LegacyFlow: Bringing OpenFlow to Legacy Network Environments

Fernando Farias, Igor Carvalho, Eduardo Cerqueira, Antônio Abelém
Federal University of Pará, Brazil
fernnf, igfuca, cerqueira, abelem@ufpa.br

Christian E. Rothenberg
CPqD/Telecom. R&D Center, Brazil
esteve@cpqd.com.br

Michael Stanton
Federal Fluminense University, Brazil
michael@ic.uff.br

ABSTRACT

The OpenFlow protocol allows production networking environments such as campus networks, metropolitan networks or R&D networks, to be used as experimental infrastructure hosting, future internet architectures, softwares and protocols, in isolation to the production traffic. During rollout, one practical problem arises with legacy switches that do not support the OpenFlow protocol and need to be replaced/upgraded or worked around by means of costly network re-engineering. This poster proposes a new OpenFlow datapath, which is able to interact with non-OpenFlow legacy equipment, creating a new approach to hybrid OpenFlow networks.

Keywords

Openflow, Computer Networks, Future Internet.

1. INTRODUCTION

Openflow [1] is an enabling technology of Software-Defined Network (SDN) [2], characterized by networking hardware being controlled via software effectively separating the control plane from the data plane (Datapath). This capability enables researchers or administrators to reprogram the forwarding behavior of network elements (e.g., routers or switches) in the OpenFlow Datapath, without interfering with the configuration of the production network. While OpenFlow has gained a lot of attraction from all players in the networking industry with the first commercially-supported products already available, the successful deployment of such a cutting edge networking is far from seamless [4]. Also, its use is growing up in experimental infrastructure (i.e., GENI or OFELIA) to the support of experiments in Future Internet.

Enabling OpenFlow in a production environment requires replacing (or at best upgrading) the legacy switching and routing infrastructure with OpenFlow-enabled devices. This brings up new costs to introduce the OpenFlow technology and may be a waste of well-performing legacy equipment.

This paper introduces an architecture to accommodate legacy switches in an OpenFlow network. The proposed solution presents virtual datapaths representing legacy devices that translate a new set of OpenFlow actions into vendor-specific configurations for Ethernet switches. This procedure is performed based on known interfaces such as Simple Network Management Protocol **(**SNMP**),** WebService or Common Language Interface (CLI). Initially, circuit-based VLANs are used to “bridge” OpenFlow switches over legacy switches.

This paper is structured as follows. Section 2 describes the proposed model for legacy equipment support. Section 3 presents the architecture and Section 4 concludes the work.

2. Proposed Model

Currently, adopting OpenFlow means replacing legacy switches with OpenFlow switches (OFSW). Despite being a costly proposition, not every switch can be replaced because, they are paramount for the production network or there is no equivalent OpenFlow-enabled equipment. Hence, it is necessary a solution be able to use the openflow without loss compatibility to legacy equipment. What will allow that experimental network based on openflow can offer this equipment in its experiments, increasing the resource availability and allowed virtualization them. Which is important to the evaluation of the Future Internet solutions.

This paper proposes a hybrid model to support both OpenFlow and legacy switches in the same infrastructure. The network operation is similar to an OpenFlow network with separation of control and data planes (e.g. controller and datapaths). The key difference is the introduction of a new virtual datapath merging the OpenFlow and legacy networks, as illustrated in Figure 1.

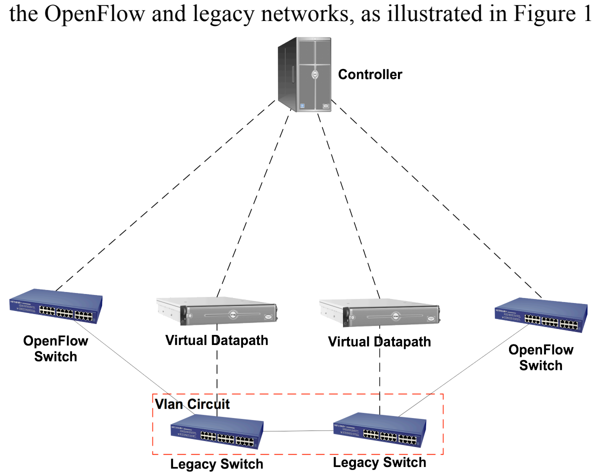

Figure 1. Openflow network is using the LegacyFlow solution.

In an OpenFlow network, when a packet reaches the OpenFlow datapath, it is checked to look for a matching flow entry. For example, to forward a flow to a certain egress point or to discard the packet. If no match is found, the packet is sent to the controller, and then, an action is applied on the switch to treat upcoming packets associated to a flow. The same rule is applied to other switches and required to forward flows to its destination.

Legacy switches cannot send information about packet flows arriving in the box. Thus, these switches will be commonly used in the network core that interconnects OpenFlow switches. This allows that just the edge switches (physical or virtual) connected to the network clients require OpenFlow capabilities. When a new flow enters the network, the first packet is sent to the controller that will install the required flow rules on datapaths, such as creating a VLAN circuits between legacy switches.

Based on the aforementioned scenario, a new OpenFlow datapath has been developed and called *virtual datapath*. Located in an outer virtual environment, the virtual datapath interacts with both OpenFlow controller and legacy switches through special actions created to support VLAN circuits. As presented in the next section, additional architectural components have been created to assist the datapath in realizing this hybrid mode of SDN that integrates legacy equipment within an OpenFlow network.

3. ARCHITECTURE

Basically, the main components of the LegacyFlow framework are the Virtual Datapath and the Virtual interface.

3.1 Virtual Datapath

The virtual datapath acts as a proxy that receives OpenFlow commands sent by the controller, interprets them, and applies the corresponding actions in a real switch. Currently, the interfaces used to access some devices are SNMP, WebService and CLI. Each legacy switch (part of the hybrid OpenFlow network) is assigned to a virtual datapath that runs in a guest machine (real or virtual) with GNU/Linux OS.

The goal of this virtual datapath is to represent an outside view of an OpenFlow datapath and some information about the features of legacy switches as following: port numbers, throughput, sent and received packets rate, among other issues specified in OpenFlow [3]. The datapath, besides basic information, supports a new set of circuit-specific specific actions. Figure 2 presents an overview of the virtual datapath.

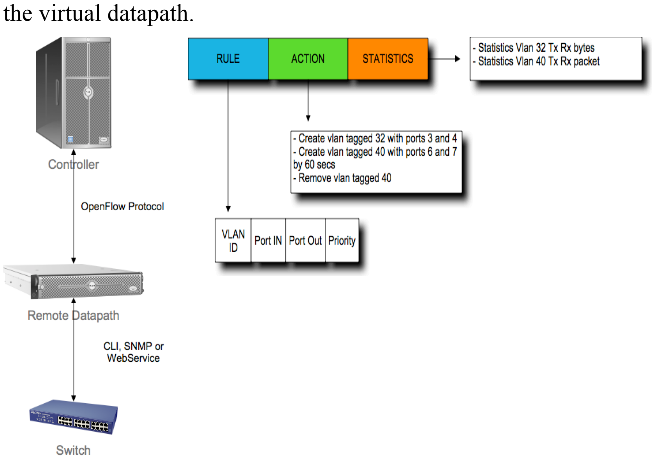

Figure 2. Overview of the virtual datapath

The developed actions deal mainly with characteristics of circuits:

- Creation of circuits without timeout, in which is created a VLAN into the switch with two interfaces, ingress and egress, with timeout to remove the switch configuration,
- Creation of VLAN with timeout, which has the same functionality of the previous one, but a small difference when the switch configuration is active for a short period of time;
- Creation of circuit with Quality of Service (QoS), in case the system must guarantee the performance of flows on switches to apply QoS in the VLAN circuit;
- Sent and received bytes rate, which will inform some statistics of sent and received data to a specific interface;
- Removal of a circuit, when performed, deletes the switch configuration immediately.

3.2 Virtual Interface

In order to start an Openflow datapath, it is necessary to pass the network interfaces as parameters. Likewise, the virtual datapath needs to be started with information of the switch interfaces. Therefore, it has developed a Linux module that creates virtual network interfaces according to the number of ports of switches and their features, mirroring the actual switch port features such as: throughput, Maximum Transmission Unit (MTU) size, sent and received packets rate, among other values. Depending on how the switch offers that information, it can be transferred from the switch to virtual interfaces via SNMP or WebService.

3.3 Legacy Flow in Operation

First of all, the virtual datapath is started receiving as mandatory parameters the legacy datapath switch model so that it can decide on the correct communication protocol. During the datapath initiation, the virtual interface module is initiated and created virtual interfaces on the operational system according to the ones available in the switch. Using an outer and dedicated out-of-band channel between the switch and the virtual datapath, SNMP connections are created to collect information of the switch interfaces and apply them to the virtual interfaces. Currently, this state is updated every 3 seconds, keeping information more accurate.

After the setup phase, the interfaces are connected to the virtual datapath*,* which can start receiving flow actions from the controller. When commands are received in the datapath, OpenFlow interprets the message and checks whether these actions are compatible with the datapath. Then, when a flow gets into a qualified OpenFlow switch and the controller identifies that the destination is reachable via another OpenFlow switch, a circuit is established between these two elements, using legacy switches.

4. CONCLUSION AND FUTURE WORK

LegacyFlow aims at improve the gradual deployment of OpenFlow in production environments, reusing the legacy equipment without changing the entire infrastructure. This proposal uses circuits among institutions, enabling experiments in federated environments using dynamic circuits among them. This is very much in spirit of the GENI network stitching based on VLANs. This work is in accordance with several activities of the Slice Federation Architecture (SFA), and investigates the interactions with FlowVisor, another OpenFlow relevant proxy that similarly aims at enabling isolated virtual networks. As future work, it is claimed to broaden tests with the elements of the architecture and develop even more its integration with other features of OpenFlow framework. A testbed has been developed to evaluate this proposal in real equipment using SummitX150 as legacy equipment and Soekris 5501 as openflow switch.

5. REFERENCES

[1] McKeown,N and *et al*. OpenFlow: Enabling innovations in campus networks. *SIGCOMM CCR 38:69-74, 2008*

[2] Lantz,B.,Heller, B., and McKeown,N. A network in a laptop: rapid prototyping for software –defined networks. In*,* Hotnets 10, Oct. 2010.

[3] Openflow. OpenFlow switch specification. Available at: http:// www.OpenFlow.org/documents/OpenFlow-spec-v1.0.0.pdf. Accessed: 20/07/2010.

[4] Davy, M. and *et al*. A Case for Expanding OpenFlow/SDN Deployments On University Campuses, GENI Workshop Whitepaper, Jul. 2011.