Cisco IOS NAT — 与MPLS VPN集成

目录

简介

Cisco IOS®网络地址转换(NAT)软件允许从多个MPLS VPN访问共享服务，即使VPN中的设备使用重叠的IP地址。Cisco IOS NAT可感知VRF，可在MPLS网络内的提供商边缘路由器上配置。

注意：IOS中的MPLS仅支持传统NAT。目前，Cisco IOS不支持NAT NVI和MPLS。

MPLS VPN的部署预计在未来几年内会迅速增加。允许快速扩展和灵活连接选项的通用网络基础设施的优势无疑将推动可向互联网社区提供的服务进一步增长。

然而，增长障碍仍然存在。IPv6及其在可预见的将来超过连接需求的IP地址空间承诺仍处于部署的初期阶段。现有网络通常使用RFC 1918中定义的私有IP编址方案 。当地址空间重叠或存在重复时，网络地址转换通常用于互连网络。

运营商和企业如果希望提供网络应用服务或与客户和合作伙伴共享网络应用服务，则希望最大程度地减轻服务用户的连接负担。为了达到预期目标或实现回报，将产品扩展到所需的尽可能多的潜在用户是可取的，甚至是强制性的。使用中的IP编址方案不能成为排除潜在用户的障碍。

通过在通用MPLS VPN基础设施中部署Cisco IOS NAT，通信服务提供商可以减轻客户的一些连接负担，并加快他们将更多共享应用服务链接到这些服务的更多消费者的能力。

NAT的优势 — MPLS集成

与MPLS的NAT集成为服务提供商及其企业客户都带来了好处。它为服务提供商提供了更多选项来部署共享服务和访问这些服务。其他服务产品可以成为竞争对手的竞争优势。

对于服务提供商	对于VPN

更多服务产品	低成本
增加访问选项	更简单的访问
增加收入	寻址灵活性

寻求外包某些当前工作负载的企业客户也可以受益于服务提供商提供的更广泛的服务。将执行任何必要地址转换的负担转移到服务提供商网络可以减轻他们复杂的管理任务。客户可以继续使用私有编址，但仍然可以访问共享服务和互联网。在服务提供商网络中整合NAT功能还可能降低企业客户的总成本，因为客户边缘路由器不必执行NAT功能。

设计注意事项

当考虑将在MPLS网络内调用NAT的设计时，第一步是从应用角度确定服务需求。您需要考虑使用的协议以及应用强加的任何特殊客户端/服务器通信。确保Cisco IOS NAT支持并处理对所采用协议的必要支持。Cisco IOS NAT应用层网关文档中提供了支持的协议列表。

接下来，需要确定共享服务的预期使用情况和每秒数据包数的预计流量速率。NAT是路由器CPU密集型功能。因此，性能要求将是选择特定部署选项和确定涉及的NAT设备数量时的一个因素。

此外，请考虑应采取的任何安全问题和预防措施。虽然MPLS VPN按定义是私有且有效分离的流量，但共享服务网络在许多VPN中通常是通用的。

部署方案

在MPLS提供商边缘内部署NAT有两个选项：

- 通过出口NAT PE集中
- 与入口NAT PE分布

在离共享服务网络最近的MPLS网络出口点配置NAT功能的一些优势包括：

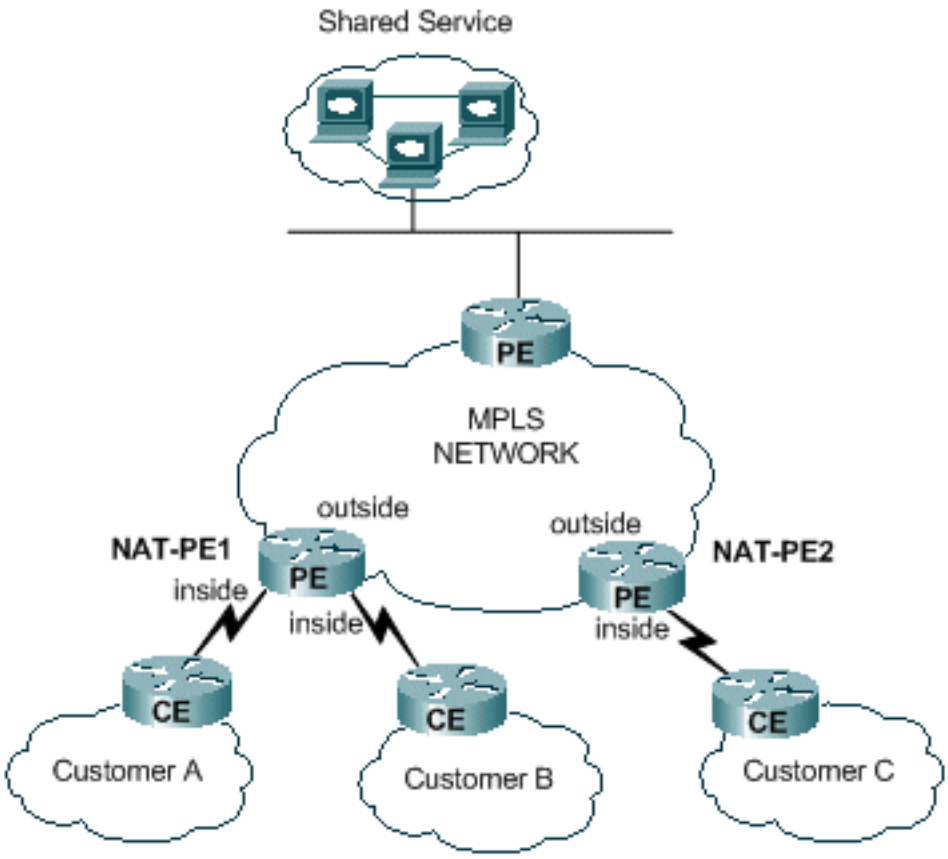
- 一种集中配置，可促进更简单的服务调配
- 简化故障排除
- 增强的运营可扩展性
- 降低IP地址分配要求

但是，可扩展性和性能的降低抵消了这些优势。这是必须考虑的主要权衡。当然，如果确定此功能与MPLS网络的集成不理想，也可以在客户网络内执行NAT功能。

入口PE NAT

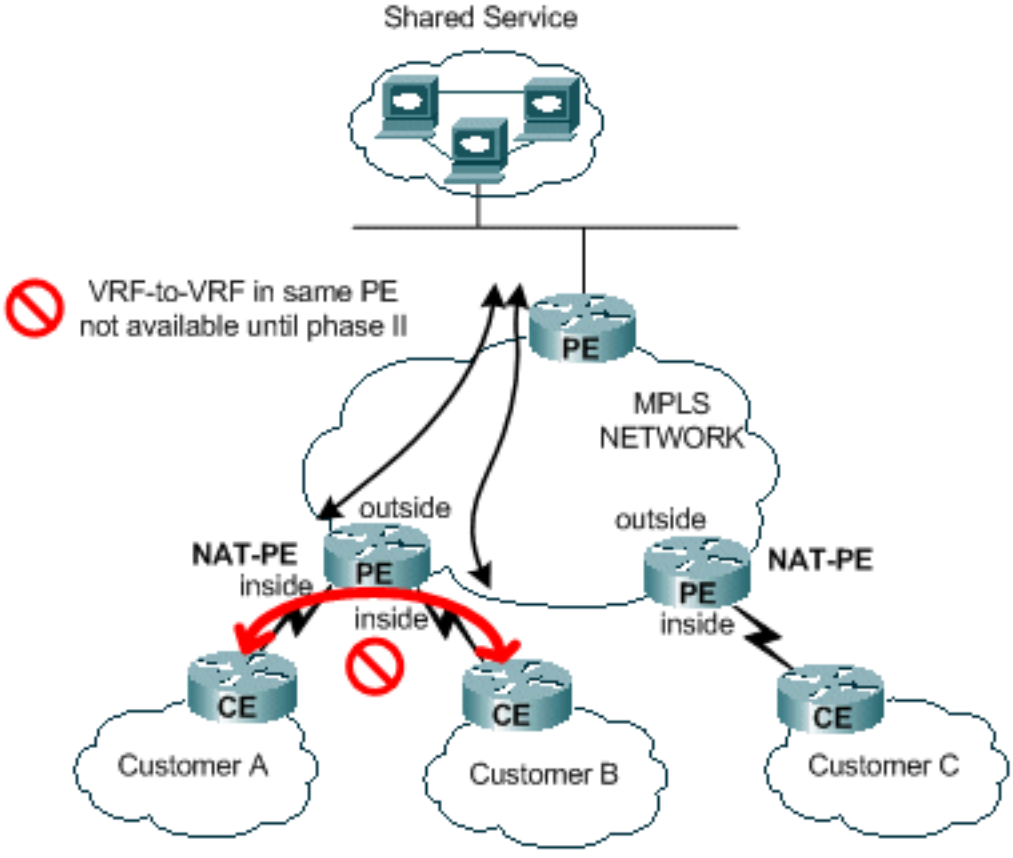
如图1所示，可以在MPLS网络入口PE路由器上配置NAT。通过此设计，可在很大程度上保持可扩展性，同时通过将NAT功能分布到许多边缘设备来优化性能。每个NAT PE处理本地连接到该PE的站点的流量。NAT规则和访问控制列表或路由映射控制需要转换的数据包。

图 1：入口PE NAT



如图2所示，存在一种限制，可防止两个VRF之间的NAT，同时为共享服务提供NAT。这是因为需要将接口指定为NAT“内部”和“外部”接口。在未来的Cisco IOS版本中，计划在单个PE中支持VRF之间的连接。

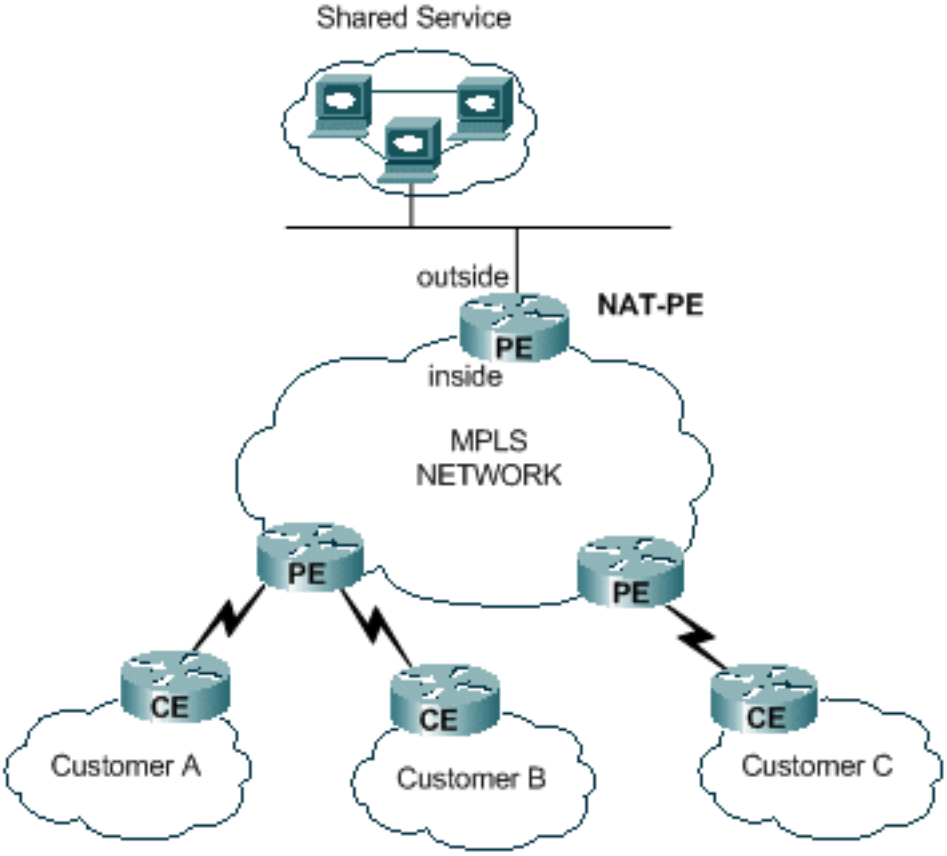
图 2：企业到企业



出口PE NAT

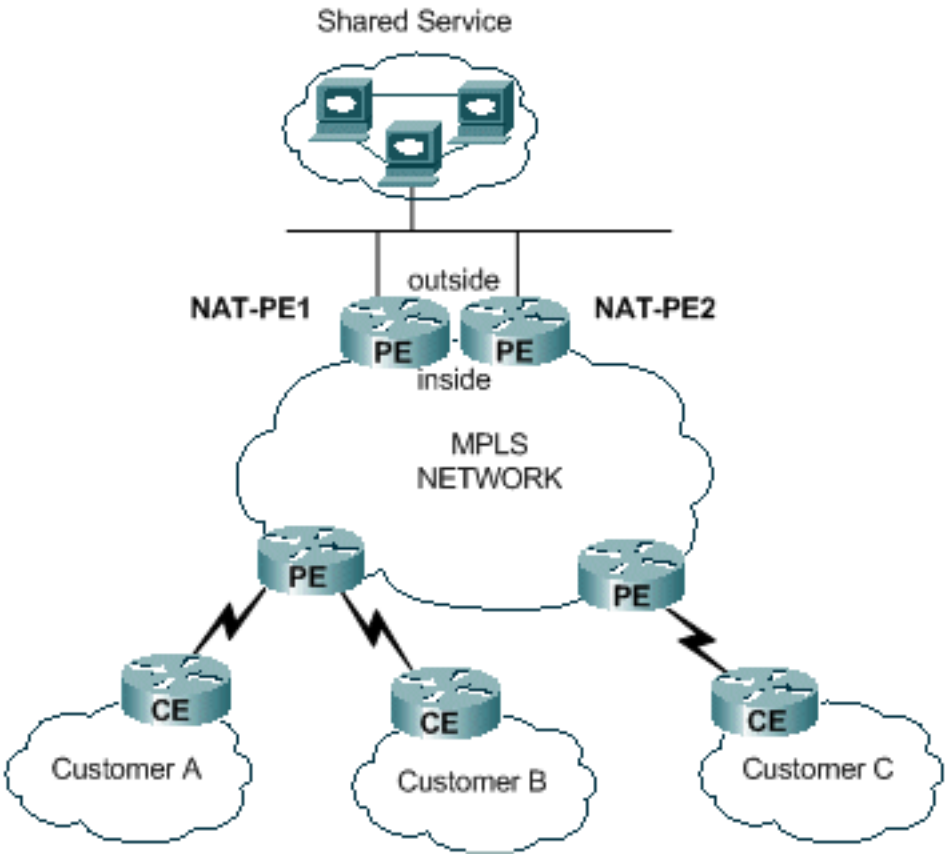
如图3所示，可以在MPLS网络出口PE路由器上配置NAT。通过此设计，可扩展性在一定程度上会降低，因为中央PE必须维护访问共享服务的所有客户网络的路由。还必须考虑应用性能要求，以使流量不会使必须转换数据包IP地址的路由器负担过重。由于NAT对使用此路径的所有客户集中进行，因此可以共享IP地址池；因此，所需子网的总数减少。

图 3：出口PE NAT



如图4所示，可以部署多台路由器来提高出口PE NAT设计的可扩展性。在此场景中，可以在特定NAT路由器上“调配”客户VPN。对于该组VPN的共享服务的汇聚流量和来自该服务的汇聚流量，将进行网络地址转换。例如，来自客户A和客户B的VPN的流量可以使用NAT-PE1，而来自客户C的VPN的流量使用NAT-PE2。每个NAT PE仅为定义的特定VPN传送流量，并且仅维护返回这些VPN中站点的路由。可以在每个NAT PE路由器中定义单独的NAT地址池，以便将数据包从共享服务网络路由到适当的NAT PE以进行转换并路由回客户VPN。

图 4：多出口PE NAT



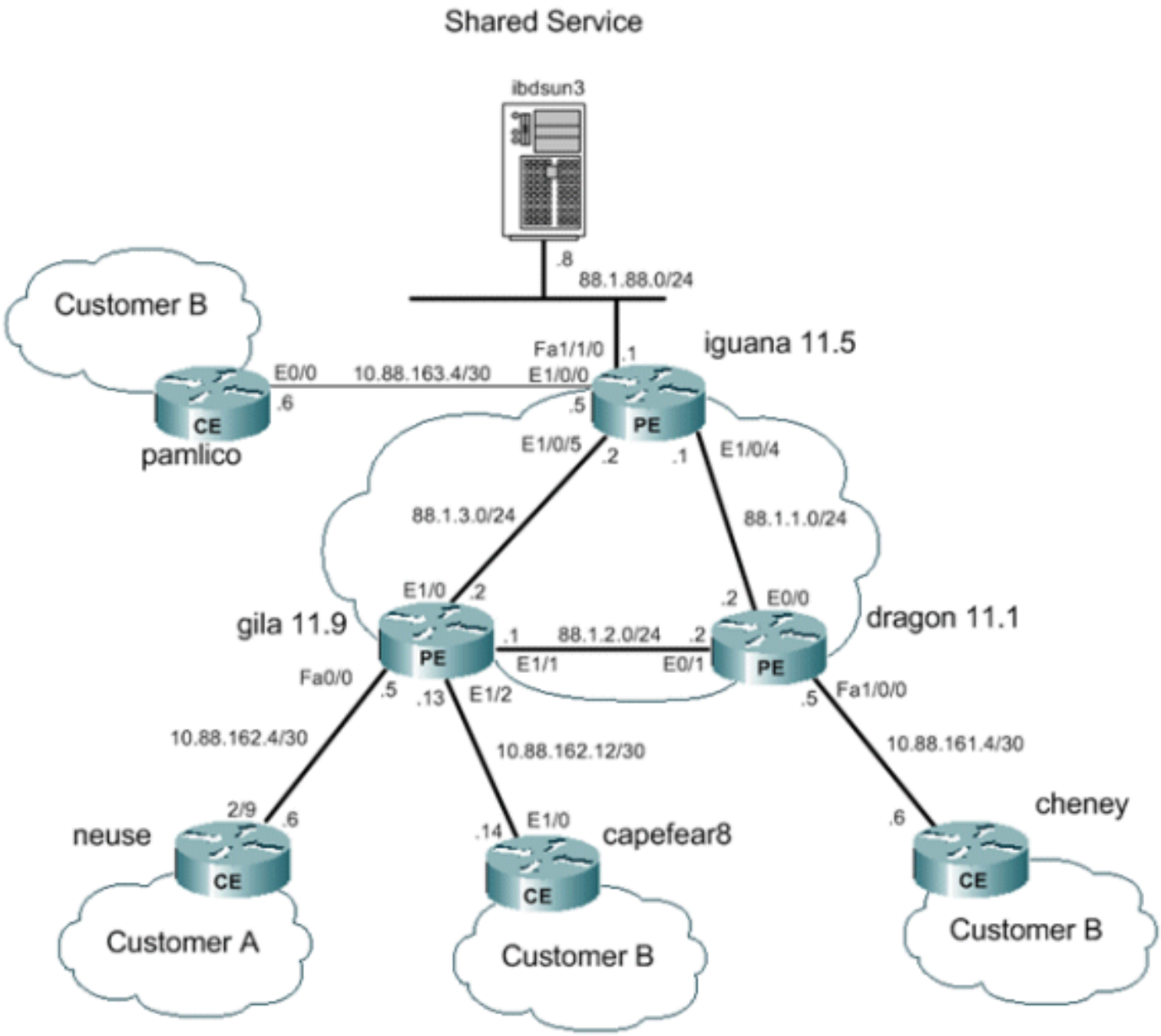
集中式设计确实限制了共享服务网络的配置方式。具体而言，在共享服务VPN和客户VPN之间不能使用MPLS VPN路由的导入/导出。这是由于RFC 2547规定的MPLS操作的性质 。当使用扩展社区和路由描述符导入和导出路由时，NAT无法确定来自进入中心NAT PE的数据包的源VPN。通常情况是使共享服务网络成为通用接口，而不是VRF接口。然后，在调配过程中，为与需要访问共享服务的客户VPN相关联的每个VRF表，在中央NAT PE中添加到共享服务网络的路由。稍后将更详细地介绍这一点。

部署选项和配置详细信息

本节包括与每个部署选项相关的一些详细信息。这些示例均取自图5所示的网络。请参阅本节其余部分的下图。

注意：在用于说明本白皮书VRF NAT操作的网络中，仅包含PE路由器。没有核心“P”路由器。但是，基本机制仍然可见。

图 5：VRF NAT配置示例



出口PE NAT

在本示例中，标有gila和dragon的提供商边缘路由器配置为简单PE路由器。共享服务LAN(iguana)附近的中心PE已配置为NAT。需要访问共享服务的每个客户VPN共享一个NAT池。NAT仅对发往88.1.88.8共享服务主机的数据包执行。

出口PE NAT数据转发

使用MPLS时，每个数据包在入口PE处进入网络，并在出口PE处退出MPLS网络。标签交换路由器从入口到出口经过的路径称为标签交换路径(LSP)。 LSP是单向的。不同的LSP用于返回流量。

当使用出口PE NAT时，会为来自共享服务用户的所有流量有效定义转发等价类(FEC)。换句话说，所有发往共享服务LAN的数据包都是通用FEC的成员。数据包仅在网络入口边缘分配一次到特定FEC，并跟随LSP到出口PE。FEC通过添加特定标签在数据包中指定。

从VPN到共享服务的数据包流

要使多个VPN中具有重叠地址方案的设备访问共享服务主机，需要NAT。当在出口PE上配置NAT时，网络地址转换表条目将包含VRF标识符，以区分重复地址并确保正确路由。

图 6：传输到出口PE NAT的数据包

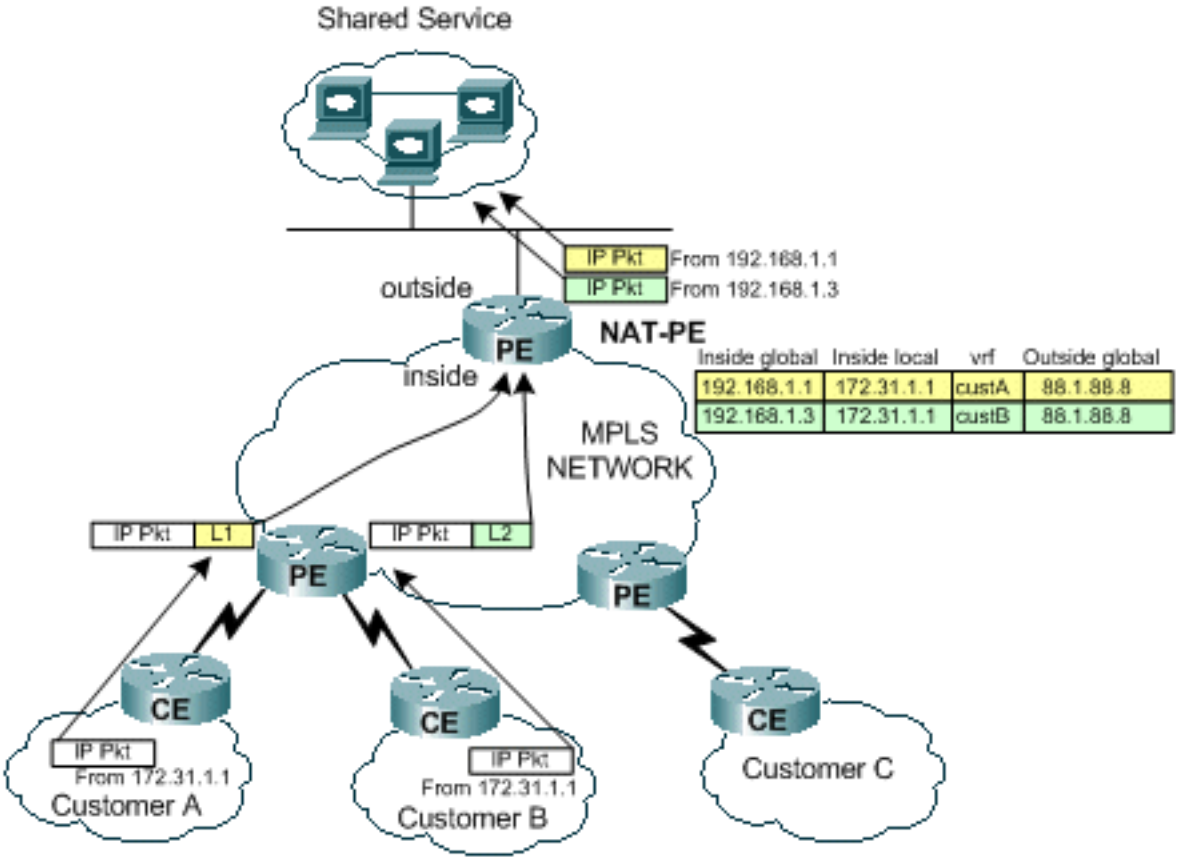


图6显示了从两个具有重复IP编址方案的客户VPN发往共享服务主机的数据包。图中显示源地址为172.31.1.1、发往地址为88.1.88.8的共享服务器的客户A的数据包。来自客户B的源IP地址相同的另一个数据包也发送到同一共享服务器。当数据包到达PE路由器时，会在转发信息库(FIB)中对目的IP网络执行第3层查找。

FIB条目告知PE路由器使用标签堆栈将流量转发到出口PE。堆栈中的底部标签由目标PE路由器分配，在本例中为路由器iguana。

```
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#
```

从显示中，我们可以看到来自VRF custA的数据包的标记值为24(0x18)，来自VRF custB的数据包的标记值为28(0x1C)。

在本例中，由于我们的网络中没有“P”路由器，因此没有附加标记。如果有核心路由器，则会施加外部标签，并且在核心网络内进行正常的标签交换过程，直到数据包到达出口PE。

由于gila 路由器直接连接到出口PE，因此我们看到标记在添加之前已弹出：

```
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#
```

下一个显示描述出口PE NAT路由器(在iguana上的接口E1/0/5处)接收的回声数据包。

```
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".]
```

这些ping操作会在出口PE路由器iguana的NAT表中创建以下**条目**。为上面显示的数据包创建的特定条目可以与其ICMP标识符进行匹配。

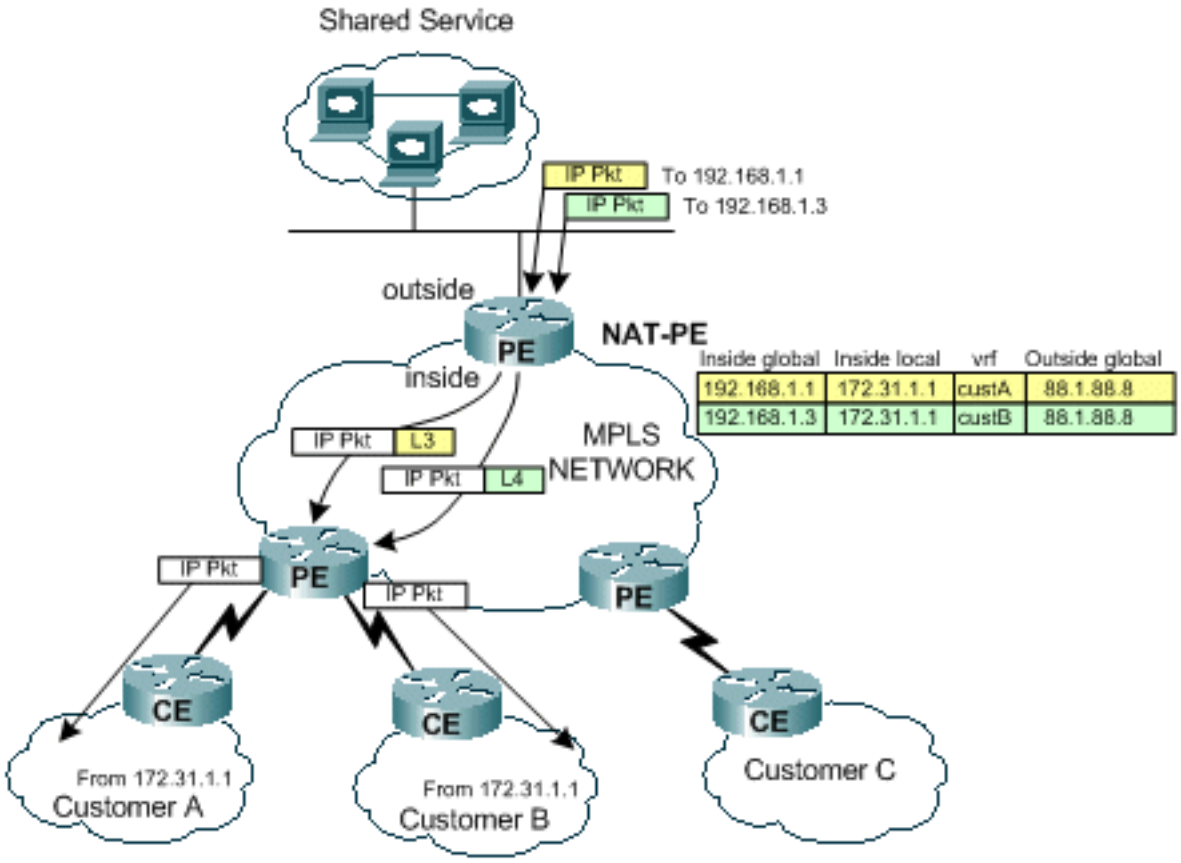
```
iguana#
show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 192.168.1.3:3365  172.31.1.1:3365    88.1.88.8:3365     88.1.88.8:3365
icmp 192.168.1.3:3366  172.31.1.1:3366    88.1.88.8:3366     88.1.88.8:3366
icmp 192.168.1.3:3367  172.31.1.1:3367    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#
```

从共享服务返回源VPN的数据包流

当数据包流回已访问共享服务主机的设备时，在路由之前会检查NAT表（数据包从NAT“外部”接口传输到“内部”接口）。 由于每个唯一条目都包括相应的VRF标识符，因此可以正确转换和路由数据包。

图 7：发回共享服务用户的数据包



如图7所示,NAT首先检查返回流量以查找匹配的转换条目。例如，数据包被发送到目的地192.168.1.1。搜索NAT表。找到匹配项后，对“内部本地”地址(172.31.1.1)执行适当的转换，然后使用NAT条目的关联VRF ID执行邻接查找。

```
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#
```

标签23(0x17)用于VRF custA中发往172.31.1.0/24的流量，标签26(0x1A)用于VRF custB中发往172.31.1.0/24的数据包。

在从路由器iguana发送的应答数据包中可以看**到这一点**:

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".]
```

当数据包到达目的PE路由器时，标签用于确定发送数据包的适当VRF和接口。

```
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#
```

配置

为简洁起见，已从配置中删除一些无关信息。

```
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
!
```

路由器Dragon的配置与Gila非常相似。

不允许导入/导出路由目标

当共享服务网络配置为VRF实例本身时，出口PE上的中心NAT不可能。这是因为传入数据包无法区分，并且只有一条返回始发子网的路由存在于出口PE NAT中。

注意：以下显示旨在说明无效配置的结果。

配置了示例网络，以便将共享服务网络定义为VRF实例（VRF名称=服务器）。 现在，入口PE上CEF表的显示显示如下：

```
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#
```

注：注意VRF custA和VRF custB*的*标记值24是如何强加的。

此显示显示共享服务VRF实例“sserver”的路由表：

```
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
```

注意：从出口PE路由器(iguana)的角度来看，目的网络仅存**在一**条路由。

因此，无法区分来自多个客户VPN的流量，并且返回的流量无法到达相应的VPN。**在必须将共享服务定义为VRF实例的情况下，必须将NAT功能移至入口PE。**

入口PE NAT

在本示例中，标记为gila和dragon的提供商边缘路由器配置为NAT。为需要访问共享服务的每个连接的客户VPN定义NAT池。每个NAT的客户网络地址都使用相应的池。NAT仅对发往88.1.88.8共享服务主机的数据包执行。

```
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
```

注意：在此方案中，不支持共享池。如果共享服务LAN（在出口PE处）通过通用接口连接，则NAT池可以共享。

从连接到neuse和capefear8的每个网络中的重复地址(172.31.1.1)发出的ping命令会导致以下NAT条目：

吉拉发来:

```
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
```

注意：根据源VRF，将相同的内部本地地址(172.31.1.1)转换为每个定义的池。在show ip nat translation verbose命令中可以看到VRF:

```
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
```

以下显示显示客户A和客户B的每个本地连接VPN的路由信息：

```
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
```

注意：已从静态配置中添加了每个NAT池的路由。这些子网随后会导入到出口PE路由器iguana的共享服务器VRF**中**:

```
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
```

配置

为简洁起见，已从配置中删除一些无关信息。

```
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 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 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
!
```

注意：面向客户网络的接口被指定为NAT“内部”接口，而MPLS接口则指定为NAT“外部”接口。

```
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 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 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
```

路由器Dragon的配置与Gila非常**相似**。

在入口PE NAT后到达中心PE的数据包

以下跟踪说明当目标共享服务网络配置为VRF实例时对唯一NAT池的要求。再次，请参阅图5中的图。当数据包进入路由器iguana的MPLS IP接口e1/0/5时，会捕获下面显示的**数据包**。

客户A VPN回应

此处，我们看到来自VRF custA中源IP地址172.31.1.1的回应请求。根据NAT配置的指定，源地址已转换为192.168.1.1:

```
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:
```

来自客户B VPN的回应

此处，我们看到来自VRF custB中源IP地址172.31.1.1的回应请求。根据NAT配置的指定，源地址已转换为192.168.2.1:

```
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".]
```

注意：在上面显示的两个数据包中，MPLS标签值为0019。

对客户A VPN的应答

接下来，我们会看到回复，返回VRF custA中的目的IP地址192.168.1.1。目的地址通过入口PE NAT功能转换为172.31.1.1。

```
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:
```

对客户B VPN的应答

此处，我们看到一条回应，返回到VRF custB中的目的IP地址192.168.1.1。目的地址通过入口PE NAT功能转换为172.31.1.1。

```
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".]
```

注意：在返回数据包中，MPLS标签值包括和不同：*001A*（用于VRF custA）*和001*D（用于VRF custB）。

客户A VPN — 目标是通用接口

下一组数据包显示了当到共享服务LAN的接口是通用接口而不是VRF实例的一部分时的区别。此处，配置已更改为对具有重叠IP地址的两个本地VPN使用公用池。

```
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".]
```

从客户B VPN回显 — 目标是通用接口

此处，我们看到来自VRF custB中源IP地址172.31.1.1的回应请求。根据NAT配置的指定，源地址被转换为192.168.1.3（从公共池SSPOOL1）：

```
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".]
```

注意：当出口PE上的接口是通用接口（而非VRF实例）时，强加的标签是不同的。在本例中为*0x*19*和0x1F*。

回应客户A VPN — 目标是通用接口

接下来，我们会看到回复，返回VRF custA中的目的IP地址192.168.1.1。目的地址通过入口PE NAT功能转换为172.31.1.1。

```
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".]
```

对客户B VPN的应答 — 目标是通用接口

此处，我们看到一条回应，返回到VRF custB中的目的IP地址192.168.1.3。目的地址通过入口PE NAT功能转换为172.31.1.1。

```
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".]
```

注意：由于回复的目的地是全局地址，因此不会强加VRF标签。

如果共享服务LAN网段的送出接口定义为通用接口，则允许使用通用池。ping操作会在路由器Gila中生成以下NAT**条目**:

```
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
```

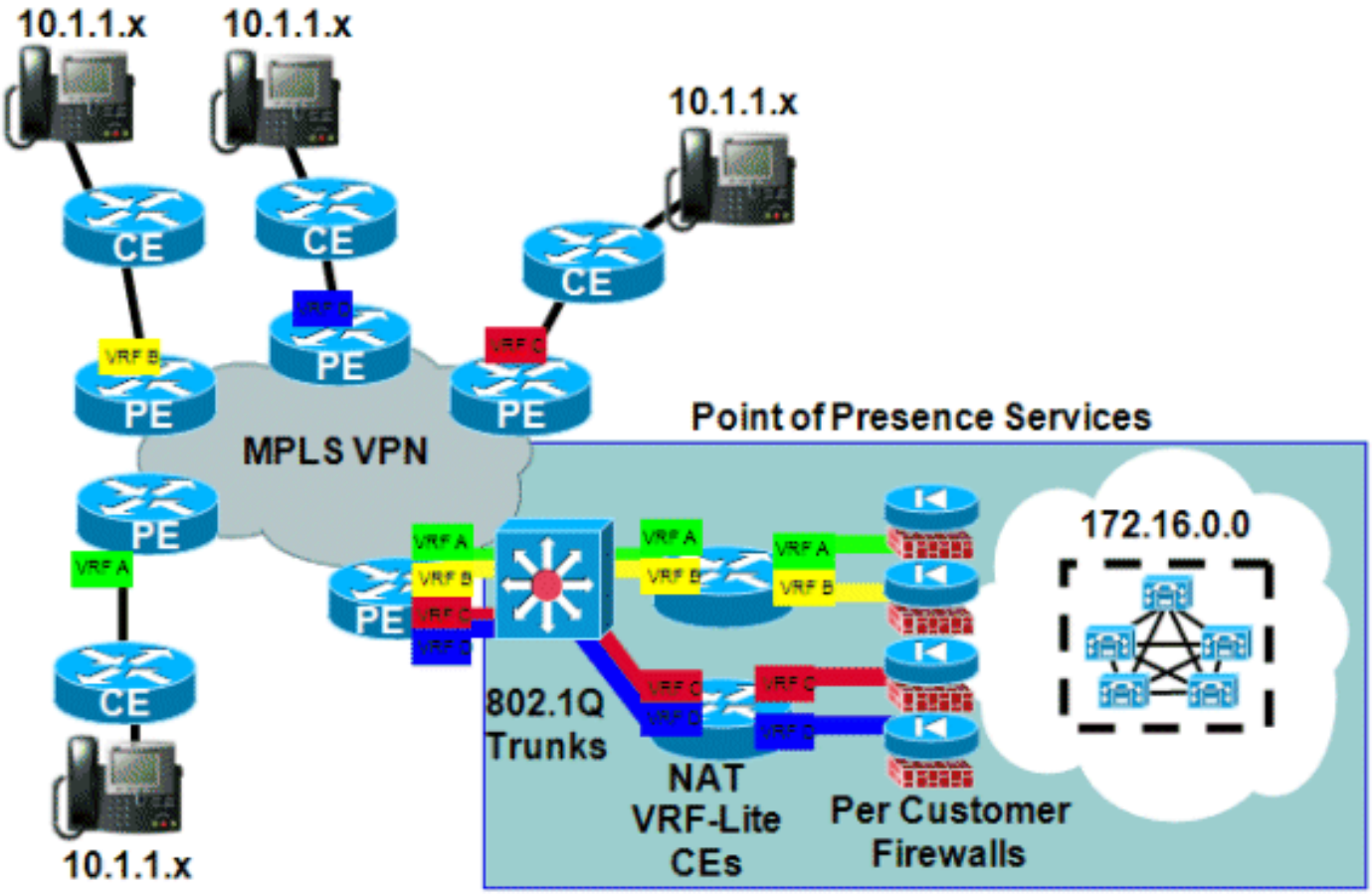
```
icmp 192.168.1.1:875   172.31.1.1:875     88.1.88.8:875      88.1.88.8:875
    create 00:00:18, use 00:00:18, left 00:00:41, Map-Id(In): 3,
    flags:
extended, use_count: 0, VRF : custA
icmp 192.168.1.1:876   172.31.1.1:876     88.1.88.8:876      88.1.88.8:876
    create 00:00:18, use 00:00:18, 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#
```

服务示例

共享虚拟IP PBX服务的示例如图8所示。这说明了前面介绍的入口和出口示例的变体。

在此设计中，共享VoIP服务由一组执行NAT功能的路由器前端。这些路由器使用称为VRF-Lite的功能具有多个VRF接口。然后，流量将流到共享的Cisco CallManager集群。防火墙服务也按公司提供。公司间呼叫必须通过防火墙，而公司内呼叫则通过客户VPN使用公司内部编址方案进行处理。

图 8：托管虚拟PBX服务示例



可用性

Cisco IOS版本12.2(13)T提供对MPLS VPN的思科IOS NAT支持，适用于支持MPLS且可以运行此早期部署版本系列的所有平台。

结论

Cisco IOS NAT具有允许当前可扩展部署共享服务的功能。思科继续为对客户重要的协议开发NAT应用级网关(ALG)支持。性能改进和转换功能的硬件加速将确保NAT和ALG在未来一段时间内提供可接受的解决方案。思科监控所有相关标准活动和社区行动。在开发其他标准时，将根据客户需求、需求和应用对其使用进行评估。

相关信息

- 思科IOS NAT应用层网关
- MPLS和VPN架构
- 高级MPLS设计和实施
- 技术支持和文档 - Cisco Systems