Build your own Internet - Preliminary Stage

Computer Networks: Project 2

Current version: **11/21/2020**
Corrections: None so far

1 Introduction
- 1.1 Schedule
- 1.2 Collaboration policy
- 1.3 Instructor/TA information

2. General project setup
- 2.1 Overview of project infrastructure
- 2.2 Getting your AS assignment
- 2.3 Accessing/Managing Your AS
- 2.4 - Network Topology
- 2.5 Accessing and configuring routers and hosts

3 Your Tasks
- 3.1 Preliminary Stage: Your first BGP session (30 points)
  - 3.1.1 Task - Establish cross-AS link to AS1000 (10 points)
  - 3.1.2 Task - Configure eBGP sessions with AS1000 (5 points)
  - 3.1.3 Task - Advertise prefix to AS1000 (15 points)
- 3.2 Stage A: Configure IGP and iBGP (60 points)
- 3.3 Stage B: Establish BGP interconnectivity (50 points)
  - 3.3.1 Task - Attend Hackathon (10 points)
- 3.4 Stage C: BGP policy & delivering Internet content (60 points)

4 Submission and other information
- 4.1 Preliminary Stage:

5 Academic integrity: Zero tolerance on plagiarism

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 the Preliminary Stage is Tuesday Nov. 24 at 11pm. (This Stage requires less work than the others, so you are strongly encouraged to complete this Stage immediately and to start Stage A before the Preliminary Stage is due).

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 below, we will assign each student an autonomous system. Each student is assigned to a TA. 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

2.1 Overview of project infrastructure

- Each student will configure one AS, including multiple routers and hosts.
- You access your AS via a container (like a lightweight VM) hosted in the cloud. You access this container via SSH, as we describe below.
- Each router and host lives in its own container. You access these containers from within your AS container, as we describe below.
- Each router operates routing software, providing equivalent functionality to a real-world hardware router.
- In the Preliminary stage, you will connect your AS to an AS managed by the instructors, to learn the process of configuring inter-AS connectivity.
- In Stage A, you will configure connectivity among the hosts and routers within your own AS, allowing traffic to flow between them.
- In later stages, you will work together with classmates to configure connectivity across the full set of ASes, resulting in a working internet.

2.2 Getting your AS assignment

We will email you important information to your UNI email address:

- You are responsible for one AS in the mini-internet we develop over the course of this project. The email provides:
    - The name of the TA in charge of your Internet.
    - The IP address of your internet (see below).
    - Your AS number. Your AS number is unique within the internet.
    - The password for accessing the AS container that hosts your AS (see below).

We will post on Piazza once we send this email, including telling you what email account we sent it from and what the subject line of the email is. After we post to Piazza, if you do not see the email, please search for it using this information. If you still do not see it, please post a question to Piazza (visible only to you and the instructors) with the following format:

Title: Project2 Credentials not received
Name: <name>
UNI: <uni>

2.3 Accessing/Managing Your AS

Detailed explanations on how to access and manage your ASes can be found on the tutorial here.

2.4 - Network Topology

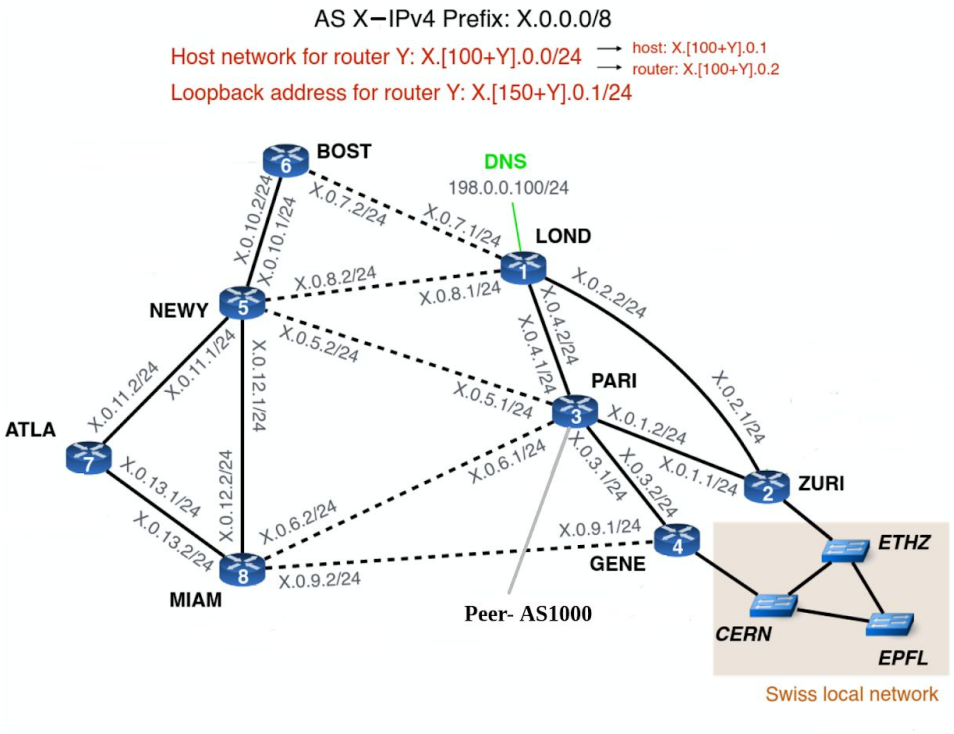

Figure 1: The network topology of your AS. Your AS is composed of 8 routers. A /8 prefix has been assigned to you. You can use it to configure your local networks. The subnets you must use for each of your local networks are indicated on each interface.

- 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)

You will configure a single BGP session with AS1000, an AS run by the TAs. The TAs will use this session to confirm that you know how to establish a session and are ready to move on to establishing all the sessions necessary for the Hackathon. The tasks in this Stage walk you through the process. In addition to you submitting the work you do, we will check that you have completed the Stage successfully from AS1000. In Stage B, you will configure BGP sessions with your classmates' ASes, so the Preliminary Stage is your chance to practice and make sure you know how to do it.

3.1.1 Task - Establish cross-AS link to AS1000 (10 points)

First you must establish a link between your **PARI** router and a router in AS1000's LOND router. For this purpose alone, we shall be using 180.0.x.0/24 as our subnet, where X is your AS number.

- AS1000 has already configured the interface *ext_1000_LOND* on its peering router to use the .1 within the subnet.
- Your PARI router will use the .2 within the subnet. Configure the interface on your PARI router to use this IP address. The tutorial describes how to configure router interfaces, as well as most other configuration commands you will need for the project.

After you have configured the *ext_1000_LOND* interface, use the `ping` command to ping the .1 address taken by your neighbor AS from your **PARI** router to test the connectivity. If you receive a ping response, you have completed the task successfully.

In your report, please paste the screenshot of the successful ping from your PARI router. (You can use Ctrl-c to kill ping after it issues a few measurements--the number does not matter as long as the router in the other AS is successfully responding to your measurements).

3.1.2 Task - Configure eBGP sessions with AS1000 (5 points)

At your **PARI** router, configure the external BGP session (eBGP) with AS1000. When specifying a BGP neighbor, use the .1 address in the subnet, as AS1000 is using this address on the interface facing your router. Again, the tutorial describes most configuration commands you will need.

The AS1000 side of the connection is already configured and ready to announce a prefix to you, and so the connection should be established as soon as you get your configuration right.

Verify that the eBGP session is established successfully using command

PARI_router# show ip bgp summary.

If the command shows that the connection is established, you have completed this task successfully. If you have trouble completing the task, read the tutorial carefully, and supplement by reading documentation online to make sure you are configuring things properly.

In your report, please paste the screenshot of your **show ip bgp summary** results from **PARI** router.

AS1000 should have also advertised a prefix to you once the connection was established. Verify that you indeed received this advertisement by running the following from the CLI of your PARI router:

PARI_router# show ip route bgp
or
PARI_router# show ip bgp

In your report, please paste the screenshot of the output of one of these commands showing that you are receiving the advertisement directly from AS1000 of a prefix originated by AS1000.

3.1.3 Task - Advertise prefix to AS1000 (15 points)

Once the eBGP sessions are up, advertise your prefix to AS1000. You must *only* originate the /8 prefix that has been assigned to you (see the section describing your network topology). The tutorial describes how to advertise a prefix in BGP.

Run (subbing in the IP of the neighbor)

PARI_router# show ip bgp neighbor <IP-ADDRESS> advertise

to show that your router is announcing the prefix. In your report, please paste the screenshot of the output of this command from the PARI router. If your PARI router receives an advertisement from AS1000, and it makes an advertisement for your /8 to AS1000, then you have successfully completed the Preliminary Stage.

3.2 Stage A: Configure IGP and iBGP (60 points)

Further details of Stage A will be released shortly.

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 (60 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.

4.1 Preliminary Stage:

You will submit it via Courseworks.
You must include the following files in a compressed file called **project2prelim_Lastname_Firstname_ASN.zip** (for example, *project2prelim_Katz-Bassett_Ethan_AS10.zip* if the professor were AS10 in the topology). The files are:

1. Written report, with filename **report.pdf**. (The report is the screenshots you are instructed to take in the Stage's Tasks).
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 project2prelim* directory):

project2prelim_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.