

Cisco IOS NAT - integratie met MPLS VPN

Inhoud

[Inleiding](#)

[Voordelen van NAT - MPLS-integratie](#)

[Ontwerpoverwegingen](#)

[Plaatsingsscenario's](#)

[Installatieopties en Configuratiegegevens](#)

[Egress PE NAT](#)

[Ingress PE NAT](#)

[Pakketten die aankomen op Central PE na Ingress PE NAT](#)

[Servicevoorbeeld](#)

[Beschikbaarheid](#)

[Conclusie](#)

[Gerelateerde informatie](#)

[Inleiding](#)

Software voor Cisco IOS[®] Network Address Translation (NAT) maakt toegang tot gedeelde services van meerdere MPLS VPN's mogelijk, zelfs wanneer de apparaten in VPN's IP-adressen gebruiken die elkaar overlappen. Cisco IOS NAT is VRF-bewust en kan worden geconfigureerd op providerrandrouters binnen het MPLS-netwerk.

Opmerking: MPLS in IOS wordt alleen ondersteund door NAT. Op dit moment is er geen ondersteuning in Cisco IOS voor NAT NVI met MPLS.

De inzet van MPLS VPN's zal naar verwachting de komende jaren snel toenemen. De voordelen van een gemeenschappelijke netwerkinfrastructuur die snelle expansie en flexibele aansluitingsopties toestaat zullen ongetwijfeld een verdere groei van de diensten aansturen die aan de gemeenschap Internetwork kunnen worden aangeboden.

Er blijven echter nog steeds belemmeringen voor groei bestaan. IPv6 en zijn belofte van een IP-adresruimte die de aansluitingsbehoeften voor de nabije toekomst overtreft, bevinden zich nog in de vroege stadia van de implementatie. Bestaande netwerken gebruiken doorgaans particuliere IP-adresseringsschema's zoals gedefinieerd in [RFC 1918](#). Netwerkadresomzetting wordt vaak gebruikt om netwerken onderling te verbinden wanneer er sprake is van overlapping of overlapping van adressen.

Serviceproviders en bedrijven die netwerktoepassingservices hebben die zij willen aanbieden of delen met klanten en partners, willen alle aansluitingslasten voor de gebruiker van de service tot een minimum beperken. Het is wenselijk, zelfs verplicht, het aanbod uit te breiden tot zoveel potentiële gebruikers als nodig is om de gewenste doelstellingen of rendementen te bereiken. Het in gebruik zijnde IP-adresseringsschema mag geen belemmering zijn die potentiële gebruikers uitsluit.

Door Cisco IOS NAT binnen de gemeenschappelijke infrastructuur van MPLS VPN te implementeren kunnen communicatieservices (MPLS) een deel van de connectiviteit-last voor klanten verlichten en hun vermogen versnellen om meer gedeelde toepassingservices te verbinden met meer consumenten van die services.

Voordelen van NAT - MPLS-integratie

NAT-integratie met MPLS heeft voordelen voor zowel serviceproviders als hun zakelijke klanten. Het biedt dienstverleners meer mogelijkheden om gedeelde diensten in te zetten en toegang tot deze diensten te verlenen. Aanvullende dienstenaanbiedingen kunnen een differentiator zijn ten opzichte van concurrenten.

Voor serviceproviders	Voor VPN
Meer servicesproducten	Gereduceerde kosten
Verhoogde toegangsopties	Eenvoudige toegang
Verhoogde inkomsten	Aanpak van flexibiliteit

Bedrijven die een deel van hun huidige werklast willen uitbesteden, kunnen ook profiteren van het grotere aanbod van serviceproviders. De overheveling van de last van het uitvoeren van de noodzakelijke adresomzetting naar het netwerk van dienstverleners ontslaat hen van een ingewikkelde administratieve taak. Klanten kunnen privé-adressering blijven gebruiken, maar toch toegang tot gedeelde services en het internet behouden. Het consolideren van de NAT-functie binnen het netwerk van serviceproviders kan ook leiden tot lagere totale kosten voor zakelijke klanten, aangezien de klant scherpste routers de NAT-functie niet hoeven uit te voeren.

Ontwerpoverwegingen

Bij het overwegen van ontwerpen die NAT binnen het MPLS-netwerk zullen invoeren, is de eerste stap het bepalen van de servicebehoefte vanuit het oogpunt van toepassingen. U dient de gebruikte protocollen en eventuele speciale client/server communicatie die door de toepassing wordt opgelegd in overweging te nemen. Zorg ervoor dat de gewenste ondersteuning voor de gebruikte protocollen wordt ondersteund en verwerkt door Cisco IOS NAT. Een lijst met ondersteunde protocollen wordt geleverd in het document [Cisco IOS NAT-toepassingsgateways](#).

Daarna zal het nodig zijn om het verwachte gebruik van de gedeelde dienst en de verwachte verkeerssnelheid in pakketten-per-seconde te bepalen. NAT is een router CPU-intensieve functie. Daarom zullen prestatie-eisen een factor zijn bij het selecteren van een bepaalde inzetoptie en bij het bepalen van het aantal betrokken NAT-apparatuur.

Overweeg ook alle veiligheidskwesties en voorzorgsmaatregelen die moeten worden genomen. Hoewel MPLS VPN's per definitie particulier en effectief afzonderlijk verkeer zijn, is het gedeelde servicenetwerk over het algemeen gemeenschappelijk voor veel VPN's.

Plaatsingsscenario's

Er zijn twee opties voor NAT-implementatie binnen de MPLS-providerrand:

- Gecentraliseerd met strik-NAT PoE
- Gedistribueerd met ingang van NAT-eenheden

Enkele voordelen om de NAT-functie te configureren op het drukpunt van het MPLS-netwerk dat het dichtst bij het gedeelde servicenetwerk ligt, zijn:

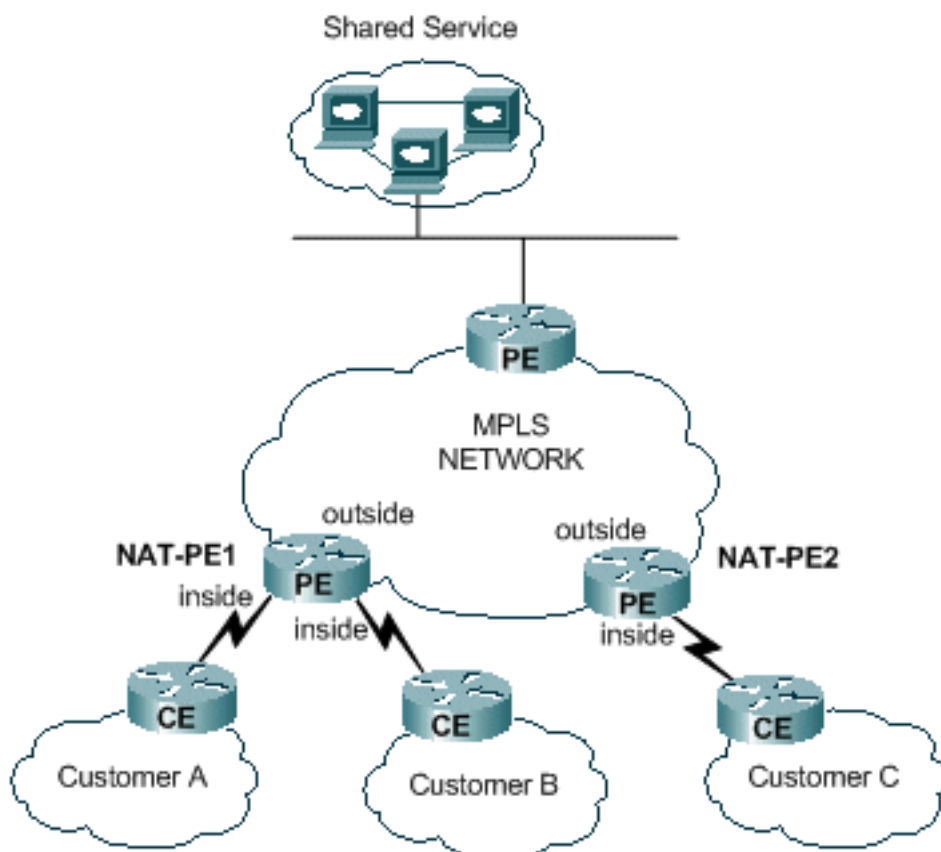
- Een gecentraliseerde configuratie die eenvoudiger serviceproviders bevordert
- Vereenvoudigde probleemoplossing
- Uitgebreide operationele schaalbaarheid
- Verlaagde IP-adrestoewijzingsvereisten

De voordelen worden echter gecompenseerd door een vermindering van schaalbaarheid en prestaties. Dit is de belangrijkste wisselwerking die in overweging moet worden genomen. Natuurlijk kan de NAT-functie ook binnen de klantennetwerken worden uitgevoerd als wordt bepaald dat de integratie van deze functie met een MPLS-netwerk niet wenselijk is.

Ingress PE NAT

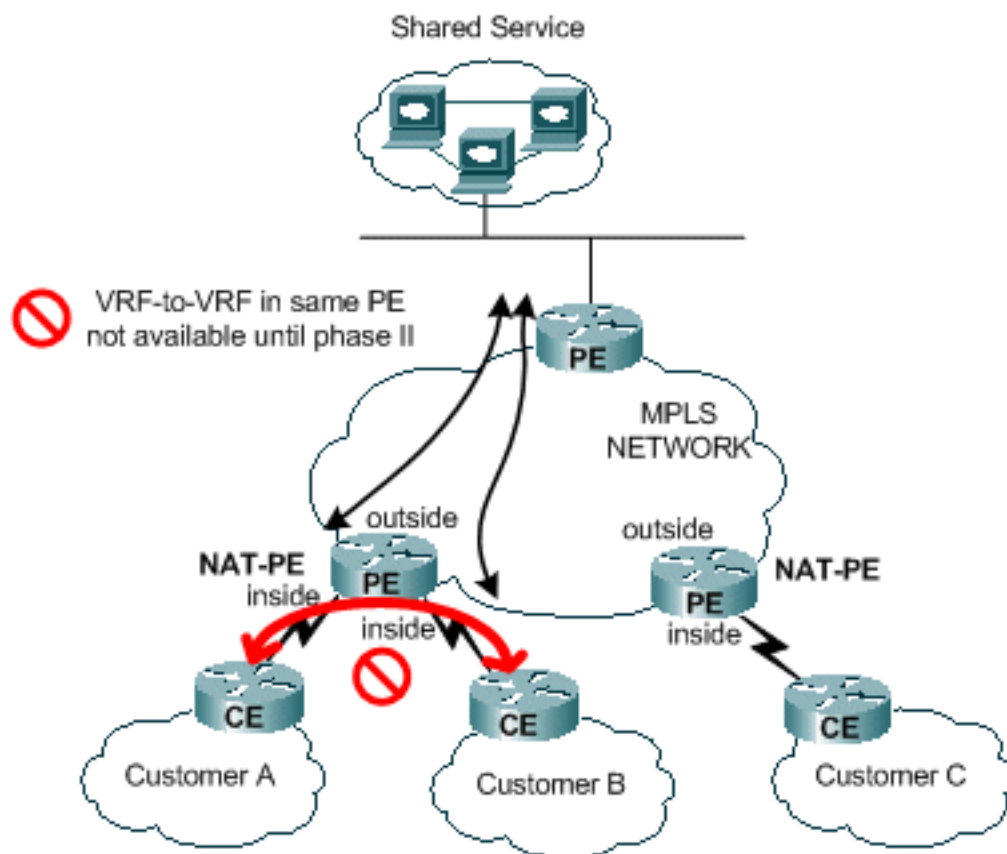
NAT kan worden geconfigureerd op de MPLS-router voor netwerktoegang zoals in [afbeelding 1](#). Bij dit ontwerp wordt de schaalbaarheid in grote mate gehandhaafd terwijl de prestaties worden geoptimaliseerd door de NAT-functie te distribueren op vele randapparaten. Elke NAT PE behandelt het verkeer voor sites die lokaal zijn verbonden met die PE. NAT-regels en toegangscontrolelijsten of routekaartecontrole die pakketten vertalen vereisen.

Afbeelding 1: Ingress PE NAT



Er is een beperking die NAT tussen twee VRF's voorkomt terwijl NAT ook wordt geleverd aan een gedeelde service zoals in [afbeelding 2](#). Dit is te wijten aan de eis om interfaces aan te wijzen als NAT-interfaces "binnen" en "buiten". Ondersteuning van verbindingen tussen VRF's in één PE is gepland voor een toekomstige Cisco IOS-release.

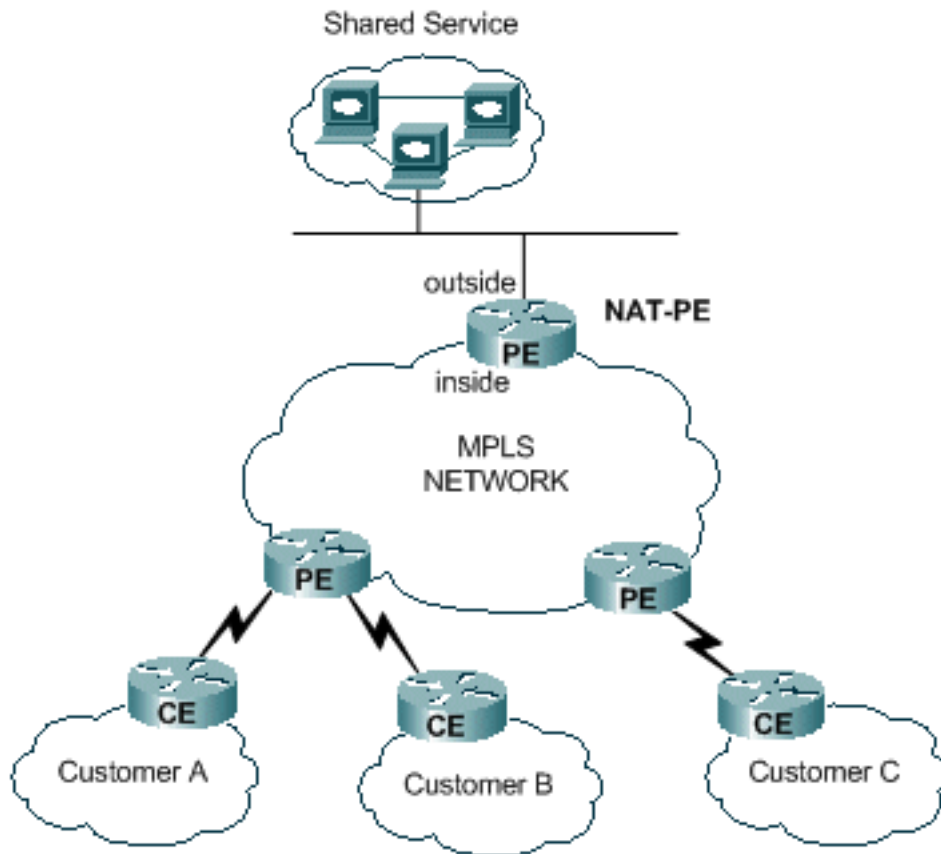
Afbeelding 2: Business to Business



Egress PE NAT

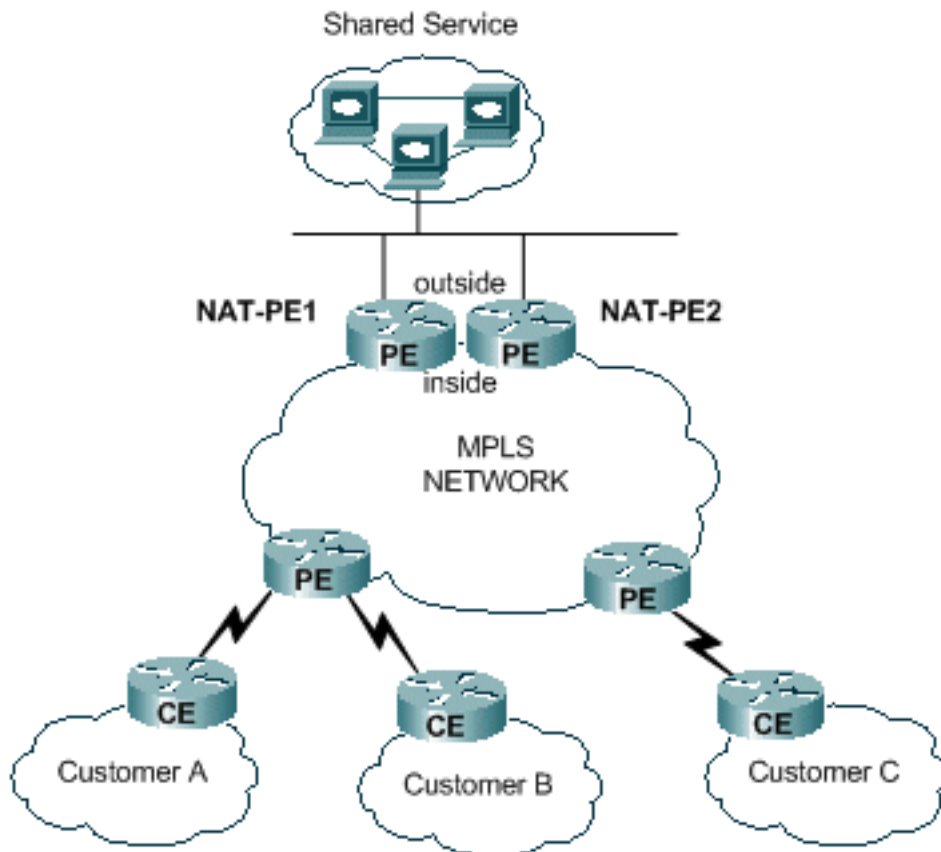
NAT kan worden geconfigureerd op de MPEG-router van het MPLS-netwerk zoals in [afbeelding 3](#). Met dit ontwerp wordt de schaalbaarheid tot op zekere hoogte verminderd omdat de centrale PE routes moet behouden voor alle klantnetwerken die toegang hebben tot de gedeelde service. De vereisten voor toepassingsprestaties moeten ook in aanmerking worden genomen zodat het verkeer de router niet overbelast die de IP-adressen van de pakketten moet vertalen. Omdat NAT voor alle klanten die dit pad gebruiken centraal optreedt, kunnen IP-adresgroepen worden gedeeld; het totale aantal vereiste subnetten wordt dus verminderd .

Afbeelding 3: Egress PE NAT



Er kunnen meerdere routers worden ingezet om de schaalbaarheid van het IP-ontwerp van [NAT](#) te verhogen zoals in [afbeelding 4](#) wordt getoond. In dit scenario kunnen klant VPN's "provisioning" worden uitgevoerd op een specifieke NAT-router. De vertaling van het netwerkadres zou voor het totale verkeer van en naar de gedeelde dienst voor die reeks VPN's plaatsvinden. Bijvoorbeeld, kon het verkeer van VPN's voor Customer A en B NAT-PE1 gebruiken, terwijl het verkeer naar en van VPN voor klant C NAT-PE2 gebruikt. Elke NAT PE zou alleen verkeer voor de specifieke VPN's die gedefinieerd zijn en alleen routes terug naar de sites in die VPN's onderhouden. Binnen elk van de NAT-routers kunnen afzonderlijke NAT-adresgroepen worden gedefinieerd, zodat pakketten worden verzonden van het gedeelde servicenetwerk naar de juiste NAT-applicatie voor vertaling en routing terug naar de klant VPN.

Afbeelding 4: Meervoudige uitgaande PE NAT



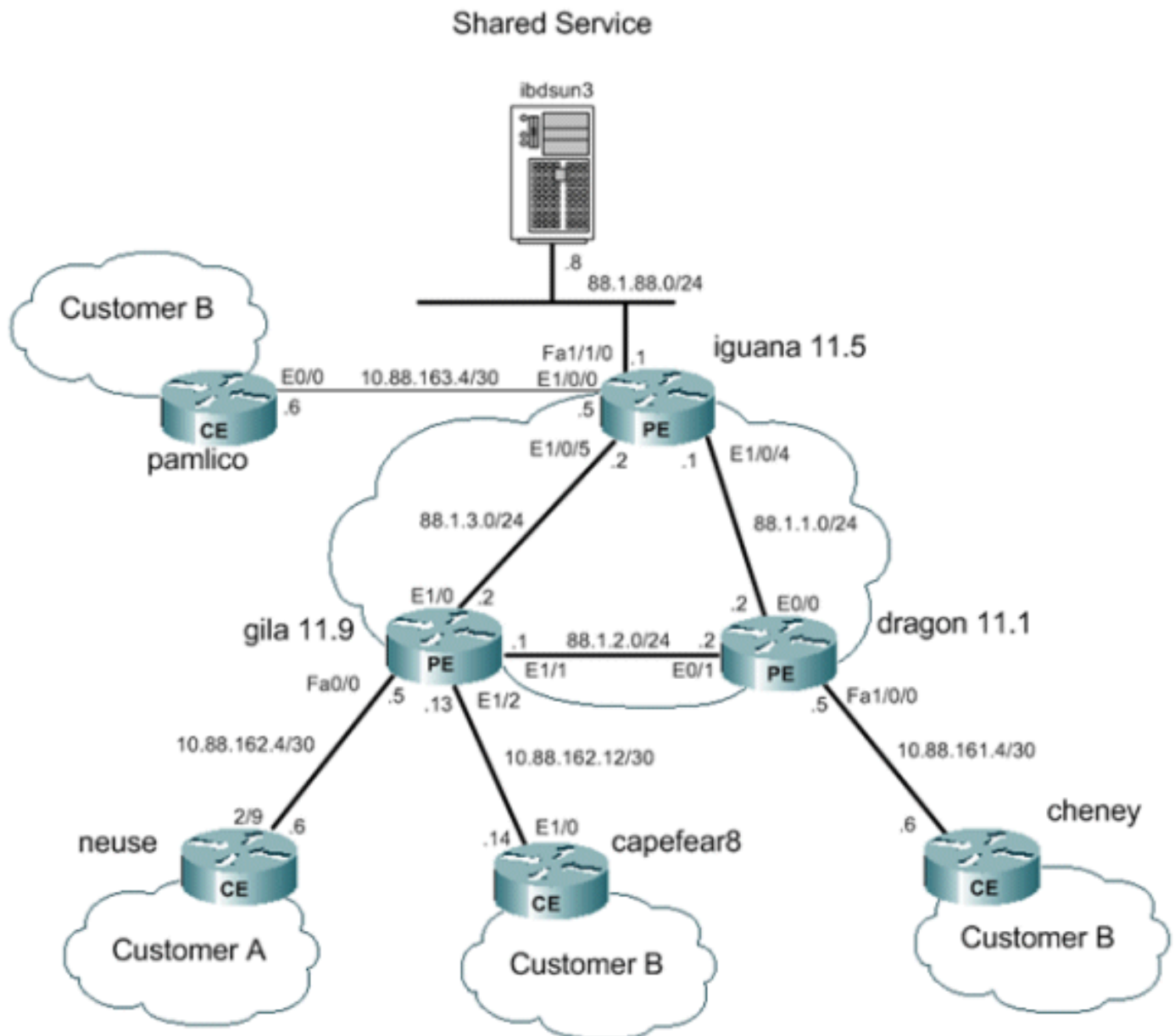
Het gecentraliseerde ontwerp legt een beperking op aan de manier waarop het gedeelde servicenetwerk moet worden geconfigureerd. Met name is het gebruik van de invoer/uitvoer van MPLS VPN-routes tussen een gedeelde service VPN en VPN's van klanten niet mogelijk. Dit is te wijten aan de aard van de MPLS-operatie zoals gespecificeerd door [RFC 2547](#). Wanneer de routes worden ingevoerd en uitgevoerd die de uitgebreide gemeenschappen en de routebeschrijvers gebruiken, kan NAT de bron VPN van het pakket bepalen dat in centrale NAT PE komt. Het gebruikelijke geval is om van het gedeelde servicenetwerk een generieke interface in plaats van een VRF-interface te maken. Een route naar het gedeelde servicenetwerk wordt dan toegevoegd in de centrale NAT-pagina voor elke VRF-tabel die is gekoppeld aan een klant VPN die toegang tot de gedeelde service nodig heeft als onderdeel van het provisioningproces. Dit wordt later nader beschreven.

[Installatieopties en Configuratiegegevens](#)

Deze sectie bevat enkele details over elk van de inzetopties. De voorbeelden zijn allemaal genomen van het netwerk dat in [afbeelding 5](#) is getoond. Raadpleeg dit schema voor de rest van deze sectie.

Opmerking: in het netwerk dat wordt gebruikt om de werking van VRF NAT voor dit document te illustreren, zijn alleen PE-routers opgenomen. Er zijn geen kern "P" routers. De essentiële mechanismen zijn echter nog steeds zichtbaar.

Afbeelding 5: VRF NAT-configuratievoorbeeld



Egress PE NAT

In dit voorbeeld, worden de van de verstrekker rand routers gemarkeerd met een **hoofddoek** en **draak** ingesteld als eenvoudige PE routers. De centrale PE in de buurt van het gedeelde serviceLAN (iguana) is geconfigureerd voor NAT. Een enkele NAT-pool wordt gedeeld door elke klant VPN die toegang tot de gedeelde service nodig heeft. NAT wordt alleen uitgevoerd op pakketten die bestemd zijn voor de gedeelde serviceshost op 8.1.8.8.

Druk op PE NAT-gegevensdoorsturen

Met MPLS gaat elk pakket het netwerk in op een IP-ingang en verlaat het MPLS-netwerk op een PE-uitgang. Het pad van labelswitchrouters die van ingang tot uitgang worden overgezet, is bekend als het label switched pad (LSP). De LSP is in één richting. Een andere LSP wordt gebruikt voor retourverkeer.

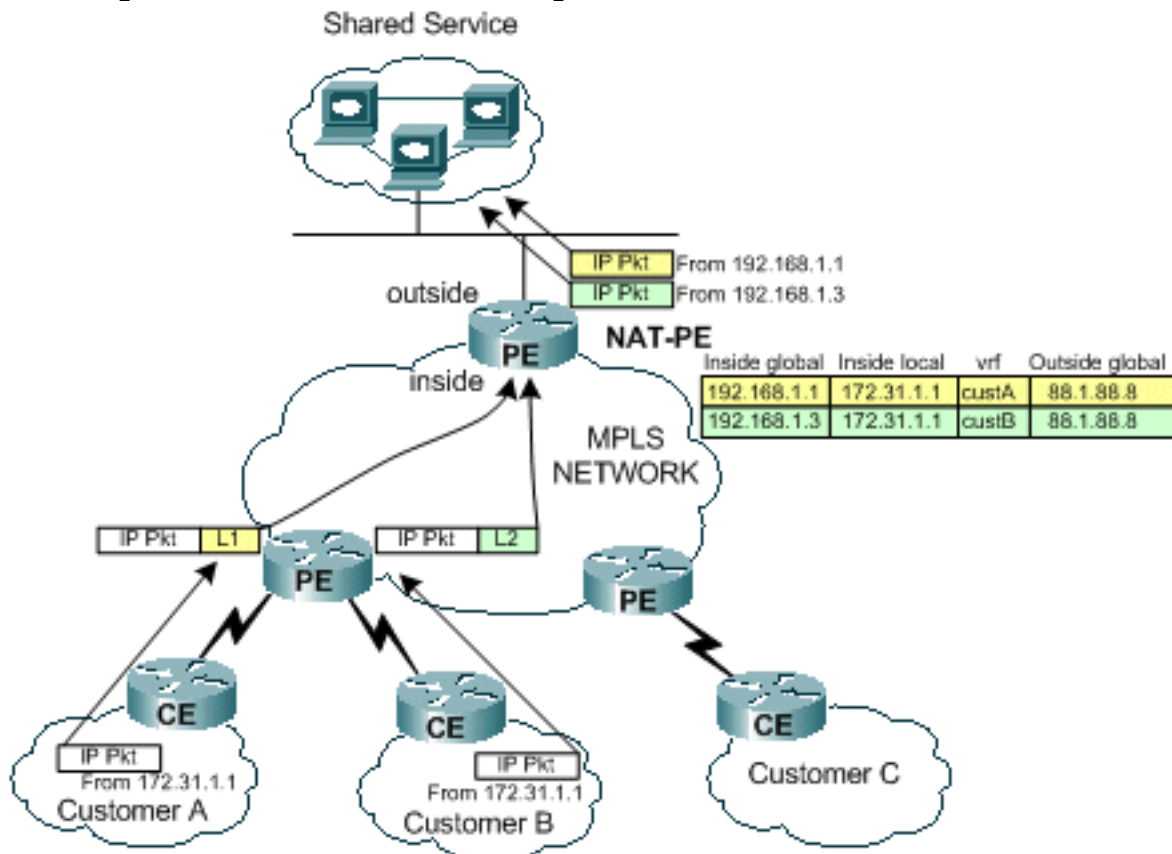
Bij gebruik van egress PE NAT is een verzendende equivalentieklasse (FEC) effectief gedefinieerd voor al het verkeer van gebruikers van de gedeelde dienst. Met andere woorden, alle pakketten die bestemd zijn voor het gedeelde netwerk zijn leden van een gemeenschappelijk FEC. Een pakket wordt aan een bepaalde FEC slechts eenmaal toegewezen aan de ingangrand

van het netwerk en volgt de LSP aan de res PE. De FEC wordt in het gegevenspakket aangewezen door een specifiek label toe te voegen.

PacketFlow-naar-gedeelde service via VPN

Om apparaten in meerdere VPN's die overlappende adresschema's hebben toegang tot een gedeelde serviceshost, is NAT vereist. Wanneer NAT bij aanvang op PE wordt geconfigureerd, zullen de ingangen in de vertaaltabel van het netwerkadres een VRF-id bevatten om dubbele adressen te differentiëren en een juiste routing te verzekeren.

Afbeelding 6: Packet verzonden naar uitgaande PE-nAT



Afbeelding 6 illustreert pakketten die voor een gedeelde serviceshost van twee klant VPN's zijn bestemd die dubbele IP-adresseringsschema's hebben. Het getal is een pakket dat afkomstig is van Customer A en een bronadres heeft van 172.31.1.1 dat bestemd is voor een gedeelde server op 88.1.88.8. Een ander pakket van Customer B met hetzelfde bron-IP-adres wordt ook naar dezelfde gedeelde server verzonden. Wanneer de pakketten de PE router bereiken, wordt een laag 3 raadpleging voor het bestemming IP netwerk in de het door:sturen informatiebasis (FIB) gedaan.

De ingang van het FIB vertelt de router van PE om het verkeer naar de grotere PE te verzenden met gebruik van een etiketstapel. Het bodemetiket in de stapel wordt toegewezen door de router van het bestemming PE, in dit geval router **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#

```

We kunnen aan de hand van de weergave zien dat pakketten van VRF custA een waarde van 24 (0x18) hebben en dat pakketten van VRF custB een waarde van 28 (0x1C) hebben.

In dit geval, omdat er geen "P"-routers in ons netwerk zijn, wordt er geen extra tag opgelegd. Indien er kernrouters waren geweest, zou er een etiket op de buitenkant zijn opgelegd en zou het normale proces van het omwisselen van label hebben plaatsgevonden binnen het kernnetwerk totdat het pakket de bovengrens van PE bereikte.

Aangezien de router van Gila direct met het graafschap PE is verbonden, zien we dat de tag wordt geprikt voordat er ooit aan wordt toegevoegd:

```

gila#
show tag-switching forwarding-table

```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
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 tag      Outgoing tag or VC  Prefix or Tunnel Id  Bytes tag switched  Outgoing interface  Next Hop

```

```
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#
```

De volgende displays tonen echo-pakketten zoals ontvangen door de egress PE NAT-router (op interface E1/0/5 op 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".]
```

Deze pings resulteren in de volgende ingangen die in de NAT-tabel in de **leguana** van de router PE worden gemaakt. De specifieke ingangen die voor de hierboven weergegeven pakketten worden gemaakt, kunnen door hun ICMP-identificator worden aangepast.

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#

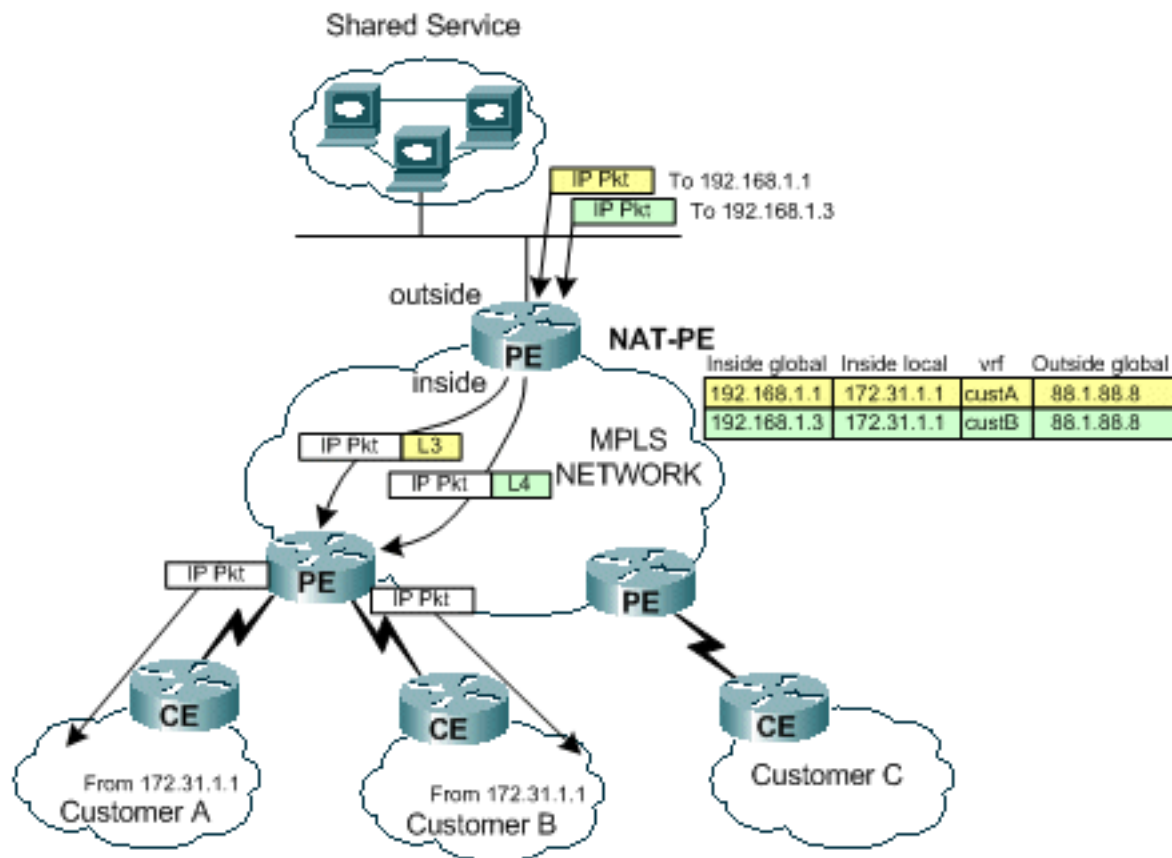
```

PacketFlow van gedeelde service terug naar Origineel VPN

Aangezien pakketten terugvloeien naar apparaten die de gedeelde serviceshost hebben

benaderd, wordt de NAT-tabel onderzocht voorafgaand aan het routing (pakketten die van NAT "buiten" interface naar "binnen" interface gaan). Omdat elke unieke ingang de corresponderende VRF herkenners omvat, kan het pakket correct worden vertaald en routeerd.

Afbeelding 7: Packet-over-naar-gedeelde servicegebruiker



Zoals in [afbeelding 7](#) is aangetoond, wordt het retourverkeer eerst door NAT onderzocht om een bijbehorende vertaling te vinden. Een pakket wordt bijvoorbeeld naar bestemming 192.168.1.1 verzonden. De NAT-tabel wordt doorzocht. Wanneer de match wordt gevonden, wordt de juiste vertaling uitgevoerd naar het "binnenste lokale" adres (172.31.1.1) en wordt er een nabijheidsraadpleging uitgevoerd met behulp van de gekoppelde VRF-ID uit de NAT-ingang.

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

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

```
tag rewrite with Et1/0/5, 88.1.3.1, tags imposed: {26}
iguana#
```

Label 23 (0x17) wordt gebruikt voor verkeer dat bestemd is voor 172.31.1.0/24 in VRF-codeA en label 26 (0x1A) wordt gebruikt voor pakketten die bestemd zijn voor 172.31.1.0/24 in VRF-codeB.

Dit wordt gezien in de antwoordpakketten van echo die van router **iguana** worden verzonden:

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".]
```

Wanneer het pakket de bestemming PE router bereikt, wordt het label gebruikt om de juiste VRF en de interface te bepalen om het pakket te verzenden.

gila#

show mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
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#

Configuraties

Sommige vreemde informatie is uit de configuraties verwijderd voor een beknoptheid.

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 vpv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.1 send-community extended
neighbor 88.1.11.9 activate
neighbor 88.1.11.9 send-community extended
no auto-summary
exit-address-family
!
address-family ipv4
neighbor 88.1.11.1 activate
neighbor 88.1.11.9 activate
neighbor 88.1.11.10 activate
no auto-summary
no synchronization
```



```
exit-address-family
!
address-family ipv4 vrf 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 vpv4
 neighbor 88.1.11.1 activate
 neighbor 88.1.11.1 send-community extended
 neighbor 88.1.11.5 activate
 neighbor 88.1.11.5 send-community extended
 no auto-summary
 exit-address-family
!
ip 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
!
```

Routerdraak zou een configuratie hebben die erg lijkt op **gila**.

Niet-toegestane invoer/uitvoer van routedoelstellingen

Wanneer het gedeelde servicenetwerk als een VRF-instantie zelf is geconfigureerd, is het centrale NAT in het IP-adres niet mogelijk. Dit is omdat de binnenkomende pakketten niet kunnen worden onderscheiden en slechts één route terug naar het oorsprong subnet is aanwezig bij het druk PE NAT.

Opmerking: de displays die volgen worden bedoeld om het resultaat van een ongeldige configuratie te illustreren.

Het voorbeeldnetwerk werd geconfigureerd zodat het gedeelde servicenetwerk werd gedefinieerd als een VRF-instantie (VRF-naam = server). Uit de CEF-tafel op de ingang van PE blijkt het volgende:

```
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 vrf tags 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#
```

Opmerking: Merk op hoe de tagwaarde 24 is opgelegd voor zowel VRF-ustA als VRF-kabelB.

Deze weergave toont de routingtabel voor de gedeelde dienst VRF-instantie "server":

```
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

```

Opmerking: Er is slechts één route aanwezig voor het bestemmingsnetwerk vanuit het perspectief van de IP-router (*iguana*).

Daarom kon geen onderscheid worden gemaakt tussen verkeer van meerdere klanten VPN's en retourverkeer kon niet het juiste VPN bereiken. **In het geval dat de gedeelde service moet worden gedefinieerd als een VRF-voorbeeld, moet de NAT-functie worden verplaatst naar de inkomende PE.**

Ingress PE NAT

In dit voorbeeld worden de **gemarkeerde** randrouters van de provider van randrouters en **draak** ingesteld voor NAT. Een NAT-pool wordt gedefinieerd voor elke aangesloten klant VPN die toegang tot de gedeelde service nodig heeft. Het juiste pool wordt gebruikt voor elk van de NATed-netwerkadressen van de klant. NAT wordt alleen uitgevoerd op pakketten die bestemd zijn voor de gedeelde serviceshost op 8.1.8.8.

```

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

```

Opmerking: in dit scenario worden gedeelde pools niet ondersteund. Als het gedeelde LAN (bij het uitgang PE) door een generieke interface wordt aangesloten, kan de NAT-pool worden gedeeld.

Een ping is afkomstig van een duplicaat adres (172.31.1.1) binnen elk van de netwerken verbonden aan **neuse** en **capefear8** resultaten in deze NAT-items:

Van **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

Opmerking: Hetzelfde lokale adres (172.31.1.1) wordt volgens de bron VRF vertaald naar elk van de gedefinieerde pools. Het VRF kan in het **bevel van het vertaalbreedband van de show ip** worden gezien:

```

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

```

Deze displays tonen de routinginformatie voor elk van de lokaal aangesloten VPN's voor klant A en klant 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

```

Opmerking: er is een route voor elk van de NAT-pools toegevoegd vanuit de statische configuratie. Deze subnetten worden vervolgens geïmporteerd in de gedeelde server VRF op de leguana van de IP-uitgang:

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
```

Configuraties

Sommige vreemde informatie is uit de configuraties verwijderd voor een beknoptheid.

GILA:

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

interface Loopback0
 ip address 88.1.11.9 255.255.255.255
!
```

```

interface FastEthernet0/0
  ip vrf forwarding custA
  ip address 10.88.162.5 255.255.255.252
  ip nat inside
  duplex full
!
interface Ethernet1/0
  ip address 88.1.3.1 255.255.255.0
  ip nat outside
  no ip mroute-cache
  duplex half
  tag-switching ip
!
interface Ethernet1/1
  ip address 88.1.2.1 255.255.255.0
  ip nat outside
  no ip mroute-cache
  duplex half
  tag-switching ip
!
interface Ethernet1/2
  ip vrf forwarding custB
  ip address 10.88.162.13 255.255.255.252
  ip nat inside
  duplex half
!
router ospf 881
  log-adjacency-changes
  redistribute static subnets
  network 88.1.0.0 0.0.255.255 area 0
  default-metric 30
!
router bgp 65002
  no synchronization
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 88.1.11.1 remote-as 65002
  neighbor 88.1.11.1 update-source Loopback0
  neighbor 88.1.11.1 activate
  neighbor 88.1.11.5 remote-as 65002
  neighbor 88.1.11.5 update-source Loopback0
  neighbor 88.1.11.5 activate
  no auto-summary
!
address-family ipv4 vrf custB
  redistribute connected
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
address-family ipv4 vrf custA
  redistribute connected
  redistribute static
  no auto-summary
  no synchronization
  exit-address-family
!
address-family vpv4
  neighbor 88.1.11.1 activate
  neighbor 88.1.11.1 send-community extended
  neighbor 88.1.11.5 activate
  neighbor 88.1.11.5 send-community extended
  no auto-summary

```



```

exit-address-family
!
ip nat pool SSPOOL1 192.168.1.1 192.168.1.254 prefix-length 24
ip nat pool SSPOOL2 192.168.2.1 192.168.2.254 prefix-length 24
ip nat inside source list 181 pool SSPOOL1 vrf custA overload
ip nat inside source list 181 pool SSPOOL2 vrf custB overload
ip classless
ip route vrf custA 172.31.1.0 255.255.255.0 FastEthernet0/0 10.88.162.6
ip route vrf custA 192.168.1.0 255.255.255.0 Null0
ip route vrf custB 172.31.1.0 255.255.255.0 Ethernet1/2 10.88.162.14
ip route vrf custB 192.168.2.0 255.255.255.0 Null0
!
access-list 181 permit ip any host 88.1.88.8
!

```

Opmerking: de interfaces die op de clientnetwerken zijn gericht, worden aangeduid als NAT-interfaces en de MPLS-interfaces worden aangeduid als NAT-interfaces.

```

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

```

Routerdraak zou een configuratie hebben die erg lijkt op gila.

[Pakketten die aankomen op Central PE na Ingress PE NAT](#)

De onderstaande sporen illustreren het vereiste voor unieke NAT-pools wanneer het gedeelde servicenetwerk van de bestemming als een VRF-voorbeeld wordt geconfigureerd. Raadpleeg opnieuw het diagram in [afbeelding 5](#). De hieronder getoonde pakketten zijn opgenomen terwijl ze

in de MPLS IP interface e1/0/5 op router **iguana** werden ingevoerd.

Echo van Customer A VPN

Hier zien we een echo-verzoek dat van het bron-IP-adres 172.31.1.1 bij VRF-code komt. Het bronadres is vertaald naar 192.168.1.1 zoals gespecificeerd door de NAT-configuratie:

```
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:

Echo van Customer B VPN

Hier zien we een echo-verzoek dat van het bron-IP-adres 172.31.1.1 komt in VRF-code CustB. Het bronadres is vertaald naar 192.168.2.1 zoals gespecificeerd door de NAT-configuratie:

```
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".]
```

Opmerking: de waarde van het MPLS-label is *0019* in beide van de *hierboven* genoemde pakketten.

[echo-antwoord op VPN van klant](#)

Daarna zien we een echo-antwoord dat teruggaat naar het IP-adres van het bestemming 192.168.1.1 in VRF-code CustA. Het bestemmingsadres wordt vertaald naar 172.31.1.1 door de ingangsfunctie PE NAT.

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:
```

ECHO-antwoord op VPN bij Customer B

Hier zien we een echo-antwoord dat teruggaat naar het IP-adres van het bestemming 192.168.1.1 bij VRF CustB. Het bestemmingsadres wordt vertaald naar 172.31.1.1 door de ingangsfunctie PE NAT.

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".]
```

Opmerking: Bij de retourpakketten zijn de MPLS-labelwaarden inbegrepen en verschillen ze van

elkaar: 001A voor VRF CustA en 001D voor VRF CustB.

ECHO van Customer A VPN - Destktop is a Generic Interface

Deze volgende set pakketten toont het verschil wanneer de interface naar het gedeelde service-LAN een generieke interface is en geen deel van een VRF-instantie. Hier is de configuratie gewijzigd in een gezamenlijke pool voor beide lokale VPN's met overlappende IP-adressen.

```
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".]
```

Echo van Customer B VPN - Destktop is a Generic Interface

Hier zien we een echo-verzoek dat van het bron-IP-adres 172.31.1.1 komt in VRF-code CustB. Het bronadres werd vertaald naar 192.168.1.3 (uit common pool SSPOOL1) zoals gespecificeerd door de NAT-configuratie:

```
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".]
```

Opmerking: Wanneer de interface bij het egress PE een generieke interface is (geen VRF-instantie), zijn de opgelegde labels anders. In dit geval, *0x19* en *0x1F*.

[Een echo-antwoord op een VPN-reactie van de klant - de bestemming is een generieke interface](#)

Daarna zien we een echo-antwoord dat teruggaat naar het IP-adres van het bestemming 192.168.1.1 in VRF-code CustA. Het bestemmingsadres wordt vertaald naar 172.31.1.1 door de ingangsfunctie PE NAT.

```
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".]
```

[ECHO reageren op VPN - bestemming is een generieke interface](#)

Hier zien we een echo-antwoord dat teruggaat naar het IP-adres van het bestemming 192.168.1.3 in VRF CustB. Het bestemmingsadres wordt vertaald naar 172.31.1.1 door de ingangsfunctie PE NAT.

```
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".]
```

Opmerking: aangezien de antwoorden bestemd zijn voor een algemeen adres, zijn er geen VRF-etiketten opgelegd.

Als de exit-interface naar het gedeelde service LAN-segment wordt gedefinieerd als een generieke interface, is een gemeenschappelijk pool toegestaan. De pings resulteren in deze NAT ingangen in 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
```

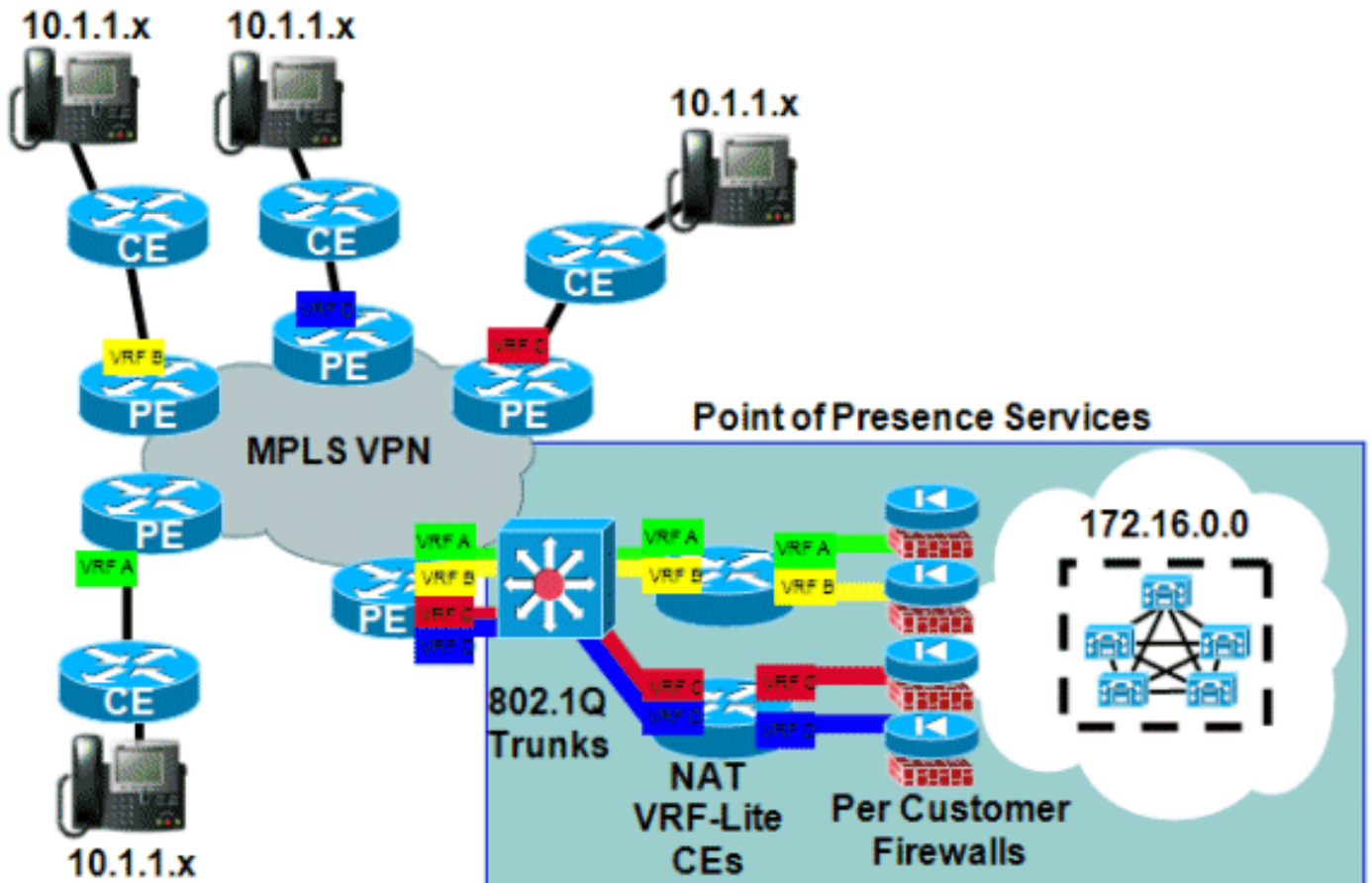
```
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting to process
.Jan 2 09:35:12 EST: NAT-ip2tag : Tag Pkt s=172.31.1.1, d=88.1.88.8, vrf=custA
.Jan 2 09:35:12 EST: NAT-ip2tag: Punting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

Servicevoorbeeld

Een voorbeeld van een gedeelde virtuele IP PBX-service wordt in [afbeelding 8](#) getoond. Dit is een variant van de eerder beschreven instap- en uitstapvoorbeelden.

In dit ontwerp wordt de gedeelde VoIP-service geleverd door een verzameling routers die de NAT-functie uitvoeren. Deze routers hebben meerdere VRF-interfaces die een functie gebruiken die bekend staat als VRF-Lite. Het verkeer stroomt vervolgens naar de gedeelde Cisco CallManager-cluster. De firewalldiensten worden ook per bedrijf verleend. Inter-firma gesprekken moeten door de firewall gaan, terwijl de binnen-firma-oproepen door de klant VPN worden behandeld met behulp van het interne adresseringsschema van het bedrijf.

Afbeelding 8: Beheerd Virtual PBX-servicevoorbeeld



Beschikbaarheid

Cisco IOS NAT-ondersteuning voor MPLS VPN's is beschikbaar in Cisco IOS release 12.2(13)T en is beschikbaar voor alle platforms die MPLS ondersteunen en deze vroege implementatielijnen kunnen uitvoeren.

Conclusie

Cisco IOS NAT heeft functies om schaalbare implementatie van gedeelde services vandaag mogelijk te maken. Cisco blijft ondersteuning voor NAT-toepassingsniveau (ALG) ontwikkelen voor protocollen die belangrijk zijn voor klanten. Prestatieverbeteringen en hardwareversnelling voor vertaalfuncties zullen ervoor zorgen dat NAT en ALG's in de komende tijd aanvaardbare oplossingen bieden. Alle relevante normalisatieactiviteiten en communautaire acties worden door Cisco gevolgd. Aangezien andere standaarden worden ontwikkeld, zal het gebruik ervan worden beoordeeld op basis van de wensen van de klant, vereisten en toepassing.

Gerelateerde informatie

- [Cisco IOS NAT-toepassingslaag-gateways](#)
- [MPLS- en VPN-architecturen](#)
- [Geavanceerde MPLS-ontwerp en -implementatie](#)
- [Technische ondersteuning en documentatie – Cisco Systems](#)