Implementing MPLS Layer 3 VPNs

A Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers.

This module provides the conceptual and configuration information for MPLS Layer 3 VPNs on Cisco NCS 5000 Series Routers.

Note You must acquire an evaluation or permanent license in order to use MPLS Layer 3 VPN functionality. For more information about licenses, see the module in the *System Management Configuration Guide for Cisco NCS 5000 Series Routers*.

For a complete description of the commands listed in this module, refer these command references:

- BGP
- MPLS
- Routing
- VPN and Ethernet Services

This chapter includes topics on:

MPLS L3VPN Overview

Before defining an MPLS VPN, VPN in general must be defined. A VPN is:

- An IP-based network delivering private network services over a public infrastructure

- A set of sites that are allowed to communicate with each other privately over the Internet or other public or private networks

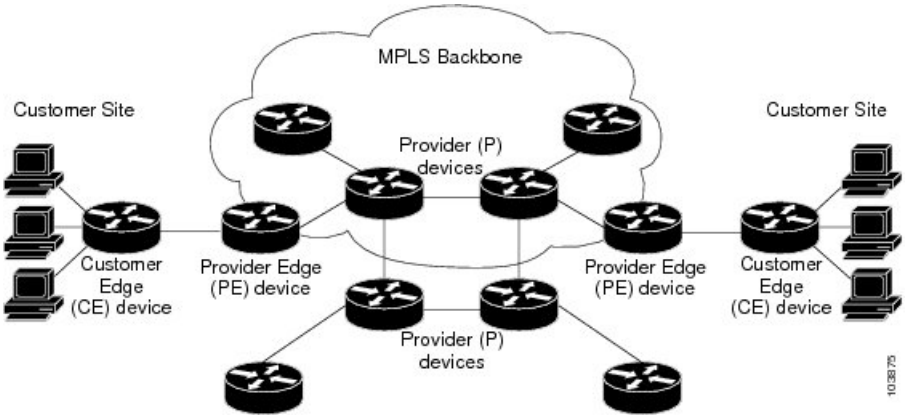
Conventional VPNs are created by configuring a full mesh of tunnels or permanent virtual circuits (PVCs) to all sites in a VPN. This type of VPN is not easy to maintain or expand, as adding a new site requires changing each edge device in the VPN.

MPLS-based VPNs are created in Layer 3 and are based on the peer model. The peer model enables the service provider and the customer to exchange Layer 3 routing information. The service provider relays the data between the customer sites without customer involvement.

MPLS VPNs are easier to manage and expand than conventional VPNs. When a new site is added to an MPLS VPN, only the edge router of the service provider that provides services to the customer site needs to be updated.

The following figure depicts a basic MPLS VPN topology.

Figure 1: Basic MPLS VPN Topology



These are the basic components of MPLS VPN:

- Provider (P) router—Router in the core of the provider network. P routers run MPLS switching and do not attach VPN labels to routed packets. VPN labels are used to direct data packets to the correct private network or customer edge router.
- PE router—Router that attaches the VPN label to incoming packets based on the interface or sub-interface on which they are received, and also attaches the MPLS core labels. A PE router attaches directly to a CE router.
- Customer (C) router—Router in the Internet service provider (ISP) or enterprise network.
- Customer edge (CE) router—Edge router on the network of the ISP that connects to the PE router on the network. A CE router must interface with a PE router.

How MPLS L3VPN Works

MPLS VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following tasks:

- Exchanges routing updates with the CE router

- Translates the CE routing information into VPN version 4 (VPNv4) routes
- Exchanges VPNv4 routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

Major Components of MPLS L3VPN

An MPLS-based VPN network has three major components:

- VPN route target communities—A VPN route target community is a list of all members of a VPN community. VPN route targets need to be configured for each VPN community member.
- Multiprotocol BGP (MP-BGP) peering of the VPN community PE routers—MP-BGP propagates VRF reachability information to all members of a VPN community. MP-BGP peering needs to be configured in all PE routers within a VPN community.
- MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN service-provider network.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes available to the site from the VPNs of which it is a member.

Read more at Major Components of MPLS L3VPN—Details, on page 35.

Restrictions for MPLS L3VPN

Implementing MPLS L3VPN in Cisco NCS 5000 Series Routers is subjected to these restrictions:

- Fragmentation of MPLS packets that exceed egress MTU is not supported. Fragmentation is not supported for IP->MPLS imposition as well. Hence, it is recommended to use Maximum MTU (9216) value on all interfaces in the MPLS core.
- L3VPN prefix lookup always yields a single path. In case of multiple paths at IGP or BGP level, path selection at each level is done using the prefix hash in control plane. The selected path is programmed in the data plane.
- TTL propagation cannot be disabled. TTL propagation always happens from IP->MPLS and MPLS->IP.

Apart from the specific ones mentioned above, these generic restrictions for implementing MPLS L3VPNs also apply for Cisco NCS 5000 Series Routers:

The following restrictions apply when configuring MPLS VPN Inter-AS with ASBRs exchanging IPv4 routes and MPLS labels:

- For networks configured with eBGP multihop, a label switched path (LSP) must be configured between non adjacent routers.
- Inter-AS supports IPv4 routes only. IPv6 is not supported.

Note The physical interfaces that connect the BGP speakers must support FIB and MPLS.

How to Implement MPLS Layer 3 VPNs

Implementing MPLS L3VPNs involves these main tasks:

Prerequisites for Implementing MPLS L3VPN

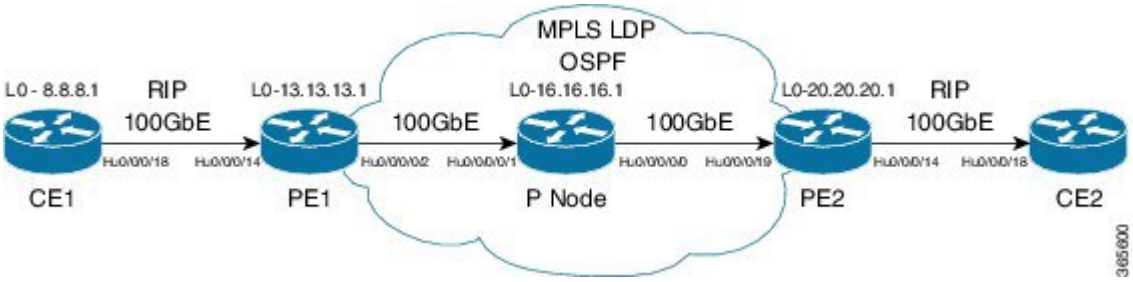
These are the prerequisites to configure MPLS L3VPN:

- You must be in a user group associated with a task group that includes the proper task IDs for these commands:
  - BGP
  - IGP
  - MPLS
  - MPLS Layer 3 VPN
- If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- To configure MPLS Layer 3 VPNs, routers must support MPLS forwarding and Forwarding Information Base (FIB).

Configure the Core Network

Consider a network topology where MPLS L3VPN services are transported over MPLS LDP core.

Figure 2: L3VPN over MPLS LDP



Configuring the core network involves these main tasks:

Assess the Needs of MPLS VPN Customers

Before configuring an MPLS VPN, the core network topology must be identified so that it can best serve MPLS VPN customers. The tasks listed below helps to identify the core network topology.

- Identify the size of the network:

  Identify the following to determine the number of routers and ports required:

  - How many customers to be supported?

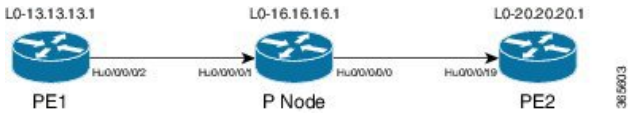
- How many VPNs are required for each customer?
- How many virtual routing and forwarding (VRF) instances are there for each VPN?

- Determine the routing protocols required in the core.
- Determine if BGP load sharing and redundant paths in the MPLS VPN core are required.

Configure Routing Protocols in the Core

You can use RIP, OSPF or IS-IS as the routing protocol in the core.

Figure 3: OSPF as Routing Protocol in the Core



Configuration Example

This example lists the steps to configure OSPF as the routing protocol in the core.

```
Router-PE1#configure
Router-PE1(config)#router ospf dc-core
Router-PE1(config-ospf)#address-family ipv4 unicast
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#interface HundredGigE0/0/0/2
Router-PE1(config-ospf-ar-if)#commit
```

Running Configuration

```
router ospf dc-core
 router-id 13.13.13.1
 address-family ipv4 unicast
 area 1
  interface HundredGigE0/0/0/2
  !
 !
!
```

Verification

- Verify the OSPF neighbor and ensure that the *State* is displayed as 'FULL'.

```
Router-PE1# show ospf neighbor
Neighbors for OSPF dc-core

Neighbor ID     Pri   State           Dead Time   Address         Interface
16.16.16.1      1     FULL/-  00:00:34    191.22.1.2      HundredGigE0/0/0/2
    Neighbor is up for 1d18h

Total neighbor count: 1
```

Related Topics

- How to Implement MPLS Layer 3 VPNs, on page 4

For more details on configuring the routing protocol, see *Routing Configuration Guide for Cisco NCS 5000 Series Routers* and *BGP Configuration Guide for Cisco NCS 5000 Series Routers*.

Associated Commands

- router-id
- router ospf

Configure MPLS in the Core

To enable MPLS on all routers in the core, you must configure a Label Distribution Protocol (LDP).

You can also transport MPLS L3VPN services using segment routing in the core. For details, see Configure Segment Routing in MPLS Core, on page 30.

Configuration Example

This example lists the steps to configure LDP in MPLS core.

```
Router-PE1#configure
Router-PE1(config)#mpls ldp
Router-PE1(config-ldp)#router-id 13.13.13.1
Router-PE1(config-ldp)#address-family ipv4
Router-PE1(config-ldp-af)#exit
Router-PE1(config-ldp)#interface HundredGigE0/0/0/2
Router-PE1(config-ldp-if)#commit
```

Repeat this configuration in PE2 and P routers as well.

Running Configuration

```
mpls ldp
 router-id 13.13.13.1
 address-family ipv4
 !
 interface HundredGigE0/0/0/2
 !
!
```

Verification

- Verify that the neighbor (16.16.16.1) is UP through the core interface:

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
```

```
LDP Discovery Sources:
  IPv4: (1)
    HundredGigE0/0/0/2
  IPv6: (0)
Addresses bound to this peer:
  IPv4: (6)
    10.64.98.32    87.0.0.2       88.88.88.14    50.50.50.50
    178.0.0.1      192.1.1.1
  IPv6: (0)
```

Related Topics

- How to Implement MPLS Layer 3 VPNs, on page 4

For more details on configuring MPLS LDP, see the *Implementing MPLS Label Distribution Protocol* chapter in the *MPLS Configuration Guide for Cisco NCS 5000 Series Routers*.

Associated Commands

- mpls ldp
- show mpls ldp neighbor

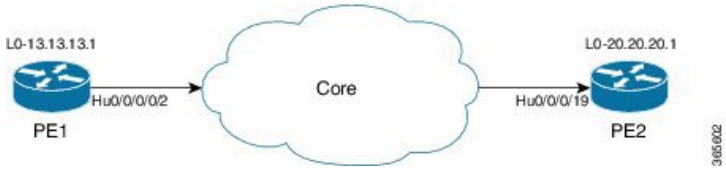
Determine if FIB is Enabled in the Core

Forwarding Information Base (FIB) must be enabled on all routers in the core, including the provider edge (PE) routers. For information on how to determine if FIB is enabled, see the *Implementing Cisco Express Forwarding* module in the *IP Addresses and Services Configuration Guide for Cisco NCS 5000 Series Routers*.

Configure Multiprotocol BGP on the PE Routers and Route Reflectors

Multiprotocol BGP (MP-BGP) propagates VRF reachability information to all members of a VPN community. You must configure MP-BGP peering in all the PE routers within a VPN community.

Figure 4: Multiprotocol BGP on PE Routers



Configuration Example

This example shows how to configure MP-BGP on PE1. The loopback address (20.20.20.1) of PE2 is specified as the neighbor of PE1. Similarly, you must perform this configuration on PE2 node as well, with the loopback address (13.13.13.1) of PE1 specified as the neighbor of PE2.

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
```

```
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#neighbor 20.20.20.1
Router-PE1(config-bgp-nbr)#remote-as 2001
Router-PE1(config-bgp-nbr)#update-source loopback 0
Router-PE1(config-bgp-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#address-family vpnv4 unicast
Router-PE1(config-bgp-nbr-af)#exit
Router-PE1(config-bgp-nbr)#exit
/* VRF configuration */
Router(config-bgp)# vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#commit
```

Running Configuration

```
router bgp 2001
 bgp router-id 13.13.13.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 20.20.20.1
  remote-as 2001
  update-source Loopback0
  address-family vpnv4 unicast
  !
  address-family ipv4 unicast
  !
 !
 vrf vrf1601
  rd 2001:1601
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
 !
```

Verification

- Verify if the BGP state is established, and if the Remote AS and local AS displays the same value (2001 in this example):

```
Router-PE1#show bgp neighbor

BGP neighbor is 20.20.20.1
 Remote AS 2001, local AS 2001, internal link
 Remote router ID 20.20.20.1
  BGP state = Established, up for 1d19h
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
```

```
 Last write before reset 00:00:00, attempted 0, written 0
 Second last write before reset 00:00:00, attempted 0, written 0
 Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
 Last write pulse rcvd before reset 00:00:00
 Socket not armed for io, armed for read, armed for write
 Last write thread event before reset 00:00:00, second last 00:00:00
 Last KA expiry before reset 00:00:00, second last 00:00:00
 Last KA error before reset 00:00:00, KA not sent 00:00:00
 Last KA start before reset 00:00:00, second last 00:00:00
 Precedence: internet
 Non-stop routing is enabled
 Multi-protocol capability received
 Neighbor capabilities:
   Route refresh: advertised (old + new) and received (old + new)
   Graceful Restart (GR Awareness): received
   4-byte AS: advertised and received
   Address family IPv4 Unicast: advertised and received
   Address family VPNv4 Unicast: advertised and received
 Received 25595 messages, 0 notifications, 0 in queue
 Sent 8247 messages, 0 notifications, 0 in queue
 Minimum time between advertisement runs is 0 secs
 Inbound message logging enabled, 3 messages buffered
 Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
 BGP neighbor version 484413
 Update group: 0.4 Filter-group: 0.3  No Refresh request being processed
 Inbound soft reconfiguration allowed
 NEXT_HOP is always this router
 AF-dependent capabilities:
   Outbound Route Filter (ORF) type (128) Prefix:
     Send-mode: advertised, received
     Receive-mode: advertised, received
   Graceful Restart capability received
     Remote Restart time is 120 seconds
     Neighbor did not preserve the forwarding state during latest restart
   Additional-paths Send: advertised and received
   Additional-paths Receive: advertised and received
 Route refresh request: received 1, sent 1
 Policy for incoming advertisements is pass-all
 Policy for outgoing advertisements is pass-all
 24260 accepted prefixes, 24260 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 2000, suppressed 0, withdrawn 0
 Maximum prefixes allowed 1048576
 Threshold for warning message 75%, restart interval 0 min
 AIGP is enabled
 An EoR was received during read-only mode
 Last ack version 484413, Last synced ack version 0
 Outstanding version objects: current 0, max 1
 Additional-paths operation: Send and Receive
 Send Multicast Attributes
 Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
option

For Address Family: VPNv4 Unicast
 BGP neighbor version 798487
 Update group: 0.2 Filter-group: 0.1  No Refresh request being processed
 AF-dependent capabilities:
   Graceful Restart capability received
     Remote Restart time is 120 seconds
     Neighbor did not preserve the forwarding state during latest restart
   Additional-paths Send: advertised and received
   Additional-paths Receive: advertised and received
```

```
 Route refresh request: received 0, sent 0
 29150 accepted prefixes, 29150 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 7200, suppressed 0, withdrawn 0
 Maximum prefixes allowed 2097152
 Threshold for warning message 75%, restart interval 0 min
 AIGP is enabled
 An EoR was received during read-only mode
 Last ack version 798487, Last synced ack version 0
 Outstanding version objects: current 0, max 1
 Additional-paths operation: Send and Receive
 Send Multicast Attributes
 Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
option

 Connections established 1; dropped 0
 Local host: 13.13.13.1, Local port: 35018, IF Handle: 0x00000000
 Foreign host: 20.20.20.1, Foreign port: 179
 Last reset 00:00:00
```

- Verify if all the IP addresses are learnt on PE1 from PE2:

```
Router-PE1#show bgp vpnv4 unicast

BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 798487
BGP NSR Initial initsync version 15151 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32       192.13.26.5                              0 7501 i
*> 20.13.1.2/32       192.13.26.5                              0 7501 i
*> 20.13.1.3/32       192.13.26.5                              0 7501 i
*> 20.13.1.4/32       192.13.26.5                              0 7501 i
*> 20.13.1.5/32       192.13.26.5                              0 7501 i
*>i20.14.1.1/3214.14.14.1                    100      0 8501 i
*>i20.14.1.2/3214.14.14.1                    100      0 8501 i
*>i20.14.1.3/3214.14.14.1                    100      0 8501 i
*>i20.14.1.4/3214.14.14.1                    100      0 8501 i
*>i20.14.1.5/3214.14.14.1                    100      0 8501 i
```

Related Topics

- Configure the Core Network, on page 4
- Define VRFs on PE Routers to Enable Customer Connectivity, on page 11

For more details on Multiprotocol BGP, see *BGP Configuration Guide for Cisco NCS 5000 Series Routers*.

Associated Commands

Associated Commands

- neighbor
- router bgp
- update-source
- vrf
- show bgp

Connect MPLS VPN Customers

Connecting MPLS VPN customers involves these main tasks:

- Define VRFs on PE Routers to Enable Customer Connectivity, on page 11
- Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 12
- Configure the Routing Protocol between the PE and CE Routers

  Use any of these options:

  - Configure BGP as the Routing Protocol Between the PE and CE Routers, on page 14
  - Configure RIPv2 as the Routing Protocol Between the PE and CE Routers, on page 18
  - Configure Static Routes Between the PE and CE Routers, on page 19
  - Configure OSPF as the Routing Protocol Between the PE and CE Routers, on page 20

Define VRFs on PE Routers to Enable Customer Connectivity

VPN routing and forwarding (VRF) defines the VPN membership of a customer site attached to a PE router. A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A VRF contains all the routes available to the site from the VPNs of which it is a member. The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities.

Configuration Example

This example configures a VRF instance (vrf1601) and specifies the import and export route-targets (2001:1601). The import route policy is the one that can be imported into the local VPN. The export route policy is the one that can be exported from the local VPN. The import route-target configuration allows exported VPN routes to be imported into the VPN if one of the route targets of the exported route matches one of the local VPN import route targets. When the route is advertised to other PE routers, the export route target is sent along with the route as an extended community.

```
Router-PE1#configure
Router-PE1(config)#vrf vrf1601
Router-PE1(config-vrf)#address-family ipv4 unicast
Router-PE1(config-vrf-af)#import route-target
```

```
Router-PE1(config-vrf-af-import-rt)#2001:1601
Router-PE1(config-vrf-af-import-rt)#exit
Router-PE1(config-vrf-af)#export route-target
Router-PE1(config-vrf-af-export-rt)#2001:1601
Router-PE1(config-vrf-af-export-rt)#commit
```

This VRF instance is then associated with the respective BGP instance.

Running Configuration

```
vrf vrf1601
 address-family ipv4 unicast
  import route-target
   2001:1601
  !
  export route-target
   2001:1601
  !
 !
!
```

Verification

Verify the import and export route targets.

```
Router-PE1#show vrf vrf1601
VRF                  RD          RT                  AFI   SAFI
vrf1601              2001:1601
                                 import  2001:1601   IPV4  Unicast
                                 export  2001:1601   IPV4  Unicast
```

Related Topics

- Configure VRF Interfaces on PE Routers for Each VPN Customer, on page 12
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 7

Associated Commands

- import route-policy
- import route-target
- export route-policy
- export route-target
- vrf

Configure VRF Interfaces on PE Routers for Each VPN Customer

After a VRF instance is created, you must associate that VRF instance with an interface or a sub-interface on the PE routers.

Note You must remove the IPv4 or IPv6 addresses from an interface prior to assigning, removing, or changing an interface's VRF. If this is not done in advance, any attempt to change the VRF on an IP interface is rejected.

Configuration Example

This example assigns an IP address *192.13.26.6* to the interface (*HundredGigE0/0/0/14.1601*) on PE1 router and associates the VRF instance *vrf1601*, to that interface.

```
Router-PE1#configure
Router-PE1(config)#interface HundredGigE0/0/0/14.1601
Router-PE1(config-if)#vrf vrf1601
Router-PE1(config-if)#ipv4 address 192.13.26.6 255.255.255.252
Router-PE1(config-if)#encapsulation dot1q 1601
Router-PE1(config)#commit
```

Running Configuration

```
interface HundredGigE0/0/0/14.1601
 vrf vrf1601
 ipv4 address 192.13.26.6 255.255.255.252
 encapsulation dot1q 1601
!
```

Verification

- Verify that the interface with which the VRF is associated, is UP.

```
Router-PE1#show ipv4 vrf vrf1601 interface
interface HundredGigE0/0/0/14.1601 is Up, ipv4 protocol is Up
  Vrf is vrf1601 (vrfid 0x60000001)
  Internet address is 192.13.26.6/30
  MTU is 1518 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000001
```

Related Topics

- Define VRFs on PE Routers to Enable Customer Connectivity, on page 11

Configure Routing Protocol Between the PE and CE Routers

Configure BGP as the Routing Protocol Between the PE and CE Routers

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. PE to PE or PE to route reflector (RR) sessions are iBGP sessions, and PE to CE sessions are eBGP sessions. PE to CE eBGP sessions can be directly or indirectly connected (eBGP multihop).

Figure 5: BGP as the Routing Protocol between PE and CE Routers

Configuration Example

This example lists the steps to configure BGP as the routing protocol between the PE and CE routers. The route policy, *pass-all* in this example, must be configured before it can be attached.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router bgp 2001
Router-PE1(config-bgp)#bgp router-id 13.13.13.1
Router-PE1(config-bgp)#address-family ipv4 unicast
Router-PE1(config-bgp-af)#exit
Router-PE1(config-bgp)#address-family vpnv4 unicast
Router-PE1(config-bgp-af)#exit
/* VRF configuration */
Router-PE1(config-bgp)#vrf vrf1601
Router-PE1(config-bgp-vrf)#rd 2001:1601
Router-PE1(config-bgp-vrf)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-af)#label mode per-vrf
Router-PE1(config-bgp-vrf-af)#redistribute connected
Router-PE1(config-bgp-vrf-af)#exit
Router-PE1(config-bgp-vrf)#neighbor 192.13.26.5
Router-PE1(config-bgp-vrf-nbr)#remote-as 7501
Router-PE1(config-bgp-vrf-nbr)#address-family ipv4 unicast
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all in
Router-PE1(config-bgp-vrf-nbr-af)#route-policy pass-all out
Router-PE1(config-bgp-vrf-nbr-af)#commit
```

CE1:

```
Router-CE1#configure
Router-CE1(config)#router bgp 2001
Router-CE1(config-bgp)#bgp router-id 8.8.8.1
Router-CE1(config-bgp)#address-family ipv4 unicast
Router-CE1(config-bgp-af)#exit
Router-CE1(config-bgp)#address-family vpnv4 unicast
Router-CE1(config-bgp-af)#exit
Router-CE1(config-bgp)#neighbor 192.13.26.6
Router-CE1(config-bgp-nbr)#remote-as 2001
Router-CE1(config-bgp-nbr)#address-family ipv4 unicast
Router-CE1(config-bgp-nbr-af)#route-policy pass-all in
Router-CE1(config-bgp-nbr-af)#route-policy pass-all out
Router-CE1(config-bgp-nbr-af)#commit
```

Running Configuration

PE1:

```
router bgp 2001
 bgp router-id 13.13.13.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 vrf vrf1601
  rd 2001:1601
  address-family ipv4 unicast
   label mode per-vrf
   redistribute connected
  !
  neighbor 192.13.26.5
   remote-as 7501
   address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
   !
  !
 !
```

CE1:

```
router bgp 7501
 bgp router-id 8.8.8.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 192.13.26.6
  remote-as 2001
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
 !
!
```

Verification

- **PE1:**

```
Router-PE1#show bgp neighbor
BGP neighbor is 192.13.26.5
 Remote AS 6553700, local AS 2001, external link
 Administratively shut down
 Remote router ID 192.13.26.5
  BGP state = Established
  NSR State: None
  Last read 00:00:04, Last read before reset 00:00:00
  Hold time is 60, keepalive interval is 20 seconds
  Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
  Last write 00:00:16, attempted 19, written 19
  Second last write 00:00:36, attempted 19, written 19
  Last write before reset 00:00:00, attempted 0, written 0
  Second last write before reset 00:00:00, attempted 0, written 0
  Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
```

```
 Last write pulse rcvd before reset 00:00:00
 Socket not armed for io, armed for read, armed for write
 Last write thread event before reset 00:00:00, second last 00:00:00
 Last KA expiry before reset 00:00:00, second last 00:00:00
 Last KA error before reset 00:00:00, KA not sent 00:00:00
 Last KA start before reset 00:00:00, second last 00:00:00
 Precedence: internet
 Non-stop routing is enabled
 Graceful restart is enabled
 Restart time is 120 seconds
 Stale path timeout time is 360 seconds
 Enforcing first AS is enabled
 Multi-protocol capability not received
 Received 0 messages, 0 notifications, 0 in queue
 Sent 0 messages, 0 notifications, 0 in queue
 Minimum time between advertisement runs is 30 secs
 Inbound message logging enabled, 3 messages buffered
 Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
 BGP neighbor version 0
 Update group: 0.2 Filter-group: 0.0  No Refresh request being processed
 Inbound soft reconfiguration allowed
 AF-dependent capabilities:
   Outbound Route Filter (ORF) type (128) Prefix:
     Send-mode: advertised
     Receive-mode: advertised
   Graceful Restart capability advertised
     Local restart time is 120, RIB purge time is 600 seconds
     Maximum stalepath time is 360 seconds
 Route refresh request: received 0, sent 0
 Policy for incoming advertisements is pass-all
 Policy for outgoing advertisements is pass-all
 0 accepted prefixes, 0 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 0, suppressed 0, withdrawn 0
 Maximum prefixes allowed 1048576
 Threshold for warning message 75%, restart interval 0 min
 An EoR was not received during read-only mode
 Last ack version 1, Last synced ack version 0
 Outstanding version objects: current 0, max 0
 Additional-paths operation: None
 Advertise VPNv4 routes enabled with defaultReoriginate,disable Local with stitching-RT
option
 Advertise VPNv6 routes is enabled with default option


 Connections established 1; dropped 0
 Local host: 192.13.26.6, Local port: 23456, IF Handle: 0x00000000
 Foreign host: 192.13.26.5, Foreign port: 179
 Last reset 03:12:58, due to Admin. shutdown (CEASE notification sent - administrative
shutdown)
 Time since last notification sent to neighbor: 03:12:58
 Notification data sent:
   None
 External BGP neighbor not directly connected.
```

- **CE1:**

```
Router-CE1#show bgp neighbor
BGP neighbor is 192.13.26.6
 Remote AS 2001, local AS 6553700, external link
 Remote router ID 192.13.26.6
```

```
 BGP state = Established
 NSR State: None
 Last read 00:00:04, Last read before reset 00:00:00
 Hold time is 60, keepalive interval is 20 seconds
 Configured hold time: 60, keepalive: 30, min acceptable hold time: 3
 Last write 00:00:16, attempted 19, written 19
 Second last write 00:00:36, attempted 19, written 19
 Last write before reset 00:00:00, attempted 0, written 0
 Second last write before reset 00:00:00, attempted 0, written 0
 Last write pulse rcvd  Apr 12 10:31:20.739 last full not set pulse count 27939
 Last write pulse rcvd before reset 00:00:00
 Socket not armed for io, armed for read, armed for write
 Last write thread event before reset 00:00:00, second last 00:00:00
 Last KA expiry before reset 00:00:00, second last 00:00:00
 Last KA error before reset 00:00:00, KA not sent 00:00:00
 Last KA start before reset 00:00:00, second last 00:00:00
 Precedence: internet
 Non-stop routing is enabled
 Graceful restart is enabled
 Restart time is 120 seconds
 Stale path timeout time is 360 seconds
 Enforcing first AS is enabled
 Multi-protocol capability not received
 Received 0 messages, 0 notifications, 0 in queue
 Sent 0 messages, 0 notifications, 0 in queue
 Minimum time between advertisement runs is 30 secs
 Inbound message logging enabled, 3 messages buffered
 Outbound message logging enabled, 3 messages buffered

For Address Family: IPv4 Unicast
 BGP neighbor version 0
 Update group: 0.1 Filter-group: 0.0  No Refresh request being processed
 Inbound soft reconfiguration allowed
 AF-dependent capabilities:
   Outbound Route Filter (ORF) type (128) Prefix:
     Send-mode: advertised
     Receive-mode: advertised
   Graceful Restart capability advertised
     Local restart time is 120, RIB purge time is 600 seconds
     Maximum stalepath time is 360 seconds
 Route refresh request: received 0, sent 0
 Policy for incoming advertisements is pass-all
 Policy for outgoing advertisements is pass-all
 0 accepted prefixes, 0 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 0, suppressed 0, withdrawn 0
 Maximum prefixes allowed 1048576
 Threshold for warning message 75%, restart interval 0 min
 An EoR was not received during read-only mode
 Last ack version 1, Last synced ack version 0
 Outstanding version objects: current 0, max 0
 Additional-paths operation: None

 Connections established 0; dropped 0
 Local host: 192.13.26.5, Local port: 179, IF Handle: 0x00000000
 Foreign host: 192.13.26.6, Foreign port: 23456
 Last reset 00:00:00
 External BGP neighbor not directly connected.
```

Related Topics

- Connect MPLS VPN Customers, on page 11

- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 7

For more details on BGP, see *BGP Configuration Guide for Cisco NCS 5000 Series Routers*.

Associated Commands

- label mode
- neighbor
- rd
- redistribute
- remote-as
- route-policy
- router bgp

Configure RIPv2 as the Routing Protocol Between the PE and CE Routers

Figure 6: RIP as the Routing Protocol between PE and CE Routers

Configuration Example

This example lists the steps to configure RIPv2 as the routing protocol between the PE and CE routers. The VRF instance vrf1601 is configured in the router rip configuration mode and the respective interface (TenGigE0/0/0/14.1601 on PE1 and TenGigE0/0/0/18.1601 on CE1) is associated with that VRF. The **redistribute** option specifies routes to be redistributed into RIP.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router rip
Router-PE1(config-rip)#vrf vrf1601
Router-PE1(config-rip-vrf)#interface TenGigE0/0/0/14.1601
Router-PE1(config-rip-vrf-if)#exit
Router-PE1(config-bgp-vrf)#redistribute bgp 2001
Router-PE1(config-bgp-vrf)#redistribute connected
Router-PE1(config-bgp-vrf)#commit
```

CE1:

```
Router-CE1#configure
Router-CE1(config)#router rip
Router-CE1(config-rip)#vrf vrf1601
Router-CE1(config-rip-vrf)#interface TenGigE0/0/0/14.1601
Router-CE1(config-rip-if)#exit
Router-CE1(config-rip)#redistribute connected
Router-CE1(config-rip)#commit
```

Running Configuration

PE1:

```
Router-PE1#show running-config router rip
router rip
 vrf vrf1601
  interface TenGigE0/0/0/14.1601
  !
  redistribute bgp 2001
  redistribute connected
 !
!
```

CE1:

```
Router-CE1#show running-config router rip
router rip
 vrf vrf1601
  interface TenGigE0/0/0/18.1601
  !
  redistribute connected
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 11

Associated Commands

- redistribute
- router rip

Configure Static Routes Between the PE and CE Routers

Configuration Example

In this example, the static route is assigned to VRF, vrf1601.

```
Router-PE1#configure
Router-PE1(config)#router static
Router-PE1(config-static)#vrf vrf1601
Router-PE1(config-static-vrf)#address-family ipv4 unicast
Router-PE1(config-static-vrf-afi)#23.13.1.1/32 TenGigE0/0/0/14.1601 192.13.3.93
Router-PE1(config-static-vrf-afi)#commit
```

Repeat the configuration in CE1, with the respective interface values.

Running Configuration

PE1:

```
router static
 vrf vrf1601
  address-family ipv4 unicast
   23.13.1.1/32 TenGigE0/0/0/14.1601 192.13.3.93
  !
 !
!
```

CE1:

```
router static
 vrf vrf1601
  address-family ipv4 unicast
   23.8.1.2/32 TenGigE0/0/0/18.1601 192.8.3.94
  !
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 11

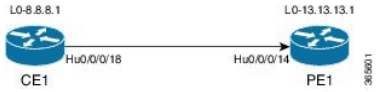
Associated Commands

- router static

Configure OSPF as the Routing Protocol Between the PE and CE Routers

You can use RIP, OSPF or ISIS as the routing protocol between the PE and CE routers.

Figure 7: OSPF as the Routing Protocol between PE and CE Routers



Configuration Example

This example lists the steps to configure PE-CE routing sessions that use OSPF routing protocol. A VRF instance *vrf1601* is configured in the **router ospf** configuration mode. The router-id for the OSPF process is 13.13.13.1. The **redistribute** option specifies routes to be redistributed into OSPF. The OSPF area is configured to be *1* and interface TenGigE0/0/0/14.1601 is associated with that area to enable routing on it.

PE1:

```
Router-PE1#configure
Router-PE1(config)#router ospf pe-ce-ospf-vrf
Router-PE1(config-ospf)#router-id 13.13.13.1
Router-PE1(config-ospf)#vrf vrf1601
Router-PE1(config-ospf-vrf)#redistribute connected
Router-PE1(config-ospf-vrf)#redistribute bgp 2001
Router-PE1(config-ospf-vrf)#area 1
Router-PE1(config-ospf-vrf-ar)#interface TenGigE0/0/0/14.1601
Router-PE1(config-ospf-vrf-ar)# commit
```

Repeat this configuration at PE2 node as well.

CE1:

```
Router-CE1#configure
Router-CE1(config)#router ospf ospf pe-ce-1
Router-CE1(config-ospf)#router-id 8.8.8.1
Router-CE1(config-ospf)#vrf vrf1601
Router-CE1(config-ospf-vrf)#area 1
Router-CE1(config-ospf-vrf-ar)#interface TenGigE0/0/0/18.1601
Router-CE1(config-ospf-vrf-ar)#commit
```

Running Configuration

PE1:

```
router ospf pe-ce-ospf-vrf
 router-id 13.13.13.1
 vrf vrf1601
  redistribute connected
  redistribute bgp 2001
  area 1
   interface TenGigE0/0/0/14.1601
   !
  !
 !
!
```

CE1:

```
router ospf pe-ce-1
 router-id 8.8.8.1
 vrf vrf1601
  area 1
   interface TenGigE0/0/0/18.1601
   !
  !
 !
!
```

Related Topics

- Connect MPLS VPN Customers, on page 11

Associated Commands

- router ospf

Verify MPLS L3VPN Configuration

You must verify these to ensure the successful configuration of MPLS L3VPN:

Verify the L3VPN Traffic Flow

- Verify the number of bytes switched for the label associated with the VRF (vrf1601):

  P node:

```
Router-P#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24119  Pop         20.20.20.1/32      Hu0/0/0/0    191.31.1.90     2170204180148
```

  PE2:

```
Router#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      11124125835
```

Verify the Underlay (transport)

- Verify if the LDP neighbor connection is established with the respective neighbor:

```
Router-PE1#show mpls ldp neighbor
Peer LDP Identifier: 16.16.16.1:0
  TCP connection: 16.16.16.1:47619 - 13.13.13.1:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 40395/35976; Downstream-Unsolicited
  Up time: 2w2d
  LDP Discovery Sources:
    IPv4: (1)

    IPv6: (0)
  Addresses bound to this peer:
    IPv4: (6)
      10.64.98.32    87.0.0.2       88.88.88.14    50.50.50.50
      178.0.0.1      192.1.1.1
    IPv6: (0)
```

- Verify if the label update is received by the FIB:

```
Router-PE1#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24036  Pop         16.16.16.1/32      Hu0/0/0/2    191.22.1.2      293294

24037  24165       18.18.18.1/32      Hu0/0/0/2    191.22.1.2      500

24039  24167       20.20.20.1/32      Hu0/0/0/2    191.22.1.2      17872433
       24167       20.20.20.1/32      Hu0/0/0/2.1  191.22.3.2      6345
24041  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      7950400999
```

- Verify if label is updated in the hardware:

```
Router-PE1#show mpls forwarding labels 24001 hardware egress

Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
24039  24167       20.20.20.1/32     Hu0/0/0/2    191.22.1.2      N/A
       24167       20.20.20.1/32     Hu0/0/0/2.1  191.22.3.2    N/A

 Show-data Print at RPLC

 LEAF - HAL pd context :
 sub-type : MPLS, ecd_marked:0, has_collapsed_ldi:0
 collapse_bwalk_required:0, ecdv2_marked:0

Leaf H/W Result:

Leaf H/W Result on NP:0
Label           SwitchAction      EgressIf      Programmed
24039                    0    0x    200185     Programmed

nrLDI eng ctx:
     flags: 0x101, proto: 2, npaths: 0, nbuckets: 1
    ldi_tbl_idx: 0xc37e40, ecd_ref_cft: 0
    pbts_ldi_tbl_idx: 0x0, fastnrldi:0x0

NR-LDI H/W Result for path 0 [index: 0xc37e40 (BE),  common to all NPs]:

 ECMP Sw Idx: 12811840 HW Idx: 200185 Path Idx: 0

NR-LDI H/W Result for path 1 [index: 0xc37e41 (BE),  common to all NPs]:

 ECMP Sw Idx: 12811841 HW Idx: 200185 Path Idx: 1

SHLDI eng ctx:
    flags: 0x0, shldi_tbl_idx: 0, num_entries:0

SHLDI HW data for path 0 [index: 0 (BE)] (common to all NPs):
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 0
SHLDI HW data for path 1 [index: 0x1 (BE)] (common to all NPs):
Unable to get HW NRLDI Element rc: 1165765120NRLDI Idx: 1

TX H/W Result for NP:0 (index: 0x187a0 (BE)):

 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100256
 Egress Next Hop IF:     100047
 Hw Next Hop Intf:       606
 HW Port:                0
 Next Hop Flags:         COMPLETE
 Next Hop MAC:           e4aa.5d9a.5f2e

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform
```

```
TX H/W Result for NP:0 (index: 0x189a8 (BE)):

 Next Hop Data
 Next Hop Valid:         YES
 Next Hop Index:         100776
 Egress Next Hop IF:     100208
 Hw Next Hop Intf:       607
 HW Port:                0
 Next Hop Flags:         COMPLETE
 Next Hop MAC:           e4aa.5d9a.5f2d

NHINDEX H/W Result for NP:0 (index: 0 (BE)):
NhIndex is NOT required on this platform

NHINDEX STATS: pkts 0, bytes 0 (no stats)

RX H/W Result on NP:0 [Adj ptr:0x40 (BE)]:
Rx-Adj is NOT required on this platform
```

Verify the Overlay (L3VPN)

Imposition Path

- Verify if the BGP neighbor connection is established with the respective neighbor node:

```
Router-PE1#show bgp summary
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 18003
BGP main routing table version 18003
BGP NSR Initial initsync version 3 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.

Process       RcvTblVer   bRIB/RIB   LabelVer  ImportVer  SendTblVer  StandbyVer
Speaker           18003      18003      18003      18003       18003           0

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
21.21.21.1        0  2001   19173    7671    18003    0    0    1d07h       4000
192.13.2.149      0  7001    4615    7773    18003    0    0 09:26:21        125
```

- Verify if BGP routes are advertised and learnt:

```
Router-PE1#show bgp vpnv4 unicast
BGP router identifier 13.13.13.1, local AS number 2001
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0   RD version: 0
BGP main routing table version 305345
BGP NSR Initial initsync version 12201 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
```

```
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 2001:1601 (default for vrf vrf1601)
*> 20.13.1.1/32       192.13.26.5                            0 7501 i
*> 20.13.1.2/32       192.13.26.5                            0 7501 i
*>i20.23.1.1/32       20.20.20.1                100          0 6553700 11501 i
*>i20.23.1.2/32       20.20.20.1                100          0 6553700 11501 i
```

- Verify BGP labels:

```
Router-PE1#show bgp label table
Label   Type               VRF/RD         Context
24041   IPv4 VRF Table   vrf1601          -
24042   IPv4 VRF Table   vrf1602          -
```

- Verify if the route is downloaded in the respective VRF:

```
Router-PE1#show cef vrf vrf1601 20.23.1.1
20.23.1.1/32, version 743, internal 0x5000001 0x0 (ptr 0x8f932174) [1], 0x0 (0x8fa99990),
 0xa08 (0x8f9fba58)
 Updated Apr 20 12:33:47.840
 Prefix Len 32, traffic index 0, precedence n/a, priority 3
   via 20.20.20.1/32, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8c0e3148 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 20.20.20.1/32 via 24039/0/21
    next hop 191.23.1.2/32 Hu0/0/1/1    labels imposed {24059 24031}
```

Disposition Path

- Verify if the imposition and disposition labels are assigned and label bindings are exchanged for L3VPN prefixes:

```
Router-PE2#show mpls lsd forwarding
In_Label, (ID), Path_Info: <Type>
24030, (IPv4, 'default':4U, 13.13.13.1/32), 5 Paths
   1/1: IPv4, 'default':4U, Hu0/0/0/19.2, nh=191.31.1.93, lbl=24155,
            flags=0x0, ext_flags=0x0
24031, (VPN-VRF, 'vrf1601':4U), 1 Paths
   1/1: PopLkup-v4, 'vrf1601':4U, ipv4
24032, (VPN-VRF, 'vrf1602':4U), 1 Paths
   1/1: PopLkup-v4, 'vrf1602':4U, ipv4
```

- Verify if the label update is received by the FIB:

```
Router-PE2#show mpls forwarding
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------

24019  Pop         18.18.18.3/32     Hu0/0/0/19   191.31.1.89     11151725032
```

```
24030  24155       13.13.13.1/32      Hu0/0/0/19   191.31.1.89     3639895

24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                vrf1601                          32167647049
```

VRF-lite

VRF-lite is the deployment of VRFs without MPLS. VRF-lite allows a service provider to support two or more VPNs with overlapping IP addresses. With this feature, multiple VRF instances can be supported in customer edge devices.

VRF-lite interfaces must be Layer 3 interface and this interface cannot belong to more than one VRF at any time. Multiple interfaces can be part of the same VRF, provided all of them participate in the same VPN.

Configure VRF-lite

Consider two customers having two VPN sites each, that are connected to the same PE router. VRFs are used to create a separate routing table for each customer. We create one VRF for each customer (say, vrf1 and vrf2) and then add the corresponding interfaces of the router to the respective VRFs. Each VRF has its own routing table with the interfaces configured under it. The global routing table of the router does not show these interfaces, whereas the VRF routing table shows the interfaces that were added to the VRF. PE routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP or RIP.

To summarize, VRF-lite configuration involves these main tasks:

- Create VRF
- Configure VRF under the interface
- Configure VRF under routing protocol

Configuration Example

- **Create VRF:**

```
Router#configure
Router(config)#vrf vrf1
Router(config-vrf)#address-family ipv4 unicast
/* You must create route-policy pass-all before this configuration */
Router(config-vrf-af)#import from default-vrf route-policy pass-all
Router(config-vrf-af)#import route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-af)#export route-target
Router(config-vrf-import-rt)#100:100
Router(config-vrf-import-rt)#exit
Router(config-vrf-import-rt)#commit
```

Similarly create vrf2, with route-target as 100:100.

- **Configure VRF under the interface:**

```
Router#configure
Router(config)#
Router(config-subif)#vrf vrf1
Router(config-subif)#ipv4 address 192.0.2.2 255.255.255.252
Router(config-subif)#encapsulation dot1q 2001
Router(config-subif)#exit

Router(config)#
Router(config-subif)#vrf vrf2
Router(config-subif)#ipv4 address 192.0.2.5/30 255.255.255.252
Router(config-subif)#encapsulation dot1q 2000
Router(config-vrf-import-rt)#commit
```

Similarly configure vrf1 under interface TenGigE0/0/0/1.2001 and vrf2 under interface TenGigE0/0/0/1.2000

- **Configure VRF under routing protocol**:

```
Router#configure
Router(config)#router rip
Router(config-rip)#vrf vrf1
Router(config-rip-vrf)#
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#
Router(config-rip-vrf-if)#exit
Router(config-rip-vrf)#default-information originate
Router(config-vrf-import-rt)#commit
```

Similarly configure vrf2 under rip, with

Running Configuration

```
/* VRF Configuration */

vrf vrf1
address-family ipv4 unicast
  import route-target
   100:100
  !
  export route-target
   100:100
  !
!
!
vrf vrf2
address-family ipv4 unicast
  import route-target
   100:100
  !
  export route-target
   100:100
  !
!
!


/* Interface Configuration */
```

```
interface
vrf vrf1
ipv4 address 192.0.2.2 255.255.255.252
encapsulation dot1q 2001
!

interface
vrf vrf2
ipv4 address 192.0.2.5/30 255.255.255.252
encapsulation dot1q 2000
!

interface
vrf vrf1
ipv4 address 203.0.113.2 255.255.255.252
encapsulation dot1q 2001
!

interface
vrf vrf2
ipv4 address 203.0.113.5 255.255.255.252
encapsulation dot1q 2000
!

/* Routing Protocol Configuration */
router rip
interface Loopback0
!
interface
!
interface
!
interface
!
interface
!
interface
!
interface
!

vrf vrf1
  interface
  !
  interface
  !
  default-information originate
 !
vrf vrf2
  interface
  !
  interface
  !
  default-information originate
 !
```

Verification

```
Router#show route vrf vrf1
Mon Jul  4 19:12:54.739 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
```

```
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

C    203.0.113.0/24 is directly connected, 00:07:01,
L    203.0.113.2/30 is directly connected, 00:07:01,
C    192.0.2.0/24 is directly connected, 00:05:51,
L    192.0.2.2/30 is directly connected, 00:05:51,


Router#show route vrf vrf2
Mon Jul  4 19:12:59.121 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

R    198.51.100.53/30 [120/1] via 192.0.2.1, 00:01:42,
C    203.0.113.0/24 is directly connected, 00:08:43,
L    203.0.113.5/30 is directly connected, 00:08:43,
C    192.0.2.0/24 is directly connected, 00:06:17,
L    192.0.2.5/30 is directly connected, 00:06:17,
```

Related Topics

- VRF-lite, on page 26

Associated Commands

- import route-target
- export route-target
- vrf

MPLS L3VPN Services using Segment Routing

Currently, MPLS Label Distribution Protocol (LDP) is the widely used transport for MPLS L3VPN services. The user can achieve better resilience and convergence for the network traffic, by transporting MPLS L3VPN services using Segment Routing (SR), instead of MPLS LDP. Segment routing can be directly applied to the

MPLS architecture without changing the forwarding plane. In a segment-routing network using the MPLS data plane, LDP or other signaling protocol is not required; instead label distribution is performed by IGP (IS-IS or OSPF) or BGP protocol. Removing protocols from the network simplifies its operation and makes it more robust and stable by eliminating the need for protocol interaction. Segment routing utilizes the network bandwidth more effectively than traditional MPLS networks and offers lower latency.

Configure MPLS L3VPN over Segment Routing

Topology

Given below is a network scenario, where MPLS L3VPN service is transported using Segment Routing.

In this topology, CE1 and CE2 are the two customer routers. ISP has two PE routers, PE1 and PE2 and a P router. RIP is used for the edge protocol support between the CE and PE routers. Label distribution can be performed by IGP (IS-IS or OSPF) or BGP. OSPF is used in this scenario.

Customer's autonomous system is 65534, which peers with ISP's autonomous system 65000. This must be a vrf peering to prevent route advertisement into the global IPv4 table. The ISP routers PE1 and PE2 contain the VRF (for example, vrf1601) for the customer. PE1 and PE2 export and import the same route targets, although this is not necessary.

Loopback interfaces are used in this topology to simulate the attached networks.

Configuration

You must complete these tasks to ensure the successful configuration of MPLS L3VPN over segment routing:

- Configure protocol support on PE-CE (refer, Connect MPLS VPN Customers, on page 11)
- Configure protocol support on PE-PE (refer, Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 7)

Configure Segment Routing in MPLS Core

This section takes you through the configuration procedure to enable segment routing in MPLS core. You must perform this configuration in PE1, P and PE2 routers in the topology, using the corresponding values.

Configuration Example

```
/* Configure Segment Routing using OSFP */

Router-PE1#configure
Router-PE1(config)# router ospf dc-sr
Router-PE1(config-ospf)#router-id 13.13.13.1
Router-PE1(config-ospf)#segment routing mpls
Router-PE1(config-ospf)#segment routing forwarding mpls
Router-PE1(config-ospf)#mpls ldp sync
Router-PE1(config-ospf)#mpls ldp auto-config
Router-PE1(config-ospf)#segment-routing mpls
Router-PE1(config-ospf)#segment-routing mpls sr-prefer
Router-PE1(config-ospf)#segment-routing prefix-sid-map advertise-local
Router-PE1(config-ospf)#exit
Router-PE1(config-ospf)#area 1
Router-PE1(config-ospf-ar)#interface HundredGigE0/0/0/2
Router-PE1(config-ospf-ar-if)#exit
```

```
Router-PE1(config-ospf-ar)#interface Loopback0
Router-PE1(config-ospf-ar-if)#prefix-sid index 1
Router-PE1(config-ospf-ar-if)#commit

/ * Configure segment routing global block */

Router# configure
Router(config)# segment-routing
Router(config-sr)# global-block 180000 200000
Router(config-sr)# commit
Router(config-sr)# exit


/* Configure Segment Routing using ISIS */

Router# configure
Router(config)# router isis ring
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.0001.1921.6800.1001.00
Router(config-isis)# nsr
Router(config-isis)# distribute link-state
Router(config-isis)# nsf cisco
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# mpls traffic-eng level-1
Router(config-isis-af)# mpls traffic-eng router-id loopback0
Router(config-isis-af)# segment-routing mpls
Router(config-isis-af)# exit
!
Router(config-isis)# interface loopback0
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-af)# prefix-sid index 30101
Router(config-isis-af)# exit
```

Running Configuration

PE1:

```
router ospf dc-sr
 router-id 13.13.13.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 1
  interface HundredGigE0/0/0/2
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
```

```
 !
!

configure
 router isis ring
  net 49.0001.1921.6800.1001.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30101
  !
 !
```

P node:

```
router ospf dc-sr
 router-id 16.16.16.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 1
  interface HundredGigE0/0/1/0
  !
  interface HundredGigE0/0/1/1
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
 !
!

configure
 router isis ring
  net 49.0001.1921.6800.1002.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
```

```
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30102
  !
 !
```

PE2:

```
router ospf dc-sr
 router-id 20.20.20.1
 segment-routing mpls
 segment-routing forwarding mpls
 mpls ldp sync
 mpls ldp auto-config
 segment-routing mpls
 segment-routing mpls sr-prefer
 segment-routing prefix-sid-map receive
 segment-routing prefix-sid-map advertise-local
 !
 area 0
  interface HundredGigE0/0/0/19
  !
  interface Loopback0
   prefix-sid index 1
  !
 !
!

configure
 segment-routing
  global-block 180000 200000
 !
!

configure
 router isis ring
  net 49.0001.1921.6800.1003.00
  nsr
  distribute link-state
  nsf cisco
  address-family ipv4 unicast
   metric-style wide
   mpls traffic-eng level-1
   mpls traffic-eng router-id Loopback0
   segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid index 30103
  !
```

Related Topics

You must perform these tasks as well to complete the MPLS L3VPN configuration over segment routing:

- Connect MPLS VPN Customers, on page 11
- Configure Multiprotocol BGP on the PE Routers and Route Reflectors, on page 7

Associated Commands

- index
- prefix-sid
- router isis
- router ospf
- segment-routing

The applicable segment routing commands are described in the *Segment Routing Command Reference for Cisco NCS 5500 Series Routers*

Verify MPLS L3VPN Configuration over Segment Routing

- Verify the statistics in core router and ensure that the counter for IGP transport label (64003 in this example) is increasing:

  P node:

```
Router-P#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ---------
64003  Pop         SR Pfx (idx 0)     Hu0/0/0/0   193.16.1.2      572842
```

- Verify the statistics in PE1 router:

  PE1:

```
Router-P#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ ------------    --------
64001  60003       SR Pfx (idx 0)     Hu0/0/0/2    191.22.1.2      532978
```

- Verify the statistics in PE2 router and ensure that the counter for the VPN label (24031 in this example) is increasing:

  PE2:

```
Router-PE2#show mpls forwarding
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ---------
24031  Aggregate   vrf1601: Per-VRF Aggr[V]   \
                                      vrf1601                      501241
```

Also, refer Verify MPLS L3VPN Configuration, on page 21 for a detailed list of commands and sample outputs.

Implementing MPLS L3VPNs - References

MPLS L3VPN Benefits

MPLS L3VPN provides the following benefits:

- Service providers can deploy scalable VPNs and deliver value-added services.
- Connectionless service guarantees that no prior action is necessary to establish communication between hosts.
- Centralized Service: Building VPNs in Layer 3 permits delivery of targeted services to a group of users represented by a VPN.
- Scalability: Create scalable VPNs using connection-oriented and point-to-point overlays.
- Security: Security is provided at the edge of a provider network (ensuring that packets received from a customer are placed on the correct VPN) and in the backbone.
- Integrated Quality of Service (QoS) support: QoS provides the ability to address predictable performance and policy implementation and support for multiple levels of service in an MPLS VPN.
- Straightforward Migration: Service providers can deploy VPN services using a straightforward migration path.
- Migration for the end customer is simplified. There is no requirement to support MPLS on the CE router and no modifications are required for a customer intranet.

Major Components of MPLS L3VPN—Details

Virtual Routing and Forwarding Tables

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE router. A VRF consists of the following components:

- An IP version 4 (IPv4) unicast routing table
- A derived FIB table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

These components are collectively called a VRF instance.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the FIB table for each VRF. A separate set of routing and FIB tables is maintained for each VRF. These tables prevent information from being

forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

VPN Routing Information: Distribution

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by BGP extended communities. VPN routing information is distributed as follows:

- When a VPN route that is learned from a CE router is injected into a BGP, a list of VPN route target extended community attributes is associated with it. Typically, the list of route target community extended values is set from an export list of route targets associated with the VRF from which the route was learned.
- An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes that a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target extended communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, or C—is imported into the VRF.

BGP Distribution of VPN Routing Information

A PE router can learn an IP prefix from the following sources:

- A CE router by static configuration
- An eBGP session with the CE router
- Open Shortest Path First (OSPF) and RIP as Interior Gateway Protocols (IGPs)

The IP prefix is a member of the IPv4 address family. After the PE router learns the IP prefix, the PE converts it into the VPN-IPv4 prefix by combining it with a 64-bit route distinguisher. The generated prefix is a member of the VPN-IPv4 address family. It uniquely identifies the customer address, even if the customer site is using globally nonunique (unregistered private) IP addresses. The route distinguisher used to generate the VPN-IPv4 prefix is specified by the **rd** command associated with the VRF on the PE router.

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. BGP communication takes place at two levels:

- Internal BGP (iBGP)—within the IP domain, known as an autonomous system.
- External BGP (eBGP)—between autonomous systems.

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by the BGP protocol extensions (see RFC 2283, Multiprotocol Extensions for BGP-4), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

MPLS Forwarding

Based on routing information stored in the VRF IP routing table and the VRF FIB table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label

and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on dynamic label switching. A customer data packet carries two levels of labels when traversing the backbone:

- The top label directs the packet to the correct PE router.
- The second label indicates how that PE router should forward the packet to the CE router.

Automatic Route Distinguisher Assignment

To take advantage of iBGP load balancing, every network VRF must be assigned a unique route distinguisher. VRF is require a route distinguisher for BGP to distinguish between potentially identical prefixes received from different VPNs.

With thousands of routers in a network each supporting multiple VRFs, configuration and management of route distinguishers across the network can present a problem. Cisco IOS XR software simplifies this process by assigning unique route distinguisher to VRFs using the **rd auto** command.

To assign a unique route distinguisher for each router, you must ensure that each router has a unique BGP router-id. If so, the **rd auto** command assigns a Type 1 route distinguisher to the VRF using the following format: *ip-address:number*. The IP address is specified by the BGP router-id statement and the number (which is derived as an unused index in the 0 to 65535 range) is unique across theVRFs.

Finally, route distinguisher values are checkpointed so that route distinguisher assignment to VRF is persistent across failover or process restart. If an route distinguisher is explicitely configured for a VRF, this value is not overridden by the autoroute distinguisher.

EVPN Default VRF Route Leaking

The EVPN Default VRF Route Leaking feature leak routes between EVPN address-family and IPv4/IPv6 unicast address-family (Default-VRF), enabling the data center hosts to access the Internet. This feature is an extension of Border Gateway Protocol (BGP) VRF Dynamic route leaking feature that provides connectivity between non-default VRF hosts and Default VRF hosts by exchanging routes between the non-default VRF and Default VRF. EVPN Default VRF Route Leaking feature extends the BGP VRF Dynamic leaking feature, by allowing EVPN/L3VPN hosts to communicate with Default VRF hosts.

The import process installs the Internet route in a VRF table or a VRF route in the Internet table, providing connectivity.

The BGP VRF Dynamic route leaking feature is enabled by:

- Importing from default-VRF to non-default-VRF using the following command in VRF address-family configuration mode.

  import from default-vrf route-policy *route-policy-name* **[advertise-as-vpn]**

  If the **advertise-as-vpn** keyword is used, the paths imported from the default-VRF to the non-default-VRF are advertised to the (EVPN/L3VPN) PEs as well as to the CEs. If the **advertise-as-vpn** keyword is not used, the paths imported from the default-VRF to the non-default-VRF are not advertised to the PEs. However, the paths are still advertised to the CEs.

  The EVPN Default VRF Route Leaking feature with **advertise-as-vpn** keyword, enables to advertise the paths imported from default-VRF to non-default VRFs to EVPN PE peers as well.

A new command **advertise vpnv4/vpnv6 unicast imported-from-default-vrf disable** is added under neighbor address-family configuration mode for EVPN and VPNv4/VPNv6 unicast to disable advertisement of Default-VRF leaked routes to that neighbor.

- Importing from non-default-VRF to default-VRF using the following command in VRF address-family configuration mode.

  export to default-vrf route-policy *route-policy-name* **[advertise-as-vpn]**

  The Dynamic Route Leaking feature enables leaking of local and CE routes to Default-VRF.

  A new optional keyword **allow-imported-vpn** is added to the above command, when configured, enables the leaking of EVPN and L3VPN imported/re-originated routes to the Default-VRF.

A route-policy is mandatory to filter the imported routes. This reduces the risk of unintended import of routes between the Internet table and the VRF tables and the corresponding security issues. There is no hard limit on the number of prefixes that can be imported. The import creates a new prefix in the destination VRF, which increases the total number of prefixes and paths.

Note Each VRF importing global routes adds workload equivalent to a neighbor receiving the global table. This is true even if the user filters out all but a few prefixes.

Scale Limitation of Default Route Leaking

Default VRF route leaking uses Dynamic Route Leaking feature to leak prefixes between the default VRF and the DC VRF. Do not use Dynamic Route Leaking feature to leak default VRF prefixes to large number of DC VRFs, even if you filter out all prefixes except a few that are to be leaked.

The following are the key factors that affect the performance:

- The default VRF prefix scale, which is approximately 0.7 million internet prefixes.
- The number of DC VRFs the default VRF prefixes that are to be imported.

To improve the scale, either the prefix scale or the number of VRFs whose prefixes that are to be imported must be reduced.

To manage the scale limitation, Cisco recommends you to do the following:

- Host the Internet prefixes on an adjacent PE with IPv4 unicast peering with DCI, and advertise a default route towards the DCI. On the DCI, import the default route from default VRF to DC VRFs.
- Host the Internet prefixes on an adjacent PE with IPv4 unicast peering with DCI. On the DCI, configure a static default route in the DC VRF with the next hop of the default VRF pointing to the adjacent PE address.
- Configure the static default route 0.0.0.0/0 on DC VRF with nexthop as “vrf default”.

Note If the static routes are re-distributed to BGP, make sure it is not unintentionally advertised out.

EVPN Default VRF Route Leaking on the DCI for Internet Connectivity

The EVPN Default VRF Route Leaking feature leak routes between the Default-VRF and Data Center-VRF on the DCI to provide Internet access to data center hosts.

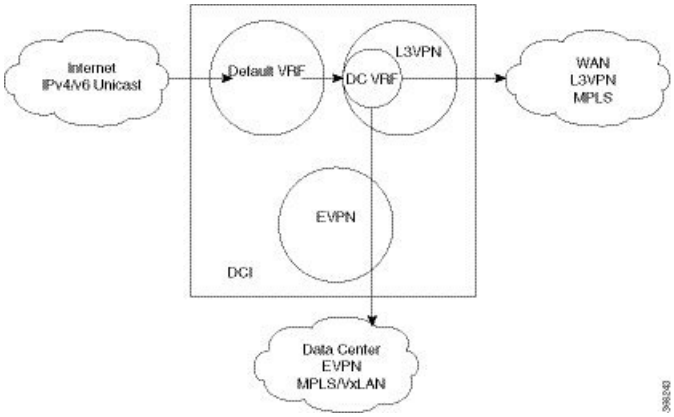
This feature is enabled by:

- Leaking routes from Default-VRF to Data Center-VRF
- Leaking routes to Default-VRF from Data Center-VRF

Leaking Routes from Default-VRF to Data Center-VRF

This section explains the process of leaking Default-VRF routes to Data Center-VRF.

Figure 8: Leaking Routes from Default-VRF to Data Center-VRF



Step 1 The Internet routes are present in the Default-VRF on the DCI.

Note A static default-route (0/0) can be configured under Default-VRF router static address-family configuration and redistributed to BGP.

Step 2 A route-policy is configured to select the routes to be leaked from Default-VRF to Data Center-VRF.

Example:

```
route-policy import-from-default-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    pass
  endif
end-policy
!

route-policy import-from-default-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    pass
  endif
end-policy
!
```

Note Instead of leaking the internet routes, you can leak the default-route 0/0 from Default-VRF to Data Center-VRF using the following policy.

```
route-policy import-from-default-policy
  if destination in (0.0.0.0/0) then
    pass
  endif
end-policy
!

route-policy import-from-default-policy-v6
  if destination in (0::0/0) then
    pass
  endif
end-policy
!
```

Step 3 Leak Default-VRF routes specified in the route-policy to Data Center-VRF by configuring **import from default-vrf route-policy import-from-default-policy(-v6)** under Data Center VRF address-family configuration mode.

Example:

```
vrf data-center-vrf
 address-family ipv4 unicast
  import from default-vrf route-policy import-from-default-policy
!
address-family ipv6 unicast
  import from default-vrf route-policy import-from-default-policy-v6
!
```

Step 4 Advertise the leaked (Default-VRF) routes in the Data Center-VRF as EVPN routes towards Data Center routers by configuring **advertise-as-vpn** option.

Example:

```
vrf data-center-vrf
 address-family ipv4 unicast
  import from default-vrf route-policy import-from-default-policy advertise-as-vpn
!
address-family ipv6 unicast
  import from default-vrf route-policy import-from-default-policy-v6 advertise-as-vpn
!
```

Note To advertise any routes from L3VPN address-family to EVPN peers, use **advertise vpnv4/vpnv6 unicast re-originated [stitching-rt]** command under neighbor address-family L2VPN EVPN.

EVPN Default-originate

Instead of advertising the Default-VRF routes towards Data Center routers, default-originate can be configured under the EVPN neighbor address-family to advertise the default route. When default-originate is configured under the neighbor address-family for EVPN/L3VPN, there is no need to advertise the Default-VRF leaked routes to the data center and **advertise-as-vpn** need not be configured.

Example:

```
router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
```

```
     default-originate

 vrf data-center-vrf
  rd auto
  address-family ipv4 unicast
    allow vpn default-originate
 !
  address-family ipv6 unicast
    allow vpn default-originate
```

Step 5 To block advertisement of the Default-VRF leaked routes towards a particular EVPN/L3VPN peer, use **advertise vpnv4/vpnv6 unicast imported-from-default-vrf disable** command under respective neighbor address-family.

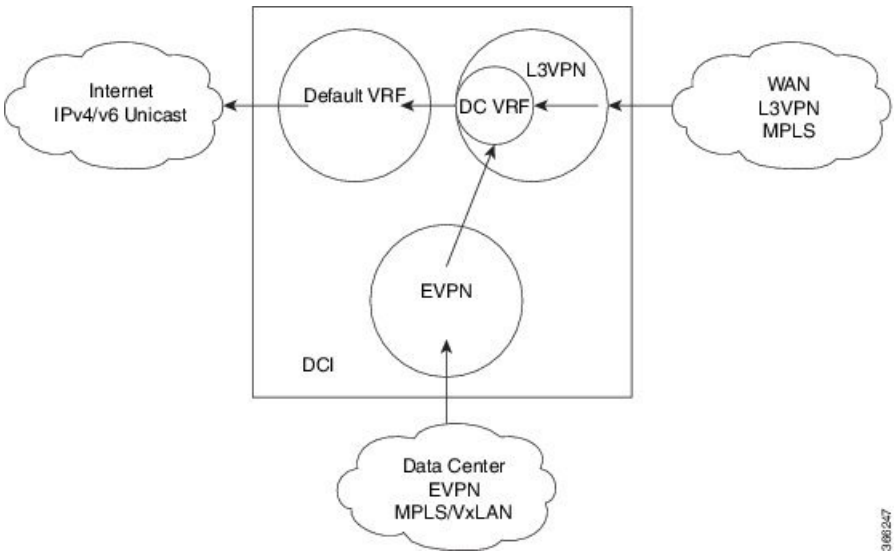
Example:

```
router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
      advertise vpnv4 unicast imported-from-default-vrf disable
      advertise vpnv6 unicast imported-from-default-vrf disable
!
router bgp 100
  neighbor 60.0.0.1
    address-family vpnv4 unicast
      advertise vpnv4 unicast imported-from-default-vrf disable
    address-family vpnv6 unicast
      advertise vpnv6 unicast imported-from-default-vrf disable
```

Leaking Routes to Default-VRF from Data Center-VRF

This section explains the process of leaking Data Center-VRF routes to Default-VRF.

Figure 9: Leaking Routes to Default-VRF from Data Center-VRF



Step 1 Data Center routes are received on the DCI as EVPN Route-type 2 and Route-type 5 NLRI and imported to the Data Center VRFs.

Step 2 A route-policy is configured to select the routes to be leaked from Data Center-VRF to Default-VRF.

Example:

```
route-policy export-to-default-policy
  if destination in (200.47.0.0/16, 200.168.0.0/16) then
    pass
  endif
end-policy
!

route-policy export-to-default-policy-v6
  if destination in (200:47::0/64, 200:168::0/64) then
    pass
  endif
end-policy
!
```

Step 3 Leak Data Center-VRF routes specified in the above policy to Default-VRF by configuring **export to default-vrf route-policy export-to-default-policy(-v6) [allow-imported-vpn]** under Data Center-VRF address-family configuration mode.

Normally only local and CE VRF routes are allowed to be leaked to the Default-VRF, but **allow-imported-vpn** configuration enables leaking of EVPN/L3VPN imported routes to the Default-VRF.

Example:

```
vrf data-center-vrf
  address-family ipv4 unicast
    export to default-vrf route-policy export-to-default-policy [allow-imported-vpn]
  !
  address-family ipv6 unicast
    export to default-vrf route-policy export-to-default-policy-v6 [allow-imported-vpn]
  !
```

Step 4 The Leaked routes in the Default VRF are advertised to the Internet.

Note Instead of advertising the leaked routes to the Internet, an aggregate can be configured and advertised to the Internet.

Sample Router Configuration

The following sample configuration specifies how EVPN Default VRF Route Leaking feature is configured on a DCI router to provide Internet access to the data center hosts.

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from default-vrf route-policy import-from-default-policy advertise-as-vpn
    export to default-vrf route-policy export-to-default-policy allow-imported-vpn
  !
```

```
  address-family ipv6 unicast
    import from default-vrf route-policy import-from-default-policy-v6 advertise-as-vpn
    export to default-vrf route-policy export-to-default-policy-v6 allow-imported-vpn
  !

route-policy import-from-default-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    pass
  endif
end-policy
!

route-policy import-from-default-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    pass
  endif
end-policy
!

route-policy export-to-default-policy
  if destination in (200.47.0.0/16, 200.168.0.0/16) then
    pass
  endif
end-policy
!

route-policy export-to-default-policy-v6
  if destination in (200:47::0/64, 200:168::0/64) then
    pass
  endif
end-policy
!

router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
      import stitching-rt re-originate
      advertise vpnv4 unicast re-originated stitching-rt
      advertise vpnv6 unicast re-originated stitching-rt

  neighbor 60.0.0.1
    address-family vpnv4 unicast
      import re-originate stitching-rt
      advertise vpnv4 unicast re-originated
      advertise vpnv4 unicast imported-from-default-vrf disable

    address-family vpnv6 unicast
      import re-originate stitching-rt
      advertise vpnv6 unicast re-originated
      advertise vpnv6 unicast imported-from-default-vrf disable
```

Sample Router Configuration: with default-originate

The following sample configuration specifies how EVPN Default VRF Route Leaking feature is configured along with default-originate on a DCI router to provide Internet access to data center hosts.

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from default-vrf route-policy import-from-default-policy <= Remove
advertise-as-vpn=>
    export to default-vrf route-policy export-to-default-policy allow-imported-vpn
  !
  address-family ipv6 unicast
```

```
    import from default-vrf route-policy import-from-default-policy-v6 <= Remove
advertise-as-vpn=>
    export to default-vrf route-policy export-to-default-policy-v6 allow-imported-vpn
  !
route-policy import-from-default-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    pass
  endif
end-policy
!
route-policy import-from-default-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    pass
  endif
end-policy
!
route-policy export-to-default-policy
  if destination in (200.47.0.0/16, 200.168.0.0/16) then
    pass
  endif
end-policy
!
route-policy export-to-default-policy-v6
  if destination in (200:47::0/64, 200:168::0/64) then
    pass
  endif
end-policy
!
router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
      import stitching-rt re-originate
      advertise vpnv4 unicast re-originated stitching-rt
      advertise vpnv6 unicast re-originated stitching-rt
      default-originate <= Added=>

  neighbor 60.0.0.1
    address-family vpnv4 unicast
      import re-originate stitching-rt
      advertise vpnv4 unicast re-originated
      advertise vpnv4 unicast imported-from-default-vrf disable

    address-family vpnv6 unicast
      import re-originate stitching-rt
      advertise vpnv6 unicast re-originated
      advertise vpnv6 unicast imported-from-default-vrf disable

 vrf data-center-vrf
  rd auto
  address-family ipv4 unicast
    allow vpn default-originate <= Added=>
  !
  address-family ipv6 unicast
    allow vpn default-originate <= Added=>
```

EVPN Service VRF Route Leaking

The EVPN Service VRF Route Leaking feature enables connectivity to the services in the Service VRF to customers in EVPN Data Center VRF. The Service VRF and Data Center VRF routes can be IPv4 and/or IPv6 addresses. The Services VRF is any L3 VRF providing services reachable through connected, static, re-distributed IGP or BGP routes.

This feature leaks routes between Data Center VRF and Service VRF, enabling the EVPN/L3VPN hosts to access the Services in the Service VRF. This feature rely on Border Gateway Protocol (BGP) VRF extranet feature that imports routes between two VRFs.

The import process installs the Data Center VRF routes in a Service VRF table or a Service VRF routes in the Data Center VRF table, providing connectivity.

The BGP Service VRF route leaking feature is enabled by:

- Importing routes from Service VRF to Data Center VRF and advertising it as EVPN/L3VPN route from Data Center VRF.
    - Importing Service VRF routes to Data Center VRF by attaching Data Center VRF import RTs to Service VRF routes.

      This can be achieved by configuring one or more Data Center VRF import RTs as export RT of Service VRF, or configuring a Service VRF export route-policy to attach import RT EXTCOMM to Service VRF routes matching the import RTs of Data Center VRF using the following command in Service VRF address-family configuration mode.

      export route-policy service-vrf-export-route-policy-name

      Where the route-policy "service-vrf-export-route-policy-name" attaches the RT EXTCOMM matching the one or more import RTs of Data Center VRF to Service VRF routes.

    - Advertising Data Center VRF imported routes that are exported from Service VRFs as EVPN/L3VPN NLRI from Data Center VRF using the following command in Data Center VRF address-family configuration mode.

      import from vrf advertise-as-vpn

      If the **advertise-as-vpn** keyword is used, the paths imported from the Service VRF to the Data Center VRF are advertised to the (EVPN/L3VPN) PEs as well as to the CEs. If the **advertise-as-vpn** keyword is not used, the paths imported from the Service VRF to the Data Center VRF are not advertised to the PEs. However, the paths are still advertised to the CEs.

    - Block advertising Data Center VRF leaked routes from being advertised to a neighbor using the following command in neighbor address-family configuration mode.

      advertise vpnv4/vpnv6 unicast imported-from-vrf disable

      A new command **advertise vpnv4/vpnv6 unicast imported-from-vrf disable** is added under neighbor address-family configuration mode for EVPN and VPNv4/VPNv6 unicast to disable advertisement of VRF to VRF leaked routes to that neighbor.

- Importing EVPN/L3VPN routes from Data Center VRF to Service VRF
    - Importing EVPN/L3VPN routes from Data Center VRF to Service VRF by attaching Service VRF import RTs.

      This can be achieved by configuring one or more Service VRF import RTs as export RT of Data Center VRF, or configuring a Data Center VRF export route-policy to attach import RT EXTCOMM to Data Center VRF routes matching the import RTs of Service VRF using the following command in Data Center VRF address-family configuration mode.

      export route-policy data-center-vrf-export-route-policy-name

      The route-policy "data-center-vrf-export-route-policy-name" attaches the RT EXTCOMM matching one or more import RTs of Service VRF.

- Allow leaking of Data Center VRF routes to Service VRF by using the following command in Data Center VRF address-family configuration mode.

  export to vrf allow-imported-vpn

Note In order to prevent un-intended import of routes to VRFs, select unique RT's to import routes between Service VRF and Data Center VRF, which are not used for normal import of VPN/EVPN routes to Data Center VRFs.

The Extranet Route Leaking feature enables leaking of local and CE routes from one VRF to another VRF. A new command **export to vrf allow-imported-vpn** is added to enable the leaking of EVPN and L3VPN imported/re-originated Data Center VRF routes to the Service VRF.

Note A route-policy is preferred to filter the imported routes. This reduces the risk of unintended import of routes between the Data Center VRF and the Service VRF, and the corresponding security issues. There is no hard limit on the number of prefixes that can be imported. The import creates a new prefix in the destination VRF, which increases the total number of prefixes and paths.

Note This feature does not advertise EVPN/L3VPN PE routes imported to Data Center VRF and leaked to Service VRF as EVPN/L3VPN PE route.

EVPN Service VRF Route Leaking on the DCI for Service Connectivity

The EVPN Service VRF Route Leaking feature leaks routes between the Service VRF and Data Center VRF on the DCI to provide access to Services to data center hosts.

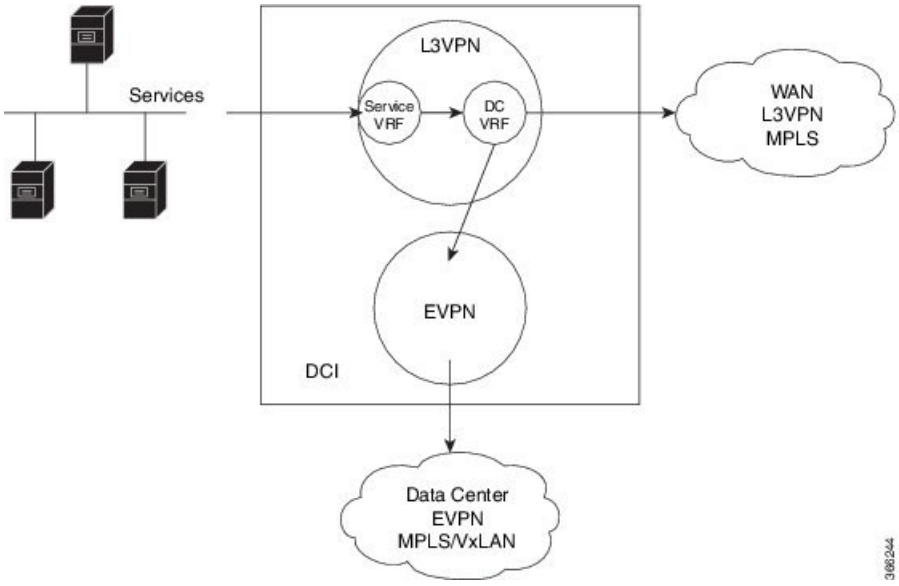
This feature is enabled by:

- Leaking routes from Service VRF to Data Center VRF
- Leaking routes to Service VRF from Data Center VRF

Leaking Routes from Service VRF to Data Center VRF

This section explains the process of leaking Service VRF routes to Data Center VRF.

Figure 10: Leaking Routes from Service VRF to Data Center VRF



Step 1 The Service routes are present in the Service VRF on the DCI.

Step 2 A route-policy is configured to select the routes to be leaked from Service VRF to Data Center VRF.

Example:

```
route-policy service-vrf-export-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    set extcommunity rt (1:1) additive <--- matches import RT of Data Center-VRF
  endif
end-policy
!
route-policy service-vrf-export-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    set extcommunity rt (1:1) additive <--- matches import RT of Data Center-VRF
  endif
end-policy
!
```

Step 3 Leak Service VRF routes specified in the route-policy to Data Center VRF by configuring **export route-policy service-vrf-export-policy(-v6)** under Service VRF address-family configuration mode.

Example:

```
vrf service-vrf
  address-family ipv4 unicast
    import route-target
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy
    export route-target
      3:1
      4:1 stitching
   !
  address-family ipv6 unicast
    import route-target
```

```
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy-v6
    export route-target
      3:1
      4:1 stitching
   !
```

Step 4 Advertise the leaked (Service VRF) routes in the Data Center VRF as EVPN/L3VPN routes towards Data Center routers by configuring **import from vrf advertise-as-vpn** under Data Center VRF address-family configuration mode..

Example:

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-target
      100:1
      200:1 stitching
  !
  address-family ipv6 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-target
      100:1
      200:1 stitching
  !
```

Note To advertise any routes from L3VPN address-family to EVPN peers, use **advertise vpnv4/vpnv6 unicast re-originated [stitching-rt]** command under neighbor address-family L2VPN EVPN.

EVPN Default-originate

Instead of advertising the Service VRF routes towards Data Center routers, default-originate can be configured under the EVPN neighbor address-family to advertise the default route. When **allow vpn default-originate** is configured under the Data Center VRF, there is no need to advertise the Service VRF leaked routes to the data center and **advertise-as-vpn** need not be configured.

Example:

```
router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
      default-originate

 vrf data-center-vrf
  rd auto
  address-family ipv4 unicast
    allow vpn default-originate
 !
  address-family ipv6 unicast
    allow vpn default-originate
```

Step 5 To block advertisement of the Service VRF leaked routes towards a particular EVPN/L3VPN peer, use **advertise vpnv4/vpnv6 unicast imported-from-vrf disable** command under respective neighbor address-family.

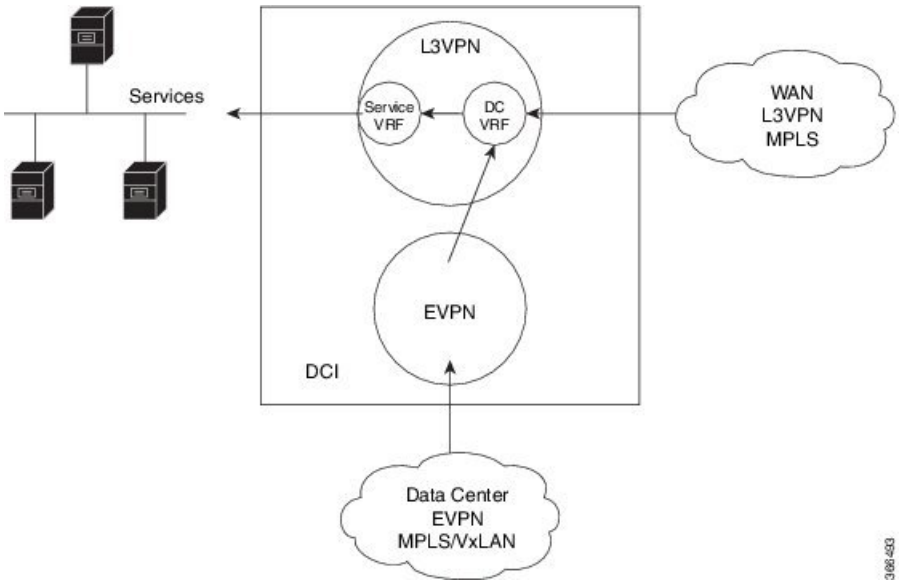
Example:

```
router bgp 100
  neighbor 40.0.0.1
    address-family l2vpn evpn
      import stitching-rt re-originate
      advertise vpnv4 unicast re-originated stitching-rt
      advertise vpnv4 unicast imported-from-vrf disable
      advertise vpnv6 unicast re-originated stitching-rt
      advertise vpnv6 unicast imported-from-vrf disable
    !
router bgp 100
  neighbor 60.0.0.1
    address-family vpnv4 unicast
      import re-originate stitching-rt
      advertise vpnv4 unicast re-originated
      advertise vpnv4 unicast imported-from-vrf disable
    address-family vpnv6 unicast
      import re-originate stitching-rt
      advertise vpnv6 unicast re-originated
      advertise vpnv6 unicast imported-from-vrf disable
```

Leaking Routes to Service VRF from Data Center VRF

This section explains the process of leaking Data Center VRF routes to Service VRF.

Figure 11: Leaking Routes to Service VRF from Data Center VRF



Step 1 Data Center routes are received on the DCI as EVPN Route-type 2 and Route-type 5 NLRI and imported to the Data Center VRFs.

Step 2 A route-policy is configured to select the routes to be leaked from Data Center VRF to Service VRF.

The policy attaches RT EXTCOMM to Data Center VRF routes matching one or more import RT of the Service VRF.

Example:

```
route-policy data-center-vrf-export-policy
  if destination in (200.47.0.0/16) then <--- EVPN PE route
    set extcommunity rt (4:1) additive <--- matches import stitching-RT of service-VRF
  if destination in (200.168.0.0/16) then <--- VPNv4 PE route
    set extcommunity rt (3:1) additive <--- matches import RT of service-VRF
  endif
end-policy
!
route-policy data-center-vrf-export-policy-v6
  if destination in (200:47::0/64) then <--- EVPN PE route
    set extcommunity rt (4:1) additive <--- matches import stitching-RT of service-VRF
  elseif destination in (200:168::0/64) then <--- VPNv6 PE route
    set extcommunity rt (3:1) additive <--- matches import RT of service-VRF
  endif
end-policy
!
```

Note An EVPN/L3VPN route received from a neighbor configured locally with "import stitching-rt re-originate" is imported to Data Center VRF if the route's RT EXTCOMM matches with one or more Data Center VRF import stitching RTs, and is leaked to Service VRF if the Data Center VRF route's RT EXTCOMM matches with one or more Service VRF import stitching RTs.

Step 3 Leak Data Center VRF routes specified in the above policy to Service VRF by configuring **export route-policy data-center-vrf-export-policy(-v6)** under Data Center VRF address-family configuration mode.

Normally only local and CE VRF routes are allowed to be leaked to the Service VRF, but **allow-imported-vpn** configuration enables leaking of EVPN/L3VPN imported routes to the Service VRF.

Example:

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !
  address-family ipv6 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy-v6
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !
```

Step 4 The Data Center VRF leaked routes in the Service VRF are advertised to Service VRF CE peers.

Sample Router Configuration

The following sample configuration specifies how EVPN Service VRF Route Leaking feature is configured on a DCI router providing access to data center hosts to Services in the Service VRF.

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !
  address-family ipv6 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy-v6
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !

vrf service-vrf
  address-family ipv4 unicast
    import route-target
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy
    export route-target
      3:1
      4:1 stitching
   !
  address-family ipv6 unicast
    import route-target
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy-v6
    export route-target
      3:1
      4:1 stitching
   !

route-policy data-center-vrf-export-policy
  if destination in (200.47.0.0/16) then
    set extcommunity rt (4:1) additive
  if destination in (200.168.0.0/16)
    set extcommunity rt (3:1) additive
  endif
end-policy
!
```

```
route-policy data-center-vrf-export-policy-v6
  if destination in (200:47::0/64) then
    set extcommunity rt (4:1) additive
  elseif destination in (200:168::0/64)
    set extcommunity rt (3:1) additive
  endif
end-policy
!

route-policy service-vrf-export-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    set extcommunity rt (1:1) additive
  endif
end-policy
!

route-policy service-vrf-export-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    set extcommunity rt (1:1) additive
  endif
end-policy
!

route-policy pass-all
  pass
end-policy
!

router bgp 100
  neighbor 40.0.0.1
    remote-as 100
    address-family l2vpn evpn
      import stitching-rt re-originate
      advertise vpnv4 unicast re-originated stitching-rt
      advertise vpnv6 unicast re-originated stitching-rt
    !
  neighbor 60.0.0.1
    remote-as 200
    address-family vpnv4 unicast
      import re-originate stitching-rt
      route-policy pass-all in
      route-policy pass-all out
      advertise vpnv4 unicast re-originated
      advertise vpnv4 unicast imported-from-vrf disable
    address-family vpnv6 unicast
      import re-originate stitching-rt
      route-policy pass-all in
      route-policy pass-all out
      advertise vpnv6 unicast re-originated
      advertise vpnv6 unicast imported-from-vrf disable
```

Sample Router Configuration: with default-originate

The following sample configuration specifies how EVPN Service VRF Route Leaking feature is configured along with default-originate on a DCI router to provide data center hosts access to Services in the Service VRF..

```
vrf data-center-vrf
  address-family ipv4 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
```

```
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !
  address-family ipv6 unicast
    import from vrf advertise-as-vpn
    import route-target
      1:1
      100:1
      200:1 stitching
    export route-policy data-center-vrf-export-policy-v6
    export to vrf allow-imported-vpn
    export route-target
      100:1
      200:1 stitching
  !

vrf service-vrf
  address-family ipv4 unicast
    import route-target
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy
    export route-target
      3:1
      4:1 stitching
   !
  address-family ipv6 unicast
    import route-target
      3:1
      4:1 stitching
    export route-policy service-vrf-export-policy-v6
    export route-target
      3:1
      4:1 stitching
   !

route-policy data-center-vrf-export-policy
  if destination in (200.47.0.0/16) then
    set extcommunity rt (4:1) additive
  if destination in (200.168.0.0/16) then
    set extcommunity rt (3:1) additive
  endif
end-policy
!

route-policy data-center-vrf-export-policy-v6
  if destination in (200:47::0/64) then
    set extcommunity rt (4:1) additive
  elseif destination in (200:168::0/64) then
    set extcommunity rt (3:1) additive
  endif
end-policy
!

route-policy service-vrf-export-policy
  if destination in (100.10.0.0/16, 100.20.0.0/16) then
    set extcommunity rt (1:1) additive
  endif
end-policy
```

```
!

route-policy service-vrf-export-policy-v6
  if destination in (100:10::0/64, 100:20::0/64) then
    set extcommunity rt (1:1) additive
  endif
end-policy
!

route-policy pass-all
  pass
end-policy
!

router bgp 100
  neighbor 40.0.0.1
    remote-as 100
    address-family l2vpn evpn
      import stitching-rt re-originate
      advertise vpnv4 unicast re-originated stitching-rt
      advertise vpnv4 unicast imported-from-vrf disable
      advertise vpnv6 unicast re-originated stitching-rt
      advertise vpnv6 unicast imported-from-vrf disable
      default-originate <= Added=>
    !
  neighbor 60.0.0.1
    remote-as 200
    address-family vpnv4 unicast
      import re-originate stitching-rt
      route-policy pass-all in
      route-policy pass-all out
      advertise vpnv4 unicast re-originated
      advertise vpnv4 unicast imported-from-vrf disable
      default-originate <= Added=>
    address-family vpnv6 unicast
      import re-originate stitching-rt
      route-policy pass-all in
      route-policy pass-all out
      advertise vpnv6 unicast re-originated
      advertise vpnv6 unicast imported-from-vrf disable
      default-originate <= Added=>

  vrf data-center-vrf
    rd auto
    address-family ipv4 unicast
      allow vpn default-originate <= Added=>
    !
    address-family ipv6 unicast
      allow vpn default-originate <= Added=>
```