

VPLS avec la note en tech de signalisation BGP

Contenu

[Introduction](#)

[Problème](#)

[Architecture de la solution](#)

[Identification de point final de routeur PE](#)

[Identité et mpls label VPLS](#)

[Les informations d'encapsulation](#)

[Préfixe de détection automatique BGP VPLS et préfixe de signalisation BGP VPLS](#)

[Configuration du logiciel de Cisco IOS d'échantillon](#)

[Bloc annoncé d'étiquette](#)

[Cibles de moteur de distinction de route et d'artère](#)

[Exemple de configuration de PE1](#)

[Chaîne d'étiquette de contrôle](#)

[Étiquettes de contrôle](#)

[Bloc d'étiquette de contrôle](#)

[Le contrôle a annoncé le préfixe](#)

[Préfixe de vue en détail](#)

[Annoncez, recevez, et des blocs d'étiquette de processus dans des messages de mise à jour BGP](#)

[PE2 : Recevez la mise à jour BGP](#)

[PE2 : Trouvez une étiquette](#)

[PE2 : Envoyez le préfixe à PE1](#)

[PE1 : Trouvez une étiquette](#)

[Commandes supplémentaires de vérification](#)

[Plusieurs préfixes du L2VPN VPLS annoncés par le routeur PE pour un VFI](#)

[Configuration PE1](#)

[Configuration PE2](#)

[Blocs initiaux d'étiquette](#)

[Échange PE1 et PE2](#)

[Analyse de l'échange PE1 et PE2](#)

[Détails de préfixe](#)

[Interopérabilité](#)

Introduction

Ce document décrit la détection automatique basée sur de Protocole BGP (Border Gateway Protocol) pour un service privé virtuel de RÉSEAU LOCAL (VPLS) avec la signalisation BGP. La détection automatique est des moyens pour un Provider Edge (PE) d'apprendre que le siège potentiel d'explosion distant sont des membres d'un domaine donné VPLS. La signalisation est

des moyens pour qu'un PE apprenne l'étiquette de pseudowire prévue par un PE distant donné pour un domaine donné VPLS.

Référez-vous à ces documents d'Internet Engineering Task Force :

- [Le service privé virtuel de RÉSEAU LOCAL RFC 4762 \(VPLS\) utilisant la signalisation du protocole de distribution d'étiquette \(LDP\)](#) décrit la détection automatique BGP avec la signalisation du protocole de distribution d'étiquette (LDP) pour VPLS (également connu sous le nom de Martini).
- [Le service privé virtuel de RÉSEAU LOCAL RFC 4761 \(VPLS\) utilisant le BGP pour la détection automatique et la signalisation](#) décrit la détection automatique BGP et la signalisation BGP pour VPLS (également connu sous le nom de Kompella).

Ce document se concentre sur RFC 4761. Avec RFC 4761, les informations d'accessibilité de couche de réseau BGP (NLRI) des mises à jour BGP tiennent les informations pour la détection automatique et la signalisation. Quand les Routeurs distants de PE reçoivent cette mise à jour BGP, ils ont toutes les informations nécessaires afin d'installer un maillage complet des pseudowires pour VPLS. La détection automatique BGP et la signalisation BGP utilisent le même address-family BGP.

L'interface de ligne de commande (CLI) et la sortie est de logiciel de Cisco IOS®. La configuration et la fonctionnalité est très semblable dans le Logiciel Cisco IOS XR et le Logiciel Cisco NX-OS.

Problème

VPLS se compose d'un ensemble de pseudowires (picowatt) d'une mode point-à-multipoint. Jusqu'ici, le LDP a été utilisé pour signaler les pseudowires entre les Routeurs de PE. Ainsi, une session visée LDP a signalé quelles étiquettes aux utiliser pour quel pseudowire entre une paire de Routeurs de PE. Vous pourriez manuellement configurer l'ensemble de Routeurs de PE qui ont participé à un domaine VPLS, ou vous pourriez employer le BGP pour découvrir la configuration automatiquement. Afin d'exécuter cette détection automatique, le BGP a annoncé quel PE était un membre dont domaine VPLS. Cependant, même avec la détection automatique BGP, le LDP a été utilisé pour signaler les étiquettes de circuit virtuel de Commutation multiprotocole par étiquette (MPLS) (circuit virtuel) et l'ID de pseudowire.

Il est maintenant possible d'employer le BGP afin de signaler les pseudowires entre les Routeurs de PE.

Quand un pseudowire doit être installé entre une paire de Routeurs, les autres Routeurs n'ont pas besoin du relatif à l'information à ce pseudowire. Par exemple, une telle informations sont l'étiquette de circuit virtuel à utiliser.

Avec le LDP comme protocole de signalisation pour installer des pseudowires, les informations sont seulement reçues par les paires de Routeurs, parce que le LDP fait la signalisation d'une mode point par point.

Avec le BGP comme protocole de signalisation pour installer des pseudowires, les informations sont reçues par tous autres Routeurs parce que BGP interne (iBGP) fait la signalisation d'une mode point-à-multipoint. l'iBGP a une condition requise de maillage complet, ainsi un routeur envoie une mise à jour d'iBGP à tous autres Routeurs d'iBGP. Ceci a pu également être fait avec un réflecteur d'artère.

Avec l'iBGP comme protocole de signalisation, il y aurait deux méthodes pour envoyer des mises à jour :

1. Chaque routeur PE annonce une mise à jour BGP à tous les voisins d'iBGP pour chaque picowatt ; chaque fois que, une étiquette de circuit virtuel MPLS est reliée. Ainsi, un routeur PE enverrait autant de mises à jour BGP car il y a des Routeurs de PE. Cependant, l'étiquette de circuit virtuel reliée à la mise à jour BGP a pu être utilisée par seulement une des Routeurs de PE - le routeur PE à l'autre bout du picowatt.
2. Pour éviter cette question d'un nombre élevé de mises à jour BGP, une architecture a été conçue par lequel un routeur PE de gens du pays envoie un positionnement ou un bloc d'étiquettes locales de circuit virtuel à tous les Routeurs distants de PE. Chaque routeur PE distant sélectionne une des étiquettes de circuit virtuel pour l'utiliser comme étiquette distante de circuit virtuel pour le picowatt vers le routeur PE local. Le routeur PE distant doit sélectionner une étiquette distante de circuit virtuel d'une seule mode de sorte qu'aucun autre routeur PE ne sélectionne la même étiquette de circuit virtuel du bloc annoncé d'étiquettes. Puisqu'un bloc d'étiquettes est envoyé, il doit y avoir assez d'étiquettes disponibles afin de servir tout le PWs possible qui pourrait être installé, mais il ne devrait pas y avoir tant d'étiquettes réservées qu'ils sont inutilisés et ont gaspillé.

Ce document décrit comment le BGP est utilisé afin de signaler les pseudowires ; notez que le BGP est également utilisé pour la détection automatique en même temps.

Architecture de la solution

Puisque c'est VPLS, il y a toujours un protocole de signalisation de saut par saut requis au centre afin de porter les paquets étiquetés du PE au routeur PE. Cette fonction de transport au centre doit encore être accomplie par LDP ou Ingénierie de trafic MPLS.

Le BGP doit envoyer les informations nécessaires afin d'installer les pseudowires d'une mode point-à-multipoint requise par VPLS. Ces informations de signalisation incluent :

- Identification de point final de routeur PE
- Identité VPLS
- Bloc de mpls label
- Les informations d'encapsulation

Identification de point final de routeur PE

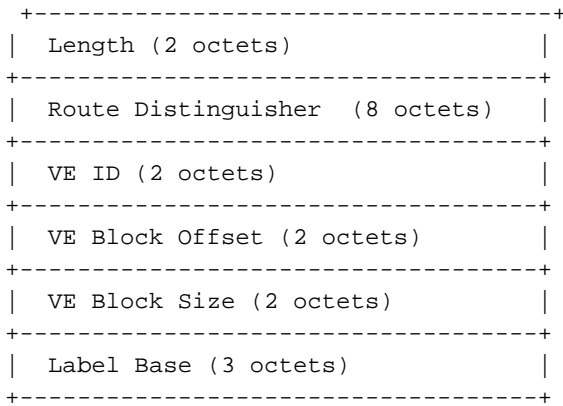
L'identification de point final de routeur PE est déterminée du routeur PE qui est l'expéditeur BGP de la mise à jour.

La mise à jour BGP concernant des réseaux privés virtuels de la couche 2 (L2VPN) VPLS est identifiée par AFI/SAFI 25/65. Cette famille d'adresse est négociée quand le BGP envoie le message OUVERT.

Identité et mpls label VPLS

Le NLRI, également connu sous le nom de préfixe, tient les informations sur l'identité VPLS et le

bloc de mpls label. Son codage a une longueur totale de 19 octets :



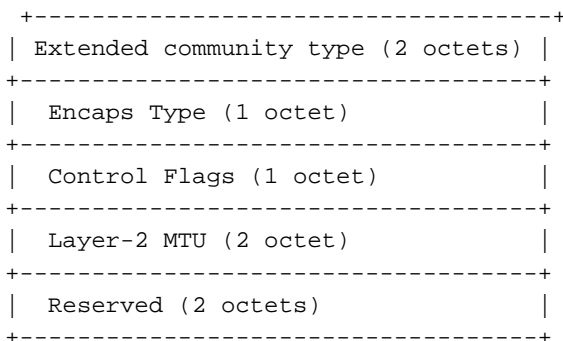
Le moteur de distinction de route (RD) associe à l'identité du VPLS.

Remarque: Dans l'implémentation de Cisco IOS et de Logiciel Cisco NX-OS, tous les Routeurs de PE doivent avoir le même RD dans le même domaine VPLS.

L'ID virtuel de l'extension (VE), le VE bloquent le décalage, longueur de bloc du VE, et la base d'étiquette (livre) associent au bloc annoncé d'étiquettes, comme expliqué dans la section suivante.

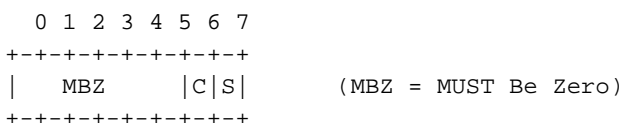
Les informations d'encapsulation

Les informations d'encapsulation sont également reliées au préfixe et sont encodées en tant que communauté étendue 'Layer2 Community étendu par informations à la mise à jour BGP. La valeur est 0x800A et est encodée en tant que :



Le type d'encaps pour VPLS est 19.

Les indicateurs de contrôle (vecteur de bit) est encodés de cette façon :



Nom Valeur Signification

- | | | |
|---|---|--|
| C | 1 | Un mot de commande DOIT être présent où des paquets VPLS sont envoyés à ce PE. |
| | 0 | Un mot de commande NE DOIT PAS être présent où des paquets VPLS sont envoyés à ce PE. |
| S | 1 | La livraison ordonnancée des trames DOIT être utilisée quand des paquets VPLS sont envoyés à ce PE. |
| | 0 | La livraison ordonnancée des trames NE DOIT PAS être utilisée quand des paquets VPLS sont envoyés à ce PE. |

envoyés à ce PE.

Il y a également des cibles d'artère (rts) reliées à la mise à jour BGP. Le rts contrôle l'importation dans et exporte du L2VPN, de la même manière que MPLS L3VPN.

Préfixe de détection automatique BGP VPLS et préfixe de signalisation BGP VPLS

Un préfixe de détection automatique BGP VPLS est un préfixe de /96, tandis qu'un préfixe de signalisation BGP VPLS est un préfixe de /136. Ce sont des exemples de chacun :

```
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
```

Configuration du logiciel de Cisco IOS d'échantillon

C'est une configuration du logiciel de Cisco IOS d'échantillon :

```
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
```

Bloc annoncé d'étiquette

Un routeur PE doit annoncer au moins un bloc d'étiquette. Le bloc d'étiquette est un ensemble continu de mpls label et est utilisé par les Routeurs distants de PE afin de sélectionner une étiquette de circuit virtuel de distant. L'étiquette distante est utilisée pour le picowatt entre le routeur PE local et distant. (Un routeur PE peut annoncer de plusieurs blocs d'étiquette, comme expliqué dans les sections postérieures.)

Le VE-ID doit être configuré sur chaque PE. Il identifie les Routeurs de PE dans le domaine VPLS.

La longueur de bloc du VE (VBS) est la taille du bloc d'étiquette et a une valeur par défaut de 10. Si le « VE s'étendent » est configuré, il est cette valeur. le « VE s'étendent » peut être configuré pour être [11 -100].

La base d'étiquette (livre) est la première valeur d'étiquette d'un ensemble libre d'étiquettes qui peuvent être réservées par le routeur PE à utiliser pour ce domaine VPLS.

Le décalage de bloc du VE (VBO) est la valeur excentrée à utiliser quand de plusieurs blocs d'étiquette doivent être créés par un routeur PE. VBO est calculé avec cette équation : $VBO = RND(VE-ID/VBS) * VBS$

Ce sont des calculs d'exemple :

- Si VBS = 8 et VE-ID = 2, VBO = $\text{RND}(2/8) * 8 = 1$
- Si VBS = 8 et VE-ID = 20, VBO = $\text{RND}(20/8) * 8 = 16$
- Si VBS = 50 et VE-ID = 199, VBO = $\text{RND}(199/50) * 50 = 150$
- Si VBS = 50 et VE-ID = 1002, VBO = $\text{RND}(1002/50) * 50 = 1000$

Le bloc d'étiquette annoncé aux Routeurs distants de PE est {livre, livre + 1 ? , LIVRE + VBS - 1}. Le bloc d'étiquette est défini par livre et le VBS ; les débuts de bloc à livre et extrémités avec (livre + VBS - 1).

De plusieurs blocs d'étiquette peuvent être créés par chaque routeur PE, une fois nécessaires. Le routeur doit s'assurer que c'est un ensemble continu d'étiquettes libres.

Cibles de moteur de distinction de route et d'artère

Exemple de configuration de 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
```

C'est une explication des valeurs de configuration :

- Le VPN ID est configuré en tant que 100.
- Le RD est pris de [ASN : vpn id], à moins qu'un RD soit explicitement configuré. Ici, le RD est 1:100.
- Les cibles d'artère d'importation/exportation sont 32:64.
- Livre est de la plage [10000 20000]. La valeur précise de livre dépend du premier ensemble d'étiquettes locales continues libres qui est assez grand pour juger toutes les étiquettes déterminées par le VBS.
- Le VE-ID est configuré en tant que 1001.
- Le VBS est configuré en tant que 50.
- Le VBO est calculé pour être : $\text{VBO} = \text{RND}(\text{VE-ID}/\text{VBS}) * \text{VBS}$ ou $\text{RND}(1001/50) * 50 = 1000$.

Chaîne d'étiquette de contrôle

Vous pouvez vérifier la plage d'étiquette avec la commande de **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
```

Il y a plage d'étiquette par défaut par la plate-forme, que vous pouvez changer avec la commande de `mpls label range`.

Étiquettes de contrôle

Vous pouvez vérifier les étiquettes utilisées par effectif pour un bloc d'étiquette dans le Forwarding Information Base d'étiquette (LFIB) avec la commande de `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
```

Dans cet exemple, PE1, le routeur local, réservé 50 étiquettes locales pour le bloc d'étiquette. 'lbl-blk-id(1:0)' signifie un bloc-id de 1 et un bloc-exemple de 0, qui identifie la première étiquette du bloc. La dernière étiquette de ce bloc est l'étiquette 10049.

L'interface « sortante » dans le LFIB est « baisse » tant que il n'y a aucun picowatt installé pour cette étiquette locale. Si un picowatt est installé, l'interface « sortante » n'en est aucune point2point.

Bloc d'étiquette de contrôle

Le bloc assigné d'étiquette peut également être vérifié avec la commande **récapitulative de bloc-base de données de lfd d'infrastructure de MPLS d'exposition** quand le « service interne » est configuré.

```
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
```

Livre est 10000. Dans cet exemple, le bloc d'étiquette est de livre à (livre + VBS - 1) ou de 10000 à (10000 + 50 - 1) = 10049.

Le contrôle a annoncé le préfixe

Vous pouvez vérifier le préfixe annoncé avec la commande du rd 1:100 de vpls de l2vpn de `show bgp` :

```
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
```


Préfixe de vue en détail

Pour visualiser ce préfixe en détail, utilisez la commande du bloc-décalage 1000 du VE-id 1001 du rd 1:100 de vpls de l2vpn de show bgp. Notez que vous spécifiez le VE-ID et le bloc d'étiquette, qui peuvent être trouvés dans le 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
```

NLRI affiche le RD de 1:100, le VE-ID de 1001, le VBO de 1000, le VBS de 50, et la livre de 10000.

La communauté étendue de l'information Layer2 tient ces informations :

- Le type d'encap est 19 (VPLS)
- Indicateurs de contrôle : C = 0 (une commande mot ne doit pas être placée) ; S = 0 (aucune livraison ordonnancée des trames)
- Le MTU est 1500

La communauté étendue droite tient ces informations :

- Droite 1:100
- Droite 32:64

Remarque: Le par défaut VBS (10) est petit de sorte que des étiquettes locales ne soient pas gaspillées.

Annoncez, recevez, et des blocs d'étiquette de processus dans des messages de mise à jour BGP

Quand un routeur PE local annonce un bloc de préfixe/étiquette du L2VPN VPLS, chaque routeur PE distant doit essayer de sélectionner une étiquette de cette plage afin de l'utiliser comme étiquette distante de circuit virtuel.

- Si le routeur PE distant réussit, il l'utilise qu'étiquette distante de circuit virtuel et la programme dans le plan de données. Il n'y a aucune autre signalisation par BGP.
- Si le routeur PE distant échoue, il doit attendre un autre préfixe du L2VPN VPLS à annoncer par le routeur PE, alors l'essai locaux pour sélectionner une autre étiquette distante de circuit virtuel de ce bloc d'étiquette.

Supposez que PE1 est un PE local avec la configuration précédente et que PE2 est un PE distant avec cette configuration :

```
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 : Recevez la mise à jour BGP

PE2 reçoit cette mise à jour BGP de 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 : Trouvez une étiquette

PE2 doit trouver une étiquette qu'il peut l'utiliser comme étiquette distante de circuit virtuel pour le picowatt vers PE1.

PE2 doit d'abord déterminer si le VBO est dans la marge de sa configuration. PE2 vérifie son VE-ID contre la plage annoncée par PE1 avec le \leq du calcul $VBO \text{ VE-ID} < VBO + VBS$. Dans ce cas, 1000 le $\leq 1002 < 1000 + 50$, ainsi le PE2 réussit.

PE2 doit alors sélectionner une étiquette distante de circuit virtuel. L'étiquette de démultiplexeur (circuit virtuel) à utiliser par le PE distant est calculée en tant que (livre + VE-ID - VBO).

Du préfixe plus tôt, livre est 10000, et VBO est 1000. Le VE-ID est celui de PE2 et est 1002. Ainsi, étiquette des sélections PE2 (livre + VE-ID - VBO) = (10000 + 1002 - 1000) = 10002.

Employez la commande du nom un de vfi de l2vpn d'exposition afin de vérifier ceci :

```
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 : Envoyez le préfixe à PE1

PE2 envoie alors son préfixe à 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 : Trouvez une étiquette

PE1 est maintenant le PE et les besoins distants de trouver une étiquette qu'il peut l'utiliser comme étiquette distante de circuit virtuel pour le picowatt vers PE2.

PE1 doit d'abord déterminer si le VBO est dans la marge de sa configuration. PE1 vérifie son VE-ID contre la plage annoncée par PE2 avec le \leq du calcul $VBO_{VE-ID} < VBO + VBS$. Dans ce cas, 1000 le \leq $1001 < 1000 + 50$, ainsi le PE1 réussit.

PE1 doit alors sélectionner une étiquette distante de circuit virtuel. L'étiquette de démultiplexeur (circuit virtuel) à utiliser par le PE distant est calculée en tant que $(livre + VE-ID - VBO)$.

Du préfixe plus tôt, livre est 3100, et VBO est 1000. Le VE-ID est celui de PE1 et est 1001. Ainsi, étiquette des sélections PE1 $(livre + VE-ID - VBO) = (3100 + 1001 - 1000) = 3101$.

Employez la commande du nom un de vfi de l2vpn d'exposition afin de vérifier ceci :

```
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
```

Commandes supplémentaires de vérification

```
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: 0x0PE1#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: 0x0PE1#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: 0x0PE1#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: 0x0PE1#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: 0x0PE1#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

```

Plusieurs préfixes du L2VPN VPLS annoncés par le routeur PE pour un VFI

Il est possible qu'un PE pourrait devoir annoncer de plusieurs bières brunes d'étiquette pour une instance de transfert virtuelle (VFI).

Si le VE-ID du PE distant ne tombe pas dans la plage annoncée par le PE local, le PE distant ne peut pas sélectionner une étiquette distante pour le picowatt. Ce calcul, décrit plus tôt, est $\leq VBO$ $VE-ID < VBO + VBS$.

Si ce contrôle échoue, le VE-ID du PE distant est hors de plage. Le PE distant ignore le préfixe reçu du PE local. Le PE local apprend que le PE distant est hors de plage quand il reçoit le préfixe que le PE distant annonce. Le PE local doit déterminer ce qui étiquette distante à l'utiliser pour ce routeur PE distant. Le PE local envoie également un nouveau, deuxième préfixe pour un nouveau bloc d'étiquettes de gens du pays au PE distant, que le PE distant devrait pouvoir employer afin de sélectionner une étiquette distante.

Configuration PE1

L'exemple précédent est continué ici ; PE1 a toujours :

```
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
```

Configuration PE2

PE2 a maintenant un VE-ID de 1002 et cette configuration :

```
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
```

Blocs initiaux d'étiquette

PE1 et début PE2 avec ces blocs initiaux d'étiquette.

```
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: 0x0PE1#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

```

Échange PE1 et PE2

Utilisez les **vpls de l2vpn BGP de débogage** que les mises à jour commandent afin de passer en revue l'échange PE1 et PE2, puis employez la commande du **rd 1:100 de vpls de l2vpn de show bgp** afin de passer en revue des détails.

```

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 OPE1#**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	?

Analyse de l'échange PE1 et PE2

PE1 et PE2 maintenant ont annoncé deux que l'étiquette bloque chacun entre eux.

PE1 annonce d'abord une première mise à jour BGP à 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 ?

```

Cette mise à jour a le NLRI réglé selon la configuration sur PE1.

PE1 reçoit alors la mise à jour BGP initiale de 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	?

PE2 annonce le préfixe initial avec les valeurs VE-ID 10002, VBO = 10000, VBS = 50, livre = 3000.

PE1 note que PE2 est hors de plage puisque PE1 commencé avec le bloc livre d'étiquette à (livre + VBS - 1) ou de 10000 à (10000 + 50 - 1) = 10049.

PE1 doit déterminer si le VBO est dans la marge de sa configuration. Ainsi, le VE-ID de PE2 doit être vérifié contre la plage annoncée par PE1. Le calcul est $\leq VBO \text{ VE-ID} < VBO + VBS$. Dans ce cas, $1000 \leq 10002 < 1000 + 50$, qui n'est pas vrai. Ainsi, PE1 doit envoyer un nouveau bloc d'étiquette afin de faciliter la -de-plage VE-ID de PE2. En réaction à la mise à jour initiale de PE2, PE1 formate et envoie une nouvelle, supplémentaire mise à jour BGP à PE2. PE1 utilise maintenant un nouveau VBO 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 ?

Pour PE1, le VBO est 10000, VBS est 50, livre est 10053. Le vérifieur PE2 est \leq VBO VE-ID < VBO + VBS. Dans ce cas, $10000 \leq 10002 < 10000 + 50$, qui est vrai. PE2 peut sélectionner une étiquette distante de ce nouveau bloc [10053 d'étiquette - 10102] de PE1. En d'autres termes, PE1 a ajouté un nouveau bloc d'étiquette afin de faciliter PE2 et a envoyé deux messages de mise à jour BGP.

Le même se produit dans le sens inverse. PE2 reçoit la mise à jour BGP initiale de PE1. Cette mise à jour a ces valeurs VE-ID 1001, VBO = 1000, VBS = 50, livre = 10000.

PE2 note que VE-ID de PE1 est -de-plage avec la mise à jour d'initiale PE2. Le contrôle PE1?s est \leq VBO VE-ID < VBO + VBS ou $10000 \leq 1001 < 10000 + 50$. Dans la réponse, PE2 envoie cette deuxième mise à jour BGP, avec un nouveau bloc [3053 d'étiquette - 3102] qui facilite le VE-ID de 1001 de PE1 parce que le contrôle PE1?s est \leq VBO VE-ID < VBO + VBS ou $1000 \leq 1001 < 1000 + 50$.

```
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	?

Détails de préfixe

Ce sont les détails des deux préfixes lancés par 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	?

Ici, deux Routeurs de PE ont des schémas discontigus de nombre, qui fait envoyer à chacun le PE deux mises à jour BGP. S'il y a beaucoup de Routeurs de PE avec des schémas discontigus de nombre, le nombre de mises à jour BGP se développe rapidement très grand.

www.cisco.com indique : « Par exemple, les ordres de numérotation VE-ID tels que 1, 2, 3 ou 501, 502, 503 sont bons parce que les VE-id sont contigus. Une structure de numérotation telle que 100, 200, 300 est mauvaise parce qu'elle est non-contiguë. »

Les premiers exemples de 1, de 2, de 3 ou de 501, 502, 503 sont les nombres contigus, ainsi chaque les besoins de routeur PE d'envoyer seulement un préfixe du L2VPN VPLS. Avec le troisième exemple (100, 200, 300), chaque PE doit envoyer beaucoup de préfixes du L2VPN VPLS. Pour les nombres non-contigus, l'assez grand VE s'étendent garderait le nombre de préfixes pour être inférieur annoncé. Cependant, la quantité d'étiquettes (gaspillées) réservées est encore plus grande.

Interopérabilité

Si le réflecteur de route BGP (rr) exécute le logiciel qui ne comprend pas RFC 4761, mais a le soutien de RFC 4762, la commande de **configuration voisine de la préfixe-longueur-taille 2 BGP x.x.x.x** d'offre spéciale est nécessaire sur le rr ainsi elle peut refléter les mises à jour BGP utilisées pour RFC 4761.

Des préfixes sont habituellement envoyés avec une longueur de 1 octet. Le logiciel de Cisco IOS a mis en application l'ébauche 'draft-ietf-l2vpn-signaling-08, qui plus tard est devenue RFC 6074. Un champ de longueur de 1 octet a été choisi alors, indiquant la longueur dans les bits.

[Le ravitaillement, la détection automatique, et la signalisation RFC 6074 dans des réseaux privés virtuels de la couche 2 \(l2vpn\)](#) spécifie que le codage NLRI pour la détection automatique BGP devrait être une longueur de 2 octets. Les 2 octets indiquent combien d'octets de préfixe suivent dans le préfixe de longueur variable.

Section 7 de RFC 6074, « Interopérabilité BGP-AD et VPLS-BGP, » états :

« BGP-AD et VPLS-BGP [RFC4761] utilisent le même AFI/SAFI. Pour que BGP-AD et VPLS-BGP coexiste, la longueur NLRI doit être utilisée comme démultiplexeur.

Le BGP-AD NLRI a une longueur NLRI de 12 octets, contenant seulement un RD 8-byte et un 4-byte VSI-ID. VPLS-BGP [RFC4761] utilise une longueur 17-byte NLRI. Par conséquent, les

réalisations de BGP-AD doivent ignorer NLRI qui sont plus grands que 12 octets. »

Si la **commande de la préfixe-longueur-taille 2 du voisin x.x.x.x** n'est pas présente sur le RRs, le voisin BGP ne monte pas, et le rr interprète le champ de longueur en tant que 1 octet seulement. Cette notification apparaît sur le 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	?

Cette notification apparaît sur le routeur 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	?

Ceci se produit parce que, dans l'implémentation d'origine de la détection automatique BGP en logiciel de Cisco IOS, le champ de longueur était de 1 octet.

Si vous mettez la **commande de la préfixe-longueur-taille 2 du voisin x.x.x.x** sur le rr, les notifications n'apparaissent pas.

```
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 ?