PRACTICAL NO 4: Configure IP ACLs to Mitigate Attacks and Configuring IPv6 ACLs

Access Control Lists (ACLs)

Network administrators must figure out how to deny unwanted access to the network while allowing internal users appropriate access to necessary services. Although security tools, such as passwords, callback equipment, and physical security devices are helpful, they often lack the flexibility of basic traffic filtering and the specific controls most administrators prefer.

For example, a network administrator may want to allow users access to the Internet, but not permit external users telnet access into the LAN.
Routers provide basic traffic filtering capabilities, such as blocking Internet traffic, with access control lists (ACLs).

An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols.

The router examines each packet to determine whether to forward or drop it, based on the conditions specified in the ACL. Some ACL decision points are:

1) IP source address
2) IP destination addresses
3) UDP or TCP protocols
4) Upper-layer (TCP/UDP) port numbers

ACLs must be defined on a:

1) Per-protocol (IP, IPX, AppleTalk)
2) Per direction (in or out)
3) Per port (interface) basis.
4) ACLs control traffic in one direction at a time on an interface.
5) A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
6) Finally every interface can have multiple protocols and directions defined.

An ACL is a group of statements that define whether packets are accepted or rejected coming into an interface or leaving an interface.

1) ACL statements operate in sequential, logical order (top down).
2) If a condition match is true, the packet is permitted or denied and the rest of the ACL statements are not checked.

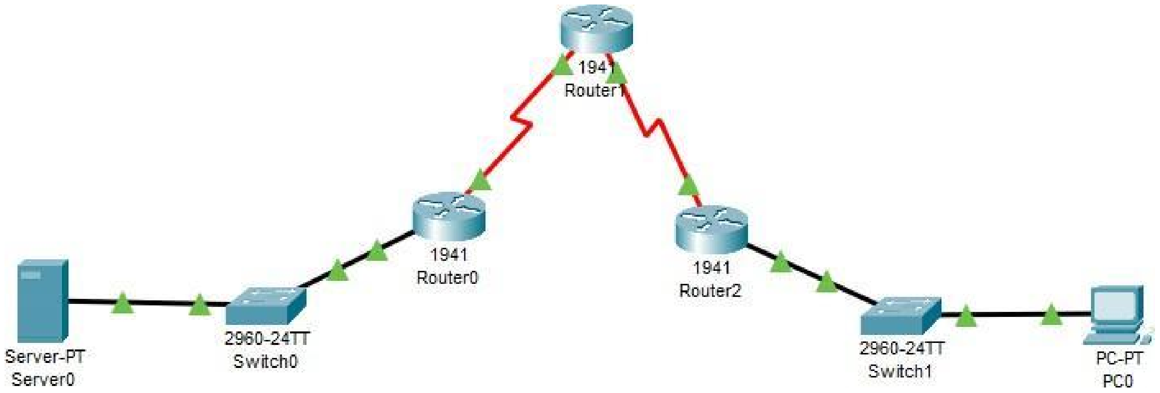
3) If all the ACL statements are unmatched, an implicit "deny any"
statement is placed at the end of the list by default. (not visible) When first learning how to create ACLs, it is a good idea to add the
implicit deny at the end of ACLs to reinforce the dynamic presence of the command line.

Standard IP ACLs
Can only filter on source IP addresses

Extended IP ACLs Can filter on:
1) Source IP address
2) Destination IP address
3) Protocol (TCP, UDP)
4) Port Numbers (Telnet – 23, http – 80, etc.) and other parameters

An access list is a sequential series of commands or filters. These lists tell the router what types of packets to: accept or deny Acceptance and denial can be based on specified conditions. ACLs applied on the router's interfaces

We use the following topology to study the present case

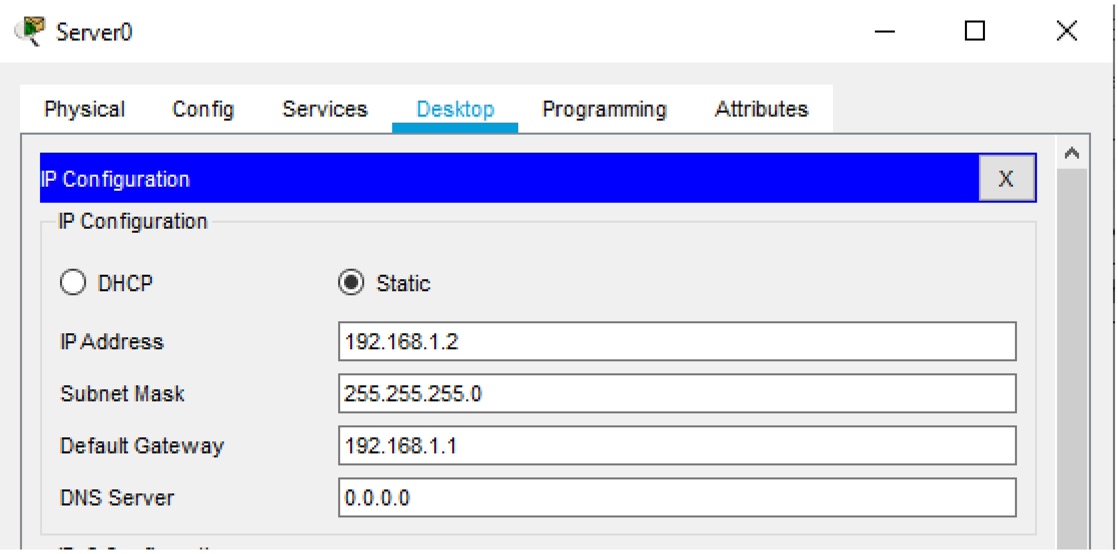


Let us consider the following Address table to configure the network devices:

Device	Interface	IP Address	Subnet Mask	Default gateway	Switch Port
PC 0	NA	192.168.4.2	255.255.255.0	192.168.4.1	Switch1 F0/1
Server0	NA	192.168.1.2	255.255.255.0	192.168.1.1	Switch0 F0/1
Router0	GE0/0	192.168.1.1	255.255.255.0	NA	Switch0 F0/5
	S0/1/0	192.168.2.1	255.255.255.0	NA	NA
Router1	S0/1/0	192.168.2.2	255.255.255.0	NA	NA
	S0/1/1	192.168.3.1	255.255.255.0	NA	NA
Router2	S0/1/1	192.168.3.2	255.255.255.0	NA	NA
	GE0/0	192.168.4.1	255.255.255.0	NA	Switch1 F0/5

Part 1 - Verify connectivity among devices before firewall configuration

Configuring Server 0



IP Configuration

○ DHCP ● Static

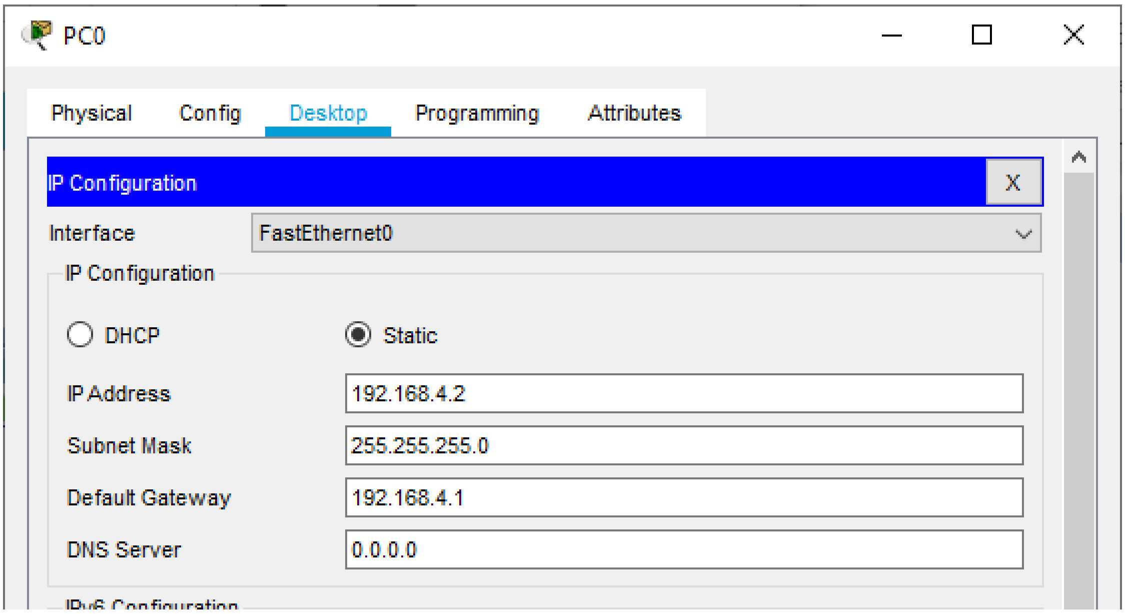
IP Address 192.168.1.2

Subnet Mask 255.255.255.0

Default Gateway 192.168.1.1

DNS Server 0.0.0.0

Configuring PC0



Interface FastEthernet0

IP Configuration

○ DHCP ● Static

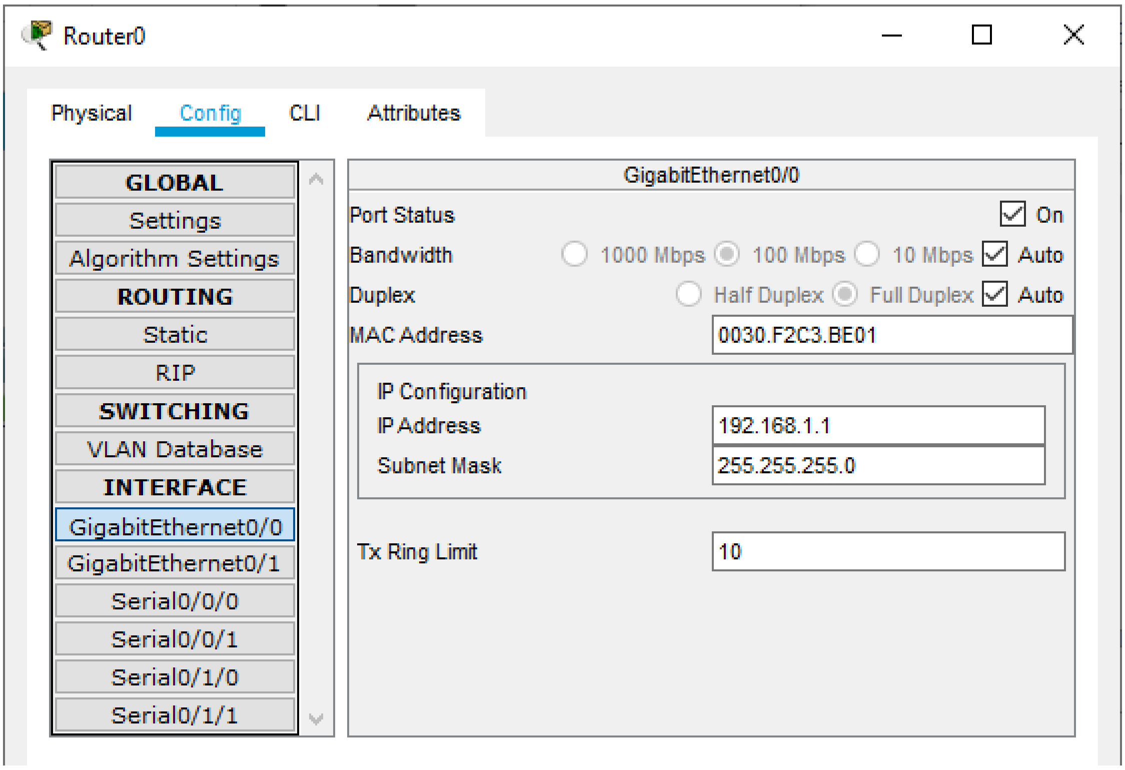
IP Address 192.168.4.2

Subnet Mask 255.255.255.0

Default Gateway 192.168.4.1

DNS Server 0.0.0.0

Configuring Router0



GigabitEthernet0/0

Port Status ☑ On

Bandwidth ○ 1000 Mbps ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

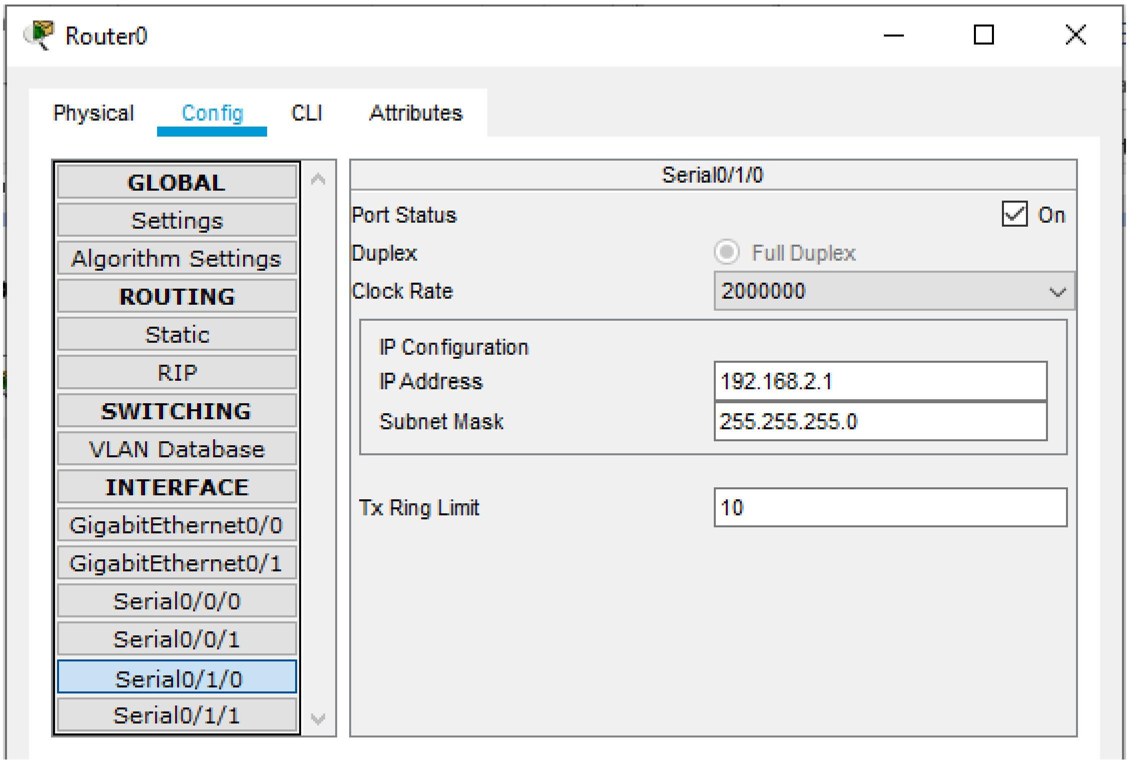
MAC Address 0030.F2C3.BE01

IP Configuration

IP Address 192.168.1.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10



Serial0/1/0

Port Status ☑ On

Duplex ● Full Duplex

Clock Rate 2000000

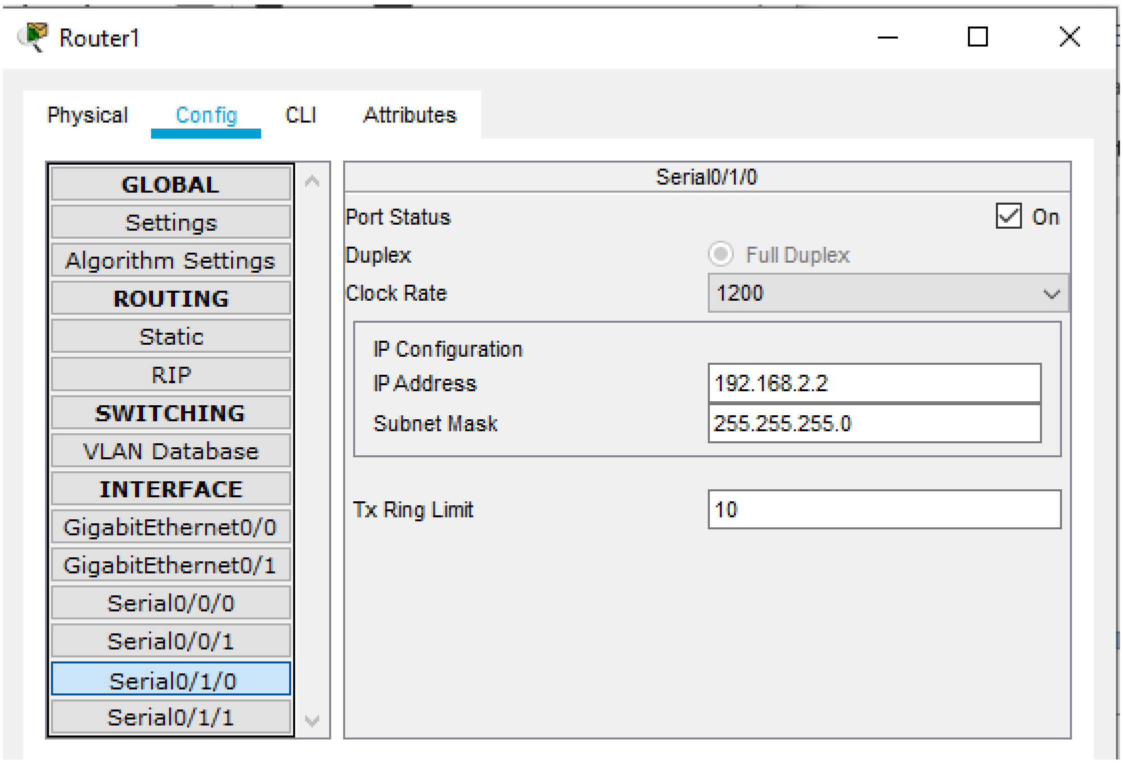
IP Configuration

IP Address 192.168.2.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10

Configuring Router1



Serial0/1/0

Port Status ☑ On

Duplex ● Full Duplex

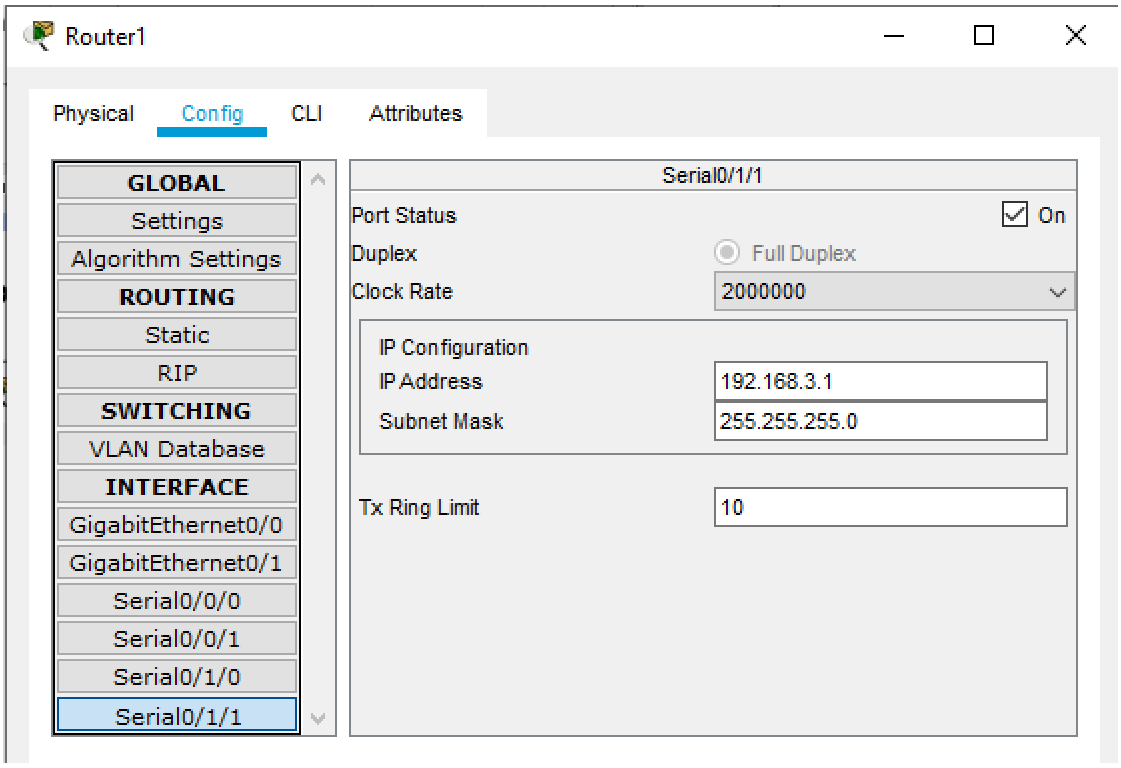
Clock Rate 1200

IP Configuration

IP Address 192.168.2.2

Subnet Mask 255.255.255.0

Tx Ring Limit 10



Serial0/1/1

Port Status ☑ On

Duplex ● Full Duplex

Clock Rate 2000000

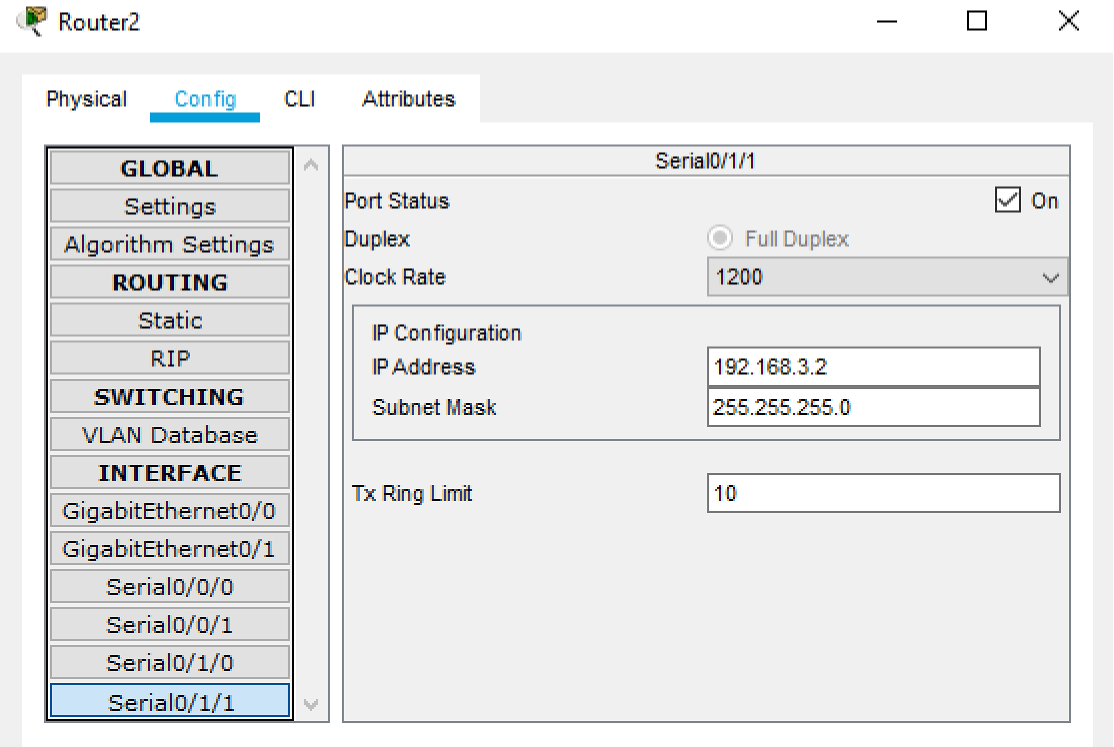
IP Configuration

IP Address 192.168.3.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10

Configuring Router2



Serial0/1/1

Port Status ☑ On

Duplex ● Full Duplex

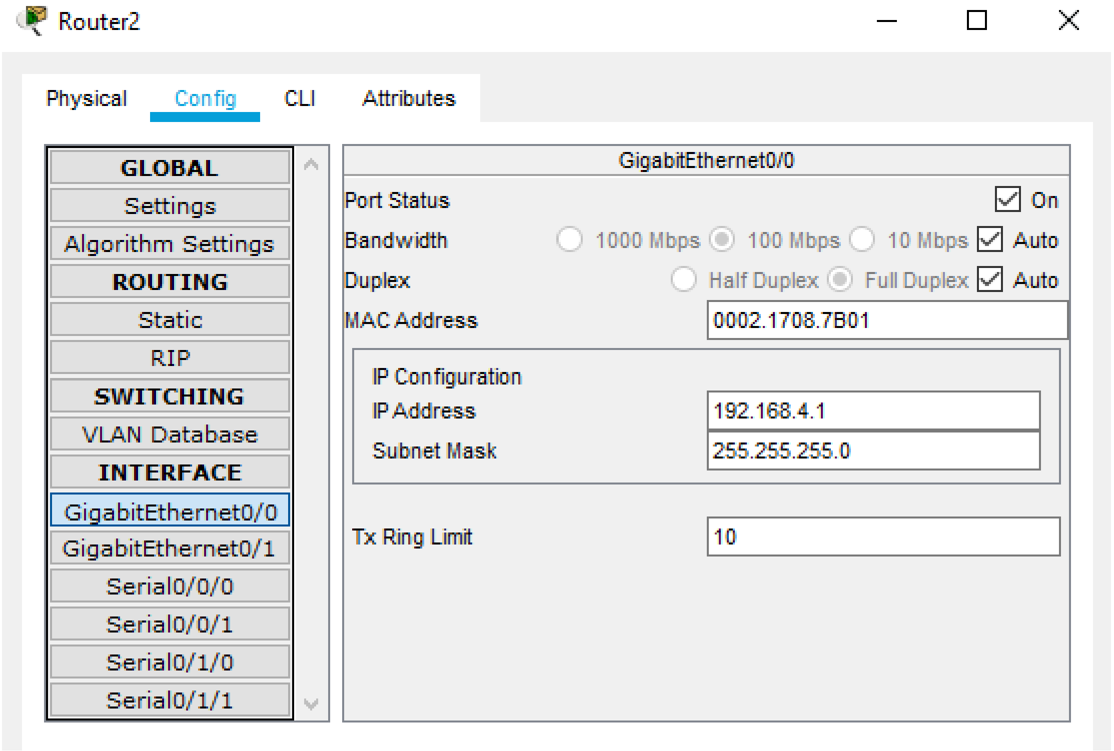
Clock Rate 1200

IP Configuration

IP Address 192.168.3.2

Subnet Mask 255.255.255.0

Tx Ring Limit 10



GigabitEthernet0/0

Port Status ☑ On

Bandwidth ○ 1000 Mbps ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

MAC Address 0002.1708.7B01

IP Configuration

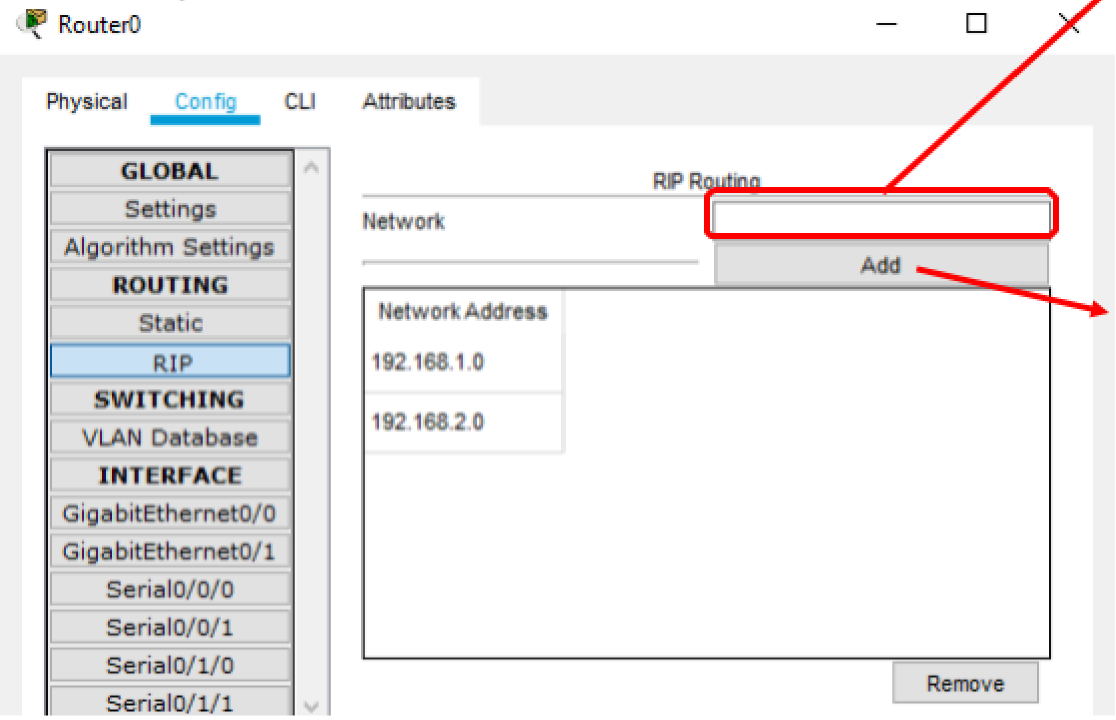
IP Address 192.168.4.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10

Type the IP address

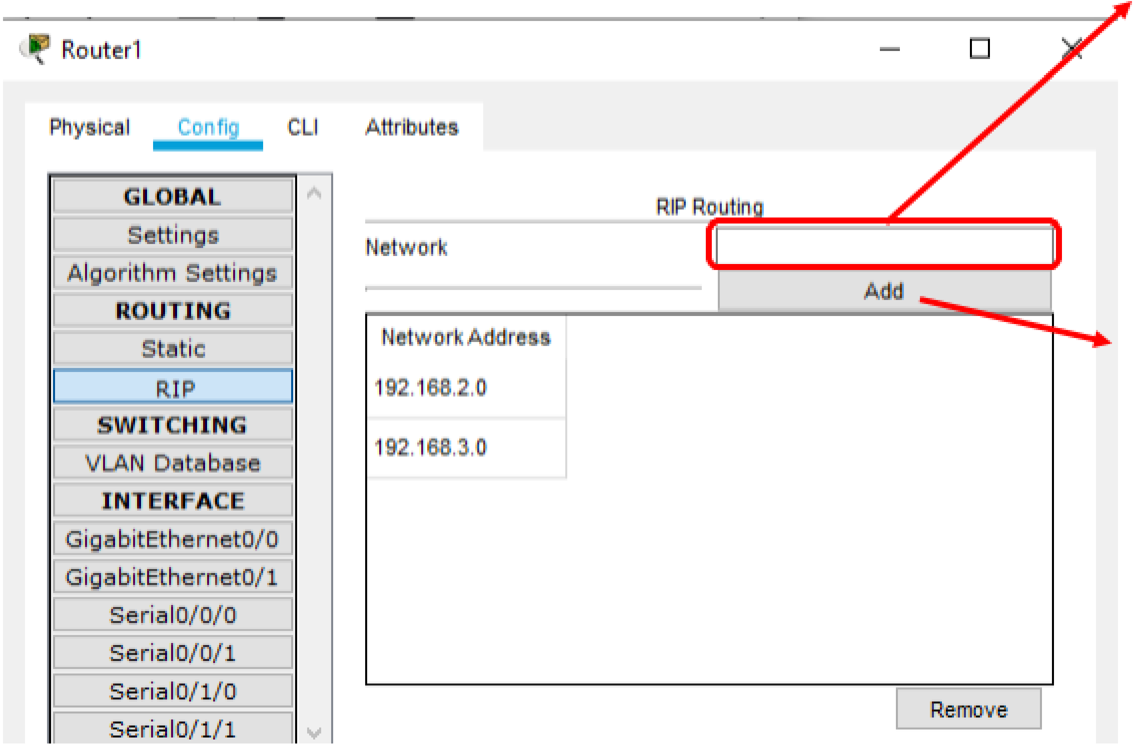
Set the RIP protocol on both the Routers as follows



Network Address
192.168.1.0
192.168.2.0

Click Add button

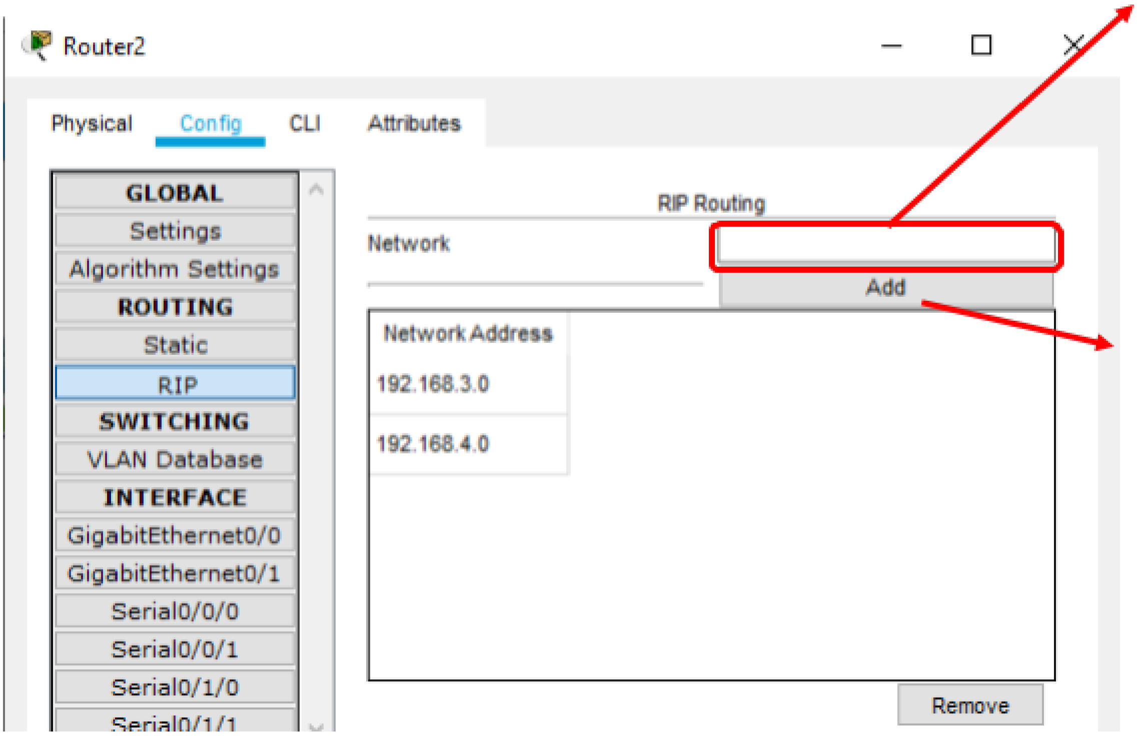
Type the IP address



Network Address
192.168.2.0
192.168.3.0

Click Add button

Type the IP address

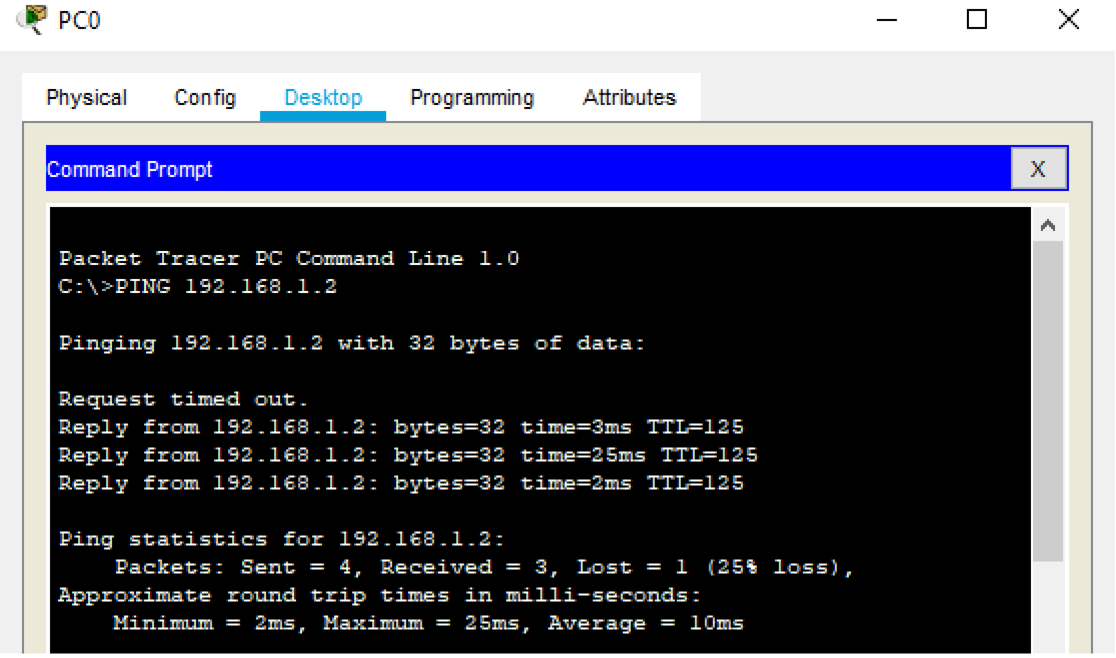


RIP Routing

Network Address
192.168.3.0
192.168.4.0

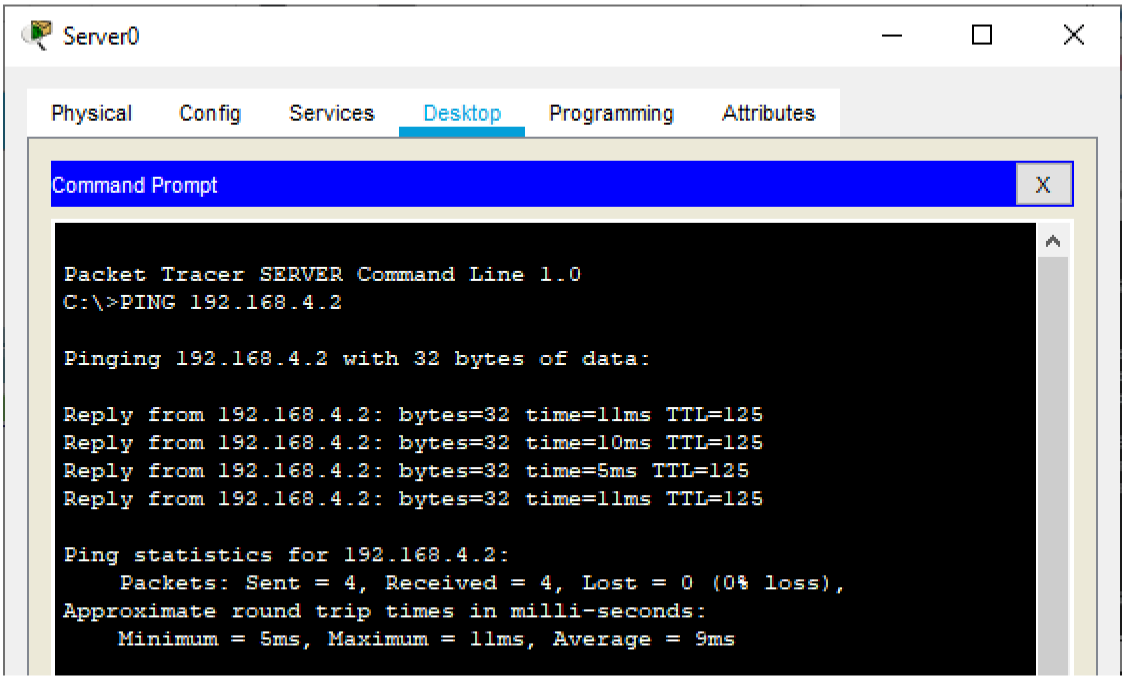
Click Add button

We can now verify the connectivity by pinging Server from PC



```
Packet Tracer PC Command Line 1.0
C:\>PING 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.1.2: bytes=32 time=3ms TTL=125
Reply from 192.168.1.2: bytes=32 time=25ms TTL=125
Reply from 192.168.1.2: bytes=32 time=2ms TTL=125

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 10ms
```

We can now verify the connectivity by pinging PC from Server



```
Packet Tracer SERVER Command Line 1.0
C:\>PING 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.4.2: bytes=32 time=11ms TTL=125
Reply from 192.168.4.2: bytes=32 time=10ms TTL=125
Reply from 192.168.4.2: bytes=32 time=5ms TTL=125
Reply from 192.168.4.2: bytes=32 time=11ms TTL=125

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 11ms, Average = 9ms
```

Part 2 – Secure Access to Routers

We configure ACL 10 to block all remote access to the Routers and allow remote access only from PC. We type the following commands in all the Routers (Router0, Router1, and Router2). This part is divided in 2 subparts

Set up the SSH protocol

Enter the following commands in CLI mode of Router0

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router0
Router0(config)#
Router0(config)#crypto key generate rsa
Router0(config)#line vty 0 4
Router0(config-line)#transport input ssh
Router0(config-line)#login local
Router0(config-line)#exit
Router0(config)#username SSHadmin privilege 15 password ismail
Router0(config)#exit
Router0#
```

Enter the following commands in CLI mode of Router1

```
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router1
Router1(config)#
Router1(config)#crypto key generate rsa
Router1(config)#line vty 0 4
Router1(config-line)#transport input ssh
Router1(config-line)#login local
Router1(config-line)#exit
Router1(config)#username SSHadmin privilege 15 password ismail
Router1(config)#exit
Router1#
```

Enter the following commands in CLI mode of Router2

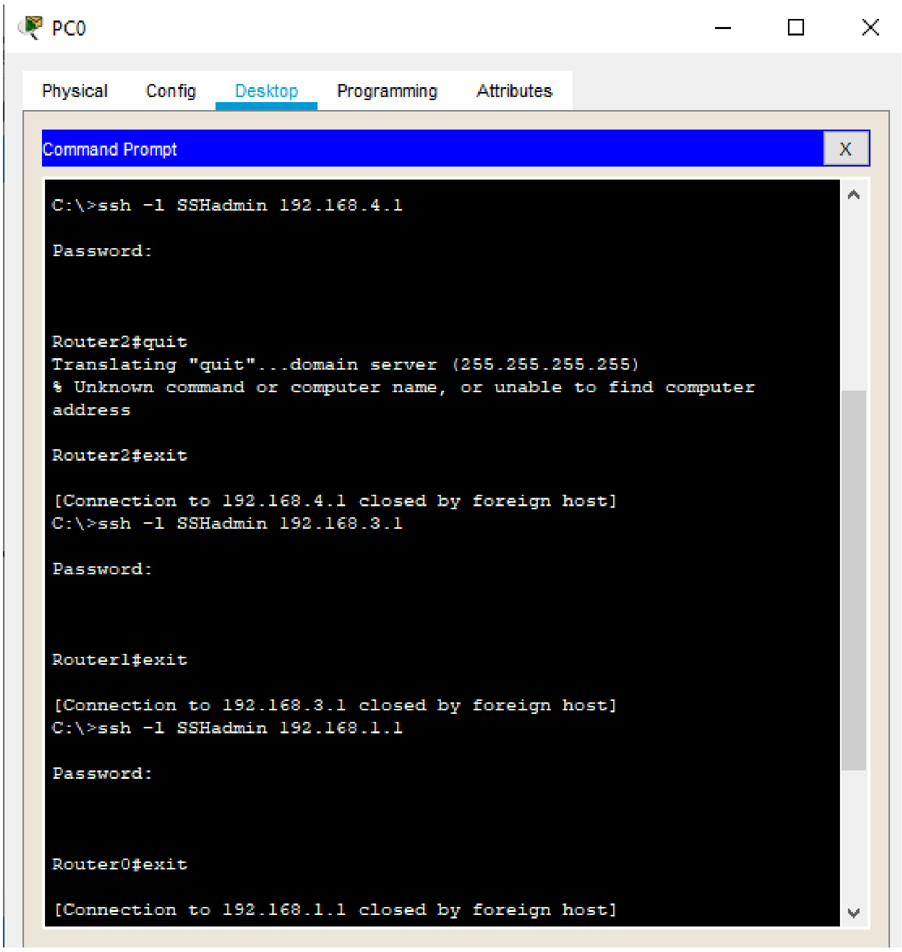
```
Router>enable
Router#configure terminal
```

```
Router(config)#ip domain-name ismail.com
Router(config)#hostname Router2
Router2(config)#
Router2(config)#crypto key generate rsa
Router2(config)#line vty 0 4
Router2(config-line)#transport input ssh
Router2(config-line)#login local
Router2(config-line)#exit
Router2(config)#username SSHadmin privilege 15 password ismail
Router2(config)#exit
Router2#
```

Create an ACL 10 to permit remote access to PC only
Enter the following commands in CLI mode of all Routers

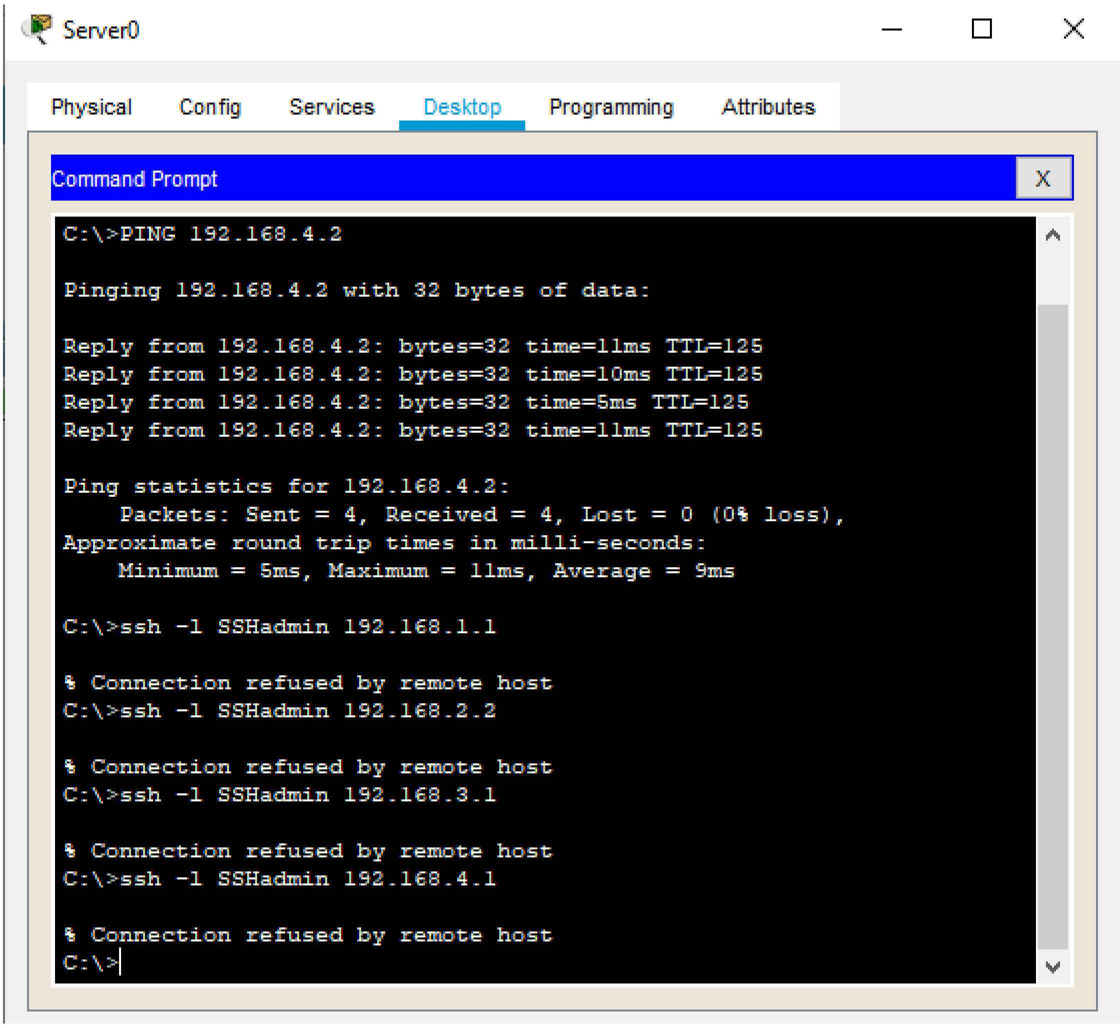
```
Router>enable
Router#configure terminal
Router(config)#access-list 10 permit host 192.168.4.2
Router(config)#line vty 0 4
Router(config-line)#access-class 10 in
```

Now we verify the remote access from PC using the following and find it to be successful



```
C:\>ssh -l SSHadmin 192.168.4.1

Password:



Router2#quit
Translating "quit"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer
address

Router2#exit

[Connection to 192.168.4.1 closed by foreign host]
C:\>ssh -l SSHadmin 192.168.3.1

Password:



Router1#exit

[Connection to 192.168.3.1 closed by foreign host]
C:\>ssh -l SSHadmin 192.168.1.1

Password:



Router0#exit

[Connection to 192.168.1.1 closed by foreign host]
```

Now we verify the remote access from Server using the following and find it to be failure



```
C:\>PING 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.4.2: bytes=32 time=11ms TTL=125
Reply from 192.168.4.2: bytes=32 time=10ms TTL=125
Reply from 192.168.4.2: bytes=32 time=5ms TTL=125
Reply from 192.168.4.2: bytes=32 time=11ms TTL=125

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 11ms, Average = 9ms

C:\>ssh -l SSHadmin 192.168.1.1

% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.2.2

% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.3.1

% Connection refused by remote host
C:\>ssh -l SSHadmin 192.168.4.1

% Connection refused by remote host
C:\>
```

Part 3 - Create a Numbered IP ACL 120 on R1

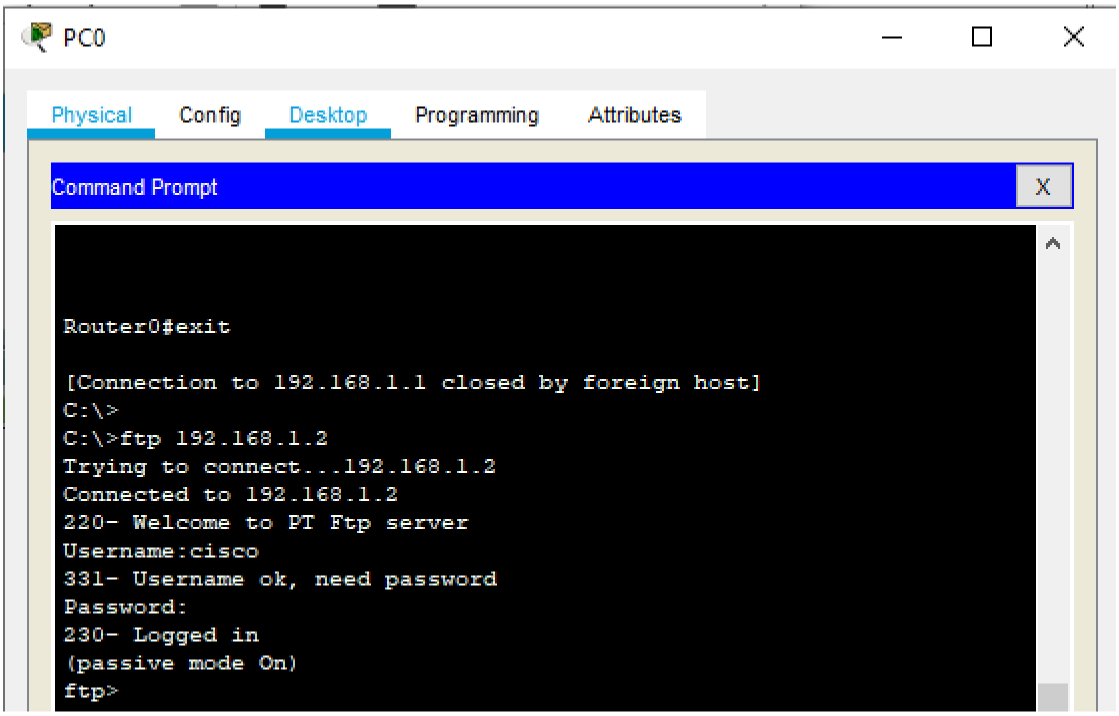
We need to perform the following in this part

1) Create an IP ACL numbered 120 on R1 using the following rules
2) Permit any outside host to access DNS, SMTP, and FTP services on server
3) Deny any outside host access to HTTPS services on **server**
4) Permit PC to access Router1 via SSH. (Done in previous part)

Enter the following commands in the CLI mode of Router1

```
Router1>enable
Router1#
Router1#configure terminal
Router1(config)#access-list 120 permit udp any host 192.168.1.2 eq domain
Router1(config)#access-list 120 permit tcp any host 192.168.1.2 eq
smtp Router1(config)#access-list 120 permit tcp any host 192.168.1.2
eq ftp Router1(config)#access-list 120 deny tcp any host 192.168.1.2 eq
443 Router1(config)#exit
Router1#configure terminal
Router1(config)#interface Serial0/1/1
Router1(config-if)#ip access-group 120
in
```

Verify the above entering the following commands in the PC



```
Router0#exit

[Connection to 192.168.1.1 closed by foreign host]
C:\>
C:\>ftp 192.168.1.2
Trying to connect...192.168.1.2
Connected to 192.168.1.2
220- Welcome to PT Ftp server
Username:cisco
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Hence, we have applied and verified all the required ACLs

Configuring IPv6 ACLs

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the deny and **permit** keywords in global configuration mode.

IPv6 extended ACLs augments standard IPv6 ACL functionality to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

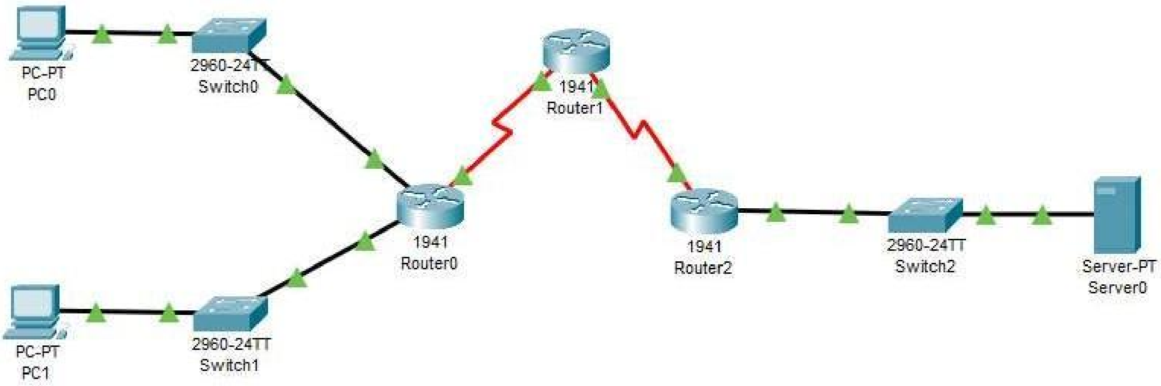
IPv6 Packet Inspection

The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access- class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local device address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local device address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

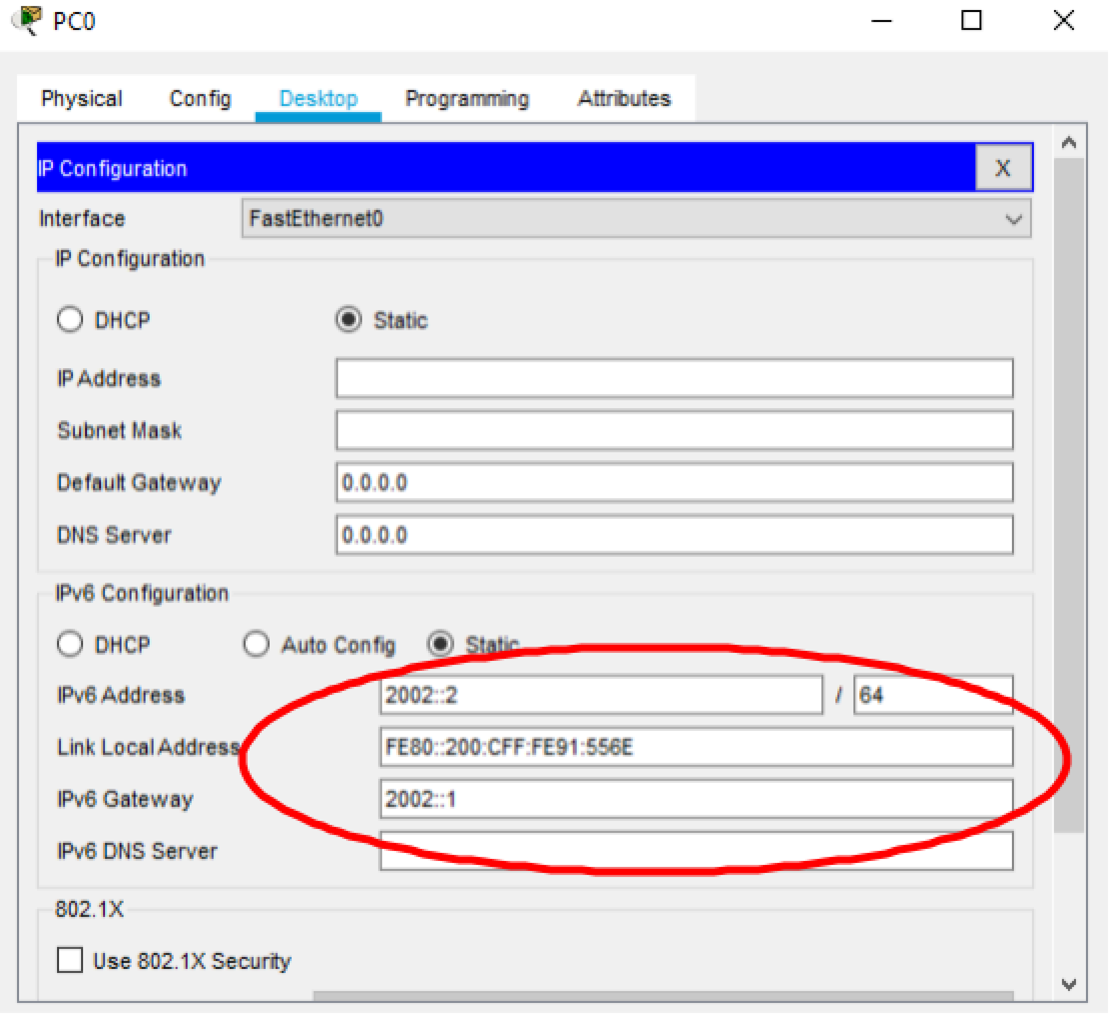
We use the following topology



Let us consider the following Address table to configure the network devices:

Device	Interface	IPv6 Address	IPv6 gateway	Switch Port
PC 0	NA	2002::2 / 64	2002::1	Switch0 F0/1
PC 1	NA	2001::2 / 64	2001::1	Switch1 F0/1
Server0	NA	2005::2 / 64	2005::1	Switch2 F0/1
Router0	GE0/0	2002::1 / 64	NA	Switch0 F0/5
	GE0/1	2001::1 / 64	NA	Switch1 F0/5
	S0/1/0	2003::1 / 64	NA	NA
Router1	S0/1/0	2003::1 / 64	NA	NA
	S0/1/1	2004::1 / 64	NA	NA
Router2	S0/1/1	2004::2 / 64	NA	NA
	GE0/0	2005::1 / 64	NA	Switch2 F0/5

Configuring PC0



Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IP Address:

Subnet Mask:

Default Gateway: 0.0.0.0

DNS Server: 0.0.0.0

IPv6 Configuration

○ DHCP ○ Auto Config ● Static

IPv6 Address: 2002::2 / 64

Link Local Address: FE80::200:CFF:FE91:556E

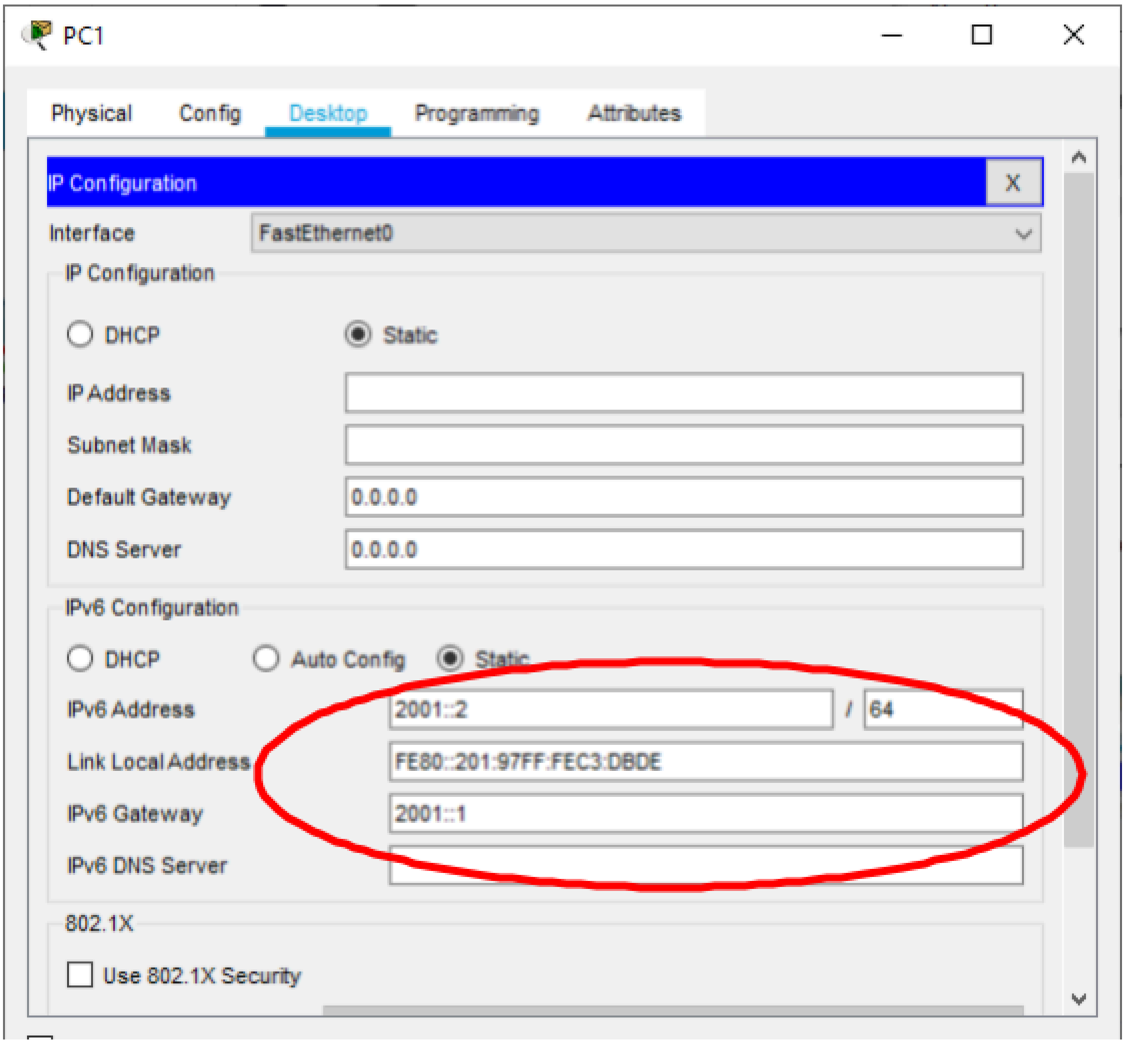
IPv6 Gateway: 2002::1

IPv6 DNS Server:

802.1X

□ Use 802.1X Security

Configuring PC1



Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IP Address:

Subnet Mask:

Default Gateway: 0.0.0.0

DNS Server: 0.0.0.0

IPv6 Configuration

○ DHCP ○ Auto Config ● Static

IPv6 Address: 2001::2 / 64

Link Local Address: FE80::201:97FF:FEC3:DBDE

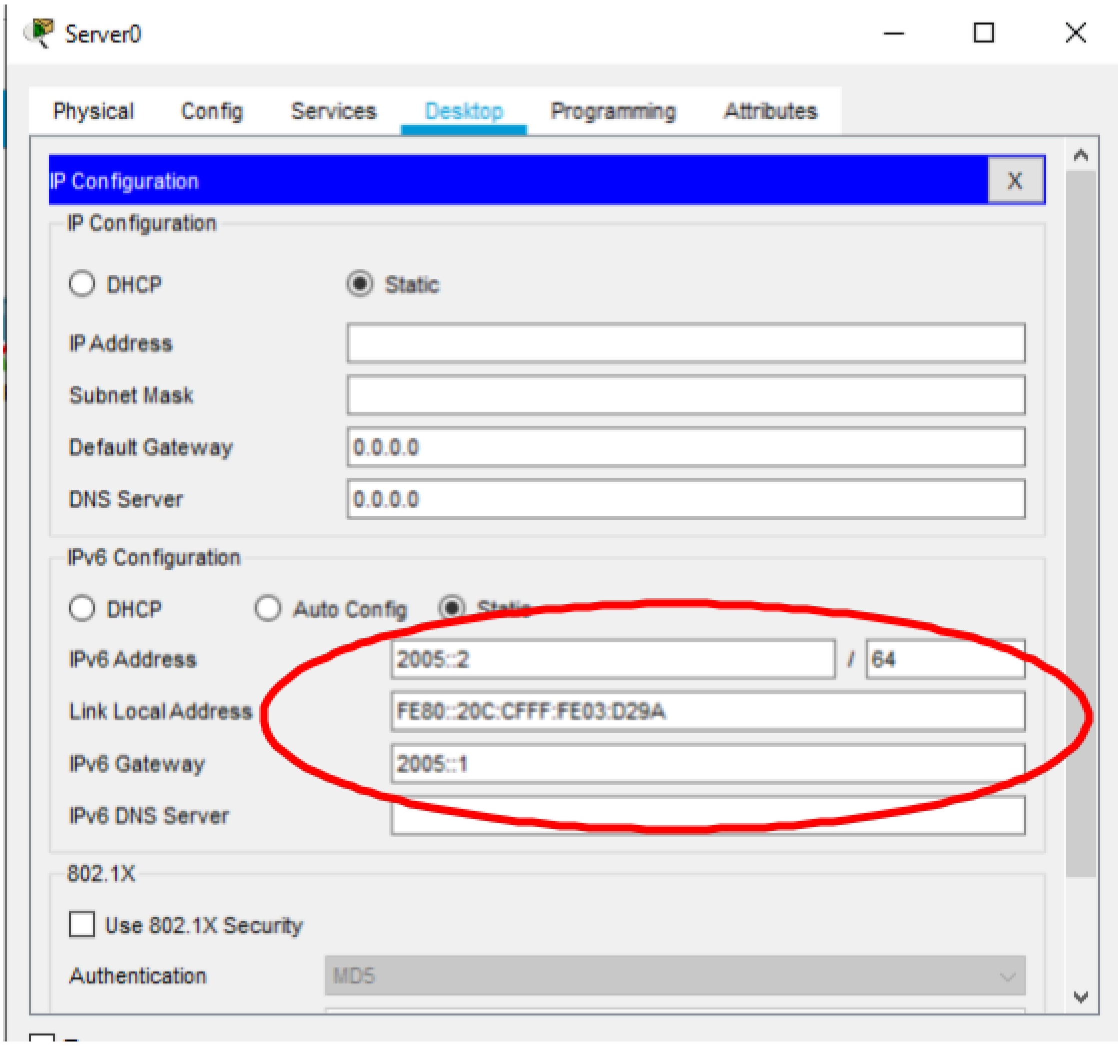
IPv6 Gateway: 2001::1

IPv6 DNS Server:

802.1X

□ Use 802.1X Security

Configuring Server0



Server0

IP Configuration

IP Configuration

○ DHCP ● Static

IP Address

Subnet Mask

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

○ DHCP ○ Auto Config ● Static

IPv6 Address 2005::2 / 64

Link Local Address FE80::20C:CFFF:FE03:D29A

IPv6 Gateway 2005::1

IPv6 DNS Server

802.1X

□ Use 802.1X Security

Authentication MD5

For setting the ipv6 addresses we need to use the CLI mode for each Router as follows

Configuring Router0

```
Router>
Router>enable
Router#
Router#configure terminal
Router(config)#ipv6 unicast-routing

Router(config)#interface
GigabitEthernet0/0 Router(config-if)#ipv6
address 2002::1/64 Router(config-if)#ipv6
rip a enable Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#

Router(config)#interface
GigabitEthernet0/1 Router(config-if)#ipv6
address 2001::1/64 Router(config-if)#ipv6
rip a enable Router(config-if)#no shutdown
Router(config-if)#exit
 Router(config)#

Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address
2003::1/64 Router(config-if)#ipv6 rip a
enable Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

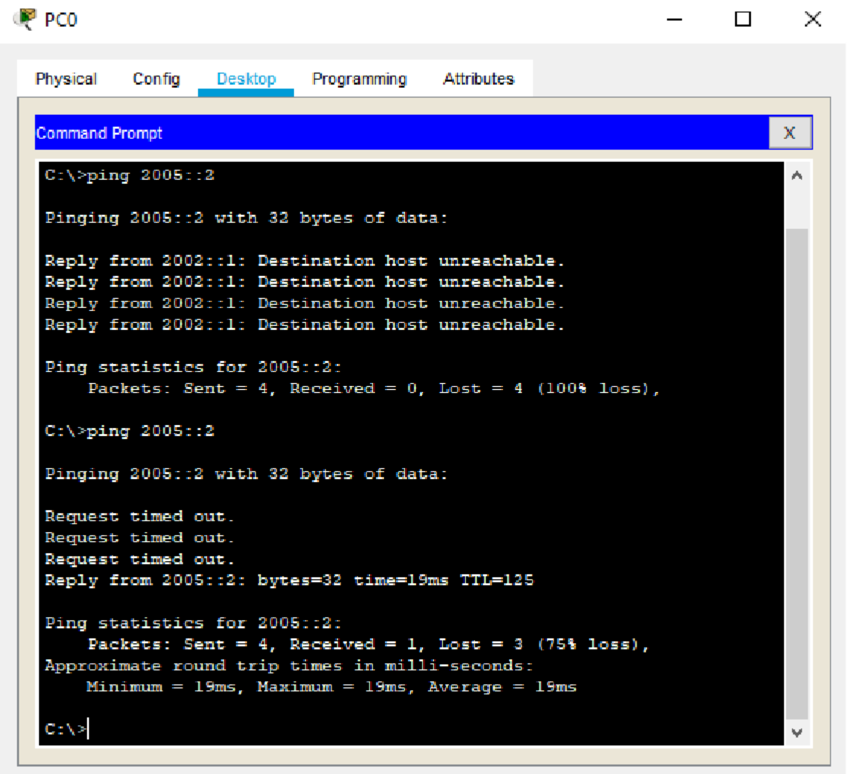
Configuring Router1

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing
Router(config)#

Router(config)#interface Serial0/1/0
Router(config-if)#ipv6 address
2003::1/64 Router(config-if)#ipv6 rip a
enable Router(config-if)#no shutdown
Router(config-if)#
Router(config-if)#exit
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

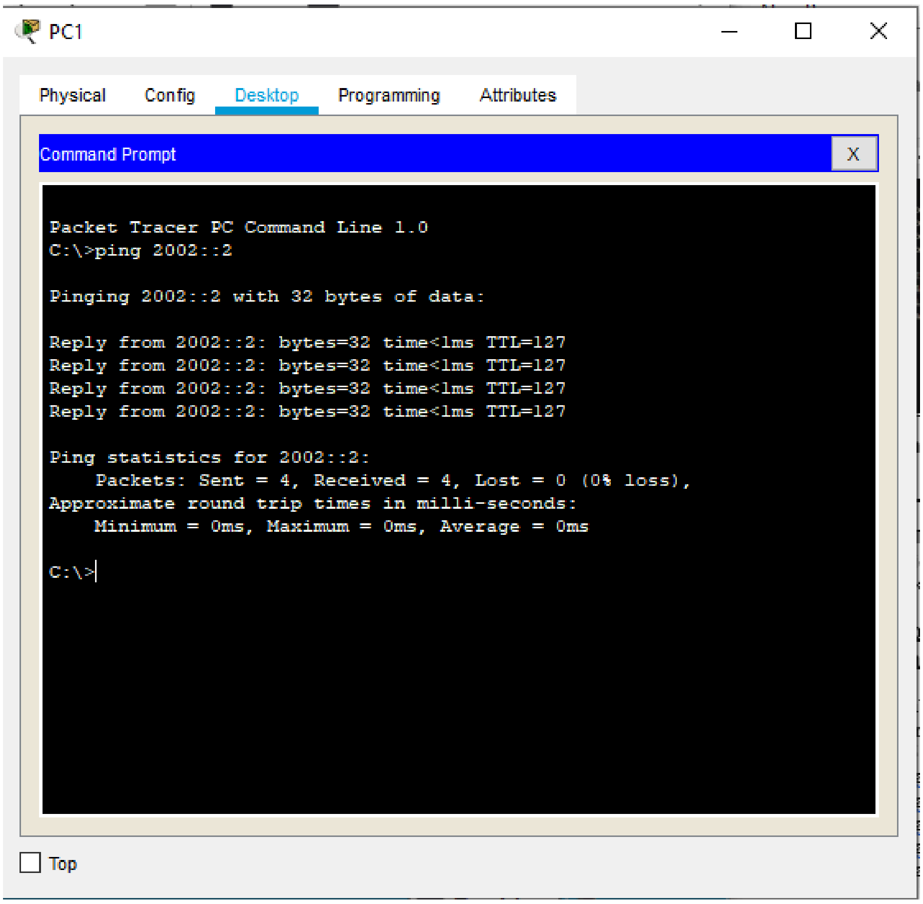
Configuring Router2

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 address 2004::2/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit

Router(config)#interface GigabitEthernet0/0
Router(config-if)#ipv6 address 2005::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Check the connectivity by pinging from PCs to Server



```
C:\>ping 2005::2

Pinging 2005::2 with 32 bytes of data:

Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.
Reply from 2002::1: Destination host unreachable.

Ping statistics for 2005::2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 2005::2

Pinging 2005::2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Reply from 2005::2: bytes=32 time=19ms TTL=125

Ping statistics for 2005::2:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 19ms, Average = 19ms

C:\>
```



```
Packet Tracer PC Command Line 1.0
C:\>ping 2002::2

Pinging 2002::2 with 32 bytes of data:

Reply from 2002::2: bytes=32 time<1ms TTL=127
Reply from 2002::2: bytes=32 time<1ms TTL=127
Reply from 2002::2: bytes=32 time<1ms TTL=127
Reply from 2002::2: bytes=32 time<1ms TTL=127

Ping statistics for 2002::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

And we see that the connectivity is established

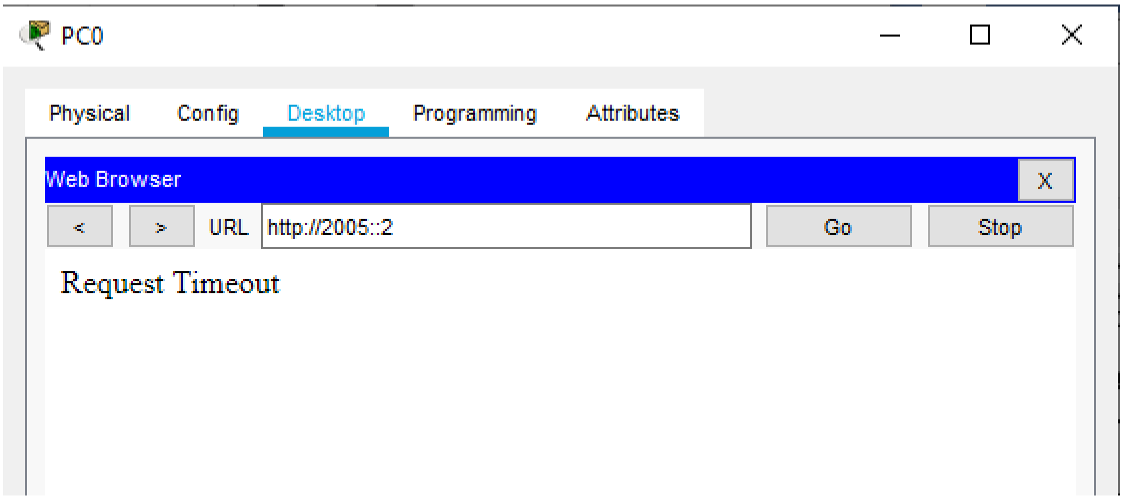
We configure the ACL and apply it to the Router1 with the following conditions

1) No HTTP or HTTPS allowed on server by any host
2) No www service accessible on the server by any host
3) Only ipv6 packets allowed towards the server

We enter the following commands in the CLI mode of the Router1 and Router2, apply it at the proper interface

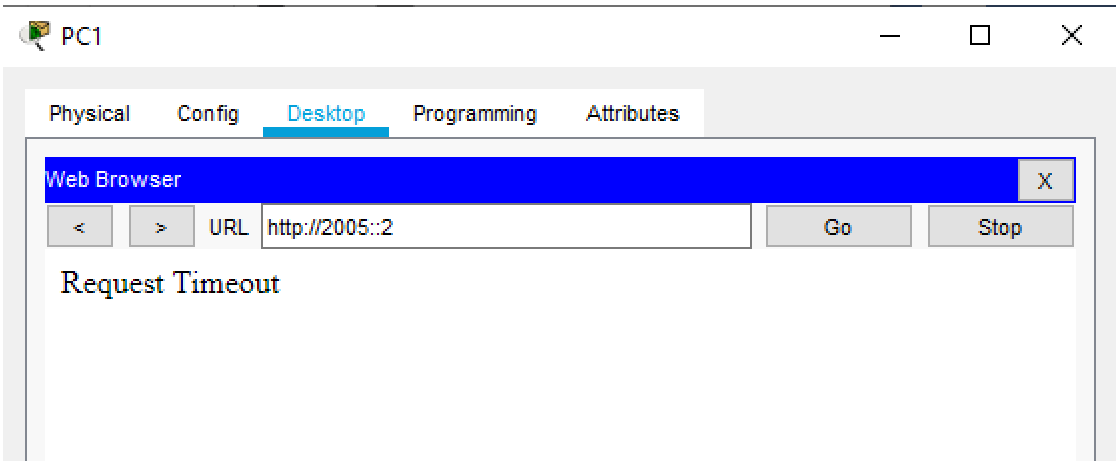
```
Router>
Router>enable
Router#configure terminal
Router(config)#ipv6 access-list smile
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq www
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq 443
Router(config-ipv6-acl)#permit ipv6 any any
Router(config-ipv6-acl)# Router(config-ipv6-acl)#exit

Router(config)# Router(config)#interface Serial0/1/1
Router(config-if)#ipv6 traffic-filter smile in
Router(config-if)#exit
Router(config)#
```

We verify the configuration by first accessing the www service from the browser of both PCs and get failure



http://2005::2

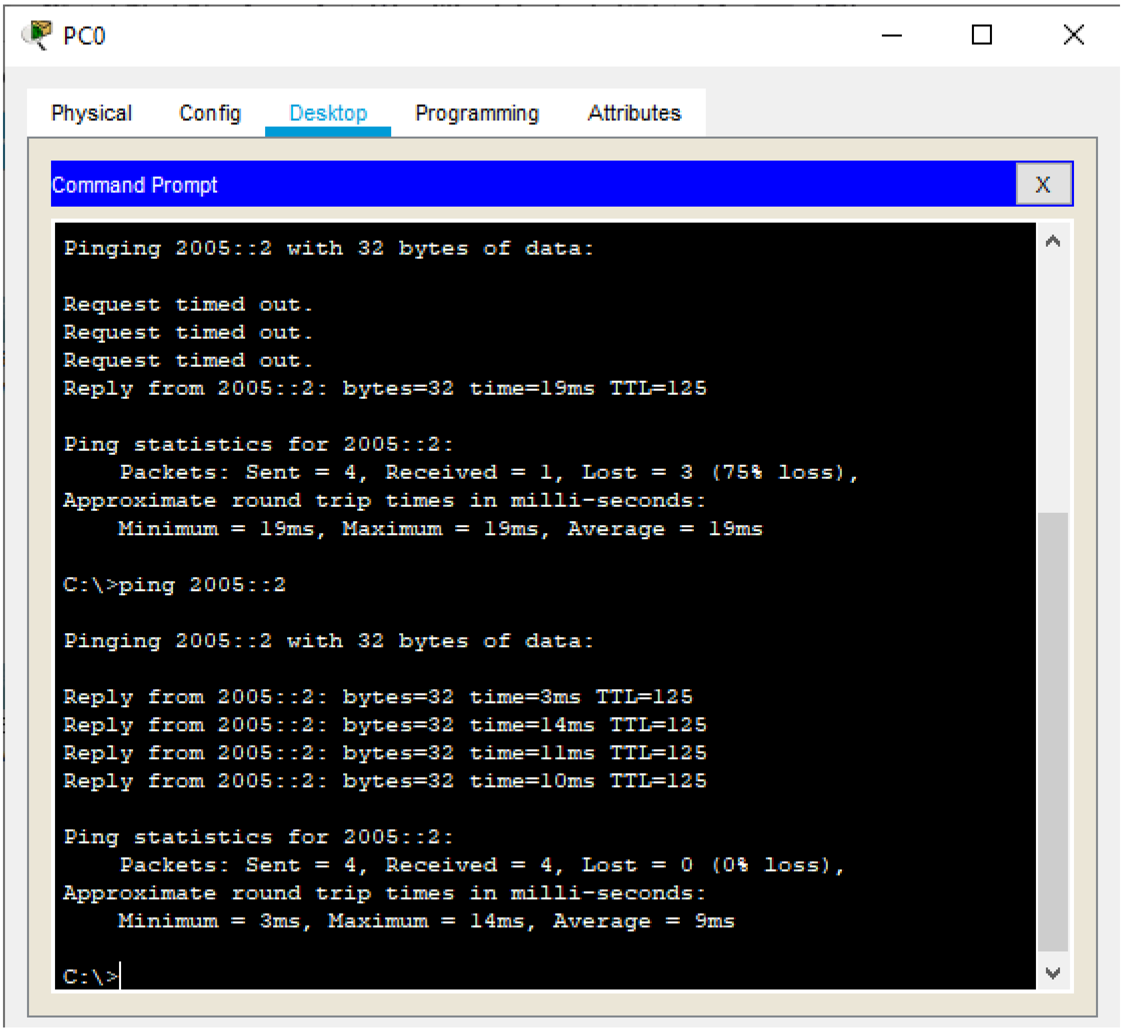
Request Timeout



http://2005::2

Request Timeout

Next we verify whether the ipv6 protocol works by pinging server from any of the PC (it must be successful)



```
Pinging 2005::2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Reply from 2005::2: bytes=32 time=19ms TTL=125

Ping statistics for 2005::2:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 19ms, Average = 19ms

C:\>ping 2005::2

Pinging 2005::2 with 32 bytes of data:

Reply from 2005::2: bytes=32 time=3ms TTL=125
Reply from 2005::2: bytes=32 time=14ms TTL=125
Reply from 2005::2: bytes=32 time=11ms TTL=125
Reply from 2005::2: bytes=32 time=10ms TTL=125

Ping statistics for 2005::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 14ms, Average = 9ms

C:\>
```

Hence the given ACLs have been applied and verified on host running on ipv6 protocol.