

# VPL com Nota Técnica da sinalização BGP

## Índice

[Introdução](#)

[Problema](#)

[Arquitetura da solução](#)

[Identificação do valor-limite do roteador de PE](#)

[Identidade VPL e etiquetas MPLS](#)

[Informação de encapsulamento](#)

[Prefixo da descoberta automática VPL BGP e de sinalização VPL BGP prefixo](#)

[Configuração de Cisco IOS Software da amostra](#)

[Bloco anunciado da etiqueta](#)

[Alvos do distinguidor de rota e da rota](#)

[Exemplo de configuração do PE1](#)

[Verifique a escala da etiqueta](#)

[Verifique etiquetas](#)

[Verifique o bloco da etiqueta](#)

[Verifique o prefixo anunciado](#)

[Veja o prefixo em detalhe](#)

[Anuncie, receba, e blocos da etiqueta do processo em mensagens da atualização BGP](#)

[PE2: Receba a atualização BGP](#)

[PE2: Encontre uma etiqueta](#)

[PE2: Envie o prefixo ao PE1](#)

[PE1: Encontre uma etiqueta](#)

[Comandos de verificação adicionais](#)

[Prefixos múltiplos L2VPN VPL anunciados pelo roteador de PE para um VFI](#)

[Configuração PE1](#)

[Configuração PE2](#)

[Blocos iniciais da etiqueta](#)

[Troca PE1 e PE2](#)

[Análise da troca PE1 e PE2](#)

[Detalhes do prefixo](#)

[Interoperabilidade](#)

## Introdução

Este documento descreve o Border Gateway Protocol (BGP) - descoberta automática baseada para um serviço virtual da LAN privada (VPL) com sinalização BGP. A descoberta automática é meios para uma ponta de provedor (PE) aprender que PE remotos são membros de um domínio dado VPL. A sinalização é meios para que um PE aprenda a etiqueta do pseudowire esperada

por um PE remoto dado para um domínio dado VPL.

Refira estes documentos do Internet Engineering Task Force:

- [O serviço virtual da LAN privada do RFC 4762 \(VPL\) que usa a sinalização do protocolo de distribuição de rótulo \(LDP\)](#) descreve a descoberta automática BGP com sinalização do protocolo de distribuição de rótulo (LDP) para os VPL (igualmente conhecidos como Martini).
- [O serviço virtual da LAN privada do RFC 4761 \(VPL\) que usa o BGP para a descoberta automática e que sinaliza](#) descreve a descoberta automática BGP e a sinalização BGP para os VPL (igualmente conhecidos como Kompella).

Este documento focaliza no RFC 4761. Com RFC 4761, a informação de alcançabilidade da camada de rede BGP (NLRI) das atualizações BGP guarda a informação para a descoberta automática e a sinalização. Quando os roteadores de PE remotos recebem esta atualização BGP, têm toda a informação necessária a fim estabelecer uma malha cheia dos pseudowires para VPL. A descoberta automática BGP e a sinalização BGP usam a mesma endereço-família BGP.

O comando line interface(cli) e a saída são do Cisco IOS ® Software. A configuração e a funcionalidade são muito similares no Software Cisco IOS XR e no Software Cisco NX-OS.

## Problema

Os VPL consistem em um grupo dos pseudowires (picowatt) em uma forma point-to-multipoint. Até aqui, o LDP foi usado para sinalizar os pseudowires entre os roteadores de PE. Assim, uma sessão LDP visada sinalizou que etiquetas a se usar para que pseudowire entre um par de roteadores de PE. Você poderia manualmente configurar o grupo de roteadores de PE que participaram em um domínio VPL, ou você poderia usar o BGP para descobrir automaticamente a configuração. A fim executar esta descoberta automática, o BGP anunciou que PE era um membro de que domínio VPL. Contudo, mesmo com descoberta automática BGP, o LDP foi usado para sinalizar as etiquetas do virtual circuit (VC) do Multiprotocol Label Switching (MPLS) e o pseudowire ID.

É agora possível usar o BGP a fim sinalizar os pseudowires entre os roteadores de PE.

Quando um pseudowire deve se estabelecer entre um par de Roteadores, o outro Roteadores não precisa o relativo à informação a este pseudowire. Por exemplo, tal informação é a etiqueta VC a ser usada.

Com o LDP como o protocolo de sinalização para estabelecer pseudowires, a informação é recebida somente pelos pares de Roteadores, porque o LDP faz a sinalização em uma forma ponto a ponto.

Com o BGP como o protocolo de sinalização para estabelecer pseudowires, a informação é recebida por todo Roteadores restante porque o Internal BGP (iBGP) faz a sinalização em uma forma point-to-multipoint. o iBGP tem uma exigência da malha cheia, assim que um roteador envia uma atualização do iBGP a todos roteadores iBGP restantes. Isto podia igualmente ser feito com um refletor de rota.

Com o iBGP como o protocolo de sinalização, haveria dois métodos para enviar atualizações:

1. Cada roteador de PE anuncia uma atualização BGP a todos os vizinhos iBGP para cada

picowatt; cada vez que, uma etiqueta MPLS VC é anexada. Assim, um roteador de PE enviaria tantas como atualizações BGP porque há roteadores de PE. Contudo, a etiqueta VC anexada à atualização BGP podia ser usada por somente um dos roteadores de PE - o roteador de PE no outro extremo do picowatt.

2. Para evitar esta introdução de um alto número de atualizações BGP, uma arquitetura foi projetada por meio de que um roteador de PE local envia um grupo ou um bloco de etiquetas locais VC a todos os roteadores de PE remotos. Cada roteador de PE remoto escolhe uma das etiquetas VC para usar-se como uma etiqueta remota VC para o picowatt para o roteador de PE local. O roteador de PE remoto deve escolher uma etiqueta remota VC em uma forma original de modo que nenhum outro roteador de PE escolha a mesma etiqueta VC do bloco anunciado de etiquetas. Desde que um bloco de etiquetas é enviado, deve haver bastante etiquetas disponíveis a fim servir todo o PWs possível que poderia se estabelecer, mas não deve haver tão muitas etiquetas reservadas que são não utilizados e desperdiçaram.

Este documento descreve como o BGP é usado a fim sinalizar os pseudowires; note que o BGP está usado igualmente para a descoberta automática ao mesmo tempo.

## Arquitetura da solução

Porque este é VPL, há ainda um protocolo de sinalização do salto a salto necessário no núcleo a fim levar os pacotes rotulados do PE ao roteador de PE. Esta função do transporte no núcleo deve ainda ser cumprida pelo LDP ou pela Engenharia de tráfego MPLS.

O BGP precisa de enviar a informação necessária a fim estabelecer os pseudowires em uma forma point-to-multipoint necessária por VPL. Esta informação de sinalização inclui:

- Identificação do valor-limite do roteador de PE
- Identidade VPL
- Bloco de etiquetas MPLS
- Informação de encapsulamento

### Identificação do valor-limite do roteador de PE

A identificação do valor-limite do roteador de PE é determinada do roteador de PE que é o remetente BGP da atualização.

A atualização BGP que refere-se as redes privadas virtuais da camada 2 (L2VPN) VPL é identificada por AFI/SAFI 25/65. Esta família do endereço é negociada quando o BGP envia a mensagem ABERTA.

### Identidade VPL e etiquetas MPLS

O NLRI, igualmente conhecido como o prefixo, guarda a informação na identidade VPL e o bloco de etiquetas MPLS. Sua codificação tem um comprimento total de 19 bytes:

+-----+

```

| Length (2 octets) |
+-----+
| Route Distinguisher (8 octets) |
+-----+
| VE ID (2 octets) |
+-----+
| VE Block Offset (2 octets) |
+-----+
| VE Block Size (2 octets) |
+-----+
| Label Base (3 octets) |
+-----+

```

O distinguidor de rota (RD) relaciona-se à identidade dos VPL.

**Note:** Na aplicação do Cisco IOS e do Software Cisco NX-OS, todos os roteadores de PE devem ter o mesmo RD dentro do mesmo domínio VPL.

A expansão virtual (VE) ID, VE obstrui o offset, tamanho de bloco VE, e a base da etiqueta (LB) relaciona-se ao bloco anunciado de etiquetas, como explicado na próxima seção.

## Informação de encapsulamento

A informação de encapsulamento igualmente é anexada ao prefixo e codificada como uma comunidade estendida 'Layer2 Community estendido informação à atualização BGP. O valor é 0x800A e é codificado como:

```

+-----+
| Extended community type (2 octets) |
+-----+
| Encaps Type (1 octet) |
+-----+
| Control Flags (1 octet) |
+-----+
| Layer-2 MTU (2 octet) |
+-----+
| Reserved (2 octets) |
+-----+

```

Os encaps datilografam para VPL são 19.

Os flags de controle (vetor do bit) são codificados esta maneira:

```

  0 1 2 3 4 5 6 7
+-----+
| MBZ   |C|S|      (MBZ = MUST Be Zero)
+-----+

```

### Nome Valor Significado

- |   |   |   |
|---|---|---|
| C | 1 | Uma palavra de controle OBRIGAÇÃO esta presente quando os pacotes VPL são enviados este PE.               |
|   | 0 | Uma palavra de controle NÃO DEVE esta presente quando os pacotes VPL são enviados a PE.                   |
| S | 1 | A entrega arranjada em sequência dos quadros DEVE ser usada quando os pacotes VPL são enviados a este PE. |
|   | 0 | A entrega arranjada em sequência dos quadros NÃO DEVE ser usada quando os pacotes V                       |

são enviados a este PE.

Há igualmente os alvos da rota (RT) anexados à atualização BGP. Os RT controlam a importação em e a exportação do L2VPN, da mesma forma como MPLS L3VPN.

## Prefixo da descoberta automática VPL BGP e de sinalização VPL BGP prefixo

Um prefixo da descoberta automática VPL BGP é um prefixo de /96, visto que um prefixo da sinalização VPL BGP é um prefixo de /136. Estes são exemplos de cada um:

```
PE2#show bgp l2vpn vpls all
BGP table version is 264, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-150/136	10.100.1.1	0	100	0	?
*> 1:100:10.100.1.2/96	0.0.0.0			32768	?

```
PE2#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 150
BGP routing table entry for 1:100:VEID-1001:Blk-150/136, version 262
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Not advertised to any peer
Refresh Epoch 1
Local
10.100.1.1 (metric 21) from 10.100.1.4 (10.100.1.4)
Origin incomplete, metric 0, localpref 100, valid, internal, best
AGI version(0), VE Block Size(50) Label Base(10105)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
Originator: 10.100.1.1, Cluster list: 10.100.1.4
rx pathid: 0, tx pathid: 0x0
```

```
PE2#show bgp l2vpn vpls rd 1:100 10.100.1.2
BGP routing table entry for 1:100:10.100.1.2/96, version 43
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Not advertised to any peer
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.100.1.2)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local,
best, AGI version(0)
Extended Community: RT:1:100 L2VPN AGI:1:100
rx pathid: 0, tx pathid: 0x0
```

## Configuração de Cisco IOS Software da amostra

Esta é uma configuração de Cisco IOS Software da amostra:

```

PE2#show bgp l2vpn vpls all
BGP table version is 264, local router ID is 10.100.1.2
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: 1:100
 *>i 1:100:VEID-1001:Blk-150/136
                10.100.1.1                0      100      0 ?
*> 1:100:10.100.1.2/96
                0.0.0.0                    32768 ?

```

```

PE2#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 150
BGP routing table entry for 1:100:VEID-1001:Blk-150/136, version 262
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Not advertised to any peer
Refresh Epoch 1
Local
 10.100.1.1 (metric 21) from 10.100.1.4 (10.100.1.4)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
  AGI version(0), VE Block Size(50) Label Base(10105)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  Originator: 10.100.1.1, Cluster list: 10.100.1.4
  rx pathid: 0, tx pathid: 0x0

```

```

PE2#show bgp l2vpn vpls rd 1:100 10.100.1.2
BGP routing table entry for 1:100:10.100.1.2/96, version 43
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Not advertised to any peer
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.2)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local,
  best, AGI version(0)
  Extended Community: RT:1:100 L2VPN AGI:1:100
  rx pathid: 0, tx pathid: 0x0

```

## Bloco anunciado da etiqueta

Um roteador de PE deve anunciar pelo menos um bloco da etiqueta. O bloco da etiqueta é um grupo contínuo de etiquetas MPLS e é usado pelos roteadores de PE remotos a fim selecionar uma etiqueta remota VC. A etiqueta remota é usada para o picowatt entre o roteador de PE local e remoto. (Um roteador de PE pode anunciar blocos do rótulo múltiplo, como explicado em seções mais recente.)

O VE-ID deve ser configurado em cada PE. Identifica os roteadores de PE dentro do domínio VPL.

O tamanho de bloco VE (VB) é o tamanho do bloco da etiqueta e tem um valor padrão do 10. Se a “VE varia” é configurada, ele é esse valor. a “VE varia” pode ser configurada para ser [11 -100].

A base da etiqueta (LB) é o primeiro valor de rótulo de um grupo livre de etiquetas que podem ser reservadas pelo roteador de PE a ser usado para este domínio VPL.

O offset do bloco VE (VBO) é o valor de deslocamento a ser usado quando os blocos do rótulo

múltiplo devem ser criados por um roteador de PE. VBO é calculado com esta equação:  $VBO = RND(VE-ID/VBS) * VB$

Estes são exemplos de cálculo:

- Se  $VB = 8$  e  $VE-ID = 2$ ,  $VBO = RND(2/8) * 8 = 1$
- Se  $VB = 8$  e  $VE-ID = 20$ ,  $VBO = RND(20/8) * 8 = 16$
- Se  $VB = 50$  pés e  $VE-ID = 199$ ,  $50$  pés  $VBO = RND(199/50) * 50 = 150$
- Se  $VB = 50$  pés e  $VE-ID = 1002$ ,  $50$  pés  $VBO = RND(1002/50) * 50 = 1000$

O bloco da etiqueta anunciado aos roteadores de PE remotos é  $\{LB, LB + 1, \dots, LB + VB - 1\}$ . O bloco da etiqueta é definido pelo LB e pelos VB; o bloco começa no LB e nas extremidades com  $(LB + VB - 1)$ .

Os blocos do rótulo múltiplo podem ser criados por cada roteador de PE, quando necessários. O roteador deve assegurar-se de que seja um grupo contínuo de etiquetas livres.

## Alvos do distinguidor de rota e da rota

### Exemplo de configuração do PE1

```
router bgp 1

l2vpn vfi context one
  vpn id 100
  autodiscovery bgp signaling bgp
  ve id 1001
  ve range 50
  route-target export 32:64
  route-target import 32:64

mpls label range 10000 20000
```

Esta é uma explicação dos valores da configuração:

- O VPN ID é AS100 configurado.
- O RD é tomado de [ASN: identificação do vpn], a menos que um RD for configurado explicitamente. Aqui, o RD é 1:100.
- Os alvos da importação/rota de exportação são 32:64.
- O LB é da escala [10000 20000]. O valor exato do LB depende do primeiro grupo de etiquetas locais contínuas livres que é grande bastante manter todas as etiquetas determinadas pelos VB.
- O VE-ID é configurado como 1001.
- Os VB são configurados como 50 pés.
- O VBO é calculado para ser:  $VBO = RND(VE-ID/VBS) * VB$  ou  $RND(1001/50) * 50$  pés = 1000.

### Verifique a escala da etiqueta

Você pode verificar a escala da etiqueta com o comando `show mpls label range`:

```

router bgp 1

l2vpn vfi context one
  vpn id 100
  autodiscovery bgp signaling bgp
  ve id 1001
  ve range 50
  route-target export 32:64
  route-target import 32:64

```

```
mpls label range 10000 20000
```

Há uma escala da etiqueta do padrão pela plataforma, que você pode mudar com o comando **mpls label range**.

## Verifique etiquetas

Você pode verificar as etiquetas usadas reais para ver se há um bloco de etiqueta na base de informação de encaminhamento de rótulo (LFIB) com o comando **show mpls forwarding-table**.

```

PE1#show mpls forwarding-table
Local      Outgoing  Prefix          Bytes Label  Outgoing  Next Hop  Label
Label     or Tunnel Id  Switched      interface
10000     No Label    lbl-blk-id(1:0) 0           drop
10001     No Label    lbl-blk-id(1:1) 0           drop
10002     No Label    lbl-blk-id(1:2) 0           drop
?
10048     No Label    lbl-blk-id(1:48) 0           drop
10049     No Label    lbl-blk-id(1:49) 0           drop
10050     Pop Label   10.100.1.4/32   0           Et1/0     10.1.1.4

```

Neste exemplo, PE1, o roteador local, etiquetas reservados do local dos 50 pés para o bloco da etiqueta. 'lbl-blk-id(1:0) significa uma bloco-identificação de 1 e um bloco-exemplo de 0, que identifica a primeira etiqueta do bloco. A última etiqueta deste bloco é a etiqueta 10049.

A relação “que parte” no LFIB é “gota” enquanto não há nenhum picowatt estabelecido para essa etiqueta local. Se um picowatt se estabelece, a relação “que parte” não é 'nenhuma point2point.

## Verifique o bloco da etiqueta

O bloco atribuído da etiqueta pode igualmente ser verificado com o comando **summary do bloco-base de dados do lfd da infraestrutura de MPLS da mostra** quando o “serviço interno” é configurado.

```

PE1#show mpls infrastructure lfd block-database summary
Block-DB entry for block-id : 0x1
Block-size : 50, App-Key type : ATOM PWID, Labels : 10000 - 10049

```

O LB é 10000. Neste exemplo, o bloco da etiqueta é do LB (LB + VB - 1) ou a 10000 (10000 + 50 pés - 1) = a 10049.

## Verifique o prefixo anunciado



Você pode verificar o prefixo anunciado com o comando **rd 1:100 dos vpls BGP l2vpn da mostra:**

```
PE1#show mpls infrastructure lfd block-database summary
Block-DB entry for block-id : 0x1
Block-size : 50, App-Key type : AToM PWID, Labels : 10000 - 10049
```

## Prefixo da vista em detalhe

Para ver em detalhe este prefixo, use o comando do bloco-**offset VE-identificação 1001 1000 rd 1:100 dos vpls BGP l2vpn da mostra**. Note que você especifica o VE-ID e o bloco da etiqueta, que podem ser encontrados no NLRI (Blk-1000).

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(50) Label Base(10000)
      Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
      rx pathid: 0, tx pathid: 0x0
```

O NLRI mostra o RD de 1:100, o VE-ID de 1001, o VBO de 1000, os VB dos 50 pés, e o LB de 10000.

A comunidade extendida da informação Layer2 guarda esta informação:

- O tipo do encaps é 19 (os VPL)
- Flags de controle: C = 0 (uma palavra de controle não deve ser ajustada); S = 0 (nenhuma entrega arranjada em sequência dos quadros)
- O MTU é 1500

A comunidade extendida RT guarda esta informação:

- RT 1:100
- RT 32:64

**Note:** O padrão VB (10) é pequeno de modo que as etiquetas locais não sejam desperdiçadas.

## Anuncie, receba, e blocos da etiqueta do processo em mensagens da atualização BGP

Quando um roteador de PE local anuncia um bloco do prefixo/etiqueta L2VPN VPL, cada roteador de PE remoto deve tentar escolher uma etiqueta dessa escala a fim usar-se como uma etiqueta remota VC.

- Se o roteador de PE remoto sucede, usa-se que etiqueta remota VC e programa-se a no

plano dos dados. Não há nenhuma sinalização mais adicional pelo BGP.

- Se o roteador de PE remoto falha, deve esperar um outro prefixo L2VPN VPL a ser anunciado pelo roteador de PE local, a seguir tenta escolher uma outra etiqueta remota VC desse bloco da etiqueta.

Supõe que o PE1 é um PE local com a configuração precedente e que o PE2 é um PE remoto com esta configuração:

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(50) Label Base(10000)
      Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
      rx pathid: 0, tx pathid: 0x0
```

## PE2: Receba a atualização BGP

O PE2 recebe esta atualização BGP do PE1:

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(50) Label Base(10000)
      Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
      rx pathid: 0, tx pathid: 0x0
```

## PE2: Encontre uma etiqueta

O PE2 precisa de encontrar uma etiqueta que pode se usar como uma etiqueta remota VC para o picowatt para o PE1.

O PE2 deve primeiramente determinar se o VBO está dentro da escala de sua configuração. O PE2 verifica seu VE-ID contra a escala anunciada pelo PE1 com o  $\leq$  do cálculo  $VBO \text{ VE-ID} < VBO + VB$ . Neste caso,  $1000 \leq 1002 < 1000 + 50$  pés, assim que o PE2 sucedem.

O PE2 precisa então de escolher uma etiqueta remota VC. A etiqueta do desmultiplexador (VC) a ser usada pelo PE remoto é computada como  $(LB + VE-ID - VBO)$ .

Do prefixo mais adiantado, o LB é 10000, e VBO é 1000. O VE-ID é esse do PE2 e é 1002. Assim, etiqueta das picaretas PE2  $(LB + VE-ID - VBO) = (10000 + 1002 - 1000) = 10002$ .

Use o comando do nome um do vfi da mostra l2vpn a fim verificar isto:

```

PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(50) Label Base(10000)
      Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
      rx pathid: 0, tx pathid: 0x0

```

## PE2: Envie o prefixo ao PE1

O PE2 envia então seu prefixo ao PE1:

```

PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.100.1.1)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      AGI version(0), VE Block Size(50) Label Base(10000)
      Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
      rx pathid: 0, tx pathid: 0x0

```

## PE1: Encontre uma etiqueta

O PE1 é agora o PE e as necessidades remotos encontrar uma etiqueta que pode se usar como uma etiqueta remota VC para o picowatt para o PE2.

O PE1 deve primeiramente determinar se o VBO está dentro da escala de sua configuração. O PE1 verifica seu VE-ID contra a escala anunciada pelo PE2 com o  $\leq$  do cálculo  $VBO \text{ VE-ID} < VBO + VB$ . Neste caso,  $1000 \leq 1001 < 1000 + 50$  pés, assim que o PE1 sucedem.

O PE1 precisa então de escolher uma etiqueta remota VC. A etiqueta do desmultiplexador (VC) a ser usada pelo PE remoto é computada como  $(LB + VE-ID - VBO)$ .

Do prefixo mais adiantado, o LB é 3100, e VBO é 1000. O VE-ID é esse do PE1 e é 1001. Assim, etiqueta das picaretas PE1  $(LB + VE-ID - VBO) = (3100 + 1001 - 1000) = 3101$ .

Use o comando do nome um do vfi da mostra l2vpn a fim verificar isto:

```

PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local

```

```
0.0.0.0 from 0.0.0.0 (10.100.1.1)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

## Comandos de verificação adicionais

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.100.1.1)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.100.1.1)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.100.1.1)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
0.0.0.0 from 0.0.0.0 (10.100.1.1)
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  AGI version(0), VE Block Size(50) Label Base(10000)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  AGI version(0), VE Block Size(50) Label Base(10000)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  rx pathid: 0, tx pathid: 0x0
```

## Prefixos múltiplos L2VPN VPL anunciados pelo roteador de PE para um VFI

É possível que um PE pòde precisar de anunciar cervejas pretas fortes do rótulo múltiplo para um exemplo do forwarding virtual (VFI).

Se o VE-ID do PE remoto não cai na escala anunciada pelo PE local, o PE remoto não pode escolher uma etiqueta remota para o picowatt. Este cálculo, descrito mais cedo, é  $\leq VBO \text{ VE-ID} < VBO + VB$ .

Se esta verificação falha, o VE-ID do PE remoto é fora da escala. O PE remoto ignora o prefixo recebido do PE local. O PE local aprende que o PE remoto é fora da escala quando recebe o prefixo que o PE remoto está anunciando. O PE local precisa de determinar que etiqueta remota a se usar para esse roteador de PE remoto. O PE local igualmente envia um prefixo novo, segundo para um bloco novo de etiquetas do local ao PE remoto, que o PE remoto deve poder usar a fim seleccionar uma etiqueta remota.

## Configuração PE1

O exemplo anterior é continuado aqui; O PE1 ainda tem:

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
```

```
Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
AGI version(0), VE Block Size(50) Label Base(10000)
Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
rx pathid: 0, tx pathid: 0x0
```

## Configuração PE2

O PE2 tem agora um VE-ID de 1002 e esta configuração:

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  AGI version(0), VE Block Size(50) Label Base(10000)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  rx pathid: 0, tx pathid: 0x0
```

## Blocos iniciais da etiqueta

PE1 e começo PE2 com estes blocos iniciais da etiqueta.

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  AGI version(0), VE Block Size(50) Label Base(10000)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  rx pathid: 0, tx pathid: 0x0
```

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000
BGP routing table entry for 1:100:VEID-1001:Blk-1000/136, version 3
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
Advertised to update-groups:
 1
Refresh Epoch 1
Local
 0.0.0.0 from 0.0.0.0 (10.100.1.1)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  AGI version(0), VE Block Size(50) Label Base(10000)
  Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
  rx pathid: 0, tx pathid: 0x0
```

## Troca PE1 e PE2

Use o comando das atualizações dos vpls BGP l2vpn debugar a fim rever a troca PE1 e PE2, a seguir use o comando rd 1:100 dos vpls BGP l2vpn da mostra a fim rever detalhes.

```

PE1#
%BGP-5-ADJCHANGE: neighbor 10.100.1.4 Up
BGP(9): update formatted for 1:100:VEID-1001:Blk-1000:VBS-50:LB-10000/136 VE ID
1001 VE Block Offset 1000 VE Block Size 50 Label Base 10000 /136
BGP(9): (base) 10.100.1.4 send UPDATE (format) 1:100:VEID-1001:Blk-1000:VBS-50:
LB-10000/136, next 10.100.1.1, metric 0, path Local, extended community RT:1:100
RT:32:64 L2VPN L2:0x0:MTU-1500
BGP(9): 10.100.1.4 rcvd UPDATE w/ attr: nexthop 10.100.1.2, origin ?,
localpref 100, metric 0, originator 10.100.1.2, clusterlist 10.100.1.4, extended
community RT:1:100 L2VPN L2:0x0:MTU-1500
BGP(9): 10.100.1.4 rcvd 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136
BGP(9): bump net 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136, non bpath added
BGP(9): nettable_walker called for 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136
BGP(9): best path[0] 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136 source
10.100.1.1 nh 10.100.1.2 vpls-id: L2VPN L2:0x0:MTU-1500
BGP(9): add XC RIB route 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136 masklen 136
L2VPN L2:0x0:MTU-1500 pathcount: 1 [0] LDP source:10.100.1.1 nexthop:10.100.1.2
RT:1:100
BGP(9): bump net 1:100:VEID-1001:Blk-10000:VBS-50:LB-10053/136, non bpath added
BGP(9): nlri update add VBS 50 LB 10053
BGP(9): nlri update add export extcomm count 4
BGPSSA ssaccount is 0
BGP(9): update formatted for 1:100:VEID-10002:Blk-10000:VBS-50:LB-3000/136 VE ID
10002 VE Block Offset 10000 VE Block Size 50 Label Base 3000 /136
BGP(9): nettable_walker called for 1:100:VEID-1001:Blk-10000:VBS-50:LB-10053/136
BGP(9): nettable_walker 1:100:VEID-1001:Blk-10000:VBS-50:LB-10053/136 route sourced
locally
BGP(9): update formatted for 1:100:VEID-1001:Blk-10000:VBS-50:LB-10053/136 VE ID
1001 VE Block Offset 10000 VE Block Size 50 Label Base 10053 /136
BGP(9): (base) 10.100.1.4 send UPDATE (format) 1:100:VEID-1001:Blk-10000:VBS-50:
LB-10053/136, next 10.100.1.1, metric 0, path Local, extended community RT:1:100
RT:32:64 L2VPN L2:0x0:MTU-1500 L2VPN L2:0x0:MTU-1500
BGP(9): 10.100.1.4 rcvd UPDATE w/ attr: nexthop 10.100.1.2, origin ?, localpref 100,
metric 0, originator 10.100.1.2, clusterlist 10.100.1.4, extended community
RT:1:100 L2VPN L2:0x0:MTU-1500
BGP(9): 10.100.1.4 rcvd 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136
BGP(9): bump net 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136, non bpath added
BGP(9): nettable_walker called for 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136
BGP(9): best path[0] 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136 source 10.100.1.1
nh 10.100.1.2 vpls-id: L2VPN L2:0x0:MTU-1500
BGP(9): add XC RIB route 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136 masklen 136
L2VPN L2:0x0:MTU-1500 pathcount: 1 [0] LDP source:10.100.1.1 nexthop:10.100.1.2
RT:1:100
BGP(9): update formatted for 1:100:VEID-10002:Blk-1000:VBS-50:LB-3053/136 VE ID
10002 VE Block Offset 1000 VE Block Size 50 Label Base 3053 /136
BGPSSA ssaccount is 0

```

```
PE1#show bgp l2vpn vpls rd 1:100
```

```

BGP table version is 5, local router ID is 10.100.1.1
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: 1:100
*>  1:100:VEID-1001:Blk-1000/136
           0.0.0.0                               32768 ?
*>  1:100:VEID-1001:Blk-10000/136
           0.0.0.0                               32768 ?
*>i 1:100:VEID-10002:Blk-1000/136

```

```

                10.100.1.2                0    100    0 ?
*>i 1:100:VEID-10002:Blk-10000/136
                10.100.1.2                0    100    0 ?

```

PE2#show bgp l2vpn vpls rd 1:100

```

BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

## Análise da troca PE1 e PE2

O PE1 e o PE2 têm anunciado agora dois que a etiqueta obstrui cada um entre si.

O PE1 anuncia primeiramente uma atualização BGP inicial ao PE2:

PE2#show bgp l2vpn vpls rd 1:100

```

BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

Esta atualização tem o NLRI ajustado de acordo com a configuração no PE1.

O PE1 recebe então a atualização BGP inicial do PE2.

PE2#show bgp l2vpn vpls rd 1:100

```

BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

O PE2 anuncia o prefixo inicial com os valores VE-ID 10002, VBO = 10000, VB = 50 pés, LB = 3000.

O PE1 observa que o PE2 é fora da escala desde que o PE1 partiu com bloco LB da etiqueta (LB + VB - 1) ou a 10000 (10000 + 50 pés - 1) = a 10049.

O PE1 deve determinar se o VBO está dentro da escala de sua configuração. Assim, o VE-ID do PE2 precisa de ser verificado contra a escala anunciada pelo PE1. O cálculo é  $\leq \text{VBO VE-ID} < \text{VBO} + \text{VB}$ . Neste caso,  $1000 \leq 10002 < 1000 + 50$  pés, que não é verdadeiro. Assim, o PE1 precisa de enviar um bloco novo da etiqueta a fim acomodar o VE-ID fora de alcance do PE2. Em reação à atualização inicial do PE2, o PE1 formata e envia uma atualização BGP nova, adicional ao PE2. O PE1 usa agora um VBO novo de 10000.

```
PE2#show bgp l2vpn vpls rd 1:100
BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

Para o PE1, o VBO é 10000, VB é 50 pés, LB é 10053. A verificação para o PE2 é  $\leq \text{VBO VE-ID} < \text{VBO} + \text{VB}$ . Neste caso,  $10000 \leq 10002 < 10000 + 50$  pés, que é verdadeiro. O PE2 pode escolher uma etiqueta remota deste bloco novo [10053 da etiqueta - 10102] do PE1. Ou seja o PE1 adicionou um bloco novo da etiqueta a fim acomodar o PE2 e enviou duas mensagens da atualização BGP.

O mesmo acontece na direção oposta. O PE2 recebe a atualização BGP inicial do PE1. Esta atualização tem estes valores VE-ID 1001, VBO = 1000, VB = 50 pés, LB = 10000.

O PE2 observa que VE-ID do PE1 é fora de alcance com atualização da inicial do PE2. A verificação PE1's é  $\leq \text{VBO VE-ID} < \text{VBO} + \text{VB}$  ou  $10000 \leq 1001 < 10000 + 50$  pés. Na resposta, o PE2 envia esta segunda atualização BGP, com um bloco novo [3053 da etiqueta - 3102] que acomoda o VE-ID de 1001 do PE1 porque a verificação PE1's é  $\leq \text{VBO VE-ID} < \text{VBO} + \text{VB}$  ou  $1000 \leq 1001 < 1000 + 50$  pés.

PE2#show bgp l2vpn vpls rd 1:100

BGP table version is 6, local router ID is 10.100.1.2  
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

## Detalhes do prefixo

Estes são os detalhes dos dois prefixos originados pelo PE1:

PE2#show bgp l2vpn vpls rd 1:100

BGP table version is 6, local router ID is 10.100.1.2  
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

PE2#show bgp l2vpn vpls rd 1:100

BGP table version is 6, local router ID is 10.100.1.2  
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

Aqui, dois roteadores de PE têm esquemas descontiguos do número, que faz com que cada PE mande duas atualizações BGP. Se há muitos roteadores de PE com esquemas descontiguos do número, o número de atualizações BGP cresce rapidamente muito grande.

[www.cisco.com](http://www.cisco.com) diz: “Por exemplo, as sequências da numeração VE-ID tais como 1, o 2,3 ou 501, 502, 503 são bons porque os VE-ID são contíguos. Um esquema de numeração tal como 100, 200, 300 é ruim porque é NON-contíguo.”

Os primeiros exemplos de 1, de 2,3 ou de 501, 502, 503 são números contíguos, assim que cada roteador de PE precisa de enviar somente um prefixo L2VPN VPL. Com o terceiro exemplo (100, 200, 300), cada PE tem que mandar muitos prefixos L2VPN VPL. Para números NON-contíguos, uma grande bastante VE varia manteria o número de prefixos a ser anunciados mais baixo. Contudo, a quantidade de etiquetas (desperdiçadas) reservados é ainda maior.

## Interoperabilidade

Se o refletor da rota de BGP (RR) executa o software que não compreende o RFC 4761, mas tem o apoio para o RFC 4762, o **comando configuration** especial do prefixo-comprimento-**tamanho 2** do vizinho de BGP **x.x.x.x** está precisado no RR assim que pode refletir as atualizações BGP usadas para o RFC 4761.

Os prefixos são enviados geralmente com um comprimento de 1 byte. O Cisco IOS Software executou o esboço 'draft-ietf-l2vpn-signaling-08, que se transformou mais tarde RFC 6074. Um campo de comprimento de 1 byte foi escolhido naquele tempo, indicando o comprimento nos bit.

[O abastecimento, a descoberta automática, e a sinalização do RFC 6074 nas redes privadas virtuais da camada 2 \(L2VPN\)](#) especificam que a codificação NLRI para a descoberta automática BGP deve estar a um comprimento de 2 bytes. Os 2 bytes indicam quantos bytes do prefixo seguem no prefixo do comprimento variável.

Secione 7 do RFC 6074, “Interoperabilidade BGP-AD e VPLS-BGP,” estados:

“BGP-AD e VPLS-BGP [RFC4761] usam o mesmo AFI/SAFI. Para que BGP-AD e VPLS-BGP coexista, o comprimento NLRI deve ser usado como um desmultiplexador.

O BGP-AD NLRI tem um comprimento NLRI de 12 bytes, contendo somente um 8-byte RD e um 4-byte VSI-ID. VPLS-BGP [RFC4761] usa um comprimento 17-byte NLRI. Consequentemente, as aplicações de BGP-AD devem ignorar o NLRI que são maiores de 12 bytes.”

Se o **comando 2 do prefixo-comprimento-tamanho do vizinho x.x.x.x** não está atual nos RR, o vizinho de BGP não vem acima, e o RR interpreta o campo de comprimento como 1 byte somente. Esta notificação aparece no RR:

```
PE2#show bgp l2vpn vpls rd 1:100
BGP table version is 6, local router ID is 10.100.1.2
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: 1:100
*>i 1:100:VEID-1001:Blk-1000/136
      10.100.1.1          0    100    0 ?
*>i 1:100:VEID-1001:Blk-10000/136
      10.100.1.1          0    100    0 ?
*> 1:100:VEID-10002:Blk-1000/136
      0.0.0.0              32768 ?
*> 1:100:VEID-10002:Blk-10000/136
      0.0.0.0              32768 ?

```

Esta notificação aparece no roteador de PE:

```

PE2#show bgp l2vpn vpls rd 1:100
BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?

Isto ocorre porque, na implementação original da descoberta automática BGP no Cisco IOS Software, o campo de comprimento era 1 byte.

Se você põe o comando 2 do prefixo-comprimento-tamanho do vizinho x.x.x.x sobre o RR, as notificações não aparecem.

```

PE2#show bgp l2vpn vpls rd 1:100
BGP table version is 6, local router ID is 10.100.1.2
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: 1:100					
*>i 1:100:VEID-1001:Blk-1000/136	10.100.1.1	0	100	0	?
*>i 1:100:VEID-1001:Blk-10000/136	10.100.1.1	0	100	0	?
*> 1:100:VEID-10002:Blk-1000/136	0.0.0.0			32768	?
*> 1:100:VEID-10002:Blk-10000/136	0.0.0.0			32768	?