A Master Course on Network Softwarization: Lectures and Practical Assignments

Frederik Hauser, Mark Schmidt, Michael Menth
Chair of Communication Networks, University of Tuebingen, Tuebingen, Germany
Email: {frederik.hauser,mark-thomas.schmidt,menth}@uni-tuebingen.de,

I. Motivation

The bwNET100G+ research project focuses on innovating campus and scientific wide area networks using software-defined networking (SDN) and network function virtualization (NFV), two concepts that are also referred to as network softwarization. In that context, Master students at the University of Tuebingen have the opportunity to work on prototypical implementations within thesis projects. Examples are [1]–[3].

As such projects are demanding, students need a solid understanding of network softwarization and ideally have some practical experience in that area. In the past, they were instructed to read scientific papers from the ONF reading list [4] and to program applications for SDN controllers using tutorials such as [5] or [6]. However, this was rather inefficient as the process was difficult for the students and still required lots of individual supervision. Therefore, we established a lecture course including practical assignments at Master level as preparation for a thesis project on network softwarization.

The course prerequisites basic knowledge in communication networks. It provides an overview of legacy communication network management and introduces network softwarization as novel and relevant development. It teaches both theoretical fundamentals and implementation approaches, and discusses new challenges in that field. Moreover, students get first practical experiences with virtualized SDN using Mininet and program SDN applications for the Ryu SDN controller. While time-intense courses on network softwarization already exist [7], [8] at other institutions, our new course is tailored to a workload of only 3 ECTS including 90 hours of weekly lectures over 10 weeks during the summer term plus work on practical assignments and exam preparation.

II. Concept

The concept of the course includes three parts: lectures, programming assignments, and a final exam.

A. Lectures

The first part of the course focuses on theoretical education in form of a lecture's presentation that is based on slides. The contents are split up in seven chapters as following.

1) Introduction to Network Softwarization: The first chapter covers the transition from legacy to softwarized networks. Legacy management concepts, e.g., SNMP and NETCONF, more active network concepts, e.g., ForCES and 4D, and more recent concepts, e.g., OpenFlow SDN and P4, are discussed.

2) OpenFlow: The second chapter focuses on covering both, the architecture and protocol of OpenFlow. The lecture introduces OpenFlow in version 1.0 and 1.5.1. and shows the transition from the first to the latest feature set while describing the respective novelties such as flow table pipelines or meters.

3) SDN Controller: The third chapter describes the SDN application and control layer. It covers the architecture, design principles, and control plane functions of SDN controllers and their interfaces along concrete implementations, e.g., REST for northbound interfaces. The chapter closes with an overview of pupular SDN controllers, e.g., Ryu, OpenDaylight, and ONOS.

4) SDN Switches: The fourth chapter revises the hardware architecture of legacy routers and switches in the beginning. SDN software switches, Whitebox switches, OpenFlow-only switches, and SDN hybrid switches and their respective deployment scenarios are described in detail afterwards. The chapter closes with an in-depth view on the status quo.

5) SDN Use Cases: The fifth chapter introduces datacenter, enterprise, campus and wide area networks as predestined environments for the application of network softwarization. Especially benefits, but also limitation in comparison to legacy technologies are discussed in detail. The chapter closes with an in-depth description of particular concepts from our current and past research activities.

6) Virtualization Techniques: The sixth chapter focuses on the strongly related concept of computer virtualization. Hypervisor-based and OS-level virtualization technologies along with concrete implementations are presented. Concepts for the orchestration of virtualized infrastructures and particular concepts from our current and past research form the end.

7) Network Function Virtualization: The seventh chapter introduces the specifics of telecommunication provider networks as motivation for the emergence of NFV. The main part of the lecture covers the ETSI NFV architecture with all relevant details and discusses use cases. OPNFV is introduced as exemplary implementation of a NFV environment.

B. Practical Assignments

The second part of the course focuses on practical experiences. Students are required to meet a total score of 60% in two assignments to get an admission for the final exam. Scores above 60% are transformed into at most 10% bonus for the final exam grading. Teamwork skills were encouraged while the grading effort is limited by the requirement to work in groups of two. Sessions for the practical assignments were part of the weekly course schedule, i.e., two lecture slots within the semester were reserved. We published one assignment sheet at the beginning of the course, another in the middle that needs to be finished within a time period of four weeks. It contains two parts, a short group test and programming tasks.

1) Group Test: The assignment sheets include a catalogue of 15 to 20 questions that are either related to contents of the lecture or to aspects of the assignment. Each student is required to give answers to five questions of the catalogue a week after the assignment sheet was handed out. The group test verifies that both group members dealt with the related theoretical questions prior to programming task work.

2) Programming Tasks: In contrast to other courses, we limited the programming tasks on SDN application implementation for Ryu. Therefore, the students learn to implement known networking concepts, e.g., L2 switching or IP routing, using the capabilities of Ryu and OpenFlow-based SDN.

The first programming task focuses on familiarization with Mininet and Ryu by extending a L2 switch sample application to support port-based segmentation. The second programming assignment extends the L2 switch implementation with IPv4 routing. Four hosts residing in two different IP subnets are attached to a SDN switch which routes packets between the subnets. Figure 1 shows the topology for the third programming task. Eight hosts are conntected to two SDN switches that act as IPv4 edge routers. Two additional SDN switches acting as core routers represent the Internet between both networks. The edge and core routers are related to two different SDN controllers. The fourth programming task furthermore extends the routing functionality by introducing longest prefix matching in a more complex routing scenario and routing of IPv6 packets. In the last programming task, students implement packet- and flow-based IP anycast.

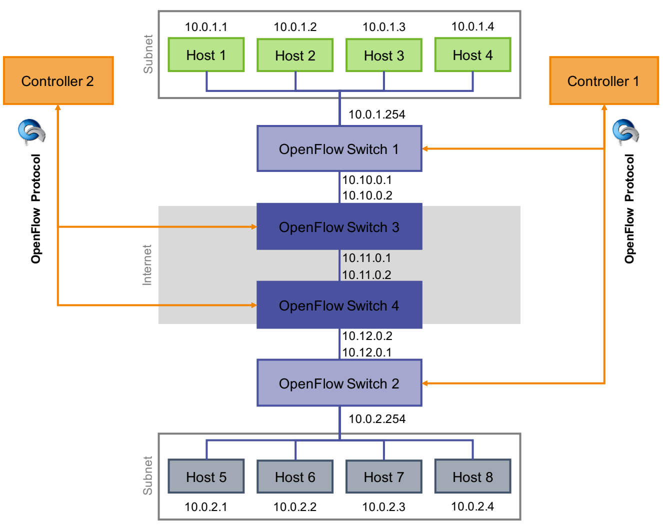

Fig. 1: Network topology for an IPv4 routing scenario with 8 hosts, 4 OpenFlow switches, and 2 SDN controllers.

To simplify the work on the programming assignments, we provided the NetSoft-VM, a virtual machine image as uniform programming and execution environment. We chose VirtualBox as virtualization platform as it can be installed on Windows, macOS, and Linux. The NetSoft-VM is based on Ubuntu 17.04 with XFCE and contains Mininet, MiniEdit as GUI for Mininet, Python 3, the Ryu SDN controller framework, and the Atom editor. Figure 2 shows the NetSoft-VM running in VirtualBox with MiniEdit as GUI for the Mininet network simulator.

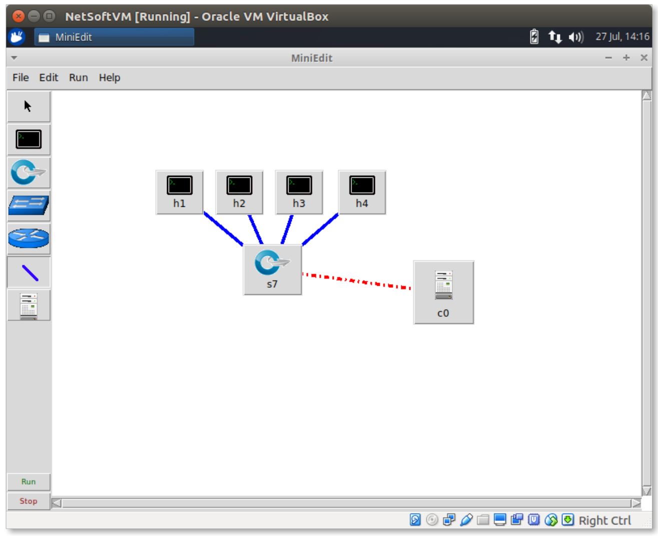

Fig. 2: Netsoft VM in VirtualBox with MiniEdit as GUI for the Mininet network simulation environment.

Successfully passing the group test was set as requirement for the grading of the programming tasks. The grading focused on source code functionality and quality.

C. Final Exam

Due to a limited number of participating students, we decided not to offer a written final exam. Instead, we chose the format of an oral exam with 25 minutes per student.

References

[1] M. Menth, M. Schmidt, D. Reutter, R. Finze, S. Neuner, and T. Kleefass, "Resilient Integration of Distributed High-Performance Zones into the BelWue Network Using OpenFlow," *IEEE Communications Magazine (Commag)*, vol. 55, no. 4, Apr. 2017.

[2] F. Hauser, M. Schmidt, and M. Menth, "Establishing a Session Database for SDN Using 802.1X and Multiple Authentication Resources," in *IEEE International Conference on Communications (ICC)*, Paris, France, Jun. 2017.

[3] M. Schmidt, R. Finze, D. Reutter, and M. Menth, "Demo: Resilient Integration of Distributed High-Performance Zones into the BelWue Network Using OpenFlow," in *International Teletraffic Congress (ITC)*, Wuerzburg, Germany, Sep. 2016.

[4] "Open Networking Foundation: SDN Reading List," https://www.opennetworking.org/sdn-resources/sdn-reading-list, accessed: 2017-08-06.

[5] "Mininet Walkthrough," http://mininet.org/walkthrough/, accessed: 2017-08-06.

[6] "Ryu Documentation: Getting Started," http://ryu.readthedocs.io/en/latest/getting_started.html, accessed: 2017-08-06.

[7] "Coursera: Software Defined Networking," https://www.coursera.org/learn/sdn, accessed: 2017-08-06.

[8] "Princeton University: COS-597E, Fall 2013: Software Defined Networking," http://www.cs.princeton.edu/courses/archive/fall13/cos597E/index.html, accessed: 2017-08-06.