NAT do Cisco IOS - Integração com VPN MPLS

Contents

Introduction

O software Cisco IOS® Network Address Translation (NAT) permite o acesso a serviços compartilhados de várias VPNs MPLS, mesmo quando os dispositivos nas VPNs usam endereços IP que se sobrepõem. O Cisco IOS NAT é preparado para VRF e pode ser configurado em roteadores na extremidade do provedor dentro da rede MPLS.

Observação: o MPLS no IOS é suportado somente com NAT legado. No momento, não há suporte no Cisco IOS para NAT NVI com MPLS.

A implantação de VPNs MPLS deverá aumentar rapidamente nos próximos anos. Os benefícios de uma infraestrutura de rede comum que permita opções de expansão rápida e conectividade flexível certamente impulsionarão um maior crescimento dos serviços que podem ser oferecidos à comunidade de redes interconectadas.

No entanto, continuam a existir obstáculos ao crescimento. O IPv6 e sua promessa de um espaço de endereço IP que excede as necessidades de conectividade no futuro previsível ainda estão nas fases iniciais da implantação. As redes existentes geralmente usam esquemas de endereçamento IP privado como definido na RFC 1918 . A conversão de endereços de rede é frequentemente usada para interconectar redes quando sobreposição ou duplicação de espaços de endereços.

Os provedores de serviços e as empresas que têm serviços de aplicativos de rede que desejam oferecer ou compartilhar com clientes e parceiros desejarão minimizar qualquer carga de conectividade imposta ao usuário do serviço. É desejável, até mesmo obrigatório, estender a oferta para tantos usuários em potencial quanto necessário para atingir os objetivos ou retorno desejados. O esquema de endereçamento IP em uso não deve ser uma barreira que exclua usuários potenciais.

Ao implantar a NAT do Cisco IOS na infraestrutura comum de VPN MPLS, os provedores de serviços de comunicação podem aliviar parte da carga de conectividade para os clientes e acelerar sua capacidade de vincular mais serviços de aplicativos compartilhados a mais consumidores desses serviços.

Benefícios da integração NAT - MPLS

A integração do NAT com o MPLS tem benefícios tanto para provedores de serviços quanto para seus clientes corporativos. Ele oferece aos provedores de serviços mais opções para implantar serviços compartilhados e fornecer acesso a esses serviços. Ofertas de serviços adicionais podem ser um diferencial em relação aos concorrentes.

Para provedor de serviços	Para VPN
Mais ofertas de serviços	Custos reduzidos
Mais opções de acesso	Acesso mais simples
Aumento da receita	Abordar a flexibilidade

Os clientes corporativos que buscam terceirizar parte de sua carga de trabalho atual também podem se beneficiar de ofertas mais amplas por parte dos provedores de serviços. Transferir a carga de executar qualquer tradução de endereço necessária para a rede do provedor de serviços libera-os de uma tarefa administrativa complicada. Os clientes podem continuar a usar o endereçamento privado, embora mantenham o acesso a serviços compartilhados e à Internet. A consolidação da função NAT na rede do provedor de serviços também pode reduzir o custo total para clientes corporativos, já que os roteadores de borda do cliente não precisam executar a função NAT.

Considerações do projeto

Ao considerar projetos que chamarão NAT dentro da rede MPLS, a primeira etapa é determinar as necessidades de serviço do ponto de vista do aplicativo. Você precisará considerar os protocolos usados e qualquer comunicação especial cliente/servidor imposta pelo aplicativo. Certifique-se de que o suporte necessário para os protocolos empregados seja suportado e tratado pelo Cisco IOS NAT. Uma lista de protocolos suportados é fornecida no documento Gateways de Camada de Aplicação NAT do Cisco IOS.

Em seguida, será necessário determinar o uso esperado do serviço compartilhado e a taxa de tráfego prevista em pacotes por segundo. O NAT é uma função que exige muito da CPU do roteador. Portanto, os requisitos de desempenho serão um fator na seleção de uma opção de implantação específica e na determinação do número de dispositivos NAT envolvidos.

Além disso, considere todos os problemas de segurança e precauções que devem ser tomadas. Embora as VPNs MPLS, por definição, sejam tráfego privado e efetivamente separadas, a rede de serviço compartilhado é geralmente comum entre muitas VPNs.

Cenários de implantação

Há duas opções para implantação de NAT na borda do provedor de MPLS:

- Centralizado com PEs NAT de saída

- Distribuído com PEs NAT de entrada

Algumas vantagens de configurar a função NAT no ponto de saída da rede MPLS mais próxima à rede de serviço compartilhado incluem:

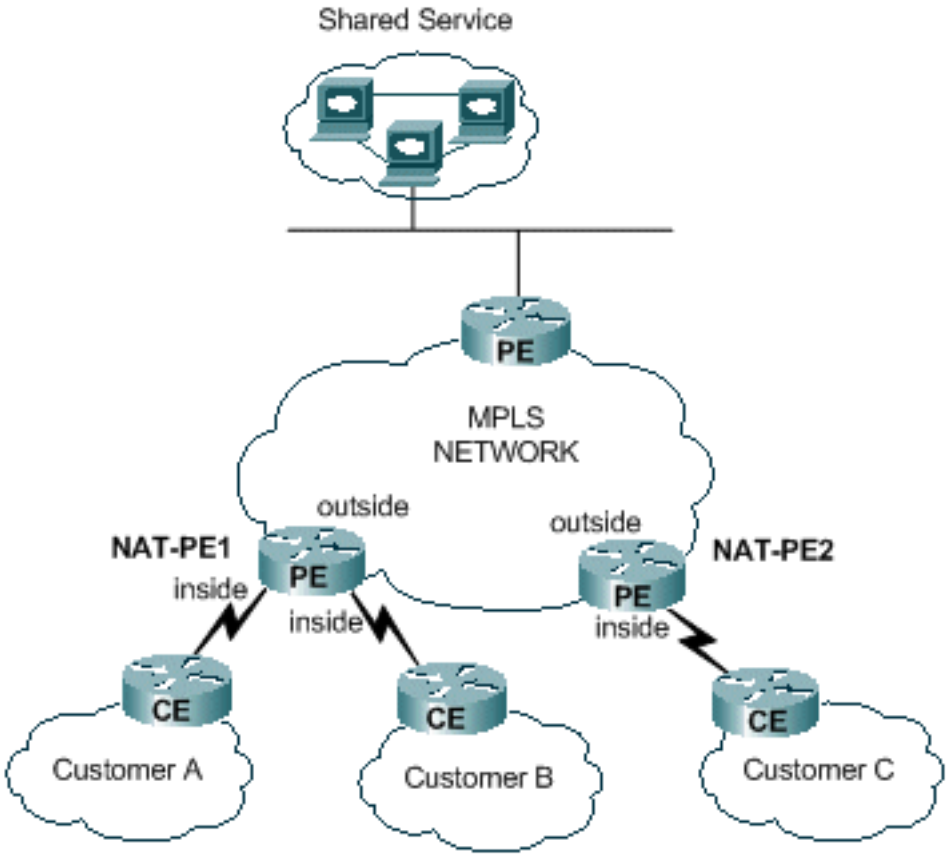
- Uma configuração centralizada que promove provisionamento de serviços mais simples
- Solução de problemas simplificada
- Maior escalabilidade operacional
- Redução dos requisitos de alocação de endereços IP

No entanto, as vantagens são compensadas por uma redução na escalabilidade e no desempenho. Esta é a principal transação que deve ser considerada. É claro que a função NAT também pode ser executada nas redes do cliente se for determinado que a integração desse recurso com uma rede MPLS não é desejável.

NAT PE de entrada

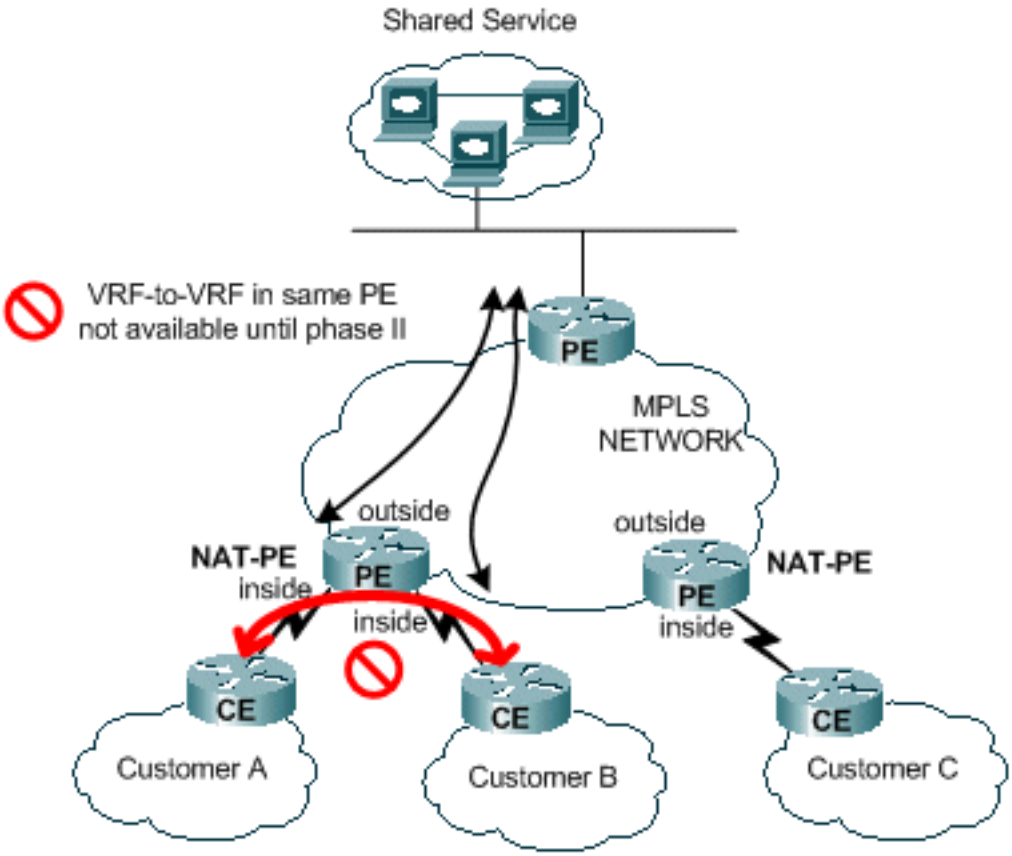
O NAT pode ser configurado no roteador PE de entrada de rede MPLS, como mostrado na Figura 1. Com esse projeto, a escalabilidade é mantida em grande parte, enquanto o desempenho é otimizado pela distribuição da função NAT em vários dispositivos de borda. Cada NAT PE lida com o tráfego de sites conectados localmente a esse PE. As regras de NAT e as listas de controle de acesso ou os mapas de rota controlam quais pacotes exigem tradução.

Figura 1: NAT PE de entrada



Há uma restrição que impede o NAT entre dois VRFs e, ao mesmo tempo, fornece o NAT a um serviço compartilhado, como mostrado na Figura 2. Isso se deve ao requisito de designar interfaces como NAT interfaces "internas" e "externas". O suporte para conexões entre VRFs em um único PE está planejado para uma futura versão do Cisco IOS.

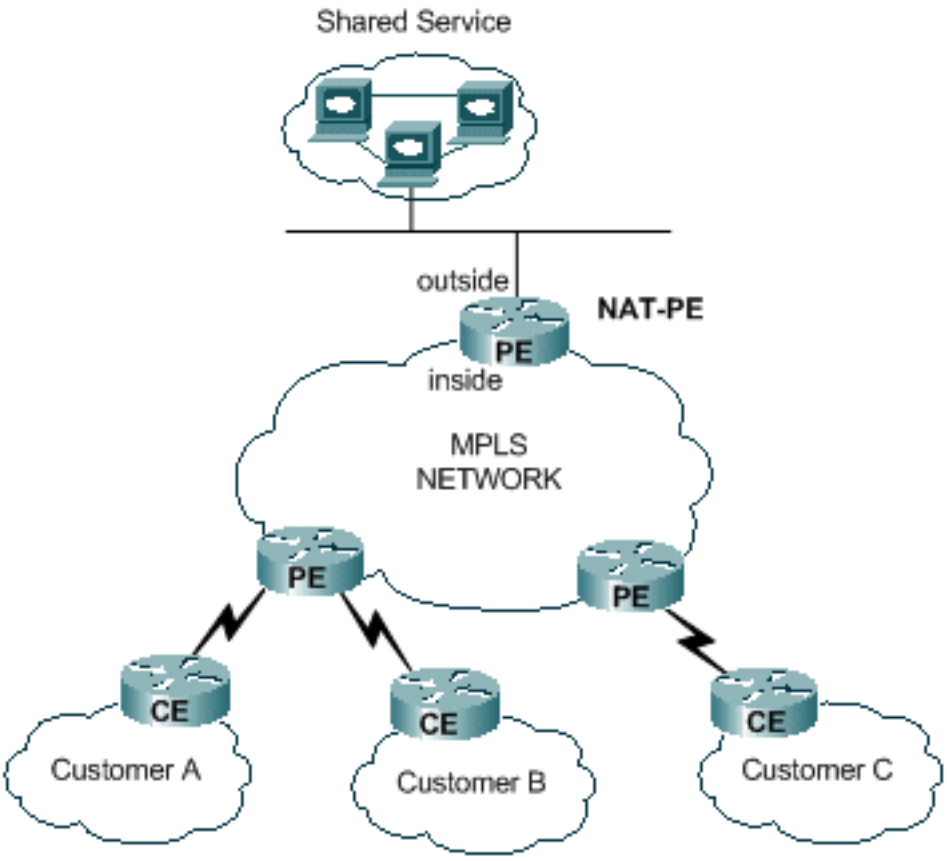
Figura 2: De empresa para empresa



NAT PE de saída

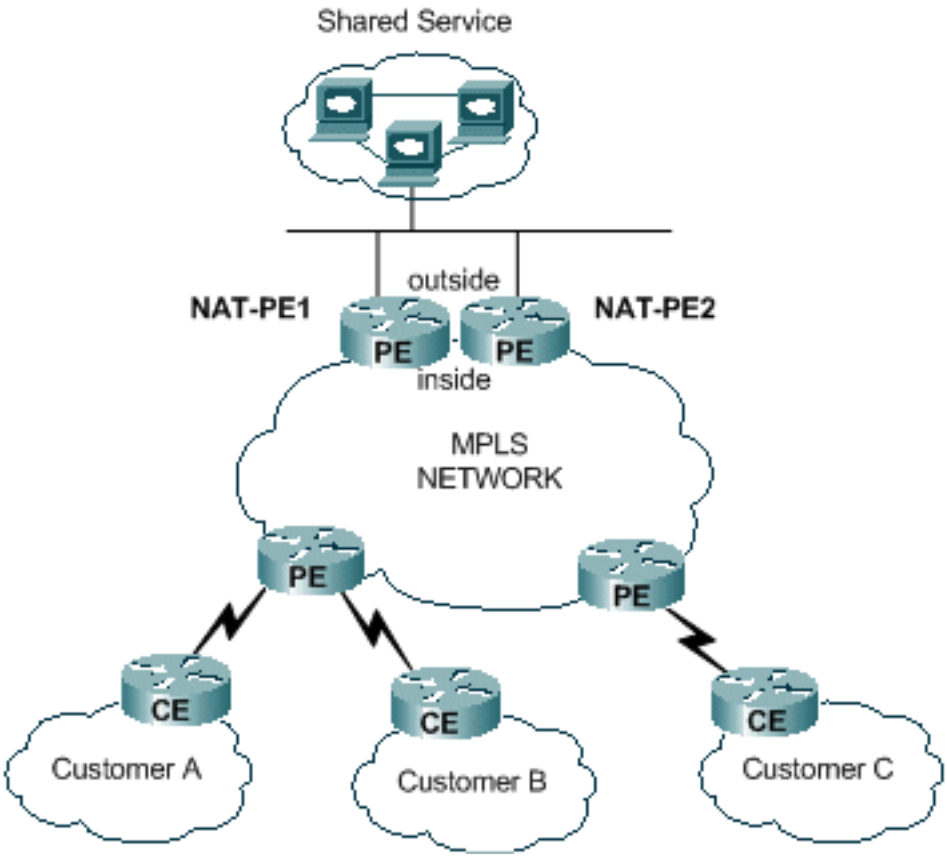
O NAT pode ser configurado no roteador PE de saída de rede MPLS, como mostrado na Figura 3. Com esse projeto, a escalabilidade é reduzida até certo ponto, já que o PE central deve manter rotas para todas as redes de clientes que acessam o serviço compartilhado. Os requisitos de desempenho do aplicativo também devem ser considerados para que o tráfego não sobrecarregue o roteador que deve converter os endereços IP dos pacotes. Como o NAT ocorre centralmente para todos os clientes que usam esse caminho, os pools de endereços IP podem ser compartilhados; assim, o número total de sub-redes necessárias é reduzido.

Figura 3: NAT PE de saída



Vários roteadores podem ser implantados para aumentar a escalabilidade do projeto de NAT PE de saída, como mostrado na Figura 4. Nesse cenário, as VPNs do cliente podem ser "provisionadas" em um roteador NAT específico. A conversão de endereço de rede ocorreria para o tráfego agregado de e para o serviço compartilhado desse conjunto de VPNs. Por exemplo, o tráfego das VPNs para o Cliente A e B pode usar NAT-PE1, enquanto o tráfego de e para a VPN para o cliente C usa NAT-PE2. Cada NAT PE transportaria tráfego somente para as VPNs específicas definidas e apenas manteria rotas de volta para os sites nessas VPNs. Podiam ser definidos pools de endereços NAT separados em cada um dos roteadores NAT PE para que os pacotes sejam roteados da rede de serviço compartilhado para o NAT PE apropriado para tradução e roteamento de volta para a VPN do cliente.

Figura 4: NAT PE de saída múltipla



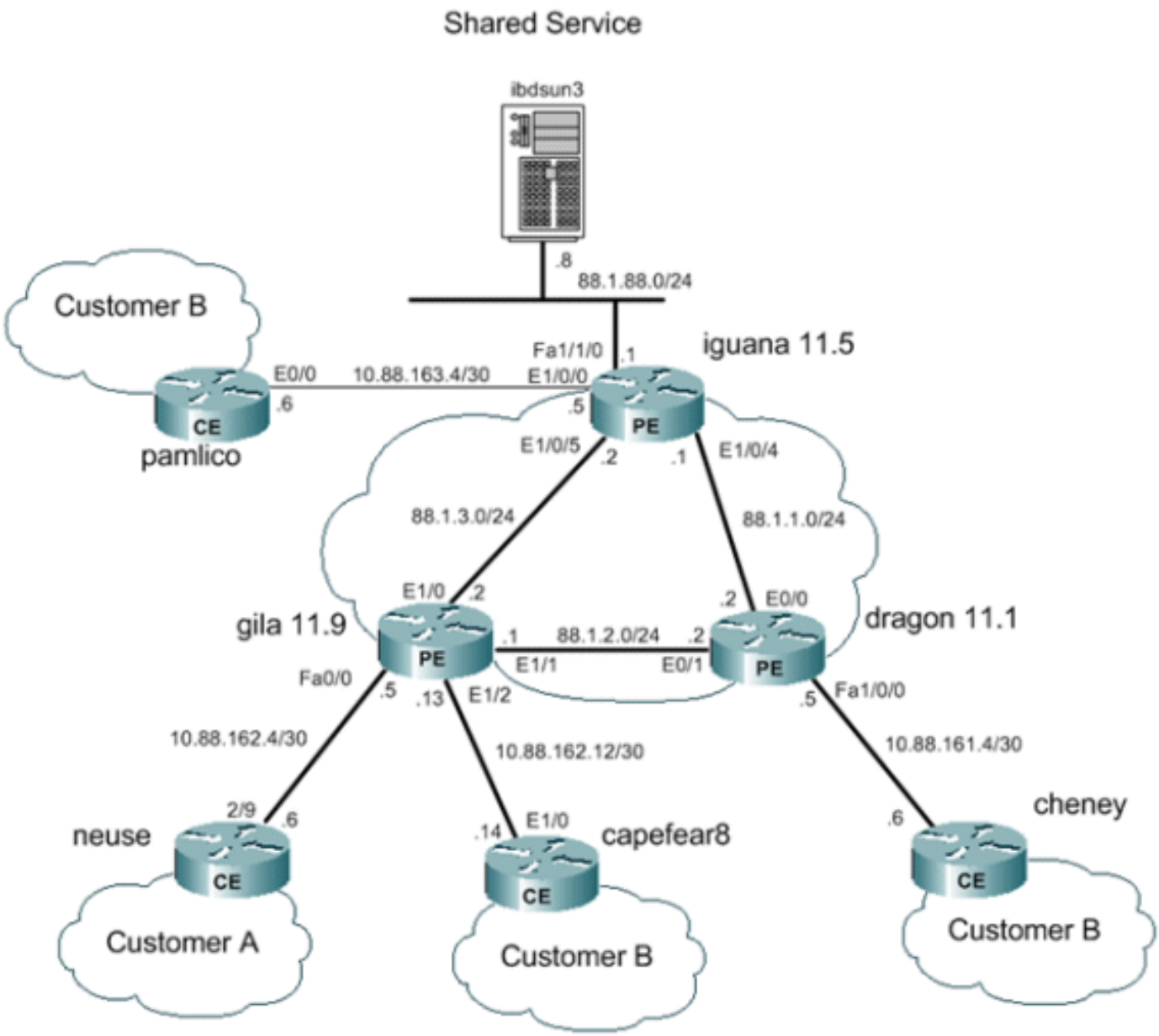
O design centralizado impõe uma restrição sobre como a rede de serviço compartilhado deve ser configurada. Especificamente, o uso da importação/exportação de rotas VPN MPLS entre uma VPN de serviço compartilhado e VPNs de cliente não é possível. Isso se deve à natureza da operação MPLS conforme especificado pelo RFC 2547 . Quando as rotas são importadas e exportadas usando as comunidades estendidas e os descritores de rotas, o NAT não pode determinar a VPN origem do pacote que chega ao NAT PE central. O caso comum é tornar a rede de serviço compartilhado uma interface genérica em vez de uma interface VRF. Uma rota para a rede de serviço compartilhado é adicionada no NAT PE central para cada tabela de VRF associada a uma VPN de cliente que precisa de acesso ao serviço compartilhado como parte do processo de provisionamento. Isso é descrito em mais detalhes posteriormente.

Opções de implantação e detalhes da configuração

Esta seção inclui alguns detalhes relacionados a cada uma das opções de implantação. Os exemplos são todos obtidos da rede mostrada na Figura 5. Consulte este diagrama para o resto desta seção.

Observação: na rede usada para ilustrar a operação do NAT de VRF para este documento, somente roteadores PE são incluídos. Não há roteadores "P" de núcleo. No entanto, os mecanismos essenciais continuam a ser visíveis.

Figura 5: Exemplo de configuração de NAT de VRF



NAT PE de saída

Neste exemplo, os roteadores de borda do provedor marcados como **gila** e **dragão** são configurados como roteadores PE simples. O PE central próximo à LAN de serviço compartilhado (**iguana**) é configurado para NAT. Um único pool NAT é compartilhado por cada VPN de cliente que precisa de acesso ao serviço compartilhado. O NAT é executado somente em pacotes destinados ao host de serviço compartilhado em 88.1.88.8.

Encaminhamento de dados NAT de saída

Com o MPLS, cada pacote entra na rede em um PE de entrada e sai da rede MPLS em um PE de saída. O caminho dos roteadores de comutação de rótulo atravessados da entrada para a saída é conhecido como caminho comutado por rótulo (LSP). O LSP é unidirecional. Um LSP diferente é usado para tráfego de retorno.

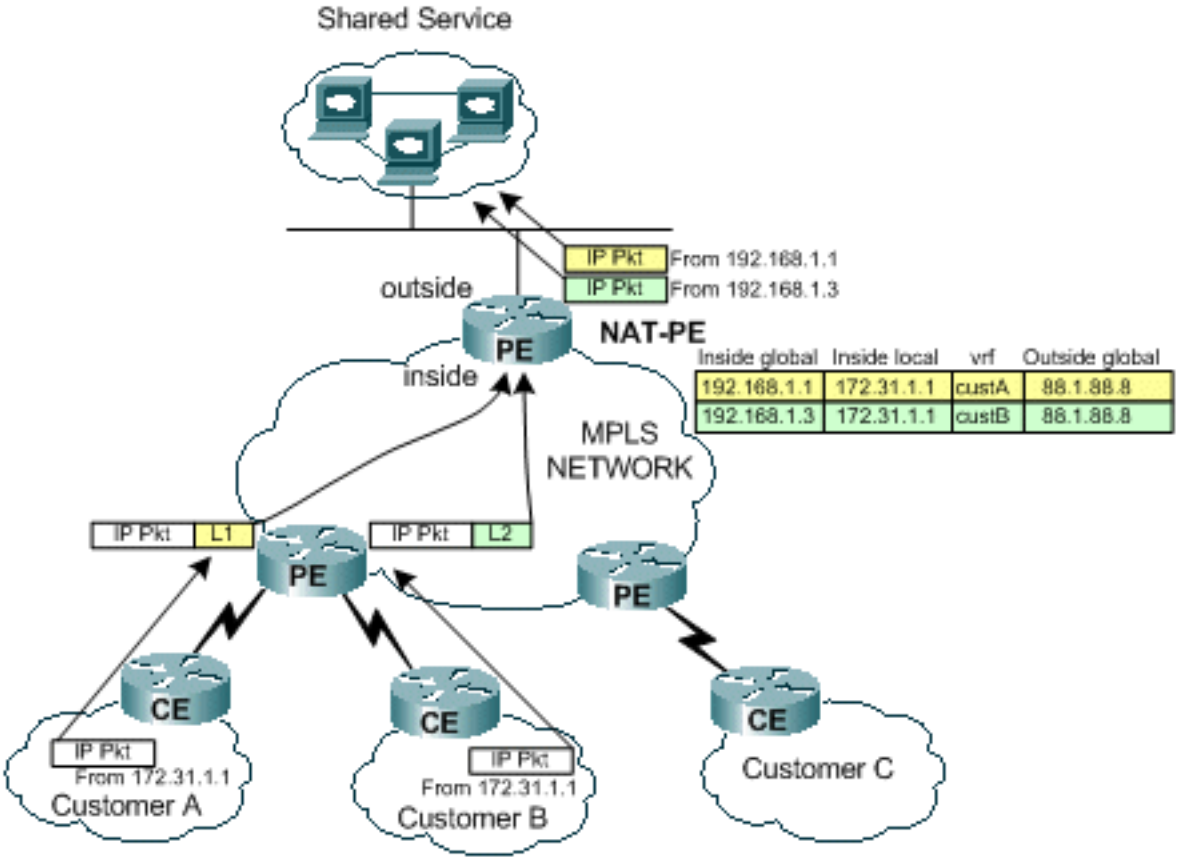
Ao usar o PE NAT de saída, uma classe de equivalência de encaminhamento (FEC) é efetivamente definida para todo o tráfego dos usuários do serviço compartilhado. Em outras palavras, todos os pacotes destinados à LAN de serviço compartilhado são membros de uma FEC comum. Um pacote é atribuído a um FEC específico apenas uma vez na borda de entrada

da rede e segue o LSP para o PE de saída. O FEC é designado no pacote de dados adicionando um rótulo específico.

Fluxo de pacote para serviço compartilhado da VPN

Para que os dispositivos em várias VPNs que têm esquemas de endereço sobrepostos acessem um host de serviço compartilhado, a NAT é necessária. Quando o NAT é configurado no PE de saída, as entradas da tabela de conversão de endereços de rede incluirão um identificador de VRF para diferenciar endereços duplicados e garantir o roteamento adequado.

Figura 6: Pacotes transmitidos para NAT PE de saída



A Figura 6 ilustra os pacotes destinados a um host de serviço compartilhado de duas VPNs de cliente que têm esquemas de endereçamento IP duplicados. A figura mostra um pacote originado no Cliente A com um endereço de origem 172.31.1.1 destinado a um servidor compartilhado em 88.1.88.8. Outro pacote do Cliente B com o mesmo endereço IP de origem também é enviado para o mesmo servidor compartilhado. Quando os pacotes chegam ao roteador PE, uma pesquisa da camada 3 é feita para a rede IP de destino na base de informações de encaminhamento (FIB).

A entrada FIB diz ao roteador PE para encaminhar o tráfego para o PE de saída usando uma pilha de rótulos. O rótulo inferior na pilha é atribuído pelo roteador PE de destino, nesse caso o roteador **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#
```

Podemos ver pela tela que os pacotes do CustA VRF terão um valor de tag de 24 (0x18) e os pacotes do CustB VRF terão um valor de tag de 28 (0x1C).

Nesse caso, como não há roteadores "P" em nossa rede, não há nenhuma marca adicional imposta. Se houvesse roteadores centrais, um rótulo externo teria sido imposto e o processo normal de troca de rótulo teria ocorrido na rede central até que o pacote chegasse ao PE de saída.

Como o roteador **gila** está conectado diretamente ao PE de saída, vemos que a marca é exibida antes de ser adicionada:

```
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#
```

As próximas exibições descrevem os pacotes de eco conforme recebidos pelo roteador de saída PE NAT (na interface E1/0/5 na iguana).

```
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".]
```

Esses pings resultam na criação das seguintes entradas na tabela NAT na iguana do roteador PE de saída. As entradas específicas criadas para os pacotes mostrados acima podem ser correspondidas pelo identificador ICMP.

```
iguana#
show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
```

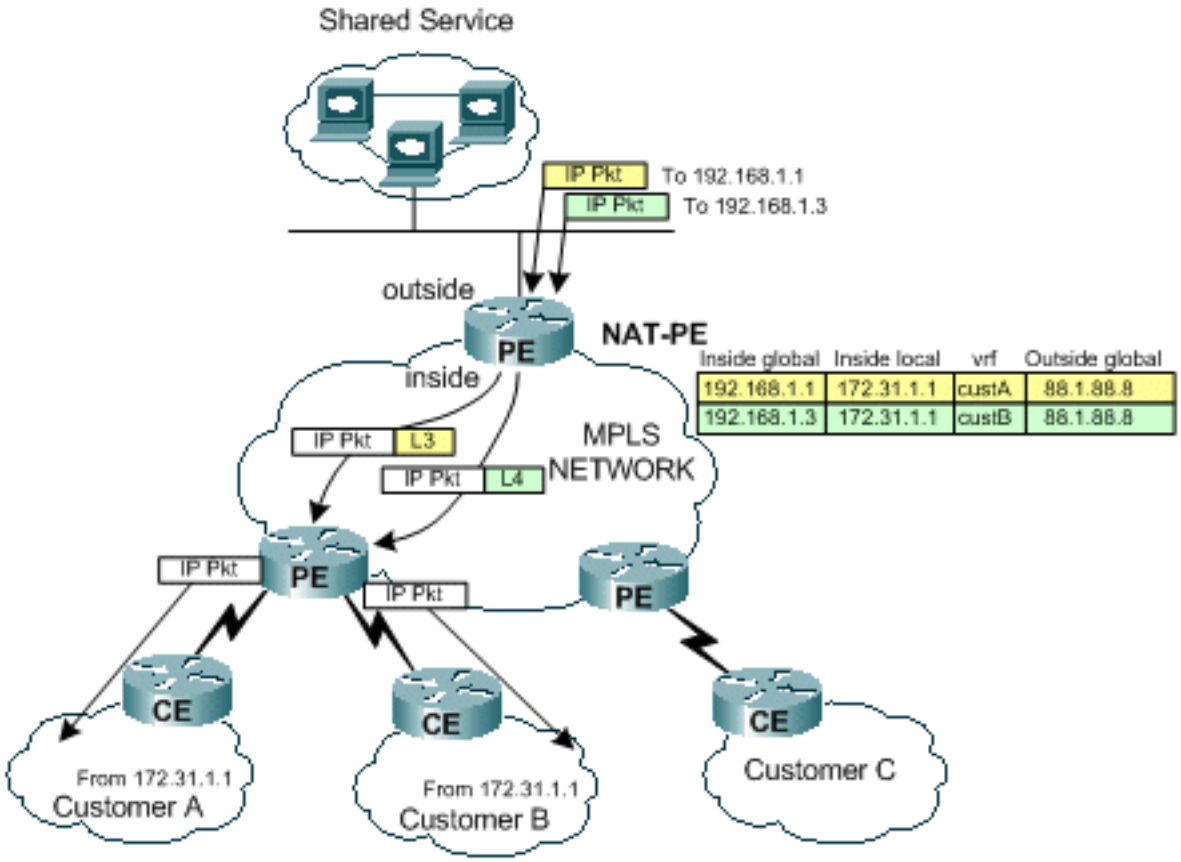
```
icmp 192.168.1.3:3365  172.31.1.1:3365    88.1.88.8:3365     88.1.88.8:3365
icmp 192.168.1.3:3366  172.31.1.1:3366    88.1.88.8:3366     88.1.88.8:3366
icmp 192.168.1.3:3367  172.31.1.1:3367    88.1.88.8:3367     88.1.88.8:3367
icmp 192.168.1.3:3368  172.31.1.1:3368    88.1.88.8:3368     88.1.88.8:3368
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#
```

Fluxo de pacote do serviço compartilhado de volta para a VPN de origem

Àmedida que os pacotes fluem de volta para os dispositivos que acessaram o host de serviço

compartilhado, a tabela NAT é examinada antes do roteamento (pacotes indo da interface "externa" da NAT para a interface "interna"). Como cada entrada exclusiva inclui o identificador de VRF correspondente, o pacote pode ser convertido e roteado adequadamente.

Figura 7: Pacotes transmitidos de volta ao usuário do serviço compartilhado



Como mostrado na Figura 7, o tráfego de retorno é examinado pela NAT primeiro para encontrar uma entrada de tradução correspondente. Por exemplo, um pacote é enviado ao destino 192.168.1.1. A tabela NAT é pesquisada. Quando a correspondência é encontrada, a tradução apropriada é feita para o endereço "local interno" (172.31.1.1) e, em seguida, uma pesquisa de adjacência é executada usando o ID de VRF associado da entrada NAT.

```
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#
```

O rótulo 23 (0x17) é usado para o tráfego destinado a 172.31.1.0/24 no VRF custA e o rótulo 26 (0x1A) é usado para pacotes destinados a 172.31.1.0/24 no VRF custB.

Isso é visto nos pacotes de resposta de eco enviados da iguana do roteador:

```
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".]
```

Quando o pacote chega ao roteador PE de destino, o rótulo é usado para determinar o VRF e a interface apropriados para enviar o pacote.

```
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#
```

Configurações

Algumas informações estranhas foram removidas das configurações para serem breves.

```
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
!
```

O **dragão** do roteador teria uma configuração muito semelhante à de **gila**.

Importação/Exportação de Destinos de Rota Não Permitidos

Quando a rede de serviço compartilhado é configurada como uma instância de VRF propriamente dita, o NAT central no PE de saída não é possível. Isso ocorre porque os pacotes de entrada não podem ser diferenciados e apenas uma rota de volta para a sub-rede de origem está presente no PE NAT de saída.

Observação: as exibições a seguir devem ilustrar o resultado de uma configuração inválida.

A rede de exemplo foi configurada para que a rede de serviço compartilhado fosse definida como uma instância de VRF (nome de VRF = servidor). Agora, uma tela da tabela CEF no PE de entrada mostra o seguinte:

```
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#
```

Observação: observe como o valor de tag *24* é imposto para o CustA do VRF e para o CustB do VRF.

Esta exibição mostra a tabela de roteamento para o "servidor" da instância de VRF do serviço compartilhado:

```
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
```

Observação: somente uma rota está presente para a rede de destino da perspectiva do roteador PE de saída (**iguana**).

Portanto, o tráfego de várias VPNs de clientes não pôde ser diferenciado e o tráfego de retorno não pôde alcançar a VPN apropriada. **Caso o serviço compartilhado deva ser definido como uma instância de VRF, a função NAT deve ser movida para o PE de entrada.**

NAT PE de entrada

Neste exemplo, os roteadores de borda do provedor marcados como **gila** e **dragão** estão configurados para NAT. Um pool NAT é definido para cada VPN de cliente conectado que precisa de acesso ao serviço compartilhado. O pool apropriado é usado para cada um dos endereços de rede do cliente que são NATed. O NAT é executado somente em pacotes destinados ao host de serviço compartilhado em 88.1.88.8.

```
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
```

Observação: neste cenário, os pools compartilhados não são suportados. Se a LAN do serviço compartilhado (no PE de saída) estiver conectada por uma interface genérica, o pool NAT pode ser compartilhado.

Um ping originado de um endereço duplicado (172.31.1.1) em cada uma das redes conectadas à **neuse** e **capefear8** resulta nestas entradas NAT:

De **Gila**:

```
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
```

Observação: o mesmo endereço local interno (172.31.1.1) é convertido para cada um dos pools definidos de acordo com o VRF de origem. O VRF pode ser visto no comando **show ip nat**

translation verbose:

```
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
```

Estas exibições mostram as informações de roteamento para cada uma das VPNs conectadas localmente para o cliente A e o cliente B:

```
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
```

Observação: uma rota para cada um dos pools de NAT foi adicionada da configuração estática. Essas sub-redes são posteriormente importadas para o servidor compartilhado VRF na iguana do roteador PE de saída:

```
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
```

Configurações

Algumas informações estranhas foram removidas das configurações para serem breves.

```
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
!
```

Observação: as interfaces que enfrentam as redes do cliente são designadas como interfaces "internas" de NAT e as interfaces de MPLS são designadas como interfaces "externas" de NAT.

```
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
```

O **dragão** do roteador teria uma configuração muito semelhante à de **gila**.

Pacotes chegando ao PE central após a entrada do PE NAT

Os rastreamentos abaixo ilustram a exigência de pools de NAT exclusivos quando a rede de

serviço compartilhado de destino é configurada como uma instância de VRF. Mais uma vez, consulte o diagrama na Figura 5. Os pacotes mostrados abaixo foram capturados à medida que entravam na interface IP e1/0/5 do MPLS na **iguana** do roteador.

Eco do cliente A VPN

Aqui, vemos uma solicitação de eco vinda do endereço IP de origem 172.31.1.1 no VRF custA. O endereço de origem foi convertido para 192.168.1.1 conforme especificado pela configuração do NAT:

```
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:
```

Eco do cliente B VPN

Aqui, vemos uma solicitação de eco vinda do endereço IP de origem 172.31.1.1 no VRF custB. O endereço de origem foi convertido para 192.168.2.1 conforme especificado pela configuração do NAT:

```
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".]
```

Observação: o valor do rótulo MPLS é *0019* em ambos os pacotes mostrados acima.

Resposta de eco para VPN do cliente A

Em seguida, vemos uma resposta de eco retornando para o endereço IP destino 192.168.1.1 no VRF custA. O endereço de destino é convertido para 172.31.1.1 pela função de NAT do PE de entrada.

```
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:
```

Resposta de eco para VPN do cliente B

Aqui, vemos uma resposta de eco retornando ao endereço IP destino 192.168.1.1 no VRF custB. O endereço de destino é convertido para 172.31.1.1 pela função de NAT do PE de entrada.

```
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".]
```

Observação: nos pacotes de retorno, os valores de rótulo MPLS são incluídos e diferentes: *001A* para o crosta VRFA e *001D* para o crosta VRFB.

Eco do cliente A VPN - o destino é uma interface genérica

Esse próximo conjunto de pacotes mostra a diferença quando a interface para a LAN de serviço compartilhado é uma interface genérica e não parte de uma instância de VRF. Aqui, a configuração foi alterada para usar um pool comum para ambas as VPNs locais com endereços IP sobrepostos.

```
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".]
```

Eco do VPN do cliente B - O destino é uma interface genérica

Aqui, vemos uma solicitação de eco vinda do endereço IP de origem 172.31.1.1 no VRF custB. O endereço de origem foi convertido para 192.168.1.3 (do pool comum SSPOOL1) conforme especificado pela configuração do NAT:

```
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".]
```

Observação: quando a interface no PE de saída é uma interface genérica (não uma instância de VRF), os rótulos impostos são diferentes. Nesse caso, *0x19* e *0x1F*.

Resposta de eco ao cliente A VPN - o destino é uma interface genérica

Em seguida, vemos uma resposta de eco retornando para o endereço IP destino 192.168.1.1 no VRF custA. O endereço de destino é convertido para 172.31.1.1 pela função de NAT do PE de entrada.

```
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".]
```

Resposta de Eco para VPN do Cliente B - O destino é uma interface genérica

Aqui, vemos uma resposta de eco retornando ao endereço IP destino 192.168.1.3 no VRF custB. O endereço de destino é convertido para 172.31.1.1 pela função de NAT do PE de entrada.

```
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".]
```

Observação: como as respostas são destinadas a um endereço global, nenhum rótulo de VRF é imposto.

Com a interface de saída para o segmento de LAN de serviço compartilhado definido como uma interface genérica, um pool comum é permitido. Os pings resultam nessas entradas NAT na **gila** do roteador:

```
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
```

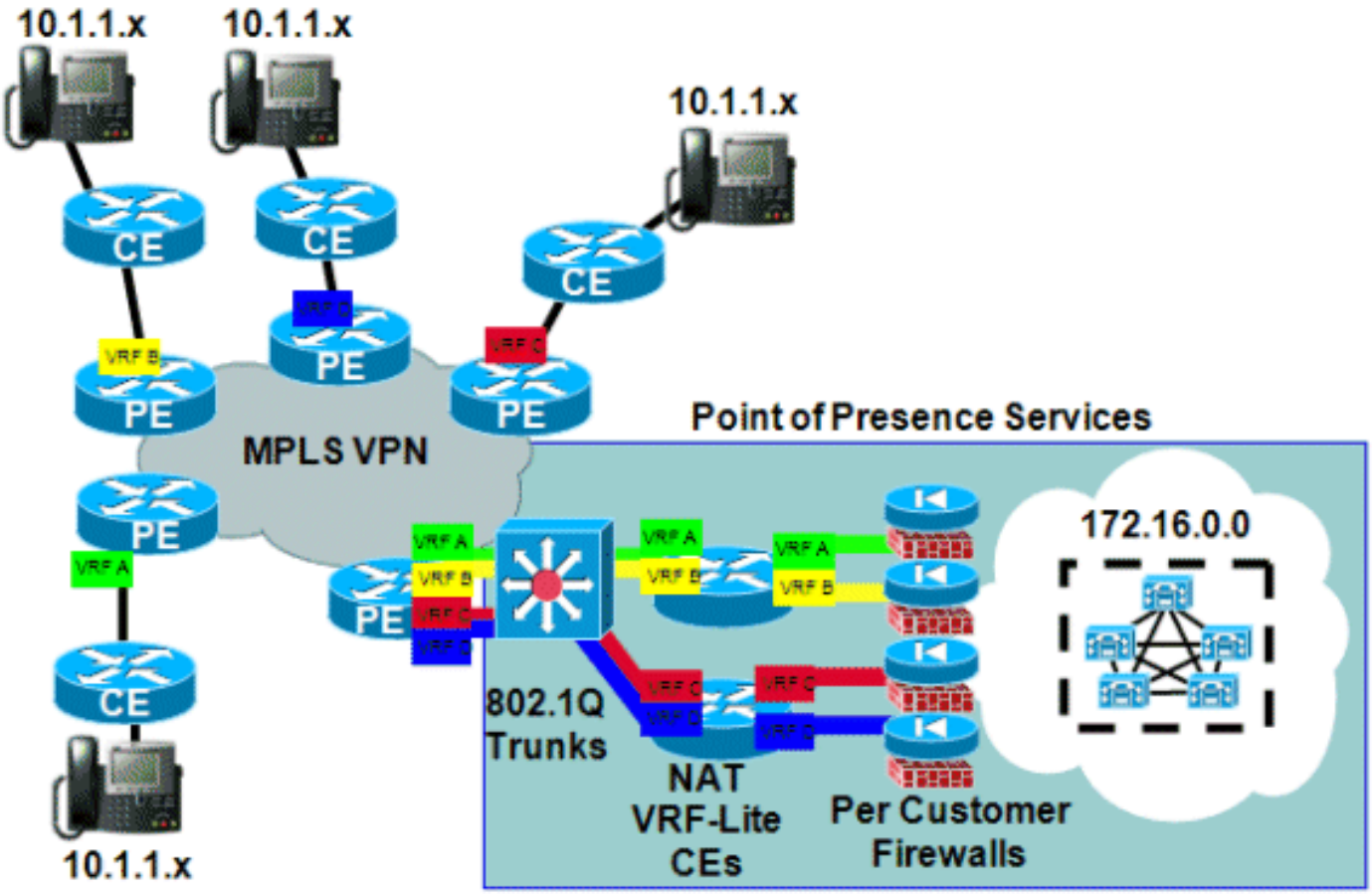
```
.Jan  2 09:35:02 EST: NAT-TAGSW(p) : Tag Pkt s=172.18.60.179, d=10.88.162.13, vrf=custB
.Jan  2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan  2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan  2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA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

Exemplo de serviço

Um exemplo de um serviço PBX IP virtual compartilhado é mostrado na Figura 8. Isso ilustra uma variante dos exemplos de entrada e saída descritos anteriormente.

Neste design, o serviço de VoIP compartilhado é front-end por um conjunto de roteadores que executam a função de NAT. Esses roteadores têm várias interfaces VRF usando um recurso conhecido como VRF-Lite. O tráfego então flui para o cluster compartilhado do Cisco CallManager. Os serviços de firewall também são fornecidos por empresa. As chamadas entre empresas devem passar pelo firewall, enquanto as chamadas dentro da empresa são tratadas pela VPN do cliente usando o esquema de endereçamento interno da empresa.

Figura 8: Exemplo de serviço de PBX virtual gerenciado



Disponibilidade

O suporte de NAT do Cisco IOS para VPNs MPLS está disponível no Cisco IOS versão 12.2(13)T e está disponível para todas as plataformas que suportam MPLS e podem executar esse treinamento de versão de implantação inicial.

Conclusão

A NAT do Cisco IOS tem recursos para permitir a implantação escalável de serviços compartilhados atualmente. A Cisco continua a desenvolver o suporte de gateway de nível de aplicativo (ALG) NAT para protocolos importantes para os clientes. Melhorias no desempenho e aceleração de hardware para funções de tradução garantirão que NAT e ALGs forneçam soluções aceitáveis por algum tempo. Todas as atividades de padrões relevantes e ações comunitárias estão sendo monitoradas pela Cisco. À medida que outros padrões forem desenvolvidos, seu uso será avaliado com base nos desejos, requisitos e aplicativos do cliente.

Informações Relacionadas

- Gateways da camada de aplicação NAT do Cisco IOS
- Arquiteturas MPLS e VPN
- Projeto e implementação avançados de MPLS
- Suporte Técnico e Documentação - Cisco Systems