Network Working Group
Internet Draft
Expiration Date: November 2003

Eric C. Rosen
Cisco Systems, Inc.

May 2003

Provisioning Models and Endpoint Identifiers in L2VPN Signaling

draft-rosen-ppvpn-l2-signaling-03.txt

Status of this Memo

Abstract

[PWE3-CONTROL] specifies a "signaling protocol" which uses extensions of LDP [RFC 3036] to set up and maintain pseudowires [PWE3-FR, PWE3-ARCH]. Like any protocol which sets up connections, the signaling protocol provides a method by which each endpoint can identify the other. [L2VPN-FW] describes a number of different ways in which sets of pseudowires may be combined together into "Provider Provisioned Layer 2 VPNs" (L2 PPVPNs, or L2VPNs), resulting in a number of different kinds of L2VPN. Different kinds of L2VPN may have different "provisioning models", i.e., different models for what information needs to be configured in what entities. Once configured, the provisioning information is distributed by a "discovery process", and once the information is discovered, the signaling protocol is automatically invoked to set up the required pseudowires. The semantics of the endpoint identifiers which the signaling protocol

uses for a particular type of L2VPN are determined by the provisioning model. This document specifies a number of PPVPN provisioning models, and specifies the semantic structure of the endpoint identifiers required. It further specifies how the endpoint identifiers are carried in the "Generalized Identifier FEC" of [PWE3-CONTROL]. It is believed that the specified identifiers can also be carried within the L2TP signaling protocol, though this is currently not specified.

Contents

1. Introduction

[PWE3-CONTROL] specifies a "signaling protocol" which uses extensions of LDP [RFC 3036] to set up and maintain pseudowires [PWE3-FR, PWE3-ARCH]. Like any protocol which sets up connections, the signaling protocol provides a method by which each endpoint can identify the other. [L2VPN-FW] describes a number of different ways in which sets of pseudowires may be combined together into "Provider Provisioned Layer 2 VPNs" (L2 PPVPNs, or L2VPNs), resulting in a number of different kinds of L2VPN. Different kinds of L2VPN may have different "provisioning models", i.e., different models for what information needs to be configured in what entities. Once configured, the provisioning information is distributed by a "discovery process", and once the information is discovered, the signaling protocol is automatically invoked to set up the required pseudowires. The semantics of the endpoint identifiers which the signaling protocol uses for a particular type of L2VPN are determined by the provisioning model. This document specifies a number of PPVPN provisioning models, and specifies the semantic structure of the endpoint identifiers required. It further specifies how the endpoint identifiers are carried in the "Generalized Identifier FEC" of [PWE3-CONTROL]. It is believed that the specified identifiers can also be carried within the L2TP signaling protocol, though this is currently not specified.

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

Section 2 provides an overview of the relevant aspects of [PWE3-CONTROL].

Section 5 details various provisioning models and relates them to the signaling process (using the Generalized Identifier FEC) and to the discovery process.

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.

(Sections 3 and 4 don't exist, but section 5 retains the same section number it had in previous versions as it has been referenced by number during working group discussions.)

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. Signaling Protocol Framework

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. In [PWE3-CONTROL], the term "Attachment Identifier", or "AI", to refer to a quantity whose purpose is to identify a Forwarder.

[PWE3-CONTROL] specifies two FEC elements which can be used for when setting up pseudowires, the PWid FEC element, and the Generalized Id FEC element. The PWid FEC element carries only one Forwarder

identifier; it can be thus be used only when both forwarders have the same identifier, and when that identifier can be coded as a 32-bit quantity. The Generalized Id FEC element carries two Forwarder identifiers, one for each of the two Forwarders being connected. (The concept of carrying a local identifier and a remote identifier is also used in the L2TP signaling proposal described in [LUO-L2TP].) Each identifier is known as an Attachment Identifier, and a signaling message carries both a "Source Attachment Identifier" (SAI) and a "Target Attachment Identifier" (TAI).

The Generalized ID FEC element also provides some additional structuring of the identifiers. It is assumed that the SAI and TAI will sometimes have a common part, called the "Attachment Group Identifier" (AGI), such that the SAI and TAI can each be thought of as the concatenation of the AGI with an "Attachment Individual Identifier" (AII). So the pair of identifiers is encoded into three fields: AGI, Source AII (SAII), and Target AII (TAII). The SAI is the concatenation of the AGI and the SAII, while the TAI is the concatenation of the AGI and the TAII. This additional structuing differs from the proposal in [LUO-L2TP], which specifies only that there be two "endpoint identifiers".

It should be noted that while different forwarders support different applications, the type of application (e.g., VPLS vs. VPWS) cannot necessarily be inferred from the forwarders' identifiers. A router receiving a signaling message with a particular TAI will have to be able to determine which of its local forwarders is identified by that TAI, and to determine the application provided by that forwarder. But other nodes may not be able to infer the application simply by inspection of the signaling messages.

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.

2.3. Attachment Identifiers and Forwarders

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).

IT MUST BE UNDERSTOOD THAT THIS NOTION OF "GROUP" HAS NOTHING WHATSOEVER TO DO WITH THE "GROUP ID" THAT IS PART OF THE PWID FEC IN [PWE3-CONTROL].

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".

3.

4.

5. Applications

In this section, we specify the way in which the pseudowire signaling using the Generalized ID FEC Element is 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.

Note that if the local name and the remote name are the same, the PWid FEC element can be used instead of the Generalized ID FEC element.

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 [L2VPN-REQ, VPLS], 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 [L2VPN-REQ, VPLS] 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. Radius-based auto-discovery

[RADIUS-L2TP-VPLS] includes a proposal for using RADIUS-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 an alphanumeric name, the first byte of the AGI field (immediately following the length) will be 0x90. This distinguishes it from any RD. The alphanumeric 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. Radius-based Auto-Discovery

The use of Radius-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:

```
U-PE A-----|             |----U-PE C
           |             |
           |             |
         N-PE E--------N-PE F
           |             |
           |             |
U-PE B-----|             |-----U-PE D
```

where the four U-PEs are in a common VPLS. We now illustrate how PWs get spliced together in the above topology in order to establish the necessary PWs from U-PE A to the other U-PEs.

There are three PWs from A to E. Call these A-E/1, A-E/2, and A-E/3. In order to connect A properly to the other U-PEs, there must be two PWs from E to F (call these E-F/1 and E-F/2), one PW 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.

Similarly, there are additional PWs which must get spliced together to properly interconnect U-PE B with U-PEs C and D, and to interconnect U-PE C with U-PE 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, relative to the particular VPLS and the particular N-PE, of its U-PEs. 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 E 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 i 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. (A PE providing non-distributed VPLS should therefore treat AII values of 1 the same as it treats AII values of null.)

5.5.4. Inter-Provider Application of Dist. VPLS Signaling

Consider the following topology:

```
   PE A ---- Network 1 ----- Border ----- Border ----- Network 2 ---- PE B
                             Router 12    Router 21       |
                                                          |
                                                         PE C
```

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. Security Considerations

Each L2VPN service has its own set of security considerations, and each signaling mechanism used for L2VPN has its own set of security considerations. The semantics of the identifiers used in the signaling protocols does not pose any additional security considerations. The choice of provisioning model also does not impose any security considerations, except insofar as the associated auto-discovery and signaling procedures may have security considerations.

7. Acknowledgments

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

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

Thanks to Vach Kompella and Wei Luo for a continuing discussion of the proper semantics of the generalized identifiers.

8. References

[BGP-AUTO] "Using BGP as an Auto-Discovery Mechanism for Network-based VPNs", Ould-Brahim et. al., draft-ietf-ppvpn-bgpvpn-auto-03.txt, Aug 2002.

[BGP-SIGNALING] "Layer 2 VPNs over Tunnels", Kompella et. al., draft-kompella-ppvpn-l2vpn-03.txt, Apr 2003

[RADIUS-L2TP-VPLS] "Radius/L2TP Based VPLS", Heinanen, draft-heinanen-radius-l2tp-vpls-00.txt, Feb 2003

[L2VPN-FW] "PPVPN L2 Framework", Andersson et. al., draft-ietf-ppvpn-l2-framework-03.txt, Mar 2003

[L2VPN-REQ] "Service Requirements for Layer 2 Provider Provisioned Virtual Private Network Services", Augustyn, Serbest, et. al., draft-augustyn-ppvpn-l2vpn-requirements-02.txt, Feb 2003

[L2VPN-TERM] "PPVPN Terminology", Andersson, Madsen, draft-andersson-ppvpn-terminology-02.txt, Nov 2002

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

[LUO-L2TP] "L2VPN Signaling Using L2TPv3", Luo, draft-luo-l2tpext-l2vpn- signaling-01.txt, Feb 2003

[PWE3-ARCH] "PWE3 Architecture", Bryant, Pate, et. al., draft-ietf-pwe3-arch-02.txt, Feb 2003

[PWE3-CONTROL] "Pseudowire Setup and Maintenance using LDP", Martini, et. al., draft-ietf-pwe3-control-protocol-02.txt, Feb. 2003

[PWE3-FR] "Framework for Pseudo Wire Emulation Edge-to-Edge ", draft-ietf=pwe3-framework-01.txt, May 2002

[RFC2547bis], "BGP/MPLS VPNs", Rosen, Rekhter, et. al., draft-ietf-ppvpn-rfc2547bis-04.txt, Apr 2003

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

[RFC3036] "LDP Specification", January 2001

[VPLS] "Transparent VLAN Services over MPLS", Laserre, et. al., draft-lasserre-vkompella-ppvpn-vpls-04.txt, Mar 2003

9. Author's Information

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

E-mail: erosen@cisco.com