



---

Junos<sup>®</sup> OS

# Connecting IPv6 Islands with IPv4 MPLS Feature Guide

Release  
12.1



---

Published: 2012-02-24

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://www.juniper.net)

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

*Junos® OS Connecting IPv6 Islands with IPv4 MPLS Feature Guide*

Release 12.1

Copyright © 2012, Juniper Networks, Inc.

All rights reserved.

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

<b>Part 1</b>	<b>Connecting IPv6 Islands with IPv4 MPLS</b>	
<b>Chapter 1</b>	<b>Connecting IPv6 Islands with IPv4 MPLS Concepts and Reference Materials</b>	<b>3</b>
	Connecting IPv6 Islands with IPv4 MPLS Overview	3
	Connecting IPv6 Islands with IPv4 MPLS System Requirements	5
	Connecting IPv6 Islands with IPv4 MPLS Terms and Acronyms	5
<b>Chapter 2</b>	<b>Connecting IPv6 Islands with IPv4 MPLS Configuration</b>	<b>7</b>
	Configuring IPv6 on the Customer and Core-Facing Interfaces	7
	Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel	7
	Enabling IPv6 Tunneling in MPLS	7
	Configuring Multiprotocol BGP to Carry IPv6 Traffic	8
<b>Chapter 3</b>	<b>Connecting IPv6 Islands with IPv4 MPLS Configuration Example</b>	<b>9</b>
	Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration	9
	Verifying Your Work	15
	Router CE1 Status	16
	Router PE1 Status	16
	Router PE2 Status	17
	Router CE2 Status	18
	For More Information	19
<b>Part 2</b>	<b>Index</b>	
	Index	23



# List of Figures

Part 1	Connecting IPv6 Islands with IPv4 MPLS	
Chapter 1	Connecting IPv6 Islands with IPv4 MPLS Concepts and Reference Materials .....	3
	Figure 1: Connecting IPv6 Islands over MPLS .....	4
Chapter 3	Connecting IPv6 Islands with IPv4 MPLS Configuration Example .....	9
	Figure 2: IPv6 over an MPLS Tunnel .....	9



## PART 1

# Connecting IPv6 Islands with IPv4 MPLS

- [Connecting IPv6 Islands with IPv4 MPLS Concepts and Reference Materials on page 3](#)
- [Connecting IPv6 Islands with IPv4 MPLS Configuration on page 7](#)
- [Connecting IPv6 Islands with IPv4 MPLS Configuration Example on page 9](#)





## CHAPTER 1

# Connecting IPv6 Islands with IPv4 MPLS Concepts and Reference Materials

This section contains the following topics:

- [Connecting IPv6 Islands with IPv4 MPLS Overview on page 3](#)
- [Connecting IPv6 Islands with IPv4 MPLS System Requirements on page 5](#)
- [Connecting IPv6 Islands with IPv4 MPLS Terms and Acronyms on page 5](#)

## Connecting IPv6 Islands with IPv4 MPLS Overview

Many service providers are looking for ways to provide new revenue-generating services to their customers. One such service is Internet Protocol version 6 (IPv6). Some enterprise customers are beginning to experiment with this new version of IP, but are reluctant to deploy it broadly. Interconnecting multiple sites that use IPv6 can be challenging. Also, most service providers would prefer to carry this traffic without making major modifications to their core network.

A technique available in Junos OS Release 5.4 allows you to connect IPv6 sites over an IPv4 Multiprotocol Label Switching (MPLS) enabled backbone. Juniper Networks supports the Multiprotocol Border Gateway Protocol (MP-BGP) over IPv4 approach detailed in RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*. With this technique, IPv6 islands are connected to each other across an IPv4 backbone enabled with MPLS label stacking while MP-BGP is used to announce the IPv6 routes across these MPLS tunnels. This feature can be implemented with label-switched paths (LSPs) using the Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP).

IPv6 packets are carried over an IPv4 MPLS tunnel. To enable this service, you need to deploy provider edge (PE) routers that can run IPv4, MPLS, and BGP toward the core and IPv6 toward the edge. Since only the PE routers need to run a dual stack of IPv4 and IPv6, the other provider (P) core routers do not need to be upgraded. As a result, this MPLS tunneling technique allows for interoperability with routers from other vendors.

Because of this flexible method of implementation, it is now more attractive for providers to carry IPv6 traffic over their existing core networks and for customers to roll out IPv6 to more sites.

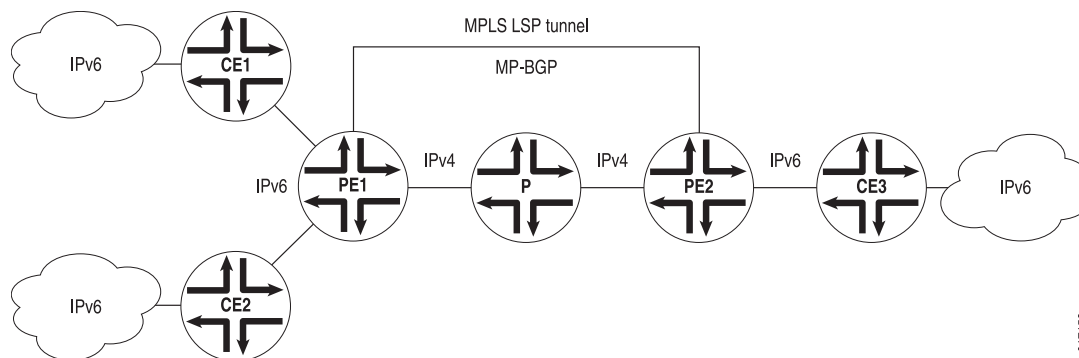
In [Figure 1 on page 4](#), PE1 and PE2 are dual-stack Border Gateway Protocol (DS-BGP) routers. They implement IPv4 and IPv6 stacks simultaneously. The IPv6 clouds are separate islands that are connected to PE routers through a customer edge (CE) router.

This example shows how to enable IPv6 connectivity between the various IPv6 islands, not how to create an IPv6 VPN service. One of the IPv6 islands can be the global IPv6 Internet.

The connection between the CE and PE routers can use any network layer protocol that carries IPv6 traffic. The provider router can exchange information with the customer routers using IPv6-enabled routing protocols, such as RIPng or MP-BGP, or static routes. The PE routers use IPv6 on the CE-facing interfaces, but use IPv4, BGP, and MPLS to connect to the core.

You must configure appropriate export policies on the PE router to share route information between IBGP and EBGP, and between BGP and other protocols.

**Figure 1: Connecting IPv6 Islands over MPLS**



Because MP-BGP requires that a BGP next hop use the same address family as the Network Layer Reachability Information (NLRI), the IPv4 address needs to be embedded in an IPv6 format. Such IPv4-mapped IPv6 addresses are defined in RFC 3513, *IP Version 6 Addressing Architecture*. After the PE routers learn the IPv6 routes from their directly attached CE neighbors, each PE router uses its own IPv4 address as the next hop for the IPv6 routes that are advertised in the BGP session.

The two PE routers establish an MP-BGP session with each other using IPv4 addresses. In the session, the routers exchange IPv6 routes with an IPv6 address family identifier (AFI) value of 2 and a subsequent AFI (SAFI) label with a value of 4. Labels with a value of 2 are explicit null labels for IPv6, as defined in RFC 3032. Before sending IPv6 traffic across the IPv4 MPLS tunnel, the PE attaches the two labels. The inner label is 2 (another value if the advertising PE router is not a Juniper Networks router) and the outer label is the LSP label.

A PE router must have MPLS LSPs pointing to the other peer PE router's IPv4 address. The LSPs are signaled across the IPv4 control plane using either LDP or RSVP. These LSPs resolve the next-hop addresses of the IPv6 routes learned through MP-BGP. The next hops are actually IPv4-mapped IPv6 addresses, whereas the LSPs are associated with IPv4 addresses. Because of this mapping technique, the IPv6 traffic can travel over the IPv4 LSP transparently.

In [Figure 1 on page 4](#), PE1 receives an IPv6 packet from CE1 and performs a lookup in the IPv6 forwarding table. If the destination matches a prefix that was learned from CE2, no labels are necessary and the IPv6 packet is sent to CE2. If the destination matches a prefix that was learned from PE2, then PE1 places two labels on the packet and sends it to P. The inner label is 2 and the outer label is the LSP label needed to reach PE2. Since P is the penultimate-hop router for the LSP to PE2 and the received packet has more than one label, Router P pops the outer label and sends the packet to PE2. When PE2 receives the packet, it has a single label with a value of 2. PE2 strips off the label and treats the remaining packet as an IPv6 packet (since 2 is the IPv6 explicit null label) and performs a lookup in the IPv6 forwarding table.

Although the MP-BGP over IPv4 approach can operate using a single level of labels, there is an advantage in using two labels. The penultimate-hop router for the MPLS LSP (P in this case) can pop the outer label and send the packet with the inner label as an MPLS packet. When the packet arrives at egress Router PE2, the second label using the explicit null value is popped and the remaining IPv6 packet is sent to the directly connected IPv6 network. Thus, the benefit of using two labels is that penultimate hop-popping (PHP) routers do not require IPv6 capabilities or the need for an upgrade.

Interconnecting IPv6 islands over an IPv4 MPLS tunnel requires:

- An exchange of IPv6 reachability information between DS-BGP routers. Using MP-BGP, the DS-BGP (PE) routers exchange IPv6 reachability information over the IPv4 core network with other similarly enabled DS-BGP PE peers. As a result, the egress DS-BGP (PE) router announces itself as the BGP next hop.
- IPv6 packets are tunneled from the ingress DS-BGP router to the egress DS-BGP router by means of MPLS. The ingress DS-BGP router tunnels an IPv6 packet over the IPv4 network toward the egress DS-BGP router identified as the BGP next hop for the packet's destination IPv6 address.

---

## Connecting IPv6 Islands with IPv4 MPLS System Requirements

---

To carry IPv6 traffic over IPv4 MPLS tunnels, your system must meet these minimum requirements:

- Junos OS Release 8.2 or later for MX Series routers
- Junos OS Release 7.2 or later for J Series Services Routers
- Junos OS Release 5.4 or later for M Series and T Series routers
- Two Juniper Networks J Series, M Series, MX Series, or T Series routers to act as the DS-BGP ingress and egress devices

---

## Connecting IPv6 Islands with IPv4 MPLS Terms and Acronyms

---

### D

#### **dual-stack BGP (DS-BGP)**

A router that processes IPv4 and IPv6 packets in a BGP-connected network.

## M

<b>Multiprotocol BGP (MP-BGP)</b>	A router enabled for MP-BGP processes packets from a variety of protocols in a BGP-connected network.
---------------------------------------	---

## S

<b>Subsequent Address Family Identifier (SAFI)</b>	A field in Multiprotocol BGP messages that identifies MPLS network layer reachability information (NLRI). Common values include 1 (unicast), 2 (multicast), and 4 (MPLS label).
--	---

## CHAPTER 2

# Connecting IPv6 Islands with IPv4 MPLS Configuration

To enable IPv6 to be carried over an IPv4 MPLS tunnel, perform the following tasks:

- [Configuring IPv6 on the Customer and Core-Facing Interfaces on page 7](#)
- [Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel on page 7](#)
- [Enabling IPv6 Tunneling in MPLS on page 7](#)
- [Configuring Multiprotocol BGP to Carry IPv6 Traffic on page 8](#)

### Configuring IPv6 on the Customer and Core-Facing Interfaces

---

Configure **family inet6** on all the CE-facing interfaces and on all the core-facing interfaces running MPLS. This enables the router to process any IPv6 packets it receives on these interfaces. You should not see any regular IPv6 traffic arrive on these interfaces, but you will receive MPLS packets tagged with label 2. Even though label 2 MPLS packets are sent in IPv4, these packets are treated as native IPv6 packets.

```
[edit]
interfaces {
  interface-name {
    unit unit-number {
      family inet6 {
        address inet6-address;
      }
    }
  }
}
```

### Configuring MPLS and RSVP from PE Router to PE Router to Create a Tunnel

---

This guide assumes you already have experience configuring MPLS and RSVP. For more information about these topics, see the *Junos MPLS Applications Configuration Guide*.

### Enabling IPv6 Tunneling in MPLS

---

Enter the **ipv6-tunneling** option on your PE routers at the **[edit protocols mpls]** hierarchy level:

```
[edit]
protocols {
  mpls {
    ipv6-tunneling;
  }
}
```

---

## Configuring Multiprotocol BGP to Carry IPv6 Traffic

You can specify the **family inet6** statement on a per-neighbor, per-group, or global basis. The statement allows BGP to carry IPv6 traffic.

At the appropriate global, group, or neighbor hierarchy level in BGP (shown below), configure the **family inet6** statement with the **labeled-unicast** parameter and the **explicit-null** option. These additional parameters enable the IPv4 MPLS label to be removed at the destination PE router. The remaining IPv6 packet without a label can then be forwarded to the connected IPv6 network.

```
[edit protocols bgp] OR
[edit protocols bgp group group-name] OR
[edit protocols bgp group group-name neighbor neighbor-name]
family inet6 {
  labeled-unicast {
    explicit-null;
  }
}
```

# Connecting IPv6 Islands with IPv4 MPLS Configuration Example

To enable IPv6 to be carried over an IPv4 MPLS tunnel, perform the following tasks:

- [Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration on page 9](#)
- [For More Information on page 19](#)

## Example: Connecting IPv6 Islands over an MPLS Tunnel Configuration

Figure 2: IPv6 over an MPLS Tunnel

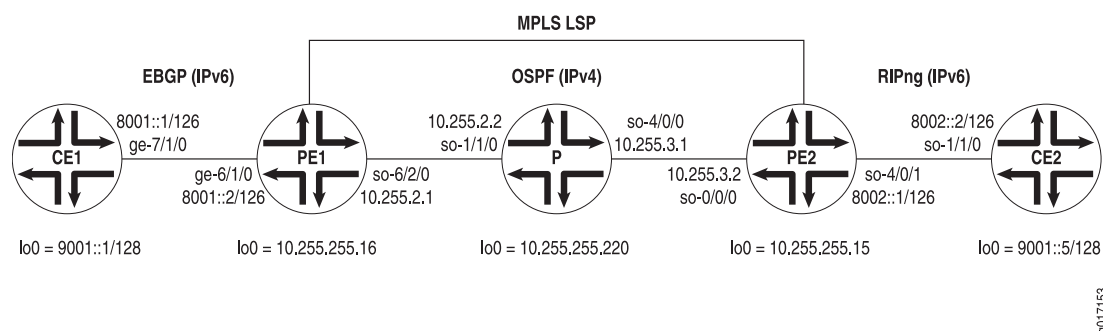


Figure 2 on page 9 shows a standard CE-PE-P-PE-CE MPLS-style network. CE1 and CE2 are the end customer CE routers using IPv6; PE1 and PE2 are the provider edge routers; and P is a provider core router. The IPv4 MPLS tunnel travels between PE1 and PE2, connecting IPv6 sites CE1 and CE2.

Since the CE-to-PE configuration can use a variety of routing protocols, this example requires that you use EBGP between CE1 and PE1 and RIPng between PE2 and CE2. You must establish policies on PE2 to import and export routes between BGP and RIPng.

To start the configuration, set up the IPv6 connection between CE1 and PE1. In your BGP routing policy, you must advertise the IPv6 loopback address of the CE1 router address to the PE1 router.

```
Router CE1 [edit]
interfaces {
  ge-7/1/0 {
    unit 0 {
      family inet6 {
```

```

        address 8001::1/126;
    }
}
lo0 {
    unit 0 {
        family inet6 {
            address 9001::1/128;
        }
    }
}
routing-options {
    autonomous-system 200;
}
protocols {
    bgp {
        group to_PE1 {
            type external;
            local-address 8001::1;
            family inet6 {
                unicast;
            }
            export policy1;
            peer-as 100;
            neighbor 8001::2;
        }
    }
    policy-options {
        policy-statement policy1 {
            term 1 {
                from {
                    family inet6;
                    route-filter 9001::1/128 exact;
                }
                then accept;
            }
            term 2 {
                then reject;
            }
        }
    }
}

```

Once you move to PE1, your tasks become more complex. You must complete the IPv6 EBGp connection to CE1 and build the first part of the MPLS tunnel. You must set the **inet**, **inet6**, and **mpls** families on the core-facing interface, configure an **inet6** address for the CE-facing interface attached to CE1, and ensure the IPv4 loopback address is advertised in OSPF, since this is the MPLS LSP target for PE2. You must also add the **ipv6-tunneling** parameter in MPLS, include the **labeled-unicast** and **explicit-null** options at the **[edit protocols bgp family inet6]** hierarchy level, and create an external BGP group pointing to CE1 and an internal group pointing to PE2.

Router PE1 [edit]  
 interfaces {



```
ge-6/1/0 {
  unit 0 {
    family inet6 {
      address 8001::2/126;
    }
  }
}
so-6/2/0 {
  unit 0 {
    family inet {
      address 10.255.2.1/24;
    }
    family inet6;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.255.255.16/32;
    }
  }
}
}
routing-options {
  autonomous-system 100;
}
protocols {
  rsvp {
    interface so-6/2/0.0;
  }
  mpls {
    ipv6-tunneling;
    label-switched-path to_PE2 {
      to 10.255.255.15;
    }
    interface so-6/2/0.0;
  }
  bgp {
    group to_PE2 {
      type internal;
      local-address 10.255.255.16;
      family inet6 {
        labeled-unicast {
          explicit-null;
        }
      }
      neighbor 10.255.255.15;
    }
    group to_CE1 {
      local-address 8001::2;
      family inet6 {
        unicast;
      }
      peer-as 200;
      neighbor 8001::1;
    }
  }
}
```

```
    }  
  }  
  ospf {  
    traffic-engineering;  
    area 0.0.0.0 {  
      interface so-6/2/0.0;  
      interface lo0.0 {  
        passive;  
      }  
    }  
  }  
}
```

On Router P, connect the MPLS tunnel between PE1 and PE2. Enable RSVP, MPLS, and IPv4 connectivity on the interfaces and ensure that IP connectivity is available through the routing protocol (in this case, OSPF).

```
Router P [edit]  
interfaces {  
  so-1/1/0 {  
    unit 0 {  
      family inet {  
        address 10.255.2.2/24;  
      }  
      family mpls;  
    }  
  }  
  so-4/0/0 {  
    unit 0 {  
      family inet {  
        address 10.255.3.1/24;  
      }  
      family mpls;  
    }  
  }  
  lo0 {  
    unit 0 {  
      family inet {  
        address 10.255.255.220/32;  
      }  
    }  
  }  
}  
routing-options {  
  autonomous-system 100;  
}  
protocols {  
  rsvp {  
    interface so-1/1/0.0;  
    interface so-4/0/0.0;  
  }  
  mpls {  
    interface so-1/1/0.0;  
    interface so-4/0/0.0;  
  }  
  ospf {
```

```

traffic-engineering;
area 0.0.0.0 {
    interface so-1/1/0.0;
    interface so-4/0/0.0;
    interface lo0.0 {
        passive;
    }
}
}
}

```

At PE2, you must complete a mirror image of the MPLS tunnel configuration started at PE1 and configure a RIPng connection to CE2. Set the **inet**, **inet6**, and **mpls** families on the core-facing interface, configure an **inet6** address for the CE facing interface attached to CE2, and ensure the IPv4 loopback address is advertised in OSPF, since this is the MPLS LSP target for PE1. You must also add the **ipv6-tunneling** parameter in MPLS and include the **labeled-unicast** and **explicit-null** options at the **[edit protocols bgp family inet6]** hierarchy level. Finally, create and apply policies that export BGP routes into RIPng and import RIPng routes to BGP.

```

Router PE2 [edit]
interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.255.3.2/24;
            }
            family inet6;
            family mpls;
        }
    }
    so-4/0/1 {
        unit 0 {
            family inet6 {
                address 8002::1/126;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.255.15/32;
            }
        }
    }
}
routing-options {
    autonomous-system 100;
}
protocols {
    rsvp {
        interface so-0/0/0.0;
    }
    mpls {
        ipv6-tunneling;
    }
}

```

```
label-switched-path to_PE1 {
  to 10.255.255.16;
}
interface so-0/0/0.0;
}
bgp {
  group to_PE1 {
    type internal;
    local-address 10.255.255.15;
    family inet6 {
      labeled-unicast {
        explicit-null;
      }
    }
    export red-export;
    neighbor 10.255.255.16;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
}
ripng {
  group to_CE2 {
    export red-import;
    neighbor so-4/0/1.0;
  }
}
}
policy-options {
  policy-statement red-export {
    term 1 {
      from protocol ripng;
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement red-import {
    from protocol bgp;
    then accept;
  }
}
```

Finally, on Router CE2, configure IPv6 addresses on the SONET/SDH and loopback interfaces, enable RIPng, and create and apply a policy for RIPng that permits the IPv6 loopback address to be exported to Router PE2. Once these tasks are accomplished, your IPv6 connection to Router CE1 should be ready for use.

**Router CE2**    [\[edit\]](#)

```

interfaces {
  so-1/1/0 {
    unit 0 {
      family inet6 {
        address 8002::2/126;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet6 {
        address 9001::5/128;
      }
    }
  }
}
routing-options {
  autonomous-system 300;
}
protocols {
  ripng {
    group to_PE2 {
      export policy1;
      neighbor so-1/1/0.0;
    }
  }
}
policy-options {
  policy-statement policy1 {
    term 1 {
      from {
        family inet6;
        route-filter 9001::5/128 exact;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

## Verifying Your Work

To verify that IPv6 traffic is being transported over the IPv4 MPLS tunnel, use the following commands:

- **ping**
- **show bgp summary**
- **show route protocol**
- **show route advertising-protocol**
- **show route receive-protocol**

- **show route table**
- **show route table (inet6.0 | inet6.3)**
- **show interfaces terse**

The following sections show the output of these commands used with the configuration example:

- [Router CE1 Status on page 16](#)
- [Router PE1 Status on page 16](#)
- [Router PE2 Status on page 17](#)
- [Router CE2 Status on page 18](#)

### Router CE1 Status

```
user@CE1> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet6.0         1          1          0           0         0         0         0
Peer           AS        InPkt    OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
8001::2        100         58        56         0         0        26:25 Establ
  inet6.0: 1/1/0

user@CE1> show route protocol bgp
inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1 hidden)
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
inet6.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
9001::5/128    *[BGP/170] 00:04:18, localpref 100
                AS path: 100 I
                > to 8001::2 via ge-7/1/0.0

user@CE1> ping 9001::5 source 9001::1
PING6(56=40+8+8 bytes) 9001::1 --> 9001::5
16 bytes from 9001::5, icmp_seq=0 hlim=62 time=0.945 ms
16 bytes from 9001::5, icmp_seq=1 hlim=62 time=0.831 ms
^C
--- 9001::5 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.831/0.887/0.945 ms
```

### Router PE1 Status

```
user@PE1> show bgp summary
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State   Pending
inet6.0         2          2          0           0         0         0         0
Peer           AS        InPkt    OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
8001::1        200         56        61         0         0        27:18 Establ
  inet6.0: 1/1/0
10.255.255.15  100         13        14         0         1         5:28 Establ
  inet6.0: 1/1/0

user@PE1> show route advertising-protocol bgp 10.255.255.15 detail
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
```

```

9001::1/128 (1 entry, 1 announced)
  BGP group to_PE2 type Internal
    Route Label: 2
    Nexthop: Self
    Localpref: 100
    AS path: 200 I
    Communities:

user@PE1> show route 9001::5
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
9001::5/128    *[BGP/170] 00:05:48, MED 2, localpref 100, from 10.255.255.15
  AS path: I
    > via so-6/2/0.0, label-switched-path to_PE2

user@PE1> show route table inet6.0
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
8001::1/126    *[Direct/0] 00:29:01
  > via ge-6/1/0.0
8001::2/128    *[Local/0] 00:29:01
  Local via ge-6/1/0.0
9001::1/128    *[BGP/170] 00:28:46, localpref 100
  AS path: 200 I
    > to 8001::1 via ge-6/1/0.0
9001::2/128    *[Direct/0] 00:29:01
  > via lo0.0
9001::5/128    *[BGP/170] 00:06:56, MED 2, localpref 100, from 10.255.255.15
  AS path: I
    > via so-6/2/0.0, label-switched-path to_PE2
fe80::/64      *[Direct/0] 00:29:01
  > via ge-6/1/0.0
fe80::280:42ff:fe10:d30c/128
  *[Direct/0] 00:29:01
  > via lo0.0
fe80::290:69ff:fe0f:1633/128
  *[Local/0] 00:29:01
  Local via ge-6/1/0.0

user@PE1> show route table inet6.3
inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
::ffff:10.255.255.15/128 *[RSVP/7] 00:06:37, metric 2, metric2 0
  > via so-6/2/0.0, label-switched-path to_PE2

```

### Router PE2 Status

```

user@PE2> show bgp summary
Groups: 1 Peers: 1 Down peers: 0
Table          Tot Paths  Act Paths Suppressed  History Damp State  Pending
inet6.0                1          1          0          0          0          0
Peer            AS      InPkt    OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Damped...
10.255.255.16    100        18        20         0         0          8:06 Estab1
inet6.0: 1/1/0

user@PE2> show interfaces terse so-4/0/1
Interface      Admin Link Proto Local                               Remote
so-4/0/1       up    up
so-4/0/1.0     up    up   inet 100.1.4.1/24

```

```

inet6 8002::1/126
fe80::280:42ff:fe10:d312/64

user@PE2> show route receive-protocol bgp 10.255.255.16 detail

inet.0: 18 destinations, 19 routes (17 active, 0 holddown, 1 hidden)
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
9001::1/128 (1 entry, 1 announced)
  Route Label: 2
  Nexthop: ::ffff:10.255.255.16
  Localpref: 100
  AS path: 200 I

inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
user@PE2> show route advertising-protocol ripng fe80::280:42ff:fe10:d312 detail
inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
9001::1/128 (1 entry, 1 announced)
  *BGP Preference: 170/-101
  Source: 10.255.255.16
  Next hop: via so-0/0/0.0, weight 1, selected
  Label-switched-path to_PE1
  Label operation: Push 2, Push 100015(top)
  Protocol next hop: ::ffff:10.255.255.16
  Push 2
  Indirect next hop: 8451440 50
  State: <Active Int Ext>
  Local AS: 100 Peer AS: 100
  Age: 2:27 Metric2: 2
  Task: BGP_100.10.255.255.16+179
  Announcement bits (3): 0-KRT 1-RIPng 3-Resolve inet6.0
  AS path: 200 I
  Route Label: 2

```

### Router CE2 Status

```

user@CE2> show ripng neighbor

```

Neighbor	State	Source Address	Dest Address	Send	Recv	In Met
so-1/1/0.0	Up	fe80::2a0:a5ff:fe12:34d9	ff02::9	yes	yes	1

```

user@CE2> show route protocol ripng

inet.0: 12 destinations, 12 routes (11 active, 0 holddown, 1 hidden)
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
inet6.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

9001::1/128          *[RIPng/100] 00:04:10, metric 2, tag 0
                    > to fe80::280:42ff:fe10:d312 via so-1/1/0.0

```



```
ff02::9/128      *[RIPng/100] 02:42:33, metric 1  
                  MultiRecv
```

---

## For More Information

For additional information about connecting IPv6 islands with IPv4 MPLS, see the following:

- *Junos MPLS Applications Configuration Guide*
- *Junos Routing Protocols Configuration Guide*
- RFC 3032, *MPLS Label Stack Encoding*
- RFC 3107, *Carrying Label Information in BGP-4*
- RFC 3513, *IP Version 6 Addressing Architecture*
- Internet draft draft-ietf-l3vpn-bgp-ipv6-07.txt, *BGP-MPLS IP VPN extension for IPv6 VPN* (expires January 2006)
- Internet draft draft-ooms-v6ops-bgp-tunnel-06.txt, *Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE)* (expires July 2006)



## PART 2

# Index

- [Index on page 23](#)



# Index

## I

### IPv6

#### tunneling over MPLS

configuration procedure.....	7, 9
example configuration.....	9
operational mode commands.....	15
overview.....	3
system requirements.....	5

## S

### system requirements

IPv6 tunneling over IPv4 MPLS.....	5
------------------------------------	---

