RSVP-VRF Lite Admission Control

Last Updated: January 15, 2013

The RSVP--VRF Lite Admission Control feature introduces support for Resource Reservation Protocol (RSVP) call admission control (CAC) in an IP session within the context of a virtual routing and forwarding (VRF) instance.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for RSVP-VRF Lite Admission Control

You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for RSVP-VRF Lite Admission Control

- Multi-topology routing (MTR) is not supported.
- Multiprotocol Label Switching (MPLS) virtual private network (VPN) VRFs are not supported.
- RSVP authentication is not supported.

Information About RSVP-VRF Lite Admission Control

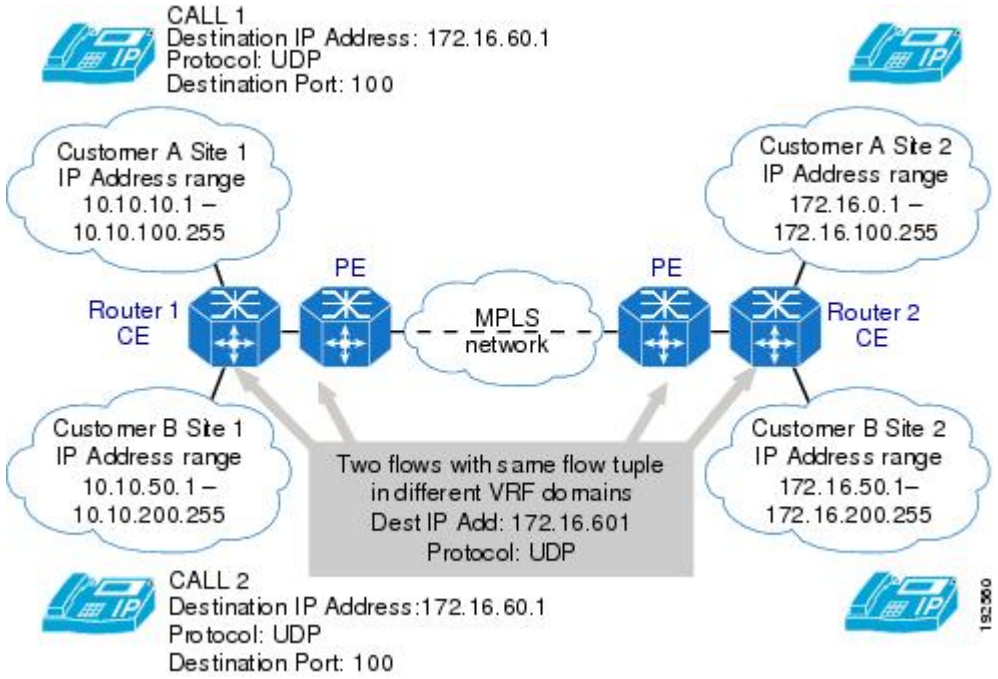
Overview of RSVP-VRF Lite Admission Control

An RSVP flow is identified by its tuple, which includes its destination IP address, its destination port, and its protocol. This tuple should be unique on all the nodes along the path from the sender to the receiver. In the context of the global routing domain, each flow can be uniquely identified through its tuple.

However, with the implementation of virtual routing and forwarding (VRF), a separate instance of the routing and forwarding table for each VRF routing domain can exist. Each of the VRF instances has its own address pool range, which could overlap between VRF routing domains. This poses a problem to the existing implementation of RSVP, where sessions are identified by the tuple. Sessions with the same tuple can exist in the context of different VRF domains. To solve the problem, the tuple has to be extended to take into account the VRF instance. The new tuple has a *VRF ID, a destination IP address* , a *destination port, and a protocol* . The VRF ID is derived based on the interface on which an RSVP packet has been received and is not signaled using RSVP. Therefore, each node needs to infer the VRF ID based on the RSVP control packet's incoming interface.

The figure below shows a VRF-lite deployment scenario.

Figure 1 RSVP VRF Deployment in VRF-Lite Network



The figure above shows VRF lite configured on Router 1 customer edge (CE) and Router 2 CE, and MPLS-VPN configured between the provider edge (PE) devices. In such a deployment scenario, the RSVP implementation needs to be VRF aware in the CE devices; that is, the flows must be recognized in the context of the VRF domain in which the sender and receiver of the flow reside. However, RSVP QoS is not enabled on the PE devices.

On the CE devices, with VRF lite functionality, VRF is identified based on the VRF configured on the incoming interface; that is, on the interface facing the customer site and the interface facing the PE.

Benefits of RSVP-VRF Lite Admission Control

The RSVP--VRF Lite Admission Control feature provides the benefits of RSVP in a VRF-lite environment to include the following:

- Guaranteed QoS through explicit admission control
- Virtualization
- Security
- Separation of routing contexts
- Overlapping of IP addresses

How to Configure RSVP-VRF Lite Admission Control

Note The tasks described in this section explain configuring a receiver proxy and a static sender, for you to quickly initiate and terminate an RSVP session, and verify your setup. In these tasks, the IOS RSVP implementation behaves as an RSVP endpoint and an RSVP initiator.

Enabling RSVP on an Interface

Perform this task to enable RSVP on all the interfaces along the path from the sender to the receiver.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ip vrf** *vrf-name*
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name*
8. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
9. Repeat the previous step for each interface that you want to enable.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)# ip routing`	Enables IP routing.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrf1`	Associates a VRF instance with an interface or subinterface.
Step 8	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 1158 100`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface that you want to enable.
Step 9	Repeat the previous step for each interface that you want to enable.	--
Step 10	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring a Receiver Proxy on a Tailend Device

Perform this task to configure a receiver proxy with a VRF on a tailend device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**} **Example:** `Device(config)# ip rsvp listener vrf myvrf 192.168.2.1 any any reply`	Configures an RSVP device to listen for PATH messages. • Enter the appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring a Static Sender on a Headend Device

Perform this task to configure a static sender with a VRF on a headend device to make the device proxy an RSVP PATH message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*] **Example:** `Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 vrf myvrf`	Enables a device to simulate a host generating RSVP PATH messages. • Enter the appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config)#` **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an RSVP Application Identity That Is VRF Aware

Perform the following task to configure an RSVP application identity that is VRF aware.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config)# ip rsvp policy vrf myvrf identity voice policy-locator voiceStream`	Creates a local policy for a VRF and enters local policy configuration mode. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring an RSVP Local Policy That Is VRF Aware

Perform the following task to configure an RSVP local policy that is VRF aware.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*\| **local**}}{**acl** *acl1*[*acl2...acl8*] \| **default** \| **identity** *alias1*[*alias2...alias4*]\| **origin-as** *as1* [*as2...as8*]}
4. {**accept** | **forward**[**all** | **path**\| **path-error** | **resv**\| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]\| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config)# ip rsvp policy vrf myvrf local default`	Creates a local policy for a VRF and enters local policy configuration mode. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 4	{**accept** | **forward**[**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Device(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. **Note** See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 5	**end** **Example:** `Device(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Verifying the RSVP-VRF Lite Admission Control Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode and in any order.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp counters** [**authentication**][**interface** *type number*| **neighbor**[**vrf**{*****| *vrf-name*}] | **state teardown**| **summary**]
3. **show ip rsvp host vrf** {***** | *vrf-name*} {**receivers** | **senders**} [*group-name* | *group-address*]
4. **show ip rsvp installed** [**vrf**{***** | *vrf-name*}] [*interface-type interface-number*] [**detail**]
5. **show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] [*interface-type interface-number*]
6. **show ip rsvp listeners** [*ip-address* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**]
7. **show ip rsvp neighbor** [**detail** | **inactive**[**detail**]] | **vrf**{***** | *vrf-name*}]
8. **show ip rsvp policy vrf** {***** | *vrf-name*} [**identity**[*alias*]] | **local**[**acl** *acl* | **default** | **detail**[**acl** *acl* | **default** | **identity** *alias* | **interface** *interface-type* | **origin-as** *as-number*]]
9. **show ip rsvp request** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]]
10. **show ip rsvp reservation** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}]
11. **show ip rsvp sender** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}]
12. **show ip rsvp signalling fast-local-repair** [**statistics**[**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp counters** [**authentication**][**interface** *type number*	**neighbor**[**vrf**{*****	*vrf-name*}]	**state teardown**	**summary**] **Example:** `Device#` **show ip rsvp counters neighbor vrf myvrf**	(Optional) Displays the number of RSVP messages that were sent and received on each interface. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 3	**show ip rsvp host vrf** {*****	*vrf-name*} {**receivers**	**senders**} [*group-name*	*group-address*] **Example:** `Device(config)# show ip rsvp vrf * senders`	(Optional) Displays specific information for an RSVP host configured with a VRF instance. • Enter the *vrf-name*name and any other appropriate keywords and arguments.

	Command or Action	Purpose
Step 4	**show ip rsvp installed** [**vrf**{***** | *vrf-name*}] [*interface-type interface-number*] [**detail**] **Example:** `Device# show ip rsvp installed vrf myvrf detail`	(Optional) Displays RSVP-related installed filters and corresponding bandwidth information. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 5	**show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] [*interface-type interface-number*] **Example:** `Device# show ip rsvp interface vrf myvrf detail`	(Optional) Displays information related to RSVP. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 6	**show ip rsvp listeners** [*ip-address* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**] **Example:** `Device# show ip rsvp listeners vrf myvrf1`	(Optional) Displays the RSVP listeners for a specified port or protocol. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 7	**show ip rsvp neighbor** [**detail** | **inactive**[**detail**]] | **vrf**{***** | *vrf-name*}] **Example:** `Device# show ip rsvp neighbor vrf myvrf1`	(Optional) Displays current RSVP neighbors. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 8	**show ip rsvp policy vrf** {***** | *vrf-name*} [**identity**[*alias*]] | **local**[**acl** *acl* | **default** | **detail**[**acl** *acl* | **default** | **identity** *alias* | **interface** *interface-type* | **origin-as** *as-number*]] **Example:** `Device# show ip rsvp policy vrf myvrf1`	(Optional) Displays information for an RSVP policy configured with a VRF instance. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 9	**show ip rsvp request** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] **Example:** `Device# show ip rsvp request vrf myvrf1`	(Optional) Displays RSVP-related request information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.

	Command or Action	Purpose
Step 10	**show ip rsvp reservation** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}] **Example:** `Device# show ip rsvp reservation vrf myvrf1`	(Optional) Displays RSVP-related receiver information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 11	**show ip rsvp sender** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}] **Example:** `Device# show ip rsvp sender vrf myvrf1`	(Optional) Displays RSVP PATH-related sender information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 12	**show ip rsvp signalling fast-local-repair** [**statistics**[**detail**]] **Example:** `Device# show ip rsvp signalling fast-local repair statistics detail`	(Optional) Displays fast-local-repair (FLR)-specific information, including VRF, maintained by RSVP.

Configuration Examples for RSVP-VRF Lite Admission Control

Examples Configuring RSVP-VRF Lite Admission Control

The following example enables RSVP on a device interface along the path from the sender to the receiver.

Note If the interface lies in a VRF domain, use the **ip rsvp bandwidth** command to enable RSVP for that VRF.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface Ethernet0/0
Device(config-if)# ip rsvp bandwidth 1158 100
Device(config-if)# end
```

The following example configures a receiver proxy with a specified VRF on a tailend device:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# ip rsvp listener vrf myvrf 192.168.2.1 any any reply

Device(config)# end
```

The following example configures a static sender with a specified VRF on a headend device:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 vrf myvrf
Device(config)# end
```

Examples Verifying RSVP-VRF Lite Admission Control

```
In the following example, all the interfaces associated with the VRF named myvrf display
in detail:
Device# show ip rsvp interface vrf myvrf detail
Se1/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 300K bits/sec
     Max. allowed (total): 400K bits/sec
     Max. allowed (per flow): 400K bits/sec
     Max. allowed for LSP tunnels using sub-pools (pool 1): 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Traffic Control:
     RSVP Data Packet Classification is OFF
     RSVP resource provider is: none
   Signalling:
     DSCP value used in RSVP msgs: 0x3F
     Number of refresh intervals to enforce blockade state: 4
   Authentication: disabled
     Key chain:   <none>
     Type:        md5
     Window size: 1
     Challenge:   disabled
   FRR Extension:
     Backup Path: Not Configured
   BFD Extension:
     State: Disabled
     Interval: Not Configured
   RSVP Hello Extension:
     State: Disabled
   RFC 3175 Aggregation: Enabled
     Role: interior
   VRF: myvrf
The following example displays details of the RSVP reservations installed for RSVP
session that belong to the VRF named myvrf:
Device# show ip rsvp installed vrf myvrf detail
RSVP: FastEthernet2/0 has the following installed reservations
RSVP Reservation. Destination is 10.10.10.10. Source is 10.10.10.12,
  Protocol is UDP, Destination port is 10, Source port is 10
  Traffic Control ID handle: C8000407
  Created: 22:51:26 UTC Sun Feb 17 2008
  Admitted flowspec:
    Reserved bandwidth: 10K bits/sec, Maximum burst: 10K bytes, Peak rate: 10K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations [0xBF000406]
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 12783 seconds
```

```
  Long-term average bitrate (bits/sec): 0 reserved, 0 best-effort
  Policy: INSTALL. Policy source(s): Default
VRF : myvrf
```

The following example shows the listeners configured for the VRF named myvrf:

```
Device# show ip rsvp listeners vrf myvrf
VRF : myvrf1
```

To Protocol DPort Description Action OutIf

10.0.2.1 any any RSVP Proxy reply

The following example shows the neighbors created for the VRF named myvrf:

```
Device# show ip rsvp neighbor vrf myvrf
VRF: myvrf
Neighbor        Encapsulation  Time since msg rcvd/sent
10.10.15.3         Raw IP         00:00:14   00:00:06
10.10.16.2         Raw IP         00:00:29   00:00:15
```

The following example displays all the locally created RSVP senders for the configured VRFs:

```
Device# show ip rsvp host vrf * senders
VRF: vrf2
To                  From                Pro DPort Sport Prev Hop      I/F      BPS
192.168.104.4       198.168.104.12      UDP 10    10    none          none     10K
  Mode(s): Host CLI
VRF: vrf1
To                  From                Pro DPort Sport Prev Hop      I/F      BPS
192.168.105.4       198.168.105.12      UDP 10    10    none          none     10K
  Mode(s): Host CLI
```

Additional References

The following sections provide references related to the RSVP--VRF Lite Admission Control feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
VRF-related internet draft	*Support for RSVP in Layer 3 VPNs*, Internet draft, November 19, 2007 [draft-davie-tsvwg-rsvp-l3vpn-01.txt]
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for RSVP-VRF Lite Admission Control

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for RSVP--VRF Lite Admission Control

Feature Name	Releases	Feature Information
RSVP--VRF Lite Admission Control	15.0(1)SY	The RSVP--VRF Lite Admission Control feature introduces support for RSVP CAC in an IP session within the context of a VRF instance. The following commands were introduced or modified by this feature: **debug ip rsvp** , **ip rsvp listener, ip rsvp policy vrf, ip rsvp reservation-host, ip rsvp sender-host, show ip rsvp counters, show ip rsvp host vrf, show ip rsvp installed, show ip rsvp interface, show ip rsvp listeners, show ip rsvp neighbor, show ip rsvp policy vrf, show ip rsvp request, show ip rsvp reservation, show ip rsvp sender, show ip rsvp signalling fast-local-repair.**

Glossary

admission control --The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability. Quality of service focuses on achieving appropriate network performance for networked applications; it is superior to best effort performance.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications that run on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

VRF --virtual routing and forwarding. An extension of IP routing that provides multiple routing instances. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) device.