University of Dhaka

Department of Computer Science and Engineering

CSE-3111 : Computer Networking Lab

Lab Report 8 : Implementation of BGP Protocol

Submitted By:

Afser Adil Olin

Roll No : AE-47

Anika Tabassum

Roll No : Rk-61

Submitted On :

April 28, 2023

Submitted To :

Dr. Md. Abdur Razzaque

Md Mahmudur Rahman

Md. Ashraful Islam

Md. Fahim Arefin

Contents

1 Introduction

The Border Gateway Protocol (BGP) is a standardized protocol used by internet service providers (ISPs) to exchange routing information between autonomous systems (AS). BGP enables the routing of data traffic over the internet by exchanging information about the reachability of IP addresses and network prefixes.

1.1 Objectives

- Understand the basics of BGP protocol and its functions.
- Implement BGP protocol on a simulated network using Cisco Packet Tracer software.
- Configure BGP neighbors, routes and route-maps.
- Verify the BGP configuration using show commands.

2 Theory

2.1 Autonomous Systems (ASes)

An Autonomous System (AS) is a network or group of networks under a single administrative domain, which is responsible for routing data packets within the network. An AS is assigned a unique Autonomous System Number (ASN), which is used to identify the AS when exchanging routing information with other ASes.

ASes use BGP to control the flow of traffic between different networks. By exchanging routing information, ASes can determine the best path to reach a particular network and route data packets accordingly. ASes can also implement policies to control the flow of traffic, such as preferring one path over another or restricting the flow of traffic through specific connections.

2.2 Path Vector Routing

Path Vector Routing is a type of routing protocol used in computer networks to determine the best path for routing data packets between different autonomous systems (AS). In this type of routing protocol, each router maintains a path vector that contains a list of autonomous systems that the route has traversed to reach the destination network.

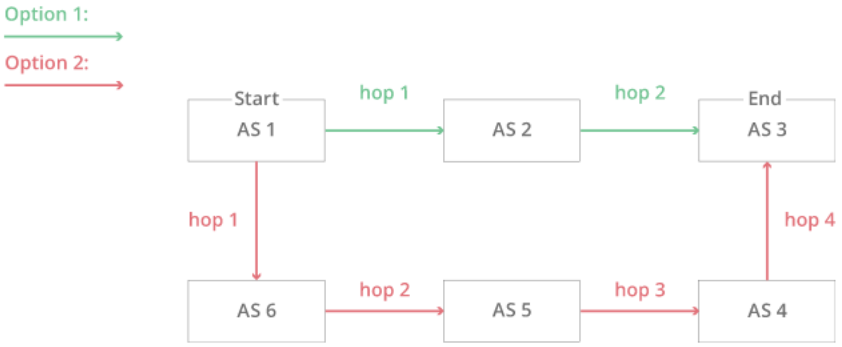
BGP uses path vector routing to make routing decisions based on the path vector, which includes the AS Path attribute. The AS Path attribute is a sequence of AS numbers that the route has traversed from its origin AS to the destination network. This attribute helps BGP to prevent routing loops and ensure efficient routing of data packets.

Path vector routing is different from other types of routing protocols, such as distance vector and link-state protocols. In distance vector routing, routers maintain information about the distance or cost to reach a destination network. In link-state routing, routers maintain information about the entire network topology.

2.3 Border Gateway Protocol (BGP)

The Border Gateway Protocol (BGP) is a protocol used for exchanging routing information between different autonomous systems (AS) on the internet. BGP is responsible for exchanging routing information between different routers and making decisions on the best path for routing data packets. BGP is used by Internet Service Providers (ISPs) to exchange information about the reachability of IP addresses and network prefixes.

If we continue to think of BGP as the Postal Service of the Internet, ASes are like individual post office branches. A town may have hundreds of mailboxes, but the mail in those boxes must go through the local postal branch before being routed to another destination. The internal routers within an AS are like mailboxes. They forward their outbound transmissions to the AS, which then uses BGP routing to get these transmissions to their destinations.



The diagram above illustrates a simplified version of BGP. In this version there are only six ASes on the Internet. If AS1 needs to route a packet to AS3, it has two different options:

Hopping to AS2 and then to AS3:

AS2 → AS3

Or hopping to AS6, then to AS5, AS4, and finally to AS3:

AS6 → AS5 → AS4 → AS3

In this simplified model, the decision seems straightforward. The AS2 route requires fewer hops than the AS6 route, and therefore it is the quickest, most efficient route. Now imagine that there are hundreds of thousands of ASes and that hop count is only one part of a complex route selection algorithm. That is the reality of BGP routing on the Internet.

2.4 Operations of BGP Autonomous Systems

ASes typically belong to Internet service providers (ISPs) or other large organizations, such as tech companies, universities, government agencies, and scientific institutions. Each AS wishing to exchange routing information must have a registered autonomous system number (ASN). Internet Assigned Numbers Authority (IANA) assigns ASNs to Regional Internet Registries (RIRs), which then assigns them to ISPs and networks. ASNs are 16 bit numbers between one and 65534 and 32 bit numbers between 131072 and 4294967294. As of 2018, there are approximately 64,000 ASNs in use worldwide. These ASNs are only required for external BGP.

2.5 BGP Algorithm

The BGP (Border Gateway Protocol) algorithm is used by BGP routers to select the best path for forwarding traffic. The BGP algorithm involves multiple steps, including path discovery, path selection, and path advertisement.

The following is a brief overview of the BGP algorithm:

1. **Path Discovery**: BGP routers discover paths to other BGP routers by exchanging BGP messages. Each BGP router builds a routing table that contains information about the available paths to other BGP routers.

2. **Path Selection**: BGP routers use a set of rules to select the best path for forwarding traffic. The path selection process involves comparing the attributes of the available paths, such as AS Path, Next Hop, Local Preference, Weight, Origin, MED, Community, and Aggregator.

3. **Path Advertisement**: BGP routers advertise the best path to their neighboring routers using BGP UPDATE messages. BGP routers can also advertise multiple paths for the same destination network, using different attributes to differentiate between the paths.

2.5.1 BGP Message Types

There are four types of BGP messages that routers exchange with each other:

- **OPEN Message:** The OPEN message is the first message sent by BGP routers to initiate a BGP session. The OPEN message contains information about the BGP router, including its AS number, BGP protocol version, and BGP capabilities.

- **UPDATE Message:** The UPDATE message is used to exchange routing information between BGP routers. The UPDATE message contains information about the network prefixes, their associated AS paths, and the attributes of the prefixes.

- **KEEPALIVE Message:** The KEEPALIVE message is used to maintain the BGP session between BGP routers. The KEEPALIVE message is sent at a regular interval to confirm that the BGP session is still active.

- **NOTIFICATION Message:** The NOTIFICATION message is used to report errors or exceptions in the BGP session. The NOTIFICATION message contains information about the error or exception and may terminate the BGP session.

The BGP protocol uses these messages to establish and maintain BGP sessions between BGP routers and exchange routing information between different autonomous systems. BGP routers use the UPDATE message to exchange information about the reachability of IP addresses and network prefixes. BGP uses the AS Path attribute in the UPDATE message to prevent routing loops and ensure efficient routing of data packets between different autonomous systems.

2.5.2 BGP Attributes

There are many BGP attributes, but some of the most commonly used ones include:

- **AS Path:** The AS Path attribute lists the AS numbers that a route has traversed to reach the BGP router. This attribute is used to prevent routing loops and ensure efficient routing of data packets between different autonomous systems.

- **Next Hop:** The Next Hop attribute specifies the IP address of the next router in the path to the destination network. This attribute is used to determine the next router that a packet should be sent to.

- **Local Preference:** The Local Preference attribute is used to influence the path selection process within the same autonomous system. BGP routers within the same AS will prefer routes with a higher Local Preference value.
- **Weight:** The Weight attribute is a Cisco-specific attribute that is used to influence the path selection process within the same router. Routes with a higher Weight value will be preferred over routes with a lower Weight value.
- **Origin:** The Origin attribute specifies how the route was originated. The possible values are IGP (Interior Gateway Protocol), EGP (Exterior Gateway Protocol), or incomplete. Routes that are originated by the local BGP router will have an Origin of incomplete.
- **MED (Multi-Exit Discriminator):** The MED attribute is used to influence the path selection process between different autonomous systems. It is used to advertise the preferred exit point for traffic leaving an AS.
- **Community:** The Community attribute is used to group routes together based on common properties. It is used to control routing policies and traffic engineering.
- **Aggregator:** The Aggregator attribute is used to identify the AS that performed route aggregation. It is used to optimize the size of routing tables.

BGP attributes can be manipulated using various techniques such as route maps, access lists, and prefix lists. BGP attributes are essential for controlling the flow of traffic across the Internet, and understanding BGP attributes is essential for network engineers working with large-scale networks.

2.5.3 How BGP Works

BGP routers exchange routing information using BGP messages, including OPEN, UPDATE, KEEPALIVE, and NOTIFICATION messages. The OPEN message is the first message sent by BGP routers to initiate a BGP session. The UPDATE message is used to exchange routing information between BGP routers. The KEEPALIVE message is used to maintain the BGP session between BGP routers. The NOTIFICATION message is used to report errors or exceptions in the BGP session.

2.6 Types of BGP

There are two types of BGP: eBGP (external BGP) and iBGP (internal BGP).

2.6.1 eBGP (external BGP):

eBGP is used to exchange routing information between different autonomous systems (AS). eBGP is typically used by ISPs (Internet Service Providers) to exchange routing information with other ISPs or with their customers.

2.6.2 iBGP (internal BGP):

iBGP is used to exchange routing information within the same autonomous system (AS). iBGP is typically used by large enterprises that have multiple locations connected by different ISPs. iBGP allows these enterprises to maintain consistent routing policies across their network and ensure efficient traffic routing.

2.6.3 Differences between External BGP and Internal BGP

Routes are exchanged and traffic is transmitted over the Internet using external BGP (eBGP). Autonomous systems can also use an internal version of BGP to route through their internal networks, which is known as internal BGP (iBGP). It should be noted that using internal BGP is NOT a requirement for using external BGP. Autonomous systems can choose from a number of internal protocols to connect the routers on their internal network.

External BGP is like international shipping. There are certain standards and guidelines that need to be followed when shipping a piece of mail internationally. Once that piece of mail reaches its destination country, it has to go through the destination country's local mail service to reach its final destination. Each country has its own internal mail service that does not necessarily follow the same guidelines as those of other countries. Similarly, each autonomous system can have its own internal routing protocol for routing data within its own network.

2.7 BGP Faults

BGP is a complex protocol that can be prone to various faults, including misconfiguration, route flapping, and route leaks. Misconfiguration can cause BGP routers to advertise incorrect routing information or select suboptimal paths. Route flapping occurs when BGP routers continuously advertise and withdraw routing information, causing network instability. Route leaks occur when BGP routers advertise routes that they are not authorized to advertise, potentially causing traffic to be routed through unauthorized paths.

3 Methodology

1. **Set up the network topology:**
   - Create a simulated network environment using virtual machines or containers
   - Configure the network topology with multiple BGP routers and switches
   - Assign unique IP addresses and AS numbers to each BGP router
2. **Configure the BGP routers:**
   - Configure BGP on each router using the appropriate commands and parameters
   - Define the BGP neighbor relationships between the routers
   - Advertise network prefixes and update BGP attributes as needed
3. **Test the BGP configuration:**
   - Verify BGP neighbor relationships using the show command on each router
   - Verify that BGP routing tables have been updated with the correct network prefixes and attributes
4. **Introduce faults to the network:**
   - Simulate network failures or misconfigurations to test the resilience of the BGP protocol
   - Observe the behavior of the BGP routers and network traffic during these faults
5. **Analyze the results:**

- Collect and analyze data on BGP performance, including convergence time, route flapping, and network throughput
- Compare the results to expected behavior and industry standards

6. **Draw conclusions:**

- Summarize the findings of the experiment and draw conclusions about the effectiveness of BGP in the tested network environment
- Identify potential areas for improvement or further experimentation

4 Network Topology Details

4.1 Graph

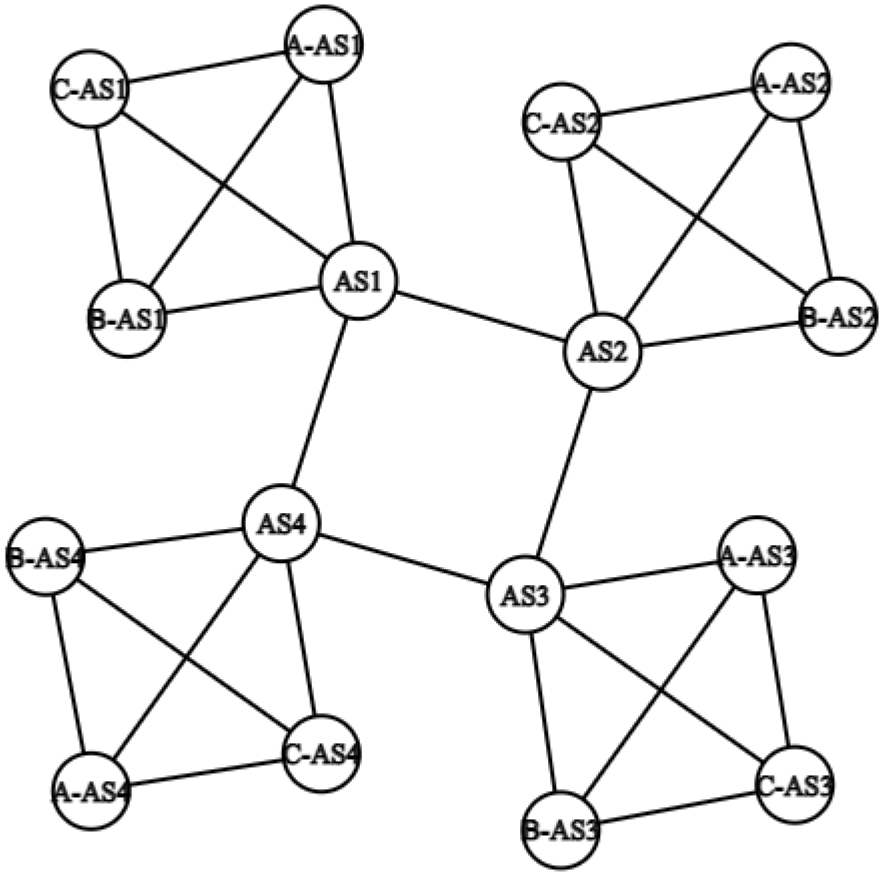


Figure 1: Graph used in our experiment

5 Implementation

5.1 Considerations for implementation

We have considered a few restrictions for implementation. Which are listed as follows

5.1.1 The graph has weight given by us

The graph is weighted and the weight was given by us randomly. BGP weights are configured locally by policy according to each AS with the various attributes. Globally it works like a unweighted graph with AShops being the condition for routing.

5.1.2 The graph is not a multigraph

The graph is not a multigraph. That is two nodes are not directly connected by more than one edge/link.

5.1.3 The graph is undirected

The graph is undirected because the connections are duplex.

5.1.4 Modeling the network

We have maintained the following abstractions to implement BGP

5.1.5 Mapping port numbers to IP addresses

Since we are working in a local machine we have abstracted IP addresses to port numbers. To get a working BGP in real life we can simply just change the prefix logic and point it to the port 179. (since BGP runs on port 179). Here we are considering a prefix to all numbers greater than the thousandth place of the decimal number of the port. In real life it is the binary prefix of a number.

5.1.6 All nodes in the AS have information of all other nodes in the AS

This is done by the array in the code. This is to create BGP connections to all other nodes in the AS. Although it could be argued that only the border nodes could have full connectivity, The algorithm would still work. But we wanted to maintain homogeneity.

5.1.7 Disregarding intra AS routing

Our implementation completely disregards intra AS routing as intra AS routing completely depends on each ISP and it's routing algorithm choice which could be very different. In theory our algorithm would still work if we added different Intra ISP routing. But that'd be too hard to test.

5.1.8 Setting Local Preference

Local preference is set at each AS's border nodes and is propagated throughout. Also, it is not shared with the outside nodes because we can't force our policy to other ASes.

5.1.9 Disregarding all messages except UPDATE

To simplify the network we have chosen not to send any messages by the OPEN, KEEPALIVE and NOTIFICATION message. Since we don't have to handle error conditions. We assume (rightfully) that all nodes are up and are successfully sending the data to other nodes. Our confidence is backed by the TCP protocol and the fact that we are using the same machine to simulate all nodes.

5.1.10 BGP attributes handled

Our goal was to show that we can infact change intra AS routing based on policy. Now in the real world this change can be affected by a large number of attributes. We choose to focus on the following to implement our algorithm simply

1. **AS_PATH** This is a must for routing based on hops
2. **NEXT_HOP** This is again a must to fill the forwarding table
3. **LOCAL_PREF** This is the attribute we have chosen to show that we can indeed influence our shortest path with policy.

5.2 Implementation and codes

In our implementation , we use AS1, AS2, AS3, AS4 as gateway router or ASes. $A - AS_i$, $B - AS_i$, $C - AS_i$ is the router A,B,C router connected to i'th AS. We also given some weight to the graph which will work as an local pref.

5.2.1 Messages:

1. **open :** This function try to establish a connection. If it fail to connect , it returns a notification.
2. **keepalive :** this function send a message after 10 second to keep the connection open. If any error occurs in the connection it send a notification.
3. **Update :** Any message regarding path or route is send via this function
4. **notification :** This function return a string containing "Error" to let us know there is a problem establishing connection

5.2.2 Router :

In router , we make two function to receive and send messages to it's AS.

5.2.3 Autonomous System:

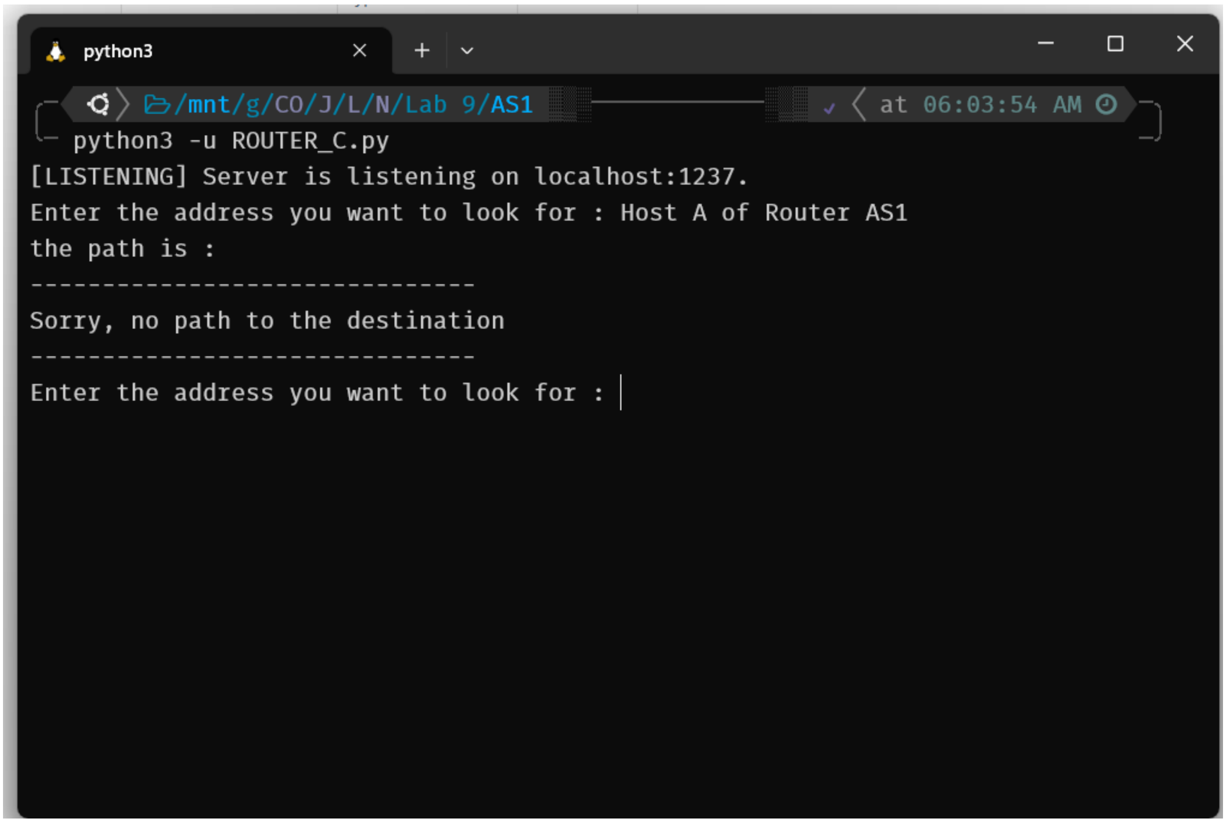
We make a function to receive messages. The message is sent to other router from this function after checking it's attributes and destination. If any message comes from it's router, we send to all ASes. If we receive a message from IBGP or another AS, we first check if the attributes it got is update able. If it is update able, we update the list and broadcast to ASes. Our implementation works like this :
Suppose , we want to learn the best route for a destination of host A of router X from the current router. Then we have to write "Host A of Router X". After that , it will give us the

path from current router to A otherwise it will tell us , there is no path exists.
We used some variable to save information

1. **PORT_NAME :** We map the port numbers with the respected name that were written in ”NAME.txt” file according to our implementation.
2. **NAME_ADDR :** We map the address to the router with their name and save it in this dictionary.
3. **MSG_PATH_LIST :** This is the list we maintain updating and storing the information about the route from X router to Y router.
4. **BGP_CONN :** This variable contains the routers that are currently is connected with BGP connection with its own.
5. **adj :** This variable is to store the adjacency list of the graph which is parsed from the ”router_path.txt” .

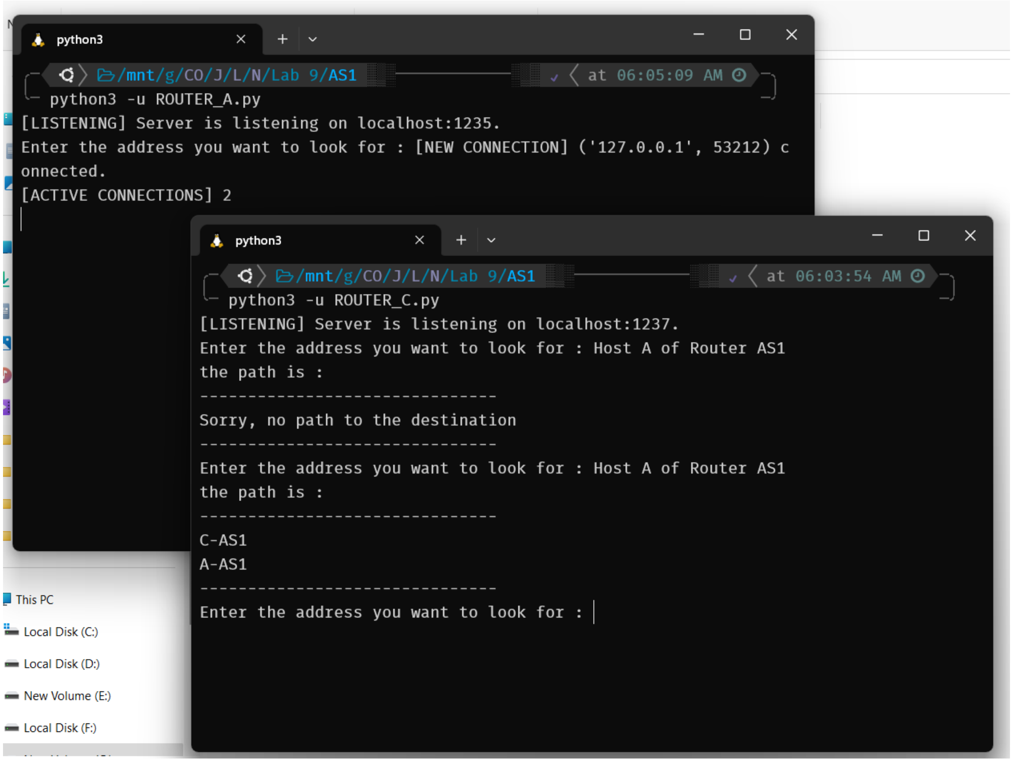
6 Experimental Results

We were trying to find route from Router C of AS1 and those are the result we got:



```
python3 -u ROUTER_C.py
[LISTENING] Server is listening on localhost:1237.
Enter the address you want to look for : Host A of Router AS1
the path is :
-------------------------------
Sorry, no path to the destination
-------------------------------
Enter the address you want to look for :
```

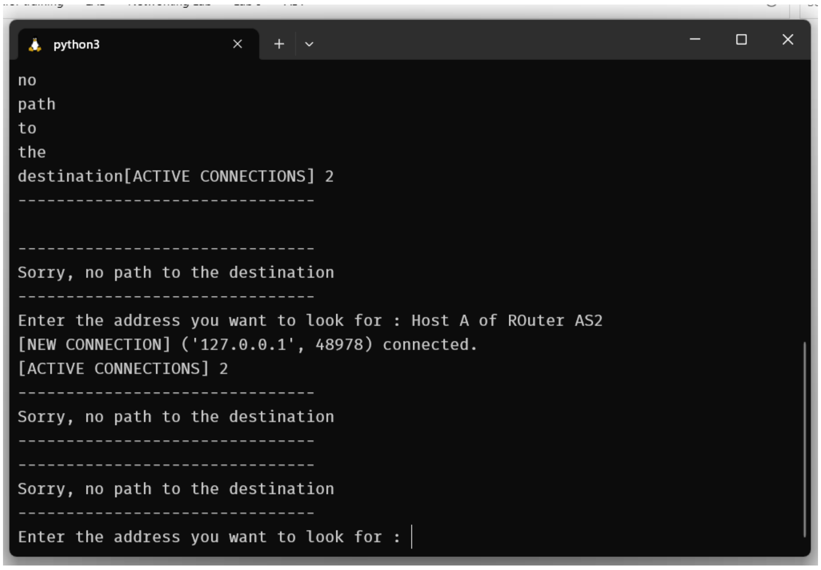
Figure 2: When the router A is unavailable in AS1



```
 /mnt/g/CO/J/L/N/Lab 9/AS1                         at 06:05:09 AM
 python3 -u ROUTER_A.py
[LISTENING] Server is listening on localhost:1235.
Enter the address you want to look for : [NEW CONNECTION] ('127.0.0.1', 53212) c
onnected.
[ACTIVE CONNECTIONS] 2
```

```
 /mnt/g/CO/J/L/N/Lab 9/AS1                         at 06:03:54 AM
 python3 -u ROUTER_C.py
[LISTENING] Server is listening on localhost:1237.
Enter the address you want to look for : Host A of Router AS1
the path is :
-------------------------------
Sorry, no path to the destination
-------------------------------
Enter the address you want to look for : Host A of Router AS1
the path is :
-------------------------------
C-AS1
A-AS1
-------------------------------
Enter the address you want to look for :
```

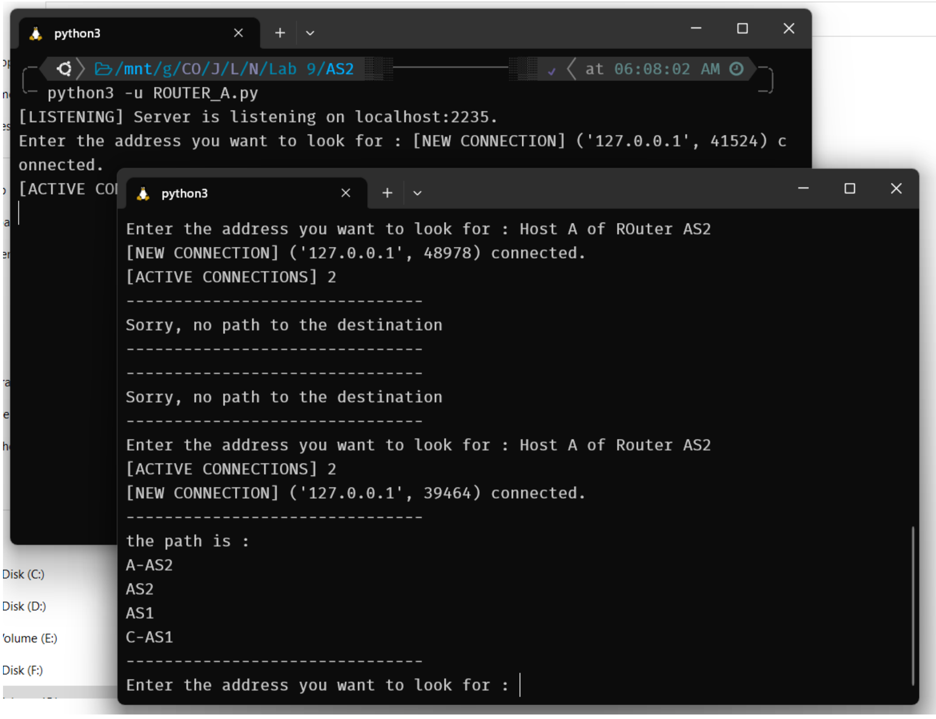
Figure 3: After router A becomes available in AS1



```
no
path
to
the
destination[ACTIVE CONNECTIONS] 2
-------------------------------

-------------------------------
Sorry, no path to the destination
-------------------------------
Enter the address you want to look for : Host A of ROuter AS2
[NEW CONNECTION] ('127.0.0.1', 48978) connected.
[ACTIVE CONNECTIONS] 2
-------------------------------
Sorry, no path to the destination
-------------------------------
-------------------------------
Sorry, no path to the destination
-------------------------------
Enter the address you want to look for :
```

Figure 4: When the router A is unavailable in AS2



```
/mnt/g/CO/J/L/N/Lab 9/AS2                    at 06:08:02 AM
python3 -u ROUTER_A.py
[LISTENING] Server is listening on localhost:2235.
Enter the address you want to look for : [NEW CONNECTION] ('127.0.0.1', 41524) c
onnected.
[ACTIVE CO
```

```
Enter the address you want to look for : Host A of ROuter AS2
[NEW CONNECTION] ('127.0.0.1', 48978) connected.
[ACTIVE CONNECTIONS] 2
-------------------------------
Sorry, no path to the destination
-------------------------------
-------------------------------
Sorry, no path to the destination
-------------------------------
Enter the address you want to look for : Host A of Router AS2
[ACTIVE CONNECTIONS] 2
[NEW CONNECTION] ('127.0.0.1', 39464) connected.
-------------------------------
the path is :
A-AS2
AS2
AS1
C-AS1
-------------------------------
Enter the address you want to look for :
```

Figure 5: After router A becomes available in AS2

7 Experience

- We should setup the network topology in GNS3 using routers and switches.
- We should configure the IP addresses and interfaces for each device in the network.
- We should enable the BGP protocol on the routers using the appropriate commands.
- We should configure the BGP attributes, such as the AS number, BGP neighbors, and network prefixes.
- We should verify that the BGP session is established between the routers using the ”show ip bgp summary” command.
- We should check the BGP routing table on each router using the ”show ip bgp” command and ensure that the expected network prefixes are present.
- Test the BGP routing by pinging different devices in the network and verifying that the traffic is being routed correctly.
- Introduce a fault in the network, such as disconnecting a link or router, and observe the effect on the BGP routing.
- Analyze the BGP convergence time and the stability of the routing after the fault is introduced.

References

[1] What is BGP? https://www.cloudflare.com/learning/security/glossary/what-is-bgp/. [Online; accessed 2023-04-15].

[2] Ivan Pepelnjak. Bgp tutorial: How the routing protocol works. *TechTarget*, jul 11 2019. [Online; accessed 2023-04-15].