Network Working Group
Internet Draft
Expiration Date: March 2003

Eric C. Rosen
Cisco Systems, Inc.

September 2002

LDP-based Signaling for L2VPNs

draft-rosen-ppvpn-l2-signaling-02.txt

Status of this Memo

Abstract

[MARTINISIG] specifies a way of using LDP [RFC3036] to set up and maintain pseudowires [PWE3-FR]. It requires that each endpoint have apriori knowledge of the IP address other endpoint, and that both endpoints have apriori knowledge of a common 32-bit pseudowire identifier. While this is adequate for the case in which pseudowires are provisioned individually within a single Service Provider's network, there are a variety of PPVPN provisioning models [L2VPN-FW] for which it is not adequate. In particular it is not adequate if it is desired to provision a pseudowire at only one endpoint, or if it is desired to use auto-discovery mechanisms to provision a mesh of pseudowires. It also is not adequate for inter-provider Virtual Private LAN Services (VPLS), or for distributed VPLS. The current draft extends [MARTINISIG] so that LDP-based signaling can be used for these cases.

Table of Contents

1 Specification of Requirements 3
2 Introduction 3
2.1 Deficiencies of Martini Signaling 3
2.2 Protocol Framework 5
2.2.1 Endpoint Identification 5
2.2.2 Association of two LSPs as one Pseudowire 6
3 Attachment Identifiers 6
4 Signaling 7
4.1 Procedures 8
4.2 FEC Element 9
5 Applications 10
5.1 Individual Point-to-Point VCs 10
5.1.1 Provisioning Models 10
5.1.1.1 Double Sided Provisioning 10
5.1.1.2 Single Sided Provisioning with Discovery 10
5.1.2 Signaling 11
5.2 Virtual Private LAN Service 11
5.2.1 Provisioning 12
5.2.2 Auto-Discovery 12
5.2.2.1 BGP-based auto-discovery 12
5.2.2.2 DNS-based auto-discovery 13
5.2.3 Signaling 13
5.3 Colored Pools: Full Mesh of Point-to-Point VCs 14
5.3.1 Provisioning 14
5.3.2 Auto-Discovery 15
5.3.2.1 BGP-based auto-discovery 15
5.3.2.2 DNS-based Auto-Discovery 16
5.3.3 Signaling 16
5.4 Colored Pools: Partial Mesh 16
5.5 Distributed VPLS 17
5.5.1 Signaling 18
5.5.2 Provisioning and Discovery 20
5.5.3 Non-distributed VPLS as a sub-case 20
5.5.4 An Inter-Provider Application of Distributed VPLS Signaling 20
5.5.5 Splicing and the Data Plane 21
6 Backwards Compatibility 22
7 IETF Sub-IP Area Positioning 22
8 Security Considerations 22
9 Acknowledgments 23
10 References 23
11 Author's Information 24

1. 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 RFC 2119

2. Introduction

We make free use of terminology from [L2VPN-FW], [L2VPN-TERM], and [PWE3-FR], in particular the terms "Attachment Circuit", "pseudowire", "PE", "CE".

2.1. Deficiencies of Martini Signaling

In [MARTINISIG], a pseudowire consists of two LSPs (Label Switched Paths), one in each direction. Each endpoint initiates the setup of the LSP that carries packets in the "incoming" direction. (Note that although each LSP is unidirectional, the pseudowire itself is bidirectional.)

Each LSP is uniquely identified by the triple <transmitter, responder, VCid>. (The VCid is a 32-bit quantity which must be unique in the context of a single LDP session between PE1 and PE2.) A pseudowire is a pair of LSPs:

<PE1, PE2, VCid_x, VC_type_y>, <PE2, PE1, VCid_x, VC_type_y>

In order for the signaling to proceed, each endpoint must have apriori knowledge of (a) the IP address of the other endpoint, and (b) the 32-bit VCid. For a given pseudowire, the same VCid must be used when setting up both of the LSPs. In this context, "apriori knowledge" simply means information that must be known prior to the initiation of signaling.

Each endpoint must also have apriori knowledge, for each pseudowire, of the local Attachment Circuit to which that pseudowire is to be bound.

This gives rise to a number of problems:

- In VPLS provisioning, (a) the PE devices are provisioned with VPN-ids, (b) auto-discovery is used to allow a given PE to discover other PEs with which it has a VPN-id in common, (c) a mesh of pseudowires is then set up among these PEs.

  As this provisioning model does not assign any identifiers to the

pseudowires, other than the VPN-id itself, the only way to use [MARTINISIG] to set up these pseudowires is to treat the VPN-id as if it were VCid. However, [MARTINISIG] only allows 32 bits for encoding a VCid. This is not adequate for VPLS. To accommodate inter-provider VPLS, VPN-ids must be globally unique. There are a number of schemes for assigning globally unique VPN-ids, but in general they require more than 32 bits for a VPN-id. (E.g., [RFC2685] uses 7 bytes; [RFC2547bis] uses 8 bytes for the quantities that play the role of VPN-id; [DNS-L2TP-VPLS] uses arbitrarily long DNS names.) An extension is needed to allow VPN-ids longer than 32 bits to be carried by the signaling protocol.

- In distributed VPLS [e.g. L2VPN-FW section 3.4.3], for a given VPN-id there may be more than one pseudowire between a given pair of nodes. This makes it impossible to treat the VPN-id as a pseudowire identifier; the VPN-id can be at most part of the pseudowire identifier.

- In the VPWS "colored pools provisioning model" of [L2VPN-FW] section 3.3.1.3, or the provisioning model of [BGP-SIGNALING], the basic identification mechanisms are endpoint identifiers, rather than pseudowire identifiers. A pseudowire can then be identified by a pair of endpoint identifiers. Encoding a pair of endpoint identifiers into a single 32-bit VCid field would be very restrictive.

- It is sometimes desirable for all the pseudowire signaling information to be provisioned at one end of the pseudowire, without any need to provision the other end in advance of signaling. This would be necessary, for example, if one were using the pseudowires to emulate Switched VCs rather than Permanent VCs. This immediately rules out any signaling technique in which both endpoints need apriori knowledge of a common identifier.

These problems can be eliminated with a small number of relatively minor extensions to [MARTINISIG]. The purpose of this paper is to specify those extensions, and to show how the resulting protocol can be used together with an auto-discovery mechanism to support a large set of L2VPN provisioning models.

We do not specify an auto-discovery procedure in this draft, but we do specify the information which needs to be obtained via auto-discovery in order for the signaling procedures to begin. The way in which the LDP-based signaling mechanisms can be integrated with BGP-based auto-discovery is covered in some detail. Later revisions of this draft will provide equivalent detail for other discovery

mechanisms.

2.2. Protocol Framework

2.2.1. Endpoint Identification

Per [L2VPN-FW], a pseudowire can be thought of as a relationship between a pair of "Forwarders". In simple instances of VPWS, a Forwarder binds a pseudowire to a single Attachment Circuit, such that frames received on the one are sent on the other, and vice versa. In VPLS, a Forwarder binds a set of pseudowires to a set of Attachment Circuits; when a frame is received from any member of that set, a MAC address table is consulted (and various 802.1d procedures executed) to determine the member or members of that set on which the frame is to be transmitted. In more complex scenarios, Forwarders may bind PWs to PWs, thereby "splicing" two PWs together; this is needed, e.g., to support distributed VPLS.

In simple VPWS, where a Forwarder binds exactly one PW to exactly one Attachment Circuit, a Forwarder can be identified by identifying its Attachment Circuit. In simple VPLS, a Forwarder can be identified by identifying its PE device and its VPN.

To set up a PW between a pair of Forwarders, the signaling protocol must allow the Forwarder at one endpoint to identify the Forwarder at the other. We use the term "Attachment Identifier", or "AI", to refer to a quantity whose purpose is to identify a Forwarder.

In [MARTINISIG], the only identifier is the VCid. The implicit endpoint identifiers are "the Forwarder that is configured to be the endpoint of the pseudowire identified by the specified VCid". We propose to replace the single VCid of [MARTINISIG] with a pair of Attachment Identifiers, one for each of the two endpoints.

Although this draft discusses only LDP-based signaling, it is possible that the very same Attachment Identifier mechanism will be useful for L2TP-based signaling. This issue is for further study.

Different Forwarders support different applications. The particular application for which a given PW is used will depend on the Forwarder that is identified when setting up the PW. There is, for example, no signaling to distinguish a PW used in VPLS from a PW used in a point-to-point service, as this can be inferred from the identity of the Forwarder.

2.2.2. Association of two LSPs as one Pseudowire

In any form of LDP-based signaling, each PW endpoint must initiate the creation of a unidirectional LSP. A PW is a pair of such LSPs. In most of the PPVPN provisioning models, the two endpoints of a given PW can simultaneously initiate the signaling for it. They must therefore have some way of determining when a given pair of LSPs are intended to be associated together as a single PW.

The way in which this association is done is different for the various different L2VPN services and provisioning models. The details appear in later sections.

3. Attachment Identifiers

Every Forwarder in a PE must be associated with an Attachment Identifier (AI), either through configuration or through some algorithm. The Attachment Identifier must be unique in the context of the PE router in which the Forwarder resides. The combination <PE router, AI> must be globally unique.

It is frequently convenient to a set of Forwarders as being members of a particular "group", where PWs may only be set up among members of a group. In such cases, it is convenient to identify the Forwarders relative to the group, so that an Attachment Identifier would consist of an Attachment Group Identifier (AGI) plus an Attachment Individual Identifier (AII).

An Attachment Group Identifier may be thought of as a VPN-id, or a VLAN identifier, some attribute which is shared by all the Attachment VCs (or pools thereof) which are allowed to be connected.

The details for how to construct the AGI and AII fields identifying the pseudowire endpoints in particular provisioning models are discussed later in this paper.

We can now consider an LSP to be identified by:

<PE1, <AGI, AII1>, PE2, <AGI, AII2>>,

and the LSP in the opposite direction will be identified by:

<PE2, <AGI, AII2>, PE1, <AGI, AII1>>;

a pseudowire is a pair of such LSPs.

When a signaling message is sent from PE1 to PE2, and PE1 needs to

refer to an Attachment Identifier which has been configured on one of its own Attachment VCs (or pools), the Attachment Identifier is called a "Source Attachment Identifier". If PE1 needs to refer to an Attachment Identifier which has been configured on one of PE2's Attachment VCs (or pools), the Attachment Identifier is called a "Target Attachment Identifier". (So an SAI at one endpoint is a TAI at the remote endpoint, and vice versa.)

In the signaling protocol, we define encodings for the following three fields:

- Attachment Group Identifier (AGI).

- Source Attachment Individual Identifier (SAII)

- Target Attachment Individual Identifier (TAII)

If the AGI is non-null, then the SAI consists of the AGI together with the SAII, and the TAI consists of the TAII together with the AGI. If the AGI is null, then the SAII and TAII are the SAI and TAI respectively.

The intention is that the PE which receives a Label Mapping Message containing a TAI will be able to map that TAI uniquely to one of its Attachment VCs (or pools). The way in which a PE maps a TAI to an Attachment VC (or pool) should be a local matter. So as far as the signaling procedures are concerned, the TAI is really just an arbitrary string of bytes, a "cookie".

4. Signaling

An LDP Label Mapping message contains a FEC TLV, a Label TLV, and zero or more optional parameter TLVs. [MARTINISIG] defines a FEC TLV, containing a single FEC element, containing a VC type, a fixed length group id, a fixed length VCid, and variable length "interface parameters".

We propose to extend [MARTINISIG] by adding a new FEC type (provisionally type 129, subject to assignment by IANA) in which the group id and and VCid are eliminated, and their place taken by variable length SAII, AGI, and TAII fields. In other respects, the Label Mapping messages will be the same.

4.1. Procedures

In order for PE1 to begin signaling PE2, PE1 must know the address of the remote PE2, and a TAI. This information may have been configured at PE1, or it may have been learned dynamically via some autodiscovery procedure.

To begin the signaling procedure, a PE (PE1) that has knowledge of the other endpoint (PE2) initiates the setup of the LSP in the incoming (PE2-->PE1) direction by sending a Label Mapping message containing the new FEC type. The FEC element includes the SAII, AGI, and TAII.

What happens when PE2 receives such a Label Mapping message?

PE2 interprets the message as a request to set up a PW whose endpoint (at PE2) is the Forwarder identified by the TAI. From the perspective of the signaling protocol, exactly how PE2 maps AIs to Forwarders is a local matter. In some VPWS provisioning models, the TAI might, e.g., be a string which identifies a particular Attachment Circuit, such as "ATM3VPI4VCI5", or it might, e.g., be a string such as "Fred" which is associated by configuration with a particular Attachment Circuit. In VPLS, the TAI would be a VPN-id, identifying a particular VPLS instance.

If PE2 cannot map the TAI to one of its Forwarders, then PE2 sends a Label Release message to PE1, with a Status Code meaning "invalid TAI", and the processing of the Mapping message is complete.

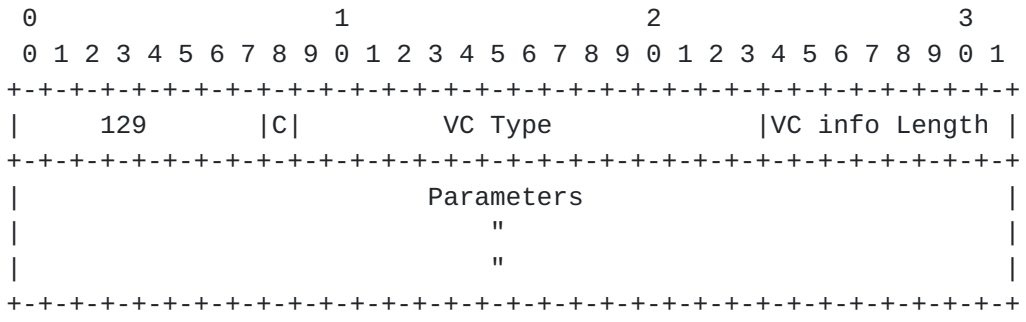
If the Label Mapping Message has a valid TAI, PE2 must decide whether to accept it or not. The procedures for so deciding will depend on the particular type of Forwarder identified by the TAI. As the details are specific to the type of Forwarder, they are specified in later sections where we discuss the different provisioning models that can be supported.

Of course, the Label Mapping message may be rejected due to standard LDP error conditions as detailed in [LDP].

If PE2 decides to accept the Label Mapping message, then it has to make sure that an LSP is set up in the opposite (PE1-->PE2) direction. If it has already signaled for the corresponding LSP in that direction, nothing more need be done. Otherwise, it must initiate such signaling by sending a Label Mapping message to PE1. This is very similar to the Label Mapping message PE2 received, but with the SAI and TAI reversed.

4.2. FEC Element

FEC element type 129 is used. The FEC element is encoded as follows:



VC Type is as defined in [MARTINISIG] and its various extensions [e.g., VPLS2].

C and VC info length are as defined in [MARTINISIG].

Parameters are:

- SAII, encoded as a one byte length field followed by the SAI.
- TAII, encoded as a one byte length field followed by the TAI.
- AGI, encoded as a one byte length field followed by the AGI.
- Interface parameters, as defined in [MARTINISIG].

The SAII, TAII, and AGI are simply carried as octet strings. The length byte specifies the size of the field, excluding the length byte itself. The null string can be sent by setting the length byte to 0.

5. Applications

In this section, we specify the way in which the above procedures are applied for a number of different applications. For some of the applications, we specify the way in which different provisioning models can be used. However, this is not meant to be an exhaustive list of the applications, or an exhaustive list of the provisioning models that can be applied to each application.

5.1. Individual Point-to-Point VCs

The signaling specified in this document can be used to set up individually provisioned point-to-point pseudowires. In this application, each Forwarder binds a single PW to a single Attachment Circuit. Each PE must be provisioned with the necessary set of Attachment Circuits, and then certain parameters must be provisioned for each Attachment Circuit.

5.1.1. Provisioning Models

5.1.1.1. Double Sided Provisioning

In this model, the Attachment Circuit must be provisioned with a local name, a remote PE address, and a remote name. During signaling, the local name is sent as the SAII, the remote name as the TAII, and the AGI is null. If two Attachment Circuits are to be connected by a PW, the local name of each must be the remote name of the other.

5.1.1.2. Single Sided Provisioning with Discovery

In this model, each Attachment circuit must be provisioned with a local name. The local name consists of a VPN-id (signaled as the AGI) and an Attachment Individual Identifier which is unique relative to the AGI. If two Attachment circuits are to be connected by a PW, only one of them needs to be provisioned with a remote name (which of course is the local name of the other Attachment Circuit). Neither needs to be provisioned with the address of the remote PE, but both must have the same VPN-id.

As part of an auto-discovery procedure, each PE advertises its <VPN-id, local AII> pairs. Each PE compares its local <VPN-id, remote AII> pairs with the <VPN-id, local AII> pairs advertised by the other PEs. If PE1 has a local <VPN-id, remote AII> pair with value <V, fred>, and PE2 has a local <VPN-id, local AII> pair with value <V,

fred>, PE1 will thus be able to discover that it needs to connect to PE2. When signaling, it will use "fred" as the TAII, and will use V as he AGI. A null SAII is sent.

The primary benefit of this provisioning model when compared to Double Sided Provisioning is that it enables one to move an Attachment Circuit from one PE to another without having to reconfigure the remote endpoint.

5.1.2. Signaling

Signaling is as specified in section 4 above, with the addition of the following:

When a PE receives a Label Mapping Message, and the TAI identifiers a particular Attachment Circuit which is configured to be bound to a point-to-point PW, then the following checks must be made.

If the Attachment Circuit is already bound to a pseudowire (including the case where only one of the two LSPs currently exists), and the remote endpoint is not PE1, then PE2 sends a Label Release message to PE1, with a Status Code meaning "Attachment Circuit bound to different PE", and the processing of the Mapping message is complete.

If the Attachment Circuit is already bound to a pseudowire (including the case where only one of the two LSPs currently exists, but the AI at PE1 is different than that specified in the AGI/SAII fields of the Mapping message) then PE2 sends a Label Release message to PE1, with a Status Code meaning "Attachment Circuit bound to different remote Attachment Circuit", and the processing of the Mapping message is complete.

These errors could occur as the result of misconfigurations.

5.2. Virtual Private LAN Service

In the VPLS application [VPLS1, VPLS2], the Attachment Circuits can be though of as LAN interfaces which attach to "virtual LAN switches", or, in the terminology of [L2VPN-FW], "Virtual Switching Instances" (VSIs). Each Forwarder is a VSI that attaches to a number of PWs and a number of Attachment Circuits. The VPLS service [VPLS1, VPLS2] requires that a single pseudowire be created between each pair of VSIs that are in the same VPLS. Each PE device may have a multiple VSIs, where each VSI belongs to a different VPLS.

5.2.1. Provisioning

Each VPLS must have a globally unique identifier, which we call a VPN-id. Every VSI must be configured with the VPN-id of the VPLS to which it belongs.

Each VSI must also have a unique identifier, but this can be formed automatically by concatenating its VPN-id with the IP address of its PE router.

5.2.2. Auto-Discovery

5.2.2.1. BGP-based auto-discovery

The framework for BGP-based auto-discovery for a VPLS service is as specified in [BGP-AUTO], section 3.2.

The AFI/SAFI used would be:

- An AFI specified by IANA for L2VPN. (This is the same for all L2VPN schemes.)

- An SAFI specified by IANA specifically for a VPLS service whose pseudowires are set up using the procedures described in the current document.

In order to use BGP-based auto-discovery as specified in [BGP-AUTO], the globally unique identifier associated with a VPLS must be encodable as an 8-byte Route Distinguisher (RD). If the globally unique identifier for a VPLS is an RFC2685 VPN-id, it can be encoded as an RD as specified in [BGP-AUTO]. However, any other method of assigning a unique identifier to a VPLS and encoding it as an RD (using the encoding techniques of [RFC2547bis]) will do.

Each VSI needs to have a unique identifier, which can be encoded as a BGP NLRI. This is formed by prepending the RD (from the previous paragraph) to an IP address of the PE containing the virtual LAN switch.

(Note that it is not strictly necessary for all the VSIs in the same VPLS to have the same RD, all that is really necessary is that the NLRI uniquely identify a virtual LAN switch.)

Each VSI needs to be associated with one or more Route Target (RT) Extended Communities, as discussed in [BGP-AUTO}. These control the distribution of the NLRI, and hence will control the formation of the overlay topology of pseudowires that constitutes a particular VPLS.

Auto-discovery proceeds by having each PE distribute, via BGP, the NLRI for each of its VSIs, with itself as the BGP next hop, and with the appropriate RT for each such NLRI. Typically, each PE would be a client of a small set of BGP route reflectors, which would redistribute this information to the other clients.

If a PE has a VSI with a particular RT, it can then receive all the NLRI which have that same RT, and from the BGP next hop attribute of these NLRI will learn the IP addresses of the other PE routers which have VSIs with the same RT. The considerations of [RFC2547bis] section 4.3.3 on the use of route reflectors apply.

If a particular VPLS is meant to be a single fully connected LAN, all its VSIs will have the same RT, in which case the RT could be (though it need not be) an encoding of the VPN-id. If a particular VPLS consists of multiple VLANs, each VLAN must have its own unique RT. A VSI can be placed in multiple VLANS (or even in multiple VPLSes) by assigning it multiple RTs.

Note that hierarchical VPLS can be set up by assigning multiple RTs to some of the virtual LAN switches; the RT mechanism allows one to have complete control over the pseudowire overlay which constitutes the VPLS topology.

5.2.2.2. DNS-based auto-discovery

[DNS-LDP-VPLS] includes a proposal for using DNS-based auto-discovery.

5.2.3. Signaling

It is necessary to create Attachment Identifiers which identify the VSIs. Given that each VPLS has at most one VSI per PE, and that only one PW is permitted between any pair of VSIs, a VSI can be uniquely identified (relative to its PE) by the VPN-id of its VPLS. Therefore the signaling messages can encode the VPN-id in the AGI field, and use the null values of the SAII and TAII fields.

The VPN-id may be encoded as an [RFC2547bis] RD, in which case the AGI field consist of a length field of value 8, followed by the 8 bytes of the RD. If the VPN-id is an RFC2685 VPN-id, it should be encoded as an RD (as specified in [BGP-AUTO]), and then the RD should be carried in the AGI field.

If the VPN-id is a DNS name, the first byte of the AGI field (immediately following the length) will be 0x90. This distinguishes

it from any RD. The DNS name itself then follows.

Note that it is not possible using this technique to set up more than one PW per pair of VSIs.

5.3. Colored Pools: Full Mesh of Point-to-Point VCs

In the "Colored Pools" model of operation, each PE may contain several pools of Attachment Circuits, each pool associated with a particular VPN. A PE may contain multiple pools per VPN, as each pool may correspond to a particular CE device. It may be desired to create one pseudowire between each pair of pools that are in the same VPN; the result would be to create a full mesh of CE-CE VCs for each VPN. (This application was originally suggested in [BGP-SIGNALING]; we show here that it can be done with LDP-based signaling.)

5.3.1. Provisioning

Each pool is configured, and associated with:

- a set of Attachment Circuits; whether these Attachment Circuits must themselves be provisioned, or whether they can be auto-allocated as needed, is independent of and orthogonal to the procedures described in this document;

- a "color", which can be thought of as a VPN-id of some sort;

- a relative pool identifier, which is unique relative to the color.

The pool identifier, and color, taken together, constitute a globally unique identifier for the pool. Thus if there are n pools of a given color, their pool identifiers can be (though they do not need to be) the numbers 1-n.

The semantics are that a pseudowire will be created between every pair of pools that have the same color, where each such pseudowire will be bound to one Attachment Circuit from each of the two pools.

If each pool is a set of Attachment Circuits leading to a single CE device, then the layer 2 connectivity among the CEs is controlled by the way the colors are assigned to the pools. To create a full mesh, the "color" would just be a VPN-id.

Optionally, a particular Attachment Circuit may be configured with the relative pool identifier of a remote pool. Then that Attachment

Circuit would be bound to a particular pseudowire only if that pseudowire's remote endpoint is the pool with that relative pool identifier. With this option, the same pairs of Attachment Circuits will always be bound via pseudowires.

5.3.2. Auto-Discovery

5.3.2.1. BGP-based auto-discovery

The framework for BGP-based auto-discovery for a colored pool service is as specified in [BGP-AUTO], section 3.2.

The AFI/SAFI used would be:

- An AFI specified by IANA for L2VPN. (This is the same for all L2VPN schemes.)

- An SAFI specified by IANA specifically for a Colored Pool L2VPN service whose pseudowires are set up using the procedures described in the current document.

In order to use BGP-based auto-discovery, the color associated with a colored pool must be encodable as both an RT (Route Target) and an RD (Route Distinguisher). The globally unique identifier of a pool must be encodable as NLRI; the color would be encoded as the RD and the pool identifier as a four-byte quantity which is appended to the RD to create the NLRI.

Auto-discovery procedures by having each PE distribute, via BGP, the NLRI for each of its pools, with itself as the BGP next hop, and with the RT that encodes the pool's color. If a given PE has a pool with a particular color (RT), it must receive, via BGP, all NLRI with that same color (RT). Typically, each PE would be a client of a small set of BGP route reflectors, which would redistribute this information to the other clients.

If a PE has a pool with a particular color, it can then receive all the NLRI which have that same color, and from the BGP next hop attribute of these NLRI will learn the IP addresses of the other PE routers which have pools switches with the same color. It also learns the unique identifier of each such remote pool, as this is encoded in the NLRI. The remote pool's relative identifier can be extracted from the NLRI and used in the signaling, as specified below.

5.3.2.2. DNS-based Auto-Discovery

The use of DNS-based auto-discovery for the colored pool model of operation is for further study.

5.3.3. Signaling

When a PE sends a Label Mapping message to set up a PW between two pools, it encodes the color as the AGI, the local pool's relative identifier as the SAII, and the remote pool's relative identifier as the TAII.

When PE2 receives a Label Mapping message from PE1, and the TAI identifies to a pool, and there is already an pseudowire connecting an Attachment Circuit in that pool to an Attachment Circuit at PE1, and the AI at PE1 of that pseudowire is the same as the SAI of the Label Mapping message, then PE2 sends a Label Release message to PE1, with a Status Code meaning "Attachment Circuit bound to different remote Attachment Circuit". This prevents the creation of multiple pseudowires between a given pair of pools.

Note that the signaling itself only identifies the remote pool to which the pseudowire is to lead, not the remote Attachment Circuit which is to be bound to the the pseudowire. However, the remote PE may examine the SAII field to determine which Attachment Circuit should be bound to the pseudowire.

5.4. Colored Pools: Partial Mesh

The procedures for creating a partial mesh of pseudowires among a set of colored pools are substantially the same as those for creating a full mesh, with the following exceptions:

- Each pool is optionally configured with a set of "import RTs" and "export RTs";

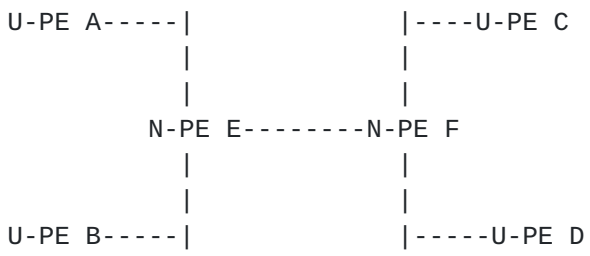
- During BGP-based auto-discovery, the pool color is still encoded in the RD, but if the pool is configured with a set of "export RTs", these are are encoded in the RTs of the BGP Update messages, INSTEAD the color.

- If a pool has a particular "import RT" value X, it will create a PW to every other pool which has X as one of its "export RTs". The signaling messages and procedures themselves are as in section 5.3.3

5.5. Distributed VPLS

In Distributed VPLS ([L2VPN-FW], [DTLS], [LPE]), the VPLS functionality of a PE router is divided among two systems: a U-PE and an N-PE. The U-PE sits between the user and the N-PE. VSI functionality (e.g., MAC address learning and bridging) is performed on the U-PE. A number of U-PEs attach to an N-PE. For each VPLS supported by a U-PE, the U-PE maintains a pseudowire to each other U-PE in the same VPLS. However, the U-PEs do not maintain signaling control connections with each other. Rather, each U-PE has only a single signaling connection, to its N-PE. In essence, each U-PE-to-U-PE pseudowire is composed of three pseudowires spliced together: one from U-PE to N-PE, one from N-PE to N-PE, and one from N-PE to U-PE.

Consider for example the following topology:



where the four U-PEs are in a common VPLS. In distributed VPLS, there will be three PWs from A to E. Call these A-E/1, A-E/2, and A-E/3. There will be two PWs from E to F. Call these E-F/1 and E-F/2. And there will be three more PWs, one from E to B (E-B/1), one from F to C (F-C/1), and one from F to D (F-D/1).

The N-PEs must then splice these pseudowires together to get the equivalent of what the non-distributed VPLS signaling mechanism would provide:

- PW from A to B: A-E/1 gets spliced to E-B/1.
- PW from A to C: A-E/2 gets spliced to E-F/1 gets spliced to F-C/1.

- PW from A to D: A-E/3 gets spliced to E-F/2 gets spliced to F-D/1.

It doesn't matter which PWs get spliced together, as long as the result is one from A to each of B, C, and D.

One can see that distributed VPLS does not reduce the number of pseudowires per U-PE, but it does reduce the number of control connections per U-PE. Whether this is worthwhile depends, of course, on what the bottleneck is.

5.5.1. Signaling

The signaling to support Distributed VPLS can be done with the mechanisms described in this paper. However, the procedures for VPLS (section 5.2.3) presuppose that, between a pair of PEs, there is only one PW per VPLS. In distributed VPLS, this isn't so. In the topology above, for example, there are two PWs between A and E for the same VPLS. For distributed VPLS therefore, one cannot identify the Forwarders merely by using the VPN-id as the AGI, while using null values of the SAII and TAII. Rather, the SAII and TAII must be used to identify particular U-PE devices.

At a given N-PE, the directly attached U-PEs in a given VPLS can be numbered from 1 to n. This number identifies the U-PE relative to a particular VPN-id and a particular PE. (That is, to uniquely identify the U-PE, the N-PE, the VPN-id, and the U-PE number must be known.)

As a result of configuration/discovery, each U-PE must be given a list of <j, IP address> pairs. Each element in this list tells the U-PE to set up j PWs to the specified IP address. When the U-PE signals to the N-PE, it sets the AGI to the proper-VPN-id, and sets the SAII to the PW number, and sets the TAII to null.

In the above example, U-PE A would be told <3, E>, telling it to set up 3 PWs to E. When signaling, A would set the AGI to the proper VPN-id, and would set the SAII to 1, 2, or 3, depending on which of the three PWs it is signaling.

As a result of configuration/discovery, each N-PE must be given the following information for each VPLS:

- A "Local" list: {<j, IP address>}, where each element tells it to set up j PWs to the locally attached U-PE at the specified address. The number of elements in this list will be n, the number of locally attached U-PEs in this VPLS. In the above

example, E would be given the local list: {<3, A>, <3, B>}, telling it to set up 3 PWs to A and 3 to B.

- A local numbering of its U-PEs, relative to a VPLS. In the above example, E could be told that U-PE A is 1, and U-PE B is 2.

- A "Remote" list: {<IP address, k>}, telling it to set up k PWs, for each U-PE, to the specified IP address. Each of these IP addresses identifies a N-PE, and k specifies the number of U-PEs at that N-PE which are in the VPLS. In the above example, E would be given the remote list: {<2, F>}. Since N-PE has two U-PEs, this tells it to set up 4 PWs to N-PE F, 2 for each of its E's U-PEs.

The signaling of a PW from N-PE to U-PE is based on the local list and the local numbering of U-PEs. When signaling a particular PW from an N-PE to a U-PE, the AGI is set to the proper VPN-id, and SAII is set to null, and the TAII is set to the PW number (relative to that particular VPLS and U-PE). In the above example, when E signals to A, it would set the TAII to be 1, 2, or 3, respectively, for the three PWs it must set up to A. It would similarly signal three PWs to B.

The LSP signaled from U-PE to N-PE is associated with an LSP from N-PE to U-PE in the usual manner, as specified in section 4. A PW between a U-PE and an N-PE is known as a "U-PW".

The signaling of a PW from N-PE to N-PE is based on the remote list. When signaling a particular PW from an N-PE to an N-PE, the AGI is set to the appropriate VPN-id. The remote list specifies the number of PWs to set up, per local U-PE, to a particular remote N-PE. If there are n such PWs, they are distinguished by the setting of the TAII, which will be a number from 1 to n inclusive. The SAII is set to the local number of the U-PE. In the above example, E would set up 4 PWs to F. The SAII/TAII fields would be set to 1/1, 1/2, 2/1, and 2/2 respectively. A PW between two N-PEs is known as an "N-PW".

Each U-PW must be "spliced" to an N-PW. This is based on the remote list. If the remote list contains an element <i, F>, then i U-PWs from each local U-PE must be spliced to N-PWs from the remote N-PE F. It does not matter which U-PWs are spliced to which N-PWs, as long as this constraint is met.

If an N-PE has more than one local U-PE for a given VPLS, it must also ensure that a U-PW from each such U-PE is spliced to a U-PW from each of the other U-PEs.

5.5.2. Provisioning and Discovery

Every N-PE must be provisioned with the set of VPLS instances it supports, a VPN-id for each one, and a list of local U-PEs for each such VPLS. As part of the discovery procedure, the N-PE advertises the number of U-PEs for each VPLS.

Auto-discovery (e.g., BGP-based) can be used to discover all the other N-PEs in the VPLS, and for each, the number of U-PEs local to that N-PE. From this, one can compute the total number of U-PEs in the VPLS. This information is sufficient to enable one to compute the local list and the remote list for each N-PE.

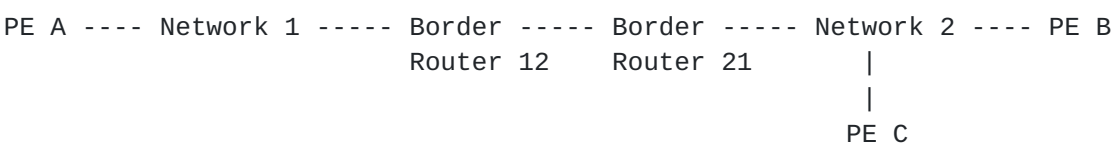
5.5.3. Non-distributed VPLS as a sub-case

A PE which is providing "non-distributed VPLS" (i.e., a PE which peforms both the U-PE and N-PE functions) can interoperate with N-PE/U-PE pairs that are providing distributed VPLS. The "non-distributed PE" simply advertises, in the discovery procedure, that it has one local U-PE per VPLS. And of course, the non-distributed PE does no splicing.

If every PE in a VPLS is providing non-distributed VPLS, and thus every PE advertises itself as an N-PE with one local U-PE, the resultant signaling is exactly the same as that specified in section 5.2.3 above, except that SAII and TAII values of 1 are used instead of SAII and TAII values of null.

5.5.4. An Inter-Provider Application of Distributed VPLS Signaling

Consider the following topology:



where A, B, and C are PEs in a common VPLS, but Networks 1 and 2 are networks of different Service Providers. Border Router 12 is Network 1's border router to network 2, and Border Router 21 is Network 2's border router to Network 1. We suppose further that the PEs are not "distributed", i.e, that each provides both the U-PE and N-PE functions.

In this topology, one needs two inter-provider pseudowires: A-B and A-C.

Suppose a Service Provider decides, for whatever reason, that it does not want each of its PEs to have a control connection to any PEs in the other network. Rather, it wants the inter-provider control connections to run only between the two border routers.

This can be achieved using the techniques of section 5.5, where the PEs behave like U-PEs, and the BRs behave like N-PEs. In the example topology, PE A would behave like a U-PE which is locally attached to BR12; PEs B and C would be have like U-PEs which are locally attached to BR21; and the two BRs would behave like N-PEs.

As a result, the PW from A to B would consist of three segments: A-BR12, BR12-BR21, and BR21-B. The border routers would have to splice the corresponding segments together.

This requires the PEs within a VPLS to be numbered from 1-n (relative to that VPLS) within a given network.

5.5.5. Splicing and the Data Plane

Splicing two PWs together is quite straightforward in the MPLS data plane, as moving a packet from one PW directly to another is just a label replace operation on the PW label. When a PW consists of two PWs spliced together, it is assumed that the data will go to the node where the splicing is being done, i.e., that the data path will include the control points.

In some cases, it may be desired to have the data go on a more direct route from one "true endpoint" to another, without necessarily passing through the splice points. This could be done by means of a new LDP TLV carried in the LDP mapping message; call it the "direct route" TLV. A direct route TLV would be placed in an LDP Label Mapping message by the LSP's "true endpoint". The TLV would specify the IP address of the true endpoint, and would also specify a label, representing the pseudowire, which is assigned by that endpoint. When PWs are spliced together at intermediate control points, this TLV would simply be passed upstream. Then when a frame is first put on the pseudowire, it can be given this pseudowire label, and routed to the true endpoint, thereby possibly bypassing the intermediate control points.

6. Backwards Compatibility

It may be desirable to have nodes which can use either the procedures described herein, or the unaltered procedures of [MARTINISIG]. In that case, the procedures described herein would be used if and only if both sides were capable of using them.

This can be done by defining a new TLV for the LDP Label Mapping message. Call it the "Extended L2 Signaling TLV". A node which can support the messages and procedures of this draft as well as the messages and procedures of [MARTINISIG] would, if so configured, initiate signaling using the [MARTINISIG] messages, but including the Extended L@ Signaling TLV in the LDP Mapping Message.

If the other node does not understand this TLV, it will simply ignore it, and [MARTINISIG] will be used.

When a node which supports this backwards compatibility feature receives an LDP mapping message containing a [MARTINISIG] FEC, but with the Extended L2 Signaling TLV, it will send a corresponding Label Release message, and will re-initiate signaling of that pseudowire with the messages described in this draft.

7. IETF Sub-IP Area Positioning

This draft is targeted at both the PPVPN WG and the MPLS WG. It appears to be in the province of the PPVPN WG to consider the requirements of signaling to support layer 2 VPNs. Specification in detail of the actual extensions to LDP would appear to be the province of the MPLS WG.

8. Security Considerations

The signaling procedures specified herein require that a node initiate and/or accept LDP sessions with entities that are not necessarily directly connected to that node. It would be advisable for a given node to use access control to restrict the set of nodes that can set up LDP sessions with it, and it would be advisable to use some form of authentication to guarantee that the remote endpoint of an LDP session is the entity that it claims to be. Using the TCP MD5 option may be adequate, or alternatively IPsec can be used.

9. Acknowledgments

Thanks to Dan Tappan, Ted Qian, Bruce Davie, Ali Sajassi, Wei Luo, and Skip Booth for their comments, criticisms, and helpful suggestions.

Thanks to Tissa Senevirathne, Hamid Ould-Brahim and Yakov Rekhter for discussing the auto-discovery issues.

10. References

[BGP-AUTO] "Using BGP as an Auto-Discovery Mechanism for Network-based VPNs", Ould-Brahim et. al., draft-ietf-ppvpn-bgpvpn-auto-02.txt, February 2002.

[BGP-SIGNALING] "Layer 2 VPNs over Tunnels", Kompella et. al., draft-kompella-ppvpn-l2vpn-02.txt, June 2002

[DNS-L2TP-VPLS] "DNS/LDP Based VPLS", Heinanen, draft-heinanen-dns-ldp-vpls-00.txt, June 2002

[L2VPN-FW] "PPVPN L2 Framework", Andersson et. al., draft-ietf-ppvpn-l2-framework-00.txt, August 2002

[L2VPN-TERM] "PPVPN Terminology", Andersson, Madsen, draft-andersson-ppvpn-terminology-01.txt, June 2002

[LDP] "LDP Specification", Andersson, et. al., RFC 3036, January 2001

[MARTINISIG] "Transport of Layer 2 Frames Over MPLS", Martini et. al., draft-martini-l2circuit-trans-mpls-10.txt, August 2002

[PWE3-FR] " Framework for Pseudo Wire Emulation Edge-to-Edge ", draft-pate-pwe3-framework-03.txt, January 2002

[RFC2547bis], "BGP/MPLS VPNs", Rosen, Rekhter, et. al., draft-ietf-ppvpn-rfc2547bis-02.txt, July 2002

[RFC2685] "Virtual Private Networks Identifier", Fox, Gleeson, September 1999

[RFC3036] "LDP Specification", January 2001

[VPLS1] "Requirements for Virtual Private Network Services", Augustyn et. al., draft-augustyn-ppvpn-l2vpn-requirements-00.txt, June 2002

[VPLS2] "Transparent VLAN Services over MPLS", Laserre, et. al.,

draft-lasserre-vkompella-ppvpn-vpls-02.txt, June 2002

11. Author's Information

Eric C. Rosen
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: erosen@cisco.com