

PacketCablePCMM Proposal

- [Name](#)
- [Repo Name](#)
- [Description](#)
- [Scope](#)
- [Non-Goals](#)
- [Work Flow Example](#)
- [Design Details](#)
 - [DOCSIS SDN Application Logic](#)
 - [Flow Programmer](#)
 - [DOCSIS Abstraction Layer](#)
 - [PCMM/COPS](#)
- [Information and Data Models](#)
 - [Developing a Model](#)

Information and Data Models

- [OpenDayLight Models](#)
- [Developing a Model](#)
- [OpenDayLight Metering](#)
 - [Openflow Match Action and PCMM Service Flow Comparison](#)
 - [Openflow Match](#)
 - [Openflow Action](#)
 - [PCMM COPS Information Model](#)
- [Use Cases](#)
 - [Proactive QoS](#)
 - [Reactive QoS](#)
 - [Proof of Concept](#)
- [Research](#)
 - [Using Netconf to configure CCAP](#)
 - [CCAP and DOCSIS 3.1 CMTS YANG Model Support](#)
 - [CCAP PCMM Configuration Management Information Model](#)
- [CCAP](#)
 - [Using SNMP for SubscriberID \(Cable Modem\) Discovery](#)
- [Expressing Support and Interest in Project](#)
- [Resources Committed \(developers committed to working\)](#)
- [Initial Committers](#)
- [Vendor Neutral](#)
- [Meets Board Policy \(including IPR\)](#)

Name

PacketCable PCMM

Develop a PacketCable PCMM/COPS southbound plugin and supporting modules to allow the OpenDayLight controller to provision CMTS as a network element that manages service flows with dynamic QoS.

Repo Name

packetcable

SCTE demo and ODL staged source can be found [at GitHub](#).

Description

Packet Cable MultiMedia (PCMM) provides an interface to control and management service flow for CMTS network elements. A service flows constitute a DOCSIS data path between a CMTS and a subscriber's cable modem (CM) guaranteed application specific quality of service (QoS), known as Dynamic Quality of Service (DQoS). PCMM offers (MSOs) the ability to deliver new services using existing cable infrastructure. MSOs have already begun to apply PCMM technology for expanding their multimedia service offerings.

[blocked URL](#)

PCMM Overview

The PCMM architecture comprises the following components:

- The Application Manager, which specifies QoS requirements to the Policy Server on a per-application basis.
- The Policy Server, which allocates network resources per subscriber and per application, ensuring that consumption meets MSO priorities.
- The Cable Modem Termination System (CMTS), which enforces policies according to bandwidth capacity.
- The Cable Modem, which resides on the client side and connects the client's network to the cable system.

PacketCable Multimedia defines a service delivery framework that provides general-purpose QoS, event-based accounting, and security functionality founded upon the mechanisms defined in PacketCable 1.x. However, due to the broader spectrum of applications and services addressed by this initiative, each of these functional areas has been revisited and generalized for the present purposes. Telephony-specific requirements and interfaces (e.g., call signaling, PSTN interconnection and electronic surveillance) are not part of PacketCable Multimedia, while core functionality such as QoS resource management mechanisms, has been enhanced. Throughout this process, one of the primary objectives of this work has been to leverage and reuse as much of the existing body of PacketCable 1.x investment, knowledge base, and technical functionality as possible. Key features of the described Multimedia service delivery framework include:

- Simple, powerful access to DOCSIS QoS mechanisms supporting both time and volume-based network resource authorizations,
- Abstract, event-based network resource auditing and management mechanisms,
- A robust security infrastructure that provides integrity and appropriate levels of protection across all interfaces.

The goal of this project is to utilize the OpenDayLight controller platform as for the Application Manager and parts of the Policy Server and leverage the as many existing components offered by the platform.

The initial southbound transport has been written to the following version of the specification: <http://www.cablelabs.com/wp-content/uploads/specdocs/PKT-SP-MM-I05-091029.pdf>

Scope

- Provision a CMTS and an associated Cable Modem to form a Subscriber service for which (service) flows can be established and metered.
- Flow Programmer match-only for managing DOCSIS (service) flows
- Northbound APIs for provisioning CMTS network elements
 - HTML Provisioning Interface or some Python RESTful examples
- Northbound APIs for provisioning Service Flow values and types
- Northbound APIs for provisioning QoS (or metering) parameters
- SAL extensions for DOCSIS specific data model and configuration APIs
- Southbound PCMM/COPS transport plugin

Non-Goals

- Functional COPS driver
- CMTS PCMM COPS emulator
- Configuring a CMTS using Netconf

Work Flow Example

[blocked URL](#)

- Restful POST from Cable SDN Application Logic to DOCSIS Abstraction Layer to add a new CMTS
- Restful PUT from Provisioning App to Cable SDN Application Logic
- Restful PUT from Cable SDN Application Logic to Flow Programmer to Create Flow
- Flow Programmer Dispatches Request to SAL
- SAL Routes Message to South Bound PCMM/COPS Plugin Based on Node Type
- PCMM/COPS Plugin Forms a Gate Message to Create a Service Flow and Sends via COPS to CMTS
- DSx control message is sent to the Cable Modem
- A Flow Status is returned to Cable SDN Application Logic

For example, add L2VPN Attributes (ID/Type/Value Encapsulation types, VLAN ID, etc).

- PCMM/COPS Plugin forms a Gate Message to Modify a Service Flow
- DSx control message is sent to the Cable Modem

Design Details

DOCSIS SDN Application Logic

This component is introduced at the Network Applications Orchestrations 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](#).

[blocked URL](#)

See [PCMM Augmented FlowProgrammer](#)

DOCSIS Abstraction Layer

This component is introduced at the Controller Platform layer. This component manages the DOCSIS (and PCMM) specific attributes, such as store and change the default QoS values that are applied to Service Flows, and addition and removal of CMTS network elements. This module exports a DOCSIS specific northbound 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](#).

Information and Data Models

Developing a Model

Information and Data Models

OpenDayLight Models

[Model Driven SAL Reference](#)

Developing a Model

[blocked URL](#)

CMTS Inventory

[CTMS Inventory](#) Provisioning model is a draft adopted from OF-Config and contains some parameters for the future features that include interfacing to being able receive SNMP traps, read/write SNMP oid and configure using CLI. These services might even be provided by another southbound plugin. Unlike Openflow, discovery is not derived from a switch and provisioning is top down.

[Flow Match Types](#) and is a modification of the existing Openflow match action as a result of the comparison with Openflow.

[Service Types](#) are the northbound methods.

[Flow Traffic Profile](#) represents the various DOCSIS service flows that can be created.

[PacketCable Service](#) represents the northbound RPC service.

OpenDayLight Metering

QoS (or Metering) are parameters pertaining to a service flow types. Is there a way to extended OF metering to adopt DOCSIS Dynamic QoS?

- Metering References
 - [Meter Types](#)
 - [Meter Management](#)
 - [Meter Statistics](#)
- - [Meter Service](#)

[blocked URL](#)

PCMM Traffic Profile

[blocked URL](#)

PCMM Service Flow Types

Openflow Match Action and PCMM Service Flow Comparison

The following tables are comparisons with Openflow and PCMM ServiceFlows types. IPv6 is considered part of this comparison.

Openflow Match

OpenFlow	PCMM Service Flow
dIDst (MAC address)	no
dISrc (MAC address)	no

vlanId	no
vlanPri	no
tosBits	nwDscp
	SF Type
nwProto (TCP or UDP protocol)	yes
ingressPort	CMIM
priority	yes
etherType	no
nwDst (Dest IP Address)	yes
nwDstMask	yes
nwSrc (Src IP Address)	yes
nwSrcMask	yes
tpSrc (Source Port)	sourcePortStart
	sourcePortEnd
tpDst (Destiation Port)	sourcePortStart
	sourcePortEnd
hardTimeout	no
idleTimeout	Timeout active QoS parameters
cookie	no
	Activation State
	IPv6: Next Header Type, Flow Label, Flags,

Openflow Action

OpenFlow	PCMM Service Flow
drop	yes UDC
loopback	
flood	
floodAll	
controller	
interface	
software path	
hardware path	
output	
enqueue	
setDISrc	
setDIDst	
setVlan	In L2VPN, but not in PCMM
setVlanPcp	
setVlanCif	
stripVlan	
pushVlan	
setDIType	
setNwSrc	
setNwDst	
setNwTos	

setTpSrc	
setTpDst	
setNextHop	
pushMpls	
popMpls	

PCMM COPS Information Model

The [CableLabs PCMM Specification](#) defines the COPS PDP to PEP interface via a message set and set of object definitions. The objects carried in the COPS message set can be viewed as the PCMM COPS Information Model for the PCMM COPS interface between the Policy Manager and CMTS. The objects are represented in the table below.

Object Name	Description	ODL YANG Exist?
TransactionID	Transaction Identifier used by the Policy Server to match responses from the CMTS to the previous requests.	Southbound Driver Internalized
AMID	Application Manager ID including the Application Manager Tag (Application responsible for handling the session) and Application Type (Type of application this gate is associated with).	Southbound Driver Internalized
SubscriberID	IPv4 or IPv6 address of the Subscriber for the service request (i.e., CM or CPE of Subscriber)	
GateID	Gate Identifier referenced in the command message or referenced by the CMTS for a response message.	Southbound Driver Internalized
GateSpec	<p>Defines the following attributes:</p> <ul style="list-style-type: none"> • Direction <ul style="list-style-type: none"> ◦ Downstream Gate ◦ Upstream Gate • DSCP/TOS <ul style="list-style-type: none"> ◦ Enabled/Disabled • SessionClassID <ul style="list-style-type: none"> ◦ Defines the proper admission control policy or <p>parameters to be applied for this gate. Attributes include:</p> <ul style="list-style-type: none"> • ◦ <ul style="list-style-type: none"> ▪ Priority ▪ Preemption ▪ Configurable • DSCP/TOS Overwrite <ul style="list-style-type: none"> ◦ Used to identify particular bits within the IPv4 DSCP/TOS byte • DSCP/TOS Mask <ul style="list-style-type: none"> ◦ Used in conjunction with the DSCP/TOS Overwrite • Timer T1 • Timer T2 • Timer T3 • Timer T4 <ul style="list-style-type: none"> ◦ Timers defined per Gate Transition Diagram 	Southbound Driver Internalized
Classifiers	Classifier, Extended Classifier or IPv6 Classifier	
Classifier	<ul style="list-style-type: none"> • Protocol ID • DSCP/TOS Field • DSCP/TOS Mask • Source IPv4 Address • Destination IPv4 Address • Source Port • Destination Port • Priority 	See Model

Extended Classifier	<ul style="list-style-type: none"> • Protocol ID • DSCP/TOS Field • DSCP/TOS Mask • IPv4 Source Address • IPv4 Source Mask • IPv4 Destination Address • IPv4 Destination Mask • Source Port Start • Source Port End • Destination Port Start • Destination Port End • ClassifierID • Priority • Activation State • Action 	See Model
IPv6 Classifier	<ul style="list-style-type: none"> • Flags • tc-low • tc-high • tc-mask • Flow Label • Next Header Type • Source Prefix Length • Destination Prefix Length • IPv6 Source Address • IPv6 Destination Address • Source Port Start • Source Port End • Destination Port Start • Destination Port End • ClassifierID • Priority • Activation State • Action 	See Model
Traffic Profiles	FlowSpec, DOCSIS Service Class Name, DOCSIS-specific parameters or Upstream Drop	
Flow Spec	<ul style="list-style-type: none"> • Envelope • Service Number • Authorized Envelope <ul style="list-style-type: none"> ◦ Token Bucket Rate (r) ◦ Token Bucket Size (b) ◦ Peak Data Rate (p) ◦ Minimum Policed Unit (m) ◦ Maximum Packet Size (M) ◦ Rate (R) ◦ Slack Term (S) • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model
DOCSIS Service Class Name	<ul style="list-style-type: none"> • Envelope • Service Class Name 	See Model
Best Effort Service	<ul style="list-style-type: none"> • Envelope • Authorized Envelope <ul style="list-style-type: none"> ◦ Traffic Priority ◦ Request Transmission Policy ◦ Maximum Sustained Traffic Rate ◦ Maximum Traffic Burst ◦ Minimum Reserved Traffic Rate ◦ Assumed Minimum Reserved Traffic Rate Packet Size ◦ Maximum Concatenated Burst ◦ Upstream Peak Traffic Rate ◦ Required Attribute Mask ◦ Forbidden Attribute Mask ◦ Attribute Aggregation Rule Mask ◦ Minimum Buffer ◦ Target Buffer ◦ Maximum Buffer • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model

Non-Real-Time Polling Service	<ul style="list-style-type: none"> • Envelope • Authorized Envelope <ul style="list-style-type: none"> ◦ Traffic Priority ◦ Request Transmission Policy ◦ Maximum Sustained Traffic Rate ◦ Maximum Traffic Burst ◦ Minimum Reserved Traffic Rate ◦ Assumed Minimum Reserved Traffic Rate Packet Size ◦ Maximum Concatenated Burst ◦ Nominal Polling Interval ◦ Upstream Peak Traffic Rate ◦ Required Attribute Mask ◦ Forbidden Attribute Mask ◦ Attribute Aggregation Rule Mask ◦ Minimum Buffer ◦ Target Buffer ◦ Maximum Buffer • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model
Real-Time Polling Service	<ul style="list-style-type: none"> • Envelope • Authorized Envelope <ul style="list-style-type: none"> ◦ Request Transmission Policy ◦ Maximum Sustained Traffic Rate ◦ Maximum Traffic Burst ◦ Minimum Reserved Traffic Rate ◦ Assumed Minimum Reserved Traffic Rate Packet Size ◦ Maximum Concatenated Burst ◦ Nominal Polling Interval ◦ Tolerated Poll Jitter ◦ Upstream Peak Traffic Rate ◦ Required Attribute Mask ◦ Forbidden Attribute Mask ◦ Attribute Aggregation Rule Mask ◦ Minimum Buffer ◦ Target Buffer ◦ Maximum Buffer • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model
Unsolicited Grant Service	<ul style="list-style-type: none"> • Envelope • Authorized Envelope <ul style="list-style-type: none"> ◦ Request Transmission Policy ◦ Unsolicited Grant Size ◦ Grants/Interval ◦ Nominal Grant Interval ◦ Tolerated Grant Interval ◦ Upstream Peak Traffic Rate ◦ Required Attribute Mask ◦ Forbidden Attribute Mask ◦ Attribute Aggregation Rule Mask ◦ Minimum Buffer ◦ Target Buffer ◦ Maximum Buffer • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model
Unsolicited Grant Service with Activity Detection	<ul style="list-style-type: none"> • Envelope • Authorized Envelope <ul style="list-style-type: none"> ◦ Request Transmission Policy ◦ Unsolicited Grant Size ◦ Grants/Interval ◦ Nominal Grant Interval ◦ Tolerated Grant Jitter ◦ Nominal Polling Interval ◦ Tolerated Poll Jitter ◦ Upstream Peak Traffic Rate ◦ Required Attribute Mask ◦ Forbidden Attribute Mask ◦ Attribute Aggregation Rule Mask ◦ Minimum Buffer ◦ Target Buffer ◦ Maximum Buffer • <i>Optional</i> Reserved Envelope • <i>Optional</i> Committed Envelope 	See Model

Downstream Service	<ul style="list-style-type: none"> Envelope Authorized Envelope <ul style="list-style-type: none"> Traffic Priority Downstream Resequencing Maximum Sustained Traffic Rate Maximum Traffic Burst Minimum Reserved Traffic Rate Assumed Minimum Reserved Traffic Rate Packet Size Maximum Downstream Latency Downstream Peak Traffic Rate Required Attribute Mask Forbidden Attribute Mask Attribute Aggregation Rule Mask Minimum Buffer Target Buffer Maximum Buffer Optional Reserved Envelope Optional Committed Envelope 	See Model
Upstream Drop	<ul style="list-style-type: none"> Envelope 	See Model
IPv4 Event Generation Info	<ul style="list-style-type: none"> Primary Record Keeping Server IPv4 Address Primary Record Keeping Server Port Secondary Record Keeping Server IPv4 Address Secondary Record Keeping Server Port Billing Correlation ID 	No
IPv6 Event Generation Info	<ul style="list-style-type: none"> Primary Record Keeping Server IPv6 Address Primary Record Keeping Server Port Secondary Record Keeping Server IPv6 Address Secondary Record Keeping Server Port Billing Correlation ID 	No
Volume-Based Usage Limit	<ul style="list-style-type: none"> Usage Limit (Kb) 	No
Time-Based Usage Limit	<ul style="list-style-type: none"> Time Limit (Sec) 	No
Opaque Data	<ul style="list-style-type: none"> Opaque Data <ul style="list-style-type: none"> Information the Policy Server or Application Manager might store on a CMTS. 	No
Gate Time Info	<ul style="list-style-type: none"> Time Committed (Sec) 	
Gate Usage Info	<ul style="list-style-type: none"> Octet Count (bytes) 	
PacketCable Error	<ul style="list-style-type: none"> Error-Code <ul style="list-style-type: none"> Pre-defined error code 	Southbound Driver Internalized
Gate State	<ul style="list-style-type: none"> State Reason 	
Version Info	<ul style="list-style-type: none"> Major Version Number Minor Version Number 	Southbound Driver Internalized
PSID	<ul style="list-style-type: none"> PSID which uniquely identifies the Policy Server 	
Synch Options	<ul style="list-style-type: none"> Report Type Synch Type 	
Msg Receipt Key	<ul style="list-style-type: none"> Msg Receipt Key 	

UserID	<ul style="list-style-type: none"> UserID which identifies the user associated with the gate 	Southbound Driver Internalized
SharedResourceID	<ul style="list-style-type: none"> SharedResourceID is assigned by the CMTS for Multicast Gate requests. 	

Use Cases

Proactive QoS

Predefined flows with known quality of service parameters.

[blocked URL](#)

Reactive QoS

Use OpenFlow packetin to trigger a PCMM and Openflow flow creation.

[blocked URL](#)

Proof of Concept

On 21 October 2013, a proof of concept PCMM southbound plugin was demonstrated at SCTE. This demonstration "shows" network flow manipulation by controlling DOCSIS video traffic flows from ODL and applying different PCMM metering parameters to creating a video image that was either degraded or high quality.

[blocked URL](#)

Research

Using Netconf to configure CCAP

See [Yang for CCAP](#)

CCAP and DOCSIS 3.1 CMTS YANG Model Support

Since both devices optionally support the NETCONF protocol, these cable devices also *optionally* support YANG data models which are translated into mandatory supported XML Schema data models. The OpenDaylight framework also uses YANG models for their data model representation and tooling.

CCAP PCMM Configuration Management Information Model

The CableLabs CCAP specification defines a UML Class Diagram Information Model for configuration management of PCMM in the CCAP device. This is represented in the figure below.

[blocked URL](#)

The above Information Model is translated into the YANG configuration management data module and corresponding XML Schema. Therefore, these objects and attributes can be provisioned via XML configuration files and *optionally*, if supported, via NETCONF messaging.

CCAP

Latest (2014-04-02) CCAP Yang file.

Error creating thumbnail: Invalid thumbnail parameters

CCAP

[blocked URL](#)

CCAP events

Using SNMP for SubscriberID (Cable Modem) Discovery

What is required in defining a service flow is the SubscriberID (or IP address). It would be useful to discover the Cable Modems (CM) that are being hosted and managed by a provisioned CMTS.

<http://www.cisco.com/en/US/docs/cable/cmts/mib/reference/guide/ubrmibb.html>

```

<docsIfCmtsCmStatusValue>
<name>Registered Modems</name>
<method>walk</method>
<source>value</source>
<direction>output</direction>
<oid>1.3.6.1.2.1.10.127.1.3.3.1.9</oid>
</docsIfCmtsCmStatusValue>

```

DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.2.1	docsIfCmtsCmStatusMacAddress.1
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.2.2	docsIfCmtsCmStatusMacAddress.2
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.2.3	docsIfCmtsCmStatusMacAddress.3
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.3.1	docsIfCmtsCmStatusIpAddress.1
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.3.2	docsIfCmtsCmStatusIpAddress.2
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.3.3	docsIfCmtsCmStatusIpAddress.3
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.4.1	docsIfCmtsCmStatusDownChannelIfIndex.1
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.4.2	docsIfCmtsCmStatusDownChannelIfIndex.2
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.4.3	docsIfCmtsCmStatusDownChannelIfIndex.3
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.5.1	docsIfCmtsCmStatusUpChannelIfIndex.1
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.5.2	docsIfCmtsCmStatusUpChannelIfIndex.2
DOCS-IF-MIB	1.3.6.1.2.1.10.127.1.3.3.1.5.3	docsIfCmtsCmStatusUpChannelIfIndex.3

Expressing Support and Interest in Project

- Prasanna Mucharikar (Cisco)
- Mohammad Kabir Chowdhury (Comcast)
- Joshua Sholes (Comcast)
- Sameer Patel (Comcast)
- Jeff Finkelstein (Cox)
- Sangeeta Ramakrishnan (Cisco)
- Brian Otte (CableLabs)
- Alon Bernstein (Cisco)
- Eduardo Jacob (The University of the Basque)
- Jeff Pedigo (Applied Broadband)
- Jason Schnitzer (Applied Broadband)

Resources Committed (developers committed to working)

- [Thomas Kee irc:xsited](#)

Initial Committers

- [Thomas Kee irc:xsited](#)

Vendor Neutral

- No vendor package names in code
- No vendor branding / trademark present in code or output of build
- No vendor branding / trademark present in documentation

Meets Board Policy (including IPR)