

L2VPN Working Group  
Internet-Draft  
Expires: January 11, 2006

Y(J). Stein  
RAD Data Communications  
S. Delord  
France Telecom  
July 10, 2005

LDP-based Autodiscovery for L2 Services  
draft-stein-ldp-auto-00.txt

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/lid-abstracts.txt>.

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

This Internet-Draft will expire on January 11, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Current L2VPN implementations that use LDP for label binding require another protocol to dynamically detect which PEs belong to a given VPN. We herein propose a simple extension to LDP consisting of a message with which a PE announces its desire to join an existing VPN. The technique is equally applicable to HVPLS, multisegment VPLS and VPWS, and may be especially attractive for access networks employing MPLS.

Internet-Draft

vpls-ldp-auto

July 2005

## 1. Introduction

VPLS networks based on LDP signaling, as described in [VPLS-LDP], transparently interconnect N physically separated LANs belonging to a single customer. VPLS provides this multipoint to multipoint service based on a full mesh of Ethernet PWs [ETHERNET-PW]. We shall call each multipoint to multipoint network a VPLS instance or simply a VPN.

In the following we assume a service provider network, consisting of PE (provider edge) and P (provider) LSRs. The PEs must be VPLS-capable, i.e. they must be able to perform all the functionality described in [VPLS-LDP], including setting up of Ethernet PWs, encapsulating and properly forwarding incoming Ethernet frames, and 802.1D learning. The PE LSRs are connected to CE (customer edge) devices in the customer network, which may be Ethernet switches or IP routers. The connection between PE and CE is called the attachment circuit (AC).

Before offering VPLS services the service provider must provision a full mesh MPLS tunnels connecting all PEs; this being accomplished by manual configuration, BGP, RSVP-TE or LDP. Each VPLS instance defines a subset of the PEs, namely those connected to CEs that require the VPLS service. We shall call this subset of PEs the Vset. The VPLS service is implemented by means of Ethernet PWs inside the MPLS tunnels that we have assumed to connect every pair of PEs in the Vset. These PWs may be set up manually, or by using the LDP extensions defined in the PWE control protocol [PWE-CONTROL].

When a PE, not originally in the Vset wants to join the VPN, new Ethernet PWs must be set up between it and all the PEs already in the Vset. The act of joining a VPLS instance conceptually consists of two parts, namely discovering the Vset, and thereafter setting up (e.g. by using the PWE control protocol) PWs to all PEs in the Vset. Assuming an efficient mechanism for Vset discovery, the entire VPLS instance may be built iteratively by growing the Vset, starting with a single PE being told that it is in the Vset for the new VPLS instance.

Vset discovery may be explicit or implicit. With explicit Vset discovery, PEn determines that PE1 through PE(n-1) are in the Vset, and joins by setting up n-1 Ethernet PWs. With implicit Vset discovery PEn announces to all VPLS-capable PEs that it wishes to

join the particular VPLS instance, and PEs already in the Vset initiate the PW setup. We adopt here the implicit method as it minimizes the amount of messaging traffic, and is conceptually more compatible with MPLS downstream label distribution procedures. Rather than trying to directly locate participating PEs (e.g. by

requiring all PEs to periodically advertise the list of VPNs they handle), it makes sense for the PE desiring to join the VPN to advertise this fact to all the other PEs, permitting them to respond when they belong to the VPN. This method minimizes overhead and shortens the time it takes until all PEs know of the new VPN member.

The discovery of the Vset of an extant VPLS instance is not the only autodiscovery problem involved in provisioning networks capable of providing VPLS services. The VPLS-capable PEs must a priori know each other in order to set up the initial set of MPLS tunnels, and the PE must recognize CEs in order to set up the attachment circuits. However, we contend that the Vset discovery problem is of a different nature than the others. The number of PEs in a network is relatively limited and static, and in most cases autodiscovery protocols already exist for this problem. Provisioning of new attachment circuits is also relatively rare, and will often require management intervention. On the other hand the adding of a LAN to an existing VPN is expected to be much more prevalent. Since a large number of PEs may be involved in any particular VPN, manual configuration of the VPLS is logistically difficult and error-prone. Scalability requires an autodiscovery protocol for this task, and several such mechanisms exist, for example in BGP [BGP-AUTO] and extensions to RADIUS [RADIUS-AUTO].

The autodiscovery protocol described here is limited to the problem of discovering the Vset. We always assume that attachment circuits are in place, and that there is some method to inform the PE that a given CE needs to join a given VPLS, and for the PE to authenticate this request. For the simplest case we further assume that MPLS tunnels capable of supporting Ethernet PWs have been preprovisioned between every two PEs in the provider network, and that LDP sessions are already running between every pair of PEs. For more complex cases, e.g. HVPLS and multisegment VPLS, we assume that the existence of MPLS tunnels and LDP sessions between the end PEs and the stitching nodes.

As we deal solely with Vset discovery, the actual setting up of PWs is beyond our scope. This setting up of Ethernet PWs trivially follows for the simple case, but the PW placement problem may be complex for other cases. For example, for HVPLS with MPLS-based spoke PWs, the MTUs that aggregate Ethernet traffic from several customers originate the request for Vset discovery, but as they are directly aware of only one PE, the discovery messaging must be forwarded by that PE to the others with which it is fully interconnected.

Another case of interest is multisegment VPLS, which we define as a VPN whose Vset consists of PEs in multiple independently managed

domains, and thus requires multisegment PWs [MSPWs]. Here there will be S-PEs that straddle the various domains, and our autodiscovery mechanism will need to traverse these S-PEs as the PEs in distinct domains are not linked by LDP sessions. Note that since our mechanism is limited to Vset discovery, it is sufficient for our purposes for a PE to know of a single S-PE enabling access to each foreign domain, and this S-PE will not necessarily be that which is eventually traversed by the PW.

The present document proposes a simple Vset autodiscovery mechanism that requires only the extension of the LDP protocol that has been assumed to already be running between the PEs. The method proposed is distributed and does not require a central server holding VPLS member information. Rather than querying all VPLS-capable PEs to determine whether they belong to the Vset, each PE desiring to join a VPLS instance announces this by a new JOIN TLV, and receiving PEs that belong to the Vset respond by setting up the required PWs. In [Section 2](#) we discuss why this method is most suitable for access networks; in [Section 3](#) we motivate the use of human readable VPN identifiers; in [Section 4](#) we detail the required extensions to LDP; [Section 5](#) extends the discussion to more complex cases; and in [Section 6](#) we briefly cover the security considerations.

## [2.](#) Access Networks

The autodiscovery method described herein may be particularly suitable for access networks. Core networks will typically be running BGP-4, and thus have at their disposal the autodiscovery mechanisms described in [VPLS-BGP]. In addition the core network may

already be providing L3VPN services, and thus already utilizing BGP-based autodiscovery mechanisms.

However, in the case of access networks, such a protocol will frequently not be available to the service provider or not be completely suitable, and this for at least two reasons.

First, the processing power of access network forwarding devices may be quite limited as compared to backbone routers. This is due to the fact that the total number of devices is significantly greater in the access/metro segment than in the core network. Typically DSLAMs represent several thousand devices, and so need to be as simple as possible to keep the global network complexity (and cost) low.

Second, the topology of an access network may also be much simpler than the topology of a backbone network. For example, it is quite common to have a DSLAM with only a single physical attachment, (or perhaps a dual physical attachment for redundancy) to the metro/backbone area. Such DSLAMs may not even run an IGP and employ only

static routes in order to avoid CPU overload.

Despite these constraints, advanced services, such as interconnecting two endpoints located on the same or different access networks (e.g. two DSLAMs that do not belong to the same geographical area) are still required by the service provider. Solutions such as [MSPWs] allow the total number of PWs to grow without impacting the control plane, by limiting the number of targeted LDP sessions between the U-PEs to only a few S-PEs. However such solutions at present still require BGP or RADIUS for the auto-discovery process, in addition to LDP for setting up of the PWs.

Our proposal extends LDP with new TLVs thus enabling auto-discovery in the access network without requiring burdening down the access network forwarding devices with additional protocols. The TLVs reuse the endpoint identifiers defined in [PWE-CONTROL]. This same LDP-based autodiscovery method may be used in the core network, but alternatively the core network may employ [BGP-AUTO].

### 3. Importance of unique and meaningful VPN identifiers

According to [VPLS-LDP] each VPLS instance must be assigned a

globally unique identifier, a VPNID. This identifier is often numeric, for example, a 7-byte VPN Identifier per [RFC 2685](#) [VPN-ID].

A useful format for the VPN identifier used for the purpose of autodiscovery is a human readable name in URL-like format. This suggestion has been made in this context before, and is realized in the VPN identifier record used in RADIUS discovery [RADIUS]. Such an identifier minimizes the chance of accidental overlap with other VPNIDs, and eliminates the need for maintaining a network-wide directory of VPNIDs.

On the other hand, for the purpose of setting up the Ethernet PWs comprising the VPLS, an AGI is required. According to [PWE-CONTROL], the AGI may be of arbitrary format, and

"The details of how to construct the AGI and AII fields identifying the pseudowire endpoints are outside the scope of this specification. Different pseudowire application, and different provisioning models, will require different sorts of AGI and AII fields. The specification of each such application and/or model must include the rules for constructing the AGI and AII fields."

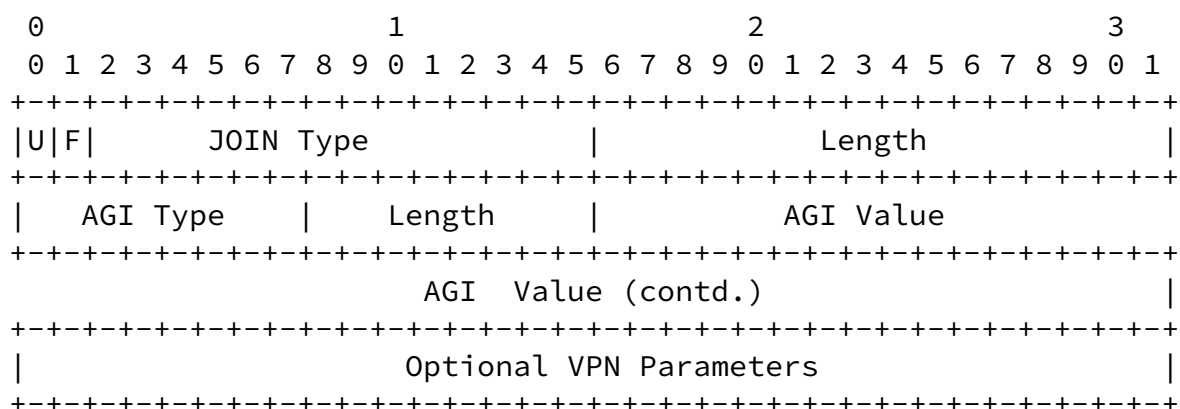
When possible the AGI should be simply be the URL-like VPNID herein described. When this is not possible, for example, when a [RFC 2685](#) VPN-ID is needed, a mechanism must be provided to translate the URL to the requisite format.

#### [4.](#) LDP extensions

In the standard model, a full mesh of LDP sessions is already running between all PEs that may wish to join the VPN. In the multisegment case, we assume LDP sessions between the U-PE and related S-PEs. It is over these LDP sessions regulating the MPLS transport tunnels that the new TLVs are advertised.

When a PE needs to add a CE to a VPN, it first consults a table to determine whether it is already participating in this VPLS instance for some other CE. If it does, then all that needs to be done is to connect the new CE to the existing bridge function and Ethernet PWs. If it does not, the joining PE must first allocate a bridge function to perform the 802.1 functionality and to add the VPNID to the table of VPNs in which it participates. Next, it sends an LDP message

containing a VPN JOIN TLV to all the PEs with which it maintains LDP sessions. This TLV has the following format.



U (1 bit) The unknown bit MUST be cleared. If the VPN join format is not understood, the sending PE must be informed that its attempt to join has been ignored. In this case an alternative mechanism must be employed to determine whether the remote PE participates in the VPN.

F (1 bit) The forwarding bit is not used, and MUST be cleared.

Type (14 bits) This field MUST be set to whatever value is assigned by IANA for this purpose.

Length (16 bits) This field specifies the total length of the VPLS identifier in bytes.

AGI TLV Attachment Group Identifier TLV per [PWE-CONTROL].

Optional VPN Parameters These optional TLVs may have to be specified for proper or optimal operation of the service.

When a PE needs to join a VPLS instance, it sends the JOIN TLV to all VPLS-capable PEs to which it is connected. The JOIN TLV is similar

to a label REQUEST, but the corresponding label mapping is contingent on the receiving PE belonging to the VPLS. However, the JOIN TLV does not specify a FEC, rather it specifies a VPNID.

When a PE receives a JOIN message it checks the VPNIDs of all the VPLS instances to which it belongs. If it finds a match it adds the joining PE to the VPN-PE table, allocates a PW label, and distributes this label to the sending PE via the LDP-based PW signaling protocol.

The label distribution message used MUST be of the Generalized PW ID FEC Element (FEC type 129), with AGI set equal to the VPNID specified in the JOIN message, and SAI and TAI set to zero. AIs are not needed for VPLS since packets received at the PE with the label mapped to the VPLS instance, since the Ethernet frame is sent to a bridging function that decides to which CE the frame needs to be forwarded.

Once the PE that had sent the JOIN message receives the mapping message with an AGI matching the VPNID, it adds the remote PE to its VPN-PE table, allocates and sends back a PW label using the Generalized PW ID FEC Element. This process is repeated for all PEs in the VPN.

When a PE needs to leave a VPLS instance, it sends the LEAVE TLV to all PEs to which it is connected that participate in the given VPN. The format of the TLV is identical to the JOIN format, except for the TYPE value, and not having any optional parameters. LEAVE messages are similar to label RELEASE, but specifies a VPNID instead of a FEC.

When a PE receives a LEAVE message it checks the VPNIDs of all the VPLS instances to which it belongs. If it finds a match, it frees the corresponding PW label and sends a label WITHDRAWAL message.

## 5. Other Cases

The previous section dealt with signaling required for autodiscovery and setting up of PWs for the simple VPLS case. In this section we will extend this treatment to other cases, namely HVPLS [VPLS-LDP], multisegment VPLS based on multisegment PWs [MSPWs], and VPWS.

In the HVPLS case the MTU originates the JOIN and LEAVE messages, but it can send these only to the PE to which it is connected. The PE, upon receiving a JOIN request, performs any authentication that is required, and then notes the MTU's VPN participation in a VPN-MTU table. It then forwards the JOIN request to all the PEs with which it maintains LDP sessions. A PE receiving such a request checks its VPN-MTU tables for membership, and if finding that its participation is required adds the originating PE to its VPN-PE table, and send a label mapping message back. Note that it does not need to inform

attached MTUs of the new association, as this will be automatically



learned by the bridging function. If BGP is running between the PEs, then [BGP-AUTO] could also be used.

When the originating PE receives a label mapping with AGI indicating that it is in response to the previously sent JOIN, it adds the remote PE to its VPN-PE table, and returns a label mapping message with the same AGI to set up the other direction.

For the multisegment case the S-PE is required to forward JOIN messages between the two segments (in both directions). The assumption here is that LDP sessions are already running between the U-PEs and their related S-PEs. Although there may be many S-PEs between a pair of domains, it is sufficient for a single S-PE to be designated as the point of contact between each pair of domains for the purpose of autodiscovery. When a PE in one domain wants to join a VPLS that extends over multiple domains, it sends JOIN messages with a globally unique VPNID to all VPLS-capable PEs in its domain, and to all S-PEs. The S-PEs forward the JOIN messages in their respective domains. A receiving PE in the same domain as the originating PE sends its label mapping back to the originating PE. A PE in a different domain sends its label mapping to an S-PE, using whatever mechanism is selected by the PWE3 WG for multisegment PW setup. If BGP is running between the PEs, then [BGP-AUTO] could also be used.

Although the scalability problem is less for VPWS, it may be necessary to use the mechanism described herein in order to set up a large number of point to point PWs. We shall deal solely with the single segment case, although a similar method is applicable VPWS based on MS-PWs.

When the PW can be uniquely identified by a 32-bit identifier, it is sufficient to use the PWid FEC Element; the problem arises when we wish to use a meaningful VPNID. In this case either side can send a JOIN message containing the VPNID as AGI and an AII indication as an optional parameter to all relevant PEs. The PE receiving the JOIN which itself wishes to be part of that PW returns a generalized PW ID label mapping with the VPNID as AGI, SAII as locally identified, and TAII as received in the JOIN message.

## 6. Security Considerations

Security mechanisms are important for all automatic admission mechanisms, and for VPNs the issues of security are paramount. The responsibility for admission into a VPN rests with the service provider, as this security is an integral part of the "private service" being offered.

In order to provide a pseudo-private service, the provider MUST check the authorization of any request to join a VPN, and MUST ensure that the packets are only delivered to the proper remote CE.

By using manual provisioning of the CE-PE portion of the VPN the first component of the joining can be made relatively safe. Mechanization of the PE to PE connection component eliminates errors, and thus mitigates security problems of the second type.

It is recommended that LDP authentication methods be utilized to deter unauthorized parties joining a VPLS instance.

## 7. IANA Considerations

In order to implement the LDP extensions defined here, we will need two new LDP types, one for the VPN JOIN and one for the VPN LEAVE TLV.

## 8. References

### 8.1 Normative References

- [ETHERNET-PW] "Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks", September 2004, [draft-ietf-pwe3-ethernet-encap-08.txt](#) (Work In Progress)
- [LDP] Andersson, et al., "LDP Specification", [RFC 3036](#)
- [MPLS] [RFC 3032](#) "MPLS Label Stack Encoding"
- [PWE-CONTROL] "Pseudowire Setup and Maintenance using LDP", March 2005, [draft-ietf-pwe3-control-protocol-16.txt](#) (Work In Progress)
- [VPLS-LDP] "Virtual Private LAN Services over MPLS", Month 200x, [draft-ietf-l2vpn-vpls-ldp-0n.txt](#) (Work In Progress)
- [VPNID] [RFC 2685](#) "Virtual Private Networks Identifier", September 1999

### 8.2 Informative References

- [BGP-AUTO] "Using BGP as an Auto-Discovery Mechanism for Network-based VPNs", [draft-ietf-l3vpn-bgpvpn-auto-05.txt](#) (Work In Progress)
- [MSPWs] "Pseudo Wire Switching", [draft-martini-pwe3-pw-switching-03.txt](#) (Work In Progress)
- [RADIUS-AUTO] "Radius/L2TP Based VPLS", [draft-ietf-l2vpn-l2tp-radius-00.txt](#) (Work In Progress)
- [VPLS-BGP] "Virtual Private LAN Service", Month 200x, [draft-ietf-l2vpn-vpls-bgp-0n.txt](#) (Work In Progress)

## 9. Acknowledgments

The authors would like to thank Yuri Gittik and Philippe Nizer for interesting discussions and valuable contributions to the work herein

Stein & Delord

Expires January 11, 2006

[Page 9]

---

Internet-Draft

vpls-ldp-auto

July 2005

presented.

#### Authors' Addresses

Yaakov (J) Stein  
RAD Data Communications  
24 Raoul Wallenberg St., Bldg C  
Tel Aviv 69719  
ISRAEL

Phone: +972 3 645-5389  
Email: yaakov\_s@rad.com

Simon Delord  
France Telecom  
2 av. Pierre Marzin  
Lannion 22300  
FRANCE

Email: simon.delord@francetelecom.com

Internet-Draft

vpls-ldp-auto

July 2005

### 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](mailto: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.