Leading the OFELIA Facility Beyond OpenFlow 1.0 Experimentations*

M. Santuari, R. Doriguzzi-Corin, M. Gerola, E. Salvadori
CREATE-NET

U. Toseef, A. Zaalouk
EICT GmbH

K. Dombek, D. Parniewicz
PSNC

A. Hammad, M. Rashidi-Fard
University of Bristol

E. Jacob, J. Matias
UPV/EHU

***Abstract*—Nowadays, several devices supporting OpenFlow versions beyond 1.0 are available on the market. However, existent OpenFlow testbeds like OFELIA do not provide support for such devices, therefore it is hard for the research community to explore and test the newest functionalities of the protocol in realistic environments. This paper gives a short overview of two different architectures investigated within the FP7 ALIEN project that both aim to overcome this limitation within the OFELIA experimental facility.**

Keywords—Testbed, OFELIA, Control Framework, Network Virtualization, OpenFlow, Software Defined Network

I. INTRODUCTION

OFELIA is a collaborative project within the European Commissions FP7 ICT Work Programme [2]. The project creates a programmable network facility that allows researchers to not only experiment on a test network, but to control and extend the network itself precisely and dynamically.

To enable experimenters to register to the facility, configure their experiments, request, setup and release resources, OFELIA provides an experiment orchestration software, the OFELIA Control Framework (OCF) [3], which has been developed within the project.

The OCF allows the sharing of OpenFlow switches through FlowVisor [4], which is a special OpenFlow Controller that acts as a transparent proxy between the resource (i.e. the switch) and multiple controllers. FlowVisor is an external entity, which is able to delegate parts of the flowspace of the switch to different controllers (the so-called slicing process) and isolate the control plane associated with each part.

FlowVisor inspects the OpenFlow protocol to enforce the isolation between experiments, and consequently, it depends on the OpenFlow version. Currently, FlowVisor only supports OpenFlow version 1.0, therefore OFELIA only allows sharing OFv1.0 resources due to its tight relation with FlowVisor.

Nowadays, there are more and more OpenFlow resources (i.e. switches and controllers) that implement OpenFlow versions beyond 1.0. In the research community it is important to experiment with the latest tools and updated elements, therefore, OFELIA faces one limitation: how to test OpenFlow resources beyond v1.0 and integrate their control under the OCF.

For instance, the devices considered in the ALIEN project [1] are a clear example of this restriction. The ALIEN hardware can implement OpenFlow v1.3 or some particular extensions to v1.0 (e.g. support for circuit switching). Since FlowVisor needs to inspect the OpenFlow protocol, the ALIEN hardware cannot be directly integrated under the OCF. In this case, the FlowVisor should be updated each time to the new version.

In this paper, we introduce two architectures for enabling the OCF to support different versions of the OpenFlow protocol: the so-called *Time-based sharing* and the *Distributed slicing*. Both are alternatives to the current FlowVisor-based implementation of the OCF and do not require the inspection of the protocol. In the following sections, we first give an overview of the current OCF architecture. Then, we describe and compare the two aforementioned architectures by analyzing the benefits and the limitations of both of them.

II. OCF ARCHITECTURE OVERVIEW

The current architecture design of the OCF software is depicted in Fig. 1(a). This architecture is based on the GENI [5] Control Framework and inherits its core components.

The upper layer component, called Expedient, is a pluggable control framework developed by Stanford University where different plug-ins can be connected to different Aggregate Managers (AMs). Expedient also provides a Graphical User Interface (GUI) where different aggregates can be added to a project and configured. Since the Expedient architecture has a modular approach, new AMs can be supported by simply adding new plug-ins. The so-called ClearingHouse (GENI terminology) serves as trusted entity. It manages the projects (slices), the identities of experiments, the privileges of actors (credentials) and the respective project responsible(s). The ClearingHouse is the only entity in the architecture which hands out user certificates and credentials. FOAM [6] is an Aggregate Manager for OpenFlow resources developed by Stanford University. It is used to handle (create, approve, reject, disable, delete, list) the OpenFlow slices. Finally, FlowVisor allows the slicing of the flowspace, enabling the sharing of OpenFlow resources between several experiments at the same time.

III. THE TIME-BASED ARCHITECTURE

The time-based is a pragmatic approach that allows an easy integration of any SDN-enabled device (even non-OpenFlow) within the OCF. Its primary purpose is to avoid the inspection of the protocol by replacing the FlowVisor with a TCP proxy agent that forwards the control messages to the user's controller. Since no flowspace slicing operations are performed on the control channel, with this approach only one experiment is allowed at a time.

Fig. 1(b) highlights the main building blocks (with labels in bold) that allows the implementation of the time-based

*This work is partially supported by the EU FP7 ALIEN project [1].

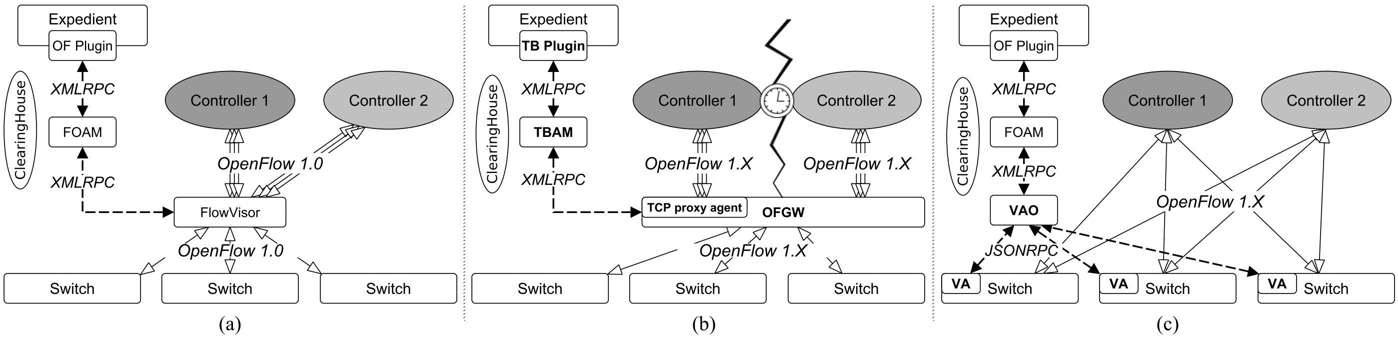

Fig. 1: (a) The current/legacy OCF software architecture, (b) the *Time-based* OCF implementation where only one controller/experiment is allowed at a time and (c) the *Distributed slicing* with no single points of failure on the control channel.

architecture within the OCF. As the resources are shared in a time-based fashion, a new aggregate manager, called *Time-Based Aggregate Manager (TBAM)*, is introduced in the OCF architecture to replace FOAM. Shortly, the TBAM receives the user's experiment configuration (e.g. time-slots, controllers IP addresses and TCP ports) from the Expedient platform, stores the allocated time-slots and manages the concurrent requests. User's experiments are configured through the *Expedient GUI* extended with a specific plug-in (the *TB Plugin* in the figure) which is in charge of providing management and control functions of the resources exposed by the TBAM.

The *OpenFlow Gateway (OFGW)* is the only entry point to access the Time-based resources under its control. It acts both as specific manager for the resources and as gateway for the data and control network. In particular, the OFGW is in charge of the following tasks (among others): allowing the access to the devices through the hardware-specific management interfaces (e.g. console, telnet, ssh etc.) and forwarding the control messages from the user's controller to the devices and vice-versa without inspecting the protocol. The latter is performed by the *TCP proxy agent* that is automatically configured by the TBAM with the user's controllers IP address and TCP port.

IV. A DISTRIBUTED SLICING ARCHITECTURE

The distributed slicing architecture has been proposed in [7] and leverages on a recent open-source datapath project named eXtensible Datapath Daemon (xDPd) [8] to perform the slicing process directly on the forwarding nodes. In brief, xDPd is a framework for building OpenFlow/SDN datapath elements for several hardware platforms and supports versions 1.0, 1.2 and 1.3 of the OpenFlow protocol.

The distributed approach tries to overcome some of the limitations of the FlowVisor-based and Time-based architectures: i) both leverages on proxy agents (FlowVisor and the OFGW) that constitute single point of failures for the control plane; ii) there is an inherent overhead due to the fact that the control messages have to be encapsulated/decapsulated twice (only FlowVisor) and transmitted/received via a socket; iii) OpenFlow protocol support of FlowVisor is limited to the version 1.0.

The distributed slicing architecture can be summarized by two macro-blocks as depicted in Fig. 1(c): the Virtualization Agent (VA), which resides on xDPd-enabled switches and leverages on xDPd's libraries to support multiple versions of the OpenFlow protocol, and the so-called Virtualization Agent Orchestrator (VAO), which is a stand-alone process in charge of configuring and monitoring the VA instances running on the network devices.

In particular, the VAO is the glue between the VA-enabled devices and the OCF. In fact, while on the southbound the VAO implements a JSONRPC interface which is used to send configuration commands to the VA instances (*createSlice*, *deleteSlice*, *addFlowSpace* etc.), on the northbound it exposes the same JSONRPC APIs used by FlowVisor and that allows a seamless integration of the VA-enabled devices within the OFELIA facility.

The major limitation of the proposed mechanism is that it is currently available for hardware platforms supported by the xDPd framework, however extending the architecture to other alternative datapath elements like Open vSwitch [9] (or others) is straightforward and requires changes only at VA level.

V. CONCLUSION

In this paper we introduced two different architectures, called *Time-based sharing* and *Distributed slicing*, that can be leveraged to deploy devices that are either non-OpenFlow or implement OpenFlow version which is beyond v1.0 within the OFELIA experimental facility. The time-based is a pragmatic approach that allows the integration of any SDN-based resources. Differently from FlowVisor, it does not perform the flowspace slicing and for this reason only one experiment at a time is admitted. On the other hand, the distributed slicing allows multiple concurrent experiments at the same time plus provides the support to many versions of the OpenFlow protocol. Nevertheless, the distributed slicing is currently only applicable to a limited subset of hardware platforms.

REFERENCES

[1] "ALIEN FP7 project," http://www.fp7-alien.eu.

[2] "OFELIA FP7 project," http://www.fp7-ofelia.eu.

[3] M. Suñé et al., "Design and Implementation of the OFELIA FP7 Facility: The European OpenFlow Testbed," *Computer Networks*, vol. 61, 2014.

[4] R. Sherwood et al., "Can the production network be the testbed?" in *Proc. of USENIX OSDI*, Canada, 4-6 Oct. 2010.

[5] "GENI consortium," http://www.geni.net.

[6] "FOAM," https://openflow.stanford.edu/display/FOAM/Home.

[7] R. Doriguzzi Corin et al., "A Datapath-centric Virtualization Mechanism for OpenFlow Networks," in *Proc. of EWSDN*, 2014.

[8] "The eXtensible Datapath daemon (xDPd)," http://www.xdpd.org/.

[9] "Open vSwitch," http://www.openvswitch.org/.