Build your own Internet - Stage A

Computer Networks: Project 2

Current version: **11/23/2020**
Corrections:
11/24/2020: Swapped MIAM and GENE /32s in example in 3.2.1

1 Introduction

In this assignment, you will learn how to build and operate a layer-3 network using traditional distributed routing protocols, how different networks managed by different organizations interconnect with each other, and how protocols, configuration, and policy combine in Internet routing.

More specifically, you will first learn how to set up a valid forwarding state within an autonomous system (AS) using OSPF, an intra-domain routing protocol (Stage A). Then, you will learn how to set up valid forwarding state between different ASes, so that an end-host in one AS (e.g., your laptop connected to the university wireless network) can communicate with an end-host in another AS (e.g., Google's server). To do that, you will need to use the only inter-domain routing protocol deployed today: BGP (Stage B). After that, you will implement different BGP policies to reflect business relationships or traffic engineering that exist in the real Internet (Stage C). You will configure both OSPF and BGP through the FRRouting Software Suite, which runs on several software routers assigned to you.

The rest of the document is organized as follows. We first describe the setup you will have to use (Section 2). Then, we list the tasks you should perform (Section 3), submission and other general information (Section 4), and the collaboration and academic integrity policies. We are also providing a separate document giving a crash course on how to configure FRRouting routers.

1.1 Schedule

- ***You can use slip days and submit late for Stages C, but you cannot use slip days or submit late for Preliminary Stage, Stage A, or Stage B--any late submissions for the Preliminary Stage, Stage A, or Stage B will receive a 0.***
- The deadline for Stage A of this project is Nov. 30, at 11pm.

1.2 Collaboration policy

This is an *individual project*, but you can discuss at a conceptual level with other students or consult Internet material, as long as the final code and configuration you submit is completely yours and as long as you do not share code or configuration. Before starting the project, be sure to read the collaboration and academic integrity policy later in this document.

1.3 Instructor/TA information

As explained in the Preliminary Stage document, we will assign each student an autonomous system. Each student is assigned to a TA, as listed on Piazza. Please ask any general questions related to this assignment on Piazza, visible to all unless they reveal private details of your solution, and only contact your responsible TA for questions that will not be relevant to other students (e.g. you are unable to access your autonomous system).

2. General project setup

See the Preliminary Stage document for further details.

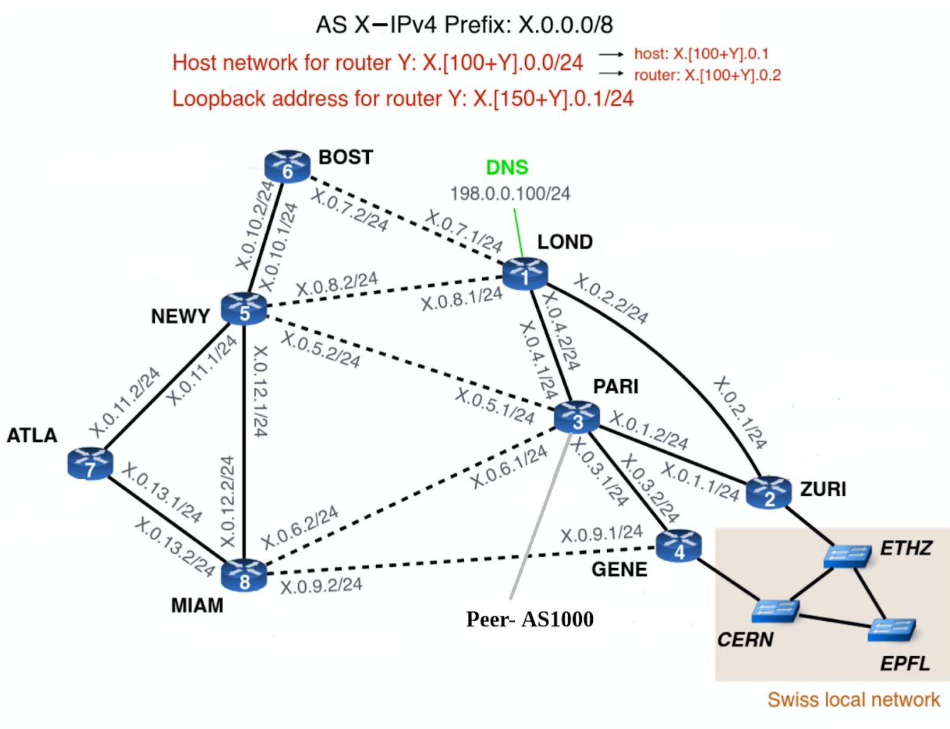

Figure 1: The network topology of your AS. Your AS is composed of 8 routers. A /8 prefix has been assigned to you. You can use it to configure your local networks. The subnets you must use for each of your local networks are indicated on each interface.

- Figure 1 shows what your AS looks like. At this stage, we will not be using the network of switches connecting to GENE and ZURI (Swiss local network), and restrict ourselves to the rest of the topology (including GENE and ZURI, just not the network hanging off them).
- As shown in the figure, your network has 8 routers, each named after a city (e.g., ATLA).
- In addition, there are 6 hosts, 1 connected to each router except for GENE and ZURI. They are named <*CITY*>**-host** (e.g., **BOST-host** is the host connected to the **BOST** router).
- If you are AS X, then the prefix X.0.0.0/8 is yours. For example, AS 5 owns the prefix 5.0.0.0/8. You will use this IP space to allocate IP addresses to your hosts and routers.
- Additionally, each AS also hosts a DNS server. The DNS service is helpful while decoding traceroute outputs- the corresponding IP addresses are replaced with host names.
  - For example, 19.0.2.2 will be translated into LOND-ZURI.group19, because this is the IP address configured on the interface of the LOND router in AS19 that connects to ZURI.

- We have preconfigured all your hosts to use the DNS service- just like in the real world, only hosts use DNS servers, and not any other component.
- The DNS server is connected to your LOND router over the *dns* interface, over the network 198.0.0.0/24. Do not forget to include this network in your OSPF configuration.
- (The fact that some links are solid and some are dotted is not relevant at this time)

2.5 Accessing and configuring routers and hosts

Detailed instructions on how to access your hosts and routers from within your AS are available in the tutorial here.

Important: DO NOT change the password assigned to you.

Important: If you break any of your AS components (host/router/AS container) (more information on the AS container can be found in the tutorial), the only fix is to revert the component into the **initial state**. Hence, it is important that you **regularly snapshot and backup your configuration**. You will then have to re-enter (by hand, unless you develop your own approach for automation) the saved configuration. If you need to revert your AS component, please post your request on Piazza in the following format:
Title: Project2
Name: Shiv Venkatagiri
UNI: skv2109
Internet Number: 5
Request: revert AS component - [Component]

Eg: Request: Revert AS component - PARI

The students get root access to the AS container. Changing the credentials, or messing around with system setting can result in your container breaking.

3 Your Tasks

This project is composed of a very short preliminary Stage, then three main Stages (A-C), with Stage B being primarily completed during the class-wide “Internet Hackathon.”

- Stage A involves setting up routing within your own network via OSPF and iBGP configuration, and must be finished before the Internet Hackathon.
- Stage B (Internet Hackathon) involves bringing up your eBGP sessions with your neighboring ASes and advertising your prefixes to your neighbors. We will provide details closer to the date of the Hackathon.
- Stage C involves implementing BGP policies according to the business relationships that you have with your neighbors (we will assign the relationships). We will provide details closer to the date of the Hackathon.

Possible plan of attack:

- Familiarize yourself with the previous section, and, using the instructions in the tutorial, access your AS and navigate to routers/hosts.
- For any stage, familiarize with the goals of the stage. Then, refer to the tutorial we provide to find the basics of the commands you will need to enact the goals. Our expectation is that you may need to experiment and try things out to figure out how to accomplish a task based on the guide.

3.1 Preliminary Stage: Your first BGP session (30 points)

Please look at the preliminary stage document for this section of the project.

3.2 Stage A: Configure IGP and iBGP (40 points)

3.2.1 Task - Set up interfaces and set up OSPF (20 points)

Configure OSPF to enable end-to-end connectivity between all the hosts inside your AS.

Before configuring OSPF, you must configure the IP addresses for each interface of your routers and hosts. Figure 1 in the section above depicts how you should assign addresses:

- You must use the IP addresses shown in Figure 1. For the router interfaces between NEWY and PARI, for example, you have to use the subnet X.0.5.0/24. The interface in PARI that is connected to NEWY must have the IP address X.0.5.1 and the interface in NEWY that is connected to PARI must have the IP address X.0.5.2 (where X is your AS number).
- Every router also has a loopback interface with the name `lo` that you have to configure. The router with ID Y has the loopback address X.[150+Y].0.1/24 where X is your AS number (router IDs are shown on each router in the figure; for example the ID of BOST is 6).
    - As an example, the loopback address of the router BOST for the AS 10 is 10.156.0.1/24.
- For the connection between the routers and their corresponding host, you have to use the subnet X.[100+Y].0.0/24, where X is your AS number, and Y is the ID of the router. Then, the host gets the IP address X.[100+Y].0.1 and the interface of the router that is connected to this host will have the IP address X.[100+Y].0.2.
    - For example, the subnet used for AS 85 between the MIAM router and the corresponding host is 85.108.0.0/24. The interface at the router MIAM that is connected to the host, is called **`host`** and uses 85.108.0.2/24. The interface of the host connected to the router is called `MIAM` router and uses the IP address 85.108.0.1/24.

- When you assign IP addresses to your routers, between each pair of routers you will have to use the subnet (within your /8 prefix) indicated in Figure 1.
    - For example, between MIAM and GENE, you *must* use the subnet X.0.9.0/24.
    - The interface in MIAM that is connected to GENE must have the IP address X.0.9.2, and the interface in GENE that is connected to MIAM must have the IP address X.0.9.1 (with X being your AS number).

The interfaces already exist but do not yet have IP addresses, which you need to configure using commands in the tutorial. You can view all the existing router interfaces using the command **show interface** in the FRR Routing CLI (as described in the tutorial).

Important: To support the infrastructure of this project, you will see multiple other interfaces, including ones connecting to a measurement server, a DNS server, an IXP, etc. **Do not modify any other interface present.**

For hosts, you will also have to configure a default gateway, which tells the host a default direction to send all traffic outside of its subnet.

Make sure that each router and host can ping its directly connected router before you start the OSPF configuration. Then, configure OSPF for each router in your network. The tutorial section on OSPF describes useful commands. After you have configured OSPF for each router, test host-to-host connectivity with pings. Do not move on to other stages of the assignment before you verify that every host can ping every other host.

In the report, please paste the screenshot of ping results from MIAM-host to BOST-host. You can use the command "**ping <ip_address> -c 10**" to conduct 10 pings to the IP address specified). The automatic grading will test all connectivities, not just the one between these two hosts.

3.2. Task - Setup iBGP (20 points)

Configure internal BGP sessions (iBGP) between all pairs of routers (full-mesh).

Not related to the important problem below: For each iBGP neighbor, please also add the following command and think about why it is essential for iBGP neighbors (we will discuss this more in Stage B):

router_name(config-router)# neighbor <ip_address> next-hop-self

Important problem (iBGP connections): By default, when a router establishes an iBGP session with a peer, it uses the IP address of the interface closest to the iBGP peer as the source address. This is fine as long as this interface is up. If this interface goes down but the actual router is still running, the iBGP connection will tear down even though the router may be reachable over a different path/interface. So you should configure all iBGP sessions to not suffer from this problem--as long as the peer router is reachable by *some* IGP route, the session should persist. *Hint: You will have to use an FRR command that we have not told you about in the tutorial, so you will have to do research on your own.*

Verify that each of your routers does have an iBGP session with all the other routers with the command **show ip bgp summary**.

If you have configured iBGP properly, then the advertisement from AS1000 should have been propagated through your iBGP sessions to all your routers. You can verify that a router indeed received that advertisement with **show ip route bgp** or **show ip bgp**.

In your report:

1. Please paste the screenshot of your **show ip bgp summary** results from ATLA router.
2. Please paste the screenshot of **show ip route bgp** results from ATLA router.
3. Describe what commands you used to solve the **important problem** for iBGP connections above.
4. Describe how those commands solve the problem.
5. Describe why it is important to address that problem for iBGP sessions but not for eBGP sessions (like the one you established to AS1000 in the Preliminary Stage).

3.3 Stage B: Establish BGP interconnectivity (50 points)

3.3.1 Task - Attend Hackathon (10 points)

Attend the in-class Hackathon.

Further details of Stage B will be released before the Hackathon.

3.4 Stage C: BGP policy & delivering Internet content (80 points)

Further details of Stage C will be released after the Hackathon.

4 Submission and other information

The project is worth 200 points (Preliminary Stage: 30 points, Stage A: 60 points, Stage B: 50 points, Stage C: 60 points). In reports that you submit, clearly label which question you are answering with your answer/screenshot by writing the Task number and name, e.g., “4.1.3 Task - Advertise prefix to AS1000.”

Remember to run ./save_configs.sh, copy the corresponding file generated to the machine you plan to submit from, and rename the folder before submitting to make sure you submit the version you intend.

You will submit it via Courseworks. The format of your submission is the same as your preliminary stage.
You must include the following files in a compressed file called **project2a_Lastname_Firstname_ASN.zip** (for example, *project2a_Katz-Bassett_Ethan_AS10.zip* if the professor were AS10 in the topology). The files are:

1. Written report, with filename **report.pdf**
2. The entire saved **configs[date][time]** directory on your AS under home directory renamed as **configs**.

The result of the above will be a list of text files named as **routers (PARI.txt, NEWY.txt, ATLA.txt etc) or Switches (ETHZ.txt, CERN.txt, EPFL.txt)**. Zip this folder along with your report.pdf. Your zipped submission file, for example, should have the following directory structure **after being unzipped** (the top level must contain the project2a* directory):

```
project2a_Venkatagiri_Shiv_AS7/
├── report.pdf
├── configs
│   ├── ATLA.txt
│   └── PARI.txt
│   ├── .
│   ├── .
│   ├── .
```

5 Academic integrity: Zero tolerance on plagiarism

The rules for Columbia University, the CS Department, and the EE Department (via SEAS: 1 and 2) apply. It is your responsibility to carefully read these policies and ask the professor (via Piazza) if you have any questions about academic integrity. **Please ask the professor before submitting the assignment, with enough time to resolve the issue before the deadline**. A misunderstanding of university or class policies is not an excuse for violating a policy.

This class requires closely obeying the policy on academic integrity, and has zero tolerance on plagiarism for all assignments, including both projects/programming assignments and written assignments. By zero tolerance, we mean that the minimum punishment for plagiarism/cheating is a 0 for the assignment, and all cases will be referred to the Dean of Students.

This assignment must be completed individually. For programming assignments, in particular, you must write all the code you hand in yourself, except for code that we give you as part of the assignments. You are not allowed to look at anyone else's solution (including solutions on the Internet, if there are any), and you are not allowed to look at code from previous years or ask people who took the class in previous years for help. You may discuss the assignments with other students at the conceptual level, but you may not write pseudocode together, or look at or copy each other's code. Helping other students violate the policy (for example, letting them look at your code) is a violation, even if you completed the code yourself. Please do not publish your code or make it available to future students -- for example, please do not make your code visible on Github. Uploading course materials to sites such as CourseHero, Chegg or Github is academic misconduct at Columbia (see pg 10).

You may look at documentation from the tools' websites. However, you may not use external libraries or any online code unless granted explicit permission by the professor or TA. For written (non-programming) answers, if you quote material from textbooks, journal articles, manuals, etc., you **must** include a citation that gives proper credit to the source to avoid suspicion of plagiarism. If you are unsure how to properly cite, you can use the web to find references on scientific citations, or ask fellow students and TAs on Piazza.

For each programming assignment, we will use software to check for plagiarized code. **So, be really careful and do not read or copy code or text.**

Note: You must set permissions on any homework assignments so that they are readable only by you. You may get reprimanded for facilitating cheating if you do not follow this rule.