Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 28, 2017

W. Cheng
L. Wang
H. Li
China Mobile
J. Dong
Huawei Technologies
A. D'Alessandro
Telecom Italia
April 26, 2017

Dual-Homing Coordination for MPLS Transport Profile (MPLS-TP) Pseudowires Protection
draft-ietf-pals-mpls-tp-dual-homing-coordination-06

Abstract

In some scenarios, MPLS Transport Profile (MPLS-TP) Pseudowires (PWs) (RFC 5921) may be statically configured, when a dynamic control plane is not available. A fast protection mechanism for MPLS-TP PWs is needed to protect against the failure of an Attachment Circuit (AC), the failure of a Provider Edge (PE), or a failure in the Packet Switched Network (PSN). The framework and typical scenarios of dual-homing PW local protection are described in [draft-ietf-pals-mpls-tp-dual-homing-protection]. This document proposes a dual-homing coordination mechanism for MPLS-TP PWs, which is used for state exchange and switchover coordination between the dual-homing PEs for dual-homing PW local protection.

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].

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on October 28, 2017.

Table of Contents

1. Introduction

[RFC6372], [RFC6378] and [RFC7771] describe the framework and mechanism of MPLS Transport Profile (MPLS-TP) linear protection, which can provide protection for the MPLS Label Switched Path (LSP) and Pseudowires (PWs) between the edge nodes. These mechanisms cannot protect the failure of the Attachment Circuit (AC) or the edge nodes. [RFC6718] and [RFC6870] specifies the PW redundancy framework and mechanism for protecting the AC or edge node failure by adding one or more edge nodes, but it requires PW switchover in case of an AC failure, also PW redundancy relies on Packet Switched Network (PSN) protection mechanisms to protect the failure of PW.

In some scenarios such as mobile backhauling, the MPLS PWs are provisioned with dual-homing topology, in which at least the CE node on one side is dual-homed to two Provider Edge (PE) nodes. If a failure occurs in the primary AC, operators usually prefer to perform local switchover in the dual-homing PE side and keep the working pseudowire unchanged if possible. This is to avoid massive PW switchover in the mobile backhaul network due to the AC failure in the mobile core site, which may in turn lead to congestion due to the migration of traffic from the paths preferred by the network planners. Similarly, as multiple PWs share the physical AC in the mobile core site, it is preferable to keep using the working AC when one working PW fails in the PSN network, which could avoid unnecessary switchover for other PWs. A fast dual-homing PW protection mechanism is needed to protect the failure in AC, the PE node and the PSN network to meet the above requirements.

[I-D.ietf-pals-mpls-tp-dual-homing-protection] describes a framework and several scenarios of dual-homing PW local protection. This document proposes a dual-homing coordination mechanism for static MPLS-TP PWs, which is used for information exchange and switchover coordination between the dual-homing PEs for the dual-homing PW local protection. The proposed mechanism has been implemented and deployed in several mobile backhaul networks which use static MPLS-TP PWs for the backhauling of mobile traffic from the radio access sites to the core site.

2. Overview of the Proposed Solution

Linear protection mechanisms for MPLS-TP network are defined in [RFC6378], [RFC7271] and [RFC7324]. When such mechanisms are applied to PW linear protection [RFC7771], both the working PW and the protection PW are terminated on the same PE node. In order to provide dual-homing protection for MPLS-TP PWs, some additional mechanisms are needed.

In MPLS-TP PW dual-homing protection, the linear protection mechanism as defined in [RFC6378] [RFC7271] and [RFC7324] on the single-homing PE (e.g. PE3 in Figure 1) is not changed, while on the dual-homing side, the working PW and protection PW are terminated on two dual-homing PEs (e.g. PE1 and PE2 in Figure 1) respectively to protect a failure occuring in a PE or a connected AC. As described in [I-D.ietf-pals-mpls-tp-dual-homing-protection], a dedicated Dual-Node Interconnection (DNI) PW is used between the two dual-homing PE nodes to forward the traffic. In order to utilize the linear protection mechanism [RFC7771] in the dual-homing PEs scenario, coordination between the dual-homing PE nodes is needed, so that the dual-homing PEs can switch the connection between the AC, the service PW and the DNI-PW properly in a coordinated fashion by the forwarder.

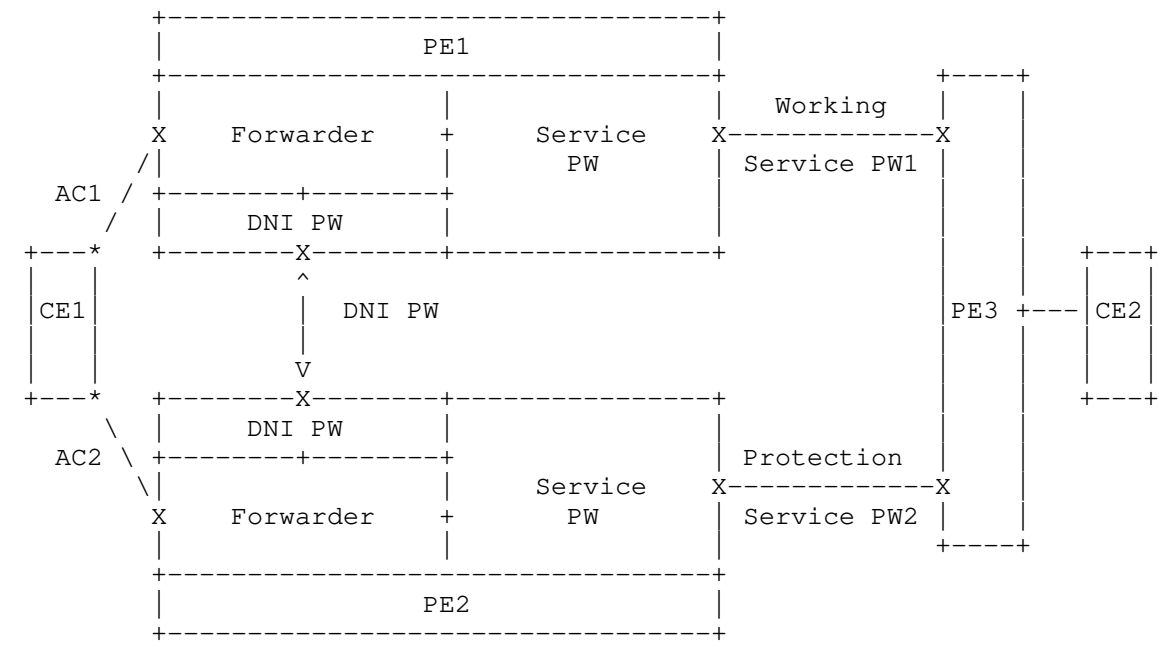


Figure 1. Dual-homing Protection with DNI-PW

3. Protocol Extensions for Dual-Homing MPLS-TP PW Protection

In dual-homing MPLS-TP PW local protection, the forwarding state of the dual-homing PEs are determined by the forwarding state machine in Table 1.

Service PW	AC	DNI PW	Forwarding Behavior
Active	Active	Up	Service PW <-> AC
Active	Standby	Up	Service PW <-> DNI PW
Standby	Active	Up	DNI PW <-> AC
Standby	Standby	Up	Drop all packets
Active	Active	Down	Service PW <-> AC
Active	Standby	Down	Drop all packets
Standby	Active	Down	Drop all packets
Standby	Standby	Down	Drop all packets

Table 1. Dual-homing PE Forwarding State Machine

In order to achieve the dual-homing MPLS-TP PW protection, coordination between the dual-homing PE nodes is needed to exchange the PW status and protection coordination requests.

3.1. Information Exchange Between Dual-Homing PEs

The coordination information will be sent on the DNI PW over the Generic Associated Channel (G-ACh) as described in [RFC5586]. A new G-ACh channel type is defined for the dual-homing coordination between the dual-homing PEs of MPLS-TP PWs. This channel type can be used for the exchange of different types of information between the dual-homing PEs. This document uses this channel type for the exchange of PW status and switchover coordination between the dual-homing PEs. Other potential usages of this channel type are for further study and are out of the scope of this document.

The MPLS-TP Dual-Homing Coordination (DHC) message is sent on the DNI PW between the dual-homing PEs. The format of the MPLS-TP DHC message is shown below:

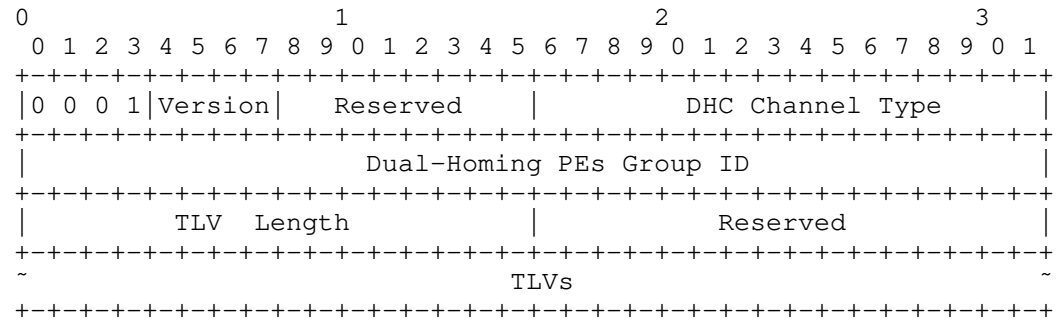


Figure 2. MPLS-TP Dual-Homing Coordination Message

The first 4-octets is the common G-ACh header as specified in [RFC5586]. The DHC Channel Type is the G-ACh channel type code point to be assigned by IANA.

The Dual-Homing Group ID is a 4-octet unsigned integer to identify the dual-homing group which the dual-homing PEs belong to. It MUST be the same at both PEs in the same group.

The TLV Length field specifies the total length in octets of the subsequent TLVs.

In this document, two TLVs are defined in MPLS-TP Dual-Homing Coordination message for dual-homing MPLS-TP PW protection:

Type	Description	Length
1	PW Status	20 Bytes
2	Dual-Node Switching	16 Bytes

The PW Status TLV is used by a dual-homing PE to report its service PW status to the other dual-homing PE in the same dual-homing group.

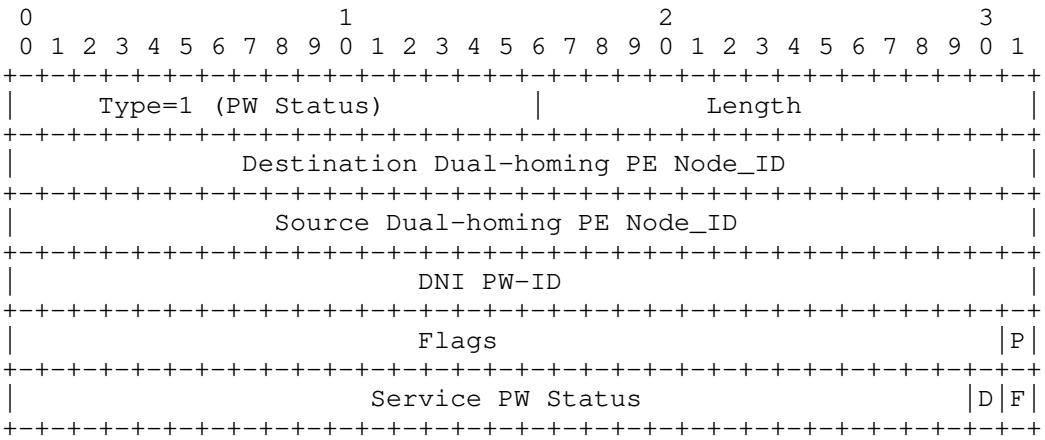


Figure 3. PW Status TLV

- The Length field specifies the length in octets of the value field of the TLV.

- The Destination Dual-homing PE Node_ID is the 32-bit identifier of the receiver PE [RFC6370] which supports both IPv4 and IPv6 environments. Usually it is the same as the LSR-ID of the receiver PE.

- The Source Dual-homing PE Node_ID is the 32-bit identifier of the sending PE [RFC6370] which supports both IPv4 and IPv6 environments. Usually it is the same as the LSR-ID of the sending PE.

- The DNI PW-ID field contains the 32-bit PW ID [RFC4447] of the DNI PW.

- The Flags field contains 32 bit flags, in which:

o The P (Protection) bit indicates whether the Source Dual-homing PE is the working PE (P=0) or the protection PE (P=1).

o Other bits are reserved for future use, which MUST be set to 0 on transmission and MUST be ignored upon receipt.

- The Service PW Status field indicates the status of the Service PW between the sending PE and the remote PE. Currently two bits are defined in the Service PW Status field:

o F bit: If set, it indicates Signal Fail (SF) [RFC6378] on the service PW. It can be either a local request generated by the PE itself or a remote request received from the remote PE.

- D bit: If set, it indicates Signal Degrade (SD) [RFC6378] on the service PW. It can be either a local request or a remote request received from the remote PE.

- Other bits are reserved for future use, which MUST be set to 0 on transmission and MUST be ignored upon receipt.

The Dual-Node Switching TLV is used by one dual-homing PE to send protection state coordination to the other PE in the same dual-homing group.

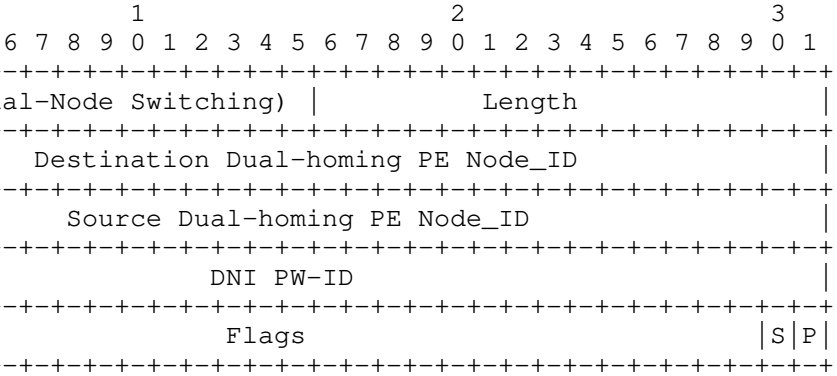


Figure 4. Dual-Node Switching TLV

- The Length field specifies the length in octets of the value field of the TLV.

- The Destination Dual-homing PE Node_ID is the 32-bit identifier of the receiver PE [RFC6370]. Usually it is the same as the LSR-ID of the receiver PE.

- The Source Dual-homing PE Node_ID is the 32-bit identifier of the sending PE [RFC6370]. Usually it is the same as the LSR-ID of the sending PE.

- The DNI PW-ID field contains the 32-bit PW-ID [RFC4447] of the DNI PW.

- The Flags field contains 32 bit flags, in which:

- The P (Protection) bit indicates whether the Source Dual-homing PE is the working PE or the protection PE. It is set to 1 when the Source PE of the dual-node switching request is the protection PE.

- The S (PW Switching) bit indicates which service PW is used for forwarding traffic. It is set to 0 when traffic will be

transported on the working PW, and is set to 1 if traffic will be transported on the protection PW. The value of the S bit is determined by the protection coordination mechanism between the dual-homing PEs and the remote PE.

- Other bits are reserved for future use, which MUST be set to 0 on transmission and MUST be ignored upon receipt.

When a change of the service PW status is detected by one of the dual-homing PEs, it MUST be reflected in the PW Status TLV and sent to the other dual-homing PE as quickly as possible to allow for fast protection switching using 3 consecutive DHC messages. This set of three messages allows for fast protection switching even if one or two of these packets are lost or corrupted. After the transmission of the three rapid messages, the dual-homing PE MUST send the most recently transmitted service PW status periodically to the other dual-homing PE on a continual basis using the DHC message.

When one dual-homing PE determines that the active service PW needs to be switched from the working PW to the protection PW, It MUST send the Dual-Node Switching TLV to the other dual-homing PE as quickly as possible to allow for fast protection switching using 3 consecutive DHC messages. After the transmission of the three messages, the protection PW would become the active service PW, and the dual-homing PE MUST send the most recently transmitted Dual-Node Switching TLV periodically to the other dual-homing PE on a continual basis using the DHC message.

It is RECOMMENDED that the default interval of the first three rapid DHC messages is 3.3 ms similar to [RFC6378], and the default interval of the subsequent messages is 1 second. Both the default interval of the three consecutive messages as well as the default interval of the periodical messages SHALL be configurable by the operator.

3.2. Protection Procedures

The dual-homing MPLS-TP PW protection mechanism can be deployed with the existing AC redundancy mechanisms. On the PSN network side, PSN tunnel protection mechanism is not required, as the dual-homing PW protection can also protect if a failure occurs in the PSN network.

This section uses the one-side dual-homing scenario as an example to describe the dual-homing PW protection procedures, the procedures for two-side dual-homing scenario would be similar.

On the dual-homing PE side, the role of working and protection PE are set by the management system or local configuration. The service PW

connecting to the working PE is the working PW, and the service PW connecting to the protection PE is called the protection PW.

On the single-homing PE side, it treats the working PW and protection PW as if they terminate on the same remote PE node, thus normal MPLS-TP protection coordination procedures still apply on the single-homing PE.

The forwarding behavior of the dual-homing PEs is determined by the components shown in the figure below:

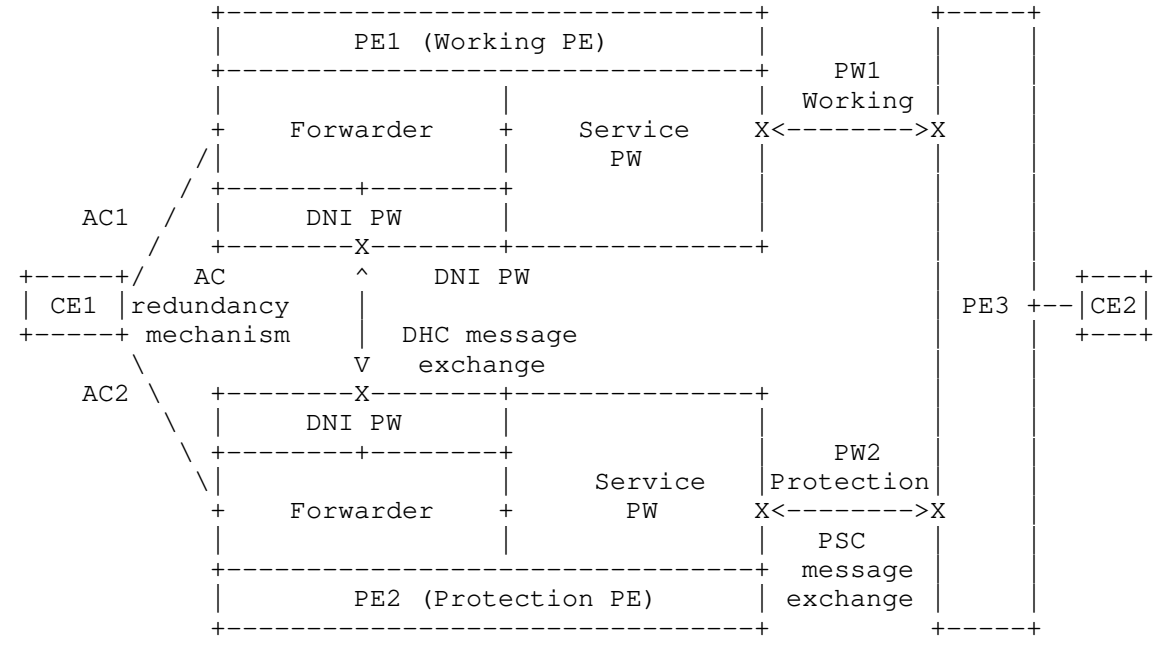


Figure 5. Components of one-side dual-homing PW protection

In Figure 5, for each dual-homing PE, the service PW is the PW used to carry service between the dual-homing PE and the remote PE. The state of the service PW is determined by the Operation Administration and Maintanence (OAM) mechanisms between the dual-homing PEs and the remote PE.

The DNI PW is provisioned between the two dual-homing PE nodes. It is used to bridge traffic when a failure occurs in the PSN network or in the ACs. The state of the DNI PW is determined by the OAM mechanism between the dual-homing PEs. Since the DNI PW is used to carry both the DHC messages and the service traffic during protection switching, it is important to ensure the robustness of the DNI PW. In order to avoid the DNI PW failure due to the failure of a

particular link, it is RECOMMENDED that multiple diverse links be deployed between the dual-homing PEs and the underlay LSP protection mechanism SHOULD be enabled.

The AC is the link which connects a dual-homing PE to the dual-homed CE. The status of AC is determined by the existing AC redundancy mechanisms, this is out of the scope of this document.

In order to perform dual-homing PW local protection, the service PW status and Dual-node switching coordination requests are exchanged between the dual-homing PEs using the DHC message defined in Section 3.1.

Whenever a change of service PW status is detected by a dual-homing PE, it MUST be reflected in the PW Status TLV and sent to the other dual-homing PE immediately using the 3 consecutive DHC messages. After the transmission of the three rapid messages, the dual-homing PE MUST send the most recently transmitted service PW status periodically to the other dual-homing PE on a continual basis using the DHC message. This way, both dual-homing PEs have the status of the working and protection PW consistently.

When there is a switchover request either generated locally or received on the protection PW from the remote PE, based on the status of the working and protection service PW, along with the local and remote request of the protection coordination between the dual-homing PEs and the remote PE, the active/standby state of the service PW can be determined by the dual-homing PEs. As the remote protection coordination request is transmitted over the protection path, in this case the active/standby status of the service PW is determined by the protection PE in the dual-homing group.

If it is determined on one dual-homing PE that switchover of service PW is needed, this dual-homing PE MUST set the S bit in the Dual-Node Switching TLV and send it to the other dual-homing PE immediately using the 3 consecutive DHC messages. With the exchange of service PW status and the switching request, both dual-homing PEs are consistent on the Active/Standby forwarding status of the working and protection service PWs. The status of the DNI PW is determined by PW OAM mechanism as defined in [RFC5085], and the status of ACs are determined by existing AC redundancy mechanisms, both are out of the scope of this document. The forwarding behavior on the dual-homing PE nodes is determined by the forwarding state machine as shown in Table 1 .

Using the topology in Figure 5 as an example, in normal state, the working PW (PW1) is in active state, the protection PW (PW2) is in standby state, the DNI PW is up, and AC1 is in active state according

to the AC redundancy mechanism. According to the forwarding state machine in Table 1, traffic will be forwarded through the working PW (PW1) and the primary AC (AC1). No traffic will go through the protection PE (PE2) or the DNI PW, as both the protection PW (PW2) and the AC connecting to PE2 are in standby state.

If a failure occurs in AC1, the state of AC2 changes to active according to the AC redundancy mechanism, while there is no change in the state of the working and protection PWs. According to the forwarding state machine in Table 1, PE1 starts to forward traffic between the working PW and the DNI PW, and PE2 starts to forward traffic between AC2 and the DNI PW. It should be noted that in this case only AC switchover takes place, in the PSN network traffic is still forwarded using the working PW.

If a failure in the PSN network brings PW1 down, the failure can be detected by PE1 or PE3 using existing OAM mechanisms. If PE1 detects the failure of PW1, it MUST inform PE2 the state of working PW using the PW Status TLV in the DHC messages and change the forwarding status of PW1 to standby. On receipt of the DHC message, PE2 SHOULD change the forwarding status of PW2 to active. Then according to the forwarding state machine in Table 1, PE1 SHOULD set up the connection between the DNI PW and AC1, and PE2 SHOULD set up the connection between PW2 and the DNI PW. According to the linear protection mechanism [RFC6378], PE2 also sends an appropriate protection coordination message [RFC6378] over the protection PW (PW2) to PE3 for the remote side to switchover from PW1 to PW2. If PE3 detects the failure of PW1, according to linear protection mechanism [RFC6378], it sends a protection coordination message on the protection PW (PW2) to inform PE2 of the failure on the working PW. Upon receipt of the message, PE2 SHOULD change the forwarding status of PW2 to active and set up the connection according to the forwarding state machine in Table 1. PE2 SHOULD send a DHC message to PE1 with the S bit set in the Dual-Node Switching TLV to coordinate the switchover on PE1 and PE2. This is useful for a unidirectional failure which cannot be detected by PE1.

If a failure brings the working PE (PE1) down, the failure can be detected by both PE2 and PE3 using existing OAM mechanisms. Both PE2 and PE3 SHOULD change the forwarding status of PW2 to active, and send a protection coordination message [RFC6378] on the protection PW (PW2) to inform the remote side to switchover. According to the existing AC redundancy mechanisms, the status of AC1 changes to standby, and the state of AC2 changes to active. According to the forwarding state machine in Table 1, PE2 starts to forward traffic between the PW2 and AC2.

4. IANA Considerations

This document requests that IANA assigns one new channel type for "MPLS-TP Dual-Homing Coordination message" from the "MPLS Generalized Associated Channel (G-ACh) Types (including Pseudowire Associated Channel Types)" registry of the "Generic Associated Channel (G-ACh) Parameters" registry.

Value	Description	Reference
TBD	MPLS-TP Dual-Homing Coordination message	[This document]

This document requests that IANA creates a new sub-registry called "MPLS-TP DHC TLVs" in the "Generic Associated Channel (G-ACh) Parameters" registry, with fields and initial allocations as follows:

Type	Description	Length	Reference
0x00	Reserved		
0x01	PW Status	20 Bytes	[this document]
0x02	Dual-Node Switching	16 Bytes	[this document]

The allocation policy for this registry is IETF Review as specified in [RFC5226].

5. Security Considerations

MPLS-TP is a subset of MPLS and so builds upon many of the aspects of the security model of MPLS. Please refer to [RFC5920] for generic MPLS security issues and methods for securing traffic privacy and integrity.

The DHC message defined in this document contains control information, if it is injected or modified by an attacker, the dual-homing PEs might not agree on which PE should be used to deliver the CE traffic, and this could be used as a denial of service attack against the CE. It is important that the DHC message is used within a trusted MPLS-TP network domain as described in [RFC6941].

The DHC message is carried in the G-ACh [RFC5586], so it is dependent on the security of the G-ACh itself. The G-ACh is a generalization of the Associated Channel defined in [RFC4385]. Thus, this document relies on the security mechanisms provided for the Associated Channel as described in those two documents.

As described in the security considerations of [RFC6378], the G-ACh is essentially connection oriented so injection or modification of control messages requires the subversion of a transit node. Such subversion is generally considered hard in connection oriented MPLS networks and impossible to protect against at the protocol level.

Management level techniques are more appropriate. The procedures and protocol extensions defined in this document do not affect the security model of MPLS-TP linear protection as defined in [RFC6378].

Uniqueness of the identifiers defined in this document is guaranteed by the assigner (e.g. the operator). Failure by an assigner to use unique values within the specified scoping for any of the identifiers defined herein could result in operational problems. Please refer to [RFC6370] for more details about the uniqueness of the identifiers.

6. Contributors

The following individuals substantially contributed to the content of this document:

Kai Liu
Huawei Technologies
Email: alex.liukai@huawei.com

Shahram Davari
Broadcom Corporation
davari@broadcom.com

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, DOI 10.17487/RFC4447, April 2006, <http://www.rfc-editor.org/info/rfc4447>.

[RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <http://www.rfc-editor.org/info/rfc5085>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <http://www.rfc-editor.org/info/rfc5586>.

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <http://www.rfc-editor.org/info/rfc6370>.

[RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <http://www.rfc-editor.org/info/rfc6378>.

[RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <http://www.rfc-editor.org/info/rfc7271>.

[RFC7324] Osborne, E., "Updates to MPLS Transport Profile Linear Protection", RFC 7324, DOI 10.17487/RFC7324, July 2014, <http://www.rfc-editor.org/info/rfc7324>.

7.2. Informative References

[I-D.ietf-pals-mpls-tp-dual-homing-protection] Cheng, W., Wang, L., Li, H., Davari, S., and J. Dong, "Dual-Homing Protection for MPLS and MPLS-TP Pseudowires", draft-ietf-pals-mpls-tp-dual-homing-protection-05 (work in progress), January 2017.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <http://www.rfc-editor.org/info/rfc4385>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <http://www.rfc-editor.org/info/rfc5920>.

[RFC6372] Sprecher, N., Ed. and A. Farrel, Ed., "MPLS Transport Profile (MPLS-TP) Survivability Framework", RFC 6372, DOI 10.17487/RFC6372, September 2011, <http://www.rfc-editor.org/info/rfc6372>.

[RFC6718] Muley, P., Aissaoui, M., and M. Bocci, "Pseudowire Redundancy", RFC 6718, DOI 10.17487/RFC6718, August 2012, <http://www.rfc-editor.org/info/rfc6718>.

[RFC6870] Muley, P., Ed. and M. Aissaoui, Ed., "Pseudowire Preferential Forwarding Status Bit", RFC 6870, DOI 10.17487/RFC6870, February 2013, <http://www.rfc-editor.org/info/rfc6870>.

[RFC6941] Fang, L., Ed., Niven-Jenkins, B., Ed., Mansfield, S., Ed., and R. Graveman, Ed., "MPLS Transport Profile (MPLS-TP) Security Framework", RFC 6941, DOI 10.17487/RFC6941, April 2013, <http://www.rfc-editor.org/info/rfc6941>.

[RFC7771] Malis, A., Ed., Andersson, L., van Helvoort, H., Shin, J., Wang, L., and A. D'Alessandro, "Switching Provider Edge (S-PE) Protection for MPLS and MPLS Transport Profile (MPLS-TP) Static Multi-Segment Pseudowires", RFC 7771, DOI 10.17487/RFC7771, January 2016, <http://www.rfc-editor.org/info/rfc7771>.

Authors' Addresses

Weiqiang Cheng
China Mobile
No.32 Xuanwumen West Street
Beijing 100053
China

Email: chengweiqiang@chinamobile.com

Lei Wang
China Mobile
No.32 Xuanwumen West Street
Beijing 100053
China

Email: Wangleiyj@chinamobile.com

Han Li
China Mobile
No.32 Xuanwumen West Street
Beijing 100053
China

Email: Lihan@chinamobile.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: jie.dong@huawei.com

Alessandro D'Alessandro
Telecom Italia
via Reiss Romoli, 274
Torino 10148
Italy

Email: alessandro.dalessandro@telecomitalia.it

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros(Ed.)
VMware

Siva Sivabalan(Ed.)
Cisco Systems

Updates: RFC7385

Expires: May 16, 2018

November 12, 2017

Signaling Root-Initiated Point-to-Multipoint Pseudowire using LDP
draft-ietf-pals-p2mp-pw-04.txt

Abstract

This document specifies a mechanism to signal Point-to-Multipoint (P2MP) Pseudowires (PW) tree using LDP. Such a mechanism is suitable for any Layer 2 VPN service requiring P2MP connectivity over an IP or MPLS enabled PSN. A P2MP PW established via the proposed mechanism is root initiated. This document updates RFC7385 by re-assigning reserved value 0xFF to be the wildcard transport tunnel type.

Table of Contents

1 Introduction

A Point-to-Multipoint (P2MP) Pseudowire (PW) emulates the essential attributes of a unidirectional P2MP Telecommunications service such as P2MP ATM over PSN. A major difference between a Point-to-Point (P2P) PW outlined in [RFC3985] and a P2MP PW is that the former is intended for bidirectional service whereas the latter is intended for both unidirectional, and optionally bidirectional service. Requirements for P2MP PW are described in [RFC7338]. P2MP PW can be constructed as either Single Segment (P2MP SS-PW) or Multi Segment (P2MP MS-PW) Pseudowires as mentioned in [RFC7338]. P2MP MS-PW is outside the scope of this document. A reference model or a P2MP PW is depicted in Figure 1 below. A transport LSP associated with a P2MP SS-PW SHOULD be a P2MP MPLS LSP (i.e., P2MP TE tunnel established via RSVP-TE [RFC4875] or P2MP LSP established via mLDP [RFC6388]) spanning from the Root-PE to the Leaf-PE(s) of the P2MP SS-PW tree. For example, in Figure 1, PW1 can be associated with a P2MP TE tunnel or P2MP LSP setup using mLDP originating from PE1 and terminating at PE2, PE3 and PE4.

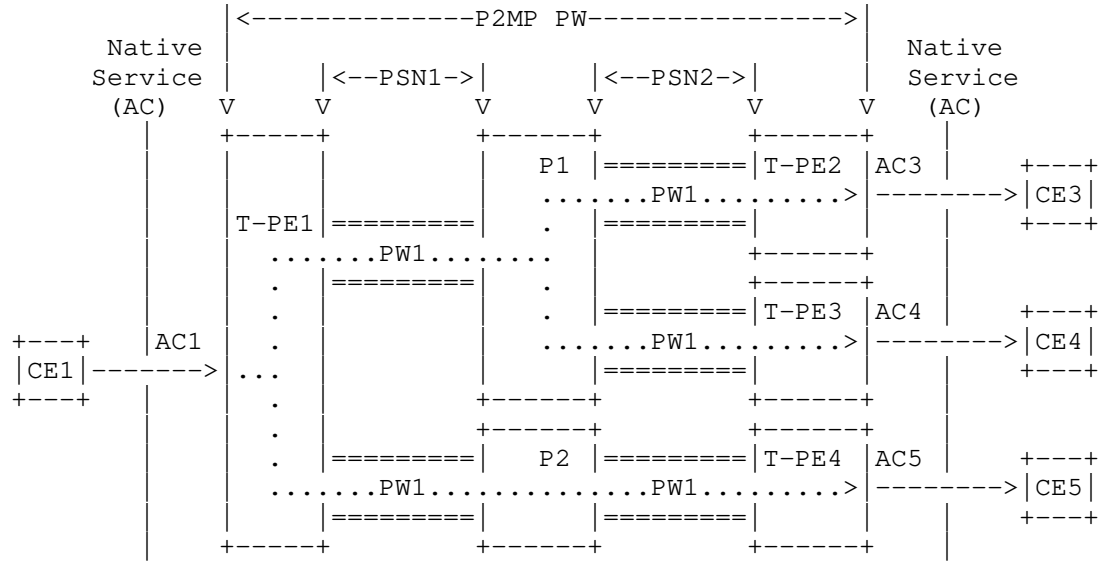


Figure 1: P2MP PW

Mechanisms for establishing P2P SS-PW using LDP are described in [RFC4447bis]. This document specify a method of signaling P2MP PW using LDP. In particular, this document defines new FEC, TLVs, parameters, and status codes to facilitate LDP to signal and maintain P2MP PWs.

As outlined in [RFC7338], even though the traffic flow from a Root-PE (R-PE) to Leaf-PE(s) (L-PEs) is P2MP in nature, it may be desirable for any L-PE to send unidirectional P2P traffic destined only to the R-PE. The proposed mechanism takes such option into consideration.

The P2MP PW requires an MPLS LSP to carry the PW traffic, and the MPLS packets carrying the PW upstream label will be encapsulated according to the methods described in [RFC5332].

1.1 Terminology

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].

FEC: Forwarding Equivalence Class

LDP: Label Distribution Protocol

mLDP: Label Distribution Protocol for P2MP/MP2MP LSP

LSP: Label Switching Path

MS-PW: Multi-Segment Pseudowire

P2P: Point to Point

P2MP: Point to Multipoint

PE: Provider Edge

PSN: Packet Switched Network

PW: Pseudowire

SS-PW: Single-Segment Pseudowire

S-PE: Switching Provider Edge of MS-PW

TE: Traffic Engineering

R-PE: Root-PE - ingress PE, PE initiating P2MP PW setup.

L-PE: Leaf-PE - egress PE.

2. Signaling P2MP PW

In order to advertise labels as well as exchange PW related LDP messages, PEs must establish LDP sessions among themselves. A PE discovers other PEs that are to be connected via P2MP PWs either via manual configuration or autodiscovery [RFC6074].

R-PE and each L-PE MUST be configured with the same FEC as defined in the following section.

P2MP PW requires that there is an active P2MP PSN LSP set up between R-PE and L-PE(s). Note that the procedure to set up the P2MP PSN LSP is different depending on the signaling protocol used (RSVP-TE or mLDP).

In case of mLDP, a Leaf-PE can decide to join the P2MP LSP at any time. In the case of RSVP-TE, the P2MP LSP is set up by the R-PE, generally at the initial service provisioning time. It should be noted that local policy can override any decision to join, add or prune existing or new L-PE(s) from the tree. In any case, the PW setup can ignore these differences, and simply assume that the P2MP PSN LSP is available when needed.

P2MP PW signaling is initiated by the R-PE which sends a separate P2MP-PW LDP label mapping message to each of the the L-PE(s) belonging to that P2MP PW. This label mapping message will contain the following:

1. A FEC TLV containing P2MP PW Upstream FEC element that includes Transport LSP sub TLV.
2. An Interface Parameters TLV, as described in [RFC4447bis].
3. A PW Grouping TLV, as described in [RFC4447bis].
4. A label TLV for the upstream-assigned label used by R-PE for the traffic going from R-PE to L-PE(s).

The R-PE imposes the upstream-assigned PW label on the outbound packets sent over the P2MP-PW, and using this label an L-PE identifies the inbound packets arriving over the P2MP PW.

Additionally, the R-PE MAY send label mapping message(s) to one or more L-PE(s) to signal unidirectional P2P PW(s). The L-PE(s) can use such PW(s) to send traffic to the R-PE. This optional label mapping message will contain the following:

1. P2P PW Downstream FEC element.
2. A label TLV for the down-stream assigned label used by the corresponding L-PE to send traffic to the R-PE.

The LDP liberal label retention mode MUST be used, and per requirements specified in [RFC5036], the Label Request message MUST also be supported.

The upstream-assigned label is allocated according to the rules in [RFC5331].

When an L-PE receives a PW Label Mapping Message, it MUST verify the associated P2MP PSN LSP is in place. If the associated P2MP PSN LSP is not in place, and its type is LDP P2MP LSP, the L-PE MUST attempt to join the P2MP LSP associated with the P2MP PW. If the associated P2MP PSN LSP is not in place, and its type is RSVP-TE P2MP LSP, the L-PE SHOULD wait till the P2MP transport LSP is signaled. If an L-PE fails to join the P2MP PSN LSP, that L-PE MUST not enable the PW, and MUST notify the user. In this case, a PW status message with status code of 0x00000008 (Local PSN-facing PW (ingress) Receive Fault) MUST also be sent to the R-PE.

2.1 PW ingress to egress incompatibility issues

If an R-PE signals a PW with a pw type, CW mode, or interface parameters that a particular L-PE cannot accept, then the L-PE MUST not enable the PW, and notify the user. In this case, a PW status message with status code of 0x00000001 (Pseudowire Not Forwarding) MUST also be sent to the R-PE.

Note that this procedure does not apply if the L-PE had not been provisioned with this particular P2MP PW. In this case according to the LDP liberal label retention rules, no action is taken.

2.2 P2MP PW FEC

[RFC4447bis] specifies two types of LDP FEC elements called "PWid FEC Element" and "Generalized PWid FEC Element" used to signal P2P PWs. This document defines two new types of FEC elements called "P2MP PW Upstream FEC Element" and "P2P PW Downstream FEC Element". These FEC elements are associated with a mandatory upstream assigned label and an optional downstream assigned label respectively.

2.2.1 P2MP PW Upstream FEC Element

A new FEC type for the P2MP PW Upstream FEC Element is encoded as follows:

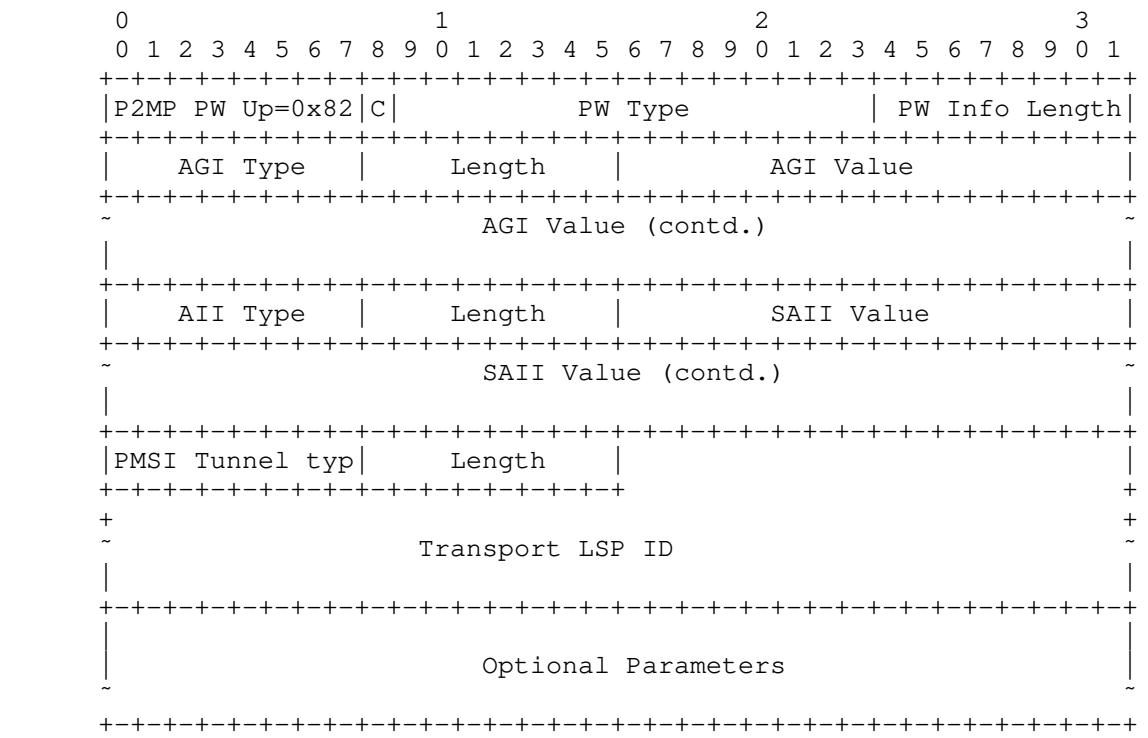


Figure 2: P2MP PW Upstream FEC Element

* P2MP PW Up:

8 bits representation for the P2MP PW Upstream FEC type.

* PW Type:

15 bits representation of PW type as specified in [RFC4447bis].

* C bit:

A value of 1 or 0 indicates whether control word is present or absent for the P2MP PW.

* PW Info Length:

Sum of the lengths of AGI, SAII, PMSI Tunnel info, and Optional Parameters field in octets. If this value is 0, then it references all PWs using the specified grouping ID. In this case, there are neither other FEC element fields (AGI, SAII, etc.) present, nor any

interface parameters TLVs. Alternatively, typed wildcard FEC described in section 3.3, can be used to achieve the same or to have better filtering.

* AGI:

Attachment Group Identifier can be used to uniquely identify VPN or VPLS instance associated with the P2MP PW. This has the same format as the Generalized PWid FEC element [RFC4447bis].

* SAII:

Source Attachment Individual Identifier is used to identify the root of the P2MP PW. The root is represented using AII type 2 format specified in [RFC5003]. Note that the SAII can be omitted by simply setting the length and type to zero.

P2MP PW is identified by the Source Attachment Identifier (SAI). If the AGI is non-null, the SAI is the combination of the SAII and the AGI, if the AGI is null, the SAI is the SAII.

* PMSI Tunnel info

PMSI Tunnel info is the combination of PMSI Tunnel Type, Length and Transport LSP ID.

A P2MP PW MUST be associated with a transport LSP which can be established using RSVP-TE or mLDP.

* PMSI Tunnel Type:

The PMSI tunnel type is defined in [RFC6514].

When the type is set to mLDP P2MP LSP, the Tunnel Identifier is a P2MP FEC Element as defined in [RFC6388]. A new mLDP Opaque Value Element type for L2VPN-MCAST application as specified in the IANA considerations MUST be used.

* Transport LSP ID: This is the Tunnel Identifier which is defined in [RFC6514].

An R-PE sends Label Mapping Message as soon as the transport LSP ID associated with the P2MP PW is known (e.g., via configuration) regardless of the operational state of that transport LSP. Similarly, an R-PE does not withdraw the labels when the corresponding transport LSP goes down. Furthermore, an L-PE retains the P2MP PW labels regardless of the operational status of the transport LSP.

Note that a given transport LSP can be associated with more than one P2MP PWs in which case P2MP PWs will be sharing the same R-PE and L-PE(s). An R-PE may also have many P2MP PWs with disjoint L-PE sets.

In the case of LDP P2MP LSP, when an L-PE receives the Label Mapping Message, it can initiate the process of joining the P2MP LSP tree associated with the P2MP PW.

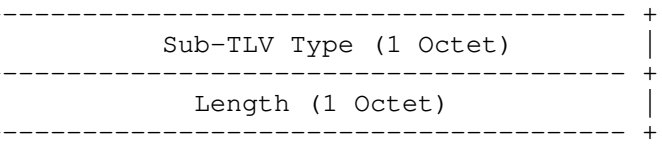
In the case of RSVP-TE P2MP LSP, only the R-PE initiates the signaling of P2MP LSP.

* Optional Parameters:

The Optional Parameter field can contain some TLVs that are not part of the FEC, but are necessary for the operation of the PW. This proposed mechanism uses two such TLVs: Interface Parameters TLV, and Group ID TLV.

The Interface Parameters TLV and Group ID TLV specified in [RFC4447bis] can also be used in conjunction with P2MP PW FEC in a label message. For Group ID TLV, the sender and receiver of these TLVs should follow the same rules and procedures specified in [RFC4447bis]. For Interface Parameters TLV, the procedure differs from the one specified in [RFC4447bis] due to specifics of P2MP connectivity. When the interface parameters are signaled by a R-PE, each L-PE must check if its configured value(s) is less than or equal to the threshold value provided by the R-PE (e.g. MTU size (Ethernet), max number of concatenated ATM cells, etc)). For other interface parameters like CEP/TDM Payload bytes (TDM), the value MUST exactly match the values signaled by the R-PE.

Multicast traffic stream associated with a P2MP PW can be selective or inclusive. To support the former, this document defines a new optional Selective Tree Interface Parameter sub-TLV, as described in the IANA considerations and according to the format described in [RFC4447bis]. The value of the sub-TLV contains the source and the group for a given multicast tree as shown in Figure 3. Also, if a P2MP PW is associated with multiple selective trees, the corresponding label mapping message will carry more than one instance of this Sub-TLV. Furthermore, in the absence of this sub-TLV, the P2MP PW is associated with all multicast traffic stream originating from the root.



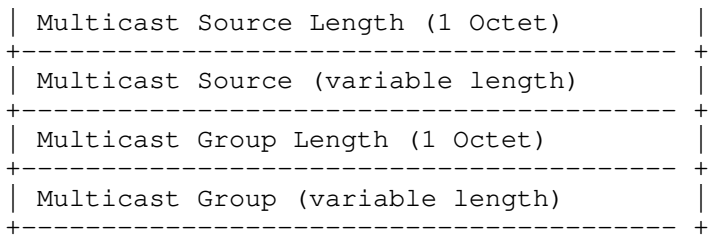


Figure 3: Selective Tree Interface Parameter Sub-TLV Value

Note that since the LDP label mapping message is only sent by the R-PE to all the L-PEs, it is not possible to negotiate any interface parameters.

2.2.2 P2P PW Downstream FEC Element

The optional P2P PW Downstream FEC Element is encoded as follows:

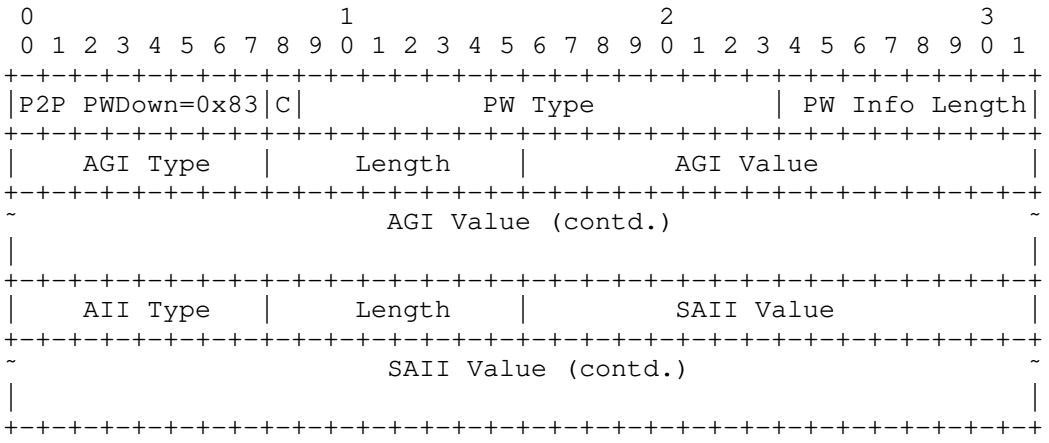


Figure 4: P2P PW Downstream FEC Element

The definition of the fields in the P2P PW Downstream FEC Element is the same as those of P2MP PW Upstream FEC Element shown in Figure 2.

2.3 Typed Wildcard FEC Format for new FEC

[RFC5918] defines the general notion of a "Typed Wildcard" FEC Element, and requires FEC designer to specify a typed wildcard FEC element for newly defined FEC element types. This document defines two new FEC elements, "P2MP PW Upstream" and "P2P PW Downstream" FEC

element, and hence requires us to define their Typed Wildcard format.

[RFC6667] defines Typed Wildcard FEC element format for other PW FEC Element types (PWid and Gen. PWid FEC Element) in section 2 as follows:

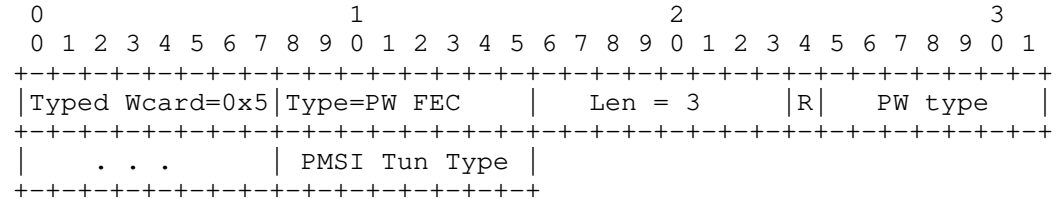


Figure 5: Typed Wildcard Format for P2MP PW FEC Elements

[RFC6667] specifies that "Type" field can be either "PWid" (0x80) or "Generalized PWid" (0x81) FEC element type. This document reuses the existing typed wildcard format as specified in [RFC6667] and illustrated in Figure 5 and extends the definition of "Type" field to also include "P2MP PW Upstream" and "P2P PW Downstream" FEC element types. This document adds an additional field "PMSI Tun Type". This document reserves PMSI tunnel Type 0xFF to mean "wildcard" transport tunnel type. The PMSI tunnel Type field only applies to Typed wildcard P2MP PW Upstream FEC and MUST be set to "wildcard" for "P2P PW Downstream FEC" typed wildcard element.

2.4 Group ID usage

The Grouping TLV as defined in [RFC4447bis] contains a group ID capable of indicating an arbitrary group membership of a P2MP-PW. This group ID can be used in LDP "wild card" status, and withdraw label messages, as described in [RFC4447bis].

2.5 Generic Label TLV

As in the case of P2P PW signaling, P2MP PW labels are carried within Generic Label TLV contained in LDP Label Mapping Message. A Generic Label TLV is formatted and processed as per the rules and procedures specified in [RFC4447bis]. For a given P2MP PW, a single upstream-assigned label is allocated by the R-PE, and is advertised to all L-PEs using the Generic Label TLV in label mapping message containing the P2MP PW Upstream FEC element.

The R-PE can also allocate a unique label for each L-PE from which it intends to receive P2P traffic. Such a label is advertised to the L-PE using Generic Label TLV and P2P PW Downstream FEC in label mapping message.

3. LDP Capability Negotiation

The capability of supporting P2MP PW MUST be advertised to all LDP peers. This is achieved by using the methods in [RFC5561] to advertise the LDP "P2MP PW Capability" TLV. If an LDP peer supports the dynamic capability advertisement, this can be done by sending a new Capability message with the S bit set for the P2MP PW capability TLV. If the peer does not supports dynamic capability advertisement, then the P2MP PW Capability TLV MUST be included in the LDP Initialization message during the session establishment. An LSR having P2MP PW capability MUST recognize both P2MP PW Upstream FEC Element and P2P PW Downstream FEC Element in LDP label messages.

In line with requirements listed in [RFC5561], the following TLV is defined to indicate the P2MP PW capability:

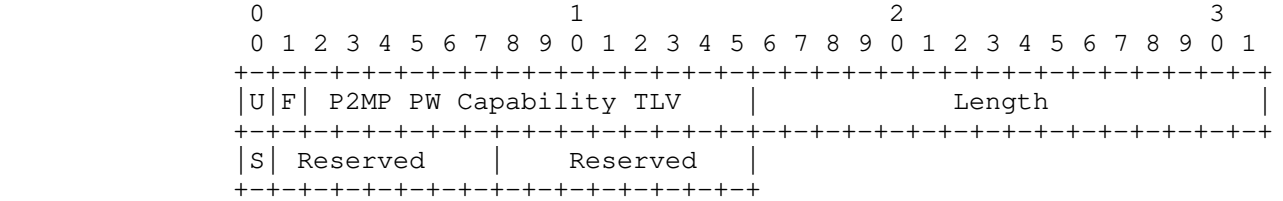


Figure 7: LDP P2MP PW Capability TLV

* U-bit:

  SHOULD be 1 (ignore if not understood).

* F-bit:

  SHOULD be 0 (don't forward if not understood).

* P2MP PW Capability TLV Code Point:

  The TLV type, which identifies a specific capability. The P2MP PW capability code point is requested in the IANA allocation section below.

* S-bit:

  The State Bit indicates whether the sender is advertising or withdrawing the P2MP PW capability. The State bit is used as follows:

  1 - The TLV is advertising the capability specified by the TLV Code Point.

0 - The TLV is withdrawing the capability specified by the TLV Code Point.

* Length:

MUST be set to 2 (octet).

4. P2MP PW Status

In order to support the proposed mechanism, an LSR MUST be capable of handling PW status. As such, PW status negotiation procedure described in [RFC4447bis] is not applicable to P2MP PW. An LSR MUST NOT claim to be P2MP PW capable by sending a LDP P2MP PW Capability TLV if it is not also capable of handling PW status.

Once an L-PE successfully processes a Label Mapping Message for a P2MP PW, it MUST send appropriate PW status according to the procedure specified [RFC4447bis] to report the PW status. If no PW status notification is required, then no PW status notification is sent (for example if the P2MP PW is established and operational with a status code of Success (0x00000000), a PW status message is not necessary). A PW status message sent from an L-PE to R-PE MUST contain the P2P PW Downstream FEC to identify the PW.

An R-PE also sends PW status to L-PE(s) to reflect its view of a P2MP PW state. Such PW status message contains P2MP PW Upstream FEC to identify the PW.

Connectivity status of the underlying P2MP LSP that P2MP PW is associated with, can be verified using LSP Ping and Traceroute procedures described in [RFC6425].

5 Security Considerations

In general the security measures described in [RFC4447bis] are adequate for this protocol. However the use of MD5 as the method of securing an LDP control plane is no longer considered adequately secure. Implementations should be written in such a way that they can migrate to a more secure cryptographic hash function when the next authentication method to be used in the LDP might not be simple hash based authentication algorithm.

6 Acknowledgment

Authors would like thank Andre Pelletier and Parag Jain for their valuable suggestions.

7 IANA Considerations

7.1. FEC Type Name Space

This document uses two new FEC element types, number 0x82 and 0x83 are suggested for assignment from the registry "FEC Type Name Space" for the Label Distribution Protocol (LDP RFC5036):

Value	Hex	Name	Reference
130	0x82	P2MP PW Upstream FEC Element	RFCxxxx
131	0x83	P2P PW Downstream FEC Element	RFCxxxx

7.2. LDP TLV Type

This document uses a new LDP TLV types, IANA already maintains a registry of name "TLV TYPE NAME SPACE" defined by RFC5036. The following values are suggested for assignment:

TLV type Description:

0x0703 P2MP PW Capability TLV

7.3. mLDP Opaque Value Element TLV Type

This document requires allocation of a new mLDP Opaque Value Element Type from "LDP MP Opaque Value Element basic type" name space defined in [RFC6388].

The following value is suggested for assignment:

TLV type Description
13 L2VPN-MCAST application TLV

Length: 4

Value: A 32-bit integer, unique in the context of the root, as identified by the root’s address.

7.4. Selective Tree Interface Parameter sub-TLV Type

This document requires allocation of a sub-TLV from the registry "Pseudowire Interface Parameters Sub-TLV Type" defined in [RFC4446].

The following value is suggested for assignment:

TLV type Description

0x1b Selective Tree Interface Parameter.

7.5. WildCard PMSI tunnel type

This document requests that IANA modify the following entry in the "P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types" registry within the "Border Gateway Protocol (BGP) Parameters" namespace previously assigned by RFC7385 as "reserved".

Value	Meaning	Reference
0xFF	wildcard transport tunnel type	[This document]

8 References

8.1. Normative References

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

[RFC4447bis] "Pseudowire Setup and Maintenance using the Label Distribution Protocol", Martini, L., et al., draft-ietf-pals-rfc4447bis-05.txt, July 2016.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5003] C. Metz, L. Martini, F. Balus, J. Sugimoto, "Attachment Individual Identifier (AII) Types for Aggregation", RFC5003, September 2007.

[RFC5331] R. Aggarwal, Y. Rekhter, E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, August 2008.

[RFC5332] T. Eckert, E. Rosen, Ed.,R. Aggarwal, Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, August 2008.

[RFC6388] I. Minei, K. Kompella, I. Wijnands, B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011.

[RFC4875] R. Aggarwal, Ed., D. Papadimitriou, Ed., S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs).", RFC 4875, May 2007.

[RFC6514] R. Aggarwal, E. Rosen, T. Morin, Y. Rekhter, "BGP Encodings

and Procedures for Multicast in MPLS/BGP IP VPNs", RFC6514, February 2012.

[RFC5561] B.Thomas, K.Raza, S.Aggarwal, R.Agarwal, JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009.

[RFC5918] R. Asati, I. Minei, and B. Thomas, "LDP Typed Wildcard Forwarding Equivalence Class", RFC 5918, August 2010.

[RFC6667] K. Raza, S. Boutros, and C. Pignataro, "LDP Typed Wildcard FEC for PWid and Generalized PWid FEC Elements", RFC 6667, July 2012.

8.2. Informative References

[RFC3985] Stewart Bryant, et al., "PWE3 Architecture", RFC3985

[RFC6074] E. Rosen,W. Luo,B. Davie,V. Radoaca "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC6074, January 2011.

[RFC7338] F. Jounay, et. al, "Requirements for Point to Multipoint Pseudowire", RFC7338, September 2014.

[RFC6425] A. Farrel, S. Yasukawa, "Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS)- Extensions to LSP Ping", RFC6425, November 2011.

Contributors

The following co-authors have also contributed to this document:

Luca Martini
Cisco Systems, Inc.
Email: lmartini@cisco.com

Maciek Konstantynowicz
Cisco Systems, Inc.
e-mail: maciek@cisco.com

Gianni Del Vecchio
Swisscom
Email: Gianni.DelVecchio@swisscom.com

Thomas D. Nadeau
Brocade

Email: tnadeau@lucidvision.com

Frederic Jounay
Orange CH
Email: Frederic.Jounay@salt.ch

Philippe Niger
Orange CH
Email: philippe.niger@orange.com

Yuji Kamite
NTT Communications Corporation
Email: y.kamite@ntt.com

Lizhong Jin
Email: lizho.jin@gmail.com

Martin Vigoureux
Nokia
Email: martin.vigoureux@nokia.com

Laurent Ciavaglia
Nokia
Email: laurent.ciavaglia@nokia.com

Simon Delord
Telstra
E-mail: simon.delord@gmail.com

Kamran Raza
Cisco Systems
Email: skraza@cisco.com

Authors' Addresses

Sami Boutros
VMware Inc.
Email: sboutros@vmware.com

Siva Sivabalan
Cisco Systems, Inc.
Email: msiva@cisco.com

BESS Workgroup
Internet-Draft
Intended status: Standards Track
Expires: September 22, 2016

P. Jain
Cisco Systems, Inc.
S. Boutros
VMWare, Inc.
S. Aldrin
Google Inc.

March 21, 2016

Definition of P2MP PW TLV for LSP-Ping Mechanisms
draft-jain-bess-p2mp-pw-lsp-ping-03

Abstract

LSP-Ping is a widely deployed Operation, Administration, and Maintenance (OAM) mechanism in MPLS networks. This document describes a mechanism to verify connectivity of Point-to-Multipoint (P2MP) Pseudowires (PW) using LSP Ping.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 22, 2016.

Copyright Notice

Table of Contents

1. Introduction

A Point-to-Multipoint (P2MP) Pseudowire (PW) emulates the essential attributes of a unidirectional P2MP Telecommunications service such as P2MP ATM over PSN. Requirements for P2MP PW are described in [RFC7338]. P2MP PWs are carried over P2MP MPLS LSP. The Procedures for P2MP PW signaling using BGP are described in [RFC7117] and LDP for single segment P2MP PWs are described in [I-D.ietf-pwe3-p2mp-pw]. Many P2MP PWs can share the same P2MP MPLS LSP and this arrangement is called Aggregate P-tree. The aggregate P2MP trees require an upstream assigned label so that on the tail of the P2MP LSP, the traffic can be associated with a VPN or a VPLS instance. When a P2MP MPLS LSP carries only one VPN or VPLS service instance, the arrangement is called Inclusive P-Tree. For Inclusive P-Trees, P2MP MPLS LSP label itself can uniquely identify the VPN or VPLS service being carried over P2MP MPLS LSP. The P2MP MPLS LSP can also be used in Selective P-Tree arrangement for carrying multicast traffic. In a Selective P-Tree arrangement, traffic to each multicast group in a VPN or VPLS instance is carried by a separate unique P-tree. In Aggregate Selective P-tree arrangement, traffic to a set of multicast groups from different VPN or VPLS instances is carried over a same shared P-tree.

The P2MP MPLS LSP are setup either using P2MP RSVP-TE [RFC4875] or Multipoint LDP (mDLP) [RFC6388]. Mechanisms for fault detection and isolation for data plane failures for P2MP MPLS LSPs are specified in

[RFC6425]. This document describes a mechanism to detect data plane failures for P2MP PW carried over P2MP MPLS LSPs.

This document defines a new P2MP Pseudowire sub-TLV for Target FEC Stack for P2MP PW. The P2MP Pseudowire sub-TLV is added in Target FEC Stack TLV by the originator of the Echo Request to inform the receiver at P2MP MPLS LSP tail, of the P2MP PW being tested.

Multi-segment Pseudowires support is out of scope of this document at present and may be included in future.

2. Specification of Requirements

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 [RFC2119].

3. Terminology

ATM: Asynchronous Transfer Mode

LSR: Label Switching Router

MPLS-OAM: MPLS Operations, Administration and Maintenance

P2MP-PW: Point-to-Multipoint PseudoWire

PW: PseudoWire

TLV: Type Length Value

4. Identifying a P2MP PW

This document introduces a new LSP Ping Target FEC Stack sub-TLV, P2MP Pseudowire sub-TLV, to identify the P2MP PW under test at the P2MP LSP Tail/Bud node.

4.1. P2MP Pseudowire Sub-TLV

The P2MP Pseudowire sub-TLV has the format shown in Figure 1. This TLV is included in the echo request sent over P2MP PW by the originator of request.

The Attachment Group Identifier (AGI) in P2MP Pseudowire Sub-TLV as described in Section 3.4.2 in [RFC4446], identifies the VPLS instance. The Originating Router's IP address is the IPv4 or IPv6 address of the P2MP PW root.

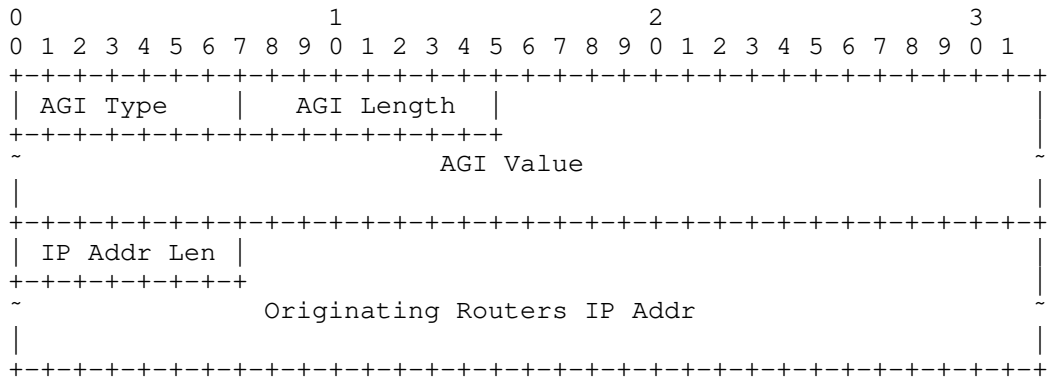


Figure 1: P2MP Pseudowire sub-TLV format

For Inclusive and Selective P2MP MPLS P-trees, the echo request is sent using the P2MP MPLS LSP label.

For Aggregate Inclusive and Aggregate Selective P-trees, the echo request is sent using a label stack of [P2MP MPLS P-tree label, upstream assigned P2MP PW label]. The P2MP MPLS P-tree label is the outer label and upstream assigned P2MP PW label is inner label.

5. Encapsulation of OAM Ping Packets

The LSP Ping Echo request IPv4/UDP packets will be encapsulated with the MPLS label stack as described in previous sections, followed by the GAL Label [RFC6426]. The GAL label will be followed by the ACH with the Pseudowire Associated Channel Type 16 bit value in the ACH set to IPv4 indicating that the carried packet is an IPv4 packet.

6. Operations

In this section, we explain the operation of the LSP Ping over P2MP PW. Figure 2 shows a P2MP PW PW1 setup from T-PE1 to remote PEs (T-PE2, T-PE3 and T-PE4). The transport LSP associated with the P2MP PW1 can be MLDP P2MP MPLS LSP or P2MP TE tunnel.

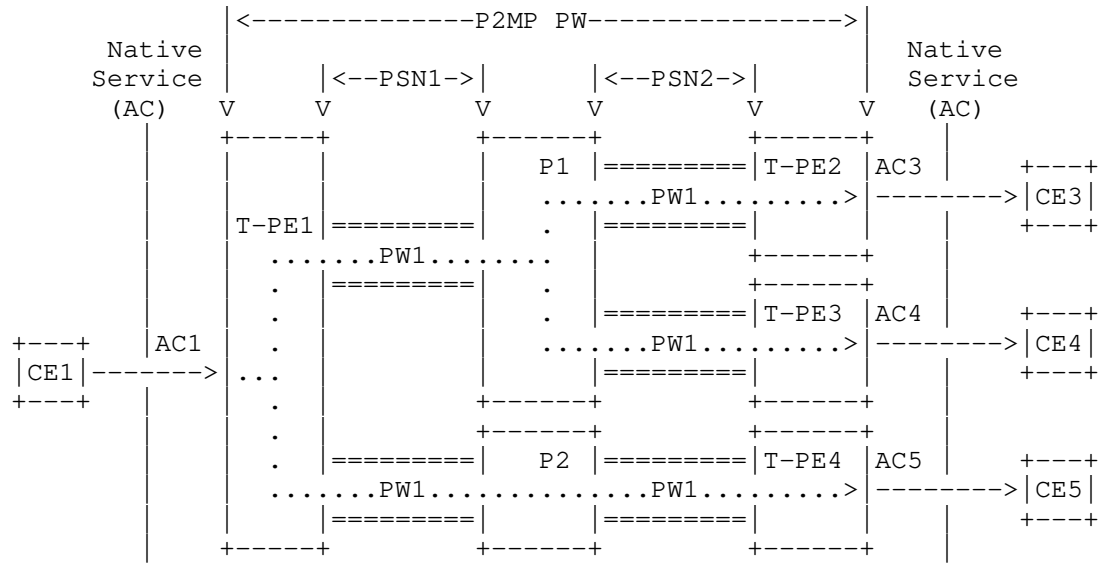


Figure 2: P2MP PW

When an operator wants to perform a connectivity check for the P2MP PW1, the operator initiate a LSP-Ping request with the Target FEC Stack TLV containing P2MP Pseudowire sub-TLV in the echo request packet. For an Inclusive P2MP P-tree arrangement, the echo request packet is sent over the P2MP MPLS LSP with {P2MP P-tree label, GAL} MPLS label stack and IP ACH Channel header. For an Aggregate Inclusive P-tree arrangement, the echo request packet is sent over the P2MP MPLS LSP with {P2MP P-tree label, P2MP PW upstream assigned label, GAL} MPLS label stack and IP ACH Channel header. The intermediate P router will do swap and replication based on the MPLS LSP label. Once the echo request packet reaches remote terminating PEs, T-PE1s will use the GAL label and the IP ACH Channel header to determine that the packet is IPv4 OAM Packet. The T-PEs will process the packet and perform checks for the P2MP Pseudowire sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC4379] and respond according to [RFC4379] processing rules.

7. Controlling Echo Responses

The procedures described in [RFC6425] for preventing congestion of Echo Responses (Echo Jitter TLV) and limiting the echo reply to a single egress node (Node Address P2MP Responder Identifier TLV) can be applied to P2MP PW LSP Ping.

8. Security Considerations

The proposal introduced in this document does not introduce any new security considerations beyond that already apply to [RFC6425].

9. IANA Considerations

This document defines a new sub-TLV type to be included in Target FEC Stack TLV (TLV Type 1) [RFC4379] in LSP Ping.

IANA is requested to assign a sub-TLV type value to the following sub-TLV from the "Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Parameters - TLVs" registry, "TLVs and sub- TLVs" sub-registry:

- P2MP Pseudowire sub-TLV

10. Acknowledgments

The authors would like to thank Shaleen Saxena, Michael Wildt, Tomofumi Hayashi, Danny Prairie for their valuable input and comments.

11. References

11.1. Normative References

[I-D.ietf-pwe3-p2mp-pw]
Sivabalan, S., Boutros, S., and L. Martini, "Signaling Root-Initiated Point-to-Multipoint Pseudowire using LDP", draft-ietf-pwe3-p2mp-pw-04 (work in progress), March 2012.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, DOI 10.17487/RFC4379, February 2006, <http://www.rfc-editor.org/info/rfc4379>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <http://www.rfc-editor.org/info/rfc4446>.

[RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <http://www.rfc-editor.org/info/rfc6425>.

[RFC6426] Gray, E., Bahadur, N., Boutros, S., and R. Aggarwal, "MPLS On-Demand Connectivity Verification and Route Tracing", RFC 6426, DOI 10.17487/RFC6426, November 2011, <http://www.rfc-editor.org/info/rfc6426>.

[RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <http://www.rfc-editor.org/info/rfc7117>.

11.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <http://www.rfc-editor.org/info/rfc4875>.

[RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <http://www.rfc-editor.org/info/rfc5085>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <http://www.rfc-editor.org/info/rfc6388>.

[RFC7338] Jounay, F., Ed., Kamite, Y., Ed., Heron, G., and M. Bocci, "Requirements and Framework for Point-to-Multipoint Pseudowires over MPLS Packet Switched Networks", RFC 7338, DOI 10.17487/RFC7338, September 2014, <http://www.rfc-editor.org/info/rfc7338>.

Authors' Addresses

Parag Jain
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K-3E8
Canada

Email: paragj@cisco.com

Sami Boutros
VMWare, Inc.
USA

Email: sboutros@vmware.com

Sam Aldrin
Google Inc.
USA

Email: aldrin.ietf@gmail.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2016

H. Shah
Ciena Corporation
P. Brissette
R. Rahman
K. Raza
Cisco Systems, Inc.
Z. Li
Z. Shunwan
W. Haibo
Huawei Technologies
I. Chen
S. Ahmed
Ericsson
M. Bocci
Alcatel-Lucent
J. Hardwick
Metaswitch
S. Esale
K. Tiruveedhula
T. Singh
Juniper Networks
I. Hussain
Infinera Corporation
B. Wen
J. Walker
Comcast
N. Delregno
L. Jalil
M. Joecylyn
Verizon

March 14, 2016

YANG Data Model for MPLS-based L2VPN
draft-shah-bess-l2vpn-yang-01.txt

Abstract

This document describes a YANG data model for Layer 2 VPN services over MPLS networks. These services include Virtual Private Wire Service (VPWS) and Virtual Private LAN service (VPLS) that uses LDP and BGP signaled Pseudowires. The current version of the document expands the L2VPN object model to include VPLS services in addition to the VPWS services described in the last revision. This is a living document and contains aspects of object models that have been discussed extensively in the working group with consensus. The intention is to continue to seek input from larger audience during evolution of the L2VPN service model through this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is a network management protocol that defines mechanisms to manage network devices. YANG [RFC6020] is a modular language that represents data structures in an XML or JSON tree format, and is used as a data modeling language for the NETCONF.

This document introduces a YANG data model for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] as well as switching between the local attachment circuits. The L2VPN services include point-to-point VPWS and Multipoint VPLS services. These services are realized by signaling Pseudowires across MPLS networks using LDP [RFC4447][RFC4762] or BGP[RFC4761].

The YANG data model in this document defines Ethernet based Layer 2 services. Other Layer 2 services, such as ATM, Frame Relay, TDM, etc are included in the scope but will be covered as the future work items. The Ethernet based Layer 2 services will leverage the definitions used in other standards organizations such as IEEE 802.1 and Metro Ethernet Forum (MEF).

The goal is to propose a data object model consisting of building blocks that can be assembled in different order to realize different services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The VPWS service definitions were covered first in the last revision of the document. The current version documents VPLS services that build on the data blocks defined for VPWS.

In the current version of this document, refinements to the configuration objects and Operational State objects for the same are added.

The data model is defined for following constructs that are used for managing the services:

- Configuration
- Operational State
- Executables (Actions)
- Notifications

The document is organized to first define the data model for the configuration of all the L2VPN services followed by definition of operational state, actions and notifications for the same. The L2VPN data object model defined in this document uses the instance centric approach. The attributes of each service, VPWS, VPLS, etc are specified for a given service instance.

2. Updates in this revision

The organization of the configuration objects has been updated. The ac-templates in the common container is removed and a new redundancy-group-templates is added.

The vpls-instances container is removed and replaced with bridge-table-instances container to include the PBB, BGP parameters. This revision also introduces a reference to EVPN instance. This revision removes the definition of Attachment Circuits, "ac". Instead, the L2VPN data object model will rely on standard definitions of Attachment Circuits that IEEE and IETF are coordinating to define. Thus, this revision uses a string as a placeholder for an Attachment Circuit within the ac-or-pw-redundancy-grp within the respective endpoints list of bridge-table-instances or VPWS instances, and expect to update this field once the standard definitions of Attachment Circuits are available.

3. Specification of Requirements

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 [RFC2119].

4. L2VPN YANG Model

4.1. Overview

One single top level container, l2vpn, is defined as a parent for three different second level containers that are vpws-instances, bridge-table-instances, and common building blocks of redundancy-grp templates and pseudowire-templates. The current version of the document is extended to include refinements to configuration of vpws-instance and bridge-table-instances. The operations state object has been added to hold read-only information of objects that has either been configured or dynamically created.

The L2VPN services have been defined in the IETF L2VPN working group but leverages the pseudowire technologies that were defined in the PWE3 working group. A large number of RFCs from these working groups cover this subject matter. Hence, it is prudent that this document state the scope of the MPLS L2VPN object model definitions.

The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

- Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC4447]
- Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]
- Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- Segmented Pseudowire [RFC6073]
- Framework for Layer 2 Virtual Private Networks [RFC4664]

- Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]
- Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]
- Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- Pseudowire status for static pseudowire [RFC6478]

Note that while pseudowire over MPLS-TP related work is in scope, the initial effort will only address definitions of object models for services that are commonly deployed.

The ietf work in L2VPN and PWE3 working group relating to L2TP, OAM, multicast (e.g. p2mp, etree, etc) and access specific protocols such as G.8032, MSTP, etc is out-of-scope for this document.

The following is the high level view of the L2VPN data model.

```
 template-ref PW // PW
                    template
                    attributes

 template-ref Redundancy-Group // redundancy-group
```

```
                    template
                    attributes


 bridge-table-instance name // container

         common attributes

         PBB-parameters // container
                          pbb specific attributes

         BGP-parameters // container
                          common attributes
                          auto-discovery attributes
                          signaling attributes

         evpn-instance // reference

         // list of PWs being used
         PW // container
                 template-ref PW
                 attribute-override

         // List of endpoints, where each member endpoint container is -
                 PW // reference
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference

 vpws-instance name // container

         common attributes

         BGP-parameters // container
                          common attributes
                          auto-discovery attributes
                          signaling attributes

         // list of PWs being used
         PW // container
                 template-ref PW
                 attribute-override
                 pw type
                    static-or-ldp
                    bgp-pw
                    bgp-ad-pw

         // ONLY 2 endpoints!!!
```

```
         endpoint-A // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference


         endpoint-Z // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference

 l2vpn-state // read-only container
```

Figure 1

4.2. L2VPN Common

4.2.1. ac-templates

The ac-templates container is removed. The AC will eventually reference standard AC definitions defined with coordination between the IEEE and IETF, and will inherit all the attributes defined in that reference.

4.2.2. pw-templates

The pw-templates container contains a list of pw-template. Each pw-template defines a list of common pseudowire attributes such as PW MTU, control word support etc.

4.3. VPWS and Bridge-Table-Instance (formerly referred as VPLS)

4.3.1. ac list

AC resides within endpoint container as member of ac-or-pw-or-redundandancy-grp.

4.3.2. pw list

Each VPWS and Bridge-Table-Instance defines a list of PWs which are participating members of the given service instance. Each entry of the PW consists of one pw-template with pre-defined attributes and values, but also defines attributes that override those defined in referenced pw-template.

No restrictions are placed on type of signaling (i.e. LDP or BGP) used for a given PW. It is entirely possible to define two PWs, one signaled by LDP and other by BGP.

The VPLS specific attribute(s) are present in the definition of the PW that are member of VPLS instance only and not applicable to VPWS service.

4.3.3. redundancy-grp choice

The redundancy-grp is a generic redundancy construct which can hold primary and backup members of AC and PWs. This flexibility permits combinations of -

- primary and backup AC
- primary and backup PW
- primary AC and backup PW
- primary PW and backup AC

4.3.4. endpoint container

The endpoint container in general holds AC, PW or redundancy-grp references. The core aspect of endpoint container is its flexible personality based on what user decides to include in it. It is future-proofed with possible extensions that can be included in the endpoint container such as Integrated Route Bridging (IRB), PW Headend, Virtual Switch Instance, etc.

The endpoint container for the VPLS service holds references to a list of ACs, a list of PWs or a redundancy group that contains a list of ACs and/or a list of PWs. This differs from the VPWS instance where an endpoint contains exactly one member; AC or PW or redundancy group and not a list.

4.3.5. vpws-instances and bridge-table-instances container

The vpws-instances container contains a list of vpws-instance. Each entry of the vpws-instance represents a layer-2 cross-connection of two endpoints. This model defines three possible types of endpoints, ac, pw, and redundancy-grp, and allows a vpws-instance to cross-connect any one type of endpoint to all other types of endpoint.

The bridge-table-instances container contains a list of bridge-table-instance. Each entry of the bridge-table-instance represent a list of endpoints that are member of the broadcast/bridge domain. The

bridge-table-instance endpoints introduces an additional forwarding characteristics to a list of PWs and/or ACs. This split-horizon forwarding behavior is typical in bridge-table instance.

The augmentation of ietf-l2vpn module is TBD. All IP addresses defined in this module are currently scoped under global VRF/table.

4.4. Operational State

The operational state of L2VPN can be queried and obtained from the read-only container defined in this document as "l2vpn-state". This container holds the runtime information of the bridge-table-instance and vpws-instance.

4.5. Open items

The design team has identified several attributes that need to be included in the YANG tree. These attributes are listed here for the purpose of keeping track and are candidates for future revisions.

These attributes are :

- configuration - vccv configuration knobs
- operational state - vccv and cw-negotiation

This list is not exhaustive and expected to grow. The list will shrink as items are processed and included in the YANG tree.

4.6. Yang tree

```
module: ietf-l2vpn
   +--rw l2vpn
   |  +--rw common
   |  |  +--rw pw-templates
   |  |  |  +--rw pw-template* [name]
   |  |  |     +--rw name              string
   |  |  |     +--rw mtu?              uint32
   |  |  |     +--rw cw-negotiation?   cw-negotiation-type
   |  |  |     +--rw tunnel-policy?    string
   |  |  +--rw redundancy-group-templates
   |  |     +--rw redundancy-group-template* [name]
   |  |        +--rw name               string
   |  |        +--rw protection-mode?   enumeration
   |  |        +--rw reroute-mode?      enumeration
   |  |        +--rw reroute-delay?     uint16
   |  |        +--rw dual-receive?      boolean
   |  |        +--rw revert?            boolean
```

```
   |  |        +--rw revert-delay?      uint16
   |  +--rw bridge-table-instances
   |  |  +--rw bridge-table-instance* [name]
   |  |     +--rw name               string
   |  |     +--rw mtu?               uint32
   |  |     +--rw mac-aging-timer?   uint32
   |  |     +--rw pbb-parameters
   |  |     |  +--rw (component-type)?
   |  |     |     +--:(i-component)
   |  |     |     |  +--rw i-tag?              uint32
   |  |     |     |  +--rw backbone-src-mac?   yang:mac-address
   |  |     |     +--:(b-component)
   |  |     |        +--rw bind-b-component?   bridge-table-instance-ref
   |  |     +--rw bgp-parameters
   |  |     |  +--rw common
   |  |     |  |  +--rw route-distinguisher?   string
   |  |     |  |  +--rw vpn-targets* [rt-value]
   |  |     |  |     +--rw rt-value    string
   |  |     |  |     +--rw rt-type     bgp-rt-type
   |  |     |  +--rw discovery
   |  |     |  |  +--rw vpn-id?   string
   |  |     |  +--rw signaling
   |  |     |     +--rw site-id?      uint16
   |  |     |     +--rw site-range?   uint16
   |  |     +--rw evpn-instance?     string
   |  |     +--rw pw* [name]
   |  |     |  +--rw name              string
   |  |     |  +--rw template?         pw-template-ref
   |  |     |  +--rw mtu?              uint32
   |  |     |  +--rw mac-withdraw?     boolean
   |  |     |  +--rw cw-negotiation?   cw-negotiation-type
   |  |     |  +--rw discovery-type?   l2vpn-discovery-type
   |  |     |  +--rw signaling-type?   l2vpn-signaling-type
   |  |     |  +--rw peer-ip?          inet:ip-address
   |  |     |  +--rw pw-id?            uint32
   |  |     |  +--rw transmit-label?   uint32
   |  |     |  +--rw receive-label?    uint32
   |  |     |  +--rw tunnel-policy?    string
   |  |     +--rw endpoint* [id]
   |  |        +--rw id                       uint16
   |  |        +--rw split-horizon-group?   string
   |  |        +--rw (ac-or-pw-or-redundancy-grp)?
   |  |           +--:(ac)
   |  |           |  +--rw ac* [name]
   |  |           |     +--rw name    string
   |  |           +--:(pw)
   |  |           |  +--rw pw* [name]
   |  |           |     +--rw name    -> ../../../pw/name
```

```
    |   |            +--:(redundancy-grp)
    |   |               +--rw (primary)
    |   |               |  +--:(primary-pw)
    |   |               |  |  +--rw primary-pw* [name]
    |   |               |  |     +--rw name    -> ../../../pw/name
    |   |               |  +--:(primary-ac)
    |   |               |     +--rw primary-ac?            string
    |   |               +--rw (backup)?
    |   |               |  +--:(backup-pw)
    |   |               |  |  +--rw backup-pw* [name]
    |   |               |  |     +--rw name          -> ../../../pw/name
    |   |               |  |     +--rw precedence?   uint32
    |   |               |  +--:(backup-ac)
    |   |               |     +--rw backup-ac?             string
    |   |               +--rw template?               -> /l2vpn/common/redundancy-gr
oup-templates/redundancy-group-template/name
    |   |               +--rw protection-mode?       enumeration
    |   |               +--rw reroute-mode?          enumeration
    |   |               +--rw reroute-delay?         uint16
    |   |               +--rw dual-receive?          boolean
    |   |               +--rw revert?                boolean
    |   |               +--rw revert-delay?          uint16
    |   +--rw vpws-instances
    |      +--rw vpws-instance* [name]
    |         +--rw name               string
    |         +--rw description?       string
    |         +--rw mtu?               uint32
    |         +--rw mac-aging-timer?   uint32
    |         +--rw service-type?      l2vpn-service-type
    |         +--rw discovery-type?    l2vpn-discovery-type
    |         +--rw signaling-type     l2vpn-signaling-type
    |         +--rw bgp-parameters
    |         |  +--rw common
    |         |  |  +--rw route-distinguisher?   string
    |         |  |  +--rw vpn-targets* [rt-value]
    |         |  |     +--rw rt-value    string
    |         |  |     +--rw rt-type     bgp-rt-type
    |         |  +--rw discovery
    |         |  |  +--rw vpn-id?   string
    |         |  +--rw signaling
    |         |     +--rw site-id?      uint16
    |         |     +--rw site-range?   uint16
    |         +--rw pw* [name]
    |         |  +--rw name              string
    |         |  +--rw template?         pw-template-ref
    |         |  +--rw mtu?              uint32
    |         |  +--rw mac-withdraw?     boolean
    |         |  +--rw cw-negotiation?   cw-negotiation-type
    |         |  +--rw vccv-ability?     boolean
```

```
   |          |  +--rw tunnel-policy?    string
   |          |  +--rw request-vlanid?   uint16
   |          |  +--rw vlan-tpid?        string
   |          |  +--rw ttl?              uint8
   |          |  +--rw (pw-type)?
   |          |     +--:(ldp-or-static-pw)
   |          |     |  +--rw peer-ip?          inet:ip-address
   |          |     |  +--rw pw-id?            uint32
   |          |     |  +--rw icb?              boolean
   |          |     |  +--rw transmit-label?   uint32
   |          |     |  +--rw receive-label?    uint32
   |          |     +--:(bgp-pw)
   |          |     |  +--rw remote-pe-id?     inet:ip-address
   |          |     +--:(bgp-ad-pw)
   |          |        +--rw remote-ve-id?     uint16
   |          +--rw endpoint-a
   |          |  +--rw (ac-or-pw-or-redundancy-grp)?
   |          |     +--:(ac)
   |          |     |  +--rw ac?                string
   |          |     +--:(pw)
   |          |     |  +--rw pw?                -> ../../pw/name
   |          |     +--:(redundancy-grp)
   |          |        +--rw (primary)
   |          |        |  +--:(primary-pw)
   |          |        |  |  +--rw primary-pw?        -> ../../pw/name
   |          |        |  +--:(primary-ac)
   |          |        |     +--rw primary-ac?        string
   |          |        +--rw (backup)
   |          |        |  +--:(backup-pw)
   |          |        |  |  +--rw backup-pw?         -> ../../pw/name
   |          |        |  +--:(backup-ac)
   |          |        |     +--rw backup-ac?         string
   |          |        +--rw template?          -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |          |        +--rw protection-mode?   enumeration
   |          |        +--rw reroute-mode?      enumeration
   |          |        +--rw reroute-delay?     uint16
   |          |        +--rw dual-receive?      boolean
   |          |        +--rw revert?            boolean
   |          |        +--rw revert-delay?      uint16
   |          +--rw endpoint-z
   |             +--rw (ac-or-pw-or-redundancy-grp)?
   |                +--:(ac)
   |                |  +--rw ac?                string
   |                +--:(pw)
   |                |  +--rw pw?                -> ../../pw/name
   |                +--:(redundancy-grp)
   |                   +--rw (primary)
   |                      +--:(primary-pw)
```

```
   |                  |  |  +--rw primary-pw?        -> ../../pw/name
   |                  |  +--:(primary-ac)
   |                  |     +--rw primary-ac?        string
   |                  +--rw (backup)
   |                  |  +--:(backup-pw)
   |                  |  |  +--rw backup-pw?         -> ../../pw/name
   |                  |  +--:(backup-ac)
   |                  |     +--rw backup-ac?         string
   |                  +--rw template?           -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |                  +--rw protection-mode?    enumeration
   |                  +--rw reroute-mode?       enumeration
   |                  +--rw reroute-delay?      uint16
   |                  +--rw dual-receive?       boolean
   |                  +--rw revert?             boolean
   |                  +--rw revert-delay?       uint16
   +--ro l2vpn-state
      +--ro bridge-table-instances-state
      |  +--ro bridge-table-instance-state* [name]
      |     +--ro name                  string
      |     +--ro mtu?                  uint32
      |     +--ro mac-aging-timer?      uint32
      |     +--ro pbb-parameters
      |     |  +--ro (component-type)?
      |     |     +--:(i-component)
      |     |     |  +--ro i-tag?              uint32
      |     |     |  +--ro backbone-src-mac?   yang:mac-address
      |     |     +--:(b-component)
      |     |        +--ro bind-b-component?   string
      |     +--ro bgp-parameters
      |     |  +--ro common
      |     |  |  +--ro route-distinguisher?   string
      |     |  |  +--ro vpn-targets* [rt-value]
      |     |  |     +--ro rt-value    string
      |     |  |     +--ro rt-type     bgp-rt-type
      |     |  +--ro discovery
      |     |  |  +--ro vpn-id?   string
      |     |  +--ro signaling
      |     |     +--ro site-id?      uint16
      |     |     +--ro site-range?   uint16
      |     +--ro evpn-instance-name?   string
      |     +--ro endpoint* [id]
      |        +--ro id                     uint16
      |        +--ro split-horizon-group?   string
      |        +--ro (ac-or-pw-or-redundancy-grp)?
      |           +--:(ac)
      |              |  +--ro ac* [name]
      |              |     +--ro name     string
      |              |     +--ro state?   operational-state-type
```

```
|            +--:(pw)
|            |  +--ro pw* [name]
|            |     +--ro name              string
|            |     +--ro state?            operational-state-type
|            |     +--ro mtu?              uint32
|            |     +--ro mac-withdraw?     boolean
|            |     +--ro cw-negotiation?   cw-negotiation-type
|            |     +--ro discovery-type?   l2vpn-discovery-type
|            |     +--ro signaling-type?   l2vpn-signaling-type
|            |     +--ro peer-ip?          inet:ip-address
|            |     +--ro pw-id?            uint32
|            |     +--ro transmit-label?   uint32
|            |     +--ro receive-label?    uint32
|            |     +--ro tunnel-policy?    string
|            +--:(redundancy-grp)
|               +--ro (primary)
|               |  +--:(primary-pw)
|               |  |  +--ro primary-pw* [name]
|               |  |     +--ro name              string
|               |  |     +--ro state?            operational-state-type
|               |  |     +--ro mtu?              uint32
|               |  |     +--ro mac-withdraw?     boolean
|               |  |     +--ro cw-negotiation?   cw-negotiation-type
|               |  |     +--ro discovery-type?   l2vpn-discovery-type
|               |  |     +--ro signaling-type?   l2vpn-signaling-type
|               |  |     +--ro peer-ip?          inet:ip-address
|               |  |     +--ro pw-id?            uint32
|               |  |     +--ro transmit-label?   uint32
|               |  |     +--ro receive-label?    uint32
|               |  |     +--ro tunnel-policy?    string
|               |  +--:(primary-ac)
|               |     +--ro primary-ac
|               |        +--ro name?    string
|               |        +--ro state?   operational-state-type
|               +--ro (backup)?
|               |  +--:(backup-pw)
|               |  |  +--ro backup-pw* [name]
|               |  |     +--ro name              string
|               |  |     +--ro state?            operational-state-type
|               |  |     +--ro mtu?              uint32
|               |  |     +--ro mac-withdraw?     boolean
|               |  |     +--ro cw-negotiation?   cw-negotiation-type
|               |  |     +--ro discovery-type?   l2vpn-discovery-type
|               |  |     +--ro signaling-type?   l2vpn-signaling-type
|               |  |     +--ro peer-ip?          inet:ip-address
|               |  |     +--ro pw-id?            uint32
|               |  |     +--ro transmit-label?   uint32
|               |  |     +--ro receive-label?    uint32
```

```
|              |  |     +--ro tunnel-policy?    string
|              |  |     +--ro precedence?       uint32
|              |  +--:(backup-ac)
|              |     +--ro backup-ac
|              |        +--ro name?    string
|              |        +--ro state?   operational-state-type
|              +--ro protection-mode?       enumeration
|              +--ro reroute-mode?          enumeration
|              +--ro reroute-delay?         uint16
|              +--ro dual-receive?          boolean
|              +--ro revert?                boolean
|              +--ro revert-delay?          uint16
|
+--ro vpws-instances-state
   +--ro vpws-instance-state* [name]
      +--ro name               string
      +--ro mtu?               uint32
      +--ro mac-aging-timer?   uint32
      +--ro service-type?      l2vpn-service-type
      +--ro discovery-type?    l2vpn-discovery-type
      +--ro signaling-type     l2vpn-signaling-type
      +--ro bgp-parameters
      |  +--ro common
      |  |  +--ro route-distinguisher?   string
      |  |  +--ro vpn-targets* [rt-value]
      |  |     +--ro rt-value    string
      |  |     +--ro rt-type     bgp-rt-type
      |  +--ro discovery
      |  |  +--ro vpn-id?   string
      |  +--ro signaling
      |     +--ro site-id?      uint16
      |     +--ro site-range?   uint16
      +--ro endpoint-a
      |  +--ro (ac-or-pw-or-redundancy-grp)?
      |     +--:(ac)
      |     |  +--ro ac
      |     |     +--ro name?    string
      |     |     +--ro state?   operational-state-type
      |     +--:(pw)
      |     |  +--ro pw
      |     |     +--ro name?             string
      |     |     +--ro state?            operational-state-type
      |     |     +--ro mtu?              uint32
      |     |     +--ro mac-withdraw?     boolean
      |     |     +--ro cw-negotiation?   cw-negotiation-type
      |     |     +--ro vccv-ability?     boolean
      |     |     +--ro tunnel-policy?    string
      |     |     +--ro request-vlanid?   uint16
      |     |     +--ro vlan-tpid?        string
```

```
|       |      +--ro ttl?              uint8
|       |      +--ro (pw-type)?
|       |         +--:(ldp-or-static-pw)
|       |         |  +--ro peer-ip?          inet:ip-address
|       |         |  +--ro pw-id?            uint32
|       |         |  +--ro icb?              boolean
|       |         |  +--ro transmit-label?   uint32
|       |         |  +--ro receive-label?    uint32
|       |         +--:(bgp-pw)
|       |         |  +--ro remote-pe-id?     inet:ip-address
|       |         +--:(bgp-ad-pw)
|       |            +--ro remote-ve-id?     uint16
|       +--:(redundancy-grp)
|          +--ro (primary)
|          |  +--:(primary-pw)
|          |  |  +--ro primary-pw
|          |  |     +--ro name?             string
|          |  |     +--ro state?            operational-state-type
|          |  |     +--ro mtu?              uint32
|          |  |     +--ro mac-withdraw?     boolean
|          |  |     +--ro cw-negotiation?   cw-negotiation-type
|          |  |     +--ro vccv-ability?     boolean
|          |  |     +--ro tunnel-policy?    string
|          |  |     +--ro request-vlanid?   uint16
|          |  |     +--ro vlan-tpid?        string
|          |  |     +--ro ttl?              uint8
|          |  |     +--ro (pw-type)?
|          |  |        +--:(ldp-or-static-pw)
|          |  |        |  +--ro peer-ip?          inet:ip-address
|          |  |        |  +--ro pw-id?            uint32
|          |  |        |  +--ro icb?              boolean
|          |  |        |  +--ro transmit-label?   uint32
|          |  |        |  +--ro receive-label?    uint32
|          |  |        +--:(bgp-pw)
|          |  |        |  +--ro remote-pe-id?     inet:ip-address
|          |  |        +--:(bgp-ad-pw)
|          |  |           +--ro remote-ve-id?     uint16
|          |  +--:(primary-ac)
|          |     +--ro primary-ac-name?   string
|          +--ro (backup)
|          |  +--:(backup-pw)
|          |  |  +--ro backup-pw
|          |  |     +--ro name?             string
|          |  |     +--ro state?            operational-state-type
|          |  |     +--ro mtu?              uint32
|          |  |     +--ro mac-withdraw?     boolean
|          |  |     +--ro cw-negotiation?   cw-negotiation-type
|          |  |     +--ro vccv-ability?     boolean
```

```
|          |  |     +--ro tunnel-policy?    string
|          |  |     +--ro request-vlanid?   uint16
|          |  |     +--ro vlan-tpid?        string
|          |  |     +--ro ttl?              uint8
|          |  |     +--ro (pw-type)?
|          |  |        +--:(ldp-or-static-pw)
|          |  |        |  +--ro peer-ip?          inet:ip-address
|          |  |        |  +--ro pw-id?            uint32
|          |  |        |  +--ro icb?              boolean
|          |  |        |  +--ro transmit-label?   uint32
|          |  |        |  +--ro receive-label?    uint32
|          |  |        +--:(bgp-pw)
|          |  |        |  +--ro remote-pe-id?     inet:ip-address
|          |  |        +--:(bgp-ad-pw)
|          |  |           +--ro remote-ve-id?     uint16
|          |  +--:(backup-ac)
|          |     +--ro backup-ac-name?    string
|          +--ro protection-mode?   enumeration
|          +--ro reroute-mode?      enumeration
|          +--ro reroute-delay?     uint16
|          +--ro dual-receive?      boolean
|          +--ro revert?            boolean
|          +--ro revert-delay?      uint16
+--ro endpoint-z
   +--ro (ac-or-pw-or-redundancy-grp)?
      +--:(ac)
      |  +--ro ac
      |     +--ro name?    string
      |     +--ro state?   operational-state-type
      +--:(pw)
      |  +--ro pw
      |     +--ro name?             string
      |     +--ro state?            operational-state-type
      |     +--ro mtu?              uint32
      |     +--ro mac-withdraw?     boolean
      |     +--ro cw-negotiation?   cw-negotiation-type
      |     +--ro vccv-ability?     boolean
      |     +--ro tunnel-policy?    string
      |     +--ro request-vlanid?   uint16
      |     +--ro vlan-tpid?        string
      |     +--ro ttl?              uint8
      |     +--ro (pw-type)?
      |        +--:(ldp-or-static-pw)
      |        |  +--ro peer-ip?          inet:ip-address
      |        |  +--ro pw-id?            uint32
      |        |  +--ro icb?              boolean
      |        |  +--ro transmit-label?   uint32
      |        |  +--ro receive-label?    uint32
```

```
|        +--:(bgp-pw)
|        |  +--ro remote-pe-id?     inet:ip-address
|        +--:(bgp-ad-pw)
|           +--ro remote-ve-id?     uint16
+--:(redundancy-grp)
   +--ro (primary)
   |  +--:(primary-pw)
   |  |  +--ro primary-pw
   |  |     +--ro name?             string
   |  |     +--ro state?            operational-state-type
   |  |     +--ro mtu?              uint32
   |  |     +--ro mac-withdraw?     boolean
   |  |     +--ro cw-negotiation?   cw-negotiation-type
   |  |     +--ro vccv-ability?     boolean
   |  |     +--ro tunnel-policy?    string
   |  |     +--ro request-vlanid?   uint16
   |  |     +--ro vlan-tpid?        string
   |  |     +--ro ttl?              uint8
   |  |     +--ro (pw-type)?
   |  |        +--:(ldp-or-static-pw)
   |  |        |  +--ro peer-ip?          inet:ip-address
   |  |        |  +--ro pw-id?            uint32
   |  |        |  +--ro icb?              boolean
   |  |        |  +--ro transmit-label?   uint32
   |  |        |  +--ro receive-label?    uint32
   |  |        +--:(bgp-pw)
   |  |        |  +--ro remote-pe-id?     inet:ip-address
   |  |        +--:(bgp-ad-pw)
   |  |           +--ro remote-ve-id?     uint16
   |  +--:(primary-ac)
   |     +--ro primary-ac-name?   string
   +--ro (backup)
   |  +--:(backup-pw)
   |  |  +--ro backup-pw
   |  |     +--ro name?             string
   |  |     +--ro state?            operational-state-type
   |  |     +--ro mtu?              uint32
   |  |     +--ro mac-withdraw?     boolean
   |  |     +--ro cw-negotiation?   cw-negotiation-type
   |  |     +--ro vccv-ability?     boolean
   |  |     +--ro tunnel-policy?    string
   |  |     +--ro request-vlanid?   uint16
   |  |     +--ro vlan-tpid?        string
   |  |     +--ro ttl?              uint8
   |  |     +--ro (pw-type)?
   |  |        +--:(ldp-or-static-pw)
   |  |        |  +--ro peer-ip?          inet:ip-address
   |  |        |  +--ro pw-id?            uint32
```

```
                    |  |         |  +--ro icb?               boolean
                    |  |         |  +--ro transmit-label?    uint32
                    |  |         |  +--ro receive-label?     uint32
                    |  |         +--:(bgp-pw)
                    |  |         |  +--ro remote-pe-id?      inet:ip-address
                    |  |         +--:(bgp-ad-pw)
                    |  |            +--ro remote-ve-id?      uint16
                    |  +--:(backup-ac)
                    |     +--ro backup-ac-name?    string
                    +--ro protection-mode?   enumeration
                    +--ro reroute-mode?      enumeration
                    +--ro reroute-delay?     uint16
                    +--ro dual-receive?      boolean
                    +--ro revert?            boolean
                    +--ro revert-delay?      uint16
```

Figure 2

5. YANG Module

The L2VPN configuration container is logically divided into following high level config areas:

```
<CODE BEGINS> file "ietf-l2vpn@2016-03-07.yang"
module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";

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

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

  organization  "ietf";
  contact       "ietf";
  description   "l2vpn";

  revision "2016-03-07" {
    description "Third revision " +
                "  - Changed the module name to ietf-l2vpn " +
                "  - Merged EVPN into L2VPN " +
                "  - Eliminated the definitions of attachment " +
                "    circuit with the intention to reuse other " +
                "    layer-2 definitions " +
```

```
                "  - Added state branch";
    reference "";
  }

  revision "2015-10-08" {
    description "Second revision " +
                "  - Added container vpls-instances " +
                "  - Rearranged groupings and typedefs to be " +
                "    reused across vpls-instance and vpws-instances";
    reference "";
  }

  revision "2015-06-30" {
    description "Initial revision";
    reference   "";
  }

  /* identities */

  identity link-discovery-protocol {
    description "Base identiy from which identities describing " +
                "link discovery protocols are derived.";
  }

  identity lacp {
    base "link-discovery-protocol";
    description "This identity represents LACP";
  }

  identity lldp {
    base "link-discovery-protocol";
    description "This identity represents LLDP";
  }

  identity bpdu {
    base "link-discovery-protocol";
    description "This identity represens BPDU";
  }

  identity cpd {
    base "link-discovery-protocol";
    description "This identity represents CPD";
  }

  identity udld {
    base "link-discovery-protocol";
    description "This identity represens UDLD";
  }
```

```
  /* typedefs */

  typedef l2vpn-service-type {
    type enumeration {
      enum ethernet {
        description "Ethernet service";
      }
      enum ATM {
        description "Asynchronous Transfer Mode";
      }
      enum FR {
        description "Frame-Relay";
      }
      enum TDM {
        description "Time Division Multiplexing";
      }
    }
    description "L2VPN service type";
  }

  typedef l2vpn-discovery-type {
    type enumeration {
      enum manual {
        description "Manual configuration";
      }
      enum bgp-ad {
        description "Border Gateway Protocol (BGP) auto-discovery";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP)";
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN discovery type";
  }

  typedef l2vpn-signaling-type {
    type enumeration {
      enum static {
        description "Static configuration of labels (no signaling)";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP) signaling";
      }
      enum bgp {
        description "Border Gateway Protocol (BGP) signaling";
```

```
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN signaling type";
  }

  typedef bgp-rt-type {
    type enumeration {
      enum import {
        description "For import";
      }
      enum export {
        description "For export";
      }
      enum both {
        description "For both import and export";
      }
    }
    description "BGP route-target type. Import from BGP YANG";
  }

  typedef cw-negotiation-type {
    type enumeration {
      enum "non-preferred" {
        description "No preference for control-word";
      }
      enum "preferred" {
        description "Prefer to have control-word negotiation";
      }
    }
    description "control-word negotiation preference type";
  }

  typedef link-discovery-protocol-type {
    type identityref {
      base "link-discovery-protocol";
    }
    description "This type is used to identify " +
                "link discovery protocol";
  }

  typedef pbb-component-type {
    type enumeration {
      enum "b-component" {
        description "Identifies as a b-component";
      }
```

```
      enum "i-component" {
        description "Identifies as an i-component";
      }
    }
    description "This type is used to identify " +
                "the type of PBB component";
  }

  typedef pw-template-ref {
    type leafref {
      path "/l2vpn/common/pw-templates/pw-template/name";
    }
    description "pw-template-ref";
  }

  typedef redundancy-group-template-ref {
    type leafref {
      path "/l2vpn/common/redundancy-group-templates" +
           "/redundancy-group-template/name";
    }
    description "redundancy-group-template-ref";
  }

  typedef bridge-table-instance-ref {
    type leafref {
      path "/l2vpn/bridge-table-instances" +
           "/bridge-table-instance/name";
    }
    description "bridge-table-instance-ref";
  }

  typedef operational-state-type {
    type enumeration {
      enum 'up' {
        description "Operational state is up";
      }
      enum 'down' {
        description "Operational state is down";
      }
    }
    description "operational-state-type";
  }

  /* groupings */

  grouping pbb-parameters-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
```

```
      description "pbb-parameters";
      choice component-type {
        description "PBB component type";
        case i-component {
          leaf i-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type bridge-table-instance-ref;
            description "Reference to the associated b-component";
          }
        }
      }
    }
  }

  grouping pbb-parameters-state-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
      description "pbb-parameters";
      choice component-type {
        description "PBB component type";
        case i-component {
          leaf i-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type string;
            description "Name of the associated b-component";
          }
        }
      }
    }
  }
```

```
  grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
      description "Parameters for BGP";
      container common {
        when "../../discovery-type = 'bgp-ad'" {
          description "Check discovery type: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "Common BGP parameters";
        leaf route-distinguisher {
          type string;
          description "BGP RD";
        }
        list vpn-targets {
          key rt-value;
          description "Route Targets";
          leaf rt-value {
            type string;
            description "Route-Target value";
          }
          leaf rt-type {
            type bgp-rt-type;
            mandatory true;
            description "Type of RT";
          }
        }
      }
      container discovery {
        when "../../discovery-type = 'bgp-ad'" {
          description "BGP parameters for discovery: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "BGP parameters for discovery";
        leaf vpn-id {
          type string;
          description "VPN ID";
        }
      }
      container signaling {
        when "../../signaling-type = 'bgp'" {
          description "Check signaling type: " +
                      "Can only configure BGP signaling if " +
                      "signaling type is BGP";
        }
        description "BGP parameters for signaling";
```

```
        leaf site-id {
          type uint16;
          description "Site ID";
        }
        leaf site-range {
          type uint16;
          description "Site Range";
        }
      }
    }
  }

  grouping pw-type-grp {
    description "pseudowire type grouping";
    choice pw-type {
      description "A choice of pseudowire type";
      case ldp-or-static-pw {
        leaf peer-ip {
          type inet:ip-address;
          description "peer IP address";
        }
        leaf pw-id {
          type uint32;
          description "pseudowire id";
        }
        leaf icb {
          type boolean;
          description "inter-chassis backup";
        }
        leaf transmit-label {
          type uint32;
          description "transmit lable";
        }
        leaf receive-label {
          type uint32;
          description "receive label";
        }
      }
      case bgp-pw {
        leaf remote-pe-id {
          type inet:ip-address;
          description "remote pe id";
        }
      }
      case bgp-ad-pw {
        leaf remote-ve-id {
          type uint16;
          description "remote ve id";
```

```
        }
      }
    }
  }

  grouping bridge-table-instance-pw-list-grp {
    description "bridge-table-instance-pw-list-grp";
    list pw {
      key "name";
      leaf name {
        type leafref {
          path "../../../pw/name";
        }
        description "name of pseudowire";
      }
      description "A bridge table instance's pseudowire list";
    }
  }

  grouping bridge-table-instance-ac-list-grp {
    description "bridge-table-instance-ac-list-grp";
    list ac {
      key "name";
      leaf name {
        type string;
        description "Name of attachment circuit.  This field " +
                    "is intended to reference standardized " +
                    "layer-2 definitions.";
      }
      description "A bridge table instance's " +
                  "attachment circuit list";
    }
  }

  grouping redundancy-group-properties-grp {
    description "redundancy-group-properties-grp";
    leaf protection-mode {
      type enumeration {
        enum "frr" {
          value 0;
          description "fast reroute";
        }
        enum "master-slave" {
          value 1;
          description "master-slave";
        }
        enum "independent" {
          value 2;
```

```
          description "independent";
        }
      }
      description "protection-mode";
    }
    leaf reroute-mode {
      type enumeration {
        enum "immediate" {
          value 0;
          description "immediate reroute";
        }
        enum "delayed" {
          value 1;
          description "delayed reroute";
        }
        enum "never" {
          value 2;
          description "never reroute";
        }
      }
      description "reroute-mode";
    }
    leaf reroute-delay {
      when "../reroute-mode = 'delayed'" {
        description "Specify amount of time to delay reroute " +
                    "only when delayed route is configured";
      }
      type uint16;
      description "amount of time to delay reroute";
    }
    leaf dual-receive {
      type boolean;
      description
      "allow extra traffic to be carried by backup";
    }
    leaf revert {
      type boolean;
      description "allow forwarding to revert to primary " +
                  "after restoring primary";
      /* This is called "revertive" during the discussion. */
    }
    leaf revert-delay {
      when "../revert = 'true'" {
        description "Specify the amount of time to wait to revert " +
                    "to primary only if reversion is configured";
      }
      type uint16;
      description "amount ot time to wait to revert to primary";
```

```
      /* This is called "wtr" during discussion. */
    }
  }

  grouping bridge-table-instance-endpoint-grp {
    description "A bridge table instance's endpoint";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        uses bridge-table-instance-ac-list-grp;
        description "reference to attachment circuits";
      }
      case pw {
        uses bridge-table-instance-pw-list-grp;
        description "reference to pseudowires";
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw {
            description "primary-pw";
            list primary-pw {
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference a pseudowire";
              }
              description "A list of primary pseudowires";
            }
          }
          case primary-ac {
            description "primary-ac";
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          description "backup options";
          case backup-pw {
            list backup-pw {
```

```
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference an attachment circuit";
              }
              leaf precedence {
                type uint32;
                description "precedence of the pseudowire";
              }
              description "A list of backup pseudowires";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
            description "backup-ac";
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        leaf ac {
          type string;
          description "Name of attachment circuit.  This " +
                      "field is intended to reference " +
```

```
                    "standardized layer-2 definitions.";
        }
      }
      case pw {
        leaf pw {
          type leafref {
            path "../../pw/name";
          }
          description "reference to a pseudowire";
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            leaf primary-pw {
              type leafref {
                path "../../pw/name";
              }
              description "primary pseudowire";
            }
          }
          case primary-ac {
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            leaf backup-pw {
              type leafref {
                path "../../pw/name";
              }
              description "backup pseudowire";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
```

```
                          "standardized layer-2 definitions.";
            }
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-state-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        container ac {
          description "ac";
          uses ac-state-grp;
        }
      }
      case pw {
        container pw {
          description "pw";
          uses vpws-pw-state-grp;
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            container primary-pw {
              description "primary pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case primary-ac {
            leaf primary-ac-name {
              type string;
```

```
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            container backup-pw {
              description "backup pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case backup-ac {
            leaf backup-ac-name {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpls-pw-state-grp {
    description "vpls-pw-state-grp";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire up/down state";
    }
    leaf mtu {
      type uint32;
      description "pseudowire mtu";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (true) or disabled (false)";
    }
    leaf cw-negotiation {
```

```
      type cw-negotiation-type;
      description "cw-negotiation";
    }
    leaf discovery-type {
      type l2vpn-discovery-type;
      description "VPLS discovery type";
    }
    leaf signaling-type {
      type l2vpn-signaling-type;
      description "VPLS signaling type";
    }
    leaf peer-ip {
      type inet:ip-address;
      description "peer IP address";
    }
    leaf pw-id {
      type uint32;
      description "pseudowire id";
    }
    leaf transmit-label {
      type uint32;
      description "transmit lable";
    }
    leaf receive-label {
      type uint32;
      description "receive label";
    }
    leaf tunnel-policy {
      type string;
      description "tunnel policy name";
    }
  }

  grouping ac-state-grp {
    description "vpls-ac-state-grp";
    leaf name {
      type string;
      description "attachment circuit name";
    }
    leaf state {
      type operational-state-type;
      description "attachment circuit up/down state";
    }
  }

  grouping vpws-pw-state-grp {
    description "vpws-pw-state-grp";
    leaf name {
```

```
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire operation state up/down";
    }
    leaf mtu {
      type uint32;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (ture) or disabled (false)";
    }
    leaf cw-negotiation {
      type cw-negotiation-type;
      description "Override the control-word negotiation " +
                  "preference specified in the " +
                  "pseudowire template.";
    }
    leaf vccv-ability {
      type boolean;
      description "vccv-ability";
    }
    leaf tunnel-policy {
      type string;
      description "Used to override the tunnel policy name " +
                  "specified in the pseduowire template";
    }
    leaf request-vlanid {
      type uint16;
      description "request vlanid";
    }
    leaf vlan-tpid {
      type string;
      description "vlan tpid";
    }
    leaf ttl {
      type uint8;
      description "time-to-live";
    }
    uses pw-type-grp;
  }

  /* L2VPN YANG Model */

  container l2vpn {
```

```
    description "l2vpn";
    container common {
      description "common l2pn attributes";
      container pw-templates {
        description "pw-templates";
        list pw-template {
          key "name";
          description "pw-template";
          leaf name {
            type string;
            description "name";
          }
          leaf mtu {
            type uint32;
            description "pseudowire mtu";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description
              "control-word negotiation preference";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
      }
      container redundancy-group-templates {
        description "redundancy group templates";
        list redundancy-group-template {
          key "name";
          description "redundancy-group-template";
          leaf name {
            type string;
            description "name";
          }
          uses redundancy-group-properties-grp;
        }
      }
    }
    container bridge-table-instances {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances";
      list bridge-table-instance {
        key "name";
        description "A bridge table instance";
        leaf name {
```

```
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance {
          type string;
          description "Eventual reference to standard EVPN instance";
        }
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            description "cw-negotiation";
          }
          leaf discovery-type {
            type l2vpn-discovery-type;
            description "VPLS discovery type";
          }
          leaf signaling-type {
            type l2vpn-signaling-type;
            description "VPLS signaling type";
```

```
          }
          leaf peer-ip {
            type inet:ip-address;
            description "peer IP address";
          }
          leaf pw-id {
            type uint32;
            description "pseudowire id";
          }
          leaf transmit-label {
            type uint32;
            description "transmit lable";
          }
          leaf receive-label {
            type uint32;
            description "receive label";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          uses bridge-table-instance-endpoint-grp;
          description "List of endpoints";
        }
      }
    }
    container vpws-instances {
      description "vpws-instances";
      list vpws-instance {
        key "name";
        description "A VPWS instance";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf description {
          type string;
```

```
          description "Description of the VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
```

```
            default "preferred";
            description "Override the control-word negotiation " +
                        "preference specified in the " +
                        "pseudowire template.";
          }
          leaf vccv-ability {
            type boolean;
            description "vccvability";
          }
          leaf tunnel-policy {
            type string;
            description "Used to override the tunnel policy name " +
                        "specified in the pseduowire template";
          }
          leaf request-vlanid {
            type uint16;
            description "request vlanid";
          }
          leaf vlan-tpid {
            type string;
            description "vlan tpid";
          }
          leaf ttl {
            type uint8;
            description "time-to-live";
          }
          uses pw-type-grp;
        }
        container endpoint-a {
          description "endpoint-a";
          uses vpws-endpoint-grp;
        }
        container endpoint-z {
          description "endpoint-z";
          uses vpws-endpoint-grp;
        }
      }
    }
  }

  container l2vpn-state {
    config false;
    description "l2vpn state";
    container bridge-table-instances-state {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances-state";
      list bridge-table-instance-state {
        key "name";
```

```
        description "A bridge table instance's state data";
        leaf name {
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-state-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance-name {
          type string;
          description "Name of associated an EVPN instance";
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          choice ac-or-pw-or-redundancy-grp {
            description "A choice ofattachment circuit or " +
                        "pseudowire or redundancy group";
            case ac {
              list ac {
                key "name";
                uses ac-state-grp;
                description "A list of attachment circuits";
              }
              description "attachment circuit endpoint state";
            }
            case pw {
              list pw {
                key "name";
                uses vpls-pw-state-grp;
                description "A list of pseudowires";
              }
              description "pseudowire endpoint state";
            }
```

```
            case redundancy-grp {
              choice primary {
                mandatory true;
                description "primary options";
                case primary-pw {
                  description "primary-pw";
                  list primary-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    description "A list of primary pseudowires";
                  }
                }
                case primary-ac {
                  description "primary-ac";
                  container primary-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              choice backup {
                description "backup options";
                case backup-pw {
                  list backup-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    leaf precedence {
                      type uint32;
                      description "precedence of the pseudowire";
                    }
                    description "A list of backup pseudowires";
                  }
                }
                case backup-ac {
                  description "backup-ac";
                  container backup-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              uses redundancy-group-properties-grp;
            }
          }
          description "List of endpoints";
        }
      }
    }
```

```
    container vpws-instances-state {
      description "vpws-instances-state";
      list vpws-instance-state {
        key "name";
        description "A VPWS instance's state data";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        container endpoint-a {
          description "endpoint-a";
          uses vpws-endpoint-state-grp;
        }
        container endpoint-z {
          description "endpoint-z";
          uses vpws-endpoint-state-grp;
        }
      }
    }
  }
}

<CODE ENDS>
```

Figure 3

6. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The security concerns listed above are, however, no different than faced by other routing protocols. Hence, this draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg]

7. IANA Considerations

None.

8. Acknowledgments

The authors would like to acknowledge TBD for their useful comments.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, DOI 10.17487/RFC3916, September 2004, <http://www.rfc-editor.org/info/rfc3916>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <http://www.rfc-editor.org/info/rfc3985>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <http://www.rfc-editor.org/info/rfc4385>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <http://www.rfc-editor.org/info/rfc4446>.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, DOI 10.17487/RFC4447, April 2006, <http://www.rfc-editor.org/info/rfc4447>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <http://www.rfc-editor.org/info/rfc4448>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <http://www.rfc-editor.org/info/rfc4664>.

[RFC4665] Augustyn, W., Ed. and Y. Serbest, Ed., "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks", RFC 4665, DOI 10.17487/RFC4665, September 2006, <http://www.rfc-editor.org/info/rfc4665>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <http://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <http://www.rfc-editor.org/info/rfc4762>.

[RFC5003] Metz, C., Martini, L., Balus, F., and J. Sugimoto, "Attachment Individual Identifier (AII) Types for Aggregation", RFC 5003, DOI 10.17487/RFC5003, September 2007, <http://www.rfc-editor.org/info/rfc5003>.

[RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, DOI 10.17487/RFC5254, October 2008, <http://www.rfc-editor.org/info/rfc5254>.

[RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <http://www.rfc-editor.org/info/rfc5659>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <http://www.rfc-editor.org/info/rfc6073>.

[RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, DOI 10.17487/RFC6074, January 2011, <http://www.rfc-editor.org/info/rfc6074>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <http://www.rfc-editor.org/info/rfc6391>.

[RFC6423] Li, H., Martini, L., He, J., and F. Huang, "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, DOI 10.17487/RFC6423, November 2011, <http://www.rfc-editor.org/info/rfc6423>.

[RFC6478] Martini, L., Swallow, G., Heron, G., and M. Bocci, "Pseudowire Status for Static Pseudowires", RFC 6478, DOI 10.17487/RFC6478, May 2012, <http://www.rfc-editor.org/info/rfc6478>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <http://www.rfc-editor.org/info/rfc6624>.

[RFC7041] Balus, F., Ed., Sajassi, A., Ed., and N. Bitar, Ed., "Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging", RFC 7041, DOI 10.17487/RFC7041, November 2013, <http://www.rfc-editor.org/info/rfc7041>.

[RFC7361] Dutta, P., Balus, F., Stokes, O., Calvignac, G., and D. Fedyk, "LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS)", RFC 7361, DOI 10.17487/RFC7361, September 2014, <http://www.rfc-editor.org/info/rfc7361>.

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Reshad Rahman
Cisco Systems, Inc.

Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.

Email: skraza@cisco.com

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies

Email: Zhuangshunwan@huawei.com

Wang Haibo
Huawei Technologies

Email: rainsword.wang@huawei.com

Ing-When Chen
Ericsson

Email: ichen@kuatrotech.com

Sajjad Ahmed
Ericsson

Email: sajjad.ahmed@ericsson.com

Mathew Bocci
Alcatel-Lucent

Email: mathew.bocci@alcatel-lucent.com

Jonathan Hardwick
Metaswitch

Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks

Email: sesale@juniper.net

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

Tapraj Singh
Juniper Networks

Email: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Jason Walker
Comcast

Email: jason_walker2@cable.comcast.com

Nick Delregno
Verizon

Email: nick.deregno@verizon.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon

Email: joecylyn.malit@verizon.com