동적 멀티포인트 IPsec VPN(IPsec VPN의 확장을 위해 멀티포인트 GRE/NHRP 사용)

목차

소개

이 문서에서는 DMVPN(Dynamic Multipoint IPsec VPN)과 Cisco IOS® 소프트웨어에서 이 새로운 IPsec VPN 솔루션을 사용하기 위해 네트워크를 설계하거나 마이그레이션하려는 이유에 대해 설명합니다.

배경 정보

기업은 많은 사이트를 주 사이트에 상호 연결하고 인터넷을 통해 서로 상호 연결해야 하며 이를 보호하기 위해 트래픽을 암호화할 수 있습니다.예를 들어 재고 및 주문을 위해 회사 본사에 연결해야 하는 소매점 세트도 제품 가용성을 확인하기 위해 회사 내의 다른 저장소에 연결해야 할 수 있습니다.과거에는 ISDN 또는 Frame Relay와 같은 레이어 2 네트워크를 사용하여 모든 것을 상호 연결하는 방법밖에 없었습니다.내부 IP 트래픽에 대해 이러한 고정 유선 링크를 설치하고 비용을 지불하려면 많은 시간과 비용이 소요될 수 있습니다.모든 사이트(기본 사이트 포함)에 이미 비교적 저렴한 인터넷 액세스가 있는 경우, IPsec 터널을 사용하여 개인 정보 보호 및 데이터 무결성을 보장함으로써 점포와 본사 간의 내부 IP 통신에 이 인터넷 액세스를 사용할 수도 있습니다.

회사가 인터넷을 통해 사이트를 상호 연결하는 대규모 IPsec 네트워크를 구축하려면 IPsec 네트워크를 확장할 수 있어야 합니다.IPsec은 두 엔드포인트(피어) 간 트래픽을 암호화하고, 두 엔드포인트에서 공유 "비밀"을 사용하여 암호화를 수행합니다. 이 비밀은 이 두 엔드포인트 간에만 공유되므로, 암호화된 네트워크는 기본적으로 포인트-투-포인트 링크의 컬렉션입니다.따라서 IPsec은 기본적으로 포인트 투 포인트 터널 네트워크입니다.대규모 포인트 투 포인트 네트워크를 확장하기 위한 가장 적합한 방법은 허브 앤 스포크(hub and spoke) 또는 전체(부분) 메시 네트워크로 구성하는 것입니다.대부분의 네트워크에서 IP 트래픽의 대부분은 스포크와 허브 사이에 있으며 스포크 사이에는 거의 있지 않기 때문에 허브 앤 스포크(hub-and-spoke) 설계가 가장 적합합니다.이 설계는 오래된 Frame Relay 네트워크와도 일치합니다. 이러한 네트워크의 모든 사이트 간 링크에 비용을 지불하는 데 막대한 비용이 들었기 때문입니다.

허브와 스포크 간의 상호 연결로 인터넷을 사용할 경우 스포크는 추가 비용 없이 상호 직접 액세스할 수 있지만, 전체(부분) 메시 네트워크를 설정 및/또는 관리하는 것은 불가능하지는 않지만 매우 어려운 일입니다.전체 또는 부분 메시 네트워크는 스포크 투 스포크(spoke-to-spoke) 트래픽이 대신 허브를 통해 직접 통과할 수 있는 경우 비용을 절감할 수 있기 때문에 바람직합니다.허브를 지나는 스포크 투 스포크 트래픽은 허브 리소스를 사용하며, 특히 IPsec 암호화를 사용하는 경우, 허브가 전송 스포크의 수신 패킷을 해독하고 트래픽을 다시 암호화하여 수신 스포크로 전송해야 하므로 추가 지연이 발생할 수 있습니다.직접 스포크 투 스포크(Spoke-to-Spoke) 트래픽이 유용할 수 있는 또 다른 예는 두 스포크가 동일한 도시에 있고 허브가 전국적으로 있는 경우입니다.

IPsec 허브 앤 스포크(hub and spoke) 네트워크가 구축되어 크기가 증가함에 따라 최대한 동적으로 IP 패킷을 라우팅하는 것이 더 바람직해졌습니다.이전 Frame Relay 허브-스포크 네트워크에서는 OSPF 또는 EIGRP와 같은 동적 라우팅 프로토콜을 Frame Relay 링크를 통해 실행하여 이러한 작업을 수행했습니다.이는 스포크 네트워크의 연결성을 동적으로 광고하고 IP 라우팅 네트워크에서 이중화를 지원하는 데 유용합니다.네트워크에서 허브 라우터가 손실되면 백업 허브 라우터가 자동으로 인수되어 스포크 네트워크에 대한 네트워크 연결을 유지할 수 있습니다.

IPsec 터널 및 동적 라우팅 프로토콜에 근본적인 문제가 있습니다.동적 라우팅 프로토콜은 IP 멀티캐스트 또는 브로드캐스트 패킷 사용에 의존하지만 IPsec은 멀티캐스트 또는 브로드캐스트 패킷 암호화를 지원하지 않습니다.이 문제를 해결하기 위한 현재 방법은 IPsec 암호화와 함께 GRE(generic routing encapsulation) 터널을 사용하는 것입니다.

GRE 터널은 IP 멀티캐스트 및 브로드캐스트 패킷을 GRE 터널의 다른 끝으로 전송하는 것을 지원합니다.GRE 터널 패킷은 IP 유니캐스트 패킷이므로 IPsec을 사용하여 GRE 패킷을 암호화할 수 있습니다.이 시나리오에서는 GRE가 터널링 작업을 수행하고 IPsec은 VPN 네트워크를 지원하는 암호화 부분을 수행합니다.GRE 터널이 구성된 경우 터널의 엔드포인트(**터널 소스 ...**, **터널 대상 ...**)에 대한 IP 주소를 다른 엔드포인트에서 알고 있어야 하며 인터넷을 통해 라우팅 가능해야 합니다.즉, 이 네트워크의 허브 및 모든 스포크 라우터에 고정 비 프라이빗 IP 주소가 있어야 합니다.

소규모 사이트 간 인터넷 연결의 경우 스포크의 외부 IP 주소가 인터넷에 연결될 때마다 변경되는 것이 일반적입니다. ISP(Internet Service Provider)는 스포크가 온라인(비대칭 ADSL(Digital Subscriber Line) 및 케이블 서비스)에 도달할 때마다 외부 인터페이스 주소(DHCP(Dynamic Host Configuration Protocol)를 통해)를 동적으로 제공하기 때문입니다. 이러한 라우터의 "외부 주소"를 동적으로 할당하면 모든 사용자가 동시에 온라인 상태가 되는 것은 아니므로 ISP는 인터넷 주소 공

간 사용을 초과 등록할 수 있습니다.스포크 라우터에 고정 주소를 할당하기 위해 제공자에게 비용을 지불하는 것은 훨씬 더 많은 비용이 들 수 있습니다.IPsec VPN을 통해 동적 라우팅 프로토콜을 실행하려면 GRE 터널을 사용해야 하지만 외부 물리적 인터페이스에 동적으로 할당된 IP 주소를 가진 스포크를 사용할 수 있는 옵션이 없습니다.

위의 제한 사항 및 기타 몇 가지 사항은 다음 4가지로 요약됩니다.

- IPsec은 ACL(Access Control List)을 사용하여 암호화할 데이터를 정의합니다.따라서 스포크 또는 허브 뒤에 새(하위) 네트워크가 추가될 때마다 고객은 허브 라우터와 스포크 라우터 모두에서 ACL을 변경해야 합니다.SP에서 라우터를 관리하는 경우 고객은 새 트래픽이 암호화되도록 IPsec ACL을 변경하려면 SP에 알려야 합니다.
- 대형 허브 앤 스포크(hub and spoke) 네트워크에서는 허브 라우터의 컨피그레이션 크기가 사용 불가능한 정도로 매우 커질 수 있습니다.예를 들어 허브 라우터는 300개의 스포크 라우터를 지원하려면 최대 3,900개의 구성 라인이 필요합니다.이는 컨피그레이션을 표시하고 디버깅 중인 현재 문제와 관련된 컨피그레이션 섹션을 찾기 어려울 정도로 충분히 큽니다.또한 이 크기 구성이 너무 커서 NVRAM에 맞지 않을 수 있으며 플래시 메모리에 저장해야 합니다.
- GRE + IPsec은 엔드포인트 피어 주소를 알아야 합니다.스포크의 IP 주소는 자체 ISP를 통해 인터넷에 직접 연결되며 외부 인터페이스 주소가 고정되지 않도록 설정되는 경우가 많습니다.IP 주소는 DHCP를 통해 사이트가 온라인 상태가 될 때마다 변경될 수 있습니다.
- 스포크가 IPsec VPN을 통해 서로 직접 통신해야 하는 경우 허브 앤 스포크 네트워크는 풀 메쉬가 되어야 합니다.어떤 스포크가 서로 직접 통화해야 하는지 아직 알 수 없으므로 각 스포크가 서로 직접 통화할 필요가 없더라도 전체 메쉬가 필요합니다.또한 소규모 스포크 라우터에서 IPsec을 구성하여 네트워크의 다른 모든 스포크 라우터와 직접 연결할 수 없습니다.따라서 스포크 라우터는 더 강력한 라우터가 필요할 수 있습니다.

DMVPN 솔루션

DMVPN 솔루션은 IPsec 및 일부 새로운 개선 사항과 함께 mGRE(Multipoint GRE) 및 NHRP(Next Hop Resolution Protocol)를 사용하여 위의 문제를 확장 가능한 방식으로 해결합니다.

자동 IPsec 암호화 시작

DMVPN 솔루션을 사용하지 않는 경우 이 IPsec 터널을 사용해야 하는 데이터 트래픽이 있을 때까지 IPsec 암호화 터널이 시작되지 않습니다.IPsec 터널 시작을 완료하는 데 1~10초가 걸릴 수 있으며 이 시간 동안 데이터 트래픽이 삭제됩니다.GRE를 IPsec과 함께 사용하는 경우 GRE 터널 컨피그레이션에는 IPsec 피어 주소인 GRE 터널 피어(**터널 대상 ...**) 주소가 이미 포함되어 있습니다.이 두 주소는 모두 사전 구성됩니다.

허브 라우터에서 TED(Tunnel Endpoint Discovery) 및 동적 암호화 맵을 사용하는 경우 허브에서 IPsec 피어 주소를 미리 구성할 필요가 없지만 ISAKMP 협상을 시작하기 전에 TED 프로브 및 응답을 전송 및 받아야 합니다.GRE를 사용할 때 피어 소스 및 목적지 주소를 이미 알고 있으므로 이 작업은 필요하지 않습니다.컨피그레이션에 있거나 NHRP로 해결됩니다(멀티포인트 GRE 터널의 경우).

DMVPN 솔루션을 사용하면 IPsec이 포인트-투-포인트 GRE 터널 모두에 대해 즉시 트리거됩니다.또한 암호화 ACL은 GRE 터널 소스 및 대상 주소에서 자동으로 파생되므로 어떤 암호화 ACL도 구성할 필요가 없습니다.다음 명령은 IPsec 암호화 매개변수를 정의하는 데 사용됩니다.**set peer ...** 또는 **match address ...** 명령이 없습니다. 이 정보는 연결된 GRE 터널 또는 NHRP 매핑에서 직접 파생됩니다.

```
crypto ipsec profile
```

```
 set transform-set
```

다음 명령은 터널 인터페이스를 IPsec 프로파일과 연결합니다.

```
interface tunnel
```

```
 ...
 tunnel protection ipsec profile
```

"Spoke-to-Hub" 링크를 위한 동적 터널 생성

DMVPN 네트워크의 허브 라우터에 스포크에 대한 GRE 또는 IPsec 정보가 구성되지 않았습니다 .스포크 라우터의 GRE 터널은 허브 라우터에 대한 정보를 사용하여 (NHRP 명령을 통해) 구성됩니다.스포크 라우터가 시작되면 위에서 설명한 대로 허브 라우터로 IPsec 터널을 자동으로 시작합니다.그런 다음 NHRP를 사용하여 허브 라우터에 현재 물리적 인터페이스 IP 주소를 알립니다.이 기능은 다음과 같은 세 가지 이유로 유용합니다.

- 스포크 라우터의 물리적 인터페이스 IP 주소가 동적으로 할당된 경우(예: ADSL 또는 CableModem), 스포크 라우터가 다시 로드될 때마다 새 물리적 인터페이스 IP 주소를 얻으므로 허브 라우터를 이 정보로 구성할 수 없습니다.
- 피어 라우터에 대한 GRE 또는 IPsec 정보가 필요하지 않으므로 허브 라우터의 컨피그레이션이 단축되고 간소화됩니다.이 모든 정보는 NHRP를 통해 동적으로 학습됩니다.
- DMVPN 네트워크에 새 스포크 라우터를 추가할 때 허브 또는 현재 스포크 라우터의 컨피그레이션을 변경할 필요가 없습니다.새 스포크 라우터는 허브 정보로 구성되며, 시작할 때 허브 라우터에 동적으로 등록됩니다.동적 라우팅 프로토콜은 이 스포크의 라우팅 정보를 허브에 전파합니다.허브는 이 새 라우팅 정보를 다른 스포크로 전파합니다.또한 다른 스포크의 라우팅 정보를 이 스포크로 전파합니다.

"스포크 투 스포크" 트래픽을 위한 동적 터널 생성

앞에서 설명한 것처럼 현재 메시 네트워크에서는 이러한 터널의 일부/대부분이 실행 중이 아니거나

항상 필요하지 않더라도 모든 포인트-투-포인트 IPsec(또는 IPsec+GRE) 터널을 모든 라우터에 구성해야 합니다.DMVPN 솔루션을 사용하면 하나의 라우터가 허브이고 다른 모든 라우터(스포크)는 허브에 대한 터널로 구성됩니다.스포크 투 허브 터널은 계속 작동하며 스포크는 다른 스포크에 대한 직접 터널을 구성할 필요가 없습니다.대신 스포크가 다른 스포크(예: 다른 스포크 뒤에 있는 서브넷)로 패킷을 전송하려는 경우 NHRP를 사용하여 대상 스포크의 필수 대상 주소를 동적으로 결정합니다.허브 라우터는 NHRP 서버 역할을 하며 소스 스포크에 대해 이 요청을 처리합니다.그런 다음 두 스포크는 단일 mGRE 인터페이스를 통해 IPsec 터널을 동적으로 생성하고 데이터를 직접 전송할 수 있습니다.이 동적 스포크 투 스포크 터널은 (구성 가능한) 기간 동안 비활성 상태가 지속되면 자동으로 해제됩니다.

동적 라우팅 프로토콜 지원

DMVPN 솔루션은 터널링 멀티캐스트/브로드캐스트 IP 패킷을 지원하는 GRE 터널을 기반으로 하므로 DMVPN 솔루션은 IPsec+mGRE 터널을 통해 실행되는 동적 라우팅 프로토콜도 지원합니다.이전에는 NHRP에서 멀티캐스트 및 브로드캐스트 IP 패킷의 GRE 터널링을 지원하기 위해 터널 대상 IP 주소에 대한 브로드캐스트/멀티캐스트 매핑을 명시적으로 구성해야 했습니다.예를 들어 허브에서 각 스포크에 **ip nhrp map multicast <spoke-n-addr>** 구성 라인이 필요합니다.DMVPN 솔루션을 사용하면 스포크 주소를 미리 알 수 없으므로 이 컨피그레이션을 수행할 수 없습니다.대신 ip nhrp map multicast dynamic 명령을 사용하여 허브의 멀티캐스트 대상 목록에 각 스포크를 자동으로 추가하도록 NHRP를 구성할 수 있습니다.이 명령을 사용하여 스포크 라우터가 NHRP 서버(허브)에 유니캐스트 NHRP 매핑을 등록할 때 NHRP는 이 스포크에 대한 브로드캐스트/멀티캐스트 매핑도 생성합니다.따라서 스포크 주소를 미리 알 필요가 없습니다.

mGRE용 Cisco Express Forwarding Fast Switching

현재 mGRE 인터페이스의 트래픽은 프로세스 스위칭되므로 성능이 저하됩니다.DMVPN 솔루션은 mGRE 트래픽을 위해 Cisco Express Forwarding 스위칭을 추가하므로 성능이 훨씬 향상됩니다.이 기능을 설정하는 데 필요한 구성 명령이 없습니다.GRE 터널 인터페이스와 발신/수신 물리적 인터페이스에서 Cisco Express Forwarding 스위칭이 허용되는 경우 멀티포인트 GRE 터널 패킷은 Cisco Express Forwarding-switched가 됩니다.

IPsec 보호 VPN을 통한 동적 라우팅 사용

이 섹션에서는 현재(DMVPN 이전 솔루션) 상태의 상황에 대해 설명합니다.IPsec은 암호화를 정의한 명령 집합을 통해 Cisco 라우터에 구현된 다음 라우터의 외부 인터페이스에 **crypto map <map-name>** 명령을 적용합니다.이러한 설계와 IPsec을 사용하여 IP 멀티캐스트/브로드캐스트 패킷을 암호화하는 표준이 없다는 사실 때문에 IP 라우팅 프로토콜 패킷은 IPsec 터널을 통해 "전달"할 수 없으며 라우팅 변경 사항은 IPsec 터널의 반대쪽으로 동적으로 전파될 수 없습니다.

참고: BGP를 제외한 모든 동적 라우팅 프로토콜은 브로드캐스트 또는 멀티캐스트 IP 패킷을 사용합니다.GRE 터널은 이 문제를 해결하기 위해 IPsec과 함께 사용됩니다.

GRE 터널은 가상 터널 인터페이스(**인터페이스 터널<#>**)를 사용하여 Cisco 라우터에 구현됩니다.GRE 터널링 프로토콜은 IP 멀티캐스트/브로드캐스트 패킷을 처리하도록 설계되어 GRE 터널을 통해 동적 라우팅 프로토콜을 "실행"할 수 있습니다.GRE 터널 패킷은 원래 IP 멀티캐스트/유니캐스트 패킷을 캡슐화하는 IP 유니캐스트 패킷입니다.그런 다음 IPsec을 사용하여 GRE 터널 패킷을 암호화할 수 있습니다.GRE가 이미 원래 데이터 패킷을 캡슐화했으므로 IPsec이 다른 IP 헤더에 GRE IP 패킷을 캡슐화할 필요가 없으므로 전송 모드에서 IPsec을 실행하고 20바이트를 저장할 수도 있습니다.

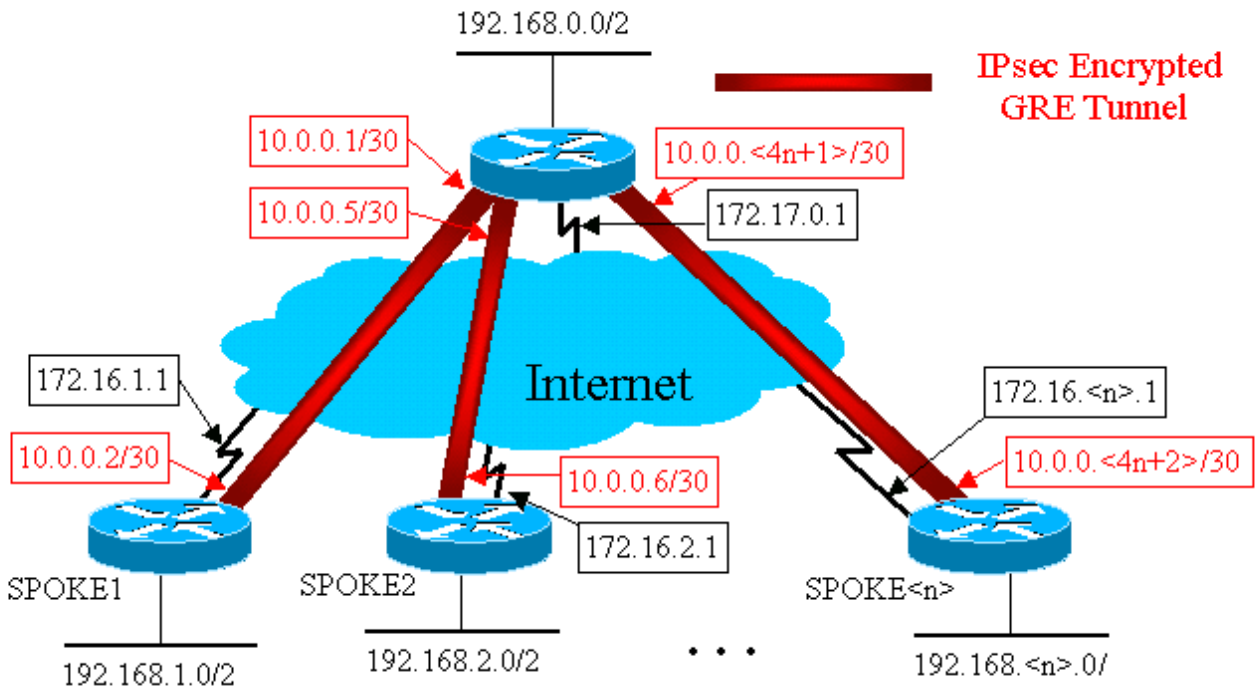
전송 모드에서 IPsec을 실행하는 경우 암호화할 패킷의 IP 소스 및 목적지 주소가 IPsec 피어 주소

(라우터 자체)와 일치해야 한다는 제한이 있습니다. 이 경우 GRE 터널 엔드포인트와 IPsec 피어 주소가 동일해야 함을 의미합니다.동일한 라우터가 IPsec 및 GRE 터널 엔드포인트이므로 이는 문제가 아닙니다.GRE 터널을 IPsec 암호화와 결합하면 동적 IP 라우팅 프로토콜을 사용하여 암호화된 터널의 양쪽 끝에서 라우팅 테이블을 업데이트할 수 있습니다.암호화된 터널을 통해 학습된 네트워크에 대한 IP 라우팅 테이블 항목은 터널의 다른 끝(GRE 터널 인터페이스 IP 주소)을 IP next hop으로 가집니다.따라서 터널의 양쪽에서 네트워크가 변경되면 상대측에서 동적으로 변경 사항을 학습하고 라우터에서 어떤 컨피그레이션도 변경하지 않고 연결을 계속 진행합니다.

기본 구성

다음은 표준 포인트-투-포인트 IPsec+GRE 컨피그레이션입니다.그 다음에는 DMVPN 솔루션의 특정 기능을 단계별로 추가하여 DMVPN의 다양한 기능을 보여 주는 일련의 컨피그레이션 예가 있습니다.각 예는 이전 예를 기반으로 점점 복잡해지는 네트워크 설계에서 DMVPN 솔루션을 사용하는 방법을 보여줍니다.이러한 예제의 연속은 현재 IPsec+GRE VPN을 DMVPN으로 마이그레이션하는 템플릿으로 사용할 수 있습니다.특정 컨피그레이션 예가 네트워크 설계 요구 사항과 일치하는 경우 언제든지 "마이그레이션"을 중지할 수 있습니다.

IPsec + GRE 허브 및 스포크(n = 1,2,3,...)



허브 라우터

```
version 12.3
 !
 hostname Hub
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
```

```
 crypto map vpnmap1 local-address Ethernet0
 crypto map vpnmap1 10 ipsec-isakmp
  set peer 172.16.1.1
  set transform-set trans2
  match address 101
 crypto map vpnmap1 20 ipsec-isakmp
  set peer 172.16.2.1
  set transform-set trans2
  match address 102
 . . .
 crypto map vpnmap1 <10*n> ipsec-isakmp
  set peer 172.16.




 interface Tunnel1
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.252
  ip mtu 1400
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.16.1.1
 !
 interface Tunnel2
  bandwidth 1000
  ip address 10.0.0.5 255.255.255.252
  ip mtu 1400
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.16.2.1
 !
 . . .
 !
 interface Tunnel




 !
 interface Ethernet0
  ip address 172.17.0.1 255.255.255.0
  crypto map vpnmap1
 !
 interface Ethernet1
  ip address 192.168.0.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
  no auto-summary
 !
 access-list 101 permit gre host 172.17.0.1 host
172.16.1.1
 access-list 102 permit gre host 172.17.0.1 host
172.16.2.1
 ...
 access-list
```

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto map vpnmap1 local-address Ethernet0
 crypto map vpnmap1 10 ipsec-isakmp
  set peer 172.17.0.1
  set transform-set trans2
  match address 101
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.2 255.255.255.252
  ip mtu 1400
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
 !
 interface Ethernet0
  ip address 172.16.1.1 255.255.255.252
  crypto map vpnmap1
 !
 interface Ethernet1
  ip address 192.168.1.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
 !
 access-list 101 permit gre host 172.16.1.1 host
172.17.0.1
```

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto map vpnmap1 local-address Ethernet0
 crypto map vpnmap1 10 ipsec-isakmp
  set peer 172.17.0.1
  set transform-set trans2
```

```
 match address 101
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.6 255.255.255.252
 ip mtu 1400
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
!
interface Ethernet0
 ip address 172.16.2.1 255.255.255.252
 crypto map vpnmap1
!
interface Ethernet1
 ip address 192.168.2.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.2.0 0.0.0.255
 no auto-summary
!
access-list 101 permit gre host 172.16.2.1 host
172.17.0.1
```

스포크<n> 라우터

```
version 12.3
!
hostname Spoke<n>
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto map vpnmap1 local-address Ethernet0
crypto map vpnmap1 10 ipsec-isakmp
 set peer 172.17.0.1
 set transform-set trans2
 match address 101
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.<4n-2> 255.255.255.252
 ip mtu 1400
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
!
interface Ethernet0
 ip address 172.16.<n>.1 255.255.255.252
 crypto map vpnmap1
!
interface Ethernet1
 ip address 192.168.<n>.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.
```

위 컨피그레이션에서는 ACL을 사용하여 어떤 트래픽이 암호화되는지를 정의합니다.허브 라우터와 스포크 라우터 모두에서 이 ACL은 GRE 터널 IP 패킷만 매칭해야 합니다.양쪽 끝에서 네트워크가 어떻게 변경되더라도 GRE IP 터널 패킷은 변경되지 않으므로 이 ACL은 변경할 필요가 없습니다.

참고: 12.2(13)T 이전 Cisco IOS 소프트웨어 버전을 사용하는 경우 GRE 터널 인터페이스(Tunnel<x>) 및 물리적 인터페이스(Ethernet0)에 **crypto map vpnmap1** 컨피그레이션 명령을 적용해야 합니다. Cisco IOS 버전 12.2(13)T 이상에서는 **crypto map vpnmap1** 컨피그레이션 명령만 물리적 인터페이스(Ethernet0)에 적용합니다.

허브 및 스포크 라우터의 라우팅 테이블 예

허브 라우터의 라우팅 테이블

```
172.17.0.0/24 is subnetted, 1 subnets
 C       172.17.0.0 is directly connected, Ethernet0
      10.0.0.0/30 is subnetted, <n> subnets
 C       10.0.0.0 is directly connected, Tunnel1
 C       10.0.0.4 is directly connected, Tunnel2
 ...
 C       10.0.0.<4n-4> is directly connected, Tunnel<n>
 C    192.168.0.0/24 is directly connected, Ethernet1
 D    192.168.1.0/24 [90/2841600] via 10.0.0.2,
18:28:19, Tunnel1
 D    192.168.2.0/24 [90/2841600] via 10.0.0.6, 2d05h,
Tunnel2
 ...
 D    192.168.<n>.0/24 [90/2841600] via 10.0.0.<4n-2>,
2d05h, Tunnel<n>
```

Spoke1 라우터의 라우팅 테이블

```
172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.1.0 is directly connected, Ethernet0
      10.0.0.0/30 is subnetted, <n> subnets
 C       10.0.0.0 is directly connected, Tunnel1
 D       10.0.0.4 [90/2841600] via 10.0.0.1, 23:00:58,
Tunnel0
 ...
 D      10.0.0.<4n-4> [90/2841600] via 10.0.0.1,
23:00:58, Tunnel0
 D    192.168.0.0/24 [90/2841600] via 10.0.0.1,
23:00:58, Tunnel0
 C    192.168.1.0/24 is directly connected, Loopback0
 D    192.168.2.0/24 [90/3097600] via 10.0.0.1,
23:00:58, Tunnel0
 ...
 D    192.168.<n>.0/24 [90/3097600] via 10.0.0.1,
23:00:58, Tunnel0
```

Spoke<n> 라우터의 라우팅 테이블

```
172.16.0.0/24 is subnetted, 1 subnets
```

```
 C       172.16.<n>.0 is directly connected, Ethernet0
      10.0.0.0/30 is subnetted, <n> subnets
 D       10.0.0.0 [90/2841600] via 10.0.0.1, 22:01:21,
Tunnel0
 D       10.0.0.4 [90/2841600] via 10.0.0.1, 22:01:21,
Tunnel0
 ...
 C        10.0.0.<4n-4> is directly connected, Tunnel0
 D    192.168.0.0/24 [90/2841600] via 10.0.0.1,
22:01:21, Tunnel0
 D    192.168.1.0/24 [90/3097600] via 10.0.0.1,
22:01:21, Tunnel0
 D    192.168.2.0/24 [90/3097600] via 10.0.0.1,
22:01:21, Tunnel0
 ...
 C    192.168.<n>.0/24 is directly connected, Ethernet0
```

이는 기본적인 작업 구성이며 DMVPN 솔루션을 사용하여 가능한 더 복잡한 컨피그레이션과 비교하기 위한 시작점으로 사용됩니다.첫 번째 변경은 허브 라우터의 컨피그레이션 크기를 줄입니다.이는 소수의 스포크 라우터에서는 중요하지 않지만, 스포크 라우터가 50~100개 이상인 경우에는 중요한 문제가 됩니다.

허브 라우터 컨피그레이션 크기 감소

다음 예에서는 허브 라우터의 컨피그레이션이 여러 GRE 지점 간 터널 인터페이스에서 단일 GRE 다중 지점 터널 인터페이스로 최소 변경됩니다.이는 DMVPN 솔루션의 첫 번째 단계입니다.

허브 라우터에는 각 스포크 라우터에 대한 암호화 맵 특성을 정의하기 위한 고유한 구성 라인 블록이 있습니다.이 컨피그레이션 부분은 해당 스포크 라우터에 대한 암호화 ACL 및 GRE 터널 인터페이스를 정의합니다.이러한 특성은 IP 주소(**set peer ...**, **터널 대상 ...**)를 제외하고 모든 스포크에 대해 대부분 동일합니다.

허브 라우터에서 위의 컨피그레이션을 보면 스포크 라우터당 최소 13개의 컨피그레이션 라인이 있음을 알 수 있습니다.암호화 맵의 경우 4개, 암호화 ACL의 경우 1개, GRE 터널 인터페이스의 경우 8개입니다.300개의 스포크 라우터가 있는 경우 총 구성 행 수는 3,900개입니다.또한 각 터널 링크 주소 지정을 위해 300(/30) 서브넷이 필요합니다.이 크기의 컨피그레이션은 관리가 매우 어렵고 VPN 네트워크 트러블슈팅을 수행할 때 더욱 어렵습니다.이 값을 줄이려면 동적 암호화 맵을 사용하여 위의 값을 1200개 회선으로 줄이고 300개 스포크 네트워크에 2700개의 회선을 남겨 둘 수 있습니다.

참고: 동적 암호화 맵을 사용할 경우 스포크 라우터에서 IPsec 암호화 터널을 시작해야 합니다.또한 **ip unnumbered <interface>**를 사용하여 GRE 터널에 필요한 서브넷 수를 줄일 수 있지만 나중에 문제를 더 어렵게 만들 수 있습니다.

DMVPN 솔루션을 사용하면 허브 라우터에서 단일 멀티포인트 GRE 터널 인터페이스 및 단일 IPsec 프로필을 구성하여 모든 스포크 라우터를 처리할 수 있습니다.이렇게 하면 VPN 네트워크에 추가된 스포크 라우터 수에 관계없이 허브 라우터의 컨피그레이션 크기가 일정하게 유지됩니다.

DMVPN 솔루션에는 다음과 같은 새로운 명령이 도입되었습니다.

crypto ipsec profile

crypto **ipsec profile <name>** 명령은 동적 암호화 맵과 같이 사용되며, 터널 인터페이스용으로 특별히 설계되었습니다.이 명령은 스포크 투 허브 및 스포크 투 스포크 VPN 터널에서 IPsec 암호화에 대한 매개변수를 정의하는 데 사용됩니다.프로파일에서 필요한 유일한 매개변수는 변형 집합입니다.IPsec 피어 주소 및 **IPsec** 프록시에 대한 **일치 주소 ...** 절은 GRE 터널에 대한 NHRP 매핑에서 자동으로 파생됩니다.

tunnel protection ipsec profile <name> 명령은 GRE 터널 인터페이스 아래에 구성되며 GRE 터널 인터페이스를 IPsec 프로필과 연결하는 데 사용됩니다.또한 **tunnel protection ipsec profile <name>** 명령을 포인트-투-포인트 GRE 터널과 함께 사용할 수도 있습니다.이 경우 IPsec 피어 및 프록시 정보가 **터널 소스 ...** 및 **터널 대상 ...** 컨피그레이션**에서 파생됩니다**.이렇게 하면 IPsec 피어와 암호화 ACL이 더 이상 필요하지 않으므로 컨피그레이션이 간소화됩니다.

참고: tunnel protection ... 명령은 GRE 캡슐화가 패킷에 추가된 후 IPsec 암호화를 수행하도록 지정합니다.

처음 두 개의 새 명령은 암호화 맵을 구성하고 crypto map **<name>** 명령을 사용하여 인터페이스에 암호화 맵**을 할당하는 것과** 유사합니다.큰 차이점은 새로운 명령을 사용하면 암호화할 패킷과 일치시키기 위해 IPsec 피어 주소 또는 ACL을 지정할 필요가 없다는 것입니다.이러한 매개변수는 mGRE 터널 인터페이스의 NHRP 매핑에서 자동으로 결정됩니다.

참고: 터널 인터페이스에서 **tunnel protection ...** 명령을 사용할 경우, **crypto map ...** 명령은 물리적 발신 인터페이스에 구성되지 않습니다.

마지막 새 명령인 **ip nhrp map multicast dynamic**을 사용하면 이러한 스포크 라우터가 mGRE+IPsec 터널을 시작하고 유니캐스트 NHRP 매핑을 등록할 때 NHRP가 멀티캐스트 NHRP 매핑에 스포크 라우터를 자동으로 추가할 수 있습니다.이는 동적 라우팅 프로토콜이 허브와 스포크 간의 mGRE+IPsec 터널을 통해 작동하도록 하려면 필요합니다.이 명령을 사용할 수 없는 경우 허브 라우터에는 각 스포크에 대한 멀티캐스트 매핑을 위한 별도의 구성 행이 있어야 합니다.

참고: 이 컨피그레이션에서는 허브 라우터가 스포크에 대한 정보로 구성되지 않았으므로 스포크 라우터가 mGRE+IPsec 터널 연결을 시작해야 합니다.그러나 DMVPN에서는 스포크 라우터가 시작될 때 mGRE+IPsec 터널이 자동으로 시작되어 항상 작동 상태로 유지되기 때문에 이 문제는 아닙니다.

참고: 다음 예는 스포크 라우터의 포인트-투-포인트 GRE 터널 인터페이스와 허브 라우터와 스포크 라우터에서 mGRE 터널을 지원하기 위해 추가된 NHRP 구성 라인을 보여줍니다.컨피그레이션 변경 사항은 다음과 같습니다.

허브 라우터(이전)

```
crypto map vpnmap1 10 IPsec-isakmp
  set peer 172.16.1.1
  set transform-set trans2
  match address 101
 crypto map vpnmap1 20 IPsec-isakmp
  set peer 172.16.2.1
  set transform-set trans2
  match address 102
 . . .
```

```
 crypto map vpnmap1 <10n> IPsec-isakmp
  set peer 172.16.
 !
 interface Ethernet0
  ip address 172.17.0.1 255.255.255.0
  crypto map vpnmap1
 !
 access-list 101 permit gre host 172.17.0.1 host
172.16.1.1
 access-list 102 permit gre host 172.17.0.1 host
172.16.2.1
 . . .
 access-list
```

허브 라우터(신규)

```
crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  no ip split-horizon eigrp 1
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address 172.17.0.1 255.255.255.0
```

Spoke<n> 라우터(이전)

```
crypto map vpnmap1 10 IPsec-isakmp
  set peer 172.17.0.1
  set transform-set trans2
  match address 101
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.<4n-2> 255.255.255.252
  ip mtu 1400
  delay 1000
  tunnel source Ethernet0
```

```
  tunnel destination 172.17.0.1
 !
interface Ethernet0
  ip address 172.16.<n>.1 255.255.255.252
  crypto map vpnmap1
!
 . . .
 !
 access-list 101 permit gre host 172.16.<n>.1 host
172.17.0.1
 !
```

Spoke<n> 라우터(신규)

```
crypto map vpnmap1 10 IPsec-isakmp
  set peer 172.17.0.1
  set transform-set trans2
  match address 101
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.




  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
 !
interface Ethernet0
  ip address 172.16.<n>.1 255.255.255.252
  crypto map vpnmap1
!
 . . .
 !
 access-list 101 permit gre host 172.16.<n>.1 host
172.17.0.1
 !
```

스포크 라우터에서 서브넷 마스크가 변경되고 터널 인터페이스 아래에 NHRP 명령이 추가되었습니다.허브 라우터가 이제 NHRP를 사용하여 스포크 터널 인터페이스 IP 주소를 스포크 물리적 인터페이스 IP 주소에 매핑하기 때문에 NHRP 명령이 필요합니다.

ip address 10.0.0.

```
      ip mtu 1400
      ip nhrp authentication test
      ip nhrp map 10.0.0.1 172.17.0.1
      ip nhrp network-id 100000
      ip nhrp holdtime 300
      ip nhrp nhs 10.0.0.1
      ...
```

```
tunnel key 100000
```

서브넷이 /30 대신 /24가 되었으므로 모든 노드가 서로 다른 서브넷 대신 동일한 서브넷에 있습니다.스포크는 point-to-point GRE 터널 인터페이스를 사용하므로 여전히 허브를 통해 스포크 투 스포크 트래픽을 전송합니다.ip nhrp **인증 ...**, ip nhrp **network-id ...** 및 **tunnel key ...** 명령을 사용하여 허브에서 수신 시 터널 패킷과 NHRP 패킷을 올바른 다중 지점 GRE 터널 인터페이스 및 NHRP 네트워크에 매핑합니다.ip nhrp **맵 ...** 및 **ip nhrp nhs ... 명령은 스포크의 NHRP**에서 스포크 NHRP 매핑(10.0.0.<n+1> —> 172.16.<n>.1)을 허브에 광고하는 데 사용됩니다.10.0.0.<n+1> 주소는 터널 인터페이스의 **ip address ...** 명령 및 터널 인터페이스의 172.16.<n>.1 주소는 **터널 대상 ...** 명령에서 검색됩니다.

300개의 스포크 라우터가 있는 경우, 이러한 변경으로 허브의 컨피그레이션 라인 수가 3900개에서 16개 라인(3884개 라인 감소)으로 줄어듭니다. 각 스포크 라우터의 컨피그레이션이 6라인씩 증가합니다.

스포크의 동적 주소 지원

Cisco 라우터에서 각 IPsec 피어를 다른 IPsec 피어의 IP 주소로 구성해야 IPsec 터널을 시작할 수 있습니다.스포크 라우터의 물리적 인터페이스에 동적 주소가 있는 경우 이 작업에 문제가 있습니다. 이는 DSL 또는 케이블 링크를 통해 연결된 라우터에 일반적으로 사용됩니다.

TED를 사용하면 암호화해야 하는 원래 데이터 패킷의 IP 목적지 주소로 특별 ISAKMP(Internet Security Association and Key Management Protocol) 패킷을 전송하여 한 IPsec 피어가 다른 IPsec 피어를 찾을 수 있습니다.이 패킷은 IPsec 터널 패킷에서 수행한 것과 동일한 경로를 따라 중간 네트워크를 통과한다고 가정합니다.이 패킷은 첫 번째 피어에 응답하는 다른 엔드 IPsec 피어에서 선택됩니다.그러면 두 라우터가 ISAKMP 및 IPsec SA(Security Associations)를 협상하고 IPsec 터널을 가져옵니다.이는 암호화할 데이터 패킷에 라우팅 가능한 IP 주소가 있는 경우에만 작동합니다.

TED는 이전 섹션에서 구성한 대로 GRE 터널과 함께 사용할 수 있습니다.이전 버전의 Cisco IOS 소프트웨어에 버그가 있었지만, TED가 GRE 터널 패킷뿐만 아니라 두 IPsec 피어 간의 모든 IP 트래픽을 암호화하도록 강요했습니다.DMVPN 솔루션은 호스트가 인터넷 라우팅 가능 IP 주소를 사용할 필요 없이 프로브 및 응답 패킷을 보낼 필요 없이 이러한 기능과 추가 기능을 제공합니다.약간 수정하면 마지막 섹션의 컨피그레이션을 사용하여 외부 물리적 인터페이스에서 동적 IP 주소가 있는 스포크 라우터를 지원할 수 있습니다.

허브 라우터(변경 없음)

```
crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  no ip split-horizon eigrp 1
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
```

```
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.0
```

Spoke<n> 라우터(이전)

```
crypto map vpnmap1 10 IPsec-isakmp
 set peer 172.17.0.1
 set transform-set trans2
 match address 101
!
...
!
access-list 101 permit gre host 172.16.
```

Spoke<n> 라우터(신규)

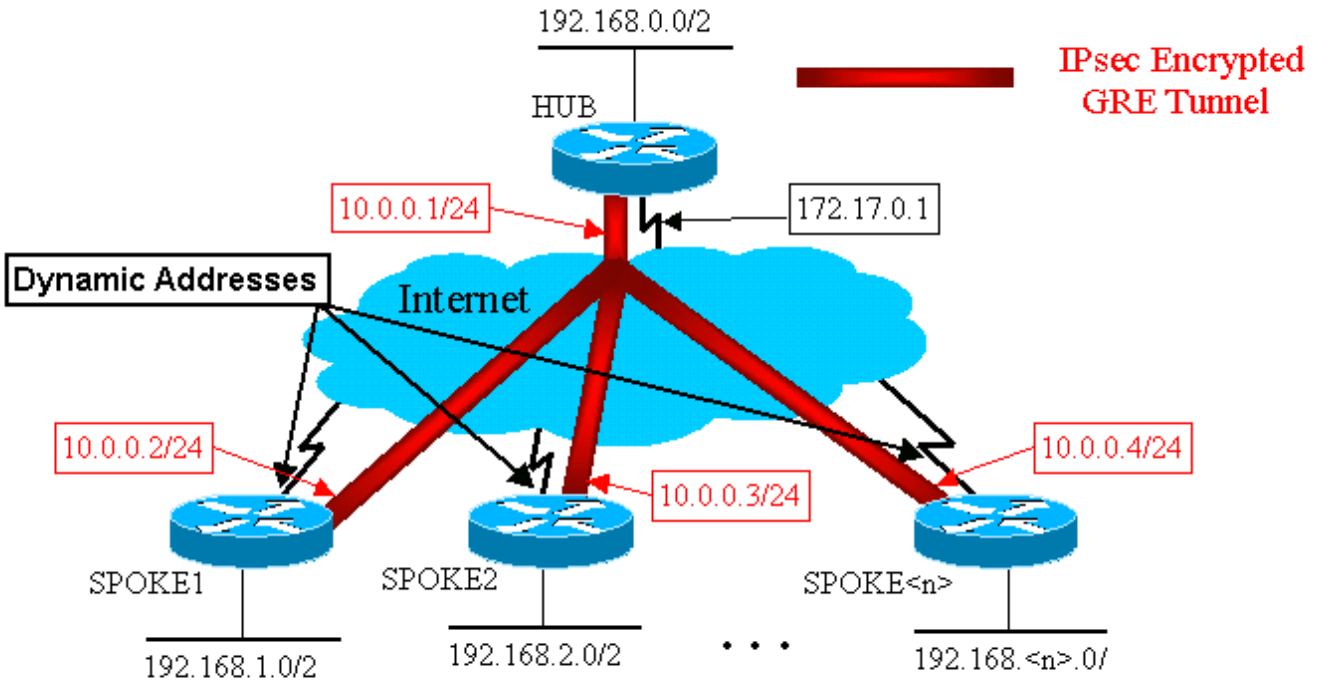
```
crypto map vpnmap1 10 IPsec-isakmp
 set peer 172.17.0.1
 set transform-set trans2
 set security-association level per-host
 match address 101
!
...
!
access-list 101 permit gre any host 172.17.0.1
```

새 스포크 컨피그레이션에서 사용되는 기능은 다음과 같습니다.

- GRE 터널 인터페이스가 작동하면 허브 라우터로 NHRP 등록 패킷을 전송하기 시작합니다.이러한 NHRP 등록 패킷은 IPsec을 트리거합니다.스포크 라우터에서 set peer <peer-address> 및 match **ip access-list <ACL>** 명령이 구성됩니다.ACL은 GRE를 프로토콜로, 소스에 대해 any, 대상에 대한 허브 IP 주소를 지정합니다.**참고:** 스포크 라우터의 IP 주소가 동적이므로 물리적 인터페이스가 활성화되기 전에는 알 수 없으므로 ACL에서 소스로 사용 중인 모든 항목이 해당된다는 점에 유의해야 합니다.동적 스포크 인터페이스 주소가 해당 서브넷 내의 주소로 제한될 경우 ACL의 소스에 IP 서브넷을 사용할 수 있습니다.
- set security-**association level per-host** 명령은 스포크 IPsec 프록시의 IP 소스가 ACL의 "any"가 아닌 스포크 현재 물리적 인터페이스 주소(/32)가 되도록 사용됩니다.ACL의 "any"가 IPsec 프록시의 소스로 사용된 경우, 다른 스포크 라우터가 이 허브로 IPsec+GRE 터널을 설정하는 것을 차단합니다.이는 허브의 결과 IPsec 프록시가 gre **호스트 172.17.0.1 any**를 **허용하는 것과** 동일하기 때문입니다.이는 모든 스포크로 향하는 모든 GRE 터널 패킷이 암호화되어 허브와 터널을 설정한 첫 번째 스포크로 전송된다는 의미입니다. IPsec 프록시가 모든 스포크의 GRE 패킷과 일치하기 때문입니다.
- IPsec 터널이 설정되면 NHRP 등록 패킷이 스포크 라우터에서 구성된 NHS(Next Hop Server)로 이동합니다. NHS는 이 허브 앤 스포크 네트워크의 허브 라우터입니다.NHRP 등록 패킷은 허브 라우터가 이 스포크 라우터에 대한 NHRP 매핑을 생성하는 데 필요한 정보를 제공합니다.이 매핑을 통해 허브 라우터는 mGRE+IPsec 터널을 통해 이 스포크 라우터에 유니캐스트 IP 데이터 패킷을 전달할 수 있습니다.또한 허브는 NHRP 멀티캐스트 매핑 목록에 스포크 라

우터를 추가합니다.그런 다음 허브는 동적 라우팅 프로토콜이 구성된 경우 스포크에 동적 IP 라우팅 멀티캐스트 패킷 전송을 시작합니다. 그런 다음 스포크는 허브의 라우팅 프로토콜 네이버가 되며 라우팅 업데이트를 교환합니다.

IPsec + mGRE 허브 및 스포크



허브 라우터

```
version 12.3
 !
 hostname Hub
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  no ip split-horizon eigrp 1
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
```

```
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
 ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
 no auto-summary
!
```

위 허브 구성에서는 스포크 라우터의 IP 주소가 구성되지 않았음을 확인합니다.스포크의 외부 물리적 인터페이스와 스포크의 터널 인터페이스 IP 주소에 대한 매핑은 NHRP를 통해 허브에서 동적으로 학습됩니다.이렇게 하면 스포크의 외부 물리적 인터페이스 IP 주소를 동적으로 할당할 수 있습니다.

Spoke1 라우터

```
version 12.3
!
hostname Spoke1
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto map vpnmap1 local-address Ethernet0
crypto map vpnmap1 10 IPsec-isakmp
 set peer 172.17.0.1
 set security-association level per-host
 set transform-set trans2
 match address 101
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
 tunnel key 100000
!
interface Ethernet0
 ip address dhcp hostname Spoke1
 crypto map vpnmap1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
```

```
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
 !
 access-list 101 permit gre 172.16.1.0 0.0.0.255 host
172.17.0.1
```

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto map vpnmap1 local-address Ethernet0
 crypto map vpnmap1 10 IPsec-isakmp
  set peer 172.17.0.1
  set security-association level per-host
  set transform-set trans2
  match address 101
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.3 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
 !
 interface Ethernet0
  ip address dhcp hostname Spoke2
  crypto map vpnmap1
 !
 interface Ethernet1
  ip address 192.168.2.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
 !
 access-list 101 permit gre 172.16.2.0 0.0.0.255 host
172.17.0.1
```

스포크 컨피그레이션에 대해 주목해야 할 주요 사항은 다음과 같습니다.

- 외부 물리적 인터페이스(ethernet0) IP 주소는 DHCP를 통해 동적입니다.**ip 주소 dhcp**

hostname spoke2

- 암호화 ACL(101)은 IPsec 프록시의 소스로 서브넷을 지정합니다.**access-list 101 permit gre 172.16.2.0 0.0.0.255 호스트 172.17.0.1**
- IPsec 암호화 맵의 다음 명령은 보안 연결이 호스트당 연결되도록 지정합니다.**호스트당 보안 연결 수준 설정**
- 모든 터널은 모두 허브 라우터의 동일한 멀티포인트 GRE 인터페이스를 통해 연결되므로 동일한 서브넷의 일부입니다.**ip 주소 10.0.0.2 255.255.255.0**

이 세 명령을 조합하면 스포크의 외부 물리적 인터페이스 IP 주소를 구성할 필요가 없습니다.사용되는 IPsec 프록시는 서브넷 기반 대신 호스트 기반 프록시입니다.

스포크 라우터의 구성에는 IPsec+GRE 터널을 시작해야 하므로 허브 라우터의 IP 주소가 구성되어 있습니다.Spoke1 및 Spoke2 구성 간의 유사성을 확인합니다.이 두 가지 구성은 비슷할 뿐만 아니라 모든 스포크 라우터 컨피그레이션도 유사합니다.대부분의 경우 모든 스포크는 인터페이스에 고유한 IP 주소를 필요로 하며 나머지 컨피그레이션은 동일합니다.따라서 많은 스포크 라우터를 신속하게 구성하고 구축할 수 있습니다.

NHRP 데이터는 허브 및 스포크의 다음과 같습니다.

허브 라우터

```
Hub#show ip nhrp
   10.0.0.2/32 via 10.0.0.2, Tunnel0 created 01:25:18,
expire 00:03:51
     Type: dynamic, Flags: authoritative unique
registered
     NBMA address: 172.16.1.4
   10.0.0.3/32 via 10.0.0.3, Tunnel0 created 00:06:02,
expire 00:04:03
     Type: dynamic, Flags: authoritative unique
registered
     NBMA address: 172.16.2.10
   ...
   10.0.0.<n>/32 via 10.0.0.<n>, Tunnel0 created
00:06:00, expire 00:04:25
     Type: dynamic, Flags: authoritative unique
registered
     NBMA address: 172.16.<n>.41
```

Spoke1 라우터

```
Spoke1#sho ip nhrp
   10.0.0.1/32 via 10.0.0.1, Tunnel0 created 4d08h,
never expire
     Type: static, Flags: authoritative
     NBMA address: 172.17.0.1
```

동적 멀티포인트 허브 및 스포크

위의 스포크 라우터의 컨피그레이션은 DMVPN 솔루션의 기능에 의존하지 않으므로 스포크 라우터는 12.2(13)T 이전 버전의 Cisco IOS 소프트웨어 버전을 실행할 수 있습니다.허브 라우터의 컨피그레이션은 DMVPN 기능을 사용하므로 Cisco IOS 버전 12.2(13)T 이상을 실행해야 합니다.이렇게 하면 이미 구축된 스포크 라우터를 언제 업그레이드해야 할지를 결정할 수 있습니다.스포크 라우터에서 Cisco IOS 버전 12.2(13)T 이상을 실행하는 경우 다음과 같이 스포크 컨피그레이션을 간소화할

수 있습니다.

스포크<n> 라우터(Cisco IOS 12.2(13)T 이전)

```
crypto map vpnmap1 10 IPsec-isakmp
  set peer 172.17.0.1
  set security-association level per-host
  set transform-set trans2
  match address 101
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.<n+1> 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
 !
 interface Ethernet0
  ip address dhcp hostname Spoke<n>
  crypto map vpnmap1
 !
 . . .
 !
 access-list 101 permit gre any host 172.17.0.1
```

스포크<n> 라우터(Cisco IOS 12.2(13)T 이후)

```
crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.<n+1> 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address dhcp hostname Spoke<n>
 !
```

다음 작업을 수행했습니다.

1. crypto map **vpnmap1 10ipsec-isakmp** 명령을 제거하고 **crypto ipsec profile vpnprof**로 **교체했습니다**.
2. Ethernet0 인터페이스**에서 crypto map vpnmap1** 명령을 제거하고 **tunnel protection ipsec profile vpnprof** 명령을 Tunnel0 인터페이스에 놓습니다.
3. crypto ACL을 제거했으며, access-list **101 permit gre any host 172.17.0.1을 제거했습니다**.

이 경우 IPsec 피어 주소와 프록시는 **터널 소스 ...** 및 **터널 대상 ...** 컨피그레이션에서 자동으로 파생됩니다.피어 및 프록시는 다음과 같습니다(**show crypto ipsec sa** 명령의 출력에 표시됨).

```
...
local ident (addr/mask/prot/port):    (172.16.1.24/255.255.255.255/47/0)
remote ident (addr/mask/prot/port):   (172.17.0.1/255.255.255.255/47/0)
...
local crypto endpt.: 172.17.1.24, remote crypto endpt.:172.17.0.1
...
```

요약하면, 다음 전체 컨피그레이션에는 기본 컨피그레이션(IPsec+GRE 허브 및 스포크)에서 이 시점까지 변경된 모든 내용이 포함됩니다.

허브 라우터

```
version 12.3
 !
 hostname Hub
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  no ip split-horizon eigrp 1
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address 172.17.0.1 255.255.255.0
 !
 interface Ethernet1
  ip address 192.168.0.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
```

```
  network 192.168.0.0 0.0.0.255
  no auto-summary
 !
```

허브 구성에 변경 사항이 없습니다.

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.2 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address dhcp hostname Spoke2
 !
 interface Ethernet1
  ip address 192.168.1.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
 !
```

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
```

```
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.3 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke2
!
interface Ethernet1
 ip address 192.168.2.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.2.0 0.0.0.255
 no auto-summary
!
```

동적 멀티포인트 IPsec VPN

이 섹션의 개념과 컨피그레이션에는 DMVPN의 모든 기능이 나와 있습니다.NHRP는 스포크 라우터가 VPN 네트워크에 있는 다른 스포크 라우터의 외부 물리적 인터페이스 주소를 동적으로 학습할 수 있는 기능을 제공합니다.즉, 스포크 라우터에는 IPsec+mGRE 터널을 다른 스포크 라우터에 직접 빌드할 수 있는 충분한 정보가 있습니다.이 스포크 투 스포크 데이터 트래픽이 허브 라우터를 통해 전송된 경우 암호화/해독해야 하므로 이 값이 유리합니다. 그러면 허브 라우터의 지연 및 로드가 두 번 증가합니다.이 기능을 사용하려면 스포크 라우터를 point-to-point GRE(p-pGRE)에서 mGRE(multipoint GRE) 터널 인터페이스로 전환해야 합니다.또한 다른 스포크 라우터의 터널 IP 주소의 IP next-hop을 사용하여 다른 스포크 뒤에서 사용할 수 있는 (하위) 네트워크를 학습해야 합니다.스포크 라우터는 허브와 함께 IPsec+mGRE 터널을 통해 실행되는 동적 IP 라우팅 프로토콜을 통해 이러한(하위) 네트워크를 학습합니다.

허브 라우터에서 실행되는 동적 IP 라우팅 프로토콜은 한 스포크에서 학습된 경로를 동일한 인터페이스로 다시 다른 모든 스포크에 반영하도록 구성할 수 있지만, 이러한 경로의 IP next-hop은 대개 허브 라우터가 되고 허브가 이 경로를 학습한 스포크 라우터가 아닙니다.

참고: 동적 라우팅 프로토콜은 허브 및 스포크 링크에서만 실행되며 동적 스포크 투 스포크 링크에서는 실행되지 않습니다.

허브 라우터에서 동적 라우팅 프로토콜(RIP, OSPF 및 EIGRP)을 구성하여 mGRE 터널 인터페이스를 통해 경로를 다시 알리고, 경로가 다른 스포크로 다시 광고될 때 한 스포크에서 학습된 경로를 위해 IP next-hop을 원래 스포크 라우터로 설정해야 합니다.

다음은 라우팅 프로토콜 컨피그레이션의 요구 사항입니다.

RIP

허브의 mGRE 터널 인터페이스에서 split horizon을 해제해야 합니다. 그렇지 않으면 RIP는 mGRE 인터페이스를 통해 학습된 경로를 동일한 인터페이스로 다시 광고하지 않습니다.

```
no ip split-horizon
```

다른 변경은 필요하지 않습니다.RIP는 원래 IP next-hop을 해당 경로를 학습한 동일한 인터페이스로 다시 광고하는 경로에 자동으로 사용합니다.

EIGRP

허브의 mGRE 터널 인터페이스에서 split horizon을 해제해야 합니다. 그렇지 않으면 EIGRP는 mGRE 인터페이스를 통해 학습된 경로를 동일한 인터페이스로 다시 광고하지 않습니다.

```
no ip split-horizon eigrp
```

EIGRP는 기본적으로 IP next-hop을 광고하는 경로의 허브 라우터로 설정합니다. 이러한 경로를 학습한 동일한 인터페이스로 다시 광고하는 경우에도 마찬가지입니다.따라서 이 경우 EIGRP가 이러한 경로를 광고할 때 원래 IP next-hop을 사용하도록 하려면 다음 컨피그레이션 명령이 필요합니다.

```
no ip next-hop-self eigrp
```

참고: no ip next-hop-self eigrp <as> 명령은 Cisco IOS 릴리스 12.3(2)부터 사용할 수 있습니다. 12.2(13)T와 12.3(2) 사이의 Cisco IOS 릴리스의 경우 다음을 수행해야 합니다.

- 스포크 투 스포크 동적 터널을 원하지 않으면 위의 명령이 필요하지 않습니다.
- 스포크 투 스포크 동적 터널이 필요한 경우 스포크 라우터의 터널 인터페이스에서 프로세스 스위칭을 사용해야 합니다.
- 그렇지 않으면 DMVPN을 통해 다른 라우팅 프로토콜을 사용해야 합니다.

OSPF

OSPF는 링크 상태 라우팅 프로토콜이므로 분할 영역 문제는 없습니다.일반적으로 다중 지점 인터페이스의 경우 OSPF 네트워크 유형을 point-to-multipoint로 구성하지만, 이로 인해 OSPF가 스포크 라우터의 라우팅 테이블에 호스트 경로를 추가할 수 있습니다.이러한 호스트 경로는 다른 스포크 라우터 뒤에 있는 네트워크로 향하는 패킷이 허브를 통해 전달되고 다른 스포크로 직접 전달되도록

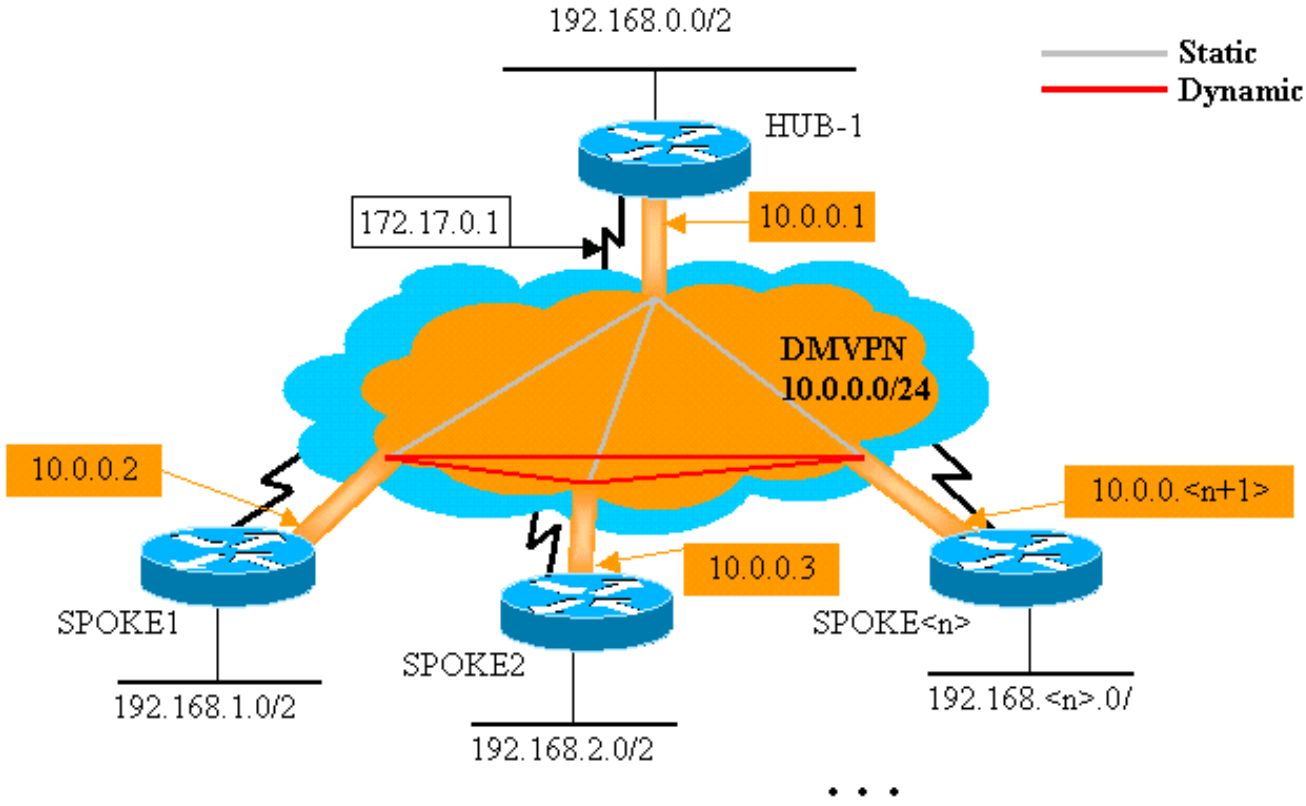
합니다.이 문제를 해결하려면 명령을 사용하여 브로드캐스트할 OSPF 네트워크 유형을 구성합니다
.

ip ospf network broadcast

또한 허브 라우터가 IPsec+mGRE 네트워크의 DR(Designated Router)이 되도록 해야 합니다.이 작업은 허브에서 OSPF 우선순위를 1보다 크게 설정하고 스포크에서 0을 설정합니다.

- 허브:**ip ospf 우선순위 2**
- 스포크:**ip ospf 우선순위 0**

DMVPN 단일 허브



허브 라우터

```
version 12.3
 !
 hostname Hub
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
```

```
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  ip ospf network broadcast
  ip ospf priority 2
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address 172.17.0.1 255.255.255.0
 !
 interface Ethernet1
  ip address 192.168.0.1 255.255.255.0
 !
 router ospf 1
  network 10.0.0.0 0.0.0.255 area 0
  network 192.168.0.0 0.0.0.255 area 0
 !
```

허브 컨피그레이션의 유일한 변경 사항은 OSPF가 EIGRP 대신 라우팅 프로토콜이라는 것입니다 .OSPF 네트워크 유형은 브로드캐스트로 설정되고 우선순위는 2로 설정됩니다. OSPF 네트워크 유형을 브로드캐스트로 설정하면 OSPF가 스포크 라우터의 GRE 터널 주소로 IP next-hop 주소가 있는 스포크 라우터 뒤의 네트워크에 대한 경로를 설치합니다.

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.2 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  ip ospf network broadcast
  ip ospf priority 0
  delay 1000
```

```
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
```

스포크 라우터의 컨피그레이션은 이제 허브의 컨피그레이션과 매우 유사합니다.차이점은 다음과 같습니다.

- OSPF 우선순위는 0으로 설정됩니다. 스포크 라우터는 mGRE NBMA(Nonbroadcast Multiaccess) 네트워크의 DR이 될 수 없습니다.허브 라우터만 모든 스포크 라우터에 직접 정적 연결을 가집니다.DR은 NBMA 네트워크의 모든 구성원에 액세스할 수 있어야 합니다.
- 허브 라우터에 대해 구성된 NHRP 유니캐스트 및 멀티캐스트 매핑이 있습니다.

  ```
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ```

  이전 컨피그레이션에서는 **ip nhrp map multicast ...** GRE 터널이 point-to-point이므로 명령이 필요하지 않았습니다.이 경우 멀티캐스트 패킷은 터널을 통해 가능한 단일 대상으로 자동으로 캡슐화됩니다.이제 스포크 GRE 터널이 멀티포인트로 변경되었으며 가능한 대상이 두 개 이상 있으므로 이 명령이 필요합니다.
- 스포크 라우터가 나타나면 허브 라우터가 스포크 라우터에 대한 정보로 구성되어 있지 않으며 스포크 라우터에 동적으로 IP 주소가 할당되었을 수 있으므로 허브와의 터널 연결을 시작해야 합니다.스포크 라우터는 허브를 NHRP NHS로 구성합니다.
  `ip nhrp nhs 10.0.0.1`
  위의 명령을 사용하면 스포크 라우터는 mGRE+IPsec 터널을 통해 NHRP 등록 패킷을 정기적으로 허브 라우터로 전송합니다.이러한 등록 패킷은 허브 라우터에서 패킷을 다시 스포크 라우터로 터널링하는 데 필요한 스포크 NHRP 매핑 정보를 제공합니다.

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
```

```
 ip address 10.0.0.3 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast 172.17.0.1
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip ospf network broadcast
 ip ospf priority 0
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.3.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
```

스포크<n> 라우터

```
version 12.3
!
hostname Spoke<n>
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.<n+1> 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast 172.17.0.1
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip ospf network broadcast
 ip ospf priority 0
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke<n>
```

```
!
interface Ethernet1
 ip address 192.168.<n>.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.




!
```

모든 스포크 라우터의 컨피그레이션이 매우 유사합니다.유일한 차이점은 로컬 인터페이스의 IP 주소입니다.이렇게 하면 많은 수의 스포크 라우터를 구축할 때 도움이 됩니다.모든 스포크 라우터는 동일하게 구성할 수 있으며 로컬 IP 인터페이스 주소만 추가해야 합니다.

이 시점에서 Hub, Spoke1 및 Spoke2 라우터의 라우팅 테이블과 NHRP 매핑 테이블을 살펴보고 초기 조건(Spoke1 및 Spoke2 라우터가 올라간 바로 후)과 Spoke1 및 Spoke2 이후 Spoke1 및 Spoke2 사이에 동적 링크가 생성된 조건을 확인합니다.

초기 조건

허브 라우터 정보

```
Hub#show ip route
      172.17.0.0/24 is subnetted, 1 subnets
 C       172.17.0.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 C    192.168.0.0/24 is directly connected, Ethernet1
 O    192.168.1.0/24 [110/2] via 10.0.0.2, 00:19:53,
Tunnel0
 O    192.168.2.0/24 [110/2] via 10.0.0.3, 00:19:53,
Tunnel0
 Hub#show ip nhrp
 10.0.0.2/32 via 10.0.0.2, Tunnel0 created 00:57:27,
expire    00:04:13
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.1.24
 10.0.0.3/32 via 10.0.0.3, Tunnel0 created 07:11:25,
expire    00:04:33
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.2.75
 Hub#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
  204 Ethernet0   172.17.0.1     set  HMAC_SHA+DES_56_CB
0       0
  205 Ethernet0   172.17.0.1     set  HMAC_SHA+DES_56_CB
0       0
 2628 Tunnel0     10.0.0.1       set  HMAC_MD5
0     402
 2629 Tunnel0     10.0.0.1       set  HMAC_MD5
357       0
 2630 Tunnel0     10.0.0.1       set  HMAC_MD5
0     427
 2631 Tunnel0     10.0.0.1       set  HMAC_MD5
```

```
308       0
```

Spoke1 라우터 정보

```
Spoke1#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.1.24 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 O    192.168.0.0/24 [110/2] via 10.0.0.1, 00:31:46,
Tunnel0
 C    192.168.1.0/24 is directly connected, Ethernet1
 O    192.168.2.0/24 [110/2] via 10.0.0.3, 00:31:46,
Tunnel0
 Spoke1#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 01:42:00,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 Spoke1#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
    2 Ethernet0   172.16.1.24    set  HMAC_SHA+DES_56_CB
0       0
 2064 Tunnel0     10.0.0.2       set  HMAC_MD5
0     244
 2065 Tunnel0     10.0.0.2       set  HMAC_MD5
276       0
```

Spoke2 라우터 정보

```
Spoke2#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.2.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 O    192.168.0.0/24 [110/2] via 10.0.0.1, 00:38:52,
Tunnel0
 O    192.168.1.0/24 [110/2] via 10.0.0.2, 00:38:52,
Tunnel0
 C    192.168.2.0/24 is directly connected, Ethernet1
 Spoke2#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 01:32:10,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 Spoke2#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
   17 Ethernet0   172.16.2.75    set  HMAC_SHA+DES_56_CB
0       0
 2070 Tunnel0     10.0.0.3       set  HMAC_MD5
0     279
 2071 Tunnel0     10.0.0.3       set  HMAC_MD5
316       0
```

이 시점에서는 192.168.1.2에서 192.168.2.3으로 ping합니다. 이 주소는 각각 Spoke1 및 Spoke2 라우터 뒤의 호스트에 대한 주소입니다.직접 스포크 대 스포크 mGRE+IPsec 터널을 구축하는 데 다음 이벤트 시퀀스가 발생합니다.

1. Spoke1 라우터는 목적지 192.168.2.3과 함께 Ping 패킷을 수신합니다. 라우팅 테이블에서 이 목적지를 조회하고 이 패킷을 Tunnel0 인터페이스에서 IP Nexthop, 10.0.0.3으로 전달해야 함을 확인합니다.
2. Spoke1 라우터는 대상 10.0.0.3에 대한 NHRP 매핑 테이블을 확인하고 항목이 없음을 확인합니다.Spoke1 라우터는 NHRP 확인 요청 패킷을 생성하여 NHS(허브 라우터)에 전송합니다.
3. 허브 라우터는 대상 10.0.0.3에 대한 NHRP 매핑 테이블을 확인하고 주소 172.16.2.75에 매핑됩니다. 허브 라우터는 NHRP 확인 응답 응답 패킷을 생성하여 Spoke1 라우터에 전송합니다.
4. Spoke1 라우터는 NHRP 확인 응답을 수신하고 NHRP 매핑 테이블에 10.0.0.3 —>172.16.2.75 매핑을 입력합니다.NHRP 매핑을 추가하면 IPsec이 피어 172.16.2.75과 함께 IPsec 터널을 시작하도록 트리거됩니다.
5. Spoke1 라우터는 172.16.2.75으로 ISAKMP를 시작하고 ISAKMP 및 IPsec SA를 협상합니다 .IPsec 프록시는 Tunnel0 **tunnel source <address>** 명령과 NHRP 매핑에서 파생됩니다.

   ```
   local ident (addr/mask/prot/port): (172.16.1.24/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.16.2.75/255.255.255.255/47/0)
   ```

6. IPsec 터널이 구축되면 192.168.2.0/24 서브넷에 대한 모든 추가 데이터 패킷이 Spoke2로 직접 전송됩니다.
7. 192.168.2.3으로 향하는 패킷이 호스트로 전달되면 이 호스트는 반환 패킷을 192.168.1.2으로 보냅니다. Spoke2 라우터가 192.168.1.2으로 향하는 이 패킷을 수신하면 라우팅 테이블에서 이 대상을 조회하고 이 패킷을 Tunnel0 인터페이스에서 IP next-hop, 10.0.0.2으로 전달해야 함을 확인합니다.
8. Spoke2 라우터는 대상 10.0.0.2에 대한 NHRP 매핑 테이블을 확인하고 항목이 없음을 확인합니다.Spoke2 라우터는 NHRP 확인 요청 패킷을 생성하여 NHS(허브 라우터)에 전송합니다.
9. 허브 라우터는 대상 10.0.0.2에 대한 NHRP 매핑 테이블을 확인하고 주소 172.16.1.24에 매핑됩니다. 허브 라우터는 NHRP 확인 응답 응답 패킷을 생성하여 Spoke2 라우터에 전송합니다.
10. Spoke2 라우터는 NHRP 확인 응답을 수신하고 NHRP 매핑 테이블에 10.0.0.2 —> 172.16.1.24 매핑을 입력합니다.NHRP 매핑을 추가하면 IPsec이 피어 172.16.1.24과 IPsec 터널을 시작하도록 트리거되지만 피어 172.16.1.24이 있는 IPsec 터널이 이미 있으므로 더 이상 수행할 필요가 없습니다.
11. Spoke1 및 Spoke2는 이제 패킷을 서로 직접 전달할 수 있습니다.NHRP 매핑이 대기 시간 동안 패킷을 전달하는 데 사용되지 않으면 NHRP 매핑이 삭제됩니다.NHRP 매핑 엔트리를 삭제하면 IPsec에서 이 직접 링크에 대한 IPsec SA를 삭제합니다.

spoke1과 spoke2 간에 동적 링크가 생성된 후의 조건

Spoke1 라우터 정보

```
Spoke1#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 02:34:16,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.0.3/32 via 10.0.0.3, Tunnel0 created 00:00:05,
expire 00:03:35
   Type: dynamic, Flags: router unique used
   NBMA address: 172.16.2.75
 Spoke1#show crypto engine connection active
  ID  Interface   IP-Address     State Algorithm
Encrypt Decrypt
    2 Ethernet0   172.16.1.24     set  HMAC_SHA+DES_56_CB
```

```
0        0
    3 Ethernet0   172.16.1.24     set  HMAC_SHA+DES_56_CB
0        0
 2064 Tunnel0     10.0.0.2        set  HMAC_MD5
0      375
 2065 Tunnel0     10.0.0.2        set  HMAC_MD5
426        0
 2066 Tunnel0     10.0.0.2        set  HMAC_MD5
0       20
 2067 Tunnel0     10.0.0.2        set  HMAC_MD5
19         0
```

Spoke2 라우터 정보

```
Spoke2#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 02:18:25,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.0.2/32 via 10.0.0.2, Tunnel0 created 00:00:24,
expire 00:04:35
   Type: dynamic, Flags: router unique used
   NBMA address: 172.16.1.24
 Spoke2#show crypto engine connection active
  ID  Interface   IP-Address     State Algorithm
Encrypt Decrypt
   17 Ethernet0   172.16.2.75     set  HMAC_SHA+DES_56_CB
0        0
   18 Ethernet0   172.16.2.75     set  HMAC_SHA+DES_56_CB
0        0
 2070 Tunnel0     10.0.0.3        set  HMAC_MD5
0      407
 2071 Tunnel0     10.0.0.3        set  HMAC_MD5
460        0
 2072 Tunnel0     10.0.0.3        set  HMAC_MD5
0       19
 2073 Tunnel0     10.0.0.3        set  HMAC_MD5
20         0
```

위의 출력에서 Spoke1 및 Spoke2가 허브 라우터에서 상호 NHRP 매핑을 가져오고 mGRE+IPsec 터널을 구축 및 사용했음을 확인할 수 있습니다.NHRP 매핑은 5분 후 만료됩니다(NHRP holdtime의 현재 값 = 300초). NHRP 매핑이 만료되기 전 마지막 분 내에 사용된 경우 NHRP 해결 요청 및 회신이 전송되어 항목이 삭제되기 전에 새로 고칩니다.그렇지 않으면 NHRP 매핑이 삭제되고 IPsec이 IPsec SA를 지우는 데 트리거됩니다.

듀얼 허브를 사용하는 동적 멀티포인트 IPsec VPN

스포크 라우터에 대한 몇 가지 추가 구성 라인을 사용하여 이중화를 위해 듀얼(또는 다중) 허브 라우터를 설정할 수 있습니다.듀얼 허브 DMVPN을 구성하는 방법에는 두 가지가 있습니다.

- 단일 다중 지점 GRE 터널 인터페이스를 사용하고 두 개의 다른 허브를 NHS(Next-Hop-Server)로 가리키는 각 스포크가 있는 단일 DMVPN 네트워크.허브 라우터에는 단일 멀티포인트 GRE 터널 인터페이스만 있습니다.
- 각 스포크가 포함된 듀얼 DMVPN 네트워크에는 GRE 터널 인터페이스(포인트-투-포인트 또는 멀티포인트)가 2개 있으며 각 GRE 터널은 다른 허브 라우터에 연결됩니다.또한 허브 라우터에는 단일 멀티포인트 GRE 터널 인터페이스만 있습니다.

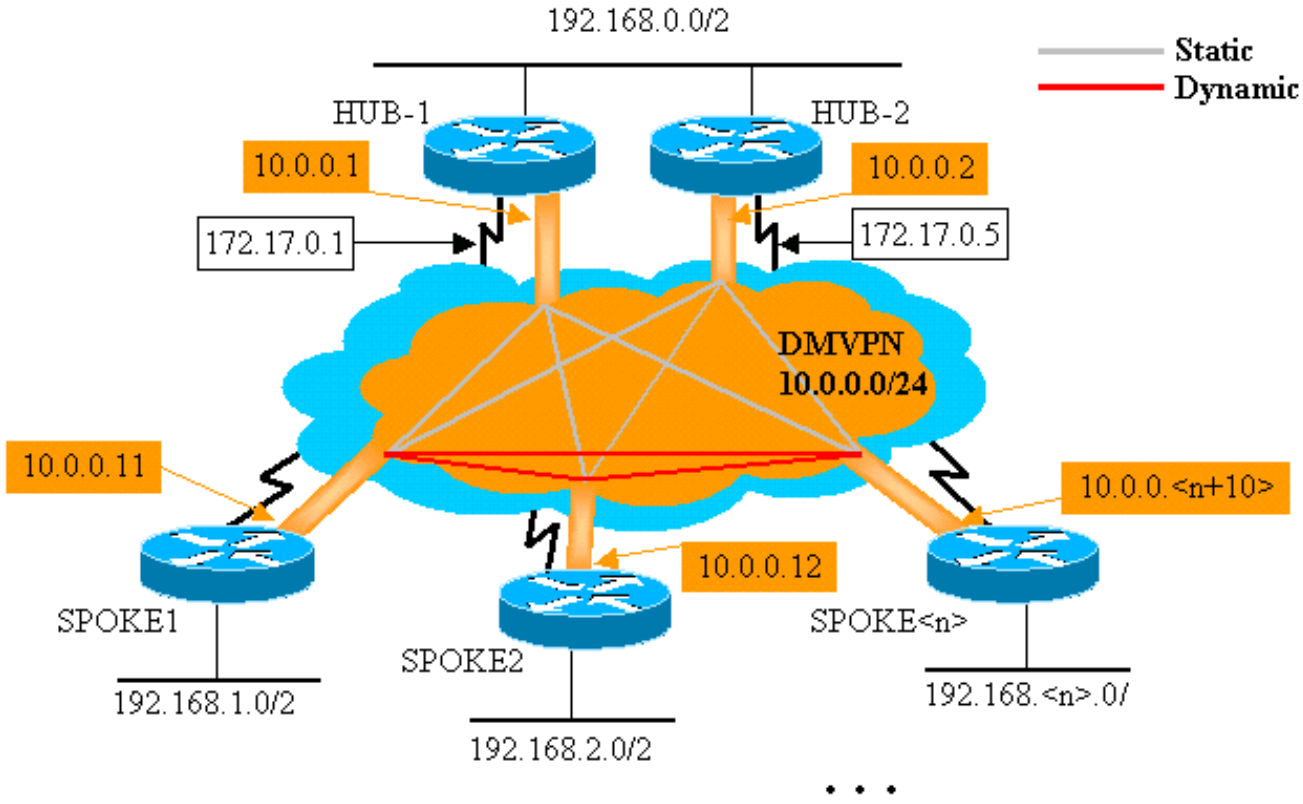
다음 예에서는 듀얼 허브 DMVPN에 대해 이러한 두 가지 시나리오를 구성하는 방법을 살펴봅니다 .두 경우 모두 강조 표시된 차이점은 DMVPN 단일 허브 컨피그레이션과 관련이 있습니다.

듀얼 허브 - 단일 DMVPN 레이아웃

단일 DMVPN 레이아웃이 있는 듀얼 허브는 설정하기가 매우 쉽지만 듀얼 DMVPN 레이아웃이 있는 듀얼 허브와 달리 DMVPN 전체의 라우팅을 제어할 수 있는 권한은 없습니다.이 경우 모든 허브(이 경우 2개)와 모든 스포크가 이 단일 서브넷("클라우드")에 연결된 단일 DMVPN "클라우드"를 사용하는 것이 좋습니다. 스포크에서 허브에 대한 고정 NHRP 매핑은 동적 라우팅 프로토콜이 실행될 고정 IPsec+mGRE 링크를 정의합니다.동적 라우팅 프로토콜은 스포크 간 동적 IPsec+mGRE 링크를 통해 실행되지 않습니다.스포크 라우터는 동일한 mGRE 터널 인터페이스를 통해 허브 라우터가 있는 라우팅 네이버이므로 링크 또는 인터페이스 차이(메트릭, 비용, 지연 또는 대역폭 등)를 사용하여 둘 다 작동 중일 때 다른 허브에 대해 하나의 허브를 선호하도록 동적 라우팅 프로토콜 메트릭을 수정할 수 없습니다.이 기본 설정이 필요한 경우 라우팅 프로토콜 컨피그레이션에 대한 내부 기술을 사용해야 합니다.따라서 동적 라우팅 프로토콜에 OSPF가 아닌 EIGRP 또는 RIP를 사용하는 것이 좋습니다.

참고: 위의 문제는 일반적으로 허브 라우터가 함께 있는 경우에만 문제가 됩니다.둘 중 하나의 허브 라우터를 통해 목적지 네트워크에 연결할 수 있더라도 일반 동적 라우팅은 올바른 허브 라우터를 선호하게 됩니다.

듀얼 허브 - 단일 DMVPN 레이아웃



허브 라우터
version 12.3

```
!
hostname Hub1
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.1 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp holdtime 600
 ip ospf network broadcast
 ip ospf priority 2
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.0
!
interface Ethernet1
 ip address 192.168.0.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 1
 network 192.168.0.0 0.0.0.255 area 0
!
```

허브2 라우터

```
version 12.3
!
hostname Hub2
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 900
 ip address 10.0.0.2 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
```

```
 ip nhrp map  multicast 172.17.0.1
 ip nhrp map multicast dynamic
 ip nhrp network-id 100000
 ip nhrp holdtime 600
 ip nhrp nhs 10.0.0.1
 ip ospf network broadcast
 ip ospf priority 1
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.5 255.255.255.0
!
interface Ethernet1
 ip address 192.168.0.2 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 1
 network 192.168.0.0 0.0.0.255 area 0
!
```

Hub1 컨피그레이션의 유일한 변경 사항은 OSPF가 두 영역을 사용하도록 변경하는 것입니다.영역 0은 두 허브 뒤의 네트워크에 사용되며, 영역 1은 스포크 라우터 뒤의 DMVPN 네트워크 및 네트워크에 사용됩니다.OSPF는 단일 영역을 사용할 수 있지만, 여기에서 두 영역을 사용하여 여러 OSPF 영역에 대한 컨피그레이션을 보여 줍니다.

Hub2에 대한 컨피그레이션은 기본적으로 적절한 IP 주소가 변경되는 Hub1 컨피그레이션과 동일합니다.한 가지 주요 차이점은 Hub2는 Hub1의 스포크(또는 클라이언트)이므로 Hub1은 기본 허브로, Hub2는 보조 허브로 만듭니다.이렇게 하면 Hub2가 mGRE 터널을 통해 Hub1이 있는 OSPF 인접 디바이스로 사용됩니다.Hub1은 OSPF DR이므로 mGRE 인터페이스(NBMA 네트워크)를 통해 다른 모든 OSPF 라우터와 직접 연결해야 합니다. Hub1과 Hub2 간의 직접 링크가 없으면 Hub2는 Hub1도 작동하면 OSPF 라우팅에 참여하지 않습니다.Hub1이 다운되면 Hub2는 DMVPN(NBMA 네트워크)의 OSPF DR이 됩니다. Hub1이 복구되면 DMVPN의 OSPF DR이 됩니다.

Hub1 및 Hub2 뒤의 라우터는 Hub1을 사용하여 패킷을 스포크 네트워크로 전송합니다. GRE 터널 인터페이스의 대역폭은 Hub2에서 1000Kb/sec로 설정되고 900Kb/sec는 Hub2로 설정되기 때문입니다. 이와 달리 스포크 라우터는 허브 라우터 뒤에 있는 네트워크에 대한 패킷을 Hub1 및 Hub2로 전송합니다. 각 스포크에 하나의 mGRE 터널 인터페이스만 있고 동일한 비용 경로가 2개 있기 때문입니다.패킷당 로드 밸런싱을 사용하는 경우 순서가 잘못된 패킷이 발생할 수 있습니다.

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
```

```
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.11 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast 172.17.0.1
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp map multicast 172.17.0.5
 ip nhrp map 10.0.0.2 172.17.0.5
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip nhrp nhs 10.0.0.2
 ip ospf network broadcast
 ip ospf priority 0
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke1
!
interface Ethernet1
 ip address 192.168.1.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 1
 network 192.168.1.0 0.0.0.255 area 1
!
```

스포크 라우터의 컨피그레이션 차이는 다음과 같습니다.

- 새 컨피그레이션에서 스포크는 Hub2 및 Hub2에 대한 고정 NHRP 매핑으로 구성되며 다음 hop 서버로 추가됩니다.원본:

  ```
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp nhs 10.0.0.1
  ```

  새로 만들기:

  ```
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp map multicast 172.17.0.5
  ip nhrp map 10.0.0.2 172.17.0.5
  ip nhrp nhs 10.0.0.1
  ip nhrp nhs 10.0.0.2
  ```

- 스포크 라우터의 OSPF 영역이 영역 1로 변경되었습니다.

허브에 대한 스포크 라우터에서 고정 NHRP 매핑 및 NHS를 정의하면 이 터널을 통해 동적 라우팅 프로토콜을 실행할 수 있습니다.허브 및 스포크 라우팅 또는 네이버 네트워크를 정의합니다 .Hub2는 모든 스포크의 허브이며 Hub1의 스포크이기도 합니다. 이렇게 하면 DMVPN 솔루션을 사용할 때 멀티레이어 허브-스포크 네트워크를 쉽게 설계, 구성 및 수정할 수 있습니다.

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.12 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast 172.17.0.1
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp map multicast 172.17.0.5
  ip nhrp map 10.0.0.2 172.17.0.5
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  ip nhrp nhs 10.0.0.2
  ip ospf network broadcast
  ip ospf priority 0
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address dhcp hostname Spoke1
 !
 interface Ethernet1
  ip address 192.168.2.1 255.255.255.0
 !
 router ospf 1
  network 10.0.0.0 0.0.0.255 area 0
  network 192.168.2.0 0.0.0.255 area 0
 !
```

스포크<n> 라우터

```
version 12.3
 !
 hostname Spoke<n>
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
```

```
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.<n+10> 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast 172.17.0.1
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp map multicast 172.17.0.5
 ip nhrp map 10.0.0.2 172.17.0.5
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 ip nhrp nhs 10.0.0.2
 ip ospf network broadcast
 ip ospf priority 0
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke<x>
!
interface Ethernet1
 ip address 192.168.<n>.1 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 network 192.168.




!
```

이때 Hub1, Hub2, Spoke1 및 Spoke2 라우터의 라우팅 테이블, NHRP 매핑 테이블 및 IPsec 연결을 확인하여 초기 조건을 확인할 수 있습니다(Spoke1 및 Spoke2 라우터가 올라간 바로 후).

초기 조건 및 변경 사항

Hub1 라우터 정보

```
Hub1#show ip route
      172.17.0.0/24 is subnetted, 1 subnets
 C       172.17.0.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 C    192.168.0.0/24 is directly connected, Ethernet1
 O    192.168.1.0/24 [110/2] via 10.0.0.11, 00:02:17,
Tunnel0
 O    192.168.2.0/24 [110/2] via 10.0.0.12, 00:02:17,
Tunnel0
 Hub1#show ip nhrp
 10.0.0.2/32 via 10.0.0.2, Tunnel0 created 1w3d, expire
00:03:15
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.17.0.5
```

```
 10.0.0.11/32 via 10.0.0.11, Tunnel0 created 1w3d,
expire 00:03:49
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.1.24
 10.0.0.12/32 via 10.0.0.12, Tunnel0 created 1w3d,
expire 00:04:06
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.2.75
 Hub1#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
    4 Ethernet0   172.17.0.1     set  HMAC_SHA+DES_56_CB
0       0
    5 Ethernet0   172.17.0.1     set  HMAC_SHA+DES_56_CB
0       0
    6 Ethernet0   172.17.0.1     set  HMAC_SHA+DES_56_CB
0       0
 3532 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
0     232
 3533 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
212       0
 3534 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
0      18
 3535 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
17        0
 3536 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
0       7
 3537 Tunnel0     10.0.0.1       set  HMAC_MD5+DES_56_CB
7       0
```

Hub2 라우터 정보

```
Hub2#show ip route
      172.17.0.0/24 is subnetted, 1 subnets
 C       172.17.0.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 C    192.168.0.0/24 is directly connected, Ethernet1
 O    192.168.1.0/24 [110/2] via 10.0.0.11, 00:29:15,
Tunnel0
 O    192.168.2.0/24 [110/2] via 10.0.0.12, 00:29:15,
Tunnel0
 Hub2#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 1w3d, never
expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.0.11/32 via 10.0.0.11, Tunnel0 created 1w3d,
expire 00:03:15
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.1.24
 10.0.0.12/32 via 10.0.0.12, Tunnel0 created 00:46:17,
expire 00:03:51
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.2.75
 Hub2#show crypto engine connection active
   ID Interface   IP-Address    State Algorithm
Encrypt Decrypt
    4 Ethernet0   171.17.0.5     set  HMAC_SHA+DES_56_CB
0       0
    5 Ethernet0   171.17.0.5     set  HMAC_SHA+DES_56_CB
0       0
```

```
    6 Ethernet0   171.17.0.5      set  HMAC_SHA+DES_56_CB
0        0
 3520 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
0      351
 3521 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
326        0
 3522 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
0      311
 3523 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
339        0
 3524 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
0       25
 3525 Tunnel0     10.0.0.2        set  HMAC_MD5+DES_56_CB
22         0
```

Spoke1 라우터 정보

```
Spoke1#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.1.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 O IA 192.168.0.0/24 [110/11] via 10.0.0.1, 00:39:31,
Tunnel0
                     [110/11] via 10.0.0.2, 00:39:31,
Tunnel0
 C    192.168.1.0/24 is directly connected, Ethernet1
 O    192.168.2.0/24 [110/2] via 10.0.0.12, 00:37:58,
Tunnel0
 Spoke1#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 00:56:40,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.0.2/32 via 10.0.0.2, Tunnel0 created 00:56:40,
never expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.5
 Spoke1#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
    1 Ethernet0   172.16.1.24    set  HMAC_SHA+DES_56_CB
0       0
    2 Ethernet0   172.16.1.24    set  HMAC_SHA+DES_56_CB
0       0
 2010 Tunnel0     10.0.0.11      set  HMAC_MD5+DES_56_CB
0     171
 2011 Tunnel0     10.0.0.11      set  HMAC_MD5+DES_56_CB
185       0
 2012 Tunnel0     10.0.0.11      set  HMAC_MD5+DES_56_CB
0      12
 2013 Tunnel0     10.0.0.11      set  HMAC_MD5+DES_56_CB
13        0
```

Spoke2 라우터 정보

```
Spoke2#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.2.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
```

```
 O IA 192.168.0.0/24 [110/11] via 10.0.0.1, 00:57:56,
Tunnel0
                     [110/11] via 10.0.0.2, 00:57:56,
Tunnel0
 O    192.168.1.0/24 [110/2] via 10.0.0.11, 00:56:14,
Tunnel0
 C    192.168.2.0/24 is directly connected, Ethernet1
 Spoke2#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 5w6d, never
expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.0.2/32 via 10.0.0.2, Tunnel0 created 6w6d, never
expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.5
 Spoke2#show crypto engine connection active
  ID  Interface   IP-Address    State Algorithm
Encrypt Decrypt
    2 Ethernet0   172.16.2.75    set  HMAC_SHA+DES_56_CB
0       0
    3 Ethernet0   172.16.2.75    set  HMAC_SHA+DES_56_CB
0       0
 3712 Tunnel0     10.0.0.12      set  HMAC_MD5+DES_56_CB
0     302
 3713 Tunnel0     10.0.0.12      set  HMAC_MD5+DES_56_CB
331       0
 3716 Tunnel0     10.0.0.12      set  HMAC_MD5+DES_56_CB
0     216
 3717 Tunnel0     10.0.0.12      set  HMAC_MD5+DES_56_CB
236       0
```

Hub1, Hub2, Spoke1 및 Spoke2의 라우팅 테이블에 대해 몇 가지 흥미로운 문제가 있습니다.

- 두 허브 라우터는 스포크 라우터 뒤의 네트워크에 대한 동일 비용 경로를 가집니다.허브1:
  ```
  O     192.168.1.0/24 [110/2] via 10.0.0.11, 00:02:17, Tunnel0
  O     192.168.2.0/24 [110/2] via 10.0.0.12, 00:02:17, Tunnel0
  ```
  허브2:
  ```
  O     192.168.1.0/24 [110/2] via 10.0.0.11, 00:29:15, Tunnel0
  O     192.168.2.0/24 [110/2] via 10.0.0.12, 00:29:15, Tunnel0
  ```
  즉, Hub1과 Hub2는 스포크 라우터 뒤의 네트워크에 대해 동일한 비용을 허브 라우터 뒤의 네트워크의 라우터에 광고합니다.예를 들어 192.168.0.0/24 LAN에 직접 연결된 라우터 R2의 라우팅 테이블은 다음과 같습니다.R2:
  ```
  O     IA 192.168.1.0/24 [110/12] via 192.168.0.1, 00:00:26, Ethernet1/0/3
                          [110/12] via 192.168.0.2, 00:00:27, Ethernet1/0/3O
  O     IA 192.168.2.0/24 [110/12] via 192.168.0.1, 00:00:27, Ethernet1/0/3
                          [110/12] via 192.168.0.2, 00:00:27, Ethernet1/0/3
  ```
- 스포크 라우터는 두 허브 라우터를 통해 허브 라우터 뒤의 네트워크로 가는 동일 비용 경로를 가집니다.스포크1:
  ```
  O     IA 192.168.0.0/24 [110/11] via 10.0.0.1, 00:39:31, Tunnel0
                          [110/11] via 10.0.0.2, 00:39:31, Tunnel0
  ```
  스포크2:
  ```
  O     IA 192.168.0.0/24 [110/11] via 10.0.0.1, 00:57:56, Tunnel0
                          [110/11] via 10.0.0.2, 00:57:56, Tunnel0
  ```
  스포크 라우터가 패킷별 로드 밸런싱을 수행하는 경우 순서가 잘못된 패킷을 가져올 수 있습니다.

두 허브에 대한 링크를 통해 비대칭 라우팅 또는 패킷별 로드 밸런싱을 수행하지 않으려면 양쪽 방향에서 스포크 투 허브 경로를 선호하는 라우팅 프로토콜을 구성해야 합니다.Hub1이 기본 및 Hub2가 백업으로 설정되도록 하려면 허브 터널 인터페이스의 OSPF 비용을 다르게 설정할 수 있습

니다.

허브1:

```
interface tunnel0
...
ip ospf cost 10
...
```

허브2:

```
interface tunnel0
...
ip ospf cost 20
...
```

이제 경로는 다음과 같습니다.

허브1:

```
O     192.168.1.0/24 [110/11] via 10.0.0.11, 00:00:28, Tunnel0
O     192.168.2.0/24 [110/11] via 10.0.0.12, 00:00:28, Tunnel0
```

허브2:

```
O     192.168.1.0/24 [110/21] via 10.0.0.11, 00:00:52, Tunnel0
O     192.168.2.0/24 [110/21] via 10.0.0.12, 00:00:52, Tunnel0
```

R2:

```
O     IA 192.168.1.0/24 [110/31] via 192.168.0.1, 00:01:06, Ethernet1/0/3
O     IA 192.168.2.0/24 [110/31] via 192.168.0.1, 00:01:06, Ethernet1/0/3
```

두 허브 라우터는 이제 스포크 라우터 뒤의 네트워크에 대한 경로에 대해 비용이 다릅니다.즉, 라우터 R2에서 볼 수 있는 것처럼 Hub1은 스포크 라우터로 트래픽을 전달하는 데 선호됩니다. 이렇게 하면 위의 첫 번째 글머리 기호에 설명된 비대칭 라우팅 문제가 해결됩니다.

위 두 번째 글머리 기호에 설명된 대로 다른 방향의 비대칭 라우팅이 여전히 있습니다.OSPF를 동적 라우팅 프로토콜로 사용할 경우 스포크의 **router ospf 1**에 있는 **distance ...** 명령을 사용하여 Hub2를 통해 학습된 경로보다 Hub1을 통해 학습된 경로를 선호하도록 하여 해결 방법으로 이를 해결할 수 있습니다.

스포크1:

```
router ospf 1
 distance 111 10.0.0.2 0.0.0.0 1
access-list 1 permit any
```

스포크2:

```
router ospf 1
 distance 111 10.0.0.2 0.0.0.0 1
access-list 1 permit any
```

이제 경로는 다음과 같습니다.

스포크1:

```
O     192.168.0.0/24 [110/11] via 10.0.0.1, 00:00:06, Tunnel0
```

스포크2:

```
O     192.168.1.0/24 [110/11] via 10.0.0.1, 00:00:10, Tunnel0
```

위의 라우팅 컨피그레이션은 비대칭 라우팅으로부터 보호하며, 동시에 Hub1이 다운되면 Hub2로 페일오버를 허용합니다.즉, 두 허브가 모두 작동하면 Hub1만 사용됩니다.장애 조치 보호 및 비대칭 라우팅이 없는 허브 전반에 걸쳐 스포크를 균형적으로 조정하여 두 허브를 모두 사용하려면 특히 OSPF를 사용할 때 라우팅 컨피그레이션이 복잡해질 수 있습니다.따라서 듀얼 DMVPN 레이아웃이 있는 다음 듀얼 허브를 선택하는 것이 더 좋습니다.

듀얼 허브 - 듀얼 DMVPN 레이아웃

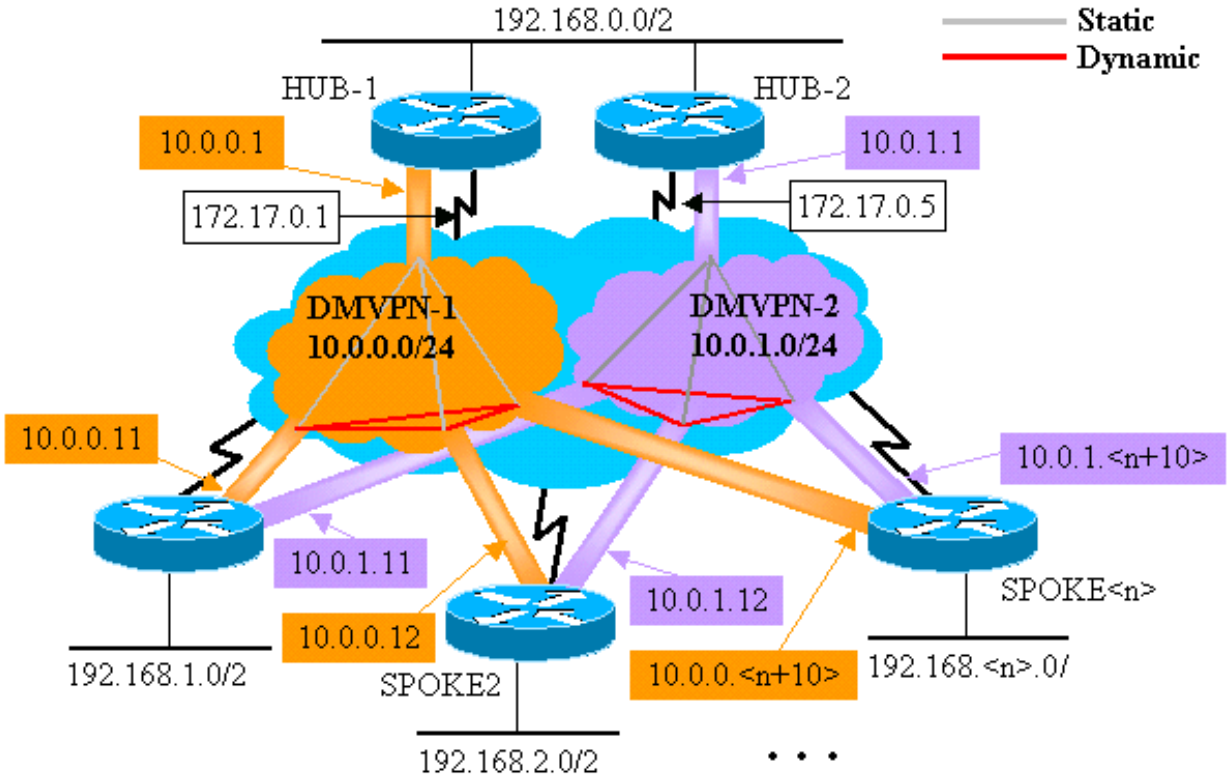
듀얼 DMVPN 레이아웃이 있는 듀얼 허브는 설정하는 것이 약간 더 어렵지만 DMVPN을 통한 라우팅을 더 효과적으로 제어할 수 있습니다.이 아이디어는 두 개의 DMVPN "클라우드"를 갖는 것입니다. 각 허브(이 경우 2개)는 하나의 DMVPN 서브넷("클라우드")에 연결되고 스포크는 두 DMVPN 서브넷("클라우드")에 연결됩니다. 스포크 라우터는 두 GRE 터널 인터페이스를 통해 두 허브 라우터가 모두 있는 라우팅 네이버이므로 인터페이스 컨피그레이션 차이(예: 대역폭, 비용 및 지연)를 사용하여 동적 라우팅 프로토콜 메트릭을 수정하여 둘 다 작동 중일 때 다른 허브에 대해 하나의 허브를 선호하도록 할 수 있습니다.

참고: 위의 문제는 일반적으로 허브 라우터가 함께 있는 경우에만 관련됩니다.둘 중 하나의 허브 라우터를 통해 목적지 네트워크에 연결할 수 있더라도 일반 동적 라우팅은 올바른 허브 라우터를 선호하게 됩니다.

스포크 라우터에서 p-pGRE 또는 mGRE 터널 인터페이스를 사용할 수 있습니다.스포크 라우터의 여러 p-pGRE 인터페이스는 동일한 **터널 소스**를 사용할 수 있습니다...IP 주소이지만 스포크 라우터의 여러 mGRE 인터페이스에는 고유한 **터널 소스가** 있어야 합니다...IP 주소.IPsec을 시작할 때 첫 번째 패킷은 mGRE 터널 중 하나와 연결해야 하는 ISAKMP 패킷이기 때문입니다.ISAKMP 패킷에는 이 연결을 설정할 대상 IP 주소(원격 IPsec 피어 주소)만 있습니다.이 주소는 **터널 소스 ...** 주소에 대해 매칭되지만 두 터널의 **터널 소스 ...** 주소가 동일하므로 첫 번째 mGRE 터널 인터페이스가 항상 매칭됩니다.즉, 수신 멀티캐스트 데이터 패킷이 잘못된 mGRE 인터페이스와 연결되어 동적 라우팅 프로토콜을 위반할 수 있습니다.

GRE 패킷 자체에는 **터널 키 ...** 두 mGRE 인터페이스를 구분하는 값이 있으므로 이 문제가 없습니다.Cisco IOS Software 릴리스 12.3(5) 및 12.3(7)T부터 이 제한을 극복하기 위한 추가 매개변수가 도입되었습니다.**터널 보호....공유**.shared 키워드는 여러 mGRE 인터페이스에서 동일한 소스 IP 주소의 IPSec 암호화를 사용함을 나타냅니다.이전 릴리스가 있는 경우 이중 DMVPN 레이아웃이 있는 이 듀얼 허브에서 p-pGRE 터널을 사용할 수 있습니다.p-pGRE 터널 케이스에서 **터널 소스 ...** 및 **터널 대상** 모두IP 주소는 일치에 사용할 수 있습니다.이 예제의 p-pGRE 터널은 이중 DMVPN 레이아웃이 있는 이 이중 허브에서 사용되며 **공유** 한정자를 사용하지 않습니다.

듀얼 허브 - 듀얼 DMVPN 레이아웃



강조 표시된 다음 변경 사항은 이 문서의 앞부분에서 설명한 동적 멀티포인트 허브 및 스포크 구성을 기준으로 합니다.

허브1 라우터

```
version 12.3
 !
 hostname Hub1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
  no ip split-horizon eigrp 1
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
```

```
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.1 255.255.255.252
!
interface Ethernet1
 ip address 192.168.0.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
 no auto-summary
!
```

허브2 라우터

```
version 12.3
!
hostname Hub2
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.1.1 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map multicast dynamic
 ip nhrp network-id 100001
 ip nhrp holdtime 600
 no ip split-horizon eigrp 1
 delay 1000
 tunnel source Ethernet0
 tunnel mode gre multipoint
 tunnel key 100001
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address 172.17.0.5 255.255.255.252
!
interface Ethernet1
 ip address 192.168.0.2 255.255.255.0
!
router eigrp 1
 network 10.0.1.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
 no auto-summary
!
```

이 경우 Hub1 및 Hub2 컨피그레이션이 유사합니다.주요 차이점은 각각 다른 DMVPN의 허브가 있다는 것입니다.각 DMVPN은 서로 다른 기능을 사용합니다.

- IP 서브넷(10.0.0.0/24, 10.0.0.1/24)
- NHRP 네트워크 ID(100000, 10001)
- 터널 키(100000, 100001)

이 문서의 뒷부분에서 설명한 대로 EIGRP를 사용하여 NBMA 네트워크를 설정하고 관리하기 쉽기 때문에 동적 라우팅 프로토콜이 OSPF에서 EIGRP로 전환되었습니다.

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Tunnel1
  bandwidth 1000
  ip address 10.0.1.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.1.1 172.17.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.1.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.5
  tunnel key 100001
  tunnel protection ipsec profile vpnprof
 !
 interface Ethernet0
  ip address dhcp hostname Spoke1
 !
 interface Ethernet1
  ip address 192.168.1.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
```

```
 network 10.0.1.0 0.0.0.255
 network 192.168.1.0 0.0.0.255
 no auto-summary
!
```

각 스포크 라우터는 2개의 p-pGRE 터널 인터페이스로 구성되며, 각 DMVPN에는 각각 하나씩 구성됩니다.ip **주소 ...**, ip nhrp **network-id ...**, **터널 키 ...** 및 **터널 대상 ....** 값은 두 터널을 구분하는 데 사용됩니다.동적 라우팅 프로토콜 EIGRP는 p-pGRE 터널 서브넷을 모두 통해 실행되며, 이를 통해 다른 DMVPN(p-pGRE 인터페이스)을 하나 선택할 수 있습니다.

Spoke2 라우터

```
version 12.3
!
hostname Spoke2
!
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel0
 bandwidth 1000
 ip address 10.0.0.12 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Tunnel1
 bandwidth 1000
 ip address 10.0.1.12 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.1.1 172.17.0.5
 ip nhrp network-id 100001
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.1.1
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.5
 tunnel key 100001
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0
 ip address dhcp hostname Spoke2
!
interface Ethernet1
 ip address 192.168.2.1 255.255.255.0
```

```
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 10.0.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
  no auto-summary
 !
```

스포크<n> 라우터

```
version 12.3
 !
 hostname Spoke<n>
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.




  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Tunnel1
  bandwidth 1000
  ip address 10.0.1.




  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.1.1 172.17.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.1.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.5
  tunnel key 100001
  tunnel protection ipsec profile vpnprof
```

```
!
interface Ethernet0
 ip address dhcp hostname Spoke<x>
!
interface Ethernet1
 ip address 192.168.<n>.1 255.255.255.0
!
router eigrp 1
 network 10.0.0.0 0.0.0.255
 network 10.0.1.0 0.0.0.255
 network 192.168.<n>.0 0.0.0.255
 no auto-summary
!
```

이제 Hub1, Hub2, Spoke1 및 Spoke2 라우터의 라우팅 테이블, NHRP 매핑 테이블 및 IPsec 연결을 살펴보고 초기 조건(Spoke1 및 Spoke2 라우터가 올라간 바로 후)을 살펴보겠습니다.

초기 조건 및 변경 사항

Hub1 라우터 정보

```
Hub1#show ip route
      172.17.0.0/30 is subnetted, 1 subnets
 C       172.17.0.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 2 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 D       10.0.1.0 [90/2611200] via 192.168.0.2,
00:00:46, Ethernet1
 C    192.168.0.0/24 is directly connected, Ethernet1
 D    192.168.1.0/24 [90/2841600] via 10.0.0.11,
00:00:59, Tunnel0
 D    192.168.2.0/24 [90/2841600] via 10.0.0.12,
00:00:34, Tunnel0
 Hub1#show ip nhrp
 10.0.0.12/32 via 10.0.0.12, Tunnel0 created 23:48:32,
expire 00:03:50
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.2.75
 10.0.0.11/32 via 10.0.0.11, Tunnel0 created 23:16:46,
expire 00:04:45
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.1.24
 Hub1#show crypto engine connection active
   ID Interface   IP-Address    State  Algorithm
Encrypt  Decrypt
   15 Ethernet0   172.17.63.18   set
HMAC_SHA+DES_56_CB      0        0
   16 Ethernet0   10.0.0.1       set
HMAC_SHA+DES_56_CB      0        0
 2038 Tunnel0     10.0.0.1       set
HMAC_MD5+DES_56_CB      0      759
 2039 Tunnel0     10.0.0.1       set
HMAC_MD5+DES_56_CB    726        0
 2040 Tunnel0     10.0.0.1       set
HMAC_MD5+DES_56_CB      0       37
 2041 Tunnel0     10.0.0.1       set
HMAC_MD5+DES_56_CB     36        0
```

Hub2 라우터 정보

```
Hub2#show ip route
      172.17.0.0/30 is subnetted, 1 subnets
 C       172.17.0.4 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 2 subnets
 D       10.0.0.0 [90/2611200] via 192.168.0.1,
00:12:22, Ethernet1
 C       10.0.1.0 is directly connected, Tunnel0
 C    192.168.0.0/24 is directly connected, Ethernet1
 D    192.168.1.0/24 [90/2841600] via 10.0.1.11,
00:13:24, Tunnel0
 D    192.168.2.0/24 [90/2841600] via 10.0.1.12,
00:12:11, Tunnel0
 Hub2#show ip nhrp
 10.0.1.12/32 via 10.0.1.12, Tunnel3 created 06:03:24,
expire 00:04:39
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.2.75
 10.0.1.11/32 via 10.0.1.11, Tunnel3 created 23:06:47,
expire 00:04:54
   Type: dynamic, Flags: authoritative unique registered
   NBMA address: 172.16.1.24
 Hub2#show crypto engine connection active
   ID Interface   IP-Address    State  Algorithm
Encrypt  Decrypt
    4 Ethernet0   171.17.0.5     set
HMAC_SHA+DES_56_CB      0        0
    6 Ethernet0   171.17.0.5     set
HMAC_SHA+DES_56_CB      0        0
 2098 Tunnel0     10.0.1.1       set
HMAC_MD5+DES_56_CB      0      722
 2099 Tunnel0     10.0.1.1       set
HMAC_MD5+DES_56_CB    690        0
 2100 Tunnel0     10.0.1.1       set
HMAC_MD5+DES_56_CB      0      268
 2101 Tunnel0     10.0.1.1       set
HMAC_MD5+DES_56_CB    254        0
```

Spoke1 라우터 정보

```
Spoke1#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.1.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 1 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 C       10.0.1.0 is directly connected, Tunnel1
 D    192.168.0.0/24 [90/2841600] via 10.0.1.1,
00:26:30, Tunnel1
                     [90/2841600] via 10.0.0.1,
00:26:30, Tunnel0
 C    192.168.1.0/24 is directly connected, Ethernet1
 D    192.168.2.0/24 [90/3097600] via 10.0.1.1,
00:26:29, Tunnel1
                     [90/3097600] via 10.0.0.1,
00:26:29, Tunnel0
 Spoke1#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 23:25:46,
never expire
   Type: static, Flags: authoritative
   NBMA address: 172.17.0.1
 10.0.1.1/32 via 10.0.1.1, Tunnel1 created 23:24:40,
never expire
```

```
   Type: static, Flags: authoritative
   NBMA address: 172.17.0.5
 Spoke1#show crypto engine connection active
   ID Interface   IP-Address    State  Algorithm
Encrypt  Decrypt
   16 Ethernet0   172.16.1.24    set
HMAC_SHA+DES_56_CB      0        0
   18 Ethernet0   172.16.1.24    set
HMAC_SHA+DES_56_CB      0        0
 2118 Tunnel0     10.0.0.11      set
HMAC_MD5+DES_56_CB      0      181
 2119 Tunnel0     10.0.0.11      set
HMAC_MD5+DES_56_CB    186        0
 2120 Tunnel1     10.0.1.11      set
HMAC_MD5+DES_56_CB      0      105
 2121 Tunnel1     10.0.1.11      set
HMAC_MD5+DES_56_CB    110        0
```

Spoke2 라우터 정보

```
Spoke2#show ip route
      172.16.0.0/24 is subnetted, 1 subnets
 C       172.16.2.0 is directly connected, Ethernet0
      10.0.0.0/24 is subnetted, 2 subnets
 C       10.0.0.0 is directly connected, Tunnel0
 C       10.0.1.0 is directly connected, Tunnel1
 D    192.168.0.0/24 [90/2841600] via 10.0.1.1,
00:38:04, Tunnel1
                     [90/2841600] via 10.0.0.1,
00:38:04, Tunnel0
 D    192.168.1.0/24 [90/3097600] via 10.0.1.1,
00:38:02, Tunnel1
                     [90/3097600] via 10.0.0.1,
00:38:02, Tunnel0
 C    192.168.2.0/24 is directly connected, Ethernet1
 Spoke2#show ip nhrp
 10.0.0.1/32 via 10.0.0.1, Tunnel0 created 1d02h, never
expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.1
 10.0.1.1/32 via 10.0.1.1, Tunnel1 created 1d02h, never
expire
   Type: static, Flags: authoritative used
   NBMA address: 172.17.0.5
 Spoke2#show crypto engine connection active
   ID Interface   IP-Address    State  Algorithm
Encrypt  Decrypt
    8 Ethernet0   172.16.2.75    set
HMAC_SHA+DES_56_CB      0        0
    9 Ethernet0   172.16.2.75    set
HMAC_SHA+DES_56_CB      0        0
 2036 Tunnel0     10.0.0.12      set
HMAC_MD5+DES_56_CB      0      585
 2037 Tunnel0     10.0.0.12      set
HMAC_MD5+DES_56_CB    614        0
 2038 Tunnel1     10.0.1.12      set
HMAC_MD5+DES_56_CB      0      408
 2039 Tunnel1     10.0.1.12      set
HMAC_MD5+DES_56_CB    424        0
```

Hub1, Hub2, Spoke1 및 Spoke2의 라우팅 테이블에 대해 몇 가지 흥미로운 사항을 확인할 수 있습니다.

- 두 허브 라우터는 스포크 라우터 뒤의 네트워크에 대한 동일 비용 경로를 가집니다.허브1:

```
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, 00:00:59, Tunnel0
D    192.168.2.0/24 [90/2841600] via 10.0.0.12, 00:00:34, Tunnel0
```

  허브2:

```
D    192.168.1.0/24 [90/2841600] via 10.0.1.11, 00:13:24, Tunnel0
D    192.168.2.0/24 [90/2841600] via 10.0.1.12, 00:12:11, Tunnel0
```

  즉, Hub1과 Hub2는 스포크 라우터 뒤의 네트워크에 대해 동일한 비용을 허브 라우터 뒤의 네트워크의 라우터에 광고합니다.예를 들어 192.168.0.0/24 LAN에 직접 연결된 라우터 R2의 라우팅 테이블은 다음과 같습니다.R2:

```
D    192.168.1.0/24 [90/2867200] via 192.168.0.1, 00:51:51, Ethernet1/0/3
                                [90/2867200] via 192.168.0.2, 00:51:51, Ethernet1/0/3
D    192.168.2.0/24 [90/2867200] via 192.168.0.2, 00:52:43, Ethernet1/0/3
                               [90/2867200] via 192.168.0.1, 00:52:43, Ethernet1/0/3
```

- 스포크 라우터는 두 허브 라우터를 통해 허브 라우터 뒤의 네트워크로 가는 동일 비용 경로를 가집니다.스포크1:

```
D    192.168.0.0/24 [90/3097600] via 10.0.1.1, 00:26:30, Tunnel1
                                [90/3097600] via 10.0.0.1, 00:26:30, Tunnel0
```

  스포크2:

```
D    192.168.0.0/24 [90/3097600] via 10.0.1.1, 00:38:04, Tunnel1
                                [90/3097600] via 10.0.0.1, 00:38:04, Tunnel0
```

  스포크 라우터가 패킷별 로드 밸런싱을 수행하는 경우 순서가 잘못된 패킷을 가져올 수 있습니다.

두 허브에 대한 링크를 통해 비대칭 라우팅 또는 패킷별 로드 밸런싱을 수행하지 않으려면 양쪽 방향에서 스포크 투 허브 경로를 선호하는 라우팅 프로토콜을 구성해야 합니다.Hub1이 기본 및 Hub2가 백업으로 설정되도록 하려면 허브 터널 인터페이스의 지연을 다르게 설정할 수 있습니다.

허브1:

```
interface tunnel0
...
delay 1000
...
```

허브2:

```
interface tunnel0
...
delay 1050
...
```

참고: 이 예에서는 두 허브(100)의 Ethernet1 인터페이스의 지연 시간보다 작기 때문에 Hub2의 터널 인터페이스의 지연 시간에 50이 추가되었습니다. 이렇게 하면 Hub2는 여전히 패킷을 스포크 라우터로 직접 전달하지만 Hub1보다 덜 바람직한 경로를 Hub1 및 Hub2 뒤의 라우터로 광고합니다. 지연이 100개 이상 증가하면 Hub2는 Ethernet1 인터페이스를 통해 스포크 라우터에 대한 패킷을 Hub1을 통해 전달하지만 Hub1 및 Hub2 뒤의 라우터는 스포크 라우터로 패킷을 전송하는 데 Hub-1을 올바르게 선호합니다.

이제 경로는 다음과 같습니다.

허브1:

```
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, 00:01:11, Tunnel0
D    192.168.2.0/24 [90/2841600] via 10.0.0.12, 00:01:11, Tunnel0
```

허브2:

```
D    192.168.1.0/24 [90/2854400] via 10.0.1.11, 00:00:04, Tunnel0
D    192.168.2.0/24 [90/2854400] via 10.0.1.12, 00:00:04, Tunnel0
```
R2:

```
D    192.168.1.0/24 [90/2867200] via 192.168.0.1, 00:02:18, Ethernet1/0/3
D    192.168.2.0/24 [90/2867200] via 192.168.0.1, 00:02:18, Ethernet1/0/3
```
두 허브 라우터는 스포크 라우터 뒤의 네트워크 경로에 대해 비용이 다르므로, 이 경우 R2에서 볼 수 있듯이 Hub1은 스포크 라우터로 트래픽을 전달하는 데 선호됩니다. 위의 첫 번째 글머리 기호에 설명된 문제를 처리합니다.

위의 두 번째 글머리 기호에 설명된 문제는 여전히 존재하지만 두 개의 p-pGRE 터널 인터페이스가 있으므로 터널 인터페이스에서 **지연 ...**을 별도로 설정하여 Hub1 대 Hub2에서 학습된 경로의 EIGRP 메트릭을 변경할 수 있습니다.

스포크1:

```
interface tunnel0
  delay 1000
interface tunnel1
  delay 1050
```
스포크2:

```
interface tunnel0
  delay 1000
interface tunnel1
  delay 1050
```
이제 경로는 다음과 같습니다.

스포크1:

```
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 00:15:44, Tunnel0
D    192.168.2.0/24 [90/3097600] via 10.0.0.1, 00:15:44, Tunnel0
```
스포크2:

```
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 00:13:54, Tunnel0
D    192.168.1.0/24 [90/3097600] via 10.0.0.1, 00:13:54, Tunnel0
```
위의 라우팅 컨피그레이션은 비대칭 라우팅으로부터 보호하며, 동시에 Hub1이 다운되면 Hub2로 페일오버를 허용합니다.즉, 두 허브가 모두 작동하면 Hub1만 사용됩니다.

장애 조치 보호 및 비대칭 라우팅 없이 허브의 스포크를 균형적으로 조정하여 두 허브를 모두 사용하려는 경우 라우팅 컨피그레이션이 더 복잡하지만 EIGRP를 사용할 때 이를 수행할 수 있습니다 .이렇게 하려면 허브 라우터의 터널 인터페이스**에서 지연 ...을** 다시 동일하게 설정한 다음 스포크 라우터에서 **offset-list <acl> out <offset> <interface>** 명령을 사용하여 GRE 터널 인터페이스를 백업 허브로 광고하는 경로의 EIGRP 메트릭을 늘립니다.스포크의 Tunnel0 및 Tunnel1 인터페이스 간의 불균형한 **지연**은 계속 사용되므로 스포크 라우터는 기본 허브 라우터를 선호합니다.스포크 라우터의 변경 사항은 다음과 같습니다.

Spoke1 라우터

```
version 12.3
 !
 hostname Spoke1
 !
 ...
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.0.1 172.17.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.0.1
  delay 1000
  tunnel source Ethernet0
  tunnel destination 172.17.0.1
  tunnel key 100000
  tunnel protection ipsec profile vpnprof
 !
 interface Tunnel1
  bandwidth 1000
  ip address 10.0.1.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
  ip nhrp map 10.0.1.1 172.17.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
  ip nhrp nhs 10.0.1.1
  delay 1500
  tunnel source Ethernet0
  tunnel destination 172.17.0.5
  tunnel key 100001
  tunnel protection ipsec profile vpnprof
 !
 ...
 !
 router eigrp 1
  offset-list 1 out 12800 Tunnel1
  network 10.0.0.0 0.0.0.255
  network 10.0.1.0 0.0.0.255
  network 192.168.1.0
  distribute-list 1 out
  no auto-summary
 !
 access-list 1 permit 192.168.1.0
 !
```

Spoke2 라우터

```
version 12.3
 !
 hostname Spoke2
 !
 ...
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.12 255.255.255.0
  ip mtu 1400
```

```
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.0.1
 delay 1500
 tunnel source Ethernet0
 tunnel destination 172.17.0.1
 tunnel key 100000
 tunnel protection ipsec profile vpnprof
!
interface Tunnel1
 bandwidth 1000
 ip address 10.0.1.12 255.255.255.0
 ip mtu 1400
 ip nhrp authentication test
 ip nhrp map 10.0.1.1 172.17.0.5
 ip nhrp network-id 100001
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.1.1
 delay 1000
 tunnel source Ethernet0
 tunnel destination 172.17.0.5
 tunnel key 100001
 tunnel protection ipsec profile vpnprof
!
...
!
router eigrp 1
 offset-list 1 out 12800 Tunnel1
 network 10.0.0.0 0.0.0.255
 network 10.0.1.0 0.0.0.255
 network 192.168.2.0
 distribute-list 1 out
 no auto-summary
!
access-list 1 permit 192.168.2.0
!
```

참고: 오프셋 값 12800(50*256)이 25600(100*256)보다 작으므로 EIGRP 메트릭에 추가되었습니다. 이 값(25600)은 허브 라우터 간에 학습된 경로에 대해 EIGRP 메트릭에 추가되는 값입니다.**offset-list** 명령에서 12800을 사용하면 백업 허브 라우터는 이더넷을 통해 이러한 패킷을 전달하여 해당 스포크에 대한 기본 허브 라우터를 통과하는 대신 스포크 라우터로 패킷을 직접 전달합니다.허브 라우터가 광고하는 경로의 메트릭은 여전히 올바른 기본 허브 라우터가 선호됩니다.스포크의 절반은 기본 라우터로 Hub1을, 나머지 절반은 기본 라우터로 Hub2를 갖습니다.

참고: 오프셋 값이 25600(100*256)을 초과하여 증가한 경우 허브는 Ethernet1 인터페이스를 통해 다른 허브를 통해 스포크 라우터의 절반의 패킷을 전달합니다. 허브 뒤에 있는 라우터는 여전히 스포크 라우터로 패킷을 전송하는 올바른 허브를 선호하지만, 허브는 Ethernet1 인터페이스를 통해 스포크 라우터의 패킷을 전달합니다.

참고: distribute-list 1 out 명령은 스포크의 한 터널 인터페이스를 통해 한 허브 라우터에서 학습한 경로를 다른 터널을 통해 다른 허브로 다시 알릴 수 있으므로 추가되었습니다.distribute**-list ...** 명령을 사용하면 스포크 라우터가 자체 경로만 알릴 수 있습니다.

참고: 스포크 라우터가 아닌 허브 라우터에서 라우팅 광고를 제어하려는 경우 스포크 대신 허브 라우터에서 **<value> <interface>의 offset-list <acl1>** 및 명령**의 distribute-list <acl2>**를 구성할 수 있습니다.<acl2> access-list는 모든 스포크의 경로를 나열하고 <acl1> access-list는 다른 허브 라우터가 기본 허브가 될 스포크의 경로를 나열합니다.

이러한 변경을 통해 경로는 다음과 같습니다.

허브1:

```
D    192.168.1.0/24 [90/2841600] via 10.0.0.11, 00:12:11, Tunnel2
D    192.168.2.0/24 [90/2854400] via 10.0.0.12, 00:13:24, Tunnel2
```

허브2:

```
D    192.168.1.0/24 [90/2854400] via 10.0.1.11, 00:09:58, Tunnel0
D    192.168.2.0/24 [90/2841600] via 10.10.1.12, 00:11:11, Tunnel0
```

R2:

```
D    192.168.1.0/24 [90/2867200] via 192.168.0.1, 00:13:13, Ethernet1/0/3
D    192.168.2.0/24 [90/2867200] via 192.168.0.2, 00:14:25, Ethernet1/0/3
```

스포크1:

```
D    192.168.0.0/24 [90/2841600] via 10.0.0.1, 00:16:12, Tunnel0
```

스포크2:

```
D    192.168.0.0/24 [90/2841600] via 10.0.1.1, 00:18:54, Tunnel1
```

결론

DMVPN 솔루션은 대규모 및 소규모 IPsec VPN 네트워크를 보다 효과적으로 확장하기 위해 다음과 같은 기능을 제공합니다.

- DMVPN을 사용하면 풀 메시 또는 부분 메시 IPsec VPN에서 더 효과적으로 확장할 수 있습니다.스포크 투 스포크 트래픽이 산발적인 경우 특히 유용합니다(예: 모든 스포크가 모든 스포크에 데이터를 지속적으로 보내지 않음). 스포크 간에 직접 IP 연결이 있는 한 모든 스포크가 다른 스포크로 직접 데이터를 보낼 수 있습니다.
- DMVPN은 동적으로 할당된 주소(예: 케이블, ISDN, DSL)가 있는 IPsec 노드를 지원합니다.이는 메시 네트워크뿐 아니라 허브 앤 스포크(hub and spoke)에도 적용됩니다.DMVPN을 사용하려면 허브-스포크 링크가 계속 작동해야 합니다.
- DMVPN은 VPN 노드 추가를 간소화합니다.새 스포크 라우터를 추가할 때는 스포크 라우터를 구성하고 네트워크에 연결하기만 하면 됩니다. 그러나 허브의 새 스포크에 대한 ISAKMP 권한 부여 정보를 추가해야 할 수도 있습니다. 허브는 새 스포크에 대해 동적으로 학습하며 동적 라우팅 프로토콜은 허브 및 다른 모든 스포크로 라우팅을 전파합니다.
- DMVPN은 VPN의 모든 라우터에 필요한 컨피그레이션 크기를 줄입니다.이는 GRE+IPsec 허브 및 스포크 전용 VPN 네트워크에서도 마찬가지입니다.
- DMVPN은 GRE를 사용하므로 VPN 전체에서 IP 멀티캐스트 및 동적 라우팅 트래픽을 지원합니다.즉, 동적 라우팅 프로토콜을 사용할 수 있으며 프로토콜에서 이중화된 "허브"를 지원할 수 있습니다.멀티캐스트 애플리케이션도 지원됩니다.
- DMVPN은 스포크에서 분할 터널링을 지원합니다.

관련 정보

- DMVPN(Dynamic Multipoint VPN)
- IPSec 지원 페이지
- Technical Support - Cisco Systems