Practical 5

Configure IPv6 ACLs

Access Control Lists (ACLs) 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).

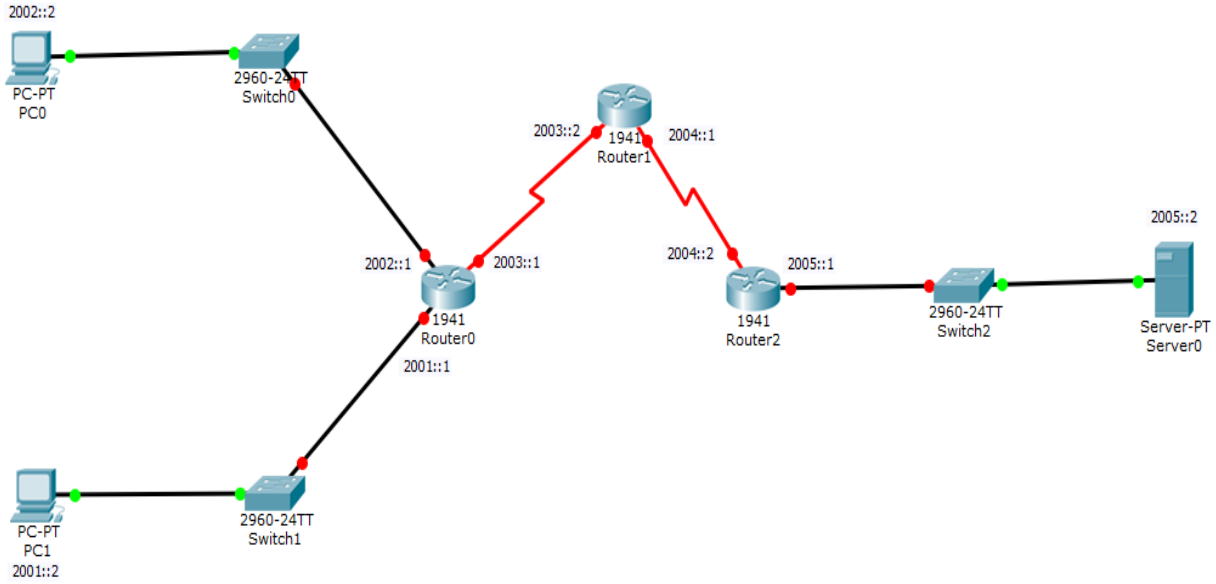
IPv6 Packet Inspection

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

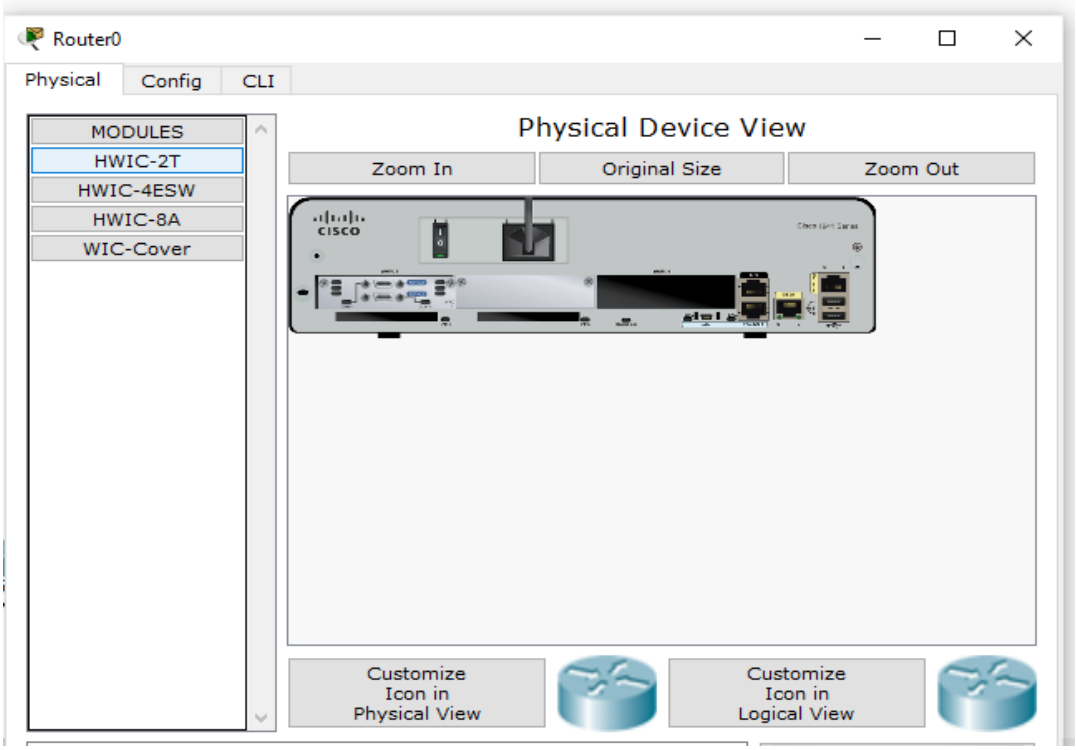
Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 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.

Consider the following topology

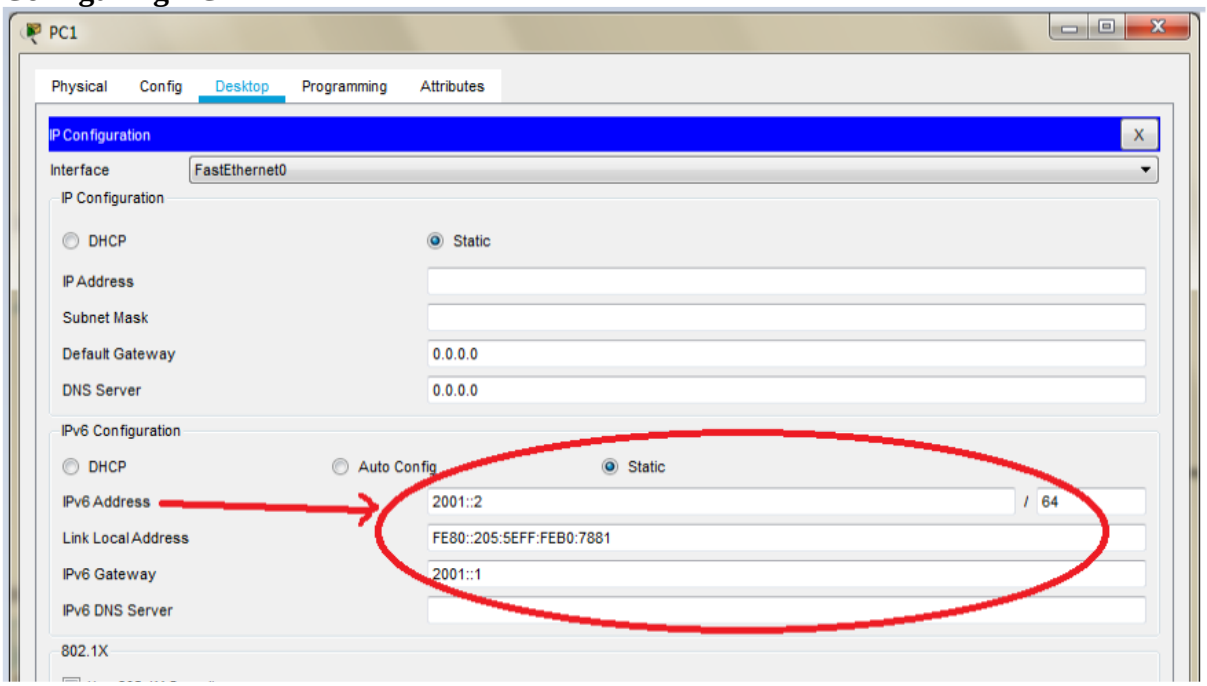


Topology Configuration

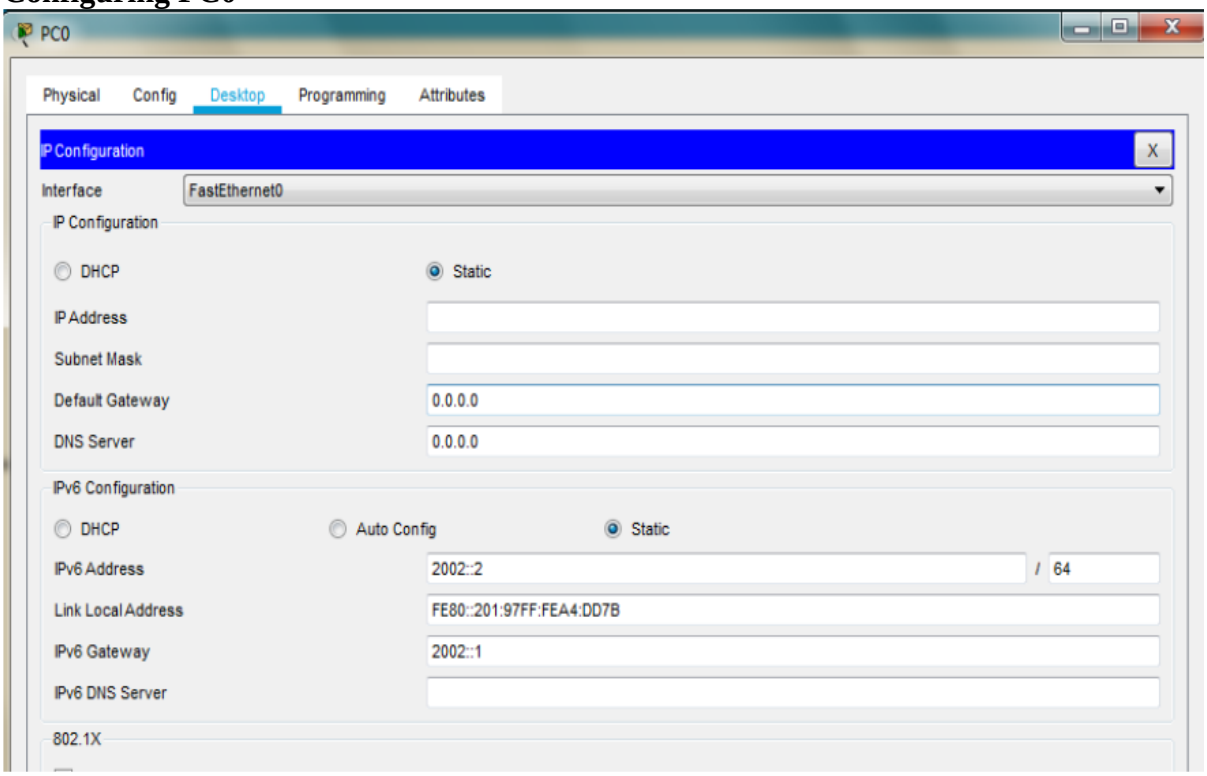
The serial interface in each Router is added as follows



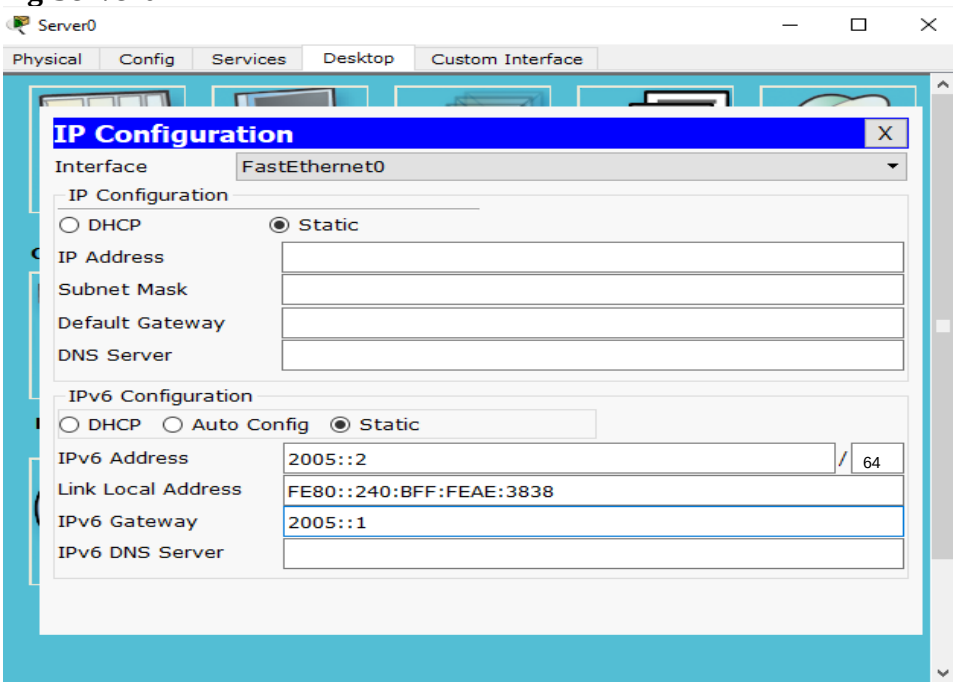
Configuring PC1



Configuring PC0



Configuring Server0



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

Configuring Router0

```
Router>
Router>en
Router#
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#int G0/0
Router(config-if)#ipv6 address 2002::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
Router(config)#int G0/1
Router(config-if)#ipv6 address 2001::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
Router(config)#int Se0/1/0
Router(config-if)#ipv6 address 2003::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```

Configuring Router1

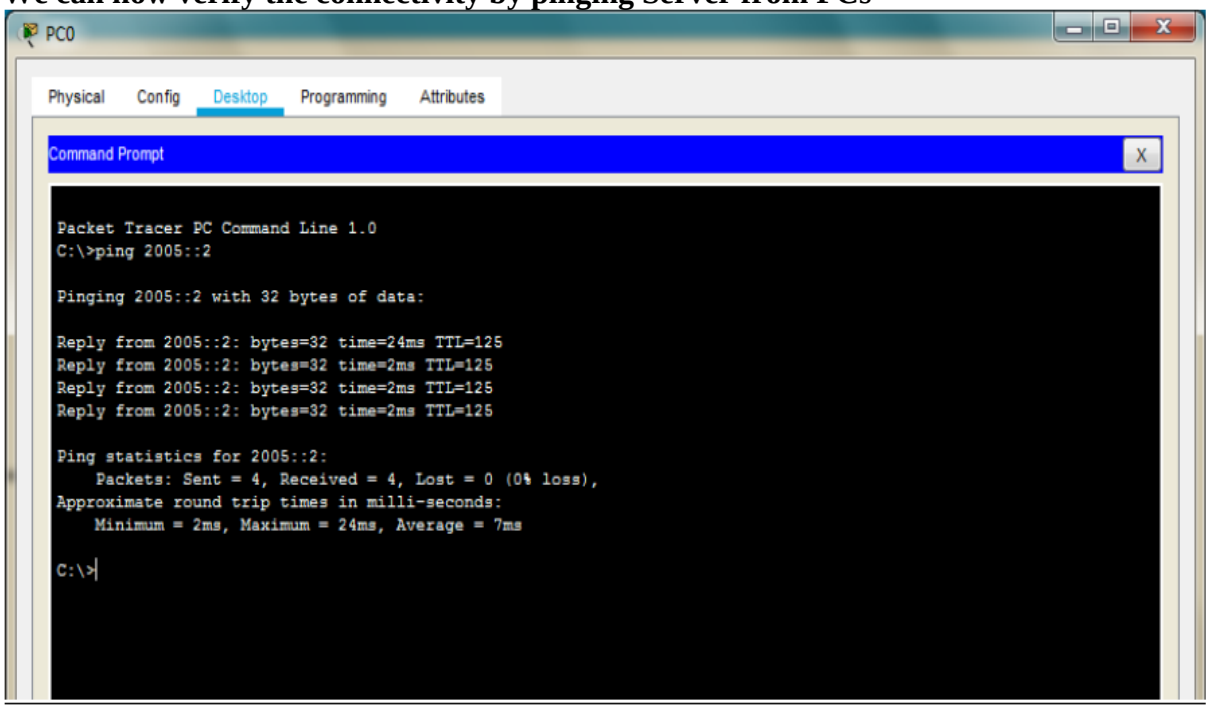
```
Router>en
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#
Router(config)#int Se0/1/0
Router(config-if)#ipv6 address 2003::2/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#
Router(config-if)#exit
Router(config)#
Router(config)#int Se0/1/1
Router(config-if)#ipv6 address 2004::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```

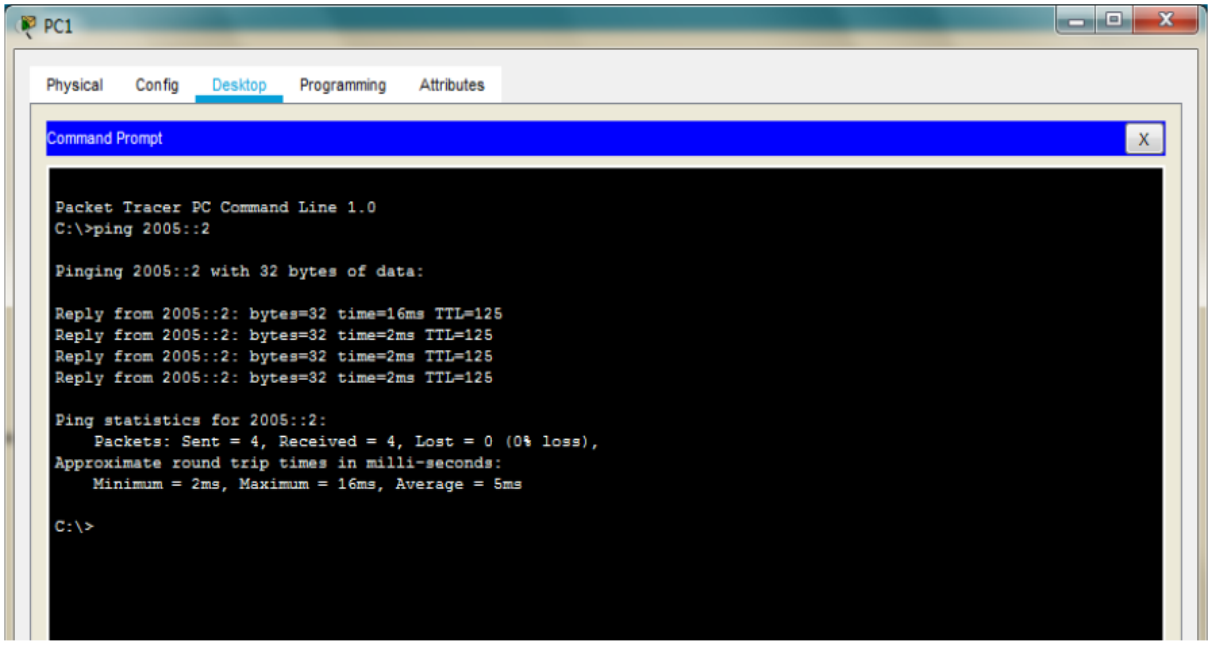
Configuring Router2

```
Router>en
Router#conf t
Router(config)#ipv6 unicast-routing
Router(config)#
Router(config)#int Se0/1/1
Router(config-if)#ipv6 address 2004::2/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#int G0/0
Router(config-if)#ipv6 address 2005::1/64
Router(config-if)#ipv6 rip a enable
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
```

Verify Connectivity

We can now verify the connectivity by pinging Server from PCs





And we see that the connectivity is established

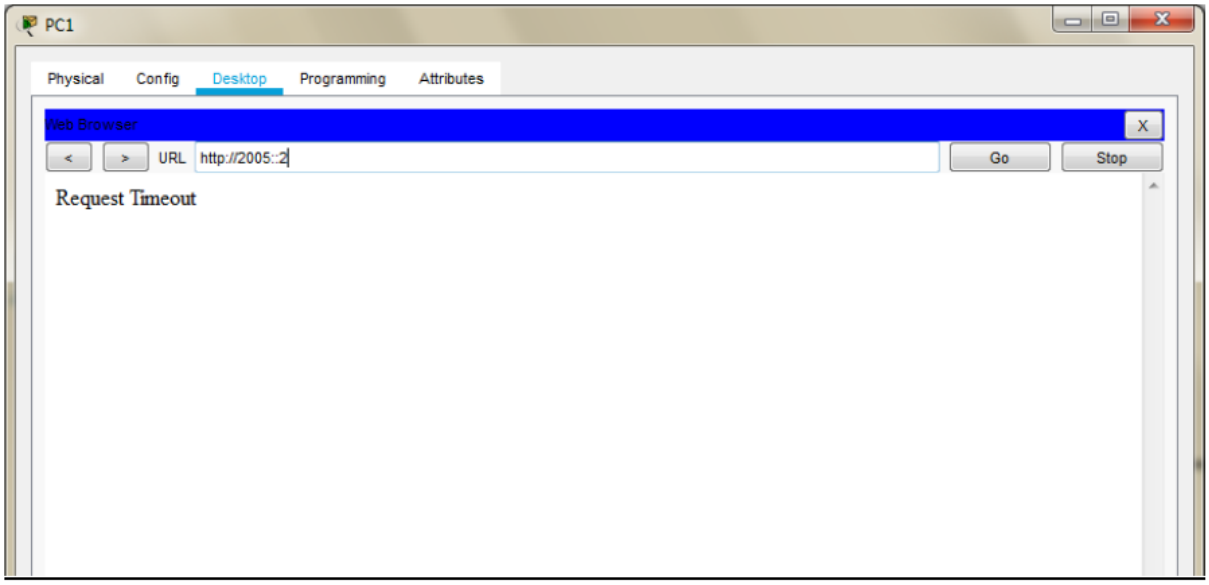
We configure the ACL and apply it to the Router1 with the following conditions

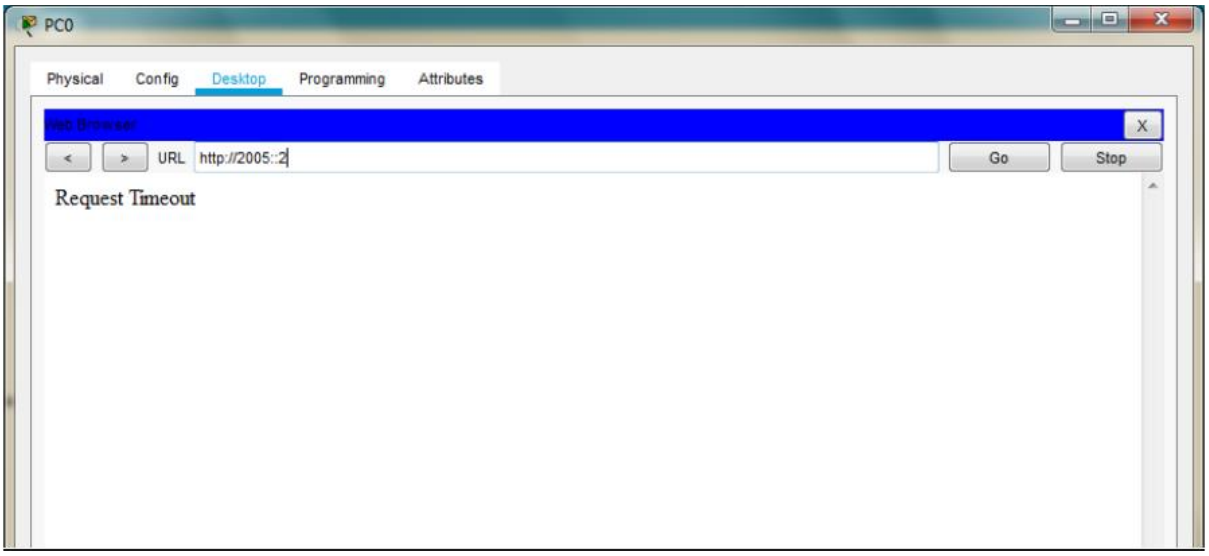
1) No HTTP or HTTPS allowed on server by any host
2) No www service accessible on the server by any host
3) Only ipv6 packets allowed towards the server

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

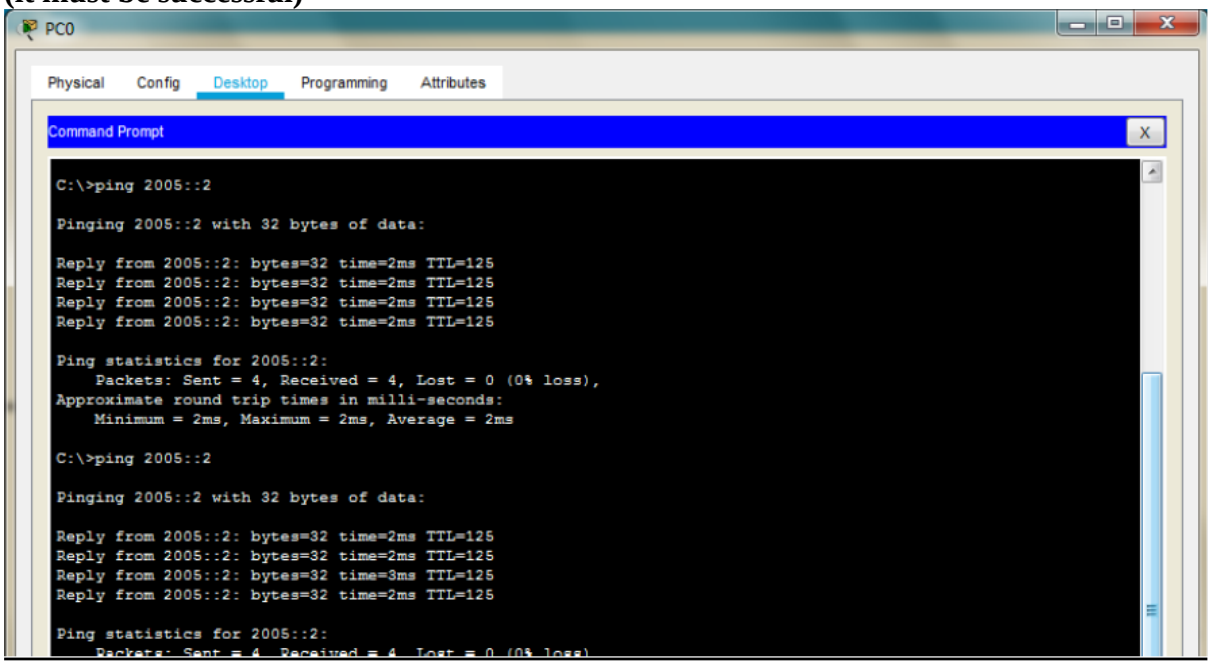
```
Router>
Router>enable
Router#conf t
Router(config)#ipv6 access-list dalmia
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq www
Router(config-ipv6-acl)#deny tcp any host 2005::2 eq 443
Router(config-ipv6-acl)#permit ipv6 any any
Router(config-ipv6-acl)#
Router(config-ipv6-acl)#exit
Router(config)#
Router(config)#int Se0/1/0
Router(config-if)#ipv6 traffic-filter dalmia 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.





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



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

****