Exercise for Lecture
Software Defined Networking

Prof. Dr. David Hausheer, Julius Rückert
Christian Koch, Jeremias Blendin, Leonhard Nobach, Matthias Wichtlhuber

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Winter Term 2016/17
Exercise No. 4
Published: 29.11.2016
Submission exclusively via Moodle, Deadline: 13.12.2016
Contact: Please use the Moodle forum to post questions and remarks on the exercise.
Web: `http://www.ps.tu-darmstadt.de/teaching/ws1617/sdn/`
Submission: `https://moodle.tu-darmstadt.de/course/view.php?id=8385`

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

Problem 4.1 - SDN Hardware

a) Explain in your own words the main differences between the OpenFlow hardware classes “SOFTWARE” and “HARDWARE”.

b) OpenFlow v1.4.0 introduces the notion of *vacancy events*. Which inherent problem does this feature address? Shortly explain what *vacancy events* are and what would happen if a controller does not make use of them.

c) Explain and briefly discuss which parts of an OpenFlow flow entry should be stored in TCAM and which could be placed in normal DRAM to minimize the required TCAM space and still ensure constant-time packet processing. (Note: You can exclude the priority of a flow entry from the discussion.)

Hint: The answer can be derived from the material presented in the lecture. For a more detailed understanding, it might be helpful to understand how CAMs and TCAMs work on a lower level: `https://www.pagiamtzis.com/cam/camintro/` (this is optional material)

Problem 4.2 - NEC guest lecture: OpenState

a) New Tables proposed by OpenState

In the lecture, an approach called *OpenState* was presented that extends the matching behavior of OpenFlow. For this, it proposes to expose two new table abstractions to the controller: the *state table* and the *XFSM table*. Explain the purpose of these two tables and how they are used in handling of incoming packets. For this, please have a look at the original paper on OpenState (Hint: Sections 2.1 and 2.2 are relevant for answering the question):

http://openstate-sdn.org/pub/openstate-ccr.pdf

b) Applicability ot OpenState

Which types of use cases or applications could benefit most from OpenState?

c) Using OpenState for a simple use case

At the end of Section 2.2 of the above referenced paper on OpenState, the "port knocking" example is realized using OpenState. Briefly explain how it is mapped to the two tables.

Problem 4.3 - Network Virtualization and Slicing

a) Duties of an SDN Controller

Name three duties of a typical SDN controller and state at least two examples for each of them.

b) Network Virtualization with VLANs

Two datacenters D and E use VLANs for traffic isolation between their hosts. The datacenters are connected over a L3 connection, e.g. the Internet, and need to share their network segments.

I Propose a solution (without using SDN or VXLAN) for preserving the slices, even when communication is carried out over the L3 link.

II On an example of two hosts A (in Datacenter D) and B (in Datacenter E), draw a communication diagram of a packet sent from A to B over a common network segment. This diagram should include every intermediate hop (except pure L2 switches and simple routers on the Internet).

III Draw the protocol stack (except Layer 1 and everything above the Transport Layer) of the packet into the diagram, between each two intermediate hops. Example: (Ethernet, IP, UDP)

IV Make yourself familiar with VXLAN and the VXLAN header. Name at least *two* advantages of VXLAN compared to your solution (Hint: Think about an intermediate NAT between the datacenters).

V Name at least *one* advantage of OpenFlow compared to (pure) VXLAN in the context of network slicing.

c) Energy saving

Imagine at least one scenario in which SDN can save energy.