



Faculty of Engineering and Technology

Computer Engineering Department

ENCS5321

BGP Scenario: Multi-AS Topology with Policy-Based Traffic
Engineering

Task - Report

Prepared by:

Manar Shawahni, 1201086.

Layan Abuershaid, 1200098.

Instructor's Name: Dr. Imad Tartir.

Section: 1

12 January, 2025

Table of Contents

Introduction.....	3
Network topology	3
Initial Configuration.....	4
Configuration of Open Shortest Path First (OSPF)	4
Enable OSPF in AS100.....	4
Enable OSPF in AS300.....	5
Enable Static Routing in AS200	6
Enable Routing Information Protocol (RIP) in AS400.....	8
Border Gateway Protocol (BGP) Configuration.....	10
iBGP within each AS	10
eBGP between ASes	16
Traffic Engineering.....	17
Route Optimization: Aggregate Prefixes in AS200.....	17
Pre-Optimization State.....	17
Post-Optimization State	19
Redundancy Testing: Simulating Link Failures	20
Pre-Failure State.....	20
Simulating the Link Failure	21
Post-Failure State	21
Restoring the Original Link	22
Test Scenarios	23
OSPF Neighbor Verification.....	23
On Router Ro4 at AS300	23
On Router R9 at AS100	23
RIP Route Propagation Verification on AS400	23
BGP Route Advertisement Verification	24
Loopback Interface Reachability	25
Static Route Verification in AS200	25
PC1 ⇔ PC2 tests	25

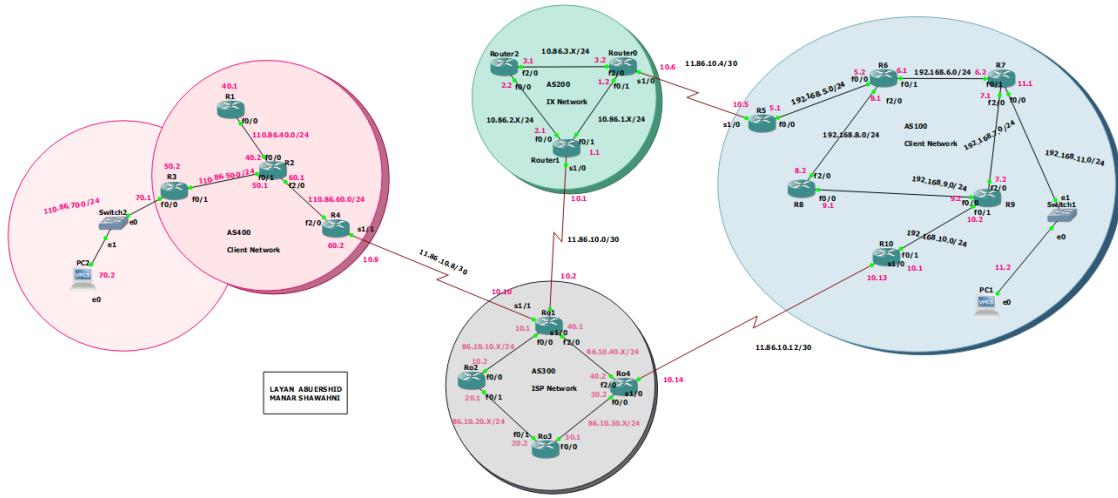
Conclusion	26
------------------	----

Introduction

The objective of this project is to design a multi-AS BGP environment to demonstrate key BGP features such as route filtering, traffic engineering, and redundancy. The scope includes configuring four autonomous systems (ASes), setting up iBGP and eBGP peering, and implementing traffic monitoring and troubleshooting techniques. GNS3 was used as the primary tool for the implementation and testing.

Network topology

For this project, we designed the IP Addressing Scheme according to the project requirements. The IP addresses were systematically derived from **Manar's ID (1201086)** and assigned to the links and networks within each AS, ensuring consistency and compliance with the requirements. The topology consists of four autonomous systems (AS100, AS200, AS300, AS400), with routers, links, and IP addresses structured accordingly.



AS Descriptions:

- **AS100:** Represents an enterprise client network, managing internal traffic and connecting to other ASes for external communication.
 - **AS200:** Acts as an IX (Internet Exchange) provider, enabling interconnection and route exchange between AS100, AS300, and AS400.
 - **AS300:** Functions as an ISP (Internet Service Provider), bridging client ASes and facilitating internet connectivity.
 - **AS400:** Another enterprise network with its own internal structure and connections to external ASes for redundancy and service provision.

Assigning IPs

We started by assigning IP addresses to the interfaces on each router in AS100, AS200, AS300, and AS400, and these commands which we used:

```
interface <interface_name>
```

```
ip address <ip_address> <subnet_mask>
```

```
no shutdown
```

and then we used this command to make sure the IPs were correctly assigned and the interfaces were up:

```
show ip interface brief
```

At AS100 for example, in router R6:

```
R6#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R6(config)#inter
R6(config)#interface fa
R6(config-if)#ip address
R6(config-if)#ip address 192.168.5.1 255.255.255.0
R6(config-if)#no shut
R6(config-if)#exit
R6(config)#do wr
Building configuration...
[OK]
R6(config)#
```

```
R6#sh ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    192.168.5.2   YES NVRAM up        up
Serial0/0          unassigned     YES NVRAM administratively down
FastEthernet0/1    192.168.6.1   YES NVRAM up        up
Serial0/1          unassigned     YES NVRAM administratively down
Serial0/2          unassigned     YES NVRAM administratively down
Serial0/0          unassigned     YES NVRAM administratively down
Serial1/1          unassigned     YES NVRAM administratively down
Serial1/1/1         unassigned     YES NVRAM administratively down
Serial1/2          unassigned     YES NVRAM administratively down
Serial1/3          unassigned     YES NVRAM administratively down
FastEthernet2/0    192.168.8.1   YES NVRAM up        up
```

Initial Configuration

Configuration of Open Shortest Path First (OSPF)

Enable OSPF in AS100

We done this by giving OSPF a process ID, which we chose 1, using this command:

```
router ospf <process_id>
```

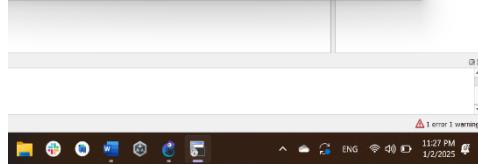
```
router ospf 1
```

Next, we configure OSPF to advertise networks by specifying the network and its area. Area 0 known as the backbone, is the main area used to organize networks.

```
network <network_address> <wildcard_mask> area <area_number>
```

for R5, it was one network:

```
R5#config#router ospf 1
R5(config-router)#network 192.168.5.0 0.0.0.255
% Incomplete command.
R5(config-router)#network 192.168.5.0 0.0.0.255 area 0
R5(config-router)#exit
R5#Building configuration...
[OK]
R5(config)#
```



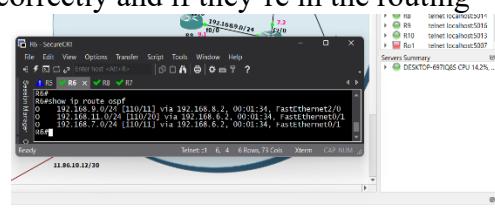
For R6, we done for the three directly connected:

```
R6(config)#router ospf 1
R6(config-router)#network 192.168.5.0 0.0.0.255 area 0
R6(config-router)#network 192.168.6.0 0.0.0.255 area 0
R6(config-router)#network 192.168.6.0 0.0.0.255 area 0
*Mar 1 00:04:52.023: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.5.1 on FastEther
net0/0 from LOADING to FULL, Loading Done
R6(config-router)#do wr
R6(config-router)#exit
R6(config)#Building configuration...
[OK]
R6(config)#
```



Then, we check if OSPF is advertising the routes correctly and if they're in the routing table: **show ip route ospf**

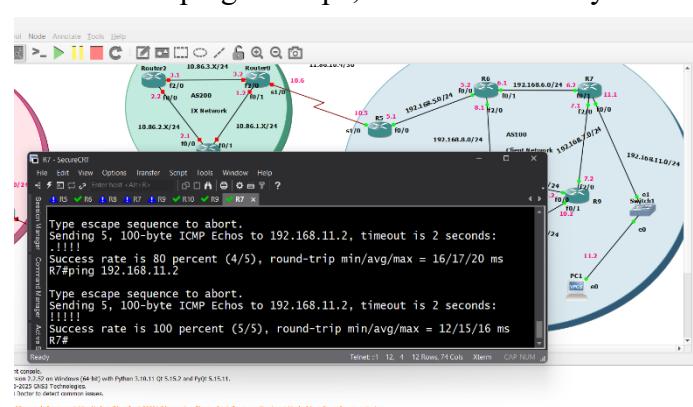
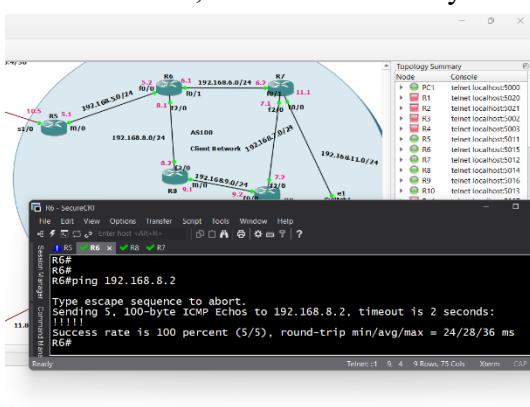
which for the other OSPF-learned routes



We tried to ping 192.168.8.2 from R6,

as shown, it done successfully.

I pinged the pc, and done correctly



Enable OSPF in AS300

We done the same steps as above, for example in Ro1:

```

Ro1#show running-config | section router ospf
router ospf 1
log-adjacency-changes

network 86.10.10.0 0.0.0.255 area 0
network 86.10.40.0 0.0.0.255 area 0
Ro1#
Ro1#show ip ospf interface brief
Interface PID Area IP Address/Mask Cost State Nbrs F/C
Fa2/0 1 0 86.10.40.1/24 1 BDR 1/1
Fa0/0 1 0 86.10.10.1/24 10 BDR 1/1

```

The image shows the OSPF configuration on Ro1, where the networks 86.10.10.0/24 and 86.10.40.0/24 are advertised in area 0. The OSPF interface details confirm that both FastEthernet interfaces (Fa2/0 and Fa0/0) with one neighbor each.

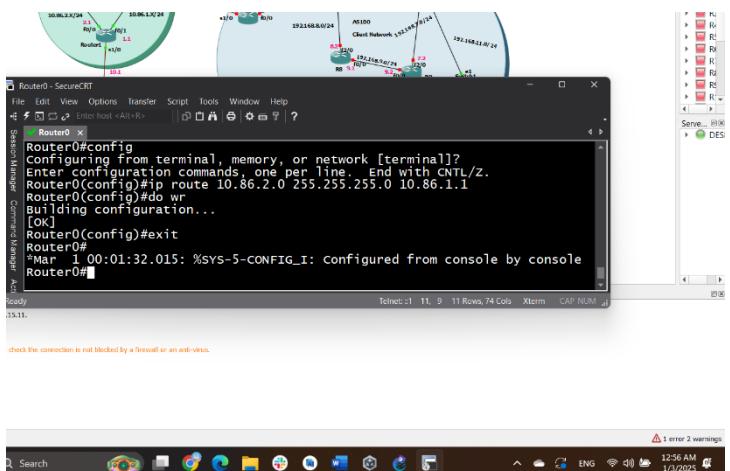
The image shows the OSPF configuration on Ro1, where the networks 86.10.10.0/24 and 86.10.40.0/24 are advertised in area 0. The OSPF interface details confirm that both FastEthernet interfaces (Fa2/0 and Fa0/0) with one neighbor each.

And we done this for all other routers.

Enable Static Routing in AS200

Static routing requires specifying the destination network, subnet mask, and the next-hop IP address or exit interface. Use the command:

```
ip route <destination network> <subnet mask> <next hop or exit interface>
```



For example, for **Router0** has two directly connected networks (10.86.1.0/24 and 10.86.3.0/24), and we need to reach **10.86.2.0/24** via another router (Router1 at 10.86.1.1).

And we repeated this for each router in this AS.

To check if the static routes are installed correctly in the routing table using:

```
show ip route
```

```

Router0#sh 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

  10.0.0.0/24 is subnetted, 3 subnets
C    10.86.1.0 is directly connected, FastEthernet0/1
S    10.86.2.0 [1/0] via 10.86.1.1
C    10.86.3.0 is directly connected, FastEthernet2/0

```

The image shows the routing table of Router0. It indicates that static routes are configured correctly.

Router0#ping 10.86.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.86.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/36 ms
Router0#
Router0#ping 10.86.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.86.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/60/64 ms
Router0#

The image shows successful ping results from Router0 to the IPs 10.86.2.1 and 10.86.2.2. This indicates that Router0 has reachability to devices in the 10.86.2.0/24 subnet.

Similar to Router0, Router1 and Router2 were configured with static routes and verified then using the ping command and show ip route to ensure reachability and correct route advertisement. The results are consistent, as shown in the accompanying images.

A static route was configured on **Router1** to direct traffic for the **10.86.3.0/24** network via the next hop **10.86.2.2**. The route was added using the ip route command, and the configuration was saved.

```
Router1#sh 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, o - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 3 subnets
    C 10.86.1.0 is directly connected, FastEthernet0/1
    C 10.86.2.0 is directly connected, FastEthernet0/0
    S 10.86.3.0 [1/0] via 10.86.2.2
```

Router1#ping 10.86.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.86.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
Router1#

The images confirms that the static route for the **10.86.3.0/24** network is active and reachable via the next hop **10.86.2.2**. The ping command further verifies connectivity to **10.86.3.1**, with a 100% success rate, so the static routing configuration is correct and functional.

On **Router2**, a static route was added to direct traffic for the **10.86.2.0/24** network through the next hop **10.86.3.1**. The routing table confirmed that the route was active and functional.

```
Router2#sh 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, o - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/24 is subnetted, 3 subnets
    S 10.86.1.0 [1/0] via 10.86.2.1
    C 10.86.2.0 is directly connected, FastEthernet0/0
    C 10.86.3.0 is directly connected, FastEthernet2/0
```

Router2#ping 10.86.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.86.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
Router2#

The images above indicate that traffic to the 10.86.1.0/24 network is routed through the next hop 10.86.2.1, confirming proper configuration. The second image done successful ping to 10.86.1.1, showing that connectivity is established and the static route is working as intended.

Enable Routing Information Protocol (RIP) in AS400

To enable RIP as the Interior Gateway Protocol (IGP) in AS400, the following steps were performed.

We entered RIP configuration mode using this command:

```
router rip
```

We added the networks connected to router:

```
network <network address>
```

for R2,

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 110.86.40.0
R2(config-router)#network 110.86.50.0
R2(config-router)#network 110.86.60.0
R2(config-router)#network 4.0.0.0
R2(config-router)#no auto-summary
```

```
R2#sh 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 external type 1, N2 - OSPF external type 2
      E1 - OSPF summary, E2 - OSPF External type 2
      I1 - 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

          4.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R        4.4.4.4/32 [120/1] via 110.86.60.2, 00:00:08, FastEthernet2/0
R        4.0.0.0/0 [120/1] via 110.86.50.1, 00:00:10, FastEthernet2/0
R        4.0.0.0/32 [120/1] via 110.86.40.1, 00:00:10, FastEthernet0/0
C        4.4.4.2/32 is directly connected, Loopback0
R        4.4.4.3/32 [120/1] via 110.86.50.2, 00:00:12, FastEthernet0/1
          110.0.0.0/24 is subnetted, 4 subnets
C        110.86.50.0 is directly connected, FastEthernet1/0
C        110.86.50.0 is directly connected, FastEthernet0/1
C        110.86.40.0 is directly connected, FastEthernet0/0
R        110.86.70.0 [120/1] via 110.86.50.2, 00:00:14, FastEthernet0/1
R2#
```

and we repeated the same steps for R1, R3, and R4, adding their connected networks to the configuration.

Adding a New Client to AS400

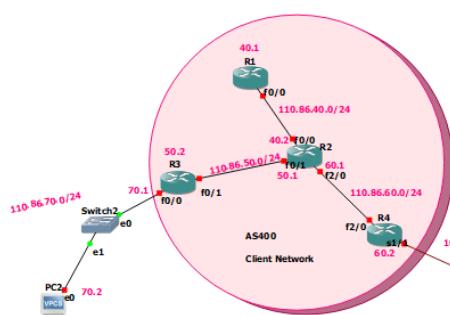
To add a new client (PC) to AS400:

1. Connecting a Switch:

We connected a switch to R3's FastEthernet interface.

2. Connecting the PC:

We connected the PC to the switch using an Ethernet cable.



Updating the RIP for R3, to include the new PC's network in RIP we did the same steps as before.

```
R3#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R3(Config)#router rip
R3(Config-router)#network 110.86.70.0
R3(Config-router)#exit
R3(config)#do wr
Building configuration...
[OK]
R3(Config)#
R3(Config)#
R3(Config)#
R3(config)#sh ip route
^
```

```
*Mar 1 00:42:27.347: %SYS-5-CONFIG_I: Configured from console by console
R3#sh 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
      110.0.0.0/24 is subnetted, 4 subnets
R    110.86.60.0 [120/1] via 110.86.50.1, 00:00:17, FastEthernet0/1
C    110.86.50.0 is directly connected, FastEthernet0/1
R    110.86.40.0 [120/1] via 110.86.50.1, 00:00:17, FastEthernet0/1
C    110.86.70.0 is directly connected, FastEthernet0/0
R3#
```

We had successful ping results from R1 to various destinations, including loopback and client networks (4.4.4.3, 4.4.4.4, 110.86.70.2). It confirms proper connectivity and reachability within AS400, demonstrating that RIP is functioning correctly for routing within the AS.

```
R1#ping 4.4.4.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/111/128 ms
R1#ping 4.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 92/102/112 ms
R1#ping 110.86.70.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 110.86.70.2, timeout is 2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 104/122/140 ms
R1#ping 110.86.70.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 110.86.70.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 104/126/148 ms
R1#
```

To config RIP at AS400 we used these commands:

`router rip`

`version 2`

`network <CONNECTED_NETWORK>`

`no auto-summary`

for example, **R1**:

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 110.86.40.0
R1(config-router)#network 4.0.0.0
R1(config-router)#no auto-summary
```

We used `sh ip route` to make sure:

```

R1#sh 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

        4.0.0.0/32 is subnetted, 4 subnets
R         4.4.4.4 [120/2] via 110.86.40.2, 00:00:20, FastEthernet0/0
C         4.4.4.1 is directly connected, Loopback0
R         4.4.4.2 [120/1] via 110.86.40.2, 00:00:20, FastEthernet0/0
R         4.4.4.3 [120/2] via 110.86.40.2, 00:00:20, FastEthernet0/0
        110.0.0.0/24 is subnetted, 4 subnets
R           110.86.60.0 [120/1] via 110.86.40.2, 00:00:20, FastEthernet0/0
R           110.86.50.0 [120/1] via 110.86.40.2, 00:00:20, FastEthernet0/0
C           110.86.40.0 is directly connected, FastEthernet0/0
R           110.86.70.0 [120/2] via 110.86.40.2, 00:00:22, FastEthernet0/0
R1#

```



Border Gateway Protocol (BGP) Configuration

iBGP within each AS

Establishing internal BGP (iBGP) sessions within each autonomous system (AS) to ensure full mesh connectivity and efficient route exchange between routers within the same AS.

Firstly, Loopback interfaces were added to each router in every AS to serve as the source for BGP neighbor relationships instead of physical interfaces. This ensures stability and reliability since loopback addresses remain active as long as the router is operational, regardless of physical link status. They also provide a consistent address for configuring BGP sessions and improve redundancy by remaining accessible through alternative paths in case of physical link failures.

Configure Loopback Interfaces:

We firstly assign a unique IP for each router loopback using these commands:

`interface loopback0`

`ip address <ip_address> 255.255.255.255`

`no shutdown`

This is an example of how we added the loopback IP in R1 (**AS400**):

```

R1(config)#interface loopback0
R1(config)#ip address 4.4.4.1 255.255.255.255
R1(config-if)#exit
R1(config)#do wr
Building configuration...
[OK]
R1(config)#

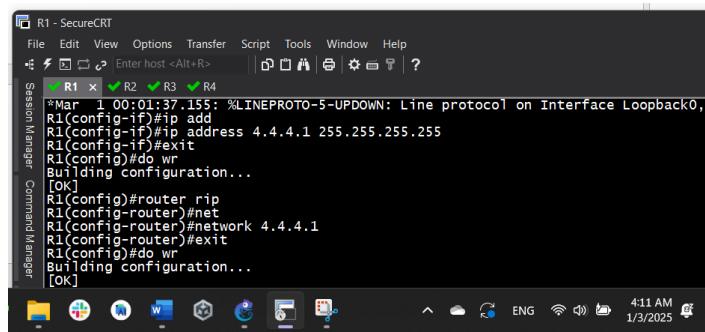
```



The rest of the router's loopback IPs:

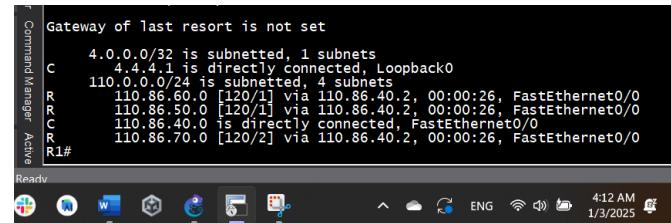
AS	Router	Loopback IP
AS400	R1	4.4.4.1
AS400	R2	4.4.4.2
AS400	R3	4.4.4.3
AS400	R4	4.4.4.4

Then, we advertised Loopback Interfaces in RIP to ensure that they are reachable by other routers within the same AS, allowing stable BGP neighbor relationships.



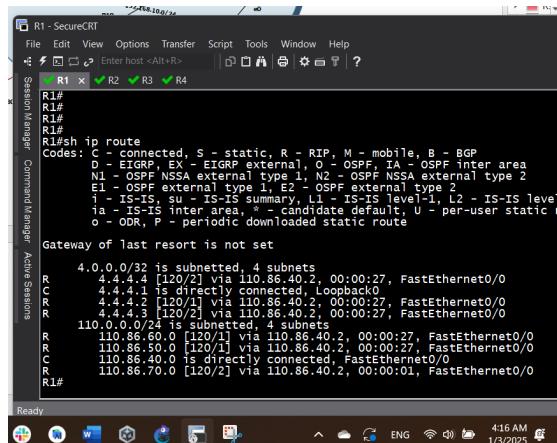
```
R1 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Session Manager Session Manager Active Sessions Active Sessions
Session Manager Command Manager Command Manager Active Active
R1# Mar 1 00:01:37.155: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
R1(config-if)#ip add
R1(config-if)#ip address 4.4.4.1 255.255.255.255
R1(config-if)#exit
R1(config)#do wr
Building configuration...
[OK]
R1(config)#router rip
R1(config-router)#net
R1(config-router)#network 4.4.4.1
R1(config-router)#exit
R1(config)#do wr
Building configuration...
[OK]
```

And make sure that it added in the routing table:



```
R1# Gateway of last resort is not set
C     4.0.0.0/32 is subnetted, 1 subnets
      4.4.4.1 is directly connected, Loopback0
R  110.0.0.0/24 is subnetted, 4 subnets
    110.86.60.0 [120/1] via 110.86.40.2, 00:00:26, FastEthernet0/0
    110.86.50.0 [120/1] via 110.86.40.2, 00:00:26, FastEthernet0/0
    C  110.86.40.0 is directly connected, FastEthernet0/0
    R  110.86.70.0 [120/2] via 110.86.40.2, 00:00:26, FastEthernet0/0
R1#
```

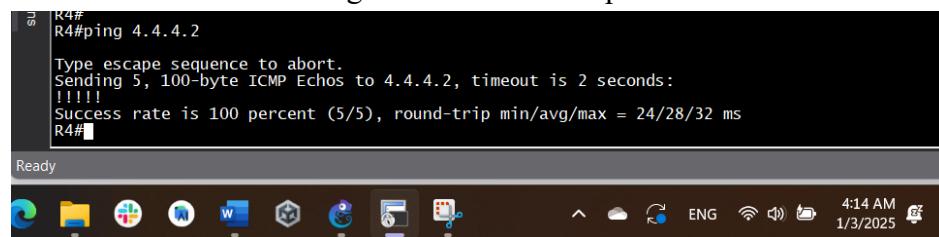
Then, we done this for all other routers in this AS400, and the final table for an R1 for example show:



```
R1# R1#
R1# R1#
R1# R1#sh 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, L1 - IS-IS level 1, L2 - IS-IS level 2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  4.0.0.0/32 is subnetted, 4 subnets
    R  4.4.4.4 [120/2] via 110.86.40.2, 00:00:27, FastEthernet0/0
    C  4.4.4.1 is directly connected, Loopback0
    R  4.4.4.2 [120/1] via 110.86.40.2, 00:00:27, FastEthernet0/0
    R  4.4.4.3 [120/1] via 110.86.40.2, 00:00:27, FastEthernet0/0
    R  110.0.0.0/24 is subnetted, 4 subnets
      R  110.86.60.0 [120/1] via 110.86.40.2, 00:00:27, FastEthernet0/0
      R  110.86.50.0 [120/1] via 110.86.40.2, 00:00:27, FastEthernet0/0
      C  110.86.40.0 is directly connected, FastEthernet0/0
      R  110.86.70.0 [120/2] via 110.86.40.2, 00:00:01, FastEthernet0/0
R1#
```

The final routing table for R1 confirms that all required loopback IPs and directly connected networks are advertised and reachable within AS400.

Testing from R4 to R2 loopback:



```
R4#ping 4.4.4.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
R4#
```

The terminal window shows a ping command being executed on router R4. The command is "ping 4.4.4.2". The output indicates 5 ICMP Echoes were sent to 4.4.4.2 with a timeout of 2 seconds. The success rate is 100% (5/5) with round-trip times ranging from 24 to 32 ms. The window title is "Ready" and the system tray shows the date and time as 4:14 AM, 1/3/2025.

Testing connectivity between the loopbacks verifies that the internal routing protocol (RIP) is functioning correctly and that the loopback interfaces are reachable, which is critical for establishing iBGP sessions.

To simplify, a table of loopback IPs for all routers and ASes can be included:

AS	Router	Loopback IP
AS100	R5	5.5.5.5
AS100	R6	6.6.6.6
AS100	R7	7.7.7.7
AS100	R8	8.8.8.8
AS100	R9	9.9.9.9
AS100	R10	10.10.10.10

AS200	Router0	10.1.0.1
AS200	Router1	10.1.1.1
AS200	Router2	10.1.2.1

AS300	Ro1	3.3.3.1
AS300	Ro2	3.3.3.2
AS300	Ro3	3.3.3.3
AS300	Ro4	3.3.3.4

Now, configure iBGP Peering:

We used these commands:

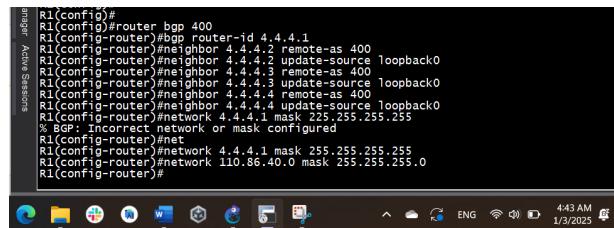
```
router bgp <AS NUMBER>
```

```
neighbor <LOOPBACK_IP> remote-as <AS NUMBER>
```

```
neighbor <LOOPBACK_IP> update-source loopback0
```

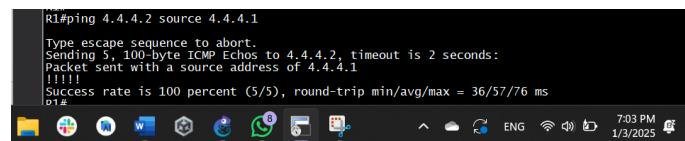
```
network <CONNECTED NETWORK> mask <SUBNET MASK>
```

For example, R1 at AS400:

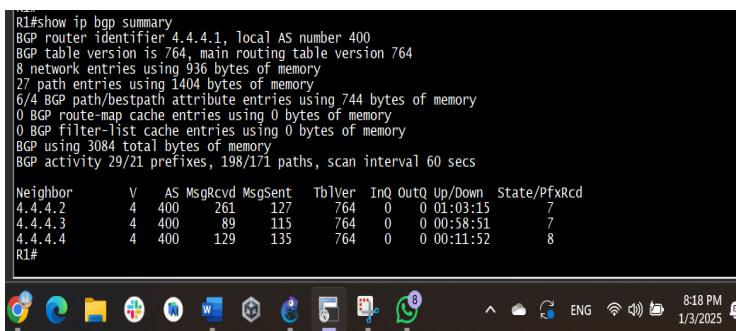


```
R1(config)#  
R1(config)#router bgp 400  
R1(config-router)#bgp router-id 4.4.4.1  
R1(config-router)#neighbor 4.4.4.1 remote-as 400  
R1(config-router)#neighbor 4.4.4.2 update-source loopback0  
R1(config-router)#neighbor 4.4.4.3 remote-as 400  
R1(config-router)#neighbor 4.4.4.4 update-source loopback0  
R1(config-router)#neighbor 4.4.4.5 update-source loopback0  
R1(config-router)#network 4.4.4.1 mask 255.255.255.255  
% BGP: Incorrect network or mask configured  
R1(config-router)#net  
R1(config-router)#network 4.4.4.1 mask 255.255.255.255  
R1(config-router)#network 110.86.40.0 mask 255.255.255.0  
R1(config-router)#
4:43 AM 1/3/2025
```

Pinging R2 loopback from R1 loopback successfully:



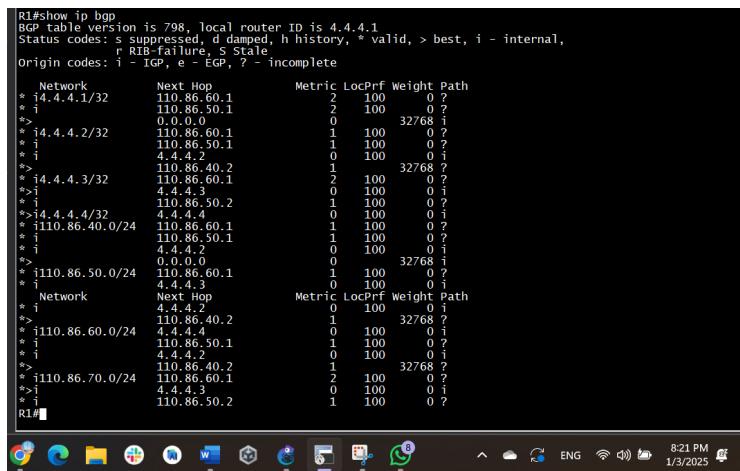
```
R1#ping 4.4.4.2 source 4.4.4.1  
Type escape sequence to abort.  
Sending 5 100-byte ICMP Echos to 4.4.4.2, timeout is 2 seconds:  
Packet sent with a source address of 4.4.4.1  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/57/76 ms  
0%#
7:03 PM 1/3/2025
```



```
R1#show ip bgp summary  
BGP router identifier 4.4.4.1, Local AS number 400  
BGP table version is 764, main routing table version 764  
8 network entries using 936 bytes of memory  
27 path entries using 1404 bytes of memory  
6/4 BGP path/bestpath attribute entries using 744 bytes of memory  
0 BGP route-map cache entries using 0 bytes of memory  
0 BGP filter-list cache entries using 0 bytes of memory  
BGP using 3084 total bytes of memory  
BGP activity 29/21 prefixes, 198/171 paths, scan interval 60 secs  
  
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd  
4.4.4.2 4 400 261 127 764 0 0 01:03:15 7  
4.4.4.3 4 400 89 115 764 0 0 00:58:51 7  
4.4.4.4 4 400 129 135 764 0 0 00:11:52 8  
R1#  
8:18 PM 1/3/2025
```

The `show ip bgp summary` output confirms that iBGP peering is active on R1 in AS400. The router has established sessions with three neighbors (4.4.4.4), and prefixes are being exchanged correctly, with 7, 8, and 8 prefixes received, verifying successful iBGP configuration and stable connectivity.

The output shows the routing table for R1 in AS400, with prefixes like 4.4.4.4/32 and /24 routes being propagated correctly. The Next Hop column confirms reachability via neighbors' loopback IPs, such as 110.86.50.1. Metrics like Weight and LocPrf indicate proper iBGP behavior, which verifying successful route propagation and connectivity within AS400.



```
R1#show ip bgp  
BGP table version is 798, Local router ID is 4.4.4.1  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S stale  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
Network Next Hop Metric LocPrf Weight Path  
* 4.4.4.1/32 110.86.60.1 2 100 0 ?  
* i 0.0.0.0 0 0 0 ?  
*> 4.4.4.2/32 110.86.60.1 1 100 0 ?  
* i 110.86.50.1 1 100 0 ?  
* i 4.4.4.2 0 100 0 i  
*> 110.86.40.2 110.86.40.2 1 100 0 ?  
* i 4.4.4.3 0 100 0 i  
*> 110.86.50.2 110.86.50.2 1 100 0 ?  
* i 4.4.4.4 0 100 0 i  
*> 110.86.60.1/24 110.86.60.1 1 100 0 ?  
* i 110.86.50.1 1 100 0 ?  
* i 110.86.50.1 0 100 0 i  
*> 110.86.60.0/24 110.86.60.0 1 100 0 ?  
* i 110.86.50.1 1 100 0 ?  
* i 4.4.4.2 0 100 0 i  
*> 110.86.40.2 110.86.40.2 1 100 0 ?  
* i 4.4.4.3 0 100 0 i  
*> 110.86.70.0/24 110.86.60.1 2 100 0 ?  
* i 4.4.4.3 0 100 0 i  
*> 110.86.50.2 110.86.50.2 1 100 0 ?  
R1#  
8:21 PM 1/3/2025
```

For AS300 we did the same steps and config them to OSPF, for example R01:

```
R01(config)#interface loopback0
R01(config-if)#ip address 3.3.3.1 255.255.255.255
R01(config-if)#do wr
Building configuration...
[OK]
R01(config-if)#ex
R01(config)#router ospf 1
R01(config-router)#network 3.3.3.1 0.0.0.0 area 0
R01(config-router)#do wr
Building configuration...
[OK]
R01(config-router)#exit
R01(config)#
Ready


12:51 AM  
1/4/2025

R01>#show ip bgp
R01(config)#router bgp 300
R01(config-router)#nei
R01(config-router)#neighbor 3.3.3.2 remo
R01(config-router)#neighbor 3.3.3.2 remot
R01(config-router)#neighbor 3.3.3.2 update-as 300
R01(config-router)#neighbor 3.3.3.2 upda
R01(config-router)#neighbor 3.3.3.2 update-source loop
R01(config-router)#neighbor 3.3.3.2 update-source loopback0
R01(config-router)#neighbor 3.3.3.3 remote-as 300
R01(config-router)#neighbor 3.3.3.3 update-source loopback0
R01(config-router)#neighbor 3.3.3.4 remote-as 300
R01(config-router)#neighbor 3.3.3.4 update-source loopback0
R01(config-router)#do wr
Building configuration...
[OK]
R01(config-router)#
*Mar 1 00:58:43.287: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed st
R01(config-router)#
R01(config-router)#
R01(config-router)#network 86.10.10.0 mask 255.255.255.0
R01(config-router)#network 86.10.40.0 mask 255.255.255.0
R01(config-router)#do wr
Building configuration...
[OK]
R01(config-router)#

11:14 AM  
1/4/2025
```

And making sure by `show ip bgp summary` and `show ip bgp` commands:

```

R01#show ip bgp summary
BGP router identifier 3.3.3.1, local AS number 300
BGP table version is 8, main routing table version 8
4 network entries using 468 bytes of memory
8 path entries using 416 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
0 BGP Route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1256 total bytes of memory
BGP activity 4/0 prefixes, 8/0 paths, scan interval 60 secs

Neighbor      V   AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.2        4   300     11      10       8      0      0 00:05:33      2
3.3.3.3        4   300      9       8       8      0      0 00:03:46      2
3.3.3.4        4   300      8       7       8      0      0 00:02:03      2
R01#show ip bgp
BGP table version is 8, local router ID is 3.3.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop        Metric LocPrf Weight Path
* i186.10.10.0/24  3.3.3.2         0       100      0 i
*->                 0.0.0.0          0           32768 i
r i186.10.20.0/24  3.3.3.3         0       100      0 i
r>i                3.3.3.2         0       100      0 i
r>i186.10.30.0/24  3.3.3.4         0       100      0 i
r i                3.3.3.3         0       100      0 i
* i186.10.40.0/24  3.3.3.4         0       100      0 i
*->                 0.0.0.0          0           32768 i
R01#

```

The image shows that iBGP on **R01** in AS300 is active, with three neighbors exchanging **8 prefixes each**. The routing table shows routes like 186.10.10.0/24 and 186.10.20.0/24 are reachable via correct next hops, verifying proper iBGP and route propagation.

For **AS100**, iBGP was set up the same way as in other ASes, using loopback interfaces for neighbor connections. OSPF was also used as the internal routing protocol to make sure the loopback interfaces could communicate and that iBGP neighbors were reachable.

We used these commands to check `show ip bgp summary` and `show ip bgp`:

R7#show ip bgp summary
BGP router identifier 7.7.7.7, local AS number 100
BGP table version is 24, main routing table version 24
13 network entries using 1521 bytes of memory
19 path entries using 988 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2881 total bytes of memory
BGP activity 13/0 prefixes, 19/0 paths, scan interval 60 secs

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
5.5.5.5	4	100	17	17	24	0	0	00:11:33	2
6.6.6.6	4	100	17	17	24	0	0	00:11:33	4
8.8.8.8	4	100	14	14	24	0	0	00:08:47	3
9.9.9.9	4	100	7	7	24	0	0	00:02:56	4
10.10.10.10	4	100	6	6	24	0	0	00:01:54	2

R7#
Telnet:1 38, 4 38 Rows, 94 Cols Xterm 4:06 AM 1/4/2025

R7#show ip bgp
BGP table version is 24, local router ID is 7.7.7.7
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
r>15.5.5.5/32	5.5.5.5	0	100	0	i
r>16.6.6.6/32	6.6.6.6	0	100	0	i
r>17.7.7.7/32	7.7.7.7	0	100	0	i
r>18.8.8.8/32	8.8.8.8	0	100	0	i
r>19.9.9.9/32	9.9.9.9	0	100	0	i
r>10.10.10.10/32	10.10.10.10	0	100	0	i
r i192.168.5.0	5.0.0.0	0	100	0	i
r i192.168.6.0	6.0.0.0	0	100	0	i
* i192.168.7.0	0.0.0.0	0	100	0	i
r i192.168.8.0	8.8.8.8	0	100	0	i
r>i192.168.9.0	6.6.6.6	0	100	0	i
r i192.168.10.0	9.9.9.9	0	100	0	i
r>i192.168.11.0	10.10.10.10	0	100	0	i

R7#
Telnet:1 38, 4 38 Rows, 94 Cols Xterm 4:06 AM 1/4/2025

The images above show that iBGP peering on R7 in AS100 is successfully established, with neighbors exchanging prefixes as expected. The BGP routing table confirms that the advertised routes are correctly propagated and reachable via the appropriate next hops.

For **AS200** with static routing:

```
Router0(config)#ip route 10.1.1.1 255.255.255.255 10.86.1.1
Router0(config)#ip route 10.1.2.1 255.255.255.255 10.86.3.1
```

Router0#
Telnet:1 38, 4 38 Rows, 94 Cols Xterm 4:28 AM 1/6/2025

The routes direct traffic for the loopback IPs 10.1.1.1 and 10.1.2.1 through the next hops 10.86.1.1 and 10.86.3.1, respectively. These routes ensure that the loopback addresses of other routers in AS200 are reachable.

```
Router0(config)#router bgp 200
Router0(config-router)#bgp router-id 10.1.0.1
Router0(config-router)#neighbor 10.1.1.1 remote-as 200
Router0(config-router)#neighbor 10.1.1.1 update-source Loopback0
Router0(config-router)#neighbor 10.1.2.1 remote-as 200
Router0(config-router)#neighbor 10.1.2.1 update-source Loopback0
Router0(config-router)#network 10.86.1.0 mask 255.255.255.0
Router0(config-router)#network 10.1.0.1 mask 255.255.255.255
```

Router0#
Telnet:1 38, 4 38 Rows, 94 Cols Xterm 1:37 AM 1/7/2025

```
Router0#show ip bgp summary
BGP router identifier 10.1.0.1, local AS number 200
BGP table version is 11, main routing table version 11
6 network entries using 702 bytes of memory
6 path entries using 312 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1386 total bytes of memory
BGP activity 6/0 prefixes, 6/0 paths, scan interval 60 secs

Neighbor      V   AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.1.1.1      4   200     8     8    11    0    0 00:03:27    2
10.1.2.1      4   200     7     7    11    0    0 00:02:12    2

Router0#show ip bgp
BGP table version is 11, local router ID is 10.1.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.1.0.1/32	0.0.0.0	0	32768	0	i
r>i10.1.1.1/32	10.1.1.1	0	100	0	i
r>i10.1.2.1/32	10.1.2.1	0	100	0	i
*> 10.86.1.0/24	0.0.0.0	0	32768	0	i
r>i10.86.2.0/24	10.1.1.1	0	100	0	i
r>i10.86.3.0/24	10.1.2.1	0	100	0	i

Router0#
Telnet:1 38, 4 38 Rows, 94 Cols Xterm 1:47 AM 1/7/2025

The results confirm that iBGP peering is established on **Router0** with neighbors 10.1.1.1 and 10.1.2.1. The State/PfxRcd column shows that prefixes are successfully exchanged. The `show ip bgp` output verifies that advertised routes, such as 10.1.1.1/32 and 10.1.2.1/32, are active and reachable via the correct next hops, ensuring proper route propagation within AS200.

```
Router0#ping 10.1.1.1 source 10.1.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
Router0#ping 10.1.2.1 source 10.1.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/30/40 ms
Router0#
```

The results confirm full connectivity and proper configuration of static routes and iBGP sessions in AS200.

eBGP between ASes

we used these commands to apply the ebgp:

```
router bgp <LOCAL AS NUMBER>
```

```
neighbor <NEIGHBOR_IP> remote-as <NEIGHBOR_AS_NUMBER>
```

network <CONNECTED_NETWORK> mask <SUBNET_MASK>

and here how it done for each line between two ASes in the topology:

1. R4 (AS400) and R01 (AS300):

```
R4#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#router bgp 400
R4(config-router)#neighbor 11.86.10.10 remote-as 300
R4(config-router)#network 11.86.10.8 mask 255.255.255.252
R4(config-router)#exit
R4(config)#do wr
Building configuration...
[OK]
R4(config)#
*Mar 1 00:07:19.051: %BGP-5-ADJCHANGE: neighbor 11.86.10.10 Up
R4(config)#
R01#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R01(config)#router bgp 300
R01(config-router)#neighbor 11.86.10.9 remote-as 400
R01(config-router)#network 11.86.10.8 mask 255.255.255.252
R01(config-router)#
*Mar 1 00:07:18.607: %BGP-5-ADJCHANGE: neighbor 11.86.10.9 Up
R01(config-router)#exit
R01(config)#
R01(config)#do wr
Building configuration...
[OK]
```

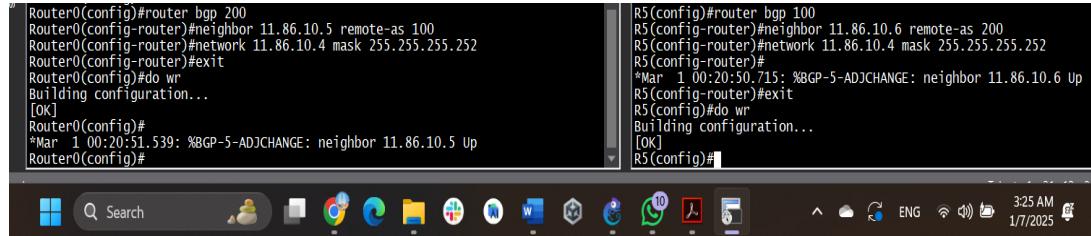
The image shows the successful eBGP configuration between R4 and Ro1. Both routers establish a neighbor relationship using their directly connected network (11.86.10.8/30) and advertise this network in their respective BGP configurations. The log messages confirm that the eBGP session is established successfully.

2. Router1 (AS200) and Ro1 (AS300):

```
Router1(config)#router bgp 200
Router1(config-router)#neighbor 11.86.10.2 remote-as 300
Router1(config-router)#network 11.86.10.0 mask 255.255.255.252
Router1(config-router)#exit
Router1(config)#do wr
Building configuration...
[OK]
Router1(config)#
*Mar 1 00:17:09.111: %BGP-5-ADJCHANGE: neighbor 11.86.10.2 Up
Router1(config)#
R01(config)#router bgp 300
R01(config-router)#neighbor 11.86.10.1 remote-as 200
R01(config-router)#network 11.86.10.0 mask 255.255.255.252
R01(config-router)#
*Mar 1 00:17:09.067: %BGP-5-ADJCHANGE: neighbor 11.86.10.1 Up
R01(config-router)#
R01(config-router)#
R01(config-router)#exit
R01(config)#do wr
Building configuration...
```

Both routers establish a neighbor relationship over the network 11.86.10.0/30 and advertise it in their respective BGP configurations.

3. Router0 (AS200) and R5 (AS100):

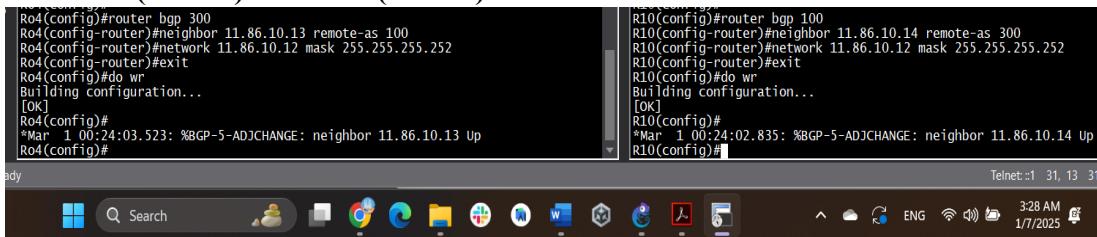


```
Router0(config)#router bgp 200
Router0(config-router)#neighbor 11.86.10.5 remote-as 100
Router0(config-router)#network 11.86.10.4 mask 255.255.255.252
Router0(config-router)#exit
Router0(config)#do wr
Building configuration...
[OK]
Router0(config)#
*Mar 1 00:20:51.539: %BGP-5-ADJCHANGE: neighbor 11.86.10.5 Up
Router0(config)#
R5(config)#router bgp 100
R5(config-router)#neighbor 11.86.10.6 remote-as 200
R5(config-router)#network 11.86.10.4 mask 255.255.255.252
R5(config-router)#
*Mar 1 00:20:50.715: %BGP-5-ADJCHANGE: neighbor 11.86.10.6 Up
R5(config-router)#exit
R5(config)#do wr
Building configuration...
[OK]
R5(config)#

```

The image shows eBGP set up over 11.86.10.4/30, with the session successfully established.

4. Ro4 (AS300) and R10 (AS100):



```
Ro4(config)#router bgp 300
Ro4(config-router)#neighbor 11.86.10.13 remote-as 100
Ro4(config-router)#network 11.86.10.12 mask 255.255.255.252
Ro4(config-router)#exit
Ro4(config)#do wr
Building configuration...
[OK]
Ro4(config)#
*Mar 1 00:24:03.523: %BGP-5-ADJCHANGE: neighbor 11.86.10.13 Up
Ro4(config)#
R10(config)#router bgp 100
R10(config-router)#neighbor 11.86.10.14 remote-as 300
R10(config-router)#network 11.86.10.12 mask 255.255.255.252
R10(config-router)#exit
R10(config)#do wr
Building configuration...
[OK]
R10(config)#
*Mar 1 00:24:02.835: %BGP-5-ADJCHANGE: neighbor 11.86.10.14 Up
R10(config)#

```

The eBGP session was established successfully.

Traffic Engineering

Route Optimization: Aggregate Prefixes in AS200

We will optimize routing in AS200 (IX Network) by implementing prefix aggregation. This reduces the number of advertised routes to neighboring autonomous systems, improving network scalability and efficiency. By aggregating multiple smaller prefixes into a single summarized route, the BGP table size is minimized, leading to enhanced performance and better resource utilization across the network.

Pre-Optimization State

We used:

```
show ip bgp neighbors <neighbor IP> advertised-routes
```

for each Router0 and Router1 in AS200

```

Router0#show ip bgp neighbors 11.86.10.5 advertised-routes
BGP table version is 43, local router ID is 10.1.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.1/32    0.0.0.0             0      32768 i
r>11.0.1.1/32    10.1.1.1             0      100   0 i
r>11.0.2.1/32    10.1.2.1             0      100   0 i
*> 10.86.1.0/24   0.0.0.0             0      32768 i
r>10.86.2.0/24   10.1.1.1             0      100   0 i
r>10.86.3.0/24   10.1.1.1             0      100   0 i
*> 10.86.10.0/30  10.1.1.1             0      100   0 i
*> 10.86.10.4/30  0.0.0.0             0      32768 i
r>11.0.1.1/32    11.86.10.2             0      100   0 300 i
r>11.0.10.0/24   11.86.10.2             0      100   0 300 i
*> 11.0.20.0/24   11.86.10.2             0      100   0 300 i
*> 11.0.30.0/24   11.86.10.2             0      100   0 300 i
*> 11.0.40.0/24   11.86.10.2             0      100   0 300 i
*> 11.0.86.40.0/24 11.86.10.2             0      100   0 300 400 i
*> 11.0.86.50.0/24 11.86.10.2             0      100   0 300 400 i
*> 11.0.86.60.0/24 11.86.10.2             0      100   0 300 400 i
*> 11.0.86.70.0/24 11.86.10.2             0      100   0 300 400 i
Network          Next Hop            Metric LocPrf Weight Path
Total number of prefixes 17
Router0#

```

The image shows **17 prefixes being advertised** to the neighboring router **11.86.10.5** in AS100. These include individual routes like 10.86.1.0/24, 10.86.2.0/24, and 10.86.3.0/24, as well as smaller routes like 10.1.1.0/32. Advertising each route separately makes the BGP table on the neighbor larger and requires more processing, which can cause inefficiencies.

```

Router1#show ip bgp neighbors 11.86.10.2 advertised-routes
BGP table version is 42, Local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.5/32    11.86.10.5             0      100   0 100 i
r>10.6.6.6/32    11.86.10.5             0      100   0 100 i
*> 17.7.7.7/32    11.86.10.5             0      100   0 100 i
*> 18.8.8.8/32    11.86.10.5             0      100   0 100 i
*> 19.9.9.9/32    11.86.10.5             0      100   0 100 i
r>10.1.0.1/32    10.1.0.1              0      100   0 i
*> 10.1.1.1/32    0.0.0.0              0      32768 i
r>10.1.2.1/32    10.1.2.1              0      100   0 i
*> 10.10.10.10/32 11.86.10.5             0      100   0 100 i
r>10.86.1.0/24   10.1.0.1              0      100   0 i
*> 10.86.2.0/24   0.0.0.0              0      32768 i
r>10.86.3.0/24   10.1.2.1              0      100   0 i
*> 10.86.10.0/30  0.0.0.0              0      32768 i
*> 11.0.10.10/4/30 10.1.0.1              0      100   0 i
*> 11.0.10.10/5/30 11.86.10.5             0      100   0 100 i
*> 11.0.10.10/6/30 11.86.10.5             0      100   0 100 i
*> 11.0.10.10/7/30 11.86.10.5             0      100   0 100 i
Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.10.10/8/30 11.86.10.5             0      100   0 i
*> 11.0.10.10/9/30 11.86.10.5             0      100   0 i
*> 11.0.10.10/10/30 11.86.10.5             0      100   0 i
*> 11.0.10.10/11/30 11.86.10.5             0      100   0 i
*> 11.0.10.10/12/30 11.86.10.5             0      100   0 i
Total number of prefixes 21
Router1#

```

Here, the image shows **21 prefixes being advertised** to the neighboring router **11.86.10.2** in AS300. These also include individual routes like 10.86.1.0/24, 10.86.2.0/24, and smaller routes like 10.1.1.0/32, along with additional prefixes such as 192.168.6.0/24 and 192.168.11.0/24. The separate advertisement of each route increases the size of the BGP table.

The routing tables before for each Ro1 AS300 and R5 AS100:

```

Gateway of last resort is not set
  86.0.0.0/24 is subnetted, 4 subnets
B     86.10.30.0 [200/0] via 11.86.10.14, 00:33:07
B     86.10.20.0 [200/0] via 11.86.10.14, 00:33:07
B     86.10.10.0 [200/0] via 11.86.10.14, 00:33:42
B     86.10.40.0 [200/0] via 11.86.10.14, 00:33:07
O     192.168.8.0/24 [110/11] via 192.168.5.2, 00:33:47, FastEthernet0/0
O     192.168.9.0/24 [110/21] via 192.168.5.2, 00:33:47, FastEthernet0/0
C     5.0.0.0/32 is subnetted, 1 subnets
O     192.168.10.0/24 [110/31] via 192.168.5.2, 00:33:49, FastEthernet0/0
D     6.0.0.0/32 is subnetted, 1 subnets
O     6.6.6.6 [110/11] via 192.168.5.2, 00:33:49, FastEthernet0/0
O     192.168.11.0/24 [110/30] via 192.168.5.2, 00:33:49, FastEthernet0/0
O     7.0.0.0/32 is subnetted, 1 subnets
O     7.7.7.7 [110/21] via 192.168.5.2, 00:33:50, FastEthernet0/0
O     8.0.0.0/32 is subnetted, 1 subnets
O     8.8.8.8 [0/12] via 192.168.5.2, 00:33:50, FastEthernet0/0
D     110.0.0.0/24 is subnetted, 4 subnets
B     110.86.60.0 [200/0] via 11.86.10.14, 00:32:45
B     110.86.50.0 [200/0] via 11.86.10.14, 00:32:45
B     110.86.40.0 [200/0] via 11.86.10.14, 00:32:45
B     110.86.70.0 [200/0] via 11.86.10.14, 00:32:45
B     9.0.0.0/32 is subnetted, 1 subnets
O     9.9.9.9 [110/22] via 192.168.5.2, 00:33:50, FastEthernet0/0
C     192.168.5.0/24 is directly connected, FastEthernet0/0
D     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O     10.10.10.10/32 [110/32] via 192.168.5.2, 00:33:50, FastEthernet0/0
B     10.1.2.1/32 [20/0] via 11.86.10.6, 00:33:12
B     10.1.1.1/32 [20/0] via 11.86.10.6, 00:33:12
B     10.86.1.0/24 [20/0] via 11.86.10.6, 00:33:12
B     10.86.2.0/24 [20/0] via 11.86.10.6, 00:33:14
B     10.86.3.0/24 [20/0] via 11.86.10.6, 00:33:14
O     192.168.6.0/24 [110/20] via 192.168.5.2, 00:33:52, FastEthernet0/0
  11.0.0.0/30 is subnetted, 4 subnets
C     11.86.10.4 is directly connected, Serial1/0
B     11.86.10.0 [20/0] via 11.86.10.6, 00:33:16
B     11.86.10.12 [200/0] via 10.10.10.10, 00:33:15
B     11.86.10.8 [200/0] via 11.86.10.14, 00:32:49
O     192.168.7.0/24 [110/21] via 192.168.5.2, 00:33:55, FastEthernet0/0
R5#

```

```

Gateway of last resort is not set
  3.0.0.0/32 is subnetted, 4 subnets
O     3.3.3.2 [110/12] via 11.86.10.13, 00:33:44, FastEthernet0/0
O     3.3.3.2 [110/21] via 86.10.40.1, 00:33:44, FastEthernet2/0
O     3.3.3.4 is directly connected, Loopback0
  86.0.0.0/24 is subnetted, 4 subnets
C     86.10.30.0 is directly connected, FastEthernet0/0
O     86.10.20.0 [110/20] via 86.10.30.1, 00:33:44, FastEthernet0/0
O     86.10.10.0 [110/11] via 86.10.40.1, 00:33:46, FastEthernet2/0
C     86.10.40.0 is directly connected, FastEthernet2/0
O     192.168.8.0/24 [110/13] via 11.86.10.13, 00:33:06
B     192.168.9.0/24 [20/0] via 11.86.10.13, 00:33:06
B     5.0.0.0/32 is subnetted, 1 subnets
B     5.5.5.5 [20/0] via 11.86.10.13, 00:33:11
B     192.168.10.0/24 [20/0] via 11.86.10.13, 00:33:11
B     6.0.0.0/32 is subnetted, 1 subnets
B     6.6.6.6 [20/0] via 11.86.10.13, 00:32:40
B     192.168.11.0/24 [20/0] via 11.86.10.13, 00:32:40
B     7.0.0.0/32 is subnetted, 1 subnets
B     7.7.7.7 [20/0] via 11.86.10.13, 00:32:40
B     8.0.0.0/32 is subnetted, 1 subnets
B     8.8.8.8 [20/0] via 11.86.10.13, 00:33:11
B     110.0.0.0/24 is subnetted, 4 subnets
B     110.86.60.0 [200/0] via 3.3.3.1, 00:33:14
B     110.86.50.0 [200/0] via 3.3.3.1, 00:33:14
B     110.86.40.0 [200/0] via 3.3.3.1, 00:33:14
B     110.86.70.0 [200/0] via 3.3.3.1, 00:33:14
B     9.0.0.0/32 is subnetted, 1 subnets
B     9.9.9.9 [120/0] via 11.86.10.13, 00:32:40
B     192.168.5.0/24 [20/0] via 11.86.10.13, 00:33:11
B     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
B     10.10.10.10/32 [20/0] via 11.86.10.13, 00:33:11
B     10.1.2.1/32 [200/0] via 3.3.3.1, 00:33:14
B     10.1.0.1/32 [200/0] via 3.3.3.1, 00:33:15
B     10.86.1.0/24 [200/0] via 3.3.3.1, 00:33:15
B     10.86.2.0/24 [200/0] via 3.3.3.1, 00:33:16
B     10.86.3.0/24 [200/0] via 3.3.3.1, 00:33:16
B     192.168.6.0/24 [120/0] via 11.86.10.13, 00:32:43
B     11.0.0.0/30 is subnetted, 4 subnets
B     11.86.10.4 [20/0] via 11.86.10.13, 00:33:14
B     11.86.10.0 [200/0] via 3.3.3.1, 00:33:16
C     11.86.10.12 is directly connected, Serial1/0
B     11.86.10.8 [200/0] via 3.3.3.1, 00:33:16
B     192.168.7.0/24 [20/0] via 11.86.10.13, 00:32:43
Ro4#■

```

The routing tables for Ro1 (AS300) and R5 (AS100) before optimization show a significant number of individual routes being advertised.

Implementation of Prefix Aggregation:

To determine the /22 summary prefix, we analyzed the individual subnets (10.86.1.0/24, 10.86.2.0/24, and 10.86.3.0/24) by converting them into binary. The common bits shared by all three subnets were identified, and it was observed that the first 22 bits were the same. Based on this, the subnet mask was determined to be /22, allowing us to summarize the subnets into 10.86.0.0/22.

The following command was used to configure the aggregated prefix on the both routers:

```
aggregate-address 10.86.0.0 255.255.252.0 summary-only
```

Router0(config)#router bgp 200
Router0(config-router)#aggregate-address 10.86.0.0 255.255.252.0 summary-only
Router0(config-router)#

Router1(config)#router bgp 200
Router1(config-router)#aggregate-address 10.86.0.0 255.255.252.0 summary-only
Router1(config-router)#

This command combines the individual subnets into a single summarized route, reducing the number of advertised prefixes to neighboring routers and optimizing the routing table.

Post-Optimization State

```
Router1#show ip bgp neighbors 11.86.10.2 advertised-routes
BGP table version is 47, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path
*>10.86.0.0/22 0.0.0.0 32768 i
*>11.86.10.0/30 0.0.0.0 32768 i
*>11.86.10.4/30 10.1.0.1 0 100 0 i
*>1192.168.5.0 11.86.10.5 0 100 0 100 i
*>1192.168.6.0 11.86.10.5 0 100 0 100 i
*>1192.168.7.0 11.86.10.5 0 100 0 100 i
*>1192.168.8.0 11.86.10.5 0 100 0 100 i
*>1192.168.9.0 11.86.10.5 0 100 0 100 i
Total number of prefixes 19
Router1#
```

```
Router0#show ip bgp neighbors 11.86.10.5 advertised-routes
*Mar 1 01:06:40.059: %SYS-5-CONFIG-I: Configured From console by console
Router0#show ip bgp neighbors 11.86.10.5 advertised-routes
BGP table version is 48, local router ID is 10.1.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path
*> 10.1.0.1/32 0.0.0.0 32768 i
r>110.1.1.1/32 10.1.1.1 0 100 0 i
r>110.1.2.1/32 10.1.2.1 0 100 0 i
*> 10.86.0.0/22 0.0.0.0 32768 i
*>111.86.10.0/30 10.1.1.1 0 100 0 i
*> 11.86.10.4/30 0.0.0.0 32768 i
*>111.86.10.8/30 11.86.10.2 0 100 0 300 i
*>186.10.10.0/24 11.86.10.2 0 100 0 300 i
*>186.10.20.0/24 11.86.10.2 0 100 0 300 i
*>186.10.30.0/24 11.86.10.2 0 100 0 300 i
*>186.10.40.0/24 11.86.10.2 0 100 0 300 i
*>110.86.40.0/24 11.86.10.2 0 100 0 300 400 i
*>110.86.50.0/24 11.86.10.2 0 100 0 300 400 i
*>110.86.60.0/24 11.86.10.2 0 100 0 300 400 i
*>110.86.70.0/24 11.86.10.2 0 100 0 300 400 i
Total number of prefixes 15
Router0#
```

As shown in the images, the individual /24 prefixes have been successfully aggregated into a single /22 prefix (10.86.0.0/22). This optimization reduced the total number of advertised prefixes, improving efficiency and simplifying the BGP table.

Router	Before Optimization (Prefixes)	After Optimization (Prefixes)	Reduction
Router0	17	15	2
Router1	21	19	2

The table shows prefix reduction after aggregation.

This process is particularly significant in real-world networking as it helps minimize the load on hardware, especially in large-scale networks. By consolidating routes, it improves scalability and efficiency, which are critical for enterprise and ISP networks.

Redundancy Testing: Simulating Link Failures

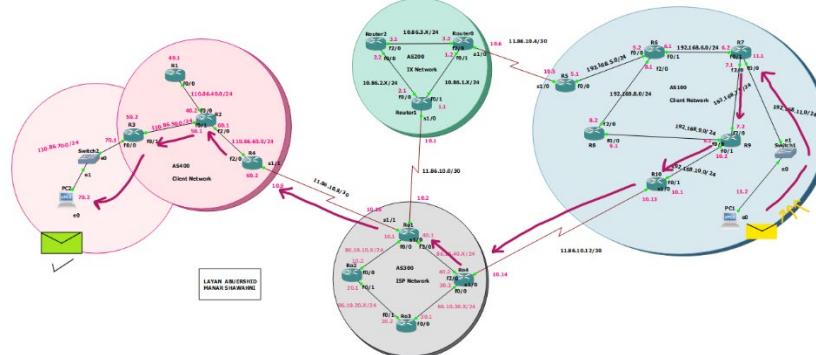
The purpose of this section is to test the network's redundancy by simulating a link failure between AS100 and AS300. The goal is to verify that traffic is automatically rerouted through alternative paths, ensuring uninterrupted connectivity and demonstrating the reliability of the network's design.

Pre-Failure State

The traceroute from **PC1 (AS100)** to **PC2 (AS400)**, as shown in the image, reveals the path taken by the traffic:

```
PC1> trace 110.86.70.2
trace to 110.86.70.2, 8 hops max, press Ctrl+C to stop
 1  192.168.11.1    15.489 ms   16.426 ms   15.820 ms
 2  192.168.7.2    46.778 ms   48.069 ms   46.281 ms
 3  192.168.10.1   76.056 ms   77.905 ms   76.542 ms
 4  11.86.10.14   107.660 ms   108.200 ms   111.116 ms
 5  86.10.40.1    138.231 ms   138.653 ms   141.093 ms
 6  11.86.10.9    171.415 ms   168.236 ms   168.226 ms
 7  110.86.60.1   197.819 ms   198.263 ms   197.838 ms
 8  110.86.50.2   227.292 ms   229.589 ms   227.618 ms
```

PC1 (R7, AS100) → R9 → R10 → Ro4 (AS300) → Ro1 → R4 (AS400) → R2 → PC2



This path flows directly through the link between Ro4 (AS300) and R10 (AS100).

Simulating the Link Failure

Since these are non-stub networks with multiple connections to other ASes, and the traceroute shows that the path from **PC2 (AS400)** to **PC1 (AS100)** flows through the link between **Ro4 (AS300)** and **R10 (AS100)**, we decided to disable this link to test the network's redundancy and observe traffic rerouting through alternative paths.

We shut down the Ro4 serial interface as shown in the figure below:

```
R04(config)#interface serial1/0
R04(config-if)#shutdown
R04(config-if)#
*Mar 1 00:05:18.851: %BGP-5-ADJCHANGE: neighbor 11.86.10.13 Down Interface flap
R04(config-if)#
*Mar 1 00:05:20.827: %LINK-5-CHANGED: Interface Serial1/0, changed state to administratively down
*Mar 1 00:05:21.827: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down
R04(config-if)#

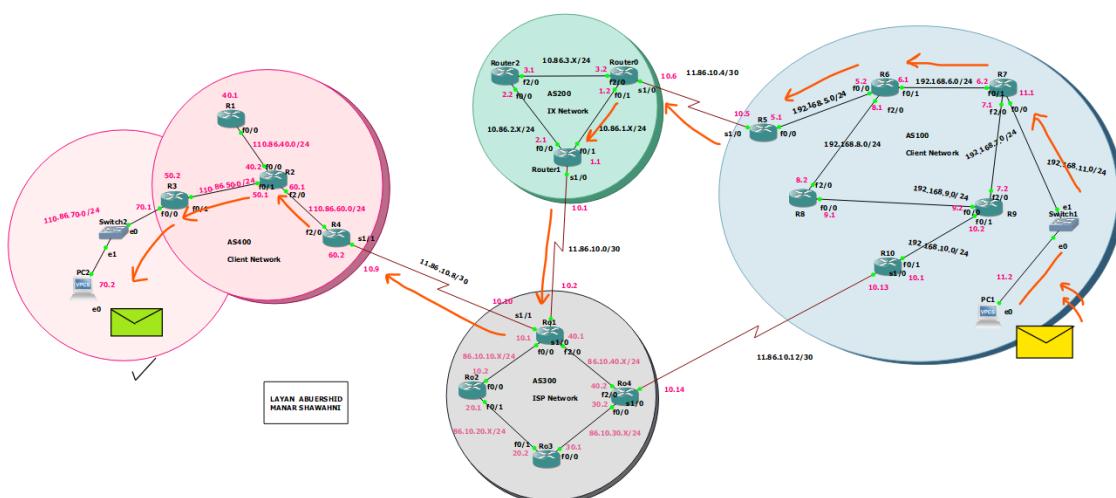
```

Post-Failure State

Here, we tried to trace PC1 (AS100) to PC2 (AS400) again after the link failure:

```
PC1> trace 110.86.70.2
trace to 110.86.70.2, 8 hops max, press Ctrl+C to stop
 1  192.168.11.1  15.717 ms  15.853 ms  15.972 ms
 2  192.168.6.1   45.703 ms  45.896 ms  46.826 ms
 3  192.168.5.1   76.816 ms  77.727 ms  75.599 ms
 4  11.86.10.6    106.529 ms 109.111 ms  107.464 ms
 5  10.86.1.1     136.278 ms 137.794 ms  137.337 ms
 6  11.86.10.2    167.502 ms 167.400 ms  167.909 ms
 7  11.86.10.9    200.122 ms 199.714 ms  183.193 ms
 8  110.86.60.1   228.648 ms 226.875 ms  227.936 ms
```

the new path from **PC1 (AS100)** to **PC2 (AS400)**, after the link failure was observed as:



PC1 (R7, AS100) → R6 → R5 → Router0 (AS200) → Router1 → Ro1 (AS300) → R4 (AS400) → R2 → PC2

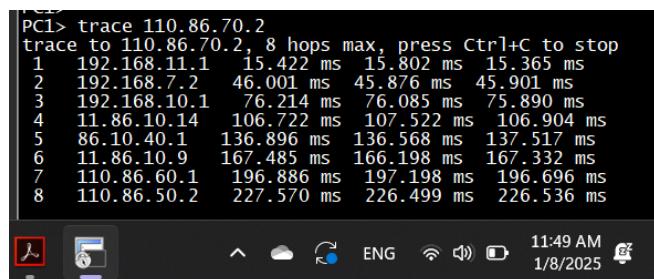
The path shifted from the direct connection between AS100 and AS300 to an alternative route via AS200, demonstrating the effectiveness of the network's redundancy. This ensures uninterrupted connectivity despite the link failure, highlighting the reliability of BGP's rerouting mechanisms in a non-stub network.

Restoring the Original Link

To restore the original connection, we re-enabled the disabled interface using the following commands:

```
no shutdown
```

After enabling the interface, the link between **Ro4 (AS300)** and **R10 (AS100)** became active again. A traceroute test was performed from **PC2 (AS400)** to **PC1 (AS100)**, and the results confirmed that the traffic returned to its original path:



```
PC1> trace 110.86.70.2
trace to 110.86.70.2, 8 hops max, press Ctrl+C to stop
 1  192.168.11.1  15.422 ms  15.802 ms  15.365 ms
 2  192.168.7.2  46.001 ms  45.876 ms  45.901 ms
 3  192.168.10.1  76.214 ms  76.085 ms  75.890 ms
 4  11.86.10.14  106.722 ms  107.522 ms  106.904 ms
 5  86.10.40.1   136.896 ms  136.568 ms  137.517 ms
 6  11.86.10.9   167.485 ms  166.198 ms  167.332 ms
 7  110.86.60.1  196.886 ms  197.198 ms  196.696 ms
 8  110.86.50.2  227.570 ms  226.499 ms  226.536 ms
```

PC1 (R7, AS100) → R9 → R10 → Ro4 (AS300) → Ro1 → R4 (AS400) → R2 → PC2

This shows that the network can quickly recover and return to the best path once the failed link is fixed. It also shows how BGP is flexible and reliable in handling changes in the network.

This process is particularly significant in real-world networking as it helps minimize the load on hardware, especially in large-scale networks. By consolidating routes, it improves scalability and efficiency, which are critical for enterprise and ISP networks.

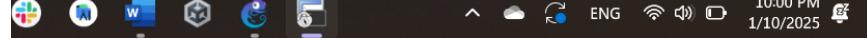
Test Scenarios

OSPF Neighbor Verification

On Router Ro4 at AS300

The output from the `show ip ospf neighbor` command on Router R04 shows that it has successfully established OSPF adjacencies with two neighboring routers, identified by their Router IDs (3.3.3.1 and 3.3.3.3). Both neighbors are in the FULL/BDR state, indicating fully established adjacencies and proper routing information exchange.

```
R04#show ip ospf neighbor
Neighbor ID      Pri  State            Dead Time    Address          Interface
3.3.3.1           1    FULL/BDR        00:00:39     86.10.40.1      FastEthernet2/0
3.3.3.3           1    FULL/BDR        00:00:30     86.10.30.1      FastEthernet0/0
R04#
```



On Router R9 at AS100

The output on Router R9 reveals three fully established OSPF adjacencies with neighboring routers. The neighbors, identified by their Router IDs, are in the FULL/DR and FULL/BDR states, indicating proper OSPF functionality.

```
R9#show ip ospf neighbor
Neighbor ID      Pri  State            Dead Time    Address          Interface
10.10.10.10      1    FULL/DR         00:00:31     192.168.10.1    FastEthernet0/1
8.8.8.8           1    FULL/BDR        00:00:31     192.168.9.1     FastEthernet0/0
7.7.7.7           1    FULL/BDR        00:00:30     192.168.7.1     FastEthernet2/0
R9#
```



RIP Route Propagation Verification on AS400

By using `show ip route rip` command:

```
R1#show ip route rip
 4.0.0.0/32 is subnetted, 4 subnets
R    4.4.4.4 [120/2] via 110.86.40.2, 00:00:25, FastEthernet0/0
R    4.4.4.2 [120/1] via 110.86.40.2, 00:00:25, FastEthernet0/0
R    4.4.4.3 [120/2] via 110.86.40.2, 00:00:25, FastEthernet0/0
  110.0.0.0/24 is subnetted, 4 subnets
R    110.86.60.0 [120/1] via 110.86.40.2, 00:00:25, FastEthernet0/0
R    110.86.50.0 [120/1] via 110.86.40.2, 00:00:25, FastEthernet0/0
R    110.86.70.0 [120/2] via 110.86.40.2, 00:00:25, FastEthernet0/0
R1#
```



```
R2#show ip route rip
 4.0.0.0/32 is subnetted, 4 subnets
R    4.4.4.4 [120/1] via 110.86.60.2, 00:00:00, FastEthernet2/0
R    4.4.4.1 [120/1] via 110.86.40.1, 00:00:03, FastEthernet0/0
R    4.4.4.3 [120/1] via 110.86.50.2, 00:00:20, FastEthernet0/1
  110.0.0.0/24 is subnetted, 4 subnets
R    110.86.70.0 [120/1] via 110.86.50.2, 00:00:20, FastEthernet0/1
R2#
```



```
R3#show ip route rip
      4.0.0.0/32 is subnetted, 4 subnets
R        4.4.4.4 [120/2] via 110.86.50.1, 00:00:17, FastEthernet0/1
R        4.4.4.1 [120/2] via 110.86.50.1, 00:00:17, FastEthernet0/1
R        4.4.4.2 [120/1] via 110.86.50.1, 00:00:17, FastEthernet0/1
      110.0.0.0/24 is subnetted, 4 subnets
R        110.86.60.0 [120/1] via 110.86.50.1, 00:00:17, FastEthernet0/1
R        110.86.40.0 [120/1] via 110.86.50.1, 00:00:17, FastEthernet0/1
```

D3#

10:10 PM 1/10/2025

```
R4#show ip route rip
      4.0.0.0/32 is subnetted, 4 subnets
R        4.4.4.1 [120/2] via 110.86.60.1, 00:00:13, FastEthernet2/0
R        4.4.4.2 [120/1] via 110.86.60.1, 00:00:13, FastEthernet2/0
R        4.4.4.3 [120/2] via 110.86.60.1, 00:00:13, FastEthernet2/0
      110.0.0.0/24 is subnetted, 4 subnets
R        110.86.50.0 [120/1] via 110.86.60.1, 00:00:13, FastEthernet2/0
R        110.86.40.0 [120/1] via 110.86.60.1, 00:00:13, FastEthernet2/0
R        110.86.70.0 [120/2] via 110.86.60.1, 00:00:13, FastEthernet2/0
```

R4#

10:09 PM 1/10/2025

By using the command on routers R4, R1, R2, and R3, we can observe that RIP is successfully propagating routes within AS400. Each router has learned routes to the loopback interfaces (4.4.4.x) and connected subnets (110.86.x.x) via RIP. The routes are reachable through the specified next-hop addresses and interfaces, with metrics indicating the number of hops to each destination. This confirms that RIP is functioning correctly, ensuring proper routing and connectivity within the AS.

BGP Route Advertisement Verification

show ip bgp neighbors <neighbor IP> advertised-routes

for Ro4 at AS300:

```
Ro4#show ip bgp neighbors 11.86.10.13 advertised-routes
BGP table version is 34, local router ID is 3.3.3.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
>>11.0.1.0/32    0.0.0.0           0       0 200 i
>>10.1.1.1/32    3.3.3.1           0       0 200 i
>>10.1.2.1/32    3.3.3.1           0       0 200 i
>>10.86.0.0/22   3.3.3.1           0       0 200 i
>>11.86.10.0/30  3.3.3.1           0       0 i
>>11.86.10.8/30  3.3.3.1           0       0 200 i
>>11.86.10.17/20 0.0.0.0           0       32768 i
>>10.86.10.17/20 0.0.0.0           0       100 0 i
>>10.86.10.20/24 3.3.3.1           0       0 i
>> 86.10.30.0/24 0.0.0.0           0       100 32768 i
>> 86.10.40.0/24 0.0.0.0           0       100 32768 i
>>110.86.40.0/24 3.3.3.1           0       100 0 400 i
>>110.86.50.0/24 3.3.3.1           0       100 0 400 i
>>110.86.60.0/24 3.3.3.1           0       100 0 400 i
>>110.86.70.0/24 3.3.3.1           0       100 0 400 i

Total number of prefixes 15
Ro4#
```

10:52 PM 1/10/2025

The output on Router R04 confirms that it is correctly advertising **15 prefixes** to its BGP neighbor at **11.86.10.13**. The advertised routes include locally originated networks (e.g., **86.10.30.0/24** and **86.10.40.0/24**), summarized prefixes (**110.86.0.0/22**), and routes learned from other ASes (AS200 and AS400). The next-hop addresses, AS paths, and local preference values indicate proper route propagation and path selection.

For Router0 at AS200:

```
Router0#show ip bgp neighbors 11.86.10.5 advertised-routes
BGP table version is 46, local router ID is 10.1.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - int
r RIB-failure, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
>>10.1.0.1/32    0.0.0.0           0       32768 i
r>i10.1.1.1/32    10.1.1.1          0       100 0 i
r>i10.1.2.1/32    10.1.2.1          0       100 0 i
>> 10.86.0.0/22   0.0.0.0           0       32768 i
>>11.86.10.0/30  10.1.1.1          0       100 0 i
>>11.86.10.4/30  0.0.0.0           0       32768 i
>>11.86.10.8/30  11.86.10.2         0       100 0 300 i
>>11.86.10.10/24 11.86.10.2         0       100 0 300 i
>>11.86.10.20/24 11.86.10.2         0       100 0 300 i
>>11.86.10.30/24 11.86.10.2         0       100 0 300 i
>>11.86.10.40/24 11.86.10.2         0       100 0 300 i
>>110.86.40.0/24 11.86.10.2         0       100 0 300 400 i
>>110.86.50.0/24 11.86.10.2         0       100 0 300 400 i
>>110.86.60.0/24 11.86.10.2         0       100 0 300 400 i
>>110.86.70.0/24 11.86.10.2         0       100 0 300 400 i
```

10:59 PM 1/10/2025

The output on Router0 shows it advertising routes, including locally originated prefixes (e.g., **10.86.0.0/22**) and routes from AS300 and AS400, to its neighbor at **11.86.10.5**. The next-hop addresses, AS paths, and local preference values confirm proper BGP functionality, ensuring efficient inter-AS routing and connectivity.

Loopback Interface Reachability

R1#ping 4.4.4.2
[connection to 4.4.4.2 Closed by foreign host]
R1#ping 6.6.6.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/36 ms
R5#

Telnet:1 36 4 36 Rows 78 Cols Xterm 1148 PM 1/10/2025

The ping command for 4.4.4.2 output on Router R1 shows a **100% success rate**, confirming successful connectivity to the destination IP **4.4.4.2**. The round-trip times (min/avg/max = 28/30/36 ms) indicate low latency and stable communication. And the same for pinging router R6 loopback 6.6.6.6 from router R5, this confirms that the network path between the routers and the destination is functioning correctly.

Static Route Verification in AS200

show ip route static

Router2#show ip route static
10.0.0.0/8 is variably subnetted, 8 subnets, 3 mask
S 10.1.0.1/32 [1/0] via 10.86.3.2
S 10.1.1.0/32 [1/0] via 10.86.2.1
S 10.86.1.0/24 [1/0] via 10.86.2.1
Router2#

Router0#show ip route static
10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
S 10.1.2.1/32 [1/0] via 10.86.2.2
S 10.1.1.1/32 [1/0] via 10.86.1.1
S 10.86.0.0/24 [1/0] via 10.86.1.1
Router0#

Router1#show ip route static
10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
S 10.1.2.1/32 [1/0] via 10.86.2.2
S 10.1.0.1/32 [1/0] via 10.86.1.2
S 10.86.3.0/24 [1/0] via 10.86.2.2
Router1#

Telnet:1 36 9 36 Rows, 56 Cols Xterm CAP NUM 1152 PM 1/10/2025

The output confirms static routes in AS200 are correctly configured for all routers.

PC1 ⇔ PC2 tests

PC1> ping 110.86.70.2
110.86.70.2 icmp_seq=1 ttl=56 time=231.788 ms
84 bytes from 110.86.70.2 icmp_seq=2 ttl=56 time=244.861 ms
84 bytes from 110.86.70.2 icmp_seq=3 ttl=56 time=247.524 ms
84 bytes from 110.86.70.2 icmp_seq=4 ttl=56 time=247.524 ms
84 bytes from 110.86.70.2 icmp_seq=5 ttl=56 time=245.792 ms
PC1> trace 110.86.70.2
trace to 110.86.70.2, 8 hops max, press Ctrl+C to stop
1 192.168.11.1 15.949 ms 17.151 ms 16.681 ms
2 192.168.7.2 47.375 ms 46.292 ms 46.274 ms
3 192.168.10.1 77.497 ms 79.111 ms 77.583 ms
4 11.86.10.14 108.110 ms 107.331 ms 108.589 ms
5 86.10.40.1 138.422 ms 139.833 ms 138.057 ms
6 11.86.10.9 169.728 ms 170.907 ms 169.530 ms
7 110.86.60.1 201.746 ms 200.779 ms 198.679 ms
8 110.86.50.2 232.842 ms 231.084 ms 232.801 ms
PC1>

PC2> ping 192.168.11.2
84 bytes from 192.168.11.2 icmp_seq=1 ttl=56 time=245.137 ms
84 bytes from 192.168.11.2 icmp_seq=2 ttl=56 time=246.990 ms
84 bytes from 192.168.11.2 icmp_seq=3 ttl=56 time=247.217 ms
84 bytes from 192.168.11.2 icmp_seq=4 ttl=56 time=246.669 ms
84 bytes from 192.168.11.2 icmp_seq=5 ttl=56 time=244.592 ms
PC2> trace 192.168.11.2
trace to 192.168.11.2, 8 hops max, press Ctrl+C to stop
1 110.86.70.1 15.808 ms 16.534 ms 16.552 ms
2 110.86.50.1 47.730 ms 47.333 ms 47.475 ms
3 110.86.60.2 77.296 ms 76.795 ms 76.814 ms
4 11.86.10.10 109.967 ms 107.547 ms 107.950 ms
5 86.10.40.2 140.202 ms 139.381 ms 140.602 ms
6 11.86.10.13 169.116 ms 169.746 ms 170.349 ms
7 192.168.10.2 200.451 ms 201.902 ms 202.629 ms
8 192.168.7.1 233.146 ms 232.883 ms 232.284 ms
PC2>

Telnet:1 36 9 36 Rows, 56 Cols Xterm CAP NUM 1155 PM 1/10/2025

The output shows successful **ping** and **traceroute** tests between **PC1** and **PC2**, confirming end-to-end connectivity across the network.

Conclusion

This project successfully designed and implemented a multi-AS BGP environment, demonstrating key features such as route filtering, traffic engineering, and efficient routing. By configuring OSPF, RIP, static routing, and BGP across four autonomous systems (AS100, AS200, AS300, AS400), we ensured proper intra-AS and inter-AS communication. Traffic engineering techniques, such as prefix aggregation, optimized routing tables and improved scalability. End-to-end connectivity tests confirmed the network's functionality, with all configurations verified through rigorous testing. This project highlights the importance of proper routing protocols and traffic management in building scalable and efficient networks.