Exercise for Lecture Software Defined Networking

Prof. Dr. David Hausheer
Julius Rückert, Leonhard Nobach

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Winter Term 2015/16
Exercise No. 2
Published at: 27.10.2015
Submission exclusively via Moodle, Deadline: 10.11.2015
Contact: `[rueckert|lnobach]@ps.tu-darmstadt.de`
Web: `http://www.ps.tu-darmstadt.de/teaching/ws1516/sdn/`
Submission: `https://moodle.tu-darmstadt.de/enrol/index.php?id=6349`

Surname (Nachname):	
First name (Vorname):	
ID# (Matrikelnummer):	

Problem 2.1 - SDN Relatives to OpenFlow

a) Name three concrete implementations/instantiations of the SDN concept, other than OpenFlow and ForCES.

b) The Forwarding and Control Element Separation (ForCES) Framework was standardized by the IETF in 2004 (RFC 3746). Please revisit the concepts presented in the lecture on ForCES by reading Section 1 and 2 of the standard document and answer the following questions. RFC 3746: https://tools.ietf.org/html/rfc3746

I) Explain the responsibilities of Control Elememts (CEs) and Forwarding Elements (FEs) in ForCES.

II) What does the standard state as reason for the separation between FEs and CEs?

Problem 2.2 - SDN and OpenFlow Basics

a) Separation of Concerns: Briefly explain in your own words the different responsibilities of the different layers of the commonly used SDN controller architecture: *SDN Control Program*, *Network Virtualization* layer, and *Network OS*.

b) Explain what the “Southbound API" of an SDN controller refer to?
Can you also imagine what the “Northbound” and “East-/Westbound API” could refer to?

c) What are implications of running the OpenFlow client as part of the software layer of a switch?

d) Taking your above answer into consideration, why is it hard to realize flow matching up to ISO/OSI layer 7?

Problem 2.3 - The OpenFlow Protocol

For the following questions you have to use the official OpenFlow specifications. You are not intended to read the whole documents! Get familiar with their structure and use them to look up details. Knowing how to navigate and find details within the specifications is essential for several later tasks and the lab. If we do not specify the version to be used, we assume version v1.5.0 in the following.
Important OpenFlow Specifications:

- v1.0.0: `https://goo.gl/G9UJP4`
- v1.1.0: `https://goo.gl/7JISrB`
- v1.3.0: `https://goo.gl/3WCVZz`
- v1.5.0: `https://goo.gl/VDxZN2`

Besides, we recommend the following webpage that can help to investigate differences between versions of the OpenFlow protocol etc.: `http://flowgrammable.org/sdn/openflow/`

a) Up to which ISO/OSI layer is packet matching supported in OpenFlow 1.0 without extensions?

b) Why does OpenFlow support switches to connect to multiple controllers?

c) What is the difference between *Instructions* and *Actions* as defined by the OpenFlow standard?

d) One very essential action that can be applied to packets of a flow is called *output*. What is the purpose of this action and what can the reserved port "CONTROLLER" be used for in this action's context.

Problem 2.4 - Virtual Machine Migration Mechanism (Case Study 2)

This task is a case study. You are supposed to demonstrate theoretical concepts defined in the lecture in an applied setting. Only the problem and its rough context is defined. The context may be extended, if necessary. You are intended to define processes and procedures to solve the problem. Your solution should be defined to an extent allowing a team of skilled staff to implement your solution, i.e., details may be omitted, if they do not have a large impact on your solution. The solution should be presented in a text-based form. Additional literature may be used and it is highly recommended to discuss your solution in a team.

Scenario and Setting:

Cloud provider Zonama owns a large data center. Each rack is equipped with an OpenFlow enabled rack switch. All rack switches are interconnected using a single OpenFlow enabled root switch (called *fat tree topology*). The OpenFlow switches are controlled by a controller VM providing a basic switching mechanism. Moreover, Zonama optimizes the assignment of vitual machines (VMs) to physical machines (PMs) constantly by migrating VMs during operation.

The third party provider Boxdrop is using Zonama's cloud to offer a backup service for his customers. As Boxdrop's VMs show constant interaction with Boxdrop's customers, a migration during runtime is difficult: as a migration takes a certain amount of time, connections to Boxdrop's customers may get lost, which is unacceptable.

Zonama proposes the following mechanism to the Boxdrop team: Zonama informs Boxdrop on the planned migration of VM1, using a common interface. In turn, Boxdrop returns a VM2 which may be used as a fallback machine during the migration. Afterwards, new connections trying to contact VM1 are transparently redirected to VM2 using OpenFlow's forwarding capabilities. As soon as the last user on VM1 has closed the last connection, VM1 will be migrated. A similar mechanism will be used to reverse the redirection to VM2, after VM1 is up and running again.

Create a system architecture to implement the discussed mechanism. Include a description of the involved components, the necessary communication between components and the necessary OpenFlow rules to control the data plane.