Padmashri Annasaheb Jadhav Bharatiya Samaj Unnati Mandal's

B. N. N. College of Arts, Science & Commerce, Bhiwandi.

(Self-Funded Courses)

(Department of Computer Science)

CERTIFICATE

This is to certify that Mr. / Miss. **Harshad Sanjivan Dhotre**

Roll No. _____ Exam Seat No. _____

has Satisfactorily completed the Practical in **SOFTWARE DEFINED NETWORKING**

As laid down in the regulation of University of Mumbai for the purpose of MSc.Computer Science Semester I Examination 2022 – 2023.

Date: _____

Place: **BHIWANDI**

_____
In- Charge Professor

_____
Signature of External Examiners

_____
Signature of HOD

_____
Signature of Principal

INDEX

Practical No 1

Aim:Implement IP SLA (IP Service Level Agreement)

Introduction:

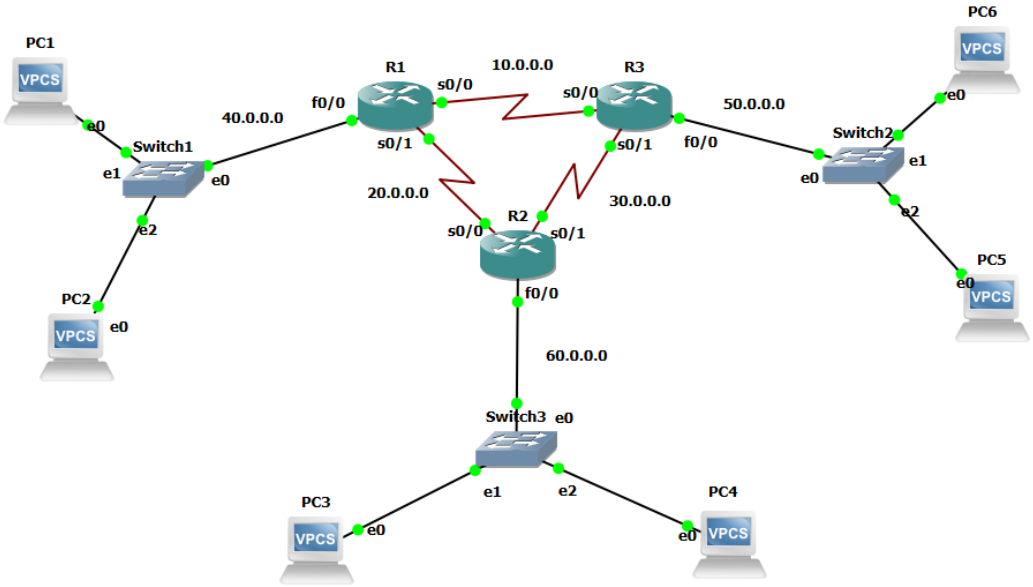
IP Service Level Agreement (IP SLA) is a feature that helps administrators collect information about network performance in real time. With increasing pressure on maintaining agreed-upon Service Level Agreements on Enterprises and ISPs alike, the IP SLA serves as a useful tool.

Any IP SLA test involves a source node and a destination node. For all discussions in this document, the source will always be an HP switch with IP SLA support. A destination can, in most cases, be any IP-enabled device. For some SLA types that expect a nonstandard response to a test packet, an "SLA responder" must be configured. An "SLA responder" is nothing but an HP switch with IP SLA configurations on it that enable it to respond to the test packet.

The IP SLA feature provides:

- Application-aware monitoring that simulates actual protocol packets.
- Predictable measures that aid in ease of deployment and help with assessment of existing network performance.
- Accurate measures of delay and packet loss for time-sensitive applications.
- End-to-end measurements to represent actual user experience.

Topology:



Commands:

R1#en

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#interface serial0

R1(config-if)#ip add 10.0.0.1 255.255.255.0

R1(config-if)#no shut

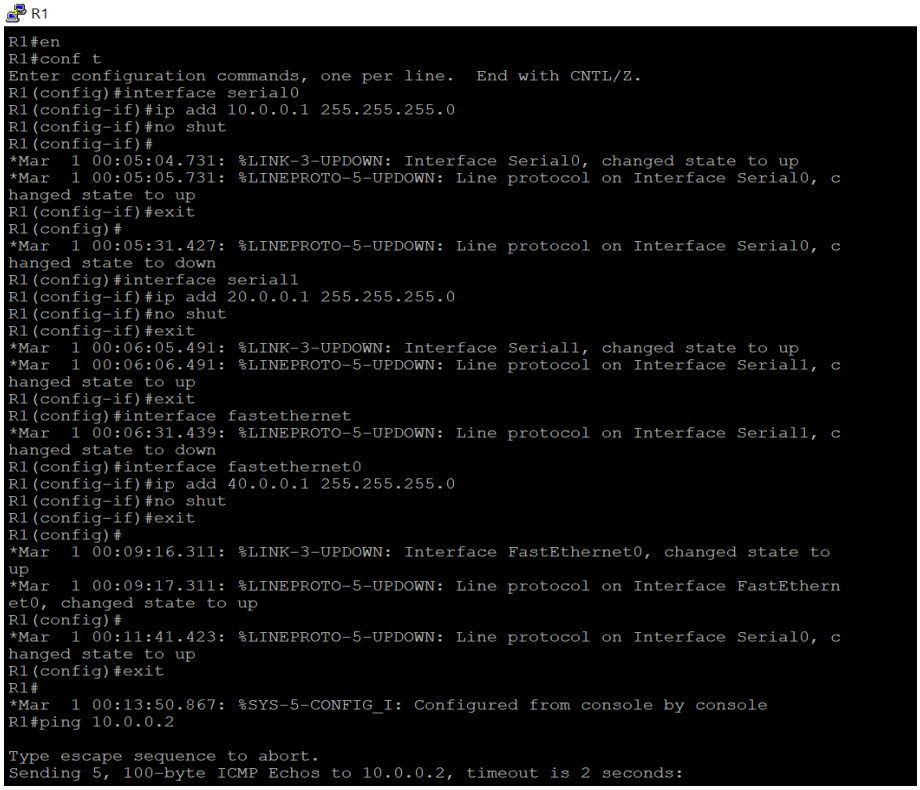
R1(config)#interface serial1

R1(config-if)#ip add 20.0.0.1 255.255.255.0

R1(config-if)#no shut

R1(config-if)#exit

Router1:



R1(config)#router ospf 1

R1(config-router)#network 10.0.0.0 0.255.255.255

% Incomplete command.

R1(config-router)#network 10.0.0.0 0.255.255.255 area 0

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#network 10.0.0.0 0.255.255.255 area 0
R1(config-router)#network 20.0.0.0 0.255.255.255 area 0
R1(config-router)#network 40.0.0.0 0.255.255.255 area 0
R1(config-router)#exit
R1(config)#exit
R1#
*Mar  1 02:49:31.755: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

Router2:

R2

```
*Mar  1 00:00:03.459: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to down
R2#en
R2#conft t
      ^
% Invalid input detected at '^' marker.

R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface serial0
R2(config-if)#ip add 20.0.0.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:18:31.331: %LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar  1 00:18:32.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, c
hanged state to up
R2(config)#interface serial1
R2(config-if)#ip add 30.0.0.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#in
*Mar  1 00:19:18.091: %LINK-3-UPDOWN: Interface Serial1, changed state to up
*Mar  1 00:19:19.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to up
R2(config)#interface fastethernet0
R2(config-if)#ip add 60.0.0.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#ex
*Mar  1 00:19:53.979: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to
up
*Mar  1 00:19:54.979: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0, changed state to up
R2(config)#exit
R2#
*Mar  1 00:19:58.495: %SYS-5-CONFIG_I: Configured from console by console
R2#ping 30.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/11/28 ms
R2#
```

Router3:

R3

```
istratively down
*Mar  1 00:00:02.407: %LINK-5-CHANGED: Interface Serial1, changed state to admin
istratively down
*Mar  1 00:00:03.343: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0, changed state to down
*Mar  1 00:00:03.407: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, c
hanged state to down
*Mar  1 00:00:03.407: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to down
R3#en
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface serial0
R3(config-if)#ip add 10.0.0.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Mar  1 00:11:32.887: %LINK-3-UPDOWN: Interface Serial0, changed state to up
*Mar  1 00:11:33.887: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, c
hanged state to up
R3(config)#interface serial1
R3(config-if)#ip add 30.0.0.1 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Mar  1 00:12:17.795: %LINK-3-UPDOWN: Interface Serial1, changed state to up
*Mar  1 00:12:18.795: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to up
R3(config)#interface fastethernet0
R3(config-if)#ip add
*Mar  1 00:12:41.415: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to down
R3(config-if)#ip add 50.0.0.1 255.255.255.0
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#
*Mar  1 00:13:28.935: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to
up
*Mar  1 00:13:29.935: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0, changed state to up
R3(config)#
*Mar  1 00:19:31.411: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, c
hanged state to up
R3(config)#
```

PC1:

PC1 - PuTTY

```
PC1> ip 40.0.0.2 255.255.255.0 40.0.0.1
Checking for duplicate address...
PC1 : 40.0.0.2 255.255.255.0 gateway 40.0.0.1

PC1> ping 60.0.0.2
60.0.0.2 icmp_seq=1 timeout
84 bytes from 60.0.0.2 icmp_seq=2 ttl=62 time=3.059 ms
84 bytes from 60.0.0.2 icmp_seq=3 ttl=62 time=1.285 ms
84 bytes from 60.0.0.2 icmp_seq=4 ttl=62 time=3.567 ms
84 bytes from 60.0.0.2 icmp_seq=5 ttl=62 time=3.691 ms

PC1>
```

PC6:

PC6 - PuTTY

```
PC6> ping 40.0.0.2
40.0.0.2 icmp_seq=1 timeout
40.0.0.2 icmp_seq=2 timeout
40.0.0.2 icmp_seq=3 timeout
84 bytes from 40.0.0.2 icmp_seq=4 ttl=62 time=1.711 ms
84 bytes from 40.0.0.2 icmp_seq=5 ttl=62 time=1.151 ms

PC6>
```

Practical No 2

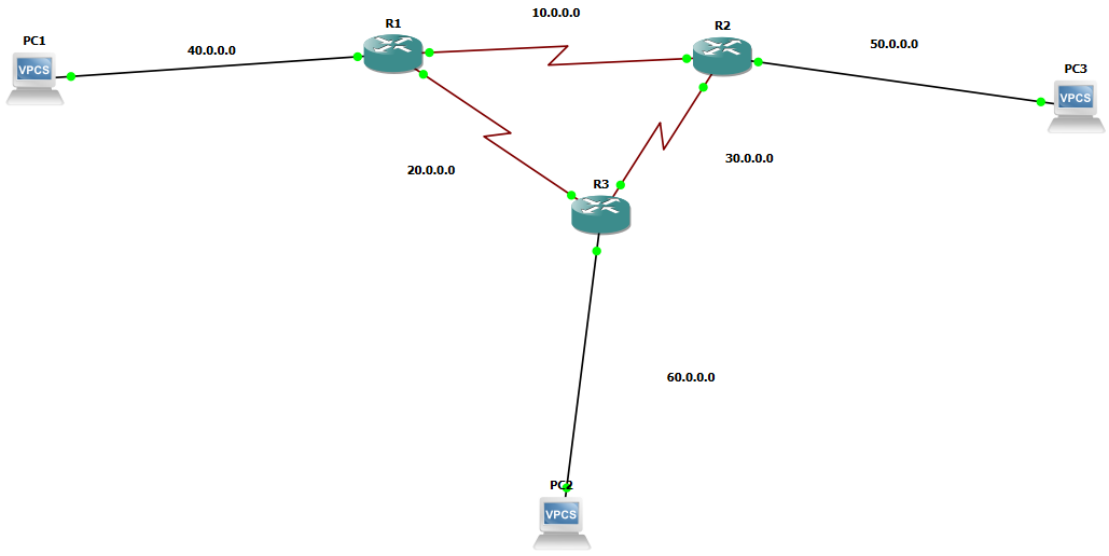
Aim: Implement BGP Communities

Introduction:

Border Gateway Protocol (BGP) refers to a gateway protocol that enables the internet to exchange routing information between autonomous systems (AS). As networks interact with each other, they need a way to communicate. This is accomplished through peering. BGP makes peering possible. Without it, networks would not be able to send and receive information with each other.

When you have a network router that connects to other networks, it does not know which network is the best one to send its data to. BGP takes into consideration all the different peering options a router has and chooses the one that is closest to where the router is.

Topology:



Commands:

R1#en

R1#conf t

R1(config)#interface serial4/0

R1(config-if)#ip add 10.0.0.1 255.255.255.0

R1(config-if)#no shut

R1(config-if)#exit

R1(config)#interface serial4/1

R1(config-if)#ip add 20.0.0.1 255.255.255.0

R1(config-if)#no shut

R1(config-if)#exit

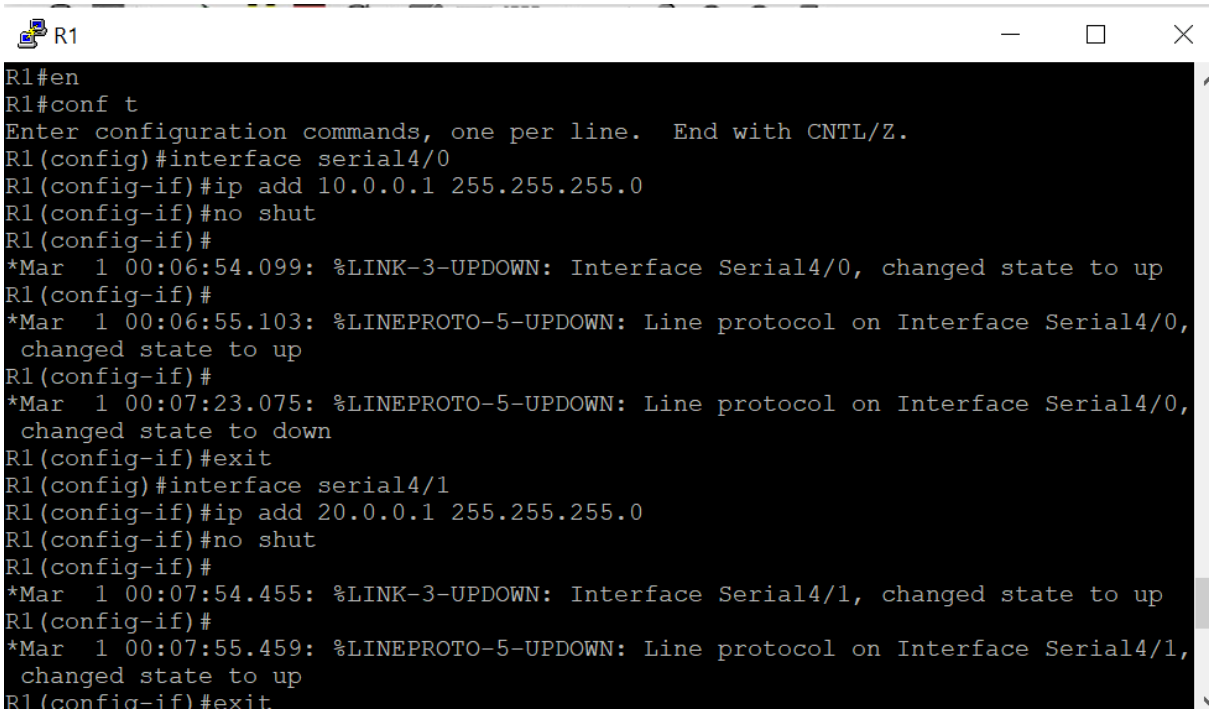
R1(config)#interface ethernet1/0

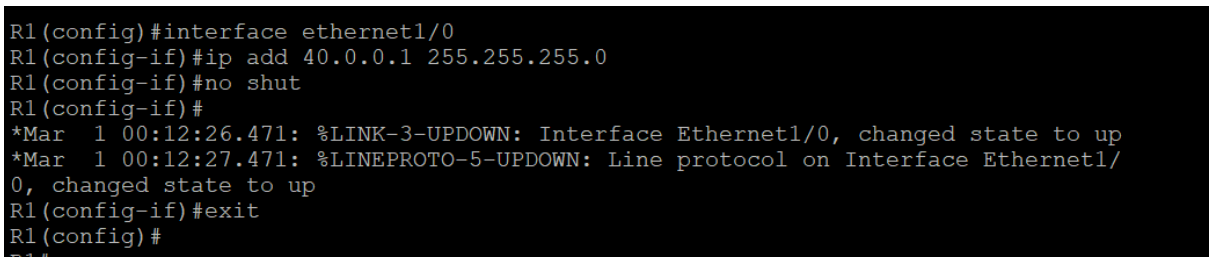
R1(config-if)#ip add 40.0.0.1 255.255.255.0

R1(config-if)#no shut

R1(config-if)#exit

Router 1:





Router2:

```
R2#en
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface serial4/0
R2(config-if)#ip add 10.0.0.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#
*Mar  1 00:09:58.151: %LINK-3-UPDOWN: Interface Serial4/0, changed state to up
R2(config-if)#
*Mar  1 00:09:59.155: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/0,
 changed state to up
R2(config-if)#exit
R2(config)#interface serial4/1
R2(config-if)#ip add 30.0.0.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#
*Mar  1 00:11:13.107: %LINK-3-UPDOWN: Interface Serial4/1, changed state to up
R2(config-if)#
*Mar  1 00:11:14.111: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/1,
 changed state to up
R2(config-if)#exit
R2(config)#
*Mar  1 00:11:43.003: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/1,
 changed state to down
R2(config)#interface ethernet1/0
R2(config-if)#ip add 50.0.0.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#e
*Mar  1 00:13:03.719: %LINK-3-UPDOWN: Interface Ethernet1/0, changed state to up
*Mar  1 00:13:04.719: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/
0, changed state to up
R2(config-if)#exit
R2(config)#
```

Router3:

```
R3#en
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface serial4/0
R3(config-if)#ip add 20.0.0.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#
*Mar  1 00:12:29.487: %LINK-3-UPDOWN: Interface Serial4/0, changed state to up
R3(config-if)#
*Mar  1 00:12:30.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/0,
 changed state to up
R3(config-if)#exit
R3(config)#interface serial4/1
R3(config-if)#ip add 30.0.0.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#
*Mar  1 00:13:20.583: %LINK-3-UPDOWN: Interface Serial4/1, changed state to up
R3(config-if)#
*Mar  1 00:13:21.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial4/1,
 changed state to up
R3(config-if)#exit
R3(config)#interface ethernet1/0
R3(config-if)#ip add 60.0.0.2 255.255.255.0
R3(config-if)#no shut
R3(config-if)#
*Mar  1 00:14:43.263: %LINK-3-UPDOWN: Interface Ethernet1/0, changed state to up
*Mar  1 00:14:44.263: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/
0, changed state to up
R3(config-if)#exit
R3(config)#
```

PC1:

PC1 - PuTTY

```
PC1> ip 40.0.0.2 255.255.255.0 40.0.0.1
Checking for duplicate address...
PC1 : 40.0.0.2 255.255.255.0 gateway 40.0.0.1

PC1> ping 40.0.0.1
84 bytes from 40.0.0.1 icmp_seq=1 ttl=255 time=15.969 ms
84 bytes from 40.0.0.1 icmp_seq=2 ttl=255 time=16.574 ms
84 bytes from 40.0.0.1 icmp_seq=3 ttl=255 time=15.656 ms
84 bytes from 40.0.0.1 icmp_seq=4 ttl=255 time=16.148 ms
84 bytes from 40.0.0.1 icmp_seq=5 ttl=255 time=16.002 ms
```

PC2:

PC3 - PuTTY

```
PC3> ip 50.0.0.2 255.255.255.0 50.0.0.1
Checking for duplicate address...
PC1 : 50.0.0.2 255.255.255.0 gateway 50.0.0.1

PC3> ping 50.0.0.1
84 bytes from 50.0.0.1 icmp_seq=1 ttl=255 time=54.038 ms
84 bytes from 50.0.0.1 icmp_seq=2 ttl=255 time=31.551 ms
84 bytes from 50.0.0.1 icmp_seq=3 ttl=255 time=16.774 ms
84 bytes from 50.0.0.1 icmp_seq=4 ttl=255 time=30.388 ms
84 bytes from 50.0.0.1 icmp_seq=5 ttl=255 time=16.107 ms
```

PC3:

PC2 - PuTTY

```
PC2> ip 60.0.0.3 255.255.255.0 60.0.0.1
Checking for duplicate address...
PC1 : 60.0.0.3 255.255.255.0 gateway 60.0.0.1

PC2> ping 60.0.0.2
84 bytes from 60.0.0.2 icmp_seq=1 ttl=255 time=22.422 ms
84 bytes from 60.0.0.2 icmp_seq=2 ttl=255 time=16.856 ms
84 bytes from 60.0.0.2 icmp_seq=3 ttl=255 time=16.127 ms
84 bytes from 60.0.0.2 icmp_seq=4 ttl=255 time=22.488 ms
84 bytes from 60.0.0.2 icmp_seq=5 ttl=255 time=16.297 ms
```

Commands for Configuring BGP:

R1#en

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#router bgp 100

R1(config-router)#neighbor 10.0.0.2 remote-as 200

R1(config-router)#neighbor 20.0.0.2 remote-as 300

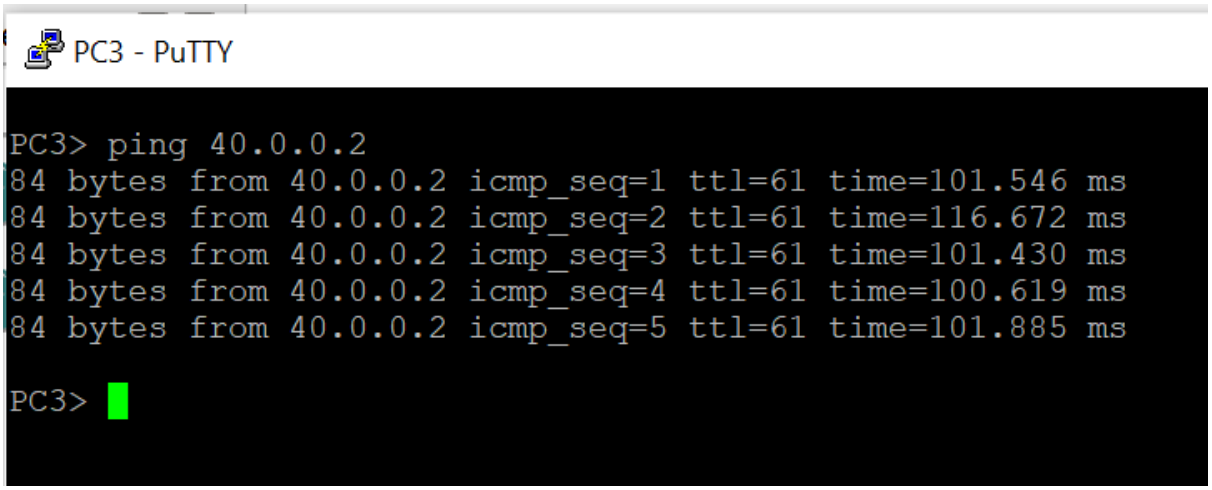
R1(config-router)#network 40.0.0.0 mask 255.255.255.0

```
R3(config)#router bgp 300
R3(config-router)#neighbor 30.0.0.1 remote-as 200
R3(config-router)#neighbor 30.0.0.1 remote-as 200
*Mar  1 00:56:51.359: %BGP-5-ADJCHANGE: neighbor 30.0.0.1 Up
R3(config-router)#neighbor 20.0.0.1 remote-as 100
R3(config-router)#
*Mar  1 00:57:57.627: %BGP-5-ADJCHANGE: neighbor 20.0.0.1 Up
R3(config-router)#network 60.0.0.0 mask 255.255.255.0
R3(config-router)#exit
R3(config)#exit
R3#
*Mar  1 00:58:37.395: %SYS-5-CONFIG_I: Configured from console by console
R3#show ip route
Codes: 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
       i - IS-IS, su - IS-IS summary, 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

     50.0.0.0/24 is subnetted, 1 subnets
B       50.0.0.0 [20/0] via 30.0.0.1, 00:01:51
     20.0.0.0/24 is subnetted, 1 subnets
C       20.0.0.0 is directly connected, Serial4/0
     40.0.0.0/24 is subnetted, 1 subnets
B       40.0.0.0 [20/0] via 20.0.0.1, 00:00:45
     60.0.0.0/24 is subnetted, 1 subnets
C       60.0.0.0 is directly connected, Ethernet1/0
     30.0.0.0/24 is subnetted, 1 subnets
C       30.0.0.0 is directly connected, Serial4/1
```

Pinging From pc3 to pc1:



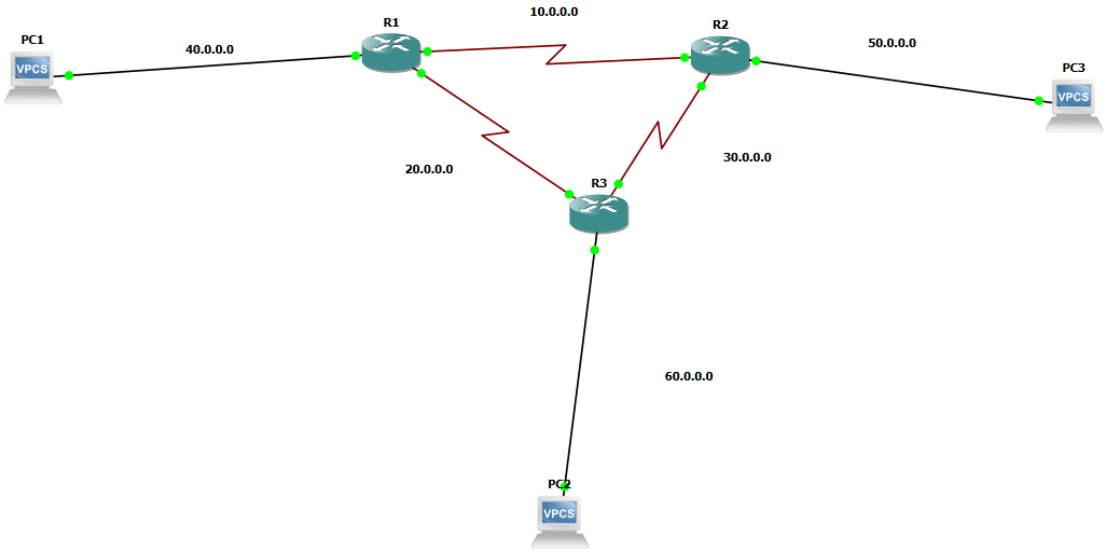
Practical No 3

Access Control List:

Access Control List (ACL) refers to a specific set of rules used for filtering network traffic, especially in computer security settings. ACLs also allow specific system objects such as directories or file access to authorized users and denies access to unauthorized users.

ACLs are mainly found in network devices with packet filtering capabilities including routers and switches.

Topology:



1.Standard ACL:

This type allows you to only evaluate packet source IP addresses. They are not as powerful as extended ACLs but use less computing power. They also use numbers 1300-1999 or 1-99 so that the router can identify the specific address as the source IP address.

Commands of Standard ACL:

R2(config)#ip access-list standard blocklist

R2(config-std-nacl)#deny 40.0.0.0 0.255.255.255

R2(config-std-nacl)#permit any

R2(config-std-nacl)#exit

R2(config)#interface ethernet1/0

R2(config-if)#ip access-group blocklist out

R2(config-if)#exit

R2(config)#exit

R2#show access-list

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#ip access-list standard blocklist
R2(config-std-nacl)#deny 40.0.0.0 0.255.255.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#exit
R2(config)#interface ethernet1/0
R2(config-if)#ip access-group blocklist out
R2(config-if)#exit
R2(config)#exit
R2#
*Mar  1 00:04:36.331: %SYS-5-CONFIG_I: Configured from console by console
R2#show access-list
Standard IP access list blocklist
    10 deny   40.0.0.0, wildcard bits 0.255.255.255
    20 permit any
R2#
```

PC3 - PuTTY

```
PC3> ping 40.0.0.2
40.0.0.2 icmp_seq=1 timeout
40.0.0.2 icmp_seq=2 timeout
40.0.0.2 icmp_seq=3 timeout
40.0.0.2 icmp_seq=4 timeout
40.0.0.2 icmp_seq=5 timeout
```

2.Extended ACL:

These types of ACL allow you to block source and destination for specific hosts or the whole network. With Extended ACLs it's possible to filter traffic based on protocols (IP, TCP, ICMP, and UDP).

Commands of Standard ACL:

R3#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R3(config)#ip access-list extended 100

R3(config-ext-nacl)#1 permit 60.0.0.0 0.0.0.255 20.0.0.2 0.0.0.255 echo-reply

R3(config-ext-nacl)#permit ip any any

R3(config-ext-nacl)#deny tcp 50.0.0.0 0.0.0.255 60.0.0.0 0.0.0.255

R3(config-ext-nacl)#exit

R3(config)#interface ethernet 1/0

R3(config-if)#ip access-group 100 out

R3(config-if)#exit

R3(config)#exit

R3#show access-list

```
R3#show ip route
Codes: 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
       i - IS-IS, su - IS-IS summary, 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

     50.0.0.0/24 is subnetted, 1 subnets
B       50.0.0.0 [20/0] via 30.0.0.1, 00:52:25
     20.0.0.0/24 is subnetted, 1 subnets
C       20.0.0.0 is directly connected, Serial4/0
     40.0.0.0/24 is subnetted, 1 subnets
B       40.0.0.0 [20/0] via 20.0.0.1, 00:51:18
     60.0.0.0/24 is subnetted, 1 subnets
C       60.0.0.0 is directly connected, Ethernet1/0
     30.0.0.0/24 is subnetted, 1 subnets
C       30.0.0.0 is directly connected, Serial4/1
R3#show access-list
Extended IP access list 100
    10 permit ip any any
    20 deny tcp 50.0.0.0 0.0.0.255 60.0.0.0 0.0.0.255
R3#
```