Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)

Cisco Unified Border Element (SP Edition) provides support for multi-VRF (VPN routing and forwarding) on customer edge (CE) devices. This feature provides the capability of suppressing provider edge (PE) checks to prevent loops when the PE is performing a mutual redistribution of packets.

VRF is only supported in DBE media address and SBE AAA/H248 control address; DBE H248 control address does not support VRF.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html.

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Note For Cisco IOS XE Release 2.4, this feature is supported in both the unified and distributed model.

Feature History for Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)

Release	Modification
Cisco IOS XE Release 2.4	This feature was introduced on the Cisco ASR 1000 Series Routers.
Cisco IOS XE Release 3.2S	SBC Voice traffic support over tunnel-interface (GRE, IPSec, MPLS, TE tunnel, BBA) was introduced on the Cisco ASR 1000 Series Routers.

Contents

This module contains the following sections:

Prerequisites—Implementing Multi-VRF

The following prerequisite is required to implement multi-VRF on Cisco Unified Border Element (SP Edition):

- Before implementing multi-VRF, Cisco Unified Border Element (SP Edition) must already be configured.

Information About Implementing Multi-VRF

Cisco Unified Border Element (SP Edition) support for multi-VRF on customer edge (CE) devices, such as customer premises routers, provides the capability of suppressing 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 MPLS and 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. Cisco Unified Border Element (SP Edition) 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.
- When using multi-VRF, the domain name server (DNS) queries are per VRF.
- One physical interface can belong to multiple virtual routers through the use 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, Cisco Unified Border Element (SP Edition) can route the media directly between them, to reduce network utilization.
- Multi-VRF on Cisco Unified Border Element (SP Edition) provides optimization where both endpoints are on the same VPN by turning media bypass on.
- When a VRF is removed from a SBC interface that is in use by an activated SBC, the IP addresses are not removed automatically by the SBC. The user has to manually remove the IP addresses when the SBC is deactivated.

For Cisco IOS XE Release 2.4, 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).

Note The vrf name under the adjacency must match the context name.

Note Media termination occurs prior to route leaking, therefore media cannot be terminated on leaked routes.

Implementing Multi-VRF

Implementing multi-VRF is described in the following sections:

Associating a SIP Adjacency with a VRF

This task associates a SIP adjacency with a VPN.

Note When an adjacency is assigned to a particular VRF, all the addresses relating to the adjacencies, such as signalling-address and remote-address, must also be routable within the VRF.

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** *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:** `Router(config-sbc-sbe)# adjacency sip sip_vrf1`	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:** `Router(config-sbc-sbe-adj-sip)# vrf my_vrf1`	Ties a SIP adjacency to a specific VPN. **Note** The vrf name under the adjacency must match the context name.
Step 3	**signaling-address ipv4** *ipv4_IP_address* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.88.88`	Specifies the local IPv4 signaling address of the SIP adjacency.
Step 4	**signaling-port** *port_num* **Example:** `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:** `Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.10.101.4 255.255.255.255`	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:** `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** *peer_address* **Example:** `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:** `Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060`	Specifies the remote signaling-peer port for the SIP adjacency to use.
Step 9	**account** *account_name* **Example:** `Router(config-sbc-sbe-adj-sip)# account sip-vrf1`	Defines the SIP adjacency as belonging to an account on an SBE.

	Command or Action	Purpose
Step 10	**media-bypass** **Example:** `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:** `Router(config-sbc-sbe-adj-sip)# media-bypass-forbid`	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:** `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. **sbc** ***sbc-name*** **dbe**
3. **vdbe** *global*
4. **unexpected-source-alerting**
5. **local-port** *abcd*
6. **control-address h248 ipv4** ***A.B.C.D***
7. **controller h248** *controller-index*
8. **remote-address** ipv4 *remote-address*
9. **remote-port** [*port-num*]
10. **transport** [**udp** | **tcp**]
11. **attach-controllers**
12. **media-address pool ipv4** *A.B.C.D E.F.G.H* **vrf** *vrfname*
13. **media-timeout timeout**
14. **overload-time-threshold** *time*
15. **deactivation-mode**
16. **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `Router# configure`	Accesses the configuration mode.
Step 2	***sbc*** *sbc-name* ***dbe*** ***Example:*** *Router(config)#* ***sbc mySbc***	Creates the DBE service on the SBC and enters into SBC-DBE configuration mode.
Step 3	**vdbe [global]** **Example:** `Router(config-sbc-dbe)# vdbe`	Enters into vDBE configuration submode. **Note** In the initial release only one vDBE (the global vDBE) is supported. The vdbe name is not required. If specified, it must be global.
Step 4	**unexpected-source-alerting** **Example:** `Router(config-sbc-dbe-vdbe-global)# unexpected-source-alerting`	Sets alerting for unexpected source addresses. The **no** form of this command removes alerting for any unexpected source addresses that are received.
Step 5	**local-port** *{abcd}* **Example:** `Router(config-sbc-dbe)# local-port 5090`	Configures a DBE to use a specific local port.
Step 6	**control-address h248 ipv4** *A.B.C.D* **Example:** `Router(config-sbc-dbe)# control-address h248 ipv4 10.0.0.1`	Configures a DBE to use a specific IPv4 H.248 control address. **Note** The control address cannot be in a VRF and must be routable in the global address table.
Step 7	***controller h248*** *controller-index* **Example:** `Router(config-sbc-dbe)# controller h248 1`	Identifies the H.248 controller for the DBE and enters into Controller H.248 configuration mode.
Step 8	**remote-address ipv4** *remote-address* **Example:** `Router(config-sbc-dbe-vdbe-h248)# remote-address ipv4 1.1.1.1`	Configures the IPv4 remote address of the H.248 controller.
Step 9	**remote-port [*port-num*]** **Example:** `Router(config-sbc-dbe-h248)# remote-port 2094`	Defines the port to connect to on the SBE for an H.248 controller.

	Command or Action	Purpose
Step 10	**transport udp** **Example:** `Router(config-sbc-dbe-h248)# transport udp`	Configures a DBE to use User Datagram Protocol (UDP) for H.248 control signaling.
Step 11	**attach-controllers** **Example:** `Router(config-sbc-dbe)# attach-controllers`	Configure a DBE to attach to an H.248 controller.
Step 12	***media-address pool ipv4*** *A.B.C.D E.F.G.H* ***vrf*** *vrfname* **Example:** `Router(config-sbc-dbe)# media-address pool ipv4 10.10.10.1 10.10.10.20 vrf my_vrf1`	Create a pool of sequential IPv4 media addresses for an IPv4 address associated with a specific VRF instance. **Note** The vrf name under the adjacency must match the context name.
Step 13	**media-timeout timeout** **Example:** `Router(config-sbc-dbe)# media-timeout 10`	Sets the maximum time a DBE waits after receiving the last media packet on a call and before cleaning up the call resources.
Step 14	**overload-time-threshold** ***time*** **Example:** `Router(config-sbc-dbe)# overload-time-threshold 400`	Configures the threshold for media gateway (MG) overload control detection.
Step 15	**deactivation-mode normal** **Example:** `Router(config-sbc-dbe)# deactivation-mode normal`	Specifies that the DBE of an SBC signals a service change and terminates all calls upon deactivation of the DBE service.
Step 16	**activate** **Example:** `Router(config-sbc-dbe)# activate`	Initiates the SBC service.

Configuration Examples for Implementing Multi-VRF

This section provides the following configuration examples:

Configuring SBC Unified Model with VRF: Example

You can configure the Cisco ASR 1000 Series Router so that traffic is routed to the SBC adjacency address. This is achieved by creating a VRF instance on the router.

The following is an example, which uses VLAN trunks to get the traffic into the SBC. In this example, a VRF is created to route traffic from the 100.0.0.0/24 network to the 12.0.0.0/24 network, where the SIP signaling address and media address reside for a particular SBC connection.

The **interface sbc** command is needed, whenever a VRF is being used. You must have a secondary IP address defined if the media IP address is going to be different than the signaling address. However, in this case the secondary IP address is automatically added when the **media-address ipv4** command is used. It must not be manually entered.

```
vrf definition cust100side// Create a VRF instance
 !
 address-family ipv4
 exit-address-family
interface SBC100// Create an interface in the VRF space
 vrf forwarding cust100side
 ip address 12.0.0.30 255.255.255.0 secondary// This contains the IP address for the
                                             // media, if different to the signaling
                                             // address. The line is not entered, but
                                             // appears automatically after the DBE
                                             // configuration is entered (see
                                             // 'media-address' CLI later.)
 ip address 12.0.0.20 255.255.255.0           // This is the SIP adjacency address

interface GigabitEthernet0/1/0
  no ip address
 media-type sfp
 negotiation auto

interface GigabitEthernet0/1/0.100            // VLAN identifier 100 defined here
 vrf forwarding cust100side
 encapsulation dot1Q 100
 ip address 100.0.0.1 255.255.255.0        // This IP is where the remote side or external
                                           // router can send traffic to, in order to get
                                           // to the internal 12.0.0.0/24 network
interface GigabitEthernet0/1/0.200        // Other VLANS that are being trunked.
 vrf forwarding cust200side
 encapsulation dot1Q 200
 ip address 200.0.0.1 255.255.255.0

sbc ted
  sbe
    adjacency sip adj_cust100
      vrf cust100side
      ...
      signaling-address 12.0.0.20         // This is the local address where call traffic
                                          // will get routed to/from
      remote-address ipv4 100.0.0.14      // This is an address for the remote side, where
                                          // traffic will be routed
      ...
      attach
  ...
  media-address ipv4 12.0.0.30 vrf cust100side// The media address is also on the
                                              // internal network. When the line
                                              // is entered, the interface SBC
                                              // will show a secondary address
                                              // containing this IP address.
  activate
```

Configuring Multi-VRF: Example

This sample configuration shows how the Service Virtual Interface (SVI) and adjacencies are added to associate a VPN to them.

1. Configure the line card interface associated with vrf my_vrf1 on the route processor (RP).

```
vrf definition my_vrf1
rd 55:1111
!
address-family ipv4
exit-address-family
!
```

2. Configure the line card interface associated with vrf, my_vrf1, on the route processor.

```
interface GigabitEthernet1/3
description ''Connected to CAT-3550-101 Fa 0/13 vlan919''
ip address 10.122.3.3 255.255.255.0

interface GigabitEthernet1/3.99
  encapsulation dot1q 99
  vrf forwarding my_vrf1
  ip address 10.122.3.3 255.255.255.0
!
```

3. Configure the media address pools.

```
media-address pool ipv4 88.88.101.12 88.88.101.15 vrf my_vrf1 activate
```

Associating a SIP Adjacency with a VRF: Example

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

```
ip route 10.10.0.0 255.255.0.0 101.101.101.100 ip route 20.20.20.0 255.255.255.0
101.101.101.4

domain default-domain

sbc mysbc
  sbe
    adjacency sip 7200-1
      vrf my_vrf1
      inherit profile preset-core
      preferred-transport udp
      redirect-mode pass-through
      authentication nonce timeout 300
      signaling-address ipv4 101.101.101.3
      signaling-port 5061
      remote-address ipv4 0.0.0.0 0.0.0.0
      signaling-peer 101.101.101.5
      signaling-peer-port 5060
      account sip-core
      attach

    adjacency sip 7200-2
      vrf my_vrf1
      inherit profile preset-access
      preferred-transport udp
      redirect-mode pass-through
      authentication nonce timeout 300
      signaling-address ipv4 101.101.101.3
```

```
  signaling-port 5060
  remote-address ipv4 0.0.0.0 0.0.0.0
  signaling-peer 101.101.101.4
  signaling-peer-port 5060
  account sip-core
  attach

adjacency sip 7200-3
  vrf my_vrf1
  nat force-on
  inherit profile preset-core
  preferred-transport udp
  redirect-mode pass-through
  authentication nonce timeout 300
  signaling-address ipv4 101.101.101.3
  signaling-port 5063
  remote-address ipv4 0.0.0.0 0.0.0.0
  signaling-peer 101.101.101.5
  signaling-peer-port 5063
  account sip-core
  reg-min-expiry 3000
  attach

sip inherit profile preset-standard-non-ims

retry-limit 3

call-policy-set 1
  first-call-routing-table invite-table
  first-reg-routing-table start-table
  rtg-src-adjacency-table invite-table
    entry 1
      action complete
      dst-adjacency 7200-2
      match-adjacency 7200-3
    entry 2
      action complete
      dst-adjacency 7200-3
      match-adjacency 7200-2
  rtg-src-adjacency-table start-table
    entry 1
      action complete
      dst-adjacency 7200-1
      match-adjacency 7200-2
    entry 2
      action complete
      dst-adjacency 7200-2
      match-adjacency 7200-1
  complete

active-call-policy-set 1

network-id 2

sip max-connections 2
sip timer
  tcp-idle-timeout 120000
  tls-idle-timeout 3600000
  udp-response-linger-period 32000
  udp-first-retransmit-interval 500
  udp-max-retransmit-interval 4000
  invite-timeout 180

blacklist
```

```
      global

    redirect-limit 2
    deactivation-mode normal
    activate

  media-address ipv4 101.101.101.160 vrf my_vrf1 port-range 11000 20000 any
  location-id 0
  media-timeout 30
  deactivation-mode normal
  activate
```

Configuring DBE with Multi-VRF (Distributed Model Only): Example

To make use of Multi-VRF when Cisco Unified Border Element (SP Edition) is running in the distributed mode, both the configuration and the corresponding H.248 messages are required to be VRF-aware.

The following sample configuration creates media pool that is tied to a particular VRF. This media pool can only be used to assign media addresses for that particular VRF and can overlap with addresses from different VRF's or from the global address space.

```
vrf definition moon
 vpn id 22AA:33334411
!
interface SBC1
  ip vrf forwarding moon
  ip address 90.0.0.1 255.0.0.0
!

sbc global dbe
vdbe global
 h248-version 3
 h248-napt-package napt
 local-port 2979
 control-address h248 ipv4 200.50.1.9
 controller h248 1
  remote-address ipv4 200.50.1.254
  remote-port 2979
 attach-controllers
location-id 1
media-address ipv4 90.0.0.1 vrf moon
 port-range 10000 20000 any

activate
!
```

The H.248 configuration is specified in the H.248 package/Extended VPN Discrimination/ (EVPND). This package has two methods, GVPNID and VRF_NAME, of specifying to which VRF the media addresses belong. These parameters are mutually exclusive but they are independent on a per side basis. For example, side A may use the VRF_NAME method for specifying the VRF and side B may use the GVPNID method.

The VRF_NAME is a quoted ASCII string corresponding to the name of the VRF in the configuration. In the following example, the name would be "moon."

```
M {
                        TS { SI = IV },
                        ST = 1 {
                            O { MO = IN,
                                EVPND/VRF_NAME = "moon"
                              },
                            R {
                                v=0
                                c=IN IP4 3.0.0.3
                                m=application 5000 udp 0
                            },
                            L {
                                v=0
                                c=IN IP4 $
                                m=application $ udp 0
                            }
                        }
                    }
```

The GVPNID is the identification number for the VRF in RFC2685 format. This is specified in the configuration as follows:

```
vrf definition moon
 vpn id 22AA:33334411
!
```

The H.248 format is then specified as:

```
                    M {
                        TS { SI = IV },
                        ST = 1 {
                            O { MO = IN,
                                EVPND/GVPNID = 22AA33334411
                              },
                            R {
                                v=0
                                c=IN IP4 3.0.0.3
                                m=application 5000 udp 0
                            },
                            L {
                                v=0
                                c=IN IP4 $
                                m=application $ udp 0
                            }
                        }
                    }
```

Supporting the SBC Voice Traffic over Tunnel Interfaces

The Cisco IOS XE Release earlier than Cisco IOS XE Release 3.2S did not support the SBC traffic over the tunnel interfaces. The Cisco IOS XE Release 3.2S provides support to the SBC traffic over the tunnel interfaces (PPPoE, GRE, MPLS-TE, IPsec SVTI or DVTI, DMVPN). The following topology diagrams (Figure 5-1 and Figure 5-2) illustrate the broadband deployment scenario and tunnel interface scenarios in which the SBC voice traffic is supported over the tunnel interfaces:

Figure 5-1 Broadband Deployment Topology Supporting the SBC Traffic

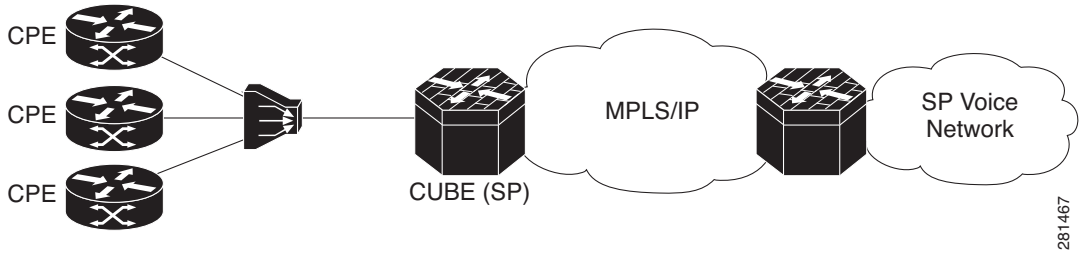

Figure 5-2 IPSec Tunnel Deployment Topology Supporting the SBC Traffic