PRACTICAL NO 6: Configuring a Zone-Based Policy Firewall (ZPF)

Cisco IOS® Software Release 12.4(6)T introduced Zone-Based Policy Firewall (ZFW), a new configuration model for the Cisco IOS Firewall feature set. This new configuration model offers intuitive policies for multiple-interface routers, increased granularity of firewall policy application, and a default deny-all policy that prohibits traffic between firewall security zones until an explicit policy is applied to allow desirable traffic.

Nearly all classic Cisco IOS Firewall features implemented before Cisco IOS Software Release 12.4(6)T are supported in the new zone-based policy inspection interface:

1) Stateful packet inspection
2) VRF-aware Cisco IOS Firewall
3) URL filtering
4) Denial-of-Service (DoS) mitigation

Cisco IOS Software Release 12.4(9)T added ZFW support for per-class session/connection and throughput limits, as well as application inspection and control:

1) HTTP
2) Post Office Protocol (POP3),
3) Internet Mail Access Protocol (IMAP),
4) Simple Mail Transfer Protocol/Enhanced Simple Mail Transfer Protocol (SMTP/ESMTP) 5) Sun Remote Procedure Call (RPC)
6) Instant Messaging (IM) applications:
   i) Microsoft Messenger ii) Yahoo! Messenger iii) AOL Instant Messenger
7) Peer-to-Peer (P2P) File Sharing:
   i) Bittorrent ii) KaZaA
   iii) Gnutella
   iv) eDonkey

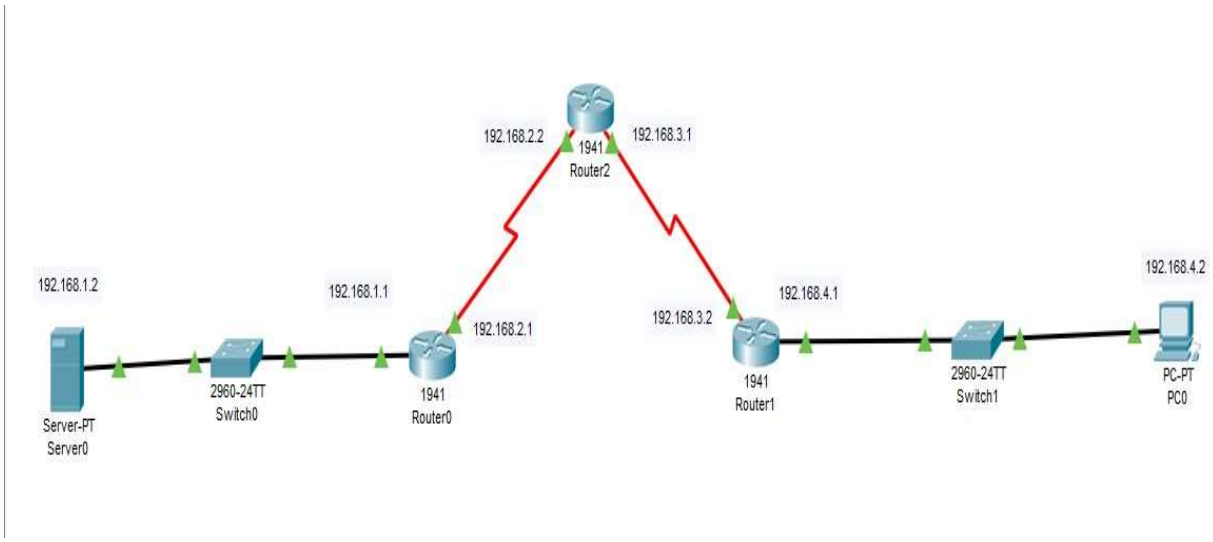
Cisco IOS Software Release 12.4(11)T added statistics for easier DoS protection tuning.

Some Cisco IOS Classic Firewall features and capabilities are not yet supported in a ZFW in Cisco IOS Software Release 12.4(15)T:

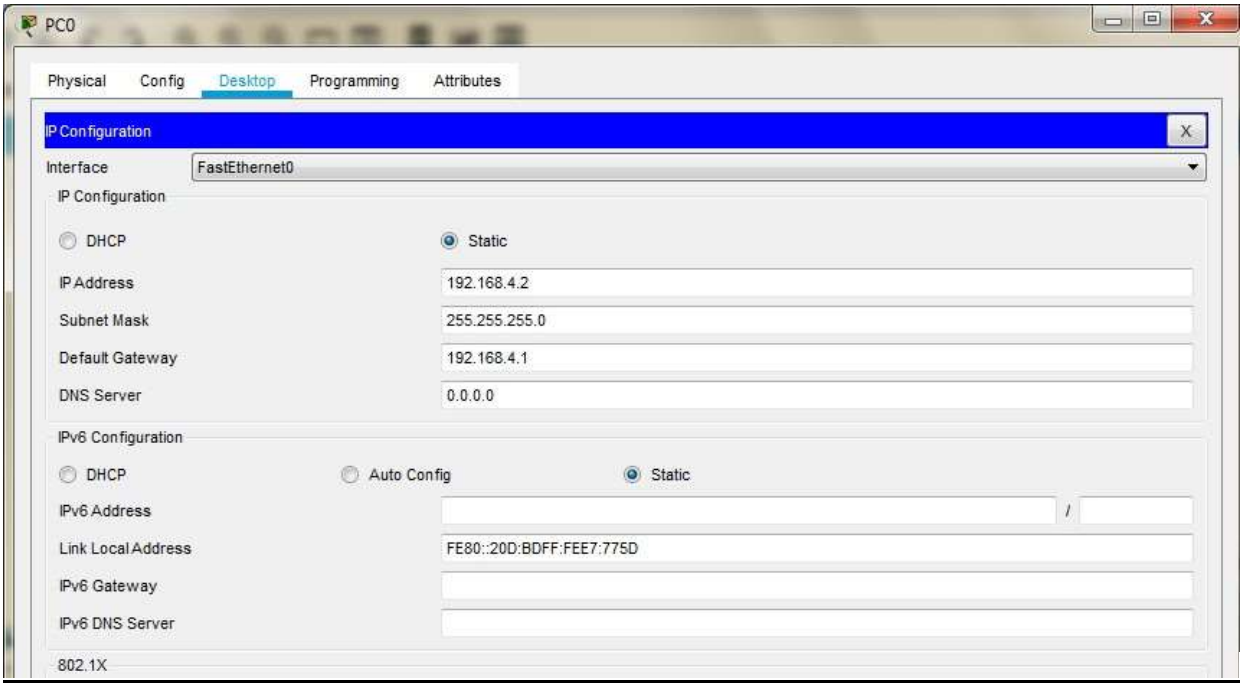
i) Authentication proxy ii) Stateful firewall failover iii) Unified firewall MIB iv) IPv6 stateful inspection
v) TCP out-of-order support

ZFW generally improves Cisco IOS performance for most firewall inspection activities. Neither Cisco IOS ZFW or Classic Firewall include stateful inspection support for multicast traffic.

We use the following Topology for the current case

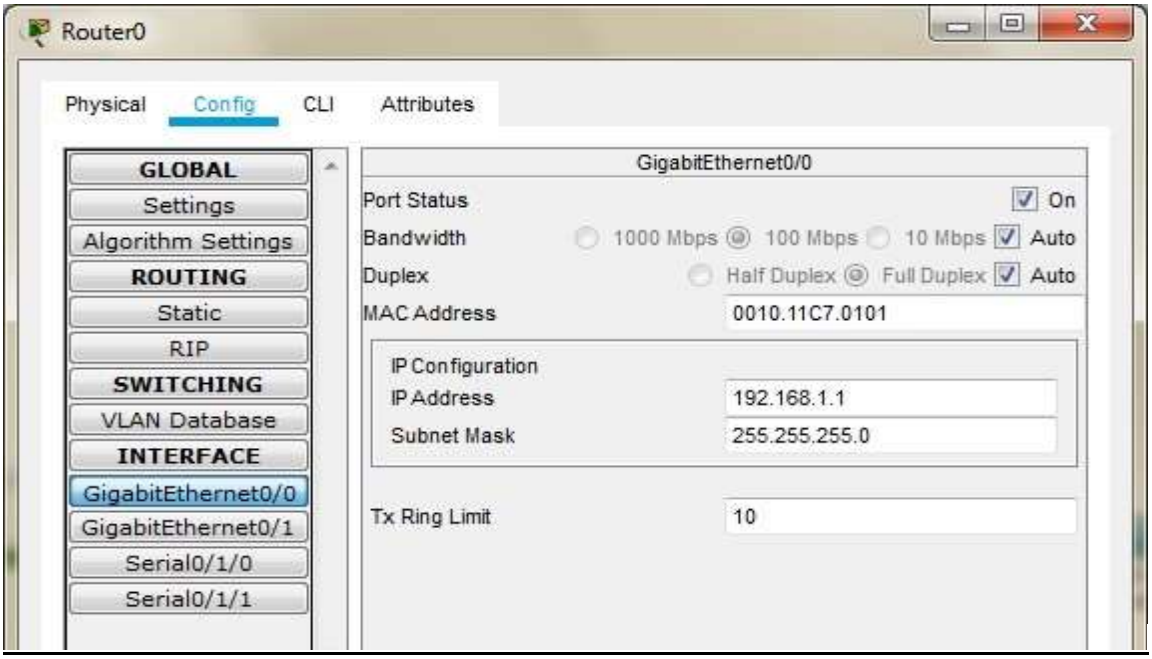


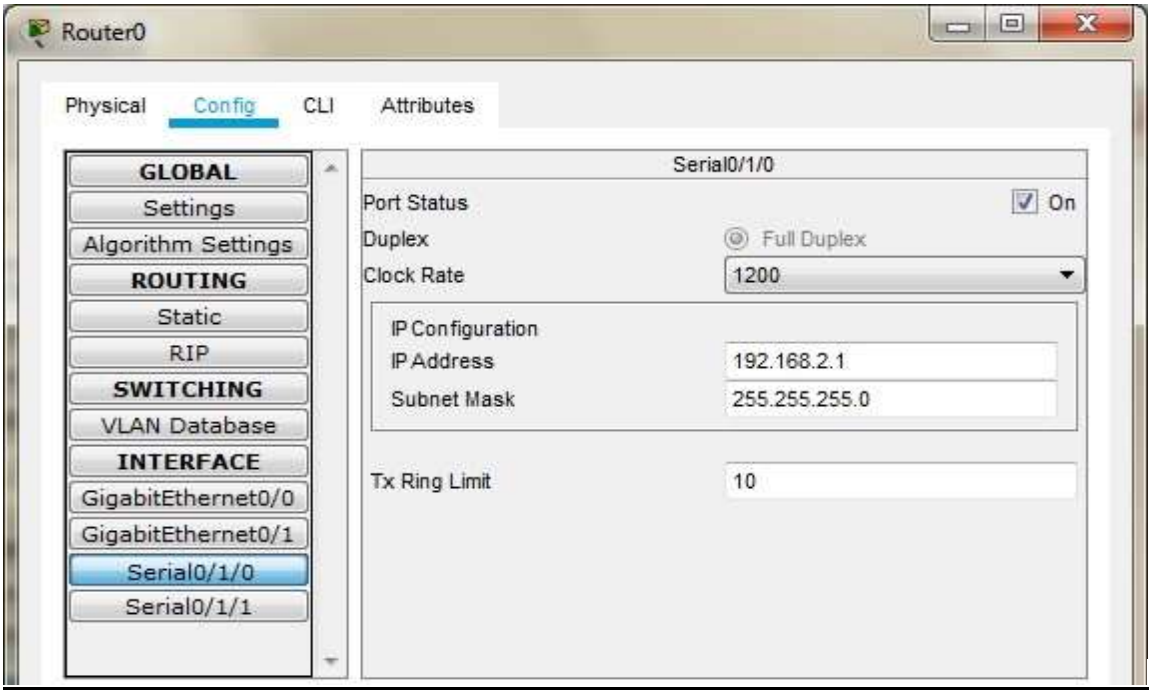
Configuring PC0



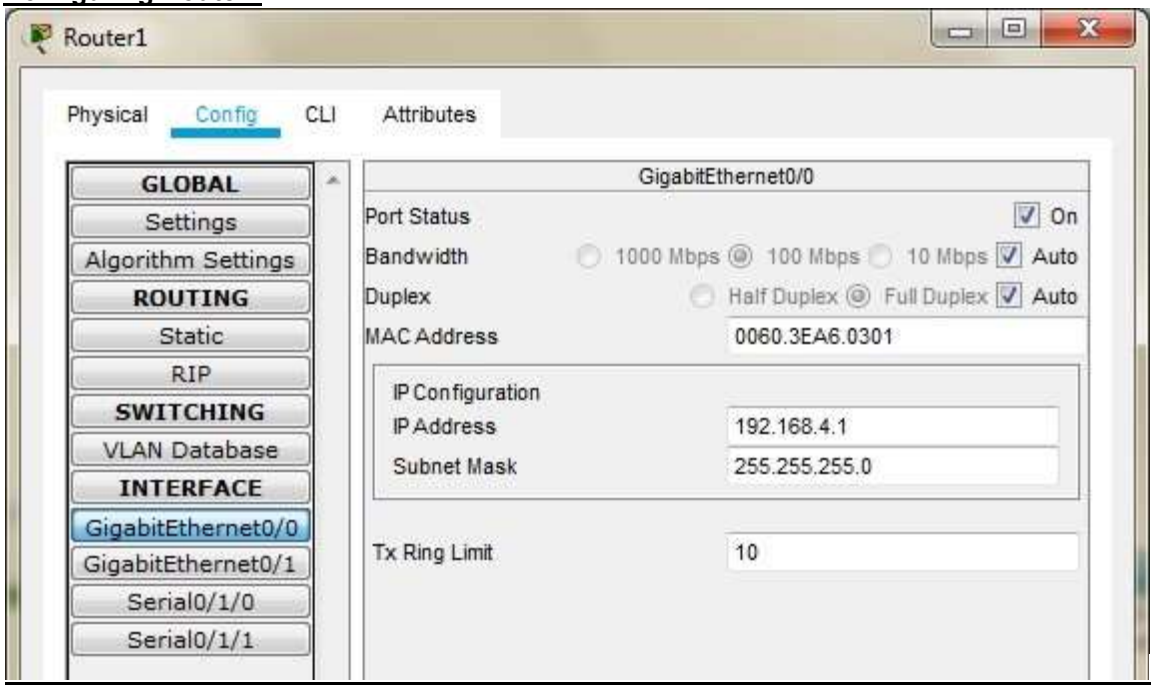
Serial Interface must be added in each Router before configuring it

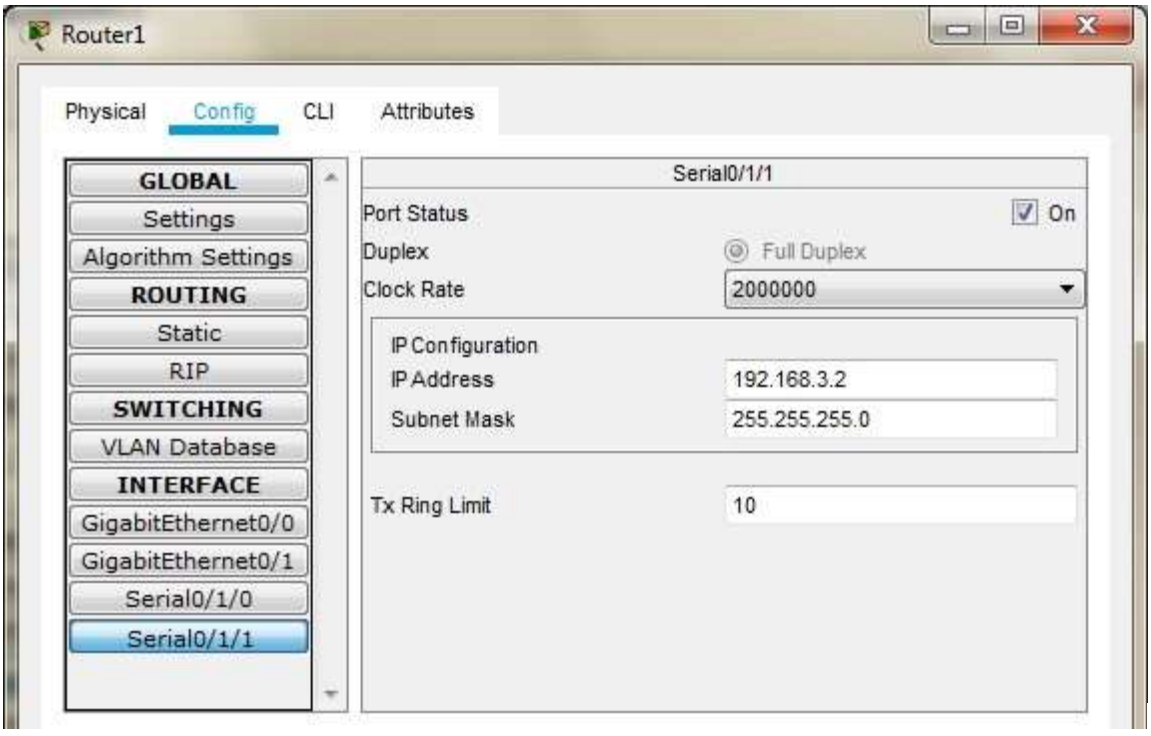
Configuring Router0



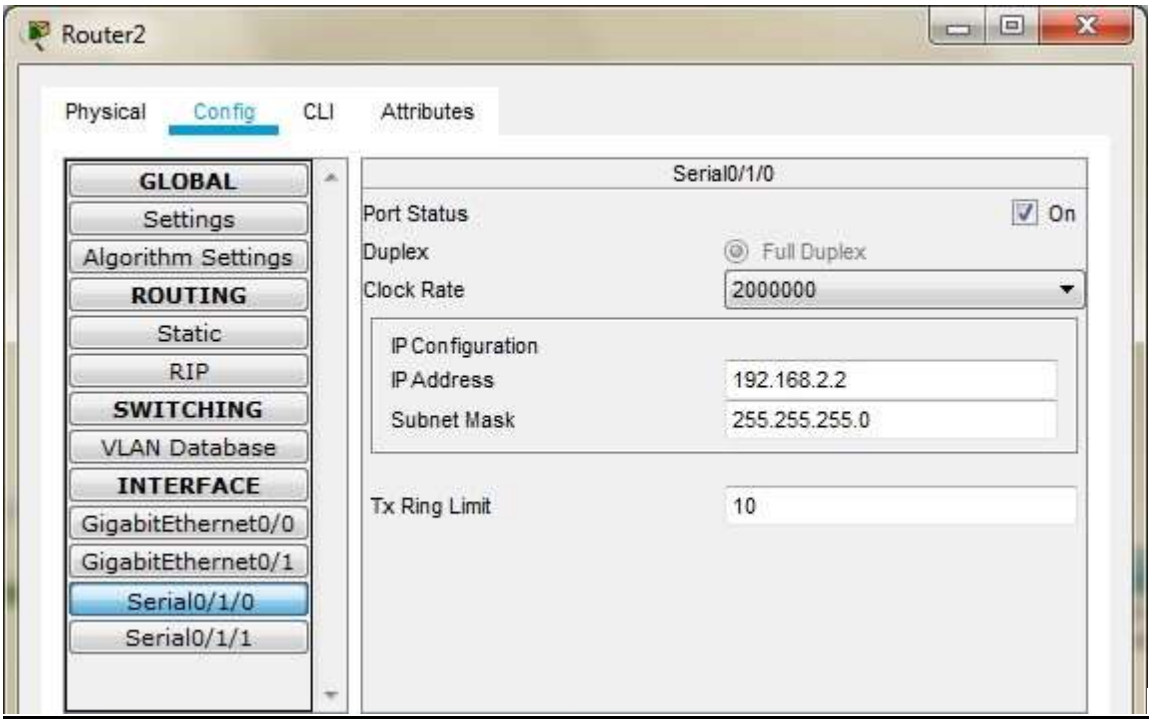


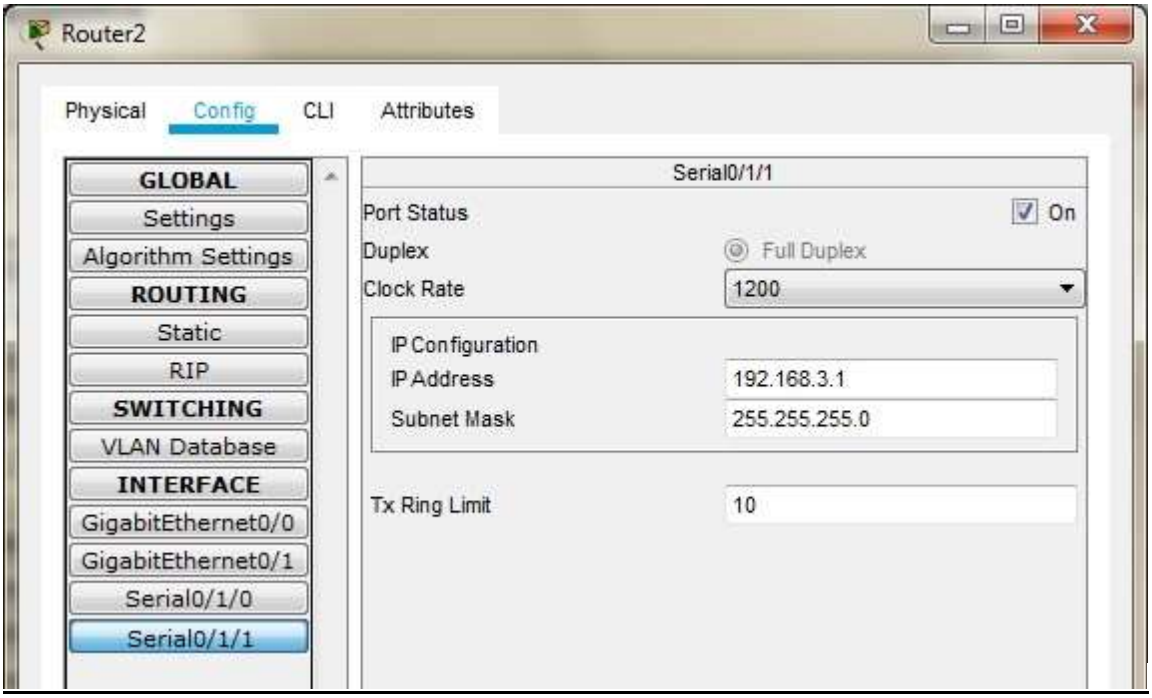
Configuring Router1



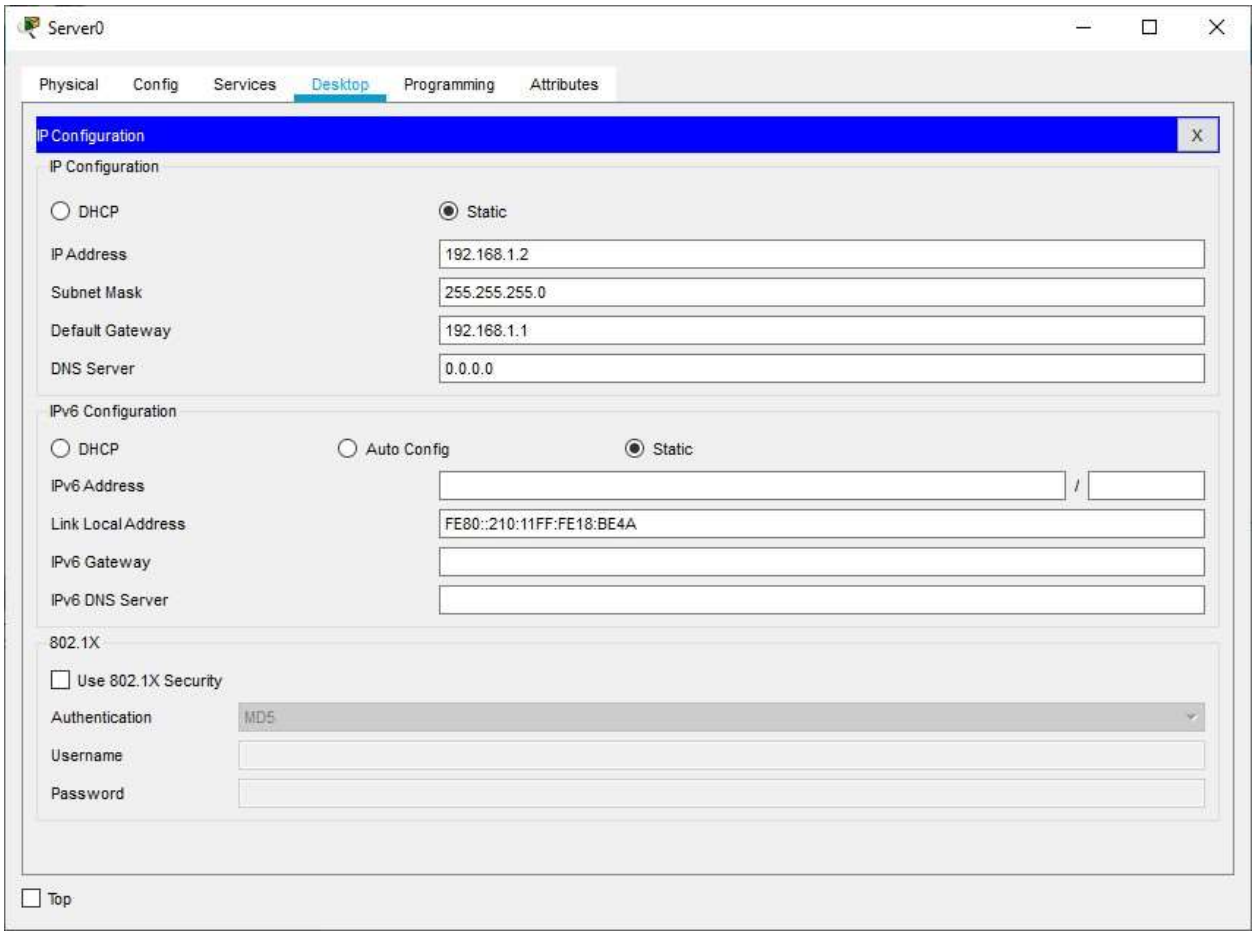


Configuring Router2





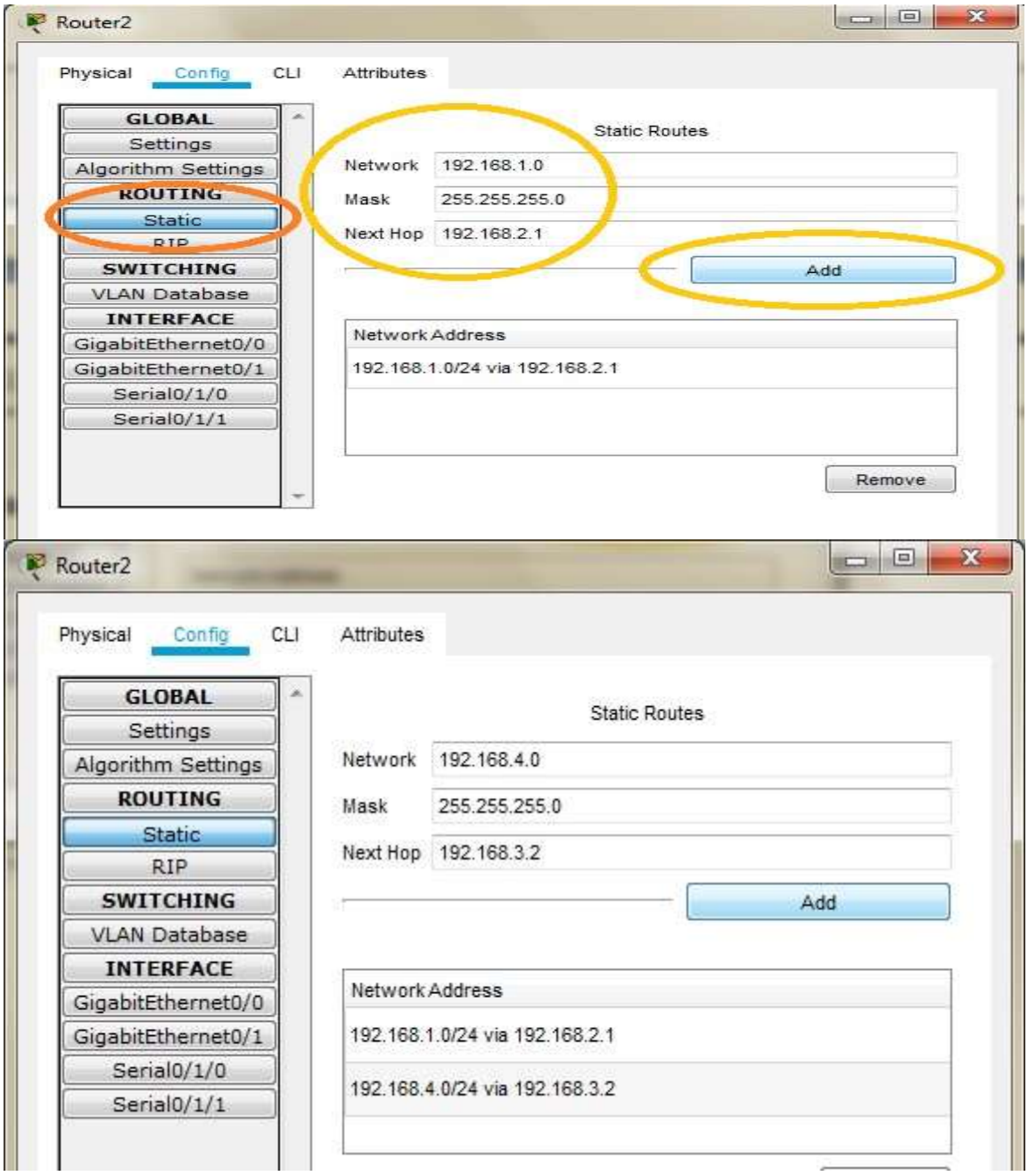
Configuring Server0



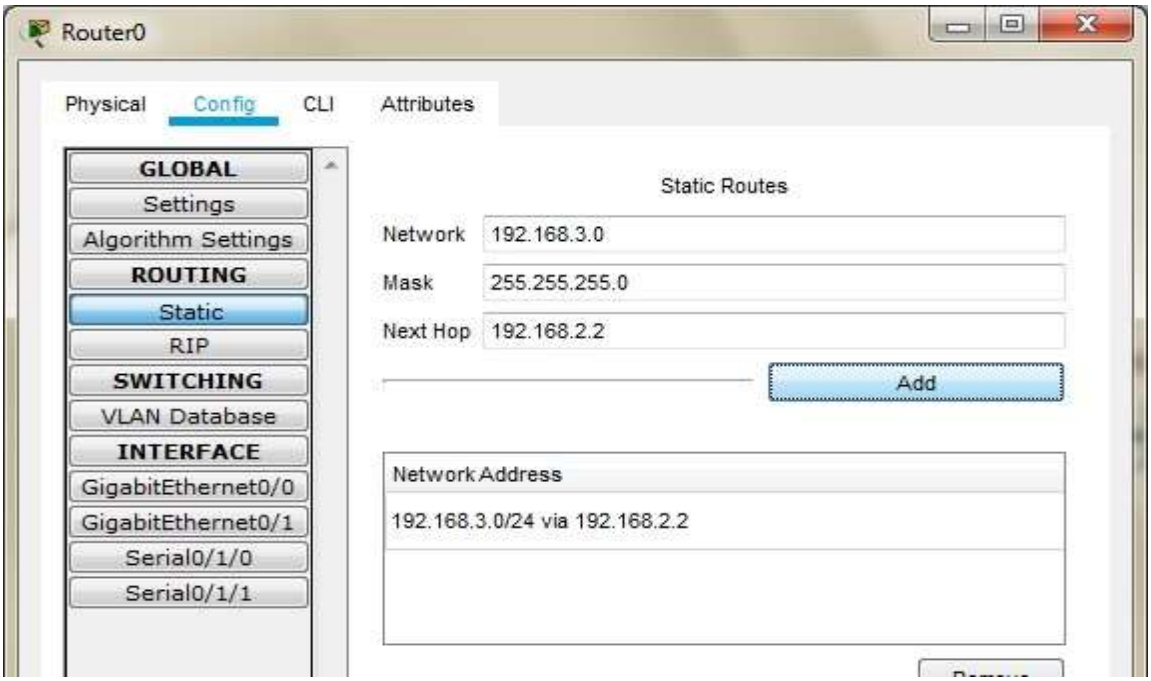
Part 1: Static Routing

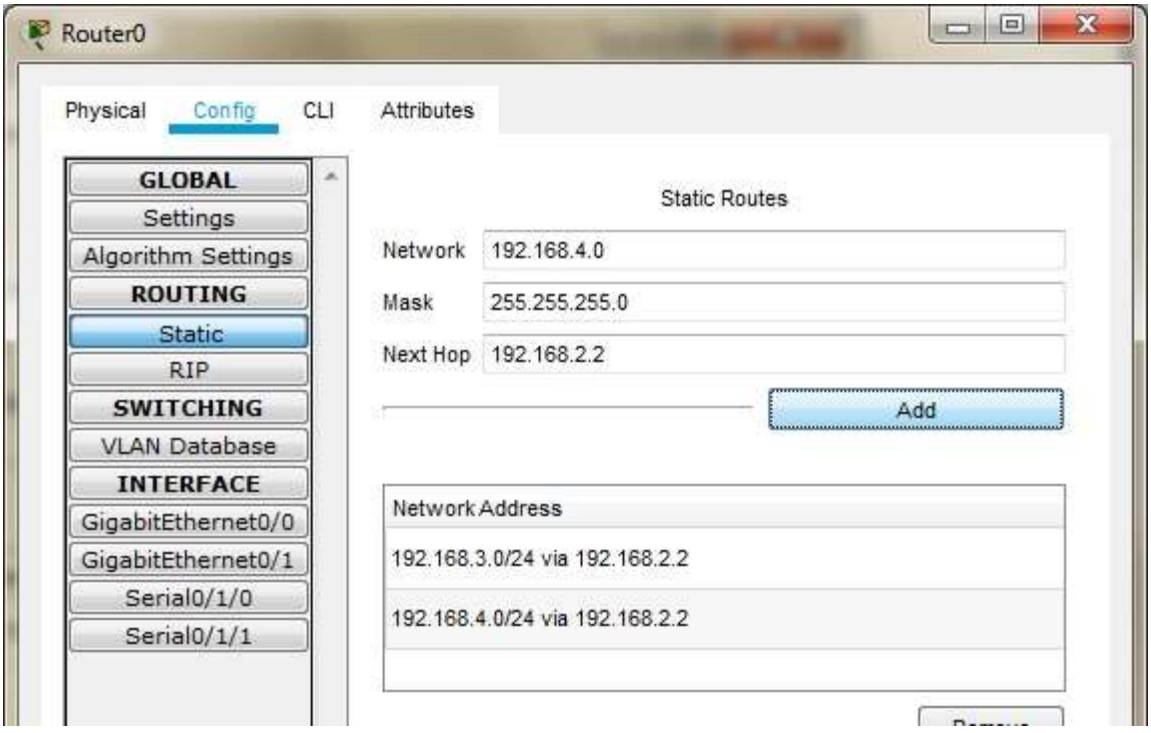
Static Routing is done using the following procedure for each Router

Router 2: Add the following Routes in the Static mode

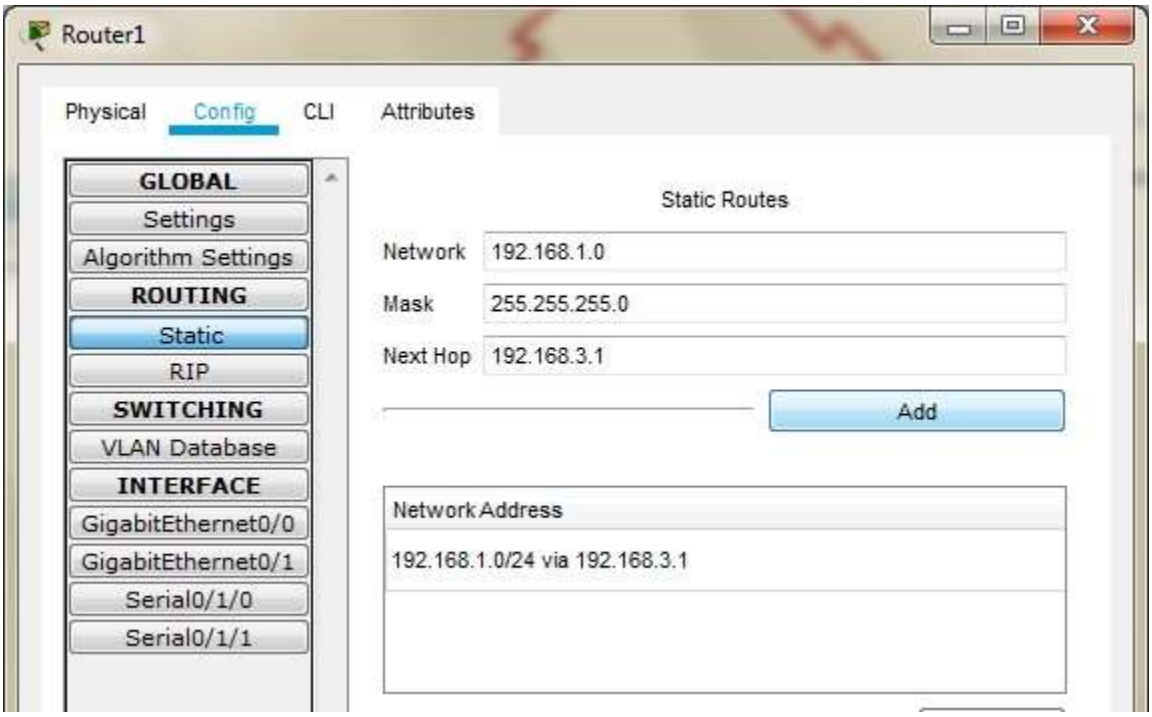


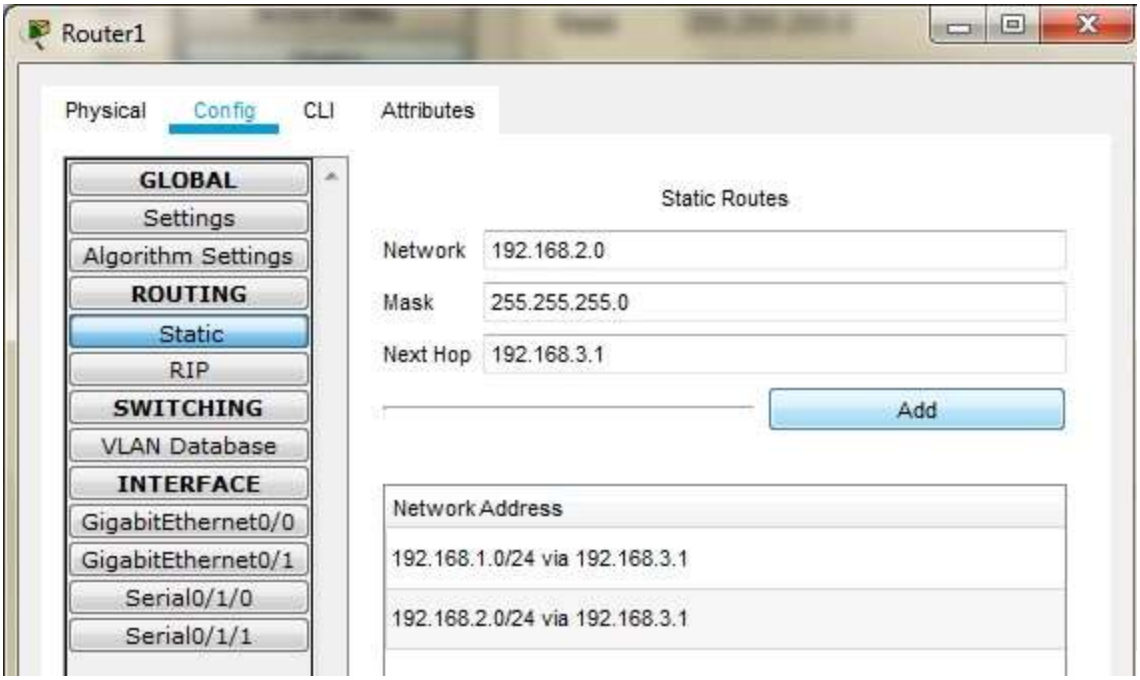
Router 0: Add the following Routes in the Static mode



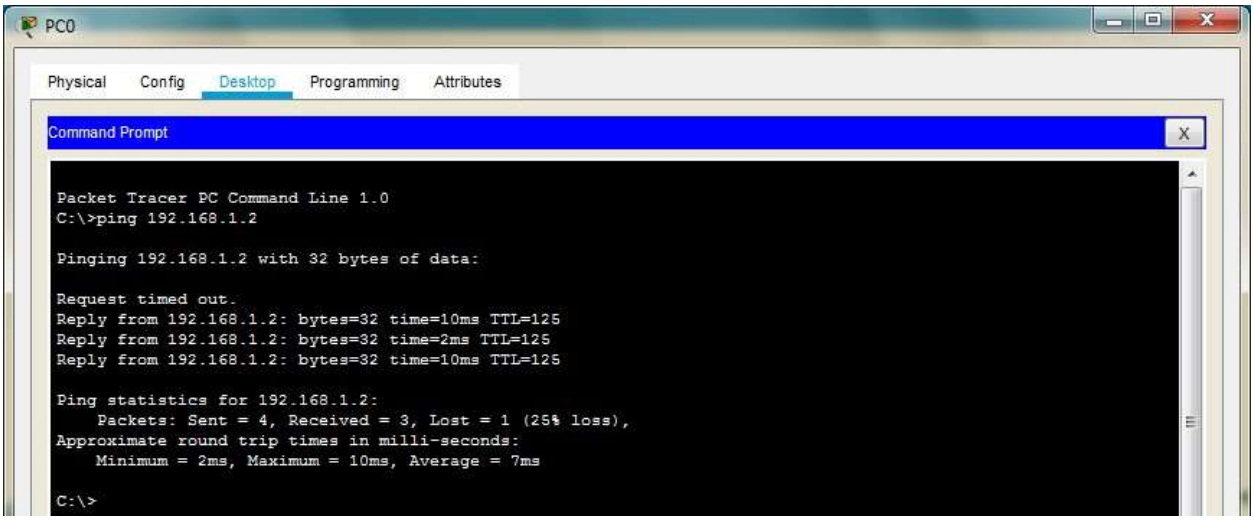


Router 1: Add the following Routes in the Static mode


Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Static Routes
Network 192.168.1.0
Mask 255.255.255.0
Next Hop 192.168.3.1
Add
Network Address
192.168.1.0/24 via 192.168.3.1


Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Static Routes
Network 192.168.2.0
Mask 255.255.255.0
Next Hop 192.168.3.1
Add
Network Address
192.168.1.0/24 via 192.168.3.1
192.168.2.0/24 via 192.168.3.1

Now we check the connectivity by pinging the Server from the PC



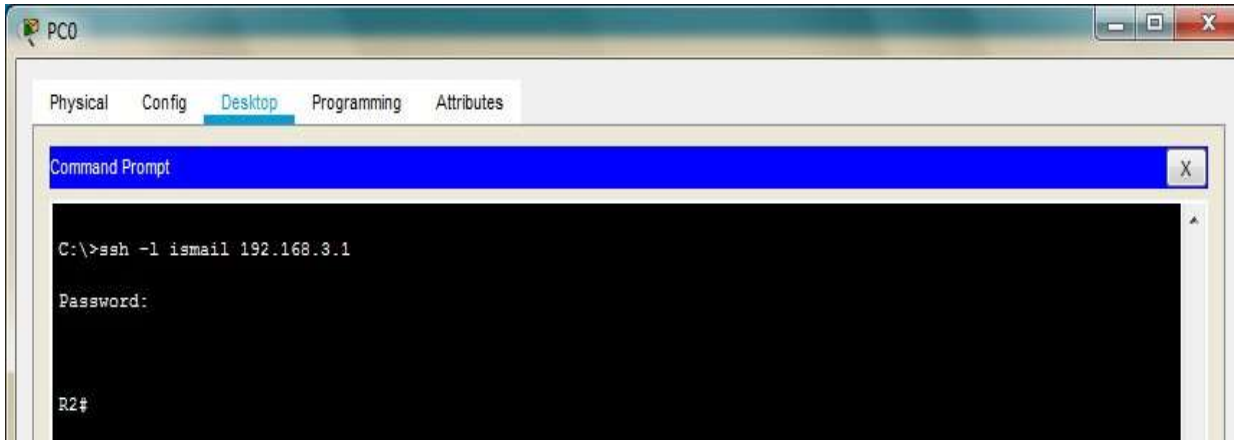
Part 2: Configuring SSH on Router 2

Type the following commands in the CLI mode of Router2

Router>en
Router>enable
Router#configure terminal
Router(config)#ip domain-name ismail.com
Router(config)#hostname R2
R2(config)#cryp
R2(config)#crypto k
R2(config)#crypto key g
R2(config)#crypto key generate r
R2(config)#crypto key generate rsa

R2(config)#line vty 0 4
R2(config-line)#transport input ssh
R2(config-line)#login local
R2(config-line)#exit
R2(config)#username ismail privilege 15 password cisco

Now we verify the SSH using PC as follows



Next we access the web services of the Server using the web browser of PC using the following

Part 3: Create the Firewall Zones on Router1

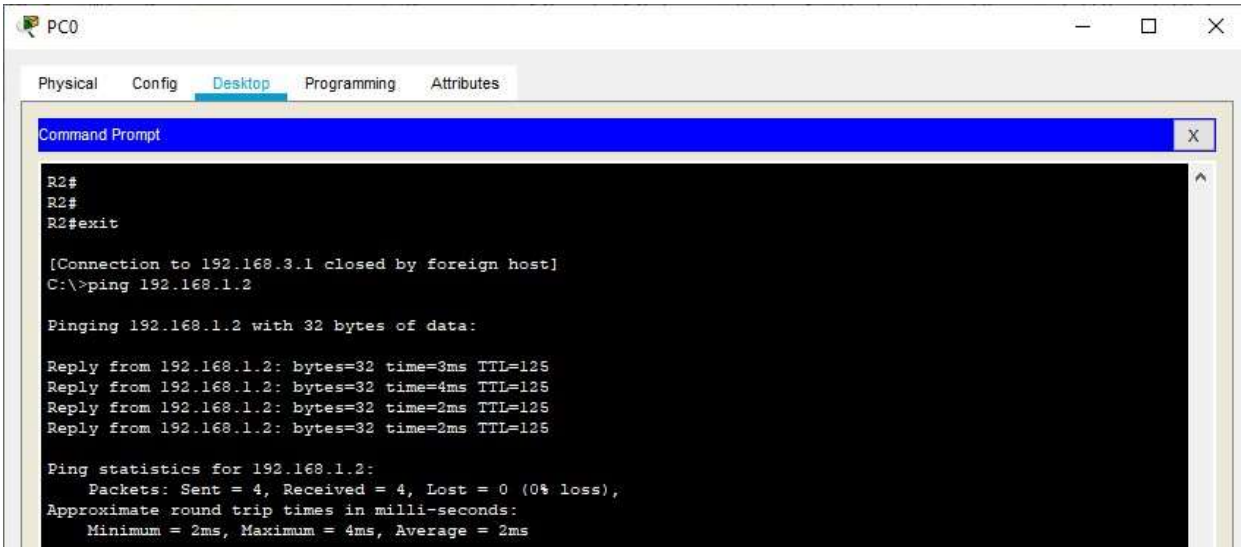
Type the following commands in the CLI mode of Router1

```
Router#
Router#configure terminal

Router(config)#zone security in-zone
Router(config-sec-zone)#exit

Router(config)#zone security out-zone
Router(config-sec-zone)#exit

Router(config)#access-list 101 permit ip 192.168.4.0 0.0.0.255 any
Router(config)#class-map type inspect match-all in-map
Router(config-cmap)#match access-group 101
Router(config-cmap)#exit

Router(config)#policy-map type inspect in-out
Router(config-pmap)#class type inspect in-map
Router(config-pmap-c)#inspect
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)#

Router(config)#zone-pair security in-out-zone source in-zone destination out-zone
Router(config-sec-zone-pair)#service-policy type inspect in-out
Router(config-sec-zone-pair)#exit
Router(config)#

Router(config)#interface GigabitEthernet0/0
Router(config-if)#zone-member security in-zone
Router(config-if)#exit
Router(config)#

Router(config)#interface Serial0/1/1
Router(config-if)#zone-member security out-zone
Router(config-if)#exit
Router(config)#exit
```

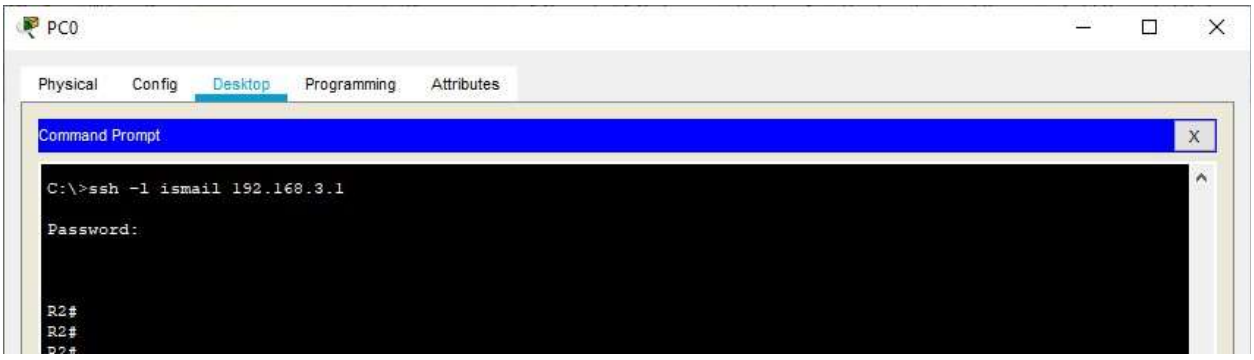
Router#copy running-config startup-config

Part 4: Testing the Firewall Functionality (from in-zone to out-zone)by the following steps

Step 1: Pinging SERVER from the PC (it will succeed)



Step 2: Start an SSH session from PC to Router 2 (ip 192.168.1.2)

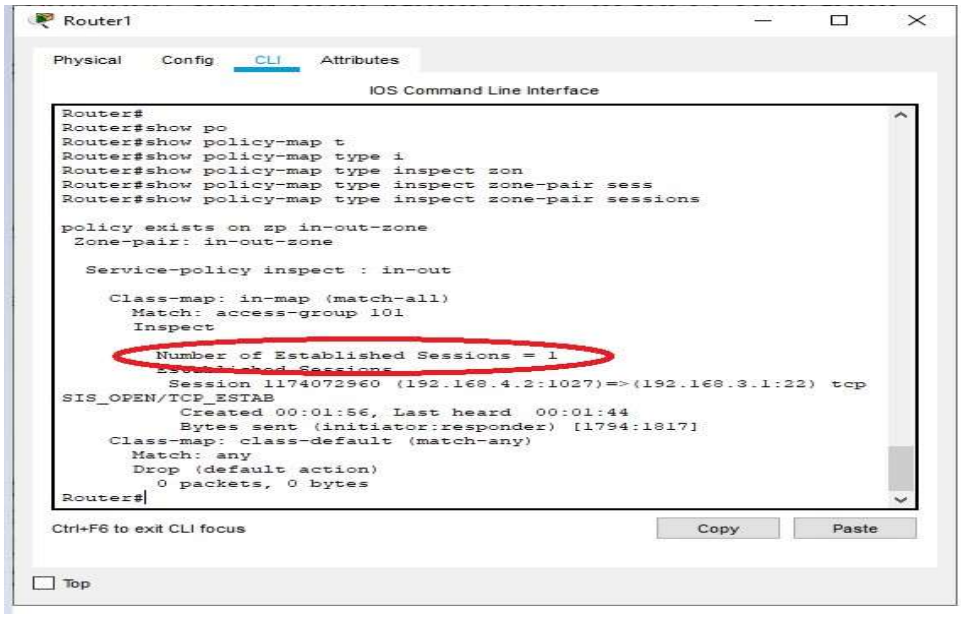


As seen above the session becomes active and we get access to Router2 (Do not exit and the session and continue to Step 3)

Step 3: Type the following command in the CLI mode of Router1

Router#show policy-map type inspect zone-pair sessions

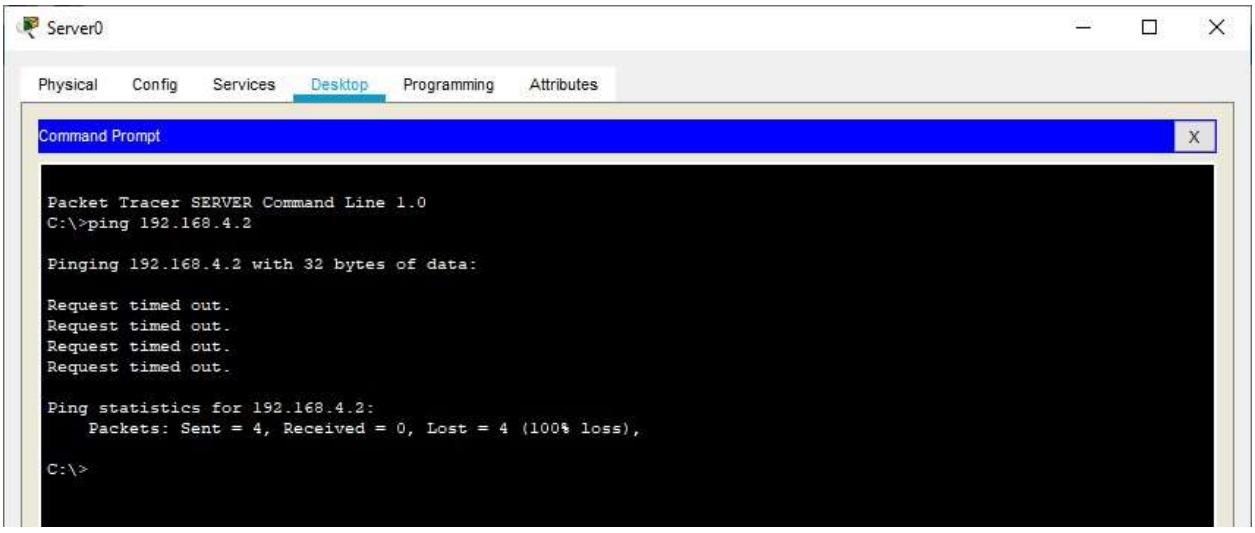
We will get the following output



Step 4: We close the SSH connection and open the web browser and access the server address (192.168.1.2) and get the following

Part 5: Testing the Firewall Functionality (from out-zone to in-zone)by the following steps

Step 1: Ping PC0 from the SERVER (it will result in Failure)



Hence the Firewall functionality has been verified

PRACTICAL NO 7: Configure IOS Intrusion Prevention System (IPS) Using the CLI

The Cisco IOS IPS acts as an in-line intrusion prevention sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When it detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. The Signature Event Action Processor (SEAP) can dynamically control actions that are to be taken by a signature event on the basis of parameters such as fidelity, severity, or target value rating. These parameters have default values but can also be configured through CLI. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

1) Send an alarm to a syslog server or a centralized management interface
2) Drop the packet
3) Reset the connection
4) Deny traffic from the source IP address of the attacker for a specified amount of time
5) Deny traffic on the connection for which the signature was seen for a specified amount of time

Cisco developed its Cisco IOS software-based intrusion-prevention capabilities and Cisco IOS Firewall with flexibility in mind, so that individual signatures could be disabled in case of false positives. Generally, it is preferable to enable both the firewall and Cisco IOS IPS to support network security policies. However, each of these features may be enabled independently and on different router interfaces.

Signatures:

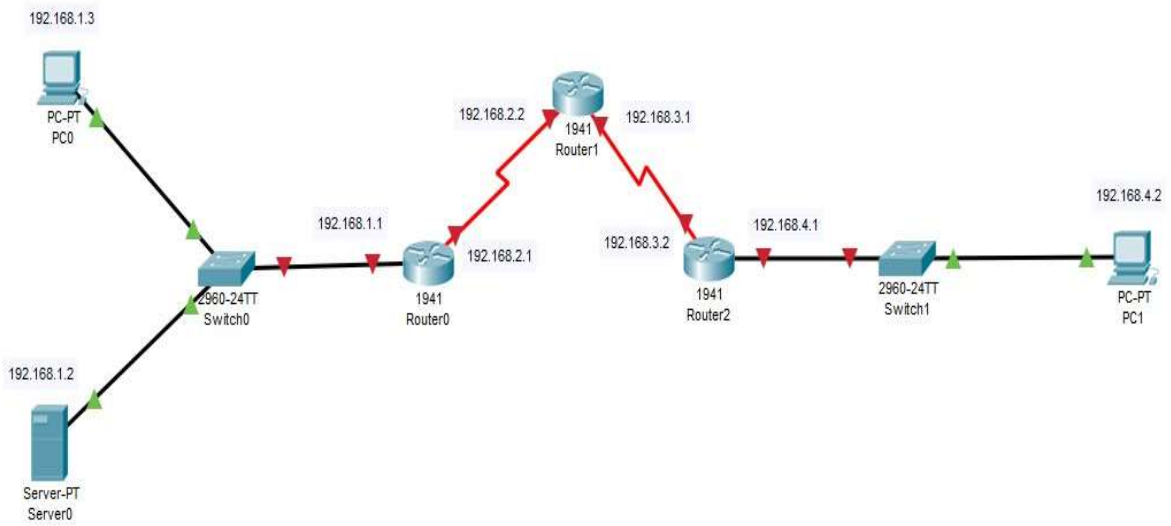
A signature is a set of rules that an IDS and an IPS use to detect typical intrusive activity, such as DoS attacks. We can easily install signatures using IDS and IPS management software such as Cisco IDM. Sensors enables us to modify existing signatures and define new ones. As sensors scan network packets, they use signatures to detect known attacks and respond with predefined actions. A malicious packet flow has a specific type of activity and signature, and an IDS or IPS sensor examines the data flow using many different signatures. When an IDS or IPS sensor matches a signature with a data flow, the sensor takes action, such as logging the event or sending an alarm to IDS or IPS management software, such as the Cisco SDM

We define some of the commands which will be used while configuring the Router for IPS

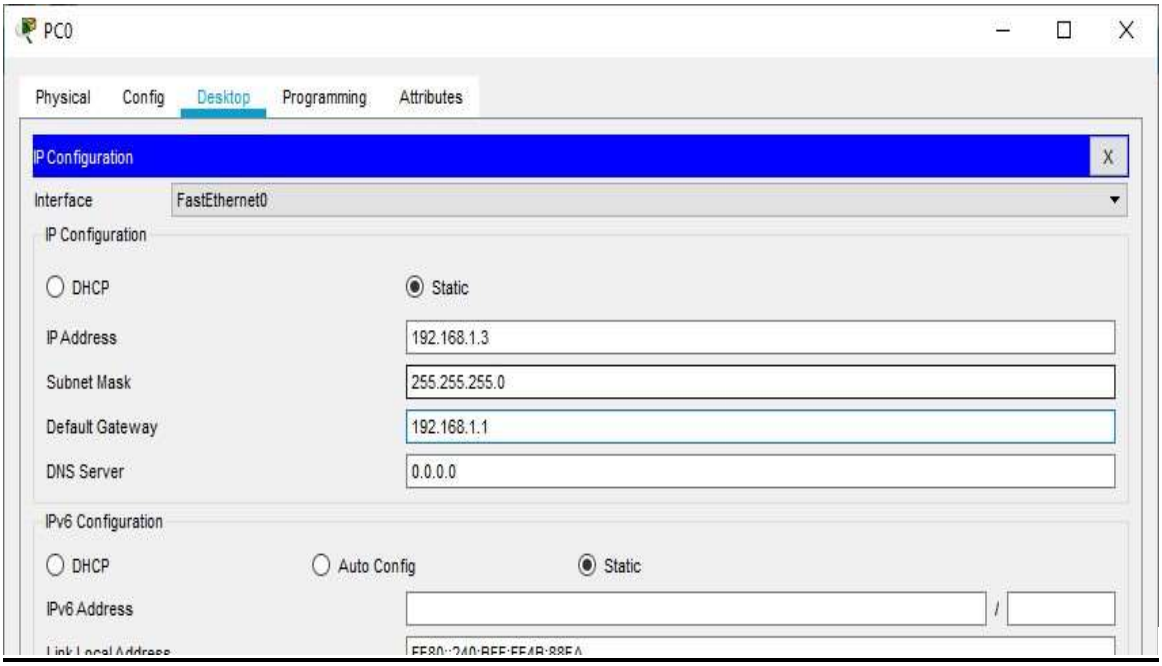
Commands	Function	Example
ip ips signaturecategory	Enters IPS category configuration mode.	Router(config)# ip ips signature-category
category	Specifies that all categories (and all signatures) are retired in the following step and enters IPS category action configuration mode Specifies the basic category (and a set of signatures) that are to be "unretired" in the following step.	Router(config-ips-category)# category all **Example:** Router(config-ips-category)# category ios_ips basic
retired {**true** | **false**}	Specifies that the device should retire all categories (and all signatures). **true** -- Retires all signatures within a given category. **false** --"Unretires" all signatures within a given category.	Router(config-ips-category-action)# retired true
mkdir flash:/ips5	Create a directory for which Cisco IOS IPS saves signature information.	**Example:** Device# mkdir flash:/ips5
ip ips name *ipsname*		**Example:** Device(config)# ip ips name myips

ip ips *ipsname* {**in** | **out**}	Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines.	**Example:** Device(config-if)# ip ips MYIPS in

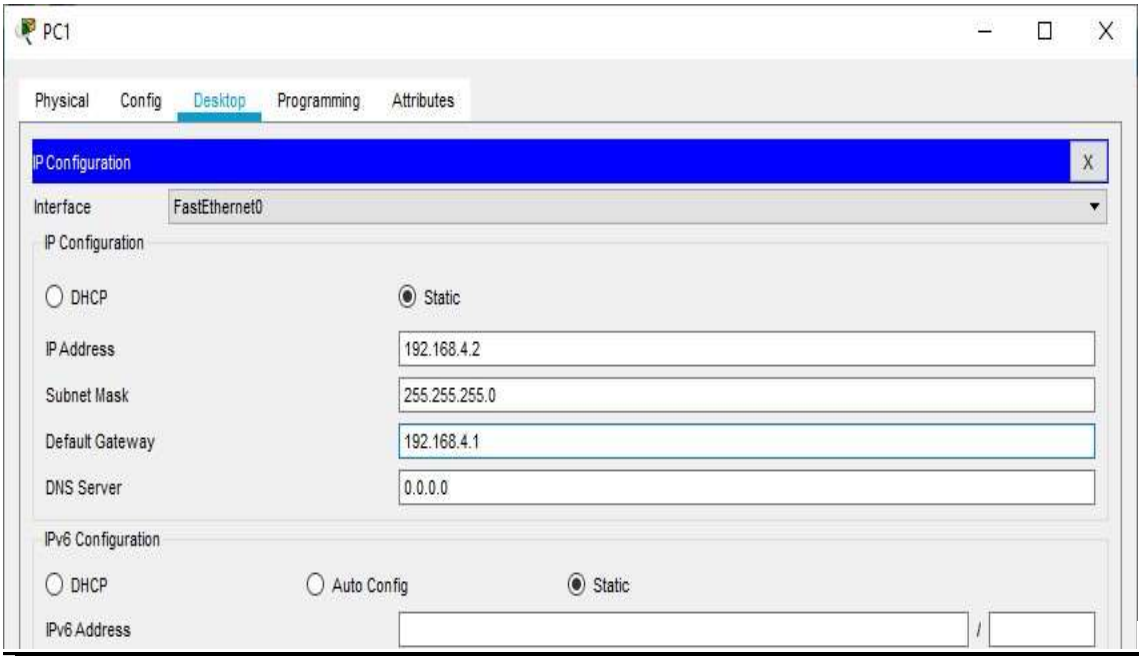
We us the following topology for the present case



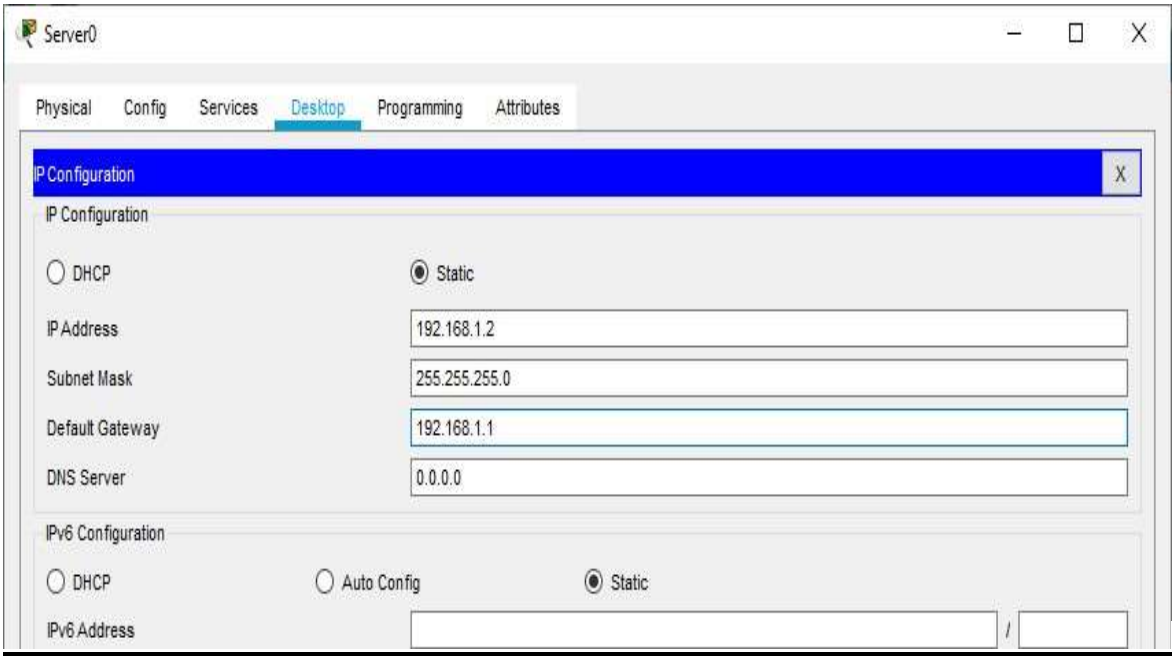
Configuring PC0



Configuring PC1

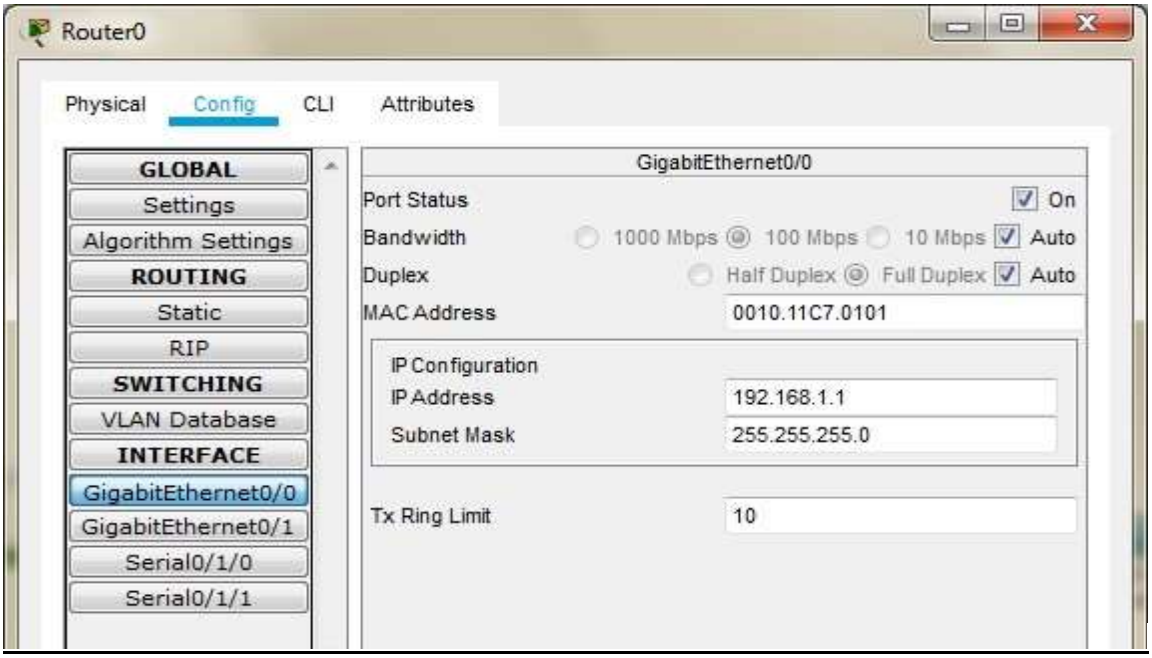


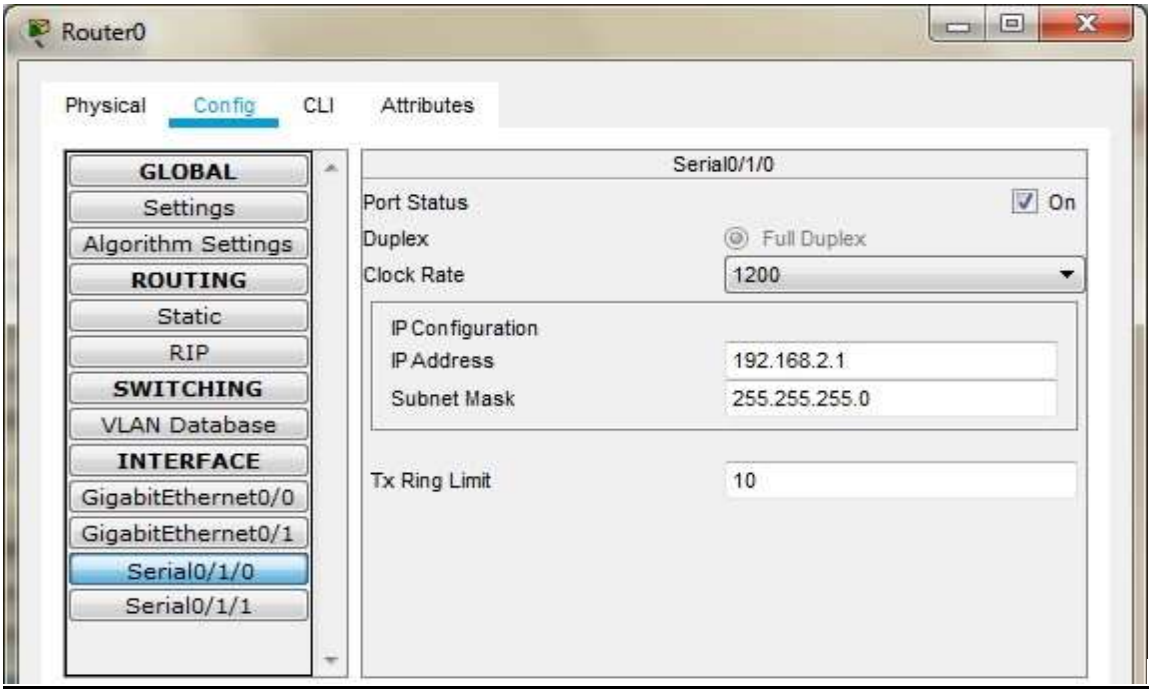
Configuring SERVER0



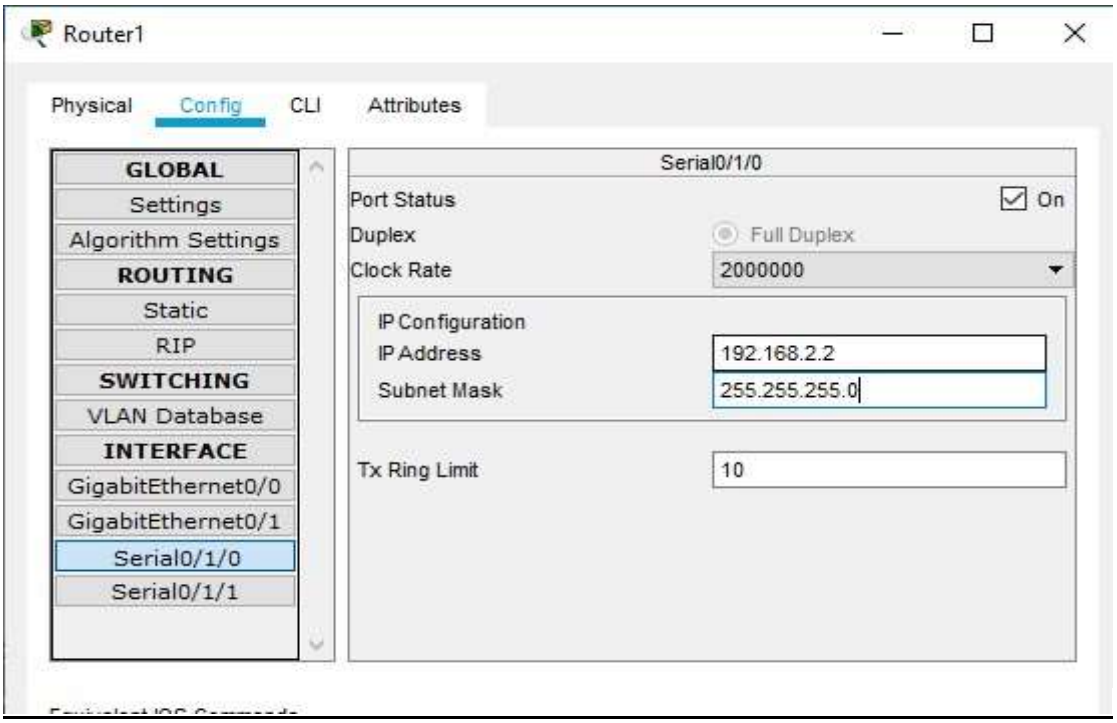
Serial Interface must be added in each Router before configuring it

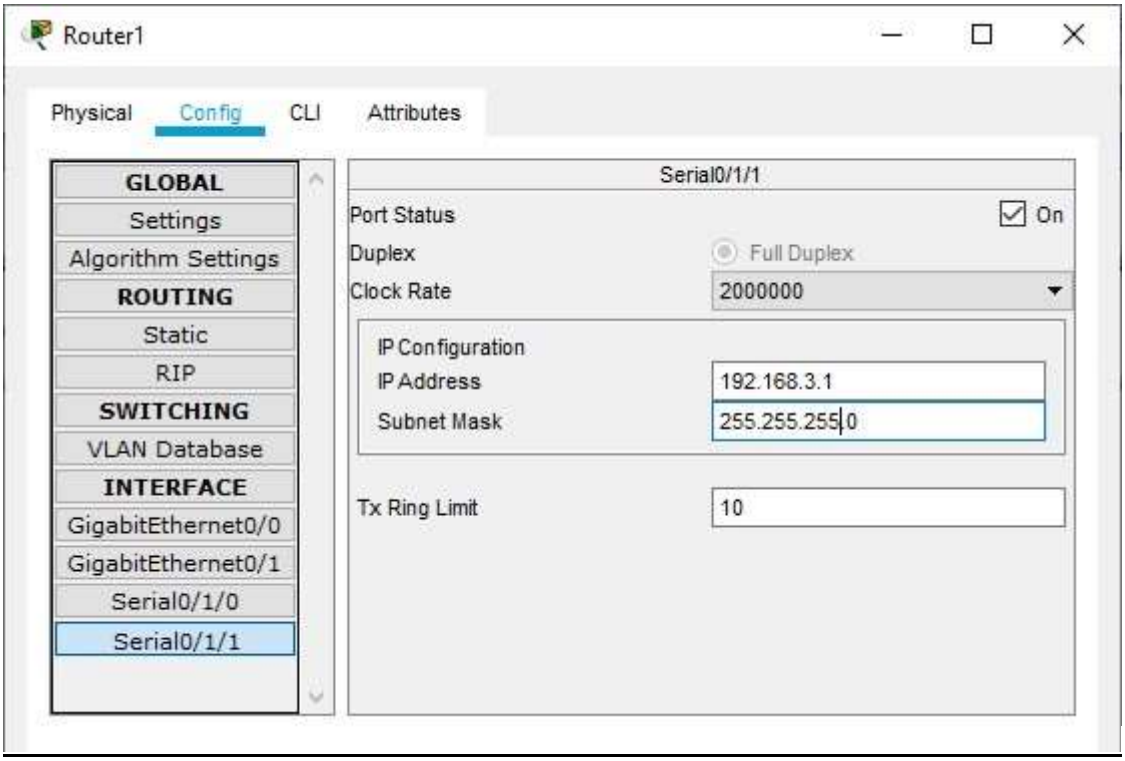
Configuring Router0



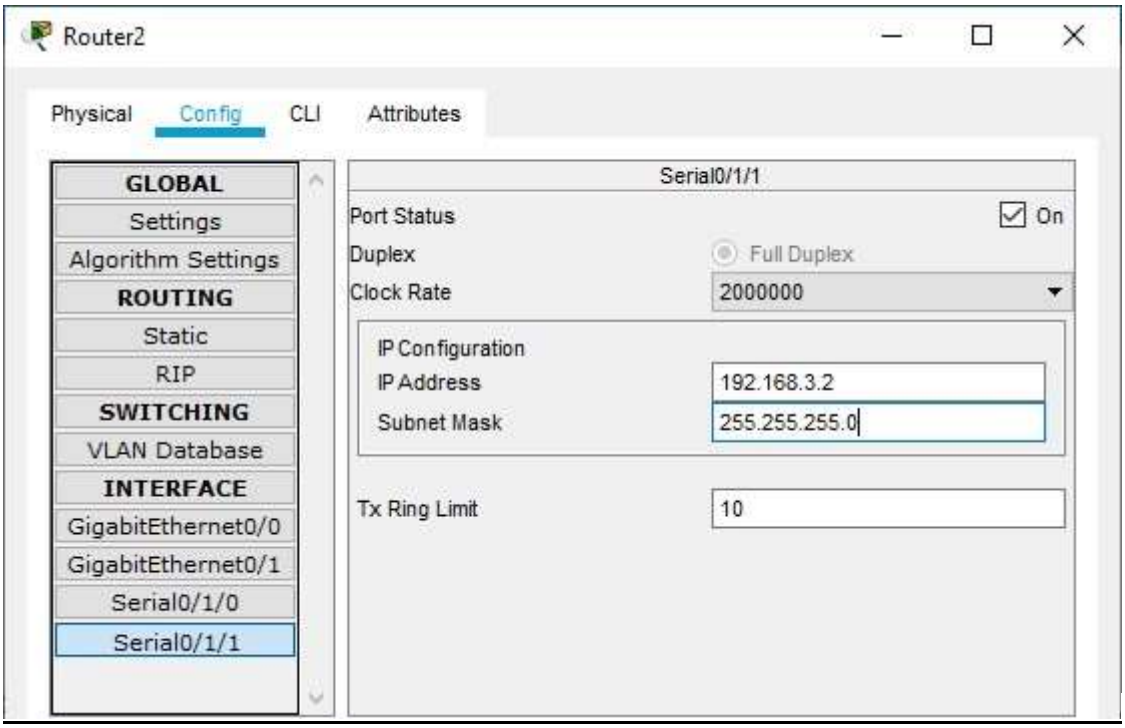


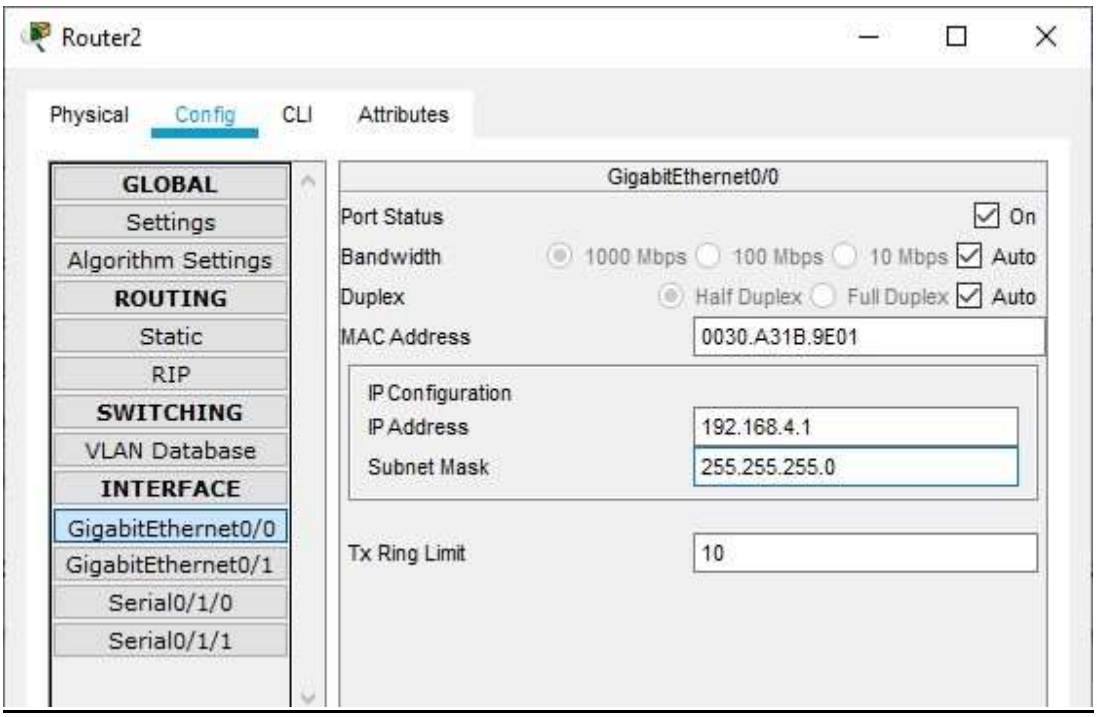
Configuring Router1





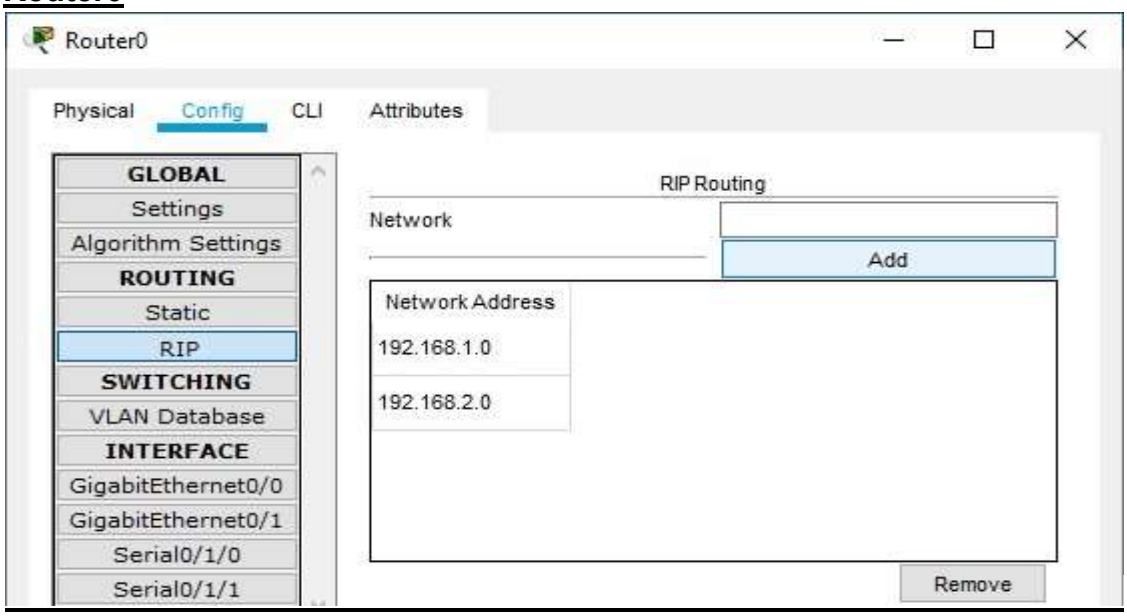
Configuring Router2


Router2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/1
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IP Address
192.168.3.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10

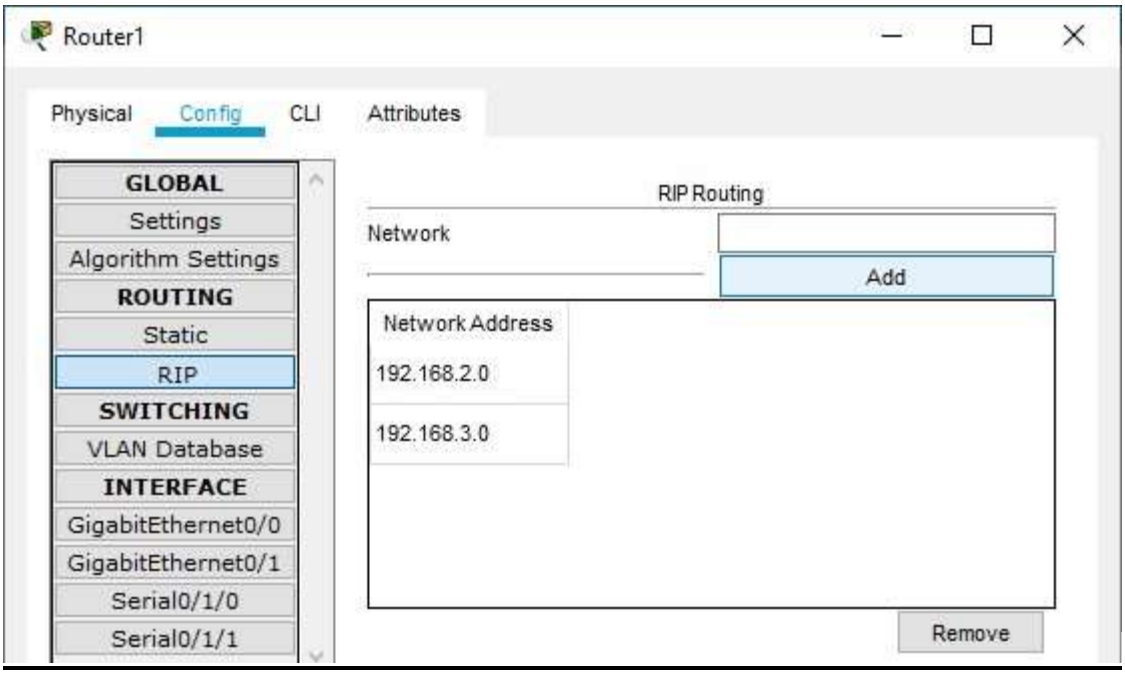

Router2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
GigabitEthernet0/0
Port Status
On
Bandwidth
1000 Mbps
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0030.A31B.9E01
IP Configuration
IP Address
192.168.4.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10

We need to set the Routing table in all the Routers so that each node could send and receive packets from others (RIP is set in all the Routers as follows)

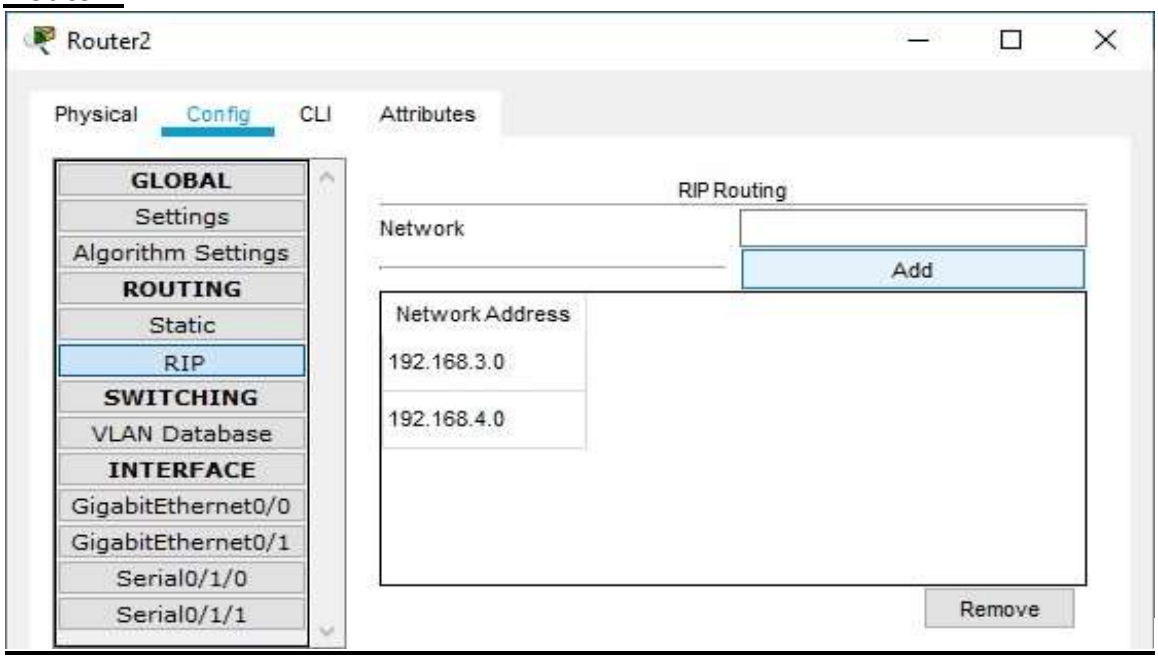
Router0



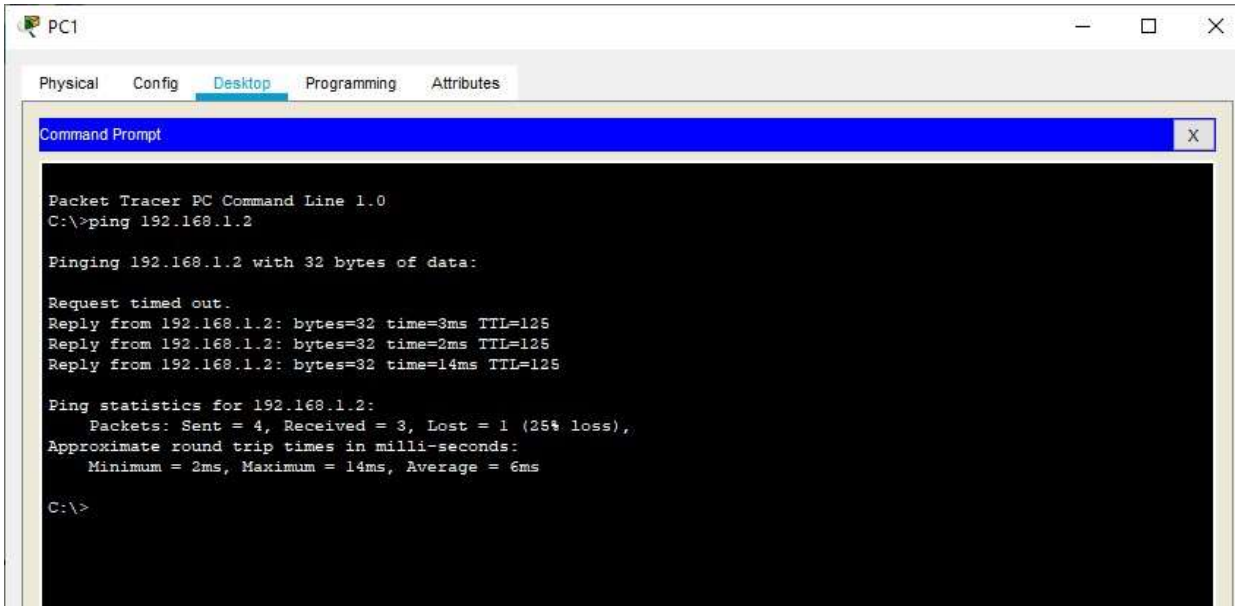
Router1



Router2



Now we can check the connectivity by sending ping commands from any node to any other node

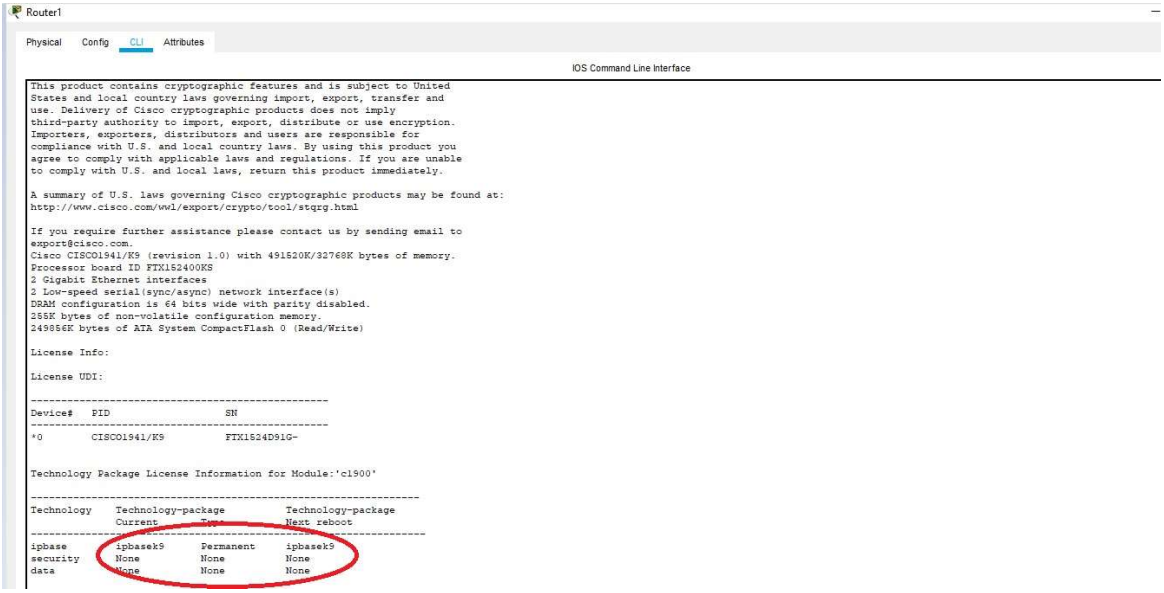


So we conclude that the connectivity has been established

PART1: Enable the IOS IPS (on Router1) Type the following command in the CLI mode of Router1

Router#show version

We will get a message informing whether the security Package is enabled or not

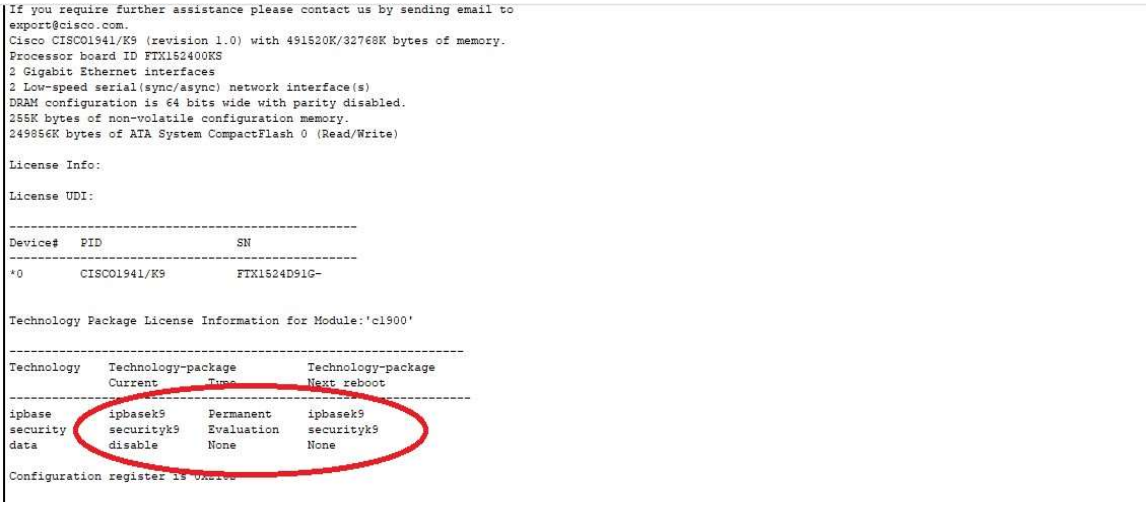


As seen above the security package is not enabled, to enable the security feature, type the following command in Router1

Router(config)#license boot module c1900 technology-package securityk9
Router(config)#exit
Router#
Router#reload

Router>enable
Router#
Router#show version

We will get a message informing whether the security package is enabled or not



As seen above now the security package has been enabled Now type the following commands in the CLI mode of Router1

Router#
Router#
Router#clock set 11:47:56 MARCH 3 2020

Router#mkdir smile
Router#configure terminal
Router(config)#ip ips config location flash:smile
Router(config)#ip ips name iosips
Router(config)#ip ips notify log
Router(config)#ip ips signature-category
Router(config-ips-category)#category all
Router(config-ips-category-action)#retired true
Router(config-ips-category-action)#exit

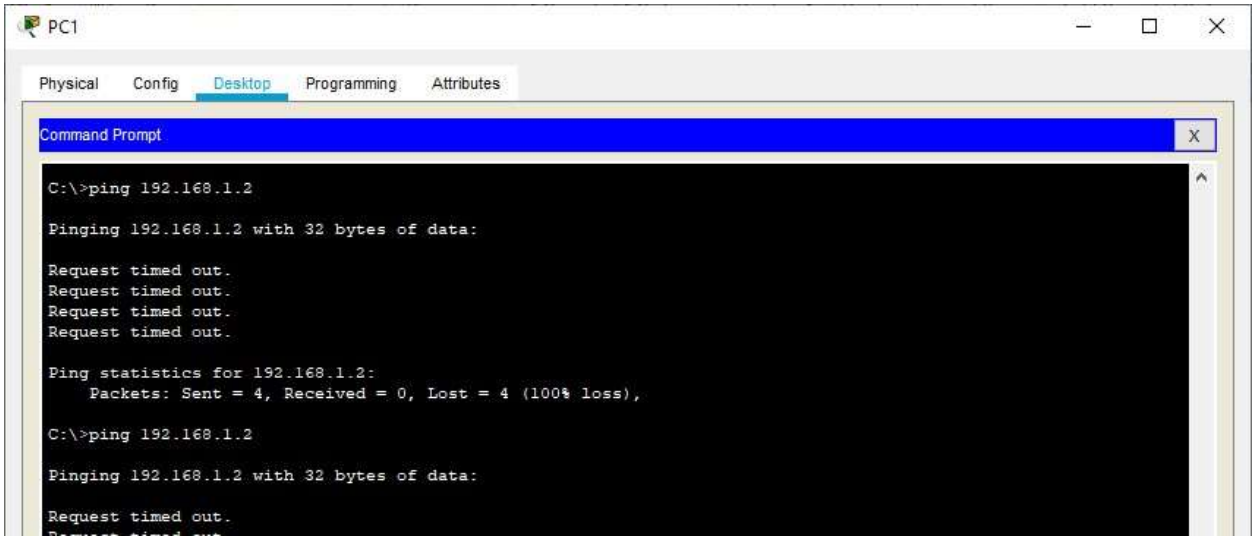
Router(config-ips-category)#category ios_ips basic
Router(config-ips-category-action)#retired false
Router(config-ips-category-action)#exit
Router(config-ips-category)#exit

Router(config)#interface Serial0/1/0
Router(config-if)#ip ips iosips out
Router(config-if)# Router(config)#

Part 2: Modify the Signature Type the following commands in the CLI mode of Router1

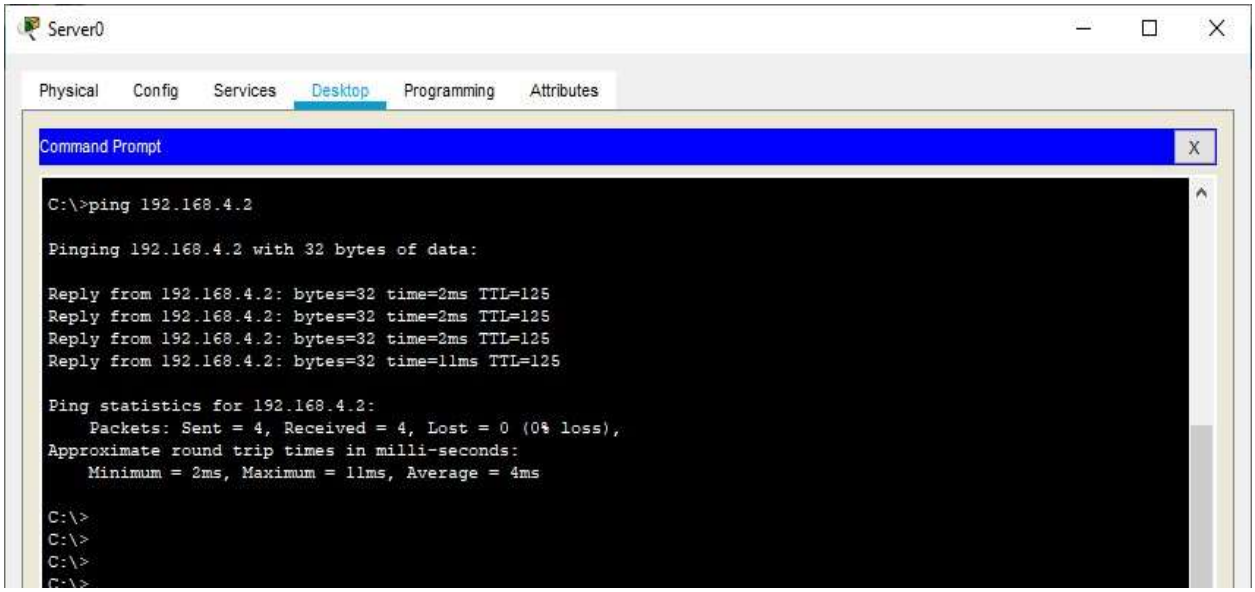
```
Router(config)#
Router(config)#ip ips signature-definition
Router(config-sigdef)#signature 2004 0
Router(config-sigdef-sig)#status
Router(config-sigdef-sig-status)#retired false
Router(config-sigdef-sig-status)#enabled true
Router(config-sigdef-sig-status)#exit
Router(config-sigdef-sig)#engine
Router(config-sigdef-sig-engine)#event-action produce-alert
Router(config-sigdef-sig-engine)#event-action deny-packet-inline
Router(config-sigdef-sig-engine)#exit
Router(config-sigdef-sig)#exit
Router(config-sigdef)#exit
Router(config)#
```

Now we need to verify the above IPS configuration, we do it first by pinging PC1 to SERVER and then from SERVER to PC1 PC1 to SERVER

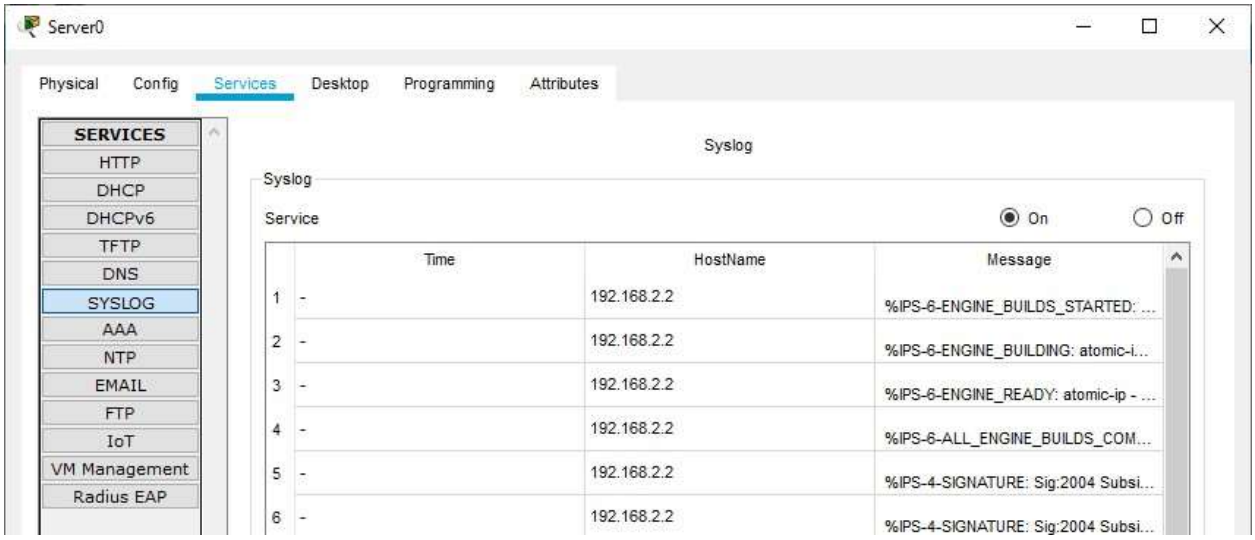


The ping FAILS

SERVER to PC1



Also we can observe the Syslog service in the SERVER to check the log activities



Hence we set the IPS and also verified it on Router1