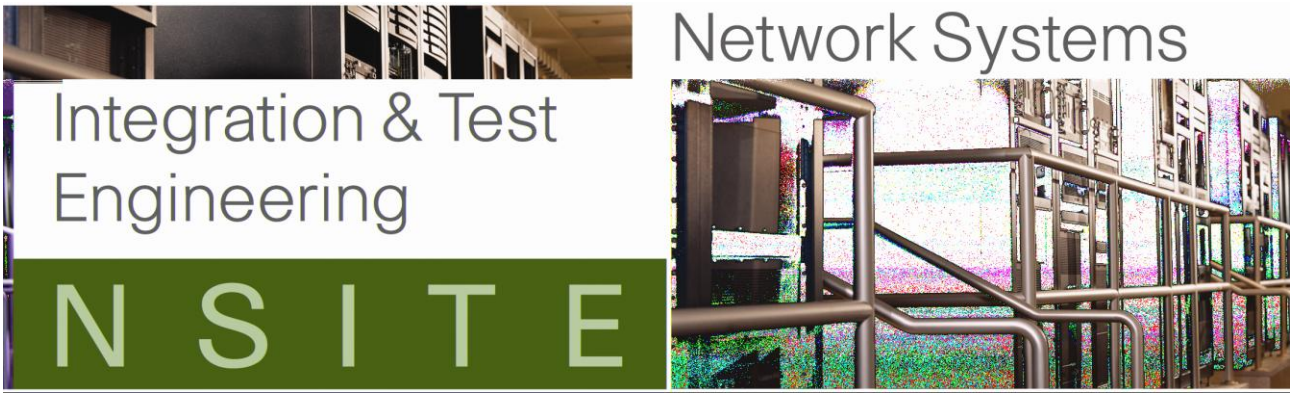


SCALABLE DMVPN
DESIGN AND IMPLEMENTATION GUIDE

Network Systems Integration & Test Engineering (NSITE)

Document Version Number: 1.1

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Abstract

This design and implementation guide (DIG) describes the design and implementation of Dynamic Multipoint virtual private networks (DMVPNs). DMVPN enables hub and spoke network designs in which traffic can securely and inexpensively move through spoke-to-spoke tunnels.

Key Technologies

DMVPN, encryption, generic routing encapsulation (GRE) and multipoint GRE (mGRE), quality of service (QoS)

Target Audience

Enterprise Architecture Council / ESE, Field / Acct Team / SE, NSSTG SPMD, TAC, Cisco Business Units

For more information about NSITE publications, see http://nsite.cisco.com/publications.

Trademark Information

Contents

Figures

Tables

Preface

This design and implementation guide (DIG) defines the comprehensive functional components required to build scalable site-to-site enterprise virtual private network (VPN) systems in the context of wide area network (WAN) connectivity. This guide covers the design and implementation of scalable Dynamic Multipoint Virtual Private Network (DMVPN) solution, using the latest DMVPN features and based on practical design principles that have been tested. This document provides comprehensive explanations of the various design guidelines and best practices.

The use of VPNs to securely interconnect remote branch sites with a central network is a common theme in enterprise deployments. With the increasing demand for secure IP security (IPsec) connections from many remote sites to a central enterprise site (and perhaps between remote sites), the scalability of secure VPN solutions becomes important.

Audience

DMVPN solutions are targeted to new and existing enterprise customers. It is assumed that administrators of a DMVPN solution have experience installing and configuring the products that comprise this solution. In addition, it is assumed that the administrator knows how to upgrade and troubleshoot network systems at a basic level.

Typical users of this DIG include network professionals and systems engineers who want to understand, design, and implement a DMVPN solution. This document provides guidelines and best practices for customer deployments.

Document Scope

This DIG describes scalable DMVPN designs that contain the following components:

- Cisco routers running Internetwork Operating System (IOS)
- Multipoint Generic Routing Encapsulation (mGRE)
- GRE tunneling over IPsec
- Use of various routing protocols to distribute routing information throughout the DMVPN network
- Dynamic spoke-to-spoke tunnels
- Next Hop Routing Protocol (NHRP)
- Tunnel Protection mode
- Quality of service (QoS) features
- Multicast over the DMVPN domain
- Cisco DMVPN scalable designs

The document is limited to DMVPN Phase 3 for all designs and data. This is the most recent phase of DMVPN and is available as of IOS 12.4(6)T. There may be references to DMVPN Phase 2 for comparison purposes, but all the designs, scaling data and configurations in this document are for Phase 3. However, familiarity to previous phases of DMVPN is not required.

Document Objectives

This DIG describes three core scalable DMVPN network designs in detail:

- DMVPN hierarchical hub design
- DMVPN IOS server load balancing (SLB) design
- DMVPN 6500 SLB with crypto offload design

These designs were tested internally at Cisco to identify the scalability of DMVPN Phase 3 in a variety of designs. Scalability test results are incorporated into the design guidance of the Scalable DMVPN deployments. It is recommended that the entire chapters of interest are read since the best practices are defined throughout the chapters.

Furthermore, this document describes spoke deployment options and ideas to show how DMVPN interacts with common features customers would use on a spoke router. These were also tested internally for functionality and completeness.

How to use this document

This document primarily covers three large-scale DMVPN hub designs. It is structured to allow the reader to use it efficiently. Chapter 1 provides a section to help you select the proper deployment for your needs, and direct you to the appropriate chapter or other resources outside this document. The goal is to provide the detailed information necessary to properly deploy a large-scale DMVPN solution. The next section describes the document organization.

Document Organization

This is the basic document organization:

Section	Description
Chapter 1 DMVPN Deployment Overview	A brief overview of the components and basic principles of DMVPN design. This chapter also contains a guide to selecting the right DMVPN deployment for the user's needs.
Chapter 2 Scalable DMVPN design Considerations	A general look into the challenges of designing, scaling, and deploying a DMVPN network.
Chapter 3 DMVPN Hierarchical Hub Design	A multilayered hierarchical DMVPN design, that is perfect for a geographically dispersed deployment.
Chapter 4 DMVPN IOS Server Load Balance Design	A single layered design that is redundant, easy to setup and maintain, and very scalable.
Chapter 5 DMVPN 6500 Crypto Server Load Balance Design	A single layer design that offloads encryption for higher scalability, perfect for high throughput deployments.

Section	Description
Chapter 6 Branch Designs	A look at the branch (spoke) side of the DMVPN solution with support for advanced features on the edge router.
Appendix A DMVPN Configurations	A detailed look at the router configurations for specific designs.

For More Information

For more information about DMVPN, visit:

http://www.cisco.com/en/US/partner/products/ps6658/products_ios_protocol_option_home.html

Acknowledgements

Thanks to Cisco Enterprise Systems Engineering (ESE) for use of their *DMVPN Phase 2 Design Guide*. Some design information appeared previously in this guide.

1 DMVPN Overview

The use of secure virtual private networks (VPNs) to dynamically emulate secure leased lines over the public Internet is common in enterprise networks. An enterprise can securely interconnect remote branch sites with a central site for a fraction of the cost of a frame relay service. Today, a branch site just needs an Internet connection where it can connect a router to build an IP security (IPsec) tunnel to a central site.

Enterprise networks might need to interconnect many remote sites to a main site, and perhaps also to each other, across the Internet while encrypting traffic to protect it. Setting up and paying for hardwired links for internal IP traffic can be time consuming and costly. If all of the sites (including the main site) already have relatively cheap Internet access, this Internet access can also be used for internal IP communication between the stores and headquarters, using IPsec tunnels to ensure privacy and data integrity.

To build large IPsec networks to interconnect sites across the Internet, companies must scale the IPsec network. Typically, IPsec encrypts traffic between two endpoints (peers). The two endpoints negotiate the shared secret key over an ISAKMP phase 1 security association (SA), which would be protected by a pre-shared key or Public Key Infrastructure (PKI). Because secret keys are shared only between the two endpoints, encrypted networks are inherently a collection of point-to-point links. Therefore, IPsec is intrinsically a point-to-point tunnel network.

When scaling a large point-to-point network, it is most efficient to organize it into a hub-and-spoke. This design was used in older Frame Relay networks, because it was prohibitively expensive to pay for links between all sites (full or partial mesh connectivity) in these networks. In most networks, the majority of IP traffic is between the branch site and the main site. Very little IP traffic is sent between individual branch sites. Therefore, a hub-and-spoke design is often the best choice, and DMVPN is inherently a hub-and-spoke setup.

When using the Internet to interconnect the hub and spoke sites, the spokes can directly access each other with no additional cost, which make it more attractive to create full or partial mesh networks. In this case, full or partial mesh networks may be desirable due to the performance improvement on the hub. Spoke-to-spoke traffic traversing the hub uses hub resources and can incur extra delays, especially when using encryption, because the hub must decrypt the incoming packets from the sending spokes and then reencrypt the traffic to send it to the receiving spoke. Another example where direct spoke-to-spoke traffic would also be useful when two spokes are in the same city and the hub is across the country. However, it becomes very difficult, if not impossible, to manually set up and manage a full or partial mesh network among the spokes. DMVPN offers dynamic spoke-to-spoke connections with little additional configuration.

As hub-and-spoke networks, like DMVPN, are deployed and grow in size, it becomes more desirable to use IP routing protocols to distribute the reachability of the private subnets. In older Frame Relay hub-and-spoke networks, this was accomplished by running a dynamic routing protocol such as Open Shortest Path First (OSPF)or Enhanced Interior Gateway Routing Protocol (EIGRP) over the Frame Relay links. This was useful for dynamically advertising the reachability of the private spoke networks, and to support redundancy in the IP routing network. If the network lost a hub router, a backup hub router could automatically take over to retain network connectivity to the spoke networks.

DMVPN uses GRE tunnels in combination with IPsec encryption to securely connect the spokes to the hubs. GRE tunnels support transporting IP multicast and broadcast packets across the GRE tunnel, therefore dynamic IP routing protocols will operate correctly. When GRE tunnels are configured, the IP addresses of the tunnel endpoint (tunnel source and destination) must be known by the other endpoint and must be routable over the Internet.

The Dynamic Multipoint VPN DMVPN solution uses Multipoint GRE (mGRE) and Next Hop Resolution Protocol (NHRP), with IPsec and some new enhancements. The solution is integrated with IPsec, which is triggered immediately for point-to-point and multipoint GRE tunnels. Using mGRE tunnels with IPsec, secure VPN tunnels can be created between the hub and spokes. When the tunnels come up, the branch sites can advertise their private networks to the hub site using any IP routing protocol.

Dynamically establishing secure VPN tunnels to connect remote sites is a very cost effective and efficient way to implement secure connectivity throughout the distributed enterprise network. However, as an enterprise's network grows in the number of branch sites, the scalability of the DMVPN solution must be able to handle the growth. This design and implementation guide (DIG) provides the information needed to properly deploy a large-scale DMVPN network.

1.1 Starting Assumptions

This document focuses on DMVPN Phase 3 technology. Readers should have a basic understanding of the following topics:

- VPN concepts
- IPsec and encryption
- mGRE tunnels
- Routing protocols
- Server load balancing (SLB)

For more information about DMVPN, go to:
http://www.cisco.com/en/US/products/ps6658/products_ios_protocol_option_home.html

1.2 DMVPN Design Components

VPNs provide an alternative to traditional WAN technologies such as leased lines, Frame Relay, and ATM. VPNs enable private WANs over a public transport, such as the Internet. LAN-to-LAN VPNs are primarily deployed to connect branch offices and home offices to the central site (or sites) of an enterprise.

The requirements that enterprise customers have for traditional private WAN services such as multiprotocol support, high availability, scalability, and security are also VPN requirements. VPNs can often meet these requirements more cost-effectively and with greater flexibility than private WAN services, like Frame-Relay and ATM.

The following are key components of this DMVPN design:

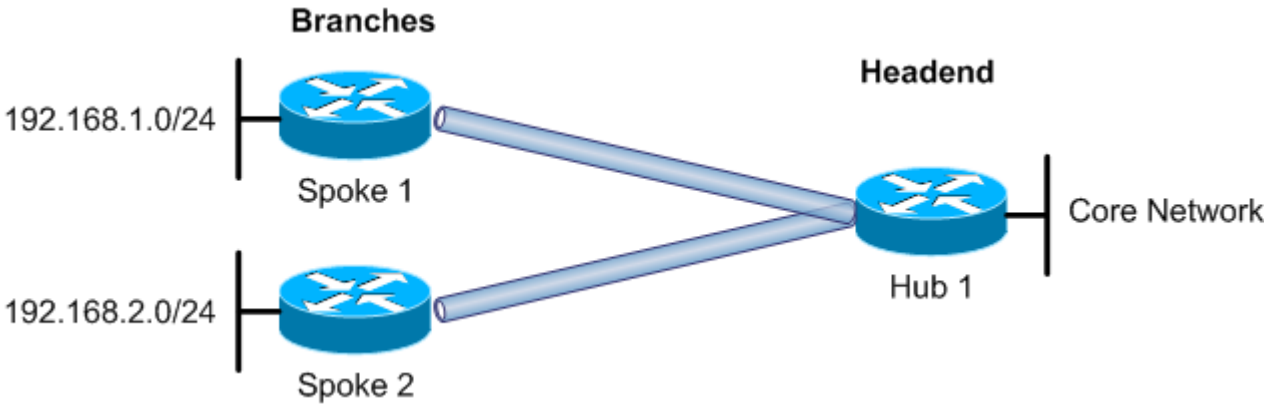
- Cisco high-end VPN routers serving as VPN termination devices at a central enterprise campus (headend or hub devices)
- Cisco VPN access routers serving as VPN branch termination devices at branch office locations (branch or spoke devices)
- DMVPN hub-and-spoke to perform headend-to-branch interconnections
- DMVPN spoke-to-spoke to perform branch-to-branch interconnections (optional)

- Internet services procured from a third-party ISP (or ISPs) serving as the WAN interconnection medium

Cisco VPN routers are a good choice for DMVPN deployments because they can accommodate any network requirement traditionally provided by Frame Relay or private line networks. These requirements include support for multicast, latency-sensitive traffic, and routing protocols. The VPN routers must have hardware encryption modules; software encryption performance cannot support these designs.

Note: The terms "hub" and "headend" are used interchangeably, as are the terms "spoke" and "branch." Typically, hub and spoke are used when describing the design from a protocol or functionality point of view. The terms headend and branch are used more often when describing the high level functions.

Figure 1-1. Basic Hub-and-Spoke Network Using Tunnels



1.3 DMVPN Phase 3

This section briefly covers key differences between DMVPN Phase 3 and DMVPN Phase 2. The major differences are summarization, and how spoke-to-spoke tunnels are formed. DMVPN Phase 3 is used for all designs described in this document. Scalability numbers in this document are based on DMVPN Phase 3, and may be different than the scalability using DMVPN Phase 2 due to the improvements described next.

For more detailed information on DMVPN Phase 2, see the DMVPN guide located here:

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a008075ea98.pdf

1.3.1 Summarization

DMVPN Phase 3 supports the summarization of spoke local network routes when the hub advertises these routes back out to the spokes. The spoke routers do not require full individual network routing knowledge of what networks are behind which remote spoke. Phase 2 DMVPN required every spoke to have full individual network routing knowledge.

1.3.2 Hub Network Design

The hub design operates slightly differently in DMVPN Phase 3 compared to DMVPN Phase 2. In Phase 2, hubs were limited to a daisy-chaining, single-layer method of interconnecting with each other. This required NHRP resolution messages to be cycled through the chain of hubs before returning to the spoke.

In Phase 3, the NHRP table and the Cisco Express Forwarding (CEF) table work together to efficiently route messages. The NHRP database is used to resolve next hop information, and the CEF table is used to route packets. The Hubs in a DMVPN Phase 3 network may be interconnected in many different ways, (full mesh, partial mesh, and hierarchical).

Note: DMVPN Phase 2 and Phase 3 spokes cannot coexist in the same DMVPN network.

1.3.3 Packet Flow Spoke to Spoke

When using a summarized route, the spoke router forwards packets destined to a subnet behind a remote spoke to the hub router indicated as the next hop server. The hub router then forwards the packet to its destination based on the hub routing table. Because this packet entered and exited the hub router over the same mGRE interface, the hub router sends an NHRP redirect to the previous GRE hop (the originating spoke router). This triggers the originating spoke router to send an NHRP resolution request for the destination IP address of the data packet that triggered the NHRP redirect.

The NHRP resolution request is forwarded along the known routed path (to the hub router). The hub router then forwards the NHRP resolution request toward the spoke router that services the destination IP address/subnet. The terminating spoke router then builds an mGRE crypto tunnel back to the originating spoke router, using information in the NHRP resolution request, and sends the answer (NHRP resolution reply) to the local spoke over this direct spoke-to-spoke crypto tunnel. After the local spoke receives the NHRP resolution, data traffic is forwarded over this spoke-to-spoke tunnel.

1.4 Chosing A Scalable DMVPN Hub Design

In a DMVPN topology, the most important part of the design is the hub site. The hub is where mGRE, routing, NHRP, and IPsec functionality come together. Therefore, the scalability of the entire topology is limited by the scalability of the hub site.

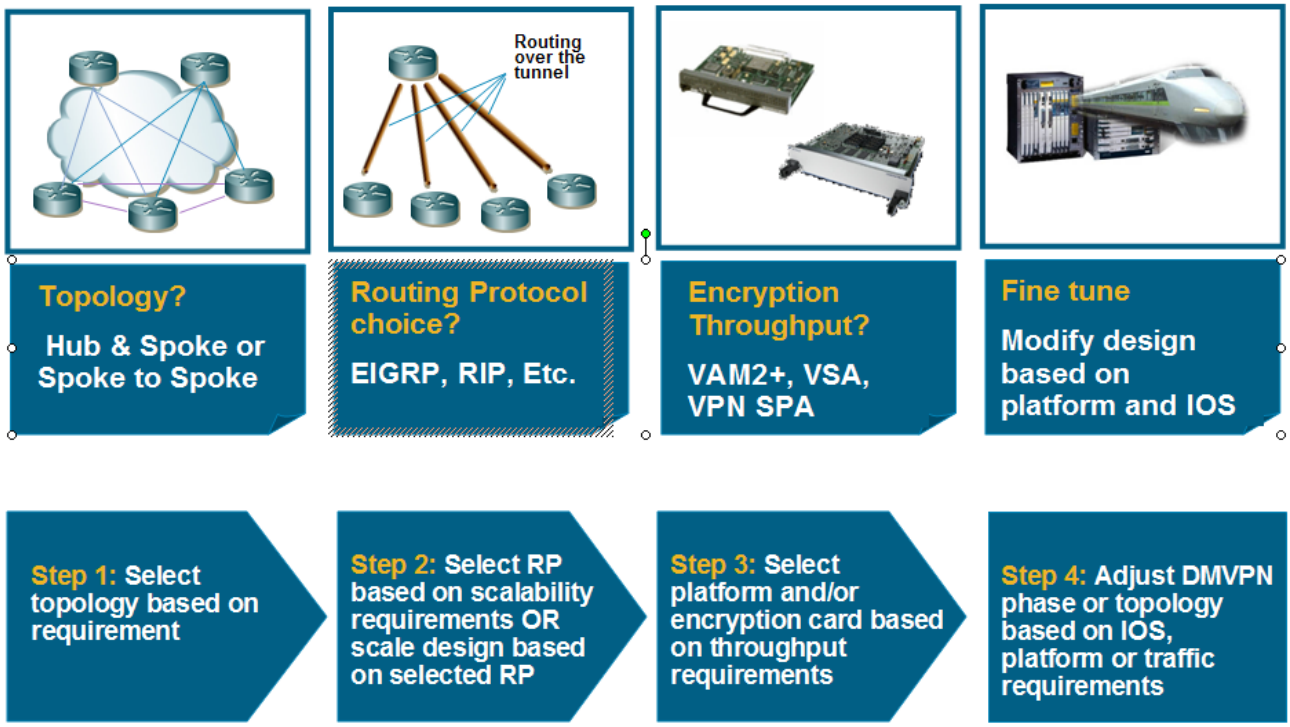
This document covers three large-scale DMVPN hub designs that Cisco tested .Each design serves different purposes and has different scalability. However, scalability is not bounded by the design itself. Instead, the routing protocol used can further limit the number of spokes that a single hub router can support in a given design. Therefore, routing protocols are an important focus when we discuss each scalable DMVPN design in detail. The designs in this guide are described using DMVPN Phase 3 because of the performance enhancements built into the most recent phase of the solution.

The next part of the decision process is choosing the DMVPN deloyment. The reader should understand the basics of DMVPN and have deployment goals in place. The following section describes the steps of selecting a DMVPN design. Depending on customer depoyment requirements, the reader might be directed to another document that covers lower scale DMVPN deployments.

1.4.1 Steps for Selecting a DMVPN Design

Figure 1-2 displays the basic flow chart to selecting the correct design for your needs.

Figure 1-2. DMVPN Design Selection Criteria



1. Select a Topology

There are two basic topologies that can be used in DMVPN networks

- Hub-and-spoke
- Spoke-to-spoke

Each topology provides a resilient version of the topology.

Basic Hub-and–Spoke Topology

Figure 1-3 illustrates the basic hub-and-spoke topology. The spokes are configured to communicate *only* with the hub router. All communication with other spokes must transit the hub router. The routing protocol can control routes to block traffic.

Figure 1-3. Basic Hub-and-Spoke Topology

Resilient Hub-and-Spoke Topology

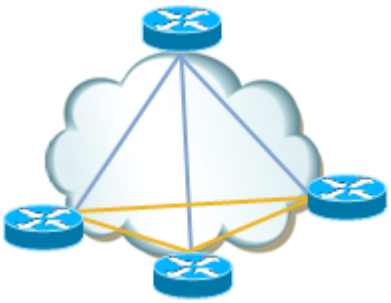
Figure 1-4 illustrates the resilient hub-and-spoke topology. All features of the basic hub and spoke design apply to this topology. However, the spokes connect to two or more hubs for resiliency. This is known as a dual-cloud DMVPN design. Based on routing, traffic can be distributed to both hubs or can always be sent to a primary hub.

Figure 1-4. Resilient Hub-and-Spoke Topology

Basic Spoke-to-Spoke Topology

Figure 1-5 illustrates the basic spoke-to-spoke topology. This topology requires the basic setup of the basic hub and spoke topology. In this topology, spokes are configured to communicate to the hub as in hub-and-spoke. However, communication between spokes triggers dynamic tunnels to be formed directly between the spokes. After a tunnel is formed, direct spoke-to-spoke tunnels transport unicast traffic between the two spokes, thereby reducing load on the hub.

Figure 1-5. Basic Spoke to Spoke Topology



Resilient Spoke-to-Spoke Topology

Figure 1-6 illustrates the resilient spoke-to-spoke topology. This topology requires the basic setup of the resilient hub and spoke topology. In this topology, all the features of basic spoke to spoke design apply. However the spokes connect to two or more hubs for resiliency. Based on routing and/or NHRP configurations, traffic can be distributed over both hubs.

Figure 1-6. Resilient Spoke-to-Spoke Topology

Note: Resilient forms of the topologies are recommended to eliminate single points of failure. These topologies add a geographic redundancy to the DMVPN deployment. Certain hub site designs offer local redundancy, but that does not necessarily resolve the single point of failure.

2. Select a Routing Protocol

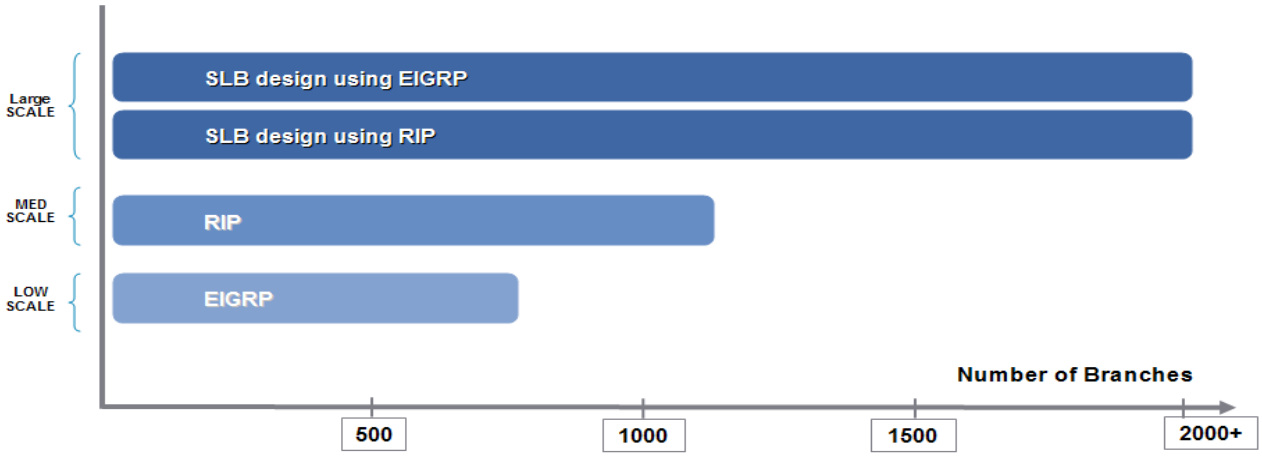
The routing protocol is an important determining factor for DMVPN deployments. You can choose the routing protocol in one of two ways. This document addresses only EIGRP and RIP, which are preferred protocols for large-scale DMVPN networks.

Other routing protocols, such as OSPF and On-Demand Routing (ODR), are outside the scope of this document because of inherent limitations and configuration complexity when used in large DMVPN networks. OSPF works well when using multiple areas with a smaller number of branches. Large DMVPN networks do not fit this paradigm because the DMVPN cloud creates a single large subnet requiring a single OSPF area, which causes scalability to suffer.

ODR would fit the DMVPN model well. However, Cisco Discovery Protocol (CDP) advertises all connected networks over the tunnel. This causes issues of recursive routing when public addresses are learned over the tunnel. The solution would be to configure route filters for each spoke, which works well in a new deployment with contiguous address LAN space so that all the LAN networks can be summarized to a few filter entries. Such a configuration is very cumbersome on an existing network, where LAN addresses cannot be easily summarized, and therefore does not scale well from a configuration point of view.

Figure 1-7 shows the relationship between the protocols and how well it scales for a single hub router. Also shown is the relative scalability for Server Load Balanced (SLB) DMVPN hub site designs, which use multiple routers. The relative scalability is a multiple of the single router scale limit for a given platform. The scalability can be increased incrementally for the SLB hub site designs.

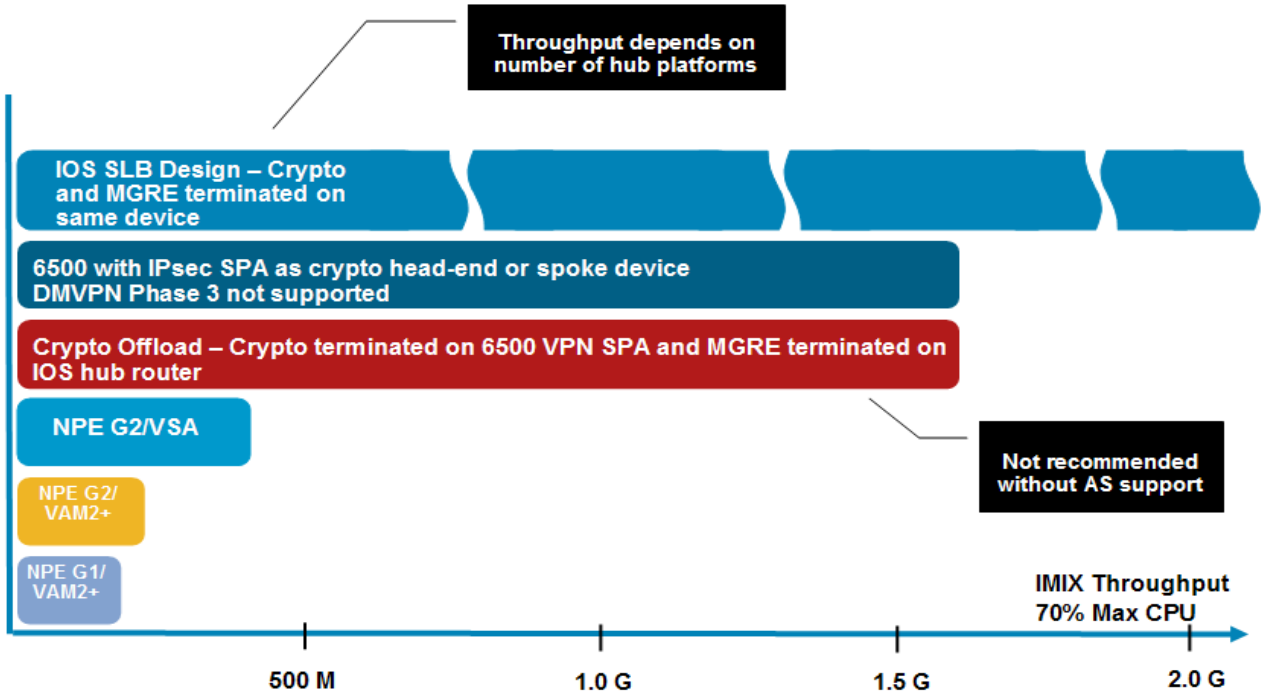
Figure 1-7. DMVPN Routing Protocol Scaling Chart



3. Select Hardware Platform and/or Encryption Modules

This section provides recommendation for hardware platform and encryption modules based on maximum throughput. Typically the hub router is the bottleneck, and should be sized sufficiently to support the anticipated throughput across the DMVPN tunnels. See the chart below to see the performance of the platforms and modules.

Figure 1-8. DMVPN Throughput Scaling Chart



Note: Throughput numbers alone should not be used to size your deployment. Routing protocol traffic, quality of service (QoS), multicast, and other factors may impact the potential throughput of a given platform or encryption card. This data is based the platform encrypting and forwarding an Internet mix of unicast traffic with a CPU utilization at 70%. This is the standard used throughout this document.

4. Make Final Design Choices and Adjustments

Based on the choice of topology, routing protocol, and hardware, you can choose which chapter of this document you need to reference, or perhaps another document is right for you.

The version of IOS that your routers must run depends on the topology you chose, and on the headend design you require.

1. Hub-and-spoke design will work the same in mainline or T train. Select a stable, well-tested release such as 12.4(15)T4, that is recommended for the designs in this document. Remember that spoke-to-spoke traffic (if allowed by the hub) will traverse the hub, so the throughput across your hub might be higher and the hardware performance should be sufficient to meet this requirement.

2. Spoke-to-spoke design using DMVPN Phase 2 requires IOS 12.4 Mainline, 12.2SX, or pre-12.4(6)T code or 6500/7600 supporting DMVPN natively. This means you will use the DMVPN Phase 2 spoke-to-spoke design. Multiple hubs must be "daisy chained." The hubs cannot summarize routes when advertising to the spokes, but must advertise all routes and the next-hop should be unchanged.

   For the preceding design choices, refer to the following DMVPN guide on Cisco.com:

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a008075ea98.pdf

3. Spoke-to-spoke design using DMVPN Phase 3 requires IOS 12.4(6)T or later. This means you will use the spoke-to-spoke design, requiring the NHRP redirect and shortcut commands discussed in chapter 2. Route summarization to the spokes is supported. This reduces the number of prefixes a hub must send to the spokes. With improvements in NHRP, hierarchical hub designs are possible for better scalability.

This document describes spoke-to-spoke capable designs using DMVPN Phase 3. These designs also work for smaller deployments, but they have the extensibility to be used for large-scale. The hub site designs in this document can emulate any of the topologies described previously.

1. If the DMVPN deployment will be hierarchical (there will be a hierarchy of DMVPN hubs) you will want to use the hierarchical hub design described in Chapter 3. These are typically used where there is large geographic separation between regional hubs and DMVPN clouds.

2. For a DMVPN hub site to scale beyond the limits of one hub router, server load balancing (SLB) designs can be used for incremental scalability and redundancy. There are two flavors of SLB hub designs to chose from:

   a. IOS SLB distributes performance across the DMVPN hubs. The hub routers in the server farm perform all crypto, NHRP, and GRE processing, so the hardware in the server farm dictates performance. This design has many virtues, and is discussed in Chapter 4.

   b. 6500 Crypto SLB is positioned in the same way as the IOS SLB hub design, but is typically used with designs having high throughput requirements. This design uses the 6500 VPN SPA to offload crypto processing from the hub routers, but still uses the server farm routers to handle NHRP and GRE processing. This design is discussed in Chapter 5. However, because of its complexity, this design is recommended only with Cisco Advanced Services support.

1.5 Spoke Designs

Along with hub designs, this document describes some DMVPN spoke designs, including the configuration of common spoke features such as QoS, Firewall, intrusion detection, and so on. The use of specific spoke features depends on deployment requirements. These designs and features are discussed in Chapter 6.

1.6 Cisco Unified Communications Voice in DMVPN Networks

A few concerns arise when using Cisco Unified Communications Voice over IP (VoIP) in spoke-to-spoke networks. One concern is related to voice quality because of latency differences between the spoke-hub-spoke path and the spoke-to-spoke path when the spoke-to-spoke tunnel is initially established. Another concern is the point-to-multipoint nature of the tunnel interfaces on the spoke routers, which creates the possibility that a spoke router access link (or associated low-latency queue) can be overwhelmed with traffic from multiple sources. These concerns and other information regarding VoIP in a DMVPN network can be found in this document:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6635/ps6658/prod_white_paper0900aecd80652a9d.html

2 Scalable DMVPN Design Considerations

This document focuses on three DMVPN core topology designs and many branch designs. Each deployment model can include a variety of IOS features to enhance the design and meet the needs of a particular enterprise.

We will now look at some common themes of large scale DMVPN deployments. There are many aspects to consider when designing a DMVPN network that can be scaled as the needs of the network increase. The most common issue (if not the most important) is the handling of the routing information and Next Hop Resolution Protocol NHRP tables as the scale increases. Other considerations might be aggregate throughput and data encryption as the network grows. We must consider the processing power needed to encrypt and decrypt data traffic. Network redundancy and ease of expandability should also be considered for robustness of the deployment.

A DMVPN "headend" refers to a central hub router that terminates IPsec tunnels from the branch or spoke routers. The headend site is typically the central site or a major geographic location in the enterprise network. The headend would typically have one or more high bandwidth uplinks (T3, OC3, and so on). The branch or spoke side is typically be a smaller office or home office. These would also have slower uplinks to the Internet, such as cable, DSL, or T1.

DMVPN has three control planes that work together: the IPsec control plane, the Generic Routing Encapsulation (GRE) and NHRP control plane, and the routing control plane. The interaction of these components must be considered when scaling DMVPN.

2.1 DMVPN Topology

DMVPN is typically deployed in a hub-and-spoke model. This model is the most scalable, and behaves much like traditional Frame-Relay or leased line networks. In fact, the multipoint GRE (mGRE) tunnels are viewed as a nonbroadcast multiaccess (NBMA) network because of their multipoint nature. The hub site topology is the main focus of this section since the spoke site topology is usually relatively simple. Depending on the scalable DMVPN design being implemented, one should consider whether the design requires expandability, redundancy, or both. This might affect the addressing scheme used on the tunnels, and consequently the routing protocol configuration.

In its simplest form, a DMVPN topology comprises a single cloud. This means that all spokes connecting to a hub are configured in the same subnet. You can have multiple DMVPN regions (geographical or logical), each with a hub router, and a set of spokes all in the same region. This is still considered one DMVPN cloud, and will be looked at in the subsequent chapters. The important point to note is that they all share the same subnet even if they are not connected to the same hub.

It is also possible to have multiple clouds connected to a DMVPN router. The router is then configured with one mGRE tunnel interface per cloud, so a dual cloud setup requires two tunnel interfaces. You can also use two separate routers, each with one tunnel interface. The end result is the same. However, each tunnel interface is configured in a different IP subnet. In this case, the NBMA networks do not share the subnet, and cannot communicate without routing through a hub router. Understanding these concepts of DMVPN clouds is necessary to understand DMVPN designs and redundancy features. In this document, the terms "DMVPN cloud" and "DMVPN network" are used interchangably, and primarily refer to all DMVPN tunnels that exist within the same subnet.

All of the designs are expandable. The expandability of the hub site typically entails server load balancing (SLB), which inherently offers redundancy. This enables more physical hubs to be added to the server farm with relative ease and no down time. It is possible to add SLB to the hierarchical hub model; however you can also add more physical hubs and connect them to the core hub in the hierarchy. Of course, you do not need to use SLB to add hubs. You can manually distribute spokes across hubs, or change the distribution of spokes per region. This will become clearer when the core designs are discussed.

Each of the scalable DMVPN designs can provide redundancy. If using a redundant model, the branch router can have multiple DMVPN clouds, where each tunnel interface is a separate tunnel to different headends. These headend routers can be geographically separated or colocated. Alternatively, the spokes can simply point to multiple headends on one mGRE interface using a single DMVPN cloud.

In this document, we discuss the single and dual cloud DMVPN topologies. Cisco recommends a single cloud topology for hierarchical hub designs, and a dual cloud topology for SLB designs. A dual cloud topology provides network managers with greater control over path selection than does a single topology. However, in the dual cloud topology you often get dual spoke-to-spoke tunnels between the same two spokes (one over each cloud), which requires twice the resources (mostly, just memory to store the two encryption instances). The primary dual cloud failover method is a dynamic routing protocol, but a single cloud topology relies on the routing protocol, and on NHRP handling failure events.

2.2 IP Addressing

There are multiple aspects of IP addressing to consider. In this section, we do not consider public IP addressing because this is typically assigned by a service provider (SP). We can use public addresses in DMVPN networks, but it is more common to use private addressing on the tunnels and internal private subnets.

Note: Private addressing in this document refers to the address space an enterprise does not advertise to the Internet. This can be RFC1918 addresses or an IANA assigned subnet.

The subnet used on the tunnel (mGRE) interfaces is typically a private address because it is encrypted inside the GRE packet and these addresses are typically not referenced outside the DMVPN network. Depending on how the DMVPN clouds are set up, you will need one subnet per cloud. So, if you have two mGRE tunnel interfaces, you must address each in a separate subnet corresponding to the rest of the cloud.

The next major addressing consideration is that of the private subnets connected to the DMVPN spokes being advertised over the DMVPN tunnel to the hubs. This requires careful planning so that the routing protocols running on the hub site can summarize the routes effectively. Addressing of the spoke internal subnets should lend itself to easy summarization.

The power of DMVPN phase 3 is in the ability to use the routing protocol to send summarized routes to other spoke devices. In a given region of spokes, the private LAN subnet routes will be advertised by the spokes to their hubs and between these hubs. These hubs in turn will advertise summaries of these routes to the spokes of their region and may advertise either individual routes or the summarized routes to central hubs or hubs in other regions.

Cisco recommends using proper address summarization at the hub routers and core network, which accomplishes the following:

- Conserves router resources, making routing table sizes smaller
- Reduces the load and convergence time for the routing protocol

- Saves memory in the routers and eases troubleshooting tasks
- Simplifies the configuration of routers in IPsec networks

2.3 mGRE Interfaces

Although IPsec provides a secure method for tunneling data across an IP network, it has several limitations. First, IPsec does not support broadcast or IP multicast, preventing the use of protocols that rely on these features, such as routing protocols.

GRE is a protocol that can be used to "carry" other passenger protocols such as IP, including broadcast or multicast IP packets. Using GRE tunnels in conjunction with IPsec provides the ability to run dynamic routing protocols or IP multicast across the network between headends and branch offices.

In the point to point (p2p) GRE over IPsec solution, all traffic between sites is encapsulated in p2p GRE packets before encryption, simplifying the proxy that classifies traffic for encryption. The **`crypto map`** statements need only one line permitting GRE (IP Protocol 47). However, in p2p GRE over IPsec, the headend router requires a unique tunnel interface for each branch router, so a large-scale design can have a very large IOS configuration file on the headend router, and does not scale from a management perspective.

DMVPN introduces an mGRE interface, which serves as a "one-to-many" interface for the creation of multiple hub-and-spoke tunnels that are similarly to point-to-multipoint Frame Relay interfaces. Unlike p2p GRE tunnels, V tunnel destination is not configured. In all DMVPN designs, the headend is configured with an mGRE interface to enable the dynamic tunnel creation for each connected branch. The mGRE interface reduces the configuration file on each headend router, which is an advantage for large-scale designs when compared to static p2p GRE topologies. Note that even with an mGRE interface, there are still individual tunnel instances, one for each connected remote spoke. The advantage is that these tunnel instances are dynamically configured, compared to statically configured p2p GRE tunnel interfaces.

The protocol header for an mGRE packet is four bytes larger than a p2p GRE packet. The additional four bytes constitute a tunnel key value, which is used to differentiate between mGRE interfaces in the same router. Without a tunnel key, a router can support only one mGRE interface, corresponding to one IP network. Tunnel keys enable a branch router to have a different mGRE interface corresponding to each DMVPN cloud in the network topology. A headend router can also be configured with two mGRE interfaces pointing to each DMVPN cloud for high availability and redundancy. Cisco IOS Software Releases 12.3(13)T, 12.3(11)T3, or later support the configuration of multiple mGRE interfaces on a single router without tunnel keys. In this case, each mGRE interface must reference a unique IP address as its tunnel source.

2.4 NHRP

NHRP, defined in RFC 2332, is a Layer 2 (L2) address resolution protocol and cache, similar to Address Resolution Protocol (ARP) and Frame Relay Inverse-ARP. Branch routers connected to NBMA subnetworks use NHRP to determine the IP address of the "NBMA next hop" (tunnel destination): in this case, the headend router or the destination IP address of another branch router.

When a branch router is first established onto a DMVPN network, the router registers its IP address with the headend router whose IP address NHRP mapping is already preconfigured on the branch router. This registration enables the mGRE interface on the headend router to forward packets through the dynamic tunnel back to the registering branch router without having to know the branch tunnel destination through

CLI configuration. NHRP maps a tunnel IP address to an NBMA IP address. NHRP tells the mGRE interface where to tunnel a packet to reach a certain address. When the data IP packet is encapsulated in an mGRE/IP header, the GRE/IP destination address is the NBMA address.

If the destination address is connected to the NBMA subnetwork, the NHRP router is the destination itself. Otherwise, the NHRP router is the egress router closest to the branch requesting a destination IP address.

Headend and branch routers should be configured with an NHRP holdtime, which sets the length of time that routers instruct other routers to keep any NHRP information that they provide. This information is kept in the NHRP cache until the NHRP holdtime expires, when the information is removed and must be then be relearned. The default NHRP holdtime is two hours; however, the recommended value is between five and ten minutes (300–600 seconds). The NHRP cache can be populated with either static or dynamic entries. On the headend router, all entries are added dynamically through registration or resolution requests.

The branch router is configured with a static NHRP map pointing to the headend router. Branch routers must be configured with the NBMA address of the headend router as their next hop server (NHS) to register with the headend router. This enables the branch router to initially connect into the NHRP network. The branch routers send a registration to the headend router that contains their tunnel IP address and NBMA address. The headend router creates an entry with this information in its NHRP cache and returns a registration reply. The branch router now views the headend router as a valid NHS and uses it as a server to locate any other branches and networks in the NHRP domain.

The NHRP network-ID is a local parameter only, and is not transmitted in NHRP packets to other NHRP nodes. The NHRP network-ID defines the NHRP domain for a GRE interface and differentiates between multiple NHRP domains or networks, when two or more NHRP domains (GRE interfaces) are available on the same NHRP node (router). Think of the NHRP network-ID as being used to help separate two DMVPN networks (clouds) when both are configured on the same router. For this reason, the actual value of the NHRP network-ID configured on a router does not have to match the same NHRP network-ID on another router where both routers are in the same NHRP domain (DMVPN network). As NHRP packets arrive on a GRE interface, the NHRP network-ID configured on that interface assigns the packets to the local NHRP domain. It is recommended to use the same NHRP network-ID on all GRE interfaces in the same DMVPN network. It is then easier to track which GRE interfaces are members of which DMVPN network.

2.5 Crypto Considerations

Even though IPsec supports transport and tunnel encryption modes, you should configure your DMVPN Phase 3 network with transport mode. Transport mode encrypts only the data portion (payload) of each GRE/IP packet, leaving the GRE/IP source and destination address in the header untouched. This mode requires less overhead than tunnel mode, and it works even if there is a NAT device between the hub and spoke, where tunnel mode does not work.

2.6 Tunnel Protection Mode

Tunnel protection is used to secure (encrypt) the data transfer inside the GRE tunnel. This is done by applying an IPsec profile to the mGRE tunnel interface. Crypto maps are unnecessary in IOS Release 12.2(13)T or later. IPsec profiles are used to accomplish the same result and share most of the same commands with the crypto map configuration. However, only a subset of the commands is needed in an IPsec profile. Only commands that pertain to an IPsec policy can be used under an IPsec profile. There is no need to specify the IPsec peer address or the ACL to match the packets that are to be encrypted. If a packet is routed through the tunnel, it is encrypted.

To associate either a p2p GRE or an mGRE tunnel with an IPsec profile on the same router, tunnel protection must be configured. Tunnel protection specifies that IPsec encryption is performed after the GRE headers are added to the tunnel packet. In p2p GRE tunnels, the tunnel destination IP address is used as the IPsec peer address. In mGRE tunnels, multiple IPsec peers are possible; the corresponding NHRP-mapped NBMA destination addresses are used as the IPsec peer address.

If more than one mGRE tunnel interface or an mGRE and p2pGRE tunnel interface is configured on a router, and they use the same tunnel source address, the **`shared`** keyword must be configured on the tunnel protection command. Each mGRE tunnel interface still requires a unique tunnel key, NHRP network-ID, and IP subnet address. This is common on a branch router when a dual DMVPN cloud topology is deployed. If a tunnel key is not configured, all mGRE tunnel interfaces must have unique tunnel source addresses. In this same case, p2p GRE tunnel interfaces can have the same tunnel source address as long as the interfaces have different tunnel destination addresses.

2.7 Internet Key Exchange (IKE) Call Admission Control (CAC)

The DMVPN hub router terminates the crypto sessions from multiple spokes. It is important to control the number and rate of simultaneous Internet Security Association and Key Management Protocol (ISAKMP) security association (SA) requests received by IKE. The router can become overloaded if more incoming ISAKMP SAs are initiated than the processor can handle. These capabilities are platform-specific. If the processor becomes overcommitted, IKE negotiation failures and the constant retransmissions of IKE packets can further degrade router performance. This typically happens if a failure causes most or all spokes to lose the crypto peering to the hub, resulting in many spoke devices initiating crypto as soon as the failure condition is fixed.

IKE CAC was introduced in IOS Release 12.3(8)T to limit the number of ISAKMP SAs permitted on a router. By limiting the amount of dynamic crypto peers that can be created, you can prevent the router from being overwhelmed if it is suddenly inundated with ISKAMP SA requests. The ideal limit depends on the particular platform, the network topology, the application, and traffic patterns. When the specified limit is reached, IKE CAC rejects all new ISAKMP SA requests. If you specify an IKE CAC limit that is less than the current number of active IKE SAs, a warning is displayed, but ISAKMP SAs are not terminated. New ISAKMP SA requests are rejected until the active ISAKMP SA count is below the configured limit.

Two CAC implementations for limiting IKE SAs can benefit a DMVPN implementation. First, the normal CAC feature is a global resource monitor that is polled to ensure that all processes, including IKE, do not overrun router CPU or memory buffers. The user can configure a resource limit, represented by a percentage of system resources from 0 to 100. If the user specifies a resource limit of 80 percent, then IKE CAC drops ISAKMP SA requests when 80 percent of the system resources are being consumed. This enables the router to finish processing the current set of ISAKMP SAs (a CPU intensive activity) when there is not sufficient CPU and memory resources to processing additional ISAKMP SAs. If the number of ISAKMP SAs being processed is not limited, the router can be overwhelmed to the point that it cannot be able to complete the processing of any of the ISAKMP SAs

Note: Once an ISAKMP SA goes from active to idle state, that SA consumes very few CPU resources. This feature is valuable on headend routers that must, by the design of the network, handle many (often many hundreds) of spokes (encryption connections). It is less useful on branch routers, because branch routers typically never need to process many ISAKMP SAs at the same time.

The second approach enables a user to configure an overall limit of both active and idle ISAKMP SAs (IKE CAC). When this limit is reached, IKE CAC drops all new ISAKMP SA requests. IPsec SA rekey requests are always allowed, because the intent is to preserve the integrity of existing sessions. This functionality is primarily targeted at branch routers in a spoke-to-spoke deployment model. Configuring a limit to the amount of dynamic tunnels that can be created to the device prevents a router from overextending itself with the number of ISAKMP SAs (DMVPN spoke-to-spoke tunnels). For example, an 871 router might be able to handle 15 encrypted tunnels, but be a member of a 1000 node DMVPN network. You would limit the ISAKMP SAs to 15 so that this node would not attempt to build, for example, 30 spoke-to-spoke tunnels and run out of memory. The ideal IKE CAC limit depends heavily on the particular platform and crypto engine (CE), the network topology, and feature set being deployed.

2.8 Routing Protocols with DMVPN

All scalable DMVPN designs described in this DIG recommend the use of a dynamic routing protocol to propagate routes from the branch offices to the headend and back to the branch offices. Using a routing protocol has several advantages in DMVPN Phase 3.

In a VPN, routing protocols provide the same when compared to traditional networks, which include:

- Network topology information
- Topology change notification (such as when a link fails)
- Remote peer status

Several routing protocols can be used in a DMVPN design, including:

- Enhanced Interior Gateway Protocol (EIGRP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol version 2 (RIPv2)
- Border Gateway Protocol (BGP)
- On-Demand Routing (ODR)

EIGRP is the Cisco recommended dynamic routing protocol because of its conservation of router CPU cycles and network bandwidth, and its quick convergence times. EIGRP also provides a range of options for address summarization and default route propagation.

Note: Other routing protocols such as OSPF, BGP, and ODR have been verified in the past, but were not tested for Scalable DMVPN network topologies, and are not discussed in great detail. The spoke-to-spoke deployment model is typically done using split tunneling on the branch device. In a nonsplit-tunneling branch, this can still be achieved with VRFs. ODR does not support native split tunneling, so using VRFs to achieve split tunneling is required when using ODR (non-split-tunneling) and dynamic spoke-spoke tunnels are required..

Routing protocols provide the only way to exchange routing information in large scale DMVPN networks. The drawback is the increase in CPU utilization on the network device, so this impact must be considered when designing the DMVPN deployment.

Note: The routing protocol is usually the main limiting factor for how the number of spoke routers that a single hub router can support. The second limiting factor is the amount of encryption throughput available on the hub router. This encryption throughput is divided by the number of supported spoke routers to give an average per-spoke value. This value must be sufficient to handle expected load of spoke routers accessing

resources behind the hub router. There is often a percentage of oversubscription, because usually all spokes are not active at the same time.

2.8.1 Route Propagation Strategy

When a branch connection to the network comes up, the branch router is ready to begin transmitting routing protocol information because it has a static NHRP entry to the headend router. Because the headend router must wait for the branch router to populate the NHRP cache, the headend router cannot begin sending routing protocol information until the branch registers its NBMA address with the next hop server (NHS).

After the spoke registers with the NHS and the tunnel is operational, the routing protocol can run across the tunnel. With the ability to summarize routes in DMVPN Phase 3, hubs should be configured to summarize the branch and core network routes sent to the spokes. Each spoke advertises its private subnets to the headend, summarized if possible. The headend readvertises these private-subnets, further summarized across all spoke advertised networks and back out to the all the other spokes in the network. The hub router also advertises its own private networks to the spokes.

2.8.2 EIGRP

EIGRP is a good choice for use with DMVPN because it conserves router CPU cycles and network bandwidth, and has quick convergence times. Because EIGRP is a distance-vector type routing protocol, it matches well with the NBMA style network that DMVPN builds. EIGRP also provides several options for address summarization and default route propagation and few restrictions for where and how these options can be used. When using EIGRP with DMVPN there are two related features that should be considered depending on the overall requirements of your routing.

Because we are using DMVPN Phase 3, we can send summary routes to all spokes. We do not require turning off EIGRP next hop self. You still must still turn off EIGRP split-horizon. This was necessary in earlier DMVPN phases, but not in Phase 3, because routing updates can be summarized to the spokes, and the original IP next-hop must not be preserved. So, the spokes can send all routing updates to the hub, and the hub will summarize the private subnets and send out to all other peers. Summary can be manual, using the **`summary-address`** command on the tunnel. Autosummary also works, but it can reduce routing granularity in the core network because it summarizes to a classful boundary.

When EIGRP converges, it uses a query process to find feasible successors of routes. In the case of a hub querying a spoke router, this is really not necessary or desired. To speed convergence in a DMVPN network, it is recommended to turn on the EIGRP Stub feature. This feature has the spoke announce it is a stub device in the hello message, telling the hub router not to query it about any other route. If the spoke router is used as a transit network (that is, the spoke router has EIGRP neighbors behind it), you would not declare it as a stub router. All nodes that are EIGRP neighbors out the same mGRE tunnel interface must be marked stub for the interface to go into stub processing mode. Otherwise, EIGRP continues to process all neighbors out this interface in nonstub mode.

EIGRP deployment is straightforward in a pure hub-and-spoke deployment. The address space should be summarized as much as possible, and the spokes should be put into an EIGRP stub network. As with all EIGRP networks, the number of neighbors should be limited to ensure the hub router can reestablish communications after a major outage. If the DMVPN network is configured with too many peers, a compromise is required to balance reconvergence with recovery. Another commonly experienced problem is a lossy infrastructure network. Remember in DMVPN the Internet is the common infrastructure, and therefore can delay EIGRP messages from reaching the hub. In large EIGRP networks, it may be necessary

to adjust the EIGRP hold time to give the hub more time to recover without thrashing. However, the reconvergence time of the network will be delayed, it would be much more delay if EIGRP has to reconverge.

The requirement of the hold timer to be adjusted higher or lower depends largely on transit network conditions. This parameter may need to be adjusted as the DMVPN network grows. The recommended hold time should be between three and seven times the EIGRP hello timer, or 15 to 35 seconds. This timer can be adjusted higher than seven times the hello timer, but this should only be done by someone who understands the impact of this change.

2.8.3 RIPv2

RIPv2 is a classic and simple routing protocol. RIPv2 is a distance vector protocol; the router simply advertises the routes reachable through itself. For DMVPN, this low overhead protocol is very attractive because the topology is hub and spoke based.

When using RIPv2 for the routing protocol in DMVPN Phase 3, you must turn off split-horizon on the hub. When split-horizon is enabled, the router cannot advertise the summary route out the interface it learned the component route. We need to use the summary-address command on the tunnel to advertise the summarized routes for all other spoke private subnets. Auto-summary would also work, but it can reduce the routing granularity in the core network since it summarizes at a classful boundary.

RIP deployment is straightforward in a pure hub-and-spoke deployment. The address space should be summarized as much as possible. In RIP networks, the number of neighbors should be limited to ensure the hub router can re-establish communications after a major outage. If the DMVPN subnet is configured with too many peers, a compromise is required to balance reconvergence with recovery. In very large RIP networks, it might be necessary to adjust the RIP timers to give the hub more time to recover without thrashing. However, the convergence time of the network is delayed. Network designs that require the timers to be adjusted often leave little room for future growth.

2.9 High Availability Design

High availability (HA) provides network resilience and availability in the event of a failure. This section provides some designs for highly-available DMVPN implementations.

2.9.1 Common Elements in HA Headends

To provide resiliency in the DMVPN design, Cisco recommends that at least two hub tunnels be configured on each branch router. Regardless of DMVPN topology or deployment model, each branch router should have a tunnel to a primary headend and an alternate tunnel to a secondary headend router.

Under normal operating conditions, both the primary and secondary tunnels have routing protocol neighbors established. The routing protocol maintains both paths, where the secondary tunnel is typically configured as a less preferred path, though load balancing across both paths is possible.

A common concern in all HA headend resilient designs are the number of routing protocol neighbors. Many redundant neighbor relationships increase the time required for routing convergence. Routing protocol convergence is a common element in all HA headend designs. However, each deployment model has unique methods of achieving HA through a routing protocol convergence. If a failure occurs at a headend device,

the routing protocol detects that the route through the primary tunnel is no longer valid and, after convergence, the route through the secondary tunnel is used. When the primary tunnel is available again, traffic is routed back to the primary tunnel, because it is the preferred route in the routing metrics. The headend resiliency design presented here provides for failure of a single headend router, with proper failover to surviving headend routers, regardless of IP subnet or DMVPN cloud.

2.10 Configuration and Implementation

In this section, we look at common basic configurations recommended for all of the designs presented in this document. These configurations should be considered best practices for any DMVPN network design. It is presumed that the reader is reasonably familiar with standard Cisco configuration practices at the command-line interface (CLI) level.

2.10.1 ISAKMP Policy Configuration

There must be at least one matching ISAKMP policy between two potential crypto peers for IKE Phase 1 to work. The following configuration shows a policy using Public Key Infrastructure (PKI) Rivest, Shamir, and Adelman (RSA) certificates (also known as digital certificates) for the ISAKMP authentication. The policy does not show up in the ISAKMP policy because it is the default, but we have shown it for completeness. It is also possible to use preshared keys (PSKs), but this is not very scalable, and is not recommended for large scale DMVPN designs.

To use PKI for ISAKMP, you must first create the crypto keys, which should be generated with a specified modulus size (the default is 512-bits). It is best to use 2048-bit or larger keys for most high performance routers, but this may not be appropriate for lower performance platforms. After the keys are created, you must authenticate and enroll with the Certificate Authority (CA). Setting up the PKI trustpoint is outside the scope of this DIG. Refer to cisco.com for information about how to set this up on the routers with your CA.

It is recommended to use a *strong* encryption algorithm, such as Triple Data Encryption Standard (3DES) or Advanced Encryption Standard (AES). In the sample configuration, the encryption algorithm used for the ISAKMP SA is Triple-DES (3DES), and the Diffe-Helman group is 2. These are recommended, although other values may be used.

```
ip domain-name cisco.com
!
crypto pki trustpoint DMVPN-CA
 enrollment url http://10.1.1.254:80
 serial-number none
 ip-address 10.21.1.2
 password
 subject-name CN=DMVPN_HUB, ou=CISCO
 revocation-check none
 rsakeypair dmvpnkey
 auto-enroll
!
crypto pki certificate chain DMVPN-CA
 certificate 3607
 <certs omitted>
 certificate ca 01
!
crypto isakmp policy 2
 encr 3des
 authentication rsa-sig
```



```
 group 2
!
```

2.10.2 IPsec Transform Set and Protocol Configuration

The transform set defines the set of crypto and hashed message authentication algorithms used for the IPsec SA. The transforms must match between the two IPsec peers. The transform set names are locally significant, however, the encryption algorithm, hash method, and the particular protocols used (Encapsulating Security Payload (ESP) or Authentication Header (AH)) must have at least one match. Data compression can also be configured, but it is not recommended on peers with high-speed links. You can use multiple transform sets between different peers. The first match is used, so the strongest algorithms should be first in the configuration.

After defining the transform set, specify the tunneling mode to be used. It was stated previously in this chapter that transport mode *should* be used for DMVPN. Because GRE manually creates the tunnel, the transport mode simply encrypts the GRE header (4 or 8 bytes) and the private IP packet traveling inside the mGRE tunnel, but does not encrypt the GRE/IP packet header.

```
crypto ipsec transform-set DMVPN_SET esp-3des esp-sha-hmac
 mode transport
!
```

2.10.3 IPsec Profile and Tunnel Protection Configuration

The IPsec profile is used to group IPsec tunnel parameters. This profile is where you specify the transform set to be used in your DMVPN tunnel. You can also specify other IPsec parameters such as SA lifetime. These commands pertain to an IPsec policy that can be issued under an IPsec profile; there is no need to specify, nor can you specify, the IPsec peer address or the ACL to match the packets that are to be encrypted. This information is automatically provided at time of connection by GRE and NHRP.

The IPsec profile is associated with a tunnel interface using the **`tunnel protection IPsec profile`** ***`profile-name`*** command, also first introduced in Cisco IOS Software Release 12.2(13)T. The **`tunnel protection`** command can be used with mGRE and p2p GRE tunnels. With p2p GRE tunnels, the tunnel destination address is used as the IPsec peer address. With mGRE tunnels, multiple IPsec peers are possible; the corresponding NHRP-mapped NBMA destination address is used as the IPsec peer address.

```
crypto ipsec profile dmvpnhub
 set transform-set gre_set
!
interface Tunnel1
 ...
 tunnel protection ipsec profile dmvpnhub
!
```

If more than one mGRE tunnel is configured on a router (for example, on a branch router with two DMVPN interfaces), the tunnels can use the same tunnel source address (interface) on each tunnel interface. In this case, the **`shared`** keyword must be used on the tunnel protection command on both interfaces. This does not mean that the two mGRE tunnels are hosting the same DMVPN cloud; each tunnel interface still requires a unique NHRP network-ID and IP subnet. This does mean that these tunnel interfaces must have different tunnel keys (mGRE) or different tunnel destinations (p2p GRE).

In Chapter 5, we see that the 6500 can be used to terminate the crypto piece of the tunnel. This configuration does not use the tunnel protection command; instead, it uses dynamic crypto maps. The 6500 configuration is described in detail in Chapter 5.

2.10.4 mGRE Tunnel Interface Configuration

mGRE configuration enables a tunnel to have multiple tunnel destinations. The mGRE configuration on one side of a tunnel does not have any relation to the tunnel properties that might exist on the other side of the tunnel. This means that an mGRE tunnel on the hub can connect to a p2p tunnel on the branch. Tunnel destination distinguishes an mGRE interface and a p2p GRE interface. An mGRE interface has no configured destination. Instead, the GRE tunnel is configured with the **`tunnel mode gre multipoint`** command. This command is used instead of the **`tunnel destination x.x.x.x`** used with p2p GRE tunnels. Enabling multiple destinations for an mGRE tunnel requires NHRP to resolve the tunnel endpoints. mGRE is required on the hub device and recommended on the spoke router. It is required on the spoke router if spoke-to-spoke tunnels are desired.

```
interface Tunnel1
 bandwidth 1536
 ip address 10.81.1.1 255.255.0.0
 ip mtu 1440
 tunnel source 10.70.101.2
 tunnel mode gre multipoint
!
```

2.10.5 NHRP Configuration

NHRP provides a mapping between the inside (VPN) and outside (NBMA) address of a tunnel endpoint. These mappings can be static or dynamic. In dynamic mapping, a next-hop server (NHS) maintains a list of possible tunnel endpoints, and facilitates a spoke in finding other mappings needed by the spoke. Each endpoint (spoke) registers its own public and private mapping with the NHS with which it is configured. Local NHS mapping must always be static. It is important to note that even though the static NHRP mappings on a spoke router point to the NHS server NBMA address, the **`ip nhrp nhs`** … command points to the tunnel interface address of the headend.

It is important to define the NHRP mappings correctly. On the branch (spoke) routers, you must configure at least one static NHRP mapping in order to reach the NHS. If you use more than one headend hub router, you must have multiple mappings. The hub devices likely also have static mappings to other hubs in the hierarchy. Additionally, you would need to enable broadcast/multicast over the spoke-hub tunnel for routing protocols and other multicast data to work over the DMVPN tunnel.

Note: IP multicast/broadcast packets are supported only on tunnels with static mappings (the spoke-hub tunnels). Hubs are configured to enable NHRP to automatically add routers to multicast NHRP mappings. This is necessary to run routing protocols over the DMVPN mGRE tunnel. NHRP can only add a peer to the multicast mapping list when it receives an NHRP registration packet (only on NHRP NHSs).

NHRP hold time is used to determine how long receiving routers should consider the cached entry information to be valid. The configured value on the sender of the information is passed to the receiver in the NHRP registration request or resolution reply packet. When the remote node adds this information to its mapping database, the remote node starts a countdown timer. When this timer expires, the node removes cached entry information. If traffic is still flowing when the timer is about to expire, the node must request the mapping again to refresh it.

NHRP registration requests are sent periodically (by default, one-third of the holdtime) to keep a NHRP registration mapping entry on the hub from timing out. Spoke routers can have different hold times, although this practice is not common. If two spokes are in session, and one timer expires before the other, that spoke, will time out the entry. In so doing it will clear the corresponding local IPsec and ISAKMP SAs. As part of

this process ISAKMP will notify the remote spoke to clear its ISAKMP and IPsec SAs. When the remote spoke clears its ISAKMP and IPsec SAs it will also clear the corresponding NHRP mapping entry.

Although spoke-to-spoke voice (VoIP) over DMVPN is not generally recommended because of QoS concerns, the NHRP hold time should be longer than the duration of the majority of calls. The hold timer should not be so long that spoke-to-spoke sessions are idle on average. This recommendation is especially important for low-end routers, where software imposes a lower limit on the number of crypto tunnels. An overall balance between idle tunnels and excessive recaching can be achieved by setting the idle time to 600 seconds.

Other NHRP options Cisco recommends to use are authentication and network-ID. NHRP authentication is used to authenticate the NHRP messages. All routers configured with NHRP within a DMVPN cloud must share the same authentication string.

An NHRP network-id is required on the DMVPN Tunnel interfaces on a node. The network id does not need to match on all nodes in a given DMVPN network, but it is recommended to match them throughout the DMVPN network to eliminate possible confusion.

If you have multiple NHRP enabled tunnel interfaces on a DMVPN node, and the interfaces are in the same DMVPN network, the NRHP Network-ID must match. This will impact the operation of NHRP between DMVPN nodes. If your network requires this configuration, you should understand what this configuration does. On a related topic, you can have multiple interfaces with separate DMVPN networks, and require different NHRP network-IDs. This is used to isolate the NHRP tables to be used by only the configured interfaces. This is commonly used in dual-DMVPN networks where a hub is part of two DMVPN clouds.

For DMVPN Phase 3 to work you must add the NHRP redirect and shortcut commands to the tunnel interfaces. The **`ip nhrp shortcut`** command is used to build shortcut paths and switch packets using the shortcut path. When NHRP receives an NHRP redirect traffic indication message, it will kick off a resolution request and find the shortcut path for the destination IP of the data packet. The **`ip nhrp redirect`** command enables sending NHRP redirects (similar to ICMP redirects). When the inbound and outbound interface for a data packet being forwarded through a router is in the same DMVPN network, an NHRP redirect traffic indication message is sent to the previous GRE hop (tunnel) from which the data packet was received. Thus the NHRP shortcut and redirect commands work together. Generally the NHRP redirect configuration will be required on the hub whereas NHRP shortcut is required to be configured on spokes. If hub also needs to build shortcuts to other hubs or other spokes that are not registered to it, then NHRP shortcut is also required to be configured on the hub's mGRE tunnel interface. It is recommended to configure both shortcut and redirect on both the hub and the spokes due to the dynamic nature of routes.

The following basic DMVPN hub and spoke configurations illustrate the preceding descriptions.

Hub Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic ← Add spoke to multicast list upon registration
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send NHRP redirects
 tunnel source GigabitEthernet0/1
```



```
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
```

Spoke Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.101.2     ← Add hub to receive multicast packets
 ip nhrp map 10.81.0.1 10.70.101.2     ← Static NHRP mapping for hub
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp nhs 10.81.0.1  ← Indicates the next hop server
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
```


3 Hierarchical Hub DMVPN Design

The hierarchical hub DMVPN design encompasses all of the functionality of the DMVPN Phase 3 solution. In this model, there can be multiple distribution hubs where the regional spokes connect. The distribution hubs connect to one or more central core hubs, thus creating a hub hierarchy. Thus, the deployment can be distributed in a highly scalable manner over a large geographic area. Because of its hierarchical nature, it is very easy to add more spokes or additional hubs to the design.

In the hierarchical hub design, each spoke creates an mGRE tunnel, which terminates on the headend (hub) router. If tunnel protection is used (IPsec), the headend router must encrypt/decrypt all traffic entering and leaving the DMVPN central site over the GRE tunnel to a given spoke. Two factors here have some impact on the deployment of the headend (hub) router. The router selected to be a DMVPN hub must be able to handle the number of routing protocol neighbors (spokes) planned for the deployment. The DMVPN hub must also be able to handle the the necessary encryption throughput for the deployment.

In DMVPN Phase 3, if spoke-to-spoke functionality is configured, traffic between spokes eventually follows a shortcut path, taking significant load off the hub. With spoke-to-spoke tunnels, traffic transits the central hub site until the tunnel is set up between the spoke sites. After the tunnel is setup between the two spokes, traffic can shortcut to the destination spoke, and does not transit the hub routers. During setup of the spoke-to-spoke tunnel, the hub router routes the traffic. This design also means that spoke-to-spoke tunnels in a region are assisted by the regional hub and do not add load to the central hubs, or to hubs in other regions.

Note: The preceding headend sizing criteria does not apply in a pure hub and spoke setup, because the hub must handle the routing protocol load, spoke to central site traffic, and all spoke to spoke traffic. The throughput load on the hub can be much higher in this case.

Typically, a hierarchical hub design is used when there is a large geographic separation between hub sites, and spokes are naturally grouped into regions. For DMVPN phase 3, the hierarchical design is multilayered. This means DMVPN is used between the distribution hubs and spokes, and between the central hubs and distribution hubs. There can be more than just these two layers, and the number of layers does not need to be symmetric across the whole DMVPN network.

The hierarchical design enables common routing protocol designs to work well. It is basically a collection of hub and spoke layers. Therefore, a well-planned topology and addressing scheme can provide a well-summarized set of routes. This keeps the routing tables on all devices as compact as possible, and supports easier network growth. This flexibility enables DMVPN Phase 3 networks to more closely match geographic layout or the data flow patterns of the network.

In Phase 3, the ability to use summarization in the routing protocol enables hubs to use the power of the routing protocol. Routing and CEF switching are used to forward both data and NHRP packets more optimally through the distributed hubs. There is no need for full routing knowledge at the spokes. In general, as routes are advertised in the spoke to hub direction they are not summarized. Conversely, routes advertised in the hub to spoke direction are summarized.

This chapter illustrates common deployment scenarios for this design.

3.1 Deployment

The hierarchical hub design is primarily used when the DMVPN network is widely distributed geographically, or when the main data traffic pattern is within a region with less traffic to the central site or spokes in other regions. This design can operate on a small scale, but because our focus is scalability, this chapter focuses on the bigger picture.

This design would typically be used in an enterprise with many regional and branch offices, separated geographically and connected to the Internet. This would be common where there are many smaller branch offices, such as sales offices or banks. Each branch office would connect to the nearest regional office using a DMVPN tunnel through the Internet. Finally, the regional offices would all connect to the main campus hub. This deployment of a DMVPN hierarchy works well for dispersed enterprise networks.

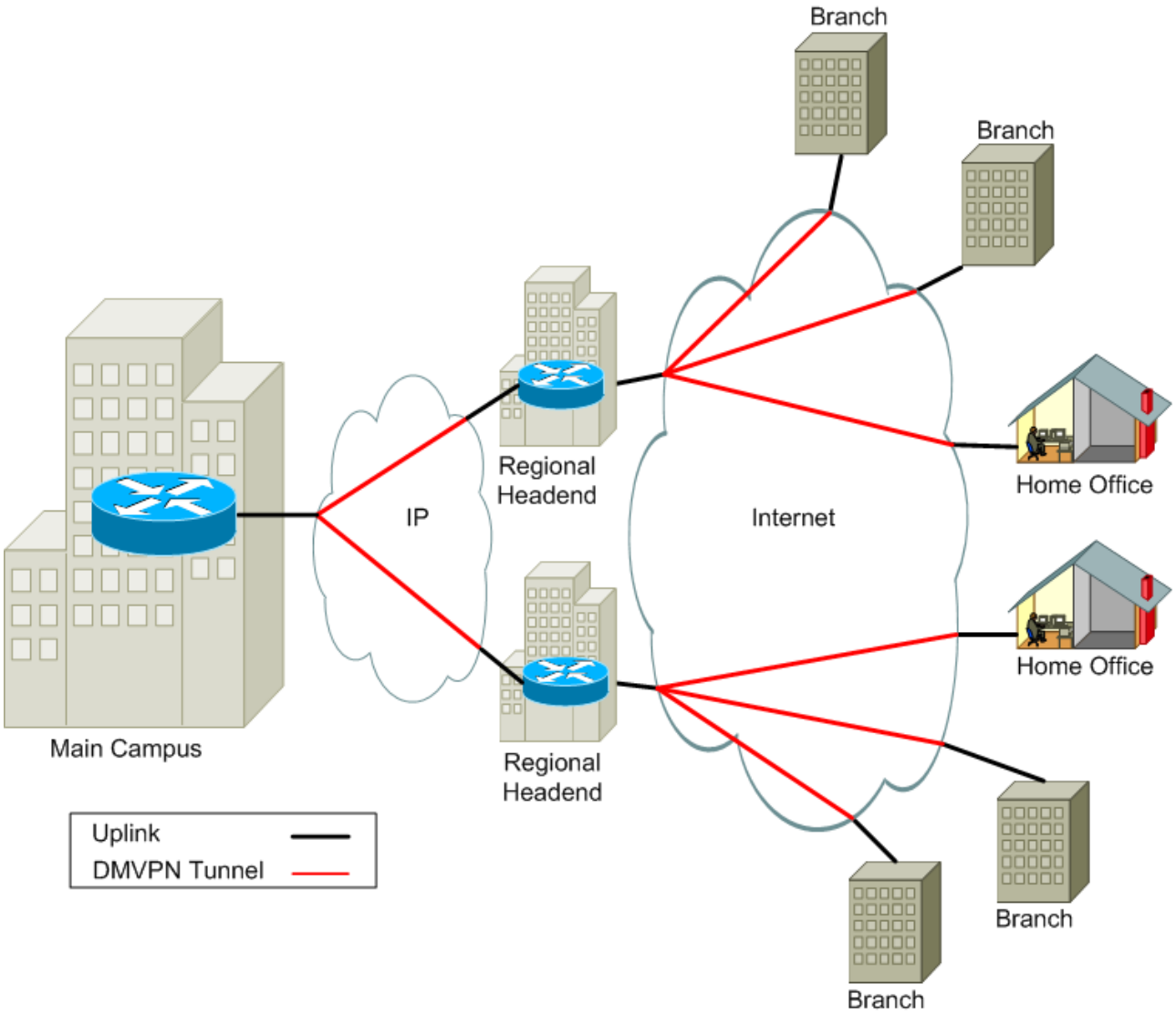
This design would not work well for small office, home office (SOHO) deployments, where the VPNs are rather dense in distribution. For SOHO, we look at the server load balancing (SLB) designs in later chapters. Note that SLB hub designs could be added to the hierarchical hub designs to improve scalability. High availability (HA) is also common in hierarchical hub design deployments.

The power of the hierarchical hub design is its ability to distribute hub routers and corresponding spoke routers into regions. The regional hubs are then connected to an upper layer of core hubs, and so on, producing a hierarchy of hubs. Figure 3-1 illustrates the basic design.

For simplicity, we have illustrated the design using a few spokes and a single hub. In reality, this design typically has a hierarchy of hubs distributing the connections to many regions. In each region, redundancy is a common concern that is addressed later in the chapter. Furthermore, distribution of routing information is very important, and must be considered when deploying this design.

Throughput must always be considered when working with encrypted traffic and VPNs. It is important to estimate the maximum throughput the deployment is expected to handle. Given the limits of encrypted throughput, you can design the DMVPN hub and spoke sites to fit your needs.

Figure 3-1. Hierarchical Hub Deployment



3.2 Basic Operation

The basic form of the hierarchical hub design is to have many regional hubs (in Figure 3-2, H1 and H2) each terminating DMVPN tunnels from spokes in a geographic region (S1, S2, S3, S4). The spokes are set up at branch sites requiring secure data connections to the regional sites. If a new branch joins the network, the spoke router simply needs to be configured to build its tunnel to the regional hub. If properly configured, the regional hub needs no configuration changes to recognizes the new spoke. The IP subnet on the tunnel interface of the all the hubs and spokes is large enough to accommodate the additional spokes. Remember that even though the network is laid out as a hierarchical hub and spoke, the tunnel interfaces are configured as a flat network all in the same IP subnet.

This design enables spoke-to-spoke tunnels between any two nodes in the network. Consider that the central hub (H12) is another spoke relative to the spokes in the regions. Spokes that access resources behind the central hub build spoke-to-spoke tunnels with the central hub. The central hub must be sized to handle the

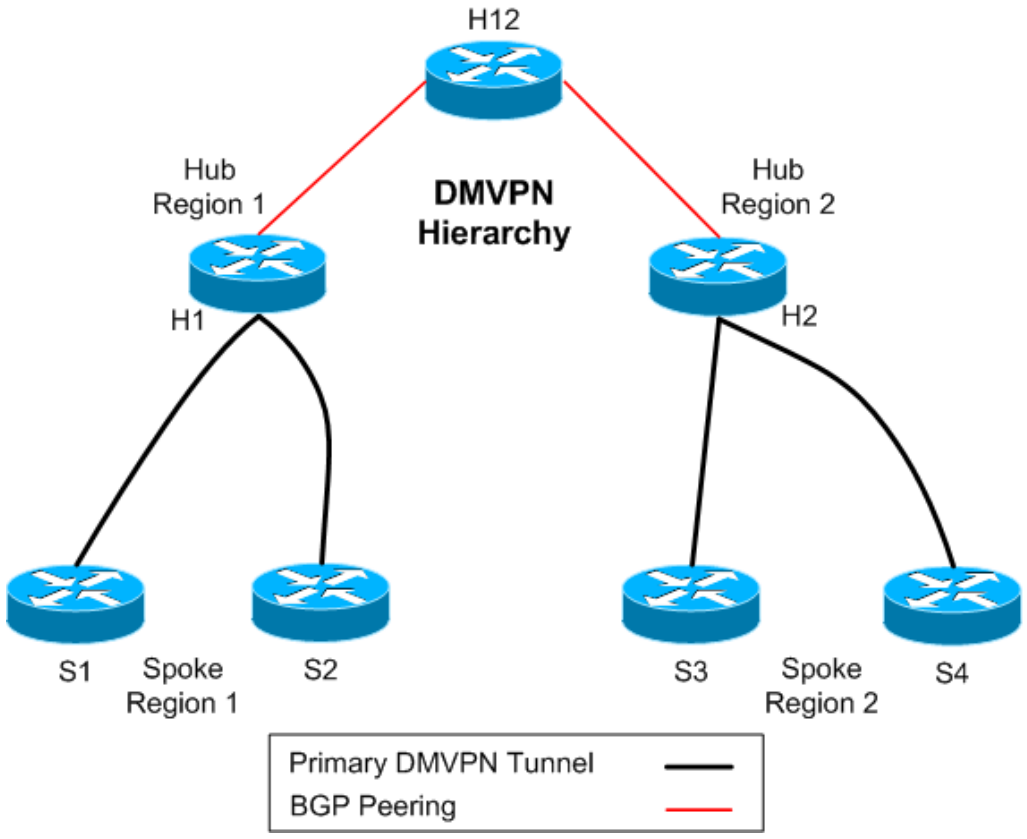
tunnels with the regional hubs and these dynamic tunnels with spokes. As an alternative, block the building of spoke-to-spoke tunnels to the central hub, thus forcing this traffic to take the path through the regional hub.

To block dynamic spoke-to-spoke tunnels with the central hub, you must:

- Not configure **`ip nhrp shortcut`** on the central hub.
- Configure **`ip nhrp interest`** ***`access-list`*** on the spokes, with ***`access-list`*** denying the networks behind the central hub.

Note that dynamic spoke-to-spoke tunnels can be built between the regional hubs, and between spokes and regional hubs in another region. This networktruly supports dynamic spoke-to-spoke tunnels between any two nodes.

Figure 3-2. Basic Hierarchical Hub Design



Let us look at the basic tunnel configuration for the spoke S1. (We will next look at the corresponding hub H1). In this example, we are leaving out the crypto configurations, and the other interfaces that were covered in Chapter 2.

Note that the tunnel source indicates the public interface. The static NHRP entry is used in the destination field of the generic routing encapsulation (GRE) header, forming the tunnel. The mapping of multicast to the hub is important because we use routing protocols that use multicast to distribute routing information. The NHRP next hop server (NHS) tells the spoke where to send the NHRP registration. The NHRP shortcut, redirect, and other NHRP options are covered in more detail in Section 2.4, NHRP.

Spoke Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.101.2     ← Forward multicasts to hub
 ip nhrp map 10.81.0.1 10.70.101.2     ← Static NHRP mapping for hub
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp nhs 10.81.0.1  ← Indicates the next hop server
 ip nhrp shortcut       ← Allows initiating shortcut tunnels
 ip nhrp redirect    ← Allows sending NHRP redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
```

The hubs, where GRE tunnels, NHRP, and crypto are terminated from the spokes, are the most complex part of a DMVPN deployment. The basic hub tunnel configuration follows. When a spoke first brings up the tunnel to the hub, it must register its public IP address (the source address of the GRE tunnel) in the NHRP table on the hub. The NHRP table tells the hub which destination IP address to use in the GRE/IP header of a packet heading from the hub to a given spoke. NHRP maps the spoke tunnel IP address to the spoke globally routable public IP address (tunnel source).

In the configuration, note that the multicast mapping is dynamic. This command instructs the hub to automatically add the spoke to the hub NHRP multicast list when the spoke registers, and is needed to enable dynamic routing protocols to work over the multipoint GRE (mGRE) and IPsec tunnels because Interior Gateway Protocol (IGP) routing protocols use multicast packets.Without this command, the hub router would need a separate configuration line to add each spoke individually to its NHRP multicast mapping list.

Regional Hub Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic  ← Add spoke to multicast list upon registration
 ip nhrp network-id 101         ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut       ← Allows initiating shortcut tunnels
                          (not always needed on hubs)
 ip nhrp redirect       ← Allows sending NHRP redirects
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
```

Each regional hub (H1 and H2) is connected to a core hub (H12) and typically uses Internal Border Gateway Protocol (iBGP) peering to distribute routing information. The regional hubs essentially act as spokes to the core hub. The regional hub would typically be located in a regional office of the enterprise. The core hub would typically be located on the main campus network of the enterprise.

A configuration for the core hub and additional configuration for the regional hub follow. It is recommended to use a separate mGRE tunnel interface for the hub-to-hub connections, rather than using the same tunnel interface that the spokes connect to. This provides more routing control. When using the the same interface, the inter-hub connections would have to exist on the same subnets as the spokes. This causes a problem with route summarization into the core network, causing inefficiency. When using a second tunnel interface, you must be sure that the NHRP network-ID is the same as that of the spoke-facing tunnel interface, thus associating both interfaces to the same NHRP process.

The NHRP network-ID is important when building a spoke-to-spoke tunnel across regions. In this case, the NHRP network-ID must be the same on each GRE interface, that is, the GRE tunnel interfaces for the hub-to-spoke tunnel, and for the hub-to-hub tunnel. The regional hub knows to send the NHRP redirect only if the packet is forwarded out a GRE tunnel having the same network-ID as the spoke originating the packet. As stated previously, the regional hub is treated as a spoke of the core hub. Therefore, note that when packets are sent between the regional hubs, a hub to hub tunnel can be formed if the hubs are in the same NHRP network-ID on the core hub, and the **`redirect`** and **`shortcut`** commands are configured. Also, note that you cannot use a tunnel key when using multiple mGRE tunnel interfaces with the same NHRP network-ID. Therefore, each mGRE interface must have a unique tunnel source IP address.

DMVPN Phase 3 also supports route summarization between the regions. Hubs in one region can forward packets toward core hub routers, higher in the hierarchicy, which can then forward the packets to other regions. The regional hubs forward the packet to the destination spoke router. Also any spoke-spoke tunnels that are built within a region are handled by the regional hub alone. If a spoke-spoke tunnel crosses between regions, the NHRP resolution request is forwarded through the central hubs on to the destination spokes.

Regional Hub Additional Configuration

```
interface Tunnel0     ← Notice this a different tunnel interface (not to spokes)
 bandwidth 1000
 ip address 10.181.1.2 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 192.168.1.1
 ip nhrp map 10.181.1.1 192.168.1.1     ← Static NHRP mapping for hub
 ip nhrp network-id 101     ← Numerically identifies the DMVPN network (same as
                              on tunnel interface to spokes)
 ip nhrp holdtime 600
 ip nhrp nhs 10.181.1.1     ← Indicates the next hop server
 ip nhrp shortcut           ← (Optional) builds hub to hub shortcut tunnel
 ip nhrp redirect           ← Indicates to send NHRP redirects (See note below)
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
!
```

Note: Configuring the **`ip nhrp redirect`** on the hub-to-hub (outbound) tunnel interface is currently mandatory. This triggers the hub to send an NHRP redirect toward the spoke if the outbound tunnel interface is configured with **`ip nhrp redirect`** when a packet is sent back out. This will change in later releases, so the **`ip nhrp redirect`** on the inbound interface will control this behavior.

Core Hub Configuration

```
interface Tunnel0
 bandwidth 1000
 ip address 10.181.1.1 255.255.255.0
 no ip redirects
 ip mtu 1400
```



```
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 101     ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut           ← (Optional) Indicates to build the shortcut tunnel
 ip nhrp redirect           ← (Optional) Send NHRP redirects to regional hubs
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
!
```

The preceding hub configurations do not have any **tunnel** protection commands. Typically, there is no need to encrypt inside the campus network. However, if the regional hubs are located away from the core hub, tunnel protection would be a good idea if the network is not trusted. Also, the shortcut and redirect commands are listed as optional. This is because you may not wish to build direct hub-to-hub DMVPN tunnels between the regional hubs. This is again an option you may wish to enable.

The preceding examples depict a DMVPN design having one distributed cloud. This is not the only possible hierarchical hub design. You could have unique DMVPN clouds per region of spokes. Looking at Figure 3-2 again, you could have separate DMVPN networks, one in Region 1 and another in Region 2. In this case, you cannot build spoke to spoke tunnels between regions. Therefore, there is no value in configuring NHRP redirect and shortcut on the central hub. This also removes the need to have identical NHRP network-IDs on spoke-regional hub and on the regional-core hub tunnel interfaces.

3.3 High Availability Design

HA provides network resilience and availability in the event of a failure. This section provides some designs for highly-available DMVPN implementations. To provide resiliency in the DMVPN design, Cisco recommends configuring at least two hubs tunnels (NHRP servers) on each branch router. Regardless of the DMVPN topology or deployment model, each branch router should have a tunnel to a primary headend and an alternate tunnel to a secondary headend router.

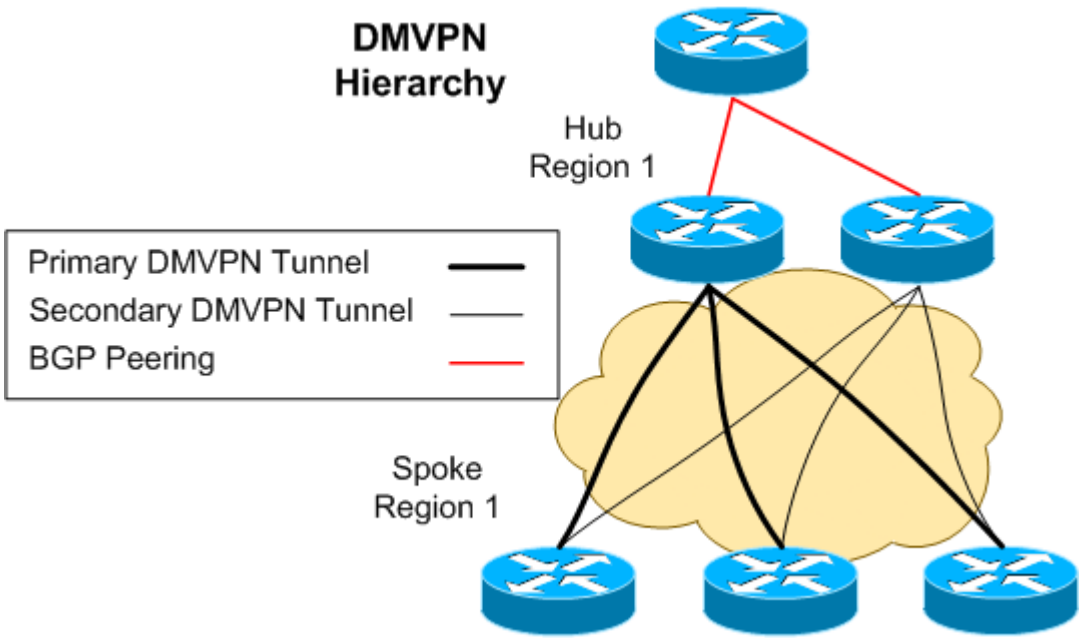
Under normal operating conditions, both the primary and secondary tunnels have routing protocol neighbors established. The routing protocol maintains both paths; the secondary tunnel is configured as a less preferred path in the routing protocol. The routing protocol metric can be configured to load-balance IP flows across both paths, with the knowledge that there may be asymmetric routing for the spoke-to-hub path compared to the hub-to-spoke path. This is part of the DMVPN active-active redundancy model.

A common concern in all HA headend designs is the number of routing protocol neighbors. Many redundant neighbor relationships increase the number of prefixes advertised, and consequently the time required for routing convergence. Routing protocol convergence is a common element in all DMVPN headend designs. However, although each deployment model has unique methods of achieving HA, the strain on the routing protocol must be taken into account and handled properly.

3.3.1 Single DMVPN Cloud Hierarchical Hub Redundancy

Figure 3-3 illustrates the single DMVPN cloud hierarchical hub redundancy design, which is an active-active redundant design. Each spoke in a region has one mGRE tunnel interface, and the spoke points to two separate NHSs. Each hub is an NHS for each spoke. This maintains redundant tunnels to hubs in a DMVPN network.

Figure 3-3. Single DMVPN Cloud Hierarchical Hub Redundancy



The routing protocol on the core hub is used to prefer one hub over another. Each hub advertises all individual private spoke subnets to the core network, and one hub should be preferred over another. As we saw in the preceding section, you configure a tunnel interface toward the parent or core hub router. For brevity , this tunnel interface is not shown in the following configuration examples, but hierarchical operation still requires the tunnel interface.

If a failure occurs at a headend device, the routing protocol detects that the route through the primary tunnel is no longer valid; after convergence, the route through the secondary tunnel is used. When the primary tunnel is available again, traffic is routed back to the primary tunnel because it is the preferred route in the routing metrics. This headend resiliency design handles failure of one headend router, with proper failover to surviving headend routers, regardless of IP subnet or DMVPN cloud.

One regional hub will learn summary routes only from its peers in the core network. If a tunnel between a regional hub and one of its spokes fails, the regional hub would lose connectivity to the remote spoke even though a path exists through its regional hub peer. The easiest way to avoid this is to run iBGP between the two regional hubs over a direct physical link or tunnel link, so that each hub has full routing knowledge within its region and can pass data packets to the other hub if it does not connect to a particular spoke router.

Spoke Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.101.2      ← Add Hub1 to multicast list
 ip nhrp map 10.81.0.1 10.70.101.2      ← Hub1 Static NHRP mapping
 ip nhrp map multicast 10.70.102.2      ← Add Hub2 to multicast list
 ip nhrp map 10.81.0.2 10.70.102.2      ← Hub2 Static NHRP mapping
 ip nhrp network-id 101
 ip nhrp holdtime 600
```



```
 ip nhrp nhs 10.81.0.1  ← Indicates Hub1 as a next hop server
 ip nhrp nhs 10.81.0.2  ← Indicates Hub2 as a next hop server
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
```

Primary Hub Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic   ← Add spoke to multicast list upon
                                   registration
 ip nhrp network-id 101          ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send NHRP redirects
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
interface GigabitEthernet0/1
 ip address 10.70.101.2 255.255.255.0
!
```

Secondary Hub Configuration

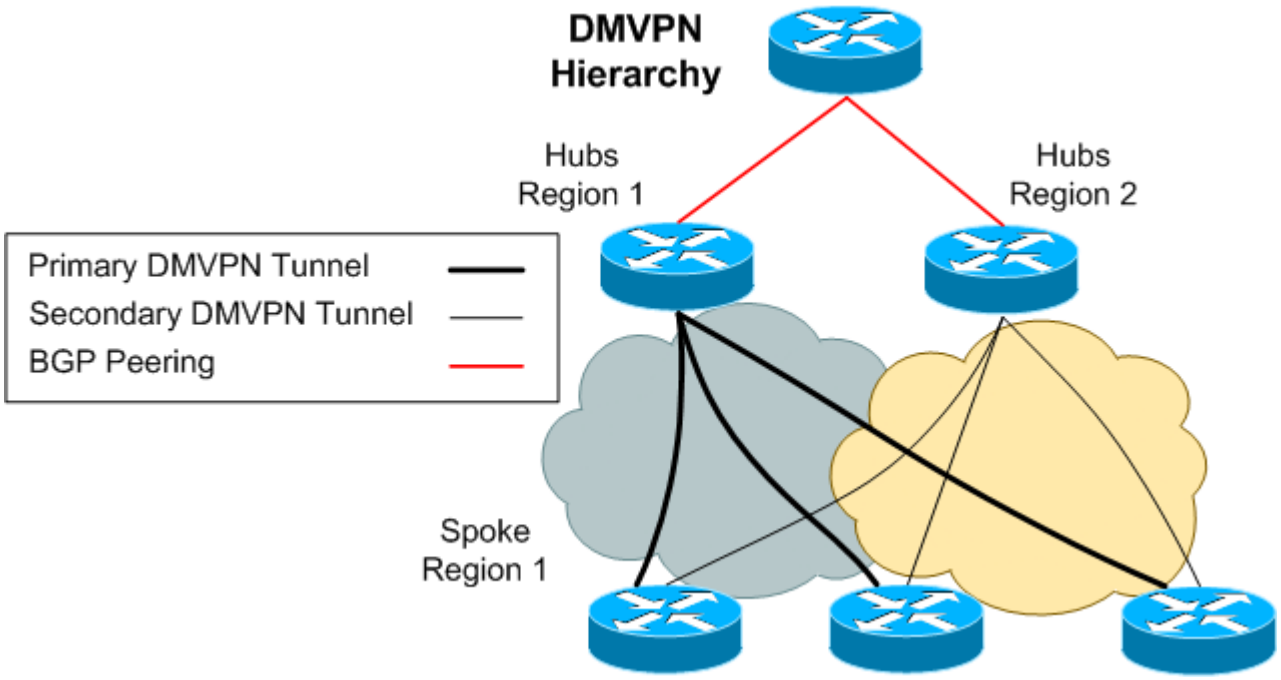
```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.2 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic   ← Add spoke to multicast list upon
                                   regirstration
 ip nhrp network-id 101          ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send NHRP redirects
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
interface GigabitEthernet0/1
 ip address 10.70.102.2 255.255.255.0
!
```

3.3.2 Dual DMVPN Cloud Hierarchical Hub Design

Figure 3-4 illustrates the dual DMVPN cloud hierarchical hub redundancy design, which is an active-active redundant design. This design supports geographic redundancy. Each spoke in the region has two mGRE tunnel interfaces. The primary tunnel interface points to the NHS of the primary hub, using the primary

subnet. The secondary tunnel interface points to the NHS of the secondary tunnel, using the backup subnet. This maintains tunnel redundancy to the hubs and core network over dual DMVPN networks.

Figure 3-4. Dual DMVPN Cloud Redundancy



With multiple mGRE tunnels configured on the router, we must reference the same tunnel source address on each tunnel interface. In this case, the **`shared`** keyword is used in the **`tunnel protection`** command on both interfaces. This does not mean that the two mGRE tunnels host the same DMVPN cloud; each tunnel interface still requires a unique NHRP network-ID and IP subnet. Just as we saw in the preceding section, you configure a tunnel interface toward the parent or core hub router. For brevity , this tunnel interface is not shown in the following configuration examples, but hierarchical operation still requires the tunnel interface.

This design requires definition of a primary and backup subnet for the primary and backup DMVPN clouds. The headend and branch routers must manipulate routing metrics to get the proper routing, with preference for the primary headend.

If a failure occurs at a headend device, the routing protocol detects that the route through the primary tunnel is no longer valid; after convergence, the route through the secondary tunnel is used. When the primary tunnel is available again, traffic is routed back to the primary tunnel because it is the preferred route in the routing metrics. This headend resiliency design handles failure of one headend router, with proper failover to surviving headend routers, regardless of IP subnet or DMVPN cloud.

This design, along with VRF-lite or "tunnel route-via" configuration can be used to run DMVPN over two infrastructure networks (two ISPs), and can even support load-balancing over the two infrastructure networks (ISPs).

Spoke Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.1.1 255.255.0.0
```



```
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.101.2      ← Add Hub1 to multicast list
 ip nhrp map 10.81.0.1 10.70.101.2      ← Hub1 Static NHRP mapping
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp nhs 10.81.0.1  ← Indicates Hub1 as the next hop server
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub shared    ← Shared keyword required
!
interface Tunnel2
 bandwidth 1000
 ip address 10.82.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.102.2      ← Add Hub2 to multicast list
 ip nhrp map 10.82.0.2 10.70.102.2      ← Hub2 Static NHRP mapping
 ip nhrp network-id 102
 ip nhrp holdtime 600
 ip nhrp nhs 10.82.0.2  ← Indicates Hub2 as the next hop server
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub shared    ← Shared keyword required
!
```

Primary Hub Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic    ← Add spoke to multicast list upon
                                    registration
 ip nhrp network-id 101          ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut       ← Indicates to build the shortcut tunnel
 ip nhrp redirect       ← Indicates to send NHRP redirects
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
interface GigabitEthernet0/1
 ip address 10.70.101.2 255.255.255.0
!
```

Secondary Hub Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 10.82.0.2 255.255.0.0
 no ip redirects
```

```
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic    ← Add spoke to multicast list upon
                                     registration
 ip nhrp network-id 102           ← Numerically identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut        ← Indicates to build the shortcut tunnel
 ip nhrp redirect        ← Indicates to send NHRP redirects
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
interface GigabitEthernet0/1
 ip address 10.70.102.2 255.255.255.0
!
```

3.4 Headend QoS

Implementing quality of service (QoS) on the headend is often necessary because a DMVPN spoke tunnel can easily become congested due to the extra overhead, and because the hub router physical interfaces do not become congested as fast as the spoke inbound Internet connection. Therefore, it is useful to configure the hub with a shaped queuing policy. This is a hierarchical Modular QoS CLI (MQC) shaper with a child policy for queuing traffic after shaping is performed.

In traffic from the headend toward the branch, there are several ways to classify traffic for queuing and shaping toward a particular spoke:

One way is to use the **qos-group** under the Internet Security Association and Key Management Protocol (ISAKMP) profile. We can use **crypto isakmp profile** to assign a destination branch to a QoS group. Traffic destined to this branch is then placed in a **qos-group** by the crypto identification. Then, it can be used for classification by its qos-group in the QoS policy. If more than one spoke is assigned to the same qos-group, their traffic is measured together for the shaping and policing limits.

It is possible to classify based on information that is encrypted as well. You can use an access-list, configured to match the private subnet behind the remote spoke. The **qos pre-classify** command is used on the tunnel interface, and is required because the traffic is classified by a parameter that is encrypted as the traffic leaves the physical outbound interface. L4 information from the IP data packet can also classify traffic destined to the same private subnet.

In a final method, an ACL is configured to match the source and destination of the GRE traffic as it leaves the physical outbound interface. This does not require the **qos pre-classify** command, but is usable only if the remote spokes had static IP addresses. Also, in this case only the IP ToS byte can be used to classify traffic within a particular tunnel because it is copied from the IP data packet.

Here, we look at the configuration for the first method. Traffic to that spoke is shaped to 256Kbps and the child policy enforces low latency queuing (LLQ) and lass-based weighted fair queuing (CBWFQ). Classification in the child policy is done using the IP ToS byte, which is copied from the IP header of the data packet to the IP header of the GRE/IP and IPsec/IP headers.

Hub Configuration

```
class-map match-all class_spoke.1.1
 match qos-group 1
!
```



```
policy-map test
class class_spoke.1.1
  shape average 256000
  service-policy child
!
class-map match-all HP1
 match ip precedence 1
class-map match-all HP2
 match ip precedence 2
class-map match-all voice
 match ip precedence 5
policy-map child
 class HP1
  bandwidth 32
 class HP2
  bandwidth 32
 class voice
  priority 64
!
crypto isakmp profile spoke.1.1
   ca trust-point DMVPN-CA
   match identity address 10.21.1.2 255.255.255.255
   match identity user domain cisco.com
   qos-group 1
!
interface GigabitEthernet0/1
 ip address 10.70.100.2 255.255.255.0
 service-policy output test
!
```

Anyone who works with QoS knows that its CPU and memory consumption is very large. The limitations can be closely correlated with processing power and packet storage while delaying their transmission time slot. With DMVPN, it becomes a little more complicated when you add the encryption card, and must consider the interaction of queuing and processing with hardware encryption.

Looking at the Cisco 7200 series router with NPE-G1 and NPE-G2 processors, along with the VAM2+ encryption card, we can compare the processing of the preceding QoS policy. The NPE-G1 with VAM2+ was found to handle a maximum of about 80 active shapers, while the NPE-G2 with VAM2+ handles a maximum of about 150 active shapers. More shapers can be defined; but only these maximum numbers of shapers can actively shape traffic at one time.

Note: It is *not* recommended to use QoS with DMVPN if running IOS 12.4(11)Tx through 12.4(15)T1, due to serious code defects.

3.5 IP Multicast

Multicast on DMVPN is similar in behavior to any nonbroadcast multiaccess (NBMA) network. These multicast deployments typically have a hub to spoke direction of data flow, where the source of the multicast stream is located somewhere in the core network. Because the DMVPN cloud is an NBMA network, Protocol Independent Multicast sparse mode (PIM SM) must be used, with PIM NBMA mode enabled on the hub device. The receivers would join at the branch site, and the PIM join would be sent up the DMVPN tunnel and toward the rendevous point (RP). The placement of the RP is somewhere in the core network, or might be at the DMVPN headend. The RP cannot be configured at a spoke.

Direct spoke to spoke multicast is *not* supported. The multicast source can be at a spoke, but the multicast flows from the spoke, to the hub, and then back to the spokes who want to receive the traffic. Using PIM NBMA-mode enables the hub to forward the multicast only to those spokes that have receivers behind them, instead of to all spokes on a DMVPN network in which a single receiver has joined. In this case, you must set the shortest-path tree (SPT) threshold to infinity. This keeps PIM on the spoke from trying to send IP multicast over a direct spoke-to-spoke tunnel. It is not possible to send IP multicast over a spoke-to-spoke tunnel.

Headend Configuration

```
interface Tunnel1
 ip address 10.81.0.1 255.255.0.0
 ip pim nbma-mode
 ip pim sparse-mode
!
ip pim rp-address 10.81.0.1
ip pim spt-threshold infinity
!
```

Branch Configuration

```
interface Tunnel1
 ip address 10.81.1.1 255.255.0.0
 ip pim nbma-mode
 ip pim sparse-mode
!
ip pim spt-threshold infinity
!
```

DMVPN multicast scalability basically involves packet replication on the headend router and the throughput of the incoming stream. This can be very CPU-intensive, and performance degrades as the scale increases. For example, with a 256Kbps multicast stream and 100 spokes that have joined that stream, the hub must replicate each IP multicast packet 100 times. This requires the hub to encrypt and send 25.6Mbps (256Kbps*100), which can be a large portion of the hub encryption, outbound physical bandwidth, or Internet link bandwidth. It is recommended to use this with QoS.

Another point about IP multicast (including routing protocol multicast) is that DMVPN tends to send copies of the multicast packets extremely rapidly. This creates large bursts of packets that can overwhelm various queues or links from the VPN hub through to the ISP routers. Overall average utilization can be much less then the link bandwidth, but the burst effective bandwidth can be much higher. Burst effective bandwidth has been measured as high as 30–60Mbps. Therefore, the queues on all devices between the VPN hub router and the ISP routers must be large enough to handle these bursts so that packets are not dropped as they are waiting to be sent out a link that has lower bandwidth than the instantaneous burst bandwidth.

It is recommended to enable LLQ on the queuing policy-map. When LLQ is enabled on the outbound interface, a buffer is created between the processor and the Crypto card. This enables the crypto engine to buffer the burst of multicast traffic which otherwise would be dropped inbound by the crypto card. The multicast traffic does not need to be placed in the LLQ to get the benefit of the buffering. The mere fact that LLQ is configured on the outbound physical interface causes the buffer to be used.

```
class-map match-all BE
 match ip precedence 0
!
class-map match-all HP1
 match ip precedence 4
!
class-map match-all HP2
```



```
 match ip precedence 3
!
class-map match-all voice
 match ip precedence 5
!
policy-map child
 class HP1
  bandwidth 32
 class HP2
  bandwidth 32
 class voice
  priority 64
!
interface GigabitEthernet0/1
 ip address 10.70.100.2 255.255.255.0
 service-policy output child
 hold-queue 4096 in
 hold-queue 4096 out
end
```

3.6 Scaling Hierarchical Hub Deployment

Now that we have covered the basics of the hierarchical hub design, we can look at how this DMVPN design scales. It should be apparent that a routing protocol is required to manage the prefixes and reachability in this type of network. The scalability of a DMVPN hub depends upon the scalability of the routing protocol being used between the hubs and spokes. Scalability also depends upon the number of prefixes the routing protocol must advertise. To reduce overhead, we can advertise a default route or large summary to each of the spokes. The spokes are basically stub networks in relation to the DMVPN network as a whole, and are typically not used as transit networks.

Summarizing branch private subnets is recommended (if the subnet allocation was designed properly) at the headends. Obviously, not all routing protocols have the same scaling limitations or potential. We will look at two protocols that scale fairly well for DMVPN: Enhanced Interior Gateway Routing Protocol (EIGRP) and Routing Information Protocol (RIP). The core network between the hubs can be any protocol, but Border Gateway Protocol (BGP) is the recommended protocol between hubs and the rest of the campus network.

The other bottleneck that exists in all encrypted networks is the processing required to encrypt and decrypt. To maximize throughput, and conserve CPU processing power, we use hardware encryption modules in all DMVPN routers.

3.6.1 Data Plane Considerations

This section lists and describes data plane considerations, including encryption throughput, router CPU utilization, and unicast and multicast packet processing.

3.6.1.1IPsec Encryption Throughput (hardware encryption engines)

IPsec encryption engine throughput in each platform (headend or branch) must be considered for scalable designs, because every encrypted/decrypted packet must traverse the encryption engine. Therefore, encryption throughput must consider bidirectional speeds. In general, encryption throughput is reported as a value that includes both encrypting and decrypting packets. For example, if encryption cards throughput is

reported as 100 Mbps, the encryption card can encrypt and decrypt at a combined rate of 100 Mbps (100/0 Mbps (encrypt/decrypt) to 50/50 Mbps to 0/100 Mbps).

In general, as throughput increases, the burden on router CPU also increases. However, with hardware-accelerated encryption available for all Cisco router products from the 871 through the 7200, most encryption processing is offloaded from the main CPU to the VPN hardware. However, main router CPU processing still occurs, so higher throughput typically results in higher CPU consumption. For the 6500, 7600, and 7200 with VSA card, CPU processing per encrypted/decrypted packet is reduced further.

3.6.1.2Unicast Packet Processing

Although bandwidth throughput must be considered, the packet rate for the connection speeds being terminated or aggregated is more important. In general, routers and encryption engines have upper boundaries for processing a given number of packets per second (pps). The size of packets used for testing and throughput evaluations can understate or overstate true performance. It also turns out that encryption engines use about the same amount of resources to encrypt a large packet as to encrypt a small packet. So, the pps that an encryption card can handle tends to stay constant as the size of packets varies.

Because of such a wide variance in throughput, pps is generally a better parameter to determine router forwarding potential than bits per second (bps). Headend scalability is the aggregate forwarding potential of all branches that terminate a tunnel to that headend. Therefore, the aggregate pps from all branches affects the headend pps rate.

Throughput varies per platform, and depends largely on the traffic pattern. Recommendations in this chapter are based on a nominal Internet mix (IMIX) traffic pattern, with router CPU utilization averaging 70%.

3.6.1.3Multicast Packet Replication

Multicast traffic affects a router more than unicast processing. This effect is a function of the number of PIM receivers. If multicast is used in the DMVPN network, the design should include fewer receivers per hub than would be used with unicast.

3.6.2 Data Plane Best Practices

You should plan your design for above-average throughput. This prevents surprises when traffic bursts and the CPU does not have enough cycles to process the control plane packets.

It is also recommended to follow these best practices:

- IP MTU – Set the IP maximum transmission unit (MTU) to 1400 on all DMVPN tunnel interfaces to eliminate the potential for fragmentation. GRE and IPsec headers add about 60 bytes to the packet, and cause the router to fragment larger packets if this exceeds the interface MTU, straining the CPU.
- TCP MSS – Set the TCP maximum segment size (MSS) value to 1360 on all DMVPN tunnel interfaces. This value is calculated by subtracting 40 bytes from the IP MTU value. Use the command **`ip tcp adjust-mss 1360`** to set the value on the mGRE tunnel interface toward the spokes. This helps TCP sessions adjust to the lower MTU and is needed if Path MTU Discovery (PMTUD) does not work between end hosts.
- Tunnel Bandwidth – The **`bandwidth`** statement is recommended on the tunnel interface of hub and spoke routers, although the actual value can vary from scenario to scenario. Without the **`bandwidth`**

statement, tunnel interfaces are currently allocated very low interface bandwidth (9 Kbps). The default will soon be raised to 100 Kbps. This could affect QoS or other features, such as routing protocols that use the configured bandwidth.

- Hardware Crypto acceleration – Always use a hardware encryption module to do most of the encryption/decryption math. The headend and the spokes should have hardware encryption engines.
- QoS – Provision Quality of Service (QoS) policies as necessary at the headend and branch routers. This helps alleviate interface congestion and ensures that latency sensitive traffic is prioritized over other traffic. If classification is done on the internal packet, you must configure **`qos pre-classify`** on the tunnel interface.

3.6.3 Control Plane Considerations

This section lists and describes control plane considerations, including tunnel aggregation stability, encryption throughput, routing protocols, route summarization, and stub routing.

3.6.3.1Tunnel Aggregation Scalability

You must consider the maximum number of IPsec tunnels that a headend can terminate. Tunnel scalability is a function of the number of branch routers that are terminated to the headend aggregation point. This number must include the primary tunnels and any alternate tunnels that each headend might be responsible for in the event of a failover.

The number of IPsec tunnels that can be aggregated by a platform, and the encryption pps rate, are the primary determining factors for recommending a platform.

Although throughput depends highly upon platform architecture, as tunnel quantities increase, overall throughput tends to decrease. When a router receives a packet from a different peer than the peer whose packet was just decrypted, a lookup based on the security parameters index (SPI) of the new packet must be performed. The transform set information and negotiated session key of the new packet is then loaded into the hardware decryption engine for processing. For traffic to be encrypted, the main CPU must match the data packet to the correct IPsec security association (SA) to hand to the encryption engine to encrypt the packet. Having traffic flow on a larger numbers of SAs tends to negatively affect throughput performance.

Increasingly, platforms with hardware-accelerated IPsec encryption are also designed to offload IPsec SA processing overhead, resulting in more linear performance regardless of the number of SAs. For example, the VPN SPA blade for the Cisco 7600 has fairly linear throughput whether the traffic load is offered on a few SAs or several thousand. Currently, the VPN SPA (6500/7600) and the VSA (7200 NPE-G2) are the only encryption processors that have this functionality.

The GRE configuration affects router encryption throughput. GRE headers are added to the beginning of each packet, and GRE headers also must be encrypted. The GRE encapsulation process, when not hardware-accelerated, increases total CPU utilization. In most cases, the amount of CPU overhead due to GRE encapsulation is quite small compared to the CPU overhead due to IPsec encryption.

3.6.3.2Routing Protocols

Running a routing protocol affects CPU overhead. Processing hello packets and maintaining a routing table uses CPU time. The amount varies with the number of routing peers and routing table size. The network

manager should design the routing protocol based on generally accepted practices for the particular routing protocol.

Routing protocol scalability is the most important limiting factor in determining the number of branches a given hub can support. Scalability becomes more cumbersome as the number of routes that the protocol must advertise to each branch increases. We will look at EIGRP and RIPv2, along with various options that can be used to increase scalability.

3.6.3.3Route Summarization

Hub routers will forward data traffic based on routing tables. Based on the routing table, summarization can occur in the core of the network, and in the spoke networks. This enables us to build a network with a hierarchical design having smaller, more efficient routing tables. In general, in a hierarchical network, at each level the routing protocol should summarize routes going up the tree toward the spokes (leaves), but send full routing information to nodes in the same level (region) or down the tree (toward the root).

Most dynamic routing protocols support summarization. However, you might want to run a separate routing protocol, such as BGP, between hub routers within the same region or within the core. Regional hub routers must have full routing knowledge of their regions. The regional hubs can summarize their routing tables when advertising them to the spokes and other hubs.

The protocols should be configured to advertise very few routes toward the branch routers. This should, in most cases, be a default route, or, in few cases, a summary route for the rest of the network. If a branch router uses split tunneling, a default route should not be advertised to the branch unless the branch uses VPN routing/forwarding instances (VRFs), in order to segregate split-tunneled traffic.

3.6.3.4Stub Routing

Branch routers are rarely used as transit networks. Therefore, it is best to configure branch routers as stub networks. Using EIGRP, you can set the routing protocol to behave as a stub; it announces this functionality in hello packets to the headend router. Using the **`stub connected`** command, the branch router sends connected prefixes to the headend. The headend has reachability of the branch networks, and does not have to include the branch router in the query process.

Note: Be careful to not advertise the tunnel source interface subnet to the hub router through the tunnel interface. If the hub uses this route the tunnel packets to this spoke will go into a forwarding loop that will either hang or crash the hub router. To protect against this possibility, either block this network from being advertised by the spoke or being accepted by the hub.

Spoke Configuration

```
router eigrp 10
 network 10.80.0.0 0.3.255.255      ← DMVPN Tunnel subnets
 network 10.211.9.0                 ← Private VPN subnet
 no auto-summary
 eigrp stub connected   ← Announces stub router, advertise connected intf.
!
```

In RIPv2, there is no stub keyword. This protocol must operate in the native manner. Alternatively, you can make the mGRE tunnel on the hub passive and use reliable static routing on the branches. In this case, the hub still receives RIP advertisements from the spokes, but does not send the spokes RIP advertisements.

Reliable static routing on the branch router uses the IP service level agreement (SLA) feature to poll an address on the primary headend using a ping probe, coupled with a tracking object tied to a static route.

As long as the ping probe operates, it installs a static route directing traffic toward the primary headend over the tunnel. If the probe is nonoperational, the static route is removed and a floating static route having the same network and mask, but higher administrative distance (AD), directs traffic to the secondary headend.

3.6.4 Control Plane Best Practices

The following options should be used to increase stability of the design:

- Protocol Timers – Set the routing protocol dead timer to a larger value. This provides more time for the WAN to recover in the event of a minor problem. Otherwise, a minor flap could cause Interior Gateway Protocol (IGP) running over the tunnels to go down and require reconvergence. Also, consider leaving the hello timer the same, rather than increasing it. This helps keep the routing protocol from flapping if the WAN network is somewhat lossy.
- Interface Queues – Set the interface input and output hold queues to 4096 on the tunnel interface and output WAN interface of the headend, and possibly other routers on the path from the headend to ISP router, to minimize protocol failure.
- PKI and Digital Certificates – Use digital certificates and Public Key Infrastructure (PKI) for the ISAKMP authentication. This increases manageability as the site count increases. Preshared keys (PSK) are not a scalable solution for distributing Internet Key Exchange (IKE) authentication credentials. PKI is highly scalable and secure. Be sure to turn off certificate revocation list (CRL) checking because it periodically drains CPU power.
- Dead Peer Detection – Enable dead peer detection (DPD) on hubs and spokes. This increases the speed at which a failure over the IPsec tunnel or connectivity loss is detected. This should be set higher than the routing protocol update timer, but less than the dead timer.
- Set the **`crypto isakmp keepalive`** timer value to be greater than the RP hello timer by 5 seconds. Set the total ISAKMP keepalive + 5* retry timer for a timeout equal to RP dead timer + 60 seconds. This prevents ISAKMP and IPsec sessions from tearing down before losing routing protocol adjacency.

```
crypto isakmp keepalive <hello+5 (sec)> <(dead+60-(hello+5))/5)(sec)>
```

Example: Hello timer = 10 secs and dead timer = 60 seconds; then keepalive = 15 (10+5) and retry = 21 (60+60-(10+5))/5

```
crypto isakmp keepalive 15 21
```

- Call Admission Control on hub – In failure scenarios, IKE processing can be debilitating for a router. It is recommended to enable Call Admission Control (CAC) to limit the number of IPsec sessions that can come up simultaneously. This prevents incoming IPsec connections from overrunning the CPU. There are two CAC methods

  Configure the absolute IKE SA limit so the router drops new IKE SA requests after the number of IKE SAs in negotiation reaches the configured limit. After the number of IKA SAs in negotiation drops below the configured limit, additional IKE SAs are processed.

```
crypto call admission limit ike in-negotiation-sa [number]
```

Configure the system resource limit so the router drops new IKE SA requests when the specified percentage of system resources is in use. It is recommended to set this at 90% or less.

```
crypto call admission load [percent]
```


3.6.5 Scale Limits

This section examines specific DMVPN scaling limits of the routing protocols and hardware crypto engines. Remember, throughput is not a focus, but a set parameter to load the system to a high operating load. The following sections illustrate the DMVPN peer scalability for the hierarchical hub design using a Cisco 7200 router serving as the headend.

It is very important to understand how scaling numbers are found and what to expect. All internal testing to derive the numbers was done in a lab, so it is likely the numbers stated are optimal for the design. The scalability is determined by both the number of DMVPN peers the design can support, and the data throughput at the maximum scale. In the following data, notice the number of routing protocol peers does not necessarily increase with a better processor or encryption card. Rather the scale of the design is consistant, and the amount of throughput increases as more powerful hardware is used.

There have been great improvements in the scalability of the routing protocols for DMVPN. Depending on the IOS image your router is running, scalability varies. It is recommended to run either IOS 12.4(9)Tx or 12.4(15)T2 and above with the Advanced Enterprise Security (adventerprisek9) feature set. IOS 12.4(11)Tx *should not* be run with DMVPN and QoS.

Note: These results are based on IOS 12.4(15)T4 with the Advanced Enterprise Security (adventerprisek9) feature set.

3.6.6 EIGRP

Cisco recommends using EIGRP as the routing protocol over your DMVPN network. EIGRP offers fast convergence and robustness. It is recommended to configure the spokes to be EIGRP stub peers. This minimizes convergence times and simplifies the convergence process. Cisco also recommends using the summarization potential offered with DMVPN Phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes.

Looking at the 7200 Series router as the headend, there are many hardware variations you can use. For the route processor, you should use either NPE-G1 or the NPE-G2. If you use an NPE-G1, you must use the VAM2+ for the encryption port adapter. When using the NPE-G2, you can use either a VAM2+ or a VSA.

Table 3-1 shows the recommended and maximum limits for EIGRP peers per headend with each hardware combination.

Table 3-1. Recommended and Maximum EIGRP Peers

EIGRP Peers	Recommended	Maximum*
7200 NPE-G1 w/VAM2+	600	900
7200 NPE-G2 w/VAM2+	600	900
7200 NPE-G2 w/VSA	700	1000

*The Maximum column indicates the maximum peer limit in an ideal environment such as a lab. It MUST be understood that this is probably not going to be obtainable in a real live network where there is more delay and other variables that adversely affect routing protocol convergence. Also, even though the NPE-G2 w/VSA card does not increase the number of supported EIGRP peers by very much, the card it greatly increases aggregate encryption pps and bps rates.

Note: These numbers assume that summarization to the spokes is used and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is around 70% load on the

router. If your network parameters differ, your results may vary from the reported numbers, which are provided as planning guidelines.

During times of convergence EIGRP will generate much more traffic then during periods of network quiescence. The following table lists the approximate peak amount of traffic in bits per second that EIGRP will generate outbound over the WAN for a given number of spokes and prefixes advertised towards the spokes. This includes the encryption overhead of the DMVPN tunnel. These numbers can be used to help determine the size of the WAN link to the Internet. Actual data traffic will of course add to these numbers. EIGRP will in all likelihood converge even if the WAN link is smaller than the given value but there will probably be some packet loss due to outbound packet drops.

As can be seen in the table, the number of prefixes advertised does not have a great impact on the traffic rate. The fact that the single prefix numbers are greater in some cases than the fifty prefix numbers is probably due to sampling errors. The inbound traffic rates are generally much lower.

Table 3-2. Minimum WAN banwidth needed by EIGRP for a given number of peers and prefixes advertised

Number of EIGRP Peers	Max. BPS for 1 Prefix Advertisement	Max. BPS for 10 Prefix Advertisement	Max. BPS for 50 Prefix Advertisement
600	2887000	2963000	3669000
1200	10075000	12886000	15752000
1800	20577000	25939000	27732000
2400	28717000	29845000	34397000
3000	36329000	35997000	36672000
4000	55736000	55056000	55004000
5000	69670000	68820000	68755000
6000	83604000	82584000	82506000

3.6.7 RIPv2

Cisco supports RIPv2 as the routing protocol over a DMVPN network. RIP is a simple, effective protocol in a DMVPN network. RIP can scale very high compared to other routing protocols over DMVPN. It is recommended to use the summarization potential offered with DMVPN phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes. Using the passive interface with reliable static routing (basically stub branches in RIPv2) did not provide any benefit compared to native RIP. The reason for this is simply the input queue limits the number of updates creating a minor bottleneck. So, regardless of active or passive RIP, the updates will still be sent from the spokes every 30 seconds. In production networks, the updates sourced from the spokes should not have the same synchronization as we see in a lab environment.

Looking at the 7200 series router as the headend, you can use many hardware variations. For the route processor, you should use either the NPE-G or the NPE-G2. If you use an NPE-G1, you will be required to use the VAM2+ for the encryption port adapter. When using the NPE-G2, you can use either a VAM2+ or a VSA. The following tables show the Cisco recommended limit of RIP peers per headend with each hardware combination.

Table 3-3. Recommended and Maximum RIPv2 Peers

RIP Peers	Recommended	Maximum*
7200 NPE-G1 w/VAM2+	1000	1125
7200 NPE-G2 w/VAM2+	1000	1125
7200 NPE-G2 w/VSA	1000	1125

* The Maximum column reports the maximum possible peer limit in an ideal environment, such as a lab. It *must* be understood that these results are unobtainable in a real network, where more delay and other variables adversely affect the solution. Although the NPE-G2 w/VSA card does not increase the number of supported RIP peers by much, the card does greatly increase the aggregate encryption pps and bps rates.

Note: These numbers assume the recommendations of summarization to the spokes is used,and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is around 70% load on the router. If your network parameters differ, your results might vary from the reported numbers, which are provided as planning guidelines.

3.6.8 Unicast Throughput

In this design, the headends handle all crypto processing, and the VAM2+ or VSA handles the crypto functions for a given headend. Because the encryption card, and not the router CPU, limits throughput, it is important to know the throughput limits of the encryption cards.

The throughput values in the following table show the aggregate throughput of each hardware combination on one 7200 headend. The table shows maximum throughput of bidirectional data traffic for each 7200 processor at 70% CPU utilization, forwarding an IMIX pattern of traffic (7×64:4×570:1×1400).

Table 3-4. Bidirectional Unicast Throughput

Bidirectional Throughput	PPS	Approx. Mbps
7200 NPE-G1 w/VAM2+	24,000	64
7200 NPE-G2 w/VAM2+	42,000	116
7200 NPE-G2 w/VSA	60,000	320

Note: These values show the approximate maximum aggregate throughput for the given processor and crypto card conbination. The VSA or VAM2+ is the throughput bottleneck in this design.

Note: The bandwidth numbers *do not* represent real traffic, but are found by simulating a realistic traffic pattern in the lab. Real traffic does not always follow a given pattern, so throughput values in a real network differ from those found in lab testing. Also, these numbers assume throughput at a maximum CPU load of 70% on the router. If your network parameters differ, your results might vary from the reported numbers, which are provided as planning guidelines.

3.6.9 Multicast Throughput

In a DMVPN network, the headends handle all multicast replication processing, and the VAM2+ or VSA handles all crypto functions after replication. Throughput of the input stream is magnified by the number of peers receiving the stream.

The code and configuration were identical on the NPE-G1 and the NPE-G2. It surprised us that the NPE-G1 with VAM2+ performed much better in multicast forwarding than the other platforms. The VAM2+ dropped packets when paired with the NPE-G2 because of **`ppq full`** errors. The theory is that the NPE-G2 replicates the multicast stream so fast that it overruns the crypto card buffers.

Table 3-5. Multicast Throughput

Processor and Crypto Card	Maximum number of Multicast Replication without Packet loss	Average Interrupt CPU Utilization on UUT	Total UUT Forwarding PPS	Total Hub Router Forwarding (bps) - Ignoring Crypto and GRE Overhead

NPE-G2 with VSA	220	11%	8,140	8,335K BPS
NPE-G2 with VAM2+	100	7%	3,700	3,788K BPS
NPE-G2 with VAM2+ and LLQ on Outbound Interface *	800	93%	29,600	30,310K BPS
NPE-G1 with VAM2+	660	90%	24,420	25,006K BPS

* To deal with the VAM2+ limited buffer space, configure QOS with low latency queuing (LLQ) on the physical outbound interface. The multicast traffic does not need to be placed into the low latency queue, nor does it require a bandwidth statement to reserve bandwidth for the multicast traffic. Simply enabling LLQ on the outbound interface causes IOS to buffer the data going to the crypto card so that it can perform QOS before crypto. The QOS before crypto feature engages when the crypto card is congested, not when the physical outbound interface is congested. This strategy is *not* expected to work for the NPE-G2/VSA combination.

4 DMVPN IOS Server Load Balancing Design

The IOS server load balancing (SLB) design increases the scale, ease of deployment and management of the DMVPN headend site. It is a single-layer hub design; it uses only one layer of DMVPN hubs. In this design, we use the IOS SLB feature on a router to distribute the load (tunnels) across DMVPN hubs. The design provides incremental scalability with the addition of headend routers as needed without changes to any spoke routers.

In this design, we are more focused on the DMVPN hub and how well it scales. A single router has a limited amount of resources to terminate tunnels from the spoke routers. To increase the scalability of a single hub site, we can use SLB to redirect connections to a "server farm" of DMVPN hub routers. In treating the DMVPN hub router as a server, the IOS SLB router can distribute the load across a group of routers. This design is highly scalable and extensible. As the number of spokes increase, more hubs can be added to handle the additional load. This design offers *n*+1 redundancy for the hub routers, and supports redundant SLB routers

Another advantage, of this design is that all spokes are configured as if there is only one DMVPN hub using the IOS SLB virtual IP address (VIP). IOS SLB dynamically distributes these spoke tunnels across the DMVPN hub routers. Load balancing of the tunnels automatically adjusts across the remaining headend routers if a headend router fails. With the IOS SLB design, DMVPN hub routers can be added and removed without having to reconfigure of any spoke routers. The alternative to this would be to manually configure the spokes to terminate their tunnels distributed across the set of hub routers. If mapping spokes to primary and secondary hubs is done manually (not using this design), 2*n* hub routers are required for the equivalent redundancy.

The SLB design also works well with common routing protocol designs. Because the SLB design is a hub and spoke topology, the hubs all independently run the same routing protocol. Therefore, a well planned topology and addressing scheme can support a well summarized set of routes. This keeps the routing tables on all devices as compact as possible, and growth of the network can be considered.

In this design, headend routers in the server farm must encrypt/decrypt all traffic entering and leaving the DMVPN central site. The size of router chosen for a DMVPN headend must be able to handle the number of routing protocol neighbors (spokes), and the potential encrypted throughput load placed on it from the spokes. In the SLB design, the total load is divided among the server farm, but the consideration for redundancy must still be taken into account.

The IOS SLB device only balances connections to the specific headend routers. As the number of hub routers increase, the additional processing power of the added hubs with their hardware crypto engines will tend to mitigate this disadvantage by increasing the crypto processing power of the entire server farm. Another minor limitation is that the hub routers must be directly connected to the server load balancer, which can be a 6500 or 7200 router running 12.2 IOS software, for IOS SLB functionality to support ISAKMP and IPsec.

Next, we look at common deployment scenarios for this design.

4.1 Deployment Scope

This design provides high availability (HA) and incremental scalability, along with high spoke scalability per hub site. The scalability is easily extended by adding extra headends to a server farm. If a hub device

fails, the connections are reformed and distributed across the remaining headend devices. This design is a single layer design and is typically used in simple large scale deployments, with low to moderate throughput.

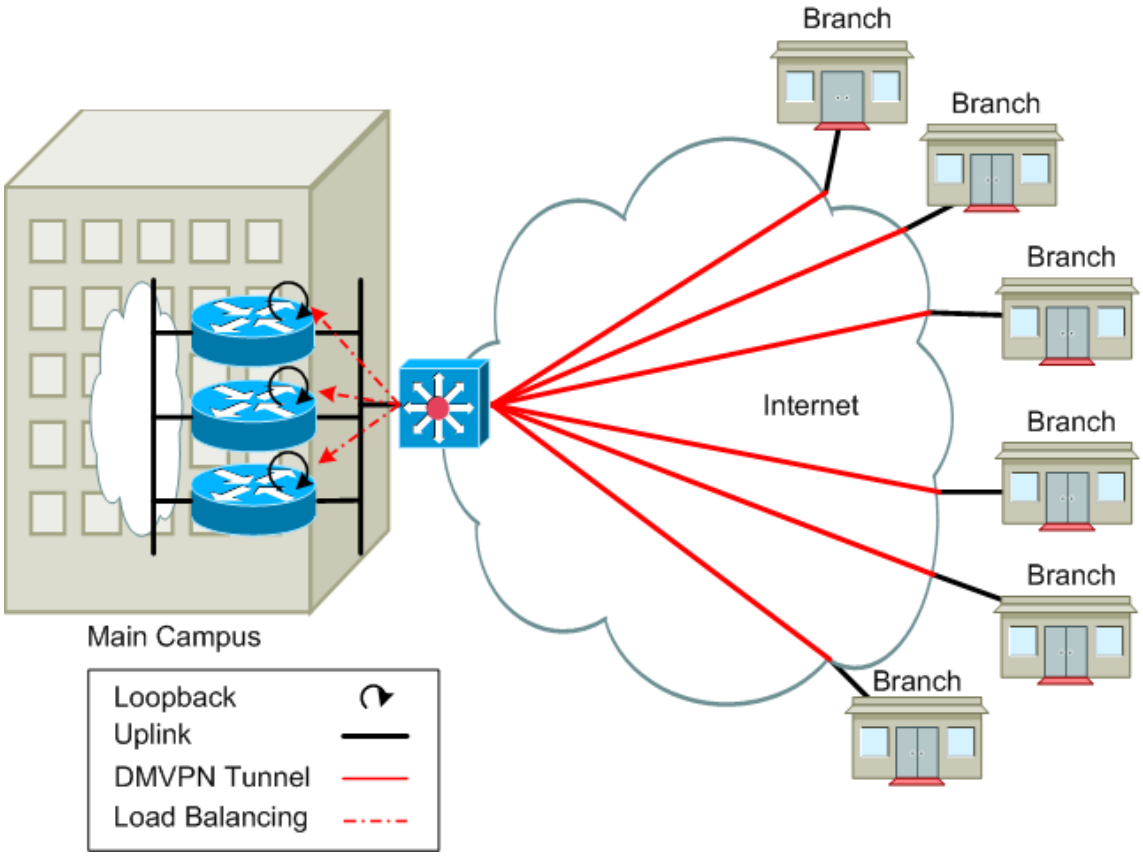
Note: IOS 12.2 running on the 6500 or 7200 is the only version that currently supports SLB for IPsec tunnels. In the IOS 12.4 releases, the **esp** keyword is not available for virtual server configurations. Furthermore, the probes and fail actions are not available. The 6500 is the recommended platform for this for this design. However, a 7200 with the older 12.2 image also works.

Two IOS SLB servers can also run in stateful failover mode so that a secondary IOS SLB device can immediately take over if a primary ISO SLB device fails. If there are two hub sites (one a disaster recovery (DR) site), a separate IOS SLB device with its own set of headend routers is deployed at the DR site. The spokes are configured to have two tunnel interfaces, one to the primary site and one to the DR site.

This deployment is useful with point-of-sale devices (such as point-of-sale (POS) and credit card terminals, gas pumps, and so on) or banking automated teller machines (ATMs). A large number of devices connect to the headend securely, but throughput is fairly low. In these deployments, the POS devices or ATMs would be the DMVPN branch. At the headend site, a number of headend routers handle the connections, and SLB distributes connections to the headend routers. This design has built-in redundancy to ensure that connections can always be handled.

Another common deployment could be when an enterprise has remote or mobile employees. The common enterprise class teleworker (ECT) can use DMVPN to securely connect small offices or home offices (SOHO) to the enterprise. These offices would typically be in the same geographic region, and would connect to the nearest headend site. This is most common in large enterprises, where the internal campus network handle all the routing throughout the network.

Figure 4-1. DMVPN IOS Server Load Balance Deployment

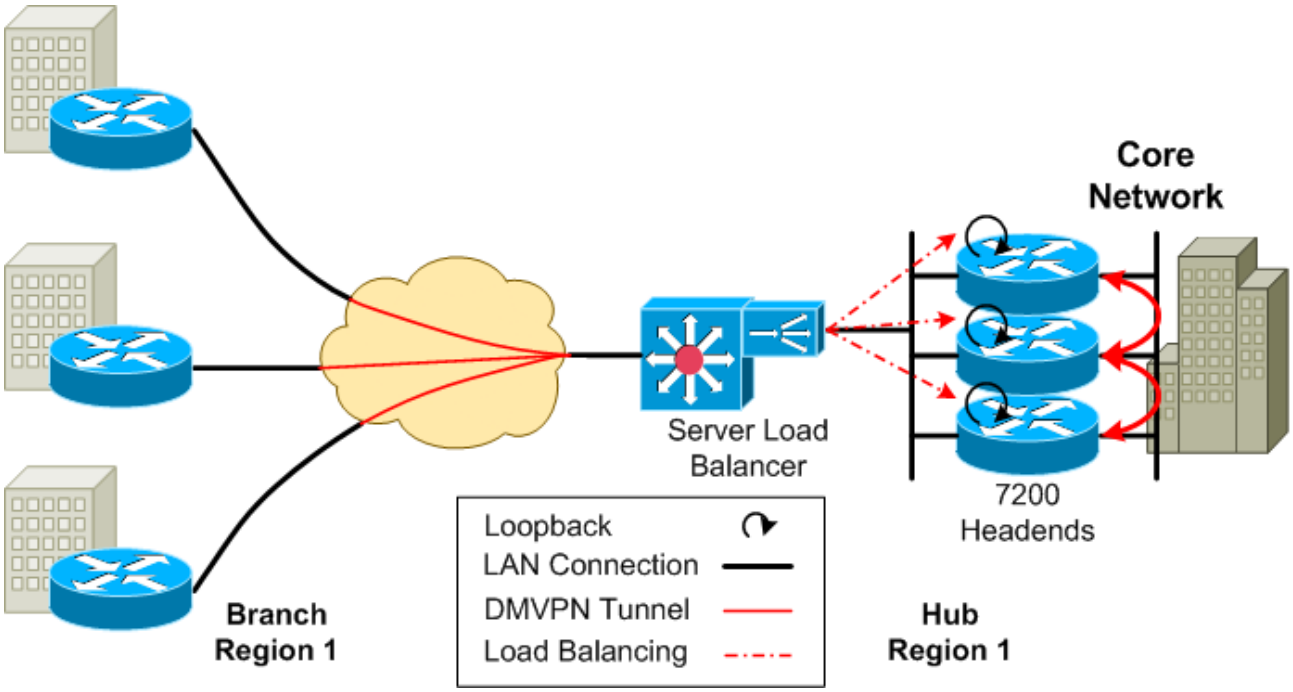


4.2 Basic Operation

In this design, the DMVPN hubs (headends) directly connect behind one or more IOS SLB routers. All branch routers are configured with the destination of SLB virtual IP address (VIP). SLB uses the public IP address of the IOS SLB device as the virtual IP address (VIP) for load balancing and is also configured on a loopback interface of each DMVPN hub router. These loopbacks on the headend routers are used as the source IP address for the multipoint generic routing encapsulation (mGRE) tunnel interface. The headends terminate the GRE session and perform crypto and Next Hop Routing Protocol (NHRP) processing. The headend also handles forwarding or processing (for example, the routing protocol) of the inner IP data packet.

The IOS SLB function in the edge router makes decisions to evenly distribute DMVPN tunnels to the various headend routers based on configured parameters. If a particular headend fails, IOS SLB redistributes the tunnels that were processed by the down headend to the other headends. The convergence time for the redistributed tunnels to become fully active is determined by the amount of time required for IPsec tunnel setup and NHRP registration processing to complete on the new headend router.

Figure 4-2. DMVPN IOS Server Load Balance System Architecture



4.2.1 Using Border Gateway Protocol to Provide Full Routing Knowledge

In the SLB design, the routers in a server farm must share full routing knowledge. This is important because spoke connections are randomly distributed to these routers. Subnet information from a given set of branches that a headend router terminates is not obvious to the other headends, so they must all share specific routing information for the spoke networks.

Headend routers in the server farm communicate with each other using an mGRE tunnel. It cannot be the same tunnel interface the spokes terminate on, but it must have the same NHRP network-ID. This allows NHRP to trigger a redirect for the creation of spoke-to-spoke tunnels when the spokes are being serviced by different hub routers.

For scalability, hubs peer with each other over Border Gateway Protocol (BGP), using the headend as route-reflector. Routes learned from a spoke that connects to a hub become present on other hubs. Two basic designs achieve this: folded and unfolded. Regardless of the method used to reflect the routes, redistribution between BGP and the core routing protocol must be performed to propagate the routes to the other headends and the rest of the core network.

The folded design uses the device or devices providing SLB functionality. The SLB device is a good choice since it is already connected to all of the headend routers. Configure an mGRE tunnel interface and peer with the headend routers through the same physical interface that load balances the hub-to-spoke tunnels. The SLB device does double duty and extra routers are not required to handle the route reflection in this case. An example of the 7200 SLB device acting as a BGP route reflector is provided in Appendix A.2.2, “7200 SLB Configuration with BGP Route Reflection.”

The unfolded design requires extra routers between the headends and the rest of the internal network. These routers only perform route reflection over the mGRE tunnels to the headends and connect the DMVPN network (clear text side) to the internal network. This is straightforward to do, and using an extra router that is used only as the BGP route reflector. This method is not covered in detail.

For redundancy, it is recommended to have at least two route reflectors in either design. Even if BGP is not used to provide the full reachability among the headends, the routing protocol between the SLB device (folded) or this other router (unfolded) and the core network, should use the same routing protocol that is used in the core network.

4.2.1.1Folded BGP Route Reflector Design

In the folded design, the 6500 or 7200 handling the SLB function is the BGP route reflector. The tunnel on the SLB device is sourced from the VLAN SVI (interface VLAN 10 in the following example), which connects to the spoke physical interfaces. The BGP weight is set to the maximum for headend clients so that the headend installs and propagates routes learned from its clients, instead of a possible BGP peering with a remote hub. Each device on the tunnel has a static NHRP entry for the other devices on the tunnel. This is more reliable than using dynamic updates. All involved IP addresses are static.

SLB Device Configuration

```
interface Tunnel2
 description ****To Route Reflector Clients****
 ip address 10.91.0.1 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp map multicast dynamic
 ip nhrp map 10.91.0.2 10.74.100.2
 ip nhrp map 10.91.0.3 10.74.100.3
 ip nhrp network-id 102
 ip nhrp cache non-authoritative
 tunnel source Vlan10
 tunnel mode gre multipoint
!
router bgp 10
 no synchronization
 bgp router-id 10.70.201.2
 bgp log-neighbor-changes
 neighbor 10.91.0.2 remote-as 10
 neighbor 10.91.0.2 route-reflector-client
 neighbor 10.91.0.2 weight 65535
 neighbor 10.91.0.3 remote-as 10
 neighbor 10.91.0.3 route-reflector-client
 neighbor 10.91.0.3 weight 65535
 no auto-summary
!
```

Note: The BGP router ID on the individual headends must be set because its loopback address is the same as all other hubs connected to the same headend. Without this configuration command, there is a strong possibility that the hubs will have duplicate BGP router IDs.

The hub advertises its routes to the route-reflector with the next hop self set. Without this command, the next hop would be the tunnel IP address of the spoke that the route was learned from. This would prevent the other hubs from reaching these destinations. Routes from the spokes are learned through Enhanced Interior Gateway Routing Protocol (EIGRP) and redistributed to BGP.

Hub Configuration

```
interface Tunnel2
 description ****Tunnel to BGP Peering with RR****
 ip address 10.91.0.3 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp map 10.91.0.2 10.74.100.2
 ip nhrp map 10.91.0.1 10.74.100.1
 ip nhrp network-id 102
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
!
router bgp 10
 bgp router-id 10.91.0.3
 bgp log-neighbor-changes
 redistribute eigrp 10
 neighbor 10.91.0.1 remote-as 10
 neighbor 10.91.0.1 next-hop-self
 no auto-summary
!
```

4.2.1.2Unfolded BGP Route Reflector Design

In the unfolded design, separate routers act as route-reflector instead of the SLB router. Configuration is nearly identical. Peering with the route-reflector-clients must occur over an mGRE interface having the same NHRP network-ID as the spoke-facing tunnels. This supports direct spoke-to-spoke tunnels for spokes that terminate on different hubs. The main difference is the requirement for an extra, separate router.

4.3 HA Design

HA design provides network resilience and availability in the event of failures. To provide resiliency in the DMVPN SLB design, Cisco recommends using an *n*+1 server farm of headend routers. You should have enough routers to handle the scale of your intended spokes (*n*), and an additional headend router (+1) in the event of a failure. Obviously, you can use *n*+2 or more for even more HA, but *n*+1 is the recommended minimum.

The value of *n* in the formula is derived by dividing the number of anticipated DMVPN peers by the scalability of one of the headend devices you are going to use. Before designing your DMVPN hub site, you need to know the type of headend routers you will use and the DMVPN scalability of the router. Remember that you can always add headends to the server farm later, so future expansion is easy.

A common concern in all HA headend designs are the number of routing protocol neighbors. Many redundant neighbor relationships increase the number of advertised prefixes and consequently increase the time required for routing convergence. Routing protocol convergence is a common element in all DMVPN headend designs. However, in the SLB HA model, the burden on the routing protocol is divided among the headend devices, providing greater site scale. However, each headend router is limited individually by a given scale limit. We look at this later in Section 4.6, “Scaling IOS SLB Hub Deployment.”

4.3.1 Single Site HA Design

To remove the SLB router as a point of failure, the single site DMVPN SLB HA design uses two 6500 routers with the stateful IOS SLB configuration. To provide HA, Hot Standby Router Protocol (HSRP) provides redundancy and SLB table synchronization, making SLB configuration stateful. HSRP runs on the public interfaces and on the private server farm interfaces. The two HSRP instances must work together to stay synchronized. One HSRP instance must also keep SLB state synchronized so that either 6500 is ready in the event of a failure. Because SLB tunnel distribution is unique, maintaining SLB state between the routers supports seamless failover.

The public HSRP instance provides a unified Layer 3 (L3) endpoint (VIP) to the Internet. Furthermore, SLB state is synchronized using the public HSRP instance. The server farm HSRP instance provides a common default gateway for traffic going from the hubs toward the spokes.

SLB has the same configuration on each 6500, and is configured with replication to ensure that active tunnel information is shared correctly. HSRP is also configured with port tracking to ensure that if a physical connection on the public or private side of the SLB routers is lost, the other load balancer takes over to prevent any loss in traffic. This means both HSRP instances switch over at the same time. At any given time, one load balancer is in "operational" state and the other is in "standby" state.

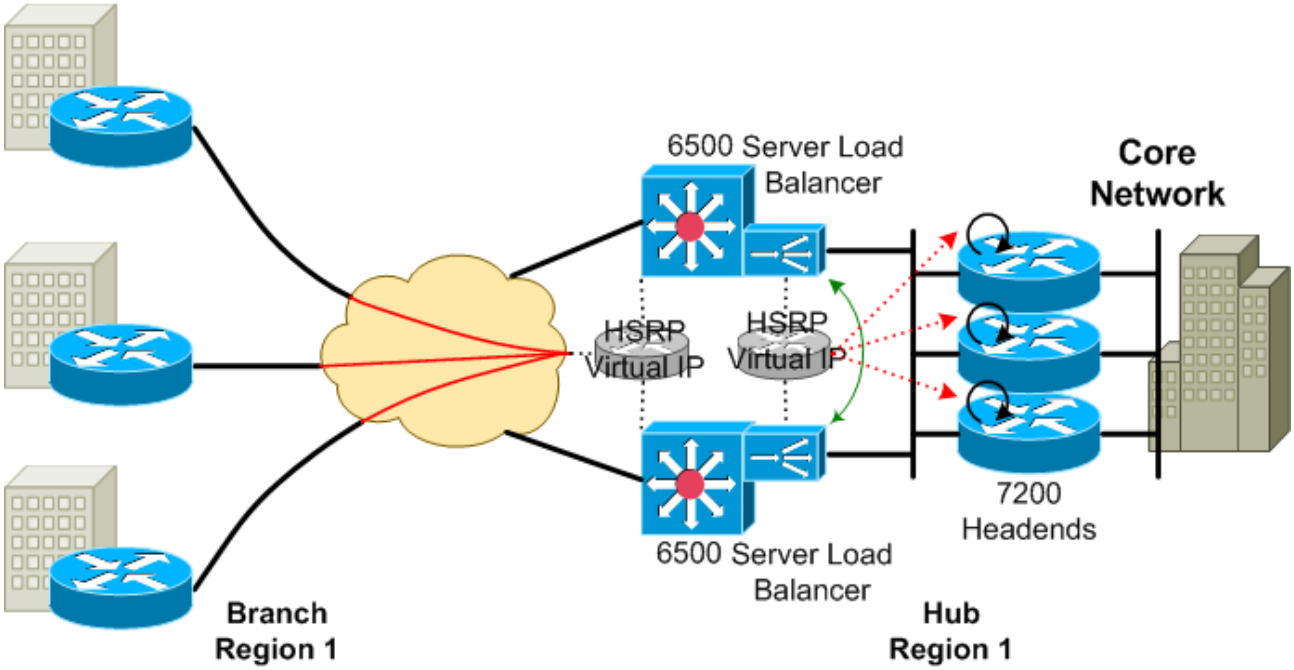
SLB is also configured with two *vservers*, representing ISAKMP and Encapsulating Security Payload (ESP). These virtual server classifiers are configured as *sticky*, to ensure that the ISAKMP (IKE) connection (UDP port 500 or 4500) and later the encrypted data packets (ESP – IP protocol 50) are mapped to the same physical headend router.

SLB is configured with a virtual server IP address. This is the same as the DMVPN next hop server physical (NBMA) IP address the spokes point to in their DMVPN tunnel configuration. This virtual server does not terminate the DMVPN tunnel. Rather, it intercepts the ISAKMP and IPsec (DMVPN packets) and load balances them to the DMVPN headend server farm. The actual headend routers terminate IPsec and GRE for the DMVPN tunnel. Each headend is configured with a loopback interface that has the same IP address as the SLB virtual server IP. This loopback interface is the tunnel source on the headend tunnel interface.

In Figure 4-3, the branch routers build their IPsec tunnels to the loopback address of the headends. This requires the 6500s to advertise this address to the public network. Because the 6500s run HSRP between the public interfaces, the next hop in the advertised route is the public VIP. The tunnels are physically routed to the active 6500. Because the private HSRP instance is in sync with the public HSRP instance, SLB on the active 6500 distributes the tunnels using L2 forwarding to a headend router. The headend router accepts the packets because it has a loopback interface with the same IP address as the VIP. SLB table state is sent to the secondary 6500 so it is aware of the SLB state if the primary 6500 fails.

To enable SLB state replication to keep the SLB states in sync, you must configure the **`replicate casa {source address} {destination address} {port}`** command under each **`vserver`** definition. The source and destination are the physical interface addresses of the SLB router inside interface. For SLB and IPsec, it is also important to configure connection purging in case of server failure. You can enable IOS SLB to automatically purge connections to failed real servers and firewalls from the connection database, even if the idle timers have not expired. This function is useful for applications that do not rotate the source port, such as IKE, and for protocols that do not have ports to differentiate flows, such as ESP. This is configured under the server farm with the **`failaction purge`** command.

Figure 4-3. Single IOS SLB Site Design



6500 SLB configuration is straightforward. The actual headends are the real servers, and the virtual server corresponds to the loopback interface for terminating the DMVPN tunnel. SLB routers are configured almost identically, except for the specific interface IP addresses. Therefore, we look at only one example.

SLB Router Configuration

```
ip slb probe PING-PROBE ping
 faildetect 3
!
ip slb serverfarm 7206-FARM
 failaction purge
 predictor leastconns
 probe PING-PROBE
 !
 real 172.16.1.4
  weight 1
  inservice
 !
 real 172.16.1.5
  weight 1
  inservice
!
 real 172.16.1.6
  weight 1
  inservice
!
ip slb vserver ESP
 virtual 10.40.0.6 esp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 replicate casa 172.16.1.2 172.16.1.3 60001 password NSITE
 inservice standby standby115
!
```

```
ip slb vserver ISAKMP
 virtual 10.40.0.6 udp isakmp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 replicate casa 172.16.1.2 172.16.1.3 60000 password NSITE
 inservice standby standby115
!
interface GigabitEthernet0/1
 description Outside HSRP
 ip address 172.25.1.2 255.255.255.0
 no ip redirects
 standby 115 ip 172.25.1.17
 standby 115 timers 1 3
 standby 115 priority 5
 standby 115 preempt
 standby 115 name standby115
 standby 115 track GigabitEthernet5/38
!
interface GigabitEthernet0/2
 description Inside HSRP
 ip address 172.16.1.2 255.255.255.0
 no ip redirects
 standby 116 ip 172.16.1.17
 standby 116 timers 1 3
 standby 116 priority 5
 standby 116 preempt
 standby 116 name standby116
 standby 116 track GigabitEthernet5/37
!
```

Configuring a DMVPN headend in the SLB design is a little different than configuring a basic DMVPN headend. The loopback interface is configured with the same IP address as the SLB virtual server IP address. This loopback interface is used as the tunnel source address. All headends in the server farm have the same loopback IP address and basic configuration. The only difference is the physical interface IP addresses used to communicate with the SLB router.

7200 Headend

```
interface Loopback1
 description Tunnel source
 ip address 10.40.0.6 255.255.255.255
!
interface Tunnel0
 description dmvpn endpoint
 bandwidth 10000000
 ip address 172.20.1.254 255.255.0.0
 no ip redirects
 ip mtu 1400
 no ip next-hop-self eigrp 10
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 99
 ip nhrp holdtime 300
 ip nhrp server-only
 ip nhrp max-send 1000 every 10
 ip nhrp registration timeout 75
 ip nhrp cache non-authoritative
 no ip split-horizon eigrp 10
 load-interval 30
```



```
 delay 1000
 tunnel source Loopback1
 tunnel mode gre multipoint
 tunnel protection ipsec profile gre_prof
!
interface GigabitEthernet0/1
 ip address 172.16.1.4 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 172.16.1.17
!
```

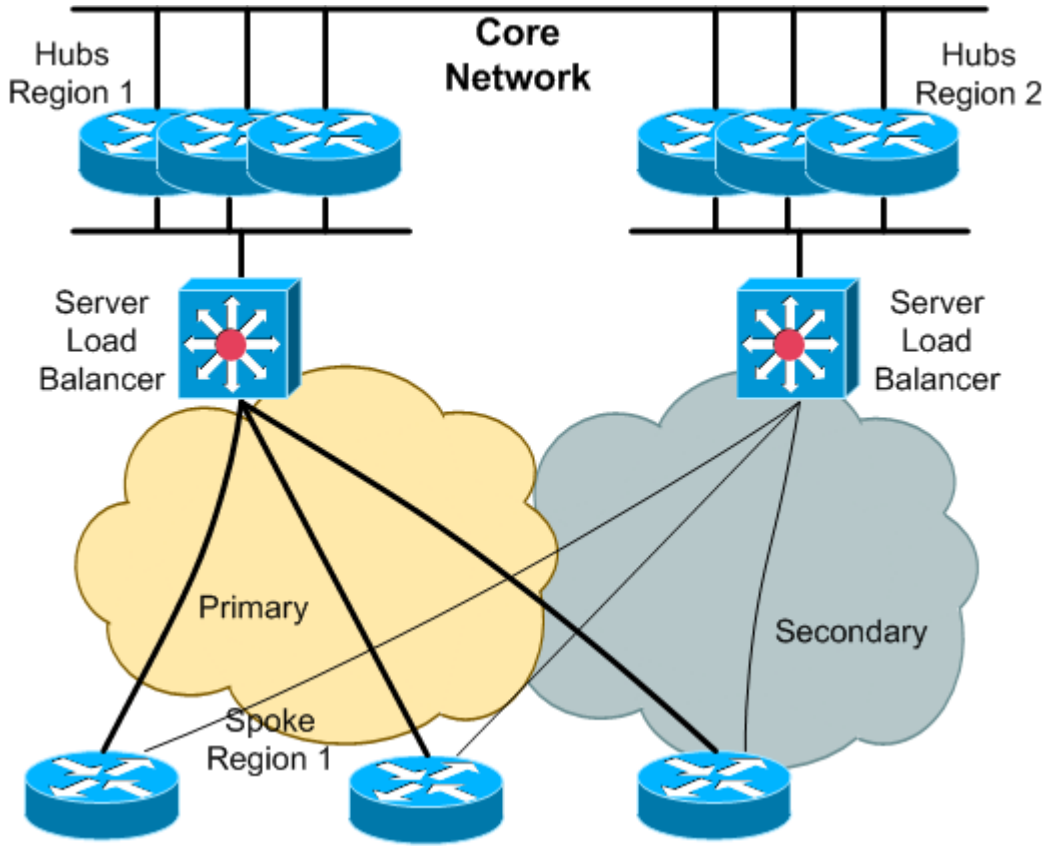
Note: This design requires the two 6500s (or 7200 with 12.2 code) performing the SLB process, and a third switch where the headends connect. This design is easier to configure. However, this design requires a third switch and introduces a single point of failure (the LAN switch between the headends and the SLB devices). Alternative designs without the extra LAN switch using Bridge Group Virtual Interface (BVI) on the headend routers can be found in A.2, "IOS SLB Configurations Using BVI Interfaces on Headend."

4.3.2 Dual Site HA Design

The SLB design provides internal HA among the hubs. When a headend goes down, reconnections are evenly distributed among the other available headend routers. Although the headend server farm is redundant, the SLB router is still a single point of failure. Even if two SLB routers doing stateful IOS SLB are used, both IOS SLB routers can still go down at the same time, for example, if the building housing the IOS SLB routers loses power.

This problem is solved simply by using geographic redundancy. In this HA model, it is recommended to configure the spokes to have secondary next hop server pointing to a secondary DMVPN SLB hub site. If the primary site goes down, the spokes have an alternate location to which they have already built a tunnel, as shown in Figure 4-4.

Figure 4-4. Dual IOS SLB Site Design



This HA model works well if the enterprise already has multiple hub locations. An important consideration when deploying this model is the *n*+1 rule for redundancy of headend routers, which applies to each SLB and server farm location. Therefore, overall 2*n*+2 headend routers would be needed (1/2 at each SLB site). If one hub goes down within the primary SLB, all spokes attached to that hub direct their packets to their secondary hub off of the secondary SLB. After the primary SLB device redirects the spoke tunnels to the remaining hubs at the primary SLB site, and IPsec and NHRP are reestablished, these spokes direct their packets back through the primary SLB site. If the SLB router at the primary site fails, all spokes direct their packets to their hubs at the secondary SLB site.

The configuration of each of the headend sites is basically the same. We just look at one side of the HA design. The spoke configuration is the trickiest, and the headend site with the SLB is similar to the previous single site HA design, although it does not require HSRP.

Spoke Configuration

```
interface Tunnel1
 bandwidth 1000
 ip address 81.1.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.101.2      ← Add Hub1 to multicast list
 ip nhrp map 81.1.0.1 10.70.101.2       ← Hub1 static NHRP mapping
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp nhs 81.1.0.1    ← Indicates Hub1 as next hop server
```



```
 ip nhrp shortcut        ← Indicates to build shortcut tunnels
 ip nhrp redirect        ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub shared    ← Shared keyword required if
                                                     both tunnels use the same
                                                     tunnel source
!
interface Tunnel2
 bandwidth 1000
 ip address 82.1.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.70.102.2      ← Add Hub2 to multicast list
 ip nhrp map 82.1.0.2 10.70.102.2       ← Hub2 Static NHRP mapping
 ip nhrp network-id 102
 ip nhrp holdtime 600
 ip nhrp nhs 82.1.0.1    ← Indicates Hub2 as next hop server
 ip nhrp shortcut        ← Indicates to build the shortcut tunnel
 ip nhrp redirect        ← Indicates to send redirects (not necessary on spoke)
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub shared    ← Shared keyword required if
                                                     both tunnels use the same
                                                     tunnel source
!
```

Primary Hub Configuration

```
interface Loopback1
 ip address 10.70.101.2 255.255.255.0
!
interface Tunnel1
 bandwidth 1000
 ip address 81.1.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic    ← Add spoke multicast list upon registration
 ip nhrp network-id 101           ← Identifies the DMVPN network
 ip nhrp holdtime 600
 ip nhrp shortcut        ← Indicates to build the shortcut tunnel
 ip nhrp redirect        ← Indicates to send NHRP redirects
 tunnel source Loopback1    ← Loopback (SLB VIP) as tunnel source
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
!
interface GigabitEthernet0/1
 ip address 172.16.1.4 255.255.255.0
!
```

Primary SLB Router Configuration

```
ip slb probe PING-PROBE ping
 faildetect 3
!
ip slb serverfarm 7206-FARM
 failaction purge
 predictor leastconns
```

```
 probe PING-PROBE
 !
 real 172.16.1.4                  ← Headend corresponds to a real
  weight 1
  inservice
 !
 real 172.16.1.5
  weight 1
  inservice
!
 real 172.16.1.6
  weight 1
  inservice
!
ip slb vserver ESP
 virtual 10.70.101.2 esp          ← Headend loopback corresponds to this vserver
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 inservice
!
ip slb vserver ISAKMP
 virtual 10.70.101.2 udp isakmp   ← Headend loopback corresponds to this vserver
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 inservice
!
interface GigabitEthernet0/1
 description Internet interface
 ip address 10.70.101.2 255.255.0.0  ← Headend loopback corresponds to intf.
!
interface GigabitEthernet0/2
 description Inside interface
 ip address 172.16.1.2 255.255.255.0
!
```

4.4 Headend Quality of Service

Implementing quality of service (QoS) on the headend is often necessary, because spoke inbound physical bandwidth can become congested. The actual hub router has a much faster physical interface that does not become congested as fast as the spoke Internet connection (that is, the hub can overrun the spoke). Because of this, it is useful to configure the hub with a shaped queuing policy. This is a hierarchical, modular QoS CLI (MQC) based shaper with a child policy for queuing the traffic after shaping is performed.

Note: Because the SLB router uses ping probes to validate the headend as active, it might be prudent to allocate a small amount of bandwidth in the queuing policy so these ICMP "control" packets can pass.

In traffic from the headend toward the branch, there are several ways to classify traffic for queuing and shaping toward a particular spoke.

One way is to use **`qos-group`** under the ISAKMP profile. We can use **`crypto isakmp profile`** to assign a destination branch to a QoS group. Traffic destined to this branch is placed in a QoS group by the crypto identification, and can then be used for classification by QoS group in the QoS policy. If more than one spoke is placed in the same QoS group, traffic for all spokes is measured in the aggregate for the QoS policy.

It is also possible to classify based on the encrypted data IP packet information. You can use an access list, configured to match the private subnet behind the remote spoke. The **qos pre-classify** command is used on the tunnel interface and is required, because the traffic is classified by a parameter that is now encrypted as the traffic leaves the physical outbound interface where the QoS function is performed.

In a final method, an access list is configured to match the source and destination of the IPsec/GRE traffic as it leaves the physical outbound interface. This does not require the use of the **qos pre-classify** command, but is usable only if the remote spokes have static IP addresses, because the spoke tunnel source address must be known so it can be used in the hub QoS configuration.

We will look at the first method of QoS configuration. The traffic to that spoke is shaped to 256 Kbps, and low latency queuing (LLQ) and class-based weighted fair queuing (CBWFQ) is enforced by the child policy.

Hub Configuration

```
class-map match-all class_spoke.1.1
 match qos-group 1
!
policy-map test
class class_spoke.1.1
  shape average 256000
  service-policy child
!
class-map match-all HP1
 match ip precedence 1
class-map match-all HP2
 match ip precedence 2
class-map match-all voice
 match ip precedence 5
policy-map child
 class HP1
  bandwidth 32
 class HP2
  bandwidth 32
 class voice
  priority 64
!
crypto isakmp profile spoke.1.1
   ca trust-point DMVPN-CA
   match identity address 10.21.1.2 255.255.255.255
   match identity user domain cisco.com
   qos-group 1
!
interface GigabitEthernet0/1
 ip address 10.70.100.2 255.255.255.0
 service-policy output test
!
```

Anyone who works with QoS knows that its CPU and memory consumption is very large. The limitations can be closely correlated with processing power and packet storage while delaying their transmission time slot. With DMVPN, it becomes a little more complicated when you add the encryption card, and the interaction of queuing and processing with hardware encryption.

Looking at the Cisco 7200 series router with NPE-G1 and NPE-G2 processors, along with the VAM2+ encryption card, we can compare the processing of the preceding QoS policy. The NPE-G1 with VAM2+ was found to handle a maximum of about 80 active shapers, while the NPE-G2 with VAM2+ handles a maximum of about 150 active shapers. More shapers can be defined; but only these maximum numbers of shapers can actively shape traffic at one time.

Note: It is *not* recommended to use QoS with DMVPN if running IOS 12.4(11)Tx through 12.4(15)T1, due to serious code defects.

4.5 IP Multicast

Multicast on DMVPN is similar in behavior to any NBMA network. These multicast deployments typically have a hub to spoke direction of multicast data flow, where the source of the multicast stream is located somewhere in the core network. Because the DMVPN cloud is a nonbroadcast multiaccess (NBMA) network, Protocol Independent Multicast (PIM) sparse mode must be used with PIM NBMA mode enabled on the hub device. The receivers join at the branch site, and the PIM join is sent over the DMVPN tunnel toward the rendezvous point (RP).

The RP is placed somewhere in the core network, or perhaps at the DMVPN headend. The RP cannot be configured at a spoke. You can use any method of distributing group to RP mapping information. It is suggested to set the shortest-path tree (SPT) threshold to infinity.

Direct spoke to spoke multicast is *not* supported. The multicast source can be at a spoke, but the multicast flows from the spoke, to the hub, and then back to the spokes that want to receive the traffic. In this case, you *must* set the SPT threshold to infinity so the IP multicast packets stay on the shared tree through the hub and do not attempt to switch to the source tree through a spoke-to-spoke tunnel, which will not work. Using PIM NBMA mode enables the hub to forward the multicast only to those spokes that have a receiver behind them, instead of to all spokes on a DMVPN network that a receiver has joined.

Headend Configuration

```
interface Tunnel1
 ip address 10.81.0.1 255.255.0.0
 ...
 ip pim nbma-mode
 ip pim sparse-mode
 ...
!
ip pim rp-address 10.81.0.1
ip pim spt-threshold infinity
!
```

Branch Configuration

```
interface Tunnel1
 ip address 10.81.1.1 255.255.0.0
 ...
 ip pim nbma-mode
 ip pim sparse-mode
 ...
!
ip pim spt-threshold infinity
!
```

Multicast scalability for DMVPN is basically a question of packet replication on the headend router and throughput of the incoming stream. This can be a very CPU intensive process that degrades as the scale increases. In this design, the multicast source should be behind the hub routers so that each multicast hub can replicate the multicast traffic to spokes that request the stream. You should use this with QoS.

It is recommended to enable LLQ on the queuing policy-map. When LLQ is enabled on the outbound interface, a buffer is created between the processor and the crypto card. This enables the crypto engine to

buffer the burst of multicast traffic which otherwise could be tail-dropped on inbound by the crypto card. Multicast traffic does not need to be placed in the LLQ to get the benefit of the buffering. Simply configuring LLQ on the outbound physical interface causes the buffer to be used.

```
class-map match-all BE
 match ip precedence 0
!
class-map match-all HP1
 match ip precedence 4
!
class-map match-all HP2
 match ip precedence 3
!
class-map match-all voice
 match ip precedence 5
!
policy-map child
 class HP1
  bandwidth 32
 class HP2
  bandwidth 32
 class voice
  priority 64
!
interface GigabitEthernet0/1
 ip address 10.70.100.2 255.255.255.0
 service-policy output child
 hold-queue 4096 in
end
```

4.6 Scaling IOS SLB Hub Deployment

Now that we have covered the basic constructs of the 6500 IOS SLB design, we can look at how this DMVPN design scales. It should be apparent that a routing protocol is required to manage the prefixes and reachability in this type of network. The scalability of a DMVPN hub depends upon the scalability of the routing protocol used between the hubs and spokes and among the hubs. Looking at this more closely, it is apparent that scalability also depends upon the number of prefixes the routing protocol must advertise. To reduce overhead, we can advertise summary routes to the spokes and, if nonsplit-tunneling is desired, a default route to each spoke. The spokes are basically stub networks in relation to the DMVPN network as a whole, and are not used as transit networks.

Summarizing branch private subnets at the headends is recommended if branch subnet allocation was designed properly to support summarization. Obviously, not all routing protocols have the same scaling limitations or potential. We look at two protocols that scale fairly well for DMVPN: EIGRP and Routing Information Protocol (RIP). The core network between the hubs can be any protocol; however BGP is the protocol of choice among server farm headends and the rest of the core network. As we looked at earlier in this chapter, BGP route reflection should be used to maintain full routing knowledge between the headends, and can be redistributed into the core routing protocol to propagate that routing information. Using BGP route reflection is discussed in Appendix A.2.2, “7200 SLB Configuration with BGP Route Reflection.”

The bottleneck in all encrypted networks is the processing required to encrypt and decrypt. To maximize throughput and conserve CPU power, we always use hardware encryption modules in DMVPN routers. Crypto processing overhead is distributed across the headend routers in the server farm.

4.6.1 Data Plane Considerations

This section lists and describes data plane considerations, including encryption throughput and unicast and multicast packet processing.

4.6.1.1IPsec Encryption Throughput (Hardware Encryption Engines)

IPsec encryption engine throughput in each platform (headend or branch) must be considered for scalable designs, because every encrypted/decrypted packet that is must traverse the encryption engine. Therefore, encryption throughput must consider bidirectional speeds. In general, encryption throughput is reported as a value that includes both encrypting and decrypting packets. For example, if encryption cards throughput is reported as 100 Mbps, the encryption card can encrypt and decrypt at a combined rate of 100 Mbps (100/0 Mbps (encrypt/decrypt) to 50/50 Mbps to 0/100 Mbps).

In general, as throughput increases, the burden on router CPU also increases. However, with hardware-accelerated encryption available for all Cisco router products from the 871 through the 7200, most encryption processing is offloaded from the main CPU to the VPN hardware. However, main router CPU processing still occurs, so higher throughput typically results in higher CPU consumption. For the 6500, 7600, and 7200 with VSA card, CPU processing per encrypted/decrypted packet is reduced further.

4.6.1.2Unicast Packet Processing

Although bandwidth throughput must be considered, the packet rate for the connection speeds being terminated or aggregated is more important. In general, routers and encryption engines have upper boundaries for processing a given number of packets per second (pps). The size of packets used for testing and throughput evaluations can understate or overstate true performance. It also turns out that encryption engines use about the same amount of resources to encrypt a large packet as to encrypt a small packet. So, the pps that an encryption card can handle tends to stay constant as the size of packets varies. This also means that the encryption throughput in bits per second can vary widely as the packet size changes. For example, if an encryption card could encrypt 10,000 pps, with 60-byte (480-bit) packets throughput would be 4.8M bits per second (bps). With 1500-byte (12,000-bit) packets, throughput would be 120Mbps.

Because of such a wide variance in throughput, pps is generally a better parameter to determine router forwarding potential than bits per second (bps).Headend scalability is the aggregate forwarding potential of all branches that terminate a tunnel to that headend. Therefore, the aggregate pps from all branches affects the headend pps rate.

Throughput varies per platform, and depends largely on the traffic pattern. Recommendations in this chapter are based on a nominal Internet mix (IMIX) traffic pattern, with router CPU utilization averaging 70%.

4.6.1.3Multicast Packet Replication

Multicast traffic affects a router more than unicast processing. This effect is a function of the number of PIM receivers. If multicast is used in the DMVPN network, the design should include fewer receivers per hub than would be used with unicast.

4.6.2 Data Plane Best Practices

You should plan your design for above-average throughput. This prevents surprises when traffic bursts and the CPU does not have enough cycles to process the control plane packets.

It is also recommended to follow these best practices:

- IP MTU – Set the IP maximum transmission unit (MTU) to 1400 on all DMVPN tunnel interfaces to eliminate the potential for fragmentation. GRE and IPsec headers add about 60 bytes to the packet, causing the router to fragment larger packets if this exceeds the interface MTU and straining the CPU. The value 1400 is easy to remember and leaves a little extra room; other protocols also "steal" bytes from the payload, such as Network Address Translation-Traversal (NAT-T) (8 bytes) and Point to Point Protocol over Ethernet (PPPoE) (4 bytes).
- TCP MSS – Set the TCP maximum segment size (MSS) value to 1360 on all DMVPN tunnel interfaces. This value is calculated by subtracting 40 bytes from the IP MTU value. Use the command **`ip tcp adjust-mss 1360`** to set the value on the mGRE tunnel interface toward the spokes. This helps TCP sessions adjust to the lower MTU and is needed if Path MTU Discovery (PMTUD) does not work between end hosts.
- Tunnel Bandwidth – The **`bandwidth`** statement is recommended on the tunnel interface of hub and spoke routers, although the actual value can vary from scenario to scenario. Without the **`bandwidth`** statement, tunnel interfaces are currently allocated very low interface bandwidth (9 Kbps). The default will soon be raised to 100 Kbps. This could affect QoS or other features, such as routing protocols that use the configured bandwidth.

  Note: this is especially important when using EIGRP, which limits its own packets to half the interface bandwidth. If tunnel interface bandwidth on the hub router is not increased, the number of EIGRP neighbors (spokes) that can be supported is artificially limited.
- Hardware Crypto acceleration – Always use a hardware encryption module to do most of the encryption/decryption math. The headend and the spokes should have hardware encryption engines.
- QoS – Provision Quality of Service (QoS) policies as necessary at the headend and branch routers. This helps alleviate interface congestion and ensures that latency sensitive traffic is prioritized over other traffic. If classification is done on the internal packet, you must configure **`qos pre-classify`** on the tunnel interface.

4.6.3 Control Plane Considerations

This section lists and describes control plane considerations, including tunnel aggregation stability, encryption throughput, routing protocols, route summarization, and stub routing.

4.6.3.1Tunnel Aggregation Scalability

You must consider the maximum number of IPsec tunnels that a headend can terminate. Tunnel scalability is a function of the number of branch routers that are terminated to the headend aggregation point. This number must include the primary tunnels and any alternate tunnels that each headend might be responsible for in the event of a failover.

The number of IPsec tunnels that can be aggregated by a platform, and the encryption pps rate, are the primary determining factors for recommending a platform.

Although throughput is highly dependent on platform architecture, as tunnel quantities are increased, overall throughput tends to decrease. When a router receives a packet from a different peer than the peer whose packet was just decrypted, a lookup based on the security parameters index (SPI) of the new packet must be performed. The transform set information and negotiated session key of the new packet is then loaded into the hardware decryption engine for processing. For traffic to be encrypted, the main CPU must match the data packet to the correct IPsec security association (SA) to hand to the encryption engine to encrypt the packet. Having traffic flowing on a larger numbers of SAs tends to negatively affect throughput performance.

Increasingly, platforms with hardware-accelerated IPsec encryption are also designed to offload IPsec SA processing overhead, resulting in more linear performance regardless of the number of SAs. For example, the VPN SPA blade for the Cisco 7600 has fairly linear throughput whether the traffic load is offered on a few SAs or several thousand. Currently, the VPN SPA (6500/7600) and the VSA (7200 NPE-G2) are the only encryption processors that have this functionality.

The GRE configuration affects router encryption throughput. GRE headers are added to the beginning of each packet, and GRE headers also must be encrypted. The GRE encapsulation process, when not hardware-accelerated, increases total CPU utilization. In most cases, the amount of CPU overhead due to GRE encapsulation is quite small compared to the CPU overhead due to IPsec encryption.

4.6.3.2SLB

In this design, SLB is an integral part of the DMVPN hub site. Here are a few recommendations for the SLB configuration:

- Max Connections – Use max-connections configuration to limit the number of spokes connecting to the real server. This value is dictated by the scalability of the routing protocol running on the real server.
- Idle timer – Set the idle timer to be higher than both the DPD and SLB sticky timers. This keeps SLB from prematurely clearing connection information for the tunnel to the real server. If SLB were to clear this entry, the tunnel could switch to another real server, thus necessitating the rebuilding the crypto session, causing the tunnel to flap. (See 4.6.3.6, “Control Plane Best Practices.”)
- Sticky timer – Stickiness refers to the functionality of remembering which “real” hub device the connections are being sent to. The sticky timer should be shorter than the idle timer.
- Buddy group – IKE and IPsec from a spoke must be tied together using group option in the sticky timer command. By setting both vservers to have the same sticky timer, and group number, this makes ISAKMP (UDP 500 or 4500) and the IPsec (ESP – IP protocol 50) packets use the same real server.
- Purge failed connections – Configure SLB to automatically remove connections to failed real servers and firewalls from the connection database even if the idle timers have not expired. This function is useful for applications that do not rotate the source port (such as IKE), and for protocols that do not have ports to differentiate flows (such as ESP).

4.6.3.3Routing Protocols

Running a routing protocol affects CPU overhead. Processing hello packets and maintaining a routing table uses CPU time. The amount varies with the number of routing peers and routing table size. The network manager should design the routing protocol based on generally accepted practices for the particular routing protocol.

Routing protocol scalability is the most important limiting factor in determining the number of branches a given hub can support. Scalability becomes more cumbersome as the number of routes that the protocol must advertise to each branch increases. We will look at EIGRP and RIPv2, along with various options that can be used to increase scalability.

4.6.3.4Route Summarization

Hub routers forward data traffic based on the routing table. Based on the routing table, summarization can occur in the core of the network, and in the spoke networks. This enables us to build a network with a hierarchical design having smaller, more efficient routing tables. In general, in a hierarchical network, at each level the routing protocol should summarize routes going up the tree toward the spokes (leaves), but send full routing information to nodes in the same level (region) or down the tree (toward the root).

Most dynamic routing protocols support summarization. However, you might want to run a separate routing protocol, such as BGP, between hub routers within the same region or within the core. Regional hub routers must have full routing knowledge of their regions. The regional hubs can summarize their routing tables when advertising them to the spokes and other hubs.

Toward the branch routers, the protocols should be configured to advertise very few routes. This should, in most cases, be a default route, or, in few cases, a summary route for the rest of the network. If a branch router uses split tunneling, a default route should not be advertised to the branch unless the branch uses VPN routing/forwarding instances (VRFs), in order to segregate split-tunneled traffic.

4.6.3.5Stub Routing

Branch routers are rarely used as transit networks. Therefore, it is best to configure them as stub networks. Using EIGRP, you can set the routing protocol to behave as a stub; it announces this functionality in hello packets to the headend router. Using the **`stub connected`** command, the branch router sends connected prefixes to the headend. The headend has reachability of the branch networks, and does not have to include the branch router in the query process.

Note: Be careful to not advertise the tunnel source interface subnet to the hub router through the tunnel interface. If the hub uses this route the tunnel packets to this spoke will go into a forwarding loop that will either hang or crash the hub router. To protect against this possibility, either block this network from being advertised by the spoke or being accepted by the hub.

Spoke Configuration

```
router eigrp 10
 network 10.80.0.0 0.3.255.255          ← DMVPN Tunnel subnet
 network 10.211.9.0                     ← Private VPN subnet
 no auto-summary
 eigrp stub connected    ← Announces stub router, advertise connected intf.
!
```

In RIPv2, there is no stub keyword. This protocol must operate in the native manner. Alternatively, you can make the mGRE tunnel on the hub passive and use reliable static routing on the branches. In this case, the hub still receives RIP advertisements from the spokes, but does not send the spokes RIP advertisements. Reliable static routing on the branch router uses the IP service level agreement (SLA) feature to poll an address on the primary headend using a ping probe, coupled with a tracking object tied to a static route.

As long as the ping probe operates, it installs a static route directing traffic toward the primary headend over the tunnel. If the probe is nonoperational, the static route is removed and a floating static route having the same network and mask, but higher administrative distance (AD), directs traffic to the secondary headend.

4.6.3.6Control Plane Best Practices

The following options should be used to increase the stability of the design:

- Protocol Timers – Set the routing protocol dead timer to a larger value. This provides more time for the WAN to recover in the event of a minor problem. Otherwise, a minor flap could cause Interior Gateway Protocol (IGP) running over the tunnels to go down and require reconvergence. Also, consider leaving the hello timer the same, rather than increasing it. This helps keep the routing protocol from flapping if the WAN network is somewhat lossy.
- Interface Queues – Set the interface input and output hold queues to 4096 on the tunnel interface and output WAN interface of the headend, and possibly other routers on the path from the headend to ISP router, to minimize protocol failure.
- PKI and Digital Certificates – Use digital certificates and Public Key Infrastructure (PKI) for the ISAKMP authentication. This increases manageability as the site count increases. Preshared keys (PSK) are not a scalable solution for distributing Internet Key Exchange (IKE) authentication credentials. PKI is highly scalable and secure. Be sure to turn off certificate revocation list (CRL) checking because it periodically drains CPU power.
- Dead Peer Detection – Enable dead peer detection (DPD) on Crypto SLB device and spokes. This increases the speed at which a failure over the IPsec tunnel or connectivity loss is detected. This should be set higher than the routing protocol update timer, but less than the dead timer and SLB sticky timer.

  Set the **`crypto isakmp keepalive`** timer value to be greater than the RP hello timer by 5 seconds. Set the total ISAKMP keepalive + 5* retry timer for a timeout equal to RP dead timer + 60 seconds. This prevents ISAKMP and IPsec sessions from tearing down before losing routing protocol adjacency.

```
crypto isakmp keepalive <hello+5 (sec)> <(dead+60-(hello+5))/5)(sec)>
```

  Example: Hello timer = 10 secs and dead timer = 60 seconds; then keepalive = 15 (10+5) and retry = 21 (60+60-(10+5))/5

```
crypto isakmp keepalive 15 21
```

- Call Admission Control on hub – In failure scenarios, IKE processing can be debilitating for a router. It is recommended to enable Call Admission Control (CAC) to limit the number of IPsec sessions that can come up simultaneously. This prevents incoming IPsec connections from overrunning the CPU. There are two CAC methods

  Configure the absolute IKE SA limit so the router drops new IKE SA requests after the number of IKE SAs in negotiation reaches the configured limit. After the number of IKA SAs in negotiation drops below the configured limit, additional IKE SAs are processed.

```
crypto call admission limit ike in-negotiation-sa [number]
```

  Configure the system resource limit so the router drops new IKE SA requests when the specified percentage of system resources is in use. It is recommended to set this at 90% or less.

```
crypto call admission load [percent]
```


4.6.4 Scale Limits

This section examines specific DMVPN scaling limits of the routing protocols and hardware crypto engines. Remember, throughput is not a focus, but a set parameter to load the system to a high operating load. The following sections illustrate the DMVPN peer scalability for the hierarchical hub design using a Cisco 7200 router serving as the headend. Scalability is the same for a given type of router. It is recommended to make the server farm routers uniform with the same crypto cards and processors.

It is very important to understand how scaling numbers are found and what to expect. All internal testing to derive the numbers was done in a lab, so it is likely the numbers stated are optimal for the design. The scalability is determined by both the number of DMVPN peers the design can support, and the data throughput at the maximum scale. In the following data, notice the number of routing protocol peers does not necessarily increase with a better processor or encryption card. Rather the scale of the design is consistant, and the amount of throughput increases as more powerful hardware is used.

There have been great improvements in the scalability of the routing protocols for DMVPN. Depending on the IOS image your router is running, scalability varies. It is recommended to run either IOS 12.4(9)Tx or 12.4(15)T2 and above with the Advanced Enterprise Security (adventerprisek9) feature set. IOS 12.4(11)Tx *should not* be run with DMVPN and QoS.

Note: These results are based on IOS 12.4(15)T4 with the Advanced Enterprise Security (adventerprisek9) feature set.

4.6.5 EIGRP

Cisco recommends using EIGRP as the routing protocol for your DMVPN network. EIGRP offers fast convergence and robustness. It is recommended to configure the spokes to be EIGRP stub peers. This minimizes convergence times and simplifies the convergence process. Cisco also recommends using the summarization potential offered with DMVPN Phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes.

Looking at the 7200 Series router as the headend, you can use many hardware variations. For the route processor, you should use either NPE-G1 or the NPE-G2. If you use an NPE-G1, you must use the VAM2+ for the encryption port adapter. When using the NPE-G2, you can use a VAM2+ or a VSA.

The following table shows the recommended limit of EIGRP peers per headend with each hardware combination.

Table 4-1. Recommended and Maximum EIGRP Peers

EIGRP Peers	Recommended	Maximum*
7200 NPE-G1 w/VAM2+	600	900
7200 NPE-G2 w/VAM2+	600	900
7200 NPE-G2 w/VSA	700	1000

* The Maximum column indicates the maximum peer limit in an ideal environment, such as a lab. It *must* be understood that the maximum results are probably unobtainable in a real network, where more delay and other variables that adversely affect routing protocol convergence. Also, although the NPE-G2 w/VSA card does not increase the number of supported EIGRP peers by very much, the card greatly increases aggregate encryption pps and bps rates.

Note: These numbers assume that summarization to the spokes is used and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is around 70% load on the router. If your network differs from these parameters, your results may vary from the reported numbers, which are provided as planning guidelines.

During times of convergence EIGRP will generate much more traffic then during periods of network quiescence. The following table lists the approximate peak amount of traffic in bits per second that EIGRP will generate outbound over the WAN for a given number of spokes and prefixes advertised towards the spokes. This includes the encryption overhead of the DMVPN tunnel. These numbers can be used to help determine the size of the WAN link to the Internet. Actual data traffic will of course add to these numbers. EIGRP will in all likelihood converge even if the WAN link is smaller than the given value but there will probably be some packet loss due to outbound packet drops.

As can be seen in the table, the number of prefixes advertised does not have a great impact on the traffic rate. The fact that the single prefix numbers are greater in some cases than the fifty prefix numbers is probably due to sampling errors. The inbound traffic rates are generally much lower.

Table 4-2. Minimum WAN banwidth needed by EIGRP for a given number of peers and prefixes advertised

Number of EIGRP Peers	Max. BPS for 1 Prefix Advertisement	Max. BPS for 10 Prefix Advertisement	Max. BPS for 50 Prefix Advertisement
600	2887000	2963000	3669000
1200	10075000	12886000	15752000
1800	20577000	25939000	27732000
2400	28717000	29845000	34397000
3000	36329000	35997000	36672000
4000	55736000	55056000	55004000
5000	69670000	68820000	68755000
6000	83604000	82584000	82506000

4.6.6 RIPv2

Cisco supports RIPv2 as the routing protocol over a DMVPN network. RIP is a simple, effective protocol in a DMVPN network. RIP can scale very high compared to other routing protocols over DMVPN. It is recommended to use the summarization potential offered with DMVPN Phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes. Using the passive interface with reliable static routing (basically, stub branches in RIPv2) did not provide any benefit compared to native RIP. The reason for this is simply the input queue limits the number of updates creating a minor bottleneck. So regardless of active or passive RIP, the updates will still be sent from the spokes every 30 seconds. In production networks, the updates sourced from the spokes should not have the same synchronization as we see in a lab environment.

Looking at the 7200 series router as the headend, you can use many hardware variations. For the route processor, you should use either the NPE-G1 or the NPE-G2. If you use an NPE-G1, you must use the VAM2+ for the encryption port adapter. When using the NPE-G2, you can use either a VAM2+ or a VSA.

The following table shows the Cisco recommended limit of RIP peers per headend with each hardware combination.

Table 4-3. Recommended and Maximum RIPv2 Peers

RIP Peers	Recommended	Maximum*

7200 NPE-G1 w/VAM2+	1000	1125
7200 NPE-G2 w/VAM2+	1000	1125
7200 NPE-G2 w/VSA	1000	1125

* The Maximum column indicates the maximum peer limit in an ideal environment, such as a lab. It *must* be understood that the maximum results are probably unobtainable in a real network, where more delay and other variables that adversely affect routing protocol convergence.

Note: These numbers assume that summarization to the spokes is used and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is around 70% load on the router. If your network differs from these parameters, your results may vary from the reported numbers, which are provided as planning guidelines.

4.6.7 Unicast Throughput

In this design, the headends handle all crypto processing, and the VAM2+ or VSA handles the crypto functions for a given headend. Because the combination of the encryption card and router CPU, limits throughput, it is important to know the throughput limits of the encryption cards.

The throughput values in the following table show the aggregate throughput of each hardware combination on one 7200 headend. The table shows maximum throughput of bidirectional data traffic for each 7200 processor at 70% CPU utilization, forwarding an IMIX pattern of traffic (7×64:4×570:1×1400).

Table 4-4. Bidirectional Unicast Throughput

Bidirectional Throughput	PPS	Approx. Mbps
7200 NPE-G1 w/VAM2+	24,000	64
7200 NPE-G2 w/VAM2+	42,000	116
7200 NPE-G2 w/VSA	60,000	320

These values show the approximate maximum aggregate throughput for the given processor and crypto card conbination. The VSA or VAM2+ is the throughput bottleneck in this design.

Note: The bandwidth numbers *do not* represent real traffic, but are found by simulating a realistic traffic pattern in the lab. Real traffic does not always follow a given pattern, so throughput values in a real network differ from those found in lab testing. Also, these numbers assume throughput at a maximum CPU load of 70% on the router. If your network parameters differ, your results might vary from the reported numbers, which are provided as planning guidelines.

4.6.8 Multicast Throughput

In a DMVPN network, the headends handle all multicast replication processing, and the VAM2+ or VSA handles all crypto functions after replication. Throughput of the input stream is magnified by the number of peers receiving the stream.

The code and configuration were identical on the NPE-G1 and the NPE-G2. It surprised us that the NPE-G1 with VAM2+ performed much better in multicast forwarding than the other platforms. The VAM2+ dropped packets when paired with the NPE-G2 because of **`ppq full`** errors. The theory is that the NPE-G2 replicates the multicast stream so fast that it overruns the crypto card buffers.

Table 4-5. Multicast Throughput

Processor and Crypto Card	Maximum number of Multicast Replication without Packet loss	Average Interrupt CPU Utilization on UUT	Total UUT Forwarding PPS	Total Hub router Forwarding BPS Ignoring Crypto and GRE Overhead
NPE-G2 with VSA	220	11%	8,140	8,335K bps
NPE-G2 with VAM2+	100	7%	3,700	3,788K bps
NPE-G2 with VAM2+ and LLQ on Outbound Interface *	800	93%	29,600	30,310K bps
NPE-G1 with VAM2+	660	90%	24,420	25,006K bps

* To deal with limited VAM2+ buffer space, configure QOS with LLQ on the physical outbound interface. The multicast traffic does not need to be placed into the low latency queue, nor does it require a bandwidth statement to reserve bandwidth for the multicast traffic. Simply enabling LLQ on the outbound interface causes IOS to buffer the data going to the crypto card so that it can perform QOS before crypto. The QOS before crypto feature engages when the crypto card is congested, not when the physical outbound interface is congested. This strategy is *not* expected to work for the NPE-G2/VSA combination.

5 DMVPN 6500 Crypto Server Load Balancing Design

The power of this server load balancing (SLB) design is the ability to multiply the scalability of a headend router by creating a server farm of routers. 6500 crypto SLB handles all encryption/decryption, and distributes the GRE tunnels evenly across the headend routers. The headends do not require the crypto hardware because the 6500 VPN-shared port adapter (SPA) handles all of the crypto functions. This design provides incremental scalability with the addition of headend routers as needed. Furthermore, there is built-in redundancy in the design: tunnel load balancing adjusts if a headend router fails.

This design also works well with common routing protocol designs. The topology is basically hub and spoke, so the hubs all independently run the same routing protocol. Therefore, a well planned topology and addressing scheme can provide a well summarized set of routes. This keeps the routing tables on all devices as compact as possible, and network growth can be considered.

This chapter describes some common deployment scenarios for this design.

5.1 Deployment Scope

This deployment is useful for a large number of connections and high encrypted throughput. The scalability is easily extended by adding an extra spoke to the server farm. This provides high availability along with incremental scalability. It also offers basic backup in the event of a hub device failure.

This design is another single layer design, featuring the highest spoke scalability and throughput. Its design is basically the same as the IOS SLB design; however the crypto functionality is offloaded to the 6500 equipped with an IPsec VPN-SPA encryption engine. The VPN-SPA takes the encryption/decryption load off the headends. Therefore, the headends do not need a crypto card and have less work to do. The headends still maintain the NHRP, mGRE and routing protocol components of the DMVPN tunnel.

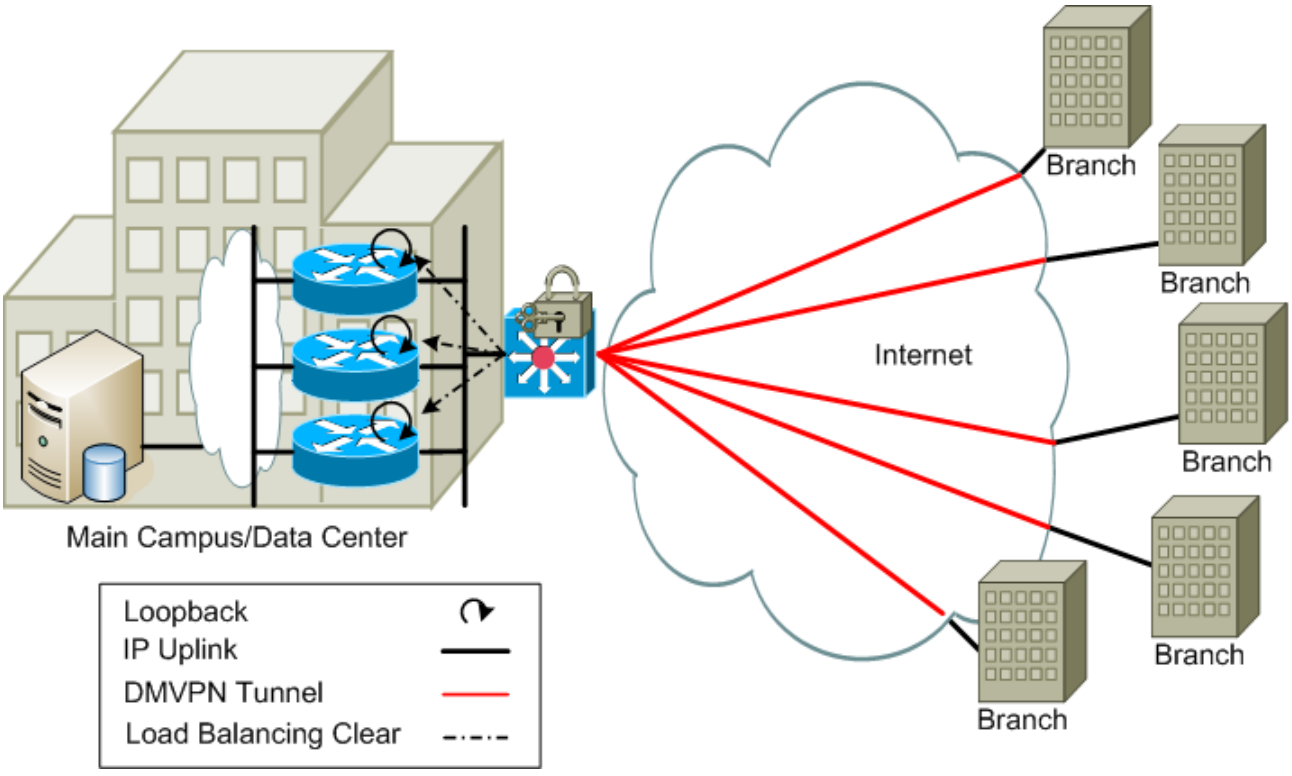
At the hub site, several headend routers terminate the GRE connections. The 6500 VPN-SPA handles crypto functionality and the SUP720 does the server load balance function, distributing GRE tunnel connections to the server farm of headend routers. This design has built in redundancy to ensure that the hub can always handle connections. If a headend device fails, the connections will be reformed and distributed across the remaining headend devices. Because the IPsec and NHRP functions are performed on different devices (6500 and headend routers), these functions can no longer communicate with each other. IPsec on the 6500 is configured use a dynamic crypto-map and can only accept inbound initiated IPsec sessions from the spokes. NHRP cannot direct the 6500 to initiate outbound IPsec sessions. Therefore, this configuration is only suited for DMVPN hub site; in this design the DMVPN spokes always initiate the IPsec sessions. Note, if hub-to-hub tunnels are required between 6500s in different areas, static crypto-maps can be used to encrypt these tunnels.

This design is typically used in simple deployments with higher throughput. The power of this design is the use of the 6500 IPsec VPN-SPA to do all the crypto operations. High throughput deployments, such as a remote data center, or regional large-scale enterprise class teleworker (ECT) deployments, can benefit from this design. In this design, many branches can securely connect to the headend with high data throughput.

A potential deployment is when an enterprise has remote offices, as shown in Figure 5-1. The data center would be located on the main campus (or campuses), and a remote office can securely link into one or many

of them. These offices can exist in any geographic region, and can connect to the nearest headend site. This is more common in large enterprises, where the internal campus network handles all routing throughout the network. Secure access to the data center enables the regional office to perform data backup and transfers securely without sacrificing performance.

Figure 5-1. DMVPN 6500 Crypto Server Load Balance Deployment



5.2 Basic Operation

The Cisco 6500 Series router is configured with IPsec and SLB. The 6500 is configured to specifically load balance GRE traffic to the headend routers. The 6500 is has a VPN-SPA installed, and is configured with dynamic crypto maps to accept inbound initiated IPsec sessions that contain GRE packets destined to the SLB virtual IP address (VIP). When using tunnel protection on spokes, these proxies are automatically set to match the GRE traffic.

The crypto endpoint public IP address of the 6500 is the same as the SLB VIP. This address must also be configured on a loopback interface of every DMVPN hub router in the server farm. Loopbacks on the headends are used as the source IP address for the return GRE traffic. On the 6500, after decryption the GRE packet is in cleartext and is load balanced to a headend router in the server farm. The headends terminating the GRE tunnel, process the GRE packet, and the inner IP data packet emerging from the tunnel. The headend then performs any NHRP processing, routing protocol processing, or data packet forwarding.

Note: The 6500 with a VPN-SPA could be a DMVPN hub with high throughput for a limited number of spokes (500-1000), but the 6500 does not yet support DMVPN Phase 3.

Because of the complexity of 6500 in this design, we look at parts of the configuration in a logical order, rather than the order it normally appears in the running configuration. This helps show how the configuration operates to provide the functionality we are describing.

First, we look at the 6500 crypto configurations. These are basically the same as those presented in Chapter 2, "Scalable DMVPN Design Considerations." However, we must use dynamic crypto maps on the 6500. Here we have the PKI trustpoint, along with the ISAKMP profile. The dynamic crypto-map is linked to a normal crypto-map with the local interface set to a VLAN interface. Remember, the 6500 is a switch, and uses VLANs for almost everything.

```
ip domain-name cisco.com
!
crypto pki trustpoint DMVPN-CA
 enrollment url http://10.24.1.130:80
 serial-number none
 ip-address 10.72.2.2
 password
 subject-name CN=6500-CSLB-2, OU=NSITE
 revocation-check none
 rsakeypair VPN1
 auto-enroll
!
crypto pki certificate chain DMVPN-CA
 certificate 3607
 <certs omitted>
 certificate ca 01
!
crypto isakmp policy 1
 encr 3des
 group 2
!
crypto isakmp keepalive 30 5
crypto isakmp profile hub
   ca trust-point DMVPN-CA
    match identity user domain cisco.com
    match identity address 0.0.0.0
!
crypto ipsec transform-set gre esp-3des esp-sha-hmac
 mode transport
!
crypto dynamic-map vpn1-dyna 10
 set transform-set gre
 set isakmp-profile hub
!
crypto map vpn1 local-address Vlan5
crypto map vpn1 10 ipsec-isakmp dynamic vpn1-dyna
!
```

Now that the crypto is configured, it must be linked to the IPsec VPN-SPA. In the 6500, encrypted traffic arrives on GigabitEthernet interface 5/1, which is associated with VLAN 5. This is the WAN interface where encrypted traffic arrives. The **`crypto connect vlan 5`** is used to direct encrypted traffic to the VPN-SPA.

After traffic is decrypted, where does the traffic go? The traffic is placed back in the VLAN, and the Multilayer Switch Feature Card (MSFC) routes traffic as needed. In our case, the traffic is put through the SLB configurations, which we look at shortly.

While on the topic, let us look at the reverse path of DMVPN GRE packets that flow from a headend toward a spoke. Obviously these packets must be encrypted. In the following configuration, notice the crypto-map is applied to the VLAN 5 interface, along with the designation of the crypto engine to use. This sends clear traffic through the VPN-SPA, where it is encrypted and then sent out on the GigabitEthernet5/1 interface.

```
vlan 5
 name CRYPTO_VLAN
!
interface GigabitEthernet5/1
 description to GSR-VPN-CORE-1:g1/0
 no ip address
 crypto connect vlan 5
 spanning-tree portfast
!
interface Vlan5
 description ****Crypto Vlan****
 ip address 10.72.2.2 255.255.255.0
 crypto map vpn1
 crypto engine slot 1/0
!
```

Next, you can see that the VPN-SPA appears in the configuration as two Gigabit Ethernet interfaces, and that VLAN 5 is listed as allowed. This is *not* configured by the user; but is automatically configured after you set up the preceding configurations.

```
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 5
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,5,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
```

Assuming that the VPN-SPA decrypted the encrypted traffic, it comes out as cleartext GRE packets destined for the vserver, which happens to be the same as interface VLAN 10. You might think that the GRE packets will reach this interface and be dropped, but the SLB code handles the GRE packets first. Notice that the vserver is the same address, which is also the address configured on the loopback interface on each headend router.

The following configuration defines each real headend router in the server farm. We also enable an ICMP probe to periodically test each real router. If three pings fail, the real router is declared dead, and the fail action purges the connections. The vserver has an IP address representing the entire server farm. The virtual address with the **gre** keyword pushes the decrypted GRE packets to a given headend. Obviously, in an established session, the GRE packet is always sent to the headend that is servicing that connection. Otherwise, the GRE packet is sent to the least loaded headend, according to the SLB connection table.


```
ip slb probe PING-PROBE ping
 faildetect 3
!
ip slb serverfarm 7204-FARM
 predictor leastconns
 failaction purge
 probe PING-PROBE
 !
 real 10.72.100.2
  weight 1
  inservice
 !
 real 10.72.100.3
  weight 1
  inservice
!
ip slb vserver GRE
 virtual 10.72.2.2 255.255.255.254 gre
 serverfarm 7204-FARM
 no advertise
 idle 30
 inservice
!
```

Notice the address on the VLAN 5 interface is the same as that of the SLB vserver, which we look at shortly. Otherwise, the only things to see in the following configuration example are the access interfaces and vlan (Vlan 10) that our headend routers connect to.

```
vlan 10
 name VLAN_TO_HUBS
!
interface Vlan10
 description ****Vlan Connected to Hubs*****
 ip address 10.72.100.1 255.255.255.0
!
interface GigabitEthernet2/1
 description ****To 7200-CSLB-1 Gig0/1****
 switchport
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface GigabitEthernet2/2
 description ****To 7200-CSLB-2 Gig0/1****
 switchport
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
```

Now we look at the routers in the server farm. Each headend is configured with a loopback interface that has the same IP address as the SLB vserver address. The mGRE tunnel is sourced from this interface. The IP address on the tunnel interface of all headend routers is also the same. All headend routers must appear identical to the spokes, because SLB can "map" a spoke to any headend router.

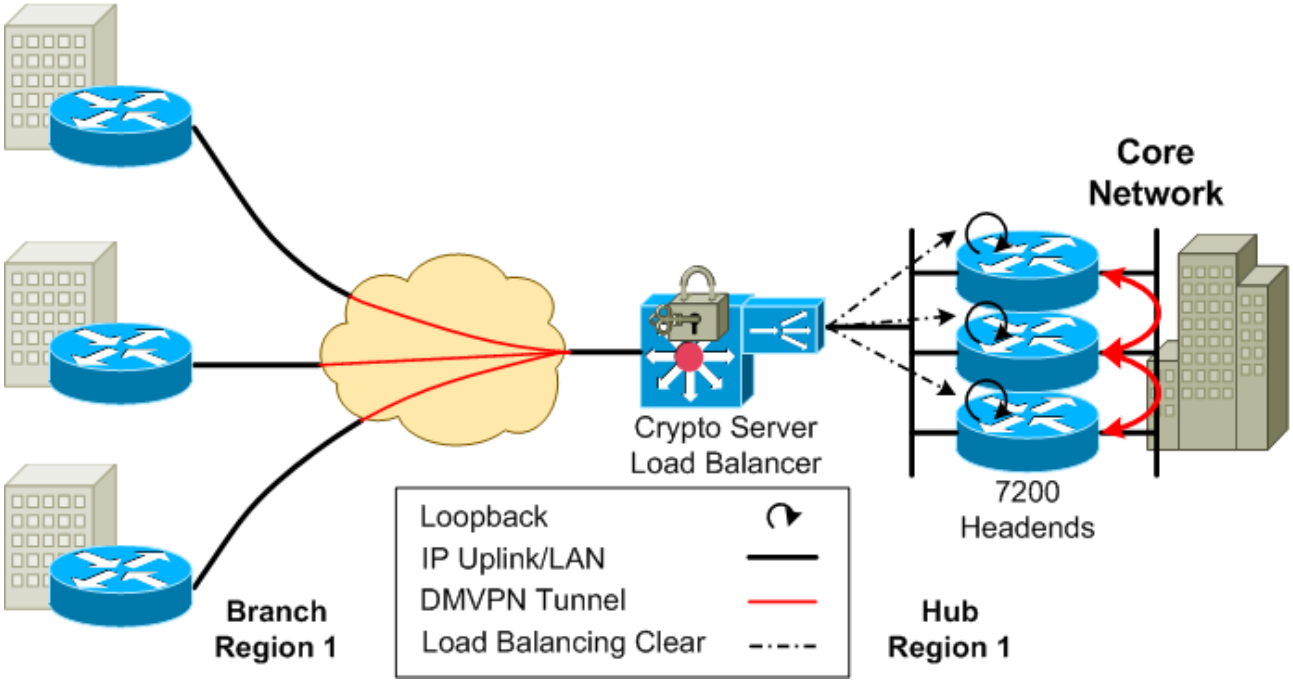
The server farm routers must share full routing knowledge because the spoke connections are randomly distributed to these routers. Subnet information from the set of branches that a headend router terminates is not obvious to the other headends, so they must all share spoke routing information.

Headend configuration:


```
interface Loopback1
 ip address 10.72.2.2 255.255.255.255
!
interface Tunnel1
 bandwidth 10000000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 no ip next-hop-self eigrp 10
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp redirect
 ip tcp adjust-mss 1360
 no ip split-horizon eigrp 10
 load-interval 30
 tunnel source Loopback1
 tunnel mode gre multipoint
 hold-queue 4096 in
 hold-queue 4096 out
!
interface GigabitEthernet0/2
 description CSLB-facing interface
 ip address 10.72.100.2 255.255.255.0
!
router eigrp 10
 network 10.80.0.0 0.3.255.255
!
ip route 0.0.0.0 0.0.0.0 10.72.100.1
!
```

Notice that most of the configuration is identical between the headend routers. The only thing that is unique is the IP address on the physical interface facing the SLB device. This makes it easy to add a new headend device to increase the scaling of the hub site.

Figure 5-2. DMVPN 6500 Crypto Server Load Balance System Architecture



5.2.1 Using BGP to Provide Full Routing Knowledge

In the SLB design, the routers in a server farm must share full routing knowledge. This is important because spoke connections are randomly distributed to these routers. Subnet information from a given set of branches that a headend router terminates is not obvious to the other headends, so they must all share specific routing information for the spoke networks.

Headend routers in the server farm communicate with each other using an mGRE tunnel. It cannot be the same tunnel interface the spokes terminate on, but it must have the same NHRP network-ID. This allows NHRP to trigger a redirect for the creation of spoke-to-spoke tunnels when the spokes are being serviced by different hub routers.

For scalability, hubs peer with each other over Border Gateway Protocol (BGP), using the headend as route-reflector. Routes learned from a spoke that connects to a hub become present on other hubs. Two basic designs achieve this: folded and unfolded. Regardless of the method used to reflect the routes, redistribution between BGP and the core routing protocol must be performed to propagate the routes to the other headends and the rest of the core network.

The folded design uses the device or devices providing SLB functionality. The SLB device is a good choice since it is already connected to all of the headend routers. Configure an mGRE tunnel interface and peer with the headend routers through the same physical interface that load balances the hub-to-spoke tunnels. The SLB device does double duty and extra routers are not required to handle the route reflection in this case.

The unfolded design requires extra routers between the headends and the rest of the internal network. These routers only perform route reflection over the mGRE tunnels to the headends and connect the DMVPN network (clear text side) to the internal network. This is straightforward to do, and using an extra router that is used only as the BGP route reflector. This method is not covered in detail.

For redundancy, it is recommended to have at least two route reflectors in either design. Even if BGP is not used to provide the full reachability among the headends, the routing protocol between the SLB device (folded) or this other router (unfolded) and the core network, should use the same routing protocol that is used in the core network.

5.2.1.1Folded BGP Route Reflector Design

In the folded design, the 6500 or 7200 handling the SLB function can also perform BGP route reflection. The tunnel on the SLB device is sourced from VLAN 10, which connects to the spoke physical interfaces. The BGP weight is set to the maximum for headend clients so that the headend installs and propagates routes learned from its clients, instead of a possible BGP peering with a remote hub. Each device on the tunnel has a static NHRP entry for the other devices on the tunnel. This is more reliable than using dynamic updates. All involved IP addresses are static.

6500 Crypto SLB device Configuration

```
interface Tunnel2
 description ****To Route Reflector Clients****
 ip address 10.91.0.1 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp map multicast dynamic
 ip nhrp map 10.91.0.2 10.74.100.2
 ip nhrp map 10.91.0.3 10.74.100.3
 ip nhrp network-id 102
 ip nhrp cache non-authoritative
 tunnel source Vlan10
 tunnel mode gre multipoint
!
router bgp 10
 no synchronization
 bgp router-id 10.70.201.2
 bgp log-neighbor-changes
 neighbor 10.91.0.2 remote-as 10
 neighbor 10.91.0.2 route-reflector-client
 neighbor 10.91.0.2 weight 65535
 neighbor 10.91.0.3 remote-as 10
 neighbor 10.91.0.3 route-reflector-client
 neighbor 10.91.0.3 weight 65535
 no auto-summary
!
```

Note: The BGP router ID on the individual headends must be set because its loopback address is the same as all other hubs connected to the same headend. Without this configuration command, there is a strong possibility that the hubs will have duplicate BGP router IDs.

The hub advertises its routes to the route-reflector with the next hop self set. Without this command, the next hop would be the tunnel IP address of the spoke from which the route was learned. This would prevent other hubs from reaching these destinations. Routes from spokes are learned through Enhanced Interior Gateway Routing Protocol (EIGRP) and redistributed to BGP.

Hub Configuration

```
interface Tunnel2
 description ****Tunnel to BGP Peering with RR****
 ip address 10.91.0.3 255.255.255.0
 no ip redirects
```



```
 ip mtu 1440
 ip nhrp map 10.91.0.2 10.74.100.2
 ip nhrp map 10.91.0.1 10.74.100.1
 ip nhrp network-id 102
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
!
router bgp 10
 bgp router-id 10.91.0.3
 bgp log-neighbor-changes
 redistribute eigrp 10
 neighbor 10.91.0.1 remote-as 10
 neighbor 10.91.0.1 next-hop-self
 no auto-summary
!
```

5.2.1.2Unfolded BGP route reflector design

In the unfolded design, separate routers act as route-reflector instead of the SLB router. Configuration is nearly identical. Peering with the route-reflector-clients must occur over an mGRE interface having the same NHRP network-ID as the spoke-facing tunnels. This supports direct spoke-to-spoke tunnels for spokes that terminate on different hubs. The main difference is the requirement for an extra, separate router.

5.3 HA Design

HA design provides network resilience and availability in the event of failures. To provide resiliency in the DMVPN SLB design, Cisco recommends using an *n*+1 server farm of headend routers. You should have enough routers to handle the scale of your intended spokes (*n*), and an additional headend router (+1) in the event of a failure. Obviously, you can use *n*+2 or more for even more HA, but *n*+1 is the recommended minimum.

The value of *n* in the formula is derived by dividing the number of anticipated DMVPN peers by the scalability of one of the headend devices you are going to use. Before designing your DMVPN hub site, you need to know the type of headend routers you will use and the DMVPN scalability of the router. Remember that you can always add headends to the server farm later, so future expansion is easy.

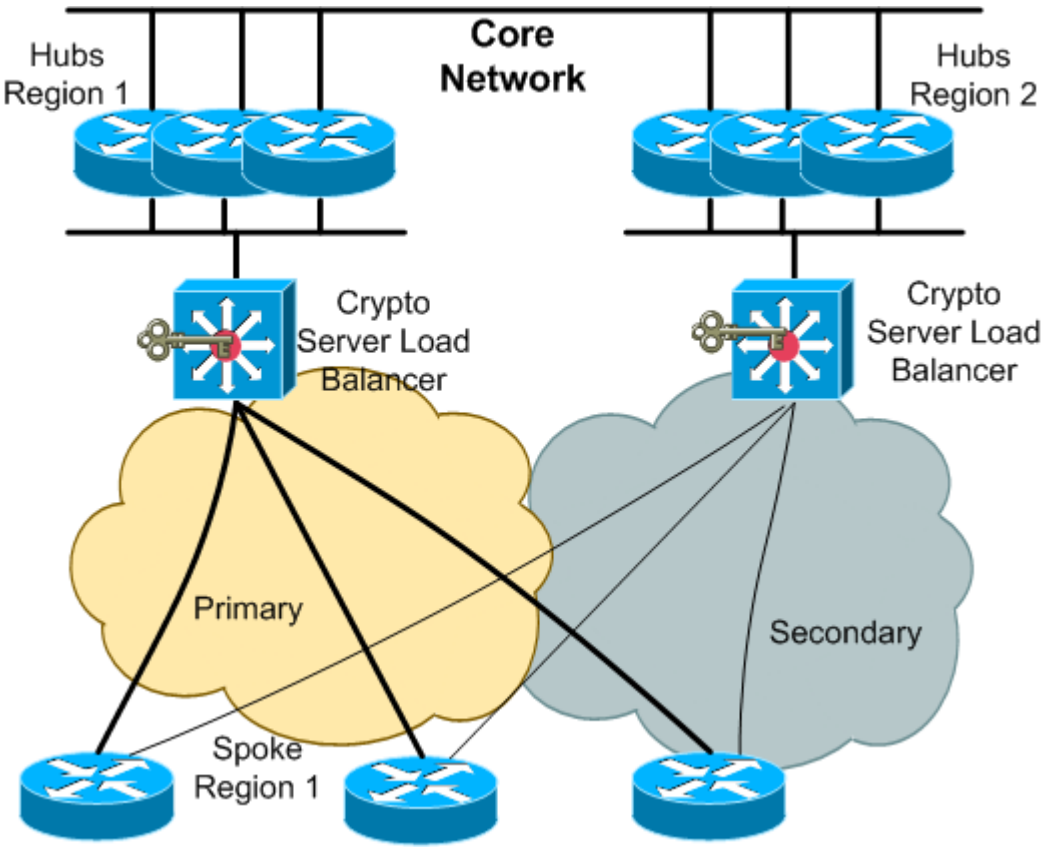
A common concern in all HA headend designs are the number of routing protocol neighbors. Many redundant neighbor relationships increase the number of advertised prefixes and consequently increase the time required for routing convergence. Routing protocol convergence is a common element in all DMVPN headend designs. However, in the SLB HA model, the burden on the routing protocol is divided among the headend devices, providing greater site scale. However, each headend router is limited individually by a given scale limit. We look at this in Section 5.6.4, “Scale Limits,” in this chapter.

5.3.1 Dual Site HA Design

In the SLB design, there is internal HA among the hubs. If a hub goes down, reconnections are evenly distributed among the other available headend routers. Even though the headend server farm is redundant, the SLB router is still a single point of failure.

This problem is simply solved using geographic redundancy. In this HA model, it is recommended to configure the spokes to have secondary next hop server pointing to a secondary DMVPN SLB hub site. If the primary site goes down, the spokes have an alternate location to which they have already built a tunnel. The routing protocol running over the tunnels will direct traffic to the primary or secondary hub site, as shown in Figure 5-3.

Figure 5-3. Dual 6500 Crypto SLB Site Design



This HA model works well if the enterprise already has multiple hub locations. Note that deploying this model applies the *n*+1 rule to a larger number of spokes. If a hub site goes down, all spokes will direct their attention to the secondary hub. Now the value of *n* has doubled. In this case, the backup site should have enough routers to handle the full load.

The branch router has two tunnel interfaces. The delay metric is increased on the tunnel that connects to the redundant hub. With EIGRP, it is recommended that the delay metric be used to retard a metric, rather than the bandwidth statement. EIGRP uses the configured bandwidth on a link to determine how quickly it can send routing updates. The branch router uses the tunnel key keyword to distinguish which tunnel inbound traffic is destined to. Because the branch router sources both mGRE and crypto tunnels from the same IP address, the shared keyword is used with the IPsec profile.

Primary Headend Configuration

```
interface Tunnel1
 description ****Tunnel to Spokes****
 bandwidth 10000000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1440
```



```
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 tunnel source Loopback1
 tunnel mode gre multipoint
 tunnel key 1
 hold-queue 4096 in
 hold-queue 4096 out
!
```

Secondary Headend Configuration

```
interface Tunnel1
 description ****Tunnel To Spokes****
 bandwidth 10000000
 ip address 10.82.0.1 255.255.0.0
 no ip redirects
 ip mtu 1440
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 delay 100000
 tunnel source Loopback1
 tunnel mode gre multipoint
 tunnel key 2
 hold-queue 4096 in
 hold-queue 4096 out
!
```

Branch Configuration

```
interface Tunnel1
 bandwidth 10000000
 ip address 10.81.0.21 255.255.0.0
 no ip redirects
 ip mtu 1440
 ip hello-interval eigrp 10 30
 ip hold-time eigrp 10 90
 ip nhrp authentication nsite
 ip nhrp map 10.81.0.1 10.72.2.2
 ip nhrp map multicast 10.72.2.2
 ip nhrp network-id 102
 ip nhrp holdtime 90
 ip nhrp nhs 10.81.0.1
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
```



```
 load-interval 30
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
 tunnel key 1
 tunnel protection ipsec profile VPN1 shared
 hold-queue 4096 in
 hold-queue 4096 out
!
interface Tunnel2
 bandwidth 10000000
 ip address 10.82.0.21 255.255.0.0
 no ip redirects
 ip mtu 1440
 ip nhrp authentication nsite
 ip nhrp map 10.82.0.1 10.74.2.2
 ip nhrp map multicast 10.74.2.2
 ip nhrp network-id 2010102
 ip nhrp holdtime 90
 ip nhrp nhs 10.82.0.1
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 delay 100000
 tunnel source GigabitEthernet0/0
 tunnel mode gre multipoint
 tunnel key 2
 tunnel protection ipsec profile VPN1 shared
 hold-queue 4096 in
 hold-queue 4096 out
!
```

5.4 Headend QoS

Implementing quality of service (QoS) on the headend is often necessary, because spoke inbound physical bandwidth can become congested. The actual hub router has a much faster physical interface that does not become congested as fast as the spoke Internet connection (that is, the hub can overrun the spoke). Because of this, it is useful to configure the hub with a shaped queuing policy. This is a hierarchical, modular QoS CLI (MQC) based shaper with a child policy for queuing the traffic after shaping is performed.

For traffic from the headend toward the branch, we can use any of several methods to classify traffic for the correct shaper and queue to a specific destination branch. One way is to use regular access-lists to match on the private subnet behind the remote spoke. To classify using this internal IP address, you must use the **qos pre-classify** command on the tunnel interface. Alternatively, you can also simply match on the GRE destination IP address if the branch has an interface with a static IP address that is ued as the tunnel source address.

The QOS scheme in the example is fairly simple. Traffic to that spoke is shaped to 256Kbps, and LLQ and CBWFQ is enforced by the child policy. Within each shaper, a service policy is applied that gives voice traffic a priority of 64Kbps and two other classes of traffic 32Kbps each, and no reserved bandwidth for best effort traffic.

Headend Configuration

```
class-map match-all class_spoke.1.1
```



```
 match access-group name Spoke1
!
policy-map test
 class class_spoke.1.1
  shape average 256000
  service-policy child
!
class-map match-all HP1
 match ip precedence 1
class-map match-all HP2
 match ip precedence 2
class-map match-all voice
 match ip precedence 5
!
policy-map child
 class HP1
  bandwidth 32
 class HP2
  bandwidth 32
 class voice
  priority 64
!
ip access-list extended Spoke1
 permit ip host 220.2.1.100 214.1.1.0 0.0.0.255
!
```

Anyone who works with QoS knows that its CPU and memory consumption is very large. The limitations can be closely correlated with processing power and packet storage while delaying their transmission time slot. With DMVPN, it becomes a little more complicated when you add the encryption card, and the interaction of queuing and processing with hardware encryption.

Looking at the Cisco 7200 series router with NPE-G1 and NPE-G2 processors, we can compare the processing of the QoS policy. Because this policy has three queues and a shaping queue, the performance hit is quite substantial. Obviously, throughput through the QoS policy adds to the CPU load. Because the 6500 SLB offloads crypto processing, however, the headend routers have more CPU to use for processing QoS.

Note: The 7200 NPE-G2 headend can support about 200 active shapers, while the 7200 NPE-G1 headend can support about 120 active shapers. The active shapers are those that actually shape traffic, where the offered load is higher then the shape rate. Other shapers may be defined but inactive (the offered load is lower then the shape rate). Inactive shapers do not increase CPU load. This scalability also includes maintaining EIGRP neighbor relationships with all the peers. The tested code contains the fix for CSCek72110, which improves QoS scalability significantly.

5.5 IP Multicast

Multicast on DMVPN is similar in behavior to any NBMA network. These multicast deployments typically have a hub to spoke direction of multicast data flow, where the source of the multicast stream is located somewhere in the core network. Because the DMVPN cloud is a nonbroadcast multiaccess (NBMA) network, Protocol Independent Multicast (PIM) sparse mode must be used with PIM NBMA mode enabled on the hub device. The receivers join at the branch site, and the PIM join is sent over the DMVPN tunnel toward the rendezvous point (RP).

The RP is placed somewhere in the core network, or perhaps at the DMVPN headend. The RP cannot be configured at a spoke. You can use any method of distributing group to RP mapping information. It is suggested to set the shortest-path tree (SPT) threshold to infinity.

Direct spoke to spoke multicast is *not* supported. The multicast source can be at a spoke, but the multicast flows from the spoke, to the hub, and then back to the spokes that want to receive the traffic. In this case, you *must* set the SPT threshold to infinity so the IP multicast packets stay on the shared tree through the hub and do not attempt to switch to the source tree through a spoke-to-spoke tunnel, which will not work. Using PIM NBMA mode enables the hub to forward the multicast only to those spokes that have a receiver behind them, instead of to all spokes on a DMVPN network that a receiver has joined.

Headend Configuration

```
interface Tunnel1
 ip address 10.81.0.1 255.255.0.0
 ip pim nbma-mode
 ip pim sparse-mode
!
ip pim rp-address 10.81.0.1
!
```

Branch Configuration

```
interface Tunnel1
 ip address 10.81.1.1 255.255.0.0
 ip pim sparse-mode
!
ip pim spt-threshold infinity
!
```

Multicast scalability for DMVPN is basically a question of packet replication on the headend router and throughput of the incoming stream. This can be a very CPU intensive process that degrades as the scale increases. In this design, you should have the multicast source behind the hub routers so that each multicast hub can replicate the multicast traffic to its spokes that request the stream.

Because crypto processing is done on the 6500 VPN-SPA, the headend routers can focus on processing the replication and forwarding of the multicast data. The 6500 receives the GRE tunnels, and encrypts the data as it arrives from the headend routers. The headend routers can replicate and GRE-encapsulate the multicast packet very quickly, which can in a large number of packets being sent to the 6500 in a short time. If there is a congestion point, either at the 6500 or up through the ISP connection, sufficient buffer space must be provided to spread out packet bursts so packets are not dropped.

5.6 Scaling 6500 Crypto SLB Hub Deployment

Now that we have covered the basic constructs of the 6500 crypto SLB design, we can look at how this DMVPN design scales. It should be apparent that a routing protocol is required to manage the prefixes and reachability in this type of network. DMVPN hub scalability depends most dependent upon the scalability of the routing protocol used between the hubs and spokes. Looking at this closely, it is also apparent that scalability depends upon the number of prefixes that the routing protocol must advertise. To reduce overhead, we can advertise a default route or large summary to the spokes. The spokes are basically stub networks in relation to the DMVPN network as a whole, and are typically not used as transit networks.

Summarization of the spoke private subnets at the hubs is also possible if the subnet allocation was designed properly. Obviously, not all routing protocols have the same scale limitations or scale. We look at two protocols that scale fairly well within DMVPN cloud; EIGRP and RIP. The routing protocol running between the hubs can be any protocol; however, BGP is the protocol of choice between hubs and the rest of the campus network.

The other bottleneck in all encrypted networks is the processing required to encrypt and decrypt. This design is not affected by this problem as much as the deployment designs discussed earlier in this DIG. This design, by utilizing the 6500 VPN-SPA, offers the highest throughput of any design. This conserves CPU power on the headends in the server farm. There are no hardware encryption modules on the actual DMVPN hubs. The 6500 feels the crypto processing overhead, but the VPN-SPA is much more powerful than the 7200 hardware crypto engines. Spoke devices still require hardware encryption.

5.6.1 Data Plane Considerations

This section lists and describes data plane considerations, including encryption throughput, router CPU utilization, and unicast and multicast packet processing.

5.6.1.1IPsec Encryption Throughput (Hardware Encryption Engines)

IPsec encryption engine throughput in each platform (6500 VPN-SPA or branch) must be considered for scalable designs, because every encrypted/decrypted packet must traverse the encryption engine. Therefore, encryption throughput must consider bidirectional speeds. In general, encryption throughput is reported as a value that includes both encrypting and decrypting packets. For example, if encryption cards throughput is reported as 100 Mbps, the encryption card can encrypt and decrypt at a combined rate of 100 Mbps (100/0 Mbps (encrypt/decrypt) to 50/50 Mbps to 0/100 Mbps).

In general, as throughput increases, the burden on router CPU also increases. However, with hardware-accelerated encryption available for all Cisco router products from the 871 through the 7200, most encryption processing is offloaded from the main CPU to the VPN hardware. In this case, the hardware encryption cards take over the processing for doing the actual encryption/decryption and mathematically intensive activities like Diffie-Hillman (DH) calculations. However, main router CPU processing still occurs, so higher throughput typically results in higher CPU consumption. For the 6500, 7600, and 7200 with VSA card, CPU processing per encrypted/decrypted packet is reduced further.

5.6.1.2Unicast Packet Processing

Although bandwidth throughput capacity must be considered, the packet rate for the connection speeds being terminated or aggregated is more important. In general, routers and encryption engines have upper boundaries for processing a given number of packets per second (pps). The size of packets used for testing and throughput evaluations can understate or overstate true performance.

It also turns out that encryption engines use about the same amount of resources to encrypt a large packet as a small packet. The pps that an encryption card can handle tends to remain constant as packet size varies. This also means that the encryption throughput in bits per second can vary widely as packet size changes. For example, if an encryption card could encrypt 10,000 pps then with 60 byte (480 bits) packets, it would do 4.8Mbps. With 1500 byte (12,000 bits) packet, the same hardware would do 120Mbps.

Because of such a wide variance in throughput, pps is generally a better parameter to determine router forwarding potential than bits per second (bps). Headend scalability is the aggregate forwarding potential of all branches that terminate a tunnel to that headend. Therefore, the aggregate pps from all branches affects the headend pps rate.

Throughput varies per platform, and depends largely on the traffic pattern. Recommendations in this chapter are based on a nominal Internet mix (IMIX) traffic pattern, with router CPU utilization averaging 70%.

5.6.1.3Multicast Packet Replication

Multicast traffic affects a router more than unicast processing. This effect is a function of the number of PIM receivers. If multicast is used in the DMVPN network, the design should include fewer receivers per hub than would be used with unicast.

5.6.2 Data Plane Best Practices

You should plan your design for above-average throughput. This prevents surprises when traffic bursts and the CPU does not have enough cycles to process the control plane packets.

It is also recommended to follow these best practices:

- IP MTU – Set the IP maximum transmission unit (MTU) to 1400 on all DMVPN tunnel interfaces to eliminate the potential for fragmentation. GRE and IPsec headers add about 60 bytes to the packet, and cause the router to fragment larger packets if this exceeds the interface MTU, straining the CPU. The value 1400 is easy to remember and leaves a little extra room; other protocols also "steal" bytes from the payload, such as Network Address Translation-Traversal (NAT-T) (8 bytes) and Point to Point Protocol over Ethernet (PPPoE) (4 bytes).
- TCP MSS – Set the TCP maximum segment size (MSS) value to 1360 on all DMVPN tunnel interfaces. This value is calculated by subtracting 40 bytes from the IP MTU value. Use the command **`ip tcp adjust-mss 1360`** to set the value on the mGRE tunnel interface toward the spokes. This helps TCP sessions adjust to the lower MTU and is needed if Path MTU Discovery (PMTUD) does not work between end hosts.
- Tunnel Bandwidth – The **`bandwidth`** statement is recommended on the tunnel interface of hub and spoke routers, although the actual value can vary from scenario to scenario. Without the **`bandwidth`** statement, tunnel interfaces are currently allocated very low interface bandwidth (9 Kbps). The default will soon be raised to 100 Kbps. This could affect QoS or other features, such as routing protocols that use the configured bandwidth.
- Hardware Crypto acceleration – Always use a hardware encryption module to do most of the encryption/decryption math. The SLB device (6500) will have a VPN-SPA and the spokes should have hardware encryption engines as well.
- QoS – Provision Quality of Service (QoS) policies as necessary at the headend and branch routers. This helps alleviate interface congestion and ensures that latency sensitive traffic is prioritized over other traffic. If classification is done on the internal packet, you must configure **`qos pre-classify`** on the tunnel interface.

5.6.3 Control Plane Considerations

This section lists and describes control plane considerations, including tunnel aggregation stability, encryption throughput, routing protocols, route summarization, and stub routing.

5.6.3.1Tunnel Aggregation Scalability

You must consider the maximum number of IPsec tunnels that a headend can terminate. Tunnel scalability is a function of the number of branch routers that are terminated to the headend aggregation point. This number must include the primary tunnels and any alternate tunnels that each headend might be responsible for in the event of a failover.

The number of IPsec tunnels that can be aggregated by a platform, and the encryption pps rate, are the primary determining factors for recommending a platform.

Although throughput depends highly upon platform architecture, as tunnel quantities increase, overall throughput tends to decrease. When a router receives a packet from a different peer than the peer whose packet was just decrypted, a lookup based on the security parameters index (SPI) of the new packet must be performed. The transform set information and negotiated session key of the new packet is then loaded into the hardware decryption engine for processing. For traffic to be encrypted, the main CPU must match the data packet to the correct IPsec security association (SA) to hand to the encryption engine to encrypt the packet. Having traffic flow on a larger numbers of SAs tends to negatively affect throughput performance.

Increasingly, platforms with hardware-accelerated IPsec encryption are also designed to offload IPsec SA processing overhead, resulting in more linear performance regardless of the number of SAs. For example, the VPN SPA blade for the Cisco 7600 has fairly linear throughput whether the traffic load is offered on a few SAs or several thousand. Currently, the VPN SPA (6500/7600) and the VSA (7200 NPE-G2) are the only encryption processors that have this functionality.

The GRE configuration affects router encryption throughput. GRE headers are added to the beginning of each packet, and GRE headers also must be encrypted. The GRE encapsulation process, when not hardware-accelerated, increases total CPU utilization. In most cases, the amount of CPU overhead due to GRE encapsulation is quite small compared to the CPU overhead due to IPsec encryption.

5.6.3.2Crypto SLB

In this design, SLB is an integral part of the DMVPN hub site. In this design, the encryption/decryption is performed by the crypto accelerator in the 6500 doing the SLB. Here are a few recommendations for the SLB configuration:

- Max Connections – Use max-connections configuration to limit the number of spokes connecting to the real server. This value is dictated by the scalability of the routing protocol running on the real server.
- Idle timer – Set the idle timer to be greater than the sticky timer. This keeps SLB from prematurely clearing connection information for the tunnel to the real server. If SLB were to clear this entry, the tunnel could be re-established to another real server, thus creating duplicate NHRP information on a different hub. This becomes very important since the hub devices in the serverfarm will still maintain the NHRP cache information. Therefore it is important that the DMVPN GRE tunnels continue to go to that hub once the crypto is re-established (See *Sticky Timer* for more details).
- Sticky timer – Stickiness refers to the functionality of remembering which "real" hub device the connection is being sent to. The sticky timer should be shorter than the idle timer, but *no less than* 5 minutes (300 seconds). In the event of any failure or reload of the VPN-SPA, causing all crypto tunnels to go down, this ensures the SLB will maintain the sticky state. If the sticky timer (or idle timer) value is less than the time it takes for tunnels to re-establish, the state will be deleted and the SLB code will direct many of the reformed connections to hubs other than the ones that they were connected to prior to the crash. The NHRP cache entries for the old connections will in all likelihood be still in place on the hub terminating the tunnel previously. With the addition of the new NHRP entries there will be approximately double the number of NHRP cache entries than there will be routing protocol neighbors. Since the multicast replication code uses the NHRP cache entries to determine which devices require the multicast encapsulated routing protocol updates, the amount of routing protocol convergence traffic will be doubled until the NHRP cache entries time out.

- Buddy group – In this design the GRE protocol is being load balanced after the decryption from the VPN-SPA. Since this is a solitary protocol being load balanced, buddying is not necessary for this design.
- SLB Probe Timeout – Ensure that the combination of the number of failed probes and the interval of time between probes consumes a greater amount of time than the routing protocol timeout value for spoke originated routes. If the probe timeout is set too aggressively a router that has a momentary spike in traffic may be declared dead. The spoke sessions connected to this hub would then be moved over to the other hubs. The first hub will still have the spokes in it its neighbor table and continue to send updates to them. This will lead to confusion on the spokes since they will be receiving conflicting updates from what appears to them to be the same IP address.
- Purge failed connections – Configure SLB to automatically remove connections to failed real servers and firewalls from the connection database even if the idle timers have not expired.

5.6.3.3Routing Protocols

Running a routing protocol affects CPU overhead. Processing keepalive and hello packets and maintaining a routing table uses CPU time. The amount varies with the number of routing peers and routing table size. The network manager should design the routing protocol based on generally accepted practices for the particular routing protocol.

Routing protocol scalability is the most important limiting factor in determining the number of branches a given hub can support. Scalability becomes more cumbersome as the number of routes that the protocol must advertise to each branch increases. We will look at EIGRP and RIPv2, along with various options that can be used to increase scalability.

5.6.3.4Route Summarization

Hub routers forward data traffic based on the routing table. Based on the routing table, summarization can occur in the core of the network, and in the spoke networks. This enables us to build a network with a hierarchical design having smaller, more efficient routing tables. In general, in a hierarchical network, at each level the routing protocol should summarize routes going up the tree toward the spokes (leaves), but send full routing information to nodes in the same level (region) or down the tree (toward the root).

Most dynamic routing protocols support summarization. However, you might want to run a separate routing protocol, such as BGP, between hub routers within the same region or within the core. Regional hub routers must have full routing knowledge of their regions. The regional hubs can summarize their routing tables when advertising them to the spokes and other hubs.

The protocols should be configured to advertise very few routes toward the branch routers. This should, in most cases, be a default route, or, in few cases, a summary route for the rest of the network. If a branch router uses split tunneling, a default route should not be advertised to the branch unless the branch uses VPN routing/forwarding instances (VRFs), in order to segregate split-tunneled traffic.

5.6.3.5Stub Routing

Branch routers are rarely used as transit networks. Therefore, it is best to configure branch routers as stub networks. Using EIGRP, you can set the routing protocol to behave as a stub; it announces this functionality in hello packets to the headend router. Using the **`stub connected`** command, the branch router sends

connected prefixes to the headend. The headend has reachability of the branch networks, and does not have to include the branch router in the query process.

Note: Be careful to not advertise the tunnel source interface subnet to the hub router through the tunnel interface. If the hub uses this route the tunnel packets to this spoke will go into a forwarding loop that will either hang or crash the hub router. To protect against this possibility, either block this network from being advertised by the spoke or being accepted by the hub.

Spoke configuration

```
router eigrp 10
 network 10.80.0.0 0.3.255.255          ← DMVPN Tunnel subnet
 network 10.211.9.0                     ← Private VPN subnet
 no auto-summary
 eigrp stub connected    ← Announces stub router, advertise connected intf.
!
```

In RIPv2, there is no stub keyword. This protocol must operate in the native manner. Alternatively, you can make the mGRE tunnel on the hub passive and use reliable static routing on the branches. In this case, the hub still receives RIP advertisements from the spokes, but does not send the spokes RIP advertisements. Reliable static routing on the branch router uses the IP service level agreement (SLA) feature to poll an address on the primary headend using a ping probe, coupled with a tracking object tied to a static route.

As long as the ping probe operates, it installs a static route directing traffic toward the primary headend over the tunnel. If the probe is nonoperational, the static route is removed and a floating static route having the same network and mask, but higher administrative distance (AD), directs traffic to the secondary headend.

5.6.3.6Control Plane Best Practices

Here is a list of options that should be used to increase stability of the design:

- Protocol Timers – Set the routing protocol dead timer to a larger value. This provides more time for the WAN to recover in the event of a minor problem. Otherwise, a minor flap could cause Interior Gateway Protocol (IGP) running over the tunnels to go down and require reconvergence. Also, consider leaving the hello timer the same, rather than increasing it. This helps keep the routing protocol from flapping if the WAN network is somewhat lossy.
- Interface Queues – Set the interface input and output hold queues to 4096 on the tunnel interface and output WAN interface of the headend, and possibly other routers on the path from the headend to ISP router, to minimize protocol failure.
- PKI and Digital Certificates – Use digital certificates and Public Key Infrastructure (PKI) for the ISAKMP authentication. This increases manageability as the site count increases. Preshared keys (PSK) are not a scalable solution for distributing Internet Key Exchange (IKE) authentication credentials. PKI is highly scalable and secure. Be sure to turn off certificate revocation list (CRL) checking because it periodically drains CPU power.
- Dead Peer Detection – Enable dead peer detection (DPD) on hubs and spokes. This increases the speed at which a failure over the IPsec tunnel or connectivity loss is detected. This should be set higher than the routing protocol update timer, but less than the dead timer.

  Set the **`crypto isakmp keepalive`** timer value to be greater than the RP hello timer by 5 seconds. Set the total ISAKMP keepalive + 5* retry timer for a timeout equal to RP dead timer + 60 seconds. This prevents ISAKMP and IPsec sessions from tearing down before losing routing protocol adjacency.


```
crypto isakmp keepalive <hello+5 (sec)> <(dead+60-(hello+5))/5)(sec)>
```

Example: Hello timer = 10 secs and dead timer = 60 seconds; then keepalive = 15 (10+5) and retry = 21 (60+60-(10+5))/5

```
crypto isakmp keepalive 15 21
```

- Call Admission Control on hub – In failure scenarios, IKE processing can be debilitating for a router. It is recommended to enable Call Admission Control (CAC) to limit the number of IPsec sessions that can come up simultaneously. This prevents incoming IPsec connections from overrunning the CPU. There are two CAC methods

  Configure the absolute IKE SA limit so the router drops new IKE SA requests after the number of IKE SAs in negotiation reaches the configured limit. After the number of IKA SAs in negotiation drops below the configured limit, additional IKE SAs are processed.

```
crypto call admission limit ike in-negotiation-sa [number]
```

Configure the system resource limit so the router drops new IKE SA requests when the specified percentage of system resources is in use. It is recommended to set this at 90% or less.

```
call admission load [percent]
```

5.6.4 Scale Limits

This section examines specific DMVPN scaling limits of the routing protocols and hardware crypto engines. Remember, throughput is not a focus, but a set parameter to load the system to a high operating load. The following sections illustrate the DMVPN peer scalability for the hierarchical hub design using a Cisco 7200 router serving as the headend. With the 6500s VPN-SPA handing encryption/decryption, the scale can be quite large. Obviously, high aggregate throughput degrades this scalability since the VPN-SPA has more data packets to encrypt/decrypt. It is also recommended to make the server farm routers uniform using the same platform and processor.

It is very important to understand how scaling numbers are found and what to expect. All internal testing to derive the numbers was done in a lab, so it is likely the numbers stated are optimal for the design. The scalability is determined by both the number of DMVPN peers the design can support, and the data throughput at the maximum scale. In the following data, notice the number of routing protocol peers does not necessarily increase with a better processor. Rather the scale of the design is consistant, and the amount of throughput increases using a more powerful route processor.

There have been great improvements in the scalability of the routing protocols for DMVPN. Depending on the IOS image your router is running, scalability varies. It is recommended to run either IOS 12.4(9)Tx or 12.4(15)T2 and above with the Advanced Enterprise Security (adventerprisek9) feature set. IOS 12.4(11)Tx *should not* be run with DMVPN and QoS.

Note: These results are based on IOS 12.4(15)T4 with the Advanced Enterprise Security (adventerprisek9) feature set for the headend. The 6500 crypto SLB router is running IOS 122-18.SXF11 with the Advanced Enterprise Security (adventerprisek9) feature set.

5.6.5 EIGRP

Cisco recommends using EIGRP as the routing protocol over your DMVPN network. EIGRP offers fast convergence and robustness. It is recommended to configure the spokes to be EIGRP stub peers. This minimizes convergence times and simplifies the convergence process. Cisco also recommends using the

summarization potential offered with DMVPN Phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes.

Looking at the 7200 Series router as the headend, you can use two hardware variations. For the route processor, you should use either NPE-G1 or the NPE-G2. The 6500 Series with a SUP720-3BXL is used for SLB, and for crypto processing with the VPN-SPA and SSC-200 jacket card.

Table 5-1 shows recommended and maximum limits for EIGRP peers per headend with each hardware combination. The 6500 with VPN-SPA is common with any 7200 processor.

Table 5-1. Recommended and Maximum EIGRP Peers

EIGRP Peers	Recommended	Maximum*
7200 NPE-G1	600	900
7200 NPE-G2	800	1300

* The Maximum column indicates the maximum peer limit in an ideal environment, such as a lab. It *must* be understood that the maximum results are probably unobtainable in a real network, where more delay and other variables that adversely affect routing protocol convergence.

Note: These numbers assume that summarization to the spokes is used and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is about a 70% load on the router. If your network parameters differ, your results may vary from the reported numbers, which are provided as planning guidelines.

It should be understood that this is the scale of one headend device. So, if you have two 7200 headend routers in the server farm, you can terminate twice the number peers listed in Table 5-1. As has been stated many times in this DIG, the routing protocol is the greatest limiting factor for DMVPN tunnel termination.

During times of convergence EIGRP will generate much more traffic then during periods of network quiescence. The following table lists the approximate peak amount of traffic in bits per second that EIGRP will generate outbound over the WAN for a given number of spokes and prefixes advertised towards the spokes. This includes the encryption overhead of the DMVPN tunnel. These numbers can be used to help determine the size of the WAN link to the Internet. Actual data traffic will of course add to these numbers. EIGRP will in all likelihood converge even if the WAN link is smaller than the given value but there will probably be some packet loss due to outbound packet drops.

As can be seen in the table, the number of prefixes advertised does not have a great impact on the traffic rate. The fact that the single prefix numbers are greater in some cases than the fifty prefix numbers is probably due to sampling errors. The inbound traffic rates are generally much lower.

Table 5-2. Minimum WAN banwidth needed by EIGRP for a given number of peers and prefixes advertised

Number of EIGRP Peers	Max. BPS for 1 Prefix Advertisement	Max. BPS for 10 Prefix Advertisement	Max. BPS for 50 Prefix Advertisement
600	2887000	2963000	3669000
1200	10075000	12886000	15752000
1800	20577000	25939000	27732000
2400	28717000	29845000	34397000
3000	36329000	35997000	36672000
4000	55736000	55056000	55004000
5000	69670000	68820000	68755000
6000	83604000	82584000	82506000

5.6.6 RIPv2

Cisco supports RIPv2 as the routing protocol over a DMVPN network. RIP is a simple, effective protocol in a DMVPN network. RIP can scale very high compared to other routing protocols over DMVPN. It is recommended to use the summarization potential offered with DMVPN phase 3. This reduces routing table size on the headend routers, and reduces the number of prefixes advertised to the spokes. Using the passive interface with reliable static routing (basically stub branches in RIPv2) did not provide any benefit compared to native RIP. The reason for this is simply the input queue limits the number of updates creating a minor bottleneck. So, regardless of active or passive RIP, the updates will still be sent from the spokes every 30 seconds. In production networks, the updates sourced from the spokes should not have the same synchronization as we see in a lab environment.

Looking at the 7200 Series router as the headend, you can use two hardware variations. For the route processor, you should use either NPE-G1 or the NPE-G2. The 6500 Series with a SUP720-3BXL is used for SLB, and for crypto processing with the VPN-SPA and SSC-200 jacket card.

Table 5-3 shows the Cisco recommended limit of RIP peers per headend with each hardware combination. The 6500 with VPN-SPA is common with any 7200 processor.

Table 5-3. Recommended and Maximum RIPv2 Peers

RIP Peers	Recommended	Maximum*
7200 NPE-G1	1125	1125
7200 NPE-G2	1600	2000

* The Maximum column indicates the maximum peer limit in an ideal environment, such as a lab. It *must* be understood that the maximum results are probably unobtainable in a real network, where more delay and other variables that adversely affect routing protocol convergence.

Note: These numbers assume that summarization to the spokes is used and the number of advertised prefixes to the spokes is 50 or less. Furthermore, throughput at a steady state is about a 70% load on the router. If your network parameters differ, your results may vary from the reported numbers, which are provided as planning guidelines.

It should be understood that this is the scale of one headend device. So, if you have two 7200 headend routers in the server farm, you can terminate twice the number peers listed in Table 5-1. As has been stated many times in this DIG, the routing protocol is the greatest limiting factor for DMVPN tunnel termination.

5.6.7 Unicast Throughput

In this design the 6500 VPN-SPA handles all crypto functions for all headends in the server farm. With the 6500 and VPN-SPA, all crypto processing handled by a different box. Where is the speedup, if not in the scale of the DMVPN peers? The answer is in the encryption/decryption throughput this design can handle.

The 6500 can terminate 4000 IPsec connections, so the server farm can have many headends. The number of headends used depends on the protocol and processor used. Assume we can terminate 800 peers per headend, using the *n*+1 approach would mean six headends in the server farm. This is just simple math.

Because throughput is really the power of this design when compared to the 6500 SLB design without VPN-SPA, it is important to know the throughput limits of the design. The VPN-SPA has a throughput rating of 2 gigabits per second (Gbps).

Table 5-4 shows aggregate throughput using the 6500 VPN-SPA and one 7200 headend. The table shows the maximum throughput of clear traffic (not encrypted) for each 7200 processor at 70% CPU utilization and forwarding an IMIX pattern of traffic (7×64:4×570:1×1400).

Table 5-4. Aggregate Unicast Throughput

Bidirectional Throughput	PPS	Approx. Mbps
7200 NPE-G1	138,000	368
7200 NPE-G2	170,000	454

Note: These values show the approximate maximum aggregate throughput for the given processor when using this DMVPN design. The constraint of the VSA or VAM2+ is lifted in this design, and the routers can forward closer to their maximum throughput.

Note: Obviously, the values do not even approach the throughput rating of the VPN-SPA for a single headend router, but if you have 6 headend routers you can reach rates that are more than can be handled by a VPN-SPA. All DMVPN tunnels can only utilize a single VPN-SPA module on the 6500 because all traffic uses a single source IPsec crypto peer address and interface.

Table 5-5. Aggregate Unicast Throughput with Six 7200s

Bidirectional Throughput	PPS	Approx. Mbps
6 x 7200 NPE-G1	828,000	2,208
6 x 7200 NPE-G2	1,002,000	2,724

Note: The bandwidth numbers *do not* represent real traffic, but are found by simulating a realistic traffic pattern in the lab. Real traffic does not always follow a given pattern, so throughput values in a real network differ from those found in lab testing. Also, these numbers assume throughput at a maximum CPU load of 70% on the router. If your network parameters differ, your results might vary from the reported numbers, which are provided as planning guidelines.

5.6.8 Multicast Throughput

In a DMVPN network, the headends handle all multicast replication processing, and the 6500 and IPsec VPN-SPA handle all e crypto functions after replication on a separate platform. The throughput of the input stream is magnified by the number of peers receiving the stream. Fortunately, in this design the multicast replication processing load is distributed across multiple routers. The RP would be located behind the server farm in the core network so all headends can reach it.

Table 5-6 represents one headend behind the 6500 crypto SLB. The scale of the server farm is a function of the number of headend routers.

In the lab, the code and configuration were the same on the NPE-G1 and NPE-G2. It is surprising to see that the NPE-G1 with VAM2+ performed much better in multicast forwarding than the NPE-G2. It is suspected that the NPE-G2 switching rate is so fast that it causes buffer starvation on the 6500 VPN SPA Crypto card.

Table 5-6. Multicast Throughput

Hub Processor	Maximum Number of Peers	Interrupt CPU Utilization on Hub	Total PPS	Kbits per Second Ignoring GRE and IPSEC Over Head
NPE-G1	1520	60%	54,720	56,033
NPE-G2	340	7%	12,580	12,882

6 DMVPN Branch Designs

In the preceding chapters, DMVPN branch router configurations were only covered briefly This chapter focuses on the branch routers, some designs in which the branch routers can be implemented, and some branch router features that can be configured.

Most DMVPN design is centered on the hub side of the network. However, the spoke side of the network can also be examined because this is typically the edge device for a given branch site. The router typically also needs protection, such as a firewall or intrusion detection system or intrusion prevention system (IDS/IPS).

Some branches might concern themselves with the type of uplink to the ISP. A branch can connect to the Internet using cable or DSL service as a low cost Internet connection. Other branches require connections with one or more high speed uplinks to one or more service providers. Furthermore, the spoke device may require QoS, split tunneling, VRF-lite (front VPN routing/forwarding instances (fVRF), inside VRF (iVRF)), or any combination of these.

DMVPN branch routers perform the following functions:

- Initiating tunnels to the headend.

  Branch routers use multipoint generic routing encapsulation (mGRE) to form tunnels with a headend router and can form dynamic spoke-to-spoke tunnels with other branch routers.

- Running an Interior Gateway Protocol (IGP) over the tunnel to distribute internal private routes to the headend router, and receiving summary or other routes from the headend router for access to the main campus network and subnets behind other branches.

See Section 1.3, “DMVPN Phase 3,” for details of Phase 3 enhancements.

6.1 Design considerations

The primary design considerations for DMVPN branch router include:

- Branch access speed – depending on the purpose of the DMVPN (primary or backup connection to the campus network), branch access speed for the DMVPN connection can be T1, T3, cable/DSL, and so on.
- Redundancy – It is strongly recommended to have redundant connections to at least two headend routers.
- IP Addressing – The IP address of the Internet-facing physical interface of the branch router can be a statically or dynamically assigned public address. It is also common to see the branch router placed behind a network address translation (NAT) device (NAT-Transversal (NAT-T) Aware DMVPN).

  A unique mGRE tunnel interface IP address within the tunnel subnet must be allocated and configured on each branch router.

  The addressing scheme of the networks behind the branch router should also be carefully designed to support efficient route summarization. This, along with DMVPN Phase 3 support for route summarization, can greatly reduce routing protocol overhead on the headend routers.

- Routing protocol to run over the DMVPN session – Selecting the IGP to run over DMVPN is critical from the headend router perspective as it plays a major role in scalability. However, impact on the branch router is not a significant design consideration. See Section 2.8, "Routing Protocols with DMVPN," for information about routing protocols in DMVPN.
- Split tunneling – In a basic DMVPN scenario, all traffic from the inside branch network is encrypted and sent to the headend router no matter what its destination. Based on the configuration and the number of supported users, such a setup can become bandwidth intensive. Split tunneling can help alleviate this problem. Split tunneling enables users to send only traffic that is destined for the corporate network across the tunnel. All other traffic, such as IM, email, or casual browsing, is sent directly to the Internet using the outside physical interface of the branch router.

  If non-split-tunneling and dynamic spoke-spoke tunnels are desired, you must configure VRF-lite on the branch router to support two routing tables, where each has its own 0/0 default route. One routing table routes host data traffic out the tunnel interface, and the other routes tunnel packets out the physical interface.
- Other services that the branch router might provide to the branch office – Providing Internet access to branch networks poses a potential security risk. Because VPN clients have unsecured Internet access, they can be compromised by an attacker. That attacker might then be able to access the corporate LAN over the IPsec tunnel. Integrated services such as IOS Firewall, IOS Intrusion Prevention System (IPS), and so on should be implemented on the branch router to protect against this threat. The branch router can also be used to provide other services such as NAT, Dynamic Host Configuration Protocol (DHCP), and so on to the clients in the branch network.

6.2 Branch Designs

A DMVPN branch router can be configured using many different designs and implementations. This section describes some popular designs:

- Multiple DMVPN clouds, based on how many routers connect to multiple ISPs for multiple DMVPN clouds
- DMVPN used as a backup to another primary ISP connection, based on how many routers provide multiple connections

6.2.1 Multiple DMVPN Clouds

If DMVPN is the primary means of connection between the branch router and the headend routers, it is recommended to include redundancy and failover mechanisms as part of the design. The following designs describe two possible ways of connecting the branch end router using multiple ISP connections.

Note: DMVPN cloud topologies are introduced in Section 2.1, "DMVPN Topology."

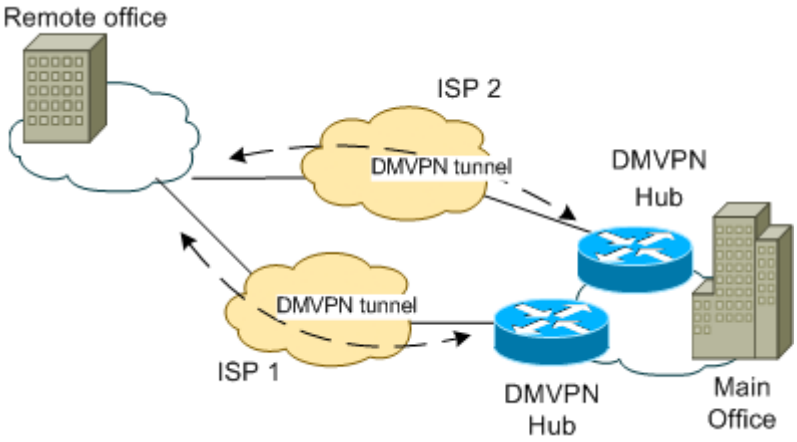
There are two common ways for customers to connect to multiple ISP networks. In the first, one router connects to both ISP devices. In the second, a separate router connects to each ISP. In the first case, routing is trivial. In the second, it is common for the two customer routers to run a routing protocol between the two routers to keep track of ISP reachability. Also, the customer LAN side of these routers may be connected to a firewall. The firewall must either run a routing protocol with the two ISP connected routers, use Hot Standby Router Protocol (HSRP) to determine its default gateway, or use Gateway Load Balancing Protocol (GLBP) to provide load balancing.

Another design choice is whether to use one DMVPN network that can run over either ISP, or use two DMVPN networks, one within each ISP. Some customers might want to load balance over the two ISP connections while others may want a primary/secondary type of solution in which the secondary ISP is not used unless the primary goes down.

6.2.1.1Single Branch Router for Multiple ISP Connections

For a small branch office, one branch router can be used to connect to the main campus network over multiple ISP connections, each terminating on a separate headend router. This design does not provide dual redundancy (that is, failover at both the branch office and headend). However, the design does provide ISP failover and headend router failover. In addition to improved redundancy this also gives the provision for load-balancing branch traffic across the multiple ISP connections. Load-balancing is provided by the IGP routing protocol. The next section (6.2.1.2) shows another method for providing load-balancing using a load-balancing protocol such as GLBP.

Figure 6-1. Single Branch Router Providing Multiple ISP Connections



For simplicity, the topology in Figure 6-1 shows a branch router that connects to two headend routers over two ISP connections. Two DMVPN (mGRE) tunnels are used, one for each ISP, to provide better load-balancing of data traffic over the two ISPs. The IGP (EIGRP in this example) can be used to choose between the available paths and to select the best path, or load-balance over both paths. In case of ISP or headend router failure, IGP can provide the alternate route. The selection of the IGP protocol is critical to the scalability of the DMVPN headend router, while it doesn't have any significant impact from the branch router's perspective. See Section 2.8, Routing Protocols with DMVPN," for more information about selecting an IGP.

The following sample branch router configuration connects to two DMVPN headend routers over separate DMVPN tunnels. The design considerations for this configuration are:

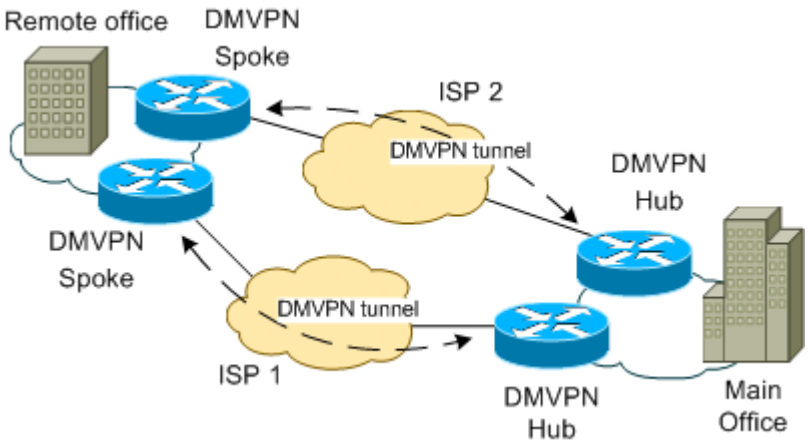
- Both tunnel interfaces have unique NHRP network-ID and IP addresses in unique IP subnets.
- The DMVPN headend router tunnel IP address and the next-hop server IP address are different for each tunnel interface.
- Tunnel source is different for both tunnel interfaces, because a different IP source address is required for packets going over each ISP network.
- The same IPsec profile can be used for both tunnels. Because the tunnel sources are different interfaces, the **`shared`** keyword is *not* used on the **`tunnel protection`** command.


```
interface Tunnel0
 description DMVPN Endpoint1
 ip address 192.168.1.201 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.1
 ip nhrp map 192.168.1.1  10.40.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.1.1
 ip nhrp registration timeout 75
 load-interval 30
 tunnel source GigabitEthernet0/0.41
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
interface Tunnel1
 description DMVPN Endpoint2
 ip address 192.168.2.201 255.255.255.0    ← Different IP subnet from Tunnel0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.2
 ip nhrp map 192.168.2.1 10.40.0.2
 ip nhrp network-id 2
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.2.1
 ip nhrp registration timeout 75
 load-interval 30
 tunnel source GigabitEthernet0/0.42
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
```

6.2.1.2Multiple Branch Routers for Multiple ISP Connections

To provide additional failover at the branch office, multiple branch routers can be used to connect to headend routers over multiple ISP connections. This not only improves redundancy, but also supports the load balancing of branch traffic across the multiple ISP connections.

Figure 6-2.Multiple Branch Routers Providing Multiple ISP Connections



For simplicity, Figure 6-2 shows two branch routers that connect to two headend routers over separate ISP connections. Each ISP connection provides a separate DMVPN tunnel. With multiple branch routers, a load balancing or failover protocol is needed on the internal branch side. In a lab, this design was tested using GLBP (Gateway Load Balancing Protocol) and HSRP (Hot Standby Router Protocol). GLBP provides the benefit of load balancing, while HSRP simply provides a single redundant default gateway. GLBP (or any other load-balancing protocol) can lead to asymmetric routing, but this is not an issue in most cases.

6.2.1.2.1Configuration with GLBP

The configuration of the branch routers using GLBP follows.

The DMVPN configuration of the branch routers is almost the same as in the earlier case, except that each branch router runs just one tunnel over its ISP connection to the corresponding headend router. The DMVPN configuration of the branch routers is also almost identical. Unlike the previous design, the branch routers here can use the same NHRP network-ID because this ID has only local relevance, although it is recommend to use unique NHRP network-IDs to simplify problem management and troubleshooting.

The design consideration here is how GLBP is implemented to provide load-balancing and failover. The configuration shown here uses the GLBP host-dependent load-balancing method. This allows the same GLBP Forwarder to always be used for a specific host. GLBP also allows other load-balancing methods.

GLBP priorities can be configured for each branch router to control which router becomes the Active Virtual Gateway (AVG).

To implement failover, GLBP weighting can be adjusted by tracking a route to a network that is behind the headend routers and advertised to the branches over the tunnel. If there is any failure which causes the DMVPN tunnel at a branch router to fail, the tracking object will detect the failure and reduce the weight of that GLBP Forwarder (by default the weight is decreased by 10, which causes the weight to drop from 100 to 90, i.e. less than 95 which is the weight on the other branch router) allowing the other branch router to take over the role as the only ‘active’ GLBP Forwarder.

Spoke-1 Configuration

```
interface Tunnel0
 description DMVPN Endpoint
 ip address 192.168.1.201 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.1
 ip nhrp map 192.168.1.1 10.40.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.1.1
 ip nhrp registration timeout 75
 load-interval 30
 tunnel source Loopback0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
track 1 ip route 5.0.0.0 255.0.0.0 reachability ← Tracking object
!
interface GigabitEthernet0/0.3
 description Inside Intf
 encapsulation dot1Q 3
 ip address 3.0.0.101 255.0.0.0
```



```
 glbp 3 ip 3.0.0.1                          ← GLBP virtual IP
 glbp 3 priority 110                        ← Primary AVG
 glbp 3 preempt delay minimum 30
 glbp 3 weighting 100 lower 95 upper 98     ← Weight threshold values
 glbp 3 load-balancing host-dependent       ← Load-balancing method
 glbp 3 weighting track 1                   ← weight-track mapping
 glbp 3 forwarder preempt delay minimum 0
!
```

Spoke-2 Configuration

```
interface Tunnel0
 description DMVPN-ISP2
 ip address 192.168.2.201 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.2
 ip nhrp map 192.168.2.1 10.40.0.2
 ip nhrp network-id 2
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.2.1
 ip nhrp registration timeout 75
 load-interval 30
 tunnel source Loopback0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
track 1 ip route 5.0.0.0 255.0.0.0 reachability ← Tracking object
!
interface GigabitEthernet0/0.3
 description Inside Intf
 encapsulation dot1Q 3
 ip address 3.0.0.102 255.0.0.0
 !
 ! GLBP Configuration
 !
 glbp 3 ip 3.0.0.1                          ← GLBP virtual IP
 glbp 3 priority 105                        ← Secondary AVG
 glbp 3 preempt delay minimum 30
 glbp 3 weighting 100 lower 95 upper 98     ← Weight threshold value
 glbp 3 load-balancing host-dependent       ← Load-balancing method
 glbp 3 weighting track 1                   ← weight-track mapping
 glbp 3 forwarder preempt delay minimum 0
!
```

6.2.2 DMVPN as Backup

DMVPN can also be implemented in scenarios where it can be used as backup connection to the headend routers. The primary connection could be one or more backbone connections. The following designs show a couple of designs to implement DMVPN as a backup connection.

DMVPN networks can provide backup paths if a primary MPLS network fails. The customer typically runs Border Gateway Protocol (BGP) over the primary link and an Interior Gateway Protocol (IGP) over the DMVPN link. Common concerns and issues involve the subnet mask for routes learned over each path, and the administrative distance the routes need to enable the network to fail over and fail back.

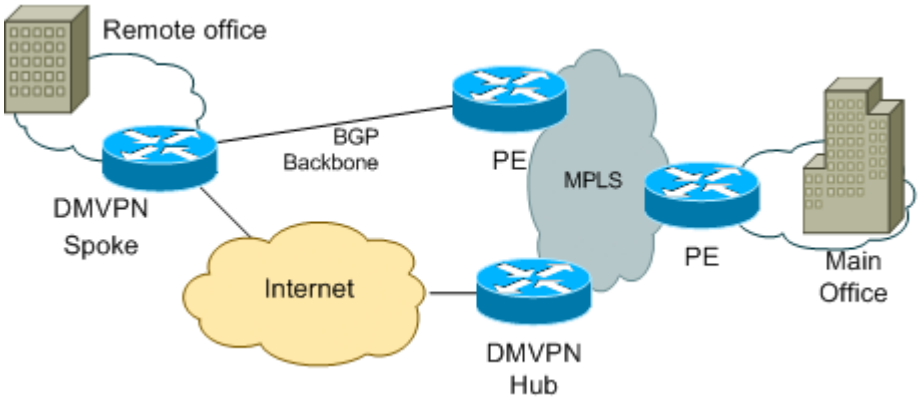
There are two commonly used network topologies. One router might connect to the primary and backup links, or two separate routers may be used, one to the primary MPLS network and one to the backup DMVPN network over the Internet. In the latter case, HSRP or a separate routing protocol might need to run between the two edge routers to provide network connectivity if one of the connections fails.

6.2.2.1Single Primary Connection

A single branch router can be used to provide the primary backbone connection as well as the back DMVPN connection. In this case, selection of the primary backbone path for all communication to the headend routers is done by this branch router itself.

For simplicity, Figure 6-3 shows the topology from the perspective of one branch router.

Figure 6-3. Single branch router with DMVPN backup



DMVPN configuration is straightforward, and involves configuring NHRP parameters, tunnel source, and mGRE.

The design consideration here is the interaction with the backbone link and routing protocols (EGP over the backbone and IGP over the DMVPN tunnel). The branch router learns the headend router routes through the backbone and then tunnel. Routing must be configured so that the routes learned over the backbone link are always preferred. In the following sample configuration, eBGP routes are preferred over external EIGRP.

```
!
interface Tunnel0
 description DMVPN Endpoint
 ip address 192.168.1.201 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.1
 ip nhrp map 192.168.1.1 10.40.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.1.1
 ip nhrp registration timeout 75
 tunnel source Serial0/0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
interface Serial0/0/0
 description Tunnel source
```

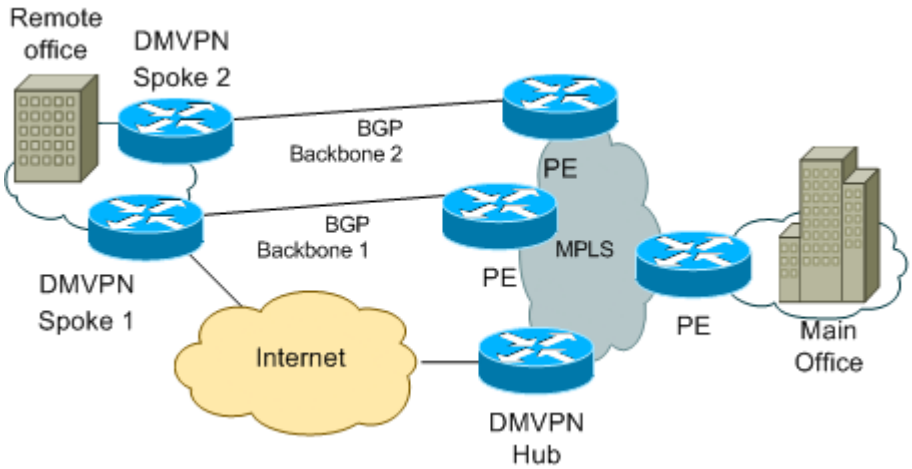


```
 ip address 41.0.0.1 255.0.0.0
!
interface Serial0/1/0
 description Backbone link
 ip address 31.0.0.1 255.0.0.0
!
router eigrp 10
 network 192.168.0.0
 connected route-map NoRedist        ← Redistribute as needed
 no auto-summary
!
router bgp 31
 no synchronization
 neighbor 31.0.0.123 remote-as 192
 bgp router-id 31.0.0.1
 bgp log-neighbor-changes
 connected route-map NoRedist        ← Redistribute as needed
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 41.0.0.254        ← default route to Internet
!
route-map NoRedist deny 10           ← Route-map prevents routing error
 match ip ConnNoRedist
!
route-map NoRedist permit 100
!
ip access-list standard ConnNoRedist
 permit 31.0.0.0 0.255.255.255
 permit 41.0.0.0 0.255.255.255
 deny any
!
```

6.2.2.2Multiple Primary Connections

To provide redundancy, load-balancing and failover, the branch office could have multiple backbone links, and then employ a DMVPN link as a last-resort backup link over the Internet.

Figure 6-4. Multiple Branch Routers with DMVPN Backup



For simplicity, Figure 6-4 shows the topology from the perspective of two branch routers.

The presence of multiple branch routers warrants some load-balancing or failover protocol at the branch office. In a lab environment, this design has been tested with both GLBP (Gateway Load Balancing Protocol) and HSRP (Hot Standby Router Protocol). GLBP provides the additional benefit of load balancing, while HSRP simply provides selection of one path. If load-balancing is not critical, it is easier to implement failover using HSRP.

Configuration with GLBP

The configuration of the branch routers using GLBP follows.

Only one of the branch routers is configured for DMVPN. The DMVPN configuration of this branch router is the same as in the earlier case. The routing configuration of the branch routers (especially Spoke-1) is almost the same as the previous design.

The design consideration here is the interaction of the multiple backbone links with the backup DMVPN link and how GLBP is implemented to provide load-balancing and failover. The configuration shown here uses the GLBP host-dependent load-balancing method. This allows the same GLBP Forwarder to always be used for a specific host. GLBP also allows other load-balancing methods.

GLBP priorities can be configured for each branch router to control which router becomes the Active Virtual Gateway (AVG).

Implementing failover along with load-balancing in this scenario gets a little tricky. Because the branch router has reachability to the main office networks, either over the backbone links or the backup DMVPN link, those networks cannot be used for tracking. In this topology, tracking, must be implemented so that if either backbone link is up, that link should be used. DMVPN should only be used as the last resort. To implement this, a combination of tracking objects is used that track whether either of the backbone links are up and accordingly adjust the weights of GLBP on the branch routers.

Spoke-1 Configuration

```
interface Tunnel0
 description DMVPN Endpoint
 ip address 192.168.1.201 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.40.0.1
 ip nhrp map 192.168.1.1 10.40.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.1.1
 ip nhrp registration timeout 75
 load-interval 30
 tunnel source Serial0/0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
track 2 ip route 31.0.0.0 255.0.0.0 reachability
track 3 ip route 10.32.0.0 255.255.0.0 reachability
track 21 list boolean or            ← Track if either is up
 object 2
 object 3
track 22 list boolean or            ← Track if only backbone1 is up
 object 2
 object 3 not
```



```
!
interface GigabitEthernet0/0.3
 description Inside Intf
 encapsulation dot1Q 3
 ip address 10.3.0.101 255.255.0.0
 glbp 3 ip 10.3.0.1
 glbp 3 priority 110
 glbp 3 preempt delay minimum 30
 glbp 3 weighting 100 lower 95 upper 98
 glbp 3 load-balancing host-dependent
 glbp 3 weighting track 21 decrement 1    ← If both down, dec by 1
 glbp 3 weighting track 22 decrement 6    ← Decrement below lower threshold
 glbp 3 forwarder preempt delay minimum 0
!
```

Spoke-2 Configuration

```
track 1 ip route 10.32.0.0 255.255.0.0 reachability
!
interface GigabitEthernet0/0.3
 description Ixia traffic; Inside Intf
 encapsulation dot1Q 3
 ip address 10.3.0.102 255.255.0.0
 glbp 3 ip 10.3.0.1
 glbp 3 priority 105
 glbp 3 preempt delay minimum 30
 glbp 3 weighting 100 lower 97 upper 98
 glbp 3 load-balancing host-dependent
 glbp 3 weighting track 1
 glbp 3 forwarder preempt delay minimum 0
!
```

6.2.3 NAT-T Aware DMVPN

A popular branch router setup is placing the branch router behind a NAT device. The service provider can conserve IP addresses, and dynamically or statically assign private addresses to the branch Internet-facing interface.

In DMVPN Phase 3, spokes behind NAT participate in dynamic direct spoke-to-spoke tunnels. The NAT box must translate the spoke to the same outside NAT IP address for the spoke-to-spoke connections as the NAT box does for the spoke-to-hub connection. If more than one DMVPN spoke is behind the same NAT box, the NAT box must translate the DMVPN spokes to different outside NAT IP addresses. Also, in this case it is possible that you may not be able to build a direct spoke-spoke tunnel between these spokes. If in any case a spoke-spoke tunnel fails to form, the spoke-to-spoke packets continue to be forwarded over the spoke-to-hub-to-spoke path.

With NAT-Transparency (NAT-T) Aware DMVPN, NHRP can learn and use the NAT public address for mappings as long as IPsec transport mode is used (this is the recommended IPsec mode for all DMVPN networks). The restriction that the private interface IP address of the spoke must be unique across the DMVPN network is removed. Although NAT-T (IKE and IPsec) can support two peers being translated to the same IP address (using UDP ports to differentiate them), this functionality is not supported for DMVPN.

Before Cisco IOS Release 12.4(6)T, DMVPN spokes behind NAT do not participate in dynamic direct spoke-to-spoke tunnels. Traffic to or from a spoke that is behind NAT is forwarded using the DMVPN hub routers. DMVPN spokes that are not behind NAT, and in the same DMVPN network, can create dynamic direct spoke-to-spoke tunnels between each other.

All DMVPN branch routers must have a unique IP address after they are NAT translated, but the branch routers can have the same IP address *before* they are NAT translated. In Cisco IOS Release 12.4(6)T or later releases, DMVPN spokes behind NAT can participate in dynamic direct spoke-to-spoke tunnels. The following restrictions still apply:

- The spokes must be behind NAT boxes that are performing NAT, not PAT.
- The NAT box must translate the spoke to the same outside NAT IP address for the spoke-spoke connections as the NAT box does for the spoke-hub connection.
- If there is more than one DMVPN spoke behind the same NAT box, then the NAT box must translate the DMVPN spokes to different outside NAT IP addresses.

You might not be able to build a direct spoke-to-spoke tunnel between these spokes. If a spoke-to-spoke tunnel fails to form, the spoke-to-spoke packets continue to be forwarded over the spoke-hub-spoke path.

Figure 6-5. DMVPN Spoke Behind NAT+ Firewall

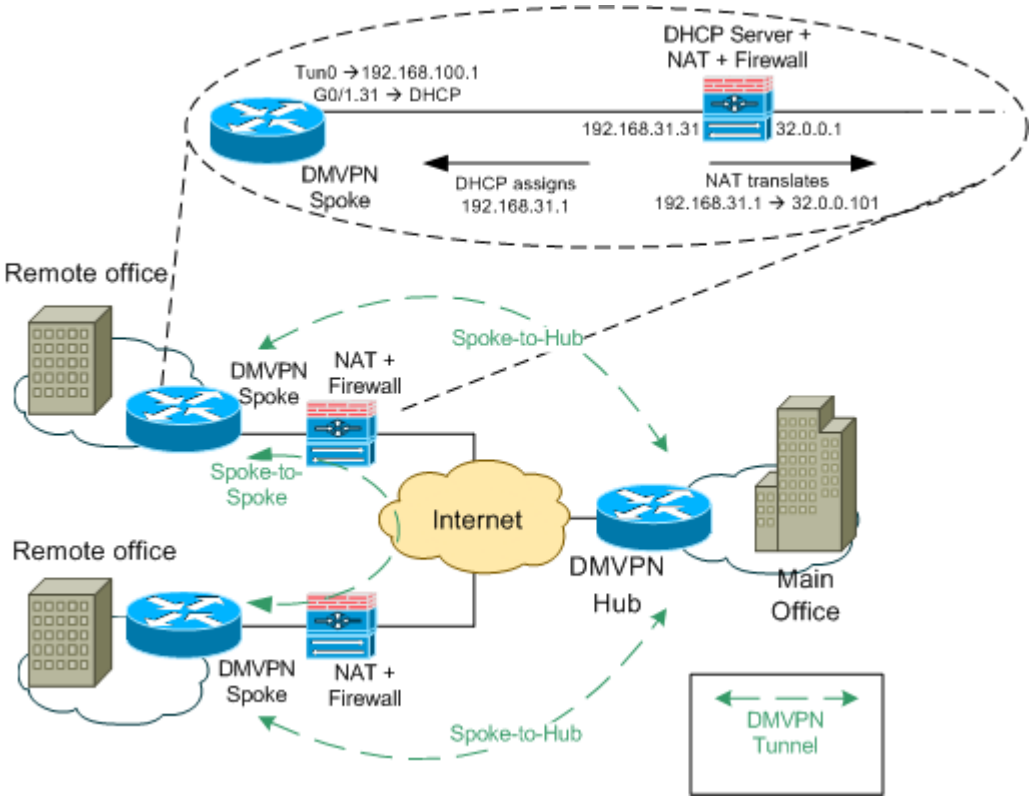


Figure 6-5 shows a simplified topology with two remote office branch routes, each connected to the Internet using a device that provides DHCP, NAT and Firewall services to the branch routers. The Internet-facing interface of the branch router is configured for DHCP, and is the source of the tunnel. The tunnel and crypto configuration of such a branch router is shown in the following sample configuration. NAT Traversal (NAT-T) is a feature where NAT og [CONTENT TBD] the IPsec peer addresses is autodetected by VPN devices.

There are no configuration steps for a router running Phase 3 DMVPN Compatible Cisco IOS Release (12.4(6)T or later). If both VPN devices are NAT-T capable, NAT-T is autodetected and autonegotiated. With NAT-T enabled, IPsec peers detect the presence of NAT during IKE Phase 1 negotiation with a series of NAT discovery (NAT-D) payloads. Once detected, IPsec packets are encapsulated in a UDP wrapper, which enables the ESP (IP protocol 50) packets to travel correctly through NAT devices.

In the following sample configuration, (IP addresses in the configuration correlate to Figure 6-5) of a DMVPN Phase 3 branch router placed behind a device that provides DHCP, NAT and Firewall services. Tests in a lab environment in a nonscaled topology with Cisco IOS Release 12.4(15)T4 show successful DMVPN tunnel formation between the spoke and hub, and successful dynamic spoke-to-spoke tunnels.

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key KEY address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 60
crypto ipsec transform-set DES esp-3des esp-sha-hmac
 mode transport                                  ← Use transport mode
crypto ipsec profile VPN1
 set transform-set DES
```



```
!
interface Tunnel0
 description DMVPN Endpoint
 ip address 192.168.100.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.32.0.4     ← 10.32.0.4 - Hub public IP
 ip nhrp map 192.168.100.4 10.32.0.4   ← 192.168.100.4 - Hub Tunnel IP
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.4
 ip nhrp registration timeout 75
 ip nhrp shortcut                    ← Shortcut and redirect allow
 ip nhrp redirect                      dynamic spoke-to-spoke tunnel
 load-interval 30
 tunnel source GigabitEthernet0/1.31
 tunnel mode gre multipoint
 tunnel protection ipsec profile VPN1
!
interface GigabitEthernet0/1.31
 description SpokeOutside-NatInside
 encapsulation dot1Q 31
 ip address dhcp
!
```

6.3 Branch Services

Additional integrated services can be configured on the branch router to help alleviate congestion, prioritizing traffic, and enhancing security. This section describes some of these recommended features.

6.3.1 QoS

If the GRE tunnel address for the spokes is static and known then they can be used to classify traffic per spoke for traffic shaping and the IP TOS field can be used to classify traffic within this shaping for policing. The quality of service (QOS) classification is statically defined per spoke and applied on the physical interface.

Alternatively, if the data networks behind the spoke are known then that can be used to classify unicast-traffic that is destined for that spoke. This classification can be used in shaping on the outbound physical interface and a ”child” policy can be used to police traffic within this shaping. This will not be able to classify any multicast traffic per spoke since all multicast traffic would have the same source and destination IP address no matter which spoke it was destined for.

ISAKMP and qos-groups, which is based on IKE identity (IP address or hostname), can also be used to mark hub to spoke the traffic. Though there are only 256 possible QoS-groups so if there are more than this number of spokes some spokes will have to share the same QoS group. In this case, all traffic to spokes using the same QoS group would be classified together as an aggregate policy.

This section describes the QoS considerations while configuring the branch router for DMVPN deployment.

6.3.1.1Core-Facing QoS

For a branch router, all headend-bound traffic is usually restricted to one exit point, that is, a tunnel interface that has the headend router as the IP next-hop, which in turn is bound to one physical interface facing the Internet. A hierarchical traffic shaper might be required to ensure that certain types of traffic from the branch router have preference over other traffic exiting the Internet-facing interface. A hierarchical shaper employs two service policies: a parent policy to apply a QoS policy (shaping) to total traffic, and a child policy to apply a QoS policy (policing) to a flow or subset of the total traffic. Logical interfaces, such as subinterfaces and tunnel interfaces require a hierarchical policy with the traffic-limiting (shaping) feature at the parent level and queuing at lower levels.

The following sample configuration gives priority to certain types of traffic, while limiting total traffic from the branch router. This configuration affects all outbound traffic from the tunnel source interface.

```
class-map match-any VoiceClass              ← Define Voice class
 match  dscp cs5  cs6  cs7
class-map match-all DataClass               ← Define Data class
 match  dscp cs4
!
policy-map Q-policy
 class VoiceClass
  priority 128                     ← Use LLQ for Voice class
 class DataClass
  bandwidth 100                    ← Use CBWFQ for non-latency sensitive traffic
  queue-limit 16
 class class-default
  fair-queue                       ← Use WFQ for non-priority traffic
  queue-limit 6
policy-map ShaperPolicy
 class class-default               ← Shape all traffic
  shape average 256000
  service-policy Q-policy
!
interface Serial0/0/0              ← Apply Shaper on tunnel source
 service-policy output ShaperPolicy
!
```

6.3.1.2Core-Facing QoS with Split Tunneling

The preceding example assumes that all traffic out of the physical interface (tunnel source) is bound for the hub. The spoke can also be used in a split tunneling scenario, in which case there will be additional traffic from the spoke internal networks to the Internet. Split-tunneled traffic is unencrypted traffic out of the tunnel source interface.

It is possible to configure the hierarchical shaper so that all traffic out the physical interface is shaped, giving only best-effort queuing to split-tunneled traffic and protecting the tunneled traffic.

The following sample configuration protects tunneled traffic while providing best-effort scheduling for Internet bound traffic. Tunneled encrypted traffic is identified and then further categorized as high priority traffic and low priority traffic. High priority traffic could be voice traffic, and low priority traffic could be all other encrypted traffic. All other nonencrypted traffic is caught by **`class-default`**.

```
class-map match-all EncryptedClass     ← Match all encrypted traffic
 match protocol ipsec
 match access-group 1
```



```
!
class-map match-all HighPriorityClass  ← Define traffic for LLQ
 match dscp cs5
 match class EncryptedClass
!
class-map match-all LowPriorityClass   ← Define traffic for CBWFQ
 match not dscp cs5
 match class EncryptedHost
!
policy-map Q-policy              ← Define Queuing policy
 class HighPriorityClass
  priority 128
 class LowPriorityClass
  bandwidth 256
  queue-limit 16
 class class-default             ← All non-encrypted traffic gets best-effort
  fair-queue
  queue-limit 6
!
policy-map ShaperPolicy          ← Define Shaper
 class class-default
  shape average 512000
  service-policy Q-policy
!
interface g0/1.30                ← Apply Shaper on core-facing intf
 service-policy output ShaperPolicy
!
access-list 1 permit 10.30.0.1      ← Match on tunnel source intf
!
```

In rare cases, only the encrypted traffic might require shaping and Internet bound traffic can bypass the shaper. Because all tunneled traffic is encrypted, a separate class can be configured to match only encrypted traffic, and the shaper can be applied to all traffic matching that class.

The following configuration can be added to the configuration in Section 6.3.1.1 to shape only tunneled traffic.

```
class-map match-any Encrypted
 match protocol ipsec
 match access-group name isakmp
!
policy-map ShaperPolicy
 class Encrypted                    ← Shape only encrypted traffic
  shape average 256000
  service-policy Q-policy
!
ip access-list extended isakmp
 permit udp any any eq isakmp
 permit udp any any eq non500-isakmp
!
```

For the hierarchical shaper to be applied, interesting traffic must be classified in order to give it any priority. The **qos pre-classify** command enables QoS for VPNs, which enables classifying packets on the output physical interface (tunnel source) before the data is tunneled and encrypted. In all of the preceding sample configurations, traffic is classified by differentiated services code point (DSCP) bits. Because the TOS field is copied to the outer packet during encapsulation, **qos pre-classify** is *not* needed. However, if classification must be done on other fields that will be encrypted, packets must be classified with information before they are tunneled and encrypted; this requires using **qos pre-classify.**

Note: Due to a known issue (CSCsg83151), pre-classify might not work with the shaping mechanism. This bug appears to be fixed in 12.4(11)T3.

6.3.1.3Internal QoS

As required, a hierarchical shaper can also be added to the internal facing interface of the branch router in order to provide certain guaranteed bandwidth to interesting traffic. A typical usage would be to use low latency queuing (LLQ) for voice traffic carried over the tunnel and is bound to hosts at the branch. LLQ provides for the low latency and guaranteed bandwidth needed for the voice traffic, but also applies a ceiling to the bandwidth. Hence, the bandwidth pipe for the LLQ should be chosen carefully.

Internal testing in a lab showed that the voice traffic could maintain an average jitter of 10–11 ms. Testing was performed on a 2800 branch router with IOS 12.4(15)T4, using the following sample configuration and with the shaper oversubscribed. In the sample configuration, all traffic bound to the internal-facing spoke interface comes from the tunnel.

This sort of shaping scheme mainly works with TCP based traffic. The idea is that any traffic that exceeds the shaped queue size is dropped, and the hope is that the application or protocol will reduce the send rate.

```
!
class-map match-any VoiceClass              ← Define Voice class
 match  dscp cs5  cs6  cs7
class-map match-all DataClass               ← Define Data class
 match  dscp cs4
!
policy-map SpokeQ-Policy
 class VoiceClass
  priority 512                     ← Use LLQ for Voice class
 class DataClass
  bandwidth 256                    ← Use CBWFQ for non-latency sensitive traffic
  queue-limit 16
 class class-default
  fair-queue                       ← Use WFQ for non-priority traffic
  queue-limit 6
policy-map SpokeShaperPolicy
 class class-default               ← Shape all traffic
  shape average 1000000
  service-policy SpokeQ-Policy
!
interface GigabitEthernet0/0.3              ← Apply Shaper on interface intf
 service-policy output SpokeShaperPolicy
!
```

6.3.2 VRFs

VPN routing and forwarding (VRF) provides multiple independent contexts (addressing, routing, and interfaces) at the branch location to separate departments, subsidiaries, or customers. All contexts can share one uplink connection to the core while still maintaining secure separation between them.

Virtualization of traffic within the DMVPN spoke is sometimes necessary. The spoke could be configured with an fVRF, iVRF, or both. An fVRF is used primarily to carve out a separate Internet routing table, where the tunnel interface exists, from the global routing table, where the private traffic going over the tunnel exists. Along similar lines, the advantage of using an iVRF is to define a private space to hold the

DMVPN and private network information. Both configurations separate internet and private routing information, to provide extra security from router attacks from the Internet.

These VRF configurations can be used on both DMVPN hubs and spokes. This can also be used to terminate multiple DMVPN networks on the same router, keeping each DMVPN network (private routing table and packet forwarding) separate. This feature could be used to enable two different default routes to be used on the router to simultaneously support dynamic spoke-to-spoke tunnels and non-split tunneling. Another possible use is to enable the service provider (SP) to access the customer edge (CE) router for management purposes, but not have access to customer data.

The IPsec tunnel can be associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, called front-door VRF (fVRF), while the inner, protected IP packet belongs to another domain called theinside VRF (iVRF). In other words, the local endpoint of the IPsec tunnel belongs to the fVRF, while the source and destination addresses of the inside packet belong to the iVRF.

The advantage of using an fVRF is primarily to carve out a separate routing table from the global routing table (where tunnel interface exists). The advantage of using an iVRF is to define a private space to hold the DMVPN and private network information. These VRF configurations can be used on both DMVPN hub and spoke. Also, the DMVPN branch router can be configured either with fVRF or iVRF or both. Having either of these configurations provides extra protection from anyone trying to attack the router from the Internet by separating out routing information.

In case of fVRF, the tunnel destination lookup needs to be done in the fVRF. Secondly, since the Internet-facing interface is in an fVRF, the ISAKMP key lookup is also done in the fVRF. As for using iVRF, the tunnel, private subnets, and routing protocol must be defined in the iVRF space. The tunnel destination and ISAKMP key are looked up in global space for this scenario.

The following configuration uses two separate VRF instances to isolate the Internet-facing routing table from the internal network-facing routing table. Using VRFs in this way, called VRF-lite, is one router having multiple VRF routing instances, without having to use MPLS. With VRF-lite, multiple VRFs can be supported on the router, giving it the ability to maintain separate VRF routing tables to extend the privacy and security of a VPN to the branch office. VRF separation is local to just this router.

Here is the sample configuration of a branch router with both fVRF and iVRF configured in separate VRF instances. The design considerations here include:

- Define the iVRF and fVRF routing instances
- Identifying the branch router inside interface and configuring it for iVRF
- Identifying the branch router core-facing interface and configuring it for fVRF
- Defining the ISAKMP key in the fVRF
- Adding the tunnel interface to the iVRF
- Mapping tunnel destination lookup to fVRF
- Configuring VRF-aware IGP to run over the tunnel
- Adding appropriate VRF specific routes

```
ip vrf vpn-int
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn-ext
 rd 101:1
```



```
 route-target export 101:1
 route-target import 101:1
!
interface GigabitEthernet0/0.3
 description Inside Intf with iVRF
 encapsulation dot1Q 3
 ip vrf forwarding vpn-int
 ip address 10.3.0.101 255.255.0.0
!
interface GigabitEthernet0/0.41
 description To Internet; Tunnel source and fVRF
 encapsulation dot1Q 41
 ip vrf forwarding vpn-ext
 ip address dhcp
!
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
 group 2
crypto keyring vpn-ext vrf vpn-ext
 pre-shared-key address 0.0.0.0 0.0.0.0 key TEST
crypto isakmp keepalive 60
crypto ipsec transform-set DES esp-3des esp-sha-hmac
 mode transport
crypto ipsec profile VPN1
 set transform-set DES
!
interface tunnel 0
 description DMVPN Endpoint
 ip vrf forwarding vpn-int
 ip address 172.16.1.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast 10.42.0.1
 ip nhrp map 172.16.1.254 10.42.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
 ip nhrp nhs 172.16.1.254
 ip nhrp registration timeout 75
 tunnel source gi0/0.41
 tunnel mode gre multipoint
 tunnel vrf vpn-ext
 tunnel protection ipsec profile VPN1
!
router eigrp 10
 !
 address-family ipv4 vrf vpn-int
  network 10.3.0.0 0.0.255.255
  network 172.16.0.0
  no auto-summary
  autonomous-system 10
  exit-address-family
!
! Add default route to the fVRF for forwarding tunnel packets
ip route vrf vpn-ext 0.0.0.0 0.0.0.0 GigabitEthernet0/0.41 dhcp
!
```

Note: The preceding could also be configured with only iVRF or only fVRF (not both), with the global routing table being the second routing table, to produce similar results. The preceding configuration shows

how to do both styles in one configuration. You can use only one in which case you would omit the configuration you do not want, and configure the global routing table to do that part instead.

6.3.2.1VRFs with Split Tunneling:

Configuring VRF-aware DMVPN and split tunneling can be tricky. If using VRF-lite, iVRF and fVRF would be in the same VRF instance, so using the same routing table and routing configuration to provide Internet access to internal networks behind the branch router is straightforward. However, if fVRF and iVRF are in separate VRF instances, as in the preceding configuration, the fVRF handles traffic from the Internet to the branch router. If this traffic is bound to any hosts behind the branch router, appropriate holes must be inserted into the fVRF routing table to facilitate that traffic. Take care while adding these routes in the routing table, as they could lessen the VRF-introduced security. It is recommended to add security (such as IOS Firewall and IPS) to protect the branch router and the network behind the branch router.

The additional configuration for the preceding configuration supports split-tunneled traffic:

```
! Add route to iVRF to reach Internet***
ip route vrf vpn-int 0.0.0.0 0.0.0.0 GigabitEthernet0/0.41 dhcp
!
! Add route to fVRF for internet traffic to reach the inside network
ip route vrf vpn-ext 10.3.0.0 255.255.0.0 GigabitEthernet0/0.3 10.3.0.1
!
```

Note: The preceding method of configuring a static route to the fVRF for the internal host works for only one host behind the spoke. Another way to insert the route for hosts behind the spoke is to use **`ip vrf receive`**. This command allows the IP addresses that are associated with an interface to be inserted as a connected route into a particular VRF. Once the IP addresses are inserted as a connected route, the interface is allowed to respond to requests (such as a ping request) directed to it from a VPN. See the MPLS command reference guide for details and examples on how to configure 'ip vrf receive'.

6.3.3 NAT

The branch router can also be configured as a NAT/PAT device for hosts in the network behind the branch router. If using VRFs, the NAT source list should be mapped to the iVRF.

The following sample configuration does a many-to-one NAT using overload.

```
!** NAT config using NVI **
!
interface GigabitEthernet0/0.3
 description Inside Intf
 ip nat enable
!
interface GigabitEthernet0/0.41
 description To Internet; Tunnel source
 ip nat enable
!
ip nat source list 50 interface GigabitEthernet0/0.41 vrf vpn-int overload
!
access-list 50 permit 10.13.1.0 0.0.0.255
!
```


6.3.4 Security

Additional features can be added to the branch router to increase its security and resilience. This DIG describes IOS Firewall and IOS IPS. These features are recommended on branch routers that connect to the main office over a public network.

In the lab, we tested a branch router running IOS 12.4(15)T4 software, using a basic branch-router DMVPN topology including split-tunneling. The branch router was configured using DMVPN, hierarchical shaping (QoS), IOS Firewall and IPS with 5.x signature format. Test results showed minimal impact on CPU, and all features operated as expected.

6.3.4.1Split Tunneling

Split tunneling enables a customer to have separate pathways: one to the corporate network, over which traffic is encrypted, and one to the Internet, over which traffic is not encrypted. This has the advantage of not forcing all of the customer's traffic to go to the corporate network before being forwarded to the Internet. This reduces traffic load on the DMVPN tunnel by default routing toward the ISP. All "internal" traffic would follow the routes learned via the DMVPN tunnel. However, this approach increases the number of entrance/exit points from the customer network where a firewall, NAT, and IPS might be needed to protect the network.

A nonsplit-tunneling setup might be desired to control access to the Internet at a single point, at the cost of extra traffic over the DMVPN tunnels (spoke to hub) and extra processing on the headend router. Typically this would require a static route pointing to the hub for the mGRE tunnel to form, and a default route directing all traffic over the DMVPN tunnel. The fVRF/iVRF configurations can be useful on the spokes when doing nonsplit tunneling and advertising a default route over the tunnel from the hub in addition to the default route from the ISP for build spoke-spoke dynamic tunnels.

6.3.4.2IOS Firewall

Cisco IOS Firewall is a stateful security software component of Cisco IOS Software. Firewall integration in IOS routers augments inherent router capabilities: multitopology interfaces, routing protocols, and a broad range of services, as along with an expanding group of other security features, such as VPN and intrusion prevention system (IPS) features. IOS Firewall operates with other Cisco IOS Software technologies, including NAT, quality of service (QoS), and IP Security (IPsec), to become a vital component of an end-to-end network security infrastructure.

The IOS Firewall comprises:

- Stateful packet inspection (SPI) that provides a granular firewall engine
- Authentication Proxy that provides per-host access control
- Application inspection features that add protocol conformance checking and network use policy control

SPI uses access control lists (ACLs) to monitor traffic. SPI monitors connections from a secured network to an unsecured network, and anticipates traffic returning to the secure host from the unsecured network. SPI maintains a session table listing all active sessions of the firewall.

SPI configuration tasks include:

- Identify traffic to be allowed out through the firewall. In the following sample configuration, it is assumed that the only traffic to be allowed through the firewall is IPsec, ISAKMP (encrypted traffic), and DHCP.
- Configure ACLs to block traffic from the unsecured network. Be sure ACLs permit legitimate traffic from the secure network to the unsecured network. In the following sample configuration, all IPsec and DHCP related traffic is allowed, while all other IP traffic is blocked.
- Create inspection rules. Apply the rules inbound to the secure-side interface, or outbound to the unsecured-side interface

IOS Firewall Configuration

```
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
! allow IPSEC, ISAKMP and DHCP packets through
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 100 permit udp any any eq bootpc
access-list 100 permit udp any any eq bootps
access-list 100 deny ip any any
!
! configure the firewall
interface g0/0.41
 ip access-group 100 in
 ip inspect IOSFW1 in
!
```

6.3.4.3IOS IPS

Cisco IOS IPS acts as an in-line intrusion prevention sensor, watching packets and sessions as they flow through the router and scanning each packet to match any IPS signatures. When IPS detects suspicious activity, IPS responds before network security can be compromised and logs the event using IOS syslog messages or Security Device Event Exchange (SDEE). When packets in a session match a signature, IPS can take various actions, such as sending an alarm to a syslog server, dropping the packet, resetting the connection, denying traffic from attacker for a specific time, and so on.

Here is a simplified IPS configuration that explains some IPS configuration tasks:

- RSA crypto key and public signature must be loaded on the router for signature decryption. The public key configuration can be accessed at the following URL: http://www.cisco.com/pcgi-bin/tablebuild.pl/ios-v5sigup.
- Router memory and resource constraints prevent a router from loading all IPS signatures. Thus, load only a selected set of signatures that are defined by the categories. IPS does not scan retired signatures, so they do not fire alarms. In the following configuration, all signatures are disabled first, and then only the basic signature set is enabled.
- The signature information must then be saved at a preconfigured location on the router. The configuration location is used to restore the IPS configuration if the router reboots, or IPS is disabled or reenabled.
- Apply the IPS rule at the interface. This automatically loads the signatures and builds the signature engines. Depending on the platform and how many signatures are being loaded, building the engine can take up to several minutes. It is recommended to enable logging messages to monitor the engine building status.

IPS Configuration

```
crypto key pubkey-chain rsa
named-key realm-cisco.pub signature
 key-string
  <key-string here>
quit
exit
ip ips name IPS
ip ips config location flash:ipsstore
ip ips signature-category
 category all
  retired true
  exit
 category ios ips basic
  retired false
  exit
 exit

interface s0/0/0
 ip ips IPS in
!
```

6.4 Best Practices and Limitations

This section describes branch router best practices and limitations.

6.4.1 Best Practices

- Use mGRE and IPsec in Transport mode.
- Configure Triple DES (3DES) or AES for encryption of transported data (exports of encryption algorithms to certain countries may be prohibited by law).
- Implement Dead Peer Detection (DPD) to detect loss of communication between peers.
- Keep IPsec packet fragmentation to a minimum on the customer network by setting MTU size and **`ip tcp adjust-mss`** on the tunnel and using Path MTU Discovery (PMTUD).
- Use Digital Certificates/Public Key Infrastructure (PKI) for scalable tunnel authentication.
- Configure a routing protocol (IGP) with route summarization for dynamic routing.
- Set up QoS service policies as appropriate on branch router interfaces to help alleviate interface congestion issues and to attempt to keep higher priority traffic from drops.
- Configure two tunnels to alternate headends, using routing metrics to designate a primary and secondary path, to provide redundancy and failover. If spoke-to-spoke tunnels are used, this deployment could result in complications when errors occur. Careful considerations must be made if this type of redundancy is required.
- Configure only **`ip nhrp shortcut`** on branch routers
- Only if layer-4 matching is needed on packets, configure **`qos pre-classify`** (but *not* with hierarchical shaping) in VPN designs where QoS and IPsec occur on the same system.
- Consider implementing other integrated services such as IOS Firewall, IPS, and NAT/ Port Address Translation (PAT).

- When scaling traffic and configuring multiple integrated services on the branch router, keep CPU utilization under 65%.

6.4.2 Limitations

The following are branch router hub-spoke deployment limitations:

- Branches must always initiate the DMVPN tunnel to the headend router; the headend cannot initiate the tunnel to the branch router.
- **`qos pre-classify`** might not work with hierarchical shaping on the branch router. If so, packets must be classified before they are tunneled. This could be done at a downstream device, or at the incoming internal-facing branch router interface.

Appendix A Sample Configurations

This appendix lists sample DMVPN configurations.

A.1 Hierarchical Hub Configurations

A.1.1 7200 Regional Hub Configuration

```
version 12.4
!
hostname 7200-HH-1
!
ip cef
!
ip domain-name cisco.com
!
crypto pki trustpoint DMVPN-CA
 enrollment url http://10.24.1.130:80
 ip-address 10.73.1.2
 revocation-check none
 rsakeypair VPN1
!
crypto pki certificate chain DMVPN-CA
 certificate 361D
 <certificate removed>
 certificate ca 01
 <certificate removed>
quit
!
crypto isakmp policy 10
 encr 3des
 group 2
!
! The isakmp keepalive is set to be 5 seconds greater than the spoke tunnel
   routing protocol hello time. The router will retry 5 times before declaring
   the crypto session down. This works out to slightly longer than the routing
   protocols dead timer.
crypto isakmp keepalive 35 11
!
crypto isakmp profile hub
   match identity user domain cisco.com
   match identity address 0.0.0.0
!
crypto ipsec transform-set gre_set esp-3des esp-sha-hmac
 mode transport
!
crypto ipsec profile dmvpnhub
 set transform-set gre_set
!
! call admission limit keeps the router from getting hammered by crypto setup
   during a period of convergence.
!
call admission limit 90
!
```

```
controller ISA 1/1
!
! Buffer tuning is very important if the hub is going to support any number of
   routing protocol peers
!
buffers small permanent 904
buffers small max-free 1293
buffers small min-free 271
buffers middle permanent 3793
buffers middle max-free 5424
buffers middle min-free 1138
buffers big permanent 1132
buffers big max-free 1619
buffers big min-free 339
buffers verybig permanent 210
buffers verybig max-free 301
buffers verybig min-free 63
buffers large permanent 70
buffers large max-free 100
buffers large min-free 21
buffers huge permanent 26
buffers huge max-free 37
buffers huge min-free 7
!
! The Tunnel1 interface is used to terminate connections from the Branch
   routers.  Advertising a RIP summary to all spokes.
!
interface Tunnel1
 bandwidth 1000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip tcp adjust-mss 1360
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp registration timeout 30
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 ip summary-address rip 10.81.0.0 255.255.0.0
 load-interval 30
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
 hold-queue 4096 in
 hold-queue 4096 out
!
!
! The Tunnel2 interface connects to a more central core hub.  The NHRP network
   ID is the same so we can send redirects to a spoke if the data is destined
   for a spoke connected to another hub in the hierarchy.  Basically the core
   hub would be a hub for many regional hubs as was discussed in chapter 3.
!
interface Tunnel2
 bandwidth 1000
 ip address 10.181.1.2 255.255.255.0
 no ip redirects
```



```
 ip mtu 1400
 ip tcp adjust-mss 1360
 ip nhrp authentication nsite
 ip nhrp map multicast 192.168.1.1
 ip nhrp map 10.181.1.1 192.168.1.1
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp nhs 10.181.1.1
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
!
interface GigabitEthernet0/1
 description ***Public interface***
 ip address 10.73.1.2 255.255.255.0
 load-interval 30
 duplex auto
 speed auto
 media-type sfp
 no negotiation auto
 hold-queue 4096 in
!
interface FastEthernet0/2
 ip address 10.24.155.20 255.255.0.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 description ***Private LAN***
 ip address 192.168.1.2 255.255.255.0
 load-interval 30
 duplex auto
 speed auto
 media-type sfp
 no negotiation auto
!
! Using EIGRP to redistribute the routes from the branches to other regional and
   core hub routers.  The protocol used for the core routing can be used out to
   the branch routers, but you would still want to summarize the routes to your
   spokes.  Any IGP could be used here for the core routing, the key is to make
   sure the routes from the spokes are advertised to all hubs to obtain full
   routing knowledge.
!
router eigrp 10
 redistribute rip
 network 10.181.1.0 0.0.0.255
 default-metric 1000 100 255 1 1400
 no auto-summary
!
! Using RIP to peer with branch routers.  EIGRP could be used here in place of
   RIP, as long as you advertise a summary route to the spokes.
!
router rip
 version 2
 network 10.80.0.0 0.3.255.255
 no auto-summary
!
line con 0
 exec-timeout 0 0
```

```
 password lab
 login
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
end
```

A.1.2 7200 Core Hub Configuration

```
version 12.4
!
hostname 7200-HH-CORE
!
ip cef
!
! The Tunnel0 interface is used to terminate connections from the Regional Hub
   routers.
!
interface Tunnel0
 bandwidth 1000
 ip address 10.181.1.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 101
 ip nhrp holdtime 600
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp registration timeout 30
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
!
interface GigabitEthernet0/1
description ***Private LAN***
 ip address 192.168.1.1 255.255.255.0
 load-interval 30
 duplex auto
 speed auto
 media-type sfp
 no negotiation auto
!
! Using EIGRP to advertise the routes between regional hub routers.  Any IGP can
   be used in the core network, as long as all hubs have full routing knowledge.
!
router eigrp 10
 network 10.181.1.0 0.0.0.255
 no auto-summary
!
line con 0
 exec-timeout 0 0
 password lab
```



```
 login
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
end
```

A.2 IOS SLB Configurations Using BVI Interfaces on Headend

A.2.1 6500 SLB Configuration

```
version 12.2
!
hostname 6500-SLB-1
!
boot system sup-bootflash:c6k222-jk9sv-mz.122-14.SY
!
ip domain-name cisco.com
!
! The SLB device could lose connection to a hub during a spanning tree
   convergence or similar event. A ping probe failure is a traumatic event
   because all of the EIGRP and crypto sessions to that hub will be reset. Set
   the failure timeout to a value that exceeds the spanning tree convergence
   time on VLAN20

ip slb probe PING-PROBE ping
 faildetect 11
!
ip slb serverfarm 7206-FARM
 predictor leastconns
 failaction purge
 probe PING-PROBE
 !
 real 10.71.100.1
  weight 1
  inservice
 !
 real 10.71.100.2
  weight 1
  inservice
!
! The sticky command causes any ISAKMP rekey event to be forwarded to the router
   that terminates the corresponding crypto session. The replicate command
   causes the SLB state to be copied to the other SLB device. The inservice
   command tracks the state of the standby group 1 and will relinquish the SLB
   active status if this router is no longer the HSRP primary for group 1.
!
ip slb vserver ESP
 virtual 10.71.101.1 esp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 replicate casa 10.71.1.2 10.71.1.3 60001 password NSITE
```



```
 inservice standby standby1
!
ip slb vserver ISAKMP
 virtual 10.71.101.1 udp isakmp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 inservice standby standby1
!
mls flow ip destination
!
spanning-tree vlan 20 priority 16384
spanning-tree vlan 20 hello-time 1
spanning-tree vlan 20 forward-time 5
spanning-tree vlan 20 max-age 8
! The connection to the hubs is via a VLAN. The spanning tree parameters are
   adjusted to speed convergence and to assure that the SLB device that is
   normally the active HSRP device is the root of the VLAN20 spanning tree.
!
redundancy
 mode rpr-plus
 main-cpu
  auto-sync running-config
  auto-sync standard
!
vlan 20
!
! This is the interface that connects to the Internet. The standby tracks the
   status of VLAN20 since that is the VLAN that connects to the hubs. VLAN20 has
   redundancy built into it so we only are concerned if the entire VLAN goes
   down.
!
interface GigabitEthernet1/1
 description to CAT6500-AGG-3:g2/1
 ip address 10.71.1.2 255.255.255.0
 no ip redirects
 standby 1 ip 10.71.1.250
 standby 1 timers 1 3
 standby 1 priority 10
 standby 1 preempt
 standby 1 name standby1
 standby 1 track Vlan20
 spanning-tree portfast
!
! These two interfaces connect to the hub routers
!
interface GigabitEthernet2/1
 no ip address
 switchport
 switchport access vlan 20
 switchport mode access
!
interface GigabitEthernet2/2
 no ip address
 switchport
 switchport access vlan 20
 switchport mode access
!
! This is a trunk that provides redundancy for the VLAN20 forwarding path.
!
interface GigabitEthernet2/8
```



```
 no ip address
 switchport
 switchport trunk encapsulation isl
 switchport trunk allowed vlan 20
 switchport mode trunk
!
! This is the VLAN that connects to the hubs. It tracks the state of the
   connection to the Internet.
!
interface Vlan20
 description ****Inside HSRP****
 ip address 10.71.100.251 255.255.255.0
 standby 100 ip 10.71.100.250
 standby 100 timers 1 3
 standby 100 priority 10
 standby 100 preempt
 standby 100 name standby100
 standby 100 track GigabitEthernet1/1
!
router ospf 10
 log-adjacency-changes
 network 10.71.1.0 0.0.0.255 area 10
!
ip classless
!
line con 0
 exec-timeout 0 0
line vty 0 4
 login
 transport input telnet ssh
!
end
```

A.2.2 7200 SLB Configuration with BGP Route Reflection

```
version 12.2
!
hostname 7200-SLB-1
!
boot-start-marker
! Need to use 12.2 image for SLB functionality
boot system disk0:c7200-js-mz.122-31.SB8
boot-end-marker
!
no logging console
enable password lab
!
no aaa new-model
ip subnet-zero
!
! Set the fail detect to a relatively high value. The connection between the
   hubs and the SLB devices is a layer 2 connection. Spanning tree needs time to
   converge before the SLB declares a hub down. This is a traumatic event which
   causes crypto and routing protocol sessions to be reset.
ip cef
ip slb probe PING-PROBE ping
 faildetect 11
!
ip slb serverfarm 7206-FARM
 predictor leastconns
```



```
 failaction purge
 probe PING-PROBE
 !
 real 10.74.100.3
  weight 1
  inservice
 !
 real 10.74.100.4
  weight 1
  inservice
!
! The replicate casa command provides stateful SLB failover. The ip address
   order is reversed on the peer router. Whether the SLB is active or standby
   depends on the HSRP state on GIG2/0
!
ip slb vserver ESP
 virtual 10.74.101.1 esp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 replicate casa 10.74.1.1 10.74.1.2 60001 password NSITE
 inservice standby standby1
!
ip slb vserver ISAKMP
 virtual 10.74.101.1 udp isakmp
 serverfarm 7206-FARM
 sticky 3650 group 1
 idle 3660
 replicate casa 10.74.1.1 10.74.1.2 60000 password NSITE
 inservice standby standby1
!
! Tunnel2 is used to reflect BGP routes from the SLB device to the hubs. You
   must define the nhrp map for all peers.
!
interface Tunnel2
 ip address 10.90.0.1 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip tcp adjust-mss 1360
 ip nhrp authentication nsite
 ip nhrp map 10.90.0.1 10.74.100.1
 ip nhrp map 10.90.0.2 10.74.100.2
 ip nhrp map 10.90.0.3 10.74.100.3
 ip nhrp map 10.90.0.4 10.74.100.4
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp max-send 65535 every 10
 load-interval 30
 tunnel source 10.74.100.1
 tunnel mode gre multipoint
 no clns route-cache
 hold-queue 4096 in
 hold-queue 4096 out
!
! The two HSRP interfaces track each other so if one goes down the other
   interface will relinquish its active status. Set the preempt after reload
   delay to be 90 seconds to allow the router to become fully functional before
   it takes over as HSRP primary.
!
interface GigabitEthernet0/0
 description to CAT2900-SLB:0/49
```



```
 ip address 10.74.100.1 255.255.255.0
 no ip redirects
 media-type gbic
 speed 1000
 duplex full
 negotiation auto
 standby 100 ip 10.74.100.254
 standby 100 timers 1 3
 standby 100 priority 10
 standby 100 preempt delay reload 90
 standby 100 name standby100
 standby 100 track GigabitEthernet2/0
 no clns route-cache
!
interface GigabitEthernet2/0
 ip address 10.74.1.1 255.255.255.0
 no ip redirects
 negotiation auto
 standby 1 ip 10.74.1.254
 standby 1 timers 1 3
 standby 1 priority 10
 standby 1 preempt delay reload 90
 standby 1 name standby1
 standby 1 track GigabitEthernet0/0
 no clns route-cache
!
router ospf 10
 log-adjacency-changes
 network 10.74.1.0 0.0.0.255 area 10
!
! This router acts as a route reflector for the hub routers. The weight is set
   so that if there are other super clusters of hubs the routes learned from the
   local hubs will be placed in the BGP table.  This design of route reflection
   is referred to as the "folded" design in this document.
!
router bgp 5000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.90.0.3 remote-as 5000
 neighbor 10.90.0.3 route-reflector-client
 neighbor 10.90.0.3 weight 65535
 neighbor 10.90.0.4 remote-as 5000
 neighbor 10.90.0.4 route-reflector-client
 neighbor 10.90.0.4 weight 65535
 no auto-summary
!
ip classless
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
end
```

A.2.3 7200 Hub Configuration

```
version 12.4
!
hostname 7200-HUB-1
!
ip cef
!
ip domain-name cisco.com
!
crypto pki trustpoint DMVPN-CA
 enrollment url http://10.24.1.130:80
 ip-address 10.71.101.1
 revocation-check none
 rsakeypair VPN1
!
crypto pki certificate chain DMVPN-CA
 certificate 361D
 <certificate removed>
 certificate ca 01
 <certificate removed>
quit
!
crypto isakmp policy 10
 encr 3des
 group 2
!
! The isakmp keepalive is set to be 5 seconds greater than the spoke tunnel
   routing protocol hello time. The router will retry 5 times before declaring
   the crypto session down. This works out to slightly longer than the routing
   protocols dead timer.
crypto isakmp keepalive 35 11
!
crypto isakmp profile hub
   match identity user domain cisco.com
   match identity address 0.0.0.0
!
crypto ipsec transform-set gre_set esp-3des esp-sha-hmac
 mode transport
!
crypto ipsec profile dmvpnhub
 set transform-set gre_set
!
! call admission limit keeps the router from getting hammered by crypto setup
   during a period of convergence.
!
call admission limit 90
!
controller ISA 1/1
!
! IRB is configured to allow the hub to have redundant connections to the SLB
   devices
!
bridge irb
!
! Buffer tuning is very important if the hub is going to support any number of
   routing protocol peers
!
buffers small permanent 373
buffers small max-free 533
```



```
buffers small min-free 111
buffers middle permanent 699
buffers middle max-free 873
buffers big permanent 361
buffers big max-free 516
buffers big min-free 108
buffers verybig permanent 384
buffers verybig max-free 480
buffers verybig min-free 115
buffers large permanent 37
buffers large max-free 46
buffers large min-free 13
buffers huge permanent 14
buffers huge max-free 17
buffers huge min-free 7
!
! The tunnel source ip address and tunnel ip address are the same for all hubs
   connected to the SLB device. Since this is a DMVPN Phase 3 design only a
   summary route needs to be sent to the spokes.
!
interface Loopback1
 description ****Tunnel Source****
 ip address 10.71.101.1 255.255.255.255
!
interface Tunnel1
 bandwidth 1000000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip tcp adjust-mss 1360
 ip hello-interval eigrp 10 15
 ip hold-time eigrp 10 90
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 ip summary-address eigrp 10 10.60.0.0 255.255.0.0 5
 load-interval 30
 tunnel source Loopback1
 tunnel mode gre multipoint
 tunnel protection ipsec profile dmvpnhub
 hold-queue 4096 in
 hold-queue 4096 out
!
! Tunnel2 is used to form an BGP peering with the other hub(s) in the cluster.
   This allows this hub to learn the spoke generated routes present on the other
   hubs. Notice that it has the same nhrp network-id as the tunnel to the
   spokes. This will cause the hub to send an NHRP redirect packet to the spoke
   originating traffic that will be terminated on another spoke within this
   cluster of hubs since the data packet entered and exited the router via
   interfaces with the same nhrp network-id.
!
interface Tunnel2
 ip address 10.71.102.1 255.255.255.0
 no ip redirects
 ip mtu 1400
```



```
 ip tcp adjust-mss 1360
 ip nhrp map 10.71.102.2 10.71.100.2
 ip nhrp map multicast 10.71.100.2
 ip nhrp network-id 102
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp cache non-authoritative
 ip nhrp shortcut
 ip nhrp redirect
 load-interval 30
 tunnel source 10.71.100.1
 tunnel mode gre multipoint
!
! These interfaces connect to the SLB devices. The use of IRB allows us to avoid
   having a single point of failure. If a plain routed interface were used and
   the switch that the port was connected to went down, the hub would become
   unreachable.
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 media-type gbic
 negotiation auto
 bridge-group 1
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 media-type gbic
 negotiation auto
 bridge-group 1
!
! Notice that the BVI is in the same subnet as VLAN20 on the SLB device. This is
   the SLB "real ip address" for this device
!
interface BVI1
 ip address 10.71.100.1 255.255.255.0
!
router eigrp 10
 network 10.81.0.0 0.0.255.255
 no auto-summary
!
! It is a good idea to set the BGP router id because each hub has the same ip
   address configured on its loopback interface. If BGP chooses that IP address
   as its router id the devices will not form a neighbor relationship. The next
   hop self command is used to change the next hop advertised to the other
   hub(s) from the remote spoke's IP address to this device's. EIGRP is
   redistributed which allows the other hubs to reach subnets behind spokes
   attached to this hub. In practice one would probably limit which EIGRP routes
   could be redistributed into EIGRP.
!
router bgp 5000
 no synchronization
 bgp router-id 10.71.102.1
 bgp log-neighbor-changes
 redistribute eigrp 10
 neighbor 10.71.102.2 remote-as 5000
 neighbor 10.71.102.2 next-hop-self
```



```
 no auto-summary
!
ip route 0.0.0.0 0.0.0.0 10.71.100.250
!
bridge 1 protocol ieee
bridge 1 route ip
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

A.36500 Crypto SLB Configuration

```
version 12.2
!
hostname 6500-CSLB-1
!
ip cef
!
ip domain-name cisco.com
!
ip slb probe PING-PROBE ping
 faildetect 3
!
ip slb serverfarm 7204-FARM
 predictor leastconns
 failaction purge
 probe PING-PROBE
 !
 real 10.72.100.2
  weight 1
  inservice
 !
 real 10.72.100.3
  weight 1
  inservice
!
ip slb vserver GRE
 virtual 10.72.2.2 255.255.255.254 gre
 serverfarm 7204-FARM
 no advertise
 idle 30
 inservice
!
crypto pki trustpoint DMVPN-CA
 enrollment url http://10.24.1.130:80
 serial-number none
 ip-address 10.72.2.2
 password
 subject-name CN=6500-CSLB-2, OU=NSITE
 revocation-check none
 rsakeypair VPN1
 auto-enroll
!
```



```
!
crypto pki certificate chain DMVPN-CA
 certificate 3607
 <certs omitted>
 certificate ca 01
!
vlan 5
 name CRYPTO_VLAN
!
vlan 10
 name VLAN_TO_HUBS
!
crypto isakmp policy 1
 encr 3des
 group 2
!
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 30 5
crypto isakmp profile hub
   ca trust-point DMVPN-CA
   match identity user domain cisco.com
   match identity address 0.0.0.0
!
crypto ipsec transform-set gre esp-3des esp-sha-hmac
 mode transport
!
crypto dynamic-map vpn1-dyna 10
 set transform-set gre
 set isakmp-profile hub
!
!
crypto map vpn1 local-address Vlan5
crypto map vpn1 10 ipsec-isakmp dynamic vpn1-dyna
!
interface GigabitEthernet1/0/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 5
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet1/0/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,5,1002-1005
 switchport mode trunk
 mtu 9216
 flowcontrol receive on
 flowcontrol send off
 spanning-tree portfast trunk
!
interface GigabitEthernet2/1
 description ****To 7200-CSLB-1 Gig0/1****
 switchport
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
```



```
interface GigabitEthernet2/2
 description ****To 7200-CSLB-2 Gig0/1****
 switchport
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface GigabitEthernet5/1
 description to GSR-VPN-CORE-1:g1/0
 no ip address
 crypto connect vlan 5
 spanning-tree portfast
!
interface Vlan5
 description ****Crypto Vlan****
 ip address 10.72.2.2 255.255.255.0
 no mop enabled
 crypto map vpn1
 crypto engine slot 1/0
!
interface Vlan10
 description ****Vlan Connected to Hubs*****
 ip address 10.72.100.1 255.255.255.0
 load-interval 30
!
!
line con 0
 exec-timeout 0 0
 password lab
 login
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
end
```

A.47200 Hub Configuration

```
version 12.4
!
hostname 7300-SLB-1
!
ip cef
!
interface Loopback1
 ip address 10.72.2.2 255.255.255.255
!
interface Tunnel1
 bandwidth 10000000
 ip address 10.81.0.1 255.255.0.0
 no ip redirects
 ip mtu 1400
 ip tcp adjust-mss 1360
 no ip next-hop-self eigrp 10
 ip nhrp authentication nsite
 ip nhrp map multicast dynamic
 ip nhrp network-id 102
```



```
 ip nhrp holdtime 900
 ip nhrp server-only
 ip nhrp max-send 65535 every 10
 ip nhrp shortcut
 ip nhrp redirect
 no ip split-horizon eigrp 10
 load-interval 30
 tunnel source Loopback1
 tunnel mode gre multipoint
 hold-queue 4096 in
 hold-queue 4096 out
!
interface GigabitEthernet0/2
 description CSLB-facing interface
 ip address 10.72.100.2 255.255.255.0
!
router eigrp 10
 network 10.81.0.0 0.0.255.255
!
ip route 0.0.0.0 0.0.0.0 10.72.100.1
!
line con 0
 exec-timeout 0 0
 password lab
 login
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 password lab
 login
!
end
```

