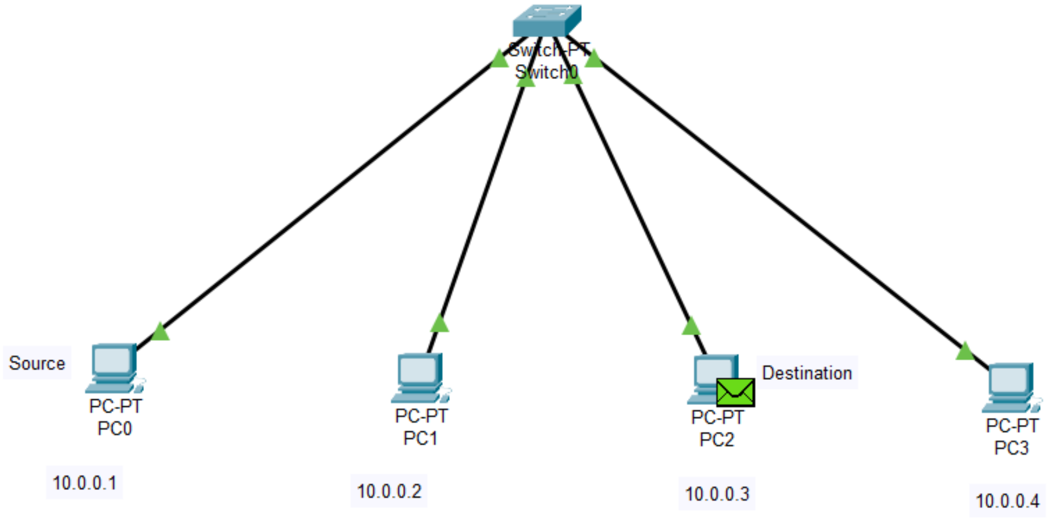
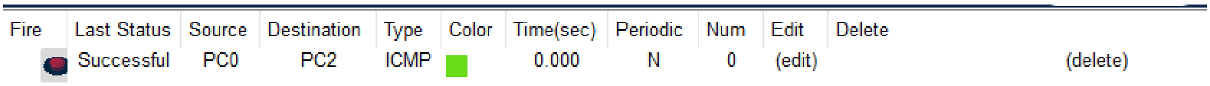
EXPERIMENT - 1

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

Switch:

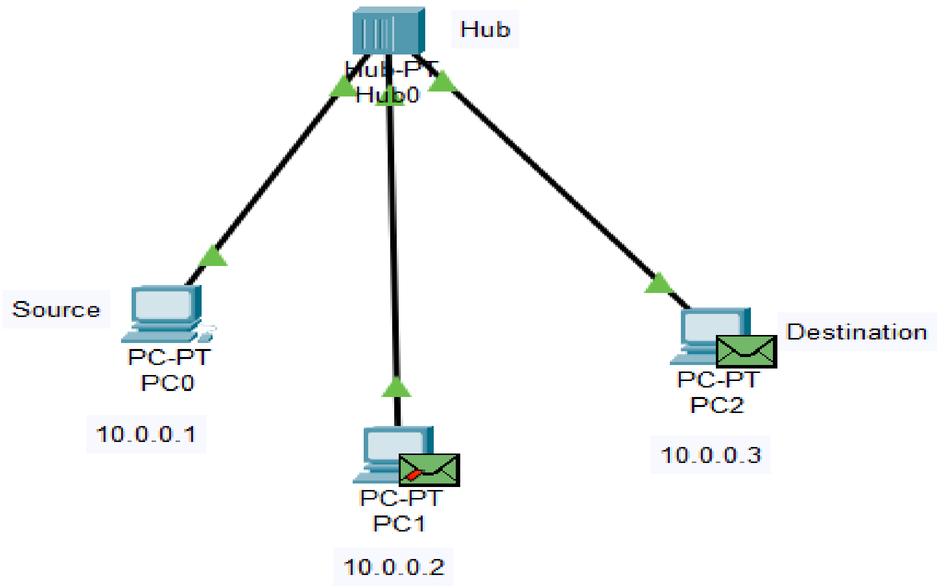


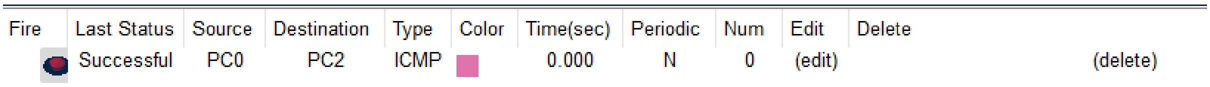
Connection: Copper Straight Through Wire



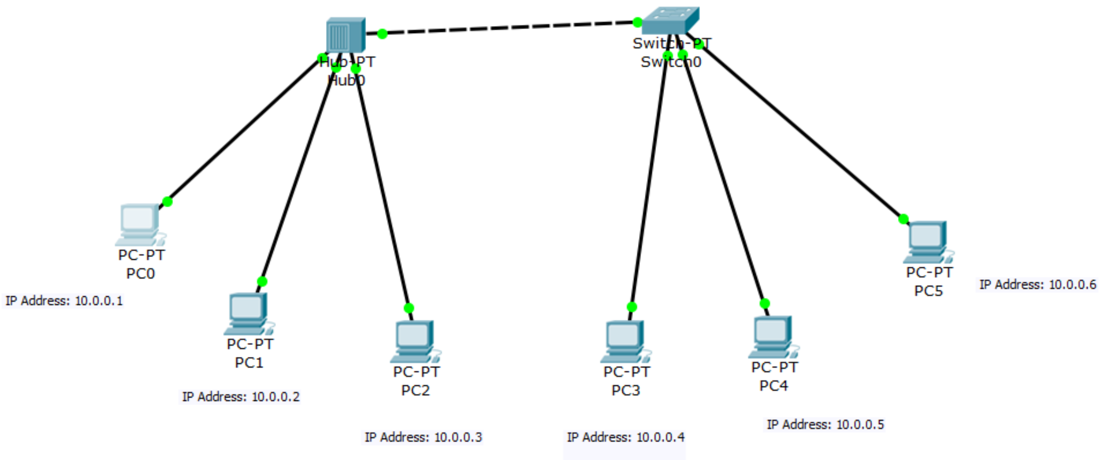
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC2	ICMP	🟩	0.000	N	0	(edit)	(delete)

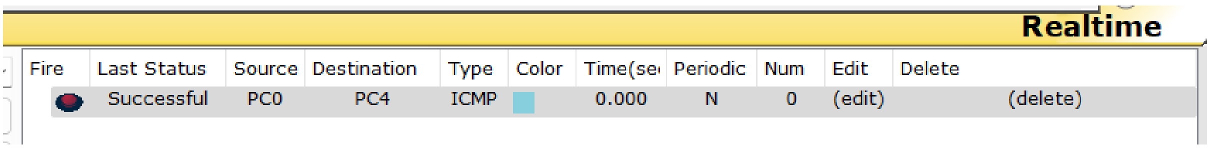
Hub:



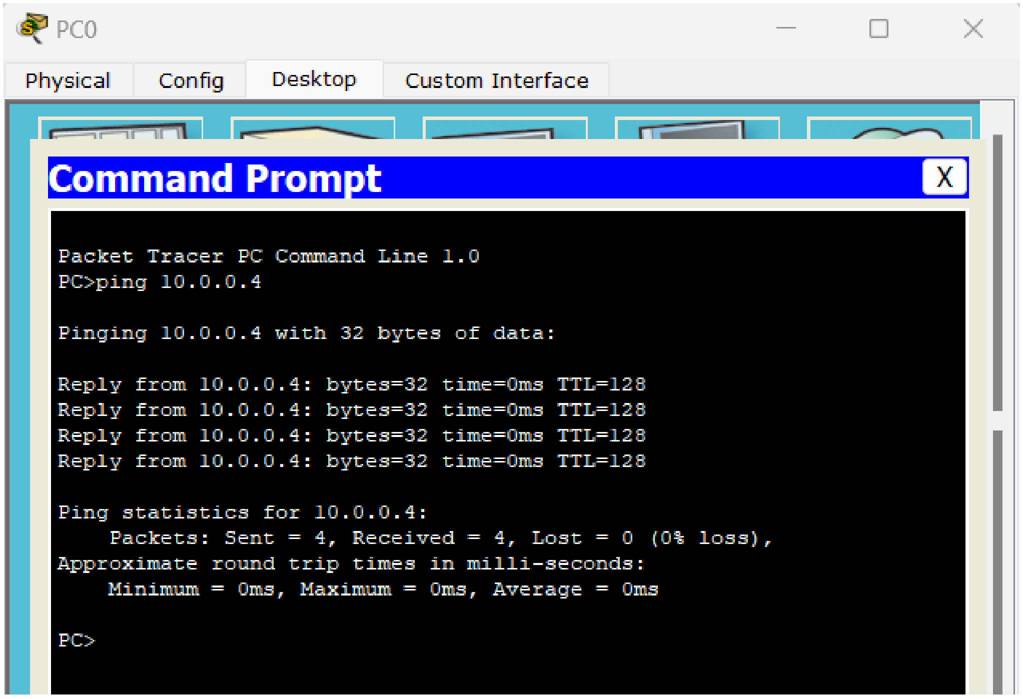


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC2	ICMP	🟪	0.000	N	0	(edit)	(delete)





Fire	Last Status	Source	Destination	Type	Color	Time(se	Periodic	Num	Edit	Delete
●	Successful	PC0	PC4	ICMP	▨	0.000	N	0	(edit)	(delete)



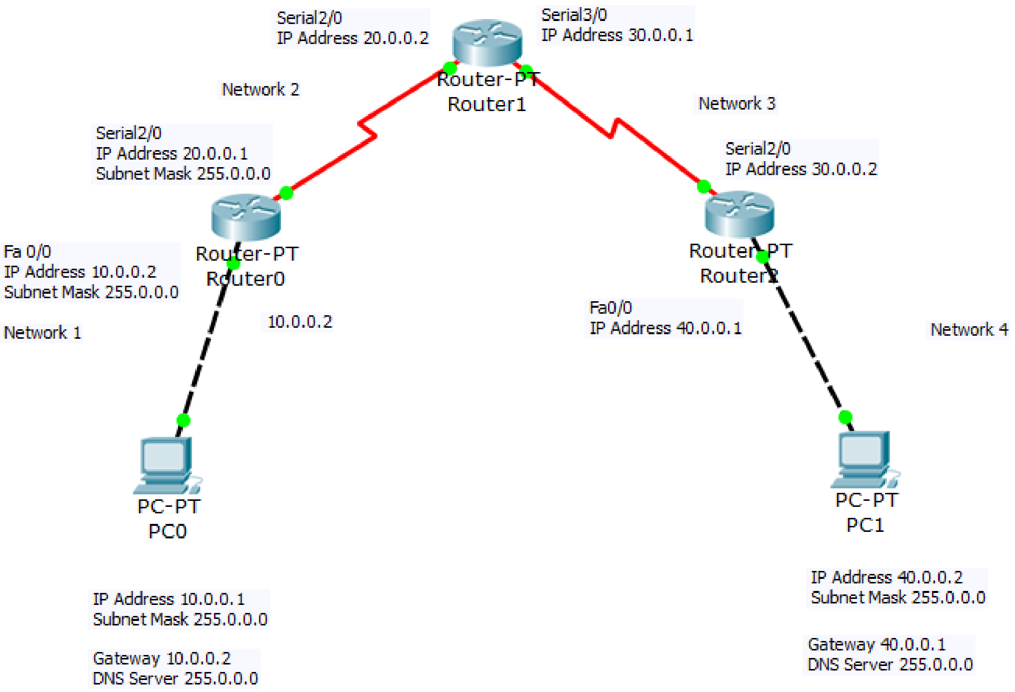
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

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

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

PC>
```

No gateway was given in this, only configuration was done using IP address by click on end devices and then config and then fast ethernet and there typing IP address 10.0.0.1 and so on and then for subnet mask, just click on it.

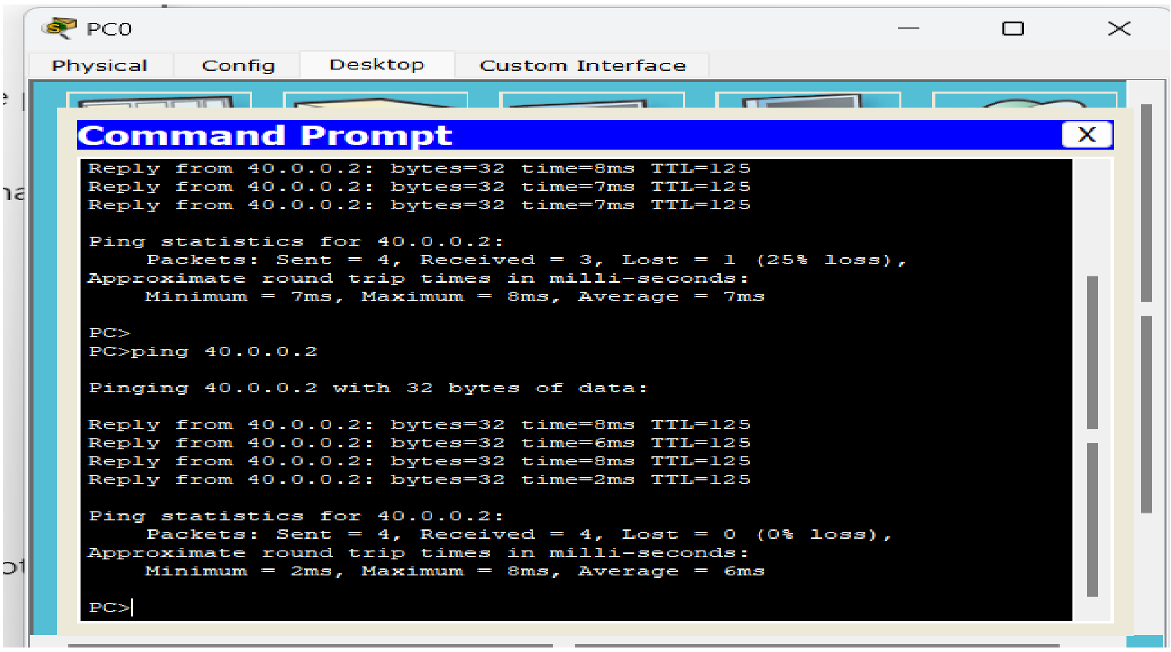
EXPERIMENT - 2

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

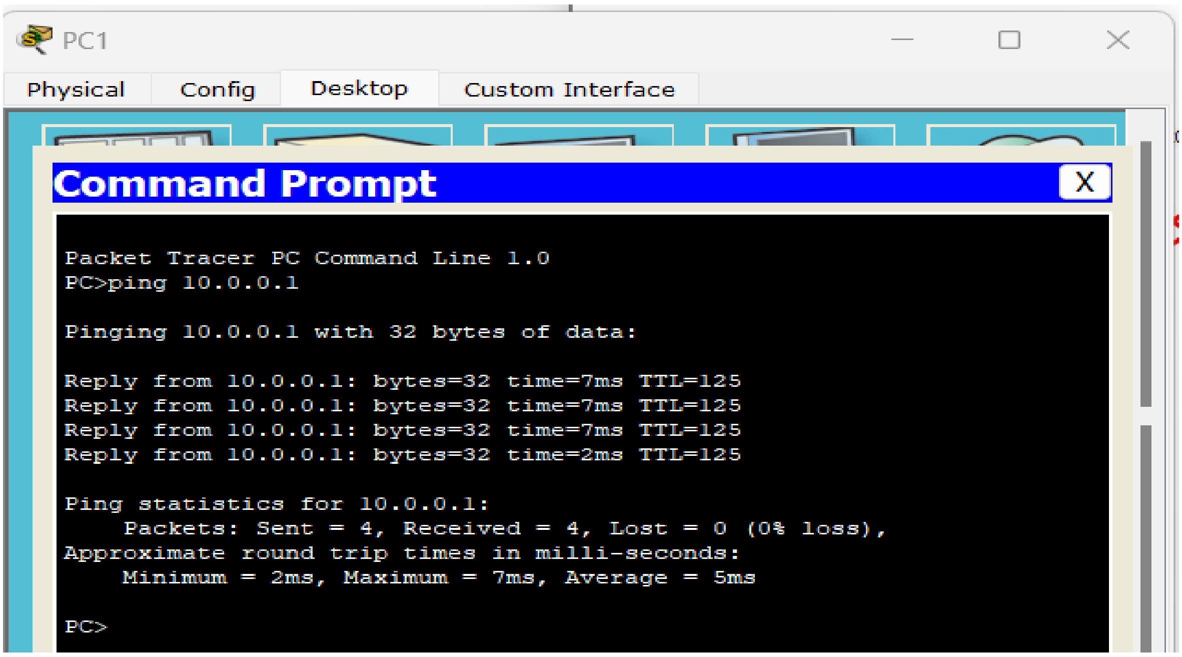
The **Default Gateway** for **PC0** should be the IP address of the router's interface that is directly connected to **PC0**, which in this case is **Router 0**.



1. **Ping Response (Successful Ping)**:
   When you ping another device (like a router or PC), and the destination is reachable, you will get a **response**. This means the device is reachable and responding to your requests.



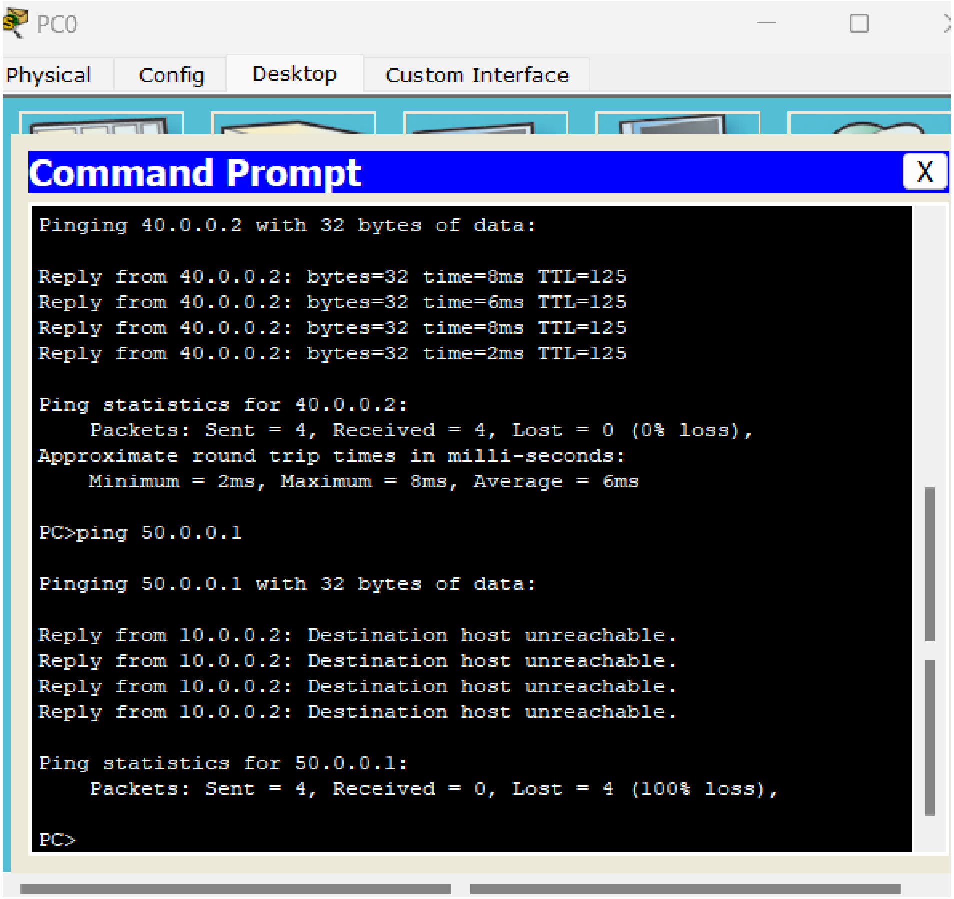
```
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 8ms, Average = 7ms

PC>
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 6ms

PC>
```



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

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=7ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125

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 = 7ms, Average = 5ms

PC>
```

2. **Destination Unreachable**:

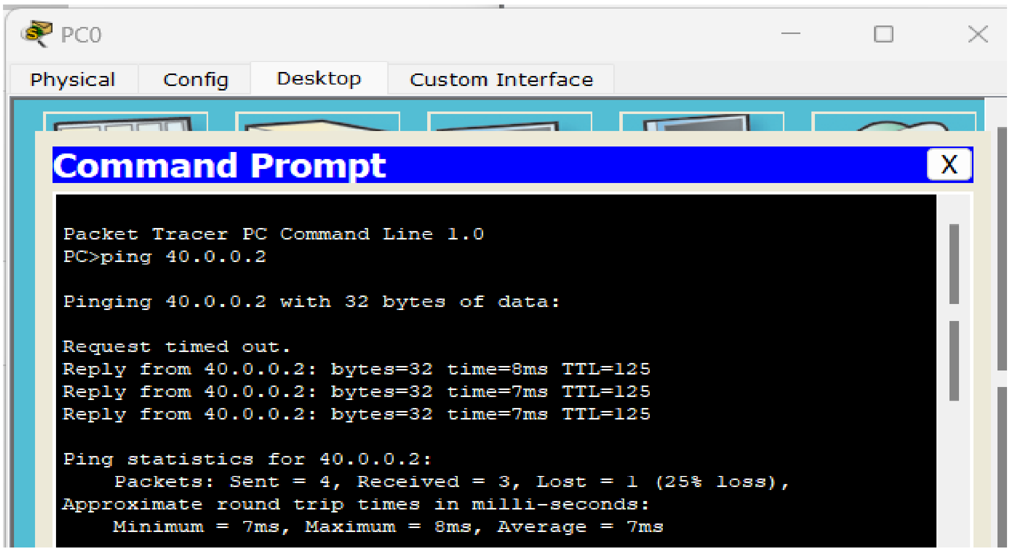
If you ping a device and it cannot be reached, you will get a **"Destination Unreachable"** message. This could happen due to a variety of reasons, such as incorrect IP addressing, routing issues, or firewall blocking.



```
Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 6ms

PC>ping 50.0.0.1

Pinging 50.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>
```

3. **Request Timed Out**:

If the ping request is sent, but no response is received within the timeout period, you will get a **"Request Timed Out"** message. This usually happens due to network congestion, incorrect routing, or the destination device not responding in time.



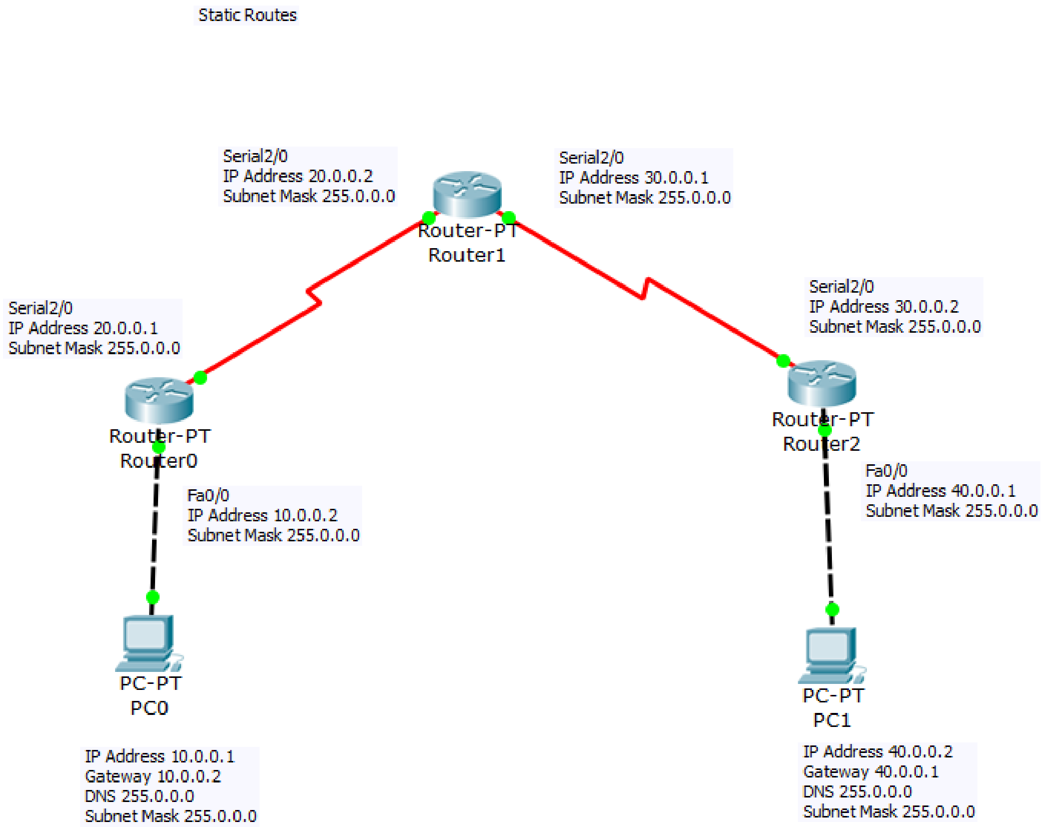
```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 8ms, Average = 7ms
```

EXPERIMENT - 3

AIM: Configure static route to the Router.

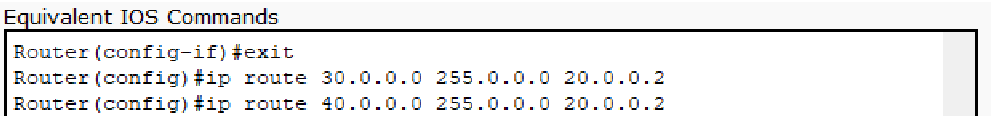
Default and Static Routes

- **Default Route:** A route used to forward packets to a destination not explicitly listed in the routing table. Represented as 0.0.0.0/0 in IPv4 or ::/0 in IPv6.
- **Static Route:** A manually configured route that defines a specific path to a destination network.



Static Route:

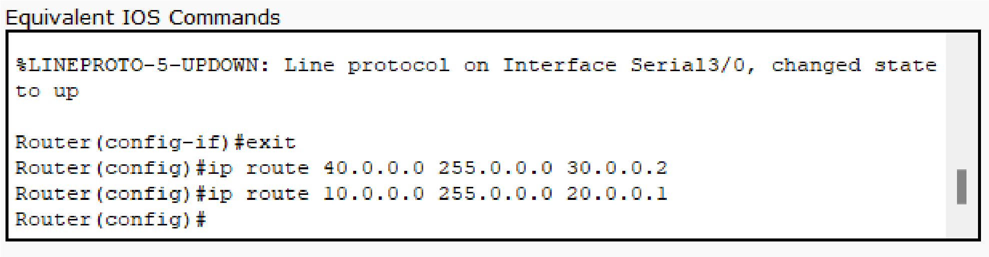
Router 0:


Equivalent IOS Commands

```
Router(config-if)#exit
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
```

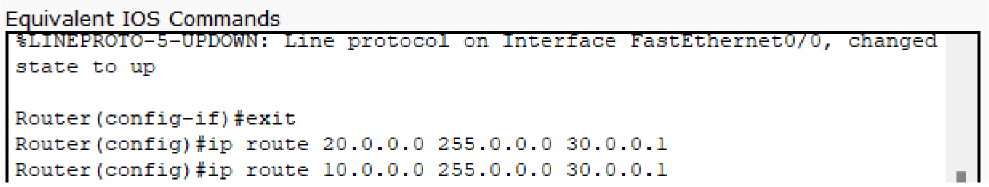
Router 1:

Only Static Route:



```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to up

Router(config-if)#exit
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#
```

Router 2:



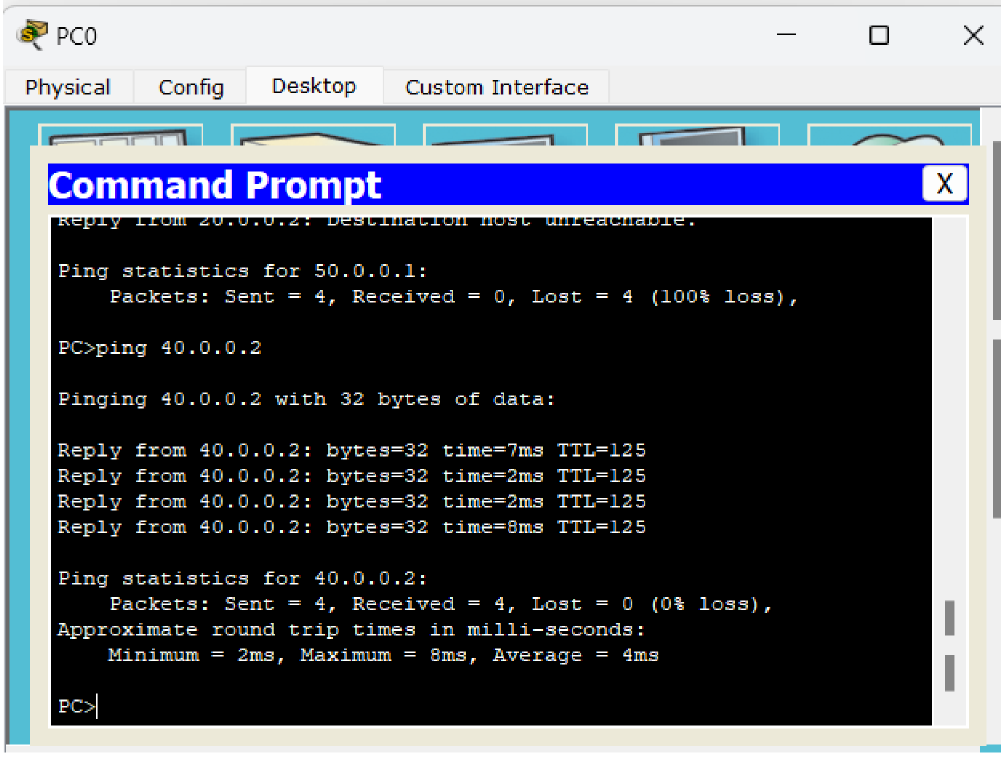
```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up

Router(config-if)#exit
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
```

Verify Connectivity

Use the following commands to test and verify connectivity:

- **From PC0 to PC1:** Open PC0's Command Prompt:

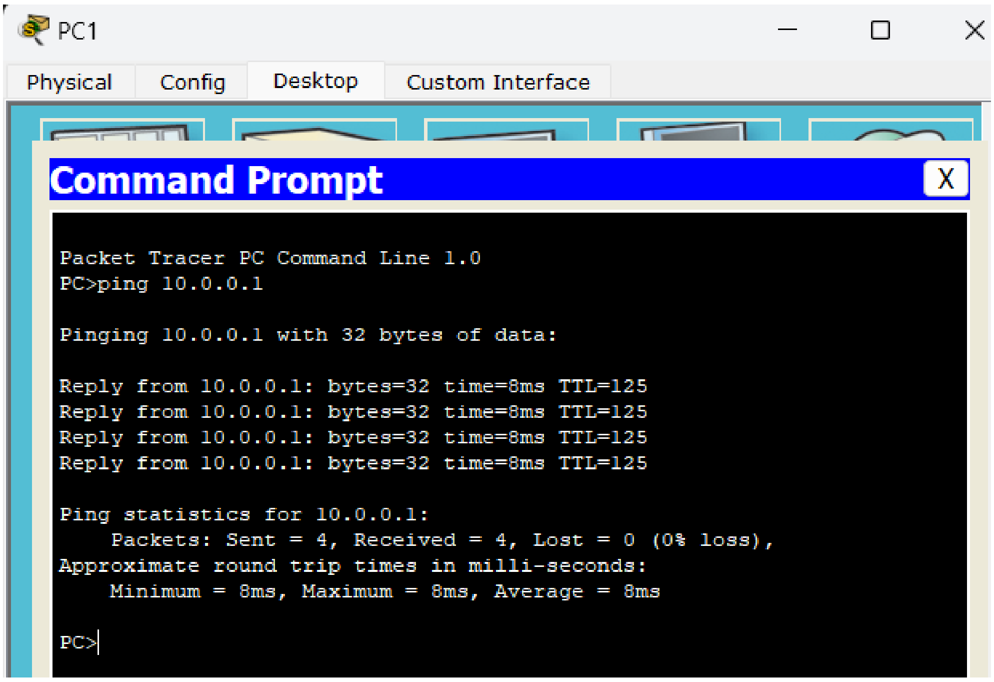
ping 40.0.0.2



```
Reply from 20.0.0.2: Destination host unreachable.

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 4ms

PC>
```

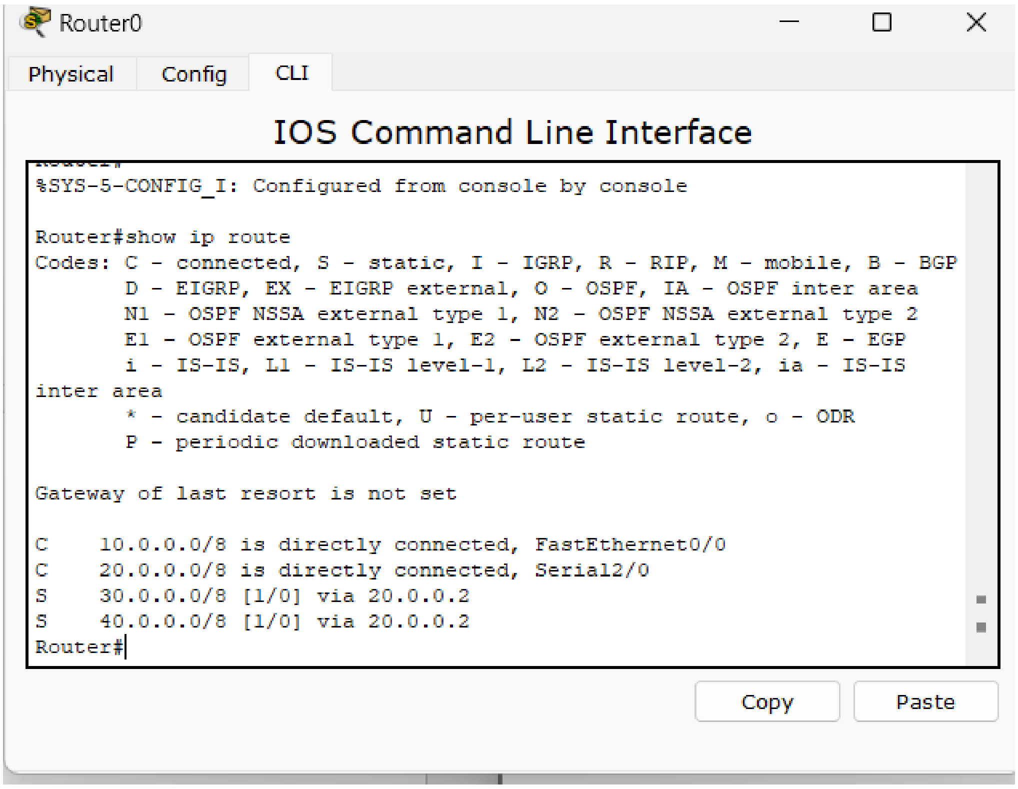
- **From PC1 to PC0:** Open PC1's Command Prompt:

ping 10.0.0.1

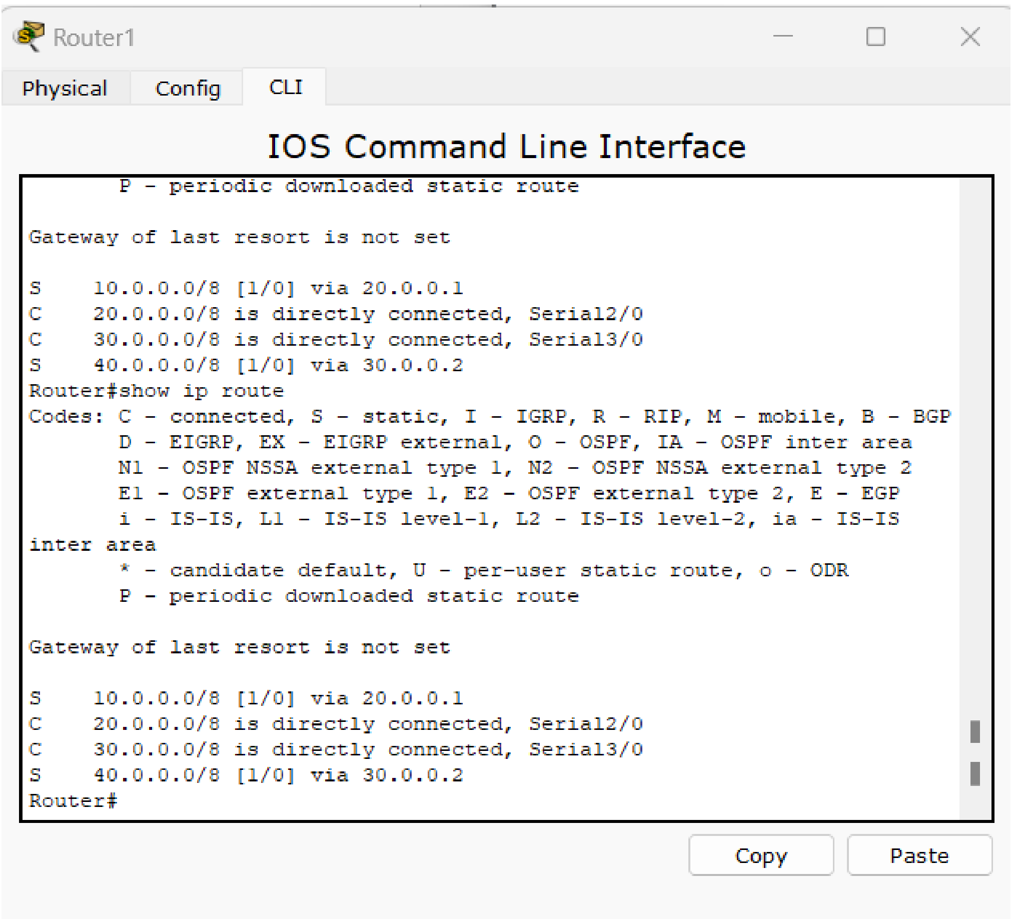


```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 8ms, Average = 8ms

PC>
```

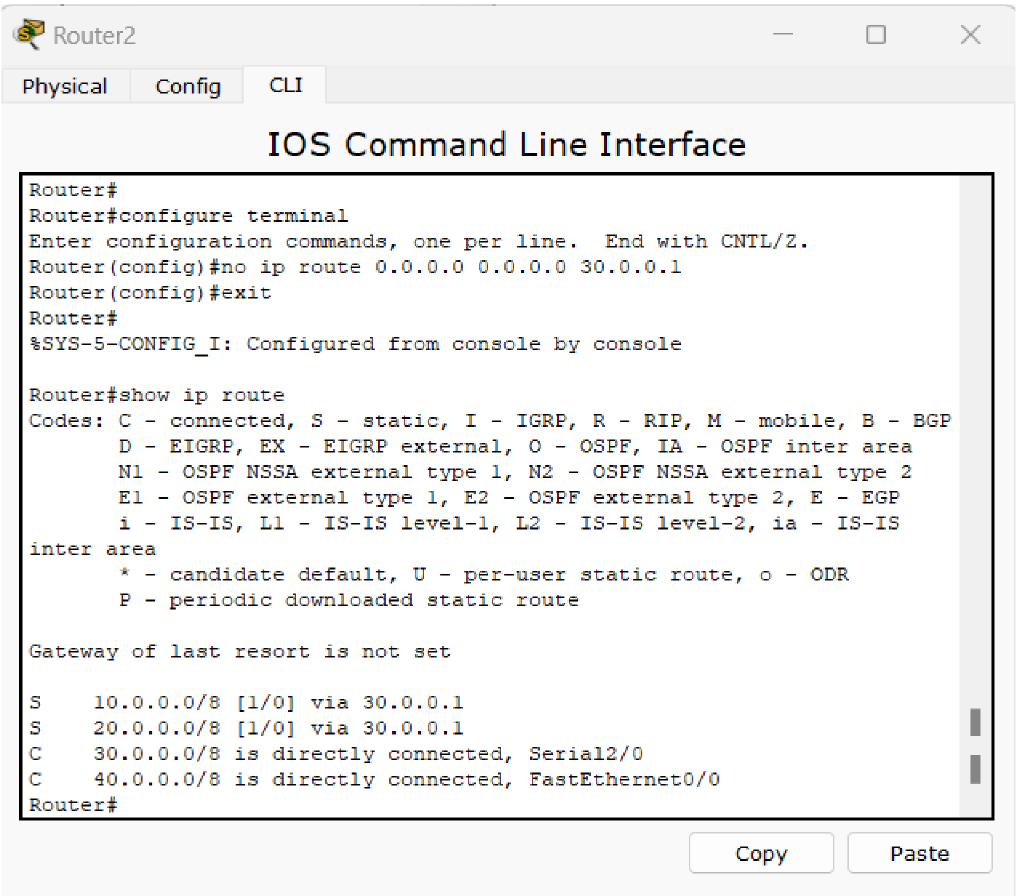
show ip route



```
%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

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2
Router#
```



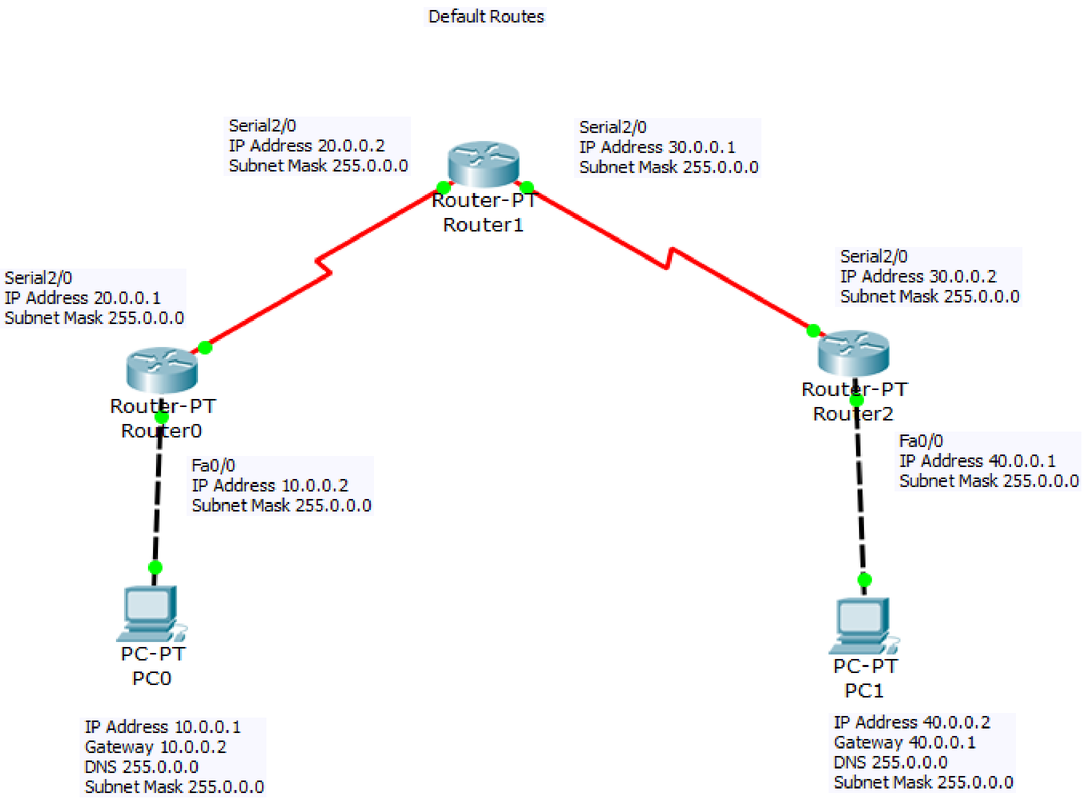
```
       P - periodic downloaded static route

Gateway of last resort is not set

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 30.0.0.2
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

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 30.0.0.2
Router#
```



```
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#no ip route 0.0.0.0 0.0.0.0 30.0.0.1
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

S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
Router#
```

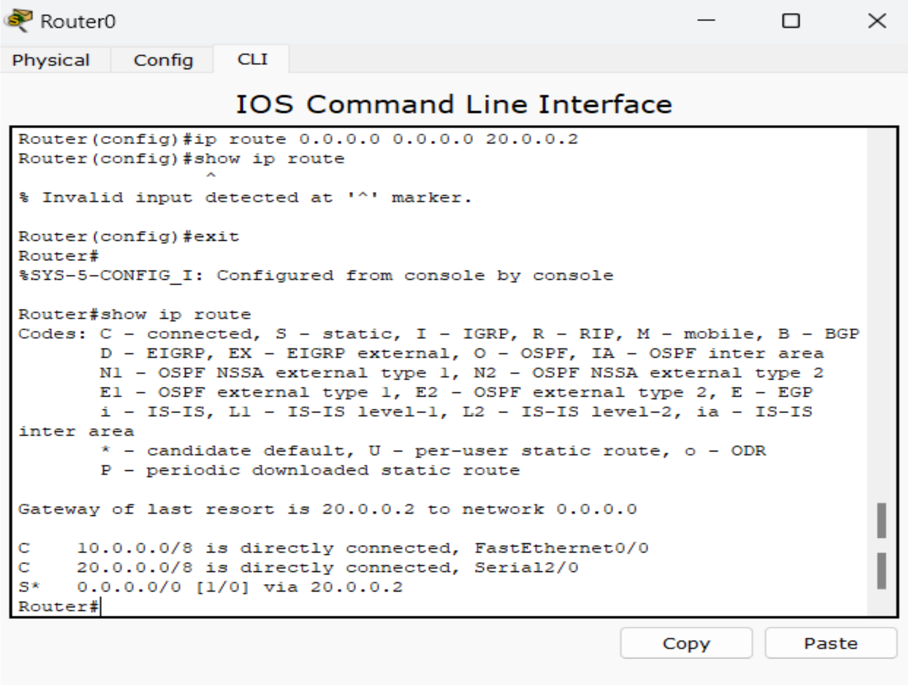
EXPERIMENT – 4 A

AIM: Configure default route to the Router.



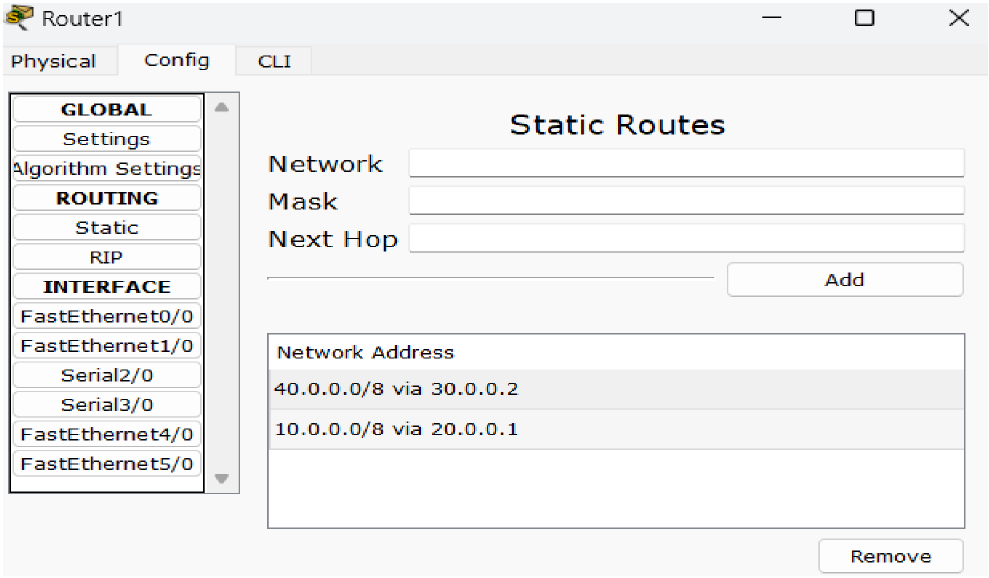
show ip route

Router 0:



```
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)#show ip route
                  ^
% Invalid input detected at '^' marker.

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 20.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
Router#
```

Router 1:



Static Routes

Network

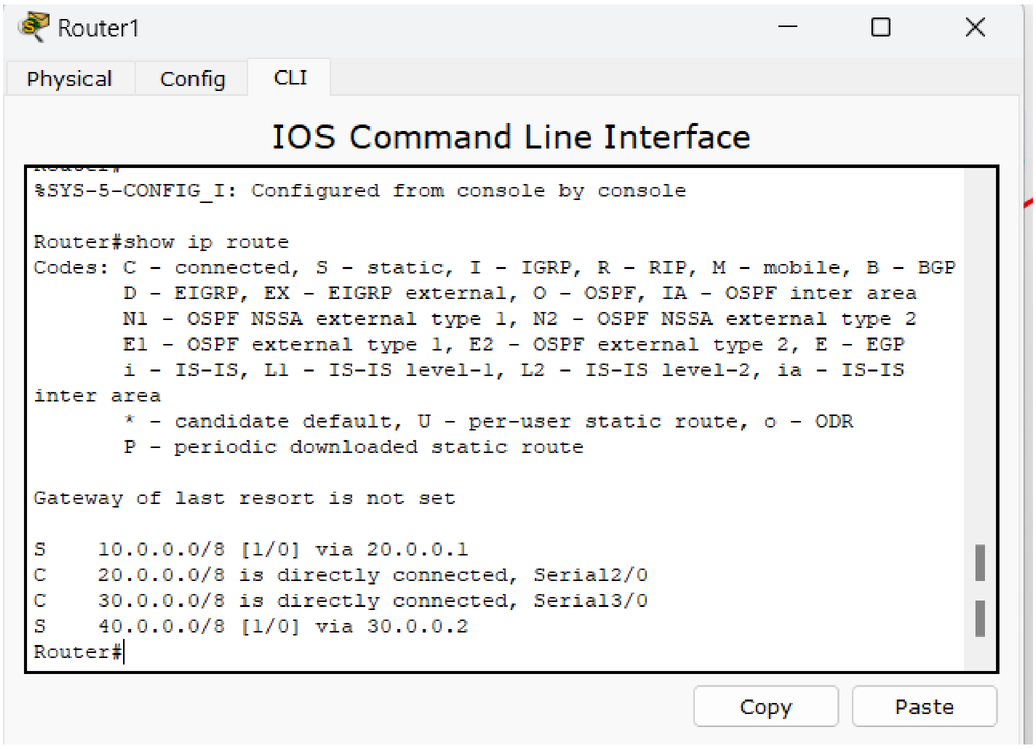
Mask

Next Hop

Network Address

40.0.0.0/8 via 30.0.0.2

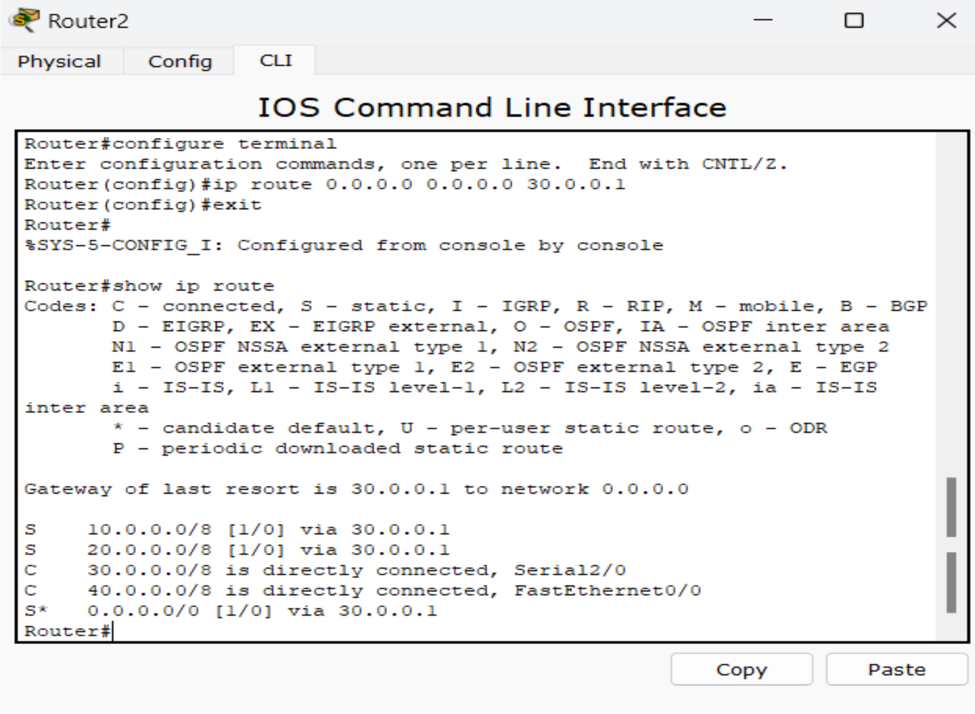
10.0.0.0/8 via 20.0.0.1



IOS Command Line Interface

```
%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

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 30.0.0.2
Router#
```

Router 2:



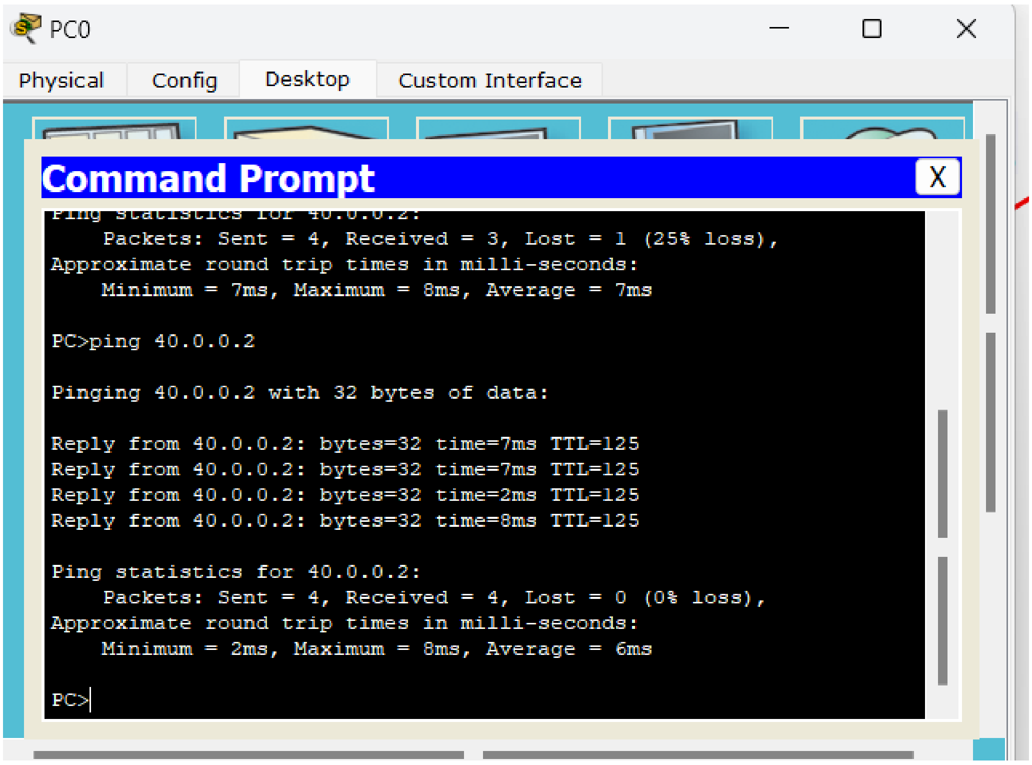
```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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 30.0.0.1 to network 0.0.0.0

S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
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#
```

Verify Connectivity

Use the following commands to test and verify connectivity:

- **From PC0 to PC1:** Open PC0's Command Prompt:

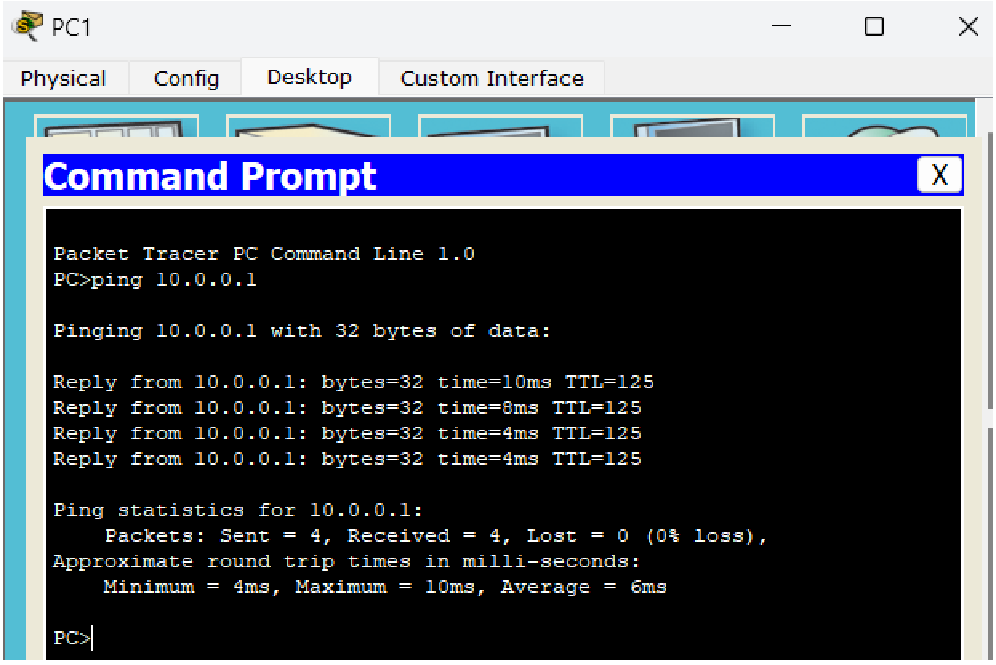
ping 40.0.0.2



```
Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 8ms, Average = 7ms

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=7ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=8ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 6ms

PC>
```

- **From PC1 to PC0:** Open PC1's Command Prompt:

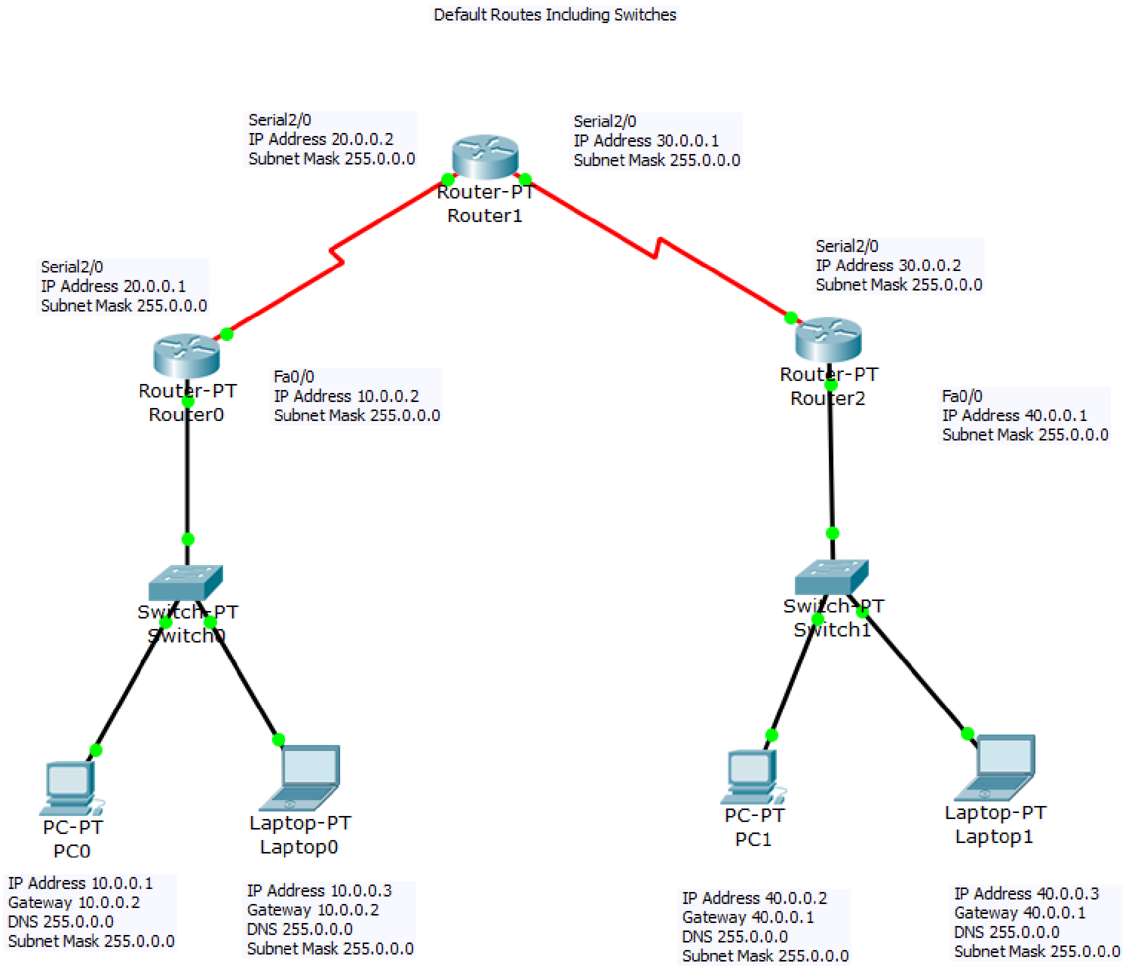
ping 10.0.0.1



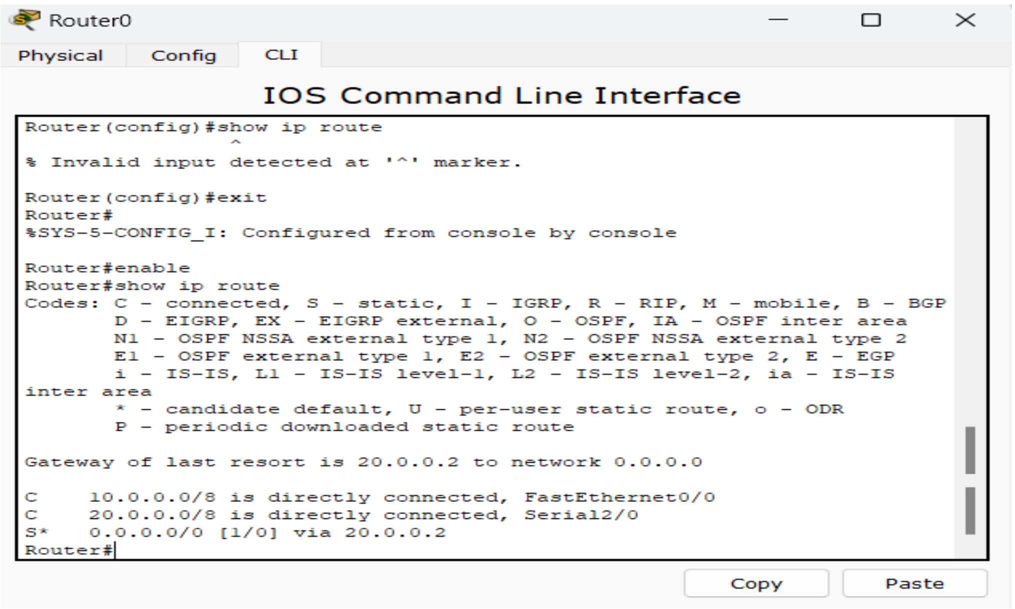
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=10ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=4ms TTL=125
Reply from 10.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 10ms, Average = 6ms

PC>
```

EXPERIMENT – 4 B

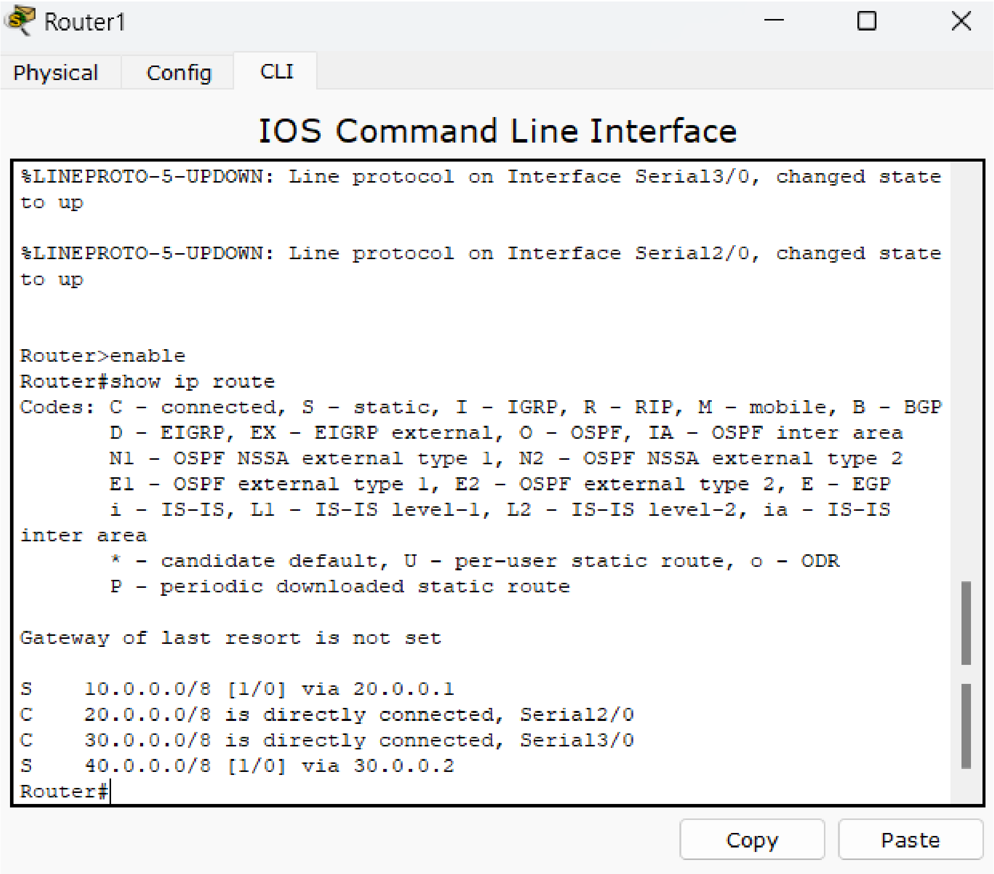
AIM: Configure default route to the Router.



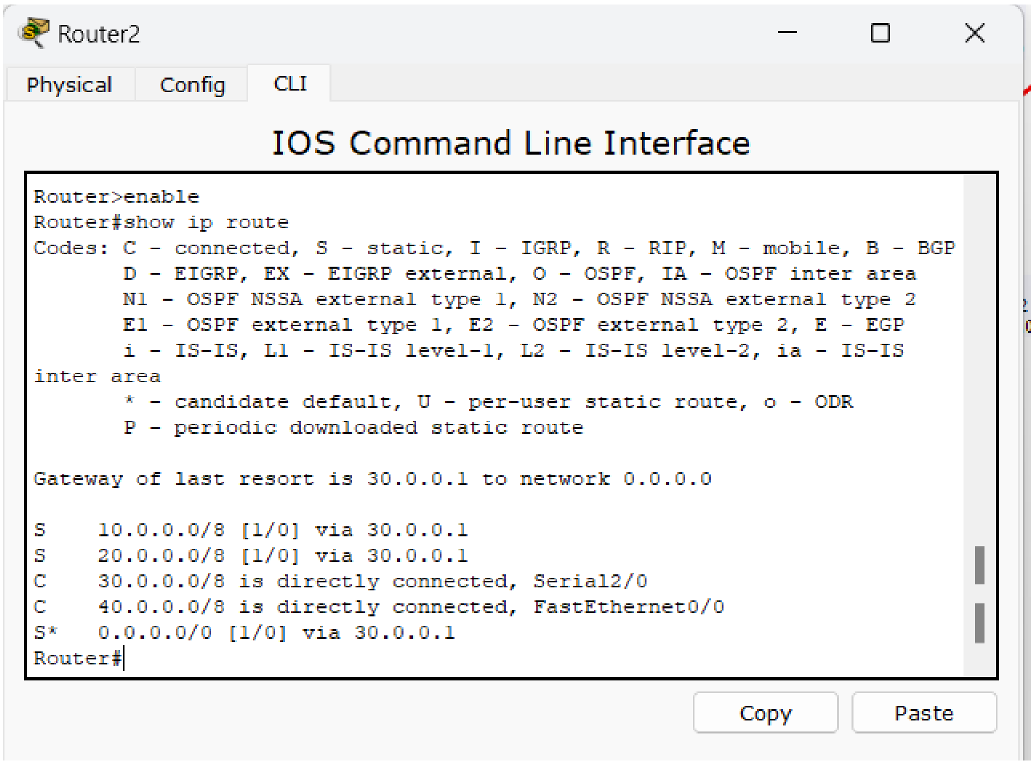
show ip route



```
Router(config)#show ip route
                ^
% Invalid input detected at '^' marker.

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

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 20.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
Router#
```



```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to up

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


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

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 30.0.0.2
Router#
```



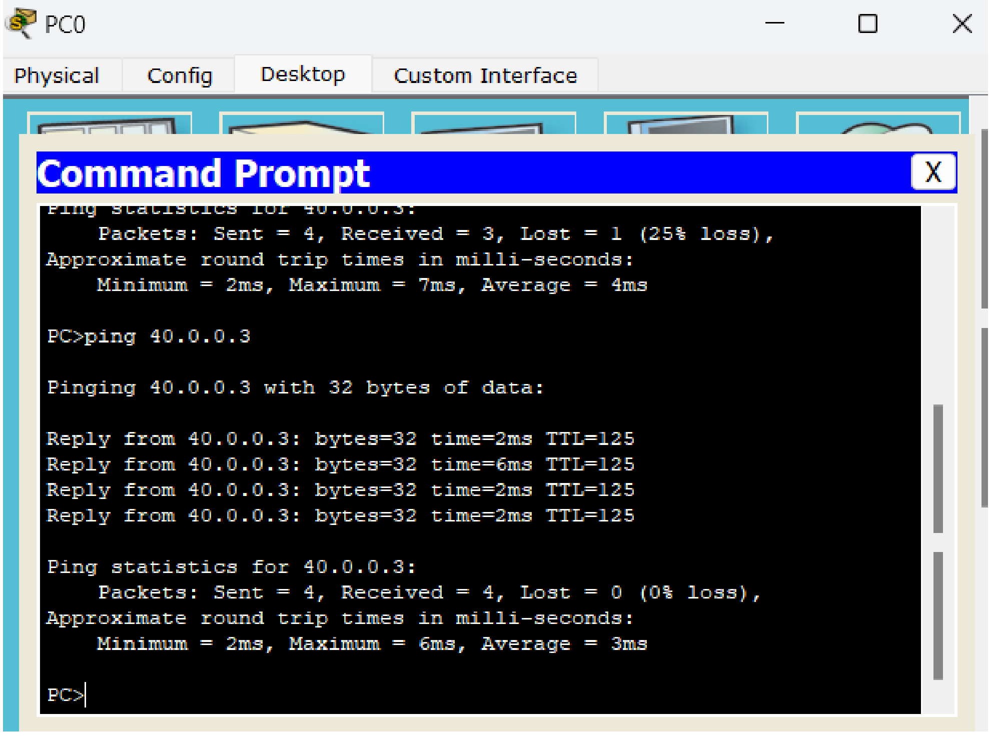
```
Router>enable
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external 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 30.0.0.1 to network 0.0.0.0

S    10.0.0.0/8 [1/0] via 30.0.0.1
S    20.0.0.0/8 [1/0] via 30.0.0.1
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#
```

Verify Connectivity

Use the following commands to test and verify connectivity:

- **From PC0 to Laptop1:** Open PC0's Command Prompt:

ping 40.0.0.3



```
Ping statistics for 40.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 4ms

PC>ping 40.0.0.3

Pinging 40.0.0.3 with 32 bytes of data:

Reply from 40.0.0.3: bytes=32 time=2ms TTL=125
Reply from 40.0.0.3: bytes=32 time=6ms TTL=125
Reply from 40.0.0.3: bytes=32 time=2ms TTL=125
Reply from 40.0.0.3: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 3ms

PC>
```

EXPERIMENT – 5

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

1. Understanding TELNET

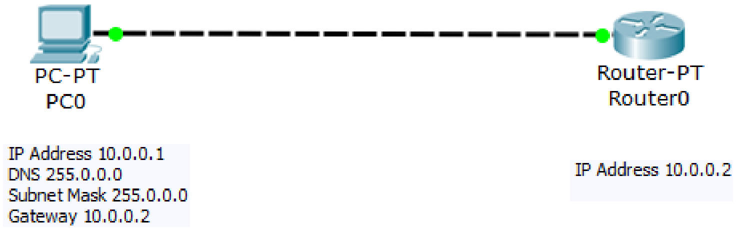
- TELNET is a network protocol used to provide bidirectional interactive communication over a TCP/IP network.
- It allows users to remotely access and manage network devices like routers, switches, and servers using command-line interfaces.

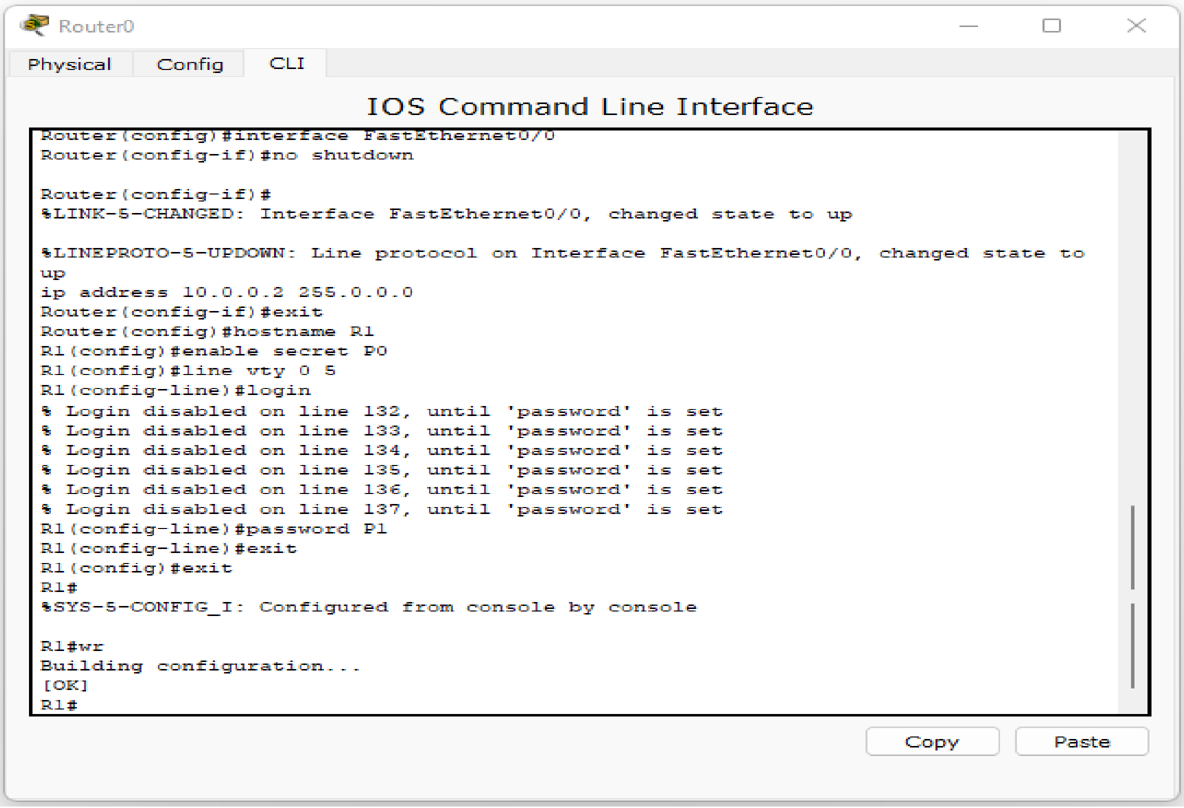
---

2. Pre-requisites

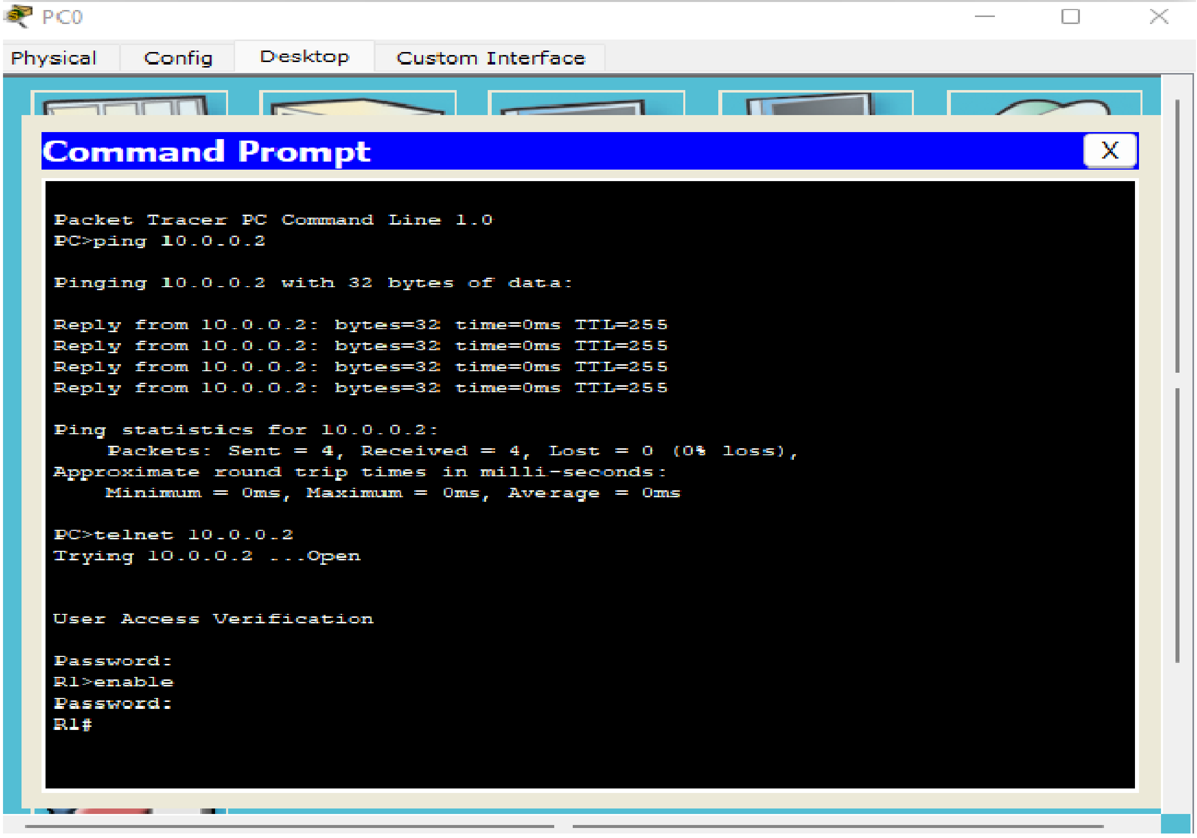
- **Router Configuration:**
  - TELNET service must be enabled on the router.
  - An IP address should be assigned to the router interface that is accessible from the PC in the IT office.
  - The router must have a username and password set up for authentication.

Topology:





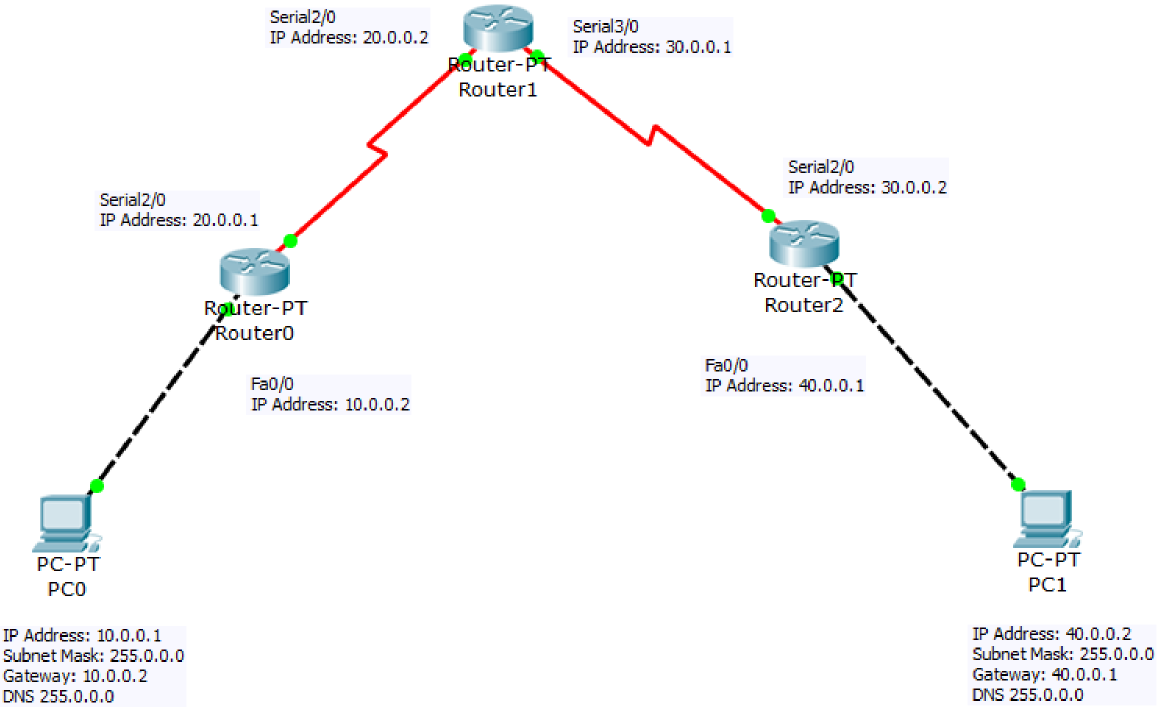
```
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown

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
ip address 10.0.0.2 255.0.0.0
Router(config-if)#exit
Router(config)#hostname R1
R1(config)#enable secret P0
R1(config)#line vty 0 5
R1(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
% Login disabled on line 136, until 'password' is set
% Login disabled on line 137, until 'password' is set
R1(config-line)#password P1
R1(config-line)#exit
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
```



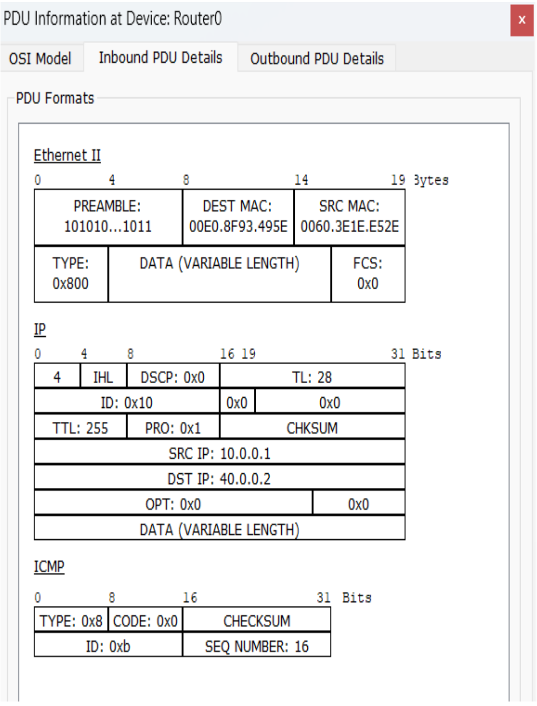
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

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

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
R1>enable
Password:
R1#
```

EXPERIMENT – 6

AIM: Demonstrate the TTL/ Life of a Packet.



Inbound and Outbound PDU Details:



Ethernet II

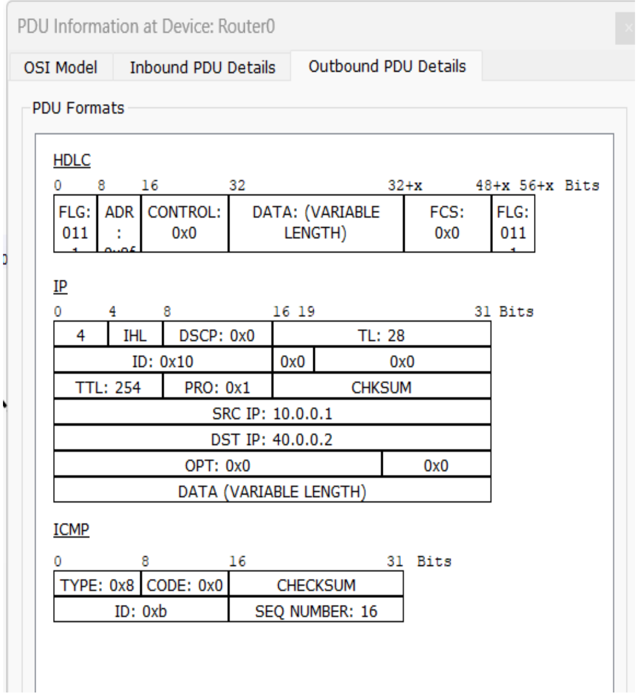
0	4	8	14	19 Bytes
PREAMBLE: 101010...1011		DEST MAC: 00E0.8F93.495E	SRC MAC: 0060.3E1E.E52E	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x10			0x0	0x0
TTL: 255		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.1				
DST IP: 40.0.0.2				
OPT: 0x0				0x0
DATA (VARIABLE LENGTH)				

ICMP

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



HDLC

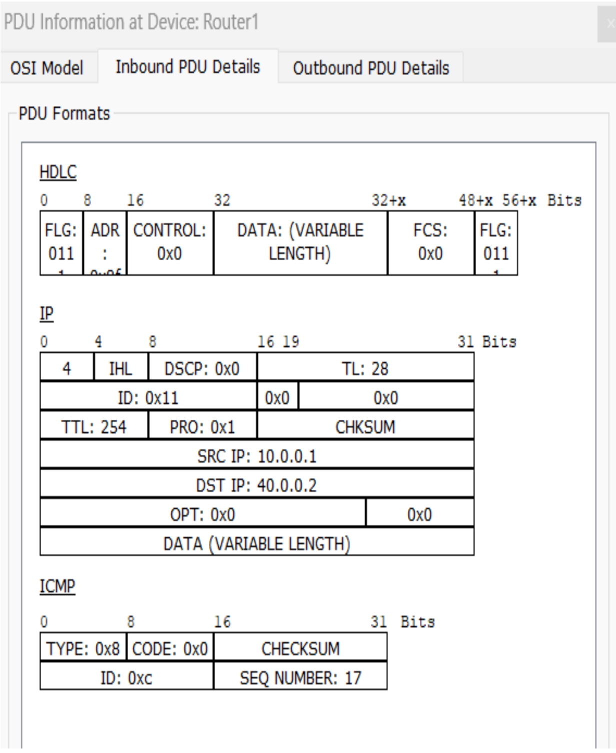
0	8	16	32	32+x	48+x	56+x Bits
FLG: 011	ADR :	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 011	

IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x10			0x0	0x0
TTL: 254		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.1				
DST IP: 40.0.0.2				
OPT: 0x0				0x0
DATA (VARIABLE LENGTH)				

ICMP

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



PDU Information at Device: Router1

PDU Formats

HDLC

0 8 16 32 32+x 48+x 56+x Bits

FLG: 011	ADR :	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 011

IP

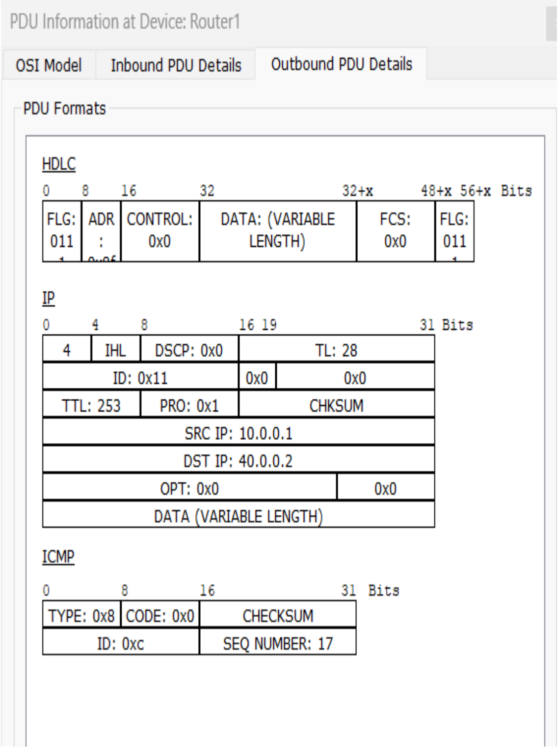
0 4 8 16 19 31 Bits

4 | IHL | DSCP: 0x0 | TL: 28
ID: 0x11 | 0x0 | 0x0
TTL: 254 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)

ICMP

0 8 16 31 Bits

TYPE: 0x8 | CODE: 0x0 | CHECKSUM
ID: 0xc | SEQ NUMBER: 17



PDU Information at Device: Router1

PDU Formats

HDLC

0 8 16 32 32+x 48+x 56+x Bits

FLG: 011	ADR :	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 011

IP

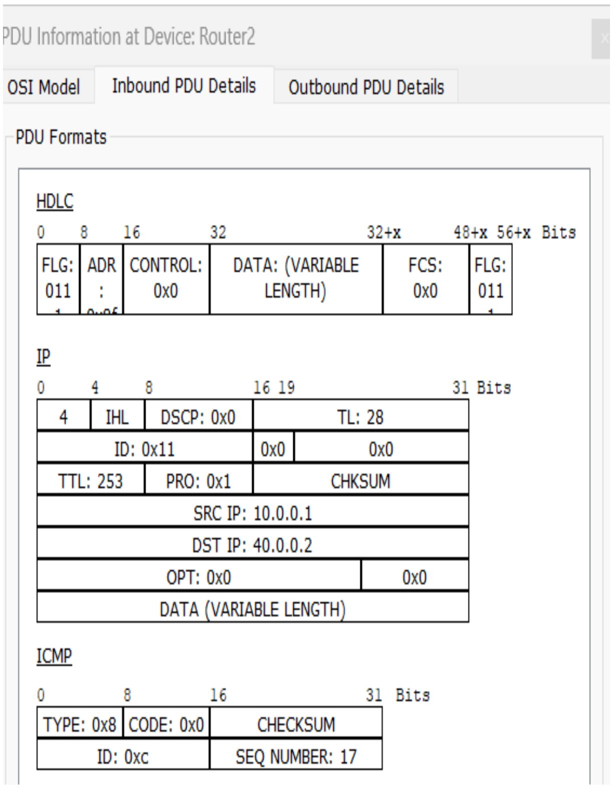
0 4 8 16 19 31 Bits

4 | IHL | DSCP: 0x0 | TL: 28
ID: 0x11 | 0x0 | 0x0
TTL: 253 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)

ICMP

0 8 16 31 Bits

TYPE: 0x8 | CODE: 0x0 | CHECKSUM
ID: 0xc | SEQ NUMBER: 17



PDU Information at Device: Router2

PDU Formats

HDLC

0 8 16 32 32+x 48+x 56+x Bits

FLG: 011	ADR :	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 011

IP

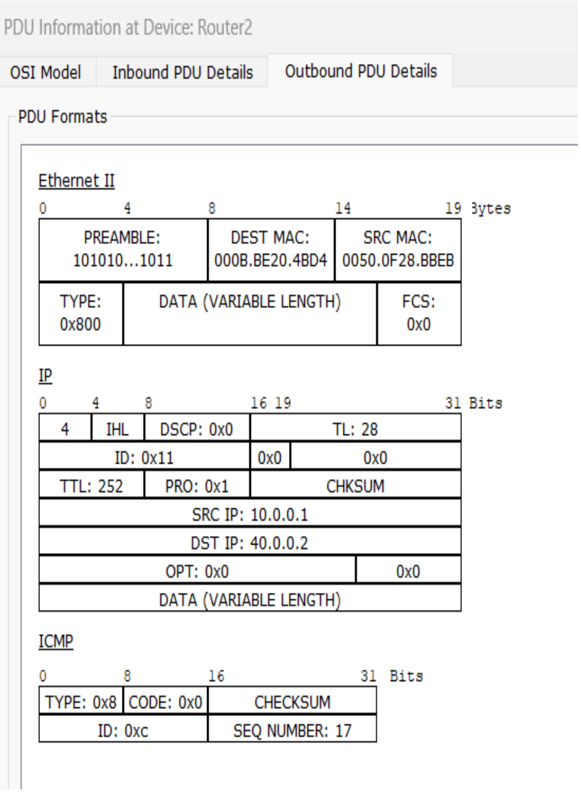
0 4 8 16 19 31 Bits

4 | IHL | DSCP: 0x0 | TL: 28
ID: 0x11 | 0x0 | 0x0
TTL: 253 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)

ICMP

0 8 16 31 Bits

TYPE: 0x8 | CODE: 0x0 | CHECKSUM
ID: 0xc | SEQ NUMBER: 17



PDU Information at Device: Router2

PDU Formats

Ethernet II

0 4 8 14 19 Bytes

PREAMBLE: 101010...1011 | DEST MAC: 000B.BE20.4BD4 | SRC MAC: 0050.0F28.BBEB
TYPE: 0x800 | DATA (VARIABLE LENGTH) | FCS: 0x0

IP

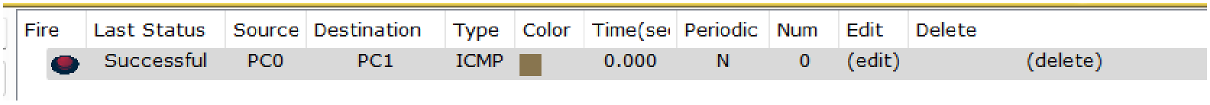
0 4 8 16 19 31 Bits

4 | IHL | DSCP: 0x0 | TL: 28
ID: 0x11 | 0x0 | 0x0
TTL: 252 | PRO: 0x1 | CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.2
OPT: 0x0 | 0x0
DATA (VARIABLE LENGTH)

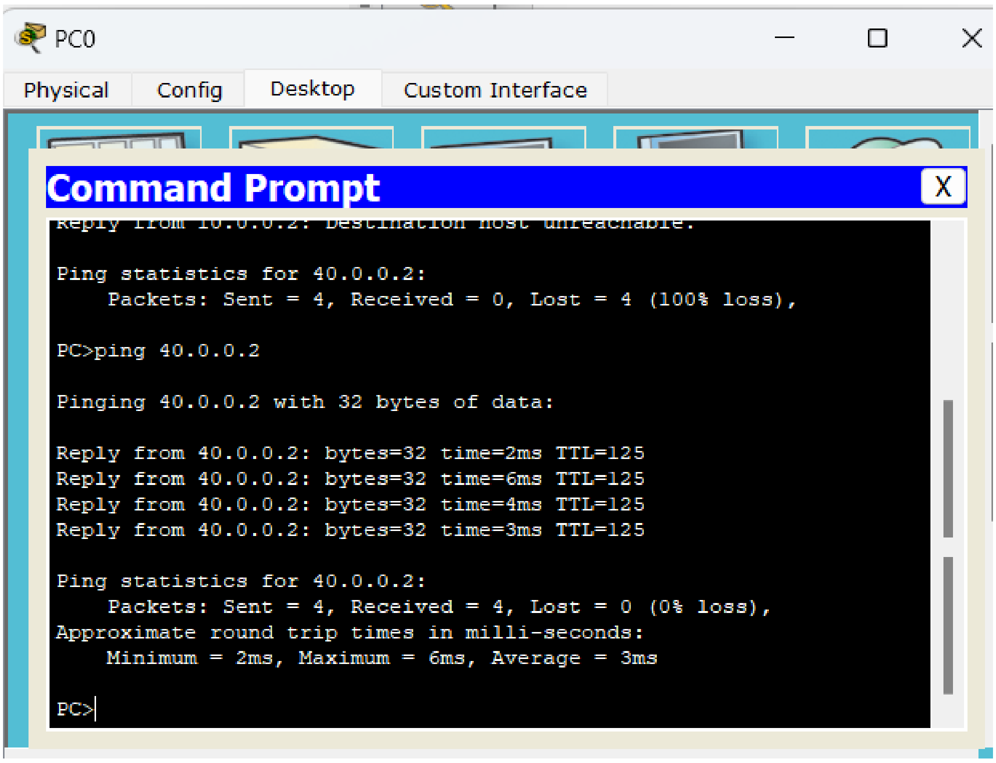
ICMP

0 8 16 31 Bits

TYPE: 0x8 | CODE: 0x0 | CHECKSUM
ID: 0xc | SEQ NUMBER: 17



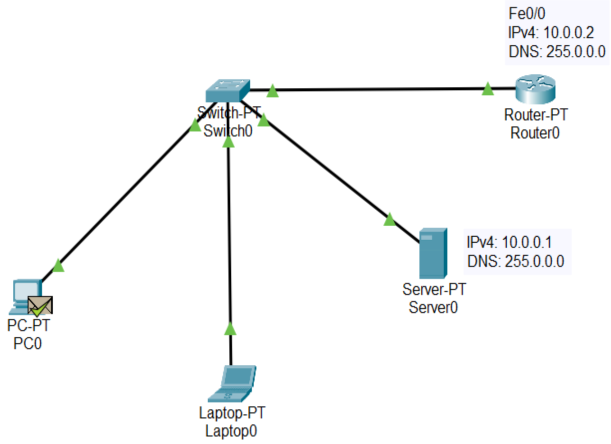
Fire	Last Status	Source	Destination	Type	Color	Time(se	Periodic	Num	Edit	Delete
●	Successful	PC0	PC1	ICMP	■	0.000	N	0	(edit)	(delete)

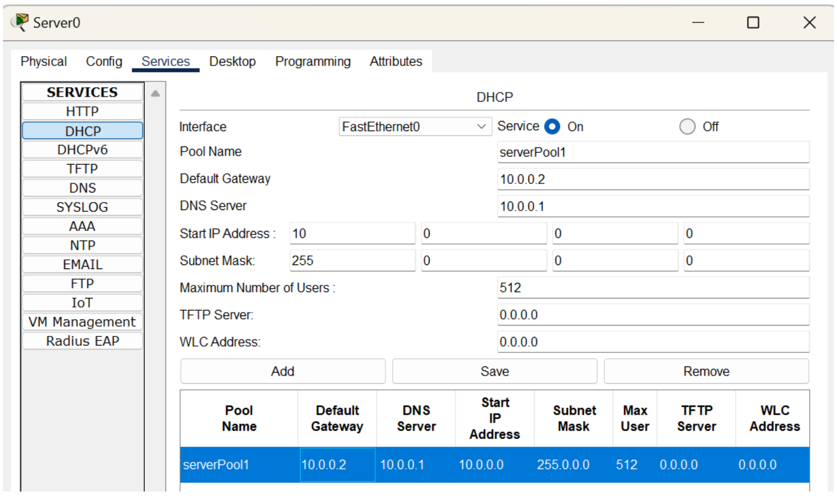


```
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=6ms TTL=125
Reply from 40.0.0.2: bytes=32 time=4ms TTL=125
Reply from 40.0.0.2: bytes=32 time=3ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 3ms

PC>
```

EXPERIMENT – 7 A

AIM: Demonstrate the TTL/ Life of a Packet.





DHCP

Interface FastEthernet0 Service ● On ○ Off

Pool Name serverPool1

Default Gateway 10.0.0.2

DNS Server 10.0.0.1

Start IP Address : 10 0 0 0

Subnet Mask: 255 0 0 0

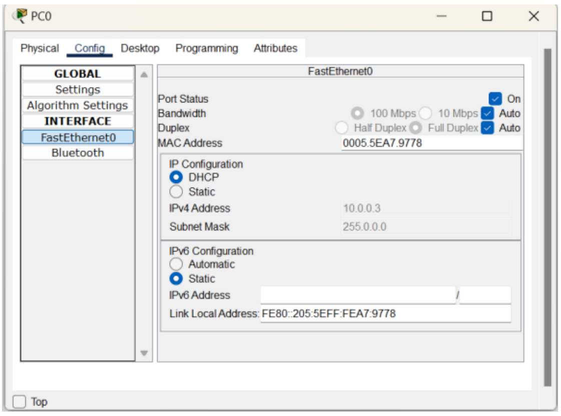
Maximum Number of Users : 512

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	10.0.0.2	10.0.0.1	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0

Figure 7.1: DHCP Service, Server0



FastEthernet0

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

MAC Address 0005.5EA7.9778

IP Configuration

● DHCP

○ Static

IPv4 Address 10.0.0.3

Subnet Mask 255.0.0.0

IPv6 Configuration

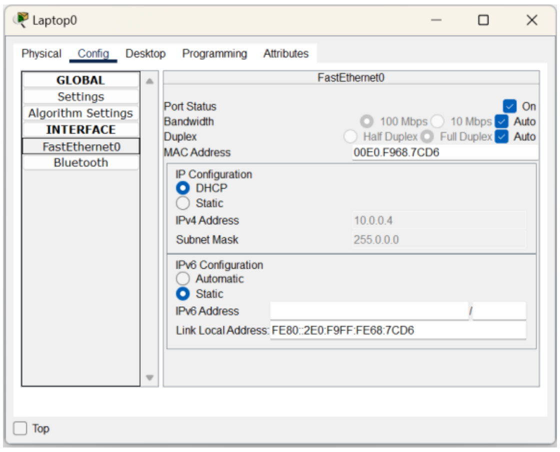
○ Automatic

● Static

IPv6 Address /

Link Local Address: FE80::205:5EFF:FEA7:9778

Figure 7.2: DHCP Service, PC0



FastEthernet0

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

MAC Address 00E0.F968.7CD6

IP Configuration

● DHCP

○ Static

IPv4 Address 10.0.0.4

Subnet Mask 255.0.0.0

IPv6 Configuration

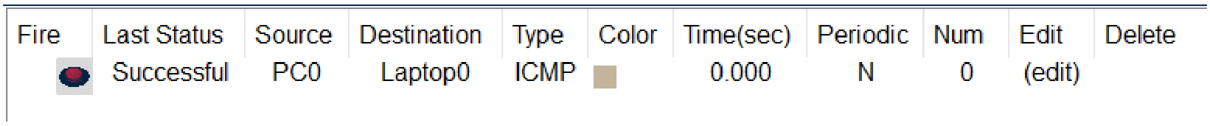
○ Automatic

● Static

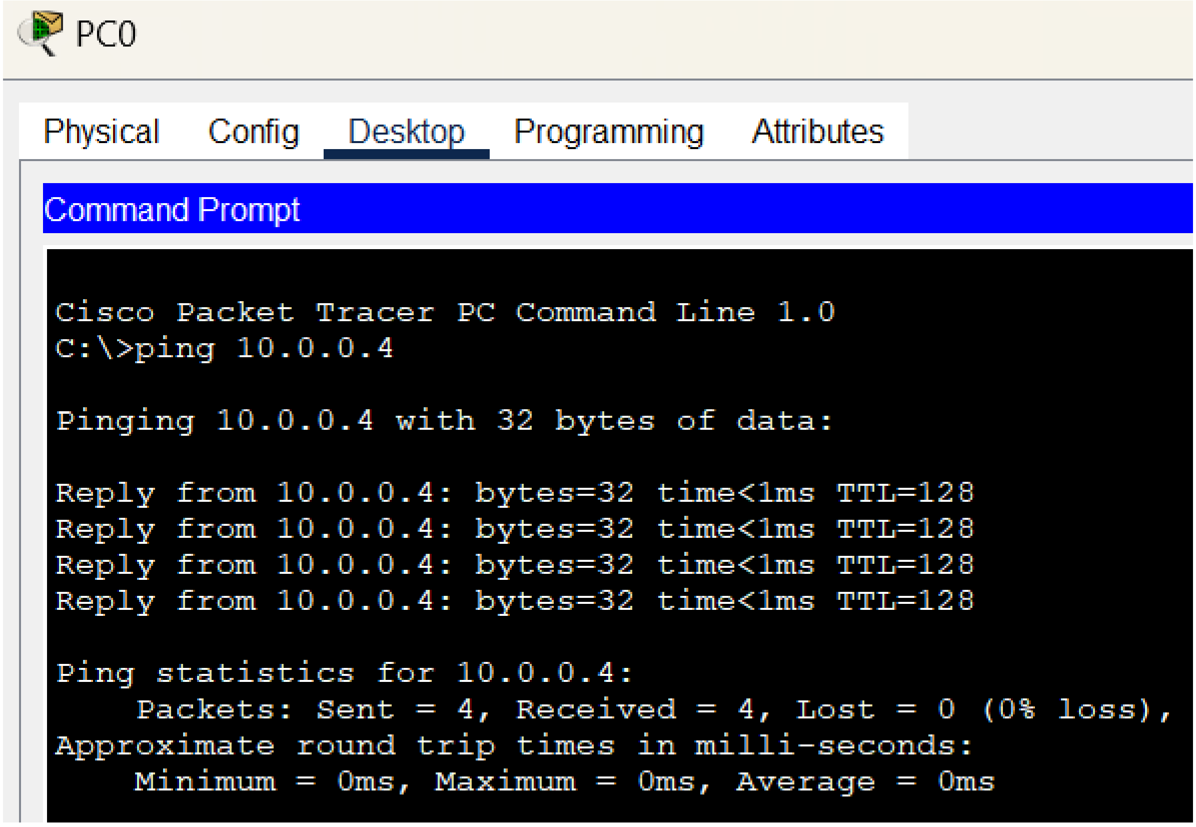
IPv6 Address /

Link Local Address: FE80::2E0:F9FF:FE68:7CD6

Figure 7.3: DHCP Service, Laptop0



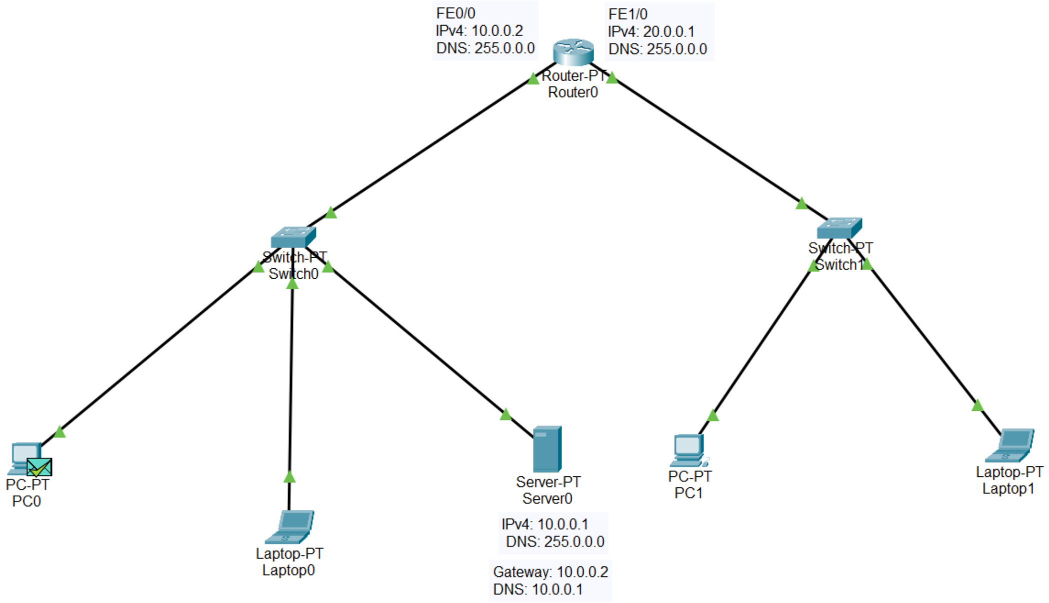
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop0	ICMP	■	0.000	N	0	(edit)	

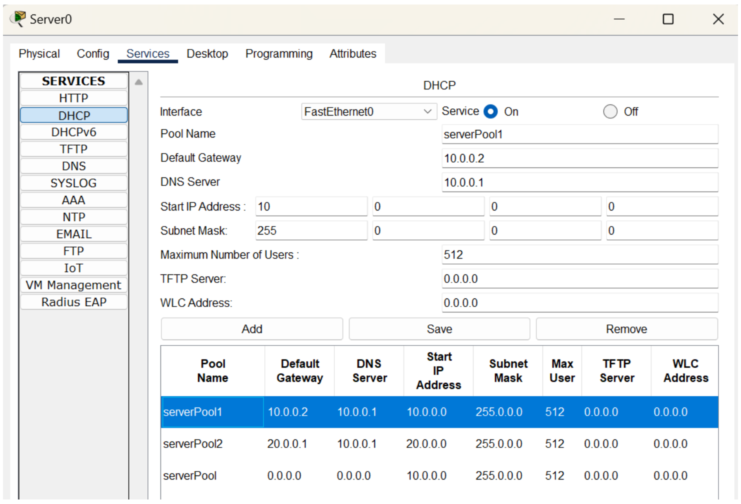


```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

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

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

EXPERIMENT – 7 B

To Configure IP addresses of the host using DHCP server outside a LAN.





DHCP

Interface FastEthernet0 Service ● On ○ Off

Pool Name serverPool1

Default Gateway 10.0.0.2

DNS Server 10.0.0.1

Start IP Address : 10 0 0 0

Subnet Mask: 255 0 0 0

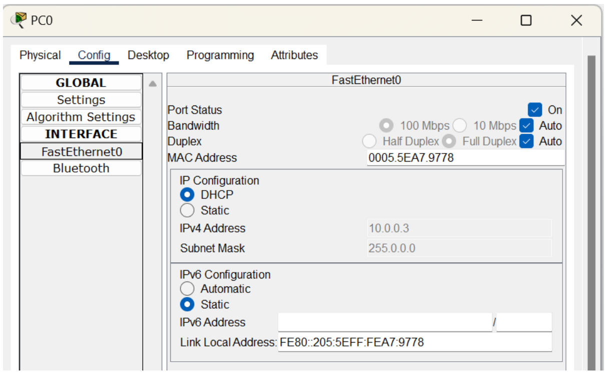
Maximum Number of Users : 512

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool1	10.0.0.2	10.0.0.1	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool2	20.0.0.1	10.0.0.1	20.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	10.0.0.0	255.0.0.0	512	0.0.0.0	0.0.0.0

Figure 7.1.1: DHCP Service, Server0



FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0005.5EA7.9778

IP Configuration
● DHCP
○ Static

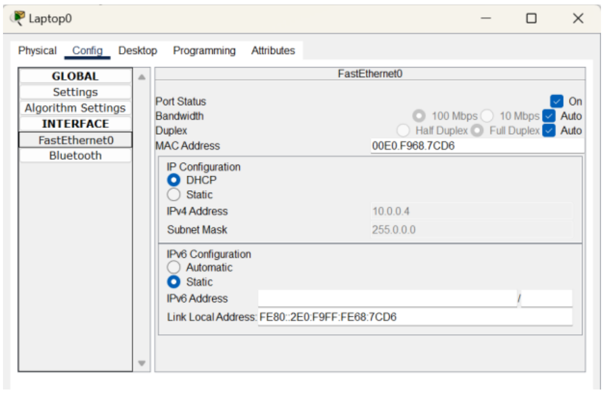
IPv4 Address 10.0.0.3

Subnet Mask 255.0.0.0

IPv6 Configuration
○ Automatic
● Static

IPv6 Address /

Link Local Address: FE80::205:5EFF:FEA7:9778



FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 00E0.F968.7CD6

IP Configuration
● DHCP
○ Static

IPv4 Address 10.0.0.4

Subnet Mask 255.0.0.0

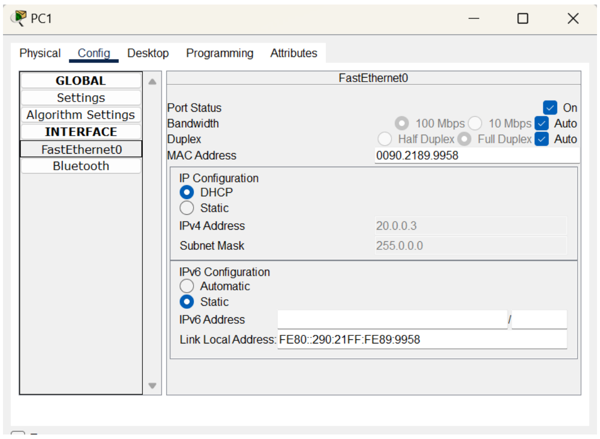
IPv6 Configuration
○ Automatic
● Static

IPv6 Address /

Link Local Address: FE80::2E0:F9FF:FE68:7CD6

Figure7.2.2: DHCP Service, PC0

Figure 7.2.3: DHCP Service, Laptop0



PC1

FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0090.2189.9958

IP Configuration

● DHCP

○ Static

IPv4 Address 20.0.0.3

Subnet Mask 255.0.0.0

IPv6 Configuration

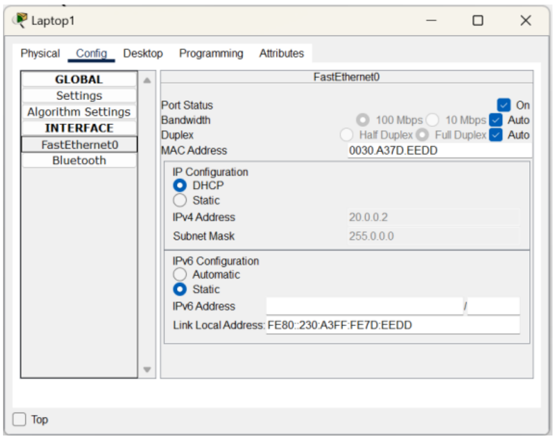
○ Automatic

● Static

IPv6 Address /

Link Local Address: FE80::290:21FF:FE89:9958

Figure 7.2.4: DHCP Service, PC1



Laptop1

FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0030.A37D.EEDD

IP Configuration

● DHCP

○ Static

IPv4 Address 20.0.0.2

Subnet Mask 255.0.0.0

IPv6 Configuration

○ Automatic

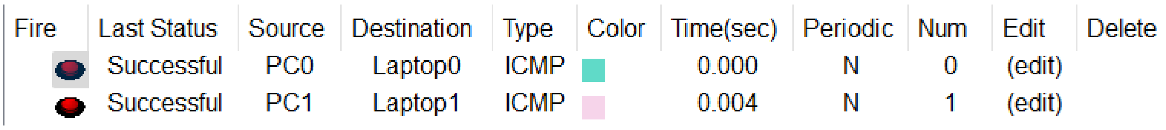
● Static

IPv6 Address /

Link Local Address: FE80::230:A3FF:FE7D:EEDD

☐ Top

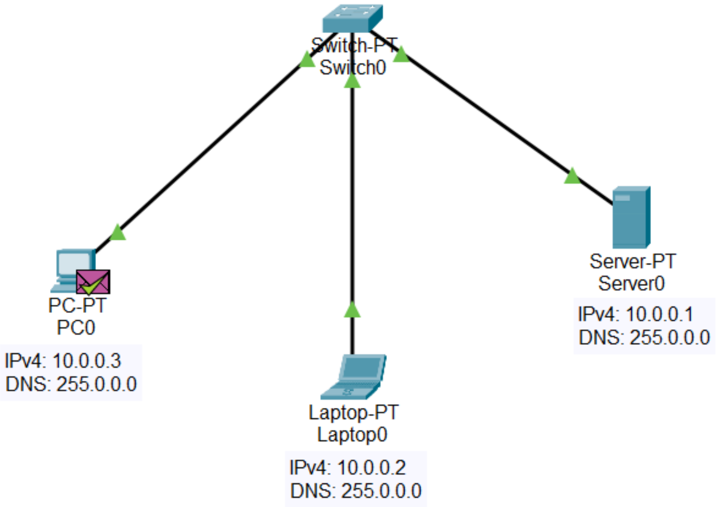
Figure 7.2.5: DHCP Service, Laptop1

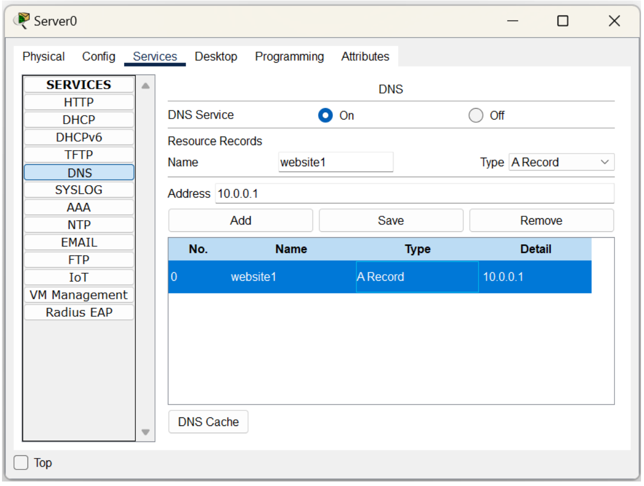


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop0	ICMP	🟩	0.000	N	0	(edit)	
●	Successful	PC1	Laptop1	ICMP	🟪	0.004	N	1	(edit)	

EXPERIMENT – 8

To Configure DNS server to demonstrate the mapping of IP addresses and Domain names.





DNS

DNS Service ● On ○ Off

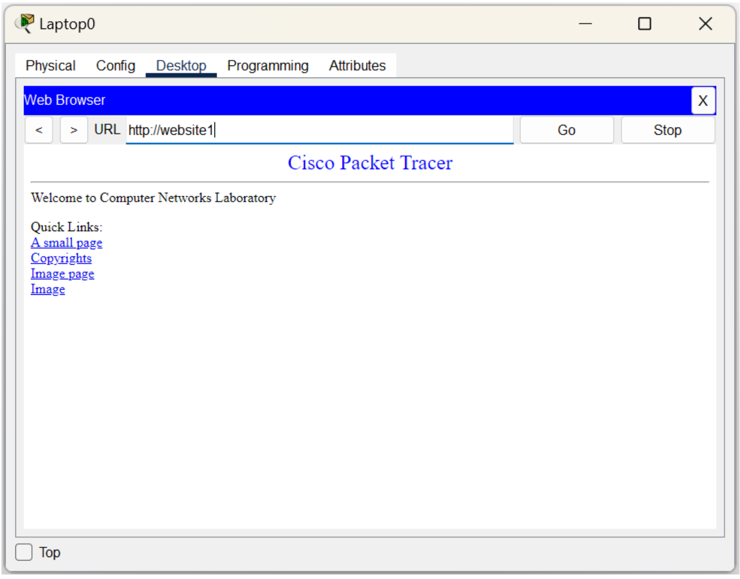
Resource Records

Name website1 Type A Record

Address 10.0.0.1

No.	Name	Type	Detail
0	website1	A Record	10.0.0.1

Figure 8.1: DNS Service, Server0



http://website1

Cisco Packet Tracer

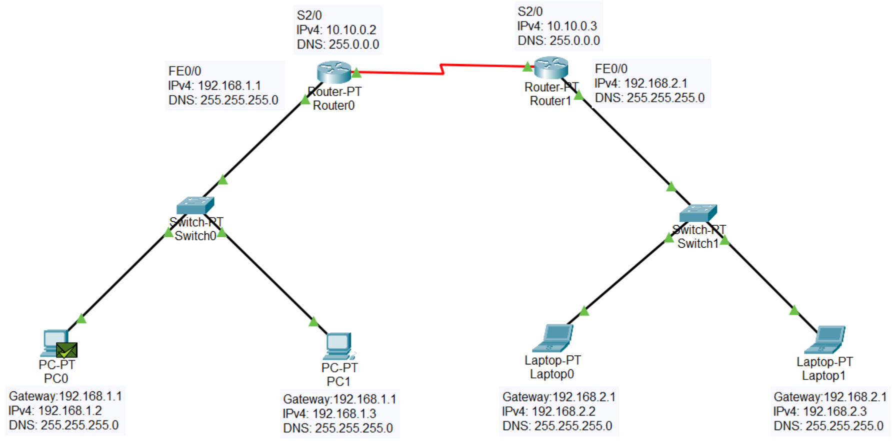
Welcome to Computer Networks Laboratory

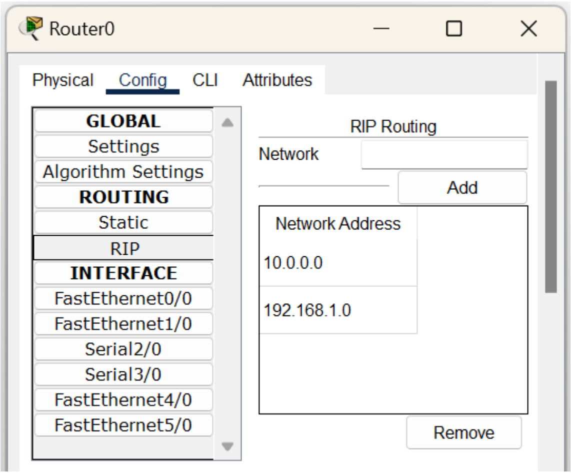
Quick Links:
A small page
Copyrights
Image page
Image

Figure 8.2: DNS Service, Laptop0

EXPERIMENT – 9

To Configure RIP routing protocol in Routers.





RIP Routing

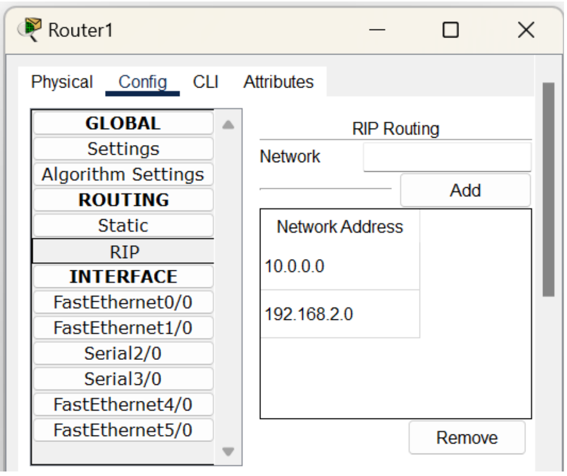
Network

Network Address

10.0.0.0

192.168.1.0

Figure 9.1: RIP, Router0



RIP Routing

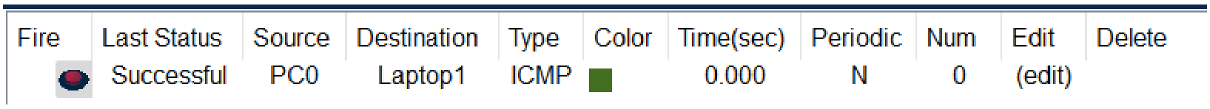
Network

Network Address

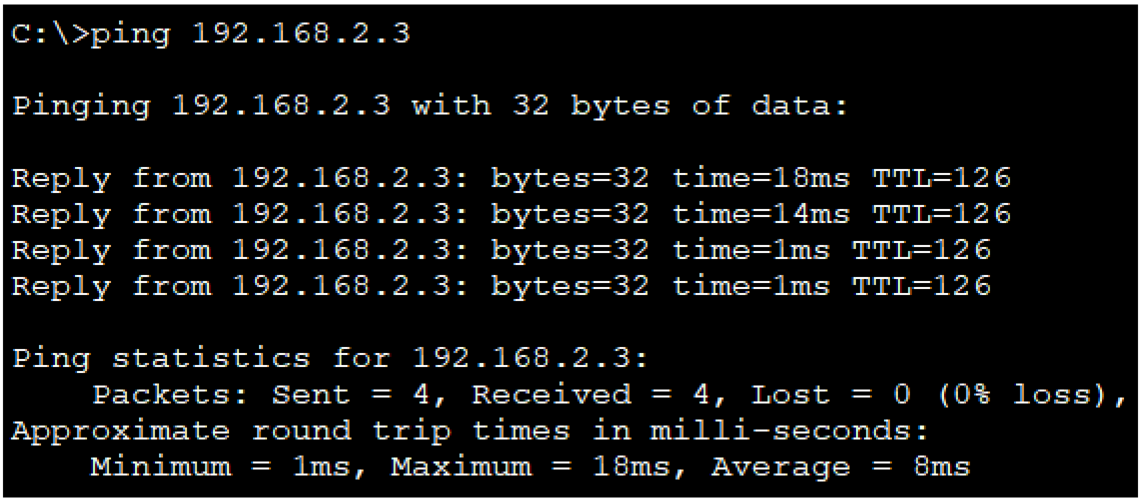
10.0.0.0

192.168.2.0

Figure 9.2: RIP, Router



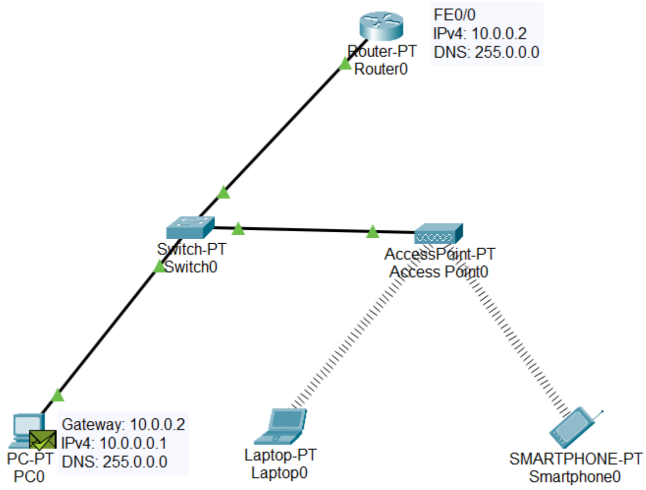
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop1	ICMP	🟩	0.000	N	0	(edit)	

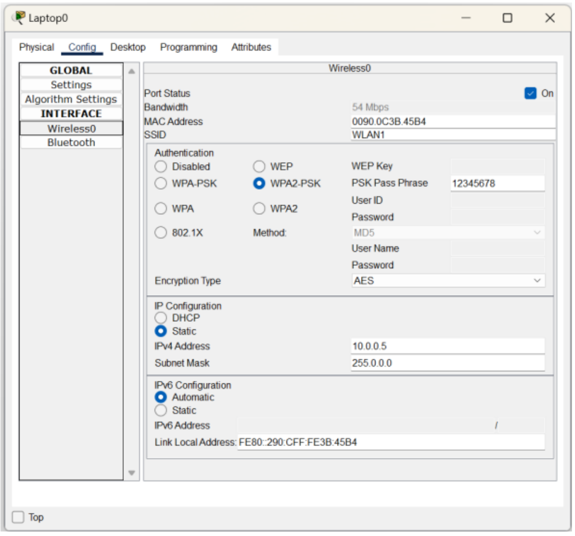


```
C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=18ms TTL=126
Reply from 192.168.2.3: bytes=32 time=14ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 18ms, Average = 8ms
```

EXPERIMENT – 10

To demonstrate communication between two devices using a wireless LAN.





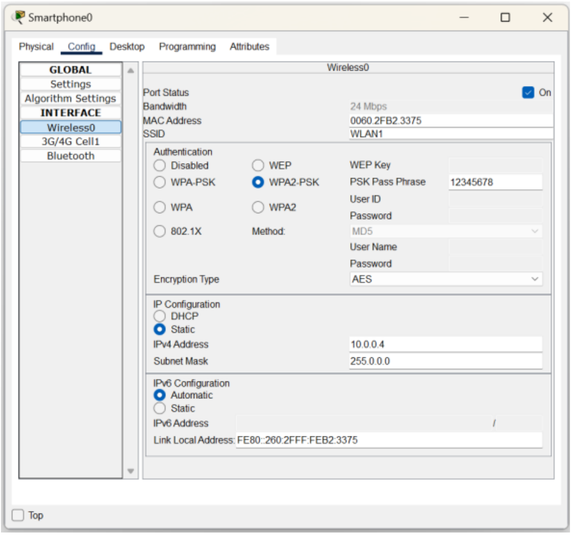
Port Status ☑ On
Bandwidth 54 Mbps
MAC Address 0090.0C3B.45B4
SSID WLAN1

Authentication
○ Disabled ○ WEP WEP Key
○ WPA-PSK ● WPA2-PSK PSK Pass Phrase 12345678
User ID
○ WPA ○ WPA2 Password
○ 802.1X Method: MD5
User Name
Password
Encryption Type AES

IP Configuration
○ DHCP
● Static
IPv4 Address 10.0.0.5
Subnet Mask 255.0.0.0

IPv6 Configuration
● Automatic
○ Static
IPv6 Address /
Link Local Address: FE80::290:CFF:FE3B:45B4

Figure 10.1: Laptop0, Wireless0



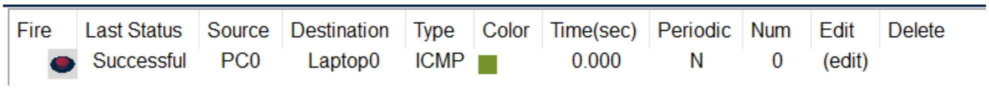
Port Status ☑ On
Bandwidth 24 Mbps
MAC Address 0060.2FB2.3375
SSID WLAN1

Authentication
○ Disabled ○ WEP WEP Key
○ WPA-PSK ● WPA2-PSK PSK Pass Phrase 12345678
User ID
○ WPA ○ WPA2 Password
○ 802.1X Method: MD5
User Name
Password
Encryption Type AES

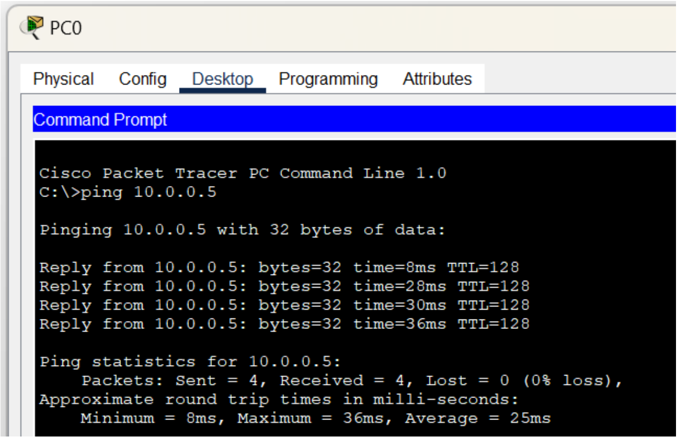
IP Configuration
○ DHCP
● Static
IPv4 Address 10.0.0.4
Subnet Mask 255.0.0.0

IPv6 Configuration
● Automatic
○ Static
IPv6 Address /
Link Local Address: FE80::260:2FFF:FEB2:3375

Figure 10.2: Smartphone0, Wireless0



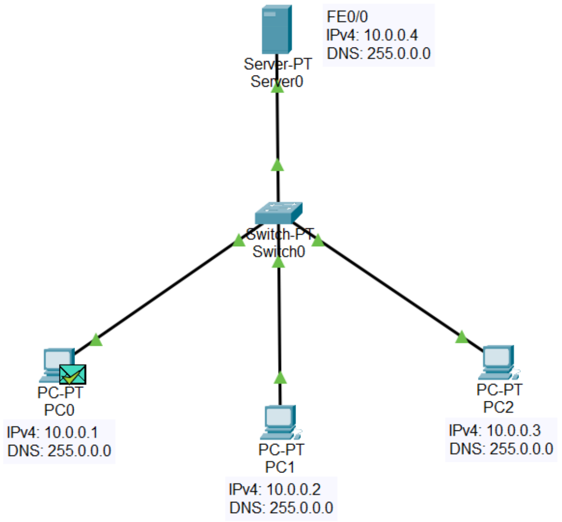
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Laptop0	ICMP	🟩	0.000	N	0	(edit)	

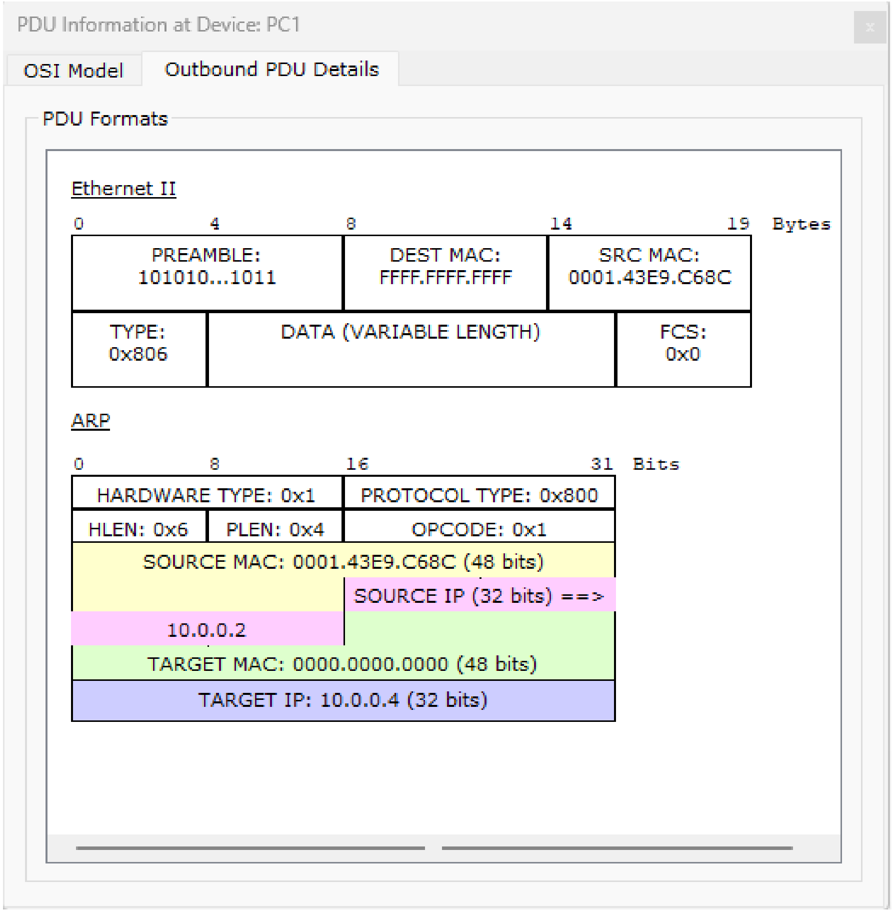


```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5: bytes=32 time=8ms TTL=128
Reply from 10.0.0.5: bytes=32 time=28ms TTL=128
Reply from 10.0.0.5: bytes=32 time=30ms TTL=128
Reply from 10.0.0.5: bytes=32 time=36ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 36ms, Average = 25ms
```

EXPERIMENT – 11

To demonstrate the working of Address Resolution Protocol (ARP) within a LAN for communication.





PDU Formats

Ethernet II

0 4 8 14 19 Bytes

PREAMBLE: 101010...1011	DEST MAC: FFFF.FFFF.FFFF	SRC MAC: 0001.43E9.C68C

TYPE: 0x806	DATA (VARIABLE LENGTH)	FCS: 0x0

ARP

0 8 16 31 Bits

HARDWARE TYPE: 0x1 | PROTOCOL TYPE: 0x800

HLEN: 0x6 | PLEN: 0x4 | OPCODE: 0x1

SOURCE MAC: 0001.43E9.C68C (48 bits)

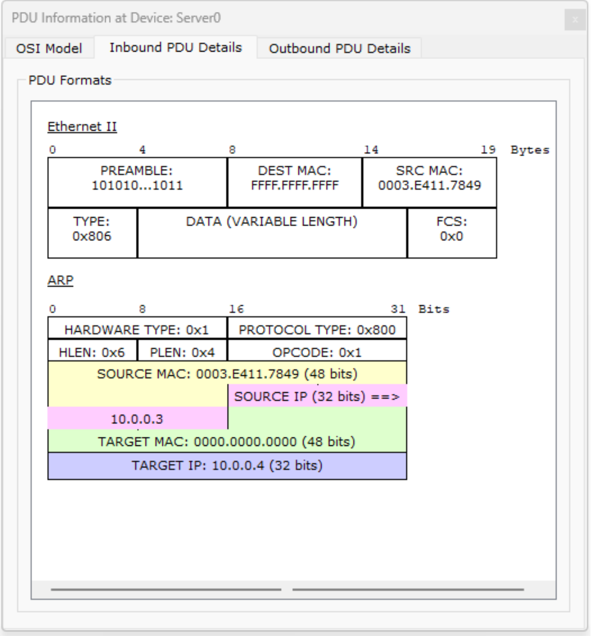
SOURCE IP (32 bits) ==>

10.0.0.2

TARGET MAC: 0000.0000.0000 (48 bits)

TARGET IP: 10.0.0.4 (32 bits)

Figure 11.1: Inbound ARP, PC1



PDU Information at Device: Server0

PDU Formats

Ethernet II

0 4 8 14 19 Bytes

PREAMBLE: 101010...1011 | DEST MAC: FFFF.FFFF.FFFF | SRC MAC: 0003.E411.7849

TYPE: 0x806 | DATA (VARIABLE LENGTH) | FCS: 0x0

ARP

0 8 16 31 Bits

HARDWARE TYPE: 0x1 | PROTOCOL TYPE: 0x800

HLEN: 0x6 | PLEN: 0x4 | OPCODE: 0x1

SOURCE MAC: 0003.E411.7849 (48 bits)

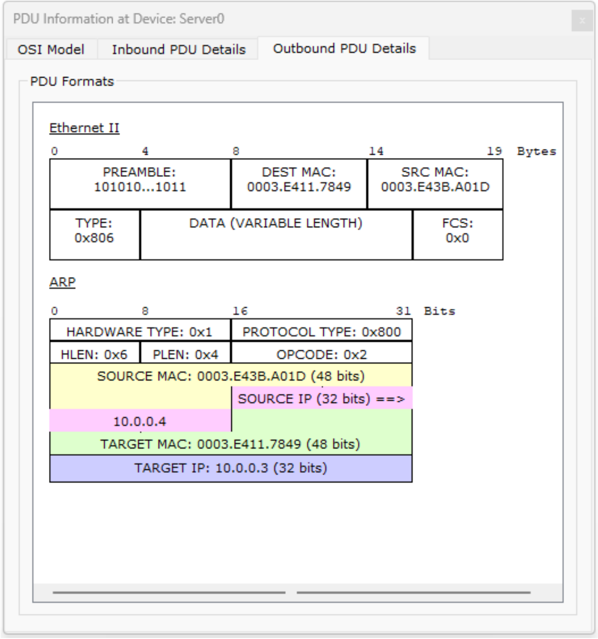
SOURCE IP (32 bits) ==>

10.0.0.3

TARGET MAC: 0000.0000.0000 (48 bits)

TARGET IP: 10.0.0.4 (32 bits)

Figure 11.2: Inbound ARP, Server0



PDU Information at Device: Server0

PDU Formats

Ethernet II

0 4 8 14 19 Bytes

PREAMBLE: 101010...1011 | DEST MAC: 0003.E411.7849 | SRC MAC: 0003.E43B.A01D

TYPE: 0x806 | DATA (VARIABLE LENGTH) | FCS: 0x0

ARP

0 8 16 31 Bits

HARDWARE TYPE: 0x1 | PROTOCOL TYPE: 0x800

HLEN: 0x6 | PLEN: 0x4 | OPCODE: 0x2

SOURCE MAC: 0003.E43B.A01D (48 bits)

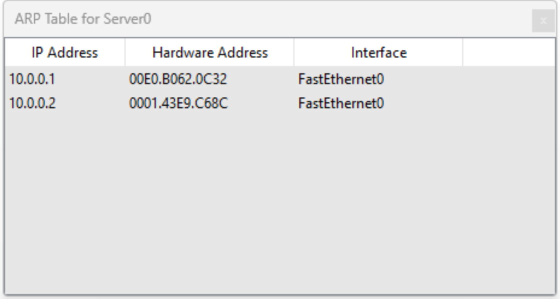
SOURCE IP (32 bits) ==>

10.0.0.4

TARGET MAC: 0003.E411.7849 (48 bits)

TARGET IP: 10.0.0.3 (32 bits)

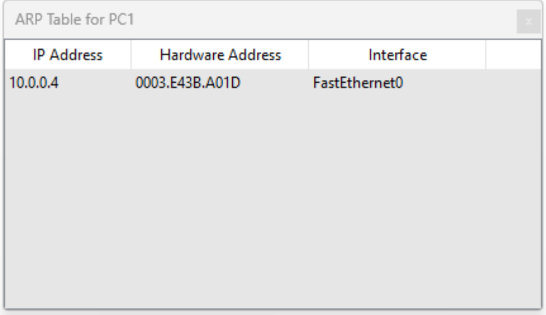
Figure 11.3: Outbound ARP, Server0



ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.1	00E0.B062.0C32	FastEthernet0
10.0.0.2	0001.43E9.C68C	FastEthernet0

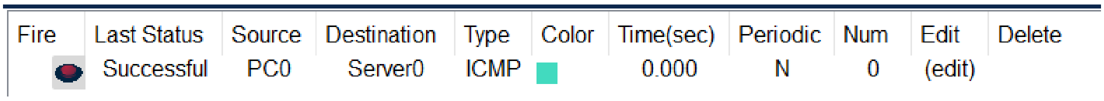
Figure 11.4: ARP Table, Server0



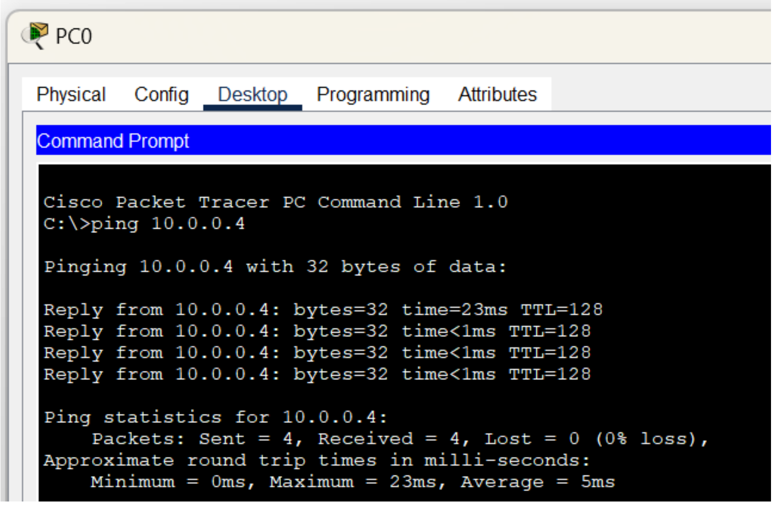
ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.4	0003.E43B.A01D	FastEthernet0

Figure 11.5: ARP Table, PC1



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Server0	ICMP	🟩	0.000	N	0	(edit)	



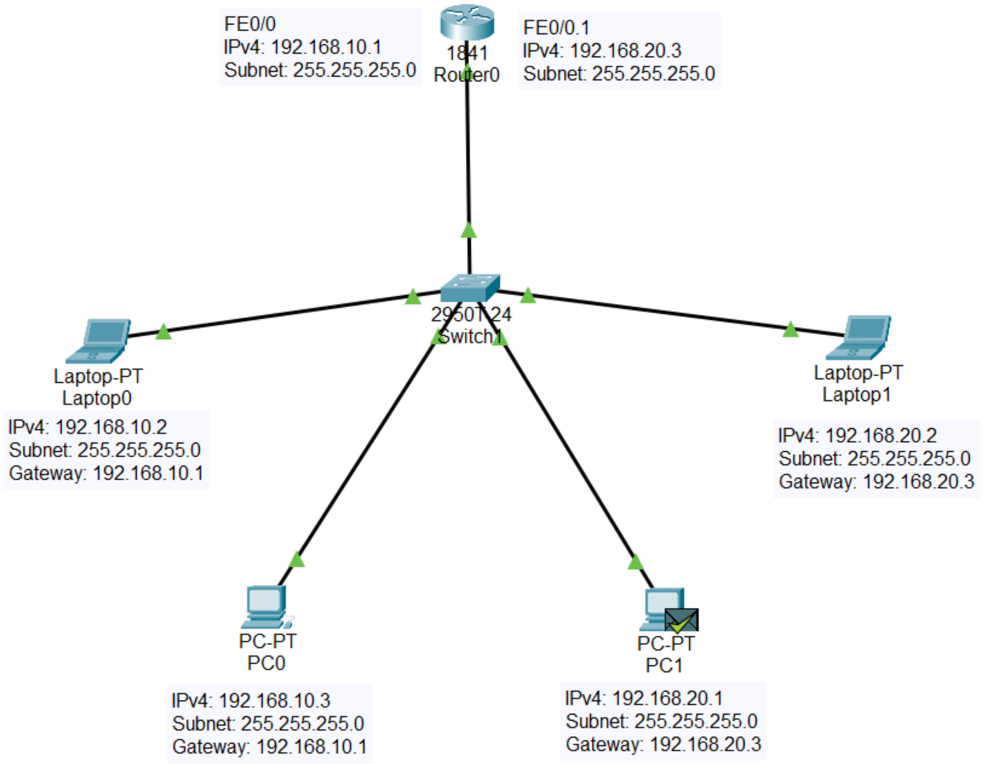
PC0

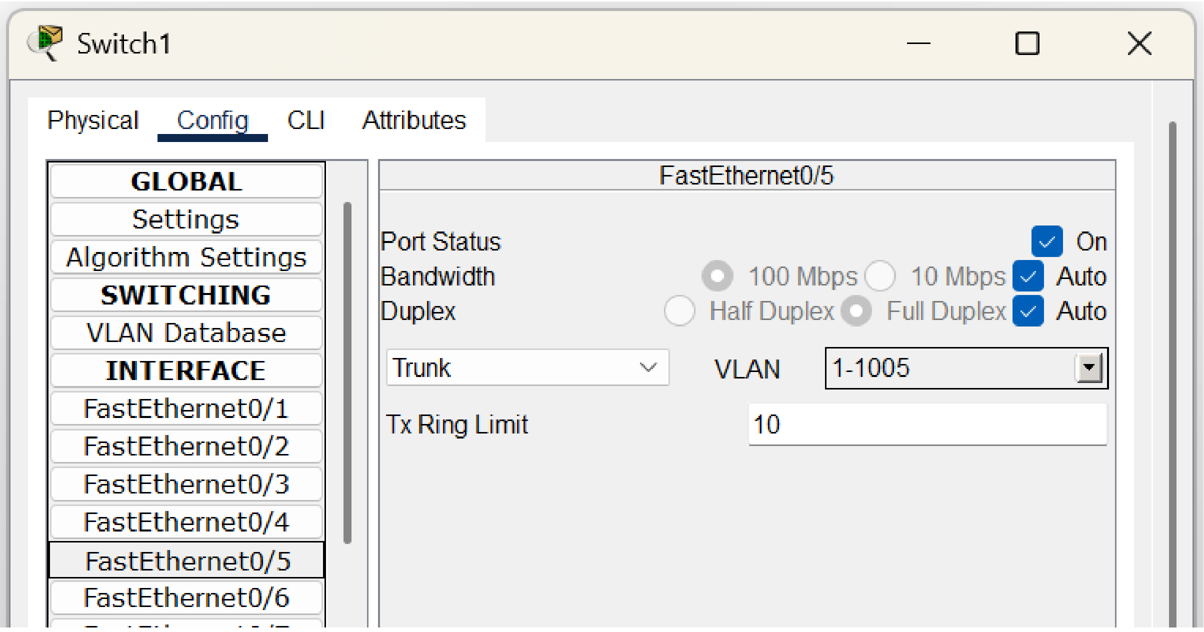
Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=23ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 23ms, Average = 5ms
```

EXPERIMENT – 12

To create a VLAN on top of the physical LAN and enable communication between physical LAN and virtual LAN.





FastEthernet0/5

Port Status ☑ On

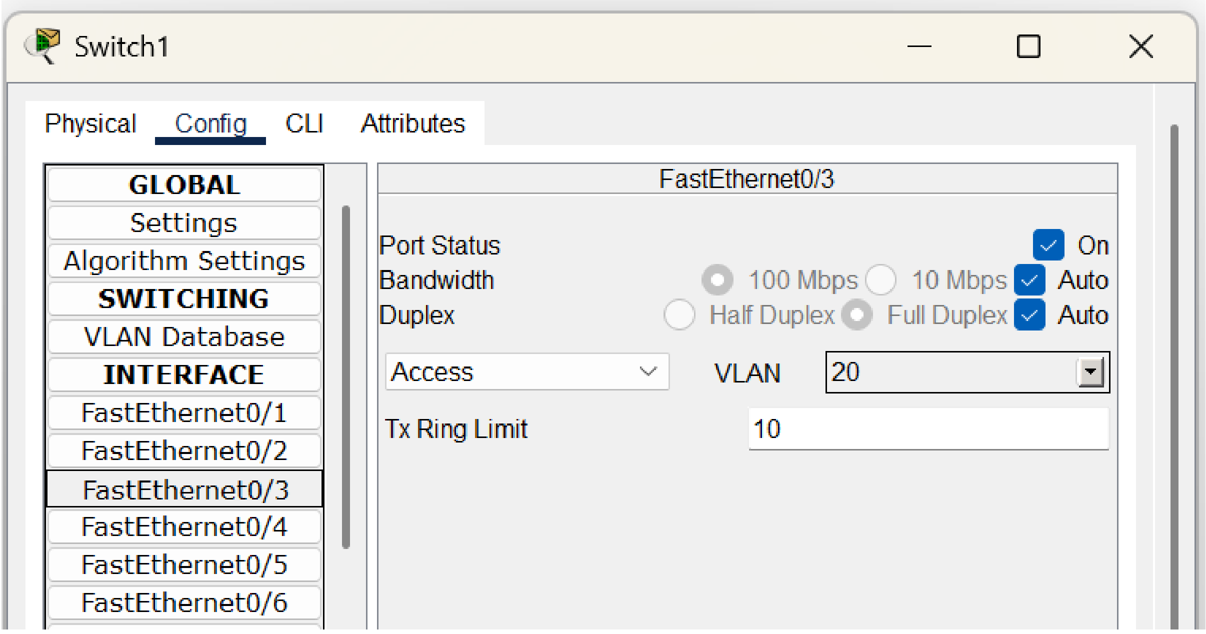
Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

Trunk VLAN 1-1005

Tx Ring Limit 10

Figure 12.1: FE0/5 Switchport Trunk



FastEthernet0/3

Port Status ☑ On

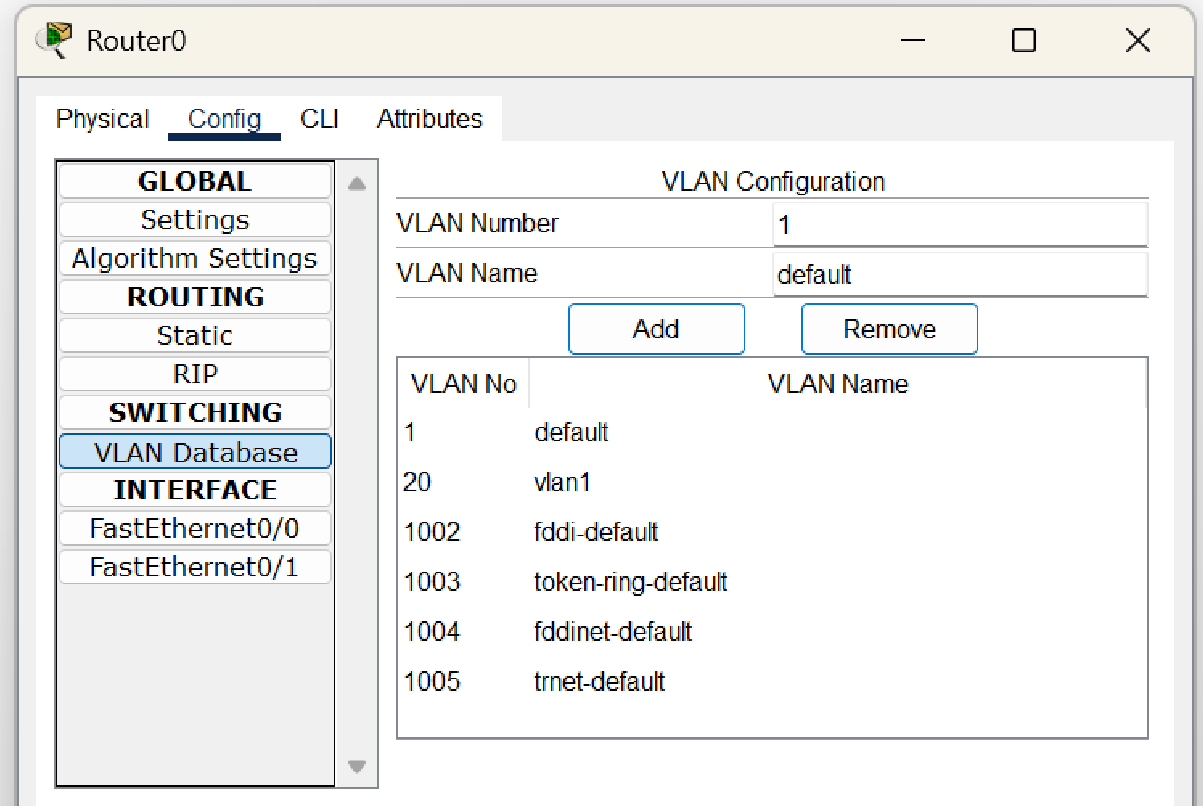
Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

Access VLAN 20

Tx Ring Limit 10

Figure 12.2: FE0/3 Switchport Access



VLAN Configuration

VLAN Number 1

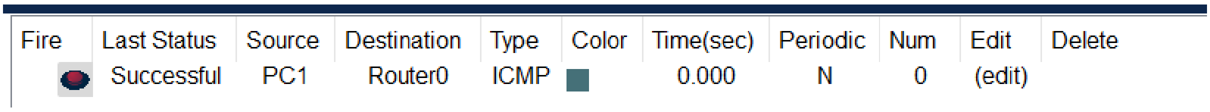
VLAN Name default

VLAN No	VLAN Name
1	default
20	vlan1
1002	fddi-default
1003	token-ring-default
1004	fddinet-default
1005	trnet-default

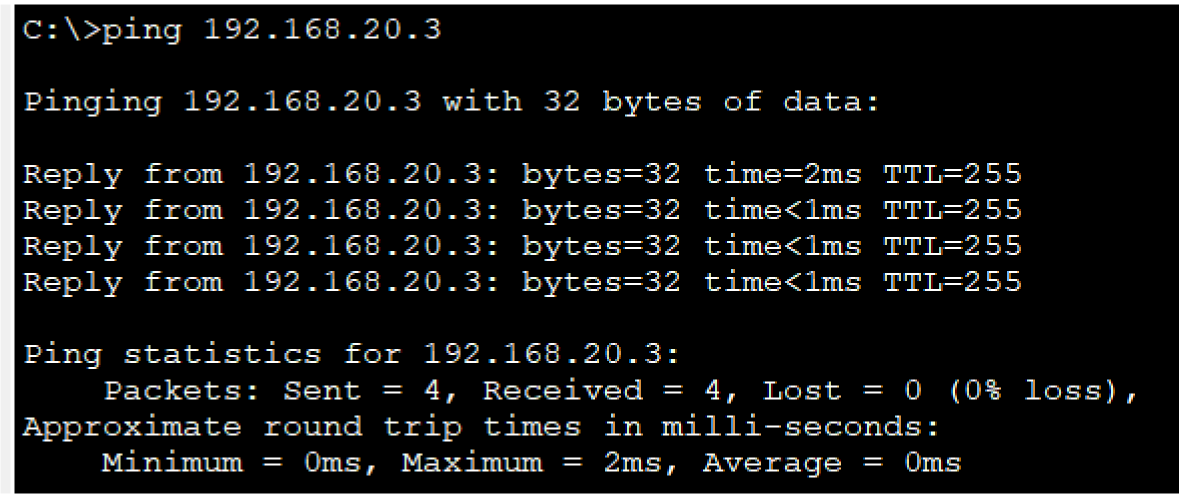
Figure 12.3: Router0 VLAN Database

```
Router(config)#interface FastEthernet0/0.1
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.20.3 255.255.255.0
Router(config-subif)#no shutdown
```

Figure 2: Router0, FE0/0.1



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC1	Router0	ICMP	■	0.000	N	0	(edit)	



```
C:\>ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Reply from 192.168.20.3: bytes=32 time=2ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255
Reply from 192.168.20.3: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

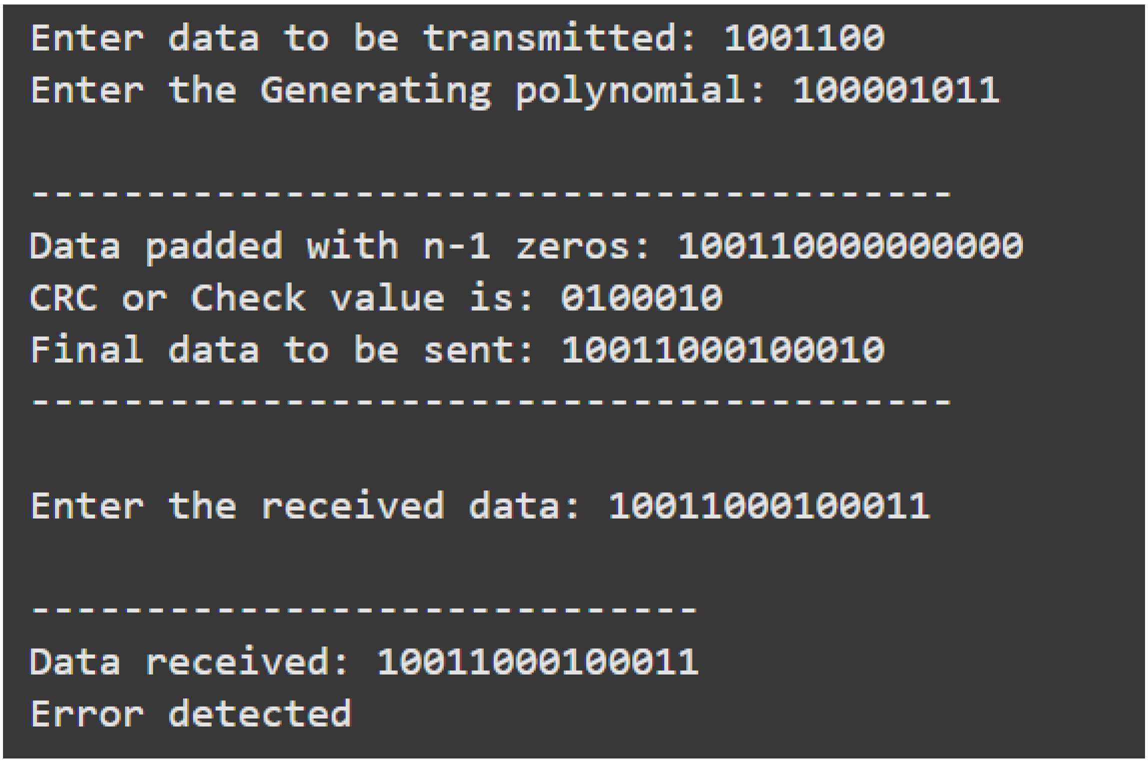
EXPERIMENT – 13

Write a program for error detecting code using CRC-CCITT (8-bits).

Code

```
def xor(dividend, divisor):
    """Perform XOR operation between dividend and divisor."""
    result = ''
    for i in range(1, len(divisor)):
        result += '0' if dividend[i] == divisor[i] else '1'
    return result


def crc(data, gen_poly):
    """Compute the CRC check value using CRC-CCITT (8-bit)."""
    data_length = len(data)
    gen_length = len(gen_poly)

    # Append n-1 zeros to the data
    padded_data = data + '0' * (gen_length - 1)
    check_value = padded_data[:gen_length]

    for i in range(data_length):
        if check_value[0] == '1':
            # XOR operation if the first bit is 1
            check_value = xor(check_value, gen_poly)
        else:
            # Retain original check value if first bit is 0
            check_value = check_value[1:]
        # Shift left and add the next data bit
        if i + gen_length < len(padded_data):
            check_value += padded_data[i + gen_length]

    return check_value[1:]  # Remove the leading bit


def receiver(data, gen_poly):
    """Simulate the receiver side to check for errors."""
    print("\n----------------------------")
    print("Data received:", data)

    # Perform CRC computation on received data
    remainder = crc(data, gen_poly)

    # Check if the remainder is all zeros
    if '1' in remainder:
        print("Error detected")
    else:
        print("No error detected")


if __name__ == "__main__":
    # Input data and generator polynomial
    data = input("Enter data to be transmitted: ")
    gen_poly = input("Enter the Generating
polynomial: ")


    # Compute CRC check value
    check_value = crc(data, gen_poly)
    print("\n--------------------------------------
--")
    print("Data padded with n-1 zeros:",
data + '0' * (len(gen_poly) - 1))
    print("CRC or Check value is:",
check_value)


    # Append check value to data for
transmission
    transmitted_data = data + check_value
    print("Final data to be sent:",
transmitted_data)
    print("---------------------------------------
\n")


    received_data = input("Enter the     #
Simulate the receiver side
received data: ")
    receiver(received_data, gen_poly)
```

Output



```
Enter data to be transmitted: 1001100
Enter the Generating polynomial: 100001011

----------------------------------------
Data padded with n-1 zeros: 1001100000000000
CRC or Check value is: 0100010
Final data to be sent: 10011000100010
----------------------------------------

Enter the received data: 10011000100011

-----------------------------
Data received: 10011000100011
Error detected
```

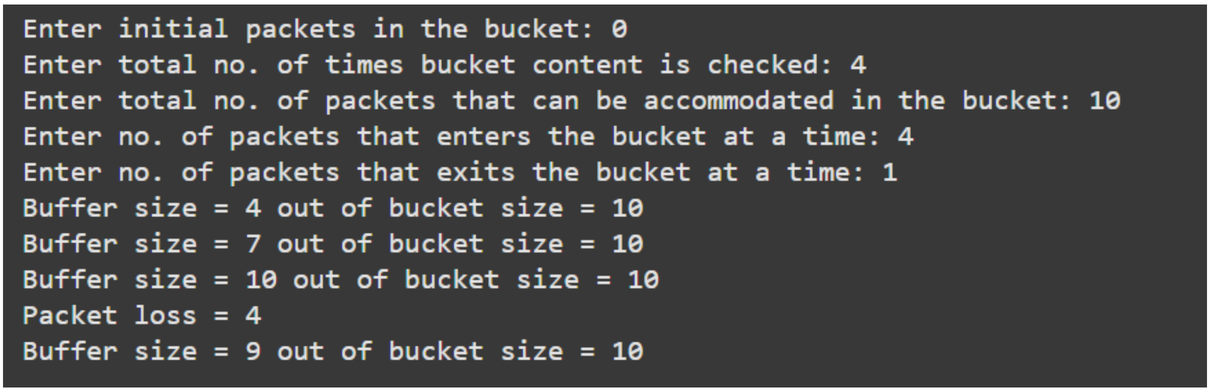
EXPERIMENT – 14

Write a program for congestion control using Leaky bucket algorithm.

Code

```
# Getting user inputs
storage = int(input("Enter initial packets in the bucket: "))
no_of_queries = int(input("Enter total no. of times bucket content is checked: "))
bucket_size = int(input("Enter total no. of packets that can be accommodated in the bucket: "))
input_pkt_size = int(input("Enter no. of packets that enters the bucket at a time: "))
output_pkt_size = int(input("Enter no. of packets that exits the bucket at a time: "))


for i in range(no_of_queries):  # space left
    size_left = bucket_size - storage
    if input_pkt_size <= size_left:
        # update storage
        storage += input_pkt_size
    else:
        print("Packet loss =", input_pkt_size)

    print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

    # as packets are sent out into the network, the size of the storage decreases
    storage -= output_pkt_size
```

Output



```
Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

EXPERIMENT – 15

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.

Code: Client.py

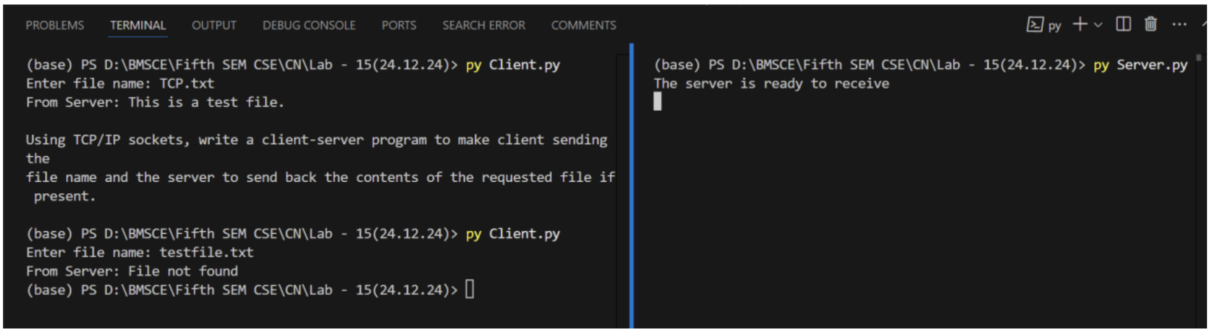
```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost)
serverPort = 12000  # Port number where the server listens

# Create TCP socket
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))  # Connect to server

# Ask user for file name to request
sentence = input("Enter file name: ")

# Send file name to server
clientSocket.send(sentence.encode())

# Receive file contents from server
filecontents = clientSocket.recv(1024).decode()
print('From Server:', filecontents)

# Close the connection
clientSocket.close()
```

Code: Server.py

```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost)
serverPort = 12000  # Port number to listen on

# Create TCP socket
```

```
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))  # Bind socket to the address and port
serverSocket.listen(1)  # Listen for 1 connection
print("The server is ready to receive")

while True:
    # Accept a connection
    connectionSocket, addr = serverSocket.accept()

    # Receive the file name from the client
    sentence = connectionSocket.recv(1024).decode()

    # Try opening the file
    try:
        file = open(sentence, "r")  # Open file in read mode
        fileContents = file.read(1024)  # Read file content (up to 1024 bytes)
        connectionSocket.send(fileContents.encode())  # Send file contents to client
        file.close()
    except FileNotFoundError:
        # Send error message if file not found
        connectionSocket.send("File not found".encode())

    # Close the connection
    connectionSocket.close()
```

Output



```
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py Client.py
Enter file name: TCP.txt
From Server: This is a test file.

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.

(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py Client.py
Enter file name: testfile.txt
From Server: File not found
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)>
```

```
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py Server.py
The server is ready to receive
```

EXPERIMENT – 16

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.

Code: ClientUDP.py

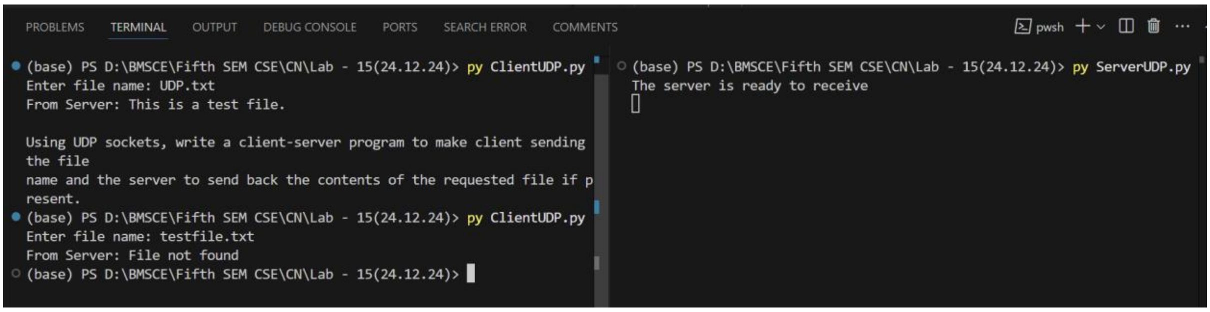
```
from socket import *
serverName = "127.0.0.1"  # Server address (localhost)
serverPort = 12000  # Port number where the server listens

# Create UDP socket
clientSocket = socket(AF_INET, SOCK_DGRAM)

# Ask user for file name to request
sentence = input("Enter file name: ")

# Send the file name to the server using UDP
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))

# Receive file contents from the server
fileContents, serverAddress = clientSocket.recvfrom(2048)

# Print the file contents received from the server
print("From Server:", fileContents.decode())

# Close the UDP socket
clientSocket.close()
```

Code: ServerUDP.py

```
from socket import *
serverPort = 12000  # Port number to listen on

# Create UDP socket
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

```
serverSocket.bind(("127.0.0.1", serverPort))  # Bind the socket to the server address and port

print("The server is ready to receive")

while True:
    # Receive file name from the client
    sentence, clientAddress = serverSocket.recvfrom(2048)

    # Try opening the file
    try:
        file = open(sentence.decode(), "r")  # Open file in read mode
        fileContents = file.read(2048)  # Read file content (up to 2048 bytes)
        serverSocket.sendto(fileContents.encode("utf-8"), clientAddress)  # Send file contents to client
        file.close()
    except FileNotFoundError:
        # Send error message if file not found
        serverSocket.sendto("File not found".encode("utf-8"), clientAddress)
```

Output



```
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py ClientUDP.py
Enter file name: UDP.txt
From Server: This is a test file.

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 p
resent.
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py ClientUDP.py
Enter file name: testfile.txt
From Server: File not found
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)>
```

```
(base) PS D:\BMSCE\Fifth SEM CSE\CN\Lab - 15(24.12.24)> py ServerUDP.py
The server is ready to receive
```