Cisco Nexus 9000 Series NX-OS SRv6 Configuration Guide, Release 10.1(x)

First Published: 2021-02-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

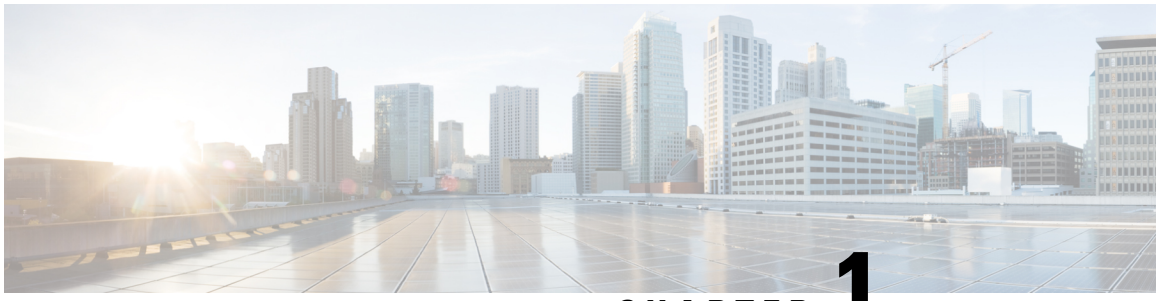
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000 Series NX-OS SRv6 Configuration Guide, Release 10.1(x)*.

- New and Changed Information, on page 1

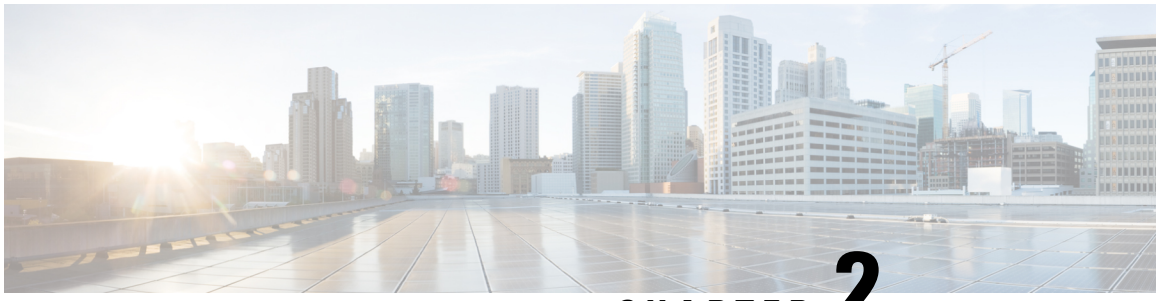
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS SRv6 Configuration Guide, Release 10.1(x)* and where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
No feature updates	First 10.1(x) release	10.1(1)	Not applicable

New and Changed Information



CHAPTER 2

Platform Support for SRv6 Features

This chapter defines platform support for features that are not supported across the entire suite of Cisco Platforms.

Platform Support for SRv6 Features

The following tables list the supported platforms for each feature and the release in which they were first introduced. See the Release Notes for details about the platforms supported in the initial product release.

SRv6

For more information about SRv6, see Configuring SRv6, on page 5.

Feature	Supported Platform(s) or Line Cards	First Supported Release
SRv6	Cisco Nexus 9300-GX platform switches	Cisco NX-OS Release 9.3(3)

SRv6 OAM

For more information about SRv6 OAM, see Configuring SRv6 OAM, on page 29.

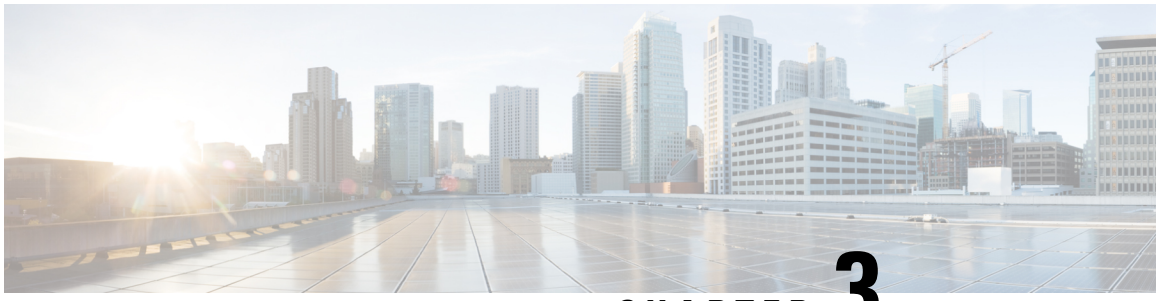
Feature	Supported Platform(s) or Line Cards	First Supported Release
SRv6 OAM	Cisco Nexus 9300-GX platform switches	Cisco NX-OS Release 9.3(3)

SRv6 Traffic Engineering

For more information about the SRv6 Traffic Engineering, see Configuring SRv6 Traffic Engineering, on page 21 .

Feature	Supported Platform(s) or Line Cards	First Supported Release
SRv6 Traffic Engineering	Cisco Nexus 9300-GX platform switches	Cisco NX-OS Release 9.3(5)

Platform Support for SRv6 Features



CHAPTER 3

Configuring SRv6

This chapter contains information on how to configure SRv6.

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

About Segment Routing Over IPv6

Segment Routing (SR) can be applied on both MPLS and IPv6 data planes. In a SR-MPLS enabled network, an MPLS label is used as the Segment Identifier (SID) and the source router chooses a path to the destination and encodes the path in the packet header as a stack of labels. In a Segment Routing over IPv6 (SRv6) network, the IPv6 address serves as the SID. The source router encodes the path to destination as an ordered list of segments (list of IPv6 addresses) in the IPv6 packet. To encode an ordered list of IPv6 addresses in an IPv6 packet, a new routing header which is an extension header is used. This new header for SRv6 is called Segment Routing Header (SRH). In an SRv6 enabled network, the active segment is indicated by the destination address of the packet, and the next segment is indicated by a pointer in the SRH.

SRv6 works on IPv6 data forwarding and is suitable for all data center deployments. SRv6 with SRH facilitates traffic engineering and path protection capabilities. Minus the SRH, SRv6 also supports traffic forwarding for multi-tenants with only the IPv6 packet header. In this case, the IPv6 destination address (128-bit) represents the reachability (locator) and the VPN function.

The forwarding methodology is such that if the destination address is within the locator prefix space is not in the SID table, it checks the standard routing table for a match.

Beginning Cisco NX-OS Release 9.3(3), Cisco Nexus 9300-GX series switches support SRv6 functionality as follows:

- IPv6
- processing of packets with SRHs at line rate
- BGP, OSPFv3, and IS-IS protocols
- L3VPN over SRv6 for both, IPv4 and IPv6 VPN prefixes
- global IPv4 and IPv6 (Internet) over SRv6

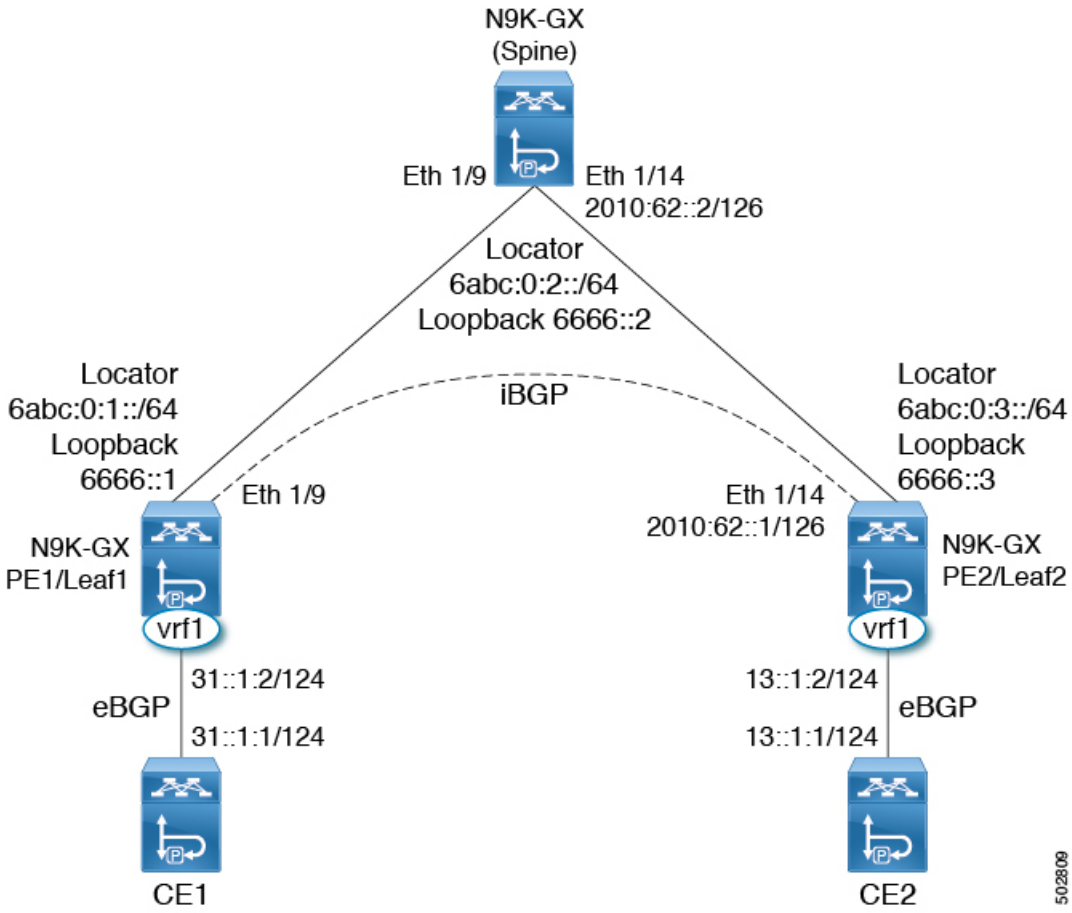
The following functions are supported in Cisco NX-Release 9.3(3):

- End
- End DT4/DT6/DT46
- T Encaps Red
- Transit Functionality (with and without SRH)

SRv6 Topology

This diagrams describes the SRv6 topology.

Figure 1: SRv6 Topology



In this example, the underlay IPv6 is enable with IS-IS. The interface between PE1 and spine are enabled with the link local addresses, while the interfaces between PE2 and spine are configured with the IPv6 addresses. In this topology, the configuration spine is also enabled for SRv6. The spine can act as a pure IPv6 underlay. The PE1/Leaf1 peers with PE2/Leaf2 over iBGP session to exchange VPN prefixes. The PE1 is attached to CE1 in vrf1 and learns VPN prefixes via eBGP session. Similarly, the PE2 is attached to CE2 in vrf1 and learns VPN prefixes via eBGP session.

Guidelines and Limitations for SRv6

SRv6 has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(3), SRv6 is supported on Cisco Nexus 9300-GX and 9300-GX2 platform switches.
- In Cisco NX-OS Release 9.3(3), only a single locator is supported.
- Layer 3 interface and Layer 3 port-channel are the supported uplinks toward the fabric. SVI and subinterfaces are not supported.

- Coexistence of the SRv6 feature and the MPLS SR-TE feature is not supported on Cisco Nexus 9000 switches.

Configuring SRv6

You can enable SRv6 and configure the locator with its prefix.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch#configure terminal`	Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enables segment routing over SRv6.

Step 3	**srv6**
	Example:
	`switch(config-sr)#srv6`
	`switch(config-sr-srv6)#`
		Enables segment routing over SRv6.

Step 4	**locators**
	Example:
	`switch(config-srv6)#locators`
	`switch(config-srv6-locators)#`
		Enter locator configuration mode.

Step 5	**locator** *name* **Example:** `switch(config-srv6-locators)#locator loc1`	Configure the locator.
Step 6	**prefix** *ipv6 address*/*len* **Example:** `switch(config-srv6-locator)# prefix 6abc:0:1::/64`	Configures the locator prefix.
Step 7	**exit** **Example:** `switch(config-srv6-locators)# exit`	Exits the locator configuration mode.

Configuring Encapsulation Parameters

You can obtain the source IPv6 address using the SRv6 encapsulation configuration.

Before you begin

Ensure that **feature srv6** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch#configure terminal`	Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 3	**srv6**
	Example:
	`switch(config-sr)#srv6`
	`switch(config-sr-srv6)#`
		Enables segment routing over SRv6.

Step 4	**locators**
	Example:
	`switch(config-sr-srv6)#locators`
	`switch(config-sr-srv6-locator)#`
		Enters the locators configuration mode.

Step 5	**locator** *name*
	Example:
	`switch(config-sr-srv6-locator)#locator loc1`
	`switch(config-sr-srv6-locator)#`
		Configures the global locator that can be used for all IPv4 and IPv6 VRFs and enters the locator configuration mode.

Step 6	**encapsulation**
	Example:
	`switch(config-sr-srv6)#encapsulation`
	`switch (config-sr-srv6-encap)#`
		Enters the encapsulation configuration mode.

Step 7	**source-address** *ipv6-address* **Example:** `switch(config-sr-srv6-encap)#source-address 6666::1`	Configures the source IPv6 address for SRv6 encapsulation.

Configuring IPv6 Underlay

You can configure IPv6 underlay with one of the following:

- IS-IS
- OSPFv3
- BGP

Configuring SRv6 with IS-IS Protocol

You can configure SRv6 with IS-IS protocol.

Before you begin

Ensure that the following conditions are met:

- The **feature srv6** is enabled.
- The **feature isis** is enabled.
- SRv6 is enabled under the IPv6 address-family in IS-IS.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router isis** *instance-tag*
	Example:
	`switch(config)# router isis 1`
	`switch(config-router)#`
		Creates a new IS-IS instance with the configured instance tag.

Step 3	**address-family** *ipv6* **unicast**
	Example:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode.

Step 4	**segment-routing srv6**
	Example:
	`switch(config-router-af)# segment-routing srv6`
	`switch(config-router-af-srv6)#`
		Configures SRv6 with IS-IS protocol.

	Command or Action	Purpose

Step 5	**locator** *name*
	Example:
	`switch(config-router-af-srv6)# locator loc1`
	`switch(config-router-af-srv6)#`
		Configure the locator.

Configuring SRv6 with OSPFv3 Protocol

Before you begin

- Ensure that **feature srv6** is enabled.
- Ensure that feature **ospfv3** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**route-map LOCATOR_MAP permit 10** **Example:** `switch(config-router)# route-map LOCATOR_MAP permit 10`	

Step 3	**router ospfv3** *process_tag*
	Example:
	`switch(config)# router ospfv3`
	`switch(config-router)#`
		Enables the OSPF mode.

Step 4	**address-family ipv6 unicast**
	Example:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		Enters address family configuration mode.

Step 5	**redistribute srv6 locator route-map LOCATOR_MAP** **Example:** `switch(config-router)# redistribute srv6 locator route-map LOCATOR_MAP`	

Configuring SRv6 with BGP

When locator is configured under BGP, it creates route of the locator prefix in its IPv6 unicast table and advertises it to its peers.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**segment-routing srv6**
	Example:
	`switch(config-router)# segment-routing srv6`
	`switch(config-router-srv6)#`
		Configures SRv6 with the BGP.

Step 4	**locator** *name*
	Example:
	`switch(config-router-srv6# locator loc1`
	`switch(config-router-srv6)#`
		Configures the locator.

Step 5	**exit**
	Example:
	`switch(config-router-srv6)# exit`
	`switch(config-router)#`
		Exits the SRv6 configuration mode.

Configuring Layer 3 VPN over SRv6

When a locator is configured under BGP, it creates route of the locator prefix in its IPv6 unicast table and advertises it to it's peers. This locator is used for allocating SRv6 SIDs for VRFs.

In Cisco NX-OS Release 9.3(3), the Cisco NX-OS switches support only one locator.

You can configure DT4 and DT6 SIDs separately under each address family. If the DT46 is configured under the VRF, then End.DT4 and End.DT6 configurations are not allowed under each address family.

Allocating DT46 SIDs for VRF

You can configure SRv6 with Layer 3 VPN fabric.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**segment-routing srv6**
	Example:
	`switch(config-router)# segment-routing srv6`
	`switch(config-router-srv6)#`
		Configures SRv6 with the BGP.

Step 4	**locator** *name*
	Example:
	`switch(config-router-srv6# locator loc1`
	`switch(config-router-srv6)#`
		Configures the locator.

Step 5	**exit**
	Example:
	`switch(config-router-srv6)# exit`
	`switch(config-router)#`
		Exits the SRv6 configuration mode.

Step 6	**vrf** *name*
	Example:
	`switch(config-router)# vrf vrf1`
	`switch(config-router-vrf)#`
		Configures the VRF.

Step 7	**segment-routing srv6**
	Example:
	`switch(config-router-vrf-af)#`
	`segment-routing srv6`
	`switch(config-router-vrf-af-srv6)#`
		Configures SRv6 and enters the VRF SRv6 configuration mode.

	Command or Action	Purpose
Step 8	**alloc mode per-vrf** **Example:** `switch(config-router-vrf-af-srv6)# alloc mode per-vrf`	Allocates SRv6 End DT46 per VRF.

Allocating DT4 and DT6 SIDs Per VRF

You can configure SRv6 with Layer 3 VPN fabric.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**segment-routing srv6**
	Example:
	`switch(config-router)# segment-routing srv6`
	`switch(config-router-srv6)#`
		Configures SRv6 with the BGP.

Step 4	**locator** *name*
	Example:
	`switch(config-router-srv6# locator loc1`
	`switch(config-router-srv6)#`
		Configures the locator.

Step 5	**exit**
	Example:
	`switch(config-router-srv6)# exit`
	`switch(config-router)#`
		Exits the SRv6 configuration mode.

Step 6	**vrf** *name* **Example:**	Configures the VRF.

	Command or Action	Purpose
	`switch(config-router)# vrf vrf1` `switch(config-router-vrf)#`	

Step 7	**address-family (ipv4 | ipv6} unicast**
	Example:
	`switch(config-router-vrf)# address-family (ipv4 | ipv6} unicast`
	`switch(config-router-vrf-af)#`
		Configures the IPv4 or IPv6 address family and enters the address family configuration mode.

Step 8	**segment-routing srv6**
	Example:
	`switch(config-router-vrf-af)# segment-routing srv6`
	`switch(config-router-vrf-af-srv6)#`
		Configures SRv6 and enters the VRF SRv6 configuration mode.

Step 9	**alloc mode per-vrf** **Example:** `switch(config-router-vrf-af-srv6)# alloc mode per-vrf`	Allocates SRv6 End DT4 or DT6 per VRF.

Allocating SRv6 DT46 SIDs for Global VRF

You can allocate SRv6 DT46 SIDs for global VRF with Layer 3 VPN fabric.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**segment-routing srv6**
	Example:
	`switch(config-router)# segment-routing srv6`
	`switch(config-router-srv6)#`
		Configures SRv6 with the BGP.

	Command or Action	Purpose

Step 4	**locator** *name*
	Example:
	`switch(config-router-srv6# locator loc1`
	`switch(config-router-srv6)#`
		Configures the locator.

Step 5	**alloc mode per-vrf** **Example:** `switch(config-router-srv6)# alloc mode per-vrf`	Allocates SRv6 End DT4 or DT6 for the global VRF.

Allocating SRv6 DT4 SIDs for IPv4 AF in Global VRF

You can allocate SRv6 DT4 SIDs for IPv4 address family in the global VRF with Layer 3 VPN fabric.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**address-family ipv4 unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		Configures the IPv4 address family and enters the address family configuration mode.

Step 4	**segment-routing srv6**
	Example:
	`switch(config-router-af)# segment-routing srv6`
	`switch(config-router-af-srv6)#`
		Configures SRv6 with the BGP.

Step 5	**alloc mode per-vrf** **Example:**	Allocates SRv6 End DT4 for address family in global VRF.

	Command or Action	Purpose
	`switch(config-router-af-srv6)# alloc mode per-vrf`	

Allocating SRv6 DT6 SIDs for IPv6 AF in Global VRF

You can allocate SRv6 DT6 SIDs for IPv6 address family in the global VRF with Layer 3 VPN fabric.

Before you begin

Ensure that **feature srv6** is enabled.

Ensure that **feature bgp** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enter BGP router configuration mode.

Step 3	**address-family ipv6 unicast**
	Example:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		Configures the IPv6 address family and enters the address family configuration mode.

Step 4	**segment-routing srv6**
	Example:
	`switch(config-router-af)# segment-routing srv6`
	`switch(config-router-af-srv6)#`
		Configures SRv6 with the BGP.

Step 5	**alloc mode per-vrf** **Example:** `switch(config-router-af-srv6)# alloc mode per-vrf`	Allocates SRv6 End DT6 for address family in global VRF.

Verifying the SRv6 Configuration

To display BGP specific SRv6 configuration, perform one of the following tasks:

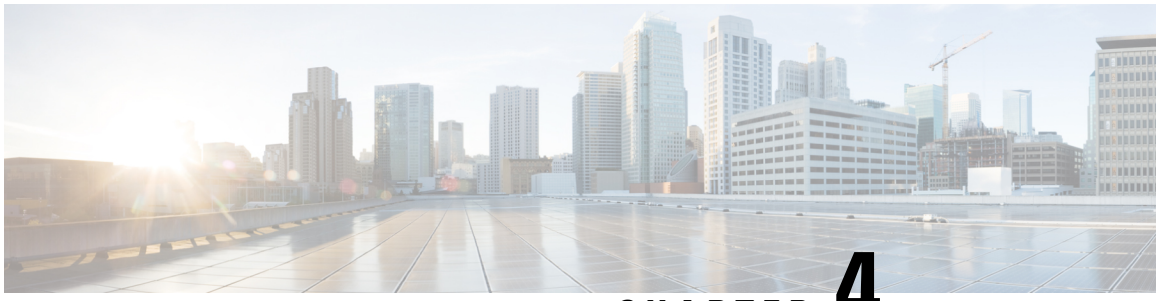
Command	Purpose
Show bgp segment-routing srv6	Displays the BGP SRv6 locator and SID for all VRFs.
Show bgp process *name*	Displays the BGP SRv6 SID for that VRF and the configured locator.

Configuration Example for SRv6

This example shows the SRv6 configuration:

```
feature bgp
feature isis
feature srv6
segment-routing
  srv6
    locators
      locator first
        prefix 6abc:0:1::/64
    encapsulation
       source-address 6666::1

route-map EVERYTHING permit 10

vrf context vrf1
  rd auto
  address-family ipv4 unicast
    route-target import 6603:1
    route-target export 6603:1
  address-family ipv6 unicast
    route-target import 6603:1
    route-target export 6603:1

interface Ethernet1/7/1
  no shutdown

interface Ethernet1/7/1.1
  encapsulation dot1q 101
  vrf member vrf1
  ip address 31.0.1.2/24
  ipv6 address 31::1:2/124
  no shutdown

interface Ethernet1/9
  ipv6 address use-link-local-only
  ipv6 router isis SR-ISIS-6
  no shutdown

interface loopback0
  ip address 6.6.6.1/32
  ipv6 address 6666::1/128
  ipv6 router isis SR-ISIS-6

router isis SR-ISIS-6
  net 66.0000.0000.0000.6001.00
  metric-style transition
  log-adjacency-changes
  address-family ipv6 unicast
    segment-routing srv6
      locator first
```

```
    maximum-paths 16

router bgp 6603
  router-id 6.6.6.1
  segment-routing srv6
    locator first
    alloc mode per-vrf
  address-family ipv4 unicast
    redistribute direct route-map EVERYTHING
  address-family ipv6 unicast
    redistribute direct route-map EVERYTHING
  neighbor 6666::3
    remote-as 6603
    update-source loopback0
    address-family ipv4 unicast
    address-family ipv6 unicast
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
vrf vrf1
    address-family ipv4 unicast
      redistribute direct route-map EVERYTHING
      segment-routing srv6
        alloc mode per-vrf
    address-family ipv6 unicast
      redistribute direct route-map EVERYTHING
      segment-routing srv6
        alloc mode per-vrf
    neighbor 31::1:1
      remote-as 1001
      update-source Ethernet1/7/1.1
      address-family ipv6 unicast
    neighbor 31.0.1.1
      remote-as 1001
      update-source Ethernet1/7/1.1
      address-family ipv4 unicast
```



CHAPTER 4

Configuring SRv6 Traffic Engineering

This chapter contains information on how to configure SRv6 traffic engineering.

About SRv6 Traffic Engineering

SRv6 traffic engineering (SRv6 TE) uses the concept of source routing, where the source calculates the path and encodes it in the packet header as a list of segments. This list of segments is added to an IPv6 routing header called the SRv6 Segment Routing Header (SRH) in the incoming packet.

With SRv6 TE, the network does not need to maintain per-application and per-flow state on each node. Instead only the head-end nodes on the edge of the network where the traffic enters the policy need to maintain state. The remaining nodes simply obey the forwarding instructions that are provided in the packet.

SRv6 traffic engineering can utilize network bandwidth more effectively than traditional MPLS RSVP-TE by using ECMP within each segment. In addition, by using a single intelligent source that it relieves remaining routers from the task of calculating the required path through the network.

SRv6 Traffic Engineering Policies

SRv6 traffic engineering uses a “policy” to steer traffic through the network. A SRv6 traffic engineering policy is a container that includes sets of segments.

The headend imposes SID list on traffic flow. Each transit node in the SID stack uses the top SID to choose the next-hop, pops the SID, and forwards the packet to the next node. The packet is forwarded with the remainder of the SID stack, until it reaches the ultimate destination.

A SRv6 traffic engineering policy is uniquely identified by a tuple (color, endpoint). Color is represented as a 32-bit number while the IPv6 address is an endpoint. Every SRv6 traffic engineering policy has a color value. Every policy between the same node pairs requires unique color value. Multiple SRv6 traffic engineering policies can be created between the same two endpoints by choosing different colors for these policies.

In Cisco NX-OS Release 9.3(5), Cisco Nexus 9000 Series switches support only explicit SRv6 policy.

Explicit SRv6 Traffic Engineering Policy

An explicit policy is a list of IPv6 addresses representing an ordered list of segment IDs. The policy path is statically configured because the segment list is defined by the operator.

To create an explicit policy, you must first define segment list (s), the policy name, endpoint, and color and reference it to a segment list from the policy. Segment lists are defined separately since these can be reused between different policies.

Currently, the list of segments in an explicit policy must contain only the SRv6 END SIDs of the nodes in the path (excluding the headend). Each policy supports a maximum of three preferences; three segment lists where only one is active at any given point. This allows you to have one active segment list and two backup segment lists.

Destination Prefix Based Traffic Steering

Global VRF

You can configure a destination prefix and a prefix length in the global VRF and steer it through a SRv6 traffic engineering policy. This destination prefixes can be either IPv4 or IPv6 addresses. A policy can be referenced for traffic engineering based on the policy name or the color and the endpoint. If the destination prefix is an IPv6 prefix which is reachable via the IGP, BGP, or static without any SRv6 encapsulation, the traffic steering occurs with a T.insert behavior with the SIDs in the SRH. In this case, the traffic engineered route takes precedence over the original best route in the forwarding.

If the destination prefix is an IPv4 or IPv6 prefix which is reachable via an SRv6 encapsulation, the traffic steering occurs with the T.encap behavior. The remote encapsulation is inherited from the remote global VRF over SRv6. The traffic engineered path is derived from a SRv6 traffic engineering policy. In this case, the final traffic engineered route takes precedence over the original T.encap route in forwarding.

You can configure a complete encapsulation without a SRv6 traffic engineering policy. In this case, encapsulation that is configured by you takes precedence over remote learned remote routes.

VPN VRF

You can configure a destination prefix and a prefix length in a VPN VRF and steer it through a SRv6 traffic engineering policy. This destination prefix can be IPv4 or IPv6 addresses. A policy can be referenced for traffic engineering based on the policy name or the color and the endpoint.

If the destination prefix is an IPv4 and IPv6 prefix and is learned from BGP, the remote encapsulation is inherited from the remote VPN route. The traffic engineering path is derived from a SRv6 traffic engineering policy. The final traffic engineering SIDs with T.Encap take precedence over the original best route in the forwarding.

You can configure a complete encapsulation without a SRv6 traffic engineering policy. In this case, encapsulation that is configured by you takes precedence over remote learned remote routes.

Guidelines and Limitations for SRv6 Traffic Engineering

SRv6 traffic engineering has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(3), SRv6 traffic engineering is supported on Cisco Nexus 9300-GX and 9300-GX2 platform switches.
- In Cisco NX-OS Release 9.3(5), only one tunnel profile is supported.
- The maximum number of SRv6 SIDs in the SR-TE path with T.Encaps is 4.
- The maximum number of SRv6 SIDs in the SR-TE path with T.Insert is 8.
- ECMP is not supported at the policy level. There is only one path per preference in the SR-TE. Maximum of three preferences are supported.
- The MPLS segment routing and SRv6 features cannot be enabled concurrently.
- IPv6 redirects must not be configured on core interfaces. Use the **no ipv6 redirects** command to disable IPv6 redirects.

Creating the Explicit SID List

You can create segment-list and explicit SRv6 traffic engineering policy.

Before you begin

You must ensure that the SRv6 feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**segment-routing**
	Example:
	`switch(config)#segment-routing`
	`switch(config-sr)#`
		Enters the segment routing configuration mode.

Step 3	**srv6**
	Example:
	`switch(config)#srv6`
	`switch(config-sr-srv6)#`
		Enables segment routing over SRv6.

Step 4	**traffic-engineering** **Example:**	Enters the traffic engineering mode.

	Command or Action	Purpose
	`switch(config-sr-srv6)# traffic-engineering` `switch(config-sr-srv6-te)#`	

Step 5	**segment-list name** *sidlist-name*
	Example:
	`switch(config-sr-srv6-te)# segment-list name black`
	`index 1 segment-routing srv6 A1:0:0:2:1::`
	`index 5 segment-routing srv6 A1:0:0:3:1::`
	`segment-list name blue`
	`index 1 segment-routing srv6 A1:0:0:4:1::`
	`index 5 segment-routing srv6 A1:0:0:5:1::`
		Creates the explicit SID list.

Step 6	**policy** *policy name* **Example:** `switch(config-sr-te-color)# policy 1`	Configures the policy.
Step 7	**color** *numberIPv6-end-point* **Example:** `switch(config-sr-te-pol)# color 201 endpoint A1:0:0:07::1`	Configures the color and the endpoint of the policy.

Step 8	**candidate-paths**
	Example:
	`switch(config-sr-te-color)# candidate-paths`
	`switch(cfg-cndpath)#`
		Specifies the candidate paths for the policy.

Step 9	**preference** *preference-number*
	Example:
	`switch(cfg-cndpath)# preference 100`
	`switch(cfg-pref)#`
		Specifies the preference of the candidate path.

Step 10	**explicit segment-list** *sidlist-name*
	Example:
	`switch(cfg-dyn)# explicit segment-list blue`
	`switch(cfg-dyn)#`
		Specifies that the explicit list.

Step 11	**exit**
	Example:
	`switch(cfg-dyn)# exit`
	`switch(config)#`
		Exits the configuration mode.

	Command or Action	Purpose

Step 12	**srv6**
	Example:
	`switch(config)# srv6`
	`switch(config-srv6)#`
		Enters the SRv6 configuration mode.

Step 13	**locators**	Enters the locators configuration.
Step 14	**locator** *name*	Configures the locator name, which is the global locator name that was globally configured for SRv6.

Associating Prefixes to an Explicit SRv6 Traffic Engineering Policy

You can contain the source IPv6 address using the SRv6 encapsulation configuration.

Before you begin

Ensure that **feature srv6** is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature ofm** **Example:** `switch (config)# feature ofm`	Enables ofm.
Step 3	**tunnel profile** *main* **Example:** `switch(config-sr-srv6)# tunnel profile main`	Creates the tunnel profile for SRv6 encapsulation.

Step 4	**encapsulation srv6**
	Example:
	`switch(config-tnl-profile)# encapsulation srv6`
	`switch(config-tnl-profile)#`
		Creates a tunnel profile for SRv6.

Step 5	**route** *prefix* / *len* [**vrf** *vpm-vrf*] **via policy color** *color* **endpoint** *endpoint address*	Associates the prefix to the policy.

	Command or Action	Purpose
	Example: `switch(config-sr-srv6-encap)# route 10.1.1.2/32 vrf vrf1 via policy BLUE_PATH`	

Configuration Example for SRv6 Traffic Engineering

This example shows the SRv6 traffic engineering configuration:

```
segment-routing
    traffic-engineering
        srv6
            locator main
    segment-list name black
        index 1 A1:0:0:2:1::
        index 5 A1:0:0:3:1::
    segment-list name blue
        index 1 A1:0:0:4:1::
        index 5 A1:0:0:5:1::
    policy policy1
        color 201 endpoint A1:0:0:07::1
        candidate-paths
            preference 70
                explicit segment-list black
            preference 100
                explicit segment-list blue
```

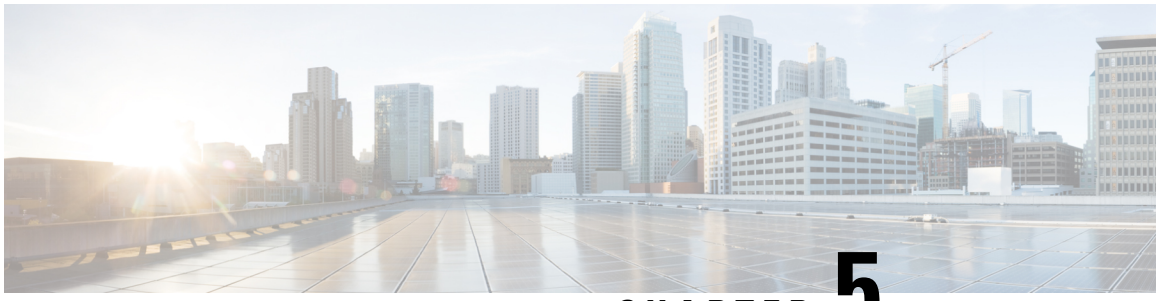
Examples of configuring prefixes for SRv6 traffic engineering. The VRF name variable (vrf_name) can be global or default, or the L3VPN VRF.

```
tunnel-profile main
    encapsulation srv6

  route vrf <vrf_name> 3.0.1.0/24  via policy name POLICY1
  route vrf <vrf_name> 3::1:0/124  via policy name POLICY1

  route vrf <vrf_name> 3.0.2.0/24  via policy color 1 endpoint fd00::a02:2
  route vrf <vrf_name> 3::2:0/124  via policy color 1 endpoint fd00::a02:2

  route vrf <vrf_name> 3.0.3.0/24 remote-locator fd01:0:0:2:: function 65533
  route vrf <vrf_name> 3::3:0/124 remote-locator fd01:0:0:2::  function 65533

  route vrf <vrf_name> 3.0.4.0/24 remote-locator fd01:0:0:2:: function 65533 via policy
color 1 endpoint fd00::a02:2
  route vrf <vrf_name> 3::4:0/124 remote-locator fd01:0:0:2:: function 65533 via policy
color 1 endpoint fd00::a02:2

  route vrf <vrf_name> 3.0.5.0/24 remote-locator fd01:0:0:3:: function 65533 via policy
name POLICY1
  route vrf <vrf_name> 3::5:0/124 remote-locator fd01:0:0:3:: function 65533 via policy
name POLICY1
```

Verifying SRv6 Traffic Engineering Configuration

To display the SRv6 traffic engineering configuration, perform one of the following tasks:

Command	Purpose
show running srte	Displays the SRv6 traffic engineering configuration.
show running ofm	Displays the static route configuration.



CHAPTER 5

Configuring SRv6 OAM

This chapter contains information on SRv6 OAM.

About SRv6 OAM

Segment Routing over IPv6 (SRv6) Operation, Administration, and Maintenance (OAM) feature monitors SRv6 path connectivity and isolates forwarding problems to assist with fault detection and troubleshooting in the network. SRv6 OAM uses IPv6 ping and pathtrace for diagnosis.

SRv6 OAM provides the capability to choose a particular path when there are multiple equal cost destination paths. It also allows you to verify the reachability to an end host.

The SRv6 OAM feature is enabled using the Next Generation OAM (NGOAM) feature.

SRv6 OAM provides the following functions for diagnostics purposes:

- Ping or pathtrace to loopback
- Ping or pathtrace to SID
- Ping or pathtrace to a host in a VRF

Terminology used in SRv6 OAM is as follows:

- Ping - One or more probe packets are sent to a specific destination in order to elicit an ICMP response.
- Pathtrace - Includes a series of probe packets that are sent with a monotonically increasing IPv6 Hop-Count (HC) value used to map a path to a destination node. A pathtrace differs from a traceroute only in that additional TLVs are included in the request and response to facilitate advanced diagnostics and reporting.
- Probe Packet - Also referred to as a probe, this is a single request packet sent by either a ping or pathtrace.

- Initiator Node - Is the node where the ping or pathtrace is run. The probe packets are crafted by NGOAM on this node, and sent out the appropriate interface, passing through transit nodes, and finally reaching the egress or final node.
- Transit Node - Nodes traversed by ping or pathtrace packets. In the case of ping, unless the transit node is a segment end no special action is taken (routing is performed as normal). In the case of a pathtrace, OAM on transit nodes processes the packet and sends a response due to TTL expiry.
- Egress Node - The remote node, that is the fabric edge node that the probe packets reach. Specifically this term is used in the overlay host ping cases where the probe is processed by OAM, but a proxy probe may be sent to the host.
- Final Node - The remote node to which the probe packets are destined to.

Guidelines and Limitations for SRv6 OAM

SRv6 OAM has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.3(3), SRv6 OAM is supported on Cisco Nexus 9300-GX and 9300-GX2 platform switches.
- The SRv6 OAM feature requires time synchronization mechanism such as PTP or NTP used on Cisco NX-OS devices in order to measure one-way delay measurements.

SRv6 OAM Operations

SRv6 OAM operations include:

- Ping and pathtrace to an IPv6 address via a segment list
- Ping and pathtrace to a SID
- Ping and pathtrace to SID with Segment List
- Ping and trace to a SID function
- Ping to IPv6 address or SID in Segment-by-Segment mode
- Ping to host in overlay
- Ping to host in overlay following Specific Application Path (Flow Tracking)
- Diagnostic Information in Replies
- Asynchronous Probes
- CLI Profiles

The following features are supported:

- Ping and pathtrace to an IPv6 address through a segment list - The ping or pathtrace is normal, but the path of the probe packets is modified to follow the configured SID list. The probes are sent with a SRH that directs the packets to follow the SID list.

- Ping and pathtrace to a SID - Instead of the IP address of a node, the ping or pathtrace is to the SID itself. Since SIDs do not terminate a packet, in order for the OAM to respond to the probe packet, End OP or End OTP SID are used
- Ping and pathtrace to a SID with a segment list - Supports the specified path using a segment list.
- Ping to an IPv6 address or SID in Segment-by-Segment mode - Supports segment-by-segment ping which provides multiple proof of transit responses. The probes use the O-bit mechanism to trigger responses from each SRv6 segment terminus, except for the last one for which the End OTP SID is used.
- Ping to host in overlay - Supports a ping from the PE to a host that is beyond a remote PE.
- Ping to Host in overlay (flow tracking) - Supports a ping based on the outer packet destination address, source address, and the flow label.
- Diagnostic Information in replies - Pathtrace includes additional fields in the packet that allow responses to carry diagnostic information, for example, interface load and statistics of the hops taken by these messages. If an intermediate device does not have SRv6 OAM enabled, the pathtrace behaves as a simple traceroute for those hops and it provides only the hop information
- Asynchronous probes - Supports ping commands in an asynchronous mode. In this case, the ping commands sends the probes in the background and does not wait for the replies.
- CLI profiles - The NGOAM feature provides an option to configure profiles that can be used in the ping and the pathtrace commands. The parameters provided in these commands can be stored as a profile and reused in the ping or pathtrace commands.

Configuring SRv6 OAM

Beginning Cisco NX-OS Release 9.3(3), you can configure SRv6 OAM on Cisco Nexus 9364C-GX, Cisco Nexus 9316D-GX, and Cisco Nexus 93600CD-GX switches.

Before you begin

Ensure that the **feature srv6** feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch#configure terminal`	Enters global configuration mode.
Step 2	[**no**] **feature ngoam** **Example:** `switch(config)#feature ngoam`	Enables or disables NGOAM feature.

SRv6 OAM Commands

SRv6 OAM supports the following commands:

Table 2: SRv6 OAM Commands

Commands	Description
{**ping** | **pathtrace**} **srv6** *IP address* [**via SID1, SID2 sid-list-end**] [**no-reduced-srh**]	Initiates a ping or a pathtrace to a regular IPv6 address. The **via** keyword defines a list of SRv6 SIDs. The **no-reduced-srh** keyword causes the ping or the pathtrace to use a full SRH instead of the default reduced SRH.
{**ping** | **pathtrace**} **srv6 sid** *SID* [**via SID1, SID2 sid-list-end**] [**end-otp SID3**]	Initiates a ping or a pathtrace to the IPv6 SID instead of the IPv6 address. The **via** keyword defines a list of intermediate SRv6 SIDs that can be traversed by the probe packets. This command introduces SRH into the SRv6 probe packets. The **end-otp** keyword is used to override the SID used for the End.OTP function on the remote node.
ping srv6 *IP address* [**via SID1, SID2 sid-list-end**]	By default, initiates a ping in the segment-by-segment mode. In this mode, the node that each SID represents sends a response to the ping. The **no-proof-of-transit** keyword is used to not receive any replies from each node in the SID list.
ping srv6 sid *SID*[**via SID1, SID2 sid-list-end**] [**no-proof-of-transit**]	
{**ping** | **pathtrace**} **srv6** *IP address* **vrf** *VRF* [**verify-host**]	Initiates a ping or pathtrace to a host in a specified layer3 overlay network. The ping is initiated from a PE node of the VPN and terminated at either the remote PE node or at the specified host in the VRF. The **verify-host** keyword is used to generate a secondary ping probe and send it from the remote PE node to the host in the VRF. This validates the connectivity.

Commands	Description
{**ping** |**pathtrace**} **srv6** *IP address* **VRF** *VRF* [**payload** [**ip** | **ipv6**] *DST-IP SRC-IP* [**port PORT**] [**proto** *PROTO*] **payload-end**] [**verify-host**]	The use of the payload keyword ensures where possible that the ECMP choices at each hop are the same as for the actual data traffic matching the profile described in the payload. This can be used to troubleshoot the case where flows for a specific application are failing due to only some links being faulty in an ECMP set. This command can also be used to validate the specific ECMP path in case of partial fabric failures.

Examples for SRv6 OAM Configuration

The following examples show ping and pathtrace configurations:

- The following example shows a ping to IPv6 address 4::4.

```
ping srv6 4::4
```

- The following example shows a ping to IPv6 address 4::4 via SID list cafe:0:0:2:1:: using the default proof of transit.

```
ping srv6 4::4 via cafe:0:0:2:1:: sid-list-end
```

- The following example shows a ping to IPv6 address 4::4 via SID list cafe:0:0:2:1:: without proof of transit.

```
ping srv6 4::4 via cafe:0:0:2:1:: sid-list-end no-proof-of-transit
```

- The following example shows a ping to IPv6 address 4::4 via SID list cafe:0:0:2:1:: using a non-reduced SRH.

```
ping srv6 4::4 via cafe:0:0:2:1:: sid-list-end no-reduced-srh
```

- The following example shows a ping to SID cafe:0:0:4:1:: using the default end-otp SID.

```
ping srv6 sid cafe:0:0:4:1::
```

- The following example shows a ping to SID cafe:0:0:4:1::using the user provided end-otp SID cafe:0:0:4:2:: .

```
ping srv6 sid cafe:0:0:4:1:: end-otp cafe:0:0:4:2::
```

- The following example shows a ping to IPv4 host 10.10.10.10 in vrf red without host verification.

```
ping srv6 1.1.1.1 vrf red
```

- The following example shows a ping to IPv6 host 104::4 in vrf red via SID list cafe:0:0:2:1:: with the default proof of transit and without host verification.

```
ping srv6 104::4 vrf red via cafe:0:0:2:1:: sid-list-end
```

- The following example shows a ping to IPv6 host 104::4 in vrf red via SID list cafe:0:0:2:1:: without proof of transit and without host verification.

```
ping srv6 104::4 vrf red via cafe:0:0:2:1:: sid-list-end no-proof-of-transit
```

- The following example shows a ping to IPv4 host 40.40.40.40 in the Global vrf without host verification.

```
ping srv6 40.40.40.40
```

- The following example shows a ping to IPv6 host 104::4 in vrf red using flow tracing and without host verification.

```
ping srv6 104::4 vrf red payload ipv6 104::4 101::1 payload-end
```

- The following example shows a ping to IPv6 host 104::4 in vrf red using flow tracing and with host verification.

```
ping srv6 104::4 vrf red payload ipv6 104::4 101::1 payload-end verify-host
```

- The following example shows a pathtrace to IPv6 address 4::4.

```
pathtrace srv6 4::4
```

- The following example shows a pathtrace to IPv6 address 4::4 via SID list cafe:0:0:2:1:: using the default proof of transit.

```
pathtrace srv6 4::4 via cafe:0:0:2:1:: sid-list-end
```

- The following example shows a pathtrace to IPv6 address 4::4 via SID list cafe:0:0:2:1:: using a non-reduced SRH.

```
pathtrace srv6 4::4 via cafe:0:0:2:1:: sid-list-end no-reduced-srh
```

- The following example shows a pathtrace to SID cafe:0:0:4:1:: using the default end-otp SID.

```
pathtrace srv6 sid cafe:0:0:4:1::
```

- The following example shows a pathtrace to SID cafe:0:0:4:1::using the user provided end-otp SID cafe:0:0:4:2:: .

```
pathtrace srv6 sid cafe:0:0:4:1:: end-otp cafe:0:0:4:2::
```

- The following example shows a pathtrace to IPv4 host 10.10.10.10 in vrf red.

```
pathtrace srv6 1.1.1.1 vrf red
```

- The following example shows a pathtrace to IPv6 host 104::4 in vrf red via SID list cafe:0:0:2:1:: .

```
pathtrace srv6 104::4 vrf red via cafe:0:0:2:1:: sid-list-end
```

- The following example shows a pathtrace to IPv4 host 40.40.40.40 in the Global vrf.

```
pathtrace srv6 40.40.40.40
```

- The following example shows an pathtrace to IPv6 host 104::4 in vrf red using flow tracing.

```
pathtrace srv6 104::4 vrf red payload ipv6 104::4 101::1 payload-end
```

- The following example shows a pathtrace to IPv6 host 104::4 in vrf red using flow tracing and with host verification.

```
pathtrace srv6 104::4 vrf red payload ipv6 104::4 101::1 payload-end verify-host
```