# OpenDaylight Project PCMM Southbound Plugin

## History

|  |  |  |
| --- | --- | --- |
| Thomas Kee | 2013/10/10 | Initial living information for integration |

# Goal

* Demonstration at SCTE Show on 21 October 2013 to preview the possibility of using an SDN controller to manage CMTS’s as network elements.
* Seed development of an southbound plugin to enable support CTMS and garner industry support with CableLabs sponsorship and possible incubation as an open source OpenDayLight project (but not for Hydrogen release)

## Documentation

[PacketCable™ Specification Multimedia Specification PKT-SP-MM-I05-091029](http://www.cablelabs.com/specifications/archives/PKT-SP-MM-I05-091029.pdf)

[OpenDaylight Project Developer Wiki](https://wiki.opendaylight.org/view/Main_Page)

## Framework

OpenDaylight is an open source project under the Linux Foundation with the mutual goal of furthering the adoption and innovation of Software Defined Networking (SDN) through the creation of a common industry supported framework.

The following diagram highlights the OpenDaylight Framework and how the PacketCable components fit in to the solution.



## Overview

OpenDaylight supports the OSGi framework and bidirectional REST for the northbound API between the Controller Platform and Network Applications.  The OSGi framework is used where applications will be running in the same address space as the controller while REST APIs will be used when applications will be running in different address spaces, or on different machines, as the controller.

### PacketCable SDN Application Logic

This component is introduced at the Network Applications Orchestration and Services layer. The component is proposed as an application responsible for the workflow logic that represents the use cases of interest: L2VPN, Buffer Management, Service Flow QoS Management, and general configuration automation.

### Flow Programmer

This standard component exists at the Controller Platform layer and is responsible for programming flows at network elements that register support for flow management.  It adheres to this [FlowProgrammer Restful API](https://jenkins.opendaylight.org/controller/job/controller-merge/ws/opendaylight/northbound/flowprogrammer/target/site/wsdocs/resource_FlowProgrammerNorthbound.html).

### PacketCable Manager

This component is introduced at the Controller Platform layer.  This component manages the PCMM/DOCSIS specific attributes that are applied to an existing Service Flow, configuration, and addition and removal of CMTS elements.  It should export a PacketCable Manager RESTful API as part of ODL architecture.

### PCMM/COPS

This component is introduced at the Southbound Interfaces & Protocol Plugins layer.  This component is responsible for the PCMM/COPS/PDP functionality required to service requests from PacketCable Manager and FlowManager.  Requests are transposed into PCMM Gate Control messages and transmitted via COPS to the CMTS. This plugin adheres to the PCMM/COPS/PDP functionality defined in the [CableLabs specification.](http://www.cablelabs.com/specifications/archives/PKT-SP-MM-I05-091029.pdf)

## Demonstration

Enables a minimal functionality southbound plugin to create and delete best effort service flows that show cases a high throughput video flow and a low throughput video flow control via the OpenDayLight controller by an SDN application utilizing the northbound restful API.



### Setting up the Development Environment

https://github.com/xsited/protocol\_plugins/blob/master/packetcable/README

### SDN Application

https://github.com/xsited/sdn/blob/master/python/restapi.py

### ODL PCMM/COPS Southbound

Pending integration of two repositories

https://github.com/xsited/protocol\_plugins/tree/master/packetcable

### PacketCable Manager

Not accounted for and any specific PCMM driven attributes or CMTS management configuration data is hardcoded in the plugin.

## Notes

SAL dispatches flows to plugins designated by node type

****

1. The flow name is translated into a Flow ID by SAL and needs to be mapped in the PCMM plugin to a service flow gate ID.
2. Test.java (or any form that it end up being) is a menu driven unit test that allows for functional testing without using the ODL integration. It is also a vehicle for remote support of the demo.
3. Merge repositories packetcable and protocol\_plugins.
4. This method should converts a FlowProgrammer dispatched Flow definition to a Gate Control message.

<https://github.com/xsited/protocol_plugins/blob/master/packetcable/src/main/java/org/opendaylight/controller/protocol_plugin/packetcable/internal/FlowConverter.java#L217>

1. Typically an Openflow enabled switch configures the controller at the switch and the conversation starts from the switch. This would be also possible with COPS, but with PCMM/COPS the conversation begins from he policy server or in this case the SB plugin. A fake CMTS is created here as a temporary solution. Connecting here might be possible as well.

<https://github.com/xsited/protocol_plugins/blob/master/packetcable/src/main/java/org/opendaylight/controller/protocol_plugin/packetcable/internal/InventoryService.java#L82>

1. Add Flow and Delete Flow

<https://github.com/xsited/protocol_plugins/blob/master/packetcable/src/main/java/org/opendaylight/controller/protocol_plugin/packetcable/internal/FlowProgrammerService.java#L69>

**Useful Links**

[OpenDaylight Main Website](http://www.opendaylight.org/)

[OpenDaylight Wiki](https://wiki.opendaylight.org/view/Main_Page)

[YANG Models Wiki](https://wiki.opendaylight.org/view/YANG_Tools:Available_Models)

[YANG Modeling in the Southbound Plugin Development (Model-Driven SAL)](https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Southbound_Plugin_Development_Guide)

**Meetings**

[Technical Steering Committee](https://wiki.opendaylight.org/view/TSC:Main)

[Technical Work Stream](https://wiki.opendaylight.org/view/Tech_Work_Stream:Main)