# Software Defined Networking

## Lab Work 2 Introduction

Jeremias Blendin, Leonhard Nobach, Christian Koch,
Julius Rückert, Matthias Wichtlhuber

Peer-to-Peer Systems
Engineering Lab (PS)

PS - Peer-to-Peer Systems Engineering Lab
Dept. of Electrical Engineering and Information Technology
Technische Universität Darmstadt
Rundeturmstr. 12, D-64283 Darmstadt, Germany
http://www.ps.tu-darmstadt.de/

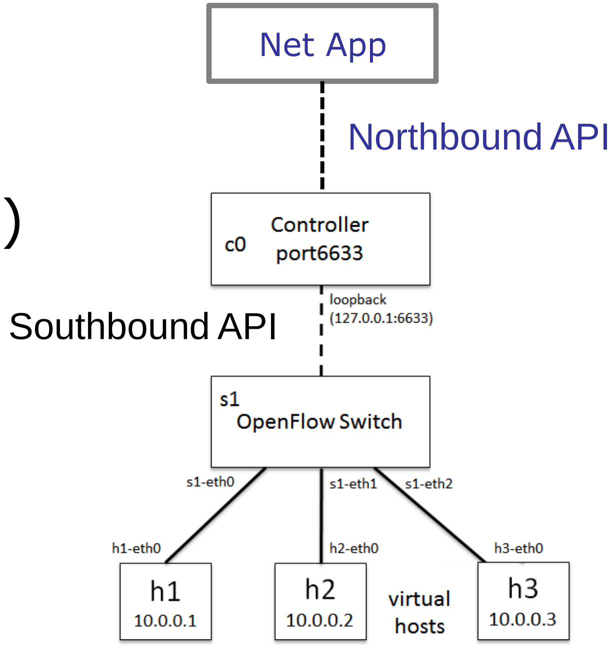
24. October 2014

# INTRODUCTION TO

# OpenFlow Controller / RYU

# Short Recap

- Previously we manually added rules in the switch
  - dpctl add-flow tcp:127.0.0.1:6634\
    in_port=1,idle_timeout=0,actions=output:2
- This should be done automatically
  - Task of a Network Application (NetApp)
  - E.g. a simple switching NetApp



[1] http://sdnhub.org/resources/useful-mininet-setups/

# Installing RYU

- Reboot your existing Mininet VM and enter:
  - sudo -s
  - apt-get install python-eventlet python-routes python-webob python-paramiko python-pip python-dev libxml2-dev libxslt-dev zlib1g-dev
  - pip install ryu
  - pip install six==1.8.0
  - mn -c

# Run a simple_switch

- Enter:
  - mn --topo single,3 --mac --arp --switch ovsk\
    --controller=remote,ip=127.0.0.1
  - h1 ping h2

- Open a second terminal and connect to the VM
- Copy the example app to your VM
  https://github.com/osrg/ryu/blob/master/ryu/app/simple_switch.py
- Execute ryu-manager ./simple_switch.py
- Now the ping from terminal 1 succeeds

# Understand how it works

- A step-by-step explanation can be found here
  - http://osrg.github.io/ryu-book/en/html/switching_hub.html
  - Read it carefully!
- Other resources like books and tutorials available
  - E.g. http://books.google.de/books?id=JC3rAgAAQBAJ

# Task 1: Port Mirroring

- Modify the *simple_switch* in a way that all received ICMP packets are sent through both *out_ports* of the switch. The packet should not be sent back to the port from where it originated.
- A ping from h1 to h2 should result in a ping from h1 to h2 and h3. As a result, h1 receives more packets then it has sent.
- Use Openflow 1.0 (like *simple_switch*)
- This can be used as a basis to allow more sophisticated network services like “lawful interception”