# Advanced Networking 2018

## Lab #5: OpenFlow

Total points: 15 pts<br>(Note that 11pts will give you already ≈ 7 ;-)

Assignment

Lab date: March 13 (with spillover date: 16) 2018
Submission date: 11:59PM March 16, 2018 CET

*Author: Ralph Koning*

*Email: r.koning@uva.nl*

University of Amsterdam

**Abstract**

In this assignment you will get some hands-on experience with OpenFlow. You will learn about the communication between an OpenFlow switch and an OpenFlow controller, and experiment with a physical OpenFlow switch.

# Preparation

1. Ask a colleague to team up with you.
2. Register your group on Blackboard. Remember your group number.
3. Please do not use 'ACTION=flood', you don't need this to complete the lab and it could interfere with your classmates experiments.

# Task 1: Deployment of the OpenFlow controller

Compile version 1.2 of the Floodlight OpenFlow controller on one of your servers, available at: `http://www.projectfloodlight.org/floodlight`. Setting it up requires `ant` and `openjdk-x-jdk-headless`, where x is the latest version. Run `make` in the floodlight directory. Then, start floodlight using `java -jar target/floodlight.jar`.

Configure the firewall on your server to accept connections to port 6653 only from the OS3 network.

Start Floodlight.

# Task 2: Network Setup

**When preparing the report make sure you do the following: For all tasks where you add flows manually, show the commands used to insert the flows and the relevant part of the flow table of the switch.**

*Tip*: You are dealing with a Pica8 3290P switch. This switch runs Debian Linux with OpenvSwitch, installed in `/ovs`. The knowledge related to Linux networking obtained in other courses will be useful here. Look in the "OVS Configuration Guide" for Picos 2.5 on `http://www.pica8.com/` for the right commands.

*Note*: In order for vlan translation to function in later examples, configure the ports using trunk mode.

## Steps

1. Connect the second interfaces of your servers to the switch labeled "AN2018 Openflow". Take note of the port numbers! Be careful to not interfere with the work of other groups. The switch is a shared environment!

2. Connect to the AN management network used in the juniper labs.
3. SSH into to the openflow switch at 10.0.1.50/32 using the credentials an/an20180penflow.
4. Create a bridge with name br_{group number}.
5. Add the switch ports connected to your servers to the bridge created previously
6. Configure the bridge to support OpenFlow versions between 1.0 and 1.3. Support for OpenFlow 1.4 is still experimental, and should not be used.
7. Configure the bridge to connect to your OpenFlow controller.

## Task 3: Basics (2 points)

The floodlight controller includes by default a L2 switch module. This module will make the OpenFlow switch behave like a traditional L2 switch. Perform a connectivity test between the servers and show the content of the flow table during the test.

What happens to the flow table a few seconds after you stop the test? Why?

## Task 4: Packet Capture (3 points)

Capture packets on the interface that connects to the controller when:

1. The switch connects to the controller.
2. The are no flows in the switch and a new connection triggers a packet being sent to the controller.

Explain what you see.

## Task 5: Static Flows (3 points)

Disable the L2 switch module of floodlight. Show it is disabled.

Make the machines reachable again by pushing flows to the floodlight controller via its “staticflowpusher” Web API. Use flows as generic as possible. Show that the machines can reach each other.

## Task 6: VLAN translation (3 points)

- Tag the traffic on the servers’ interfaces. Use your desk number as VLAN ID. Have the OpenFlow switch doing the VLAN translation so the machines can communicate. Explain the procedure.

- Look at the flows applied on the switch, does this match with the flow that you sent to the controller?
- Add `"eth_vlan_pcp":"0"` as selector to your flow. Look at the installed flow, explain what changed; what could be the cause of this?

## Task 7: Traffic firewalling (4 points)

Pair up with another group and put all of the switchports in the same virtual switch. Make sure they are removed from the previous virtual switch. Create the following scenario using the staticflowpusher (do not use the firewall module/API).

- Server 1 can only the reached by the MAC addresses of the server 3 and 4.
- Server 2 can only be reached by the IP address of server 3 and MAC address of server 4. Only HTTP traffic should be forwarded from server 4 to 2.
- Server 3 can handle HTTP requests from server 2. Can be reached by the MAC address of server 4 and by the IP address of server 1.
- Server 4 can be reached by the MAC address of server 1 and 3, and IP address of server 2. Only allow incoming SSH traffic from server 1.

Tests that can be performed to verify the scenario above:

- Server 1 can SSH to server 4
- Server 1 can ping server 3
- Server 2 can perform HTTP requests to server 3
- Server 3 can SSH to server 4
- Server 4 can perform HTTP requests to server 2

Test your configuration and include the output of your tests in the report.

## Submission

Every group of two should submit the following file:

- **report** (PDF format) that describes in detail steps performed and answers.

  `lab5-report-$groupnumber.pdf`

*Note:* it is sufficient that one of the two group members submits in Blackboard!

**Any other kind of submission will not be taken into account.**