                       Virtual Private LAN Service

Status of this Memo

Copyright Notice

Abstract

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

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

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. 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 a 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 several individual LANs across a packet-switched network
to appear and function as a single LAN [2].

This document describes the functions needed to offer VPLS, and goes
on to describe a mechanism for signaling a VPLS, as well as a
mechanism for transport of VPLS frames over tunnels across a packet
switched network. The 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: [3], which allows one to build a
Layer 2 VPN with Ethernet as the interconnect; and [4], 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
[5].

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

[6] 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. Section 3 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 [3], [7] and [8]): what is described here is based on
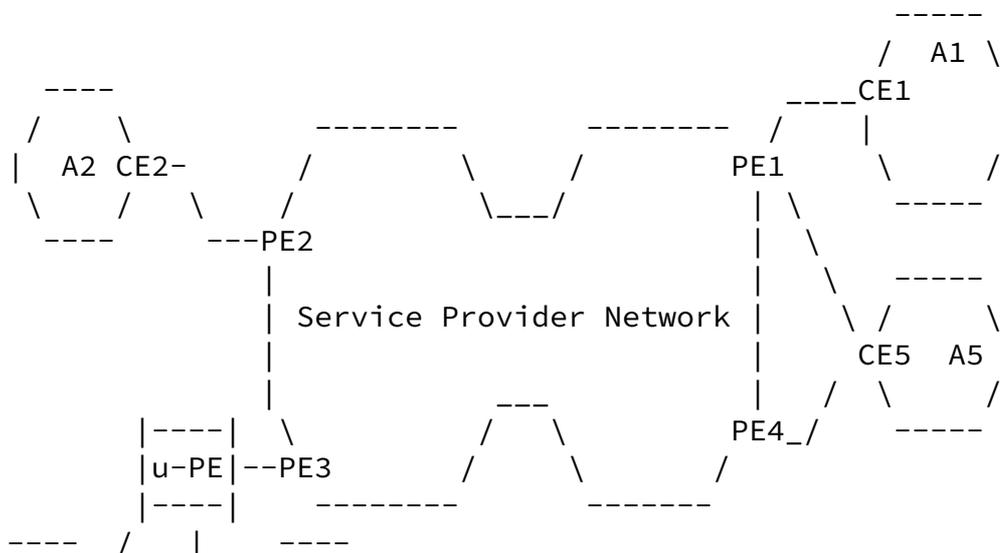the mechanisms proposed in [7].

The forwarding plane and the actions that a participating PE 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.

2. Functional Model

This will be described with reference to Figure 1.

Figure 1: Example of a VPLS

```
                                                              _____
                                                             /  A1 \
        ____                                    ____CE1        |
       /    \         --------      --------   /    |          |
      |  A2 CE2-      /        \    /       PE1    \      /
       \    /  \    /          \___/        |  \    -----
        ----      ---PE2                    |   \
                   |                        |    \  _____
                   | Service Provider Network |    \ /     \
                   |                        |       CE5  A5 |
                   |              ___       |    / \     /
        |----|  \         /   \         PE4_/     -----
        |u-PE|--PE3       /     \        /
        |----|    -------        -------
   ----  /   |    ----
```
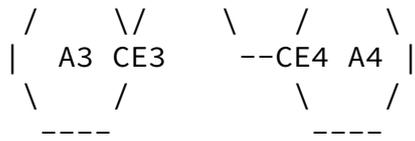
```
   /    \/    \   /    \              CE = Customer Edge Device
  |  A3 CE3    --CE4 A4 |             PE = Provider Edge Router
   \   /         \    /              u-PE = Layer 2 Aggregation
    ----          ----              A<n> = Customer site n
```

2.1. Terminology

Terminology similar to that in [7] is used, with the addition of "u-PE", a Layer 2 PE device used for Layer 2 aggregation. A u-PE is owned and operated by the Service Provider (as is the PE). 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) full-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,
i.e., every (ingress) PE can send a VPLS packet to the egress PE(s)
directly, without the need for an intermediate PE (see the section
below on "Split Horizon" Flooding). This assumption reduces (but
does not eliminate) the need to run Spanning Tree Protocol among the
PEs.

[2.3](#). Interactions

VPLS is a successful "LAN Service" if CE devices that belong to VPLS
V can interact through the SP network as if they were connected by a
LAN. VPLS is "private" if CE devices that belong to different VPLSs
cannot interact. VPLS is "virtual" if 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 (i.e., that are attached to CE devices that belong to
the same VPLS), and to exchange demultiplexors. These interactions
are control-driven, not data-driven.

U-PEs interact with PEs to establish connections with remote PEs or
u-PEs in the same VPLS. Again, this interaction is control-driven.

[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. The first two

subsections describe these functions. The next subsection describes
the setting up of pseudowires that span Autonomous Systems. The last
subsection details how multi-homing is handled.

3.1. Autodiscovery

Discovery refers to the process of finding all the PEs that
participate in a given VPLS. 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 (in this and other VPLS approaches) that the PEs
participating in a given VPLS are fully meshed (i.e., every pair of
PEs in a given VPLS establish pseudowires to each other).
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 that each customer belongs to, not the identity of every
other PE in that VPLS. 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 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
[3] and [7]; it uses BGP extended communities [9] to identify members
of a VPLS. A more generic autodiscovery mechanism is described in
[8]. The specific extended community used is the Route Target, whose
format is described in [9]. The semantics of the use of Route
Targets is described in [7]; 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 initiate "relearning", and to transmit
certain characteristics of the PE regarding 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, even though the PE-to-PE tunnels may not be MPLS tunnels.

3.2.1. Setup and Teardown

The VPLS BGP NLRI described below, with a new AFI and SAFI (see [6])
is used to exchange demultiplexors.

A PE advertises a VPLS NLRI for each VPLS that it participates in.
If the PE is doing learning and flooding, i.e., it is the VE, it
announces a single set of VPLS NLRIs for each VPLS that it is in. If
the PE is connected to several u-PEs, it announces one set of VPLS
NLRIs for each u-PE. A hybrid scheme is also possible, where the PE
learns MAC addresses on some interfaces (over which it is directly
connected to CEs) and delegates learning on other interfaces (over
which it is connected to u-PEs). In this case, the PE would announce
one set of VPLS NLRIs for each u-PE that has customer ports in a
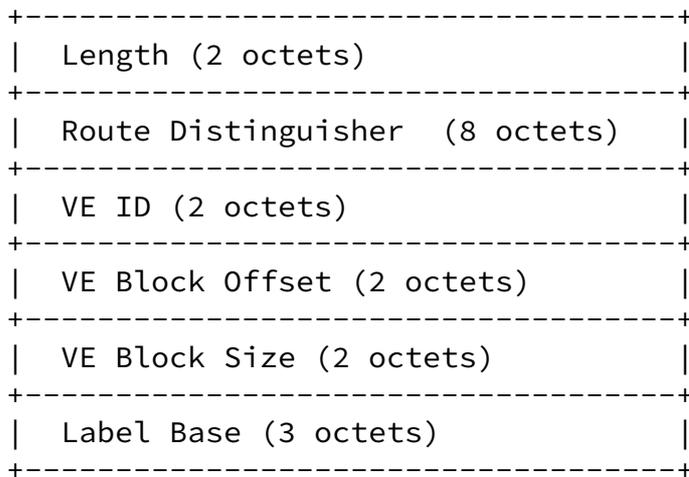given VPLS, and one set for itself, if it has customer ports in that
VPLS.

Each set of NLRIs defines the demultiplexors for a range of other PEs
in the VPLS. Ideally, a single NLRI suffices to cover all PEs in a
VPLS; however, there are cases (such as a newly added PE) where the
pre-existing NLRI does not have enough labels. In such cases,
advertising an additional NLRI for the same VPLS serves to add labels
for the new PEs without disrupting service to the pre-existing PEs.
If service disruption is acceptable (or when the PE restarts its BGP
process), a PE MAY consider coalescing all NLRIs for a VPLS into a
single NLRI.

If a PE X is part of VPLS V, and X receives a VPLS NLRI for V from PE
Y that includes a demultiplexor that X can use, X sets up its ends of
a pair of pseudowires between X and Y. X may also have to advertise
a new NLRI for V that includes a demultiplexor that Y can use, if its
pre-existing NLRI for V did not include a demultiplexor for Y.

If Y's configuration is changed to remove it from VPLS V, then Y MUST
withdraw all its NLRIs for V. If all Y's links to CEs in V go down,
then Y SHOULD either withdraw all its NLRIs for V, or let other PEs
in the VPLS V know in some way that Y is no longer connected to its
CEs.

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

The format of the VPLS NLRI is given below. The AFI and SAFI are the
same as for the L2 VPN NLRI [3].

Figure 2: BGP NLRI for VPLS Information

```
+----------------------------------+
|   Length (2 octets)              |
+----------------------------------+
|   Route Distinguisher  (8 octets)   |
+----------------------------------+
|   VE ID (2 octets)               |
+----------------------------------+
|   VE Block Offset (2 octets)     |
+----------------------------------+
|   VE Block Size (2 octets)       |
+----------------------------------+
|   Label Base (3 octets)          |
+----------------------------------+
```

3.2.2. Signaling PE Capabilities

   The Encaps Type and Control Flags are encoded in an extended
   attribute. The community type also is used in L2 VPNs [3].

   The Encaps Type for VPLS is 19.
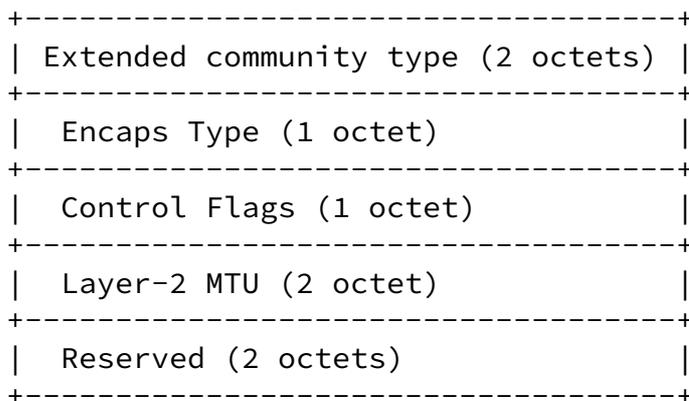

Figure 3: layer2-info extended community

```
+----------------------------------+
| Extended community type (2 octets) |
+----------------------------------+
|   Encaps Type (1 octet)          |
+----------------------------------+
|   Control Flags (1 octet)        |
+----------------------------------+
|   Layer-2 MTU (2 octet)          |
+----------------------------------+
|   Reserved (2 octets)            |
+----------------------------------+
```


Figure 4: Control Flags Bit Vector

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| MBZ |P|Q|F|C|S|      (MBZ = MUST Be Zero)
+-+-+-+-+-+-+-+-+
```

With reference to Figure 4, the following bits are defined; the MBZ
bits MUST be set to zero.

        Name   Meaning
           P   If set to 1, then the PE will strip the outermost VLAN
                tag from the customer frame on ingress, and push a
                VLAN tag on egress. If set to 0, the customer frame
                is left unchanged.
           Q   Reserved.
           F   If set to 1 (0), the PE is (not) capable of flooding.
           C   If set to 1 (0), Control word is (not) required when
                encapsulating Layer 2 frames [10].
           S   If set to 1 (0), Sequenced delivery of frames is (not)
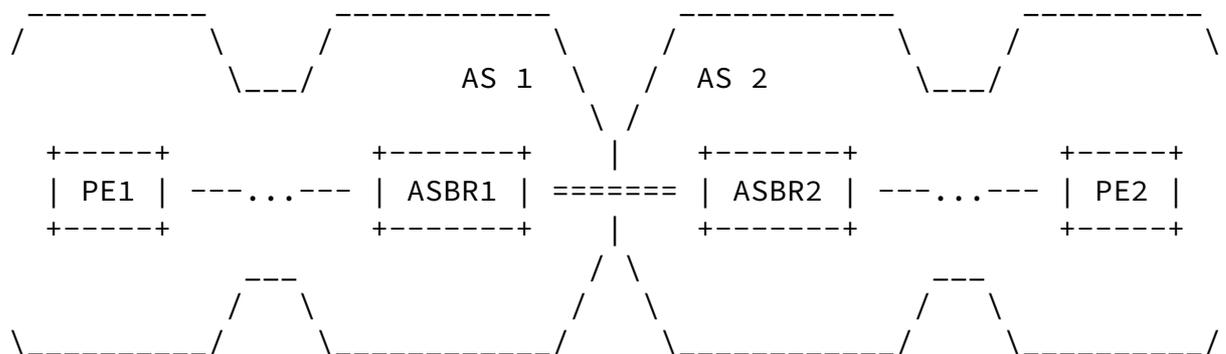   required.

3.3. Multi-AS VPLS

   As in [3] and [7], 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 may be
   needed; and 2) there may not be PE-to-PE tunnels between the ASes.

   A similar problem is solved in [7], 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:

      _____   _____   _____   _____
     /   \   /   \   /   \   /   \
               ___/   AS 1 \   /   AS 2   ___/
                                 \ /
    +-----+   +-------+ | +-------+   +-----+
    | PE1 | ---...--- | ASBR1 | ======= | ASBR2 | ---...--- | PE2 |
    +-----+   +-------+ | +-------+   +-----+
          ___   / \   ___
         /   \   /   \   /   \
     _____/   _____/   _____/   _____/

   a) VPLS-to-VPLS connections at the AS border routers.

      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 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 the Spanning Tree
Protocol for each such VPLS..

  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).

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 [7], 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.

    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 is no
    overlap between VE ID ranges among ASes. The exception to this rule
    is multi-homing, which is dealt with below.

3.4. 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.

    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 MUST
announce the same VE Block Size for a given VE Offset.

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 [10], 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

Forwarding of VPLS packets is based on the interface over which the
packet is received, which determines which VPLS the packet belongs
to, and 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 connections from the SP edge, be it a PE or a
u-PE, to the CE. 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.

There are two modes of learning: qualified and unqualified learning.

In qualified learning, the learning decisions at the VE are based on
the customer ethernet packet's MAC address and VLAN tag, if one
exists. This VLAN is often called the "service delimiting VLAN".
Each VLAN on a given port is mapped to a different service (VPLS, IP
VPN, point-to-point Layer 2 VPN, etc.); each VLAN that is mapped to a
VPLS service has its own VPLS FIB.

In unqualified learning, learning is based on a customer ethernet
packet's MAC address only. This is also called "port-mode VPLS".

4.2.2. 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.

4.2.3. "Split Horizon" Flooding

When a PE capable of flooding 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, the PE must send a copy of
the frame to every other attached CE, as well as to all PEs
participating in the VPLS. If the frame arrived from another PE,
however, the PE must only send a copy of the packet to attached CEs.
The PE MUST NOT send the frame to other PEs. This notion has been
termed "split horizon" flooding, and is a consequence of the PEs
being logically full-meshed -- if a broadcast frame is received from
PEx, then PEx would have sent a copy to all other PEs.

5. Deployment Options

In deploying a network that supports VPLS, the SP must decide whether
the VPLS-aware device closest to the customer (the VE) is a u-PE or a
PE. 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 u-
PE, i.e., a device that does layer 2 functions such as MAC address
learning and flooding, and some limited layer 3 functions such as
communicating to its PE, but doesn't do full-fledged discovery and
PE-to-PE signaling.

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. PE1 may be directly connected to CE devices; PE2
may be connected to u-PEs that are connected to CEs; and PE3 may be
connected directly to a customer over some interfaces and to u-PEs
over others. All these PEs do 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.

6. Normative References

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

    [6] Bates, T., Rekhter, Y., Chandra, R., and Katz, D.,
         "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000

    [9] Sangli, S., D. Tappan, and Y. Rekhter, "BGP Extended Communities
         Attribute", draft-ietf-idr-bgp-ext-communities-07.txt (work in
         progress)

    [10] Martini, L., et al, "Encapsulation Methods for Transport of
         Ethernet Frames Over IP/MPLS Networks", draft-ietf-
         pwe3-ethernet-encap-06.txt (work in progress)

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

7. Informative References

    [2] Andersson, L., and Rosen, E., "Framework for Layer 2 Virtual
         Private Networks (L2VPNs)", draft-ietf-l2vpn-l2-framework-04.txt
         (work in progress)

    [3] Kompella, K., (Editor), "Layer 2 VPNs Over Tunnels", draft-
         kompella-l2vpn-l2vpn-00.txt (work in progress)

    [4] Martini, L., et al, "Pseudowire Setup and Maintenance using LDP"
         draft-ietf-pwe3-control-protocol-06.txt (work in progress)

    [5] Kompella, V., et al, "Virtual Private LAN Services over MPLS",
         draft-ietf-ppvpn-vpls-ldp-03.txt (work in progress)

    [7] Rosen, E., and Rekhter, Y., Editors, "BGP/MPLS VPNs", draft-
         ietf-l3vpn-rfc2547bis-01.txt (work in progress)

    [8] Ould-Brahim, H., Rosen, E., and Rekhter, Y., "Using BGP as an
         Auto-Discovery Mechanism for Layer-3 and Layer-2 VPNs", draft-
         ietf-l3vpn-bgpvpn-auto-04.txt (work in progress)

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 [11] 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 [11] 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.

IANA Considerations

   IANA is asked to allocate an AFI for Layer 2 information (suggested
   value: 25).

Contributors

   The following contributed to this document:

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

Acknowledgments

   Thanks to Joe Regan and Alfred Nothaft for their contributions.

Authors' Addresses

   Kireeti Kompella
   Juniper Networks
   1194 N. Mathilda Ave
   Sunnyvale, CA 94089
   kireeti@juniper.net

   Yakov Rekhter
   Juniper Networks
   1194 N. Mathilda Ave

    Sunnyvale, CA 94089
    yakov@juniper.net

IPR Notice

    The IETF takes no position regarding the validity or scope of any
    intellectual property 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; neither does it represent that it
    has made any effort to identify any such rights. Information on the
    IETF's procedures with respect to rights in standards-track and
    standards-related documentation can be found in BCP-11. Copies of
    claims of rights made available for publication 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 implementors or users of this specification can
    be obtained from the IETF Secretariat.

    The IETF invites any interested party to bring to its attention any
    copyrights, patents or patent applications, or other proprietary
    rights which may cover technology that may be required to practice
    this standard. Please address the information to the IETF Executive
    Director.

document itself may not be modified in any way, such as by removing
the copyright notice or references to the Internet Society or other
Internet organizations, except as needed for the purpose of
developing Internet standards in which case the procedures for
copyrights defined in the Internet Standards process must be
followed, or as required to translate it into languages other than
English.

The limited permissions granted above are perpetual and will not be
revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an
"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
TASK FORCE DISCLAIMS 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."

Acknowledgement