DMVPN

CE2 M19

Agenda

- This chapter covers the following content:
  - Generic Routing Encapsulation (GRE) Tunnels - This section explains how GRE tunnels operate and explains the configuration of GRE tunnels.
  - Next Hop Resolution Protocol (NHRP) - This section describes the NHRP protocol and how it dynamically maps underlay IP addresses to overlay tunnel IP addresses.
  - Dynamic Multipoint VPN (DMVPN) - This section explains the three DMVPN phases and the technologies involved with DMVPN tunnels.
  - DMVPN Configuration - This section explains the configuration of DMVPN tunnels.
  - Spoke-to-Spoke Communication - This section explains how spoke-to-spoke DMVPN tunnels form.

Generic Routing Encapsulation (GRE) Tunnels

GRE

- A GRE tunnel provides connectivity to a wide variety of network layer protocols by encapsulating and forwarding those packets over an IP-based network.
  - The original use of GRE tunnels was to provide a transport mechanism for nonroutable legacy protocols such as DECnet, Systems Network Architecture (SNA), and IPX.
- DMVPN uses Multipoint GRE (mGRE) encapsulation and supports dynamic routing protocols, which eliminates many of the support issues associated with other VPN technologies.
- GRE tunnels are classified as an overlay network because a GRE tunnel is built on top of an existing transport network, also known as an underlay network.

GRE Tunnel Configuration

- Figure 19-1 illustrates the configuration of a GRE tunnel. The 172.16.0.0/16 network range is the transport (underlay) network, and 192.168.100.0/24 is used for the GRE tunnel (overlay network).

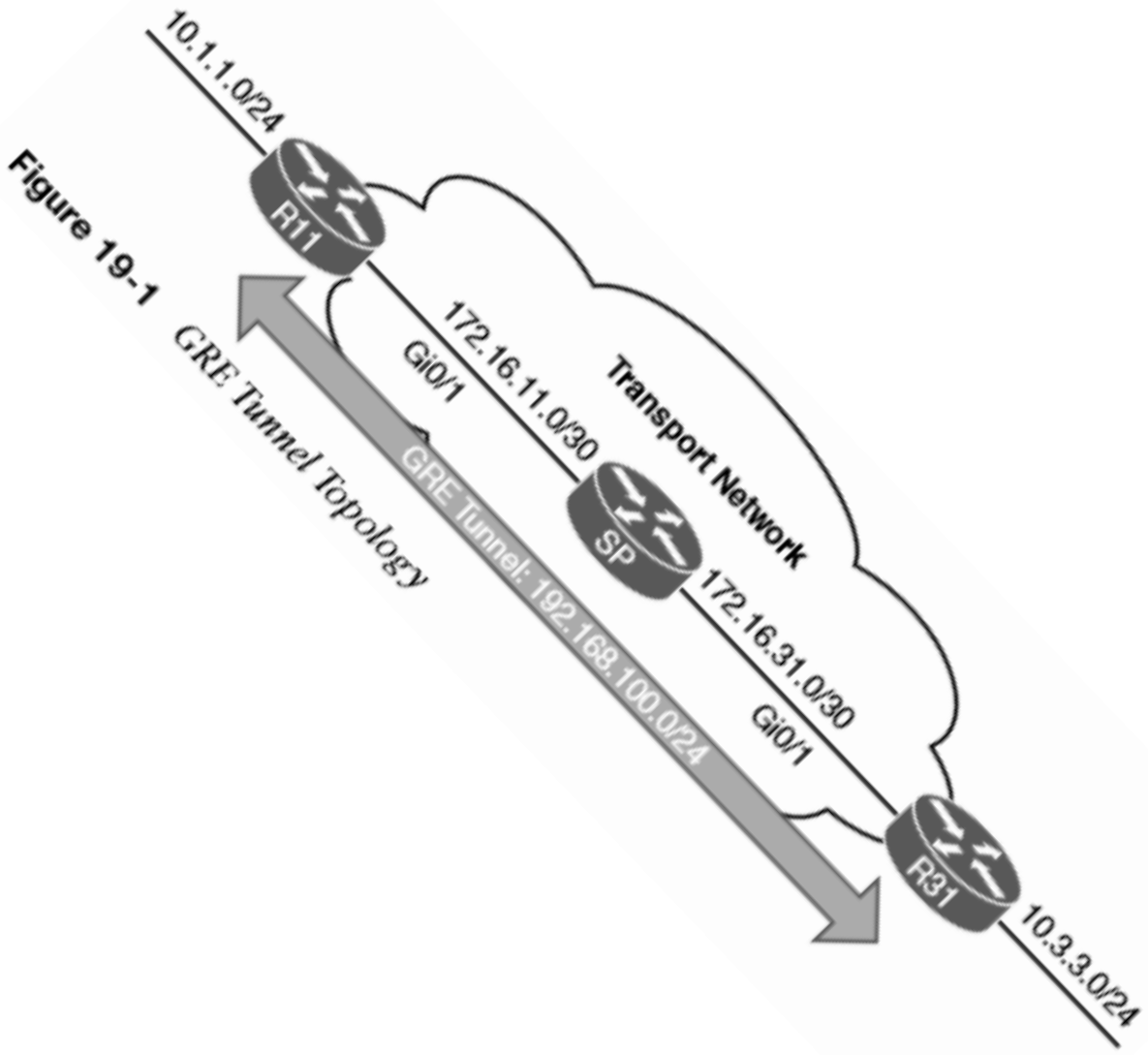


Figure 19-1

GRE Tunnel Configuration (Cont.)

- Example 19-1 shows the routing table of R11 before the GRE tunnel is created. Notice that the 10.3.3.0/24 network is reachable by RIP and is two hops away.

Example 19-1 *R11 Routing Table Without the GRE Tunnel*

```
R11# show ip route
! Output omitted for brevity
Codes: L - local,    C - connected, S - static, R - RIP,    M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/2
R       10.3.3.0/24 [120/2] via 172.16.11.2, 00:00:01, GigabitEthernet0/1
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.11.0/30 is directly connected, GigabitEthernet0/1
R       172.16.31.0/30 [120/1] via 172.16.11.2, 00:00:10, GigabitEthernet0/1
```

```
R11# trace 10.3.3.3 source 10.1.1.1
Tracing the route to 10.3.3.3
   1 172.16.11.2 0 msec 0 msec 1 msec
   2 172.16.31.3 0 msec
```

GRE Tunnel Configuration (Cont.)

- The steps for configuring GRE tunnels are as follows:

1) **Step 1.** Create the tunnel interface by using the global configuration command **interface tunnel** *tunnel-number*.
2) **Step 2.** Identify the local source of the tunnel by using the interface parameter **command tunnel source** *{ip-address* | *interface-id}.*
3) **Step 3.** Identify the tunnel destination by using the interface parameter command **tunnel destination** *ip-address***.**
4) **Step 4.** Allocate an IP address to the tunnel interface by using the command **ip address** *ip-address subnet-mask***.**
5) **Step 5.** Optionally define the tunnel bandwidth, measured in kilobits per second, by using the interface parameter command **bandwidth** [*1-10000000*].
6) **Step 6.** Optionally specify a GRE tunnel keepalive by using the interface parameter command **keepalive** [*seconds* [*retries*]].
7) **Step 7.** Optionally define the IP maximum transmission unit (MTU) for the tunnel interface by using the interface parameter command **ip mtu** *mtu*.

Generic Routing Encapsulation (GRE) Tunnels

GRE Sample Configuration

- Example 19-2 provides the GRE tunnel configuration for R11 and R31.
- EIGRP is enabled on the LAN (10.0.0.0/8) and GRE tunnel (192.168.100.0/24) networks.
- RIP is enabled on the LAN (10.0.0.0/8) and transport (172.16.0.0/16) networks but is not enabled on the GRE tunnel.
- R11 and R31 become direct EIGRP peers on the GRE tunnel because all the network traffic is encapsulated between them.

Example 19-2 *GRE Configuration*

```
R11
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.31.1
!
router eigrp GRE-OVERLAY
 address-family ipv4 unicast autonomous-system 100
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
 exit-address-family
!
router rip
 version 2
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
```

```
R31
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
!
router eigrp GRE-OVERLAY
 address-family ipv4 unicast autonomous-system 100
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
 exit-address-family
!
router rip
 version 2
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
```

GRE Sample Configuration (Cont.)

- When the GRE tunnel is configured, the state of the tunnel can be verified with the command **show interface tunnel** *number*. Example 19-3 displays output from this command.

Example 19-3 *Display of GRE Tunnel Parameters*

```
R11# show interface tunnel 100
! Output omitted for brevity
Tunnel100 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 192.168.100.1/24
  MTU 17916 bytes, BW 400 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation TUNNEL, loopback not set
 Keepalive set (5 sec), retries 3
 Tunnel source 172.16.11.1 (GigabitEthernet0/1), destination 172.16.31.1
 Tunnel Subblocks:
    src-track:
       Tunnel100 source tracking subblock associated with GigabitEthernet0/1
      Set of tunnels with source GigabitEthernet0/1, 1 member (includes
      iterators), on interface <OK>
 Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
 Tunnel TTL 255, Fast tunneling enabled
 Tunnel transport MTU 1476 bytes
 Tunnel transmit bandwidth 8000 (kbps)
 Tunnel receive bandwidth 8000 (kbps)
 Last input 00:00:02, output 00:00:02, output hang never
```

Next Hop Resolution Protocol (NHRP)

NHRP

- **Next Hop Resolution Protocol (NHRP)** is defined in RFC2332 as a method to provide address resolution for hosts or networks (with ARP-like capability) for nonbroadcast multi-access (NBMA) networks such as Frame Relay and ATM networks.
- NHRP is a client/server protocol that allows devices to register themselves over directly connected or disparate networks.
- DMVPN uses mGRE tunnels and therefore requires a method of mapping tunnel IP addresses to the transport (underlay) IP address. NHRP provides the technology for mapping those IP addresses.

Next Hop Resolution Protocol
NHRP Message Types

- All NHRP packets must include the source NBMA address, source protocol address, destination protocol address, and NHRP message type. The NHRP message types are explained in Table 19-3.

Type	Description
Registration	Registration messages are sent by the **NHC (DMVPN spoke)** toward the **NHS (DMVPN hub)**. Registration allows the hubs to know a spoke's NBMA information.
Resolution	Resolution messages are NHRP messages designed to locate and provide the address resolution information of the egress router toward the destination.
Redirect	Redirect messages are an essential component of DMVPN Phase 3. They allow an intermediate router to notify the encapsulator (a router) that a specific network can be reached by using a more optimal path (spoke-to-spoke tunnel).
Purge	Purge messages are sent to remove a cached NHRP entry. Purge messages notify routers of the loss of a route used by NHRP. A purge is typically sent by an NHS to an NHC (which it answered) to indicate that the mapping for an address/network that it answered is not valid anymore.
Error	Error messages are used to notify the sender of an NHRP packet that an error has occurred.

NHRP Message Extensions

- NHRP messages can contain additional information that is included in the extension part of a message. Table 19-4 lists the common NHRP message extensions.

Extension	Description
Responder address	This is used to determine the address of the responding node for reply messages.
Forward transit NHS record	This contains a list of NHSs that the NHRP request packet may have traversed.
Reverse transit NHS record	This contains a list of NHSs that the NHRP reply packet may have traversed.
Authentication	This conveys authentication information between NHRP speakers. Authentication is done pairwise on a hop-by-hop basis. This field is transmitted in plaintext.
Vendor private	This conveys vendor private information between NHRP speakers.
NAT	DMVPN works when a hub or spoke resides behind a device that performs NAT and when the tunnel is encapsulated in IPsec.

Dynamic Multipoint VPN (DMVPN)

DMVPN

- DMVPN provides complete connectivity while simplifying configuration as new sites are deployed.
  - It is considered a zero-touch technology because no configuration is needed on the DMVPN hub routers as new spokes are added to the DMVPN network.
- A spoke site initiates a persistent VPN connection to the hub router.
- Network traffic between spoke sites does not have to travel through the hubs. DMVPN dynamically builds a VPN tunnel between spoke sites on an as-needed basis.

DMVPN Benefits

- DMVPN provides the following benefits to network administrators:

- **Zero-touch provisioning** - DMVPN hubs do not require additional configuration when additional spokes are added.
- **Scalable deployment** - Minimal peering and minimal permanent state on spoke routers allow for massive scale.
- **Spoke-to-spoke tunnels** - DMVPN provides full-mesh connectivity while requiring configuration of only the initial spoke-to-hub tunnel.
- **Flexible network topologies** - DMVPN operation does not make any rigid assumptions about either the control plane or data plane overlay topologies.
- **Multiprotocol support** - DMVPN can use IPv4, IPv6, and MPLS as the overlay or transport network protocol.
- **Multicast support** - DMVPN allows multicast traffic to flow on the tunnel interfaces.
- **Adaptable connectivity** - DMVPN routers can establish connectivity behind NAT.
- **Standardized building blocks** - DMVPN uses industry-standardized technologies (NHRP, GRE, and IPsec) to build an overlay network.

DMVPN Phases 1, 2, and 3

- DMVPN was released in three phases, each phase built on the previous one with additional functions. All three phases of DMVPN need only one tunnel interface on a router, and the DMVPN network size should accommodate all the endpoints associated with that tunnel network.

1) **Phase 1: Spoke-to-Hub** – Phase 1 provides a zero-touch deployment for VPN sites. VPN tunnels are created only between spoke and hub sites. Traffic between spokes must traverse the hub to reach any other spoke.

2) **Phase 2: Spoke-to-Spoke** – Phase 2 provides additional capability beyond DMVPN Phase 1 and allows spoke-to-spoke communication on a dynamic basis by creating an on-demand VPN tunnel between the spoke devices.

3) **Phase 3: Hierarchical Tree Spoke-to-Spoke** - Phase 3 refines spoke-to-spoke connectivity by enhancing the NHRP messaging and interacting with the routing table. With DMVPN Phase 3, the hub sends an NHRP redirect message to the spoke that originated the packet flow. The NHRP redirect message provides the necessary information so that the originator spoke can initiate a resolution of the destination host/network.

Dynamic Multipoint VPN (DMVPN)

DMVPN Phase Comparison

- Figure 19-2 illustrates the differences in traffic patterns for the three DMVPN phases.
- All three models support direct spoke-to-hub communication, as shown by R1 and R2.
- Spoke-to-spoke packet flow in DMVPN Phase 1 is different from the packet flow in DMVPN Phases 2 and 3.
- Traffic between R3 and R4 must traverse the hub for Phase 1 DMVPN, whereas a dynamic spoke-to-spoke tunnel is created for DMVPN Phase 2 and Phase 3 that allows direct communication.

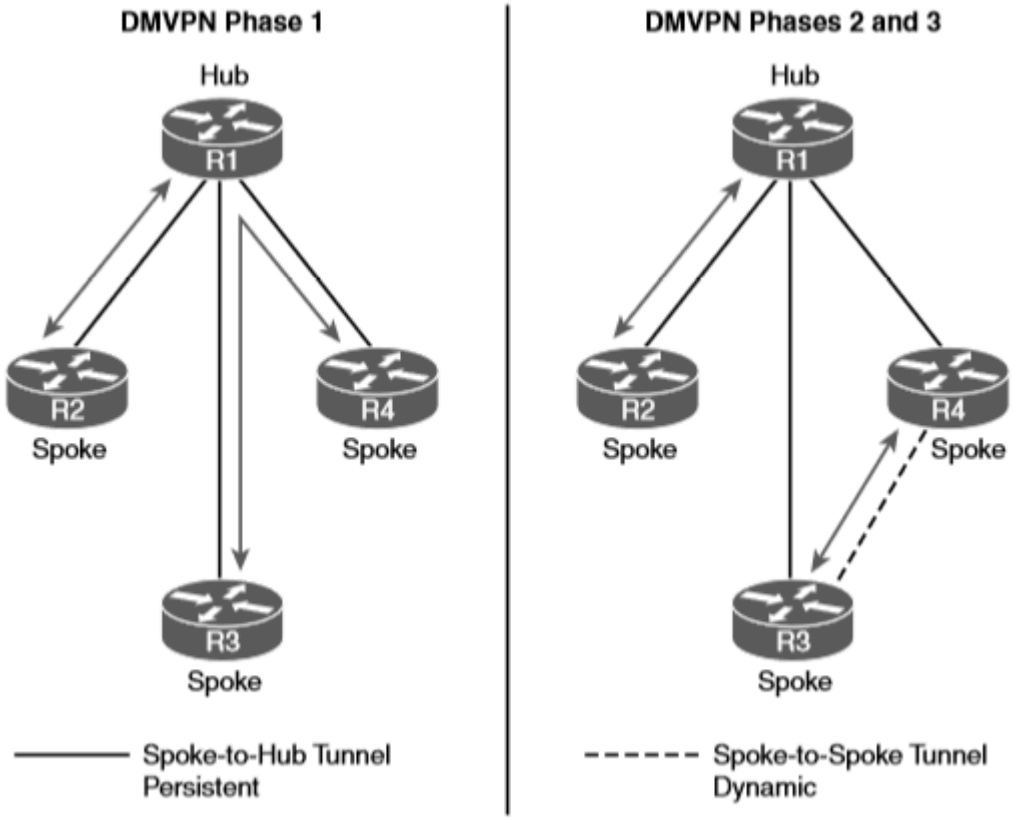


Figure 19-2 *DMVPN Traffic Patterns in the Different DMVPN Phases*

DMVPN Phase Comparison (Cont.)

- Figure 19-3 illustrates the difference in traffic patterns between Phase 2 and Phase 3 DMVPN with hierarchical topologies (multilevel).
- In this two-tier hierarchical design, R2 is the hub for DMVPN tunnel 20, and R3 is the hub for DMVPN tunnel 30.
- This chapter explains the DMVPN fundamentals with DMVPN Phase 1 and then explains DMVPN Phase 3.
  - It does not cover DMVPN Phase 2.

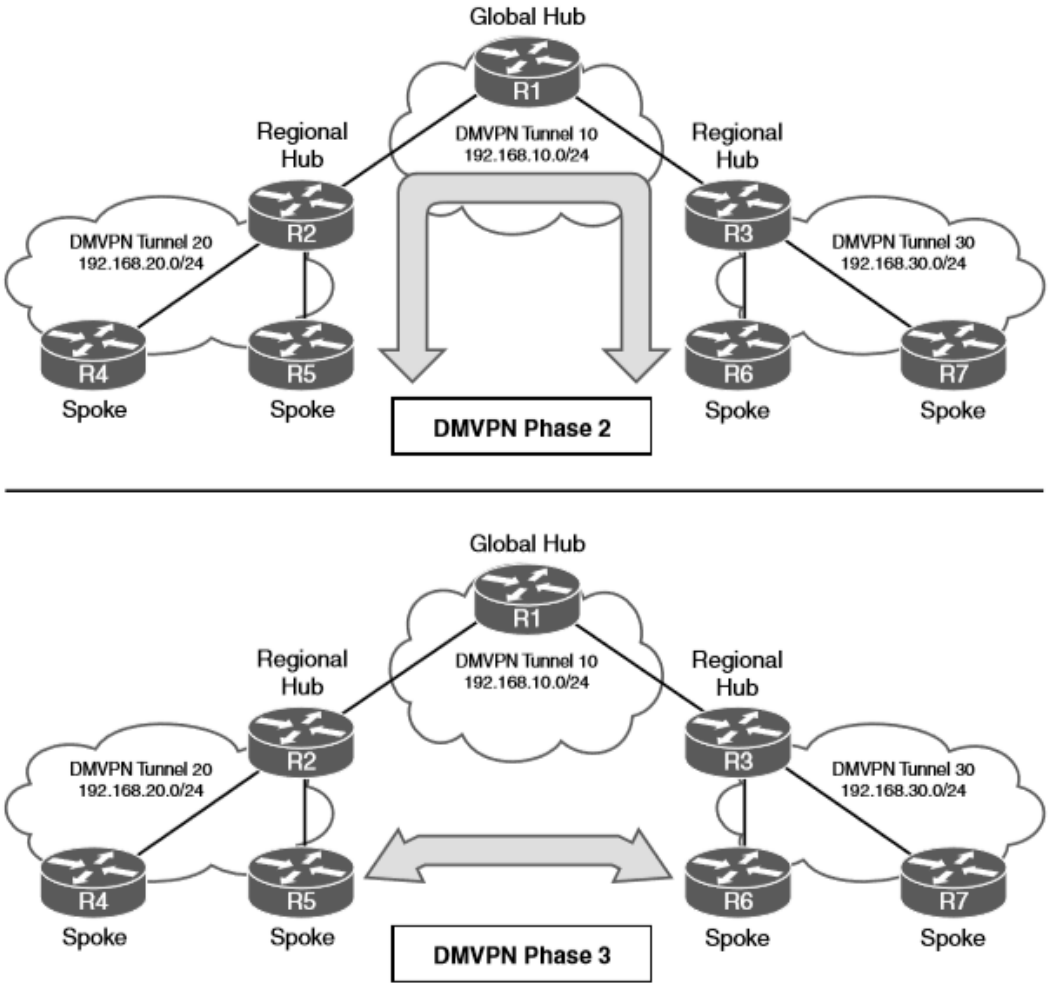


Figure 19-3 *Comparison of DMVPN Phase 2 and Phase 3*

DMVPN Configuration

DMVPN Configuration

- There are two types of DMVPN configurations, hub and spoke.
  - Each is used depending on a router's role. The DMVPN hub is the NHRP NHS, and the DMVPN spoke is the NHRP NHC.
- The spokes should be preconfigured with the hub's static IP address, but a spoke's NBMA IP address can be static or assigned from DHCP.
- The terms spoke router and branch router are used interchangeably, as are the terms hub router and headquarters/data center router.

DMVPN Configuration

Simple DMVPN Topology

- Figure 19-4 shows the first topology used to explain DMVPN configuration and functions.
- R11 acts as the DMVPN hub, and R31 and R41 are the DMVPN spokes.
- All three routers use a static default route to the SP router that provides connectivity for the NBMA (transport) networks in the 172.16.0.0/16 network range. EIGRP has been configured to operate on the DMVPN tunnel and to advertise the local LAN networks.

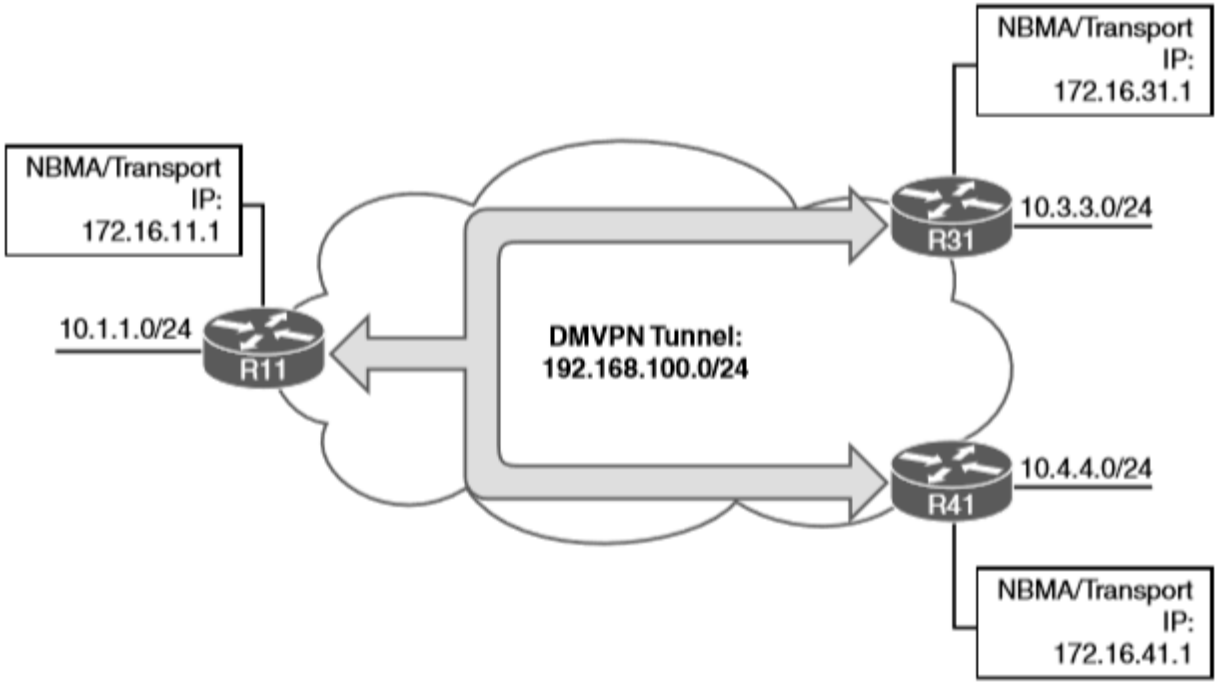


Figure 19-4 *Simple DMVPN Topology*

DMVPN Hub Configuration

- The steps for configuring DMVPN on a hub router are as follows:

1) **Step 1**. Create the tunnel interface by using the **interface tunnel** *tunnel-number* global configuration command.
2) **Step 2**. Identify the local source of the tunnel by using the **tunnel source** {*ip-address* | *interface-id*} interface parameter command.
3) **Step 3**. Configure the DMVPN tunnel as an mGRE tunnel by using the **tunnel mode gre multipoint** interface parameter command.
4) **Step 4**. Allocate an IP address for the DMVPN network (tunnel) by using the **ip address** *ip-address subnet-mask* command.
5) **Step 5**. Enable NHRP on the tunnel interface and uniquely identify the DMVPN tunnel for the virtual interface by using the **ip nhrp network-id** *1-4294967295* interface parameter command.
6) **Step 6**. Optionally define the tunnel key, which adds 4 bytes to the DMVPN header, with the **tunnel key** *0 - 4294967295* command.
7) **Step 7**. Optionally enable multicast support for NHRP on DMVPN hub routers by using the **ip nhrp map multicast dynamic** tunnel command.

DMVPN Hub Configuration (Cont.)

- The steps for configuring DMVPN on a hub router are as follows:

8) **Step 8**. For Phase 3, enable NHRP redirect functions by using the **ip nhrp redirect** command.
9) **Step 9**. Optionally define the tunnel bandwidth, measured in kilobits per second, by using the **bandwidth** [*1-10000000*] interface parameter command.
10) **Step 10**. Optionally configure the IP MTU for the tunnel interface by using the **ip mtu** *mtu* interface parameter command.
11) **Step 11**. Optionally define the TCP maximum segment size (MSS) by using the **ip tcp adjust-mss** *mss-size* command.

- **Note**: mGRE tunnels do not support the option for using a keepalive.

DMVPN Spoke Configuration for DMVPN Phase 1 (Point-to-Point

- Configuration of DMVPN Phase 1 spokes is similar to the configuration for a hub router except in two ways:
  - It does not use an mGRE tunnel. Instead, the tunnel destination is specified.
  - The NHRP mapping points to at least one active NHS.

- The process for configuring a DMVPN Phase 1 spoke router is as follows:

1) **Step 1**. Create the tunnel interface by using the **interface tunnel** *tunnel-number* global configuration command.
2) **Step 2**. Identify the local source of the tunnel by using the **tunnel source** {*ip-address* | *interface-id*} interface parameter command.
3) **Step 3**. Identify the tunnel destination by using the **tunnel destination** *ip-address* interface parameter command.
4) **Step 4**. Allocate an IP address for the DMVPN network (tunnel) by using the **ip address** {*ip-address subnet-mask* | **dhcp**} command or the **ipv6 address** *ipv6-address/prefix-length* command.
5) **Step 5**. Enable NHRP on the tunnel interface and uniquely identify the DMVPN tunnel for the virtual interface by using the **ip nhrp network-id** *1-4294967295* interface parameter command.

DMVPN Spoke Configuration for DMVPN Phase 1 (Point-to-Point) (Cont.)

- The process for configuring a DMVPN Phase 1 spoke router is as follows:

6) **Step 6**. Optionally define the NHRP tunnel key, which adds 4 bytes to the DMVPN header, by using the **tunnel key** *0-4294967295* command.
   - **Note**: If the tunnel key is defined on the hub router, it must be defined on all the spoke routers.
7) **Step 7**. Specify the address of one or more NHRP NHSs by using the **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**] command.
   - **Note**: Remember that the NBMA address is the transport IP address, and the NHS address is the protocol address for the DMVPN hub.
8) **Step 8**. Optionally define the tunnel bandwidth, measured in kilobits per second, by using the **bandwidth** [*1-10000000*] interface parameter command.
9) **Step 9**. Optionally define the IP MTU for the tunnel interface by using the **ip mtu** *mtu* interface parameter command.
10) **Step 10**. Optionally define the TCP MSS by using the **ip tcp adjust-mss** *mss-size* command.

DMVPN Spoke Configuration for DMVPN Phase 1 (Point-to-Point) (Cont.)

- Example 19-6 provides a sample configuration for R11 (hub), R31 (spoke), and R41 (spoke).

Example 19-6 *Phase 1 DMVPN Configuration*

```
R11-Hub
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R31-Spoke (Single Command NHRP Configuration)
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
 tunnel key 100
```

```
R41-Spoke (Multi-Command NHRP Configuration)
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.41 255.255.255.0
 ip mtu 1400
 ip nhrp map 192.168.100.11 172.16.11.1
 ip nhrp map multicast 172.16.11.1
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
 tunnel key 100
```

Viewing DMVPN Tunnel Status

- The **show dmvpn** [**detail**] command provides the tunnel interface, tunnel role, tunnel state, and tunnel peers with uptime. When the DMVPN tunnel interface is administratively shut down, there are no entries associated with that tunnel interface. These are the tunnel states, in order of establishment:
  - **INTF** – The line protocol of the DMVPN tunnel is down.
  - **IKE** – DMVPN tunnels configured with IPsec have not yet successfully established an Internet Key Exchange (IKE) session.
  - **IPsec** - An IKE session has been established, but an IPsec security association (SA) has not yet been established.
  - **NHRP** - The DMVPN spoke router has not yet successfully registered.
  - **Up** - The DMVPN spoke router has registered with the DMVPN hub and received an ACK (positive registration reply) from the hub.

Viewing DMVPN Tunnel Status (Cont.)

- Example 19-8 provides output of the **show dmvpn detail** command. The **detail** keyword provides the local tunnel and NBMA IP addresses, tunnel health monitoring, and VRF contexts.

Example 19-8 *Viewing the DMVPN Tunnel Status for Phase 1 DMVPN*

```
R11-Hub# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
         UpDn Time --> Up or Down Time for a Tunnel
==========

Interface Tunnel100 is up/up, Addr. is 192.168.100.11, VRF ""
   Tunnel Src./Dest. addr: 172.16.11.1/MGRE, Tunnel VRF ""
   Protocol/Transport: '
   Interface State Contr
   nhrp event-publisher
Type:Hub, Total NBMA Pe

# Ent  Peer NBMA Addr Pe
-----  --------------
    1 172.16.31.1
    1 172.16.41.1
```

```
IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 1

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Ne
-----  --------------- --------------- -----  -------- -----  ------------
    1 172.16.11.1      192.168.100.11    UP 00:00:28     S  192.168.100
```

```
R31-Spoke# show dmvpn d
! Output omitted for br

Interface Tunnel100 is
  Tunnel Src./Dest. add
  Protocol/Transport: "
  Interface State Contr
  nhrp event-publisher
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity

Interface Tunnel100 is up/up, Addr. is 192.168.100.41, VRF ""
   Tunnel Src./Dest. addr: 172.16.41.1/172.16.11.1, Tunnel VRF ""
   Protocol/Transport: "GRE/IP", Protect ""
   Interface State Control: Disabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 1

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
-----  --------------- --------------- -----  -------- -----  ----------------
    1 172.16.11.1      192.168.100.11    UP 00:02:00     S  192.168.100.11/32
```

DMVPN Configuration

Viewing the NHRP Cache

- Every router maintains a cache of requests that it receives or is processing. The NHRP cache contains the following fields displayed using the **show ip nhrp [brief]** :
  - Network entry for hosts or for a network/xx and the tunnel IP address to NBMA IP address.
  - The interface number, duration of existence, and expiration (hours:minutes:seconds).
  - The NHRP mapping entry type. Table 19-6 provides a list of NHRP mapping entries in the local cache.

Entry	Description
static	An entry created statically on a DMVPN interface.
dynamic	An entry created dynamically. DMVPN Phase 1: an entry created from a spoke that registered with an NHS with an NHRP registration request.
incomplete	A temporary entry placed locally while an NHRP resolution request processes. Incomplete entries prevent repetitive NHRP requests for the same entry, avoiding unnecessary consumption of router resources.
local	Displays local mapping info. Represents a local network that was advertised for an NHRP resolution reply.
(no-socket)	Mapping entries that do not have associated IPsec sockets and where encryption is not triggered.
NBMA address	Nonbroadcast multi-access address, or the transport IP address where the entry was received.

Viewing the NHRP Cache (Cont.)

- NHRP message flags specify attributes of an NHRP cache entry or of the peer for which the entry was created. Table 19-7 provides a list of the NHRP message flags and their meanings.

NHRP Flag	Description
Used	Indication that this NHRP mapping entry was used to forward data packets within the past 60 seconds.
Implicit	Indicates that the NHRP mapping entry was learned implicitly.
Unique	Indicates that this NHRP mapping entry must be unique and that it cannot be overwritten with a mapping entry that has the same tunnel IP address but a different NBMA address.
Router	Indicates that this NHRP mapping entry is from a remote router that provides access to a network or host that is located behind the remote router.
Rib	Indicates that this NHRP mapping entry has a corresponding routing entry in the routing table.
Nho	Indicates that this NHRP mapping entry has a corresponding path that overrides the next hop for a remote network, as installed by another routing protocol.
Nhop	Indicates an NHRP mapping entry for a remote next-hop address and its associated NBMA address.

Viewing the NHRP Cache (Cont.)

- Example 19-9 shows the local NHRP cache for the various routers in the sample topology.
- The traceroute shown in Example 19-12 verifies that R31 can connect to R41, but network traffic must still pass through R11.

Example 19-9 *Local NHRP Cache for DMVPN Phase 1*

```
R11-Hub# show ip nhrp
192.168.100.31/32 via 192.168.100.31
  Tunnel100 created 23:04:04, expire 01:37:26
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 172.16.31.1
192.168.100.41/32 via 192.168.100.41
  Tunnel100 created 23:04:00, expire 01:37:42
  Type: dynamic, Flags: unique registered used nhop
  NBMA address: 172.16.41.1
```

```
R31-Spoke# show ip nhrp
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 23:02:53, never expire
   Type: static, Flags:
   NBMA address: 172.16.11.1
```

```
R41-Spoke# show ip nhrp
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 23:02:53, never expire
   Type: static, Flags:
   NBMA address: 172.16.11.1
```

Example 19-12 *Phase 1 DMVPN Traceroute from R31 to R41*

```
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
  1 192.168.100.11 0 msec 0 msec 1 msec
  2 192.168.100.41 1 msec * 1 msec
```

DMVPN Configuration for Phase 3 DMVPN (Multipoint)

- The Phase 3 DMVPN configuration for the hub router adds the **ip nhrp redirect** interface parameter command on the hub router. This command checks the flow of packets on the tunnel interface and sends a redirect message to the source spoke router when it detects packets hairpinning out of the DMVPN cloud.
- Hairpinning means that traffic is received and sent out an interface in the same cloud (identified by the NHRP network ID).
- The steps for configuring a DMVPN Phase 3 spoke router are as follows:

1) **Step 1**. Create the tunnel interface by using the **interface tunnel** *tunnel-number* global configuration command.
2) **Step 2**. Identify the local source of the tunnel by using the **tunnel source** {*ip-address* | *interface-id*} interface parameter command.
3) **Step 3**. Configure the DMVPN tunnel as a GRE multipoint tunnel by using the **tunnel mode gre multipoint** interface parameter command.
4) **Step 4**. Allocate an IP address for the DMVPN network (tunnel) by using the **ip address** *ip-address subnet-mask* command.
5) **Step 5**. Enable NHRP and uniquely identify the DMVPN tunnel for the virtual interface by using the **ip nhrp network-id** *1-4294967295* interface parameter command.
6) **Step 6**. Optionally configure the tunnel key by using the **tunnel key** *0-4294967295* command.
7) **Step 7**. Enable the NHRP shortcut function by using the **ip nhrp shortcut** command.

DMVPN Configuration for Phase 3 DMVPN (Multipoint) (Cont.)

- The steps for configuring a DMVPN Phase 3 spoke router are as follows:

8) **Step 8**. Specify the address of one or more NHRP NHSs by using the **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**] command.

9) **Step 9**. Optionally define the IP MTU for the tunnel interface by using the **ip mtu** *mtu* interface parameter command.

10) **Step 10**. Optionally define the TCP MSS feature, which ensures that the router will edit the payload of a TCP three-way handshake if the MSS exceeds the configured value. The command is **ip tcp adjust-mss** *mss-size*.

DMVPN Configuration for Phase 3 DMVPN (Multipoint) (Cont.)

- Example 19-13 provides a sample configuration for R11 (hub), R21 (spoke), and R31 (spoke) configured with Phase 3 DMVPN.

Example 19-13 *DMVPN Phase 3 Configuration for Spokes*

```
R11-Hub
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R31-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R41-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.41 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.12 nbma 172.16.11.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

IP NHRP Authentication and Unique IP NHRP Registration

- NHRP authentication is weak because the password is stored in plaintext!
- Enable NHRP authentication by using the **ip nhrp authentication** *password* interface parameter command.
- When an NHC registers with an NHS, it provides the protocol address and the NBMA address. The NHS maintains a local cache of these settings. This capability is indicated by the NHRP message flag unique on the NHS, as shown in Example 19-14.

Example 19-14 *Unique NHRP Registration*

```
R11-Hub# show ip nhrp 192.168.100.31
192.168.100.31/32 via 192.168.100.31
    Tunnel100 created 00:11:24, expire 01:48:35
    Type: dynamic, Flags: unique registered used nhop
    NBMA address: 172.16.31.1
```

Spoke-to-Spoke Communication

Spoke-to-spoke communication

- This section focuses on the underlying mechanisms used to establish spoke-to-spoke communication.
- In DMVPN Phase 1, the spoke devices rely on the configured tunnel destination to identify where to send the encapsulated packets.
- Phase 3 DMVPN uses mGRE tunnels and thereby relies on NHRP redirect and resolution request messages to identify the NBMA addresses for any destination networks.

Spoke-to-Spoke Communication

- Packets flow through the hub in a traditional hub-and-spoke manner until the spoke-to-spoke tunnel has been established in both directions. As packets flow across the hub, the hub engages NHRP redirection to begin finding a more optimal path with spoke-to-spoke tunnels.
- In Example 19-16, R31 initiates a traceroute to R41. Notice that the first packet travels across R11 (hub), but by the time a second stream of packets is sent, the spoke-to-spoke tunnel has been initialized so that traffic flows directly between R31 and R41 on the transport and overlay networks.

Example 19-16 *Initiation of Traffic Between Spoke Routers*

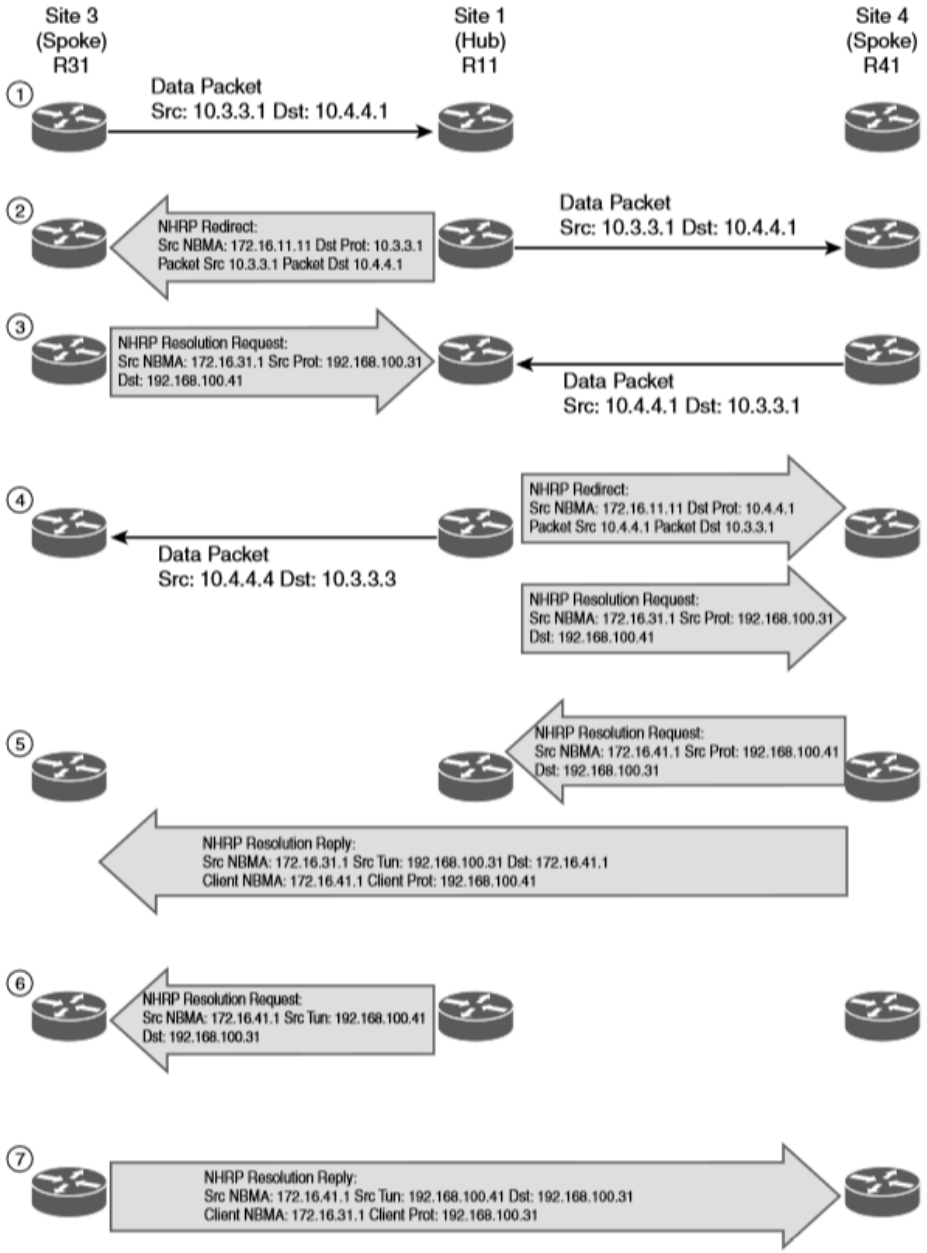
```
! Initial Packet Flow
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
  1 192.168.100.11 5 msec 1 msec 0 msec <- This is the Hub Router (R11-Hub)
  2 192.168.100.41 5 msec * 1 msec
```

```
! Packetflow after Spoke-to-Spoke Tunnel is Established
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
 1 192.168.100.41 1 msec * 0 msec
```

Spoke-to-Spoke Communication

Forming Spoke-to-Spoke Tunnels

- This section explains in detail how a spoke-to-spoke DMVPN tunnel is formed.
- Figure 19-5 illustrates the packet flow among all three devices—R11, R31, and R41—to establish a bidirectional spoke-to-spoke DMVPN tunnel.



Forming Spoke-to-Spoke Tunnels (Cont.)

- Example 19-17 shows the status of DMVPN tunnels on R31 and R41, where there are two new spoke-to-spoke tunnels (highlighted).
- The DLX entries represent the local (no-socket) routes. The original tunnel to R11 remains a static tunnel.

Example 19-17 *Detailed NHRP Mapping with Spoke-to-Hub Traffic*

```
R31-Spoke# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
       # Ent --> Number of NHRP entries with same NBMA peer
       NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
       UpDn Time --> Up or Down Time for a Tunnel
============================================================================
Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
    Tunnel Src./Dest. addr: 172.16.31.1/MGRE, Tunnel VRF ""
    Protocol/Transport: "multi-GRE/IP", Protect ""
    Interface State Control: Disabled
    nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1     192.168.100.31    UP 00:00:10   DLX         10.3.3.0/24
    2 172.16.41.1     192.168.100.41    UP 00:00:10   DT2    10.4.4.0/24
      172.16.41.1     192.168.100.41    UP 00:00:10   DT1    192.168.100.41/32
    1 172.16.11.1     192.168.100.11    UP 00:00:51     S    192.168.100.11/32
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    2 172.16.31.1     192.168.100.31    UP 00:00:34   DT2         10.3.3.0/24
      172.16.31.1     192.168.100.31    UP 00:00:34   DT1    192.168.100.31/32
    1 172.16.41.1     192.168.100.41    UP 00:00:34   DLX         10.4.4.0/24
    1 172.16.11.1     192.168.100.11    UP 00:01:15     S    192.168.100.11/32
```

Forming Spoke-to-Spoke Tunnels (Cont.)

- Example 19-18 shows the NHRP cache for R31 and R41.
- Notice the NHRP mappings: router, rib, nho, and nhop.

Example 19-18 *NHRP Mapping with Spoke-to-Hub Traffic*

```
R31-Spoke# show ip nhrp detail
10.3.3.0/24 via 192.168.100.31
   Tunnel100 created 00:01:44, expire 01:58:15
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.31.1
   Preference: 255
    (no-socket)
   Requester: 192.168.100.41 Request ID: 3
10.4.4.0/24 via 192.168.100.41
   Tunnel100 created 00:01:44, expire 01:58:15
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.16.41.1
   Preference: 255
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 10:43:18, never expire
   Type: static, Flags: used
   NBMA address: 172.16.11.1
   Preference: 255
192.168.100.41/32 via 192.168.100.41
   Tunnel100 created 00:01:45, expire 01:58:15
   Type: dynamic, Flags: router used nhop rib
   NBMA address: 172.16.41.1
   Preference: 255
```

```
R41-Spoke# show ip nhrp detail
10.3.3.0/24 via 192.168.100.31
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.16.31.1
   Preference: 255
10.4.4.0/24 via 192.168.100.41
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.41.1
   Preference: 255
     (no-socket)
   Requester: 192.168.100.31 Request ID: 3
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 10:43:42, never expire
   Type: static, Flags: used
   NBMA address: 172.16.11.1
   Preference: 255
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router used nhop rib
   NBMA address: 172.16.31.1 Preference: 255
```

Spoke-to-Spoke Communication

NHRP Routing Table Manipulation

- NHRP interacts with the routing/forwarding tables and installs or modifies routes in the routing table.
- In the event that an entry exists with an exact match for the network and prefix length, NHRP overrides the existing next hop with a shortcut.
- The original protocol is still responsible for the prefix, but overwritten next-hop addresses are indicated in the routing table by the percent sign (%).
- Example 19-19 provides the routing tables for R31 and R41.

Example 19-19 *NHRP Routing Table Manipulation*

```
R31-Spoke# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:44:45, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
D %      10.4.4.0/24 [90/52992000] via 192.168.100.11, 10:44:45, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:03:21, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0
S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:44:34, Tunnel100
D %      10.3.3.0/24 [90/52992000] via 192.168.100.11, 10:44:34, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:03:10, Tunnel100
```

Spoke-to-Spoke Communication

NHRP Routing Table Manipulation (Cont.)

- The command **show ip route next-hop-override** displays the routing table with the explicit NHRP shortcuts that were added.
- Example 19-20 shows the command's output for the sample topology.
- Notice that the NHRP shortcut is indicated by the NHO marking and shown underneath the original entry with the correct next-hop IP address.

Example 19-20 *Next-Hop Override Routing Table*

```
R31-Spoke# show ip route next-hop-override
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       + - replicated route, % - next hop override

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:46:38, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
D %      10.4.4.0/24 [90/52992000] via 192.168.100.11, 10:46:38, Tunnel100
                     [NHO][90/255] via 192.168.100.41, 00:05:14, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:05:14, Tunnel100
```

```
R41-Spoke# show ip route next-hop-override
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:45:44, Tunnel100
D %      10.3.3.0/24 [90/52992000] via 192.168.100.11, 10:45:44, Tunnel100
                     [NHO][90/255] via 192.168.100.31, 00:04:20, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:04:20, Tunnel100
```

Spoke-to-Spoke Communication

NHRP Routing Table Manipulation with Summarization

- Summarizing routes on WAN links provides stability by hiding network convergence and thereby adding scalability.
- This section demonstrates NHRP's interaction on the routing table when the exact route does not exist there.
- R11's EIGRP configuration now advertises the 10.0.0.0/8 summary prefix out tunnel 100.
- The spoke routers use the summary route for forwarding traffic until the NHRP establishes the spoke-to-spoke tunnel.
- The more explicit entries from NHRP are installed into the routing table after the spoke-to-spoke tunnels have been initialized.
- Example 19-21 shows the change to R11's EIGRP configuration for summarizing the 10.0.0.0/8 networks out the tunnel 100 interface.

Example 19-21 *R11's Summarization Configuration*

```
R11-Hub
router eigrp OVERLAY
 address-family ipv4 unicast autonomous-system 100
 af-interface Tunnel100
   summary-address 10.0.0.0 255.0.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
 exit-af-interface
 !
 topology base
 exit-af-topology
 network 10.0.0.0
 network 192.168.100.0
 exit-address-family
```

NHRP Routing Table Manipulation with Summarization (Cont.)

- You can clear the NHRP cache on all routers by using the command **clear ip nhrp**, which removes any NHRP entries.
- Example 19-22 shows the routing tables for R11, R31, and R41. Notice that only the 10.0.0.0/8 summary route provides initial connectivity among all three routers.

Example 19-22 *Routing Table with Summarization*

```
R11-Hub# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.11.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.11.2
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
D        10.0.0.0/8 is a summary, 00:28:44, Null0
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/27392000] via 192.168.100.31, 11:18:13, Tunnel100
D        10.4.4.0/24 [90/27392000] via 192.168.100.41, 11:18:13, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.11.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:29:28, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:29:54, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

NHRP Routing Table Manipulation with Summarization (Cont.)

- Traffic was re-initiated from 10.3.3.1 to 10.4.4.1 to initialize the spoke-to-spoke tunnels.
- R11 still sends the NHRP redirect for hairpinned traffic, and the pattern would complete as shown earlier except that NHRP would install a more specific route into the routing table on R31 (10.4.4.0/24) and R41 (10.3.3.0/24).
- The NHRP injected route is indicated by the H entry, as shown in Example 19-23.

Example 19-23 *Routing Table with Summarization and Spoke-to-Spoke Traffic*

```
R31-Spoke# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:31:06, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
H        10.4.4.0/24 [250/255] via 192.168.100.41, 00:00:22, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:00:22, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity

Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:31:24, Tunnel100
H        10.3.3.0/24 [250/255] via 192.168.100.31, 00:00:40, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:00:40, Tunnel100
```

NHRP Routing Table Manipulation with Summarization (Cont.)

- Example 19-24 shows the DMVPN tunnels after R31 and R41 have initialized the spoke-to-spoke tunnel with summarization on R11.
- Notice that both of the new spoke-to-spoke tunnel entries are DT1 because they are new routes in the RIB.
- If the routes had been more explicit (as shown in Example 19-19), NHRP would have overridden the next-hop address and used a DT2 entry.

Example 19-24 *Detailed DMVPN Tunnel Output*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
============================================================================
IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31    UP 00:01:17   DLX         10.3.3.0/24
    2 172.16.41.1      192.168.100.41    UP 00:01:17   DT1         10.4.4.0/24
      172.16.41.1      192.168.100.41    UP 00:01:17   DT1  192.168.100.41/32
    1 172.16.11.1      192.168.100.11    UP  11:21:33    S  192.168.100.11/32
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11 RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    2 172.16.31.1      192.168.100.31    UP 00:01:56   DT1         10.3.3.0/24
      172.16.31.1      192.168.100.31    UP 00:01:56   DT1  192.168.100.31/32
    1 172.16.41.1      192.168.100.41    UP 00:01:56   DLX         10.4.4.0/24
    1 172.16.11.1      192.168.100.11    UP 11:22:09     S  192.168.100.11/32
```

Problems with Overlay Networks

- Two problems are frequently found with tunnel or overlay networks: recursive routing and outbound interface selection.
- The following sections explain these problems and describe solutions to them.

Problems with Overlay Networks

Recursive Routing Problems

- If a router tries to reach the remote router's encapsulating interface (transport IP address) through the tunnel (overlay network), problems will occur.
- This is a common issue when a transport network is advertised into the same routing protocol that runs on the overlay network.
- Figure 19-6 demonstrates a simple GRE tunnel between R11 and R31. R11, R31, and the SP routers are running OSPF on the 100.64.0.0/16 transport networks. R11 and R31 are running EIGRP on the 10.0.0.0/8 LAN and 192.168.100.0/24 tunnel network.
- Example 19-25 shows R11's routing table, with everything working properly.

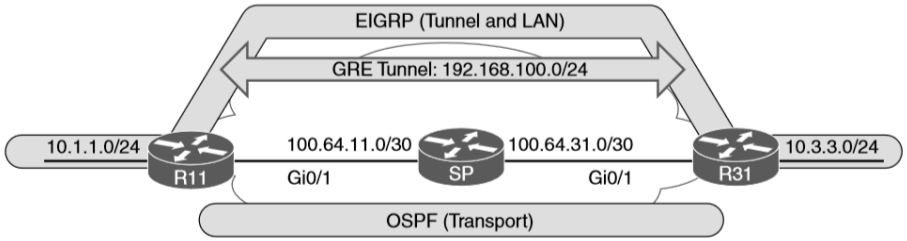


Figure 19-6 *Typical LAN Network*

Example 19-25 *R11 Routing Table with GRE Tunnel*

```
R11# show ip route
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/25610240] via 192.168.100.31, 00:02:35, Tunnel0
      100.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
```

Recursive Routing Problems (Cont.)

- An administrator has accidentally added the 100.64.0.0/16 network interfaces to EIGRP on R11 and R31. The SP router is not running EIGRP, so an adjacency does not form, but R11 and R31 add the transport network to EIGRP, which has a lower AD than OSPF. The routers then try to use the tunnel to reach the tunnel endpoint address, which is not possible. This scenario is known as recursive routing.
- The router detects recursive routing and provides an appropriate syslog message, as shown in Example 19-26.
- The tunnel is brought down, which terminates the EIGRP neighbors, and then R11 and R31 find each other again by using OSPF.
- The tunnel is reestablished, EIGRP forms a relationship, and the problem repeats over and over again.
- Only point-to-point GRE tunnels provide the syslog message "temporarily disabled due to recursive routing." Both DMVPN and GRE tunnels use the message "looped chained attempting to stack."

Example 19-26 *Recursive Routing Syslog Messages on R11 for GRE Tunnels*

```
00:49:52: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.100.31 (Tunnel100)
          is up: new adjacency
00:49:52: %ADJ-5-PARENT: Midchain parent maintenance for IP midchain out of
          Tunnel100 - looped chain attempting to stack
00:49:57: %TUN-5-RECURDOWN: Tunnel100 temporarily disabled due recursive routing
00:49:57: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100, changed
          state to down
00:49:57: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.30.3 (Tunnel100) is
          down: interface down
00:50:12: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100, changed
          state to up
00:50:15: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.100.31 (Tunnel100)
          is up: new adjacency
```

Outbound Interface Selection

- In certain scenarios, it is difficult for a router to properly identify the outbound interface for encapsulating packets for a tunnel.
- Typically a branch site uses multiple transports (one DMVPN tunnel per transport) for network resiliency.
- Imagine that R31 is connected to two different Internet service providers that receive their IP addresses from DHCP. R31 would have only two default routes for providing connectivity to the transport networks, as shown in Example 19-27.

Example 19-27 *Two Default Routes and Path Selection*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 172.16.31.2
                [254/0] via 100.64.31.2
C      100.64.31.0/30 is directly connected, GigabitEthernet0/2
C      172.16.31.0/30 is directly connected, GigabitEthernet0/1
```

Front Door Virtual Routing and Forwarding

- Virtual routing and forwarding (VRF) contexts create unique logical routers on a physical router so that router interfaces, routing tables, and forwarding tables are completely isolated from other VRF instances.
  - The routing table of one transport network is isolated from the routing table of the other transport network and that the routing table of the LAN interfaces is separate from those of all the transport networks.
  - DMVPN tunnels are VRF aware in the sense that the tunnel source or destination can be associated to a different VRF instance from the DMVPN tunnel itself.
  - Using an FVRF instance for every DMVPN tunnel prevents route recursion because the transport and overlay networks remain in separate routing tables.
  - VRF instances are locally significant, but the configuration/naming should be consistent to simplify the operational aspects.

Configuring Front Door VRF (FVRF)

- The following steps are required to create an FVRF instance, assign it to the transport interface, and make the DMVPN tunnel aware of the FRF instance:

1) **Step 1.** Create the FVRF instance by using the **vrf definition** *vrf-name* command.
2) **Step 2.** Initialize the appropriate address family for the transport network by using the command **address-family {ipv4 | ipv6}**. The address family can be IPv4, IPv6, or both.
3) **Step 3.** Enter interface configuration submode and specify the interface to be associated with the VRF instance by using the command **interface** *interface-id*. The VRF instance is linked to the interface with the interface parameter command **vrf forwarding** *vrf-name*.
4) **Step 4.** Configure an IPv4 address by using the command **ip address** *ip-address* subnet-mask or an IPv6 address by using the command **ipv6 address** *ipv6-address/prefix-length*.
5) **Step 5.** Associate the FVRF instance with the DMVPN tunnel by using the interface parameter command **tunnel vrf** *vrf-name* on the DMVPN tunnel.
   - If an IP address is already configured on the interface, when the VRF instance is linked to the interface, the IP address is removed from that interface.

Configuring Front Door VRF (FVRF) (Cont.)

- Example 19-28 shows how the FVRF instances named INET01 and INET02 are created on R31.
- Notice that when the FVRF instances are associated, the IP addresses are removed from the interfaces.
- The IP addresses are reconfigured and the FVRF instances are associated with the DMVPN tunnels.

Example 19-28 *FVRF Configuration Example*

```
R31-Spoke(config)# vrf definition INET01
R31-Spoke(config-vrf)# address-family ipv4
R31-Spoke(config-vrf-af)# vrf definition INET02
R31-Spoke(config-vrf)# address-family ipv4
R31-Spoke(config-vrf-af)# interface GigabitEthernet0/1
R31-Spoke(config-if)# vrf forwarding INET01
% Interface GigabitEthernet0/1 IPv4 disabled and address(es) removed due to
     enabling VRF INET01
R31-Spoke(config-if)# ip address 172.16.31.1 255.255.255.252
R31-Spoke(config-if)# interface GigabitEthernet0/2
R31-Spoke(config-if)# vrf forwarding INET02
% Interface GigabitEthernet0/2 IPv4 disabled and address(es) removed due to
     enabling VRF INET02
R31-Spoke(config-if)# ip address dhcp
R31-Spoke(config-if)# interface tunnel 100
R31-Spoke(config-if)# tunnel vrf INET01
R31-Spoke(config-if)# interface tunnel 200
R31-Spoke(config-if)# tunnel vrf INET02
```

FVRF Static Routes

- FVRF interfaces that are assigned an IP address by DHCP automatically install a default route with an AD of 254. FVRF interfaces with static IP addressing require only a static default route in the FVRF context. This is accomplished with the command ip route vrf vrf-name 0.0.0.0 0.0.0.0 next-hop-ip. Example 19-29 shows the configuration for R31 for the INET01 FVRF instance. The INET02 FVRF instance does not need a static default route because it gets the route from the DHCP server.

Example 19-29 *FVRF Static Default Route Configuration*

```
R31-Spoke
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.31.2
```

DMVPN Failure Detection and High Availability

Failure Detection and HA

- An NHRP mapping entry stays in the NHRP cache for a finite amount of time.
  - The entry is valid based on the NHRP holdtime period, which defaults to 7200 seconds (2 hours).
  - The NHRP holdtime can be modified with the interface parameter command **ip nhrp holdtime** *1-65535* and should be changed to the recommended value of 600 seconds.
- A secondary function of the NHRP registration packets is to verify that connectivity is maintained to the NHSs (hubs).
  - NHRP registration messages are sent every NHRP timeout period, and if the NHRP registration reply is not received for a request, the NHRP registration request is sent again with the first packet delayed for 1 second, the second packet delayed for 2 seconds, and the third packet delayed for 4 seconds.
  - The NHS is declared down if the NHRP registration reply has not been received after the third retry attempt.

DMVPN Failure Detection and High Availability

- The spoke-to-hub registration is taken down and displays as NHRP for the tunnel state when examined with the **show dmvpn** command. The actual tunnel interface still has a line protocol state of up.
- During normal operation of the spoke-to-hub tunnels, the spoke continues to send periodic NHRP registration requests, refreshing the NHRP timeout entry and keeping the spoke-to-hub tunnel up.
  - However, in spoke-to-spoke tunnels, if a tunnel is still being used within 2 minutes of the expiration time, an NHRP request refreshes the NHRP timeout entry and keeps the tunnel.
  - If the tunnel is not being used, it is torn down.
- The NHRP timeout period defaults to one-third of the NHRP holdtime, which equates to 2400 seconds (40 minutes).
  - The NHRP timeout period can be modified using the interface parameter command **ip nhrp registration timeout** *1-65535*.
- When an NHS is declared down, NHCs still attempt to register with the down NHS.
  - This is known as the probe state.
  - The delay between retry packets increments between iterations and uses the following delay pattern: 1, 2, 4, 8, 16, 32, and 64 seconds. The delay never exceeds 64 seconds, and after a registration reply is received, the NHS (hub) is declared up again.

DMVPN Hub Redundancy

- Connectivity from a DMVPN spoke to a hub is essential to maintain connectivity.
  - If the hub fails, or if a spoke loses connectivity to a hub, that DMVPN tunnel loses its ability to transport packets.
- Deploying multiple DMVPN hubs for the same DMVPN tunnel provides redundancy and eliminates a single point of failure.
  - Additional DMVPN hubs are added simply by adding NHRP mapping commands to the tunnel interface.
  - All active DMVPN hubs participate in the routing domain for exchanging routes.
  - DMVPN spoke routers maintain multiple NHRP entries (one per DMVPN hub).

IPv6 DMVPN Configuration

IPv6 Compatibility

- DMVPN uses GRE tunnels and is capable of tunneling multiple protocols. Enhancements to NHRP added support for IPv6 so that mGRE tunnels can find the appropriate IPv6 addresses.
- This means that DMVPN supports the use of IPv4 and IPv6 as the tunnel protocol or the transport protocol in the combination required.

IPv6 DMVPN Configuration Commands

- For all the commands explained earlier for IPv4, there are equivalent commands to support IPv6. Table 19-8 provides a list of the tunneled protocol commands for IPv4 and the equivalent commands for IPv6.

Table 19-8 Correlation of IPv4-to-IPv6 Tunneled Protocol Commands

IPv4 Command	IPv6 Command
ip mtu *mtu*	ipv6 mtu *mtu*
ip tcp adjust-mss *mss-size*	ipv6 tcp adjust-mss *mss-size*
ip nhrp network-id *1-4294967295*	ipv6 nhrp network-id *1-4294967295*
ip nhrp nhs *nhs-address* nbma *nbma-address* [multicast] [priority *0-255*]	ipv6 nhrp nhs *nhs-address* nbma *nbma-address* [multicast] [priority *0-255*]
ip nrhp redirect	ipv6 nhrp redirect
ip nhrp shortcut	ipv6 nhrp shortcut
ip nhrp authentication *password*	ipv6 nhrp authentication *password*
ip nhrp registration no-unique	ipv6 nhrp registration no-unique
ip nhrp holdtime *1-65535*	ipv6 nhrp holdtime *1-65535*
ip nhrp registration timeout *1-65535*	ipv6 nhrp registration timeout *1-65535*

IPv6 DMVPN Configuration Commands (Cont.)

- Table 19-9 provides a list of the configuration commands that are needed to support an IPv6 transport network.
- Any tunnel commands not listed in Table 19-9 are transport agnostic and are used regardless of the transport IP protocol version.

Table 19-9 Correlation of IPv4-to-IPv6 Transport Protocol Commands

IPv4 Command	IPv6 Command
tunnel mode gre multipoint	tunnel mode gre multipoint ipv6
ip route vrf *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*	ipv6 route vrf *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*

IPv6 DMVPN Configuration Commands (Cont.)

- IPv6 over DMVPN can be interpreted differently depending on the perspective. There are three possible interpretations:
  - IPv4 over IPv6: IPv4 is the tunneled protocol over an IPv6 transport network.
  - IPv6 over IPv6: IPv6 is the tunneled protocol over an IPv6 transport network.
  - IPv6 over IPv4: IPv6 is the tunneled protocol over an IPv4 transport network.
- Regardless of the interpretation, DMVPN supports the IPv4 or IPv6 protocol as the tunneled protocol or the transport, but choosing the correct set of command groups is vital and should be based on the tunneling technique selected.
- Table 19-10 provides a matrix to help you select the appropriate commands from Table 19-8 and Table 19-9. It is important to note that nhs-address or NBMA-address in Table 19-8 can be IPv4 or IPv6 addresses.

Table 19-10 Matrix of DMVPN Tunnel Technique to Configuration Commands

Tunnel Mode	Tunnel Protocol Commands	Transport Commands
IPv4 over IPv4	IPv4	IPv4
IPv4 over IPv6	IPv4	IPv6
IPv6 over IPv4	IPv6	IPv4
IPv6 over IPv6	IPv6	IPv6

IPv6 DMVPN Configuration Commands (Cont.)

- Table 19-11 provides a list of IPv4 display commands correlated to the IPv6 equivalents.

Table 19-11 Display Commands for IPv6 DMVPN

IPv4 Command	IPv6 Command
show ip nhrp [brief | detail]	show ipv6 nhrp [brief | detail]
show dmvpn [ipv4][detail]	show dmvpn [ipv6][detail]
show ip nhrp traffic	show ipv6 nhrp traffic
show ip nhrp nhs [detail]	show ipv6 nhrp nhs [detail]

IPv6-over-IPv6 Sample Configuration

- This section provides a sample configuration using the topology from Figure 19-4 for the IPv6-over-IPv6 topology.
- To simplify the IPv6 addressing scheme, the first two hextets of the book's IPv6 addresses use 2001:db8 (the RFC-defined address space for IPv6 documentation).
- After the first two hextets, an IPv4 octet number is copied into an IPv6 hextet, so the IPv6 addresses should look familiar.
- Table 19-12 provides an example of how the book converts existing IPv4 addresses and networks to IPv6 format.

Table 19-12 IPv6 Addressing Scheme

IPv4 Address	IPv4 Network	IPv6 Address	Network
10.1.1.11	10.1.1.0/24	2001:db8:10:1:1::11	2001:db8:10:1:1::/80
172.16.11.1	172.16.11.0/30	2001:db8:172:16:11::1	2001:db8:172:16:11::/126
10.1.0.11	10.1.0.11/32	2001:db8:10:1:0::11	2001:db8:10.1::11/128

IPv6-over-IPv6 Sample Configuration

- Example 19-30 provides the IPv6-over-IPv6 DMVPN configuration for hub router R11.
- The VRF definition uses the **address-family ipv6** command, and the GRE tunnel is defined with the command **tunnel mode gre multipoint ipv6**.
- Notice that the tunnel interface has a regular IPv6 address configured as well as a link-local IPv6 address.

Example 19-30 *IPv6 DMVPN Hub Configuration on R11*

```
R11-Hub
vrf definition INET01
 address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-INET
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:100::11 link-local
 ipv6 address 2001:DB8:192:168:100::11/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 100
 ipv6 nhrp holdtime 600
 ipv6 nhrp redirect
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf INET01
!
interface GigabitEthernet0/1
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:172:16:11::1/126
interface GigabitEthernet1/0
 description LAN
 ipv6 address 2001:DB8:10:1:111::11/80
!
ipv6 route vrf INET01 ::/0 GigabitEthernet0/1 2001:DB8:172:16:11::2
```

IPv6 DMVPN Verification

- The **show dmvpn [detail]** command can be used for viewing any DMVPN tunnel, regardless of the tunnel or transport protocol.
- The data is structured slightly differently because of the IPv6 address format, but it still provides the same information as before.
- Example 19-32 shows the DMVPN tunnel state from R31 after it has established its static tunnels to the DMVPN hubs.
- Notice that the protocol transport now shows IPv6, and the NHS devices are using IPv6 addresses.

Example 19-32 *Verification of IPv6 DMVPN*

```
R31-Spoke# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==============================================================================


Interface Tunnel100 is up/up, Addr. is 2001:DB8:192:168:100::31, VRF ""
   Tunnel Src./Dest. addr: 2001:DB8:172:16:31::1/MGRE, Tunnel VRF "INET01"
   Protocol/Transport: "multi-GRE/IPv6", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv6 NHS:
2001:DB8:192:168:100::11  RE NBMA Address: 2001:DB8:172:16:11::1 priority = 0 clus-
ter = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2
    1.Peer NBMA Address: 2001:DB8:172:16:11::1
        Tunnel IPv6 Address: 2001:DB8:192:168:100::11
        IPv6 Target Network: 2001:DB8:192:168:100::11/128
        # Ent: 2, Status: UP, UpDn Time: 00:00:53, Cache Attrib: S
! Following entry is shown in the detailed view and uses link-local addresses
    2.Peer NBMA Address: 2001:DB8:172:16:11::1
        Tunnel IPv6 Address: FE80:100::11
        IPv6 Target Network: FE80:100::11/128
        # Ent: 0, Status: NHRP, UpDn Time: never, Cache Attrib: SC
```

IPv6 DMVPN Verification (Cont.)

- Example 19-33 demonstrates the connectivity between R31 and R41 before and after the spoke-to-spoke DMVPN tunnel is established.

Example 19-33 *IPv6 Connectivity Between R31 and R41*

```
! Initial packet flow
R31-Spoke# traceroute 2001:db8:10:4:4::41
Tracing the route to 2001:DB8:10:4:4::41
  1 2001:DB8:192:168:100::11 2 msec
  2 2001:DB8:192:168:100::41 5 msec 4 msec 5 msec
```

```
! Packet flow after spoke-to-spoke tunnel is established
R31-Spoke# traceroute 2001:db8:10:4:4::41
Tracing the route to 2001:DB8:10:4:4::41
  1 2001:DB8:192:168
```

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 19

Description	
Generic Routing Encapsulation (GRE) tunnels	Phase 1 DMVPN spoke configuration
GRE tunnel configuration	Alternative NHRP mapping commands
Next Hop Resolution Protocol (NHRP)	Viewing DMVPN tunnel status
NHRP message types	Phase 3 DMVPN spoke configuration
Dynamic Multipoint VPN (DMVPN)	IP NHRP authentication
Phase 1 DMVPN	Unique IP NHRP registration
Phase 3 DMVPN	Forming spoke-to-spoke DMVPN tunnels
DMVPN hub configuration	NHRP routing table manipulation

Prepare for the Exam

Key Topics for Chapter 19 (Cont.)

Description	
NHRP route table manipulation with summarization	DMVPN failure detection and high availability
Recursive routing problems	DMVPN hub redundancy
Outbound interface selection	IPv6 DMVPN configuration
Front door virtual routing and forwarding (FVRF)	

Prepare for the Exam

Key Terms for Chapter 19

Term	
Dynamic Multipoint Virtual Private Network (DMVPN)	GRE tunnel
DMVPN Phase 1	Next Hop Resolution Protocol (NHRP)
DMVPN Phase 3	NHRP redirect
encapsulating interface	NHRP shortcut
front door VRF	next-hop server (NHS, recursive routing)

Prepare for the Exam

Command Reference for Chapter 19

Task	Command Syntax
Specify the source IP address or interface used for encapsulating packets for a tunnel	**tunnel source** *{ip-address | interface-id}*
Specify the destination IP address for establishing a tunnel	**tunnel destination ip-address**
Convert a GRE tunnel into an mGRE tunnel	**tunnel mode gre multipoint**
Enable NRHP and uniquely identify a DMVPN tunnel locally	**ip nhrp network-id** *1-4294967295*
Define a tunnel key globally on a DMVPN tunnel interface to allow routers to identify when multiple tunnels use the same encapsulating interface	**tunnel key 0-4294967295**
Enable plaintext NHRP authentication	**ip nhrp authentication password**
Associate a front door VRF instance to a DMVPN tunnel interface	**tunnel vrf** *vrf-name*

Prepare for the Exam

Command Reference for Chapter 19 (Cont.)

Task	Command Syntax
Allow for an NHRP client to register with a different IP address before timing out at the hub	**ip nhrp registration no-unique**
Enable the NHRP redirect function on a DMVPN hub tunnel interface	**ip nhrp redirect**
Enable the ability to install NHRP shortcuts into a spoke router's RIB	**ip nhrp shortcut**
Enable the mapping of multicast on a DMVPN hub tunnel interface	**ip nhrp map multicast dynamic**
Specify the NHRP NHS, NBMA address, and multicast mapping on a spoke	**ip nhrp nhs** *nhs-address* **nbma** *nbma-address* **[multicast]** **Or** **ip nhrp nhs** nhs-address **ip nhrp map** **ip nhrp map multicast** [nbma-address | dynamic]

Prepare for the Exam

Command Reference for Chapter 19 (Cont.)

Task	Command Syntax
Display the tunnel interface state and statistics	**show interface tunnel** *number*
Display DMVPN tunnel interface association, NHRP mappings, and IPsec session details	**show dmvpn** [detail]
Display the NHRP cache for a router	**show ip nhrp** [brief]
Display the NHRP shortcut that is installed for an overridden route	**show ip route next-hop-override**

Slides adapted by
Vladimír Veselý and Matěj Grégr
partially from official course materials

Last update: 2025-04-16

Securing DMVPN Tunnels

CE2 M20

Chapter 20 Content

- **Elements of Secure Transport** - This section explains the need for data integrity, data confidentiality, and data availability.
- **IPsec Fundamentals** - This section explains the core concepts involved with IP security encryption.
- **IPsec Tunnel Protection** - This section explains how IPsec protection integrates with DMVPN tunnels.

Elements of Secure Transport

- A properly designed network provides data confidentiality, integrity, and availability.
- Without these components, a business might lose potential customers if the customers do not think that their information is secure.

Elements of Secure Transport concerning Data

- **Data confidentiality** - Ensuring that data is viewable only by authorized users. Data confidentiality is maintained through encryption.
- **Data integrity** - Ensuring that data is modified only by authorized users. Information is valuable only if it is accurate. Inaccurate data can result in an unanticipated cost. Data integrity is maintained by using an encrypted digital signature, which is typically a checksum.
- **Data availability** - Ensuring that the network is always available allows for the secure transport of the data. Redundancy and proper design ensure data availability. encryption.

Elements of Secure Transport Typical WAN Network

- Figure 20-1 shows the traditional approach to securing data on a network.
- The entire controlled infrastructure (enterprise and SP) is assumed to be safe.
- Traffic is encrypted only when exposed to the public internet.

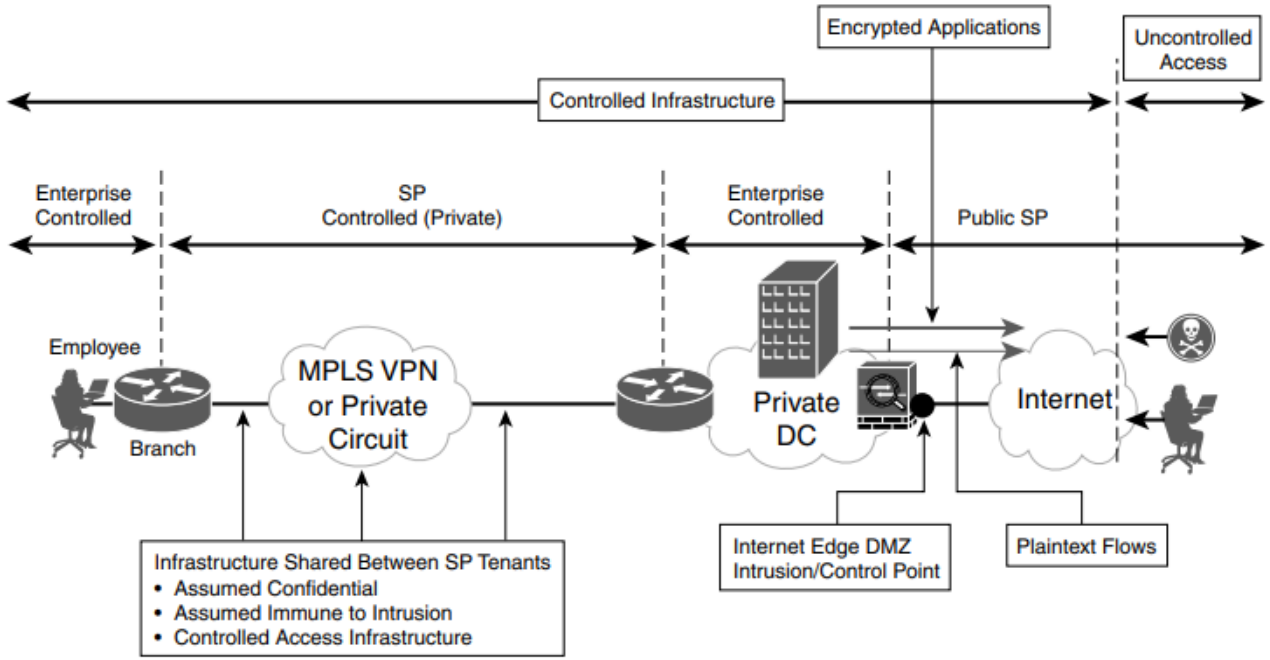


Figure 20-1 *Typical WAN Network*

Elements of Secure Transport Internet as a WAN Transport

- In Figure 20-2, the internet is used as the transport for the WAN. The internet does not provide controlled access and cannot guarantee data integrity or data confidentiality.
- Data confidentiality and integrity are maintained by adding IPsec encryption to the DMVPN tunnel that uses the internet as a transport.
- IPsec is a set of industry standards defined in RFC 2401 to secure IP-based network traffic.

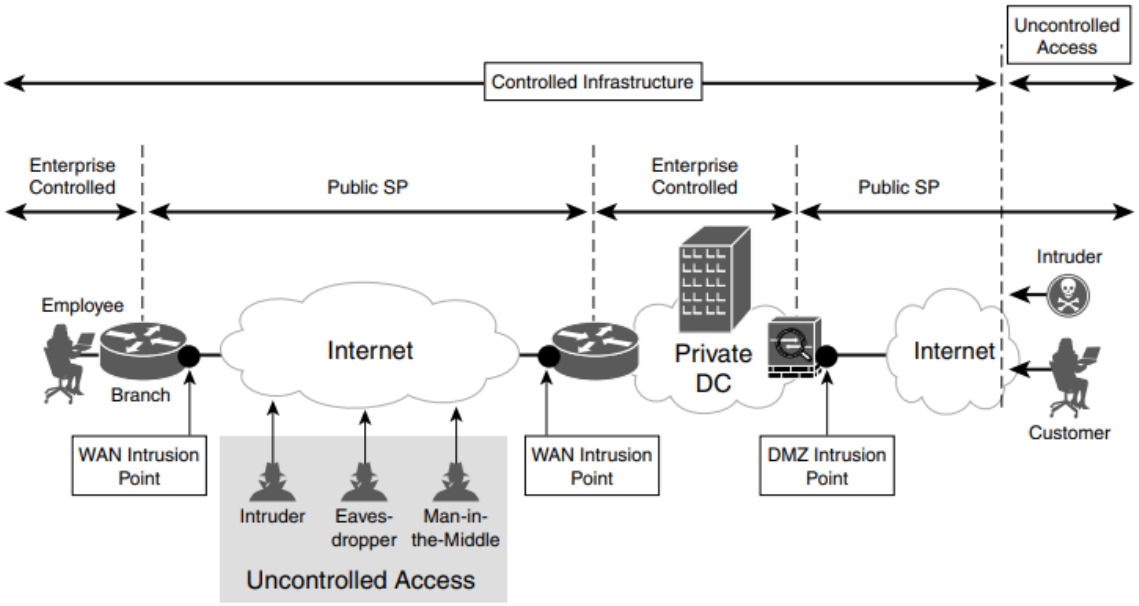


Figure 20-2 *Internet as a WAN Transport*

IPsec Fundamentals

- DMVPN tunnels are not encrypted by default, but they can be encrypted by using IPsec.
- IPsec provides encryption through cryptographically based security.
- The IPsec security architecture is composed of the following independent components: security protocols, security associations, and key management.

IPsec Fundamentals
IPsec with DMVPN Tunnels

▪When IPsec is integrated with DMVPN tunnels, the encrypted DMVPN tunnels provide a secure overlay network over any transport with the following functions:

- **Origin authentication** - Authentication of origin is accomplished by Pre-Shared Key (static) or through certificate-based authentication (dynamic).
- **Data confidentiality** - Ensuring that data is viewable only by authorized users. Data confidentiality is maintained through encryption. A variety of encryption algorithms are used to preserve confidentiality.
- **Data integrity** - Hashing algorithms ensure that packets are not modified in transit.
- **Replay detection** - This provides protection against hackers trying to capture and insert network traffic.
- **Periodic rekey** - New security keys are created between endpoints every specified time interval or within a specific volume of traffic.
- **Perfect forward secrecy** - Each session key is derived independently of the previous key. A compromise of one key does not mean compromise of future keys.

Security Protocols

- IPsec uses two encapsulation protocols
  - **Authentication Header** - The IP authentication header provides data integrity, authentication, and protection from hackers replaying packets. It uses protocol number 51 (located in the IP header) to create a digital signature to ensure that the packet has not been modified during transport.
  - **Encapsulating Security Payload (ESP)** - The Encapsulating Security Payload (ESP) provides data confidentiality, authentication, and protection from hackers replaying packets. Typically, payload refers to the actual data minus any headers, but in the context of ESP, the payload is the portion of the original packet that is encapsulated in the IPsec headers. ESP uses the protocol number 50 located in the IP header.

IPsec Fundamentals
Key Management and Security Associations

- **Key Management** – Part of secure encryption is communicating the keys used to encrypt and decrypt traffic that is being transported over the insecure network.
- The process of generating, distributing, and storing these keys is called key management. IPSec uses **Internet Key Exchange (IKE)** protocol by default.
  - IKEv2 provides mutual authentication of each party.
  - IKEv2 introduced support of Extensible Authentication Protocol (EAP) (certificate-based authentication), reduction of bandwidth consumption, Network Address Translation (NAT), and the ability to detect whether a tunnel is still alive.
- **Security Associations (SAs)** – SAs contain the security parameters that were agreed upon between the two endpoint devices. There are two types of SAs:
    - **IKE SA** - Used for control plane functions like IPsec key management and management of IPsec SAs. Can have one IKE SA between endpoints.
    - **IPsec SA** - Used for data plane functions to secure data transmitted between two different sites. IPsec SAs are unidirectional. They require one inbound and one outbound to exchange network traffic between two sites.

DMVPN Packet Headers

- **ESP Modes -** Traditional IPsec provides two ESP modes of packet protection:
  - **Tunnel Mode** – Encrypts the entire original packet and adds a new set of IPsec headers. These new headers are used to route the packet and also to provide overlay functions.
  - **Transport Mode** – Encrypts and authenticates only the packet payload. This mode does not provide overlay functions and routes based on the original IP headers.

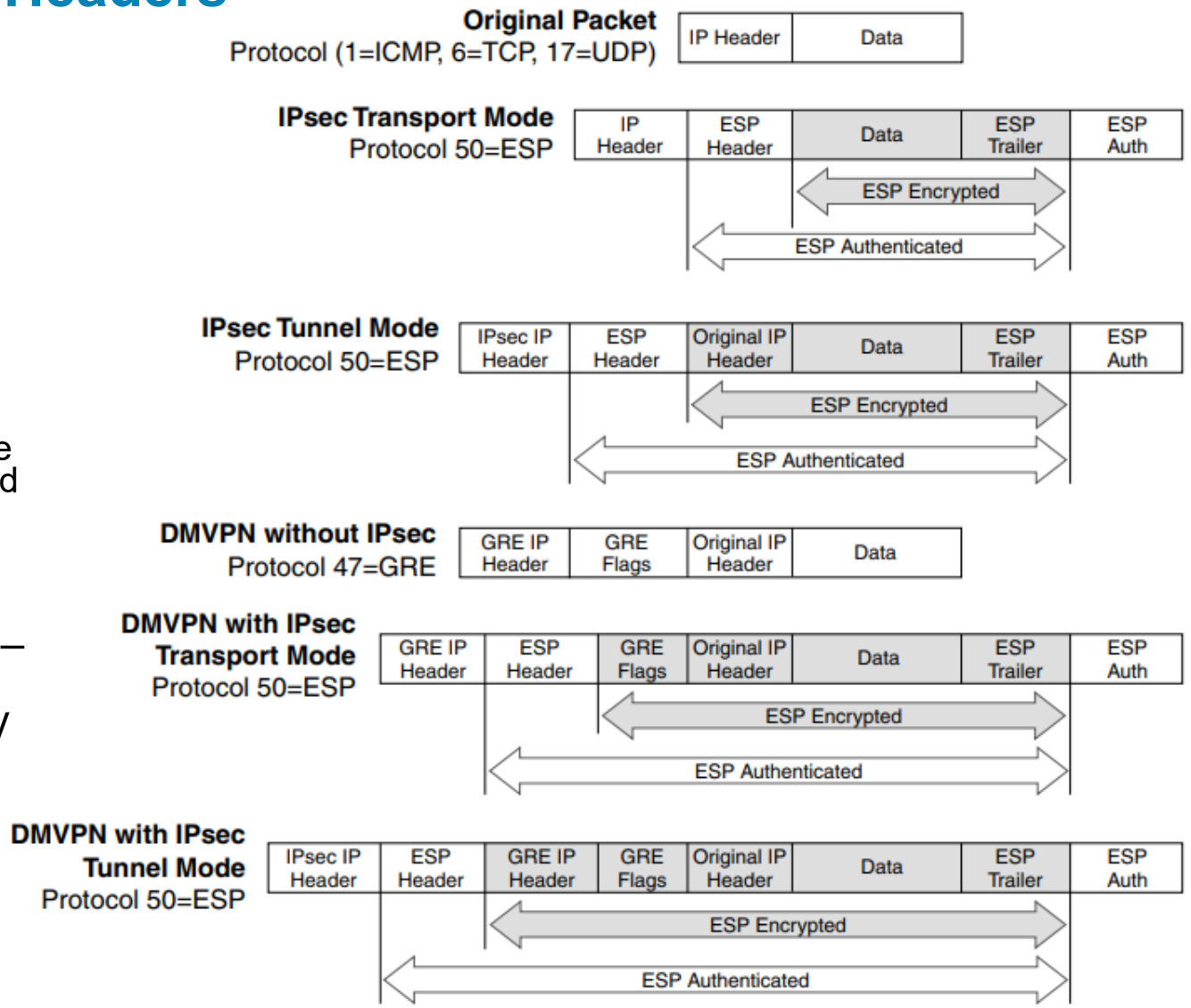


Figure 20-3 *DMVPN Packet Headers*

IPsec Fundamentals: ESP Modes

- **DMVPN Without Ipsec**
  - In unencrypted DMVPN packets, the original packets have GRE flags added to them.
  - Then the new GRE IP header is added for routing the packets on the transport (underlay) network.
  - The GRE IP header adds an extra 20 bytes of overhead, and the GRE flags add an extra 4 bytes. These packets use the protocol field of GRE (47).
- **DMVPN with IPsec in Transport Mode**
  - For encrypted DMVPN packets that use ESP transport mode, the original packets have the GRE flags added, then that portion of the packets is encrypted. A signature for the encrypted payload is added, and then a GRE IP header is added for routing the packets on the transport network.
  - The GRE IP header adds an extra 20 bytes of overhead, the GRE flags add an extra 4 bytes, and depending on the encryption mechanism, a varying number of bytes are added for the encrypted signature.
  - These packets use the protocol field of ESP (50).
- **DMVPN with IPsec in Tunnel Mode**
  - For encrypted DMVPN packets that use ESP tunnel mode, the original packets have GRE flags added to them, and then a new GRE IP header is added for routing the packets on the transport network. That portion of the packets is encrypted, a signature for the encrypted payload is added, and a new IPsec IP header is added for routing the packets on the transport network.
  - The GRE IP header adds an extra 20 bytes of overhead, the GRE flags add an extra 4 bytes, the IPsec IP header adds an extra 20 bytes, and depending on the encryption mechanism, a varying number of bytes are added for the encrypted signature.
  - These packets use the IP protocol field of ESP (50). IPsec tunnel mode for DMVPN does not add value, transport mode should be used for encrypted DMVPN tunnels.

IPsec Tunnel Protection

- Enabling IPsec protection on a DMVPN network requires that all devices have IPsec protection enabled.
- If some routers have IPsec enabled and others do not, devices with mismatched settings will not be able to establish connections on the tunnel interfaces.

IPsec Tunnel Protection

Pre-Shared Key Authentication

- The first scenario for deploying IPsec tunnel protection is with the use of static Pre-Shared Key, which involves the creation of the following:
  - IKEv2 keyring
  - IKEv2 profile
  - IPsec transform set
  - IPsec profile
- In this section, emphasis is on the DMVPN routers that are attached to the internet, as shown in Figure 20-4.

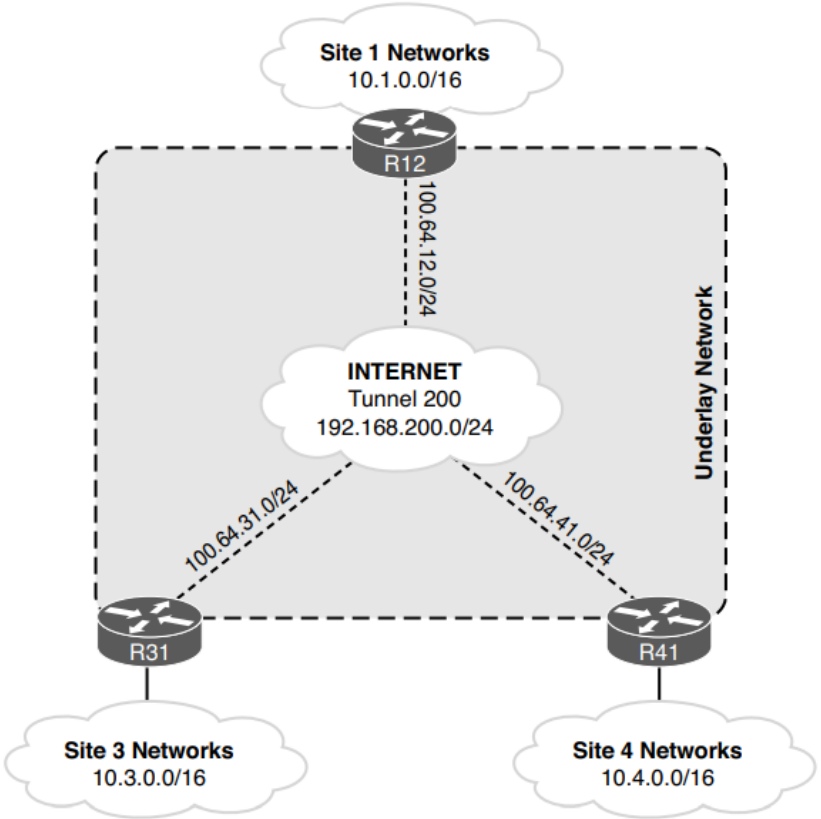


Figure 20-4 *Sample DMVPN Network*

IPsec Tunnel Protection

IKEv2 Keyring

- The IKEv2 keyring is a repository of the pre-shared keys and is created with the following steps:
- **Step 1**. Create the keyring with the command **crypto ikev2 keyring** *keyring-name*.
- **Step 2**. Create the peer with the command **peer** *peer-name*. Multiple peers can exist in a keyring. Each peer has a matching qualifier and can use a different password.
- **Step 3**. Identify the IP address so that the appropriate peer configuration is used, based on the remote device's IP address. The command **address** *network subnet-mask* defines the IP address range.
- **Step 4**. Define the pre-shared key with the command **pre-shared-key** *secure-key*. Generally a long and alphanumeric password is used for increased security.

Example 20-1 *IKEv2 Keyring*

```
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
 address 0.0.0.0 0.0.0.0
 pre-shared-key CISCO456
```

IPsec Tunnel Protection

IKEv2 Profile

- The IKEv2 profile is a collection of nonnegotiable security parameters used during the IKE security association:
  - **Step 1**. Define the IKEv2 profile by using the command **crypto ikev2 profile** *ike-profile-name*.
  - **Step 2**. Define the peer IP address with the command **match identity remote address** *ip-address*.
  - **Step 3**. Optionally set the local router's identity based on an IP address by using the command **identity local** address *ip-address*.
  - **Step 4**. If Front Door VRF (FVRF) is used on the DMVPN tunnel, associate the FVRF instance with the IKEv2 profile with the command **match fvrf** {*vrf-name* | **any**}.
  - **Step 5**. Define the authentication method for connection requests received by remote peers by using the command **authentication local** {**pre-share** | **rsa-sig**}. The **pre-share** keyword is used for static keys, and **rsa-sig** is used for certificate-based authentication.
  - **Step 6**. Define the authentication method for connection requests sent to remote peers by using the command **authentication remote** {**pre-share** | **rsa-sig**}.
  - **Step 7**. For pre-shared authentication, associate the IKEv2 keyring with the IKEv2 profile by using the command **keyring local** *keyring-name*.

IKEv2 Profile (Cont.)

- The IKEv2 profile settings are displayed with the command **show crypto ikev2 profile**, as shown in Example 20-3.
- Notice that the authentication, FVRF, IKE keyring, and identity IP address are displayed along with the IKE lifetime.

Example 20-3 *Display of IKEv2 Profile Settings*

```
R12-DC1-Hub2# show crypto ikev2 profile
IKEv2 profile: DMVPN-IKE-PROFILE-INET
 Ref Count: 1
 Match criteria:
  Fvrf: INET01
  Local address/interface: none
  Identities:
   address 0.0.0.0
  Certificate maps: none
 Local identity: none
 Remote identity: none
 Local authentication method: pre-share
 Remote authentication method(s): pre-share
 EAP options: none
 Keyring: DMVPN-KEYRING-INET
 Trustpoint(s): none
 Lifetime: 86400 seconds
 DPD: disabled
 NAT-keepalive: disabled
 Ivrf: none
 Virtual-template: none
 mode auto: none
 AAA AnyConnect EAP authentication mlist: none
 AAA EAP authentication mlist: none
 AAA Accounting: none
 AAA group authorization: none
 AAA user authorization: none
```

IPsec Transform Set

- The transform set identifies the security protocols (such as ESP) for encrypting traffic. It specifies the protocol ESP or authentication header that is used to authenticate the data:
- **Step 1**. Create the transform set and identify the transforms by using the command **crypto ipsec transform-set** *transform-set-name* [*esp-encryption-name*] [*esp-authentication-name*] [*ah-authentication-name*].
- **Step 2**. Configure the ESP mode by using the command **mode** {**transport** | **tunnel**}.

Example 20-4 provides a sample IPsec transform set.

Example 20-4 *Sample IPsec Transform Set*

```
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
```

The transform set can be verified with the command **show crypto ipsec transform-set**, as shown in Example 20-5.

Example 20-5 *Verification of the IPsec Transform Set*

```
R12-DC1-Hub2# show crypto ipsec transform-set
! Output omitted for brevity
Transform set AES256/SHA/TRANSPORT: { esp-256-aes esp-sha-hmac }
  will negotiate = { Transport, },
```

IPsec Tunnel Protection

IPsec Profile

- The IPsec profile combines the IPsec transform set and the IKEv2 profile:
- **Step 1**. Create the IPsec profile by using the command **crypto ipsec profile** *profile-name*.
- **Step 2**. Specify the transform set by using the command **set transform-set** *transform-set-name.*
- **Step 3**. Specify the IKEv2 profile by using the command **set ikev2-profile** *ike-profile-name*.

Example 20-6 provides a sample IPsec profile configuration.

Example 20-6 *Sample IPsec Profile*

```
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
```

The command **show crypto ipsec profile** displays the components of the IPsec profile, as shown in Example 20-7.

Example 20-7 *Verification of the IPsec Profile*

```
R12-DC1-Hub2# show crypto ipsec profile
! Output omitted for brevity
IPSEC profile DMVPN-IPSEC-PROFILE-INET
          IKEv2 Profile: DMVPN-IKE-PROFILE-INET
          Security association lifetime: 4608000 kilobytes/3600 seconds
          Responder-Only (Y/N): N
          PFS (Y/N): N
          Mixed-mode : Disabled
          Transform sets={
                  AES256/SHA/TRANSPORT: { esp-256-aes esp-sha-hmac } ,
```

Encrypt the Tunnel Interface/IPsec Packet Replay Protection

- When all the required IPsec components have been configured, the IPsec profile is associated to the DMVPN tunnel interface with the command **tunnel protection ipsec profile** *profile-name* [**shared**]. The **shared** keyword is required for routers that terminate multiple encrypted DMVPN tunnels on the same transport interface. The command shares the IPsec security association database (SADB) among multiple DMVPN tunnels.

- Cisco IPsec includes an anti-replay mechanism that prevents intruders from duplicating encrypted packets. A unique sequence number is assigned to each encrypted packet. When a router decrypts the IPsec packets, it keeps track of the packets it has received. The IPsec anti-replay service rejects (discards) duplicate packets or old packets. The router maintains a sequence number window size (default of 64 packets). The minimum sequence number is the highest sequence number for a packet minus the window size. A packet is considered of age when the sequence number is between the minimum sequence number and the highest sequence number.

- The window size is increased globally with the command **crypto ipsec security-association replay window-size** *window-size*. Cisco recommends using the largest window size possible for the platform, which is 1024.

Dead Peer Detection/NAT Keepalives

- Dead Peer Detection (DPD) helps detect the loss of connectivity to a remote IPsec peer. When DPD is enabled in on-demand mode, the two routers check for connectivity only when traffic needs to be sent to the IPsec peer and the peer's active status is not certain. The router sends a DPD R-U-THERE request to query the status of the remote peer. If the remote router does not respond to the R-U-THERE request, the requesting router starts to transmit additional R-U-THERE messages every retry interval for a maximum of five retries. After that, the peer is declared dead. DPD is configured with the command **crypto ikev2 dpd** [*interval-time*] [*retry-time*] **on-demand** in the IKEv2 profile.

- NAT keepalives keep the dynamic NAT mapping alive during a connection between two peers. A NAT keepalive is a UDP packet that contains an unencrypted payload of 1 byte. When DPD is used to detect peer status, NAT keepalives are sent if the IPsec entity has not transmitted or received a packet within a specified time period. NAT keepalives are enabled with the command **crypto isakmp nat keepalive** *seconds*.

IPsec DMVPN Configuration with Pre-Shared Authentication

- Example 20-9 displays the complete configuration to enable IPsec protection on the internet
- DMVPN tunnel on R12, R31, and R41 with all the settings from this section.

R12

```
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
   address 0.0.0.0 0.0.0.0
   pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
!
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
!
crypto ipsec security-association replay window-size 1024
```

R31 and R41

```
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
 dpd 40 5 on-demand
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
!
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

IPsec Tunnel Protection

Verification of Encryption on DMVPN Tunnels

- When the DMVPN tunnels have been configured for IPsec protection, verify the status. The command **show dmvpn detail** provides the relevant IPsec information. Example 20-10 demonstrates the command on R31. The output lists the status of the DMVPN tunnel, the underlay IP addresses, and packet counts. Examining the packet counts can help to verify that network traffic is being transmitted out of a DMVPN tunnel or received on a DMVPN tunnel.

- The command **show crypto ipsec sa** provides additional information that is not included in the output of the command **show dmvpn detail**, such as the path MTU, tunnel mode and replay detection.

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
# Ent Peer NBMA Addr  Peer Tunnel Add State UpDn Tm Attrb Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1  100.64.12.1      192.168.200.12     UP  00:03:39    S   192.168.200.12/32

Crypto Session Details:
--------------------------------------------------------------------------------
Interface: Tunnel200
Session: [0xE7192900]
 Session ID: 1
 IKEv2 SA: local 100.64.31.1/500 remote 100.64.12.1/500 Active
 Capabilities:(none) connid:1 lifetime:23:56:20
 Crypto Session Status: UP-ACTIVE
 fvrf: INET01, Phase1_id: 100.64.12.1
 IPSEC FLOW: permit 47 host 100.64.31.1 host 100.64.12.1
        Active SAs: 2, origin: crypto map
        Inbound: #pkts dec'ed 22 drop 0 life (KB/Sec) 4280994/3380
        Outbound: #pkts enc'ed 20 drop 0 life (KB/Sec) 4280994/3380
 Outbound SPI : 0x35CF62F4, transform : esp-256-aes esp-sha-hmac
   Socket State: Open

Pending DMVPN Sessions:
```

IPsec Tunnel Protection

IKEv2 Protection

- IKEv2 was developed, in part, to protect routers from various IKE intrusion methods. Primarily, it limits the number of packets required to process IKE establishment. During high CPU utilization, a session that has started may not complete because other sessions are consuming limited CPU resources. Problems can occur when the number of expected sessions is different from the number of sessions that can be established. Limiting the number of sessions that can be in negotiation minimizes the CPU resources needed so that the expected number of established sessions can be obtained.

- The command **crypto ikev2 limit** {**max-in-negotiation-sa** *limit* | **max-sa** *limit*} [**outgoing**] limits the number of sessions being established or that are allowed to be established:

  - The **max-sa** keyword limits the total count of SAs that a router can establish under normal conditions. You set the value to double the number of ongoing sessions in order to achieve renegotiation.

  - To limit the number of SAs being negotiated at one time, you can use the **max-in-negotiation-sa** keyword.

  - To protect IKE from half-open sessions, a cookie can be used to validate that sessions are valid IKEv2 sessions and not denial-of-service intrusions. The command **crypto ikev2 cookie-challenge** *challenge-number* defines the threshold of half-open SAs before issuing an IKEv2 cookie challenge.

IPsec Tunnel Protection

IKEv2 Protection (Cont.)

- In Example 20-12, R41 limits the number of SAs to 10, limits the number in negotiation to 6, and sets an IKEv2 cookie challenge for sessions above 4. R41 has 1 static session to the hub router (R11) and is limited to 9 additional sessions that all use the IKEv2 cookie challenge.
- The command **show crypto ikev2 stats** displays the SA restrictions and shows that the four sessions are currently established to the four DMVPN hub routers.

```
R41-Spoke(config)# crypto ikev2 limit max-sa 10
R41-Spoke(config)# crypto ikev2 limit max-in-negotation-sa 6 outgoing
R41-Spoke(config)# crypto ikev2 limit max-in-negotation-sa 6
R41-Spoke(config)# crypto ikev2 cookie-challenge 4
R41-Spoke(config)# end
```

```
R41-Spoke# show crypto ikev2 stats
------------------------------------------------------------------------------
                Crypto IKEv2 SA Statistics
------------------------------------------------------------------------------
System Resource Limit:    0     Max  IKEv2 SAs: 10  Max in nego(in/out): 6/6
Total incoming IKEv2 SA Count: 0           active:    0    negotiating: 0
Total outgoing IKEv2 SA Count: 4           active:    4    negotiating: 0
Incoming IKEv2 Requests: 1      accepted:    1        rejected:     0
Outgoing IKEv2 Requests: 4      accepted:    4        rejected:      0
Rejected IKEv2 Requests: 0     rsrc low:     0        SA limit:      0
IKEv2 packets dropped at dispatch: 0
Incoming IKEV2 Cookie Challenged Requests: 0
               accepted: 0      rejected: 0           rejected no cookie: 0
Total Deleted sessions of Cert Revoked Peers: 0
conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 20

Description	
Data security terms	IPsec transform set
Security associations	Encrypting the tunnel interface
ESP modes	IPsec packet replay protection
IKEv2 keyring	Verification of encryption on DMVPN tunnels
IKEv2 profile	IKEv2 protection

Prepare for the Exam

Key Terms for Chapter 20

Key Terms
Authentication Header (AH) protocol
Encapsulating Security Payload (ESP)
Data confidentiality
Data integrity
Data availability
Origin authentication
Replay detection
Periodic rekey
Security association (SA)

Prepare for the Exam

Command Reference for Chapter 20

Task	Command Syntax
Configure an IKEv2 keyring	**crypto ikev2 keyring** *keyring-name* **peer** *peer-name* **address** *network subnet-mask* **pre-shared-key** *secure-key*
Configure an IKEv2 profile	**crypto ikev2 profile** *ike-profile-name* **match identity remote address** *ip-address* **match fvrf** *{vrf-name | any}* **authentication local pre-share** **authentication remote pre-share** **keyring local** *keyring-name*
Configure an IPsec transform set	**crypto ipsec transform-set** *transform-set-name [esp-encryption-name] [esp-authentication-name] [ah-authentication-name]* **mode** {**transport** | **tunnel**}

Prepare for the Exam

Command Reference for Chapter 20 (Cont.)

Task	Command Syntax
Configure an IPsec profile	**crypto ipsec profile** *profile-name* **set transform-set transform-set-name** **set ikev2-profile** *ike-profile-name*
Encrypt the DMVPN tunnel interface	**tunnel protection ipsec profile profile-name** [**shared**]
Modify the default IPsec replay window size	**crypto ipsec security-association replay window-size** *window-size*
Enable IPsec NAT keepalives	**crypto isakmp nat keepalive** *seconds*
Display the IKEv2 profile	**show crypto ikev2 profile**
Display the IPsec profile	**show crypto ipsec profile**

Slides adapted by
Vladimír Veselý and Matěj Grégr
partially from official course materials

Last update: 2025-04-16