

Implementação do IOS do recurso iBGP PE-CE

Contents

[Introduction](#)

[Informações de Apoio](#)

[Implementar iBGP PE-CE](#)

[Atributo de rota do cliente BGP](#)

[Configurar](#)

[Novo comando](#)

[Análise detalhada de ATTR SET](#)

[Manuseio do próximo salto](#)

[RD](#)

[Recurso iBGP PE-CE com Local-AS](#)

[Regras para troca de rotas entre diferentes sites de VRF](#)

[Reflexão CE-to-CE VRF-Lite](#)

[Cisco IOS mais antigo no roteador PE](#)

[Próximo salto para eBGP em VRF](#)

Introduction

Este documento descreve como o recurso Internal Border Gateway Protocol (iBGP) entre Provider Edge (PE) e Customer Edge (CE) é implementado no Cisco IOS®.

Informações de Apoio

Até o novo recurso iBGP PE-CE, o iBGP entre PE e CE (portanto, em uma interface Virtual Routing and Forwarding (VRF) no roteador PE) não era oficialmente suportado. Uma exceção é o iBGP em interfaces VRF em uma configuração multi-VRF CE (VRF-Lite). A motivação para implantar esse recurso é:

- O cliente deseja ter um único número de sistema autônomo (ASN) em vários locais do VRF, sem a implantação do protocolo eBGP (External Border Gateway Protocol) com as-override.
- O cliente deseja fornecer reflexão de rota interna para os roteadores CE, atuando como se o núcleo do provedor de serviços (SP) fosse um refletor de rota (RR) transparente.

Com esse recurso, os locais do VRF podem ter o mesmo ASN que o núcleo do SP. No entanto, caso o ASN dos sites VRF seja diferente do ASN do núcleo da controladora de armazenamento, ele pode ser feito para parecer o mesmo com o uso do recurso Local-Autonomous System (AS).

Implementar iBGP PE-CE

Aqui estão as duas partes principais para que este recurso funcione:

- Um novo atributo ATTR_SET foi adicionado ao protocolo BGP para transportar os atributos do VPN BGP através do núcleo do SP de maneira transparente.
- Faça do roteador PE um RR para as sessões iBGP em direção aos roteadores CE no VRF e como um RR em direção aos vizinhos VPNv4 (outros roteadores PE ou RR).

O novo atributo ATTR_SET permite que o SP transporte todos os atributos de BGP do cliente de maneira transparente e não interfere com os atributos do SP e as políticas de BGP. Esses atributos são a lista de clusters, a preferência local, as comunidades e assim por diante.

Atributo de rota do cliente BGP

ATTR_SET é o novo atributo de BGP usado para transportar os atributos de BGP de VPN do cliente de SP. É um atributo transitivo opcional. Neste atributo, todos os atributos de BGP do cliente da mensagem de atualização de BGP, exceto os atributos MP_REACH e MP_UNREACH, podem ser transportados.

O atributo ATTR_SET tem este formato:

```
+-----+
| Attr Flags (O|T) Code = 128 |
+-----+
| Attr. Length (1 or 2 octets) |
+-----+
| Origin AS (4 octets)         |
+-----+
| Path Attributes (variable)   |
+-----+
```

Os sinalizadores de atributo são os sinalizadores de atributo BGP regulares (consulte RFC 4271). O comprimento do atributo indica se o comprimento do atributo é um ou dois octetos. A finalidade do campo Origem AS é evitar que o vazamento de uma rota originada em um AS vaze para outro AS sem a manipulação adequada do AS_PATH. O campo de atributos de caminho de comprimento variável transporta os atributos de VPN BGP que devem ser transportados através do núcleo do SP.

No roteador PE de saída, os atributos do VPN BGP são enviados para esse atributo. No roteador PE de entrada, esses atributos são exibidos do atributo, antes que o prefixo BGP seja enviado ao roteador CE. Este atributo fornece o isolamento dos atributos BGP entre a rede SP e a VPN do cliente e vice-versa. Por exemplo, o atributo de lista de cluster de reflexão de rota SP não é visto e considerado dentro da rede VPN. Mas também, o atributo de lista de cluster de reflexão de rota VPN não é visto e considerado dentro da rede SP.

Examine a Figura 1 para ver a propagação de um prefixo BGP de cliente através da rede SP.

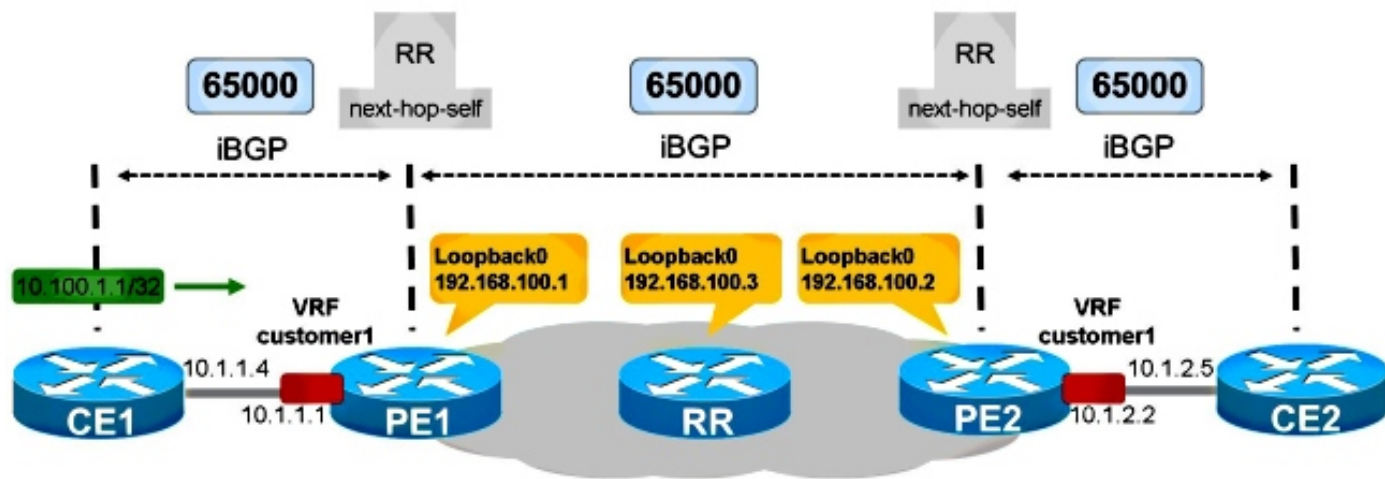


Figure 1

CE1 e CE2 estão no mesmo AS que a rede da controladora de armazenamento: 65000. O PE1 tem o iBGP configurado para CE1. O PE1 reflete o caminho para o prefixo 10.100.1.1/32 em direção ao RR na rede da controladora de armazenamento. O RR reflete o caminho do iBGP em direção aos roteadores PE como de costume. O PE2 reflete o caminho em direção ao CE2.

Para que isso funcione corretamente, você deve:

- Possuem código em PE1 e PE2 que tenha o suporte ao recurso iBGP PE-CE
- Configure PE1 e PE2 para executar a reflexão de rota em sua sessão BGP em direção aos respectivos roteadores CE
- Possuir o próximo salto nos roteadores PE para a sessão BGP em direção aos seus roteadores CE
- Certifique-se de que cada site VPN usa diferentes Distinguidores de Rota (RD - Route Distinguishers)

Configurar

Veja a Figura 1.

Esta é a configuração necessária para PE1 e PE2:

```
PE1

vrf definition customer1
rd 65000:1
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family

router bgp 65000
```

```

bgp log-neighbor-changes
neighbor 192.168.100.3 remote-as 65000
neighbor 192.168.100.3 update-source Loopback0
!
address-family vpnv4
neighbor 192.168.100.3 activate
neighbor 192.168.100.3 send-community extended
exit-address-family
!
address-family ipv4 vrf customer1
neighbor 10.1.1.4 remote-as 65000
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 internal-vpn-client
neighbor 10.1.1.4 route-reflector-client
neighbor 10.1.1.4 next-hop-self
exit-address-family

```

PE2

```

vrf definition customer1
rd 65000:2
route-target export 1:1
route-target import 1:1
!
address-family ipv4
exit-address-family

```

```

router bgp 65000
bgp log-neighbor-changes
neighbor 192.168.100.3 remote-as 65000
neighbor 192.168.100.3 update-source Loopback0
!
address-family vpnv4
neighbor 192.168.100.3 activate
neighbor 192.168.100.3 send-community extended
exit-address-family
!
address-family ipv4 vrf customer1
neighbor 10.1.2.5 remote-as 65000
neighbor 10.1.2.5 activate
neighbor 10.1.2.5 internal-vpn-client
neighbor 10.1.2.5 route-reflector-client
neighbor 10.1.2.5 next-hop-self
exit-address-family

```

Note: Se o PE não tiver o comando **neighbor <internal-CE> internal-vpn-client** para o vizinho CE, ele não propagará os prefixos do CE para os roteadores SP RR/PE.

Note: Se o PE não for o RR no VRF, ele não propaga os prefixos dos roteadores RR/PE para o roteador CE.

Novo comando

Há um novo comando, **neighbor <internal-CE> internal-vpn-client**, para fazer esse recurso funcionar. Ele deve ser configurado no roteador PE somente para a sessão iBGP em direção aos roteadores CE.

Note: O recurso iBGP PE-CE Multi-VRF CE (VRF-Lite) ainda é suportado sem o comando `neighbor <internal-CE> internal-vpn-client`.

Note: Quando o comando `neighbor <internal-CE> internal-vpn-client` é configurado, os comandos `neighbor <internal-CE> route-reflector-client` e `neighbor <internal-CE> next-hop-self` também são colocados automaticamente na configuração. Quando os comandos `neighbor <internal-CE> route-reflector-client` e `neighbor <internal-CE> next-hop-self` (ou ambos) são removidos e uma recarga é executada, eles são automaticamente colocados de volta na configuração.

Análise detalhada de ATTR_SET

Veja a Figura 1.

Este é o prefixo anunciado por CE1:

```
CE1#show bgp ipv4 unicast 10.100.1.1/32
BGP routing table entry for 10.100.1.1/32, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    4
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
      rx pathid: 0, tx pathid: 0x0
```

Quando o PE1 recebe o prefixo de BGP 10.100.1.1/32 do CE1, ele o armazena duas vezes:

```
PE1#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 21
Paths: (2 available, best #1, table customer1)
  Advertised to update-groups:
    5
  Refresh Epoch 1
  Local, (Received from ibgp-pece RR-client)
    10.1.1.4 (via vrf customer1) from 10.1.1.4 (10.100.1.1)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      mpls labels in/out 18/nolabel
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  Local, (Received from ibgp-pece RR-client), (ibgp sourced)
    10.1.1.4 (via vrf customer1) from 10.1.1.4 (10.100.1.1)
      Origin IGP, localpref 100, valid, internal
      Extended Community: RT:1:1
      mpls labels in/out 18/nolabel
      rx pathid: 0, tx pathid: 0
```

O primeiro caminho é o caminho real em PE1, porque é recebido de CE1.

O segundo caminho é o caminho anunciado em direção aos roteadores RRs/PE. Ele está marcado com **ibgp de origem**. Contém o atributo ATTR_SET. Observe que esse caminho tem um ou mais RTs (Route Targets, Destinos de Rota) conectados a ele.

O PE1 anuncia o prefixo como mostrado aqui:

```
PE1#show bgp vpnv4 unicast all neighbors 192.168.100.3 advertised-routes
BGP table version is 7, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65000:1 (default for vrf customer1)
*>i 10.100.1.1/32      10.1.1.4              0      200      0 i
```

Total number of prefixes 1

Éassim que o RR vê o caminho:

```
RR#show bgp vpnv4 un all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 10
Paths: (1 available, best #1, no table)
Advertised to update-groups:
  3
Refresh Epoch 1
Local, (Received from a RR-client)
  192.168.100.1 (metric 11) (via default) from 192.168.100.1 (192.168.100.1)
    Origin IGP, localpref 100, valid, internal, best
    Extended Community: RT:1:1
    Originator: 10.100.1.1, Cluster list: 192.168.100.1
ATTR_SET Attribute:
  Originator AS 65000
  Origin IGP
  Aspath
  Med 0
LocalPref 200
  Cluster list
  192.168.100.1,
  Originator 10.100.1.1
mpls labels in/out nolabel/18
rx pathid: 0, tx pathid: 0x0
```

Observe que a preferência local deste prefixo unicast VPNv4 no núcleo é 100. No ATTR_SET, a preferência local original de 200 é armazenada. No entanto, isso é transparente para o RR no núcleo do SP.

No PE2, você vê o prefixo como mostrado aqui:

```
PE2#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 5
Paths: (1 available, best #1, no table)
Not advertised to any peer
Refresh Epoch 2
Local
  192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
    Origin IGP, localpref 100, valid, internal, best
    Extended Community: RT:1:1
    Originator: 10.100.1.1, Cluster list: 192.168.100.3, 192.168.100.1
ATTR_SET Attribute:
  Originator AS 65000
  Origin IGP
  Aspath
  Med 0
```

```

LocalPref 200
Cluster list
192.168.100.1,
Originator 10.100.1.1
mpls labels in/out nolabel/18
rx pathid: 0, tx pathid: 0x0
BGP routing table entry for 65000:2:10.100.1.1/32, version 6
Paths: (1 available, best #1, table customer1)
Advertised to update-groups:
1
Refresh Epoch 2
Local, imported path from 65000:1:10.100.1.1/32 (global)
192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
Origin IGP, metric 0, localpref 200, valid, internal, best
Originator AS(ibgp-pece): 65000
Originator: 10.100.1.1, Cluster list: 192.168.100.1
mpls labels in/out nolabel/18
rx pathid:0, tx pathid: 0x0

```

O primeiro caminho é o recebido do RR, com o ATTR_SET. Observe que o RD é 65000:1, o RD de origem. O segundo caminho é o caminho importado da tabela VRF com RD 65000:1. O ATTR_SET foi removido.

Esse é o caminho conforme visto no CE2:

```

CE2#show bgp ipv4 unicast 10.100.1.1/32
BGP routing table entry for 10.100.1.1/32, version 10
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
Local
10.1.2.2 from 10.1.2.2 (192.168.100.2)
Origin IGP, metric 0, localpref 200, valid, internal, best
Originator: 10.100.1.1, Cluster list: 192.168.100.2, 192.168.100.1
rx pathid: 0, tx pathid: 0x0

```

Observe que o salto seguinte é **10.1.2.2**, que é PE2. A lista de clusters contém os roteadores PE1 e PE2. Esses são os RR que importam dentro da VPN. O RR do SP (10.100.1.3) não está na lista de clusters.

A preferência local de 200 foi preservada dentro da VPN através da rede SP.

O comando **debug bgp vpnv4 unicast updates** mostra a atualização propagada na rede SP:

```

PE1#
BGP(4): Revise route installing 1 of 1 routes for 10.100.1.1/32 -> 10.1.1.4
(customer1) to customer1 IP table
BGP(4): 192.168.100.3 NEXT_HOP changed SELF for ibgp rr-client pe-ce net
65000:1:10.100.1.1/32,
BGP(4): 192.168.100.3 Net 65000:1:10.100.1.1/32 from ibgp-pece 10.1.1.4 format
ATTR_SET
BGP(4): (base) 192.168.100.3 send UPDATE (format) 65000:1:10.100.1.1/32, next
192.168.100.1, label 16, metric 0, path Local, extended community RT:1:1
BGP: 192.168.100.3 Next hop is our own address 192.168.100.1
BGP: 192.168.100.3 Route Reflector cluster loop; Received cluster-id 192.168.100.1
BGP: 192.168.100.3 RR in same cluster. Reflected update dropped

RR#
BGP(4): 192.168.100.1 rcvd UPDATE w/ attr: nexthop 192.168.100.1, origin i, localpref
100, originator 10.100.1.1, clusterlist 192.168.100.1, extended community RT:1:1,

```

```
[ATTR_SET attribute: originator AS 65000, origin IGP, aspath , med 0, localpref 200, cluster list 192.168.100.1 , originator 10.100.1.1]
```

```
BGP(4): 192.168.100.1 rcvd 65000:1:10.100.1.1/32, label 16  
RT address family is not configured. Can't create RTC route  
BGP(4): (base) 192.168.100.1 send UPDATE (format) 65000:1:10.100.1.1/32, next  
192.168.100.1, label 16, metric 0, path Local, extended community RT:1:1
```

```
PE2#
```

```
BGP(4): 192.168.100.3 rcvd UPDATE w/ attr: nexthop 192.168.100.1, origin i, localpref  
100, originator 10.100.1.1, clusterlist 192.168.100.3 192.168.100.1, extended community  
RT:1:1, [ATTR_SET attribute: originator AS 65000, origin IGP, aspath , med 0, localpref  
200, cluster list 192.168.100.1 , originator 10.100.1.1]  
BGP(4): 192.168.100.3 rcvd 65000:1:10.100.1.1/32, label 16  
RT address family is not configured. Can't create RTC route  
BGP(4): Revise route installing 1 of 1 routes for 10.100.1.1/32 -> 192.168.100.1  
(customer1) to customer1 IP table  
BGP(4): 10.1.2.5 NEXT_HOP is set to self for net 65000:2:10.100.1.1/32,
```

Note: O PE1 recebeu sua própria atualização do RR e a descartou. Isso ocorre porque PE1 e PE2 estão no mesmo grupo de atualização no RR.

Note: Se quiser descarregar a mensagem de atualização completa em hexadecimal, use a palavra-chave **detail** para o comando **debug BGP updates**.

```
PE2# debug bgp vpnv4 unicast updates detail
```

```
BGP updates debugging is on with detail for address family: VPNv4 Unicast
```

```
PE2#
```

```
BGP(4): 192.168.100.3 rcvd UPDATE w/ attr: nexthop 192.168.100.1, origin i,  
localpref 100, originator 10.100.1.1, clusterlist 192.168.100.3 192.168.100.1,  
extended community RT:1:1, [ATTR_SET attribute: originator AS 65000, origin IGP,  
aspath , med 0, localpref 200, cluster list 192.168.100.1 , originator 10.100.1.1]  
BGP(4): 192.168.100.3 rcvd 65000:1:10.100.1.1/32, label 17  
RT address family is not configured. Can't create RTC route  
BGP: 192.168.100.3 rcv update length 125  
BGP: 192.168.100.3 rcv update dump: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF  
0090 0200 00  
PE2#00 7980 0E21 0001 800C 0000 0000 0000 0000 0000 C0A8 6401 0078 0001 1100 00FD E800  
0000 010A 6401 0140 0101 0040 0200 4005 0400 0000 64C0 1008 0002 0001 0000 0001 800A  
08C0 A864 03C0 A864 0180 0904 0A64 0101 C080 2700 00FD E840 0101 0040 0200 8004 0400  
0000 0040 0504 0000 00C8 800A 04C0 A864 0180 0904 0A64 0101  
BGP(4): Revise route installing 1 of 1 routes for 10.100.1.1/32 -> 192.168.100.1  
(customer1) to customer1 IP table  
BGP(4): 10.1.2.5 NEXT_HOP is set to self for net 65000:2:10.100.1.1/32,
```

Manuseio do próximo salto

O Next-Hop-Self deve ser configurado nos roteadores PE para esse recurso. A razão para isso é que normalmente o próximo salto é transportado sem alterações com o iBGP. No entanto, há duas redes separadas: a rede VPN e a rede SP, que executam protocolos Interior Gateway Protocols (IGPs) separados. Portanto, a métrica IGP não pode ser facilmente comparada e usada para o cálculo do melhor caminho entre as duas redes. A abordagem escolhida pelo RFC 6368 é ter o next-hop-self obrigatório para a sessão iBGP em direção ao CE, o que evita o problema descrito anteriormente em conjunto. Uma vantagem é que os locais de VRF podem executar IGPs diferentes com essa abordagem.

RD

O RFC 6368 menciona que é recomendável que diferentes sites de VRF da mesma VPN usem diferentes RDs (exclusivos). No Cisco IOS, isso é obrigatório para esse recurso.

Recurso iBGP PE-CE com Local-AS

Veja a Figura 2. O VPN customer1 tem ASN 65001.

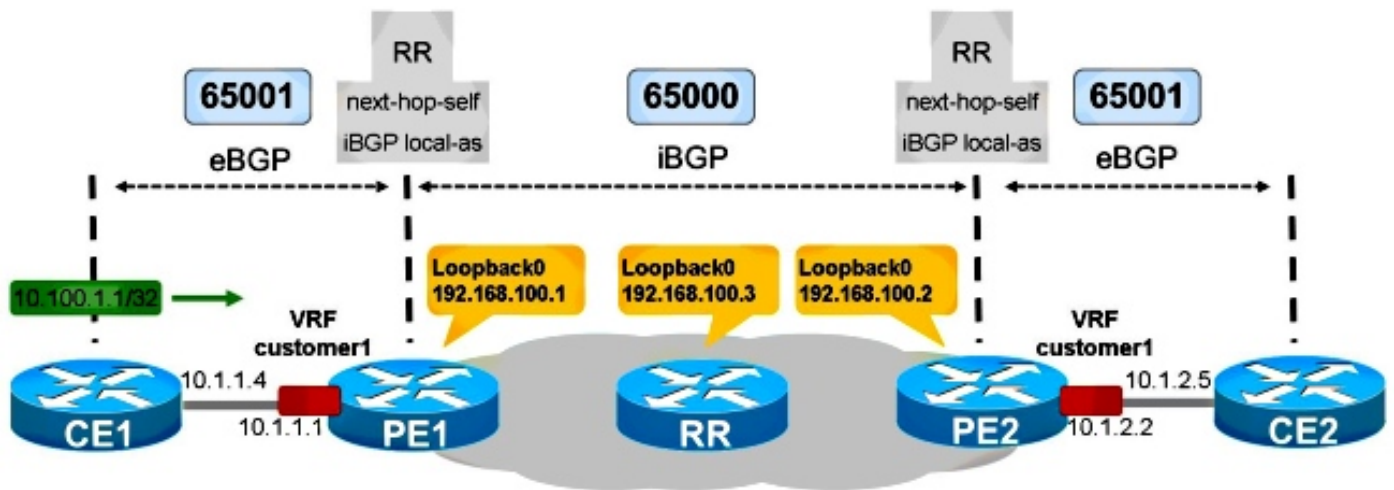


Figure 2

CE1 está no AS 65001. Para tornar esse BGP interno do ponto de vista do PE1, ele precisa do recurso `local-as` do iBGP.

CE1

```
router bgp 65001
  bgp log-neighbor-changes
  network 10.100.1.1 mask 255.255.255.255
  neighbor 10.1.1.1 remote-as 65001
```

PE1

```
router bgp 65000
  bgp log-neighbor-changes
  neighbor 192.168.100.3 remote-as 65000
  neighbor 192.168.100.3 update-source Loopback0
  !
  address-family vpnv4
  neighbor 192.168.100.3 activate
  neighbor 192.168.100.3 send-community extended
  exit-address-family
  !
  address-family ipv4 vrf customer1
  neighbor 10.1.1.4 remote-as 65001
  neighbor 10.1.1.4 local-as 65001
  neighbor 10.1.1.4 activate
  neighbor 10.1.1.4 internal-vpn-client
  neighbor 10.1.1.4 route-reflector-client
  neighbor 10.1.1.4 next-hop-self
```

exit-address-family

PE2 e CE2 são configurados da mesma forma.

O PE1 vê o prefixo do BGP como mostrado aqui:

```
PE1#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 41
Paths: (2 available, best #1, table customer1)
  Advertised to update-groups:
    5
  Refresh Epoch 1
  Local, (Received from ibgp-pece RR-client)
    10.1.1.4 (via vrf customer1) from 10.1.1.4 (10.100.1.1)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      mpls labels in/out 18/nolabel
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  Local, (Received from ibgp-pece RR-client), (ibgp sourced)
    10.1.1.4 (via vrf customer1) from 10.1.1.4 (10.100.1.1)
      Origin IGP, localpref 100, valid, internal
      Extended Community: RT:1:1
      mpls labels in/out 18/nolabel
      rx pathid: 0, tx pathid: 0
```

O prefixo é um BGP interno.

O PE2 vê isso:

```
PE2#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 33
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  Refresh Epoch 5
  Local
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.100.1.1, Cluster list: 192.168.100.3, 192.168.100.1
      ATTR_SET Attribute:
        Originator AS 65001
        Origin IGP
        Aspath
        Med 0
        LocalPref 200
        Cluster list
        192.168.100.1,
        Originator 10.100.1.1
      mpls labels in/out nolabel/18
      rx pathid: 0, tx pathid: 0x0
  BGP routing table entry for 65000:2:10.100.1.1/32, version 34
  Paths: (1 available, best #1, table customer1)
  Advertised to update-groups:
    5
  Refresh Epoch 2
  Local, imported path from 65000:1:10.100.1.1/32 (global)
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      Originator AS(ibgp-pece): 65001
      Originator: 10.100.1.1, Cluster list: 192.168.100.1
      mpls labels in/out nolabel/18
```

```
rx pathid: 0, tx pathid: 0x0
```

O AS do originador é 65001, que é o AS usado quando o prefixo é enviado de PE2 para CE2. Então, o AS é preservado, assim como a preferência local neste exemplo.

```
CE2#show bgp ipv4 unicast 10.100.1.1/32
BGP routing table entry for 10.100.1.1/32, version 3
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
Local
  10.1.2.2 from 10.1.2.2 (192.168.100.2)
    Origin IGP, metric 0, localpref 200, valid, internal, best
    Originator: 10.100.1.1, Cluster list: 192.168.100.2, 192.168.100.1
    rx pathid: 0, tx pathid: 0x0
```

Você vê **Local** em vez de um AS Path. Isso significa que é uma rota BGP interna originada no AS 65001, que também é o ASN configurado do roteador CE2. Todos os atributos BGP foram retirados do atributo ATTR_SET. Isso obedece às regras para o Caso 1 na próxima seção.

Regras para troca de rotas entre diferentes sites de VRF

O ATTR_SET contém o AS do originador do VRF de origem. Este AS de origem é verificado pelo PE remoto, quando ele remove o ATTR_SET antes de enviar o prefixo ao roteador CE.

Caso 1: Se o AS de origem corresponder ao AS configurado para o roteador CE, os atributos do BGP serão retirados do atributo ATTR_SET quando o PE importar o caminho para o VRF de destino.

Caso 2: Se o AS de origem não corresponder ao AS configurado para o roteador CE, então o conjunto de atributos para o caminho construído é tomado como mostrado aqui:

1. Os atributos de caminho são definidos para os atributos contidos no atributo ATTR_SET.
2. Os atributos específicos do iBGP são descartados (LOCAL_PREF, ORIGINATOR e CLUSTER_LIST).
3. O número **AS de origem** contido no atributo ATTR_SET é anexado ao AS_PATH e segue as regras que se aplicam a um peering de BGP externo entre os ASs de origem e de destino.
4. Se o sistema autônomo associado ao VRF for o mesmo que o sistema autônomo do provedor de VPN e o atributo AS_PATH da rota VPN não estiver vazio, DEVE ser anexado ao atributo AS_PATH da rota VRF.

Veja a Figura 3. CE1 e PE1 têm o AS 65000 e são configurados com o recurso iBGP PE-CE. CE2 tem ASN 65001. Isso significa que há eBGP entre PE2 e CE2.

Figure 3

O PE2 vê a rota da seguinte maneira:

```

PE2#show bgp vpv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 43
Paths: (1 available, best #1, no table)
Not advertised to any peer
Refresh Epoch 6
Local
  192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
  Origin IGP, localpref 100, valid, internal, best
  Extended Community: RT:1:1
  Originator: 10.100.1.1, Cluster list: 192.168.100.3, 192.168.100.1
  ATTR_SET Attribute:
    Originator AS 65000
    Origin IGP
    Aspath
    Med 0
    LocalPref 200
    Cluster list
    192.168.100.1,
    Originator 10.100.1.1
  mpls labels in/out nolabel/17
  rx pathid: 0, tx pathid: 0x0
BGP routing table entry for 65000:2:10.100.1.1/32, version 44
Paths: (1 available, best #1, table customer1)
Advertised to update-groups:
  6
Refresh Epoch 6
Local, imported path from 65000:1:10.100.1.1/32 (global)
  192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
  Origin IGP, metric 0, localpref 200, valid, internal, best
  Originator AS(ibgp-pece): 65000
  Originator: 10.100.1.1, Cluster list: 192.168.100.1
  mpls labels in/out nolabel/17
  rx pathid: 0, tx pathid: 0x0

```

Este é o prefixo conforme visto em CE2:

```

CE2#show bgp ipv4 unicast 10.100.1.1/32
BGP routing table entry for 10.100.1.1/32, version 5
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
65000
  10.1.2.2 from 10.1.2.2 (192.168.100.2)
  Origin IGP, localpref 100, valid, external, best
  rx pathid: 0, tx pathid: 0x0

```

Este é o caso 2. O número **AS de origem** contido no atributo ATTR_SET é anexado ao AS_PATH pelo PE2 e segue as regras que se aplicam a um peering eBGP entre o AS de origem e de destino. Os atributos específicos do iBGP são ignorados pelo PE2 quando ele cria a rota a ser anunciada para CE2. Portanto, a preferência local é 100 e não 200 (como visto no atributo ATTR_SET).

Reflexão CE-to-CE VRF-Lite

Veja a Figura 4.

Figure 4

A Figura 4 mostra um roteador CE adicional, CE3, conectado ao PE1. CE1 e CE3 estão conectados a PE1 na mesma instância de VRF: cliente1. Isso significa que CE1 e CE3 são roteadores CE multi-VRF (também conhecidos como VRF-Lite) de PE1. O PE1 se coloca como próximo salto quando anuncia os prefixos de CE1 a CE3. Caso esse comportamento não seja desejado, você pode configurar o **vizinho 10.1.3.6 próximo salto inalterado** em PE1. Para configurar isso, você deve remover o **vizinho 10.1.3.6 next-hop-self** em PE1. Em seguida, CE3 vê as rotas de CE1 com CE1 como o próximo salto para esses prefixos de BGP. Para que isso funcione, você precisa das rotas para os próximos saltos BGP na tabela de roteamento de CE3. Você precisa de um Dynamic Routing Protocol (IGP) ou rotas estáticas em CE1, PE1 e CE3 para garantir que os roteadores tenham uma rota para os endereços IP do próximo salto uns dos outros. Entretanto, há um problema com essa configuração.

A configuração em PE1 é:

```
router bgp 65000
!
address-family ipv4 vrf customer1
neighbor 10.1.1.4 remote-as 65000
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 internal-vpn-client
neighbor 10.1.1.4 route-reflector-client
neighbor 10.1.1.4 next-hop-self
neighbor 10.1.3.6 remote-as 65000
neighbor 10.1.3.6 activate
neighbor 10.1.3.6 internal-vpn-client
neighbor 10.1.3.6 route-reflector-client
neighbor 10.1.3.6 next-hop-unchanged
exit-address-family
```

O prefixo de CE1 é visto como fino em CE3:

```
CE3#show bgp ipv4 unicast 10.100.1.1
BGP routing table entry for 10.100.1.1/32, version 9
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
Local
  10.1.1.4 from 10.1.3.1 (192.168.100.1)
    Origin IGP, metric 0, localpref 200, valid, internal, best
    Originator: 10.100.1.1, Cluster list: 192.168.100.1
    rx pathid: 0, tx pathid: 0x0
```

No entanto, o prefixo de CE2 é visto em CE3 como mostrado aqui:

```
CE3#show bgp ipv4 unicast 10.100.1.2
BGP routing table entry for 10.100.1.2/32, version 0
Paths: (1 available, no best path)
Not advertised to any peer
Refresh Epoch 1
Local
  192.168.100.2 (inaccessible) from 10.1.3.1 (192.168.100.1)
    Origin IGP, metric 0, localpref 100, valid, internal
    Originator: 10.100.1.2, Cluster list: 192.168.100.1, 192.168.100.2
    rx pathid: 0, tx pathid: 0
```

O salto seguinte do BGP é **192.168.100.2**, o endereço IP de loopback de PE2. O PE1 não regravou o próximo salto do BGP para si mesmo quando anunciou o prefixo 10.100.1.2/32 para CE3. Isso torna esse prefixo inutilizável em CE3.

Portanto, no caso de uma combinação do recurso iBGP PE-CE entre MPLS-VPN e iBGP VRF-Lite, você deve certificar-se de que sempre tem o Next-Hop-Self nos roteadores PE.

Você não pode preservar o próximo salto quando um roteador PE é um RR que reflete rotas iBGP de um CE para outro CE através de interfaces VRF localmente no PE. Quando você executa o iBGP PE-CE através de uma rede VPN MPLS, você deve usar **internal-vpn-client** para as sessões iBGP em direção aos roteadores CE. Quando você tem mais de um CE local em um VRF em um roteador PE, você deve manter o **next-hop-self** para esses peers BGP.

Você pode examinar os mapas de rotas para definir o próximo salto como auto para prefixos recebidos de outros roteadores PE, mas não para prefixos refletidos de outros roteadores CE conectados localmente. No entanto, não há suporte atualmente para definir o próximo salto como auto em um mapa de rota de saída. Essa configuração é mostrada aqui:

```
router bgp 65000

address-family ipv4 vrf customer1
neighbor 10.1.1.4 remote-as 65000
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 internal-vpn-client
neighbor 10.1.1.4 route-reflector-client
neighbor 10.1.1.4 next-hop-self
neighbor 10.1.3.6 remote-as 65000
neighbor 10.1.3.6 activate
neighbor 10.1.3.6 internal-vpn-client
neighbor 10.1.3.6 route-reflector-client
neighbor 10.1.3.6 route-map NH-setting out
exit-address-family

ip prefix-list PE-loopbacks seq 10 permit 192.168.100.0/24 ge 32
!

route-map NH-setting permit 10
description set next-hop to self for prefixes from other PE routers
match ip route-source prefix-list PE-loopbacks
set ip next-hop self
!

route-map NH-setting permit 20
description advertise prefixes with next-hop other than the prefix-list in
route-map entry 10 above
!
```

No entanto, isso não é suportado:

```
PE1(config)#route-map NH-setting permit 10
PE1(config-route-map)# set ip next-hop self
% "NH-setting" used as BGP outbound route-map, set use own IP/IPv6 address for the nexthop not supported
```

Cisco IOS mais antigo no roteador PE

Se o PE1 executa o software Cisco IOS mais antigo que não tem o recurso iBGP PE-CE, o PE1 nunca se define como o próximo salto para os prefixos iBGP refletidos. Isso significa que o prefixo BGP refletido (10.100.1.1/32) de CE1 (10.100.1.1) para CE2 - via PE1 - teria CE1 (10.1.1.4) como o próximo salto.

```
CE3#show bgp ipv4 unicast 10.100.1.1
BGP routing table entry for 10.100.1.1/32, version 32
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    10.1.1.4 from 10.1.3.1 (192.168.100.1)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      Originator: 10.100.1.1, Cluster list: 192.168.100.1
      rx pathid: 0, tx pathid: 0x0
```

O prefixo do CE2 (10.100.1.2/32) é visto com PE2 como o próximo salto, porque o PE1 não faz o próximo salto de si mesmo para este prefixo:

```
CE3#show bgp ipv4 unicast 10.100.1.2
BGP routing table entry for 10.100.1.2/32, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    192.168.100.2 (inaccessible) from 10.1.3.1 (192.168.100.1)
      Origin IGP, localpref 100, valid, internal
      Originator: 10.100.1.2, Cluster list: 192.168.100.1, 192.168.100.3, 192.168.100.2
      ATTR_SET Attribute:
        Originator AS 65000
        Origin IGP
        Aspath
        Med 0
        LocalPref 100
        Cluster list
        192.168.100.2,
        Originator 10.100.1.2
      rx pathid: 0, tx pathid: 0
```

Para que o recurso iBGP PE-CE funcione corretamente, todos os roteadores PE para a VPN onde o recurso está ativado devem ter o código para suportar o recurso e ter o recurso ativado.

Próximo salto para eBGP em VRF

Veja a Figura 5.

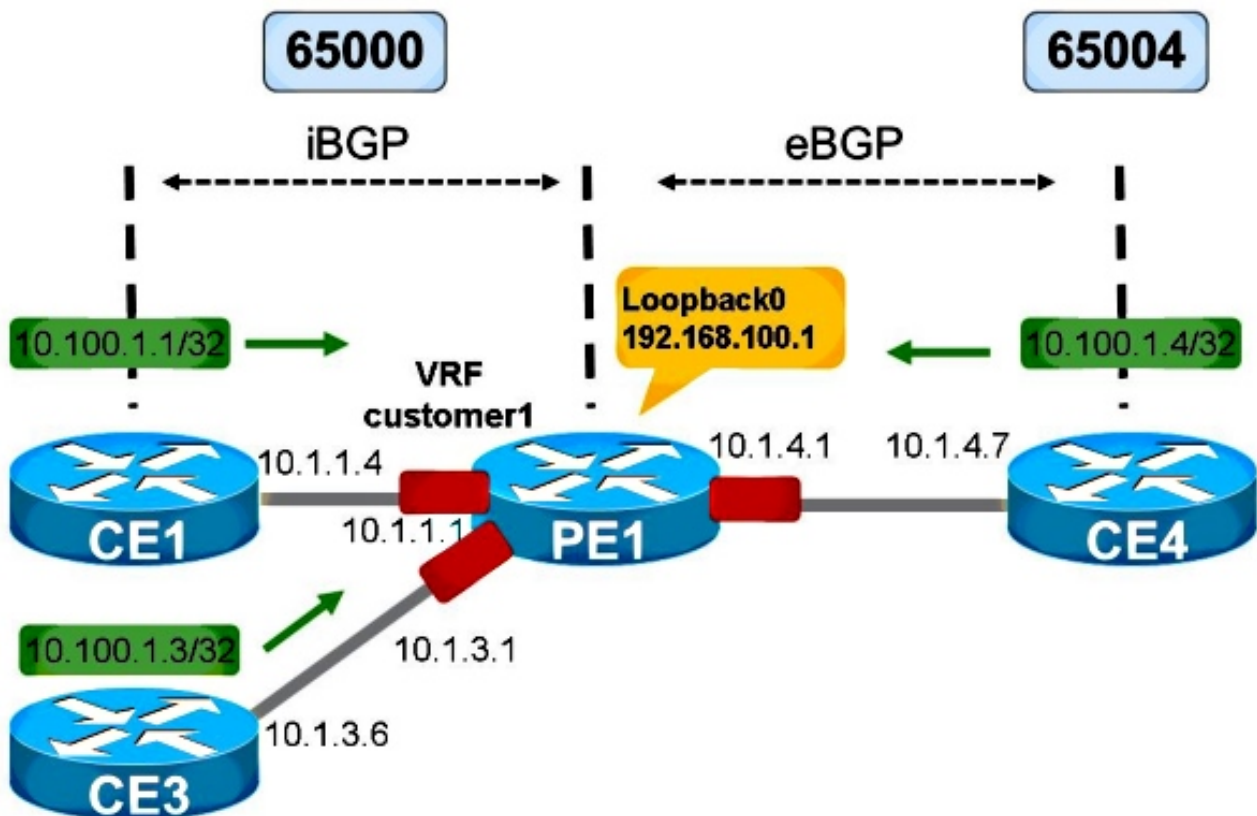


Figure 5

A Figura 5 mostra uma configuração do VRF-Lite. A sessão de PE1 para CE4 é eBGP. A sessão de PE1 para CE3 ainda é iBGP.

Para prefixos eBGP, o próximo salto é sempre definido como auto quando anuncia os prefixos em direção a um vizinho iBGP no VRF. Isso independentemente do fato de que a sessão em direção ao vizinho iBGP através de VRF tem o Next-Hop-Self definido ou não.

Na Figura 5, CE3 vê os prefixos de CE4 com PE1 como o próximo salto.

```
CE3#show bgp ipv4 unicast 10.100.1.4
BGP routing table entry for 10.100.1.4/32, version 103
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
65004
 10.1.3.1 from 10.1.3.1 (192.168.100.1)
  Origin IGP, metric 0, localpref 100, valid, internal, best
  rx pathid: 0, tx pathid: 0x0
```

Isso ocorre com o Next-Hop-Self em PE1 em direção a CE3 ou sem.

Se as interfaces em PE1 em direção a CE3 e CE4 não estiverem em um VRF, mas no contexto global, o Next-Hop-Self em direção a CE3 faz a diferença.

Sem o salto seguinte no PE1 em direção ao CE3, você vê:

```
PE1#show bgp vrf customer1 vpnv4 unicast neighbors 10.1.3.6
```



```
BGP neighbor is 10.1.3.6, vrf customer1, remote AS 65000, internal link
...
For address family: VPNv4 Unicast
Translates address family IPv4 Unicast for VRF customer1
Session: 10.1.3.6
BGP table version 1, neighbor version 1/0
Output queue size : 0
Index 12, Advertise bit 0
Route-Reflector Client
12 update-group member
Slow-peer detection is disabled
Slow-peer split-update-group dynamic is disabled
Interface associated: (none)
```

Embora o next-hop-self esteja implicitamente ativado, a saída não indica isso.

Com o Next Hop Self no PE1 em direção ao CE3, você vê:

```
PE1#show bgp vrf customer1 vpnv4 unicast neighbors 10.1.3.6
BGP neighbor is 10.1.3.6, vrf customer1, remote AS 65000, internal link
..
For address family: VPNv4 Unicast
...
NEXT_HOP is always this router for eBGP paths
```

Considerando que, se as interfaces para CE3 e CE4 estiverem em um contexto global, o próximo salto para prefixos do CE4 será o próprio CE4 quando o Next Hop-Self não estiver configurado:

```
CE3#show bgp ipv4 unicast 10.100.1.4
BGP routing table entry for 10.100.1.4/32, version 124
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
65004
10.1.4.7 from 10.1.3.1 (192.168.100.1)
Origin IGP, metric 0, localpref 100, valid, internal, best
rx pathid: 0, tx pathid: 0x0
```

Para o próximo salto em PE1 em direção a CE3:

```
CE3#show bgp ipv4 unicast 10.100.1.4
BGP routing table entry for 10.100.1.4/32, version 125
Paths: (1 available, best #1, table default)
Not advertised to any peer
Refresh Epoch 1
65004
10.1.3.1 from 10.1.3.1 (192.168.100.1)
Origin IGP, metric 0, localpref 100, valid, internal, best
rx pathid: 0, tx pathid: 0x0
```

Isso foi feito com base no RFC 4364.

Se você não quiser definir o Next-Hop-Self para prefixos eBGP em direção a uma sessão iBGP através de uma interface VRF, você deve configurar o próximo salto inalterado. O suporte para isso ocorreu somente com a ID de bug da Cisco [CSCuj11720](#).

```
router bgp 65000
...
address-family ipv4 vrf customer1
```

```
neighbor 10.1.1.4 remote-as 65000
neighbor 10.1.1.4 activate
neighbor 10.1.1.4 route-reflector-client
neighbor 10.1.3.6 remote-as 65000
neighbor 10.1.3.6 activate
neighbor 10.1.3.6 route-reflector-client
neighbor 10.1.3.6 next-hop-unchanged
neighbor 10.1.4.7 remote-as 65004
neighbor 10.1.4.7 activate
exit-address-family
```

Agora, o CE3 vê o CE4 como o próximo salto para os prefixos anunciados pelo CE4:

```
CE3#show bgp ipv4 unicast 10.100.1.4
BGP routing table entry for 10.100.1.4/32, version 130
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 3
  65004
    10.1.4.7 from 10.1.3.1 (192.168.100.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

Se você tentar configurar a palavra-chave **next-hop-changed** para a sessão iBGP em direção ao CE3 no código do Cisco IOS antes do bug da Cisco ID [CSCuj11720](#), você encontrará este erro:

```
PE1(config-router-af)# neighbor 10.1.3.6 next-hop-unchanged
%BGP: Can propagate the nexthop only to multi-hop EBGP neighbor
```

Após o ID de bug da Cisco [CSCuj11720](#), a palavra-chave **next-hop inalterado** é válida para vizinhos eBGP multi-hop e vizinhos iBGP VRF-Lite.