IOS Implementation of the iBGP PE−CE Feature

Document ID: 117567

Contributed by Luc De Ghein, Cisco TAC Engineer.
Apr 04, 2014

Contents

Introduction

This document describes how the Internal Border Gateway Protocol (iBGP) between Provider Edge (PE) and Customer Edge (CE) feature is implemented in Cisco IOS®.

Background Information

Until the new iBGP PE−CE feature, iBGP between PE and CE (hence on a Virtual Routing and Forwarding (VRF) interface on the PE router) was not officially supported. One exception is iBGP on VRF interfaces in a Multi−VRF CE (VRF−Lite) setup. The motivation to deploy this feature is:

- The customer wants to have one single Autonomous System Number (ASN) on the multiple sites of the VRF, without the deployment of External Border Gateway Protocol (eBGP) with as−override.
- The customer wants to provide internal route reflection towards the CE routers, acting as if the Service Provider (SP) core is one transparent route reflector (RR).

With this feature, the sites of the VRF can have the same ASN as the SP core. However, in case the ASN of the VRF sites are different than the ASN of the SP core, it can be made to appear the same with the use of the feature local−Autonomous System (AS).

Implement iBGP PE−CE

Here are the two major parts in order to make this feature work:

- A new attribute ATTR_SET was added to the BGP protocol in order to carry the VPN BGP attributes across the SP core in a transparent manner.

- Make the PE router a RR for the iBGP sessions towards the CE routers in the VRF and as a RR towards the VPNv4 neighbors (other PE routers or RRs).

The new ATTR_SET attribute allows the SP to carry all of the BGP attributes of the customer in a transparent manner and does not interfere with the SP attributes and BGP policies. Such attributes are the cluster list, local preference, communities, and so on.

BGP Customer Route Attribute

ATTR_SET is the new BGP attribute used in order to carry the VPN BGP attributes of the SP customer. It is an optional transitive attribute. In this attribute, all of the customer BGP attributes from the BGP Update message, except for the MP_REACH and MP_UNREACH attributes, can be carried.

The ATTR_SET attribute has this format:

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

The attribute flags are the regular BGP attribute flags (refer to RFC 4271). The attribute length indicates whether the attribute length is one or two octets. The purpose of the Origin AS field is to prevent the leak of one route originated in one AS to be leaked to another AS without the proper manipulation of the AS_PATH. The variable length Path Attributes field carries the VPN BGP attributes that must be carried across the SP core.

On the egress PE router, the VPN BGP attributes are pushed into this attribute. On the ingress PE router, these attributes are popped from the attribute, before the BGP prefix is sent to the CE router. This attribute provides the isolation of BGP attributes between the SP network and the customer VPN and vice versa. For example, the SP route reflection cluster list attribute is not seen and considered inside the VPN network. But also, the VPN route reflection cluster list attribute is not seen and considered inside the SP network.

Look at Figure 1 in order to see the propagation of a customer BGP prefix across the SP network.

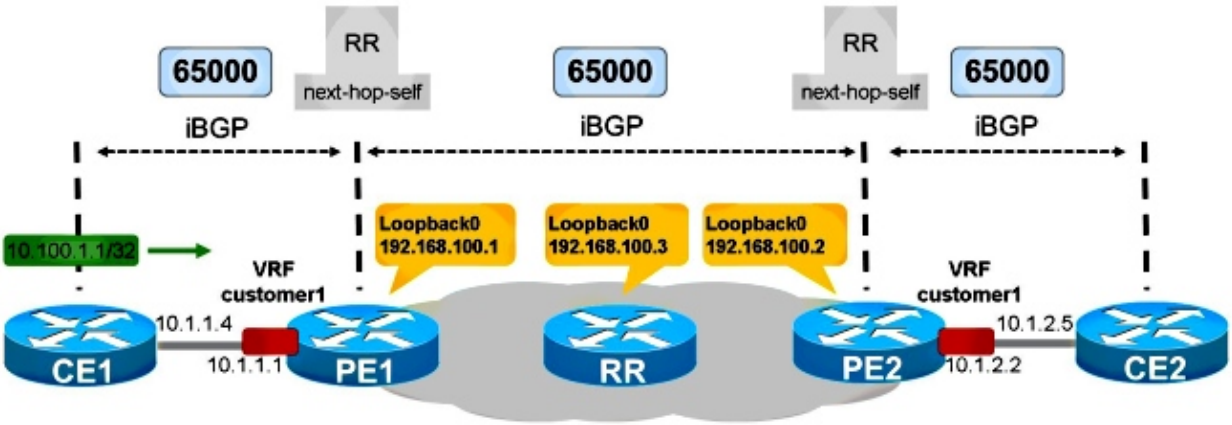

Figure 1

CE1 and CE2 are in the same AS as the SP network: 65000. PE1 has iBGP configured towards CE1. PE1 reflects the path for the prefix 10.100.1.1/32 towards the RR in the SP network. The RR reflects the iBGP path towards the PE routers as usual. PE2 reflects the path towards CE2.

In order for this to work properly, you must:

- Have code on PE1 and PE2 that has the iBGP PE−CE feature support
- Configure PE1 and PE2 in order to perform route reflection on their BGP session towards their respective CE routers
- Have next−hop−self on the PE routers for the BGP session towards their CE routers
- Make sure that each VPN site uses different Route Distinguishers (RD)

Configure

Refer to Figure 1.

Here is the needed configuration for PE1 and PE2:

```
PE1

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

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

PE2

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

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

Note: If the PE does not have the ***neighbor <internal-CE> internal-vpn-client*** command for the CE neighbor, it does not propagate the prefixes from the CE towards the SP RRs/PE routers.

Note: If the PE is not the RR in the VRF, it does not propagate the prefixes from the RRs/PE routers towards the CE router.

New Command

There is a new command, ***neighbor <internal-CE> internal-vpn-client***, to make this feaure work. It must be configured on the PE router only for the iBGP session towards the CE routers.

Note: The iBGP PE-CE Multi-VRF CE (VRF-Lite) feature is still supported without the ***neighbor <internal-CE> internal-vpn-client*** command.

Note: When the ***neighbor <internal-CE> internal-vpn-client*** command is configured, the ***neighbor <internal-CE> route-reflector-client*** and ***neighbor <internal-CE> next-hop-self*** commands are automatically put in the configuration as well. When either of the ***neighbor <internal-CE> route-reflector-client*** and ***neighbor <internal-CE> next-hop-self*** commands (or both) are removed and a reload is performed, then they are automatically put back in the configuration.

Detailed Look at ATTR_SET

Refer to Figure 1.

This is the prefix advertised by CE1:

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

When PE1 receives the BGP prefix 10.100.1.1/32 from CE1, it stores it twice:

```
PE1#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 21
```

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

The first path is the actual path on PE1, because it is received from CE1.

The second path is the path that is advertised towards the RRs/PE routers. It is marked with ***ibgp sourced***. It contains the ATTR_SET attribute. Notice that this path has one or more Route Targets (RTs) attached to it.

PE1 advertises the prefix as shown here:

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

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

Total number of prefixes 1
```

This is how the RR sees the path:

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

Notice that the local preference of this VPNv4 unicast prefix in the core is 100. In the ATTR_SET, the original local preference of 200 is stored. However, this is transparent to the RR in the SP core.

On PE2, you see the prefix as shown here:

```
PE2#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 5
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.100.1.1, Cluster list: 192.168.100.3, 192.168.100.1
      ATTR_SET Attribute:
        Originator AS 65000
        Origin IGP
        Aspath
        Med 0
        LocalPref 200
        Cluster list
        192.168.100.1,
        Originator 10.100.1.1
      mpls labels in/out nolabel/18
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for 65000:2:10.100.1.1/32, version 6
Paths: (1 available, best #1, table customer1)
  Advertised to update-groups:
     1
  Refresh Epoch 2
  Local, imported path from 65000:1:10.100.1.1/32 (global)
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      Originator AS(ibgp-pece): 65000
      Originator: 10.100.1.1, Cluster list: 192.168.100.1
      mpls labels in/out nolabel/18
      rx pathid:0, tx pathid: 0x0
```

The first path is the one received from the RR, with the ATTR_SET. Note that the RD is 65000:1, the origin RD. The second path is the imported path from the VRF table with RD 65000:1. The ATTR_SET has been removed.

This is the path as seen on CE2:

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

Notice that the next-hop is ***10.1.2.2***, which is PE2. The cluster list contains routers PE1 and PE2. These are the RRs that matter inside the VPN. The SP RR (10.100.1.3) is not in the cluster list.

The local preference of 200 has been preserved inside the VPN across the SP network.

The ***debug bgp vpnv4 unicast updates*** command shows the update propagated in the SP network:

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

RR#
BGP(4): 192.168.100.1 rcvd UPDATE w/ attr: nexthop 192.168.100.1, origin i, localpref
100, originator 10.100.1.1, clusterlist 192.168.100.1, extended community RT:1:1,
[ATTR_SET attribute:  originator AS 65000, origin IGP, aspath , med 0, localpref 200,
cluster list 192.168.100.1 , originator 10.100.1.1]
BGP(4): 192.168.100.1 rcvd 65000:1:10.100.1.1/32, label 16
RT address family is not configured. Can't create RTC route
BGP(4): (base) 192.168.100.1 send UPDATE (format) 65000:1:10.100.1.1/32, next
192.168.100.1, label 16, metric 0, path Local, extended community RT:1:1

PE2#
BGP(4): 192.168.100.3 rcvd UPDATE w/ attr: nexthop 192.168.100.1, origin i, localpref
100, originator 10.100.1.1, clusterlist 192.168.100.3 192.168.100.1, extended community
RT:1:1, [ATTR_SET attribute:  originator AS 65000, origin IGP, aspath , med 0, localpref
200, cluster list 192.168.100.1 , originator 10.100.1.1]
BGP(4): 192.168.100.3 rcvd 65000:1:10.100.1.1/32, label 16
RT address family is not configured. Can't create RTC route
BGP(4): Revise route installing 1 of 1 routes for 10.100.1.1/32 -> 192.168.100.1
(customer1) to customer1 IP table
BGP(4): 10.1.2.5 NEXT_HOP is set to self for net 65000:2:10.100.1.1/32,
```

Note: PE1 received its own update from RR and then dropped it. This is because PE1 and PE2 are in the same update group on RR.

Note: If you want to dump the complete Update message in hexadecimal, use the ***detail*** keyword for the ***debug BGP updates*** command.

```
PE2#   debug bgp vpnv4 unicast updates detail
BGP updates debugging is on with detail for address family: VPNv4 Unicast

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

Next Hop Handling

Next−hop−self must be configured on the PE routers for this feature. The reason for this is that normally the next−hop is transported unchanged with iBGP. However, here there are two separate networks: the VPN network and the SP network, which run separate Interior Gateway Protocols (IGPs). Hence, the IGP metric cannot be easily compared and used for best path calculation between the two networks. The approach chosen by RFC 6368 is to have next−hop−self mandatory for the iBGP session towards the CE, which avoids the previously described issue all together. An advantage is that the VRF sites can run different IGPs with this approach.

RD

RFC 6368 mentions that it is recommended that different VRF sites of the same VPN use different (unique) RDs. In Cisco IOS, this is mandatory for this feature.

iBGP PE−CE Feature with Local−AS

Refer to Figure 2. The VPN customer1 has ASN 65001.

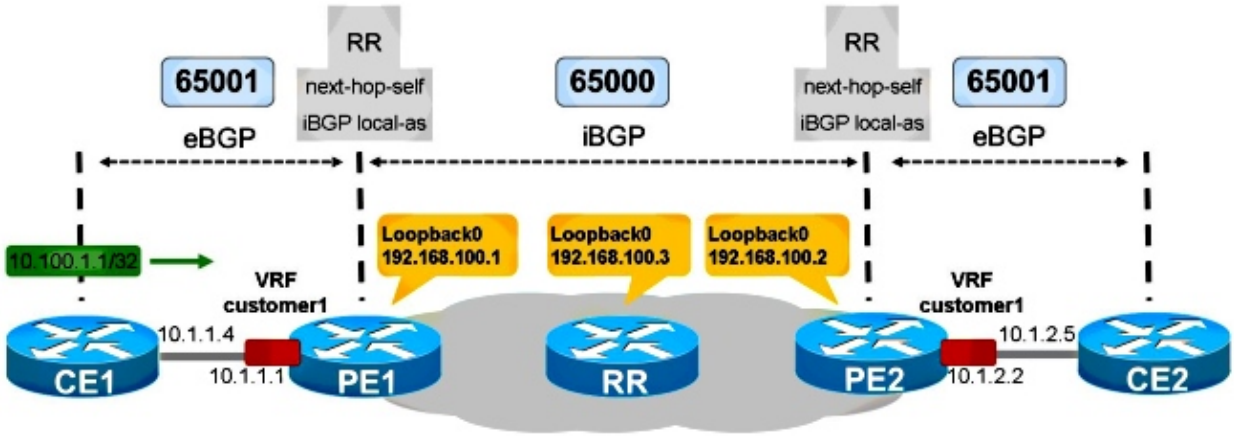

Figure 2

CE1 is in AS 65001. In order to make this internal BGP from the standpoint of PE1, it needs the iBGP local−as feature.

```
CE1

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

PE1

router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.100.3 remote-as 65000
 neighbor 192.168.100.3 update-source Loopback0
 !
 address-family vpnv4
  neighbor 192.168.100.3 activate
  neighbor 192.168.100.3 send-community extended
 exit-address-family
 !
```

```
 address-family ipv4 vrf customer1
  neighbor 10.1.1.4 remote-as 65001
  neighbor 10.1.1.4 local-as 65001
  neighbor 10.1.1.4 activate
  neighbor 10.1.1.4 internal-vpn-client
  neighbor 10.1.1.4 route-reflector-client
  neighbor 10.1.1.4 next-hop-self
 exit-address-family
```

PE2 and CE2 are configured similarly.

PE1 sees the BGP prefix as shown here:

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

The prefix is an internal BGP.

PE2 sees this:

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

```
      Origin IGP, metric 0, localpref 200, valid, internal, best
      Originator AS(ibgp-pece): 65001
      Originator: 10.100.1.1, Cluster list: 192.168.100.1
      mpls labels in/out nolabel/18
      rx pathid: 0, tx pathid: 0x0
```

The Originator AS is 65001, which is the AS used when the prefix is sent from PE2 to CE2. So, the AS is preserved, and so is the local preference in this example.

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

You see ***Local*** instead of an AS Path. This means it is an internal BGP route originated in AS 65001, which is also the configured ASN of router CE2. All of the BGP attributes have been taken from the ATTR_SET attribute. This adheres to the rules for Case 1 in the next section.

Rules for Route Exchange Between Different VRF Sites

The ATTR_SET contains the Originator AS of the originating VRF. This Originating AS is checked by the remote PE, when it removes the ATTR_SET before it sends the prefix to the CE router.

Case 1: If the Originating AS matches the configured AS for the CE router, then the BGP attributes are taken from the ATTR_SET attribute when the PE imports the path into the destination VRF.

Case 2: If the Originating AS does not match the configured AS for the CE router, then the set of attributes for the constructed path are taken as shown here:

1. The path attributes are set to the attributes contained in the ATTR_SET attribute.

2. The iBGP−specific attributes are discarded (LOCAL_PREF, ORIGINATOR, and CLUSTER_LIST).

3. The ***Origin AS*** number contained in the ATTR_SET attribute is prepended to the AS_PATH and follows the rules that apply to an External BGP peering between the source and destination ASs.

4. If the autonomous system associated with the VRF is the same as the VPN provider autonomous system and the AS_PATH attribute of the VPN route is not empty, it SHALL be prepended to the AS_PATH attribute of the VRF route.

   Refer to Figure 3. CE1 and PE1 have the AS 65000 and are configured with the iBGP PE−CE feature. CE2 has ASN 65001. This means that there is eBGP between PE2 and CE2.

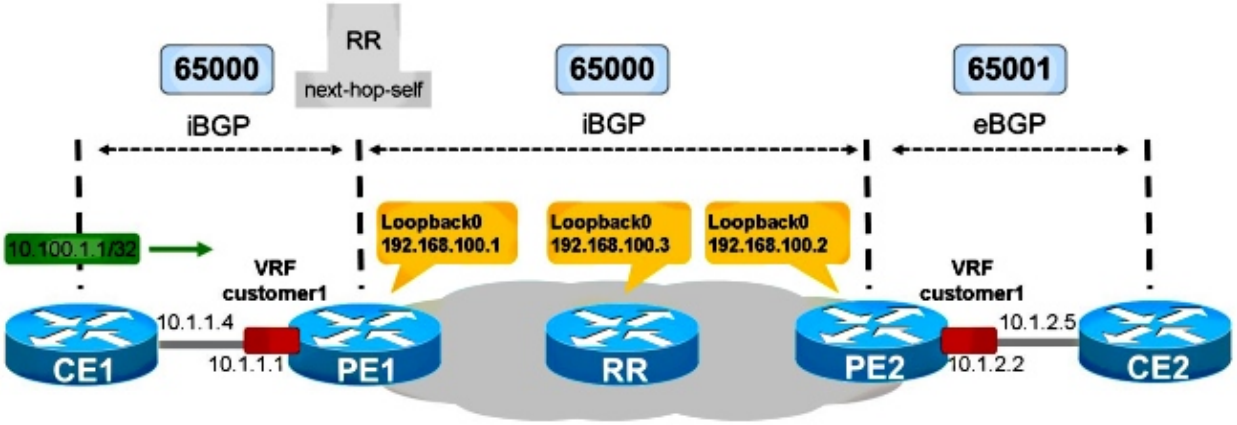

Figure 3

PE2 sees the route as follows:

```
PE2#show bgp vpnv4 unicast all 10.100.1.1/32
BGP routing table entry for 65000:1:10.100.1.1/32, version 43
Paths: (1 available, best #1, no table)
  Not advertised to any peer
  Refresh Epoch 6
  Local
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 10.100.1.1, Cluster list: 192.168.100.3, 192.168.100.1
      ATTR_SET Attribute:
        Originator AS 65000
        Origin IGP
        Aspath
        Med 0
        LocalPref 200
        Cluster list
        192.168.100.1,
        Originator 10.100.1.1
      mpls labels in/out nolabel/17
      rx pathid: 0, tx pathid: 0x0
BGP routing table entry for 65000:2:10.100.1.1/32, version 44
Paths: (1 available, best #1, table customer1)
  Advertised to update-groups:
     6
  Refresh Epoch 6
  Local, imported path from 65000:1:10.100.1.1/32 (global)
    192.168.100.1 (metric 21) (via default) from 192.168.100.3 (192.168.100.3)
      Origin IGP, metric 0, localpref 200, valid, internal, best
      Originator AS(ibgp-pece): 65000
      Originator: 10.100.1.1, Cluster list: 192.168.100.1
      mpls labels in/out nolabel/17
      rx pathid: 0, tx pathid: 0x0
```

This is the prefix as seen on CE2:

```
CE2#show bgp ipv4 unicast 10.100.1.1/32
BGP routing table entry for 10.100.1.1/32, version 5
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  65000
    10.1.2.2 from 10.1.2.2 (192.168.100.2)
```

```
      Origin IGP, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

This is Case 2. The ***Origin AS*** number contained in the ATTR_SET attribute is prepended to the AS_PATH by PE2 and follows the rules that apply to an eBGP peering between the source and destination AS. The iBGP−specific attributes are ignored by PE2 when it creates the route to be advertised to CE2. So, the local preference is 100 and not 200 (as seen in the ATTR_SET attribute).

CE−to−CE VRF−Lite Reflection

Refer to Figure 4.

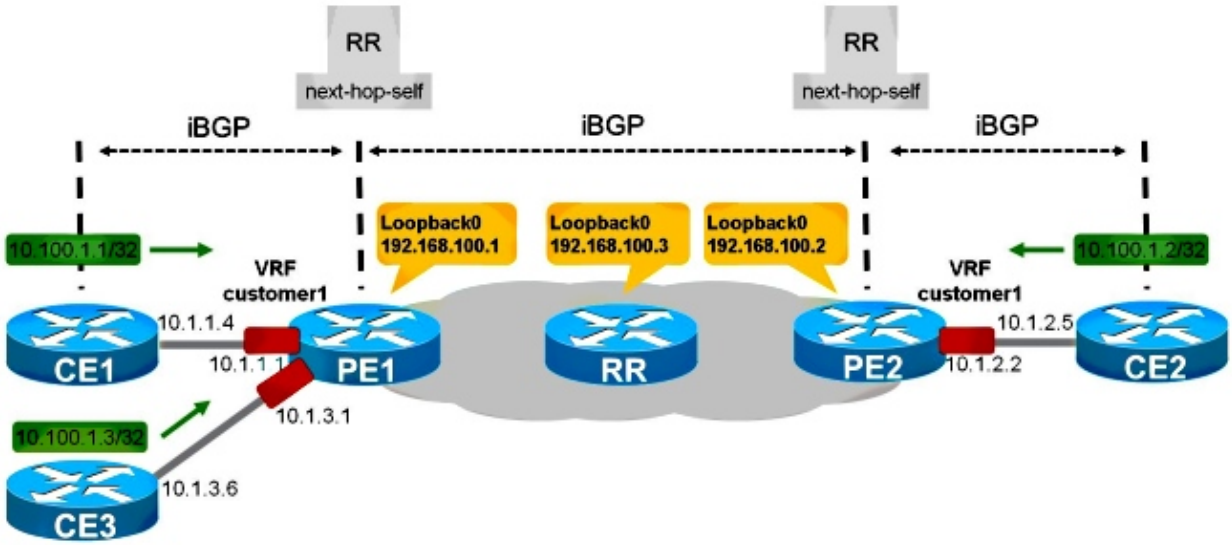

Figure 4

Figure 4 shows an additional CE router, CE3, connected to PE1. CE1 and CE3 are both connected to PE1 on the same VRF instance: customer1. This means that CE1 and CE3 are Multi−VRF CE routers (also known as VRF−Lite) of PE1. PE1 puts itself as next−hop when it advertises the prefixes from CE1 to CE3. In the case that this behavior is not wanted, you could configure ***neighbor 10.1.3.6 next−hop−unchanged*** on PE1. In order to configure this, you must remove ***neighbor 10.1.3.6 next−hop−self*** on PE1. Then CE3 sees the routes from CE1 with CE1 to be the next−hop for those BGP prefixes. In order to make this work, you need the routes for those BGP next−hops in the routing table of CE3. You need a dynamic routing protocol (IGP) or static routes on CE1, PE1, and CE3 in order to make sure the routers have a route for each others next−hop IP addresses. However, there is a problem with this configuration.

The configuration on PE1 is:

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

The prefix from CE1 is seen fine on CE3:

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

However, the prefix from CE2 is seen on CE3 as shown here:

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

The BGP next−hop is ***192.168.100.2***, the loopback IP address of PE2. PE1 did not rewrite the BGP next−hop to itself when it advertised the prefix 10.100.1.2/32 to CE3. This makes this prefix unusable on CE3.

So, in the case of a mix of the iBGP PE−CE feature across MPLS−VPN and iBGP VRF−Lite, you must make sure that you always have next−hop−self on the PE routers.

You cannot preserve the next−hop when a PE router is an RR that reflects iBGP routes from one CE to another CE across VRF interfaces locally on the PE. When you run iBGP PE−CE across an MPLS VPN network, you must use ***internal−vpn−client*** for the iBGP sessions towards the CE routers. When you have more than one local CE in a VRF on a PE router, then you must keep ***next−hop−self*** for those BGP peers.

You could look at route−maps in order to set the next−hop to self for prefixes received from other PE routers, but not for reflected prefixes from other locally−connected CE routers. However, it is not currently supported to set the next−hop to self in an outbound route−map. That configuration is shown here:

```
router bgp 65000

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

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

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

```
 set ip next-hop self
!

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

However, this is not supported:

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

Older Cisco IOS on the PE Router

If PE1 runs older Cisco IOS software that lacks the feature iBGP PE−CE, then PE1 never sets itself as the next−hop for the reflected iBGP prefixes. This means that the reflected BGP prefix (10.100.1.1/32) from CE1 (10.100.1.1) to CE2 −via PE1− would have CE1 (10.1.1.4) as the next−hop.

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

The prefix from CE2 (10.100.1.2/32) is seen with PE2 as the next−hop, because PE1 does not do next−hop−self for this prefix either:

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

In order for the iBGP PE−CE feature to work properly, all PE routers for the VPN where the feature is enabled must have the code to support the feature and have the feature enabled.

Next−hop−self for eBGP on VRF

Refer to Figure 5.

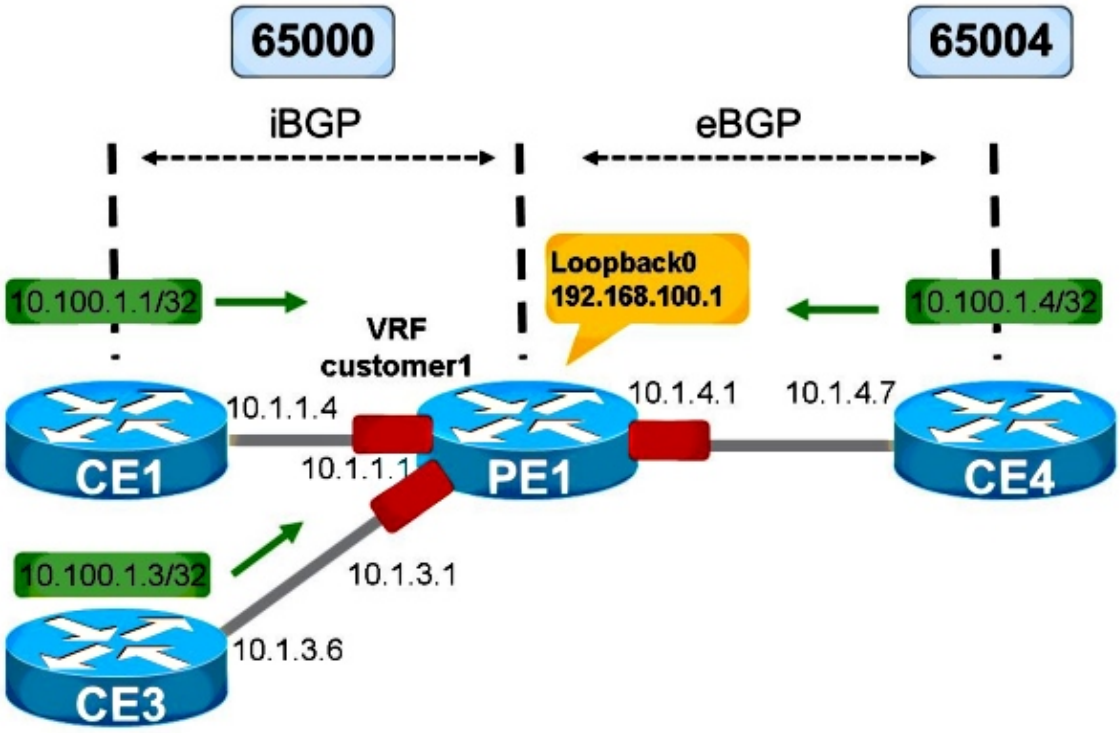

Figure 5

Figure 5 shows a VRF−Lite setup. The session from PE1 towards CE4 is eBGP. The session from PE1 towards CE3 is still iBGP.

For eBGP prefixes, the next−hop is always set to self when it advertises the prefixes towards an iBGP neighbor on VRF. This is regardless of the fact if the session towards the iBGP neighbor across VRF has next−hop−self set or not.

In Figure 5, CE3 sees the prefixes from CE4 with PE1 as the next−hop.

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

This occurs with next−hop−self on PE1 towards CE3 or without.

If the interfaces on PE1 towards CE3 and CE4 are not in a VRF, but in the global context, the next−hop−self towards CE3 does makes a difference.

Without next−hop−self on PE1 towards CE3, you see:

```
PE1#show bgp vrf customer1 vpnv4 unicast neighbors 10.1.3.6
BGP neighbor is 10.1.3.6,  vrf customer1,  remote AS 65000, internal link
...
 For address family: VPNv4 Unicast
  Translates address family IPv4 Unicast for VRF customer1
```

```
  Session: 10.1.3.6
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 12, Advertise bit 0
  Route-Reflector Client
  12 update-group member
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
  Interface associated: (none)
```

Although the next−hop−self is implicitly enabled, the output does not indicate this.

With next−hop−self on PE1 towards CE3, you see:

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

Whereas, if the interfaces towards CE3 and CE4 are in a global context, the next−hop for prefixes from CE4 is CE4 itself when next−hop−self is not configured:

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

For next−hop−self on PE1 towards CE3:

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

This was done based on RFC 4364.

If you want to not set next−hop−self for eBGP prefixes towards an iBGP session across a VRF interface, you must configure ***next−hop−unchanged***. The support for this only occured with Cisco bug ID CSCuj11720.

```
router bgp 65000
...
 address-family ipv4 vrf customer1
  neighbor 10.1.1.4 remote-as 65000
  neighbor 10.1.1.4 activate
  neighbor 10.1.1.4 route-reflector-client
  neighbor 10.1.3.6 remote-as 65000
  neighbor 10.1.3.6 activate
  neighbor 10.1.3.6 route-reflector-client
  neighbor 10.1.3.6 next-hop-unchanged
  neighbor 10.1.4.7 remote-as 65004
```

```
  neighbor 10.1.4.7 activate
 exit-address-family
```

Now, CE3 sees CE4 as the next−hop for the prefixes advertised by CE4:

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

If you try to configure the ***next−hop−unchanged*** keyword for the iBGP session towards CE3 on Cisco IOS code prior to Cisco bug ID CSCuj11720, you encounter this error:

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

After Cisco bug ID CSCuj11720, the ***next−hop−unchanged*** keyword is valid for multi−hop eBGP neighbors and iBGP VRF−Lite neighbors.

Updated: Apr 04, 2014 Document ID: 117567