



# Configuring MPLS over FlexVPN

**Last Published Date: March 28, 2014**

The MPLS over FlexVPN feature implements Multiprotocol Label Switching (MPLS) over a dynamically established IPsec tunnel thereby supporting duplicate address spaces.

- [Prerequisites for MPLS over FlexVPN, on page 1](#)
- [Information About Configuring MPLS over FlexVPN, on page 1](#)
- [How to Configure MPLS over FlexVPN, on page 4](#)
- [Configuration Examples for Configuring MPLS over FlexVPN, on page 5](#)
- [Additional References for Configuring MPLS over FlexVPN, on page 13](#)
- [Feature Information for Configuring MPLS over FlexVPN, on page 14](#)

## Prerequisites for MPLS over FlexVPN

- Internet Key Exchange Version 2 (IKEv2) and IPsec must be configured.
- MPLS must be configured.
- NHRP redirect must be configured.

## Information About Configuring MPLS over FlexVPN

### MPLS and FlexVPN

Network domains having overlapping addressing spaces use VPN routing and forwarding (VRF) to segregate traffic so that data intended for one domain does not enter another domain. Data security between the provider-edge (PE) devices is achieved by defining an tunnel interface with IPsec protection for every VRF. This ensures that traffic from every domain passes over the corresponding IPsec tunnel. However as the number of domains and nodes grow in a network, this may not be scalable because every protected domain requires a separate IPsec tunnel and an interface.

Multiprotocol Label Switching (MPLS) provides the ability to assign labels per VRF or per prefix, which identifies the correct VRF into which data needs to be routed to. This can be achieved with just a single MPLS-aware interface having IPsec protection and a single IPsec tunnel between the PEs.

The MPLS over FlexVPN feature provides a solution to achieve communication between overlapping addresses in customer networks when a remote customer network needs to be discovered dynamically using Next Hop

Resolution Protocol (NHRP) and at the same time secure the data traffic between the PE devices using IPsec. This solution can be used by customers who have deployed MPLS network and want to extend their MPLS network to a newly configured network (determined dynamically) in a different region over the Internet in a secure way.

The components of the MPLS over FlexVPN solution are as follows:

- IPsec—Secures the data traffic between the spoke and the hub and between the spokes after the remote spoke is discovered dynamically.
- Internet Key Exchange Version 2 (IKEv2)—Adds static routes to the peer's tunnel overlay address as a directly connected route. This route results in adding an implicit null label to the Label Information Base (LIB) for the peer's tunnel overlay address.




---

**Note** IKEv2 is used instead of LDP because LDP involves establishing TCP channel with every LDP neighbor. Enabling LDP keeps the spoke-to-spoke channel active due to the LDP hello traffic thereby never bringing down the spoke-to-spoke channel. Therefore, the **mpls ip** command must never be executed on the tunnel interface or virtual template when configuring the MPLS over FlexVPN feature.

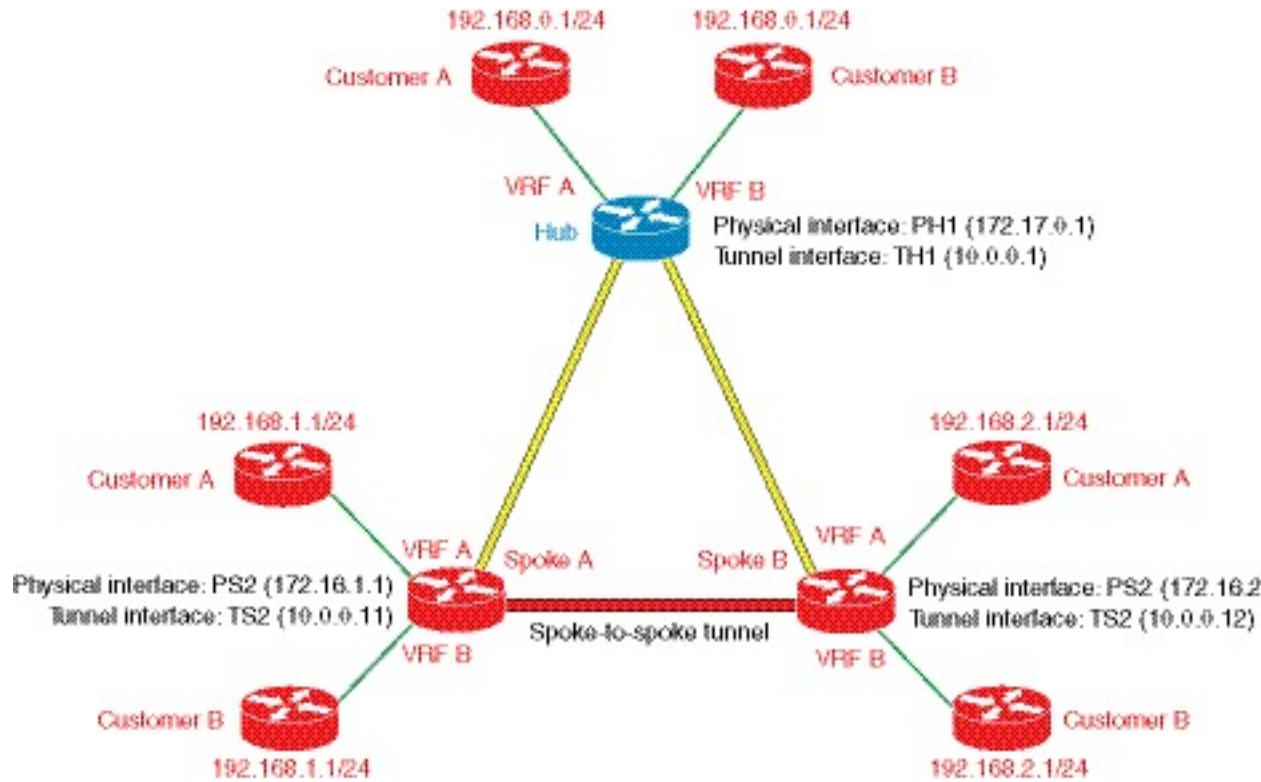
---

- NHRP—Used to resolve the remote overlay address and dynamically discover the transport end point needed to establish a secure tunnel. If a multipoint generic routing encapsulation (GRE) interface is used, the tunnel end point database stores the mapping between the overlay and corresponding nonbroadcast multiaccess (NBMA) address.
- MPLS—Enables MPLS tag switching for data packets. By default, Label Distribution Protocol (LDP) is not enabled and is not enabled between the spokes because LDP keepalive will try to keep the spoke-spoke tunnel up and is not desired in the absence of data traffic.
- MPLS Forwarding Infrastructure (MFI)—Allocates and releases labels by the applications; NHRP is an application that call MFI for label management.
- Multiprotocol BGP (MP-BGP)—Distributes overlay labels for the network on different VRFs.

## Working of MPLS over FlexVPN

The following figure along description explains the working of MPLS over FlexVPN solution:

Figure 1: Spoke to Hub to Spoke Topology



The MPLS over FlexVPN solution has the following assumptions:

- Multiprotocol BGP (MP-BGP) allows distributing labels per VPN routing and forwarding (VRF) or per prefix.
  - Label 10 is assigned to VRF A for packets that arrive from hub to spoke A.
  - Label 20 is assigned to VRF A for packets that arrive from the hub to spoke B.
  - Label 30 is assigned to VRF A on the hub for packets that arrive from spoke A to the hub.
  - Label 40 is assigned to VRF B on the hub for packets that arrive from spoke B to the hub.
1. IKEv2 and IPsec security associations are established from each spoke to the hub. IKEv2 installs implicit null label values for the spoke's overlay address that is received in the mode config reply and mode config set.



**Note** Implicit null label is installed since the spoke and hub are always next-hop to each other in the overlay space.

2. MP-BGP exchanges the label per VRF or label per prefix with all the VRFs.
3. After the labels and routes have been exchanged, data forwarding begins. When the first data packet destined for 192.168.2.1 arrives on spoke A on VRF A, it is forwarded to the hub. The packet is label encapsulated using generic routing encapsulation (GRE), only containing the overlay label, and encrypted.
4. The data packet is decrypted when it reaches the hub on the physical (virtual access) interface or the tunnel interface which is 172.17.0.1 and 10.0.0.1 respectively. The overlay label is looked up in the hub, the packet is encapsulated using GRE, encrypted and sent to spoke B.

5. An NHRP redirect packet is sent from the hub to spoke A. As label 30 identifies the VRF on which the data packet arrived, the VRF information is conveyed to NHRP.
6. NHRP processes the redirect packet and triggers an NHRP resolution request. An NHRP mapping entry is created and VRF A is associated for the prefix that needs to be resolved.
7. The resolution request is sent to the hub, which looks up its overlay label and sends the resolution request to the appropriate destination, which in this case is Spoke B.
8. NHRP resolution request arrives on Spoke B and creates a virtual access interface or an multipoint GRE (mGRE) interface on Spoke B.
9. An IKEv2 and IPsec session is initiated from Spoke B to Spoke A resulting in the creation of a virtual access interface or mGRE interface on Spoke A. NHRP adds the route for IP address of Spoke A tunnel via the newly created virtual access interface.
10. NHRP resolution reply from Spoke B carries the label value that may be used by Spoke A for sending data over the spoke-to-spoke tunnel. Therefore, NHRP allocates a label from the MPLS forwarding instance (MFI) and sends this label information to Spoke A to be used for the spoke-to-spoke tunnel.




---

**Note** MFI tracks the labels. If a label is already allocated and assigned to MP-BGP for a particular VRF, the label is returned to NHRP. MFI tracks the number of applications using this a particular label and returns the label back to pool only when all the applications have released the label.

---

11. NHRP resolution reply also contains an implicit null label for the IP address of the virtual access interface or mGRE interface on Spoke B. In this example, the reply would be 192.168.2.0/24, label 40, 10.0.0.12, 172.16.2.1, [implicit-NULL].
12. NHRP resolution reply is received at the virtual access interface or mGRE interface on Spoke A. The NHRP request ID present in reply packet is matched with the request ID of the request that was initially sent by Spoke A to know the VRF for which the request was sent. NHRP cache is looked up to find the NHRP entry and the entry is termed “Complete”. NHRP inserts a route into the VRF routing table with the label information.
13. Routes and labels are setup between Spoke A and Spoke B. Data is now label encapsulated and encrypted over the spoke-to-spoke dynamically established tunnel between Spoke A and Spoke B.

## IVRF Support for FlexVPN

The Inside VPN Routing and Forwarding (IVRF) support for FlexVPN provides the capability of performing the following NHRP routing operations in the IVRF configured on the tunnel interface:

- Sending NHRP resolution request after performing the route lookup.
- Forwarding of NHRP resolution request on the hub.
- Creating an H route or next-hop override (NHO) in the IVRF when creating a shortcut tunnel
- Deleting the H route or NHO from the IVRF when the shortcut tunnel is deleted

# How to Configure MPLS over FlexVPN

## Configuring MPLS over FlexVPN

Perform this task to configure MPLS over FlexVPN.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **mpls nhrp**
5. **end**
6. **show mpls forwarding-table**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface tunnel</b> <i>number</i> <b>Example:</b> Device(config)# interface tunnel 1	Configures the FlexVPN client interface and enters interface configuration mode.
<b>Step 4</b>	<b>mpls nhrp</b> <b>Example:</b> Device(config-if)# mpls nhrp	Enables MPLS tag switching without enabling Label Distribution Protocol (LDP).
<b>Step 5</b>	<b>end</b> <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and returns to global configuration mode.
<b>Step 6</b>	<b>show mpls forwarding-table</b> <b>Example:</b> Device# show mpls forwarding-table	Displays information about the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB).

# Configuration Examples for Configuring MPLS over FlexVPN

## Example: Configuring MPLS over FlexVPN

The following example shows how to transport multiple customer VRFs on FlexVPN leveraging MPLS functionality. The following is the configuration on spoke 1.

```
hostname R3-Spoke1
boot-start-marker
```

```

boot-end-marker
!
!
vrf definition cust1
 rd 1:1
  route-target export 1:1
  route-target import 1:1
  !
  address-family ipv4
  exit-address-family
!
vrf definition cust2
 rd 2:2
  route-target export 2:2
  route-target import 2:2
  !
  address-family ipv4
  exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
mpls ldp loop-detection
!
crypto pki trustpoint CA
 enrollment url http://172.16.1.1:80
 password
 fingerprint E0AFefd7F08070BAB33C8297C97E6457
 subject-name cn=R3-spoke.cisco.com,OU=FLEX,O=Cisco
 revocation-check crl none
!
crypto pki certificate map mymap 10
 subject-name co ou = flex
!
crypto pki certificate chain CA
 certificate 03
 certificate ca 01
crypto ikev2 authorization policy default
 route set interface
!
crypto ikev2 profile default
 match certificate mymap
 identity local fqdn R3-Spoke.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 dpd 60 2 on-demand
 aaa authorization group cert list default default
!
!
!
!
crypto ipsec profile default
 set ikev2-profile default
!
!
!
!
!
interface Tunnel0

```

```

ip address negotiated
mpls bgp forwarding
tunnel source Ethernet0/0
tunnel destination 172.16.0.1
tunnel protection ipsec profile default
!
interface Ethernet0/0
description WAN
ip address 172.16.1.103 255.255.255.0
!
interface Ethernet0/1
description LAN
no ip address
no ip unreachable
!
interface Ethernet0/1.10
encapsulation dot1Q 10
vrf forwarding cust1
ip address 192.168.113.1 255.255.255.0
!
interface Ethernet0/1.20
encapsulation dot1Q 20
vrf forwarding cust2
ip address 192.168.123.1 255.255.255.0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.1 remote-as 10
neighbor 10.0.0.1 ebgp-multihop 255
neighbor 10.0.0.1 update-source Tunnel0
!
address-family ipv4
neighbor 10.0.0.1 activate
exit-address-family
!
address-family vpv4
neighbor 10.0.0.1 activate
neighbor 10.0.0.1 send-community both
exit-address-family
!
address-family ipv4 vrf cust1
redistribute connected
exit-address-family
!
address-family ipv4 vrf cust2
redistribute connected
exit-address-family
!
ip route 10.0.0.1 255.255.255.255 Tunnel0 name workaround
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB

```

The following is spoke 2 configuration.

```

hostname R4-Spoke
!
vrf definition cust1
rd 1:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family
!
vrf definition cust2
rd 2:2

```

```

route-target export 2:2
route-target import 2:2
!
address-family ipv4
exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
crypto pki token default removal timeout 0
!
crypto pki trustpoint CA
  enrollment url http://172.16.1.1:80
  password
  fingerprint E0AFEF7F08070BAB33C8297C97E6457
  subject-name cn=R4-Spoke.cisco.com,OU=Flex,O=Cisco
  revocation-check crl none
!
crypto pki certificate map mymap 10
  subject-name co ou = flex
!
crypto pki certificate chain CA
  certificate 04
  certificate ca 01
!
crypto ikev2 authorization policy default
  route set interface
!
crypto ikev2 profile default
  match certificate mymap
  identity local fqdn R4.cisco.com
  authentication local rsa-sig
  authentication remote rsa-sig
  pki trustpoint CA
  dpd 60 2 on-demand
  aaa authorization group cert list default default
  virtual-template 1
!
crypto ipsec profile default
  set ikev2-profile default
!
interface Loopback100
  vrf forwarding cust1
  ip address 192.168.114.1 255.255.255.0
!
interface Loopback101
  vrf forwarding cust2
  ip address 192.168.124.1 255.255.255.0
!
interface Tunnel0
  ip address negotiated
  mpls bgp forwarding
  tunnel source Ethernet0/0
  tunnel destination 172.16.0.1
  tunnel protection ipsec profile default
!
interface Ethernet0/0
  description WAN
  ip address 172.16.1.104 255.255.255.0
!

```



```

interface Ethernet0/1
  description LAN
  ip address 192.168.104.1 255.255.255.0
  !
router bgp 100
  bgp log-neighbor-changes
  neighbor 10.0.0.1 remote-as 10
  neighbor 10.0.0.1 ebgp-multihop 255
  neighbor 10.0.0.1 update-source Tunnel0
  !
  address-family ipv4
    neighbor 10.0.0.1 activate
  exit-address-family
  !
  address-family vpv4
    neighbor 10.0.0.1 activate
    neighbor 10.0.0.1 send-community both
  exit-address-family
  !
  address-family ipv4 vrf cust1
    redistribute connected
  exit-address-family
  !
  address-family ipv4 vrf cust2
    redistribute connected
  exit-address-family
  !
ip route 10.0.0.1 255.255.255.255 Tunnel0
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB

```

The following is the hub configuration.

```

hostname R1-HUB
aaa new-model
!
!
aaa authorization network default local
!
!
clock timezone CET 1 0
!
ip vrf cust1
  rd 1:1
  route-target export 1:1
  route-target import 1:1
!
ip vrf cust2
  rd 2:2
  route-target export 2:2
  route-target import 2:2
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
mpls ldp loop-detection
!
crypto pki trustpoint CA
  enrollment url http://172.16.0.2:80
  password
  fingerprint E0AFefd7f08070BAb33C8297C97E6457
  subject-name CN=R1-HUB.cisco.com,OU=FLEX,OU=VPN,O=Cisco Systems,C=US,L=Linux
  revocation-check crl none

```

```

    rsakeypair R1-HUB.cisco.com 2048
    auto-enroll 95
    !
    !
    crypto pki certificate chain CA
    certificate 02
    certificate ca 01
    !
    redundancy
    !
    !
    !
    crypto ikev2 authorization policy default
    pool mypool
    banner ^C Welcome ^C
    def-domain cisco.com
    !
    !
    !
    crypto ikev2 profile default
    match identity remote fqdn domain cisco.com
    identity local dn
    authentication local rsa-sig
    authentication remote rsa-sig
    pki trustpoint CA
    dpd 60 2 on-demand
    aaa authorization group cert list default default
    virtual-template 1
    !

    crypto ipsec profile default
    set ikev2-profile default
    !
    !
    !
    !
    !
    interface Loopback0
    description VT source interface
    ip address 10.0.0.1 255.255.255.255
    !
    interface Ethernet0/0
    description WAN
    ip address 172.16.0.1 255.255.255.252
    !
    interface Ethernet0/1
    description LAN
    ip address 192.168.100.1 255.255.255.0
    !
    interface Ethernet0/2
    ip vrf forwarding cust1
    ip address 192.168.110.1 255.255.255.0
    !
    interface Ethernet0/3
    ip vrf forwarding cust2
    ip address 192.168.111.1 255.255.255.0
    !
    interface Virtual-Template1 type tunnel
    ip unnumbered Loopback0
    ip nhrp network-id 1
    ip nhrp redirect
    mpls bgp forwarding

```

```

    tunnel protection ipsec profile default
    !
router bgp 10
  bgp log-neighbor-changes
  bgp listen range 0.0.0.0/0 peer-group mpls
  bgp listen limit 5000
  neighbor mpls peer-group
  neighbor mpls remote-as 100
  neighbor mpls transport connection-mode passive
  neighbor mpls update-source Loopback0
  !
  address-family ipv4
    redistribute static route-map global
    neighbor mpls activate
    neighbor mpls next-hop-self
  exit-address-family
  !
  address-family vpv4
    neighbor mpls activate
    neighbor mpls send-community both
  exit-address-family
  !
  address-family ipv4 vrf cust1
    redistribute connected
    redistribute static route-map cust1
    default-information originate
  exit-address-family
  !
  address-family ipv4 vrf cust2
    redistribute connected
    redistribute static route-map cust2
    default-information originate
  exit-address-family
  !
ip local pool mypool 10.1.1.1 10.1.1.254
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 172.16.0.2 name route_to_internet
ip route vrf cust1 0.0.0.0 0.0.0.0 Null0 tag 666 name default_originate
ip route vrf cust2 0.0.0.0 0.0.0.0 Null0 tag 667 name default_originate
!
route-map cust1 permit 10
  match tag 666
!
route-map cust2 permit 10
  match tag 667

```

The following is sample output from the spoke.

```
Device# show ip cef vrf cust1 192.168.110.1
```

```

192.168.110.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
sharing
sources: RIB
feature space:
  IPRM: 0x00018000
  LFD: 192.168.110.0/24 0 local labels
    contains path extension list
ifnums: (none)
  path EF36CA28, path list EF36DEB4, share 1/1, type recursive, for IPv4, flags
must-be-labelled
  MPLS short path extensions: MOI flags = 0x0 label 19

```

```

recursive via 10.0.0.1[IPv4:Default] label 19, fib F0C5926C, 1 terminal fib,
v4:Default:10.0.0.1/32
  path EF36CBE8, path list EF36DFF4, share 1/1, type attached host, for IPv4
  MPLS short path extensions: MOI flags = 0x1 label implicit-null
  attached to Tunnel0, adjacency IP midchain out of Tunnel0 F0481718
  output chain: label 19 label implicit-null TAG midchain out of Tunnel0 F1D97A90 IP adj
out of Ethernet0/0, addr 172.16.1.1 F0481848
R4-Spoke#sh ip bgp vpnv4 all label
Network          Next Hop        In label/Out label
Route Distinguisher: 1:1 (cust1)
  0.0.0.0         10.0.0.1        nolabel/18
  192.168.110.0   10.0.0.1        nolabel/19
  192.168.114.0   0.0.0.0         16/nolabel(cust1)
Route Distinguisher: 2:2 (cust2)
  0.0.0.0         10.0.0.1        nolabel/20
  192.168.111.0   10.0.0.1        nolabel/21
  192.168.124.0   0.0.0.0         17/nolabel(cust2)

```

The following is sample output from the hub.

```

Device# show ip cef vrf cust1 192.168.113.1

192.168.113.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
sharing
sources: RIB, LTE
feature space:
IPRM: 0x00018000
LFD: 192.168.113.0/24 1 local label
local label info: other/25
  contains path extension list
  disposition chain 0xF1E1D9B0
  label switch chain 0xF1E1D9B0
ifnums: (none)
path F16ECA10, path list F16EDFBC, share 1/1, type recursive, for IPv4, flags
must-be-labelled
  MPLS short path extensions: MOI flags = 0x0 label 16
  recursive via 10.1.1.3[IPv4:Default] label 16, fib F0CCD6E8, 1 terminal fib,
v4:Default:10.1.1.3/32
  path F16ECE00, path list F16EE28C, share 1/1, type attached host, for IPv4
  MPLS short path extensions: MOI flags = 0x1 label implicit-null
  attached to Virtual-Access1, adjacency IP midchain out of Virtual-Access1 F04F35D8
  output chain: label 16 label implicit-null TAG midchain out of Virtual-Access1 F1E1DF60
IP adj out of Ethernet0/0, addr 172.16.0.2 F04F3708
R1-HUB#sh ip bgp vpnv4 all
BGP table version is 49, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f
RT-Filter, a additional-path
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop        Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust1)
*> 0.0.0.0         0.0.0.0         0                32768 ?
*> 192.168.110.0   0.0.0.0         0                32768 ?
*> 192.168.113.0   10.1.1.3        0                0 100 ?
*> 192.168.114.0   10.1.1.4        0                0 100 ?
Route Distinguisher: 2:2 (default for vrf cust2)
*> 0.0.0.0         0.0.0.0         0                32768 ?
*> 192.168.111.0   0.0.0.0         0                32768 ?
*> 192.168.123.0   10.1.1.3        0                0 100 ?
*> 192.168.124.0   10.1.1.4        0                0 100 ?
Device# show ip bgp vpnv4 all 192.168.113.1

BGP routing table entry for 1:1:192.168.113.0/24, version 48
Paths: (1 available, best #1, table cust1)

```

```

Advertised to update-groups:
  3
Refresh Epoch 1
100
  10.1.1.3 from *10.1.1.3 (172.16.1.103)
    Origin incomplete, metric 0, localpref 100, valid, external, best
    Extended Community: RT:1:1
    mpls labels in/out 25/16
BGP routing table entry for 2:2:0.0.0.0/0, version 8
Paths: (1 available, best #1, table cust2)
  Advertised to update-groups:
    3
Refresh Epoch 1
Local
  0.0.0.0 from 0.0.0.0 (10.0.0.1)
    Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
    Extended Community: RT:2:2
    mpls labels in/out 20/aggregate(cust2)

```

## Additional References for Configuring MPLS over FlexVPN

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference Commands S to Z</a></li> </ul>
Recommended cryptographic algorithms	<a href="#">Next Generation Encryption</a>

### Standards and RFCs

Standard/RFC	Title
RFC 5586	<i>MPLS Generic Associated Channel</i>

**Technical Assistance**

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

## Feature Information for Configuring MPLS over FlexVPN

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

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

**Table 1: Feature Information for Configuring MPLS over FlexVPN**

Feature Name	Releases	Feature Information
MPLS over FlexVPN		The following commands were introduced or modified: <b>clear ip nhrp</b> , <b>clear ipv6 nhrp</b> , <b>mpls nhrp</b> , <b>show dmvpn</b> , <b>show ip nhrp</b> , <b>show ipv6 nhrp</b> .