*Oriental Education Society's*

SANPADA COLLEGE OF COMMERCE & TECHNOLOGY

SECTOR-2, PLOT-3/4/5, ADJACENT SANPADA RAILWAY STATION, SANPADA (W),
NAVI MUMBAI- 400 705.

**DEPARTMENT OF INFORMATION TECHNOLOGY**

CERTIFICATE

This is to certify that **Sadan Acharya** of Class SYCS bearing Roll No 53 of Semester IV has successfully completed the Practical work in the subject of **"COMPUTER NETWORKING"** during the academic year **2025-2026** under the guidance of Prof. **TEJASHREE MA'AM** being the partial requirement for the fulfilment of the curriculum of Degree of Bachelor of Science in Information Technology, University of Mumbai.

Place : **SANPADA**
Date :

Sign Of Subject-In-charge

Sign of External Examiner

Sign Of HOD

Sign Of Principle

# INDEX

| Sr.No | NAME | DATE | SIGN |
|---|---|---|---|
| 1 | Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo | | |
| 2 | Building Peer-to-Peer Network. Create a Network with two PCs connected using crossover cable using CISCO packet tracer. Assign static IP addresses. And Test Connectivity. | | |
| 3 | Using Packet Tracer, create a basic network of one server and two computers using fast Ethernet. | | |
| 4 | Design and configure Local Area Network(LAN) for inter-connectivity conputers within a limited are using start topology. | | |
| 5 | Dynamic IP allocation with server & clients. Use DHCP services from a server to assign IPs to two PCs. Enable and configure DHCP on the server. Verify IP allocation and connectivity using | | |

| | | | |
|---|---|---|---|
| | ipconfig. | | |
| 6 | Configure IP routing using RIP. Two routers connnected to two switches & four PCs connected to it. Implement RIP routing between routers. Verify inter-network connectivity using ping and route tables. | | |
| 7 | Creating a mixed network with wired and wireless devices. One server, two wired PCs and two mobiles. Use appropriate cabling and access points, Assign IPs and test cross-devices communication. | | |
| 8 | Three routers and their networks using OSPF. Assign area IDs, router IDs, and enable OSPF. Monitor OSPF neighbor relationships and path selections. | | |
| 9 | Three routers, each connected to at least three PCs. Implement RIPv1 routing between routers Verify inter-network connectivity using ping and route tables. | | |
| 10 | Three Routers, each connected to at least 3 PCs. Enable RIPv2 and observe mask handling. Using Packet Tracer's sim to observe routing updates. | | |

# Practical-1

**Aim**: Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo

1) Ping:

```
❯ ping www.google.com
PING www.google.com (142.251.220.4) 56(84) bytes of data.
64 bytes from pnbomb-ay-in-f4.1e100.net (142.251.220.4): icmp_seq=1 ttl=117 time=77.6 ms
64 bytes from pnbomb-ay-in-f4.1e100.net (142.251.220.4): icmp_seq=2 ttl=117 time=10.6 ms
^C
--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 10.619/44.123/77.627/33.504 ms
```

2) tracert:

```
C:\Users\Matt>tracert 8.8.8.8

Tracing route to google-public-dns-a.google.com [8.8.8.8]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms
  2     4 ms     7 ms     1 ms
  3     1 ms     1 ms     1 ms
  4    24 ms    24 ms    25 ms
  5    24 ms    24 ms    24 ms
  6    25 ms    25 ms    25 ms
  7    25 ms    25 ms    25 ms
  8    25 ms    25 ms    25 ms  google-public-dns-a.google.com [8.8.8.8]

Trace complete.
```

3) netstat:

```
❯ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0                        bom12s03-in-f10.1:https ESTABLISHED
tcp        0      0                        192.28.147.68:https     ESTABLISHED
tcp        0      0                        93.243.107.34.bc.:https ESTABLISHED
tcp        0      0                        sm-in-f119.1e100.:https ESTABLISHED
tcp        0      0                        172.64.41.4:https       ESTABLISHED
tcp        0      0                        sd-in-f119.1e100.:https TIME_WAIT
tcp        0      0                        whatsapp-cdn-shv-:https ESTABLISHED
tcp        0   2963                        bom12s07-in-f14.1:https ESTABLISHED
tcp        0      0                        sd-in-f119.1e100.:https TIME_WAIT
```

4) arp

```
❯ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
gpon.net                 ether   a8:74:84:d0:2f:02   C                     wlp0s20f3
```

5) ifconfig:

```
❯ ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether f4:a8:0d:ef:c6:72  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0x82200000-82220000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
```

6) getmac:

```
C:>getmac

Physical Address    Transport Name
=================== ==========================================================
30-5A-3A-7F-17-A1   \Device\Tcpip_{278F1BFF-B571-444D-B35F-3CF3FBF03B0C}
00-FF-49-B8-4F-89   Media disconnected
N/A                 Hardware not present
Disabled            Disconnected
N/A                 Hardware not present

C:>
```

7) hostname:

```
❯ hostname
Nova
```

8) nslookup:

```
❯ nslookup
> hostname
;; Got SERVFAIL reply from 127.0.0.53
Server:
Address:

** server can't find hostname: SERVFAIL
> exit
```

9) pathping:

```
PS C:\Users\ashle> pathping -q 255 1.1.1.1

Tracing route to one.one.one.one [1.1.1.1]
over a maximum of 30 hops:
  0  Ashley [192.168.50.144]
  1  RT-AX86U-FBC0 [192.168.50.1]
  2  bb121-7-163-254.singnet.com.sg [121.7.163.254]
  3  165.21.193.18
  4  165.21.193.17
  5  165.21.138.181
  6  SN-SINTP1-BO402-ae1.singnet.com.sg [165.21.138.94]
  7  ip-202-147-32-126.asianetcom.net [202.147.32.126]
  8     *     i-92.sgcn-core01.telstraglobal.net [202.84.219.174]
  9  i-92.sgcn-core01.telstraglobal.net [202.84.219.174]
 10  i-91.istt04.telstraglobal.net [202.84.224.197]
 11  unknown.telstraglobal.net [210.57.38.181]
 12  162.158.160.55
 13     *     one.one.one.one [1.1.1.1]

Computing statistics for 828 seconds...
```

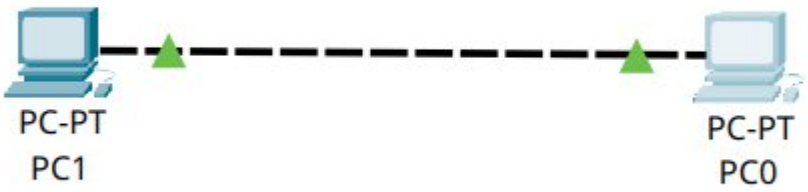
10) systeminfo

```
C:\>systeminfo

Host Name:
OS Name:                   Microsoft Windows 7 Ultimate
OS Version:                6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          Admin
Registered Organization:
```

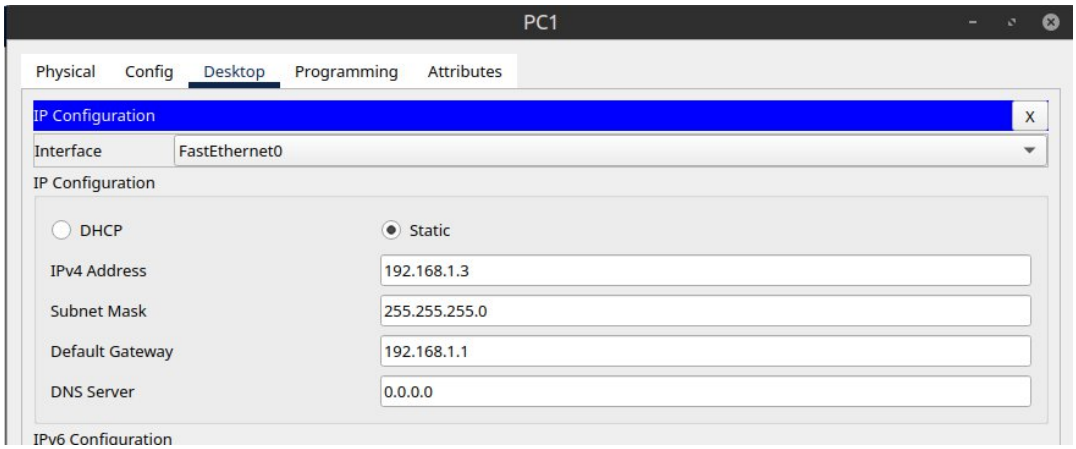
# Practical-2

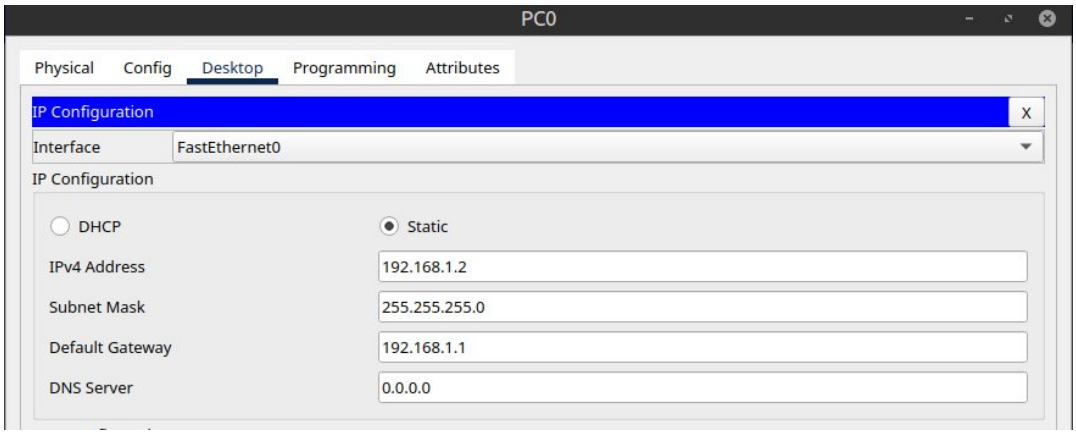
**Aim**: Building Peer-to-Peer Network. Create a Network with two PCs connected using crossover cable using CISCO packet tracer. Assign static IP addresses. And Test Connectivity.

**Implementation:**

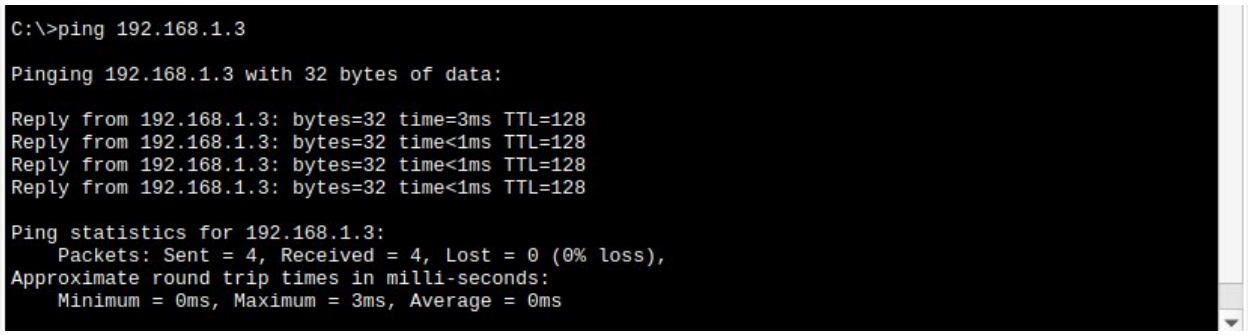



**Now we set the ip address of the devices as follows:**



In order to check the connectivity we send a ping command from PC0 to PC1 as follows:

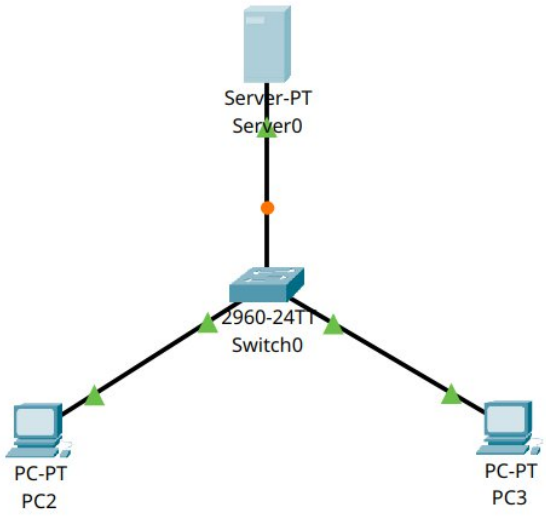



**Conclusion:** Hence the Connectivity between the PCs has been verified.

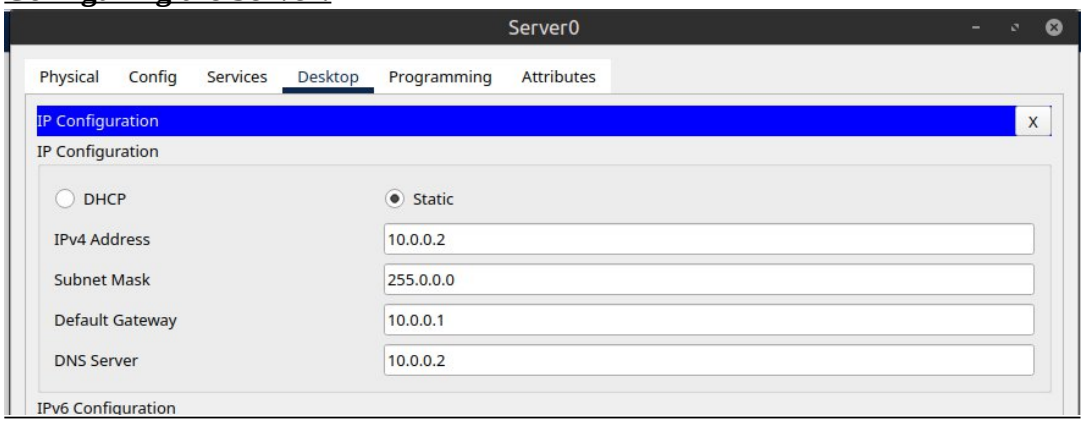
# Practical-3

**Aim**: Using Packet Tracer, create a basic network of one server and two computers using fast Ethernet.

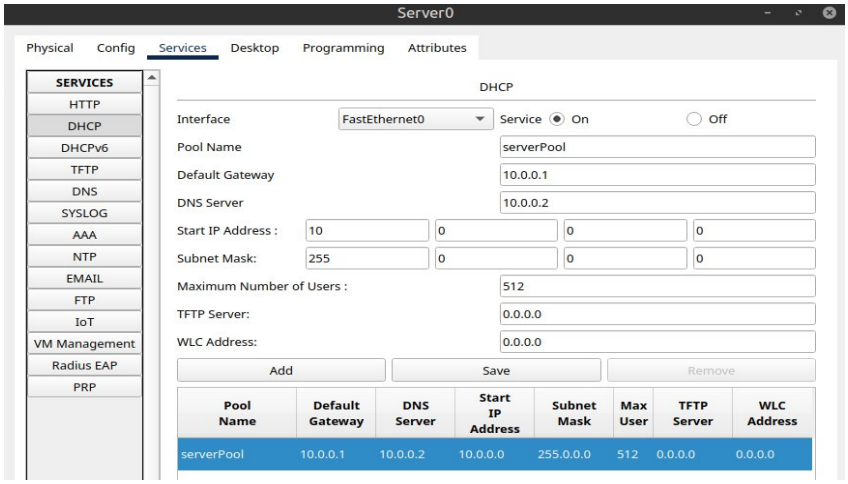
## Topology:

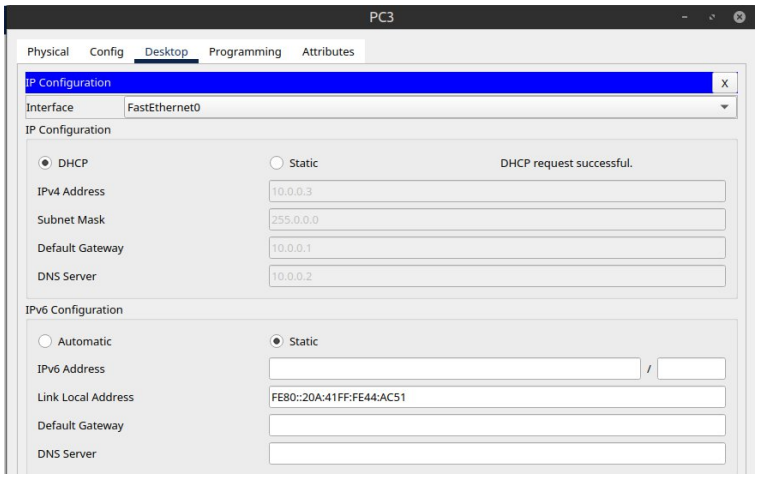


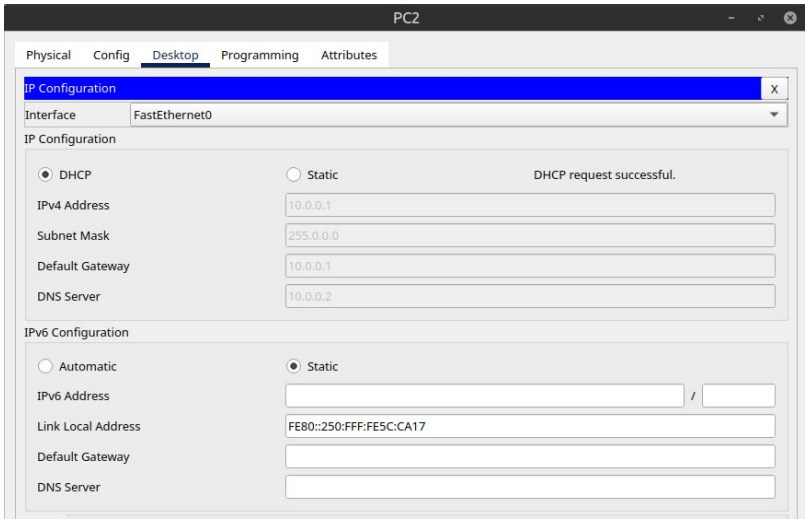

## Configuring the Server:




## Enabling and setting the DHCP Service on the Server:






## Checking for connectivity:

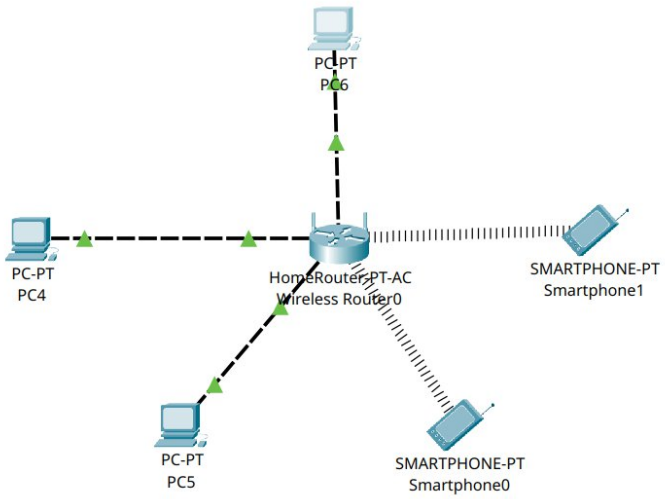
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 6ms, Average = 4ms

Control-C
^C
C:\>
```

**Conclusion:** Hence the Connectivity between the PCs has been verified.

# Practical-4

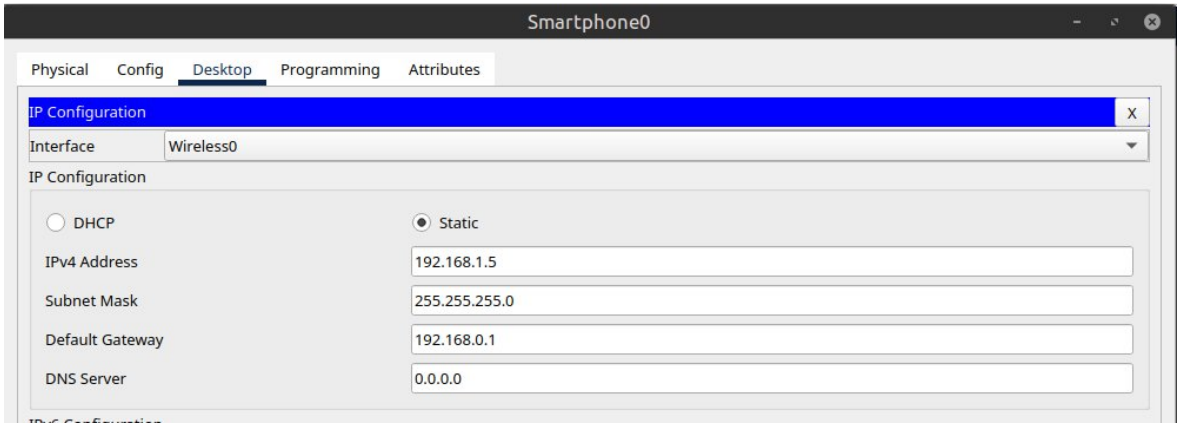
**Aim**: Design and configure Local Area Network(LAN) for inter-connectivity conputers within a limited are using start topology.

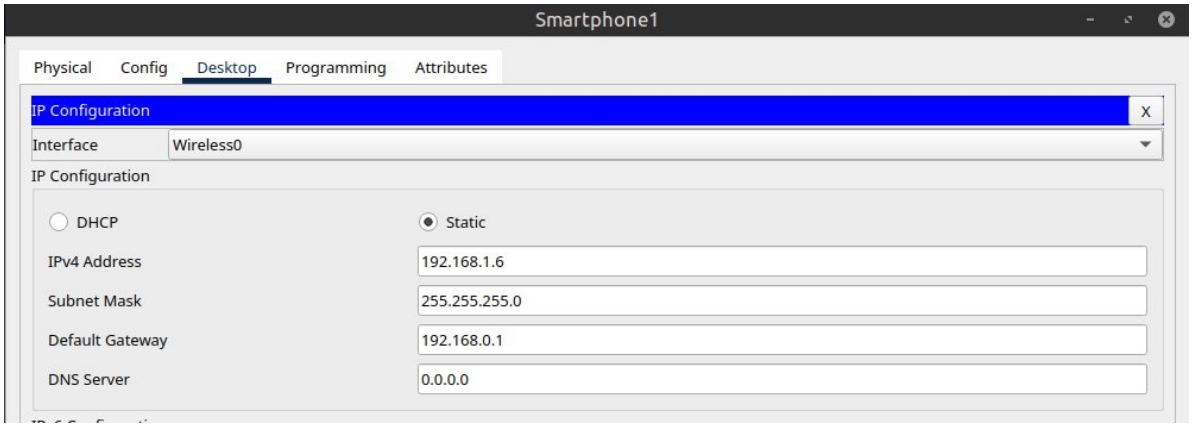
## Topology:



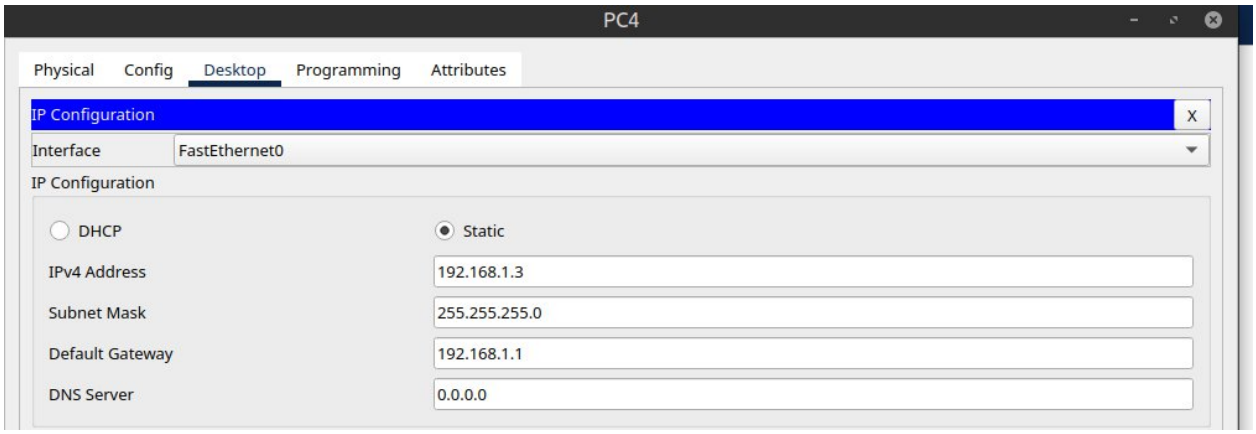

---

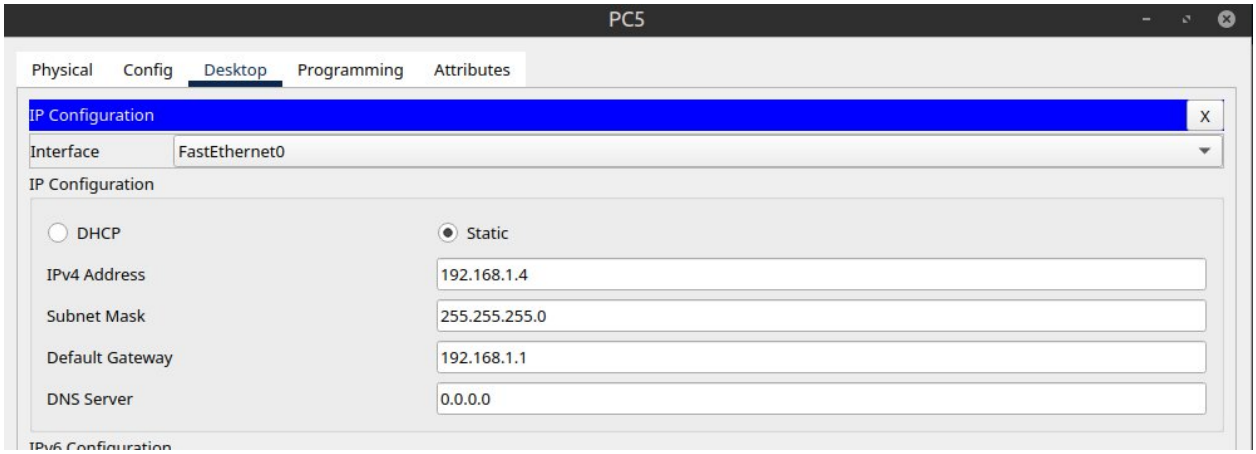
## Smartphone Configuration:



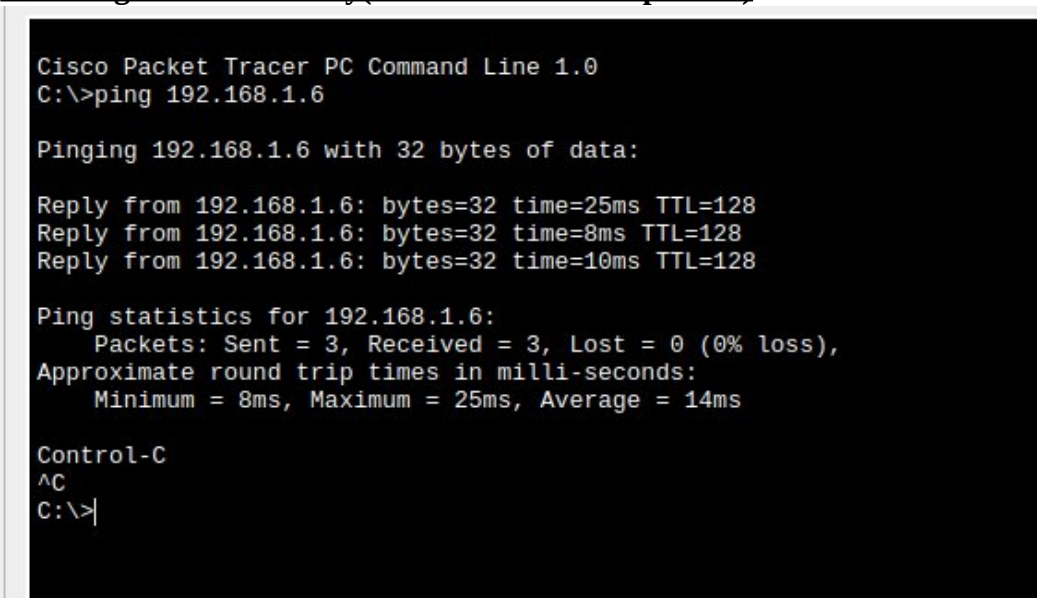



## Computer Configurations:




## Checking for Connectivity(From PC4 to Smartphone1)

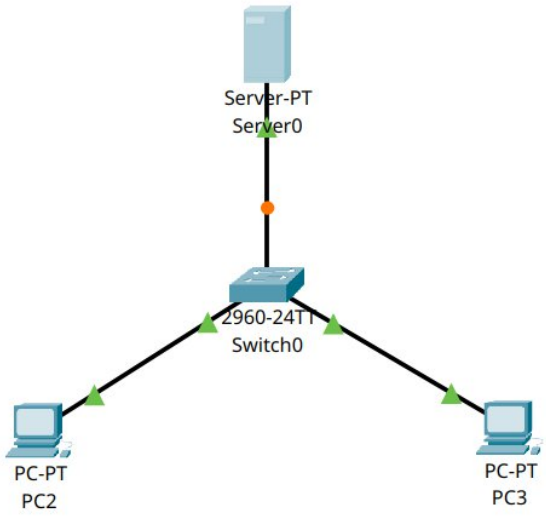



**Conclusion:** Hence the Connectivity of the network has been verified.

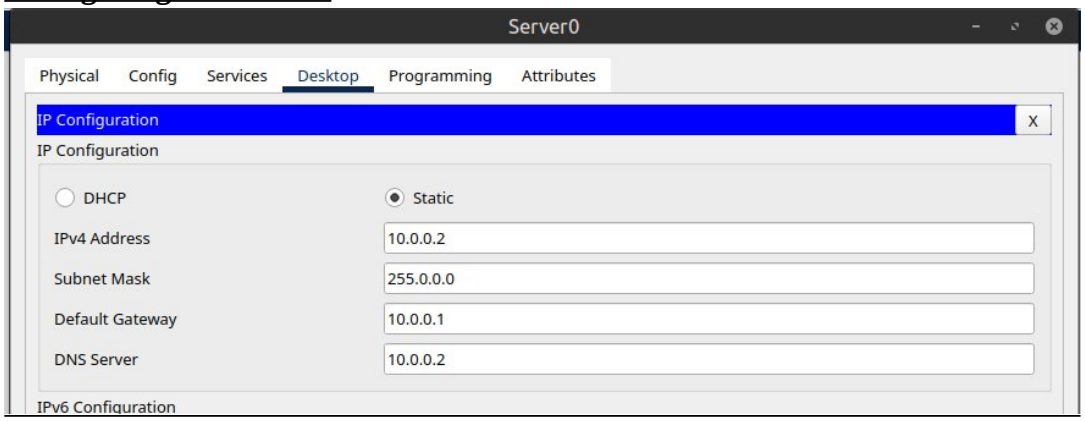
# Practical-5

**Aim:** Dynamic IP allocation with server & clients. Use DHCP services from a server to assign IPs to two PCs. Enable and configure DHCP on the server. Verify IP allocation and connectivity using ipconfig.

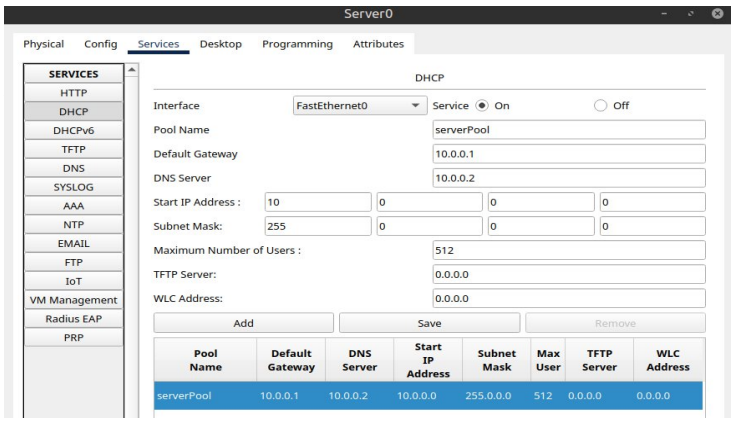
**Topology:**

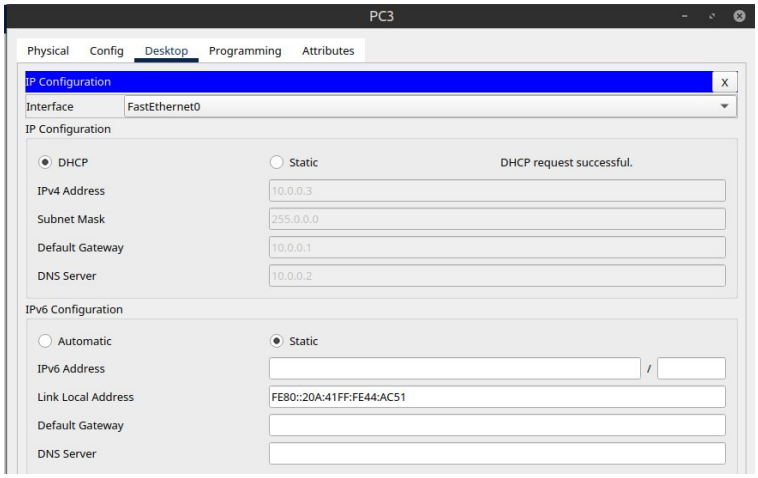


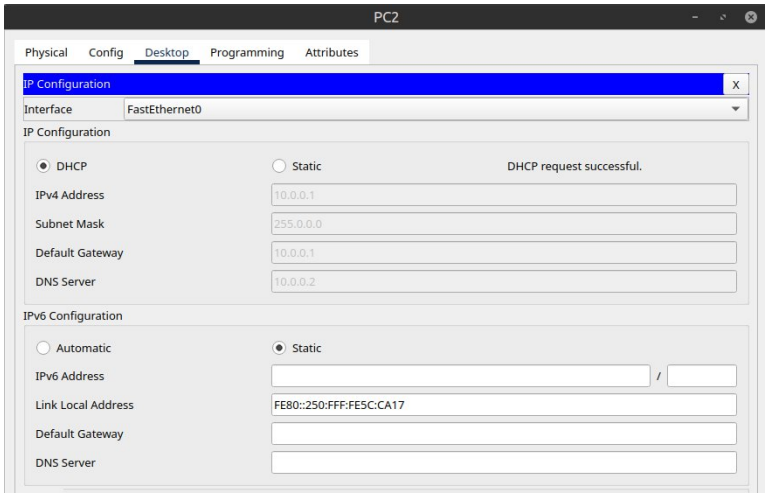

**Configuring the Server:**



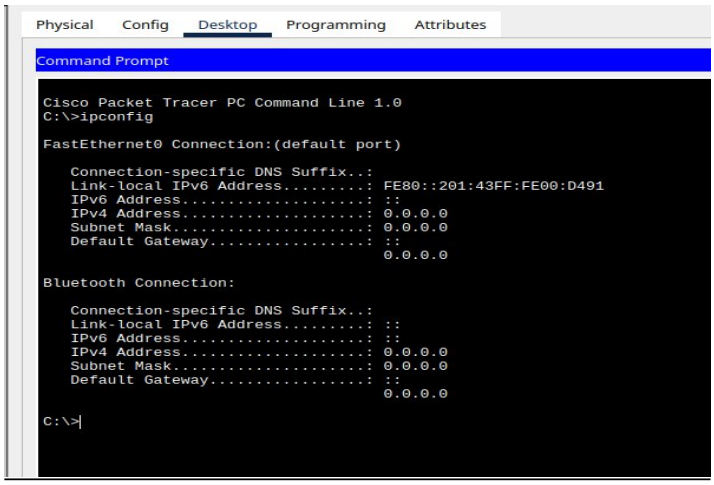

**Enabling and setting the DHCP Service on the Server & Checking Clients:**






## Checking for connectivity:

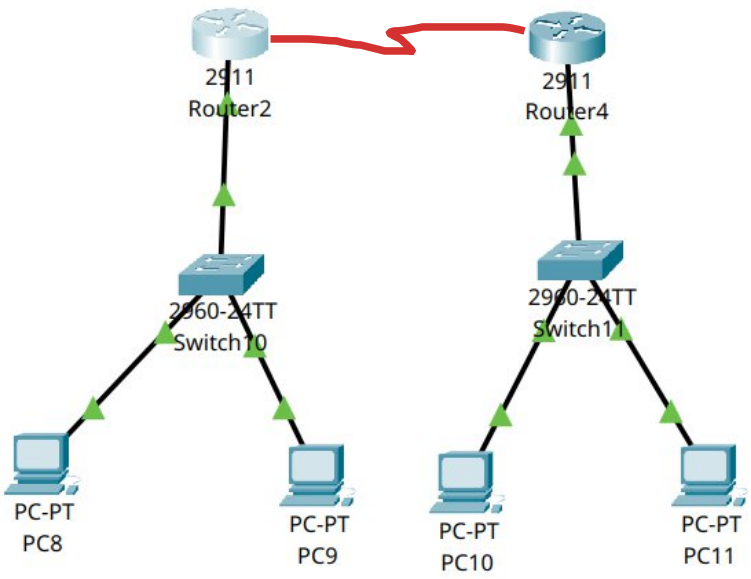



**Conclusion:** Hence the Connectivity between the PCs has been verified Using ipconfig.

# Practical-6

**Aim:** Configure IP routing using RIP. Two routers connnected to two switches & four PCs connected to it. Implement RIP routing between routers. Verify inter-network connectivity using ping and route tables.

**Topology:**




---

**PCs IP Configuration:**

| Device | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|
| **PC0** | 192.168.1.2 | 255.255.255.0 | 192.168.1.1 |
| **PC1** | 192.168.1.3 | 255.255.255.0 | 192.168.1.1 |
| **PC2** | 192.168.2.2 | 255.255.255.0 | 192.168.2.1 |
| **PC3** | 192.168.2.3 | 255.255.255.0 | 192.168.2.1 |

**Configure Router Interfaces (CLI):**

**Router2:**

```
Router>emen
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#nonno shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
int g0/1
Router(config-if)#ip add 10.0.0.1 255.255.255.252
Router(config-if)#no shut
```

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g0/0
Router(config-if)# ip add 192.168.2.1 255.255.255.0
Router(config-if)# no shut

Router(config-if)#int g0/1
Router(config-if)# ip add 10.0.0.2 255.255.255.252
Router(config-if)# no shut
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
```

## Implement RIP Routing:

## Router2:

```
Router(config-if)#router rip
Router(config-router)# version 2
Router(config-router)# no auto-summary
Router(config-router)# network 192.168.1.0
Router(config-router)# network 10.0.0.0
```

## Router4:

```
Router(config-if)#
%LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down
router rip
Router(config-router)# version 2
Router(config-router)# no auto-summary
Router(config-router)# network 192.168.2.0
Router(config-router)# network 10.0.0.0
Router(config-router)#
```

### Checking Routing table:

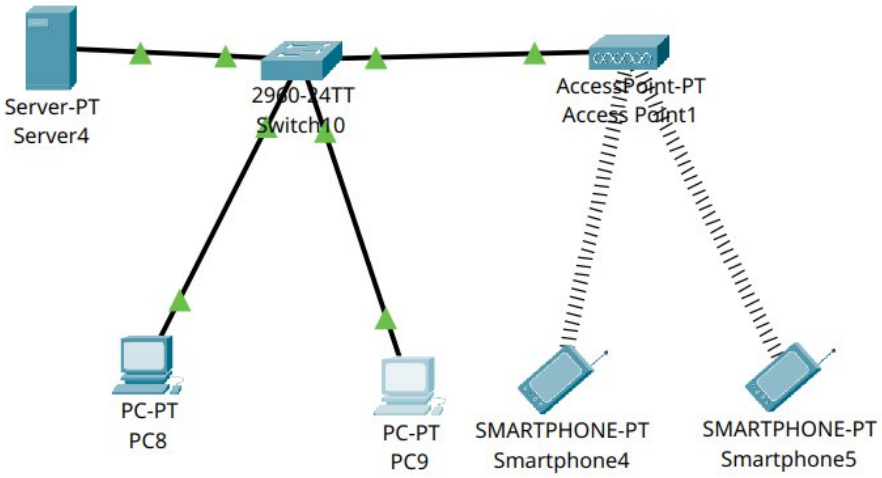
```
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - 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

     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0

Router#




Router con0 is now available
```

**Conclusion:** Hence verified the connection of two routers with RIP.

# Practical-7

**Aim:** Creating a mixed network with wired and wireless devices. One server, two wired PCs and two mobiles. Use appropriate cabling and access points, Assign IPs and test cross-devices communication.

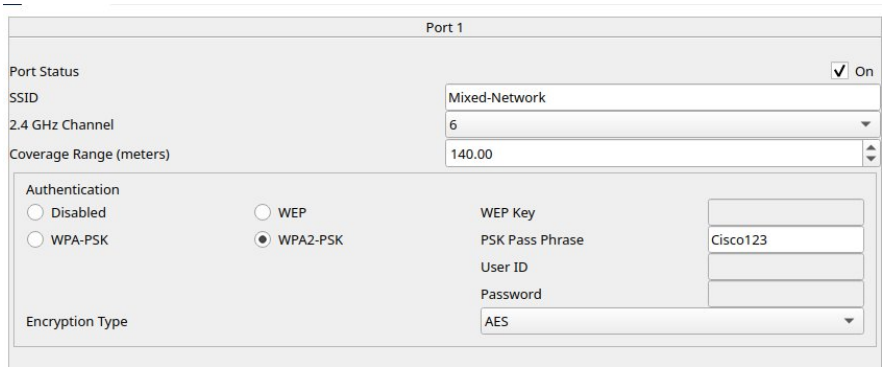
**Topology:**



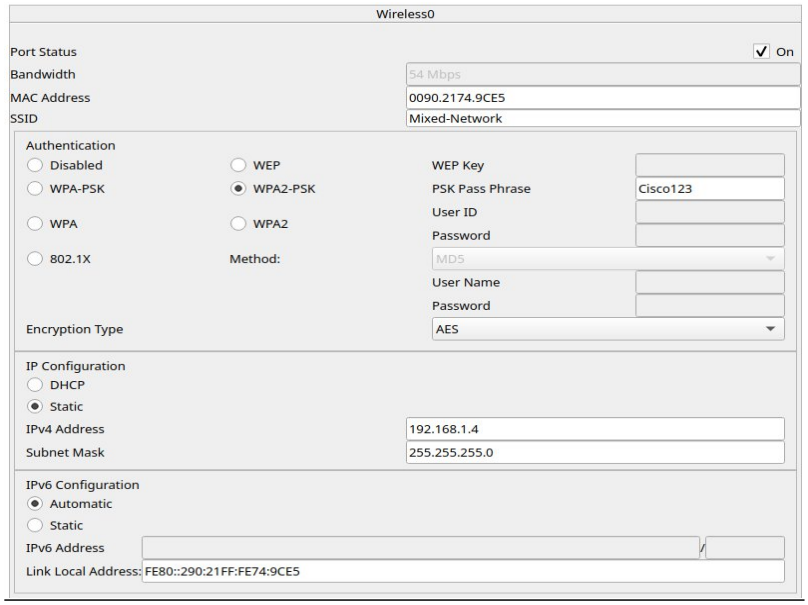

**Addressing Table (Static):**

| Device | IP Address | Subnet Mask |
|---|---|---|
| **Server** | 192.168.1.1 | 255.255.255.0 |
| **PC0** | 192.168.1.2 | 255.255.255.0 |
| **PC1** | 192.168.1.3 | 255.255.255.0 |
| **Smartphone0** | 192.168.1.4 | 255.255.255.0 |
| **Smartphone1** | 192.168.1.5 | 255.255.255.0 |

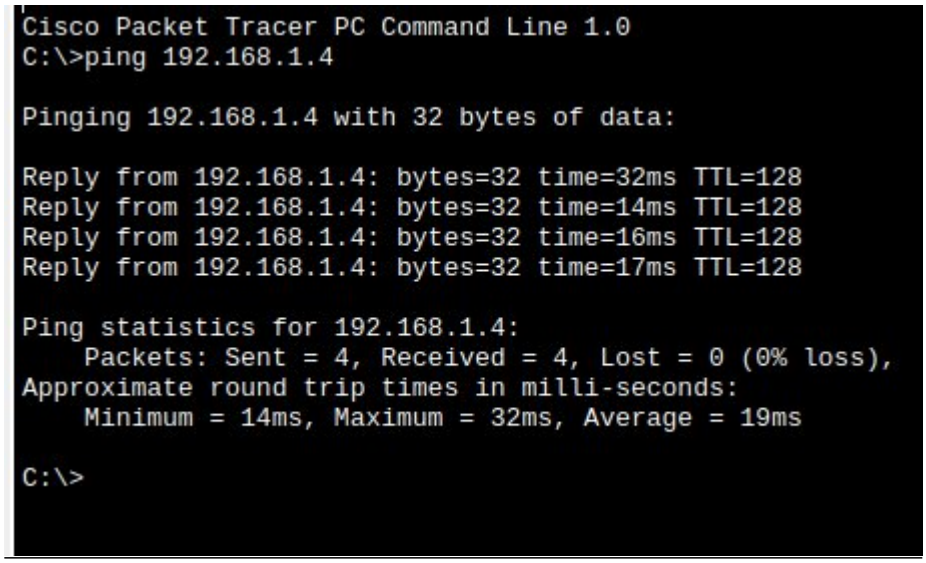
**Configuring Wireless Access point:**



## Configuring Smartphone To Connect to wireless network (Same for both Smartphones):




## Verification (Testing Communication):

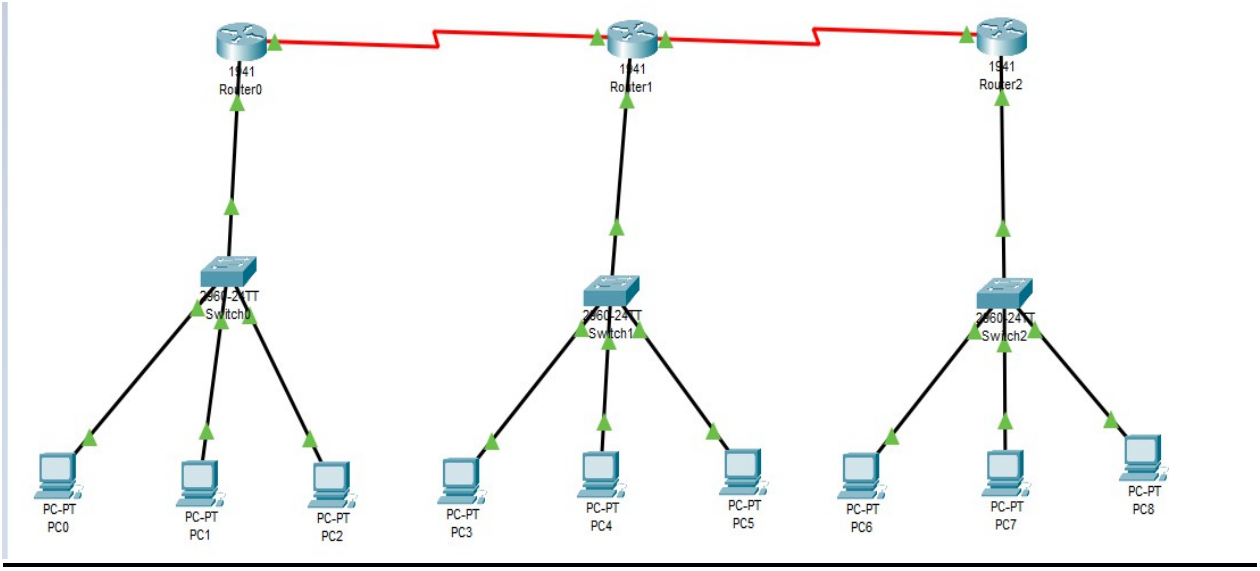



**Conclusion:** Hence assigned IPs and tested cross-devices communication.

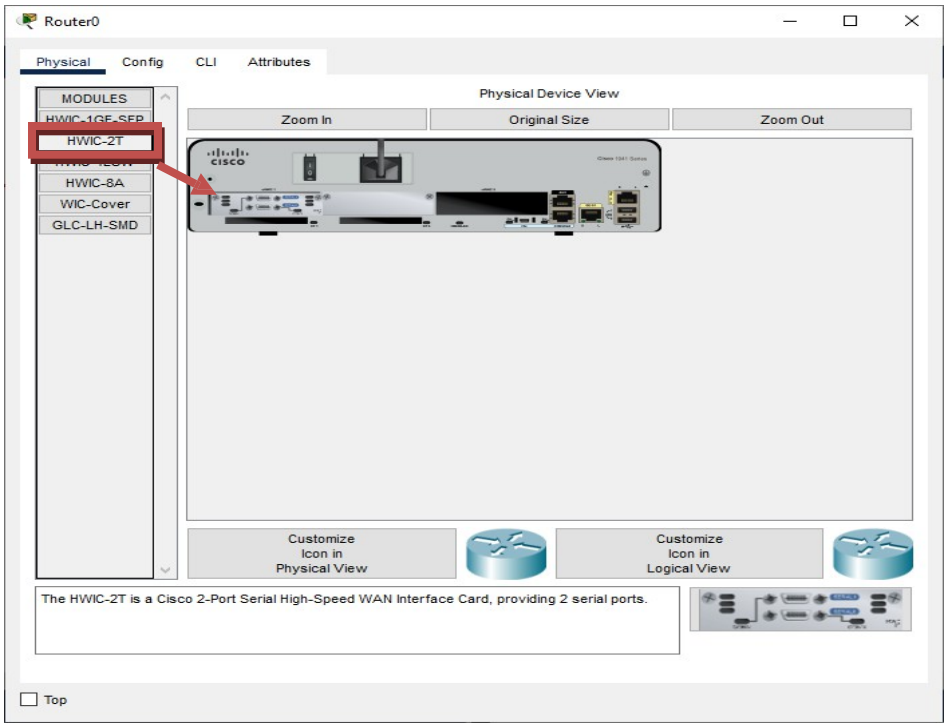
# Practical-8

**Aim:** Three routers and their networks using OSPF. Assign area IDs, router IDs, and enable OSPF. Monitor OSPF neighbor relationships and path selections.

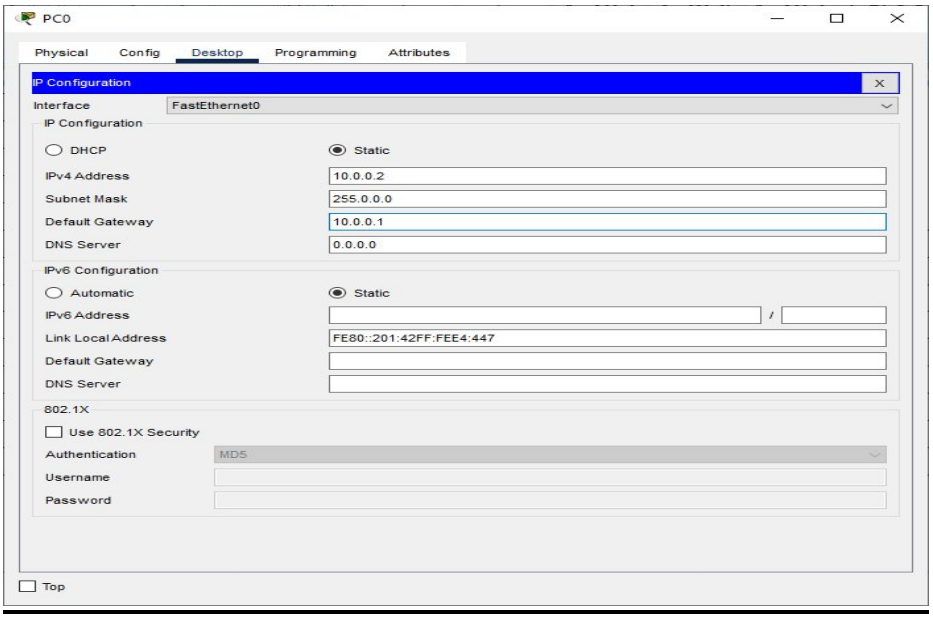
**Topology:**




Adding Serial Interface in each Router (**Same of Others According to the table**):

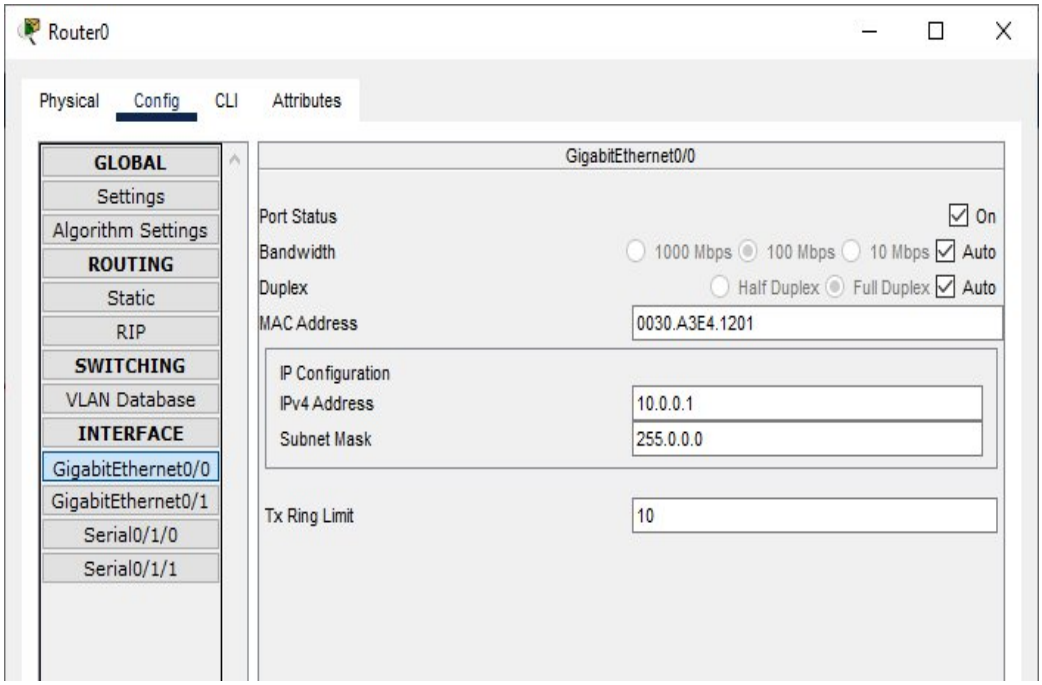


## Configuring PCs (Same of Others According to the table):

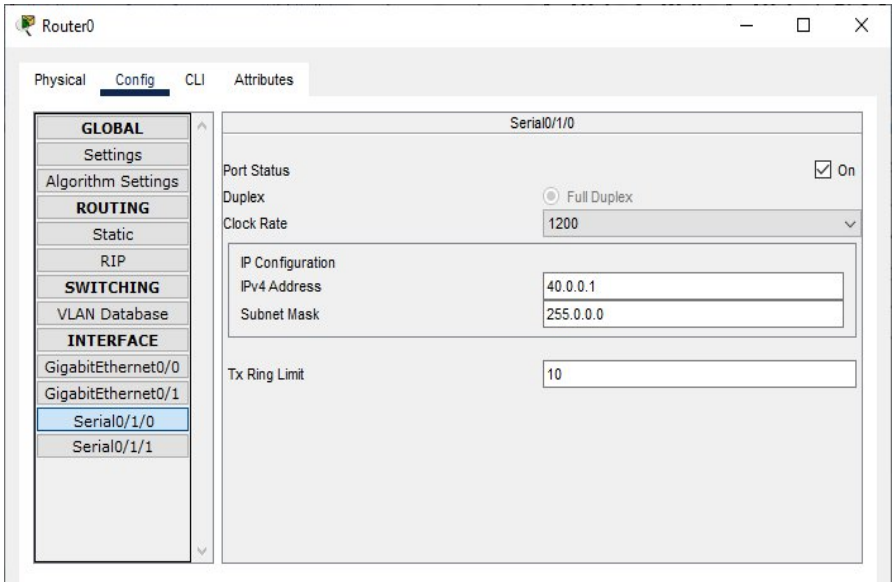



## Configuring IP addresses on Router 0(Same of Others According to the table):

I) Interface G0/0



II) Interface S0/1/0




We configure the above network using the following these IP addresses:

| | Interface | IP address | Default Gateway | Subnet Mask | Wildcard Mask |
|---|---|---|---|---|---|
| Router 0 | G0/0 | 10.0.0.1 | | 255.0.0.0 | 0.255.255.255 |
| | S0/1/0 | 40.0.0.1 | | | |
| Router 1 | G0/0 | 20.0.0.1 | | | |
| | S0/1/0 | 40.0.0.2 | | | |
| | S0/1/1 | 50.0.0.1 | | | |
| Router 2 | G0/0 | 30.0.0.1 | | | |
| | S0/1/1 | 50.0.0.2 | | | |
| PC0 | FastEthernet0 | 10.0.0.2 | 10.0.0.1 | | |
| PC1 | FastEthernet0 | 10.0.0.3 | | | |
| PC2 | FastEthernet0 | 10.0.0.4 | | | |
| PC3 | FastEthernet0 | 20.0.0.2 | 20.0.0.1 | | |
| PC4 | FastEthernet0 | 20.0.0.3 | | | |
| PC5 | FastEthernet0 | 20.0.0.4 | | | |
| PC6 | FastEthernet0 | 30.0.0.2 | 30.0.0.1 | | |
| PC7 | FastEthernet0 | 30.0.0.3 | | | |
| PC8 | FastEthernet0 | 30.0.0.4 | | | |

**Configuring Router 0 for OSPF (using the CLI mode)**

Router(config)#
Router(config)#router ospf 1

Router(config-router)#network 10.0.0.0 0.0.0.255 area 1

Router(config-router)#network 40.0.0.0 0.0.0.255
area 1

Router(config-router)#exit

Router(config)#

**Configuring Router 1 for OSPF (using the CLI mode)**

Router(config)#
Router(config)#router
ospf 1

Router(config-router)#

Router(config-router)#network 20.0.0.0 0.0.0.255 area 1

Router(config-router)#network 40.0.0.0 0.0.0.255 area 1

Router(config-router)#network 50.0.0.0 0.0.0.255
area 1

Router(config-router)#exit

Router(config)#

**Configuring Router 2 for OSPF (using the CLI mode)**

Router(config)#
Router(config)#router
ospf 1

Router(config-router)#

Router(config-router)#network 30.0.0.0 0.0.0.255 area 1

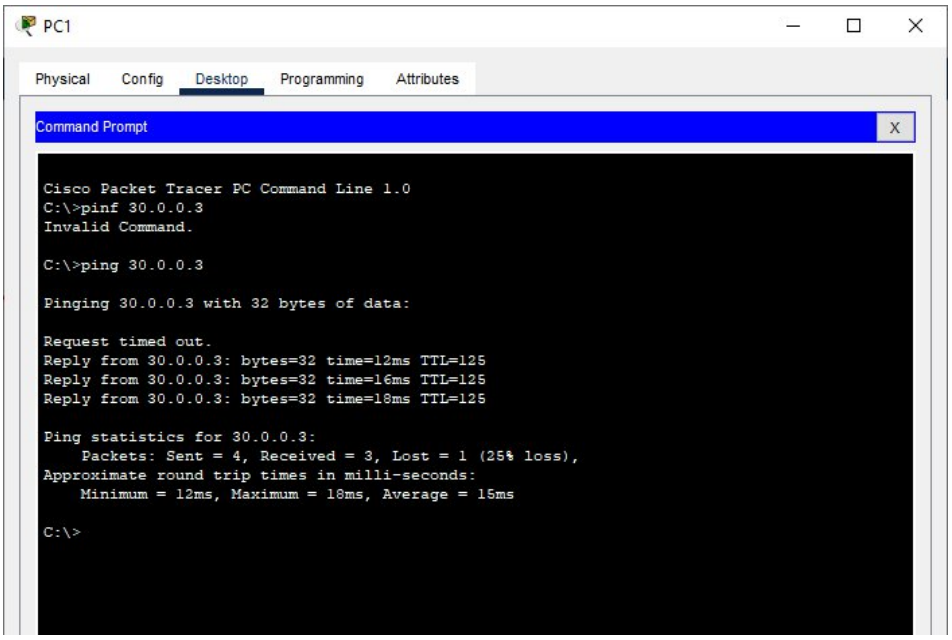
Router(config-router)#network 50.0.0.0 0.0.0.255 area 1

Router(config-router)# exit

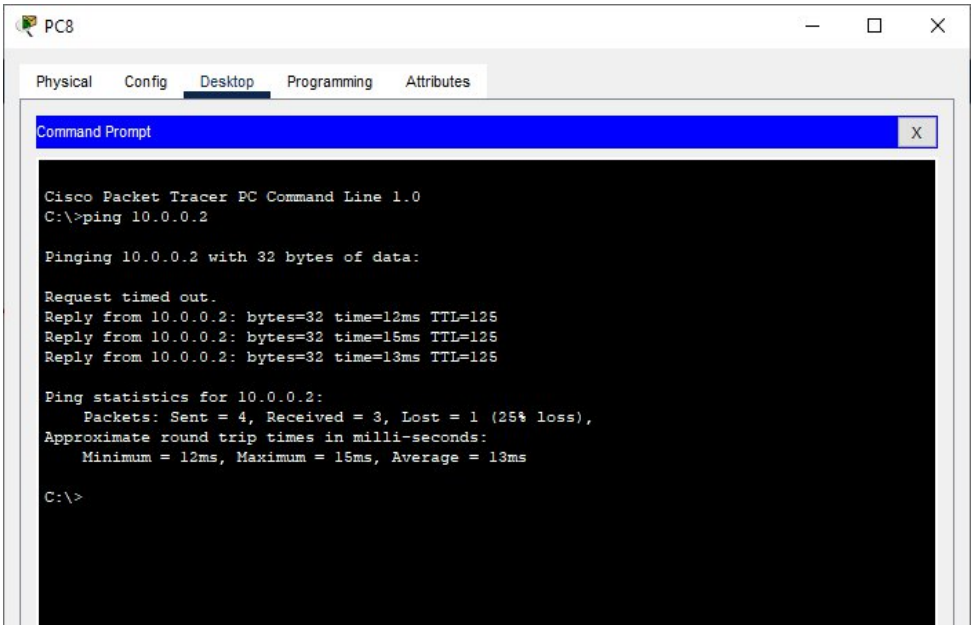
Router(config)#

**Checking the connectivity by using the ping command:**

Pinging PC8 (ip address 10.30.0.4) from PC1




Pinging PC0 (ip address 10.10.0.2) from PC8

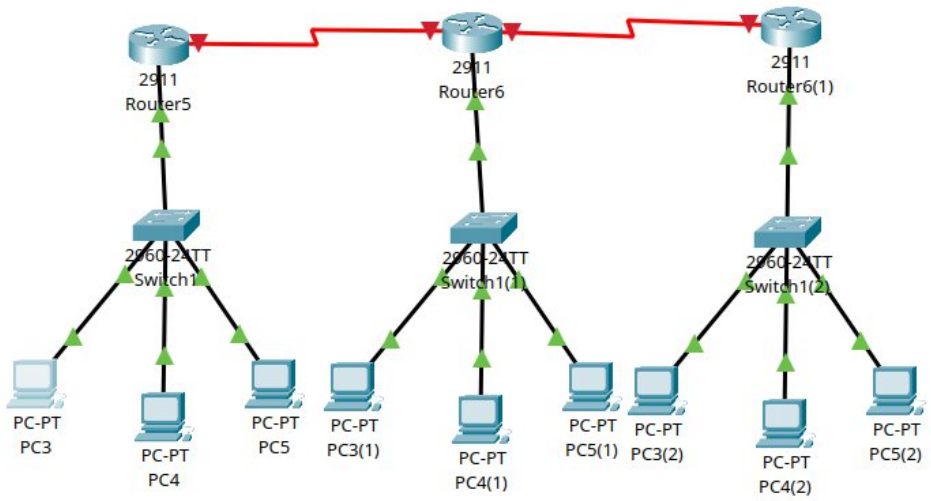



**Conclusion:**Hence the OSPF has been studied and verified through the given network

# Practical-9

**Aim:** Three routers, each connected to at least three PCs. Implement RIPv1 routing between routers Verify inter-network connectivity using ping and route tables.

**Topology:**

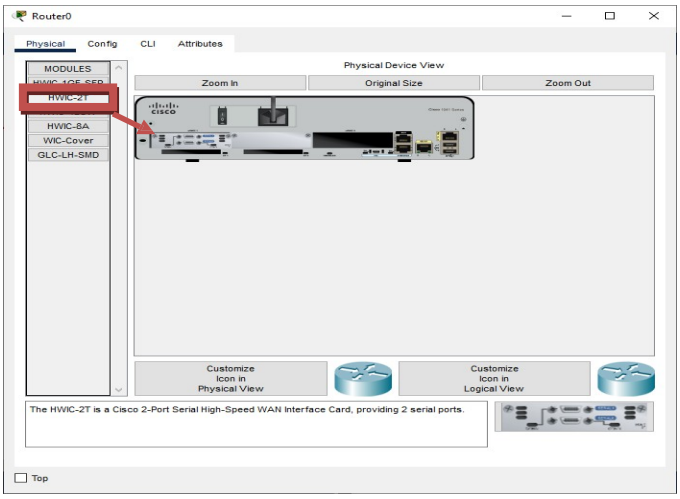



We configure the above network using the following IP addresses

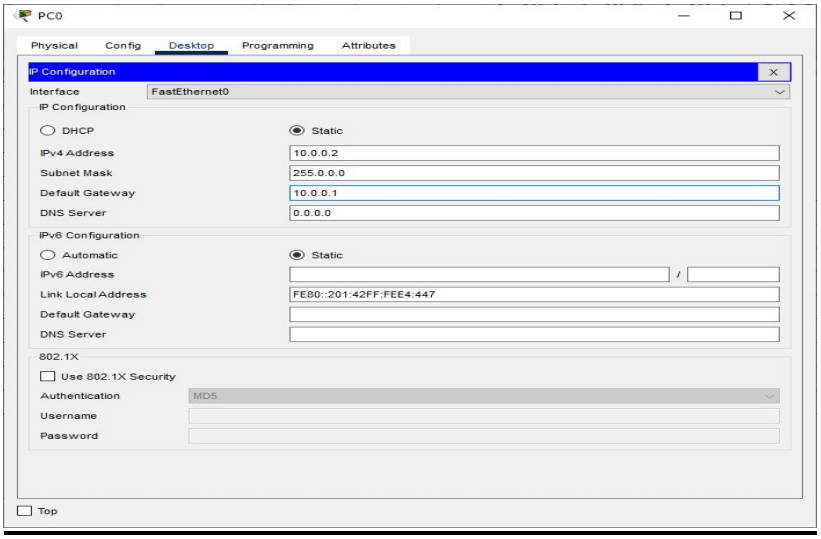
| Host | Interface | IP address | Network Address | Default Gateway |
|---|---|---|---|---|
| Router 0 | G0/0 | 10.0.0.1 | 10.0.0.0 | |
| | S0/1/0 | 192.168.0.1 | 192.168.0.0 | |
| Router 1 | G0/0 | 20.0.0.1 | 20.0.0.0 | |
| | S0/1/0 | 192.168.0.2 | 192.168.0.0 | |
| | S0/1/1 | 192.168.1.1 | 192.168.1.0 | |
| Router 2 | G0/0 | 30.0.0.1 | 30.0.0.0 | |
| | S0/1/1 | 192.168.1.2 | 192.168.1.0 | |
| PC0 | FastEthernet0 | 10.0.0.2 | 10.0.0.0 | 10.0.0.1 |
| PC1 | FastEthernet0 | 10.0.0.3 | 10.0.0.0 | 10.0.0.1 |
| PC2 | FastEthernet0 | 10.0.0.4 | 10.0.0.0 | 10.0.0.1 |

| | | | | |
|---|---|---|---|---|
| PC3 | FastEthernet0 | 20.0.0.2 | 20.0.0.0 | 20.0.0.1 |
| PC4 | FastEthernet0 | 20.0.0.3 | 20.0.0.0 | 20.0.0.1 |
| PC5 | FastEthernet0 | 20.0.0.4 | 20.0.0.0 | 20.0.0.1 |
| PC6 | FastEthernet0 | 30.0.0.2 | 30.0.0.0 | 30.0.0.1 |
| PC7 | FastEthernet0 | 30.0.0.3 | 30.0.0.0 | 30.0.0.1 |
| PC8 | FastEthernet0 | 30.0.0.4 | 30.0.0.0 | 30.0.0.1 |

Adding Serial Interface in each Router (**Same of Others According to the table**):




## Configuring PCs (Same of Others According to the table):



## Configuring Router Via CLI:

### Router0:

```
enable
configure terminal
interface GigabitEthernet0/0
 ip address 10.0.0.1 255.0.0.0
 no shutdown
exit
interface Serial0/1/0
 ip address 192.168.0.1 255.255.255.0
 clock rate 64000
 no shutdown
exit
router rip
 network 10.0.0.0
 network 192.168.0.0
Exit
```

### Rounter1:

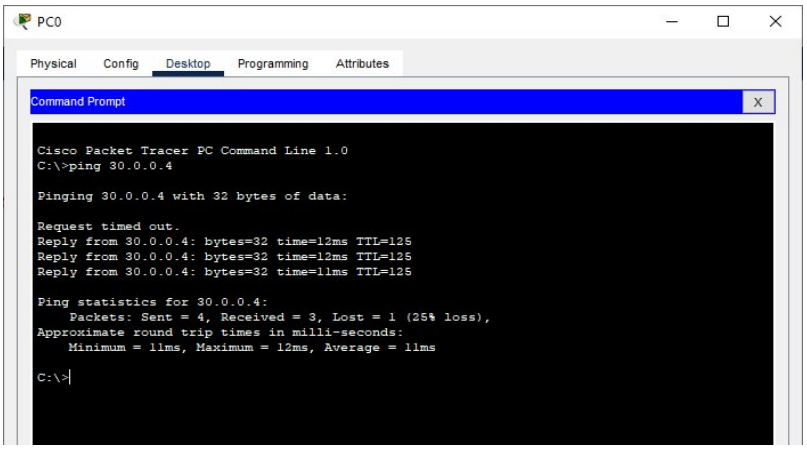
```
enable
configure terminal
interface GigabitEthernet0/0
 ip address 20.0.0.1 255.0.0.0
 no shutdown
exit
interface Serial0/1/0
 ip address 192.168.0.2 255.255.255.0
 no shutdown
exit
interface Serial0/1/1
 ip address 192.168.1.1 255.255.255.0
 clock rate 64000
 no shutdown
exit
router rip
 network 20.0.0.0
 network 192.168.0.0
 network 192.168.1.0
exit
```

### Rounter2:

```
enable
configure terminal
```

```
interface GigabitEthernet0/0
 ip address 30.0.0.1 255.0.0.0
 no shutdown
exit
interface Serial0/1/1
 ip address 192.168.1.2 255.255.255.0
 no shutdown
exit
router rip
 network 30.0.0.0
 network 192.168.1.0
Exit
```

## Pinging PC8 (ip address 30.0.0.4) from PC0:

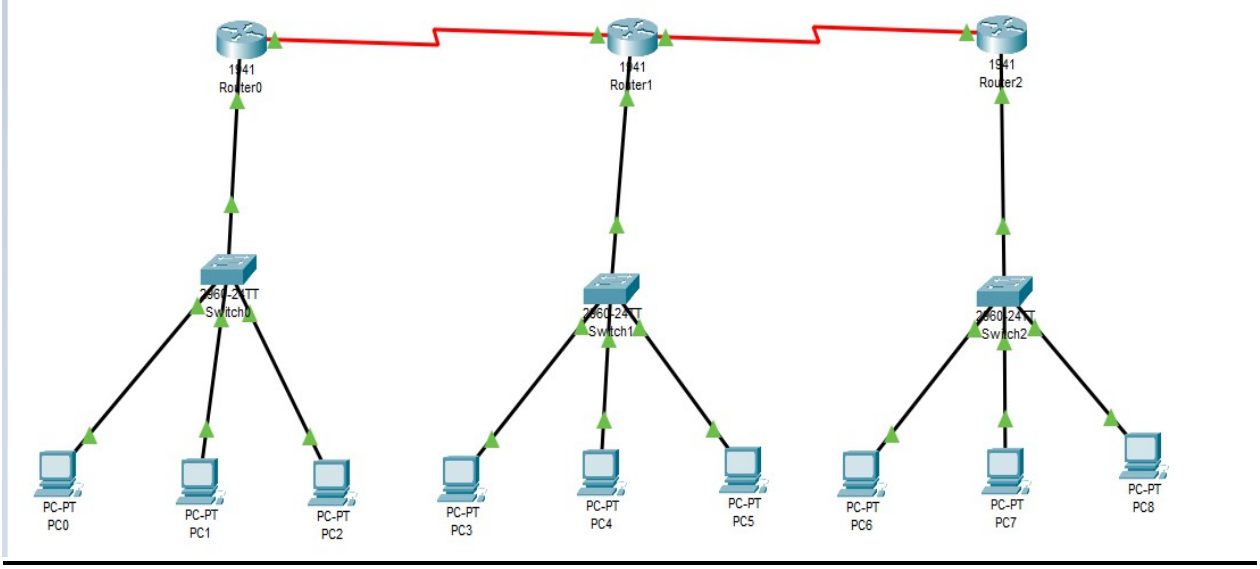



**Conclusion:** Hence the RIPv1 has been studied and verified through the given network

# Practical-10

**Aim:** Three Routers, each connected to at least 3 PCs. Enable RIPv2 and observe mask handling. Using Packet Tracer's sim to observe routing updates.

**Topology:**

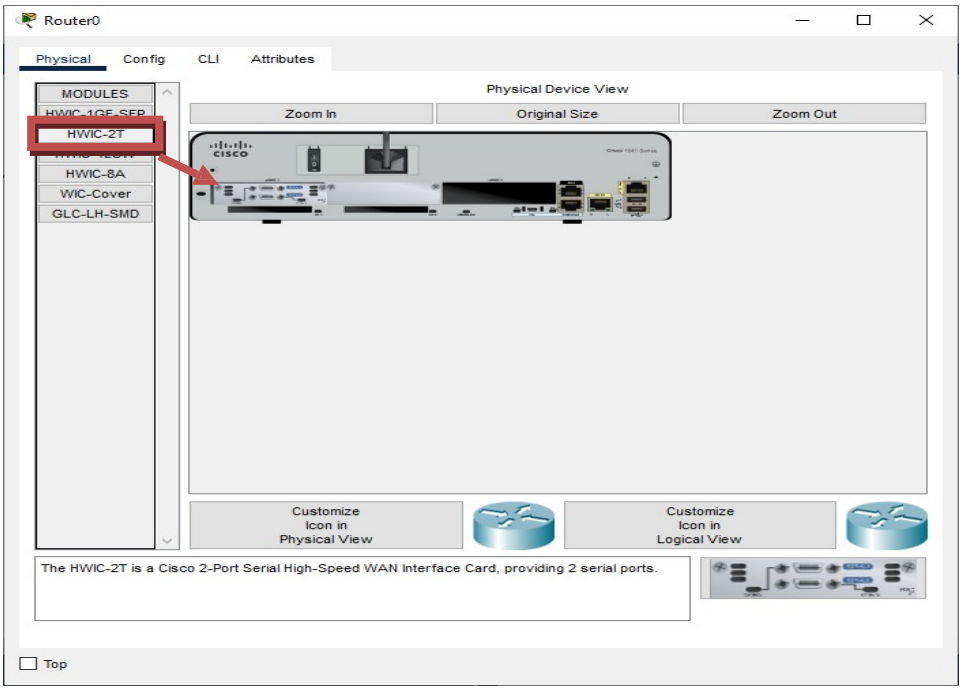



We configure the above network using the following IP addresses

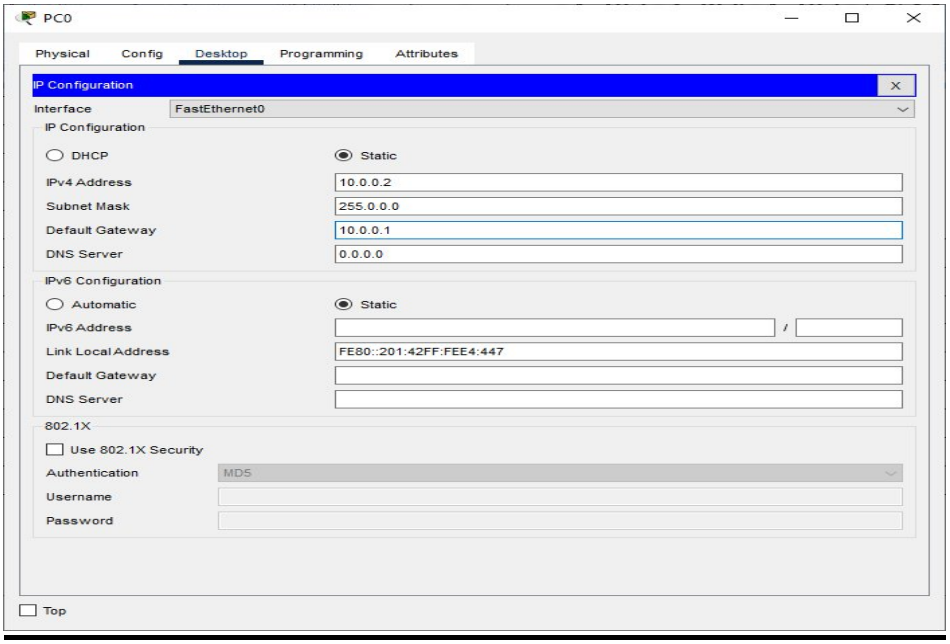
| Host | Interface | IP address | Subnet Mask | Network Address | Default Gateway |
|---|---|---|---|---|---|
| Router 0 | G0/0 | 10.10.0.1 | 255.255.255.248 | 10.10.0.0 | |
| | S0/1/0 | 192.168.0.1 | 255.255.255.252 | 192.168.0.0 | |
| Router 1 | G0/0 | 10.20.0.1 | 255.255.255.248 | 10.20.0.0 | |
| | S0/1/0 | 192.168.0.2 | 255.255.255.252 | 192.168.0.0 | |
| | S0/1/1 | 192.168.1.1 | 255.255.255.252 | 192.168.1.0 | |
| Router 2 | G0/0 | 10.30.0.1 | 255.255.255.248 | 10.30.0.0 | |
| | S0/1/1 | 192.168.1.2 | 255.255.255.252 | 192.168.1.0 | |
| PC0 | FastEthernet0 | 10.10.0.2 | 255.255.255.248 | 10.10.0.0 | 10.10.0.1 |
| PC1 | FastEthernet0 | 10.10.0.3 | 255.255.255.248 | 10.10.0.0 | 10.10.0.1 |
| PC2 | FastEthernet0 | 10.10.0.4 | 255.255.255.248 | 10.10.0.0 | 10.10.0.1 |
| PC3 | FastEthernet0 | 10.20.0.2 | 255.255.255.248 | 10.20.0.0 | 10.20.0.1 |
| PC4 | FastEthernet0 | 10.20.0.3 | 255.255.255.248 | 10.20.0.0 | 10.20.0.1 |
| PC5 | FastEthernet0 | 10.20.0.4 | 255.255.255.248 | 10.20.0.0 | 10.20.0.1 |
| PC6 | FastEthernet0 | 10.30.0.2 | 255.255.255.248 | 10.30.0.0 | 10.30.0.1 |

| | | | | | |
|---|---|---|---|---|---|
| PC7 | FastEthernet0 | 10.30.0.3 | 255.255.255.248 | 10.30.0.0 | 10.30.0.1 |
| PC8 | FastEthernet0 | 10.30.0.4 | 255.255.255.248 | 10.30.0.0 | 10.30.0.1 |

## Adding Serial Interface in each Router (Same of Others According to the table):

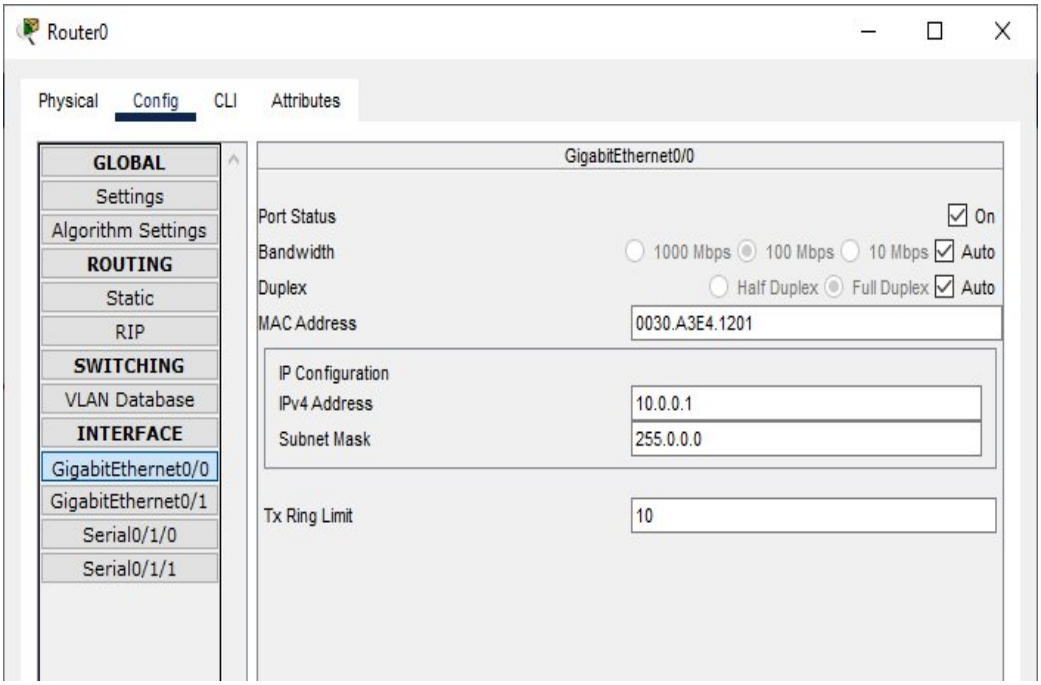



## Configuring PCs (Same of Others According to the table):

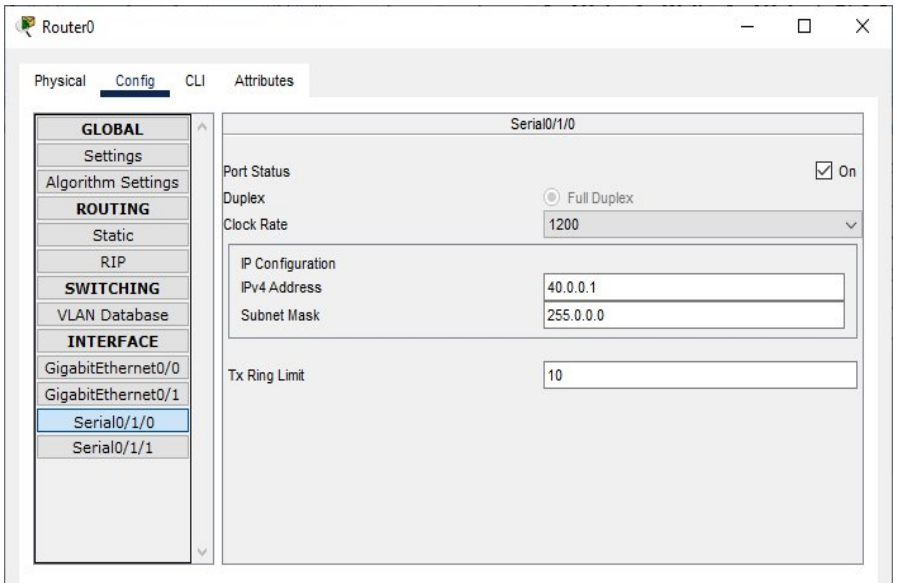


**Configuring IP addresses on Router 0(Same of Others According to the table):**

I) Interface G0/0




II) Interface S0/1/0



## Configuring Router Via CLI:

### Router0:

```
Router>enable
Router#configure terminal
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.10.0.0
Router(config-router)#network 192.168.0.0
Router(config-router)#exit
Router(config)#
```

### Rounter1:

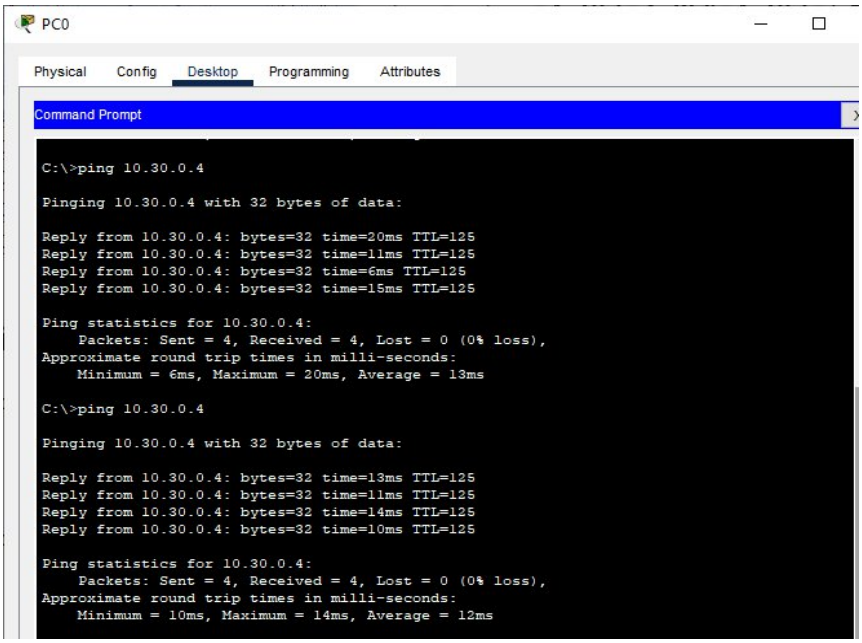
```
Router>enable
Router#configure terminal
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.20.0.0
Router(config-router)#network 192.168.0.0
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#
```

### Rounter2:

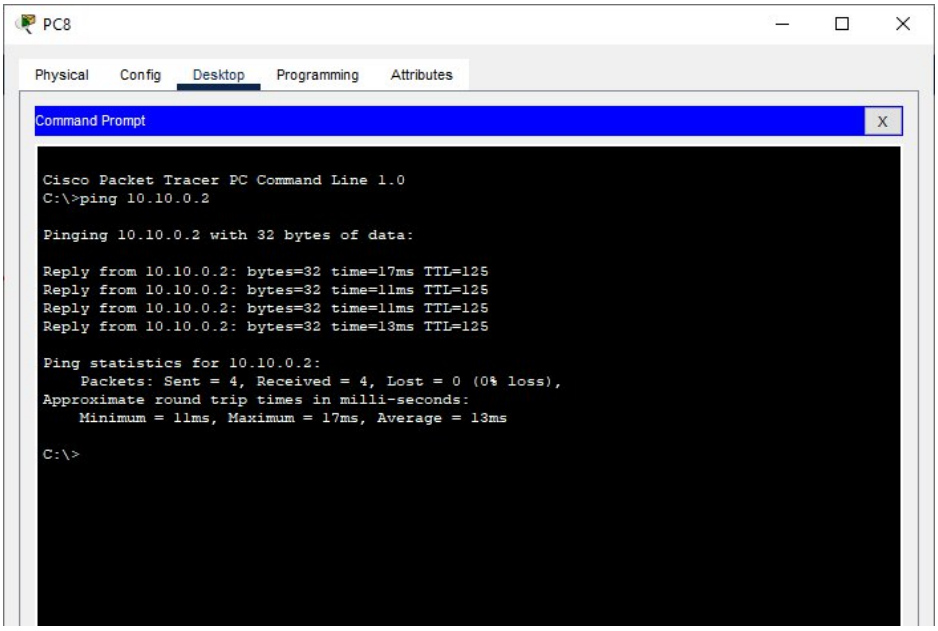
```
Router>enable
Router#configure terminal
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.30.0.0
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#
```

**Checking connectivity By Using Ping command:**

I) Pinging PC8 from PC1:




II) Pinging PC0 from PC8:




**Conclusion:** Hence the RIPv2 has been studied and verified through the given network