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

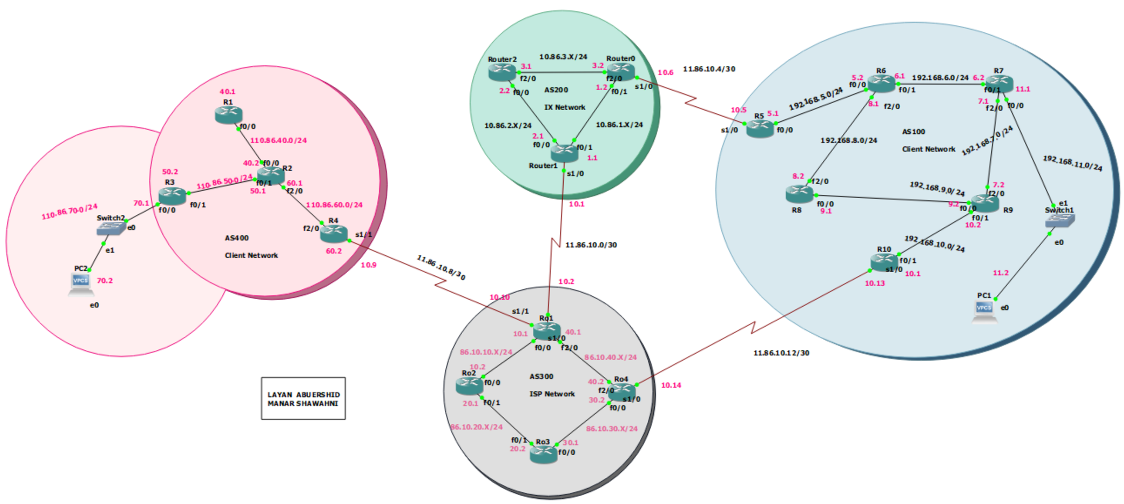
Table of Contents

Introduction

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>

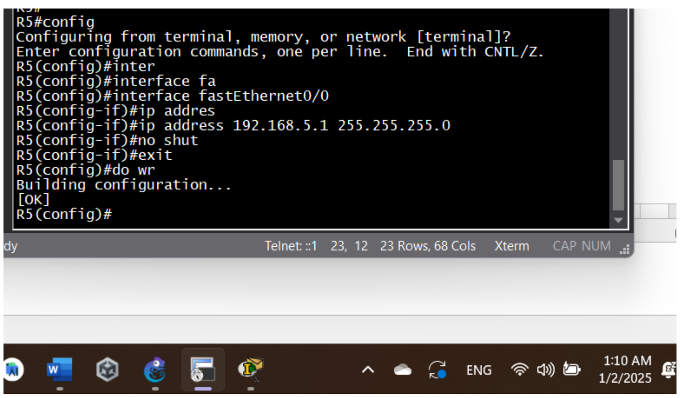
ip address <ip_address> <subnet_mask>

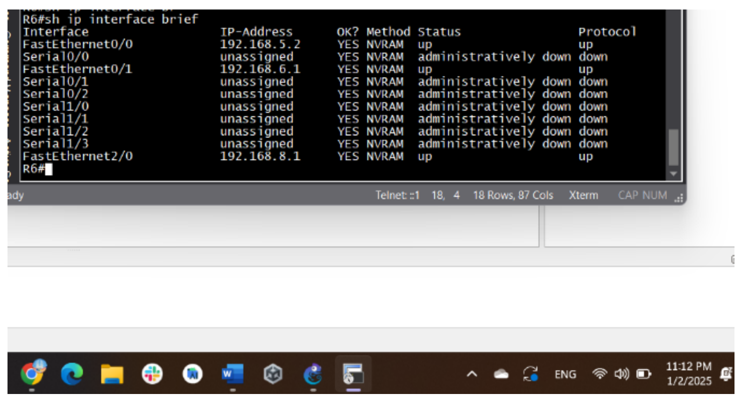
no shutdown

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:





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:

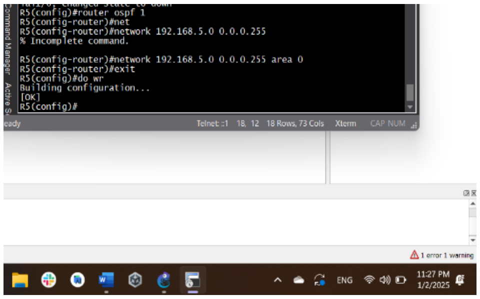
router ospf <process_id>

router ospf 1

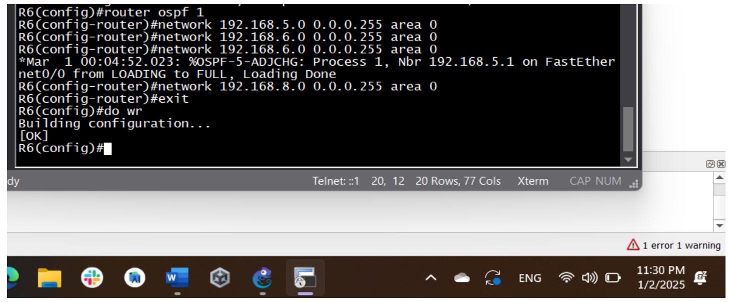
Next, we configure OSPF to advertise networks by specifying the network and its area. Area 0 known as the backbone, is the main area used to organize networks.

network <network_address> <wildcard_mask> area <area_number>

for R5, it was one network:

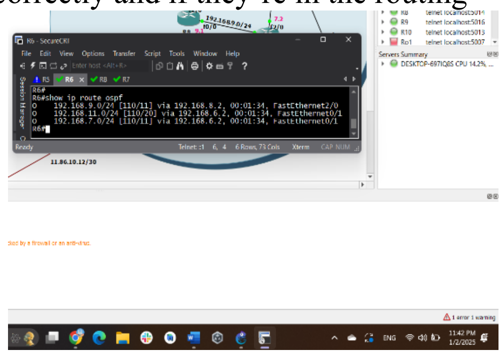


For R6, we done for the three directly connected:



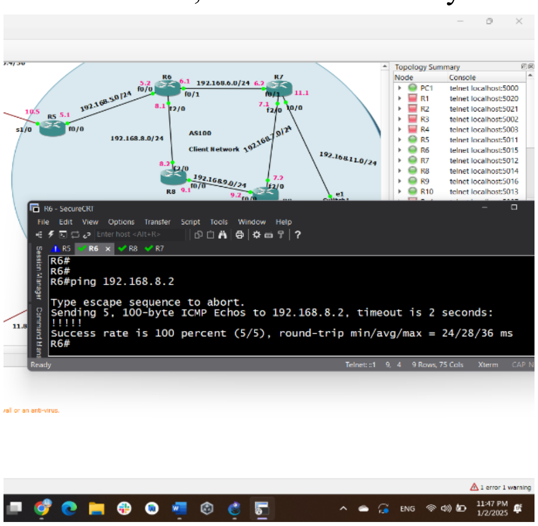
Then, we check if OSPF is advertising the routes correctly and if they're in the routing table: show ip route ospf

which for the other OSPF-learned routes

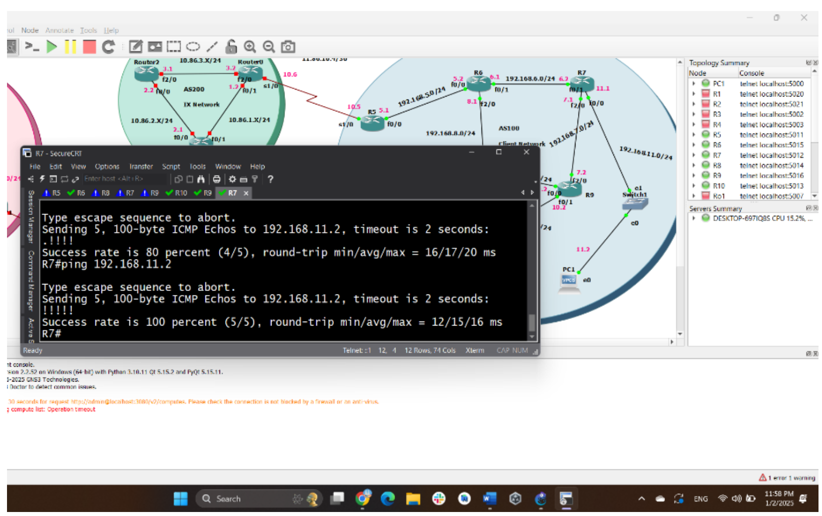


We tried to ping 192.168.8.2 from R6,

as shown, it done successfully.

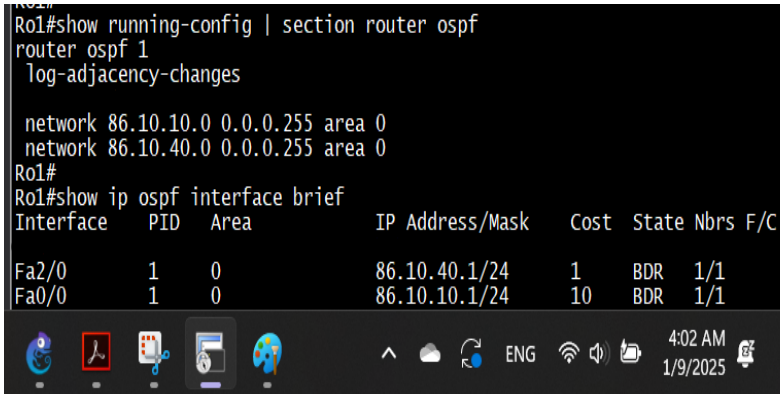


I pinged the pc, and done correctly



Enable OSPF in AS300

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



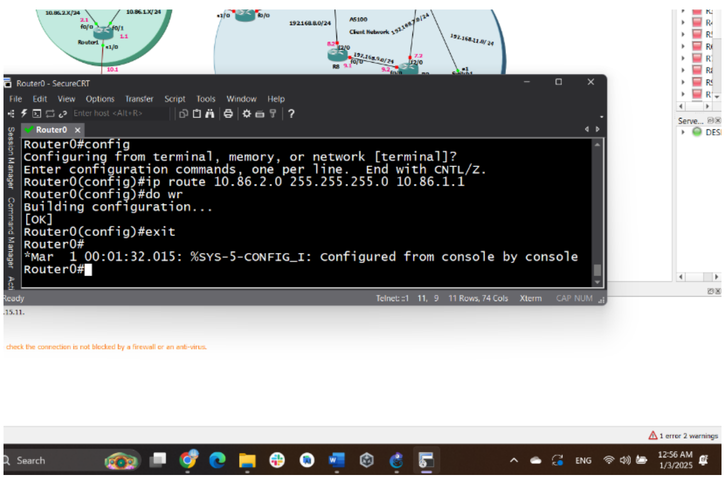
The image shows the OSPF configuration on Ro1, where the networks 86.10.10.0/24 and 86.10.40.0/24 are advertised in area 0. The OSPF interface details confirm that both FastEthernet interfaces (Fa2/0 and Fa0/0) with one neighbor each.

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>

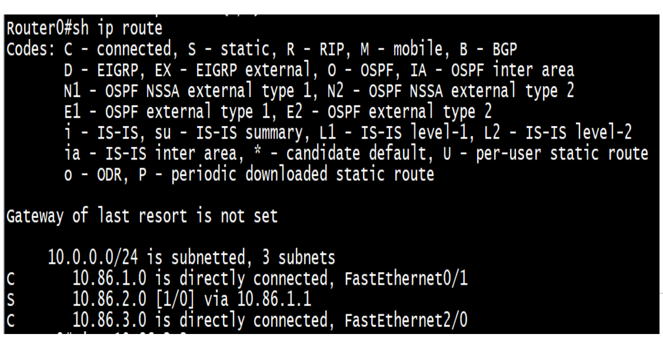


For example, for **Router0** has two directly connected networks (10.86.1.0/24 and 10.86.3.0/24), and we need to reach **10.86.2.0/24** via another router (Router1 at 10.86.1.1).

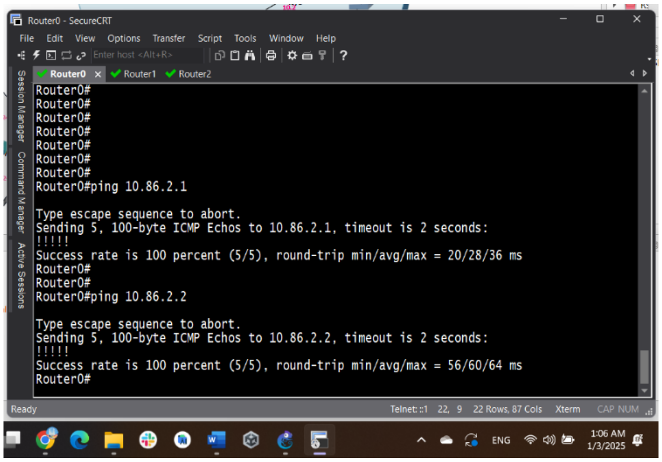
And we repeated this for each router in this AS.

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

`show ip route`



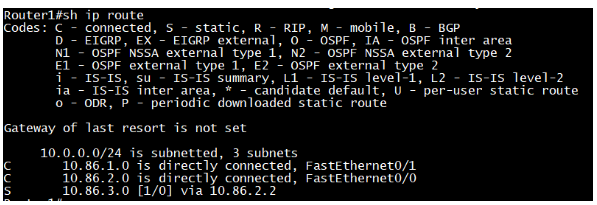
The image shows the routing table of Router0. It indicates that static routes are configured correctly.

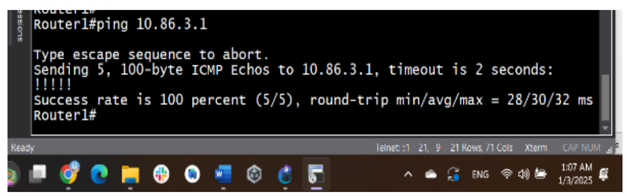


The image shows successful ping results from Router0 to the IPs 10.86.2.1 and 10.86.2.2. This indicates that Router0 has reachability to devices in the 10.86.2.0/24 subnet.

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.

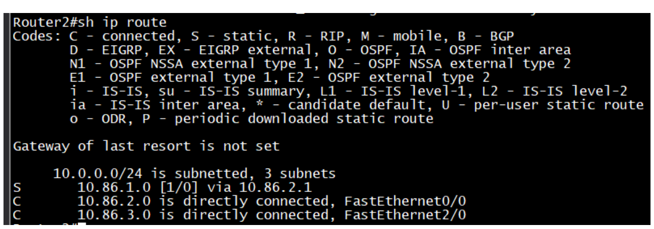
A static route was configured on **Router1** to direct traffic for the **10.86.3.0/24** network via the next hop **10.86.2.2**. The route was added using the ip route command, and the configuration was saved.





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.



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.

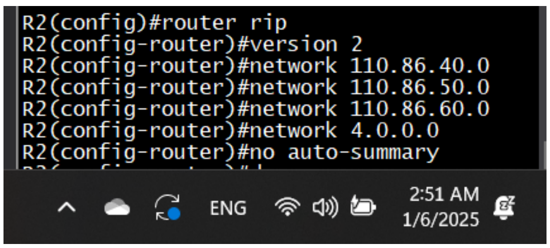
We entered RIP configuration mode using this command:

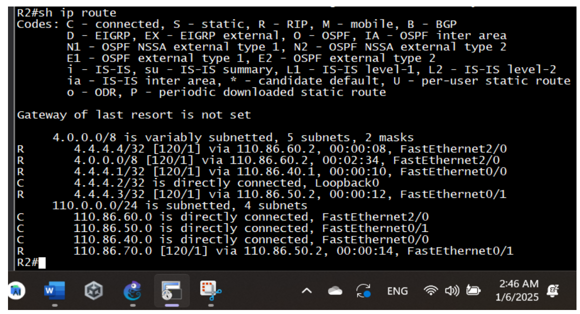
router rip

We added the networks connected to router:

network <network_address>

for R2,



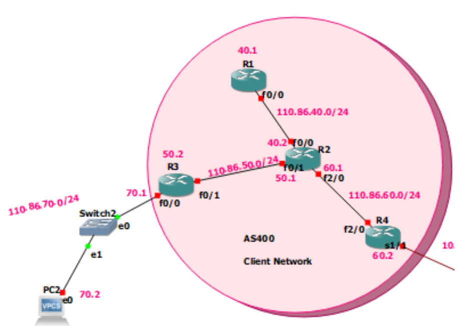


and we repeated the same steps for R1, R3, and R4, adding their connected networks to the configuration.

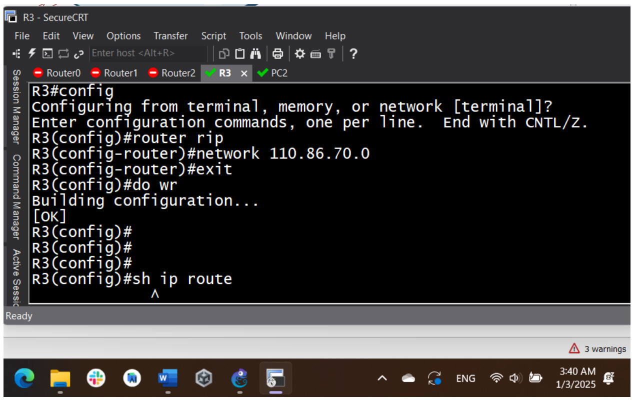
Adding a New Client to AS400

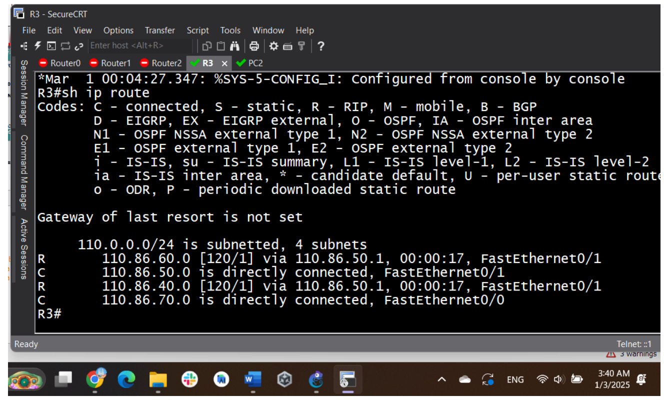
To add a new client (PC) to AS400:

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.

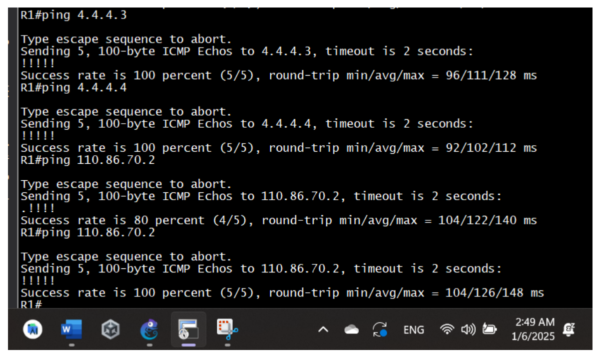


Updating the RIP for R3, to include the new PC's network in RIP we did the same steps as before.





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.



To config RIP at AS400 we used these commands:

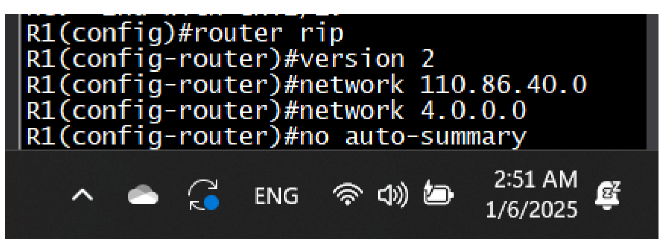
router rip

version 2

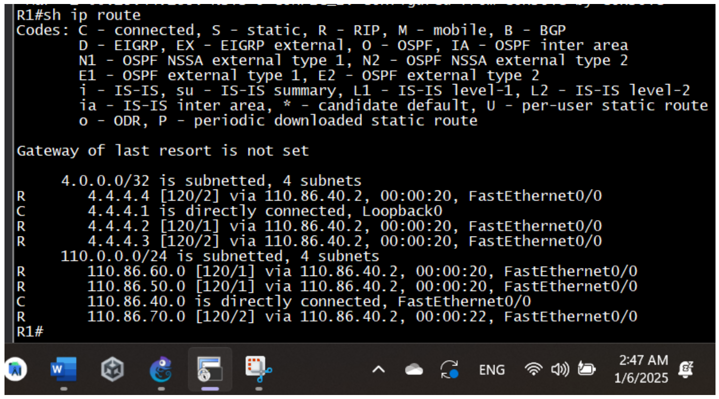
network <CONNECTED_NETWORK>

no auto-summary

for example, **R1**:



We used sh ip route to make sure:



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:

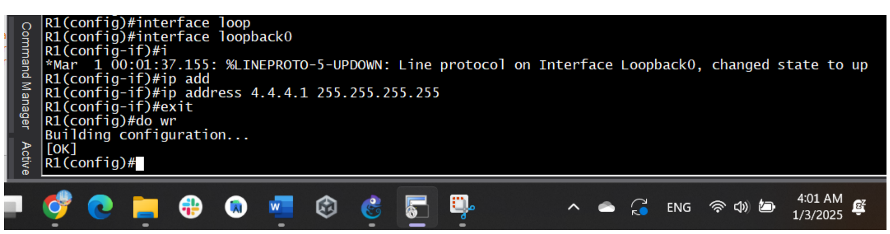
We firstly assign a unique IP for each router loopback using these commands:

`interface loopback0`

`ip address <ip_address> 255.255.255.255`

`no shutdown`

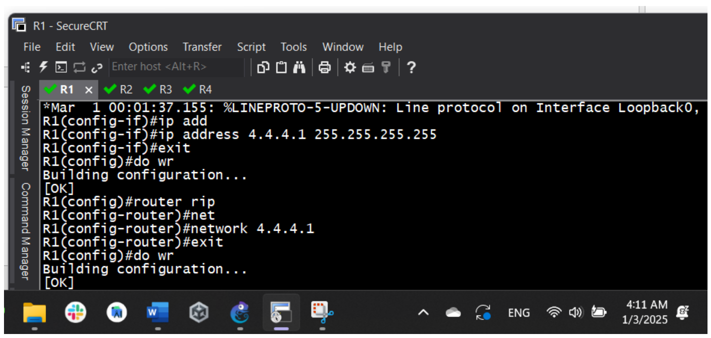
This is an example of how we added the loopback IP in R1 (**AS400**):



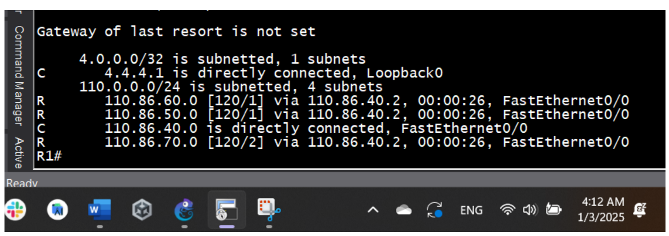
The rest of the router’s loopback IPs:

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

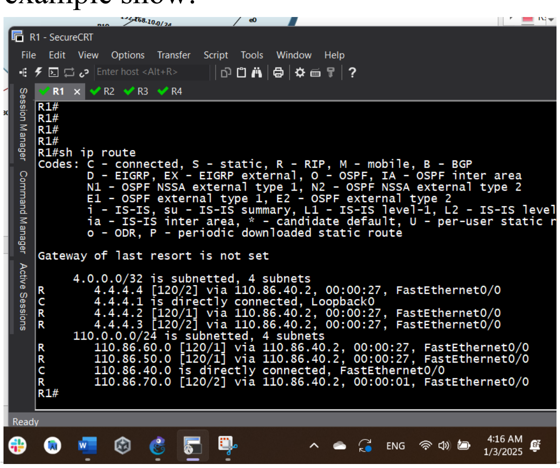
Then, we advertised Loopback Interfaces in RIP to ensure that they are reachable by other routers within the same AS, allowing stable BGP neighbor relationships.



And make sure that it added in the routing table:

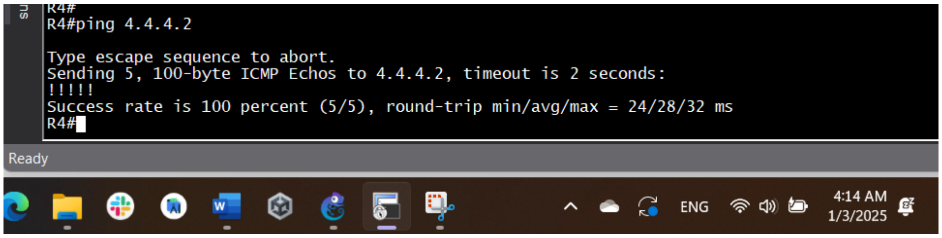


Then, we done this for all other routers in this AS400, and the final table for an R1 for example show:



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:



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:

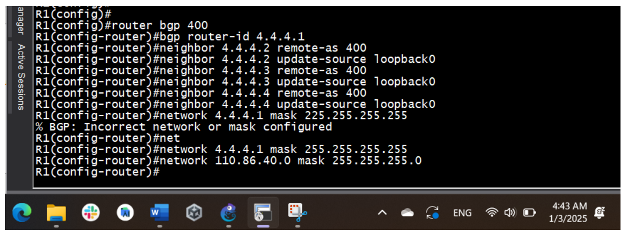
router bgp <AS_NUMBER>

neighbor <LOOPBACK_IP> remote-as <AS_NUMBER>

neighbor <LOOPBACK_IP> update-source loopback0

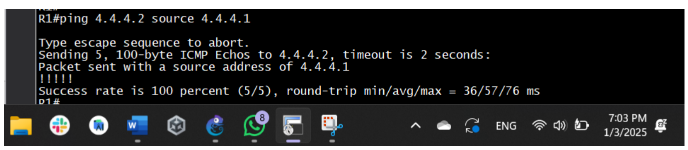
network <CONNECTED_NETWORK> mask <SUBNET_MASK>

For example, R1 at AS400:

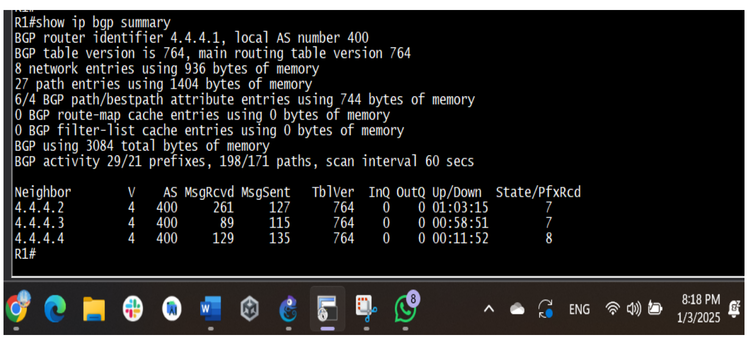


```
R1(config)#
R1(config)#router bgp 400
R1(config-router)#bgp router-id 4.4.4.1
R1(config-router)#neighbor 4.4.4.2 remote-as 400
R1(config-router)#neighbor 4.4.4.2 update-source loopback0
R1(config-router)#neighbor 4.4.4.3 remote-as 400
R1(config-router)#neighbor 4.4.4.3 update-source loopback0
R1(config-router)#neighbor 4.4.4.4 remote-as 400
R1(config-router)#neighbor 4.4.4.4 update-source loopback0
R1(config-router)#network 4.4.4.1 mask 225.255.255.255
% 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)#
```

Pining R2 loopback from R1 loopback successfully:

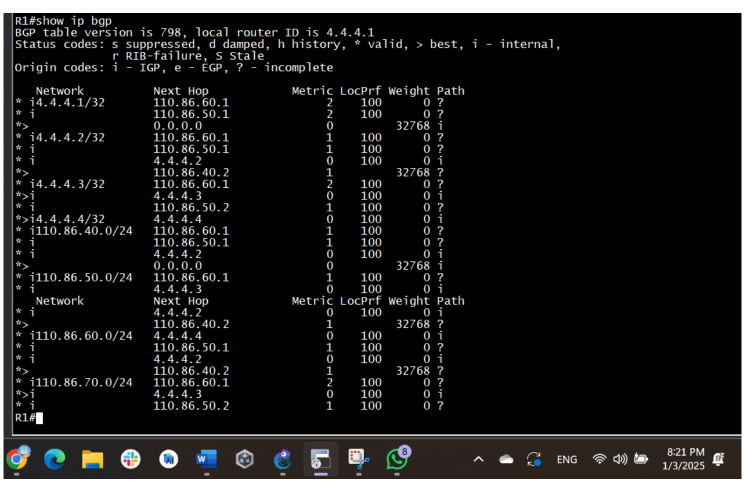


```
R1#ping 4.4.4.2 source 4.4.4.1

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
R1#
```



```
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#
```

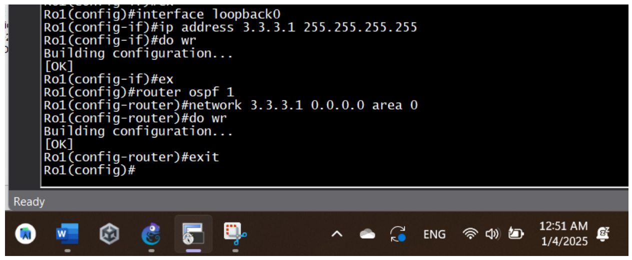
The show ip bgp summary output confirms that iBGP peering is active on R1 in AS400. The router has established sessions with three neighbors (4.4.4.4), and prefixes are being exchanged correctly, with 7, 8, and 8 prefixes received, verifying successful iBGP configuration and stable connectivity.

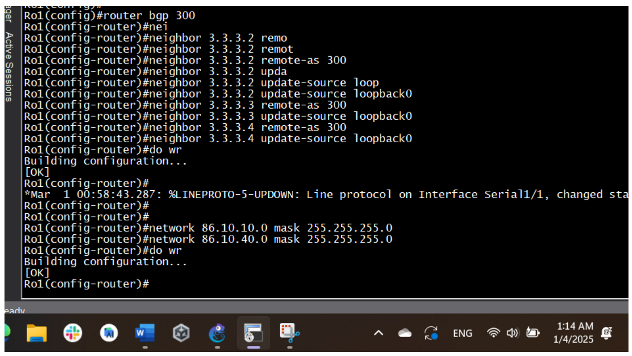


```
R1#show ip bgp
BGP table version is 798, local router ID is 4.4.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i4.4.4.1/32       110.86.60.1              2    100      0 ?
* i                 110.86.50.1              2    100      0 ?
*>                  0.0.0.0                  0         32768 i
* i4.4.4.2/32       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              1         32768 ?
* i4.4.4.3/32       110.86.60.1              2    100      0 ?
*>i                 4.4.4.3                  0    100      0 i
* i                 110.86.50.2              1    100      0 ?
*>i4.4.4.4/32       4.4.4.4                  0    100      0 i
* i110.86.40.0/24   110.86.60.1              1    100      0 ?
* i                 110.86.50.1              1    100      0 ?
* i                 4.4.4.2                  0    100      0 i
*>                  0.0.0.0                  0         32768 i
* i110.86.50.0/24   110.86.60.1              1    100      0 ?
* i                 4.4.4.3                  0    100      0 i
   Network          Next Hop            Metric LocPrf Weight Path
* i                 4.4.4.2                  0    100      0 i
*>                  110.86.40.2              1         32768 ?
* i110.86.60.0/24   4.4.4.4                  0    100      0 i
* i                 110.86.50.1              1    100      0 ?
* i                 4.4.4.2                  0    100      0 i
*>                  110.86.40.2              1         32768 ?
* i110.86.70.0/24   110.86.60.1              2    100      0 ?
*>i                 4.4.4.3                  0    100      0 i
* i                 110.86.50.2              1    100      0 ?
R1#
```

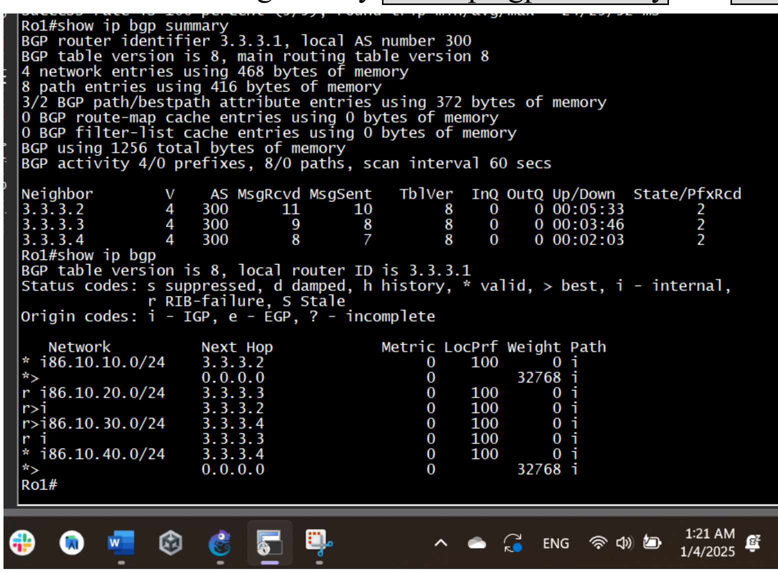
The output shows the routing table for R1 in AS400, with prefixes like 4.4.4.4/32 and /24 routes being propagated correctly. The Next Hop column confirms reachability via neighbors' loopback IPs, such as 110.86.50.1. Metrics like Weight and LocPrf indicate proper iBGP behavior, which verifying successful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

For **AS300** we did the same steps and config them to OSPF, for example **Ro1**:





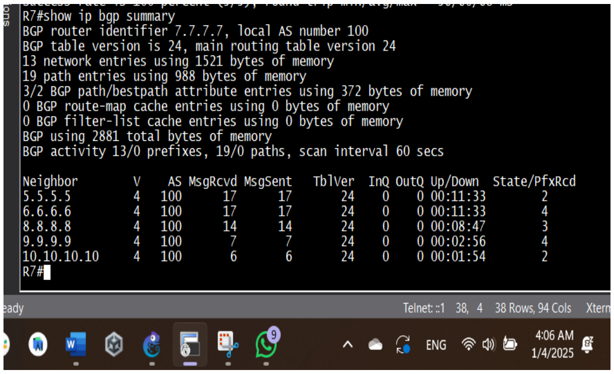
And making sure by show ip bgp summary and show ip bgp commands:

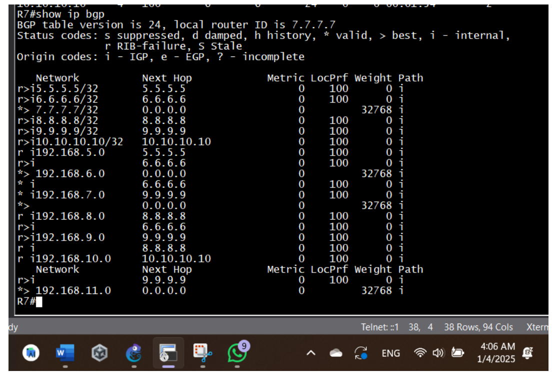


The image show that iBGP on **Ro1 in AS300** is active, with three neighbors exchanging **8 prefixes each**. The routing table shows routes like 186.10.10.0/24 and 186.10.20.0/24 are reachable via correct next hops, verifying proper iBGP and route propagation.

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.

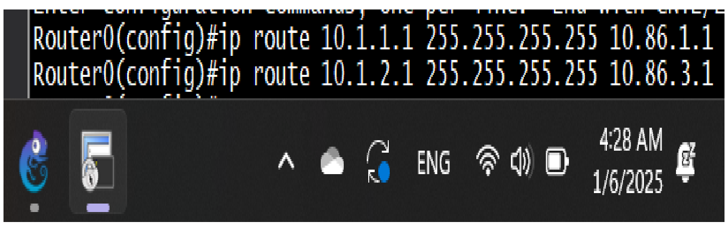
We used these commands to check show ip bgp summary and show ip bgp:



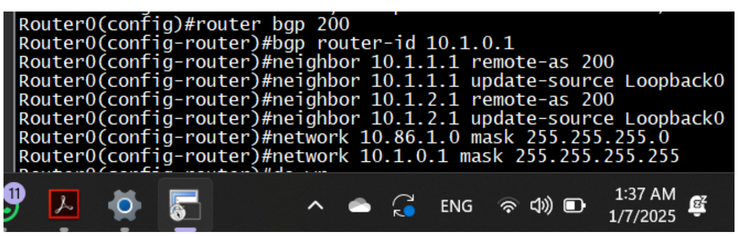


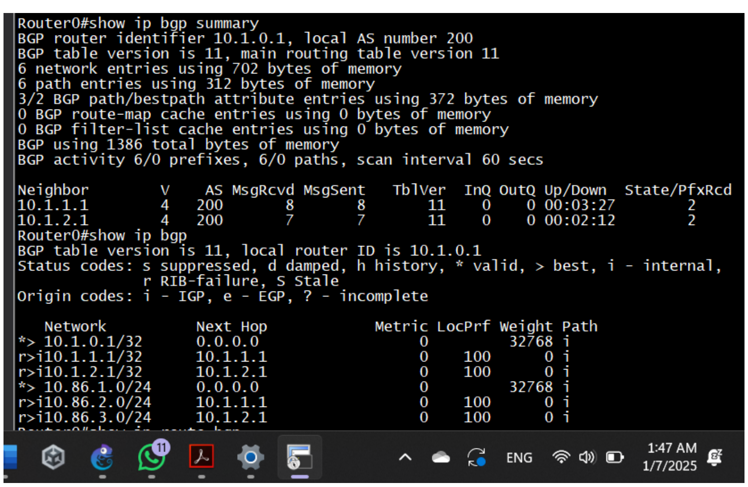
The images above show that iBGP peering on R7 in AS100 is successfully established, with neighbors exchanging prefixes as expected. The BGP routing table confirms that the advertised routes are correctly propagated and reachable via the appropriate next hops.

For **AS200** with static routing:

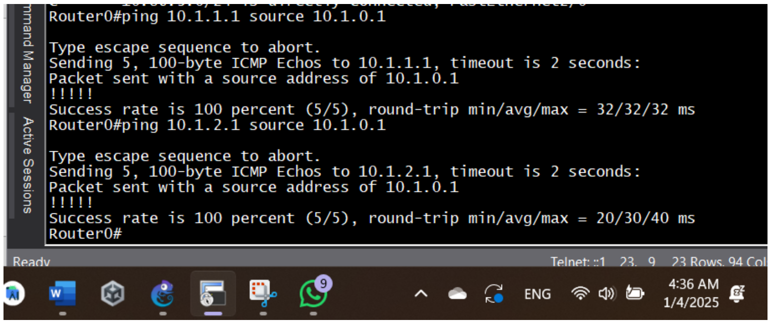


The routes direct traffic for the loopback IPs 10.1.1.1 and 10.1.2.1 through the next hops 10.86.1.1 and 10.86.3.1, respectively. These routes ensure that the loopback addresses of other routers in AS200 are reachable.





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.



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:

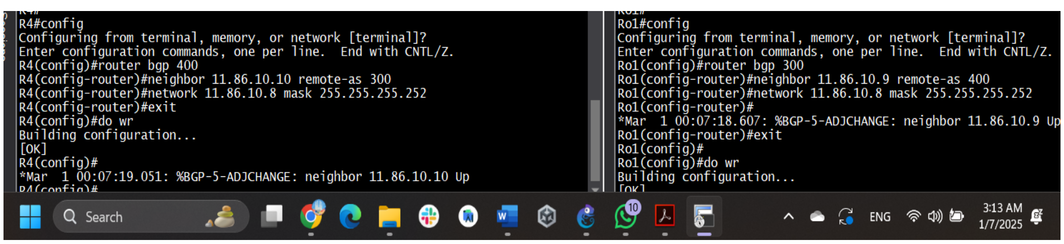
router bgp <LOCAL_AS_NUMBER>

neighbor <NEIGHBOR_IP> remote-as <NEIGHBOR_AS_NUMBER>

network <CONNECTED_NETWORK> mask <SUBNET_MASK>

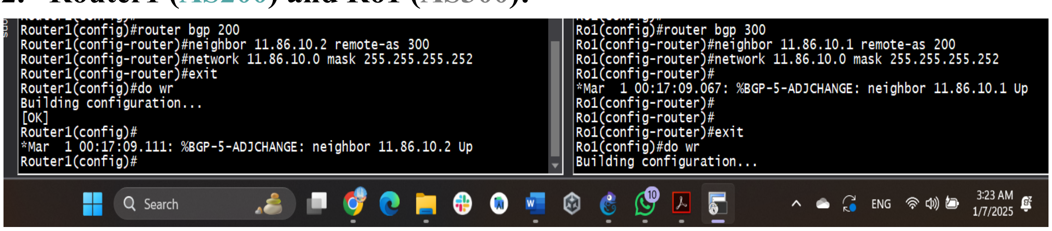
and here how it done for each line between two ASes in the topology:

1. R4 (AS400) and Ro1 (AS300):



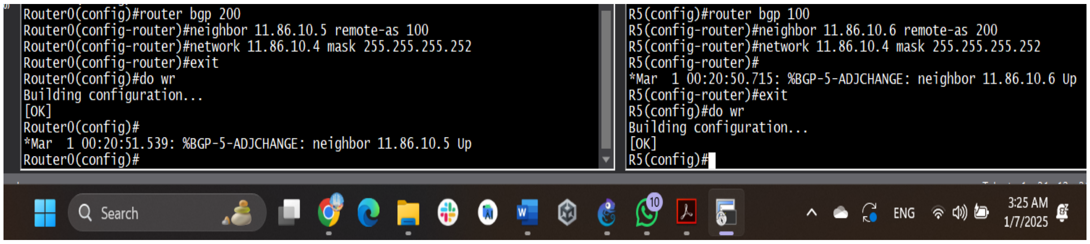
The image shows the successful eBGP configuration between R4 and Ro1. Both routers establish a neighbor relationship using their directly connected network (11.86.10.8/30) and advertise this network in their respective BGP configurations. The log messages confirm that the eBGP session is established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

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



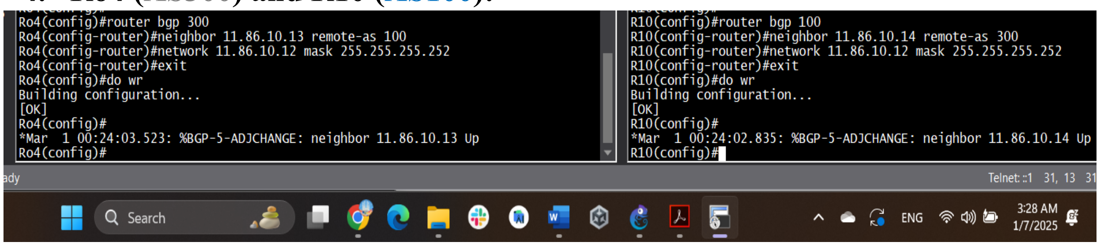
Both routers establish a neighbor relationship over the network 11.86.10.0/30 and advertise it in their respective BGP configurations.

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



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

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



The eBGP session was established successfully.

Traffic Engineering

Route Optimization: Aggregate Prefixes in AS200

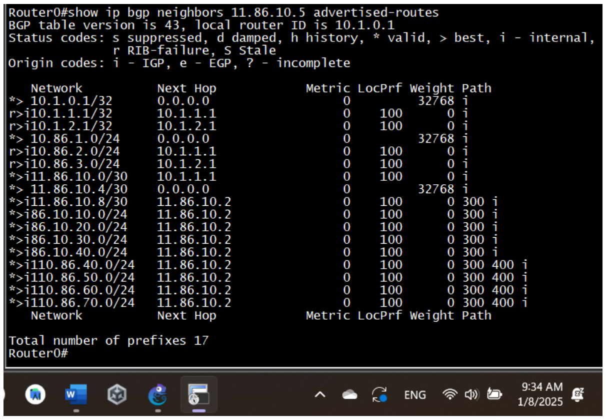
we will optimize routing in AS200 (IX Network) by implementing prefix aggregation. This reduces the number of advertised routes to neighboring autonomous systems, improving network scalability and efficiency. By aggregating multiple smaller prefixes into a single summarized route, the BGP table size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

Pre-Optimization State

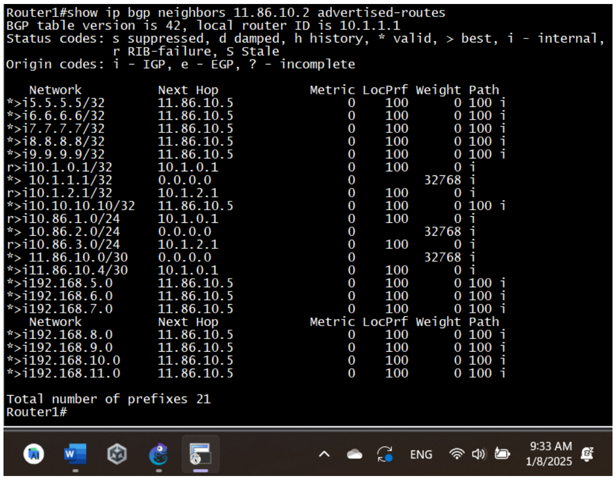
We used:

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

for each Router0 and Router1 in AS200

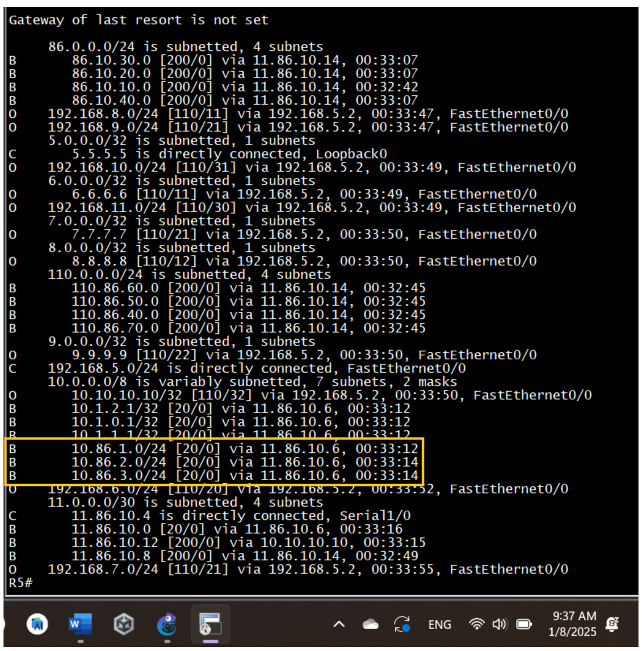


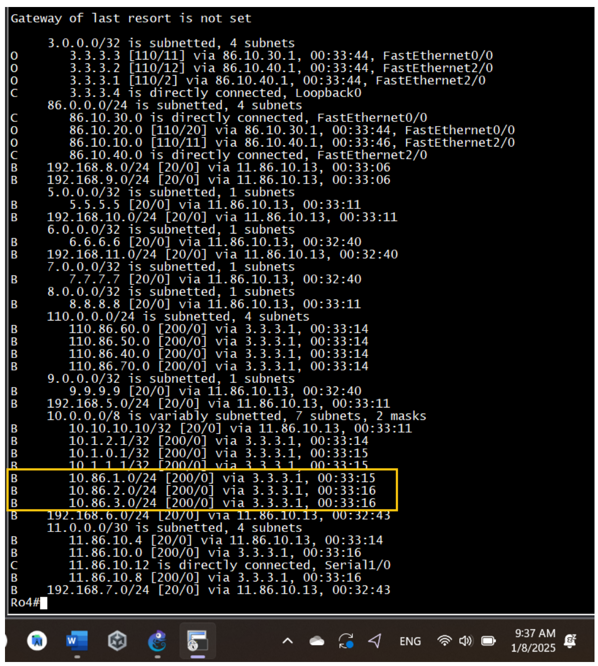
The image shows **17 prefixes being advertised** to the neighboring router **11.86.10.5** in AS100. These include individual routes like 10.86.1.0/24, 10.86.2.0/24, and 10.86.3.0/24, as well as smaller routes like 10.1.1.0/32. Advertising each route separately makes the BGP table on the neighbor larger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



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:





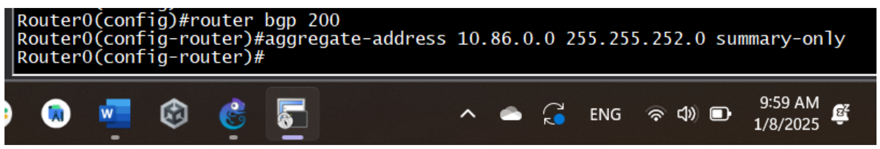
The routing tables for Ro1 (AS300) and R5 (AS100) before optimization show a significant number of individual routes being advertised.

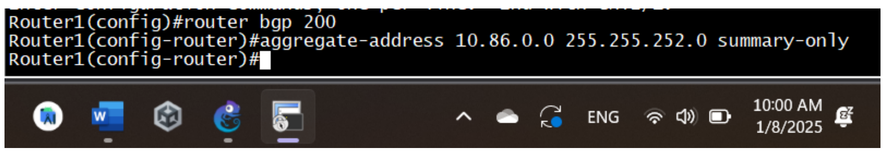
Implementation of Prefix Aggregation:

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:

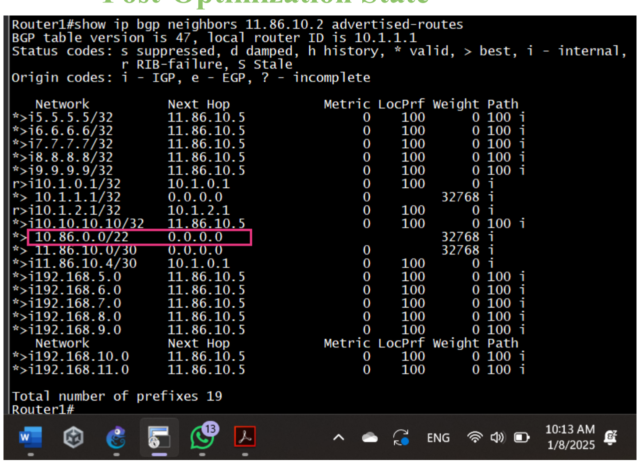
aggregate-address 10.86.0.0 255.255.252.0 summary-only

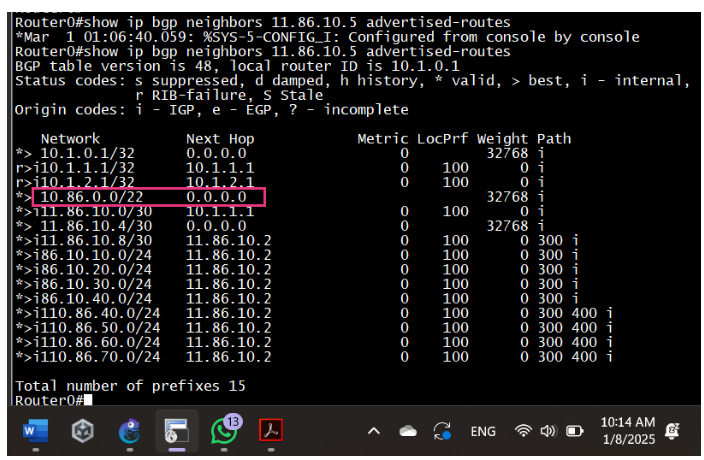




This command combines the individual subnets into a single summarized route, reducing the number of advertised prefixes to neighboring routers and optimizing the routing table.

Post-Optimization State





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.

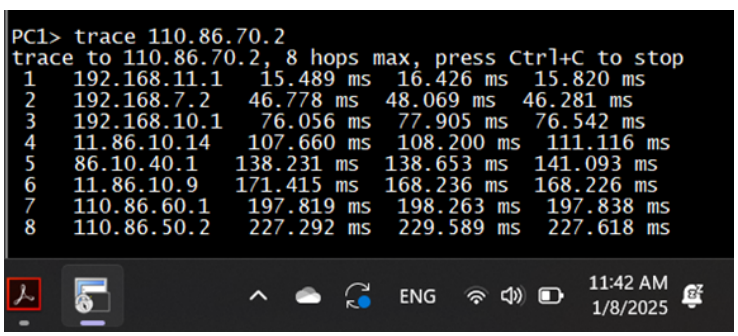
This process is particularly significant in real-world networking as it helps minimize the load on hardware, especially in large-scale networks. By consolidating routes, it improves scalability and efficiency, which are critical for enterprise and ISP networks.

Redundancy Testing: Simulating Link Failures

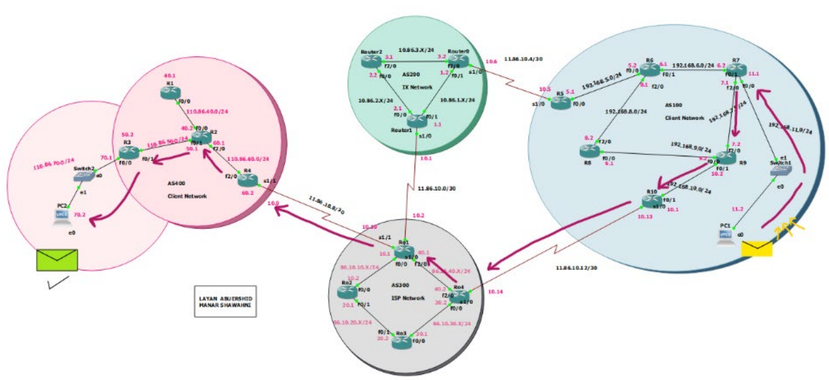
The purpose of this section is to test the network's redundancy by simulating a link failure between AS100 and AS300. The goal is to verify that traffic is automatically rerouted through alternative paths, ensuring uninterrupted connectivity and demonstrating the reliability of the network's design.

Pre-Failure State

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



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

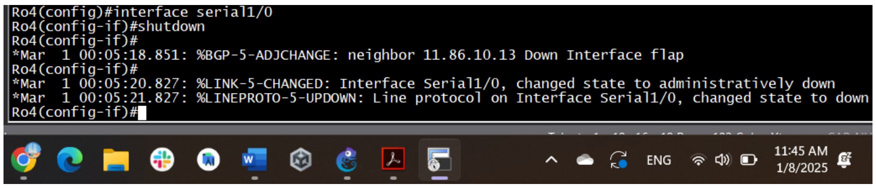


This path flows directly through the link between **Ro4 (AS300)** and **R10 (AS100).**

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.

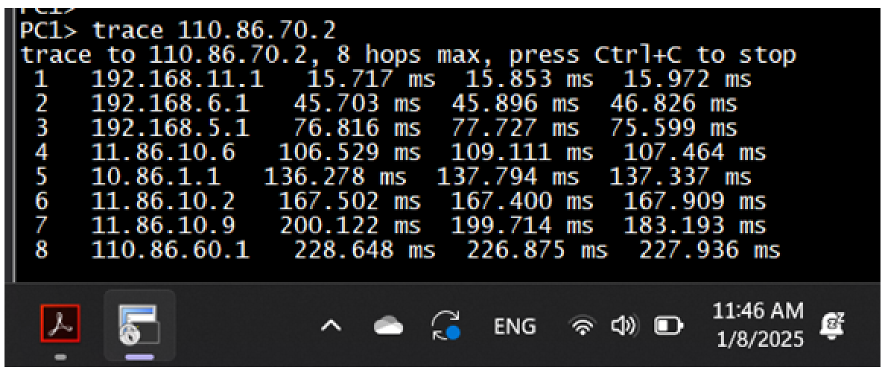
We shut down the Ro4 serial interface as shown in the figure below:



```
Ro4(config)#interface serial1/0
Ro4(config-if)#shutdown
Ro4(config-if)#
*Mar  1 00:05:18.851: %BGP-5-ADJCHANGE: neighbor 11.86.10.13 Down Interface flap
Ro4(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
Ro4(config-if)#
```

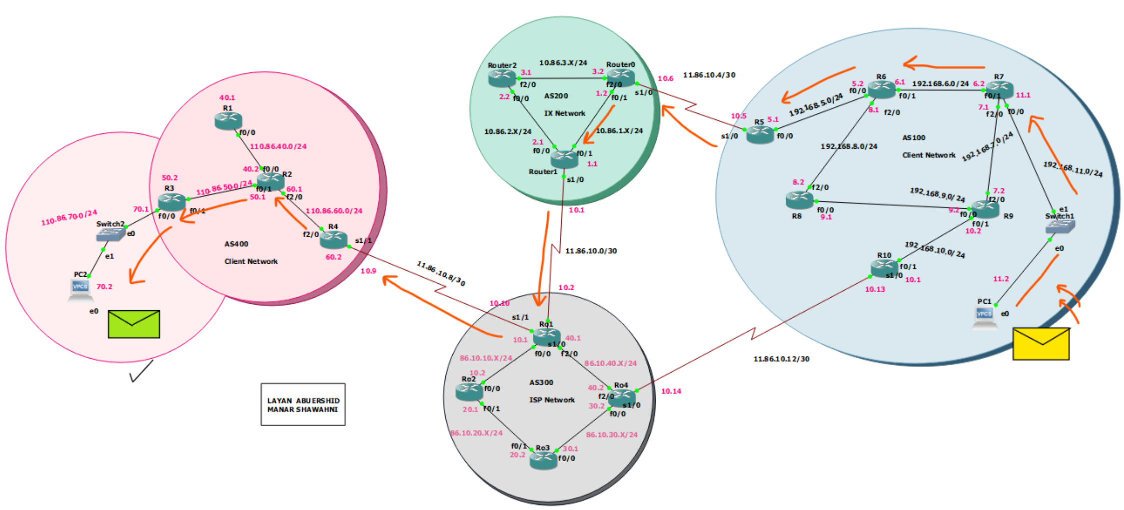
Post-Failure State

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

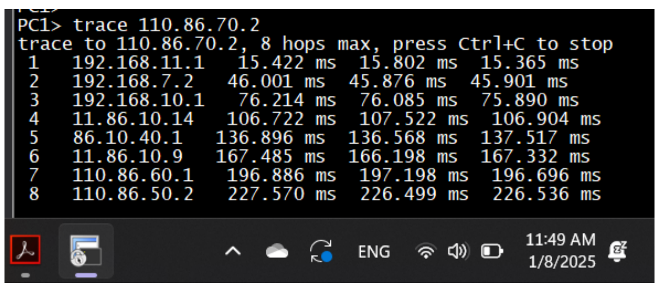
The path shifted from the direct connection between AS100 and AS300 to an alternative route via AS200, demonstrating the effectiveness of the network's redundancy. This ensures uninterrupted connectivity despite the link failure, highlighting the reliability of BGP's rerouting mechanisms in a non-stub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

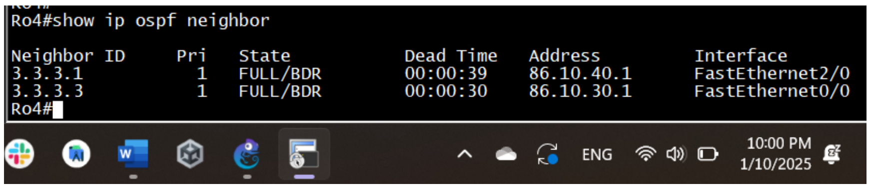
This process is particularly significant in real-world networking as it helps minimize the load on hardware, especially in large-scale networks. By consolidating routes, it improves scalability and efficiency, which are critical for enterprise and ISP networks.

Test Scenarios

OSPF Neighbor Verification

On Router Ro4 at AS300

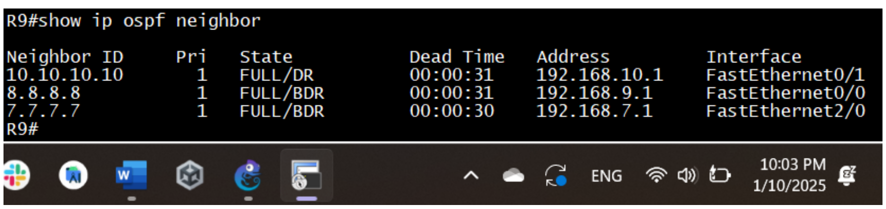
The output from the show ip ospf neighbor command on Router R04 shows that it has successfully established OSPF adjacencies with two neighboring routers, identified by their Router IDs (3.3.3.1 and 3.3.3.3). Both neighbors are in the FULL/BDR state, indicating fully established adjacencies and proper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



```
Ro4#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
Ro4#
```


On Router R9 at AS100

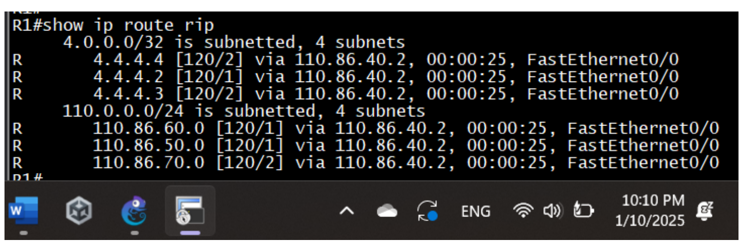
The output on Router R9 reveals three fully established OSPF adjacencies with neighboring routers. The neighbors, identified by their Router IDs, are in the FULL/DR and FULL/BDR states, indicating proper OSPF functionality.



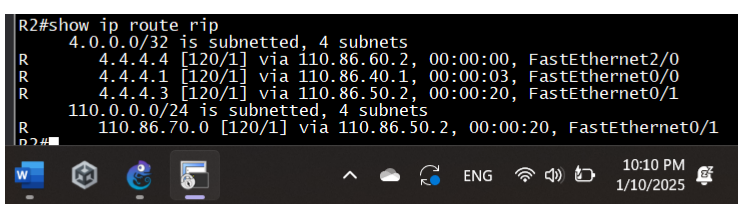
```
R9#show ip ospf neighbor

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

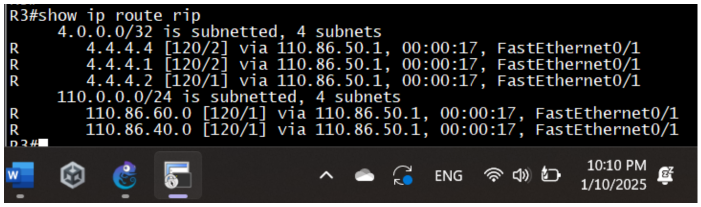
By using show ip route rip command:



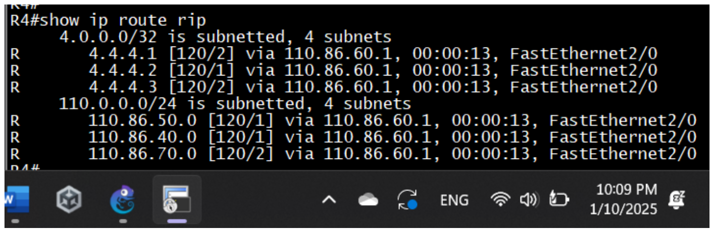
```
R1#show ip route rip
     4.0.0.0/32 is subnetted, 4 subnets
R       4.4.4.4 [120/2] via 110.86.40.2, 00:00:25, FastEthernet0/0
R       4.4.4.2 [120/1] via 110.86.40.2, 00:00:25, FastEthernet0/0
R       4.4.4.3 [120/2] via 110.86.40.2, 00:00:25, FastEthernet0/0
     110.0.0.0/24 is subnetted, 4 subnets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

```
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
```



```
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
R3#
```



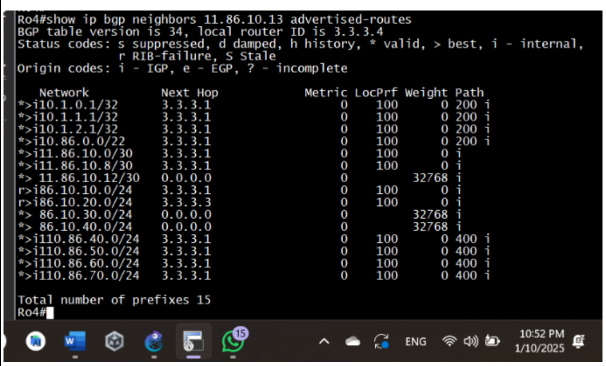
```
R4#show ip route rip
     4.0.0.0/32 is subnetted, 4 subnets
R       4.4.4.1 [120/2] via 110.86.60.1, 00:00:13, FastEthernet2/0
R       4.4.4.2 [120/1] via 110.86.60.1, 00:00:13, FastEthernet2/0
R       4.4.4.3 [120/2] via 110.86.60.1, 00:00:13, FastEthernet2/0
     110.0.0.0/24 is subnetted, 4 subnets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

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

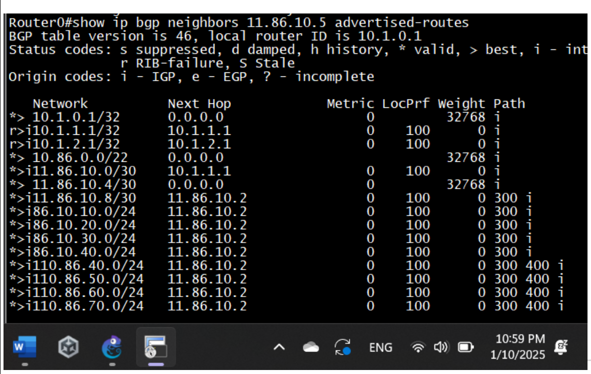
for Ro4 at AS300:



```
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
*>i10.1.0.1/32      3.3.3.1                  0    100      0 200 i
*>i10.1.1.1/32      3.3.3.1                  0    100      0 200 i
*>i10.1.2.1/32      3.3.3.1                  0    100      0 200 i
*>i10.86.0.0/22     3.3.3.1                  0    100      0 200 i
*>i11.86.10.0/30    3.3.3.1                  0    100      0 i
*>i11.86.10.8/30    3.3.3.1                  0    100      0 i
*> 11.86.10.12/30   0.0.0.0                  0         32768 i
r>i86.10.10.0/24    3.3.3.1                  0    100      0 i
r>i86.10.20.0/24    3.3.3.3                  0    100      0 i
*> 86.10.30.0/24    0.0.0.0                  0         32768 i
*> 86.10.40.0/24    0.0.0.0                  0         32768 i
*>i110.86.40.0/24   3.3.3.1                  0    100      0 400 i
*>i110.86.50.0/24   3.3.3.1                  0    100      0 400 i
*>i110.86.60.0/24   3.3.3.1                  0    100      0 400 i
*>i110.86.70.0/24   3.3.3.1                  0    100      0 400 i

Total number of prefixes 15
Ro4#
```

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.

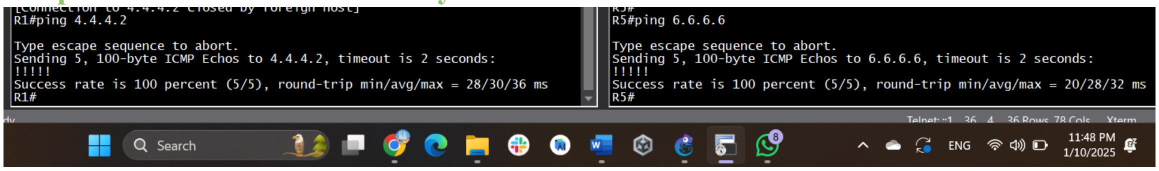
For Router0 at AS200:



```
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                            32768 i
*>i11.86.10.0/30    10.1.1.1                 0    100      0 i
*> 11.86.10.4/30    0.0.0.0                  0         32768 i
*>i11.86.10.8/30    11.86.10.2               0    100      0 300 i
*>i86.10.10.0/24    11.86.10.2               0    100      0 300 i
*>i86.10.20.0/24    11.86.10.2               0    100      0 300 i
*>i86.10.30.0/24    11.86.10.2               0    100      0 300 i
*>i86.10.40.0/24    11.86.10.2               0    100      0 300 i
*>i110.86.40.0/24   11.86.10.2               0    100      0 300 400 i
*>i110.86.50.0/24   11.86.10.2               0    100      0 300 400 i
*>i110.86.60.0/24   11.86.10.2               0    100      0 300 400 i
*>i110.86.70.0/24   11.86.10.2               0    100      0 300 400 i
```

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.

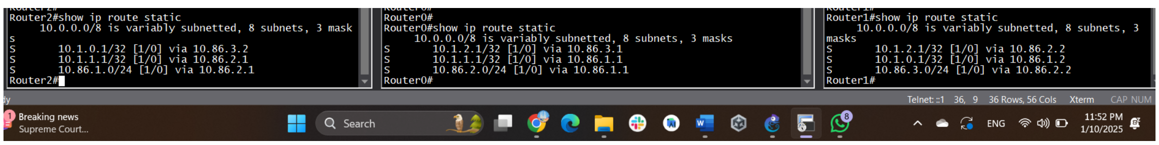
Loopback Interface Reachability



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 pining 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.

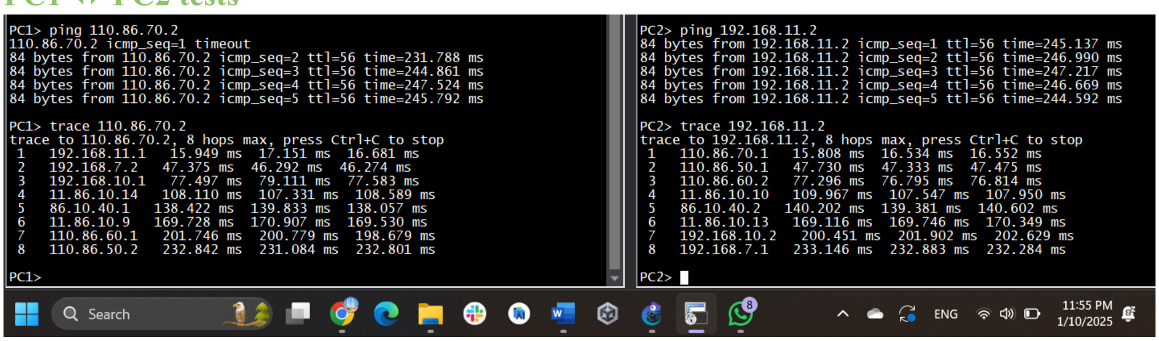
Static Route Verification in AS200

show ip route static



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

PC1 ⇔ PC2 tests



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.