                    Supporting IP Multicast over VPLS

Status of this memo

  By submitting this Internet-Draft, we represent that any applicable
  patent or other IPR claims of which we are aware have been disclosed,
  or will be disclosed, and any of which we become aware will be
  disclosed in accordance with RFC 3668.

  This document is an Internet-Draft and is in full conformance with
  Sections 5 and 6 of RFC 3667 and Section 5 of RFC 3668.

  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.

IPR Disclosure Acknowledgement

  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.

Abstract

  In Virtual Private LAN Service (VPLS), the PE devices provide a
  logical interconnect such that CE devices belonging to a specific
  VPLS instance appear to be connected by a single LAN. A VPLS
  solution performs replication for multicast traffic at the ingress PE
  devices. When replicated at the ingress PE, multicast traffic wastes
  bandwidth when 1. Multicast traffic is sent to sites with no members,
  and 2. Pseudo wires to different sites go through a shared path.

This document is addressing the former by IGMP and PIM snooping.

  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](RFC 2119).

Table of Contents

1. Contributing Authors

   This document was the combined effort of several individuals. The
   following are the authors, in alphabetical order, who contributed to
   this document:

          Suresh Boddapati
          Venu Hemige
          Sunil Khandekar
          Vach Kompella
          Marc Lasserre
          Rob Nath
          Ray Qiu
          Yetik Serbest
          Himanshu Shah

2. Introduction

   In Virtual Private LAN Service (VPLS), the Provider Edge (PE) devices
   provide a logical interconnect such that Customer Edge (CE) devices
   belonging to a specific VPLS instance appear to be connected by a
   single LAN. Forwarding information base for particular VPLS instance
   is populated dynamically by source MAC address learning. This is a
   straightforward solution to support unicast traffic, with reasonable

flooding for unicast unknown traffic. Since a VPLS provides LAN
emulation for IEEE bridges as wells as for routers, the unicast and
multicast traffic need to follow the same path for layer-2 protocols
to work properly. As such, multicast traffic is treated as broadcast
traffic and is flooded to every site in the VPLS instance.

VPLS solutions (i.e., [VPLS-LDP] and [VPLS-BGP]) perform replication
for multicast traffic at the ingress PE devices. When replicated at
the ingress PE, multicast traffic wastes bandwidth when: 1. Multicast
traffic is sent to sites with no members, 2. Pseudo wires to
different sites go through a shared path, and 3. Multicast traffic is
forwarded along a shortest path tree as opposed to the minimum cost
spanning tree. This document is addressing the first problem by IGMP
and PIM snooping. Using VPLS in conjunction with IGMP and/or PIM
snooping has the following advantages:
    -    It improves VPLS to support IP multicast efficiently (not
    necessarily optimum, as there can still be bandwidth waste),
    -    It prevents sending multicast traffic to sites with no
    members,
    -    It keeps P routers in the core stateless,
    -    The Service Provider (SP) does not need to perform the tasks
    to provide multicast service (e.g., running PIM, managing P-group
    addresses, managing multicast tunnels)
    -    The SP does not need to maintain PIM adjacencies with the
    customers.

In this document, we describe the procedures for Internet Group
Management Protocol (IGMP) and Protocol Independent Multicast (PIM)
snooping over VPLS for efficient distribution of IP multicast
traffic.

3. Overview of VPLS

In case of VPLS, the PE devices provide a logical interconnect such
that CE devices belonging to a specific VPLS appear to be connected
by a single LAN. End-to-end VPLS consists of a bridge module and a
LAN emulation module ([L2VPN-FR]).

In a VPLS, a customer site receives Layer-2 service from the SP. The
PE is attached via an access connection to one or more CEs. The PE
performs forwarding of user data packets based on information in the
Layer-2 header, that is, MAC destination address. The CE sees a
bridge.

The details of VPLS reference model, which we summarize here, can be
found in [L2VPN_FR]. In VPLS, the PE can be viewed as containing a
Virtual Switching Instance (VSI) for each L2VPN that it serves. A CE
device attaches, possibly through an access network, to a bridge

module of a PE. Within the PE, the bridge module attaches, through
an Emulated LAN Interface to an Emulated LAN. For each VPLS, there

is an Emulated LAN instance. The Emulated LAN consists of VPLS
Forwarder module (one per PE per VPLS service instance) connected by
pseudo wires (PW), where the PWs may be traveling through Packet
Switched Network (PSN) tunnels over a routed backbone. VSI is a
logical entity that contains a VPLS forwarder module and part of the
bridge module relevant to the VPLS service instance [L2VPN-FR].
Hence, the VSI terminates PWs for interconnection with other VSIs and
also terminates attachment circuits (ACs) for accommodating CEs. A
VSI includes the forwarding information base for a L2VPN [L2VPN-FR]
which is the set of information regarding how to forward Layer-2
frames received over the AC from the CE to VSIs in other PEs
supporting the same L2VPN service (and/or to other ACs), and contains
information regarding how to forward Layer-2 frames received from PWs
to ACs. Forwarding information bases can be populated dynamically
(such as by source MAC address learning) or statically (e.g., by
configuration). Each PE device is responsible for proper forwarding
of the customer traffic to the appropriate destination(s) based on
the forwarding information base of the corresponding VSI.

[4](4). Multicast Traffic over VPLS

In VPLS, if a PE receives a frame from an Attachment Circuit (AC)
with no matching entry in the forwarding information base for that
particular VPLS instance, it floods the frame to all other PEs (which
are part of this VPLS instance) and to directly connected ACs (other
than the one that the frame is received from). The flooding of a
frame occurs when:
-    The destination MAC address has not been learned,
-    The destination MAC address is a broadcast address,
-    The destination MAC address is a multicast address.

Malicious attacks (e.g., receiving unknown frames constantly) aside,
the first situation is handled by VPLS solutions as long as
destination MAC address can be learned. After that point on, the
frames will not be flooded. A PE is REQUIRED to have safeguards,
such as unknown unicast limiting and MAC table limiting, against
malicious unknown unicast attacks.

There is no way around flooding broadcast frames. To prevent runaway
broadcast traffic from adversely affecting the VPLS service and the
SP network, a PE is REQUIRED to have tools to rate limit the
broadcast traffic as well.

Similar to broadcast frames, multicast frames are flooded as well, as
a PE can not know where multicast members reside. Rate limiting
multicast traffic, while possible, should be should be done carefully
since several network control protocols relies on multicast. For one
thing, layer-2 and layer-3 protocols utilize multicast for their

operation. For instance, Bridge Protocol Data Units (BPDUs) use an
IEEE assigned all bridges multicast MAC address, and OSPF is
multicast to all OSPF routers multicast MAC address. If the rate-

limiting of multicast traffic is not done properly, the customer
network will experience instability and poor performance. For the
other, it is not straightforward to determine the right rate limiting
parameters for multicast.

A VPLS solution MUST NOT affect the operation of customer layer-2
protocols (e.g., BPDUs). Additionally, a VPLS solution MUST NOT
affect the operation of layer-3 protocols.

In the following section, we describe procedures to constrain the
flooding of IP multicast traffic in a VPLS.

5. Constraining of IP Multicast in a VPLS

The objective of improving the efficiency of VPLS for multicast
traffic that we are trying to optimize here has the following
constraints:
- The service is VPLS, i.e., a layer-2 VPN,
- In VPLS, ingress replication is required,
- There is no layer-3 adjacency (e.g., PIM) between a CE and a
  PE.

Under these circumstances, the most obvious approach is
implementation of IGMP and PIM snooping in VPLS. Other multicast
routing protocols such as DVMRP and MOSPF are outside the scope of
this document.

Another approach to constrain multicast traffic in a VPLS is to
utilize point-multipoint LSPs (e.g., [PMP-RSVP-TE]). In such case,
one has to establish a point-multipoint LSP from a source PE (i.e.,
the PE to which the source router is connected to) to all other PEs
participating in the VPLS instance. In this case, if nothing is
done, all PEs will receive multicast traffic even if they do not have
any members hanging off of them. One can apply IGMP/PIM snooping,
but this time IGMP/PIM snooping should be done in P routers as well.
One can propose a dynamic way of establishing point-multipoint LSPs,
for instance by mapping IGMP/PIM messages to RSVP-TE signaling. One
should consider the effect of such approach on the signaling load and
on the delay between the time the join request received and the
traffic is received (this is important for IPTV application for
instance). This approach is outside the scope of this document.

Although, in some extremely controlled cases, such as a ring topology
of PE routers with no P routers or a tree topology, the efficiency of
the replication of IP multicast can be improved. For instance, spoke
PWs of a hierarchical VPLS can be daisy-chained together and some
replication rules can be devised. These cases are not expected to be
common and will not be considered in this document.

In the following sub-sections, we provide some guidelines for the
implementation of IGMP and PIM snooping in VPLS. Snooping techniques

need to be employed on ACs at the downstream PEs. Snooping techniques
can also be employed on PWs at the upstream PEs. This may work well
for small to medium scale deployments. However, if there are a large
number of VPLS instances with a large number of PEs per instances,
then the amount of snooping required at the upstream PEs can
overwhelm the upstream PEs. In section 5.5. , we provide an
alternative approach using LDP to build multicast replication states
on the upstream PEs. Using a reliable mechanism like LDP allows the
upstream PEs to eliminate the requirement to snoop on PWs. It also
eliminates the need to refresh multicast states on the upstream PEs.

5.1. IPv6 Considerations

In VPLS, PEs forward Ethernet frames received from CEs and as such
are agnostic of the layer-3 protocol used by the CEs. However, as an
IGMP and PIM snooping switch, the PE would have to look deeper into
the IP and IGMP/PIM packets and build snooping state based on that.
As already stated, the scope of this document is limited to snooping
IGMP/PIM packets. So, we are concerned with snooping specific IP
payloads. Nonetheless, there are two IP versions a PE would have to
be able to interpret. IGMP is the Group Management Protocol which
applies only to IPv4. MLD [MLD] is the equivalent of IGMPv2 defined
for IPv6. MLDv2 [MLDv2] is the equivalent of IGMPv3 defined for
IPv6. PIM runs on top of both IPv4 and IPv6.

This document mandates that a PE MUST be able to snoop IGMP and PIM
encapsulated as IPv4 payloads. The PE SHOULD also be capable of
snooping MLD/MLDv2 packets and PIM packets encapsulated as IPv6
payloads. If the PE cannot snoop IPv6 payloads, then it MUST NOT
build any snooping state for such multicast groups and MUST simply
flood any data traffic sent to such groups. This allows an IPv6-
unaware PE to perform the snooping function only on IPv4 multicast
groups. This is possible because an IPv4 multicast address and an
IPv6 multicast address never share the same MAC address.

To avoid confusion, this document describes the procedures for
IGMP/PIM snooping for IPv4. The procedures described for IGMP can
also be applied to MLD and MLDv2. Please refer to Section 3 of
[MAGMA-SNOOP] for a list of IPv4/IPv6 differences an IGMP/MLD
snooping switch has to be aware of. In addition to those
differences, some of the other differences of interest are:

    -    IPv4 multicast addresses map to multicast MAC address
    starting with 01:00:5E and IPv6 multicast addresses map to
    multicast MAC addresses starting with 33:33. So the MAC ddresses
    used for IPv4 and IPv6 never overlap.

5.2. General Rules for IGMP/PIM Snooping in VPLS

The following rules for the correct operation of IGMP/PIM snooping
MUST be followed.

Rule 1: IGMP and PIM messages forwarded by PEs MUST follow the split-horizon rule for mesh PWs as defined in [VPLS-LDP].

Rule 2: IGMP/PIM snooping states in a PE MUST be per VPLS instance.

Rule 3: If a PE does not have any entry in a IGMP/PIM snooping state for multicast group (*,G) or (S,G), the multicast traffic to that group in the VPLS instance MUST be flooded.

Rule 4: A PE MUST support PIM mode selection per VPLS instance via CLI and/or EMS. Another option could be to deduce the PIM mode from RP address for a specific multicast group. For instance, a RP address can be learned during the Designated Forwarder (DF) election stage for Bidirectional-PIM.

[5.3](5.3). IGMP Snooping for VPLS

IGMP is a mechanism to inform the routers on a subnet of a hosts request to become a member of a particular multicast group. IGMP is a stateful protocol. The router (i.e., the querier) regularly verifies that the hosts want to continue to participate in the multicast groups by sending periodic queries, transmitted to all hosts multicast group (IP:224.0.0.1, MAC:01-00-5E-00-00-01) on the subnet. If the hosts are still interested in that particular multicast group, they respond with membership report message, transmitted to the multicast group of which they are members. In IGMPv1 [[RFC1112](RFC1112)], the hosts simply stop responding to IGMP queries with membership reports, when they want to leave a multicast group. IGMPv2 [[RFC2236](RFC2236)] adds a leave message that a host will use when it needs to leave a particular multicast group. IGMPv3 [[RFC3376](RFC3376)] extends the report/leave mechanism beyond multicast group to permit joins and leaves to be issued for specific source/group (S,G) pairs.

In IGMP snooping, a PE snoops on the IGMP protocol exchange between hosts and routers, and based on that restricts the flooding of IP multicast traffic. In the following, we explore the mechanisms involved in implementing IGMP snooping for VPLS. Please refer to Figure 1 as an example of VPLS with IGMP snooping. In the figure, Router 1 is the Querier. If multiple routers exist on a single subnet (basically that is what a VPLS instance is), they can mutually elect a designated router (DR) that will manage all of the IGMP messages for that subnet.

```
                         VPLS Instance
     +------+ AC1 +------+            +------+ AC4 +------+
     | Host |-----|  PE  |------------|  PE  |-----|Router|
     |  1   |     |  1   |\   PW1to3  /|  3   |     |  1   |
     +------+     +------+ \          / +------+     +------+
                   |     \       /      |
                   |      \     /       |
                   |       \   /PW2to3  |
                   |        \ /         |
           PW1to2| |         \          |PW3to4
                   |        / \         |
                   |       /   \PW1to4  |
                   |      /     \       |
                   |     /       \      |
     +------+     +------+ /       \ +------+     +------+
     | Host |     |  PE  |/  PW2to4 \|  PE  |     |Router|
     |  2   |-----|  2   |-----------|  4   |-----|  2   |
     +------+ AC2 +------+            +------+ AC5 +------+
                   |
                   |AC3
                +------+
                | Host |
                |  3   |
                +------+
```
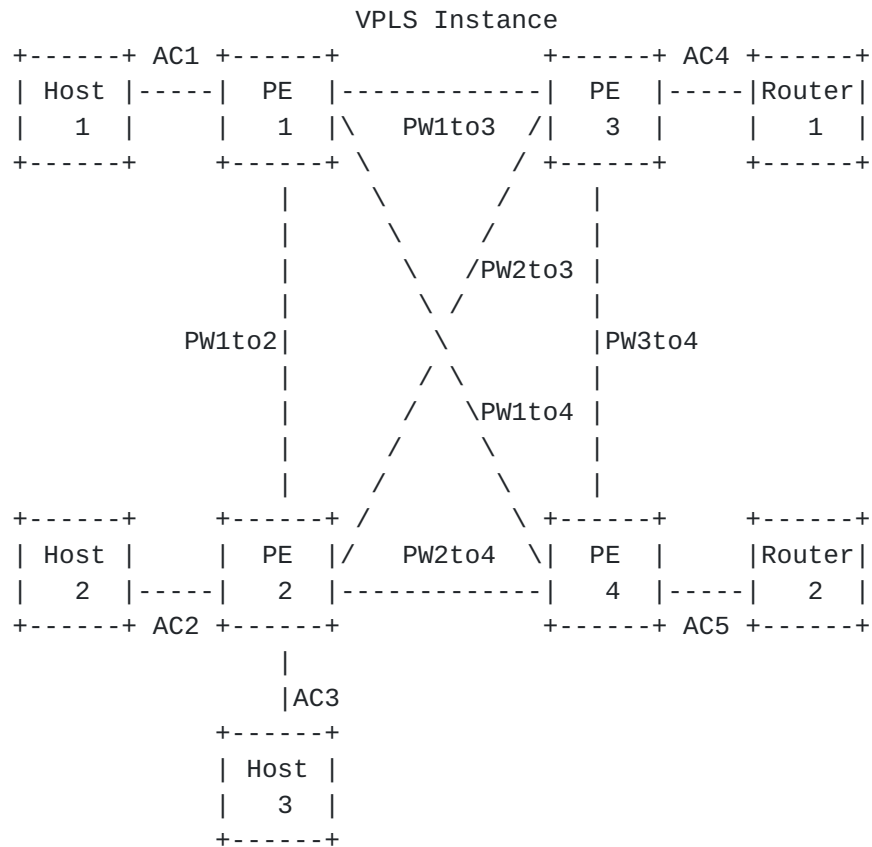
            Figure 1 Reference Diagram for IGMP Snooping for VPLS

[5.3.1](5.3.1). Discovering Multicast Routers

  A PE need to discover the multicast routers in VPLS instances. This
  is necessary because:
    -    Designated Router can be different from the Querier on a LAN.
    -    It is not always the Querier that initiates PIM joins
    -    Multicast traffic to the LAN could arrive from a non-querying
    router because it could be the closest to the source.

  As recommended in [MAGMA-SNOOP], the PEs can discover multicast
  routers using Multicast Router Discovery Protocol or they can be
  statically configured. Since multicast routing protocols other than
  PIM is out scope, multicast routers can also be discovered by
  snooping PIM Hello packets as described in Section 5.4.2. .

[5.3.2](5.3.2). IGMP Snooping Protocol State

The IGMP snooping mechanism described here builds the following state
on the PEs.

For each VPLS Instance
     o Set of Multicast Routers (McastRouters) in the VPLS instance
         using mechanisms listed in Section 5.2.1.
     o The IGMP Querying Router (Querier) in the VPLS instance.

For each Group entry (*,G) or Source Filtering entry (S,G) in a VPLS
instance

     o Set of interfaces (ACs and/or PWs) from which IGMP membership
         reports were received. For (*,G) entries, we will call this
         set igmp_include(*,G). For (S,G) entries, we will call this
         set igmp_include(S,G).
     o Set of interfaces from which IGMPv3 hosts have requested to
         not receive traffic from the specified sources. We will call
         this set igmp_exclude(S,G).

On each interface I, for each (*,G) or (S,G) entry
     o A Group Timer (GroupTimer(*,G,I)) representing the hold-time
         for each downstream (*,G) report received on interface I.
     o A Source Timer (SrcTimer(S,G,I)) representing the hold-time
         for each downstream (S,G) report received on interface I.

[5.3.3](5.3.3). IGMP Join

The IGMP snooping mechanism for joining a multicast group (for all
IGMP versions) works as follows:

   -    The PE does querier election (by tracking query messages and
   the source IP addresses) to determine the Querier when there are
   multiple routers present. Additionally, the query must be received
   with a non-zero source-ip-address to perform the Querier election
   -    At this point all PEs learn the place of the Querier. For
   instance, for PE 1 it is behind PW1to3, for PE 2 behind PW2to3,
   for PE 3 behind AC4, for PE 4 behind PW3to4.
   -    The Querier sends a membership query on the LAN. The
   membership query can be either general query or group specific
   query.
   -    PE 3 replicates the query message and forwards it to all PEs
   participating in the VPLS instance (i.e., PE 1, PE 2, PE 4).
   -    PE 3 keeps a state of {[McastRouters: AC4, PW3to4], [Querier:
   AC4]}.
   -    All PEs then forward the query to ACs which are part of the
   VPLS instance.
   -    Suppose that all hosts (Host 1, Host 2, and Host 3) want to
   participate in the multicast group.

-    Host 2 first (for the sake of the example) sends a membership report to the multicast group (e.g., IP: 239.1.1.1, MAC: 01-00-5E-01-01-01), of which Host 2 wants to be a member.

-    PE 2 replicates the membership report message and forwards it to all PEs participating in the VPLS instance (i.e., PE 1, PE3, PE 4).
-    PE 2 notes that there is a directly connected host, which is willing to participate in the multicast group and updates its state to {[McastRouters: PW2to3, PW2to4], [Querier: PW2to3], [igmp_include(*,G):AC2; GroupTimer(*,G,AC2)=GMI]}.

Guideline 1: A PE MUST forward a membership report message to ACs that are part of "McastRouters" state. This is necessary to avoid report suppression for other members in order for the PEs to construct correct states and to not have any orphan receiver hosts.

There are still some scenarios that can result in orphan receivers. For instance, a multicast router and some hosts could be connected to a customer layer-2 switch, and that layer-2 switch can be connected to a PE via an AC. In such scenario, the customer layer-2 switch MUST perform IGMP snooping as well, and it MUST NOT forward the IGMP report messages coming from the PE to the hosts directly connected to it. There can be some cases such that the layer-2 switch does not have IGMP snooping capability or that device is a dummy hub/bridge. In such cases, one can statically configure the AC, through which the IGMP incapable layer-2 device is connected, to be a (S,G)/(*,G) member on the PE. This way, multicast traffic will always be sent to the hosts connected to that layer-2 device, even they do not send joins because of join suppression.

Continuing with the example:

-    PE 2 does not forward the membership report of Host 2 to Host 3.
-    Per the guideline above, PE 1 does not forward the membership report of Host 2 to Host 1.
-    Per the guideline above, PE 3 does forward the membership report of Host 2 to Router 1 (the Querier).
-    PE 3 notes that there is a host in the VPLS instance, which is willing to participate in the multicast group and updates its state to {[McastRouters: AC4, PW3to4], [Querier: AC4], [igmp_include(*,G): PW2to3], [GroupTimer(*,G,PW2to3)=GMI]} regardless of the type of the query.
-    Let us assume that Host 1 subsequently sends a membership report to the same multicast group.

- PE 1 replicates the membership report message and forwards it to all PEs participating in the VPLS instance (i.e., PE 2, PE 3, PE 4).

- PE 1 notes that there is a directly connected host, which is willing to participate in the multicast group. Basically, it keeps a state of {[McastRouters: PW1to3, PW1to4], [Querier: PW1to3], [igmp_include(*,G): AC1,PW1to2], [GroupTimer(*,G,AC1)=GMI]}.
- Per Guideline 1, PE 2 does not forward the membership report of Host 1 to Host 2 and Host 3.
- PE 3 and PE 4 receive the membership report message of Host 1 and check their states. Per Guideline 1, they send the report to Router 1 and Router 2 respectively. They also update their states to reflect Host 1.
- Now, Host 3 sends a membership report to the same multicast group.
- PE 2 updates its state to {[McastRouters: PW2to3, PW2to4], [Querier: PW2to3], [igmp_include(*,G): AC2,AC3,PW1to2], GroupTimer(*,G,AC3)=GMI]}. It then floods the report message to all PEs participating in the VPLS instance. Per Guideline 1, PE 3 forwards the membership report of Host 3 to Router 1, and PE 4 forwards the membership report of Host 3 to Router 2.

At this point, all PEs have necessary states to ensure that no multicast traffic will be sent to sites with no members.

The previous steps work the same way for IGMPv1 and IGMPv2, when the query is general or source specific.

The group and source specific query for IGMPv3 is considered separately below. In IGMPv3, there is no simple membership join or leave report. IGMPv3 reports are one of IS_INCLUDE, IS_EXCLUDE, ALLOW, BLOCK, TO_INCLUDE, TO_EXCLUDE. The PEs MUST implement the "router behavior" portion of the state machine defined in Section 6 of [RFC3376].

The IGMP snooping mechanism for joining a multicast group (for IGMPv3) works as follows:

- The Querier sends a membership query to the LAN. The membership query is group and source specific query with a list of sources (e.g., S1, S2, .., Sn).
- PE 3 replicates the query message and forwards it to all PEs participating in the VPLS instance (i.e., PE 1, PE 2, PE 4).
- PE 3 keeps a state of {[McastRouters: AC4, PW3to4], [Querier: AC4]}.

-    All PEs then forward the query to ACs which are part of the
VPLS instance.
-    Suppose that all hosts (Host 1, Host 2, and Host 3) want to
participate in the multicast group. Host 1 and Host 2 want to
subscribe to (Sn,G), and Host 3 wants to subscribe to (S3,G).
-    Host 2 first (for the sake of the example) sends a membership
report message with group record type IS_INCLUDE for (Sn,G).
-    PE 2 replicates the membership report message and forwards it
to all PEs participating in the VPLS instance (i.e., PE 1, PE 3,
PE 4).
-    PE 2 notes that there is a directly connected host, which is
willing to participate in the multicast group and updates its
state to {[McastRouters: PW2to3, PW2to4], [Querier: PW2to3],
[igmp_include(Sn,G): AC2], [SrcTimer(Sn,G,AC2)=GMI]}.
-    Per Guideline 1, PE 2 does not forward the membership report
of Host 2 to Host 3.
-    Per Guideline 1, PE 1 does not forward the membership report
of Host 2 to Host 1.
-    Per Guideline 1, PE 3 does forward the membership report of
Host 2 to Router 1 (the Querier).
-    Per Guideline 1, PE 4 does forward the membership report of
Host 2 to Router 2.
-    PE 3 notes that there is a host in the VPLS instance, which
is willing to participate in the multicast group. Basically, it
updates its state to {[McastRouters: AC4, PW3to4], [Querier: AC4],
[igmp_include(Sn,G): PW2to3], [SrcTimer(Sn,G,PW2to3)=GMI]}.
-    Likewise, PE 4 updates its state to {[McastRouters: PW3to4,
AC5], [Querier: PW3to4], [igmp_include(Sn,G):PW2to4],
[SrcTimer(Sn,G,PW2to4)=GMI]}.
-    Let us say Host 1 now sends a membership report message with
group record type IS_INCLUDE for (Sn,G).
-    Similar procedures are followed by PEs as explained in the
previous steps. For instance, PE 1 updates its state to
{[McastRouters: PW1to3, PW1to4], [Querier: PW1to3],
[igmp_include(Sn,G): PW1to2, AC1], SrcTimer(Sn,G,AC1)=GMI}. PE 3
updates its state to {[McastRouters: AC4, PW3to4], [Querier: AC4],
[(S1,G); Hosts:], [igmp_include(Sn,G): PW2to3, PW1to3],
[SrcTimer(Sn,G,PW1to3)=GMI]}.
-    Finally, Host 3 sends a membership report message with group
record type IS_INCLUDE for (S3,G).
-    PE 2 replicates the membership report message and forwards it
to all PEs participating in the VPLS instance (i.e., PE 1, PE 3,
PE 4).
-    Per Guideline 1, PE 2 does not forward the membership report
of Host 3 to Host 2.

-     Per Guideline 1, PE 1 does not forward the membership report
     of Host 3 to Host 1.
-     Per Guideline 1, PE 3 does forward the membership report of
     Host 3 to Router 1.
-     Per Guideline 1, PE 4 does forward the membership report of
     Host 3 to Router 2.
-     All PEs update their states accordingly. For instance, PE 2
     updates its state to {[McastRouters: PW2to3, PW2to4], [Querier:
     PW2to3], [igmp_include(S3,G): AC3], [igmp_include(Sn,G): PW1to2,
     AC2], [SrcTimer(S3,G,AC3)=GMI]}. PE 4 updates its state to
     {[McastRouters: AC5, PW3to4], [Querier: PW3to4],
     [igmp_include(S3,G): PW2to4], [igmp_include(Sn,G): PW1to4,
     PW2to4], [SrcTimer(S3,G,PW2to4)=GMI]}.

At this point, all PEs have necessary states to not send multicast
traffic to sites with no members.

Based on above description of IGMPv3 based snooping for VPLS, one may
conclude that the PEs MUST have the capability to store (S,G) state
and MUST forward/replicate traffic accordingly. This is, however,
not MANDATORY. A PE MAY only keep (*,G) based states rather than on
a per (S,G) basis with the understanding that this will result in a
less efficient IP multicast forwarding within each VPLS instance.

Guideline 2: If a PE receives unsolicited report message and if it
does not possess a state for that particular multicast group, it MUST
flood that unsolicited membership report message to all PEs
participating in the VPLS instance, as well as to the multicast
router if it is locally attached.

5.3.4. IGMP Leave

The IGMP snooping mechanism for leaving a multicast group works as
follows:

-     In the case of IGMPv2, when a PE receives a leave (*,G)
     message from a host via its AC, it lowers the corresponding
     GroupTimer(*,G,AC) to "Last Member Query Time" (LMQT).
-     In the case of IGMPv3, when a PE receives a membership report
     message with group record type of IS_EXCLUDE or TO_EXCLUDE or
     BLOCK for (S,G) from a host via its AC, it lowers the
     corresponding SrcTimer(S,G,AC) for all affected (S,G)s to LMQT.

In the following guideline, a "leave (*,G)/(S,G) message" also means
IGMPv3 membership report message with group record type of IS_EXCLUDE
or TO_EXCLUDE or BLOCK for (S,G).

Guideline 3: A PE MUST NOT forward a leave (*,G)/(S,G) message to
ACs participating in the VPLS instance, If the PE still has
locally connected hosts or hosts connected over a H-VPLS spoke in
its state.

Guideline 4: A PE MUST forward a leave (*,G)/(S,G) message to all
PEs participating in the VPLS instance. A PE MAY forward the
leave (*,G)/(S,G) message to the "McastRouters" ONLY, if there are
no member hosts in its state.

Guideline 5: If a PE does not receive a (*,G) membership report
from an AC before GroupTimer(*,G,AC) expires, the PE MUST remove
the AC from its state. In case of IGMPv3, if a PE does not
receive a (S,G) membership report from an AC before the
SrcTimer(S,G,AC) expires, the PE MUST remove the AC from its
state.

[5.3.5](5.3.5). Failure Scenarios

Up to now, we did not consider any failures, which we will focus in
this section.

-     In case the Querier fails (e.g., AC to the querier fails),
another router in the VPLS instance will be selected as the
Querier. The new Querier will be sending queries. In such
circumstances, the IGMP snooping states in the PEs will be
updated/overwritten by the same procedure explained above.
-     In case a Multicast router fails, the IGMP snooping states in
the PEs will be updated/overwritten by the multicast router
discovery procedures provided in Section 5.3.1. .
-     In case a host fails (e.g., AC to the host fails), a PE
removes the host from its IGMP snooping state for that particular
multicast group. Guidelines 3, 4 and 5 still apply here.
-     In case a PW (which is in IGMP snooping state) fails, the PEs
will remove the PW from their IGMP snooping state. For instance,
if PW1to3 fails, then PE 1 will remove PW1to3 from its state as
the Querier connection, and PE 3 will remove PW1to3 from its state
as one of the host connections. Guidelines 3, 4 and 5 still apply
here. After PW is restored, the IGMP snooping states in the PEs
will be updated/overwritten by the same procedure explained above.
One can implement a dead timer before making any changes to IGMP
snooping states upon PW failure. In that case, IGMP snooping
states will be altered if the PW can not be restored before the
dead timer expires.

5.3.6. **Scaling Considerations for IGMP Snooping**

  In scenarios where there are multiple ACs connected to a PE, it is
  quite likely that IGMP membership reports for the same group are
  received from multiple ACs. The normal behavior would be to have each
  of the membership reports sent to McastRouters. But in scenarios
  where many ACs send IGMP membership reports to the same groups, the
  burden on all the other PEs can be overwhelming. To make matters
  worse, there can be a large number of hosts on the same AC that all
  request IGMP membership reports to the same group. While [IGMPv2]
  suggests the use of report suppression, [IGMPv3] does not.
  Regardless, if hosts do not implement report suppression, this can be
  a scaling issue on the PEs. This section outlines the optimization
  suggested in [MAGMA-SNOOOP] to perform proxy-querying and proxy-
  reporting function on the PEs to avoid report explosion.

  For this optimization, we separate out the IGMP group state on the
  PEs into downstream state and upstream state.

  Note that the following sub-sections describe the procedures for
  (*,G). The same procedures must be be extended to (S,G)s. Furthermore
  the behavior described is for a downstream PE. While it is very
  important for downstream PEs to implement the proxy behavior
  described here, the scalability issues are not as bad on upstream
  PEs. Optimizing upstream PEs would be designed to alleviate the
  burden on the upstream CEs. Nevertheless the same procedures can be
  applied to upstream PEs as well as an added optimization. The only
  difference would be that ACs would be upstream interface(s) and PWs
  would be downstream interface(s) for such PEs.

5.3.7. **Downstream Proxy Behavior**

  When a IGMP membership Report for a group is received on an AC, the
  PE adds the AC to the corresponding igmp_include set and resets the
  GrpTimer to GMI.

  When an IGMP membership Leave for a group is received on an AC, the
  PE lowers the corresponding GrpTimer to LMQT and sends out a proxy
  group-specific query on that AC. When sending the group-specific
  query, the PE encodes 0.0.0.0 (or :: in case of IPv6) in the source-
  ip address field. If there is no other host interested in that group,
  then the AC is removed from the corresponding igmp_include set after
  the GrpTimer expires.

5.3.8. **Upstream Proxy Behavior**

  When the igmp_include set for a group becomes non-null, the PE sends
  out a proxy IGMP Join report for that group to McastRouters. When the
  igmp_include set for a group becomes empty, the PE sends out a proxy
  IGMP Leave report for that group to McastRouters.

  When the PE receives a general query, it replies with its current
  snooping state for all groups and group-sources. It also forwards the
  general query to all ACs thus removing the need for proxy general
  queries. When the PE receives a group-specific or group-source
  specific query, the PE does not forward such queries to the ACs.
  Instead it replies with a proxy report if it has snooping state for
  that group or group-source. When sending the proxy report, the PE
  encodes 0.0.0.0 (or :: in the case of IPv6) in the source-ip address
  field.

5.4. **PIM Snooping for VPLS**

  IGMP snooping procedures described above provide efficient delivery
  of IP multicast traffic in a given VPLS service when end stations are
  connected to the VPLS. However, when VPLS is offered as a WAN
  service it is likely that the CE devices are routers and would run
  PIM between them. To provide efficient IP multicasting in such
  cases, it is necessary that the PE routers offering the VPLS service
  do PIM snooping. This section describes the procedures for PIM
  snooping.

  PIM is a multicast routing protocol, which runs exclusively between
  routers. PIM shares many of the common characteristics of a routing
  protocol, such as discovery messages (e.g., neighbor discovery using
  Hello messages), topology information (e.g., multicast tree), and
  error detection and notification (e.g., dead timer and designated
  router election). On the other hand, PIM does not participate in any
  kind of exchange of databases, as it uses the unicast routing table
  to provide reverse path information for building multicast trees.
  There are a few variants of PIM. In PIM-DM ([PIM-DM]), multicast
  data is pushed towards the members similar to broadcast mechanism.
  PIM-DM constructs a separate delivery tree for each multicast group.
  As opposed to PIM-DM, other PIM versions (PIM-SM [RFC2362], PIM-SSM
  [PIM-SSM], and BIDIR-PIM [BIDIR-PIM]) invokes a pull methodology
  instead of push technique.

  PIM routers periodically exchange Hello messages to discover and
  maintain stateful sessions with neighbors. After neighbors are
  discovered, PIM routers can signal their intentions to join/prune
  specific multicast groups. This is accomplished by having downstream
  routers send an explicit join message (for the sake of

generalization, consider Graft messages for PIM-DM as join messages)

to the upstream routers. The join/prune message can be group
specific (*,G) or group and source specific (S,G).

In PIM snooping, a PE snoops on the PIM message exchange between
routers, and builds its multicast states. Based on the multicast
states, it forwards IP multicast traffic accordingly to avoid
unnecessary flooding.

5.4.1. **PIM Snooping State Summarization Macros**

The following sets are defined to help build the forwarding state on
a PE. Some sets may apply only to a subset of the PIM Protocol
variants as noted along with the definition of the sets.

pim_joins(*,G) =
Set of all downstream interfaces on which PIM (*,G) Joins are
received. This set applies only to PIM-SM, PIM-SSM, PIM-BIDIR.

pim_joins(S,G) =
Set of all downstream interfaces on which PIM (S,G) Joins are
received. This set applies only to PIM-SM, PIM-SSM.

All_Pim_DM_OifList =
If the upstream interface (the interface towards the upstream PIM
neighbor) is a PW, then this set is the set of all ACs on which there
are PIM neighbors. If the upstream interface is an AC, then this is
the set of all interfaces (both AC and PW) on which there are PIM
neighbors. This set applies only to PIM-DM.

pim_prunes(S,G) =
Set of all downstream interfaces on which PIM (S,G) prunes are
received. This set applies only to PIM-DM.

pim_prunes(S,G,rpt) =
Set of all downstream interfaces on which PIM (S,G,rpt) prunes are
received. This set applies only to PIM-SM.

Pim_oiflist(*,G) =
Set of interfaces that PIM contributes to the list of outgoing
interfaces to which data traffic must be forwarded on a (*,G) match.

Pim_oiflist(S,G) =
Set of interfaces that PIM contributes to the list of outgoing
interfaces to which data traffic must be forwarded on an (S,G) match.

Note that pim_oiflist is not the complete list of outgoing interfaces
(oiflist). IGMP/MLD also contribute to this list.

For PIM-DM,

    pim_oiflist(S,G) = All_Pim_DM_OifList (-) pim_prunes(S,G)

For PIM-SM and PIM-SSM,

 Pim_inherited_oiflist(S,G,rpt) = pim_joins(*,G) (-)
                                        pim_prunes(S,G,rpt)

 pim_oiflist(*,G) = pim_joins(*,G)

 pim_oiflist(S,G) = pim_inherited_oiflist(S,G,rpt) (+)
                    pim_joins(S,G)

For PIM-BIDIR,

 Pim_oiflist(*,G) = DF(RP(G)) + pim_joins(*,G)
Where DF(RP(G)) is the AC/PW towards the router that is the
designated forwarder for RP(G).

In the following, the mechanisms involved for implementing PIMv2
([RFC2362]) snooping in VPLS are specified. PIMv1 is out of the
scope of this document. Please refer to Figure 2 as an example of
VPLS with PIM snooping.


```
                          VPLS Instance
      +------+ AC1 +------+              +------+ AC4 +------+
      |Router|-----|  PE  |-------------|  PE  |-----|Router|
      |  1   |     |  1   |\   PW1to3  /|  3   |     |  4   |
      +------+     +------+ \          / +------+     +------+
                    |     \        /     |
                    |      \      /      |
                    |       \   /PW2to3  |
                    |        \ /         |
              PW1to2|         \          |PW3to4
                    |        / \         |
                    |       /   \PW1to4  |
                    |      /     \       |
                    |     /       \      |
      +------+     +------+ /       \ +------+     +------+
      |Router|     |  PE  |/   PW2to4  \|  PE  |     |Router|
      |  2   |-----|  2   |-------------|  4   |-----|  5   |
      +------+ AC2 +------+              +------+ AC5 +------+
                    |
                    |AC3
              +------+
              |Router|
              |  3   |
              +------+
```
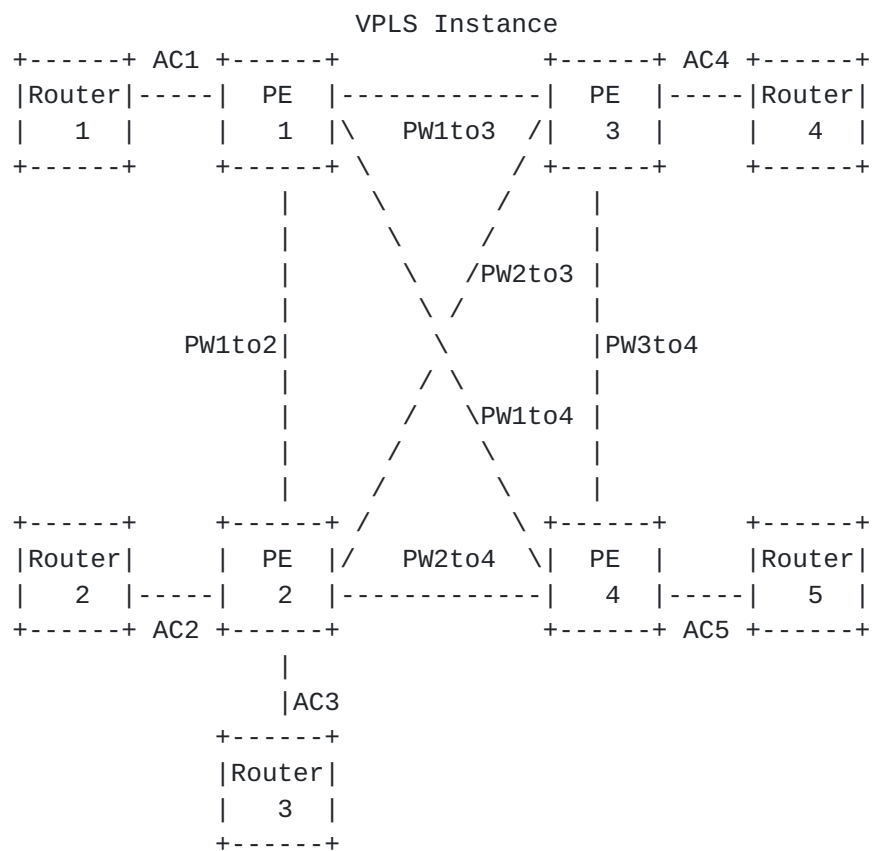
Figure 2 Reference Diagram for PIM Snooping for VPLS

In the following sub-sections, snooping mechanisms for each variety of PIM are specified.

5.4.2. PIM-DM

The characteristics of PIM-DM is flood and prune behavior. Shortest path trees are built as a multicast source starts transmitting.

In Figure 2, the multicast source is behind Router 4, and all routers have at least one receiver except Router 3 and Router 5.

5.4.2.1. Discovering Multicast Routers

The PIM-DM snooping mechanism for neighbor discovery works as follows:

-     To establish PIM neighbor adjacencies, PIM multicast routers (all routers in this example) send PIM Hello messages to the ALL PIM Routers group address (IPv4: 224.0.0.13, MAC: 01-00-5E-00-00-0D) on every PIM enabled interfaces. The IPv6 ALL PIM Routers group is "ff02::d". In addition, PIM Hello messages are used to elect Designated Router for a multi-access network. In PIM-DM, the DR acts as the Querier if IGMPv1 is used. Otherwise, DR has no function in PIM-DM.

Guideline 6: PIM Hello messages MUST be flooded in the VPLS instance. A PE MUST populate its "PIM Neighbors" list according to the snooping results. This is a general PIM snooping guideline and applies to all variants of PIM snooping.

Guideline 7: For PIM-DM only. pim_oiflist(S,G) is populated with All_Pim_DM_Interfaces (the ACs/PWs in the "PIM Neighbors" list). Changes to the "PIM Neighbors" list MUST be replicated to All_Pim_DM_Interfaces.

-     Every router starts sending PIM Hello messages. Per Guideline 6, every PE replicates Hello messages and forwards them to all PEs participating in the VPLS instance.
-     Based on PIM Hello exchanges PE routers populate PIM snooping states as follows. PE 1: {[(,); Source:; Flood to: AC1, PW1to2, PW1to3, PW1to4], [PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)] }, PE 2: {[(,); Source:; Flood to: AC2, AC3, PW1to2, PW2to3, PW2to4], [PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[(,); Source:; Flood to: AC4, PW1to3, PW2to3, PW3to4], [PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4), (Router

5,PW3to4)]}, PE 4: {[(,); Source:; Flood to: AC5, PW1to4, PW2to4, PW3to4], [PIM Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}. The original pim_oiflist(S,G) is populated with ACs/PWs in the PIM neighbor list per Guideline 7..
-    PIM Hello messages contain a Holdtime value, which tells the receiver when to expire the neighbor adjacency (which is three times the Hello period).

Guideline 8: If a PE does not receive a Hello message from a router within its Holdtime, the PE MUST remove that router from the PIM snooping state. If a PE receives a Hello message from a router with Holdtime value set to zero, the PE MUST remove that router from the PIM snooping state immediately. PEs MUST track the Hello Holdtime value per PIM neighbor.

[5.4.2.2](#). PIM-DM Multicast Forwarding

The PIM-DM snooping mechanism for multicast forwarding works as follows:

-    When the source starts sending traffic to multicast group (S,G), PE 3 updates its state to PE 3: {[(S,G) ; Source: (Router 4,AC4); pim_oiflist(S,G): PW1to3, PW2to3, PW3to4], [PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]}. AC4 is removed from the pim_oiflist list for (S,G), since it is where the multicast traffic comes from.

Guideline 9: Multicast traffic MUST be replicated per PW and AC basis, i.e., even if there are more than one PIM neighbor behind a PW/AC, only one replication MUST be sent to that PW/AC.

-    PE 3 replicates the multicast traffic and sends it to the other PE routers in its pim_oiflist(S,G).
-    Consequently, all PEs update their states as follows. PE 1: {[(S,G); Source: (Router 4,PW1to3); pim_oiflist(S,G): AC1], [PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2: {[(S,G); Source: (Router 4,PW2to3); pim_oiflist(S,G): AC2, AC3], [PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)]}, PE 4: {[(S,G); Source: (Router 4,PW3to4); pim_oiflist(S,G): AC5], [PIM Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}.

[5.4.2.3](#). PIM-DM Pruning

At this point all the routers (Router 1, Router 2,Router 3, Router 5)
receive the multicast traffic.

- However, Router 3 and Router 5 do not have any members for
that multicast group, so they send prune messages to leave the
multicast group to the ALL PIM Routers group. PE 2 updates its
state to PE 2: {[(S,G); Source: (Router 4,PW2to3);
pim_prunes(S,G): AC3, pim_oiflist(S,G): AC2], [PIM Neighbors:
(Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
4,PW2to3), (Router 5,PW2to4)]}. PE 4 also removes Router 3 and
Router 5 from its state as well.

Guideline 10:.The PIM-DM prune message MUST be forwarded towards
the upstream PE only if pim_oiflist(S,G) became empty as a result
of the received prune message. If pim_oiflist(S,G) was already
null when the PIM-DM prune was received, then the prune MUST NOT
be forwarded upstream.

- PE 2 does not forward the prune message per Guideline 10. PE
4 updates its state to PE 4: {[(S,G); Source: (Router 4,AC4);
pim_prunes(S,G): AC5, pim_oiflist(S,G):], [PIM Neighbors:
(Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router4, PW3to4).
- PIM-DM prune messages contain a Holdtime value, which
specifies how many seconds the prune state should last.

Guideline 11: For PIM-DM only. A PE MUST keep the prune state for
a PW/AC according to the Holdtime in the prune message, unless a
corresponding Graft message is received.

- Upon receiving the prune messages, each PE 3 updates its
state accordingly to PE 3: {[(S,G); Source: (Router 4,AC4);
pim_prunes(S,G): PW2to4, pim_oiflist(S,G): PW1to3, PW2to3],
[PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3),
(Router 4,AC4), (Router 5, PW3to4)]}.

Guideline 12: For PIM-DM only. To avoid overriding joins, a PE
SHOULD suppress the PIM prune messages to directly connected
routers (i.e., ACs), as long as there is a PW/AC in its
corresponding pim_oiflist(S,G).

- In this case, PE 1, PE 2, and PE 3 do not forward the prune
messages to their directly connected routers.

5.4.2.4. PIM-DM Grafting

The multicast traffic is now flowing only to points in the network
where receivers are present.

Guideline 13: For PIM-DM only. A PE MUST remove the AC/PW from
its corresponding prune state (pim_prunes(S,G)) when it receives a
graft message from the AC/PW. That is, the corresponding AC/PW
MUST be added to the pim_oiflist(S,G) list.

Guideline 14: For PIM-DM only. PIM-DM graft messages MUST be
forwarded based on the destination MAC address. If the
destination MAC address is 01-00-5E-00-00-0D, then the graft
message MUST be flooded in the VPLS instance. PIM-DM graft
messages MUST NOT be flooded if pim_oiflist is non-null.

-      For the sake of example, suppose now Router 3 has a receiver
the multicast group (S,G). Assuming Router 3 sends a graft
message in IP unicast to Router 4 to restart the flow of multicast
traffic. PE 2 updates its state to PE 2: {[(S,G); Source: (Router
4,PW2to3); pim_prunes(S,G): , pim_oiflist(S,G): AC2, AC3], [PIM
Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3),
(Router 4,PW2to3), (Router 5,PW2to4)]}. PE 2 then forwards the
graft message to PE 3 according to Guideline 14.
-      Upon receiving the graft message, PE 3 updates its state
accordingly to PE 3: {[(S,G); Source: (Router 4,AC4);
pim_prunes(S,G): PW3to4, pim_oiflist(S,G): PW1to3, PW2to3], [PIM
Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router
4,AC4), (Router 5,PW3to4)]}.

[5.4.2.5](5.4.2.5). Failure Scenarios

Guideline 15: PIM Assert messages MUST be flooded in the VPLS
instance.

Guideline 16: If an AC/PW goes down, a PE MUST remove it from its
PIM snooping state.

Failures can be easily handled in PIM-DM snooping, as it uses push
technique. If an AC or a PW goes down, PEs in the VPLS instance will
remove it from their snooping state (if the AC/PW is not already
pruned). After the AC/PW comes back up, it will be automatically
added to the snooping state by PE routers, as all PWs/ACs MUST be in
the snooping state, unless they are pruned later on.

[5.4.3](5.4.3). PIM-SM

The key characteristics of PIM-SM is explicit join behavior. In this
model, the multicast traffic is only sent to locations that
specifically request it. The root node of a tree is the Rendezvous
Point (RP) in case of a shared tree or the first hop router that is

directly connected to the multicast source in the case of a shortest path tree.

In Figure 2, the RP is behind Router 4, and all routers have at least one member except Router 3 and Router 5.

As in the case with IGMPv3 snooping, we assume that the PEs have the capability to store (S,G) states for PIM-SM snooping and forward/replicate traffic accordingly. This is not mandatory. An implementation, can fall back to (*,G) states, if its hardware can not support it. In such case, the efficiency of multicast forwarding will be less.

[5.4.3.1](5.4.3.1). Discovering Multicast Routers

The PIM-SM snooping mechanism for neighbor discovery works the same way as the procedure defined in PIM-DM section, with the exception of PIM-DM only guidelines.

   -    Based on PIM Hello exchanges PE routers populate PIM snooping states as follows. PE 1: {[(,); Flood to:], [PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2: {[(,); Flood to:], [PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[(,); Flood to:], [PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]}, PE 4: {[(,); Flood to:], [PIM Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}.

To reduce the amount of PIM Join/Prune traffic in the VPLS network, it is important that Explicit-Tracking capability be disabled between the CEs. If a CE advertises tracking support, it is recommended that the PEs modify the tracking-support option in CE Hello packets before forwarding them to ensure that tracking support is disabled between the CEs. Otherwise, the mechanism listed for "JP_Optimization" throughout the PIM-SM and PIM-SSM sections of this document MUST NOT be employed.

NOTE: The examples below are for scenarios where JP_Optimization is not employed.

For PIM-SM to work properly, all routers within the domain must use the same mappings of group addresses to RP addresses. Currently, there are three methods for RP discovery: 1. Static RP configuration, 2, Auto-RP, and 3. PIMv2 Bootstrap Router mechanism.

[5.4.3.2](5.4.3.2). PIM-SM (*,G) Join

The PIM-SM snooping mechanism for joining a multicast group (*,G) works as follows:

    Guideline 18: PIM-SM join messages MUST be sent only to the remote
    PE, which is connected to the router to which the Join is
    addressed.
    JP_Optimization: The PIM-SM join message MUST be forwarded towards
    the upstream CE only if pim_joins(*,G) became non-empty as a
    result of the received join message. If pim_joins(*,G) was already
    non-null when the PIM-SM join was received, then the join MUST NOT
    be forwarded upstream.

PIM-SM join messages MUST be sent only to the remote PE, which is
connected to the router to which the Join is addressed. The remote
PE can be determined by the "Upstream Neighbor Address" field of the
Join message. The "Upstream Neighbor Address" can be correlated to a
PW or an AC in the "PIM Neighbors" state. By Guideline 18, we are
ensuring that the other routers that are part of the VPLS instance do
not receive the PIM join messages and will initiate their own join
messages if they are interested in receiving that particular
multicast traffic.

    -    Assume Router 1 wants to join the multicast group (*,G) sends
    a join message for the multicast group (*,G). PE 1 sends the join
    message to PE 3 by Guideline 18.

    Guideline 19: A PE MUST add a PW/AC to its pim_joins(*,G) list, if
    it receives a (*,G) join message from the PW/AC.

    -    PE 1 updates their states as follows: PE 1: {[pim_joins(*,G):
    AC1], [PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2),
    (Router 4,PW1to3), (Router 5,PW1to4)]}.

A periodic refresh mechanism is used in PIM-SM to maintain the proper
state. PIM-SM join messages contain a Holdtime value, which
specifies for how many seconds the join state should be kept.

    Guideline 20: If a PE does not receive a refresh join message from
    a PW/AC within its Holdtime, the PE MUST remove the PW/AC from its
    pim_joins(*,G) list.

    -    All PEs update their states accordingly as follows: PE 1:
    {[pim_joins(*,G): AC1], [PIM Neighbors: (Router 1,AC1), (Router
    2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2:
    {[(,); Flood to:], [PIM Neighbors: (Router 1,PW1to2), (Router
    2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)]}, PE
    3: {[pim_joins(*,G): PW1to3], [PIM Neighbors: (Router 1,PW1to3),

(Router 2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]},
PE 4: {[(,); Flood to:], [PIM Neighbors: (Router 1,PW1to4),
(Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}.
-     After Router 2 joins the same multicast group, the states
become as follows: PE 1: {[pim_joins(*,G): AC1], [PIM Neighbors:
(Router 1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3),
(Router 5,PW1to4)]}, PE 2: {[pim_joins(*,G): AC2], [PIM Neighbors:
(Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[pim_joins(*,G): PW1to3,
PW2to3], [PIM Neighbors: (Router 1,PW1to3), (Router 2,Router
3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]}, PE 4: {[(,); Flood
to:], [PIM Neighbors: (Router 1,PW1to4), (Router 2,Router
3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}.
-     For the sake of example, Router 3 joins the multicast group.
PE 2 sends the join message to PE 3.
-     Next Router 5 joins the group, and the states are updated
accordingly: PE 1: {[pim_joins(*,G): AC1], [PIM Neighbors: (Router
1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router
5,PW1to4)]}, PE 2: {[pim_joins(*,G): AC2, AC3], [PIM Neighbors:
(Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[pim_joins(*,G): PW1to3,
PW2to3, PW3to4], [PIM Neighbors: (Router 1,PW1to3), (Router
2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]}, PE 4:
{[pim_joins(*,G): AC5],[PIM Neighbors: (Router 1,PW1to4), (Router
2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}

At this point, all PEs have necessary states to not send multicast
traffic to sites with no members.

[5.4.3.3](5.4.3.3). PIM-SM Pruning

The PIM-SM snooping mechanism for leaving a multicast group works as
follows:
-     Assume Router 5 sends a prune message.

Guideline 21: PIM-SM prune messages MUST be flooded in the VPLS
instance.
JP_Optimization: Instead of the above guideline, a PE MUST forward
prune messages only towards the upstream CE and only if
pim_joins(*,G) became empty as a result of the received prune
message. If pim_joins(*,G) is non-empty after receiving the prune
message, the PE MUST NOT forward the prune message.

Guideline 22: A PE MUST remove a PW/AC from its pim_joins(*,G)
list if it receives a (*,G) prune message from the PW/AC. A
prune-delay timer SHOULD be implemented to support prune override.

However, the prune-delay timer is not required if there is only
one PIM neighbor on that AC/PW on which the prune was received.

-    PE 4 floods the (*,G) prune to the VPLS instance per
Guideline 21. PE routers participating in the VPLS instance also
forward the (*,G) prune to the ACs, which are connected to the
VPLS instance. The states are updated as follows: PE 1:
{[pim_joins(*,G): AC1], [PIM Neighbors: (Router 1,AC1), (Router
2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2:
{[pim_joins(*,G): AC2, AC3], [PIM Neighbors: (Router 1,PW1to2),
(Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router
5,PW2to4)]}, PE 3: {[pim_joins(*,G): PW1to3, PW2to3], [PIM
Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router
4,AC4), (Router 5,PW3to4)]}, PE 4: {[(,); Flood to:],[PIM
Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router
4,PW3to4), (Router 5,AC5)]}.

In PIM-SM snooping, prune messages are flooded by PE routers. In
such implementation, PE routers may receive overriding join messages,
which will not affect anything.

5.4.3.4. PIM-SM (S,G) Join

The PIM-SM snooping mechanism for source and group specific join
works as follows:

Guideline 23: A PE MUST add a PW/AC to its pim_joins(S,G) list if
it receives a (S,G) join message from the PW/AC. The PE MUST
forward the received join message towards the upstream CE.
JP_Optimization: The PE MUST forward the Join message towards the
upstream neighbor only if the pim_joins(S,G) list becomes non-
empty as a result of the received join. If the pim_joins(S,G) list
was non-empty prior to receiving the join message, then the PE
MUST NOT forward the join message.

Guideline 24: A PE MUST remove a PW/AC from its pim_joins(S,G)
list if it receives a (S,G) prune message from the PW/AC. The PE
MUST flood the prune message in the VPLS instance. A prune-delay
timer SHOULD be implemented to support prune override on the
downstream AC/PW. However, the prune-delay timer is not required
if there is only one PIM neighbor on that AC/PW on which the prune
was received.
JP_Optimization: Instead of flooding the prune message in the VPLS
instance, the PE MUST forward the prune message towards the
upstream neighbor only if the pim_joins(S,G) list becomes empty as
a result of the received prune. If the pim_joins(S,G) list remains

   non-empty after receiving the prune message, then the PE MUST NOT
   forward the prune message.

   Guideline 25: A PE MUST prefer (S,G) state to (*,G), if both S and
   G match.

5.4.3.5. PIM-SM (S,G,rpt) Prunes

   Guideline 28: When a PE receives a Prune(S,G,rpt) on an AC/PW, it
   MUST add the AC/PW to the pim_prunes(S,G,rpt) list. Additionally,
   if pim_snoop_inherited_olist(S,G,rpt) becomes empty, the PE MUST
   forward the Prune(S,G,rpt) towards the upstream neighbor. If
   pim_snoop_inherited_olist(S,G,rpt) is still non-empty, then the PE
   MUST NOT forward the Prunes(S,G,rpt).

5.4.3.6. PIM-SM (*,*,RP) State
  PIM-SM defines a (*,*,RP) state which is used when traffic needs to
  cross multicast domains. A (*,*,RP) receiver requests all multicast
  traffic within a PIM domain to be sent to it. If the two multicast
  domains are both PIM-SM, they can use MSDP to leak multicast routes.
  But, if one is PIM-SM and the other is PIM-DM (hence, MSDP can not be
  used), then the border router would initiate a (*,*,RP) join to all
  RPs in the PIM-SM domain.

  If the customers will configure multiple and different PIM domains,
  PIM-SM snooping MUST support (*,*,RP) state as well. Depending on
  how likely scenario this is, future versions may include (*,*,RP)
  states.

5.4.3.7. Failure Scenarios

  Failures can be easily handled in PIM-SM snooping, as it employs
  state-refresh technique. PEs in the VPLS instance will remove any
  entry for non-refreshing routers from their states.

5.4.3.8. Special Cases for PIM-SM Snooping

  There are some special cases to consider for PIM-SM snooping. First
  one is the RP-on-a-stick. The RP-on-a-stick scenario may occur when
  the Shortest Path Tree and the Shared Tree shares a common Ethernet
  segment, as all routers will be connected over a multicast access
  network (i.e., VPLS). Such a scenario will be handled by PIM-SM
  rules (particularly, the incoming interface can not also appear in
  the outgoing interface list) very nicely. Second scenario is the
  turnaround router. The turnaround router scenario occurs when
  shortest path tree and shared tree share a common path. The router
  at which these tree merge is the turnaround router. PIM-SM handles
  this case by proxy (S,G) join implementation by the turnaround
  router.

There can be some scenarios where CE routers can receive duplicate
multicast traffic. Let us consider the scenario in Figure 3.

```
                               +------+ AC3 +------+
                               | PE2 |-----|  R3  |
                              /|      |     |      |
                             / +------+     +------+
                            /     |            |
                           /      |            |
                      /PW1to2     |            |
                        /         |         +-----+
                       /          |PW2to3   | Src |
                      /           |         +-----+
                     /            |            |
                    /             |            |
                   /              |            |
   +------+    +------+ /     +------+     +------+
   |  R1  |    | PE1 |/  PW1to3  | PE3 |     |  R4  |
   |      |-----|      |------------|      |-----|      |
   +------+ AC1 +------+           +------+ AC4 +------+
               |                      |
               |AC2                   |AC5
           +------+              +------+
           |  R2  |              |  R5  |     +---+
           |      |              |      |-----|RP |
           +------+              +------+     +---+
```
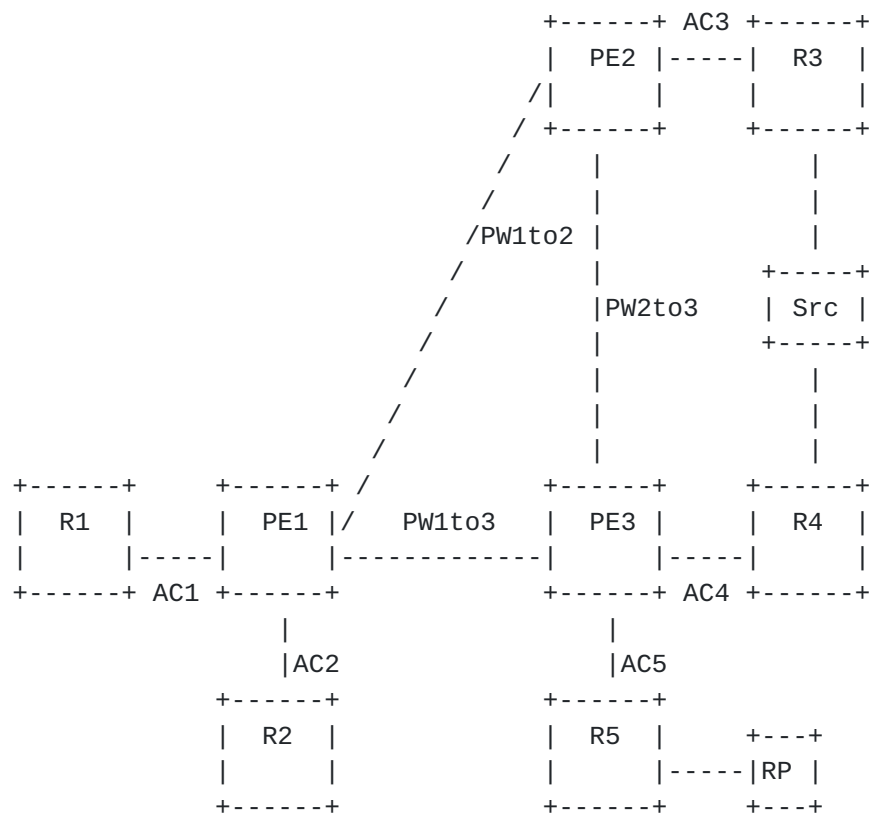
                Figure 3 CE Routers Receive Duplicate Traffic

   In the scenario depicted in Figure 3, both R1 and R2 has two ECMP
   routes to reach the source Src. Hence, R1 may pick R3 as its next
   hop ("Upstream Neighbor"), and R2 may pick R4 as its next hop. As a

result, both R1 and R2 will receive duplicate traffic.

This issue can be solved as follows. PEs can keep the PW/AC that the join message is forwarded to (upstream PW/AC) in "Flood to" list in addition to the PW/AC that the join message is received (downstream PWAC). If the traffic arrives from a different PW/AC, that traffic is not forwarded downstream. Hence, in the example depicted in Figure 3 where source is dual homed to R3 and R4, R1 will receive (S,G) traffic if it comes from PW1to2, and R2 will receive (S,G) traffic if it comes from PW1to3.

Again, in Figure 3, R1 may send (S,G) join to R3 and R2 may send (*,G) join to the RP behind R5. In this scenario as well, both R1 and R2 will receive duplicate traffic, as Guideline 25 will be no help to prevent it.

In this case, where R1 joins for (S,G), and R2 joins for (*,G), we can do the following. The PEs SHOULD keep the upstream PW/AC in the state as described above. In addition, the PEs need to act on (S,G,RPT) prunes and remove the related upstream PW/AC from "Flood to" list of (S,G) state copied from (*,G) state. As a result, Ces will not receive duplicate traffic.

However, there will still be bandwidth waste as the egress PE takes care of the duplicate traffic problem. We can further enhance the proposal by triggering Assert mechanism in CE routers. The PE which detects the duplicate traffic problem can simply remove the snooping state for that particular multicast group, and can send out "flush" message to other PEs participating in the VPLS instance. In return, other PEs also flush their snooping state for that multicast group. As a result, all the PEs will flood the multicast traffic in the VPLS instance (by Rule 3). Consequently, CEs will do Assert. The flush message TLV can be sent over the targeted LDP sessions running among PEs. Future versions will include the details.

[5.4.4](#). PIM-SSM

The key characteristics of PIM-SSM is explicit join behavior, but it eliminates the shared tree and the rendezvous point in PIM-SM. In this model, a shortest path tree for each (S,G) is built with the first hop router (that is directly connected to the multicast source) being the root node. PIM-SSM is ideal for one-to-many multicast services.

In Figure 2, S1 is behind Router 1, and S4 is behind Router 4. Routers 2 and 4 want to join (S1,G), while Router 5 wants to join (S4,G).

We assume that the PEs have the capability to store (S,G) states for PIM-SSM snooping and constrain multicast flooding scope accordingly.

An implementation, can fall back to (*,G) states, if its hardware can
not support it. In such case, the efficiency of multicast forwarding
will be less.

5.4.4.1. Discovering Multicast Routers

  The PIM-SSM snooping mechanism for neighbor discovery works the same
  way as the procedure defined in PIM-DM section, with the exception of
  PIM-DM only guidelines.

    -    Based on PIM Hello exchanges PE routers populate PIM snooping
    states as follows. PE 1: {[(,); Flood to:], [PIM Neighbors:
    (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3),
    (Router 5,PW1to4)]}, PE 2: {[(,); Flood to:], [PIM Neighbors:
    (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
    4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[(,); Flood to:], [PIM
    Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router
    4,AC4), (Router 5,PW3to4)]}, PE 4: {[(,); Flood to:], [PIM
    Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router
    4,PW3to4), (Router 5,AC5)]}.

5.4.4.2. Guidelines for PIM-SSM Snooping
  PIM-SSM snooping is actually simpler than PIM-SM and only the
  following guidelines (some of which are repetitions from PIM-SM
  section) apply.

    Guideline 28: A PE MUST add a PW/AC to its (S,G) pim_joins(S,G)
    list if it receives a (S,G) join message from the PW/AC.

    Guideline 29: PIM-SSM join messages MUST be sent only to the
    remote PE, which is connected to the router to which the Join is
    addressed.
    JP_Optimization: The PE MUST forward the Join message towards the
    upstream neighbor only if the pim_joins(S,G) list becomes non-
    empty as a result of the received join. If the pim_joins(S,G) list
    was non-empty prior to receiving the join message, then the PE
    MUST NOT forward the join message.

    Guideline 30: PIM prune messages MUST be flooded in the VPLS
    instance. A prune-delay timer SHOULD be implemented to support
    prune override on the downstream AC/PW. However, the prune-delay
    timer is not required if there is only one PIM neighbor on that
    AC/PW on which the prune was received.
    JP_Optimization: Instead of flooding the prune message in the VPLS
    instance, the PE MUST forward the prune message towards the
    upstream neighbor only if the pim_joins(S,G) list becomes empty as
    a result of the received prune. If the pim_joins(S,G) list remains
    non-empty after receiving the prune message, then the PE MUST NOT
    forward the prune message.

Guideline 31: If A PE does not receive a refresh join message from
a PW/AC within its Holdtime, the PE MUST remove the PW/AC from its
pim_joins(S,G) list.

Guideline 32: A PE MUST remove a PW/AC from its pim_joins(S,G)
list if it receives a (S,G) prune message from the PW/AC. A
prune-delay timer SHOULD be implemented to support prune override.

[5.4.4.3](5.4.4.3). PIM-SSM Join

The PIM-SSM snooping mechanism for joining a multicast group works as
follows:

-    Assume Router 2 requests to join the multicast group (S1,G).
-    PE 2 updates its state, and then sends the join message to PE
1.
-    All PEs update their states as follows: PE 1: {[(S1,G); Flood
to: PW1to2], [PIM Neighbors: (Router 1,AC1), (Router 2,Router
3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2:
{[pim_joins(S1,G): AC2], [PIM Neighbors: (Router 1,PW1to2),
(Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router
5,PW2to4)]}, PE 3: {[(,); Flood to:], [PIM Neighbors: (Router
1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4), (Router
5,PW3to4)]}, PE 4: {[(,); Flood to:], [PIM Neighbors: (Router
1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router
5,AC5)]}.
-    Next, assume Router 4 sends a join (S1,G) message. Following
the same procedures, all PEs update their states as follows: PE 1:
{[pim_joins(S1,G): PW1to2, PW1to3], [PIM Neighbors: (Router
1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router
5,PW1to4)]}, PE 2: {[pim_joins(S1,G): AC2], [PIM Neighbors:
(Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[pim_joins(S1,G): AC4],
[PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3),
(Router 4,AC4), (Router 5,PW3to4)]}, PE 4: {[(,); Flood to:],
[PIM Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4),
(Router 4,PW3to4), (Router 5,AC5)]}.
-    Then, assume Router 5 requests to join the multicast group
(S4,G). After the same procedures are applied, all PEs update
their states as follows: PE 1: {[pim_joins(S1,G): PW1to2, PW1to3],
[PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2),
(Router 4,PW1to3), (Router 5,PW1to4)]}, PE 2: {[pim_joins(S1,G):
AC2], [PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router
3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)]}, PE 3:
{[pim_joins(S1,G): AC4], [pim_joins(S4,G): PW3to4], [PIM
Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router

4,AC4), (Router 5,PW3to4)]}, PE 4: {[pim_joins(S4,G): AC5], [PIM
Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router
4,PW3to4), (Router 5,AC5)]}.

[5.4.4.4](#). PIM-SSM Prune

At this point, all PEs have necessary states to not send multicast
traffic to sites with no members.

The PIM-SSM snooping mechanism for leaving a multicast group works as
follows:

Assume Router 2 sends a (S1,G) prune message to leave the multicast
group. The prune message gets flooded in the VPLS instance. All PEs
update their states as follows: PE 1: {[pim_joins(S1,G): PW1to3],
[PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router
4,PW1to3), (Router 5,PW1to4)]}, PE 2: {[Deletes (S1,G) state], [PIM
Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router
4,PW2to3), (Router 5,PW2to4)]}, PE 3: {[(S1,G); Flood to: AC4],
[(S4,G); Flood to: PW3to4], [PIM Neighbors: (Router 1,PW1to3),
(Router 2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)]}, PE
4: {[(S4,G); Flood to: AC5], [PIM Neighbors: (Router 1,PW1to4),
(Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router 5,AC5)]}.

In PIM-SSM snooping, prune messages are flooded by PE routers. In
such implementation, PE routers may receive overriding join messages,
which will not affect anything.

[5.4.4.5](#). Failure Scenarios

Similar to PIM-SSM snooping, failures can be easily handled in PIM-
SSM snooping, as it employs state-refresh technique. The PEs in the
VPLS instance will remove entry for non-refreshing routers from their
states.

[5.4.4.6](#). Special Cases for PIM-SSM Snooping

The scenarios with duplicate traffic as depicted in Figure 3 apply to
PIM-SSM snooping as well. Again, the issue can be solved by the
method described in [Section 5.4.3.8](#). .

[5.4.5](#). Bidirectional-PIM (BIDIR-PIM)

BIDIR-PIM is a variation of PIM-SM. The main differences between
PIM-SM and Bidirectional-PIM are as follows:
    -    There are no source-based trees, and source-specific
    multicast is not supported (i.e., no (S,G) states) in BIDIR-PIM.
    -    Multicast traffic can flow up the shared tree in BIDIR-PIM.
    -    To avoid forwarding loops, one router on each link is elected

as the Designated Forwarder (DF) for each RP in BIDIR-PIM.

The main advantage of BIDIR-PIM is that it scales well for many-to-
many applications. However, the lack of source-based trees means
that multicast traffic is forced to remain on the shared tree.

In Figure 2, the RP for (*,G4) is behind Router 4, and the RP for
(*,G1) is behind Router 1. Router 2 and Router 4 want to join
(*,G1), whereas Router 5 wants to join (*,G4). On the VPLS instance,
Router 4 is the DF for the RP of (*,G4), and Router 1 is the DF of
the RP for (*,G1).

5.4.5.1. Discovering Multicast Routers

The PIM-SSM snooping mechanism for neighbor discovery works the same
way as the procedure defined in PIM-DM section, with the exception of
PIM-DM only guidelines.
- Based on PIM Hello exchanges PE routers populate PIM snooping
  states as follows. PE 1: { [PIM Neighbors: (Router 1,AC1),
  (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)]
  }, PE 2: { [PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2),
  (Router 3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)] }, PE 3:
  {[PIM Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3),
  (Router 4,AC4), (Router 5,PW3to4)]}, PE 4: { [PIM Neighbors:
  (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4),
  (Router 5,AC5)]}.

For BIDIR-PIM to work properly, all routers within the domain must
know the address of the RP. There are three methods to do that: 1.
Static RP configuration, 2, Auto-RP, and 3. PIMv2 Bootstrap.
Guideline 17 applies here as well.

During RP discovery time, PIM routers elect DF per subnet for each
RP. The algorithm to elect the DF is as follows: all PIM neighbors
in a subnet advertise their unicast route to elect the RP and the
router with the best route is elected.

   Guideline 33: All PEs MUST snoop the DF elections messages and
   determine the DF for AC/PW towards the DF (DF(RP)) MUST be added
   to the oiflist for each (*,G) whose RP(G) is RP. When DF(RP)
   changes., the oiflist must be updated accordingly, the oiflist
   must be updated accordingly

   - In Figure 2, there is one RP (call it RPA) behind Router 5.
   Based on DF election messages, PE routers populate PIM snooping
   states as follows: PE 1: {[PIM Neighbors: (Router 1,AC1), (Router
   2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)],
   [DF(RPA): PW1to3], PE 2: {[PIM Neighbors: (Router 1,PW1to2),
   (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3), (Router
   5,PW2to4)], [DF(RPA): PW2to3]}, PE 3: {[PIM Neighbors: (Router

1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4), (Router
5,PW3to4)], [DF(RPA): AC5]}, PE 4: {[PIM Neighbors: (Router
1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router
5,AC5)], [DF(RPA): PW3to4]}.

[5.4.5.2](5.4.5.2). Guidelines for BIDIR-PIM Snooping

The BIDIR-PIM snooping for Join and Prune messages is similar to
PIM-SM and the following guidelines (some of which are repetitions
from PIM-SM section) apply.

Guideline 34: A PE MUST add a PW/AC to its pim_joins(*,G) list if
it receives a (*,G) join message from the PW/AC.

Guideline 35: BIDIR-PIM join messages MUST be flooded to all PEs
in the VPLS instance. BIDIR-PIM join messages received on remote
PEs MUST be forwarded only towards the router to which the Join is
addressed.

Guideline 36: BIDIR-PIM prune messages MUST be flooded in the VPLS
instance.

Guideline 37: If A PE does not receive a refresh join message from
a PW/AC within its Holdtime, the PE MUST remove the PW/AC from its
pim_joins(*,G) list.

Guideline 38: A PE MUST remove a PW/AC from its pim_joins(*,G)
list if it receives a (*,G) prune message from the PW/AC. A
prune-delay timer SHOULD be implemented to support prune override.

[5.4.5.3](5.4.5.3). BIDIR-PIM Join

The BIDIR-PIM snooping mechanism for joining a multicast group works
as follows:
-     As before, assume the RP for both G1 and G4 (RPA) is behind
Router 4. Assume Router 2 wants to join the multicast group
(*,G1). PE 2 sends the join message to the other PEs. All PEs
update their states as follows: PE 1: { [PIM Neighbors: (Router
1,AC1), (Router 2,Router 3,PW1to2), (Router 4,PW1to3), (Router
5,PW1to4)], [DF(RPA): PW1to3], [pim_joins(*,G1): PW1to2]}, PE 2:
{[PIM Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router
3,AC3), (Router 4,PW2to3), (Router 5,PW2to4)], [DF(RPA): PW2to3],
[pim_joins(*,G1): AC2]}, PE 3: {[PIM Neighbors: (Router 1,PW1to3),
(Router 2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)],
[DF(RPA): AC4], [pim_joins(*,G1): PW2to3]}, PE 4: { [PIM
Neighbors: (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router

      4,PW3to4), (Router 5,AC5)], [DF(RPA): PW3to4], [pim_joins(*,G1):
      PW2to4]}.
      -     Next, assume Router 4 wants to join the multicast group
      (*,G1). All PEs update their states as follows: PE 1: {[PIM
      Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2), (Router
      4,PW1to3), (Router 5,PW1to4)], [DF(RPA): PW1to3],
      [pim_joins(*,G1): PW1to2, PW1to3]}, PE 2: {[PIM Neighbors: (Router
      1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3),
      (Router 5,PW2to4)], [DF(RPA): PW2to3], [pim_joins(*,G1): AC2,
      PW2to3], PE 3: {[PIM Neighbors: (Router 1,PW1to3), (Router
      2,Router 3,PW2to3), (Router 4,AC4), (Router 5,PW3to4)], [DF(RPA):
      AC4], pim_joins(*,G1): PW2to3, AC4]}, PE 4: {[PIM Neighbors:
      (Router 1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4),
      (Router 5,AC5)], [DF(RPA): PW3to4], [pim_joins(*,G1): PW2to4,
      PW3to4]}.
      -     Then, assume Router 5 wants to join the multicast group
      (*,G4). Following the same procedures, all PEs update their states
      as follows: PE 1: {[PIM Neighbors: (Router 1,AC1), (Router
      2,Router 3,PW1to2), (Router 4,PW1to3), (Router 5,PW1to4)],
      [DF(RPA): PW1to3], [pim_joins(*,G1): PW1to2, PW1to3],
      [pim_joins(*,G4): PW1to4]}, PE 2: {[PIM Neighbors: (Router
      1,PW1to2), (Router 2,AC2), (Router 3,AC3), (Router 4,PW2to3),
      (Router 5,PW2to4)], [DF(RPA): PW2to3], [pim_joins(*,G1): AC2,
      PW2to3], [pim_joins(*,G4): PW2to4]}, PE 3: {[PIM Neighbors:
      (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router 4,AC4),
      (Router 5,PW3to4)], [DF(RPA): AC4], pim_joins(*,G1): PW2to3, AC4],
      pim_joins(*,G4): PW3to4]}, PE 4: {[PIM Neighbors: (Router
      1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router
      5,AC5)], [DF(RPA): PW3to4], [pim_joins(*,G1): PW2to4, PW3to4],
      [pim_joins(*,G4): AC5]}.

5.4.5.4. BIDIR-PIM Prune

  At this point, all PEs have necessary states to not send multicast
  traffic to sites with no members.

  One example of the BIDIR-PIM snooping mechanism for leaving a
  multicast group works as follows:
      -     Assume Router 2 wants to leave the multicast group (*,G1) and
      sends a (*,G1) prune message. The prune message gets flooded in
      the VPLS instance. All PEs update their states as follows: PE 1:
      {[PIM Neighbors: (Router 1,AC1), (Router 2,Router 3,PW1to2),
      (Router 4,PW1to3), (Router 5,PW1to4)], [DF(RPA): PW1to3],
      [pim_joins(*,G1): PW1to3], [pim_joins(*,G4): PW1to4]}, PE 2: {[PIM
      Neighbors: (Router 1,PW1to2), (Router 2,AC2), (Router 3,AC3),
      (Router 4,PW2to3), (Router 5,PW2to4)], [DF(RPA): PW2to3],

    [pim_joins(*,G1): PW2to3], [pim_joins(*,G4): PW2to4]}, PE 3: {[PIM
    Neighbors: (Router 1,PW1to3), (Router 2,Router 3,PW2to3), (Router
    4,AC4), (Router 5,PW3to4)], [DF(RPA): AC4], [pim_joins(*,G1):
    AC1], [pim_joins(*,G4): PW3to4]}, PE 4: {[PIM Neighbors: (Router
    1,PW1to4), (Router 2,Router 3,PW2to4), (Router 4,PW3to4), (Router
    5,AC5)], [DF(RPA): PW3to4], [pim_joins(*,G1): PW3to4],
    [pim_joins(*,G4): AC5]}.

5.4.5.5. Failure Scenarios

  Once again, failures can be easily handled in BIDIR-PIM snooping, as
  it employs state-refresh technique. PEs in the VPLS instance will
  remove entry for non-refreshing routers from their states.

5.4.6. Multicast Source Directly Connected to the VPLS Instance

  If there is a source in the CE network that connects directly into
  the VPLS instance, then multicast traffic from that source MUST be
  sent to all PIM routers on the VPLS instance apart from the outgoing
  interface list for the corresponding snooping state. If there is
  already (S,G)/(*,G) snooping state that is formed on any PE, this
  will not happen per the current forwarding rules and guidelines. The
  (S,G)/(*,G) state may not send traffic towards all the routers. So,
  in order to determine if traffic needs to be flooded to all routers,
  a PE must be able to determine if the traffic came from a host on
  that LAN. There are three ways to address this problem:
    -    The PE would have to do ARP snooping to determine if a source
    is directly connected.
    -    Another option is to have configuration on all PEs to say
    there are CE sources that are directly connected to the VPLS
    instance and disallow snooping for the groups for which the source
    is going to send traffic. This way traffic from that source to
    those groups will always be flooded within the provider network.
    -    A third option is to require that sources of CE multicast
    routers must appear behind a router.

5.5. VPLS Multicast on the Upstream PE

  An implementation MAY use native PIM Snooping procedures on the
  upstream PE to build multicast state as described in the previous
  sections. But snooping on PWs may overwhelm the upstream PEs. In this
  section, we propose an alternate approach for building multicast
  states on the upstream PE and thus avoid snooping on the PWs.

  As discussed earlier in the previous sections, VPLS Multicast for the
  various flavors of PIM requires only the downstream PE(s) and the
  upstream PE to build multicast state for a given multicast flow.
  Join/Prune messages need only be sent towards the upstream CE and it

is wasteful to distribute and/or build multicast state on all PEs.

Unless otherwise noted, an upstream PE in this section refers to the PE to which the upstream RPF neighbor in the C-instance is connected.

[VPLS-LDP] already uses LDP as the infrastructure to build PWs between the PEs and to exchange VPLS information. It suits VPLS multicast very well to leverage this existing infrastructure to send PIM multicast state information from the downstream PE to the upstream PE. While the procedures described here are extensible to other IP multicast protocols as well, we will define the procedures for PIM in this draft. The rules and procedures for this are described below.

5.5.1. Negotiating PIM Multicast capability in LDP

When a PE is capable of exchanging PIM Multicast states using LDP, it signals this capability to its peers. When the two ends of a PW are both capable of exchanging PIM control information using LDP, the procedures described in the following sub-sections are employed. Otherwise, the following procedures are simply skipped.

If a LDP-multicast capable PE determines that the other end of a PW is LDP-multicast capable, it SHOULD turn off native PIM snooping procedures on that PW. Otherwise, it MAY employ PIM snooping procedures to build multicast states.

Packet format for exchanging PIM Multicast capability information in LDP will be defined in a future revision of the draft. One option is to define a bit in the LDP Hello Message to signal this capability.

5.5.2. Exchanging PIM Hellos

We introduce a new PIM Hello TLV to carry the PIM Hellos received at the downstream PEs to all the other PEs using LDP. This TLV is sent in the LDP Address Message to all other PEs. The scope of this TLV is the VPLS instance specified in the FEC TLV in the LDP Address Message.

When a new PIM Hello is received at the downstream PE, it is sent to all other PEs using the PIM Hello TLV. Subsequently, if a PIM Hello received from a C-router is the same as the previous Hello received from that C-router, it SHOULD NOT be sent in the PIM Hello TLV. If a PIM Hello received from a C-router is different from the previous received Hello from that C-router, the PE MUST send it in the PIM Hello TLV to all PEs. If the downstream PE ages out a PIM Hello, it MUST send a PIM Hello TLV with zero hold time to remove the Hello state on all other PEs.

The downstream PE MUST also forward the PIM Hello on all PWs.

When a PE receives a PIM Hello TLV for a C-router, it replaces the
old PIM Hello information with the new one received in the TLV. The
upstream PE never ages out a PIM Hello state. It MUST remove a PIM

Hello state only when it receives a PIM Hello TLV with zero hold-time
OR when the PW is torn down.

When a PE receives a new PIM Hello TLV, then all multicast states
with that PW as the RPF interface MUST be refreshed with PIM
Join/Prune TLVs.

Packet format for the PIM Hello TLV will be defined in a future
revision of this draft.

[5.5.3](5.5.3). Exchanging PIM Join/Prune States

We introduce a new PIM Join/Prune TLV to advertise C-Join/Prune
messages received at a downstream PE to the upstream PE. This TLV is
sent in the LDP Address Message to the upstream PE. The scope of this
TLV is the VPLS instance specified in the FEC TLV in the LDP Address
Message.

When a downstream PE receives a new C-Join/Prune message for a
multicast state, it MUST send the C-Join/Prune message in a PIM
Join/Prune TLV to the upstream PE. If it receives a C-Join/Prune
message information different than what was received before (e.g.
newer hold-time, change in Join/Prune state, different RPF-neighbor
field, etc), the PE MUST send the Join/Prune message in a PIM
Join/Prune TLV to the new upstream PE. Note that if a C-Join is
received for a new upstream PE, it may not imply that the state on
the old upstream PE needs to be torn down. Different C-Routers may be
sending C-Joins to different upstream RPF neighbors.

If the downstream PE cannot determine who the upstream PE is, then it
SHOULD save the Join/Prune state and send the PIM Join/Prune TLV when
it is able to determine who the upstream PE is i.e. when it receives
a PIM Hello TLV on a PW from the corresponding C-RPF-neighbor.

The downstream PE MUST also forward the C-Join/Prune message. These
packets will not be snooped at the upstream PE(s) and are intended
only for the upstream C-router(s).

Packet format for the PIM Join/Prune TLV will be defined in a future
revision of this draft.

[5.5.3.1](5.5.3.1). PIM Join Suppression Issues

For VPLS Multicast to work, the C-routers MUST disable PIM Join
suppression. However, it is our understanding that existing
deployments from several vendors do not support the capability to
disable PIM Join suppression. If that is so, then VPLS Multicast
simply does not work if we multicast the C-Joins to all C-routers.

Also, the provider has no control over the configuration on a C-router (to ensure that C-Join Suppression is disabled).

If the downstream PE determines that PIM Join suppression is active
in a VPLS instance, then it MUST unicast-forward the C-Joins towards
the RPF-neighbor field in the C-Join. This allows the C-Join to not
be seen by other C-routers. Since we recommend that it unicast-
forward the C-Join/Prune packets, it is important to ensure that the
PIM control packets are received in order at the upstream C-router.
To achieve this, the same ordering restriction that apply to
broadcast and unknown frames apply to PIM control packets.

5.5.3.2. Resiliency against soft-state failures

PIM is a soft-state protocol. So it is possible for packets to get
dropped. Even though the Join/Prune exchange between the PEs is
reliable; if certain packets are not received at the downstream PE,
it can leave stale state on the upstream PE. Specifically, an RPF
change on a downstream C-Router results in a C-Prune message to be
sent to the old RPF-neighbor and a C-Join to be sent to the new RPF-
neighbor. If the C-Prune message is not received at the downstream
PE, then the downstream PE will not be able to forward that message
to the upstream PE. This will result in stale state in the upstream
PE.

An implementation MUST implement one of the following two procedures
to handle this.

5.5.3.2.1. Explicit Tracking of C-Joins at the downstream PE

This method allows us to not require refreshes on PWs and yet achieve
resiliency from soft-state failures. If this method is used, then the
hold-time encoded in the PIM Join/Prune TLVs MUST be set to infinity.
This is the recommended method since it eliminates refreshes on PWs.

For each C-Join(S,G) received at the downstream PE on an AC, the
downstream PE MUST keep the following state per Upstream C-Router:

  - The Set of Upstream C-Router Addresses
      o Per Upstream C-Router, a set of:
            . Downstream C-Router Address
            . Downstream Join Expiry Timer

When a C-Join received at the downstream PE results in the set of
Downstream C-Routers for a (C-Source, C-Group, C-RPF-Neighbor) to
become non-empty, then a PIM Join TLV MUST be sent to the Upstream
PE.

If the set of Downstream C-Routers for a (C-Source, C-Group, C-RPF-
Neighbor) becomes empty, then a PIM Prune TLV MUST be sent to the
corresponding Upstream PE. This may happen as a result of a received

C-Prune message or as a result of the Downstream Join Timer Expiry.

[5.5.3.2.2](#). **Refreshing PIM Join TLVs on the PWs**

   If this method is employed, the downstream PE MUST forward received
   C-Joins in the form of PIM Join TLVs on the PWs at periodic
   intervals. The refresh interval across the PWs should be configurable
   in multiples of the C-Join refresh interval. If this refresh multiple
   is N, then every Nth C-Join refresh for a given multicast state MUST
   also be sent as a PIM Join TLV to the upstream PE. The HoldTime field
   in the PIM Join TLV MUST be set to ((N * HoldTime in C-Join) + 20)
   seconds.

[5.5.3.3](#). **PIM-BIDIR Considerations**

   Unlike other PIM variants, in PIM-BIDIR, a traffic source need not be
   behind the RPF-neighbor. Traffic can come from any AC/PW and it MUST
   be forwarded by the switches. Following are the deviations from the
   procedures defined earlier to handle PIM-BIDIR.

   PIM-BIDIR Join/Prune TLVs MUST be forwarded to all PEs instead of
   just the upstream PE towards the RP. PIM BIDIR Join/Prune Packets
   MUST also be multicast-forwarded as is on all PWs.

[5.6](#). **Data Forwarding Rules**

   The final list of outgoing interfaces for a given (S,G) or (*,G) is
   computed by combining the IGMP and PIM state summarization macros.

   OifList(*,G) = igmp_include(*,G) (+) pim_oiflist(*,G)

   Oiflist(S,G) = igmp_include(*,G) (-) igmp_exclude(S,G) (+)
                  Igmp_include(S,G) (+) pim_oiflist(S,G)

   If PIM Snooping is active for a given (*,G) or (S,G), then the PE
   also tracks the upstream AC/PW as the RPF interface. Data traffic
   MUST be forwarded ONLY IF traffic arrives on the RPF interface. If
   data traffic arrives on any other interface, then the following rules
   apply:
       -    If the traffic arrives on an AC and the PE determines that
       the traffic is coming from a directly connected source, then the
       rules described in [Section 5.4.6](#). apply.
       -    Otherwise, it could be a PIM ASSERT scenario. Then the rules
       described in [Section 5.4.3.8](#). apply.

   In the presence of only IGMP Snooping state, there is no RPF
   interface that can be remembered. In such a scenario, traffic should
   simply be forwarded to the Oiflist after performing source interface
   pruning.

[6](#). **Security Considerations**

Security considerations provided in VPLS solution documents (i.e.,
[VPLS-LDP] and [VPLS-BGP) apply to this document as well.

7. References

7.1. Normative References

7.2. Informative References

[VPLS-LDP]	Lasserre, M, et al. "Virtual Private LAN Services over MPLS", work in progress
[VPLSD-BGP]	Kompella, K, et al. "Virtual Private LAN Service", work in progress
[L2VPN-FR]	Andersson, L, et al. "L2VPN Framework", work in progress
[PMP-RSVP-TE]	Aggarwal, R, et al. "Extensions to RSVP-TE for Point to Multipoint TE LSPs", work in progress
[RFC1112]	Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.
[RFC2236]	Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, November 1997.
[RFC3376]	Cain, B., et al. "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
[MAGMA-SNOOP]	Christensen, M., et al. "Considerations for IGMP and MLD Snooping Switches", work in progress
[PIM-DM]	Deering, S., et al. "Protocol Independent Multicast Version 2 Dense Mode Specification", draft-ietf-pim-v2-dm-03.txt, June 1999.
[RFC2362]	Estrin, D, et al. "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", RFC 2362, June 1998.
[PIM-SSM]	Holbrook, H., et al. "Source-Specific Multicast for IP", work in progress
[BIDIR-PIM]	Handley, M., et al. "Bi-directional Protocol Independent Multicast (BIDIR-PIM)", work in progress

Authors' Addresses

  Yetik Serbest
  SBC Labs
  9505 Arboretum Blvd.
  Austin, TX 78759
  Yetik_serbest@labs.sbc.com

  Ray Qiu
  Alcatel North America
  701 East Middlefield Rd.

Mountain View, CA 94043
Ray.Qiu@alcatel.com

Venu Hemige

   Alcatel North America
   701 East Middlefield Rd.
   Mountain View, CA 94043
   Venu.hemige@alcatel.com

   Rob Nath
   Riverstone Networks
   5200 Great America Parkway
   Santa Clara, CA 95054
   Rnath@riverstonenet.com

   Suresh Boddapati
   Alcatel North America
   701 East Middlefield Rd.
   Mountain View, CA 94043
   Suresh.boddapati@alcatel.com

   Sunil Khandekar
   Alcatel North America
   701 East Middlefield Rd.
   Mountain View, CA 94043
   Sunil.khandekar@alcatel.com

   Vach Kompella
   Alcatel North America
   701 East Middlefield Rd.
   Mountain View, CA 94043
   Vach.kompella@alcatel.com

   Marc Lasserre
   Riverstone Networks
   Marc@riverstonenet.com

   Himanshu Shah
   Ciena
   hshah@ciena.com

Intellectual Property Statement

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.

Full copyright statement