Network Working Group
Internet-Draft
Expires: July 1, 2006

K. Kompella, Ed.
Y. Rekhter, Ed.
Juniper Networks
December 28, 2005

Virtual Private LAN Service
draft-ietf-l2vpn-vpls-bgp-06

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

This Internet-Draft will expire on July 1, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Virtual Private LAN (Local Area Network) Service (VPLS), also known as Transparent LAN Service, and Virtual Private Switched Network service, is a useful Service Provider offering. The service offers a Layer 2 Virtual Private Network (VPN); however, in the case of VPLS, the customers in the VPN are connected by a multipoint Ethernet LAN, in contrast to the usual Layer 2 VPNs, which are point-to-point in

nature.

This document describes the functions required to offer VPLS, a mechanism for signaling a VPLS, and rules for forwarding VPLS frames across a packet switched network.

Table of Contents

1. Introduction 4
1.1. Scope of this Document 4
1.2. Conventions used in this document 5
1.3. Changes from version 05 to 06 5
1.4. Changes from version 04 to 05 5
1.5. Changes from version 03 to 04 6
2. Functional Model 7
2.1. Terminology 7
2.2. Assumptions 8
2.3. Interactions 8
3. Control Plane 10
3.1. Autodiscovery 10
3.1.1. Functions 10
3.1.2. Protocol Specification 11
3.2. Signaling 11
3.2.1. Label Blocks 12
3.2.2. VPLS BGP NLRI 12
3.2.3. PW Setup and Teardown 13
3.2.4. Signaling PE Capabilities 14
3.3. BGP VPLS Operation 15
3.4. Multi-AS VPLS 16
3.4.1. a) VPLS-to-VPLS connections at the ASBRs. 17
3.4.2. b) EBGP redistribution of VPLS information between ASBRs. 17
3.4.3. c) Multi-hop EBGP redistribution of VPLS information between ASes. 18
3.4.4. Allocation of VE IDs Across Multiple ASes 19
3.5. Multi-homing and Path Selection 19
3.6. Hierarchical BGP VPLS 20
4. Data Plane 22
4.1. Encapsulation 22
4.2. Forwarding 22
4.2.1. MAC address learning 22

4.2.2. Aging 23
4.2.3. Flooding 23
4.2.4. Broadcast and Multicast 23
4.2.5. "Split Horizon" Forwarding 24
4.2.6. Qualified and Unqualified Learning 24
4.2.7. Class of Service 25
5. Deployment Options 26
6. Security Considerations 27

7. IANA Considerations 28
8. References 29
8.1. Normative References 29
8.2. Informative References 29
Appendix A. Contributors 31
Appendix B. Acknowledgements 32
Authors' Addresses 33
Intellectual Property and Copyright Statements 34

1. Introduction

Virtual Private LAN Service (VPLS), also known as Transparent LAN Service, and Virtual Private Switched Network service, is a useful service offering. A Virtual Private LAN appears in (almost) all respects as an Ethernet LAN to customers of a Service Provider. However, in a VPLS, the customers are not all connected to a single LAN; the customers may be spread across a metro or wide area. In essence, a VPLS glues together several individual LANs across a packet-switched network to appear and function as a single LAN ([7]). This is accomplished by incorporating MAC address learning, flooding and forwarding functions in the context of pseudowires that connect these individual LANs across the packet-switched network.

This document details the functions needed to offer VPLS, and then goes on to describe a mechanism for the autodiscovery of the endpoints of a VPLS as well as for signaling a VPLS. It also describes how VPLS frames are transported over tunnels across a packet switched network. The autodiscovery and signaling mechanism uses BGP as the control plane protocol. This document also briefly discusses deployment options, in particular, the notion of decoupling functions across devices.

Alternative approaches include: [13], which allows one to build a Layer 2 VPN with Ethernet as the interconnect; and [12]), which allows one to set up an Ethernet connection across a packet-switched network. Both of these, however, offer point-to-point Ethernet services. What distinguishes VPLS from the above two is that a VPLS

offers a multipoint service. A mechanism for setting up pseudowires for VPLS using the Label Distribution Protocol (LDP) is defined in [8].

1.1. Scope of this Document

This document has four major parts: defining a VPLS functional model; defining a control plane for setting up VPLS; defining the data plane for VPLS (encapsulation and forwarding of data); and defining various deployment options.

The functional model underlying VPLS is laid out in Section 2. This describes the service being offered, the network components that interact to provide the service, and at a high level their interactions.

The control plane described in this document uses Multiprotocol BGP [3] to establish VPLS service, i.e., for the autodiscovery of VPLS members and for the setup and teardown of the pseudowires that constitute a given VPLS instance. Section 3 focuses on this, and

also describes how a VPLS that spans Autonomous System boundaries is set up, as well as how multi-homing is handled. Using BGP as the control plane for VPNs is not new (see [13], [10] and [9]): what is described here is based on the mechanisms proposed in [10].

The forwarding plane and the actions that a participating Provider Edge (PE) router offering the VPLS service must take is described in Section 4.

In Section 5, the notion of 'decoupled' operation is defined, and the interaction of decoupled and non-decoupled PEs is described. Decoupling allows for more flexible deployment of VPLS.

1.2. Conventions used in this document

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 ([1]).

1.3. Changes from version 05 to 06

[NOTE to RFC Editor: this section is to be removed before

publication.]

Changes in response to GenART review.

Updated Abstract and Introduction to make it clear that VPLS is an Ethernet-based service.

Added sections on Aging, Broadcast and Multicast, Qualified and Unqualified learning and CoS. Also added a section on scaling the BGP control plane. These were requested for consistency between the BGP and LDP VPLS documents.

Added a section clarifying the concepts of label blocks, why they are necessary and how they are used.

For multi-AS operation, added a short introduction to the three options, comparing their usage.

Lots of clean-up: consistent usage of terms, expansion of acronyms before use, references.

1.4. Changes from version 04 to 05

[NOTE to RFC Editor: this section is to be removed before publication.]

Updated IANA section to reflect agreement with authors of [9] that the two docs should use the same AFI for L2VPN information.

Addressed comments received from Alex Zinin. No technical changes, but a more complete description to cover the issues that Alex raised:

1. encoding of BGP NEXT_HOP for the new AFI/SAFI is not described
2. VE ID, Block offset, Block size, Label base are not described anywhere
3. no information on how the receiving PE choose the PW label
4. section 3.2.2 talks about PE capabilities all of a sudden and introduces a L2 Info Community, whose fields and use are not described

Changes to address these:

1. Broke up section 3.2.1 into "Concepts" and "PW Setup".

2. Expanded section on "Signaling PE Capabilities".

3. Added a new section 3.3 "BGP VPLS Operation".

4. Minor tweaking, e.g. to fix section number references.

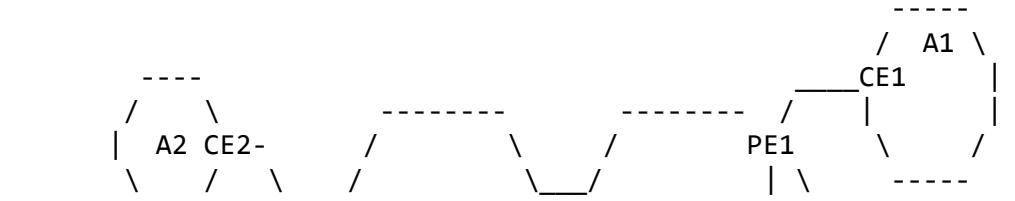
1.5. Changes from version 03 to 04

[NOTE to RFC Editor: this section is to be removed before publication.]

Incorporated IDR review comments from Eric Ji, Chaitanya Kodeboyina, and Mike Loomis. Most changes are clarifications and rewording for better readability. The substantive changes are to remove several flags from the control field.

2. Functional Model

This will be described with reference to the following figure.



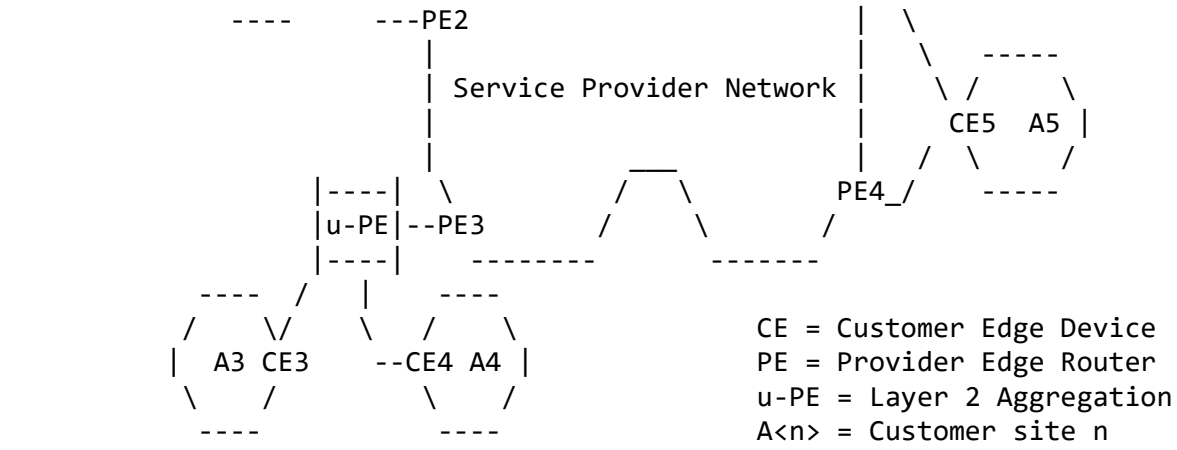


Figure 1: Example of a VPLS

2.1. Terminology

Terminology similar to that in [10] is used: a Service Provider (SP) network with P (Provider-only) and PE (Provider Edge) routers, and customers with CE (Customer Edge) devices. Here, however, there is an additional concept, that of a "u-PE", a Layer 2 PE device used for Layer 2 aggregation. The notion of u-PE is described further in Section 5. PE and u-PE devices are "VPLS-aware", which means that they know that a VPLS service is being offered. We will call these VPLS edge devices, which could be either a PE or an u-PE, a VE.

In contrast, the CE device (which may be owned and operated by either the SP or the customer) is VPLS-unaware; as far as the CE is concerned, it is connected to the other CEs in the VPLS via a Layer 2 switched network. This means that there should be no changes to a CE device, either to the hardware or the software, in order to offer VPLS.

A CE device may be connected to a PE or a u-PE via Layer 2 switches that are VPLS-unaware. From a VPLS point of view, such Layer 2 switches are invisible, and hence will not be discussed further. Furthermore, a u-PE may be connected to a PE via Layer 2 and Layer 3

devices; this will be discussed further in a later section.

The term "demultiplexor" refers to an identifier in a data packet that identifies both the VPLS to which the packet belongs as well as

the ingress PE. In this document, the demultiplexor is an MPLS label.

The term "VPLS" will refer to the service as well as a particular instantiation of the service (i.e., an emulated LAN); it should be clear from the context which usage is intended.

2.2. Assumptions

The Service Provider Network is a packet switched network. The PEs are assumed to be (logically) fully meshed with tunnels over which packets that belong to a service (such as VPLS) are encapsulated and forwarded. These tunnels can be IP tunnels, such as GRE, or MPLS tunnels, established by RSVP-TE or LDP. These tunnels are established independently of the services offered over them; the signaling and establishment of these tunnels are not discussed in this document.

"Flooding" and MAC address "learning" (see Section 4) are an integral part of VPLS. However, these activities are private to an SP device, i.e., in the VPLS described below, no SP device requests another SP device to flood packets or learn MAC addresses on its behalf.

All the PEs participating in a VPLS are assumed to be fully meshed in the data plane, i.e., there is a bidirectional pseudowire between every pair of PEs participating in that VPLS, and thus every (ingress) PE can send a VPLS packet to the egress PE(s) directly, without the need for an intermediate PE (see Section 4.2.5.) This requires that VPLS PEs are logically fully meshed in the control plane so that a PE can send a message to another PE to set up the necessary pseudowires. See Section 3.6 for a discussion on alternatives to achieve a logical full mesh in the control plane.

2.3. Interactions

VPLS is a "LAN Service" in that CE devices that belong to VPLS V can interact through the SP network as if they were connected by a LAN. VPLS is "private" in that CE devices that belong to different VPLSs cannot interact. VPLS is "virtual" in that multiple VPLSs can be offered over a common packet switched network.

PE devices interact to "discover" all the other PEs participating in the same VPLS, and to exchange demultiplexors. These interactions are control-driven, not data-driven.

Dell et al., Ex. 1018
Page 9 of 39

u-PEs interact with PEs to establish connections with remote PEs or u-PEs in the same VPLS. This interaction is control-driven.

PE devices can participate simultaneously in both VPLS and IP VPNs ([10]). These are independent services, and the information exchanged for each type of service is kept separate as the Network Layer Reachability Information (NLRI) used for this exchange have different Address Family Identifiers (AFI) and Subsequent Address Family Identifiers (SAFI). Consequently, an implementation MUST maintain a separate routing storage for each service. However, multiple services can use the same underlying tunnels; the VPLS or VPN label is used to demultiplex the packets belonging to different services.

3. Control Plane

There are two primary functions of the VPLS control plane: autodiscovery, and setup and teardown of the pseudowires that constitute the VPLS, often called signaling. Section 3.1 and Section 3.2 describe these functions. Both of these functions are accomplished with a single BGP Update advertisement; Section 3.3 describes how this is done by detailing BGP protocol operation for VPLS. Section 3.4 describes the setting up of pseudowires that span Autonomous Systems. Section 3.5 describes how multi-homing is handled.

3.1. Autodiscovery

Discovery refers to the process of finding all the PEs that participate in a given VPLS instance. A PE can either be configured with the identities of all the other PEs in a given VPLS, or the PE can use some protocol to discover the other PEs. The latter is called autodiscovery.

The former approach is fairly configuration-intensive, especially since it is required that the PEs participating in a given VPLS are fully meshed (i.e., that every PE in a given VPLS establish pseudowires to every other PE in that VPLS). Furthermore, when the topology of a VPLS changes (i.e., a PE is added to, or removed from the VPLS), the VPLS configuration on all PEs in that VPLS must be changed.

In the autodiscovery approach, each PE "discovers" which other PEs are part of a given VPLS by means of some protocol, in this case BGP. This allows each PE's configuration to consist only of the identity of the VPLS instance established on this PE, not the identity of every other PE in that VPLS instance -- that is auto-discovered. Moreover, when the topology of a VPLS changes, only the affected PE's configuration changes; other PEs automatically find out about the change and adapt.

3.1.1. Functions

A PE that participates in a given VPLS instance V must be able to tell all other PEs in VPLS V that it is also a member of V. A PE must also have a means of declaring that it no longer participates in a

VPLS. To do both of these, the PE must have a means of identifying a VPLS and a means by which to communicate to all other PEs.

U-PE devices also need to know what constitutes a given VPLS; however, they don't need the same level of detail. The PE (or PEs) to which a u-PE is connected gives the u-PE an abstraction of the

VPLS; this is described in section 5.

3.1.2. Protocol Specification

The specific mechanism for autodiscovery described here is based on [13] and [10]; it uses BGP extended communities [4] to identify members of a VPLS, in particular, the Route Target community, whose format is described in [4]. The semantics of the use of Route Targets is described in [10]; their use in VPLS is identical.

As it has been assumed that VPLSs are fully meshed, a single Route Target RT suffices for a given VPLS V, and in effect that RT is the identifier for VPLS V.

A PE announces (typically via I-BGP) that it belongs to VPLS V by annotating its NLRIs for V (see next subsection) with Route Target RT, and acts on this by accepting NLRIs from other PEs that have Route Target RT. A PE announces that it no longer participates in V by withdrawing all NLRIs that it had advertised with Route Target RT.

3.2. Signaling

Once discovery is done, each pair of PEs in a VPLS must be able to establish (and tear down) pseudowires to each other, i.e., exchange (and withdraw) demultiplexors. This process is known as signaling. Signaling is also used to transmit certain characteristics of the pseudowires that a PE sets up for a given VPLS.

Recall that a demultiplexor is used to distinguish among several different streams of traffic carried over a tunnel, each stream possibly representing a different service. In the case of VPLS, the demultiplexor not only says to which specific VPLS a packet belongs, but also identifies the ingress PE. The former information is used for forwarding the packet; the latter information is used for learning MAC addresses. The demultiplexor described here is an MPLS label. However, note that the PE-to-PE tunnels need not be MPLS

tunnels.

Using a distinct BGP Update message to send a demultiplexor to each remote PE would require the originating PE to send N such messages for N remote PEs. The solution described in this document allows a PE to send a single (common) Update message that contains demultiplexors for all the remote PEs, instead of N individual messages. Doing this reduces the control plane load both on the originating PE as well as on the BGP Route Reflectors that may be involved in distributing this Update to other PEs.

3.2.1. Label Blocks

To accomplish this, we introduce the notion of "label blocks". A label block, defined by a label base LB and a VE block size VBS, is a contiguous set of labels {LB, LB+1, ..., LB+VBS-1}. Here's how label blocks work. All PEs within a given VPLS are assigned unique VE IDs as part of their configuration. A PE X wishing to send a VPLS update sends the same label block information to all other PEs. Each receiving PE infers the label intended for PE X by adding their (unique) VE ID to the label base. In this manner, each receiving PE gets a unique demultiplexor for PE X for that VPLS.

This simple notion is enhanced with the concept of a VE block offset VBO. A label block defined by <LB, VBO, VBS> is the set {LB+VBO, LB+VBO+1, ..., LB+VBO+VBS-1}. Thus, instead of a single large label block to cover all VE IDs in a VPLS, one can have several label blocks, each with a different label base. This makes label block management easier, and also allows PE X to cater gracefully to a PE joining a VPLS with a VE ID that is not covered by the set of label blocks that that PE X has already advertised.

When a PE starts up, or is configured with a new VPLS instance, the BGP process may wish to wait to receive several advertisements for that VPLS instance from other PEs to improve the efficiency of label block allocation.

3.2.2. VPLS BGP NLRI

The VPLS BGP NLRI described below, with a new AFI and SAFI (see [3]) is used to exchange VPLS membership and demultiplexors.

A VPLS BGP NLRI has the following information elements: a VE ID, a VE Block Offset, a VE Block Size and a label base. The format of the VPLS NLRI is given below. The AFI is the L2VPN AFI (to be assigned by IANA), and the SAFI is the VPLS SAFI (65). The Length field is in octets.

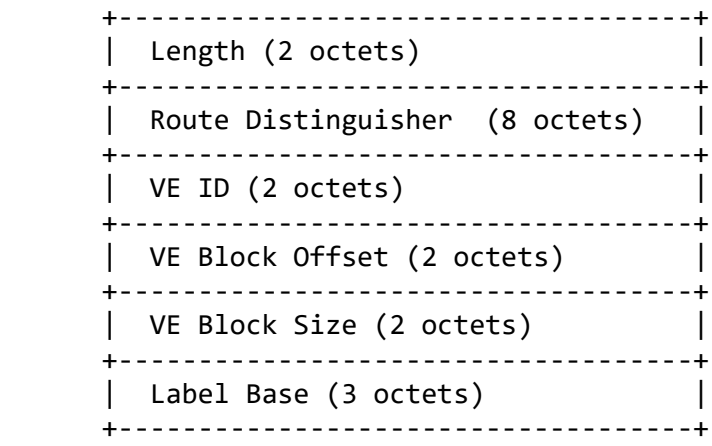


Figure 2: BGP NLRI for VPLS Information

A PE participating in a VPLS must have at least one VE ID. If the PE is the VE, it typically has one VE ID. If the PE is connected to several u-PEs, it has a distinct VE ID for each u-PE. It may additionally have a VE ID for itself, if it itself acts as a VE for that VPLS. In what follows, we will call the PE announcing the VPLS NLRI PE-a, and we will assume that PE-a owns VE ID V (either belonging to PE-a itself, or to a u-PE connected to PE-a).

VE IDs are typically assigned by the network administrator. Their scope is local to a VPLS. A given VE ID should belong to only one PE, unless a CE is multi-homed (see Section 3.5).

A label block is a set of demultiplexor labels used to reach a given VE ID. A VPLS BGP NLRI with VE ID V, VE Block Offset VBO, VE Block Size VBS and label base LB communicates to its peers the following:

label block for V: labels from LB to (LB + VBS - 1), and

remote VE set for V: from VBO to (VBO + VBS - 1).

There is a one-to-one correspondence between the remote VE set and the label block: VE ID (VBO + n) corresponds to label (LB + n).

3.2.3. PW Setup and Teardown

Suppose PE-a is part of VPLS foo, and makes an announcement with VE ID V, VE Block Offset VBO, VE Block Size VBS and label base LB. If PE-b is also part of VPLS foo, and has VE ID W, PE-b does the following:

1. checks if W is part of PE-a's 'remote VE set': if VBO <= W < VBO + VBS, then W is part of PE-a's remote VE set. If not, PE-b

ignores this message, and skips the rest of this procedure.

2. sets up a PW to PE-a: the demultiplexor label to send traffic from PE-b to PE-a is computed as (LB + W - VBO).

3. checks if V is part of any 'remote VE set' that PE-b announced, i.e., PE-b checks if V belongs to some remote VE set that PE-b announced, say with VE Block Offset VBO', VE Block Size VBS' and label base LB'. If not, PE-b MUST make a new announcement as described in Section 3.3.

4. sets up a PW from PE-a: the demultiplexor label over which PE-b should expect traffic from PE-a is computed as: (LB' + V - VBO').

If Y withdraws an NLRI for V that X was using, then X MUST tear down its ends of the pseudowire between X and Y.

3.2.4. Signaling PE Capabilities

The following extended attribute, the "Layer2 Info Extended Community", is used to signal control information about the pseudowires to be setup for a given VPLS. This information includes the Encaps Type (type of encapsulation on the pseudowires), Control Flags (control information regarding the pseudowires) and the Maximum Transmission Unit (MTU) to be used on the pseudowires.

The Encaps Type for VPLS is 19.

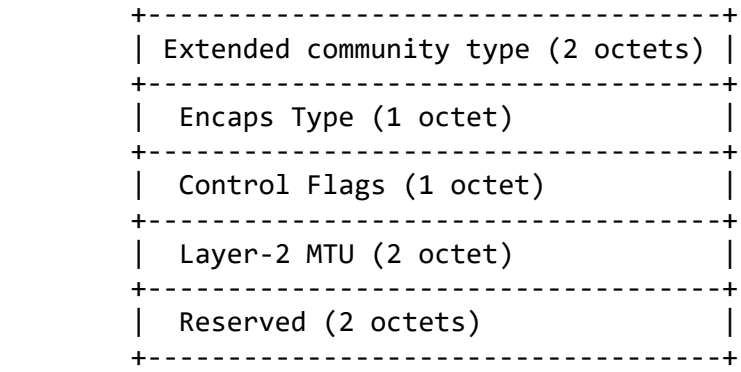


Figure 3: Layer2 Info Extended Community

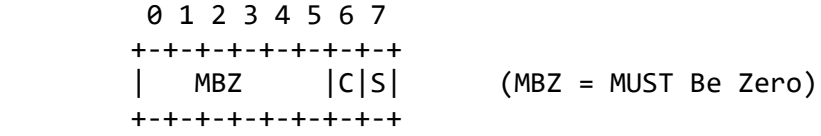


Figure 4: Control Flags Bit Vector

With reference to Figure 4, the following bits in the Control Flags are defined; the remaining bits, designated MBZ, MUST be set to zero when sending and MUST be ignored when receiving this community.

Name	Meaning
C	A Control word ([5]) MUST or MUST NOT be present when sending VPLS packets to this PE, depending on whether C is 1 or 0, respectively
S	Sequenced delivery of frames MUST or MUST NOT be used when sending VPLS packets to this PE. depending on

whether S is 1 or 0, respectively

3.3. BGP VPLS Operation

To create a new VPLS, say VPLS foo, a network administrator must pick a RT for VPLS foo, say RT-foo. This will be used by all PEs that serve VPLS foo. To configure a given PE, say PE-a, to be part of VPLS foo, the network administrator only has to choose a VE ID V for PE-a. (If PE-a is connected to u-PEs, PE-a may be configured with more than one VE ID; in that case, the following is done for each VE ID). The PE may also be configured with a Route Distinguisher (RD); if not, it generates a unique RD for VPLS foo. Say the RD is RD-foo-a. PE-a then generates an initial label block and a remote VE set for V, defined by VE Block Offset VBO, VE Block Size VBS and label base LB. These may be empty.

PE-a then creates a VPLS BGP NLRI with RD RD-foo-a, VE ID V, VE Block Offset VBO, VE Block Size VBS and label base LB. To this, it attaches a Layer2 Info Extended Community and a RT, RT-foo. It sets the BGP Next Hop for this NLRI as itself, and announces this NLRI to its peers. The Network Layer protocol associated with the Network Address of the Next Hop for the combination <AFI=L2VPN AFI, SAFI=VPLS SAFI> is IP; this association is required by [3], Section 5. If the value of the Length of the Next Hop field is 4, then the Next Hop contains an IPv4 address. If this value is 16, then the Next Hop contains an IPv6 address.

If PE-a hears from another PE, say PE-b, a VPLS BGP announcement with RT-foo and VE ID W, then PE-a knows that PE-b is a member of the same VPLS (autodiscovery). PE-a then has to set up its part of a VPLS pseudowire between PE-a and PE-b, using the mechanisms in Section 3.2. Similarly, PE-b will have discovered that PE-a is in the same VPLS, and PE-b must set up its part of the VPLS pseudowire. Thus, signaling and pseudowire setup is also achieved with the same Update message.

If W is not in any remote VE set that PE-a announced for VE ID V in VPLS foo, PE-b will not be able to set up its part of the pseudowire to PE-a. To address this, PE-a can choose to withdraw the old announcement(s) it made for VPLS foo, and announce a new Update with a larger remote VE set and corresponding label block that covers all VE IDs that are in VPLS foo. This however, may cause some service

disruption. An alternative for PE-a is to create a new remote VE set and corresponding label block, and announce them in a new Update, without withdrawing previous announcements.

If PE-a's configuration is changed to remove VE ID V from VPLS foo, then PE-a MUST withdraw all its announcements for VPLS foo that contain VE ID V. If all of PE-a's links to its CEs in VPLS foo go down, then PE-a SHOULD either withdraw all its NLRIs for VPLS foo, or let other PEs in the VPLS foo know in some way that PE-a is no longer connected to its CEs.

3.4. Multi-AS VPLS

As in [13] and [10], the above autodiscovery and signaling functions are typically announced via I-BGP. This assumes that all the sites in a VPLS are connected to PEs in a single Autonomous System (AS).

However, sites in a VPLS may connect to PEs in different ASes. This leads to two issues: 1) there would not be an I-BGP connection between those PEs, so some means of signaling across ASes is needed; and 2) there may not be PE-to-PE tunnels between the ASes.

A similar problem is solved in [10], Section 10. Three methods are suggested to address issue (1); all these methods have analogs in multi-AS VPLS.

Here is a diagram for reference:

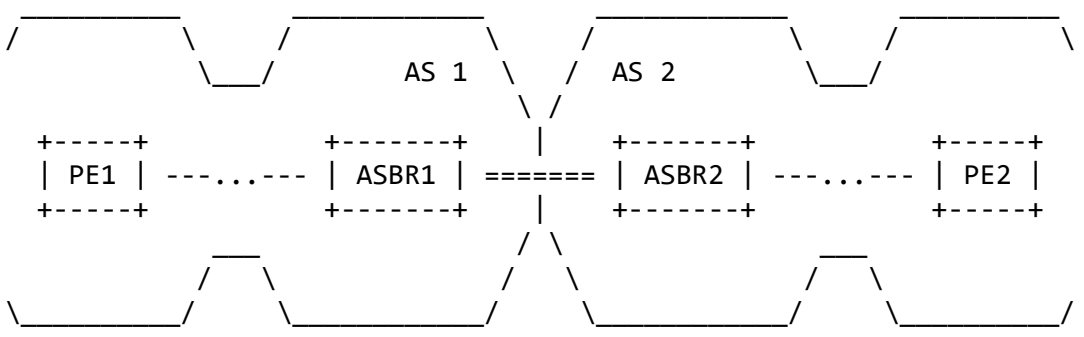


Figure 6: Inter-AS VPLS

As in the above reference, three methods for signaling inter-provider VPLS are given; these are presented in order of increasing scalability. Method (a) is the easiest to understand conceptually, and the easiest to deploy; however, it requires an Ethernet interconnect between the ASes, and both VPLS control and data plane state on the AS border routers (ASBRs). Method (b) requires VPLS control plane state on the ASBRs and MPLS on the AS-AS interconnect (which need not be Ethernet). Method (c) requires MPLS on the AS-AS interconnect, but no VPLS state of any kind on the ASBRs.

3.4.1. a) VPLS-to-VPLS connections at the ASBRs.

In this method, an AS Border Router (ASBR1) acts as a PE for all VPLSs that span AS1 and an AS to which ASBR1 is connected, such as AS2 here. The ASBR on the neighboring AS (ASBR2) is viewed by ASBR1 as a CE for the VPLSs that span AS1 and AS2; similarly, ASBR2 acts as a PE for this VPLS from AS2's point of view, and views ASBR1 as a CE.

This method does not require MPLS on the ASBR1-ASBR2 link, but does require that this link carry Ethernet traffic, and that there be a separate VLAN sub-interface for each VPLS traversing this link. It further requires that ASBR1 does the PE operations (discovery, signaling, MAC address learning, flooding, encapsulation, etc.) for all VPLSs that traverse ASBR1. This imposes a significant burden on ASBR1, both on the control plane and the data plane, which limits the number of multi-AS VPLSs.

Note that in general, there will be multiple connections between a pair of ASes, for redundancy. In this case, the Spanning Tree Protocol (STP) ([14]), or some other means of loop detection and prevention, must be run on each VPLS that spans these ASes, so that a loop-free topology can be constructed in each VPLS. This imposes a further burden on the ASBRs and PEs participating in those VPLSs, as these devices would need to run a loop detection algorithm for each such VPLS. How this may be achieved is outside the scope of this document.

3.4.2. b) EBGP redistribution of VPLS information between ASBRs.

This method requires I-BGP peerings between the PEs in AS1 and ASBR1 in AS1 (perhaps via route reflectors), an E-BGP peering between ASBR1 and ASBR2 in AS2, and I-BGP peerings between ASBR2 and the PEs in AS2. In the above example, PE1 sends a VPLS NLRI to ASBR1 with a label block and itself as the BGP nexthop; ASBR1 sends the NLRI to ASBR2 with new labels and itself as the BGP nexthop; and ASBR2 sends the NLRI to PE2 with new labels and itself as the nexthop.

The VPLS NLRI that ASBR1 sends to ASBR2 (and the NLRI that ASBR2

sends to PE2) is identical to the VPLS NLRI that PE1 sends to ASBR1, except for the label block. To be precise, the Length, the Route Distinguisher, the VE ID, the VE Block Offset, and the VE Block Size MUST be the same; the Label Base may be different. Furthermore, ASBR1 must also update its forwarding path as follows: if the Label Base sent by PE1 is L1, the Label-block Size is N, the Label Base sent by ASBR1 is L2, and the tunnel label from ASBR1 to PE1 is T, then ASBR1 must install the following in the forwarding path:

swap L2 with L1 and push T,

swap L2+1 with L1+1 and push T, ...

swap L2+N-1 with L1+N-1 and push T.

ASBR2 must act similarly, except that it may not need a tunnel label if it is directly connected with ASBR1.

When PE2 wants to send a VPLS packet to PE1, PE2 uses its VE ID to get the right VPLS label from ASBR2's label block for PE1, and uses a tunnel label to reach ASBR2. ASBR2 swaps the VPLS label with the label from ASBR1; ASBR1 then swaps the VPLS label with the label from PE1, and pushes a tunnel label to reach PE1.

In this method, one needs MPLS on the ASBR1-ASBR2 interface, but there is no requirement that the link layer be Ethernet. Furthermore, the ASBRs take part in distributing VPLS information. However, the data plane requirements of the ASBRs is much simpler than in method (a), being limited to label operations. Finally, the construction of loop-free VPLS topologies is done by routing decisions, viz. BGP path and nexthop selection, so there is no need to run the Spanning Tree Protocol on a per-VPLS basis. Thus, this method is considerably more scalable than method (a).

3.4.3. c) Multi-hop EBGP redistribution of VPLS information between ASes.

In this method, there is a multi-hop E-BGP peering between the PEs (or preferably, a Route Reflector) in AS1 and the PEs (or Route Reflector) in AS2. PE1 sends a VPLS NLRI with labels and nexthop self to PE2; if this is via route reflectors, the BGP nexthop is not changed. This requires that there be a tunnel LSP from PE1 to PE2. This tunnel LSP can be created exactly as in [10], section 10 (c), for example using E-BGP to exchange labeled IPv4 routes for the PE

loopbacks.

When PE1 wants to send a VPLS packet to PE2, it pushes the VPLS label corresponding to its own VE ID onto the packet. It then pushes the

tunnel label(s) to reach PE2.

This method requires no VPLS information (in either the control or the data plane) on the ASBRs. The ASBRs only need to set up PE-to-PE tunnel LSPs in the control plane, and do label operations in the data plane. Again, as in the case of method (b), the construction of loop-free VPLS topologies is done by routing decisions, i.e., BGP path and nexthop selection, so there is no need to run the Spanning Tree Protocol on a per-VPLS basis. This option is likely to be the most scalable of the three methods presented here.

3.4.4. Allocation of VE IDs Across Multiple ASes

In order to ease the allocation of VE IDs for a VPLS that spans multiple ASes, one can allocate ranges for each AS. For example, AS1 uses VE IDs in the range 1 to 100, AS2 from 101 to 200, etc. If there are 10 sites attached to AS1 and 20 to AS2, the allocated VE IDs could be 1-10 and 101 to 120. This minimizes the number of VPLS NLRIs that are exchanged while ensuring that VE IDs are kept unique.

In the above example, if AS1 needed more than 100 sites, then another range can be allocated to AS1. The only caveat is that there be no overlap between VE ID ranges among ASes. The exception to this rule is multi-homing, which is dealt with below.

3.5. Multi-homing and Path Selection

It is often desired to multi-home a VPLS site, i.e., to connect it to multiple PEs, perhaps even in different ASes. In such a case, the PEs connected to the same site can either be configured with the same VE ID or with different VE IDs. In the latter case, it is mandatory to run STP on the CE device, and possibly on the PEs, to construct a loop-free VPLS topology. How this can be accomplished is outside the scope of this document; however, the rest of this section will describe in some detail the former case.

In the case where the PEs connected to the same site are assigned the same VE ID, a loop-free topology is constructed by routing

mechanisms, in particular, by BGP path selection. When a BGP speaker receives two equivalent NLRIs (see below for the definition), it applies standard path selection criteria such as Local Preference and AS Path Length to determine which NLRI to choose; it MUST pick only one. If the chosen NLRI is subsequently withdrawn, the BGP speaker applies path selection to the remaining equivalent VPLS NLRIs to pick another; if none remain, the forwarding information associated with that NLRI is removed.

Two VPLS NLRIs are considered equivalent from a path selection point

of view if the Route Distinguisher, the VE ID and the VE Block Offset are the same. If two PEs are assigned the same VE ID in a given VPLS, they MUST use the same Route Distinguisher, and they SHOULD announce the same VE Block Size for a given VE Offset.

3.6. Hierarchical BGP VPLS

This section discusses how one can scale the VPLS control plane when using BGP. There are at least three aspects of scaling the control plane:

1. alleviating the full mesh connectivity requirement among VPLS BGP speakers;

2. limiting BGP VPLS message passing to just the interested speakers rather than all BGP speakers; and

3. simplifying the addition and deletion of BGP speakers, whether for VPLS or other applications.

Fortunately, the use of BGP for Internet routing as well as for IP VPNs has yielded several good solutions for all these problems. The basic technique is hierarchy, using BGP Route Reflectors (RRs) ([6]). The idea is to designate a small set of Route Reflectors which are themselves fully meshed, and then establish a BGP session between each BGP speaker and one or more RRs. In this way, there is no need of direct full mesh connectivity among all the BGP speakers. If the particular scaling needs of a provider requires a large number of RRs, then this technique can be applied recursively: the full mesh connectivity among the RRs can be brokered by yet another level of RRs. The use of RRs solves problems 1 and 3 above.

It is important to note that RRs, as used for VPLS and VPNs, are purely a control plane technique. The use of RRs introduces no data plane state and no data plane forwarding requirements on the RRs, and does not in any way change the forwarding path of VPLS traffic. This is in contrast to the technique of Hierarchical VPLS defined in [8].

Another consequence of this approach is that it is not required that one set of RRs handles all BGP messages, or that a particular RR handle all messages from a given PE. One can define several sets of RRs, for example a set to handle VPLS, another to handle IP VPNs and another for Internet routing. Another partitioning could be to have some subset of VPLSs and IP VPNs handled by one set of RRs, and another subset of VPLSs and IP VPNs handled by another set of RRs; the use of Route Target Filtering (RTF), described in [11] can make this simpler and more effective.

Finally, problem 2 (that of limiting BGP VPLS message passing to just the interested BGP speakers) is addressed by the use of RTF. This technique is orthogonal to the use of RRs, but works well in conjunction with RRs. RTF is also very effective in inter-AS VPLS; more details on how RTF works and its benefits are provided in [11].

It is worth mentioning an aspect of the control plane that is often a source of confusion. No MAC addresses are exchanged via BGP. All MAC address learning and aging is done in the data plane individually by each PE. The only task of BGP VPLS message exchange is autodiscovery and label exchange.

Thus, BGP processing for VPLS occurs when

1. a PE joins or leaves a VPLS; or

2. a failure occurs in the network, bringing down a PE-PE tunnel or a PE-CE link.

These events are relatively rare, and typically, each such event causes one BGP update to be generated. Coupled with BGP's messaging efficiency when used for signaling VPLS, these observations lead to the conclusion that BGP as a control plane for VPLS will scale quite well both in terms of processing and memory requirements.

4. Data Plane

This section discusses two aspects of the data plane for PEs and u-PEs implementing VPLS: encapsulation and forwarding.

4.1. Encapsulation

Ethernet frames received from CE devices are encapsulated for transmission over the packet switched network connecting the PEs. The encapsulation is as in [5], with one change: a PE that sets the P bit in the Control Flags strips the outermost VLAN from an Ethernet frame received from a CE before encapsulating it, and pushes a VLAN onto a decapsulated frame before sending it to a CE.

4.2. Forwarding

VPLS packets are classified as belonging to a given service instance and associated forwarding table based on the interface over which the packet is received. Packets are forwarded in the context of the service instance based on the destination MAC address. The former

mapping is determined by configuration. The latter is the focus of this section.

4.2.1. MAC address learning

As was mentioned earlier, the key distinguishing feature of VPLS is that it is a multipoint service. This means that the entire Service Provider network should appear as a single logical learning bridge for each VPLS that the SP network supports. The logical ports for the SP "bridge" are the customer ports as well as the pseudowires on a VE. Just as a learning bridge learns MAC addresses on its ports, the SP bridge must learn MAC addresses at its VEs.

Learning consists of associating source MAC addresses of packets with the (logical) ports on which they arrive; this association is the Forwarding Information Base (FIB). The FIB is used for forwarding packets. For example, suppose the bridge receives a packet with source MAC address S on (logical) port P. If subsequently, the bridge receives a packet with destination MAC address S, it knows that it should send the packet out on port P.

If a VE learns a source MAC address S on logical port P, then later sees S on a different port P', then the VE MUST update its FIB to reflect the new port P'. A VE MAY implement a mechanism to damp flapping of source ports for a given MAC address.

4.2.2. Aging

VPLS PEs SHOULD have an aging mechanism to remove a MAC address associated with a logical port, much the same as learning bridges do. This is required so that a MAC address can be relearned if it "moves" from a logical port to another logical port, either because the station to which that MAC address belongs really has moved, or because of a topology change in the LAN that causes this MAC address to arrive on a new port. In addition, aging reduces the size of a VPLS MAC table to just the active MAC addresses, rather than all MAC addresses in that VPLS.

The "age" of a source MAC address S on a logical port P is the time since it was last seen as a source MAC on port P. If the age exceeds

the aging time T, S MUST be flushed from the FIB. This of course means that every time S is seen as a source MAC address on port P, S's age is reset.

An implementation SHOULD provide a configurable knob to set the aging time T on a per-VPLS basis. In addition, an implementation MAY accelerate aging of all MAC addresses in a VPLS if it detects certain situations, such as a Spanning Tree topology change in that VPLS.

4.2.3. Flooding

When a bridge receives a packet to a destination that is not in its FIB, it floods the packet on all the other ports. Similarly, a VE will flood packets to an unknown destination to all other VEs in the VPLS.

In Figure 1 above, if CE2 sent an Ethernet frame to PE2, and the destination MAC address on the frame was not in PE2's FIB (for that VPLS), then PE2 would be responsible for flooding that frame to every other PE in the same VPLS. On receiving that frame, PE1 would be responsible for further flooding the frame to CE1 and CE5 (unless PE1 knew which CE "owned" that MAC address).

On the other hand, if PE3 received the frame, it could delegate further flooding of the frame to its u-PE. If PE3 was connected to 2 u-PEs, it would announce that it has two u-PEs. PE3 could either announce that it is incapable of flooding, in which case it would receive two frames, one for each u-PE, or it could announce that it is capable of flooding, in which case it would receive one copy of the frame, which it would then send to both u-PEs.

4.2.4. Broadcast and Multicast

There is a well-known broadcast MAC address. An Ethernet frame whose

destination MAC address is the broadcast MAC address must be sent to all stations in that VPLS. This can be accomplished by the same means that is used for flooding.

There is also an easily recognized set of "multicast" MAC addresses. Ethernet frames with a destination multicast MAC address MAY be broadcast to all stations; a VE MAY also use certain techniques to restrict transmission of multicast frames to a smaller set of

receivers, those that have indicated interest in the corresponding multicast group. Discussion of this is outside the scope of this document.

4.2.5. "Split Horizon" Forwarding

When a PE capable of flooding (say PEx) receives a broadcast Ethernet frame, or one with an unknown destination MAC address, it must flood the frame. If the frame arrived from an attached CE, PEx must send a copy of the frame to every other attached CE, as well as to all other PEs participating in the VPLS. If, on the other hand, the frame arrived from another PE (say PEy), PEx must send a copy of the packet only to attached CEs. PEx MUST NOT send the frame to other PEs, since PEy would have already done so. This notion has been termed "split horizon" forwarding, and is a consequence of the PEs being logically fully meshed for VPLS.

Split horizon forwarding rules apply to broadcast and multicast packets, as well as packets to an unknown MAC address.

4.2.6. Qualified and Unqualified Learning

The key for normal Ethernet MAC learning is usually just the (6-octet) MAC address. This is called "unqualified learning". However, it is also possible that the key for learning includes the VLAN tag when present; this is called "qualified learning".

In the case of VPLS, learning is done in the context of a VPLS instance, which typically corresponds to a customer. If the customer uses VLAN tags, one can make the same distinctions of qualified and unqualified learning. If the key for learning within a VPLS is just the MAC address, then this VPLS is operating under unqualified learning. If the key for learning is (customer VLAN tag + MAC address), then this VPLS is operating under qualified learning.

Choosing between qualified and unqualified learning involves several factors, the most important of which is whether one wants a single global broadcast domain (unqualified), or a broadcast domain per VLAN (qualified). The latter makes flooding and broadcasting more efficient, but requires larger MAC tables. These considerations

apply equally to normal Ethernet forwarding and to VPLS.

4.2.7. Class of Service

In order to offer different Classes of Service within a VPLS, an implementation MAY choose to map 802.1p bits in a customer Ethernet frame with a VLAN tag to an appropriate setting of EXP bits in the pseudowire and/or tunnel label, allowing for differential treatment of VPLS frames in the packet-switched network.

To be useful, an implementation SHOULD allow this mapping function to be different for each VPLS, as each VPLS customer may have their own view of the required behavior for a given setting of 802.1p bits.

Dell et al., Ex. 1018
Page 28 of 39

5. Deployment Options

In deploying a network that supports VPLS, the SP must decide what functions the VPLS-aware device closest to the customer (the VE) supports. The default case described in this document is that the VE is a PE. However, there are a number of reasons that the VE might be a device that does all the Layer 2 functions (such as MAC address learning and flooding), and a limited set of Layer 3 functions (such as communicating to its PE), but, for example, doesn't do full-fledged discovery and PE-to-PE signaling. Such a device is called a "u-PE".

As both of these cases have benefits, one would like to be able to "mix and match" these scenarios. The signaling mechanism presented here allows this. For example, in a given provider network, one PE may be directly connected to CE devices; another may be connected to u-PEs that are connected to CEs; and a third may be connected directly to a customer over some interfaces and to u-PEs over others. All these PEs perform discovery and signaling in the same manner. How they do learning and forwarding depends on whether or not there is a u-PE; however, this is a local matter, and is not signaled. However, the details of the operation of a u-PE and its interactions with PEs and other u-PEs is beyond the scope of this document.

6. Security Considerations

The focus in Virtual Private LAN Service is the privacy of data, i.e., that data in a VPLS is only distributed to other nodes in that VPLS and not to any external agent or other VPLS. Note that VPLS does not offer security or authentication: VPLS packets are sent in the clear in the packet-switched network, and a man-in-the-middle can eavesdrop, and may be able to inject packets into the data stream. If security is desired, the PE-to-PE tunnels can be IPsec tunnels. For more security, the end systems in the VPLS sites can use appropriate means of encryption to secure their data even before it enters the Service Provider network.

There are two aspects to achieving data privacy in a VPLS: securing the control plane, and protecting the forwarding path. Compromise of the control plane could result in a PE sending data belonging to some VPLS to another VPLS, or blackholing VPLS data, or even sending it to an eavesdropper, none of which are acceptable from a data privacy point of view. Since all control plane exchanges are via BGP, techniques such as in [2] help authenticate BGP messages, making it harder to spoof updates (which can be used to divert VPLS traffic to the wrong VPLS), or withdraws (denial of service attacks). In the multi-AS options (b) and (c), this also means protecting the inter-AS BGP sessions, between the ASBRs, the PEs or the Route Reflectors. Note that [2] will not help in keeping VPLS labels private -- knowing the labels, one can eavesdrop on VPLS traffic. However, this requires access to the data path within a Service Provider network.

Protecting the data plane requires ensuring that PE-to-PE tunnels are well-behaved (this is outside the scope of this document), and that VPLS labels are accepted only from valid interfaces. For a PE, valid interfaces comprise links from P routers. For an ASBR, a valid interface is a link from an ASBR in an AS that is part of a given VPLS. It is especially important in the case of multi-AS VPLSs that one accept VPLS packets only from valid interfaces.

7. IANA Considerations

IANA is asked to allocate an AFI for L2VPN information (suggested value: 25). [NOTE to IANA: This should be the same as the AFI requested by [9].]

8. References

8.1. Normative References

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

[2] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[3] Bates, T., "Multiprotocol Extensions for BGP-4", draft-ietf-idr-rfc2858bis-07 (work in progress), August 2005.

[4] Rekhter, Y., "BGP Extended Communities Attribute", draft-ietf-idr-bgp-ext-communities-09 (work in progress), July 2005.

[5] Martini, L., "Encapsulation Methods for Transport of Ethernet Over MPLS Networks", draft-ietf-pwe3-ethernet-encap-11 (work in progress), December 2005.

8.2. Informative References

[6] Bates, T., Chandra, R., and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP", RFC 2796, April 2000.

[7] Andersson, L. and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", draft-ietf-l2vpn-l2-framework-05

(work in progress), June 2004.

[8] Lasserre, M. and V. Kompella, "Virtual Private LAN Services over MPLS", draft-ietf-l2vpn-vpls-ldp-08 (work in progress), November 2005.

[9] Ould-Brahim, H., "Using BGP as an Auto-Discovery Mechanism for Layer-3 and Layer-2 VPNs", draft-ietf-l3vpn-bgpvpn-auto-06 (work in progress), June 2005.

[10] Rosen, E., "BGP/MPLS IP VPNs", draft-ietf-l3vpn-rfc2547bis-03 (work in progress), October 2004.

[11] Marques, P., "Constrained VPN Route Distribution", draft-ietf-l3vpn-rt-constrain-02 (work in progress), June 2005.

[12] Martini, L., "Pseudowire Setup and Maintenance using the Label Distribution Protocol", draft-ietf-pwe3-control-protocol-17 (work in progress), June 2005.

[13] Kompella, K., "Layer 2 VPNs Over Tunnels", draft-kompella-l2vpn-l2vpn-00 (work in progress), January 2004.

[14] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) Bridges: Revision. This is a revision of ISO/IEC 10038: 1993, 802.1j-1992 and 802.6k-1992. It incorporates P802.11c, P802.1p and P802.12e. ISO/IEC 15802-3: 1998.", IEEE Standard 802.1D, July 1998.

Appendix A. Contributors

The following contributed to this document:

Javier Achirica, Telefonica
Loa Andersson, Acreo
Chaitanya Kodeboyina, Juniper
Giles Heron, Tellabs
Sunil Khandekar, Alcatel
Vach Kompella, Alcatel
Marc Lasserre, Riverstone
Pierre Lin
Pascal Menezes
Ashwin Moranganti, Appian
Hamid Ould-Brahim, Nortel
Seo Yeong-il, Korea Tel

Appendix B. Acknowledgements

Thanks to Joe Regan and Alfred Nothaft for their contributions. Many thanks too to Eric Ji, Chaitanya Kodeboyina, Mike Loomis and Elwyn Davies for their detailed reviews.

Authors' Addresses

Kireeti Kompella (editor)
Juniper Networks

1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kireeti@juniper.net

Yakov Rekhter (editor)
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: yakov@juniper.net

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

Dell et al., Ex. 1018
Page 39 of 39