University of Crete
Computer Science Department
Lecturer: Prof. X. Dimitropoulos

Spring Semester 2023
HY335B - Computer Networks
Assigned: 03 March 2023
Due: 19 May 2023, 23:59

Mini Internet Project

In this project, you together with more than 100 of your fellow classmates will build and operate your very own mini-Internet. Your main goal? Enabling end-to-end connectivity across ≈ 80 Autonomous Systems (ASes) composed of hundreds of network devices. In doing so, you will experiment with the most common switching and routing technologies used today in the Internet. You will also face the same challenges actual network operators experience every day.
To reach Internet-wide connectivity, you will first need to enable internal connectivity, *within* your own AS, before interconnecting your AS with others ASes, managed by other groups of students. To establish connectivity *within* your AS, you will use the Open Shortest Path First (OSPF) protocol. To establish connectivity *across* different ASes, you will use the only inter-domain routing protocol available today: the Border Gateway Protocol (BGP). At the end of the project, any end-host should be able to communicate with each other, independently of the AS they are located in.
To help you, we have pre-built a base network topology on top of virtual layer-2 switches, running Open vSwitch [1] and virtual routers, running the FRRouting software routing suite [2]. You will configure the virtual switches and routers through a Command Line Interface (CLI). This interface is virtually identical to the one used by actual network operators.
The rest of this document is organized as follows: Section 1 provides general information about the project, including **submission instructions**. Section 2 gives an overview of the mini-Internet and the network you will be configuring.

1 General Information

This section tells you what to do if you have questions, how to backup and submit your work and how it will be graded. Furthermore, it explains our policies on academic integrity and misuse of the resources.

1.1 If you have questions

In case of questions, please ask the TAs during the exercise sessions or use the Moodle forum.

1.2 Regularly backup your work

We provide you with a script that automatically saves all configs of all your routers and switches (`save_configs.sh`) in one place (see also the tutorial pdf). We advise you to use this script regularly and to copy the generated folder to your local machine in order to prevent losing your work in the case of unexpected problems.

1.3 Submit your work

Send your **report and configuration** using TURNIN. Make sure that your deliverable includes a zip or tar.gz archive containing a PDF report as well as all your configuration files (the directory generated with the `save_configs.sh` script). Please make sure that your PDF report includes your group number as well as the name of the members in your group. The maximum length for your PDF report is 10 A4 pages (including screenshots, traceroutes, looking glass etc.).

Important: Do not use the mailing list for your questions.

1.4 Our grading policy

This assignment will be graded and counts for 40% of your final grade. The project consists of 2 phases with 5 questions per phase.

1.5 Academic integrity

We adopt a strict zero tolerance policy when it comes to cheating. Cheating will immediately result in the group failing the assignment. In particular, you can only do your assignment with the other members of your group. Do not look at other groups' configuration and do not copy configurations from anywhere. It is OK to discuss things or find help online, but you must do the work by yourself.
Your configuration and report may be checked with automated tools so as to discover plagiarism. Again, **do not copy-and-paste** code, text, etc.

1.6 Misuse of the resources and infrastructure

It is prohibited to use and modify the network in other ways than expressly allowed in this task description. The forbidden misuse includes, but is not limited to, BGP hijacks, DDoS attacks, resource-/bandwidth-hungry programs, and the attempt to access the docker containers of other groups. We monitor, investigate, and apply the appropriate disciplinary actions for cases of misuse.

2 Network Topologies

Similarly to real networks, your network spans over layer-2 (using switches) and layer-3 (using routers). Your network also connects (at layer-3) to other networks, creating an *Inter*net. We now describe each aspect of the network topology.

L2 topology Your layer-2 network is composed of three switches (Fig. 1) located at three different locations: ETHZ, EPFL, and CERN. The switch at ETHZ is connected to a layer-3 router (ZURI), the one at CERN is connected to GENE. Both routers can act as a gateway, meaning that a host in the local network must send a packet to it to reach any non-local destination. The router will then take care of sending that packet to the destination.

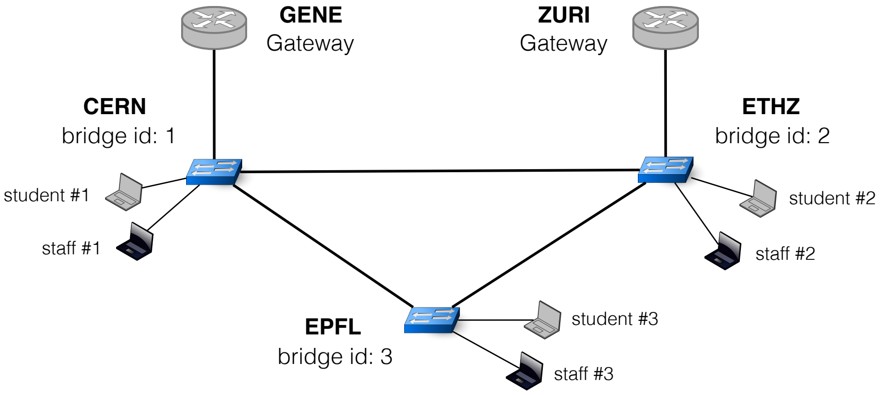

Figure 1: Each group will have to manage its own local network. This layer-2 network is composed of three Open vSwitches located at different locations. The switch at ETHZ connects to a layer-3 router (ZURI) which acts as the gateway and CERN is connected to GENE.

Two types of users exist in your layer-2 network: students and staff. Each switch is connected to one student and one staff member. Each switch also has a bridge ID which is indicated in Figure 1. For example, the switch at ETHZ has bridge ID 2.

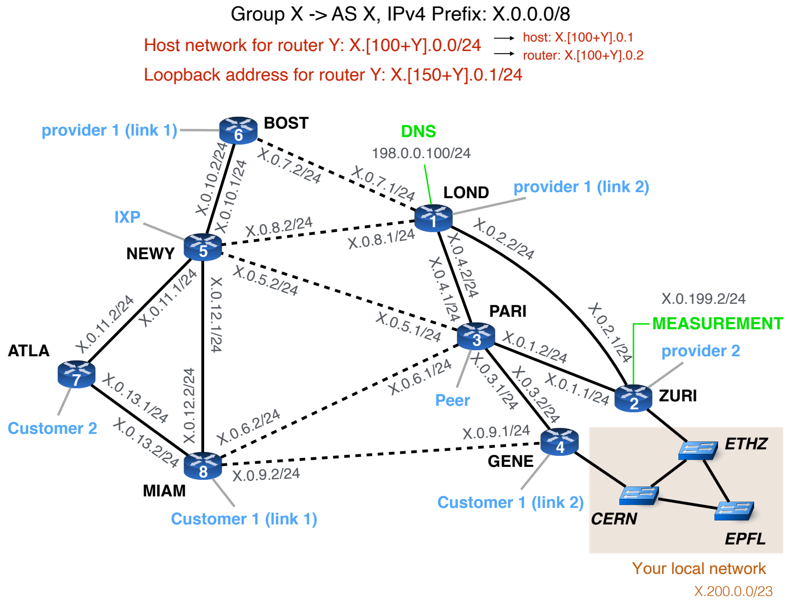

Figure 2: Each group will have to manage an entire AS. Your AS is composed of 8 routers. A /8 prefix has been assigned to each group. You can use it to configure your local networks. One host is also connected to each router, but ZURI and GENE. The subnets you must use are indicated on each interface. ZURI and GENE are connected to your local network.

L3 topology For this project, imagine that your layer-2 network is part of an AS spanning across the atlantic that you also manage. Your AS number is your group number: *e.g.,* AS 28 for group 28. Your AS has routers located on two continents: four routers in Europe (Geneva, London, Paris, and Zürich) and four in the US (Atlanta, Boston, New York, and Miami) see Figure 2.

Every AS has been allocated one /8 prefix that it can allocate internally. If you are group X, then the prefix X.0.0.0/8 is yours, meaning that group 48 has the prefix 48.0.0.0/8. You will use this IP space to allocate IP addresses to your hosts and routers.

Finally, one host is connected to each router with the exception of ZURI and GENE as these two routers are connected to your local network.

Internet topology Every router has an external connection to one of your neighboring ASes. Some are connected to a provider, some to a customer and others to a peer. NEWY is connected to an Internet eXchange Point (IXP). You will have to establish eBGP sessions on these external links. Figure 3 shows the mini-Internet topology you will end up building.

The red ASes (1, 2, etc.) are all Tier1 ASes, meaning their neighboring ASes are either peers or customers. The grey ASes (13, 14, etc.) are stub ASes, meaning their neighboring ASes are either peers or providers but they have no customers. We (the TA team) will take care of the Tier1 ASes as well as the stub ASes.

The Tier2 ASes (blue ASes) have peers, customers and providers. For example, group 5 has two providers (3 and 4), two peers (6 and the IXP 81) and two customers (7 and 8).

There are seven IXPs within our mini-Internet. The primary purpose of an IXP is to allow networks to interconnect directly. One advantage of using an IXP is that an AS can directly peer with another AS through the IXP, instead of reaching it via a provider that it has to pay. Another advantage is that only one physical connection with an IXP is needed to potentially interconnect with all the other IXP participants. An IXP uses a BGP Route Server to advertise prefixes between its participants.

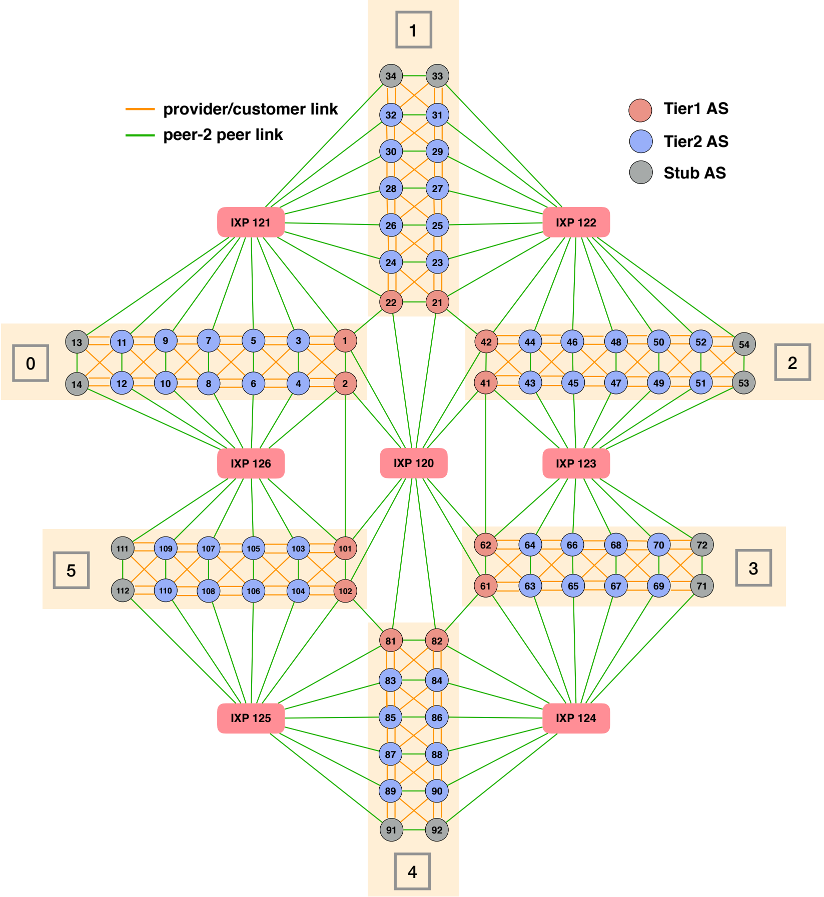

Figure 3: The AS-level topology of our mini-Internet. There are 12 Tier1 ASes, 12 stub ASes, and 54 Tier2 ASes. The topology is divided in 6 blocks (0, ..., 5), which are connected to each other via the Tier1 ASes or an IXP. The students operate the Tier2 ASes, while the Communication Networks TA team takes care of the Tier1 and the Stub ASes.

One IXP is connected to all the Tier1 ASes, allowing them to be connected in a full-mesh fashion. The other IXPs are always interconnecting two blocks. This enables these ASes to peer between them (as long as they respect the BGP customer/provider policies), instead of using (and paying!) their providers. The following example illustrates the benefit of being connected to an IXP: AS6 can send traffic to AS105 via the IXP126, instead of paying AS 4 to send the traffic via the path 4-2-101-103-105 if IXP126 is not used.

3 Questions

The assignment is split in two parts: (i) intra-domain, (ii) inter-domain and policy routing. In the first part you establish connectivity within your network (intra-domain). You will start by configuring the layer-2 network, followed by setting up your OSPF. The second part includes iBGP configuration as well as interconnecting all the networks (inter-domain). It involves bringing your eBGP sessions with your neighboring ASes up and advertising your prefixes. For the second you should also implement your BGP policies according to the business relationships that you have with your neighbors. As a last step, you will implement communication between the end hosts in your network using socket programming.

To help you, we will give you a crash course on how to configure FRRouting routers and Open vSwitches in a separate tutorial available in the project folder.

For each question, we precisely tell you what you must include in your report. In addition to your report, you must also send us your switch and router configurations. To make your life easier, we provide you a script named `save_configs.sh` in the main docker container that puts all the configurations (routers and switches) in a single directory. It also generates a zip out of the directory. Then, you just need to download the zipped directory, add your report (pdf), and send it to us.

3.1 Phase I (45% of your project grade)

Question 1.1 (5% of your project grade)

Your first task is to enable end-to-end connectivity within your own local network. For this reason, you need to configure an IP address as well as a default gateway on each host (student and staff). For this question, you must use IP addresses belonging to your local subnet, which is X.200.0.0/23 where X is your group number. You are free to use any IP address as long as it is in that subnet. To test connectivity, you can use `ping`.

In addition, every host needs to have a standard gateway to be able to reach external destinations. Configure it such that all hosts connected to CERN and EPFL use GENE as standard gateway and the hosts at ETHZ use ZURI.

To include in your report: In your report, include screenshots for the configuration in each host and router that you implemented. Explain what IP addresses you assigned to the different hosts. Finally, include 2 screenshots executing **ping** from student#1 to GENE router and from staff#2 to ZURI router.

Important: Please DO NOT remove any interface from routers, hosts or switches.

Question 1.2 (10% of your project grade)

As a network engineer, your goal for this question is to enable direct layer-2 connectivity between students, between staff members, but not in between them. Obviously, students and staff members should still be able to communicate between themselves, but via a layer-3 router. This will prevent typical layer-2 attacks such as MAC spoofing used to impersonate a type of user and get access to sensitive data.

For this reason, you have to configure VLANs: use VLAN 10 for the staff and VLAN 20 for the students. Please ignore VLAN 30. The interface of ZURI connected to ETHZ in VLAN 10 is named `ZURI-L2.10`, and the one in VLAN 20 is named `ZURI-L2.20` (you can see them with a `show interface brief` in the FRRouting CLI). The same holds for the interfaces in GENE: use `GENE-L2.10` for VLAN 10 and `GENE-L2.20` for VLAN 20. Do not use the interface `ZURI-L2` and `GENE-L2`.

Important: Similarly to previous question, all hosts connected to CERN and EPFL use GENE as standard gateway and the hosts at ETHZ use ZURI.

To include in your report: Explain what IP addresses you assigned to the different hosts and also explain possible changes that you made compared to the configuration presented in Question 1.1. Finally, show the output for one `traceroute` from EPFL-student to EPFL-staff, one from ETHZ-staff to EPFL-student and one from EPFL-student to ETHZ-staff. In a few sentences, explain what you observe.

Question 1.3 (10% of your project grade)

Configure OSPF network-wide by establishing OSPF adjacencies between neighboring routers. Then, make sure to advertise all your subnets into OSPF so as to enable end-to-end connectivity between all the hosts in your AS.

Before configuring OSPF, you will have to configure all the IP addresses on each interface of your routers and hosts. Unlike for Question 1.1, you must use the IP addresses shown in Figure 2. 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 group 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 group number (router IDs are shown on each router, for example the ID of BOST is 6). As an example, the loopback address of the router BOST for the group 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 group 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 group 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 `MIAMrouter` and uses the IP address 85.108.0.1/24.

Be sure that each host can ping its directly connected router. Then, you can start configuring OSPF.

Verify that the subnet of the DNS server and the measurement container are visible in OSPF (for instance with `show ip route ospf`). From now on, always prefer to launch `traceroute` from the hosts because they can use the DNS service (routers cannot). If one host cannot access the DNS server because the OSPF configuration is not ready yet, run `traceroute` with the option `-n` so that it does not try to translate each IP address found on the path.

Note: Do not modify the `dns`-interface on LOND, the `measurement`-interface on ZURI, and the `matrix`-interface on PARI.

To include in your report: Include the result of a `traceroute` from PARI-host to ATLA-host.

Question 1.4 (10% of your project grade)

As a network operator, your goal is now to provide the best performance to your customers. In this question, your goal is to minimize latency and prevent traffic congestion.

Your top priority is to minimize latency, and to do so you must configure OSPF weights such that the traffic never traverses two submarine links (dashed links in Figure 2). For example, you do not want the traffic from BOST to MIAM to pass through Europe, but to stay on the same continent. Then, to minimize congestion, you must configure the OSPF weights such that submarine paths with higher bandwidth are preferred whenever it is possible.

For the submarine links, there are four different bandwidth configurations as shown in Figure 4. In a first step, you need to identify the configuration your AS has by using `iperf3`. Once you have identified the bandwidth configuration, you can assign the weights such that the high bandwidth links are preferred.

In addition, you need to make sure that all traffic from MIAM to NEWY is loadbalanced on

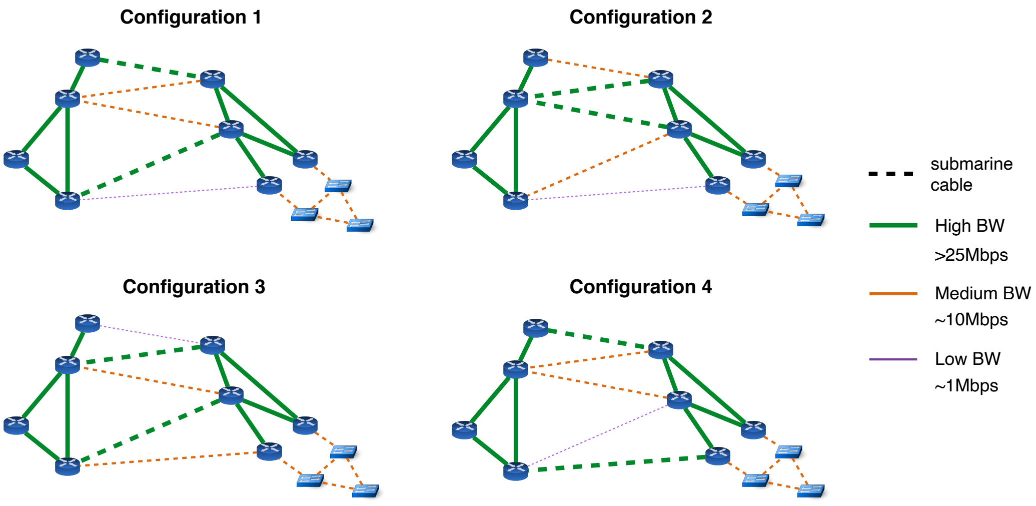

Figure 4: The bandwidth (BW) of your submarine cables follows one of the four depicted configurations.

the two paths MIAM-NEWY, and MIAM-ATLA-NEWY and the traffic from ZURI to LOND is loadbalanced on the two paths ZURI-LOND, and ZURI-PARI-LOND.

As a final requirement, you need to balance the traffic between ATLA and ZURI across the two submarine links with high bandwidth.

To include in your report: Include in your report your conclusions for the configuration your AS has by attaching screenshots of the `iperf3` output that show the configuration of your AS. Then, list all the OSPF weights you used. Finally, include the results of a `traceroute` from ATLA-host to the loopback interface of ZURI. Comment the results of your `traceroutes`: do you see what you expect according to the weights you have configured, why?

Question 1.5 (10% of your project grade)

As a network engineer, you should configure your AS in a way that traffic from ATLA host towards NEWY host should use the direct link between them and vice versa. For this step, you **should use the weight values using OSPF that you assigned in Question 1.4**. For security reasons, this policy should now change so that traffic from ATLA host towards NEWY host should pass via MIAM router and vice versa. In this case, you should use **static routes, keeping the existing weight values that you assigned before.**

To include in your report: Explain the weights that you assigned in order to establish the direct communication between the ATLA host and the NEWY host as well as a screenshot from a traceroute output for this use case. Additionally, describe the technique you used to redirect traffic (for the communication between ATLA host and NEWY host) via MIAM router as well as a screenshot for a traceroute for this communication and a screenshot from the output of **show ip route command in the MIAM router**.

3.2 Phase II (55% of your project grade)

Question 2.1 (5% of your project grade)

Configure internal BGP sessions (iBGP) between all pairs of routers (full-mesh). Verify that each one of your routers does have an iBGP session with all the other routers with the command `show ip bgp summary`.

When you establish a BGP session, you must use the loopback address for each endpoint of the connection. The loopback address is a virtual address that is always up as long as the router is running. Using the loopback interface instead of any other physical interface prevents the BGP session to go down if a physical interface becomes unavailable. To use loopback addresses for your BGP sessions, you will have to use the `update-source` command when you configure the internal BGP sessions. We explain why and how to configure it in our configuration tutorial.

To include in your report: Explain what `update-source` does and why you have to use it. Show the result of a `show ip bgp summary` for the router ATLA.

Question 2.2 (10% of your project grade)

Configure the external BGP sessions (eBGP) with your neighboring ASes (including the IXPs). Normally, you would need to negotiate with your neighboring ASes and agree on which IP addresses should be used by you and your peer during the hackathon. **This year, we provide you these IP addresses alongside the information about where and with whom you are supposed to have an eBGP session in the** *as connections.pdf* **file uploaded in the project folder.** For every eBGP session, the file shows its type (peer2peer, customer2provider or provider2customer), which router is connected to the neighboring AS, and what IP address (and subnet) you should use for the interface in your router. Table 1 is an example of what the file `as_connections` tells you if you are group 6 (AS 6).

6	ZURI	customer2provider	3	179.0.51.2/24
6	BOST	customer2provider	4	179.0.53.2/24
6	LOND	customer2provider	4	179.0.54.2/24
6	PARI	peer2peer	5	179.0.59.2/24
6	ATLA	provider2customer	7	179.0.62.1/24
6	GENE	provider2customer	8	179.0.61.1/24
6	MIAM	provider2customer	8	179.0.60.1/24
6	NEWY	peer2peer	126	180.126.0.6/24

Table 1: An example of what you can find in the file `as_connections`

Based on Table 1 we can see that AS 6 has two peers (AS5 and IXP126), two customers (AS7 and AS8) and two providers (AS3 and AS4). As an illustration, AS6 has two connections with AS4, one via its router BOST (where the IP address of the interface is 179.0.53.2/24) and one via its router LOND (where the IP address of the interface is 179.0.54.2/24). AS6 is connected to its customer AS7 via its router ATLA, and uses the IP address 179.0.62.1/24. The neighboring AS, AS 7 uses the IP address 179.0.62.2/24. This you can see when you look at the corresponding line of AS 7: `7 ZURI customer2provider 6 179.0.62.2/24`

AS6 is also connected to the IXP126 via its router NEWY. In this case, you must configure the IP address 180.126.0.6/24 on the interface of NEWY connected to the IXP. In our Internet, the AS number of an IXP is its identification number. For example, IXP126 has the AS number 126. The IP address of the IXP route server is 180.Z.0.Z with Z the IXP number. The route server of IXP126, for example, has the IP address 180.126.0.126.

Once the eBGP sessions are up, advertise your prefix to your peers. You must only advertise the /8 that has been assigned to you. Unfortunately, if you `redistribute ospf` routes into BGP, you will advertise all the /24 prefixes to your peers. In the mean time, your peers should advertise to you their /8 prefix, as well as all the /8 prefixes they have learned (since there are no BGP policies yet).

Measurement container We have setup a measurement container which will enable you to launch `traceroutes` from any **Tier2 AS** (and not necessarily only your own AS), towards any destination in the mini-Internet. This will help you to know the paths used *towards* your network. The measurement container is connected to each AS via the interface `measurement_X` of the router ZURI. The IP address of this interface is pre-configured and follows the convention X.0.199.1/24 (see Fig. 2), with X your group number. For example if you are group 15, your pre-configured IP address on the interface `measurement_15` at ZURI will be 15.0.199.1/24. The X.0.199.1/24 subnet must be reachable from anywhere in your network. You must therefore add it in your OSPF configuration. To access the measurement container, use the following command:

```
> ssh -p 2099 root@147.52.203.13
```

The password is f600459f32d4c2cd. To launch a traceroute, you can use the script `launch_traceroute.sh`, which takes as argument the group number from which the traceroute starts, and the destination IP address (possibly in another AS). For instance, if you want to perform a traceroute from group 11 to 22.107.0.1 (*i.e.,* the host connected to ATLA in group 22), just use the following command in the measurement container:

```
> ./launch_traceroute.sh 11 22.107.0.1
```

Note that the traceroute will start from the router ZURI of group 11, since the measurement container is connected to that router. In practice, network operators can use large-scale Internet measurement platforms such as RIPE Atlas[1] to assess the connectivity of their network from outside.

BGP looking glass We have setup a looking glass service. In practice, looking glasses are servers remotely accessible which display the routing information of an IP router. For example, SWITCH, the Swiss educational network, gives public access to its looking glass[2]. This is useful to see how your BGP advertisements look like from a remote point of view. For this assignment, we make publicly available a database containing information per group and per router showing the result of a `show ip bgp`.

Prerequisites: Before using the BGP looking glass service, initially you should install Python. For LINUX users, Python is already installed. For Windows users, detailed information for Python installation can be found in [3]. Python distributions can be downloaded from [4]. In our case, Python 3.7.7 is recommended. After installing Python, you should also install Psycopg library for the BGP looking glass service. Installation instructions are described in [5].
In order to access the information for the routers, you should use the Python script named `database-query.py`. For example, if you want to get the result of a `show ip bgp` at MIAM router for group 23, you should use the following command: `python database-query.py 23-MIAM`. The measurements are updated every 30 minutes for all ASes.

Hint: to answer this question, you will have to use the `next-hop-self` command when you configure the external BGP sessions. We explain why and how to configure it in the tutorial.

[1] https://atlas.ripe.net

[2] https://www.switch.ch/network/tools/lookingglass/

Reminder: the IP address of the IXP Route Server is 180.X.0.X with X the IXP number.

Note: to check whether a BGP session is working and a connection has been established, you can use the command `show ip bgp summary`. You will see a list of all BGP neighbors. If there is a time entry in the column `Up/Down`, then the session has successfully been established.

To include in your report: Explain what `next-hop-self` does and why you have to use it using an example in your own network. Also, explain on which BGP sessions `next-hop-self` is required. Then, show us the results of a `show ip bgp` for the router PARI. You should see the prefixes advertised by your neighboring ASes, which would indicate that your eBGP sessions are correctly configured and that the advertisements are correctly propagated through your iBGP sessions. Then, show us that your neighboring ASes do receive the advertisement for your /8 prefix. To do that, show in your report the result of the looking glass for one router located in a neighboring AS. You should see your prefix in the looking glass. Finally, show us that you have data-plane connectivity with your neighbors by showing the result of a `traceroute` from your PARI-host to the PARI-host of one of your neighboring ASes.

Question 2.3 (10% of your project grade)

By default, we have configured the IXPs to not relay your BGP advertisements to their other peers. To announce a prefix to another peer via an IXP, you must specify it using a BGP community value. IXPs are configured to relay a BGP advertisement to a peer X if the advertisement has a community value equal to N:X with N the IXP number. For example, if you are AS7 and you want to advertise a prefix to AS28 via the IXP121, you must add the community value 121:28 in your BGP advertisements.

Use the community values to send BGP advertisements to the peers connected to you through an IXP.

To include in your report: Take a screenshot of the relevant parts of the `out` route-map you configured on the session from the router in NEWY to the IXP. In a few sentences explain what all the lines in the route-map mean and do. Then, show a looking glass entry of another AS which proves that your prefix has been advertised through the IXP. Finally, use the measurement container to perform a `traceroute` from another AS (in another region) to your AS for a destination where the traffic should go through the IXP. Show the result in your report.

Question 2.4 (10% of your project grade)

Configure your local-preference as well as the exportation rules to implement the customer/provider and peer/peer business relationships with your neighbors [6]. The connections you have through your IXP must be considered as peer-to-peer connections.

Hint: To configure the exportation rules, you can tag incoming routes using BGP communities to keep track of where the routes have been learned, and then match on the tag when exporting the routes. We advice you to verify with `traceroutes` or with the looking glass that the paths used do respect the business relationships.

To include in your report: Briefly explain what BGP communities you used for your peers, customers and providers. Then show a screenshot of one `in` and one `out` route-map and briefly explain the different lines in the route-map. Then, show that your configuration works properly by adding the result of the looking glass of one of your peers, which is supposed to show that this peer does receive the prefixes of your customers, but does not receive the prefixes of your other peers. Finally, launch a `traceroute` from one of your customers towards one of your peers using the measurement container. Verify that your AS forwards the packet directly to your peer and not to your provider. Include the result of the `traceroute` in your report.

Question 2.5 (20% of your project grade)

In this question, students should implement their own monitoring tool in order to check network performance within their AS. As we know, 2 major indicators for characterizing network behavior is the latency and the number of hops between 2 nodes. In previous questions, `ping` and `traceroute` tools were used in order to study the communication between hosts. Also, `ping` provides information for the latency over a path between 2 hosts.

Now, students should create their custom monitoring mechanism for the latency and the number of intermediate routers (hops) among a path. In more details, the system should follow the client-server architecture. Assume, a host in your AS is interested in monitoring the latency and the number of hops for the paths connecting a group of hosts in the network. This host acts as a client in the application, triggering the hosts described in group of hosts to start running measurements about the latency and the number of hops between them as well as the sequence of intermediate routers that form the path for a pair of hosts, acting as servers. The measurement process should be executed by each possible pair of hosts in the list. The measurements should use `ping` and `traceroute` commands. When each host in the above list has accomplished the measurements for the other members, this information should be transmitted and stored in the client host. For the communication and data exchange between hosts, TCP sockets should be utilized. When the client host has received the measurements about the latency, the number of hops and the sequence of routers in terms of IPs for the path for each pair of hosts, the client has a detailed overview for the communication between the server hosts based on fundamental network indicators. For your application, Python should be used as programming language.

Based on the above description, your application in the client side should be executed as

`> python client.py` $[host_1, host_2, \cdots, host_N]$

where client.py is the client part of your application that should inform the server hosts described in the list $[host_1, host_2, \cdots, host_N]$ in order to start executing measurements about the latency, the number of hops and the routers that form the path for each possible pair between them. **The list $[host_1, host_2, \cdots, host_N]$ contains the hosts that act as servers with their host names (not their IPs)**. For this reason, you should use a configuration file in .txt form that includes the IPs that you assigned to each host and each router interface. Your client script should initially read this configuration file and translate each host name in the list $[host_1, host_2, \cdots, host_N]$ to the corresponding IP assigned to each host. So the initial list $[host_1, host_2, \cdots, host_N]$ should be translated to the list $[IP_1, IP_2, \cdots, IP_N]$, where IP_i is the IP assigned to $host_i$. The above list including host IPs should be transmitted to the server hosts for the measurement process. When each server has finished the measurements, the results should be transmitted to the client host, that stores this information. When the client receives the traceroute output from a server, each IP described as hop in the traceroute should be also translated to the corresponding router name. This is feasible using the configuration file. So finally, for each pair of hosts in the initial list, the client should store **the latency**, **the number of hops** and the **full description of the path in terms of router names** between the 2 hosts. On the other hand, the hosts described in the list, act as servers that start executing measurements using ping and traceroute tools for the other hosts in the list when they are triggered by the client. Also, as soon as each measurement is accomplished, they transmit the required information to the client. The server functionality should be implemented in the server.py file.

For submission: During the submission, students should submit client.py and server.py files that implement the communication between the client and the group of servers. Also, in your report include an example of your application as well as screenshots with the results.

4 Acknowledgments

The assignment is based on material from a similar course at ETH Zurich [7] and the corresponding platform [8].

References

[1] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, "The design and implementation of open vswitch," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15).* Oakland, CA: USENIX Association, 2015, pp. 117–130. [Online]. Available: https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/pfaff

[2] FRRouting. [Online]. Available: https://frrouting.org/

[3] Install Python 3 and PIP on Windows 10. [Online]. Available: https://www.youtube.com/watch?v=gFNApsyhpKk

[4] Python3. [Online]. Available: https://www.python.org/downloads/release/python-377/

[5] Psycopg. [Online]. Available: https://pypi.org/project/psycopg2/

[6] L. Gao and J. Rexford, "Stable internet routing without global coordination," *SIGMETRICS Perform. Eval. Rev.*, vol. 28, no. 1, pp. 307–317, Jun. 2000.

[7] ETH Zurich, Communication Networks. [Online]. Available: https://comm-net.ethz.ch/

[8] T. Holterbach, T. Bühler, T. Rellstab, and L. Vanbever, "An open platform to teach how the internet practically works," *arXiv preprint arXiv:1912.02031*, 2019.