NAT de Cisco IOS - Integración con MPLS VPN

Contenido

Introducción

El software de traducción de direcciones de red (NAT) de Cisco IOS® permite el acceso a servicios compartidos desde varias VPN MPLS, incluso cuando los dispositivos de las VPN utilizan direcciones IP que se superponen. Cisco IOS NAT está preparado para VRF y se puede configurar en los routers de borde del proveedor dentro de la red MPLS.

Nota: MPLS en IOS se soporta solamente con NAT heredada. En este momento, no hay soporte en Cisco IOS para NAT NVI con MPLS.

Se prevé que la implementación de VPN MPLS aumente rápidamente en los próximos años. Las ventajas de una infraestructura de red común que permita una rápida expansión y opciones de conectividad flexibles impulsarán sin duda un mayor crecimiento de los servicios que se pueden ofrecer a la comunidad de redes internas.

Sin embargo, todavía persisten obstáculos al crecimiento. IPv6 y su promesa de disponer de un espacio de direcciones IP que supere las necesidades de conectividad en un futuro próximo se encuentran todavía en las primeras fases de la implementación. Las redes existentes suelen utilizar esquemas de direccionamiento IP privados según lo definido en RFC 1918 . La traducción de direcciones de red se utiliza a menudo para interconectar redes cuando los espacios de direcciones se superponen o se duplican.

Los proveedores de servicios y las empresas que disponen de servicios de aplicaciones de red que desean ofrecer o compartir con clientes y partners querrán minimizar cualquier carga de conectividad que pesen sobre el usuario del servicio. Es deseable, incluso obligatorio, extender la oferta a tantos usuarios potenciales como sea necesario para lograr los objetivos deseados o el retorno. El esquema de direccionamiento IP en uso no debe ser una barrera que excluya a los usuarios potenciales.

Al implementar Cisco IOS NAT en la infraestructura VPN MPLS común, los proveedores de servicios de comunicaciones pueden aliviar parte de la carga de conectividad de los clientes y acelerar su capacidad de vincular más servicios de aplicaciones compartidas a más consumidores de esos servicios.

Ventajas de NAT - Integración de MPLS

La integración de NAT con MPLS ofrece ventajas tanto para los proveedores de servicios como para sus clientes empresariales. Ofrece a los proveedores de servicios más opciones para implementar servicios compartidos y proporcionar acceso a esos servicios. Las ofertas de servicios adicionales pueden marcar la diferencia con respecto a la competencia.

Para proveedor de servicios	Para VPN
Más ofertas de servicios	Reducción de costes
Más opciones de acceso	Acceso más sencillo
Aumento de los ingresos	Abordar la flexibilidad

Los clientes empresariales que deseen externalizar parte de su carga de trabajo actual también pueden beneficiarse de ofertas más amplias de los proveedores de servicios. Cambiar la carga de realizar cualquier traducción de dirección necesaria a la red del proveedor de servicios los libera de una tarea administrativa complicada. Es posible que los clientes continúen utilizando el direccionamiento privado y, al mismo tiempo, mantengan el acceso a los servicios compartidos y a Internet. La consolidación de la función NAT dentro de la red del proveedor de servicios también puede reducir el costo total para los clientes empresariales, ya que los routers de borde del cliente no tienen que realizar la función NAT.

Aspectos del diseño

Al considerar los diseños que invocarán NAT dentro de la red MPLS, el primer paso es determinar las necesidades de servicio desde el punto de vista de una aplicación. Deberá tener en cuenta los protocolos utilizados y cualquier comunicación especial cliente/servidor impuesta por la aplicación. Asegúrese de que el soporte necesario para los protocolos empleados sea soportado y manejado por Cisco IOS NAT. En el documento Gateways de Capa de Aplicación NAT de Cisco IOS se proporciona una lista de los protocolos soportados.

A continuación, será necesario determinar el uso esperado del servicio compartido y la tasa de tráfico esperada en paquetes por segundo. NAT es una función de uso intensivo de la CPU del router. Por lo tanto, los requisitos de rendimiento serán un factor a la hora de seleccionar una opción de implementación determinada y determinar el número de dispositivos NAT involucrados.

Además, tenga en cuenta cualquier problema de seguridad y las precauciones que deban tomarse. Aunque las VPN MPLS, por definición, son tráfico privado y separado de manera efectiva, la red de servicio compartido es generalmente común entre muchas VPN.

Escenarios de implementación

Hay dos opciones para la implementación de NAT dentro del perímetro del proveedor MPLS:

- Centralizado con NAT de salida PE

- Distribuido con NAT PE de ingreso

Algunas ventajas para configurar la función NAT en el punto de salida de la red MPLS más cercana a la red de servicio compartido son:

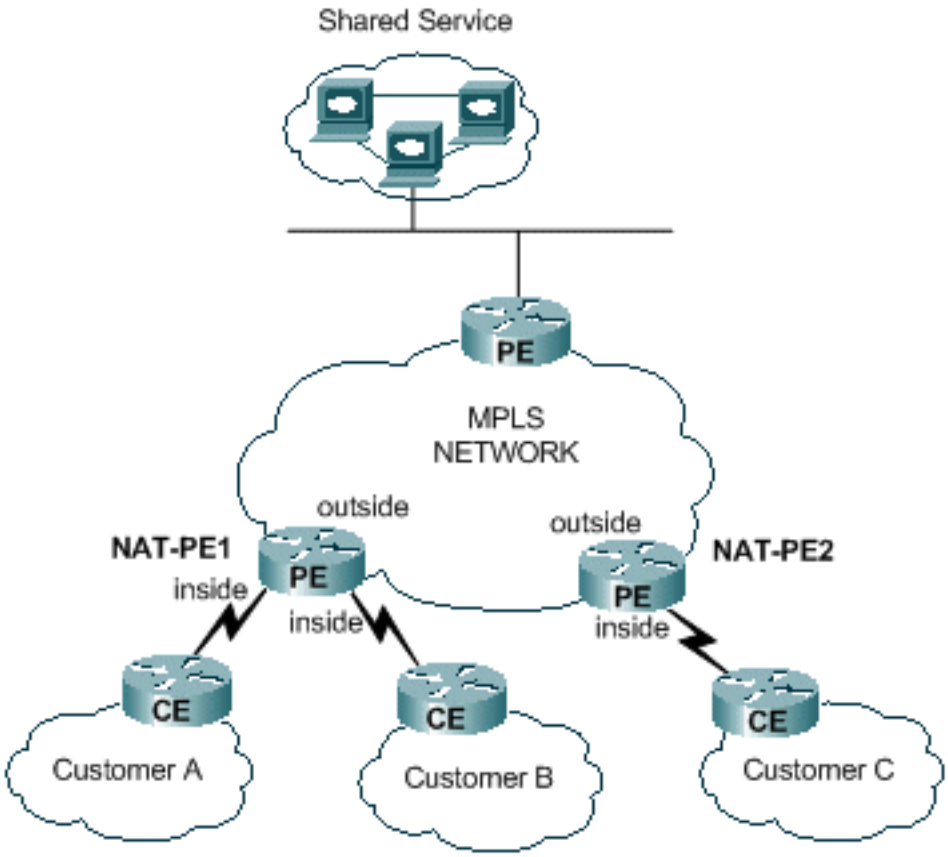
- Una configuración centralizada que fomenta un aprovisionamiento de servicios más sencillo
- Resolución de problemas simplificada
- Escalabilidad operativa mejorada
- Disminución de los requisitos de asignación de direcciones IP

Sin embargo, las ventajas se compensan con una reducción de la escalabilidad y el rendimiento. Esta es la principal disyuntiva que debe considerarse. Por supuesto, la función NAT también se puede realizar dentro de las redes del cliente si se determina que la integración de esta función con una red MPLS no es deseable.

NAT PE de entrada

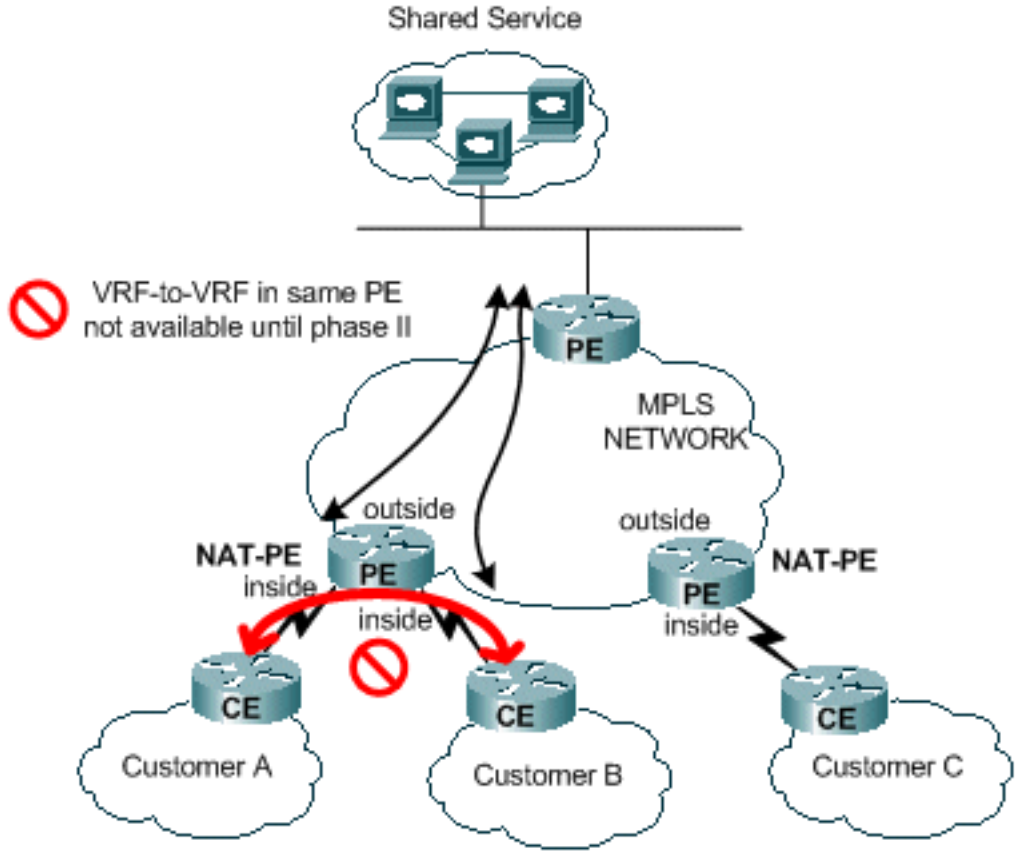
La NAT se puede configurar en el router PE de ingreso a la red MPLS como se muestra en la Figura 1. Con este diseño, la escalabilidad se mantiene en gran medida mientras que el rendimiento se optimiza mediante la distribución de la función NAT en muchos dispositivos periféricos. Cada NAT PE maneja el tráfico para los sitios conectados localmente a ese PE. Las reglas NAT y las listas de control de acceso o los mapas de ruta controlan qué paquetes requieren traducción.

Figura 1: NAT PE de entrada



Hay una restricción que impide la NAT entre dos VRF al tiempo que proporciona NAT a un servicio compartido como se muestra en la Figura 2. Esto se debe al requisito de designar las interfaces como interfaces NAT "interna" y "externa". Se planifica el soporte para conexiones entre VRF en un único PE para una futura versión de Cisco IOS.

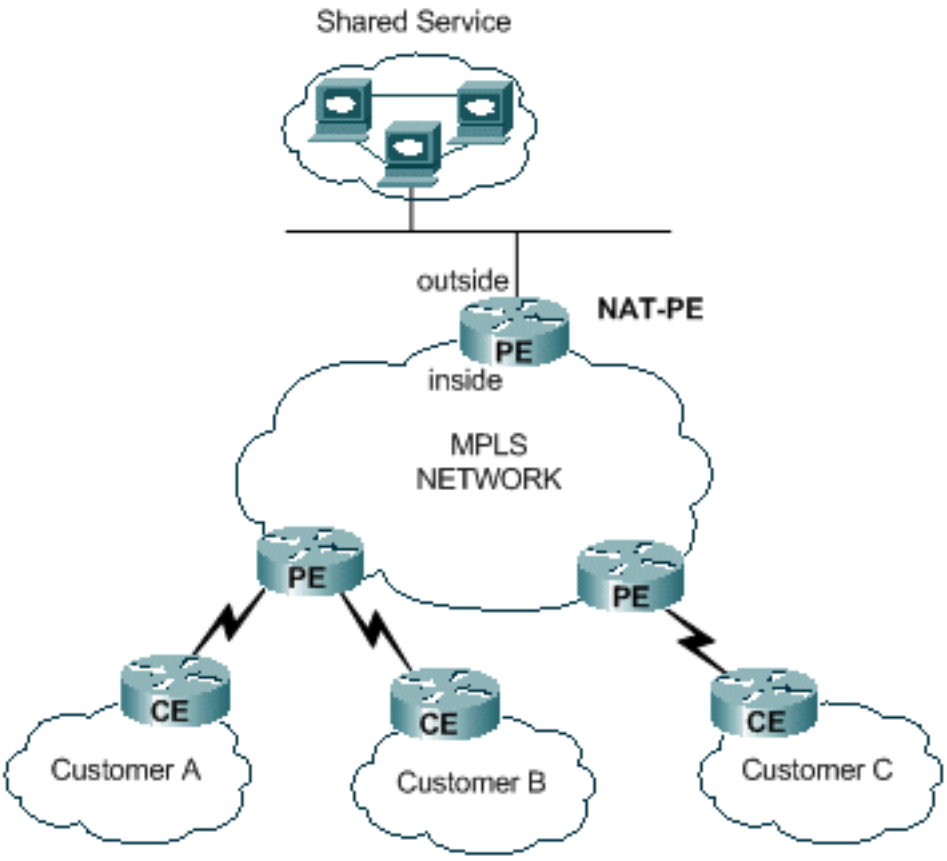
Figura 2: Empresa a empresa



NAT PE de salida

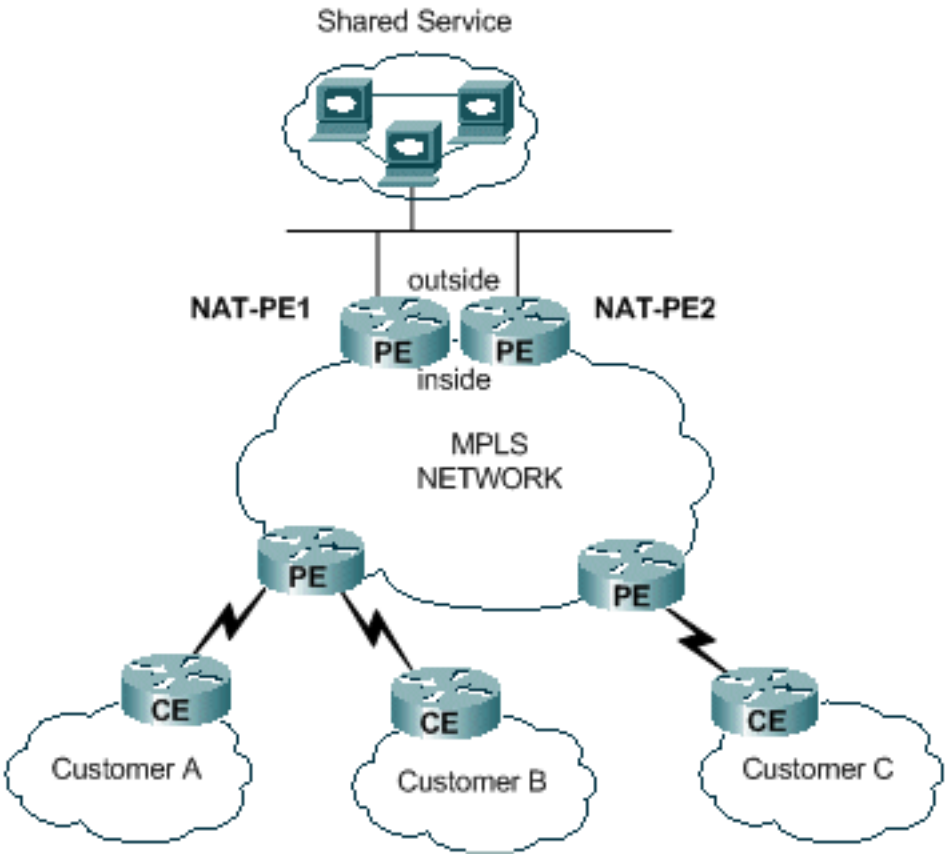
La NAT se puede configurar en el router PE de salida de red MPLS como se muestra en la Figura 3. Con este diseño, la escalabilidad se reduce en cierta medida, ya que el PE central debe mantener rutas para todas las redes de clientes que acceden al servicio compartido. Los requisitos de rendimiento de la aplicación también deben tenerse en cuenta para que el tráfico no sobrecargue al router que debe traducir las direcciones IP de los paquetes. Debido a que la NAT se produce de forma centralizada para todos los clientes que utilizan esta ruta, los grupos de direcciones IP se pueden compartir; por lo tanto, se reduce el número total de subredes necesarias.

Figura 3: NAT PE de salida



Se podrían implementar varios routers para aumentar la escalabilidad del diseño NAT PE de salida, como se muestra en la Figura 4. En esta situación, las VPN del cliente se podrían "aprovisionar" en un router NAT específico. La traducción de direcciones de red se produciría para el tráfico agregado hacia y desde el servicio compartido para ese conjunto de VPN. Por ejemplo, el tráfico de las VPN para los clientes A y B podría utilizar NAT-PE1, mientras que el tráfico hacia y desde la VPN para el cliente C utiliza NAT-PE2. Cada NAT PE transportaría tráfico solamente para las VPNs específicas definidas y solamente mantendría las rutas de regreso a los sitios en esas VPNs. Se pueden definir grupos de direcciones NAT separados dentro de cada uno de los routers NAT PE para que los paquetes se rutee desde la red de servicio compartido al PE NAT adecuado para la traducción y el ruteo de regreso a la VPN del cliente.

Figura 4: NAT PE de salida múltiple



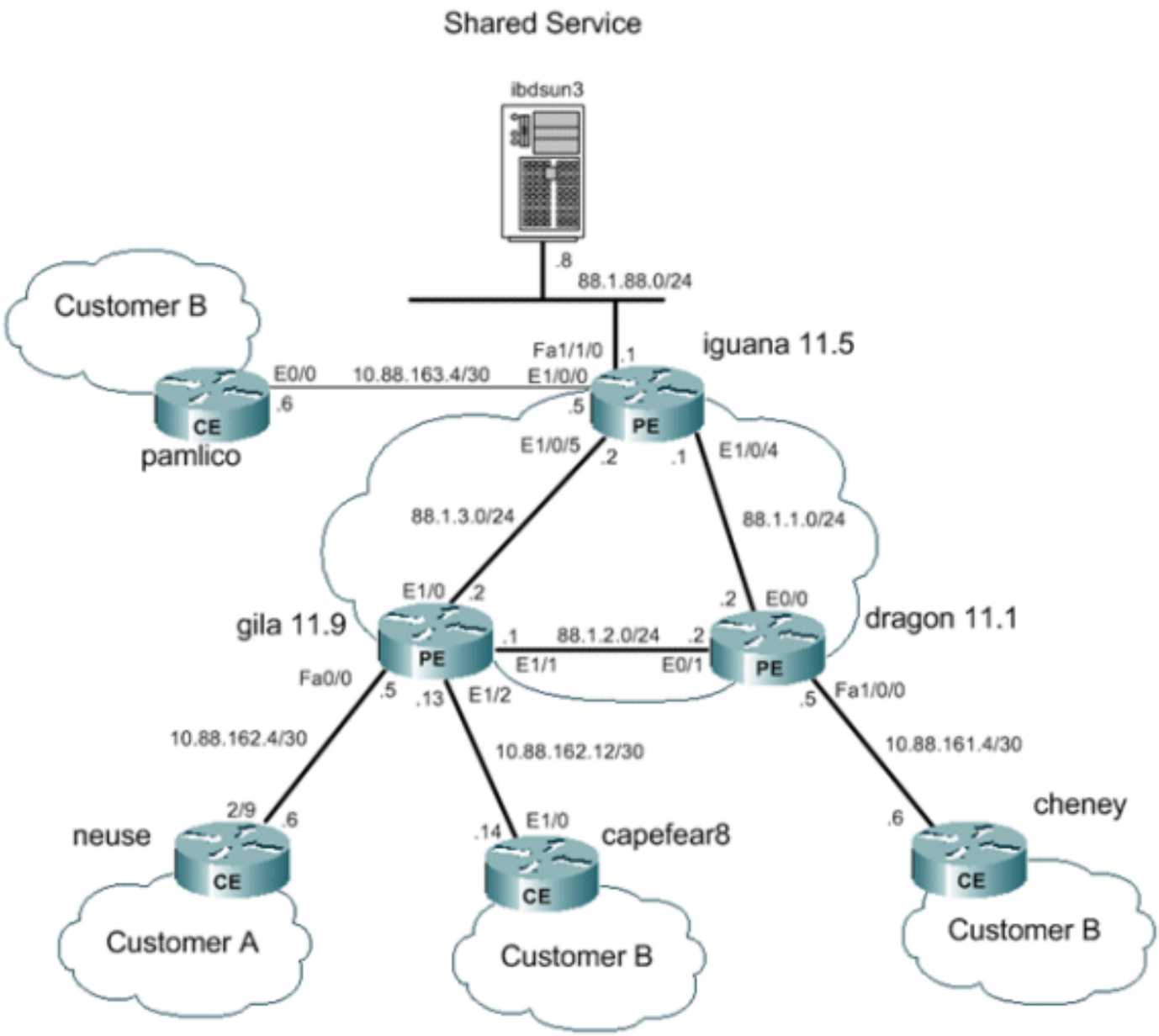
El diseño centralizado sí impone una restricción sobre cómo se debe configurar la red de servicio compartido. Específicamente, el uso de la importación/exportación de rutas VPN MPLS entre una VPN de servicio compartido y VPN de cliente no es posible. Esto se debe a la naturaleza de la operación MPLS según lo especificado por RFC 2547 . Cuando las rutas se importan y exportan usando las comunidades extendidas y los descriptores de ruta, NAT no puede determinar la VPN de origen del paquete que ingresa al PE de NAT central. El caso habitual es hacer de la red de servicio compartido una interfaz genérica en lugar de una interfaz VRF. A continuación, se agrega una ruta a la red de servicio compartido en el PE de NAT central para cada tabla VRF asociada a una VPN del cliente que necesita acceso al servicio compartido como parte del proceso de aprovisionamiento. Esto se describe con más detalle más adelante.

Opciones de implementación y detalles de configuración

Esta sección incluye algunos detalles relacionados con cada una de las opciones de implementación. Todos los ejemplos se toman de la red que se muestra en la Figura 5. Consulte este diagrama para ver el resto de esta sección.

Nota: En la red utilizada para ilustrar el funcionamiento de la NAT VRF para este documento, sólo se incluyen los routers PE. No hay routers "P" de núcleo. Sin embargo, todavía se pueden ver los mecanismos esenciales.

Figura 5: Ejemplo de Configuración de NAT de VRF



NAT PE de salida

En este ejemplo, los routers de borde del proveedor marcados con **gila** y **dragon** se configuran como routers PE simples. El PE central cerca de la LAN de servicio compartido (**iguana**) está configurado para NAT. Cada VPN del cliente que necesita acceso al servicio compartido comparte un único conjunto NAT. La NAT se realiza solamente en los paquetes destinados al host de servicio compartido en 88.1.88.8.

Reenvío de datos NAT PE de salida

Con MPLS, cada paquete ingresa a la red en un PE de ingreso y sale de la red MPLS en un PE de egreso. La ruta de los routers de switching de etiquetas que pasan de entrada a salida se conoce como la ruta conmutada de etiquetas (LSP). El LSP es unidireccional. Se utiliza un LSP diferente para el tráfico de retorno.

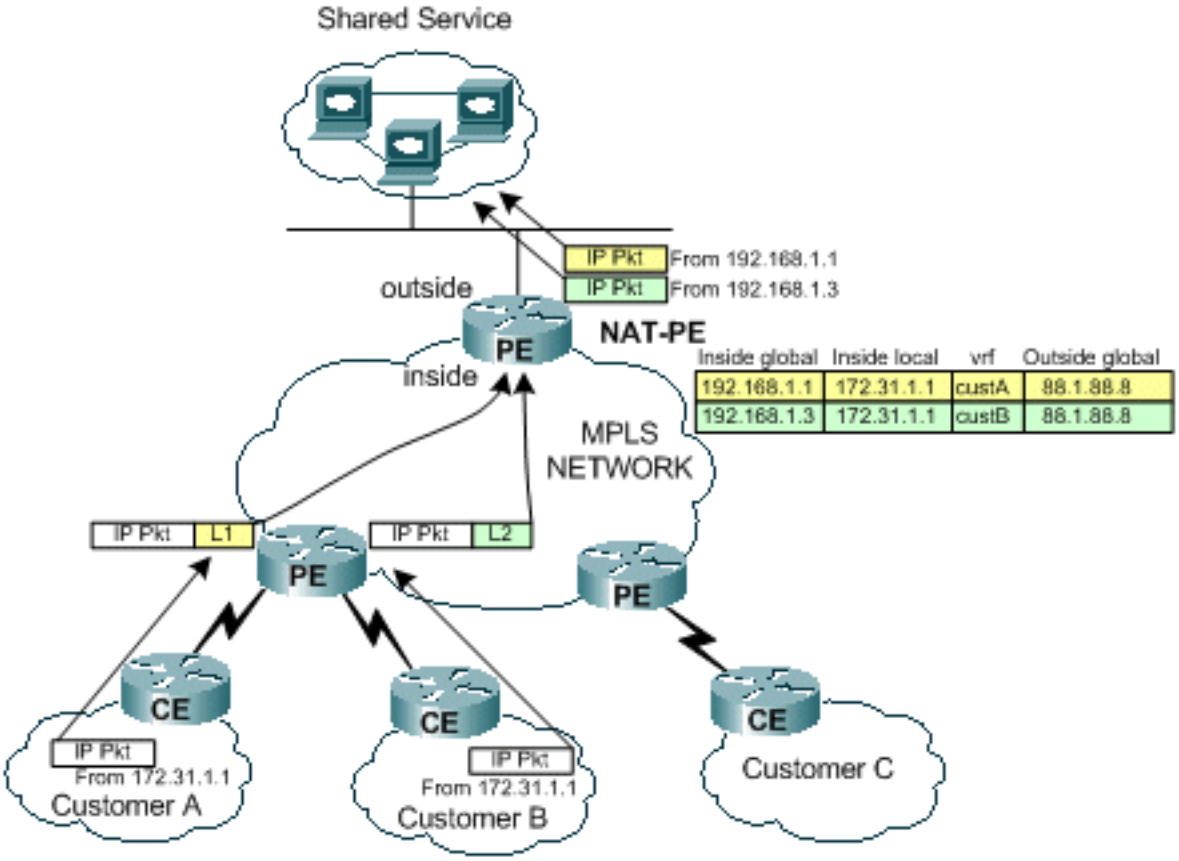
Al utilizar NAT PE de salida, se define de forma eficaz una clase de equivalencia de reenvío (FEC) para todo el tráfico de los usuarios del servicio compartido. En otras palabras, todos los paquetes destinados a la LAN de servicio compartido son miembros de una FEC común. Un paquete se asigna a un FEC determinado una sola vez en el borde de ingreso de la red y sigue el

LSP al PE de egreso. La FEC se designa en el paquete de datos mediante la adición de una etiqueta determinada.

Flujo de paquetes a servicio compartido desde VPN

Para que los dispositivos en varias VPN que tienen esquemas de direcciones superpuestas accedan a un host de servicio compartido, se requiere NAT. Cuando NAT se configura en el PE de salida, las entradas de la tabla de traducción de direcciones de red incluirán un identificador VRF para diferenciar las direcciones duplicadas y asegurar el ruteo adecuado.

Figura 6: Paquetes transmitidos a NAT PE de salida



La figura 6 ilustra los paquetes destinados a un host de servicio compartido desde dos VPN de cliente que tienen esquemas de direccionamiento IP duplicados. La figura muestra un paquete que se origina en el Cliente A con una dirección de origen de 172.31.1.1 destinado a un servidor compartido a 88.1.88.8. Otro paquete del Cliente B con la misma dirección IP de origen también se envía al mismo servidor compartido. Cuando los paquetes llegan al router PE, se realiza una búsqueda de capa 3 para la red IP de destino en la base de información de reenvío (FIB).

La entrada FIB indica al router PE que reenvíe el tráfico al PE de salida usando una pila de etiquetas. El router PE de destino asigna la etiqueta inferior en la pila, en este caso, **iguana** del router.

```
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 en la pantalla que los paquetes de VRF CastA tendrán un valor de etiqueta de 24 (0x18) y los paquetes de VRF CastB tendrán un valor de etiqueta de 28 (0x1C).

En este caso, como no hay routers "P" en nuestra red, no se ha impuesto ninguna etiqueta adicional. Si hubiera habido routers de núcleo, se habría impuesto una etiqueta externa y el proceso normal de intercambio de etiquetas habría tenido lugar dentro de la red de núcleo hasta que el paquete alcanzara el PE de salida.

Dado que el router **gila** está conectado directamente al PE de salida, vemos que la etiqueta se muestra antes de que se añada alguna vez:

```
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#
```

La siguiente visualización muestra los paquetes de eco recibidos por el router NAT PE de egreso (en la interfaz E1/0/5 en 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".]
```

Estos pings resultan en que se crean las siguientes entradas en la tabla NAT en la iguana del router PE de salida. Las entradas específicas creadas para los paquetes que se muestran arriba pueden coincidir con su 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
```

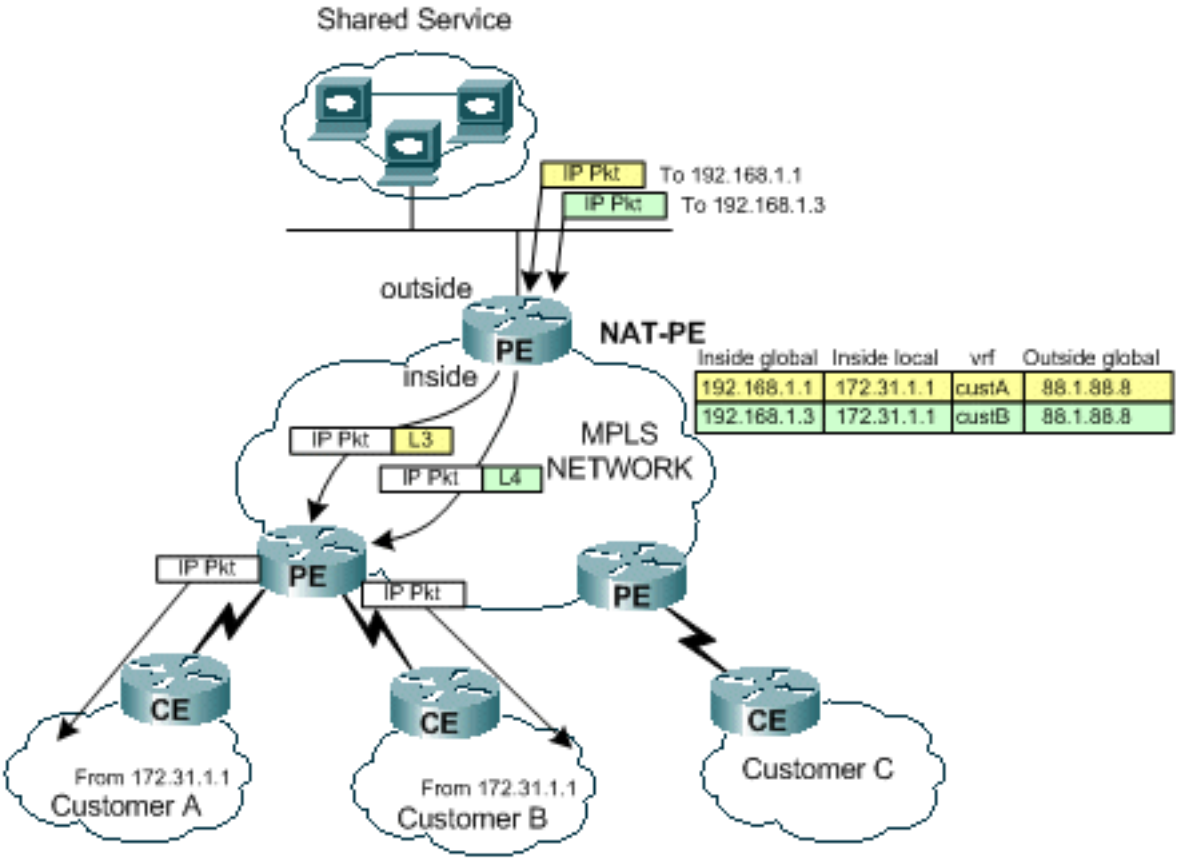
```
icmp 192.168.1.3:3366  172.31.1.1:3366    88.1.88.8:3366     88.1.88.8:3366
icmp 192.168.1.3:3367  172.31.1.1:3367    88.1.88.8:3367     88.1.88.8:3367
icmp 192.168.1.3:3368  172.31.1.1:3368    88.1.88.8:3368     88.1.88.8:3368
icmp 192.168.1.3:3369  172.31.1.1:3369    88.1.88.8:3369     88.1.88.8:3369
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#
```

Flujo de paquetes de un servicio compartido a una VPN de origen

A medida que los paquetes regresan a los dispositivos que han accedido al host de servicio compartido, la tabla NAT se examina antes del ruteo (paquetes que van desde la interfaz

"externa" de NAT a la interfaz "interna"). Debido a que cada entrada única incluye el identificador VRF correspondiente, el paquete se puede traducir y rutear adecuadamente.

Figura 7: Paquetes devueltos al usuario de servicios compartidos



Como se muestra en la Figura 7, la NAT examina primero el tráfico de retorno para encontrar una entrada de traducción coincidente. Por ejemplo, un paquete se envía al destino 192.168.1.1. Se busca en la tabla NAT. Cuando se encuentra la coincidencia, la traducción apropiada se realiza a la dirección "local interna" (172.31.1.1) y luego se realiza una búsqueda de adyacencia usando el ID de VRF asociado de la 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#
```

La etiqueta 23 (0x17) se utiliza para el tráfico destinado a 172.31.1.0/24 en VRF custA y la etiqueta 26 (0x1A) se utiliza para los paquetes destinados a 172.31.1.0/24 en VRF custB.

Esto se ve en los paquetes de respuesta de eco enviados desde la iguana del router:

```
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".]
```

Cuando el paquete alcanza el router PE de destino, la etiqueta se utiliza para determinar el VRF y

la interfaz adecuados para enviar el paquete.

```
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#
```

Configuraciones

Se ha eliminado cierta información superflua de las configuraciones para su brevedad.

```
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
!
```

El **dragón** del router tendría una configuración muy similar a **gila**.

Importación/Exportación de Destinos de Ruta no Permitidos

Cuando la red de servicio compartido se configura como una instancia VRF en sí, la NAT central en el PE de salida no es posible. Esto se debe a que los paquetes entrantes no se pueden distinguir y sólo una ruta de regreso a la subred de origen está presente en la NAT PE de salida.

Nota: Las visualizaciones que se muestran a continuación están pensadas para ilustrar el resultado de una configuración no válida.

La red de ejemplo se configuró de modo que la red de servicio compartido se definió como una instancia de VRF (nombre de VRF = servidor). Ahora, una visualización de la tabla CEF en el PE de ingreso muestra esto:

```
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#
```

Nota: Observe cómo el valor de etiqueta *24* se impone tanto para el VRF custA como para el VRF custB.

Esta visualización muestra la tabla de ruteo para la instancia de VRF de servicio compartido "sserver":

```
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
```

Nota: Solamente una ruta está presente para la red de destino desde la perspectiva del router PE de salida (**iguana**).

Por lo tanto, no se pudo distinguir el tráfico de varias VPN de clientes y el tráfico de retorno no pudo alcanzar la VPN adecuada. **En el caso en que el servicio compartido se debe definir como una instancia VRF, la función NAT se debe mover al PE de ingreso.**

NAT PE de entrada

En este ejemplo, los routers de borde del proveedor marcados con **gila** y **dragon** están configurados para NAT. Se define un conjunto NAT para cada VPN de cliente conectada que necesita acceso al servicio compartido. El conjunto apropiado se utiliza para cada una de las direcciones de red del cliente que son NATed. La NAT se realiza solamente en los paquetes destinados al host de servicio compartido en 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
```

Nota: En este escenario, no se admiten grupos compartidos. Si la LAN de servicio compartido (en el PE de salida) está conectada a través de una interfaz genérica, el conjunto NAT puede ser compartido.

Un ping originado en una dirección duplicada (172.31.1.1) dentro de cada una de las redes conectadas a **neuse** y **capefear8** da como resultado estas 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
```

Nota: La misma dirección local interna (172.31.1.1) se traduce a cada uno de los conjuntos definidos según el VRF de origen. El VRF se puede ver en el 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 visualizaciones muestran la información de ruteo para cada una de las VPN conectadas localmente para el cliente A y el 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
```

Nota: Desde la configuración estática se ha agregado una ruta para cada uno de los conjuntos NAT. Estas subredes se importan posteriormente en el VRF del servidor compartido en la **iguana** del router PE de salida:

```
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
```

Configuraciones

Se ha eliminado cierta información superflua de las configuraciones para su brevedad.

```
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
!
```

Nota: Las interfaces que se enfrentan a las redes del cliente se designan como interfaces "internas" NAT y las interfaces MPLS se designan como interfaces "externas" 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
```

El **dragón** del router tendría una configuración muy similar a **gila**.

Paquetes que llegan a PE central después de NAT PE de entrada

Los seguimientos a continuación ilustran el requisito de conjuntos NAT únicos cuando la red de servicio compartido de destino se configura como una instancia VRF. Una vez más, consulte el diagrama de la Figura 5. Los paquetes que se muestran a continuación fueron capturados cuando ingresaron a la interfaz IP MPLS e1/0/5 en la **iguana** del router.

Eco del cliente A VPN

Aquí, vemos una solicitud de eco proveniente de la dirección IP de origen 172.31.1.1 en VRF custA. La dirección de origen se ha traducido a 192.168.1.1 según lo especificado por la configuración 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 desde VPN del cliente B

Aquí, vemos una solicitud de eco proveniente de la dirección IP de origen 172.31.1.1 en VRF custB. La dirección de origen se ha traducido a 192.168.2.1 según lo especificado por la configuración 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".]
```

Nota: El valor de la etiqueta MPLS es *0019* en ambos paquetes mostrados arriba.

Respuesta de eco al cliente A VPN

A continuación, vemos una respuesta de eco que vuelve a la dirección IP de destino 192.168.1.1 en VRF custA. La dirección de destino es traducida a 172.31.1.1 por la función NAT de PE de ingreso.

```
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:
```

Respuesta de eco a VPN del cliente B

Aquí, vemos una respuesta de eco que vuelve a la dirección IP de destino 192.168.1.1 en VRF custB. La dirección de destino es traducida a 172.31.1.1 por la función NAT de PE de ingreso.

```
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".]
```

Nota: En los paquetes devueltos, los valores de etiqueta MPLS se incluyen y difieren: *001A* para VRF custA y *001D* para VRF custB.

Eco del cliente A VPN - El destino es una interfaz genérica

Este siguiente conjunto de paquetes muestra la diferencia cuando la interfaz a la LAN de servicio compartido es una interfaz genérica y no parte de una instancia VRF. Aquí, la configuración se ha cambiado para utilizar un conjunto común para ambas VPN locales con direcciones IP superpuestas.

```
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 desde VPN cliente B: el destino es una interfaz genérica

Aquí, vemos una solicitud de eco proveniente de la dirección IP de origen 172.31.1.1 en VRF custB. La dirección de origen se tradujo a 192.168.1.3 (del conjunto común SSPOOL1) según lo especificado por la configuración 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".]
```

Nota: Cuando la interfaz en el PE de salida es una interfaz genérica (no una instancia VRF), las etiquetas impuestas son diferentes. En este caso, *0x19* y *0x1F*.

Respuesta de eco al cliente A VPN - El destino es una interfaz genérica

A continuación, vemos una respuesta de eco que vuelve a la dirección IP de destino 192.168.1.1 en VRF custA. La dirección de destino es traducida a 172.31.1.1 por la función NAT de PE de ingreso.

```
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".]
```

Respuesta de eco a VPN del cliente B: el destino es una interfaz genérica

Aquí, vemos una respuesta de eco que vuelve a la dirección IP de destino 192.168.1.3 en VRF

custB. La dirección de destino es traducida a 172.31.1.1 por la función NAT de PE de ingreso.

```
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".]
```

Nota: Dado que las respuestas están destinadas a una dirección global, no se imponen etiquetas VRF.

Con la interfaz de salida al segmento de LAN de servicio compartido definido como una interfaz genérica, se permite un conjunto común. Los pings resultan en estas entradas NAT en el router **gila**:

```
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
```

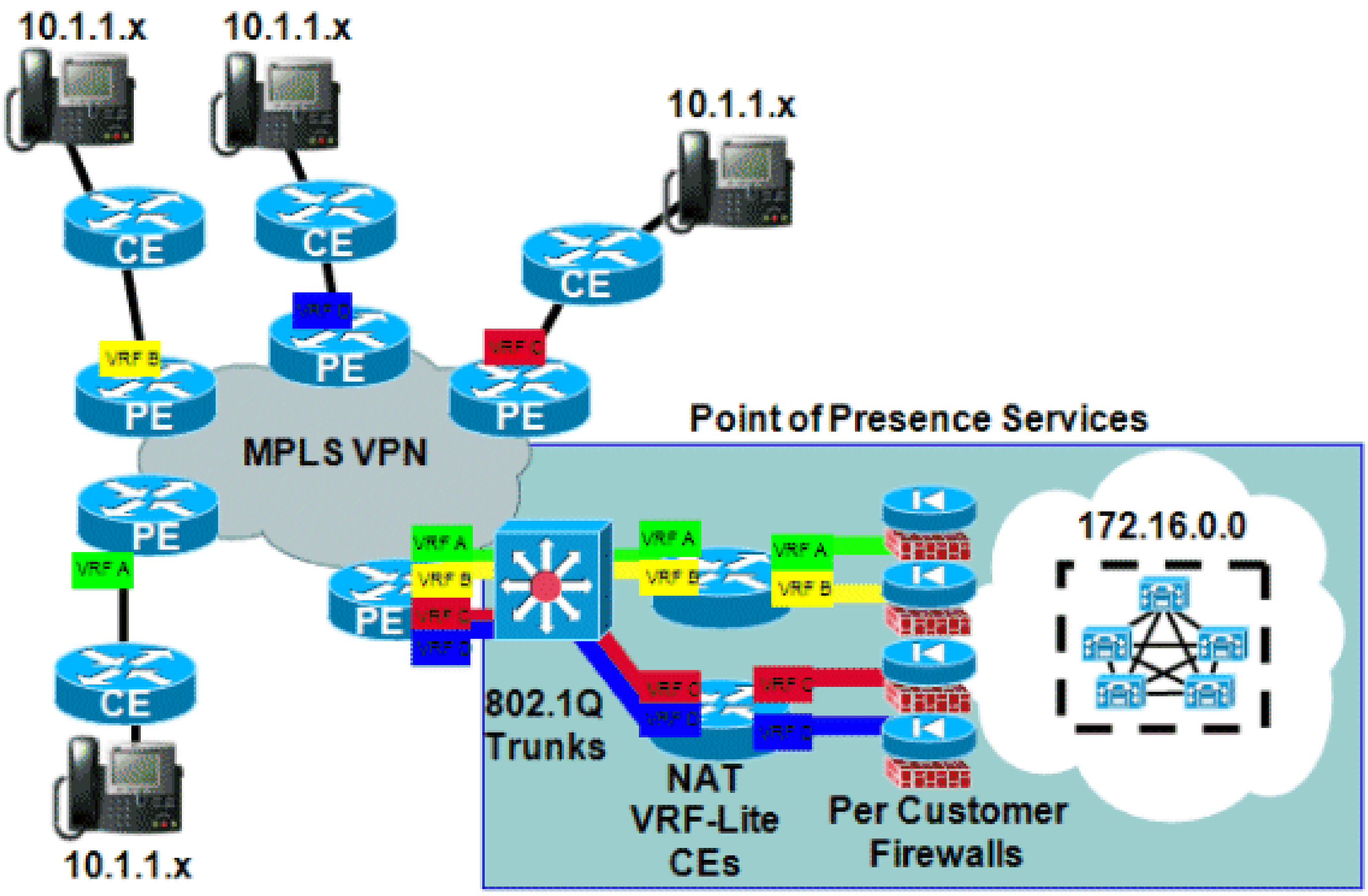
```
.Jan  2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan  2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan  2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan  2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan  2 09:35:12 EST: NAT-ip2tag: 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#
```

Ejemplo de servicio

En la Figura 8 se muestra un ejemplo de un servicio PBX IP virtual compartido. Esto ilustra una variante de los ejemplos de ingreso y egreso descritos anteriormente.

En este diseño, el servicio VoIP compartido es de extremo frontal por un conjunto de routers que realizan la función NAT. Estos routers tienen varias interfaces VRF usando una función conocida como VRF-Lite. El tráfico entonces fluye al clúster compartido de Cisco CallManager. Los servicios de firewall también se proporcionan por empresa. Las llamadas entre empresas deben pasar a través del firewall, mientras que las llamadas dentro de la empresa se gestionan a través de la VPN del cliente mediante el esquema de direccionamiento interno de la empresa.

Figura 8: Ejemplo de Servicio PBX Virtual Administrado



Disponibilidad

El soporte NAT de Cisco IOS para VPNs MPLS está disponible en la versión 12.2(13)T del IOS de Cisco y está disponible para todas las plataformas que soportan MPLS y pueden ejecutar esta versión de implementación temprana.

Conclusión

La NAT de Cisco IOS cuenta con funciones que permiten la implementación escalable de servicios compartidos en la actualidad. Cisco continúa desarrollando compatibilidad con el gateway de nivel de aplicación (ALG) NAT para protocolos importantes para los clientes. Las mejoras en el rendimiento y la aceleración del hardware para las funciones de traducción garantizarán que NAT y ALG proporcionen soluciones aceptables durante algún tiempo. Cisco supervisa todas las actividades relacionadas con estándares y las acciones de la comunidad. A medida que se desarrollen otros estándares, su uso se evaluará en función de los deseos, los requisitos y la aplicación del cliente.

Información Relacionada

- Gateways de capa de aplicación NAT de Cisco IOS
- Arquitecturas MPLS y VPN
- Diseño e implementación de MPLS avanzados
- Soporte Técnico y Documentación - Cisco Systems