IMT Atlantique
Département Informatique
Technopôle de Brest-Iroise - CS 83818
29238 Brest Cedex 3
URL: **www.imt-atlantique.fr**

UE PRIP
2021

Lab 4: Interdomain Routing, the Border Gateway Protocol

Edited: September 15, 2021
Version: 1.2

Report filled-in by:

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. Objectives

General objective

Understand how BGP protocol works and learn how to make a functional configuration of the protocol in a router.

Specific objectives

- Understand the functions of the BGP protocol on the Internet
- Analyse different key scenarios that will help you to understand BGP in a more practical way
- Troubleshoot common problems presented in BGP to better understand the bases of its configuration
- Understand some potential security constraints of the protocol

2. Pre-Lab

Make sure to prepare your lab with some readings:

- The introduction of this lab
- BGP (for instance in [11], [12],[3] or S. Bortzmayer's BGP course slides and videos available here [13].)
- BGP hijack attack [6], to know more: `http://www.slideshare.net/dyninc/bgp-prefix-hijack-defense-at-nanog-46`.

Go through Appendix B, you will need to work in the course's VM, it won't take you more than some minutes

3. Introduction

The Border Gateway Protocol (BGP) is a routing protocol used among autonomous systems (AS). An AS is a network or group of networks under a common administration and with common routing policies [1]. A common example of an AS is your Internet service provider (ISP). The objective of BGP is to exchange routing and reachability information for the Internet, which means that the ISPs will use BGP to exchange customer and ISP routes. The BGP peering between two routers is done over a TCP session on port 179.

BGP is considered as a path vector protocol. This type of network routing protocols are known for maintaining path information that gets updated dynamically, and also by detecting and discarding those updates that have looped through the network and returned to the same node. For path vector protocols, the destination network, the next hop, and the path to reach the destination are contained in each entry of the routing table.

When BGP is used between ASs is known as External BGP (EBGP). On the other hand, if an ISP is using BGP to exchange routes within an AS, then the protocol is referred to as Internal BGP (IBGP).

You can also see BGP as the routing protocol of the Internet, even though BGP presents several floss, it is a robust and scalable protocol, which explains its great success. By January 2017, the number of prefixes in the Internet BGP routing table reached 646000 [2] To achieve scalability at this level, BGP uses a set of attributes, to define routing policies and maintain a stable routing environment [1]. This attributes are:

- Weight (Cisco-dependent attribute)
- Local preference

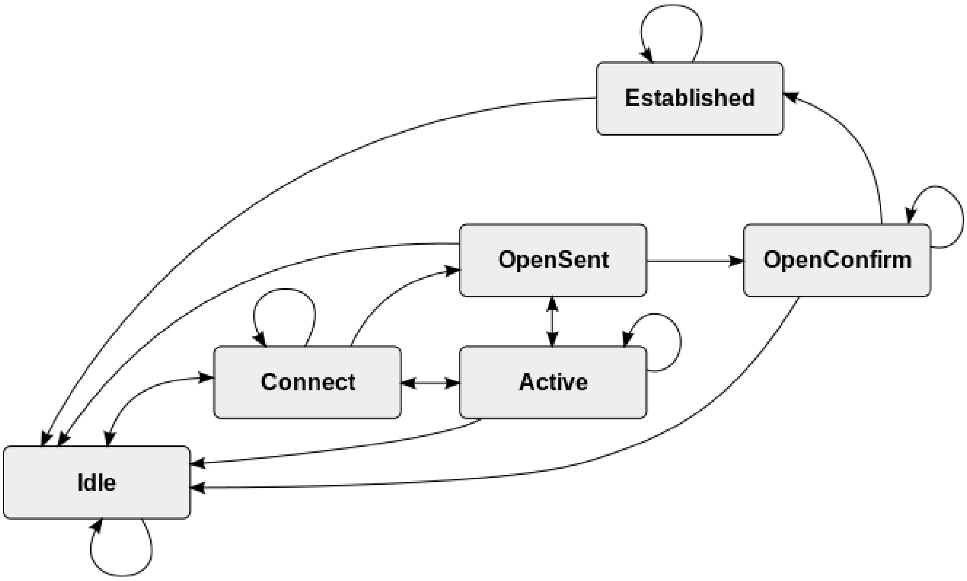


Figure 1: BGP finite state machine diagram. Source: [3]

- Multi-exit discriminator
- Origin
- AS_path
- Next hop
- Community

It is required to understand how these attributes affect the route selection when designing robust networks.

For establishing an EBGP connection between two routers, these must be physically connected and the **BGP configuration must be manually done on both of them**. Two routers sharing a BGP connection are known as BGP neighbors or BGP peers.

Figure 1 shows the BGP finite state machine (FSM). For detailed information on the transitions into the different states please visit [3] or the protocol specification [4].

There are also a set of four messages that are used in BGP. These messages are:

- Open
- Update
- Notification
- Keep-alive

To understand the way these messages are related to the states if the BGP FSM, please refer to [3]. You can also have access to the RFCs specifying BGP in the IETF webiste in particular see [4] and to its extensions, for instance for allowing IPv6 [5].

This lab uses Mininet to emulate the network topology being used, and also Quagga as routing daemon.

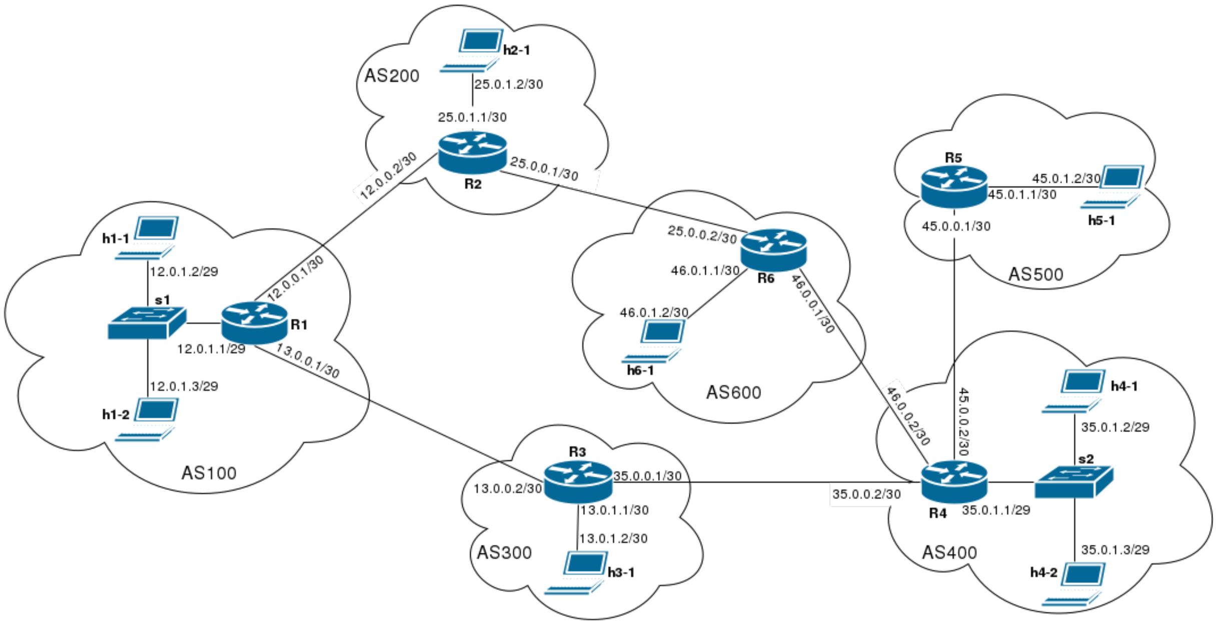


Figure 2: Topology connection to two ISPs

4. Hands on

4.1. EBGP

You are ready to start your own online digital distribution platform for games, called Zteam, you already solved all the financial details and now your are looking for a reliable Internet connection for your servers and headquarters. You find two ISPs, and as a skilled network engineer, you discuss with them and get to an arrangement for connecting your AS to their ASs.

You get all the information needed from the ISPs and get to a design for the connection, presented in Figure 2. We will assume that the Internet is represented by the 6 ASs in the figure. Your company is located in the AS100 and your service providers, as it's shown in the design, are located in AS200 and AS300.

4.1.1. First part: set up the network topology

1. If you haven't started the course's virtual machine please do so, go to the `net_labs` directory and update the content of the directory (`git pull`).

2. Go to the ebgp repository where the configuration files are located. Do it by executing the command: `cd ~/net_labs/bgp/ebgp/`

   All the routers in the topology, except by R1, are already configured to work with BGP. Your work will consist in configuring R1 and solving problems that are presented in some of the other routers. You will have to access the configuration files of the routers to solve these problems. In real life, you will only have access to your router and you will have to agree with your ISP so they properly configure their router or routers to connect with yours. For this exercise, you will have access to R1 and also to the other routers shown in the topology.

   You can have access to the configuration files of each router, they are present in directory `/net_labs/bgp/ebgp/conf/`.

   You will be interested in the `bgpd-R*.conf` files, where * is the number of the router to work with.

3. Run the emulation with the command: `sudo python run.py`

   Make sure you are in the right folder when running the emulation (`/net_labs/bgp/ebgp/`).

4.1.2. Second part: Router configuration

We are first going to analyse the configuration of router R6. This will help you to, later on, configure R1.

1. Open a terminal at R6. As usual, you need to do this from the Mininet CLI, by typing `xterm R6`
2. Once in R6's terminal, execute the following command to connect to the BGP daemon: `telnet localhost bgpd`.

Once there type:

```
Password: en
bgpd-R6>enable
Password: en
bgpd-R6#
```

You should be familiar with the different EXEC levels and configuration modes, from previous lab.

Indeed, the enable command will allow you to enter into the **privileged EXEC mode** of your router. You know you are in this mode when the symbol in front of **bgpd-R6** has changed from **>** to **#**.

There are two main steps for configuring BGP in a router, the first one is to **inject the network to be advertised** and the second one to **configure the connection to your neighbors**. You must also set an id for your router, but for maintaining a common configuration for this lab, the id of all the routers is already set. You can see the current running configuration for each router with the command:

`bgpd-R6# show running-config`

You can do this instead of opening the configuration files directly. This way can be faster and more effective, since you will be able to see also the changes you apply to the router by terminal.

Example

The configuration below is already done in R6 and it is provided only with demonstrational purposes, you don't need to do it again.

To configure the router we first enter to the **router configuration mode**:

```
bgpd-R6# configure terminal
bgpd-R6(config)# router bgp 600
bgpd-R6(config-router)#
```

Here, 600 is the AS number of your AS.

For injecting the network R6 will advertise we use the command:

```
bgpd-R6(config-router)# network 46.0.1.0/30
```

The connection to each neighbor is done individually, so in this case we need the following two commands:

```
bgpd-R6(config-router)# neighbor 25.0.0.1 remote-as 200
bgpd-R6(config-router)# neighbor 25.0.0.1 timers 1 5
bgpd-R6(config-router)# neighbor 46.0.0.2 remote-as 400
bgpd-R6(config-router)# neighbor 46.0.0.2 timers 1 5
```

You can exit any mode you enter with the command: `exit`

For these commands, the IP address is the one from the neighbors physically connected to your router, and the number after `remote-as` is the AS that your neighbors belong to. The timers option corresponds to the time values that are set to the Keepalive interval and the Holdtime respectively. The Keepalive indicates the time interval the peers use to send Keepalive messages between each other. On the other hand, the Holdtime indicates the time interval when a peer can be considered down and routes coming from it will be flushed. By default, the Keepalive timer is 60 seconds and the Holdtime is 180 seconds, but here to make it faster for this lab, these values are set to 1 and 5 respectively, as you can see above.

A good practice before configuring BGP, and a good trouble finder after it has been configured, is to ping the IP of your neighbor that you are directly connected to, in order to check first if the physical connection between you and your neighbor was correctly set up.

3. Configure BGP in R1 so it is connected to R2 and R3. You can get inspiration from the previous examples and from the already provided configuration files.

Question 1.
What commands did you use for your configuration?

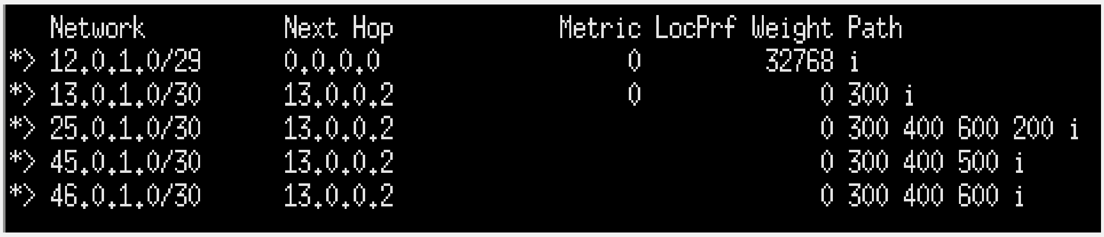
A useful way of checking that BGP is working as is supposed to is by analyzing the BGP routing tables in your router. Access this table by executing: `bgpd-R1# show ip bgp`

First go to the router configuration mode, once there use the following commands configure the router.

```
network 12.0.1.0/29
neighbor 12.0.0.2 remote-as 200
neighbor 12.0.0.2 timers 1 5
neighbor 13.0.0.2 remote-as 300
neighbor 13.0.0.2 timers 1 5
```

4. Give time, about 1 minute and 20 seconds, to BGP to converge and advertise all the routes in the topology. Check the BGP routing tables for R1, R2 and R3 to verify if the connection is correctly configured and verify it also by using the **ping** command.

Question 2.
What is the content of the BGP routing table of R1? Explain the function of each column and explain the information present in the table.



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      0.0.0.0                  0         32768 i
*> 13.0.1.0/30      13.0.0.2                 0             0 300 i
*> 25.0.1.0/30      13.0.0.2                               0 300 400 600 200 i
*> 45.0.1.0/30      13.0.0.2                               0 300 400 500 i
*> 46.0.1.0/30      13.0.0.2                               0 300 400 600 i
```

Question 3.
The routing table in R1 says that for going to the network in AS200 you have to go to AS300, which does not seem as the shortest path since you should be able to go directly to AS200. What is the cause of the connectivity problem with R2? Solve it and include in your answer the new content of R1's BGP routing table. Tip: check the configuration of R2.

The problem is caused because the connection was configured only in one way. This connection must be configured in both routers. Access the router configuration mode in R2 and add insert the following commands:

```
neighbor 12.0.0.1 remote-as 100
neighbor 12.0.0.1 timers 1 5
```

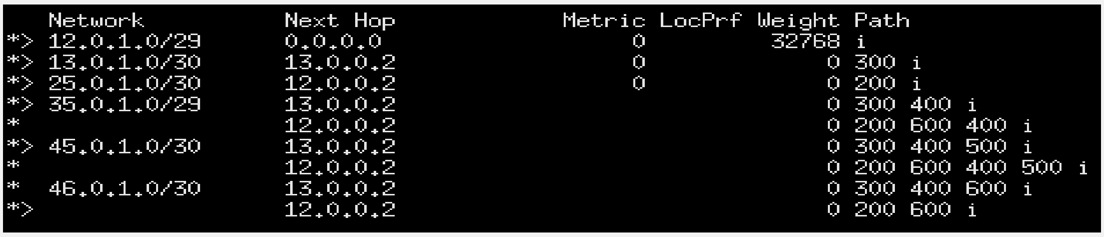
Question 4.
Take a closer look to the R1's BGP routing table, How does the path affect the route selection process?

Question 5.
Again, analysing the content of R1's BGP routing table, why can't you reach network 35.0.1.0/29 in AS400 but you can reach network 45.0.1.0/30 in AS500? Solve the problem and include in your answer a description of how you solved it and the new content of R1's BGP routing table. Tip: give a look to R2's configuration.

With the actual configuration, if more than one route available for a same prefix, the route with the shortest is selected. Actually, other BGP attributes can also play a role on route selection, as we shall see later on.

This problem is due to R4 not injecting the network, so it's not being advertised by BGP. This problem is solved by going to R4, and advertising the network with the following command (again, you must go into the router configuration mode to insert the command):

```
network 35.0.1.0/29
```



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      0.0.0.0                  0         32768 i
*> 13.0.1.0/30      13.0.0.2                 0             0 300 i
*> 25.0.1.0/30      12.0.0.2                 0             0 200 i
*> 35.0.1.0/29      13.0.0.2                               0 300 400 i
*                   12.0.0.2                               0 200 600 400 i
*> 45.0.1.0/30      13.0.0.2                               0 300 400 500 i
*                   12.0.0.2                               0 200 600 400 500 i
*  46.0.1.0/30      13.0.0.2                               0 300 400 600 i
*>                  12.0.0.2                               0 200 600 i
```

Make sure you have solved the problems with R2 and the network in AS400 before continuing. If you were not able to do it, refer to the section "Router configuration" in this document and check the running configuration of the routers presenting the problems.

5. Open Wireshark at R1 (from Mininet CLI `R1 wireshark &`). Select R1-eth0 and start capturing packets.

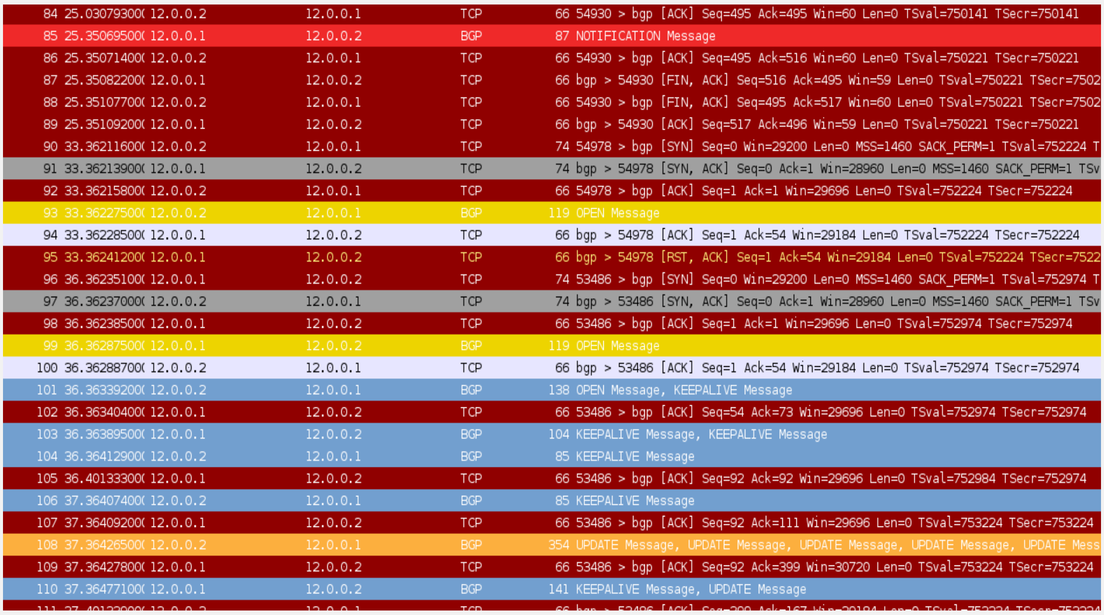
6. Connect to the bgpd service in R1 and reset the session between R1 and R2 by running the command

`bgpd-R1# clear ip bgp 12.0.0.2`. By reseting the session you are triggering an event that uses all the different message types in BGP.

Question 6.

Analyse the capture. Identify all the types of messages used in a BGP session. What message type is used by BGP to let the router know the session is over? Identify the sender and receiver of this message. What is the sequence of messages? It might be useful to use some visualization filters.

The student should provide an image where the four different type of messages are shown.

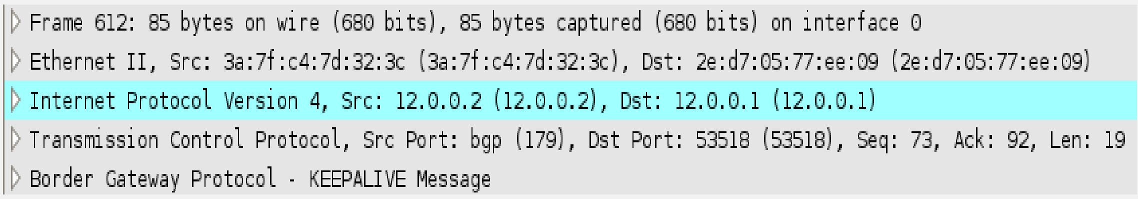


```
 84 25.03079300( 12.0.0.2    12.0.0.1    TCP    66 54930 > bgp [ACK] Seq=495 Ack=495 Win=60 Len=0 TSval=750141 TSecr=750141
 85 25.35069500( 12.0.0.1    12.0.0.2    BGP    87 NOTIFICATION Message
 86 25.35071400( 12.0.0.2    12.0.0.1    TCP    66 54930 > bgp [ACK] Seq=495 Ack=516 Win=60 Len=0 TSval=750221 TSecr=750221
 87 25.35082200( 12.0.0.1    12.0.0.2    TCP    66 bgp > 54930 [FIN, ACK] Seq=516 Ack=495 Win=59 Len=0 TSval=750221 TSecr=7502
 88 25.35107700( 12.0.0.2    12.0.0.1    TCP    66 54930 > bgp [FIN, ACK] Seq=495 Ack=517 Win=60 Len=0 TSval=750221 TSecr=7502
 89 25.35109200( 12.0.0.1    12.0.0.2    TCP    66 bgp > 54930 [ACK] Seq=517 Ack=496 Win=59 Len=0 TSval=750221 TSecr=750221
 90 33.36211600( 12.0.0.2    12.0.0.1    TCP    74 54978 > bgp [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=752224 T
 91 33.36213900( 12.0.0.1    12.0.0.2    TCP    74 bgp > 54978 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv
 92 33.36215800( 12.0.0.2    12.0.0.1    TCP    66 54978 > bgp [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=752224 TSecr=752224
 93 33.36227500( 12.0.0.2    12.0.0.1    BGP   119 OPEN Message
 94 33.36228500( 12.0.0.1    12.0.0.2    TCP    66 bgp > 54978 [ACK] Seq=1 Ack=54 Win=29184 Len=0 TSval=752224 TSecr=752224
 95 33.36241200( 12.0.0.1    12.0.0.2    TCP    66 bgp > 54978 [RST, ACK] Seq=1 Ack=54 Win=29184 Len=0 TSval=752224 TSecr=7522
 96 36.36235100( 12.0.0.1    12.0.0.2    TCP    74 53486 > bgp [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=752974 T
 97 36.36237000( 12.0.0.2    12.0.0.1    TCP    74 bgp > 53486 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSv
 98 36.36238500( 12.0.0.1    12.0.0.2    TCP    66 53486 > bgp [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=752974 TSecr=752974
 99 36.36287500( 12.0.0.1    12.0.0.2    BGP   119 OPEN Message
100 36.36288700( 12.0.0.2    12.0.0.1    TCP    66 bgp > 53486 [ACK] Seq=1 Ack=54 Win=29184 Len=0 TSval=752974 TSecr=752974
101 36.36339200( 12.0.0.2    12.0.0.1    BGP   138 OPEN Message, KEEPALIVE Message
102 36.36340400( 12.0.0.1    12.0.0.2    TCP    66 53486 > bgp [ACK] Seq=54 Ack=73 Win=29696 Len=0 TSval=752974 TSecr=752974
103 36.36389500( 12.0.0.1    12.0.0.2    BGP   104 KEEPALIVE Message, KEEPALIVE Message
104 36.36412900( 12.0.0.2    12.0.0.1    BGP    85 KEEPALIVE Message
105 36.40133300( 12.0.0.1    12.0.0.2    TCP    66 53486 > bgp [ACK] Seq=92 Ack=92 Win=29696 Len=0 TSval=752984 TSecr=752974
106 37.36407400( 12.0.0.2    12.0.0.1    BGP    85 KEEPALIVE Message
107 37.36409200( 12.0.0.1    12.0.0.2    TCP    66 53486 > bgp [ACK] Seq=92 Ack=111 Win=29696 Len=0 TSval=753224 TSecr=753224
108 37.36426500( 12.0.0.2    12.0.0.1    BGP   354 UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Message, UPDATE Mess
109 37.36427800( 12.0.0.1    12.0.0.2    TCP    66 53486 > bgp [ACK] Seq=92 Ack=399 Win=30720 Len=0 TSval=753224 TSecr=753224
110 37.36477100( 12.0.0.1    12.0.0.2    BGP   141 KEEPALIVE Message, UPDATE Message
111 37.40133900( 12.0.0.2    12.0.0.1    TCP    66 bgp > 53486 [ACK] Seq=399 Ack=167 Win=29184 Len=0 TSval=753224 TSecr=753224
```

Question 7.

What is the protocol stack being used for the BGP session? List the protocols and layers. What layer does BGP belong to?

The students can select any message type from the four available to check the protocol stack. The protocol stack shown in the image. We can say that BGP belongs to the application layer, sin it runs over TCP. However, it is important to know that even if BGP is known to belong to this layer, its function is more related to the network layer, thanks to its capability to manage routing information.



```
▷ Frame 612: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
▷ Ethernet II, Src: 3a:7f:c4:7d:32:3c (3a:7f:c4:7d:32:3c), Dst: 2e:d7:05:77:ee:09 (2e:d7:05:77:ee:09)
▷ Internet Protocol Version 4, Src: 12.0.0.2 (12.0.0.2), Dst: 12.0.0.1 (12.0.0.1)
▷ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 53518 (53518), Seq: 73, Ack: 92, Len: 19
▷ Border Gateway Protocol - KEEPALIVE Message
```

Question 8.

Does the command used for question 5 restart the TCP and BGP sessions? Or only restarts the BGP session? Explain your answer.

> *The command used in question 5 restarts both, the BGP and the TCP session. This can be seen in Wireshark. Right after the notification message, which indicates that there was an error in the BGP session, the packets to end the TCP session are exchange between the peers. The might necessary for the students to review the TCP connection termination process.*

7. Let's suppose that it is very important for your company to connect to the host 45.0.1.2/30. Right now that connection is being held by going through R3, but you noticed that this route presents delay problems that affect your connection with AS500. As a BGP expert you know that a way to solve this problem is by changing your route preferences with the attribute **weight**.

You can change the weight attribute with the command:

```
bgpd-R1(config-router)# neighbor <ip-address> weight <0-65535>
```

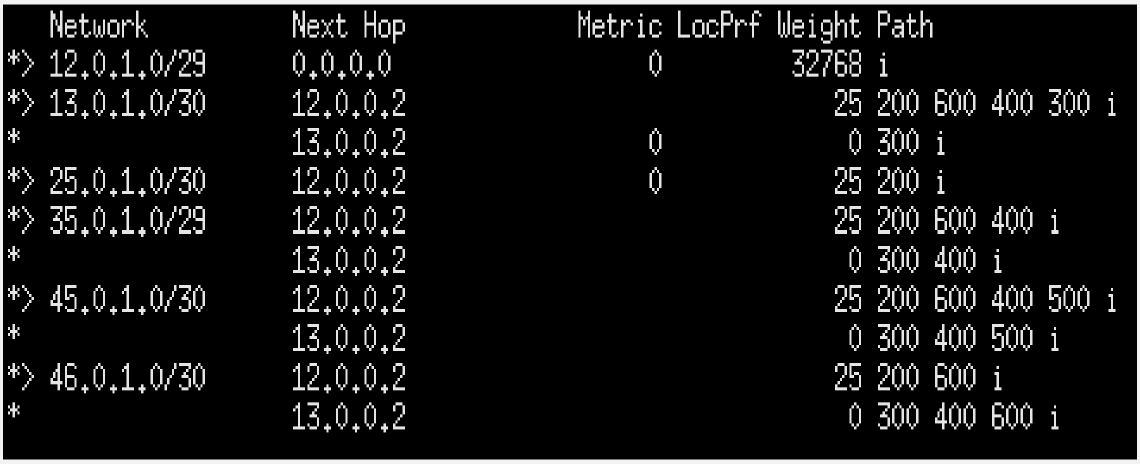
If you want to prefer a route over another one, you just need to set a higher weight to it. The way BGP uses the weight and, in general, selects the best path is linked to the attributes that were described in the introduction. The algorithm for best path selection performs and ordered check of the different attributes, in order to consider one path more or less important than other. For the specific case of the weight, this attribute is the first one on the BGP's checking list, which makes it more important that the AS_PATH attribute, which ranks fourth. You can see the complete list and read more about the BGP best path selection algorithm in the site: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html.

Please note that weight attribute is a Cisco-defined attribute, and is a local attribute (it is not advertised to neighboring routers). In this lab we are going to use this attribute, which is as well available in quagga. If working outside Cisco equipments, we could use the local preference attribute, for instance, which requires a bit more sophisticated configuration.

8. Change the weight attribute in R1 so your new preferred route to go out to the Internet goes through R2 and not R3, don't forget to use the command `clear ip bgp *` after changing the attribute so you can see the updated table.

Question 9.

Analysing the content of R1's BGP routing table: what is its content now? What are the consequences of changing this attribute? Can you think of disadvantages of using this method to select a route over another one?



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      0.0.0.0                  0         32768 i
*> 13.0.1.0/30      12.0.0.2                               25 200 600 400 300 i
*                   13.0.0.2                 0              0 300 i
*> 25.0.1.0/30      12.0.0.2                 0             25 200 i
*> 35.0.1.0/29      12.0.0.2                               25 200 600 400 i
*                   13.0.0.2                                0 300 400 i
*> 45.0.1.0/30      12.0.0.2                               25 200 600 400 500 i
*                   13.0.0.2                                0 300 400 500 i
*> 46.0.1.0/30      12.0.0.2                               25 200 600 i
*                   13.0.0.2                                0 300 400 600 i
```

One of the disadvantages is that there is no easy way of doing a feasible change of weights. As it can be seen in the image above, the simplest command for changing the weight changed not only the route for the destination we wanted, but also for all the other destinations. Cases like this one can be managed with a more advanced and specific command, using lists for examples.

A second disadvantage is that the flexibility of the route decision process is affected. Setting a higher weight to a route conditions the preference for this one.

9. Set the weight you have changed back to zero. After doing it, don' forget to use the command `clear ip bgp *` to see the changes applied.

A fellow network engineer studied what you have done with your AS and told you that you must apply the following configuration to your router R1:

```
bgpd-R1(config)# ip as-path access-list 1 permit ^$
bgpd-R1(config)#router bgp <AS>
bgpd-R1(config-router)# neighbor 12.0.0.2 filter-list 1 out
bgpd-R1(config-router)# neighbor 13.0.0.2 filter-list 1 out
```

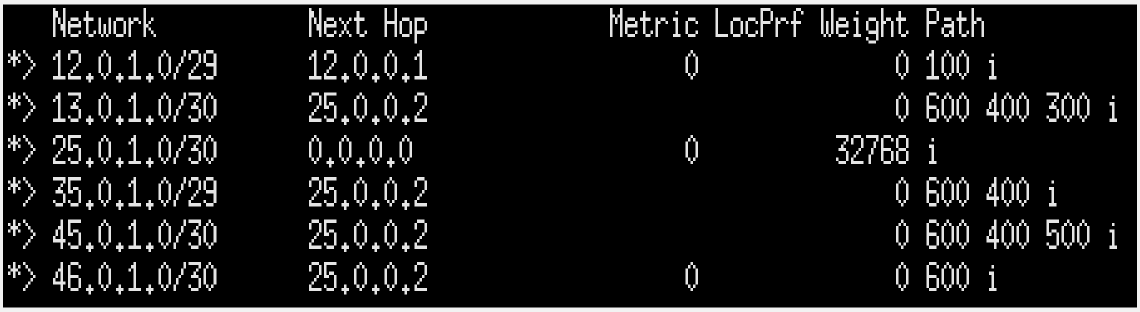
He says that with this, you will avoid AS100 to become a transit AS. Indeed, the `^$` regular expression indicates origination from this AS. We then specify for both neighbors that we announce only prefixes matching access list 1, i.e. locally originated prefixes.

10. Check the content of R2's and R3's BGP routing tables

11. After applying the configuration above, use the command `clear ip bgp *`, give some time BGP to converge, and then check the routing table of R1, R2 and R3.

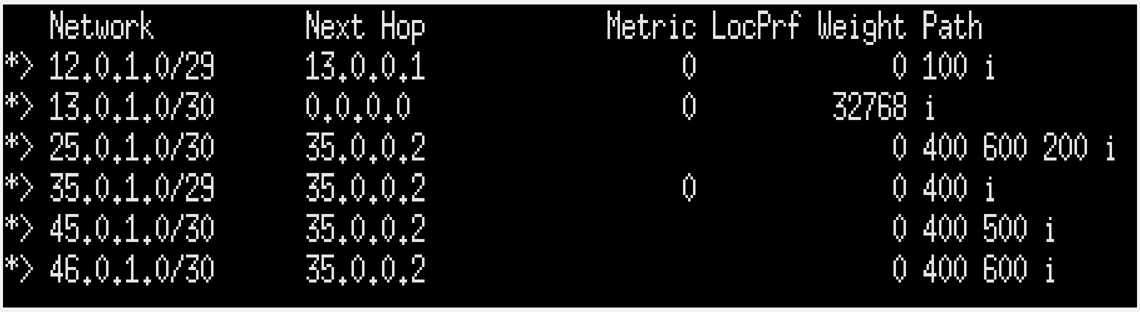
Question 10.
What changes do you notice regarding the routes in the table? How is this related to a transit AS?

The difference now is that R2 won't use R1 to go to R3's network, and R3 won't use R1 to go to R2's network. This can be seen in the images below. The first image shows the BGP routing table for R2, and it can be seen that the only path now for reaching networks different from the one in AS100 is by going through AS600.



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      12.0.0.1                 0             0 100 i
*> 13.0.1.0/30      25.0.0.2                               0 600 400 300 i
*> 25.0.1.0/30      0.0.0.0                  0         32768 i
*> 35.0.1.0/29      25.0.0.2                               0 600 400 i
*> 45.0.1.0/30      25.0.0.2                               0 600 400 500 i
*> 46.0.1.0/30      25.0.0.2                 0             0 600 i
```

The same can be seen for in R3's BGP routing table.



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      13.0.0.1                 0             0 100 i
*> 13.0.1.0/30      0.0.0.0                  0         32768 i
*> 25.0.1.0/30      35.0.0.2                               0 400 600 200 i
*> 35.0.1.0/29      35.0.0.2                 0             0 400 i
*> 45.0.1.0/30      35.0.0.2                               0 400 500 i
*> 46.0.1.0/30      35.0.0.2                               0 400 600 i
```

Ref: https://networklessons.com/bgp/bgp-prevent-transit-as/

We will now explore the protocol's behaviour when an interface goes down. You will be interested in measuring the convergence time of the protocol and observing the exchanged messages.

12. Open Wireshark in router R2 and listen to interface connecting to R6 (R2-eth0).

13. In R1, put the interface eth0 down with the next procedure:

In R1's xterm:

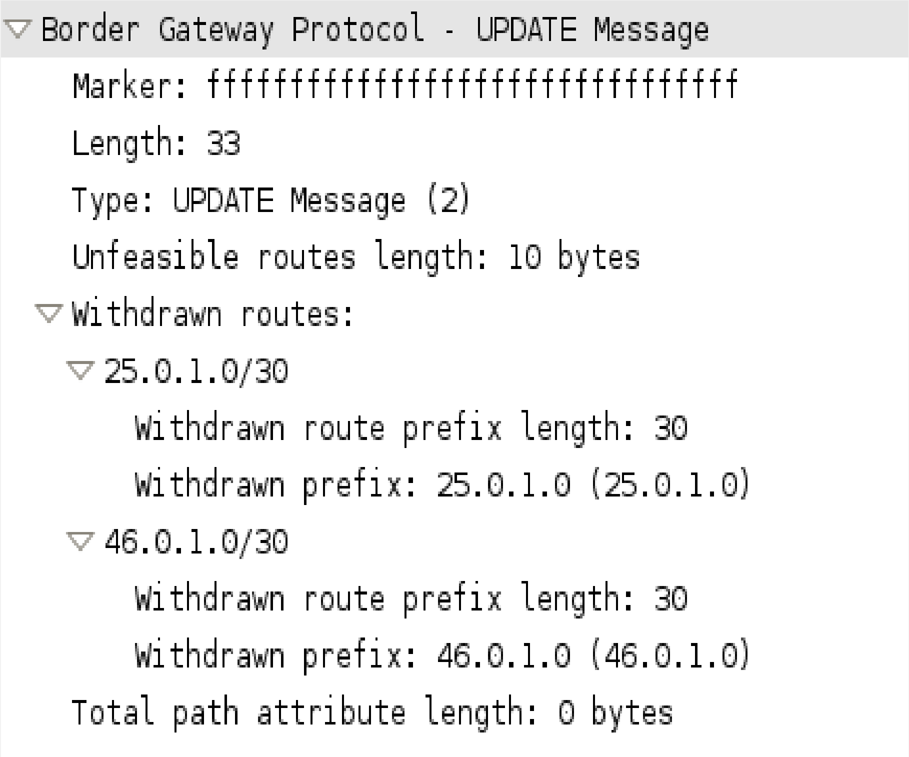
```
telnet localhost zebra

Password: en
R1> enable
Password: en
R1# configure terminal
R1(config)# interface R1-eth0
R1(config-if)# shutdown
```

You can verify the interface is down by checking R1's BGP table. If the tables don't update right after using **shutdown** and **no shutdown**, wait for BGP to converge.

Question 11.
Check Wireshark when you shutdown the interface in R2. What is the procedure by which routers learn that a link is down? Which BGP message indicates that there is a link problem? Check the content of this message in the BGP layer, what field is used for announcing the change in the routes? Illustrate this with one message found on the capture.

The routers learn that a link is down, and in general about changes in routes, by sending messages containing the corresponding information that changed. The type of message used for this is the Update message. The field used for carrying this information is the Withdrawn routes field, which can be seen by expanding the details of the BGP section of the packet in Wireshark.



```
▽ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 33
    Type: UPDATE Message (2)
    Unfeasible routes length: 10 bytes
  ▽ Withdrawn routes:
    ▽ 25.0.1.0/30
        Withdrawn route prefix length: 30
        Withdrawn prefix: 25.0.1.0 (25.0.1.0)
    ▽ 46.0.1.0/30
        Withdrawn route prefix length: 30
        Withdrawn prefix: 46.0.1.0 (46.0.1.0)
    Total path attribute length: 0 bytes
```

14. After putting the interface down, put it back up by using now `no shutdown` instead of `shutdown`. You can use Wireshark to have an idea of the convergence time by measuring the times of some packages.

Question 12.
Measure the approximate convergence time of the BGP routing table in R1 when setting R1-eth0 back up. You can open several xterm windows for a router. Is this convergence time lower or higher than the initial 1 minute 20 seconds? Why?

When putting the interface R1-eth0 down, you can see the change in R1's BGP routing table takes place almost immediately. When putting it back up it usually takes more time, usually between 20 and 25 seconds. The initial discovery process of the network takes longer due to the routing tables being empty, which implies building the routing tables from scratch. If there is only some information that must be updated, as it is the case for a link down, this process will take less time.

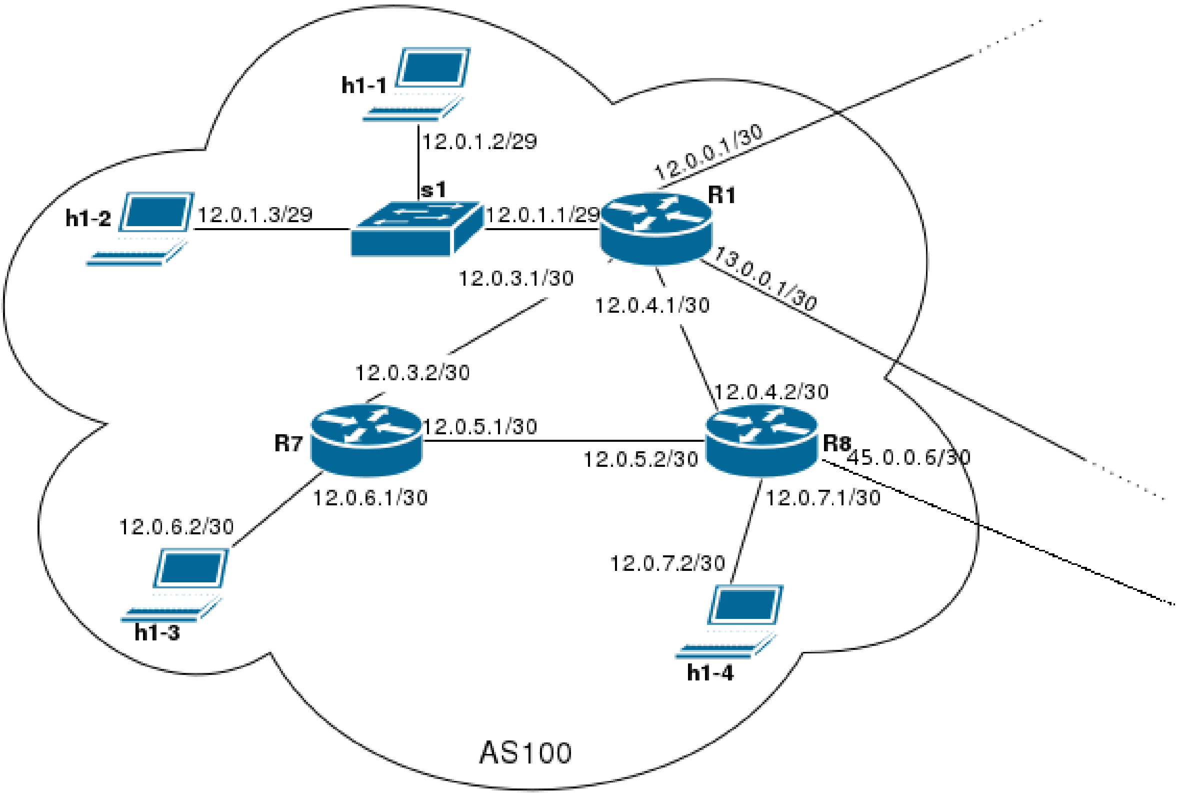


Figure 3: Topology of AS100 for IBGP configuration

4.2. IBGP

Your digital distribution company starts growing, and you think it's a good time to expand your company and hence, to expand your network. Two small teams join the Zteam family, and you decide to place two new routers to your AS in order to have a more unified and fast network among the teams. Figure 3 shows the network topology inside your AS. We have decided also to add a connection between R8 and AS500 (R5).

As it was explained in the introduction, IBGP is the same BGP protocol when is being used by several routers inside an AS. In order to run IBGP, these routers need to know the routing information inside the AS, this is, the routes and reachability information inside the AS. This is due to a characteristic in IBGP that won't assign the right next hop parameter inside the AS. This, as useless as it might sound, it's made this way for a reason, but we won't cover that in detail in this session.

There are two ways to get these routes and reachability information for BGP, you can fill the routing table of the router manually, or you can use an IGP that fills this table for you. For this practice we are using OSPF as our IGP to fill automatically the routing tables of the routers inside the AS. With this information IBGP will be able to fill the BGP routing table in each router.

4.2.1. First part: Running the emulation

1. Exit the previous emulation with the command: `mininet>exit`

2. Go to the folder where the IBGP emulation files are located and run the emulation as follows:

```
cd ~/net_labs/bgp/ibgp/
sudo python run.py
```

Wait 2 minutes and 10 seconds for BGP to converge.

4.2.2. Second part: configuration

Example

There are three steps when configuring IBGP in one of your routers:

i) You need to define your network, just as it's done in EBGP. The only difference is that your remote-as will be the same where the router is located.

```
R1(config-router)# neighbor <x.x.x.x> remote-as <ASN>
```

ii) You need to define your update source. It's recommended to do the configuration of IBGP routers using their loopback interfaces, in this way if a physical interfaces fail your IBGP session won't be terminated.

```
R1(config-router)# neighbor <x.x.x.x> update-source lo
```

The loopback interfaces in our working topology are:

R1: 1.1.1.1

R7: 7.7.7.7

R8: 8.8.8.8

iii) Each time you configure a new connection you need to specify that each router is going to be the next hop to reach a destination. This applies for each router in your AS.

```
R1(config-router)# neighbor <x.x.x.x> next-hop-self
```

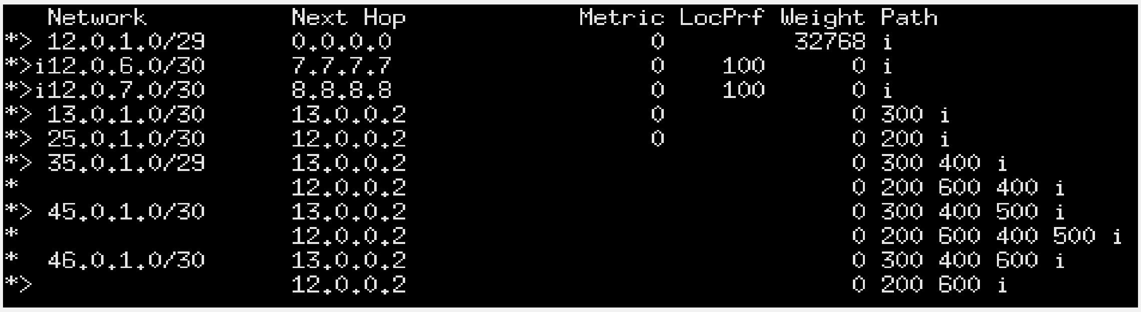
1. Configure IBGP for router R1. IBGP is already configured in routers R7 and R8 in the configuration files. Use the three commands specified above for each neighbor you want to connect to. It will take to BGP from 20 to 25 seconds to update the tables in the routers after you have applied the configuration.

Question 1.

Specify the commands you used for configuring IBGP in R1. What is the content of the BGP routing table of R1?

As it is specified in the example, you need to go first into the router configuration mode and then insert the following commands:

```
neighbor 7.7.7.7 remote-as 100
neighbor 7.7.7.7 update-source lo
neighbor 7.7.7.7 next-hop-self
neighbor 8.8.8.8 remote-as 100
neighbor 8.8.8.8 update-source lo
neighbor 8.8.8.8 next-hop-self
```



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.1.0/29      0.0.0.0                  0         32768 i
*>i12.0.6.0/30      7.7.7.7                  0    100      0 i
*>i12.0.7.0/30      8.8.8.8                  0    100      0 i
*> 13.0.1.0/30      13.0.0.2                 0             0 300 i
*> 25.0.1.0/30      12.0.0.2                 0             0 200 i
*> 35.0.1.0/29      13.0.0.2                               0 300 400 i
*                   12.0.0.2                               0 200 600 400 i
*> 45.0.1.0/30      13.0.0.2                               0 300 400 500 i
*                   12.0.0.2                               0 200 600 400 500 i
*  46.0.1.0/30      13.0.0.2                               0 300 400 600 i
*>                  12.0.0.2                               0 200 600 i
```

2. Check the routing table in router R7.

Question 2.
What makes possible that R7 has in its routing table the reachability information to get to all the networks in AS100?

This is possible because of an IGP that fills the routing table in R7. In this case, the IGP being used is OSPF.

3. Take a look at R2's BGP routing table. Check that the internal routes from AS100 are now in R2's BGP table. Indeed, the internal routes are advertised just as external routes in BGP. Once one of these networks is injected and advertised in BGP, it will be known inside the AS and also by routers outside the AS.

Question 3.
Which router(s) is (are) in charge of advertising internal routes to the outside of AS100?

The router in charge of advertising routes from the inside to the outside of AS100 is the border router R1.

4. Take a look at R7's BGP routing table. Verify that its routing table has all the prefixes from the outside of AS100. Just like in the previous step, there are no limitations regarding external routes being advertise inside an AS. The external routes will be known by R1 thanks to EBGP, and once they are in the its BGP routing table, they will be advertised to the routers inside the AS.

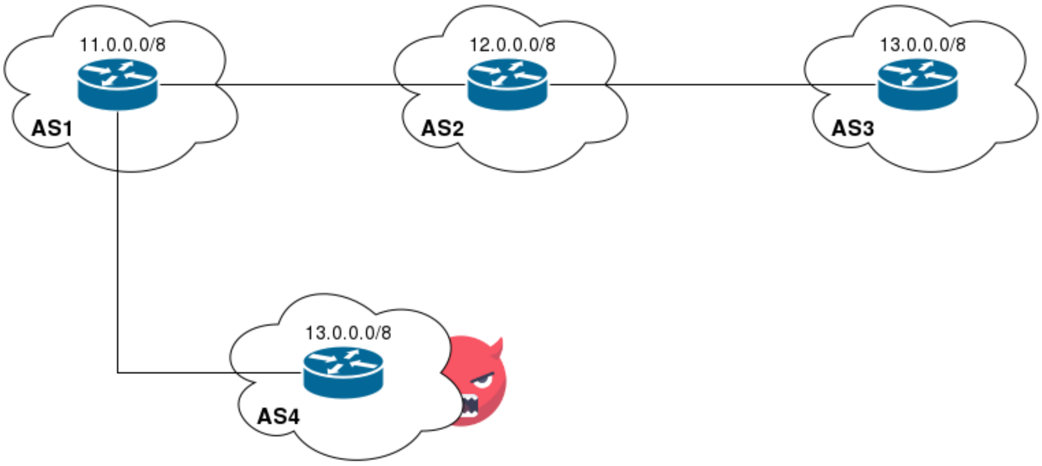


Figure 4: Topology for BGP Path Hijacking Attack Demo

Question 4.
Which router is in charge of advertising external routes to the inside of AS100?

> *The router in charge of advertising routes from the outside to the inside of AS100 is the border router R1.*

4.3. BGP Path Hijacking Attack

In this exercise, you are going to recreate a "BGP path hijacking attack" [6] inside Mininet.

Recall that the Internet predominantly consists of interconnected Autonomous Systems (ASs) that exchange routing information with each other using a common protocol called the Border Gateway Protocol.

The goal of this exercise is to safely demonstrate a specific attack that was possible using BGP, in which an malicious AS falsely advertises a shorter path to reach a prefix P, which causes other ASs to route traffic destined to the prefix P through that shortest path, and thus through the malicious AS.

We will emulate the network topology shown in Figure 4. There are four ASs: AS1, AS2, AS3 and AS4 (rogue, the malicious one). The BGP sessions are established as follows:

- R1 peers with R2 and R4.
- R2 peers with R1 and R3.

4.3.1. First part: running the emulation

1. Exit the previous emulation with the command:

```
mininet> exit
```

2. Go to the folder where the bgpAttack emulation files are located and run the emulation as follows:

```
cd ~/net_labs/bgp/bgpAttack
sudo python run.py
```

This sets up the network topology and starts BGP, with normal functioning (there's no attack in place yet).

4.3.2. Second part: briefly analyse regular functioning

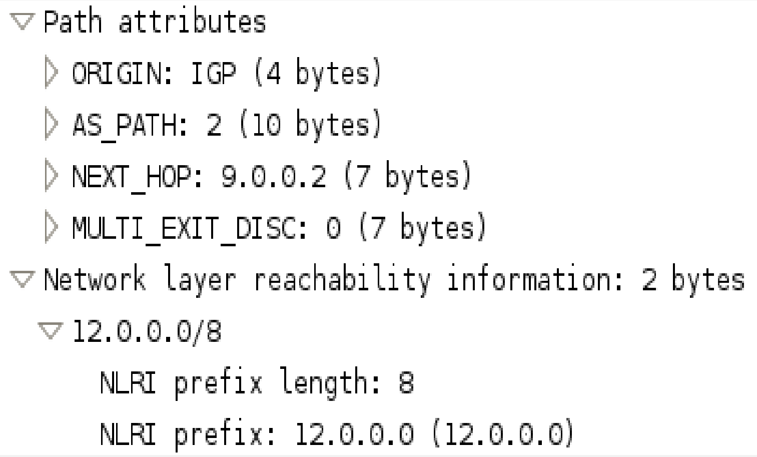
1. Execute Wireshark on R1 and start a new capture in R1-eth4. After you are capturing packets

2. Enter to the bgpd service in R1's xterm, execute the command `clear bgp *` and go to Wireshark. Once in Wireshark look for the update messages. Tip: you can use visualization filter bgp.type==2 to see only BGP update messages.

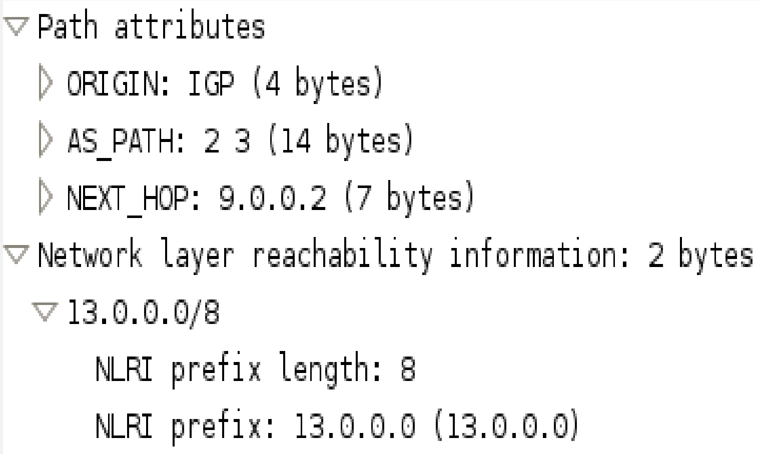
Question 1.

Check the content of the Update messages to see what prefixes are being advertised. How many prefixes are being advertised? How many advertised prefixes are being received and from whom? How many advertised prefixes are being sent and from whom?

Three networks are being advertised in total: 11.0.0.0, 12.0.0.0, and 13.0.0.0.
The next two images show the information for the prefixes 12.0.0.0 and 13.0.0.0. If you check the source and destination of these Update messages, you will see that these prefixes have as destination address 9.0.0.1 and as source address 9.0.0.2, which meand that in this context these prefixes are being received.

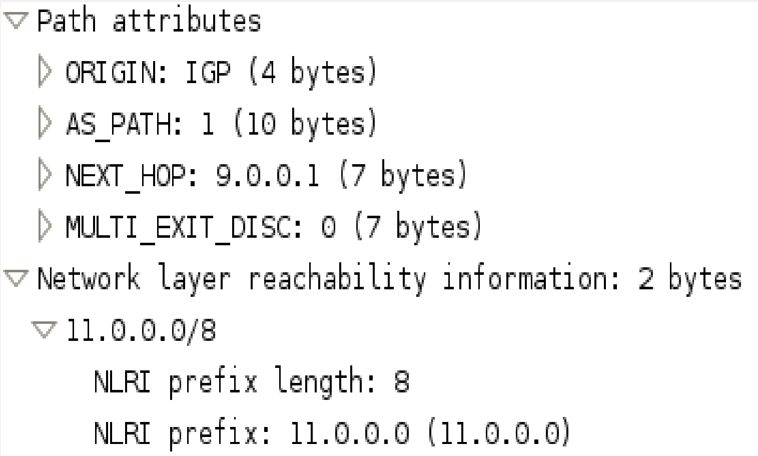


```
▽ Path attributes
  ▷ ORIGIN: IGP (4 bytes)
  ▷ AS_PATH: 2 (10 bytes)
  ▷ NEXT_HOP: 9.0.0.2 (7 bytes)
  ▷ MULTI_EXIT_DISC: 0 (7 bytes)
▽ Network layer reachability information: 2 bytes
  ▽ 12.0.0.0/8
      NLRI prefix length: 8
      NLRI prefix: 12.0.0.0 (12.0.0.0)
```



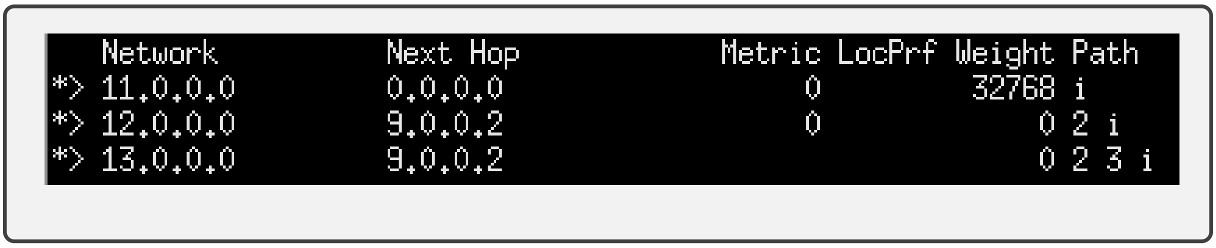
```
▽ Path attributes
  ▷ ORIGIN: IGP (4 bytes)
  ▷ AS_PATH: 2 3 (14 bytes)
  ▷ NEXT_HOP: 9.0.0.2 (7 bytes)
▽ Network layer reachability information: 2 bytes
  ▽ 13.0.0.0/8
      NLRI prefix length: 8
      NLRI prefix: 13.0.0.0 (13.0.0.0)
```

This image shows the information of the prefix 11.0.0.0. Again, by checking the source and destination address you can see thet in this context this prefix is being sent, as the source address is 9.0.0.1 and the destination 9.0.0.2.



```
▽ Path attributes
  ▷ ORIGIN: IGP (4 bytes)
  ▷ AS_PATH: 1 (10 bytes)
  ▷ NEXT_HOP: 9.0.0.1 (7 bytes)
  ▷ MULTI_EXIT_DISC: 0 (7 bytes)
▽ Network layer reachability information: 2 bytes
  ▽ 11.0.0.0/8
      NLRI prefix length: 8
      NLRI prefix: 11.0.0.0 (11.0.0.0)
```

Question 2.
Check the BGP routing table in router 1. What are the prefixes in the table? In particular, which is the path chosen to reach destination 13.0.0.0/8?



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         9.0.0.2                  0             0 2 i
*> 13.0.0.0         9.0.0.2                                0 2 3 i
```

Now, you will visit a default web server that our first script (`run.py`) has started in AS3 and verify that you can reach it from host h1-1 connected to AS1.

3. Open a new xterm for R1 without closing the one you have already opened. In this new xterm execute the file website.sh:

   `./website.sh`

   You should see the message below appearing several times:

   Fri Aug 11 06:14:52 PDT 2017 -- <h1>Default web server</h1>

4.3.3. Third part: start the attack

1. Change the interface where you are capturing packets in Wireshark from R1-eth4 to R1-eth5.

2. Start a new xterm for R1 without closing the previous two you already opened. In this new terminal you will execute file start_rogue.sh with the command presented below. This will start the attack. The rogue AS will peer with AS1 and advertise a route to 13.0.0.0/8 using a shorter path (i.e., a direct path from AS1 to AS4). Thus, AS1 will choose this shorter path by default.

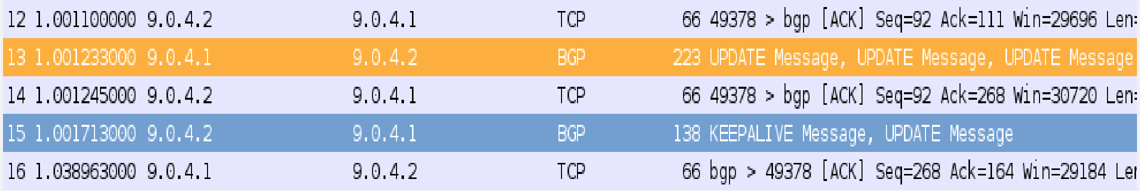
   `./start_rogue.sh`

   You should see a new message in the xterm where you are visiting the webserver:

   Fri Aug 11 06:16:33 PDT 2017 -- <h1>*** Attacker web server ***</h1>

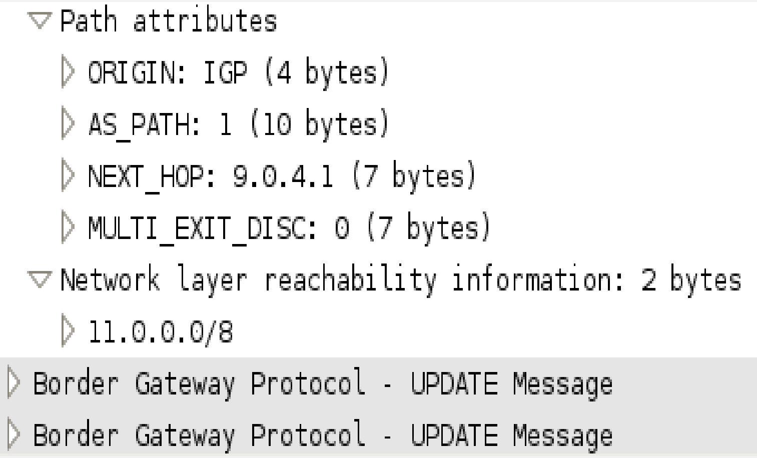
Question 3.
Check the content of the Update messages. Find the one(s)related to the attack. Which information do they contain? Who is sending the concerned Update message?

It can be seen that from AS1 to AS4 are being sent three Update messages, which correspond to the prefixes 11.0.0.0, 12.0.0.0, and 13.0.0.0 that were advertise to AS1's router before. On the other hand, there is one Update message coming from AS4, which is advertising the fake 13.0.0.0 prefix. So there is a total of 4 prefixes being advertised, and two of them in this case happen to be the same due to the attack that is taking place. The Update messages can be seen in the figure below.

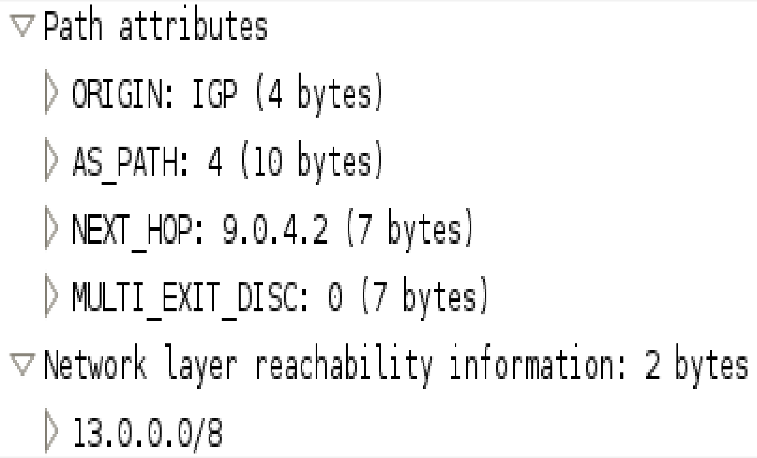


```
12 1.001100000 9.0.4.2    9.0.4.1    TCP    66 49378 > bgp [ACK] Seq=92 Ack=111 Win=29696 Len=
13 1.001233000 9.0.4.1    9.0.4.2    BGP    223 UPDATE Message, UPDATE Message, UPDATE Message
14 1.001245000 9.0.4.2    9.0.4.1    TCP    66 49378 > bgp [ACK] Seq=92 Ack=268 Win=30720 Len=
15 1.001713000 9.0.4.2    9.0.4.1    BGP    138 KEEPALIVE Message, UPDATE Message
16 1.038963000 9.0.4.1    9.0.4.2    TCP    66 bgp > 49378 [ACK] Seq=268 Ack=164 Win=29184 Len
```

If you inspect the packets you will confirm that the three previous prefixes are being advertised and also that the router in AS4 is advertising a prefix that has been advertised already. This information can be seen in the images below.



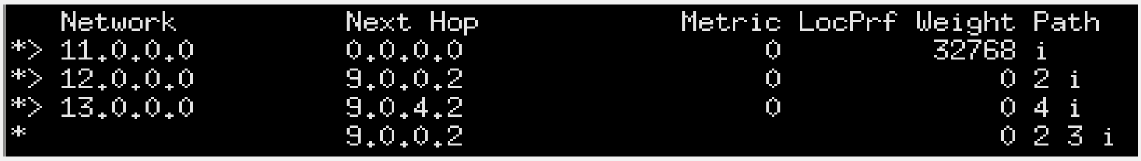
```
 ▽ Path attributes
   ▷ ORIGIN: IGP (4 bytes)
   ▷ AS_PATH: 1 (10 bytes)
   ▷ NEXT_HOP: 9.0.4.1 (7 bytes)
   ▷ MULTI_EXIT_DISC: 0 (7 bytes)
 ▽ Network layer reachability information: 2 bytes
   ▷ 11.0.0.0/8
▷ Border Gateway Protocol - UPDATE Message
▷ Border Gateway Protocol - UPDATE Message
```



```
▽ Path attributes
  ▷ ORIGIN: IGP (4 bytes)
  ▷ AS_PATH: 4 (10 bytes)
  ▷ NEXT_HOP: 9.0.4.2 (7 bytes)
  ▷ MULTI_EXIT_DISC: 0 (7 bytes)
▽ Network layer reachability information: 2 bytes
  ▷ 13.0.0.0/8
```

Question 4.

Check again the BGP routing table for R1. What is its content? In particular, which is the route selected for prefix 13.0.0.0/8?



```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         9.0.0.2                  0             0 2 i
*> 13.0.0.0         9.0.4.2                  0             0 4 i
*                   9.0.0.2                                0 2 3 i
```

3. You can stop the attack from the rogue AS by executing the file `stop_rogue.sh`. You can do this in the same xterm where you started the attack executing the file `start_rogue.sh`.

   ```
   ./stop_rogue.sh
   ```

   After stopping the attack the BGP routing table should return to its original, with the legit routes to AS2 and AS3.

4. Check the routing table and make sure this is the case.

Question 5.
With what objective can be used a path hijacking attack with BGP as it was presented in this lab?

The traffic hijacked can be redirected to a specific site to perform a DoS attack.
Another scenario would be one where the hijacker redirects the traffic to servers that plays the role of specific services, stealing private information from the users (Phishing)

5. Conclusion

Question 1.
What differences can you find between OSPF and BGP, regarding that they are respectively a link state protocol and a path vector protocol? What are the differences between these two approaches?

One difference between BGP and OSPF is their scope. OSPF is used as routing protocol inside a network, that's is why is an IGP. On the other hand, BGP is used as a routing protocol among networks, and is known as an EGP.
Another difference is the complexity of both protocols. BGP is used to manage routes in a bigger scale than OSPF, that is why it has to make it in a simple way. BGP works at the level of the whole Internet, reason why building a complete map of the network, as OSPF do, is a very difficult task. On the other hand, OSPF is able to do this since it works only in one domain, being with this the number of devices much less than for BGP.
The main difference between the two approaches is the way they perform the recognition of different paths. Link State Protocols create a whole map of the network that will be stored on each router, then each router will use a shortest path first algorithm to decide what is the shortest path to each reachable destination. Path Vector Protocols rely on the information provided to them about certain raoutes to reach a desination, which means that it's not necessary for these protocols to know a whole map of the network. The search of a valid path will depend on the path being loop free or not.

Question 2.
Why is it necessary to have Interior Gateway Protocols (IGP) and Exterior Gateway Protocols (EGP)?

BGP fits in the category of Exterior Gateway Protocols (EGP), which means that it works as routing protocol among ASs. There is another category of protocols known as Interior Gateway Protocols (IGP), designed to work inside ASs. These two types of protocols are needed mainly for scalability. With an IGP a router can know the complete set of paths that users need to follow to get to certain destination inside their AS. The small number of networks in an AS, comparatively speaking, allows that an IGP can gather all the information regarding the possible paths, however, if a user wants to send something to a destination outside their AS, the number of networks out there is way larger than inside any AS, so an IGP trying to gather all the possible routes a user can take to reach certain destination is unthinkable. Here is when having an EGP is useful, thanks to its path vector protocol nature, it is suitable for working with big scale networks.

Question 3.
What are the advantages of BGP running over TCP?

Using TCP as transport protocol eliminates the need to implement update fragmentation, retransmission, acknowledgement, and sequencing.

Question 4.
Why is it good for you as a client to avoid your AS from being a transit AS?

If you hire the services of two ISPs and at the same time you serve them as transit AS, this means that you are paying these ISPs to deliver your information to the Internet and also to let them communicate each other through your AS, basically you are also paying to work for them.

References

[1] Border Gateway Protocol. Retrieved from: http://docwiki.cisco.com/wiki/Border_Gateway_Protocol. Last consulted on: 10/08/2017.

[2] BGP in 2016. Retrieved from: https://labs.apnic.net/?p=952. Last consulted on: 10/08/2017.

[3] Border Gateway Protocol. Retrieved from: https://en.wikipedia.org/wiki/Border_Gateway_Protocol. Last consulted on: 10/08/2017.

[4] A Border Gateway Protocol 4 (BGP-4). Retrieved from: https://tools.ietf.org/html/rfc4271. Last consulted on: 10/08/2017.

[5] Multiprotocol Extensions for BGP-4. Internet Task Force [onlince] https://tools.ietf.org/html/rfc4760. Last consulted on: 20/09/2017.

[6] BGP Path Hijacking Attack Demo. Retrieved from: https://github.com/mininet/mininet/wiki/BGP-Path-Hijacking-Attack-Demo. Last consulted on: 28/08/2017.

[7] BGP Visualization Using Python. Retrieved from: https://basimaly.wordpress.com/2017/07/03/bgp-visualization-using-python/. Last consulted on: 25/09/2017.

[8] Chapter: Using the Command-Line Interface. Retrieved from: http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf001.html#wp1000994. Last consulted on: 04/08/2017.

[9] Chapter: Cisco IOS Command Modes. Retrieved from: http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf019.html. Last consulted on: 04/08/2017.

[10] Quagga. Retrieved from: http://www.nongnu.org/quagga/docs/docs-info.html#Basic-commands. Last consulted on: 04/08/2017.

[11] Computer Networks, Fifth Edition Andrew S. Tanenbaum, David J. Wetherall, Prentice Hall

[12] Les réseaux, 4th edition, Guy Pujolle, EYROLLES

[13] Stéphane Bortzmayer, a 3-hour cours about BGP https://www.bortzmeyer.org/cours-bgp-cnam.html

A. Useful Commands

A.1. In Mininet's Emulation Terminal

`xterm <component>`: opens an external terminal for the router specified.

`pingall`: verifies the connection of all the elements in the network.

`link <component_1> <component_2> down/up`: disables or enables the link between component_1 and component_2.

`exit`: exits the emulation.

Note: enter a question mark (?) and hit enter to see all the available commands.

A.2. In a Router's xterm Terminal

`route`: shows the routing table for that component.

`ping <ip>`: verifies the connection to the ip specified.

`nmap localhost`: shows the services running in your localhost. Use it to check if all the services needed for the emulation are up and running.

`telnet localhost <service>`: connects you to an specific service, in this case your router's specific service. Connect to bgpd for checking BGP related issues, e.g. checking the BGP routing table. Connect to zebra for general router's related issues, e.g; checking the routing table for the router (password: en).

`wireshark &`: opens wireshark in the component.

`ifconfig -a`: shows the interfaces in the component.

A.3. Inside bgpd service

`en`: enables privileged EXEC mode (password: en).

`show ip bgp`: shows the bgp routing table.

`show ip bgp neighbors`: shows BGP connectivity related information of router's neighbors.

`configure terminal`: from privileged EXEC mode, enters global configuration mode.

`router bgp <AS number>`: from the global configuration mode, specifies the router to be configured, and enters router configuration mode. Enter here the commands for configuring the bgp connection in your router.

Note: check [8] and [9] to know more about cisco commands and config modes. The commands might not be exactly the same used by cisco, in case of needing variation go to Quagga's documentation for basic commands in [10].

B. BGP Visualization

You will start, before the lab, by visualizing how the interconnection of several AS looks like in real life. For this, you are gonna use a Python module built for this purpose. Make sure to read the related blog article in [7].

To have access to the code for the visualizing, go to the course's VM, to directory `~/net_labs/bgp/`. Once you are there, you will see the file **ASVisualize.py**. Execute this file by using the command: `python ASVisualize.py`.

Open the file to find the code that makes possible this visualization, you will notice that there is not much to it and that it is very intuitive.

Now is your turn to play! Change the country in the code, which now is Saudi Arabia, and visualize the number of ASs and their connections per country. Here you have the list of acronyms per country: http://www.worldatlas.com/aatlas/ctycodes.htm.