Build your own Internet - Stage B

Computer Networks: Project 2

Current version: **11/23/2019**

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 Quagga software routing suite, which runs on several virtual routers in your virtual machine (VM).

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 Quagga routers.

1.1 Schedule

- ***You can use slip days and submit late for Stages A and C, but you cannot use slip days or submit late for Preliminary Stage or Stage B--any late submissions for the Preliminary Stage or Stage B will receive a 0.***
- The deadline for the Preliminary Stage is Nov. 16 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).
- The deadline for Stage A of this project is Nov. 21, at 11pm.
- Stage B includes a MANDATORY in-class Hackathon on Nov. 25.
- The deadline for the Stage B submission is Dec. 2 at 11pm.
- The deadline for Stage C is Dec. 9th, 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 Internets

The class will be divided into six independent internets (18~19 students each). Each internet has the same network topology specified in figure 1. Additionally, each internet will be assigned a TA who is responsible for configuring the TA ASes within your network, helping you during the hackathon, and monitoring the states of your VM.

Please ask any general questions related to this assignment on Piazza, and only contact your responsible TA for personal questions (e.g. unable to access the VM, need rollback to initial VM state).

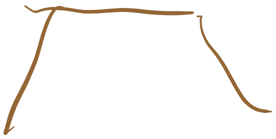
2. General project setup

2.1 Getting your AS assignment

See Stage A document.

2.2 Accessing Your VM

See Stage A document.



2.3 - Network Topology

TCP HTT

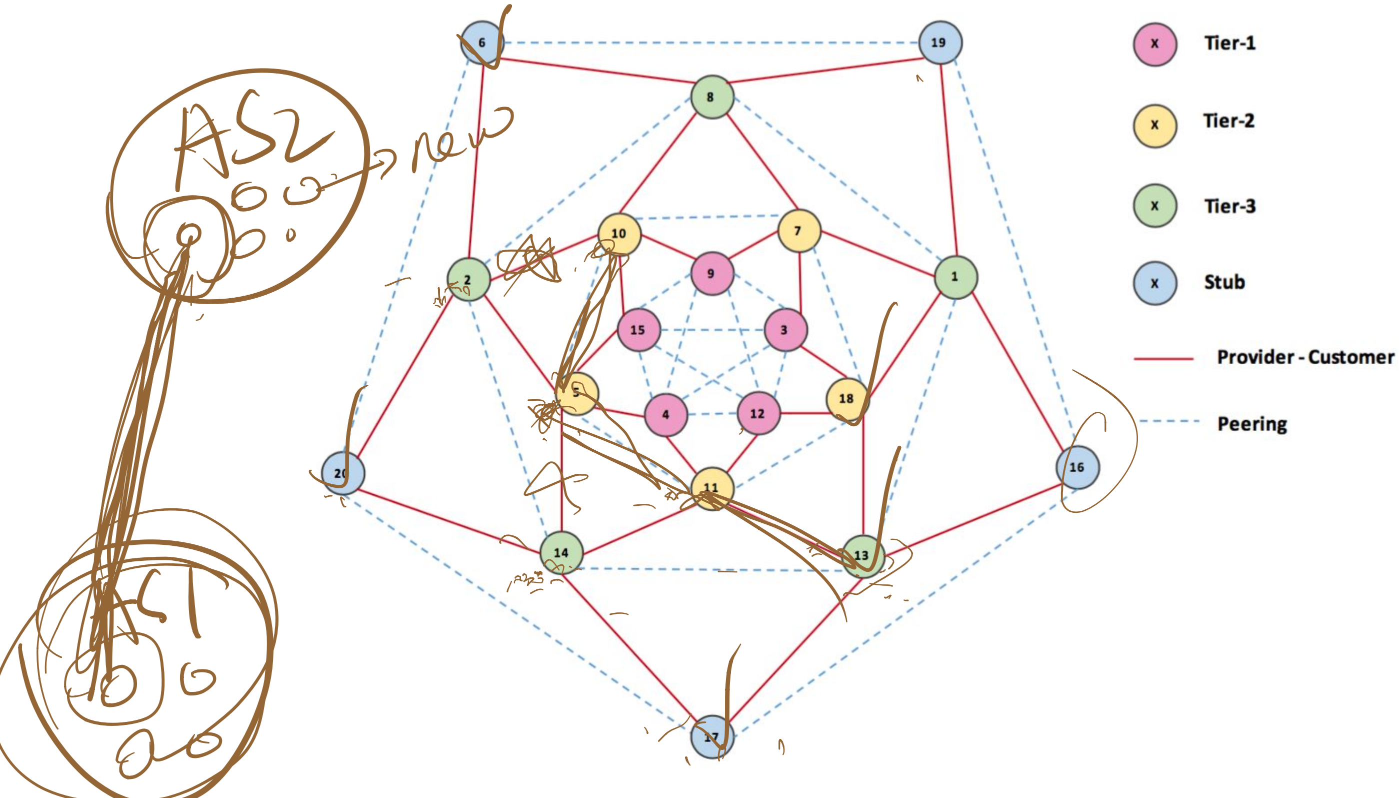


Figure 2: AS-level network topology

The Stage A document described the intra-AS router-level topology of your AS in Figure 1 (not in this document). Different ASes are connected with each other as shown in Figure 2 (where each node represents an entire AS, with the router-level topology from Figure 1 in Stage A). There are 4 types of networks: tier-1, tier-2, tier-3, and stub networks. There are 5 networks of each type. For example, AS 9, 3, 12, 4, 15 are tier-1 networks. For each tier-1 AS, it has 4 tier-1 peers and 2 tier-2 customers. For each tier-2 and tier-3 AS, it has 2 providers, 2 peers, and 2 customers. For each stub AS, it has 2 providers and 2 peers. Additionally, AS22 is not shown in the figure but connects to every AS. You established your connections to AS22 in the Preliminary Stage, and you will configure it as your customer in Stage C.

2.4 Accessing routers and hosts

See Stage A document.

2.5 Saving / snapshotting router configuration

See Stage A document.

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.
- Stage C involves implementing BGP policies according to the business relationships that you have with your neighbors (we will assign the relationships).

Possible plan of attack:

- Familiarize yourself with previous section, access your AS and navigate to routers/hosts.
- For any stage, familiarize with the goals of the stage. Then, refer to the Quagga guide 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)

See Stage A document.

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

See Stage A document.

3.3 Stage B: Establish BGP interconnectivity (50 points)

3.3.0 Task - Reassign OSPF weights (8 points awarded as part of Stage A)

(you do not need to complete this task first, so do not let it delay your hackathon participation, but make sure to complete it at some point)

Assign the OSPF weights according to the link (edge) labels in the topology figure from the Stage A assignment, undoing any changes you made to realize ECMP in Task 3.2.4. The goal of this task is the same as for "3.2.3 Task - Assign OSPF weights" from Stage A.

3.3.1 Task - Attend Hackathon (10 points)

Attend the in-class Hackathon.

3.3.2 Task - Establish cross-AS links (10 points)

A file `as-connections.txt` (`/root/configs/as-connections.txt` of your VM) shows all the connections that should be implemented between ASes. Each connection record shows the type of the connection, which router to use to connect to a neighboring AS, and the subnet that should be used for this connection.

AS_type	connection_type	local_AS	remote_AS	peering_location	subnet
Tier2	Prov1	10	15	NEWY	179.24.19.0/24
Tier2	Prov2	10	9	SEAT	179.24.4.0/24
Tier2	Peer1	10	5	WASH	179.24.30.0/24
Tier2	Peer2	10	7	SALT	179.24.20.0/24
Tier2	Cust1	10	2	KANS	179.24.33.0/24
Tier2	Cust2	10	8	LOSA	179.24.34.0/24
Tier2	Mgnt	10	99	HOUS-MGT	10.0.199.1/24

Table 1: An example of what you can find in `as-connections.txt`

Table 1 shows some connection records related to AS 10 in `as-connections.txt`. The records show that AS 10 is a tier-2 network, and has two providers (AS 15 and AS 9), two peers (AS 5 and AS 7), and two customers (AS 2 and AS 8). AS 10 is connected to AS 15 via its router NEWY. Between AS 10 and AS 15, the subnet 179.24.19.0/24 must be used.

During the hackathon, you must negotiate with your neighboring ASes to decide who takes which IP address in the subnet used to connect your AS and your neighboring AS. After that, you have to configure the interface ***ebgp*** with the IP address you took. For example, Table 1 shows AS 10's NEWY router connecting to AS 15. AS 10 can configure the *ebgp* interface at NEWY using 179.24.19.27, and AS 15 can use 179.24.19.32 after negotiations (or any other combination they agree on within the specified subnet).

Ping the address taken by your neighboring AS from your peering router to test the connectivity after both ASes have configured the *ebgp* interface.

AS 99 is a special ASN for the Internet's looking glass service. You will configure the cross-AS link with AS 99 in task 3.3.5 below. You do not need to establish a BGP session with AS 99.

NEWY	2.
SEAT	8
WASH	20
SALT.	19

3.3.3 Task - Configure eBGP sessions (5 points)

Configure the external BGP sessions (eBGP) with your neighboring ASes. Use the IP address of the neighboring router's *ebgp* interface (as specified in the previous task) when specifying a BGP neighbor.

Verify that eBGP sessions are established successfully using command **show ip bgp summary**.

3.3.4 Task - Advertise prefix to other ASes (10 points)

Once the eBGP sessions are up, advertise your prefix to your neighboring ASes. You must *only* originate the /8 that has been assigned to you.

In the meantime, your peers should advertise to you their /8 prefix, as well as all the /8 prefixes they have learned from other ASes (since there are no BGP policies, yet). Verify that you indeed received those advertisements with **show ip route bgp** or **show ip bgp**, and that those advertisements have been correctly propagated through your iBGP sessions.

Test your connectivity from your hosts towards hosts in other ASes using ping and traceroute.

Important. If you can reach destinations in another AS from a peering router, but not from a non-peering router, you likely have a configuration problem. To fix it, double check that you have added **next-hop-self** command when configuring iBGP sessions. The advertisements from external networks will have the next-hop attribute set to the IP address belonging to the subnet connecting your AS and the other AS, and only one of your routers (the peering one) knows how to reach that address. Thus, this router needs to use **next-hop-self** to modify the next-hop attribute, so that other routers know that they can send traffic to this router if they want to reach the destinations in the external advertisement.

3.3.5 Task - List eBGP configuration (10 points)

Please generate a new as-connections.txt ASCII file where the local_AS is your AS. The file format should be similar to the existing `as-connections.txt` file, but it should contain two additional columns indicating the IP addresses you choose with your neighbors to configure the ebgp interfaces. The file should be called **`as`*<X>*`-connections.txt`**. In place of ***<X>***, please put the number of your as (for example, **`as5-connections.txt`**). Below we describe how to submit the file.

AS_type	connection_type	local_AS	remote_AS	peering_location	subnet	local_ip	remote_ip

Table 2: An example of the headers of the new table

6.0.199.1

3.3.5 Task - Configure management interface (5 points)

We will set up a management VM which allows you to launch pings and traceroutes from any AS (not necessarily your AS), towards any destination in the mini-Internet. It will help you to debug your configuration in Stage C. To do so, you must first set up a connection with this VM via the interface **mgt** of the router **HOUS**. You must configure the IP address for this interface using X.0.199.1/24, where X is your AS number. For example, AS 5 must set the IP address to 5.0.199.1/24. If it is configured correctly, you should be able to ping the other side of the link, X.0.199.2/24, where X is your AS number.

Once you have verified the interface's IP address, activate OSPF on the interface (See Stage A, task 3.2.1 and appendix A.4). You do not need to assign the interface a weight. This will ensure that all routers in your network have a route to the hosts in X.0.199.0/24.

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 AS22."

Remember to write your configurations to file before submitting, to make sure you submit the version you intent.

4.1 Preliminary Stage: See Stage A document for details

4.2 Stage A: See Stage A document for details

4.3 Stage B:

You will submit it via Courseworks.
You must include the following files in a compressed file called **`project2b_Lastname_Firstname_ASN.zip`** (for example, **`project2b_Venkatagiri_Shiv_AS10.zip`** if TA Shiv were AS10 in the topology). The files are:

1. A new ASCII file called **`as<X>-connections.txt`**, using the format described in Task - List eBGP configuration. In place of ***<X>***, please put the number of your as (for example, **`as10-connections.txt`**).
2. The entire **`/root/configs/quagga`** directory on your VM. ***Remember to save your configuration first***.

The result of the above will be a list of directories named as **`routers (NEWY, WASH, ATLA`** **etc)** and one named **`templates`** with their configuration files inside them inside the **configs** folder. 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 project2b* directory):

```
project2b_Venkatagiri_Shiv_AS10/
├── as10-connections.txt
├── quagga
|    ├── ATLA
|    |    ├── bgpd.conf
...
|    └── WASH
|         ├── bgpd.conf
|         ├── daemons
|         ├── debian.conf
|         ├── ospfd.conf
|         ├── vtysh.conf
|         └── zebra.conf
```

4.4 Stage C:

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

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.

Unless explicitly stated otherwise on the assignment itself, assignments 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. 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. Please do not publish your code or make it available to future students -- for example, please do not make your code visible on Github. 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.