Implementing SBC Multi-VRF

The SBC support for multi-VRF (VPN routing and forwarding) on customer edge (CE) devices (that is, customer premise routers) feature provides the capability of suppressing provider edge (PE) checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets. Multi-VRF allows for the use of only one router to accomplish the tasks that multiple routers usually perform. It runs on a network without the requirement of Multiprotocol Label Switching (MPLS) and Border Gateway Protocol (BGP) installed.

Note For a complete description of commands used in this chapter, refer to the *Cisco IOS XR Session Border Controller Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index, or search online.

Feature History for Implementing SBC Multi-VRF

Release	Modification
Release 3.3.0	This feature was introduced on the Cisco XR 12000 Series Router.
Release 3.4.0	No modification.
Release 3.4.1	On the Cisco XR 12000 Series Router, media bypass can occur between different adjacencies. The **media-bypass** command has been deprecated and the **media-bypass-forbid** command has been added.
Release 3.5.0	No modification.
Release 3.6.0	No modification.

Contents

Prerequisites for Implementing Multi-VRF

The following prerequisites are required to implement SBC multi-VRF:

- You must be in a user group associated with a task group that includes the proper task IDs for SBC commands being used. For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of the *Cisco IOS XR System Security Configuration Guide*.
- You must install and activate the package installation envelope (PIE) for the SBC software.

  For detailed information about PIE installation, refer to the *Upgrading and Managing Cisco IOS XR Software* module in the *Cisco IOS XR Getting Started Guide*.
- Before implementing multi-VRF, the SBC must already be created. See the procedures described in the "SBC Configuration Prerequisites" module.

Information About Implementing Multi-VRF

The SBC support for multi-VRF (VPN routing and forwarding) on customer edge (CE) devices (that is, customer premise routers) feature provides the capability of suppressing provider edge (PE) checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets. Multi-VRF allows for the use of only one router to accomplish the tasks that multiple routers usually perform. It runs on a network without the requirement of Multiprotocol Label Switching (MPLS) and Border Gateway Protocol (BGP) installed.

When VRF is used on a router that is not a PE, the checks can be turned off to allow for correct population of the VRF routing table with routes to IP prefixes. Multi-VRF is also important because virtual private network (VPN) functionality is not completely supported on low-end systems. Multi-VRF provides logical separation of routing instances (and by the implication address space) within one router.

The following summarizes the features of multi-VRF.

- Allows a single physical router to be split into multiple virtual routers, where each router contains its own set of interfaces, routing table, and forwarding table. SBC supports multiple (overlapping and independent) routing tables (addressing) per customer. Virtual routing contexts are used to separate routing domains within a single router.
- Multi-VRF can be used where multiple routers are required but only one is available.
- One physical interface can belong to multiple virtual routers through the usage of subinterfaces (Frame Relay, ATM, VLANs).
- BGP and MPLS are not used.
- No connectivity is provided between VRFs (would require using BGP for internal exporting and importing between VRFs).
- When a call is placed between two endpoints in the same VPN site, SBC can route the media directly between them, to reduce network utilization.
- Multi-VRF on SBC provides optimization where both endpoints are on the same VPN by turning media bypass on.

  In Release 3.4.1, by default, all adjacencies on the same VPN have media bypass turned on. Media bypass can be turned off by using the **media-bypass-forbid** command (this command is implemented for CAC policies only).

How to Implement Multi-VRF

Implementing SBC multi-VRF is described in the following sections:

- Configuring Multi-VRF
- Associating an H.323 Adjacency with a VRF
- Associating a SIP Adjacency with a VRF
- Configuring DBE with VRF (Distributed Model Only)

Configuring Multi-VRF

This task configures the router with the SBC running in multi-VRF mode in unified deployment model. Note the relationship between the interface and SBC's SVI, adjacency, and DBE media-address as required.

SUMMARY STEPS

1. **configure**
2. **interface sbc** *number*
3. **vrf** *vrf-name*
4. **ipv4 address** *address prefix*
5. **service-location preferred-active** *node-id* [**preferred-standby** *node-id]*
6. **sbc** *service-name*
7. **service-location preferred-active** *node-id* [**preferred-standby** *node-id]*
8. **sbe**
9. **activate**
10. Associate H.323 adjacency with a VRF (see the "Associating an H.323 Adjacency with a VRF" section).
11. Associate SIP adjacency with a VRF (see the "Associating a SIP Adjacency with a VRF" section).
12. **dbe**
13. **media-address ipv4** *ipv4_address* [**vrf** vrf_name]
14. **activate**
15. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`RP/0/0/CPU0:router# configure`
	`RP/0/0/CPU0:router(config)#`
		Enables global configuration mode.

	Command or Action	Purpose
Step 2	**interface sbc** *number* **Example:** `RP/0/0/CPU0:router(config-if)# interface SBC1`	Enters the mode of an SBC interface, creating it, if necessary. The *number* argument must be a value between 1 and 2000.
Step 3	**vrf** *vrf_name* **Example:** `RP/0/0/CPU0:router(config-if)# vrf my_vrf1`	Ties an H.323 adjacency to a specific virtual private network (VPN).
Step 4	**ipv4 address** *address prefix* **Example:** `RP/0/0/CPU0:router(config-if)# ipv4 address 88.88.101.10 255.255.255.0`	Assigns an IPv4 address to an interface.
Step 5	**service-location preferred-active** *node-id* [**preferred standby** *node-id*] **Example:** `RP/0/0/CPU0:router(config-if)# service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0`	Enables a service card to run SBC function as a primary, and optionally, secondary location.
Step 6	**sbc** *service-name* **Example:** `RP/0/0/CPU0:router(config-if)# sbc lite_8`	Enters the mode of an SBC service. • Use the *service-name* argument to define the name of the service.
Step 7	**service-location preferred-active** *node-id* [**preferred standby** *node-id*] **Example:** `RP/0/0/CPU0:router(config-sbc)# service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0`	Enables a service card to run SBC function as a primary and, optionally, as a secondary location.

Step 8	**sbe**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc)# sbe`
	`RP/0/0/CPU0:router(config-sbc-sbe)#`
		Enters the mode of an SBE entity within an SBC service.

Step 9	**activate** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-acc)# activate`	Initiates the SBC service.
Step 10	Associate H.323 adjacency h323my_vrf1 with vpn1. **Example:** See the "Associating an H.323 Adjacency with a VRF: Example" section.	Associates an H.323 adjacency with a VPN. See the "Associating an H.323 Adjacency with a VRF" section.

	Command or Action	Purpose
Step 11	Associate SIP adjacency sipmy_vrf1 with vpn1. **Example:** See the "Associating a SIP Adjacency with a VRF: Example" section.	Associates a SIP adjacency with a VPN. See the "Associating a SIP Adjacency with a VRF" section.

Step 12	**dbe**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-sbe)# dbe`
	`RP/0/0/CPU0:router(config-sbc-dbe)#`
		Enters the mode of a DBE entity within an SBC service.

Step 13	**media-address ipv4** *ipv4_address* [**vrf** *vrf_name*]
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe)#`
	`media-address`
		Creates an IPv4 address within a DBE media address pool.

Step 14	**activate**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-sbe-acc)#`
	`activate`
		Initiates the SBC service.

Step 15	**end**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-sbe-acc)#`
	`activate`
		Ends the current configuration session.

Associating an H.323 Adjacency with a VRF

This task associates an H.323 adjacency with a VPN.

SUMMARY STEPS

1. **adjacency h323** *adjacency-name*
2. **vrf** *vrf_name*
3. **signaling-address ipv4** *local_signaling_IP_address*
4. **signaling-port** *port_num*
5. **remote-address ipv4** *remote_IP_address/prefix*
6. **signaling-peer** [**gk**] *peer_address*
7. **signaling-peer-port** *port_num*
8. **dtmf-relay rtp-nte**
9. **account** *account_name*
10. **media-bypass** *(Optional command)*
11. **media-bypass-forbid**

12. **attach**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**adjacency h323** *adjacency-name*
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-sbe)# adjacency h323 h323my_vrf1`
	`RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)#`
		Enters the mode of an SBE H.323 adjacency.
		• Use the *adjacency-name* argument to define the name of the service.

Step 2	**vrf** *vrf_name* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# vrf my_vrf1`	Ties an H.323 adjacency to a specific virtual private network (VPN).
Step 3	**signaling-address ipv4** *local_signaling_IP_address* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-address ipv4 88.88.101.11`	Specifies the local IPv4 signaling address of the H.323 adjacency.
Step 4	**signaling-port** *port_num* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-port 1720`	Specifies the local signaling port of the H.323 adjacency.
Step 5	**remote-address ipv4** *ipv4_IP_address/prefix* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# remote-address ipv4 10.10.101.4/32`	Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.
Step 6	**signaling-peer [gk]** *peer_address* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-peer gk 10.10.101.4`	Specifies the remote signaling peer for the H.323 adjacency to use.
Step 7	**signaling-peer-port** *port_num* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# signaling-peer-port 1720`	Specifies the remote signaling-peer port for the H.323 adjacency to use.
Step 8	**dtmf-relay [rtp-nte]** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# dtmf-relay`	Configures DTMF relay for an H.323 adjacency. • Specifying **rtp-nte** enables RFC 2833 *Named Telephone Event* (rtp-nte) support. Not specifying it disables rtp-nte support.

	Command or Action	Purpose
Step 9	**account** *account_name* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# account h323-my_vrf1`	Defines the H.323 adjacency as belonging to an account on an SBE.
Step 10	**media-bypass** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# media-bypass`	(Optional) Configure the adjacency to allow media traffic to bypass the DBE. This command is optional and will only work on one adjacency.
Step 11	**media-bypass-forbid** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# media-bypass-forbid`	(Release 3.4.1) Configures the H.323 adjacency to forbid media traffic to bypass the DBE. If this is not configured, media traffic for calls originating and terminating on this adjacency flows directly between the endpoints and does not pass through the DBE, as long as both adjacencies are on the same VPN.
Step 12	**attach** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# attach`	Attaches the adjacency.

Associating a SIP Adjacency with a VRF

This task associates a SIP adjacency with a VPN.

SUMMARY STEPS

1. **adjacency sip** *adjacency-name*
2. **vrf** *vrf_name*
3. **signaling-address ipv4** *local_signaling_IP_address*
4. **signaling-port** *port_num*
5. **remote-address ipv4** *local_signaling_IP_address/prefix*
6. **local-id host** *name*
7. **signaling-peer** [**gk**] *peer_address*
8. **signaling-peer-port** *port_num*
9. **account** *account-name*
10. **media-bypass** *(optional)*
11. **media-bypass-forbid**
12. **attach**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**adjacency sip** *adjacency-name*
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-sbe)# adjacency sip sipGW`
	`RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)#`
		Enters the mode of an SBE SIP adjacency.
		• Use the *adjacency-name* argument to define the name of the service.

Step 2	**vrf** *vrf_name* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-h323)# vrf my_vrf1`	Ties an H.323 adjacency to a specific virtual private network (VPN).
Step 3	**signaling-address ipv4** *ipv4_IP_address* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.88.88.101.11`	Specifies the local IPv4 signaling address of the SIP adjacency.
Step 4	**signaling-port** *port_num* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-port 5060`	Specifies the local signaling port of the SIP adjacency.
Step 5	**remote-address ipv4** *remote_IP_address/prefix* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.10.101.4/32`	Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.
Step 6	**local-id host** *address* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# local-id host 88.88.101.11`	Configures the local identity name on a SIP adjacency.
Step 7	**signaling-peer [gk]** *peer_address* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-peer 10.10.101.4`	Specifies the remote signaling peer for the SIP adjacency to use.
Step 8	**signaling-peer-port** *port_num* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060`	Specifies the remote signaling-peer port for the SIP adjacency to use.

	Command or Action	Purpose
Step 9	**account** *account_name* **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# account sipmy_vrf2`	Defines the SIP adjacency as belonging to an account on an SBE.
Step 10	**media-bypass** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# media-bypass`	(Optional) Configures the adjacency to allow media traffic to bypass the DBE. This command is optional and only works on one adjacency.
Step 11	**media-bypass-forbid** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# media-bypass-forbid`	(Release 3.4.1) Configures the SIP adjacency to forbid media traffic to bypass the DBE. If this is not configured, media traffic for calls originating and terminating on this adjacency flows directly between the endpoints and does not pass through the DBE, as long as both adjacencies are on the same VPN.
Step 12	**attach** **Example:** `RP/0/0/CPU0:router(config-sbc-sbe-adj-sip)# attach`	Attaches the adjacency.

Configuring DBE with VRF (Distributed Model Only)

This task configures DBE with VRF in the Distributed Model.

SUMMARY STEPS

1. **configure**
2. **interface sbc** *number*
3. **vrf** *vrf-name*
4. **ipv4 address** *address prefix*
5. **service-location preferred-active** *node-id* [**preferred-standby** *node-id]*
6. **interface sbc** *number*
7. **ipv4 address** *address prefix*
8. **service-location preferred-active** *node-id* [**preferred-standby** *node-id]*
9. **sbc** *service-name*
10. **service-location preferred-active** *node-id* [**preferred-standby** *node-id]*
11. **dbe**
12. **vdbe** [**global**]
13. **control-address h248 ipv4** *IPv4_H.248_IP_address*
14. **controller h248** *controller-index*

15. **transport** [**udp** | **tcp**]
16. **remote-address ipv4** *remote-address*
17. **attach-controllers**
18. **media-address ipv4** *ipv4_address* [**vrf** *vrf_name*]
19. **activate**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure**
	
	Example:
	`RP/0/0/CPU0:router# configure`
	`RP/0/0/CPU0:router(config)#`
		Enables global configuration mode.

Step 2	**interface sbc***number*
	
	Example:
	`RP/0/0/CPU0:router(config)# interface SBC1`
	`RP/0/0/CPU0:router(config-if)#`
		Enters the mode of an SBC interface, creating it if necessary. The *number* argument must be a value between 1 and 2000.

Step 3	**vrf** *vrf_name* **Example:** `RP/0/0/CPU0:router(config-if)# vrf my_vrf1`	Ties an H.323 adjacency to a specific virtual private network (VPN).
Step 4	**ipv4 address** *address prefix* **Example:** `RP/0/0/CPU0:router(config-if)# ipv4 address 88.88.130.10 255.255.255.0`	Assigns an ipv4 address to an interface.
Step 5	**service-location preferred-active** *node-id* [**preferred standby** *node-id*] **Example:** `RP/0/0/CPU0:router(config-if)# service-location preferred-active 0/1/CPU0`	Enables a service card to run SBC function as a primary, and optionally, secondary location.
Step 6	**interface sbc** *number* **Example:** `RP/0/0/CPU0:router(config-if)# interface SBC2`	Enters the mode of an SBC interface, creating it if necessary. The *number* argument must be a value between 1 and 2000.
Step 7	**ipv4 address** *address prefix* **Example:** `RP/0/0/CPU0:router(config-if)# ipv4 address 88.88.130.10 255.255.255.0`	Assigns an ipv4 address to an interface.

	Command or Action	Purpose
Step 8	**service-location preferred active** *node-id* [**preferred standby** *node-id*] **Example:** `RP/0/0/CPU0:router(config-if)# service-location preferred-active 0/1/CPU0`	Enables a service card to run SBC function as a primary, and optionally, secondary location.

Step 9	**sbc** *service-name*
	
	Example:
	`RP/0/0/CPU0:router(config-if)# sbc lite_1`
	`RP/0/0/CPU0:router(config-sbc)#`
		Enters the mode of an SBC service.
		• Use the *service-name* argument to define the name of the service.

Step 10	**service-location preferred active** *node-id* [**preferred standby** *node-id*]
	
	Example:
	`RP/0/0/CPU0:router(config-sbc)#`
	`service-location preferred-active 0/1/CPU0`
		Enables a service card to run SBC function as a primary, and optionally, secondary location.

Step 11	**dbe**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc)# dbe`
	`RP/0/0/CPU0:router(config-sbc-dbe)#`
		Enters the mode for configuring a DBE entity within an SBC service.

Step 12	**vdbe [global]**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe)# vdbe`
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#`
		Enters the mode for configuring virtual DBE parameters.

Step 13	**control-address h248 ipv4** *ipv4_H.248_IP_address*
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#`
	`control-address h248 ipv4 88.88.130.100`
		Configures a DBE to use a given IPv4 H.248 control address.

Step 14	**controller h248** *controller-index*
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#`
	`controller h248 1`
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)#`
		Enters the mode for configuring an H.248 controller for a DBE.

Step 15	**transport [udp | tcp]**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)#`
	`transport udp`
		Configures a DBE to use either UDP or TCP for H.248 control signaling with the H.248 controller.

	Command or Action	Purpose
Step 16	**remote-address ipv4** *remote-address* **Example:** `RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)# remote-address ipv4 88.88.101.21`	Defines the remote address to connect to on the SBE for an H.248 controller.

Step 17	**attach-controllers**
	
	Example:
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe-h248)# attach-controllers`
	`RP/0/0/CPU0:router(config-sbc-dbe-vdbe)#`
		Configures a DBE to attach to a controller.

Step 18	**media-address ipv4** *ipv4_address* [**vrf** *vrf_name*] **Example:** `RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# media-address ipv4 88.88.130.2 vrf my_vrf`	Enters the mode for configuring a DBE media address pool and creates an address pool for use in a VRF.
Step 19	**activate** **Example:** `RP/0/0/CPU0:router(config-sbc-dbe-vdbe)# activate`	Initiates the SBC service.

Configuration Examples for Implementing Multi-VRF

This section provides the following configuration examples:

- Configuring Multi-VRF: Example
- Associating an H.323 Adjacency with a VRF: Example
- Associating a SIP Adjacency with a VRF: Example
- Configuring DBE with VRF (Distributed Model Only): Example

Configuring Multi-VRF: Example

This sample configuration shows how the Service Virtual Interface (SVI) and adjacencies are added to associate a VPN to them. When a call is made from callgen7, sbc1 takes it and bridges it to sbc2, then goes to callgen8.

Note This configuration has the adjacencies for the gateways and number analysis removed.

1. Configure the line card interface associated with my_vrf2:

```
cdp
 vrf my_vrf2
 ipv4 address 192.168.229.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/3/0/1
```

```
 description vrf-lite 1 to node1
```

2. Configure the line card interface associated with my_vrf1:

```
cdp
 vrf my_vrf1
 ipv4 address 1.0.4.1 255.255.255.0
 negotiation auto
!
interface SBC1
 vrf my_vrf1
```

3. Configure the SVI associated with my_vrf1:

```
ipv4 address 88.88.101.10 255.255.255.0
 service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
!
interface SBC2
 vrf my_vrf2
```

4. Configure the SVI associated with my_vrf2:

```
ipv4 address 88.88.129.10 255.255.255.0
 service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
!
!
sbc lite_8
 service-location preferred-active 0/4/CPU0 preferred-standby 0/5/CPU0
 sbe
  activate
  adjacency h323 h323my_vrf1
   vrf my_vrf1
```

5. Configure adjacency associated with my_vrf1 for H.323:

```
signaling-address ipv4 88.88.101.11
   signaling-port 1720
   remote-address ipv4 10.10.101.4/32
   signaling-peer 10.10.101.4
   signaling-peer-port 1720
   dtmf-relay
   account h323-my_vrf1
   attach
  !
  adjacency h323 h323my_vrf2
   vrf my_vrf2
```

6. Configure adjacency associated with my_vrf2 for H.323:

```
signaling-address ipv4 88.88.129.11
   signaling-port 1720
   remote-address ipv4 10.10.115.4/32
   signaling-peer 10.10.115.4
   signaling-peer-port 1720
   dtmf-relay
   account h323-my_vrf2
   attach
  !
  adjacency sip sipmy_vrf1
   vrf my_vrf1
```

7. Configure adjacency associated with my_vrf1 for SIP:

```
signaling-address ipv4 88.88.101.11
   signaling-port 5060
   remote-address ipv4 10.10.101.4/32
```

```
   local-id host 88.88.101.11
   signaling-peer 10.10.101.4
   signaling-peer-port 5060
   account sip-my_vrf1
   attach
  !
  adjacency sip sipmy_vrf2
   vrf my_vrf2
```

8. Configure adjacency associated with my_vrf2 for SIP:

```
signaling-address ipv4 88.88.129.11
   signaling-port 5060
   remote-address ipv4 10.10.115.4/32
   local-id host 88.88.129.11
   signaling-peer 10.10.115.4
   signaling-peer-port 5060
   account sip-my_vrf2
   attach
  !
!
 dbe
  media-address ipv4 88.88.101.2 vrf my_vrf1
```

9. Configure the media-address associated with my_vrf1:

```
  media-address ipv4 88.88.129.2 vrf my_vrf2
```

10. Configure the media-address associated with my_vrf2:

```
  activate
 !
!
end
```

Associating an H.323 Adjacency with a VRF: Example

This sample configuration creates an H.323 adjacency associated with a VPN.

```
adjacency h323 h323my_vrf1
   vrf my_vrf1
   signaling-address ipv4 88.88.101.11
   signaling-port 1720
   remote-address ipv4 10.10.101.4/32
   signaling-peer 10.10.101.4
   signaling-peer-port 1720
   dtmf-relay
   account h323-my_vrf1
   attach
```

Associating a SIP Adjacency with a VRF: Example

This example configuration creates a SIP adjacency associated with a VPN.

```
adjacency sip sipmy_vrf1
   vrf my_vrf1
   signaling-address ipv4 88.88.101.11
   signaling-port 5060
   remote-address ipv4 10.10.101.4/32
   local-id host 88.88.101.11
   signaling-peer 10.10.101.4
```

```
signaling-peer-port 5060
account sip-my_vrf1
attach
```

Configuring DBE with VRF (Distributed Model Only): Example

This example shows how to configure DBE with VRF in the Distributed Model.

Only the media address can be in my_vrf1, control address signaling must be in the default VRF. Like the Unified Model's SBE and DBE, the media address is associated with the SVI and a physical interface in my_vrf1.

Note A second SVI is needed to be in the default VRF for the control signaling.

```
interface SBC1
vrf my_vrf1
ipv4 address 88.88.130.10 255.255.255.0
service-location preferred-active 0/1/CPU0
!
interface SBC2
ipv4 address 88.88.130.10 255.255.255.0
service-location preferred-active 0/1/CPU0
!
sbc lite_1
service-location preferred-active 0/1/CPU0
dbe
vdbe
control-address h248 ipv4 88.88.130.100
controller h248 1
transport udp
remote-address ipv4 88.88.101.21
!
attach-controllers
!
media-address ipv4 88.88.130.2 vrf my_vrf1
activate
```

Additional References

The following sections provide references related to implementing SBC multi-VRF.

Related Documents

Related Topic	Document Title
Cisco IOS XR master command reference	Cisco IOS XR Master Commands List
Cisco IOS XR SBC interface configuration commands	*Cisco IOS XR Session Border Controller Command Reference*

Related Topic	Document Title
Initial system bootup and configuration information for a router using the Cisco IOS XR Software	*Cisco IOS XR Getting Started Guide*
Cisco IOS XR command modes	*Cisco IOS XR Command Mode Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support from existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2685	*Virtual Private Networks Identifier*
RFC 1918	*Address Allocation for Private Internets*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Related Command Summary

This section provides an alphabetical list of the related commands to configure SBC Multi-VRF on the Cisco XR 12000 Series Router. For more information about the commands, see the *Cisco IOS XR Session Border Controller Command Reference.*

Command	Purpose
interface sbc *number*	Enters the mode of an SBC interface, creating it if necessary. The *number* argument must be a value between 1 and 2000.
ipv4 address *address prefix*	Assigns an ipv4 address to an interface.
media-address ipv4 *ipv4_address* [**vrf** *vrf_name*]	Creates an ipv4 address within a DBE media address pool.
service-location preferred active *node-id* [**preferred standby** *node-id*]	Enables a service card to run SBC function as a primary, and optionally, secondary location.
vdbe [**global**]	Enters the mode for configuring virtual DBE parameters.
vrf *vrf_name*	Ties an H.323 adjacency to a specific virtual private network (VPN).