PRACTICAL NO 2: Configure ACLs

The Cisco Access Control List (ACL) are used for filtering traffic based on a given filtering criteria on a router or switch interface. Based on the conditions supplied by the ACL, a packet is allowed or blocked from further movement.

Cisco ACLs are available for several types of routed protocols including IP, IPX, AppleTalk, XNS, DECnet, and others. However, we will be discussing ACLs pertaining to TCP/IP protocol only.

ACLs for TCP/IP traffic filtering are primarily divided into two types:

- Standard Access Lists, and
- Extended Access Lists
-

Standard Access Control Lists:

Standard IP ACLs range from 1 to 99. A Standard Access List allows you to permit or deny traffic FROM specific IP addresses. The destination of the packet and the ports involved can be anything.

This is the command syntax format of a standard ACL.

access-list *access-list-number* {permit|deny}
{host|source source-wildcard|any}

Standard ACL example:

access-list 10 permit 192.168.2.0 0.0.0.255

This list allows traffic from all addresses in the range 192.168.2.0 to 192.168.2.255

Note that when configuring access lists on a router, you must identify each access list uniquely by assigning either a name or a number to the protocol's access list.

There is an implicit deny added to every access list. If you entered the command:

show access-list 10

The output looks like:

access-list 10 permit 192.168.2.0 0.0.0.255 access-list 10 deny any

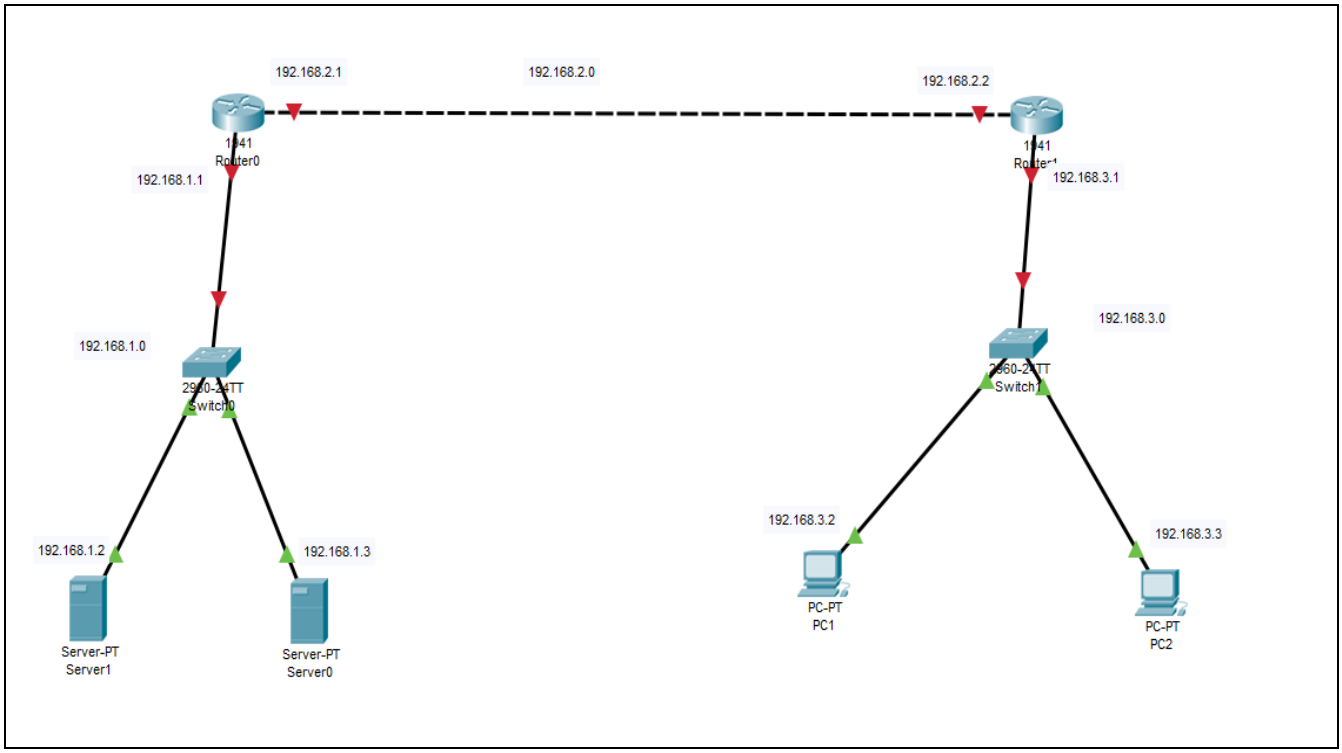
Extended Access Control Lists:

Extended IP ACLs allow you to permit or deny traffic from specific IP addresses to a specific destination IP address and port. It also allows you to have granular control by specifying controls for different types of protocols such as ICMP, TCP, UDP, etc within the ACL statements. Extended IP ACLs range from 100 to 199. In Cisco IOS Software Release 12.0.1, extended ACLs began to use additional numbers (2000 to 2699).

The syntax for IP Extended ACL is given below:

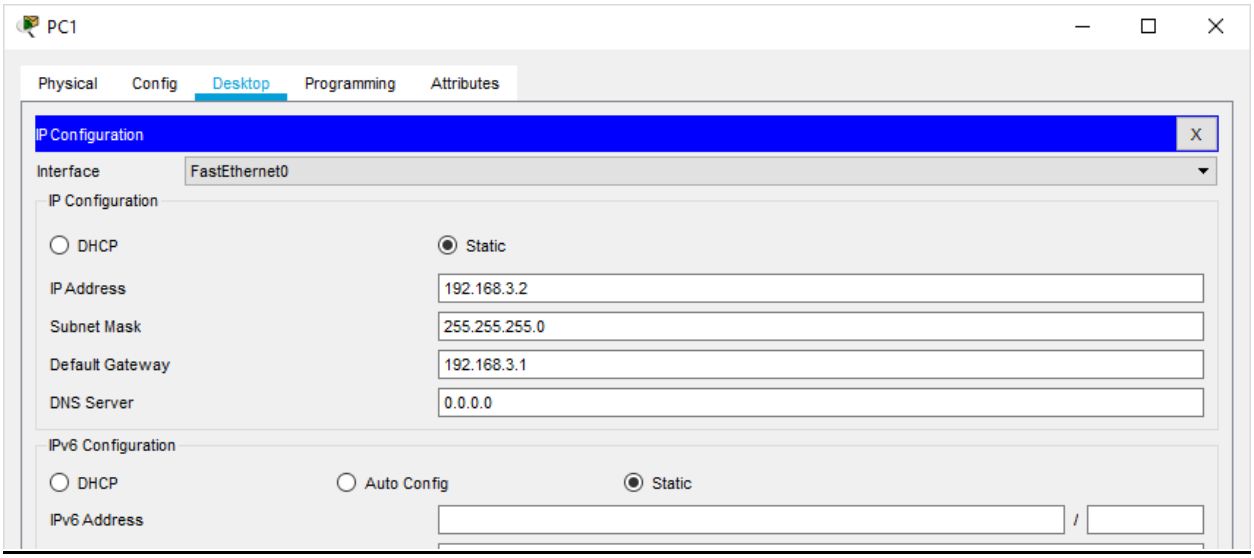
access-list access-list-number {deny | permit} protocol source source-wildcard *destination* destination-wildcard [precedence precedence]
Note that the above syntax is simplified, and given for general understanding only.
Extended ACL example:
access-list 110 - Applied to traffic leaving the office (outgoing)
access-list 110 permit tcp 92.128.2.0 0.0.0.255 any eq 80
ACL 110 permits traffic originating from any address on the 92.128.2.0 network. The 'any' statement means that the traffic is allowed to have any destination address with the limitation of going to port 80. The value of 0.0.0.0/255.255.255.255 can be specified as 'any'.
Applying an ACL to a router interface:
After the ACL is defined, it must be applied to the interface (inbound or outbound). The syntax for applying an ACL to a router interface is given below:
interface <interface>
ip access-group {number|name} {in|out}
An Access List may be specified by a name or a number. "in" applies the ACL to the inbound traffic, and "out" applies the ACL on the outbound traffic.
Example: To apply the standard ACL created in the previous example, use the following commands:
Rouer(config)#interface serial0
Rouer(config-if)#ip access-group 10 out

Consider the following topology

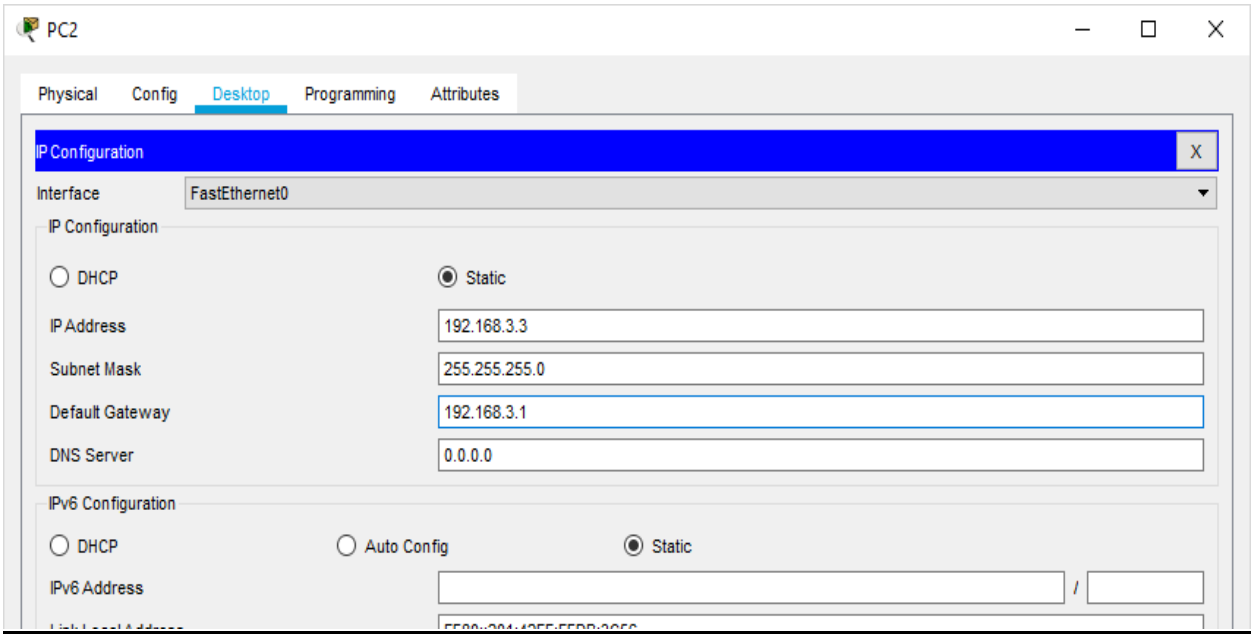


Part 1: Configure, Apply and Verify an Extended Numbered ACL

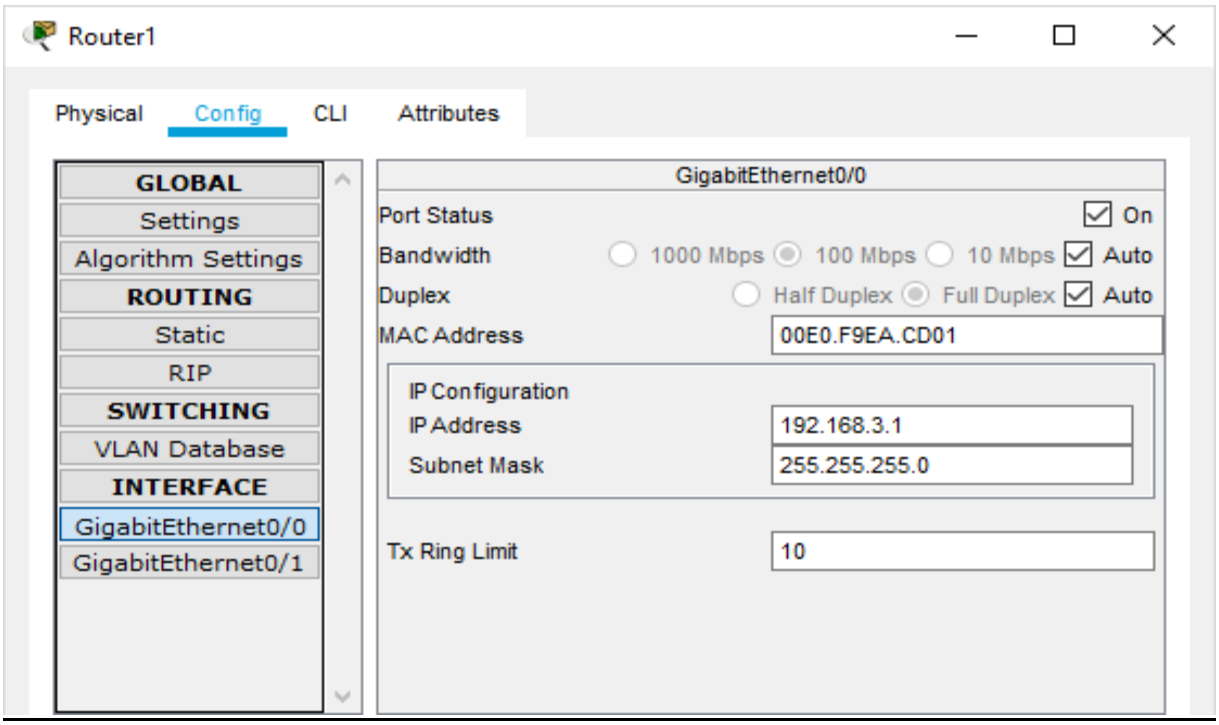
Configuring PC1

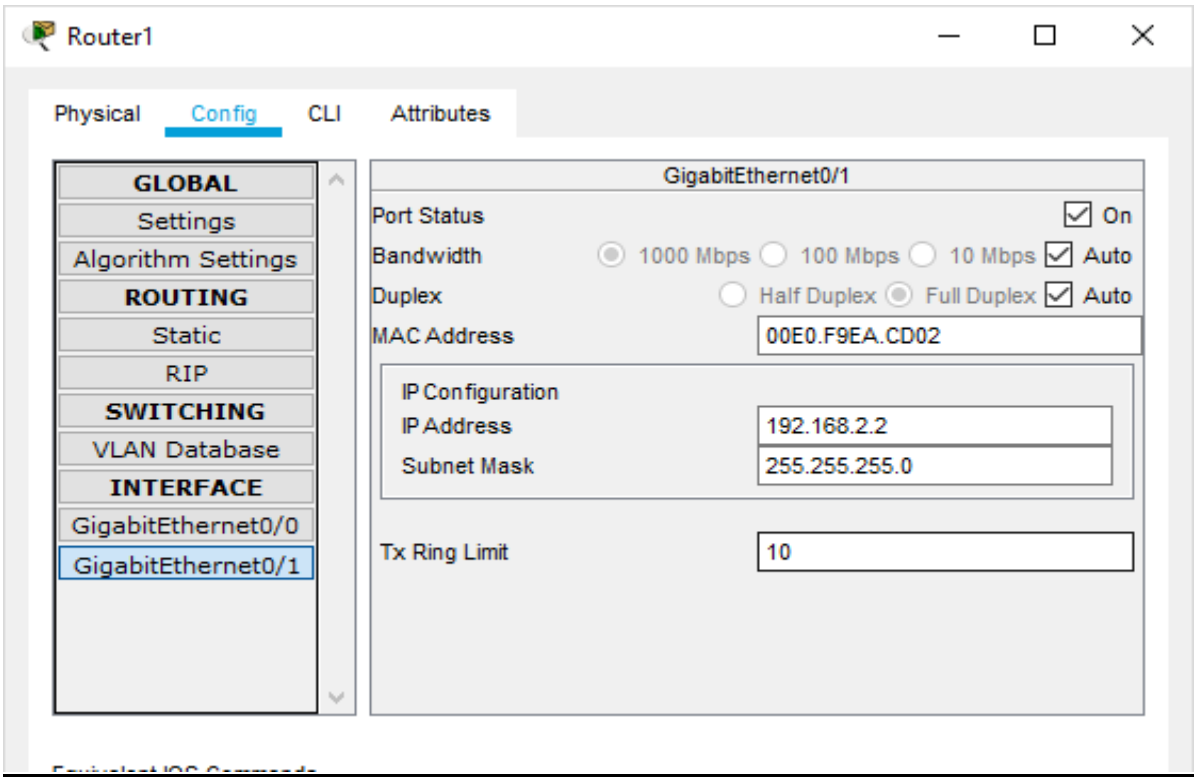


Configuring PC2

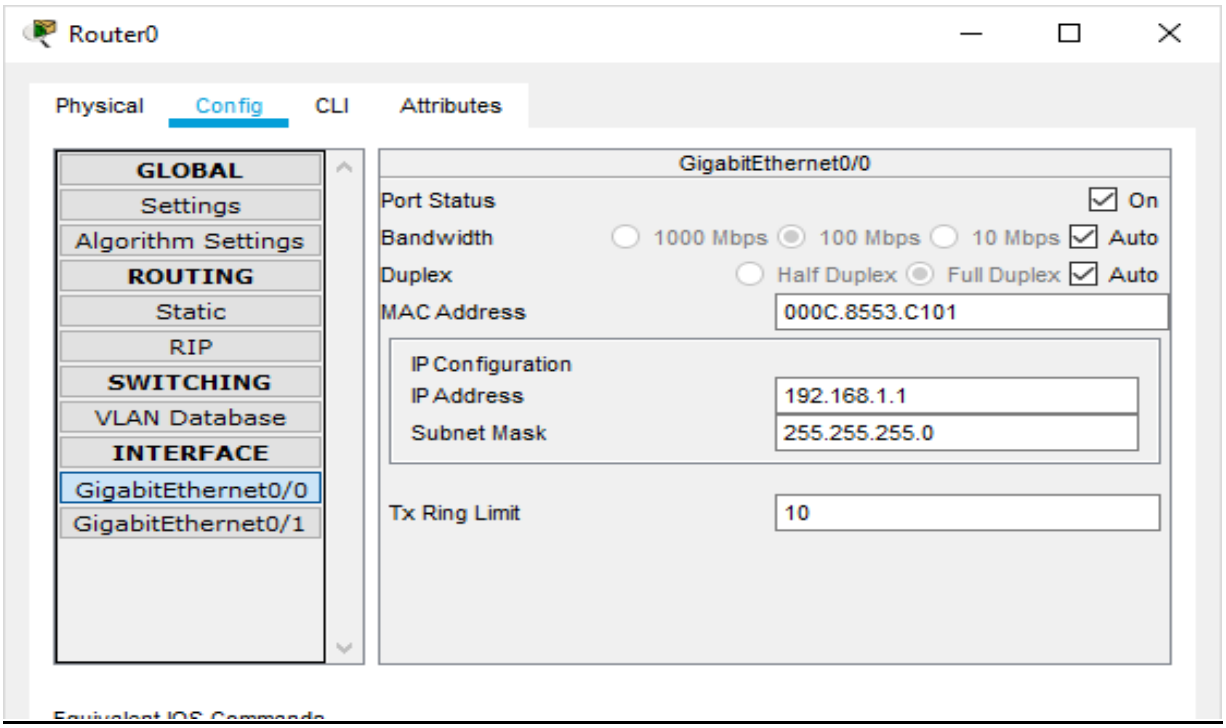


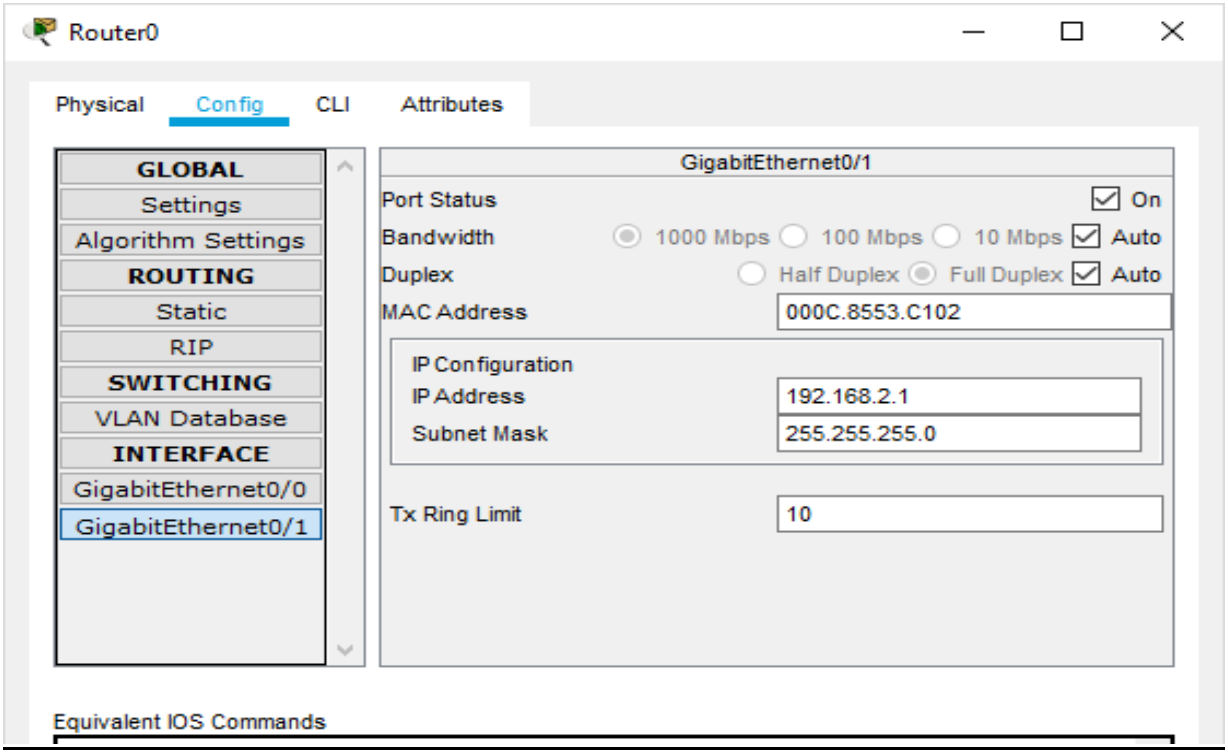
Configuring Router1



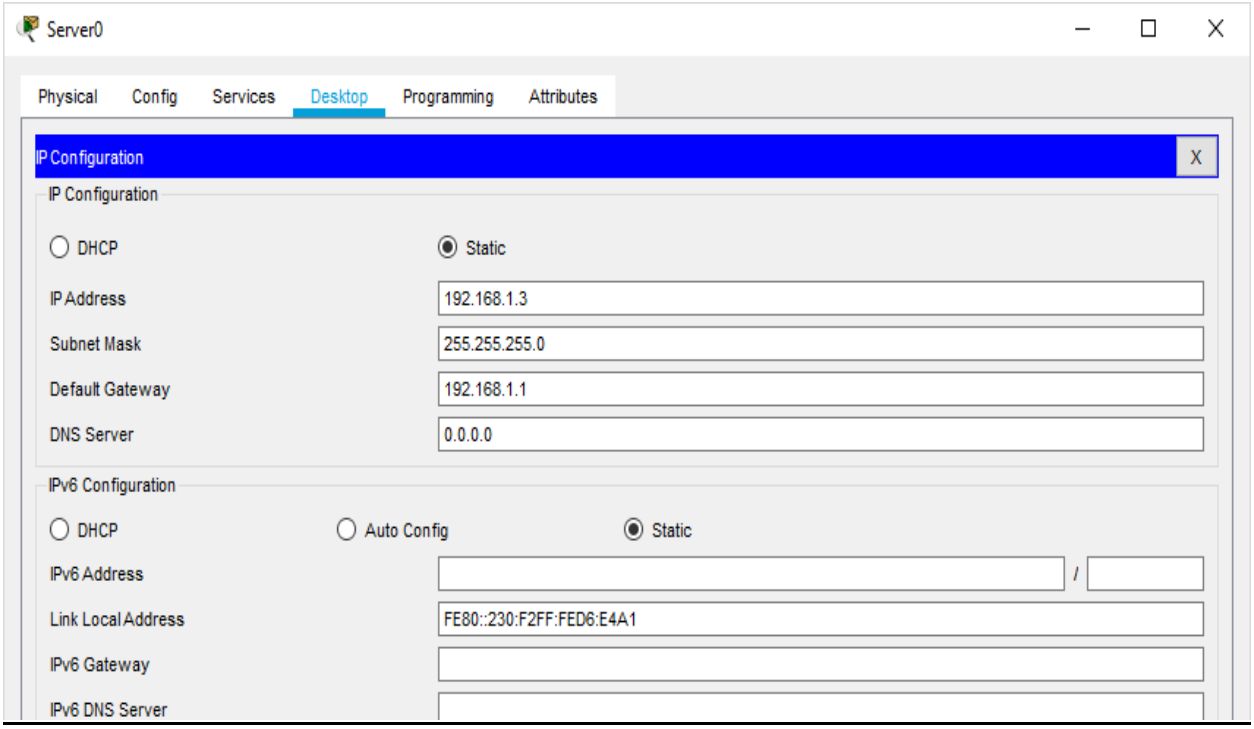


Configuring Router0

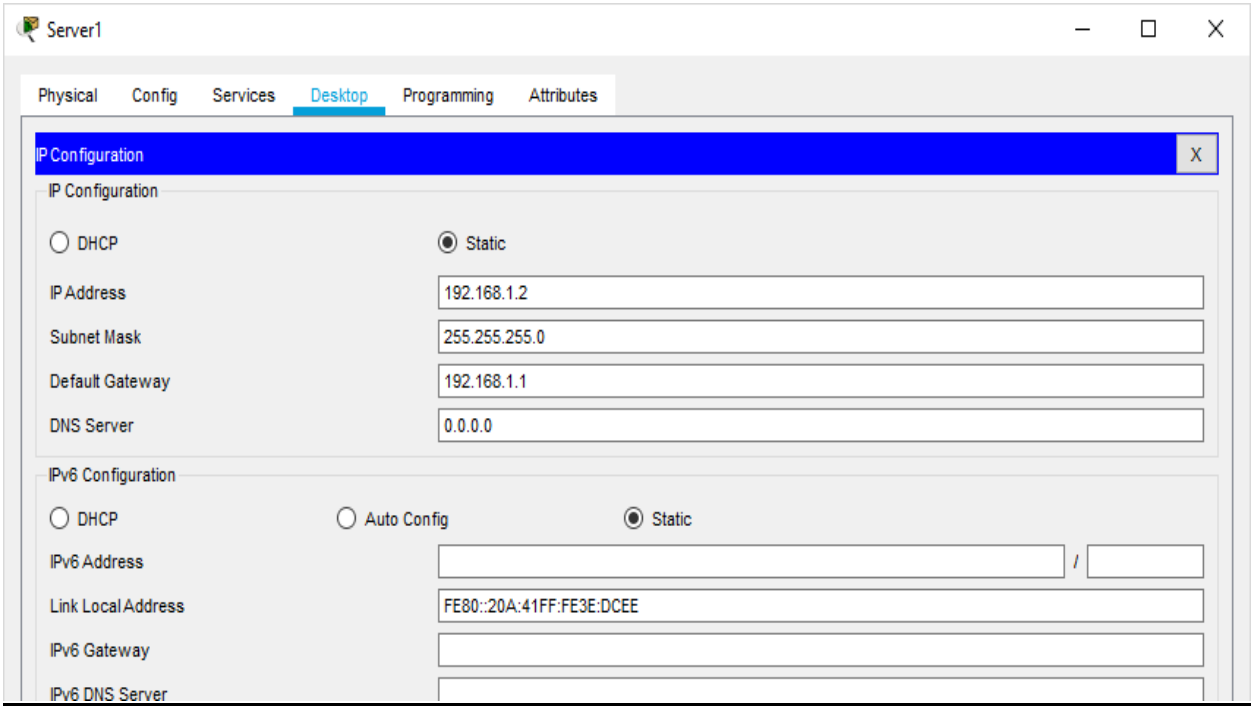




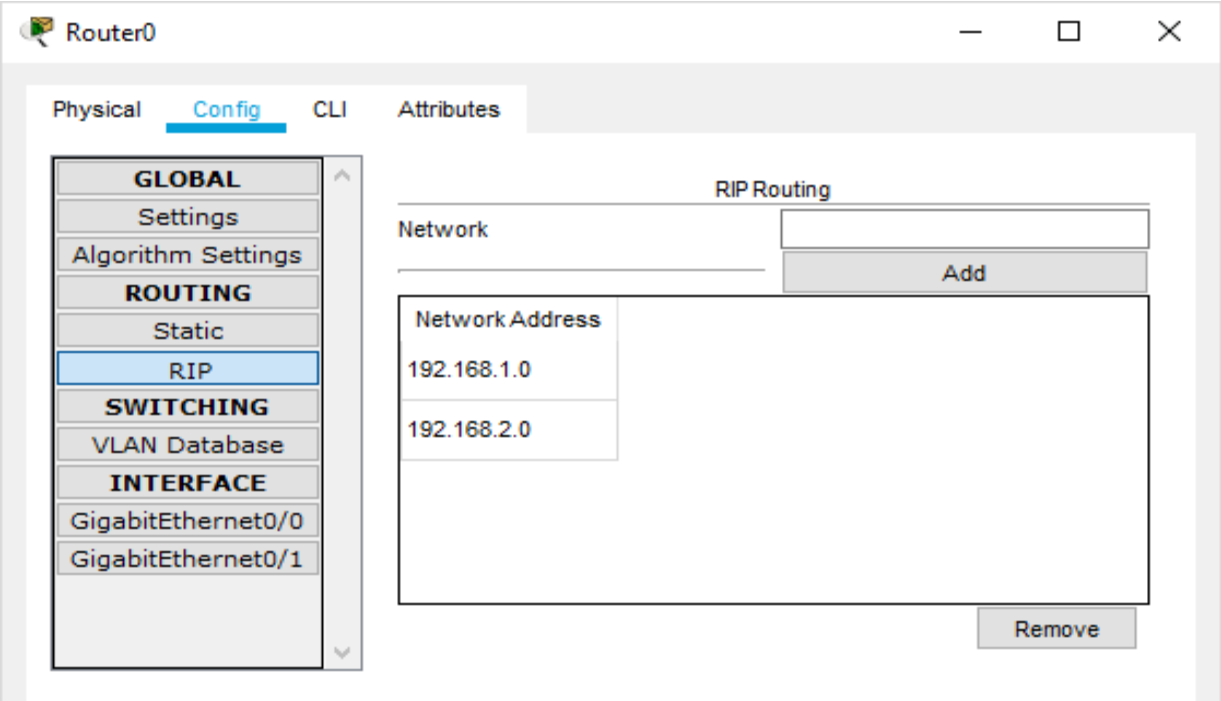
Configuring Server0

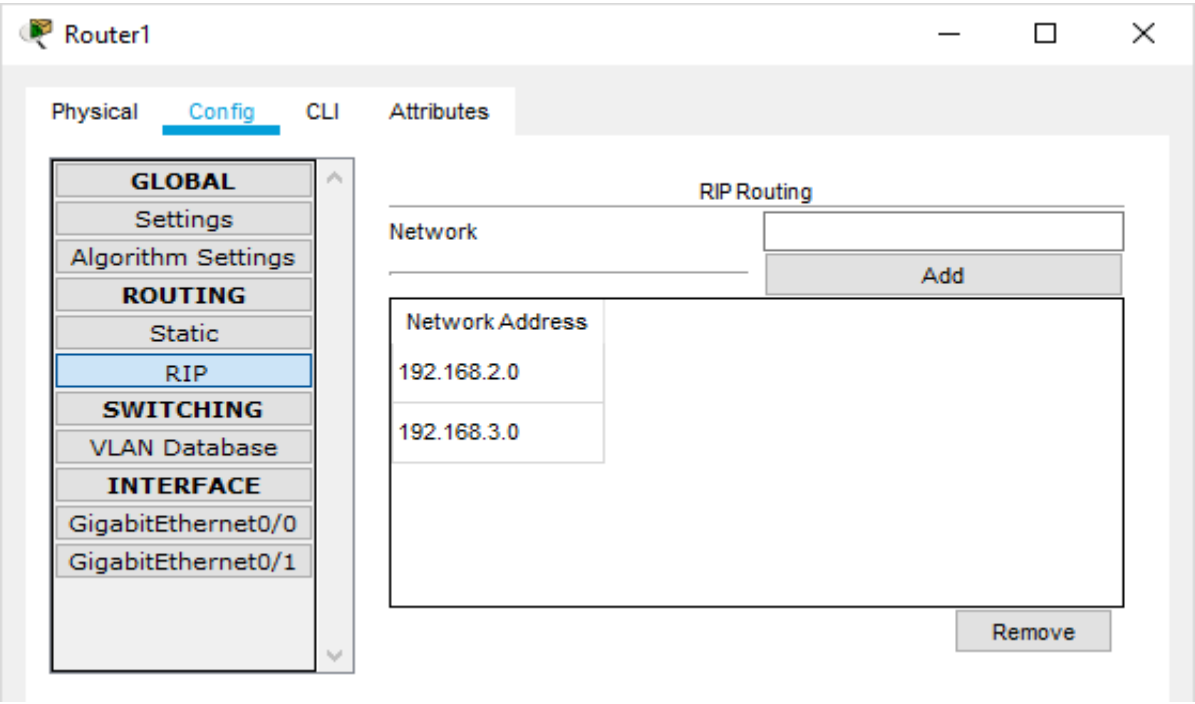


Configuring Server1



Set the RIP protocol on both the Routers as follows





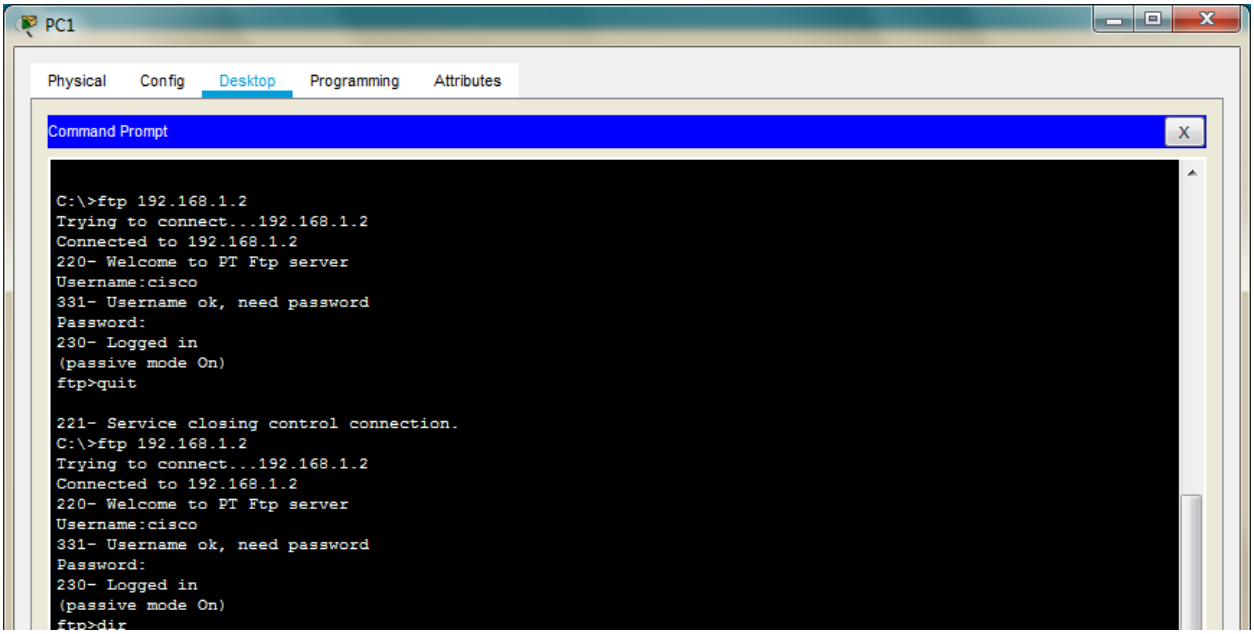
Check the connectivity by using the ping command

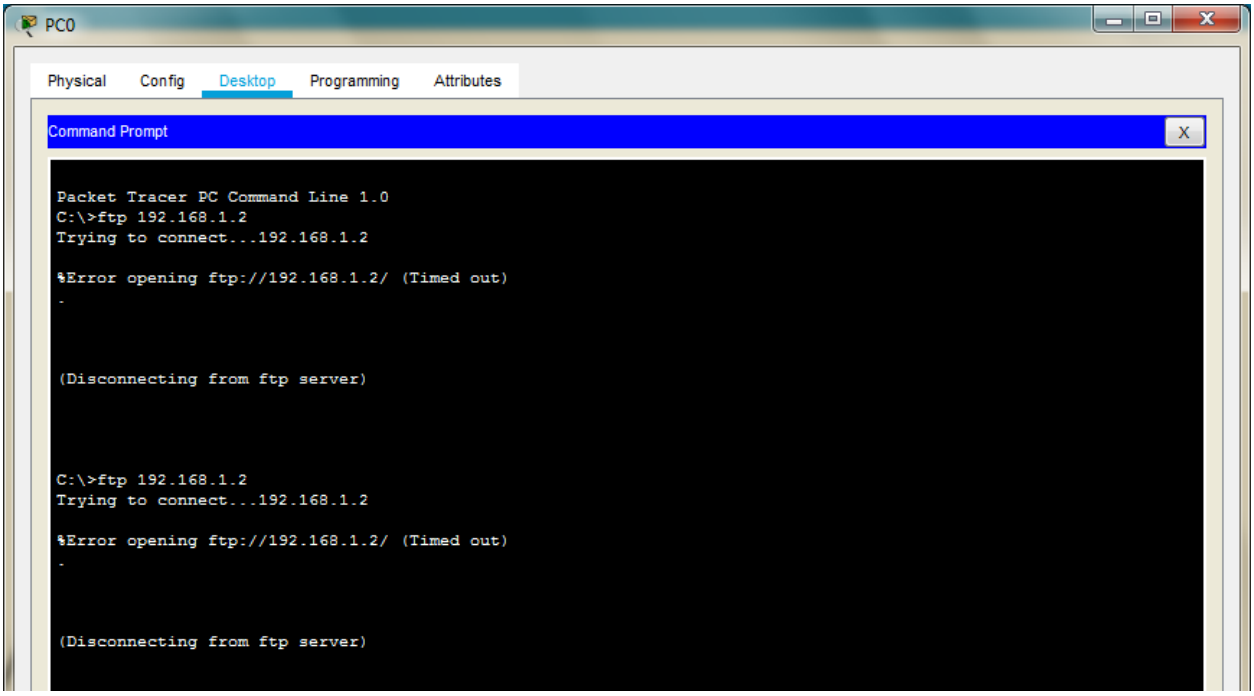
Part 1: Configure, Apply and Verify an Extended Numbered ACL

Type the following commands in Router1

```
Router#configure  terminal
Router(config)#
Router(config)#access-list 100 permit tcp host 192.168.3.2 host 192.168.1.2 eq ftp
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group 100 out
Router(config-if)#exit
Router(config)#
```

Now verify the ftp (ftp 192.168.1.2) command from both the PCs, one would be successful (PC1) and other (PC0) would fail





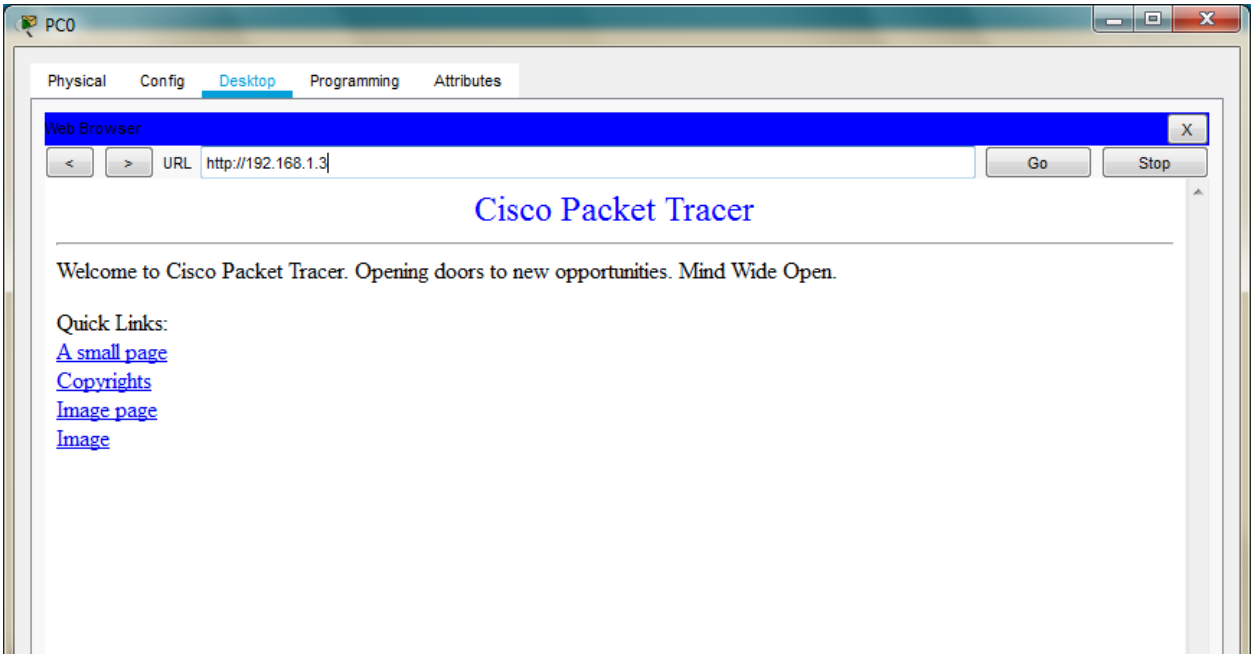
Part 2: Configure, Apply and Verify an Extended Named ACL

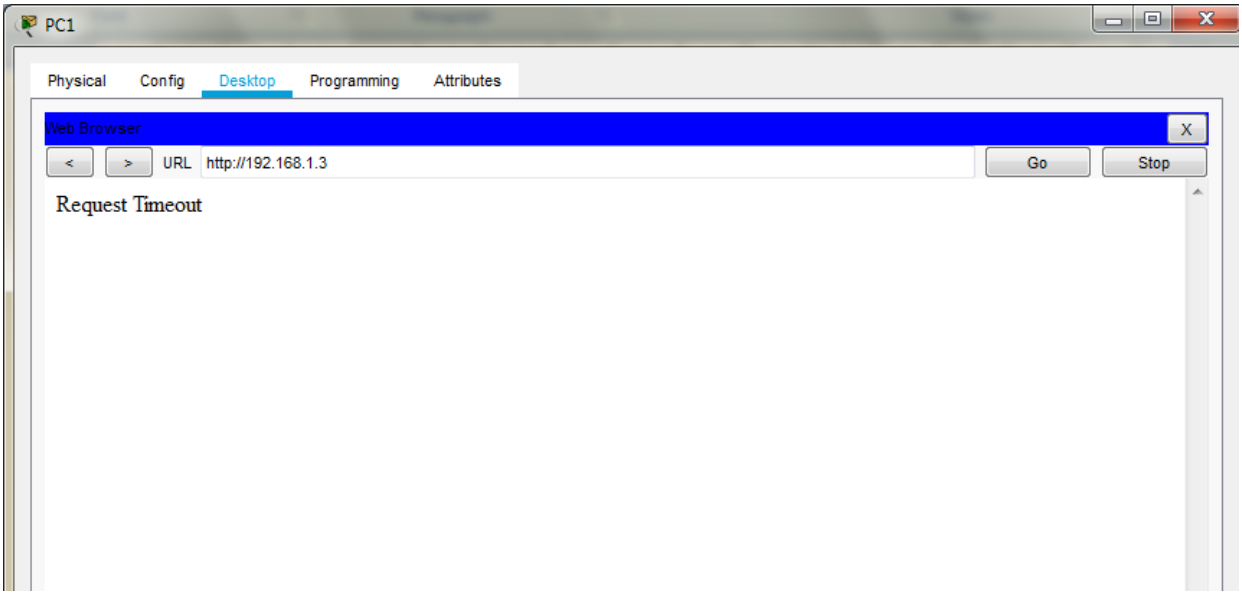
We use the same topology for this case

Type the following command in the CLI mode of Router1

```
Router>
Router>en
Router#configure terminal
Router(config)#ip access-list extended SMILE
Router(config-ext-nacl)#permit tcp host 192.168.3.3 host 192.168.1.3 eq www
Router(config-ext-nacl)#exit
Router(config)#
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip access-group SMILE out
Router(config-if)#exit
Router(config)#
```

Now verify the www (192.168.1.3) command from both the PCs browser, one would be successful (PC0) and other (PC1) would fail





Hence Extended Numbered ACLs as well as Extended Named ACLs have been verified