

Rapid Feature Implementation on Edge-Routers With OpenFlow

Kirill Kogan ¹ Alejandro López-Ortiz ¹

¹University of Waterloo

December 4, 2012

Outline

1 Problem Statement

2 Solution

Problem Definition

Problem

Given a set of features that are proprietary or implemented on top of different types of controllers.

Our goal: provide a framework that allows to apply the desired features simultaneously on the same traffic.

Requirements:

- The framework should be transparent as possible to managing controllers.
- Coexistence with existing proprietary features.
- Minimal code changes of an existing proprietary SW.
- For simplicity, all OpenFlow controllers managing the same network element use the same version of the OpenFlow protocol.

Taxonomy of Implemented features

- Self-contained (no sharable state with the others)
- Producers (DPI)
- Consumers (e.g. QoS consumes DPI state)
- Producers-consumers (e.g. Netflow consumes DPI produces flow state)

Assumption

Sharing of a feature state is possible only between features that are implemented on top of the same controller.

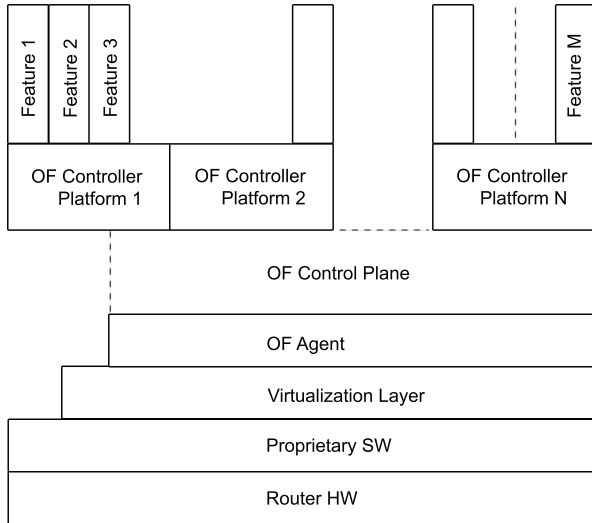
- Proactive
- Reactive

Outline

1 Problem Statement

2 Solution

Proposed SW Architecture



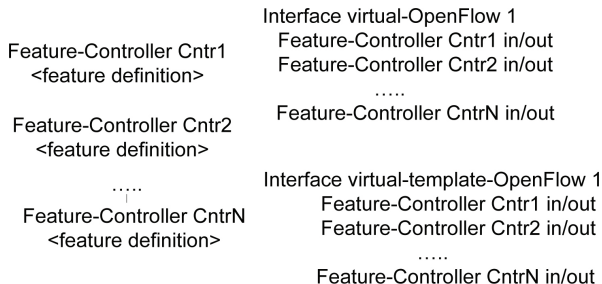
Interface as Entity for Features Attachment

Question

How to define an appropriate part of a network element that should be registered to different instances of OpenFlow controllers?

- The interface between OpenFlow agent and OpenFlow channel is implementation-specific (we can define how to register a network element or its parts with OpenFlow controllers)
- OpenFlow supports OpenFlow-hybrid mode.

Attachment Model



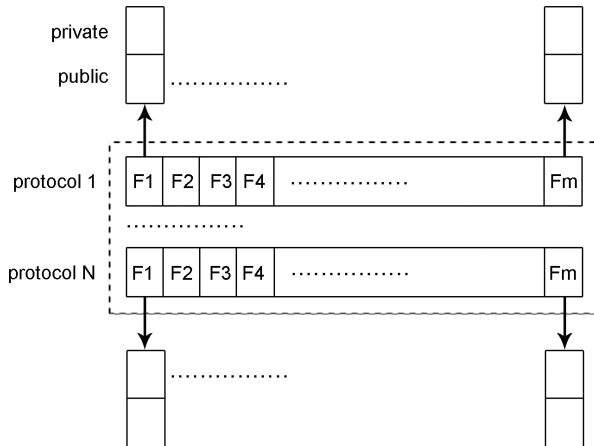
Attachment Model Cont.

- Controller configuration is defined as any other proprietary feature.
- Execution order of the applied features is defined by the order of the applied features.
- Execution order of the features implemented on top of the same controller is defined by OpenFlow controller.
- Registration with OpenFlow controllers occurs during attachment.

Advantages of Interfaces

- Already existing entities (resource allocation in proprietary SW)
- Two types of interface provision: reactive and proactive.
- Clearly defined location for feature replication and high-availability.
- Capabilities are defined implicitly by interface location.
- By product slot-slot HA (if exists in proprietary SW).
- Minimal changes to existing configuration model.
- Transparent to data path.
- OpenFlow is used only as a standard way for data path provision.

Interface Context



Summary

- Simple way to enable new features in existing architecture
- Short development cycle.
- Potential opportunity to create new economical models.

Thank you!

Thank you for your attention!