Hardware Abstraction Layer as an SDN-enabler for non-OpenFlow network equipment

B. Belter, D. Parniewicz, Ł. Ogrodowczyk, A. Binczewski, M. Stroiński

Poznan Supercomputing and Networking Center
ul. Noskowskiego 10, 61-704 Poznań, Poland

V. Fuentes, J. Matias, M. Huarte, E. Jacob

University of the Basque Country
Alameda Urquijo s/n - 48013 Bilbao, Spain

Abstract—This paper describes an experimentally-tested approach towards programmability for legacy network elements in the software-defined networking architecture. As OpenFlow is a leading control-plane protocol enabling SDN in modern networks, yet not all equipment is compatible with this framework. This problem is addressed in ALIEN, the FP7 research project, where Hardware Abstraction Layer (HAL) for non-OpenFlow capable devices is introduced. This paper describes outcomes of the project, and specifically, gives a set of examples of the successful implementation of OpenFlow through the HAL on 'alien' devices.

Keywords—OpenFlow; Software Defined Networking; Programmability; Hardware Abstraction Layer

I. INTRODUCTION

Hardware abstraction mechanisms are well known tools in operating systems. Unified and standardized routines allow an easy integration of various devices and hide the internal complexity of the hardware. This concept maps very well to the concept of SDN [1] and the Network Operating System [2]. The ALIEN project [3] provides an architecture and implementation of the Hardware Abstraction Layer for network equipment for a seamless integration of non-OpenFlow devices with the OpenFlow-based Network Operating System, realized with OpenFlow Controllers [4].

The concept of the HAL has been widely discussed in several papers [5][6]. This paper consequently focuses on specific implementations of the HAL on non-OpenFlow network hardware.

II. HARDWARE THEMES

The following hardware themes have been identified by the project to cluster network devices in several groups exposing similar features and behaviour:

- **X86-based packet processing devices**: this group comprises general purpose network devices that perform packet handling in software.
- **Programmable network processors**: this group refers to network devices which allow their data plane to be programmed to perform packet processing.
- **Lightpath devices**: since the OpenFlow protocol is limited to an Ethernet-like abstraction, for optical devices such as reconfigurable optical add-drop multiplexer (ROADM) systems, the abstraction layer must be adapted to meet the OpenFlow extension requirements to support circuit-switched networking.
- **Point to multi-point access networks**: for devices such as those based on standards like Gigabit Ethernet Passive Optical Network (GEPON) and Data Over Cable Service Interface Specification (DOCSIS), with deployments based on "head" and "tails" topologies, some kind of orchestration is necessary for exposing several devices as a single OpenFlow-enabled "device" through HAL.

Each of these four groups has different constraints and imposes various implementation challenges which have been explained in detail in several publications [7][8].

III. IMPLEMENTATION OF THE HAL

This section discusses implementation details of the HAL developments made on different categories of network equipment. Due to the paper's size limits, only programmable network processors and point to multi-point equipment are presented.

A. EZappliance (Programmable Network Processor)

Programmable Network Platforms represent a set of network equipment containing a re-programmable hardware unit (NPU or FPGA) that can be adapted to a wide range of network processing tasks (i.e. packet switching, routing, network monitoring, firewall protection, deep packet inspection, load balancing, etc.). These platforms allow for expressing packet processing control/service logic, using a programming language, in the form of compiled source code which can be implemented on a single hardware unit.

The heart of the EZappliance platform (Fig. 1) is the EZchip NP-3 network processor, a fully programmable chip which enables flexible parsing, classification, packet header manipulation and switching of pass through packets. The NP-3 processor is accompanied with a standard CPU foreseen for the deployment of control and management plane functionalities. It was used to deploy both Cross-Hardware Platform Layer (CHPL) and Hardware Specific Layer (HSL), as specified in the HAL architecture.

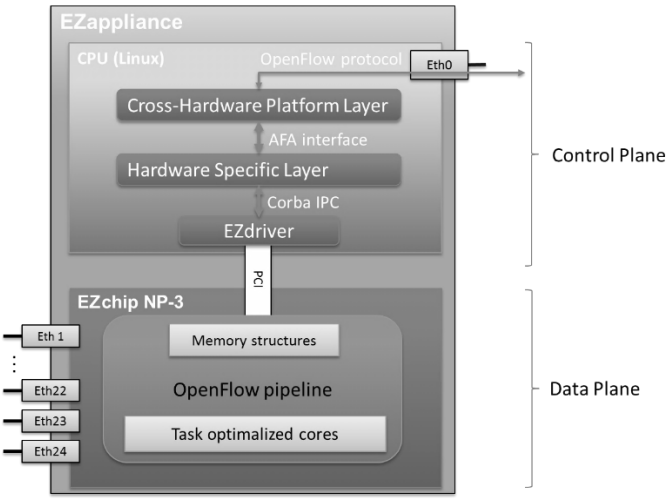

Fig. 1. HAL adaptation for EZappliance network processor platform

The HSL for EZappliance devices supports discovery and translation functionalities. The discovery functionality is based on automatic retrieval of information about all data plane ports, along with the corresponding attributes and status. As EZappliance is a standalone device, topology discovery is not required (for the same reason, the HSL for EZappliance does not include the orchestrator functionality).

The most complex part of the HSL is the translation functionality which transforms OpenFlow-based AFA messages into memory structures located within the NP-3 network processor. The NP-3 memory structures are accessed via the EZdriver provided by EZchip. The semantics used for the EZappliance memory structures is quite similar to OpenFlow. That is, the memory contains a structure with flow entries but the syntax is mostly different: proprietary binary encoding of packet matching and actions. Translation in the HSL is stateless.

B. DOCSIS (Point to Multi-Point Network equipment)

In general, point to multi-point devices consist of a "head-end" which communicates with several "tail-end" devices, usually through broadcast means with some multiplexing that allows the devices to know which traffic is intended for them. This approach is very common in access technologies.

The DOCSIS platform (Fig. 2) comprises three main elements: the CMTS, the cable, and the cable-modems (CMs). The CMTS is the head-end and "intelligent" part, which determines the use of the shared media by the CMs. The CMTS must be configured in the bridge mode (i.e. TLAN or L2VPN) to be compatible with OpenFlow abstractions. The cable is the shared media (coaxial) between the CMTS and several CMs. Finally, the CMs are the tails of the system located at the subscriber's location.

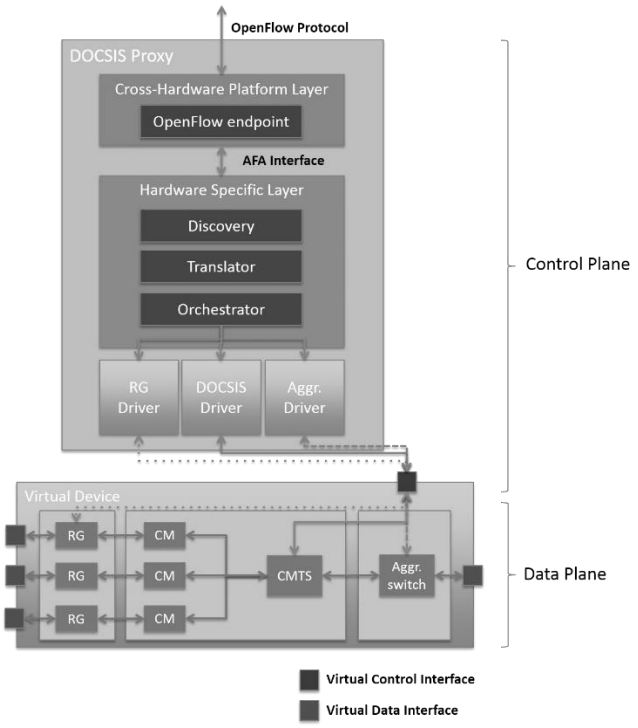

Fig. 2. HAL adaptation DOCSIS system

Since the DOCSIS platform is closed, a direct programming of the devices cannot be achieved, hence the control is only possible through vendor-supported standard interfaces. In principle, this limits the integration of DOCSIS under an OpenFlow interface. However, by adding the OpenFlow User Instance (OUI) and aggregation switch (AGS) in the picture (i.e. as helper boxes), it is possible to orchestrate the whole system to overcome these limitations and implement a fully compatible solution. As a result, the ALHINP (ALien HAL Integrating Network Proxy) performs the proper abstraction from the whole system by sitting (in the control plane) between the set of network devices and the OpenFlow controller. This proxy is based on AFA, since the actual data plane remains outside the DOCSIS proxy.

The HSL for DOCSIS implements the discovery, orchestration and translation functionalities. The discovery component provides information each time a new CM is connected to the system. As a result, ALHINP dynamically updates the virtual ports exposed to the controller, since each CM is abstracted as a new virtual port of the virtual OpenFlow switch. The orchestration component enables the coordination of multiple hardware components (i.e. OUI, CMs, CMTS and AGS) so they act as a single device.

IV. CONCLUSIONS

This article reports on the implementation details of the ALIEN Hardware Abstraction Layer on different hardware groups to enable OpenFlow processing on specific network equipment. The abstract part of the HAL is out of scope of this paper and reported in other articles ([5][6]), while the emphasis here is on the hardware-specific part (Hardware Specific Layer, HSL) and its implementation on EZappliance and DOCSIS.

The HSL implementation has to be carried out for each underlying hardware platform and, sometimes, it may even vary from device to device within the same network platform category. Therefore, depending on the hardware architecture of the underlying device, the HSL has to be implemented accordingly in order to offer the functionality described in the HAL specification.

Design choices, implementation issues and lessons learned are not reported in this article. Anyone willing to create a HAL for other hardware is advised to read [5] and [6], where specific implementation details have been provided.

ACKNOWLEDGMENT

This work was conducted within the framework of the FP7 ALIEN project, which is partially funded by the Commission of the European Union under grant agreement no. 317880.

REFERENCES

[1] K. Kirkpatrick "Software-defined networking." Commun. ACM 56.9 (2013): 16-19.

[2] N. Gude, "NOX: towards an operating system for networks." ACM SIGCOMM Computer Communication Review 38.3 (2008): 105-110.

[3] ALIEN Project Webstie: http://www.fp7-alien.eu/

[4] N. McKeown, "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[5] D. Parniewicz, "Design and Implementation of an OpenFlow Hardware Abstraction Layer", Proc. SIGCOMM DCC 2014, Chicago, USA

[6] Ł. Ogrodowczyk, "Hardware Abstraction Layer for Non-OpenFlow Capable Devices", Proc. TNC 2014, May 2014, Dublin, Ireland.

[7] Richard G. Clegg, "Hardware platforms and switching constraints", ALIEN Project deliverable, 2013

[8] R. Rajewski, "Specification of Hardware Specific Parts", ALIEN Project deliverable, 2013