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:

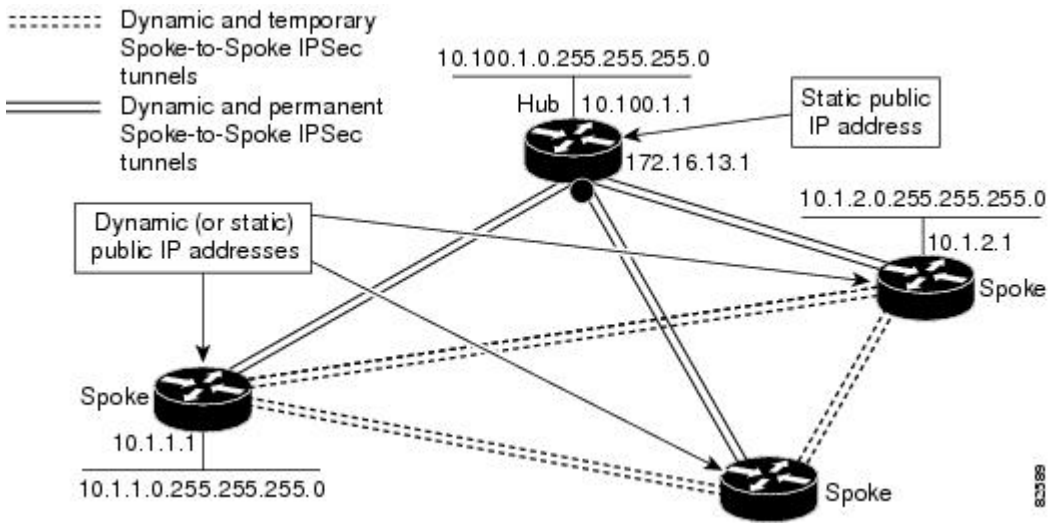
- NHRP--A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its

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

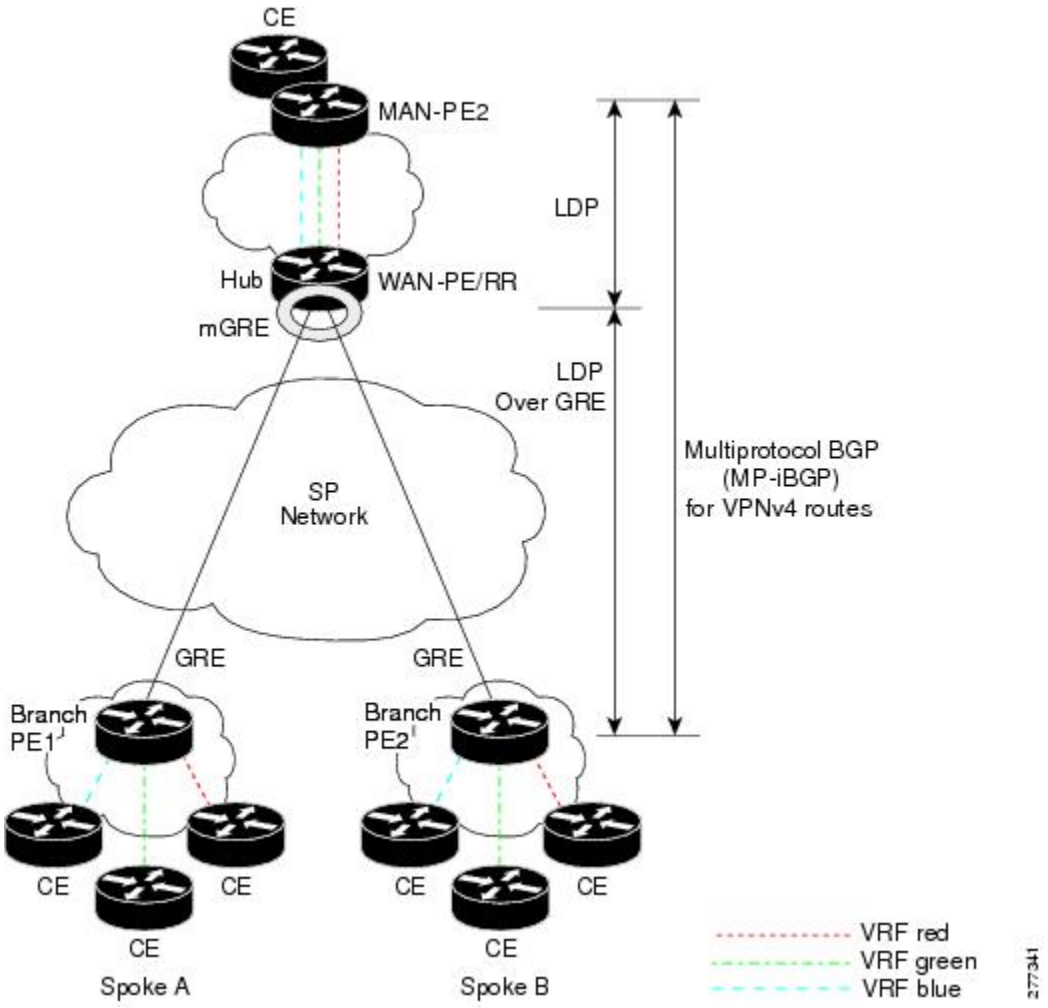
an entire crypto map configuration. An IPsec profile contains only IPsec information; that is, it does not contain any access list information or peering information.

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.

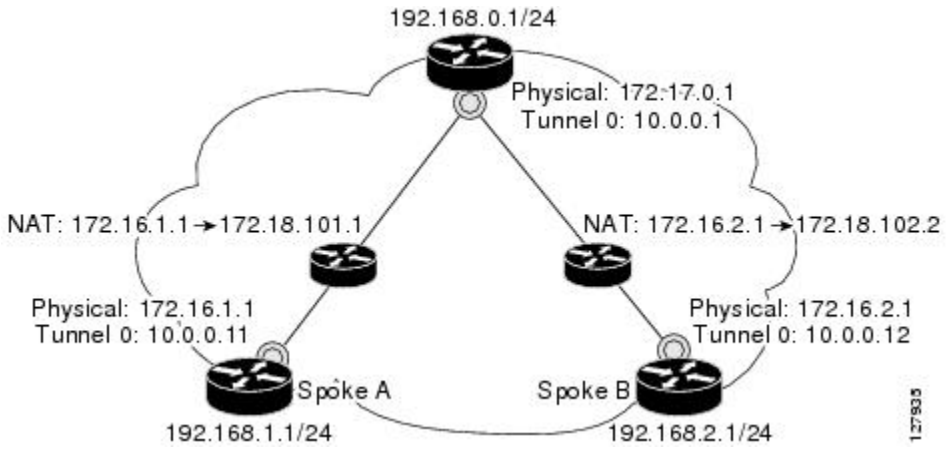
With this NAT Transparency enhancement, the hub DMVPN router can be behind the static NAT. For this functionality to be used, all the DMVPN spoke routers and hub routers must be upgraded, and IPsec must use transport mode.

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.