

BESS Working Group
Internet-Draft
Intended Status: Standard Track
Expires: September 13, 2017

D. Wang
Y. Liu
Huawei Technologies

March 13, 2017

A YANG data model for Multicast in Virtual Private LAN Service
draft-wang-bess-vpls-multicast-yang-00

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on September 10, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

VPLS Multicast Yang Model

March 2017

Abstract

This document defines a YANG data model that can be used to configure and manage multicast devices in Virtual Private LAN Service.

Table of Contents

1.	Introduction	2
1.1.	Requirements Language.....	2
1.2.	Terminology	3
2.	Design of Data model.....	3
2.1.	Scope of model	3
2.2.	Optional capabilities.....	3
2.3.	Position of address family in hierarchy.....	4
3.	Module Structure	4
3.1.	IGMP and MLD Snooping Configuration.....	4
3.2.	IGMP and MLD Snooping Operational State.....	9
4.	IGMP and MLD YANG Modules.....	19
5.	Security Considerations.....	65
6.	IANA Considerations	65
7.	References	65
7.1.	Normative References.....	65
7.2.	Informative References.....	66

[1.](#) Introduction

YANG [[RFC6020](#)] [[RFC6087](#)] is a data definition language that was introduced to model the configuration and running state of a device managed using NETCONF [[RFC6241](#)]. YANG is now also being used as a component of wider management interfaces, such as CLIs.

This document defines a draft YANG data model that can be used to configure and manage multicast devices in Virtual Private LAN Service. Currently this model is incomplete, but it will support the core IGMP and MLD snooping protocols in Virtual Private LAN Service, as well as many other features mentioned in separate multicast RFCs. Non-core features are defined as optional in the provided data model.

[1.1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [[RFC2119](#)].

[1.2](#). Terminology

The terminology for describing YANG data models is found in[RFC6020].

This draft employs YANG tree diagrams, which are explained in [I-D.ietf-netmod-rfc6087bis].

[2](#). Design of Data model

[2.1](#). Scope of model

The model covers IGMP and MLD snooping[RFC4541], and multicast in BGP based VPLS[RFC4761] or LDP based VPLS[RFC4762].

The representation of some of extension features is not completely specified in this draft of the data model. This model is being circulated in its current form for early oversight and review of the basic hierarchy.

The operational state fields of this model are also incomplete, though the structure of what has been written may be taken as representative of the structure of the model when complete.

This model does not cover other multicast protocols in VPLS such as [[RFC7117](#)] or [[RFC7524](#)] etc., these will be specified in separate documents.

[2.2](#). Optional capabilities

This model is designed to represent the capabilities of IGMP and MLD snooping devices in VPLS with various specifications, including some with basic subsets of the IGMP and MLD snooping protocols in VPLS. The main design goals of this draft are that any major now-existing implementation may be said to support the basic model, and that the configuration of all implementations meeting the specification is

easy to express through some combination of the features in the basic model and simple vendor augmentations.

There is also value in widely-supported features being standardized, to save work for individual vendors, and so that mapping between different vendors' configuration is not needlessly complicated. Therefore these modules declare a number of features representing capabilities that not all deployed devices support.

The extensive use of feature declarations should also substantially simplify the capability negotiation process for a vendor's IGMP and MLD snooping in VPLS implementations.

On the other hand, operational state parameters are not so widely designated as features, as there are many cases where the defaulting of an operational state parameter would not cause any harm to the system, and it is much more likely that an implementation without native support for a piece of operational state would be able to derive a suitable value for a state variable that is not natively supported.

For the same reason, wide constant ranges (for example, timer maximum and minimum) will be used in the model. It is expected that vendors will augment the model with any specific restrictions that might be required. Vendors may also extend the features list with proprietary extensions.

[2.3](#). Position of address family in hierarchy

The current draft contains IGMP or MLD snooping in VPLS as separate schema branches in the structure. The reason for this is to make it easier for implementations which may optionally choose to support specific address families. And the names of objects may be different between the ipv4 (IGMP snooping) and ipv6 (MLD snooping) address families.

[3](#). Module Structure

[3.1](#). VPLS Multicast Configuration

The IGMP or MLD snooping in VPLS modules define the configuration options in a two-level hierarchy as listed below:

Global level: IGMP or MLD snooping configuration attributes for the entire system and applicable to all of the l2vpn instances.

L2vpn instance level: IGMP or MLD snooping configuration attributes applicable to a specific l2vpn instance.

Where fields are not genuinely essential to protocol operation, they are marked as optional. Some fields will be essential but have a default specified, so that they need not be configured explicitly. The module structure also applies, where applicable, to the operational state and notifications as well.

The IGMP or MLD snooping in VPLS model will augments
"/l2vpn:l2vpn:".

```
augment /l2vpn:l2vpn:
  +--rw igmp-snooping
  |   +--rw global
  |   |   +--rw enable?                boolean {global-admin-enable}?
  |   |   +--rw max-entry?             uint32
```

```
| | +--rw bandwidth?                uint32
| | +--rw ssm-mapping-policys
| | |   +--rw ssm-mapping-policy* [name]
| | |   |   +--rw name                string
| | |   |   +--rw ssm-map* [group-address group-mask-length source-address]
| | |   |   |   +--rw group-address    inet:ipv4-address
| | |   |   |   +--rw group-mask-length uint8
| | |   |   |   +--rw source-address   inet:ipv4-address
| | |   +--rw unspecified-channel-deny? boolean
| | |   +--rw global-cac-configs
| | |   |   +--rw global-cac-config* [channel-name channel-mode]
| | |   |   |   +--rw channel-name      string
| | |   |   |   +--rw channel-mode      channel-mode
| | |   |   |   +--rw max-entry?        uint32
| | |   |   +--rw channel-policys
| | |   |   |   +--rw channel-policy* [group-address group-mask-length source-address]
| | |   |   |   |   +--rw group-address    inet:ipv4-address
| | |   |   |   |   +--rw group-mask-length uint8
| | |   |   |   |   +--rw source-address   inet:ipv4-address
| | |   |   |   |   +--rw source-mask-length uint8
| | |   |   |   +--rw bandwidth-each-entry? uint32
| | |   +--rw send-query-source?    inet:ipv4-address
```

```

|  +---rw l2vpn-multicast-instances
|      +---rw l2vpn-multicast-instance* [name type]
|          +---rw name                string
|          +---rw type                identityref
|          +---rw enable?             boolean
|          +---rw version?            uint8
|          +---rw snooping-mode?      snooping-mode
|          +---rw ssm-policy?         string
|          +---rw group-policy?       string
|          +---rw group-policy-version? uint8
|          +---rw ip-policy?          string
|          +---rw query-ip-policy?    string
|          +---rw last-member-query-interval? uint8
|          +---rw query-interval?     uint16
|          +---rw query-max-response-time? uint8
|          +---rw router-aging-time?  uint16
|          +---rw prompt-leave?       boolean
|          +---rw router-learning?    boolean
|          +---rw require-router-alert? boolean
|          +---rw send-router-alert?  boolean
|          +---rw ssm-mapping?        boolean
|          +---rw ssm-mapping-policy? string
|          +---rw robustness-variable? uint8
|          +---rw source-life-time?   uint32
|          +---rw unspecified-channel-deny? boolean
|          +---rw max-entry?          uint32
|          +---rw max-entry-except-policy? string
|          +---rw bandwidth-lose-mode? boolean

```

```

|      +---rw proxy-enable?           boolean
|      +---rw proxy-router-protocol-pass? boolean
|      +---rw query-enable?           boolean
|      +---rw querier-election?       boolean
|      +---rw report-suppress?        boolean
|      +---rw ssm-map* [group-address group-mask-length source-address]
|          | +---rw group-address      inet:ipv4-address
|          | +---rw group-mask-length  uint8
|          | +---rw source-address     inet:ipv4-address
|      +---rw static-pw-router* [peer-ip pw-id]
|          | +---rw peer-ip            inet:ip-address
|          | +---rw pw-id              uint32
|      +---rw static-ac-router-port* [name pe-id ce-id]
|          | +---rw name               if:interface-ref
|          | +---rw pe-id              uint16

```



```

|         | +--rw bandwidth      uint32
|         | +--rw vlan-bitmap?   string
|         +--rw dot1q-bandwidth* [bandwidth]
|         | +--rw bandwidth      uint32
|         | +--rw vlan-bitmap?   string
|         +--rw dot1q-channel-bandwidth* [channel-name bandwidth]
|         | +--rw channel-name   string
|         | +--rw bandwidth      uint32
|         | +--rw vlan-bitmap?   string
+--rw mld-snooping
  +--rw global
    | +--rw enable?              boolean {global-admin-enable}?
    | +--rw max-entry?          uint32
    | +--rw bandwidth?          uint32
    | +--rw ssm-mapping-policys
    | | +--rw ssm-mapping-policy* [name]
    | | | +--rw name            string
    | | | +--rw ssm-map* [group-address group-mask-length source-address]
    | | | +--rw group-address    inet:ipv6-address
    | | | +--rw group-mask-length uint8
    | | | +--rw source-address   inet:ipv6-address
    | +--rw unspecified-channel-deny? boolean
    +--rw global-cac-configs
    | +--rw global-cac-config* [channel-name channel-mode]
    | | +--rw channel-name      string
    | | +--rw channel-mode      channel-mode
    | | +--rw channel-policys
    | | | +--rw channel-policy* [group-address group-mask-length source-address]
    | | | +--rw group-address    inet:ipv6-address
    | | | +--rw group-mask-length uint8
    | | | +--rw source-address   inet:ipv6-address
    | | | +--rw source-mask-length uint8
    | | | +--rw bandwidth-each-entry? uint32
    | +--rw send-query-source?   inet:ipv6-address
+--rw l2vpn-multicast-instances
  +--rw l2vpn-multicast-instance* [name type]
    +--rw name                    string
    +--rw type                    identityref
    +--rw enable?                 boolean
    +--rw version?                uint8
    +--rw snooping-mode?          snooping-mode
    +--rw ssm-policy?             string

```

```

+--rw group-policy?             string

```



```

+--rw group-policy-version?      uint8
+--rw ip-policy?                 string
+--rw query-ip-policy?           string
+--rw last-member-query-interval? uint8
+--rw query-interval?            uint16
+--rw query-max-response-time?   uint8
+--rw router-aging-time?         uint16
+--rw prompt-leave?              boolean
+--rw router-learning?           boolean
+--rw require-router-alert?      boolean
+--rw send-router-alert?         boolean
+--rw ssm-mapping?               boolean
+--rw ssm-mapping-policy?        string
+--rw robustness-variable?       uint8
+--rw source-life-time?          uint32
+--rw unspecified-channel-deny?   boolean
+--rw max-entry?                 uint32
+--rw max-entry-except-policy?   string
+--rw bandwidth-lose-mode?       boolean
+--rw proxy-enable?              boolean
+--rw proxy-router-protocol-pass? boolean
+--rw query-enable?              boolean
+--rw querier-election?          boolean
+--rw report-suppress?           boolean
+--rw ssm-map* [group-address group-mask-length source-address]
|   +--rw group-address          inet:ipv6-address
|   +--rw group-mask-length      uint8
|   +--rw source-address         inet:ipv6-address
+--rw static-pw-router* [peer-ip pw-id]
|   +--rw peer-ip               inet:ip-address
|   +--rw pw-id                 uint32
+--rw static-ac-router-port* [name pe-id ce-id]
|   +--rw name                  if:interface-ref
|   +--rw pe-id                 uint16
|   +--rw ce-id                 uint16
+--rw static-pw-group* [peer-ip pw-id]
|   +--rw peer-ip               inet:ip-address
|   +--rw pw-id                 uint32
|   +--rw static-group-sources
|       +--rw static-group-source* [group-address source-address]
|           +--rw group-address    inet:ipv6-address
|           +--rw source-address    inet:ipv6-address
+--rw static-ac-static-group* [name pe-id ce-id]
|   +--rw name                  if:interface-ref
|   +--rw pe-id                 uint16
|   +--rw ce-id                 uint16
|   +--rw static-group-sources
|       +--rw static-group-source* [group-address source-address]
|           +--rw group-address    inet:ipv6-address

```

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

|         +--rw source-address      inet:ipv6-address
+--rw cac-configs
|   +--rw cac-config* [channel-name channel-mode]
|     +--rw channel-name            string
|     +--rw channel-mode            channel-mode
|     +--rw channel-policys
|       +--rw channel-policy* [group-address group-mask-length
|         +--rw group-address        inet:ipv6-address
|         +--rw group-mask-length    uint8
|         +--rw source-address        inet:ipv6-address
|         +--rw source-mask-length    uint8
|         +--rw bandwidth-each-entry? uint32
+--rw interfaces
  +--rw interface* [interface]
    +--rw interface                if:interface-ref
    +--rw bandwidth-lose-mode?     boolean
    +--rw max-entry?               uint32
    +--rw group-policy?            string
    +--rw group-policy-version?    uint8
    +--rw (group-policy-vlan)?
    |   +--:(qinq-range)
    |   |   +--rw qinq-vlan* [pe-id]
    |   |   |   +--rw pe-id        uint16
    |   |   |   +--rw vlan-bitmap? string
    |   |   +--:(dot1q-range)
    |   |   |   +--rw vlan-bitmap? string
    +--rw trunk-bandwidth* [bandwidth]
    |   +--rw bandwidth            uint32
    |   +--rw vlan-bitmap?        string
    +--rw dot1q-bandwidth* [bandwidth]
    |   +--rw bandwidth            uint32
    |   +--rw vlan-bitmap?        string
    +--rw dot1q-channel-bandwidth* [channel-name bandwidth]
    |   +--rw channel-name        string
    |   +--rw bandwidth            uint32
    |   +--rw vlan-bitmap?        string

```

[3.2.](#) VPLS Multicast Operational State

The IGMP or MLD snooping in VPLS module contains operational state information also in a two-level hierarchy as mentioned earlier.

Global level: IGMP or MLD snooping operational state attributes for the entire system.

L2vpn instance level: IGMP or MLD snooping operational state attributes specific to the given l2vpn instance.

augment /l2vpn:l2vpn-state:

```

+--ro igmp-snooping
|   +--ro global
|   |   +--ro enable?                boolean {global-admin-enable}?
|   |   +--ro max-entry?             uint32
|   |   +--ro bandwidth?             uint32
|   |   +--ro ssm-mapping-policys
|   |   |   +--ro ssm-mapping-policy* [name]
|   |   |   |   +--ro name            string
|   |   |   |   +--ro ssm-map* [group-address group-mask-length source-address]
|   |   |   |   |   +--ro group-address      inet:ipv4-address
|   |   |   |   |   +--ro group-mask-length  uint8
|   |   |   |   |   +--ro source-address    inet:ipv4-address
|   |   |   +--ro unspecified-channel-deny? boolean
|   |   +--ro global-cac-configs
|   |   |   +--ro global-cac-config* [channel-name channel-mode]
|   |   |   |   +--ro channel-name        string
|   |   |   |   +--ro channel-mode        channel-mode
|   |   |   |   +--ro max-entry?          uint32
|   |   |   |   +--ro channel-policys
|   |   |   |   |   +--ro channel-policy* [group-address group-mask-length source-address]
|   |   |   |   |   |   +--ro group-address      inet:ipv4-address
|   |   |   |   |   |   +--ro group-mask-length  uint8
|   |   |   |   |   |   +--ro source-address    inet:ipv4-address
|   |   |   |   |   |   +--ro source-mask-length  uint8
|   |   |   |   |   |   +--ro bandwidth-each-entry? uint32
|   |   |   +--ro dynamic-entry?          uint32
|   |   |   +--ro dynamic-bandwidth?       uint32
|   |   |   +--ro channel-cac* [channel-name channel-mode]
|   |   |   |   +--ro channel-name        string
|   |   |   |   +--ro channel-mode        channel-mode
|   |   |   |   +--ro dynamic-entry-count? uint32
|   |   |   |   +--ro dynamic-bandwidth?   uint32
|   |   |   +--ro send-query-source?      inet:ipv4-address
|   +--ro l2vpn-multicast-instances
|   |   +--ro l2vpn-multicast-instance* [name type]
|   |   |   +--ro name                    string
|   |   |   +--ro type                    identityref

```

	+++ro enable?	boolean
	+++ro version?	uint8
	+++ro snooping-mode?	snooping-mode
	+++ro ssm-policy?	string
	+++ro group-policy?	string
	+++ro group-policy-version?	uint8
	+++ro ip-policy?	string
	+++ro query-ip-policy?	string
	+++ro last-member-query-interval?	uint8
	+++ro query-interval?	uint16
	+++ro query-max-response-time?	uint8
	+++ro router-aging-time?	uint16
	+++ro prompt-leave?	boolean

	+++ro router-learning?	boolean
	+++ro require-router-alert?	boolean
	+++ro send-router-alert?	boolean
	+++ro ssm-mapping?	boolean
	+++ro ssm-mapping-policy?	string
	+++ro robustness-variable?	uint8
	+++ro source-life-time?	uint32
	+++ro unspecified-channel-deny?	boolean
	+++ro max-entry?	uint32
	+++ro max-entry-except-policy?	string
	+++ro bandwidth-lose-mode?	boolean
	+++ro proxy-enable?	boolean
	+++ro proxy-router-protocol-pass?	boolean
	+++ro query-enable?	boolean
	+++ro querier-election?	boolean
	+++ro report-suppress?	boolean
	+++ro querier-state	
	+++ro querierState?	boolean
	+++ro querierAddr?	inet:ip-address
	+++ro statistics	
	+++ro received	
	+++ro v1-report?	yang:counter64
	+++ro v2-report?	yang:counter64
	+++ro v3-report?	yang:counter64
	+++ro v1-query?	yang:counter64
	+++ro v2-query?	yang:counter64
	+++ro v3-query?	yang:counter64
	+++ro leave?	yang:counter64
	+++ro hello?	yang:counter64
	+++ro zero-source-query?	yang:counter64

```

|         | |   +--ro no-zero-source-query?   yang:counter64
|         | |   +--ro sent
|         | |       +--ro general-query?           yang:counter64
|         | |       +--ro source-group-specific-query? yang:counter64
|         | |       +--ro group--specific-query?    yang:counter64
|         +--ro ssm-map* [group-address group-mask-length source-address]
|         |   +--ro group-address      inet:ipv4-address
|         |   +--ro group-mask-length  uint8
|         |   +--ro source-address     inet:ipv4-address
|         +--ro static-pw-router-port* [peer-ip pw-id]
|         |   +--ro peer-ip      inet:ip-address
|         |   +--ro pw-id       uint32
|         +--ro static-ac-router-port* [name pe-id ce-id]
|         |   +--ro name        if:interface-ref
|         |   +--ro pe-id       uint16
|         |   +--ro ce-id       uint16
|         +--ro static-pw-group* [peer-ip pw-id]
|         |   +--ro peer-ip      inet:ip-address
|         |   +--ro pw-id       uint32
|         |   +--ro static-group-sources

```

```

|         |   +--ro static-group-source* [group-address source-address]
|         |   +--ro group-address      inet:ipv4-address
|         |   +--ro source-address     inet:ipv4-address
|         +--ro static-ac-group* [name pe-id ce-id]
|         |   +--ro name        if:interface-ref
|         |   +--ro pe-id       uint16
|         |   +--ro ce-id       uint16
|         |   +--ro static-group-sources
|         |       +--ro static-group-source* [group-address source-address]
|         |       +--ro group-address      inet:ipv4-address
|         |       +--ro source-address     inet:ipv4-address
|         +--ro cac-configs
|         |   +--ro cac-config* [channel-name channel-mode]
|         |   +--ro channel-name      string
|         |   +--ro channel-mode      channel-mode
|         |   +--ro max-entry?       uint32
|         |   +--ro channel-policys
|         |       +--ro channel-policy* [group-address group-mask-length
|         |       +--ro group-address      inet:ipv4-address
|         |       +--ro group-mask-length  uint8
|         |       +--ro source-address     inet:ipv4-address
|         |       +--ro source-mask-length uint8
|         |       +--ro bandwidth-each-entry? uint32

```

```

|      +---ro interfaces
|      | +---ro interface* [interface]
|      |   +---ro interface                                if:interface-ref
|      |   +---ro bandwidth-lose-mode?                    boolean
|      |   +---ro max-entry?                               uint32
|      |   +---ro group-policy?                           string
|      |   +---ro group-policy-version?                   uint8
|      |   +---ro (group-policy-vlan)?
|      |   | +---:(qinq-range)
|      |   | | +---ro qinq-vlan* [pe-id]
|      |   | |   +---ro pe-id                               uint16
|      |   | |   +---ro vlan-bitmap?                        string
|      |   | +---:(dot1q-range)
|      |   |   +---ro vlan-bitmap?                          string
|      |   +---ro trunk-bandwidth* [bandwidth]
|      |   | +---ro bandwidth                               uint32
|      |   | +---ro vlan-bitmap?                            string
|      |   +---ro dot1q-bandwidth* [bandwidth]
|      |   | +---ro bandwidth                               uint32
|      |   | +---ro vlan-bitmap?                            string
|      |   +---ro dot1q-channel-bandwidth* [channel-name bandwidth]
|      |   | +---ro channel-name                           string
|      |   | +---ro bandwidth                               uint32
|      |   | +---ro vlan-bitmap?                            string
|      |   +---ro dynamic-entry?                           uint32
|      |   +---ro dynamic-trunk-bandwidth* [vlan-id]
|      |   | +---ro vlan-id                                 uint16

```

```

|      |   +---ro bandwidth?                               uint32
|      | +---ro dynamic-dot1q-bandwidth* [vlan-id]
|      |   +---ro vlan-id                                 uint16
|      |   +---ro bandwidth?                               uint32
|      | +---ro dynamic-dot1q-channel-bandwidth* [channel-name]
|      |   +---ro channel-name                             string
|      |   +---ro vlan-bandwidth* [vlan-id]
|      |   +---ro vlan-id                                 uint16
|      |   +---ro bandwidth?                               uint32
|      +---ro (dynamic-router-port)?
|      | +---:(pw)
|      | | +---ro dynamic-pw-router-port* [peer-ip pw-id]
|      | |   +---ro peer-ip                               inet:ip-address
|      | |   +---ro pw-id                                 uint32
|      | |   +---ro expire?                               uint32
|      | |   +---ro up-time?                              uint32

```



```

|         +---ro address                               inet:ip
|         +---ro filter-mode?                         filter-m
|         +---ro up-time?                             uint32
|         +---ro group-timer?                         exitOrNo
|         +---ro expire?                             uint32
|         +---ro retransmission-count?               uint32
|         +---ro last-member-query?                  boolean
|         +---ro compatibility?                      compatib
|         +---ro v1-host-timer?                      exitOrNo
|         +---ro v2-host-timer?                      exitOrNo
|         +---ro last-source-member-query-timer?     exitOrNo
|         +---ro last-source-member-query?           boolean
|         +---ro sources
|             +---ro source* [address]
|                 +---ro address                     inet:ip
|                 +---ro expire?                     uint32
|                 +---ro up-time?                    uint32
|                 +---ro source-timer?               exitOrNo
|                 +---ro retransmission-count?       uint32
|                 +---ro last-source-member-query?   boolean
+---ro mld-snooping
    +---ro global
        | +---ro enable?                             boolean {global-admin-enable}?
        | +---ro max-entry?                           uint32
        | +---ro bandwidth?                           uint32
        | +---ro ssm-mapping-policys
        | | +---ro ssm-mapping-policy* [name]
        | | | +---ro name                             string
        | | | +---ro ssm-map* [group-address group-mask-length source-address]
        | | | | +---ro group-address                 inet:ipv6-address
        | | | | +---ro group-mask-length             uint8
        | | | | +---ro source-address                 inet:ipv6-address
        | +---ro unspecified-channel-deny?            boolean
        +---ro global-cac-configs
            | +---ro global-cac-config* [channel-name channel-mode]
            | | +---ro channel-name                   string
            | | +---ro channel-mode                     channel-mode
            | +---ro channel-policys
            | | +---ro channel-policy* [group-address group-mask-length source-address]
            | | | +---ro group-address                 inet:ipv6-address
            | | | +---ro group-mask-length             uint8

```

```

| |         +---ro source-address                 inet:ipv6-address
| |         +---ro source-mask-length             uint8

```



```

| |          +---ro bandwidth-each-entry?   uint32
| +---ro dynamic-entry?                     uint32
| +---ro dynamic-bandwidth?                 uint32
| +---ro channel-cac* [channel-name channel-mode]
| |   +---ro channel-name                   string
| |   +---ro channel-mode                   channel-mode
| |   +---ro dynamic-entry-count?          uint32
| |   +---ro dynamic-bandwidth?           uint32
| +---ro entry-count?                      uint32
| +---ro send-query-source?                 inet:ipv6-address
+---ro l2vpn-multicast-instances
    +---ro l2vpn-multicast-instance* [name type]
        +---ro name                         string
        +---ro type                         identityref
        +---ro enable?                     boolean
        +---ro version?                    uint8
        +---ro snooping-mode?              snooping-mode
        +---ro ssm-policy?                 string
        +---ro group-policy?              string
        +---ro group-policy-version?       uint8
        +---ro ip-policy?                 string
        +---ro query-ip-policy?           string
        +---ro last-member-query-interval? uint8
        +---ro query-interval?            uint16
        +---ro query-max-response-time?    uint8
        +---ro router-aging-time?         uint16
        +---ro prompt-leave?              boolean
        +---ro router-learning?            boolean
        +---ro require-router-alert?       boolean
        +---ro send-router-alert?          boolean
        +---ro ssm-mapping?                boolean
        +---ro ssm-mapping-policy?         string
        +---ro robustness-variable?        uint8
        +---ro source-life-time?           uint32
        +---ro unspecified-channel-deny?    boolean
        +---ro max-entry?                  uint32
        +---ro max-entry-except-policy?    string
        +---ro bandwidth-lose-mode?        boolean
        +---ro proxy-enable?               boolean
        +---ro proxy-router-protocol-pass? boolean
        +---ro query-enable?               boolean
        +---ro querier-election?           boolean
        +---ro report-suppress?            boolean
        +---ro querier-state
            | +---ro querierState?    boolean
            | +---ro querierAddr?     inet:ipv6-address
        +---ro statistics
            | +---ro received

```

```
| | +--ro v1-report?                yang:counter64
| | +--ro v2-report?                yang:counter64
| | +--ro v3-report?                yang:counter64
| | +--ro v1-query?                 yang:counter64
| | +--ro v2-query?                 yang:counter64
| | +--ro v3-query?                 yang:counter64
| | +--ro leave?                    yang:counter64
| | +--ro hello?                    yang:counter64
| | +--ro zero-source-query?        yang:counter64
| | +--ro no-zero-source-query?     yang:counter64
| +--ro sent
|   +--ro general-query?             yang:counter64
|   +--ro source-group-specific-query? yang:counter64
|   +--ro group--specific-query?     yang:counter64
+--ro ssm-map* [group-address group-mask-length source-address]
| +--ro group-address                inet:ipv6-address
| +--ro group-mask-length            uint8
| +--ro source-address                inet:ipv6-address
+--ro static-pw-router-port* [peer-ip pw-id]
| +--ro peer-ip                      inet:ip-address
| +--ro pw-id                        uint32
+--ro static-ac-router-port* [name pe-id ce-id]
| +--ro name                          if:interface-ref
| +--ro pe-id                         uint16
| +--ro ce-id                         uint16
+--ro static-pw-group* [peer-ip pw-id]
| +--ro peer-ip                      inet:ip-address
| +--ro pw-id                        uint32
| +--ro static-group-sources
|   +--ro static-group-source* [group-address source-address]
|     +--ro group-address              inet:ipv6-address
|     +--ro source-address              inet:ipv6-address
+--ro static-ac-group* [name pe-id ce-id]
| +--ro name                          if:interface-ref
| +--ro pe-id                         uint16
| +--ro ce-id                         uint16
| +--ro static-group-sources
|   +--ro static-group-source* [group-address source-address]
|     +--ro group-address              inet:ipv6-address
|     +--ro source-address              inet:ipv6-address
+--ro cac-configs
| +--ro cac-config* [channel-name channel-mode]
|   +--ro channel-name                string
|   +--ro channel-mode                channel-mode
|   +--ro channel-policys
|     +--ro channel-policy* [group-address group-mask-length
|       +--ro group-address            inet:ipv6-address
```

	+++ro group-mask-length	uint8
	+++ro source-address	inet:ipv6-address
	+++ro source-mask-length	uint8

```

|           +++ro bandwidth-each-entry?  uint32
+++ro dynamic-entry?                      uint32
+++ro dynamic-bandwidth?                  uint32
+++ro channel-cac* [channel-name channel-mode]
|   +++ro channel-name                    string
|   +++ro channel-mode                    channel-mode
|   +++ro dynamic-entry-count?           uint32
|   +++ro dynamic-bandwidth?             uint32
+++ro interfaces
|   +++ro interface* [interface]
|       +++ro interface                    if:interface-ref
|       +++ro bandwidth-lose-mode?        boolean
|       +++ro max-entry?                  uint32
|       +++ro group-policy?               string
|       +++ro group-policy-version?       uint8
|       +++ro (group-policy-vlan)?
|           |   +---:(qinq-range)
|           |   |   +++ro qinq-vlan* [pe-id]
|           |   |   |   +++ro pe-id        uint16
|           |   |   |   +++ro vlan-bitmap?  string
|           |   +---:(dot1q-range)
|           |   |   +++ro vlan-bitmap?      string
|       +++ro trunk-bandwidth* [bandwidth]
|           |   +++ro bandwidth            uint32
|           |   +++ro vlan-bitmap?         string
|       +++ro dot1q-bandwidth* [bandwidth]
|           |   +++ro bandwidth            uint32
|           |   +++ro vlan-bitmap?         string
|       +++ro dot1q-channel-bandwidth* [channel-name bandwidth]
|           |   +++ro channel-name         string
|           |   +++ro bandwidth            uint32
|           |   +++ro vlan-bitmap?         string
|       +++ro dynamic-entry?              uint32
|       +++ro dynamic-trunk-bandwidth* [vlan-id]
|           |   +++ro vlan-id              uint16
|           |   +++ro bandwidth?           uint32
|       +++ro dynamic-dot1q-bandwidth* [vlan-id]
|           |   +++ro vlan-id              uint16
|           |   +++ro bandwidth?           uint32
|       +++ro dynamic-dot1q-channel-bandwidth* [channel-name]

```

```

|         +--ro channel-name      string
|         +--ro vlan-bandwidth* [vlan-id]
|             +--ro vlan-id      uint16
|             +--ro bandwidth?   uint32
+--ro (dynamic-router-port)?
|   +--:(pw)
|   |   +--ro dynamic-pw-router-port* [peer-ip pw-id]
|   |   |   +--ro peer-ip      inet:ip-address
|   |   |   +--ro pw-id       uint32
|   |   |   +--ro expire?     uint32

```

```

|   |   +--ro up-time?      uint32
|   |   +--ro flag?        flagType
+--:(ac)
|   +--ro dynamic-ac-router-port* [name pe-id ce-id]
|       +--ro name          if:interface-ref
|       +--ro pe-id         uint16
|       +--ro ce-id         uint16
|       +--ro expire?       uint32
|       +--ro up-time?      uint32
|       +--ro flag?         flagType
+--ro (dynamic-group)?
+--:(pw-dynamic-group)
|   +--ro dynamic-pw-group* [peer-ip pw-id]
|       +--ro peer-ip      inet:ip-address
|       +--ro pw-id        uint32
|       +--ro groups
|           +--ro group* [address]
|               +--ro address          inet:ipv
|               +--ro filter-mode?     filter-m
|               +--ro up-time?          uint32
|               +--ro group-timer?     exitOrNo
|               +--ro expire?          uint32
|               +--ro retransmission-count? uint32
|               +--ro last-member-query? boolean
|               +--ro compatibility?   compatib
|               +--ro v1-host-timer?   exitOrNo
|               +--ro v2-host-timer?   exitOrNo
|               +--ro last-source-member-query-timer? exitOrNo
|               +--ro last-source-member-query? boolean
|               +--ro sources
|                   +--ro source* [address]
|                       +--ro address          inet:ipv
|                       +--ro expire?          uint32

```

```

|           +---ro up-time?                uint32
|           +---ro source-timer?           exitOrNo
|           +---ro retransmission-count?   uint32
|           +---ro last-source-member-query? boolean
+---:(ac-dynamic-group)
  +---ro dynamic-ac-dynamic-group* [name pe-id ce-id]
    +---ro name          if:interface-ref
    +---ro pe-id         uint16
    +---ro ce-id         uint16
    +---ro groups
      +---ro group* [address]
        +---ro address          inet:ipv
        +---ro filter-mode?     filter-m
        +---ro up-time?        uint32
        +---ro group-timer?     exitOrNo
        +---ro expire?         uint32
        +---ro retransmission-count? uint32

```

Wang & Liu, etc

Expires September 13, 2017

[Page 18]

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

+---ro last-member-query?           boolean
+---ro compatibility?               compatib
+---ro v1-host-timer?               exitOrNo
+---ro v2-host-timer?               exitOrNo
+---ro last-source-member-query-timer? exitOrNo
+---ro last-source-member-query?    boolean
+---ro sources
  +---ro source* [address]
    +---ro address          inet:ipv
    +---ro expire?         uint32
    +---ro up-time?        uint32
    +---ro source-timer?   exitOrNo
    +---ro retransmission-count? uint32
    +---ro last-source-member-query? boolean

```

4. VPLS Multicast YANG Modules

```

<CODE BEGINS> file "ietf-l2vpn-igmp-mld-snooping@2017-03-13.yang"
module ietf-l2vpn-igmp-mld-snooping {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn-igmp-mld-snooping";
  // replace with IANA namespace when assigned
  prefix l2vpn-igmp-mld-snooping;

```

```

import ietf-inet-types {
    prefix "inet";
}

import ietf-yang-types {
    prefix "yang";
}

import ietf-interfaces {
    prefix "if";
}

import ietf-l2vpn {
    prefix "l2vpn";
}

organization
    "IETF BESS Working Group";

contact
    "WG Web:    <http://tools.ietf.org/wg/bess/>
    WG List:    <mailto:bess@ietf.org>

    Editors:    ";

```

```

description
    "The module defines a collection of YANG definitions common for
    IGMP and MLD snooping in Virtual Private LAN Service.";

revision 2017-03-13 {
    description
        "Initial revision.";
    reference
        "RFC XXX: A YANG Data Model for VPLS Multicast";
}

/*
 * Features
 */
feature global-admin-enable {
    description
        "Support global configuration to enable or disable protocol.";
}

```

```

/*
 * Typedefs
 */
typedef ssm-map-ipv4-addr-type {
    type union {
        type enumeration {
            enum 'policy' {
                description
                "Source address is specified in SSM map policy.";
            }
        }
        type inet:ipv4-address;
    }
    description
    "Multicast source IP address type for SSM map.";
} // ssm-map-ipv4-addr-type

typedef ssm-map-ipv6-addr-type {
    type union {
        type enumeration {
            enum 'policy' {
                description
                "Source address is specified in SSM map policy.";
            }
        }
        type inet:ipv6-address;
    }
    description
    "Multicast source IP address type for SSM map.";
} // ssm-map-ipv6-addr-type

```

```

typedef source-ipv4-addr-type {
    type union {
        type enumeration {
            enum '*' {
                description
                "Any source address.";
            }
        }
        type inet:ipv4-address;
    }
    description

```

```

    "Multicast source IP address type.";
} // source-ipv4-addr-type

typedef source-ipv6-addr-type {
    type union {
        type enumeration {
            enum '*' {
                description
                "Any source address.";
            }
        }
        type inet:ipv6-address;
    }
    description
    "Multicast source IP address type.";
} // source-ipv6-addr-type

typedef channel-mode {
    type enumeration {
        enum "asm" {
            value 0;
            description
            "Channel-Name-Mode ASM.";
        }
        enum "ssm" {
            value 1;
            description
            "Channel-Name-Mode SSM.";
        }
    }
    description
    "Channel Mode";
}

typedef flagType {
    type enumeration {
        enum "STATIC" {
            value 0;

```

```

        description
        "Static.";
    }
    enum "DYNAMIC" {
        value 1;

```



```

        description
            "Dynamic.";
    }
    enum "MAPPING" {
        value 2;
        description
            "SSM Mapping.";
    }
    enum "STATIC_DYNAMIC" {
        value 3;
        description
            "Both static and dynamic.";
    }
    enum "STATIC_MAPPING" {
        value 4;
        description
            "Both statically and SSM mapping joined groups exist.";
    }
    enum "DYNAMIC_MAPPING" {
        value 5;
        description
            "Both dynamically and SSM mapping joined groups exist.";
    }
    enum "STATIC_DYNAMIC_MAPPING" {
        value 6;
        description
            "Both statically dynamically and SSM mapping joined groups exist.
    }
}
description
    "Flag Type";
}

typedef snooping-mode {
    type enumeration {
        enum "asm-ssm" {
            value 0;
            description
                "A device learns multicast entries with group
                addresses in the ASM and SSM address ranges.";
        }
        enum "asm-only" {
            value 1;
            description
                "A device only learns multicast entries with group

```

```
        addresses in the ASM address range.";
    }
    enum "ssm-only" {
        value 2;
        description
            "A device only learns multicast entries with group
            addresses in the SSM address range.";
    }
}
description
    "Snooping mode";
} //snooping-mode

typedef VLAN_BITMAP {
    type string {
        length "2049";
    }
    description
        "vlan bitmap";
}

typedef exitOrNot {
    type enumeration {
        enum "Not-exist" {
            value 0;
            description
                "Not exist.";
        }
        enum "Exist" {
            value 1;
            description
                "Exist.";
        }
    }
    description
        "Exist or Not-exist";
}

typedef filter-mode {
    type enumeration {
        enum "include" {
            description
                "";
        }
        enum "exclude" {
            description
                "";
        }
    }
}
```

```
}  
description
```

```
    "group filter mode";  
}  
  
typedef compatibility-mode {  
    type enumeration {  
        enum "V1" {  
            value 0;  
            description  
                "L2VPN service multicast version 1.";  
        }  
        enum "V2" {  
            value 1;  
            description  
                "L2VPN service multicast version 2.";  
        }  
        enum "V3" {  
            value 2;  
            description  
                "L2VPN service multicast version 3.";  
        }  
    }  
}  
description  
    "Group Compatible Mode";  
}  
  
/*  
 * Identities  
 */  
  
/*  
 * Groupings  
 */  
  
grouping group-states-igmp-mld-snooping {  
    description  
        "Per port state attributes for both IGMP and MLD snooping  
        groups.";  
  
    leaf filter-mode {  
        type filter-mode;  
        description "Group filter mode.";  
    }  
}
```

```

}
leaf up-time {
    type uint32;
    units seconds;
    description
        "Time since a multicast group was discovered, in seconds.";
}
leaf group-timer {
    type exitOrNot;

```

```

    description
        "Group timer.";
}
leaf expire {
    type uint32;
    units seconds;
    description
        "Time after which a group will be deleted from the
        group record table, in seconds.";
}
leaf retransmission-count {
    type uint32;
    description
        "Number of times that the last-member
        query message is transmitted.";
}
leaf last-member-query {
    type boolean;
    default "false";
    description
        "Whether group-specific Query messages are being sent.";
}
leaf compatibility {
    type compatibility-mode;
    description
        "Group compatible mode.";
}
leaf v1-host-timer {
    type exitOrNot;
    description
        "Timeout period of a L2VPN service multicastv1 host.";
}
leaf v2-host-timer {
    type exitOrNot;

```

```

        description
            "Timeout period of a L2VPN service multicastv2 host.";
    }
    leaf last-source-member-query-timer {
        type exitOrNot;
        description
            "Expire time of a last-member query message.";
    }
    leaf last-source-member-query {
        type boolean;
        default "false";
        description
            "Whether source-specific Query messages are being sent.";
    }
} // group-states-igmp-mld-snooping

```

```

grouping source-states-igmp-mld-snooping {
    description
        "Per port state attributes for both IGMP and MLD snooping
        sources.";

    leaf expire {
        type uint32;
        units seconds;
        description
            "Time after which a source will be deleted
            from the source record table, in seconds.";
    }

    leaf up-time {
        type uint32;
        units seconds;
        description "Time since a multicast source was discovered.";
    }

    leaf source-timer {
        type exitOrNot;
        description
            "source timer.";
    }

    leaf retransmission-count {
        type uint32;
        description

```

```

        "Number of times that the last-member
        query message is transmitted.";
    }
    leaf last-source-member-query {
        type boolean;
        default "false";
        description
            "Whether source-specific Query messages are being sent.";
    }
} // source-states-igmp-mld-snooping

grouping ipv4-static-group-source {
    description "static group source address.";
    container static-group-sources {
        description
            "List of static groups. ";
        list static-group-source {
            key "group-address source-address";
            description
                "Static group.";
            leaf group-address {
                type inet:ipv4-address;
                description

```

```

        "Address of a multicast group.";
    }
    leaf source-address {
        type inet:ipv4-address;
        description
            "Multicast source address.";
    }
}
} //ipv4-static-group-source

grouping ipv6-static-group-source {
    description "static group source address.";
    container static-group-sources {
        description
            "List of static groups. ";
        list static-group-source {
            key "group-address source-address";
            description
                "Static group.";

```

```

        leaf group-address {
            type inet:ipv6-address;
            description
                "Address of a multicast group.";
        }
        leaf source-address {
            type inet:ipv6-address;
            description
                "Multicast source address.";
        }
    }
} //ipv6-static-group-source

grouping ipv4-ssm-mapping-attributes {
    description "ssm mapping attribute.";
    leaf group-address {
        type inet:ipv4-address;
        description
            "Address of a multicast group.";
    }
    leaf group-mask-length {
        type uint8 {
            range "4..32";
        }
        description
            "IP address mask length.";
    }
    leaf source-address {
        type inet:ipv4-address;
    }
}

```

```

        description
            "Multicast source address.";
    }
} //ipv4-ssm-mapping-attributes

grouping ipv6-ssm-mapping-attributes {
    description "ssm mapping attribute.";
    leaf group-address {
        type inet:ipv6-address;
        description
            "Address of a multicast group.";
    }
    leaf group-mask-length {

```

```

        type uint8 {
            range "8..128";
        }
        description
            "IPv6 address mask length.";
    }
    leaf source-address {
        type inet:ipv6-address;
        description
            "Multicast source address.";
    }
} //ipv6-ssm-mapping-attributes

grouping cac-ipv4-policy-attributes {
    description "cac policy config attributes.";

    leaf channel-name {
        type string {
            length "1..31";
        }
        description
            "Name of the channel.";
    }
    leaf channel-mode {
        type channel-mode;
        description
            "Channel type.";
    }
    leaf max-entry {
        type uint32;
        description
            "Maximum number of entries that can be generated.";
    }
    container channel-policys {
        description
            "L2-multicast CAC Global Channel Policy Config List. ";
        list channel-policy {

```

```

        key
            "group-address group-mask-length
            source-address source-mask-length";
        max-elements "16";
        description
            "L2-multicast CAC Global Channel Policy Config. ";

```



```

    leaf group-address {
        type inet:ipv4-address;
        description
            "Address of a multicast group.";
    }
    leaf group-mask-length {
        type uint8 {
            range "4..32";
        }
        description
            "IP address mask length.";
    }
    leaf source-address {
        type inet:ipv4-address;
        description
            "Multicast source address.";
    }
    leaf source-mask-length {
        type uint8 {
            range "0..32";
        }
        description
            "Specify the source mask length.";
    }
    leaf bandwidth-each-entry {
        type uint32 {
            range "0..131072";
        }
        default "0";
        description
            "Specify the bandwidth of each group (in Kbits/sec).";
    }
} //channel-policy
} //channel-policys
} //cac-ipv4-policy-attributes

grouping cac-ipv6-policy-attributes {
    description "cac policy config attributes.";

    leaf channel-name {
        type string {
            length "1..31";
        }
        description

```

```

        "Name of the channel.";
    }
    leaf channel-mode {
        type channel-mode;
        description
            "Channel type.";
    }
    container channel-policys {
        description
            "L2-multicast CAC Global Channel Policy Config List. ";
        list channel-policy {
            key
                "group-address group-mask-length
                source-address source-mask-length";
            max-elements "16";
            description
                "L2-multicast CAC Global Channel Policy Config. ";
            leaf group-address {
                type inet:ipv6-address;
                description
                    "Address of a multicast group.";
            }
            leaf group-mask-length {
                type uint8 {
                    range "8..128";
                }
                description
                    "IP address mask length.";
            }
            leaf source-address {
                type inet:ipv6-address;
                description
                    "Multicast source address.";
            }
            leaf source-mask-length {
                type uint8 {
                    range "0..128";
                }
                description
                    "Specify the source mask length.";
            }
            leaf bandwidth-each-entry {
                type uint32 {
                    range "0..131072";
                }
                default "0";
                description
                    "Specify the bandwidth of each group (in Kbits/sec).";
            }
        }
    } //channel-policy

```

Internet-Draft

VPLS Multicast Yang Model

March 2017

```
    } //channel-policys
  } //cac-ipv6-policy-attributes

  grouping cac-dynamic-attributes {
    description "Global IGMP or MLD snooping state.";
    leaf dynamic-entry {
      type uint32;
      description
        "Maximum number of entries generAted";
    }
    leaf dynamic-bandwidth {
      type uint32;
      description
        "Maximum bandwidth of entries generated";
    }
  }

  list channel-cac {
    key "channel-name channel-mode";
    max-elements "1024";
    description
      "L2-multicast CAC Global Channel entry count. ";

    leaf channel-name {
      type string {
        length "1..31";
      }
      description
        "Name of the channel.";
    }
    leaf channel-mode {
      type channel-mode;
      description
        "Channel type.";
    }
    leaf dynamic-entry-count {
      type uint32;
      description
        "Maximum number of entries generated.";
    }
  }

  leaf dynamic-bandwidth {
    type uint32;
    description
```

```

        "Maximum bandwidth of entries generated";
    }
} //global-cac-count

} // cac-dynamic-attributes

grouping global-ipv4-config-attributes {

```

```

description "Global IGMP or MLD snooping configuration.";

leaf enable {
    if-feature global-admin-enable;
    type boolean;
    description
        "true to enable IGMP or MLD snooping;
        false to disable IGMP or MLD snooping.";
}

leaf max-entry {
    type uint32;
    description
        "Maximum number of entries that can be generated";
}

leaf bandwidth {
    type uint32;
    description
        "Maximum bandwidth of entries that can be generated";
}

container ssm-mapping-policys {
    description
        "L2-multicast SSM mapping policy List. ";
    list ssm-mapping-policy {
        key "name";
        description "ssm mapping policy. ";
        leaf name {
            type string {
                length "1..31";
            }
            description "name of ssm mapping policy. ";
        }

        list ssm-map {

```

```

        key "group-address group-mask-length source-address";
        description "";
        uses ipv4-ssm-mapping-attributes;
    } // list ssm-map
} //ssm-mapping-policy
} //ssm-mapping-policys

leaf unspecified-channel-deny {
    type boolean;
    description
        "Specify the unspecified channel,deny the join of
        Multicast group that does not belong to any
        channel.By default, no deny channel is configured.";
}

```

```

container global-cac-configs {
    description
        "L2-multicast CAC Global Channel Config List. ";
    list global-cac-config {
        key "channel-name channel-mode";
        max-elements "1024";
        description
            "L2-multicast CAC Global Channel Config. ";
        uses cac-ipv4-policy-attributes;
    } //global-cac-config
}

} // global-ipv4-config-attributes

grouping global-ipv6-config-attributes {
    description "Global IGMP or MLD snooping configuration.";

    leaf enable {
        if-feature global-admin-enable;
        type boolean;
        description
            "true to enable IGMP or MLD snooping;
            false to disable IGMP or MLD snooping.";
    }

    leaf max-entry {
        type uint32;
        description

```

```

        "Maximum number of entries that can be generated";
    }

    leaf bandwidth {
        type uint32;
        description
            "Maximum bandwidth of entries that can be generated";
    }

    container ssm-mapping-policys {
        description
            "L2-multicast SSM mapping policy List. ";
        list ssm-mapping-policy {
            key "name";
            description "ssm mapping policy";
            leaf name {
                type string {
                    length "1..31";
                }
                description "name of ssm mapping policy. ";
            }
        }
    }

```

```

        list ssm-map {
            key "group-address group-mask-length source-address";
            description "";
            uses ipv6-ssm-mapping-attributes;
        } // list ssm-map
    } //ssm-mapping-policy
} //ssm-mapping-policys

leaf unspecified-channel-deny {
    type boolean;
    description
        "Specify the unspecified channel,deny the join of
        Multicast group that does not belong to any
        channel.By default, no deny channel is configured.";
}

container global-cac-configs {
    description
        "L2-multicast CAC Global Channel Config List. ";
    list global-cac-config {
        key "channel-name channel-mode";
    }
}

```

```

        max-elements "1024";
        description
            "L2-multicast CAC Global Channel Config. ";
        uses cac-ipv6-policy-attributes;
    } //global-cac-config
}

} // global-ipv6-config-attributes

grouping proxy-config-attributes {
    description "";
    leaf proxy-enable {
        type boolean;
        default "false";
        description
            "Enable L2VPN service multicast snooping proxy.";
    }

    leaf proxy-router-protocol-pass {
        type boolean;
        default "false";
        description
            "L2VPN service multicast snooping proxy-enabled
            device to transparently transmit
            multicast Report messages.";
    }
    leaf query-enable {
        type boolean;
    }
}

```

```

        default "false";
        description
            "Enable L2VPN service multicast querier.";
    }

    leaf querier-election {
        type boolean;
        description
            "Enable L2VPN service multicast querier election.";
    }

    leaf report-suppress {
        type boolean;
        default "false";
        description

```

```

        "Enable the function of suppressing L2VPN service
        multicast Report messages.";
    }

} //proxy-config-attributes

grouping l2vpn-config-attributes-igmp-mld-snooping {
    description "igmp snooping configuration attributes";

    leaf enable {
        type boolean;
        default "false";
        description
            "true to enable IGMP or MLD snooping in the L2VPN service
            instance, false to disable IGMP or MLD snooping
            in the L2VPN service instance";
    }
    leaf version {
        type uint8 {
            range "1..3";
        }
        description
            "L2VPN service multicast version. IPv4: By default,
            IGMPv2 is used; IPv6: By default, MLDv2 is used.";
    }
    leaf snooping-mode {
        type snooping-mode;
        description
            "L2VPN service multicast group join mode.";
    }
    leaf ssm-policy {
        type string;
        description
            "Name of the access policy used to filter
            IGMP or MLD SSM group range.";
    }

```

```

    }
    leaf group-policy {
        type string;
        description
            "Name of the access policy used to filter
            IGMP or MLD membership.";
    }
    leaf group-policy-version {

```



```

    type uint8 {
        range "1..3";
    }
    description
        "A group policy applies only to a specified
        L2VPN service multicast version.";
}
leaf ip-policy {
    type string;
    description
        "Name of the access policy used to filter IGMP or MLD ;
        report source/destination address";
}
leaf query-ip-policy {
    type string;
    description
        "Name of the access policy used to filter IGMP or MLD ;
        query source address";
}
leaf last-member-query-interval {
    type uint8 {
        range "1..5";
    }
    description
        "Interval at which a querier sends
        last-member query messages.";
}
leaf query-interval {
    type uint16 {
        range "1..65535";
    }
    units seconds;
    description
        "Interval at which a router sends General Query messages.";
}
leaf query-max-response-time {
    type uint8 {
        range "1..25";
    }
    units seconds;
    description
        "Maximum response time for a L2VPN service

```

```

}
leaf router-aging-time {
    type uint16{
        range "1..1000";
    }
    units seconds;
    description
        "Aging time of dynamic router ports.";
}
leaf prompt-leave {
    type boolean;
    description
        "Prompt-Leave allows an interface to immediately delete a
        group record after receiving a Leave message for
        this group, without sending a last-member query message.";
}
leaf router-learning {
    type boolean;
    default "true";
    description
        "Enable dynamic router port learning.";
}
leaf require-router-alert {
    type boolean;
    description
        "Configure a device to permit only L2VPN service
        multicast messages that contain the Router-Alert
        option in the IP header.";
}
leaf send-router-alert {
    type boolean;
    description
        "Configure a device to send L2VPN service
        multicast messages that contain the Router-Alert
        option in the IP header.";
}
leaf ssm-mapping {
    type boolean;
    default "false";
    description
        "Enable SSM mapping.";
}
leaf ssm-mapping-policy {
    type string;
    description
        "Specifies the name of an SSM mapping policy.";
}
leaf robustness-variable {
    type uint8 {

```

Internet-Draft

VPLS Multicast Yang Model

March 2017

```
        range "2..7";
    }
    default 2;
    description
        "Number of times for retransmitting a
        message to avoid packet loss. ";
}
leaf source-life-time {
    type uint32{
        range "60..1000";
    }
    units seconds;
    description
        "Aging time for entries triggered by multicast flows.";
}

leaf unspecified-channel-deny {
    type boolean;
    description
        "Specify the unspecified channel,deny the join of
        Multicast group that does not belong to any channel.";
}

leaf max-entry {
    type uint32;
    description
        "Maximum number of entries that can be generated";
}
leaf max-entry-except-policy {
    type string;
    description
        "Name of the access policy used to prevent groups
        from being counted against the multicast
        group number limit";
}

leaf bandwidth-lose-mode {
    type boolean;
    default "false";
    description
        "Dynamic Eth-Trunk Bandwidth Limit Lose Mode.";
}

uses proxy-config-attributes;
```

```
} // l2vpn-config-attributes-igmp-mld-snooping
```

```
grouping vlan-set {  
    description "VLAN IDs attributes";  
    leaf vlan-bitmap {
```

```
        type string;  
        description  
            "List of VLAN IDs.";  
    }  
} //vlan-set  
  
grouping l2vpn-if-config-attributes-igmp-mld-snooping {  
    description "igmp or mld snooping configuration attributes";  
  
    leaf bandwidth-lose-mode {  
        type boolean;  
        default "false";  
        description  
            "Dynamic Eth-Trunk Bandwidth Limit Lose Mode.";  
    }  
    leaf max-entry {  
        type uint32;  
        description  
            "Maximum number of entries that can be generated";  
    }  
  
    leaf group-policy {  
        type string;  
        description  
            "Name of the access policy used to filter  
            IGMP or MLD membership.";  
    }  
    leaf group-policy-version {  
        type uint8 {  
            range "1..3";  
        }  
        description  
            "A group policy applies only to a specified  
            L2VPN service multicast version.";  
    }  
}
```

```

choice group-policy-vlan {
  description "group policy vlan config";
  case qinq-range {
    list qinq-vlan {
      key "pe-id";
      description "qinq type interface config";
      leaf pe-id {
        type uint16;
        description
          "Outer VLAN ID.";
      }
      uses vlan-set;
    } // qinq-vlan
  }
}

```

```

    } //qinq-range
  case dot1q-range {
    description "dot1q type interface config";
    uses vlan-set;
  } //dot1q-range
} //group-policy-vlan

list trunk-bandwidth {
  key "bandwidth";
  description
    "Dynamic Bandwidth Limit on
     Eth-Trunk Sub-interface.";
  leaf bandwidth {
    type uint32 {
      range "1..4194304";
    }
    description
      "Specify the bandwidth parameter of multicast group.";
  }
  uses vlan-set;
}

list dot1q-bandwidth {
  key "bandwidth";
  description "Bandwidth Limit on Dot1q Sub-interface. ";
  leaf bandwidth {
    type uint32 {
      range "1..4194304";
    }
    description

```

```

        "Specify the bandwidth parameter of multicast group.";
    }
    uses vlan-set;
}

list dot1q-channel-bandwidth {
    key "channel-name bandwidth";
    description "Bandwidth Limit on Dot1q Sub-interface. ";
    leaf channel-name {
        type string {
            length "1..31";
        }
        description
            "Name of the channel.";
    }
    leaf bandwidth {
        type uint32 {
            range "1..4194304";
        }
        description

```

```

        "Specify the bandwidth parameter of multicast group.";
    }
    uses vlan-set;
}

} // l2vpn-if-config-attributes-igmp-mld-snooping

grouping l2vpn-if-dynamic-cac-state-igmp-mld-snooping {
    description "igmp or mld snooping configuration attributes";

    leaf dynamic-entry {
        type uint32;
        description
            "Maximum number of entries generated";
    }

    list dynamic-trunk-bandwidth {
        key "vlan-id";
        description
            "Dynamic Bandwidth used on Eth-Trunk Sub-interface.";
        leaf vlan-id {
            type uint16 {
                range "1..4095";
            }

```

```

    }
    description
        "vlan-id.";
}
leaf bandwidth {
    type uint32 {
        range "1..4194304";
    }
    description
        "Bandwidth used.";
}
}

list dynamic-dot1q-bandwidth {
    key "vlan-id";
    description "Bandwidth used on Dot1q Sub-interface. ";
    leaf vlan-id {
        type uint16 {
            range "1..4095";
        }
        description
            "vlan-id.";
    }
    leaf bandwidth {
        type uint32 {
            range "1..4194304";
        }
    }
}

```

```

    description
        "Bandwidth used.";
}
}

list dynamic-dot1q-channel-bandwidth {
    key "channel-name";
    description "Bandwidth Limit on Dot1q Sub-interface. ";
    leaf channel-name {
        type string {
            length "1..31";
        }
        description
            "Name of the channel.";
    }
}

```

```

list vlan-bandwidth {
  key "vlan-id";
  description "Bandwidth used on Dot1q Sub-interface. ";
  leaf vlan-id {
    type uint16 {
      range "1..4095";
    }
    description
      "vlan-id.";
  }
  leaf bandwidth {
    type uint32 {
      range "1..4194304";
    }
    description
      "Bandwidth used.";
  }
} //vlan-bandwidth

} //dot1q-channel-bandwidth

} // l2vpn-if-dynamic-cac-state-igmp-mld-snooping

grouping statistics-received {
  description
    "A grouping defining statistics attributes.";
  leaf v1-report {
    type yang:counter64;
    description
      "Number of received IGMPv1/MLDv1 Report messages.";
  }
  leaf v2-report {
    type yang:counter64;
    description

```

```

    "Number of received IGMPv2/MLDv2 Report messages.";
  }
  leaf v3-report {
    type yang:counter64;
    description
      "Number of received IGMPv3 Report messages.";
  }
  leaf v1-query {
    type yang:counter64;

```



```

        description
            "Number of received IGMPv1/MLDv1 query messages.";
    }
    leaf v2-query {
        type yang:counter64;
        description
            "Number of received IGMPv2/MLDv2 query messages.";
    }
    leaf v3-query {
        type yang:counter64;
        description
            "Number of received IGMPv3 Query messages.";
    }
    leaf leave {
        type yang:counter64;
        description
            "Number of received IGMP/MLD Leave messages.";
    }
    leaf hello {
        type yang:counter64;
        description
            "Number of Received PIM Hello messages.";
    }
    leaf zero-source-query {
        type yang:counter64;
        description
            "Number of received query messages
            in which the source addresss is zero.";
    }
    leaf no-zero-source-query {
        type yang:counter64;
        description
            "Number of received query messages
            in which the source addresss is not zero.";
    }
} // statistics-received

```

```

grouping statistics-send {
    description
        "A grouping defining statistics attributes.";
}

```

```

    leaf general-query {
        type yang:counter64;
    }

```

```

        description
            "Number of sent General Query messages.";
    }
    leaf source-group-specific-query {
        type yang:counter64;
        description
            "Number of sent source and group specific Query messages.";
    }
    leaf group--specific-query {
        type yang:counter64;
        description
            "Number of sent group specific Query messages.";
    }
} // statistics-send

/*
 * Configuration data nodes
 */
augment "/l2vpn:l2vpn"
{
    description
        "IGMP and MLD snooping augmentation to
        l2vpn instance configuration.";

    container igmp-snooping {
        description
            "IGMP snooping configuration data.";

        container global {
            description
                "Global attributes.";
            uses global-ipv4-config-attributes;

            leaf send-query-source {
                type inet:ipv4-address;
                default "192.168.0.1";
                description
                    "Source IP address for L2VPN service
                    multicast query messages.";
            }
        } // global

        container l2vpn-multicast-instances {
            description
                "List of L2VPN service multicast
                configurations in the VSI view. ";
            list l2vpn-multicast-instance {
                key "name type";
            }
        }
    }
}

```

```
description
  "L2VPN common parameters.";
leaf name {
  type string;
  description "Name of L2VPN service instance";
}
leaf type {
  type identityref {
    base l2vpn:l2vpn-instance-type;
  }
  description "Type of L2VPN service instance";
}

uses l2vpn-config-attributes-igmp-mld-snooping;

list ssm-map {
  key "group-address group-mask-length source-address";
  description "";
  uses ipv4-ssm-mapping-attributes;
} // list ssm-map

list static-pw-router {
  key "peer-ip pw-id";
  description "";
  leaf peer-ip {
    type inet:ip-address;
    description
      "peer IP address";
  }
  leaf pw-id {
    type uint32;
    description
      "pseudowire id";
  }
} // static-pw-router

list static-ac-router-port {
  key "name pe-id ce-id";
  description "";
  leaf name {
    type if:interface-ref;
    description
      "interface name";
  }
  leaf pe-id {
    type uint16;
    description
```

```
        "Outer VLAN ID.";
    }
    leaf ce-id {
```

```
        type uint16;
        description
            "Inner VLAN ID.";
    }
} //static-ac-router-port

list static-pw-group {
    key "peer-ip pw-id";
    description "";
    leaf peer-ip {
        type inet:ip-address;
        description
            "peer IP address";
    }
    leaf pw-id {
        type uint32;
        description
            "pseudowire id";
    }
    uses ipv4-static-group-source;
} // static-pw-group

list static-ac-static-group {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
    leaf ce-id {
        type uint16;
        description
            "Inner VLAN ID.";
    }
}
```

```

    uses ipv4-static-group-source;
} //static-ac-static-group

container cac-configs {
    description
        "L2-multicast CAC L2VPN service Channel Config List.";
    list cac-config {
        key "channel-name channel-mode";
        max-elements "1024";
        description

```

Wang & Liu, etc

Expires September 13, 2017

[Page 46]

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

        "L2-multicast CAC L2VPN service Channel Config.";
        uses cac-ipv4-policy-attributes;
    }
} //cac-configs

container interfaces {
    description
        "Containing a list of interfaces.";

    list interface {
        key "interface";
        description
            "List of igmp snooping interfaces.";
        leaf interface {
            type if:interface-ref;
            description
                "Reference to an entry in the
                global interface list.";
        }
        uses l2vpn-if-config-attributes-igmp-mld-snooping;
    } // interface
} // interfaces
} //l2vpn-multicast-instance
} //l2vpn-multicast-instances
} // igmp-snooping

container mld-snooping {
    description
        "MLD snooping configuration data.";

    container global {
        description
            "Global attributes.";
    }
}

```

```

uses global-ipv6-config-attributes;

leaf send-query-source {
  type inet:ipv6-address;
  default "FE80::";
  description
    "Source IPv6 address for L2VPN service
    multicast query messages.";
}
} // global

container l2vpn-multicast-instances {
  description
    "List of L2VPN service multicast configurations
    in the VSI view. ";
  list l2vpn-multicast-instance {
    key "name type";

```

```

  description
    "L2VPN common parameters.";
  leaf name {
    type string;
    description "Name of L2VPN service instance";
  }
  leaf type {
    type identityref {
      base l2vpn:l2vpn-instance-type;
    }
    description "Type of L2VPN service instance";
  }
}

uses l2vpn-config-attributes-igmp-mld-snooping;

list ssm-map {
  key "group-address group-mask-length source-address";
  description "";
  uses ipv6-ssm-mapping-attributes;
} // list ssm-map

list static-pw-router {
  key "peer-ip pw-id";
  description "";
  leaf peer-ip {
    type inet:ip-address;

```

```

        description
            "peer IP address";
    }
    leaf pw-id {
        type uint32;
        description
            "pseudowire id";
    }
} // static-pw-router

list static-ac-router-port {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
    leaf ce-id {

```

```

        type uint16;
        description
            "Inner VLAN ID.";
    }
} //static-ac-router-port

list static-pw-group {
    key "peer-ip pw-id";
    description "";
    leaf peer-ip {
        type inet:ip-address;
        description
            "peer IP address";
    }
    leaf pw-id {
        type uint32;
        description
            "pseudowire id";
    }
}

```

```

    uses ipv6-static-group-source;
} // static-pw-group

list static-ac-static-group {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
    leaf ce-id {
        type uint16;
        description
            "Inner VLAN ID.";
    }
    uses ipv6-static-group-source;
} //static-ac-static-group

container cac-configs {
    description
        "L2-multicast CAC L2VPN service Channel Config List. ";
    list cac-config {
        key "channel-name channel-mode";
        max-elements "1024";
        description

```

```

        "L2-multicast CAC L2VPN service Channel Config. ";
        uses cac-ipv6-policy-attributes;
    }
} //cac-configs

container interfaces {
    description
        "Containing a list of interfaces.";

    list interface {
        key "interface";
        description

```



```

        "List of igmp snooping interfaces.";
    leaf interface {
        type if:interface-ref;
        description
            "Reference to an entry in the global
            interface list.";
    }
    uses l2vpn-if-config-attributes-igmp-mld-snooping;
} // interface
} // interfaces
} //l2vpn-multicast-instance
} //l2vpn-multicast-instances
} // mld-snooping

} //l2vpn

/*
 * Operational state data nodes
 */
augment "/l2vpn:l2vpn-state"
{
    description
        "IGMP and MLD snooping augmentation to
        l2vpn instance configuration.";

    container igmp-snooping {
        description
            "IGMP snooping state data.";

        container global {
            description
                "Global attributes.";
            uses global-ipv4-config-attributes;

            uses cac-dynamic-attributes;

            leaf send-query-source {
                type inet:ipv4-address;
            }
        }
    }
}

```

```

    default "192.168.0.1";
    description
        "Source IP address for L2VPN service
        multicast query messages.";
}

```

```

} // global

container l2vpn-multicast-instances {
  description
    "List of L2VPN service multicast
    configurations in the VSI view. ";
  list l2vpn-multicast-instance {
    key "name type";
    description
      "L2VPN common parameters.";
    leaf name {
      type string;
      description "Name of L2VPN service instance";
    }
    leaf type {
      type identityref {
        base l2vpn:l2vpn-instance-type;
      }
      description "Type of L2VPN service instance";
    }
  }

  uses l2vpn-config-attributes-igmp-mld-snooping;

  container querier-state {
    description
      "L2VPN service multicast querier. ";
    leaf querierState {
      type boolean;
      description
        "L2VPN service multicast querier state.";
    }
    leaf querierAddr {
      type inet:ip-address;
      description
        "L2VPN service multicast querier address.";
    }
  }
}

container statistics {
  description
    "Statistics information of a L2VPN service instance. ";

  container received {
    description "Statistics of received messages.";
    uses statistics-received;
  }
}

```

```
    }
    container sent {
        description "Statistics of sent messages.";
        uses statistics-send;
    }

} //statistics

list ssm-map {
    key "group-address group-mask-length source-address";
    description "";
    uses ipv4-ssm-mapping-attributes;
} // list ssm-map

list static-pw-router-port {
    key "peer-ip pw-id";
    description "";
    leaf peer-ip {
        type inet:ip-address;
        description
            "peer IP address";
    }
    leaf pw-id {
        type uint32;
        description
            "pseudowire id";
    }
} // static-pw-router-port

list static-ac-router-port {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
    leaf ce-id {
        type uint16;
        description
            "Inner VLAN ID.";
    }
} //static-ac-router-port
```

```
list static-pw-group {
```

```
    key "peer-ip pw-id";
    description "";
    leaf peer-ip {
        type inet:ip-address;
        description
            "peer IP address";
    }
    leaf pw-id {
        type uint32;
        description
            "pseudowire id";
    }
    uses ipv4-static-group-source;
} // static-pw-group

list static-ac-group {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
    leaf ce-id {
        type uint16;
        description
            "Inner VLAN ID.";
    }
    uses ipv4-static-group-source;
} //static-ac-group

container cac-configs {
    description
        "L2-multicast CAC L2VPN service Channel Config List. ";
    list cac-config {
        key "channel-name channel-mode";
```

```

        max-elements "1024";
        description
            "L2-multicast CAC L2VPN service Channel Config. ";
        uses cac-ipv4-policy-attributes;
    }
} //cac-configs

container interfaces {
    description

```

Wang & Liu, etc

Expires September 13, 2017

[Page 53]

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

    "Containing a list of interfaces.";

    list interface {
        key "interface";
        description
            "List of igmp snooping interfaces.";
        leaf interface {
            type if:interface-ref;
            description
                "Reference to an entry in the
                 global interface list.";
        }
        uses l2vpn-if-config-attributes-igmp-mld-snooping;

        uses l2vpn-if-dynamic-cac-state-igmp-mld-snooping;

    } // interface
} // interfaces

choice dynamic-router-port {
    description "dynamic router port";
    case pw {

        list dynamic-pw-router-port {
            key "peer-ip pw-id";
            description "";
            leaf peer-ip {
                type inet:ip-address;
                description
                    "peer IP address";
            }
            leaf pw-id {
                type uint32;
                description

```

```

        "pseudowire id";
    }
    leaf expire {
        type uint32;
        units seconds;
        description
            "Time after which a port will be deleted
            from the router port record table.";
    }
    leaf up-time {
        type uint32;
        units seconds;
        description
            "Time since a router port
            was discovered.";
    }
}

```

Wang & Liu, etc

Expires September 13, 2017

[Page 54]

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

    leaf flag {
        type flagType;
        description
            "Type of a router port: static or dynamic.";
    }
} //dynamic-pw-router-port
} //pw

case ac {
    list dynamic-ac-router-port {
        key "name pe-id ce-id";
        description "";
        leaf name {
            type if:interface-ref;
            description
                "interface name";
        }
        leaf pe-id {
            type uint16;
            description
                "Outer VLAN ID.";
        }
        leaf ce-id {
            type uint16;
            description
                "Inner VLAN ID.";
        }
    }
}

```

```

leaf expire {
    type uint32;
    units seconds;
    description
        "Time after which a port will be deleted
        from the router port record table.";
}
leaf up-time {
    type uint32;
    units seconds;
    description
        "Time since a router port was discovered.";
}
leaf flag {
    type flagType;
    description
        "Type of a router port: static or dynamic.";
}
} //dynamic-ac-router-port
} //ac
} //dynamic-router-port

choice dynamic-group {

```

```

description "dynamic group";
case pw-dynamic-group {

    list dynamic-pw-group {
        key "peer-ip pw-id";
        description "";
        leaf peer-ip {
            type inet:ip-address;
            description
                "peer IP address";
        }
        leaf pw-id {
            type uint32;
            description
                "pseudowire id";
        }
    }
    container groups {
        description "multicast group information";
        list group {
            key "address";

```

```

        description "multicast group information";
        leaf address {
            type inet:ipv4-address;
            description
                "address of a multicast group.";
        }

        uses group-states-igmp-mld-snooping;

        container sources {
            description "multicast group information";
            list source {
                key "address";
                description "multicast group information";
                leaf address {
                    type inet:ipv4-address;
                    description
                        "address of a multicast source.";
                }
                uses source-states-igmp-mld-snooping;
            } //source
        } //sources

    } //group
} //groups

} // dynamic-pw-group
} //pw-dynamic-group

case ac-dynamic-group {

```

```

list dynamic-ac-dynamic-group {
    key "name pe-id ce-id";
    description "";
    leaf name {
        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description
            "Outer VLAN ID.";
    }
}

```



```

leaf ce-id {
    type uint16;
    description
        "Inner VLAN ID.";
}
container groups {
    description "multicast group information";
    list group {
        key "address";
        description "multicast group information";
        leaf address {
            type inet:ipv4-address;
            description
                "address of a multicast group.";
        }
        uses group-states-igmp-mld-snooping;

        container sources {
            description "multicast group information";
            list source {
                key "address";
                description "multicast group information";
                leaf address {
                    type inet:ipv4-address;
                    description
                        "address of a multicast source.";
                }
                uses source-states-igmp-mld-snooping;
            } //source
        } //sources

    } //group
} //groups

} //dynamic-ac-dynamic-group
} //ac-dynamic-group
} //dynamic-group

```

```

    } //l2vpn-multicast-instance
} //l2vpn-multicast-instances
} // igmp-snooping

container mld-snooping {

```

```

description
  "MLD snooping state data.";

container global {
  description
    "Global attributes.";
  uses global-ipv6-config-attributes;

  uses cac-dynamic-attributes;

  leaf entry-count {
    type uint32;
    description
      "Maximum number of entries generated";
  }

  leaf send-query-source {
    type inet:ipv6-address;
    default "FF80::";
    description
      "Source IPv6 address for L2VPN service
      multicast query messages.";
  }
} // global

container l2vpn-multicast-instances {
  description
    "List of L2VPN service multicast configurations
    in the VSI view. ";
  list l2vpn-multicast-instance {
    key "name type";
    description
      "L2VPN common parameters.";
    leaf name {
      type string;
      description "Name of L2VPN service instance";
    }
    leaf type {
      type identityref {
        base l2vpn:l2vpn-instance-type;
      }
      description "Type of L2VPN service instance";
    }
  }
}

```

```

uses l2vpn-config-attributes-igmp-mld-snooping;

container querier-state {
  description
    "L2VPN service multicast querier. ";
  leaf querierState {
    type boolean;
    description
      "L2VPN service multicast querier state.";
  }
  leaf querierAddr {
    type inet:ipv6-address;
    description
      "L2VPN service multicast querier address.";
  }
}

container statistics {
  description
    "Statistics information of a L2VPN service instance. ";

  container received {
    description "Statistics of received messages.";
    uses statistics-received;
  }
  container sent {
    description "Statistics of sent messages.";
    uses statistics-send;
  }
} //statistics

list ssm-map {
  key "group-address group-mask-length source-address";
  description "";
  uses ipv6-ssm-mapping-attributes;
} // list ssm-map

list static-pw-router-port {
  key "peer-ip pw-id";
  description "";
  leaf peer-ip {
    type inet:ip-address;
    description
      "peer IP address";
  }
  leaf pw-id {
    type uint32;
    description
      "pseudowire id";
  }
}

```

Internet-Draft

VPLS Multicast Yang Model

March 2017

```
    }
  } // static-pw-router-port

  list static-ac-router-port {
    key "name pe-id ce-id";
    description "";
    leaf name {
      type if:interface-ref;
      description
        "interface name";
    }
    leaf pe-id {
      type uint16;
      description
        "Outer VLAN ID.";
    }
    leaf ce-id {
      type uint16;
      description
        "Inner VLAN ID.";
    }
  } //static-ac-router-port

  list static-pw-group {
    key "peer-ip pw-id";
    description "";
    leaf peer-ip {
      type inet:ip-address;
      description
        "peer IP address";
    }
    leaf pw-id {
      type uint32;
      description
        "pseudowire id";
    }
    uses ipv6-static-group-source;
  } // static-pw-group

  list static-ac-group {
    key "name pe-id ce-id";
    description "";
    leaf name {
```

```

        type if:interface-ref;
        description
            "interface name";
    }
    leaf pe-id {
        type uint16;
        description

```

```

        "Outer VLAN ID.";
    }
    leaf ce-id {
        type uint16;
        description
            "Inner VLAN ID.";
    }
    uses ipv6-static-group-source;
} //static-ac-group

container cac-configs {
    description
        "L2-multicast CAC L2VPN service Channel Config List. ";
    list cac-config {
        key "channel-name channel-mode";
        max-elements "1024";
        description
            "L2-multicast CAC L2VPN service Channel Config. ";
        uses cac-ipv6-policy-attributes;
    }
} //cac-configs

uses cac-dynamic-attributes;

container interfaces {
    description
        "Containing a list of interfaces.";

    list interface {
        key "interface";
        description
            "List of igmp snooping interfaces.";
        leaf interface {
            type if:interface-ref;
            description
                "Reference to an entry in the

```

```

        global interface list.";
    }
    uses l2vpn-if-config-attributes-igmp-mld-snooping;

    uses l2vpn-if-dynamic-cac-state-igmp-mld-snooping;
} // interface
} // interfaces

choice dynamic-router-port {
    description "dynamic router port";
    case pw {

        list dynamic-pw-router-port {
            key "peer-ip pw-id";

```

```

description "";
leaf peer-ip {
    type inet:ip-address;
    description
        "peer IP address";
}
leaf pw-id {
    type uint32;
    description
        "pseudowire id";
}
leaf expire {
    type uint32;
    units seconds;
    description
        "Time after which a port will be deleted
        from the router port record table.";
}
leaf up-time {
    type uint32;
    units seconds;
    description
        "Time since a router port was discovered.";
}
leaf flag {
    type flagType;
    description
        "Type of a router port: static or dynamic.";
}

```

```

    } //dynamic-pw-router-port
} //pw

```

```

case ac {
    list dynamic-ac-router-port {
        key "name pe-id ce-id";
        description "";
        leaf name {
            type if:interface-ref;
            description
                "interface name";
        }
        leaf pe-id {
            type uint16;
            description
                "Outer VLAN ID.";
        }
        leaf ce-id {
            type uint16;
            description
                "Inner VLAN ID.";
        }
    }
}

```

Wang & Liu, etc

Expires September 13, 2017

[Page 62]

Internet-Draft

VPLS Multicast Yang Model

March 2017

```

    }
    leaf expire {
        type uint32;
        units seconds;
        description
            "Time after which a port will be deleted
             from the router port record table.";
    }
    leaf up-time {
        type uint32;
        units seconds;
        description
            "Time since a router port was discovered.";
    }
    leaf flag {
        type flagType;
        description
            "Type of a router port: static or dynamic.";
    }
} //dynamic-ac-router-port
} //ac
} //dynamic-router-port

```

```

choice dynamic-group {
    description "dynamic group";
    case pw-dynamic-group {

        list dynamic-pw-group {
            key "peer-ip pw-id";
            description "";
            leaf peer-ip {
                type inet:ip-address;
                description
                    "peer IP address";
            }
            leaf pw-id {
                type uint32;
                description
                    "pseudowire id";
            }
        }
        container groups {
            description "multicast group information";
            list group {
                key "address";
                description "multicast group information";
                leaf address {
                    type inet:ipv6-address;
                    description
                        "address of a multicast group.";
                }
            }
        }
    }
}

```

```

uses group-states-igmp-mld-snooping;

container sources {
    description "multicast group information";
    list source {
        key "address";
        description "multicast group information";
        leaf address {
            type inet:ipv6-address;
            description
                "address of a multicast source.";
        }
        uses source-states-igmp-mld-snooping;
    } //source
} //sources

```



```

        } //group
    } //groups

    } // dynamic-pw-group
} //pw-dynamic-group

case ac-dynamic-group {
    list dynamic-ac-dynamic-group {
        key "name pe-id ce-id";
        description "";
        leaf name {
            type if:interface-ref;
            description
                "interface name";
        }
        leaf pe-id {
            type uint16;
            description
                "Outer VLAN ID.";
        }
        leaf ce-id {
            type uint16;
            description
                "Inner VLAN ID.";
        }
        container groups {
            description "multicast group information";
            list group {
                key "address";
                description "multicast group information";
                leaf address {
                    type inet:ipv6-address;
                    description
                        "address of a multicast group.";
                }
            }
        }
    }
}

```

```

    }
    uses group-states-igmp-mld-snooping;

    container sources {
        description "multicast group information";
        list source {
            key "address";
            description "multicast group information";
        }
    }
}

```

```

        leaf address {
            type inet:ipv6-address;
            description
                "address of a multicast source.";
        }
        uses source-states-igmp-mld-snooping;
    } //source
} //sources

    } //group
} //groups

    } //dynamic-ac-dynamic-group
    } //ac-dynamic-group
} //dynamic-group

    } //l2vpn-multicast-instance
    } //l2vpn-multicast-instances
} // mld-snooping
} //l2vpn-state
}
<CODE ENDS>

```

[5. Security Considerations](#)

The data model defined does not introduce any security implications. This draft does not change any underlying security issues inherent in [[RFC8022](#)].

[6. IANA Considerations](#)

TBD

[7. References](#)

[7.1. Normative References](#)

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011

[RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for RoutingManagement", [RFC 8022](#), November 2016

[I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", [draft-ietf-netmod-rfc6087bis-12](#)(work in progress), March 2017

[7.2](#). Informative References

[RFC4541] Christensen, M., Kimball, K., and Solensky, F., "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", [RFC 4541](#), May 2006.

[RFC4761] K. Kompella and Y. Rekhter, " Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", [RFC 4761](#), January 2007.

[RFC4762] M. Lasserre and V. Kompella, " Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", [RFC 4762](#), January 2007.

[RFC7117] R. Aggarwal, Y. Kamite, L. Fang, Y. Rekhter and C. Kodeboniya, " Multicast in Virtual Private LAN Service (VPLS)", [RFC 7117](#), February 2014.

[RFC7524] Y. Rekhter, E. Rosen, R. Aggarwal, T. Morin, I. Grosclaude, N. Leymann and S. Saad, " Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", [RFC 7524](#), May 2015.

Authors' Addresses

Dong Wang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: michael.wangdong@huawei.com

Yisong Liu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: liuyisong@huawei.com

