Cisco IOS NAT -与MPLS VPN的集成

目录

简介

Cisco IOS网络地址转换(NAT)软件允许对共享服务的访问从多个MPLS VPN，既使当在VPN的设备使用交迭的IP地址。Cisco IOS NAT知道，并且可以配置在MPLS网络内的供应商边缘路由器。

注意： 在IOS的MPLS仅支持与传统NAT。此时，没有在Cisco IOS的支持与MPLS的NAT的NVI。

MPLS VPN的部署设想迅速地增加以后几年。允许迅速扩展和灵活的连通性选项通用网络基础架构的好处在使用中可以提供到互联网络社区的将无疑地导致进一步增长。

然而，成长障碍仍然依然存在。IPv6和超出对可预见的将来的连接需要的其IP地址空间承诺仍然是在早期的相位部署。普通现有的网络使用私有IP编址方案如定义在RFC 1918内 。网络地址转换是常用的互联网络，当地址空间交迭时或复制存在。

有网络应用程序服务他们的服务提供商和企业要提供或共享与客户和合作伙伴将要最小化给服务的用户添负担所有连接。因为需要达到希望的目标或返回，是理想，均等必须，致以提供对许多个潜在的用户。IP编址方案在使用中不能是排除潜在的用户的障碍。

通过部署在普通的MPLS VPN基础设施内的Cisco IOS NAT，通信服务提供商能免除一些在客户的连接负担和加速他们的能力与那些服务的更多消费者连接更加共享的应用服务。

从NAT的好处– MPLS集成

与MPLS的NAT集成有两个服务提供商和他们的企业用户的好处。它提供服务提供商更多选项部署共享服务和提供存取对于那些服务。其它服务提供可以是在竞争对手的一区分标志。

服务提供商	VPN

更多提供的业务	低成本
增加的访问选项	更加简单的访问
增加收入	寻址灵活性

寻求的企业用户外包他们的一些当前工作量能也受益于更宽的提供由服务提供商。转移负担执行所有必要的地址转换对服务提供商网络免除他们一项复杂管理任务。客户可能继续使用私有寻址，维护对共享服务和互联网的访问。因为用户边缘路由器不必须执行NAT功能，统一在服务提供商网络内的NAT功能可能也降低总成本对企业用户。

设计注意事项

当考虑将调用在MPLS网络内时的NAT的设计，第一步将确定服务需要从应用程序观点。您将需要思考协议应用程序强加的使用的和所有特殊客户端/服务器通信。确保Cisco IOS NAT支持被使用的协议的必要的支持并且处理。支持的协议列表在本文Cisco IOS NAT应用层网关提供。

其次，确定共享服务和期望的流量速率的预计使用情况在每秒数据包数将是必要的。NAT是路由器CPU密集型功能。所以，性能要求将是在选择一个特定的部署选项的一个要素和确定NAT设备数量包括的。

并且，请考虑应该采取的所有安全问题和注意事项。虽然MPLS VPN，根据定义，私有，并且有效分开的流量，共享服务网络在许多VPN中通常是普通。

部署方案

有NAT部署的两个选项在MPLS运营商边缘内：

- 集中与出口NAT观点扫描器
- 分配与入口NAT观点扫描器

对配置NAT功能的一些优点在近MPLS网络的出口点对共享服务网络包括：

- 促进更加简单的服务规定的集中配置
- 被简化的故障排除
- 被提高的可操作的可扩展性
- 减小的IP地址分配需求

然而，优点由对可扩展性和性能的减少抵消。这是必须考虑的主要折衷。当然， NAT功能可能在客户网络内也执行，如果确定此功能的集成与MPLS网络的不是理想。

入口PE NAT

如图1所显示， NAT可以配置在MPLS网络入口PE路由器。使用此设计，可扩展性到大规模范围维护，当性能通过分配在许多边缘设备的NAT功能优化时。每个NAT PE处理站点的流量本地连接对该PE。NAT规则和访问控制列表或者数据包要求转换的路由映射控制。

图 1：入口PE NAT

有防止在两VRF之间的NAT，虽然同样提供NAT给共享服务如图2.所显示的限制。这归结于需求指派接口作为NAT “里面”和“外部”接口。连接的支持在单个PE的VRF之间对将来Cisco IOS版本计划。

图 2：企业间

出口PE NAT

如图3.所显示， NAT可以配置在MPLS网络插PE路由器。使用此设计，可扩展性减少到某度，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。必须也考虑应用程序性能需求，以便流量不装载过多必须翻译数据包的IP地址的路由器。由于NAT为使用此路径的所有客户在中央发生， IP地址池可以共享;因此，减少要求的子网总数。

图 3：出口PE NAT

如图4.所显示，多个路由器可能配置增加出口PE NAT设计的可扩展性。在此方案中，客户VPN能是“已配置”在特定NAT路由器。网络地址转换为总流量将发生到/从该套的共享服务VPN。例如，而到/从VPN的流量客户C的使用NAT-PE2，从VPN的流量客户的A和B可能使用NAT-PE1。每个NAT PE将运载仅流量定义的特定VPN的和只维护路由回到那些VPN的站点。独立的NAT地址池可能在其中每一个NAT PE路由器内定义，以便数据包从共享服务网络发送到转换和路由的适当的NAT PE回到客户VPN。

图 4：多个出口PE NAT

集中化设计强加一限制关于怎样必须配置共享服务网络。特别地，使用MPLS VPN路由导入/出口在共享服务VPN之间的和客户VPN不是可能的。这归结于MPLS操作的本质如指定由RFC 2547 。使用扩展团体和路由描述符时，当路由导入并且导出， NAT不能确定从进入中央NAT PE的数据包的来源VPN。通常案件是做共享服务网络通用接口而不是VRF接口。对共享服务网络的一个路由在每个VRF表的中央NAT PE然后被添加关联与需要对共享服务的客户VPN访问作为提供的流程一部分。这较详细地描述的以后。

部署选项和配置细节

此部分包括与其中每一个涉及的一些详细信息部署选项。示例全部从下面列出的网络在表5.采取参考其余的此图表此部分。

注意： 在用于的网络中说明VRF NAT的操作本文的，只有PE路由器包括。没有核心“P”路由器。然而，重要机制能仍然被看到。

图 5：VRF NAT配置示例

出口PE NAT

在本例中，供应商边缘路由器被标记的**吉拉**和**龙**配置作为简单PE路由器。在共享服务LAN (**鬣蜥**)附近的中央PE为NAT配置。单个NAT池由需要对共享服务的访问的每客户VPN共享。NAT在为在88.1.88.8的共享服务主机注定的数据包仅执行。

出口PE NAT数据转发

使用MPLS，每数据包进入网络在入口PE并且退出MPLS网络在出口PE。从入口横断的标签交换路由器路径到出口叫作标签交换路径(LSP)。LSP是单向的。一不同的LSP使用回程数据流。

当曾经出口PE NAT时，转发等效类(FEC)为从共享服务的用户的所有流量有效定义。换句话说，为共享服务LAN注定的所有信息包是普通的FEC的成员。数据包一次分配到特定FEC在网络的入口边缘并且跟随LSP对出口PE。FEC在数据包被选定通过添加一个特定的标签。

对共享服务的数据包流从VPN

为了在有访问重迭的地址的机制共享服务主机的多个VPN的设备， NAT要求。当NAT配置在出口PE，网络地址转换条目将包括VRF标识符区分重复地址和保证适当的路由。

图 6：数据包传送对出口PE NAT

图6说明为从有相同的IP编址方案的两个客户VPN的一台共享服务主机注定的数据包。图显示产生在有为一个共享服务器注定了的172.31.1.1源地址的客户A的数据包在88.1.88.8。从客户B的另一数据包有同样源IP地址的也发送对同样共享的服务器。当数据包到达PE路由器时，第3层查找为在转发信息库(FIB)的目的地IP网络完成。

使用标签栈， FIB条目告诉PE路由器转发流量到出口PE。在堆叠的底部标签由目的地PE路由器分配，在这种情况下路由器**鬣鳞蜥**。

```
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#
```

我们能从显示看到从VRF custA的数据包将有标记值为24 (0x18)，并且从VRF custB的数据包将有标记值为28 (0x1C)。

在这种情况下，因为没有我们的网络的“P”路由器，那里是没有强加的另外的标记。有核心路由器，将强加外标签，并且标签交换正常进程在核心网络内将发生，直到数据包到达了出口PE。

因为**吉拉**路由器直接地连接对出口PE，我们看到标记弹出，在被添加前：

```
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#
```

下显示表示echo数据包如接收由出口PE NAT路由器(在**鬣鳞蜥的**接口E1/0/5)。

```
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导致在出口PE路由器**鳖鳞蜥的**NAT表里创建的以下条目。为数据包创建的特定条目显示如上可以由他们的ICMP标识符匹配。

```
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#
```

从共享服务的数据包流回到始发地VPN

当信息包流回到访问共享服务主机的设备，NAT表在路由(去从NAT “外部”接口的数据包之前被检查到“里面”接口)。由于每个唯一条目包括对应的VRF标识符，数据包可以翻译和路由适当地。

图 7：数据包传送回到共享服务用户

如Figure7所显示，回程数据流由NAT首先检查查找一匹配的转换条目。例如，数据包发送对目的地192.168.1.1。NAT表被搜索。当找到时匹配，适当的转换完成对“Inside local”地址(172.31.1.1)使用从NAT条目的相关的VRF ID邻接查找然后执行。

```
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)使用为172.31.1.0/24注定的流量在VRF custA和标签26 (0x1A)使用为172.31.1.0/24注定的数据包在VRF custB。

这在从路由器**鬣蜥蜥**发送的ECHO回复数据包被看到：

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".]

当数据包到达目的地PE路由器时，标签用于确定适当的VRF和接口发送数据包。

```
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

一些额外的信息从配置为简要起见删除。

```
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**龙将**有一配置非常类似于**吉拉**。

路由目标导入/导出没允许的

当共享服务网络配置作为VRF实例时，在出口PE的中央NAT不是可能的。这是因为流入数据包不可以是著名的，并且仅回到产生的子网的一个路由是存在出口PE NAT。

注意： 跟随的显示被认为说明无效的配置的结果。

示例网络配置，以便共享服务网络定义作为VRF实例(VRF名字= sserver)。现在， CEF表的显示在入口PE的显示此：

```
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#
```

注意： 公告标记值*24*如何为VRF custA和VRF custB强加。

此显示显示共享服务VRF实例的“sserver”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： 仅一个路由为目的地网络是存在从出口PE路由器的(**鬣蜥蜥**)方面。

所以，从多个用户VPN的流量不可能是著名的，并且回程数据流不可能到达适当的VPN。**在共享服务必须定义作为VRF实例的案件中，必须移动NAT功能向入口PE。**

入口PE NAT

在本例中，供应商边缘路由器被标记的**吉拉**和**龙**为NAT配置。NAT池为需要对共享服务的访问的每附加的客户VPN定义。适当的池使用NAT的其中每一个客户网络网络地址。NAT在为在88.1.88.8的共享服务主机注定的数据包仅执行。

```
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)通过通用接口连接，则NAT池可能共享。

从一重复地址来源的ping (172.31.1.1)在其中每一网络内附加对**neuse，**并且**capefear8**导致这些NAT条目：

从**吉拉**：

```
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： 同一个内部本地地址(172.31.1.1)根据来源VRF翻译对其中每一个定义的池。VRF在**verbose命令的show ip nat translation**能被看到：

```
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
```

这些显示显示其中每一个的路由信息客户A和客户的B本地附加的VPN ：

```
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： 其中每一个的一个路由NAT池从静态配置被添加了。这些子网随后导入到在出口PE路由器**鬣蜥蜥的**共享服务器VRF ：

```
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

一些额外的信息从配置为简要起见删除。

```
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： 面对客户网络的接口被选定作为NAT “里面”接口和MPLS接口被选定作为NAT “外部”建立接口。

```
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**龙将**有一配置非常类似于**吉拉**。

在中央印制厂PE的到达数据包在入口PE NAT以后

当目的地共享服务网络配置作为VRF实例时，下面的跟踪说明唯一NAT池的需求。再次，在表5.参考图表。当他们进入MPLS IP接口e1/0/5在路由器**鳜鳞蜥**，下面显示的数据包捕获。

从客户A VPN的响应

这里，我们看到ECHO请求来自在VRF custA的源IP地址172.31.1.1。源地址翻译对192.168.1.1如指定由NAT配置：

```
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:
```

从客户B VPN的响应

这里，我们看到ECHO请求来自在VRF custB的源IP地址172.31.1.1。源地址翻译对192.168.2.1如指定由NAT配置：

```
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： MPLS标签值是*0019*在显示的两个数据包如上。

对客户A VPN的Echo replies

其次，我们看到echo replies去的上一步对在VRF custA的目的IP地址192.168.1.1。目的地址由入口PE NAT功能翻译对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Echo replies

这里，我们看到echo replies去的上一步对在VRF custB的目的IP地址192.168.1.1。目的地址由入口PE NAT功能翻译对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： 在返回信息包中， MPLS标签值包括并且有所不同：VRF custA的VRF custB的*001A*和*001D*。

从客户的响应VPN目标是通用接口

当对共享服务LAN的接口是VRF实例的而不是通用接口部分时，数据包此下一组显示出差异。这里，配置更改使用公用池两个本地VPN与重叠IP地址。

```
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目标的响应是通用接口

这里，我们看到ECHO请求来自在VRF custB的源IP地址172.31.1.1。源地址翻译对192.168.1.3 (从公用池SSPOOL1)如指定由NAT配置：

```
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： 当在出口PE的接口是通用接口(不是VRF实例)，被强加的标签不同的。在这种情况下，*0x19*和*0x1F*。

对客户的Echo replies VPN目标是通用接口

其次，我们看到echo replies去的上一步对在VRF custA的目的IP地址192.168.1.1。目的地址由入口PE NAT功能翻译对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目标的Echo replies是通用接口

这里，我们看到echo replies去的上一步对在VRF custB的目的IP地址192.168.1.3。目的地址由入口PE NAT功能翻译对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： 因为回复被注定对全局地址，没有强加VRF标签。

使用对作为通用接口定义的共享服务LAN分段的退出接口，公用池允许。ping导致在路由器**吉拉的**这些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

一共享虚拟IP PBX服务的示例在表8.显示。 这说明一变量对描述的入口和出口示例前。

在此设计，共享VoIP服务由执行NAT功能的一组路由器前端。这些路由器有多个VRF接口使用叫作VRF-Lite的功能。流量然后流到共享Cisco CallManager集群。防火墙服务在每公司基本类型也提供。使用公司的内部编址方案，而公司内部的呼叫在客户VPN间被处理公司间呼叫必须穿过防火墙。

图 8：托管型虚拟PBX服务示例

可用性

MPLS VPN的Cisco IOS NAT支持是可用的在Cisco IOS版本12.2(13)T并且为支持MPLS，并且能运行此早期部署版本系列的所有平台是可用的。

结论

Cisco IOS NAT有允许的功能共享服务今天的可扩展部署。思科继续开发NAT应用级协议的网关(ALG)支持重要对客户。性能改进和硬件加速转换功能的保证NAT和ALGs在将来提供可接受解决方案。所有相关标准活动和社区活动由思科监控。因为其他标准开发，他们的使用将被评估根据客户欲望、要求和应用程序。

相关信息

- Cisco IOS NAT应用层网关
- MPLS和VPN体系结构
- 先进的MPLS设计和实施
- 技术支持和文档 - Cisco Systems