JUNIPER NETWORKS

Network Configuration Example

Configuring Protocol Independent Multicast Join Load Balancing

Release
12.1

Published: 2012-03-06

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Network Configuration Example Configuring Protocol Independent Multicast Join Load Balancing
Release 12.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction

This document describes how you can configure multipath routing for external and internal virtual private network (VPN) routes with unequal interior gateway protocol (IGP) metrics, and Protocol Independent Multicast (PIM) join load balancing on provider edge (PE) routers in a multicast VPN (MVPN) network.

PIM Join Load Balancing on Multipath MVPN Routes Overview

A multicast virtual private network (MVPN) is a technology to deploy the multicast service in an existing MPLS/BGP VPN.

The two main MVPN services are:

- Dual PIM MVPNs (also referred to as Draft-Rosen)
- Multiprotocol BGP-based MVPNs (also referred to as next-generation)

Next-generation MVPNs constitute the next evolution after the Draft-Rosen MVPN and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs. A Draft-Rosen MVPN uses Protocol Independent Multicast (PIM) for customer multicast (C-multicast) signaling, and a next-generation MVPN uses BGP for C-multicast signaling.

Multipath routing in an MVPN is applied to make data forwarding more robust against network failures and to minimize shared backup capacities when resilience against network failures is required.

By default, PIM join messages are sent toward a source based on the reverse path forwarding (RPF) routing table check. If there is more than one equal-cost path toward the source [S, G] or rendezvous point (RP) [*, G], then one upstream interface is used to send the join messages. The upstream path can be:

- A single active external BGP (EBGP) path when both EBGP and internal BGP (IBGP) paths are present.
- A single active IBGP path when there is no EBGP path present.

With the introduction of the multipath PIM join load-balancing feature, customer PIM (C-PIM) join messages are load-balanced in the following ways:

- In the case of a Draft-Rosen MVPN, unequal EBGP and IBGP paths are utilized.
- In the case of next-generation MVPN:
  - Available IBGP paths are utilized when no EBGP path is present.
  - Available EBGP paths are utilized when both EBGP and IBGP paths are present.

This feature is applicable to IPv4 C-PIM join messages over the Layer 3 MVPN service.

By default, a customer source (C-S) or a customer RP (C-RP) is considered remote if the active **rt_entry** is a secondary route and the primary route is present in a different

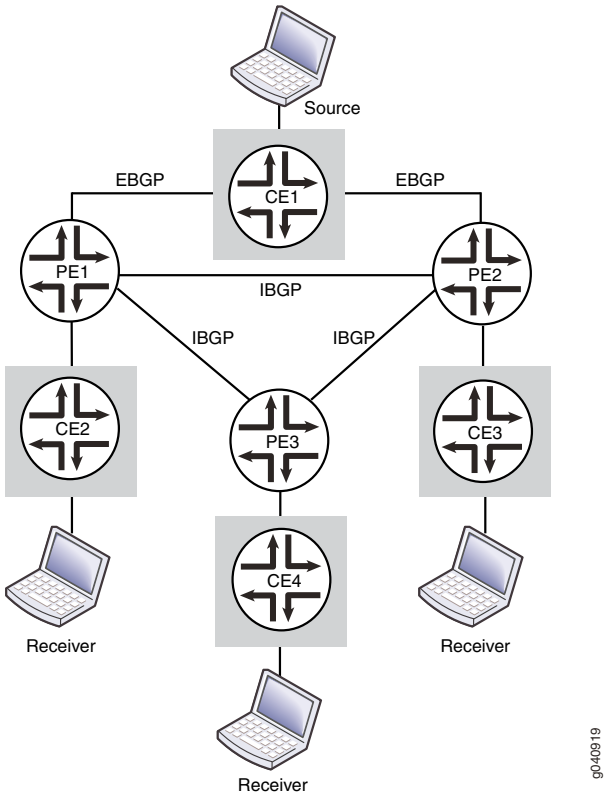
routing instance. Such determination is being done without taking into consideration the (C-*,G) or (C-S,G) state for which the check is being performed. The multipath PIM join load-balancing feature determines if a source (or RP) is remote by taking into account the associated (C-*,G) or (C-S,G) state.

When the provider network does not have provider edge (PE) routers with the multipath PIM join load-balancing feature enabled, hash-based join load balancing is used. Although the decision to configure this feature does not impact PIM or overall system performance, network performance can be affected temporarily, if the feature is not enabled.

With hash-based join load balancing, adding new PE routers to the candidate upstream toward the C-S or C-RP results in C-PIM join messages being redistributed to new upstream paths. If the number of join messages is large, network performance is impacted because of join messages being sent to the new RPF neighbor and prune messages being sent to the old RPF neighbor. In next-generation MVPN, this results in BGP C-multicast data messages being withdrawn from old upstream paths and advertised on new upstream paths, impacting network performance.

In Figure 1 on page 2, PE1 and PE2 are the upstream PE routers. Router PE1 learns route Source from EBGP and IBGP peers—the customer edge CE1 router and the PE2 router, respectively.

Figure 1: PIM Join Load Balancing



- If the PE routers run the Draft-Rosen MVPN, the PE1 router distributes C-PIM join messages between the EBGP path to the CE1 router and the IBGP path to the PE2

router. The join messages on the IBGP path are sent over a multicast tunnel interface through which the PE routers establish C-PIM adjacency with each other.

If a PE router loses one or all EBGP paths toward the source (or RP), the C-PIM join messages that were previously using the EBGP path are moved to a multicast tunnel interface, and the RPF neighbor on the multicast tunnel interface is selected based on a hash mechanism.

On discovering the first EBGP path toward the source (or RP), only new join messages get load-balanced across EBGP and IBGP paths, whereas the existing join messages on the multicast tunnel interface remain unaffected.

- If the PE routers run the next-generation MVPN, the PE1 router sends C-PIM join messages directly to the CE1 router over the EBGP path. There is no C-PIM adjacency between the PE1 and PE2 routers. Router PE3 distributes the C-PIM join messages between the two IBGP paths to PE1 and PE2. The Bytewise-XOR hash algorithm is used to send the C-multicast data according to Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*.

Because the multipath PIM join load-balancing feature in a Draft-Rosen MVPN utilizes unequal EBGP and IBGP paths to the destination, loops can be created when forwarding unicast packets to the destination. To avoid or break such loops:

- Traffic arriving from a core or master instance should not be forwarded back to the core facing interfaces.
- A single multicast tunnel interface should either be selected as the upstream interface or the downstream interface.
- An upstream or downstream multicast tunnel interface should point to a non-multicast tunnel interface.

As a result of the loop avoidance mechanism, join messages arriving from an EBGP path get load-balanced across EIBGP paths as expected, whereas join messages from an IBGP path are constrained to choose the EBGP path only.

In Figure 1 on page 2, if the CE2 host sends unicast data traffic to the CE1 host, the PE1 router could send the multicast flow to the PE2 router over the MPLS core due to traffic load balancing. A data forwarding loop is prevented by ensuring that PE2 does not forward traffic back on the MPLS core because of the load-balancing algorithm.

In the case of C-PIM join messages, assuming that both the CE2 host and the CE3 host are interested in receiving traffic from the source (S, G), and if both PE1 and PE2 choose each other as the RPF neighbor toward the source, then a multicast tree cannot be formed completely. This feature implements mechanisms to prevent such join loops in the multicast control plane in a Draft-Rosen MVPN scenario.

NOTE:

Disruption of multicast traffic or creation of join loops can occur, resulting in a multicast distribution tree (MDT) not being formed properly due to one of the following reasons:

- During a graceful Routing Engine switchover (GRES), the EIBGP path selection for C-PIM join messages can vary, because the upstream interface selection is performed again for the new Routing Engine based on the join messages it receives from the CE and PE neighbors. This can lead to disruption of multicast traffic depending on the number of join messages received and the load on the network at the time of the graceful restart. However, nonstop active routing (NSR) is not supported and has no impact on the multicast traffic in a Draft-Rosen MVPN scenario.
- Any PE router in the provider network is running another vendor's implementation that does not apply the same hashing algorithm implemented in this feature.
- The multipath PIM join load-balancing feature has not been configured properly.

Related Documentation

- Use Case for PIM Join Load Balancing on page 4
- Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN on page 5
- Example: Configuring PIM Join Load Balancing On Next-Generation Multicast VPN on page 13

Use Case for PIM Join Load Balancing

Large-scale service providers often have to meet the dynamic requirements of rapidly growing, worldwide virtual private network (VPN) markets. Service providers use the VPN infrastructure to deliver sophisticated services, such as video and voice conferencing, over highly secure, resilient networks. These services are usually loss-sensitive or delay-sensitive, and their data packets need to be delivered over a large-scale IP network in real time. The use of IP Multicast bandwidth-conserving technology has enabled service providers to exceed the most stringent service-level agreements (SLAs) and resiliency requirements.

IP multicast enables service providers to optimize network utilization while offering new revenue-generating value-added services, such as voice, video, and collaboration-based applications. IP multicast applications are becoming increasingly popular among enterprises, and as new applications start using multicast to deploy high-bandwidth and mission-critical services, it raises a new set of challenges for deploying IP multicast in the network.

IP multicast applications act as an essential communication protocol to effectively manage bandwidth and to reduce application server load by replicating the traffic on the network when the need arises. IP Protocol Independent Multicast (PIM) is the most

important IP multicast routing protocol that is used to communicate between the multicast routers, and is the industry standard for building multicast distribution trees of receiving hosts. The multipath PIM join load-balancing feature in a multicast VPN provides bandwidth efficiency by utilizing unequal paths toward a destination, improves scalability for large service providers, and minimizes service disruption.

The large-scale demands of service providers for IP access require Layer 3 VPN composite next hops along with external and internal BGP (EIBGP) VPN load balancing. The multipath PIM join load-balancing feature meets the large-scale requirements of enterprises by enabling **l3vpn-composite-nh** to be turned on along with EIBGP load balancing.

When the service provider network does not have the multipath PIM join load-balancing feature enabled on the provider edge (PE) routers, a hash-based algorithm is used to determine the best route to transmit multicast datagrams throughout the network. With hash-based join load balancing, adding new PE routers to the candidate upstream toward the destination results in PIM join messages being redistributed to new upstream paths. If the number of join messages is large, network performance is impacted because join messages are being sent to the new reverse path forwarding (RPF) neighbor and prune messages are being sent to the old RPF neighbor. In next-generation multicast virtual private network (MVPN), this results in multicast data messages being withdrawn from old upstream paths and advertised on new upstream paths, impacting network performance.

Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN

This example shows how to configure multipath routing for external and internal virtual private network (VPN) routes with unequal interior gateway protocol (IGP) metrics, and Protocol Independent Multicast (PIM) join load balancing on provider edge (PE) routers running Draft-Rosen multicast VPN (MVPN). This feature allows customer PIM (C-PIM) join messages to be load-balanced across external and internal BGP (EIBGP) upstream paths when the PE router has both external BGP (EBGP) and internal BGP (IBGP) paths toward the source or rendezvous point (RP).

Requirements

This example requires the following hardware and software components:

- Three routers that can be a combination of M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers.
- Junos OS Release 12.1 or later running on all the devices.

Before you begin:

1. Configure the device interfaces.

2. Configure the following routing protocols on all PE routers:
   - OSPF
   - MPLS
   - LDP
   - PIM
   - BGP
3. Configure a multicast VPN.

Overview and Topology

Junos OS Release 12.1 and later support multipath configuration along with PIM join load balancing. This allows C-PIM join messages to be load-balanced across unequal EIBGP routes, if a PE router has EBGP and IBGP paths toward the source (or RP). In previous releases, only the active EBGP path was used to send the join messages. This feature is applicable to IPv4 C-PIM join messages.

During load balancing, if a PE router loses one or more EBGP paths toward the source (or RP), the C-PIM join messages that were previously using the EBGP path are moved to a multicast tunnel interface, and the reverse path forwarding (RPF) neighbor on the multicast tunnel interface is selected based on a hash mechanism.

On discovering the first EBGP path toward the source (or RP), only the new join messages get load-balanced across EIBGP paths, whereas the existing join messages on the multicast tunnel interface remain unaffected.

Though the primary goal for multipath PIM join load balancing is to utilize unequal EIBGP paths for multicast traffic, potential join loops can be avoided if a PE router chooses only the EBGP path when there are one or more join messages for different groups from a remote PE router. If the remote PE router's join message arrives after the PE router has already chosen IBGP as the upstream path, then the potential loops can be broken by changing the selected upstream path to EBGP.

NOTE: During a graceful Routing Engine switchover (GRES), the EIBGP path selection for C-PIM join messages can vary, because the upstream interface selection is performed again for the new Routing Engine based on the join messages it receives from the CE and PE neighbors. This can lead to disruption of multicast traffic depending on the number of join messages received and the load on the network at the time of the graceful restart. However, the nonstop active routing feature is not supported and has no impact on the multicast traffic in a Draft-Rosen MVPN scenario.

In this example, PE1 and PE2 are the upstream PE routers for which the multipath PIM join load-balancing feature is configured. Routers PE1 and PE2 have one EBGP path and one IBGP path each toward the source. The Source and Receiver attached to customer edge (CE) routers are Free BSD hosts.

On PE routers that have EIBGP paths toward the source (or RP), such as PE1 and PE2, PIM join load balancing is performed as follows:

1. The existing join-count-based load balancing is performed such that the algorithm first selects the least loaded C-PIM interface. If there is equal or no load on all the C-PIM interfaces, the join messages get distributed equally across the available upstream interfaces.

   In Figure 2 on page 9, if the PE1 router receives PIM join messages from the CE2 router, and if there is equal or no load on both the EBGP and IBGP paths toward the source, the join messages get load-balanced on the EIBGP paths.

2. If the selected least loaded interface is a multicast tunnel interface, then there can be a potential join loop if the downstream list of the customer join (C-join) message already contains the multicast tunnel interface. In such a case, the least loaded interface among EBGP paths is selected as the upstream interface for the C-join message.

   Assuming that the IBGP path is the least loaded, the PE1 router sends the join messages to PE2 using the IBGP path. If PIM join messages from the PE3 router arrive on PE1, then the downstream list of the C-join messages for PE3 already contains a multicast tunnel interface, which can lead to a potential join loop, because both the upstream and downstream interfaces are multicast tunnel interfaces. In this case, PE1 uses only the EBGP path to send the join messages.

3. If the selected least loaded interface is a multicast tunnel interface and the multicast tunnel interface is not present in the downstream list of the C-join messages, the loop prevention mechanism is not necessary. If any PE router has already advertised data multicast distribution tree (MDT) type, length, and values (TLVs), that PE router is selected as the upstream neighbor.

   When the PE1 router sends the join messages to PE2 using the least loaded IBGP path, and if PE3 sends its join messages to PE2, no join loop is created.

4. If no data MDT TLV corresponds to the C-join message, the least loaded neighbor on a multicast tunnel interface is selected as the upstream interface.

On PE routers that have only IBGP paths toward the source (or RP), such as PE3, PIM join load balancing is performed as follows:

1. The PE router only finds a multicast tunnel interface as the RPF interface, and load balancing is done across the C-PIM neighbors on a multicast tunnel interface.

   Router PE3 load-balances PIM join messages received from the CE4 router across the IBGP paths to the PE1 and PE2 routers.

2. If any PE router has already advertised data MDT TLVs corresponding to the C-join messages, that PE router is selected as the RPF neighbor.

For a particular C-multicast flow, at least one of the PE routers having EIBGP paths toward the source (or RP) must use only the EBGP path to avoid or break join loops. As a result of the loop avoidance mechanism, a PE router is constrained to choose among EIBGP paths when a multicast tunnel interface is already present in the downstream list.

In Figure 2 on page 9, assuming that the CE2 host is interested in receiving traffic from the Source and CE2 initiates multiple PIM join messages for different groups (Group 1 with group address 225.1.1.1, and Group 2 with group address 225.1.1.2), the join messages for both groups arrive on the PE1 router.

Router PE1 then equally distributes the join messages between the EIBGP paths toward the Source. Assuming that Group 1 join messages are sent to the CE1 router directly using the EBGP path, and Group 2 join messages are sent to the PE2 router using the IBGP path, PE1 and PE2 become the RPF neighbors for Group 1 and Group 2 join messages, respectively.

When the CE3 router initiates Group 1 and Group 2 PIM join messages, the join messages for both groups arrive on the PE2 router. Router PE2 then equally distributes the join messages between the EIBGP paths toward the Source. Since PE2 is the RPF neighbor for Group 2 join messages, it sends the Group 2 join messages directly to the CE1 router using the EBGP path. Group 1 join messages are sent to the PE1 router using the IBGP path.

However, if the CE4 router initiates multiple Group 1 and Group 2 PIM join messages, there is no control over how these join messages received on the PE3 router get distributed to reach the Source. The selection of the RPF neighbor by PE3 can affect PIM join load balancing on EIBGP paths.

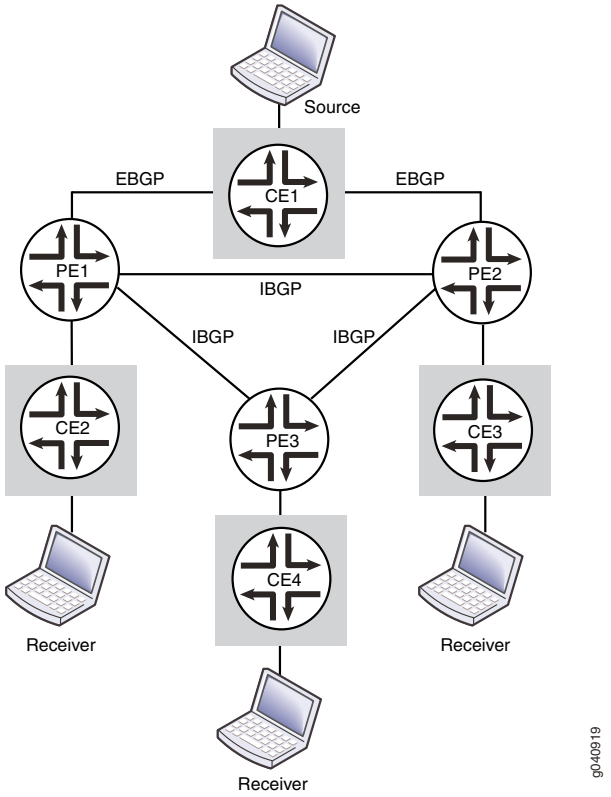
- If PE3 sends Group 1 join messages to PE1 and Group 2 join messages to PE2, there is no change in RPF neighbor. As a result, no join loops are created.
- If PE3 sends Group 1 join messages to PE2 and Group 2 join messages to PE1, there is a change in the RPF neighbor for the different groups resulting in the creation of join loops. To avoid potential join loops, PE1 and PE2 do not consider IBGP paths to send the join messages received from the PE3 router. Instead, the join messages are sent directly to the CE1 router using only the EBGP path.

The loop avoidance mechanism in a Draft-Rosen MVPN has the following limitations:

- Because the timing of arrival of join messages on remote PE routers determines the distribution of join messages, the distribution could be sub-optimal in terms of join count.
- Because join loops cannot be avoided and can occur due to the timing of join messages, the subsequent RPF interface change leads to loss of multicast traffic. This can be avoided by implementing the PIM make-before-break feature.

  The PIM make-before-break feature is an approach to detect and break C-PIM join loops in a Draft-Rosen MVPN. The C-PIM join messages are sent to the new RPF neighbor after establishing the PIM neighbor relationship, but before updating the related multicast forwarding entry. Though the upstream RPF neighbor would have updated its multicast forwarding entry and started sending the multicast traffic downstream, the downstream router does not forward the multicast traffic (because of RPF check failure) until the multicast forwarding entry is updated with the new RPF neighbor. This helps to ensure that the multicast traffic is available on the new path before switching the RPF interface of the multicast forwarding entry.

Figure 2: PIM Join Load Balancing on Draft-Rosen MVPN



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

PE1

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-5/0/4.0
set routing-instances vpn1 interface ge-5/2/0.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 1:1
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 44.44.44.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 44.44.44.2 peer-as 3
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 11.11.11.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 11.11.11.2 peer-as 4
set routing-instances vpn1 protocols pim vpn-group-address 224.1.1.1
set routing-instances vpn1 protocols pim rp static address 10.255.8.168
```

```
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
```

PE2

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-2/0/3.0
set routing-instances vpn1 interface ge-4/0/5.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 2:2
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 10.90.10.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 10.90.10.2 peer-as 45
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.50.10.2
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.50.10.1 peer-as 4
set routing-instances vpn1 protocols pim vpn-group-address 224.1.1.1
set routing-instances vpn1 protocols pim rp static address 10.255.8.168
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode. To configure the PE1 router:

NOTE: Repeat this procedure for every Juniper Networks router in the MVPN domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure a VPN routing and forwarding (VRF) instance.

```
[edit routing-instances vpn1]
user@PE1# set instance-type vrf
user@PE1# set interface ge-5/0/4.0
user@PE1# set interface ge-5/2/0.0
user@PE1# set interface lo0.1
user@PE1# set route-distinguisher 1:1
user@PE1# set vrf-target target:1:1
```

2. Enable protocol-independent load balancing for the VRF instance.

```
[edit routing-instances vpn1]
user@PE1# set routing-options multipath vpn-unequal-cost equal-external-internal
```

3. Configure BGP groups and neighbors to enable PE to CE routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set bgp export direct
user@PE1# set bgp group bgp type external
user@PE1# set bgp group bgp local-address 44.44.44.1
```

```
user@PE1# set bgp group bgp family inet unicast
user@PE1# set bgp group bgp neighbor 44.44.44.2 peer-as 3
user@PE1# set bgp group bgp1 type external
user@PE1# set bgp group bgp1 local-address 11.11.11.1
user@PE1# set bgp group bgp1 family inet unicast
user@PE1# set bgp group bgp1 neighbor 11.11.11.2 peer-as 4
```

4. Configure PIM to enable PE to CE multicast routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim vpn-group-address 224.1.1.1
user@PE1# set pim rp static address 10.255.8.168
```

5. Enable PIM on all network interfaces.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim interface all
```

6. Enable PIM join load balancing for the VRF instance.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim join-load-balance
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
routing-instances {
    vpn1 {
        instance-type vrf;
        interface ge-5/0/4.0;
        interface ge-5/2/0.0;
        interface lo0.1;
        route-distinguisher 1:1;
        vrf-target target:1:1;
        routing-options {
            multipath {
                vpn-unequal-cost equal-external-internal;
            }
        }
        protocols {
            bgp {
                export direct;
                group bgp {
                    type external;
                    local-address 44.44.44.1;
                    family inet {
                        unicast;
                    }
                    neighbor 44.44.44.2 {
                        peer-as 3;
                    }
                }
                group bgp1 {
                    type external;
                    local-address 11.11.11.1;
                    family inet {
```

```
            unicast;
          }
          neighbor 11.11.11.2 {
            peer-as 4;
          }
        }
      }
      pim {
        vpn-group-address 224.1.1.1;
        rp {
          static {
            address 10.255.8.168;
          }
        }
        interface all;
        join-load-balance;
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying PIM Join Load Balancing for Different Groups of Join Messages on page 12

Verifying PIM Join Load Balancing for Different Groups of Join Messages

Purpose Verify PIM join load balancing for the different groups of join messages received on the PE1 router.

Action From operational mode, run the **show pim join instance extensive** command.

```
user@PE1> show pim join instance extensive
 Instance: PIM.vpn1 Family: INET
 R = Rendezvous Point Tree, S = Sparse, W = Wildcard

 Group: 225.1.1.1
     Source: *
     RP: 10.255.8.168
     Flags: sparse,rptree,wildcard
     Upstream interface: ge-5/2/0.1
     Upstream neighbor: 10.10.10.2
     Upstream state: Join to RP
     Downstream neighbors:
         Interface: ge-5/0/4.0
             10.40.10.2 State: Join Flags: SRW Timeout: 207

 Group: 225.1.1.2
     Source: *
     RP: 10.255.8.168
     Flags: sparse,rptree,wildcard
     Upstream interface:  mt-5/0/10.32768
     Upstream neighbor: 19.19.19.19
```

```
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207

Group: 225.1.1.3
    Source: *
    RP: 10.255.8.168
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-5/2/0.1
    Upstream neighbor: 10.10.10.2
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207

Group: 225.1.1.4
    Source: *
    RP: 10.255.8.168
    Flags: sparse,rptree,wildcard
    Upstream interface:  mt-5/0/10.32768
    Upstream neighbor: 19.19.19.19
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207
```

Meaning The output shows how the PE1 router has load-balanced the C-PIM join messages for four different groups.

- For Group 1 (group address: 225.1.1.1) and Group 3 (group address: 225.1.1.3) join messages, the PE1 router has selected the EBGP path toward the CE1 router to send the join messages.
- For Group 2 (group address: 225.1.1.2) and Group 4 (group address: 225.1.1.4) join messages, the PE1 router has selected the IBGP path toward the PE2 router to send the join messages.

Related Documentation

- PIM Join Load Balancing on Multipath MVPN Routes Overview on page 1
- Example: Configuring PIM Join Load Balancing On Next-Generation Multicast VPN on page 13

Example: Configuring PIM Join Load Balancing On Next-Generation Multicast VPN

This example shows how to configure multipath routing for external and internal virtual private network (VPN) routes with unequal interior gateway protocol (IGP) metrics and Protocol Independent Multicast (PIM) join load balancing on provider edge (PE) routers running next-generation multicast VPN (MVPN). This feature allows customer PIM (C-PIM) join messages to be load-balanced across available internal BGP (IBGP) upstream paths when there is no external BGP (EBGP) path present, and across available

EBGP upstream paths when external and internal BGP (EIBGP) paths are present toward the source or rendezvous point (RP).

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of M Series, MX Series, or T Series routers.
- Junos OS Release 12.1 running on all the devices.

Before you begin:

1. Configure the device interfaces.
2. Configure the following routing protocols on all PE routers:
   - OSPF
   - MPLS
   - LDP
   - PIM
   - BGP
3. Configure a multicast VPN.

Overview and Topology

Junos OS Release 12.1 and later support multipath configuration along with PIM join load balancing. This allows C-PIM join messages to be load-balanced across all available IBGP paths when there are only IBGP paths present, and across all available upstream EBGP paths when EIBGP paths are present toward the source (or RP). Unlike Draft-Rosen MVPN, next-generation MVPN does not utilize unequal EIBGP paths to send C-PIM join messages. This feature is applicable to IPv4 C-PIM join messages.

By default, only one active IBGP path is used to send the C-PIM join messages for a PE router having only IBGP paths toward the source (or RP). When there are EIBGP upstream paths present, only one active EBGP path is used to send the join messages.

In a next-generation MVPN, C-PIM join messages are translated into (or encoded as) BGP customer multicast (C-multicast) MVPN routes and advertised with the BGP MCAST-VPN address family toward the sender PE routers. A PE router originates a C-multicast MVPN route in response to receiving a C-PIM join message through its PE router to customer edge (CE) router interface. The two types of C-multicast MVPN routes are:

- Shared tree join route (C-*, C-G)
  - Originated by receiver PE routers.
  - Originated when a PE router receives a shared tree C-PIM join message through its PE-CE router interface.
- Source tree join route (C-S, C-G)
  - Originated by receiver PE routers.
  - Originated when a PE router receives a source tree C-PIM join message (C-S, C-G), or originated by the PE router that already has a shared tree join route and receives a source active autodiscovery route.

  The upstream path in a next-generation MVPN is selected using the Bytewise-XOR hash algorithm as specified in Internet draft draft-ietf-l3vpn-2547bis-mcast, *Multicast in MPLS/BGP IP VPNs*. The hash algorithm is performed as follows:

  1. The PE routers in the candidate set are numbered from lower to higher IP address, starting from **0**.
  2. A bytewise exclusive-or of all the bytes is performed on the C-root (source) and the C-G (group) address.
  3. The result is taken modulo *n*, where *n* is the number of PE routers in the candidate set. The result is **N**.
  4. **N** represents the IP address of the upstream PE router as numbered in Step 1.

During load balancing, if a PE router with one or more upstream IBGP paths toward the source (or RP) discovers a new IBGP path toward the same source (or RP), the C-PIM join messages distributed among previously existing IBGP paths get redistributed due to the change in the candidate PE router set.

In this example, PE1, PE2, and PE3 are the PE routers that have the multipath PIM join load-balancing feature configured. Router PE1 has two EBGP paths and one IBGP upstream path, PE2 has one EBGP path and one IBGP upstream path, and PE3 has two IBGP upstream paths toward the Source. Router CE4 is the customer edge (CE) router attached to PE3. Source and Receiver are the Free BSD hosts.

On PE routers that have EIBGP paths toward the source (or RP), such as PE1 and PE2, PIM join load balancing is performed as follows:

1. The C-PIM join messages are sent using EBGP paths only. IBGP paths are not used to propagate the join messages.

In Figure 3 on page 17, the PE1 router distributes the join messages between the two EBGP paths to the CE1 router, and PE2 uses the EBGP path to CE1 to send the join messages.

2. If a PE router loses one or more EBGP paths toward the source (or RP), the RPF neighbor on the multicast tunnel interface is selected based on a hash mechanism.

   On discovering the first EBGP path, only new join messages get load-balanced across available EBGP paths, whereas the existing join messages on the multicast tunnel interface are not redistributed.

   If the EBGP path from the PE2 router to the CE1 router goes down, PE2 sends the join messages to PE1 using the IBGP path. When the EBGP path to CE1 is restored, only new join messages that arrive on PE2 use the restored EBGP path, whereas join messages already sent on the IBGP path are not redistributed.

On PE routers that have only IBGP paths toward the source (or RP), such as the PE3 router, PIM join load balancing is performed as follows:

1. The C-PIM join messages from CE routers get load-balanced only as BGP C-multicast data messages among IBGP paths.

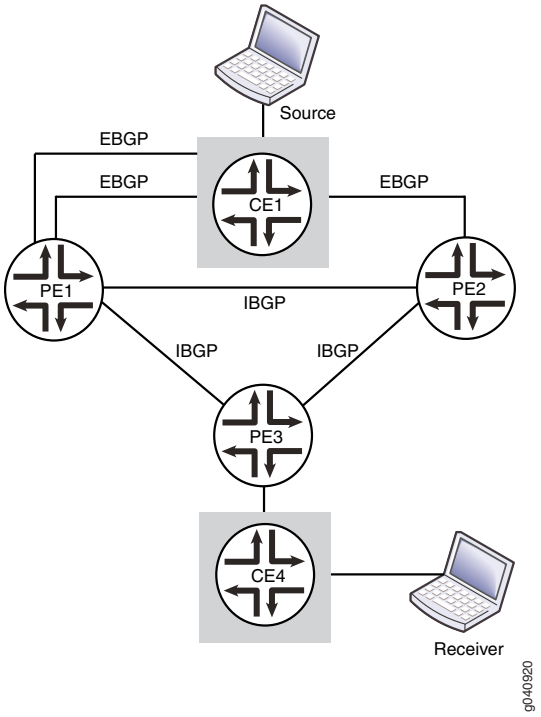
   In Figure 3 on page 17, assuming that the CE4 host is interested in receiving traffic from the Source, and CE4 initiates source join messages for different groups (Group 1 [C-S,C-G1] and Group 2 [C-S,C-G2]), the source join messages arrive on the PE3 router.

   Router PE3 then uses the Bytewise-XOR hash algorithm to select the upstream PE router to send the C-multicast data for each group. The algorithm first numbers the upstream PE routers from lower to higher IP address starting from **0**.

   Assuming that Router PE1 router is numbered **0** and Router PE2 is **1**, and the hash result for Group 1 and Group 2 join messages is **0** and **1**, respectively, the PE3 router selects PE1 as the upstream PE router to send Group 1 join messages, and PE2 as the upstream PE router to send the Group 2 join messages to the Source.

2. The shared join messages for different groups [C-*,C-G] are also treated in a similar way to reach the destination.

Figure 3: PIM Join Load Balancing on Next-Generation MVPN



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

PE1

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-3/0/1.0
set routing-instances vpn1 interface ge-3/3/2.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 1:1
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.40.10.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.40.10.2 peer-as 3
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 10.10.10.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
```

```
set routing-instances vpn1 protocols bgp group bgp1 neighbor 10.10.10.2 peer-as 3
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

PE2

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-1/0/9.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 2:2
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp local-address 10.50.10.2
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.50.10.1 peer-as 3
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

PE3

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-0/0/8.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 3:3
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 routing-options autonomous-system 1
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.80.10.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.80.10.2 peer-as 2
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode. To configure the PE1 router:

NOTE: Repeat this procedure for every Juniper Networks router in the MVPN domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure a VPN routing forwarding (VRF) routing instance.

```
[edit routing-instances vpn1]
user@PE1# set instance-type vrf
user@PE1# set interface ge-3/0/1.0
user@PE1# set interface ge-3/3/2.0
user@PE1# set interface lo0.1
user@PE1# set route-distinguisher 1:1
user@PE1# set provider-tunnel rsvp-te label-switched-path-template
  default-template
user@PE1# set vrf-target target:1:1
user@PE1# set vrf-table-label
```

2. Enable protocol-independent load balancing for the VRF instance.

```
[edit routing-instances vpn1]
user@PE1# set routing-options multipath vpn-unequal-cost equal-external-internal
```

3. Configure BGP groups and neighbors to enable PE to CE routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set bgp export direct
user@PE1# set bgp group bgp type external
user@PE1# set bgp group bgp local-address 10.40.10.1
user@PE1# set bgp group bgp family inet unicast
user@PE1# set bgp group bgp neighbor 10.40.10.2 peer-as 3
user@PE1# set bgp group bgp1 type external
user@PE1# set bgp group bgp1 local-address 10.10.10.1
user@PE1# set bgp group bgp1 family inet unicast
user@PE1# set bgp group bgp1 neighbor 10.10.10.2 peer-as 3
```

4. Configure PIM to enable PE to CE multicast routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim rp static address 10.255.10.119
```

5. Enable PIM on all network interfaces.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim interface all
```

6. Enable PIM join load balancing for the VRF instance.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim join-load-balance
```

7. Configure the mode for C-PIM join messages to use rendezvous-point trees, and switch to the shortest-path tree after the source is known.

```
[edit routing-instances vpn1 protocols]
user@PE1# set mvpn mvpn-mode rpt-spt
```

8. Configure the VRF instance to use the Bytewise-XOR hash algorithm.

```
[edit routing-instances vpn1 protocols]
user@PE1# set mvpn mvpn-join-load-balance bytewise-xor-hash
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show routing-instances
routing-instances {
  vpn1 {
    instance-type vrf;
    interface ge-3/0/1.0;
    interface ge-3/3/2.0;
    interface lo0.1;
    route-distinguisher 1:1;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    vrf-target target:1:1;
    vrf-table-label;
    routing-options {
      multipath {
        vpn-unequal-cost equal-external-internal;
      }
    }
    protocols {
      bgp {
        export direct;
        group bgp {
          type external;
          local-address 10.40.10.1;
          family inet {
            unicast;
          }
          neighbor 10.40.10.2 {
            peer-as 3;
          }
        }
        group bgp1 {
          type external;
          local-address 10.10.10.1;
          family inet {
            unicast;
          }
          neighbor 10.10.10.2 {
            peer-as 3;
          }
```



```
        }
      }
      pim {
        rp {
          static {
            address 10.255.10.119;
          }
        }
        interface all;
        join-load-balance;
      }
      mvpn {
        mvpn-mode {
          rpt-spt;
        }
        mvpn-join-load-balance {
          bytewise-xor-hash;
        }
      }
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying MVPN C-Multicast Route Information for Different Groups of Join Messages on page 21

Verifying MVPN C-Multicast Route Information for Different Groups of Join Messages

Purpose Verify MVPN C-multicast route information for different groups of join messages received on the PE3 router.

Action From operational mode, run the **show mvpn c-multicast** command.

```
user@PE3> show mvpn c-multicast
MVPN instance:
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Ptnl                              St
  0.0.0.0/0:225.1.1.1/32        RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  4.4.4.2/32:225.1.1.1/32       RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  0.0.0.0/0:225.1.1.2/32        RSVP-TE P2MP:10.255.10.14, 47575,10.255.10.14
  4.4.4.2/32:225.1.1.2/32       RSVP-TE P2MP:10.255.10.14, 47575,10.255.10.14
```

Meaning The output shows how the PE3 router has load-balanced the C-multicast data for the different groups.

- For source join messages (S,G):
  - 4.4.4.2/32:225.1.1.1/32 (S,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 4.4.4.2/32:225.1.1.2/32 (S,G2) toward the PE2 router (10.255.10.14 is the loopback address of Router PE2).
- For shared join messages (*,G):
  - 0.0.0.0/0:225.1.1.1/32 (*,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 0.0.0.0/0:225.1.1.2/32 (*,G2) toward the PE2 router (10.255.10.14 is the loopback address of Router PE2).

Related Documentation

- PIM Join Load Balancing on Multipath MVPN Routes Overview on page 1
- Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN on page 5