

Cisco IOS NAT - MPLS VPN과의 통합

목차

[소개](#)

[NAT의 이점 - MPLS 통합](#)

[설계 고려 사항](#)

[구축 시나리오](#)

[구축 옵션 및 구성 세부 정보](#)

[이그레스 PE NAT](#)

[인그레스 PE NAT](#)

[인그레스 PE NAT 이후 중앙 PE에 도착하는 패킷](#)

[서비스 예](#)

[사용 가능성](#)

[결론](#)

[관련 정보](#)

소개

Cisco IOS[®] NAT(Network Address Translation) 소프트웨어를 사용하면 VPN의 디바이스가 중복되는 IP 주소를 사용하는 경우에도 여러 MPLS VPN에서 공유 서비스에 액세스할 수 있습니다. Cisco IOS NAT는 VRF 인식이며 MPLS 네트워크 내의 공급자 에지 라우터에서 구성할 수 있습니다.

참고: IOS의 MPLS는 레거시 NAT에서만 지원됩니다. 현재 MPLS를 사용하는 NAT NVI에 대한 Cisco IOS에서는 지원되지 않습니다.

MPLS VPN의 구축은 향후 몇 년간 빠르게 증가할 것으로 예상됩니다. 빠른 확장 및 유연한 연결 옵션을 허용하는 공통 네트워크 인프라의 이점은 네트워크 커뮤니티에 제공할 수 있는 서비스의 성장을 의심할 여지 없이 촉진할 것입니다.

그러나 성장의 장애물은 여전히 남아있다. IPv6와 가까운 미래에 접속 요구 사항을 증가하는 IP 주소 공간의 확대는 아직 구축 초기 단계에 있습니다. 기존 네트워크는 일반적으로 RFC 1918에 정의된 대로 프라이빗 IP 주소 지정 체계를 사용합니다. 네트워크 주소 변환은 주소 공간이 중복되거나 중복되는 경우 네트워크를 상호 연결하는 데 자주 사용됩니다.

고객 및 파트너와 제공 또는 공유하려는 네트워크 애플리케이션 서비스를 보유한 서비스 제공자와 기업은 서비스 사용자에게 부과되는 연결 부담을 최소화하고자 합니다. 원하는 목표나 수익을 달성하기 위해 필요한 만큼 잠재적 사용자에게 서비스를 확장하는 것이 바람직하며, 심지어 필수적입니다. 사용 중인 IP 주소 지정 체계가 잠재 사용자를 제외하는 장애물이 되어서는 안 됩니다.

통신 서비스 공급업체는 공통 MPLS VPN 인프라 내에 Cisco IOS NAT를 구축함으로써 고객의 연결 부담을 덜고 더 많은 공유 애플리케이션 서비스를 더 많은 소비자와 연결하는 기능을 가속화할 수 있습니다.

NAT의 이점 - MPLS 통합

NAT와 MPLS의 통합은 서비스 제공자와 엔터프라이즈 고객 모두에게 혜택을 제공합니다. 이 서비스는 서비스 제공자에게 공유 서비스를 구축하고 이러한 서비스에 대한 액세스를 제공할 수 있는 더 많은 옵션을 제공합니다. 추가 서비스 오퍼링은 경쟁사에 비해 차별화 요소가 될 수 있습니다.

통신 사업자용	VPN용
더 많은 서비스 오퍼링	비용 절감
향상된 액세스 옵션	더 간단한 액세스
매출 증대	유연성 해결

현재 워크로드 중 일부를 아웃소싱하려는 엔터프라이즈 고객은 서비스 제공업체의 광범위한 오퍼링을 활용할 수 있습니다. 필요한 주소 변환을 통신 사업자 네트워크로 수행하는 데 따르는 부담을 서비스 사업자 네트워크로 옮기면 복잡한 관리 작업으로 인한 부담이 해소됩니다. 고객은 사실 주소 지정을 계속 사용하면서 공유 서비스 및 인터넷에 대한 액세스를 유지할 수 있습니다. 통신 사업자 네트워크 내에서 NAT 기능을 통합하면 고객 에지 라우터가 NAT 기능을 수행할 필요가 없으므로 기업 고객의 총 비용도 절감할 수 있습니다.

설계 고려사항

MPLS 네트워크 내에서 NAT를 호출하는 설계를 고려할 때, 첫 번째 단계는 애플리케이션 관점에서 서비스 요구 사항을 확인하는 것입니다. 사용되는 프로토콜과 애플리케이션에서 적용한 특별한 클라이언트/서버 통신을 고려해야 합니다. 사용된 프로토콜에 필요한 지원이 Cisco IOS NAT에서 지원되고 처리되는지 확인합니다. 지원되는 프로토콜 목록은 [Cisco IOS NAT Application Layer Gateway](#) 문서에서 제공됩니다.

다음으로, 공유 서비스의 예상 사용량 및 초당 패킷 수로 예상되는 트래픽 속도를 확인해야 합니다. NAT는 라우터 CPU 집약적 기능입니다. 따라서 성능 요구 사항은 특정 구축 옵션을 선택하고 관련된 NAT 디바이스의 수를 결정하는 데 중요한 요소가 됩니다.

또한 보안 문제와 예방 조치를 고려해야 합니다. MPLS VPN은 기본적으로 사실 및 효과적으로 분리된 트래픽이지만, 공유 서비스 네트워크는 일반적으로 많은 VPN에서 일반적입니다.

구축 시나리오

MPLS 제공자 에지에서 NAT 구축을 위한 두 가지 옵션이 있습니다.

- 이그레스 NAT PE로 중앙 집중화
- 인그레스 NAT PE로 분산

공유 서비스 네트워크에 가장 가까운 MPLS 네트워크의 이그레스 지점에서 NAT 기능을 구성할 경우 다음과 같은 이점이 있습니다.

- 더 간단한 서비스 프로비저닝을 촉진하는 중앙 집중식 구성
- 간소화된 문제 해결
- 향상된 운영 확장성
- IP 주소 할당 요구 사항 감소

그러나 확장성과 성능이 감소하면 이점이 상쇄됩니다. 이것이 고려해야 할 주요 거래입니다. 물론 이 기능을 MPLS 네트워크와 통합하는 것이 바람직하지 않다고 판단될 경우 고객 네트워크에서 NAT 기능을 수행할 수도 있습니다.

인그레스 PE NAT

그림 1과 같이 MPLS 네트워크 인그레스 PE 라우터에서 NAT를 구성할 수 있습니다. 이 설계에서는 확장성이 크게 유지되고 여러 에지 디바이스에 NAT 기능을 분산하여 성능이 최적화됩니다. 각 NAT PE는 해당 PE에 로컬로 연결된 사이트의 트래픽을 처리합니다. NAT 규칙 및 액세스 제어 목록 또는 경로 맵은 변환이 필요한 패킷을 제어합니다.

그림 1:인그레스 PE NAT

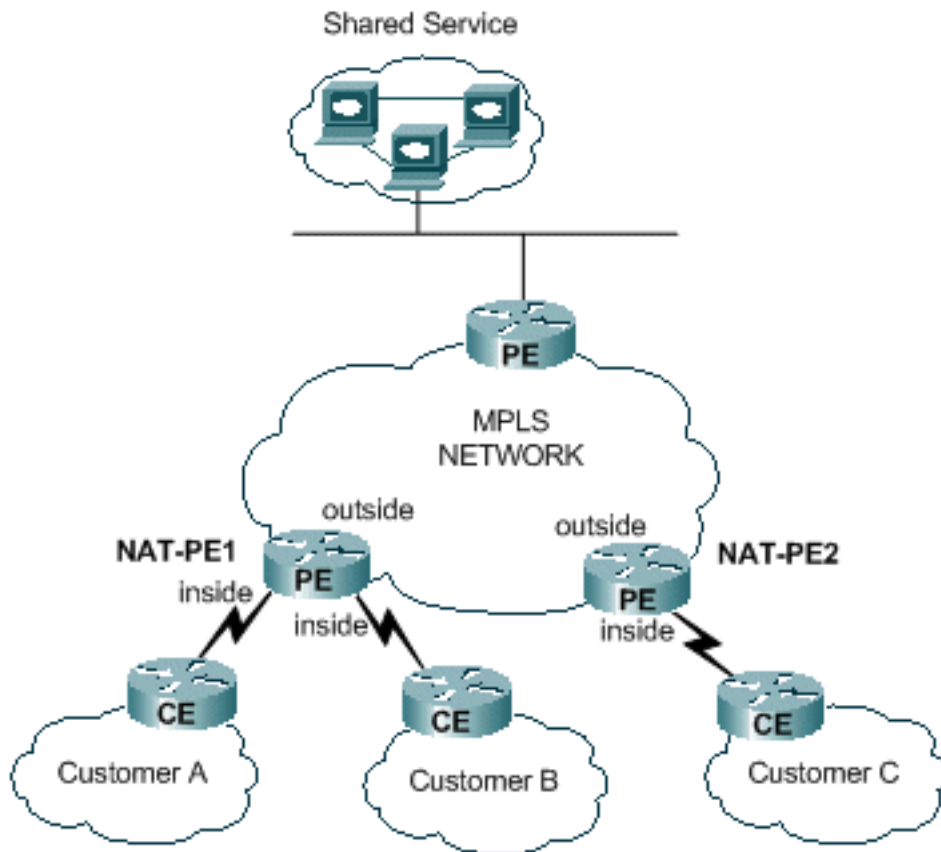
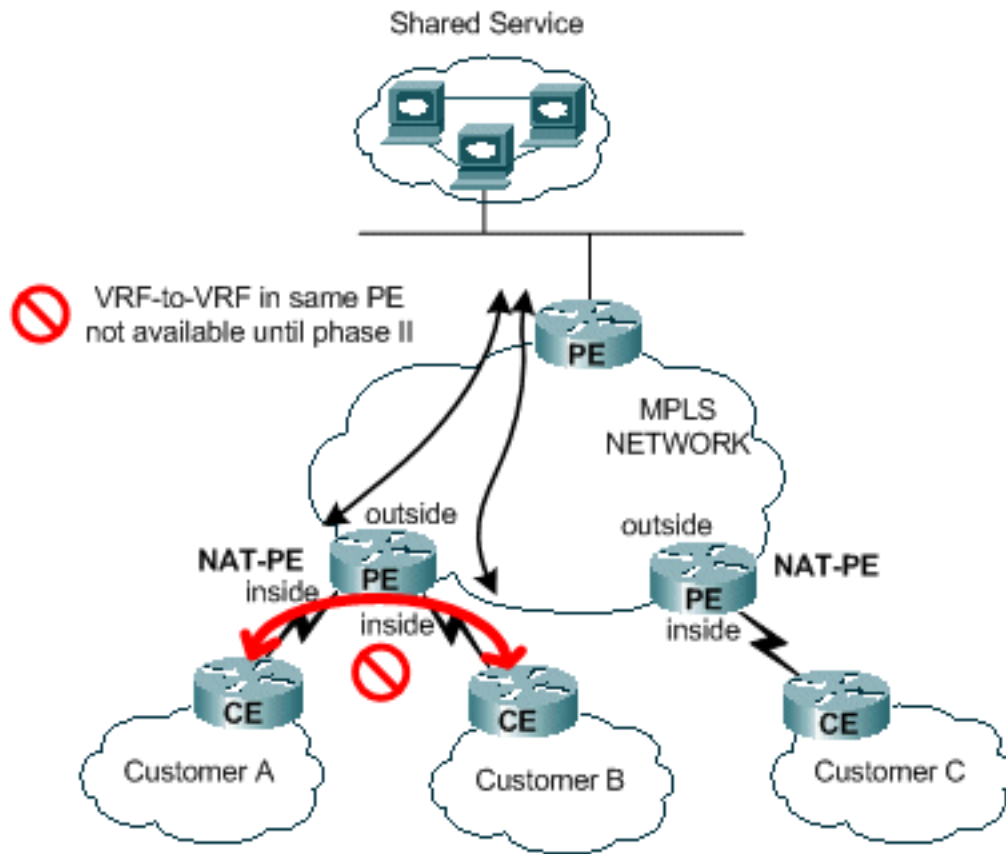


그림 2에 나와 있는 것처럼 두 VRF 간에 NAT를 방지하는 동시에 공유 서비스에 NAT를 제공하는 제한이 있습니다. 이는 인터페이스를 NAT "inside" 및 "outside" 인터페이스로 지정해야 하기 때문입니다. 단일 PE에서 VRF 간 연결을 지원하는 기능은 향후 Cisco IOS 릴리스에 제공될 예정입니다.

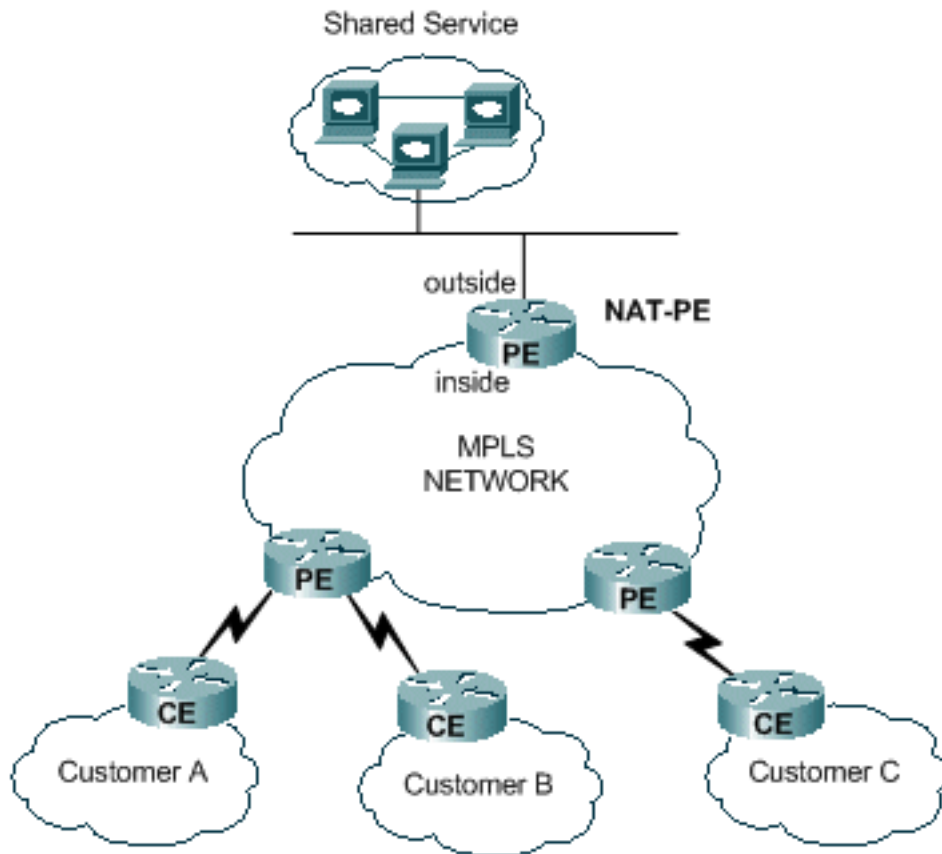
그림 2:비즈니스 대 비즈니스



이그레스 PE NAT

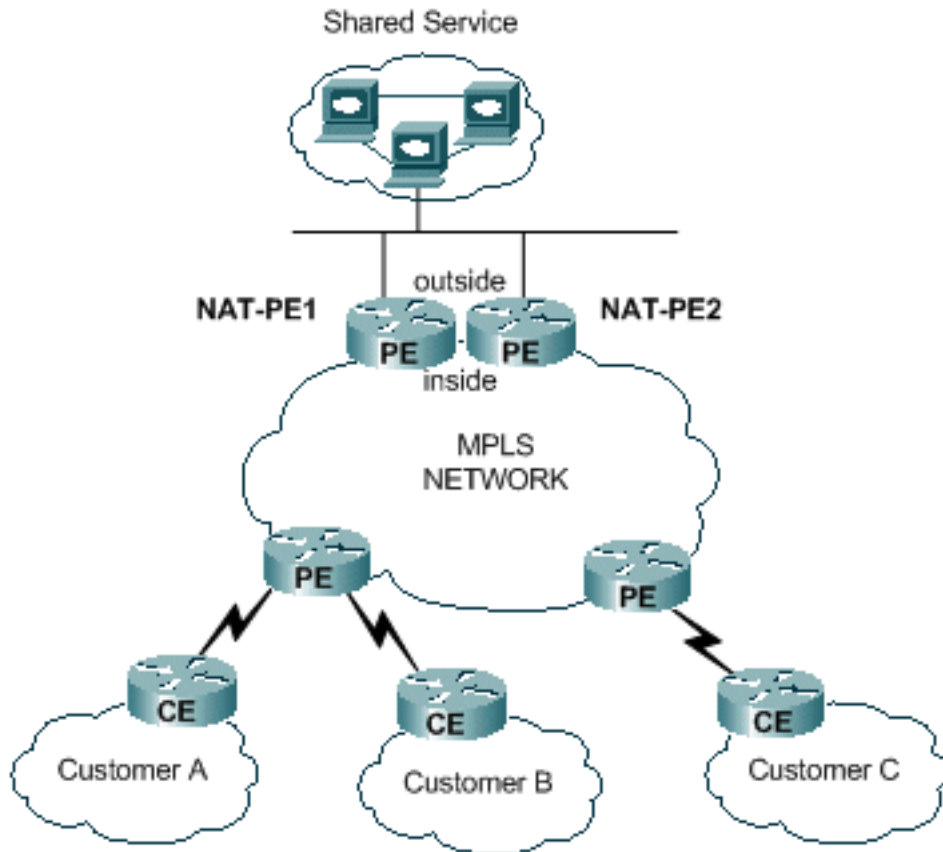
그림 3과 같이 MPLS 네트워크 이그레스 PE 라우터에서 NAT를 구성할 수 있습니다. 이 설계에서는 중앙 PE가 공유 서비스에 액세스하는 모든 고객 네트워크에 대한 경로를 유지해야 하므로 확장성이 어느 정도 줄어듭니다. 트래픽이 패킷의 IP 주소를 변환해야 하는 라우터에 과도한 부담을 주지 않도록 애플리케이션 성능 요구 사항도 고려해야 합니다. NAT는 이 경로를 사용하는 모든 고객에게 중앙에서 발생하므로 IP 주소 풀을 공유할 수 있습니다. 따라서 필요한 총 서브넷 수가 감소합니다.

그림 3: 이그레스 PE NAT



[그림 4](#)와 같이 이그레스 PE NAT 설계의 확장성을 높이기 위해 여러 라우터를 구축할 수 있습니다. 이 시나리오에서는 고객 VPN을 특정 NAT 라우터에서 "프로비저닝"할 수 있습니다. 네트워크 주소 변환은 해당 VPN 집합에 대해 공유 서비스를 오가는 집계 트래픽에 대해 발생합니다. 예를 들어 고객 A와 B에 대한 VPN의 트래픽은 NAT-PE1을 사용할 수 있고 고객 C에 대한 VPN의 트래픽은 NAT-PE2를 사용합니다. 각 NAT PE는 정의된 특정 VPN에 대해서만 트래픽을 전달하며 해당 VPN의 사이트로의 경로만 유지합니다. 각 NAT PE 라우터 내에서 별도의 NAT 주소 풀을 정의하여 패킷이 공유 서비스 네트워크에서 올바른 NAT PE로 라우팅되어 고객 VPN으로 다시 변환 및 라우팅되도록 할 수 있습니다.

그림 4: 다중 이그레스 PE NAT



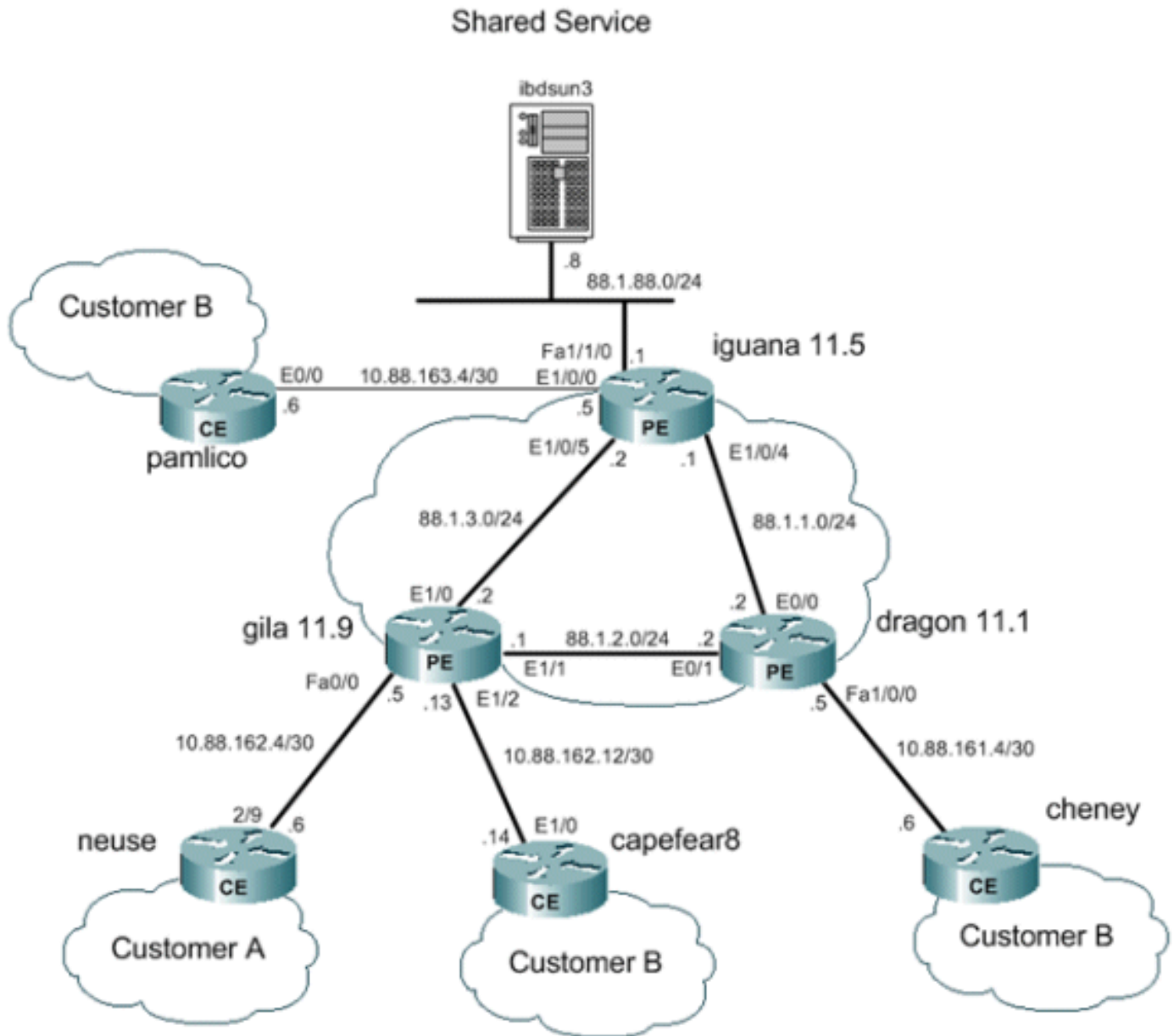
중앙 집중식 설계에서는 공유 서비스 네트워크의 구성 방법에 대한 제한이 적용됩니다. 특히 공유 서비스 VPN과 고객 VPN 간의 MPLS VPN 경로 가져오기/내보내기는 사용할 수 없습니다. 이는 [RFC 2547](#)에 지정된 MPLS 작업의 특성에 따른 것입니다. 확장 커뮤니티 및 경로 설명자를 사용하여 경로를 가져오고 내보내는 경우 NAT는 중앙 NAT PE로 들어오는 패킷에서 소스 VPN을 확인할 수 없습니다. 일반적인 경우는 공유 서비스 네트워크를 VRF 인터페이스가 아닌 일반 인터페이스로 만드는 것입니다. 그런 다음 프로비저닝 프로세스의 일부로 공유 서비스에 대한 액세스 권한이 필요한 고객 VPN과 연결된 각 VRF 테이블에 대해 공유 서비스 네트워크에 대한 경로가 중앙 NAT PE에 추가됩니다. 자세한 내용은 나중에 설명하겠습니다.

구축 옵션 및 구성 세부 정보

이 섹션에는 각 구축 옵션과 관련된 몇 가지 세부 정보가 포함되어 있습니다. [그림 5](#)에 나와 있는 네트워크에서 이 예를 모두 가져왔습니다. 이 섹션의 나머지 부분은 이 다이어그램을 참조하십시오.

참고: 이 문서에 대한 VRF NAT 작동을 설명하기 위해 사용된 네트워크에서는 PE 라우터만 포함됩니다. 코어 "P" 라우터가 없습니다. 그러나 중요한 메커니즘은 여전히 볼 수 있습니다.

그림 5: VRF NAT 컨피그레이션 예



이그레스 PE NAT

이 예에서는 gila로 표시된 공급자 에지 라우터와 드래곤이 단순 PE 라우터로 구성됩니다. 공유 서비스 LAN(iguana) 근처의 중앙 PE가 NAT에 대해 구성됩니다. 단일 NAT 풀은 공유 서비스에 액세스해야 하는 각 고객 VPN에 의해 공유됩니다. NAT는 88.1.88.8에서 공유 서비스 호스트로 향하는 패킷에 대해서만 수행됩니다.

이그레스 PE NAT 데이터 포워딩

MPLS를 사용하면 각 패킷이 인그레스 PE에서 네트워크에 진입하여 이그레스 PE에서 MPLS 네트워크를 종료합니다. 인그레스(ingress)에서 이그레스(egress)로 이동하는 레이블 스위칭 라우터의 경로를 LSP(Label Switched Path)라고 합니다. LSP는 단방향입니다. 반환 트래픽에 다른 LSP가 사용됩니다.

이그레스 PE NAT를 사용하는 경우 공유 서비스 사용자의 모든 트래픽에 대해 FEC(forwarding equivalent class)가 효과적으로 정의됩니다. 즉, 공유 서비스 LAN으로 향하는 모든 패킷은 공통 FEC의 멤버입니다. 패킷은 네트워크의 인그레스 에지에서 한 번만 특정 FEC에 할당되고 LSP를 이그레스 PE에 따릅니다. FEC는 특정 레이블을 추가하여 데이터 패킷에 지정됩니다.

VPN에서 공유 서비스로 패킷 흐름

공유 서비스 호스트에 액세스하기 위해 주소 체계가 겹치는 여러 VPN의 디바이스에 대해서는 NAT가 필요합니다. 이그레스 PE에서 NAT가 구성된 경우 네트워크 주소 변환 테이블 엔트리에 중복 주소를 구별하고 올바른 라우팅을 보장하기 위한 VRF 식별자가 포함됩니다.

그림 6: 이그레스 PE NAT로 전송된 패킷

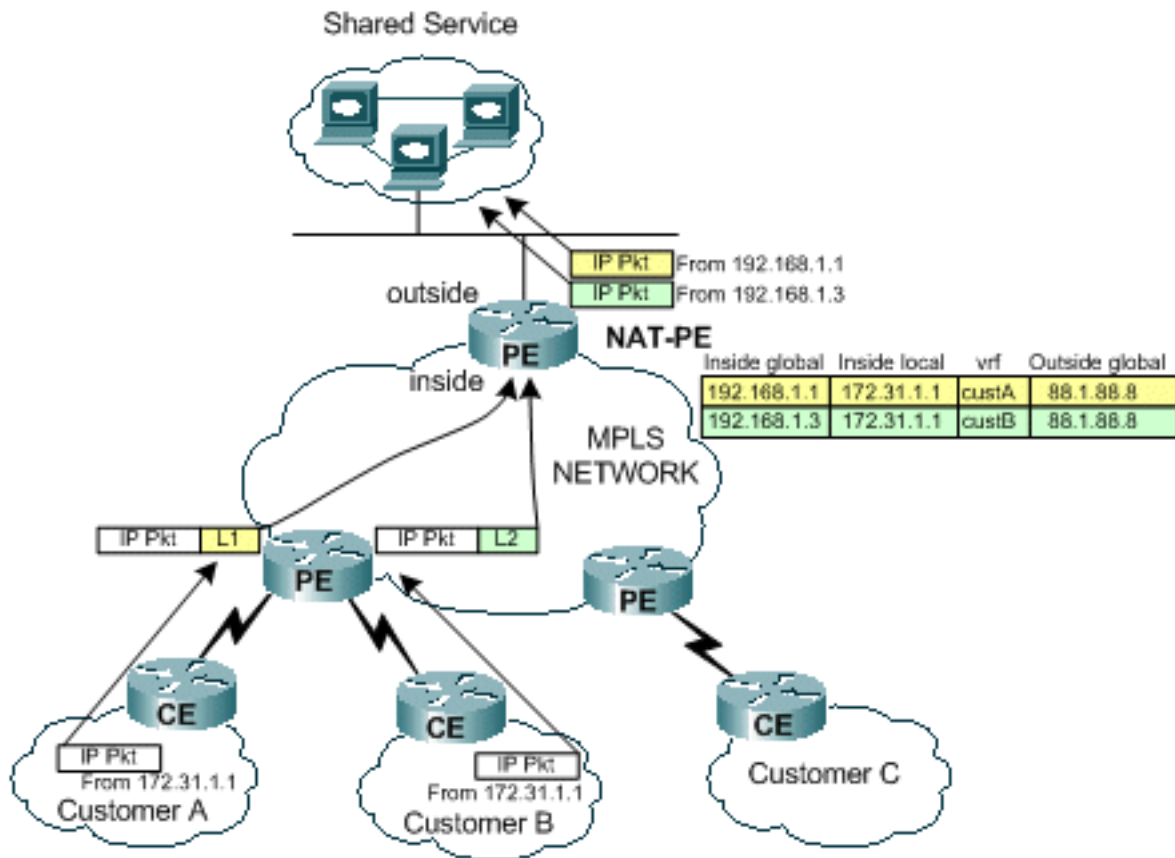


그림 6은 중복 IP 주소 지정 체계가 있는 두 고객 VPN에서 공유 서비스 호스트로 향하는 패킷을 보여줍니다. 이 그림은 88.1.88.8에서 공유 서버로 향하는 소스 주소가 172.31.1.1인 고객 A에서 시작되는 패킷을 보여줍니다. 동일한 소스 IP 주소를 가진 고객 B의 다른 패킷도 동일한 공유 서버로 전송됩니다. 패킷이 PE 라우터에 도달하면 FIB(forwarding information base)의 대상 IP 네트워크에 대해 레이어 3 조치가 수행됩니다.

FIB 항목은 PE 라우터에 레이블 스택을 사용하여 트래픽을 이그레스 PE로 전달하도록 지시합니다. 스택의 하단 레이블은 대상 PE 라우터에 의해 할당됩니다(이 경우 라우터 이구아나).

```
iguana#
show ip cef vrf custA 88.1.88.8
88.1.88.8/32, version 47, epoch 0, cached adjacency 88.1.3.2
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
via 88.1.11.5, 0 dependencies, recursive
  next hop 88.1.3.2, Ethernet1/0 via 88.1.11.5/32
  valid cached adjacency
  tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
```

```
iguana# show ip cef vrf custB 88.1.88.8
88.1.88.8/32, version 77, epoch 0, cached adjacency 88.1.3.2
```



```

0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0, 88.1.3.2, tags imposed: {28}
via 88.1.11.5, 0 dependencies, recursive
  next hop 88.1.3.2, Ethernet1/0 via 88.1.11.5/32
  valid cached adjacency
  tag rewrite with Et1/0, 88.1.3.2, tags imposed: {28}
iguana#

```

표시에서 VRF custA의 패킷에 24(0x18)의 태그 값이 있고 VRF custB의 패킷에 28(0x1C)의 태그 값이 있음을 확인할 수 있습니다.

이 경우 네트워크에 "P" 라우터가 없으므로 추가 태그가 없습니다. 코어 라우터가 있다면 외부 레이블이 지정되었을 것이며 패킷이 이그레스 PE에 도달할 때까지 코어 네트워크 내에서 레이블 스와핑의 정상적인 프로세스가 수행되었을 것입니다.

길라 라우터가 이그레스 PE에 직접 연결되어 있으므로 태그가 추가되기 전에 팝업이 표시되는 것을 확인할 수 있습니다.

```

gila#
show tag-switching forwarding-table
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
16     Pop tag    88.1.1.0/24     0          Et1/1     88.1.2.2
      Pop tag    88.1.1.0/24     0          Et1/0     88.1.3.2
17     Pop tag    88.1.4.0/24     0          Et1/1     88.1.2.2
18     Pop tag    88.1.10.0/24    0          Et1/1     88.1.2.2
19     Pop tag    88.1.11.1/32    0          Et1/1     88.1.2.2
20     Pop tag    88.1.5.0/24     0          Et1/0     88.1.3.2
21     19        88.1.11.10/32   0          Et1/1     88.1.2.2
      22        88.1.11.10/32   0          Et1/0     88.1.3.2
22     20        172.18.60.176/32 0          Et1/1     88.1.2.2
      23        172.18.60.176/32 0          Et1/0     88.1.3.2
23     Untagged  172.31.1.0/24[V] 4980      Fa0/0     10.88.162.6
24     Aggregate 10.88.162.4/30[V] 1920
25     Aggregate 10.88.162.8/30[V] 137104
26     Untagged  172.31.1.0/24[V] 570       Et1/2     10.88.162.14
27     Aggregate 10.88.162.12/30[V] \
      273480
30     Pop tag    88.1.11.5/32    0          Et1/0     88.1.3.2
31     Pop tag    88.1.88.0/24    0          Et1/0     88.1.3.2
32     16        88.1.97.0/24    0          Et1/0     88.1.3.2
33     Pop tag    88.1.99.0/24    0          Et1/0     88.1.3.2
gila#

```

```

gila# show tag-switching forwarding-table 88.1.88.0 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
31     Pop tag    88.1.88.0/24    0          Et1/0     88.1.3.2
      MAC/Encaps=14/14, MRU=1504, Tag Stack{
      005054D92A250090BF9C6C1C8847
      No output feature configured
      Per-packet load-sharing
gila#

```

다음 화면에는 이그레스 PE NAT 라우터(iguana의 인터페이스 E1/0/5)에서 수신한 에코 패킷이 표시됩니다.

From CustA:

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 1 arrived at 16:21:34.8415; frame size is 118 (0076 hex)
      bytes.
DLC: Destination = Station 005054D92A25
DLC: Source       = Station 0090BF9C6C1C
DLC: Ethertype    = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 00018
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:   000. .... = routine
IP:   ...0 .... = normal delay
IP:   .... 0... = normal throughput
IP:   .... .0.. = normal reliability
IP:   .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:   .... ...0 = CE bit - no congestion
IP: Total length = 100 bytes
IP: Identification = 175
IP: Flags        = 0X
IP:   .0.. .... = may fragment
IP:   ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 254 seconds/hops
IP: Protocol      = 1 (ICMP)
IP: Header checksum = 5EC0 (correct)
IP: Source address       = [172.31.1.1]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 4AF1 (correct)
ICMP: Identifier = 4713
ICMP: Sequence number = 6957
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]
```

From CustB:

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 11 arrived at 16:21:37.1558; frame size is 118 (0076 hex)
      bytes.
DLC: Destination = Station 005054D92A25
DLC: Source       = Station 0090BF9C6C1C
DLC: Ethertype    = 8847 (MPLS)
```

```

DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value = 0001C
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value = 1 (Bottom of Stack)
MPLS: Time to Live = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE
bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 100 bytes
IP: Identification = 165
IP: Flags = 0X
IP: .0.. .... = may fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 254 seconds/hops
IP: Protocol = 1 (ICMP)
IP: Header checksum = 5ECA (correct)
IP: Source address = [172.31.1.1]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = AD5E (correct)
ICMP: Identifier = 3365
ICMP: Sequence number = 7935
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

이러한 ping을 수행하면 이그레스 PE 라우터 이구아나의 NAT 테이블에 다음과 같은 항목이 생성됩니다. 위에 표시된 패킷에 대해 생성된 특정 항목은 ICMP 식별자와 일치할 수 있습니다.

```

iguana#
show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp 192.168.1.3:3365 172.31.1.1:3365 88.1.88.8:3365 88.1.88.8:3365
icmp 192.168.1.3:3366 172.31.1.1:3366 88.1.88.8:3366 88.1.88.8:3366
icmp 192.168.1.3:3367 172.31.1.1:3367 88.1.88.8:3367 88.1.88.8:3367
icmp 192.168.1.3:3368 172.31.1.1:3368 88.1.88.8:3368 88.1.88.8:3368
icmp 192.168.1.3:3369 172.31.1.1:3369 88.1.88.8:3369 88.1.88.8:3369
icmp 192.168.1.1:4713 172.31.1.1:4713 88.1.88.8:4713 88.1.88.8:4713
icmp 192.168.1.1:4714 172.31.1.1:4714 88.1.88.8:4714 88.1.88.8:4714
icmp 192.168.1.1:4715 172.31.1.1:4715 88.1.88.8:4715 88.1.88.8:4715
icmp 192.168.1.1:4716 172.31.1.1:4716 88.1.88.8:4716 88.1.88.8:4716
icmp 192.168.1.1:4717 172.31.1.1:4717 88.1.88.8:4717 88.1.88.8:4717

```

```

iguana#
show ip nat translations verbose

Pro Inside global      Inside local      Outside local      Outside global
icmp 192.168.1.3:3365  172.31.1.1:3365   88.1.88.8:3365    88.1.88.8:3365
    create 00:00:34, use 00:00:34, left 00:00:25, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:3366  172.31.1.1:3366   88.1.88.8:3366    88.1.88.8:3366
    create 00:00:34, use 00:00:34, left 00:00:25, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:3367  172.31.1.1:3367   88.1.88.8:3367    88.1.88.8:3367
    create 00:00:34, use 00:00:34, left 00:00:25, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:3368  172.31.1.1:3368   88.1.88.8:3368    88.1.88.8:3368
    create 00:00:34, use 00:00:34, left 00:00:25, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:3369  172.31.1.1:3369   88.1.88.8:3369    88.1.88.8:3369
    create 00:00:34, use 00:00:34, left 00:00:25, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.1:4713  172.31.1.1:4713   88.1.88.8:4713    88.1.88.8:4713
    create 00:00:37, use 00:00:37, left 00:00:22, Map-Id(In): 1,
Pro Inside global      Inside local      Outside local      Outside global
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:4714  172.31.1.1:4714   88.1.88.8:4714    88.1.88.8:4714
    create 00:00:37, use 00:00:37, left 00:00:22, Map-Id(In): 1,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:4715  172.31.1.1:4715   88.1.88.8:4715    88.1.88.8:4715
    create 00:00:37, use 00:00:37, left 00:00:22, Map-Id(In): 1,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:4716  172.31.1.1:4716   88.1.88.8:4716    88.1.88.8:4716
    create 00:00:37, use 00:00:37, left 00:00:22, Map-Id(In): 1,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:4717  172.31.1.1:4717   88.1.88.8:4717    88.1.88.8:4717
    create 00:00:37, use 00:00:37, left 00:00:22, Map-Id(In): 1,
    flags:
extended, use_count: 0, VRF : custA
iguana#

```

공유 서비스에서 오리지널 VPN으로의 패킷 흐름

패킷이 공유 서비스 호스트에 액세스한 디바이스로 다시 전달되면 라우팅 전에 NAT 테이블을 검사합니다(NAT "외부" 인터페이스에서 "내부" 인터페이스로 이동하는 패킷). 각 고유한 항목에는 해당 VRF 식별자가 포함되므로 패킷을 적절하게 변환하고 라우팅할 수 있습니다.

그림 7:공유 서비스 사용자로 다시 전송된 패킷

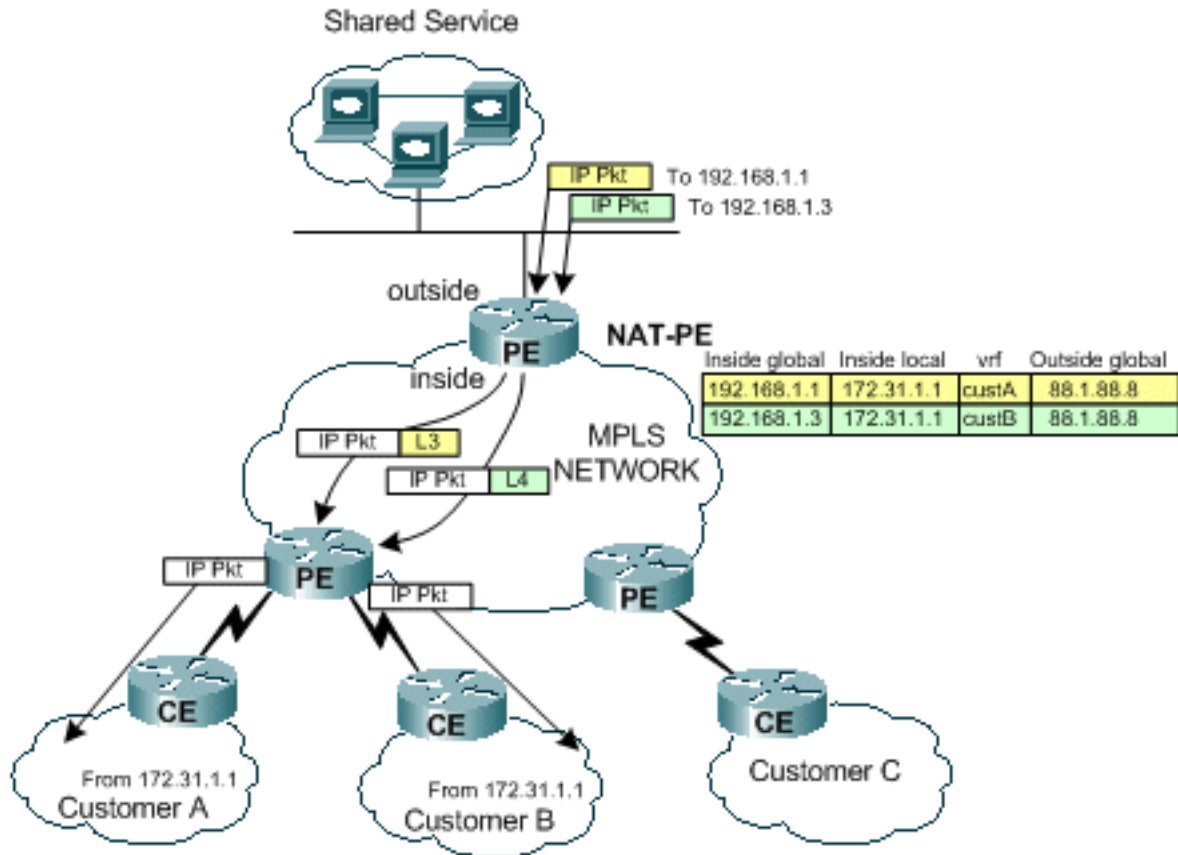


그림 7에 나와 있는 것처럼, 반환 트래픽은 먼저 NAT에서 검사하여 일치하는 변환 엔트리를 찾습니다. 예를 들어 패킷이 대상 192.168.1.1으로 전송됩니다. NAT 테이블이 검색됩니다. 일치가 발견되면 "내부 로컬" 주소(172.31.1.1)으로 적절한 변환을 수행한 다음 NAT 항목의 관련 VRF ID를 사용하여 인접성 조회가 수행됩니다.

```
iguana# show ip cef vrf custA 172.31.1.0
172.31.1.0/24, version 12, epoch 0, cached adjacency 88.1.3.1
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0/5, 88.1.3.1, tags imposed: {23}
via 88.1.11.9, 0 dependencies, recursive
  next hop 88.1.3.1, Ethernet1/0/5 via 88.1.11.9/32
  valid cached adjacency
  tag rewrite with Et1/0/5, 88.1.3.1, tags imposed: {23}

iguana# show ip cef vrf custB 172.31.1.0
172.31.1.0/24, version 18, epoch 0, cached adjacency 88.1.3.1
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0/5, 88.1.3.1, tags imposed: {26}
via 88.1.11.9, 0 dependencies, recursive
  next hop 88.1.3.1, Ethernet1/0/5 via 88.1.11.9/32
  valid cached adjacency
  tag rewrite with Et1/0/5, 88.1.3.1, tags imposed: {26}
iguana#
```

Label 23(0x17)은 VRF custA에서 172.31.1.0/24으로 향하는 트래픽에 사용되며 레이블 26(0x1A)은 VRF custB에서 172.31.1.0/24으로 향하는 패킷에 사용됩니다.

라우터 이쿠아나에서 보낸 에코 응답 패킷에서 확인할 수 있습니다.

To custA:

```
DLC: ----- DLC Header -----
DLC:
DLC: Frame 2 arrived at 16:21:34.8436; frame size is 118 (0076 hex)
      bytes.
DLC: Destination = Station 0090BF9C6C1C
DLC: Source       = Station 005054D92A25
DLC: Ethertype    = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 00017
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value            = 1 (Bottom of Stack)
MPLS: Time to Live           = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 56893
IP: Flags          = 4X
IP:      .1.. .... = don't fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 4131 (correct)
IP: Source address  = [88.1.88.8]
IP: Destination address = [172.31.1.1]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 0 (Echo reply)
ICMP: Code = 0
ICMP: Checksum = 52F1 (correct)
ICMP: Identifier = 4713
ICMP: Sequence number = 6957
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]
```

패킷이 대상 PE 라우터에 도달하면 레이블을 사용하여 패킷을 전송할 적절한 VRF 및 인터페이스를 결정합니다.

gila#

[show mpls forwarding-table](#)

Local	Outgoing	Prefix	Bytes tag	Outgoing	Next Hop
-------	----------	--------	-----------	----------	----------

tag	tag or VC	or Tunnel Id	switched	interface	
16	Pop tag	88.1.1.0/24	0	Et1/1	88.1.2.2
	Pop tag	88.1.1.0/24	0	Et1/0	88.1.3.2
17	Pop tag	88.1.4.0/24	0	Et1/1	88.1.2.2
18	Pop tag	88.1.10.0/24	0	Et1/1	88.1.2.2
19	Pop tag	88.1.11.1/32	0	Et1/1	88.1.2.2
20	Pop tag	88.1.5.0/24	0	Et1/0	88.1.3.2
21	19	88.1.11.10/32	0	Et1/1	88.1.2.2
	22	88.1.11.10/32	0	Et1/0	88.1.3.2
22	20	172.18.60.176/32	0	Et1/1	88.1.2.2
	23	172.18.60.176/32	0	Et1/0	88.1.3.2
23	Untagged	172.31.1.0/24 [V]	6306	Fa0/0	10.88.162.6
24	Aggregate	10.88.162.4/30[V]	1920		
25	Aggregate	10.88.162.8/30[V]	487120		
26	Untagged	172.31.1.0/24 [V]	1896	Et1/2	10.88.162.14
27	Aggregate	10.88.162.12/30[V]	\		
			972200		
30	Pop tag	88.1.11.5/32	0	Et1/0	88.1.3.2
31	Pop tag	88.1.88.0/24	0	Et1/0	88.1.3.2
32	16	88.1.97.0/24	0	Et1/0	88.1.3.2
33	Pop tag	88.1.99.0/24	0	Et1/0	88.1.3.2

gila#

구성

일부 관련 없는 정보가 간결성을 위해 구성에서 제거되었습니다.

```

IGUANA:
!
ip vrf custA
 rd 65002:100
 route-target export 65002:100
 route-target import 65002:100
!
ip vrf custB
 rd 65002:200
 route-target export 65002:200
 route-target import 65002:200
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
!
interface Loopback0
 ip address 88.1.11.5 255.255.255.255
 no ip route-cache
 no ip mroute-cache
!
interface Loopback11
 ip vrf forwarding custA
 ip address 172.16.1.1 255.255.255.255
!
interface Ethernet1/0/0
 ip vrf forwarding custB
 ip address 10.88.163.5 255.255.255.252
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet1/0/4
 ip address 88.1.1.1 255.255.255.0
 ip nat inside

```

```
no ip mroute-cache
tag-switching ip
!
interface Ethernet1/0/5
ip address 88.1.3.2 255.255.255.0
ip nat inside
no ip mroute-cache
tag-switching ip
!
!
interface FastEthernet1/1/0
ip address 88.1.88.1 255.255.255.0
ip nat outside
full-duplex
!
interface FastEthernet5/0/0
ip address 88.1.99.1 255.255.255.0
speed 100
full-duplex
!
router ospf 881
log-adjacency-changes
redistribute static subnets
network 88.1.0.0 0.0.255.255 area 0
!
router bgp 65002
no synchronization
no bgp default ipv4-unicast
bgp log-neighbor-changes
neighbor 88.1.11.1 remote-as 65002
neighbor 88.1.11.1 update-source Loopback0
neighbor 88.1.11.9 remote-as 65002
neighbor 88.1.11.9 update-source Loopback0
neighbor 88.1.11.10 remote-as 65002
neighbor 88.1.11.10 update-source Loopback0
no auto-summary
!
address-family ipv4 multicast
no auto-summary
no synchronization
exit-address-family
!
address-family vpnv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.1 send-community extended
neighbor 88.1.11.9 activate
neighbor 88.1.11.9 send-community extended
no auto-summary
exit-address-family
!
address-family ipv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.9 activate
neighbor 88.1.11.10 activate
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf custB
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family
```



```

!
address-family ipv4 vrf custA
redistribute static
no auto-summary
no synchronization
exit-address-family
!
ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL1 vrf custB overload
ip classless
ip route 88.1.88.0 255.255.255.0 FastEthernet1/1/0
ip route 88.1.97.0 255.255.255.0 FastEthernet5/0/0 88.1.99.2
ip route 88.1.99.0 255.255.255.0 FastEthernet5/0/0 88.1.99.2
ip route 192.168.1.0 255.255.255.0 Null0
ip route vrf custA 88.1.88.8 255.255.255.255 FastEthernet1/1/0 88.1.88.8 global
ip route vrf custB 10.88.208.0 255.255.240.0 10.88.163.6
ip route vrf custB 64.102.0.0 255.255.0.0 10.88.163.6
ip route vrf custB 88.1.88.8 255.255.255.255 FastEthernet1/1/0 88.1.88.8 global
ip route vrf custB 128.0.0.0 255.0.0.0 10.88.163.6
no ip http server
!
access-list 181 permit ip any host 88.1.88.8
!

```

GILA:

```

!
ip vrf custA
rd 65002:100
route-target export 65002:100
route-target import 65002:100
!
ip vrf custB
rd 65002:200
route-target export 65002:200
route-target import 65002:200
!
ip cef
mpls label protocol ldp
tag-switching tdp router-id Loopback0
!
interface Loopback0
ip address 88.1.11.9 255.255.255.255
!
interface FastEthernet0/0
ip vrf forwarding custA
ip address 10.88.162.5 255.255.255.252
duplex full
!
interface Ethernet1/0
ip address 88.1.3.1 255.255.255.0
no ip mroute-cache
duplex half
tag-switching ip
!
interface Ethernet1/1
ip address 88.1.2.1 255.255.255.0
no ip mroute-cache
duplex half
tag-switching ip
!

```

```

interface Ethernet1/2
 ip vrf forwarding custB
 ip address 10.88.162.13 255.255.255.252
 ip ospf cost 100
 duplex half
!
interface FastEthernet2/0
 ip vrf forwarding custA
 ip address 10.88.162.9 255.255.255.252
 duplex full
!
router ospf 881
 log-adjacency-changes
 redistribute static subnets
 network 88.1.0.0 0.0.255.255 area 0
 default-metric 30
!
router bgp 65002
 no synchronization
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 88.1.11.1 remote-as 65002
 neighbor 88.1.11.1 update-source Loopback0
 neighbor 88.1.11.1 activate
 neighbor 88.1.11.5 remote-as 65002
 neighbor 88.1.11.5 update-source Loopback0
 neighbor 88.1.11.5 activate
 no auto-summary
!
 address-family ipv4 vrf custB
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 exit-address-family
!
 address-family ipv4 vrf custA
 redistribute connected
 redistribute static
 no auto-summary
 no synchronization
 exit-address-family
!
 address-family vpnv4
 neighbor 88.1.11.1 activate
 neighbor 88.1.11.1 send-community extended
 neighbor 88.1.11.5 activate
 neighbor 88.1.11.5 send-community extended
 no auto-summary
 exit-address-family
!
ip classless
ip route vrf custA 172.31.1.0 255.255.255.0 FastEthernet0/0 10.88.162.6
ip route vrf custB 172.31.1.0 255.255.255.0 Ethernet1/2 10.88.162.14
!

```

라우터 드래곤의 컨피그레이션은 길라와 매우 유사할 수 있습니다.

경로 대상 가져오기/내보내기가 허용되지 않음

공유 서비스 네트워크가 VRF 인스턴스 자체로 구성된 경우 이그레스 PE의 중앙 NAT는 가능하지 않습니다. 이는 수신 패킷을 구분할 수 없으며 원래 서브넷에 대한 경로가 이그레스 PE NAT에 하나

만 있기 때문입니다.

참고: 다음 표시는 유효하지 않은 구성의 결과를 설명하기 위한 것입니다.

공유 서비스 네트워크가 VRF 인스턴스(VRF 이름 = sserver)로 정의되도록 샘플 네트워크가 구성되었습니다. 이제 인그레스 PE에 CEF 테이블이 표시되면 다음과 같습니다.

```
gila# show ip cef vrf custA 88.1.88.0
88.1.88.0/24, version 45, epoch 0, cached adjacency 88.1.3.2
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
via 88.1.11.5, 0 dependencies, recursive
  next hop 88.1.3.2, Ethernet1/0 via 88.1.11.5/32
  valid cached adjacency
  tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
gila#
```

```
gila# show ip cef vrf custB 88.1.88.0
88.1.88.0/24, version 71, epoch 0, cached adjacency 88.1.3.2
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
via 88.1.11.5, 0 dependencies, recursive
  next hop 88.1.3.2, Ethernet1/0 via 88.1.11.5/32
  valid cached adjacency
  tag rewrite with Et1/0, 88.1.3.2, tags imposed: {24}
gila#
```

```
iguana#
show tag-switching forwarding vrftags 24
Local   Outgoing   Prefix           Bytes tag   Outgoing   Next Hop
tag     tag or VC  or Tunnel Id     switched   interface
24     Aggregate  88.1.88.0/24[V]  10988
iguana#
```

참고: 태그 값 24가 VRF custA 및 VRF custB에 대해 어떻게 적용되는지 확인합니다.

이 표시는 공유 서비스 VRF 인스턴스 "sserver"에 대한 라우팅 테이블을 보여줍니다.

```
iguana#
show ip route vrf sserver 172.31.1.1
Routing entry for 172.31.1.0/24
  Known via "bgp 65002", distance 200, metric 0, type internal
  Last update from 88.1.11.9 1d01h ago
  Routing Descriptor Blocks:
  * 88.1.11.9 (Default-IP-Routing-Table), from 88.1.11.9, 1d01h ago
    Route metric is 0, traffic share count is 1
    AS Hops 0
```

참고: 이그레스 PE 라우터의(iguana) 관점에서 대상 네트워크에 대한 경로는 하나만 있습니다.

따라서 여러 고객 VPN의 트래픽을 구분할 수 없으며 반환 트래픽이 적절한 VPN에 도달할 수 없습

니다.공유 서비스를 VRF 인스턴스로 정의해야 하는 경우 NAT 기능을 인그레스 PE로 이동해야 합니다.

인그레스 PE NAT

이 예에서는 gila로 표시된 공급자 에지 라우터와 드래곤이 NAT용으로 구성됩니다.NAT 풀은 공유 서비스에 액세스해야 하는 연결된 각 고객 VPN에 대해 정의됩니다.해당 풀은 NATed인 각 고객 네트워크 주소에 사용됩니다.NAT는 88.1.88.8에서 공유 서비스 호스트로 향하는 패킷에 대해서만 수행됩니다.

```
ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat pool SSPOOL2 192.168.2.1 192.168.2.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL2 vrf custB overload
```

참고: 이 시나리오에서는 공유 풀이 지원되지 않습니다.이그레스 PE에 있는 공유 서비스 LAN이 일반 인터페이스를 통해 연결된 경우 NAT 풀을 공유할 수 있습니다.

neuse 및 capefear8에 연결된 각 네트워크 내의 중복 주소(172.31.1.1)에서 소싱된 ping은 다음 NAT 항목을 생성합니다.

길라에서는:

```
gila#
show ip nat translations
Pro Inside global      Inside local          Outside local         Outside global
icmp 192.168.1.1:2139 172.31.1.1:2139     88.1.88.8:2139      88.1.88.8:2139
icmp 192.168.1.1:2140 172.31.1.1:2140     88.1.88.8:2140      88.1.88.8:2140
icmp 192.168.1.1:2141 172.31.1.1:2141     88.1.88.8:2141      88.1.88.8:2141
icmp 192.168.1.1:2142 172.31.1.1:2142     88.1.88.8:2142      88.1.88.8:2142
icmp 192.168.1.1:2143 172.31.1.1:2143     88.1.88.8:2143      88.1.88.8:2143
icmp 192.168.2.2:676  172.31.1.1:676      88.1.88.8:676       88.1.88.8:676
icmp 192.168.2.2:677  172.31.1.1:677      88.1.88.8:677       88.1.88.8:677
icmp 192.168.2.2:678  172.31.1.1:678      88.1.88.8:678       88.1.88.8:678
icmp 192.168.2.2:679  172.31.1.1:679      88.1.88.8:679       88.1.88.8:679
icmp 192.168.2.2:680  172.31.1.1:680      88.1.88.8:680       88.1.88.8:680
```

참고: 동일한 내부 로컬 주소(172.31.1.1)은 소스 VRF에 따라 정의된 각 풀로 변환됩니다.VRF는 show ip nat translation verbose 명령에서 볼 수 있습니다.

```
gila# show ip nat translations verbose
Pro Inside global      Inside local          Outside local         Outside global
icmp 192.168.1.1:2139 172.31.1.1:2139     88.1.88.8:2139      88.1.88.8:2139
      create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 3,
      flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:2140 172.31.1.1:2140     88.1.88.8:2140      88.1.88.8:2140
      create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 3,
      flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:2141 172.31.1.1:2141     88.1.88.8:2141      88.1.88.8:2141
      create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 3,
      flags:
extended, use_count: 0, VRF : custA
```

```

icmp 192.168.1.1:2142 172.31.1.1:2142 88.1.88.8:2142 88.1.88.8:2142
  create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 3,
  flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:2143 172.31.1.1:2143 88.1.88.8:2143 88.1.88.8:2143
  create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 3,
  flags:
extended, use_count: 0, VRF : custA
icmp 192.168.2.2:676 172.31.1.1:676 88.1.88.8:676 88.1.88.8:676
  create 00:00:10, use 00:00:10, left 00:00:49, Map-Id(In): 2,
  flags:
extended, use_count: 0, VRF : custB
icmp 192.168.2.2:677 172.31.1.1:677 88.1.88.8:677 88.1.88.8:677
  create 00:00:10, use 00:00:10, left 00:00:49, Map-Id(In): 2,
  flags:
extended, use_count: 0, VRF : custB
icmp 192.168.2.2:678 172.31.1.1:678 88.1.88.8:678 88.1.88.8:678
  create 00:00:10, use 00:00:10, left 00:00:49, Map-Id(In): 2,
  flags:
extended, use_count: 0, VRF : custB
icmp 192.168.2.2:679 172.31.1.1:679 88.1.88.8:679 88.1.88.8:679
  create 00:00:10, use 00:00:10, left 00:00:49, Map-Id(In): 2,
  flags:
extended, use_count: 0, VRF : custB
icmp 192.168.2.2:680 172.31.1.1:680 88.1.88.8:680 88.1.88.8:680
  create 00:00:10, use 00:00:10, left 00:00:49, Map-Id(In): 2,
  flags:
extended, use_count: 0, VRF : custB

```

이 표시는 고객 A 및 고객 B에 대해 로컬로 연결된 각 VPN에 대한 라우팅 정보를 보여줍니다.

```

gila# show ip route vrf custA
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

```

Gateway of last resort is 88.1.11.1 to network 0.0.0.0

```

172.18.0.0/32 is subnetted, 2 subnets
B    172.18.60.179 [200/0] via 88.1.11.1, 00:03:59
B    172.18.60.176 [200/0] via 88.1.11.1, 00:03:59
172.31.0.0/24 is subnetted, 1 subnets
S    172.31.1.0 [1/0] via 10.88.162.6, FastEthernet0/0
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
B    10.88.0.0/20 [200/0] via 88.1.11.1, 00:03:59
B    10.88.32.0/20 [200/0] via 88.1.11.1, 00:03:59
C    10.88.162.4/30 is directly connected, FastEthernet0/0
C    10.88.162.8/30 is directly connected, FastEthernet2/0
B    10.88.161.8/30 [200/0] via 88.1.11.1, 00:04:00
88.0.0.0/24 is subnetted, 2 subnets
B    88.1.88.0 [200/0] via 88.1.11.5, 00:04:00
B    88.1.99.0 [200/0] via 88.1.11.5, 00:04:00
S    192.168.1.0/24 is directly connected, Null0
B*  0.0.0.0/0 [200/0] via 88.1.11.1, 00:04:00

```

```
gila# show ip route vrf custB
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
        64.0.0.0/16 is subnetted, 1 subnets
B       64.102.0.0 [200/0] via 88.1.11.5, 1d21h
        172.18.0.0/32 is subnetted, 2 subnets
B       172.18.60.179 [200/0] via 88.1.11.1, 1d21h
B       172.18.60.176 [200/0] via 88.1.11.1, 1d21h
        172.31.0.0/24 is subnetted, 1 subnets
S       172.31.1.0 [1/0] via 10.88.162.14, Ethernet1/2
        10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
B       10.88.194.16/28 [200/100] via 88.1.11.1, 1d20h
B       10.88.208.0/20 [200/0] via 88.1.11.5, 1d21h
B       10.88.194.4/30 [200/100] via 88.1.11.1, 1d20h
B       10.88.163.4/30 [200/0] via 88.1.11.5, 1d21h
B       10.88.161.4/30 [200/0] via 88.1.11.1, 1d21h
C       10.88.162.12/30 is directly connected, Ethernet1/2
        11.0.0.0/24 is subnetted, 1 subnets
B       11.1.1.0 [200/100] via 88.1.11.1, 1d20h
        88.0.0.0/24 is subnetted, 2 subnets
B       88.1.88.0 [200/0] via 88.1.11.5, 1d21h
B       88.1.99.0 [200/0] via 88.1.11.5, 1d21h
S       192.168.2.0/24 is directly connected, Null0
B       128.0.0.0/8 [200/0] via 88.1.11.5, 1d21h
```

참고: 각 NAT 플레에 대한 경로가 고정 컨피그레이션에서 추가되었습니다.이러한 서브넷은 이후에 이그레스 PE 라우터 이구아나에서 공유 서버 VRF로 가져옵니다.

```
iguana# show ip route vrf sserver
```

```
Routing Table: sserver
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        I - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
        64.0.0.0/16 is subnetted, 1 subnets
B       64.102.0.0 [20/0] via 10.88.163.6 (custB), 1d20h
        172.18.0.0/32 is subnetted, 2 subnets
B       172.18.60.179 [200/0] via 88.1.11.1, 1d20h
B       172.18.60.176 [200/0] via 88.1.11.1, 1d20h
        172.31.0.0/24 is subnetted, 1 subnets
B       172.31.1.0 [200/0] via 88.1.11.9, 1d05h
        10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
```

```

B      10.88.194.16/28 [200/100] via 88.1.11.1, 1d20h
B      10.88.208.0/20 [20/0] via 10.88.163.6 (custB), 1d20h
B      10.88.194.4/30 [200/100] via 88.1.11.1, 1d20h
B      10.88.162.4/30 [200/0] via 88.1.11.9, 1d20h
B      10.88.163.4/30 is directly connected, 1d20h, Ethernet1/0/0
B      10.88.161.4/30 [200/0] via 88.1.11.1, 1d20h
B      10.88.162.8/30 [200/0] via 88.1.11.9, 1d20h
B      10.88.162.12/30 [200/0] via 88.1.11.9, 1d20h
      11.0.0.0/24 is subnetted, 1 subnets
B      11.1.1.0 [200/100] via 88.1.11.1, 1d20h
      12.0.0.0/24 is subnetted, 1 subnets
S      12.12.12.0 [1/0] via 88.1.99.10
      88.0.0.0/24 is subnetted, 3 subnets
C      88.1.88.0 is directly connected, FastEthernet1/1/0
S      88.1.97.0 [1/0] via 88.1.99.10
C      88.1.99.0 is directly connected, FastEthernet5/0/0
B      192.168.1.0/24 [200/0] via 88.1.11.9, 1d20h
B      192.168.2.0/24 [200/0] via 88.1.11.9, 01:59:23
B      128.0.0.0/8 [20/0] via 10.88.163.6 (custB), 1d20h

```

구성

일부 관련 없는 정보가 간결성을 위해 구성에서 제거되었습니다.

GILA:

```

ip vrf custA
  rd 65002:100
  route-target export 65002:100
  route-target export 65002:1001
  route-target import 65002:100
!
ip vrf custB
  rd 65002:200
  route-target export 65002:200
  route-target export 65002:2001
  route-target import 65002:200
  route-target import 65002:10
!
ip cef
mpls label protocol ldp
!

interface Loopback0
  ip address 88.1.11.9 255.255.255.255
!
interface FastEthernet0/0
  ip vrf forwarding custA
  ip address 10.88.162.5 255.255.255.252
  ip nat inside
  duplex full
!
interface Ethernet1/0
  ip address 88.1.3.1 255.255.255.0
  ip nat outside
  no ip mroute-cache
  duplex half
  tag-switching ip
!
interface Ethernet1/1
  ip address 88.1.2.1 255.255.255.0
  ip nat outside
  no ip mroute-cache

```

```

duplex half
tag-switching ip
!
interface Ethernet1/2
  ip vrf forwarding custB
  ip address 10.88.162.13 255.255.255.252
  ip nat inside
  duplex half
!
router ospf 881
  log-adjacency-changes
  redistribute static subnets
  network 88.1.0.0 0.0.255.255 area 0
  default-metric 30
!
router bgp 65002
  no synchronization
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 88.1.11.1 remote-as 65002
  neighbor 88.1.11.1 update-source Loopback0
  neighbor 88.1.11.1 activate
  neighbor 88.1.11.5 remote-as 65002
  neighbor 88.1.11.5 update-source Loopback0
  neighbor 88.1.11.5 activate
  no auto-summary
  !
  address-family ipv4 vrf custB
  redistribute connected
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family ipv4 vrf custA
  redistribute connected
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family vpv4
  neighbor 88.1.11.1 activate
  neighbor 88.1.11.1 send-community extended
  neighbor 88.1.11.5 activate
  neighbor 88.1.11.5 send-community extended
  no auto-summary
  exit-address-family
!
ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat pool SSPOOL2 192.168.2.1 192.168.2.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL2 vrf custB overload
ip classless
ip route vrf custA 172.31.1.0 255.255.255.0 FastEthernet0/0 10.88.162.6
ip route vrf custA 192.168.1.0 255.255.255.0 Null0
ip route vrf custB 172.31.1.0 255.255.255.0 Ethernet1/2 10.88.162.14
ip route vrf custB 192.168.2.0 255.255.255.0 Null0
!
access-list 181 permit ip any host 88.1.88.8
!

```

참고: 고객 네트워크를 접하는 인터페이스는 NAT "내부" 인터페이스로 지정되고 MPLS 인터페이스는 NAT "외부" 인터페이스로 지정됩니다.


```
iguana:
ip vrf custB
  rd 65002:200
  route-target export 65002:200
  route-target export 65002:2001
  route-target import 65002:200
  route-target import 65002:10
!
ip vrf sserver
  rd 65002:10
  route-target export 65002:10
  route-target import 65002:2001
  route-target import 65002:1001
!
ip cef distributed
mpls label protocol ldp
!

interface Loopback0
  ip address 88.1.11.5 255.255.255.255
  no ip route-cache
  no ip mroute-cache
!
interface Ethernet1/0/0
  ip vrf forwarding custB
  ip address 10.88.163.5 255.255.255.252
  no ip route-cache
  no ip mroute-cache
!
interface Ethernet1/0/4
  ip address 88.1.1.1 255.255.255.0
  no ip route-cache
  no ip mroute-cache
  tag-switching ip
!
interface Ethernet1/0/5
  ip address 88.1.3.2 255.255.255.0
  no ip route-cache
  no ip mroute-cache
  tag-switching ip
!
interface FastEthernet1/1/0
  ip vrf forwarding sserver
  ip address 88.1.88.1 255.255.255.0
  no ip route-cache
  no ip mroute-cache
  full-duplex
!
router ospf 881
  log-adjacency-changes
  redistribute static subnets
  network 88.1.0.0 0.0.255.255 area 0
!
router bgp 65002
  no synchronization
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 88.1.11.1 remote-as 65002
  neighbor 88.1.11.1 update-source Loopback0
  neighbor 88.1.11.9 remote-as 65002
  neighbor 88.1.11.9 update-source Loopback0
  neighbor 88.1.11.10 remote-as 65002
```

```

neighbor 88.1.11.10 update-source Loopback0
no auto-summary
!
address-family ipv4 multicast
no auto-summary
no synchronization
exit-address-family
!
address-family vpv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.1 send-community extended
neighbor 88.1.11.9 activate
neighbor 88.1.11.9 send-community extended
no auto-summary
exit-address-family
!
address-family ipv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.9 activate
neighbor 88.1.11.10 activate
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf sserver
redistribute connected
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf custB
redistribute connected
redistribute static
no auto-summary
no synchronization
exit-address-family

```

라우터 드래곤의 컨피그레이션은 길라와 매우 유사할 수 있습니다.

[인그레스 PE NAT 이후 중앙 PE에 도착하는 패킷](#)

아래 추적은 대상 공유 서비스 네트워크가 VRF 인스턴스로 구성된 경우 고유한 NAT 풀에 대한 요구 사항을 보여줍니다. 다시 [그림 5](#)의 다이어그램을 참조하십시오. 아래 표시된 패킷은 라우터 이구 하나의 MPLS IP 인터페이스 e1/0/5에 들어갈 때 캡처되었습니다.

[고객 A VPN의 에코](#)

여기에서는 VRF custA의 소스 IP 주소 172.31.1.1에서 에코 요청을 수신합니다.소스 주소는 NAT 컨피그레이션에서 지정한 대로 192.168.1.1으로 변환되었습니다.

```

ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload

```

```

DLC: ----- DLC Header -----
      DLC:
      DLC: Frame 1 arrived at 09:15:29.8157; frame size is 118 (0076 hex)
          bytes.

```

```

DLC: Destination = Station 005054D92A25
DLC: Source      = Station 0090BF9C6C1C
DLC: Ethertype   = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 00019
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 0
IP: Flags          = 0X
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 4AE6 (correct)
IP: Source address       = [192.168.1.1]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 932D (correct)
ICMP: Identifier = 3046
ICMP: Sequence number = 3245
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]
ICMP:

```

고객 B VPN에서 에코

여기에서는 VRF custB의 소스 IP 주소 172.31.1.1에서 에코 요청을 수신합니다. 소스 주소는 NAT 컨피그레이션에서 지정한 대로 192.168.2.1으로 변환되었습니다.

```

ip nat pool SSPOOL2 192.168.2.1 192.168.2.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL2 vrf custB overload

```

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 11 arrived at 09:15:49.6623; frame size is 118 (0076 hex)
      bytes.

```

```

DLC: Destination = Station 005054D92A25
DLC: Source      = Station 0090BF9C6C1C
DLC: Ethertype   = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 00019
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 15
IP: Flags          = 0X
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 49D6 (correct)
IP: Source address       = [192.168.2.2]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = AB9A (correct)
ICMP: Identifier = 4173
ICMP: Sequence number = 4212
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

참고: MPLS 레이블 값은 위에 표시된 두 패킷 모두에서 0019입니다.

[고객 A VPN에 에코 응답](#)

다음으로, VRF custA에서 목적지 IP 주소 192.168.1.1으로 되돌아가는 에코 회신이 표시됩니다. 수신 주소는 인그레스 PE NAT 기능에 의해 172.31.1.1으로 변환됩니다.

To VRF custA:

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 2 arrived at 09:15:29.8198; frame size is 118 (0076 hex)
      bytes.
DLC: Destination = Station 0090BF9C6C1C

```

```

DLC: Source      = Station 005054D92A25
DLC: Ethertype   = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 0001A
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 18075
IP: Flags          = 4X
IP:      .1.. .... = don't fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = C44A (correct)
IP: Source address  = [88.1.88.8]
IP: Destination address = [192.168.1.1]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 0 (Echo reply)
ICMP: Code = 0
ICMP: Checksum = 9B2D (correct)
ICMP: Identifier = 3046
ICMP: Sequence number = 3245
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]
ICMP:

```

고객 B VPN에 에코 응답

여기에서는 VRF custB의 대상 IP 주소 192.168.1.1으로 되돌아가는 에코 회신을 볼 수 있습니다. 수신 주소는 인그레스 PE NAT 기능에 의해 172.31.1.1으로 변환됩니다.

To VRF custB:

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 12 arrived at 09:15:49.6635; frame size is 118 (0076 hex) bytes.
DLC: Destination = Station 0090BF9C6C1C
DLC: Source      = Station 005054D92A25
DLC: Ethertype   = 8847 (MPLS)
DLC:

```

```

MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value = 0001D
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value = 1 (Bottom of Stack)
MPLS: Time to Live = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 100 bytes
IP: Identification = 37925
IP: Flags = 4X
IP: .1.. .... = don't fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 254 seconds/hops
IP: Protocol = 1 (ICMP)
IP: Header checksum = 75BF (correct)
IP: Source address = [88.1.88.8]
IP: Destination address = [192.168.2.2]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 0 (Echo reply)
ICMP: Code = 0
ICMP: Checksum = B39A (correct)
ICMP: Identifier = 4173
ICMP: Sequence number = 4212
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

참고: 반환 패킷에서 MPLS 레이블 값이 포함되고 다릅니다.001A(VRF custA) 및 001D(VRF custB용)

[고객 A VPN에서 에코 - 대상이 일반 인터페이스입니다.](#)

다음 패킷 집합은 공유 서비스 LAN에 대한 인터페이스가 VRF 인스턴스의 일부가 아닌 일반 인터페이스인 경우 차이를 보여줍니다.여기서는 중복 IP 주소가 있는 두 로컬 VPN에 대해 공통 풀을 사용하도록 컨피그레이션이 변경되었습니다.

```

ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL1 vrf custB overload

```

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 1 arrived at 09:39:19.6580; frame size is 118 (0076 hex)
bytes.

```

```

DLC: Destination = Station 005054D92A25
DLC: Source      = Station 0090BF9C6C1C
DLC: Ethertype   = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 00019
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 55
IP: Flags          = 0X
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 4AAF (correct)
IP: Source address       = [192.168.1.1]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 0905 (correct)
ICMP: Identifier = 874
ICMP: Sequence number = 3727
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

고객 B VPN에서 에코 - 대상이 일반 인터페이스임

여기에서는 VRF custB의 소스 IP 주소 172.31.1.1에서 에코 요청을 수신합니다. NAT 컨피그레이션에서 지정한 대로 소스 주소가 192.168.1.3(공통 풀 SSPOOL1에서)로 변환되었습니다.

```

ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL1 vrf custB overload

```

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 11 arrived at 09:39:26.4971; frame size is 118 (0076 hex)
      bytes.

```

```

DLC: Destination = Station 005054D92A25
DLC: Source      = Station 0090BF9C6C1C
DLC: Ethertype   = 8847 (MPLS)
DLC:
MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS: Label Value           = 0001F
MPLS: Reserved For Experimental Use = 0
MPLS: Stack Value           = 1 (Bottom of Stack)
MPLS: Time to Live          = 254 (hops)
MPLS:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP:      000. .... = routine
IP:      ...0 .... = normal delay
IP:      .... 0... = normal throughput
IP:      .... .0.. = normal reliability
IP:      .... ..0. = ECT bit - transport protocol will ignore the CE
      bit
IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 75
IP: Flags          = 0X
IP:      .0.. .... = may fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 4A99 (correct)
IP: Source address       = [192.168.1.3]
IP: Destination address = [88.1.88.8]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 8 (Echo)
ICMP: Code = 0
ICMP: Checksum = 5783 (correct)
ICMP: Identifier = 4237
ICMP: Sequence number = 977
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

참고: 이그레스 PE의 인터페이스가 일반 인터페이스인 경우(VRF 인스턴스가 아님), 적용된 레이블은 다릅니다.이 경우 0x19 및 0x1F.

[고객에게 에코 응답 A VPN - 대상이 일반 인터페이스입니다.](#)

다음으로, VRF custA에서 목적지 IP 주소 192.168.1.1으로 되돌아가는 에코 회신이 표시됩니다. 수신 주소는 인그레스 PE NAT 기능에 의해 172.31.1.1으로 변환됩니다.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 2 arrived at 09:39:19.6621; frame size is 114 (0072 hex)
      bytes.
DLC: Destination = Station 0090BF9C6C1C
DLC: Source      = Station 005054D92A25

```



```

DLC: Ethertype = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE
bit
IP: .... ...0 = CE bit - no congestion
IP: Total length = 100 bytes
IP: Identification = 54387
IP: Flags = 4X
IP: .1.. .... = don't fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 254 seconds/hops
IP: Protocol = 1 (ICMP)
IP: Header checksum = 3672 (correct)
IP: Source address = [88.1.88.8]
IP: Destination address = [192.168.1.1]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 0 (Echo reply)
ICMP: Code = 0
ICMP: Checksum = 1105 (correct)
ICMP: Identifier = 874
ICMP: Sequence number = 3727
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

고객 B VPN에 에코 응답 - 대상은 일반 인터페이스입니다.

여기에서는 VRF custB의 대상 IP 주소 192.168.1.3으로 되돌아가는 에코 회신을 볼 수 있습니다. 수신 주소는 인그레스 PE NAT 기능에 의해 172.31.1.1으로 변환됩니다.

```

DLC: ----- DLC Header -----
DLC:
DLC: Frame 12 arrived at 09:39:26.4978; frame size is 114 (0072 hex)
bytes.
DLC: Destination = Station 0090BF9C6C1C
DLC: Source = Station 005054D92A25
DLC: Ethertype = 0800 (IP)
DLC:
IP: ----- IP Header -----
IP:
IP: Version = 4, header length = 20 bytes
IP: Type of service = 00
IP: 000. .... = routine
IP: ...0 .... = normal delay
IP: .... 0... = normal throughput
IP: .... .0.. = normal reliability
IP: .... ..0. = ECT bit - transport protocol will ignore the CE
bit

```

```

IP:      .... ...0 = CE bit - no congestion
IP: Total length   = 100 bytes
IP: Identification = 61227
IP: Flags         = 4X
IP:      .1.. .... = don't fragment
IP:      ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live   = 254 seconds/hops
IP: Protocol       = 1 (ICMP)
IP: Header checksum = 1BB8 (correct)
IP: Source address  = [88.1.88.8]
IP: Destination address = [192.168.1.3]
IP: No options
IP:
ICMP: ----- ICMP header -----
ICMP:
ICMP: Type = 0 (Echo reply)
ICMP: Code = 0
ICMP: Checksum = 5F83 (correct)
ICMP: Identifier = 4237
ICMP: Sequence number = 977
ICMP: [72 bytes of data]
ICMP:
ICMP: [Normal end of "ICMP header".]

```

참고: 회신은 전역 주소로 보내지므로 VRF 레이블이 추가되지 않습니다.

일반 인터페이스로 정의된 공유 서비스 LAN 세그먼트에 대한 종료 인터페이스를 사용하면 공통 풀이 허용됩니다. ping은 라우터 길라에 다음 NAT 항목을 생성합니다.

```

gila# show ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
icmp 192.168.1.3:4237  172.31.1.1:4237  88.1.88.8:4237   88.1.88.8:4237
icmp 192.168.1.3:4238  172.31.1.1:4238  88.1.88.8:4238   88.1.88.8:4238
icmp 192.168.1.3:4239  172.31.1.1:4239  88.1.88.8:4239   88.1.88.8:4239
icmp 192.168.1.3:4240  172.31.1.1:4240  88.1.88.8:4240   88.1.88.8:4240
icmp 192.168.1.3:4241  172.31.1.1:4241  88.1.88.8:4241   88.1.88.8:4241
icmp 192.168.1.1:874   172.31.1.1:874   88.1.88.8:874    88.1.88.8:874
icmp 192.168.1.1:875   172.31.1.1:875   88.1.88.8:875    88.1.88.8:875
icmp 192.168.1.1:876   172.31.1.1:876   88.1.88.8:876    88.1.88.8:876
icmp 192.168.1.1:877   172.31.1.1:877   88.1.88.8:877    88.1.88.8:877
icmp 192.168.1.1:878   172.31.1.1:878   88.1.88.8:878    88.1.88.8:878
gila#

```

```

gila# show ip nat tr ver
Pro Inside global      Inside local      Outside local     Outside global
icmp 192.168.1.3:4237  172.31.1.1:4237  88.1.88.8:4237   88.1.88.8:4237
    create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:4238  172.31.1.1:4238  88.1.88.8:4238   88.1.88.8:4238
    create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:4239  172.31.1.1:4239  88.1.88.8:4239   88.1.88.8:4239
    create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.3:4240  172.31.1.1:4240  88.1.88.8:4240   88.1.88.8:4240
    create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 2,
    flags:

```

```

extended, use_count: 0, VRF : custB
icmp 192.168.1.3:4241 172.31.1.1:4241 88.1.88.8:4241 88.1.88.8:4241
    create 00:00:08, use 00:00:08, left 00:00:51, Map-Id(In): 2,
    flags:
extended, use_count: 0, VRF : custB
icmp 192.168.1.1:874 172.31.1.1:874 88.1.88.8:874 88.1.88.8:874
    create 00:00:16, use 00:00:16, left 00:00:43, Map-Id(In): 3,
Pro Inside global Inside local Outside local Outside global
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:875 172.31.1.1:875 88.1.88.8:875 88.1.88.8:875
    create 00:00:18, use 00:00:18, left 00:00:41, Map-Id(In): 3,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:876 172.31.1.1:876 88.1.88.8:876 88.1.88.8:876
    create 00:00:18, use 00:00:18, left 00:00:41, Map-Id(In): 3,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:877 172.31.1.1:877 88.1.88.8:877 88.1.88.8:877
    create 00:00:18, use 00:00:18, left 00:00:41, Map-Id(In): 3,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:878 172.31.1.1:878 88.1.88.8:878 88.1.88.8:878
    create 00:00:18, use 00:00:18, left 00:00:41, Map-Id(In): 3,
    flags:
extended, use_count: 0, VRF : custA

```

gila#

debug ip nat vrf

IP NAT VRF debugging is on

gila#

```

.Jan 2 09:34:54 EST: NAT-TAGSW(p) : Tag Pkt s=172.18.60.179, d=10.88.162.9, vrf=custA
.Jan 2 09:35:02 EST: NAT-TAGSW(p) : Tag Pkt s=172.18.60.179, d=10.88.162.13, vrf=custB
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:19 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custB
.Jan 2 09:35:19 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:19 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custB
.Jan 2 09:35:19 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:19 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custB
.Jan 2 09:35:19 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:19 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custB
.Jan 2 09:35:19 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:19 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custB
.Jan 2 09:35:19 EST: NAT-ip2tag: Punting to process

```

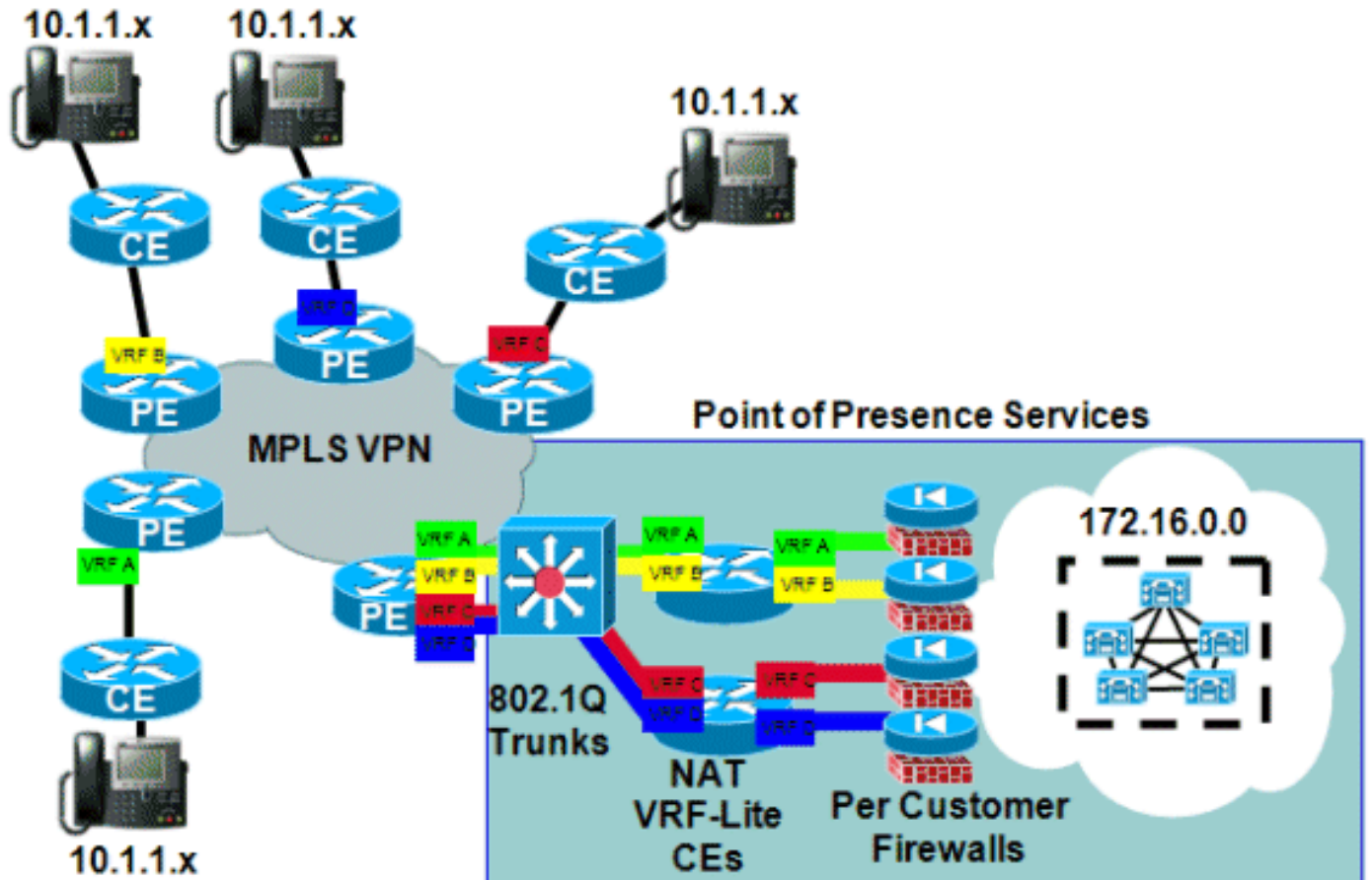
gila#

서비스 예

그림 8에 공유 가상 IP PBX 서비스의 예가 나와 있습니다. 앞에서 설명한 인그레스 및 이그레스 예제의 변종을 보여줍니다.

이 설계에서는 NAT 기능을 수행하는 라우터 집합에 의해 공유 VoIP 서비스가 전면적으로 종료됩니다. 이러한 라우터에는 VRF-Lite라는 기능을 사용하는 여러 VRF 인터페이스가 있습니다. 그러면 트래픽이 공유 Cisco CallManager 클러스터로 이동합니다. 방화벽 서비스는 회사별로 제공됩니다. 회사 간 통화는 방화벽을 통과해야 하며 회사 내부 통화는 회사의 내부 주소 지정 체계를 사용하여 고객 VPN에서 처리됩니다.

그림 8: 관리되는 가상 PBX 서비스 예



사용 가능성

MPLS VPN에 대한 Cisco IOS NAT 지원은 Cisco IOS 릴리스 12.2(13)T에서 사용할 수 있으며 MPLS를 지원하고 이 초기 구축 릴리스 교육을 실행할 수 있는 모든 플랫폼에 사용할 수 있습니다.

결론

Cisco IOS NAT에는 현재 공유 서비스의 확장 가능한 구축을 허용하는 기능이 있습니다. Cisco는 고객에게 중요한 프로토콜에 대한 NAT ALG(Application Level Gateway) 지원을 지속적으로 개발합니다. 변환 기능을 위한 성능 향상 및 하드웨어 가속화를 통해 NAT 및 ALG는 향후 일정 기간 동안 적절한 솔루션을 제공할 수 있습니다. Cisco는 모든 관련 표준 활동 및 커뮤니티 활동을 모니터링하고 있습니다. 다른 표준이 개발되면 고객의 요구 사항, 요구 사항 및 애플리케이션에 따라 그 사용을 평가합니다.

관련 정보

- [Cisco IOS NAT 애플리케이션 레이어 게이트웨이](#)
- [MPLS 및 VPN 아키텍처](#)

- [고급 MPLS 설계 및 구현](#)
- [기술 지원 및 문서 - Cisco Systems](#)