Lab 5: BGP routing

50.012 Networks

Hand-out: November 20
eDimension hand-in: December 2

1 Objectives

- Get first-hand experience about how a routing protocol works using Zebra tools
- Get familiar with BGP
- Discover more about Mininet and learn about its basics

2 Experiments

2.1 Setting up the environment

- We assume you already have the mininet environment setup (refer to lab4 or `http://mininet.org/download/` if you need help)
- You also need to install wireshark tool. You can find installation instruction for wireshark at `https://linuxhint.com/install_wireshark_ubuntu/` and `https://www.wireshark.org/`
- Download the lab5.zip file from eDimension
- Unpack zip file contents into a directory, e.g. ~/**lab5**/. You can use command like 'chmod 666 lab5' to give it the right permission. `cd` into that directory.
- Install the quagga routing software suite (quagga is a fork of the original GNU zebra tool, see `https://www.nongnu.org/quagga/`: the name quagga comes from a subspecies of zebra, and quagga implements multiple routing protocols including BGP) and some other supporting tools, and set up configuration by running: (you will be asked to provide the password if you are not root already)

  ```
  ./install.sh
  ```

- You should now be able to start `sudo python bgp.py` (the script does several things, including: creating and starting the mininet environment, starting zebra and bgpd, starting some web servers for the testing later)
- Here are some potential issues you may face at this point:
  - If you have installed ovs-testcontroller, e.g., by

```
sudo apt-get install openvswitch-testcontroller
```

but mininet complained it cannot find required executable ovs-controller, you can use the following command to tell mininet how to find it

```
sudo ln /usr/bin/ovs-testcontroller /usr/bin/ovs-controller
```

– Remember to stop your openvswitch-controller or openvswitch-testcontroller service by running the following commands:

```
sudo service openvswitch-controller stop
sudo update-rc.d openvswitch-controller disable
```

or

```
sudo service openvswitch-testcontroller stop
sudo update-rc.d openvswitch-testcontroller disable
```

– Double check that you have the /var/run/quagga folder in your machine, otherwise zebra cannot start properly and the route from bgpd will not be written to the routing table. You can find the commands to create that folder in the install.sh

– After you run the bgp.py, start another terminal and use commands like 'ps aux | grep bgpd' to ensure that your bgpd daemons have started successfully. If it is not, then go to the logs/ subfolder and read the logs there to figure out what may have gone wrong.

– As mininet by default uses python 2, you likely need to run the bgp.py using python2. You may see the termcolor library missing from python2. In that case, you can use pip2 to install that library. Useful commands here:

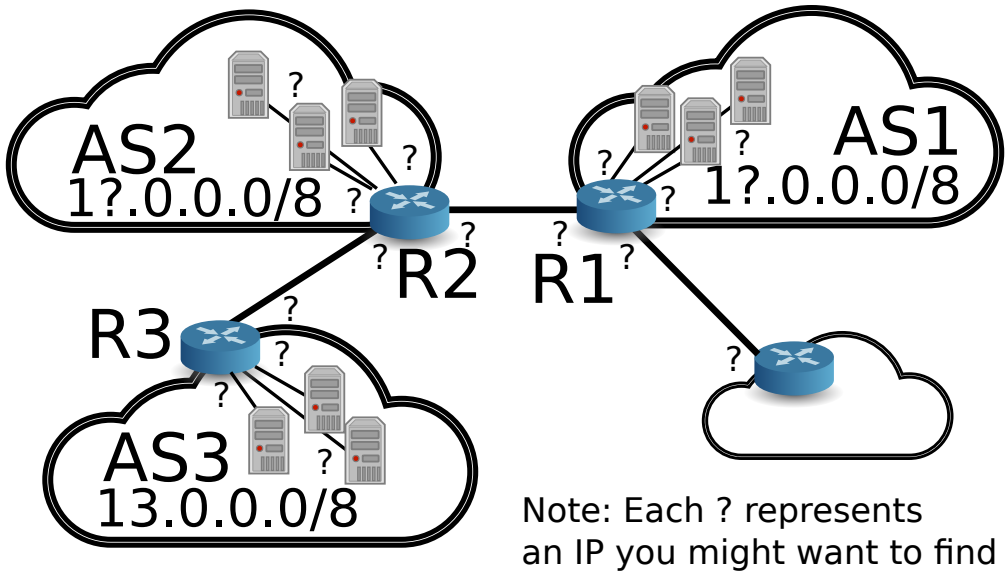
```
sudo apt-get install python-pip
sudo pip2 install termcolor
```

– If the executable zebra and bgpd are not in the folder of /usr/lib/quagga/ (as assumed by the bgp.py script), you may use the "which" command to find out their location and update the bgp.py file accordingly

2.2 Getting started

- After starting mininet with the above command, try to find out more about the current topology in mininet using `nodes` and `net` (also note `help <topic>`). You can also use `ping`, `ifconfig`, `zenmap`, or similar on the emulated nodes.

  – Annotate the following figure with the AS's announced prefix (network) and the IP addresses of the routers' interfaces. Please note that the "small" AS is not up yet at this stage, so you can't interact with it so much.



- There are also a number of hosts in each network. They represent different smaller AS internal networks connected to internal interfaces of the BGP routers.
- You can connect to the bgp daemons (`bgpd`) running on the nodes by running a script on the terminal (outside mininet)

  ```
  ./connect.sh R1
  ```

- The password for the bgpd is **zebra**
- The command line interface is belonging to bgpd (as provided by Zebra), a widely used BGP routing daemon.
- Experiment around with the different offered commands. Type `help` for a quick introduction about the bgpd (Quagga VTY) console. For example, using `show ip bgp`, you can list all IPv4 BGP routes. Note also that subcommands are auto-completable using `Tab`.

2.3 Observing BGP in action

- In mininet, start wireshark sessions on the individual routers by opening an xterm window for each router and then issuing the `wireshark &` command
- Start a wireshark session on one of the routers (make sure to select the right interface), and also open a bgpd command line session to it with the connect script.
- Type `enable` in the bgpd terminal (outside mininet) to enable admin mode (pw: zebra). Note that the prompt changed from `bgpd-R1>` to `bgpd-R1#`. Look at current routes with the `show ip bgp` command. Type the `clear bgp external` command to clear the exchanged routes.
- Watch the bgp traffic establishing the routes again in BGP.
- From h11, try to reach h33. Does it work? If no, why not?
- From R1, try to reach 13.0.1.1. Does it work? If no, why not?

- Modify the configuration files to allow R1 to reach 13.0.1.1. Configuration files can be found in the `conf` folder as explained briefly below. Alternatively, you could try to use the `route` command (hint: on R3).
  - bgpd-R1.conf and similar, that configure the bgpd setup of each router
  - zebra-R1.conf and similar, that configure the network setup of each router

2.4 Malicious BGP abuse

1. Introduction
   - Assume the following setting: a user from AS1 wants to visit a website on 13.0.1.1. A malicious attacker wants to redirect the user to its own webserver instead.
   - The attacker has control over AS4, which is BGP-peering with AS1
   - How can the attacker reach his goal?
2. To perform the attack, you will have to modify bgpd-R4.conf.
3. Performing the attack
   - Use the provided website script in a terminal like this: `./website.sh R1`
     - It will continuously contact a webserver on 13.0.1.1 from R1 (if you fixed R3's config). Leave the script running in that terminal.
   - Open a wireshark session on R1, make sure to listen on all `eth` interfaces
   - Run `./start_rogue.sh` script in a terminal. Observe the website results. Observe the wireshark traffic.
   - If you successfully configured R4, the victim should now see the attack website. Make sure that this is the case.
   - Running `./stop_rogue.sh` will stop the attack again if needed.

3 What to Hand in

For this lab, you can work together with another student, but please write up your findings and hand in your writeup individually. Please submit to eDimension a writeup (in PDF format with your name and student ID) that includes the following information:

- The topology as you were able to derive it (you could annotate the provided topology figure or compile a table):
  - IP addresses of all routers
  - Hosts/ IPs in the ASs
- Describe in detail the BGP traffic you were able to observe during re-establishment of routes.
- Was it initially possible to reach 13.0.1.1 from AS1 (h11 and R1, respectively)? If it didn't work initially, what caused that and what did you do to fix it?
- Describe in detail what happened when you started the attack on BGP.