OPEN NETWORKING
FOUNDATION

Outcomes of the Hybrid Working Group

March 2013

1 Introduction

The earliest OpenFlow enabled switches that were considered commercially viable were available starting around 2010. These switches were already integrating OpenFlow functionality with existing feature sets. They had to answer the many resulting questions in an ad-hoc manner to move forward with those implementations. As a result, there was active discussion within the OpenFlow community about this integration with the release of the 1.0 specification at the beginning of 2010.

At the Open Networking Summit in October, 2011, there was a group discussion addressing considerations related to the integration of OpenFlow with non-OpenFlow control mechanisms. There ensued discussion in ONF of a working group to address these questions and the next month a draft charter for the group was circulated. The group was chartered in January of 2012. The work continued through October, 2012.

This document summarizes the findings of the group.

2 Site of Interaction

Two places where OpenFlow and non-OpenFlow might interact were considered. The first was called the *hybrid network*. This discussion considered the implications of connecting a pure OpenFlow network to a non-OpenFlow network. The second was called the *hybrid switch*. This addressed how different control mechanisms might interact or interfere with each other on a single (usually physical) switch or router.

3 Motivation

3.1 Levels of Interaction

The Hybrid Working Group was motivated by two objectives. The first was *coexistence*. From this perspective, the goal was to consider OpenFlow coexisting with other functionality. This applied to both the hybrid switch and the hybrid network. The second was *transition*. Here, the goal was to consider the states of transition from an existing non-OpenFlow deployment to one which is fully controlled by OpenFlow. The expectation was that intervening states may require some considerations of coexistence, but such considerations would be transitory.

3.2 Use Cases

The group identified a variety of use cases. A partial list follows.

- Integrating OpenFlow with an existing authentication service by leveraging OpenFlow's flow based traffic identification

- Adding multi-path L2 forwarding with non-OpenFlow L3 forwarding
- Augmenting perceived limitations of OpenFlow, for example for queuing support
- Leveraging OpenFlow to identify traffic for an offload or firewall mechanism

4 Forwarding Models

The group explored in detail two forwarding models representing how OpenFlow and non-OpenFlow approaches might interact. These models were intended to delineate the engineering space for addressing how to combine OpenFlow and non-OpenFlow functions. They were identified as *Ships in the Night* and *Integrated.*

In the *Ships in the Night* model, the data plane traffic is segregated into OpenFlow and non-OpenFlow categories by some extra-OpenFlow criteria. For example, either physical ports or VLAN tags might be used to direct packets to a specific function. In the ideal, the OpenFlow and non-OpenFlow functions are decoupled. The state each requires to forward packets is independent and activity of one function does not impact the resources available to the other.

The *Integrated* model attempts to capture how OpenFlow might be folded into an existing non-OpenFlow forwarding model. There are many existing non-OpenFlow approaches to forwarding, and there are many ways to integrate OpenFlow into each one. The group identified a few canonical ways this might happen, typically inserting OpenFlow as a functional unit into a predefined pipeline.

In addressing the *Integrated* model, there was significant discussion of how OpenFlow could provide services to existing layer-specific protocols. At the same time, there was a thread considering whether this was an appropriate use of OpenFlow, or if, even in a hybrid environment, it could maintain its fundamentally layer-agnostic forwarding model.

5 Outcomes

5.1 Recommendations for Switch Developers

The group used the *Ships in the Night* model to address the implications of both on-switch resource constraints (table sizes, internal bus or memory bandwidth, processing power) as well as more general network constraints such as link bandwidth. The group recommended that switch developers ensure proper mediation of resources between the OpenFlow and non-OpenFlow functions to avoid over-subscription.

5.2 Recommendations for Specification

The greater attention of the group was focused on the *hybrid switch* rather than on *hybrid networks*. At least in part, this was justified as follows: to the extent that there is interaction between a pure OpenFlow switch and a non-OpenFlow network, this must happen at the control plane level. Hence, the consideration of control plane interactions as discussed in the *Integrated* architecture should address these considerations.

Towards this interest of addressing control plane interactions, the group made a strong recommendation to consider mechanisms allowing greater bi-directional communication between the switch and the controller so as to be able to alert the controller of switch-side state changes. These suggestions have resulted in proposed updates to the specification by the Extensibility Working Group, for example in the consideration of group modification notifications.

5.3 Recommendations for Future Investigations

Upon review of the work done by the Hybrid Working Group, a majority of the Board and other Working Group chairs recommended that the integration of OpenFlow and non-OpenFlow functionality be carried out without hybrid switches. Network deployments are already heterogenous entities with the ability to incorporate diverse functional elements. Integrating OpenFlow directly provides significant benefit at a much lower cost and complexity.

5.4 Summary

On the whole, the group determined that industry can address many of the issues related to the hybrid switch. ONF does not plan or intend to incorporate details of legacy protocols in OpenFlow. The priority of ONF in this context is to explore the migration of networks to OpenFlow.