Communication Networks
Spring 2020

Rüdiger Birkner, TA
Thomas Holterbach, TA
Noah Hütter, TA
Eric Marty, TA

https://comm-net.ethz.ch/

ETH Zürich
April 23 2020
Communication Networks

Exercise 10

- Wrap-up of the routing project
- Intro to the reliable transport project
- Intro to Python and Git
- Current assignment
Communication Networks

Exercise 10

Wrap-up of the routing project

Intro to the reliable transport project

Intro to Python and Git

Current assignment
Communication Networks 2020
How we build a mini-Internet

Thomas Holterbach
https://comm-net.ethz.ch/

ETH Zurich (D-ITET)
April 23, 2020
You did it: 100% connectivity!

Jonas Mehr  Apr 8th at 9:51 AM
We did it!
2020-04-08-105159_1017x1155_scroll.png

1 reply

Also sent to the channel

Laurent Vanbeyer  1d ago
Very impressive 😊 And, as far as I can remember, in now over 5 iterations of the routing project, this is a first! Good job everyone!
Many of you managed to solve the bonus question!

Killian 11:37 PM
Hello everyone, there is a surprise GIF at 49.200.30.30 waiting for you 🎉

Congratulations
You did it!
You completed 95% of the challenge. 100% will win your XPR. 😎

Time to preview your profile information. Take a look at these new, fun, and unique features! 🎈

Congratulations! You've completed the challenge. Take a look at the new features!

Special thanks to Prof. You're not only an accomplished coder but also a true genius! 🎓

83.200.30.58 works like a charm!!!
Today, we will see how we designed the mini-Internet
We rely on virtualisation

Option #1: virtual machines

Option #2: linux containers
We rely on virtualisation

Option #1: virtual machines
we used VMs between 2016 and 2019

Option #2: linux containers
We rely on virtualisation

Option #1: virtual machines
we used VMs between 2016 and 2019

Option #2: linux containers
This year
Each router, switch and host runs in its dedicated container.
Each router, switch and host runs in its dedicated container
We virtually connect the containers to build the mini-Internet
We use additional containers for the different monitoring services
We use additional containers for the different monitoring services
We use "proxy" containers so that you can only access your virtual devices.

```
ssh -p 2001 root@snowball
```
We use "proxy" containers so that you can only access your virtual devices

```
ssh -p 2001 root@snowball
```

network interface

SSH Proxy
Group 1

Group 1
- host 1
- switch 1
- Router 1

- host n
- switch k
- Router p

SSH Proxy
Group 2

Group X
- host 1
- switch 1
- Router 1

- host n
- switch k
- Router p

ping results uploaded

snowball.ethz.ch

SSH port forwarding

docker container

virtual link

dNS

MEASUREMENT

MATRIX

MEASUREMENT

MEASUREMENT
We use "proxy" containers so that you can only access your virtual devices.

```
ssh -p 2001 root@snowball
```

SSH port forwarding

```
ssh -L local_port:remote_ip:remote_port user@remote_ip
```
We use "proxy" containers so that you can only access your virtual devices.

ssh -p 2001 root@snowball

ssh -L local_port:remote_ip:remote_port user@remote_ip

2001
We use "proxy" containers so that you can only access your virtual devices.

```
ssh -p 2001 root@snowball
```

We use "proxy" containers so that you can only access your virtual devices.

```
ssh -L local_port:remote_ip:remote_port user@remote_ip
```

The IP of SSH Proxy Group 1 is:

```
snowball.ethz.ch
```
We use "proxy" containers so that you can only access your virtual devices.

```
ssh -p 2001 root@snowball
```

SSH port forwarding

```
ssh -L local_port:remote_ip:remote_port user@remote_ip
```

22
We use "proxy" containers so that you can only access your virtual devices.

We use ssh with the command `ssh -p 2001 root@snowball` for SSH port forwarding.

We can also use the command `ssh -L local_port:remote_ip:remote_port user@remote_ip` to create a virtual link.
We use "proxy" containers so that you can only access your virtual devices.
We use "proxy" containers so that you can only access your virtual devices.
We use "proxy" containers so that you can only access your virtual devices

```
ssh -p 2001 root@snowball
```

```
openvpn --config client.conf
```

```
docker container
```

```
virtual link
```

```
dns
```

```
measurement
```

```
matrix
```

```
ssh (goto.sh)
```

```
routing
```

```
ping results uploaded
```

```
snowball.ethz.ch
```

```
Group 1
```

```
Group X
```

```
Router 1
```

```
Router p
```

```
host 1
```

```
switch 1
```

```
switch k
```

```
host n
```

```
switch 1
```

```
switch k
```

```
Router 1
```

```
Router p
```

```
ssh proxy Group 1
```

```
ssh proxy Group 2
```

```
VPN
```

```
network interface
```

```
SSH port forwarding
```

```
SSH (goto.sh)
```

```
docker container
```

```
virtual link
```

```
dns
```

```
measurement
```

```
matrix
```

```
routing
```

```
ping results uploaded
```

```
snowball.ethz.ch
```

```
Group 1
```

```
Group X
```

```
Router 1
```

```
Router p
```

```
host 1
```

```
switch 1
```

```
switch k
```

```
host n
```

```
switch 1
```

```
switch k
```

```
Router 1
```

```
Router p
```

```
ssh proxy Group 1
```

```
ssh proxy Group 2
```

```
VPN
```

```
network interface
```

```
SSH port forwarding
```

```
SSH (goto.sh)
```

```
docker container
```

```
virtual link
```

```
dns
```

```
measurement
```

```
matrix
```

```
routing
```

```
ping results uploaded
```

```
snowball.ethz.ch
```
Our server can easily run a 78-ASes mini-Internet
For further information about the mini-Internet:  mini-inter.net

Open source implementation
~3700 lines of bash
An Open Platform to Teach How the Internet Practically Works

Thomas Holthaus, ETH Zurich
Tobias Bühler, ETH Zurich
Tino Rellstab, ETH Zurich
Laurent Vanbever, ETH Zurich

ABSTRACT
Each year, ETH Zurich around 20 students voluntarily build and operate their very own Internet infrastructure composed of hundreds of routers and devices based on Open Source Software (OSS) technologies.

We developed a class-wide mini-project to enable students to learn about Internet operations and services. Students can see live traffic and device statistics, which make it possible to learn about Internet operations together with the corresponding protocols.

This course concept is novel and unique as most Internet courses do not include hands-on Internet operations. The fact that all students operate their own ASes has given us a unique opportunity to deeply understand how the Internet really works.

In this paper, we describe the overall design of our teaching project: clearly the fact that all of them need to cooperate for the Internet infrastructure to work. Students from different backgrounds can cooperate with each other to troubleshoot Internet-wide problems; and...
Every year we improve the project
Every year we improve the project

We still too often observe such incorrect paths
Every year we improve the project

We still too often observe such incorrect paths

We are designing a visualisation framework to help students detecting those incorrect paths
Every year we improve the project

Although you eliminated the hijacker very well…
Every year we improve the project

Although you eliminated the hijacker very well… …It is always better to prevent a hijack before it actually happens
Every year we improve the project

Although you eliminated the hijacker very well…

…It is always better to prevent a hijack before it actually happens

We plan to implement the RPKI infrastructure into the mini-internet so that you can validate the origin of the BGP routes

We offer this as a semester thesis, check out our website for further information!
Please let us know if you have any feedback or ideas on how to improve the project :-}
Communication Networks

Exercise 10

Wrap-up of the routing project

Intro to the reliable transport project

Intro to Python and Git

Current assignment
Implement your own Reliable Transport Protocol
Implement your own Reliable Transport Protocol
recovery from packet loss and reordering
Implement your own Reliable Transport Protocol

Lossy Link

Sender

Receiver
Implement your own Reliable Transport Protocol

Input

Sender

Lossy Link

Data

Receiver
Implement your own Reliable Transport Protocol
Implement your own Reliable Transport Protocol
Implement your own Reliable Transport Protocol
The Go-Back-N Protocol
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

Sender

Receiver
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

cumulative ACKs
make up for losses

Sender

0 1 2 3 4 5 6 7 8 9

Receiver

0 1 2 3
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

When a timeout occurs, the sender retransmits all segments in the window.
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions

When a timeout occurs, the sender retransmits all segments in the window.
The Go-Back-N Protocol

a simple reliable transport protocol with
a sliding window, cumulative ACKs, timeouts and retransmissions
Reliable Transport Project Assignment

Part 1
Simple Go-Back-N implementation
Retransmit all packets after a timeout

Part 2
Support for Selective Repeat
Fast retransmission after repeated ACKs

Part 3
Support for Selective Acknowledgements (SACK)
SACK contains blocks of correctly received segments

Bonus
Congestion Control
Don’t worry, we provide you with a code skeleton
As always:

Ask your questions on Slack (#transport_project) or during the exercise and Q&A sessions
Communication Networks

Exercise 10

Wrap-up of the routing project

Intro to the reliable transport project

Intro to Python and Git

Current assignment
Python Development
And a bit of Git
Contents

• Python
• Integrated Development Environments (IDEs)
• Version Control with Git
Python

https://xkcd.com/353/
Python
Installation

Windows

realpython.com/installing-python

Ubuntu

sudo apt-get install python3.7 python3-pip \ python3.7-venv

Mac OSX

brew install python
Python

Getting started

$ python3.7
>>> print("Hello World!")
Hello World!

Familiarise yourself with Python before you start.

Beginners Guide
learnpython.org

Advanced/Refresh Guide
learnxinyminutes.com/docs/python3

8 - 1  # => 7
10 * 2  # => 20
35 / 5  # => 7.0

# Integer division rounds down for both positive and negative numbers.
5 // 3  # => 1
-5 // 3  # => -2
5.0 // 3.0  # => 1.0  # works on floats too
-5.0 // 3.0  # => -2.0
Python
From Text Editors to IDEs

Text Editors

- Sublime Text
- atom.io
- vim

Integrated Development Environments

- JetBrains PyCharm
- Visual Studio Code
Python
From Text Editors to IDEs

- Free, no registration required
- Text Editor only
- All platforms

Sublime Text
Python
From Text Editors to IDEs

• Free, open source community edition
• More sophisticated features, such as a debugger
• All platforms
# Python

## Virtual Environments (Advanced topic)

Helps to use correct Python version and packages.

https://realpython.com/python-virtual-environments-a-primer/

```bash
# Create virtual environment (only done to setup)
$ python3.7 -m venv venv
# Activate the environment
$ source venv/bin/activate
# python executable is now used from environment
(venv) $ which python
/Users/noah/venv/bin/python
# Install packages with pip
(venv) $ pip install numpy
# Create list of packages
(venv) $ pip freeze > requirements.txt
# Install all packages from requirements file
(venv) $ pip install -r requirements.txt
# Deactivate environment
(venv) $ deactivate
```
Python
Virtual Environments (Advanced topic)
Supported by PyCharm
Git
Version Control

https://xkcd.com/1597/
**Git**

**Tracks Changes in your Code**

Without git

Every collaborator has its own version of the files, merging is manual, going back in time is not possible.

With git

File changes are tracked, merging is assisted, history can be accessed (and much more)
Git

Setup

1. Create a repository for your group

https://gitlab.ethz.ch/projects/new

You can set the visibility to private (only group members).
2. Invite group members

Settings → Members

Set role to developer so they can push to non-protected branches, the master branch is protected.
3. Create SSH key and add it to Gitlab

https://docs.gitlab.com/ee/gitlab-basics/create-your-ssh-keys.html

$ ssh-keygen

This allows you to access the repository from the console.
4. Upload the project files to Gitlab

Go to the repository (Projects → repository name) and follow the instructions for *Push an existing folder*. 

```bash
cd existing_folder
git init
git remote add origin git@gitlab.ethz.ch:huettern/git-demo.git
git add .
git commit -m "Initial commit"
git push -u origin master
```
5. Download the repository to your local machine

This way you can work on your machine without VM connection

$ git clone <link_to_repo>
Git Workflow

Start

**git pull**
Get most recent version

<do work>

**git push**
Upload changes to Gitlab

**git commit**
Store changes

**git add .**
Select files to update
# Download latest changes from Gitlab
$ git pull
# Do work on files...
$ vim main.py
# Show what has changed
$ git status
# Add the files you want to update
$ git add main.py
# Store changes in history with a short description
$ git commit -m "very important bug fix"
# Upload the changes to Gitlab
$ git push
Git Workflow
Git
Tips and Tricks

• No branching required for the assignment
• Run the git commands from the correct directory
• Always pull before you push

Cheat Sheet & Installation Guide

rogerdudler.github.io/git-guide
Communication Networks
Exercise 10

Wrap-up of the routing project

Intro to the reliable transport project

Intro to Python and Git

Current assignment
Task 1: Reliable Transport

Analyze a Go-Back-N transfer of 10 segments (10’000 bits) on a 10Mbps link with a 100ms propagation delay with and without loss.
Communication Networks

Exercise 10

Wrap-up of the routing project

Intro to the reliable transport project

Intro to Python and Git

Current assignment