Configure ISP Redundancy on a DMVPN Spoke with the VRF−Lite Feature

TAC **Document ID: 119022**

Contributed by Adesh Gairola, Rudresh V, and Atri Basu, Cisco TAC Engineers.
Jun 22, 2015

Contents

Introduction

This document describes how to configure Internet Service Provider (ISP) redundancy on a Dynamic Multipoint VPN (DMVPN) spoke via the Virtual Routing and Forwarding−Lite (VRF−Lite) feature.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics before you attempt the configuration that is described in this document:

- Basic knowledge of VRF
- Basic knowledge of Enhanced Interior Gateway Routing Protocol (EIGRP)
- Basic knowledge of DMVPN

Components Used

The information in this document is based on Cisco IOS® Version 15.4(2)T.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

The VRF is a technology included in the IP network routers that allows multiple instances of a routing table to coexist in a router and work simultaneously. This increases functionality because it allows the network paths to be segmented without the use of multiple devices.

The use of dual ISPs for redundancy has become a common practice. Administrators use two ISP links; one acts as a primary connection and the other acts as a backup connection.

The same concept can be implemented for DMVPN redundancy on a spoke with the use of dual ISPs. The objective of this document is to demonstrate how *VRF−Lite* can be used in order to segregate the routing table when a spoke has dual ISPs. Dynamic routing is used in order to provide path redundancy for the traffic that traverses the DMVPN tunnel. The configuration examples that are described in this document use this configuration schema:

Interface	IP Address	VRF	Description
Ethernet0/0	172.16.1.1	ISP1 VRF	Primary ISP
Ethernet0/1	172.16.2.1	ISP2 VRF	Secondary ISP

With the VRF−Lite feature, multiple VPN routing/forwarding instances can be supported on the DMVPN spoke. The VRF−Lite feature forces the traffic from multiple Multipoint Generic Routing Encapsulation (mGRE) tunnel interfaces to use their respective VRF routing tables. For example, if the primary ISP terminates in the *ISP1* VRF and the secondary ISP terminates in the *ISP2* VRF, the traffic that is generated in the *ISP2* VRF uses the *ISP2* VRF routing table, while the traffic that is generated in the *ISP1* VRF uses the *ISP1* VRF routing table.

An advantage that comes with the use of a *front door* VRF (fVRF) is primarily to carve out a separate routing table from the global routing table (where tunnel interfaces exist). The advantage with the use of an *inside* VRF (iVRF) is to define a private space in order to hold the DMVPN and private network information. Both of these configurations provide extra security from attacks on the router from the Internet, where the routing information is separated.

These VRF configurations can be used on both the DMVPN hub and spoke. This gives great advantage over a scenario in which both of the ISPs terminate in the global routing table.

If both of the ISPs terminate in the global VRF, they share the same routing table and both of the mGRE interfaces rely on the global routing information. In this case, if the primary ISP fails, the primary ISP interface might not go down if the failure point is in the backbone network of ISPs and not directly connected. This results in a scenario where both of the mGRE tunnel interfaces still use the default route that points to the primary ISP, which causes the DMVPN redundancy to fail.

Though there are some workarounds that use IP Service Level Agreements (IP SLA) or Embedded Event Manager (EEM) scripts in order to address this issue without VRF−Lite, they might not always be the best choice.

Deployment Methods

This section provides brief overviews of split tunneling and spoke−to−spoke tunnels.

Split Tunneling

When specific subnets or summarized routes are learned via an mGRE interface, then it is called *split tunneling*. If the default route is learned via an mGRE interface, then it is called *tunnel−all*.

The configuration example that is provided in this document is based on split tunneling.

Spoke−to−Spoke Tunnels

The configuration example that is provided in this document is a good design for the tunnel−all deployment method (the default route is learned via the mGRE interface).

The use of two fVRFs segregates the routing tables and ensures that the post−GRE encapsulated packets are forwarded to the respective fVRF, which helps to ensure that the spoke−to−spoke tunnel comes up with an active ISP.

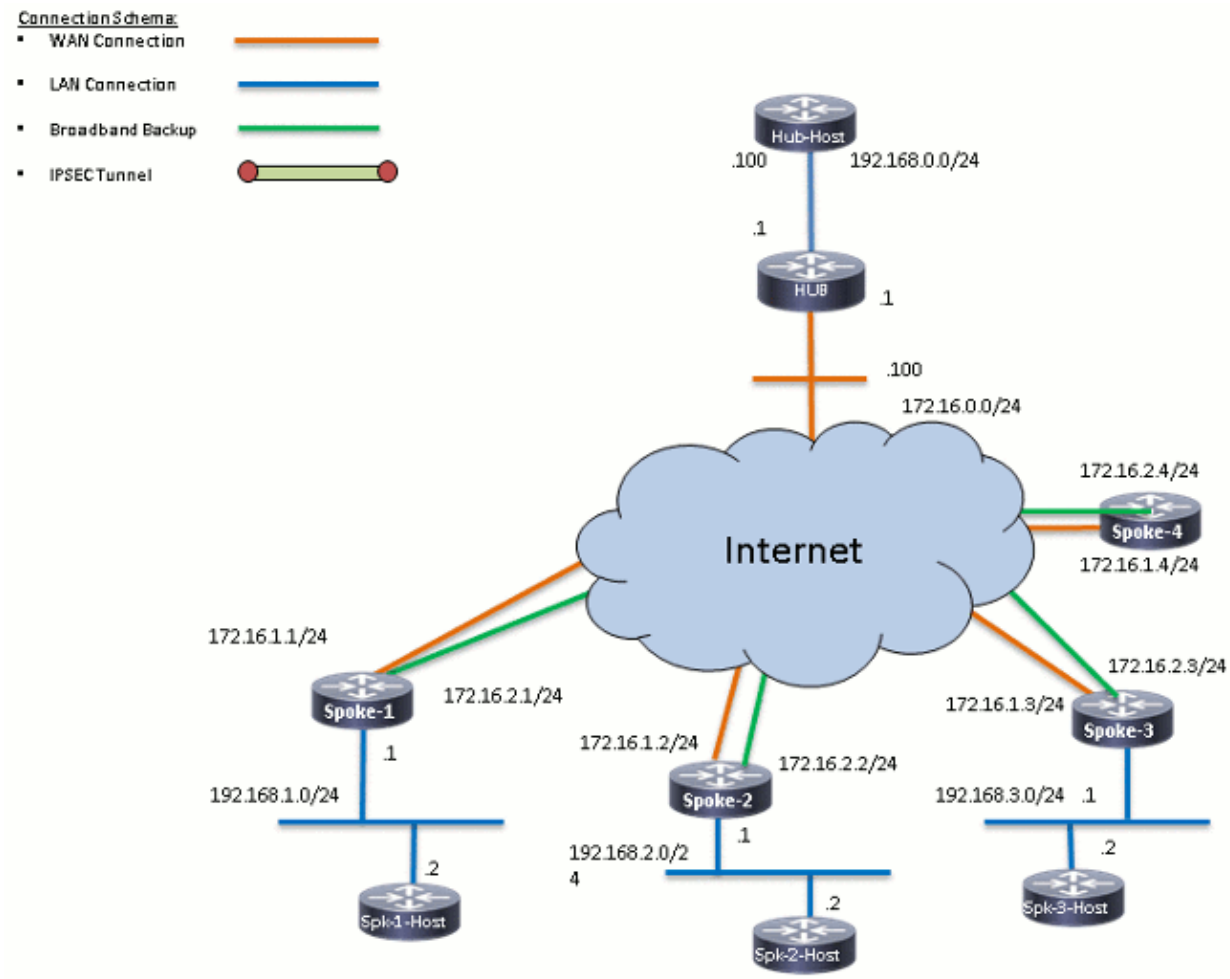
Configure

This section describes how to configure ISP redundancy on a DMVPN spoke via the VRF−Lite feature.

Note: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this section.

Network Diagram

This is the topology that is used for the examples within this document:



Hub Configuration

Here are some notes about the relevant configuration on the hub:

- In order to set *Tunnel0* as the primary interface in this configuration example, the *delay* parameter has been changed, which allows the routes that are learned from *Tunnel0* to become more preferred.
- The ***shared*** keyword is used with tunnel protection and a unique *tunnel key* is added on all of the mGRE interfaces because they use the same *tunnel source <interface>*. Otherwise, the inbound Generic Routing Encapsulation (GRE) tunnel packets might be punted to the incorrect tunnel interface after decryption.
- A route summarization is performed in order to ensure that all of the spokes learn the default route via the mGRE tunnels (***tunnel−all***).

Note: Only the relevant sections of the configuration are included in this example.

```
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname HUB1
!
crypto isakmp policy 1
 encr aes 256
 hash sha256
```

```
 authentication pre-share
 group 24
crypto isakmp key cisco123 address 0.0.0.0
!
crypto ipsec transform-set transform-dmvpn esp-aes 256 esp-sha256-hmac
 mode transport
!
crypto ipsec profile profile-dmvpn
 set transform-set transform-dmvpn
!
interface Loopback0
 description LAN
 ip address 192.168.0.1 255.255.255.0
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 no ip split-horizon eigrp 1
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp holdtime 600
 ip nhrp redirect
 ip summary-address eigrp 1 0.0.0.0 0.0.0.0
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile profile-dmvpn shared
!
interface Tunnel1
 bandwidth 1000
 ip address 10.0.1.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 no ip split-horizon eigrp 1
 ip nhrp map multicast dynamic
 ip nhrp network-id 100001
 ip nhrp holdtime 600
 ip nhrp redirect
 ip summary-address eigrp 1 0.0.0.0 0.0.0.0
 ip tcp adjust-mss 1360
 delay 1500
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel key 100001
 tunnel protection ipsec profile profile-dmvpn shared
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 10.0.1.0 0.0.0.255
 network 192.168.0.0 0.0.255.255
!
ip route 0.0.0.0 0.0.0.0 172.16.0.100
!
end
```

Spoke Configuration

Here are some notes about the relevant configuration on the spoke:

- For spoke redundancy, *Tunnel0* and *Tunnel1* have *Ethernet0/0* and *Ethernet0/1* as the tunnel source

interfaces, respectively. Ethernet0/0 is connected to the primary ISP and Ethernet0/1 is connected to secondary ISP.

- In order to segregate the ISPs, the VRF feature is used. The primary ISP uses the *ISP1* VRF. For the secondary ISP, a VRF named *ISP2* is configured.

- The *tunnel vrf ISP1* and *tunnel vrf ISP2* are configured on interfaces Tunnel0 and Tunnel1, respectively, in order to indicate that the forwarding lookup for the post−GRE encapsulated packet is performed in either VRF ISP1 or ISP2.

- In order to set Tunnel0 as the primary interface in this configuration example, the *delay* parameter has been changed, which allows the routes that are learned from Tunnel0 to become more preferred.

Note: Only the relevant sections of the configuration are included in this example.

```
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password−encryption
!
hostname SPOKE1
!
vrf definition ISP1
 rd 1:1
 !
 address−family ipv4
 exit−address−family
!
vrf definition ISP2
 rd 2:2
 !
 address−family ipv4
 exit−address−family
!
crypto keyring ISP2 vrf ISP2
  pre−shared−key address 0.0.0.0 0.0.0.0 key cisco123
crypto keyring ISP1 vrf ISP1
  pre−shared−key address 0.0.0.0 0.0.0.0 key cisco123
!
crypto isakmp policy 1
 encr aes 256
 hash sha256
 authentication pre−share
 group 24
crypto isakmp keepalive 10 periodic
!
crypto ipsec transform−set transform−dmvpn esp−aes 256 esp−sha256−hmac
 mode transport
!
!
crypto ipsec profile profile−dmvpn
 set transform−set transform−dmvpn
!
interface Loopback10
 ip address 192.168.1.1 255.255.255.0
!
interface Tunnel0
 description Primary mGRE interface source as Primary ISP
 bandwidth 1000
 ip address 10.0.0.10 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp network−id 100000
```

```
 ip nhrp holdtime 600
 ip nhrp nhs 10.0.0.1 nbma 172.16.0.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 delay 1000
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel vrf ISP1
 tunnel protection ipsec profile profile-dmvpn
!
interface Tunnel1
 description Secondary mGRE interface source as Secondary ISP
 bandwidth 1000
 ip address 10.0.1.10 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp network-id 100001
 ip nhrp holdtime 360
 ip nhrp nhs 10.0.1.1 nbma 172.16.0.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 delay 1500
 tunnel source Ethernet0/1
 tunnel mode gre multipoint
 tunnel key 100001
 tunnel vrf ISP2
 tunnel protection ipsec profile profile-dmvpn
!
interface Ethernet0/0
 description Primary ISP
 vrf forwarding ISP1
 ip address 172.16.1.1 255.255.255.0
!
interface Ethernet0/1
 description Seconday ISP
 vrf forwarding ISP2
 ip address 172.16.2.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 10.0.1.0 0.0.0.255
 network 192.168.0.0 0.0.255.255
!
ip route vrf ISP1 0.0.0.0 0.0.0.0 172.16.1.254
ip route vrf ISP2 0.0.0.0 0.0.0.0 172.16.2.254
!
logging dmvpn
!
end
```

Verify

Use the information that is described in this section in order to verify that your configuration works properly.

Primary and Secondary ISPs Active

In this verification scenario, both the primary and secondary ISPs are active. Here are some additional notes about this scenario:

- Phase 1 and phase 2 for both of the mGRE interfaces are up.

- Both of the tunnels come up, but the routes via Tunnel0 (sourced via the primary ISP) are preferred.

Here are the relevant ***show*** commands that you can use in order to verify your configuration in this scenario:

```
SPOKE1#show ip route
<snip>
Gateway of last resort is 10.0.0.1 to network 0.0.0.0

D*    0.0.0.0/0 [90/2944000] via 10.0.0.1, 1w0d, Tunnel0

!--- This is the default route for all of the spoke and hub LAN segments.

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.10/32 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.10/32 is directly connected, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback10
L        192.168.1.1/32 is directly connected, Loopback10

SPOKE1#show ip route vrf ISP1

Routing Table: ISP1
<snip>

Gateway of last resort is 172.16.1.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.1.254
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/24 is directly connected, Ethernet0/0
L        172.16.1.1/32 is directly connected, Ethernet0/0

SPOKE1#show ip route vrf ISP2

Routing Table: ISP2
<snip>

Gateway of last resort is 172.16.2.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.2.254
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.2.0/24 is directly connected, Ethernet0/1
L        172.16.2.1/32 is directly connected, Ethernet0/1

SPOKE1#show crypto session
Crypto session current status

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.1.1/500 remote 172.16.0.1/500 Active

!--- Tunnel0 is Active and the routes are preferred via Tunnel0.

  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.0.1
        Active SAs: 2, origin: crypto map

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.2.1/500 remote 172.16.0.1/500 Active
```

```
!--- Tunnel0 is Active and the routes are preferred via Tunnel0.

  IPSEC FLOW: permit 47 host 172.16.2.1 host 172.16.0.1
        Active SAs: 2, origin: crypto map
```

Primary ISP Down/Secondary ISP Active

In this scenario, the EIGRP *Hold* timers expire for the neighborship through Tunnel0 when the ISP1 link goes down, and the routes to the hub and the other spokes now point to Tunnel1 (sourced with Ethernet0/1).

Here are the relevant ***show*** commands that you can use in order to verify your configuration in this scenario:

```
*Sep  2 14:07:33.374: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.1 (Tunnel0)
 is down: holding time expired

SPOKE1#show ip route
<snip>

Gateway of last resort is 10.0.1.1 to network 0.0.0.0

D*    0.0.0.0/0 [90/3072000] via 10.0.1.1, 00:00:20, Tunnel1

!--- This is the default route for all of the spoke and hub LAN segments.

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.10/32 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.10/32 is directly connected, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback10
L        192.168.1.1/32 is directly connected, Loopback10

SPOKE1#show ip route vrf ISP1

Routing Table: ISP1
<snip>

Gateway of last resort is 172.16.1.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.1.254
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/24 is directly connected, Ethernet0/0
L        172.16.1.1/32 is directly connected, Ethernet0/0

SPOKE1#show ip route vrf ISP2

Routing Table: ISP2
<snip>

Gateway of last resort is 172.16.2.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.2.254
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.2.0/24 is directly connected, Ethernet0/1
L        172.16.2.1/32 is directly connected, Ethernet0/1

SPOKE1#show crypto session
Crypto session current status

Interface: Tunnel0
Session status: DOWN
Peer: 172.16.0.1 port 500
  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.0.1
```

```
!--- Tunnel0 is Inactive and the routes are preferred via Tunnel1.

        Active SAs: 0, origin: crypto map

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.2.1/500 remote 172.16.0.1/500 Active

!--- Tunnel0 is Inactive and the routes are preferred via Tunnel1.

  IPSEC FLOW: permit 47 host 172.16.2.1 host 172.16.0.1
        Active SAs: 2, origin: crypto map

Interface: Tunnel0
Session status: DOWN-NEGOTIATING
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.1.1/500 remote 172.16.0.1/500 Inactive

!--- Tunnel0 is Inactive and the routes are preferred via Tunnel1.

  Session ID: 0
  IKEv1 SA: local 172.16.1.1/500 remote 172.16.0.1/500 Inactive
```

Primary ISP Link Restoration

When the connectivity through the primary ISP is restored, the Tunnel0 crypto session becomes active, and the routes that are learned via the Tunnel0 interface are preferred.

Here is an example:

```
*Sep  2 14:15:59.128: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.1 (Tunnel0)
 is up: new adjacency

SPOKE1#show ip route
<snip>

Gateway of last resort is 10.0.0.1 to network 0.0.0.0

D*    0.0.0.0/0 [90/2944000] via 10.0.0.1, 00:00:45, Tunnel0

!--- This is the default route for all of the spoke and hub LAN segments.

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.0.0/24 is directly connected, Tunnel0
L        10.0.0.10/32 is directly connected, Tunnel0
C        10.0.1.0/24 is directly connected, Tunnel1
L        10.0.1.10/32 is directly connected, Tunnel1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback10
L        192.168.1.1/32 is directly connected, Loopback10

SPOKE1#show crypto session
Crypto session current status

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.1.1/500 remote 172.16.0.1/500 Active
```

```
!--- Tunnel0 is Active and the routes are preferred via Tunnel0.

  IPSEC FLOW: permit 47 host 172.16.1.1 host 172.16.0.1
        Active SAs: 2, origin: crypto map

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  Session ID: 0
  IKEv1 SA: local 172.16.2.1/500 remote 172.16.0.1/500 Active

!--- Tunnel0 is Active and the routes are preferred via Tunnel0.

  IPSEC FLOW: permit 47 host 172.16.2.1 host 172.16.0.1
        Active SAs: 2, origin: crypto map
```

Troubleshoot

In order to troubleshoot your configuration, enable ***debug ip eigrp*** and ***logging dmvpn***.

Here is an example:

```
################### Tunnel0 Failed and Tunnel1 routes installed ####################

*Sep  2 14:07:33.374: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.1 (Tunnel0)
 is down: holding time expired
*Sep  2 14:07:33.374: EIGRP-IPv4(1): table(default): route installed for 0.0.0.0/0
 (90/3072000) origin(10.0.1.1)
*Sep  2 14:07:33.391: EIGRP-IPv4(1): table(default): 0.0.0.0/0 - do advertise
 out Tunnel1
*Sep  2 14:07:33.399: EIGRP-IPv4(1): table(default): 0.0.0.0/0 - do advertise
 out Tunnel1
*Sep  2 14:07:36.686: %DMVPN-5-CRYPTO_SS:  Tunnel0: local address : 172.16.1.1 remote
 address : 172.16.0.1 socket is DOWN
*Sep  2 14:07:36.686: %DMVPN-5-NHRP_NHS_DOWN: Tunnel0: Next Hop Server : (Tunnel:
 10.0.0.1 NBMA: 172.16.0.1 ) for (Tunnel: 10.0.0.10 NBMA: 172.16.1.1) is DOWN, Reason:
 External(NHRP: no error)

################### Tunnel0 came up and routes via Tunnel0 installed #################

*Sep  2 14:15:55.120: %DMVPN-5-CRYPTO_SS:  Tunnel0: local address : 172.16.1.1 remote
 address : 172.16.0.1 socket is UP
*Sep  2 14:15:56.109: %DMVPN-5-NHRP_NHS_UP: Tunnel0: Next Hop Server : (Tunnel:
 10.0.0.1 NBMA: 172.16.0.1) for (Tunnel: 10.0.0.10 NBMA: 172.16.1.1) is  UP
*Sep  2 14:15:59.128: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.0.0.1 (Tunnel0)
 is up: new adjacency
*Sep  2 14:16:01.197: EIGRP-IPv4(1): table(default): route installed for 0.0.0.0/0
 (90/3072000) origin(10.0.1.1)
*Sep  2 14:16:01.197: EIGRP-IPv4(1): table(default): route installed for 0.0.0.0/0
 (90/2944000) origin(10.0.0.1)
*Sep  2 14:16:01.214: EIGRP-IPv4(1): table(default): 0.0.0.0/0 - do advertise
 out Tunnel0
*Sep  2 14:16:01.214: EIGRP-IPv4(1): table(default): 0.0.0.0/0 - do advertise
 out Tunnel1
```

Related Information

- ***Most Common DMVPN Troubleshooting Solutions***
- ***Cisco MDS 9000 Family Troubleshooting Guide, Release 2.x Troubleshooting IPsec***
- ***Technical Support & Documentation - Cisco Systems***