Communication Networks
Spring 2022

Routing Project
Q/A session
March 31, 2022
Today’s schedule

1. More detailed introduction on the routing project

2. Introduction to Git

3. How to minimize the size of a forwarding table (Exam question)
Today’s schedule

1. More detailed introduction on the routing project

2. Introduction to Git

3. How to minimize the size of a forwarding table (Exam question)
The project counts for 20% of your final grade
- There is a total of 10 points (+2 bonus questions)

You can ask questions
- During the Q/A sessions
- On Slack, in the #routing_project channel
- Maybe your question is in the FAQ already

Use GitLab to submit your work
- Submit your routers and switches configuration
- Submit your report (max 10 pages!)
- Sign the declaration of originality and submit it

We wrote a tutorial where we give useful information
- Including how you can access
Your AS-level topology
This is where your mini-Internet is running
Routing project timetable

Week 1
- Q1.1 - 1.8 Intra-domain routing

Week 2
- Q2.1 - 2.3 Inter-domain routing

Week 3
- Q3.1 - 3.4 Routing security
- Easter break

Week 4
- 22.04 Q3.2 active
- 26.04 Q3.3 active

Week 5
- 29.04 Project end

- 28.03 Project start
- 31.03 Today
- 07.04 Connectivity Fest
- 14.04 Q3.1 active
- 22.04 Q3.2 active
- 26.04 Q3.3 active
Question 1.1: Enabling connectivity in the North Data Center
Question 1.1: Enabling connectivity in the North Data Center

Where to configure an IP address and a default gateway
Question 1.1: Enabling connectivity in the North Data Center

Where to configure an IP address and a default gateway

Where to configure an IP address for each VLAN
Question 1.1: Enabling connectivity in the North Data Center

Where to configure an IP address and a default gateway

Where to configure an IP address for each VLAN

Where to configure the VLANs
Question 1.2: Enabling connectivity in your IP network

High bandwidth links
Low bandwidth links
Question 1.2: Enabling connectivity in your IP network

Where to configure an IP address and a default gateway
Question 1.2: Enabling connectivity in your IP network

Where to configure an IP address and a default gateway

Where to configure IP addresses and routing protocols
General advice: do not forget to use the debugging tools

Linux networking tools
- Ping and traceroute to verify connectivity and IP paths
- Tcpdump to sniff packets on an interface

Routers and switches debugging commands
- You can show the current config, the content of the routing table, etc
- You can see information about each protocol

Monitoring tools we provide (and document in the Wiki)
- Connectivity matrix
- BGP looking glass
- BGP policy analyser
- Measurement container

https://duvel.ethz.ch
General advice: do not forget to use the debugging tools

Linux networking tools
- Ping and traceroute to verify connectivity and IP paths
- Tcpdump to sniff packets on an interface

Routers and switches debugging commands
- You can show the current config, the content of the routing table, etc
- You can see information about each protocol

Monitoring tools we provide (and document in the Wiki)
- Connectivity matrix
- BGP looking glass
- BGP policy analyser
- Measurement container

https://duvel.ethz.ch
We are here to guide you through the project
do not hesitate to ask questions! :-)}
Today’s schedule

1. More detailed introduction on the routing project

2. Introduction to Git

3. How to minimize the size of a forwarding table (Exam question)
- gsn routing porject
+ nsg routing project
git organizes **code in repositories**, usually hosted online

`gitlab.ethz.ch`
setup: **use ssh keys for security and convenience**

1. Add your public key under preferences > SSH keys
2. Clone repositories using ssh (not https) the ssh urls look like `git@gitlab.ethz.ch:...`

![Clone with SSH](image_url)
every set of changes to the code is called a commit

gitlab.ethz.ch
to create commits, **clone the repository** to get a local copy

```bash
$ git clone <repository>
```

**gitlab.ethz.ch**
if you have cloned in the past, you **pull to get up-to-date**

gitlab.ethz.ch

git **pull**
    updates local copy
all code changes and **commits are created locally**

codecodecodecodecodecodecode…

**git status**

*shows all changed (and staged) files*

**git add <all relevant changes>**

*stages files to be committed*

**git commit**

*creates a local commit with all added changes*
finished **commits need to be pushed** online

gitlab.ethz.ch

```bash
git push
uploads commits
```
if someone else pushed before you, **git prevents pushing**

**git push**
fails if not up-to-date!
you first need to **solve conflicts locally**

1. **git pull**
gets updates. you need to **merge**, i.e. make sure all fits together

2. **git push**

**gitlab.ethz.ch**
Today’s schedule

1. More detailed introduction on the routing project

2. Introduction to Git

3. How to minimize the size of a forwarding table (Exam question)
Task 5.3: Summer Pruning (Exam Question 2018)

Goal:

- Simplify a forwarding table, i.e., least number of entries
- Forwarding behaviour should be equivalent to initial table

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.130.32.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.64.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.80.0/20</td>
<td>2</td>
</tr>
<tr>
<td>82.130.96.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.112.0/21</td>
<td>1</td>
</tr>
<tr>
<td>82.130.120.0/21</td>
<td>1</td>
</tr>
<tr>
<td>82.130.122.0/24</td>
<td>1</td>
</tr>
<tr>
<td>82.130.123.0/24</td>
<td>1</td>
</tr>
<tr>
<td>82.130.124.0/24</td>
<td>2</td>
</tr>
</tbody>
</table>
Task 5.3: Summer Pruning (Exam Question 2018)

Goal:

- Simplify a forwarding table i.e., least number of entries
- Forwarding behaviour should be equivalent to initial table

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.130.32.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.64.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.80.0/20</td>
<td>2</td>
</tr>
<tr>
<td>82.130.96.0/20</td>
<td>1</td>
</tr>
<tr>
<td>82.130.112.0/21</td>
<td>1</td>
</tr>
<tr>
<td>82.130.120.0/21</td>
<td>1</td>
</tr>
<tr>
<td>82.130.122.0/24</td>
<td>1</td>
</tr>
<tr>
<td>82.130.123.0/24</td>
<td>1</td>
</tr>
<tr>
<td>82.130.124.0/24</td>
<td>2</td>
</tr>
</tbody>
</table>

A lot of students had trouble with this question
Task 5.3: Solution intuition

Two main simplification approaches:
- Replace multiple prefixes with a larger one
- Exploit longest prefix match behaviour

Possible pitfalls:
- Some IPs are forwarded to a different next-hop
- The new table covers more IPs than the old one
Task 5.3: Drawing a prefix tree can help you

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.0.0/20</td>
<td>1</td>
</tr>
<tr>
<td>1.2.16.0/20</td>
<td>2</td>
</tr>
<tr>
<td>1.2.32.0/20</td>
<td>1</td>
</tr>
<tr>
<td>1.2.48.0/20</td>
<td>1</td>
</tr>
</tbody>
</table>
Task 5.3: Drawing a prefix tree can help you

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.0.0/20</td>
<td>1</td>
</tr>
<tr>
<td>1.2.16.0/20</td>
<td>2</td>
</tr>
<tr>
<td>1.2.32.0/20</td>
<td>1</td>
</tr>
<tr>
<td>1.2.48.0/20</td>
<td>1</td>
</tr>
</tbody>
</table>
Task 5.3: Drawing a prefix tree can help you
Task 5.3: Drawing a prefix tree can help you understand the longest prefix match. The prefix tree below shows the following prefixes:

- 1.2.0.0/18
- 1.2.16.0/20
- 1.2.32.0/20
- 1.2.48.0/20

The root of the tree is 1.2.0.0 with a prefix length of 18. The tree is structured such that:

- Prefix 1.2.0.0/18 is matched at the root.
- Prefix 1.2.16.0/20 is matched at the next level with a prefix length of 19.
- Prefix 1.2.32.0/20 is matched at the next level with a prefix length of 20.
- Prefix 1.2.48.0/20 is matched at the next level with a prefix length of 21.

It works because of the longest prefix match! The prefix 1.2.0.0 has priority over the other prefixes.