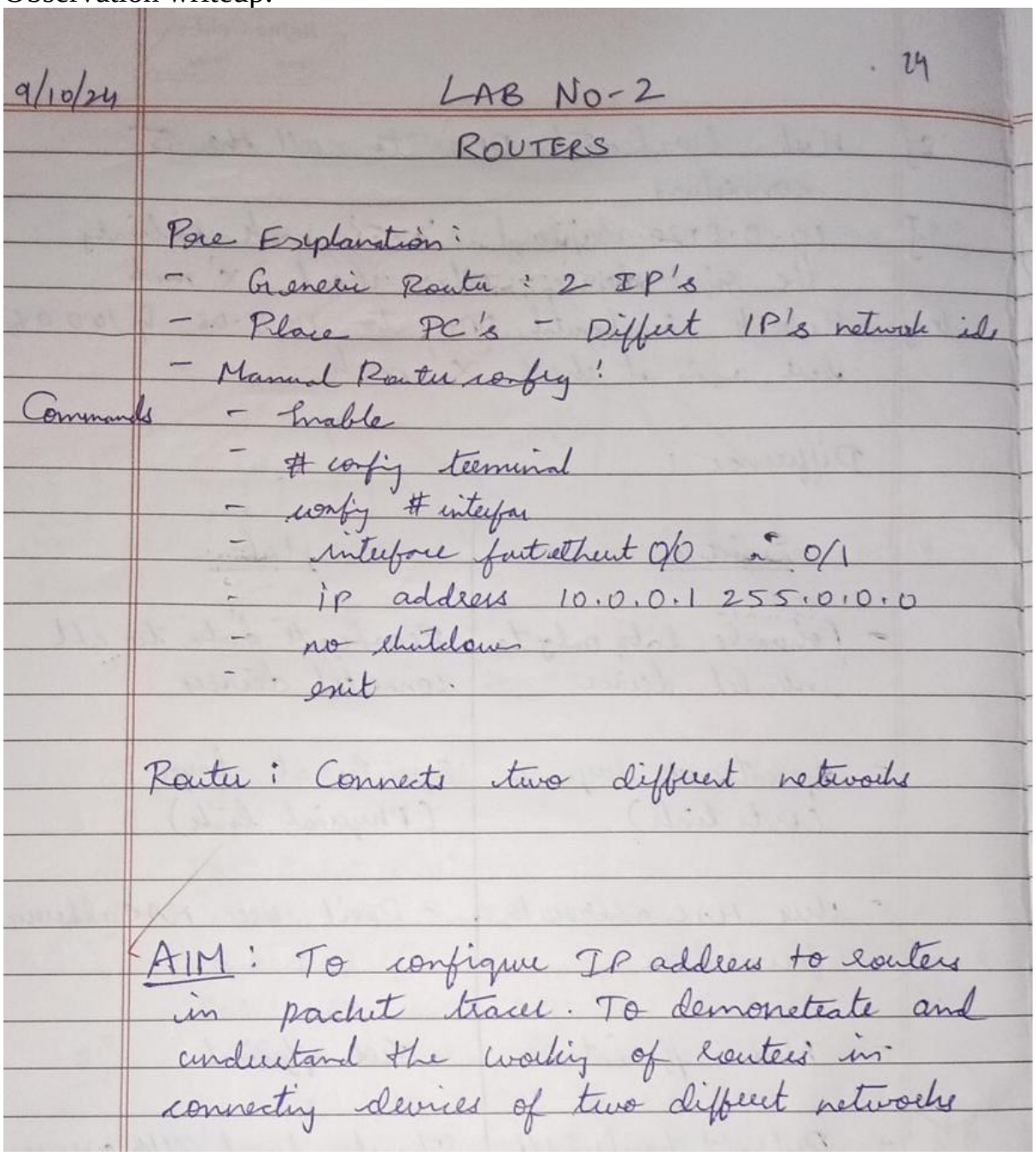


Experiment 2: Router Configuration

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

A. SINGLE ROUTER

Observation writeup:



9/10/24 — 24

LAB No-2

ROUTERS

Pre Explanation:
- Generic Router : 2 IP's
- Place PC's - Diffrt IP's network id
- Manual Router config:

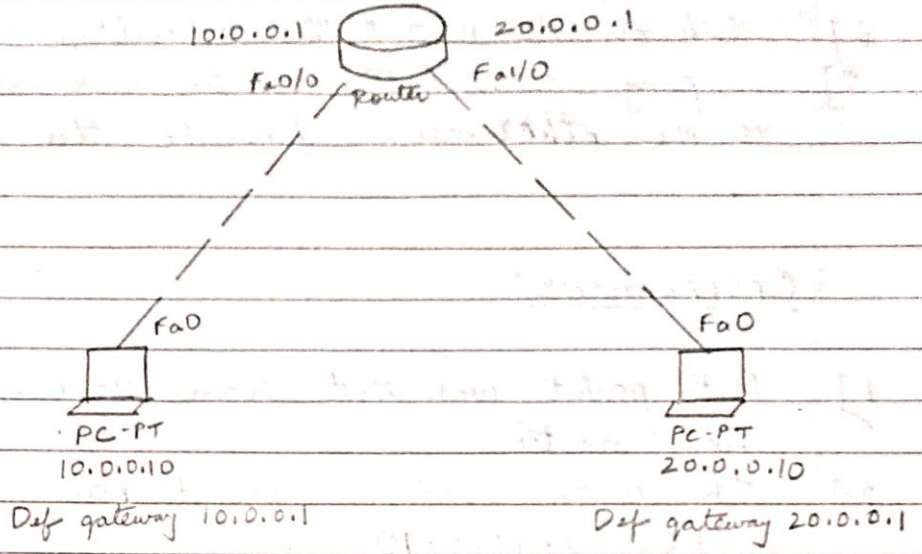
Commands
- Enable
- # config terminal
- config # interface
- interface fastethernet 0/0 & 0/1
- ip address 10.0.0.1 255.0.0.0
- no shutdown
- exit

Router : Connects two different networks

AIM: To configure IP address to routers in packet tracer. To demonstrate and understand the working of routers in connecting devices of two different networks

Bafna Gold
Date: Page:

TOPOLOGY :



PROCEDURE :

1.] Add two PC's and one generic router
2.] Configure end devices : 10.0.0.10 and 20.0.0.10 and mention default gateway 10.0.0.1 and 20.0.0.1 - and write below PC
3.] Connect the PC's to the router via copper cross over
4.] Click on Router → CLI → Manual configuration

Commands:

```
Router > enable
Router # config terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0

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

- Repeat for other PC : fastethernet 1/0

5.] After successful configuration, the connection turns green

6.] click on PC 10.0.0.10 → Desktop → Command prompt

7.] ping 20.0.0.10 → to send data packet to the ~~other~~ device from the other network

OBSERVATIONS:

1.] Data packet was sent from PC 10.0.0.10 to router

2.] The router sent the packet to PC 20.0.0.10

3.] Data packet back to Router → back to PC 10.0.0.10 and a tick mark is blinked

Ping command results were observed as:

Reply from 20.0.0.10 : bytes = 32 time = 4ms TTL = 127

9/10/24

Ping statistics for 20.0.0.10:

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

Approx round trip time in milli seconds:

Minimum = 4ms, Maximum = 4ms, Average = 4ms

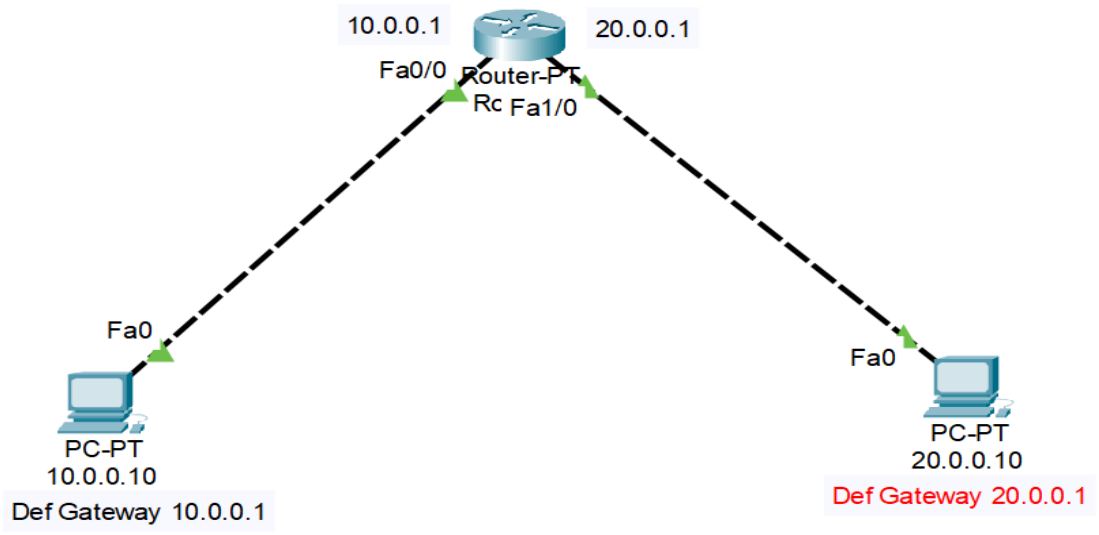
IP route was observed as:

Router# show ip route

C 10.0.0.0/8 is directly connected, FastEthernet0/0

C 20.0.0.0/8 is directly connected, FastEthernet1/0

Screenshot of the topology:



Screenshot(s) of the output:

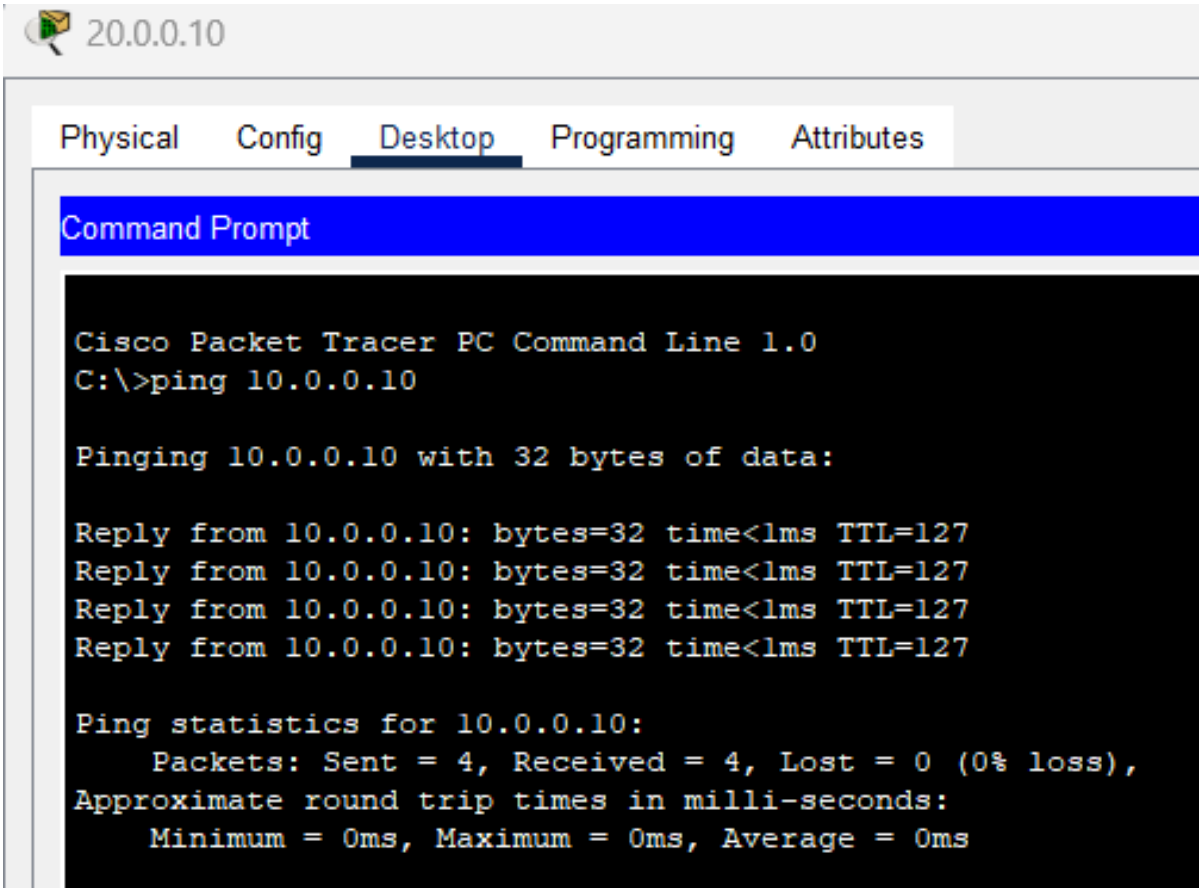
Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

```
C:\>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=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms TTL=127
Reply from 20.0.0.10: bytes=32 time<1ms 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 = 0ms, Average = 0ms

C:\>
```



B. TWO ROUTERS:

Observation writeup:

Bafna Gold 17

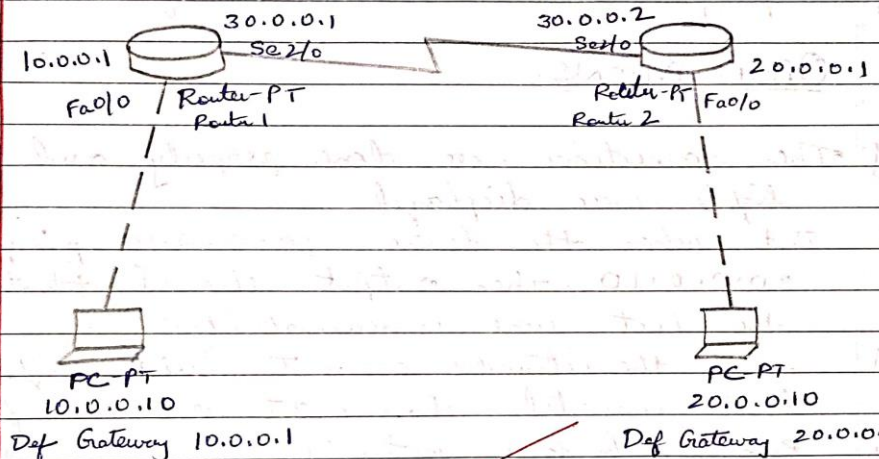
16/10/24

LAB No-3

2 ROUTERS

AIM: To configure IP address to routers in packet tracer. Explore following messages: ping responses, destination unreachable, request timed out, reply

TOPOLOGY:



PROCEDURE:

1. Add ~~two~~ PC's and two Generic Routers
2. Configure end devices 10.0.0.10 & 20.0.0.10 and define gateways
3. Connect PC to Router using copper cross over Connect the routers to each other using Serial DCE
4. Configure the routers:
   Click on Router → CLI
   - Configure to end devices similar to last experiment

- Connecting the routers:

Commands:

```
Router > enable
Router # config terminal
Router(config) # interface serial 2/0
Router(config-if) # ip address 30.0.0.1
Router(config-if) # no shut
```

5.] Give appropriate naming

6.] ping the device 20.0.0.1 from 10.0.0.1

OBSERVATIONS:

1.] The connections were done properly and green lights were displayed

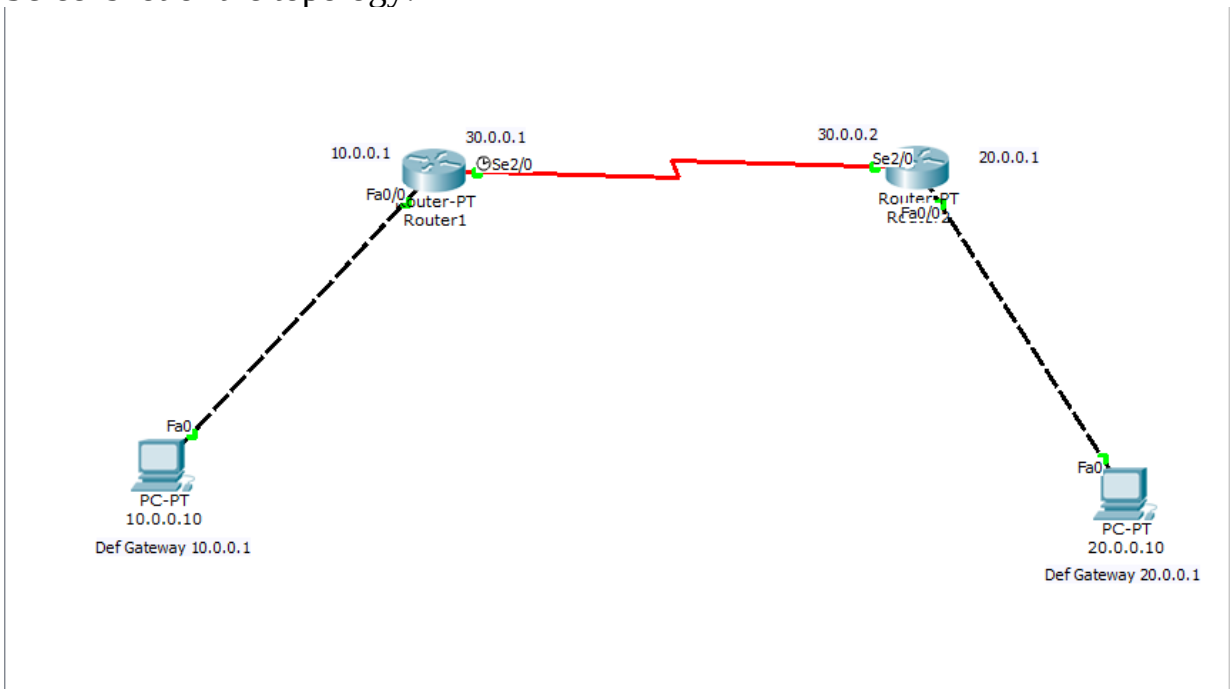
2.] But when the device 10.0.0.10 pinged 20.0.0.10, the output showed that the host was unreachable

3.] Since the networks are not direct neighbours / directly connected, the data sent was failed (Not)

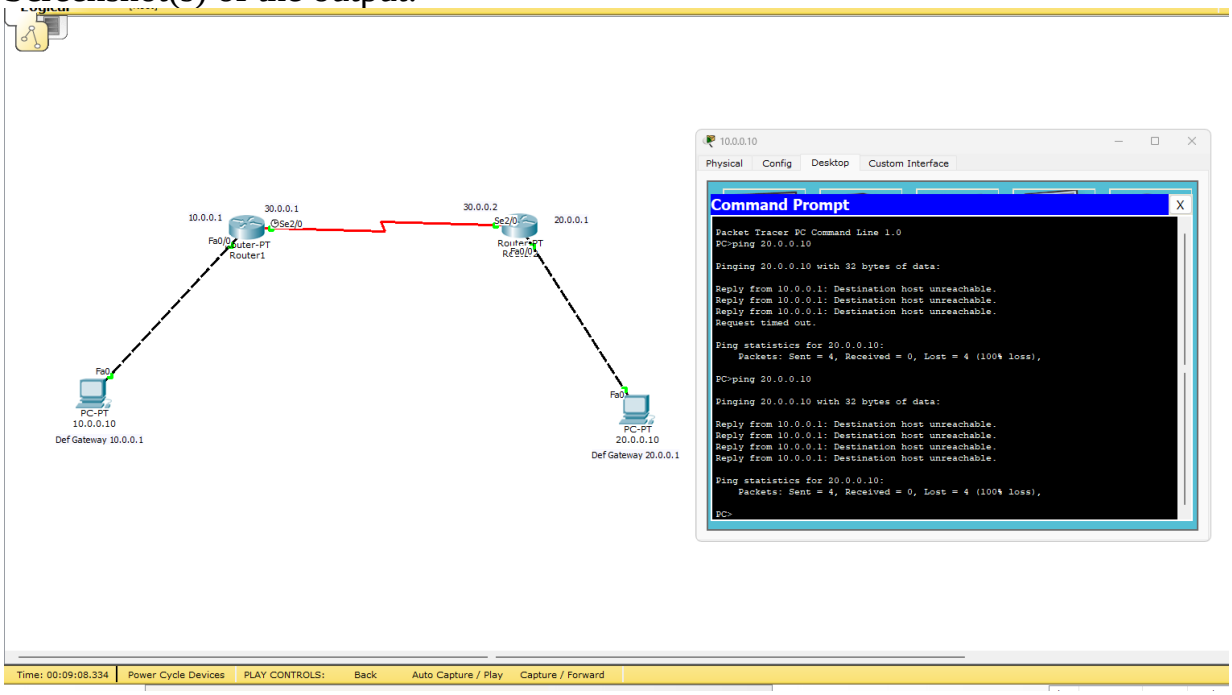
4.] Pinging the same networks was a success since they are directly connected, piging 30.0.0.1 from 10.0.0.10 was a success

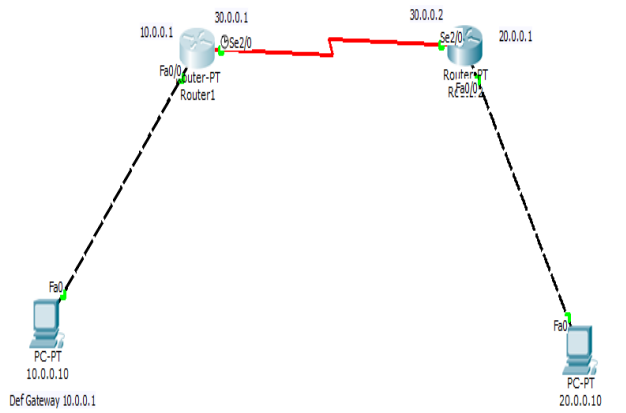
23/10

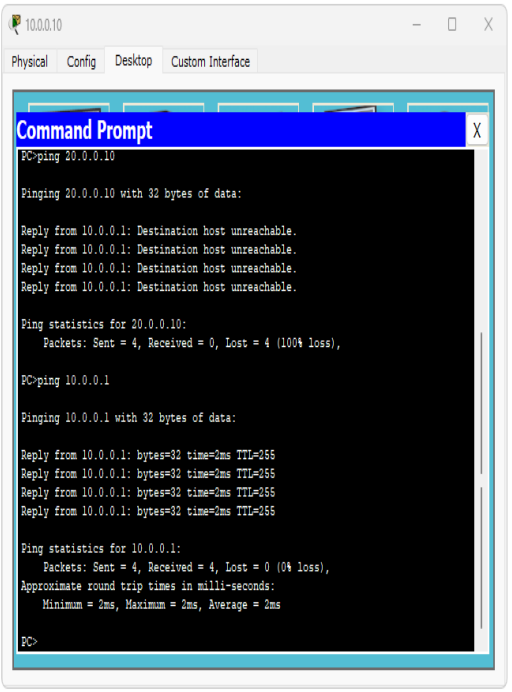
Screenshot of the topology:

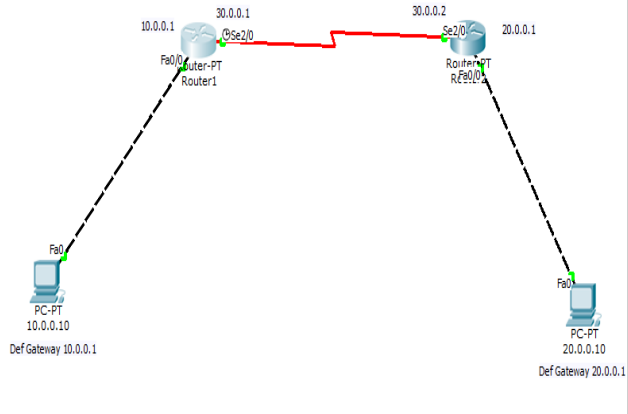


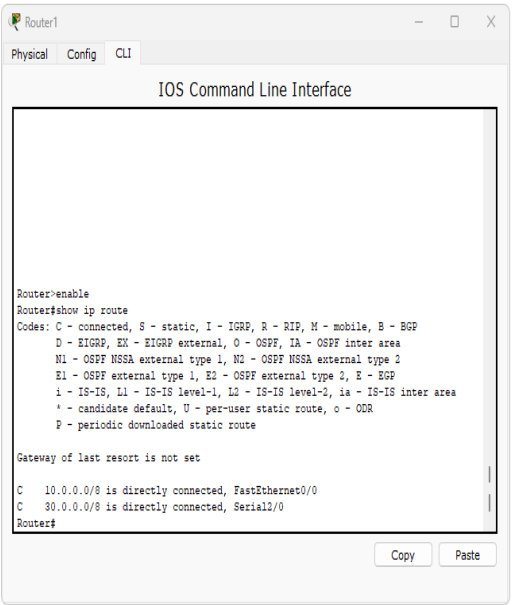
Screenshot(s) of the output:




10.0.0.1
30.0.0.1
Se2/0
30.0.0.2
Se2/0
20.0.0.1
Fa0/0
Router-PT
Router1
Router-PT
Fa0/0
Fa0
PC-PT
10.0.0.10
Def Gateway 10.0.0.1
Fa0
PC-PT
20.0.0.10
Def Gateway 20.0.0.1


10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 20.0.0.10
Pinging 20.0.0.10 with 32 bytes of data:
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Ping statistics for 20.0.0.10:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=255
Reply from 10.0.0.1: bytes=32 time=2ms TTL=255
Reply from 10.0.0.1: bytes=32 time=2ms TTL=255
Reply from 10.0.0.1: bytes=32 time=2ms TTL=255
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 = 2ms, Average = 2ms
PC>


10.0.0.1
30.0.0.1
Se2/0
30.0.0.2
Se2/0
20.0.0.1
Fa0/0
Router-PT
Router1
Router-PT
Fa0/0
Fa0
PC-PT
10.0.0.10
Def Gateway 10.0.0.1
Fa0
PC-PT
20.0.0.10
Def Gateway 20.0.0.1


Router1
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#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
C 30.0.0.0/8 is directly connected, Serial2/0
Router#
Copy
Paste

Experiment 3: Static and Default Routing

Question: Configure default route, static route to the Router

Observation writeup:

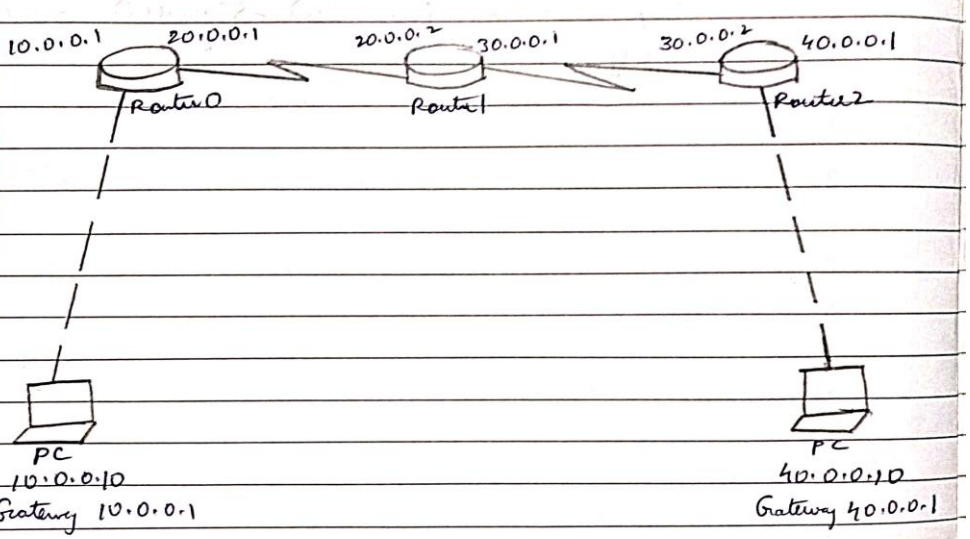
23/10/24

LAB NO. 4

STATIC ROUTING & DEFAULT ROUTING

AIM: To configure default routing and static routing to the router

TOPOLOGY



PROCEDURE:

1.] Place three routers and two PCs
2.] Connect PC's to Router 0 & Router 2 using cross over
3.] Connect Router 0 & Router 1, Router 1 & Router 2 using serial DCE
4.] Configure the PCs and the routers in the CLI using previous command lines learnt

Bafna Gold
Date: Page:

5.] Static Routing:

To configure ip routes manually rather than dynamic routing

- Router1 must know about the 10.0.0.0 network and also the 40.0.0.0 network
- To achieve this:

Open CLI in Router1:

Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1
(left network, next hop address)

Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

6.] Default Routing:

The route that takes effect when no other route is available for an IP destination address

Go to Router0:

(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

In Router2:

(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

7.] ping 40.0.0.10 from 10.0.0.10

OBSERVATIONS:

1.] IP route for Router1:

S 10.0.0.0/8 [1/0] 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 [1/0] via 50.0.0.2

C → connected S → Static

the two networks are connected via static routing

In Router 0:

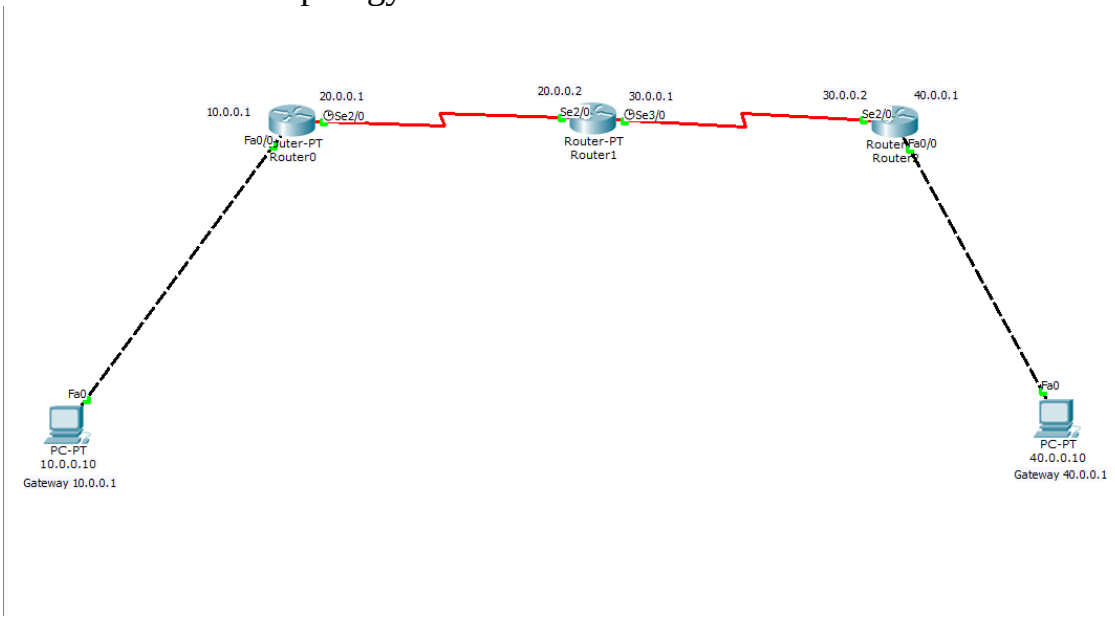
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2

S* : Static default routing

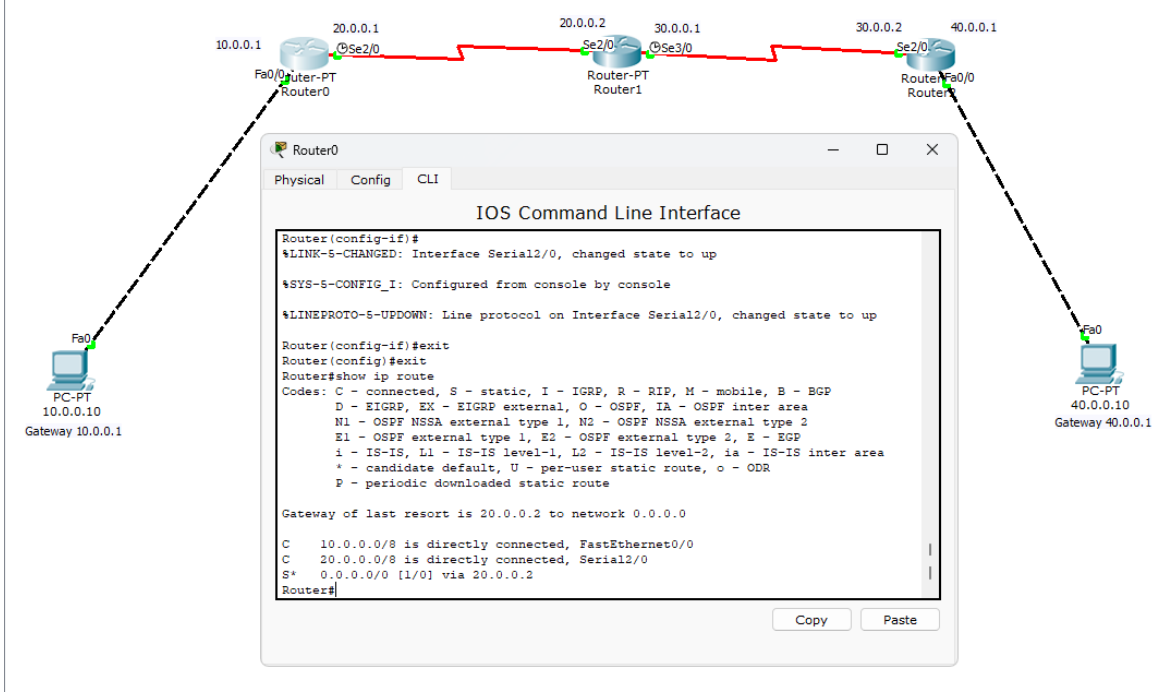
ping was successful:
Sent = 4, Received = 4, Lost = 0
Min = 8 ms Max = 8 ms, Avg = 8 ms

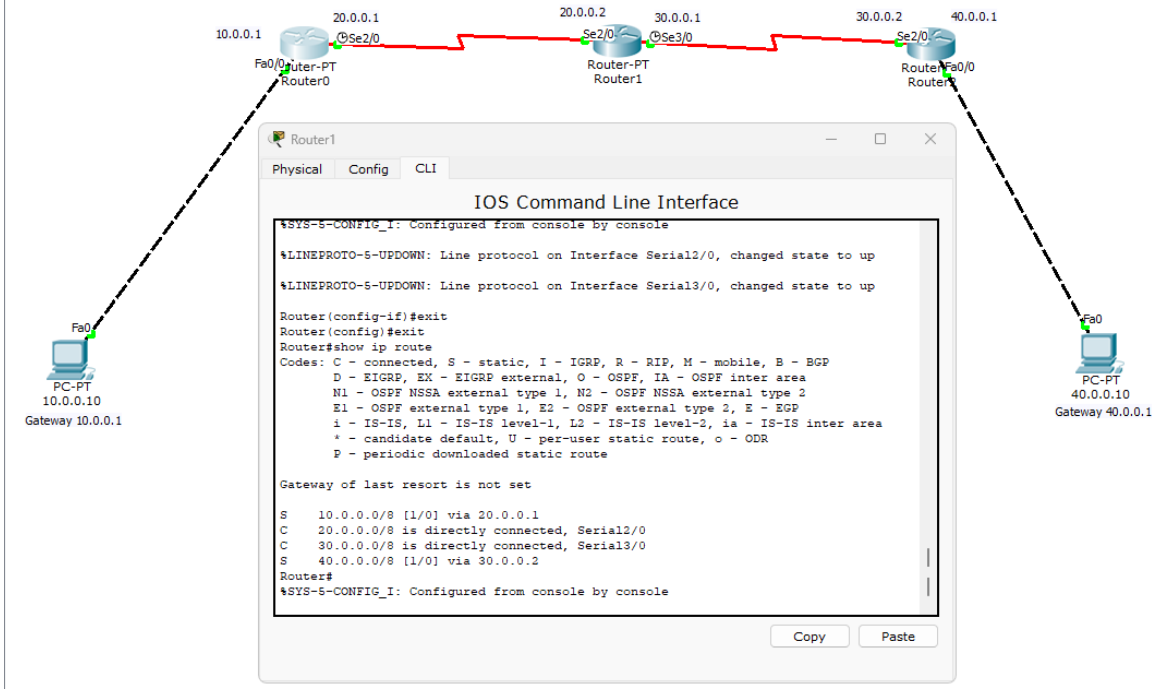
23/10

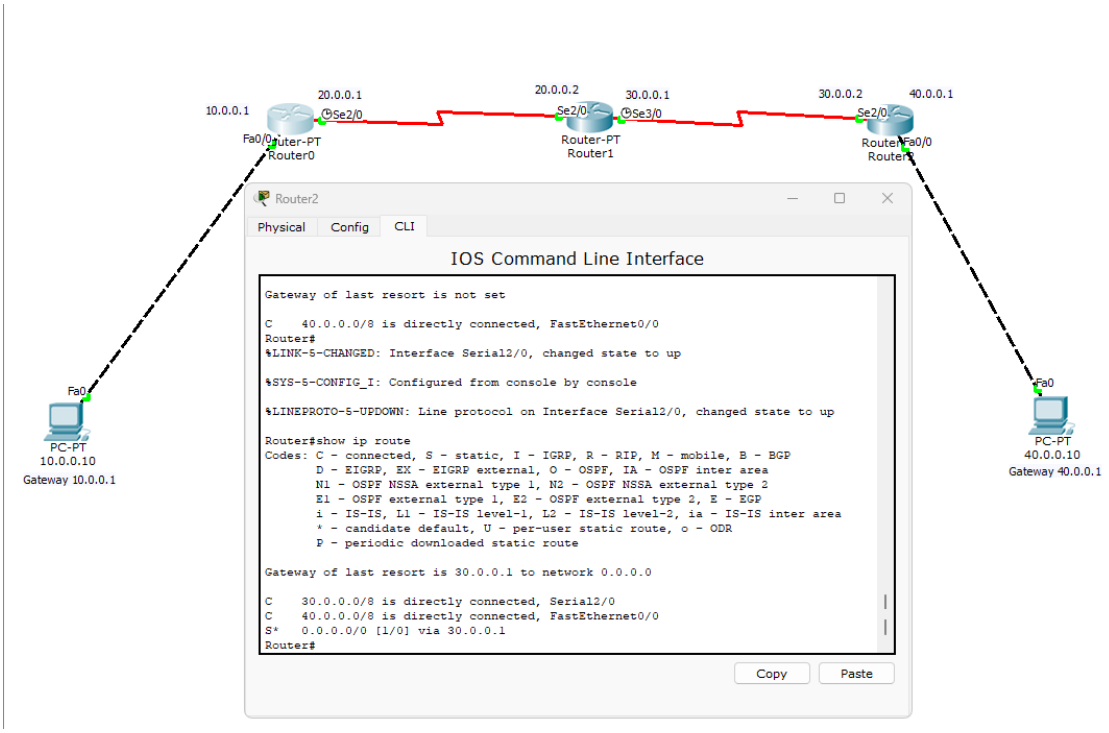
Screenshot of the topology:

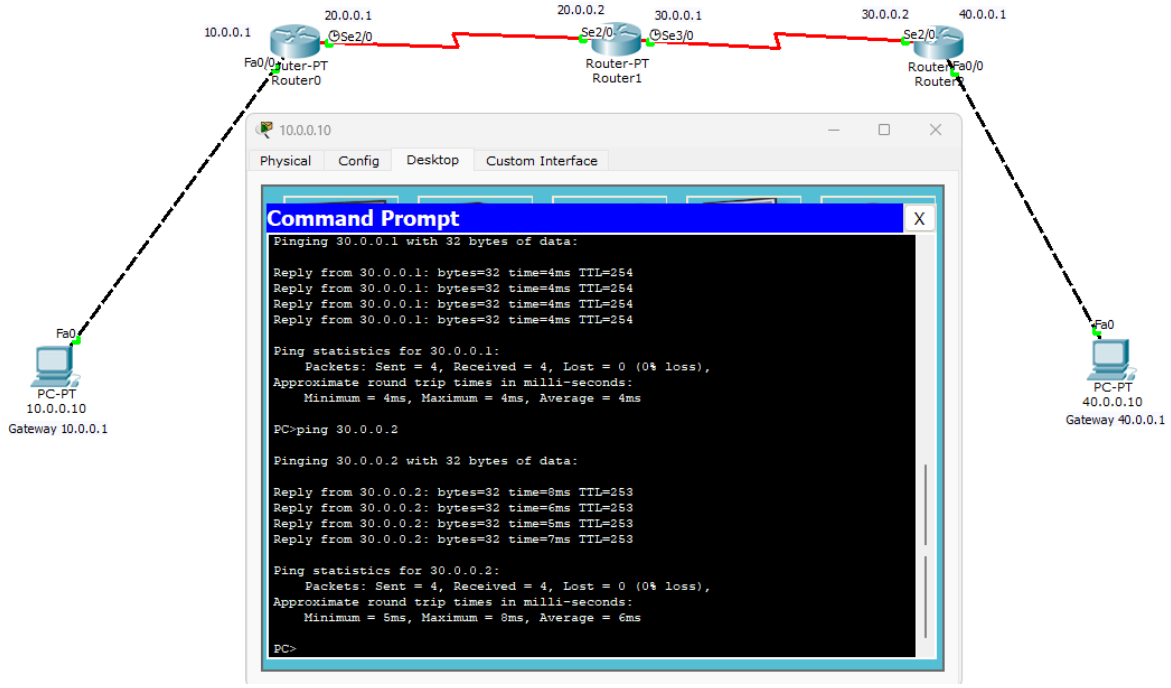


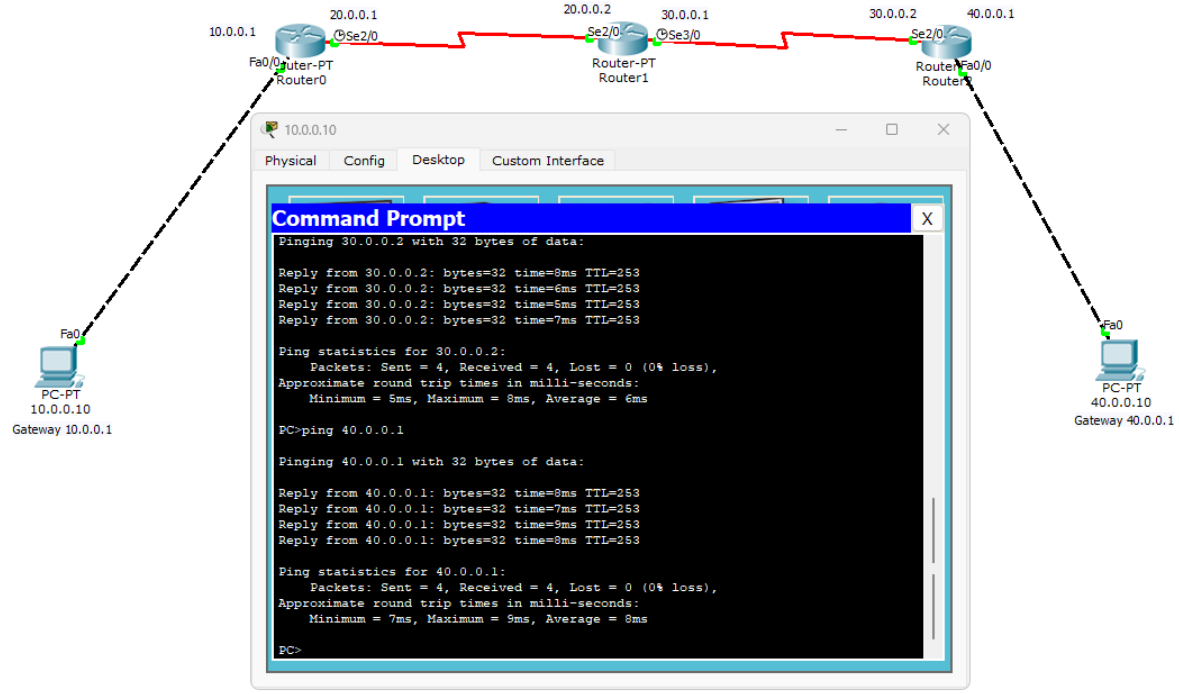
Screenshot(s) of the output:

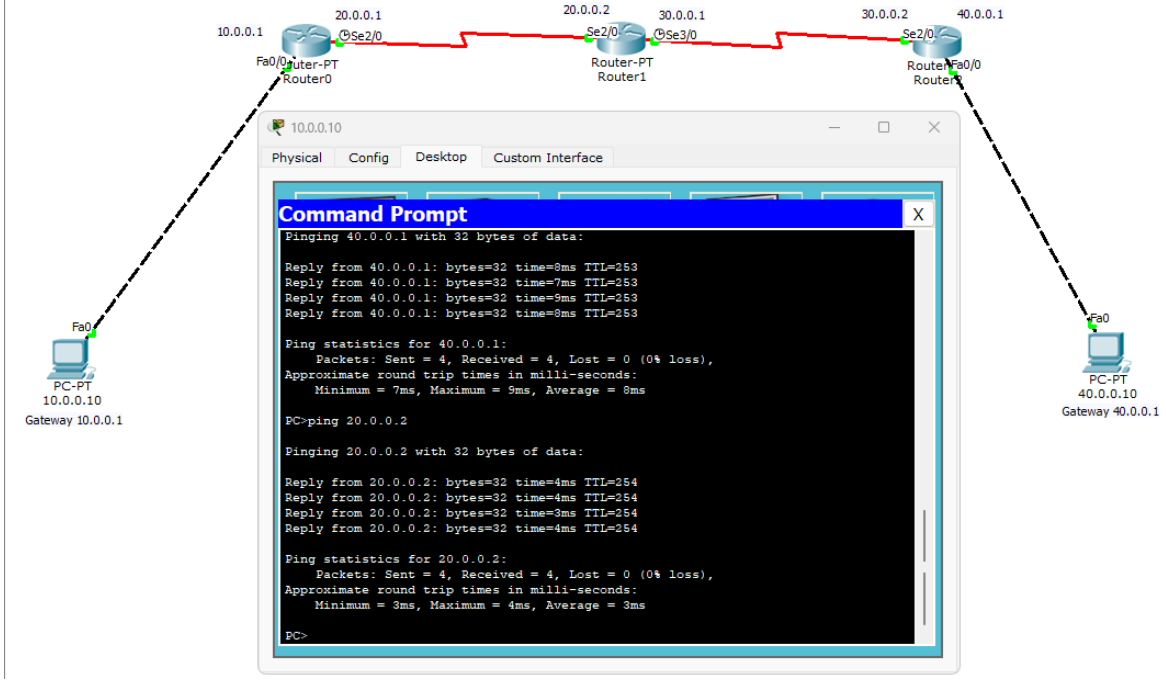





10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
40.0.0.1
Se2/0
Fa0/0
Router2
Router2
Physical
Config
CLI
IOS Command Line Interface
Gateway of last resort is not set
C 40.0.0.0/8 is directly connected, FastEthernet0/0
Router#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%SYS-5-CONFIG_I: Configured from console by console
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router#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 30.0.0.1 to network 0.0.0.0
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1
Router#
Copy
Paste
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
40.0.0.1
Se2/0
Fa0/0
Router2
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Reply from 30.0.0.1: bytes=32 time=4ms TTL=254
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 4ms, Average = 4ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=8ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=5ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 8ms, Average = 6ms
PC>
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=8ms TTL=253
Reply from 30.0.0.2: bytes=32 time=6ms TTL=253
Reply from 30.0.0.2: bytes=32 time=5ms TTL=253
Reply from 30.0.0.2: bytes=32 time=7ms TTL=253
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 8ms, Average = 6ms
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=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=7ms TTL=253
Reply from 40.0.0.1: bytes=32 time=9ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 9ms, Average = 8ms
PC>


10.0.0.1
20.0.0.1
Se2/0
Fa0/0
Router-PT
Router0
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
30.0.0.2
Se2/0
40.0.0.1
Fa0/0
Fa0
PC-PT
10.0.0.10
Gateway 10.0.0.1
Fa0
PC-PT
40.0.0.10
Gateway 40.0.0.1
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Reply from 40.0.0.1: bytes=32 time=7ms TTL=253
Reply from 40.0.0.1: bytes=32 time=9ms TTL=253
Reply from 40.0.0.1: bytes=32 time=8ms TTL=253
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 9ms, Average = 8ms
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Reply from 20.0.0.2: bytes=32 time=3ms TTL=254
Reply from 20.0.0.2: bytes=32 time=4ms TTL=254
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 3ms, Maximum = 4ms, Average = 3ms
PC>

Experiment 4: Dynamic Host Configuration Protocol (DHCP)

Question: Configure DHCP within a LAN and outside LAN.

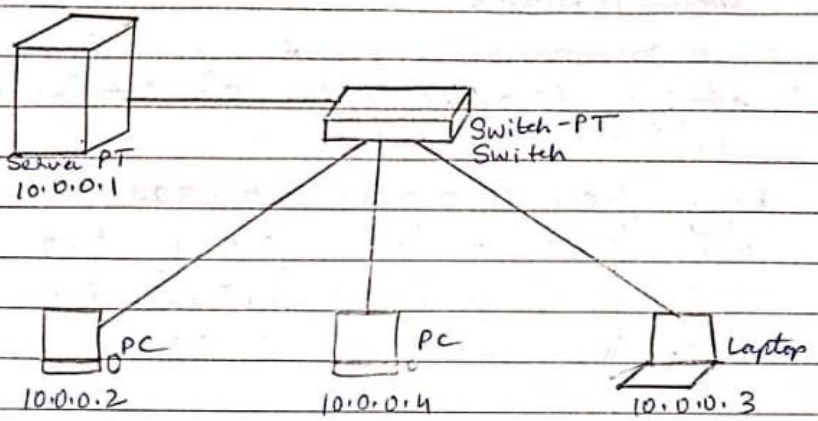
Observation writeup:

13/11/24 LAB NO-5 Date: Page: 23

DHCP

AIM: Design a DHCP within LAN and outside LAN

TOPOLOGY:



PROCEDURE:

1.] Place a switch, a server, two PCs & one laptop

2.] Connect using Automatically Choose Connection Type

3.] click Server → Desktop → IP Configuration - set IP address as 10.0.0.1 and default gateway as 10.0.0.0

4.] click on PC → Desktop → IP Configuration → Select DHCP : automatic/dynamic IP address is assigned

5.] click Server → Config → DHCP →
Pool Name : SwitchOne
Default Gateway : 10.0.0.0
Start IP address : 10.0.0.3
Max Users : 100

Observation:

1.] The IP addresses were allocated dynamically to the end devices

2.] Ping from one end device to another device was successful

→ 10.0.0.2

ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

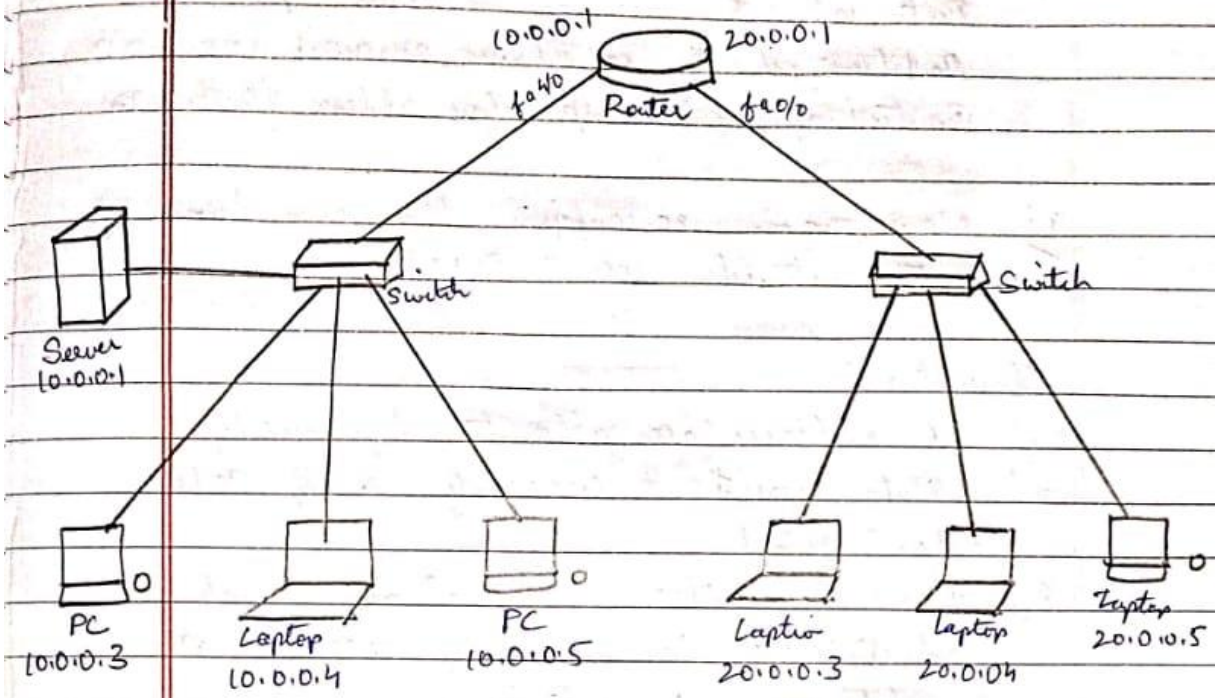
Reply from 10.0.0.3: byte = 32 t = 1ms TTL = 128
Reply from 10.0.0.3: byte = 32 t = 0ms TTL = 128
Reply from 10.0.0.3: byte = 32 t = 0ms TTL = 128
Reply from 10.0.0.3: byte = 32 t = 7ms TTL = 128

ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approx round trip time in milli seconds:
min = 0ms, max = 7ms, avg = 2ms

- This eliminates the need for network administrator to manually assign IP addresses to each device

Bafna Gold
Date: Page:

TOPOLOGY : TWO Networks



PROCEDURE:

1.] Retain the first network
2.] Add another network with a switch & 3 end devices
3.] Add a router to connect the 2 networks
4.] Edit the IP configuration of Server
- change IP of Server to 10.0.0.2
5.] Edit SwitchOne Pool :
Default Gateway : 10.0.0.1
Start IP : 10.0.0.3
6.] Add another Pool : Switch Two :
Default Gateway : 20.0.0.1
Start IP : 20.0.0.3
7.] Configure the router:
Router > enable
Router # config terminal
Router (config) # interface fastethernet 4/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # ip helper-address 10.0.0.2

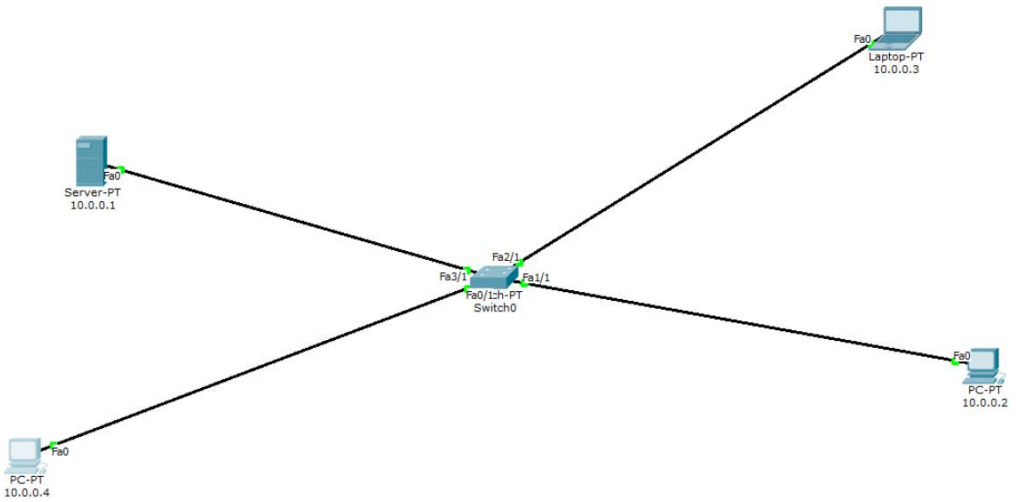
8] Similarly configure the second network
Router(config)# interface fastEthernet 4/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# ip helper-address 10.0.0.2

9] Now again reconfigure the end devices of both networks using DHCP

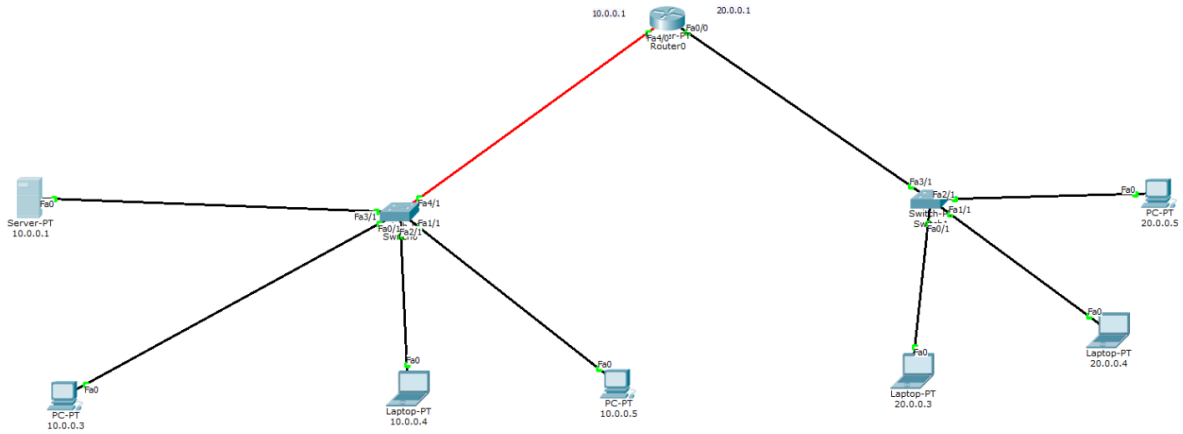
Observations:
1] IP addresses are allocated dynamically
2] Data was sent successfully among PCs when pinged
3] This eliminates the need of network administrator to manually assign IP addresses to each device

13/11

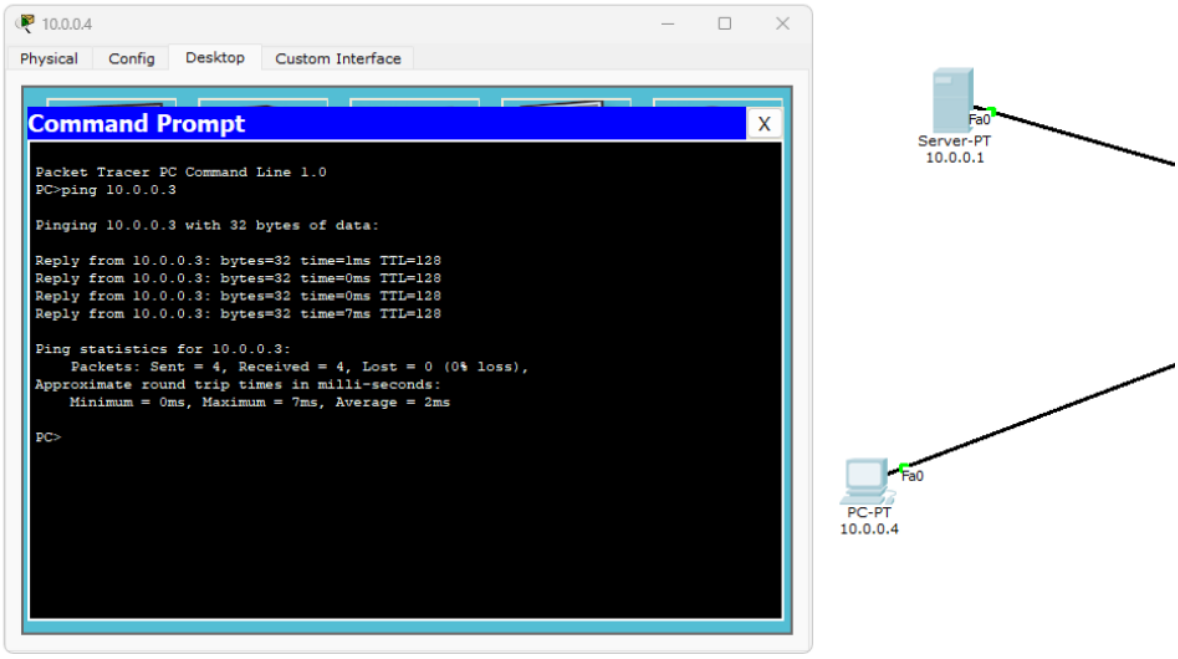
Screenshot of the topology:
Within Lan:

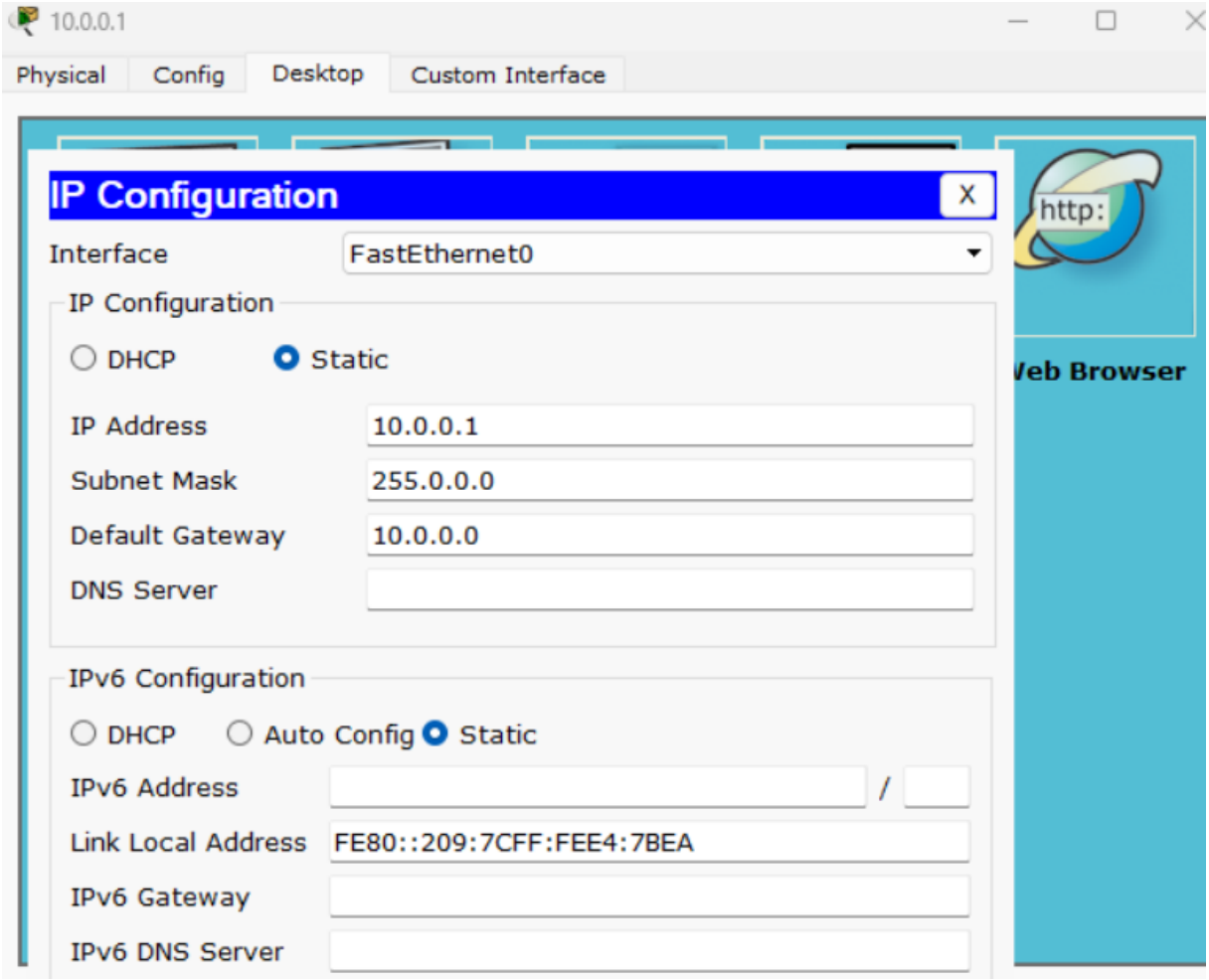


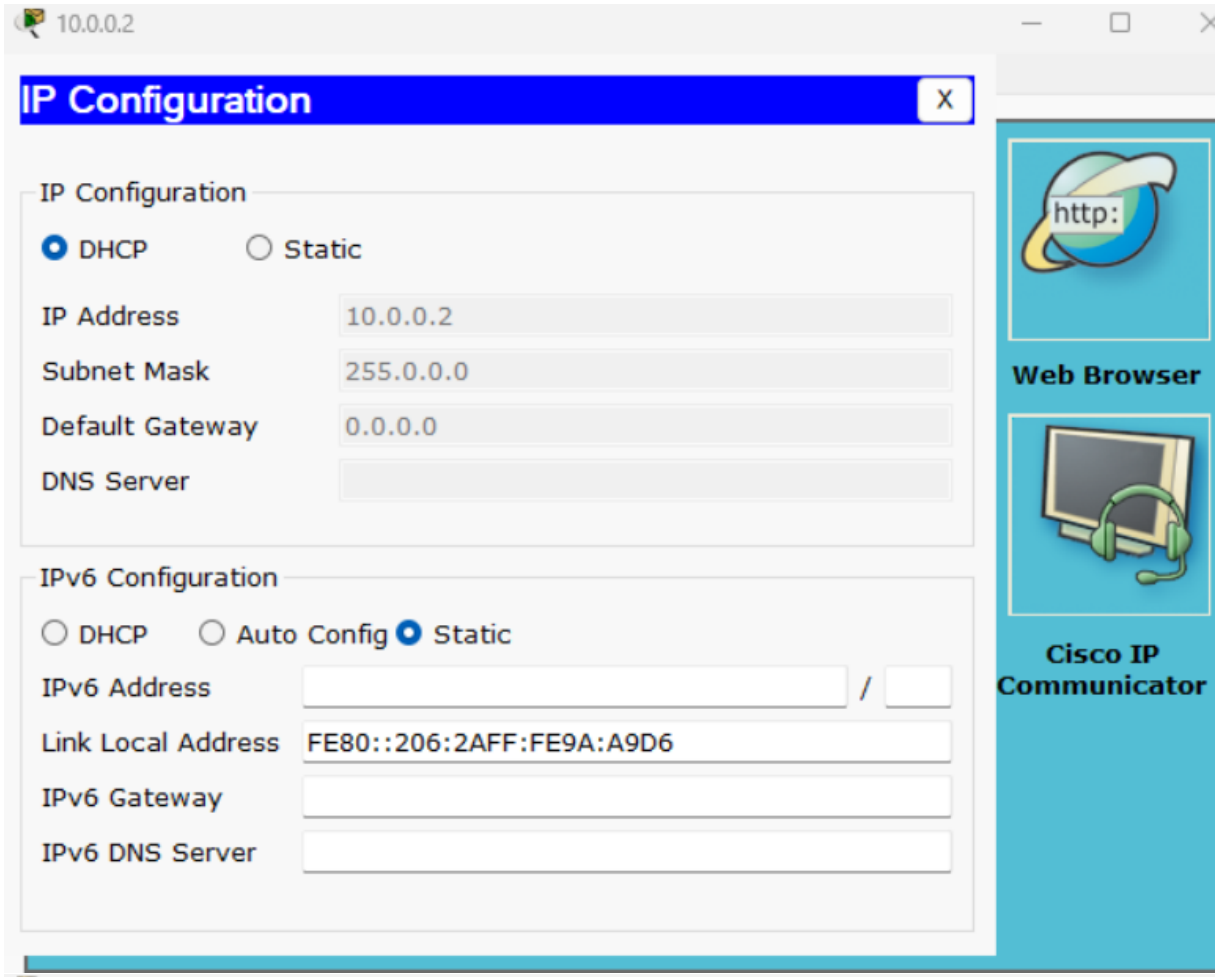
Outside Lan:

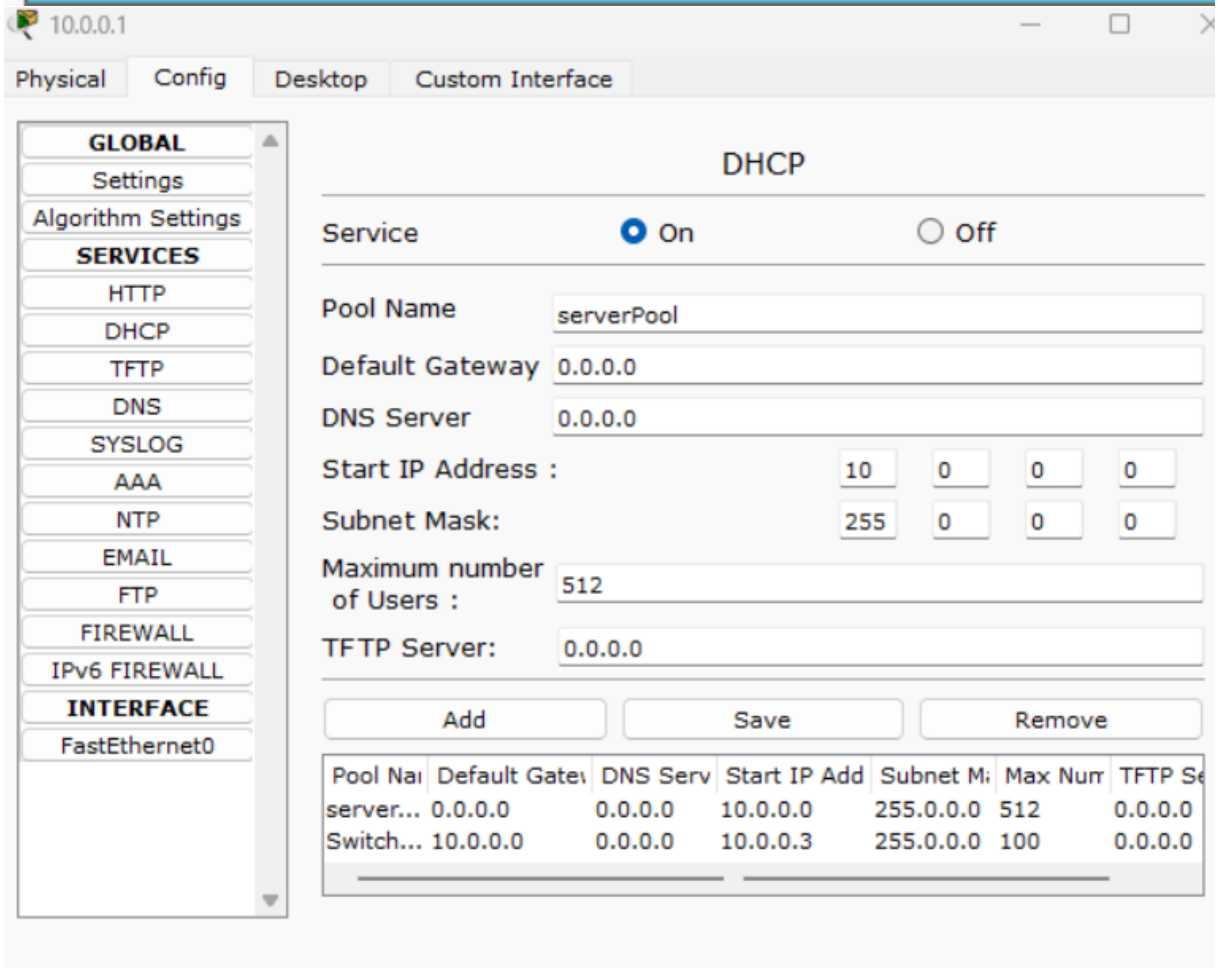


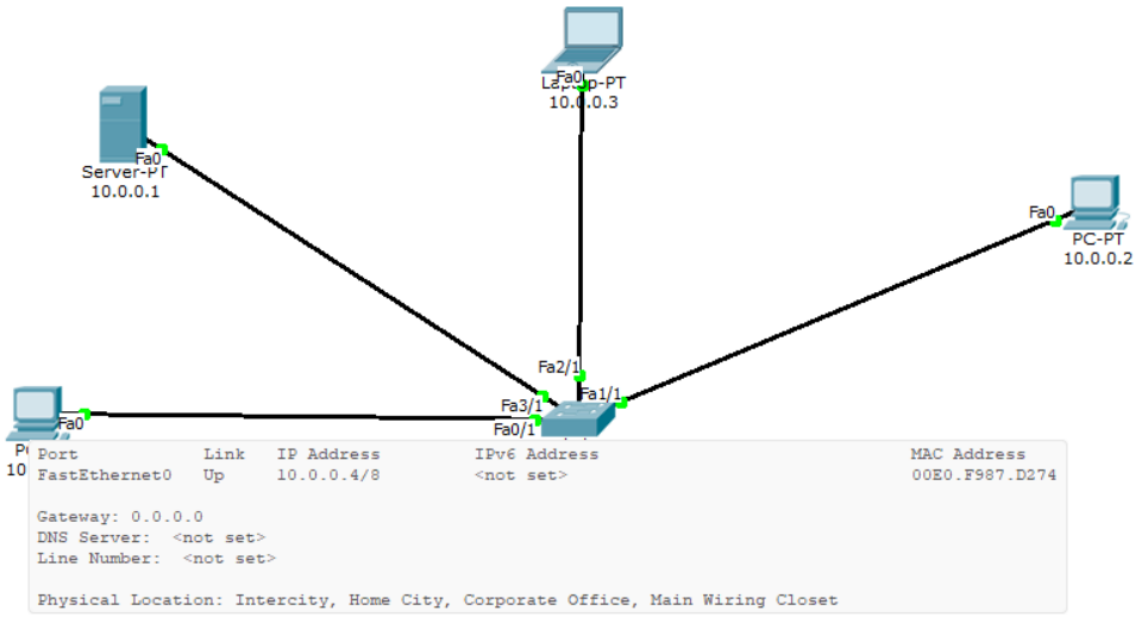
Screenshot(s) of the output:






10.0.0.2
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
10.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
0.0.0.0
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
/
Link Local Address
FE80::206:2AFF:FE9A:A9D6
IPv6 Gateway
IPv6 DNS Server
http:
Web Browser
Cisco IP Communicator

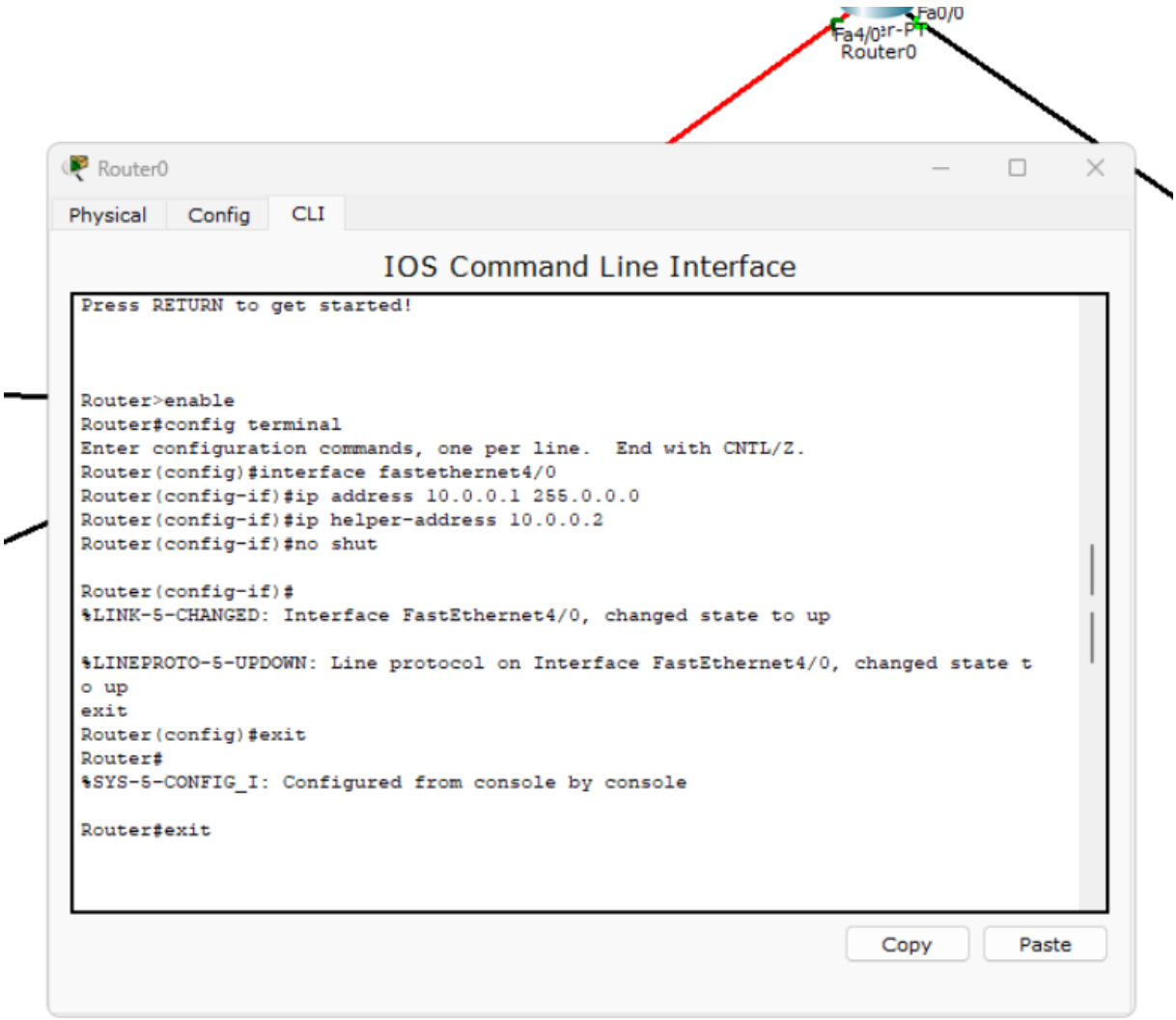

10.0.0.1
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
SERVICES
HTTP
DHCP
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
FIREWALL
IPv6 FIREWALL
INTERFACE
FastEthernet0
DHCP
Service
On
Off
Pool Name
serverPool
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
Start IP Address :
10
0
0
0
Subnet Mask:
255
0
0
0
Maximum number of Users :
512
TFTP Server:
0.0.0.0
Add
Save
Remove
Pool Nai
Default Gatev
DNS Serv
Start IP Add
Subnet Mi
Max Num
TFTP Se
server...
0.0.0.0
0.0.0.0
10.0.0.0
255.0.0.0
512
0.0.0.0
Switch...
10.0.0.0
0.0.0.0
10.0.0.3
255.0.0.0
100
0.0.0.0

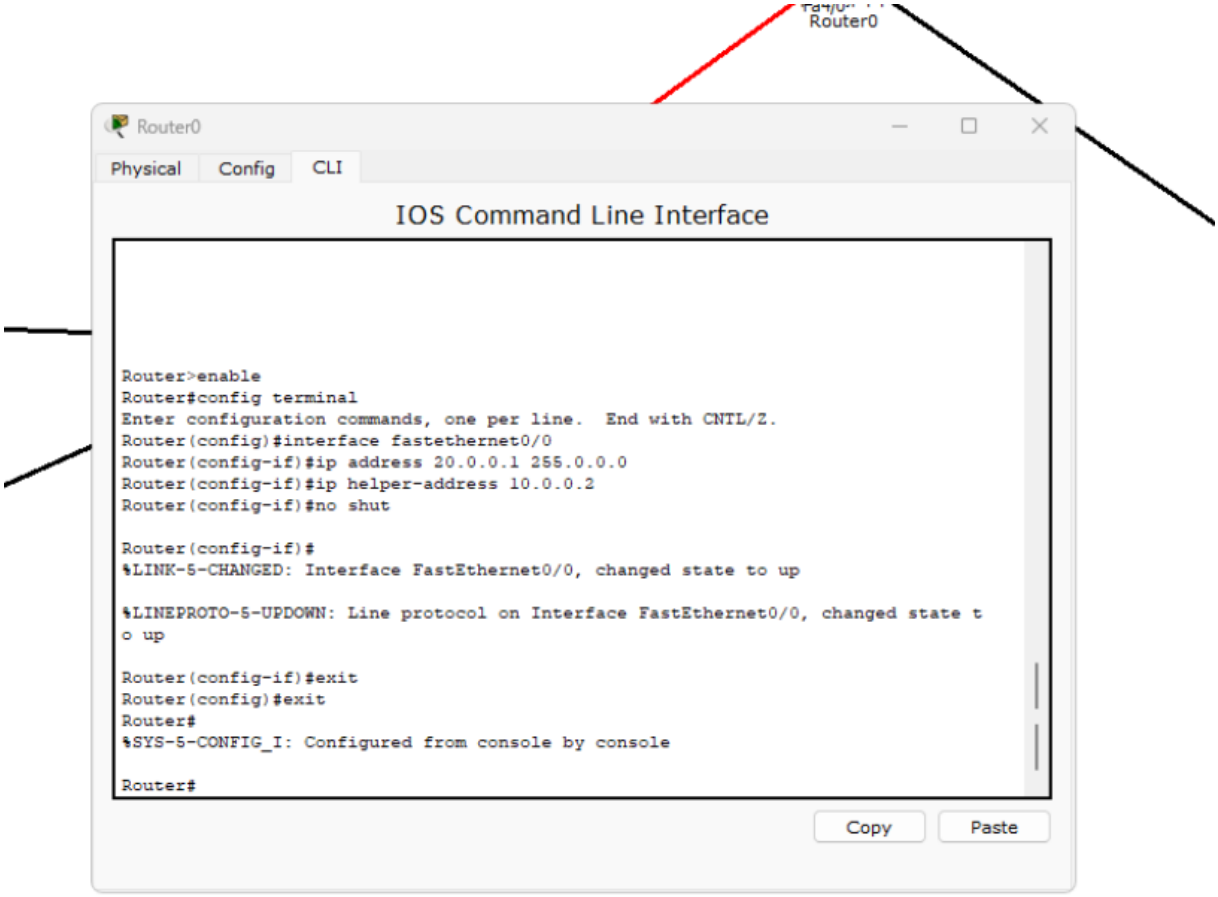


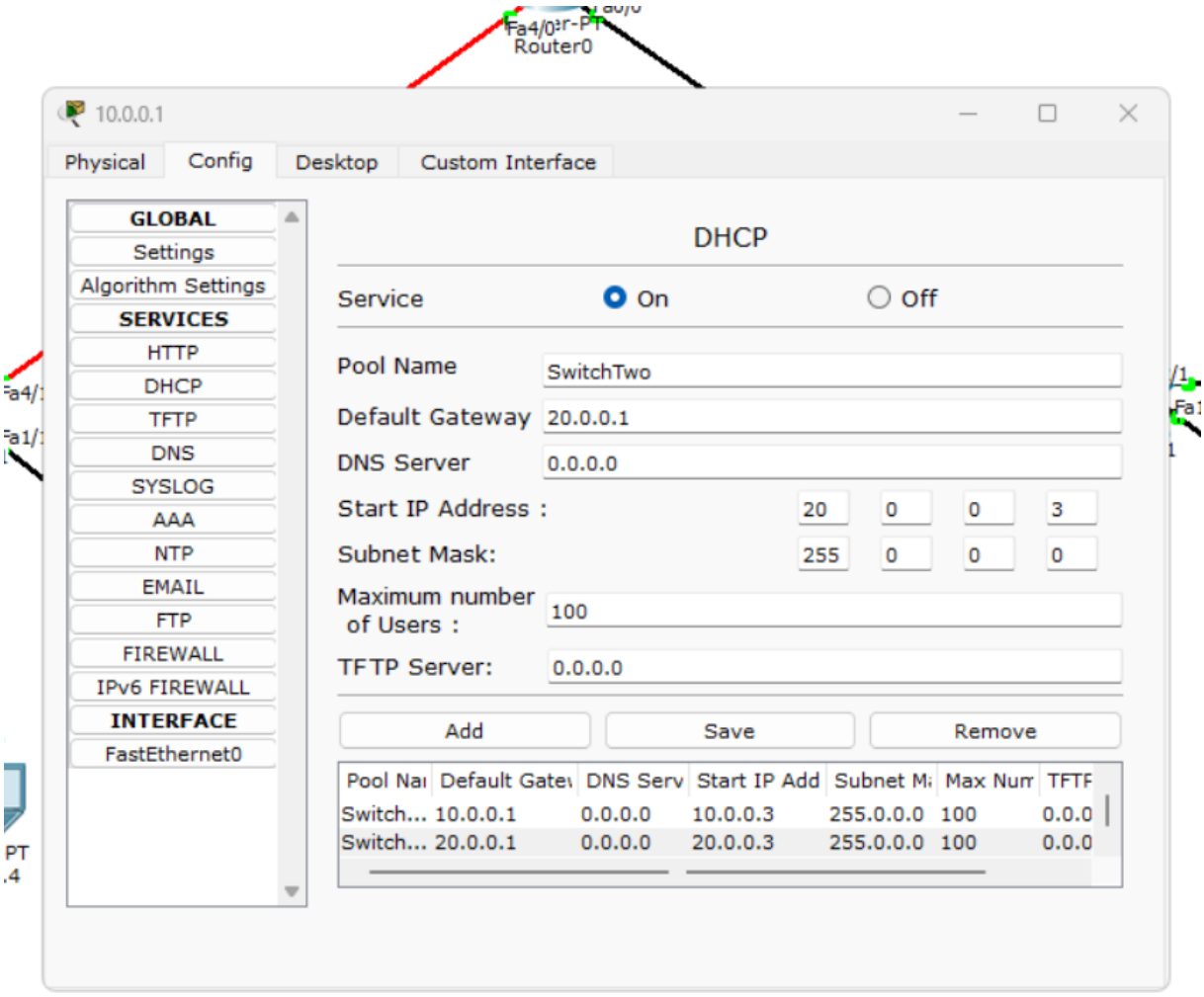
Port	Link	IP Address	IPv6 Address	MAC Address
FastEthernet0	Up	10.0.0.4/8	<not set>	00E0.F987.D274

Gateway: 0.0.0.0
DNS Server: <not set>
Line Number: <not set>

Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet



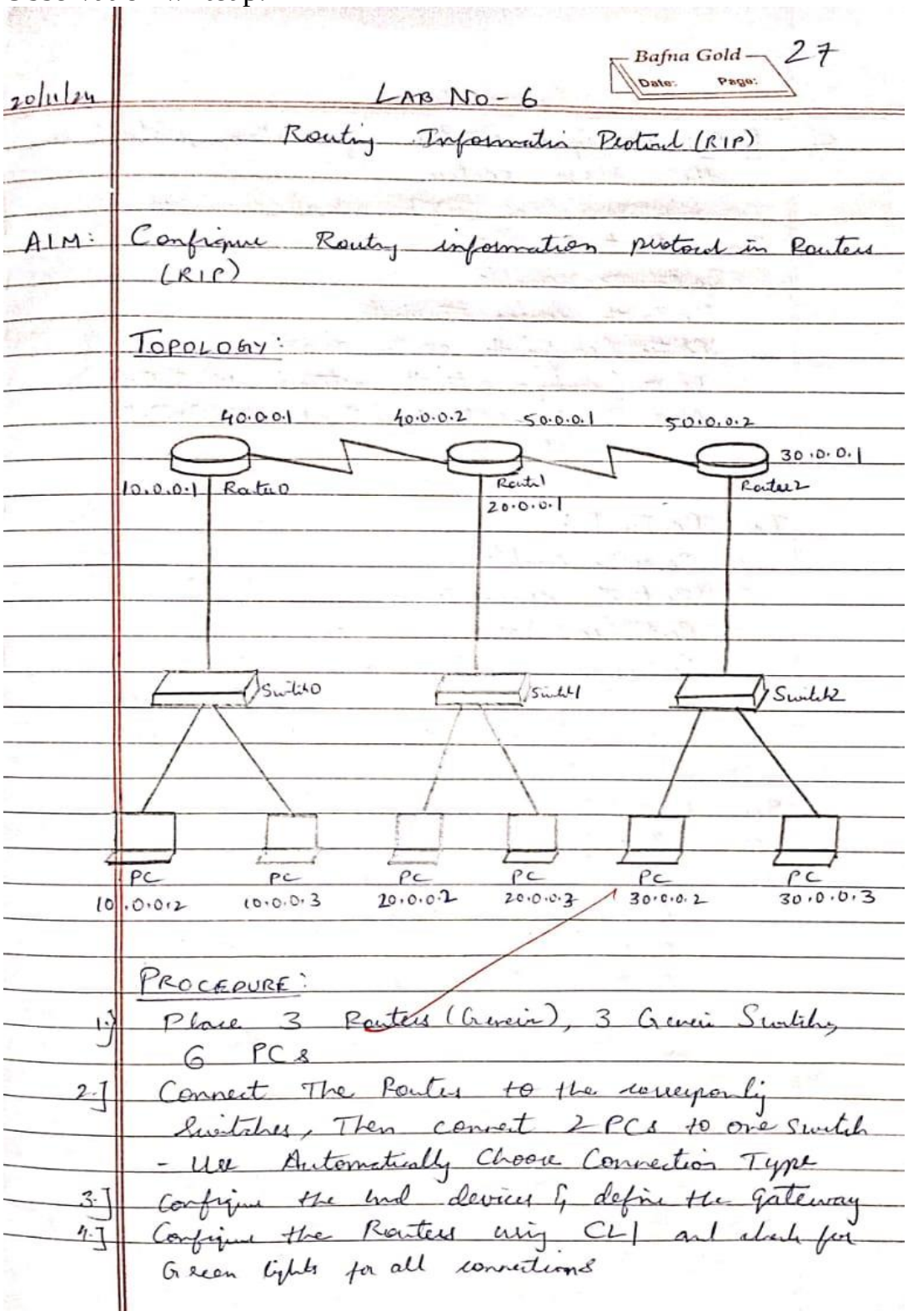

Router0
Router0
Physical
Config
CLI
IOS Command Line Interface
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state t
o up
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#
Copy
Paste


Router0
10.0.0.1
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
SERVICES
HTTP
DHCP
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
FIREWALL
IPv6 FIREWALL
INTERFACE
FastEthernet0
DHCP
Service
On
Off
Pool Name
SwitchTwo
Default Gateway
20.0.0.1
DNS Server
0.0.0.0
Start IP Address :
20 0 0 3
Subnet Mask:
255 0 0 0
Maximum number of Users :
100
TFTP Server:
0.0.0.0
Add
Save
Remove
Pool Nai
Default Gatev
DNS Serv
Start IP Add
Subnet Mi
Max Num
TFTF
Switch... 10.0.0.1 0.0.0.0 10.0.0.3 255.0.0.0 100 0.0.0
Switch... 20.0.0.1 0.0.0.0 20.0.0.3 255.0.0.0 100 0.0.0

Experiment 5: Routing Information Protocol (RIP)

Question: Configure RIP routing Protocol in Routers

Observation writeup:



5.] RIP: Configure Routing information protocol in the three routers:

In Router0:

```
Router > enable
Router # config terminal
Router (config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 40.0.0.0
```

In Router1:

```
Router> enable
Router# config terminal
Router(config)# router rip
Router(config-router)# network 40.0.0.0
Router(config-router)# network 50.0.0.0
Router(config-router)# network 20.0.0.0
```

In Router2:

```
Router > enable
Router# config terminal
Router(config)# router rip
Router(config-router)# network 50.0.0.0
Router(config-router)# network 30.0.0.0
```

Bafna Gold
Date: Page:

Observations:

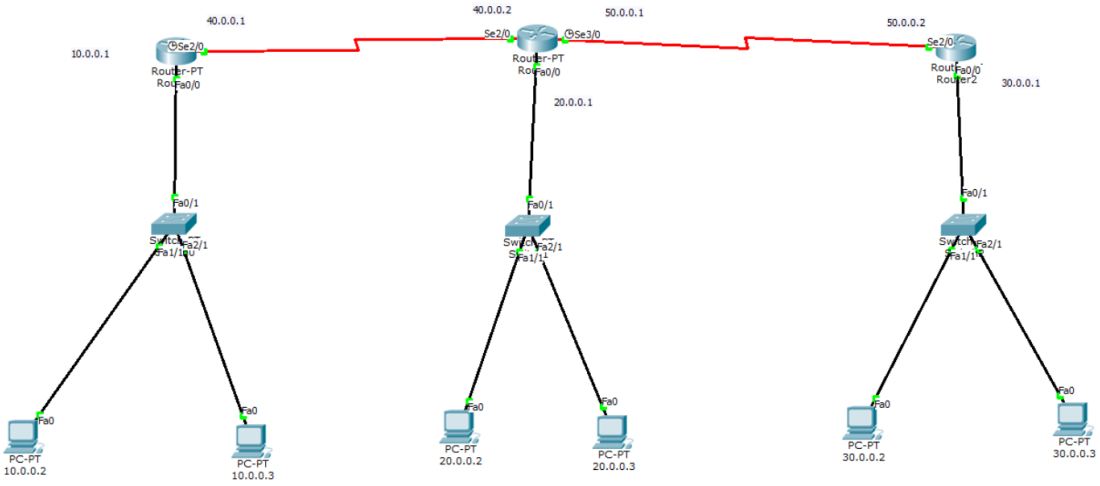
1] Before Routij information protocol:
In Router 2:
Router# show ip route
C 30.0.0.0/8 is directly connected
C 50.0.0.0/8 is directly connected

2] After Routij infomation protocol:
In Router 2:
Router# show ip route
R 10.0.0.0/8 via 50.0.0.1
R 20.0.0.0/8 via 50.0.0.1
C 30.0.0.0/8 directly connected
R 40.0.0.0/8 via 50.0.0.1
C 50.0.0.0/8 directly connected

3] - Before RIP pinging from diffrnt network failed

4] After RIP, we were successfully able to ping across networks as connections were established through rip

Screenshot of the topology:



Screenshot(s) of the output:

Router1

Physical | Config | CLI

```
Continue with configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#enable
Router#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state t
o up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial3/0
Router(config-if)#ip address 50.0.0.1 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
Router(config)#exit
```

```
Router>enable
Router#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    30.0.0.0/8 is directly connected, FastEthernet0/0
C    50.0.0.0/8 is directly connected, Serial2/0
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#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

R    10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:09, Serial2/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:09, Serial2/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:09, Serial2/0
C    50.0.0.0/8 is directly connected, Serial2/0
Router#
```

Router2

Physical Config CLI

IOS Command Line Interface

```
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#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

R    10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:09, Serial2/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:09, Serial2/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:09, Serial2/0
C    50.0.0.0/8 is directly connected, Serial2/0
Router#
```

Copy Paste

10.0.0.3

Physical Config Desktop Custom Interface

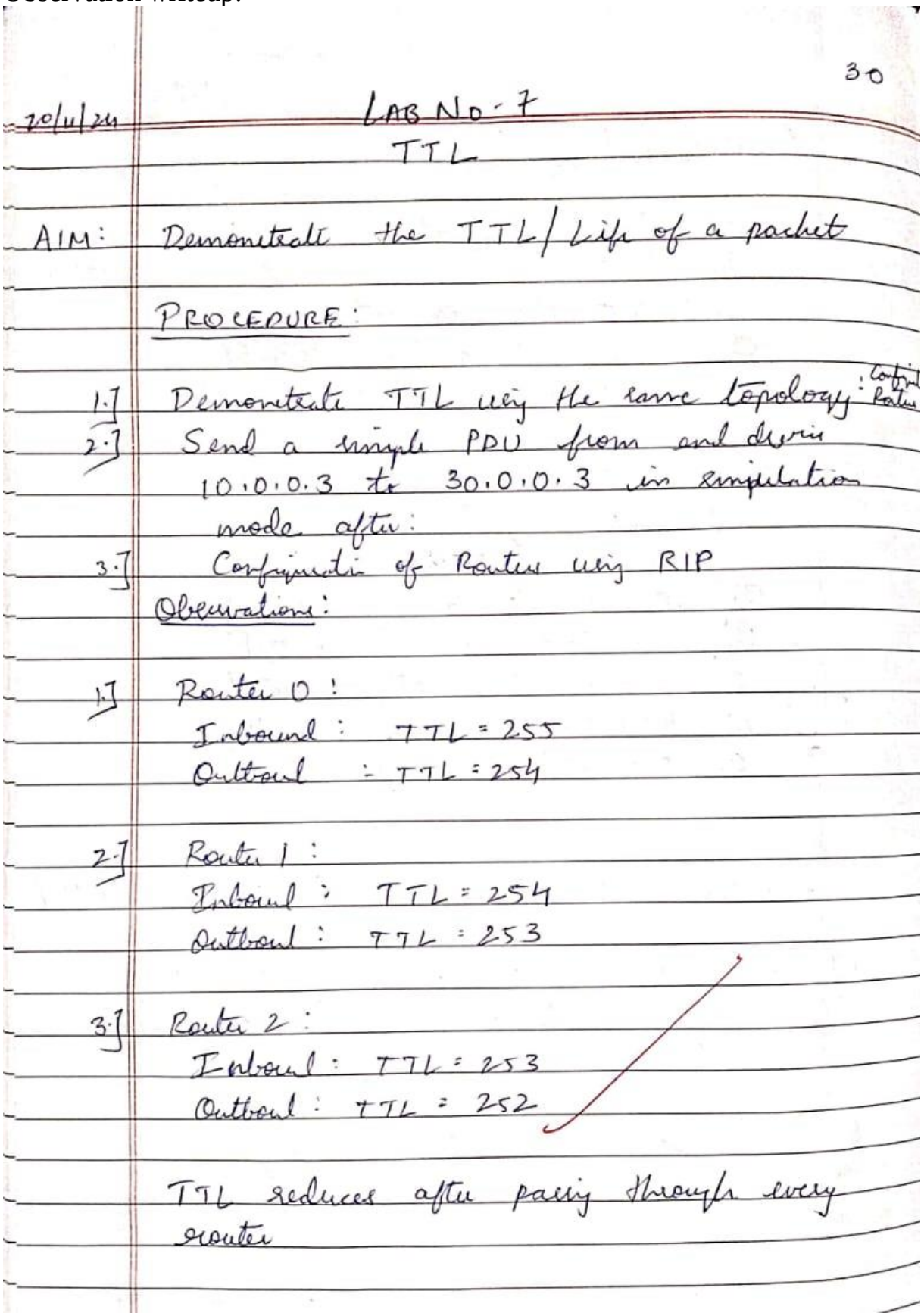
Command Prompt X

```
Packet Tracer PC Command Line 1.0
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=10ms TTL=125
Reply from 30.0.0.3: bytes=32 time=9ms TTL=125
Reply from 30.0.0.3: bytes=32 time=6ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 10ms, Average = 7ms

PC>
```

Experiment 6: Time To Live (TTL)

Question: Demonstrate the TTL/ Life of a Packet

Observation writeup:



30

20/4/24

LAB No. 7

TTL

AIM: Demonstrate the TTL/ Life of a packet

PROCEDURE:

1.] Demonstrate TTL using the same topology: Configure Routers

2.] Send a simple PDU from end device 10.0.0.3 to 30.0.0.3 in simulation mode after:

3.] Configuration of Routers using RIP

Observations:

1.] Router 0 :
Inbound : TTL = 255
Outbound : TTL = 254

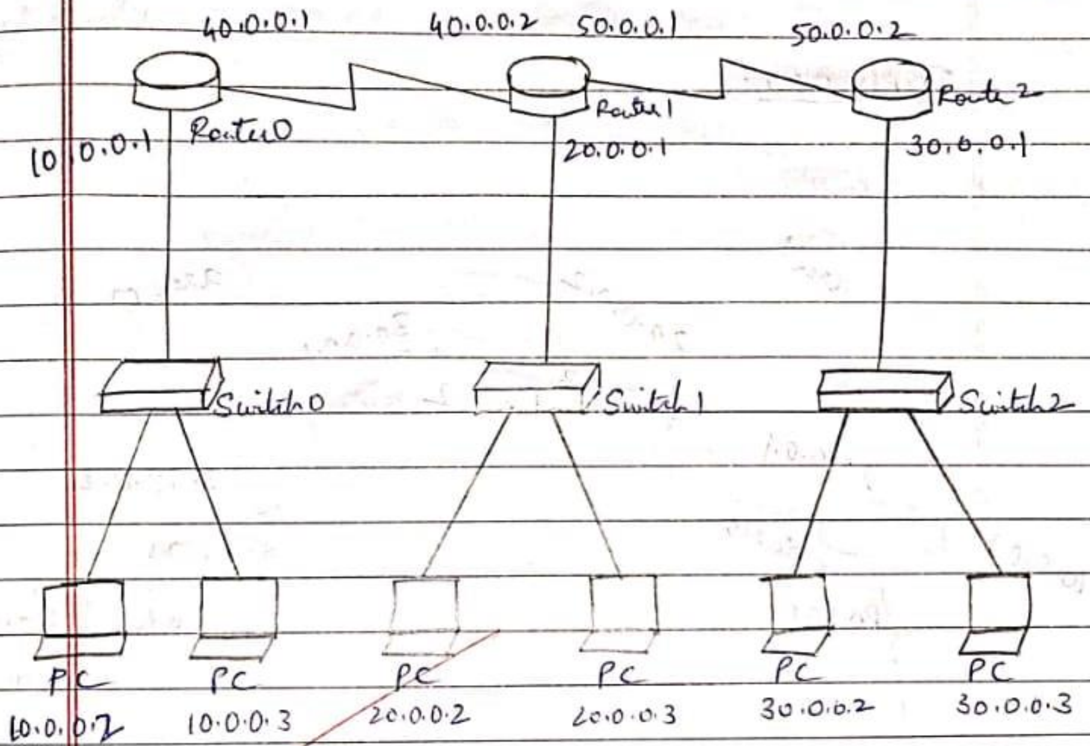
2.] Router 1 :
Inbound : TTL = 254
Outbound : TTL = 253

3.] Router 2 :
Inbound : TTL = 253
Outbound : TTL = 252

TTL reduces after passing through every router

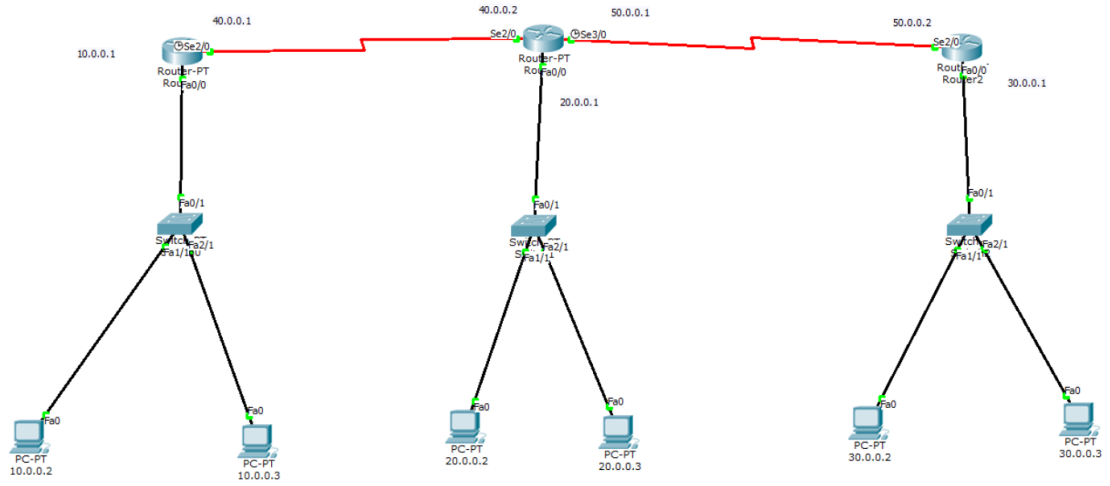
Bafna Gold — Date: Page:

TOPOLOGY:



20/11/24

Screenshot of the topology:



Screenshot(s) of the output:

PDU Information at Device: Router0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

Ethernet II

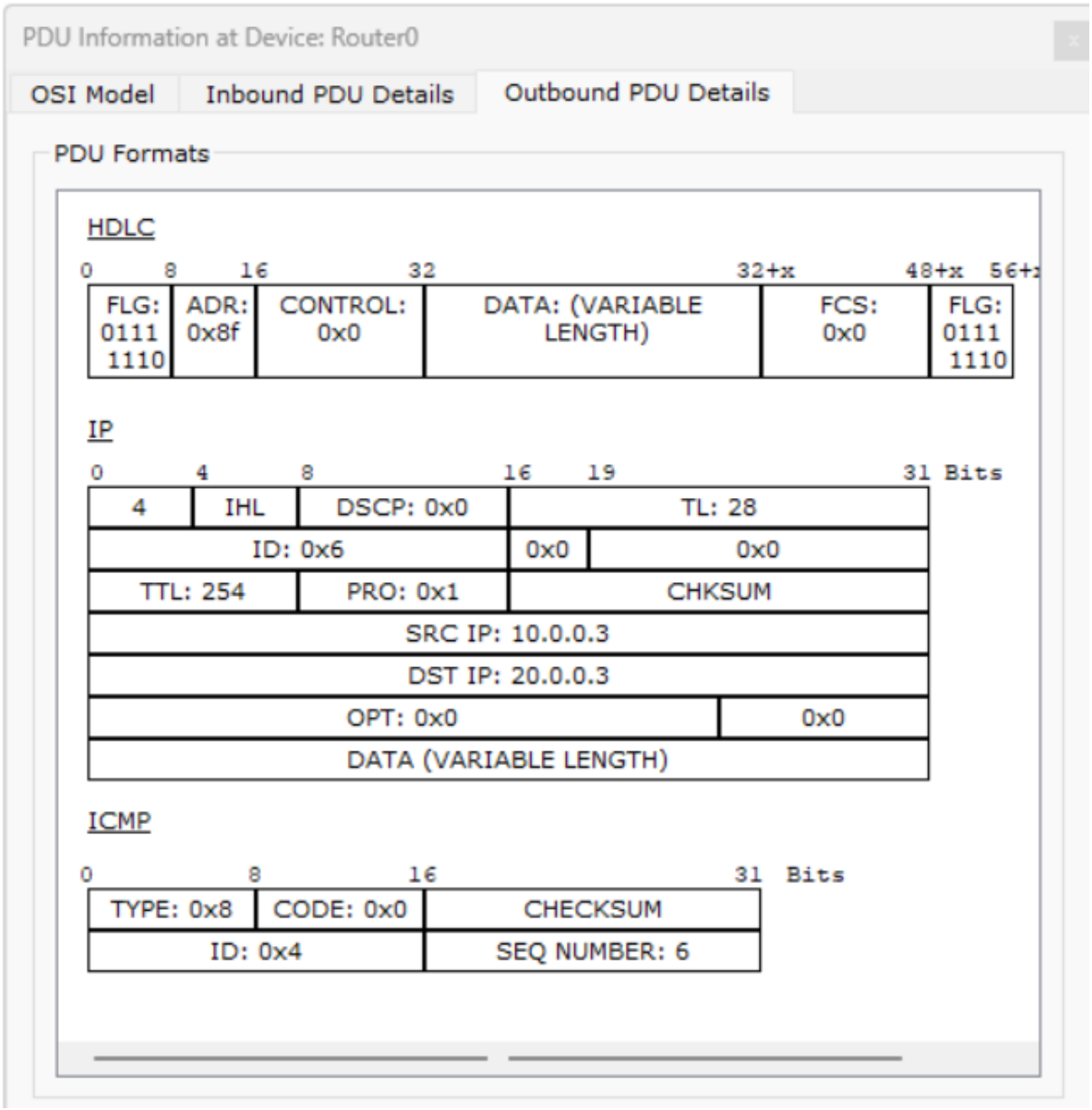
0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 0090.2181.972B	SRC MAC: 000A.4135.9310	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x6			0x0	0x0	
TTL: 255		PRO: 0x1	CHKSUM		
SRC IP: 10.0.0.3					
DST IP: 20.0.0.3					
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
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: 0x6
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.3
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 6

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: 0x7			0x0	0x0	
TTL: 253		PRO: 0x1	CHKSUM		
SRC IP: 10.0.0.3					
DST IP: 30.0.0.3					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x5		SEQ NUMBER: 7	

PDU Information at Device: Router2

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 0000.0C37.A59A	SRC MAC: 0030.A357.D9C7	
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x7			0x0	0x0	
TTL: 252		PRO: 0x1	CHKSUM		
SRC IP: 10.0.0.3					
DST IP: 30.0.0.3					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

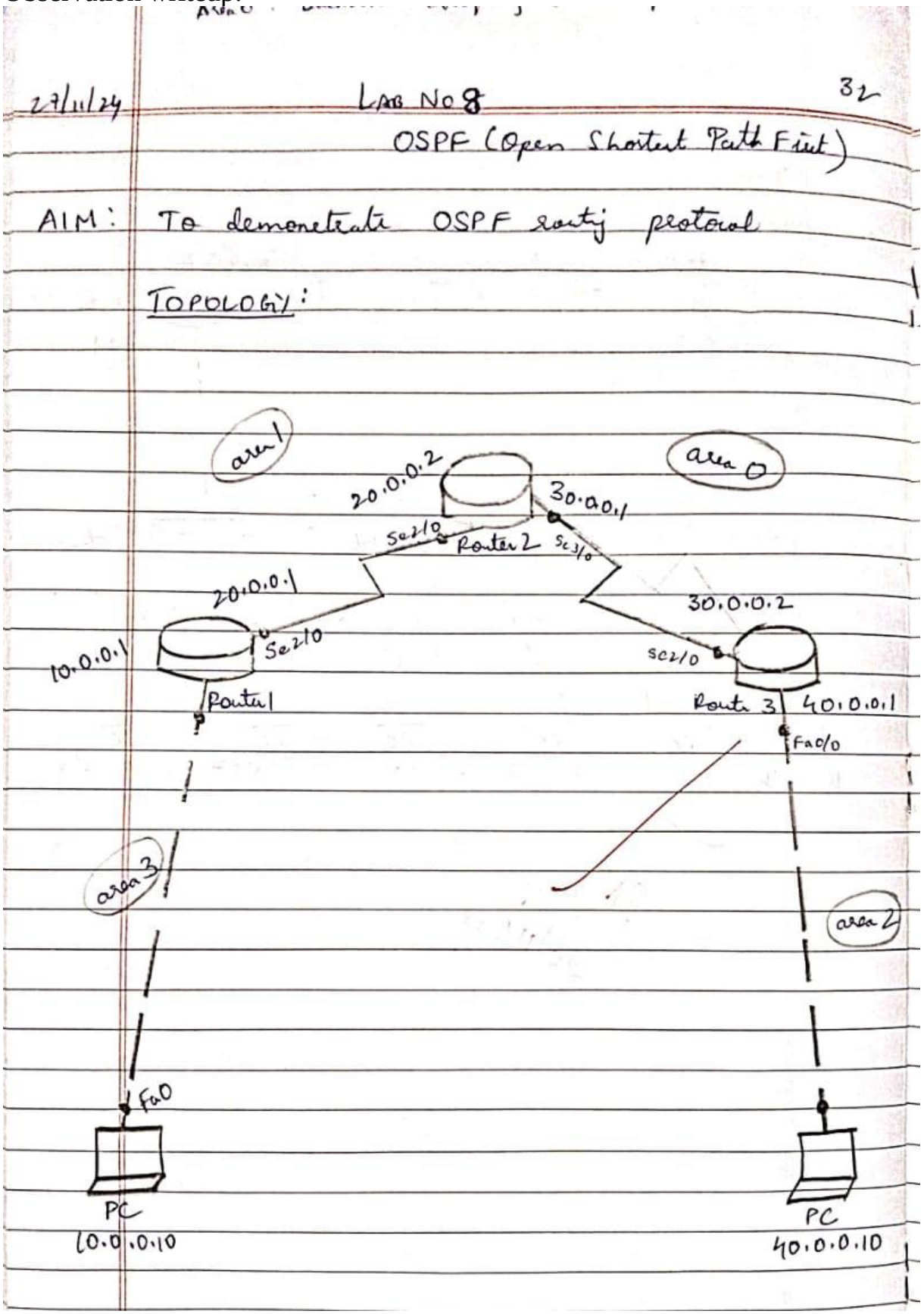
ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	

Experiment 7: Open Shortest Path First (OSPF)

Question: Configure OSPF routing protocol

Observation writeup:



Bafna Gold
Date: Page:

PROCEDURE:

1] Place three routers & 2 end devices, connect them as per topology & assign ip addresses

2] Set IP addresses to Routers:

Router 1:

```
Router(config)# interface serial 2/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
Router(config-if)# encapsulation ppp
Router(config-if)# clock rate 64000
```

Router 2:

```
Router(config)# interface serial 2/0
Router(config)# ip address 20.0.0.2 255.0.0.0
```

- Here it doesn't have (D) clock symbol, so no clock rate

```
Router(config)# encapsulation ppp
```

- Similarly set for the other 3 networks

3] OSPF Routing:

Router 1:

```
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
```

- Similarly set for other 2 Routers

2.2.2.2

3.3.3.3

4.] Loopback:

Router1:

Router(config)# interface loopback 0

Router(config-if)# ip address 172.16.1.252 255.255.0.0

Router(config-if)# no shutdown

Similarly for other two Routers:

172.16.1.253

172.16.1.254

5.] Ping device 40.0.0.10 from 10.0.0.10

OBSERVATIONS:

1.] Routing Table:

- For Router3:

Router# show ip route

O IA 10.0.0.0/8 via 30.0.0.1, Serial2/0

O IA 20.0.0.0/8 via 30.0.0.1, Serial2/0

30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

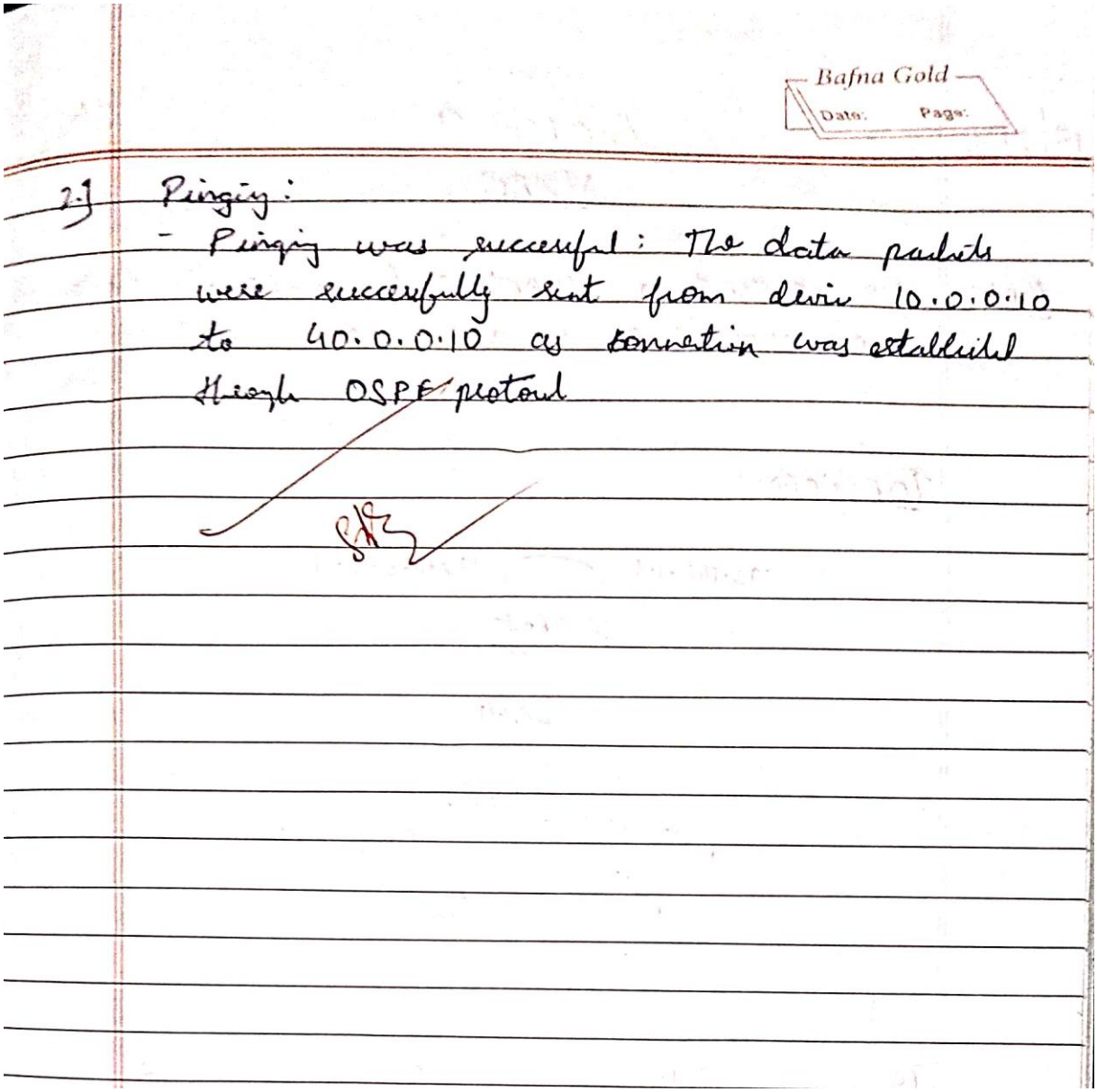
C 30.0.0.0/8 is directly connected, Serial2/0

C 30.0.0.1/32 is directly connected, Serial2/0

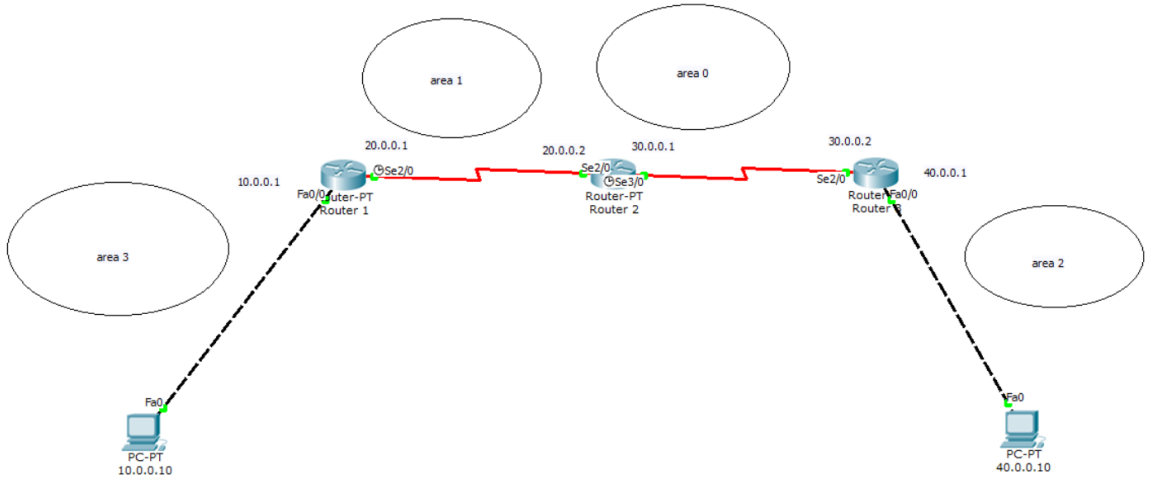
C 40.0.0.0/8 is directly connected, FastEthernet0/0

C 172.16.0.0/16 is directly connected, Loopback0

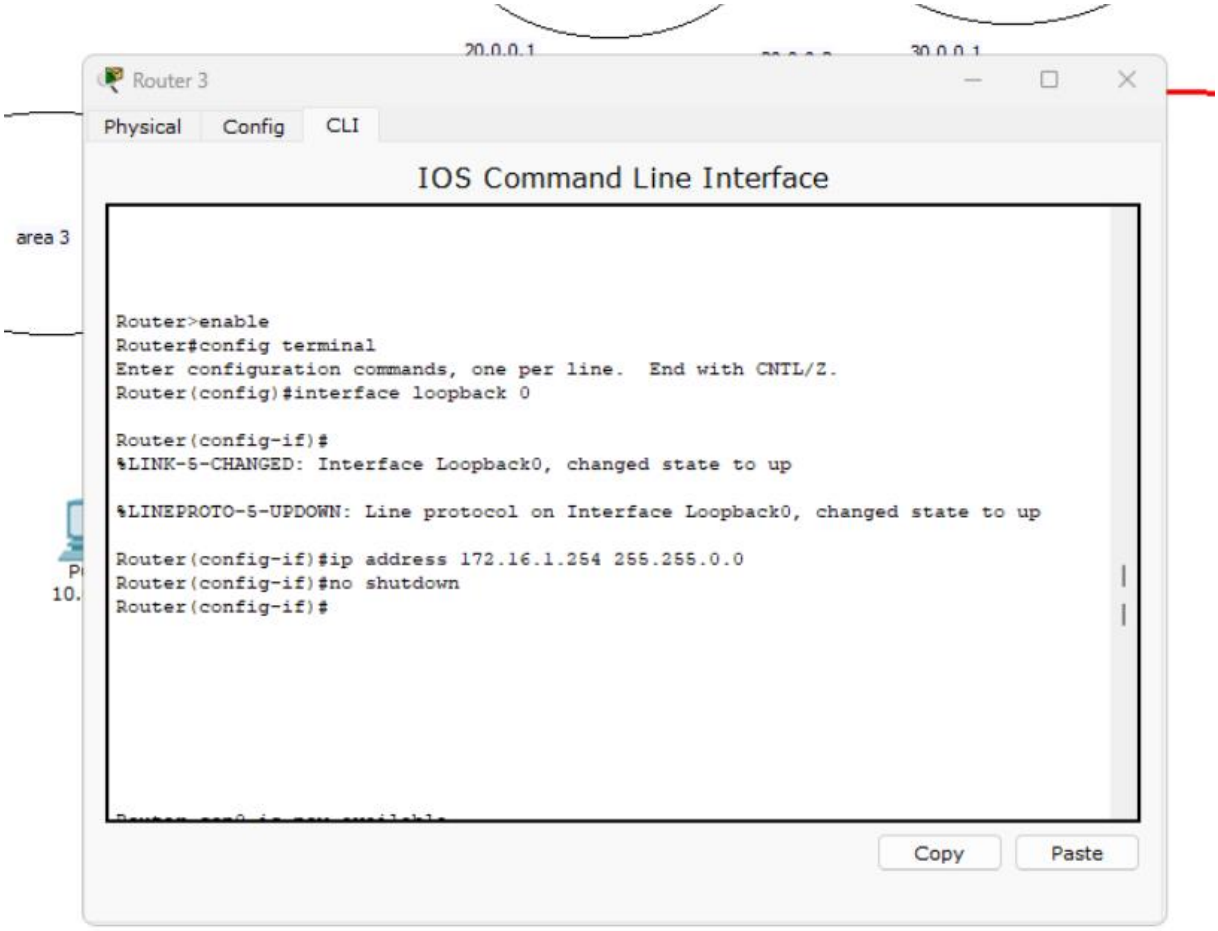
- All the routers know about each other.
- Thus connection successfully established through OSPF protocol

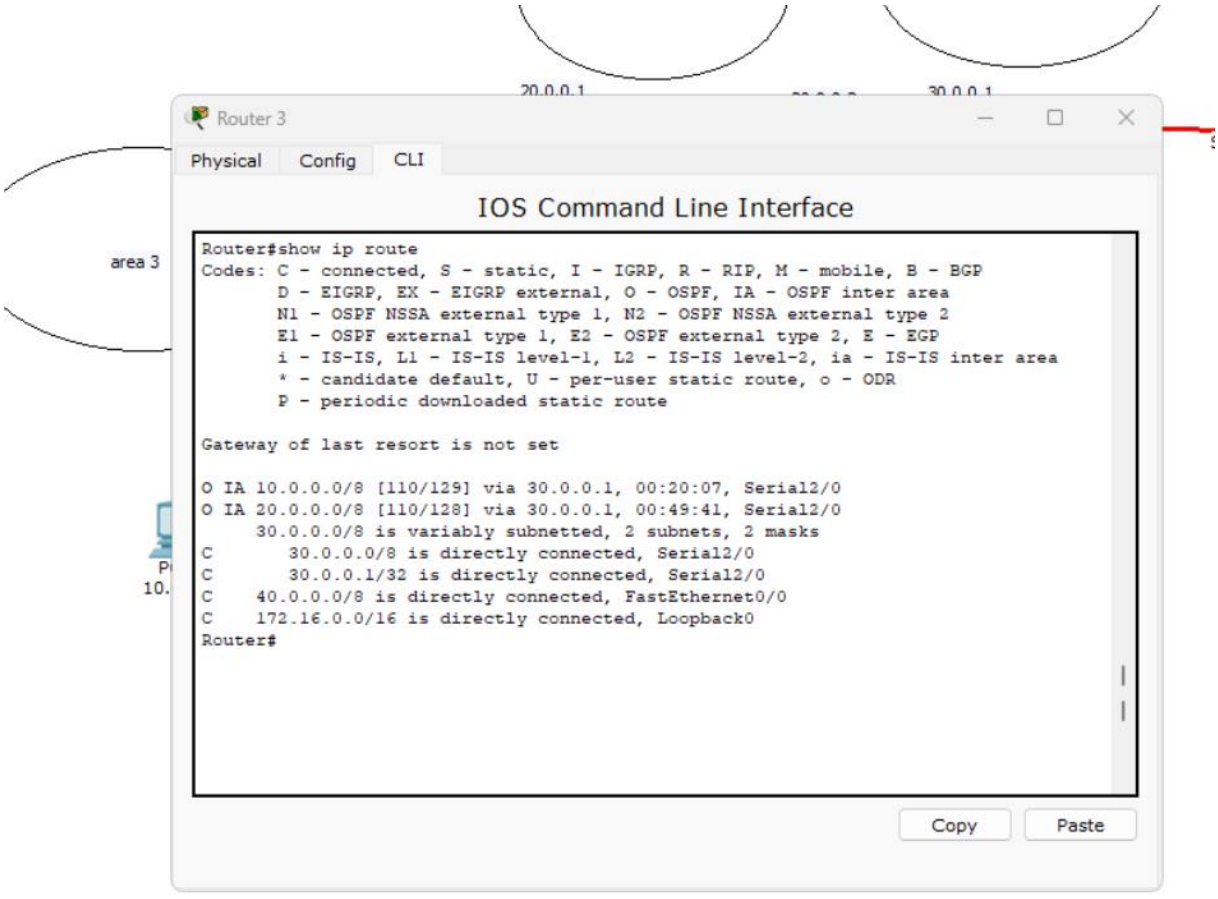


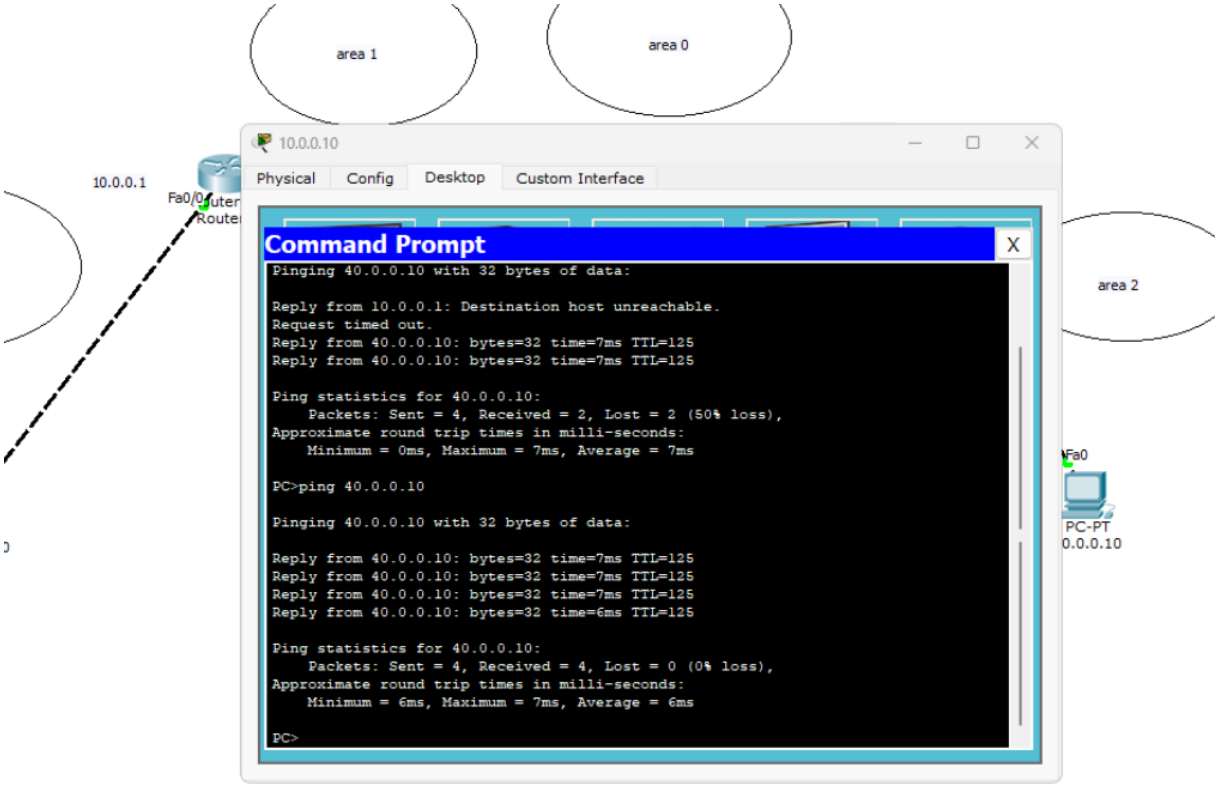
Screenshot of the topology:



Screenshot(s) of the output:






area 1
area 0
10.0.0.1
Fa0/0
Router
10.0.0.10
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 10.0.0.1: Destination host unreachable.
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
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, 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=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
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 = 7ms, Average = 6ms
PC>
area 2
Fa0
PC-PT
0.0.0.10

Experiment 8: Virtual Lan (VLAN)

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

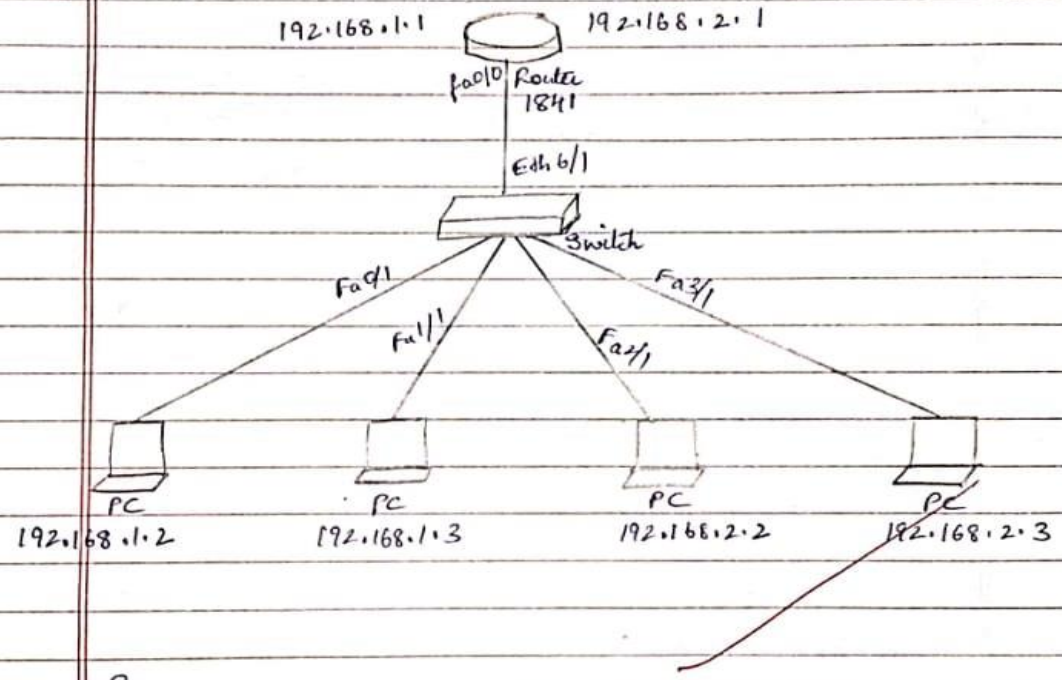
Observation writeup:

18/12/24 LAB NO 9 36

VLAN

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

TOPOLOGY:



PROCEDURE:

1.] Place a 1841 Router, a switch and 4 PCs

2.] Connect the four PCs to the switch via fastethernet

3.] Since only 4 fastethernet ports are available in the Switch, we have to add an ethernet port

4.] - Switch off the Power button of Switch

Bafna Gold
Date. Page.

- Add the ethernet port to the switch
- Switch on the power button
- Connect the router to the Switch via Ethernet 6/1

5.] In the switch, go to Config Tab and
- Select VLAN Database
- Give VLAN number say 2
- Give VLAN name say 'cseise'
- Add it to the Database

6.] Select the Switch:
- Go to Config
- Go to Ethernet 6/1 ie connected to Router
- Make it the trunk

7.] Configure the PCs as shown in the topology

8.] Select Switch:
- Go to Config
- Go to FastEthernet 2/1
- Set VLAN number as 2 ie 'cseise'
- Similarly set VLAN 2 for FastEthernet 3/1 interface

9.] Configure the Router:

```
Router(config)#interface fastEthernet 0/0
Router(config-if)# ip address 192.168.1.1
                                255.255.255.0
Router(config-if)# no shut
Router(config-if)# exit
```

Now, to configure the router's VLAN interface:

```
Router(config)# interface fastethernet 0/0.1
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ip address 192.168.2.1
                                 255.255.255.0
Router(config-subif)# no shut
Router(config-subif)# exit
```

16.] Ping devices within the same VLAN and to devices of different VLAN

OBSERVATIONS:

1.] When devices are pinged within same VLAN:
- Pinging 192.168.1.3 from 192.168.1.2
- The data packet doesn't go to the router
- The switch forwards the packet without the need of the router

2.] When a device pings a device of another VLAN:
- Pinging 192.168.2.3 from 192.168.1.2
- The data packet's journey is as follows:

192.168.1.2 → Switch → Router

192.168.2.3 ← Switch ←

3.] VLANs divide a single switch into multiple logical switches
- Devices in one VLAN cannot directly communicate with devices in another VLAN without a router

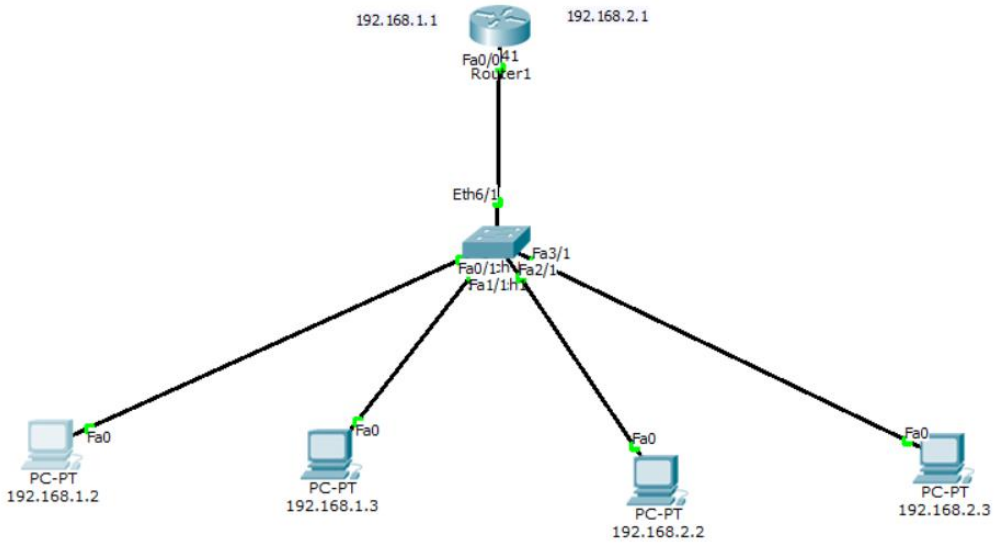
Bafna Gold
Date: Page:

4.] Traffic Isolation :
- Each VLAN maintains its own broadcast domain
- Broadcasts sent by devices in one VLAN do not reach devices in another VLAN

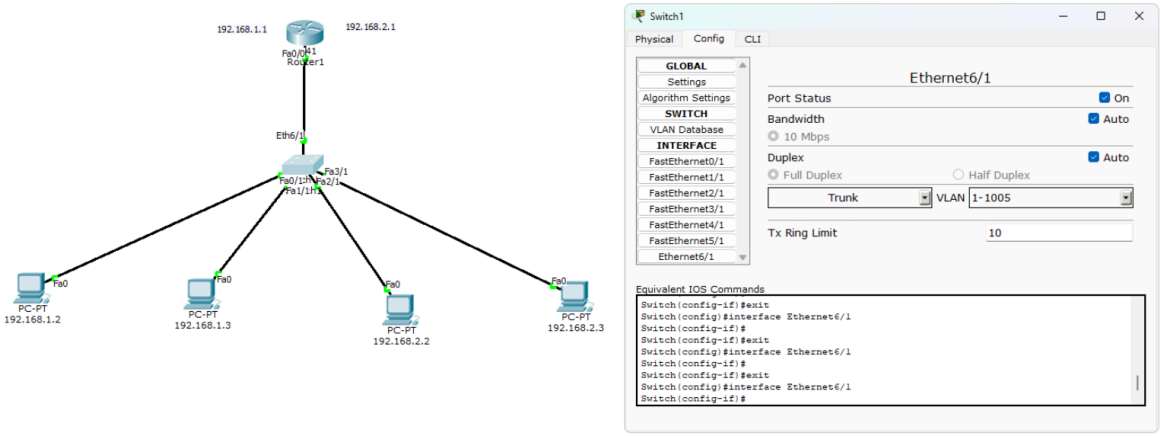
5.] VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk

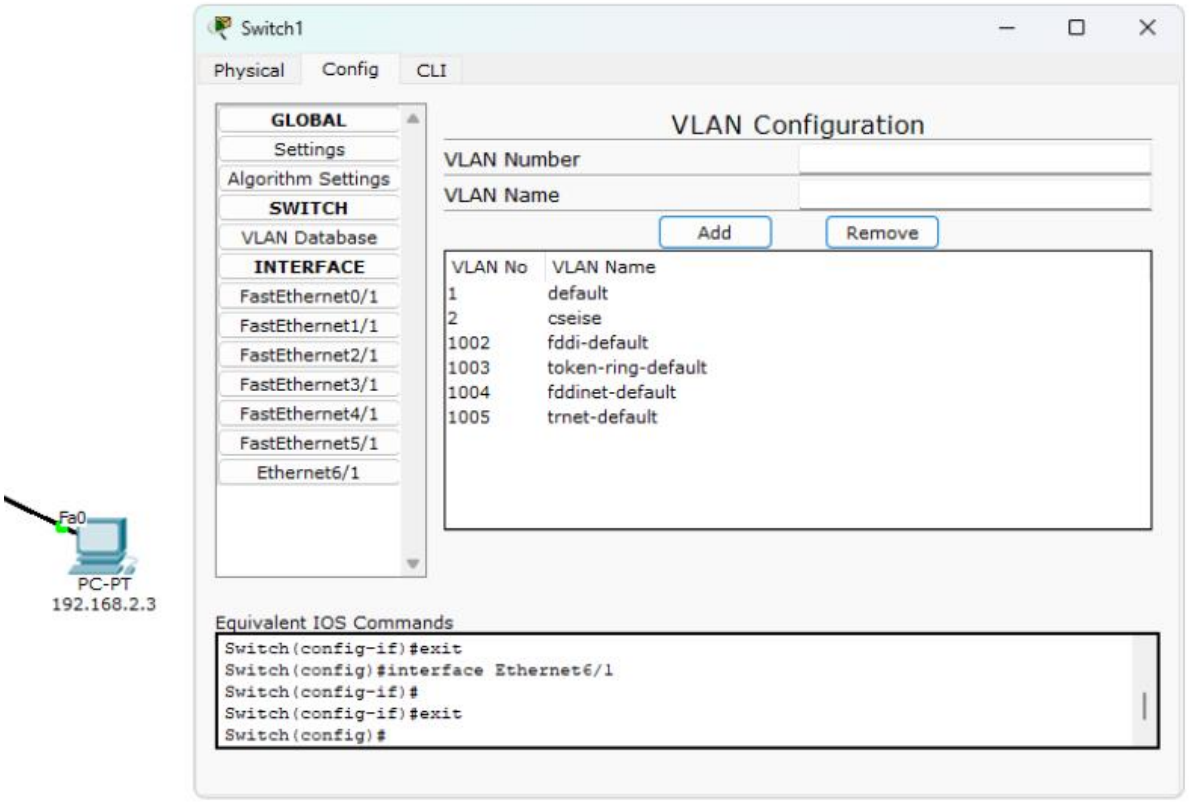
- This is done by adding an additional header information called tag to the Ethernet frame - VLAN tagging

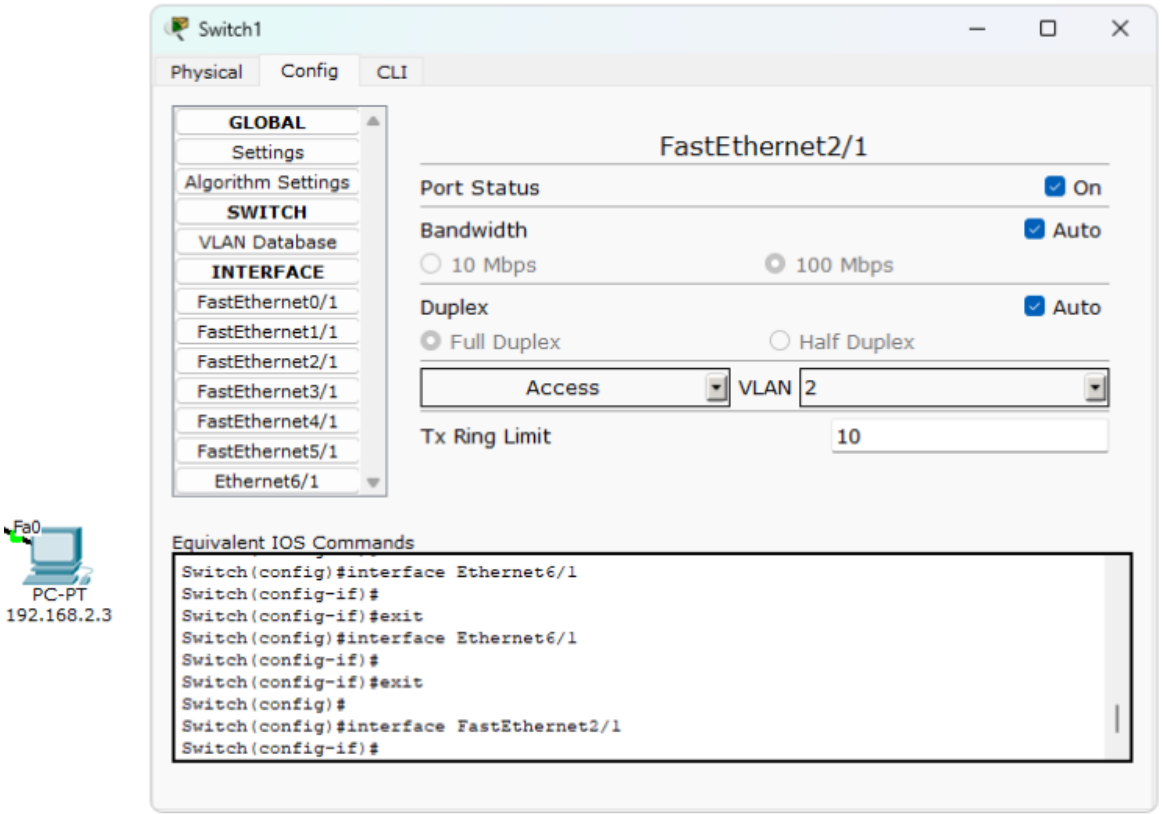
Screenshot of the topology:

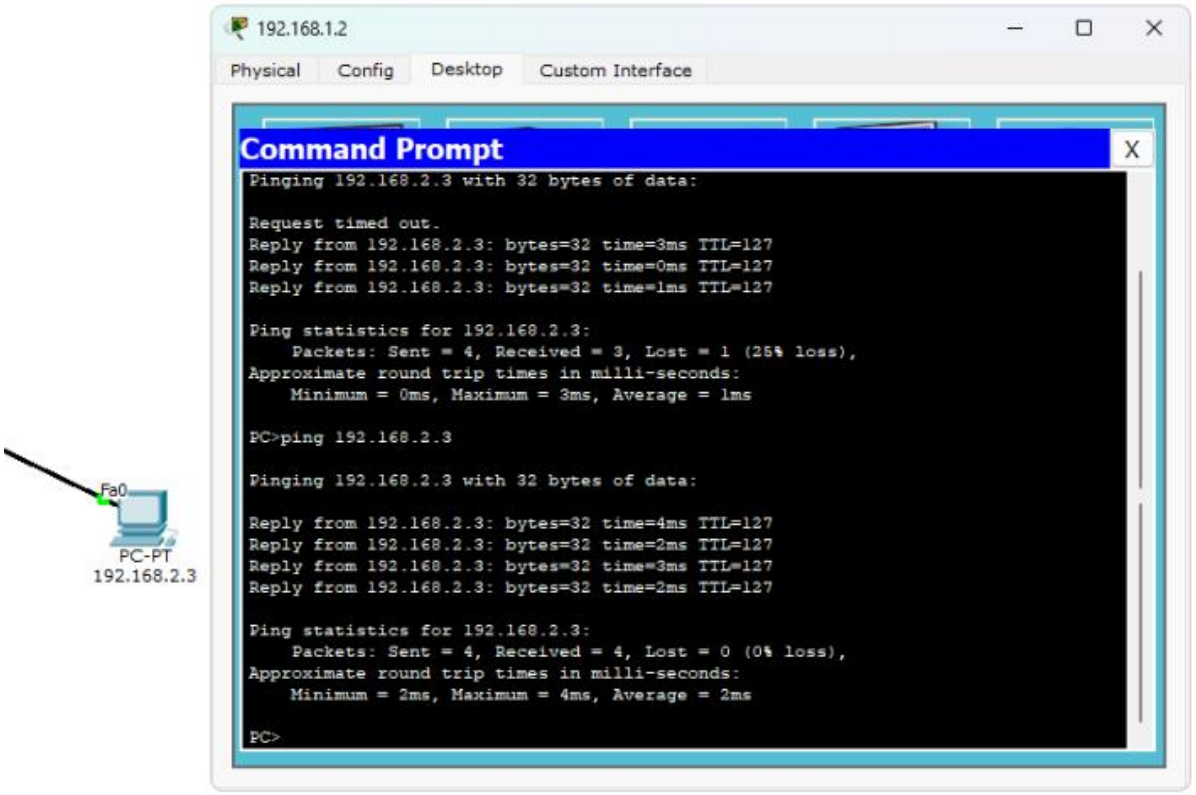


Screenshot(s) of the output:




Switch1
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet5/1
Ethernet6/1
VLAN Configuration
VLAN Number
VLAN Name
Add
Remove
VLAN No
VLAN Name
1 default
2 cseise
1002 fddi-default
1003 token-ring-default
1004 fddinet-default
1005 trnet-default
Equivalent IOS Commands
Switch(config-if)#exit
Switch(config)#interface Ethernet6/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#
Fa0
PC-PT
192.168.2.3


Switch1
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
SWITCH
VLAN Database
INTERFACE
FastEthernet0/1
FastEthernet1/1
FastEthernet2/1
FastEthernet3/1
FastEthernet4/1
FastEthernet5/1
Ethernet6/1
FastEthernet2/1
Port Status
On
Bandwidth
Auto
10 Mbps
100 Mbps
Duplex
Auto
Full Duplex
Half Duplex
Access
VLAN
2
Tx Ring Limit
10
Equivalent IOS Commands
Switch(config)#interface Ethernet6/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#interface Ethernet6/1
Switch(config-if)#
Switch(config-if)#exit
Switch(config)#
Switch(config)#interface FastEthernet2/1
Switch(config-if)#
Fa0
PC-PT
192.168.2.3


192.168.1.2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 192.168.2.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.2.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms
PC>ping 192.168.2.3
Pinging 192.168.2.3 with 32 bytes of data:
Reply from 192.168.2.3: bytes=32 time=4ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Ping statistics for 192.168.2.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 4ms, Average = 2ms
PC>
Fa0
PC-PT
192.168.2.3

Experiment 9: Address Resolution Protocol (ARP)

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

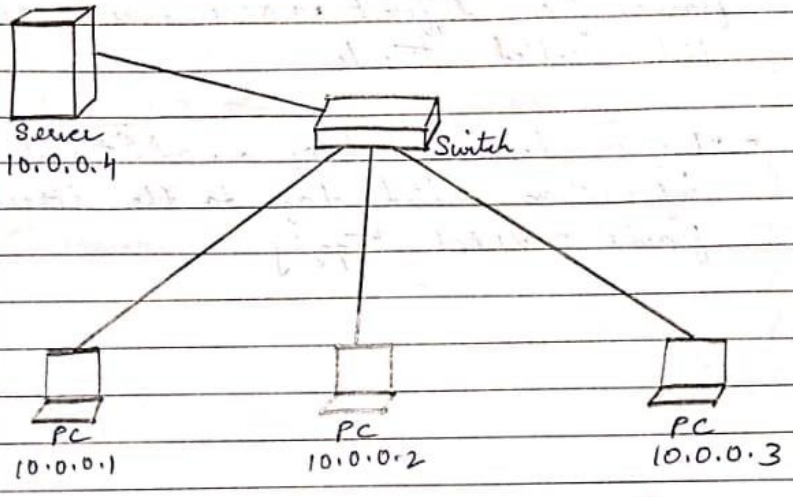
Observation writeup:

18/12/24 LAB No 10 40

ARP

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

TOPOLOGY:



Procedure:

1.] Create the topology as shown above

2.] Configure the PC's and the Server

3.] Click on Inspect mode (Q), then click on the end devices and open ARP table

4.] Send a data packet from any end device say server to other end device say 10.0.0.3 PC.

5.] Open Simulation mode to capture each step of data transfer

Bafna Gold
Date: Page:

Observations:

1.] The ARP tables of all end devices are initially empty

2.] When the data packet from Server arrives at the switch, since the source MAC address is unknown, it sends a broadcast message to all devices

3.] The device with the IP address present in the destination address of the data packet responds to the message

4.] The server and the PC update their ARP tables matching IP addresses to MAC address

5.] Over time, the ARP tables grows as data packets are sent

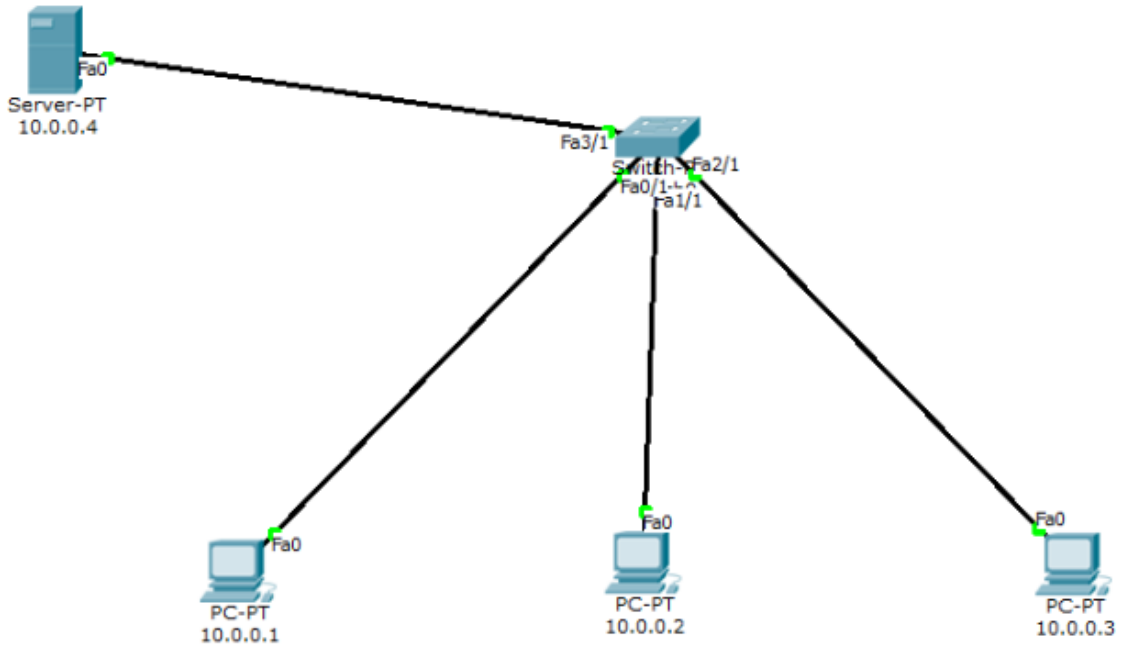
6.] The MAC table of the switch which was initially empty updates its MAC table gradually too

ARP Table for 10.0.0.4:

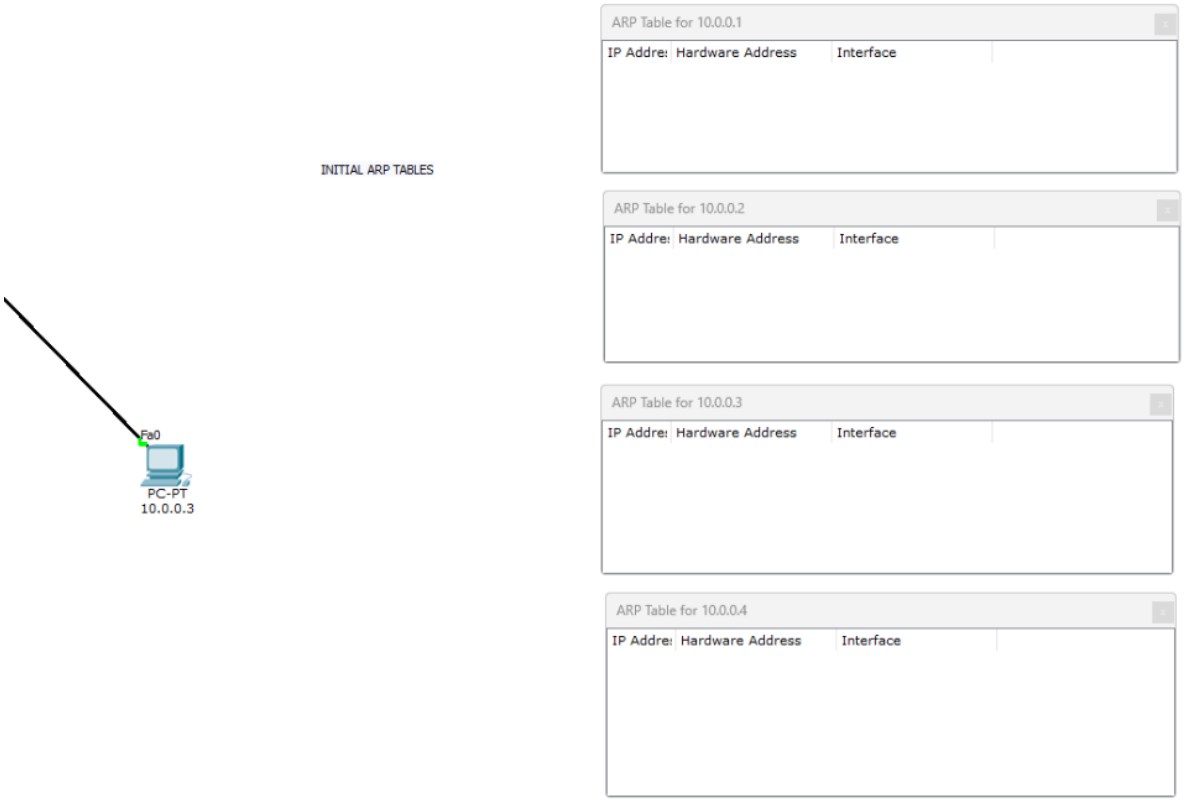
IP Address	Hardware Address	Interface
10.0.0.3	0001.C726.47E5	FastEthernet0

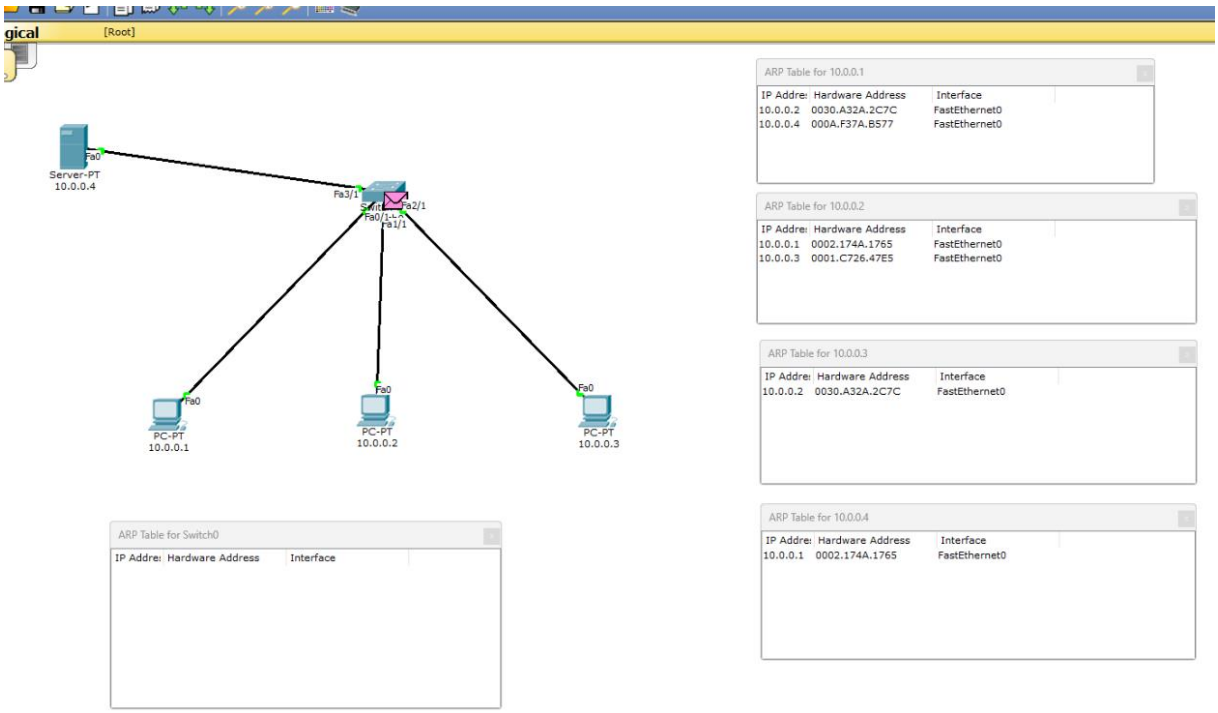
7.] Similarly other ARP tables are updated

Screenshot of the topology:



Screenshot(s) of the output:





Switch0

Physical Config CLI

IOS Command Line Interface

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state t
o up

%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state t
o up


Switch>show mac address -table
                        ^
% Invalid input detected at '^' marker.

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

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

   1    0001.c726.47e5    DYNAMIC     Fa2/1
   1    0002.174a.1765    DYNAMIC     Fa0/1
   1    000a.f37a.b577    DYNAMIC     Fa3/1
   1    0030.a32a.2c7c    DYNAMIC     Fa1/1
Switch>
```

Copy Paste

Experiment 10: Domain Name System (DNS)

Question: Configure Web Server, DNS within a LAN.

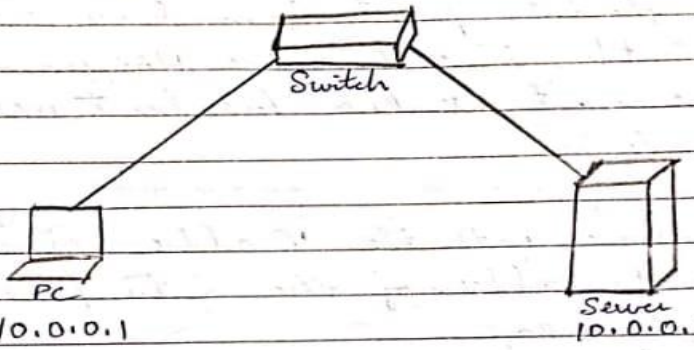
Observation writeup:

18/12/24 LAB NO 11 42

DNS

AIM: Configure Web Server, DNS within a LAN

TOPOLOGY:



Procedure:

1.] Set up the LAN as per the topology mentioned above & configure the devices.

2.] Go to Server → Services → DNS:
Name: bmsce [Domain name]
Address: 10.0.0.2
Add the mapping of domain name to address

3.] Go to PC → Config → Global → Setting →
DNS Server: 10.0.0.2
[The Server that provides the DNS mapping]

4.] Go to PC → Desktop → Web Browser
Type the URL: http://bmsce

Bafna Gold
Date: Page:

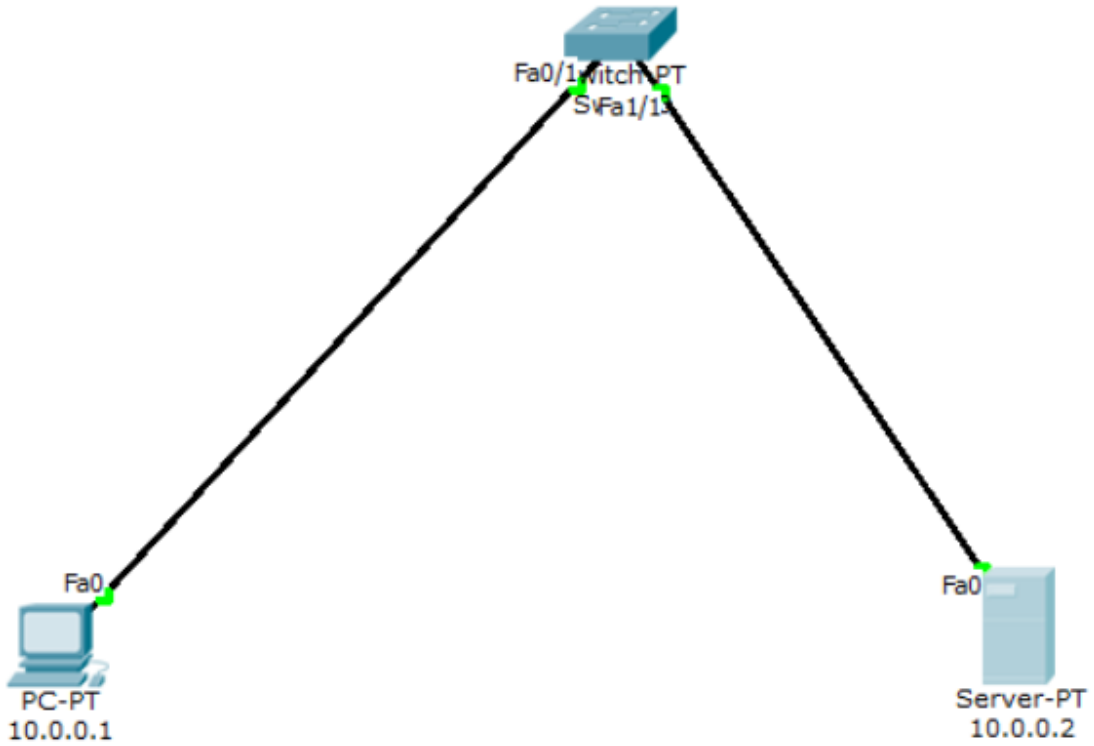
Observation :

1.] The Webpages hosted by the Server were visible on the browser

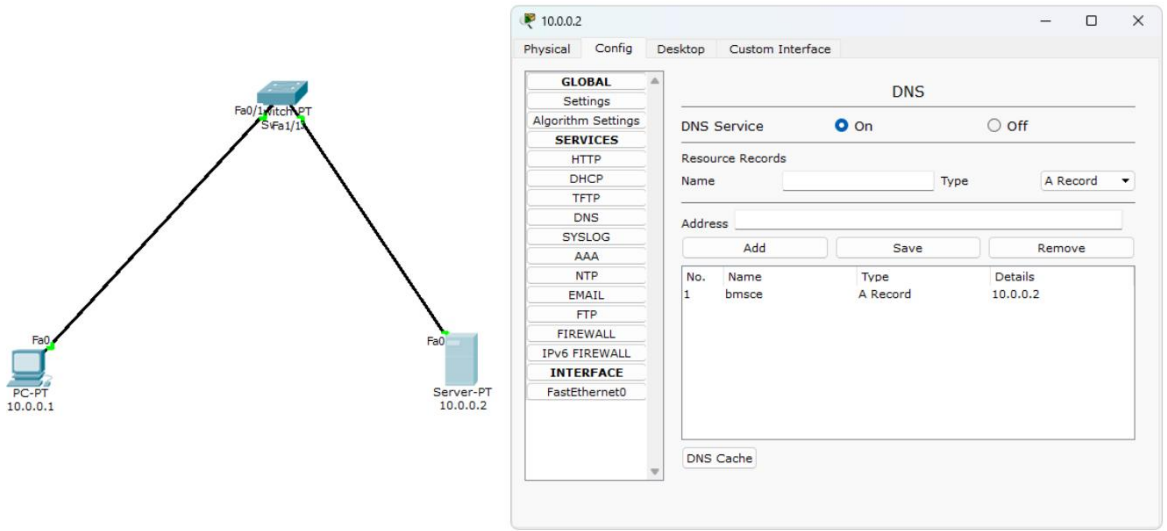
2.] The DNS was successful in mapping the domain name to the IP address

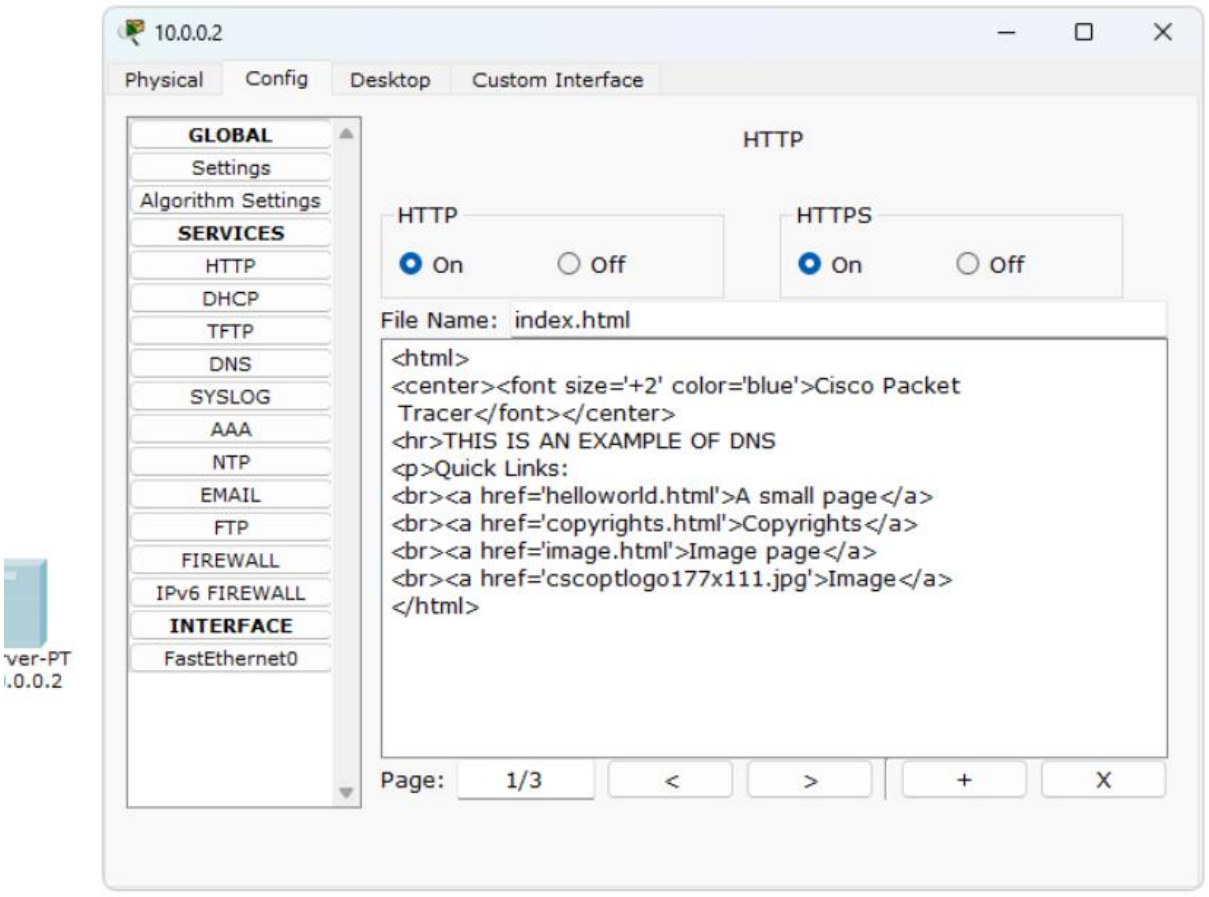
3.] DNS Server is a server that contains a Domain name : IP address mapping to which the end devices send requests to map the Name to IP address

Screenshot of the topology:



Screenshot(s) of the output:



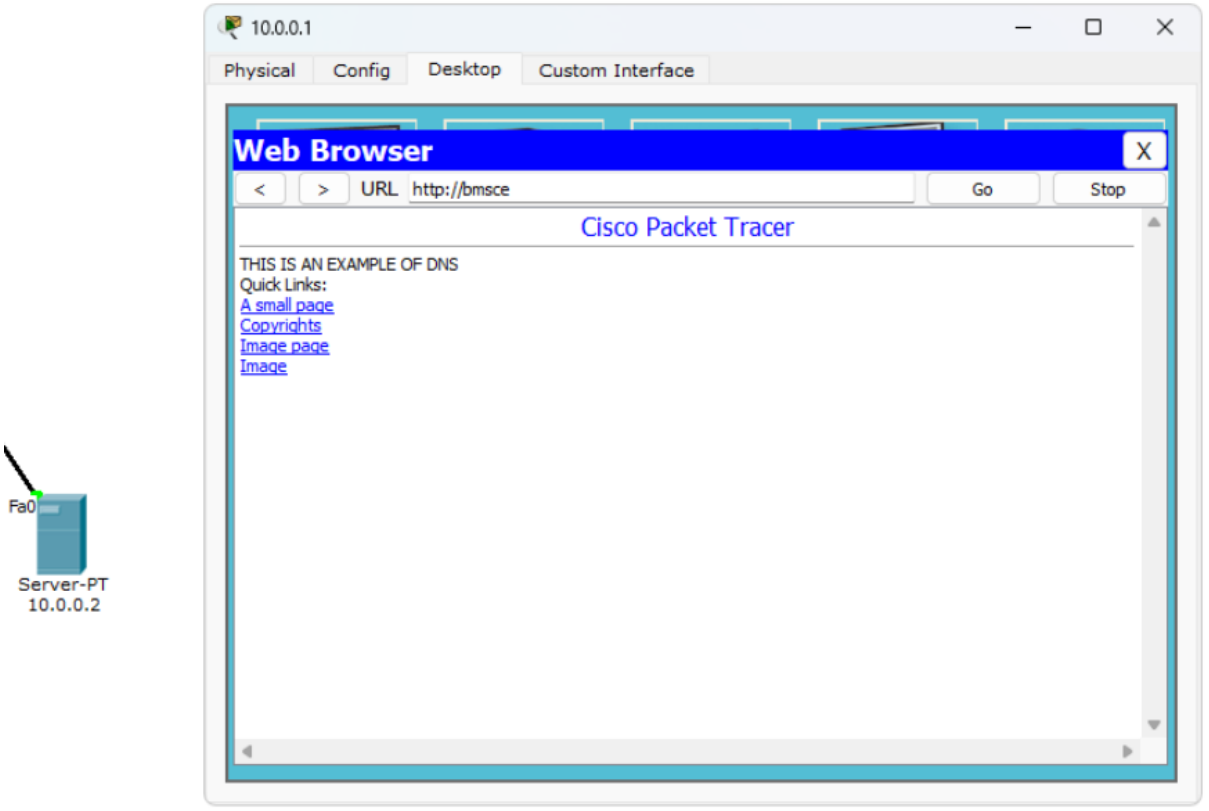

10.0.0.2
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
SERVICES
HTTP
DHCP
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
FIREWALL
IPv6 FIREWALL
INTERFACE
FastEthernet0
HTTP
HTTP
On
Off
HTTPS
On
Off
File Name: index.html
<html>
Cisco Packet
Tracer
<hr>THIS IS AN EXAMPLE OF DNS
<p>Quick Links:

A small page

Copyrights

Image page

Image
</html>
Page: 1/3
<
>
+
X


10.0.0.1
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL http://bmsce
Go
Stop
Cisco Packet Tracer
THIS IS AN EXAMPLE OF DNS
Quick Links:
A small page
Copyrights
Image page
Image
Fa0
Server-PT
10.0.0.2

Experiment 11: TELNET

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

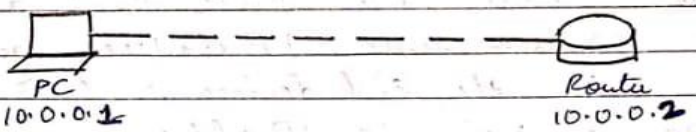
Observation writeup:

18/12/24 LAB No 12 44

TELNET

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

TOPOLOGY:



Procedure:

1.] Create the topology as given above and configure the devices

2.] Commands in Router:

```
Router> enable
Router # config terminal
Router(config)# hostname R1
R1(config)# enable secret 1234
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 3
R1(config-line)# login
% Login disabled on line 194, until 'password' is set
R1(config-line)# password 4321
R1(config-line)# exit
```

enable password

user access verification password

Bafna Gold
Date: Page:

R1(config)# exit

R1 # wr
Building configuration...
[OK]

Note: vty 0 3 : First four virtual terminal lines for Telnet access

3.] In PC: Command Prompt:
- First try pinging to see if devices are connected

PC> telnet 10.0.0.2
Trying 10.0.0.2 ... Open

User Access Verification

Password: 4321
Password: 4321
R1 > enable
Password: 1234
R1 # show ip route

C 10.0.0.0/8 is directly connected, FastEthernet0/0
R1 #

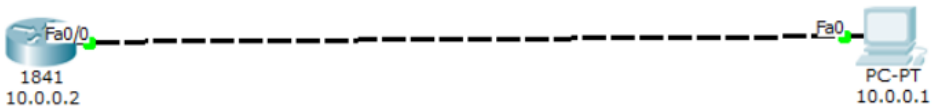
Observations:

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

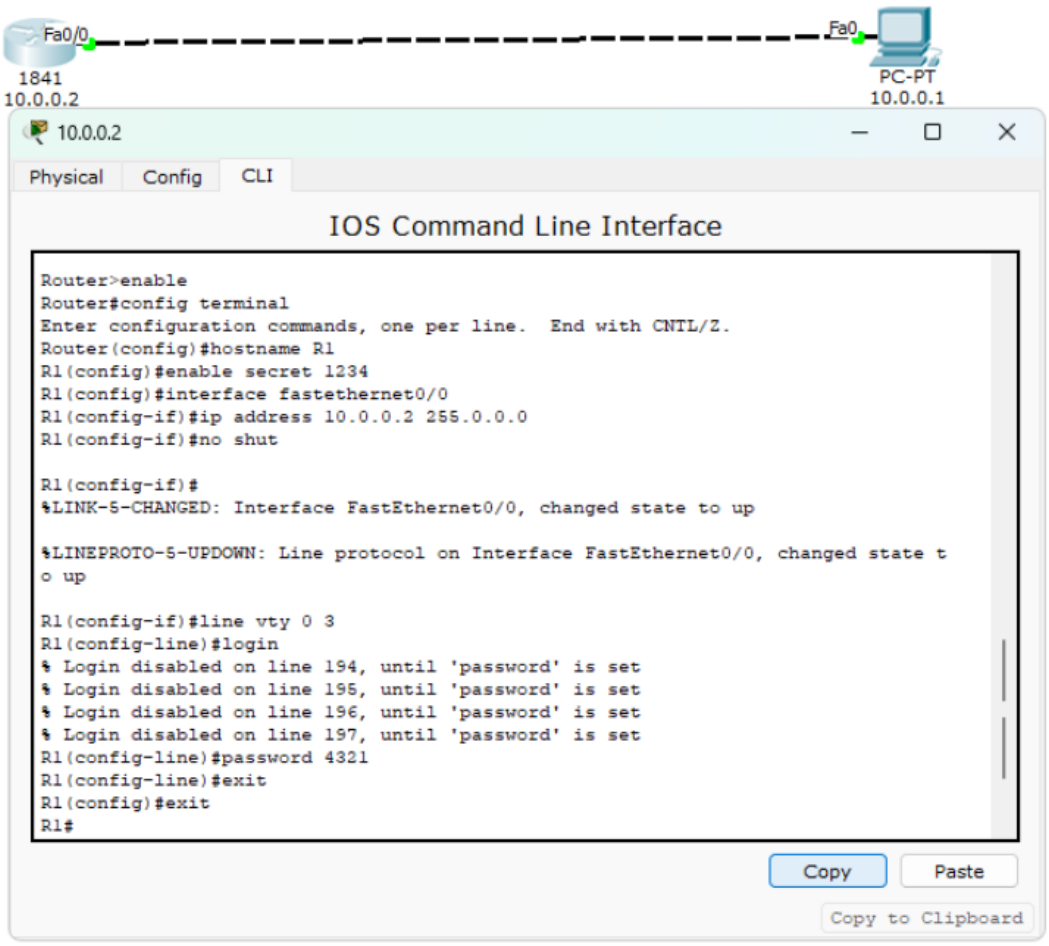
2.] Telnet allows users to establish a remote session with another device like router, over a TCP/IP network

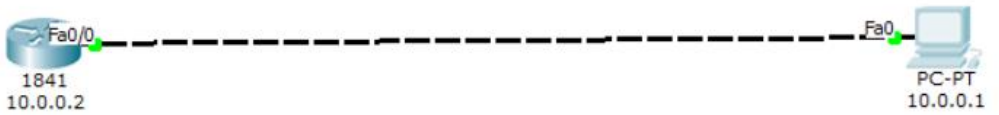
3.] Using Telnet, we can access and control the remote device's CLI as if you were physically connected to it

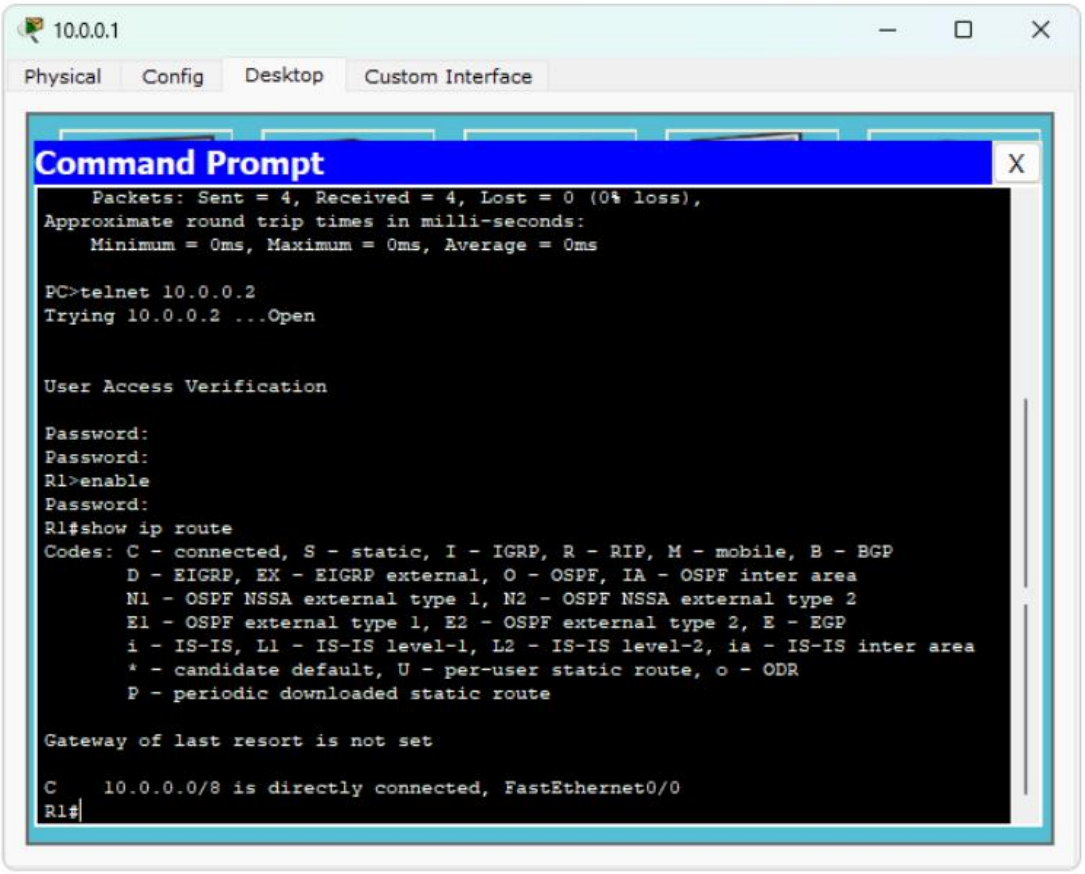
Screenshot of the topology:



Screenshot(s) of the output:




Fa0/0
1841
10.0.0.2
Fa0
PC-PT
10.0.0.1


10.0.0.1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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:
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#

Experiment 12: Wireless Lan (WLAN)

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

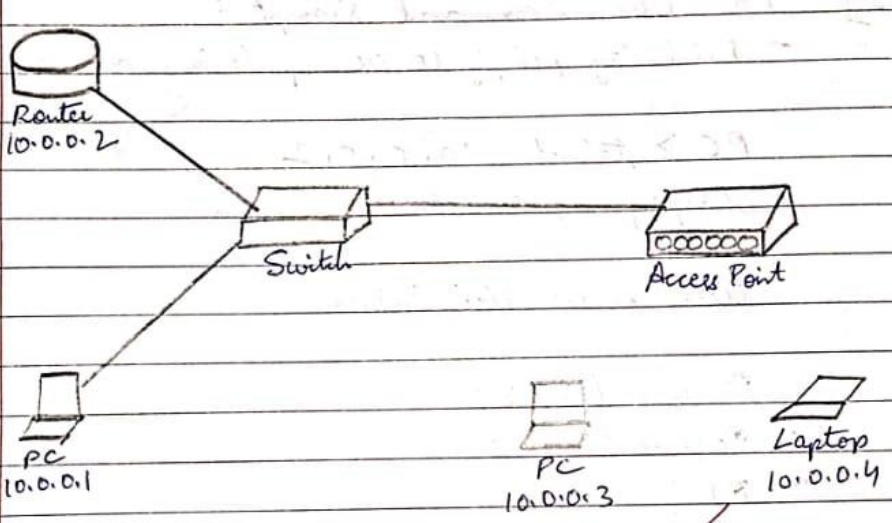
Observation writeup:

18/12/24 LAB NO 13 46

WLAN

AIM: To construct a Wireless LAN and make the nodes communicate wirelessly

INITIAL TOPOLOGY:



Procedure:

1.] Create the topology as given above and configure the devices

2.] Configure Access Point:
click AccessPoint → Config → Port1:
SSID : bmsce
Select ⊙ WEP
Set Key : 1234567890

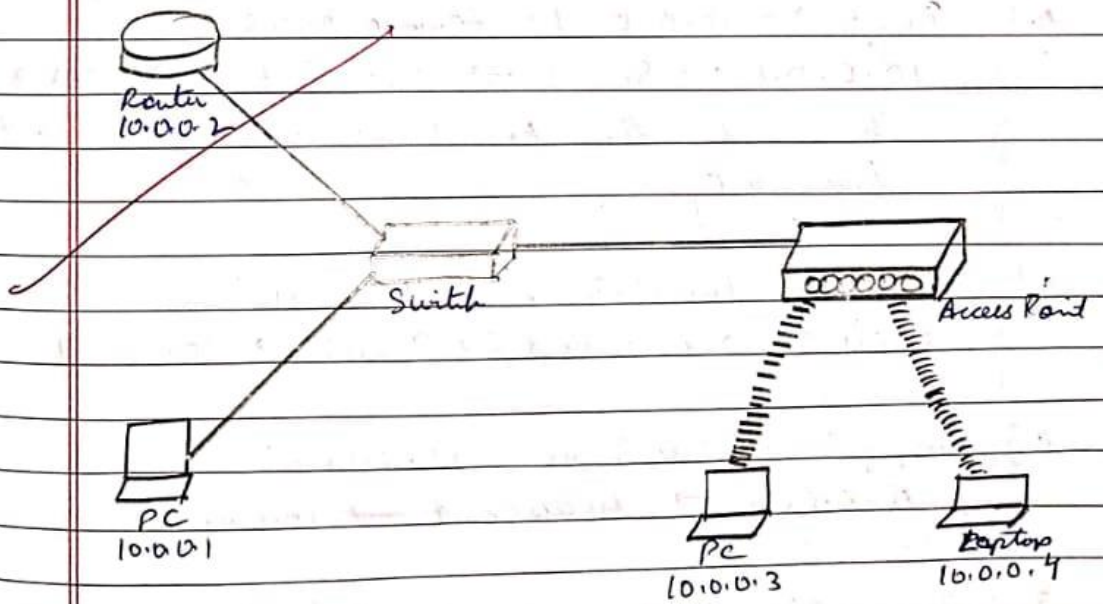
Bafna Gold
Date: Page:

3.] Configure PC & Laptop with Wireless standards:
- Switch off device
- Drag the existing PT-HOST-NM-1AM to the component listed in the LHS of Physical
- Drag WMP300N wireless interface to the empty port
- Switch on the device

4.] In the config tab, a new wireless interface was added

5.] Configure the device by entering SSID, WEP, WEP Key, IP address and Gateway

Topology after Wireless Configuration:



6.] Ping from every device to every other device to check for connection

Observations:

1.] We were able to ping from every device to every other device

2.] Access Point:
Creates bridge between wired and wireless devices
- SSID Broadcasting: announces the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2

3.] WMP300N wireless interface:
- Wireless network adapter that enables devices to communicate with access Point using wireless signals

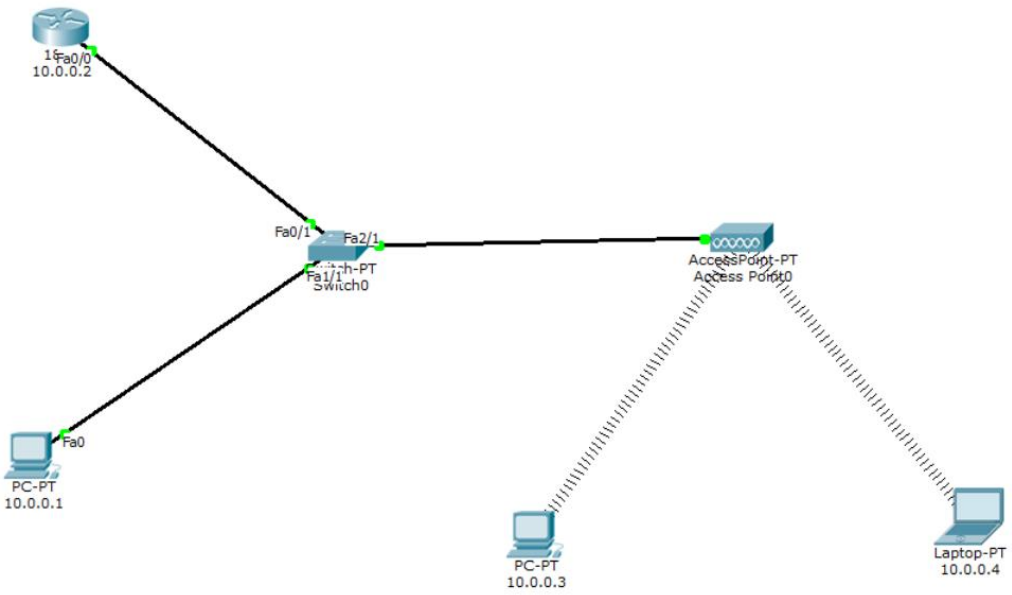
4.] Pinging: 10.0.0.1 to 10.0.0.3:
10.0.0.1 → Switch → Access Point → 10.0.0.3
- This is after the ARP tables are updated after broadcasting

5.] Pinging: 10.0.0.3 to 10.0.0.1:
10.0.0.3 → Access Point → Switch → 10.0.0.1

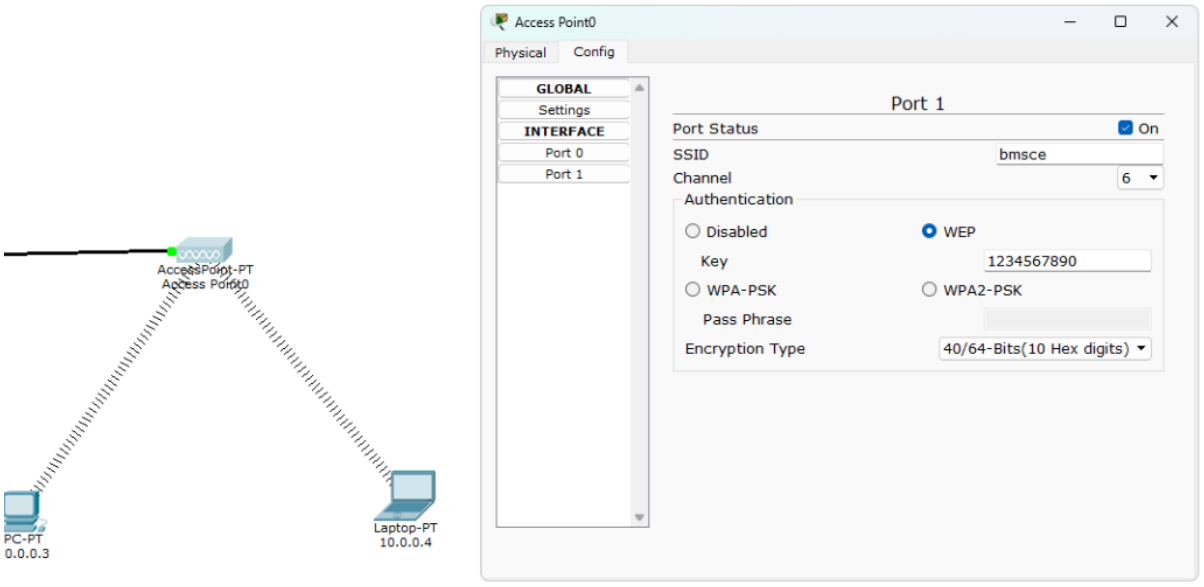
6.] Pinging: 10.0.0.3 to 10.0.0.4:
10.0.0.3 → Access Point → 10.0.0.4

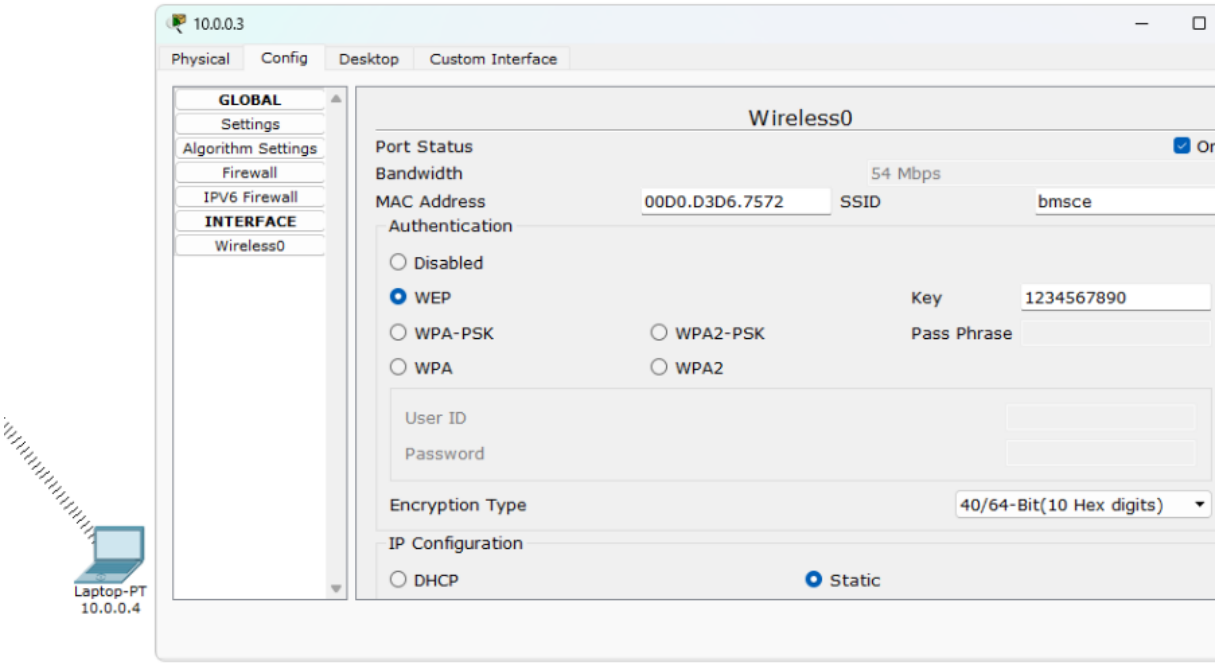
7.] Every device is now connected to every other device in the WLAN

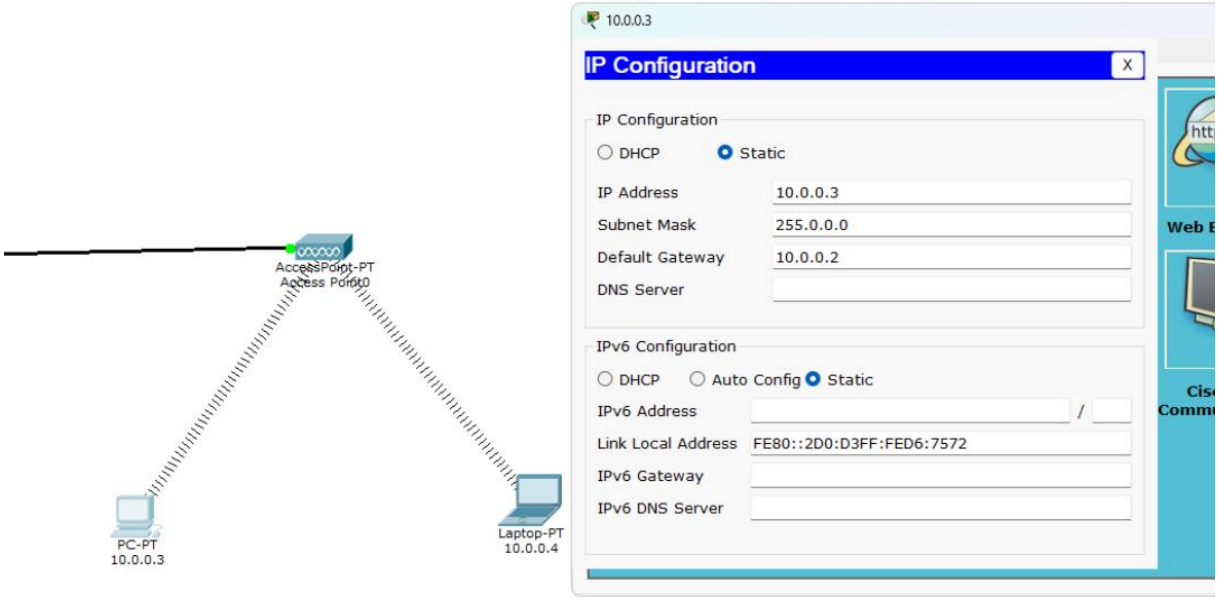
Screenshot of the topology:

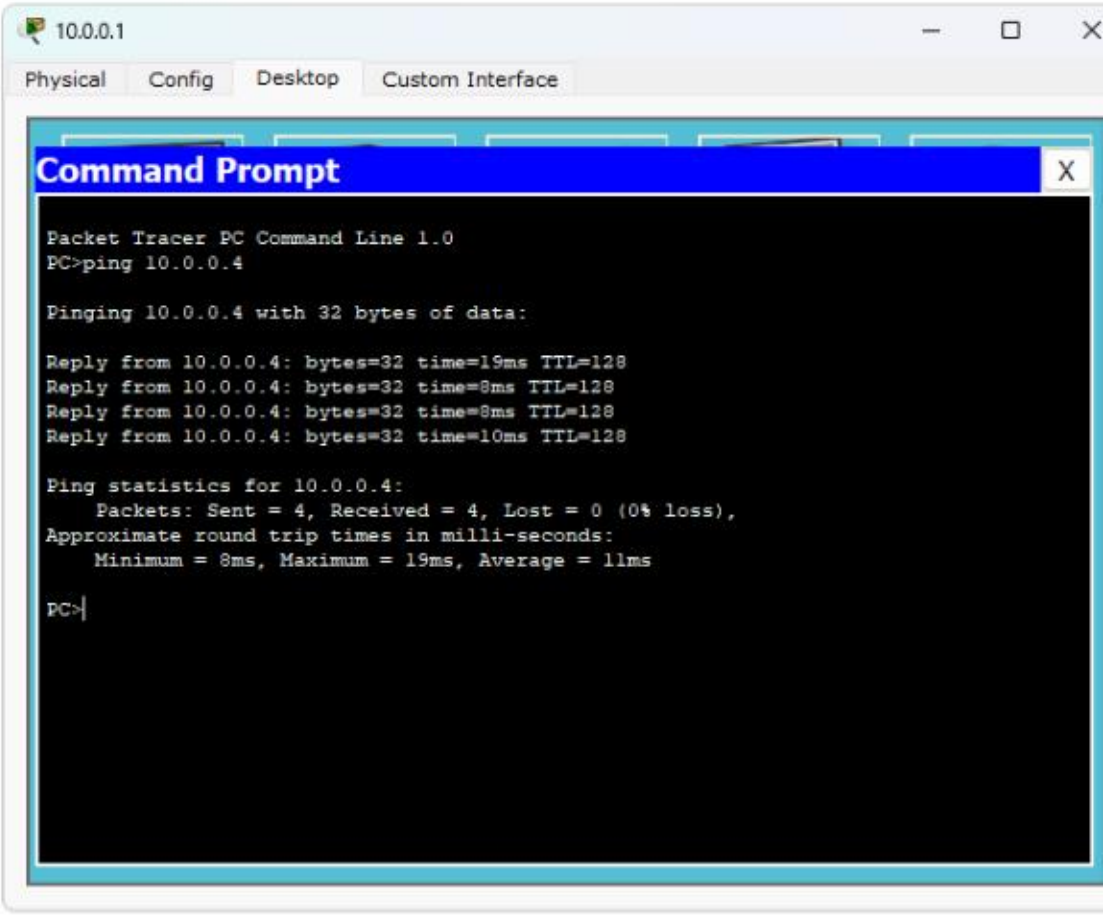


Screenshot(s) of the output:




10.0.0.3
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
Firewall
IPV6 Firewall
INTERFACE
Wireless0
Wireless0
Port Status
Bandwidth
54 Mbps
MAC Address
00D0.D3D6.7572
SSID
bmsce
Authentication
Disabled
WEP
Key
1234567890
WPA-PSK
WPA2-PSK
Pass Phrase
WPA
WPA2
User ID
Password
Encryption Type
40/64-Bit(10 Hex digits)
IP Configuration
DHCP
Static
Laptop-PT
10.0.0.4


10.0.0.3
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.2
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2D0:D3FF:FED6:7572
IPv6 Gateway
IPv6 DNS Server
AccessPoint-PT
Access Point0
PC-PT
10.0.0.3
Laptop-PT
10.0.0.4


10.0.0.1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
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=8ms TTL=128
Reply from 10.0.0.4: bytes=32 time=8ms TTL=128
Reply from 10.0.0.4: bytes=32 time=10ms 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 = 8ms, Maximum = 19ms, Average = 11ms
PC>

Cycle-2

Program 1: CRC-CCITT

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

Program:

49

Bafna Gold

Date: Page:

25/10/24

CYCLE-2

PROGRAMS

PROGRAM 1: CRC

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

Python code:

```
def crc_ccitt(data):

    polynomial = 0x1021
    initial_crc = 0xFFFF

    crc = initial_crc

    for byte in data.encode():
        crc ^= (byte << 8)
        for _ in range(8):
            if crc & 0x8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1
            crc &= 0xFFFF
    return crc


data = input('Enter data to calculate CRC-CCITT:');
print(f'checksum: {crc_ccitt(data):04X}')
```

- if MSB is 1, simulate division else move to next
- Masking ensures CRC remains 16 bit

OUTPUT:

Enter data to calculate CRC-CCITT: 1234

CRC-CCITT checksum: 5349

Program (soft copy):

```
def crc_ccitt(data: str, polynomial: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    """
    Calculate the CRC-CCITT checksum for the given data.

    :param data: Input data as a string
    :param polynomial: CRC polynomial (default is 0x1021)
    :param init_crc: Initial CRC value (default is 0xFFFF)
    :return: The computed CRC value
    """
    crc = init_crc
    print(data.encode())
    for byte in data.encode():  # Convert string to bytes
        crc ^= (byte << 8)
        for _ in range(8):  # Process 8 bits
            if crc & 0x8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1
            crc &= 0xFFFF  # Keep CRC to 16 bits
    return crc

# Example usage
def main():
    data = input("Enter the data to calculate CRC-CCITT: ")
    print(f"CRC-CCITT checksum: {crc_ccitt(data):04X}")

if __name__ == "__main__":
    main()
```

[1]:
```python
def crc_ccitt(data: str, polynomial: int = 0x1021, init_crc: int = 0xFFFF) -> int:
    """
    Calculate the CRC-CCITT checksum for the given data.

    :param data: Input data as a string
    :param polynomial: CRC polynomial (default is 0x1021)
    :param init_crc: Initial CRC value (default is 0xFFFF)
    :return: The computed CRC value
    """
    crc = init_crc
    print(data.encode())
    for byte in data.encode():  # Convert string to bytes
        crc ^= (byte << 8)
        for _ in range(8):  # Process 8 bits
            if crc & 0x8000:
                crc = (crc << 1) ^ polynomial
            else:
                crc <<= 1
            crc &= 0xFFFF  # Keep CRC to 16 bits
    return crc

# Example usage
def main():
    data = input("Enter the data to calculate CRC-CCITT: ")
    print(f"CRC-CCITT checksum: {crc_ccitt(data):04X}")

if __name__ == "__main__":
    main()
```

Output:

```
Enter the data to calculate CRC-CCITT:  1234
b'1234'
CRC-CCITT checksum: 5349
```

Program 2: LEAKY BUCKET

Question: Write a program for congestion control using Leaky bucket algorithm.

Program:

50

25/12/24

PROGRAM-2

2.] Write a program for Congestion Control using Leaky Bucket algorithm

- Imagine a bucket with a small hole at the bottom
- Water(packets) can be added to the bucket, but it can only leave through the hole at constant rate (output rate)
- If the bucket is full and more water is added, the excess water overflows (packets are dropped)

Python code:

```
import time
import random

NUM-Packets = 5

def leaky-bucket(output-rate, bucket-size):

    packet-sizes = [random.randint(1,100) for
                    - in range(NUM-Packets)]

    print('Incoming packets:')
    for i, packet in enumerate(packet-sizes):
        print(f"Packet [{i}] : {packet} bytes")

    remaining-bytes = 0
```

Bafna Gold
Date: Page:

```
for i, packet in enumerate(packet_sizes):
    print(f"Processing Packet [{i}] of size {packet} bytes...")

    if packet > bucket_size:
        print(f" Packet size {packet} exceed bucket capacity {bucket_size}
              - PACKET Rejected")
        continue

    remaining_bytes += packet
    print(f" Packet accepted. Bytes remaining in bucket: {remaining_bytes}")

    while remaining_bytes > 0:
        time.sleep(1)
        if remaining_bytes > output_rate:
            transmitted = output_rate
            remaining_bytes -= output_rate
        else:
            transmitted = remaining_bytes
            remaining_bytes = 0

        print(f" Transmitted {transmitted} bytes.
              Remaining {remaining_bytes} bytes")

print("Transmission complete")

output_rate = int(input("Enter Output rate:"))
bucket_size = int(input("Enter Bucket size:"))

leaky_bucket(output_rate, bucket_size)
```

Output :

Enter Output rate : 30
Enter bucket size : 70

Incoming Packets :
Packet [0] : 82 bytes
Packet [1] : 39 bytes
Packet [2] : 43 bytes
Packet [3] : 74 bytes
Packet [4] : 67 bytes

Processing Packet [0] of size 82 bytes :
Packet size 82 exceed bucket size 70 : Packet rejected

Processing Packet [1] of size 39 bytes..
Packet accepted. Bytes remaining in bucket : 39
Transmitted 30 bytes. Remaining : 9 bytes
Transmitted 9 bytes. Remaing : 0 bytes

Processing Packet [2] of size 43 bytes
Packet accepted. Bytes remaing in bucket : 43
Transmitted 30 bytes Remaing : 13 bytes
Transmitted 13 byte Remaing : 0 bytes

Processing Packet [3] of size 74 bytes
Packet size 74 exceed bucket size 70 : Packet rejected

Processing Packet [4] of size 67 bytes
Packet accepted. Bytes remg in bucket : 67
Transmitted 30 bytes Remaig : 37 bytes
Transited 30 bytes Remaig : 7 bytes
Transmitted 7 bytes Remaig : 0 bytes
Transmission complete

Program (soft copy):

```
import time
import random

# Constants
NUM_PACKETS = 5  # Number of incoming packets

# Function to simulate the leaky bucket algorithm
def leaky_bucket(output_rate, bucket_size):
    # Generate random packet sizes
    packet_sizes = [random.randint(1, 100) for _ in range(NUM_PACKETS)]

    print("Incoming Packets:")
    for i, packet in enumerate(packet_sizes):
        print(f"Packet[{i}]: {packet} bytes")

    remaining_bytes = 0  # Bytes left in the bucket

    for i, packet in enumerate(packet_sizes):
        print(f"\nProcessing Packet[{i}] of size {packet} bytes...")

        # Check if the packet size is greater than the bucket capacity
        if packet > bucket_size:
            print(f"  Incoming packet size ({packet} bytes) exceeds bucket capacity
({bucket_size} bytes) - PACKET REJECTED")
            continue

        # Check if the bucket can hold the incoming packet
        #if remaining_bytes + packet > bucket_size:
            #print(f"  Bucket capacity exceeded - PACKET REJECTED")
            #continue

        # Add packet to the bucket
        remaining_bytes += packet
        print(f"  Packet accepted. Bytes remaining in bucket: {remaining_bytes}")
```

```
        # Transmit packets from the bucket at the defined output rate
        while remaining_bytes > 0:
            time.sleep(1)  # Simulate transmission time
            if remaining_bytes > output_rate:
                transmitted = output_rate
                remaining_bytes -= output_rate
            else:
                transmitted = remaining_bytes
                remaining_bytes = 0
            print(f"  Transmitted {transmitted} bytes. Remaining: {remaining_bytes} bytes")

    print("\nTransmission complete.")

# Input
output_rate = int(input("Enter the Output Rate (bytes/sec): "))
bucket_size = int(input("Enter the Bucket Size (bytes): "))

# Run the leaky bucket simulation
leaky_bucket(output_rate, bucket_size)
```

```python
import time
import random

# Constants
NUM_PACKETS = 5  # Number of incoming packets

# Function to simulate the leaky bucket algorithm
def leaky_bucket(output_rate, bucket_size):
    # Generate random packet sizes
    packet_sizes = [random.randint(1, 100) for _ in range(NUM_PACKETS)]

    print("Incoming Packets:")
    for i, packet in enumerate(packet_sizes):
        print(f"Packet[{i}]: {packet} bytes")

    remaining_bytes = 0  # Bytes left in the bucket

    for i, packet in enumerate(packet_sizes):
        print(f"\nProcessing Packet[{i}] of size {packet} bytes...")

        # Check if the packet size is greater than the bucket capacity
        if packet > bucket_size:
            print(f"  Incoming packet size ({packet} bytes) exceeds bucket capacity ({bucket_size} bytes) - PACKET REJECTED")
            continue

        # Check if the bucket can hold the incoming packet
        #if remaining_bytes + packet > bucket_size:
            #print(f"  Bucket capacity exceeded - PACKET REJECTED")
            #continue

        # Add packet to the bucket
        remaining_bytes += packet
        print(f"  Packet accepted. Bytes remaining in bucket: {remaining_bytes}")

        # Transmit packets from the bucket at the defined output rate
        while remaining_bytes > 0:
            time.sleep(1)  # Simulate transmission time
            if remaining_bytes > output_rate:
                transmitted = output_rate
                remaining_bytes -= output_rate
            else:
                transmitted = remaining_bytes
                remaining_bytes = 0
            print(f"  Transmitted {transmitted} bytes. Remaining: {remaining_bytes} bytes")

    print("\nTransmission complete.")

# Input
output_rate = int(input("Enter the Output Rate (bytes/sec): "))
bucket_size = int(input("Enter the Bucket Size (bytes): "))

# Run the leaky bucket simulation
leaky_bucket(output_rate, bucket_size)
```

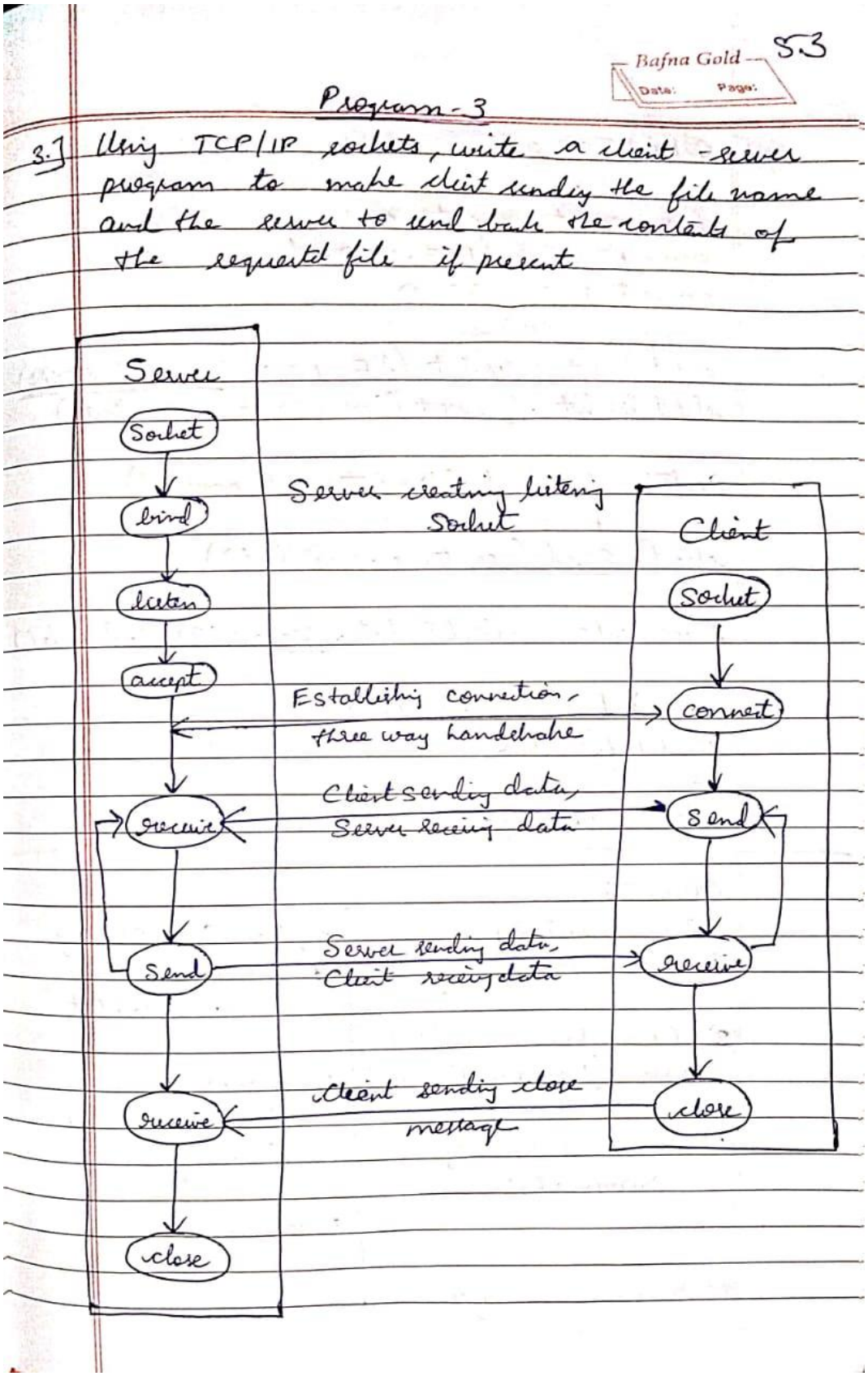
Output:

```
Enter the Output Rate (bytes/sec):  30
Enter the Bucket Size (bytes):  70
Incoming Packets:
Packet[0]: 81 bytes
Packet[1]: 37 bytes
Packet[2]: 64 bytes
Packet[3]: 94 bytes
Packet[4]: 22 bytes

Processing Packet[0] of size 81 bytes...
  Incoming packet size (81 bytes) exceeds bucket capacity (70 bytes) - PACKET REJECTED

Processing Packet[1] of size 37 bytes...
  Packet accepted. Bytes remaining in bucket: 37
  Transmitted 30 bytes. Remaining: 7 bytes
  Transmitted 7 bytes. Remaining: 0 bytes

Processing Packet[2] of size 64 bytes...
  Packet accepted. Bytes remaining in bucket: 64
  Transmitted 30 bytes. Remaining: 34 bytes
  Transmitted 30 bytes. Remaining: 4 bytes
  Transmitted 4 bytes. Remaining: 0 bytes

Processing Packet[3] of size 94 bytes...
  Incoming packet size (94 bytes) exceeds bucket capacity (70 bytes) - PACKET REJECTED

Processing Packet[4] of size 22 bytes...
  Packet accepted. Bytes remaining in bucket: 22
  Transmitted 22 bytes. Remaining: 0 bytes

Transmission complete.
```

Program 3: TCP/IP SOCKETS

Question: 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.

Program:



1) ClientTCP.py:

```
from socket import *
ServerName = '127.0.0.1'
serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((ServerName, ServerPort))

Sentence = input("\n Enter file name:")
clientSocket.send(sentence.encode())

filecontents = clientSocket.recv(1024).decode()

print('\n From Server: \n')
print(filecontents)

clientSocket.close()
```

OUTPUT:

python ServerTCP.py

① The server is ready to rec

② Connection established with ('127.0.0.1', 60472)

④ Sent contents of sample.txt

⑦ Server is ready to receive

python ClientTCP.py

③ Enter file name: sample.txt

⑤ From Server: This is a sample Test file

⑥ # done

Bafna Gold
Date: Page:

2.] ServerTCP.py

```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000

ServerSocket = socket (AF_INET, SOCK_STREAM)
ServerSocket.bind ((ServerName, ServerPort))

ServerSocket.listen (1)

while 1:
    print ('The server is ready to receive')
    connectionSocket, addr = ServerSocket.accept()
    print (f'Connection established with {addr}')

    sentence = connectionSocket.recv (1024).decode()

    file = open (sentence, 'r')

    l = file.read (1024)

    connectionSocket.send (l.encode())
    print (f'Sent contents of {sentence}')
    file.close()

    connectionSocket.close()
```

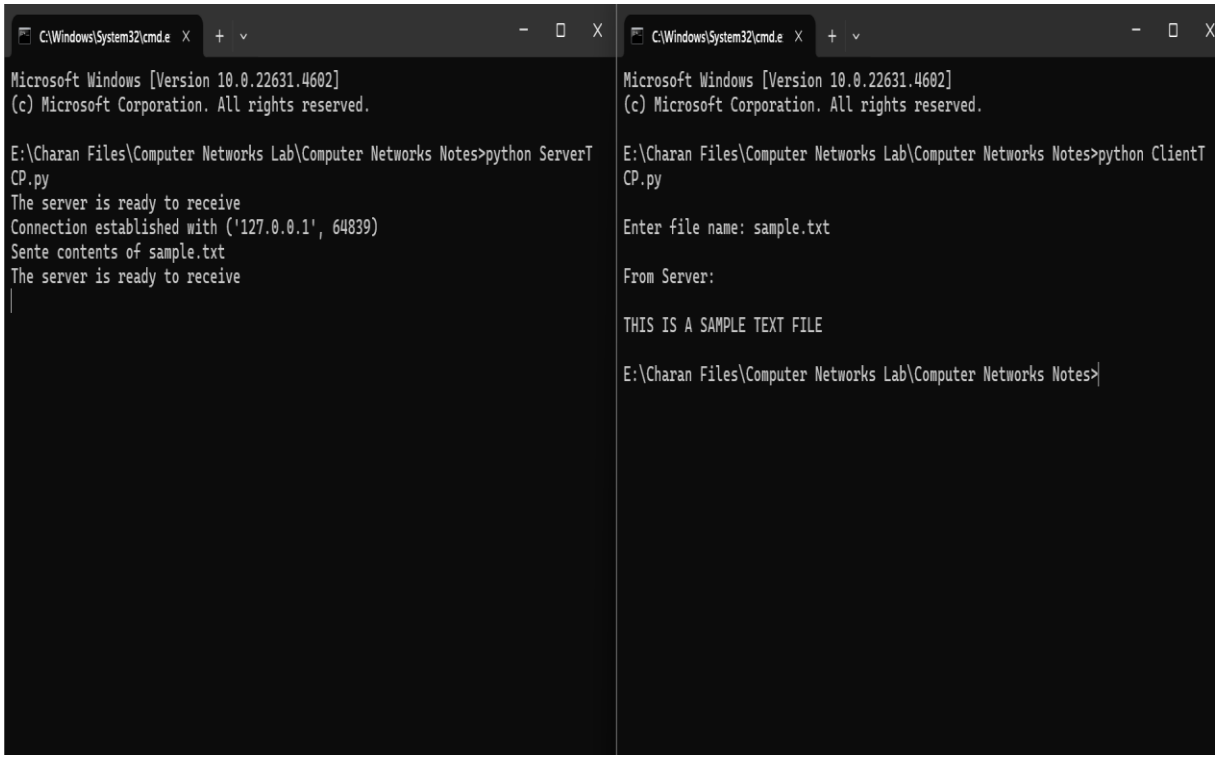
#1024 - no of bytes that the server will try to read at once from the socket

Program (soft copy):

```
#ClientTCP.py
from socket import *

serverName = '127.0.0.1'
serverPort = 12000

clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))

sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())

filecontents = clientSocket.recv(1024).decode()
print("\nFrom Server:\n")
print(filecontents)

clientSocket.close()



#ServerTCP.py
from socket import *

serverName = "127.0.0.1"
serverPort = 12000

serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)




while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
```

```
print(f"Connection established with {addr}")

sentence = connectionSocket.recv(1024).decode()

file = open(sentence, 'r')
l = file.read(1024)
connectionSocket.send(l.encode())
print(f"Sente contents of {sentence}")
file.close()

connectionSocket.close()
```

Output:



Program 4: UDP SOCKETS

Question: 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.

Program:

PROGRAM-4 56

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

1.] Client UDP.py:

```
from socket import *

ServerName = '127.0.0.1'
ServerPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input ('Enter file name')

clientSocket.sendto(bytes(sentence,"utf-8"),
                    (ServerName, ServerPort))

fileContents, ServerAddress = clientSocket.recvfrom
                              (2048)

print('Reply from Server:')

print (fileContents.decode('utf-8'))

for i in fileContents:
    print(str(i), end = '')

clientSocket.close()
```

Bafna Gold
Date: Page:

2.] Server UDP.py:

```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK_DGRAM)
ServerSocket.bind((ServerName '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(f'Sent contents of {sentence}')

    for i in sentence:
        print(str(i), end = '')

    file.close()
```

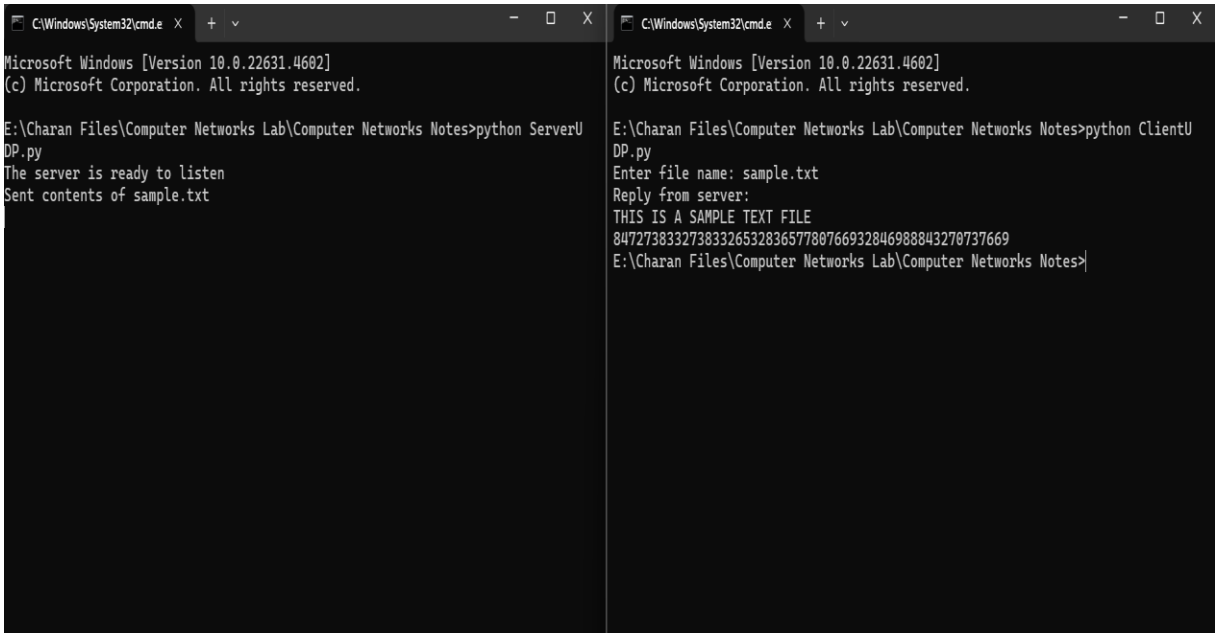
Observation:
- No connections is established

Program (soft copy):

```
#ClientUDP
from socket import *

serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input('Enter file name')

clientSocket.sendto(bytes(sentence, 'utf-8'), (serverName, serverPort))

filecontents, serverAddress = clientSocket.recvfrom(2048)

print("Reply from server:")
print(filecontents.decode('utf-8'))

for i in filecontents:
        print(str(i), end = '')
clientSocket.close()



#ServerUDP
from socket import *

serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', 12000))

print('The server is ready to listen')

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('Sent contents of ' + sentence)
for i in sentence:
        print(str(i), end = '')
file.close()
```

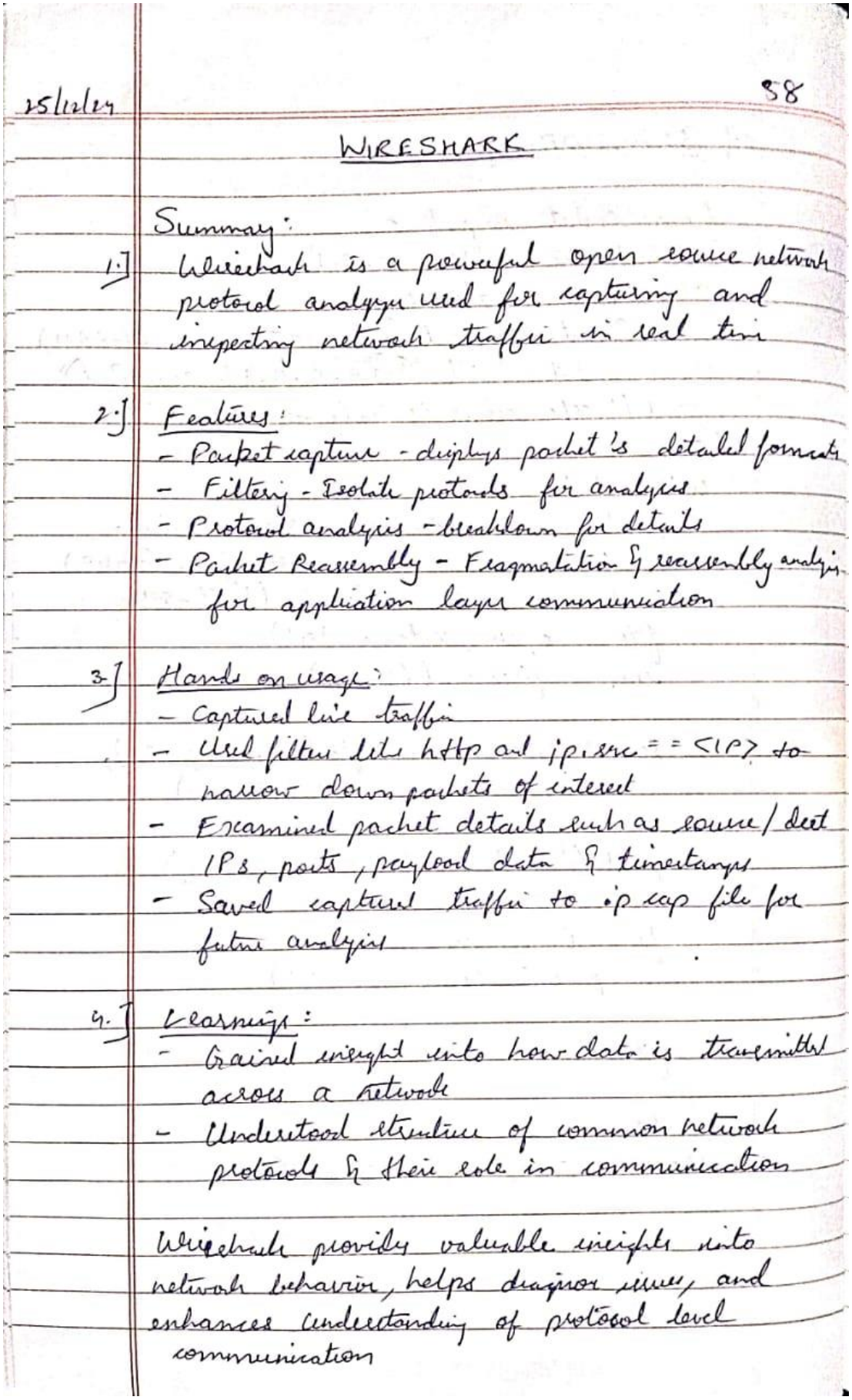
Output:



Wireshark

Question: Tool Exploration –Wireshark

Observation writeup:



25/12/24 58

WIRESHARK

Summary:

1.] Wireshark is a powerful open source network protocol analyzer used for capturing and inspecting network traffic in real time

2.] Features:
- Packet capture - displays packet's detailed format
- Filtering - Isolate protocols for analysis
- Protocol analysis - breakdown for details
- Packet Reassembly - Fragmentation & reassembly analysis for application layer communication

3.] Hands on usage:
- Captured live traffic
- Used filters like http and ip.src == <IP> to narrow down packets of interest
- Examined packet details such as source/dest IPs, ports, payload data & timestamps
- Saved captured traffic to .p cap file for future analysis

4.] Learnings:
- Gained insight into how data is transmitted across a network
- Understood structure of common network protocols & their role in communication

Wireshark provides valuable insights into network behavior, helps diagnose issues, and enhances understanding of protocol level communication