WHITE PAPER

UNDERSTANDING JUNOS NEXT-GENERATION MULTICAST VPNS

Table of Contents

Table of Figures

List of Tables:

Executive Summary

This white paper provides an overview of next-generation multicast VPNs (NG MVPNs) and describes how NG MVPN control and data plane protocols work together in Juniper Networks® JUNOS® Software. The target audience of this document is network architects, engineers, and operators. The paper consists of two main parts.

Overview of NG MVPNs—These sections include background material of how NG MVPNs work in general: concepts, terminology, control plane, and data plane.

JUNOS NG MVPNs—These sections detail how Juniper Networks routers operate and interact with each other to set up NG MVPN routing and forwarding state in the network.

Scope

The scope of this paper includes these JUNOS features.

- Intra-AS MVPN membership discovery via BGP MCAST-VPN address family
- BGP C-multicast route exchange when the PE-CE protocol is PIM-SM (SSM), PIM-SM (ASM), PIM-DM or IGMP (source-tree-only mode)
- IP/GRE based inclusive P-tunnels signaled by PIM-SM (ASM)
- MPLS inclusive P-tunnels signaled by RSVP-TE P2MP LSPs
- MPLS selective P-tunnels signaled by RSVP-TE P2MP LSPs

In addition to features identified in this paper, JUNOS Software also supports these features.

- NG MVPN applications: extranet with P2MP TE
- IP/GRE-based inclusive P-tunnels signaled by PIM-SM (SSM)
- NG MVPN applications: hub and spoke

It is assumed that you are familiar with the following drafts and RFCs.

- BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs (draft-ietf-l3vpn-2547bis-mcast-bgp)
- Multicast in MPLS/BGP IP VPNs (draft-ietf-l3vpn-2547bis-mcast)
- BGP/MPLS IP Virtual Private Networks (RFC 4364)
- Protocol Independent Multicast - Sparse Mode: Protocol Specification (RFC 4601)
- Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution (draft-ietf-l3vpn-mvpn-considerations)

Introduction

Layer 3 BGP-MPLS VPNs are widely deployed in today's networks worldwide. Multicast applications, such as IPTV, are rapidly gaining popularity as is the number of networks with multiple, media-rich services merging over a shared MPLS infrastructure. As such, the demand for delivering multicast service across a BGP-MPLS infrastructure in a scalable and reliable way is also increasing.

RFC 4364 describes protocols and procedures for building unicast BGP-MPLS VPNs. However, there is no framework specified in the RFC for provisioning multicast VPN (MVPN) services. Up to now, MVPN traffic has been overlaid on top of a BGP-MPLS network using a virtual LAN model based on Draft Rosen. Using the Draft Rosen approach, service providers were faced with control and data plane scaling issues of an overlay model and the maintenance of two routing/forwarding mechanisms: one for VPN unicast and one for VPN multicast service. For more information on the limitations of Draft Rosen, see draft-rekhter-mboned-mvpn-deploy.

As a result, the IETF Layer 3 VPN working group published an IETF draft (2547bis-mcast) that outlines the new architecture for NG MVPNs, as well as an accompanying draft (2547bis-mcast-bgp) that proposes a BGP control plane for MVPNs. In turn, Juniper Networks delivered the industry's first implementation of BGP NG MVPNs in 2007.

Example Network Topology

All examples in this document refer to the network in Figure 1.

- The service provider in this example offers VPN unicast and multicast service to Customer A (`vpna`).
- The VPN multicast source is connected to Site 1 and transmits data to groups `232.1.1.1` and `224.1.1.1`.
- VPN multicast receivers are connected to Site 2 and Site 3.
- The provider edge router 1 (PE1) VRF table acts as the C-RP (using address `10.12.53.1`) for C-PIM-SM ASM groups.
- The service provider uses RSVP-TE P2MP LSPs for transmitting VPN multicast data across the network.

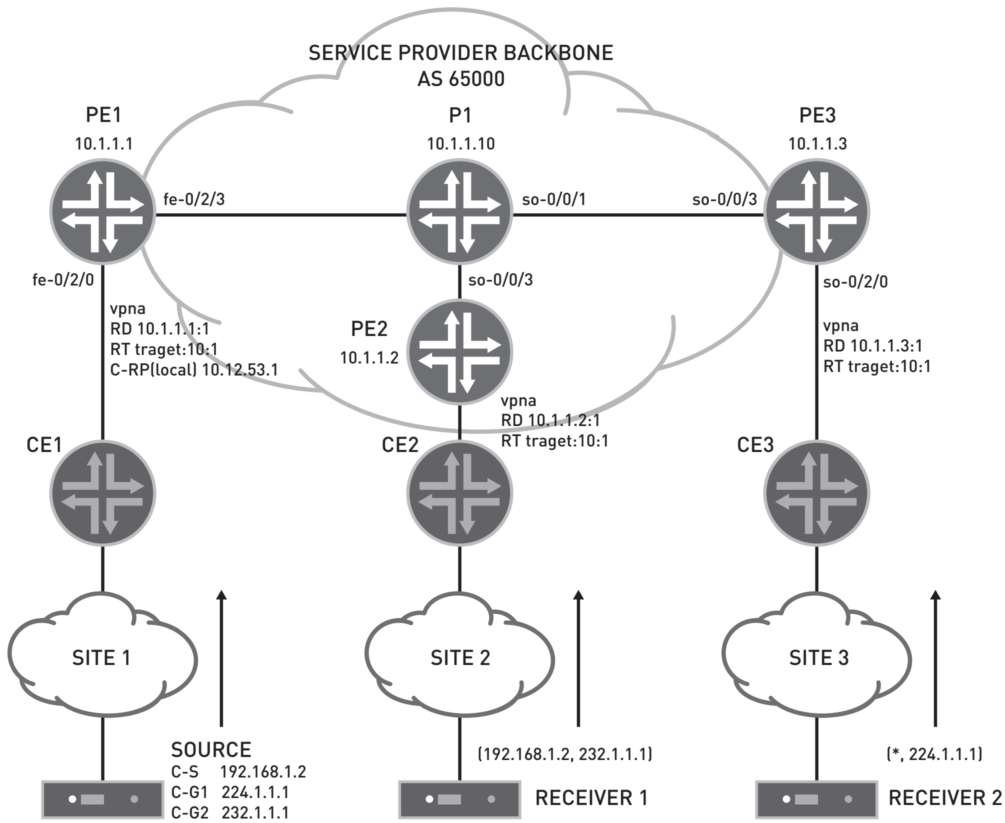


Figure 1: Example NG MVPN network

NG MVPN Concepts and Terminology

Route Distinguisher and VRF Route Target Extended Community

Route distinguisher (`RD`) and VRF route target (`RT`) extended communities are an integral part of unicast BGP-MPLS VPNs. `RD` and `RT` are often confused in terms of their purpose in BGP-MPLS networks. Because they play an important role in BGP NG MVPNs, it is important to understand what they are and how they are used as described in RFC 4364.

RFC 4364 describes the purpose of route distinguisher as the following.

> *"A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte Route Distinguisher (RD) and ending with a 4-byte IPv4 address. If several VPNs use the same IPv4 address prefix, the PEs translate these into unique VPN-IPv4 address prefixes. This ensures that if the same address is used in several different VPNs, it is possible for BGP to carry several completely different routes to that address, one for each VPN."*

Typically, each VRF table on a provider edge (PE) router is configured with a unique `RD`. Depending on the routing design, the `RD` can be unique or the same for a given VRF on other PE routers. `RD` is an 8-byte number with two fields. The first field can be either an AS number (2 or 4 bytes) or an IP address (4 bytes). The second field is assigned by the user. RFC 4364 describes the purpose of VRF route target extended community as the following.

> *"Every VRF is associated with one or more Route Target (RT) attributes.*
>
> *When a VPN-IPv4 route is created (from an IPv4 route that the PE has learned from a CE) by a PE router, it is associated with one or more route target attributes. These are carried in BGP as attributes of the route.*
>
> *Any route associated with Route Target T must be distributed to every PE router that has a VRF associated with Route Target T. When such a route is received by a PE router, it is eligible to be installed in those of the PE's VRFs that are associated with Route Target T."*

`RT` also contains two fields and is structured similar to `RD`; the first field of `RT` is either an AS number (2 or 4 bytes) or an IP address (4 bytes), and the second field is assigned by the user. Each PE router advertises its VPN-IPv4 routes with the `RT` (as one of the BGP path attributes) configured for the VRF table. The `RT` attached to the advertised route is referred to as the *export RT*. On the receiving PE, the `RT` attached to the route is compared to the `RT` configured for the local VRF tables. The locally configured `RT` that is used in deciding whether a VPN-IPv4 route should be installed in a VRF table is referred to as the *import RT*.

C-Multicast Routing

Customer multicast (C-multicast) routing information exchange refers to the distribution of customer PIM (C-PIM) `join/prune` messages received from local customer edge (CE) routers to other PEs (towards the VPN multicast source).

BGP MVPNs

BGP MVPNs use BGP as the control plane protocol between PEs for MVPNs, including the exchange of C-multicast routing information. The support of BGP as a PE-PE protocol for exchanging C-multicast routes is mandated by draft-ietf-l3vpn-mvpn-considerations. The use of BGP for distributing C-multicast routing information is closely modeled after its highly successful counterpart of VPN unicast route distribution. Using BGP as the control plane protocol allows service providers to take advantage of this widely deployed, feature-rich protocol. It also enables service providers to leverage their knowledge and investment in managing BGP-MPLS VPN unicast service to offer VPN multicast services.

Sender and Receiver Site Sets

The 2547bis-mcast draft describes an MVPN as a set of administrative policies that determine the PEs that are in sender and receiver site sets. A PE router can be a sender, a receiver, or both a sender and a receiver, depending on the configuration.

- A sender site set includes PEs with local VPN multicast sources (VPN customer multicast sources either directly connected or connected via a CE router). A PE router that is in the sender site set is the *sender PE*.
- A receiver site set includes PEs that have local VPN multicast receivers. A PE that is in the receiver site set is the *receiver PE*.

P-tunnels

The 2547bis-mcast draft defines P-tunnels as the transport mechanisms used for forwarding VPN multicast traffic across service provider networks. Different tunneling technologies, such as GRE and MPLS, can be used to create P-tunnels. P-tunnels can be signaled by a variety of signaling protocols. This paper describes only PIM-SM (ASM) signaled IP/GRE P-tunnels and RSVP-TE signaled MPLS P-tunnels.

In BGP MVPNs, the sender PE distributes information about the P-tunnel in a new BGP attribute called `PMSI` (provider multicast service interface). By default, all receiver PEs join and become the leaves of the P-tunnel rooted at the sender PE.

P-tunnels can be inclusive or selective. An inclusive P-tunnel (I-PMSI P-tunnel) enables a PE router that is in the sender site set of an MVPN to transmit multicast data to all PE routers that are members of that MVPN. A selective P-tunnel (S-PMSI P-tunnel) enables a PE router that is in the sender site set of an MVPN to transmit multicast data to a subset of the PEs.

NG MVPN Control Plane

The BGP NG MVPN control plane, as specified in 2547bis-mcast and 2547bis-mcast-bgp, distributes all the necessary information to enable end-to-end C-multicast routing exchange via BGP. The main tasks of the control plane (Table 1) include MVPN autodiscovery, distribution of P-tunnel information, and PE-PE C-multicast route exchange.

Table 1: NG MVPN Control Plane Tasks

Control Plane Task	Description
MVPN autodiscovery	A PE router discovers the identity of the other PE routers that participate in the same MVPN.
Distribution of P-tunnel information	A sender PE router advertises the type and identifier of the P-tunnel that it will be using for transmitting VPN multicast packets.
PE-PE C-multicast route exchange	A receiver PE router propagates C-multicast `join` messages (C-joins) received over its VPN interface towards the VPN multicast sources.

BGP MCAST-VPN Address Family and Route Types

The 2547bis-mcast-bgp draft introduced a new BGP address family called *MCAST-VPN* for supporting NG MVPN control plane operations. The new address family is assigned the subsequent address family identifier (SAFI) of 5 by IANA.

A PE router that participates in a BGP-based NG MVPN network is required to send a BGP update message that contains an MCAST-VPN NLRI. An MCAST-VPN NLRI contains `route type`, `length`, and `variable` fields. The value of the `variable` field depends on the `route type`.

Seven types of NG MVPN BGP routes (also referred as *`mvpn` routes* in this document) are specified (Table 2). The first five route types are called *autodiscovery* (AD) *`mvpn` routes*. This paper also refers to Type 1-5 routes as *non-C-multicast `mvpn` routes*. Type 6 and Type 7 routes are called *C-multicast `mvpn` routes*.

Table 2: NG MVPN BGP Route Types

Used For ...	Type	Name	Description
Membership autodiscovery routes for inclusive P-tunnels	1	Intra-AS I-PMSI AD route	• Originated by all NG MVPN PE routers. • Used for advertising and learning intra-AS MVPN membership information.
	2	Inter-AS I-PMSI AD route	• Originated by NG MVPN ASBR routers. • Used for advertising and learning inter-AS MVPN membership information.
Autodiscovery routes for selective P-tunnels	3	S-PMSI AD route	• Originated by sender PEs. • Used for initiating a selective P-tunnel for a particular (C-S, C-G).
	4	Leaf AD route	• Originated by receiver PEs in response to receiving a Type 3 route. • Used by sender PE to discover the leaves of a selective P-tunnel. • Also used for inter-AS operations that are not covered in this paper.
VPN multicast source discovery routes	5	Source active AD route	• Originated by the PE router that discovers an active VPN multicast source. • Used by PEs to learn the identity of active VPN multicast sources.
C-multicast routes	6	Shared tree join route	• Originated by receiver PE routers. • Originated when a PE receives a shared tree C-join (C-*, C-G) through its PE-CE interface.
	7	Source tree join route	• Originated by receiver PE routers. • Originated when a PE receives a source tree C-join (C-S, C-G) or originated by the PE that already has a Type 6 route and receives a Type 5 route.

Intra-AS MVPN Membership Discovery (Type 1 Routes)

All NG MVPN PE routers create and advertise a Type 1 intra-AS AD route (Figure 2) for each MVPN to which they are connected.

```
+-----------------------------------+
| RD                                |
+-----------------------------------+
| Originating Router's IP Address   |
+-----------------------------------+
```

Figure 2: Intra-AS I-PMSI AD route type MCAST-VPN NLRI format

Field	Description
RD	Set to the `RD` configured for the VPN
Originating Router's IP Address	Set to the IP address of the router originating this route, which is typically the primary loopback address of the PE router.

Example: In Figure 1, PE1 originates the following intra-AS AD route.

```
1:10.1.1.1:1:10.1.1.1

Type   PE1 RD    PE1 lo0
```

- `1` is the route type, indicating that this is an intra-AS AD route
- `10.1.1.1:1` is the `RD` configured for `vpna` on PE1
- `10.1.1.1` is the loopback address of PE1

Similarly, in Figure 1, PE2 and PE3 originate the following intra-AS AD routes.

`1:10.1.1.2:1:10.1.1.2`

`1:10.1.1.3:1:10.1.1.3`

Inter-AS MVPN Membership Discovery (Type 2 Routes)

Type 2 routes are used for membership discovery between PE routers that belong to different ASs. Their use is not covered in this paper.

Selective P-Tunnels (Type 3 and Type 4 Routes)

A sender PE that initiates a selective P-tunnel is required to originate a Type 3 intra-AS S-PMSI AD route with the appropriate `PMSI` attribute.

A receiver PE router responds to a Type 3 route by originating a Type 4 leaf AD route if it has local receivers interested in the traffic transmitted on the selective P-tunnel. Type 4 routes inform the sender PE of the leaf PE routers.

Source Active AD Routes (Type 5 Routes)

Type 5 routes carry information about active VPN sources and the groups to which they are transmitting data. These routes can be generated by any PE router that becomes aware of an active source. Type 5 routes apply only for

PIM-SM (ASM) when intersite source–tree-only mode is being used.

C-multicast Route Exchange (Type 6 and Type 7 Routes)

The C-multicast route exchange between PE routers refers to the propagation of C-joins from receiver PEs to the sender PEs.

In an NG MVPN, C-joins are translated into (or encoded as) BGP C-multicast `mvpn` routes and advertised via BGP MCAST-VPN address family towards the sender PEs. Two types of C-multicast `mvpn` routes are specified.

- Type 6 C-multicast routes are used in representing information contained in a shared tree (C-*, C-G) join.
- Type 7 C-multicast routes are used in representing information contained in a source tree (C-S, C-G) join.

PMSI Attribute

The `PMSI` attribute (Figure 3) carries information about the P-tunnel. In an NG MVPN network, the sender PE router sets up the P-tunnel, and therefore is responsible for originating the `PMSI` attribute. The `PMSI` attribute can be attached to Type 1, Type 2, or Type 3 routes.

```
+-----------------------------------+
| Flags (1 octet)                   |
+-----------------------------------+
| Tunnel Type (1 octet)             |
+-----------------------------------+
| MPLS Label (3 octets)             |
+-----------------------------------+
| Tunnel Identifier (variable)      |
+-----------------------------------+
```

Figure 3: PMSI tunnel attribute format

Field	Description
Flags	Currently has only one flag specified: `Leaf Information Required`. This flag is used for S-PMSI P-tunnel setup.
Tunnel Type	Identifies the tunnel technology used by the sender. Currently there are seven types of tunnels supported.
MPLS Label	Used when the sender PE allocates the MPLS labels (also called *upstream label allocation*). This technique is described in RFC 5331 and is outside the scope of this paper.
Tunnel Identifier	Uniquely identifies the tunnel. Its value depends on the value set in the tunnel type field.

Example: In Figure 1, PE1 originates the following `PMSI` attribute.

```
PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:6574:10.1.1.1]
```

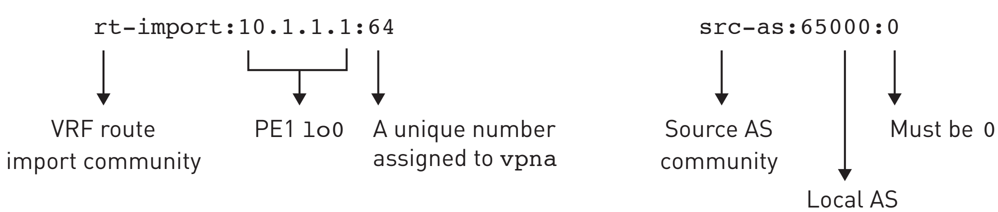
VRF Route Import and Source AS Extended Communities

Two new extended communities are specified to support NG MVPNs: source AS (`src-as`) and VRF route import (`rt-import`) extended communities.

The source AS extended community is an AS-specific extended community that identifies the AS from which a route originates. This community is mostly used for inter-AS operations, which is not covered in this paper.

The VRF route import extended community is an IP-address-specific extended community that is used for importing C-multicast routes in the active sender PE's VRF table to which the source is attached.

Each PE router creates a unique `rt-import` and `src-as` community for each VPN and attaches them to the VPN-IPv4 routes.



Example: In Figure 1, PE1 originates the following `rt-import` and `src-as` extended communities.

Similarly, in Figure 1, PE2 and PE3 originate the following `rt-import` and `src-as` extended communities.

```
rt-import:10.1.1.2:62 src-as:65000:0
rt-import:10.1.1.3:63 src-as:65000:0
```

Distribution of C-multicast Routes towards VPN Sources

While non-C-multicast `mvpn` routes (Type 1 – Type 5) are generally used by all PE routers in the network, C-multicast `mvpn` routes (Type 6 and Type 7) are only useful to the PE router connected to the active C-S or C-RP. Therefore, C-multicast routes need to be installed only in the VRF table on the active sender PE for a given C-G.

To accomplish this, 2547bis-mcast proposes to attach a special and dynamic `RT` to C-multicast `mvpn` routes (Figure 4).

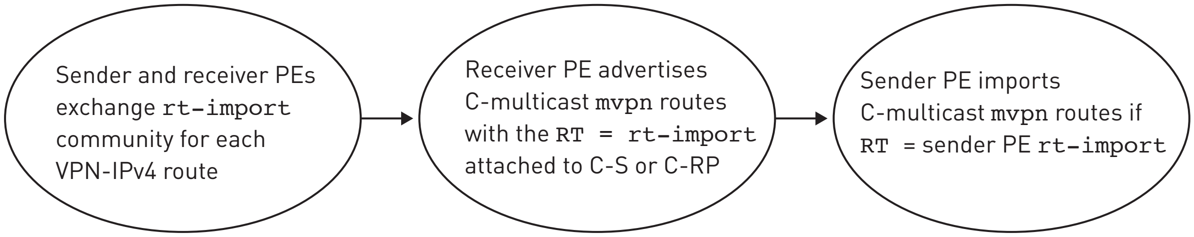


Figure 4: Attaching a special and dynamic RT to C-multicast mvpn routes

The `RT` attached to C-multicast routes is also referred to as *C-multicast import RT* and should not to be confused with `rt-import` (Table 3). Note that C-multicast `mvpn` routes differ from other `mvpn` routes in one essential way: they carry a dynamic `RT` whose value depends on the identity of the active sender PE at a given time and may change if the active PE changes.

Table 3: Distinction Between rt-import Attached to VPN-IPv4 Routes and RT Attached to C-multicast mvpn Routes

rt-import Attached to VPN-IPv4 Routes	RT Attached to C-multicast mvpn Routes
Value generated by the originating PE; must be unique per VRF table.	Value depends on the identity of the active PE.
Static. Created upon configuration to help identify to which PE and to which VPN the VPN unicast routes belong.	Dynamic because if the active sender PE changes, then the `RT` attached to the C-multicast routes must change to target the new sender PE. For example, a new VPN source attached to a different PE becomes active and preferred.

A PE router that receives a local C-join determines the identity of the active sender PE router by performing a unicast route lookup for the C-S or C-RP in the unicast VRF table. If there is more than one route, the receiver PE chooses a single forwarder PE. The procedures used for choosing a single forwarder are outlined in 2547bis-mcast-bgp and are not covered in this paper.

After the active sender (upstream) PE is selected, the receiver PE constructs the C-multicast `mvpn` route corresponding to the local C-join.

Once the C-multicast route is constructed, the receiver PE needs to attach the correct `RT` to this route targeting the active sender PE. As mentioned, each PE router creates a unique VRF route import (`rt-import`) community and attaches it to the VPN-IPv4 routes. When the receiver PE does a route lookup for C-S or C-RP, it can extract the value of the `rt-import` associated with this route and set the value of C-multicast import `RT` to the value of `rt-import` (Figure 5).

On the active sender PE, C-multicast routes are imported only if they carry an `RT` whose value is the same as the `rt-import` that the sender PE generated.

Exchange of `rt-import` community	
Step 1	PE1 creates a unique `rt-import` community for `vpna`. `rt-import:10.1.1.1:64`
Step 2	PE1 advertises local VPN routes to PE2 and PE3.
Step 3	PE1 attaches `rt-import` community to these routes.
Step 4	PE2 and PE3 install the VPN routes they learned from PE1 in their `vpna` unicast route tables.

Advertising C-multicast `mvpn` routes with the correct `RT`	
Step 1	PE2 receives a C-join. (`192.168.1.2, 232.1.1.1`)
Step 2	PE2 constructs a C-multicast `mvpn` route based on the C-join.
Step 3	PE2 finds the `rt-import` community attached to the C-S received in Step 1 (`192.168.1.2`). `rt-import:10.1.1.1:64`
Step 4	PE2 copies `rt-import` to the C-multicast `mvpn` route RT. `target:10.1.1.1:64`
Step 5	PE2 advertises C-multicast `mvpn` route to PE1 and PE3.

Importing C-multicast `mvpn` routes	

Step 1	PE1 compares the `RT` attached to the C-multicast mvpn routes to the `rt-import` it created.
	
	`RT` received: `target:10.1.1.1:64`
	`rt-import` created by PE1: `rt-import:10.1.1.1:64`

Step 2	If there is a match in Step 1, the C-multicast `mvpn` route is imported into the VRF table and translated back into a C-join message. It can now be processed as a normal C-join.
Step 3	The check in Step 1 happens on PE3 as well, but since PE3's `rt-import` (`10.1.1.3:63`) is different than the `RT` attached to the C-multicast mvpn route (`10.1.1.1:64`), PE3 discards the route.

Figure 5: Example of C-multicast mvpn route distribution

Constructing C-multicast Routes

A PE router originates a C-multicast `mvpn` route in response to receiving a C-join through its PE-CE interface. Refer to Figure 6 for the fields in the C-multicast route encoded in MCAST-VPN NLRI.

```
+-----------------------------------+
| RD (8 octets)                     |
+-----------------------------------+
| Source AS (4 octets)              |
+-----------------------------------+
| Multicast Source Length (1 octet) |
+-----------------------------------+
| Multicast Source (variable)       |
+-----------------------------------+
| Multicast Group Length (1 octet)  |
+-----------------------------------+
| Multicast Group (variable)        |
+-----------------------------------+
```

Figure 6: C-multicast route type MCAST-VPN NLRI format

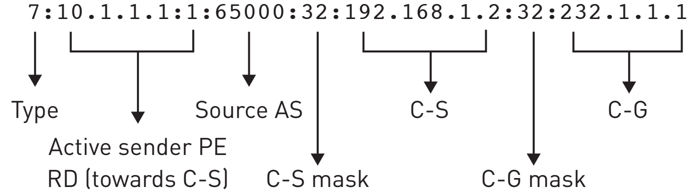
Field	Description
RD	Set to the `RD` of the C-S or C-RP (the `RD` associated with the upstream PE router).
Source AS	Set to the value found in the `src-as` community of the C-S or C-RP.
Multicast Source Length	Set to `32` for IPv4 and to `128` for IPv6 C-S or C-RP IP addresses.
Multicast Source	Set to the IP address of the C-S or C-RP.
Multicast Group Length	Set to `32` for IPv4 and to `128` for IPv6 C-G addresses.
Multicast Group	Set to the C-G of the received C-join.

This same structure is used for encoding both Type 6 and Type 7 routes with two differences.

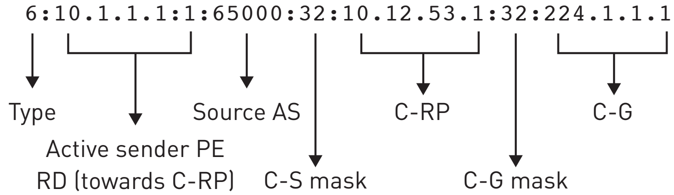
- The first difference is the value used for the multicast source field. For Type 6 routes, this field is set to the IP address of the C-RP configured. For Type 7 routes, this field is set to the IP address of the C-S contained in the (C-S, C-G) message.
- The second difference is the value used for `RD`. For Type 6 routes, this field is set to the `RD` that was attached to the IP address of the C-RP. For Type 7 routes, this field is set to the `RD` that was attached to IP address of the C-S.

Example: In Figure 1, PE2 creates the following Type 7 route in response to receiving (C-S, C-G) of (`192.168.1.2, 232.1.1.1`). C-S is reachable via PE1.

`7:10.1.1.1:1:65000:32:192.168.1.2:32:232.1.1.1`



Example: In Figure 1, PE3 creates the following Type 6 route in response to receiving (C-*, C-G) of (`*, 224.1.1.1`). C-RP is reachable via PE1.



Eliminating PE-PE Distribution of (C-*, C-G) State Using Source Active AD Routes

PE routers must maintain additional state when the C-multicast routing protocol is PIM-SM in ASM mode. This requirement is because with ASM, the receivers first join the shared tree rooted at C-RP (called *C-RP Tree* or *C-RPT*). However, as the VPN multicast sources become active, receivers learn the identity of the sources and join the tree rooted at the source (called *customer shortest-path tree* or *C-SPT*). The receivers then send a `prune` message to C-RP to stop the traffic coming through the shared tree for the group that they joined to C-SPT. The switch from C-RPT to C-SPT is a complicated process requiring additional state.

The 2547bis-mcast draft specifies optional procedures that completely eliminate the need for joining to C-RPT. These procedures require PE routers to keep track of all active VPN sources using one of two options. One option is to colocate C-RP on one of the PE routers. The second option is to use MSDP between one of the PEs and the customer C-RP.

In this approach, a PE router that receives a local (C-*, C-G) `join` creates a Type 6 route, but does not advertise the route to the remote PEs until it receives information about an active source. The PE router acting as the C-RP (or that learns about active sources via MSDP) is responsible for originating a Type 5 route. A Type 5 route carries information about the active source and the group addresses. The information contained in a Type 5 route is enough for receiver PEs to join C-SPT by originating a Type 7 route towards the sender PE, completely skipping the advertisement of Type 6 route that was created when a C-join was received.

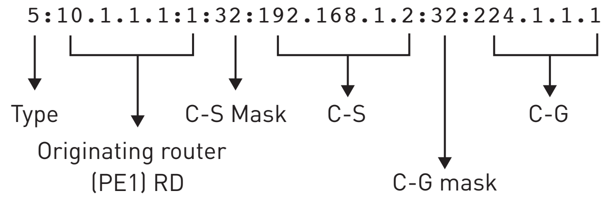
Figure 7 shows the format of source active (SA) AD route.

```
+-----------------------------------+
| RD (8 octets)                     |
+-----------------------------------+
| Multicast Source Length (1 octet) |
+-----------------------------------+
| Multicast Source (variable)       |
+-----------------------------------+
| Multicast Group Length (1 octet)  |
+-----------------------------------+
| Multicast Group (variable)        |
+-----------------------------------+
```

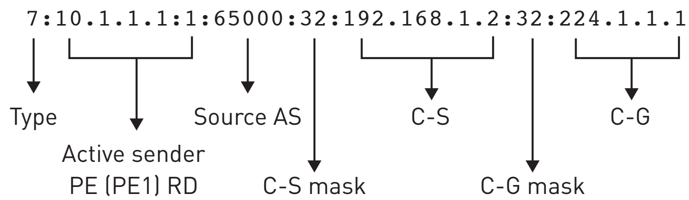
Figure 7: Source active AD route type MCAST-VPN NLRI format

Field	Description
RD	Set to the `RD` configured on the router originating the SA AD route.
Multicast Source Length	Set to `32` for IPv4 and to `128` for IPv6 C-S IP addresses.
Multicast Source	Set to the IP address of the C-S that is actively transmitting data to C-G.
Multicast Group Length	Set to `32` for IPv4 and to `128` for IPv6 C-G addresses.
Multicast Group	Set to the IP address of the C-G to which C-S is transmitting data.

Example: In Figure 1, PE1 originates the following Type 5 route in response to receiving `register` messages from CE1 (since it is the C-RP).



Example: In Figure 1, PE3 originates the following Type 7 route in response to receiving (`*, 224.1.1.1`) C-join and the Type 5 route.



Receiving C-multicast Routes

Sender PE imports C-multicast routes into the VRF table based on the route's `RT`. If the `RT` attached to the C-multicast `mvpn` route matches the `rt-import` community originated by this router, the C-multicast `mvpn` route is imported into the VRF table. If not, it is discarded.

Once the C-multicast `mvpn` routes are imported, they are translated back to C-joins and passed on to the VRF C-PIM protocol for further processing per normal PIM procedures.

NG MVPN Data Plane

An NG MVPN data plane is composed of P-tunnels originated by and rooted at the sender PE routers and the receiver PE routers as the leaves of the P-tunnel.

A P-tunnel can carry data for one or more VPNs. Those P-tunnels that carry data for more than one VPN are called *aggregate P-tunnels* and are outside the scope of this paper. Here, we assume that a P-tunnel carries data for one VPN only.

This paper covers two types of tunnel technologies: IP/GRE P-tunnels signaled by PIM-SM (ASM) and MPLS P-tunnels signaled by RSVP-TE.

When a P-tunnel is signaled by PIM, the sender PE router runs another instance of PIM protocol on the provider's network (P-PIM) that signals a P-tunnel for that VPN. When a P-tunnel is signaled by RSVP-TE, the sender PE router initiates a P2MP LSP towards receiver PEs by using P2MP RSVP-TE protocol messages. In either case, the sender PE advertises the tunnel signaling protocol and the tunnel ID to other PE routers via BGP by attaching the `PMSI` attribute to either the Type 1 intra-AS AD routes (inclusive P-tunnels) or Type 3 S-PMSI AD routes (selective P-tunnels).

Note that the sender PE goes through two steps when setting up the data plane. One, using the `PMSI` attribute, it advertises the P-tunnel it will be using via BGP. Two, it actually signals the tunnel using whatever tunnel signaling protocol is configured for that VPN. This allows receiver PE routers to bind the tunnel that is being signaled to the VPN that imported the Type 1 intra-AS AD route. Binding a P-tunnel to a VRF table enables a receiver PE router to map the incoming traffic from the core network on the P-tunnel to the local target VRF table.

The `PMSI` attribute contains P-tunnel type and an identifier. The value of the P-tunnel identifier depends on the tunnel type. Table 4 identifies the tunnel types specified in 2547bis-mcast-bgp (Table 4).

Table 4: Tunnel Types Supported by PMSI Tunnel Attribute

Tunnel Type	Description
0	No tunnel information present
1	RSVP-TE P2MP LSP
2	mLDP P2MP LSP
3	PIM-SSM tree
4	PIM-SM tree
5	PIM-Bidir tree
6	Ingress replication
7	mLDP MP2MP LSP

Inclusive P-tunnels

PMSI Attribute of Inclusive P-tunnels Signaled by PIM-SM

When the tunnel type field of the `PMSI` attribute is set to `4` (PIM-SM Tree), the tunnel identifier field contains `<Sender Address, P-Multicast Group Address>`. The sender address field is set to the `router-id` (`rid`) of the sender PE. The P-multicast group address is set to a multicast group address from the service provider's P-multicast address space and uniquely identifies the VPN. A receiver PE router that receives an intra-AS AD route with a `PMSI` attribute whose tunnel type is `PIM-SM` is required to join the P-tunnel.

Example: In Figure 1, if the service provider had deployed PIM-SM P-tunnels (instead of RSVP-TE P-tunnels) PE1 would have advertised the following `PMSI` attribute.

```
PMSI: 0:PIM-SM:label[0:0:0]:Sender10.1.1.1 Group 239.1.1.1
```

PMSI Attribute of Inclusive P-tunnels Signaled by RSVP-TE

When the tunnel type field of the PMSI attribute is set to 1 (RSVP-TE P2MP LSP), the tunnel identifier field contains RSVP-TE P2MP session object as described in RFC 4875. The session object contains the <Extended Tunnel ID, Reserved, Tunnel ID, P2MP ID> associated with the P2MP LSP.

The PE router that originates the `PMSI` attribute is required to signal an RSVP-TE P2MP LSP and the sub-LSPs. A PE router that receives this `PMSI` attribute must establish the appropriate state to properly handle the traffic received over the sub-LSP.

Example: In Figure 1, PE1 advertises the following `PMSI` attribute.

```
PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:6574:10.1.1.1]
```

Selective P-tunnels (S-PMSI AD/Type 3 and Leaf AD/Type 4 Routes)

A selective P-tunnel is used for mapping a specific C-multicast flow (a (C-S, C-G) pair) onto a specific P-tunnel. There are a variety of situations in which selective P-tunnels can be useful. For example, they can be used for putting high-bandwidth VPN multicast data traffic onto a separate P-tunnel than the default inclusive P-tunnel, thus restricting the distribution of traffic to only those PE routers with active receivers.

In BGP NG MVPNs, selective P-tunnels are signaled using Type 3 S-PMSI AD routes (Figure 8). The sender PE sends a Type 3 route to signal that it is sending traffic for a particular (C-S, C-G) flow using an S-PMSI P-tunnel.

```
+-----------------------------------+
| RD (8 octets)                     |
+-----------------------------------+
| Multicast Source Length (1 octet) |
+-----------------------------------+
| Multicast Source (variable)       |
+-----------------------------------+
| Multicast Group Length (1 octet)  |
+-----------------------------------+
| Multicast Group (variable)        |
+-----------------------------------+
```

Figure 8: S-PMSI AD route type MCAST-VPN NLRI format

Field	Description
RD	Set to the `RD` configured on the router originating this route.
Multicast Source Length	Set to `32` for IPv4 and to `128` for IPv6 C-S IP addresses.
Multicast Source	Set to the C-S IP address.
Multicast Group Length	Set to `32` for IPv4 and to `128` for IPv6 C-G addresses.
Multicast Group	Set to the C-G address.

The S-PMSI AD (Type 3) route carries a `PMSI` attribute similar to the `PMSI` attribute carried with intra-AS AD (Type 1) routes. The Flags field of the `PMSI` attribute carried by the S-PMSI AD route is set to `Leaf Information`

```
+-----------------------------------+
| Route Key (variable)              |
+-----------------------------------+
| Originating Router's IP Address   |
+-----------------------------------+
```

`Required`. This flag signals receiver PE routers to originate a Type 4 leaf AD route (Figure 9) to join the selective P-tunnel if they have active receivers.

Figure 9: Leaf AD route type MCAST-VPN NLRI format

Field	Description
Route Keu	Contains the original Type 3 route received.
Originating Router's IP Address	Set to the IP address of the PE originating the leaf AD route, typically the primary loopback address.

JUNOS NG MVPNs

Juniper introduced the industry's first implementation of BGP NG MVPNs. Refer to Figure 10 for a summary of a JUNOS NG MVPN routing flow.

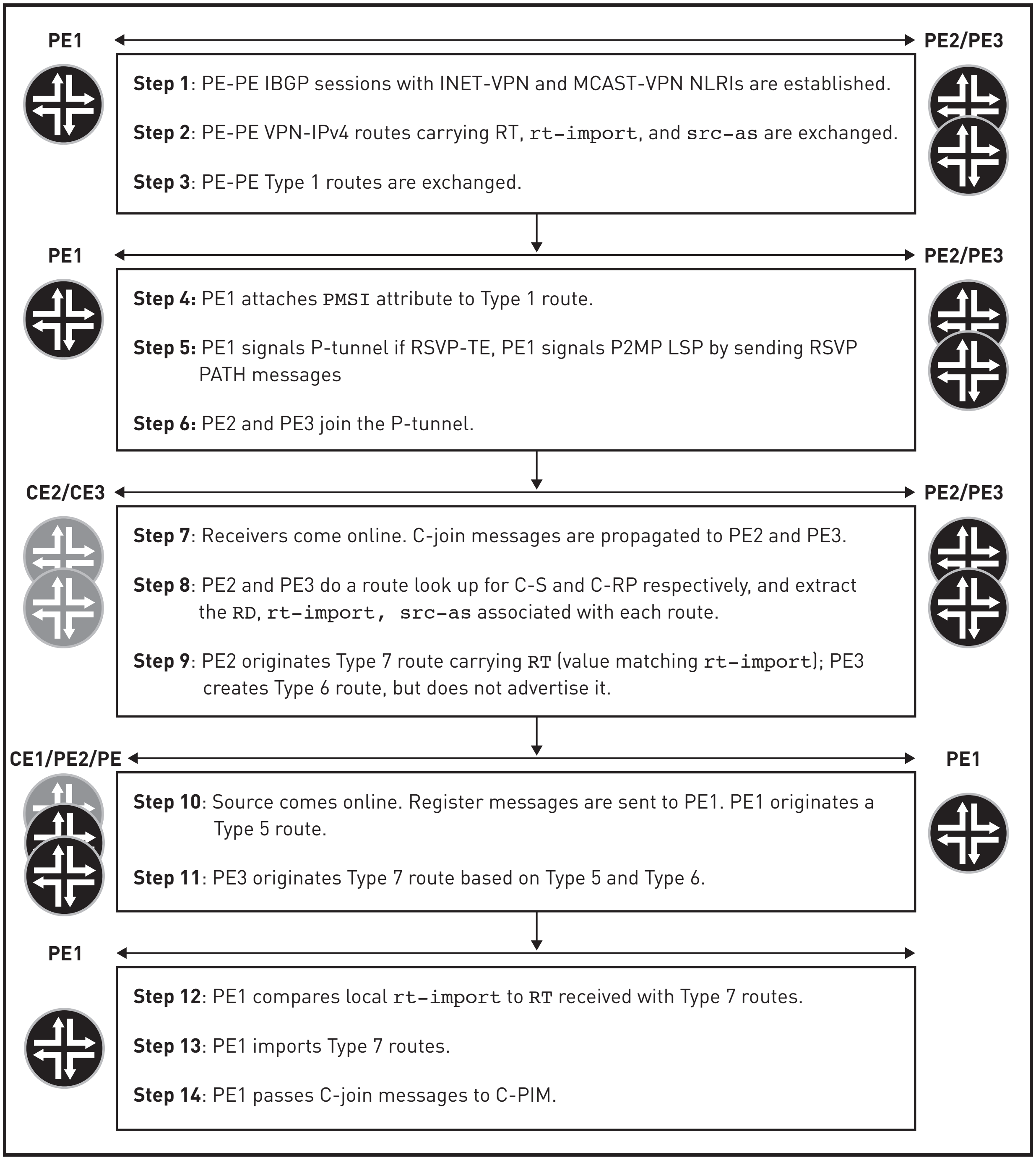


Figure 10: JUNOS NG MVPN routing flow

Turning on NG MVPN Services

NG MVPN services are configured on top of BGP-MPLS unicast VPN services. You can configure a Juniper PE router that is already providing unicast BGP-MPLS VPN connectivity to support multicast VPN connectivity in three steps.

1. Configure the PE routers to support BGP MCAST-VPN address family by adding the `family inet-mvpn signaling` statement to the IBGP configuration. This address family enables PE routers to exchange `mvpn` routes.
2. Configure the PE routers to support MVPN control plane tasks by adding the `protocols mvpn` statement to the `routing-instances` configuration. This statement signals PE routers to initialize their `MVPN` module that is responsible for the majority of NG MVPN control plane tasks.
3. Configure the sender PE router to signal a P-tunnel by adding the `provider-tunnel` statement to the `routing-instances` configuration. You must also configure the tunnel signaling protocol (RSVP-TE or P-PIM) if it was not part of unicast VPN service configuration already.

Once these three statements are configured and each PE router has established IBGP sessions using both INET-VPN and MCAST-VPN address families, four routing tables are automatically created. These tables are `bgp.l3vpn.0`, `bgp.mvpn.0`, `<routing-instance-name>.inet.0`, and `<routing-instance-name>.mvpn.0` (Table 5).

Table 5: Automatically Generated Routing Tables

Automatically Generated Routing Table	Description
`bgp.l3vpn.0`	Populated with VPN-IPv4 routes received from remote PE routers via INET-VPN address family. The routes in the `bgp.l3vpn.0` table are in the form of `RD:IPv4-Address` and carry one or more `RT` communities. In an NG MVPN network, these routes also carry `rt-import` and `src-as` communities.
`bgp.mvpn.0`	Populated by `mvpn` routes (Type 1 – Type 7) received from remote PE routers via the MCAST-VPN address family. Routes in this table carry one or more `RT` communities.
`<routing-instance-name>.inet.0`	Populated by local and remote VPN unicast routes. The local VPN routes are typically learned from local CE routers via protocols like BGP, OSPF, and RIP, or via a static configuration. The remote VPN routes are imported from the `bgp.l3vpn.0` table if their `RT` matches one of the import `RTs` configured for the VPN. When remote VPN routes are imported from the `bgp.l3vpn.0` table, their `RD` is removed, leaving them as regular unicast IPv4 addresses.
`<routing-instance-name>.mvpn.0`	Populated by local and remote `mvpn` routes. The local `mvpn` routes are typically the locally originated routes, such as Type 1 intra-AS AD routes, or Type 7 C-multicast routes. The remote `mvpn` routes are imported from the `bgp.mvpn.0` table based on their `RT`. The import `RT` used for accepting `mvpn` routes into the `<routing-instance-name>.mvpn.0` table is different for C-multicast `mvpn` routes (Type 6 and Type 7) versus non-C-multicast `mvpn` routes (Type 1 – Type 5).

Generating NG MVPN VRF Import and Export Policies

In JUNOS Software, the `Policy` module is responsible for VRF route import and export decisions. You can configure these policies explicitly, or JUNOS can generate them internally for you to reduce user-configured statements and simplify configuration. JUNOS Software generates all necessary policies for supporting NG MVPN import and export decisions. Some of these policies affect normal VPN unicast routes.

The system gives a name to each internal policy it creates. The name of an internal policy starts and ends with a "__" notation. Also the keyword `internal` is added at the end of each internal policy name. You can display these internal policies using a `show policy` command.

Policies that Support Unicast BGP-MPLS VPN Services

A Juniper PE router requires a `vrf-import` and a `vrf-export` policy to control unicast VPN route import and export decisions for a VRF. You can configure these policies explicitly under `[routing-instances <routing-instance-name> vrf-import <import_policy_name>]` and `[routing-instances <routing-instance-name> vrf-export <export_policy_name>]` hierarchies. Alternatively, you can configure only the `RT` for the VRF under the `[routing-instances <routing-instance-name> vrf-target]` hierarchy, and JUNOS then generates these policies automatically for you.

Policy: `vrf-import`
Naming convention: `__vrf-import-<routing-instance-name>-internal__`
Applied to: VPN-IPv4 routes in the `bgp.l3vpn.0` table

Policy: `vrf-export`
Naming convention: `__vrf-export-<routing-instance-name>-internal__`
Applied to: Local VPN routes in the `<routing-instance-name>.inet.0` table

Example: In Figure 1, PE1 creates the following `vrf-import` and `vrf-export` policies based on `vrf-target` of `target:10:1`. In this example, we see that the `vrf-import` policy is constructed to accept a route if the route's `RT` matches `target:10:1`. Similarly, a route is exported with an `RT` of `target:10:1`.

```
user@PE1> show policy __vrf-import-vpna-internal__
Policy __vrf-import-vpna-internal__:
     Term unnamed:
         from community __vrf-community-vpna-common-internal__ [target:10:1]
         then accept
    Term unnamed:
         then reject

user@PE1> show policy __vrf-export-vpna-internal__
Policy __vrf-export-vpna-internal__:
    Term unnamed:
        then community + __vrf-community-vpna-common-internal__ [target:10:1]     accept
```

The values in this example are as follows.
Internal import policy name: `__vrf-import-vpna-internal__`

Internal export policy name: `__vrf-export-vpna-internal__`
`RT` community used in both import and export policies: `__vrf-community-vpna-common-internal__`
`RT` value: `target:10:1`

Policies that Support NG MVPN Services

When you configure the `protocols mvpn` statement under the `[routing-instances <routing-instance-name>]` hierarchy, JUNOS automatically creates three new internal policies: one for export, one for import, and one for handling Type 4 routes.

Policy 1: This policy is used to attach `rt-import` and `src-as` extended communities to VPN-IPv4 routes.
Policy name: `__vrf-mvpn-export-inet-<routing-instance-name>-internal__`
Applied to: All routes in the `<routing-instance-name>inet.0` table

Example: In Figure 1, PE1 creates the following export policy. PE1 adds `rt-import:10.1.1.1:64` and `src-as:65000:0` communities to unicast VPN routes through this policy.

```
user@PE1> show policy __vrf-mvpn-export-inet-vpna-internal__
Policy __vrf-mvpn-export-inet-vpna-internal__:
    Term unnamed:
        then community + __vrf-mvpn-community-rt_import-vpna-internal__ [rt-import:10.1.1.1:64
] community + __vrf-mvpn-community-src_as-vpna-internal__ [src-as:65000:0 ] accept
```

The values in this example are as follows.

Policy name: `__vrf-mvpn-export-inet-vpna-internal__`
`rt-import` community name: `__vrf-mvpn-community-rt_import-vpna-internal__`
`rt-import` community value: `rt-import:10.1.1.1:64`
`src-as` community name: `__vrf-mvpn-community-src_as-vpna-internal__`
`src-as` community value: `src-as:65000:0`

Policy 2: This policy is used to import C-multicast routes from the `bgp.mvpn.0` table to the `<routing-instance-name>.mvpn.0` table.
Policy name: `__vrf-mvpn-import-cmcast-<routing-instance-name>-internal__`
Applied to: C-multicast (`mvpn`) routes in the `bgp.mvpn.0` table

Example: In Figure 1, PE1 creates the following import policy. The policy accepts those C-multicast `mvpn` routes carrying an `RT` of `target:10.1.1.1:64` and installs them in the `vpna.mvpn.0` table.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-vpna-internal__
 Policy __vrf-mvpn-import-cmcast-vpna-internal__:
    Term unnamed:
        from community __vrf-mvpn-community-rt_import-target-vpna-internal__
[target:10.1.1.1:64 ]
        then accept
    Term unnamed:
        then reject
```

The values in this example are as follows.

Policy name: `__vrf-mvpn-import-cmcast-vpna-internal__`
C-multicast import `RT` community: `__vrf-mvpn-community-rt_import-target-vpna-internal__`
Community value: `target:10.1.1.1:64`

Policy 3: This policy is used for importing Type 4 routes and is created by default even if a selective P-tunnel is not configured. The policy affects only Type 4 routes received from receiver PEs.
Policy name: `__vrf-mvpn-import-cmcast-leafAD-global-internal__`
Applied to: Type 4 routes in the `bgp.mvpn.0` table

Example: In Figure 1, PE1 creates the following import policy.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__
Policy __vrf-mvpn-import-cmcast-leafAD-global-internal__:
    Term unnamed:
        from community __vrf-mvpn-community-rt_import-target-global-internal__
[target:10.1.1.1:0 ]
        then accept
    Term unnamed:
        then reject
```

Generating src-as and rt-import Communities

Both `rt-import` and `src-as` communities contain two fields (following their respective keywords). In JUNOS Software, a PE router constructs the `rt-import` community using its `rid` in the first field and a per-VRF unique number in the second field. The `rid` is normally set to the PE's primary loopback IP address. The unique number used in the second field is an internal number derived from the `routing-instance` table index. The combination of the two numbers creates an `rt-import` community that is unique to the originating PE and unique to the VRF from which it is created.

Example: In Figure 1, PE1 creates the following `rt-import` community: `rt-import:10.1.1.1:64`.

Since the `rt-import` community is constructed using the PE router's primary loopback address and the `routing-instance` table index, any event that causes either number to change triggers a change in the value of `rt-import` community. This in turn requires VPN-IPv4 routes to be re-advertised with the new `rt-import` community. Under normal circumstances, the primary loopback address and the `routing-instance` table index numbers do not change. If they do change, JUNOS updates all related internal policies and re-advertises VPN-IPv4 routes with the new `rt-import` and `src-as` values per those policies.

To ensure that the `rt-import` community generated by a PE is unique across VRF tables, the JUNOS `Policy` module restricts the use of primary loopback addresses to NG MVPN internal policies only. You are not permitted to configure an `RT` for any VRF table (MVPN or otherwise) using the primary loopback address. The `commit` fails with an error if the system finds a user-configured `RT` that contains the IP address used in constructing the `rt-import` community.

The global administrator field of the `src-as` community is set to the local AS number of the PE originating the community and the local administrator field is set to `0`. This community is used for inter-AS operations but needs to be carried along with all VPN-IPv4 routes.

Example: In Figure 1, PE1 creates a `src-as` community with a value of `src-as:65000:0`.

Originating Type 1 Intra-AS AD Routes

Every PE router that is participating in the NG MVPN network is required to originate a Type 1 intra-AS AD route. In JUNOS Software, the `MVPN` module is responsible for installing the intra-AS AD route in the local `<routing-instance-name>.mvpn.0` table. All PE routers advertise their local Type 1 routes to each other.

Example: In Figure 1, PE1 installs the following intra-AS AD route in its `vpna.mvpn.0` table. The route is installed by the MVPN protocol (meaning it was the `MVPN` module that originated the route), and the mask for the entire route is `/240`.

```
user@PE1> show route table vpna.mvpn.0

vpna.mvpn.0: 6 destinations, 9 routes (6 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:10.1.1.1:1:10.1.1.1/240
                   *[MVPN/70] 04:09:44, metric2 1
                      Indirect
```

Attaching RT Community to Type 1 Routes

Intra-AS AD routes are picked up by BGP protocol from the `<routing-instance-name>.mvpn.0` table and advertised to the remote PE routers via the MCAST-VPN address family. By default, intra-AS AD routes carry the same `RT` community that is attached to the unicast VPN-IPv4 routes. If the unicast and multicast network topologies are not congruent, then you can configure a different set of import `RT` and export `RT` communities for non-C-multicast `mvpn` routes (C-multicast mvpn routes always carry a dynamic import `RT`).

Multicast `RTs` are configured using `import-target` and `export-target` statements under the `[routing-instances <routing-instance-name> protocols mvpn route-target]` hierarchy.

JUNOS Software creates two additional internal policies in response to configuring multicast RTs. These polices are applied to non-C-multicast mvpn routes during import and export decisions. Multicast VRF internal import and export polices follow a naming convention similar to unicast VRF import and export policies. The contents of these polices are also similar to policies applied to unicast VPN routes.

Multicast VRF import policy: __vrf-mvpn-import-target-<routing-instance-name>-internal__
Multicast VRF export policy: __vrf-mvpn-export-target-<routing-instance-name>-internal__

Example: In Figure 1, PE1 creates the following internal mvpn policies if import-target and export-target are configured to be target:10:2.

```
user@PE1> show policy __vrf-mvpn-import-target-vpna-internal__
Policy __vrf-mvpn-import-target-vpna-internal__:
    Term unnamed:
        from community __vrf-mvpn-community-import-vpna-internal__ [target:10:2 ]
        then accept
    Term unnamed:
        then reject

user@PE1> show policy __vrf-mvpn-export-target-vpna-internal__
Policy __vrf-mvpn-export-target-vpna-internal__:
    Term unnamed:
        then community + __vrf-mvpn-community-export-vpna-internal__ [target:10:2 ] accept
```

The values in this example are as follows.

Multicast import RT community: __vrf-mvpn-community-import-vpna-internal__
Multicast export RT community: __vrf-mvpn-community-export-vpna-internal__
Value: target:10:2

Attaching PMSI Attribute to Type 1 Routes

The PMSI attribute is originated and attached to Type 1 intra-AS AD routes by the sender PE routers when the provider-tunnel statement is configured under the [routing-instances <routing-instance-name>] hierarchy. Since P-tunnels are signaled by the sender PE routers, this statement is not necessary on the PE routers that are known to have VPN multicast receivers only.

If the P-tunnel configured is PIM-SM (ASM), then the PMSI attribute carries the IP address of the sender-PE and P-tunnel group address. The P-tunnel group address is assigned by the service provider (through configuration) from provider's multicast address space and not to be confused by the multicast addresses used by the VPN customer.

Example: In Figure 1, PE1 originates the following PMSI attribute if the P-tunnel is signaled by PIM-SM (ASM).

```
PMSI: Flags 0:PIM-SM:label[0:0:0]:Sender 10.1.1.1 Group 239.1.1.1
```

Tunnel type

Originating router's IP address

P-group address

If the P-tunnel configured is RSVP-TE, then the PMSI attribute carries the RSVP-TE P2MP Session Object. This P2MP Session Object is used as the identifier for the parent P2MP LSP and contains the following fields (Figure 11).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       P2MP ID                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MUST be zero                 |      Tunnel ID                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Tunnel ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: RSVP-TE P2MP session object format

In JUNOS Software, `P2MP ID` and `Extended Tunnel ID` fields are set to the `rid` of the sender PE. The `Tunnel ID` is set to the `Port number` used for the P2MP RSVP session that is unique for the length of the RSVP session.

Example: In Figure 1, PE1 originates the following `PMSI` attribute. C-Type 13 (`Session_13`) indicates that this is an IPv4 P2MP LSP as defined in RFC 4875. The `P2MP ID` and `Extended Tunnel ID` fields are set to PE1's loopback address (`10.1.1.1`). The `Tunnel ID` is set to `6574`, which is the `Port number` of the RSVP session originated from PE1 to PE2 and PE3.

```
PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:6574:10.1.1.1]
```

Tunnel type　No MPLS label　P2MP session object

Example: In Figure 1, PE1 signals the following RSVP sessions to PE2 and PE3 (using `Port number 6574`). In this example, we see that PE1 is signaling a P2MP LSP named `10.1.1.1:65535:mvpn:vpna` with two sub-LSPs. Both sub-LSPs `10.1.1.3:10.1.1.1:65535:mvpn:vpna` and `10.1.1.2:10.1.1.1:65535:mvpn:vpna`. use the same RSVP `Port number` (`6574`) as the parent P2MP LSP.

```
user@PE1> show rsvp session p2mp detail
Ingress RSVP: 2 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2

10.1.1.3
  From: 10.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: 10.1.1.3:10.1.1.1:65535:mvpn:vpna, LSPpath: Primary
  P2MP LSPname: 10.1.1.1:65535:mvpn:vpna
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299968
  Resv style: 1 SE, Label in: -, Label out: 299968
  Time left:    -, Since: Wed May 27 07:36:22 2009
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 6574 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  RESV rcvfrom: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  Explct route: 10.12.100.6 10.12.100.22
  Record route: <self> 10.12.100.6 10.12.100.22

10.1.1.2
  From: 10.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: 10.1.1.2:10.1.1.1:65535:mvpn:vpna, LSPpath: Primary
```

```
  P2MP LSPname: 10.1.1.1:65535:mvpn:vpna
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299968
  Resv style: 1 SE, Label in: -, Label out: 299968
  Time left:    -, Since: Wed May 27 07:36:22 2009
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 6574 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  RESV rcvfrom: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  Explct route: 10.12.100.6 10.12.100.9
  Record route: <self> 10.12.100.6 10.12.100.9
Total 2 displayed, Up 2, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sender-Only and Receiver-Only Sites

In JUNOS Software, you can configure a PE router to be a sender-site only or a receiver-site only. These options are configured under the `[routing-instances <routing-instance-name> protocols mvpn]` hierarchy using `sender-site` and `receiver-site` statements.

- A sender-site only PE router does not join the P-tunnels advertised by remote PE routers.
- A receiver-site only PE router does not send a `PMSI` attribute.

The `commit` fails if you configure `receiver-site` and `provider-tunnel` statements in the same VPN.

Signaling P-tunnels and Data Plane Setup

In an NG MVPN network, P-tunnel information is communicated to the receiver PEs in an out-of-band manner. This information is advertised via BGP and is independent of the actual tunnel signaling process. Once the tunnel is signaled, the sender PE binds the VRF table to the locally configured tunnel. The receiver PEs bind the tunnel signaled to the VRF table where the Type 1 AD route with the matching `PMSI` attribute is installed. The same binding process is used for both PIM and RSVP-TE signaled P-tunnels.

P-tunnels Signaled by PIM (Inclusive)

A sender PE router configured to use an inclusive PIM-SM (ASM) P-tunnel for a VPN creates a multicast tree (using the P-group address configured) in the service provider network. This tree is rooted at the sender PE and has the receiver PEs as the leaves. VPN multicast packets received from the local VPN source are encapsulated by the sender PE with a multicast GRE header containing the P-group address configured for the VPN. These packets are then forwarded on the service provider network as normal IP multicast packets per normal P-PIM procedures. At the leaf nodes, the GRE header is stripped and the packets are passed on to the local VRF C-PIM protocol for further processing.

In JUNOS Software, a logical interface called `mt` is used for GRE encapsulation and de-encapsulation of VPN multicast packets. The `mt` interface is created automatically if a Tunnel PIC is present.

- Encapsulation subinterfaces are created from an `mt-x/y/z.[32768-49151]`range.
- De-encapsulation subinterfaces are created from an `mt-x/y/z.[49152-65535]`range.

The `mt` subinterfaces act as pseudo upstream or downstream interfaces between C-PIM and P-PIM.

In the following two examples, assume that the network in Figure 1 uses PIM-SM (ASM) signaled GRE tunnels as the tunneling technology.

Example: In Figure 1, PE1 creates the following `mt` subinterface. The logical interface number is `32768`, indicating that this sub-unit will be used for GRE encapsulation.

```
user@PE1> show interfaces mt-0/1/0 terse
Interface               Admin Link Proto    Local                 Remote
mt-0/1/0                up    up
mt-0/1/0.32768          up    up   inet
                                   inet6
```

Example: In Figure 1, PE2 creates the following `mt` subinterface. The logical interface number is `49152`, indicating that this sub-unit will be used for GRE de-encapsulation.

```
user@PE2> show interfaces mt-0/1/0 terse
Interface               Admin Link Proto    Local                 Remote
mt-0/1/0                up    up
mt-0/1/0.49152          up    up   inet
                                   inet6
```

P-PIM and C-PIM on the Sender PE

The sender PE installs a `Local join` entry in its P-PIM database for each VRF table configured to use PIM as the P-tunnel. The outgoing interface list (`OIL`) of this entry points to the core-facing interface. Since the P-PIM entry is installed as `Local`, the sender PE sets `Source` address to its primary loopback IP address.

Example: In Figure 1, PE1 installs the following entry in its P-PIM database.

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local Source
    Keepalive timeout: 339
   Downstream neighbors:
        Interface: fe-0/2/3.0
            10.12.100.6 State: Join Flags: S Timeout: 195

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

On the VRF side of the sender PE, C-PIM installs a `Local Source` entry in its C-PIM database for the active local VPN source. The `OIL` of this entry points to `Pseudo-MVPN`, indicating that the downstream interface points to the receivers in the NG MVPN network.

Example: In Figure 1, PE1 installs the following entry in its C-PIM database.

```
user@PE1> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 0
    Downstream neighbors:
        Interface: Pseudo-MVPN
```

The forwarding entry corresponding to the C-PIM `Local Source` (or `Local RP`) on the sender PE router points to the `mt` encapsulation subinterface as the downstream interface. This indicates that the local multicast data packets will be encapsulated as they are passed on to the P-PIM protocol.

Example: In Figure 1, PE1 has the following multicast forwarding entry for group `224.1.1.1`. The `Upstream interface` is the PE-CE interface and the `Downstream interface` is the `mt` encapsulation subinterface.

```
user@PE1> show multicast route extensive instance vpna group 224.1.1.1
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: fe-0/2/0.0
    Downstream interface list:
        mt-0/1/0.32768
    Session description: ST Multicast Groups
    Statistics: 7 kBps, 79 pps, 719738 packets
    Next-hop ID: 262144
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
```

P-PIM and C-PIM on the Receiver PE

On the receiver PE, multicast data packets received from the network are de-encapsulated as they are passed through the `mt` de-encapsulation interface.

The P-PIM database on the receiver PE contains two P-joins. One is for P-RP, and the other is for the sender PE. For both entries, the `OIL` contains the `mt` de-encapsulation interface from which the GRE header is stripped. The `Upstream interface` for P-joins is the core-facing interface that faces towards the sender PE.

Example: In Figure 1, PE3 has the following entry in its P-PIM database. The `Downstream interface` points to the GRE de-encapsulation subinterface.

```
user@PE3> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: *
    RP: 10.1.1.10
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/3.0
    Upstream neighbor: 10.12.100.21
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: mt-1/2/0.49152
            10.12.53.13 State: Join Flags: SRW  Timeout: Infinity

Group: 239.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: so-0/0/3.0
    Upstream neighbor: 10.12.100.21
    Upstream state: Join to Source
    Keepalive timeout: 351
    Downstream neighbors:
        Interface: mt-1/2/0.49152
            10.12.53.13 State: Join Flags: S   Timeout: Infinity

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

On the VRF side of the receiver PE, C-PIM installs a `join` entry in its C-PIM database. The `OIL` of this entry points to the local VPN interface, indicating active local receivers. The `Upstream protocol`, `interface`, and `neighbor` of this entry point to the NG-MVPN network.

Example: In Figure 1, PE3 has the following entry in its C-PIM database.

```
user@PE3> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 10.12.53.1
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP
    Upstream neighbor: Through MVPN
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: SRW  Timeout: Infinity

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse
    Upstream protocol: BGP
    Upstream interface: Through BGP
```

```
    Upstream neighbor: Through MVPN
    Upstream state: Join to Source
    Keepalive timeout:
    Downstream neighbors:
        Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: S Timeout: 195

Instance: PIM.vpna Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

The forwarding entry corresponding to the C-PIM entry on the receiver PE uses the `mt` de-encapsulation subinterface as the `Upstream interface`.

Example: In Figure 1, PE3 installs the following multicast forwarding entry for the local receiver.

```
user@PE3> show multicast route extensive instance vpna group 224.1.1.1
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: mt-1/2/0.49152
    Downstream interface list:
        so-0/2/0.0
    Session description: ST Multicast Groups
    Statistics: 1 kBps, 10 pps, 149 packets
    Next-hop ID: 262144
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
```

P-tunnels Signaled by RSVP-TE (Inclusive and Selective)

JUNOS Software supports signaling both inclusive and selective P-tunnels by RSVP-TE P2MP LSPs. You can configure a combination of inclusive and selective P-tunnels per VPN.

- If you configure a VPN to use an inclusive P-tunnel, the sender PE signals one P2MP LSP for the VPN.
- If you configure a VPN to use selective P-tunnels, the sender PE signals a P2MP LSP for each selective tunnel configured.

Sender (ingress) PEs and receiver (egress) PEs play different roles in the P2MP LSP setup. Sender PEs are mainly responsible for initiating the parent P2MP LSP and the sub-LSPs associated with it. Receiver PEs are responsible for setting up state such that they can forward packets received over a sub-LSP to the correct VRF table (binding P-tunnel to the VRF).

Inclusive Tunnels: Ingress PE P2MP LSP Setup

The P2MP LSP and associated sub-LSPs are signaled by the ingress PE. The information about the P2MP LSP is advertised to egress PEs in the `PMSI` attribute via BGP.

The ingress PE router signals P2MP sub-LSPs by originating P2MP RSVP `PATH` messages towards egress PE routers. The ingress PE learns the identity of the egress PEs from Type 1 routes installed in its `<routing-instance-name>.mvpn.0` table. Each RSVP `PATH` message carries an `S2L_Sub_LSP Object` along with the `P2MP Session Object`. The `S2L_Sub_LSP Object` carries a 4-byte sub-LSP destination (egress) IP address.

In JUNOS Software, sub-LSPs associated with a P2MP LSP can be signaled automatically by the system or via a static sub-LSP configuration. When they are automatically signaled, the system chooses a name for the P2MP LSP and each sub-LSP associated with it using the following naming convention.

P2MP LSPs naming convention:

```
<ingress PE rid>:<a per VRF unique number>:mvpn:<routing-instance-name>
```

Sub-LSPs naming convention:

```
<egress PE rid>:<ingress PE rid>:<a per VRF unique>:mvpn:<routing-instance-name>
```

Example: In Figure 1, PE1 originates the following LSPs.

Parent P2MP LSP: `10.1.1.1:65535:mvpn:vpna`
Sub-LSPs: `10.1.1.2:10.1.1.1:65535:mvpn:vpna` (PE1 to PE2) and
`10.1.1.3:10.1.1.1:65535:mvpn:vpna` (PE1 to PE3)

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 1 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2
To              From            State Rt P  ActivePath    LSPname
10.1.1.2        10.1.1.1        Up     0 *                10.1.1.2:10.1.1.1:65535:mvpn:vpna
10.1.1.3        10.1.1.1        Up     0 *                10.1.1.3:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

The values in this example are as follows.

I-PMSI P2MP LSP name: `10.1.1.1:65535:mvpn:vpna`
I-PMSI P2MP sub-LSP name (to PE2): `10.1.1.2:10.1.1.1:65535:mvpn:vpna`
I-PMSI P2MP sub-LSP name (to PE3): `10.1.1.3:10.1.1.1:65535:mvpn:vpna`

Inclusive Tunnels: Egress PE P2MP LSP Setup

An egress PE router responds to an RSVP `PATH` message by originating an RSVP `RESV` message per normal RSVP procedures. The `RESV` message contains the MPLS label allocated by the egress PE for this sub-LSP and is forwarded hop by hop towards the ingress PE, thus setting up state on the network.

Example: In Figure 1, PE2 assigns label `299840` for the sub-LSP `10.1.1.2:10.1.1.1:65535:mvpn:vpna`.

```
user@PE2> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 1 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.1.1.2   10.1.1.1   Up       0  1 SE  299840        - 10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 1 displayed, Up 1, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Example: In Figure 1, PE3 assigns label 16 for the sub-LSP `10.1.1.3:10.1.1.1:65535:mvpn:vpna`.

```
user@PE3> show mpls lsp p2mp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 1 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname
10.1.1.3   10.1.1.1   Up       0  1 SE      16        - 10.1.1.3:10.1.1.1:65535:mvpn:vpna
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Inclusive Tunnels: Egress PE Data Plane Setup

The egress PE router installs a forwarding entry in its `mpls` table for the label it allocated for the sub-LSP. The MPLS label is installed with a Pop operation[1] and the packet is passed on to the VRF table for a second route lookup. The second lookup on the egress PE is necessary in order for VPN multicast data packets to be processed inside the VRF table using normal C-PIM procedures.

Example: In Figure 1, PE3 installs the following label entry in its `mpls` forwarding table.

```
user@PE3> show route table mpls label 16

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[VPN/0] 03:03:17
                      to table vpna.inet.0, Pop
```

In JUNOS Software, VPN multicast routing entries are stored in the <routing-instance-name>.inet.1 table, which is where the second route lookup occurs. In the example above, even though `vpna.inet.0` is listed as the routing table where the second lookup happens after the Pop operation, internally the lookup is pointed to the `vpna.inet.1` table.

Example: In Figure 1, PE3 contains the following entry in its VPN multicast routing table.

```
user@PE3> show route table vpna.inet.1

vpna.inet.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

224.1.1.1,192.168.1.2/32*[MVPN/70] 00:04:10
                      Multicast (IPv4)
```

[1] A Pop operation removes the top MPLS label.

Example: In Figure 1, PE3 contains the following VPN multicast forwarding entry corresponding to the multicast routing entry for the Local `join`. The `Upstream interface` points to `lsi.0` and the `Downstream interface` (`OIL`) points to `so-0/2/0.0` (towards local receivers). The `Upstream protocol` is `MVPN` because the VPN multicast source is reachable via the NG-MVPN network. The `lsi.0` interface is similar to the `mt` interface used when PIM-based P-tunnels are used. The `lsi.0` interface is used for removing the top MPLS header.

```
user@PE3> show multicast route extensive instance vpna
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: lsi.0
    Downstream interface list:
        so-0/2/0.0
    Session description: ST Multicast Groups
    Statistics: 1 kBps, 10 pps, 3472 packets
    Next-hop ID: 262144
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Family: INET6
```

The requirement for a double route lookup on the VPN packet header requires two additional configuration statements on the egress PE routers when P-tunnels are signaled by RSVP-TE.

First, since the top MPLS label used for the P2MP sub-LSP is actually tied to the VRF table on the egress PE routers, the penultimate-hop popping (PHP) operation must be disabled. PHP allows the penultimate router (router before the egress PE) to remove the top MPLS label. PHP works well for VPN unicast data packets because they typically carry two MPLS labels: one for the VPN and one for the transport LSP. Once the LSP label is removed, unicast VPN packets still have a VPN label that can be used for determining the VPN to which the packets belong. VPN multicast data packets, on the other hand, carry only one MPLS label that is directly tied to the VPN. Therefore, the MPLS label carried by VPN multicast packets must be preserved until the packets reach the egress PE. Normally, PHP must be disabled through manual configuration. To simplify configuration, PHP is disabled by default on Juniper PE routers when you configure the `protocols mvpn` statement under the `routing-instance` hierarchy. You do not need to explicitly disable it.

Second, in JUNOS Software, VPN labels associated with a VRF table can be allocated in two ways.

- Allocate a unique label for each VPN next hop (PE-CE interface). This is the default behavior.
- Allocate one label for the entire VRF table, which requires additional configuration. Only allocating a label for the entire VRF table allows a second lookup on the VPN packet's header. Therefore, PE routers supporting NG-MVPN services must be configured to allocate labels for the VRF table. There are two ways to do this (Figure 12).
    - One is by configuring a special tunnel interface called `vt` under the `[routing-instances <routing-instance-name> interfaces]` hierarchy, which requires a Tunnel PIC.
    - The second is by configuring the `vrf-table-label` statement under the `[routing-instances <routing-instance-name>]` hierarchy, which does not require a Tunnel PIC.

Both of these options enable an egress PE to perform two route lookups, however there are some differences in the way in which the second lookup is done.

If `vt` is configured, the allocated label is installed in the `mpls` table with a `Pop` operation and a forwarding next hop pointing to the `vt` interface.

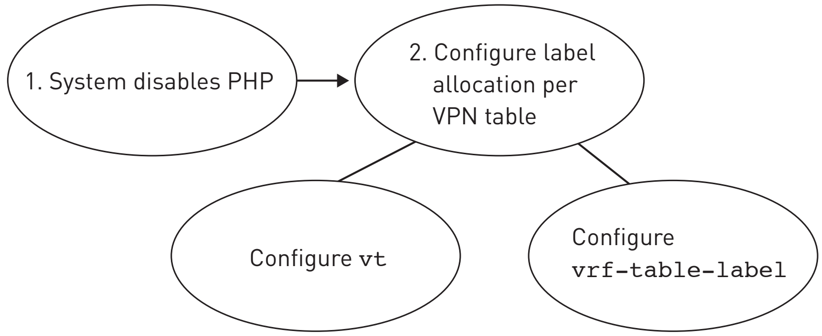


Figure 12: Enabling double route lookup on VPN packet headers

Example: In Figure 1, PE2 uses a `vt` configuration and installs the following entry in its `mpls` table. The label associated with the P2MP sub-LSP (`299840`) is installed with a Pop and a forward operation with `vt-0/1/0.0` being the next hop. VPN multicast packets received from the core exit the `vt-0/1/0.0` interface without their MPLS header and the egress PE2 does a second lookup on the packet header in the `vpna.inet.1` table.

```
user@PE2> show route table mpls label 299840

mpls.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299840             *[VPN/0] 00:00:22
                    > via vt-0/1/0.0, Pop
```

If the `vrf-table-label` is configured, the allocated label is installed in the `mpls` table with a `Pop` operation and the forwarding entry points to the `<routing-instance-name>.inet.0` table (which internally triggers the second lookup to be done in the `<routing-instance-name>.inet.1` table).

Example: In Figure 1, PE3 uses `vrf-table-label` configuration and installs the following entry in its `mpls` table.

```
user@PE3> show route table mpls label 16

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[VPN/0] 03:03:17
                      to table vpna.inet.0, Pop
```

Configuring label allocation for each VRF table affects both unicast VPN and `mvpn` routes. However, you can enable per-VRF label allocation for `mvpn` routes only if per-VRF allocation is configured via `vt`. This feature is configured via `multicast` and `unicast` keywords under the `[routing-instances <routing-instance-name> interface vt-x/y/z.0]` hierarchy.

Note that configuring `vrf-table-label` enables per-VRF label allocation for both unicast and `mvpn` routes and can not be turned off for either type of routes (it is either on or off for both).

If a PE is a bud router, meaning it has local receivers and also forwards MPLS packets received over a P2MP LSP downstream to other P and PE routers, then there is a difference in how `vrf-table-label` and `vt` work. With `vrf-table-label` configuration, the bud PE receives two copies of the packet from the penultimate router: one to be forwarded to local receivers and the other to be forwarded to downstream P and PE routers. With `vt` configuration, the PE receives a single copy of the packet.

Inclusive Tunnels: Ingress and Branch PE Data Plane Setup

On the ingress PE, local VPN data packets are encapsulated with the MPLS label received from the network for sub-LSPs.

Example: On the ingress PE1, VPN multicast data packets are encapsulated with MPLS label `300016` (advertised by P1 per normal RSVP RESV procedures) and forwarded towards P1 down the sub-LSPs `10.1.1.3:10.1.1.1:65535:mvpn:vpna` and `10.1.1.2:10.1.1.1:65535:mvpn:vpna`.

```
user@PE1> show rsvp session
Ingress RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.1.1.3   10.1.1.1   Up       0  1 SE       -   300016 10.1.1.3:10.1.1.1:65535:mvpn:vpna
10.1.1.2   10.1.1.1   Up       0  1 SE       -   300016 10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

RFC 4875 describes a branch node as "an LSR that replicates the incoming data on to one or more outgoing interfaces." On a branch node, the incoming data carrying an MPLS label is replicated onto one or more outgoing interfaces that can use different MPLS labels. Branch nodes keep track of incoming and outgoing labels associated with P2MP LSPs.

Example: In Figure 1, branch node P1 has the incoming label `300016` and outgoing labels `16` for sub-LSP `10.1.1.3:10.1.1.1:65535:mvpna` (to PE3) and 299840 for sub-LSP `10.1.1.2:10.1.1.1:65535:mvpn:vpna` (to PE2).

```
user@P1> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
10.1.1.3   10.1.1.1   Up       0  1 SE  300016       16 10.1.1.3:10.1.1.1:65535:mvpn:vpna
10.1.1.2   10.1.1.1   Up       0  1 SE  300016   299840 10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0
```

Example: The corresponding forwarding entry on P1 shows that the packets coming in with one MPLS label (`300016`) are swapped with labels `16` and `299840` and forwarded out through their respective interfaces (`so-0/0/3.0` and `so-0/0/1.0` respectively towards PE2 and PE3).

```
user@P1> show route table mpls label 300016

mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

300016             *[RSVP/7] 01:58:15, metric 1
                    > via so-0/0/3.0, Swap 16
                      via so-0/0/1.0, Swap 299840
```

Selective Tunnels: Type 3 S-PMSI AD and Type 4 Leaf AD Routes

Selective P-tunnels are configured using the `selective` statement under the `[routing-instances <routing-instance-name> provider-tunnel]` hierarchy. You can configure a threshold to trigger the signaling of a selective P-tunnel. Configuring a `selective` statement triggers the following events.

First, the ingress PE originates a Type 3 S-PMSI AD route. The S-PMSI AD route contains the `RD` of the VPN where the tunnel is configured and the (C-S, C-G) pair that will be using the selective P-tunnel.

In this section assume that PE1 is signaling a selective tunnel for (`192.168.1.2, 224.1.1.1`) and PE3 has an active receiver.

Example: In Figure 1, PE1 installs the following Type 3 route upon selective P-tunnel configuration.

```
user@PE1> show route table vpna.mvpn.0 | find 3:
3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1/240
                   *[MVPN/70] 00:05:07, metric2 1
                      Indirect
```

Second, the ingress PE attaches a `PMSI` attribute to a Type 3 route. This `PMSI` attribute is similar to the `PMSI` attribute advertised for inclusive P-tunnels with one difference: the `PMSI` attribute carried with Type 3 routes has its `Flags` bit set to `Leaf Information Required`. This means that the sender PE router is requesting receiver PE routers to send a Type 4 route if they have active receivers for the (C-S, C-G) carried in the Type 3 route. Also, remember that for each selective P-tunnel, a new P2MP and associated sub-LSPs are signaled. The `PMSI` attribute of a Type 3 route carries information about the new P2MP LSP.

Example: In Figure 1, PE1 advertises the following Type 3 route and the `PMSI` attribute. The P2MP `Session Object` included in the `PMSI` attribute has a different `Port number` (`29499`) than the one used for the inclusive tunnel (`6574`) indicating that this is a new P2MP tunnel.

```
user@PE1> show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn | find 3:
* 3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1/240 (1 entry, 1 announced)
 BGP group int type Internal
     Route Distinguisher: 10.1.1.1:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:10:1
     PMSI: Flags 1:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:29499:10.1.1.1]
```

Egress PE routers with active receivers should respond to a Type 3 route by originating a Type 4 leaf AD route. A leaf AD route contains a route key and the originating router's IP address fields. The Route Key field of the Leaf AD route contains the original Type 3 route that was received. The Originating Router's IP Address field is set to the `rid` of the PE originating the leaf AD route.

The ingress PE adds each egress PE that originated the leaf AD route as a leaf (destination of the sub-LSP for selective P2MP LSP). Similarly, the egress PE router that originated the leaf AD route sets up forwarding state to start receiving data through the selective P-tunnel.

Egress PEs advertise Type 4 routes with an `RT` that is specific to the PE signaling the selective P-tunnel. This `RT` is in the form of `target:<rid of the sender PE>:0`. The sender PE (the PE signaling the selective P-tunnel) applies a special internal import policy to Type 4 routes that looks for an `RT` with its own `rid`.

Example: In Figure 1, PE3 originates the following Type 4 route. The local Type 4 route is installed by the `mvpn` module.

```
user@PE3> show route table vpna.mvpn | find 4:3:
4:3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1:1.1.1.3/240
                   *[MVPN/70] 00:15:29, metric2 1
                      Indirect
```

Example: PE3 advertises the local Type 4 route with the following `RT` community. This `RT` carries the IP address of the sender PE (`10.1.1.1`) followed by a `0`.

```
user@PE3> show route advertising-protocol bgp 10.1.1.1 table vpna.mvpn detail | find 4:3:

* 4:3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1:10.1.1.3/240 (1 entry, 1 announced)
 BGP group int type Internal
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:10.1.1.1:0
```

Example: PE1 (the PE signaling the selective P-tunnel) applies the following import policy to Type 4 routes. The routes are accepted if their `RT` matches `target:10.1.1.1:0`.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__
Policy __vrf-mvpn-import-cmcast-leafAD-global-internal__:
    Term unnamed:
        from community __vrf-mvpn-community-rt_import-target-global-internal__
[target:10.1.1.1:0 ]
        then accept
    Term unnamed:
        then reject
```

For each selective P-tunnel configured, a Type 3 route is advertised and a new P2MP LSP is signaled. P2MP LSPs created by JUNOS Software for selective P-tunnels are named using the following naming convention.

Selective P2MP LSPs naming convention:
`<ingress PE rid>:<a per VRF unique number>:mv<a unique number>:<routing-instance-name>`

Selective P2MP sub-LSP naming convention:
`<egress PE rid>:<ingress PE rid>:<a per VRF unique>:mv<a unique number>:<routing-instance-name>`

Example: In Figure 1, PE1 signals P2MP LSP `10.1.1.1:65534:mv5:vpna` with one sub-LSP `10.1.1.3:10.1.1.1:65534:mv5:vpna`. The first P2MP LSP `10.1.1.1:65534:mvpn:vpna` is the LSP created for the inclusive tunnel.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 2 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
10.1.1.3   10.1.1.1   Up     0 *                      10.1.1.3:10.1.1.1:65535:mvpn:vpna
10.1.1.2   10.1.1.1   Up     0 *                      10.1.1.2:10.1.1.1:65535:mvpn:vpna
P2MP name: 10.1.1.1:65535:mv5:vpna, P2MP branch count: 1
To              From            State Rt P     ActivePath       LSPname
10.1.1.3   10.1.1.1   Up     0 *                      10.1.1.3:10.1.1.1:65535:mv5:vpna
Total 3 displayed, Up 3, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

The values in this example are as follows.

I-PMSI P2MP LSP name: `10.1.1.1:65535:mvpn:vpna`
I-PMSI P2MP sub-LSP name (to PE2): `10.1.1.2:10.1.1.1:65535:mvpn:vpna`
I-PMSI P2MP sub-LSP name (to PE3): `10.1.1.3:10.1.1.1:65535:mvpn:vpna`
S-PMSI P2MP LSP name: `10.1.1.1:65535:mv5:vpna`
S-PMSI P2MP sub-LSP name (to PE3): `10.1.1.3:10.1.1.1:65535:mv5:vpna`

C-multicast Route Exchange (Type 7 Routes Only)

This section describes PE-PE distribution of Type 7 routes discussed earlier.

In source-tree-only mode, a receiver PE router generates and installs a Type 6 route in its `<routing-instance-name>.mvpn.0` table in response to receiving a (C-*, C-G) message from a local receiver, but does not advertise this route to other PE routers via BGP. The receiver PE waits for a Type 5 route corresponding to the C-join.

Type 5 routes carry information about active sources and can be advertised by any PE router. In JUNOS Software, a PE router originates a Type 5 route if one of the following conditions occurs.

- PE starts receiving multicast data directly from a VPN multicast source.
- PE is the C-RP and starts receiving C-PIM `register` messages.
- PE has an MSDP session with the C-RP and starts receiving MSDP SA routes.

Once both Type 6 and Type 5 routes are installed in the `<routing-instance-name>.mvpn.0` table, the receiver PE is ready to originate a Type 7 route.

Advertising C-multicast Routes via BGP

If the C-join received over a VPN interface is a source tree join (C-S, C-G), then the receiver PE simply originates a Type 7 route (Step 7 below). If the C-join is a shared tree join (C-*, C-G), then the receiver PE needs to go through a few steps (Steps 1-7) before originating a Type 7 route.

Note that in Figure 1, PE1 is the C-RP that is conveniently located in the same router as the sender PE. If the sender PE and the PE acting as (or MSDP peering with) the C-RP are different, then the VPN multicast `register` messages first need to be delivered to the PE acting as the C-RP that is responsible for originating the Type 5 route.

Step 1. A PE router that receives a (C-*, C-G) `join` message processes the message using normal C-PIM procedures and updates its C-PIM database accordingly.

Example: In Figure 1, PE3 creates the following entry in its C-PIM database upon receiving (`*, 224.1.1.1`) C-join from CE3.

```
user@PE3> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 10.12.53.1
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP
    Upstream neighbor: Through MVPN
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: SRW  Timeout: Infinity
```

Step 2. The (C-*, C-G) entry in the C-PIM database triggers the generation of a Type 6 route that is then installed in the `<routing-instance-name>.mvpn.0` table by C-PIM. The Type 6 route uses the C-RP IP address as the source.

Example: In Figure 1, PE3 installs the following Type 6 route in the `vpna.mvpn.0` table.

```
user@PE3> show route table vpna.mvpn.0 detail | find 6:10.1.1.1
6:10.1.1.1:1:65000:32:10.12.53.1:32:224.1.1.1/240 (1 entry, 1 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4), Next hop index: 262144
                Next-hop reference count: 11
                State: <Active Int>
                Age: 1d 1:32:58
                Task: PIM.vpna
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: no-advertise target:10.1.1.1:64
```

Step 3. The `RD` and `RT` attached to the Type 6 route are learned from a route lookup in the `<routing-instance-name>.inet.0` table for the IP address of the C-RP.

Example: In Figure 1, PE3 has the following entry for C-RP `10.12.53.1` in the `vpna.inet.0` table.

```
user@PE3> show route table vpna.inet.0 10.12.53.1 detail

vpna.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
10.12.53.1/32 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.1.1.1:1
                Next hop type: Indirect
                Next-hop reference count: 6
                Source: 10.1.1.1
                Next hop type: Router, Next hop index: 588
                Next hop: via so-0/0/3.0, selected
                Label operation: Push 16, Push 299808(top)
                Protocol next hop: 10.1.1.1
                Push 16
                Indirect next hop: 8da91f8 262143
                State: <Secondary Active Int Ext>
                Local AS: 65000 Peer AS: 65000
                Age: 4:49:25    Metric2: 1
                Task: BGP_65000.10.1.1.1+179
                Announcement bits (1): 0-KRT
                AS path: I
                Communities: target:10:1 src-as:65000:0 rt-import:10.1.1.1:64
                Import Accepted
                VPN Label: 16
                Localpref: 100
                Router ID: 10.1.1.1
                Primary Routing Table bgp.l3vpn.0
```

Step 4. Once the VPN source starts transmitting data, the first PE that becomes aware of the active source (either by receiving `register` messages or the MSDP SA routes) installs a Type 5 route in its VRF `mvpn` table.

Example: In Figure 1, PE1 starts receiving C-PIM `register` messages from CE1 and installs the following entry in the `vpna.mvpn.0` table.

```
user@PE1> show route table vpna.mvpn.0 detail | find 5:10.1.1.1
5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4)
                Next-hop reference count: 30
                State: <Active Int>
                Age: 1d 1:36:33
                Task: PIM.vpna
                Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP RT Background
                AS path: I
```

Step 5. Type 5 routes that are installed in the `<routing-instance-name>.mvpn.0` table are picked up by BGP and advertised to remote PE routers.

Example: In Figure 1, PE1 advertises the following Type 5 route to remote PE routers.

```
user@PE1> show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn.0 | find 5:
* 5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
 BGP group int type Internal
     Route Distinguisher: 10.1.1.1:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:10:1
```

Step 6. The receiver PE that has both a Type 5 and Type 6 route for (C-*, C-G) is now ready to originate a Type 7 route.

Example: In Figure 1, PE3 has the following Type 5, 6, and 7 routes in the `vpna.mvpn.0` table. The Type 6 route is installed by C-PIM in Step 2. The Type 5 route is learned via BGP in Step 5. The Type 7 route is originated by the `MVPN` module in response to having both Type 5 and Type 6 routes for the same (C-*, C-G). The `RT` of the Type 7 route is the same as the `RT` of the Type 6 route due to the fact that both routes (IP address of the C-RP [`10.12.53.1`] and the address of the VPN multicast source [`192.168.1.2`]) are reachable via the same PE1 router). Therefore, `10.12.53.1` and `192.168.1.2` carry the same `rt-import` (`10.1.1.1:64`) community.

```
user@PE3> show route table vpna.mvpn.0 detail
5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.1.1.1
                Protocol next hop: 10.1.1.1
                Indirect next hop: 2 no-forward
                State: <Secondary Active Int Ext>
                Local AS: 65000 Peer AS: 65000
                Age: 1d 1:43:13         Metric2: 1
                Task: BGP_65000.10.1.1.1+55384
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: target:10:1
                Import Accepted
                Localpref: 100
                Router ID: 10.1.1.1
                Primary Routing Table bgp.mvpn.0
```

```
6:10.1.1.1:1:65000:32:10.12.53.1:32:224.1.1.1/240 (1 entry, 1 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4), Next hop index: 262144
                Next-hop reference count: 11
                State: <Active Int>
                Age: 1d 1:44:09
                Task: PIM.vpna
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: no-advertise target:10.1.1.1:64

7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
        *MVPN   Preference: 70
                Next hop type: Multicast (IPv4), Next hop index: 262144
                Next-hop reference count: 11
                State: <Active Int Ext>
                Age: 1d 1:44:09         Metric2: 1
                Task: mvpn global task
                Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP RT Background
                AS path: I
                Communities: target:10.1.1.1:64
```

Step 7. The Type 7 route installed in the VRF `mvpn` table is picked up by BGP and advertised to remote PE routers.

Example: In Figure 1, PE3 advertises the following Type 7 route.

```
user@PE3> show route advertising-protocol bgp 10.1.1.1 detail table vpna.mvpn.0 | find
7:10.1.1.1
* 7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
 BGP group int type Internal
     Route Distinguisher: 10.1.1.3:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:10.1.1.1:64
```

If the C-join is a source tree `join`, then the Type 7 route is originated immediately (without waiting for a Type 5 route).

Example: In Figure 1, PE2 originates the following Type 7 route in response to receiving a (`192.168.1.2, 232.1.1.1`) C-join.

```
user@PE2> show route table vpna.mvpn.0 detail | find 7:10.1.1.1
7:10.1.1.1:1:65000:32:192.168.1.2:32:232.1.1.1/240 (1 entry, 1 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4), Next hop index: 262146
                Next-hop reference count: 4
                State: <Active Int>
                Age: 2d 18:59:56
                Task: PIM.vpna
                Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP RT Background
                AS path: I
                Communities: target:10.1.1.1:64
```

Receiving C-multicast Routes

A sender PE router imports a Type 7 route if the route is carrying an `RT` that matches the locally originated `rt-import` community. All Type 7 routes must pass the `__vrf-mvpn-import-cmcast-<routing-instance-name>-internal__` policy in order to be installed in the `<routing-instance-name>.mvpn.0` table.

When a sender PE router receives a Type 7 route via BGP, this route is installed in the `<routing-instance-name>.mvpn.0` table. The BGP route is then translated back into a normal C-join inside the VRF table, and the C-join is installed in the receiver PE's local C-PIM database. A new C-join added to C-PIM database triggers C-PIM to originate a Type 6 or Type 7 route. The C-PIM on the sender PE creates its own version of same Type 7 route received via BGP.

Example: In Figure 1, PE1 contains the following entries for a Type 7 route in the `vpna.mvpn.0` table corresponding to a (`192.168.1.2, 224.1.1.1`) `join` message. There are two entries; one entry is installed by PIM and the other entry is installed by BGP. This example also shows the Type 7 route corresponding to the (`192.168.1.2, 232.1.1.1`) `join`.

```
user@PE1> show route table vpna.mvpn.0 detail | find 7:10.1.1.1
7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (2 entries, 2 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4)
                Next-hop reference count: 30
                State: <Active Int>
                Age: 1d 2:19:04
                Task: PIM.vpna
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: no-advertise target:10.1.1.1:64
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.1.1.3
                Protocol next hop: 10.1.1.3
                Indirect next hop: 2 no-forward
                State: <Secondary Int Ext>
                Inactive reason: Route Preference
                Local AS: 65000 Peer AS: 65000
                Age: 53:27      Metric2: 1
                Task: BGP_65000.10.1.1.3+179
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: target:10.1.1.1:64
                Import Accepted
                Localpref: 100
                Router ID: 10.1.1.3
                Primary Routing Table bgp.mvpn.0

7:10.1.1.1:1:65000:32:192.168.1.2:32:232.1.1.1/240 (2 entries, 2 announced)
        *PIM    Preference: 105
                Next hop type: Multicast (IPv4)
                Next-hop reference count: 30
                State: <Active Int>
                Age: 2d 19:21:17
                Task: PIM.vpna
                Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                AS path: I
                Communities: no-advertise target:10.1.1.1:64
         BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 4
                Source: 10.1.1.2
```

```
               Protocol next hop: 10.1.1.2
               Indirect next hop: 2 no-forward
               State: <Secondary Int Ext>
               Inactive reason: Route Preference
               Local AS: 65000 Peer AS: 65000
               Age: 53:27      Metric2: 1
               Task: BGP_65000.10.1.1.2+49165
               Announcement bits (2): 0-PIM.vpna 1-mvpn global task
               AS path: I
               Communities: target:10.1.1.1:64
               Import Accepted
               Localpref: 100
               Router ID: 10.1.1.2
               Primary Routing Table bgp.mvpn.0
```

Remote C-joins (Type 7 routes learned via BGP translated back to normal C-joins) are installed in the VRF C-PIM database on the sender PE router and are processed based on regular C-PIM procedures. This process completes the end-to-end C-multicast routing exchange.

Example: In Figure 1, PE1 installs the following entries in its C-PIM database.

```
user@PE1> show pim join extensive instance vpna
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 201
    Downstream neighbors:
        Interface: Pseudo-MVPN

Group: 232.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout:
    Downstream neighbors:
        Interface: Pseudo-MVPN

Instance: PIM.vpna Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Conclusion

NG MVPNs provide service providers a new way of offering multicast VPN service. The strength of NG MVPN comes from its architecture that brings together the multicast protocols used at the edge and the BGP-MPLS technology deployed in the core. In particular, the use of BGP for transferring multicast routes across geographic locations and P2MP LSPs for distributing multicast data bring VPN multicast service offering to the same reliable and scalable level as the VPN unicast service.

Acronyms

AD	autodiscovery
ASBR	autonomous system border router
ASM	any-source multicast
Bidir	bidirectional
BGP	Border Gateway Protocol
CE	customer edge
C-G	customer multicast group address
C-join	customer `join` message
C-multicast	customer multicast
C-PIM	customer PIM
C-RP	customer rendezvous point
C-RPT	customer RP Tree
C-S	customer multicast source address
C-SPT	customer Shortest Path Tree
GRE	generic routing encapsulation
IANA	Internet Assigned Numbers Authority
INET	internet
I-PMSI	inclusive PMSI
LSP	label switched path
MCAST	multicast
mLDP	multipoint Label Distribution Protocol
MP2MP	multipoint to multipoint
MPLS	Multiprotocol Label Switching
MSDP	Multicast Source Delivery Protocol
MVPN	multicast VPN
NG MVPN	next-generation multicast VPN
NLRI	network layer reachability information
OIL	outgoing interface list
OSPF	Open Shortest Path First
P2MP	point to multipoint
PE	provider edge
PHP	penultimate-hop popping
P-group	provider multicast group
P-join	provider `join` message
PIC	Physical Interface Card
PIM	Protocol Independent Multicast
PIM SM	Protocol Independent Multicast sparse mode
PIM SSM	PIM source-specific multicast
PMSI	provider multicast service interface
P-PIM	provider PIM

P-RP	provider rendezvous point
RD	route distinguisher
RIP	Routing Information Protocol
RP	rendezvous point
RSVP-TE	Resource Reservation Protocol traffic engineering
RT	route target
SA	source active
SAFI	subsequent address family identifier
S-PMSI	selective PMSI
VPN	virtual private network
VRF	VPN routing and forwarding
VPN-IPv4	8-byte RD : 4-byte IPv4 address

References

BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs – draft-ietf-l3vpn-2547bis-mcast-bgp
www.ietf.org/internet-drafts/draft-ietf-l3vpn-2547bis-mcast-bgp

BGP/MPLS IP Virtual Private Networks (VPNs) – RFC 4364
www.ietf.org/rfc/rfc4364.txt?number=4364

Border Gateway Protocol (BGP) Data Collection Standard Communities – (IANA-MVPN-Extend, per RFC 4360)
www.iana.org/assignments/bgp-extended-communities

Extensions to RSVP-TE for Point-to-Multipoint TE LSPs – RFC 4875
www.ietf.org/rfc/rfc4875.txt

IETF Layer 3 Virtual Private Networks (l3vpn) Working Group
www.ietf.org/html.charters/l3vpn-charter.html

Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution
http://tools.ietf.org/html/draft-ietf-l3vpn-mvpn-considerations-04

Multicast in MPLS/BGP IP VPNs – draft-ietf-l3vpn-2547bis-mcast
www.ietf.org/internet-drafts/draft-ietf-l3vpn-2547bis-mcast

Multicast in MPLS/BGP VPNs – draft-rosen-vpn-mcast
http://tools.ietf.org/html/draft-rosen-vpn-mcast

Multicast Source Discovery Protocol (MSDP)
www.ietf.org/rfc/rfc3618.txt

Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification – RFC 4601
www.ietf.org/rfc/rfc4601.txt?number=4601

Subsequent Address Family Identifiers (SAFI) – (IANA-SAFI-MVPN, per RFC 4760)
www.iana.org/assignments/safi-namespace

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER
(888.586.4737)
or 408.745.2000
Fax: 408.745.2100

APAC Headquarters
Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters
Juniper Networks Ireland
Airside Business Park
Swords, County Dublin,
Ireland
Phone: 35.31.8903.600
Fax: 35.31.8903.601

2000320-001-EN Aug 2009

Printed on recycled paper.