# Software Defined Networking

## Lab Work 5 – Load Balancing with OpenFlow

TECHNISCHE UNIVERSITÄT DARMSTADT

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


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

# Lab 5: Overview and Getting Started

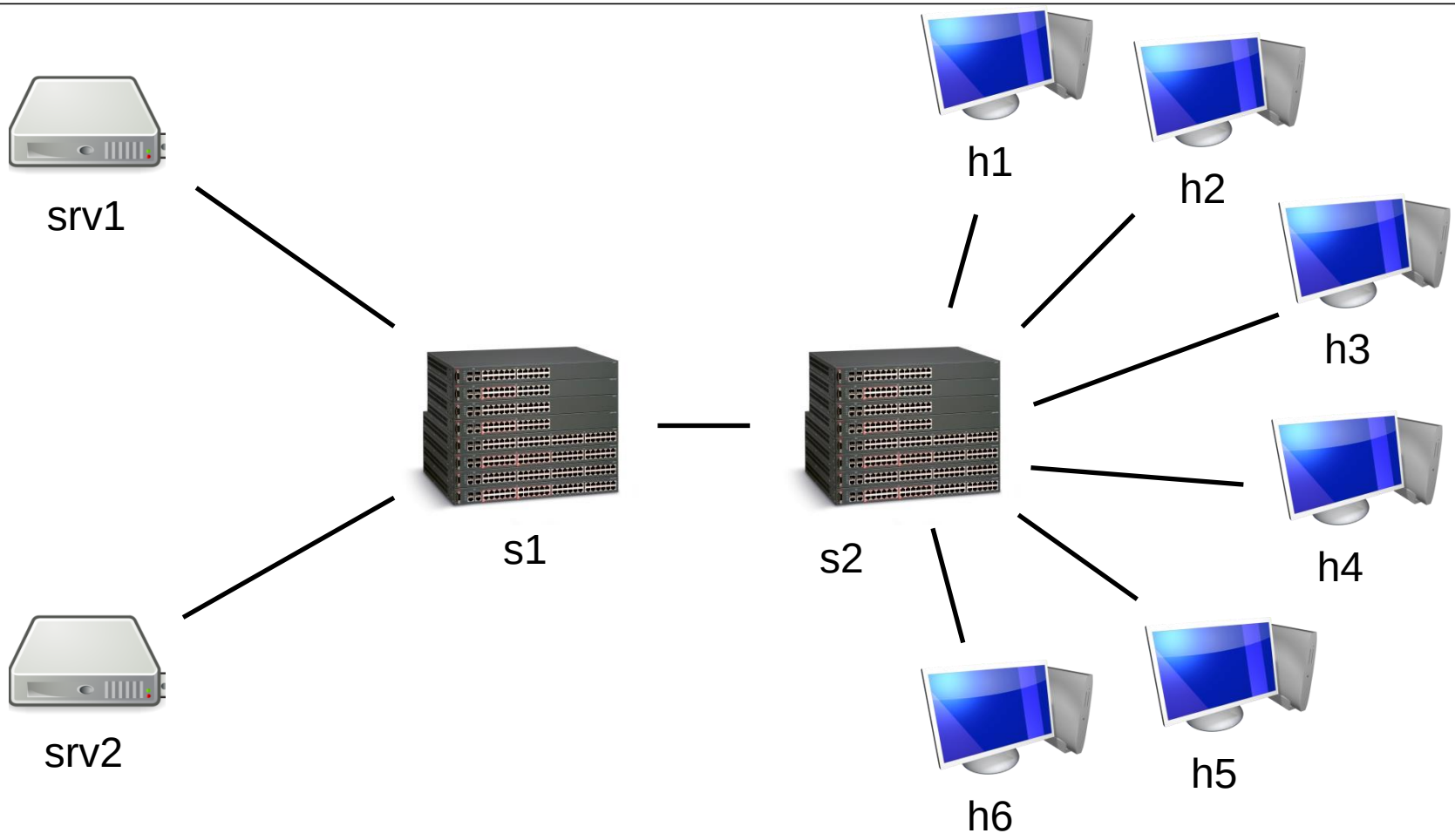
TECHNISCHE UNIVERSITÄT DARMSTADT

- This lab gives you experience how to achieve **load balancing** with OpenFlow

- Put the script **lab5-topo.py** into your Mininet home directory
  - Simplified topology of a datacenter and client network linked together.

# Lab 5 Topology

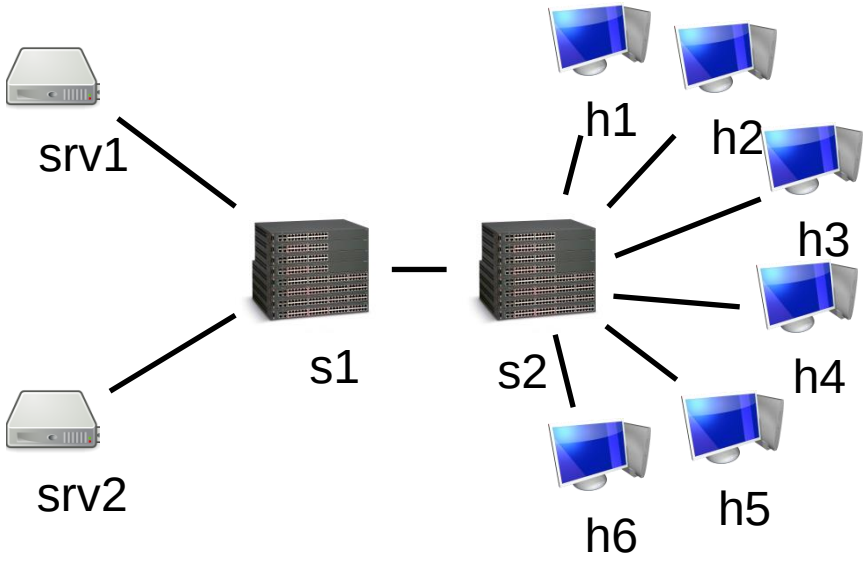
TECHNISCHE UNIVERSITÄT DARMSTADT



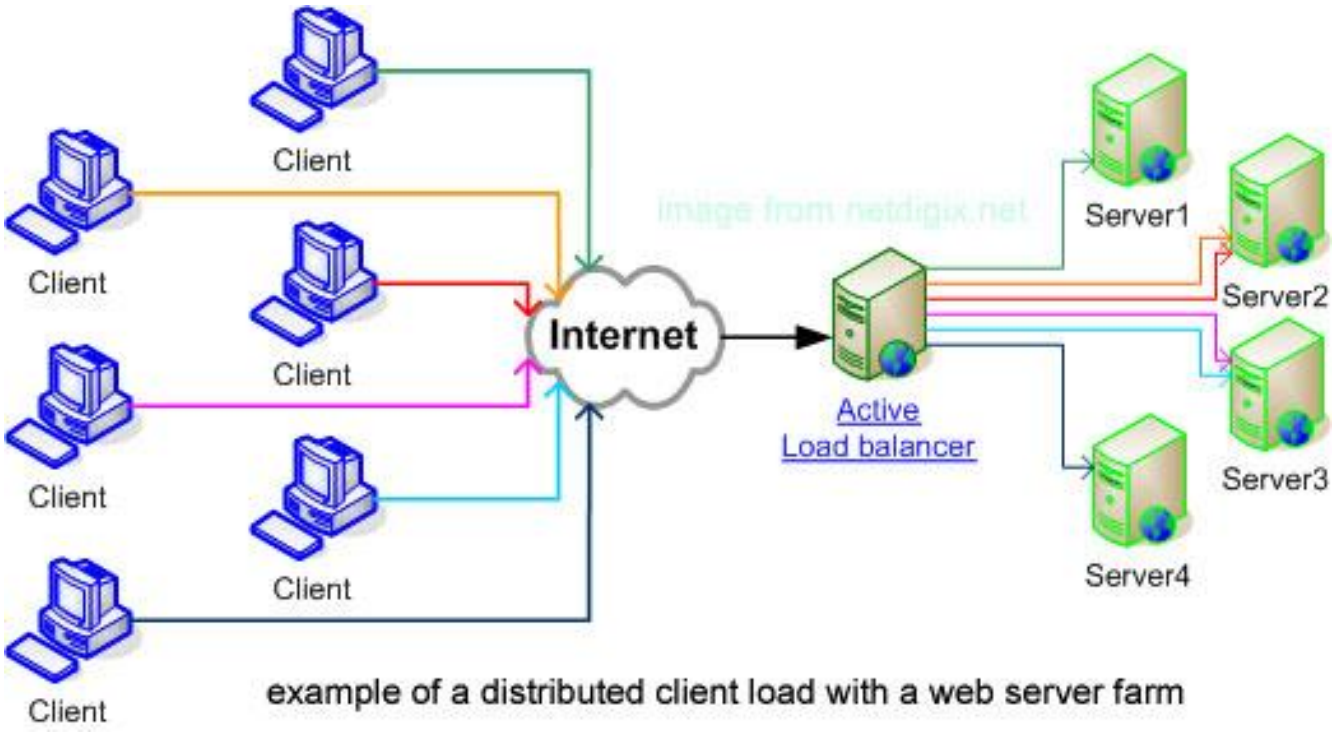
# Lab 5 Topology

TECHNISCHE UNIVERSITÄT DARMSTADT

- **s2** is linked to several **client** hosts
  - E.g. customers of an ISP
- **s1** is a datacenter switch connected to **servers**



# Load Balancer




Source: http://www.netdigix.com/images/load-balance.jpg

# Task 1: Design a Load Balancer with OpenFlow

TECHNISCHE UNIVERSITÄT DARMSTADT

- Create a Ryu application which **load-balances** requests from clients (e.g. h1-h6) between srv1 and srv2
  - Select a **virtual** IP address.
  - Whenever this virtual IP address is used by the clients, connections to it are load-balanced between the servers.
  - This way, servers are equally utilized
- **Requirements:**
  - A connection from the same IP address **must** use the same server over and over again
  - Different IP addresses may use different servers
  - The scheme used must be able to cope with arbitrary client IP addresses, not only with the addresses of h1-h6 (Hint: IP Wildcard matching)
  - Implement a **stateless** load balancer
- **Challenges**:
  - Use the controller to handle ARP requests to the virtual IP.
  - You can hard-code MAC addresses of the servers, if you want.

# Programming with Ryu

TECHNISCHE UNIVERSITÄT DARMSTADT

- Resources
  - http://osrg.github.io/ryu/
  - Documentation http://ryu.readthedocs.org/en/latest/index.html
  - Book with examples http://osrg.github.io/ryu-book/en/html/

- Things to be aware of
  - Ryu relies on a concurrent networking library called Eventlet (http://eventlet.net)
  - Eventlet is a concurrent networking library that uses coroutines for parallelism
    - Approach is similar to cooperative multitasking
    - No preemption of processes
  - What does that mean?
    - Make tasks that can take a long time to process preemptible
    - How?
      - Allow the scheduler to select another task by giving back the control from time to time
      - Normal function call:

        ```
        self.fancy_method(A,B,C)
        ```

      - Function call that allows the scheduler to switch tasks:

        ```
        hub.spawn(self.fancy_method,A,B,C)
        ```

# Running Ryu

TECHNISCHE UNIVERSITÄT DARMSTADT

- Use Linux
  - Use either the existing VM, which already has Ryu installed
  - Install it on a new OS by running `pip install ryu`

- But my Main Computer Runs Windows!
  - Ryu can made run on Windows, but does not work properly
  - Therefore: run Linux inside a VM
    - Connect the Linux VM to the outer network using „Bridged networking“
    - For the duration of the work, disable DHCP on the network interface of your Windows Computer