



Dynamic Multipoint VPN Configuration Guide, Cisco IOS XE 17| Cisco Catalyst 8000 Edge Platforms

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.

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.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. 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. (1721R)

© 2021 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Read Me First 1

CHAPTER 2

Dynamic Multipoint VPN 3

Prerequisites for Dynamic Multipoint VPN	3
Restrictions for Dynamic Multipoint VPN	3
Information About Dynamic Multipoint VPN	5
Benefits of Dynamic Multipoint VPN	5
Feature Design of Dynamic Multipoint VPN	5
IPsec Profiles	6
Enabling Traffic Segmentation Within DMVPN	7
NAT-Transparency Aware DMVPN	8
Call Admission Control with DMVPN	9
NHRP Rate-Limiting Mechanism	9
How to Configure Dynamic Multipoint VPN	10
Configuring an IPsec Profile	10
Configuring the Hub for DMVPN	11
Configuring the Spoke for DMVPN	15
Configuring the Forwarding of Clear-Text Data IP Packets into a VRF	19
Configuring the Forwarding of Encrypted Tunnel Packets into a VRF	19
Configuring Traffic Segmentation Within DMVPN	20
Prerequisites	20
Enabling MPLS on the VPN Tunnel	21
Configuring Multiprotocol BGP on the Hub Router	22
Configuring Multiprotocol BGP on the Spoke Routers	24
Troubleshooting Dynamic Multipoint VPN	26
What to Do Next	30

Configuration Examples for Dynamic Multipoint VPN Feature	30
Example Hub Configuration for DMVPN	30
Example Spoke Configuration for DMVPN	31
Example 2547oDMVPN with BGP Only Traffic Segmentation	32
Example 2547oDMVPN with Enterprise Branch Traffic Segmentation	36
Additional References for Dynamic Multipoint VPN	44
Feature Information for Dynamic Multipoint VPN	44
Glossary	45

CHAPTER 3**IPv6 over DMVPN 47**

Prerequisites for IPv6 over DMVPN	47
Information About IPv6 over DMVPN	48
DMVPN for IPv6 Overview	48
NHRP Routing	48
IPv6 Routing	49
IPv6 Addressing and Restrictions	50
How to Configure IPv6 over DMVPN	50
Configuring an IPsec Profile in DMVPN for IPv6	50
Configuring the Hub for IPv6 over DMVPN	52
Configuring the NHRP Redirect and Shortcut Features on the Hub	55
Configuring the Spoke for IPv6 over DMVPN	57
Verifying DMVPN for IPv6 Configuration	60
Monitoring and Maintaining DMVPN for IPv6 Configuration and Operation	62
Configuration Examples for IPv6 over DMVPN	63
Example: Configuring an IPsec Profile	63
Example: Configuring the Hub for DMVPN	63
Example: Configuring the Spoke for DMVPN	65
Example: Configuring the NHRP Redirect and Shortcut Features on the Hub	66
Example: Configuring NHRP on the Hub and Spoke	66
Additional References	67
Feature Information for IPv6 over DMVPN	68

CHAPTER 4**DMVPN Configuration Using FQDN 71**

Finding Feature Information	71
-----------------------------	----

Prerequisites for DMVPN Configuration Using FQDN	72
Restrictions for DMVPN Configuration Using FQDN	72
Information About DMVPN Configuration Using FQDN	72
DNS Functionality	72
DNS Server Deployment Scenarios	72
How to Configure DMVPN Configuration Using FQDN	73
Configuring a DNS Server on a Spoke	73
Configuring a DNS Server	73
Configuring an FQDN with a Protocol Address	74
Configuring a FQDN Without an NHS Protocol Address	75
Verifying DMVPN FQDN Configuration	77
Configuration Examples for DMVPN Configuration Using FQDN	78
Example Configuring a Local DNS Server	78
Example Configuring an External DNS Server	78
Example Configuring NHS with a Protocol Address and an NBMA Address	79
Example Configuring NHS with a Protocol Address and an FQDN	79
Example Configuring NHS Without a Protocol Address and with an NBMA Address	79
Example Configuring NHS Without a Protocol Address and with an FQDN	79
Additional References	80
Feature Information for DMVPN Configuration Using FQDN	81

CHAPTER 5

DMVPN-Tunnel Health Monitoring and Recovery Backup NHS	83
Finding Feature Information	83
Information About DMVPN-Tunnel Health Monitoring and Recovery Backup NHS	84
NHS States	84
NHS Priorities	84
NHS Clusterless Model	84
NHS Clusters	85
NHS Fallback Time	86
NHS Recovery Process	87
Alternative Spoke to Hub NHS Tunnel	87
Returning to Preferred NHS Tunnel upon Recovery	88
How to Configure DMVPN-Tunnel Health Monitoring and Recovery Backup NHS	89
Configuring the Maximum Number of Connections for an NHS Cluster	89

- Configuring NHS Fallback Time 90
- Configuring NHS Priority and Group Values 91
- Verifying the DMVPN-Tunnel Health Monitoring and Recovery Backup NHS Feature 92
- Configuration Examples for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS 93
 - Example Configuring Maximum Connections for an NHS Cluster 93
 - Example Configuring NHS Fallback Time 94
 - Example Configuring NHS Priority and Group Values 94
- Additional References 94
- Feature Information for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS 95

CHAPTER 6

DMVPN Tunnel Health Monitoring and Recovery 97

- Finding Feature Information 97
- Prerequisites for DMVPN Tunnel Health Monitoring and Recovery 97
- Restrictions for DMVPN Tunnel Health Monitoring and Recovery 98
- Information About DMVPN Tunnel Health Monitoring and Recovery 98
 - NHRP Extension MIB 98
 - DMVPN Syslog Messages 99
 - Interface State Control 99
 - Interface State Control Configuration Workflow 100
- How to Configure DMVPN Tunnel Health Monitoring and Recovery 101
 - Configuring Interfaces to Generate SNMP NHRP Notifications 101
 - Troubleshooting Tips 102
 - Configuring Interface State Control on an Interface 102
- Configuration Examples for DMVPN Tunnel Health Monitoring and Recovery 103
 - Example: Configuring SNMP NHRP Notifications 103
 - Example: Configuring Interface State Control 103
- Additional References for DMVPN Tunnel Health Monitoring and Recovery 104
- Feature Information for DMVPN Tunnel Health Monitoring and Recovery 105

CHAPTER 7

DMVPN Event Tracing 107

- Finding Feature Information 107
- Information About DMVPN Event Tracing 107
 - Benefits of DMVPN Event Tracing 107
 - DMVPN Event Tracing Options 108

How to Configure DMVPN Event Tracing	108
Configuring DMVPN Event Tracing in Privileged EXEC Mode	108
Configuring DMVPN Event Tracing in Global Configuration Mode	109
Configuration Examples for DMVPN Event Tracing	110
Example Configuring DMVPN Event Tracing in Privileged EXEC Mode	110
Example Configuring DMVPN Event Tracing in Global Configuration Mode	110
Additional References	110
Feature Information for DMVPN Event Tracing	111

CHAPTER 8**NHRP MIB 113**

Prerequisites for NHRP MIB	113
Restrictions for NHRP MIB	113
Information About NHRP MIB	114
CISCO-NHRP-MIB	114
RFC-2677	114
How to Use NHRP MIB	114
Verifying NHRP MIB Status	114
Configuration Examples for NHRP MIB	115
Example Verifying NHRP MIB Status	115
Example VRF-Aware NHRP MIB Configuration	115
Additional References	117
Feature Information for NHRP MIB	118

CHAPTER 9**DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device 119**

Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	119
Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	120
Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device	120
DMVPN Spoke-to-Spoke Tunneling Limited to Spokes Not Behind a NAT Device	120
NHRP Registration	121
NHRP Resolution	122
NHRP Spoke-to-Spoke Tunnel with a NAT Device	122
NHRP Registration Process	123
NHRP Resolution and Purge Process	123
Additional References	124

CHAPTER 10	Sharing IPsec with Tunnel Protection	127
	Prerequisites for Sharing IPsec with Tunnel Protection	127
	Restrictions for Sharing IPsec with Tunnel Protection	127
	Information About Sharing IPsec with Tunnel Protection	128
	Single IPsec SAs and GRE Tunnel Sessions	128
	How to Configure Sharing IPsec with Tunnel Protection	129
	Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN	129
	Configuration Examples for Sharing IPsec with Tunnel Protection	131
	Example: Dual-Hub Router, Dual-DMVPN Topology	131
	Example: Configuring an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN	132
	Example: HUB-1 Configuration	132
	Example: HUB-2 Configuration	133
	Example: SPOKE 1 Configuration	133
	Example: SPOKE 2 Configuration	134
	Example: Results on SPOKE 1	136
	Additional References	141
	Feature Information for Sharing IPsec with Tunnel Protection	142
	Glossary	142

CHAPTER 11	Per-Tunnel QoS for DMVPN	145
	Finding Feature Information	145
	Prerequisites for Per-Tunnel QoS for DMVPN	145
	Restrictions for Per-Tunnel QoS for DMVPN	146
	Information About Per-Tunnel QoS for DMVPN	146
	Per-Tunnel QoS for DMVPN Overview	146
	Benefits of Per-Tunnel QoS for DMVPN	146
	NHRP QoS Provisioning for DMVPN	147
	Per-Tunnel QoS for Spoke to Spoke Connections	147
	How to Configure Per-Tunnel QoS for DMVPN	148
	Configuring an NHRP Group on a Spoke	148
	Configuring an NHRP Group Attribute on a Spoke	148
	Mapping an NHRP Group to a QoS Policy on the Hub	149
	Enabling DMVPN Per-tunnel QoS Sourced from Port Channel	150

Verifying Per-Tunnel QoS for DMVPN	151
Configuration Examples for Per-Tunnel QoS for DMVPN	152
Example: Configuring an NHRP Group on a Spoke	152
Example: Configuring an NHRP Group Attribute on a Spoke	153
Example: Mapping an NHRP Group to a QoS Policy on the Hub	154
Example: Enabling DMVPN Per-tunnel QoS Sourced from Port Channel	155
Example: Verifying Per-Tunnel QoS for DMVPN	156
Additional References for Per-Tunnel QoS for DMVPN	159
Feature Information for Per-Tunnel QoS for DMVPN	160

CHAPTER 12**Configuring TrustSec DMVPN Inline Tagging Support 163**

Prerequisites for Configuring TrustSec DMVPN Inline Tagging Support	163
Restrictions for Configuring TrustSec DMVPN Inline Tagging Support	163
Information About Configuring TrustSec DMVPN Inline Tagging Support	164
Cisco TrustSec	164
SGT and IPsec	165
SGT on the IKEv2 Initiator and Responder	165
Handling Fragmentation	166
How to Configure TrustSec DMVPN Inline Tagging Support	166
Enabling IPsec Inline Tagging	166
Monitoring and Verifying TrustSec DMVPN Inline Tagging Support	167
Enabling IPsec Inline Tagging on IKEv2 Networks	169
Configuration Examples for TrustSec DMVPN Inline Tagging Support	169
Example: Enabling IPsec Inline Tagging on IKEv2 Networks	169
Additional References for TrustSec DMVPN Inline Tagging Support	173
Feature Information for TrustSec DMVPN Inline Tagging Support	174

CHAPTER 13**Spoke-to-Spoke NHRP Summary Maps 175**

Finding Feature Information	175
Information About Spoke-to-Spoke NHRP Summary Maps	175
Spoke-to-Spoke NHRP Summary Maps	175
NHRP Summary Map Support for IPv6 Overlay	177
How to Configure Spoke-to-Spoke NHRP Summary Maps	177
Configuring Spoke-to-Spoke NHRP Summary Maps on Spoke	177

Verifying Spoke-to Spoke NHRP Summary Maps	179
Troubleshooting Spoke-to-Spoke NHRP Summary Maps	180
Configuration Examples for Spoke-to-Spoke NHRP Summary Maps	181
Example: Spoke-to-Spoke NHRP Summary Maps	181
Additional References for Spoke-to-Spoke NHRP Summary Maps	183
Feature Information for Spoke-to-Spoke NHRP Summary Maps	183

CHAPTER 14**BFD Support on DMVPN 185**

Finding Feature Information	185
Prerequisites for BFD Support on DMVPN	185
Restrictions for BFD Support on DMVPN	185
Information About BFD Support on DMVPN	186
BFD Operation	186
Benefits of BFD Support on DMVPN	186
How to Configure BFD Support on DMVPN	186
Configuring BFD Support on DMVPN	186
Example: BFD Support on DMVPN	187
Additional References for BFD Support on DMVPN	191
Feature Information for BFD Support on DMVPN	191

CHAPTER 15**DMVPN Support for IWAN 193**

Prerequisites for DMVPN Support for IWAN	193
Restrictions for DMVPN Support for IWAN	193
Information About DMVPN Support for IWAN	193
Transport Independence	193
DMVPN for IWAN	194
Secondary Paths	194
DMVPN Multiple Tunnel Termination	195
How to Configure DMVPN Support for IWAN	196
Configuring DMVPN Support for IWAN	196
Configuring DMVPN Multiple Tunnel Termination	197
Configuration Examples for DMVPN Support for IWAN	197
Example: DMVPN Support for IWAN	197
Troubleshooting	202

Additional References for DMVPN Support for IWAN 202

Feature Information for DMVPN Support for IWAN 203

CHAPTER 16

Configuring MPLS over DMVPN 205

Finding Feature Information 205

Prerequisites for Configuring MPLS over DMVPN 205

Information About MPLS over DMVPN 206

MPLS over DMVPN Networks 206

The Need for MPLS 206

Components of MPLS over Dynamic IPsec Tunnels Feature 207

Working of MPLS over Dynamic IPsec Tunnels Feature 208

Support for Spoke Nodes as P Nodes in MPLS over DMVPN Phase 3 209

Enhancements to BGP and NHRP 211

IVRF Support 212

How to Configure MPLS over DMVPN 212

Configuring MPLS over FlexVPN 212

Configuration Examples for MPLS over FlexVPN 213

Restrictions for Configuring 6VPE and 6PE Support in MPLS over DMVPN Phase 2 225

Configuring 6VPE Support in MPLS over DMVPN Phase 2 225

Enabling Components for the Hub 225

Configuring VRF for the Hub 226

Enabling Tunnel for the Hub 226

Enabling IPsec Tunnel Protection for the Hub 226

Enabling WAN Interfaces for the Hub 226

Enabling Transport Routing for the Hub 226

Enabling Overlay Routing for the Hub 226

Enabling the Components for the Spokes 227

Configuring VRF for the Spokes 227

Enabling Tunnel for the Spokes 228

Enabling IPsec Tunnel Protection for Spokes 228

Enabling WAN Facing Interfaces for Spokes 228

Enabling PE-CE Interfaces for Spokes 228

Enabling Transport Routing for Spokes 228

Enabling Overlay Routing for the Spokes 228

Enabling Transport Routing for IPv6	229
Enabling WAN Interfaces for IPv6	229
Enabling Tunnel for Hubs	229
Enabling Tunnel for Spokes	230
Configuring 6PE Support in MPLS over DMVPN Phase 2	230
Enabling Components for the Hub	230
Enabling Tunnel for Hub	231
Enabling IPsec Tunnel Protection for the Hub	231
Enabling WAN Facing Interfaces for Hub	231
Enabling Transport Routing for Hub	231
Enabling Overlay Routing for Hub	231
Enabling Components for the Spokes	232
Enabling Tunnel for Spokes	232
Enabling IPsec Tunnel Protection for Spokes	232
Enabling WAN Facing Interfaces for Spokes	232
Enabling PE-CE Interface for Spokes	232
Enabling Transport Routing for Spokes	233
Enabling Overlay Routing for Spokes	233
Verifying the 6VPE support in MPLS over DMVPN Phase 2 Configurations	233
Verifying the 6PE support in MPLS over DMVPN Phase 2 Configurations	233
Configure a Spoke Node as a P Node in MPLS over DMVPN Phase 3	234
Feature Information for MPLS over DMVPN	234

CHAPTER 17
DHCP Tunnels Support 237

Finding Feature Information	237
Restrictions for DHCP Tunnels Support	237
Information About DHCP Tunnels Support	238
DHCP Overview	238
DHCP Behavior on a Tunnel Network	238
DMVPN Hub as a DHCP Relay Agent	239
DMVPN Topologies	239
Dual-Hub Single-DMVPN Topology	239
Dual-Hub Dual-DMVPN Topology	239
Hierarchical DMVPN Topology	239

How to Configure DHCP Tunnels Support	239
Configuring the DHCP Relay Agent to Unicast DHCP Replies	239
Configuring a DMVPN Spoke to Clear the Broadcast Flag	240
Configuration Examples for DHCP Tunnels Support	241
Example Configuring a DHCP Relay Agent to Unicast DHCP Replies	241
Example Configuring a DMVPN Spoke to Clear the Broadcast Flag and Set the IP Address to DHCP	242
Additional References	242
Feature Information for DHCP Tunnels Support	243

APPENDIX A

Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)	245
Prerequisites Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)	245
Information About Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)	246
Per-Tunnel QoS and Multiple Policy Maps (MPOL)	246
Supported Configurations	246
Components in MPOL	246
Class Maps	246
Policy Maps	247
Per-Spoke Policy Maps	247
Traffic Shaping	247
How to Configure Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)	247
Setting Up MPOL Components	247
Configuring Policy Maps	247
Applying Policy Maps to Spoke	248
Applying Shaping	248
Enabling MPOL	248
Verifying MPOL Configuration	248
Additional References for Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)	250



CHAPTER 1

Read Me First

Important Information



Note For CUBE feature support information on Cisco IOS XE Bengaluru 17.6.1 and later releases, see [Cisco Unified Border Element IOS-XE Configuration Guide](#).



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Feature Information

Use [Cisco Feature Navigator](#) to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- [Cisco IOS Command References, All Releases](#)

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco Marketplace](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).



CHAPTER 2

Dynamic Multipoint VPN

The Dynamic Multipoint VPN 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).



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Prerequisites for Dynamic Multipoint VPN, on page 3](#)
- [Restrictions for Dynamic Multipoint VPN, on page 3](#)
- [Information About Dynamic Multipoint VPN, on page 5](#)
- [How to Configure Dynamic Multipoint VPN, on page 10](#)
- [Configuration Examples for Dynamic Multipoint VPN Feature, on page 30](#)
- [Additional References for Dynamic Multipoint VPN, on page 44](#)
- [Feature Information for Dynamic Multipoint VPN, on page 44](#)
- [Glossary, on page 45](#)

Prerequisites for Dynamic Multipoint VPN

- 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.
- To use the 2547oDMPVN--Traffic Segmentation Within DMVPN feature you must configure Multiprotocol Label Switching (MPLS) by using the **mpls ip** command.

Restrictions for Dynamic Multipoint VPN

- Bidirectional protocol-independent multicast (PIM) is not supported over DMVPN. Therefore, you must use PIM Sparse mode (ASM) over DMVPN.
- Protocol-independent multicast (PIM) dense mode is not supported over DMVPN on IOS-XE devices.

- If you use the 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 an 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.
- The device running Cisco IOS XE 16.12.4 or 17.3.1a image is unable to establish CDP neighborhood with peers through a mGRE tunnel.
- If one spoke is behind one Network Address Translation (NAT) device and a different spoke is behind another NAT device, and Port 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 FastEthernet0/0/1 overload
```

- When using OSPF point-to-multipoint, you must block the OSPF /32 routes. Add the following on all hub and spoke routers to block these host routes:

```
router ospf <#>
...
distribute-list prefix-list Block-32 out //block OSPF/32 connected routes//
ip prefix-list Block-32 deny <tunnel-subnet> <mask> ge 32
ip prefix-list Block-32 permit any le 32
```

SSO Restrictions

- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPsec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPsec sessions in such cases for the duration of the reload.
- This platform does not support stateful ISSU for IPsec sessions. Before performing an ISSU, you must explicitly terminate all existing IPsec sessions or tunnels prior to the operation and reestablish them post ISSU. Specifically, ensure that there are no half-open or half-established IPsec tunnels present before performing ISSU. To do this, we recommend a interface shutdown in the case of interfaces that may initiate a tunnel setup, such as a routing protocol initiating a tunnel setup, or interfaces that have keepalive enabled, or where there is an auto trigger for an IPsec session. Traffic disruption over the IPsec sessions during ISSU is obvious in this case.

Information About Dynamic Multipoint VPN

Benefits of Dynamic Multipoint VPN

Hub Router Configuration Reduction

- 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 the 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.

Feature Design of Dynamic Multipoint VPN

The Dynamic Multipoint VPN 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:

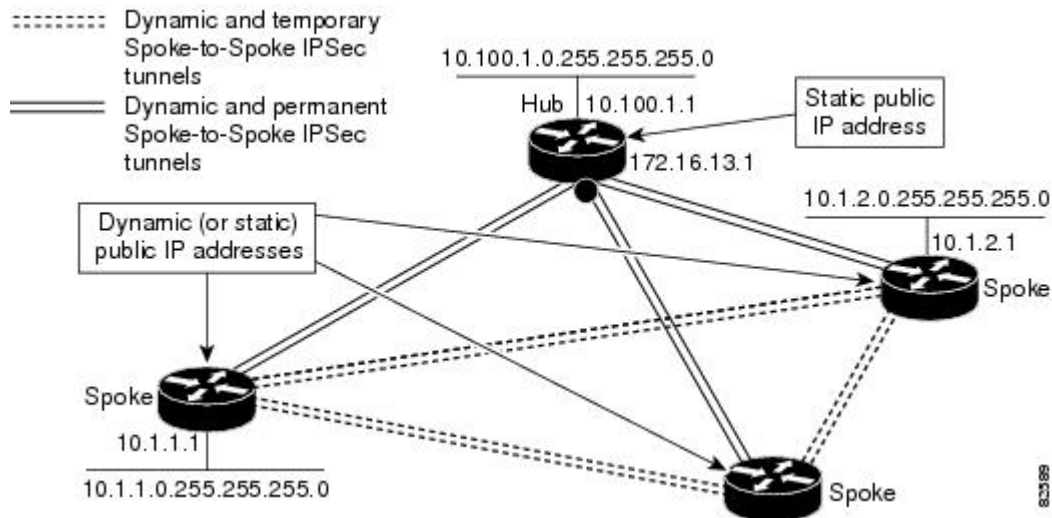
- NHRP--A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its

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 figure 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 need not configure

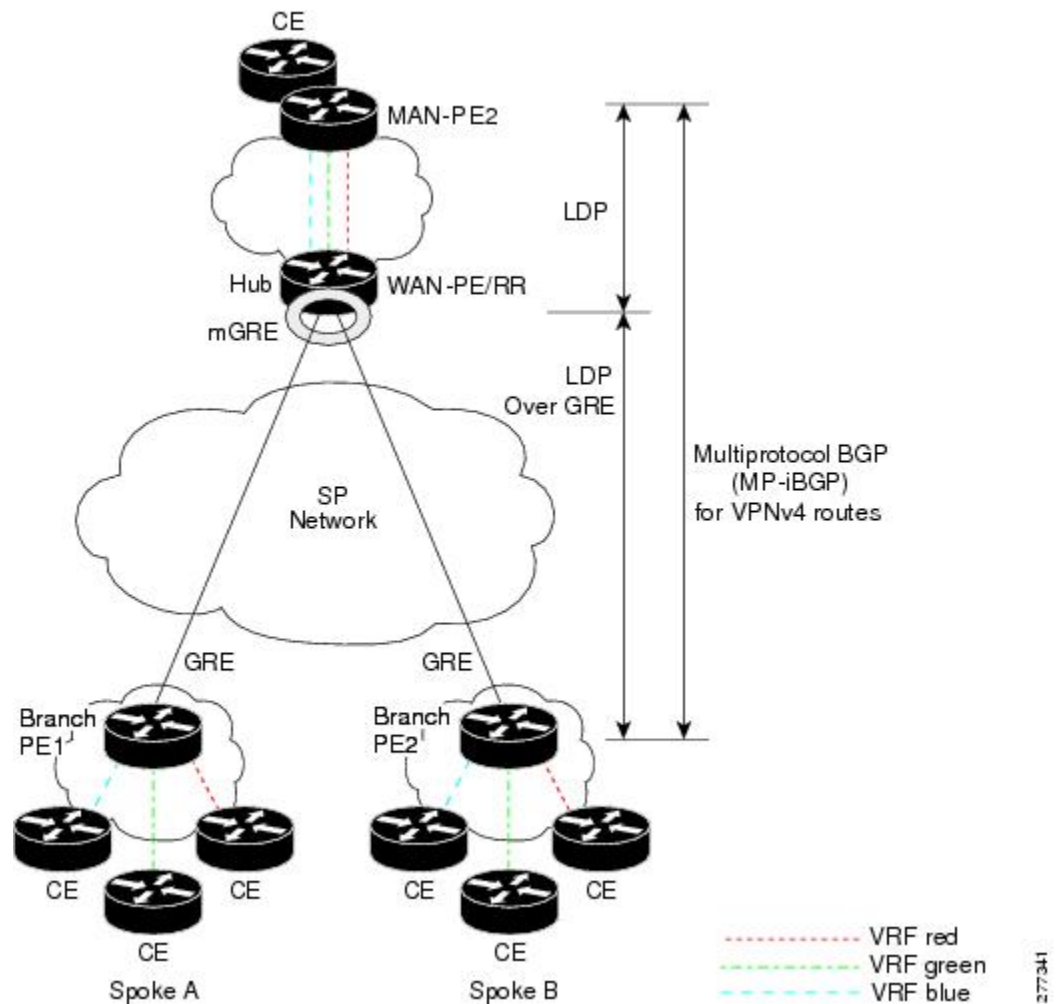
an entire crypto map configuration. An IPsec profile contains only IPsec information; that is, it does not contain any access list information or peering information.

Enabling Traffic Segmentation Within DMVPN

Cisco IOS XE Release 2.5 provides an enhancement that allows you to segment VPN traffic within a DMVPN tunnel by using a PE-PE mGRE tunnel. This secured mGRE tunnel can be used to transport all (or a set of) VPN traffic.

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

Figure 2: 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 internal Border Gateway Protocol [MP-iBGP] peering) and a GRE tunnel to the hub.

- Each spoke advertises its routes and VPN-IPv4 (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 Internet Service Provider [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).

With the NAT-Transparency Aware DMVPN enhancement, NHRP can learn and use the NAT public address for its mappings as long as IPsec transport mode is used (which is the recommended IPsec mode for DMVPN networks). It is recommended that all DMVPN routers be upgraded to the new code before you try to use the NAT-Transparency Aware DMVPN functionality even though spoke routers that are not behind NAT need not 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.

With this NAT Transparency enhancement, the hub DMVPN router can be behind the static NAT. 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.

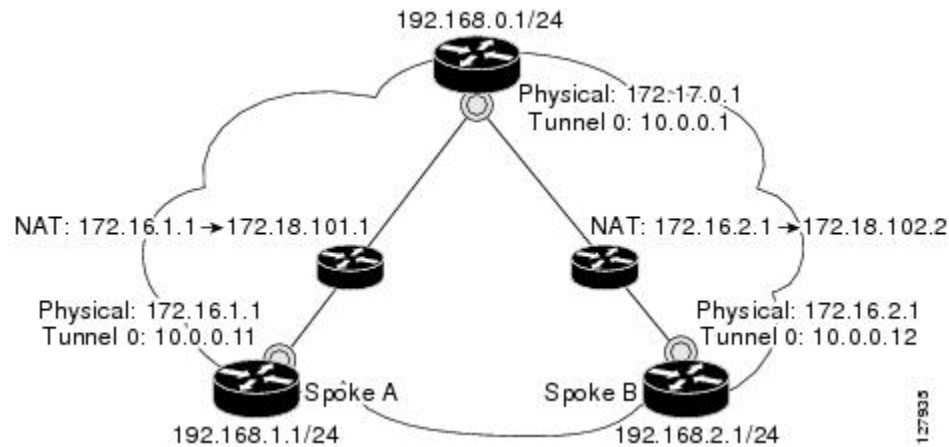
The figure below illustrates a NAT-Transparency Aware DMVPN scenario.



Note

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-to-spoke connections as the NAT box does for the spoke-to-hub connection. If there is more than one DMVPN spoke behind the same NAT box, 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-to-spoke tunnel between these spokes. If a spoke-to-spoke tunnel fails to form, the spoke-to-spoke packets will continue to be forwarded via the spoke-to-hub-spoke path.

Figure 3: 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 “Call Admission Control for IKE” module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*.

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 10,000 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 [System Messages for Cisco IOS XE Software](#).

How to Configure Dynamic Multipoint VPN

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

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.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption \(NGE\)](#) white paper.

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	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
Step 3	crypto ipsec profile <i>name</i> Example: <pre>Router(config)# crypto ipsec profile vpnprof</pre>	Defines the IPsec parameters that are to be used for IPsec encryption between “spoke and hub” and “spoke and spoke” routers. <ul style="list-style-type: none"> • This command enters crypto map configuration mode. • The <i>name</i> argument specifies the name of the IPsec profile.
Step 4	set transform-set <i>transform-set-name</i> Example: <pre>Router(config-crypto-map)# set transform-set trans2</pre>	Specifies which transform sets can be used with the IPsec profile. <ul style="list-style-type: none"> • The <i>transform-set-name</i> argument specifies the name of the transform set.
Step 5	set identity Example: <pre>Router(config-crypto-map)# set identity</pre>	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 6	set security association lifetime { seconds <i>seconds</i> kilobytes <i>kilobytes</i> } Example: <pre>Router(config-crypto-map)# set security association lifetime seconds 1800</pre>	(Optional) Overrides the global lifetime value for the IPsec profile. <ul style="list-style-type: none"> • The seconds <i>seconds</i> option specifies the number of seconds a security association will live before expiring; the kilobytes <i>kilobytes</i> option specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires. • The default for the <i>seconds</i> argument is 3600 seconds.
Step 7	set pfs [group1 group2] Example: <pre>Router(config-crypto-map)# set pfs group2</pre>	(Optional) Specifies that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. <ul style="list-style-type: none"> • If this command is not specified, the default (group1) is enabled. • 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.

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. Do one of the following:
 - **tunnel protection ipsec profile name**
 - **tunnel protection psk key**
13. **bandwidth kbps**
14. **ip tcp adjust-mss max-segment-size**
15. **ip nhrp holdtime seconds**
16. **delay number**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables 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	interface tunnel number Example: Router(config)# interface tunnel 5	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
Step 4	<p>ip address <i>ip-address mask secondary</i></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>
Step 5	<p>ip mtu <i>bytes</i></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>
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 must be set to the same value on all hubs and spokes that are in the same DMVPN network.</p>
Step 7	<p>ip nhrp map multicast dynamic</p> <p>Example:</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>Note Effective with Cisco IOS XE Denali 16.3 ip nhrp map multicast dynamic is enabled by default.</p>
Step 8	<p>ip nhrp network-id <i>number</i></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>Note Effective with Cisco IOS XE Denali 16.3 ip nhrp network-id is enabled by default.</p>
Step 9	<p>tunnel source <i>{ip-address type number}</i></p> <p>Example:</p> <pre>Router(config-if)# tunnel source Gigabitethernet 0/0/0</pre>	<p>Sets the source address for a tunnel interface.</p>
Step 10	<p>tunnel key <i>key-number</i></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>

	Command or Action	Purpose
Step 11	tunnel mode gre multipoint Example: <pre>Router(config-if)# tunnel mode gre multipoint</pre>	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	Do one of the following: <ul style="list-style-type: none"> • tunnel protection ipsec profile <i>name</i> • tunnel protection psk <i>key</i> Example: <pre>Router(config-if)# tunnel protection ipsec profile vpnprof</pre> Example: <pre>Router(config-if)# tunnel protection psk test1</pre>	Associates a tunnel interface with an IPsec profile. <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. or Simplifies the tunnel protection configuration for pre-shared key (PSK) by creating a default IPsec profile.
Step 13	bandwidth <i>kbps</i> Example: <pre>Router(config-if)# bandwidth 1000</pre>	Sets the current bandwidth value for an interface to higher-level protocols. <ul style="list-style-type: none"> • The <i>kbps</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommended bandwidth value is 1000 or greater. • 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.
Step 14	ip tcp adjust-mss <i>max-segment-size</i> Example: <pre>Router(config-if)# ip tcp adjust-mss 1360</pre>	Adjusts the maximum segment size (MSS) value of TCP packets going through a router. <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. • 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.
Step 15	ip nhrp holdtime <i>seconds</i> Example: <pre>Router(config-if)# ip nhrp holdtime 450</pre>	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. <ul style="list-style-type: none"> • The <i>seconds</i> argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The

	Command or Action	Purpose
		recommended value ranges from 300 seconds to 600 seconds.
Step 16	delay <i>number</i> Example: Router(config-if)# delay 1000	(Optional) Changes 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 recommended 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. Do one of the following:
 - **tunnel protection ipsec profile** *name*
 - **tunnel protection psk** *key*
15. **bandwidth** *kbps*
16. **ip tcp adjust-mss** *max-segment-size*
17. **ip nhrp holdtime** *seconds*
18. **delay** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	interface tunnel number 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.
Step 4	ip address ip-address mask secondary Example: <pre>Router(config-if)# ip address 10.0.0.2 255.255.255.0</pre>	Sets a primary or secondary IP address for the tunnel interface. Note All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.
Step 5	ip mtu bytes Example: <pre>Router(config-if)# ip mtu 1400</pre>	Sets the MTU size, in bytes, of IP packets sent on an interface.
Step 6	ip nhrp authentication string Example: <pre>Router(config-if)# ip nhrp authentication donttell</pre>	Configures the authentication string for an interface using NHRP. Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 7	ip nhrp map hub-tunnel-ip-address hub-physical-ip-address Example: <pre>Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1</pre>	Statically configures the IP-to-NBMA address mapping of IP destinations connected to an NBMA network. <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.

	Command or Action	Purpose
Step 8	<p>ip nhrp map multicast <i>hub-physical-ip-address</i></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp map multicast 172.17.0.1</pre>	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.
Step 9	<p>ip nhrp nhs <i>hub-tunnel-ip-address</i></p> <p>Example:</p> <pre>Router(config-if)# ip nhrp nhs 10.0.0.1</pre>	Configures the hub router as the NHRP next-hop server.
Step 10	<p>ip nhrp network-id <i>number</i></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>Note Effective with Cisco IOS XE Denali 16.3 ip nhrp network-id is enabled by default.</p>
Step 11	<p>tunnel source <i>{ip-address type number}</i></p> <p>Example:</p> <pre>Router(config-if)# tunnel source GigabitEthernet 0/0/0</pre>	Sets the source address for a tunnel interface.
Step 12	<p>tunnel key <i>key-number</i></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.
Step 13	<p>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> <pre>Router(config-if)# tunnel destination 172.17.0.1</pre>	<p>Sets the encapsulation mode to mGRE for the tunnel interface.</p> <ul style="list-style-type: none"> Use this command if data traffic can use dynamic spoke-to-spoke traffic. <p>Specifies the destination for a tunnel interface.</p> <ul style="list-style-type: none"> Use this command if data traffic can use hub-and-spoke tunnels.
Step 14	Do one of the following:	Associates a tunnel interface with an IPsec profile.

	Command or Action	Purpose
	<ul style="list-style-type: none"> • tunnel protection ipsec profile <i>name</i> • tunnel protection psk <i>key</i> <p>Example:</p> <pre>Router(config-if)# tunnel protection ipsec profile vpnprof</pre> <p>Example:</p> <pre>Router(config-if)# tunnel protection psk test1</pre>	<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>or</p> <p>Simplifies the tunnel protection configuration for pre-shared key (PSK) by creating a default IPsec profile.</p>
Step 15	<p>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 recommended bandwidth value is 1000 or greater. • The bandwidth setting for the spoke need not equal the bandwidth setting for the DMVPN hub. It is usually easier if all of the spokes use the same or similar value.
Step 16	<p>ip tcp adjust-mss <i>max-segment-size</i></p> <p>Example:</p> <pre>Router(config-if)# ip tcp adjust-mss 1360</pre>	<p>Adjusts the 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. • 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.
Step 17	<p>ip nhrp holdtime <i>seconds</i></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.
Step 18	<p>delay <i>number</i></p> <p>Example:</p> <pre>Router(config-if)# delay 1000</pre>	<p>(Optional) Changes 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 recommended value is 1000.

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

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



Note To configure VRF Blue, use the **ip vrf vrf-name** command in global configuration mode.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables 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	interface <i>type number</i> Example: Router(config)# interface tunnel 0	Configures an interface type and enters interface configuration mode.
Step 4	ip vrf forwarding <i>vrf-name</i> Example: Router(config-if)# ip vrf forwarding Blue	Allows the forwarding of clear-text data IP packets into a VRF.

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.



Note To configure VRF Red, use the **ip vrf vrf-name** command in global configuration mode.

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 privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
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 tunnel 0</pre>	Configures an interface type and enters interface configuration mode.
Step 4	tunnel vrf <i>vrf-name</i> Example: <pre>Router(config-if)# tunnel vrf RED</pre>	Associates a VPN VRF instance with a specific tunnel destination, interface, or subinterface and allows the forwarding of encrypted tunnel packets into a VRF.

Configuring Traffic Segmentation Within DMVPN

Cisco IOS XE Release 2.5 introduces no new commands to use when configuring traffic segmentation, but you must complete the tasks described in the following sections in order to segment traffic within a DMVPN tunnel:

Prerequisites

The tasks that follow assume that the DMVPN tunnel and the VRFs Red and Blue have already been configured.

To configure VRF Red or Blue, use the **ip vrf** *vrf-name* command in global configuration mode.

For information on configuring a DMVPN tunnel, see the [Configuring the Hub for DMVPN, on page 11](#) and the [Configuring the Spoke for DMVPN, on page 15](#). For details about VRF configuration, see the [Configuring the Forwarding of Clear-Text Data IP Packets into a VRF, on page 19](#) and the [Configuring the Forwarding of Encrypted Tunnel Packets into a VRF, on page 19](#).

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.



Note On the Cisco ASR 1000 Series Aggregation Services Routers, only distributed switching is supported. Use the following commands for distributed switching: **ip multicast-routing [vrf vrf-name] [distributed]**, **debug ip bgp vpnv4 unicast**, and **ip cef distributed**.



Note On this platform, only distributed switching is supported. Use the following commands for distributed switching: **ip multicast-routing [vrf vrf-name] [distributed]**, **debug ip bgp vpnv4 unicast**, and **ip cef distributed**.

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 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	interface <i>type number</i> Example: Router(config)# interface tunnel 0	Configures an interface type and enters interface configuration mode.
Step 4	mpls ip Example: Router(config-if)# mpls ip	Enables MPLS tagging of packets on the specified tunnel interface.

Configuring Multiprotocol BGP on the Hub Router

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

For more information about the BGP routing protocol, see the “Cisco BGP Overview” module in the *Cisco IOS XE IP Routing: BGP Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
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. **neighbor** *ipaddress* **route-reflector-client**
10. **neighbor** *ipaddress* **route-map** **nexthop out**
11. **exit**
12. **address-family ipv4** *vrf-name*
13. **redistribute** **connected**
14. **route-map** *map-tag* [**permit**|**deny**] [*sequence-number*]
15. **set ip next-hop** *ipaddress*

DETAILED STEPS

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

	Command or Action	Purpose
	<pre>Router(config-router)# neighbor 10.0.0.11 remote-as 1</pre>	
Step 5	<p>neighbor <i>ipaddress</i> update-source <i>interface</i></p> <p>Example:</p> <pre>Router(config-router)# neighbor 10.10.10.11 update-source Tunnell</pre>	Configures the Cisco IOS XE software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	<p>address-family <i>vpn4</i></p> <p>Example:</p> <pre>Router(config)# address-family vpn4</pre>	Enters address family configuration mode to configure a routing session using VPNv4 address prefixes.
Step 7	<p>neighbor <i>ipaddress</i> activate</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.11 activate</pre>	Enables the exchange of information with a BGP neighbor.
Step 8	<p>neighbor <i>ipaddress</i> send-community extended</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.11 send-community extended</pre>	Specifies that extended community attributes should be sent to a BGP neighbor.
Step 9	<p>neighbor <i>ipaddress</i> route-reflector-client</p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.11 route-reflector-client</pre>	Configures the router as a BGP Route Reflector and configures the specified neighbor as its client.
Step 10	<p>neighbor <i>ipaddress</i> route-map <i>nexthop out</i></p> <p>Example:</p> <pre>Router(config-router-af)# neighbor 10.0.0.11 route-map nexthop out</pre>	Forces all traffic to be routed via the hub.
Step 11	<p>exit</p> <p>Example:</p> <pre>Router(config-router-af)# exit</pre>	Exits the address family configuration mode for VPNv4.
Step 12	<p>address-family <i>ipv4 vrf-name</i></p> <p>Example:</p> <pre>Router(config)# address-family ipv4 red</pre>	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
Step 13	redistribute connected Example: <pre>Router(config-router-af)# 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	route-map map-tag [permit deny] [sequence-number] Example: <pre>Router(config-router-af)# route-map cisco permit 10</pre>	Enters route map configuration mode to configure the next-hop that will be advertised to the spokes.
Step 15	set ip next-hop ipaddress Example: <pre>Router(config-route-map)# set ip next-hop 10.0.0.1</pre>	Sets the next hop to be the hub.

Configuring Multiprotocol BGP on the Spoke Routers

In order to segment traffic within a DMVPN tunnel, Multiprotocol-iBGP (MP-iBGP) must be configured on both the spoke routers and the hub. Perform the following task for each spoke router in the DMVPN.

SUMMARY STEPS

1. **enable**
2. configure terminal
3. **router bgp autonomous-system-number**
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**
10. **address-family ipv4 vrf-name**
11. **redistribute connected**
12. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
	Router# configure terminal	
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 1	Enters BGP configuration mode.
Step 4	neighbor <i>ipaddress</i> remote-as <i>as - number</i> Example: Router(config-router)# neighbor 10.0.0.1 remote-as 1	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 5	neighbor <i>ipaddress</i> update-source <i>interface</i> Example: Router(config-router)# neighbor 10.10.10.1 update-source Tunnell	Configures the Cisco IOS XE software to allow BGP sessions to use any operational interface for TCP connections.
Step 6	address-family vpnv4 Example: Router(config)# address-family vpnv4	Enters address family configuration mode to configure a routing session using VPNv4 address prefixes.
Step 7	neighbor <i>ipaddress</i> activate Example: Router(config-router-af)# neighbor 10.0.0.1 activate	Enables the exchange of information with a BGP neighbor.
Step 8	neighbor <i>ipaddress</i> send-community extended Example: Router(config-router-af)# neighbor 10.0.0.1 send-community extended	Specifies that extended community attributes should be sent to a BGP neighbor.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.
Step 10	address-family ipv4 <i>vrf-name</i> Example: Router(config)# address-family ipv4 red	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.

	Command or Action	Purpose
Step 11	redistribute connected Example: <pre>Router(config-router-af)# 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 12	exit Example: <pre>Router(config-router-af)# exit</pre>	Exits address family configuration mode. Note Repeat Steps 10 through 12 for each VRF.

Troubleshooting Dynamic Multipoint VPN

After configuring DMVPN, perform the following optional steps in this task to verify that DMVPN is operating correctly, to clear DMVPN statistics or sessions, or to debug DMVPN. These commands may be used in any order.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption \(NGE\)](#) white paper.

SUMMARY STEPS

1. **clear dmvpn session**
2. **clear dmvpn statistics**
3. **debug dmvpn**
4. **debug dmvpn condition**
5. **debug nhrp condition**
6. **debug nhrp error**
7. **logging dmvpn**
8. **show crypto ipsec sa**
9. **show crypto isakmp sa**
10. **show crypto map**
11. **show dmvpn**
12. **show ip nhrp traffic**

DETAILED STEPS

Step 1 clear dmvpn session

This command clears DMVPN sessions. The following example clears only dynamic DMVPN sessions, for the specified tunnel:

Example:


```
Router# clear dmvpn session interface tunnel 5
```

The following example clears all DMVPN sessions, both static and dynamic, for the specified tunnel:

Example:

```
Router# clear dmvpn session interface tunnel 5 static
```

Step 2 clear dmvpn statistics

This command is used to clear DMVPN-related counters. The following example shows how to clear DMVPN-related session counters for the specified tunnel interface:

Example:

```
Router#  
clear dmvpn statistics interface tunnel 5
```

Step 3 debug dmvpn

This 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 NHRP, sockets, tunnel protection, and crypto information:

Example:

```
Router# debug dmvpn error all
```

Step 4 debug dmvpn condition

This command displays conditional debug DMVPN session information. The following example shows how to enable conditional debugging for a specific tunnel interface:

Example:

```
Router# debug dmvpn condition interface tunnel 5
```

Step 5 debug nhrp condition

This command enables or disables debugging based on a specific condition. The following example shows how to enable conditional NHRP debugging:

Example:

```
Router#  
debug nhrp condition
```

Step 6 debug nhrp error

This command displays information about NHRP error activity. The following example shows how to enable debugging for NHRP error messages:

Example:

```
Router#
debug nhrp error
```

Step 7 logging dmvpn

This command is used to enable DMVPN system logging. The following example shows how to enable DMVPN system logging at the rate of 1 message every 20 seconds:

Example:

```
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 8 show crypto ipsec sa

This 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: gigabitethernet0/0/0
  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 9 show crypto isakmp sa

This 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 10 show crypto map

This 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 11 show dmvpn

This command displays DMVPN-specific session information. The following sample 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 12 show ip nhrp traffic

This command displays NHRP statistics. The following example shows output for a specific tunnel (tunnel7):

Example:

```

Router# s
how ip nhrp traffic interface tunnel7
Tunnel7: Max-send limit:10000Pkts/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

Proceed to the following sections “Configuring the Hub for DMVPN” and “Configuring the Spoke for DMVPN.”

Configuration Examples for Dynamic Multipoint VPN Feature

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 GigabitEthernet 0/0/0
 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 FastEthernet0/0/0
 ip address 172.17.0.1 255.255.255.0
!
interface FastEthernet0/0/1
 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 “Certificate to ISAKMP Profile Mapping” module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*.

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 GigabitEthernet 0/0/0
 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 FastEthernet0/0/0
 ip address dhcp hostname Spoke1
!
interface FastEthernet0/0/1
 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 2547oDMVPN with BGP Only Traffic Segmentation

The following example show a traffic segmentation configuration in which traffic is segmented between two spokes that serve as PE devices:

Hub Configuration

```

hostname hub-pel
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 GigabitEthernet 0/0/0
  tunnel mode gre multipoint
  tunnel protection ipsec profile prof
interface Loopback0
  ip address 10.0.0.1 255.255.255.255
interface Ethernet0/0/0
  ip address 172.0.0.1 255.255.255.0
!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes
learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.0.0.11 remote-as 1
  neighbor 10.0.0.11 update-source Tunnell
  neighbor 10.0.0.12 remote-as 1
  neighbor 10.0.0.12 update-source Tunnell
  no auto-summary
  address-family vpnv4
    neighbor 10.0.0.11 activate
    neighbor 10.0.0.11 send-community extended
    neighbor 10.0.0.11 route-reflector-client
    neighbor 10.0.0.11 route-map nexthop out
    neighbor 10.0.0.12 activate
    neighbor 10.0.0.12 send-community extended
    neighbor 10.0.0.12 route-reflector-client
    neighbor 10.0.0.12 route-map nexthop out
  exit
  address-family ipv4 vrf red
    redistribute connected
    no synchronization
  exit
  address-family ipv4 vrf blue
    redistribute connected
    no synchronization
  exit
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 cisco 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 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 GigabitEthernet 0/0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof
interface Loopback0
  ip address 10.9.9.11 255.255.255.255
interface FastEthernet0/0/0
  ip address 172.0.0.11 255.255.255.0
!
!
interface FastEthernet1/0/0
  ip vrf forwarding red
  ip address 192.168.11.2 255.255.255.0
interface FastEthernet2/0/0
  ip vrf forwarding blue
  ip address 192.168.11.2 255.255.255.0
!The multiprotocol BGP route reflector (the hub) configuration changes the next-hop
information to set itself as the next-hop and assigns a new VPN label for the prefixes

```



```

learned from the spokes and advertises the VPN prefix:
router bgp 1
  no synchronization
  bgp log-neighbor-changes
  neighbor 10.0.0.1 remote-as 1
  neighbor 10.0.0.1 update-source Tunnell
  no auto-summary
  address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 send-community extended
  exit
!
  address-family ipv4 vrf red
  redistribute connected
  no synchronization
  exit
!
  address-family ipv4 vrf blue
  redistribute connected
  no synchronization
  exit
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 GigabitEthernet 0/0/0
tunnel mode gre multipoint
tunnel protection ipsec profile prof
!
interface Loopback0
 ip address 10.9.9.12 255.255.255.255
interface FastEthernet0/0/0
 ip address 172.0.0.12 255.255.255.0
interface FastEthernet1/0/0
 ip vrf forwarding red
 ip address 192.168.12.2 255.255.255.0
interface FastEthernet2/0/0
 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 Tunnel1
 no auto-summary
 address-family vpnv4
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 send-community extended
 exit
 address-family ipv4 vrf red
 redistribute connected
 no synchronization
 exit
 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit
 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 Enterprise Branch Traffic Segmentation

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 GigabitEthernet 0/0/0
  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 FastEthernet0/0/0
  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 ipv4 vrf red
redistribute connected
no synchronization
exit
address-family ipv4 vrf blue
redistribute connected
no synchronization
exit
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 GigabitEthernet 0/0/0
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 FastEthernet0/0/0
 ip address 172.0.0.11 255.255.255.0
interface FastEthernet1/0/0
 ip vrf forwarding red
 ip address 192.168.11.2 255.255.255.0
interface FastEthernet2/0/0
 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 ipv4 vrf red
 redistribute connected
 no synchronization
 exit
 address-family ipv4 vrf blue
 redistribute connected
 no synchronization
 exit
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 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 GigabitEthernet 0/0/0
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 FastEthernet0/0/0
  ip address 172.0.0.12 255.255.255.0
interface FastEthernet1/0/0
  ip vrf forwarding red
  ip address 192.168.12.2 255.255.255.0
interface FastEthernet2/0/0
  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 ipv4 vrf red
  redistribute connected
  no synchronization

```

```

exit
address-family ipv4 vrf blue
redistribute connected
no synchronization
exit
no ip http server
no ip http secure-server
control-plane
line con 0
  logging synchronous
line aux 0
line vty 0 4
  no login
end

```

Sample Command Output: show mpls ldp bindings

```

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

```

Sample Command Output: show mpls forwarding-table

```

Spoke2# show mpls forwarding-table

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

```

Sample Command Output: show ip route vrf red

```

Spoke2# show ip route vrf red
Routing Table: red
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

```

```

Gateway of last resort is not set
B   192.168.12.0/24 [200/0] via 10.9.9.12, 00:00:02
C   192.168.11.0/24 is directly connected, FastEthernet1/0/0
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, FastEthernet2/0/0
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, Tunnell 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 for Dynamic Multipoint VPN

Related Documents

Related Topic	Document Title
Call Admission Control	<i>Call Admission Control for IKE</i>
IKE configuration tasks such as defining an IKE policy	<i>Configuring Internet Key Exchange for IPsec VPNs</i>
IPsec configuration tasks	<i>Configuring Security for VPNs with IPsec</i>
Configuring VRF-aware IPsec	<i>VRF-Aware IPsec</i>
Configuring MPLS	<i>Multiprotocol Label Switching (MPLS) on Cisco Routers</i>
Configuring BGP	<i>Cisco BGP Overview</i>
Defining and configuring ISAKMP profiles	<i>Certificate to ISAKMP Profile Mapping</i>
Security commands	Cisco IOS Security Command Reference
Recommended cryptographic algorithms	Next Generation Encryption

RFCs

RFCs	Title
RFC 2547	BGP/MPLS VPNs

Feature Information for Dynamic Multipoint VPN

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

Feature Name	Releases	Feature Information
Dynamic Multipoint VPN (DMVPN) Phase 1	Cisco IOS XE Release 2.1	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).

Feature Name	Releases	Feature Information
DMVPN Phase 2	Cisco IOS XE Release 2.1	DMVPN Spoke-to-Spoke functionality was made more production ready.
NAT-Transparency Aware DMVPN	Cisco IOS XE Release 2.1	The Network Address Translation-Transparency (NAT-T) Aware DMVPN enhancement was added. In addition, DMVPN hub-to-spoke functionality was made more production ready.
Manageability Enhancements for DMVPN	Cisco IOS XE Release 2.5	DMVPN session manageability was expanded with DMVPN-specific commands for debugging, show output, session and counter control, and system log information. The following section provides information about this feature: <ul style="list-style-type: none"> • Troubleshooting Dynamic Multipoint VPN The following commands were introduced or modified by this feature: clear dmvpn session , clear dmvpn statistics , debug dmvpn , debug dmvpn condition , debug nhrp condition , debug nhrp error , logging dmvpn , show dmvpn , show ip nhrp traffic
DMVPN--Enabling Traffic Segmentation Within DMVPN	Cisco IOS XE Release 2.5	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.

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 XE 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 an 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, FastEthernet, SMDS, and multipoint tunnel networks. Although NHRP is available on FastEthernet, NHRP need not be implemented over FastEthernet media because FastEthernet is capable of broadcasting. FastEthernet 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 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.

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.



CHAPTER 3

IPv6 over DMVPN

This document describes how to implement the Dynamic Multipoint VPN for IPv6 feature, which 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 the Next Hop Resolution Protocol (NHRP). In Dynamic Multipoint Virtual Private Network (DMVPN) for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable.

IPv6 support on DMVPN was extended to the public network (the Internet) facing the Internet service provider (ISP). The IPv6 transport for DMVPN feature builds IPv6 WAN-side capability into NHRP tunnels and the underlying IPsec encryption, and enables IPv6 to transport payloads on the Internet.

The IPv6 transport for DMVPN feature is enabled by default. You need not upgrade your private internal network to IPv6 for the IPv6 transport for DMVPN feature to function. You can have either IPv4 or IPv6 addresses on your local networks.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Prerequisites for IPv6 over DMVPN, on page 47](#)
- [Information About IPv6 over DMVPN, on page 48](#)
- [How to Configure IPv6 over DMVPN, on page 50](#)
- [Configuration Examples for IPv6 over DMVPN, on page 63](#)
- [Additional References, on page 67](#)
- [Feature Information for IPv6 over DMVPN, on page 68](#)

Prerequisites for IPv6 over DMVPN

- One of the following protocols must be enabled for DMVPN for IPv6 to work: Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), On-Demand Routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
- Every IPv6 NHRP interface is configured with one IPv6 unicast address. This address can be a globally reachable or unique local address.

- Every IPv6 NHRP interface is configured with one IPv6 link-local address that is unique across all DMVPN hosts in the DMVPN cloud (that is, the hubs and spokes).

Information About IPv6 over DMVPN

DMVPN for IPv6 Overview

The DMVPN feature combines NHRP routing, multipoint generic routing encapsulation (mGRE) tunnels, and IPsec encryption to provide users 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 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 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--An mGRE tunnel interface allows a single GRE interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.
- IPsec encryption--An IPsec tunnel interface facilitates for the protection of site-to-site IPv6 traffic with native encapsulation.

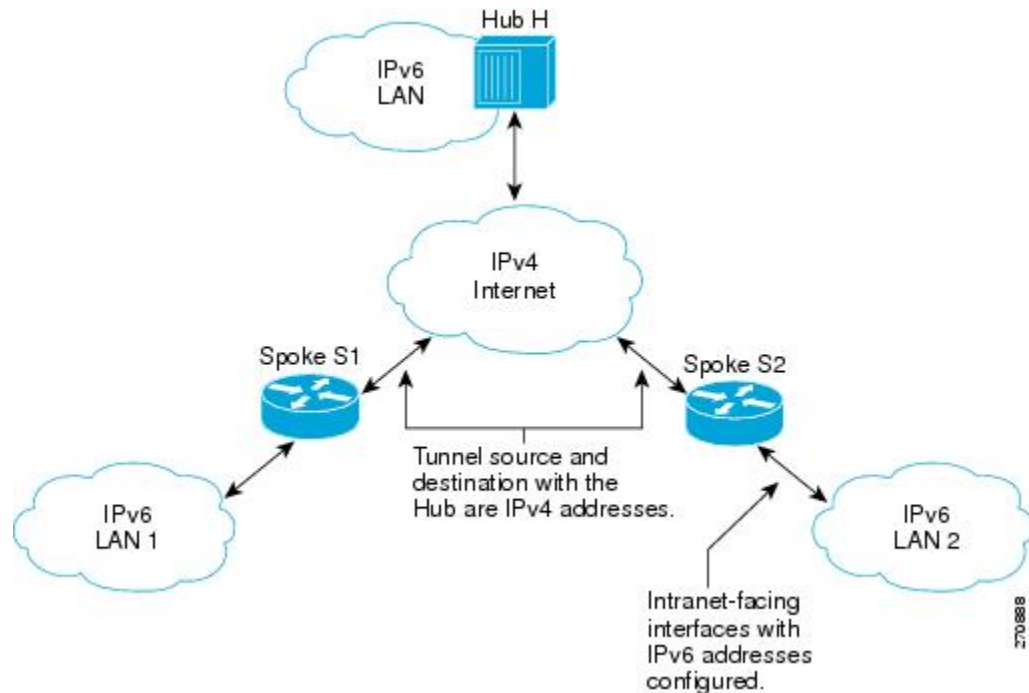
In DMVPN for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable. The intranets could be a mix of IPv4 or IPv6 clouds connected to each other using DMVPN technologies, with the underlying carrier being a traditional IPv4 network.

NHRP Routing

The NHRP protocol resolves a given intranet address (IPv4 or IPv6) to an Internet address (IPv4 nonbroadcast multiaccess [NBMA] address).

In the figure below, the intranets that are connected over the DMVPN network are IPv6 clouds, and the Internet is a pure IPv4 cloud. Spokes S1 and S2 are connected to Hub H over the Internet using a statically configured tunnel. The address of the tunnel itself is the IPv6 domain, because it is another node on the intranet. The source and destinations address of the tunnel (the mGRE endpoints), however, are always in IPv4, in the Internet domain. The mGRE tunnel is aware of the IPv6 network because the GRE passenger protocol is an IPv6 packet, and the GRE transport (or carrier) protocol is an IPv4 packet.

Figure 4: IPv6 Topology That Triggers NHRP



When an IPv6 host in LAN L1 sends a packet destined to an IPv6 host in LAN L2, the packet is first routed to the gateway (which is Spoke S1) in LAN L1. Spoke S1 is a dual-stack device, which means both IPv4 and IPv6 are configured on it. The IPv6 routing table in S1 points to a next hop, which is the IPv6 address of the tunnel on Spoke S2. This is a VPN address that must be mapped to an NBMA address, triggering NHRP.

IPv6 NHRP Redirect and Shortcut Features

When IPv6 NHRP redirect is enabled, NHRP examines every data packet in the output feature path. If the data packet enters and leaves on the same logical network, NHRP sends an NHRP traffic indication message to the source of the data packet. In NHRP, a logical network is identified by the NHRP network ID, which groups multiple physical interfaces into a single logical network.

When IPv6 NHRP shortcut is enabled, NHRP intercepts every data packet in the output feature path. It checks to see if there is an NHRP cache entry to the destination of the data packet and, if yes, it replaces the current output adjacency with the one present in the NHRP cache. The data packet is therefore switched out using the new adjacency provided by NHRP.

IPv6 Routing

NHRP is automatically invoked for mGRE tunnels carrying the IPv6 passenger protocol. When a packet is routed and sent to the switching path, NHRP looks up the given next hop and, if required, initiates an NHRP resolution query. If the resolution is successful, NHRP populates the tunnel endpoint database, which in turn populates the Cisco Express Forwarding adjacency table. The subsequent packets are Cisco Express Forwarding switched if Cisco Express Forwarding is enabled.

IPv6 Addressing and Restrictions

IPv6 allows multiple unicast addresses on a given IPv6 interface. IPv6 also allows special address types, such as anycast, multicast, link-local addresses, and unicast addresses.

DMVPN for IPv6 has the following addressing restrictions:

- Every IPv6 NHRP interface is configured with one IPv6 unicast address. This address can be a globally reachable or unique local address.
- Every IPv6 NHRP interface is configured with one IPv6 link-local address that is unique across all DMVPN hosts in the DMVPN cloud (that is, the hubs and spokes).
 - If no other tunnels on the device are using the same tunnel source, then the tunnel source address can be embedded into an IPv6 address.
 - If the device has only one DMVPN IPv6 tunnel, then manual configuration of the IPv6 link-local address is not required. Instead, use the **ipv6 enable** command to autogenerate a link-local address.
 - If the device has more than one DMVPN IPv6 tunnel, then the link-local address must be manually configured using the **ipv6 address fe80::2001 link-local** command.

How to Configure IPv6 over DMVPN

Configuring an IPsec Profile in DMVPN for IPv6



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption \(NGE\)](#) white paper.

The IPsec profile shares most 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 you begin

Before configuring an IPsec profile, you must do the following:

- Define a transform set by using the **crypto ipsec transform-set** command.
- Make sure that the Internet Security Association Key Management Protocol (ISAKMP) profile is configured with default ISAKMP settings.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto identity *name***
4. **exit**

5. **crypto ipsec profile** *name*
6. **set transform-set** *transform-set-name*
7. **set identity**
8. **set security-association lifetime** *seconds seconds* | *kilobytes kilobytes*
9. **set pfs** [*group1* | *group14* | *group15* | *group16* | *group19* | *group2* | *group20* | *group24* | *group5*]
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto identity <i>name</i> Example: Device(config)# crypto identity device1	Configures the identity of the device with a given list of distinguished names (DNs) in the certificate of the device.
Step 4	exit Example: Device(config-crypto-identity)# exit	Exits crypto identity configuration mode and enters global configuration mode.
Step 5	crypto ipsec profile <i>name</i> Example: Device(config)# crypto ipsec profile example1	Defines the IPsec parameters that are to be used for IPsec encryption between "spoke and hub" and "spoke and spoke" routers. This command places the device in crypto map configuration mode.
Step 6	set transform-set <i>transform-set-name</i> Example: Device(config-crypto-map)# set transform-set example-set	Specifies which transform sets can be used with the IPsec profile.
Step 7	set identity Example: Device(config-crypto-map)# set identity router1	(Optional) Specifies identity restrictions to be used with the IPsec profile.

	Command or Action	Purpose
Step 8	<p>set security-association lifetime <i>seconds seconds kilobytes kilobytes</i></p> <p>Example:</p> <pre>Device(config-crypto-map)# set security-association lifetime seconds 1800</pre>	(Optional) Overrides the global lifetime value for the IPsec profile.
Step 9	<p>set pfs [<i>group1 group14 group15 group16 group19 group2 group20 group24 group5</i>]</p> <p>Example:</p> <pre>Device(config-crypto-map)# set pfs group14</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 Diffie-Hellman (DH) group, group1 will be enabled.</p> <ul style="list-style-type: none"> • 1—768-bit DH (No longer recommended.) • 2—1024-bit DH (No longer recommended) • 5—1536-bit DH (No longer recommended) • 14—Specifies the 2048-bit DH group. • 15—Specifies the 3072-bit DH group. • 16—Specifies the 4096-bit DH group. • 19—Specifies the 256-bit elliptic curve DH (ECDH) group. • 20—Specifies the 384-bit ECDH group. • 24—Specifies the 2048-bit DH/DSA group.
Step 10	<p>end</p> <p>Example:</p> <pre>Device(config-crypto-map)# end</pre>	Exits crypto map configuration mode and returns to privileged EXEC mode.

Configuring the Hub for IPv6 over DMVPN

Perform this task to configure the hub device for IPv6 over DMVPN for mGRE and IPsec integration (that is, associate the tunnel with the IPsec profile configured in the previous procedure).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ipv6 address** *{ipv6-address / prefix-length | prefix-name sub-bits / prefix-length}*
5. **ipv6 address** *ipv6-address / prefix-length link-local*
6. **ipv6 mtu** *bytes*

7. **ipv6 nhrp authentication** *string*
8. **ipv6 nhrp map multicast dynamic**
9. **ipv6 nhrp network-id** *network-id*
10. **tunnel source** *ip-address* | *ipv6-address* | *interface-type interface-number*
11. **tunnel mode** {aurp | cayman | dvmrp | eon | gre| gre multipoint[ipv6] | gre ipv6 | ipip decapsulate-any] | ipsec ipv4 | iptalk | ipv6| ipsec ipv6 | mpls | nos | rbscp
12. Do one of the following:
 - **tunnel protection ipsec profile** *name* [shared]
 - **tunnel protection psk** *key*
13. **bandwidth** {*kpbs* | inherit [*kpbs*] | receive [*kpbs*]}
14. **ipv6 nhrp holdtime** *seconds*
15. **ipv6 nhrp max-send** *pkt-count* every *seconds*
16. **ip nhrp registration** [timeout *seconds* | no-unique]
17. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 5	Configures a tunnel interface and enters interface configuration mode. <ul style="list-style-type: none"> • The number argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address { <i>ipv6-address / prefix-length</i> <i>prefix-name sub-bits / prefix-length</i> } Example: Device(config-if)# ipv6 address 2001:DB8:1:1::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	ipv6 address <i>ipv6-address / prefix-length</i> link-local Example: Device(config-if)# ipv6 address fe80::2001 link-local	Configures an IPv6 link-local address for an interface and enables IPv6 processing on the interface. <ul style="list-style-type: none"> • A unique IPv6 link-local address (across all DMVPN nodes in a DMVPN network) must be configured.

	Command or Action	Purpose
Step 6	ipv6 mtu <i>bytes</i> Example: Device(config-if)# ipv6 mtu 1400	Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface.
Step 7	ipv6 nhrp authentication <i>string</i> Example: Device(config-if)# ipv6 nhrp authentication examplexx	Configures the authentication string for an interface using the NHRP. Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 8	ipv6 nhrp map multicast dynamic Example: Device(config-if)# ipv6 nhrp map multicast dynamic	Allows NHRP to automatically add routers to the multicast NHRP mappings. Note Effective with Cisco IOS XE Denali 16.3 ipv6 nhrp map multicast dynamic is enabled by default.
Step 9	ipv6 nhrp network-id <i>network-id</i> Example: Device(config-if)# ipv6 nhrp network-id 99	Enables the NHRP on an interface. Effective with Cisco IOS XE Denali 16.3 ipv6 nhrp network-id is enabled by default.
Step 10	tunnel source <i>ip-address ipv6-address interface-type interface-number</i> Example: Device(config-if)# tunnel source ethernet 0	Sets the source address for a tunnel interface.
Step 11	tunnel mode {aurp cayman dvmrp eon gre gre multipoint[ipv6] gre ipv6 ipip decapsulate-any} ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbscp Example: Device(config-if)# tunnel mode gre multipoint	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	Do one of the following: <ul style="list-style-type: none"> • tunnel protection ipsec profile <i>name</i> [shared] • tunnel protection psk <i>key</i> Example: Router(config-if)# tunnel protection ipsec profile vpnprof Example: Router(config-if)# tunnel protection psk test1	Associates a tunnel interface with an IPsec profile. <ul style="list-style-type: none"> • The <i>name</i> argument specifies the name of the IPsec profile; this value must match the <i>name</i> specified in the crypto ipsec profile name command. or Simplifies the tunnel protection configuration for pre-shared key (PSK) by creating a default IPsec profile.

	Command or Action	Purpose
Step 13	bandwidth { <i>kbps</i> inherit [<i>kbps</i>] receive [<i>kbps</i>]} Example: <pre>Device(config-if)# bandwidth 1200</pre>	Sets the current bandwidth value for an interface to higher-level protocols. <ul style="list-style-type: none"> The <i>bandwidth-size</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommended bandwidth value is 1000 or greater.
Step 14	ipv6 nhrp holdtime <i>seconds</i> Example: <pre>Device(config-if)# ipv6 nhrp holdtime 600</pre>	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. The default time is 600 seconds.
Step 15	ipv6 nhrp max-send <i>pkt-count</i> every <i>seconds</i> Example: <pre>Device(config-if)# ipv6 nhrp max-send 10000 every 10</pre>	Changes the maximum frequency at which NHRP packets can be sent. Number of packets that can be sent in the range from 1 to 65535. Default is 100 packets.
Step 16	ip nhrp registration [<i>timeout seconds</i> no-unique] Example: <pre>Device(config-if)# ip nhrp registration no-unique</pre>	Enables the client to not set the unique flag in the NHRP request and reply packets. The default is no-unique.
Step 17	end Example: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the NHRP Redirect and Shortcut Features on the Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel number**
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits / prefix-length*}
5. Do one of the following:
 - **ipv6 nhrp redirect** [*timeout seconds*]
 - **ipv6 nhrp redirect** [*interest acl*]
6. **ipv6 nhrp shortcut**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface tunnel number Example: <pre>Device(config)# interface tunnel 5</pre>	Configures a tunnel interface and enters interface configuration mode. <ul style="list-style-type: none"> • The number argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address {ipv6-address / prefix-length prefix-name sub-bits / prefix-length} Example: <pre>Device(config-if)# ipv6 address 2001:DB8:1:1::72/64</pre>	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	Do one of the following: <ul style="list-style-type: none"> • ipv6 nhrp redirect [timeout seconds] • ipv6 nhrp redirect [interest acl] Example: <pre>Device(config-if)# ipv6 nhrp redirect</pre> Example: <pre>Device(config-if)# ipv6 nhrp redirect interest</pre>	Enables NHRP redirect. or Enables the user to specify an ACL. Note You must configure the ipv6 nhrp redirect command on a hub.
Step 6	ipv6 nhrp shortcut Example: <pre>Device(config-if)# ipv6 nhrp shortcut</pre>	Enables NHRP shortcut switching. <ul style="list-style-type: none"> • You must configure the ipv6 nhrp shortcut command on a spoke. Note Effective with Cisco IOS XE Denali 16.3 ipv6 nhrp shortcut is enabled by default.
Step 7	end Example: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the Spoke for IPv6 over DMVPN

Perform this task to configure the spoke for IPv6 over DMVPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ipv6 address** *{ipv6-address / prefix-length | prefix-name sub-bits / prefix-length}*
5. **ipv6 address** *ipv6-address / prefix-length* **link-local**
6. **ipv6 mtu** *bytes*
7. **ipv6 nhrp authentication** *string*
8. **ipv6 nhrp map** *ipv6-address nbma-address*
9. **ipv6 nhrp map multicast** *ipv4-nbma-address*
10. **ipv6 nhrp nhs** *ipv6- nhs-address*
11. **ipv6 nhrp network-id** *network-id*
12. **tunnel source** *ip-address | ipv6-address | interface-type interface-number*
13. Do one of the following:
 - **tunnel mode** *{aurp | cayman | dvmrp | eon | gre| gre multipoint [ipv6] | gre ipv6 | ipip decapsulate-any} | ipsec ipv4 | iptalk | ipv6| ipsec ipv6 | mpls | nos | rbscp*
 - **tunnel destination** *{host-name | ip-address | ipv6-address}*
14. Do one of the following:
 - **tunnel protection ipsec profile** *name [shared]*
 - **tunnel protection psk** *key*
15. **bandwidth** *{interzone | total | session} {default | zone zone-name} bandwidth-size*
16. **ipv6 nhrp holdtime** *seconds*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example:	Configures a tunnel interface and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface tunnel 5	<ul style="list-style-type: none"> The <i>number</i> argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address { <i>ipv6-address / prefix-length prefix-name sub-bits / prefix-length</i> Example: Device(config-if) ipv6 address 2001:DB8:1:1::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	ipv6 address <i>ipv6-address / prefix-length link-local</i> Example: Device(config-if)# ipv6 address fe80::2001 link-local	Configures an IPv6 link-local address for an interface and enables IPv6 processing on the interface. <ul style="list-style-type: none"> A unique IPv6 link-local address (across all DMVPN nodes in a DMVPN network) must be configured.
Step 6	ipv6 mtu <i>bytes</i> Example: Device(config-if)# ipv6 mtu 1400	Sets the MTU size of IPv6 packets sent on an interface.
Step 7	ipv6 nhrp authentication <i>string</i> Example: Device(config-if)# ipv6 nhrp authentication examplexx	Configures the authentication string for an interface using the NHRP. <p>Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.</p>
Step 8	ipv6 nhrp map <i>ipv6-address nbma-address</i> Example: Device(config-if)# ipv6 nhrp map 2001:DB8:3333:4::5 10.1.1.1	Statically configures the IPv6-to-NBMA address mapping of IPv6 destinations connected to an NBMA network. <p>Note Only IPv4 NBMA addresses are supported, not ATM or Ethernet addresses.</p>
Step 9	ipv6 nhrp map multicast <i>ipv4-nbma-address</i> Example: Device(config-if)# ipv6 nhrp map multicast 10.11.11.99	Maps destination IPv6 addresses to IPv4 NBMA addresses.
Step 10	ipv6 nhrp nhs <i>ipv6- nhs-address</i> Example: Device(config-if)# ipv6 nhrp nhs 2001:0DB8:3333:4::5 2001:0DB8::/64	Specifies the address of one or more IPv6 NHRP servers.
Step 11	ipv6 nhrp network-id <i>network-id</i> Example:	Enables the NHRP on an interface.

	Command or Action	Purpose
	Device(config-if)# ipv6 nhrp network-id 99	Note Effective with Cisco IOS XE Denali 16.3 ipv6 nhrp network-id is enabled by default.
Step 12	<p>tunnel source <i>ip-address</i> <i>ipv6-address</i> <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Device(config-if)# tunnel source ethernet 0</pre>	Sets the source address for a tunnel interface.
Step 13	<p>Do one of the following:</p> <ul style="list-style-type: none"> • tunnel mode {aurp cayman dvmrp eon gre gre multipoint [ipv6] gre ipv6 ipip decapsulate-any} ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbscp • tunnel destination {host-name ip-address ipv6-address} <p>Example:</p> <pre>Device(config-if)# tunnel mode gre multipoint</pre> <p>Example:</p> <pre>Device(config-if)# tunnel destination 10.1.1.1</pre>	<p>Sets the encapsulation mode to mGRE for the tunnel interface.</p> <ul style="list-style-type: none"> • Use the tunnel mode command if data traffic can use dynamic spoke-to-spoke traffic. <p>or</p> <p>Specifies the destination for a tunnel interface.</p> <ul style="list-style-type: none"> • Use the tunnel destination command if data traffic can use hub-and-spoke tunnels.
Step 14	<p>Do one of the following:</p> <ul style="list-style-type: none"> • tunnel protection ipsec profile <i>name</i> [shared] • tunnel protection psk <i>key</i> <p>Example:</p> <pre>Router(config-if)# tunnel protection ipsec profile vpnprof</pre> <p>Example:</p> <pre>Router(config-if)# tunnel protection psk test1</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>or</p> <p>Simplifies the tunnel protection configuration for pre-shared key (PSK) by creating a default IPsec profile.</p>
Step 15	<p>bandwidth {interzone total session} {default zone <i>zone-name</i>} <i>bandwidth-size</i></p> <p>Example:</p> <pre>Device(config-if)# bandwidth total 1200</pre>	<p>Sets the current bandwidth value for an interface to higher-level protocols.</p> <ul style="list-style-type: none"> • The <i>bandwidth-size</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommended bandwidth value is 1000 or greater. • The bandwidth setting for the spoke need not equal the bandwidth setting for the DMVPN hub. It is usually easier if all of the spokes use the same or similar value.

	Command or Action	Purpose
Step 16	ipv6 nhrp holdtime <i>seconds</i> Example: Device(config-if)# ipv6 nhrp holdtime 3600	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses.
Step 17	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying DMVPN for IPv6 Configuration

SUMMARY STEPS

1. **enable**
2. **show dmvpn** [**ipv4** [**vrf vrf-name**] | **ipv6** [**vrf vrf-name**]] [**debug-condition** | [**interface tunnel number** | **peer** {**nbma ip-address** | **network network-mask** | **tunnel ip-address**}] [**static**] [**detail**]
3. **show ipv6 nhrp** [**dynamic** [**ipv6-address**] | **incomplete** | **static**] [**address** | **interface**] [**brief** | **detail**] [**purge**]
4. **show ipv6 nhrp multicast** [**ipv4-address** | **interface** | **ipv6-address**]
5. **show ip nhrp multicast** [**nbma-address** | **interface**]
6. **show ipv6 nhrp summary**
7. **show ipv6 nhrp traffic** [**interface****tunnel number**]
8. **show ip nhrp shortcut**
9. **show ip route**
10. **show ipv6 route**
11. **show nhrp debug-condition**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show dmvpn [ipv4 [vrf vrf-name] ipv6 [vrf vrf-name]] [debug-condition [interface tunnel number peer { nbma ip-address network network-mask tunnel ip-address }] [static] [detail] Example: Device# show dmvpn 2001:0db8:1:1::72/64	Displays DMVPN-specific session information.

	Command or Action	Purpose
Step 3	<p>show ipv6 nhrp [dynamic [ipv6-address] incomplete static] [address interface] [brief detail] [purge]</p> <p>Example:</p> <pre>Device# show ipv6 nhrp</pre>	Displays NHRP mapping information.
Step 4	<p>show ipv6 nhrp multicast [ipv4-address interface ipv6-address]</p> <p>Example:</p> <pre>Device# show ipv6 nhrp multicast</pre>	Displays NHRP multicast mapping information.
Step 5	<p>show ip nhrp multicast [nbma-address interface]</p> <p>Example:</p> <pre>Device# show ip nhrp multicast</pre>	Displays NHRP multicast mapping information.
Step 6	<p>show ipv6 nhrp summary</p> <p>Example:</p> <pre>Device# show ipv6 nhrp summary</pre>	Displays NHRP mapping summary information.
Step 7	<p>show ipv6 nhrp traffic [interface tunnel number]</p> <p>Example:</p> <pre>Device# show ipv6 nhrp traffic</pre>	Displays NHRP traffic statistics information.
Step 8	<p>show ip nhrp shortcut</p> <p>Example:</p> <pre>Device# show ip nhrp shortcut</pre>	Displays NHRP shortcut information.
Step 9	<p>show ip route</p> <p>Example:</p> <pre>Device# show ip route</pre>	Displays the current state of the IPv4 routing table.
Step 10	<p>show ipv6 route</p> <p>Example:</p> <pre>Device# show ipv6 route</pre>	Displays the current contents of the IPv6 routing table.
Step 11	<p>show nhrp debug-condition</p> <p>Example:</p> <pre>Device# show nhrp debug-condition</pre>	Displays the NHRP conditional debugging information.

Monitoring and Maintaining DMVPN for IPv6 Configuration and Operation

SUMMARY STEPS

1. **enable**
2. **clear dmvpn session** [**interface tunnel** *number* | **peer** {*ipv4-address* | *fqdn-string* | *ipv6-address*} | **vrf** *vrf-name*] [**static**]
3. **clear ipv6 nhrp** [*ipv6-address* | **counters**]
4. **debug dmvpn** {**all** | **error** | **detail** | **packet**} {**all** | *debug-type*}
5. **debug nhrp** [**cache** | **extension** | **packet** | **rate**]
6. **debug nhrp condition** [**interface tunnel** *number* | **peer** {**nbma** {*ipv4-address* | *fqdn-string* | *ipv6-address*} | **tunnel** {*ip-address* | *ipv6-address*}} | **vrf** *vrf-name*]
7. **debug nhrp error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	clear dmvpn session [interface tunnel <i>number</i> peer { <i>ipv4-address</i> <i>fqdn-string</i> <i>ipv6-address</i> } vrf <i>vrf-name</i>] [static] Example: Device# clear dmvpn session	Clears DMVPN sessions.
Step 3	clear ipv6 nhrp [<i>ipv6-address</i> counters] Example: Device# clear ipv6 nhrp	Clears all dynamic entries from the NHRP cache.
Step 4	debug dmvpn { all error detail packet } { all <i>debug-type</i> } Example: Device# debug dmvpn	Displays debug DMVPN session information.
Step 5	debug nhrp [cache extension packet rate] Example: Device# debug nhrp ipv6	Enables NHRP debugging.
Step 6	debug nhrp condition [interface tunnel <i>number</i> peer { nbma { <i>ipv4-address</i> <i>fqdn-string</i> <i>ipv6-address</i> } tunnel { <i>ip-address</i> <i>ipv6-address</i> }} vrf <i>vrf-name</i>]	Enables NHRP conditional debugging.

	Command or Action	Purpose
	Example: Device# debug nhrp condition	
Step 7	debug nhrp error Example: Device# debug nhrp ipv6 error	Displays NHRP error-level debugging information.

Examples

Sample Output for the debug nhrp Command

The following sample output is from the **debug nhrp** command with the **ipv6** keyword:

```
Device# debug nhrp ipv6
Aug  9 13:13:41.486: NHRP: Attempting to send packet via DEST
- 2001:DB8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: Encapsulation succeeded.
Aug  9 13:13:41.486: NHRP: Tunnel NBMA addr 11.11.11.99
Aug  9 13:13:41.486: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 105
Aug  9 13:13:41.486: src: 2001:DB8:3c4d:0015:0000:0000:1a2f:3d2c/32,
      dst: 2001:DB8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: 105 bytes out Tunnel0
Aug  9 13:13:41.486: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 125
```

Configuration Examples for IPv6 over DMVPN

Example: Configuring an IPsec Profile

```
Device(config)# crypto identity router1

Device(config)# crypto ipsec profile example1
Device(config-crypto-map)# set transform-set example-set
Device(config-crypto-map)# set identity router1

Device(config-crypto-map)# set security-association lifetime seconds 1800

Device(config-crypto-map)# set pfs group14
```

Example: Configuring the Hub for DMVPN

```
Device# configure terminal
Device(config)# interface tunnel 5
```

Example: Configuring the Hub for DMVPN

```

Device(config-if)# ipv6 address 2001:DB8:1:1::72/64
Device(config-if)# ipv6 address fe80::2001 link-local
Device(config-if)# ipv6 mtu 1400
Device(config-if)# ipv6 nhrp authentication examplexx
Device(config-if)# ipv6 nhrp map multicast dynamic
Device(config-if)# ipv6 nhrp network-id 99
Device(config-if)# tunnel source ethernet 0
Device(config-if)# tunnel mode gre multipoint
Device(config-if)# tunnel protection ipsec profile example_profile
Device(config-if)# bandwidth 1200
Device(config-if)# ipv6 nhrp holdtime 3600

```

The following sample output is from the **show dmvpn** command, with the **ipv6** and **detail** keywords, for the hub:

```

Device# show dmvpn ipv6 detail

Legend: Attrib --> 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 10.0.0.3, VRF ""
  Tunnel Src./Dest. addr: 192.169.2.9/MGRE, Tunnel VRF ""
  Protocol/Transport: "multi-GRE/IP", Protect "test_profile"
Type:Hub, Total NBMA Peers (v4/v6): 2
  1.Peer NBMA Address: 192.169.2.10
    Tunnel IPv6 Address: 2001::4
    IPv6 Target Network: 2001::4/128
    # Ent: 2, Status: UP, UpDn Time: 00:01:51, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  2.Peer NBMA Address: 192.169.2.10
    Tunnel IPv6 Address: 2001::4
    IPv6 Target Network: FE80::2/128
    # Ent: 0, Status: UP, UpDn Time: 00:01:51, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  3.Peer NBMA Address: 192.169.2.11
    Tunnel IPv6 Address: 2001::5
    IPv6 Target Network: 2001::5/128
    # Ent: 2, Status: UP, UpDn Time: 00:26:38, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  4.Peer NBMA Address: 192.169.2.11
    Tunnel IPv6 Address: 2001::5
    IPv6 Target Network: FE80::3/128
    # Ent: 0, Status: UP, UpDn Time: 00:26:38, Cache Attrib: D
Pending DMVPN Sessions:

Interface: Tunnell
  IKE SA: local 192.169.2.9/500 remote 192.169.2.10/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 192.169.2.10
  IPSEC FLOW: permit 47 host 192.169.2.9 host 192.169.2.10
    Active SAs: 2, origin: crypto map
  Outbound SPI : 0x BB0ED02, transform : esp-aes esp-sha-hmac
  Socket State: Open

Interface: Tunnell
  IKE SA: local 192.169.2.9/500 remote 192.169.2.11/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 192.169.2.11
  IPSEC FLOW: permit 47 host 192.169.2.9 host 192.169.2.11

```

```

Active SAs: 2, origin: crypto map
Outbound SPI : 0xB79B277B, transform : esp-aes esp-sha-hmac
Socket State: Open

```

Example: Configuring the Spoke for DMVPN

```

Device# configure terminal
Device(config)# crypto ikev2 keyring DMVPN
Device(config)# peer DMVPN
Device(config)# address 0.0.0.0 0.0.0.0
Device(config)# pre-shared-key cisco123
Device(config)# peer DMVPNv6
Device(config)# address ::/0
Device(config)# pre-shared-key cisco123v6
Device(config)# crypto ikev2 profile DMVPN
Device(config)# match identity remote address 0.0.0.0
Device(config)# match identity remote address ::/0
Device(config)# authentication local pre-share
Device(config)# authentication remote pre-share
Device(config)# keyring DMVPN
Device(config)# dpd 30 5 on-demand
Device(config)# crypto ipsec transform-set DMVPN esp-aes esp-sha-hmac
Device(config)# mode transport
Device(config)# crypto ipsec profile DMVPN
Device(config)# set transform-set DMVPN
Device(config)# set ikev2-profile DMVPN
Device(config)# interface tunnel 5

Device(config-if)# bandwidth 1000
Device(config-if)# ip address 10.0.0.11 255.255.255.0
Device(config-if)# ip mtu 1400
Device(config-if)# ip nhrp authentication test
Device(config-if)# ip nhrp network-id 100000
Device(config-if)# ip nhrp nhs 10.0.0.1 nbma 2001:DB8:0:FFFF:1::1 multicast
Device(config-if)# vip nhrp shortcut
Device(config-if)# delay 1000
Device(config-if)# ipv6 address 2001:DB8:0:100::B/64
Device(config-if)# ipv6 mtu 1400
Device(config-if)# ipv6 nd ra mtu suppress
Device(config-if)# no ipv6 redirects
Device(config-if)# ipv6 eigrp 1
Device(config-if)# ipv6 nhrp authentication testv6
Device(config-if)# ipv6 nhrp network-id 100006
Device(config-if)# ipv6 nhrp nhs 2001:DB8:0:100::1 nbma 2001:DB8:0:FFFF:1::1 multicast
Device(config-if)# ipv6 nhrp shortcut
Device(config-if)# tunnel source Ethernet0/0
Device(config-if)# tunnel mode gre multipoint ipv6
Device(config-if)# tunnel key 100000
Device(config-if)# end
.
.

```

The following sample output is from the **show dmvpn** command, with the **ipv6** and **detail** keywords, for the spoke:

```

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
=====

```

Example: Configuring the NHRP Redirect and Shortcut Features on the Hub

```

Interface Tunnell is up/up, Addr. is 10.0.0.1, VRF ""
  Tunnel Src./Dest. addr: 192.169.2.10/MGRE, Tunnel VRF ""
  Protocol/Transport: "multi-GRE/IP", Protect "test_profile"

IPv6 NHS: 2001::6 RE
Type:Spoke, Total NBMA Peers (v4/v6): 1
  1.Peer NBMA Address: 192.169.2.9
    Tunnel IPv6 Address: 2001::6
    IPv6 Target Network: 2001::/112
    # Ent: 2, Status: NHRP, UpDn Time: never, Cache Attrib: S

IPv6 NHS: 2001::6 RE
Type:Unknown, Total NBMA Peers (v4/v6): 1
  2.Peer NBMA Address: 192.169.2.9
    Tunnel IPv6 Address: FE80::1
    IPv6 Target Network: FE80::1/128
    # Ent: 0, Status: UP, UpDn Time: 00:00:24, Cache Attrib: D

Pending DMVPN Sessions:

Interface: Tunnell
  IKE SA: local 192.169.2.10/500 remote 192.169.2.9/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phasel_id: 192.169.2.9
  IPSEC FLOW: permit 47 host 192.169.2.10 host 192.169.2.9
    Active SAs: 2, origin: crypto map
  Outbound SPI : 0x6F75C431, transform : esp-aes esp-sha-hmac
  Socket State: Open

```

Example: Configuring the NHRP Redirect and Shortcut Features on the Hub

```

Device(config)# interface tunnel 5
Device(config-if)# ipv6 address 2001:DB8:1:1::72/64

Device(config-if)# ipv6 nhrp redirect

Device(config-if)# ipv6 nhrp shortcut

```

Example: Configuring NHRP on the Hub and Spoke

Hub

```

Device# show ipv6 nhrp

2001::4/128 via 2001::4
  Tunnell created 00:02:40, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.10
2001::5/128 via 2001::5
  Tunnell created 00:02:37, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.11
FE80::2/128 via 2001::4
  Tunnell created 00:02:40, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.10

```



```
FE80::3/128 via 2001::5
  Tunnel1 created 00:02:37, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.11
```

Spoke

```
Device# show ipv6 nhrp

2001::8/128
  Tunnel1 created 00:00:13, expire 00:02:51
  Type: incomplete, Flags: negative
  Cache hits: 2
2001::/112 via 2001::6
  Tunnel1 created 00:01:16, never expire
  Type: static, Flags: used
  NBMA address: 192.169.2.9
FE80::1/128 via FE80::1
  Tunnel1 created 00:01:15, expire 00:00:43
  Type: dynamic, Flags:
  NBMA address: 192.169.2.9
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Dynamic Multipoint VPN	<i>Dynamic Multipoint VPN Configuration Guide</i>
IPv6 commands	<i>IPv6 Command Reference</i>
Cisco IOS IPv6 features	IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

Technical Assistance

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

Feature Information for IPv6 over 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 IPv6 over DMVPN

Feature Name	Releases	Feature Information
IPv6 over DMVPN		<p>The 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 the Next Hop Resolution Protocol (NHRP). In Dynamic Multipoint Virtual Private Network (DMVPN) for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable.</p> <p>The following commands were introduced or modified: clear dmvpn session, clear ipv6 nhrp, crypto ipsec profile, debug dmvpn, debug dmvpn condition, debug nhrp condition, debug nhrp error, ipv6 nhrp authentication, ipv6 nhrp holdtime, ipv6 nhrp interest, ipv6 nhrp map, ipv6 nhrp map multicast, ipv6 nhrp map multicast dynamic, ipv6 nhrp max-send, ipv6 nhrp network-id, ipv6 nhrp nhs, ipv6 nhrp record, ipv6 nhrp redirect, ipv6 nhrp registration, ipv6 nhrp responder, ipv6 nhrp server-only, ipv6 nhrp shortcut, ipv6 nhrp trigger-svc, ipv6 nhrp use, set pfs, set security-association lifetime, set transform-set, show dmvpn, show ipv6 nhrp, show ipv6 nhrp multicast, show ipv6 nhrp nhs, show ipv6 nhrp summary, show ipv6 nhrp traffic.</p>
IPv6 Transport for DMVPN		<p>The IPv6 transport for DMVPN feature builds IPv6 WAN-side capability into NHRP tunnels and the underlying IPsec encryption, and enables IPv6 to transport payloads on the Internet.</p> <p>The IPv6 transport for DMVPN feature is enabled by default.</p>



CHAPTER 4

DMVPN Configuration Using FQDN

The DMVPN Configuration Using FQDN feature enables next hop clients (NHCs) to register with the next hop server (NHS).

This feature allows you to configure a fully qualified domain name (FQDN) for the nonbroadcast multiple access network (NBMA) address of the hub (NHS) on the spokes (NHCs). The spokes resolve the FQDN to IP address using the DNS service and get registered with the hub using the newly resolved address. This allows spokes to dynamically locate the IP address of the hub using FQDN.

With this feature, spokes need not configure the protocol address of the hub. Spokes learn the protocol address of the hub dynamically from the NHRP registration reply of the hub. According to RFC 2332, the hub to which the NHRP registration was sent responds with its own protocol address in the NHRP registration reply and hence the spokes learn the protocol address of the hub from the NHRP registration reply packet.

In Cisco IOS Release 15.1(2)T and earlier releases, in Dynamic Multipoint VPN (DMVPN), NHS NBMA addresses were configured with either IPv4 or IPv6 addresses. Because NHS was configured to receive a dynamic NBMA address, it was difficult for NHCs to get the updated NBMA address and register with the NHS. This limitation is addressed with the DMVPN Configuration Using FQDN feature. This feature allows NHC to use an FQDN instead of an IP address to configure NBMA and register with the NHS dynamically.

- [Finding Feature Information, on page 71](#)
- [Prerequisites for DMVPN Configuration Using FQDN, on page 72](#)
- [Restrictions for DMVPN Configuration Using FQDN, on page 72](#)
- [Information About DMVPN Configuration Using FQDN, on page 72](#)
- [How to Configure DMVPN Configuration Using FQDN, on page 73](#)
- [Configuration Examples for DMVPN Configuration Using FQDN, on page 78](#)
- [Additional References, on page 80](#)
- [Feature Information for DMVPN Configuration Using FQDN, on page 81](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfnng.cisco.com/>. An account on Cisco.com is not required.

Prerequisites for DMVPN Configuration Using FQDN

Cisco IOS Domain Name System (DNS) client must be available on the spoke.

Restrictions for DMVPN Configuration Using FQDN

If the NBMA IP address resolved from the FQDN is not mapped to an NHS configured with the protocol address, the spoke cannot register with the hub.

Information About DMVPN Configuration Using FQDN

DNS Functionality

A Domain Name System (DNS) client communicates with a DNS server to translate a hostname to an IP address.

The intermediate DNS server or the DNS client on the route enters the FQDN DNS reply from the DNS server into the cache for a lifetime. If the DNS client receives another query before the lifetime expires, the DNS client uses the entry information from the cache. If the cache expires, the DNS client queries the DNS server. If the NBMA address of the NHS changes frequently, the DNS entry lifetime must be short, otherwise the spokes may take some time before they start using the new NBMA address for the NHS.

DNS Server Deployment Scenarios

A DNS server can be located either in a hub network or outside a hub and spoke network.

Following are the four DNS server load balancing models:

- Round robin--Each DNS request is assigned an IP address sequentially from the list of IP addresses configured for an FQDN.
- Weighted round robin--This is similar to round-robin load balancing except that the IP addresses are assigned weights and nodes, where higher weights can take more load or traffic.
- Geography or network--Geography-based load balancing allows the requests to be directed to the optimal node that is geographically the nearest or the most efficient to the requester.
- Failover--Failover load balancing sends all requests to a single host until the load balancer determines a particular node to be no longer available. It then directs traffic to the next node available in the list.

How to Configure DMVPN Configuration Using FQDN

Configuring a DNS Server on a Spoke

Perform this task to configure a DNS server on a spoke. You must perform this task only if you want to resolve FQDN using an external DNS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip name-server** *ip-address*
4. **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.
Step 3	ip name-server <i>ip-address</i> Example: Router(config)# ip name-server 192.0.2.1	Configures a DNS server on a spoke.
Step 4	exit Example: Router(config)# exit	Exits global configuration mode.

Configuring a DNS Server

Perform this task to configure a DNS server. You must perform the configuration on a DNS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ip dns server**
4. **ip host** *hostname ip-address*
5. **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.
Step 3	ip dns server Example: Router(config)# ip dns server	Enables a DNS server.
Step 4	ip host <i>hostname ip-address</i> Example: Router(config)# ip host host1.example.com 192.0.2.2	Maps a FQDN (hostname) with the IP address in the DNS hostname cache for a DNS view. Note Configure the ip host command on a DNS server if you have configured a DNS server on the spoke and configure the command on the spoke if you have not configured a DNS server on the spoke. See the Configuring a DNS Server on a Spoke task.
Step 5	exit Example: Router(config)# exit	Exits global configuration mode.

Configuring an FQDN with a Protocol Address

Perform this task to configure an FQDN with a protocol address. You must know the protocol address of the NHS while you are configuring the FQDN. This configuration registers spoke to a hub using NBMA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*

4. **ip nhrp nhs** *nhs-address* [**nbma** {*nbma-address* | *FQDN-string*}] [**multicast**] [**priority** *value*] [**cluster** *number*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • 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 1</pre>	Enters interface configuration mode.
Step 4	ip nhrp nhs <i>nhs-address</i> [nbma { <i>nbma-address</i> <i>FQDN-string</i> }] [multicast] [priority <i>value</i>] [cluster <i>number</i>] Example: <pre>Router(config-if)# ip nhrp nhs 192.0.2.1 nbma examplehub.example1.com multicast</pre>	Registers a spoke to a hub. <ul style="list-style-type: none"> • You can configure the command in the following two ways: <ul style="list-style-type: none"> • ip nhrp nhs <i>protocol-ipaddress</i> nbma <i>FQDN-string</i>--Use this command to register spoke to a hub using the FQDN string. • ip nhrp nhs <i>protocol-ipaddress</i> nbma <i>nbma-ipaddress</i>--Use this command to register spoke to a hub using the NHS NBMA IP address. <p>Note You can use the ipv6 nhrp nhs <i>protocol-ipaddress</i> [nbma {<i>nhs-ipaddress</i> <i>FQDN-string</i>}] [multicast] [priority <i>value</i>] [cluster <i>number</i>] command for registering IPv6 address.</p>
Step 5	end Example: <pre>Router(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a FQDN Without an NHS Protocol Address

Perform this task to configure an FQDN without an NHS protocol address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip nhrp nhs dynamic nbma** {*nbma-address* | *FQDN-string*} [**multicast**] [**priority value**] [**cluster value**]
5. **end**

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.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 1	Enters interface configuration mode.
Step 4	ip nhrp nhs dynamic nbma { <i>nbma-address</i> <i>FQDN-string</i> } [multicast] [priority value] [cluster value] Example: Router(config-if)# ip nhrp nhs dynamic nbma examplehub.example1.com	Registers a spoke to a hub. <ul style="list-style-type: none"> • The NHS protocol address is dynamically fetched by the spoke. You can configure the command in the following two ways: <ul style="list-style-type: none"> • ip nhrp nhs dynamic nbma <i>FQDN-string</i>--Use this command to register a spoke to a hub using the FQDN string. • ip nhrp nhs dynamic nbma <i>nbma-address</i>--Use this command to register a spoke to a hub using the NHS NBMA IP address. <p>Note You can use the ipv6 nhrp nhs dynamic nbma {<i>nbma-address</i> <i>FQDN-string</i>} [multicast] [priority value] [cluster value] command for registering IPv6 address.</p>
Step 5	end Example: Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying DMVPN FQDN Configuration

This task shows how to display information to verify DMVPN FQDN configuration. The following **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dmvpn**
3. **show ip nhrp nhs**
4. **show running-config interface tunnel** *tunnel-number*
5. **show ip nhrp multicast**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router# enable
```

Step 2 **show dmvpn**

Displays DMVPN-specific session information.

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
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====
Interface: Tunnell1, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
      1      192.0.2.1      192.0.2.2 UP 00:00:12      S
      (h1.cisco.com)
```

Step 3 **show ip nhrp nhs**

Displays the status of the NHS.

Example:

```
Router# show ip nhrp nhs
IPv4 Registration Timer: 10 seconds
Legend: E=Expecting replies, R=Responding, W=Waiting
Tunnell1:
192.0.2.1 RE NBMA Address: 192.0.2.2 (h1.cisco.com) priority = 0 cluster = 0
```

Step 4 **show running-config interface tunnel** *tunnel-number*

Displays the contents of the current running configuration file or the tunnel interface configuration.

Example:

```
Router# show running-config interface tunnel 1
Building configuration...
Current configuration : 462 bytes
!
interface Tunnell
 ip address 192.0.2.1 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp authentication testing
 ip nhrp group spoke_group2
 ip nhrp network-id 123
 ip nhrp holdtime 150
 ip nhrp nhs dynamic nbma h1.cisco.com multicast
 ip nhrp registration unique
 ip nhrp registration timeout 10
 ip nhrp shortcut
 no ip route-cache cef
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel key 1001
 tunnel protection ipsec profile DMVPN
end
```

Step 5 show ip nhrp multicast

Displays NHRP multicast mapping information.

Example:

```
Route# show ip nhrp multicast
I/F      NBMA address
Tunnell  192.0.2.1  Flags: nhs
```

Configuration Examples for DMVPN Configuration Using FQDN

Example Configuring a Local DNS Server

The following example shows how to configure a local DNS server:

```
enable
configure terminal
 ip host host1.example.com 192.0.2.2
```

Example Configuring an External DNS Server

The following example shows how to configure an external DNS server:

On a spoke

```
enable
configure terminal
ip name-server 192.0.2.1
```

On a DNS Server

```
enable
configure terminal
ip dns server
ip host host1.example.com 192.0.2.2
```

Example Configuring NHS with a Protocol Address and an NBMA Address

The following example shows how to configure NHS with a protocol address and an NBMA address:

```
enable
configure terminal
interface tunnel 1
ip nhrp nhs 192.0.2.1 nbma 209.165.200.225
```

Example Configuring NHS with a Protocol Address and an FQDN

The following example shows how to configure NHS with a protocol address and an FQDN:

```
enable
configure terminal
interface tunnel 1
ip nhrp nhs 192.0.2.1 nbma examplehub.example1.com
```

Example Configuring NHS Without a Protocol Address and with an NBMA Address

The following example shows how to configure NHS without a protocol address and with an NBMA address:

```
enable
configure terminal
interface tunnel 1
ip nhrp nhs dynamic nbma 192.0.2.1
```

Example Configuring NHS Without a Protocol Address and with an FQDN

The following example shows how to configure NHS without a protocol address and with an FQDN:

```
enable
configure terminal
interface tunnel 1
ip nhrp nhs dynamic nbma examplehub.example1.com
```

Additional References

Related Documents

Related Topic	Document Title
DMVPN complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS Security Command Reference</i>

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
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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 2332	<i>NBMA Next Hop Resolution Protocol (NHRP)</i>

Technical Assistance

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

Feature Information for DMVPN Configuration Using FQDN

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 DMVPN Configuration Using FQDN

Feature Name	Releases	Feature Information
DMVPN Configuration Using FQDN		<p>The DMVPN Configuration Using FQDN feature enables the NHC to register with the NHS. It uses the NHRP without using the protocol address of the NHS.</p> <p>The following commands were introduced or modified: clear dmvpn session, debug nhrp condition, ip nhrp nhs, and ipv6 nhrp nhs.</p>



CHAPTER 5

DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

The DMVPN-Tunnel Health Monitoring and Recovery (Backup NHS) feature allows you to control the number of connections to the Dynamic Multipoint Virtual Private Network (DMVPN) hub and allows you to switch to alternate hubs in case of a connection failure to the primary hubs.

The recovery mechanism provided by the DMVPN-Tunnel Health Monitoring and Recovery (Backup NHS) feature allows spokes to recover from a failed spoke-to-hub tunnel path by replacing the tunnel by another active spoke-to-hub tunnel. Spokes can select the next hop server (NHS) [hub] from a list of NHSs configured on the spoke. You can configure priority values to the NHSs that control the order in which spokes select the NHS.

- [Finding Feature Information, on page 83](#)
- [Information About DMVPN-Tunnel Health Monitoring and Recovery Backup NHS, on page 84](#)
- [How to Configure DMVPN-Tunnel Health Monitoring and Recovery Backup NHS, on page 89](#)
- [Configuration Examples for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS, on page 93](#)
- [Additional References, on page 94](#)
- [Feature Information for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS, on page 95](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Information About DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

NHS States

An NHS attains different states while associating with the hubs to form a spoke-to-hub tunnel. The table below describes different NHS states.

Table 4: NHS States

State	Description
DOWN	NHS is waiting to get scheduled.
PROBE	NHS is declared as “DOWN” but it is still actively probed by the spoke to bring it “UP”.
UP	NHS is associated with a spoke to establish a tunnel.

NHS Priorities

NHS priority is a numerical value assigned to a hub that controls the order in which spokes select hubs to establish a spoke-to-hub tunnel. The priority value ranges from 0 to 255, where 0 is the highest and 255 is the lowest priority.

You can assign hub priorities in the following ways:

- Unique priorities to all NHS.
- Same priority level to a group of NHS.
- Unspecified priority (value 0) for an NHS, a group of NHSs, or all NHSs.

NHS Clusterless Model

NHS clusterless model is a model where you assign the priority values to the NHSs and do not place the NHSs into any group. NHS clusterless model groups all NHSs to a default group and maintains redundant connections based on the maximum NHS connections configured. Maximum NHS connections is the number of NHS connections in a cluster that must be active at any point in time. The valid range for maximum NHS connections is from 0 to 255.

Priority values are assigned to the hubs to control the order in which the spokes select hubs to establish the spoke-to-hub tunnel. However, assigning these priorities in a clusterless model has certain limitations.

The table below provides an example of limitations for assigning priorities in a clusterless model.

Table 5: Limitations of Clusterless Mode

Maximum Number of Connections = 3	
-----------------------------------	--

NHS	NHS Priority	Scenario 1	Scenario 2
NHS A1	1	UP	UP
NHS B1	1	UP	PROBE
NHS C1	1	UP	UP
NHS A2	2	DOWN	UP
NHS B2	2	DOWN	DOWN
NHS C2	2	DOWN	DOWN

Consider a scenario with three data centers A, B, and C. Each data center consists of two NHSs: NHSs A1 and A2 comprise one data center, NHS B1 and B2 another, and C1 and C3 another.

Although two NHSs are available for each data center, the spoke is connected to only one NHS of each data center at any point in time. Hence, the maximum connection value is set to 3. That is, three spoke-to-hub tunnels are established. If any one NHS, for example, NHS B1, becomes inactive, the spoke-to-hub tunnel associated with NHS B1 goes down. Based on the priority model, NHS A2 has the next priority value and the next available NHS in the queue, so it forms the spoke-to-hub tunnel and goes up. However, this does not meet the requirement that a hub from data center B be associated with the spoke to form a tunnel. Hence, no connection is made to data center B.

This problem can be addressed by placing NHSs into different groups. Each group can be configured with a group specific maximum connection value. NHSs that are not assigned to any groups belong to the default group.

NHS Clusters

The table below presents an example of cluster functionality. NHSs corresponding to different data centers are grouped to form clusters. NHS A1 and NHS A2 with priority 1 and 2, respectively, are grouped as cluster1, NHS B1 and NHS B2 with priority 1 and 2, respectively, are grouped as cluster2, and NHS C1 and NHS C2 with priority 1 and 2, respectively, are grouped as cluster3. NHS 7, NHS 8, and NHS 9 are part of the default cluster. The maximum cluster value is set to 1 for each cluster so that at least one spoke-to-hub tunnel is continuously established with all the four clusters.

In scenario 1, NHS A1, NHS B1, and NHS C1 with the highest priority in each cluster are in the UP state. In scenario 2, the connection between the spoke and NHS A1 breaks, and a connection is established between the spoke and NHS A2 (hub from the same cluster). NHS A1 with the highest priority attains the PROBE state. In this way, at any point in time a connection is established to all the three data centers.

Table 6: Cluster Functionality

NHS	NHS Priority	Cluster	Maximum Number of Connections	Scenario 1	Scenario 2
NHS A1	1	1	1	UP	PROBE
NHS A2	2			DOWN	UP

NHS	NHS Priority	Cluster	Maximum Number of Connections	Scenario	Scenario
				1	2
NHS B1	1	2	1	UP	UP
NHS B2	2			DOWN	DOWN
NHS C1	1	3	1	UP	UP
NHS C2	2			DOWN	DOWN
NHS 7	1	Default	2	UP	DOWN
NHS 8	2			UP	UP
NHS 9	0			PROBE	UP

NHS Fallback Time

Fallback time is the time that the spoke waits for the NHS to become active before detaching itself from an NHS with a lower priority and connecting to the NHS with the highest priority to form a spoke-to-hub tunnel. Fallback time helps in avoiding excessive flaps.

The table below shows how the spoke flaps from one NHS to another excessively when the fallback time is not configured on the spoke. Five NHSs having different priorities are available to connect to the spoke to form a spoke-to-hub tunnel. All these NHSs belong to the default cluster. The maximum number of connection is one.

Table 7: NHS Behavior when Fallback Time is not Configured

NHS	NHS Priority	Cluster	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
NHS 1	1	Default	PROBE	PROBE	PROBE	PROBE	UP
NHS 2	2	Default	PROBE	PROBE	PROBE	UP	DOWN
NHS 3	3	Default	PROBE	PROBE	UP	DOWN	DOWN
NHS 4	4	Default	PROBE	UP	DOWN	DOWN	DOWN
NHS 5	5	Default	UP	DOWN	DOWN	DOWN	DOWN

In scenario 1, NHS 5 with the lowest priority value is connected to the spoke to form a tunnel. All the other NHSs having higher priorities than NHS 5 are in the PROBE state.

In scenario 2, when NHS 4 becomes active, the spoke breaks connection with the existing tunnel and establishes a new connection with NHS 4. In scenario 3 and scenario 4, the spoke breaks the existing connections as soon as an NHS with a higher priority becomes active and establishes a new tunnel. In scenario 5, as the NHS with the highest priority (NHS 1) becomes active, the spoke connects to it to form a tunnel and continues with it until the NHS becomes inactive. Because NHS 1 is having the highest priority, no other NHS is in the PROBE state.

The table below shows how to avoid the excessive flapping by configuring the fallback time. The maximum number of connection is one. A fallback time period of 30 seconds is configured on the spoke. In scenario 2, when an NHS with a higher priority than the NHS associated with the spoke becomes active, the spoke does not break the existing tunnel connection until the fallback time. Hence, although NHS 4 becomes active, it does not form a tunnel and attain the UP state. NHS 4 remains active but does not form a tunnel until the fallback time elapses. Once the fallback time elapses, the spoke connects to the NHS having the highest priority among the active NHSs.

This way, the flaps that occur as soon as an NHS of higher priority becomes active are avoided.

Table 8: NHS Behavior when Fallback Time is Configured

NHS	NHS Priority	Cluster	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
NHS 1	1	Default	PROBE	PROBE	PROBE	UP-hold	UP
NHS 2	2	Default	PROBE	PROBE	UP-hold	UP-hold	DOWN
NHS 3	3	Default	PROBE	UP-hold	UP-hold	UP-hold	DOWN
NHS 4	4	Default	UP-hold	UP-hold	UP-hold	UP-hold	DOWN
NHS 5	5	Default	UP	UP	UP	UP	DOWN

NHS Recovery Process

NHS recovery is a process of establishing an alternative spoke-to-hub tunnel when the existing tunnel becomes inactive, and connecting to the preferred hub upon recovery.

The following sections explain NHS recovery:

Alternative Spoke to Hub NHS Tunnel

When a spoke-to-hub tunnel fails it must be backed up with a new spoke-to-hub tunnel. The new NHS is picked from the same cluster to which the failed hub belonged. This ensures that the required number of spoke-to-hub tunnels are always present although one or more tunnel paths are unavailable.

The table below presents an example of NHS backup functionality.

Table 9: NHS Backup Functionality

NHS	NHS Priority	Cluster	Maximum Number of Connections	Scenario 1	Scenario 2	Scenario 3
NHS A1	1	1	1	UP	PROBE	PROBE
NHS A2	2			DOWN	UP	DOWN
NHS A3	2			DOWN	DOWN	UP
NHS A4	2			DOWN	DOWN	DOWN

NHS	NHS Priority	Cluster	Maximum Number of Connections	Scenario	Scenario	Scenario
				1	2	3
NHS B1	1	3	1	UP	PROBE	PROBE
NHS B2	2			DOWN	UP	DOWN
NHS B3	2			DOWN	DOWN	UP
NHS B4	2			DOWN	DOWN	DOWN
NHS 9	Default	Default	1	UP	UP	DOWN
NHS 10				DOWN	DOWN	UP

Four NHSs belonging to cluster 1 and cluster 3 and two NHSs belonging to the default cluster are available for setting up spoke-to-hub tunnels. All NHSs have different priorities. The maximum number of connections is set to 1 for all the three clusters. That is, at any point in time, at least one NHS from each cluster must be connected to the spoke to form a tunnel.

In scenario 1, NHS A1 from cluster 1, NHS B1 from cluster 3, and NHS 9 from the default cluster are UP. They establish a contact with the spoke to form different spoke-to-hub tunnels. In scenario 2, NHS A1 and NHS B1 with the highest priority in their respective clusters become inactive. Hence a tunnel is established from the spoke to NHS A2 and NHS B2, which have the next highest priority values. However, the spoke continues to probe NHS A1 and NHS B1 because they have the highest priority. Hence, NHS A1 and NHS B1 remain in the PROBE state.

In scenario 3, NHS A2, NHS B2, and NHS 9 become inactive. The spoke checks if the NHSs in PROBE state have turned active. If yes, then the spoke establishes a connection to the NHS that has turned active. However, as shown in scenario 3, because none of the NHSs in the PROBE state is active, the spoke connects to NHS A3 of cluster 1 and NHS B3 of cluster 2. NHS A1 and NHS B1 continue to be in the PROBE state until they associate themselves with the spoke to form a tunnel and attain the UP state.

Returning to Preferred NHS Tunnel upon Recovery

When a spoke-to-hub tunnel fails, a backup tunnel is established using an NHS having the next higher priority value. Even though the tunnel is established with an NHS of lower priority, the spoke continuously probes the NHS having the highest priority value. Once the NHS having the highest priority value becomes active, the spoke establishes a tunnel with the NHS and hence the NHS attains the UP state.

The table below presents NHS recovery functionality. Four NHSs belonging to cluster 1 and cluster 3 and two NHSs belonging to the default cluster are available for setting up spoke-to-hub tunnels. All NHSes have different priorities. The maximum connection value is set to 1. In scenario 1, NHS A4, NHS B4, and NHS 10 with the least priority in their respective clusters associate with the spoke in establishing a tunnel. The spoke continues to probe NHSs of higher priority to establish a connection with the NHS having the highest priority value. Hence, in scenario 1, NHSs having the highest priority value in their respective clusters are in the PROBE state. In scenario 2, NHS A1 is ACTIVE, forms a tunnel with the spoke, and attains the UP state. Because NHS A1 has the highest priority, the spoke does not probe any other NHS in the cluster. Hence, all the other NHSs in cluster1 are in the DOWN state.

When the connection with NHS B4 breaks, the spoke connects to NHS B3, which has the next higher priority value, because NHS B1 of cluster 3 is not active. In scenario 3, NHS A1 continues to be in the UP state and NHS B1 with the highest priority in cluster 2 becomes active, forms a tunnel, and attains the UP state. Hence,

no other NHSs in cluster 2 are in the PROBE state. However, because NHS 10 having the lowest priority value in the default cluster is in the UP state, the spoke continues to probe NHS 9 having the highest priority in the cluster.

In scenario 4, NHS A1 and NHS B1 continue to be in the UP state and NHS 9 having the highest priority in the default cluster attains the UP state. Hence, because the spoke is associated with the NHSs having the highest priority in all the clusters, none of the NHSs are in the PROBE state.

Table 10: NHS Recovery Functionality

NHS	NHS Priority	Cluster	Maximum Number of Connections	Scenario 1	Scenario 2	Scenario 3	Scenario 4
NHS A1	1	1	1	PROBE	UP	UP	UP
NHS A2	2			DOWN	DOWN	DOWN	DOWN
NHS A3	2			DOWN	DOWN	DOWN	DOWN
NHS A4	2			UP	DOWN	DOWN	DOWN
NHS B1	1	3	1	PROBE	PROBE	UP	UP
NHS B2	10			PROBE	DOWN	DOWN	DOWN
NHS B3	10			PROBE	UP	DOWN	DOWN
NHS B4	30			UP	DOWN	DOWN	DOWN
NHS 9	Default	Default	1	PROBE	PROBE	PROBE	UP
NHS 10	100			UP	UP	UP	DOWN

How to Configure DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

Configuring the Maximum Number of Connections for an NHS Cluster

Perform this task to configure the desired maximum number of connections for an NHS cluster.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel *number***
4. **ip nhrp nhs cluster *cluster-number* max-connections *value***

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 1	Enters interface configuration mode.
Step 4	ip nhrp nhs cluster <i>cluster-number</i> max-connections <i>value</i> Example: Router(config-if)# ip nhrp nhs cluster 5 max-connections 100	Configures the desired maximum number of connections. Note Use the ipv6 nhrp nhs cluster <i>cluster-number</i> max-connections <i>value</i> command for IPv6 configuration.

Configuring NHS Fallback Time

Perform this task to configure NHS fallback time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel *number***
4. **ip nhrp nhs fallback *fallback-time***

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Router# configure terminal	
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 1	Enters interface configuration mode.
Step 4	ip nhrp nhs fallback <i>fallback-time</i> Example: Router(config-if)# ip nhrp nhs fallback 25	Configures NHS fallback time. Note Use the ipv6 nhrp nhs fallback <i>fallback-time</i> command for IPv6 configuration.

Configuring NHS Priority and Group Values

Perform this task to configure NHS priority and group values.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip nhrp nhs** *nhs-address* **priority** *nhs-priority* **cluster** *cluster-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 1	Enters interface configuration mode.
Step 4	ip nhrp nhs <i>nhs-address</i> priority <i>nhs-priority</i> cluster <i>cluster-number</i> Example:	Configures the desired priority and cluster values. Note Use the ipv6 nhrp nhs <i>nhs-address</i> priority <i>nhs-priority</i> cluster <i>cluster-number</i> command for IPv6 configuration.

	Command or Action	Purpose
	Router(config-if)# ip nhrp nhs 172.0.2.1 priority 1 cluster 2	

Verifying the DMVPN-Tunnel Health Monitoring and Recovery Backup NHS Feature

Perform this task to display information and verify DMVPN-Tunnel Health Monitoring and Recovery (Backup NHS) feature configuration. You can enter these **show** commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ip nhrp nhs**
3. **show ip nhrp nhs redundancy**
4. **show ipv6 nhrp nhs**
5. **show ipv6 nhrp nhs redundancy**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router# enable
```

Step 2 **show ip nhrp nhs**

Displays NHRP NHS information.

Example:

```
Router# show ip nhrp nhs
Legend: E=Expecting replies, R=Responding, W=Waiting
Tunnel0:
10.0.0.1 RE priority = 0 cluster = 0
```

Step 3 **show ip nhrp nhs redundancy**

Displays NHRP NHS recovery information.

Example:

```
Router# show ip nhrp nhs redundancy
Legend: E=Expecting replies, R=Responding, W=Waiting
No.  Interface  Cluster  NHS           Priority  Cur-State  Cur-Queue  Prev-State  Prev-Queue
1    Tunnel0     0        10.0.0.253    3         RE         Running    E           Running
2    Tunnel0     0        10.0.0.252    2         RE         Running    E           Running
3    Tunnel0     0        10.0.0.251    1         RE         Running    E           Running
```

No.	Interface	Cluster	Status	Max-Con	Total-NHS	Responding	Expecting	Waiting	Fallback
1	Tunnel0	0	Enable	3	3	3	0	0	0

Step 4 `show ipv6 nhrp nhs`

Displays IPv6, specific NHRP NHS information.

Example:

```
Router# show ipv6 nhrp nhs
Legend: E=Expecting replies, R=Responding, W=Waiting
Tunnel0:
2001::101 RE priority = 1 cluster = 5
```

Step 5 `show ipv6 nhrp nhs redundancy`

Displays IPv6, specific NHRP NHS recovery information.

Example:

```
Router# show ipv6 nhrp nhs redundancy
Legend: E=Expecting replies, R=Responding, W=Waiting
No. Interface Cluster NHS Priority Cur-State Cur-Queue Prev-State Prev-Queue
1 Tunnel0 5 2001::101 1 E Running RE Running
No. Interface Cluster Status Max-Con Total-NHS Responding Expecting Waiting Fallback
1 Tunnel0 5 Disable Not Set 1 0 1 0 0
```

Configuration Examples for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

Example Configuring Maximum Connections for an NHS Cluster

The following example shows how to configure a “max-connections” value of 3 for three NHSs that belong to cluster 0:

```
interface tunnel 0
 bandwidth 1000
 ip address 10.0.0.1 255.0.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast 172.0.2.1
 ip nhrp map 10.0.0.253 172.0.2.1
 ip nhrp map multicast 172.0.2.2
 ip nhrp map 10.0.0.251 172.0.2.2
 ip nhrp map multicast 172.0.2.3
 ip nhrp map 10.0.0.252 172.0.2.3
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.252 priority 2
 ip nhrp nhs 10.0.0.251 priority 1
 ip nhrp nhs 10.0.0.253 priority 3
 ip nhrp nhs cluster 0 max-connections 3
```

```

ip nhrp shortcut
delay 100
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel key 100000
tunnel protection ipsec profile vpnprof
!
!

```

Example Configuring NHS Fallback Time

The following example shows how to configure NHS fallback time to 25 seconds:

```

configure terminal
interface tunnel 1
 ip nhrp nhs fallback 25

```

Example Configuring NHS Priority and Group Values

The following example shows how to group NHSs under different clusters and then assign different maximum connection values to the clusters:

```

Configure terminal
interface tunnel 0
 ip nhrp nhs 10.0.0.251 priority 1 cluster 1
 ip nhrp map 10.0.0.251 192.0.2.4
 ip nhrp map multicast 192.0.2.4
end
configure terminal
interface tunnel 0
 ip nhrp nhs 10.0.0.252 priority 2 cluster 2
 ip nhrp map 10.0.0.252 192.0.2.5
 ip nhrp map multicast 192.0.2.5
end
configure terminal
interface tunnel 0
 ip nhrp nhs 10.0.0.253 priority 3 cluster 3
 ip nhrp map 10.0.0.253 192.0.2.6
 ip nhrp map multicast 192.0.2.6
end
configure terminal
interface tunnel 0
 ip nhrp nhs cluster 1 max 1
 ip nhrp nhs cluster 2 max 1
 ip nhrp nhs cluster 3 max 1
end

```

Additional References

Related Documents

Related Topic	Document Title

Related Topic	Document Title
DMVPN complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS Security Command Reference</i>

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

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 DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

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 11: Feature Information for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

Feature Name	Releases	Feature Information
DMVPN-Tunnel Health Monitoring and Recovery (Backup NHS)		<p>The DMVPN-Tunnel Health Monitoring and Recovery (Backup NHS) feature allows you to control the number of connections to the DMVPN hub and allows you to switch to alternate hubs in case of connection failure to primary hubs.</p> <p>The following commands were introduced or modified: ip nhrp nhs, ipv6 nhrp nhs, show ip nhrp nhs, show ipv6 nhrp nhs.</p>



CHAPTER 6

DMVPN Tunnel Health Monitoring and Recovery

The Dynamic Multipoint VPN Tunnel Health Monitoring and Recovery feature enhances the ability of the system to monitor and report Dynamic Multipoint VPN (DMVPN) events. It includes support for Simple Network Management Protocol (SNMP) Next Hop Resolution Protocol (NHRP) notifications for critical DMVPN events and support for DMVPN syslog messages. It also enables the system to control the state of the tunnel interface based on the health of the DMVPN tunnels.

- [Finding Feature Information, on page 97](#)
- [Prerequisites for DMVPN Tunnel Health Monitoring and Recovery, on page 97](#)
- [Restrictions for DMVPN Tunnel Health Monitoring and Recovery, on page 98](#)
- [Information About DMVPN Tunnel Health Monitoring and Recovery, on page 98](#)
- [How to Configure DMVPN Tunnel Health Monitoring and Recovery, on page 101](#)
- [Configuration Examples for DMVPN Tunnel Health Monitoring and Recovery, on page 103](#)
- [Additional References for DMVPN Tunnel Health Monitoring and Recovery, on page 104](#)
- [Feature Information for DMVPN Tunnel Health Monitoring and Recovery, on page 105](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Prerequisites for DMVPN Tunnel Health Monitoring and Recovery

SNMP NHRP notifications

- SNMP is enabled in the system.
- Generic SNMP configurations for Get and Set operations and for notifications are implemented in the system.

- All relevant NHRP traps are enabled.

Restrictions for DMVPN Tunnel Health Monitoring and Recovery

MIB SNMP

- SNMP SET UNDO is not supported.
- The MIB Persistence feature that enables the MIB-SNMP data to persist across reloads is not supported. However, a virtual persistence for the MIB notification control object happens, because that information is also captured via the configuration command line interface (CLI).
- Notifications and syslogs are not virtual routing and forwarding (VRF)-aware.
- The Rate Limit Exceeded notification does not differentiate between the IPv4 or IPv6 protocol type.

Interface State Control

- Interface state control can be configured on leaf spoke nodes only.
- Interface state control supports IPv4 only.

Information About DMVPN Tunnel Health Monitoring and Recovery

NHRP Extension MIB

The NHRP Extension MIB module comprises objects that maintain redirect-related statistics for both clients and servers, and for the following SNMP notifications for critical DMVPN events:

- A spoke perceives that a hub has gone down. This can occur even if the spoke was not previously registered with the hub.
- A spoke successfully registers with a hub.
- A hub perceives that a spoke has gone down.
- A hub perceives that a spoke has come up.
- A spoke or hub perceives that another NHRP peer, not related by an NHRP registration, has gone down. For example, a spoke-spoke tunnel goes down.
- A spoke or hub perceives that another NHRP peer, not related by an NHRP registration, has come up. For example, a spoke-spoke tunnel comes up.
- The rate limit set for NHRP packets on the interface is exceeded.

The agent implementation of the MIB provides a means to enable and disable specific traps, from either the network management system or the CLI.

DMVPN Syslog Messages

The DMVPN syslog feature provides syslog messages for the following events:

- All next-hop state change events. For example, when the system declares that a Next Hop Server (NHS), Next Hop Client (NHC), or a Next Hop Peer (NHP) is up or down. The severity level for these messages is set to critical.
- NHRP resolution events. For example, when a spoke sends a resolution to a remote spoke, or when an NHRP resolution times out without receiving a response. The severity level for these messages is set to informational.
- DMVPN cryptography events. For example, when a DMVPN socket entry changes from open to closed, or from closed to open. The severity level for these messages is set to notification.
- NHRP error notifications. For example, when an NHRP registration or resolution event fails, when a system check event fails, or when an NHRP encapsulation error occurs, an NHRP error notification is displayed. The severity level for these messages is set to errors.

A sample NHRP error message is given below:

```
Received Error Indication from 209.165.200.226, code: administratively prohibited(4), (trigger src:
209.165.200.228 (nbma: 209.165.200.230) dst: 209.165.202.140), offset: 0, data: 00 01 08 00 00 00 00
00 00 FE 00 68 F4 03 00 34
```

The error message includes the IP address of the node where the error originates, the source nonbroadcast multiaccess (NBMA), and the destination address.

- DMVPN error notifications. For example, when the NET_ID value is not configured, or when an NHRP multicast replication failure occurs. The severity level is set to notification for the unconfigured NET_ID value message, and set to errors if an NHRP multicast replication failure occurs.
- The rate limit set for NHRP packets on the interface is exceeded. This event occurs when the NHRP packets handled by the NHRP process exceeds the rate limit set on the interface. The severity level for this message is set to warning.

Interface State Control

The Interface State Control feature allows NHRP to control the state of the interface based on whether the tunnels on the interface are live. If NHRP detects that all NHSs configured on the interface are in the down state, NHRP can change the interface state to down. However, if NHRP detects that any one of the NHSs configured on the interface is up, then it can change the state of the interface to up.

When the NHRP changes the interface state, other Cisco services can react to the state change, for example:

- If the interface state changes, the generic routing and encapsulation (GRE) interface generates IF-MIB notifications (traps) that report a LinkUp or LinkDown message. The system uses these traps to monitor the connectivity to the DMVPN cloud.
- If the interface state changes to down, the Cisco IOS backup interface feature can be initiated to allow the system to use another interface to provide an alternative path to the failed primary path.
- If the interface state changes to down, the system generates an update that is sent to all dynamic routing protocols. The Interface State Control feature a failover mechanism for dynamic routing when the multipoint GRE (mGRE) interface is down.

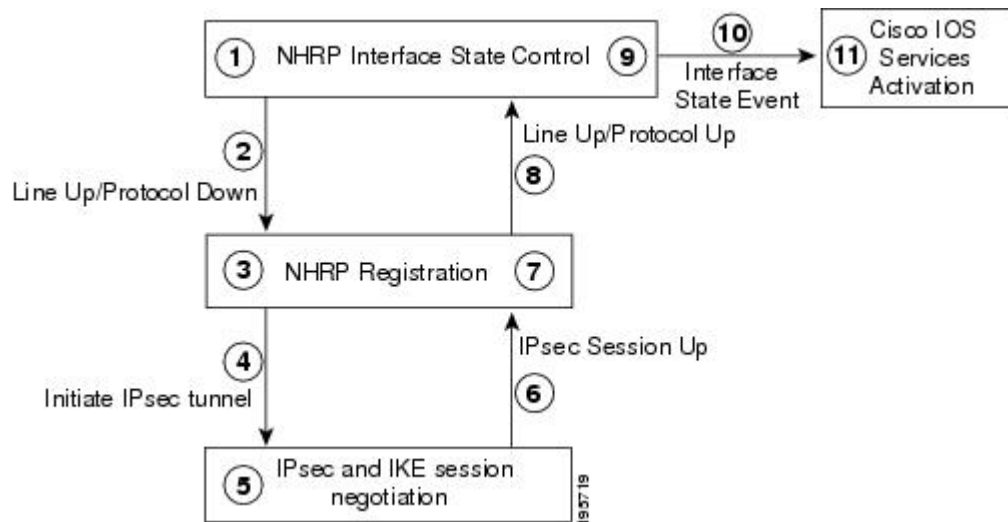
- If the interface state changes to down, the system clears any static routes that use the mGRE interface as the next hop. The Interface State Control feature provides a failover mechanism for routing when the mGRE interface is down.

The interface state control feature works on both point-to-point and mGRE interfaces.

Interface State Control Configuration Workflow

The diagram below illustrates how the system behaves when the Interface State Control feature is initialized.

Figure 5: Interface State Control Configuration Initialization Workflow



The Interface State Control initialization works as follows:

1. The Interface State Control feature is enabled on the GRE interface with NHRP configured.
2. The system reevaluates the protocol state and changes the state to line up and protocol down if none of the configured NHSs is responding.
3. The line up state change initiates the NHRP registration process.
4. The NHRP registration process initiates the IPsec tunnel.
5. The IPsec tunnel initiation starts the IPsec and IKE tunnel negotiation process.
6. On successful completion of the tunnel negotiation process, the system sends an IPsec Session Up message.
7. The NHRP registration process receives the IPsec Session Up message.
8. The NHRP registration process reports the line up and protocol up state to the GRE interface.
9. The GRE interface state changes to line up and protocol up.
10. The system reports the GRE interface state change to Cisco software.
11. The state change triggers Cisco services, such as interface event notifications, syslog events, DHCP renew, IP route refresh, and SNMP traps.

How to Configure DMVPN Tunnel Health Monitoring and Recovery

The DMVPN Tunnel Health Monitoring and Recovery feature allows you to configure SNMP NHRP notifications and interface states.

Configuring Interfaces to Generate SNMP NHRP Notifications

You can configure an interface so that SNMP NHRP traps are generated for NHRP events. In addition, you can configure the system to send the traps to particular trap receivers. To configure SNMP NHRP notifications on an interface, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community *string* rw**
4. **snmp-server enable traps nhrp nhs**
5. **snmp-server enable traps nhrp nhc**
6. **snmp-server enable traps nhrp nhp**
7. **snmp-server enable traps nhrp quota-exceeded**
8. **snmp-server host *ip-address* version *snmpversion* community-string**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp-server community <i>string</i> rw Example: Device(config)# snmp-server community public rw	Configures the community access string to permit access to the SNMP.
Step 4	snmp-server enable traps nhrp nhs Example:	Enables NHRP NHS notifications.

	Command or Action	Purpose
	Device(config)# snmp-server enable traps nhrp nhc	
Step 5	snmp-server enable traps nhrp nhc Example: Device(config)# snmp-server enable traps nhrp nhc	Enables NHRP NHC notifications.
Step 6	snmp-server enable traps nhrp nhp Example: Device(config)# snmp-server enable traps nhrp nhc	Enables NHRP NHP notifications.
Step 7	snmp-server enable traps nhrp quota-exceeded Example: Device(config)# snmp-server enable traps nhrp quota-exceeded	Enables notifications for when the rate limit set on the NHRP packets is exceeded on the interface.
Step 8	snmp-server host ip-address version snmpversion community-string Example: Device(config)# snmp-server host 192.40.3.130 version 2c public	Specifies the recipient of an SNMP notification operation. <ul style="list-style-type: none"> • By default, SNMP notifications are sent as traps. • All NHRP traps are sent to the notification receiver with the IP address 192.40.3.130 using the community string public.
Step 9	end Example: Device(config)# end	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the `debug snmp mib nhrp` command to troubleshoot SNMP NHRP notifications.

Configuring Interface State Control on an Interface

The Interface State Control feature enables the system to control the state of an interface based on whether the DMVPN tunnels connected to the interface are live or not. To configure interface state control on an interface, perform the steps in this section.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `interface type number`
4. `if-state nhrp`
5. `end`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type</i> <i>number</i> Example: Device(config)# interface tunnel 1	Configures an interface type and enters interface configuration mode.
Step 4	if-state nhrp Example: Device(config-if)# if-state nhrp	Enables NHRP to control the state of the tunnel interface.
Step 5	end Example: Device(config-if)# end	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for DMVPN Tunnel Health Monitoring and Recovery

Example: Configuring SNMP NHRP Notifications

The following example shows how to configure SNMP NHRP notifications on a hub or spoke:

```
Device(config)# snmp-server community public rw
Device(config)# snmp-server enable traps nhrp nhs
Device(config)# snmp-server enable traps nhrp nhc
Device(config)# snmp-server enable traps nhrp nhp
Device(config)# snmp-server enable traps nhrp quota-exceeded
Device(config)# snmp-server host 209.165.200.226 version 2c public
```

Example: Configuring Interface State Control

The following example shows how to configure the Interface State Control feature for a spoke:

```

interface Tunnel 1
 ip address 209.165.200.228 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map 209.165.201.2 209.165.201.10
 ip nhrp map 209.165.201.3 209.165.201.11
 ip nhrp map multicast 209.165.201.10
 ip nhrp map multicast 209.165.201.11
 ip nhrp network-id 1
 ip nhrp holdtime 90
 ip nhrp nhs 209.165.201.3
 ip nhrp nhs 209.165.201.2
 ip nhrp shortcut
 if-state nhrp
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 !
end

```

Additional References for DMVPN Tunnel Health Monitoring and Recovery

Related Documents

Related Topic	Document Title
Dynamic Multipoint VPN information	“Dynamic Multipoint VPN (DMVPN)” module in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
IKE configuration tasks such as defining an IKE policy	“Configuring Internet Key Exchange for IPsec VPNs” module in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
IPsec configuration tasks	“Configuring Security for VPNs with IPsec” module in the <i>Cisco IOS Security Configuration Guide: Secure Connectivity</i>
System messages	<i>System Messages Guide</i>

Standards and RFCs

Standard/RFC	Title
RFC 2332	<i>NBMA Next Hop Resolution Protocol (NHRP)</i>
RFC 2677	<i>Definitions of Managed Objects for the NBMA Next Hop Resolution Protocol (NHRP)</i>

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> • CISCO-NHRP-EXT-MIB • NHRP-MIB 	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

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 DMVPN Tunnel Health Monitoring and Recovery

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 12: Feature Information for Tunnel Health Monitoring and Recovery

Feature Name	Releases	Feature Information
DMVPN—Tunnel Health Monitoring and Recovery (Interface Line Control)		The DMVPN—Tunnel Health Monitoring and Recovery (Interface Line Control) feature enables NHRP to control the state of the tunnel interface based on the health of the DMVPN tunnels. The following command was introduced: if-state nhrp .



CHAPTER 7

DMVPN Event Tracing

The DMVPN Event Tracing feature provides a trace facility for troubleshooting Cisco IOS Dynamic Multipoint VPN (DMVPN). This feature enables you to monitor DMVPN events, errors, and exceptions. During runtime, the event trace mechanism logs trace information in a buffer space. A display mechanism extracts and decodes the debug data.

You can use the DMVPN Event Tracing feature to analyze the cause of a device failure. When you configure the DMVPN Event Tracing feature, the router logs messages from specific DMVPN subsystem components into the device memory. You can view trace messages stored in the memory or save them to a file.

- [Finding Feature Information, on page 107](#)
- [Information About DMVPN Event Tracing, on page 107](#)
- [How to Configure DMVPN Event Tracing, on page 108](#)
- [Configuration Examples for DMVPN Event Tracing, on page 110](#)
- [Additional References, on page 110](#)
- [Feature Information for DMVPN Event Tracing, on page 111](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Information About DMVPN Event Tracing

Benefits of DMVPN Event Tracing

- Displays debug information on the console during runtime.
- Avoids multiple debug calls, and hence improves device performance.
- Saves memory space.

DMVPN Event Tracing Options

The DMVPN Event Tracing feature defines the event data type, provides functionalities to capture the event, and prints the events and the CLI extensions required to access and modify the log. The table below lists different options that can be monitored using the DMVPN Event Tracing feature.

Table 13: DMVPN Event Trace Options

Event Type	Description
NHRP Event Trace	General Next Hop Resolution Protocol (NHRP) events, such as NHRP protocol, NHRP messages, changes in NHRP data structure, NHRP NBMA or protocol address change, and NHRP traps.
NHRP Error Trace	All NHRP error events.
NHRP Exception Trace	All NHRP exception events.
Tunnel Event Trace	All tunnel events.

How to Configure DMVPN Event Tracing

You can configure the DMVPN Event Tracing feature in privileged EXEC mode or global configuration mode based on the desired parameters. See the *Cisco IOS Security Command Reference* for information on different parameters available in privileged EXEC mode or global configuration mode.

Perform one of the following tasks to configure the DMVPN Event Tracing feature:

Configuring DMVPN Event Tracing in Privileged EXEC Mode

Perform this task to configure DMVPN event tracing in privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **monitor event-trace dmvpn {nhrp {error | event | exception} | tunnel} {clear | continuous [cancel] | disable | enable | one-shot} | tunnel}**

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	monitor event-trace dmvpn {nhrp {error event exception} tunnel} {clear continuous [cancel] disable enable one-shot} tunnel}	Monitors and controls DMVPM traces.

	Command or Action	Purpose
	Example: Router# monitor event-trace dmvpn nhrp error enable	

Configuring DMVPN Event Tracing in Global Configuration Mode

Perform this task to configure DMVPN event tracing in global configuration mode.

SUMMARY STEPS

1. enable
2. configure terminal
3. monitor event-trace dmvpn {dump-file url | {nhrp {error | event | exception} | tunnel} {disable | dump-file url | enable | size | stacktrace value}}
4. 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.
Step 3	monitor event-trace dmvpn {dump-file url {nhrp {error event exception} tunnel} {disable dump-file url enable size stacktrace value}} Example: Router(config)# monitor event-trace dmvpn nhrp error enable	Monitors and controls DMVPM traces.
Step 4	exit Example: Router(config)# exit	Exits global configuration mode.

Configuration Examples for DMVPN Event Tracing

Example Configuring DMVPN Event Tracing in Privileged EXEC Mode

The following example shows how to monitor NHRP error traces in privileged EXEC mode:

```
Router> enable
Router# monitor event-trace dmvpn nhrp error enable
```

Example Configuring DMVPN Event Tracing in Global Configuration Mode

The following example shows how to monitor NHRP error traces in global configuration mode:

```
Router> enable
Router# configure terminal
Router(config)# monitor event-trace dmvpn nhrp error enable
```

Additional References

Related Documents

Related Topic	Document Title
DMVPN commands	<i>Cisco IOS Security Command Reference</i>

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	--

RFCs

RFC	Title
None	--

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 DMVPN Event Tracing

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 14: Feature Information for DMVPN Event Tracing

Feature Name	Releases	Feature Information
DMVPN Event Tracing		<p>The DMVPN Event Tracing feature provides a trace facility for troubleshooting Cisco IOS DMVPN. This feature enables you to monitor DMVPN events, errors, and exceptions. During runtime, the event trace mechanism logs trace information in a buffer space. A display mechanism extracts and decodes the debug data.</p> <p>The following commands were introduced or modified: monitor event-trace dmvpn, show monitor event-trace dmvpn.</p>



CHAPTER 8

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 VPN Routing and Forwarding (VRF) aware and supports VRF-aware queries.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Prerequisites for NHRP MIB, on page 113](#)
- [Restrictions for NHRP MIB, on page 113](#)
- [Information About NHRP MIB, on page 114](#)
- [How to Use NHRP MIB, on page 114](#)
- [Configuration Examples for NHRP MIB, on page 115](#)
- [Additional References, on page 117](#)
- [Feature Information for NHRP MIB, on page 118](#)

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

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 to implement the NHRP MIB 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.

This section contains the following task:

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

Configuration Examples for NHRP MIB

Example Verifying NHRP MIB Status

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

```
Router# 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 Vrf1, for monitoring by SNMP:

```
ip vrf Vrf1
 rd 198102
 ! Name of the SNMP VPN context
 context Vrf1-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 Vrf1 VPN
ip vrf forwarding Vrf1
ip address 10.0.0.1 255.255.255.0
ip mtu 1400
ip nhrp authentication sample
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 Vrf1
    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 Vrf1
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 Vrf1_context
!
!
snmp mib community-map abc context Vrf1-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 sample
  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
Description of SNMP, SNMP MIBs, and how to configure SNMP on Cisco devices	“Configuring SNMP Support” chapter in the <i>Cisco IOS Network Management Configuration Guide</i>
<i>Security commands</i>	<i>Cisco IOS Security Command Reference</i>
Recommended cryptographic algorithms	Next Generation Encryption

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 15: Feature Information for NHRP MIB

Feature Name	Releases	Feature Information
NHRP MIB for DMVPN Networks	Cisco IOS XE Release 2.5	<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>



CHAPTER 9

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.

- [Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, on page 119](#)
- [Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, on page 120](#)
- [Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device, on page 120](#)
- [Additional References, on page 124](#)

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 16: 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	Cisco IOS XE Release 17.4	This feature was introduced.

Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

In order for two 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 Port 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 the DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device feature 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

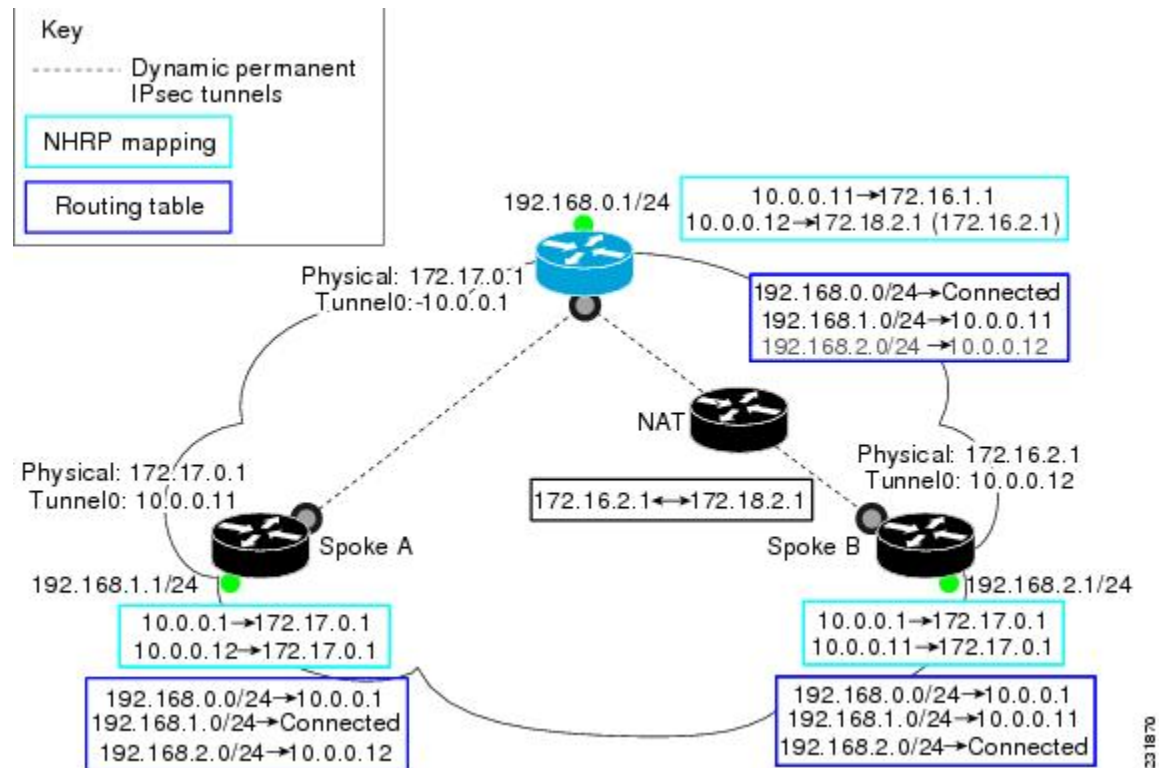
NAT allows a single device, such as a router, to act as agent between the Internet (or “public network”) and a local (or “private”) network, and is often used because of the scarcity of available IP addresses. A single

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 for an extended period.

The figure 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 6: Implementation of DMVPN Spoke-to-Spoke Tunneling Limited to Spokes Not Behind a NAT Device



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

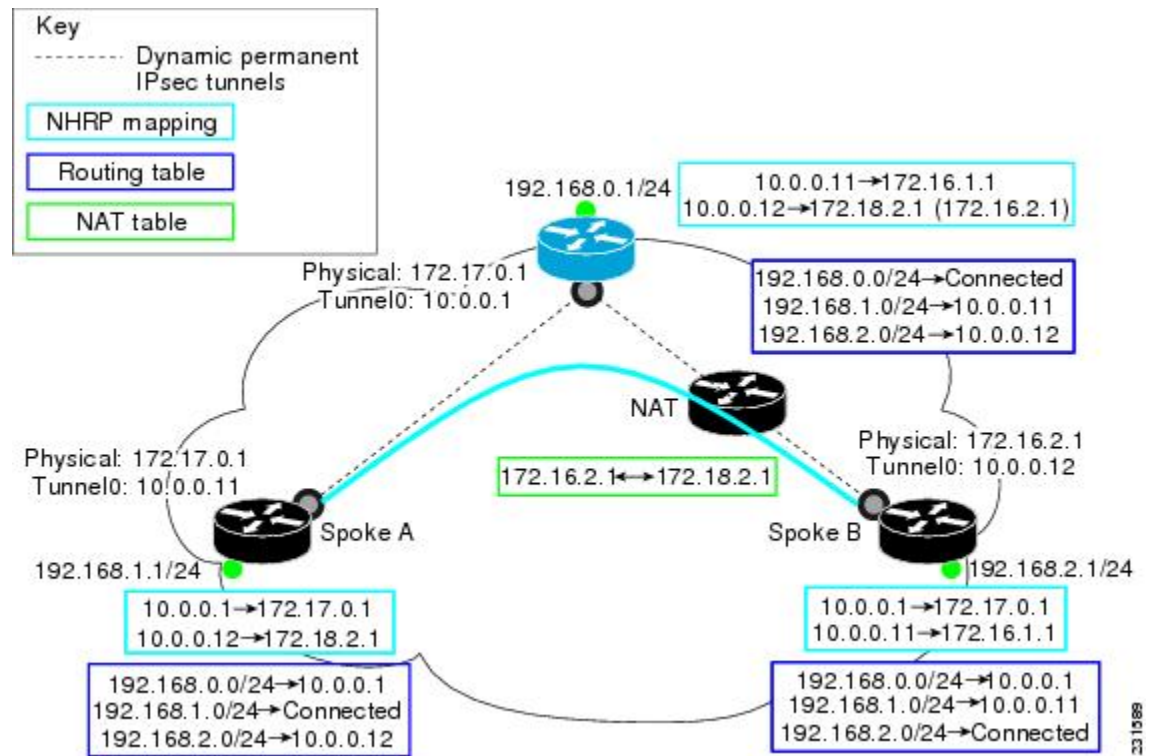
The NHRP Spoke-to-Spoke Tunnel with NAT feature introduces NAT extension in the NHRP protocol and is enabled automatically. The NHRP NAT extension is a Client Information Entry (CIE) entry with information about the protocol and post-NAT NBMA address. This additional information allows the support of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



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.

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

Figure 7: NHRP Between Spoke-to-Spoke Tunnels



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 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)” module in the <i>Cisco IOS XE Security Configuration Guide: Secure Connectivity</i>

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
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://tools.cisco.com/ITDIT/MIBS/servlet/index

RFCs

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

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



CHAPTER 10

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. These tunnel interfaces share a single underlying cryptographic SADB, cryptographic map, and IPsec profile in the Dynamic Multipoint Virtual Private Network (DMVPN) configuration.

If IPsec security association (SA) sessions are not shared in the same IPsec SADB, then an IPsec SA may get associated with an undesired IPsec SADB, and may also get associated with a wrong tunnel interface, causing duplication of IPsec SAs and flapping of tunnel interfaces. If the tunnel interfaces flap (change rapidly and repeatedly between online and offline states), then network connectivity problems occur.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Prerequisites for Sharing IPsec with Tunnel Protection](#), on page 127
- [Restrictions for Sharing IPsec with Tunnel Protection](#), on page 127
- [Information About Sharing IPsec with Tunnel Protection](#), on page 128
- [How to Configure Sharing IPsec with Tunnel Protection](#), on page 129
- [Configuration Examples for Sharing IPsec with Tunnel Protection](#), on page 131
- [Additional References](#), on page 141
- [Feature Information for Sharing IPsec with Tunnel Protection](#), on page 142
- [Glossary](#), on page 142

Prerequisites for Sharing IPsec with Tunnel Protection

- 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.

Restrictions for Sharing IPsec with Tunnel Protection

- The **tunnel source** command on all the tunnel interfaces that use the same tunnel source must be configured using interface type and number, not the tunnel's IP address.

- All tunnels with the same tunnel source interface must use the same IPsec profile and must have the **tunnel protection shared** command configured. 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, 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 through 10 use **loopback1**, then define the profile **ipsec-profile-1** for tunnels 1 through 5 and **ipsec-profile-2** for tunnels 6 through 10.

- 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 use the ISAKMP profile to connect 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.
- Sharing IPsec is not supported on Virtual-Template type tunnel interfaces. It cannot be used either in the default **tunnel mode gre ip** mode with IPsec protection, (for example, FlexVPN) or with the **tunnel mode ipsec ipv4** (for example, Dynamic Virtual Tunnel interface - DVTI). Each virtual-template interface must have a separate and unshared IPsec profile. Otherwise, the router might crash after the virtual-access is deleted.

Information About Sharing IPsec with Tunnel Protection

Single IPsec SAs and 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 two endpoints of the same type. In this case, you should use a single IPsec SA to secure both GRE tunnel sessions. It is not possible to determine the tunnel interface under which 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 configuration allows a single IPsec SA to be used for all GRE tunnels (same tunnel source and destination, but different tunnel keys) between two endpoints of the same type. The **tunnel protection ipsec profile shared** command also makes IPsec

QM processing unambiguous because there is one SADB 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 cryptographic map parameters. On the cryptodata plane, the decrypted and GRE decapsulated packets are demultiplexed to the appropriate tunnel interface by the GRE module using a local address, a remote address, and optional tunnel key information.

When the IPsec path maximum transmission unit (MTU) changes, the value of SA MTU in the Quantum Flow Processor (QFP) and the hardware cryptographic engine gets updated and becomes consistent with the IPsec MTU. While the MTU changes, the system may drop some packets and transient %ATTN-3-SYNC_TIMEOUT errors may be displayed on the console.



Note The tunnel source, tunnel destination, and tunnel key (triplet) must be unique for all tunnel interfaces on a router. For a multipoint GRE (mGRE) interface where the 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 no match, then they are matched to mGRE tunnels.

How to Configure Sharing IPsec with Tunnel Protection

Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

Perform this task to configure a Cisco IOS router to share an IPsec SADB between multiple tunnel interfaces in a DMVPN.

If your configuration requires more spoke routers in a dual-hub, dual DMVPN topology, repeat the steps listed in this task to configure additional spokes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel source** {*ip-address* | *interface-type interface-number*}
5. **tunnel protection ipsec 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.

	Command or Action	Purpose
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.
Step 4	tunnel source {<i>ip-address</i> <i>interface-type interface-number</i>} Example: <pre>Router(config-if)# tunnel source GigabitEthernet 0</pre>	Sets the source IP address or source interface type number for a tunnel interface. <ul style="list-style-type: none"> When you are using the tunnel protection ipsec profile command, you must specify an interface, not an IP address for the tunnel source.
Step 5	tunnel protection ipsec profile <i>name</i> [shared] Example: <pre>Router(config-if)# tunnel protection ipsec profile vpnprof shared</pre>	Associates a tunnel interface with an IPsec profile. <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. The shared keyword allows IPsec sessions to be shared between multiple tunnel interfaces configured with the same tunnel source IP.
Step 6	exit Example: <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and enters global configuration mode.
Step 7	exit Example: <pre>Router(config)# exit</pre>	Exits global configuration mode and enters privileged EXEC mode.

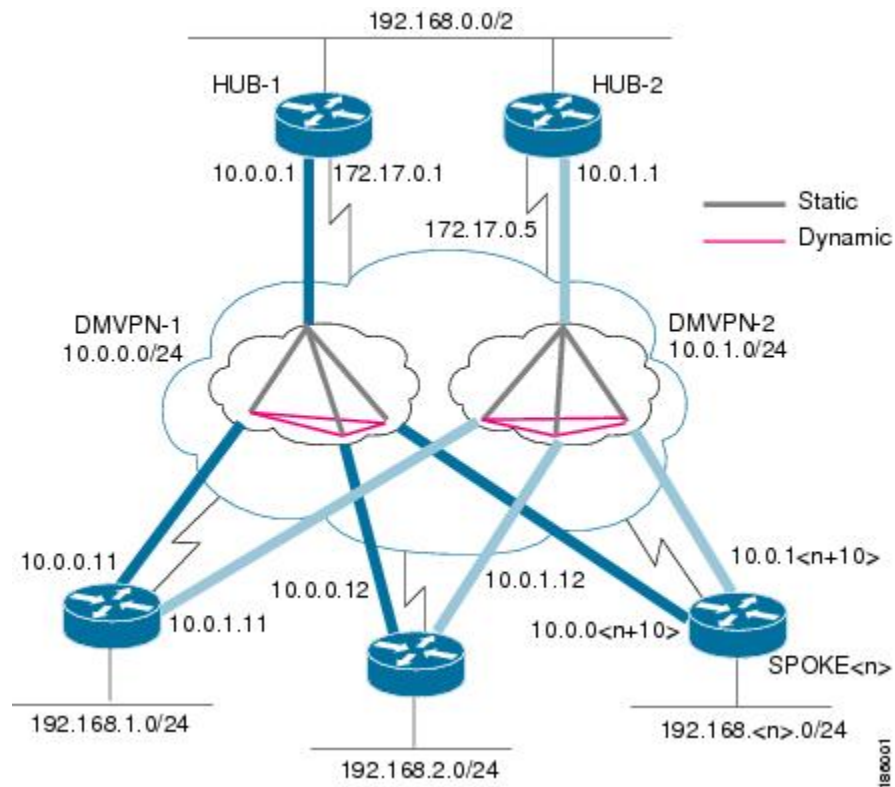
Configuration Examples for Sharing IPsec with Tunnel Protection

Example: Dual-Hub Router, Dual-DMVPN Topology

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

- Each hub router is configured with a single mGRE tunnel interface.
- Each hub router is connected to one DMVPN subnet (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 the same tunnel source IP address and uses shared tunnel protection between them.

Figure 8: Dual-Hub Router, Dual-DMVPN Topology



Example: Configuring an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

Example: HUB-1 Configuration

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 Tunnel 5
 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 GigabitEthernet 0/0/0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection IPsec profile vpnprof
!
interface GigabitEthernet 0/0/0
 ip address 172.17.0.1 255.255.255.252
!
interface GigabitEthernet 0/0/1
 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
!

```

Example: HUB-2 Configuration

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 Tunnel 5
 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 GigabitEthernet 0/0/0
 tunnel mode gre multipoint
 tunnel key 100001
 tunnel protection ipsec profile vpnprof
!
interface GigabitEthernet 0/0/0
 ip address 172.17.0.5 255.255.255.252
!
interface GigabitEthernet 0/0/1
 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
!

```

Example: SPOKE 1 Configuration

SPOKE 1 has the following DMVPN configuration:

```

!
hostname Spoke1
!

```

Example: SPOKE 2 Configuration

```

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 Tunnel 5
  bandwidth 1000
.
.
.
  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 protection ipsec profile vpnprof shared
!
interface Tunnel 5
  bandwidth 1000
.
.
.
  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 protection ipsec profile vpnprof shared
!
interface GigabitEthernet 0/0/0
  ip address dhcp hostname Spoke1
!
interface GigabitEthernet 0/0/1
  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
!

```

Example: SPOKE 2 Configuration

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 Tunnel 5  
  bandwidth 1000  
.  
.  
.  
  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 protection ipsec profile vpnprof shared  
!  
interface Tunnel 5  
  bandwidth 1000  
.  
.  
.  
  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 protection ipsec profile vpnprof shared  
!  
interface GigabitEthernet 0/0/0  
  ip address dhcp hostname Spoke2  
!  
interface GigabitEthernet 0/0/1  
  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  
!
```

Example: 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, Tunnel 0 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, Tunnel 0 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, Tunnel 1 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, Tunnel 1 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 three crypto connections (172.17.0.12, 172.17.0.5 and 172.17.0.1). The two NHRP sessions (10.0.0.12, Tunnel 0) and (10.0.1.12, Tunnel 1) represent the same IPsec session because 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: FastEthernet0/0/0 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 The three crypto sessions are shown under both tunnel interface (three entries, twice) in the **show crypto ipsec sa** output because both interfaces are mapped to the same IPsec SADB, which has three entries. This duplication of output is expected in this case.

```

Spoke1# show crypto ipsec sa

interface: Tunnel 0
  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 FastEthernet0/0/0
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 FastEthernet0/0/0
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 FastEthernet0/0/0
    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: Tunnel 1
      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 FastEthernet0/0/0
      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 FastEthernet0/0/0
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 FastEthernet0/0/0
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:

```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Dynamic Multipoint VPN	<i>Dynamic Multipoint VPN Configuration Guide</i>
IPv6 commands	<i>IPv6 Command Reference</i>
Cisco IOS IPv6 features	IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

Technical Assistance

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

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 17: Feature Information for Sharing IPsec with Tunnel Protection

Feature Name	Releases	Feature Information
Sharing IPSec with Tunnel Protection	Cisco IOS XE 17.4	This feature was introduced.

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 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, 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 Encapsulating Security Payload (ESP) protocol with the Hash-based Message Authentication Code (HMAC)-Message Digest Algorithm (MD5) authentication algorithm; another transform is the Authentication Header (AH) protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-Secure Hash Algorithm (SHA) authentication algorithm.

tunnel—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.



CHAPTER 11

Per-Tunnel QoS for DMVPN

The Per-Tunnel QoS for DMVPN feature introduces per-tunnel QoS support for DMVPN and increases per-tunnel QoS performance for IPsec tunnel interfaces.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Finding Feature Information](#), on page 145
- [Prerequisites for Per-Tunnel QoS for DMVPN](#), on page 145
- [Restrictions for Per-Tunnel QoS for DMVPN](#), on page 146
- [Information About Per-Tunnel QoS for DMVPN](#), on page 146
- [How to Configure Per-Tunnel QoS for DMVPN](#), on page 148
- [Configuration Examples for Per-Tunnel QoS for DMVPN](#), on page 152
- [Additional References for Per-Tunnel QoS for DMVPN](#), on page 159
- [Feature Information for Per-Tunnel QoS for DMVPN](#), on page 160

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Prerequisites for Per-Tunnel QoS for DMVPN

- Before you configure the Per-Tunnel QoS for DMVPN feature, you must configure Cisco Express Forwarding switching.

- Before you can configure an Next Hop Resolution Protocol (NHRP) group on a spoke and map the NHRP group to a QoS policy on a hub, the spoke and the hub must already be configured for DMVPN without the per-tunnel QoS.

Restrictions for Per-Tunnel QoS for DMVPN

Information About Per-Tunnel QoS for DMVPN

Per-Tunnel QoS for DMVPN Overview

The Per-Tunnel QoS for DMVPN feature lets you apply a quality of service (QoS) policy on a Dynamic Multipoint VPN (DMVPN) hub on a per-tunnel instance (per-spoke basis) in the egress direction for DMVPN hub-to-spoke tunnels. The QoS policy on a DMVPN hub on a per-tunnel instance lets you shape 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.

You can use this feature with DMVPN with or without Internet Protocol Security (IPsec).

When the Per-Tunnel QoS for DMVPN feature is enabled, queuing and shaping are performed at the outbound physical interface for generic routing encapsulation (GRE)/IPsec tunnel packets. The Per-Tunnel QoS for DMVPN feature ensures that the GRE header, the IPsec header, and the Layer 2 (for the physical interface) header are included in the packet-size calculations for shaping and bandwidth queuing of packets under QoS.

Benefits of Per-Tunnel QoS for DMVPN

Before the introduction of Per-Tunnel QoS for DMVPN feature, quality of service (QoS) on a Dynamic Multipoint VPN (DMVPN) hub could be configured to measure only either the outbound traffic in the aggregate (overall spokes) or outbound traffic on a per-spoke basis (with extensive manual configuration).

The Per-Tunnel QoS for DMVPN feature provides the following benefits:

- The QoS policy is attached to the DMVPN hub, and the 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; therefore, the traffic cannot monopolize a hub’s resources and starve other spokes.

NHRP QoS Provisioning for DMVPN

Next Hop Resolution Protocol (NHRP) performs the provisioning for the Per-Tunnel QoS for DMVPN feature by using NHRP groups.

An NHRP group, a new functionality introduced by this feature, is the group identity information signaled by a Dynamic Multipoint VPN (DMVPN) node (a spoke) to the DMVPN hub. The hub uses this information to select a locally defined quality of service (QoS) policy instance for the remote node.

You can configure an NHRP group on the spoke router on the DMVPN generic routing encapsulation (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.

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 to that hub-to-spoke tunnel in the egress direction.

After an NHRP group is configured on a spoke, the group is not immediately sent to the hub, but is sent in the next periodic registration request. The spoke can belong to only one NHRP group per GRE tunnel interface. If a spoke is configured as 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 to the spoke, and any existing QoS policy applied to 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 is received from a spoke similar to the earlier 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.

Per-Tunnel QoS for Spoke to Spoke Connections

The QoS: Spoke to Spoke per tunnel QoS for DMVPN feature enables a DMVPN client to establish a direct crypto tunnel with another DMVPN client leveraging the per-tunnel QoS policy, using Next Hop Resolution Protocol (NHRP) to build spoke-to-spoke connections.

This feature enhances the Adaptive QoS over DMVPN feature, which ensures effective bandwidth management using dynamic shapers based on available bandwidth.

A spoke-to-spoke connection is established when a group identity information, configured on the spokes using the **nhrp attribute group** command, is exchanged between the spokes through the NHRP Vendor Private Extension (VPE). The NHRP Vendor Private Extensions, encapsulated in NHRP control packets—NHRP resolution request and reply packets.

Assume a network with two spokes—Spoke A and Spoke B, connected to hub. If Spoke A is configured with the **nhrp attribute group** command and traffic exists between the Spoke A and Spoke B, a resolution request from the Spoke A carries the group identity information as part of Vendor Private Extension (VPE). On receiving the resolution request, Spoke B extracts the VPE header and checks the extension types received as part of the resolution request packet. If the VPE extension has group type, the NHRP VPE parser extracts the group information and checks if a matching map is present. If a matching map is present, QoS applies the policy on the target interface.

How to Configure Per-Tunnel QoS for DMVPN

To configure the Per-Tunnel QoS for DMVPN feature, you define a Next Hop Resolution Protocol (NHRP) group on the spokes and then map the NHRP group to a quality of service (QoS) policy on the hub.

Configuring an NHRP Group on a Spoke

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **nhrp group** *group-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 1	Configures a tunnel interface and enters interface configuration mode.
Step 4	nhrp group <i>group-name</i> Example: Device(config-if)# nhrp group spoke_group1	Configures a Next Hop Resolution Protocol (NHRP) group on the spoke.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring an NHRP Group Attribute on a Spoke

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface tunnel** *number*
4. **nhrp attribute group** *group-name*
5. **nhrp map group** *group-name* **service-policy output** *qos-policy-map-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 1	Configures a tunnel interface and enters interface configuration mode.
Step 4	nhrp attribute group <i>group-name</i> Example: Device(config-if)# nhrp attribute group spokel	Configures the QoS group identity information on the spoke.
Step 5	nhrp map group <i>group-name</i> service-policy output <i>qos-policy-map-name</i> Example: Device(config-if)# nhrp map group spoke_group1 service-policy output group1_parent	Adds the Next Hop Resolution Protocol (NHRP) group to the quality of service (QoS) policy mapping.
Step 6	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Mapping an NHRP Group to a QoS Policy on the Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **nhrp map group** *group-name* **service-policy output** *qos-policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 1	Configures a tunnel interface and enters interface configuration mode.
Step 4	nhrp map group <i>group-name</i> service-policy output <i>qos-policy-map-name</i> Example: Device(config-if)# nhrp map group spoke_group1 service-policy output group1_parent	Adds the Next Hop Resolution Protocol (NHRP) group to the quality of service (QoS) policy mapping on the hub.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling DMVPN Per-tunnel QoS Sourced from Port Channel

To enable the feature, you must configure the command `platform qos port-channel-aggregate <port-channel number>` before configuring port channel.

The `platform qos port-channel-aggregate <port-channel number>` is required for this feature. The order of the configuration steps are important to enable DMVPN Per-tunnel QoS Sourced from Port-Channel feature. The `platform qos port-channel-aggregate <port-channel number>` command must be configured first. Then, the port-channel interface must be created. Lastly, `channel-group x` command must be applied to member ports.

Both port-channel main-interface and sub-interface are supported in aggregate mode.



Note Before configuring the command, you must remove the 'port channel interface' and 'channel-group' configuration from physical interface.

1. Enable the command `platform qos port-channel-aggregate <port-channel number>` before configuring port channel.

2. Configure per-tunnel QoS.
3. Reset the NHRP registration process to ensure the spokes register now that the new configuration is present on the hub BR. Use the command `show dmvpn detail` to display the NHRP group for each spoke.

Verifying Per-Tunnel QoS for DMVPN

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show dmvpn detail Example: Device# show dmvpn detail	Displays detailed Dynamic Multipoint VPN (DMVPN) information for each session, including the Next Hop Server (NHS) and NHS status, crypto session information, and socket details. <ul style="list-style-type: none"> • The output includes the Next Hop Resolution Protocol (NHRP) group received from the spoke and the quality of service (QoS) policy applied to the spoke tunnel.
Step 3	show nhrp Example: Device# show nhrp	Displays the NHRP cache and the NHRP group received from the spoke.
Step 4	show nhrp group [group-name] Example: Device# show nhrp group	Displays NHRP group mapping. <ul style="list-style-type: none"> • The output includes the associated QoS policy name and the list of tunnel endpoints using the QoS policy.
Step 5	show nhrp group-map [group-name] Example: Device# show nhrp group-map group1-parent	Displays the group-to-policy maps configured on the hub and also displays the tunnels on which the QoS policy is applied.

	Command or Action	Purpose
Step 6	show policy-map multipoint [tunnel tunnel-interface-number] Example: Device# show policy-map multipoint tunnel 1	Displays QoS policy details applied to multipoint tunnels.
Step 7	show tunnel endpoints Example: Device# show tunnel endpoints	Displays information about the source and destination endpoints for multipoint tunnels and the QoS policy applied on the spoke tunnel.

Configuration Examples for Per-Tunnel QoS for DMVPN

Example: Configuring an NHRP Group on a Spoke

The following example shows how to configure two Next Hop Resolution Protocol (NHRP) groups on three spokes:

Configuring the First Spoke

```
interface tunnel 1
 ip address 209.165.200.225 255.255.255.224
 no ip redirects
 ip mtu 1400
 ip nhrp authentication testing
 nhrp group spoke_group1
 ip nhrp map 209.165.200.226 203.0.113.1
 ip nhrp map multicast 203.0.113.1
 ip nhrp network-id 172176366
 ip nhrp holdtime 300
 ip tcp adjust-mss 1360
 ip nhrp nhs 209.165.200.226
 tunnel source fastethernet 2/1/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile DMVPN
interface fastethernet 2/1/1
 ip address 203.0.113.2 255.255.255.0
```

Configuring the Second Spoke

```
interface tunnel 1
 ip address 209.165.200.227 255.255.255.224
 no ip redirects
 ip mtu 1400
 ip nhrp authentication testing
 nhrp group spoke_group1
 ip nhrp map 209.165.200.226 203.0.113.1
 ip nhrp map multicast 203.0.113.1
 ip nhrp network-id 172176366
 ip nhrp holdtime 300
```

```

ip tcp adjust-mss 1360
ip nhrp nhs 209.165.200.226
tunnel source fastethernet 2/1/1
tunnel mode gre multipoint
tunnel protection ipsec profile DMVPN
interface fastethernet 2/1/1
ip address 203.0.113.3 255.255.255.0

```

Configuring the Third Spoke

```

interface tunnel 1
ip address 209.165.200.228 255.255.255.224
no ip redirects
ip mtu 1400
ip nhrp authentication testing
nhrp group spoke_group2
ip nhrp map 209.165.200.226 203.0.113.1
ip nhrp map multicast 203.0.113.1
ip nhrp network-id 172176366
ip nhrp holdtime 300
ip tcp adjust-mss 1360
ip nhrp nhs 209.165.200.226
tunnel source fastethernet 2/1/1
tunnel mode gre multipoint
tunnel protection ipsec profile DMVPN
interface fastethernet 2/1/1
ip address 203.0.113.4 255.255.255.0

```

Example: Configuring an NHRP Group Attribute on a Spoke

The following example shows how to configure two Next Hop Resolution Protocol (NHRP) groups attributes on two spokes:

Configuring the First Spoke

```

class-map match-any class2
match ip precedence 5
end
!
policy-map p2
class class2
priority percent 60
end
!
interface Tunnel0
ip address 10.0.0.2 255.255.255.0
no ip redirects
ip mtu 1436
ip nhrp authentication h1there
ip nhrp attribute group1
ip nhrp map group group1 service-policy output p2
ip nhrp map multicast 172.17.0.1
ip nhrp map 10.0.0.1 172.17.0.1
ip nhrp network-id 253
ip nhrp nhs 10.0.0.1
ip nhrp registration timeout 600
ip nhrp cache non-authoritative
no ip mroute-cache
tunnel source 172.17.0.2
tunnel mode gre multipoint

```

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

```
tunnel key 253
tunnel protection ipsec profile dmvpn-profile
end
```

Configuring the Second Spoke

```
class-map match-any class1
match ip precedence 5

policy-map policy p1
class class1
priority 70

interface Tunnel0
ip address 10.0.0.1 255.255.255.0
no ip redirects
ip mtu 1436
ip nhrp authentication hlthere
ip nhrp attribute group1
ip nhrp map group group1 service-policy output p1
ip nhrp map multicast 172.17.0.2
ip nhrp map 10.0.0.2 172.17.0.2
ip nhrp network-id 253
ip nhrp nhs 10.0.0.2
ip nhrp registration timeout 600
ip nhrp cache non-authoritative
no ip mroute-cache
tunnel source 172.17.0.1
tunnel mode gre multipoint
tunnel key 253
tunnel protection ipsec profile dmvpn-profile
end
```

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

The following example shows how to map Next Hop Resolution Protocol (NHRP) groups to a quality of service (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 configuring Per-tunnel QoS for Dynamic Multipoint VPN (DMVPN) feature. 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 209.165.200.225 255.255.255.224
  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 unique
  tunnel source fastethernet 2/1/1
  tunnel mode gre multipoint
  tunnel protection ipsec profile DMVPN
interface fastethernet 2/1/1
  ip address 209.165.200.226 255.255.255.224

```

Example: Enabling DMVPN Per-tunnel QoS Sourced from Port Channel

The following example shows how to enable DMVPN Per-tunnel QoS Sourced from Port Channel.

Example: Configuring on hub

```

platform qos port-channel-aggregate 1
!
class-map match-any class2
match ip precedence 5
!
policy-map p1
class class2
  priority percent 60
!
interface Port-channel1
ip address 203.0.113.1 255.255.255.0
!
interface GigabitEthernet0/0/0
channel-group 1
!
interface GigabitEthernet0/0/1
channel-group 1
!
interface Tunnell1
ip address 10.9.9.1 255.255.255.0
no ip redirects
ip nhrp authentication cisco
nhrp map group group1 service-policy output p1
ip nhrp map multicast dynamic
ip nhrp network-id 1
tunnel source Port-channel 1
tunnel mode gre multipoint

```

Example: Configuring on spoke

```

platform qos port-channel-aggregate 1
!
interface Port-channel1
ip address 203.0.113.100 255.255.255.0
!
interface GigabitEthernet0/0/0
channel-group 1
!
interface GigabitEthernet0/0/1
channel-group 1
!
interface Tunnell
ip address 10.9.9.11 255.255.255.0
no ip redirects
ip nhrp authentication cisco
ip nhrp map 10.9.9.1 203.0.113.1
ip nhrp map multicast 203.0.113.1
ip nhrp network-id 1
ip nhrp nhs 10.9.9.1
tunnel source Port-channel 1
nhrp group group1
tunnel mode gre multipoint

```

Example: Verifying Per-Tunnel QoS for DMVPN

The following example shows how to display the information about Next Hop Resolution Protocol (NHRP) groups received from the spokes and display the quality of service (QoS) policy that is applied to each spoke tunnel. You can enter this command on the hub.

```

Device# 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 209.165.200.225, VRF ""
  Tunnel Src./Dest. addr: 209.165.200.226/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
-----
   1 209.165.200.227 192.0.2.2 UP 00:19:20 D 192.0.2.2/32
NHRP group: spoke_group1
  Output QoS service-policy applied: group1_parent
   1 209.165.200.228 192.0.2.3 UP 00:19:20 D 192.0.2.3/32
NHRP group: spoke_group1
  Output QoS service-policy applied: group1_parent
   1 209.165.200.229 192.0.2.4 UP 00:19:23 D 192.0.2.4/32
NHRP group: spoke_group2
  Output QoS service-policy applied: group2_parent
Crypto Session Details:
-----
Interface: tunnell
Session: [0x04AC1D00]
  IKE SA: local 209.165.200.226/500 remote 209.165.200.227/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 209.165.200.227
  IPSEC FLOW: permit 47 host 209.165.200.226 host 209.165.200.227
  Active SAs: 2, origin: crypto map

```

```

    Outbound SPI : 0x9B264329, transform : ah-sha-hmac
    Socket State: Open
Interface: tunnell
Session: [0x04AC1C08]
  IKE SA: local 209.165.200.226/500 remote 209.165.200.228/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 209.165.200.228
  IPSEC FLOW: permit 47 host 209.165.200.226 host 209.165.200.228
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0x36FD56E2, transform : ah-sha-hmac
    Socket State: Open
Interface: tunnell
Session: [0x04AC1B10]
  IKE SA: local 209.165.200.226/500 remote 209.165.200.229/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_id: 209.165.200.229
  IPSEC FLOW: permit 47 host 209.165.200.226 host 209.165.200.229
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0xAC96818F, transform : ah-sha-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 can enter this command on the hub.

```

Device# show ip nhrp
192.0.2.240/32 via 192.0.2.240
  Tunnell created 00:22:49, expire 00:01:40
  Type: dynamic, Flags: registered
  NBMA address: 209.165.200.227
  Group: spoke_group1
192.0.2.241/32 via 192.0.2.241
  Tunnell created 00:22:48, expire 00:01:41
  Type: dynamic, Flags: registered
  NBMA address: 209.165.200.228
  Group: spoke_group1
192.0.2.242/32 via 192.0.2.242
  Tunnell created 00:22:52, expire 00:03:27
  Type: dynamic, Flags: registered
  NBMA address: 209.165.200.229
  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 can enter this command on the hub.

```

Device# show nhrp group-map
Interface: tunnell
  NHRP group: spoke_group1
  QoS policy: group1_parent
  Tunnels using the QoS policy:
  Tunnel destination overlay/transport address
  198.51.100.220/203.0.113.240
  198.51.100.221/203.0.113.241
  NHRP group: spoke_group2
  QoS policy: group2_parent
  Tunnels using the QoS policy:
  Tunnel destination overlay/transport address
  198.51.100.222/203.0.113.242

```

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

Example: Verifying Per-Tunnel QoS for DMVPN

```

Device# show policy-map multipoint

Interface tunnel1 <--> 203.0.113.252
  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 tunnel1 <--> 203.0.113.253
  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 tunnel1 <--> 203.0.113.254
  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 for Per-Tunnel QoS for DMVPN

Related Documents

Related Topic	Document Title

Related Topic	Document Title
Security commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IP NHRP commands	Cisco IOS IP Addressing Services Command Reference
Configuring Basic Cisco Express Forwarding	IP Switching Cisco Express Forwarding Configuration Guide
Configuring NHRP	IP Addressing: NHRP Configuration Guide
Recommended cryptographic algorithms	Next Generation Encryption

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 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 18: Feature Information for Per-Tunnel QoS for DMVPN

Feature Name	Releases	Feature Information
Per-Tunnel QoS	15.4(1)T / 3.11S	<p>The Per-Tunnel QoS for DMVPN feature introduces per-tunnel QoS support for DMVPN and increases per-tunnel QoS performance for IPsec tunnel interfaces.</p> <p>In , this feature was enhanced to provide support for IPv6 addresses.</p> <p>The following commands were introduced or modified: ip nhrp map, nhrp group, nhrp map group, show dmvpn, show ip nhrp, show ip nhrp group-map, show nhrp group-map, show policy-map multipoint tunnel.</p> <p>The commands ip nhrp group and ip nhrp map group were depreciated and hidden in the CLI. They are replaced with protocol agnostic nhrp group and nhrp map group. The configuration needs to be manually migrated to the new syntax.</p>
	16.6.5, 16.8.1	The commands ip nhrp group and ip nhrp map group are removed from CLI. Manual migration before or after upgrade is required.
QoS: Spoke to Spoke Per-tunnel QoS for DMVPN		<p>The QoS: Spoke to Spoke per tunnel QoS for DMVPN feature enables a DMVPN client to establish a direct crypto tunnel with another DMVPN client leveraging the per-tunnel QoS policy, using Next Hop Resolution Protocol (NHRP) to build spoke-to-spoke connections.</p> <p>The following commands were introduced or modified: nhrp attribute group, show dmvpn, show ip nhrp.</p> <p>Note The command show ip nhrp group is deprecated and is not in use.</p>
QoS: DMVPN Per-tunnel QoS over Aggregate GEC	Cisco IOS XE Everest 16.4.1	The QoS: DMVPN Per-tunnel QoS over Aggregate GEC feature is supported.



CHAPTER 12

Configuring TrustSec DMVPN Inline Tagging Support

The TrustSec DMVPN Inline Tagging Support feature enables IPsec to carry the Cisco TrustSec (CTS) Security Group Tag (SGT) between IPsec peers.

- [Prerequisites for Configuring TrustSec DMVPN Inline Tagging Support, on page 163](#)
- [Restrictions for Configuring TrustSec DMVPN Inline Tagging Support, on page 163](#)
- [Information About Configuring TrustSec DMVPN Inline Tagging Support, on page 164](#)
- [How to Configure TrustSec DMVPN Inline Tagging Support, on page 166](#)
- [Configuration Examples for TrustSec DMVPN Inline Tagging Support, on page 169](#)
- [Additional References for TrustSec DMVPN Inline Tagging Support, on page 173](#)
- [Feature Information for TrustSec DMVPN Inline Tagging Support, on page 174](#)

Prerequisites for Configuring TrustSec DMVPN Inline Tagging Support

Internet Key Exchange Version 2 (IKEv2) and IPsec must be configured on the router. For more information, see the “*Configuring Internet Key Exchange Version 2 and FlexVPN Site-to-Site*” and “*Configuring Security for VPNs with IPsec*” modules.

Restrictions for Configuring TrustSec DMVPN Inline Tagging Support

The TrustSec DMVPN Inline Tagging Support feature via IKEv2 supports the following:

- Dynamic Virtual Tunnel Interface (dVTI)
- GRE with Tunnel Protection
- Site-to-site VPNs
- Static crypto maps
- Static Virtual Tunnel Interface (sVTI)

The TrustSec DMVPN Inline Tagging Support feature does not support the following:

- Cisco AnyConnect
- Cisco VPNClient
- DMVPN with IKEv1
- EasyVPN
- FlexVPN
- GetVPN
- IKEv1 IPsec methods
- SSLVPN

crypto ikev2 cts sgt and **cts sgt inline** commands on tunnel are two different features. Do not configure these two features together as it causes the packets getting tagged twice. These commands are also not supported when the DMVPN is configured on MPLS using IKEv2.

cts sgt inline command does not rely on crypto or IKEv2. It can be configured statically or by NHRP. **cts sgt inline** command works with DMVPN IPSEC tunnel and also in transport mode.

The TrustSec DMVPN Inline Tagging Support feature via the **cts sgt inline** command is supported on all combinations of DMVPN (IKEv1, IKEv2, non-crypto, crypto accelerators such as ISM-VPN, point-to-point, multipoint) except when running MPLS (as an MPLS cloud extension or as MPLS L3VPN) over DMVPN.

Information About Configuring TrustSec DMVPN Inline Tagging Support

Cisco TrustSec

The Cisco TrustSec (CTS) architecture helps to build secure networks by establishing a domain of trusted network devices by combining identity, trust, and policy to protect user transactions and enforce role-based policies. CTS uses the user and the device identification information acquired during the authentication phase to classify packets as they enter the network. CTS maintains a classification of each packet by tagging packets on ingress to the CTS network so that they can be properly identified for applying security and other policy criteria along the data path. The packets or frames are tagged using the Security Group Tag (SGT), which allows network intermediaries such as switches and firewalls, to enforce an access control policy based on the classification.

The IPsec Inline Tagging for TrustSec feature is used to propagate the SGT to other network devices.



Note

If this feature is not supported, you can use the SGT Exchange Protocol over TCP (SXP) feature.

For more information on CTS and SXP, see the [Cisco TrustSec Switch Configuration Guide](#).

SGT and IPsec

IPsec uses the IKE protocol for negotiating algorithms, keys, and capabilities. IKEv2 is used to negotiate and inform IPsec about the SGT capability. Once the peers acknowledge the SGT tagging capability, an SGT tag number (a 16-bit) is added as the SGT Cisco Meta Data (CMD) payload into IPsec and sent to the receiving peer.

The access layer device authenticates the incoming packets. The access layer device receives an SGT from the authentication server and assigns the SGT along with an IP address to the incoming packets. In other words, an IP address is bound to an SGT. This IP address/SGT binding is propagated to upstream devices to enforce SGT-based policy and inline tagging.

If IKEv2 is configured to negotiate the SGT capability in the initiator, the initiator proposes the SGT capability information in the SA_INIT request. If IKEv2 is configured to negotiate the SGT capability in the responder, the responder acknowledges in the SA_INIT response and the initiator and the responder inform IPsec to use inline tagging for all packets to the peer.

During egress, IPsec adds the SGT capability and prefixes to the IPsec payload if the peer supports inline tagging; otherwise the packet is not tagged.

During ingress, IPsec inspects the packet for the SGT capability. If a tag is available, IPsec extracts the tag information and passes the information to the device only if inline tagging is negotiated. If there is no tag, IPsec processes the packet as a normal packet.

The tables below describe how IPsec behaves during egress and ingress.

Table 19: IPsec Behavior on the Egress Path

Inline Tagging Negotiated	CTS Provides SGT	IPsec Behavior
Yes	Yes	An SGT CMD is added to the packet.
Yes	No	The packet is sent without the SGT CMD.
No	Yes or no	The packet is sent without the SGT CMD.

Table 20: IPsec Behavior on the Ingress Path

Packet Is Tagged	Inline Tagging Negotiated	IPsec Behavior
Yes	Yes	The SGT CMD in the packet is processed.
Yes	No	The SGT CMD in the packet is not processed.
No	Yes or no	The packet is processed as a normal IPsec packet.

SGT on the IKEv2 Initiator and Responder

To enable SGT on an IKEv2 session, the SGT capability support must be sent to the peers using the **crypto ikev2 cts** command. SGT is a Cisco proprietary capability; hence, it is sent as a Vendor ID (VID) payload in the SA_INIT exchange.

The table below explains the scenarios when SGT capability is configured on the initiator and the responder:

Table 21: SGT Capability on IKEv2 Initiator and Responder

SGT Enabled on Initiator	SGT Enabled on Responder	What Happens . . .
Yes	Yes	The VID is exchanged between the initiator and the responder, and IPsec SA is enabled with the SGT inline tagging capability.
Yes	No	The initiator proposes the VID, but the responder ignores the VID. IPsec SA is not enabled with the SGT inline tagging capability.
No	Yes	The initiator does not propose the VID, and the responder does not send the VID payload. IPsec SA is not enabled with the SGT inline tagging capability.
No	No	The initiator does not propose the VID, and responder also does not send the VID payload. IPsec SA is not enabled with the SGT inline tagging capability.

Handling Fragmentation

Fragmentation is handled in the following two ways:

- Fragmentation before IPsec—If IPsec receives fragmented packets, each fragment is tagged.
- Fragmentation after IPsec—If IPsec packets are fragmented after encryption, the first fragment will be tagged.

How to Configure TrustSec DMVPN Inline Tagging Support

Enabling IPsec Inline Tagging

SUMMARY STEPS

1. enable
2. configure terminal
3. interface tunnel *tunnel id*
4. cts sgt inline
5. exit

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>tunnel id</i> Example: Device(config)# interface tunnel 1	Specifies a tunnel interface number, and enters interface configuration mode.
Step 4	cts sgt inline Example: Device(config-if)# cts sgt inline	Enables TrustSec on DMVPN. This command is valid for generic routing encapsulation (GRE) and to tunnel interfaces modes only.
Step 5	exit Example: Device(config)# exit	Exits global configuration mode.

Monitoring and Verifying TrustSec DMVPN Inline Tagging Support

To monitor and verify the TrustSec DMVPN Inline Tagging Support configuration, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show dmvpn**
3. **show ip nhrp nhs detail**
4. **show tunnel endpoints**
5. **show adjacency *interface-type interface-number detail***

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Step 2 **show dmvpn**

Example:

```
Device# show dmvpn
```

```
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket
T1 - Route Installed, T2 - Nexthop-override
C - CTS Capable
# Ent --> Number of NHRP entries with same NBMA peer
```

```
NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
UpDn Time --> Up or Down Time for a Tunnel
```

```
Interface: Tunnel0, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,
```

```
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
1 1.1.1.99 10.1.1.99 UP 00:00:01 SC
```

Use this command to display Dynamic Multipoint VPN (DMVPN)-specific session information.

Step 3 show ip nhrp nhs detail

Example:

```
Device# show ip nhrp nhs detail
```

```
Legend: E=Expecting replies, R=Responding, W=Waiting
```

```
Tunnel0:
```

```
10.1.1.99 RE NBMA Address: 1.1.1.99 priority = 0 cluster = 0 req-sent 44 req-failed 0 repl-recv
43 (00:01:37 ago)
TrustSec Enabled
```

Use this command to display Next Hop Resolution Protocol (NHRP) next hop server (NHS) information.

Step 4 show tunnel endpoints

Example:

```
Device# show tunnel endpoints
```

```
Tunnel0 running in multi-GRE/IP mode
```

```
Endpoint transport 1.1.1.99 Refcount 3 Base 0xF3FB79B4 Create Time 00:03:15
overlay 10.1.1.99 Refcount 2 Parent 0xF3FB79B4 Create Time 00:03:15
```

```
Tunnel Subblocks:
```

```
tunnel-nhrp-sb:
```

```
NHRP subblock has 1 entries; TrustSec enabled
```

Use this command to display the contents of the tunnel endpoint database that is used for tunnel endpoint address resolution, when running a tunnel in multipoint generic routing encapsulation (mGRE) mode.

Step 5 show adjacency interface-type interface-number detail

Example:

```
Device# show adjacency tunnel0 detail
```

```
Protocol Interface Address
IP Tunnel0 10.1.1.99(2)
0 packets, 0 bytes
epoch 0
sourced in sev-epoch 1
Encap length 32
4500000000000000FF2FB76901010101
01010163000089090800010100010000
Tun endpt
Next chain element:
```

```
.
.
.
```

Use this command to display information about the protocol.

Enabling IPsec Inline Tagging on IKEv2 Networks

Configuring the `cts sgt inline` and `crypto ikev2 cts sgt` commands results in the packets getting tagged twice - once each by each command.

Before you begin

IKEv2 and IPsec must be configured.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto ikev2 cts sgt`
4. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Device> <code>enable</code>	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>crypto ikev2 cts sgt</code> Example: Device(config)# <code>crypto ikev2 cts sgt</code>	Enables TrustSec on DMVPN on IKEv2 networks. This command is valid for generic routing encapsulation (GRE) and to tunnel interfaces modes only.
Step 4	<code>exit</code> Example: Device(config)# <code>exit</code>	Exits global configuration mode.

Configuration Examples for TrustSec DMVPN Inline Tagging Support

Example: Enabling IPsec Inline Tagging on IKEv2 Networks

Static VTI Initiator Configuration

The following example shows how to enable IPsec inline tagging on a static VTI initiator. You can use this configuration for configuring crypto maps and VTIs.

Example: Enabling IPsec Inline Tagging on IKEv2 Networks

```

crypto ikev2 proposal p1
  encryption 3des
  integrity md5
  group 2
!
crypto ikev2 policy policy1
  proposal p1
!
crypto ikev2 keyring key
  peer peer
    address ::/0
    pre-shared-key cisco
!
  peer v4
    address 0.0.0.0 0.0.0.0
    pre-shared-key cisco
!
!
!
crypto ikev2 profile prof3
  match identity remote address 0.0.0.0
  authentication local pre-share
  authentication remote pre-share
  keyring key
!
crypto ikev2 cts sgt
!
crypto ipsec transform-set trans esp-3des esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
  set peer 10.1.1.2
  set transform-set trans
  set ikev2-profile prof3
  match address ipv4acl
!
!
interface Loopback1
  ip address 209.165.201.1 255.255.255.224
  ipv6 address 2001::4:1/112
!
interface Loopback2
  ip address 209.165.200.1 255.255.255.224
  ipv6 address 2001::40:1/112
!
interface Embedded-Service-Engine0/0
  no ip address
  shutdown
!
interface GigabitEthernet0/0
  ip address 192.168.210.74 255.255.255.0
  duplex auto
  speed auto
!
interface GigabitEthernet0/1
  ip address 172.16.0.1 255.240.0.0
  duplex auto
  speed auto
  ipv6 address 2001::5:1/112
  ipv6 enable
  crypto map cmap
!
ip forward-protocol nd
!
no ip http server

```



```

no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 172.16.0.2
ip route 10.12.255.200 255.0.0.0 172.31.255.254
!
ip access-list extended ipv4acl
 permit ip host 209.165.201.1 host 192.168.12.125
 permit ip host 209.165.200.1 host 172.18.0.1
 permit ip host 172.28.0.1 host 10.10.10.1
 permit ip host 10.12.255.200 host 192.168.14.1
!
logging esm config
ipv6 route ::/0 2001::5:2
!
!
!
!!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
exception data-corruption buffer truncate
scheduler allocate 20000 1000

```

Dynamic VTI Responder Configuration

The following example shows how to enable IPsec inline tagging on a dynamic VTI responder. You can use this configuration for configuring crypto maps and VTIs.

```

crypto ikev2 proposal p1
 encryption 3des
 integrity md5
 group 2
!
crypto ikev2 policy policy1
 proposal p1
!
crypto ikev2 keyring key
 peer peer
 address 172.160.1.1 255.240.0.0
 pre-shared-key cisco
!
 peer v4_p2
 address 172.31.255.1 255.240.0.0
 pre-shared-key cisco
!
crypto ikev2 profile prof
 match identity remote address 0.0.0.0
 authentication local pre-share

```

Example: Enabling IPsec Inline Tagging on IKEv2 Networks

```

authentication remote pre-share
keyring key
virtual-template 25
!
crypto ikev2 cts sgt
!
crypto ipsec transform-set trans esp-null esp-sha-hmac
!
crypto ipsec profile prof_ipv4
set transform-set trans
set ikev2-profile prof1_ipv4
!
!
interface Loopback0
ip address 192.168.12.1 255.255.0.0
!
interface Loopback1
no ip address
!
interface Loopback2
ip address 172.18.0.1 255.240.0.0
!
interface Loopback10
no ip address
ipv6 address 2001::8:1/112
!
interface Loopback11
no ip address
ipv6 address 2001::80:1/112
!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
ip address 10.1.1.2 255.0.0.0
duplex auto
speed auto
ipv6 address 2001::7:1/112
ipv6 enable
!
interface GigabitEthernet0/1
ip address 10.10.10.2 255.255.255.0
duplex auto
speed auto
!
interface GigabitEthernet0/2
ip address 192.168.210.144 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/0/0
no ip address
shutdown
!
interface FastEthernet0/0/1
no ip address
!
interface FastEthernet0/0/2
no ip address
!
interface FastEthernet0/0/3
no ip address
!

```

```

!
interface Virtual-Template25 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile prof_ipv4
!
interface Vlan1
 no ip address
!
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
ip route 172.17.0.0 255.240.0.0 10.10.10.1
!
logging esm config
ipv6 route ::/0 2001::7:2
!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
exception data-corruption buffer truncate
scheduler allocate 20000 1000
end

```

Additional References for TrustSec DMVPN Inline Tagging Support

Related Documents

Related Topic	Document Title
Security commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Related Topic	Document Title
Cisco TrustSec and SXP configuration	Cisco TrustSec Switch Configuration Guide
IPsec configuration	Configuring Security for VPNs with IPsec
IKEv2 configuration	Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site
Cisco Secure Access Control Server	Configuration Guide for the Cisco Secure ACS

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 TrustSec DMVPN Inline Tagging Support

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 22: Feature Information for Configuring TrustSec DMVPN Inline Tagging Support

Feature Name	Releases	Feature Information
TrustSec DMVPN Inline Tagging Support		<p>The TrustSec DMVPN Inline Tagging Support feature enables IPsec to carry Cisco Trust Sec (CTS) Security Group Tag (SGT) between IPsec peers.</p> <p>The following commands were introduced or modified: cts sgt inline, show dmvpn, show ip nhrp nhs, show tunnel endpoints, show adjacency.</p>



CHAPTER 13

Spoke-to-Spoke NHRP Summary Maps

The Spoke-to-Spoke NHRP Summary Maps feature summarizes and reduces the NHRP resolution traffic on the network.

- [Finding Feature Information, on page 175](#)
- [Information About Spoke-to-Spoke NHRP Summary Maps, on page 175](#)
- [How to Configure Spoke-to-Spoke NHRP Summary Maps, on page 177](#)
- [Configuration Examples for Spoke-to-Spoke NHRP Summary Maps, on page 181](#)
- [Additional References for Spoke-to-Spoke NHRP Summary Maps, on page 183](#)
- [Feature Information for Spoke-to-Spoke NHRP Summary Maps, on page 183](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Information About Spoke-to-Spoke NHRP Summary Maps

Spoke-to-Spoke NHRP Summary Maps

In DMVPN phase 3, route summarization is performed at a hub. The hub is the next-hop for any spoke to reach any network behind a spoke. On receiving a packet, the hub sends a redirect message to a local spoke and indicates the local spoke to send Next Hop Resolution Protocol (NHRP) resolution request for the destination network. The resolution request is forwarded by the hub to a remote spoke with the destination LAN network. The remote spoke responds to the resolution request and initiates a tunnel with the local spoke.

When a spoke answers an NHRP resolution request for a local host, it uses the explicit IP address network and subnet mask from the Routing Information Base (RIB) in response. Multiple networks behind a local spoke require similar NHRP messages for a host behind remote spoke to exchange packets with the hosts in these networks. It is difficult to handle NHRP messages for a huge number of spokes and large networks behind each spoke.

The number of NHRP messages between spokes can be limited when the first NHRP resolution reply provides information about the network behind a local spoke instead of a specific network. The spoke-to-spoke NHRP summary map uses the configured IP address network and subnet mask in the NHRP resolution response instead of the IP address network and subnet mask from RIB. If RIB has more number of IP address networks (lesser subnet mask length) than the configured IP address network and subnet mask, the spoke still uses the configured IP address network and subnet mask for NHRP resolution response thereby summarizing and reducing the NHRP resolution traffic on the network. Use the **ip nhrp summary-map** command to configure NHRP summary map on a spoke.



Note In DMVPN, it is recommended to configure a Rendezvous Point (RP) at or behind the hub. If there is an IP multicast source behind a spoke, the **ip pim spt-threshold infinity** command must be configured on spokes to avoid multicast traffic going through spoke-to-spoke tunnels.

How Spoke-to-Spoke NHRP Summary Maps Works

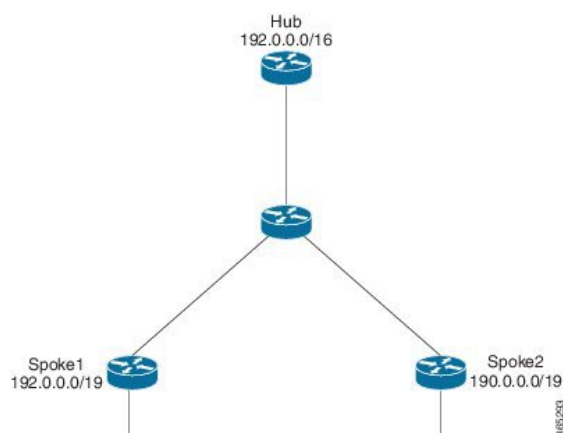
On receiving the resolution request, the spoke

1. Looks into the RIB for the IP address and subnet mask and returns.
2. Checks the IP address and subnet mask against the configured NHRP summary map and verifies if the destination IP address is covered.
3. Sends the summary map in the NHRP resolution reply to the remote spoke and NHRP on the remote spoke adds the IP address and subnet mask with the next-hop of the local spoke to the RIB.

The entire network behind the local spoke is identified to the remote spoke with one NHRP resolution request.

The following figure shows the working of spoke-to-spoke NHRP summary maps.

Figure 9: Spoke-to-Spoke NHRP Summary Maps



A local spoke with the address space 192.0.0.0/19 on its local LAN has all 32-24 RIB entries – 192.0.0.0/24, ..., 192.0.31.0/24. When a routing protocol like EIGRP is used to advertise this local address space, the routing protocol is configured to summarize the networks to 192.0.0.0/19 and advertise that to the hub. The hub summarizes this further, to 192.0.0.0/16, when it advertises it to the other spokes. The other spokes start with only a 192.0.0.0/16 routing table entry with the next-hop of the hub in the RIB.

If a remote host communicates with 192.0.12.1, the local spoke receives the NHRP resolution request for 192.0.12.1/32. It looks into the RIB and returns 192.0.12.0/24 in NHRP resolution reply.

If the local spoke is configured with NHRP summary map for eg. "ip nhrp summary-map 192.0.0.0/19", the local spoke upon receiving the resolution request for 192.0.12.1 checks the RIB which returns 192.0.12.0/24. The local spoke then checks for summary map configuration 192.0.0.0/19 and verifies if the destination 192.0.12.1/32 is covered and returns 192.0.0.0/19 in NHRP resolution reply.

NHRP Summary Map Support for IPv6 Overlay

Spoke-to-spoke NHRP summary maps feature is supported on IPv6 and is configured using **ipv6 nhrp summary-map** command.

How to Configure Spoke-to-Spoke NHRP Summary Maps

Configuring Spoke-to-Spoke NHRP Summary Maps on Spoke



Note The following task can be performed to configure the spoke device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** *ip-address mask secondary ip-address mask*
5. **ip nhrp authentication** *string*
6. **ip nhrp summary-map** {*ip-address* | *mask*}
7. **ip nhrp network-id** *number*
8. **ip nhrp nhs** [*hub-tunnel-ip-address*] **nbma** [*hub-wan--ip*] **multicast**
9. **ip nhrp shortcut**
10. **tunnel source** {*ip-address* | *type number*}
11. **tunnel mode gre multipoint**
12. **tunnel key** *key-number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: <pre>Device(config)# interface tunnel 5</pre>	Configures a tunnel interface and enters interface configuration mode. <ul style="list-style-type: none"> • <i>number</i>—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	ip address <i>ip-address mask secondary ip-address mask</i> Example: <pre>Device(config-if)# ip address 10.0.0.2 255.255.255.0</pre>	Sets a primary or secondary IP address for the tunnel interface. <p>Note All hubs and spokes that are in the same DMVPN network must be addressed in the same IP subnet.</p>
Step 5	ip nhrp authentication <i>string</i> Example: <pre>Device(config-if)# ip nhrp authentication donttell</pre>	Configures an authentication string for an interface using NHRP.
Step 6	ip nhrp summary-map {<i>ip-address mask</i>} Example: <pre>Device(config-if)# ip nhrp summary-map 10.0.0.0/24</pre>	Summarizes and reduces the NHRP resolution traffic on the network.
Step 7	ip nhrp network-id <i>number</i> Example: <pre>Device(config-if)# ip nhrp network-id 99</pre>	Enables NHRP on an interface. <ul style="list-style-type: none"> • <i>number</i>—Specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network.
Step 8	ip nhrp nhs [<i>hub-tunnel-ip-address</i>] nbma [<i>hub-wan--ip</i>] multicast Example: <pre>Device(config-if)# ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast</pre>	Configures the hub router as the NHRP next-hop server.
Step 9	ip nhrp shortcut Example: <pre>Device(config-if)# ip nhrp shortcut</pre>	Enables NHRP shortcut switching.

	Command or Action	Purpose
Step 10	tunnel source <i>{ip-address type number}</i> Example: <pre>Device(config-if)# tunnel source Gigabitethernet 0/0/0</pre>	Sets the source address for a tunnel interface.
Step 11	tunnel mode gre multipoint Example: <pre>Device(config-if)# tunnel mode gre multipoint</pre>	Sets the encapsulation mode to Multiple Generic Routing Encapsulation (mGRE) for the tunnel interface. <ul style="list-style-type: none"> Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 12	tunnel key <i>key-number</i> Example: <pre>Device(config-if)# tunnel key 100000</pre>	(Optional) Enables an ID key for a tunnel interface. <ul style="list-style-type: none"> <i>key-number</i>—Specifies a number to identify a tunnel key. This must be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 13	end Example: <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Spoke-to Spoke NHRP Summary Maps

SUMMARY STEPS

- enable
- show ip nhrp

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 show ip nhrp

Example:

The following is an example of show command output on spoke.

```
Device# show ip nhrp
```

```

15.0.0.1/32 (vrf1) via 15.0.0.1
  Tunnel3 created 09:09:00, never expire
  Type: static, Flags: used
  NBMA address: 123.0.0.1
15.0.0.20/32 (vrf1) via 15.0.0.20
  Tunnel3 created 00:00:54, expire 00:04:05
  Type: dynamic, Flags: router nhop rib
  NBMA address: 42.0.0.1
190.0.0.0/22 (vrf1) via 15.0.0.10
  Tunnel3 created 09:09:00, never expire
  Type: static, Flags: local
  NBMA address: 121.0.0.1
  (no-socket)
201.0.0.0/22 (vrf1) via 15.0.0.20
  Tunnel3 created 00:00:54, expire 00:04:05
  Type: dynamic, Flags: router rib nho
  NBMA address: 42.0.0.1

```

Displays Next Hop Resolution Protocol (NHRP) mapping information.

Troubleshooting Spoke-to-Spoke NHRP Summary Maps

SUMMARY STEPS

1. debug dmvpn all nhrp

DETAILED STEPS

debug dmvpn all nhrp

Checks the IP address and subnet mask received by the spoke for a resolution request.

Example:

```

Device# debug dmvpn all nhrp

NHRP-RT: Attempting to create instance PDB for vrf global(0x0) (0x0)
NHRP-CACHE: Tunnel0: Cache add for target 67.0.0.1/32 vrf global(0x0) label none next-hop 67.0.0.1

NHRP-CACHE: Tunnel0: Cache add for target 67.0.0.0/24 vrf global(0x0) label none next-hop 15.0.0.30
80.0.0.1
NHRP-CACHE: Inserted subblock node(2 now) for cache: Target 67.0.0.0/24 nhop 15.0.0.30
NHRP-CACHE: Converted internal dynamic cache entry for 67.0.0.0/24 interface Tunnel0 vrf global(0x0)
to external
NHRP-RT: Adding route entry for 67.0.0.0/24 (Tunnel0 vrf:global(0x0)) to RIB
NHRP-RT: Route addition to RIB Successful
NHRP-RT: Route watch started for 67.0.0.0/23
NHRP-CACHE: Updating label on Tunnel0 for 15.0.0.30 vrf global(0x0), old none new none nhop 15.0.0.30
NHRP-CACHE: Tunnel0: Cache update for target 15.0.0.30/32 vrf global(0x0) label none next-hop 15.0.0.30
80.0.0.1

NHRP-CACHE: Deleting incomplete entry for 67.0.0.1/32 interface Tunnel0 vrf global(0x0)
NHRP-CACHE: Still other cache entries with same overlay nhop 67.0.0.1
NHRP-RT: Received route watch notification for 67.0.0.0/24
NHRP-RT: Covering prefix is 67.0.0.0/22
NHRP-RT: Received route watch notification for 67.0.0.0/24

```

```
NHRP-RT: (0x0):NHRP RIB entry for 67.0.0.0/24 is unreachable
```

Configuration Examples for Spoke-to-Spoke NHRP Summary Maps

Example: Spoke-to-Spoke NHRP Summary Maps

Example: Spoke-to-Spoke NHRP Summary Maps

The following is an example of configuring DMVPN phase 3 on hub for summary map .

```
interface Tunnel0
 ip address 15.0.0.1 255.255.255.0
 no ip redirects
 no ip split-horizon eigrp 2
 ip nhrp authentication cisco123
 ip nhrp network-id 23
 ip nhrp redirect
 ip summary-address eigrp 2 190.0.0.0 255.255.252.0
 ip summary-address eigrp 2 201.0.0.0 255.255.252.0
 tunnel source GigabitEthernet1/0/0
 tunnel mode gre multipoint
 tunnel key 6
end
```

The following example shows how to configure spoke-to-spoke NHRP summary maps on spoke 1.

```
interface Tunnel0
 vrf forwarding vrf1
 ip address 15.0.0.10 255.255.255.0
 ip nhrp authentication cisco123
 ip nhrp summary-map 190.0.0.0/22
 ip nhrp network-id 5
 ip nhrp nhs 15.0.0.1 nbma 123.0.0.1 multicast
 ip nhrp shortcut
 tunnel source GigabitEthernet0/1/0
 tunnel mode gre multipoint
 tunnel key 6
end
```

The following example shows how to configure spoke-to-spoke NHRP summary maps on spoke 2.

```
interface Tunnel0
 ip address 15.0.0.20 255.255.255.0
 ip nhrp authentication cisco123
 ip nhrp summary-map 201.0.0.0/22
```

Example: Spoke-to-Spoke NHRP Summary Maps

```

ip nhrp network-id 5
ip nhrp nhs 15.0.0.1 nbma 123.0.0.1 multicast
ip nhrp shortcut
tunnel source GigabitEthernet0/0/0
tunnel mode gre multipoint
tunnel key 6
end

```

The following is a sample output of the show ip nhrp command on the hub.

```

Device# show ip nhrp

15.0.0.10/32 via 15.0.0.10
  Tunnel0 created 00:22:26, expire 00:07:35
  Type: dynamic, Flags: registered used nhop
  NBMA address: 41.0.0.1
15.0.0.20/32 via 15.0.0.20
  Tunnel0 created 00:13:43, expire 00:09:36
  Type: dynamic, Flags: registered used nhop
  NBMA address: 42.0.0.1

```

The following is a sample output of the show ip nhrp command on spoke 1.

```

Device# show ip nhrp

15.0.0.1/32 (vrf1) via 15.0.0.1
  Tunnel3 created 09:09:00, never expire
  Type: static, Flags: used
  NBMA address: 123.0.0.1
15.0.0.20/32 (vrf1) via 15.0.0.20
  Tunnel3 created 00:00:54, expire 00:04:05
  Type: dynamic, Flags: router nhop rib
  NBMA address: 42.0.0.1
190.0.0.0/22 (vrf1) via 15.0.0.10
  Tunnel3 created 09:09:00, never expire
  Type: static, Flags: local
  NBMA address: 121.0.0.1
  (no-socket)
201.0.0.0/22 (vrf1) via 15.0.0.20
  Tunnel3 created 00:00:54, expire 00:04:05
  Type: dynamic, Flags: router rib nho
  NBMA address: 42.0.0.1

```

The following is a sample output of the show ip nhrp command on spoke 2.

```

Device# show ip nhrp

15.0.0.1/32 via 15.0.0.1
  Tunnel0 created 09:08:16, never expire
  Type: static, Flags: used
  NBMA address: 123.0.0.1
15.0.0.10/32 via 15.0.0.10
  Tunnel0 created 00:00:04, expire 01:59:55
  Type: dynamic, Flags: router nhop rib
  NBMA address: 121.0.0.1
190.0.0.0/22 via 15.0.0.10
  Tunnel0 created 00:00:04, expire 01:59:55
  Type: dynamic, Flags: router rib nho
  NBMA address: 121.0.0.1

```

```

201.0.0.0/22 via 15.0.0.20
Tunnel0 created 09:08:16, never expire
Type: static, Flags: local
NBMA address: 42.0.0.1
(no-socket)

```

Additional References for Spoke-to-Spoke NHRP Summary Maps

Related Documents

Related Topic	Document Title
Cisco IOS security commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z

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 Spoke-to-Spoke NHRP Summary Maps

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 23: Feature Information for Spoke-to-Spoke NHRP Summary Maps

Feature Name	Releases	Feature Information
Spoke-to-Spoke NHRP Summary Maps		<p>The Spoke-to-Spoke Next Hop Resolution Protocol (NHRP) Summary Maps feature summarizes and reduces the NHRP resolution traffic on the network.</p> <p>The following commands were introduced or modified by this feature: ip nhrp summary-map, ipv6 summary-map.</p>



CHAPTER 14

BFD Support on DMVPN

Bidirectional Forwarding Detection (BFD) support on DMVPN provides fast peer failure detection by sending rapid failure detection notices to the control protocols and reducing overall network convergence time.

- [Finding Feature Information](#), on page 185
- [Prerequisites for BFD Support on DMVPN](#), on page 185
- [Restrictions for BFD Support on DMVPN](#), on page 185
- [Information About BFD Support on DMVPN](#), on page 186
- [How to Configure BFD Support on DMVPN](#), on page 186
- [Example: BFD Support on DMVPN](#), on page 187
- [Additional References for BFD Support on DMVPN](#), on page 191
- [Feature Information for BFD Support on DMVPN](#), on page 191

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Prerequisites for BFD Support on DMVPN

BFD for DMVPN supports both IPv4 and IPv6 overlay address and is agnostic to transport address family.

For more BFD prerequisites refer [Prerequisites for Bidirectional Forwarding Detection](#)

Restrictions for BFD Support on DMVPN

- NHRP currently acts only on BFD down events and not on up events.
- Both peers must configure BFD to get BFD support. If one of the peers is not configured with BFD, the other peer creates BFD sessions in down or unknown state.

- Before configuring BFD support on DMVPN, in case of point-to-point (P2P) tunnel, next hop server (NHS) must be configured.
- BFD intervals configured on the peers should be the same in the BFD echo mode for spoke to spoke refresh to work as expected.
- A single DMVPN hub with BFD can be scaled to a maximum of 4095 sessions on a Cisco Aggregation Service Router 1000 Series since the number of BFD sessions on these platforms is limited to 4095 currently. Regular methods of scaling DMVPN like clustering, Server Load Balancing (SLB), hierarchical designs, etc still apply. This does not impact DMVPN scale without BFD.

Information About BFD Support on DMVPN

BFD Operation

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent routers, including the interfaces, data links, and forwarding planes.

BFD is a detection protocol that is enabled at the interface and protocol levels. Cisco supports BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers. Therefore, in order for a BFD session to be created, BFD must be configured on both systems (or BFD peers). Once BFD has been enabled on the interfaces and at the router level for the appropriate protocols (NHRP and the routing protocol on overlay), a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

Benefits of BFD Support on DMVPN

- Faster detection of link failure.
- In non-crypto deployments, spoke can detect hub failure only after NHRP registration timeout but hub cannot detect a spoke failure until cache on hub expires (even though routing can re-converge much earlier). BFD allows for a very fast detection for such a failure.
- BFD validates the forwarding path between non authoritative sessions, for example, in scenarios where the hub is configured to respond on behalf of the spoke.
- BFD validates end-to-end data path including the tunnel unlike IKE keepalives/DPD that doesn't pass through the tunnel.
- BFD probes can be off-loaded.

There is no special NHRP configuration needed for BFD support on DMVPN, enabling BFD on an NHRP enabled interface suffices. For DMVPN configuration refer [How to Configure Dynamic Multipoint VPN](#)

How to Configure BFD Support on DMVPN

Configuring BFD Support on DMVPN

BFD intervals can be directly configured on tunnel interface as shown below:


```

enable
configure terminal
interface tunnell
bfd interval 1000 min_rx 1000 multiplier 5
no echo

```

BFD intervals can also be configured by defining a template and attaching it to the tunnel interface as shown below

```

enable
configure terminal
bfd-template single-hop sample
interval min-tx 1000 min-rx 1000 multiplier 5
interface tunnell
bfd template sample

```

Example: BFD Support on DMVPN

Example: BFD Support on DMVPN

The following is an example of configuring BFD support on DMVPN on hub.

```

bfd-template single-hop sample
  interval min-tx 1000 min-rx 1000 multiplier 5
!
interface Tunnel0
  ip address 10.0.0.1 255.255.255.0
  no ip redirects
  ip nhrp authentication cisco123
  ip nhrp network-id 5
  ip nhrp redirect
  ip mtu 1400
  ip tcp adjust-mss 1360
  bfd template sample
  tunnel source GigabitEthernet0/0/0
  tunnel mode gre multipoint
  tunnel key 6
!
interface GigabitEthernet0/0/0
  ip address 10.0.0.1 255.0.0.0
  negotiation auto
!
router eigrp 2
network 10.0.0.0 0.0.0.255
bfd all-interfaces
auto-summary
!

```

The following is an example of configuring BFD support on DMVPN on spoke.

```

bfd-template single-hop sample
  interval min-tx 1000 min-rx 1000 multiplier 5

```

```

!
interface Tunnell
 ip address 10.0.0.10 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp network-id 5
 ip nhrp nhs 10.0.0.1 nbma 10.0.0.10 multicast
 bfd template sample
 tunnel source GigabitEthernet0/0/0
 tunnel mode gre multipoint
 tunnel key 6
!
interface GigabitEthernet0/0/0
 mtu 4000
 ip address 11.0.0.1 255.0.0.0
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/0/1
 mtu 6000
 ip address 111.0.0.1 255.255.255.0
 negotiation auto
!
router eigrp 2
 network 11.0.0.0 0.0.0.255
 network 111.0.0.0 0.0.0.255
 network 10.0.0.0 0.0.0.255
 bfd all-interfaces
 auto-summary
!
 ip route 0.0.0.0 0.0.0.0 10.0.0.2

```

The following is an example to illustrate faster convergence on spoke.

```

interface Tunnell
 ip address 18.0.0.10 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp network-id 12
 ip nhrp nhs 10.0.0.1 nbma 10.0.0.10 multicast
 bfd template sample
 tunnel source GigabitEthernet0/0/0
 tunnel mode gre multipoint
 tunnel key 18
 tunnel protection ipsec profile MY_PROFILE
!
bfd-template single-hop sample
 interval min-tx 1000 min-rx 1000 multiplier 3
 echo
!
router eigrp 2
 bfd interface Tunnell -----> Specify the interface on which the routing
  protocol must act for BFD up/down events
 network 11.0.0.0 0.0.0.255
 network 111.0.0.0 0.0.0.255

```

With the above configuration, as soon as BFD is reported down (3 seconds to detect), EIGRP will remove the routes installed from RIB.

The following sample output shows a summary output on hub:

```

device#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel1, IPv4 NHRP Details
Type:Hub, NHRP Peers:2,

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
  1 172.17.0.1          10.0.0.1   UP 00:00:14   D
  1 172.17.0.2          10.0.0.2   BFD 00:00:03   D

```

BFD is a new state which implies that while the session is UP as seen by lower layers (IKE, IPSec and NHRP), BFD sees the session as DOWN. As usual, the state is an indication of the lower most layer where the session is not UP. Also, this applies only to the parent cache entry. This could be because it was detected as DOWN by BFD or BFD is not configured on the other side.

The following sample output shows a summary output on spoke:

```

device#show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface: Tunnel2, IPv4 NHRP Details
Type:Spoke, NHRP Peers:2,

# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb
-----
  2 172.17.0.2          10.0.0.2   BFD 00:00:02   DT1
  2 172.17.0.2          10.0.0.2   UP 00:00:02   DT2
  1 172.17.0.11         10.0.0.11  UP 00:05:35   S

```

The following sample shows output for **show ip/ipv6 nhrp** command

```

device#show ip nhrp
10.0.0.2/32 via 10.0.0.2
  Tunnel2 created 00:00:15, expire 00:04:54
  Type: dynamic, Flags: router nhop rib bfd
  NBMA address: 172.17.0.2
10.0.0.11/32 via 10.0.0.11
  Tunnel2 created 00:09:04, never expire
  Type: static, Flags: used bfd
  NBMA address: 172.17.0.11
192.168.1.0/24 via 10.0.0.1
  Tunnel2 created 00:00:05, expire 00:04:54

```

```

Type: dynamic, Flags: router unique local
NBMA address: 172.17.0.1
(no-socket)
192.168.2.0/24 via 10.0.0.2
Tunnel2 created 00:00:05, expire 00:04:54
Type: dynamic, Flags: router rib nho
NBMA address: 172.17.0.2

```

BFD flag here implies that there is a BFD session for this peer. This marking is only for parent entries.

The following sample shows output for **show tunnel endpoints** command

```

device#show tunnel endpoints
Tunnel2 running in multi-GRE/IP mode

Endpoint transport 172.17.0.2 Refcount 3 Base 0x2ABF53ED09F0 Create Time 00:00:07
overlay 10.0.0.2 Refcount 2 Parent 0x2ABF53ED09F0 Create Time 00:00:07
Tunnel Subblocks:
  tunnel-nhrp-sb:
    NHRP subblock has 2 entries; BFD(0x2):U
Endpoint transport 172.17.0.11 Refcount 3 Base 0x2ABF53ED0B80 Create Time 00:09:07
overlay 10.0.0.11 Refcount 2 Parent 0x2ABF53ED0B80 Create Time 00:09:07
Tunnel Subblocks:
  tunnel-nhrp-sb:
    NHRP subblock has 1 entries; BFD(0x1):U

```

For every tunnel endpoint, a new text "**BFD(handle):state**" is added. State here is UP(U), DOWN(D), NONE(N) or INVALID(I).

- In case, BFD is not configured on peer or a session is not UP for the first time, then the state will be N.

The following sample shows output for **show nhrp interfaces** command. This shows the configuration (and not operational) states on the interface or globally.

```

device#show nhrp interfaces
NHRP Config State
-----
Global:
  BFD: Registered

Tunnel1:
  BFD: Disabled

Tunnel2:
  BFD: Enabled

```

This is an internal and hidden command. This will currently display if NHRP is client of BFD and if BFD is enabled on the NHRP interface.

Additional References for BFD Support on DMVPN

Related Documents

Related Topic	Document Title
Dynamic Multipoint VPN Configuration Guide	Dynamic Multipoint VPN Configuration Guide
IP Routing: BFD Configuration Guide	IP Routing: BFD Configuration Guide

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> • CISCO-MIB • NHRP MIB • Cisco NHRP Extension MIB • BFD MIB • Tunnel MIB • IPSec MIBs 	<p>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

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 BFD Support on 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 24: Feature Information for BFD Support on DMVPN

Feature Name	Releases	Feature Information
BFD Support on DMVPN	Cisco IOS Release 16.3	<p>Bidirectional Forwarding Detection (BFD) support on DMVPN feature provides fast peer failure detection by sending rapid failure detection notices to the routing protocols and reducing overall network convergence time.</p> <p>The following commands were modified by this feature: show dmvpn, show ip nhrp, show ipv6 nhrp, show tunnel endpoints, show nhrp interfaces.</p>



CHAPTER 15

DMVPN Support for IWAN

DMVPN supports Cisco Intelligent WAN architecture to provide transport independence through overlay routing. The DMVPN Multiple Tunnel Termination feature enables support for secondary paths for the supported routing protocols in the Routing Information Base (RIB).

- [Prerequisites for DMVPN Support for IWAN, on page 193](#)
- [Restrictions for DMVPN Support for IWAN, on page 193](#)
- [Information About DMVPN Support for IWAN, on page 193](#)
- [How to Configure DMVPN Support for IWAN, on page 196](#)
- [Configuration Examples for DMVPN Support for IWAN, on page 197](#)
- [Additional References for DMVPN Support for IWAN, on page 202](#)
- [Feature Information for DMVPN Support for IWAN, on page 203](#)

Prerequisites for DMVPN Support for IWAN

For DMVPN Multiple Tunnel Termination feature to work, the following prerequisites must be considered

- DMVPN Multiple Tunnel Termination feature requires support from crypto maps and DMVPN.
- Only BGP and EIGRP routing protocols are supported on this feature. One of the two routing protocols, BGP and EIGRP, must be enabled for this feature to work.

Restrictions for DMVPN Support for IWAN

For DMVPN Multiple Tunnel Termination feature the overlay routing should be active-passive in nature.

Information About DMVPN Support for IWAN

Transport Independence

DMVPN supports Cisco IWAN by providing transport independence through overlay routing. Overlay routing simplifies the WAN transport (dial-up, leased circuits, MPLS, and IPsec VPNs), by deploying and supporting

consistent routing protocol across any transport, controlling traffic and load sharing. Overlay routing provides transport independence so that the user can select any WAN technology.

Transport independence eases change in transport options and service providers. Changing transports does not impact the overlay routing design. This technology supports use of multiple WAN transports, as the transport type is associated to the underlay network and is not relevant to the overlay network which is consistent to the DMVPN tunnel.

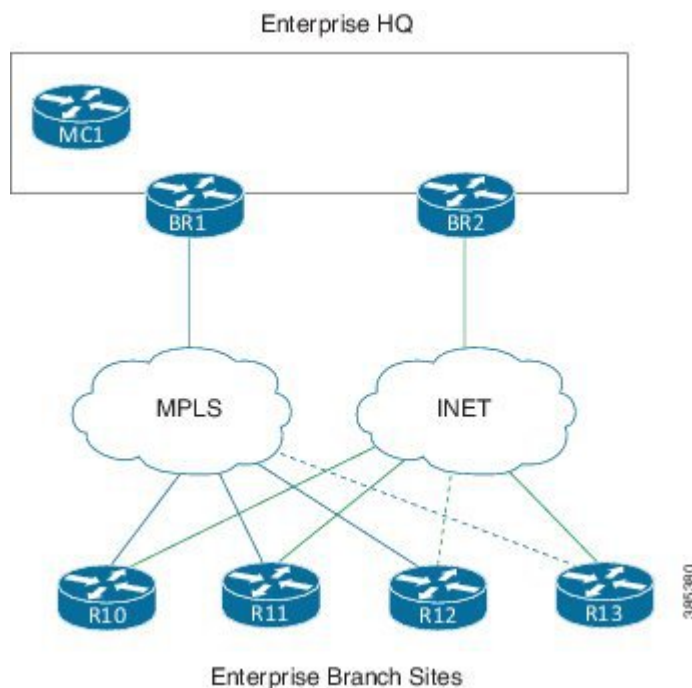
Transport independence provides single routing domain, consistent troubleshooting and topology for WAN transports. As long as the transport network delivers the DMVPN packets between the hub and the spoke, the transport device topology is not relevant to the traffic flowing across the DMVPN tunnel.

DMVPN for IWAN

DMVPN uses multipoint generic routing encapsulation (mGRE) tunnels to interconnect the hubs and all of the spokes. For IWAN deployments, DMVPN provides integration with PfR and simplifies route control across any transport. DMVPN supports full mesh connectivity over any carrier transport with a simple hub-and-spoke configuration. DMVPN also supports spoke that have dynamically assigned IP addresses.

The following figure shows IWAN deployments with multiple WAN transports. This design enables convergence across WAN transports when all channels in a given transport fail or reach their maximum bandwidth limits.

Figure 10: DMVPN for IWAN



Secondary Paths

For a single tunnel case, the routing method installs multiple paths in the RIB, one or more leaving each tunnel. Based on the configuration, this includes some or all of the available free paths. The paths can be classified into following classes:

- Regular next-hops/paths are the most common kind of paths. They are also referred to as primary paths; other alternate next-hops are sometimes referred to as secondary paths.
- Repair next-hops/paths forward traffic during a routing transition and are not used as long as one or more regular next hops are active.
- Secondary next-hops/paths are special loop free paths that is used as an alternate to regular and repair paths.

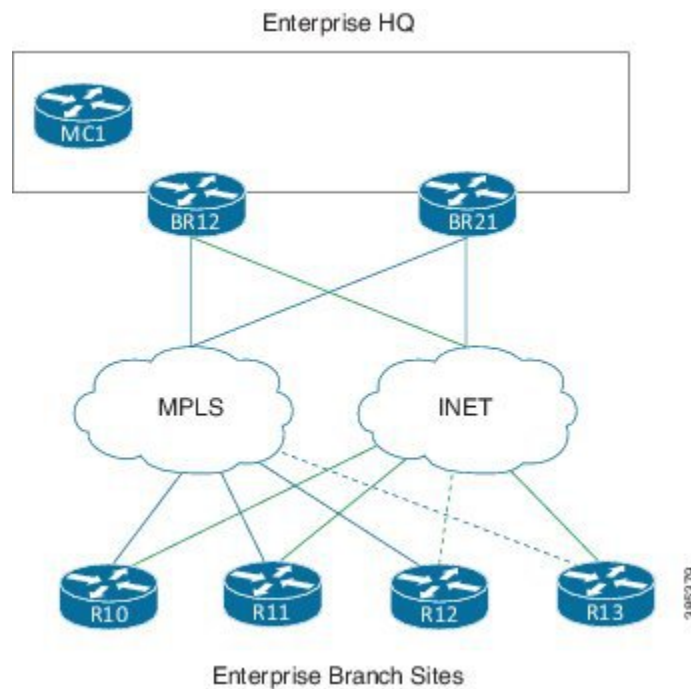
When at least one of the primary paths are in use, the secondary paths are not used for regular forwarding. The secondary paths should be distinguishable from other regular and alternate paths. The secondary paths can still be overridden using next hop overrides. The routing protocol computes "n" secondary paths with the following requirement from RIB:

- Allow the routing protocol to install the "n1" primary paths as a regular path
- Allow the routing protocol to install the "n2" secondary paths as alternate paths.

DMVPN Multiple Tunnel Termination

Network access resiliency at a single hub in Cisco IWAN without having to add any network devices, involves terminating multiple WAN links on the same device. The DMVPN Multiple Tunnel Termination feature provides support for multiple tunnel terminations (interfaces) in the same VRF on the same hub device.

Figure 11: DMVPN Multiple Tunnel Termination



The DMVPN Multiple Tunnel Termination feature also provides transport resiliency to DMVPN. Using one tunnel per-transport provides better visibility to Performance Routing (PfR), about the conditions in the underlying transport and still being transport independent. IWAN as a whole is transport independent along with the services running on the overlay.

DMVPN Multiple Tunnel Termination feature brings in support for secondary paths for the supported routing protocols in the RIB. The routing protocols are configured in such a way that there is only one primary/regular path and one or more secondary paths for a network. When PfR is used in conjunction with this feature, PfR

is used as the primary as well the secondary path so that all paths can be used in an active-active manner. Use the **maximum-secondary-paths** [**eigrp** | **ibgp**] *path* command to configure this feature, where the *path* indicates the number of secondary paths a routing protocol is allowed to install. The range for *path* is from zero to 32.

How to Configure DMVPN Support for IWAN

Configuring DMVPN Support for IWAN

Perform this task to configure IPsec profile on the device.

```
crypto ikev2 keyring keyring1
peer peer1
  address 0.0.0.0 0.0.0.0
  pre-shared-key key1

crypto ikev2 proposal proposal1
  encryption aes-cbc-128
  prf sha256 sha512
  group 14

crypto ikev2 policy proposal1
  match fvrf vrf1
  proposal proposal1

crypto ikev2 profile profile1
  description This is an IKEv2 profile
  match fvrf vrf1
  match identity remote address 10.0.0.1
  identity local address 10.0.0.0
  authentication remote pre-share
  authentication local pre-share
  keyring local key1

crypto ipsec transform-set transform1 esp-gcm 256
  mode transport

crypto ipsec profile profile2
  set transform-set esp-gcm 256
  set ikev2-profile profile1

crypto ipsec security-association replay window-size 15
```

Perform this task to configure the tunnel.

```
interface Tunnel 10
  tunnel source GigabitEthernet0/0/0
  tunnel mode gre multipoint
  tunnel key 10000
  tunnel vrf vrf1
  tunnel protection ipsec profile profile2
```

Perform the following task to configure BGP routing process.

```

router bgp 45000
  bgp router-id 172.17.1.99
  bgp log-neighbor-changes
  timers bgp 70 120
  neighbor 192.168.1.2 remote-as 40000
  neighbor 192.168.3.2 remote-as 50000
  neighbor 192.168.3.2 description finance
  !
  address-family ipv4
    neighbor 192.168.1.2 activate
    neighbor 192.168.3.2 activate
    no auto-summary
    no synchronization
    network 172.17.1.0 mask 255.255.255.0
  exit-address-family
  !

```

Configuring DMVPN Multiple Tunnel Termination

Perform the following task to configure DMVPN Multiple Tunnel Termination

```

router bgp 1
  bgp log-neighbor-changes
  bgp listen range 192.168.0.0/16
  peer-group SPOKES2
  bgp listen range 190.168.0.0/16
  peer-group SPOKES network 192.168.0.0
  aggregate-address 192.168.0.0 255.255.0.0 summary-only
  timers bgp 10 30
  neighbor SPOKES2 peer-group
  neighbor SPOKES2 remote-as 1
  neighbor SPOKES2 next-hop-self
  maximum-secondary-paths eigrp 1

```

Configuration Examples for DMVPN Support for IWAN

Example: DMVPN Support for IWAN

The following is an example for configuring DMVPN on hub.

```

router eigrp DMVPN
  !
  address-family ipv4 unicast autonomous-system 100
  !
  af-interface Tunnel0 \
  summary-address 192.168.0.0 255.255.0.0
  no split-horizon
  exit-af-interface
  !
  af-interface Tunnel1
  summary-address 192.168.0.0 255.255.0.0
  no split-horizon
  exit-af-interface

```

```

!
topology base
maximum-secondary-paths 4
fast-reroute per-prefix all
fast-reroute tie-break interface-disjoint 1
exit-af-topology
network 10.0.0.0
network 20.0.0.0
network 192.168.149.0
exit-address-family
!

```

The following is an example for configuring DMVPN on spoke 1.

```

router bgp 1
bgp log-neighbor-changes
bgp listen range 20.0.0.0/8
peer-group SPOKES2
bgp listen range 10.0.0.0/8
peer-group SPOKES
network 192.168.149.0
aggregate-address 192.168.0.0 255.255.0.0 summary-only
timers bgp 10 30
neighbor SPOKES peer-group
neighbor SPOKES remote-as 1
neighbor SPOKES next-hop-self
neighbor SPOKES2 peer-group
neighbor SPOKES2 remote-as 1
neighbor SPOKES2 next-hop-self
maximum-secondary-paths eigrp 1
!

```

The following is an example for configuring DMVPN on spoke 2.

```

router bgp 1
bgp log-neighbor-changes
bgp listen range 20.0.0.0/8
peer-group SPOKES2
bgp listen range 10.0.0.0/8
peer-group SPOKES
bgp additional-paths install
network 192.168.149.0
aggregate-address 192.168.0.0 255.255.0.0 summary-only
timers bgp 10 30
neighbor SPOKES peer-group
neighbor SPOKES remote-as 1
neighbor SPOKES next-hop-self
neighbor SPOKES2 peer-group
neighbor SPOKES2 remote-as 1
neighbor SPOKES2 next-hop-self
maximum-secondary-paths eigrp 1
!

```

The following is the sample output for the **show ip bgp** command.

```

Device# show ip bgp

BGP table version is 10, local router ID is 192.168.149.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
   Network                Next Hop                Metric LocPrf Weight Path

```

```
*> 192.168.0.0/16 0.0.0.0 32768 I
s i t 192.168.40.0 20.0.0.41 0 100 0 I
s>i 10.0.0.41 0 100 0 I
s>i 192.168.50.0 10.0.0.51 0 100 0 I
s i t 20.0.0.51 0 100 0 I
s> 192.168.149.0 0.0.0.0 0 32768 I
```

Device#

The following is the sample output for the **show ip bgp** command in two different interfaces.

Device# **show ip bgp 192.168.40.0**

```
BGP routing table entry for 192.168.40.0/24, version 6
Paths: (2 available, best #2, table default, Advertisements suppressed by an aggregate.)
  Not advertised to any peer
  Refresh Epoch 1
  Local 20.0.0.41 from *20.0.0.41 (192.168.40.2)
    Origin IGP, metric 0, localpref 100, valid, internal, secondary path
    rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  Local 10.0.0.41 from *10.0.0.41 (192.168.40.1)
    Origin IGP, metric 0, localpref 100, valid, internal, best
    rx pathid: 0, tx pathid: 0x0
```

Device# **show ip bgp 192.168.50.0**

```
BGP routing table entry for 192.168.50.0/24, version 10
Paths: (2 available, best #1, table default, Advertisements suppressed by an aggregate.)
  Not advertised to any peer
  Refresh Epoch 1
  Local 10.0.0.51 from *10.0.0.51 (192.168.50.1)
    Origin IGP, metric 0, localpref 100, valid, internal, best
    rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  Local 20.0.0.51 from *20.0.0.51 (192.168.50.2)
    Origin IGP, metric 0, localpref 100, valid, internal, secondary path
    rx pathid: 0, tx pathid:
```

The following is the sample output for the **show ip route** command.

Device# **show ip route**

```
Codes: L - local, 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, H - NHRP, l - LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, Tunnel0
L       10.0.0.149/32 is directly connected, Tunnel0
 20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Tunnel1
L       20.0.0.149/32 is directly connected, Tunnel1
B       192.168.0.0/16 [200/0], 00:02:26, Null0
B       192.168.40.0/24 [200/0] via 10.0.0.41, 00:02:26
B       192.168.50.0/24 [200/0] via 10.0.0.51, 00:01:55
 192.168.149.0/24 is variably subnetted, 2 subnets, 2 masks
```

Example: DMVPN Support for IWAN

```
C      192.168.149.0/24 is directly connected, Ethernet1/0
L      192.168.149.1/32 is directly connected, Ethernet1/0
```

The following is the sample output for the **show ip route** command for the secondary path.

```
Device# show ip route
```

```
sec Codes: L - local, 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, H - NHRP, l - LISP
           a - application route
           + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      10.0.0.0/8 is directly connected, Tunnel0
L      10.0.0.149/32 is directly connected, Tunnel0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      20.0.0.0/8 is directly connected, Tunnel1
L      20.0.0.149/32 is directly connected, Tunnel1
B      192.168.0.0/16 [200/0], 00:02:26, Null0
B      192.168.40.0/24 [200/0] via 10.0.0.41, 00:02:26
           [SEC][200/0] via 20.0.0.41, 00:02:26
B      192.168.50.0/24 [200/0] via 10.0.0.51, 00:01:55
           [SEC][200/0] via 20.0.0.51, 00:01:55
192.168.149.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.149.0/24 is directly connected, Ethernet1/0
L      192.168.149.1/32 is directly connected, Ethernet1/0
```

The following is the sample output for the **show ip cef** command.

```
Device# show ip cef 192.168.40.0 detail
```

```
192.168.40.0/24, epoch 0, flags [rib only nolabel, rib defined all labels]
recursive via 10.0.0.41
attached to Tunnel0
```

```
Device# show ip cef 192.168.40.0 int
```

```
192.168.40.0/24, epoch 0, flags [rnlbl, rlbls], RIB[B], refcnt 5, per-destination sharing
sources: RIB
feature space:
  IPRM: 0x00018000
ifnums:
  Tunnel0(19): 10.0.0.41
  path list F3BDA6DC, 3 locks, per-destination, flags 0x269 [shble, rif, rcrsv, hwc, bgp]
    path F3BDABAC, share 1/1, type recursive, for IPv4
      recursive via 10.0.0.41[IPv4:Default], fib F693B80C, 1 terminal fib,
v4:Default:10.0.0.41/32
    path list F3BDA72C, 2 locks, per-destination, flags 0x49 [shble, rif, hwc]
      path F3BDAC14, share 1/1, type adjacency prefix, for IPv4
        attached to Tunnel0, IP midchain out of Tunnel0, addr 10.0.0.41 F555A5E0
output chain:  IP midchain out of Tunnel0, addr 10.0.0.41 F555A5E0
               IP adj out of Ethernet0/0, addr 11.0.0.41 F3CCCC10
```

The following is the sample output for the **show ip route** command for the repair paths.

```
Device# show ip route repair-paths
```

```
Codes: L - local, 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, H - NHRP, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.0.0.0/8 is directly connected, Tunnel0
L    10.0.0.149/32 is directly connected, Tunnel0
  20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    20.0.0.0/8 is directly connected, Tunnel1
L    20.0.0.149/32 is directly connected, Tunnel1
B    192.168.0.0/16 [200/0], 00:02:26, Null0
B    192.168.40.0/24 [200/0] via 10.0.0.41, 00:00:10
      [RPR][200/0] via 20.0.0.41, 00:00:10
B    192.168.50.0/24 [200/0] via 10.0.0.51, 00:00:10
      [RPR][200/0] via 20.0.0.51, 00:00:10
  192.168.149.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.149.0/24 is directly connected, Ethernet1/0
L    192.168.149.1/32 is directly connected, Ethernet1/0

```

The following is the sample output for the **show ip route** command.

```

Device# show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.0.0.0/8 is directly connected, Tunnel0
L    10.0.0.149/32 is directly connected, Tunnel0
  20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    20.0.0.0/8 is directly connected, Tunnel1
L    20.0.0.149/32 is directly connected, Tunnel1
D    192.168.0.0/16 is a summary, 00:08:53, Null0
D    192.168.40.0/24 [90/30378666] via 20.0.0.41, 00:08:45, Tunnel1
D    192.168.50.0/24 [90/30378666] via 20.0.0.51, 00:08:34, Tunnel1
  192.168.149.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.149.0/24 is directly connected, Ethernet1/0
L    192.168.149.1/32 is directly connected, Ethernet1/0

```

The following is the sample output for the **show ip route** command for the secondary path.

```

Device# show ip route sec
Codes: L - local, 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, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C    10.0.0.0/8 is directly connected, Tunnel0
L    10.0.0.149/32 is directly connected, Tunnel0
  20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

```

```

C      20.0.0.0/8 is directly connected, Tunnel1
L      20.0.0.149/32 is directly connected, Tunnel1
D      192.168.0.0/16 is a summary, 00:08:53, Null0
D      192.168.40.0/24 [90/30378666] via 20.0.0.41, 00:08:45, Tunnel1
          [SEC][90/31232000] via 10.0.0.41, 00:08:45, Tunnel0
D      192.168.50.0/24 [90/30378666] via 20.0.0.51, 00:08:34, Tunnel1
          [SEC][90/31232000] via 10.0.0.51, 00:08:34, Tunnel0
          192.168.149.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.149.0/24 is directly connected, Ethernet1/0
L      192.168.149.1/32 is directly connected, Ethernet1/0

```

Troubleshooting

```

NHRP-INT: Multipath nexthop lookup requested(if_in:, netid:1) for 192.168.50.1 in vrf
global(0x0)
NHRP-INT: Multipath recursive lookup for 192.168.50.1 (192.168.50.0/24)
NHRP-INT: Path(1/1): [0x1]192.168.50.0/24 via 20.0.0.51, Tunnel1
NHRP-INT: Current first level nexthop: 20.0.0.51
NHRP-INT: Path(1) for 192.168.50.1 in vrf global(0x0) recursively resolved to 20.0.0.51,
Tunnel1, path metric: D/O/, ll_nhop 20.0.0.51
NHRP-INT: Found a better path(old: X/X//0, new: D/O//1); updating nhop: 20.0.0.51, Tunnel1
NHRP-INT: Updated best path to 20.0.0.51, Tunnel1(D/O/)
NHRP-INT: Path(2/1): [0x100]192.168.50.0/24 via 10.0.0.51, Tunnel0
NHRP-INT: Current first level nexthop: 10.0.0.51
NHRP-INT: Path(2) for 192.168.50.1 in vrf global(0x0) recursively resolved to 10.0.0.51,
Tunnel0, path metric: S/C/, ll_nhop 10.0.0.51
NHRP-INT: Found a better path(old: D/O//1, new: S/C//1); updating nhop: 10.0.0.51, Tunnel0
NHRP-INT: Updated best path to 10.0.0.51, Tunnel0(S/C/)
NHRP-INT: Multipath recursive path walk(if_in:, netid:1) for 192.168.50.1(pfx:192.168.50.0/24)
in global(0x0) yielded 10.0.0.51, Tunnel0
NHRP-DETAIL: Multipath recursive nexthop lookup(if_in:, netid:1) for 192.168.50.1 in
global(0x0) yielded 10.0.0.51, Tunnel0
..

```

Additional References for DMVPN Support for IWAN

Related Documents

Related Topic	Document Title
Cisco Intelligent WAN - An SD-WAN Solution	Cisco Intelligent WAN - An SD-WAN Solution

MIBs

MIB	MIBs Link
• CRCOMB	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

Feature Information for DMVPN Support for IWAN

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 25: Feature Information for DMVPN Support for IWAN

Feature Name	Releases	Feature Information
DMVPN Multiple Tunnel Termination	Cisco IOS XE Denali 16.3.2 Cisco IOS XE Everest 16.4.1	DMVPN supports Cisco Intelligent WAN architecture to provide transport independence through overlay routing. The DMVPN Multiple Tunnel Termination feature enables support for secondary paths for the supported routing protocols in the Routing Information Base (RIB). The following command was introduced by this feature: maximum-secondary-paths.



CHAPTER 16

Configuring MPLS over DMVPN

The MPLS over DMVPN feature implements Multiprotocol Label Switching (MPLS) over a dynamically established IPsec tunnel, thereby enabling communication between overlapping addresses in customer sites.

- [Finding Feature Information, on page 205](#)
- [Prerequisites for Configuring MPLS over DMVPN, on page 205](#)
- [Information About MPLS over DMVPN, on page 206](#)
- [IVRF Support, on page 212](#)
- [How to Configure MPLS over DMVPN, on page 212](#)
- [Restrictions for Configuring 6VPE and 6PE Support in MPLS over DMVPN Phase 2, on page 225](#)
- [Configuring 6VPE Support in MPLS over DMVPN Phase 2, on page 225](#)
- [Configuring 6PE Support in MPLS over DMVPN Phase 2, on page 230](#)
- [Verifying the 6VPE support in MPLS over DMVPN Phase 2 Configurations, on page 233](#)
- [Verifying the 6PE support in MPLS over DMVPN Phase 2 Configurations, on page 233](#)
- [Configure a Spoke Node as a P Node in MPLS over DMVPN Phase 3, on page 234](#)
- [Feature Information for MPLS over DMVPN, on page 234](#)

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 Configuring MPLS over DMVPN

- MP-BGP must be configured as MP-BGP allows labels to be distributed for every prefix or per VRF; label assignment per VRF would make it easy to maintain.
- NHRP Redirect feature must be installed as an MPLS output feature. To send the NHRP redirect, NHRP must know the VRF to which the redirect must be sent to.

Information About MPLS over DMVPN

MPLS over DMVPN Networks

Traffic in network domains having overlapping addressing spaces are segregated via VRFs. This is to ensure that traffic intended for one customer does not enter into another customer's domain. To protect data between provider-edge (PE) devices using IPsec, a tunnel interface with IPsec protection can be defined for each VRF, which ensures that traffic from every customer domain passes over the corresponding IPsec tunnel. However, as the number of customer sites and nodes grow in the network, this is not scalable since there is a need for separate IPsec tunnel and an interface for each customer site that must be protected.

MPLS provides the ability to assign labels per-VRF or per-prefix, thereby identifying the correct VRF into which traffic needs to be routed to. This is achieved with an MPLS-aware interface having IPsec protection and an IPsec tunnel built between the PE devices. The basic methodologies in MPLS are as follows:

MPLS forwarding—This is used in the transport networks where a label is pushed at the ingress PE device for a particular prefix and the labels are swapped as the data moves towards the egress PE device. At the egress PE device or a device before the egress PE (penultimate hop pop), the label is popped and data is forwarded based on the Layer 3 protocol. LDP is typically the label distribution protocol run in the transport space along with unicast routing protocol.

MPLS VPNs—This is used to carry data across a transport network between customer sites on VRFs. The overlay prefixes are identified by a VPN overlay label and is used as an inner label in a MPLS data packet. The outer label is the MPLS transport label and is for switching the packet in the core. LDP is used along with a IGP to achieve MPLS unicast IP forwarding in the core network and MP-BGP provides a mechanism to identify the customer VRF network to which a packet is forwarded when a packet arrives at Egress Label Edge Router (E-LER). Each of the protocols – LDP and MP-BGP protocols distribute labels to help in achieving this.

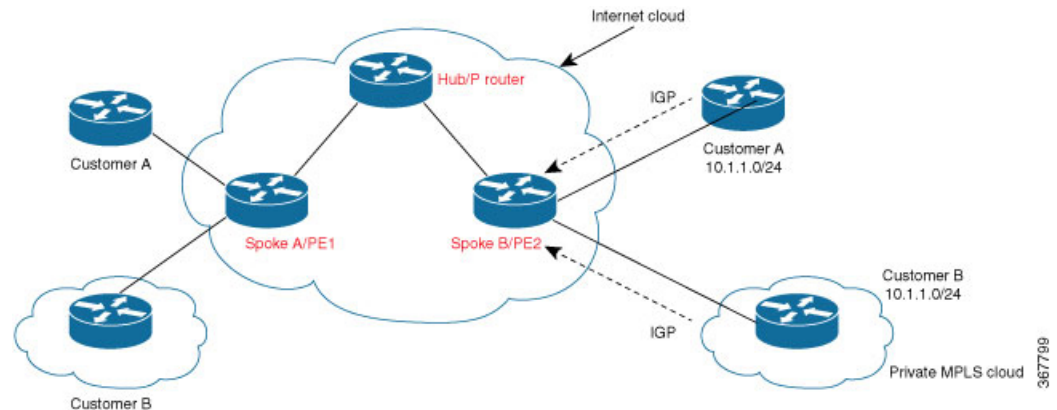
The goal of the MPLS over Dynamic IPsec Tunnels feature is to provide a solution that helps communication between overlapping addresses in customer sites when a remote customer site needs to be discovered dynamically using NHRP and at the same time secure the data traffic between the PE routers using IPsec. This solution can be used to deploy an MPLS network and to extend their MPLS network on a new network (determined dynamically), in a different region, securely over the Internet.

Until this feature was introduced, MPLS support on DMVPN existed in a DMVPN hub and spoke network only. The feature extends support in DMVPN spoke to spoke networks where data packets are tag-switched on the hub and cannot trigger a NHRP redirect thereby addressing a scalable solution using multipoint GRE interface on DMVPN networks and point-to-point interface on FlexVPN networks.

The Need for MPLS

The basic goal of a Layer 3 VPN network is to allow sites in a customer network to communicate with each other. The following diagram explains the need for MPLS with the help of an example.

Figure 12: Overlapping addresses in Customer Edge (CE) Domain



Per the above diagram, Customer A network behind spoke A/PE1 router needs to communicate with the customer A subnet 10.1.1.0/24. However, because of overlapping address space with customer B, spoke B/PE2 router would learn about two different 10.1.1.0/24 prefixes and if it picks the route to customer B as best route, packets would never reach the customer A network behind spoke B.

MPLS solves this problem by associating labels for each customer prefix present in different VRF tables. These labels are distributed between PEs, used during packet forwarding to determine the correct customer network to which a packet should be forwarded to. MPLS deals with overlapping prefixes by prepending another number to the BGP NLRI (prefix). MP-BGP has the provision of adding a variable-length number called address family in front of the prefix. MPLS VPNs use the address family to carry route distinguishers (RDs). The combined VPNv4 address (64-bit RD + 32-bit prefix) makes the address unique. The steps involved are:

- Provider and provider-edge devices run LDP and IGP to support unicast IP routing. IGP only advertises routes for subnets inside the MPLS network but does not include any customer routes.
- PEs learn customer specific routes using IGP and store the routes in per-customer VRF routing tables.
- PEs use MP-BGP to exchange customer routes with other PEs.

Components of MPLS over Dynamic IPsec Tunnels Feature

The essential components of this solution comprise:

IKEv2 and IPsec—Internet Key Exchange version 2 (IKEv2) and IPsec secure traffic between spoke and the hub and later between the spokes when the remote spoke is discovered dynamically. IKEv2 is used to add static routes to the peer's tunnel overlay address as a directly connected route in FlexVPN. This results in an implicit-NULL label to be added to the LIB for the peer's tunnel overlay address. (IPRM (IP Resource manager) adds the implicit-NULL label and is the common component that is used for implicit-NULL label addition by applications such as LDP and now IKEv2). IKEv2 is used instead of LDP for the following reasons:

If LDP is used for distributing transports labels, it involves establishing TCP channel with every LDP neighbor making it heavy-weight in a scaled scenario.

LDP keepalive will try to keep the spoke-to-spoke tunnel active, even in the absence of traffic, and never bring the spoke-to-spoke tunnel down.

NHRP—Next Hop Resolution Protocol (NHRP) resolves the remote overlay address and dynamically discovers the transport end point needed to establish a secure tunnel. If a multipoint GRE interface is used, the tunnel

end point database stores the mapping between the overlay and corresponding nonbroadcast multiaccess (NBMA) address. NHRP control packets that are not specific to a VRF are forwarded to global addresses. Control packets specific to a virtual domain context (for example, resolution request destined for a customer network or host address) are forwarded to a specific VRF.

MPLS—Multiprotocol label switching (MPLS) enables MPLS tag switching for data packets. Label Distribution Protocol (LDP) is not enabled between spokes.

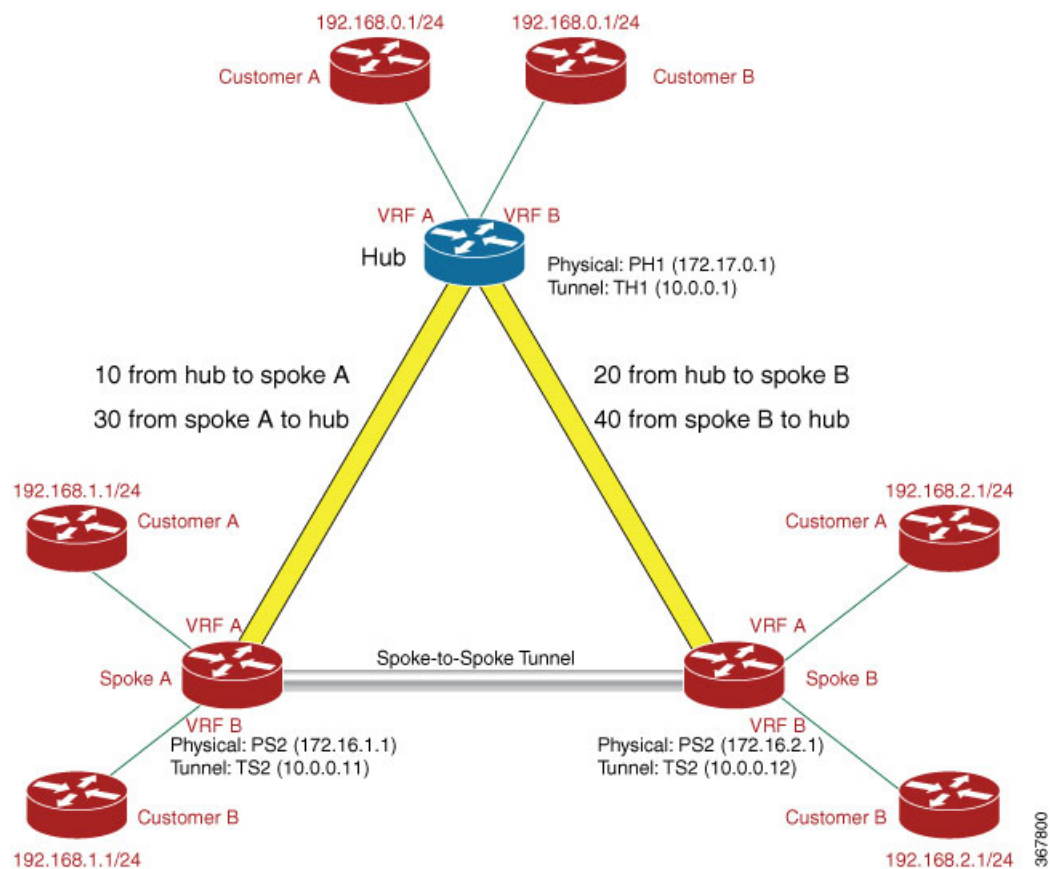
MFI—Multicast Forwarding Information (MFI) allocates and releases labels assigned to tunnels.

MP-BGP—Multiprotocol BGP (MP-BGP) distributes overlay labels for the customer network on different VRFs.

Working of MPLS over Dynamic IPsec Tunnels Feature

This section describes the working of the MPLS over Dynamic IPsec Tunnels feature with the help of the following topology as an example, where traffic flows from IP address 192.168.1.1 of Customer A, behind Spoke A to IP address 192.168.2.1 of Customer A, behind Spoke B.

Figure 13: DMVPN Spoke-Hub-Spoke Topology



1. IKEv2 and IPsec security associations (SA) are established from the spoke to the hub. IKEv2 installs the implicit-NULL label values for the peer's overlay address received in the mode config reply and mode config set. Implicit-NULL label is installed because the spoke and hub are always next hop to each other

in the overlay space. To enable MPLS tag switching, use the `mpls nhrp` command on the tunnel interface or virtual template interface.



Note Using the `mpls ip` command performs the same function as `mpls nhrp` command but enables LDP also, which is not recommended.

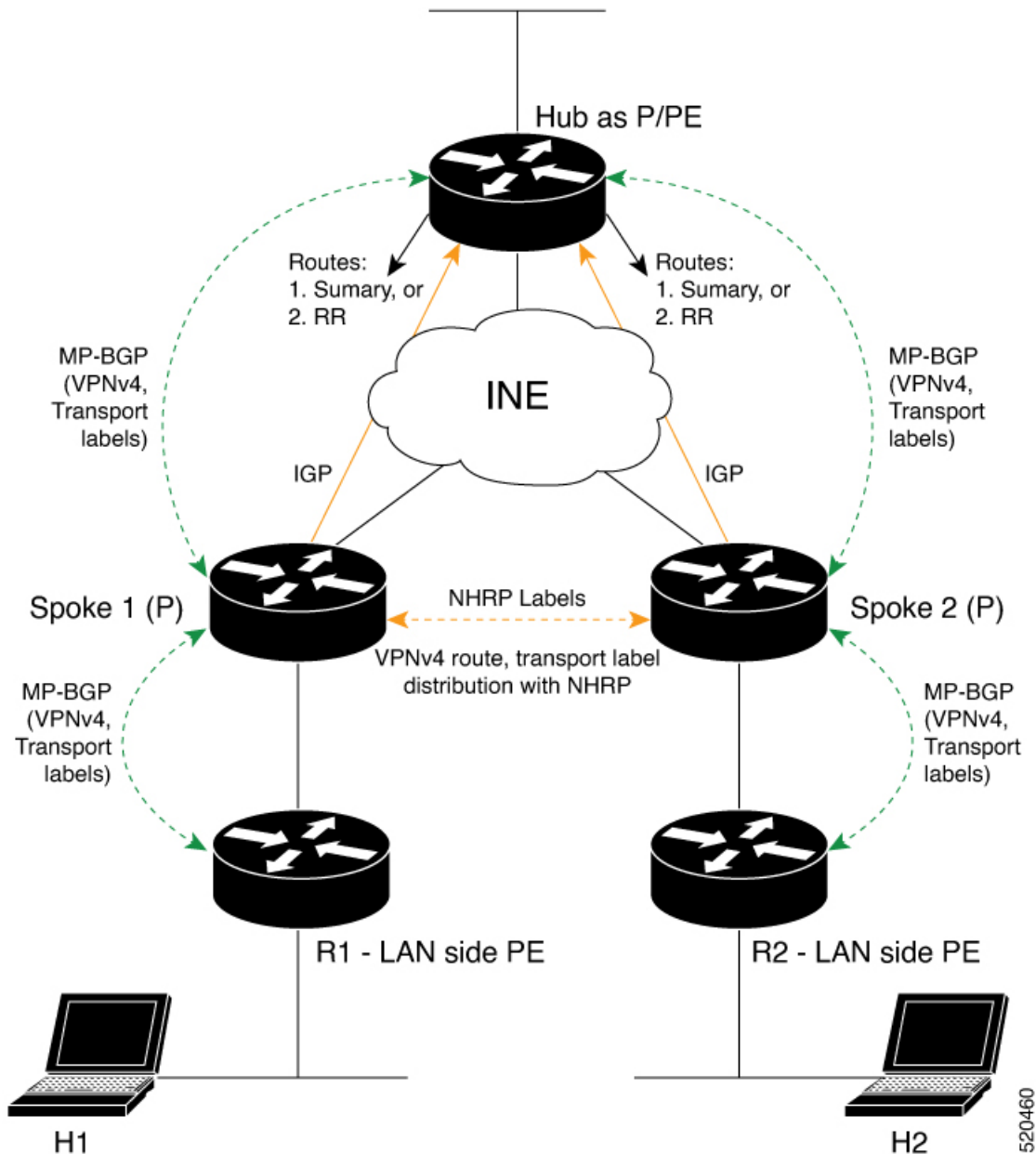
2. After establishing an IPsec session between a spoke and a hub and the implicit-NULL label is installed, MP-BGP exchanges label per-VRF or label per-prefix for all VRFs.
3. Data is forwarded when label and route exchange is complete. When the first packet destined for 192.168.2.1 arrives on spoke A on VRF A, the packet is forwarded to the hub. The packet is label encapsulated (with just the overlay label), GRE encapsulated and encrypted.
4. When the packet reaches the virtual access interface or GRE interface on the hub, the packet is decrypted and GRE decapsulated. The label identifies the VRF on which the packet arrives and the VRF information corresponding to the label is conveyed to NHRP. NHRP constructs the redirect packet and dispatches the packet in the MPLS switching path and sends the packet to MPLS LSP. The packet is label encapsulated, GRE encapsulated, encrypted, and sent to the host behind Spoke A.
5. The redirect packet (NHRP) arrives at spoke A, is decrypted, and is GRE decapsulated. The redirect packet is processed and a NHRP resolution request is triggered. The request is sent to a specific VRF in a host network behind Spoke B. This is because the host network behind Spoke B needs to be resolved and it is also possible that the network can have overlapping address with another network. MPLS provides the VRF information, which corresponds to the outer VRF label. This resolution packet is label encapsulated, GRE encapsulated, encrypted and sent to the hub. An NHRP mapping entry is created and VRF A is also associated for the prefix that needs to be resolved.
6. NHRP resolution request arrives at the hub and is decrypted and GRE decapsulated. NHRP looks up the route in the VRF table and identifies the outgoing interface. The resolution request is label encapsulated, GRE encapsulated, encrypted and sent to Spoke 2.
7. Spoke B decrypts the resolution request packet gets decrypted on the spoke B and learns the VRF label. A virtual access is created on Spoke B for point-to-point solution and an IKEv2 or IPsec session is initiated from Spoke B to Spoke A. This result in the creation of virtual access on Spoke A also by IKEv2 in a point-to-point solution. NHRP adds the route for Spoke A tunnel IP address via the new virtual access interface.
8. NHRP resolution reply is received at virtual access interface on Spoke A. NHRP request ID in the reply packet is matched with the request ID of the request, which is sent by Spoke A to know the VRF for which the request was sent. NHRP looks up to find the NHRP entry and the entry is said to be "Complete." NHRP also inserts a route into the VRF routing table with the label information. With the routes and labels setup between Spoke A and Spoke B, traffic is VPN label encapsulated and encrypted over the spoke-spoke dynamically established tunnel between Spoke A to Spoke B.

Support for Spoke Nodes as P Nodes in MPLS over DMVPN Phase 3

In IOS XE Amsterdam 17.1.x and earlier releases, in an MPLS over DMVPN Phase 3 deployment you could configure a spoke node only as a PE node. From IOS XE Amsterdam 17.2.1, you can configure spoke node as either a P or PE node.

Overview of the Support for Spoke Nodes as P Nodes in MPLS over DMVPN Phase 3

Consider the configuration in the following figure, in which Spoke 1 and Spoke 2 are P nodes:



H1 and H2 are hosts connected to the PE nodes R1 and R2, respectively, and part of a customer network. There may be other customer networks connected to R1 and R2. So, VRFs are configured on both R1 and R2 to segregate the traffic of one customer network from another.

Spoke 1 and Spoke 2 are P nodes in the MPLS DMVPN cloud. Spoke 1 learns of the VPNv4 prefixes and overlay labels for each VRF defined on R1 via MP-BGP. The VPNv4 prefixes and labels are also imported, on demand, to and from NHRP as part of NHRP resolution. Similarly, Spoke 2 learns of the VPNv4 prefixes and labels for the VRFs defined on R2.

Spoke 1 and Spoke 2 register with the Hub and exchange routing information (VPNv4 prefixes and labels) through MP-BGP via the Hub.

Suppose that the host H1, connected to the PE node R1, attempts to send some traffic to the host H2, connected to the PE node R2. R1 forwards the packets to Spoke 1 after tagging the packets with the appropriate VRF and outer labels.

When the first packet arrives at Spoke 1, Spoke 1 forwards the packet to the Hub after swapping the outer labels appropriately. The Hub examines the VRF label and sends an NHRP traffic indication to Spoke 1 to indicate the availability of a more optimal route.

On receiving the NHRP traffic indication, Spoke 1 sends an NHRP resolution request for the VPNv4 destination address to Spoke 2 via the Hub.

Spoke 2 sends an NHRP resolution reply to Spoke 1 after an on-demand dynamic tunnel is established between the two spokes. NHRP inserts route and label into the BGP (VPNv4), RIB (Global/Default), and LFIB. MP-BGP imports the route/label and redistributes the information to R1.

Packets from H1 to H2 are tagged with the appropriate VRF label and transport label to be sent over the spoke-to-spoke tunnel.

Restrictions for Support for Spoke Nodes as P Nodes in MPLS over DMVPN Phase 3

- This feature requires BGP VPNv4 peering between the P nodes deployed as spoke nodes and the PE nodes on the LAN side.
- This feature is supported only with IPv4 addressing.
- All nodes must use the same RD for DMVPN Phase 3.

Enhancements to BGP and NHRP

To support a configuration with spoke nodes as P nodes, BGP and NHRP are enhanced to redistribute routes and labels between the P nodes acting as spoke nodes and the PE nodes on the LAN side.

Enhancements to BGP

- BGP receives a notification from NHRP when NHRP learns of a new VPNv4 prefix. In response, BGP imports the prefix information from NHRP and propagates the information to the LAN-side PE node.
- BGP provides a mechanism for NHRP to import information from BGP.
- BGP notifies NHRP when the information redistributed to NHRP must be updated.
- BGP propagates prefix updates or deletes prefix information based on NHRP notifications.

Enhancements to NHRP

- MPLS-labelled packets are inspected and packets that carry a GAL label and an NHRP channel number are punted to control plane for further processing.
- NHRP imports information from BGP. Information imported by NHRP includes route(prefix/mask), RD, RT and other opaque data (needed by BGP to reconstruct the VPNv4 prefix at the other end).
- NHRP adds, updates, and deletes prefix information to the RIB and LFIB on the P nodes acting as spoke nodes.

- NHRP notifies BGP to add, update, or withdraw prefix information.
- In response to resolution request, NHRP provides the forwarding information in an enhanced CIE.
- Between peers, NHRP purges information on a per-prefix basis. Earlier, the purge was based on the peer address and request ID. Now, prefix is also taken into account.

Enhanced NHRP CIE

The NHRP CIE is enhanced to include a TLV block in which NHRP packs the forwarding information. The **E** flag is set to indicate the presence of this new TLV block. The CIE and TLV block formats are as follows:

```
CIE Format:
+-----+-----+-----+-----+
| Code      | Prefix Length | |E|      | unused  |
+-----+-----+-----+-----+
| Maximum Transmission Unit | Holding Time |
+-----+-----+-----+-----+
| Cli Addr T/L | Cli SAddr T/L | Cli Proto Len | Preference |
+-----+-----+-----+-----+
| Client NBMA Address (variable length) |
+-----+-----+-----+-----+
| Client NBMA Subaddress (variable length) |
+-----+-----+-----+-----+
| Client Protocol Address (variable length) |
+-----+-----+-----+-----+
```

To manage a scenario in which one peer node is updated to IOS XE 17.2.1 or a later release, but the other peer node is not, nodes use capability negotiation to check the capability of a peer before sending the enhanced CIE.

NHRP Enhanced CIE TLV structure:

```
+-----+-----+-----+-----+
|C| Reserved | Type      | Reserved | Length  |
+-----+-----+-----+-----+
| Value ... |
+-----+-----+-----+-----+
```

C - Compulsory (CIE can't be interpreted if this extension is not understood)

IVRF Support

If a tunnel interface belongs to an IVRF, routing related operations, such as, route lookup, route addition and deletion, that happen in NHRP are performed in the routing table of IVRF configured on tunnel interface.

How to Configure MPLS over DMVPN

Configuring MPLS over FlexVPN

Perform this task to configure MPLS over DMVPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel *number***
4. Do one of the following: **mpls nhrpor mpls bgp forwarding**
5. **end**
6. **show mpls forwarding-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Device(config)# interface tunnel 1	Configures the FlexVPN client interface and enters interface configuration mode.
Step 4	Do one of the following: mpls nhrpor mpls bgp forwarding Example: Device(config-if)# mpls nhrp Device(config-if)# mpls bgp forwarding	
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode and returns to global configuration mode.
Step 6	show mpls forwarding-table Example: Device# show mpls forwarding-table	Displays information about the Multiprotocol Label Switching (MPLS) Label Forwarding Information Base (LFIB).

Configuration Examples for MPLS over FlexVPN**Example: MPLS over DMVPN—Using LDP and BGP**

This section lists a sample configuration on spokes and the hub using LDP and BGP. The following is the configuration on Spoke A:

```
ip vrf custA
rd 10:100
route-target export 10:1000
```

```

route-target import 10:1000
!
ip vrf custB
rd 10:110
route-target export 10:2000
route-target import 10:2000
mpls label mode all-vrfs protocol bgp-ipv4 per-vrf
!
crypto ikev2 authorization policy default
route set interface
!
!
!
crypto ikev2 keyring KR
peer All
address 0.0.0.0 0.0.0.0
pre-shared-key Cisco123
!
!
crypto ikev2 profile default
match identity remote fqdn domain cisco.com
identity local fqdn R2.cisco.com
authentication local pre-share
authentication remote pre-share
keyring local KR
aaa authorization group psk list default default
virtual-template 2
!
crypto ipsec profile default
set ikev2-profile default
interface Loopback0
ip address 10.0.0.101 255.255.255.255
!
interface Tunnel0
ip address 10.0.0.11 255.255.255.255
mpls bgp forwarding
ip nhrp network-id 1
ip nhrp shortcut virtual-template 2
tunnel source Ethernet0/1
tunnel destination 172.17.0.1
tunnel protection ipsec profile default
!
interface Ethernet0/0
ip vrf forwarding custA
ip address 192.168.1.1 255.255.255.0
!
interface Ethernet0/1
ip address 172.16.1.1 255.255.255.0
!
interface Ethernet0/2
ip vrf forwarding custB
ip address 192.168.1.1 255.255.255.0
interface Ethernet1/0
ip vrf forwarding custA
ip address 192.168.50.254 255.255.255.0
router ospf 10
network 172.16.1.0 0.0.0.255 area 0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.103 remote-as 100
neighbor 10.0.0.103 update-source Loopback0
neighbor 10.0.0.103 soft-reconfiguration inbound
!

```

```

address-family vpnv4
neighbor 10.0.0.103 activate
neighbor 10.0.0.103 send-community both
exit-address-family
!
address-family ipv4 vrf custA
network 192.168.1.0
network 192.168.50.0
exit-address-family
!
address-family ipv4 vrf custB
network 192.168.1.0
exit-address-family

```

The following is the configuration on Spoke B:

```

ip vrf custA
rd 10:100
route-target export 10:100
route-target export 10:1000
route-target import 10:100
route-target import 10:1000
!
ip vrf custB
rd 10:110
route-target export 10:2000
route-target import 10:2000
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
!
crypto ikev2 authorization policy default
route set interface
!
!
crypto ikev2 keyring KR
peer All
address 0.0.0.0 0.0.0.0
pre-shared-key Cisco123
!
!
crypto ikev2 profile default
match identity remote fqdn domain cisco.com
identity local fqdn R3.cisco.com
authentication local pre-share
authentication remote pre-share
keyring local KR
aaa authorization group psk list default default
virtual-template 2
!
crypto ipsec profile default
set ikev2-profile default
!
interface Loopback0
ip address 10.0.0.104 255.255.255.255
interface Tunnel0
ip address 10.0.0.12 255.255.255.255
mpls bgp forwarding
ip nhrp network-id 1
ip nhrp shortcut virtual-template 2
tunnel source Ethernet0/0
tunnel destination 172.17.0.1
tunnel protection ipsec profile default
!
interface Ethernet0/0
ip address 172.16.2.1 255.255.255.0

```

```

!
interface Ethernet0/1
ip vrf forwarding custA
ip address 192.168.2.1 255.255.255.0
!
interface Ethernet0/2
ip vrf forwarding custB
ip address 192.168.2.1 255.255.255.0
router ospf 10
network 172.16.2.0 0.0.0.255 area 0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.101 remote-as 100
neighbor 10.0.0.101 update-source Loopback0
neighbor 10.0.0.101 soft-reconfiguration inbound
neighbor 10.0.0.103 remote-as 100
neighbor 10.0.0.103 update-source Loopback0
neighbor 10.0.0.103 soft-reconfiguration inbound
!
address-family vpnv4
neighbor 10.0.0.101 activate
neighbor 10.0.0.101 send-community both
neighbor 10.0.0.103 activate
neighbor 10.0.0.103 send-community both
exit-address-family
!
address-family ipv4 vrf custA
network 192.168.2.0
network 192.168.70.0
exit-address-family
!
address-family ipv4 vrf custB
network 192.168.2.0
exit-address-family
!

```

The following is the hub configuration.

```

ip vrf custA
rd 10:100
route-target export 10:1000
route-target import 10:1000
!
mpls label mode all-vrfs protocol bgp-vpnv4 per-vrf
!
crypto ikev2 authorization policy default
pool FPool
route set interface
!
crypto ikev2 keyring KR
peer All
address 0.0.0.0 0.0.0.0
pre-shared-key Cisco123
!
!
!
crypto ikev2 profile default
match identity remote fqdn domain cisco.com
identity local fqdn R1.cisco.com
authentication local pre-share
authentication remote pre-share
keyring local KR
aaa authorization group psk list default default
virtual-template 1

```

```

!
!
crypto ipsec profile default
set ikev2-profile default
!
interface Loopback0
ip address 10.0.0.103 255.255.255.255
!
interface Loopback1
ip address 10.0.0.1 255.255.255.0
!
!
interface Ethernet0/0
ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1/0
ip vrf forwarding custA
ip address 192.168.70.254 255.255.255.0
!
interface Virtual-Templatel type tunnel
ip unnumbered Loopback1
mpls bgp forwarding
ip nhrp network-id 1
ip nhrp redirect
tunnel protection ipsec profile default
!
ip local pool FPool 10.1.0.1 10.1.0.100
!
router ospf 10
network 172.17.0.0 0.0.0.255 area 0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.101 remote-as 100
neighbor 10.0.0.101 update-source Loopback0
neighbor 10.0.0.101 soft-reconfiguration inbound
neighbor 10.0.0.104 remote-as 100
neighbor 10.0.0.104 update-source Loopback0
neighbor 10.0.0.104 soft-reconfiguration inbound
auto-summary
!
address-family vpnv4
neighbor 10.0.0.101 activate
neighbor 10.0.0.101 send-community both
neighbor 10.0.0.101 next-hop-self
neighbor 10.0.0.104 activate
neighbor 10.0.0.104 send-community both
neighbor 10.0.0.104 next-hop-self
exit-address-family
!
address-family ipv4 vrf custA
redistribute static route-map rm
exit-address-family
!
ip route vrf custA 0.0.0.0 0.0.0.0 Null0 tag 10
ip route vrf custA 192.168.0.0 255.255.0.0 Null0 tag 10
!
ip access-list extended out1
permit ip any any
!
!
route-map rm permit 10
match tag 10

```

Example: MPLS over DMVPN - Using MPLS

The following is the configuration on Spoke 1:

```

hostname R3-Spoke
!
boot-start-marker
boot-end-marker
!
!
vrf definition cust1
rd 1:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family
!
vrf definition cust2
rd 2:2
route-target export 2:2
route-target import 2:2
!
address-family ipv4
exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
mpls ldp loop-detection
!
crypto pki trustpoint CA
enrollment url http://172.16.1.1:80
password
fingerprint E0AFEF7F08070BAB33C8297C97E6457
subject-name cn=R3-spoke.cisco.com,OU=FLEX,O=Cisco
revocation-check crl none
!
crypto pki certificate map mymap 10
subject-name co ou = flex
!
crypto pki certificate chain CA
certificate 03
certificate ca 01
crypto ikev2 authorization policy default
route set interface
!
crypto ikev2 profile default
match certificate mymap
identity local fqdn R3-Spoke.cisco.com
authentication local rsa-sig
authentication remote rsa-sig
pki trustpoint CA
dpd 60 2 on-demand
aaa authorization group cert list default default
!
!
!
!
crypto ipsec profile default
set ikev2-profile default
!

```



```

!
!
!
!
interface Tunnel0
ip address negotiated
ip nhrp map multicast
ip nhrp map
ip nhrp nhs
mpls bgp forwarding
tunnel source Ethernet0/0
tunnel destination 172.16.0.1
tunnel protection ipsec profile default
!
interface Ethernet0/0
description WAN
ip address 172.16.1.103 255.255.255.0
!
interface Ethernet0/1
description LAN
no ip address
no ip unreachable
!
interface Ethernet0/1.10
encapsulation dot1Q 10
vrf forwarding cust1
ip address 192.168.113.1 255.255.255.0
!
interface Ethernet0/1.20
encapsulation dot1Q 20
vrf forwarding cust2
ip address 192.168.123.1 255.255.255.0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.1 remote-as 10
neighbor 10.0.0.1 ebgp-multihop 255
neighbor 10.0.0.1 update-source Tunnel0
!
address-family ipv4
neighbor 10.0.0.1 activate
exit-address-family
!
address-family vpnv4
neighbor 10.0.0.1 activate
neighbor 10.0.0.1 send-community both
exit-address-family
!
address-family ipv4 vrf cust1
redistribute connected
exit-address-family
!
address-family ipv4 vrf cust2
redistribute connected
exit-address-family
!
ip route 10.0.0.1 255.255.255.255 Tunnel0 name workaround
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB

```

The following is the configuration on Spoke B.

```

hostname R4-Spoke
!
vrf definition cust1

```

```

rd 1:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family
!
vrf definition cust2
rd 2:2
route-target export 2:2
route-target import 2:2
!
address-family ipv4
exit-address-family
!
clock timezone CET 1 0
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
crypto pki token default removal timeout 0
!
crypto pki trustpoint CA
enrollment url http://172.16.1.1:80
password
fingerprint E0AFEF7D7F08070BAB33C8297C97E6457
subject-name cn=R4-Spoke.cisco.com,OU=Flex,O=Cisco
revocation-check crl none
!
crypto pki certificate map mymap 10
subject-name co ou = flex
!
crypto pki certificate chain CA
certificate 04
certificate ca 01
!
crypto ikev2 authorization policy default
route set interface
!
crypto ikev2 profile default
match certificate mymap
identity local fqdn R4.cisco.com
authentication local rsa-sig
authentication remote rsa-sig
pki trustpoint CA
dpd 60 2 on-demand
aaa authorization group cert list default default
virtual-template 1
!
crypto ipsec profile default
set ikev2-profile default
!
interface Loopback100
vrf forwarding cust1
ip address 192.168.114.1 255.255.255.0
!
interface Loopback101
vrf forwarding cust2
ip address 192.168.124.1 255.255.255.0
!
interface Tunnel0
ip address negotiated

```

```

mpls bgp forwarding
tunnel source Ethernet0/0
tunnel destination 172.16.0.1
tunnel protection ipsec profile default
!
interface Ethernet0/0
description WAN
ip address 172.16.1.104 255.255.255.0
!
interface Ethernet0/1
description LAN
ip address 192.168.104.1 255.255.255.0
!
router bgp 100
bgp log-neighbor-changes
neighbor 10.0.0.1 remote-as 10
neighbor 10.0.0.1 ebgp-multihop 255
neighbor 10.0.0.1 update-source Tunnel0
!
address-family ipv4
neighbor 10.0.0.1 activate
exit-address-family
!
address-family vpnv4
neighbor 10.0.0.1 activate
neighbor 10.0.0.1 send-community both
exit-address-family
!
address-family ipv4 vrf cust1
redistribute connected
exit-address-family
!
address-family ipv4 vrf cust2
redistribute connected
exit-address-family
!
ip route 10.0.0.1 255.255.255.255 Tunnel0
ip route 172.16.0.1 255.255.255.255 172.16.1.1 name FlexHUB
The hub configuration is as follows:
hostname R1-HUB
aaa new-model
!
!
aaa authorization network default local
!
!
clock timezone CET 1 0
!
ip vrf cust1
rd 1:1
route-target export 1:1
route-target import 1:1
!
ip vrf cust2
rd 2:2
route-target export 2:2
route-target import 2:2
!
no ip domain lookup
ip domain name cisco.com
ip cef
no ipv6 cef
!
multilink bundle-name authenticated

```

```

mpls ldp loop-detection
!
crypto pki trustpoint CA
enrollment url http://172.16.0.2:80
password
fingerprint E0AFEF7F08070BAB33C8297C97E6457
subject-name CN=R1-HUB.cisco.com,OU=FLEX,OU=VPN,O=Cisco Systems,C=US,L=Linux
revocation-check crl none
rsa-keypair R1-HUB.cisco.com 2048
auto-enroll 95
!
!
crypto pki certificate chain CA
certificate 02
certificate ca 01
!
redundancy
!
!
!
crypto ikev2 authorization policy default
pool mypool
banner ^C Welcome ^C
def-domain cisco.com
!
!
!
!
crypto ikev2 profile default
match identity remote fqdn domain cisco.com
identity local dn
authentication local rsa-sig
authentication remote rsa-sig
pki trustpoint CA
dpd 60 2 on-demand
aaa authorization group cert list default default
virtual-template 1
!
crypto ipsec profile default
set ikev2-profile default
!
!
!
!
!
interface Loopback0
description VT source interface
ip address 10.0.0.1 255.255.255.255
!
interface Ethernet0/0
description WAN
ip address 172.16.0.1 255.255.255.252
!
interface Ethernet0/1
description LAN
ip address 192.168.100.1 255.255.255.0
!
interface Ethernet0/2
ip vrf forwarding cust1
ip address 192.168.110.1 255.255.255.0
!
interface Ethernet0/3
ip vrf forwarding cust2

```

```

ip address 192.168.111.1 255.255.255.0
!
interface Virtual-Templatel type tunnel
ip unnumbered Loopback0
ip nhrp network-id 1
ip nhrp redirect
mpls bgp forwarding
tunnel protection ipsec profile default
!
router bgp 10
bgp log-neighbor-changes
bgp listen range 0.0.0.0/0 peer-group mpls
bgp listen limit 5000
neighbor mpls peer-group
neighbor mpls remote-as 100
neighbor mpls transport connection-mode passive
neighbor mpls update-source Loopback0
!
address-family ipv4
redistribute static route-map global
neighbor mpls activate
neighbor mpls next-hop-self
exit-address-family
!
address-family vpnv4
neighbor mpls activate
neighbor mpls send-community both
exit-address-family
!
address-family ipv4 vrf cust1
redistribute connected
redistribute static route-map cust1
default-information originate
exit-address-family
!
address-family ipv4 vrf cust2
redistribute connected
redistribute static route-map cust2
default-information originate
exit-address-family
!
ip local pool mypool 10.1.1.1 10.1.1.254
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 172.16.0.2 name route_to_internet
ip route vrf cust1 0.0.0.0 0.0.0.0 Null0 tag 666 name default_originate
ip route vrf cust2 0.0.0.0 0.0.0.0 Null0 tag 667 name default_originate
!
route-map cust1 permit 10
match tag 666
!
route-map cust2 permit 10
match tag 667

```

The following is the spoke output:

```

R4-Spoke# show ip cef vrf cust1 192.168.110.1
192.168.110.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
sharing
sources: RIB
feature space:
IPRM: 0x00018000

```

```

LFD: 192.168.110.0/24 0 local labels
contains path extension list
ifnums: (none)
path EF36CA28, path list EF36DEB4, share 1/1, type recursive, for IPv4, flags must-be-labelled
MPLS short path extensions: MOI flags = 0x0 label 19
recursive via 10.0.0.1[IPv4:Default] label 19, fib F0C5926C, 1 terminal fib,
v4:Default:10.0.0.1/32
path EF36CBE8, path list EF36DFF4, share 1/1, type attached host, for IPv4
MPLS short path extensions: MOI flags = 0x1 label implicit-null
attached to Tunnel0, adjacency IP midchain out of Tunnel0 F0481718
output chain: label 19 label implicit-null TAG midchain out of Tunnel0 F1D97A90 IP adj out
of Ethernet0/0, addr 172.16.1.1 F0481848
R4-Spoke# show ip bgp vpnv4 all label
Network Next Hop In label/Out label
Route Distinguisher: 1:1 (cust1)
0.0.0.0 10.0.0.1 nolabel/18
192.168.110.0 10.0.0.1 nolabel/19
192.168.114.0 0.0.0.0 16/nolabel(cust1)
Route Distinguisher: 2:2 (cust2)
0.0.0.0 10.0.0.1 nolabel/20
192.168.111.0 10.0.0.1 nolabel/21
192.168.124.0 0.0.0.0 17/nolabel(cust2)

```

The following is the hub output:

```

R1-HUB# show ip cef vrf cust1 192.168.113.1 in
192.168.113.0/24, epoch 0, flags rib defined all labels, RIB[B], refcount 5, per-destination
sharing
sources: RIB, LTE
feature space:
IPRM: 0x00018000
LFD: 192.168.113.0/24 1 local label
local label info: other/25
contains path extension list
disposition chain 0xF1E1D9B0
label switch chain 0xF1E1D9B0
ifnums: (none)
path F16ECA10, path list F16EDFBC, share 1/1, type recursive, for IPv4, flags must-be-labelled
MPLS short path extensions: MOI flags = 0x0 label 16
recursive via 10.1.1.3[IPv4:Default] label 16, fib F0CCD6E8, 1 terminal fib,
v4:Default:10.1.1.3/32
path F16ECE00, path list F16EE28C, share 1/1, type attached host, for IPv4
MPLS short path extensions: MOI flags = 0x1 label implicit-null
attached to Virtual-Access1, adjacency IP midchain out of Virtual-Access1 F04F35D8
output chain: label 16 label implicit-null TAG midchain out of Virtual-Access1 F1E1DF60 IP
adj out of Ethernet0/0, addr 172.16.0.2 F04F3708
R1-HUB#sh ip bgp vpnv4 all
BGP table version is 49, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-Filter, a
additional-path
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf cust1)
*> 0.0.0.0 0.0.0.0 0 32768 ?
*> 192.168.110.0 0.0.0.0 0 32768 ?
*> 192.168.113.0 10.1.1.3 0 0 100 ?
*> 192.168.114.0 10.1.1.4 0 0 100 ?
Route Distinguisher: 2:2 (default for vrf cust2)
*> 0.0.0.0 0.0.0.0 0 32768 ?
*> 192.168.111.0 0.0.0.0 0 32768 ?
*> 192.168.123.0 10.1.1.3 0 0 100 ?
*> 192.168.124.0 10.1.1.4 0 0 100 ?
R1-HUB# show ip bgp vpnv4 all 192.168.113.1
BGP routing table entry for 1:1:192.168.113.0/24, version 48

```

```
Paths: (1 available, best #1, table cust1)
Advertised to update-groups:
3
Refresh Epoch 1
100
10.1.1.3 from *10.1.1.3 (172.16.1.103)
Origin incomplete, metric 0, localpref 100, valid, external, best
Extended Community: RT:1:1
mpls labels in/out 25/16
BGP routing table entry for 2:2:0.0.0.0/0, version 8
Paths: (1 available, best #1, table cust2)
Advertised to update-groups:
3
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.0.0.1)
Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
Extended Community: RT:2:2
mpls labels in/out 20/aggregate(cust2)
```

Restrictions for Configuring 6VPE and 6PE Support in MPLS over DMVPN Phase 2

- 6VPE and 6PE in DMVPN Phase 3 behaves like DMVPN Phase 1. All the packets from spoke-to-spoke travel through the hub as no dynamic spoke-to-spoke tunnel is created.
- In DMVPN Phase 2, if the dynamic spoke-to-spoke tunnel is not created for some reason, the packets do not travel through the hub, causing failure in connectivity.
- Initial packets from spoke-to-spoke travel in cleartext and drop by the hub until the dynamic tunnel is established between spokes.

Configuring 6VPE Support in MPLS over DMVPN Phase 2

To configure 6VPE support in MPLS over DMVPN phase 2, you must enable various components such as VRF, Tunnel, IPsec Tunnel Protection, WAN Facing Interface, Transport routing and Overlay Routing for the hub and the spokes.

Enabling Components for the Hub

To configure 6VPE support in MPLS over DMVPN phase 2 for the hub, you must enable the following in the order:

1. VRF
2. Tunnel
3. IPsec Tunnel Protection
4. WAN Facing Interface
5. Transport Routing

6. Overlay Routing

Configuring VRF for the Hub

```
enable
config terminal
vrf definition blue
rd 100:1
address-family ipv6
route-target export 100:1
route-target import 100:1
exit-address-family
vrf definition red
rd 100:2
address-family ipv6
route-target export 100:2
route-target import 100:2
exit-address-family
```

Enabling Tunnel for the Hub

```
interface Tunnell
ip address 192.168.1.1 255.255.255.0
no ip redirects
ip nhrp authentication cisco123
ip nhrp network-id 101
mpls nhrp
tunnel source GigabitEthernet0/0/1
tunnel mode gre multipoint
tunnel key 101
```

Enabling IPsec Tunnel Protection for the Hub

```
interface Tunnell
tunnel protection ipsec profile ipsec_ikev2
no shut
end
```

Enabling WAN Interfaces for the Hub

```
interface GigabitEthernet0/0/1
ip address 10.1.1.1 255.255.255.0
negotiation auto
cdp enable
ipv6 address 10::1/64
hold-queue 4096 in
hold-queue 4096 out
```

Enabling Transport Routing for the Hub

```
router eigrp 100
network 10.1.1.0 0.0.0.255
```

Enabling Overlay Routing for the Hub

```
router bgp 1
bgp router-id 192.168.1.1
bgp log-neighbor-changes
neighbor 192.168.1.101 remote-as 1
neighbor 192.168.1.101 update-source Tunnell
```



```

neighbor 192.168.1.102 remote-as 1
neighbor 192.168.1.102 update-source Tunnel1
address-family ipv4
neighbor 192.168.1.101 activate
neighbor 192.168.1.102 activate
exit-address-family
address-family vpnv6
neighbor 192.168.1.101 activate
neighbor 192.168.1.101 send-community extended
neighbor 192.168.1.101 route-reflector-client
no neighbor 192.168.1.101 next-hop-self all
neighbor 192.168.1.102 activate
neighbor 192.168.1.102 send-community extended
neighbor 192.168.1.102 route-reflector-client
no neighbor 192.168.1.102 next-hop-self all
exit-address-family
address-family ipv6 vrf blue
redistribute connected
exit-address-family
address-family ipv6 vrf red
redistribute connected
exit-address-family

```

Enabling the Components for the Spokes

To configure 6VPE support in MPLS over DMVPN phase 2, you must enable the following for the spokes in the order:

1. VRF
2. Tunnel
3. IPsec Tunnel Protection
4. WAN Facing Interface
5. PE-CE Interfaces
6. Transport Routing
7. Overlay Routing

Configuring VRF for the Spokes

```

vrf definition blue
 rd 100:1
  address-family ipv6
   route-target export 100:1
   route-target import 100:1
 exit-address-family
vrf definition red
 rd 100:2
  address-family ipv6
   route-target export 100:2
   route-target import 100:2
 exit-address-family

```

Enabling Tunnel for the Spokes

```
interface Tunnell
 ip address 192.168.1.101 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp map multicast 10.1.1.1
 ip nhrp map 192.168.1.1 10.1.1.1
 ip nhrp network-id 101
 ip nhrp nhs 192.168.1.1
 mpls nhrp
 tunnel source GigabitEthernet0/0/1
 tunnel mode gre multipoint
 tunnel key 101
```

Enabling IPsec Tunnel Protection for Spokes

```
interface Tunnell
 tunnel protection ipsec profile ipsec_ikev2
 no shut
 end
```

Enabling WAN Facing Interfaces for Spokes

```
interface GigabitEthernet0/0/1
 ip address 40.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 40::6/64
 ipv6 enable
```

Enabling PE-CE Interfaces for Spokes

```
interface GigabitEthernet0/0/3.1
 vrf forwarding blue
 encapsulation dot1q 1
 ip address 60.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 60::6/64
 ipv6 enable
interface GigabitEthernet0/0/3.2
 vrf forwarding red
 encapsulation dot1q 2
 ip address 80.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 80::6/64
 ipv6 enable
```

Enabling Transport Routing for Spokes

```
router eigrp 100
 network 40.1.1.0 0.0.0.255
```

Enabling Overlay Routing for the Spokes

```
router bgp 1
 bgp router-id 192.168.1.101
 bgp log-neighbor-changes
 neighbor 192.168.1.1 remote-as 1
 neighbor 192.168.1.1 update-source Tunnell
 address-family ipv4
 neighbor 192.168.1.1 activate
```

```

exit-address-family
address-family vpnv6
  neighbor 192.168.1.1 activate
  neighbor 192.168.1.1 send-community extended
exit-address-family
address-family ipv6 vrf blue
  redistribute connected
exit-address-family
address-family ipv6 vrf red
  redistribute connected
exit-address-family

```

Enabling Transport Routing for IPv6

The 6VPE over DMVPN with IPv6 transport feature allows IPv6 LAN prefixes over an IPv4 overlay neighbourhood created over an IPv6 DMVPN transport. Multi-tenant IPv6 LAN extension (L3VPN) over DMVPN supports IPv6 transport. It supports IPv6 transport and Inter-region connectivity with daisy-chained hubs.

```

!
ipv6 router eigrp 1
  eigrp router-id 1.1.1.1
!

```

Enabling WAN Interfaces for IPv6

```

!
interface GigabitEthernet2
  no ip address
  negotiation auto
  ipv6 address 172:16:1::1/64
  ipv6 eigrp 1
  no mop enabled
  no mop sysid
!
interface GigabitEthernet3
  no ip address
  negotiation auto
  ipv6 address 172:16:2::1/64
  ipv6 eigrp 1
  no mop enabled
  no mop sysid
!
interface GigabitEthernet4
  no ip address
  negotiation auto
  ipv6 address 172:16:3::1/64
  ipv6 eigrp 1
  no mop enabled
  no mop sysid

```

Enabling Tunnel for Hubs

The following configuration allows you one of the hubs to get daisy-chained with other hubs.

```

!
interface Tunnell
  ip address 50.0.1.1 255.255.0.0

```

```

ip nhrp network-id 1
ip nhrp nhs 50.0.2.2 nbma 172:16:52::52 multicast
ip nhrp nhs 50.0.2.3 nbma 172:16:53::53 multicast
ip nhrp nhs 50.0.3.4 nbma 172:16:54::54 multicast
load-interval 30
ipv6 mtu 1450
mpls nhrp
if-state nhrp
tunnel source Loopback0
tunnel mode gre multipoint ipv6
tunnel key 1
tunnel path-mtu-discovery
end

```

Enabling Tunnel for Spokes

```

!
interface Tunnell
ip address 50.0.1.6 255.255.0.0
ip nhrp network-id 1
ip nhrp nhs 50.0.1.1 nbma 172:16:51::51 multicast
load-interval 30
ipv6 mtu 1450
mpls nhrp
if-state nhrp
tunnel source Loopback0
tunnel mode gre multipoint ipv6
tunnel key 1
tunnel path-mtu-discovery
end

```

Configuring 6PE Support in MPLS over DMVPN Phase 2

To configure 6PE Support in MPLS over DMVPN Phase 2, you must enable various components such as Tunnel, IPsec Tunnel Protection, WAN Facing Interfaces, Transport Routing, and Overlay Routing for the hub and spokes.

Enabling Components for the Hub

To configure 6PE support in MPLS over DMVPN phase 2, you must enable the following in the order:

1. Tunnel
2. IPsec Tunnel Protection
3. WAN Facing Interfaces
4. PE-CE Interfaces
5. Transport Routing
6. Overlay Routing

Enabling Tunnel for Hub

```
interface Tunnell
 ip address 192.168.1.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp network-id 101
 mpls nhrp
 tunnel source GigabitEthernet0/0/1
 tunnel mode gre multipoint
 tunnel key 101
```

Enabling IPsec Tunnel Protection for the Hub

```
interface Tunnell
 tunnel protection ipsec profile ipsec_ikev2
 no shut
 end
```

Enabling WAN Facing Interfaces for Hub

```
interface GigabitEthernet0/0/1
 ip address 10.1.1.1 255.255.255.0
 negotiation auto
 cdp enable
 ipv6 address 10::1/64
 hold-queue 4096 in
 hold-queue 4096 out
```

Enabling Transport Routing for Hub

```
router eigrp 100
 network 10.1.1.0 0.0.0.255
```

Enabling Overlay Routing for Hub

```
router bgp 1
 bgp router-id 192.168.1.1
 bgp log-neighbor-changes
 neighbor 192.168.1.101 remote-as 1
 neighbor 192.168.1.101 update-source Tunnell
 neighbor 192.168.1.102 remote-as 1
 neighbor 192.168.1.102 update-source Tunnell
 address-family ipv4
  neighbor 192.168.1.101 activate
  neighbor 192.168.1.102 activate
 exit-address-family
 address-family ipv6
  redistribute connected
  neighbor 192.168.1.101 activate
  neighbor 192.168.1.101 send-community extended
  neighbor 192.168.1.101 route-reflector-client
  no neighbor 192.168.1.101 next-hop-self all
  neighbor 192.168.1.102 activate
  neighbor 192.168.1.102 send-community extended
  neighbor 192.168.1.102 route-reflector-client
  no neighbor 192.168.1.102 next-hop-self all
 exit-address-family
```

Enabling Components for the Spokes

To configure 6PE support in MPLS over DMVPN phase 2, you must enable the following for the spokes:

1. Tunnel
2. IPsec Tunnel Protection
3. WAN Facing Interface
4. PE-CE Interfaces
5. Transport Routing
6. Overlay Routing

Enabling Tunnel for Spokes

```
interface Tunnell
 ip address 192.168.1.101 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco123
 ip nhrp map multicast 10.1.1.1
 ip nhrp map 192.168.1.1 10.1.1.1
 ip nhrp network-id 101
 ip nhrp nhs 192.168.1.1
 mpls nhrp
 tunnel source GigabitEthernet0/0/1
 tunnel mode gre multipoint
 tunnel key 101
```

Enabling IPsec Tunnel Protection for Spokes

```
interface Tunnell
 tunnel protection ipsec profile ipsec_ikev2
 no shut
 end
```

Enabling WAN Facing Interfaces for Spokes

```
interface GigabitEthernet0/0/1
 ip address 40.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 40::6/64
 ipv6 enable
```

Enabling PE-CE Interface for Spokes

```
interface GigabitEthernet0/0/3.1
 encapsulation dot1q 1
 ip address 60.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 60::6/64
 ipv6 enable
interface GigabitEthernet0/0/3.2
 encapsulation dot1q 2
 ip address 80.1.1.6 255.255.255.0
 negotiation auto
 ipv6 address 80::6/64
 ipv6 enable
```

Enabling Transport Routing for Spokes

```
router eigrp 100
 network 40.1.1.0 0.0.0.255
```

Enabling Overlay Routing for Spokes

```
router bgp 1
 bgp router-id 192.168.1.101
 bgp log-neighbor-changes
 neighbor 192.168.1.1 remote-as 1
 neighbor 192.168.1.1 update-source Tunnell
 address-family ipv4
   neighbor 192.168.1.1 activate
 exit-address-family
 address-family ipv6
   redistribute connected
   neighbor 192.168.1.1 activate
   neighbor 192.168.1.1 send-community extended
 exit-address-family
```

Verifying the 6VPE support in MPLS over DMVPN Phase 2 Configurations

Use the following show commands to verify that the 6VPE support in MPLS over DMVPN phase 2 configurations are enabled on the router:

```
show ipv6 route vrf blue 60::/64
show ipv6 route vrf blue 70::/64
show mpls forwarding-table
show mpls forwarding-table vrf blue 60::/64 detail
show mpls forwarding-table vrf blue 70::/64 detail
show ipv6 cef vrf blue 60::/64
show ipv6 cef vrf blue 70::/64
show ipv6 cef vrf red 61::/64
show ipv6 cef vrf red 71::/64
show bgp vpnv6 unicast all
show dmvpn
show ip nhrp
```

Verifying the 6PE support in MPLS over DMVPN Phase 2 Configurations

Use the following show commands to verify that the 6PE support in MPLS over DMVPN phase 2 configurations are enabled on the router:

```
show ipv6 route vrf blue 60::/64
show ipv6 route vrf blue 70::/64
show mpls forwarding-table
show mpls forwarding-table vrf blue 60::/64 detail
show mpls forwarding-table vrf blue 70::/64 detail
show ipv6 cef vrf blue 60::/64
show ipv6 cef vrf blue 70::/64
show ipv6 cef vrf red 61::/64
show ipv6 cef vrf red 71::/64
```

```
show bgp ipv6 unicast
show dmvpn
show ip nhrp
```

Configure a Spoke Node as a P Node in MPLS over DMVPN Phase 3

To deploy spoke node as a P node, you must configure

- the spoke node as you would configure a P node in an MPLS L3VPN deployment
- the following NHRP and BGP enhancements on the spoke node:
 - Configure inspection of MPLS-labelled packets.
 - Configure BGP to import routes from NHRP.
 - Configure NHRP to import routes from BGP.

Configure Inspection of MPLS-labelled Packets

Configure the inspection of MPLS-labelled packets using the command **mpls nhrp inspect**.

```
interface tunnel-name
 ip address ipv4-address subnet-mask
 ...
 mpls nhrp inspect
 ...
```

Configure BGP to Import Routes from NHRP

Configure BGP to import routes from NHRP using the command **import nhrp**.

```
router bgp autonomous-system-number
 ...
 address-family vpnv4
 import nhrp
 ...
```

Configure NHRP to Import Routes from BGP

Configure NHRP to import routes from BGP using the command **import bgp**.

```
...
 address-family vpnv4
 import bgp autonomous-system-number
 ...
```

Feature Information for MPLS over 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 26: Feature Information for Configuring MPLS over DMVPN

Feature Name	Releases	Feature Information
6VPE and 6PE Support in MPLS over DMVPN	Cisco IOS XE Gibraltar 16.10.x	The 6VPE and 6PE Support in MPLS over DMVPN feature enables service providers running an MPLS/IPv4 infrastructure to offer IPv6 services without any major changes in the infrastructure. It enables IPv6 sites to communicate with each other over a DMVPN MPLS/IPv4 core network using MPLS label switched paths (LSPs).
Support for Spoke Nodes as P Nodes in MPLS over DMVPN Phase 3	Cisco IOS XE Amsterdam 17.2.1	In IOS XE Amsterdam 17.1.x and earlier releases, in an MPLS over DMVPN Phase 3 deployment you could configure a spoke node only as a PE node. From IOS XE Amsterdam 17.2.1, you can configure spoke node as either a P or PE node.



CHAPTER 17

DHCP Tunnels Support

The DHCP Tunnels Support feature provides the capability to configure the node (or spoke) of the generic routing encapsulation (GRE) tunnel interfaces dynamically using DHCP.

In a Dynamic Multipoint VPN (DMVPN) network, each participating spoke must have a unique IP address belonging to the same IP subnet. It is difficult for a network administrator to configure the spoke addresses manually on a large DMVPN network. Hence, DHCP is used to configure the spoke address dynamically on a DMVPN network.

- [Finding Feature Information, on page 237](#)
- [Restrictions for DHCP Tunnels Support, on page 237](#)
- [Information About DHCP Tunnels Support, on page 238](#)
- [How to Configure DHCP Tunnels Support, on page 239](#)
- [Configuration Examples for DHCP Tunnels Support, on page 241](#)
- [Additional References, on page 242](#)
- [Feature Information for DHCP Tunnels Support, on page 243](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

Restrictions for DHCP Tunnels Support

- The DHCP functionality of address validation is not supported on DMVPN.
- The DHCP IP address is not assigned to the spoke when configured in DMVPN phase 1.
- When you register the spoke to the hub using the `ip nhrp nhs {dynamic nbma nbma-address | FQDN-string} [multicast]` command, the unicast adjacency is only created after the session comes up.

- When using the Dual-hub single-DMVPN topology, Cisco DHCP server automatically changes the unicast flag to broadcast mode. To prevent this automatic change, run the following command on the Cisco DHCP server:

```
no ip dhcp auto-broadcast
```

- When DHCP is configured on an interface, the interface may take more time than usual to shutdown.

Information About DHCP Tunnels Support

DHCP Overview

DHCP is based on the Bootstrap Protocol (BOOTP), which provides the framework for passing configuration information to hosts on a TCP/IP network. DHCP adds the capability to automatically allocate reusable network addresses and configuration options to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocating network addresses to hosts. DHCP is built on a client/server model, where designated DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. See the “DHCP” section of the *Cisco IOS IP Addressing Configuration Guide* for more information.

DHCP Behavior on a Tunnel Network

DMVPN spoke nodes establish a tunnel with a preconfigured DMVPN next hop server (NHS) (hub node) and exchange IP packets with the NHS before an IP address is configured on the tunnel interface. This allows the DHCP client on the spoke and the DHCP relay agent or the DHCP server on the NHS to send and receive the DHCP messages. A DHCP relay agent is any host that forwards DHCP packets between clients and servers.

When the tunnel on a spoke is in the UP state or becomes active, the spoke establishes a tunnel with the preconfigured hub node. The tunnel formation may include setting up IP Security (IPsec) encryption for the tunnel between the spoke and the hub. DHCP receives the GRE tunnel interface UP notification only after the spoke establishes a tunnel with the hub. The DHCP client configured on the spoke must exchange the DHCP IP packets with the hub (DHCP relay agent or server) to obtain an IP address for the GRE tunnel interface. Therefore, the spoke-to-hub tunnel must be in active state before the GRE tunnel interface UP notification is sent to the DHCP server or the relay agent.

IP packets that are broadcast on the DMVPN spoke reach the DMVPN hub. The spoke broadcasts a DHCPDISCOVER message to the DHCP relay agent on the DMVPN hub, before the spoke has an IP address on the GRE tunnel interface. By using the DHCPDISCOVER message, DHCP unicasts the offer back to the client. The hub cannot send IP packets to the spoke before the hub receives a Next Hop Resolution Protocol (NHRP) registration from the spoke. The DHCP relay agent configured on the DMVPN hub adds mapping information to the DHCP client packets (DHCPDISCOVER and DHCPREQUEST).

Depending on whether the hub is a DHCP server or a DHCP relay agent, the mapping is handled differently.

- If the hub is a DHCP server, the Non-Broadcast Multiple Access (NBMA) address is known and a temporary mapping is created on the hub. The hub then unicasts a reply to the spoke.
- If the hub is a DHCP relay agent, the server behind the relay assigns the address. To preserve the NBMA address of the spoke, the address is attached to the DHCP message. When the reply is received, the NBMA address is fetched from the message. The address is sent to the spoke to create the mapping.



Note The NHRP registration sent by the spoke is suppressed until DHCP obtains an address for the GRE tunnel interface. Hence allows reliable exchange of standard DHCP messages.

DMVPN Hub as a DHCP Relay Agent

Relay agents are not required for DHCP to work. Relay agents are used only when the DHCP client and server are in different subnets. The relay agent acts as a communication channel between the DHCP client and server. The DHCP--Tunnels Support feature requires the DMVPN hub to act as a relay agent to relay the DHCP messages to the DHCP server.

The DHCP server is located outside the DMVPN network and is accessible from the DMVPN hub nodes through a physical path. The spoke nodes reach the DHCP servers through the hub-to-spoke tunnel (GRE tunnel). The DHCP server is not directly reachable from the DMVPN spoke. The DHCP relay agent on the DMVPN hub helps the DHCP protocol message exchange between the DHCP client on the spoke and the DHCP server.

DMVPN Topologies

Dual-Hub Single-DMVPN Topology

In a dual-hub single-DMVPN topology, both the hubs must be connected to the same DHCP server that has the high availability (HA) support to maintain DMVPN redundancy. If the hubs are connected to different DHCP servers, they must be configured with mutually exclusive IP address pools for address allocation.

Dual-Hub Dual-DMVPN Topology

In the dual-hub dual-DMVPN topology, each hub is connected to a separate DHCP server. The DMVPN hubs (DHCP relay agents) include a client-facing tunnel IP address in the relayed DHCP requests. DHCP requests are used by the DHCP server to allocate an IP address from the correct pool.

Hierarchical DMVPN Topology

In a DMVPN hierarchical topology, there are multiple levels of DMVPN hubs. However, all the tunnel interface IP addresses are allocated from the same IP subnet address. The DHCP client broadcast packets are broadcast to the directly connected hubs. Hence, the DMVPN hubs at all levels must either be DHCP servers or DHCP relay agents. If DHCP servers are used then the servers must synchronize their databases. The DMVPN hubs must be configured as DHCP relay agents to forward the DHCP client packets to the central DHCP servers. If the DHCP server is located at the central hub, all DHCP broadcasts are relayed through the relay agents until they reach the DHCP server.

How to Configure DHCP Tunnels Support

Configuring the DHCP Relay Agent to Unicast DHCP Replies

Perform this task to configure the DHCP relay agent (hub) to unicast DHCP replies.

By default, the DHCP replies are broadcast from the DMVPN hub to the spoke. Therefore a bandwidth burst occurs. The DHCP Tunnels Support feature does not function if the DHCP messages are broadcast. Hence, you must configure the DHCP relay agent to unicast the DHCP messages for the DHCP to be functional in a DMVPN environment.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp support tunnel unicast**
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip dhcp support tunnel unicast Example: Router(config)# ip dhcp support tunnel unicast	Configures a spoke-to-hub tunnel to unicast DHCP replies over the DMVPN network.
Step 4	exit Example: Router(config)# exit	Exits global configuration mode.

Configuring a DMVPN Spoke to Clear the Broadcast Flag

Perform this task to configure a DMVPN spoke to clear the broadcast flag.

By default, DMVPN spokes set the broadcast flag in the DHCP DISCOVER and REQUEST messages. Therefore the DHCP relay agent is forced to broadcast the DHCP replies back to the spokes, even though the relay agent has sufficient information to unicast DHCP replies. Hence, you must clear the broadcast flag from the DMVPN spoke.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel *number***

4. `ip dhcp client broadcast-flag clear`
5. `exit`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • 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 1</pre>	Configures a tunnel interface and enters interface configuration mode.
Step 4	ip dhcp client broadcast-flag clear Example: <pre>Router(config-if)# ip dhcp client broadcast-flag clear</pre>	Configures the DHCP client to clear the broadcast flag.
Step 5	exit Example: <pre>Router(config-if)# exit</pre>	Exits interface configuration mode and returns to global configuration mode.

Configuration Examples for DHCP Tunnels Support

Example Configuring a DHCP Relay Agent to Unicast DHCP Replies

The following example shows how to configure a DHCP relay agent to unicast DHCP replies:

```
Device# configure terminal
Device(config)# ip dhcp support tunnel unicast
Device(config)# exit
.
.
.
```

Example Configuring a DMVPN Spoke to Clear the Broadcast Flag and Set the IP Address to DHCP

The following example shows how to configure a DMVPN spoke to clear the broadcast flag and set the IP address to DHCP:

```
Device# configure terminal
Device(config)# interface tunnel 1
Device(config-if)# ip dhcp client broadcast-flag clear
Device(config-if)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS security commands	<i>Cisco IOS Security Command Reference</i>
Cisco IOS IP addressing configuration tasks	<i>Cisco IOS IP Addressing Configuration Guide</i>
Cisco IOS IP addressing services commands	<i>Cisco IOS IP Addressing Services Command Reference</i>

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
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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 2131	<i>Dynamic Host Configuration Protocol</i>

Technical Assistance

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

Feature Information for DHCP Tunnels Support

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 27: Feature Information for DHCP-Tunnels Support

Feature Name	Releases	Feature Information
DHCP--Tunnels Support	Cisco IOS XE Release 16.12	<p>The DHCP--Tunnels Support feature provides the capability to configure the node (or spoke) of the GRE tunnel interfaces dynamically using DHCP.</p> <p>The following commands were introduced or modified: ip address dhcp, ip dhcp client broadcast-flag, ip dhcp support tunnel unicast.</p>



APPENDIX **A**

Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)

The document explains the support for Per-Tunnel QoS configurations using port-channels (referred to as multiple policy maps (MPOL)) on the Cisco 4000 Series Integrated Services Routers and Cisco ASR 1000 Series Aggregation Routers.

- [Prerequisites Per-Tunnel QoS Support for Multiple Policy Maps \(MPOL\)](#), on page 245
- [Information About Per-Tunnel QoS Support for Multiple Policy Maps \(MPOL\)](#), on page 246
- [How to Configure Per-Tunnel QoS Support for Multiple Policy Maps \(MPOL\)](#), on page 247
- [Additional References for Per-Tunnel QoS Support for Multiple Policy Maps \(MPOL\)](#), on page 250

Prerequisites Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)

The following command must be configured before Per-Tunnel QoS is applied on a port-channel interface as the tunnel source:

```
platform qos port-channel-aggregate port-channel-interface-number
```

If a port-channel is already configured, the above command will fail. This command must be defined *before* configuring the port-channel, else, the following error occurs:

```
Port-channel 1 has been configured with non-aggregate mode already, please use different interface number that port-channel interface hasn't been configured
```

If you encounter the above error you must delete the port-channel and reconfigure the port-channel by using this command.

Information About Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)

Per-Tunnel QoS and Multiple Policy Maps (MPOL)

Per-Tunnel QoS offers the ability to police traffic in a hub-and-spoke environment on a per-spoke basis. Per-Tunnel QoS is configured on a hub router to ensure that the circuit bandwidth in the download direction at the spoke does not go beyond the circuit bandwidth. This is because the aggregate bandwidth on the hub router is significantly higher than the spoke.

However, there are various network designs or configurations that may be considered in the context of the Per-Tunnel QoS feature. One design, which is becoming more prevalent in today's networks, is sourcing tunnels on these hub routers from port-channel main or subinterfaces.

Supported Configurations

The following table lists MPOL configurations and the releases in which the support is available:

MPOL Configurations	On Cisco ASR 1000 Series	On Cisco 4000 Series
MPOL with tunnel sourced from port-channel main interface	Cisco IOS XE Everest16.5.1	Cisco IOS XE Everest 16.6.1
MPOL with tunnel sourced from port-channel sub-interface	Cisco IOS XE 3.16.4S/Cisco IOS XE Everest16.4.1	Cisco IOS XE Everest16.6.1
MPOL with tunnels sourced from different port-channel sub-interfaces of the same port-channel main interface	Cisco IOS XE Denali 16.3.6/Cisco IOS XE Everest16.6.3/Cisco IOS XE Fuji 16.8.1	Cisco IOS XE Everest16.6.1
MPOL with two tunnels in different VRF's sourced from the same port-channel sub-interface in a third VRF	Cisco IOS XE 3.16.4S/Cisco IOS XE Everest16.4.1	Cisco IOS XE Everest16.6.1

Components in MPOL

Before configuring MPOL, it is important to understand each component in reference to the broader solution thereby helping in understanding the supported and recommended configurations in each component.

Class Maps

Class maps segment traffic to match the Differentiated Services Code Point (DSCP) profile supported by the service provider. You mark traffic on ingress to any number of DSCP that are supported in your enterprise network. Alternatively, these markings could be available from a LAN device, which handles the marking for the site. However, on egress of the tunnel, the markings must be grouped into a set of DSCP supported by the class model defined by the ISP for the customer (4-class, 8-class, etc.).

Policy Maps

Child policy map provide a common queuing policy to each spoke. This policy map groups the DSCP into a smaller subset of classes and provide queuing definition as well as sets the tunnel DSCP for egress marking.

Per-Spoke Policy Maps

Policy maps are applied to each spoke based on NHRP group registration. These policy maps are defined according to the download speeds at the spokes. Typically, the policy maps may be grouped into a select number of values and a policy map exists for each value. It is within these values that a child policy map is nested to provide queuing in the context of the policed rate.

Traffic Shaping

A *Flat* policy map is used for shaping traffic that is applied on the parent WAN interface. This WAN interface acts as the tunnel source (in our case, a port-channel interface of some type). This shaper ensures that the egress shaped rate outbound from the hub router does not exceed the specified upload speed.

How to Configure Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)

Setting Up MPOL Components

Configuring Policy Maps

```

policy-map WAN
  class INTERACTIVE-VIDEO
    bandwidth remaining percent 30
    random-detect dscp-based
    set dscp tunnel af41
  class STREAMING-VIDEO
    bandwidth remaining percent 10
    random-detect dscp-based
    set dscp tunnel af31
  class NET-CTRL
    bandwidth remaining percent 5
    set dscp tunnel cs6
  class CALL-SIGNALING
    bandwidth remaining percent 4
    set dscp tunnel af21
  class CRITICAL-DATA
    bandwidth remaining percent 25
    random-detect dscp-based
    set dscp tunnel af21
  class SCAVENGER
    bandwidth remaining percent 1
    set dscp tunnel af11
  class VOICE
    priority level 1
    police cir percent 10
    set dscp tunnel ef
  class class-default

```

```
bandwidth remaining percent 25
random-detect
```

Applying Policy Maps to Spoke

```
policy-map RS-GROUP-300MBPS-POLICY
class class-default
  shape average 300000000
  bandwidth remaining ratio 300
  service-policy WAN
```

Applying Shaping

```
policy-map TRANSPORT-1-SHAPE-ONLY
class class-default
  shape average 600000000
```

Enabling MPOL

The recommended configuration order for enabling MPOL is as follows:

1. Define the routers to use QoS on the port-channel interface that will be configured
2. Define policy shaper.
3. Define the port-channel interface and subinterface and apply the policy shaper.
4. Define class maps to match the ingress traffic or DSCP for egress marking.
5. Define the child policy map for queuing definition and setting the tunnel DSCP.
6. Define the per-spoke policy maps to shape traffic on each spoke based on NHRP group registration and nest the child policy map in each spoke
7. Apply the per-spoke policy-maps to the tunnel interfaces and define the tunnel source to be the port-channel main or subinterface.

```
platform qos port-channel-aggregate <#>
policy-map TRANSPORT-1-SHAPE-ONLY
  class class-default
    shape average 600000000
interface Port-channel1
!
interface Port-channel1.10
  ...
  service-policy output TRANSPORT-1-SHAPE-ONLY
interface Tunnel100
  nhrp map group SPOKE-10MBPS service-policy output SPOKE-POLICE-10MBPS
  ...
  tunnel source Port-channel1.10
```

Verifying MPOL Configuration

After configuring MPOL, use the following commands to verify that the NHRP group is attached to the respective peer and to display the active policy:

- **show dmvpn detail**
- **show policy-map**

Router# **show dmvpn detail**

```

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket
T1 - Route Installed, T2 - Nexthop-override
C - CTS Capable, I2 - Temporary
# Ent --> Number of NHRP entries with same NBMA peer
NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
UpDn Time --> Up or Down Time for a Tunnel
=====
Interface Tunnel10 is up/up, Addr. is 172.17.10.1, VRF ""
  Tunnel Src./Dest. addr: 192.168.10.1/MGRE, Tunnel VRF "IWAN-TRANSPORT-MPLS"
  Protocol/Transport: "multi-GRE/IP", Protect "IWAN-TRANSPORT-MPLS"
  Interface State Control: Disabled
  nhrp event-publisher : Disabled
Type:Hub, Total NBMA Peers (v4/v6): 2
# Ent Peer NBMA Addr Peer Tunnel Add State UpDn Tm Attrb Target Network
-----
  1 192.168.10.3      172.17.10.3    UP 00:00:08      D      172.17.10.3/32
NHRP group: RS-GROUP-30MBPS
  Output QoS service-policy applied: RS-GROUP-30MBPS-POLICY
Router# show policy-map multipoint tunnel 10

```

```

Interface Tunnel10 <--> 192.168.10.3
  Service-policy output: RS-GROUP-30MBPS-POLICY
  Class-map: class-default (match-any)
    122 packets, 14444 bytes
    30 second offered rate 1000 bps, drop rate 0000 bps
  Match: any
  Queueing
  queue limit 124 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 117/21166
  shape (average) cir 30000000, bc 120000, be 120000
  target shape rate 30000000
  bandwidth remaining ratio 300
  Service-policy : WAN
  Class-map: INTERACTIVE-VIDEO (match-all)
    0 packets, 0 bytes
    30 second offered rate 0000 bps, drop rate 0000 bps
  Match: dscp af41 (34)
  Queueing
  queue limit 124 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 30%
  Exp-weight-constant: 4 (1/16)
  Mean queue depth: 0 packets
  dscp      Transmitted      Random drop      Tail drop      Minimum
Maximum    Mark      pkts/bytes      pkts/bytes      pkts/bytes      thresh
thresh     prob
  QoS Set
  dscp tunnel af41
  Marker statistics: Disabled

```

Additional References for Per-Tunnel QoS Support for Multiple Policy Maps (MPOL)

Related Documents

Related Topic	Document Title
Security commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Per-Tunnel QoS	Per-Tunnel QoS for DMVPN
Cisco Intelligent WAN Deployment Guide	Cisco Validated Design Intelligent WAN Deployment Guide

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