Dynamic Multipoint VPN Configuration Guide, Cisco IOS XE Release 3S

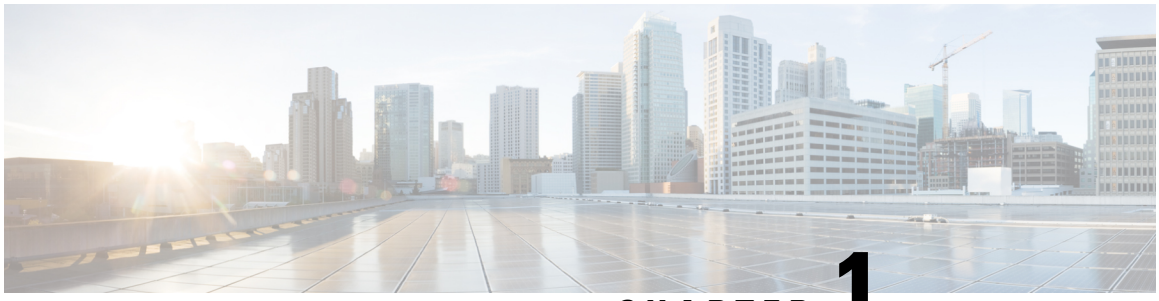
First Published: 2011-10-14

Last Modified: 2014-01-10

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

CONTENTS

CHAPTER 1



CHAPTER 1

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

- 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 mulitcast (PIM) is not supported over DMVPN. Therefore, you must use PIM Sparse mode (ASM) over DMVPN.
- 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.
- 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

- The Cisco ASR 1000 Series Routers support stateful IPSec sessions on Embedded Services Processor (ESP) switchover. During ESP switchover, all IPSec sessions will stay up and no user intervention is needed to maintain IPSec sessions.
- 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.
- The Cisco ASR 1000 Series Router currently does not support Stateful Switchover (SSO) IPSec sessions on Route Processors (RPs). The IPSec sessions will go down on initiation of the switchover, but will come back up when the new RP becomes active. No user intervention is needed. Traffic disruption might happen over the IPSec sessions for the duration of the switchover, until the sessions are back up.
- The Cisco ASR 1000 Series Router 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:

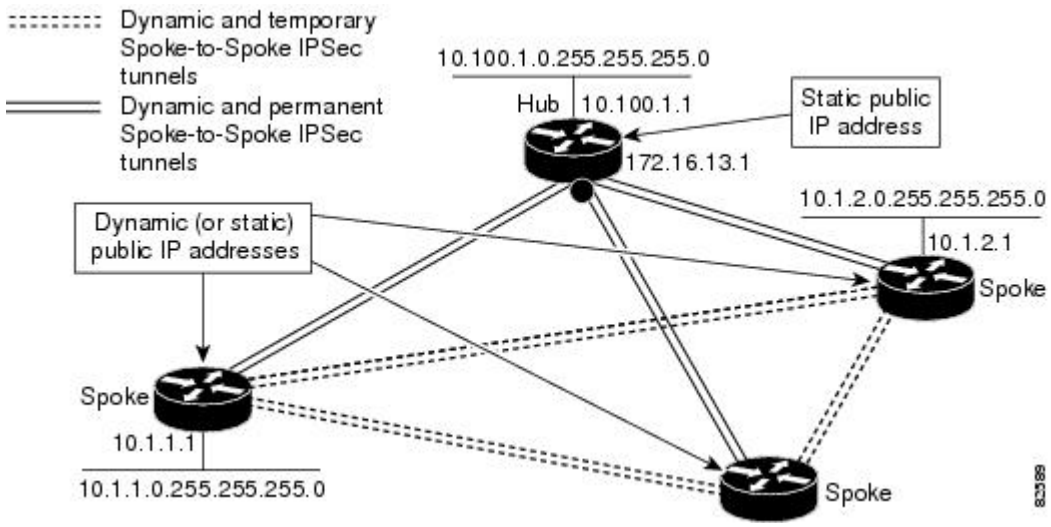
- NHRP--A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its

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

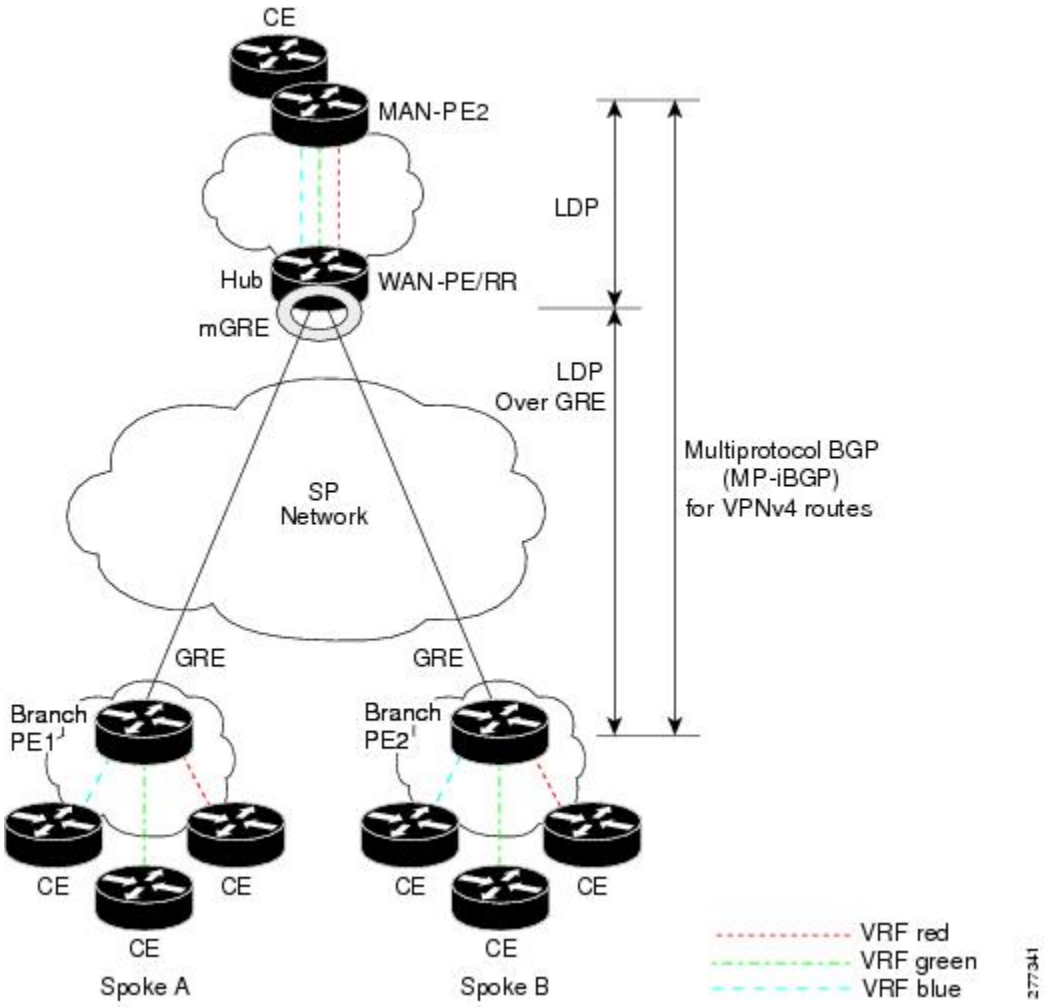
an entire crypto map configuration. An IPsec profile contains only IPsec information; that is, it does not contain any access list information or peering information.

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.

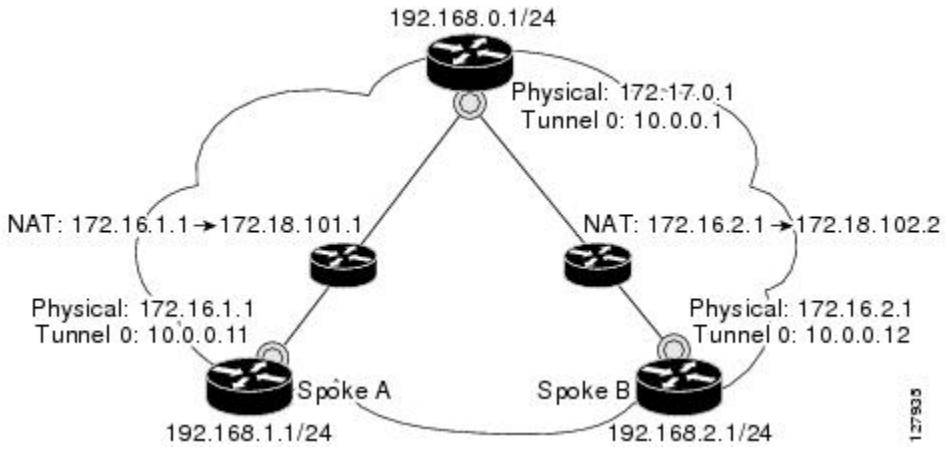
With this NAT Transparency enhancement, the hub DMVPN router can be behind the static NAT. For this functionality to be used, all the DMVPN spoke routers and hub routers must be upgraded, and IPsec must use transport mode.

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 preforming 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. • 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** *name*
	Example:
	`Router(config)#`
	`crypto ipsec profile vpnprof`
		Defines the IPsec parameters that are to be used for IPsec encryption between "spoke and hub" and "spoke and spoke" routers.
		• This command enters crypto map configuration mode.
		• The *name* argument specifies the name of the IPsec profile.

Step 4	**set transform-set** *transform-set-name* **Example:** `Router(config-crypto-map)# set transform-set trans2`	Specifies which transform sets can be used with the IPsec profile. • The *transform-set-name* argument specifies the name of the transform set.
Step 5	**set identity** **Example:** `Router(config-crypto-map)# set identity`	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 6	**set security association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*} **Example:** `Router(config-crypto-map)# set security association lifetime seconds 1800`	(Optional) Overrides the global lifetime value for the IPsec profile. • The **seconds** *seconds* option specifies the number of seconds a security association will live before expiring; the **kilobytes** *kilobytes*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 *seconds* argument is 3600 seconds.
Step 7	**set pfs** [**group1** | **group2**] **Example:** `Router(config-crypto-map)# set pfs group2`	(Optional) Specifies that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile. • If this command is not specified, the default (**group1**) 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. • 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 • The *number* 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	**ip address** *ip-address mask secondary* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.0`	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:** `Router(config-if)# ip mtu 1400`	Sets the maximum transmission unit (MTU) size, in bytes, of IP packets sent on an interface.
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	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 multicast dynamic** **Example:** `Router(config-if)# ip nhrp map multicast dynamic`	Allows NHRP to automatically add spoke routers to the multicast NHRP mappings. **Note** Effective with Cisco IOS XE Denali 16.3 **ip nhrp map multicast dynamic** is enabled by default.
Step 8	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network. The range is from 1 to 4294967295. **Note** Effective with Cisco IOS XE Denali 16.3 **ip nhrp network-id** is enabled by default.
Step 9	**tunnel source** {*ip-address* | *type number*} **Example:** `Router(config-if)# tunnel source Gigabitethernet 0/0/0`	Sets the source address for a tunnel interface.
Step 10	**tunnel key** *key-number* **Example:** `Router(config-if)# tunnel key 100000`	(Optional) Enables an ID key for a tunnel interface. • The *key-number* argument specifies a number from 0 to 4,294,967,295 that identifies the tunnel key. **Note** The key number must be set to the same value on all hubs and spokes that are in the same DMVPN network.

	Command or Action	Purpose
Step 11	**tunnel mode gre multipoint** **Example:** `Router(config-if)# tunnel mode gre multipoint`	Sets the encapsulation mode to mGRE for the tunnel interface.

Step 12	Do one of the following:
	• **tunnel protection ipsec profile** *name*
	• **tunnel protection psk** *key*
	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.
		• The *name* argument specifies the name of the IPsec profile; this value must match the *name* 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.

Step 13	**bandwidth** *kbps* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface to higher-level protocols. • The *kbps* 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** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1360`	Adjusts the maximum segment size (MSS) value of TCP packets going through a router. • The *max-segment-size* 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** *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 450`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. • The *seconds* 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** *number* **Example:** `Router(config-if)# delay 1000`	(Optional) Changes the EIGRP routing metric for routes learned over the tunnel interface. • The *number* 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:** `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** *number* **Example:** `Router(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • The *number* 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:** `Router(config-if)# ip address 10.0.0.2 255.255.255.0`	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:** `Router(config-if)# ip mtu 1400`	Sets the MTU size, in bytes, of IP packets sent on an interface.
Step 6	**ip nhrp authentication** *string* **Example:** `Router(config-if)# ip nhrp authentication donttell`	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:** `Router(config-if)# ip nhrp map 10.0.0.1 172.17.0.1`	Statically configures the IP-to-NBMA address mapping of IP destinations connected to an NBMA network. • *hub-tunnel-ip-address* --Defines the NHRP server at the hub, which is permanently mapped to the static public IP address of the hub. • *hub-physical-ip-address* --Defines the static public IP address of the hub.

	Command or Action	Purpose
Step 8	**ip nhrp map multicast** *hub-physical-ip-address* **Example:** `Router(config-if)# ip nhrp map multicast 172.17.0.1`	Enables the use of a dynamic routing protocol between the spoke and hub, and sends multicast packets to the hub router.
Step 9	**ip nhrp nhs** *hub-tunnel-ip-address* **Example:** `Router(config-if)# ip nhrp nhs 10.0.0.1`	Configures the hub router as the NHRP next-hop server.
Step 10	**ip nhrp network-id** *number* **Example:** `Router(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • The *number* argument specifies a globally unique 32-bit network identifier from a NBMA network. The range is from 1 to 4294967295. **Note** Effective with Cisco IOS XE Denali 16.3 **ip nhrp network-id** is enabled by default.
Step 11	**tunnel source** {*ip-address* | *type number*} **Example:** `Router(config-if)# tunnel source Gigabitethernet 0/0/0`	Sets the source address for a tunnel interface.
Step 12	**tunnel key** *key-number* **Example:** `Router(config-if)# tunnel key 100000`	(Optional) Enables an ID key for a tunnel interface. • The *key-number* 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	Do one of the following:
	• **tunnel mode gre multipoint**
	• **tunnel destination** *hub-physical-ip-address*
	Example:
	`Router(config-if)# tunnel mode gre multipoint`
	Example:
	`Router(config-if)# tunnel destination 172.17.0.1`
		Sets the encapsulation mode to mGRE for the tunnel interface.
		• Use this command if data traffic can use dynamic spoke-to-spoke traffic.
		Specifies the destination for a tunnel interface.
		• 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

	• **tunnel protection ipsec profile** *name*
	• **tunnel protection psk** *key*
	Example:
	`Router(config-if)# tunnel protection ipsec profile vpnprof`
	Example:
	`Router(config-if)# tunnel protection psk test1`
		• The *name* argument specifies the name of the IPsec profile; this value must match the *name* 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.

Step 15	**bandwidth** *kbps* **Example:** `Router(config-if)# bandwidth 1000`	Sets the current bandwidth value for an interface to higher-level protocols. • The *kbps* 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	**ip tcp adjust-mss** *max-segment-size* **Example:** `Router(config-if)# ip tcp adjust-mss 1360`	Adjusts the MSS value of TCP packets going through a router. • The *max-segment-size* 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	**ip nhrp holdtime** *seconds* **Example:** `Router(config-if)# ip nhrp holdtime 450`	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses. • The *seconds* argument specifies the time in seconds that NBMA addresses are advertised as valid in positive authoritative NHRP responses. The recommended value ranges from 300 seconds to 600 seconds.
Step 18	**delay** *number* **Example:** `Router(config-if)# delay 1000`	(Optional) Changes the EIGRP routing metric for routes learned over the tunnel interface. • The *number* 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. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface tunnel 0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip vrf forwarding** *vrf-name* **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:** `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** *type number* **Example:** `Router(config)# interface tunnel 0`	Configures an interface type and enters interface configuration mode.
Step 4	**tunnel vrf** *vrf-name* **Example:** `Router(config-if)# tunnel vrf RED`	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 9 and the Configuring the Spoke for DMVPN, on page 13. For details about VRF configuration, see the Configuring the Forwarding of Clear-Text Data IP Packets into a VRF, on page 17 and the Configuring the Forwarding of Encrypted Tunnel Packets into a VRF, on page 17.

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.**

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. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **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** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Enables configuration of the BGP routing process.
Step 4	**neighbor** *ipaddress* **remote-as** *as* **-** *number* **Example:** `Router(config-router)# neighbor 10.0.0.11 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table.

	Command or Action	Purpose
Step 5	**neighbor** *ipaddress* **update-source** *interface* **Example:** `Router(config-router)# neighbor 10.10.10.11 update-source Tunnel1`	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** *ipaddress* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.11 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** *ipaddress* **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.11 send-community extended`	Specifies that extended community attributes should be sent to a BGP neighbor.
Step 9	**neighbor** *ipaddress* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor 10.0.0.11 route-reflector-client`	Configures the router as a BGP Route Reflector and configures the specified neighbor as its client.
Step 10	**neighbor** *ipaddress* **route-map nexthop out** **Example:** `Router(config-router-af)# neighbor 10.0.0.11 route-map nexthop out`	Forces all traffic to be routed via the hub.
Step 11	**exit** **Example:** `Router(config-router-af)# exit`	Exits the address family configuration mode for VPNv4.
Step 12	**address-family ipv4** *vrf-name* **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 13	**redistribute connected** **Example:** `Router(config-router-af)# redistribute connected`	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 14	**route-map** *map-tag* [**permit**| **deny**] [*sequence-number*] **Example:** `Router(config-router-af)# route-map cisco permit 10`	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:** `Router(config-route-map)# set ip next-hop 10.0.0.1`	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:** `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	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 1`	Enters BGP configuration mode.
Step 4	**neighbor** *ipaddress* **remote-as** *as - number* **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** *ipaddress* **update-source** *interface* **Example:** `Router(config-router)# neighbor 10.10.10.1 update-source Tunnel1`	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** *ipaddress* **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 activate`	Enables the exchange of information with a BGP neighbor.
Step 8	**neighbor** *ipaddress* **send-community extended** **Example:** `Router(config-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** *vrf-name* **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:** `Router(config-router-af)# redistribute connected`	Redistributes routes that are established automatically by virtue of having enabled IP on an interface from one routing domain into another routing domain.
Step 12	**exit** **Example:** `Router(config-router-af)# exit`	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-pe1
boot-start-marker
boot-end-marker
no aaa new-model
resource policy
clock timezone EST 0
ip cef
no ip domain lookup
!This section refers to the forwarding table for VRF blue:
ip vrf blue
 rd 2:2
 route-target export 2:2
 route-target import 2:2
!This section refers to the forwarding table for VRF red:
ip vrf red
 rd 1:1
```

```
 route-target export 1:1
 route-target import 1:1
mpls label protocol ldp
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
crypto ipsec transform-set t1 esp-des
 mode transport
crypto ipsec profile prof
 set transform-set t1
interface Tunnel1
 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 Tunnel1
 neighbor 10.0.0.12 remote-as 1
 neighbor 10.0.0.12 update-source Tunnel1
 no auto-summary
 address-family vpnv4
 neighbor 10.0.0.11 activate
 neighbor 10.0.0.11 send-community extended
 neighbor 10.0.0.11 route-reflector-client
 neighbor 10.0.0.11 route-map nexthop out
 neighbor 10.0.0.12 activate
 neighbor 10.0.0.12 send-community extended
 neighbor 10.0.0.12 route-reflector-client
 neighbor 10.0.0.12 route-map nexthop out
 exit
 address-family 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 Tunnel1
 ip address 10.0.0.11 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1
!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source 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 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
```

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 Tunnel1
 ip address 10.0.0.12 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
```

```
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1
!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source 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 Tunnel1
 ip address 10.0.0.1 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp network-id 1
!EIGRP is enabled on the DMVPN network to learn the IGP prefixes:
no ip split-horizon eigrp 1
!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source 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 Tunnel1
 ip address 10.0.0.11 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1
!The command below enables MPLS on the DMVPN network:
```

```
 mpls ip
 tunnel source 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 Tunnel1
 ip address 10.0.0.12 255.255.255.0
 no ip redirects
 ip nhrp authentication cisco
 ip nhrp map multicast dynamic
 ip nhrp map 10.0.0.1 172.0.0.1
 ip nhrp map multicast 172.0.0.1
 ip nhrp network-id 1
 ip nhrp nhs 10.0.0.1
!The command below enables MPLS on the DMVPN network:
 mpls ip
 tunnel source 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, Tunnel1 via 10.9.9.12/32
    valid adjacency
    tag rewrite with Tu1, 10.0.0.1, tags imposed: {17 18}
Spoke2#
```

Sample Command Output: show ip bgp neighbors

```
Spoke2# show ip bgp neighbors

BGP neighbor is 10.9.9.1,  remote AS 1, internal link
  BGP version 4, remote router ID 10.9.9.1
  BGP state = Established, up for 00:02:09
  Last read 00:00:08, last write 00:00:08, hold time is 180, keepalive interval is 60 seconds

  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                4          4
    Keepalives:             4          4
    Route Refresh:          0          0
    Total:                  9          9
  Default minimum time between advertisement runs is 0 seconds
 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
```

```
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              0          0
    Prefixes Total:                0          0
    Implicit Withdraw:             0          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          0
    Used as multipath:           n/a          0
                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    Total:                                0          0
  Number of NLRIs in the update sent: max 0, min 0
 For address family: VPNv4 Unicast
  BGP table version 9, neighbor version 9/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                                Sent       Rcvd
  Prefix activity:              ----       ----
    Prefixes Current:              2          2 (Consumes 136 bytes)
    Prefixes Total:                4          2
    Implicit Withdraw:             2          0
    Explicit Withdraw:             0          0
    Used as bestpath:            n/a          2
    Used as multipath:           n/a          0
                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    ORIGINATOR loop:                    n/a          2
    Bestpath from this peer:              4        n/a
    Total:                                4          2
  Number of NLRIs in the update sent: max 1, min 1
  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled
Local host: 10.9.9.11, Local port: 179
Foreign host: 10.9.9.1, Foreign port: 12365
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x2D0F0):
Timer          Starts    Wakeups            Next
Retrans             6          0             0x0
TimeWait            0          0             0x0
AckHold             7          3             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss: 3328307266  snduna: 3328307756  sndnxt: 3328307756     sndwnd:  15895
irs: 4023050141  rcvnxt: 4023050687  rcvwnd:      16384  delrcvwnd:      0
SRTT: 165 ms, RTTO: 1457 ms, RTV: 1292 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, nagle, gen tcbs
IP Precedence value : 6
Datagrams (max data segment is 536 bytes):
Rcvd: 13 (out of order: 0), with data: 7, total data bytes: 545
Sent: 11 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0), with data:
 6, total data bytes: 489
Spoke2#
```

Additional References for Dynamic Multipoint VPN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Call Admission Control	*Call Admission Control for IKE*
IKE configuration tasks such as defining an IKE policy	*Configuring Internet Key Exchange for IPSec VPNs*
IPsec configuration tasks	*Configuring Security for VPNs with IPsec*
Configuring VRF-aware IPsec	*VRF-Aware IPsec*
Configuring MPLS	*Multiprotocol Label Switching (MPLS) on Cisco Routers*
Configuring BGP	*Cisco BGP Overview*
Defining and configuring ISAKMP profiles	*Certificate to ISAKMP Profile Mapping*
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).
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: • 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.

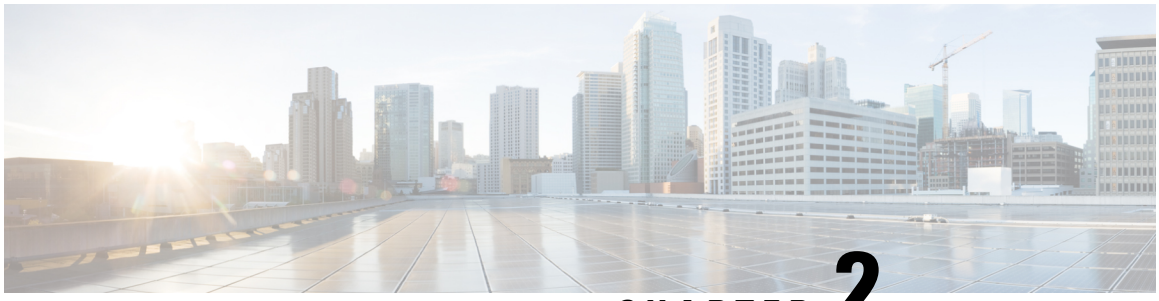
PFS--perfect forward secrecy. A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

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 2

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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.

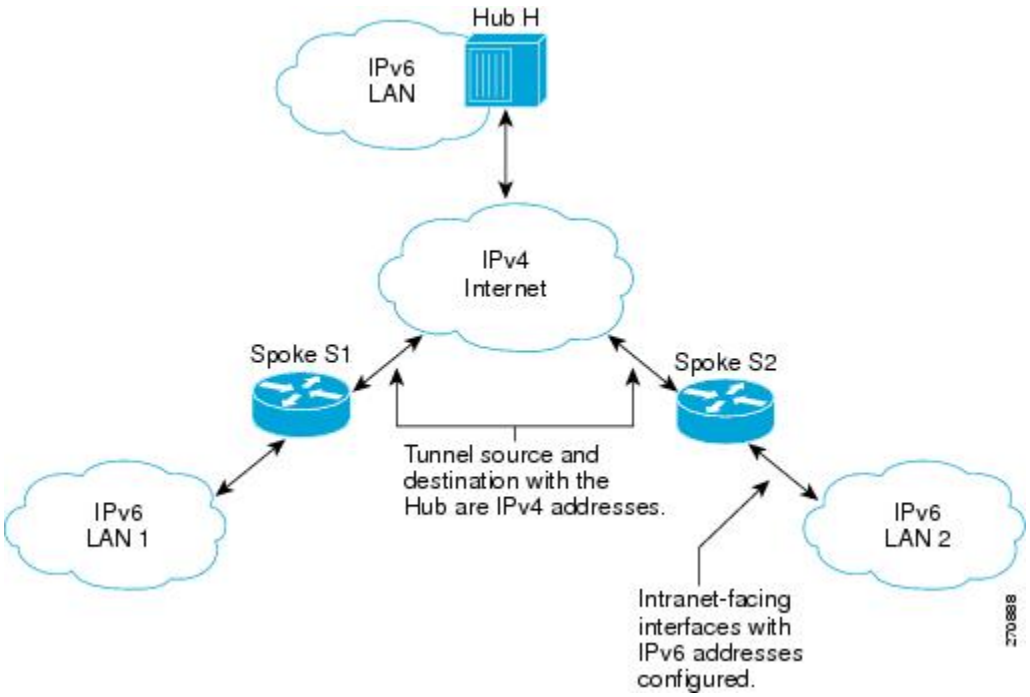
In DMVPN for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable. The intranets could be a mix of IPv4 or IPv6 clouds connected to each other using DMVPN technologies, with the underlying carrier being a traditional IPv4 network.

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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto identity** *name* **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** *name* **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** *transform-set-name* **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	**set security-association lifetime seconds** *seconds* | **kilobytes** *kilobytes* **Example:** `Device(config-crypto-map)# set security-association lifetime seconds 1800`	(Optional) Overrides the global lifetime value for the IPsec profile.
Step 9	**set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**] **Example:** `Device(config-crypto-map)# set pfs group14`	(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. • **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	end **Example:** `Device(config-crypto-map)# end`	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** {*kbps* | **inherit** [*kbps*] | **receive** [*kbps*]}
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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • 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:** `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** *ipv6-address* / *prefix-length* **link-local** **Example:**	Configures an IPv6 link-local address for an interface and enables IPv6 processing on the interface. • A unique IPv6 link-local address (across all DMVPN nodes in a DMVPN network) must be configured.

	Command or Action	Purpose
	`Device(config-if)# ipv6 address fe80::2001 link-local`	
Step 6	**ipv6 mtu** *bytes* **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** *string* **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** *network-id* **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** *ip-address* | *ipv6-address* | *interface-type interface-number* **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: • **tunnel protection ipsec profile** *name* [**shared**] • **tunnel protection psk** *key* **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof` **Example:**	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* 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
	`Router(config-if)# tunnel protection psk test1`	
Step 13	**bandwidth** {*kbps* | **inherit** [*kbps*] | **receive** [*kbps*]} **Example:** `Device(config-if)# bandwidth 1200`	Sets the current bandwidth value for an interface to higher-level protocols. • The *bandwidth-size* 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** *seconds* **Example:** `Device(config-if)# ipv6 nhrp holdtime 600`	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** *pkt-count* **every** *seconds* **Example:** `Device(config-if)# ipv6 nhrp max-send 10000 every 10`	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** [**timeout** *seconds* | **no-unique**] **Example:** `Device(config-if)# ip nhrp registration no-unique`	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:** `Device(config-if)# end`	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:** `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 number** **Example:** `Device(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • 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:** `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	Do one of the following:
	• **ipv6 nhrp redirect** [**timeout** *seconds*]
	• **ipv6 nhrp redirect** [**interest** *acl*]
	Example:
	`Device(config-if)# ipv6 nhrp redirect`
	Example:
	`Device(config-if)# ipv6 nhrp redirect interest`
		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:** `Device(config-if)# ipv6 nhrp shortcut`	Enables NHRP shortcut switching. • 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:** `Device(config-if)# end`	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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:**	Configures a tunnel interface and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface tunnel 5`	• 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:** `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** *ipv6-address* / *prefix-length* **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. • A unique IPv6 link-local address (across all DMVPN nodes in a DMVPN network) must be configured.
Step 6	**ipv6 mtu** *bytes* **Example:** `Device(config-if)# ipv6 mtu 1400`	Sets the MTU size of IPv6 packets sent on an interface.
Step 7	**ipv6 nhrp authentication** *string* **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** *ipv6-address nbma-address* **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. **Note** Only IPv4 NBMA addresses are supported, not ATM or Ethernet addresses.
Step 9	**ipv6 nhrp map multicast** *ipv4-nbma-address* **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** *ipv6- nhs-address* **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** *network-id* **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	**tunnel source** *ip-address* | *ipv6-address* | *interface-type interface-number* **Example:** `Device(config-if)# tunnel source ethernet 0`	Sets the source address for a tunnel interface.

Step 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*}
	Example:
	`Device(config-if)# tunnel mode gre multipoint`
	Example:
	`Device(config-if)# tunnel destination 10.1.1.1`
		Sets the encapsulation mode to mGRE for the tunnel interface.
		• Use the **tunnel mode** command if data traffic can use dynamic spoke-to-spoke traffic.
		or
		Specifies the destination for a tunnel interface.
		• Use the **tunnel destination** command if data traffic can use hub-and-spoke tunnels.

Step 14	Do one of the following:
	• **tunnel protection ipsec profile** *name* [**shared**]
	• **tunnel protection psk** *key*
	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.
		• The *name* argument specifies the name of the IPsec profile; this value must match the *name* 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.

Step 15	**bandwidth** {**interzone** | **total** | **session**} {**default** | **zone** *zone-name*} *bandwidth-size* **Example:** `Device(config-if)# bandwidth total 1200`	Sets the current bandwidth value for an interface to higher-level protocols. • The *bandwidth-size* 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** *seconds* **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. • 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	**show ipv6 nhrp** [**dynamic** [*ipv6-address*] | **incomplete** | **static**] [*address* | *interface*] [**brief** | **detail**] [**purge**] **Example:** `Device# show ipv6 nhrp`	Displays NHRP mapping information.
Step 4	**show ipv6 nhrp multicast** [*ipv4-address* | *interface* | *ipv6-address*] **Example:** `Device# show ipv6 nhrp multicast`	Displays NHRP multicast mapping information.
Step 5	**show ip nhrp multicast** [*nbma-address* | *interface*] **Example:** `Device# show ip nhrp multicast`	Displays NHRP multicast mapping information.
Step 6	**show ipv6 nhrp summary** **Example:** `Device# show ipv6 nhrp summary`	Displays NHRP mapping summary information.
Step 7	**show ipv6 nhrp traffic** [*interface***tunnel** *number* **Example:** `Device# show ipv6 nhrp traffic`	Displays NHRP traffic statistics information.
Step 8	**show ip nhrp shortcut** **Example:** `Device# show ip nhrp shortcut`	Displays NHRP shortcut information.
Step 9	**show ip route** **Example:** `Device# show ip route`	Displays the current state of the IPv4 routing table.
Step 10	**show ipv6 route** **Example:** `Device# show ipv6 route`	Displays the current contents of the IPv6 routing table.
Step 11	**show nhrp debug-condition** **Example:** `Device# show nhrp debug-condition`	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** *number*	**peer** {*ipv4-address*	*fqdn-string*	*ipv6-address*}	**vrf** *vrf-name*] [**static**] **Example:** `Device# clear dmvpn session`	Clears DMVPN sessions.
Step 3	**clear ipv6 nhrp** [*ipv6-address*	**counters** **Example:** `Device# clear ipv6 nhrp`	Clears all dynamic entries from the NHRP cache.
Step 4	**debug dmvpn** {**all**	**error**	**detail**	**packet**} {**all**	*debug-type*} **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** *number*	**peer** {**nbma** {*ipv4-address*	*fqdn-string*	*ipv6-address*}	**tunnel** {*ip-address*	*ipv6-address*}}	**vrf** *vrf-name*]	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
```

```
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: 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 Tunnel1 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: Tunnel1
  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: Tunnel1
  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
==========================================================================
```

```
Interface Tunnel1 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: Tunnel1
  IKE SA: local 192.169.2.10/500 remote 192.169.2.9/500 Active
  Crypto Session Status: UP-ACTIVE
  fvrf: (none), Phase1_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
   Tunnel1 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
   Tunnel1 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
   Tunnel1 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	*IPv6 Configuration Guide*
Dynamic Multipoint VPN	*Dynamic Multipoint VPN Configuration Guide*
Cisco IOS commands	Master Command List, All Releases
IPv6 commands	*IPv6 Command Reference*
Cisco IOS IPv6 features	IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFcs*

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

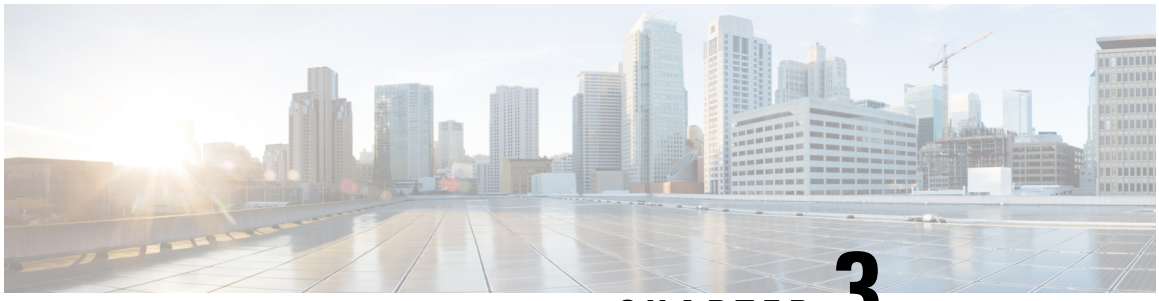
Feature Information for IPv6 over DMVPN

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	Cisco IOS XE Release 3.7S	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. 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**.
IPv6 Transport for DMVPN	Cisco IOS XE Release 3.8S	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.



CHAPTER 3

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

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 www.cisco.com/go/cfn. 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip name-server** *ip-address* **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. • 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** *hostname ip-address* **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:** `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** *number* **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode.
Step 4	**ip nhrp nhs** *nhs-address* [**nbma** {*nbma-address* | *FQDN-string*}] [**multicast**] [**priority** *value*] [**cluster** *number*] **Example:** `Router(config-if)# ip nhrp nhs 192.0.2.1 nbma examplehub.example1.com multicast`	Registers a spoke to a hub. • You can configure the command in the following two ways: • **ip nhrp nhs** *protocol-ipaddress* **nbma** *FQDN-string*--Use this command to register spoke to a hub using the FQDN string. • **ip nhrp nhs** *protocol-ipaddress* **nbma** *nbma-ipaddress*--Use this command to register spoke to a hub using the NHS NBMA IP address. **Note** You can use the **ipv6 nhrp nhs** *protocol-ipaddress* [**nbma** {*nhs-ipaddress* | *FQDN-string*}] [**multicast**] [**priority** *value*] [**cluster** *number*] command for registering IPv6 address.
Step 5	**end** **Example:** `Router(config-if)# end`	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. • 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 1`	Enters interface configuration mode.
Step 4	**ip nhrp nhs dynamic nbma** {*nbma-address* | *FQDN-string*} [**multicast**] [**priority** *value*] [**cluster** *value*] **Example:** `Router(config-if)# ip nhrp nhs dynamic nbma examplehub.example1.com`	Registers a spoke to a hub. • The NHS protocol address is dynamically fetched by the spoke. You can configure the command in the following two ways: • **ip nhrp nhs dynamic nbma** *FQDN-string*--Use this command to register a spoke to a hub using the FQDN string. • **ip nhrp nhs dynamic nbma** *nbma-address*--Use this command to register a spoke to a hub using the NHS NBMA IP address. **Note** You can use the **ipv6 nhrp nhs dynamic nbma** {*nbma-address* | *FQDN-string*} [**multicast**] [**priority** *value*] [**cluster** *value*] command for registering IPv6 address.
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: Tunnel1, 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
Tunnel1:
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 Tunnel1
 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
Tunnel1   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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
DMVPN complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

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	*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

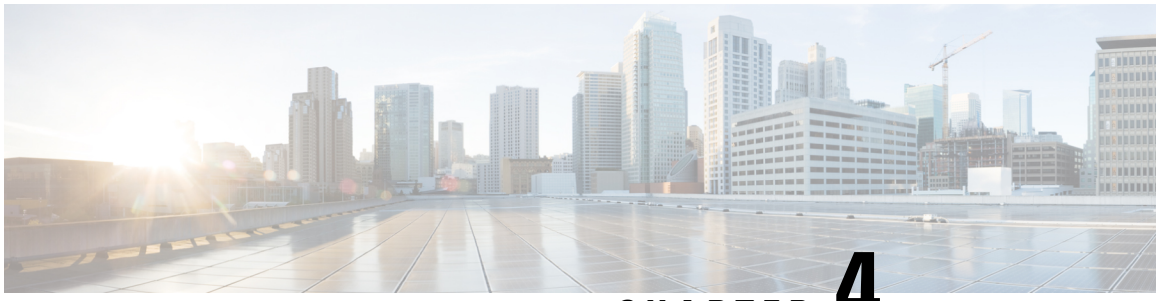
Feature Information for DMVPN Configuration Using FQDN

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	Cisco IOS XE Release 3.9S	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. The following commands were introduced or modified: **clear dmvpn session**, **debug nhrp condition**, **ip nhrp nhs**,and **ipv6 nhrp nhs**.



CHAPTER 4

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

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 www.cisco.com/go/cfn. 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 from 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 prirority 1 and 2, respectively, are grouped as cluster2, and NHS C1 and NHS C2 with prirority 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 1	Scenario 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 untill 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 1	Scenario 2	Scenario 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 maxmum 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 prirority 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** *number* **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode.
Step 4	**ip nhrp nhs cluster** *cluster-number* **max-connections** *value* **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** *cluster-number* **max-connections** *value* 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** *number* **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode.
Step 4	**ip nhrp nhs fallback** *fallback-time* **Example:** `Router(config-if)# ip nhrp nhs fallback 25`	Configures NHS fallback time. **Note** Use the **ipv6 nhrp nhs fallback** *fallback-time*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** *number* **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode.
Step 4	**ip nhrp nhs** *nhs-address* **priority** *nhs-priority* **cluster** *cluster-number* **Example:**	Configures the desired priority and cluster values. **Note** Use the **ipv6 nhrp nhs** *nhs-address* **priority** *nhs-priority* **cluster** *cluster-number* 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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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

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

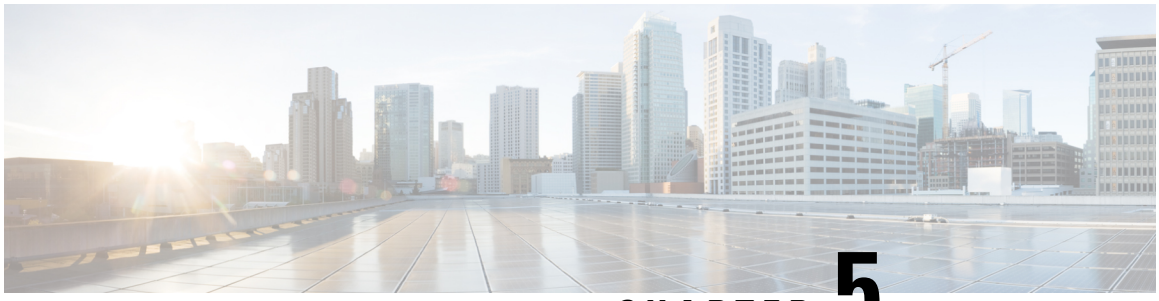
Feature Information for DMVPN-Tunnel Health Monitoring and Recovery Backup NHS

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)	Cisco IOS XE Release 3.9S	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. The following commands were introduced or modified: **ip nhrp nhs**, **ipv6 nhrp nhs**, **show ip nhrp nhs**, **show ipv6 nhrp nhs**.



CHAPTER 5

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

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 www.cisco.com/go/cfn. 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.

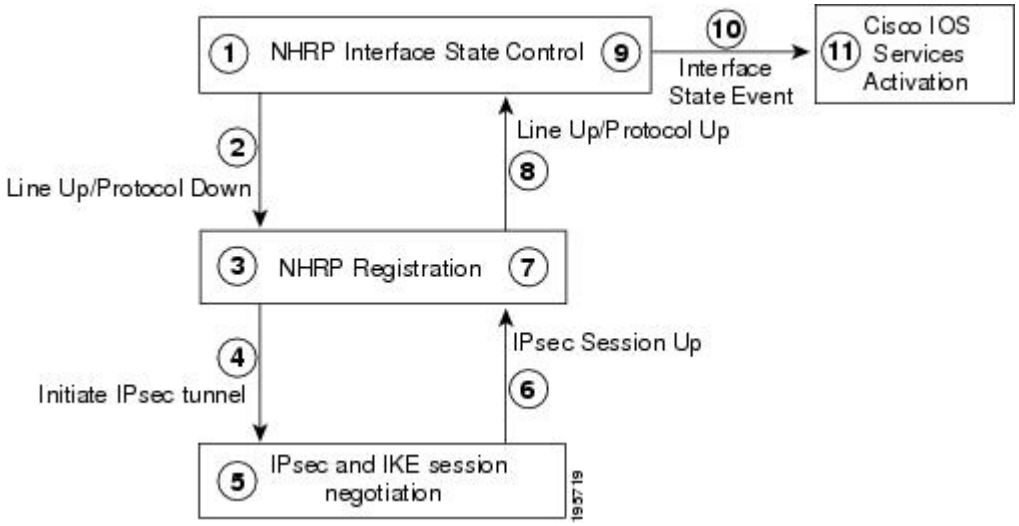
- If the interface state changes to down, the system clears any static routes that use the mGRE interface as the next hop. The Interface State Control feature provides a failover mechanism for routing when the mGRE interface is down.

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** *string* **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. • 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Dynamic Multipoint VPN information	"Dynamic Multipoint VPN (DMVPN)" module in the *Cisco IOS Security Configuration Guide: Secure Connectivity*
IKE configuration tasks such as defining an IKE policy	"Configuring Internet Key Exchange for IPsec VPNs" module in the *Cisco IOS Security Configuration Guide: Secure Connectivity*
IPsec configuration tasks	"Configuring Security for VPNs with IPsec" module in the *Cisco IOS Security Configuration Guide: Secure Connectivity*
System messages	*System Messages Guide*

Standards and RFCs

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

MIBs

MIB	MIBs Link
• 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

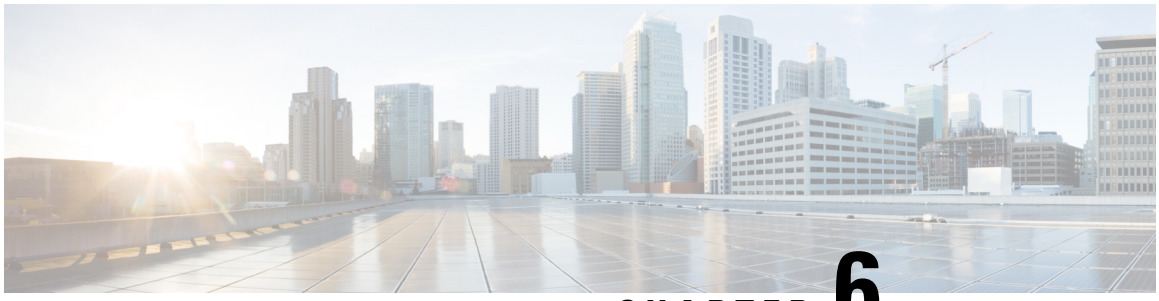
Feature Information for DMVPN Tunnel Health Monitoring and Recovery

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)	Cisco IOS XE Release 3.9S	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. In Cisco IOS XE Release 3.9S, this feature was implemented on Cisco CSR 1000V Series Cloud Services Router. In Cisco IOS XE Release 3.13S, this feature was implemented on Cisco 4000 Series Integrated Services Routers. In Cisco IOS XE Release 3.13.1S, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers. The following command was introduced: **if-state nhrp**.



CHAPTER 6

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

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 www.cisco.com/go/cfn. 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. • 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. • 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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
DMVPN commands	*Cisco IOS Security Command Reference*

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

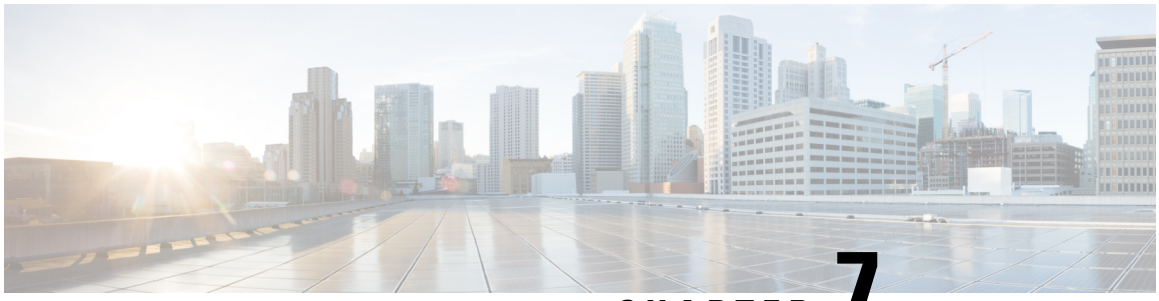
Feature Information for DMVPN Event Tracing

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	Cisco IOS XE Release 3.9S	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. The following commands were introduced or modified: **monitor event-trace dmvpn**, **show monitor event-trace dmvpn**.



CHAPTER 7

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NHRP MIB

- You should be familiar with configuring SNMP.

Restrictions for NHRP MIB

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

Information About NHRP MIB

CISCO-NHRP-MIB

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. • 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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Description of SNMP, SNMP MIBs, and how to configure SNMP on Cisco devices	"Configuring SNMP Support" chapter in the *Cisco IOS Network Management Configuration Guide*
Security commands	*Cisco IOS Security Command Reference*
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

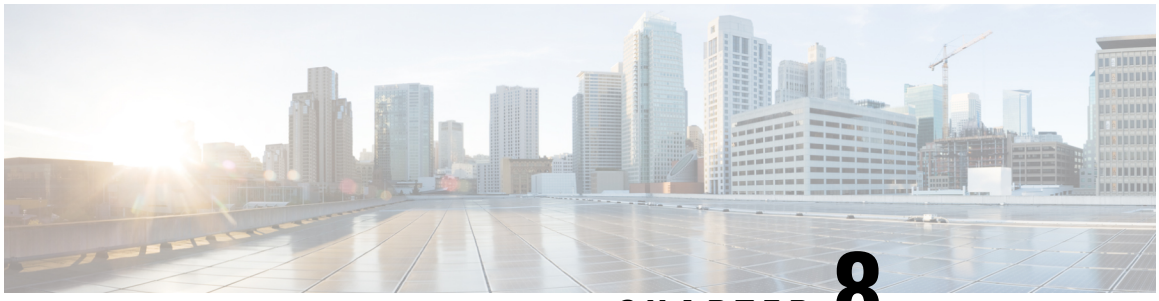
Feature Information for NHRP MIB

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	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. The following commands were introduced or modified: debug snmp mib nhrp**, show snmp mib nhrp status**.



CHAPTER 8

DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

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

Finding Feature Information

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

In order for 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 Peer Address Translation (PAT) is the type of NAT used on both NAT devices, then a session initiated between the two spokes cannot be established.

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

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

Information About DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

The following sections describe how 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:

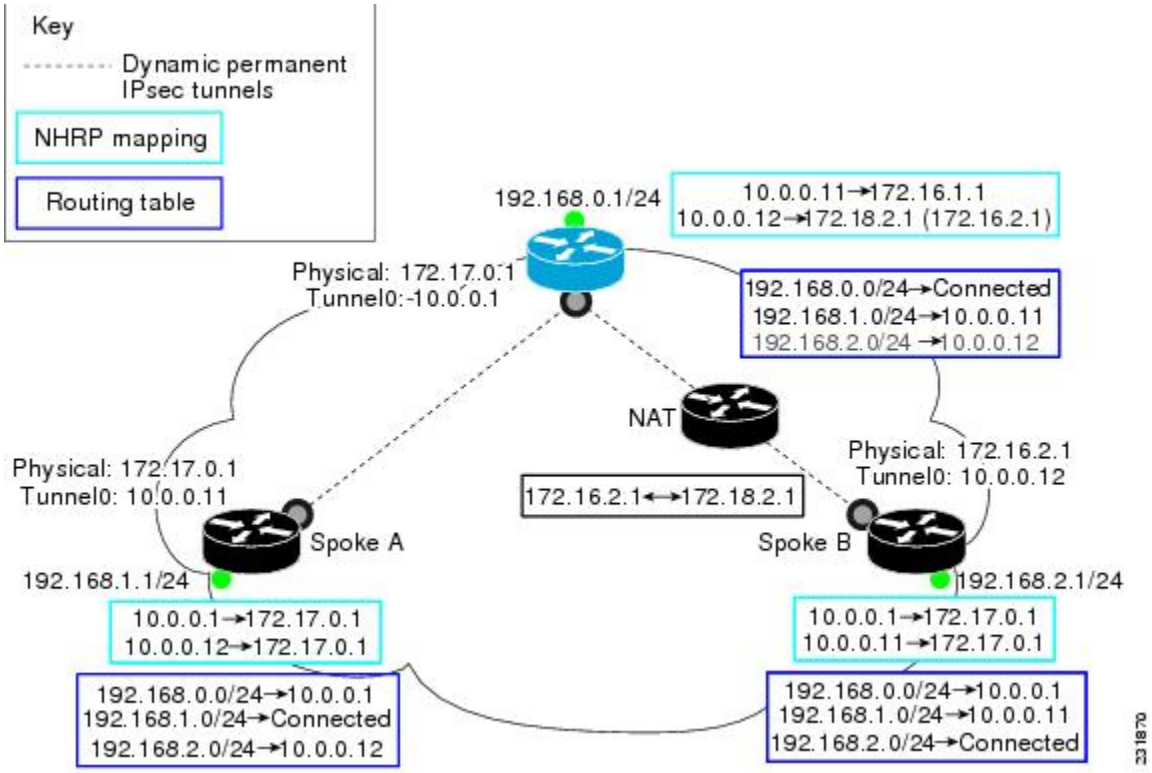
DMVPN Spoke-to-Spoke Tunneling Limited to Spokes Not Behind a NAT Device

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 "black-holed" 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.

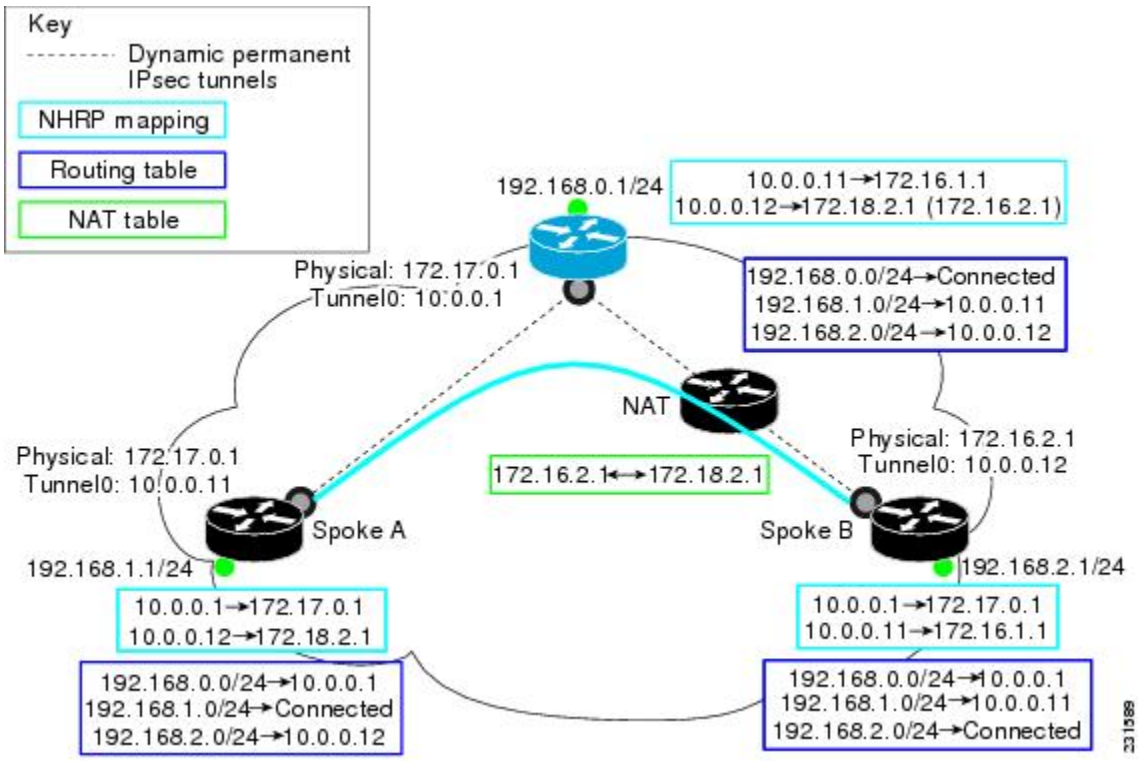
NHRP Spoke-to-Spoke Tunnel with a NAT Device

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 (black-holing) 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 (black-holed).

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 the192.168.1.0/24 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
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
NHRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*
Dynamic Multipoint VPN	"Dynamic Multipoint VPN (DMVPN)" module in the *Cisco IOS XE Security Configuration Guide: Secure Connectivity*

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

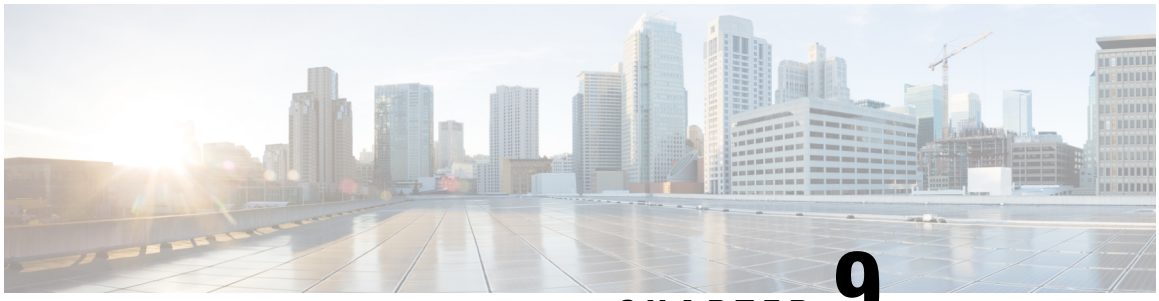
Feature Information for DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device

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 2.5	The DMVPN Dynamic Tunnels Between Spokes Behind a NAT Device feature allows NHRP spoke-to-spoke tunnels to be built in DMVPN networks, even if one or more spokes is behind a Network Address Translation (NAT) device. In Cisco IOS XE Release 2.5, this feature was introduced on the Cisco ASR 1000 Series Aggregation Routers.



CHAPTER 9

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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. • 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. • The *number* 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** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source GigabitEthernet 0`	Sets the source IP address or source interface type number for a tunnel interface. • 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** *name* [**shared]** **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof shared`	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto ipsec profile** *name* 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:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

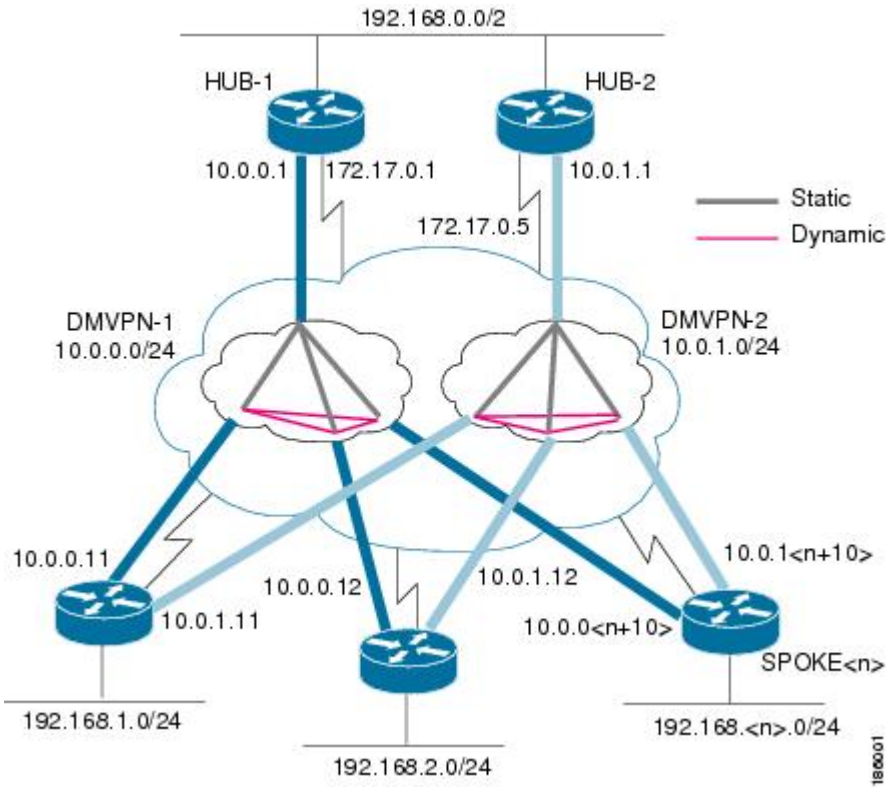
Configuration Examples for Sharing IPsec with Tunnel Protection

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
 !
 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	*IPv6 Configuration Guide*
Dynamic Multipoint VPN	*Dynamic Multipoint VPN Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
IPv6 commands	*IPv6 Command Reference*
Cisco IOS IPv6 features	IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFcs*

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 Release 2.5	The Sharing IPsec with Tunnel Protection feature allows an IPsec session to be shared between two or more GRE tunnel interfaces. In Cisco IOS XE Release 2.5, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was modified by this feature: **tunnel protection ipsec profile shared.**

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.

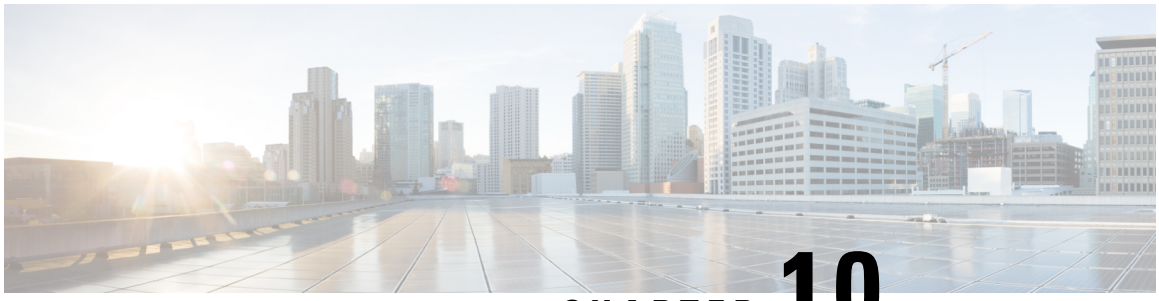
Both IPsec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPsec SA is established either by IKE or by manual user configuration.

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 10

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

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 www.cisco.com/go/cfn. 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

- The Per-Tunnel QoS for DMVPN feature only supports the following encapsulation and transport protocol combinations:
  - Per-Tunnel QoS for IPv4 over DMVPN with IPv4 transport (Effective from Cisco IOS XE Release 3.6S).
  - Per-Tunnel QoS for IPv6 over DMVPN with IPv4 transport (Effective from Cisco IOS XE Release 3.8S).
  - Per-Tunnel QoS for IPv4 over DMVPN with IPv6 transport (Effective from Cisco IOS XE Release 3.11S).
  - Per-Tunnel QoS for IPv6 over DMVPN with IPv6 transport (Effective from Cisco IOS XE Release 3.11S).
  - Per-Tunnel QoS for MPLS VPN over DMVPN with IPv4 transport (2547oDMVPN) (Effective from Cisco IOS XE Release 3.15S).
  - Per-Tunnel QoS for MPLS VPN over DMVPN with IPv6 transport (2547oDMVPN) (Effective from Cisco IOS XE Release 3.15S).
- For a given DMPVN tunnel interface, one transport protocol, either IPv4 or IPv6, can only be used. However, different DMVPN tunnel interfaces on the same device may use IPv4 or IPv6 transport protocol at the same time. Per-tunnel QoS can be configured for IPv4 and IPv6 DMVPN passenger traffic packets and be associated with an outbound physical interface that is either IPv4, IPv6 or both. This DMPVN tunnel traffic may be mixed with non-DMVPN IPv4 and IPv6 traffic, or both, on the outbound physical interface with its own QoS policy with restrictions.
- The Per-Tunnel QoS for DMVPN feature does not support the following:
  - Per-Tunnel QoS for IPv4 or IPv6 or Multiprotocol Label Switching (MPLS) VPN over DMVPN with Layer 2 Tunnel Protocol (L2TP) transport.
  - Per-Tunnel QoS for IPv4 or IPv6 or MPLS VPN over DMVPN.
- Per-Tunnel QoS service policies are only supported in the egress direction.
- This feature does not support adding the capability of user configurable queuing and schedules before the crypto engine.
- Fair queueing should not be used in a per-tunnel QoS for DMVPN policy map because the outer header with nonchanging IP addresses is used for individual flow queue selection. This results in the same queue being selected for all traffic flowing through the class with fair queueing.
- A QoS service policy is supported on the main interface or subinterface that the tunnel is sourced from in conjunction with a per-tunnel QoS service policy on the DMVPN tunnel interface. However, there are certain restrictions for the main or subinterface service policy, which are as follows:

- A service policy is supported on either the main interface or the subinterface, but not both, in conjunction with the per-tunnel QoS service policy.
- The main interface or subinterface QoS service policy is limited to only a class-default shaper (it can only contain the **class class-default** and **shape** commands). Additional QoS configurations are not supported on the main interface or subinterface when two different QoS service policies are applied to the main or subinterface and the tunnel interface simultaneously.
- The main interface or subinterface QoS service policy must be applied before the tunnel interface service policy.
- The main interface or subinterface QoS service policy is checked for validity only when a QoS service policy is applied on the tunnel interface. The main interface or subinterface service policy is not checked during a tunnel movement or modification.
- Adding new classes or features to the main interface or subinterface policy map is not supported. The classes or features may not be blocked on CLI and could result in unpredictable behavior.
- The policy-map counters for the main interface or subinterface service policy (from the **show policy-map interface** command) may not account for all packets and therefore should not be used or referenced. However, this does not affect the QoS functionality. The shaper will still limit the traffic on the main interface or subinterface, including all DMVPN tunnel traffic over that interface.

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 checka 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. Enter one of the following
   - **ip nhrp group** *group-name*
   - **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** *number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	Enter one of the following • **ip nhrp group** *group-name* • **nhrp group** *group-name* **Example:**	Configures a Next Hop Resolution Protocol (NHRP) group on the spoke.

	Command or Action	Purpose

	`Device(config-if)# ip nhrp group spoke_group1`
	Example:
	`Device(config-if)# nhrp group spoke_group1`	

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** *number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	**nhrp attribute group** *group-name* **Example:** `Device(config-if)# nhrp attribute group spoke1`	Configures the QoS group identity information on the spoke.
Step 5	**nhrp map group** *group-name* **service-policy output** *qos-policy-map-name* **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.

	Command or Action	Purpose
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. Do one of the following:
   - **ip nhrp map group** *group-name* **service-policy output** *qos-policy-map-name*
   - **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** *number* **Example:** `Device(config)# interface tunnel 1`	Configures a tunnel interface and enters interface configuration mode.
Step 4	Do one of the following: • **ip nhrp map group** *group-name* **service-policy output** *qos-policy-map-name* • **nhrp map group** *group-name* **service-policy output** *qos-policy-map-name* **Example:** `Device(config-if)# ip nhrp map group spoke_group1 service-policy output group1_parent` **Example:**	Adds the Next Hop Resolution Protocol (NHRP) group to the quality of service (QoS) policy mapping on the hub.

	Command or Action	Purpose
	`Device(config-if)# nhrp map group spoke_group1 service-policy output group1_parent`	
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Per-Tunnel QoS for DMVPN

SUMMARY STEPS

1. **enable**
2. **show dmvpn detail**
3. **show ip nhrp**
4. **show ip nhrp group** [*group-name*]
5. Do one of the following:
   - **show ip nhrp group-map** [*group-name*]
   - **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. • 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. • 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 ip nhrp** **Example:** `Device# show ip nhrp`	Displays the NHRP cache and the NHRP group received from the spoke.
Step 4	**show ip nhrp group** [*group-name*] **Example:**	Displays NHRP group mapping.

	Command or Action	Purpose
	`Device# show ip nhrp group`	• The output includes the associated QoS policy name and the list of tunnel endpoints using the QoS policy.

Step 5	Do one of the following:
	• **show ip nhrp group-map** [*group-name*]
	• **show nhrp group-map** [*group-name*]
	Example:
	`Device# show ip nhrp group-map group1-parent`
	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.

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
 ip 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
 ip 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
 ip 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
 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 h1there
 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: 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 Tunnel1 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: tunnel1
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: tunnel1
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: tunnel1
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
   Tunnel1 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
   Tunnel1 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
   Tunnel1 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 ip nhrp group-map

Interface: tunnel1
          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.

```
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
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *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

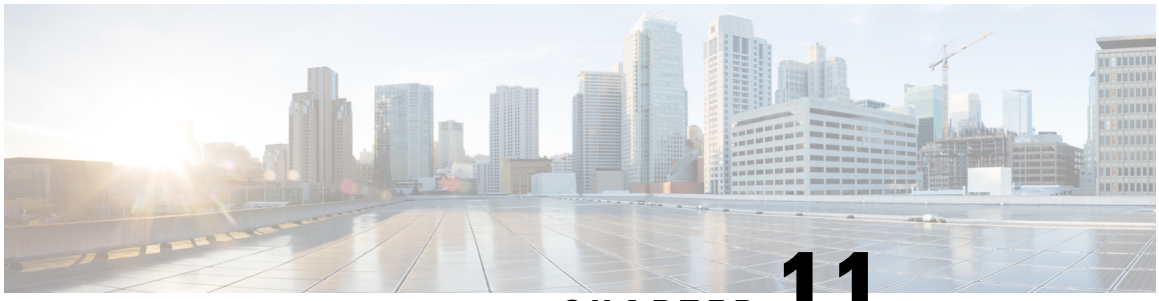
Feature Information for Per-Tunnel QoS for DMVPN

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	Cisco IOS XE Release 3.11S	The Per-Tunnel QoS for DMVPN feature introduces per-tunnel QoS support for DMVPN and increases per-tunnel QoS performance for IPsec tunnel interfaces. In Cisco IOS XE Release 3.11S, this feature was enhanced to provide support for IPv6 addresses. The following commands were introduced or modified: **ip nhrp group**, **ip nhrp map**, **ip nhrp map group**, **nhrp group**, **nhrp map group**, **show dmvpn**, **show ip nhrp**, **show ip nhrp group-map**, **show nhrp group-map**, **show policy-map multipoint tunnel**.
QoS: Spoke to Spoke Per-tunnel QoS for DMVPN	Cisco IOS XE Release 3.15S	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. The following commands were introduced or modified: **nhrp attribute group**, **show dmvpn**, **show ip nhrp**, **show ip nhrp group**.
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 on port-channel interface.



CHAPTER 11

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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.

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.

Packet Is Tagged	Inline Tagging Negotiated	IPsec Behavior
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.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel id* **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 adjaceny 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
                                   0101016300008909080001010001000 0
                                   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	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**crypto ikev2 cts sgt** **Example:** `Device(config)# crypto ikev2 cts sgt`	Enables TrustSec on DMVPN on IKEv2 networks. This command is valid for generic routing encapsulation (GRE) and to tunnel interfaces modes only.
Step 4	**exit** **Example:** `Device(config)# exit`	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.

```
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 209.165.201.1host 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
 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
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *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*
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

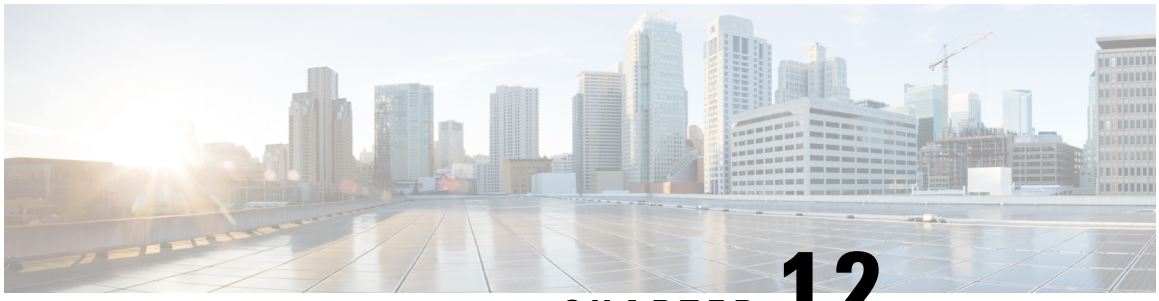
Feature Information for TrustSec DMVPN Inline Tagging Support

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	Cisco IOS XE Release 3.13S	The TrustSec DMVPN Inline Tagging Support feature enables IPsec to carry Cisco Trust Sec (CTS) Security Group Tag (SGT) between IPsec peers. The following commands were introduced or modified: **cts sgt inline**, **show dmvpn**, **show ip nhrp nhs**, **show tunnel endpoints**, **show adjacency**.



CHAPTER 12

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

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 www.cisco.com/go/cfn. 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.

Spoke-to-Spoke NHRP Summary Maps

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

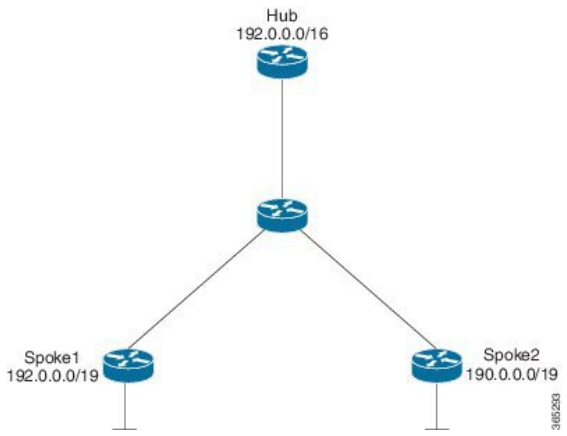
On receiving the resolution request, the spoke

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 starts 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 return 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 receing the resolution request for 192.0.12.1 checks the RIB which return 192.0.12.0/24. the local spoke then check 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. • 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** *number* **Example:** `Device(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • *number*—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 ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.2 255.255.255.0`	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 nhrp authentication** *string* **Example:** `Device(config-if)# ip nhrp authentication donttell`	Configures an authentication string for an interface using NHRP.
Step 6	**ip nhrp summary-map** {*ip-address* | *mask*} **Example:** `Device(config-if)# ip nhrp summary-map 10.0.0.0/24`	Summarizes and reduces the NHRP resolution traffic on the network.
Step 7	**ip nhrp network-id** *number* **Example:** `Device(config-if)# ip nhrp network-id 99`	Enables NHRP on an interface. • *number*—Specifies a globally unique 32-bit network identifier from a nonbroadcast multiaccess (NBMA) network.
Step 8	**ip nhrp nhs** [*hub-tunnel-ip-address*] **nbma** [*hub-wan--ip*] **multicast** **Example:** `Device(config-if)# ip nhrp nhs 10.0.0.1 nbma 172.17.0.1 multicast`	Configures the hub router as the NHRP next-hop server.
Step 9	**ip nhrp shortcut** **Example:** `Device(config-if)# ip nhrp shortcut`	Enables NHRP shortcut switching.

	Command or Action	Purpose
Step 10	**tunnel source** {*ip-address* | *type number*} **Example:** `Device(config-if)# tunnel source Gigabitethernet 0/0/0`	Sets the source address for a tunnel interface.
Step 11	**tunnel mode gre multipoint** **Example:** `Device(config-if)# tunnel mode gre multipoint`	Sets the encapsulation mode to Multiple Generic Routing Encapsulation (mGRE) for the tunnel interface. • Use this command if data traffic can use dynamic spoke-to-spoke traffic.
Step 12	**tunnel key** *key-number* **Example:** `Device(config-if)# tunnel key 100000`	(Optional) Enables an ID key for a tunnel interface. • *key-number*—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:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Spoke-to Spoke NHRP Summary Maps

SUMMARY STEPS

1. **enable**
2. **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
```

```
 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 commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	• 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	Cisco IOS XE Release 3.17S	The Spoke-to-Spoke Next Hop Resolution Protocol (NHRP) Summary Maps feature summarizes and reduces the NHRP resolution traffic on the network. The following commands were introduced or modified by this feature: **ip nhrp summary-map, ipv6 summary-map**.