



Dynamic Multipoint VPN (DMVPN)

First Published: November 25, 2002

Last Updated: December 11, 2006

The Dynamic Multipoint VPN (DMVPN) feature allows users to better scale large and small IP Security (IPsec) Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP).

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the [“Feature Information for Dynamic Multipoint VPN \(DMVPN\)”](#) section on page 65.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

- [Prerequisites for Dynamic Multipoint VPN \(DMVPN\)](#), page 2
- [Restrictions for Dynamic Multipoint VPN \(DMVPN\)](#), page 2
- [Information About Dynamic Multipoint VPN \(DMVPN\)](#), page 3
- [How to Configure Dynamic Multipoint VPN \(DMVPN\)](#), page 11
- [Configuration Examples for Dynamic Multipoint VPN \(DMVPN\) Feature](#), page 29
- [Additional References](#), page 48
- [Command Reference](#), page 50
- [Feature Information for Dynamic Multipoint VPN \(DMVPN\)](#), page 65
- [Glossary](#), page 66



Americas Headquarters:

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

© 2002–2007 Cisco Systems, Inc. All rights reserved.

Prerequisites for Dynamic Multipoint VPN (DMVPN)

- Before a multipoint GRE (mGRE) and IPsec tunnel can be established, you must define an Internet Key Exchange (IKE) policy by using the **crypto isakmp policy** command.
- For the NAT-Transparency Aware enhancement to work, you must use IPsec transport mode on the transform set. Also, even though NAT-Transparency can support two peers (IKE and IPsec) being translated to the same IP address (using the User Datagram Protocol [UDP] ports to differentiate them [that is, Peer Address Translation (PAT)]), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.
- To enable 2547oDMPVN—Traffic Segmentation Within DMVPN you must configure multiprotocol label switching (MPLS) by using the **mpls ip** command.

Restrictions for Dynamic Multipoint VPN (DMVPN)

- If you use the [Dynamic Creation for Spoke-to-Spoke Tunnels](#) benefit of this feature, you must use IKE certificates or wildcard preshared keys for Internet Security Association Key Management Protocol (ISAKMP) authentication.



Note It is highly recommended that you *do not use* wildcard preshared keys because the attacker will have access to the VPN if one spoke router is compromised.

- GRE tunnel keepalives (that is, the **keepalive** command under a GRE interface) are not supported on point-to-point or multipoint GRE tunnels in a DMVPN Network.
- For best DMVPN functionality, it is recommended that you run the latest Cisco IOS software Release 12.4 mainline, 12.4T, or 12.2(18)SXF.

DMVPN Support on the Cisco 6500 and Cisco 7600

Blade-to-Blade Switchover on the Cisco 6500 and Cisco 7600

- DMVPN does not support blade-to-blade switchover on the Cisco 6500 and Cisco 7600.

Cisco 6500 or Cisco 7600 As a DMVPN Hub

- A Cisco 6500 or Cisco 7600 that is functioning as a DMVPN hub cannot be located behind a NAT router.
- If a Cisco 6500 or Cisco 7600 is functioning as a DMVPN hub, the spoke behind NAT must be a Cisco 6500 or Cisco 7600, respectively, or the router must be upgraded to Cisco IOS software Release 12.3(11)T02 or a later release.

Cisco 6500 or Cisco 7600 As a DMVPN Spoke

- If a Cisco 6500 or Cisco 7600 is functioning as a spoke, the hub cannot be behind NAT.
- If a Cisco 6500 or Cisco 7600 is functioning as a DMVPN spoke behind NAT, the hub must be a Cisco 6500 or Cisco 7600, respectively, or the router must be upgraded to Cisco IOS Release 12.3(11)T02 or a later release.

DMVPN Hub or Spoke Supervisor Engine

- Only a Supervisor Engine 720 can be used as a DMVPN hub or spoke. A Supervisor Engine 2 cannot be used.

Encrypted Multicast with GRE

- Encrypted Multicast with GRE is not supported on the Cisco 6500 nor on the Cisco 7600.

mGRE Interfaces

- If there are two mGRE interfaces on the same DMVPN node and they both do not have a tunnel key, the two mGRE interfaces must each have a unique tunnel source address (or interface) configured.
- On the Cisco 6500 and Cisco 7600, each GRE interface (multipoint or point-to-point) must have a unique tunnel source address (or interface).
- The following commands are not supported under mGRE with DMVPN: **ip tcp adjust-mss**, **qos pre-classify tunnel vrf**, **tunnel path-mtu-discovery**, and **tunnel vrf**.

Quality of Service (QoS)

- You cannot use QoS for DMVPN packets on a Cisco 6500 or Cisco 7600.

Tunnel Key

- The use of a tunnel key on a GRE (multipoint or point-to-point) interface is not supported in the hardware switching ASICs on the Cisco 6500 and Cisco 7600 platforms. If a tunnel key is configured, throughput performance is greatly reduced.
- In Cisco IOS Release 12.3(11)T3 and Release 12.3(14)T, the requirement that a mGRE interface must have a tunnel key was removed. Therefore, in a DMVPN network that includes a Cisco 6500 or Cisco 7600 as a DMVPN node, you should remove the tunnel key from all DMVPN nodes in the DMVPN network, thus preserving the throughput performance on the Cisco 6500 and Cisco 7600 platforms.
- If the tunnel key is not configured on any DMVPN node within a DMVPN network, it must not be configured on all DMVPN nodes with the DMVPN network.

VRF-Aware DMVPN Scenarios

- The **mls mpls tunnel-recir** command must be configured on the provider equipment (PE) DMVPN hub if customer equipment (CE) DMVPN spokes need to “talk” to other CEs across the MPLS cloud.
- The mGRE interface should be configured with a large enough IP maximum transmission unit (1400 packets) to avoid having the route processor doing fragmentation.

Information About Dynamic Multipoint VPN (DMVPN)

To configure the Dynamic Multipoint VPN (DMVPN) feature, you must understand the following concepts:

- [Benefits of Dynamic Multipoint VPN \(DMVPN\), page 4](#)
- [Feature Design of Dynamic Multipoint VPN \(DMVPN\), page 4](#)
- [IPsec Profiles, page 6](#)
- [VRF Integrated DMVPN, page 6](#)

- [DMVPN—Enabling Traffic Segmentation Within DMVPN, page 7](#)
- [NAT-Transparency Aware DMVPN, page 9](#)
- [Call Admission Control with DMVPN, page 10](#)
- [NHRP Rate-Limiting Mechanism, page 10](#)

Benefits of Dynamic Multipoint VPN (DMVPN)

Hub Router Configuration Reduction

- Currently, for each spoke router, there is a separate block of configuration lines on the hub router that define the crypto map characteristics, the crypto access list, and the GRE tunnel interface. This feature allows users to configure a single mGRE tunnel interface, a single IPsec profile, and no crypto access lists on the hub router to handle all spoke routers. Thus, the size of the configuration on the hub router remains constant even if spoke routers are added to the network.
- DMVPN architecture can group many spokes into a single multipoint GRE interface, removing the need for a distinct physical or logical interface for each spoke in a native IPsec installation.

Automatic IPsec Encryption Initiation

- GRE has the peer source and destination address configured or resolved with NHRP. Thus, this feature allows IPsec to be immediately triggered for the point-to-point GRE tunneling or when the GRE peer address is resolved via NHRP for the multipoint GRE tunnel.

Support for Dynamically Addressed Spoke Routers

- When using point-to-point GRE and IPsec hub-and-spoke VPN networks, the physical interface IP address of the spoke routers must be known when configuring the hub router because IP address must be configured as the GRE tunnel destination address. This feature allows spoke routers to have dynamic physical interface IP addresses (common for cable and DSL connections). When the spoke router comes online, it will send registration packets to the hub router: within these registration packets, is the current physical interface IP address of this spoke.

Dynamic Creation for Spoke-to-Spoke Tunnels

- This feature eliminates the need for spoke-to-spoke configuration for direct tunnels. When a spoke router wants to transmit a packet to another spoke router, it can now use NHRP to dynamically determine the required destination address of the target spoke router. (The hub router acts as the NHRP server, handling the request for the source spoke router.) The two spoke routers dynamically create an IPsec tunnel between them so data can be directly transferred.

VRF Integrated DMVPN

- DMVPNs can be used to extend the Multiprotocol Label Switching (MPLS) networks that are deployed by service providers to take advantage of the ease of configuration of hub and spokes, to provide support for dynamically addressed customer premises equipment (CPEs), and to provide zero-touch provisioning for adding new spokes into a DMVPN.

Feature Design of Dynamic Multipoint VPN (DMVPN)

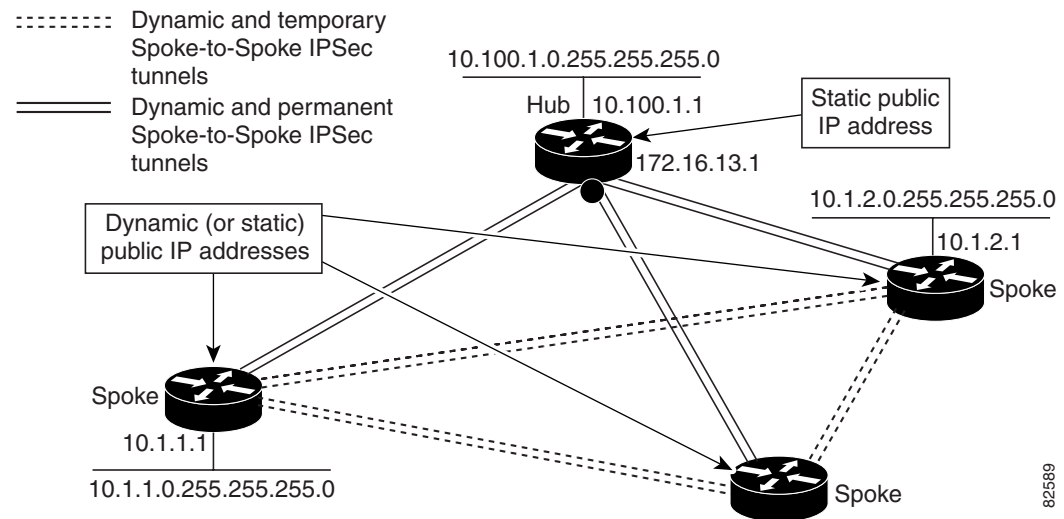
The Dynamic Multipoint VPN (DMVPN) feature combines GRE tunnels, IPsec encryption, and NHRP routing to provide users an ease of configuration via crypto profiles—which override the requirement for defining static crypto maps—and dynamic discovery of tunnel endpoints.

This feature relies on the following two Cisco enhanced standard technologies:

- NHRP—A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of the each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.
- mGRE Tunnel Interface —Allows a single GRE interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.

The topology shown in [Figure 1](#) and the corresponding bullets explain how this feature works.

Figure 1 Sample mGRE and IPsec Integration Topology



- Each spoke has a permanent IPsec tunnel to the hub, not to the other spokes within the network. Each spoke registers as clients of the NHRP server.
- When a spoke needs to send a packet to a destination (private) subnet on another spoke, it queries the NHRP server for the real (outside) address of the destination (target) spoke.
- After the originating spoke “learns” the peer address of the target spoke, it can initiate a dynamic IPsec tunnel to the target spoke.
- The spoke-to-spoke tunnel is built over the multipoint GRE interface.
- The spoke-to-spoke links are established on demand whenever there is traffic between the spokes. Thereafter, packets can bypass the hub and use the spoke-to-spoke tunnel.



Note After a preconfigured amount of inactivity on the spoke-to-spoke tunnels, the router will tear down those tunnels to save resources (IPsec security associations [SAs]).

IPsec Profiles

IPsec profiles abstract IPsec policy information into a single configuration entity, which can be referenced by name from other parts of the configuration. Therefore, users can configure functionality such as GRE tunnel protection with a single line of configuration. By referencing an IPsec profile, the user does not have to configure an entire crypto map configuration. An IPsec profile contains only IPsec information; that is, it does not contain any access list information or peering information.

VRF Integrated DMVPN

VPN Routing and Forwarding (VRF) Integrated DMVPN enables users to map DMVPN multipoint interfaces into MPLS VPNs. This mapping allows Internet service providers (ISPs) to extend their existing MPLS VPN services by mapping off-network sites (typically a branch office) to their respective MPLS VPNs. Customer equipment (CE) routers are terminated on the DMVPN PE router, and traffic is placed in the VRF instance of an MPLS VPN.

DMVPN can interact with MPLS VPNs in two ways:

1. The **ip vrf forwarding** command is used to inject the data IP packets (those packets inside the mGRE+IPsec tunnel) into the MPLS VPN. The **ip vrf forwarding** command is supported for DMVPN in Cisco IOS Release 12.3(6) and Release 12.3(7)T.
2. The **tunnel vrf** command is used to transport (route) the mGRE+IPsec tunnel packet itself within an MPLS VPN. The **tunnel vrf** command is supported in Cisco IOS Release 12.3(11)T but not in Cisco IOS Release 12.2(18)SXE.

**Note**

Clear-text data IP packets are forwarded in a VRF using the **ip vrf forwarding** command, and encrypted tunnel IP packets are forwarded in a VRF using the **tunnel vrf** command.

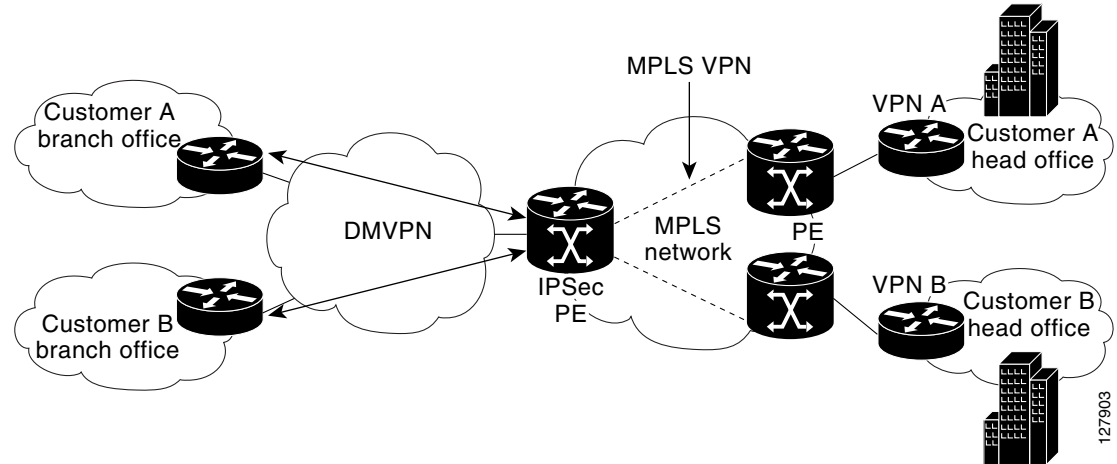
The **ip vrf forwarding** and **tunnel vrf** commands may be used at the same time. If they are used at the same time, the VRF name of each command may be the same or different.

For information about configuring the forwarding of clear-text data IP packets into a VRF, see the section “[Configuring the Forwarding of Clear-Text Data IP Packets into a VRF](#).” For information about configuring the forwarding of encrypted tunnel packets into a VRF, see the section “[Configuring the Forwarding of Encrypted Tunnel Packets into a VRF](#).”

For more information about configuring VRF, see reference in the “[Related Documents](#)” section.

[Figure 2](#) illustrates a typical VRF Integrated DMVPN scenario.

Figure 2 VRF Integrated DMVPN



DMVPN—Enabling Traffic Segmentation Within DMVPN

Cisco IOS Release 12.4(11)T provides an enhancement that allows you to segment VPN traffic within a DMVPN tunnel. VRF instances are labeled, using MPLS, to indicate their source and destination.

The diagram in [Figure 3](#) and the corresponding bullets explain how traffic segmentation within DMVPN works.

NAT-Transparency Aware DMVPN

DMVPN spokes are often situated behind a NAT router (which is often controlled by the ISP for the spoke site) with the outside interface address of the spoke router being dynamically assigned by the ISP using a private IP address (per Internet Engineering Task Force [IETF] RFC 1918).

Prior to Cisco IOS Release 12.3(6) and 12.3(7)T, these spoke routers had to use IPsec tunnel mode to participate in a DMVPN network. In addition, their assigned outside interface private IP address had to be unique across the DMVPN network. Even though ISAKMP and IPsec would negotiate NAT-T and “learn” the correct NAT public address for the private IP address of this spoke, NHRP could only “see” and use the private IP address of the spoke for its mapping entries. Effective with the NAT-Transparency Aware DMVPN enhancement, NHRP can now learn and use the NAT public address for its mappings as long as IPsec transport mode is used (which is the recommended IPsec mode for DMVPN networks). The restriction that the private interface IP address of the spoke must be unique across the DMVPN network has been removed. It is recommended that all DMVPN routers be upgraded to the new code before you try to use the new functionality even though spoke routers that are not behind NAT do not need to be upgraded. In addition, you cannot convert upgraded spoke routers that are behind NAT to the new configuration (IPsec transport mode) until the hub routers have been upgraded.

Also added in Cisco IOS Releases 12.3(9a) and 12.3(11)T is the capability to have the hub DMVPN router behind static NAT. This was a change in the ISAKMP NAT-T support. For this functionality to be used, all the DMVPN spoke routers and hub routers must be upgraded, and IPsec must use transport mode.

For these NAT-Transparency Aware enhancements to work, you must use IPsec transport mode on the transform set. Also, even though NAT-Transparency (IKE and IPsec) can support two peers (IKE and IPsec) being translated to the same IP address (using the UDP ports to differentiate them), this functionality is not supported for DMVPN. All DMVPN spokes must have a unique IP address after they have been NAT translated. They can have the same IP address before they are NAT translated.

Figure 4 illustrates a NAT-Transparency Aware DMVPN scenario.

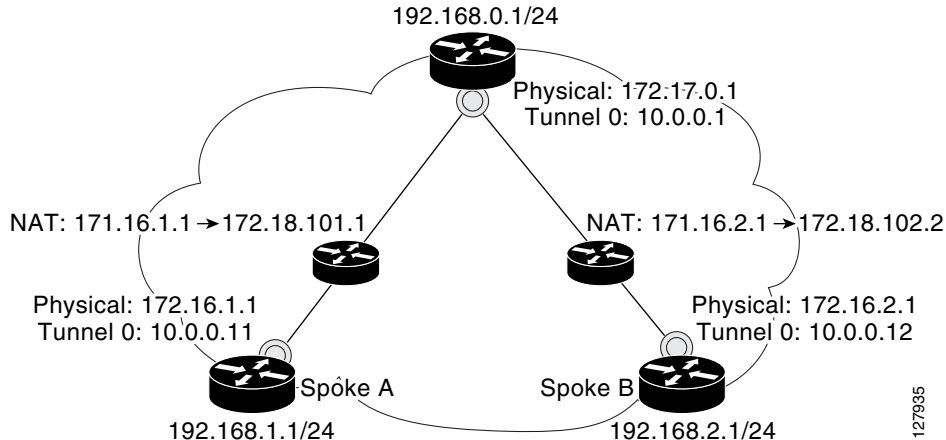


Note

In Cisco IOS Release 12.4(6)T or earlier, DMVPN spokes behind NAT *will not* participate in dynamic direct spoke-to-spoke tunnels. Any traffic to or from a spoke that is behind NAT will be forwarded using the DMVPN hub routers. DMVPN spokes that are not behind NAT in the same DMVPN network may create dynamic direct spoke-to-spoke tunnels between each other.

In Cisco IOS Release 12.4(6)T or later releases, DMVPN spokes behind NAT *will* participate in dynamic direct spoke-to-spoke tunnels. The spokes must be behind NAT boxes that are performing NAT, not PAT. The NAT box must translate the spoke to the same outside NAT IP address for the spoke-spoke connections as the NAT box does for the spoke-hub connection. If there is more than one DMVPN spoke behind the same NAT box, then the NAT box *must* translate the DMVPN spokes to different outside NAT IP addresses. It is also likely that you may not be able to build a direct spoke-spoke tunnel between these spokes. If a spoke-spoke tunnel fails to form, then the spoke-spoke packets will continue to be forwarded via the spoke-hub-spoke path.

Figure 4 NAT-Transparency Aware DMVPN



Call Admission Control with DMVPN

In a DMVPN network, it is easy for a DMVPN router to become “overwhelmed” with the number of tunnels it is trying to build. Call Admission Control can be used to limit the number of tunnels that can be built at any one time, thus protecting the memory of the router and CPU resources.

It is most likely that Call Admission Control will be used on a DMVPN spoke to limit the total number of ISAKMP sessions (DMVPN tunnels) that a spoke router will attempt to initiate or accept. This limiting is accomplished by configuring an IKE SA limit under Call Admission Control, which configures the router to drop new ISAKMP session requests (inbound and outbound) if the current number of ISAKMP SAs exceeds the limit.

It is most likely that Call Admission Control will be used on a DMVPN hub to rate limit the number of DMVPN tunnels that are attempting to be built at the same time. The rate limiting is accomplished by configuring a system resource limit under Call Admission Control, which configures the router to drop new ISAKMP session requests (new DMVPN tunnels) when the system utilization is above a specified percentage. The dropped session requests allow the DMVPN hub router to complete the current ISAKMP session requests, and when the system utilization drops, it can process the previously dropped sessions when they are reattempted.

No special configuration is required to use Call Admission Control with DMVPN. For information about configuring Call Admission Control, see the reference in the section “[Related Documents](#).”

NHRP Rate-Limiting Mechanism

NHRP has a rate-limiting mechanism that restricts the total number of NHRP packets from any given interface. The default values, which are set using the `ip nhrp max-send` command, are 100 packets every 10 seconds per interface. If the limit is exceeded, you will get the following system message:

```
%NHRP-4-QUOTA: Max-send quota of [int]pkts/[int]Sec. exceeded on [chars]
```

For more information about this system message, see the document [12.4T System Message Guide](#).

How to Configure Dynamic Multipoint VPN (DMVPN)

To enable mGRE and IPsec tunneling for hub and spoke routers, you must configure an IPsec profile that uses a global IPsec policy template and configure your mGRE tunnel for IPsec encryption. This section contains the following procedures:

- [Configuring an IPsec Profile, page 11](#) (required)
- [Configuring the Hub for DMVPN, page 13](#) (required)
- [Configuring the Spoke for DMVPN, page 16](#) (required)
- [Configuring the Forwarding of Clear-Text Data IP Packets into a VRF, page 19](#) (optional)
- [Configuring the Forwarding of Encrypted Tunnel Packets into a VRF, page 20](#) (optional)
- [Configuring DMVPN—Traffic Segmentation Within DMVPN, page 20](#)
- [Troubleshooting Dynamic Multipoint VPN \(DMVPN\), page 26](#) (optional)

Configuring an IPsec Profile

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

Prerequisites

Before configuring an IPsec profile, you must define a transform set by using the **crypto ipsec transform-set** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile *name***
4. **set transform-set *transform-set-name***
5. **set identity**
6. **set security association lifetime {seconds *seconds* | kilobytes *kilobytes*}**
7. **set pfs [group1 | group2]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example: Router> enable</p>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <p>Enter your password if prompted.</p>
Step 2	<p>configure terminal</p> <p>Example: Router# configure terminal</p>	<p>Enters global configuration mode.</p>
Step 3	<p>crypto ipsec profile <i>name</i></p> <p>Example: Router(config)# crypto ipsec profile vpnprof</p>	<p>Defines the IPsec parameters that are to be used for IPsec encryption between “spoke and hub” and “spoke and spoke” routers.</p> <p>This command enters crypto map configuration mode.</p> <ul style="list-style-type: none"> The <i>name</i> argument specifies the name of the IPsec profile.
Step 4	<p>set transform-set <i>transform-set-name</i></p> <p>Example: Router(config-crypto-map)# set transform-set trans2</p>	<p>Specifies which transform sets can be used with the IPsec profile.</p> <ul style="list-style-type: none"> The <i>transform-set-name</i> argument specifies the name of the transform set.
Step 5	<p>set identity</p> <p>Example: Router(config-crypto-map)# set identity</p>	<p>(Optional) Specifies identity restrictions to be used with the IPsec profile.</p>
Step 6	<p>set security association lifetime {seconds <i>seconds</i> kilobytes <i>kilobytes</i>}</p> <p>Example: Router(config-crypto-map)# set security association lifetime seconds 1800</p>	<p>(Optional) Overrides the global lifetime value for the IPsec profile.</p> <ul style="list-style-type: none"> The seconds <i>seconds</i> option specifies the number of seconds a security association will live before expiring; the kilobytes <i>kilobytes</i> option specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires. The default for the <i>seconds</i> argument is 3600 seconds.
Step 7	<p>set pfs [<i>group1</i> <i>group2</i>]</p> <p>Example: Router(config-crypto-map)# set pfs group2</p>	<p>(Optional) Specifies that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. If this command is not specified, the default (group1) will be enabled.</p> <ul style="list-style-type: none"> The group1 keyword specifies that IPsec should use the 768-bit Diffie-Hellman (DH) prime modulus group when performing the new DH exchange; the group2 keyword specifies the 1024-bit DH prime modulus group.

What to Do Next

Proceed to the following sections “[Configuring the Hub for DMVPN](#)” and “[Configuring the Spoke for DMVPN](#).”

Configuring the Hub for DMVPN

To configure the hub router for mGRE and IPsec integration (that is, associate the tunnel with the IPsec profile configured in the previous procedure), use the following commands:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask [secondary]*
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map multicast dynamic**
8. **ip nhrp network-id** *number*
9. **tunnel source** {*ip-address* | *type number*}
10. **tunnel key** *key-number*
11. **tunnel mode gre multipoint**
12. **tunnel protection ipsec profile** *name*
13. **bandwidth** *kbps*
14. **ip tcp adjust-mss** *max-segment-size*
15. **ip nhrp holdtime** *seconds*
16. **delay** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>interface tunnel <i>number</i></p> <p>Example: Router(config)# interface tunnel 5</p>	<p>Configures a tunnel interface and enters interface configuration mode</p> <ul style="list-style-type: none"> The <i>number</i> argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	<p>ip address <i>ip-address mask [secondary]</i></p> <p>Example: Router(config-if)# ip address 10.0.0.1 255.255.255.0</p>	<p>Sets a primary or secondary IP address for the tunnel interface.</p> <p>Note All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.</p>
Step 5	<p>ip mtu <i>bytes</i></p> <p>Example: Router(config-if)# ip mtu 1400</p>	<p>Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.</p>
Step 6	<p>ip nhrp authentication <i>string</i></p> <p>Example: Router(config-if)# ip nhrp authentication donttell</p>	<p>Configures the authentication string for an interface using NHRP.</p> <p>Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.</p>
Step 7	<p>ip nhrp map multicast dynamic</p> <p>Example: Router(config-if)# ip nhrp map multicast dynamic</p>	<p>Allows NHRP to automatically add spoke routers to the multicast NHRP mappings.</p>
Step 8	<p>ip nhrp network-id <i>number</i></p> <p>Example: Router(config-if)# ip nhrp network-id 99</p>	<p>Enables NHRP on an interface.</p> <ul style="list-style-type: none"> The <i>number</i> argument specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295.
Step 9	<p>tunnel source {<i>ip-address</i> <i>type number</i>}</p> <p>Example: Router (config-if)# tunnel source Ethernet0</p>	<p>Sets source address for a tunnel interface.</p>
Step 10	<p>tunnel key <i>key-number</i></p> <p>Example: Router (config-if)# tunnel key 100000</p>	<p>(Optional) Enables an ID key for a tunnel interface.</p> <ul style="list-style-type: none"> The <i>key-number</i> argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key. <p>Note The key number must be set to the same value on all hubs and spokes that are in the same DMVPN network.</p> <p>Note This command should not be configured if you are using a Cisco 6500 or Cisco 7600 platform.</p>

	Command or Action	Purpose
Step 11	<pre>tunnel mode gre multipoint</pre> <p>Example: Router(config-if)# tunnel mode gre multipoint</p>	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	<pre>tunnel protection ipsec profile name</pre> <p>Example: Router(config-if)# tunnel protection ipsec profile vpnprof</p>	<p>Associates a tunnel interface with an IPsec profile.</p> <ul style="list-style-type: none"> The <i>name</i> argument specifies the name of the IPsec profile; this value must match the <i>name</i> specified in the crypto ipsec profile name command.
Step 13	<pre>bandwidth kbps</pre> <p>Example: Router(config-if)# bandwidth 1000</p>	<p>Sets the current bandwidth value for an interface to higher-level protocols.</p> <ul style="list-style-type: none"> The <i>kbps</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommend bandwidth value is 1000 or greater. <p>Setting the bandwidth value to at least 1000 is critical if EIGRP is used over the tunnel interface. Higher bandwidth values may be necessary depending on the number of spokes supported by a hub.</p>
Step 14	<pre>ip tcp adjust-mss max-segment-size</pre> <p>Example: Router(config-if)# ip tcp adjust-mss 1360</p>	<p>Adjusts the maximum segment size (MSS) value of TCP packets going through a router.</p> <ul style="list-style-type: none"> The <i>max-segment-size</i> argument specifies the maximum segment size, in bytes. The range is from 500 to 1460. <p>The recommended value is 1360 when the number of IP MTU bytes is set to 1400. With these recommended settings, TCP sessions quickly scale back to 1400-byte IP packets so the packets will “fit” in the tunnel.</p>
Step 15	<pre>ip nhrp holdtime seconds</pre> <p>Example: Router(config-if)# ip nhrp holdtime 450</p>	<p>Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses.</p> <ul style="list-style-type: none"> The <i>seconds</i> argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The recommended value ranges from 300 seconds to 600 seconds.
Step 16	<pre>delay number</pre> <p>Example: Router(config-if)# delay 1000</p>	<p>(Optional) Used to change the EIGRP routing metric for routes learned over the tunnel interface.</p> <ul style="list-style-type: none"> The <i>number</i> argument specifies the delay time in seconds. The recommend value is 1000.

Configuring the Spoke for DMVPN

To configure spoke routers for mGRE and IPsec integration, use the following commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask [secondary]*
5. **ip mtu** *bytes*
6. **ip nhrp authentication** *string*
7. **ip nhrp map** *hub-tunnel-ip-address hub-physical-ip-address*
8. **ip nhrp map multicast** *hub-physical-ip-address*
9. **ip nhrp nhs** *hub-tunnel-ip-address*
10. **ip nhrp network-id** *number*
11. **tunnel source** {*ip-address | type number*}
12. **tunnel key** *key-number*
13. **tunnel mode gre multipoint**
or
tunnel destination *hub-physical-ip-address*
14. **tunnel protection ipsec profile** *name*
15. **bandwidth** *kbps*
16. **ip tcp adjust-mss** *max-segment-size*
17. **ip nhrp holdtime** *seconds*
18. **delay** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>interface tunnel <i>number</i></p> <p>Example: Router(config)# interface tunnel 5</p>	<p>Configures a tunnel interface and enters interface configuration mode.</p> <ul style="list-style-type: none"> The <i>number</i> argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	<p>ip address <i>ip-address mask [secondary]</i></p> <p>Example: Router(config-if)# ip address 10.0.0.2 255.255.255.0</p>	<p>Sets a primary or secondary IP address for the tunnel interface.</p> <p>Note All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.</p>
Step 5	<p>ip mtu <i>bytes</i></p> <p>Example: Router(config-if)# ip mtu 1400</p>	<p>Sets the MTU size, in bytes, of IP packets sent on an interface.</p>
Step 6	<p>ip nhrp authentication <i>string</i></p> <p>Example: Router(config-if)# ip nhrp authentication donttell</p>	<p>Configures the authentication string for an interface using NHRP.</p> <p>Note The NHRP authentication string be set to the same value on all hubs and spokes that are in the same DMVPN network.</p>
Step 7	<p>ip nhrp map <i>hub-tunnel-ip-address hub-physical-ip-address</i></p> <p>Example: Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1</p>	<p>Statically configures the IP-to-NBMA address mapping of IP destinations connected to an MBMA network.</p> <ul style="list-style-type: none"> <i>hub-tunnel-ip-address</i>—Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. <i>hub-physical-ip-address</i>—Defines the static public IP address of the hub.
Step 8	<p>ip nhrp map multicast <i>hub-physical-ip-address</i></p> <p>Example: Router(config-if)# ip nhrp map multicast 172.17.0.1</p>	<p>Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.</p>
Step 9	<p>ip nhrp nhs <i>hub-tunnel-ip-address</i></p> <p>Example: Router(config-if)# ip nhrp nhs 10.0.0.1</p>	<p>Configures the hub router as the NHRP next-hop server.</p>
Step 10	<p>ip nhrp network-id <i>number</i></p> <p>Example: Router(config-if)# ip nhrp network-id 99</p>	<p>Enables NHRP on an interface.</p> <ul style="list-style-type: none"> The <i>number</i> argument specifies a globally unique 32-bit network identifier from a NBMA network. The range is from 1 to 4294967295.
Step 11	<p>tunnel source <i>{ip-address type number}</i></p> <p>Example: Router (config-if)# tunnel source Ethernet0</p>	<p>Sets the source address for a tunnel interface.</p>

	Command or Action	Purpose
Step 12	<p>tunnel key <i>key-number</i></p> <p>Example: Router (config-if)# tunnel key 100000</p>	<p>(Optional) Enables an ID key for a tunnel interface.</p> <ul style="list-style-type: none"> The <i>key-number</i> argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key. The key number must be set to the same value on all hubs and spokes that are in the same DMVPN network. <p>Note This command should not be configured if you are using a Cisco 6500 or Cisco 7600 platform.</p>
Step 13	<p>tunnel mode gre multipoint</p> <p>or</p> <p>tunnel destination <i>hub-physical-ip-address</i></p> <p>Example: Router(config-if)# tunnel mode gre multipoint or Router(config-if)# tunnel destination 172.17.0.1</p>	<p>Sets the encapsulation mode to mGRE for the tunnel interface.</p> <p>Use this command if data traffic can use dynamic spoke-to-spoke traffic.</p> <p>Specifies the destination for a tunnel interface.</p> <p>Use this command if data traffic can use hub-and-spoke tunnels.</p>
Step 14	<p>tunnel protection ipsec profile <i>name</i></p> <p>Example: Router(config-if)# tunnel protection ipsec profile vpnprof</p>	<p>Associates a tunnel interface with an IPsec profile.</p> <ul style="list-style-type: none"> The <i>name</i> argument specifies the name of the IPsec profile; this value must match the <i>name</i> specified in the crypto ipsec profile <i>name</i> command.
Step 15	<p>bandwidth <i>kbps</i></p> <p>Example: Router(config-if)# bandwidth 1000</p>	<p>Sets the current bandwidth value for an interface to higher-level protocols.</p> <ul style="list-style-type: none"> The <i>kbps</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommend bandwidth value is 1000 or greater. <p>The bandwidth setting for the spoke does not need to equal the bandwidth setting for the DMVPN hub. It is usually easier if all of the spokes use the same or similar value.</p>
Step 16	<p>ip tcp adjust-mss <i>max-segment-size</i></p> <p>Example: Router(config-if)# ip tcp adjust-mss 1360</p>	<p>Adjusts the maximum segment size (MSS) value of TCP packets going through a router.</p> <ul style="list-style-type: none"> The <i>max-segment-size</i> argument specifies the maximum segment size, in bytes. The range is from 500 to 1460. <p>The recommended number value is 1360 when the number of IP MTU bytes is set to 1400. With these recommended settings, TCP sessions quickly scale back to 1400-byte IP packets so the packets will “fit” in the tunnel.</p>

	Command or Action	Purpose
Step 17	ip nhrp holdtime <i>seconds</i> Example: Router(config-if)# ip nhrp holdtime 450	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. <ul style="list-style-type: none"> The <i>seconds</i> argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The recommended value ranges from 300 seconds to 600 seconds.
Step 18	delay <i>number</i> Example: Router(config-if)# delay 1000	(Optional) Used to change the EIGRP routing metric for routes learned over the tunnel interface. <ul style="list-style-type: none"> The <i>number</i> argument specifies the delay time in seconds. The recommend value is 1000.

Configuring the Forwarding of Clear-Text Data IP Packets into a VRF

To configure the forwarding of clear-text data IP packets into a VRF, perform the following steps. This configuration assumes that the VRF BLUE has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip vrf forwarding** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router (config)# interface tunnel0	Configures an interface type and enters interface configuration mode.
Step 4	ip vrf forwarding <i>vrf-name</i> Example: Router (config-if)# ip vrf forwarding BLUE	Associates a VPN VRF with an interface or subinterface.

Configuring the Forwarding of Encrypted Tunnel Packets into a VRF

To configure the forwarding of encrypted tunnel packets into a VRF, perform the following steps. This configuration assumes that the VRF RED has already been configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **tunnel vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router (config)# interface tunnel0	Configures an interface type and enters interface configuration mode.
Step 4	tunnel vrf <i>vrf-name</i> Example: Router (config-if)# tunnel vrf RED	Associates a VPN VRF instance with a specific tunnel destination, interface, or subinterface.

Configuring DMVPN—Traffic Segmentation Within DMVPN

There are no new commands to use for configuring traffic segmentation, but there are tasks you must complete in order to segment traffic within a DMVPN tunnel:

- [Enabling MPLS on the VPN Tunnel, page 21](#)
- [Configuring Multiprotocol BGP on the Hub Router, page 21](#)
- [Configuring Multiprotocol BGP on the Spoke Routers, page 24](#)

Prerequisites

The tasks that follow assume that the DMVPN tunnel and the VRFs “red” and “blue” have already been configured.

For information on configuring a DMVPN tunnel, see the “Configuring the Hub for DMVPN” section on page 13 and the “Configuring the Spoke for DMVPN” section on page 16. For details about VRF configuration, see the “Configuring the Forwarding of Clear-Text Data IP Packets into a VRF” section on page 19 and the “Configuring the Forwarding of Encrypted Tunnel Packets into a VRF” section on page 20.

Enabling MPLS on the VPN Tunnel

Because traffic segmentation within a DMVPN tunnel depends upon MPLS, you must configure MPLS for each VRF instance in which traffic will be segmented. For detailed information about configuring MPLS, see *Cisco IOS Multiprotocol Label Switching Configuration Guide*, Release 12.4.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **mpls ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router (config)# interface tunnel0	Configures an interface type and enters interface configuration mode.
Step 4	mpls ip Example: Router (config-if)# mpls ip	Enables MPLS tagging of packets on the specified tunnel interface.

Configuring Multiprotocol BGP on the Hub Router

You must configure multiprotocol iBGP (MP-iBGP) to enable advertisement of VPNv4 prefixes and labels to be applied to the VPN traffic. Use BGP to configure the hub as a route reflector. To force all traffic to be routed via the hub, configure the BGP route reflector to change the next hop to itself when it advertises VPNv4 prefixes to the route reflector clients (spokes).

For more information about the BGP routing protocol, see the “BGP” chapter in the *Cisco IOS IP Routing Protocols Configuration Guide*, Release 12.4.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp**
4. **neighbor *ipaddress* remote-as *as-number***
5. **neighbor *ipaddress* update-source *interface***
6. **address-family vpv4**
7. **neighbor *ipaddress* activate**
8. **neighbor *ipaddress* send-community extended**
9. **neighbor *ipaddress* route-reflector-client**
10. **neighbor *ipaddress* route-map *nexthop* out**
11. **exit-address family**
12. **address-family ipv4 *vrf-name***
13. **redistribute connected**
14. **route-map**
15. **set ip next-hop *ipaddress***

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp Example: Router (config)# router bgp	Enters BGP configuration mode.
Step 4	neighbor <i>ipaddress</i> remote-as <i>as-number</i> Example: Router (config)# neighbor 10.0.0.11 remote-as 1	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
Step 5	neighbor ipaddress update-source interface Example: Router (config)# neighbor 10.10.10.11 update-source Tunnel1	Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	address-family vpnv4 Example: Router (config)# address-family vpnv4	Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.
Step 7	neighbor ipaddress activate Example: Router (config)# neighbor 10.0.0.11 activate	Enables the exchange of information with a BGP neighbor.
Step 8	neighbor ipaddress send-community extended Example: Router (config)# neighbor 10.0.0.11 send-community extended	Specifies that extended community attributes should be sent to a BGP neighbor.
Step 9	neighbor ipaddress route-reflector-client Example: Router (config)# neighbor 10.0.0.11 route-reflector-client	Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 10	neighbor ipaddress route-map nexthop out Example: Router (config)# neighbor 10.0.0.11 route-map nexthop out	Forces all traffic to be routed via the hub.
Step 11	exit-address-family Example: Router (config)# exit-address-family	Exits the address family configuration mode for VPNv4.
Step 12	address-family ipv4 vrf-name Example: Router (config)# address-family ipv4 vrf red	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 13	redistribute connected Example: Router (config)# redistribute connected	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.

	Command or Action	Purpose
Step 14	<code>route-map</code> Example: Router (config)# route-map nexthop permit 10	Enters route map configuration mode to configure the next-hop that will be advertised to the spokes.
Step 15	<code>set ip next-hop ipaddress</code> Example: Router (config)# set ip next-hop 10.0.0.1	Sets the next hop to be the hub.

Configuring Multiprotocol BGP on the Spoke Routers

Multiprotocol-iBGP (MP-iBGP) must be configured on the spoke routers and the hub. Follow the steps below for each spoke router in the DMVPN.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `router bgp`
4. `neighbor ipaddress remote-as as-number`
5. `neighbor ipaddress update-source interface`
6. `address-family vpnv4`
7. `neighbor ipaddress activate`
8. `neighbor ipaddress send-community extended`
9. `exit-address-family`
10. `address-family ipv4 vrf-name`
11. `redistribute connected`
12. `exit-address-family`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp Example: Router (config)# router bgp 1	Enters BGP configuration mode.
Step 4	neighbor ipaddress remote-as as-number Example: Router (config)# neighbor 10.0.0.1 remote-as 1	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	neighbor ipaddress update-source interface Example: Router (config)# neighbor 10.10.10.1 update-source Tunnel1	Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	address-family vpnv4 Example: Router (config)# address-family vpnv4	Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.
Step 7	neighbor ipaddress activate Example: Router (config)# neighbor 10.0.0.1 activate	Enables the exchange of information with a BGP neighbor.
Step 8	neighbor ipaddress send-community extended Example: Router (config)# neighbor 10.0.0.1 send-community extended	Specifies that extended community attributes should be sent to a BGP neighbor.
Step 9	exit-address-family Example: Router (config)# exit-address-family	Exits the address family configuration mode.
Step 10	address-family ipv4 vrf-name Example: Router (config)# address-family ipv4 vrf red	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 11	redistribute connected Example: Router (config)# redistribute connected	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 12	exit-address-family Example: Router (config)# exit-address-family	Exits the address family configuration mode. Note Repeat Steps 10–12 for each VRF.

Troubleshooting Dynamic Multipoint VPN (DMVPN)

After configuring DMVPN, to verify that DMVPN is operating correctly, to clear DMVPN statistics or sessions, or to debug DMVPN, you may perform the following optional steps:

SUMMARY STEPS

1. **clear dmvpn session** [peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf vrf-name] [static]
2. **clear dmvpn statistics** [peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf vrf-name]
3. **debug dmvpn** {{{condition [unmatched] | [peer [nbma | tunnel {ip-address}]] | [vrf {vrf-name}] | [interface {tunnel number}]]} | [{error | detail | packet | all} {nhrp | crypto | tunnel | socket | all}]}
4. **debug nhrp condition**
5. **debug nhrp error**
6. **logging dmvpn** [rate-limit seconds]
7. **show crypto ipsec sa** [active | standby]
8. **show crypto isakmp sa**
9. **show crypto map**
10. **show dmvpn** [peer [nbma | tunnel {ip-address}] | [network {ip-address} {mask}]] [vrf {vrf-name}] [interface {tunnel number}] [detail] [static] [debug-condition]
11. **show ip nhrp traffic** [interface {tunnel number}]

DETAILED STEPS

-
- Step 1** The **clear dmvpn session** command is used to clear DMVPN sessions.
- The following example clears only dynamic DMVPN sessions:
- ```
Router# clear dmvpn session peer nbma
```
- The following example clears all DMVPN sessions, both static and dynamic, for the specified tunnel:
- ```
Router# clear dmvpn session interface tunnel 100 static
```
- Step 2** The **clear dmvpn statistics** command is used to clear DMVPN related counters. The following example shows how to clear DMVPN related session counters for the specified tunnel interface:
- ```
Router# clear dmvpn statistics peer tunnel 192.0.2.3
```
- Step 3** The **debug dmvpn** command is used to debug DMVPN sessions. You can enable or disable DMVPN debugging based on a specific condition. There are three levels of DMVPN debugging, listed in the order of details from lowest to highest:
- Error level
  - Detail level
  - Packet level
- The following example shows how to enable conditional DMVPN debugging that displays all error debugs for next hop routing protocol (NHRP), sockets, tunnel protection and crypto information:
- ```
Router# debug dmvpn error all
```

- Step 4** The **debug nhrp condition** command enables or disables debugging based on a specific condition. The following example shows how to enable conditional NHRP debugging:

```
Router# debug nhrp condition
```

- Step 5** The **debug nhrp error** command displays information about NHRP error activity. The following example shows how to enable debugging for NHRP error messages:

```
Router# debug nhrp error
```

- Step 6** The **logging dmvpn** command is used to enable DMVPN system logging. The following command shows how to enable DMVPN system logging at the rate of 1 message every 20 seconds:

```
Router(config)# logging dmvpn rate-limit 20
```

The following example shows a sample system log with DMVPN messages:

```
%DMVPN-7-CRYPTO_SS: Tunnel101-192.0.2.1 socket is UP
%DMVPN-5-NHRP_NHS: Tunnel101 192.0.2.251 is UP
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel1 Registered.
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel101 came UP.
%DMVPN-3-NHRP_ERROR: Registration Request failed for 192.0.2.251 on Tunnel101
```

- Step 7** The **show crypto ipsec sa** command displays the settings used by the current SAs. The following example output shows the IPsec SA status of only the active device:

```
Router# show crypto ipsec sa active

interface: Ethernet0/0
  Crypto map tag: to-peer-outside, local addr 209.165.201.3
  protected vrf: (none)
  local ident (addr/mask/prot/port): (192.168.0.1/255.255.255.255/0/0)
  remote ident (addr/mask/prot/port): (172.16.0.1/255.255.255.255/0/0)
  current_peer 209.165.200.225 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 3
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
    local crypto endpt.: 209.165.201.3, remote crypto endpt.: 209.165.200.225
    path mtu 1500, media mtu 1500
    current outbound spi: 0xD42904F0(3559458032)
    inbound esp sas:
      spi: 0xD3E9ABD0(3555306448)
        transform: esp-3des ,
        in use settings ={Tunnel, }
        conn id: 2006, flow_id: 6, crypto map: to-peer-outside
        sa timing: remaining key lifetime (k/sec): (4586265/3542)
        HA last key lifetime sent(k): (4586267)
        ike_cookies: 9263635C CA4B4E99 C14E908E 8EE2D79C
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE
```

- Step 8** The **show crypto isakmp sa** command displays all current IKE SAs at a peer. For example, the following sample output is displayed after IKE negotiations have successfully completed between two peers.

```
Router# show crypto isakmp sa
```

dst	src	state	conn-id	slot
172.17.63.19	172.16.175.76	QM_IDLE	2	0
172.17.63.19	172.17.63.20	QM_IDLE	1	0
172.16.175.75	172.17.63.19	QM_IDLE	3	0

- Step 9** The **show crypto map** command displays the crypto map configuration.

The following sample output is displayed after a crypto map has been configured:

```
Router# show crypto map

Crypto Map "Tunnel5-head-0" 10 ipsec-isakmp
  Profile name: vpnprof
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 20 ipsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.16.175.75
  Extended IP access list
    access-list permit gre host 172.17.63.19 host 172.16.175.75
  Current peer: 172.16.175.75
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 30 ipsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.17.63.20
  Extended IP access list
    access-list permit gre host 172.17.63.19 host 172.17.63.20
  Current peer: 172.17.63.20
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={trans2, }

Crypto Map "Tunnel5-head-0" 40 ipsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.16.175.76
  Extended IP access list
    access-list permit gre host 172.17.63.19 host 172.16.175.76
  Current peer: 172.16.175.76
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={trans2, }
  Interfaces using crypto map Tunnel5-head-0:
  Tunnel5
```

Step 10 The **show dmvpn** command displays DMVPN specific session information. The following example shows example summary output:

```
Router# show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
         N - NATed, L - Local, X - No Socket
         # Ent --> Number of NHRP entries with same NBMA peer

! The line below indicates that the sessions are being displayed for Tunnel1.
! Tunnel1 is acting as a spoke and is a peer with three other NBMA peers.

Tunnel1, Type: Spoke, NBMA Peers: 3,
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
  2   192.0.2.21   192.0.2.116   IKE      3w0d D
  1   192.0.2.102   192.0.2.11   NHRP    02:40:51 S
  1   192.0.2.225   192.0.2.10   UP       3w0d S

Tunnel2, Type: Spoke, NBMA Peers: 1,
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
  1   192.0.2.25   192.0.2.171   IKE      never S
```

Step 11 The **show ip nhrp traffic** command displays NHRP statistics. The following example shows output for a specific tunnel, tunnel7:

```
Router# show ip nhrp traffic interface tunnel7
Tunnel7: Max-send limit:100Pkts/10Sec, Usage:0%
  Sent: Total 79
        18 Resolution Request   10 Resolution Reply   42 Registration Request
         0 Registration Reply     3 Purge Request      6 Purge Reply
         0 Error Indication      0 Traffic Indication
  Rcvd: Total 69
        10 Resolution Request   15 Resolution Reply   0 Registration Request
        36 Registration Reply    6 Purge Request      2 Purge Reply
         0 Error Indication      0 Traffic Indication
```

What to Do Next

If you have troubleshooted your DMVPN configuration and proceed to contact technical support, the **show tech-support** command includes information for DMVPN sessions. For more information, see the **show tech-support** command in the Cisco IOS Configuration Fundamentals Command Reference.

Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature

This section provides the following comprehensive configuration examples:

- [Hub Configuration for DMVPN: Example, page 29](#)
- [Spoke Configuration for DMVPN: Example, page 30](#)
- [VRF Aware DMVPN: Example, page 31](#)

Hub Configuration for DMVPN: Example

In the following example, which configures the hub router for multipoint GRE and IPsec integration, no explicit configuration lines are needed for each spoke; that is, the hub is configured with a global IPsec policy template that all spoke routers can talk to. In this example, EIGRP is configured to run over the private physical interface and the tunnel interface.

```
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
mode transport
!
crypto ipsec profile vpnprof
  set transform-set trans2
!
interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
! Ensures longer packets are fragmented before they are encrypted; otherwise, the
receiving router would have to do the reassembly.
```

```

ip mtu 1400
! The following line must match on all nodes that "want to use" this mGRE tunnel:
ip nhrp authentication donttell
! Note that the next line is required only on the hub.
ip nhrp map multicast dynamic
! The following line must match on all nodes that want to use this mGRE tunnel:
ip nhrp network-id 99
ip nhrp holdtime 300
! Turns off split horizon on the mGRE tunnel interface; otherwise, EIGRP will not
advertise routes that are learned via the mGRE interface back out that interface.
no ip split-horizon eigrp 1
! Enables dynamic, direct spoke-to-spoke tunnels when using EIGRP.
no ip next-hop-self eigrp 1
ip tcp adjust-mss 1360
delay 1000
! Sets IPsec peer address to Ethernet interface's public address.
tunnel source Ethernet0
tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel.
tunnel key 100000
tunnel protection ipsec profile vpnprof
!
interface Ethernet0
ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
network 10.0.0.0 0.0.0.255
network 192.168.0.0 0.0.0.255
!

```

For information about defining and configuring ISAKMP profiles, see the references in the “[Related Documents](#)” section.

Spoke Configuration for DMVPN: Example

In the following example, all spokes are configured the same except for tunnel and local interface address, thereby, reducing necessary configurations for the user:

```

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1400
! The following line must match on all nodes that want to use this mGRE tunnel:
ip nhrp authentication donttell
! Definition of NHRP server at the hub (10.0.0.1), which is permanently mapped to the
static public address of the hub (172.17.0.1).
ip nhrp map 10.0.0.1 172.17.0.1

```

```

! Sends multicast packets to the hub router, and enables the use of a dynamic routing
protocol between the spoke and the hub.
 ip nhrp map multicast 172.17.0.1
! The following line must match on all nodes that want to use this mGRE tunnel:
 ip nhrp network-id 99
 ip nhrp holdtime 300
! Configures the hub router as the NHRP next-hop server.
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
! The following line must match on all nodes that want to use this mGRE tunnel:
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
! This is a spoke, so the public address might be dynamically assigned via DHCP.
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
!
! EIGRP is configured to run over the inside physical interface and the tunnel.
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.1.0 0.0.0.255

```

VRF Aware DMVPN: Example

When configuring VRF Aware DMVPN, you must create a separate DMVPN network for each VRF instance. In the following example, there are two DMVPN networks: BLUE and RED. In addition, a separate source interface has been used on the hub for each DMVPN tunnel—a must for Cisco IOS Release 12.2(18)SXE. For other Cisco IOS releases, you can configure the same tunnel source for both of the tunnel interfaces, but you must configure the **tunnel key** and **tunnel protection (tunnel protection ipsec profile {name} shared)** commands.



Note

If you use the **shared** keyword, then you should be running Cisco IOS Release 12.4(5) or Release 12.4(6)T, or a later release. Otherwise the IPsec/GRE tunnels under the two mGRE tunnel interfaces may not function correctly.

Hub Configuration

```

interface Tunnel0
! Note the next line.
 ip vrf forwarding BLUE
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
 ip mtu 1436
! Note the next line.
 ip nhrp authentication BLUE!KEY
 ip nhrp map multicast dynamic
! Note the next line
 ip nhrp network-id 100000
 ip nhrp holdtime 600
 no ip split-horizon eigrp 1
 no ip next-hop-self eigrp 1
 ip tcp adjust-mss 1360

```

```

delay 1000
! Note the next line.
tunnel source Ethernet0
tunnel mode gre multipoint
tunnel protection ipsec profile vpnprof!
interface Tunnel1
! Note the next line.
ip vrf forwarding RED
bandwidth 1000
ip address 10.0.0.1 255.255.255.0
ip mtu 1436
! Note the next line.
ip nhrp authentication RED!KEY
ip nhrp map multicast dynamic
! Note the next line.
ip nhrp network-id 20000
ip nhrp holdtime 600
no ip split-horizon eigrp 1
no ip next-hop-self eigrp 1
ip tcp adjust-mss 1360
delay 1000
! Note the next line.
tunnel source Ethernet1
tunnel mode gre multipoint
tunnel protection ipsec profile vpnprof!
interface Ethernet0
ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
ip address 192.0.2.171 255.255.255.0

```

**Note**

For the hub configuration shown above, a separate DMVPN network is configured for each VPN. The NHRP network ID and authentication keys must be unique on the two mGRE interfaces.

EIGRP Configuration on the Hub

```

router eigrp 1
auto-summary
!
address-family ipv4 vrf BLUE
network 10.0.0.0 0.0.0.255
no auto-summary
autonomous-system 1
exit-address-family
!
address-family ipv4 vrf RED
network 10.0.0.0 0.0.0.255
no auto-summary
autonomous-system 1
exit-address-family

```

Spoke Configurations**Spoke 1:**

```

interface Tunnel0
bandwidth 1000
ip address 10.0.0.2 255.255.255.0
ip mtu 1436
! Note the next line.
ip nhrp authentication BLUE!KEY
ip nhrp map 10.0.0.1 172.17.0.1

```

```

ip nhrp network-id 100000
ip nhrp holdtime 300
ip nhrp nhs 10.0.0.1
ip tcp adjust-mss 1360
delay 1000
tunnel mode gre multipoint
tunnel source Ethernet0
tunnel destination 172.17.0.1
tunnel protection ipsec profile vpnprof

```

Spoke 2:

```

interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1436
 ip nhrp authentication RED!KEY
 ip nhrp map 10.0.0.1 192.0.2.171
 ip nhrp network-id 200000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0
 tunnel destination 192.0.2.171
 tunnel protection ipsec profile vpnprof!

```

2547oDMVPN with Traffic Segmentation (with BGP only): Example

The following example show a traffic segmentation configuration in which traffic is segmented between two spokes that serve as provider edge (PE) devices.

Hub Configuration

```

hostname hub-pe1

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

```

```

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
mode transport

crypto ipsec profile prof
  set transform-set t1

interface Tunnell
  ip address 10.9.9.1 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco
  ip nhrp map multicast dynamic
  ip nhrp network-id 1

!The command below enables MPLS on the DMVPN network:
mpls ip
  tunnel source Ethernet0/0
  tunnel mode gre multipoint
  tunnel protection ipsec profile prof

interface Loopback0
  ip address 10.0.0.1 255.255.255.255

interface Ethernet0/0
  ip address 172.0.0.1 255.255.255.0

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.0.0.11 remote-as 1
  neighbor 10.0.0.11 update-source Tunnell
  neighbor 10.0.0.12 remote-as 1
  neighbor 10.0.0.12 update-source Tunnell
  no auto-summary

  address-family vpnv4
    neighbor 10.0.0.11 activate
    neighbor 10.0.0.11 send-community extended
    neighbor 10.0.0.11 route-reflector-client
    neighbor 10.0.0.11 route-map NEXTHOP out
    neighbor 10.0.0.12 activate
    neighbor 10.0.0.12 send-community extended
    neighbor 10.0.0.12 route-reflector-client
    neighbor 10.0.0.12 route-map NEXTHOP out
  exit-address-family

  address-family ipv4 vrf red
    redistribute connected
    no synchronization
  exit-address-family

  address-family ipv4 vrf blue
    redistribute connected
    no synchronization
  exit-address-family

no ip http server
no ip http secure-server

```

```
!In this route map information, the hub sets the next hop to itself, and the VPN prefixes
are advertised:
route-map NEXTHOP permit 10
  set ip next-hop 10.0.0.1

control-plane

line con 0
  logging synchronous
line aux 0
line vty 0 4
  no login

end
```

Spoke Configurations

Spoke 2

```
hostname spoke-pe2

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
  rd 2:2
  route-target export 2:2
  route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
  rd 1:1
  route-target export 1:1
  route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
  mode transport

crypto ipsec profile prof
  set transform-set t1

interface Tunnel1
  ip address 10.0.0.11 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco
  ip nhrp map multicast dynamic
```

```
ip nhrp map 10.0.0.1 172.0.0.1
ip nhrp map multicast 172.0.0.1
ip nhrp network-id 1
ip nhrp nhs 10.0.0.1
```

!The command below enables MPLS on the DMVPN network:

```
mpls ip
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof
```

```
interface Loopback0
ip address 10.9.9.11 255.255.255.255
```

```
interface Ethernet0/0
ip address 172.0.0.11 255.255.255.0
!
```

```
!  
interface Ethernet1/0  
  ip vrf forwarding red  
  ip address 192.168.11.2 255.255.255.0  
  
interface Ethernet2/0  
  ip vrf forwarding blue  
  ip address 192.168.11.2 255.255.255.0  
  
!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop  
information to set itself as the next-hop and assigns a new VPN label for the prefixes  
learned from the spokes and advertises the VPN prefix:  
router bgp 1  
  no synchronization  
  bgp log-neighbor-changes  
  neighbor 10.0.0.1 remote-as 1  
  neighbor 10.0.0.1 update-source Tunnell  
  no auto-summary  
  
  address-family vpnv4  
  neighbor 10.0.0.1 activate  
  neighbor 10.0.0.1 send-community extended  
  exit-address-family  
  
!  
  address-family ipv4 vrf red  
  redistribute connected  
  no synchronization  
  exit-address-family  
  
!  
  address-family ipv4 vrf blue  
  redistribute connected  
  no synchronization  
  exit-address-family  
  
no ip http server  
no ip http secure-server  
  
control-plane  
  
line con 0  
  logging synchronous  
line aux 0  
line vty 0 4  
  no login  
  
end
```

Spoke 3

```
hostname spoke-PE3  
  
boot-start-marker  
boot-end-marker  
  
no aaa new-model  
  
resource policy  
  
clock timezone EST 0  
ip cef  
no ip domain lookup
```

```

!This section refers to the forwarding table for VRF blue:
ip vrf blue
  rd 2:2
  route-target export 2:2
  route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
  rd 1:1
  route-target export 1:1
  route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
  mode transport

crypto ipsec profile prof
  set transform-set t1

interface Tunnell
  ip address 10.0.0.12 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco
  ip nhrp map multicast dynamic
  ip nhrp map 10.0.0.1 172.0.0.1
  ip nhrp map multicast 172.0.0.1
  ip nhrp network-id 1
  ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
mpls ip
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof
!
interface Loopback0
  ip address 10.9.9.12 255.255.255.255

interface Ethernet0/0
  ip address 172.0.0.12 255.255.255.0

interface Ethernet1/0
  ip vrf forwarding red
  ip address 192.168.12.2 255.255.255.0

interface Ethernet2/0
  ip vrf forwarding blue
  ip address 192.168.12.2 255.255.255.0

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.0.0.1 remote-as 1
  neighbor 10.0.0.1 update-source Tunnell
  no auto-summary

```

```
address-family vpnv4
neighbor 10.0.0.1 activate
neighbor 10.0.0.1 send-community extended
exit-address-family

address-family ipv4 vrf red
redistribute connected
no synchronization
exit-address-family

address-family ipv4 vrf blue
redistribute connected
no synchronization
exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

2547oDMVPN with Traffic Segmentation (Enterprise Branch): Example

The following example shows a configuration for segmenting traffic between two spokes located at branch offices of an enterprise. In this example, EIGRP is configured to learn routes to reach BGP neighbors within the DMVPN.

Hub Configuration

```
hostname HUB

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2
```

```

!This refers to the forwarding table for VRF red:
ip vrf red
  rd 1:1
  route-target export 1:1
  route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
  mode transport

crypto ipsec profile prof
  set transform-set t1

interface Tunnel1
  ip address 10.0.0.1 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco
  ip nhrp map multicast dynamic
  ip nhrp network-id 1

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
no ip split-horizon eigrp 1

!The command below enables MPLS on the DMVPN network:
mpls ip
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
  ip address 10.9.9.1 255.255.255.255

interface Ethernet0/0
  ip address 172.0.0.1 255.255.255.0

!EIGRP is configured to learn the BGP peer addresses (10.9.9.x networks)
router eigrp 1
  network 10.9.9.1 0.0.0.0
  network 10.0.0.0 0.0.0.255
  no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp router-id 10.9.9.1
  bgp log-neighbor-changes
  neighbor 10.9.9.11 remote-as 1
  neighbor 10.9.9.11 update-source Loopback0
  neighbor 10.9.9.12 remote-as 1
  neighbor 10.9.9.12 update-source Loopback0
  no auto-summary

address-family vpnv4
  neighbor 10.9.9.11 activate
  neighbor 10.9.9.11 send-community extended
  neighbor 10.9.9.11 route-reflector-client

```

```
neighbor 10.9.9.12 activate
neighbor 10.9.9.12 send-community extended
neighbor 10.9.9.12 route-reflector-client
exit-address-family

address-family ipv4 vrf red
redistribute connected
no synchronization
exit-address-family

address-family ipv4 vrf blue
redistribute connected
no synchronization
exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
  logging synchronous
line aux 0
line vty 0 4
  no login

end
```

Spoke Configurations

Spoke 2

```
hostname Spoke2

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
  rd 2:2
  route-target export 2:2
  route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
  rd 1:1
  route-target export 1:1
  route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
```

```

crypto ipsec transform-set t1 esp-des
  mode transport

crypto ipsec profile prof
  set transform-set t1

interface Tunnell
  ip address 10.0.0.11 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco
  ip nhrp map multicast dynamic
  ip nhrp map 10.0.0.1 172.0.0.1
  ip nhrp map multicast 172.0.0.1
  ip nhrp network-id 1
  ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
mpls ip
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
  ip address 10.9.9.11 255.255.255.255

interface Ethernet0/0
  ip address 172.0.0.11 255.255.255.0

interface Ethernet1/0
  ip vrf forwarding red
  ip address 192.168.11.2 255.255.255.0

interface Ethernet2/0
  ip vrf forwarding blue
  ip address 192.168.11.2 255.255.255.0

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
router eigrp 1
  network 10.9.9.11 0.0.0.0
  network 10.0.0.0 0.0.0.255
  no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp router-id 10.9.9.11
  bgp log-neighbor-changes
  neighbor 10.9.9.1 remote-as 1
  neighbor 10.9.9.1 update-source Loopback0
  no auto-summary

address-family vpnv4
  neighbor 10.9.9.1 activate
  neighbor 10.9.9.1 send-community extended
  exit-address-family

address-family ipv4 vrf red
  redistribute connected
  no synchronization
  exit-address-family

```

```
address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit-address-family

no ip http server
no ip http secure-server

control-plane

line con 0
 logging synchronous
line aux 0
line vty 0 4
 no login

end
```

Spoke 3

```
hostname Spoke3

boot-start-marker
boot-end-marker

no aaa new-model

resource policy

clock timezone EST 0
ip cef
no ip domain lookup

!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2

!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
 route-target export 1:1
 route-target import 1:1

mpls label protocol ldp

crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0

crypto ipsec transform-set t1 esp-des
 mode transport

crypto ipsec profile prof
 set transform-set t1

interface Tunnel1
 ip address 10.0.0.12 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
```

```

ip nhrp map 10.0.0.1 172.0.0.1
ip nhrp map multicast 172.0.0.1
ip nhrp network-id 1
ip nhrp nhs 10.0.0.1

!The command below enables MPLS on the DMVPN network:
mpls ip
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof

!This address is advertised by EIGRP and used as the BGP endpoint:
interface Loopback0
ip address 10.9.9.12 255.255.255.255

interface Ethernet0/0
ip address 172.0.0.12 255.255.255.0

interface Ethernet1/0
ip vrf forwarding red
ip address 192.168.12.2 255.255.255.0

interface Ethernet2/0
ip vrf forwarding blue
ip address 192.168.12.2 255.255.255.0

!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
router eigrp 1
network 10.9.9.12 0.0.0.0
network 10.0.0.0 0.0.0.255
no auto-summary

!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
no synchronization
bgp router-id 10.9.9.12
bgp log-neighbor-changes
neighbor 10.9.9.1 remote-as 1
neighbor 10.9.9.1 update-source Loopback0
no auto-summary

address-family vpnv4
neighbor 10.9.9.1 activate
neighbor 10.9.9.1 send-community extended
exit-address-family

address-family ipv4 vrf red
redistribute connected
no synchronization
exit-address-family

address-family ipv4 vrf blue
redistribute connected
no synchronization
exit-address-family

no ip http server
no ip http secure-server

control-plane

```

```

line con 0
  logging synchronous
line aux 0
line vty 0 4
  no login

end

```

Sample Command Output: show mpls ldp bindings

```

Spoke2# show mpls ldp bindings

tib entry: 10.9.9.1/32, rev 8
  local binding: tag: 16
  remote binding: tsr: 10.9.9.1:0, tag: imp-null
tib entry: 10.9.9.11/32, rev 4
  local binding: tag: imp-null
  remote binding: tsr: 10.9.9.1:0, tag: 16
tib entry: 10.9.9.12/32, rev 10
  local binding: tag: 17
  remote binding: tsr: 10.9.9.1:0, tag: 17
tib entry: 10.0.0.0/24, rev 6
  local binding: tag: imp-null
  remote binding: tsr: 10.9.9.1:0, tag: imp-null
tib entry: 172.0.0.0/24, rev 3
  local binding: tag: imp-null
  remote binding: tsr: 10.9.9.1:0, tag: imp-null
Spoke2#

```

Sample Command Output: show mpls forwarding-table

```

Spoke2# show mpls forwarding-table

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id   switched  interface
16     Pop tag    10.9.9.1/32    0         Tu1       10.0.0.1
17     17        10.9.9.12/32  0         Tu1       10.0.0.1
18     Aggregate 192.168.11.0/24[V] \
0
19     Aggregate 192.168.11.0/24[V] \
0
Spoke2#

```

Sample Command Output: show ip route vrf red

```

Spoke2# show ip route vrf red

Routing Table: red
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    192.168.12.0/24 [200/0] via 10.9.9.12, 00:00:02
C    192.168.11.0/24 is directly connected, Ethernet1/0
Spoke2#

```

Sample Command Output: show ip route vrf blue

```
Spoke2# show ip route vrf blue

Routing Table: blue
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    192.168.12.0/24 [200/0] via 10.9.9.12, 00:00:08
C    192.168.11.0/24 is directly connected, Ethernet2/0
Spoke2#
Spoke2# show ip cef vrf red 192.168.12.0
```

```
192.168.12.0/24, version 5, epoch 0
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Tu1, 10.0.0.1, tags imposed: {17 18}
  via 10.9.9.12, 0 dependencies, recursive
  next hop 10.0.0.1, Tunnel1 via 10.9.9.12/32
  valid adjacency
  tag rewrite with Tu1, 10.0.0.1, tags imposed: {17 18}
Spoke2#
```

Sample Command Output: show ip bgp neighbors

```
Spoke2# show ip bgp neighbors

BGP neighbor is 10.9.9.1, remote AS 1, internal link
  BGP version 4, remote router ID 10.9.9.1
  BGP state = Established, up for 00:02:09
  Last read 00:00:08, last write 00:00:08, hold time is 180, keepalive interval is 60
seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

              Sent          Rcvd
  Opens:                1            1
  Notifications:        0            0
  Updates:               4            4
  Keepalives:           4            4
  Route Refresh:        0            0
  Total:                 9            9

Default minimum time between advertisement runs is 0 seconds

For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
```

```

                Sent      Rcvd
Prefix activity:  ----      ----
  Prefixes Current:      0      0
  Prefixes Total:        0      0
  Implicit Withdraw:     0      0
  Explicit Withdraw:     0      0
  Used as bestpath:     n/a     0
  Used as multipath:     n/a     0

                Outbound  Inbound
Local Policy Denied Prefixes:  -----  -----
  Total:                      0      0
Number of NLRI in the update sent: max 0, min 0

For address family: VPNv4 Unicast
  BGP table version 9, neighbor version 9/0
Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member

                Sent      Rcvd
Prefix activity:  ----      ----
  Prefixes Current:      2      2 (Consumes 136 bytes)
  Prefixes Total:        4      2
  Implicit Withdraw:     2      0
  Explicit Withdraw:     0      0
  Used as bestpath:     n/a     2
  Used as multipath:     n/a     0

                Outbound  Inbound
Local Policy Denied Prefixes:  -----  -----
  ORIGINATOR loop:          n/a     2
  Bestpath from this peer:    4      n/a
  Total:                     4      2
Number of NLRI in the update sent: max 1, min 1

Connections established 1; dropped 0
Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 10.9.9.11, Local port: 179
Foreign host: 10.9.9.1, Foreign port: 12365

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x2D0F0):
Timer          Starts    Wakeups      Next
Retrans         6         0           0x0
TimeWait        0         0           0x0
AckHold         7         3           0x0
SendWnd         0         0           0x0
KeepAlive       0         0           0x0
GiveUp          0         0           0x0
PmtuAger        0         0           0x0
DeadWait        0         0           0x0

iss: 3328307266  snduna: 3328307756  sndnxt: 3328307756   sndwnd: 15895
irs: 4023050141  rcvnxt: 4023050687  rcvwnd: 16384     delrcvwnd: 0

SRTT: 165 ms, RTTO: 1457 ms, RTV: 1292 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6

```

```
Datagrams (max data segment is 536 bytes):
Rcvd: 13 (out of order: 0), with data: 7, total data bytes: 545
Sent: 11 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with
data: 6, total data bytes: 489
Spoke2#
```

Additional References

The following sections provide references related to Dynamic Multipoint VPN (DMVPN):

Related Documents

Related Topic	Document Title
Call Admission Control	Call Admission Control for IKE , Cisco IOS Release 12.4
GRE tunnel keepalive information	Generic Routing Encapsulation (GRE) Tunnel Keepalive , Cisco IOS Release 12.2(8)T
IKE configuration tasks such as defining an IKE policy	The chapter “ Configuring Internet Key Exchange for IPSec VPNs ” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.4
IPsec configuration tasks	The chapter “ Configuring Security for VPNs with IPSec ” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.4
Tunnel interface configuration tasks	The section “ Implementing Tunnels ” in the chapter “Interface Configuration Overview” in the <i>Cisco IOS Interface and Hardware Component Configuration Guide</i> , Release 12.4
Configuring VRF-Aware IPsec	VRF-Aware IPsec , in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.4
Configuring MPLS	Cisco IOS Multiprotocol Label Switching Configuration Guide , Release 12.4.
Configuring BGP	The chapter “ BGP ” in the <i>Cisco IOS IP Routing Protocols Configuration Guide</i> , Release 12.4
System messages	12.4T System Message Guide
Defining and configuring ISAKMP profiles	“ Certificate to ISAKMP Profile Mapping ” chapter in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.4

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2547	BGP/MPLS VPNs

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password</p>	http://www.cisco.com/techsupport

Command Reference

This section documents new and modified commands only.

Modified Command

- [show ip nhrp traffic](#)

New Commands

- [clear dmvpn session](#)
- [clear dmvpn statistics](#)
- [debug dmvpn](#)
- [debug nhrp condition](#)
- [debug nhrp error](#)
- [logging dmvpn](#)
- [show dmvpn](#)

clear dmvpn session

To clear Dynamic Multipoint VPN (DMVPN) sessions, use the **clear dmvpn session** command in privileged EXEC mode.

```
clear dmvpn session [peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf
vrf-name] [static]
```

Syntax Description	
peer	(Optional) Specifies a DMVPN peer.
nbma	(Optional) Specifies nonbroadcast multiaccess (NBMA) addresses.
tunnel	(Optional) Specifies a tunnel.
<i>ip-address</i>	(Optional) Specifies the IP address for the DMVPN peer.
interface	(Optional) Displays DMVPN information based on a specific interface.
<i>tunnel number</i>	(Optional) Specifies tunnel address for DMVPN peer.
vrf <i>vrf-name</i>	(Optional) Clears all Next Hop Resolution Protocol (NHRP) sessions related to the specified virtual routing forwarding (VRF) configuration.
static	(Optional) Clears all static and dynamic NHRP entries.
	Note If the static keyword is not specified, only dynamic NHRP entries are cleared.

Command Default The DMVPN sessions will not be cleared.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.4(9)T	This command was introduced.

Usage Guidelines This command clears existing DMVPN sessions based on input parameters.

Examples The following example clears all DMVPN sessions, both static and dynamic, for the specified peer NBMA address:

```
Router# clear dmvpn session peer nbma 192.0.2.1 static
```

Related Commands	Command	Description
	clear ip nhrp	Clears all dynamic entries from the NHRP cache.

clear dmvpn statistics

To clear Dynamic Multipoint VPN (DMVPN) related counters, use the **clear dmvpn statistics** command in privileged EXEC mode.

```
clear dmvpn statistics [peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf
vrf-name]
```

Syntax	Description
peer	(Optional) Specifies a DMVPN peer.
nbma	(Optional) Specifies nonbroadcast multiaccess (NBMA) addresses.
tunnel	(Optional) Specifies a tunnel.
<i>ip-address</i>	(Optional) Specifies the IP address for the DMVPN peer.
interface	(Optional) Displays DMVPN information based on a specific interface.
tunnel number	(Optional) Specifies tunnel address for DMVPN peer.
vrf <i>vrf-name</i>	(Optional) Clears all DMVPN counters related to the specified virtual routing forwarding (VRF) configuration.

Command Default DMVPN counters will not be cleared.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.4(9)T	This command was introduced.

Usage Guidelines Based on input parameters, DMVPN related session counters will be cleared.

Examples The following example shows how to clear DMVPN related session counters for the specified tunnel interface:

```
Router# clear dmvpn statistics peer tunnel 192.0.2.3
```

Related Commands	Command	Description
	clear dmvpn session	Clears DMVPN sessions.

debug dmvpn

To debug Dynamic Multipoint VPN (DMVPN) sessions, use the **debug dmvpn** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug dmvpn {{{ condition [unmatched] | [peer [nbma | tunnel {ip-address}] ] | [vrf {vrf-name}]
| [interface {tunnel number}] ] } | { error | detail | packet | all } { nhrp | crypto | tunnel | socket
| all } }
```

```
no debug dmvpn {{{ condition [unmatched] | [peer [nbma | tunnel {ip-address}] ] | [vrf
{vrf-name}] | [interface {tunnel number}] ] } | { error | detail | packet | all } { nhrp | crypto |
tunnel | socket | all } }
```

Syntax Description

condition	(Optional) Specifies conditional debugging based on error level setting.
unmatched	(Optional) Specifies debugging when context information is not available.
peer	(Optional) Specifies information for a specific DMVPN peer.
nbma	(Optional) Displays DMVPN information based on nonbroadcast multiaccess (NBMA) addresses.
tunnel	(Optional) Displays DMVPN information based on the peer virtual private network (VPN) address.
<i>ip-address</i>	(Optional) Specifies DMVPN peer IP address.
vrf <i>vrf-name</i>	(Optional) Displays information based on the specified virtual routing forwarding (VRF) name.
interface	(Optional) Displays DMVPN information based on a specific interface.
tunnel <i>number</i>	(Optional) Specifies the tunnel address for a DMVPN peer.
error	(Optional) Enables error level debugging.
detail	(Optional) Enables detail level debugging.
packet	(Optional) Enables packet level debugging.
all	(Optional) Enables all levels of debugging.
nhrp	(Optional) Enables Next Hop Resolution Protocol (NHRP) debugging only.
crypto	(Optional) Enables crypto Internet Key Exchange (IKE) and IPsec debugging only.
tunnel	(Optional) Enables tunnel protection debugging only.
socket	(Optional) Enables crypto secure socket debugging only.
all	(Optional) Enables NHRP, sockets, tunnel protection and crypto debugging.

Command Default

DMVPN debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

Conditional debugging will be in effect once the debug level has been specified. Either error level, detail level, packet level, or all debug levels may be turned on. Once conditional debugging is enabled, the **condition** keyword must be specified if you wish to disable conditional debugging.

There are three levels of debugging. From the least detailed information to the most detailed information, the DMVPN debugging levels are: error level, detail level, and packet level.

Error Level Debugging

When error level debugging is enabled with the **debug dmvpn error** command, the following debugging commands are enabled by default:

- **debug crypto ipsec error**
- **debug crypto isakmp error**
- **debug nhrp error**

Detail Level Debugging

When event level debugging is enabled with the **debug dmvpn detail** command, the following debugging commands are enabled by default:

- **debug crypto ipsec**
- **debug crypto isakmp**
- **debug crypto sockets**
- **debug nhrp**
- **debug nhrp cache**
- **debug nhrp rate**
- **debug tunnel protection**

Packet Level Debugging

When event level debugging is enabled with the **debug dmvpn packet** command, the following debugging commands are enabled by default:

- **debug nhrp extension**
- **debug nhrp packet**

**Note**

Executing **debug dmvpn all all** with a high number of active sessions, may result in high CPU utilization and large data output.

Examples

The following example shows how to enable conditional DMVPN debugging for a specific peer NBMA address:

```
Router# debug dmvpn condition peer nbma 192.0.2.1
```

The following example shows how to enable conditional DMVPN debugging when context is not available to check against debugging conditions:

```
Router# debug dmvpn condition unmatched
```

The following example shows how to disable conditional debugging for a specific tunnel interface, tunnel1:

```
Router# no debug dmvpn condition interface tunnel1
```

The following example shows how to disable all conditional debugging:

```
Router# no debug dmvpn condition
```

Related Commands

Command	Description
debug nhrp condition	Enables NHRP conditional debugging.
debug nhrp error	Displays NHRP error level debugging information.

debug nhrp condition

To enable Next Hop Resolution Protocol (NHRP) conditional debugging, use the **debug nhrp condition** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug nhrp condition {[peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf
vrf-name]}
```

```
no debug nhrp condition {[peer {nbma | tunnel} ip-address] [interface {tunnel number}] [vrf
vrf-name]}
```

Syntax Description

peer	(Optional) Specifies an NHRP peer.
nbma	(Optional) Specifies nonbroadcast multiaccess (NBMA) addresses.
tunnel	(Optional) Specifies a tunnel.
<i>ip-address</i>	(Optional) Specifies the IP address for the NHRP peer.
interface	(Optional) Displays NHRP information based on a specific interface.
tunnel number	(Optional) Specifies tunnel address for NHRP peer.
vrf vrf-name	(Optional) Specifies debugging information for sessions related to the specified virtual routing forwarding (VRF) configuration.

Command Default

Conditional NHRP debugging is not enabled.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(9)T	This command was introduced.

Examples

The following example shows how to enable conditional NHRP debugging for a specific tunnel:

```
Router# debug nhrp condition peer tunnel 192.0.2.1
```

Related Commands

Command	Description
debug dmvpn	Displays DMVPN session debugging information.
debug nhrp error	Displays NHRP error level debugging information.

debug nhrp error

To display Next Hop Resolution Protocol (NHRP) error level debugging information, use the **debug nhrp error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug nhrp error

no debug nhrp error

Syntax Description This command has no arguments or keywords.

Command Default Error level NHRP debugging is not enabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.4(9)T	This command was introduced.

Examples The following example shows how to enable error level debugging for NHRP:

```
Router# debug nhrp error
NHRP errors debugging is on
```

Related Commands	Command	Description
	debug dmvpn	Displays DMVPN session debugging information.
	debug nhrp condition	Enables NHRP conditional debugging.

logging dmvpn

To display Dynamic Multipoint VPN (DMVPN) specific system logging information, use the **logging dmvpn** command in global configuration mode. To turn off logging, use the **no** form of this command.

logging dmvpn [*rate-limit seconds*]

no logging dmvpn [*rate-limit seconds*]

Syntax Description	rate-limit <i>seconds</i> (Optional) Specifies the message rate limit in seconds. The default rate limit is one message every 60 seconds.
---------------------------	--

Command Default	DMVPN system logging messages are not enabled.
------------------------	--

Command Modes	Global configuration
----------------------	----------------------

Command History	Release	Modification
	12.4(9)T	This command was introduced.

Usage Guidelines	Use this command to specify a rate-limit time for DMVPN system logging messages.
-------------------------	--

Examples	The following example shows a sample system log with DMVPN messages:
-----------------	--

```
%DMVPN-7-CRYPTO_SS: Tunnel101-192.0.2.1 socket is UP
%DMVPN-5-NHRP_NHS: Tunnel101 192.0.2.251 is UP
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel1 Registered.
%DMVPN-5-NHRP_CACHE: Client 192.0.2.2 on Tunnel101 came UP.
%DMVPN-3-NHRP_ERROR: Registration Request failed for 192.0.2.251 on Tunnel101
```

Related Commands	Command	Description
	debug dmvpn	Debugs DMVPN sessions.

show dmvpn

To display Dynamic Multipoint VPN (DMVPN) specific session information, use the **show dmvpn** command in privileged EXEC mode.

```
show dmvpn [peer [nbma | tunnel {ip-address}] | [network {ip-address} {mask}]] [vrf
{vrf-name}] [interface {tunnel number}] [detail] [static] [debug-condition]
```

Syntax Description		
peer	(Optional)	Displays information for a specific DMVPN peer.
nbma	(Optional)	Displays DMVPN information based on nonbroadcast multiaccess (NBMA) addresses.
tunnel	(Optional)	Displays DMVPN information based on the peer virtual private network (VPN) address.
<i>ip-address</i>	(Optional)	Specifies DMVPN peer IP address.
network <i>ip-address</i> <i>mask</i>	(Optional)	Displays DMVPN information based on a specific destination network and mask address.
vrf <i>vrf-name</i>	(Optional)	Displays information based on the specified virtual routing forwarding (VRF).
interface	(Optional)	Displays DMVPN information based on a specific interface.
tunnel <i>number</i>	(Optional)	Specifies tunnel address for DMVPN peer.
detail	(Optional)	Displays detail DMVPN information for each session, including Next Hop Server (NHS) and NHS status, crypto session information, and socket details.
static	(Optional)	Displays only static DMVPN information.
debug-condition	(Optional)	Displays DMVPN conditional debugging.

Command Default This command is not enabled.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.4(9)T	This command was introduced.

Usage Guidelines Use this command to obtain DMVPN specific session information. By default, summary information will be displayed.

When the **detail** keyword is used, command output will include information from the **show crypto session detail** command, including inbound and outbound security parameter indexes (SPI) and the **show crypto socket** command.

Examples

The following example shows sample summary output:

```
Router# show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
         N - NATed, L - Local, X - No Socket
         # Ent --> Number of NHRP entries with same NBMA peer

! The line below indicates that the sessions are being displayed for Tunnel1.
! Tunnel1 is acting as a spoke and is a peer with three other NBMA peers.

Tunnel1, Type: Spoke, NBMA Peers: 3,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
  2    192.0.2.21      192.0.2.116  IKE      3w0d D
  1    192.0.2.102      192.0.2.11  NHRP 02:40:51 S
  1    192.0.2.225      192.0.2.10  UP       3w0d S

Tunnel2, Type: Spoke, NBMA Peers: 1,
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
-----
  1    192.0.2.25      192.0.2.171  IKE      never S
```

Table 1 describes the significant fields shown in the display.

Table 1 show dmvpn Field Descriptions

Field	Description
# Ent	The number of Next Hop Routing Protocol (NHRP) entries in the current session.
Peer NBMA Addr	The remote NBMA address.
Peer Tunnel Add	The remote tunnel endpoint IP address.
State	The state of the DMVPN session. The DMVPN session is either up or down. If the DMVPN state is down, the reason for the down state error is displayed—Internet Key Exchange (IKE), IPsec, or NHRP.
UpDn Tm	Displays how long the session has been in the current state.
Attrb	Displays any associated attributes of the current session. One of the following attributes will be displayed—dynamic (D), static (S), incomplete (I), Network Address Translation (NAT) for the peer address, or NATed, (N), local (L), no socket (X).

The following example shows example detail output:

```
Router# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
         N - NATed, L - Local, X - No Socket
         # Ent --> Number of NHRP entries with same NBMA peer
----- Interface Tunnel1 info: -----
Intf. is up, Line Protocol is up, Addr. is 192.0.2.5
  Source addr: 192.0.2.229, Dest addr: MGRE
  Protocol/Transport: "multi-GRE/IP", Protect "gre_prof",
  Tunnel VRF "" ip vrf forwarding ""

NHRP Details: NHS: 192.0.2.10 RE 192.0.2.11 E
```

```

Type: Spoke, NBMA Peers: 4
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
      2      192.0.2.21      192.0.2.116      UP 00:14:59 D      192.0.2.118/24
                                     UP 00:14:59 D      192.0.2.116/32

IKE SA: local 192.0.2.229/500 remote 192.0.2.21/500 Active
      Capabilities:(none) connid:1031 lifetime:23:45:00
Crypto Session Status: UP-ACTIVE
fvrf: (none)
IPSEC FLOW: permit 47 host 192.0.2.229 host 192.0.2.21
      Active SAs: 2, origin: crypto map
      Inbound: #pkts dec'ed 1 drop 0 life (KB/Sec) 4494994/2700
      Outbound: #pkts enc'ed 1 drop 0 life (KB/Sec) 4494994/2700
      Outbound SPI : 0xD1EA3C9B, transform : esp-3des esp-sha-hmac
      Socket State: Open

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
      1      192.0.2.229      192.0.2.5      UP 00:15:00 DLX      192.0.2.5/32

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
      1      192.0.2.102      192.0.2.11 NHRP 02:55:47 S      192.0.2.11/32

IKE SA: local 192.0.2.229/4500 remote 192.0.2.102/4500 Active
      Capabilities:N connid:1028 lifetime:11:45:37
Crypto Session Status: UP-ACTIVE
fvrf: (none)
IPSEC FLOW: permit 47 host 192.0.2.229 host 192.0.2.102
      Active SAs: 2, origin: crypto map
      Inbound: #pkts dec'ed 199056 drop 393401 life (KB/Sec) 4560270/1524
      Outbound: #pkts enc'ed 416631 drop 10531 life (KB/Sec) 4560322/1524
      Outbound SPI : 0x9451AF5C, transform : esp-3des esp-sha-hmac
      Socket State: Open

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
      1      192.0.2.225      192.0.2.10      UP      3w0d S      192.0.2.10/32

IKE SA: local 192.0.2.229/500 remote 192.0.2.225/500 Active
      Capabilities:(none) connid:1030 lifetime:03:46:44
Crypto Session Status: UP-ACTIVE
fvrf: (none)
IPSEC FLOW: permit 47 host 192.0.2.229 host 192.0.2.225
      Active SAs: 2, origin: crypto map
      Inbound: #pkts dec'ed 430261 drop 0 life (KB/Sec) 4415197/3466
      Outbound: #pkts enc'ed 406232 drop 4 life (KB/Sec) 4415197/3466
      Outbound SPI : 0xAF3E15F2, transform : esp-3des esp-sha-hmac
      Socket State: Open

----- Interface Tunnel2 info: -----
Intf. is up, Line Protocol is up, Addr. is 192.0.2.172
      Source addr: 192.0.2.20, Dest addr: MGRE
      Protocol/Transport: "multi-GRE/IP", Protect "gre_prof",
      Tunnel VRF "" ip vrf forwarding ""

NHRP Details: NHS:      192.0.2.171 E

Type: Spoke, NBMA Peers: 1
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
      1      192.0.2.25      192.0.2.171 IKE      never S      192.0.2.171/32

```

■ show dmvpn

```

IKE SA: local 192.0.2.20/500 remote 192.0.2.25/500 Inactive
      Capabilities:(none) connid:0 lifetime:0
IKE SA: local 192.0.2.20/500 remote 192.0.2.25/500 Inactive
      Capabilities:(none) connid:0 lifetime:0
Crypto Session Status: DOWN-NEGOTIATING
fvrf: (none)
IPSEC FLOW: permit 47 host 192.0.2.20 host 192.0.2.25
      Active SAs: 0, origin: crypto map
      Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
      Outbound: #pkts enc'ed 0 drop 436431 life (KB/Sec) 0/0
      Outbound SPI : 0x      0, transform :
      Socket State: Closed

```

```

Pending DMVPN Sessions:
!There are no pending DMVPN sessions.

```

The following example shows example configured conditions displays for DMVPN debugging:

```

Router# show dmvpn debug-condition
NBMA addresses under debug are:

```

```

Interfaces under debug are:
Tunnel101,
Crypto DMVPN filters:
Interface = Tunnel101

```

```

DMVPN Conditional debug context unmatched flag: OFF

```

Related Commands

Command	Description
debug dmvpn	Debugs DMVPN sessions.

show ip nhrp traffic

To display Next Hop Resolution Protocol (NHRP) traffic statistics, use the **show ip nhrp traffic** command in privileged EXEC mode.

```
show ip nhrp traffic [interface {tunnel number}]
```

Syntax

Field	Description
interface	(Optional) Displays NHRP traffic information for a given interface.
tunnel number	(Optional) Specifies the tunnel interface number.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.3	This command was introduced.
12.4(9)T	The interface and tunnel keywords and the <i>number</i> argument were added.

Examples

The following example shows output for a specific tunnel, tunnel0:

```
Router# show ip nhrp traffic interface tunnel0
Tunnel0: Max-send limit:100Pkts/10Sec, Usage:0%
  Sent: Total 79
        18 Resolution Request  10 Resolution Reply  42 Registration Request
         0 Registration Reply  3 Purge Request   6 Purge Reply
         0 Error Indication  0 Traffic Indication
  Rcvd: Total 69
        10 Resolution Request  15 Resolution Reply  0 Registration Request
        36 Registration Reply  6 Purge Request   2 Purge Reply
         0 Error Indication  0 Traffic Indication
```

[Table 2](#) describes the significant fields shown in the display.

Table 2 *show ip nhrp traffic Field Descriptions*

Field	Description
Tunnel0	Interface type and number.
Max-Send limit	Maximum number of NHRP messages that can be sent by this station in the given interval.
Resolution Request	Number of NHRP resolution request packets originated from or received by this station.
Resolution Reply	Number of NHRP resolution reply packets originated from or received by this station.
Registration Request	Number of NHRP registration request packets originated from or received by this station.
Registration Reply	Number of NHRP registration reply packets originated from or received by this station.

Table 2 *show ip nhrp traffic Field Descriptions (continued)*

Field	Description
Purge Request	Number of NHRP purge request packets originated from or received by this station.
Purge Reply	Number of NHRP purge reply packets originated from or received by this station.
Error Indication	Number of NHRP error packets originated from or received by this station.
Traffic Indication	Number of NHRP traffic indication packets (redirects) originated from or received by this station.

Related Commands

Command	Description
debug nhrp condition	Enables NHRP conditional debugging.
debug nhrp error	Enables NHRP error level debugging.

Feature Information for Dynamic Multipoint VPN (DMVPN)

Table 3 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Dynamic Multipoint VPN (DMVPN)

Feature Name	Releases	Feature Information
2547oDMVPN—Enabling Traffic Segmentation Within DMVPN	12.4(11)T	The 2547oDMVPN feature allows users to segment VPN traffic within a DMVPN tunnel by applying MPLS labels to VRF instances to indicate the source and destination of each VRF.
Mangeability Enhancements for DMVPN	12.4(9)T	DMVPN session manageabilty was expanded with DMVPN specific commands for debugging, show output, session and counter control, and system log information. The following sections provide information about this feature: <ul style="list-style-type: none"> • Troubleshooting Dynamic Multipoint VPN (DMVPN) The following commands were introduced or modified by this feature: clear dmvpn session , clear dmvpn statistics , debug dmvpn , debug nhrp condition , debug nhrp error , logging dmvpn , show dmvpn , show ip nhrp traffic
DMVPN Phase 2	12.2(18)SXE 12.3(9)a 12.3(8)T1	DMVPN Spoke-to-Spoke functionality was made more production ready. If you are using this functionality in a production network, the minimum release is Release 12.3(9a) or Release 12.3(8)T1. In Release 12.2(18)SXE, support was added for the Cisco Catalyst 6500 series switch and the Cisco 7600 series router.

Table 3 Feature Information for Dynamic Multipoint VPN (DMVPN)

Feature Name	Releases	Feature Information
—	12.3(6) 12.3(7)T	Virtual Route Forwarding Integrated DMVPN and Network Address Translation-Transparency (NAT-T) Aware DMVPN enhancements were added. In addition, DMVPN Hub-to-Spoke functionality was made more production ready. If you are using this functionality in a production network, the minimum release requirement is Cisco IOS Release 12.3(6) or 12.3(7)T. The enhancements added in Cisco IOS Release 12.3(6) were integrated into Cisco IOS Release 12.3(7)T.
Dynamic Multipoint VPN (DMVPN) Phase 1	12.2(13)T	The Dynamic Multipoint VPN (DMVPN) feature allows users to better scale large and small IPsec Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IP security (IPsec) encryption, and Next Hop Resolution Protocol (NHRP).

Glossary

AM—aggressive mode. A mode during IKE negotiation. Compared to MM, AM eliminates several steps, making it faster but less secure than MM. Cisco IOS software will respond in aggressive mode to an IKE peer that initiates aggressive mode.

GRE—generic routing encapsulation. Tunnels that provide a specific pathway across the shared WAN and encapsulate traffic with new packet headers to ensure delivery to specific destinations. The network is private because traffic can enter a tunnel only at an endpoint. Tunnels do not provide true confidentiality (encryption does) but can carry encrypted traffic.

GRE tunneling can also be used to encapsulate non-IP traffic into IP and send it over the Internet or IP network. The Internet Package Exchange (IPX) and AppleTalk protocols are examples of non-IP traffic.

IKE—Internet Key Exchange. A hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations.

IPsec—IP security. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (“peers”), such as Cisco routers.

ISAKMP—Internet Security Association Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

MM—main mode. Mode that is slower than aggressive mode but more secure and more flexible than aggressive mode because it can offer an IKE peer more security proposals. The default action for IKE authentication (rsa-sig, rsa-encr, or preshared) is to initiate main mode.

NHRP—Next Hop Resolution Protocol. Routers, access servers, and hosts can use NHRP to discover the addresses of other routers and hosts connected to a NBMA network.

The Cisco implementation of NHRP supports the IETF draft version 11 of *NBMA Next Hop Resolution Protocol (NHRP)*.

The Cisco implementation of NHRP supports IP Version 4, Internet Packet Exchange (IPX) network layers, and, at the link layer, ATM, Ethernet, SMDS, and multipoint tunnel networks. Although NHRP is available on Ethernet, NHRP need not be implemented over Ethernet media because Ethernet is capable of broadcasting. Ethernet support is unnecessary (and not provided) for IPX.

PFS—Perfect Forward Secrecy. A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

SA—security association. Describes how two or more entities will utilize security services to communicate securely. For example, an IPsec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPsec connection.

Both IPsec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPsec SA is established either by IKE or by manual user configuration.

transform—The list of operations done on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

VPN—Virtual Private Network. A framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.

**Note**

See [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2002–2007 Cisco Systems, Inc. All rights reserved.

