VPLS with BGP Signaling Tech Note

Contents

Introduction

This document describes Border Gateway Protocol (BGP)-based auto-discovery for a Virtual Private LAN Service (VPLS) with BGP signaling. Auto-discovery is a means for a provider edge (PE) to learn which remote PEs are members of a given VPLS domain.Signaling is a means for a PE to learn the pseudowire label expected by a given remote PE for a given VPLS domain.

Refer to these Internet Engineering Task Force documents:

- RFC 4762 Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling describes BGP auto-discovery with Label Distribution Protocol (LDP) signaling for VPLS (also known as Martini).
- RFC 4761 Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling describes BGP auto-discovery and BGP signaling for VPLS (also known as Kompella).

This document focuses on RFC 4761. With RFC 4761, the BGP Network Layer Reachability Information (NLRI) of the BGP updates holds the information for both auto-discovery and signaling. When remote PE routers receive this BGP update, they have all the information necessary in order to set up a full mesh of pseudowires for VPLS. BGP auto-discovery and BGP signaling use the same BGP address-family.

The command-line interface (CLI) and output is from Cisco IOS® Software. The configuration and functionality is very similar in Cisco IOS-XR software and Cisco NX-OS software.

Problem

VPLS consists of a set of pseudowires (PW) in a point-to-multipoint fashion. Until now, LDP was used to signal the pseudowires between the PE routers. So, a targeted LDP session signaled which labels to use for which pseudowire between one pair of PE routers. You could manually configure the set of PE routers that participated in one VPLS domain, or you could use BGP to discover the configuration automatically. In order to perform this auto-discovery, BGP advertised which PE was a member of which VPLS domain. However, even with BGP auto-discovery, LDP was used to signal the Multiprotocol Label Switching (MPLS) virtual circuit (VC) labels and pseudowire ID.

It is now possible to use BGP in order to signal the pseudowires between the PE routers.

When one pseudowire is to be set up between a pair of routers, the other routers do not need the information related to this pseudowire. For example, such information is the VC label to be used.

With LDP as the signaling protocol to set up pseudowires, the information is only received by the pair of routers, because LDP does the signaling in a point-to-point fashion.

With BGP as the signaling protocol to set up pseudowires, the information is received by all other routers because internal BGP (iBGP) does the signaling in a point-to-multipoint fashion. iBGP has a full mesh requirement, so one router sends an iBGP update to all other iBGP routers. This could also be done with a route reflector.

With iBGP as the signaling protocol, there would be two methods to send updates:

1. Each PE router advertises one BGP update to all iBGP neighbors for each PW; each time, one MPLS VC label is attached. Thus, one PE router would send as many BGP updates as there are PE routers. However, the VC label attached to the BGP update could be used by only one of the PE routers - the PE router at the other end of the PW.
2. To avoid this issue of a high number of BGP updates, an architecture was designed whereby one local PE router sends a set or a block of local VC labels to all remote PE routers. Each remote PE router picks one of the VC labels to use as a remote VC label for the PW toward

the local PE router. The remote PE router must pick a remote VC label in a unique fashion so that no other PE router picks the same VC label from the advertised block of labels. Since a block of labels is sent, there must be enough labels available in order to serve all possible PWs that could be set up, but there should not be so many labels reserved that they are unused and wasted.

This document describes how BGP is used in order to signal the pseudowires; note that BGP is also used for auto-discovery at the same time.

Architecture of the Solution

Because this is VPLS, there is still a hop-by-hop signaling protocol needed in the core in order to carry the labeled packets from PE to PE router. This transport function in the core must still be fulfilled by LDP or MPLS traffic engineering.

BGP needs to send the necessary information in order to set up the pseudowires in a point-to-multipoint fashion needed by VPLS. This signaling information includes:

- PE router endpoint identification
- VPLS identity
- Block of MPLS labels
- Encapsulation information

PE Router Endpoint Identification

The PE router endpoint identification is determined from the PE router that is the BGP sender of the update.

The BGP update pertaining to Layer 2 Virtual Private Networks (L2VPN) VPLS is identified by AFI/SAFI 25/65. This address family is negotiated when BGP sends the OPEN message.

VPLS Identity and MPLS Labels

The NLRI, also known as the prefix, holds the information on VPLS identity and the block of MPLS labels. Its encoding has a total length of 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)              |
+------------------------------------+
```

The Route Distinguisher (RD) relates to the identity of the VPLS.

> **Note**: In the Cisco IOS and Cisco NX-OS software implementation, all the PE routers must have the same RD within the same VPLS domain.

The virtual expansion (VE) ID, VE Block Offset, VE Block Size, and the Label Base (LB) relate to the advertised block of labels, as explained in the next section.

Encapsulation Information

Encapsulation information is also attached to the prefix and is encoded as an extended community 'Layer2 Info Extended Community' to the BGP update. The value is 0x800A and is encoded as:

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

The Encaps Type for VPLS is 19.

The Control Flags (bit vector) is encoded this way:

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

Name	Value	Meaning
C	1	A control word MUST be present when VPLS packets are sent to this PE.
	0	A control word MUST NOT be present when VPLS packets are sent to this PE.
S	1	Sequenced delivery of frames MUST be used when VPLS packets are sent to this PE.
	0	Sequenced delivery of frames MUST NOT be used when VPLS packets are sent to this PE.

There are also route targets (RTs) attached to the BGP update. RTs control the import into and export from the L2VPN, in the same manner as MPLS L3VPN.

VPLS BGP Auto-Discovery Prefix and VPLS BGP Signaling Prefix

A VPLS BGP auto-discovery prefix is a /96 prefix, whereas a VPLS BGP signaling prefix is a /136 prefix. These are examples of each:

```
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
```

Sample Cisco IOS Software Configuration

This is a sample Cisco IOS software configuration:

```
!
l2vpn vfi context one
 vpn id 100
 autodiscovery bgp signaling bgp     <<< "signaling ldp" would be RFC 4762
 ve id 1001
 ve range 50
 route-target export 32:64
 route-target import 32:64

mpls label range 10000 20000

!
bridge-domain 1
 member Ethernet0/0 service-instance 100
 member vfi one

!
l2 router-id 10.100.1.1
!
```

```
interface Ethernet0/0
 no ip address
 service instance 100 ethernet
 !

!
router bgp 1
 bgp log-neighbor-changes
 neighbor 10.100.1.4 remote-as 1
 neighbor 10.100.1.4 update-source Loopback0
 !
 address-family l2vpn vpls
 neighbor 10.100.1.4 activate
 neighbor 10.100.1.4 send-community extended
 neighbor 10.100.1.4 suppress-signaling-protocol ldp
 exit-address-family
```

Advertised Label Block

One PE router must advertise at least one label block. The label block is a continuous set of MPLS labels and is used by the remote PE routers in order to select one remote VC label. The remote label is used for the PW between the local and remote PE router. (A PE router can advertise multiple label blocks, as explained in later sections.)

The VE-ID must be configured on each PE. It identifies the PE routers within the VPLS domain.

The VE Block Size (VBS) is the size of the label block and has a default value of 10. If 've range' is configured, it is that value. 've range' can be configured to be [11 -100].

The Label Base (LB) is the first label value of a free set of labels that can be reserved by the PE router to be used for this VPLS domain.

VE Block Offset (VBO) is the offset value to be used when multiple label blocks must be created by a PE router. VBO is calculated with this equation: VBO = RND(VE-ID/VBS) * VBS

These are example calculations:

- If VBS = 8 and VE-ID = 2, VBO = RND(2/8) * 8 = 1
- If VBS = 8 and VE-ID = 20, VBO = RND (20/8) * 8 = 16
- If VBS = 50 and VE-ID = 199, VBO = RND (199/50) * 50 = 150
- If VBS = 50 and VE-ID = 1002, VBO = RND (1002/50) * 50 = 1000

The label block advertised to the remote PE routers is {LB, LB + 1, ? , LB + VBS - 1}. The label block is defined by the LB and the VBS; the block starts at LB and ends with (LB + VBS - 1).

Multiple label blocks can be created by each PE router, when needed. The router must ensure that it is a continuous set of free labels.

Route Distinguisher and Route Targets

Example Configuration of 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
```

This is an explanation of the configuration values:

- The VPN ID is configured as 100.
- The RD is taken from [ASN:vpn id], unless a RD is explicitely configured. Here, the RD is 1:100.
- The import/export route targets are 32:64.
- The LB is from the range [10000 20000]. The exact value of LB depends on the first set of free continuous local labels that is large enough to hold all the labels determined by the VBS.
- The VE-ID is configured as 1001.
- The VBS is configured as 50.
- The VBO is calculated to be: VBO = RND(VE-ID/VBS) * VBS or RND(1001/50) * 50 = 1000.

Check Label Range

You can check the label range with the **show mpls label range** command:

```
PE1#show mpls label range
Downstream Generic label region: Min/Max label: 10000/20000
```

There is default label range by platform, which you can change with the **mpls label range** command.

Check Labels

You can check the actual used labels for one label block in the label forwarding information base (LFIB) with the **show mpls forwarding-table** command.

```
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
```

In this example, PE1, the local router, reserved 50 local labels for the label block. 'lbl-blk-id(1:0)' means a block-id of 1 and a block-instance of 0, which identifies the first label of the block. The last label of this block is label 10049.

The 'Outgoing' interface in the LFIB is 'drop' as long as there is no PW set up for that local label. If a PW is set up, the 'Outgoing' interface is 'none point2point.'

Check Label Block

The assigned label block can also be checked with the **show mpls infrastructure lfd block-database summary** command when 'service internal' is configured.

```
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
```

LB is 10000. In this example, the label block is from LB to (LB + VBS - 1) or from 10000 to (10000 + 50 - 1) = 10049.

Check Advertised Prefix

You can check the advertised prefix with the **show bgp l2vpn vpls rd 1:100** command:

```
PE1#show bgp l2vpn vpls rd 1:100
BGP table version is 3, 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 ?
```

View Prefix in Detail

To view this prefix in detail, use the **show bgp l2vpn vpls rd 1:100 ve-id 1001 block-offset 1000** command. Note that you specify the VE-ID and the label block, which can be found in the 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 shows the RD of 1:100, the VE-ID of 1001, the VBO of 1000, the VBS of 50, and the LB of 10000.

The Layer2 Info Extended Community holds this information:

- Encap type is 19 (VPLS)
- Control flags: C = 0 (a Control Word must not be set); S = 0 (no sequenced delivery of frames)
- MTU is 1500

The RT extended community holds this information:

- RT 1:100
- RT 32:64

> **Note**: The default VBS (10) is small so that local labels are not wasted.

Advertise, Receive, and Process Label Blocks in BGP Update Messages

When a local PE router advertises a L2VPN VPLS prefix/label block, each remote PE router must try to pick one label from that range in order to use as a remote VC label.

- If the remote PE router succeeds, it uses that remote VC label and programs it in the data plane. There is no further signaling by BGP.
- If the remote PE router fails, it must wait for another L2VPN VPLS prefix to be advertised by the local PE router, then try to pick another remote VC label from that label block.

Assume that PE1 is a local PE with the previous configuration and that PE2 is a remote PE with this configuration:

```
l2vpn vfi context one
 vpn id 100
 autodiscovery bgp signaling bgp
 ve id 1002
 ve range 50
!
mpls label range 3000 60000
```

PE2: Receive BGP Update

PE2 receives this BGP update from PE1:

```
PE2#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 5
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
 Not advertised to any peer
 Refresh Epoch 2
 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(10000)
     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: Find a Label

PE2 needs to find a label it can use as a remote VC label for the PW towards PE1.

PE2 must first determine if the VBO is within the range of its configuration. PE2 checks its VE-ID against the range advertised by PE1 with the calculation VBO <= VE-ID < VBO + VBS. In this case, 1000 <= 1002 < 1000 + 50, so PE2 succeeds.

PE2 then needs to pick a remote VC label. The demultiplexer (VC) label to be used by the remote PE is computed as (LB + VE-ID - VBO).

From the earlier prefix, LB is 10000, and VBO is 1000. The VE-ID is the one from PE2 and is 1002. So, PE2 picks label (LB + VE-ID - VBO) = (10000 + 1002 - 1000) = 10002.

Use the **show l2vpn vfi name one** command in order to verify this:

```
PE2#show l2vpn vfi name one
Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No

VFI name: one, state: up, type: multipoint, signaling: BGP
 VPN ID: 100, VE-ID: 1002, VE-SIZE: 50
 RD: 1:100, RT: 1:100
 Bridge-Domain 100 attachment circuits:
 Pseudo-port interface: pseudowire100001
 Interface          Peer Address    VE-ID  Local Label  Remote Label    S
 pseudowire100002   10.100.1.1      1001   3101         10002           Y
```

PE2: Send Prefix to PE1

PE2 then sends its prefix to PE1:

```
PE1#show bgp l2vpn vpls rd 1:100 ve-id 1002 block-offset 1000
BGP routing table entry for 1:100:VEID-1002:Blk-1000/136, version 4
Paths: (1 available, best #1, table L2VPN-VPLS-BGP-Table)
 Not advertised to any peer
 Refresh Epoch 1
 Local
   10.100.1.2 (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(3100)
     Extended Community: RT:1:100 L2VPN L2:0x0:MTU-1500
     Originator: 10.100.1.2, Cluster list: 10.100.1.4
     rx pathid: 0, tx pathid: 0x0
```

PE1: Find a Label

PE1 is now the remote PE and needs to find a label it can use as a remote VC label for the PW towards PE2.

PE1 must first determine if the VBO is within the range of its configuration. PE1 checks its VE-ID against the range advertised by PE2 with the calculation VBO <= VE-ID < VBO + VBS. In this case, 1000 <= 1001 < 1000 + 50, so PE1 succeeds.

PE1 then needs to pick a remote VC label. The demultiplexer (VC) label to be used by the remote PE is computed as (LB + VE-ID - VBO).

From the earlier prefix, LB is 3100, and VBO is 1000. The VE-ID is the one from PE1 and is 1001. So, PE1 picks label (LB + VE-ID - VBO) = (3100 + 1001 - 1000) = 3101.

Use the **show l2vpn vfi name one** command in order to verify this:

```
PE1#show l2vpn vfi name one
 Legend: RT=Route-target, S=Split-horizon, Y=Yes, N=No

VFI name: one, state: up, type: multipoint, signaling: BGP
 VPN ID: 100, VE-ID: 1001, VE-SIZE: 50
 RD: 1:100, RT: 1:100, 32:64
 Bridge-Domain 1 attachment circuits:
 Pseudo-port interface: pseudowire100001
 Interface          Peer Address    VE-ID  Local Label  Remote Label    S
 pseudowire100002   10.100.1.2      1002   10002        3101            Y
```

Additional Verification Commands

```
PE1#show mpls l2transport vc detail
Local interface: VFI one vfi up
 Interworking type is Ethernet
 Destination address: 10.100.1.2, VC ID: 100, VC status: up
   Output interface: Et1/0, imposed label stack {17 3101}
   Preferred path: not configured
   Default path: active
   Next hop: 10.1.1.4
 Create time: 02:06:08, last status change time: 02:06:08
   Last label FSM state change time: 02:06:08
 Signaling protocol: BGP
   Status TLV support (local/remote)   : Not Applicable
     LDP route watch                   : Not Applicable
     Label/status state machine        : established, LruRru
     Last local dataplane   status rcvd: No fault
     Last BFD dataplane     status rcvd: Not Applicable
     Last BFD peer monitor  status rcvd: Not Applicable
     Last local AC  circuit status rcvd: No fault
     Last local AC  circuit status sent: No fault
     Last local PW i/f circ status rcvd: No fault
     Last local LDP TLV     status sent: Not Applicable
     Last remote LDP TLV    status rcvd: Not Applicable
     Last remote LDP ADJ    status rcvd: Not Applicable
   MPLS VC labels: local 10002, remote 3101
   Group ID: local 0, remote 0
   MTU: local 1500, remote 1500
 Control Word: Off
 Dataplane:
   SSM segment/switch IDs: 8195/4097 (used), PWID: 3
 VC statistics:
   transit packet totals: receive 0, send 0
   transit byte totals:   receive 0, send 0
   transit packet drops:  receive 0, seq error 0, send 0

PE1#show mpls infrastructure lfd block-database id 1
Block-DB entry for block-id : 0x1
Block-size : 50, App-Key type : AToM PWID
```

```
App-Key entries:
 l2ckt(1)  10000
 l2ckt(2)  10001
 l2ckt(3)  10002
 l2ckt(4)  10003
 l2ckt(5)  10004
 l2ckt(6)  10005
 l2ckt(7)  10006
 l2ckt(8)  10007
 l2ckt(9)  10008
 l2ckt(10)  10009
 l2ckt(11)  10010
 l2ckt(12)  10011
 l2ckt(13)  10012
 l2ckt(14)  10013
 l2ckt(15)  10014
 l2ckt(16)  10015
 l2ckt(17)  10016
 l2ckt(18)  10017
 l2ckt(19)  10018
 l2ckt(20)  10019
 l2ckt(21)  10020
 l2ckt(22)  10021
 l2ckt(23)  10022
 l2ckt(24)  10023
 l2ckt(25)  10024
 l2ckt(26)  10025
 l2ckt(27)  10026
 l2ckt(28)  10027
 l2ckt(29)  10028
 l2ckt(30)  10029
 l2ckt(31)  10030
 l2ckt(32)  10031
 l2ckt(33)  10032
 l2ckt(34)  10033
 l2ckt(35)  10034
 l2ckt(36)  10035
 l2ckt(37)  10036
 l2ckt(38)  10037
 l2ckt(39)  10038
 l2ckt(40)  10039
 l2ckt(41)  10040
 l2ckt(42)  10041
 l2ckt(43)  10042
 l2ckt(44)  10043
 l2ckt(45)  10044
 l2ckt(46)  10045
 l2ckt(47)  10046
 l2ckt(48)  10047
 l2ckt(49)  10048
 l2ckt(50)  10049

PE1#show l2vpn atom vc destination 10.100.1.2

                                      Service
Interface Dest Address    VC ID      Type   Name                     Status
--------- --------------- ---------- ------ ------------------------ ----------
pw100002  10.100.1.2      100        vfi    one                      UP

PE1#show l2vpn atom vc destination 10.100.1.2 detail
pseudowire100002 is up, VC status is up PW type: Ethernet
 Create time: 02:11:13, last status change time: 02:11:13
   Last label FSM state change time: 02:11:13
 Destination address: 10.100.1.2 VC ID: 100
   Output interface: Et1/0, imposed label stack {17 3101}
   Preferred path: not configured
   Default path: active
   Next hop: 10.1.1.4
 Member of vfi service one
   Bridge-Domain id: 1
   Service id: 0xe7000001
 Signaling protocol: BGP
   Local VE ID: 1001, Remote VE ID: 1002
   Status TLV support (local/remote)         : Not Applicable
     LDP route watch                         : Not Applicable
     Label/status state machine              : established, LruRru
     Local dataplane status received         : No fault
     BFD dataplane status received           : Not Applicable
     BFD peer monitor status received        : Not Applicable
     Status received from access circuit     : No fault
     Status sent to access circuit           : No fault
     Status received from pseudowire i/f     : No fault
     Status sent to network peer             : Not Applicable
     Status received from network peer       : Not Applicable
     Adjacency status of remote peer         : Not Applicable
 Bindings
   Parameter    Local                          Remote
   ------------ ------------------------------ ------------------------------
   Label        10002                          3101
   Group ID     0                              0
   Interface
   MTU          1500                           1500
   Control word off                            off
   PW type      Ethernet                       Ethernet
   VCCV CV type 0x32                           0x32
                  LSPV [2], BFD/Raw [5]          LSPV [2], BFD/Raw [5]
                  BFD/Raw + sig [6]              BFD/Raw + sig [6]
   VCCV CC type 0x07                           0x07
                  CW [1], RA [2], TTL [3]       CW [1], RA [2], TTL [3]
   Status TLV   disabled                       N/A
 Dataplane:
   SSM segment/switch IDs: 8195/4097 (used), PWID: 3
 Rx Counters
   0 input transit packets, 0 bytes
   0 drops, 0 seq err
 Tx Counters
   0 output transit packets, 0 bytes
   0 drops

PE1#show l2vpn signaling rib rd 1:100
+- Origin of entry                                (i=iBGP/e=eBGP)
| +- Provisioned                                  (Yes/No)?
| | +- Stale entry                                (Yes/No)?
| | |
v v v
O P S     RD             VE-ID    VBO     VBS       LB       Next-Hop
-+-+-+-----------------+-------+-------+-------+---------+-----------------+
i Y N  1:100             1002    1000    50      3100       10.100.1.2

PE1#show l2vpn signaling rib rd 1:100 detail
Route 1:100:1002 (epoch:0) from iBGP peer 10.100.1.2
Provisioned (Y) Stale (N)
 Route-Target: 1:100
NLRI [FF000001]
   VE-ID:1002 VBO:1000 VBS:50 LB:3100
   MTU: 1500 Control Word: off
```

```
RIB Filter [27000002]
   RD: 1:100
   VE-ID: 1001, VBO: 1000, VBS: 50 LB: 10000
Forwarder [58000001] VFI one

PE1#show l2vpn atom pwid
AToM Pseudowire IDs: In use: 50, In holddown: 0

Label  Peer-Address    VCID       PWID       In-Use FirstUse ResuedAt FreedAt
------ --------------- ---------- ---------- ------ -------- -------- --------
10000  0.0.0.0         0          1          Yes    00:00:15 Never    Never
10001  0.0.0.0         0          2          Yes    00:00:15 Never    Never
10002  10.100.1.2      100        3          Yes    00:00:15 Never    Never
10003  0.0.0.0         0          4          Yes    00:00:15 Never    Never
10004  0.0.0.0         0          5          Yes    00:00:15 Never    Never

PE1#show l2vpn atom summary
Destination address: 10.100.1.2, total number of vc: 1
 0 unknown, 1 up, 0 down, 0 admin down, 0 recovering, 0 standby, 0 hotstandby
 1 active vc on MPLS interface Et1/0
```

Multiple L2VPN VPLS Prefixes Advertised by PE Router for One VFI

It is possible that one PE might need to advertise multiple label bocks for one virtual forwarding instance (VFI).

If the VE-ID of the remote PE does not fall in the range advertised by the local PE, the remote PE cannot pick a remote label for the PW. This calculation, described earlier, is VBO <= VE-ID < VBO + VBS.

If this check fails, the VE-ID of the remote PE is out of range. The remote PE ignores the prefix received from the local PE. The local PE learns that the remote PE is out of range when it receives the prefix which the remote PE is advertising. The local PE needs to determine what remote label to use for that remote PE router. The local PE also sends a new, second prefix for a new block of local labels to the remote PE, which the remote PE should be able to use in order to select a remote label.

PE1 Configuration

The previous example is continued here; PE1 still has:

```
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
```

PE2 Configuration

PE2 now has a VE-ID of 1002 and this configuration:

```
l2vpn vfi context one
 vpn id 100
 autodiscovery bgp signaling bgp
 ve id 10002
 ve range 50
!
mpls label range 3000 60000
```

Initial Label Blocks

Both PE1 and PE2 start with these initial label blocks.

```
PE1#show bgp l2vpn vpls rd 1:100
BGP table version is 2, 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 ?

PE2#show bgp l2vpn vpls rd 1:100
BGP table version is 3, 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
 *>  1:100:VEID-10002:Blk-10000/136
                      0.0.0.0                            32768 ?
```

PE1 and PE2 Exchange

Use the **debug bgp l2vpn vpls updates** command in order to review the PE1 and PE2 exchange, then use the **show bgp l2vpn vpls rd 1:100** command in order to review details.

```
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 ssacount 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 ssacount 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 ?
```

Analysis of PE1 and PE2 Exchange

PE1 and PE2 now have advertised two label blocks each to each other.

PE1 first advertises an initial BGP update to PE2:

```
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
```

This update has the NLRI set according to the configuration on PE1.

PE1 then receives the initial BGP update from PE2.

```
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
```

PE2 advertises the initial prefix with the values VE-ID 10002, VBO = 10000, VBS = 50, LB = 3000.

PE1 notices that PE2 is out of range since PE1 started off with label block LB to (LB + VBS - 1) or from 10000 to (10000 + 50 - 1) = 10049.

PE1 must determine if the VBO is within the range of its configuration. So, the VE-ID of PE2 needs to be checked against the range advertised by PE1. The calculation is VBO <= VE-ID < VBO + VBS. In this case, 1000 <= 10002 < 1000 + 50, which is not true. So, PE1 needs to send a new label block in order to accomodate the out-of-range VE-ID of PE2. In reaction to the initial update from PE2, PE1 formats and sends a new, additional BGP update to PE2. PE1 now uses a new VBO of 10000.

```
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
```

For PE1, the VBO is 10000, VBS is 50, LB is 10053. The check for PE2 is VBO <= VE-ID < VBO + VBS. In this case, 10000 <= 10002 < 10000 + 50, which is true. PE2 is able to pick a remote label from this new label block [10053 - 10102] from PE1. In other words, PE1 added a new label block in order to accommodate PE2 and sent two BGP update messages.

The same happens in the opposite direction. PE2 receives the initial BGP update from PE1. This update has these values VE-ID 1001, VBO = 1000, VBS = 50, LB = 10000.

PE2 notices that VE-ID of PE1 is out-of-range with PE2's initial update. PE1?s check is VBO <= VE-ID < VBO + VBS or 10000 <= 1001 < 10000 + 50. In response, PE2 sends this second BGP update, with a new label block [3053 - 3102] that accommodates the VE-ID of 1001 of PE1 because PE1?s check is VBO <= VE-ID < VBO + VBS or 1000 <= 1001 < 1000 + 50.

```
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
```

Prefix Details

These are the details of the two prefixes originated by 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 2
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 (10.100.1.1)
     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 10000
BGP routing table entry for 1:100:VEID-1001:Blk-10000/136, version 4
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 (10.100.1.1)
     Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
     AGI version(0), VE Block Size(50) Label Base(10053)
     Extended Community: RT:1:100 RT:32:64 L2VPN L2:0x0:MTU-1500
       L2VPN L2:0x0:MTU-1500
     rx pathid: 0, tx pathid: 0x0
```

Here, two PE routers have discontiguous number schemes, which causes each PE to send out two BGP updates. If there are many PE routers with discontiguous number schemes, the number of BGP updates quickly grows very large.

www.cisco.com says: "For example, VE-ID numbering sequences such as 1, 2, 3 or 501, 502, 503 are good because the VE-IDs are contiguous. A numbering scheme such as 100, 200, 300 is bad because it is non-contiguous."

The first examples of 1, 2, 3 or 501, 502, 503 are contiguous numbers, so each PE router needs to send only one L2VPN VPLS prefix. With the third example (100, 200, 300), each PE has to send out many L2VPN VPLS prefixes. For non-contiguous numbers, a large enough VE range would keep the number of prefixes to be advertised lower. However, the amount of reserved (wasted) labels is still larger.

Interoperability

If the BGP Route Reflector (RR) runs software that does not understand RFC 4761, but does have support for RFC 4762, the special BGP **neighbor x.x.x.x prefix-length-size 2** configuration command is needed on the RR so it can reflect the BGP updates used for RFC 4761.

Prefixes are usually sent with a length of 1 byte. Cisco IOS software implemented the draft 'draft-ietf-l2vpn-signaling-08,' which later became RFC 6074. A length field of 1 byte was chosen at the time, indicating the length in bits.

RFC 6074 Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) specifies that the NLRI encoding for BGP auto-discovery should be a length of 2 bytes. The 2 bytes indicate how many bytes of prefix follow in variable length prefix.

Section 7 of RFC 6074, "BGP-AD and VPLS-BGP Interoperability," states:

"Both BGP-AD and VPLS-BGP [RFC4761] use the same AFI/SAFI. In order for both BGP-AD and VPLS-BGP to co-exist, the NLRI length must be used as a demultiplexer.

The BGP-AD NLRI has an NLRI length of 12 bytes, containing only an 8-byte RD and a 4-byte VSI-ID. VPLS-BGP [RFC4761] uses a 17-byte NLRI length. Therefore, implementations of BGP-AD must ignore NLRI that are greater than 12 bytes."

If the **neighbor x.x.x.x prefix-length-size 2** command is not present on the RRs, the BGP neighbor does not come up, and the RR interprets the length field as 1 byte only. This notification appears on the RR:

```
%BGP-3-NOTIFICATION: sent to neighbor 10.100.1.2 3/10 (illegal network) 1 bytes FF
%BGP-4-MSGDUMP: unsupported or mal-formatted message received from 10.100.1.2:
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 005E 0200 0000 4780 0E1C 0019 4104 0A64
0102 0000 1100 0000 0100 0000 6427 1227 1000 3200 BB80 4001 0102 4002 0080 0404
*Feb 15 12:14:11.561: %BGP_SESSION-5-ADJCHANGE: neighbor 10.100.1.2 L2VPN Vpls
topology base removed from session  BGP Notification sent
*Feb 15 12:14:11.561: %BGP_SESSION-5-ADJCHANGE: neighbor 10.100.1.2 IPv4 Unicast
topology base removed from session  BGP Notification sent
```

This notification appears on the PE router:

```
%BGP-3-NOTIFICATION: received from neighbor 10.100.1.4 3/10 (illegal network)
1 bytes FD
```

This occurs because, in the original implementation of BGP auto-discovery in Cisco IOS software, the length field was 1 byte.

If you put the **neighbor x.x.x.x prefix-length-size 2** command on the RR, the notifications do not appear.

```
router bgp 1
 neighbor 10.100.1.2 remote-as 1
 neighbor 10.100.1.2 update-source Loopback0
!
 address-family l2vpn vpls
 neighbor 10.100.1.2 activate
 neighbor 10.100.1.2 send-community extended
 neighbor 10.100.1.2 prefix-length-size 2
 neighbor 10.100.1.2 route-reflector-client
```