



Dynamic Multipoint VPN Configuration Guide, Cisco IOS Release 12.4T

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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

© 2011 Cisco Systems, Inc. All rights reserved.



CONTENTS

Dynamic Multipoint VPN 1

Finding Feature Information 1

Prerequisites for Dynamic Multipoint VPN (DMVPN) 1

Restrictions for Dynamic Multipoint VPN (DMVPN) 2

DMVPN Support on the Cisco 6500 and Cisco 7600 2

Information About Dynamic Multipoint VPN (DMVPN) 3

Benefits of Dynamic Multipoint VPN (DMVPN) 4

Feature Design of Dynamic Multipoint VPN (DMVPN) 4

IPsec Profiles 5

VRF Integrated DMVPN 6

DMVPN--Enabling Traffic Segmentation Within DMVPN 7

NAT-Transparency Aware DMVPN 8

Call Admission Control with DMVPN 9

NHRP Rate-Limiting Mechanism 10

How to Configure Dynamic Multipoint VPN (DMVPN) 10

Configuring an IPsec Profile 10

What to Do Next 12

Configuring the Hub for DMVPN 12

Configuring the Spoke for DMVPN 16

Configuring the Forwarding of Clear-Text Data IP Packets into a VRF 20

Configuring the Forwarding of Encrypted Tunnel Packets into a VRF 21

Configuring DMVPN--Traffic Segmentation Within DMVPN 22

Prerequisites 22

Enabling MPLS on the VPN Tunnel 22

Configuring Multiprotocol BGP on the Hub Router 23

Configuring Multiprotocol BGP on the Spoke Routers 26

Troubleshooting Dynamic Multipoint VPN (DMVPN) 28

What to Do Next 32

Configuration Examples for Dynamic Multipoint VPN (DMVPN) Feature 32

Example Hub Configuration for DMVPN	32
Example Spoke Configuration for DMVPN	33
Example VRF Aware DMVPN	34
Example 2547oDMVPN with Traffic Segmentation (with BGP only)	36
Example 2547oDMVPN with Traffic Segmentation (Enterprise Branch)	40
Additional References	46
Feature Information for Dynamic Multipoint VPN (DMVPN)	47
Glossary	49
Per-Tunnel QoS for DMVPN	51
Finding Feature Information	51
Prerequisites for Per-Tunnel QoS for DMVPN	51
Restrictions for Per-Tunnel QoS for DMVPN	51
Information About Per-Tunnel QoS for DMVPN	52
Benefits of Per-Tunnel QoS for DMVPN	52
NHRP QoS Provisioning for DMVPN	52
How to Configure Per-Tunnel QoS for DMVPN	53
Configuring an NHRP Group on a Spoke	53
Mapping an NHRP Group to a QoS Policy on the Hub	54
Verifying Per-Tunnel QoS for DMVPN	55
Configuration Examples for Per-Tunnel QoS for DMVPN	57
Configuring an NHRP Group on a Spoke Example	57
Mapping an NHRP Group to a QoS Policy on the Hub Example	58
Verifying Per-Tunnel QoS for DMVPN Examples	59
Additional References	62
Feature Information for Per-Tunnel QoS for DMVPN	63
NHRP MIB	65
Finding Feature Information	65
Prerequisites for NHRP MIB	65
Restrictions for NHRP MIB	65
Information About NHRP MIB	66
CISCO-NHRP-MIB	66
RFC-2677	66
How to Use NHRP MIB	66
Verifying NHRP MIB Status	66
Configuration Examples for NHRP MIB	67

Example Verifying NHRP MIB Status	67
Example VRF-Aware NHRP MIB Configuration	67
Additional References	69
Feature Information for NHRP MIB	70
DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	73
Finding Feature Information	73
Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	73
Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	74
DMVPN Spoke-to-spoke Tunneling Limited to Spokes not Behind a NAT Device	74
NHRP Registration	75
NHRP Resolution	76
NHRP Spoke-to-Spoke Tunnel with a NAT Device	76
NHRP Registration Process	77
NHRP Resolution and Purge Process	77
Additional References	78
Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	79
Sharing IPsec with Tunnel Protection	81
Finding Feature Information	81
Restrictions for Sharing IPsec with Tunnel Protection	81
Information About Sharing IPsec with Tunnel Protection	82
Using a Single IPsec SA to Secure GRE Tunnel Sessions	82
How to Share an IPsec Session Between Multiple Tunnels	83
Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN	83
What to Do Next	84
Configuration Examples for Sharing IPsec with Tunnel Protection	84
Dual-hub Router and Dual-DMVPN Topology	85
Hub 1 Configuration Example	85
Hub 2 Configuration Example	86
Spoke 1 Configuration Example	87
Spoke 2 Configuration Example	88
Results on Spoke 1	89
Additional References	93
Feature Information for Sharing IPsec with Tunnel Protection	95
Glossary	95



Dynamic Multipoint VPN

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, page 1](#)
- [Prerequisites for Dynamic Multipoint VPN \(DMVPN\), page 1](#)
- [Restrictions for Dynamic Multipoint VPN \(DMVPN\), page 2](#)
- [Information About Dynamic Multipoint VPN \(DMVPN\), page 3](#)
- [How to Configure Dynamic Multipoint VPN \(DMVPN\), page 10](#)
- [Configuration Examples for Dynamic Multipoint VPN \(DMVPN\) Feature, page 32](#)
- [Additional References, page 46](#)
- [Feature Information for Dynamic Multipoint VPN \(DMVPN\), page 47](#)
- [Glossary, page 49](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

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.
- If one spoke is behind one NAT device and another different spoke is behind another NAT device, and Peer Address Translation (PAT) is the type of NAT used on both NAT devices, then a session initiated between the two spokes cannot be established.

One example of a PAT configuration on a NAT interface is:

```
ip nat inside source list nat_acl interface FastEthernet0/1 overload
```

- [DMVPN Support on the Cisco 6500 and Cisco 7600, page 2](#)

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-recursive** 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.
- Enhanced Interior Gateway Routing Protocol (EIGRP) should be avoided.

Information About Dynamic Multipoint VPN (DMVPN)

- [Benefits of Dynamic Multipoint VPN \(DMVPN\), page 4](#)
- [Feature Design of Dynamic Multipoint VPN \(DMVPN\), page 4](#)
- [IPsec Profiles, page 5](#)
- [VRF Integrated DMVPN, page 6](#)
- [DMVPN--Enabling Traffic Segmentation Within DMVPN, page 7](#)
- [NAT-Transparency Aware DMVPN, page 8](#)

- [Call Admission Control with DMVPN](#), page 9
- [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)

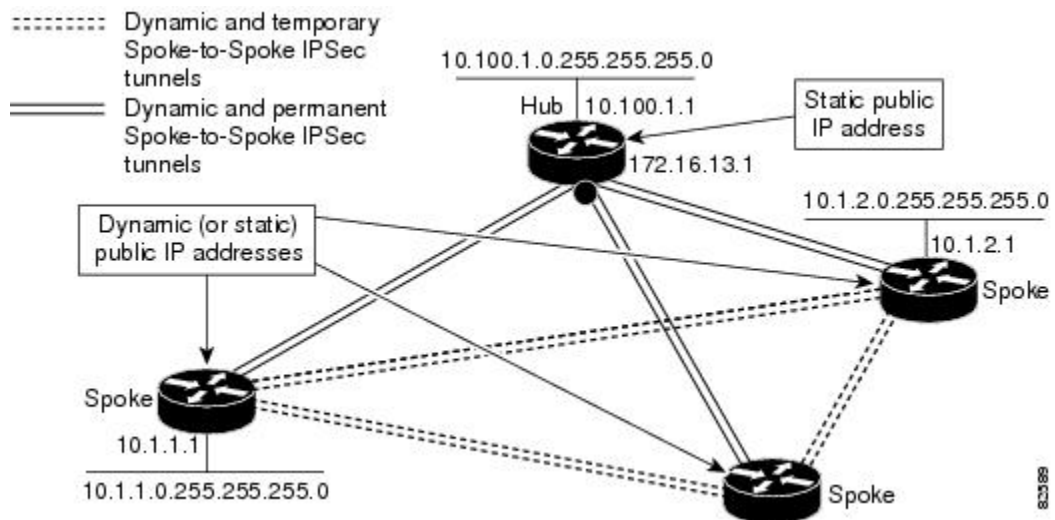
The Dynamic Multipoint VPN (DMVPN) feature combines GRE tunnels, IPsec encryption, and NHRP routing to provide users an ease of configuration via crypto profiles--which override the requirement for defining static crypto maps--and dynamic discovery of tunnel endpoints.

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 the diagram below 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.

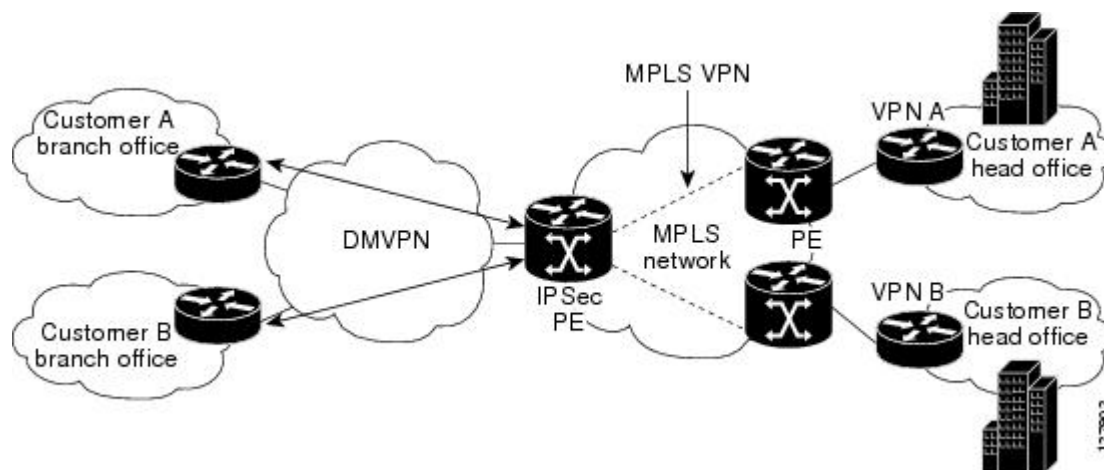
The **ip vrf forwarding** and **tunnel vrf** commands may be used at the same time. If they are used at the same time, the VRF name of each command may be the same or different.

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.

The diagram below illustrates a typical VRF Integrated DMVPN scenario.

Figure 2 VRF Integrated DMVPN

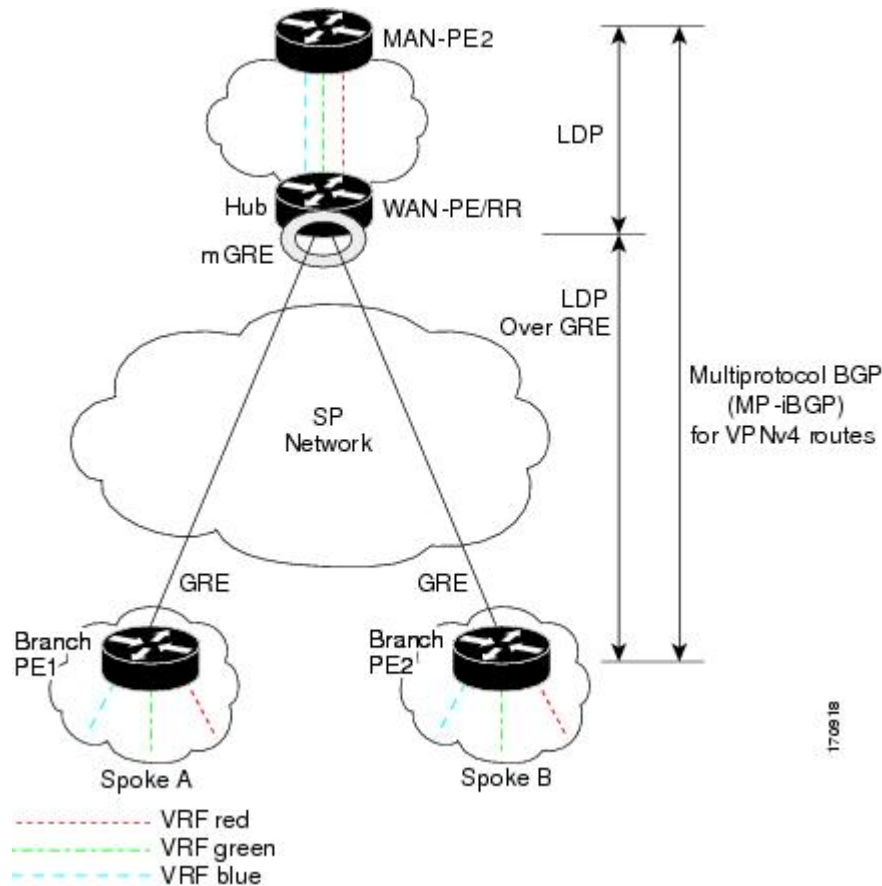


DMVPN--Enabling Traffic Segmentation Within DMVPN

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 below and the corresponding bullets explain how traffic segmentation within DMVPN works.

Figure 3 Traffic Segmentation with DMVPN



- The hub shown in the diagram is a WAN-PE and a route reflector, and the spokes (PE routers) are clients.
- There are three VRFs, designated “red,” “green,” and “blue.”
- Each spoke has both a neighbor relationship with the hub (multiprotocol Border Gateway Protocol [MP-iBGP] peering) and a GRE tunnel to the hub.
- Each spoke advertises its routes and VPNv4 prefixes to the hub.
- The hub sets its own IP address as the next-hop route for all the VPNv4 addresses it learns from the spokes and assigns a local MPLS label for each VPN when it advertises routes back to the spokes. As a result, traffic from Spoke A to Spoke B is routed via the hub.

An example illustrates the process:

- 1 Spoke A advertises a VPNv4 route to the hub, and applies the label X to the VPN.
- 2 The hub changes the label to Y when the hub advertises the route to Spoke B.

- 3 When Spoke B has traffic to send to Spoke A, it applies the *Y* label, and the traffic goes to the hub.
- 4 The hub swaps the VPN label, by removing the *Y* label and applying an *X* label, and sends the traffic to Spoke A.

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 recommend 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.

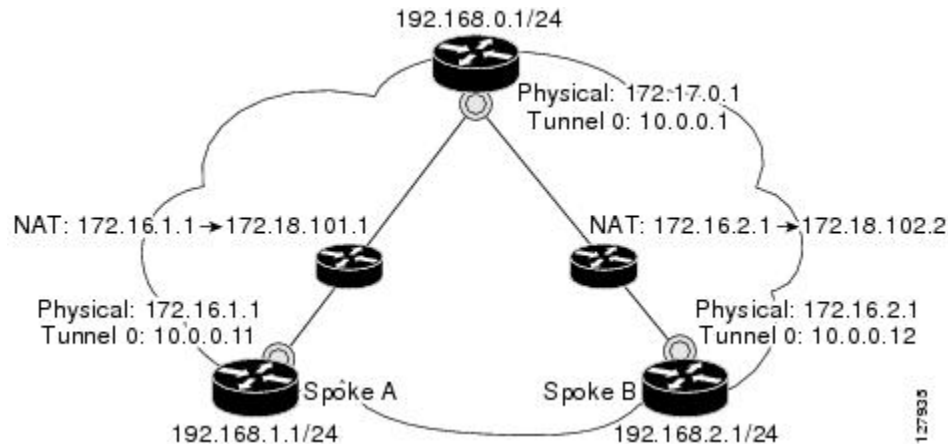
For these NAT-Transparency Aware enhancements to work, you must use IPsec transport mode on the transform set. Also, even though NAT-Transparency (IKE and IPsec) can support two peers (IKE and IPsec) being translated to the same IP address (using the UDP ports to differentiate them), this functionality 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.

The diagram below 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 10](#)
- [Configuring the Hub for DMVPN, page 12](#)
- [Configuring the Spoke for DMVPN, page 16](#)
- [Configuring the Forwarding of Clear-Text Data IP Packets into a VRF, page 20](#)
- [Configuring the Forwarding of Encrypted Tunnel Packets into a VRF, page 21](#)
- [Configuring DMVPN--Traffic Segmentation Within DMVPN, page 22](#)
- [Troubleshooting Dynamic Multipoint VPN \(DMVPN\), page 28](#)

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.

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
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <p>Enter your password if prompted.</p>
<p>Step 2 <code>configure terminal</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>crypto ipsec profile name</code></p> <p>Example:</p> <pre>Router(config)# crypto ipsec profile vpnprof</pre>	<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.
<p>Step 4 <code>set transform-set transform-set-name</code></p> <p>Example:</p> <pre>Router(config-crypto-map)# set transform-set trans2</pre>	<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.
<p>Step 5 <code>set identity</code></p> <p>Example:</p> <pre>Router(config-crypto-map)# set identity</pre>	<p>(Optional) Specifies identity restrictions to be used with the IPsec profile.</p>
<p>Step 6 <code>set security association lifetime {seconds seconds kilobytes kilobytes}</code></p> <p>Example:</p> <pre>Router(config-crypto-map)# set security association lifetime seconds 1800</pre>	<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.

Command or Action	Purpose
<p>Step 7 <code>set pfs [group1 group2]</code></p> <p>Example:</p> <pre>Router(config-crypto-map)# set pfs group2</pre>	<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, page 12](#)

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:



Note

NHRP network IDs are locally significant and can be different. It makes sense from a deployment and maintenance perspective to use unique **network ID** numbers (using the `ip nhrp network-id` command) across all routers in a DMVPN network, but it is not necessary that they be the same.

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: <pre>Router> enable</pre>	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: <pre>Router(config)# interface tunnel 5</pre>	Configures a tunnel interface and enters interface configuration mode <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.

Command or Action	Purpose
<p>Step 4 <code>ip address <i>ip-address mask secondary</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip address 10.0.0.1 255.255.255.0</pre>	<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>
<p>Step 5 <code>ip mtu <i>bytes</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip mtu 1400</pre>	<p>Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.</p>
<p>Step 6 <code>ip nhrp authentication <i>string</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp authentication donttell</pre>	<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>
<p>Step 7 <code>ip nhrp map multicast dynamic</code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp map multicast dynamic</pre>	<p>Allows NHRP to automatically add spoke routers to the multicast NHRP mappings.</p>
<p>Step 8 <code>ip nhrp network-id <i>number</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp network- id 99</pre>	<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.
<p>Step 9 <code>tunnel source {<i>ip-address</i> <i>type number</i>}</code></p> <p>Example:</p> <pre>Router (config-if)# tunnel source Ethernet0</pre>	<p>Sets source address for a tunnel interface.</p>

Command or Action	Purpose
<p>Step 10 <code>tunnel key <i>key-number</i></code></p> <p>Example:</p> <pre>Router (config-if)# tunnel key 100000</pre>	<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>
<p>Step 11 <code>tunnel mode gre multipoint</code></p> <p>Example:</p> <pre>Router(config-if)# tunnel mode gre multipoint</pre>	<p>Sets the encapsulation mode to mGRE for the tunnel interface.</p>
<p>Step 12 <code>tunnel protection ipsec profile <i>name</i></code></p> <p>Example:</p> <pre>Router(config-if)# tunnel protection ipsec profile vpnprof</pre>	<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.
<p>Step 13 <code>bandwidth <i>kbps</i></code></p> <p>Example:</p> <pre>Router(config-if)# bandwidth 1000</pre>	<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>
<p>Step 14 <code>ip tcp adjust-mss <i>max-segment-size</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip tcp adjust- mss 1360</pre>	<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>
<p>Step 15 <code>ip nhrp holdtime <i>seconds</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp holdtime 450</pre>	<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.

Command or Action	Purpose
Step 16 <code>delay number</code> Example: <code>Router(config-if)# delay 1000</code>	(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 Spoke for DMVPN

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



Note

NHRP network IDs are locally significant and can be different. It makes sense from a deployment and maintenance perspective to use unique **network ID** numbers (using the **ip nhrp network-id** command) across all routers in a DMVPN network, but it is not necessary that they be the same.

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. Do one of the following:
 - **tunnel mode gre multipoint**
 -
 -
 -
 -
 - **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	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<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:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>interface tunnel <i>number</i></p> <p>Example:</p> <pre>Router(config)# interface tunnel 5</pre>	<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:</p> <pre>Router(config-if)# ip address 10.0.0.2 255.255.255.0</pre>	<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:</p> <pre>Router(config-if)# ip mtu 1400</pre>	<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:</p> <pre>Router(config-if)# ip nhrp authentication donttell</pre>	<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>

Command or Action	Purpose
<p>Step 7 <code>ip nhrp map <i>hub-tunnel-ip-address</i> <i>hub-physical-ip-address</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1</pre>	<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.
<p>Step 8 <code>ip nhrp map multicast <i>hub-physical-ip-address</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp map multicast 172.17.0.1</pre>	<p>Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.</p>
<p>Step 9 <code>ip nhrp nhs <i>hub-tunnel-ip-address</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp nhs 10.0.0.1</pre>	<p>Configures the hub router as the NHRP next-hop server.</p>
<p>Step 10 <code>ip nhrp network-id <i>number</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp network- id 99</pre>	<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.
<p>Step 11 <code>tunnel source {<i>ip-address</i> <i>type number</i>}</code></p> <p>Example:</p> <pre>Router (config-if)# tunnel source Ethernet0</pre>	<p>Sets the source address for a tunnel interface.</p>
<p>Step 12 <code>tunnel key <i>key-number</i></code></p> <p>Example:</p> <pre>Router (config-if)# tunnel key 100000</pre>	<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>

Command or Action	Purpose
<p>Step 13 Do one of the following:</p> <ul style="list-style-type: none"> • tunnel mode gre multipoint • • • • • tunnel destination <i>hub-physical-ip-address</i> <p>Example:</p> <pre>Router(config-if)# tunnel mode gre multipoint</pre> <p>Example:</p> <p>Example:</p> <pre>Router(config-if)# tunnel destination 172.17.0.1</pre>	<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>
<p>Step 14 tunnel protection ipsec profile <i>name</i></p> <p>Example:</p> <pre>Router(config-if)# tunnel protection ipsec profile vpnprof</pre>	<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.
<p>Step 15 bandwidth <i>kbps</i></p> <p>Example:</p> <pre>Router(config-if)# bandwidth 1000</pre>	<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>

Command or Action	Purpose
<p>Step 16 <code>ip tcp adjust-mss <i>max-segment-size</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip tcp adjust- mss 1360</pre>	<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>
<p>Step 17 <code>ip nhrp holdtime <i>seconds</i></code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp holdtime 450</pre>	<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.
<p>Step 18 <code>delay <i>number</i></code></p> <p>Example:</p> <pre>Router(config-if)# delay 1000</pre>	<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 Forwarding of Clear-Text Data IP Packets into a VRF

To configure the forwarding of clear-text date 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
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <p>Enter your password if prompted.</p>

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

Command or Action	Purpose
Step 3 <code>interface <i>type number</i></code> Example: <pre>Router (config)# interface tunnel0</pre>	Configures an interface type and enters interface configuration mode.
Step 4 <code>tunnel vrf <i>vrf-name</i></code> Example: <pre>Router (config-if)# tunnel vrf RED</pre>	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:

- [Prerequisites, page 22](#)
- [Enabling MPLS on the VPN Tunnel, page 22](#)
- [Configuring Multiprotocol BGP on the Hub Router, page 23](#)
- [Configuring Multiprotocol BGP on the Spoke Routers, page 26](#)

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 task and the Configuring the Spoke for DMVPN. For details about VRF configuration, see the Configuring the Forwarding of Clear-Text Data IP Packets into a VRF task and the Configuring the Forwarding of Encrypted Tunnel Packets into a VRF task.

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).

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 vpvv4**
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. <ul style="list-style-type: none"> • 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
<p>Step 5 neighbor <i>ipaddress</i> update-source <i>interface</i></p> <p>Example:</p> <pre>Router (config)# neighbor 10.10.10.11 update-source Tunnel1</pre>	<p>Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.</p>
<p>Step 6 address-family vpv4</p> <p>Example:</p> <pre>Router (config)# address-family vpv4</pre>	<p>Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.</p>
<p>Step 7 neighbor <i>ipaddress</i> activate</p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.11 activate</pre>	<p>Enables the exchange of information with a BGP neighbor.</p>
<p>Step 8 neighbor <i>ipaddress</i> send-community extended</p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.11 send-community extended</pre>	<p>Specifies that extended community attributes should be sent to a BGP neighbor.</p>
<p>Step 9 neighbor <i>ipaddress</i> route-reflector-client</p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.11 route-reflector-client</pre>	<p>Configures the router as a BGP route reflector and configures the specified neighbor as its client.</p>
<p>Step 10 neighbor <i>ipaddress</i> route-map <i>nexthop out</i></p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.11 route-map nexthop out</pre>	<p>Forces all traffic to be routed via the hub.</p>
<p>Step 11 exit-address-family</p> <p>Example:</p> <pre>Router (config)# exit-address-family</pre>	<p>Exits the address family configuration mode for VPNv4.</p>

Command or Action	Purpose
Step 12 <code>address-family ipv4 vrf-name</code> Example: <pre>Router (config)# address-family ipv4 vrf red</pre>	Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.
Step 13 <code>redistribute connected</code> Example: <pre>Router (config)# redistribute connected</pre>	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 14 <code>route-map</code> Example: <pre>Router (config)# route-map nexthop permit 10</pre>	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: <pre>Router (config)# set ip next-hop 10.0.0.1</pre>	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	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables higher privilege levels, such as privileged EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
Step 3	<p>router bgp</p> <p>Example:</p> <pre>Router (config)# router bgp 1</pre>	<p>Enters BGP configuration mode.</p>
Step 4	<p>neighbor <i>ipaddress</i> remote-as <i>as - number</i></p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.1 remote-as 1</pre>	<p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p>
Step 5	<p>neighbor <i>ipaddress</i> update-source <i>interface</i></p> <p>Example:</p> <pre>Router (config)# neighbor 10.10.10.1 update-source Tunnell</pre>	<p>Configures the Cisco IOS software to allow BGP sessions to use any operational interface for TCP connections.</p>
Step 6	<p>address-family vpnv4</p> <p>Example:</p> <pre>Router (config)# address-family vpnv4</pre>	<p>Enters address family configuration mode to configure a routing session using Virtual Private Network (VPN) Version 4 address prefixes.</p>
Step 7	<p>neighbor <i>ipaddress</i> activate</p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.1 activate</pre>	<p>Enables the exchange of information with a BGP neighbor.</p>

Command or Action	Purpose
<p>Step 8 <code>neighbor ipaddress send-community extended</code></p> <p>Example:</p> <pre>Router (config)# neighbor 10.0.0.1 send-community extended</pre>	<p>Specifies that extended community attributes should be sent to a BGP neighbor.</p>
<p>Step 9 <code>exit-address-family</code></p> <p>Example:</p> <pre>Router (config)# exit-address-family</pre>	<p>Exits the address family configuration mode.</p>
<p>Step 10 <code>address-family ipv4 vrf-name</code></p> <p>Example:</p> <pre>Router (config)# address-family ipv4 vrf red</pre>	<p>Enters address family configuration mode to configure a routing session using standard IP Version 4 address prefixes.</p>
<p>Step 11 <code>redistribute connected</code></p> <p>Example:</p> <pre>Router (config)# redistribute connected</pre>	<p>Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.</p>
<p>Step 12 <code>exit-address-family</code></p> <p>Example:</p> <pre>Router (config)# exit-address-family</pre>	<p>Exits the address family configuration mode.</p> <p>Note Repeat Steps 10-12 for each VRF.</p>

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. The **clear dmvpn session** command is used to clear DMVPN sessions.
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:
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:
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:
5. The **debug nhrp error** command displays information about NHRP error activity. The following example shows how to enable debugging for NHRP error messages:
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:
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:
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.
9. The **show crypto map** command displays the crypto map configuration.
10. The **show dmvpn** command displays DMVPN specific session information. The following example shows example summary output:
11. The **show ip nhrp traffic** command displays NHRP statistics. The following example shows output for a specific tunnel, tunnel7:

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:

Example:

```
%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:

Example:

```
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.

Example:

```
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:

Example:

```
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:

Example:

```
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`

Example:

```
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, page 32](#)

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

- [Example Hub Configuration for DMVPN, page 32](#)
- [Example Spoke Configuration for DMVPN, page 33](#)
- [Example VRF Aware DMVPN, page 34](#)
- [Example 2547oDMVPN with Traffic Segmentation \(with BGP only\), page 36](#)
- [Example 2547oDMVPN with Traffic Segmentation \(Enterprise Branch\), page 40](#)

Example Hub Configuration for DMVPN

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.

Example Spoke Configuration for DMVPN

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

```

Example VRF Aware DMVPN

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 Tunnell
! 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!

```

Example 2547oDMVPN with Traffic Segmentation (with BGP only)

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 Tunnel1
  neighbor 10.0.0.12 remote-as 1
  neighbor 10.0.0.12 update-source Tunnel1
  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 Tunnel1
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

```

Example 2547oDMVPN with Traffic Segmentation (Enterprise Branch)

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 Tunnell
  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 tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
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 NLRIs 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 NLRIs 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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Call Admission Control	Call Admission Control for IKE
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: Secure Connectivity</i>
IPsec configuration tasks	The chapter “Configuring Security for VPNs with IPsec” in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
Configuring VRF-Aware IPsec	The chapter “VRF-Aware IPsec” in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
Configuring MPLS	The chapter “Configuring Multiprotocol Label Switching” in the <i>Cisco IOS Multiprotocol Label Switching Configuration Guide</i>
Configuring BGP	The chapter “Cisco BGP Overview” in the <i>Cisco IOS IP Routing: BGP Protocols Configuration Guide</i>
System messages	System Message Guide
Defining and configuring ISAKMP profiles	“Certificate to ISAKMP Profile Mapping” chapter in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software 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
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Dynamic Multipoint VPN (DMVPN)

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

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

Table 1 **Feature Information for Dynamic Multipoint VPN (DMVPN)**

Feature Name	Releases	Feature Information
DMVPN--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	<p>DMVPN session manageability was expanded with DMVPN specific commands for debugging, show output, session and counter control, and system log information.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Troubleshooting Dynamic Multipoint VPN (DMVPN) <p>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.</p>
DMVPN Phase 2	12.2(18)SXE 12.3(9)a 12.3(8)T1	<p>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.</p> <p>In Release 12.2(18)SXE, support was added for the Cisco Catalyst 6500 series switch and the Cisco 7600 series router.</p>

Feature Name	Releases	Feature Information
--	12.3(6) 12.3(7)T	<p>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.</p> <p>The enhancements added in Cisco IOS Release 12.3(6) were integrated into Cisco IOS Release 12.3(7)T.</p>
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.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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



Per-Tunnel QoS for DMVPN

The Per-Tunnel QoS for DMVPN feature introduces per-tunnel quality of service (QoS) support for Dynamic Multipoint VPN (DMVPN) tunnel interfaces and increases QoS performance for IP Security (IPSec) Virtual Tunnel Interfaces (VTIs).

- [Finding Feature Information, page 51](#)
- [Prerequisites for Per-Tunnel QoS for DMVPN, page 51](#)
- [Restrictions for Per-Tunnel QoS for DMVPN, page 51](#)
- [Information About Per-Tunnel QoS for DMVPN, page 52](#)
- [How to Configure Per-Tunnel QoS for DMVPN, page 53](#)
- [Configuration Examples for Per-Tunnel QoS for DMVPN, page 57](#)
- [Additional References, page 62](#)
- [Feature Information for Per-Tunnel QoS for DMVPN, page 63](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for Per-Tunnel QoS for DMVPN

Before you can configure per-tunnel QoS for DMVPN, you must configure Cisco Express Forwarding (CEF) switching.

Restrictions for Per-Tunnel QoS for DMVPN

- You cannot configure a per-tunnel QoS policy on a tunnel interface *and* a separate QoS policy on the outbound physical interface at the same time.
- You can attach a per-tunnel QoS policy on the tunnel only in the egress direction.

Information About Per-Tunnel QoS for DMVPN

This feature lets you apply a QoS policy on a DMVPN hub on a per-tunnel-instance (per-spoke) basis in the egress direction for DMVPN hub-to-spoke tunnels. This lets you shape the tunnel traffic to individual spokes (a parent policy) and differentiate individual data flows going through the tunnel for policing (a child policy). The QoS policy that the hub uses for a specific spoke is selected according to the specific Next Hop Resolution Protocol (NHRP) group into which that spoke is configured. Although you can configure many spokes into the same NHRP group, the tunnel traffic for each spoke is measured individually for shaping and policing.

For IPsec VTIs, per-tunnel QoS support existed before this feature was introduced. But this feature does improve performance (because HQF performs the queuing at the egress physical interface instead of in the switching path).

You can use this feature with DMVPN with or without IPsec.

With Per-Tunnel QoS for DMVPN, the queuing and shaping is performed at the outbound physical interface for the GRE/IPsec tunnel packets. This means that the GRE header, the IPsec header and the L2 (for the physical interface) header are included in the packet-size calculations for shaping and bandwidth queuing of packets under QoS.

To configure the Per-Tunnel QoS for DMVPN feature, you should understand the following concepts:

- [Benefits of Per-Tunnel QoS for DMVPN, page 52](#)
- [NHRP QoS Provisioning for DMVPN, page 52](#)

Benefits of Per-Tunnel QoS for DMVPN

Without this feature, QoS on a DMVPN hub could only be configured to either measure outbound traffic in the aggregate (over all spokes) or (with extensive manual configuration) the DMVPN hub could measure outbound traffic on a per-spoke basis.

Per-Tunnel QoS for DMVPN provides the following benefits:

- The QoS policy is attached to the DMVPN hub, and criteria for matching the tunnel traffic are set up automatically as each spoke registers with the hub (which means that extensive manual configuration is not needed).
- Traffic can be regulated from the hub to spokes on a per-spoke basis.
- The hub cannot send excessive traffic to (and overrun) a small spoke.
- The amount of outbound hub bandwidth that a “greedy” spoke can consume can be limited, and therefore it cannot monopolize a hub's resources and starve other spokes.

NHRP QoS Provisioning for DMVPN

NHRP performs the provisioning for Per-Tunnel QoS for DMVPN by using NHRP groups.

An NHRP group, which is the new concept introduced by this feature, is the group identity information signaled by a DMVPN node (a spoke) to the DMVPN hub. The hub uses this information to select a locally-defined QoS policy instance for the remote node.

You configure the NHRP group name on the spoke router on the DMVPN GRE tunnel interface. The NHRP group name is communicated to the hub in each of the periodic NHRP registration requests sent from the spoke to the hub. A new Cisco vendor-private NHRP extension **nhrp-group** is defined to carry the NHRP group string in the NHRP registration request.

The NHRP-group-to-QoS-policy mappings are configured on the hub DMVPN GRE tunnel interface. The NHRP group string received from a spoke is mapped to a QoS policy, which is applied for that hub-to-spoke tunnel in the egress direction.

When the NHRP group is configured on the spoke, the group is not immediately sent to the *hub*, but is sent in the next periodic registration request. The spoke can be in only one NHRP group per GRE tunnel interface. If a spoke is configured as a part of two or more DMVPN networks (multiple GRE tunnel interfaces), then the spoke can have a different NHRP group name on each of the GRE tunnel interfaces.

If an NHRP group is not received from the spoke, then a QoS policy is *not* applied, and any existing QoS policy applied for that spoke is removed. If an NHRP group is received from the spoke when previous NHRP registrations did not have an NHRP group, then the corresponding QoS policy is applied. If the same NHRP group name is received from a spoke as was received in the previous NHRP registration request, then no action is taken (because a QoS policy would have already been applied for that spoke). If a different NHRP group is received from the spoke than what was received in the previous NHRP registration request, any applied QoS policy is removed, and the QoS policy corresponding to the new NHRP group is applied.

How to Configure Per-Tunnel QoS for DMVPN

To configure the Per-Tunnel QoS for DMVPN feature, you define an NHRP group on the spokes and then map the NHRP group to a QoS policy on the hub.

- [Configuring an NHRP Group on a Spoke, page 53](#)
- [Mapping an NHRP Group to a QoS Policy on the Hub, page 54](#)
- [Verifying Per-Tunnel QoS for DMVPN, page 55](#)

Configuring an NHRP Group on a Spoke

To configure an NHRP group on a spoke, perform the steps in this section.

The spoke and the hub must already be configured for DMVPN without the Per-Tunnel QoS feature.

SUMMARY STEPS

1. **enable**
2. **configure** {terminal | memory | network}
3. **interface tunnel** *number*
4. **ip nhrp group** *group-name*
5. **end**

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Router> enable	Enables higher privilege levels (such as privileged EXEC mode). <ul style="list-style-type: none"> • Enter your password if prompted.

Command or Action	Purpose
Step 2 <code>configure {terminal memory network}</code> Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3 <code>interface tunnel number</code> Example: <pre>Router(config)# interface tunnel 1</pre>	Configures a tunnel interface and enters interface configuration mode. <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 that you can create.
Step 4 <code>ip nhrp group group-name</code> Example: <pre>Router(config-if# ip nhrp group spoke_group1</pre>	Configures an NHRP group on the spoke.
Step 5 <code>end</code> Example: <pre>Router(config-if)# end</pre>	Returns to privileged EXEC mode.

Mapping an NHRP Group to a QoS Policy on the Hub

To configure an NHRP-group-to-QoS-policy mapping on a hub, perform the steps in this section.



Note

The `qos pre-classify` command is not required with this feature, because classification is performed before encapsulation.

The spoke and the hub must already be configured for DMVPN without the Per-Tunnel QoS feature.

SUMMARY STEPS

1. `enable`
2. `configure {terminal | memory | network}`
3. `interface tunnel number`
4. `ip nhrp map group group-name service-policy output qos-policy-map-name`
5. `end`

DETAILED STEPS

Command or Action	Purpose
<p>Step 1 <code>enable</code></p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables higher privilege levels (such as privileged EXEC mode).</p> <ul style="list-style-type: none"> Enter your password if prompted.
<p>Step 2 <code>configure {terminal memory network}</code></p> <p>Example:</p> <pre>Router# configure terminal</pre>	<p>Enters global configuration mode.</p>
<p>Step 3 <code>interface tunnel number</code></p> <p>Example:</p> <pre>Router(config)# interface tunnel 1</pre>	<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 that you can create.
<p>Step 4 <code>ip nhrp map group group-name service-policy output qos-policy-map-name</code></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp map group spoke_group1 service-policy output group1_parent</pre>	<p>Adds the NHRP group to the QoS policy mapping on the hub.</p>
<p>Step 5 <code>end</code></p> <p>Example:</p> <pre>Router(config-if)# end</pre>	<p>Returns to privileged EXEC mode.</p>

Verifying Per-Tunnel QoS for DMVPN

To verify the configuration of Per-Tunnel QoS for DMVPN, perform the steps in this section.

SUMMARY STEPS

1. enable
2. show dmvpn detail
3. show ip nhrp
4. show ip nhrp group-map [*group-name*]
5. show policy-map multipoint [tunnel *tunnel-interface-number*]
6. show tunnel endpoints

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: <pre>Router> enable</pre>	Enables higher privilege levels (such as privileged EXEC mode). <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2 show dmvpn detail Example: <pre>Router# show dmvpn detail</pre>	Displays detailed DMVPN information for each session, including the Next Hop Server (NHS) and NHS status, crypto session information, and socket details. Also displays the NHRP group received from the spoke and the QoS policy applied to the spoke tunnel.
Step 3 show ip nhrp Example: <pre>Router# show ip nhrp</pre>	Displays the NHRP cache and the NHRP group received from the spoke.
Step 4 show ip nhrp group-map [<i>group-name</i>] Example: <pre>Router# show ip nhrp group-map</pre> Example: <pre>Router# show ip nhrp group-map group1-parent</pre>	Displays the group-to-policy maps configured on the hub. Also displays the tunnels on which the QoS policy is applied.

Command or Action	Purpose
<p>Step 5 <code>show policy-map multipoint [tunnel <i>tunnel-interface-number</i>]</code></p> <p>Example:</p> <pre>Router# show policy-map multipoint</pre> <p>Example:</p> <pre>Router# show policy-map multipoint tunnel 1</pre>	<p>Displays QoS policy details applied to multipoint tunnels.</p>
<p>Step 6 <code>show tunnel endpoints</code></p> <p>Example:</p> <pre>Router# show tunnel endpoints</pre>	<p>Displays information about the source and destination endpoints for multipoint tunnels. Also displays the QoS policy applied on the spoke tunnel.</p>

Configuration Examples for Per-Tunnel QoS for DMVPN

- [Configuring an NHRP Group on a Spoke Example, page 57](#)
- [Mapping an NHRP Group to a QoS Policy on the Hub Example, page 58](#)
- [Verifying Per-Tunnel QoS for DMVPN Examples, page 59](#)

Configuring an NHRP Group on a Spoke Example

The following example shows how to configure two NHRP groups on three spokes.

Configuring the First Spoke

```
interface Tunnel 1
 ip address 20.1.1.2 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication testing
 ip nhrp group spoke_group1
 ip nhrp map 20.1.1.1 200.0.0.1
 ip nhrp map multicast 200.0.0.1
 ip nhrp network-id 172176366
 ip nhrp holdtime 300
 ip tcp adjust-mss 1360
 ip nhrp nhs 20.1.1.1
 tunnel source Ethernet 0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile DMVPN
interface Ethernet 0/0
 ip address 200.0.0.2 255.255.255.0
```

Configuring the Second Spoke

```

interface Tunnel 1
 ip address 20.1.1.3 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication testing
 ip nhrp group spoke_group1
 ip nhrp map 20.1.1.1 200.0.0.1
 ip nhrp map multicast 200.0.0.1
 ip nhrp network-id 172176366
 ip nhrp holdtime 300
 ip tcp adjust-mss 1360
 ip nhrp nhs 20.1.1.1
 tunnel source Ethernet 0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile DMVPN
interface Ethernet 0/0
 ip address 200.0.0.3 255.255.255.0

```

Configuring the Third Spoke

```

interface Tunnel 1
 ip address 20.1.1.4 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication testing
 ip nhrp group spoke_group2
 ip nhrp map 20.1.1.1 200.0.0.1
 ip nhrp map multicast 200.0.0.1
 ip nhrp network-id 172176366
 ip nhrp holdtime 300
 ip tcp adjust-mss 1360
 ip nhrp nhs 20.1.1.1
 tunnel source Ethernet 0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile DMVPN
interface Ethernet 0/0
 ip address 200.0.0.4 255.255.255.0

```

Mapping an NHRP Group to a QoS Policy on the Hub Example

The following example shows how to map NHRP groups to a QoS policy on the hub. The example shows a hierarchical QoS policy (parent: group1_parent/group2_parent, child: group1/group2) that will be used for Per-Tunnel QoS for DMVPN. The example also shows how to map the NHRP group spoke_group1 to the QoS policy group1_parent and map the NHRP group spoke_group2 to the QoS policy group2_parent on the hub:

```

class-map match-all group1_Routing
 match ip precedence 6
class-map match-all group2_Routing
 match ip precedence 6
class-map match-all group2_voice
 match access-group 100
class-map match-all group1_voice
 match access-group 100
policy-map group1
 class group1_voice
  priority 1000
 class group1_Routing
  bandwidth percent 20
policy-map group1_parent
 class class-default
  shape average 3000000
  service-policy group1
policy-map group2
 class group2_voice

```



```

    priority percent 20
  class group2_Routing
    bandwidth percent 10
  policy-map group2_parent
  class class-default
    shape average 2000000
  service-policy group2
interface Tunnel 1
  ip address 20.1.1.1 255.255.255.0
  no ip redirects
  ip mtu 1400
  ip nhrp authentication testing
  ip nhrp map multicast dynamic
  ip nhrp map group spoke_group1 service-policy output group1_parent
  ip nhrp map group spoke_group2 service-policy output group2_parent
  ip nhrp network-id 172176366
  ip nhrp holdtime 300
  ip nhrp registration no-unique
  tunnel source Ethernet0/0
  tunnel mode gre multipoint
  tunnel protection ipsec profile DMVPN
interface Ethernet 0/0
  ip address 200.0.0.1 255.255.255.0

```

Verifying Per-Tunnel QoS for DMVPN Examples

The following example shows how to display the information about NHRP groups received from the spokes as well as the QoS policy that is applied to each spoke tunnel. You enter this command on the hub:

```

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
NHS Status: E --> Expecting Replies, R --> Responding
UpDn Time --> Up or Down Time for a Tunnel
=====
Interface Tunnell is up/up, Addr. is 20.1.1.1, VRF ""
  Tunnel Src./Dest. addr: 200.0.0.1/MGRE, Tunnel VRF ""
  Protocol/Transport: "multi-GRE/IP", Protect "DMVPN"
Type:Hub, Total NBMA Peers (v4/v6): 3
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
  1      200.0.0.2      20.1.1.2  UP 00:19:20  D      20.1.1.2/32
NHRP group: spoke_group1
  Output QoS service-policy applied: group1_parent
  1      200.0.0.3      20.1.1.3  UP 00:19:20  D      20.1.1.3/32
NHRP group: spoke_group1
  Output QoS service-policy applied: group1_parent
  1      200.0.0.4      20.1.1.4  UP 00:19:23  D      20.1.1.4/32
NHRP group: spoke_group2
  Output QoS service-policy applied: group2_parent
Crypto Session Details:
-----
Interface: Tunnell
Session: [0x04AC1D00]
  IKE SA: local 200.0.0.1/500 remote 200.0.0.2/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phasel_id: 200.0.0.2
  IPSEC FLOW: permit 47 host 200.0.0.1 host 200.0.0.2
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0x9B264329, transform : ah-md5-hmac
    Socket State: Open
Interface: Tunnell
Session: [0x04AC1C08]
  IKE SA: local 200.0.0.1/500 remote 200.0.0.3/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phasel_id: 200.0.0.3
  IPSEC FLOW: permit 47 host 200.0.0.1 host 200.0.0.3
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0x36FD56E2, transform : ah-md5-hmac
    Socket State: Open

```

```

Interface: Tunnell
Session: [0x04AC1B10]
IKE SA: local 200.0.0.1/500 remote 200.0.0.4/500 Active
Crypto Session Status: UP-ACTIVE
fvrf: (none), Phase1_id: 200.0.0.4
IPSEC FLOW: permit 47 host 200.0.0.1 host 200.0.0.4
Active SAs: 2, origin: crypto map
Outbound SPI : 0xAC96818F, transform : ah-md5-hmac
Socket State: Open
Pending DMVPN Sessions:

```

The following example shows how to display information about the NHRP groups that are received from the spokes. You enter this command on the hub:

```

Router# show ip nhrp
20.1.1.2/32 via 20.1.1.2
Tunnell created 00:22:49, expire 00:01:40
Type: dynamic, Flags: registered
NBMA address: 200.0.0.2
Group: spoke_group1
20.1.1.3/32 via 20.1.1.3
Tunnell created 00:22:48, expire 00:01:41
Type: dynamic, Flags: registered
NBMA address: 200.0.0.3
Group: spoke_group1
20.1.1.4/32 via 20.1.1.4
Tunnell created 00:22:52, expire 00:03:27
Type: dynamic, Flags: registered
NBMA address: 200.0.0.4
Group: spoke_group2

```

The following example shows how to display the details of NHRP group mappings on a hub and the list of tunnels using each of the NHRP groups defined in the mappings. You enter this command on the hub:

```

Router# show ip nhrp group-map
Interface: Tunnell
NHRP group: spoke_group1
QoS policy: group1_parent
Tunnels using the QoS policy:
Tunnel destination overlay/transport address
20.1.1.2/200.0.0.2
20.1.1.3/200.0.0.3
NHRP group: spoke_group2
QoS policy: group2_parent
Tunnels using the QoS policy:
Tunnel destination overlay/transport address
20.1.1.4/200.0.0.4

```

The following example shows how to display statistics about a specific QoS policy as it is applied to a tunnel endpoint. You enter this command on the hub:

```

Router# show policy-map multipoint
Interface Tunnell <--> 200.0.0.2
Service-policy output: group1_parent
Class-map: class-default (match-any)
 29 packets, 4988 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
Match: any
Queueing
queue limit 750 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 0/0
shape (average) cir 3000000, bc 12000, be 12000
target shape rate 3000000
Service-policy : group1
  queue stats for all priority classes:
  queue limit 250 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
Class-map: group1_voice (match-all)
 0 packets, 0 bytes

```

```

    5 minute offered rate 0 bps, drop rate 0 bps
    Match: access-group 100
    Priority: 1000 kbps, burst bytes 25000, b/w exceed drops: 0
Class-map: group1_Routing (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: ip precedence 6
  Queueing
    queue limit 150 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
    bandwidth 20% (600 kbps)
Class-map: class-default (match-any)
  29 packets, 4988 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
  queue limit 350 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
Interface Tunnell <--> 200.0.0.3
Service-policy output: group1_parent
  Class-map: class-default (match-any)
    29 packets, 4988 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
    Queueing
      queue limit 750 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 3000000, bc 12000, be 12000
      target shape rate 3000000
  Service-policy : group1
    queue stats for all priority classes:
      queue limit 250 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
    Class-map: group1_voice (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      Priority: 1000 kbps, burst bytes 25000, b/w exceed drops: 0
    Class-map: group1_Routing (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip precedence 6
      Queueing
        queue limit 150 packets
        (queue depth/total drops/no-buffer drops) 0/0/0
        (pkts output/bytes output) 0/0
        bandwidth 20% (600 kbps)
    Class-map: class-default (match-any)
      29 packets, 4988 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      queue limit 350 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
Interface Tunnell <--> 200.0.0.4
Service-policy output: group2_parent
  Class-map: class-default (match-any)
    14 packets, 2408 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
    Queueing
      queue limit 500 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      shape (average) cir 2000000, bc 8000, be 8000
      target shape rate 2000000
  Service-policy : group2
    queue stats for all priority classes:
      queue limit 100 packets
      (queue depth/total drops/no-buffer drops) 0/0/0

```

```

(pkts output/bytes output) 0/0
Class-map: group2_voice (match-all)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: access-group 100
 Priority: 20% (400 kbps), burst bytes 10000, b/w exceed drops: 0
Class-map: group2_Routing (match-all)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: ip precedence 6
 Queueing
 queue limit 50 packets
 (queue depth/total drops/no-buffer drops) 0/0/0
 (pkts output/bytes output) 0/0
 bandwidth 10% (200 kbps)
Class-map: class-default (match-any)
 14 packets, 2408 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
 Match: any
 queue limit 350 packets
 (queue depth/total drops/no-buffer drops) 0/0/0
 (pkts output/bytes output) 0/0

```

Additional References

The following sections provide references related to the Per-Tunnel QoS for DMVPN feature.

Related Documents

Related Topic	Document Title
General information about QoS	“Quality of Service Overview” module in the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i> .
Configuring hierarchical queuing	<ul style="list-style-type: none"> “QoS--Hierarchical Queuing Framework (HQF)” module in the <i>Cisco IOS Quality of Service Solutions Configuration Guide</i>. <i>Cisco IOS Quality of Service Solutions Command Reference</i>
Configuring NHRP	<ul style="list-style-type: none"> “Configuring NHRP” module in the <i>Cisco IOS IP Addressing Services Configuration Guide</i>. <i>Cisco IOS IP Addressing Services Command Reference</i>

Standards

Standard	Title
None	--

MIBs

MIB	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

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

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

Feature Information for Per-Tunnel QoS for DMVPN

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

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

Table 2 **Feature Information for Per-Tunnel QoS for DMVPN**

Feature Name	Releases	Feature Information
Per-Tunnel QoS for DMVPN	12.4(22)T	<p>The Per-Tunnel QoS for DMVPN feature introduces per-tunnel QoS support for DMVPN and increases per-tunnel QoS performance for IPsec VTIs.</p> <p>The following sections provide information about this feature:</p> <p>The following commands were introduced or modified: ip nhrp group, ip nhrp map, ip nhrp map group, show dmvpn, show ip nhrp, show ip nhrp group-map, show policy-map multipoint tunnel.</p>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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



NHRP MIB

The Cisco NHRP MIB feature introduces support for the NHRP MIB, which helps to manage and monitor the Next Hop Resolution Protocol (NHRP) via Simple Network Management Protocol (SNMP). Statistics can be collected and monitored via standards-based SNMP techniques (get operations) to query objects defined in the NHRP MIB. The NHRP MIB is VRF aware and supports VRF aware queries.

- [Finding Feature Information, page 65](#)
- [Prerequisites for NHRP MIB, page 65](#)
- [Restrictions for NHRP MIB, page 65](#)
- [Information About NHRP MIB, page 66](#)
- [How to Use NHRP MIB, page 66](#)
- [Configuration Examples for NHRP MIB, page 67](#)
- [Additional References, page 69](#)
- [Feature Information for NHRP MIB, page 70](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Prerequisites for NHRP MIB

- You should be familiar with configuring SNMP.

Restrictions for NHRP MIB

- Cisco does not support all the MIB variables defined in RFC-2677, Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP). For a list of variables supported and other caveats of this feature, see the Agent Capabilities file. Cisco does not support the set operations defined in RFC-2677.

Information About NHRP MIB

- [CISCO-NHRP-MIB, page 66](#)
- [RFC-2677, page 66](#)

CISCO-NHRP-MIB

CISCO-NHRP-MIB provides NHRP MIB information on managed objects relating to clients only, servers only, and clients and servers.

The NHRP MIB module contains ten tables of objects as follows:

- NHRP Cache Table
- NHRP Purge Request Table
- NHRP Client Table
- NHRP Client Registration Table
- NHRP Client NHS Table
- NHRP Client Statistics Table
- NHRP Server Table
- NHRP Server Cache Table
- NHRP Server NHC Table
- NHRP Server Statistics Table

The Cisco implementation supports all of the tables except the NHRP Purge Request Table.

RFC-2677

RFC-2677 - Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP), describes managed objects that can be used to remotely monitor NHRP using SNMP and provide management information on the performance of NHRP.

How to Use NHRP MIB

No special configuration is needed for this feature. The SNMP framework can be used to manage NHRP MIB. See the section “Configuration Examples for NHRP MIB” for an example of how to manage a VRF aware NHRP MIB.

- [Verifying NHRP MIB Status, page 66](#)

Verifying NHRP MIB Status

Use this task to verify the NHRP MIB status.

SUMMARY STEPS

1. **enable**
2. `show snmp mib nhrp status`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Router> enable</pre>	<p>Enables EXEC mode.</p> <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<p>show snmp mib nhrp status</p> <p>Example:</p> <pre>Router# show snmp mib nhrp status</pre>	<p>Displays the status of the NHRP MIB.</p>

Configuration Examples for NHRP MIB

- [Example Verifying NHRP MIB Status, page 67](#)
- [Example VRF-Aware NHRP MIB Configuration, page 67](#)

Example Verifying NHRP MIB Status

The following output is from the show snmp mib nhrp status command:

```
Spoke_103# show snmp mib nhrp status
NHRP-SNMP Agent Feature: Enabled
NHRP-SNMP Tree State: Good
ListEnqueue Count = 0 Node Malloc Counts = 1
Spoke_103#
```

The “Enabled” status of “NHRP-SNMP Agent Feature:” indicates that the NHRP MIB is enabled. If the NHRP MIB was disabled, it would display “Disabled”. “ListEnqueue Count” and “Node Malloc Counts” counts are internal counts. “ListEnqueue Count” indicates how many nodes are queued for freeing. “Node Malloc Counts” displays how many nodes are allocated.

Example VRF-Aware NHRP MIB Configuration

The following is an example of how to configure a VRF Table with the name V3red, for monitoring by SNMP:

```
ip vrf V3red
 rd 198102
 ! Name of the SNMP VPN context
 context V3red_context
 !
 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
 !
```

```

crypto ipsec profile vpnprof
  set transform-set trans2
!
interface Tunnel0
  bandwidth 1000
! DMVPN tunnel for V3red VPN
  ip vrf forwarding V3red
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication donttell
  ip nhrp map multicast dynamic
  ip nhrp network-id 99
  ip nhrp holdtime 300
  no ip split-horizon eigrp 1
  ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  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
  address-family ipv4 vrf V3red
  network 10.0.0.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
  no auto-summary
  autonomous-system 1
  exit-address-family
!
! V2C Community ABC for VRF V3red
snmp-server group abc v2c context V3red_context read view_V3
snmp-server view view_V3 iso included
snmp-server community abc RO
snmp-server community public RO
snmp-server context V3red_context
!
!
snmp mib community-map abc context V3red_context
Spoke Configuration for DMVPN Example
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
!
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
  ip nhrp authentication donttell
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp map multicast 172.17.0.1
  ip nhrp network-id 99
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
!
interface Ethernet0
  ip address dhcp hostname Spoke1

```

```

!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.1.0 0.0.0.255

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of SNMP, SNMP MIBs, and how to configure SNMP on Cisco devices	The chapter “Configuring SNMP Support ” in the <i>Cisco IOS Network Management Configuration Guide</i>

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
CISCO-NHRP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2677	Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)

Technical Assistance

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

Feature Information for NHRP MIB

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

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

Table 3 *Feature Information for NHRP MIB*

Feature Name	Releases	Feature Information
NHRP MIB	12.4(20)T	<p>The Cisco NHRP MIB feature introduces support for the NHRP MIB, which helps to manage and monitor Next Hop Resolution Protocol (NHRP) via Simple Network Management Protocol (SNMP). Statistics can be collected and monitored via standards-based SNMP techniques (get operations) to query objects defined in the NHRP MIB.</p> <p>The following commands were introduced or modified: <code>debug snmp mib nhrp</code>, <code>show snmp mib nhrp status</code>.</p>

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party

trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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



DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

The DMVPN: Dynamic Tunnels Between Spokes Behind a NAT Device feature allows Next Hop Resolution Protocol (NHRP) spoke-to-spoke tunnels to be built in Dynamic Multipoint Virtual Private Networks (DMVPNs), even if one or more spokes is behind a Network Address Translation (NAT) device.

- [Finding Feature Information, page 73](#)
- [Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, page 73](#)
- [Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, page 74](#)
- [Additional References, page 78](#)
- [Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, page 79](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

In order for spokes to build tunnels between them, they need to know the post-NAT address of the other spoke.

Consider the following restrictions when using spoke-to-spoke tunneling in NAT environments:

- **Multiple NAT translations** --A packet can go across multiple NAT devices in a nonbroadcast multiaccess (NBMA) DMVPN cloud and make several (unimportant) translations before it reaches its destination. The last translation is the important translation because it is used to create the NAT translation for all devices that reach a spoke through the last NAT device.
- **Hub or spoke can be reached through pre-NAT addresses** --It is possible for two or more spokes to be behind the same NAT device, which can be reached through a pre-NAT IP address. Only the post-

NAT IP address is relied on even if it means that a tunnel may take a less desirable path. If both spokes use NAT through the same device, then a packet may not travel inside-out or outside-in as expected by the NAT device and translations may not occur correctly.

- **Interoperability between NAT and non-NAT capable devices** --In networks that are deployed with DMVPN, it is important that a device with NHRP NAT functionality operate together with non-NAT supported devices. A capability bit in the NHRP packet header indicates to any receiver whether a sending device understands a NAT extension.
- **Same NAT translation** --A spoke's post-NAT IP address must be the same when the spoke is communicating with its hubs and when it is communicating with other spokes. For example, a spoke must have the same post-NAT IP address no matter where it is sending tunnel packets within the DMVPN network.
- If one spoke is behind one NAT device and another different spoke is behind another NAT device, and Peer Address Translation (PAT) is the type of NAT used on both NAT devices, then a session initiated between the two spokes cannot be established.

One example of a PAT configuration on a NAT interface is:

```
ip nat inside source list nat_acl interface FastEthernet0/1 overload
```

Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

The following sections describe how DMVPN: Dynamic Tunnels Between Spokes Behind a NAT Device allows spoke-to-spoke tunnels to be built even if one or both spoke devices are behind a NAT device:

- [DMVPN Spoke-to-spoke Tunneling Limited to Spokes not Behind a NAT Device, page 74](#)
- [NHRP Spoke-to-Spoke Tunnel with a NAT Device, page 76](#)

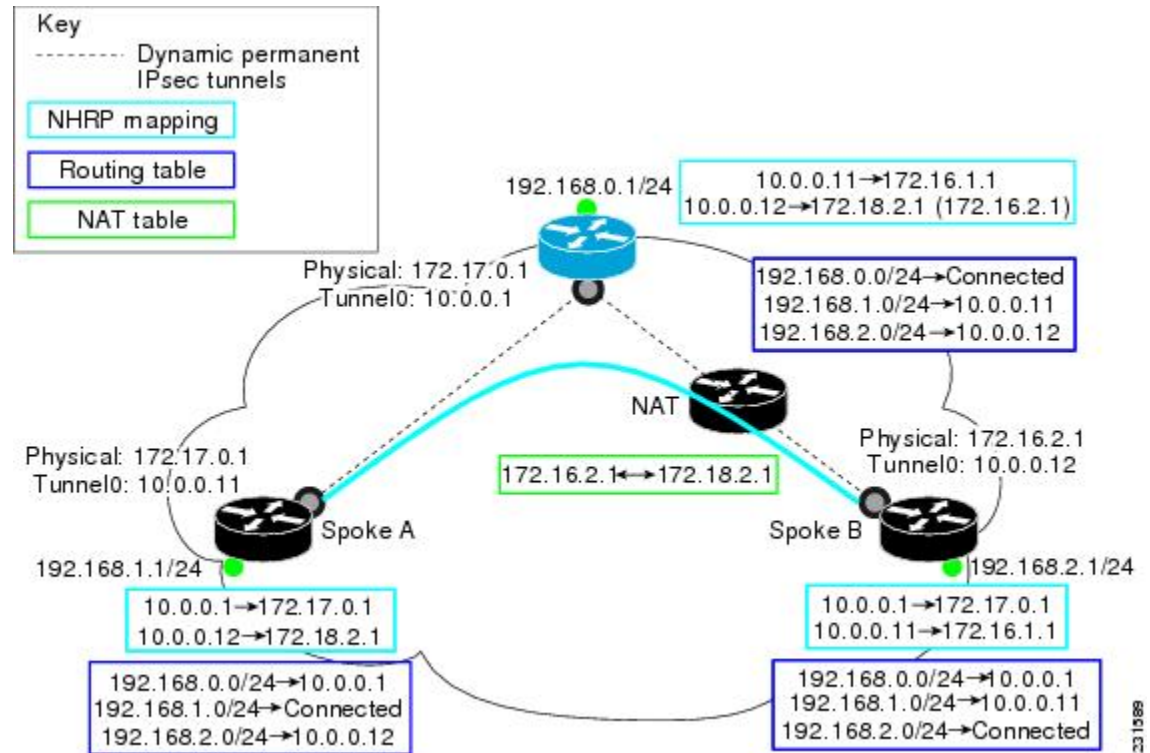
DMVPN Spoke-to-spoke Tunneling Limited to Spokes not Behind a NAT Device

NAT allows a single device, such as a router, to act as agent between the Internet (or “public network”) and a local (or “private”) network, and is often used because of the scarcity of available IP addresses. A single unique IP address is required to represent an entire group of devices to anything outside the NAT device. NAT is also deployed for security and administration purposes.

In DMVPN networks, spoke-to-spoke tunneling is limited to spokes that are not behind the NAT device. If one or both spokes are behind a NAT device, a spoke-to-spoke tunnel cannot be built to or from the NAT device because it is possible for the spoke-to-spoke tunnel traffic to fail or be lost “black-holed” for an extended period of time.

The diagram below and the following sections describe how DMVPN works when spoke-to-spoke tunneling is limited to spokes that are not behind a NAT device.

Figure 5 Implementation of DMVPN Spoke-to-spoke Tunneling Limited to Spokes Not Behind a NAT Device



- [NHRP Registration, page 75](#)
- [NHRP Resolution, page 76](#)

NHRP Registration

When an NHRP registration is received, the hub checks the source IP address on the encapsulating GRE/IP header of the NHRP packet with the source NBMA IP address, which is contained in the NHRP registration packet. If these IP addresses are different, then NHRP knows that NAT is changing the outer IP header source address. The hub preserves both the pre- and post-NAT address of the registered spoke.



Note

If encryption is used, then IPsec transport mode must be used to enable NHRP.

The following **show ip nhrp** command output example shows the source IP address of the NHRP packet and tunnel information for Spoke B in the figure above:



Note

The NBMA (post-NAT) address for Spoke B is 172.18.2.1 (the claimed NBMA (pre-NAT) source address is 172.16.2.1).

```
Router# show ip nhrp
```

```
10.0.0.11/32 via 10.0.0.11, Tunnel0 created 00:00:21, expire 00:05:38
Type: dynamic, Flags: authoritative unique registered used
NBMA address: 172.18.2.1
(Claimed NBMA address: 172.16.2.1)
```

NHRP Resolution

The following describes the NHRP resolution process between Spoke A and Spoke B shown in the figure above, where Spoke B is behind a NAT device with pre-NAT address of 172.16.2.1 and a post-NAT address of 172.18.2.1:

- The NHRP table entry for Spoke B on the hub contains both the post-NAT and pre-NAT addresses. When the hub receives an NHRP resolution request for the VPN address (tunnel address) of Spoke B, it answers with its own NBMA address instead of Spoke B's NBMA address.
- When the hub receives an NHRP resolution request sourced from Spoke B for any other spoke, the hub also answers with its own NBMA address. This ensures that any attempt to build a spoke-to-spoke tunnel with Spoke B results in the data packets being sent through the hub rather than through a spoke-to-spoke tunnel.

For example:

- Data traffic from source IP address 192.168.1.1 (behind Spoke A) to destination IP address 192.168.2.1 (behind Spoke B) triggers Spoke A to send a resolution request for Spoke B (10.0.0.12) to the next hop router (hub).
- The hub receives the resolution request and finds a mapping entry for Spoke B (10.0.0.12). Because Spoke B is behind a NAT device, it acts as a proxy and replies with its own NBMA address (172.17.0.1).
- The hub also receives a resolution request from Spoke B for Spoke A (10.0.0.11). Because Spoke B is behind a NAT device, it acts as a proxy and replies with its own NBMA address (172.17.0.1). This restricts any spoke-to-spoke traffic to or from Spoke B to travel through the hub router, which is done rather than having a tunnel between the spokes.

NHRP Spoke-to-Spoke Tunnel with a NAT Device

The NHRP Spoke-to-Spoke Tunnel with NAT introduces NAT extension in the NHRP protocol and is enabled automatically. The NHRP NAT extension is a Client Information Entry (CIE) entry with information about the protocol and post-NAT NBMA address. This additional information allows the support of spoke-to-spoke tunnels between spokes where one or both are behind a NAT device without the problem of losing (black-holing) traffic for an extended period of time.

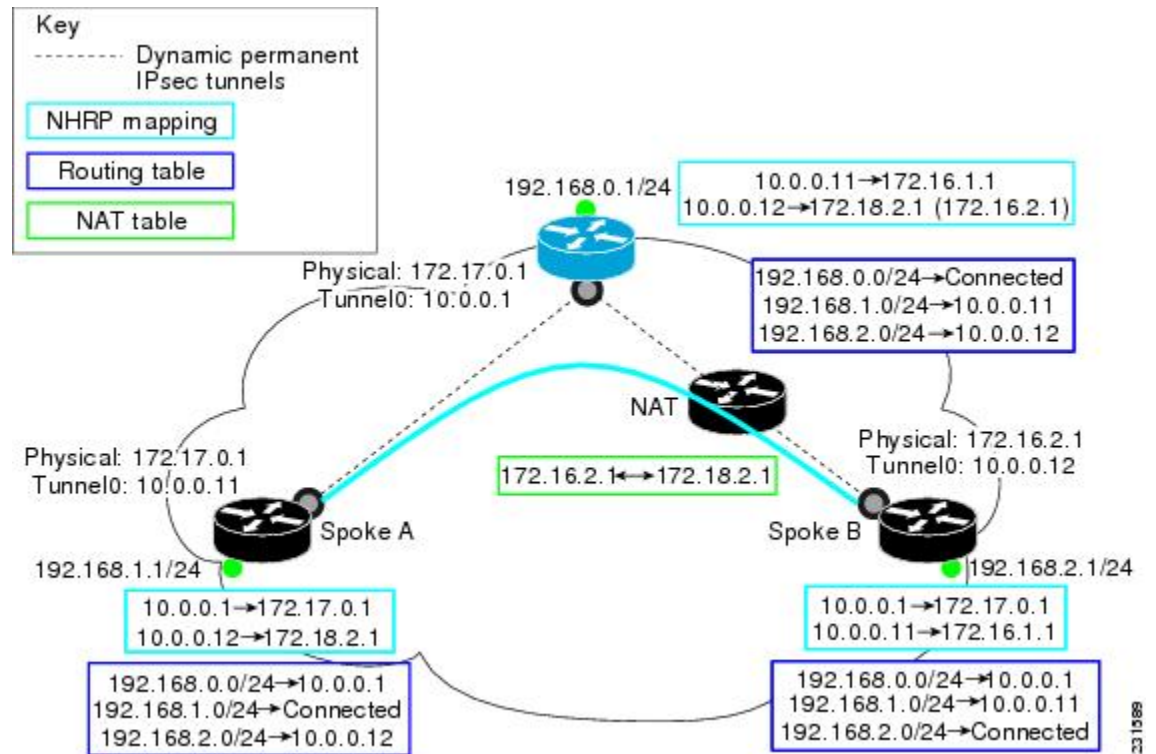


Note

The spoke-to-spoke tunnel may fail to come up, but it is detected and the data traffic flows through the hub, rather than being lost (black-holed).

the diagram below shows how the NHRP spoke-to-spoke tunnel works with NAT.

Figure 6 NHRP Between Spoke-to-Spoke Tunnels



- [NHRP Registration Process, page 77](#)
- [NHRP Resolution and Purge Process, page 77](#)

NHRP Registration Process

The following steps describe the NHRP registration process:

- 1 A spoke sends a registration request with the NAT-Capability=1 parameter and a NAT NHRP extension of the NBMA address of the hub as configured on the spoke.
- 2 The hub compares the NHRP (NAT) extension with its configured NBMA address and determines whether it itself is or is not behind a NAT device. The hub also makes a note of whether the spoke is behind a NAT device by comparing the incoming GRE/IP source address with the spoke's NBMA address in the NHRP packet.
- 3 The registration reply from the hub to the spoke includes a NAT NHRP extension with the post-NAT address of the spoke, if the hub detects if it is behind a NAT device.
- 4 If the spokes get a NAT NHRP extension in the NHRP registration reply it then records its post-NAT IP address for possible use later.

NHRP Resolution and Purge Process

The following steps describe the NHRP resolution and purge process:

- 1 When a spoke is behind a NAT device, it includes a NAT NHRP extension when it sends NHRP resolution requests.
- 2 The hub receives the resolution request. If the spoke is behind a NAT device and there is no NAT extension, then the hub adds a NAT extension before forwarding this extension to the next node (spoke or next hop server) along the path. However, if the hub is forwarding the request to a non-NAT extension capable node, it rewrites the source-NBMA inside the packet to be the post-NAT IP address for the requesting spoke rather than its pre-NAT IP address.
- 3 The receiver (spoke) uses a NAT NHRP extension record (NAT capable) or the source NBMA address (non-NAT capable information) to build the tunnel. This spoke's reply includes its own NAT extension if it is behind a NAT device.

**Note**

Hubs do not answer NHRP resolution requests on behalf of spokes. Hubs always forward NHRP resolution requests to the end spoke that has the requested tunnel IP address or services the requested data from the host IP address.

The following describes the NHRP resolution process between Spoke A and Spoke B shown in the figure above, where Spoke B is behind a NAT device with pre-NAT address 172.16.2.1 and post-NAT address of 172.18.2.1:

- Data traffic to the 192.168.2.0/24 network from hosts behind Spoke A triggers an NHRP resolution request for Spoke B's tunnel IP address (10.0.0.12) to be sent through the hub. The hub receives a resolution request and forwards it to Spoke B. Spoke B creates a dynamic spoke-to-spoke tunnel using the source NBMA IP address for Spoke A from the NHRP resolution request and sends an NHRP resolution reply directly to Spoke A. It includes its post-NAT address in the NAT NHRP-extension header.
- Alternatively, traffic to the 192.168.1.0/24 network from hosts behind the NAT device on Spoke B triggers an NHRP resolution request for Spoke A's tunnel IP address (10.0.0.11). Spoke B adds its own post-NAT IP address in the NHRP NAT-extension in the resolution request. The hub receives a resolution request and forwards it to Spoke A. Spoke A parses the NHRP NAT-extension and builds a tunnel using Spoke B's post-NAT address and replies directly to Spoke B.

Additional References

Related Documents

Related Topic	Document Title
NHRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS IP Addressing Services Command Reference</i>
Dynamic multipoint VPN	Dynamic Multipoint VPN (DMVPN)

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature. 	<p>To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFC	Title
No new or modified RFCs are supported by this release.	--

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>	<p>http://www.cisco.com/techsupport</p>

Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software

release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 4 *Feature Information for DMVPN: Dynamic Tunnels Between Spokes Behind a NAT Device*

Feature Name	Releases	Feature Information
DMVPN: Dynamic Tunnels Between Spokes Behind a NAT Device	12.4(15)T	The DMVPN: Dynamic Tunnels Between Spokes Behind a NAT Device feature allows NHRP spoke-to-spoke tunnels to be built in DMVPN networks, even if one or more spokes is behind a Network Address Translation (NAT) device.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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



Sharing IPsec with Tunnel Protection

The Sharing IPsec with Tunnel Protection feature allows an IP Security (IPsec) security association database (SADB) to be shared between two or more Generic Routing Encapsulation (GRE) tunnel interfaces, when tunnel protection is used. When these tunnel interfaces are shared, they have a single underlying cryptographic SADB, cryptographic map and IPsec profile in the Dynamic Multipoint Virtual Private Network (DMVPN) configuration.

The Sharing IPsec with Tunnel Protection feature is required by some DMVPN configurations. If IPsec SA sessions are not shared within the same IPsec SADB, then an IPsec security association (SA) may get associated with the wrong IPsec SADB and therefore the wrong tunnel interface, causing duplication of IPsec SAs and tunnel interfaces to flap. If the tunnel interfaces flap, then network connectivity problems occur.

- [Finding Feature Information, page 81](#)
- [Restrictions for Sharing IPsec with Tunnel Protection, page 81](#)
- [Information About Sharing IPsec with Tunnel Protection, page 82](#)
- [How to Share an IPsec Session Between Multiple Tunnels, page 83](#)
- [Configuration Examples for Sharing IPsec with Tunnel Protection, page 84](#)
- [Additional References, page 93](#)
- [Feature Information for Sharing IPsec with Tunnel Protection, page 95](#)
- [Glossary, page 95](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

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

Restrictions for Sharing IPsec with Tunnel Protection

Consider the following restrictions when sharing IPsec with tunnel protection:

- The **tunnel source** command on all tunnel interfaces using the same tunnel source *must* be configured using the interface type and number, not its IP address.
- All tunnels with the same **tunnel source** interface must use the same IPsec profile and the **shared** keyword with the **tunnel protection** command on all such tunnels. The only exception is a scenario

when there are only peer-to-peer (P2P) GRE tunnel interfaces configured with the same tunnel source in the system all with unique tunnel destination IP addresses.

- Different IPsec profile names must be used for shared and unshared tunnels.

For example, if “tunnel 1” is configured with the **tunnel source loopback0** command, and “tunnel 2” and “tunnel 3” are shared using the **tunnel source loopback1** command, then use **IPsec_profile_1** for tunnel 1 and **IPsec_profile_2** for tunnels 2 and 3.

- A different IPsec profile must be used for each set of shared tunnels.

For example, if tunnels 1 through 5 use **loopback0** as their tunnel source and tunnels 6-10 use **loopback1**, then define **IPsec_profile_1** for tunnels 1 through 5 and **ipsec_profile_2** for tunnels 6 to 10.

- Sometimes it may be desirable to not share an IPsec session between two or more tunnel interfaces using the same tunnel source.

For example, in a service provider environment, each DMVPN cloud can represent a different customer. It is desirable to lock the connections from a customer to a tunnel interface and not share or allow IPsec sessions from other customers. For such scenarios, Internet Security Association and Key Management Protocol (ISAKMP) profiles can be used to identify and bind customer connections to an ISAKMP profile and through that to an IPsec profile. This ISAKMP profile limits the IPsec profile to accept only those connections that matched the corresponding ISAKMP profile. Separate ISAKMP and IPsec profiles can be obtained for each DMVPN cloud (tunnel interface) without sharing the same IPsec SADB.

- Sharing IPsec is not desired and not supported for a Virtual Tunnel Interface (VTI). A VTI provides a routable interface type for terminating IPsec tunnels and a way to define protection between sites to form an overlay network.

Information About Sharing IPsec with Tunnel Protection

The following section describes how the Sharing IPsec with Tunnel Protection feature allows an IP Security (IPsec) security association database (SADB) to be shared between two or more Generic Routing Encapsulation (GRE) tunnel interfaces:

- [Using a Single IPsec SA to Secure GRE Tunnel Sessions, page 82](#)

Using a Single IPsec SA to Secure GRE Tunnel Sessions

In a dual-hub dual-DMVPN topology, it is possible to have two or more GRE tunnel sessions (same tunnel source and destination, but different tunnel keys) between the same two endpoints. In this case, it is desirable to use a single IPsec SA to secure both GRE tunnel sessions. Also, it is not possible to decide under which tunnel interface an IPsec Quick Mode (QM) request must be processed and bound when two tunnel interfaces use the same tunnel source.

The **tunnel protection IPsec profile shared** command is used to create a single IPsec SADB for all the tunnel interfaces that use the same profile and tunnel source interface. This allows a single IPsec SA to be used for all GRE tunnels (same tunnel source and destination, but different tunnel keys) between the same two endpoints. It also makes IPsec QM processing unambiguous since there is one SADB under which to process the incoming IPsec QM request for all shared tunnel interfaces as opposed to multiple SADBs, one for each tunnel interface when not shared.

The SA of a QM proposal to a tunnel interface is processed by using the shared SADB and crypto map parameters. On the crypto-data plane, the decrypted and GRE decapsulated packets are de-multiplexed to

the appropriate tunnel interface by the GRE module using a local address, remote address, and optional tunnel key information.

**Note**

The tunnel source, tunnel destination, and tunnel key (triplet) must be unique for all tunnel interfaces on a router. For a multipoint GRE interfaces where tunnel destination is not configured, the pair (tunnel source and tunnel key) must be unique. Incoming GRE packets are also matched to P2P GRE tunnels first; if there is not a match, then they are matched to mGRE tunnels.

How to Share an IPsec Session Between Multiple Tunnels

- [Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN, page 83](#)
- [What to Do Next, page 84](#)

Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

Use the following commands to configure a Cisco IOS router to share IPsec SADB between multiple tunnel interfaces in a DMVPN:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel *number***
4. **tunnel source {*ip-address* | *interface-type number* }**
5. **tunnel protection IPsec profile *profile-name* shared**
6. **exit**
7. **exit**

DETAILED STEPS

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

Command or Action	Purpose
<p>Step 3 <code>interface tunnel <i>number</i></code></p> <p>Example:</p> <pre>Router(config)# interface tunnel 5</pre>	<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.
<p>Step 4 <code>tunnel source {<i>ip-address</i> <i>interface-type number</i> }</code></p> <p>Example:</p> <pre>Router(config-if)# tunnel source Ethernet0</pre>	<p>Sets source IP address or source interface type number for a tunnel interface.</p> <ul style="list-style-type: none"> When using the tunnel protection IPsec profile shared command, the tunnel source must specify an interface, not an IP address.
<p>Step 5 <code>tunnel protection IPsec profile <i>profile-name</i> shared</code></p> <p>Example:</p> <pre>Router(config-if)# tunnel protection IPsec profile vpnprof shared</pre>	<p>Associates a tunnel interface with an IPsec profile.</p> <ul style="list-style-type: none"> The <i>profile-name</i> argument specifies the name of the IPsec profile; this value must match the <i>profile-name</i> specified in the crypto IPsec profile <i>profile-name</i> command. The shared keyword allows IPsec sessions to be shared between multiple tunnel interfaces configured with the same tunnel source IP.
<p>Step 6 <code>exit</code></p> <p>Example:</p> <pre>Router(config-if)# exit</pre>	<p>Exits the tunnel interface.</p>
<p>Step 7 <code>exit</code></p>	<p>Exits global configuration mode.</p>

What to Do Next

If your configuration requires more spoke routers in a dual-hub, dual DMVPN topology, repeat the steps in [How to Share an IPsec Session Between Multiple Tunnels, page 83](#) to configure additional spokes.

Configuration Examples for Sharing IPsec with Tunnel Protection

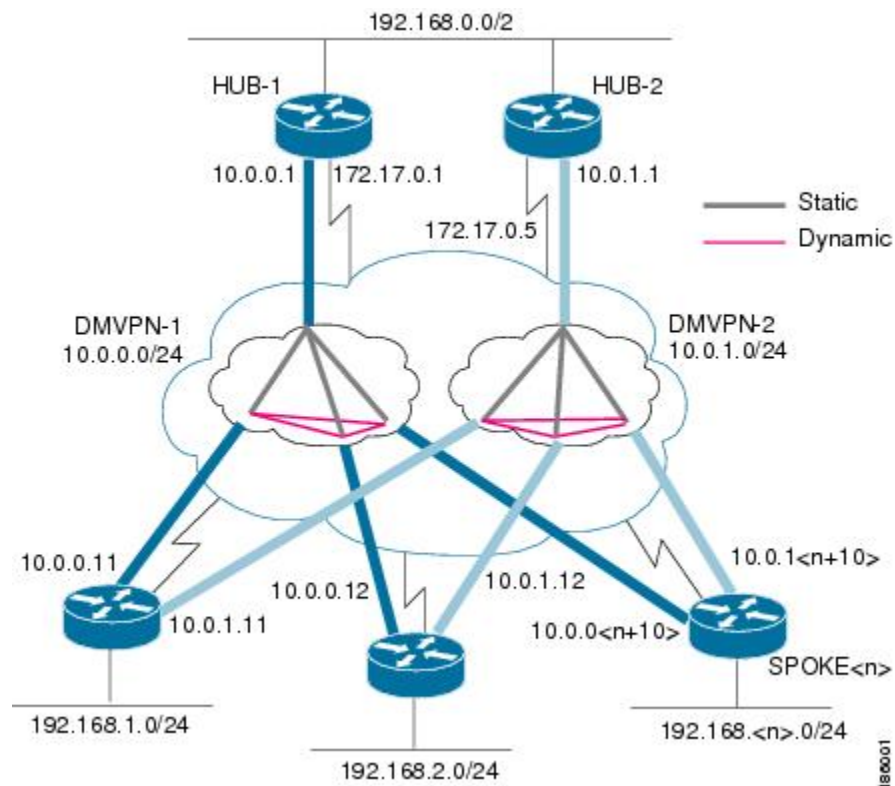
- [Dual-hub Router and Dual-DMVPN Topology, page 85](#)
- [Hub 1 Configuration Example, page 85](#)
- [Hub 2 Configuration Example, page 86](#)
- [Spoke 1 Configuration Example, page 87](#)
- [Spoke 2 Configuration Example, page 88](#)

Dual-hub Router and Dual-DMVPN Topology

The dual-hub router, dual DMVPN topology, shown in the figure below, has the following attributes:

- Each hub router is configured with a single mGRE tunnel interface.
- Each hub router is connected to one DMVPN subnet (blue cloud), and the spokes are connected to both DMVPN 1 and DMVPN 2.
- Each spoke router is configured with two mGRE tunnel interfaces.
- One mGRE tunnel interface belongs to DMVPN 1 and the other mGRE tunnel interface belongs to DMVPN 2.
- Each mGRE tunnel interface is configured with a same tunnel source IP address and uses shared tunnel protection between them.

Figure 7 Dual-hub Router, Dual-DMVPN Topology.



Hub 1 Configuration Example

Hub 1 and Hub 2 configurations are similar, except that each hub belongs to a different DMVPN.

Hub 1 has the following DMVPN configuration:

- IP subnet: 10.0.0.0/24
- Next Hop Address Resolution Protocol (NHRP) network ID: 100000
- Tunnel key: 100000

- Dynamic routing protocol: Enhanced Interior Gateway Routing Protocol (EIGRP)

```

!
hostname Hub1
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 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
 ip mtu 1400
 no ip next-hop-self eigrp 1
 ip nhrp authentication test
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp holdtime 600
no ip split-horizon eigrp 1
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection IPsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.252
!
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
 no auto-summary
!

```

Hub 2 Configuration Example

Hub 2 has the following DMVPN configuration:

- IP subnet: 10.0.1.0/24
- NHRP network ID: 100001
- Tunnel key: 100001
- Dynamic routing protocol: EIGRP

```

!
hostname Hub2
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 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 10.0.1.1 255.255.255.0
ip mtu 1400
no ip next-hop-self eigrp 1
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 100001
ip nhrp holdtime 600
no ip split-horizon eigrp 1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection IPsec profile vpnprof
!
interface Ethernet0
ip address 172.17.0.5 255.255.255.252
!
interface Ethernet1
ip address 192.168.0.2 255.255.255.0
!
router eigrp 1
network 10.0.1.0 0.0.0.255
network 192.168.0.0 0.0.0.255
no auto-summary
!

```

Spoke 1 Configuration Example

Spoke 1 has the following DMVPN configuration:

```

!
hostname Spoke1
!
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 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 10.0.0.11 255.255.255.0
ip mtu 1400
ip nhrp authentication test
ip nhrp map 10.0.0.1 172.17.0.1
ip nhrp map multicast 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 source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof shared
!
interface Tunnel1
bandwidth 1000
ip address 10.0.1.11 255.255.255.0
ip mtu 1400
ip nhrp authentication test
ip nhrp map 10.0.1.1 172.17.0.5
ip nhrp map multicast 172.17.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
ip nhrp nhs 10.0.1.1

```

```

ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection IPsec profile vpnprof shared
!
interface Ethernet0
  ip address dhcp hostname Spoke1
!
interface Ethernet1
  ip address 192.168.1.1 255.255.255.0
!
router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 10.0.1.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
!

```

Spoke 2 Configuration Example

Spoke 2 has the following DMVPN configuration:

```

!
hostname Spoke2
!
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 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 10.0.0.12 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.0.1 172.17.0.1
ip nhrp map multicast 172.170.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
ip nhrp nhs 10.0.0.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof shared
!
interface Tunnell
  bandwidth 1000
  ip address 10.0.1.12 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.1.1 172.17.0.5
ip nhrp map multicast 172.17.05
  ip nhrp network-id 100001
  ip nhrp holdtime 300
ip nhrp nhs 10.0.1.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection IPsec profile vpnprof shared
!

```

```

interface Ethernet0
 ip address dhcp hostname Spoke2
!
interface Ethernet1
 ip address 192.168.2.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 10.0.1.0 0.0.0.255
 network 192.168.2.0 0.0.0.255
 no auto-summary
!

```

- [Results on Spoke 1, page 89](#)

Results on Spoke 1

Spoke 1 has the following results for its DMVPN configuration:

```

Spoke1# show ip nhrp
10.0.0.1/32 via 10.0.0.1, Tunnel0 created 00:06:52, never expire
  Type: static, Flags: used
  NBMA address: 172.17.0.1
10.0.0.12/32 via 10.0.0.12, Tunnel0 created 00:03:17, expire 00:01:52
  Type: dynamic, Flags: router
  NBMA address: 172.17.0.12
10.0.1.1/32 via 10.0.1.1, Tunnel1 created 00:13:45, never expire
  Type: static, Flags: used
  NBMA address: 172.17.0.5
10.0.1.12/32 via 10.0.1.12, Tunnel1 created 00:00:02, expire 00:04:57
  Type: dynamic, Flags: router
  NBMA address: 172.17.0.12
Spoke1# show crypto socket

```



Note

There are only 3 crypto connections because the two NHRP sessions (10.0.0.12, Tunnel0) and (10.0.1.12, Tunnel1) are only one IPsec session, since they both have the same nonbroadcast multiaccess (NBMA) IPsec peer address.

```

Number of Crypto Socket connections 3
  Shd Peers (local/remote): 172.17.0.11
/172.17.0.12
  Local Ident (addr/mask/port/prot): (172.17.0.11/255.255.255.255/0/47)
  Remote Ident (addr/mask/port/prot): (172.17.0.12/255.255.255.255/0/47)
  Flags: shared
  IPsec Profile: "vpnprof"
  Socket State: Open
  Client: "TUNNEL SEC" (Client State: Active)
  Shd Peers (local/remote): 172.17.0.11
/172.17.0.5
  Local Ident (addr/mask/port/prot): (172.17.0.11/255.255.255.255/0/47)
  Remote Ident (addr/mask/port/prot): (172.17.0.5/255.255.255.255/0/47)
  Flags: shared
  IPsec Profile: "vpnprof"
  Socket State: Open
  Client: "TUNNEL SEC" (Client State: Active)
  Shd Peers (local/remote): 172.17.0.11
/172.17.0.1
  Local Ident (addr/mask/port/prot): (172.17.0.11/255.255.255.255/0/47)
  Remote Ident (addr/mask/port/prot): (172.17.0.1/255.255.255.255/0/47)
  Flags: shared
  IPsec Profile: "vpnprof"
  Socket State: Open
  Client: "TUNNEL SEC" (Client State: Active)
Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "vpnprof" Map-name: "vpnprof-head-1"
Spoke1# show crypto map

```

```

Crypto Map: "vpnprof-head-1" idb: Ethernet0/0 local address: 172.17.0.11
Crypto Map "vpnprof-head-1" 65536 IPsec-isakmp
  Profile name: vpnprof
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    trans2,
  }
Crypto Map "vpnprof-head-1" 65537 IPsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.17.0.5
  Extended IP access list
    access-list permit gre host 172.17.0.11 host 172.17.0.5
  Current peer: 172.17.0.5
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    trans2,
  }
Crypto Map "vpnprof-head-1" 65538 IPsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.17.0.1
  Extended IP access list
    access-list permit gre host 172.17.0.11 host 172.17.0.1
  Current peer: 172.17.0.1
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    trans2,
  }
Crypto Map "vpnprof-head-1" 65539 IPsec-isakmp
  Map is a PROFILE INSTANCE.
  Peer = 172.17.0.12
  Extended IP access list
    access-list permit gre host 172.17.0.11 host 172.17.0.12
  Current peer: 172.17.0.12
  Security association lifetime: 4608000 kilobytes/3600 seconds
  PFS (Y/N): N
  Transform sets={
    trans2,
  }
  Interfaces using crypto map vpnprof-head-1:
    Tunnel1
    Tunnel0

```

**Note**

All three crypto sessions are shown under each tunnel interface (3 entries, twice) in the **show crypto IPsec sa** output, because both interfaces are mapped to the same IPsec SADB, which has 3 entries. This duplication of output is expected in this case.

```

Spoke1# show crypto IPsec sa
interface: Tunnel0
  Crypto map tag: vpnprof-head-1, local addr 172.17.0.11
  protected vrf: (none)
  local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
  remote ident (addr/mask/prot/port): (172.17.0.1/255.255.255.255/47/0)
  current_peer 172.17.0.1 port 500
    PERMIT, flags={origin_is_acl,}
  #pkts encaps: 134, #pkts encrypt: 134, #pkts digest: 134
  #pkts decaps: 118, #pkts decrypt: 118, #pkts verify: 118
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 22, #recv errors 0
  local crypto endpt.: 172.17.0.11, remote crypto endpt.: 172.17.0.1
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0xA75421B1(2807308721)
  inbound esp sas:
    spi: 0x96185188(2518176136)
      transform: esp-des esp-md5-hmac ,

```



```

    in use settings ={Transport, }
    conn id: 3, flow_id: SW:3, crypto map: vpnprof-head-1
    sa timing: remaining key lifetime (k/sec): (4569747/3242)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
  spi: 0xA75421B1(2807308721)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 4, flow_id: SW:4, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4569745/3242)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
outbound ah sas:
outbound pcp sas:
protected vrf: (none)
local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.17.0.5/255.255.255.255/47/0)
current_peer 172.17.0.5 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 244, #pkts encrypt: 244, #pkts digest: 244
  #pkts decaps: 253, #pkts decrypt: 253, #pkts verify: 253
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 1, #recv errors 0
  local crypto endpt.: 172.17.0.11, remote crypto endpt.: 172.17.0.5
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0x3C50B3AB(1011921835)
inbound esp sas:
  spi: 0x3EBE84EF(1052673263)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 1, flow_id: SW:1, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4549326/2779)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
  spi: 0x3C50B3AB(1011921835)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 2, flow_id: SW:2, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4549327/2779)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
outbound ah sas:
outbound pcp sas:
protected vrf: (none)
local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.17.0.12/255.255.255.255/47/0)
current_peer 172.17.0.12 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
  #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 2
  #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.: 172.17.0.11, remote crypto endpt.: 172.17.0.12
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0x38C04B36(952126262)
inbound esp sas:
  spi: 0xA2EC557(170837335)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }

```

```

conn id: 5, flow_id: SW:5, crypto map: vpnprof-head-1
sa timing: remaining key lifetime (k/sec): (4515510/3395)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
spi: 0x38C04B36(952126262)
transform: esp-des esp-md5-hmac ,
in use settings ={Transport, }
conn id: 6, flow_id: SW:6, crypto map: vpnprof-head-1
sa timing: remaining key lifetime (k/sec): (4515511/3395)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE
outbound ah sas:
outbound pcp sas:
interface: Tunnell
Crypto map tag: vpnprof-head-1, local addr 172.17.0.11
protected vrf: (none)
local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.17.0.1/255.255.255.255/47/0)
current_peer 172.17.0.1 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 134, #pkts encrypt: 134, #pkts digest: 134
#pkts decaps: 118, #pkts decrypt: 118, #pkts verify: 118
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 22, #recv errors 0
local crypto endpt.: 172.17.0.11, remote crypto endpt.: 172.17.0.1
path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
current outbound spi: 0xA75421B1(2807308721)
inbound esp sas:
spi: 0x96185188(2518176136)
transform: esp-des esp-md5-hmac ,
in use settings ={Transport, }
conn id: 3, flow_id: SW:3, crypto map: vpnprof-head-1
sa timing: remaining key lifetime (k/sec): (4569747/3242)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
spi: 0xA75421B1(2807308721)
transform: esp-des esp-md5-hmac ,
in use settings ={Transport, }
conn id: 4, flow_id: SW:4, crypto map: vpnprof-head-1
sa timing: remaining key lifetime (k/sec): (4569745/3242)
IV size: 8 bytes
replay detection support: Y
Status: ACTIVE
outbound ah sas:
outbound pcp sas:
protected vrf: (none)
local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.17.0.5/255.255.255.255/47/0)
current_peer 172.17.0.5 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 244, #pkts encrypt: 244, #pkts digest: 244
#pkts decaps: 253, #pkts decrypt: 253, #pkts verify: 253
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 1, #recv errors 0
local crypto endpt.: 172.17.0.11, remote crypto endpt.: 172.17.0.5
path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
current outbound spi: 0x3C50B3AB(1011921835)
inbound esp sas:
spi: 0x3EBE84EF(1052673263)
transform: esp-des esp-md5-hmac ,

```

```

    in use settings ={Transport, }
    conn id: 1, flow_id: SW:1, crypto map: vpnprof-head-1
    sa timing: remaining key lifetime (k/sec): (4549326/2779)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
  spi: 0x3C50B3AB(1011921835)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 2, flow_id: SW:2, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4549327/2779)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
outbound ah sas:
outbound pcp sas:
protected vrf: (none)
local ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.17.0.12/255.255.255.255/47/0)
current_peer 172.17.0.12 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
  #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 2
  #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.: 172.17.0.11, remote crypto endpt.: 172.17.0.12
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0x38C04B36(952126262)
inbound esp sas:
  spi: 0xA2EC557(170837335)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 5, flow_id: SW:5, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4515510/3395)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
inbound ah sas:
inbound pcp sas:
outbound esp sas:
  spi: 0x38C04B36(952126262)
  transform: esp-des esp-md5-hmac ,
  in use settings ={Transport, }
  conn id: 6, flow_id: SW:6, crypto map: vpnprof-head-1
  sa timing: remaining key lifetime (k/sec): (4515511/3395)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
outbound ah sas:
outbound pcp sas:
Spoke1#

```

Additional References

The following sections provide references related to the Sharing IPsec with Tunnel Protection feature.

Related Documents

Related Topic	Document Title
Configuring a DMVPN	“Dynamic Multipoint VPN (DMVPN)” module in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
Implementing DMVPN with IPsec VPN solution	Dynamic Multipoint IPsec VPNs (Using Multipoint GRE/NHRP to Scale IPsec VPNs) white paper
Configuring basic IP Security (IPsec) virtual private networks (VPNs)	“Configuring Security for VPNs with IPsec” module in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>

Standards

Standard	Title
None	--

MIBs

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

RFCs

RFC	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2547	<i>BGP/MPLS VPNs</i>
RFC 2784	<i>Generic Routing Encapsulation (GRE)</i>

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>	<p>http://www.cisco.com/techsupport</p>

Feature Information for Sharing IPsec with Tunnel Protection

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

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

Table 5 Feature Information for Sharing IPsec with Tunnel Protection

Feature Name	Releases	Feature Information
Sharing IPsec with Tunnel Protection	12.4(15)T	<p>The Sharing IPsec with Tunnel Protection feature allows an Internet Protocol Security (IPsec) session to be shared between two or more Generic Routing Encapsulation (GRE) tunnel interfaces.</p> <p>In 12.4(15)T, this feature was introduced.</p> <p>The following command was introduced or modified by this feature: tunnel protection IPsec profile</p>

Glossary

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 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.

NHRP-- Next Hop Resolution Protocol. Protocol that routers, access servers, and hosts can use to discover the addresses of other routers and hosts connected to an NBMA network.

The Cisco implementation of NHRP supports the IETF draft version 11 of NBMA 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.

SA-- security association. Describes how two or more entities use 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-- List of operations performed on a data flow 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.

tunnel-- In the context of this module, a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.

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.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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

