Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Networked Systems
ETH Zürich — seit 2015

Spring 2019 Prof. L. Vanbever/ T. Bühler, R. Birkner, T. Holterbach, R. Meier

Communication Networks

Project 1: Build your own Internet

Deadline: April 19 2019 at 11.59pm

1 Introduction

In this assignment, you will learn how to build and operate a layer-2 and layer-3 network and how different networks, managed by different organizations, interconnect with each other.

More particularly, you will first learn how to configure a local network composed of several layer-2 switches using the Spanning Tree Protocol (STP). Then, you will learn how to set-up a valid forwarding state within an Autonomous System (AS) by configuring layer-3 routers. To do that, you will use an intra-domain routing protocol: OSPF. Finally, you will learn how to set-up a valid forwarding state between different ASes so that an end-host in one AS (e.g. Swisscom) can communicate with an end-host in another AS (e.g. Google). For this task you will use the only inter-domain routing protocol available today: BGP. We will build a mini Internet using virtual switches running Open vSwitch [1] and virtual routers running the Quagga software routing suite [2].

Traditional layer-2 switches and layer-3 routers can be configured through a Command Line Interface (CLI). Each switch and router vendor (e.g. Cisco or Juniper) or software routing suite (e.g. Quagga) has its own CLI. Fortunately, those CLIs are similar, and if you know how to configure a router using the Quagga CLI, you can easily configure a Cisco or a Juniper router as well. Unlike Quagga, Open vSwitch does not provide a CLI but a set of commands that are very similar to the one available in a traditional layer-2 switch. As a result, if you can configure a Quagga router and an Open vSwitch, you can easily configure a Cisco or a Juniper switch or router as well.

In a separate document[1], we give you a crash course on how to configure a Quagga router and an Open vSwitch. The rest of this document is organized as follow. Section 2 first describes the setup you will have to use. Section 3 explains how you can access and configure your Open vSwitches and Quagga routers and Section 4 presents verification tools. Section 5 then lists the actual questions you have to answer, while Section 6 answers Frequently Asked Questions (FAQ). Finally, Section 7 provides general information about the project, including **submission instructions**.

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[2] Your layer-2 network is composed of four switches (Fig. 1) located at four different locations: ETH Zentrum, Irchel, Hönggerberg and Oerlikon. Each link has a 10Gb/s capacity

[1] available at `http://comm-net.ethz.ch/routing_project/quagga_ovs_tuto.pdf`

[2] This is not the actual topology used by the ICT-Networks team at ETH `https://www.ethz.ch/en/the-eth-zurich/organisation/departments/informatikdienste.html`

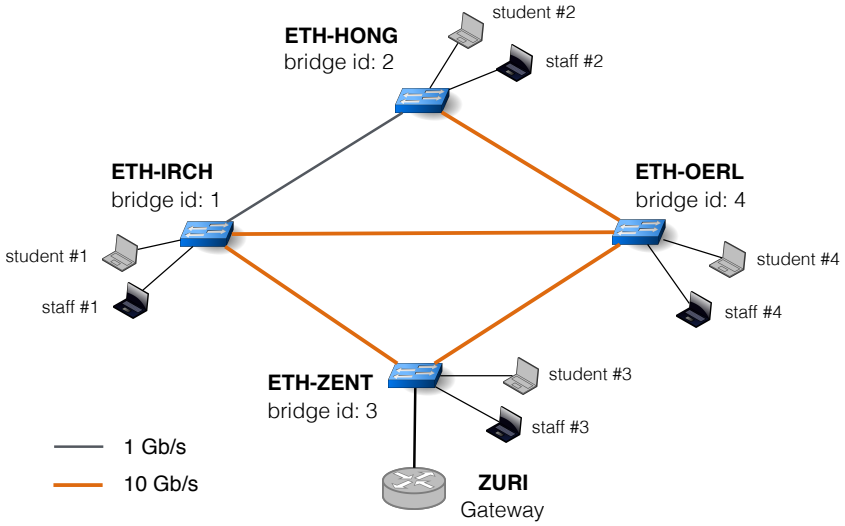

Figure 1: Each group will have to manage its own local ETH network. This local network is composed of four Open vSwitches located at different locations within ETH. The switch at ETH-ZENT is connected to a layer-3 router which acts as the gateway.

except the link between ETH-IRCH and ETH-HONG which has a 1Gb/s capacity. The switch at ETH-ZENT is connected to a layer-3 router which acts as a gateway, meaning that a host in the local network must send a packet to this router to reach any non-local destination. The router will then take care of sending that packet to the destination.

Two types of user 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, ETH-HONG has bridge ID 2.

L3 topology For this project, imagine that your layer-2 network is part of the Geant AS[3] that you also manage[4]. Your AS number is your group number: *e.g.,* AS 28 for group 28. Your AS has routers located in four continents: four routers in Europe (London, Zürich, Barcelona and Roma), two in the US (New York and Houston), one in Japan (Tokyo) and one in Africa (Abidjan), see Figure 2. The links connecting the routers have a capacity of 100, 10 or 1 Gb/s. For example, there is a 100Gb/s link between LOND and ZURI while there is only a 1Gb/s link between BARC and ABID. Each AS has been allocated with 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 22 has the prefix 22.0.0.0/8. You will use this IP space to allocate IP addresses to your hosts and routers. Regarding the latter, the subnets to use are indicated in Figure 2. For example, between HOUS and LOND, you must use the subnet X.0.6.0/24. The interface in LOND that is connected to HOUS must have the IP address X.0.6.1 and the interface in HOUS that is connected to LOND must have the IP address X.0.6.2 (with X your group number).

Each router also has a loopback address that you will configure. The router with ID Y has the loopback address X.[200+Y].0.1/24 where X is your group number (router IDs are shown on each router, for example the ID of TOKY is 7). As an example, the loopback address of the router ROMA for the group 10 is 10.204.0.1/24.

Finally, one host is connected to each router. Between each router and its host, you will 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 will have 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. As an example, if you are group 15 and you want to configure the host connected to NEWY, it will have the IP address 15.105.0.1/24 and the interface of the NEWY router connected to this host will have the IP address 15.105.0.2/24. The router ZURI is the gateway of the ETH local network that we have

[3] `https://www.geant.org`

[4] In this project we use a simplified network topology

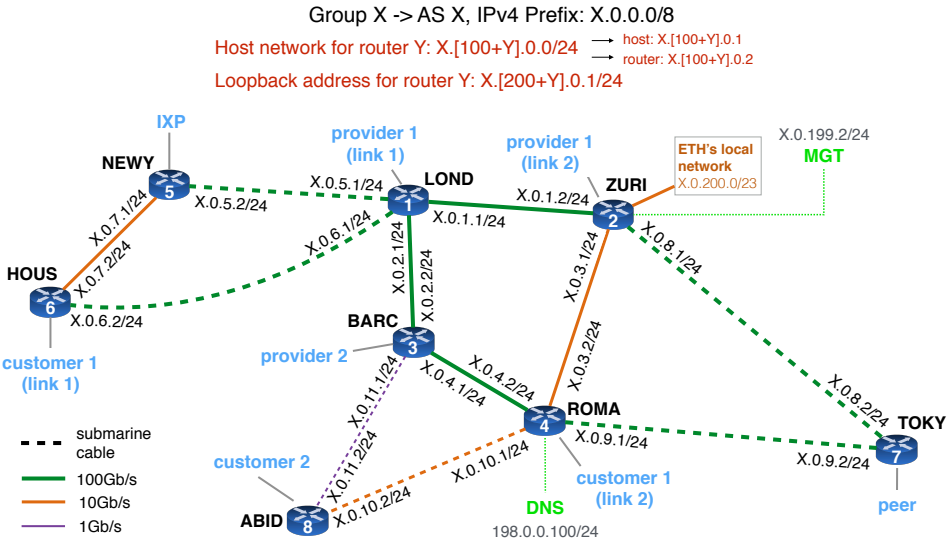

Figure 2: Each group will have to manage an entire AS. Your AS is composed of 8 routers. A /8 prefix has been assigned to each group. You can use it to configure your local networks. One host is also connected to each router. The subnets you must use for each of your local networks are indicated on each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

described in the previous subsection. The subnet X.0.200.0/23 is dedicated to this local network.

Internet topology Each router has an external connection to one of your neighboring ASes. LOND and ZURI are connected to the same provider whereas BARC is connected to a different provider. HOUS and ROMA are connected to the same customer and ABID is connected to a different one. TOKY is connected to a peer, and 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.

Group 1, 2, 11, 12, 21, 22, 31, 32, 41, 42, 51 and 52 are Tier1 ASes, meaning their neighboring ASes are either peers or customers. Group 9, 10, 19, 20, 29, 30, 39, 40, 49, 50, 59 and 60 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 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). The file `as_connections`, available at `http://comm-net.ethz.ch/routing_project/as_connections`, shows all the connections between ASes. For each connection, it shows its type (peer, customer or provider), which router is connected to the neighboring AS, and what subnet should be used between the two ASes. Table 1 is an example of what the file `as_connections` tells you if you are group 6 (AS 6).

Tier2	Prov1(link1)	6	4	LOND	179.24.52.0/24
Tier2	Prov1(link2)	6	4	ZURI	179.24.53.0/24
Tier2	Prov2	6	3	BARC	179.24.50.0/24
Tier2	Cust1(link1)	6	8	HOUS	179.24.59.0/24
Tier2	Cust1(link2)	6	8	ROMA	179.24.60.0/24
Tier2	Cust2	6	7	ABID	179.24.61.0/24
Tier2	Peer1	6	5	TOKY	179.24.58.0/24
Tier2	IXPOut	6	86	NEWY	180.86.0.6/24

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

Based on Table 1 we can see that AS 6 is a Tier2. It has two peers (AS5 and IXP86), two customers (AS7 and AS8) and two providers (AS3 and AS4). As an illustration, AS6 has two connections with AS4, one via its router LOND (where the subnet must be 179.24.52.0/24) and one via its router ZURI (where the subnet must be 179.24.53.0/24). AS6 is connected to AS7 via

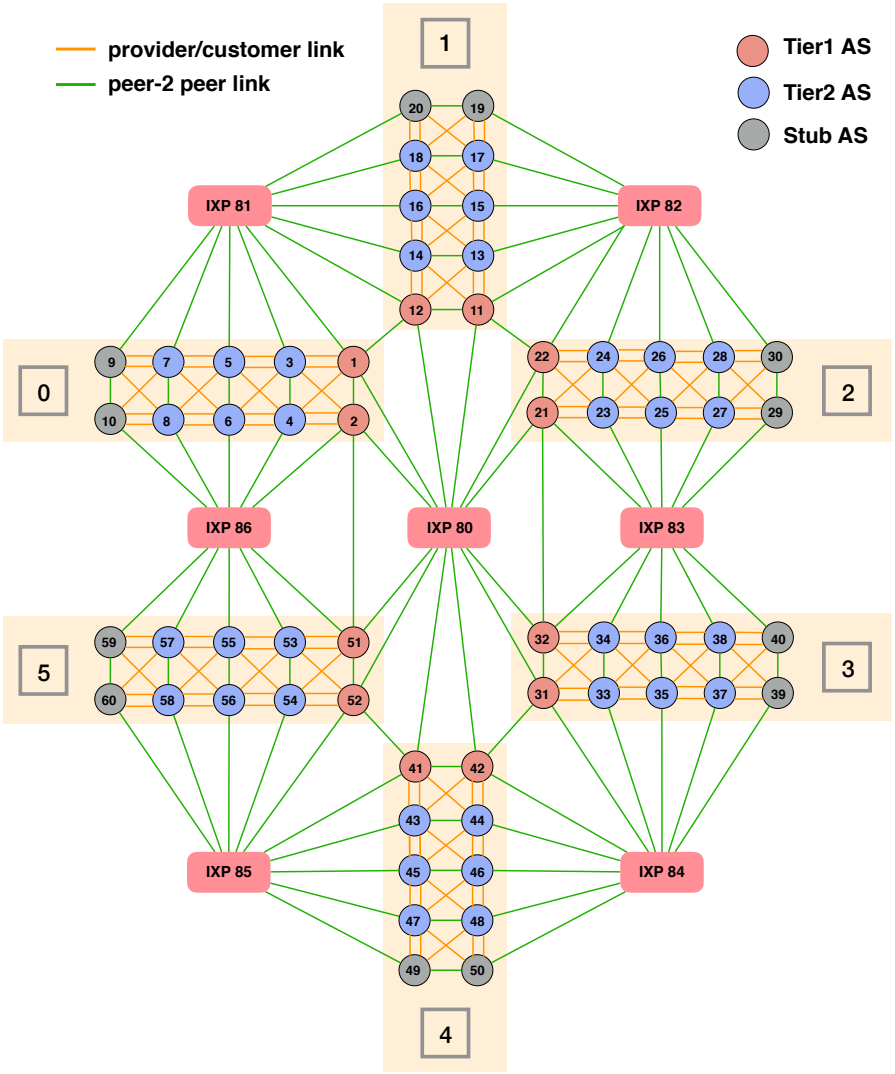

Figure 3: The AS-level topology of our mini-Internet. There are 12 Tier1 ASes, 12 stub ASes, and 36 Tier2 ASes. The topology is divided in 6 blocks (0, ..., 5), which are connected to each other via the Tier1 ASes or an IXP. Students will operate the Tier2 ASes, while the Communication Networks TA team will take care of the Tier1 ASes as well as the Stub ASes.

its router ABID, and the subnet 179.24.61.0/24 must be used. During the Internet Hackathon, you will have to talk with your neighboring ASes to decide who takes which IP address in the subnet between you and your neighboring AS. AS6 is also connected to the IXP86 via its router NEWY. In this case, you must configure the IP address 180.86.0.6/24 on the interface of NEWY connected to the IXP. If you are connected to a Stub AS, the file tells you the IP address (and not just the subnet) you have to configure.

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. In our Internet, the AS number of an IXP is its identification number. For example, IXP82 has the AS number 82. The IP address of the IXP Route Server is 180.X.0.X with X the IXP number. For example the Route Server of IXP83 has the IP address 180.83.0.83. One IXP is connected to all the Tier1 ASes, allowing them to be connected in a full-mesh fashion. Each other IXP is connected to ten ASes. For example, IXP81 is connected to

AS 1, 3, 5, 7, 9, 12, 14, 16, 18, 20. 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: AS7 can send traffic to AS18 via the IXP81, instead of paying AS 5 to send the traffic via the path 7-5-3-1-12-14-16-18 if IXP81 is not used.

3 Configure your network

In this section, we show you how you can access your Quagga routers and Open vSwitches to configure them.

Accessing your Virtual Machine (VM). To make your life easier, you do not need to run the virtual network inside your laptop. Instead, we have setup a remote VM for each group where your virtual network already runs and desperately awaits you to configure it. We have also setup our server so that each AS is connected to its neighboring ASes only. To access your VM, you will use SSH. SSH is a UNIX based command interface and protocol for securely getting access to a remote computer. It is widely used by system administrators to control network devices and servers remotely. An SSH client is available by default on any Linux and MAC installation through the Terminal application. For Windows user, a good and free SSH client is PuTTY.[5] Once you have installed a SSH client, use the following command to connect yourself to your VM:

```
> ssh -p X root@samichlaus.ethz.ch
```

where $X = 2000 + group\_number$. For instance if you are group 7, use the following command:

```
> ssh -p 2007 root@samichlaus.ethz.ch
```

We have sent you your password and group number by email. If you cannot connect to your VM, please report it immediately in the slack channel `#routing_project`. If you want to simplify the access to your VM (optional), please use SSH key authentication,[6] but do not change your password. If you want to download an entire directory (e.g. the `configs` directory) from your VM to the current directory of your own machine, you can use scp:

```
> scp -r -P X root@samichlaus.ethz.ch:~/path_to_the_directory .
```

where $X = 2000 + group\_number$. On Windows, you can use WinSCP[7] to do that. Note the dot at the end of the command and the capitalized `P`.

If you use `tmux` [3], a terminal multiplexer, please do not use the session *mininext*. The virtual network is running in this session. If you need to reboot your VM, please contact Thomas Holterbach (`@thomas_holterbach` on Slack).

Accessing your routers, switches and hosts When you are in your VM, you can use the script `go_to.sh` to connect to a router, a switch or a host. For example, with the following command, you will access the router NEWY:

```
> ./go_to.sh NEWY
```

Once connected to NEWY, you can access its Quagga CLI using the following command.

```
NEWY> vtysh
```

Once you are in the CLI of a router, you can see its interfaces (along with its current configuration) with the command `show run`. In the case of the NEWY router, the interface connected to LOND is named `lond`, the interface connected to HOUS is named `hous`, the interface connected to the host is named `host` and the interface which is connected to another AS is named `ebgp_peer`. Use `exit` to leave the CLI, and another `exit` to leave the router.

[5] `https://www.chiark.greenend.org.uk/~sgtatham/putty/`
[6] Ask us or Google it if you want to know more about this.
[7] `https://winscp.net/eng/docs/start`

Similarly, you can access a switch with the `go_to.sh` script. For example, the following command shows you how to access the switch ETH-IRCH:

```
> ./go_to.sh ETH-IRCH
```

Unlike Quagga, there is no CLI with Open vSwitch, thus `vtysh` will not work. Directly run the Open vSwitch commands in the terminal. You can find more information regarding Quagga CLI and Open vSwitch commands in the tutorial file.[8]

From your VM, you can also go to a host. For example, if you want to go to the host which is connected to ABID, just use the following command:

```
> ./go_to.sh ABID-host
```

When you are in the host, you can use `ifconfig` to see the interface which is connected to the ABID router. In this case, the name of the interface is `abid`. If you want to access the student host connected to ETH-ZENT in the ETH local network, you can use the following command:

```
> ./go_to.sh ETH-ZENT-student
```

If you want to access a staff host, just replace student by staff.

Maintaining your configuration files A quagga router either boots with a fresh configuration or with a configuration previously saved in a file. Routers configuration can then be modified through the CLI, or by modifiying these files directly and reloading the routers on it. Once connected to the CLI of a Quagga router, you can save its running configuration into the configuration file using the following command:

```
router# write file
```

The configuration files are named `Quagga.conf` and are located in the `configs` directory (one for each router). If you want to save the configuration of all your routers, we provide you a script named `save_quagga_configs.sh`, which automatically runs the `write file` on every router.

Important: We advice you to configure your routers using the CLI as much as possible. However, in some situations you may need to reload a previously saved configuration. To load the configuration that are in the config files into the Quagga router, go into that router (not the CLI), and use the command `reload`. This will reload the Quagga router, which will boot with the configuration file previously saved. Be careful though, this will erase the configuration that was used by the router just before the `reload`. To prevent accidental loss of your current configuration because you used `reload` by mistake, we always save you current configuration in the directory `/tmp` when you run `reload`. We recommend you to regularly save your configuration (the directory `configs`) into your own machine, as your configuration will be reset should we have to reboot your VM. In case of a reboot, you can quickly put back your configurations in the routers by using the `reload` command or just copy/pasting your configurations into the Quagga CLI.

4 Testing and verifying your network configurations

As any network operator, you must verify that your configuration does what you want, and debug it in case something goes wrong. We offer you several tools that you can use to verify your configuration. These tools are similar to the ones network operators use in practice.

Management VM We have setup a management VM which will enable you to launch `traceroutes` from any 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 management VM is connected to each AS via the interface `mgt` 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 `mgt` at

[8] available at `http://comm-net.ethz.ch/routing_project/quagga_ovs_tuto.pdf`

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 management VM, use the following command:

```
> ssh -p 2099 students@samichlaus.ethz.ch
```

The password will be available in the `#routing_project` Slack channel. 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 TOKY in group 22), just use the following command in the management VM:

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

Note that the traceroute will start from the router ZURI of group 11, since the management VM is connected to that router. In practice, network operators can use large-scale Internet measurement platforms such as RIPE Atlas[9] to assess the connectivity of their network from outside.

DNS service To help you decoding your traceroute output, we have setup a DNS server and have pre-configured your hosts to use it. If the DNS server is used, the IP addresses in the traceroute output will be replaced by the corresponding router names. For example, 19.0.1.2 will be translated into ZURI-lond.group19, because this is the IP address configured on the interface `lond` of the router ZURI in AS19. The DNS server is located in a VM connected to the interface `dns` of the router ROMA. The IP address on that interface is pre-configured, you do not need to modify it. Also, each host is pre-configured to use the DNS server. Only hosts use the DNS server, routers do not. Of course, hosts can only use the DNS server if they can reach the network 198.0.0.0/24 (where the DNS server is located). As such, do not forget to include this network in your OSPF configuration.

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[10]. This is useful to see how your BGP advertisements look like from a remote point of view. For this assignment, we make publicly available on our website (under the `looking_glass` directory) one file per group and per router showing the result of a `show ip bgp`. The files for group X are in the directory GX, there exists one file for each router. For example, if you want to get the result of a `show ip bgp` at ROMA for group 23, download the file `http://comm-net.ethz.ch/routing_project/looking_glass/G23/ROMA`[11]. These files are updated every minute.

Connectivity matrix We also provide you with a connectivity matrix, which shows you whether two ASes can ping between each other. Before you setup the eBGP sessions, everything will be red, meaning you can't communicate with another group. During the Internet Hackathon, as soon as you will setup the eBGP sessions with your neighbors, the matrix will turn green for some pairs of ASes. At the end of this assignment, we hope to see this matrix completely green! The matrix is available at `http://comm-net.ethz.ch/routing_project/matrix/matrix.html` and is updated every minute.

5 Questions

The assignment is split in three parts with the class-wide "Internet Hackathon"[12] being the middle one. The first part must be finished **before** the Internet Hackathon. It involves configuring the layer-2 network and setting up your OSPF and iBGP configuration. The second part will be done **during** the Hackathon. It involves bringing your eBGP sessions up with your neighboring ASes and advertising your prefixes to your neighbors and the IXPs. The third and last part

[9] `https://atlas.ripe.net`
[10] `https://www.switch.ch/network/tools/lookingglass/`
[11] The same file ending with `.txt` is also available so that you can open it with your browser.
[12] April 4, 2019 at 6pm

will be done **after** the Hackathon. It involves implementing your BGP policies according to the business relationships that you have with your neighbors. In addition to these three parts, there is also one bonus question related to BGP security. To help you, we give you a crash course on how to configure Quagga routers and Open vSwitches in a separate document available at `http://comm-net.ethz.ch/routing_project/quagga_ovs_tuto.pdf`

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 `generate_submission_folder.sh` in the VM that puts all the configurations (routers and switches) in a single directory. Then, you just need to download this directory, add your report (pdf) inside, generate a .zip or .tar.gz archive, and send it to us. Before creating the submission folder, you probably want to save the configuration of all your routers with the script `save_quagga_configs.sh`.

5.1 Before the Hackathon (3.75 points)

Question 1.1 (1.25 point)

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.

To enable end-to-end connectivity, you will 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 the ETH subnet, which is X.0.200.0/23 with X your group number. You are free to use any IP address as long as it is in the ETH subnet. To test connectivity, you can use `ping`.

You also have to configure VLANs. To help you, we have already pre-configured the link between ETH-ZENT and the router ZURI to be a trunk link for VLANs 10 and 20. The interface of ZURI connected to ETH-ZENT in VLAN 10 is named `eth-zent.10`, and the one in VLAN 20 is named `eth-zent.20` (you can see them with a `show run` in the Quagga CLI). Do not use the interface `eth-zent`.

Note: the links between the switches are by default trunk links for VLAN 0. VLAN 0 is the default VLAN, *i.e.,* a packet that appears to have no VLAN is part of VLAN 0.

To include in your report: Explain what IP addresses you assigned to the different hosts. Finally, show the output of a `traceroute` from ETH-OERL-student to ETH-OERL-staff.

Question 1.2 (0.5 point)

Given that the cost of the links is the same by default, what is the spanning tree obtained before configuring anything?

Note: Open vSwitch does not support per-VLAN spanning-trees *i.e.,* the same spanning tree is used for both VLAN 10 and 20.

Now, without any a priori knowledge about the traffic, your goal is to configure the bridge ids and/or the path costs so as to use the links with higher bandwidth. This allows the network to minimize the chance of having traffic congestion.
Just as a reminder, with the Spanning Tree Protocol, you can modify the switch priority (*i.e.* bridge id) to define the root bridge, and you can modify the cost of the paths in order to deactivate a set of links and obtain the spanning tree you want. Similarly to OSPF, the path with the lowest cost is always preferred. When a port is blocked, it means that the switch does not

forward traffic from/to it.

To include in your report: Draw the spanning tree before configuring anything, and the spanning tree obtained after applying your configuration. Then, explain in few sentences why you chose this spanning tree and how you configured it (what bridge ids and/or path costs you used). In addition, include the output of the command `ovs-ofctl show SWITCH_NAME` for ETH-IRCH and ETH-OERL.

Question 1.3 (0.5 point)

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 in each interface of your routers and hosts. In each host, you will also have to configure a default gateway. Unlike for Question 1.2, you must use the IP addresses shown in Figure 2. Be sure that each host can ping its directly connected router. Then, you can start configuring OSPF.

Verify that you can reach the DNS server and the management VM from any host in your network (for instance with `ping`). From now on, always prefer to launch `traceroute` from the hosts because they can use the DNS service (routers do not). 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.

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

Question 1.4 (0.5 point)

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 NEWY to HOUS to pass through Europe, but to stay on the same continent. Then, to minimize congestion, you must *(i)* configure OSPF weights such that paths with higher capacity are preferred whenever it is possible (*e.g.,* traffic from LOND to ABID should use the path LOND-BARC-ROMA-ABID), and *(ii)* load-balance the traffic on the two paths LOND-ZURI-TOKY and LOND-BARC-ROMA-TOKY for the traffic going from LOND to TOKY.

To include in your report: List all the OSPF weights you used. Then, include the results of a `traceroute` from LOND-host to ABID-host. Then, run multiple `traceroutes` from LOND-host to TOKY-host but include only the result for one of them. Comment the results of your `traceroutes`: do you see what you expect to see according to the weights you have configured, why?

Question 1.5 (0.5 point)

The traffic going from LOND to the host connected to ROMA is critical and needs a very low latency. As a result, your goal in this question is to use the path LOND-ZURI-ROMA for the traffic destined to the host connected to ROMA. The rest of the traffic (for instance the traffic going from LOND and destined to ABID) should still use the path LOND-BARC-ROMA-ABID.

To include in your report: Explain what technique you used to achieve the result and discuss potential drawbacks of the solution. Then, show the result of a `traceroute` from LOND to the

host connected to ROMA and a `traceroute` from LOND to the host connected to ABID.

Question 1.6 (0.5 point)

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

When you establish a BGP session, you must use to 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 a loopback address instead of an interface address 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 quagga 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 ROMA.

5.2 During the Hackathon (1 point)

Question 2.1 (0.5 point)

Configure the external BGP sessions (eBGP) with your neighboring ASes (including the IXPs). You must negotiate with your neighboring ASes and agree on which IP addresses should be used by you and your peer. The information about where and with whom you are supposed to have an eBGP session is available at `http://comm-net.ethz.ch/routing_project/as_connections`. 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. Initially, your routers won't let you advertise your /8 prefix, as a router does not advertise an unreachable prefix (and this is the case with your routers, as they only know how to reach a few /24 prefixes belonging to your /8). To force your routers to advertise your /8, you must configure a static route to this prefix, and set the next hop to null. At 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).

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

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 ZURI. 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 ZURI-host to the ZURI-host of one of your neighboring ASes.

Question 2.2 (0.5 point)

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 AS2 and you want to advertise a prefix to AS21 via the IXP80, you must add the community value 80:21 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 management VM 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.

5.3 After the Hackathon (5.25 points + 0.5 bonus points)

Question 3.1 (2 points)

Configure your local-preference as well as the exportation rules to implement the customer/provider and peer/peer business relationships with your neighbors [4]. 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 management VM. 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 3.2 (1 point)

The AS topology (Fig.3) shows six main regions (0, 1, 2, 3, 4, and 5). Configure your BGP policies such that you can leverage your connection with your IXP at NEWY. You do want to peer through this IXP with ASes that are located in another region. However, for business reasons, you do **not** want to peer through this IXP with ASes that are located in the same region. To not peer through the IXP with ASes in the same region, you must (i) not advertise them any prefixes and (ii) deny any advertisements coming from them. Explain your configuration in the report.

To check whether you properly configured (ii), we have configured the stubs ASes to advertise their prefix to all the ASes connected to their IXP.

To include in your report: Again take a screenshot of the relevant parts of the route-map at NEWY and explain what the different lines mean and do. Show that the advertisement from the stub AS in the same region as you and connected to the same IXP as you is denied by showing the result of a `sh ip bgp` in your router NEWY. For clarity, you do not need to write the full output, just the part that is interesting (*i.e.,* the part which could have the prefix of the stub AS). Then, include in the report the output of the looking glass for the router NEWY of the

stub AS in the same region than you and connected to the same IXP than you. Finally, include in your report the output of the looking glass for a group in another region but connected to the IXP. When you include the output of a looking glass in your report, only keep the parts that prove the correctness of your configuration and omit the irrelevant ones.

Question 3.3 (0.75 point)

In this question, the goal will be to configure your BGP policies in order to influence the **inbound** traffic destined to your **own** prefix. More precisely, your goal is to configure BGP policies such that the inbound traffic coming from a provider and destined to your own prefix uses the provider connected to BARC in priority.

To include in your report: Explain in a few sentences the technique you used and discuss any potential drawbacks. Then, include the result of the looking glass for both of your providers. You can omit parts of the output that are irrelevant, and only show the part that shows that your configuration is correct (*i.e.,* the part where your own prefix is shown).

Question 3.4 (0.75 point)

In this question, the goal will be to configure your BGP policies such that the **inbound** traffic coming from the provider with whom you have two external BGP sessions (*e.g.,* AS 15 if you are AS 17) arrives in priority via your router ZURI (instead of LOND).

To include in your report: Explain in a few sentences the technique you used and discuss any potential drawbacks. Then, include the result of the looking glass of your provider with whom you have two external BGP sessions which shows that your configuration is working as expected (for instance from the router HOUS). You can omit parts of the output that are irrelevant.

Question 3.5 (0.75 point)

In this question, the goal is to configure your BGP policies in order to influence the **outbound** traffic (*i.e.,* how the traffic exits your network) such that the traffic destined to the prefix owned by the Tier 1 AS that is located in the same region and on the same side (even/odd) as your AS (*e.g.,* AS 41 if you are AS 45) exits your AS in priority via your router BARC. For the rest of the traffic, your provider connected to LOND and ZURI must always be preferred over the provider connected in BARC.

To include in your report: Explain in a few sentences the technique you used. Then, include the output of `sh ip bgp` for the router ROMA. With this output, you should see the BGP path used to reach each prefix, and you can confirm that they are correct. Finally, include the result of two `traceroutes`, one targeting the prefix owned by the Tier1 in the same region and on the same side as your AS, and one targeting the prefix owned by the Tier1 in the same region but on the other side than your AS (*e.g.,* AS 42 if you are AS 45).

Bonus question 3.6 (0.5 point)

For research purposes, your goal is now to modify your BGP configuration such that the traffic destined to your /8 prefix never traverses the Tier1 AS that is in the region following your region clock-wise and on the same side (*e.g.,* AS 21 if you are AS 15).

To include in your report: Explain how you achieved this and discuss any potential drawbacks of the approach. Then, include the result of a `traceroute` launched from another group (for instance group 23 if you need to avoid AS 21) and destined to your /8 prefix before and after you have modified your configuration. The `traceroute` launched after your modifications should show that the traffic does not traverse AS 21 anymore.

6 Frequently Asked Questions

I have a problem when running a traceroute. It takes quite some time and the DNS service does not work.

When you run a `traceroute`, keep in mind that only the hosts connected to the routers are configured to use the DNS service. If you run a `traceroute` from a router, it will not translate the IP addresses. If the DNS service is not reachable (*e.g.,* because you have not configured OSPF yet), run the `traceroute` with the option `-n`. This will tell `traceroute` to not translate the IP addresses, and it will save you some time.

How can I erase a configuration I have done on a Quagga router?

To erase a configuration on a Quagga router, it's very easy, just use the same command you used, but add "`no`" at the beginning. For example, if you configured an IP address with `ip address 1.0.0.1/24`, just run `no ip address 1.0.0.1/24` to erase it.

I have correctly configured an IP address on one interface of my router. I can see it when I do a `sh run`, but I can't ping any other IP address in the network.

Sometimes, when you configure something wrong on one interface or forget to erase a previous configuration, a desynchronization between the quagga configuration and the linux interface can happen. In other words, you will see your configuration on quagga when you do a `sh run`, but the configuration is not used by the linux interface. To find this problem, you can do an `ifconfig` on the host running the router, and check whether the IP address configured matches the one in the quagga configuration. If not, you need to synchronize quagga with this linux interface. To do that, first erase the IP address configuration on the linux interface with `ifconfig INTERFACE_NAME 0.0.0.0`. Then erase the IP address(es) configured in the quagga router for this interface with `no ip address ...`. After that, you can start configuring the interface through Quagga again, it should be synchronized with linux.

I have the error `"VTY configuration is locked by other VTY"` when I run a `conf t` on a Quagga router.

Only one VTY session can configure a Quagga router at a time. If you have this problem, it's either because one member of your group is already configuring this router, or because VTY sessions are still running in the background and block your access to the router (for example because you lost a previous ssh connection). In the latter case, you can kill the vtysh sessions running by first getting their `pid` with `ps aux | grep vtysh`, and then killing them with `kill -9 pid`.

My neighboring ASes are not active and I can't get connectivity with the rest of the mini-Internet because of that.

Although each group has two providers, two customers and two peers, it can happen that some of them are inactive or have misconfigured BGP, which makes you unreachable from some parts of the mini-Internet. This can also prevent you to test your configuration or run `ping` or `traceroute`. If you experience this problem when you answer the questions, please describe it in your report and explain what you were not able to do because of it. If this is really a big problem (*e.g.,* both of your providers are inactive and you can't reach the rest of the mini-Internet), please let us know and we will find a way to solve the problem. If you could not show something in your report due to failures of your neighbors, it will not negatively influence your grade!

7 General Information

7.1 If you have questions: use Slack or visit one of the exercise sessions

Use the Slack channel available at `comm-net19.slack.com`. Please do not ask questions in the `#general` channel, but use the `#routing_project` channel instead.

7.2 Submit your work by e-mail

Send your final report and configuration by email to Laurent Vanbever (`lvanbever@ethz.ch`), Thomas Holterbach (`thomahol@ethz.ch`) and Tobias Bühler (`buehlert@ethz.ch`). Make sure that your email includes a zip or tar.gz archive containing a PDF report as well as all your configuration files (the directory `final_configs` generated with the `generate_submission_folder.sh` script). Please make sure that your PDF report includes your group number as well as the name of the members composing your group. The maximum length for your PDF report is 10 A4 pages (including screenshots, traceroutes, looking glass etc.). You can always remove the parts from the screenshots, looking glass entries, etc. which are not needed to answer the question or demonstrate the correct functionioning of your configuration.

Important: the subject of your email must follow this format: **[comm_net] groupX project 1**, where X is your group number.

7.3 Grading

This assignment will be graded and counts as 10% towards your final Communication Networks grade. There are a maximum of 10 points (plus half a bonus point). Each group member will receive the same grade: $\min\{1 + \frac{\sum pts}{2}, 6\}$

7.4 Academic integrity

We adopt a strict zero tolerance policy when it comes to cheating. Cheating will immediately translate to the group failing the assignment and being reported to ETH administration. 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.

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] Quagga Routing Suite. [Online]. Available: http://www.nongnu.org/quagga/

[3] Tmux, a terminal multiplexer. [Online]. Available: https://tmux.github.io

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