Border Gateway Protocol (BGP) Optimal Route Reflection

Contents

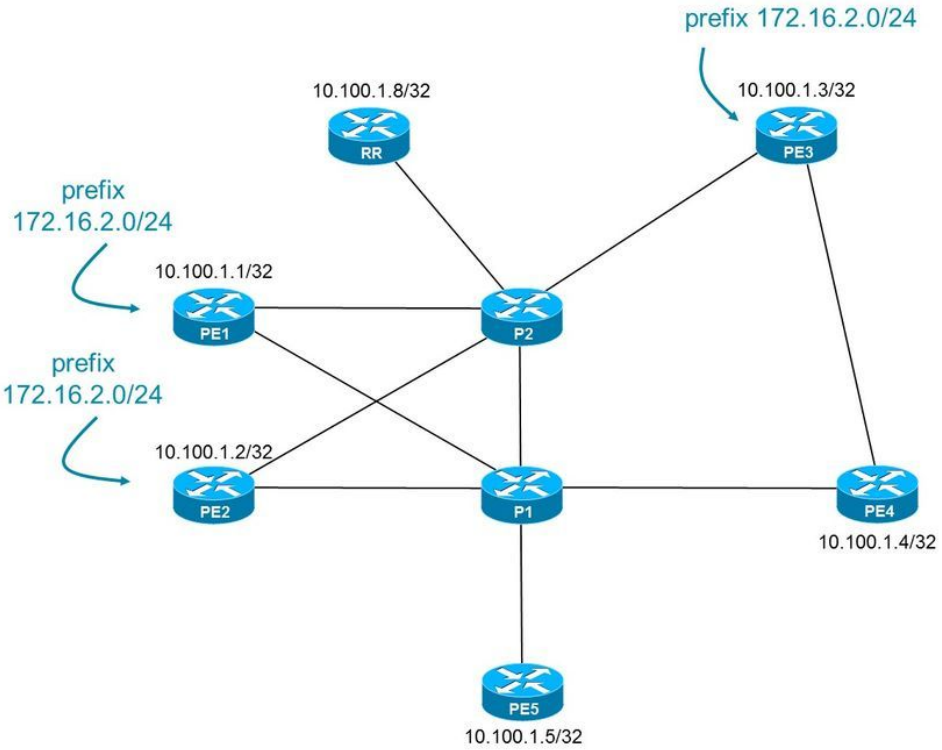
Introduction

This document describes how routing can be influenced when there is one or more Route Reflectors (RRs) in the network to avoid a full mesh between iBGP routers.

Background Information

Step 8 in the BGP best path selection algorithm is to prefer the path with the lowest IGP metric to the BGP next hop. So, if all the steps before step 8 are equal, then step 8 can be the deciding factor on what the best path is on the RR. The IGP cost from the RR to the advertising iBGP router is then determined by the placement of the RR. By default, the RR only advertises the best path to its clients. Depending on where the RR is placed, the IGP cost to the advertising router can be smaller or bigger. If all the IGP costs of the paths are the same, then it is likely going to end up to the tie-breaker of the advertising router having the lowest BGP router ID.

Network Diagram



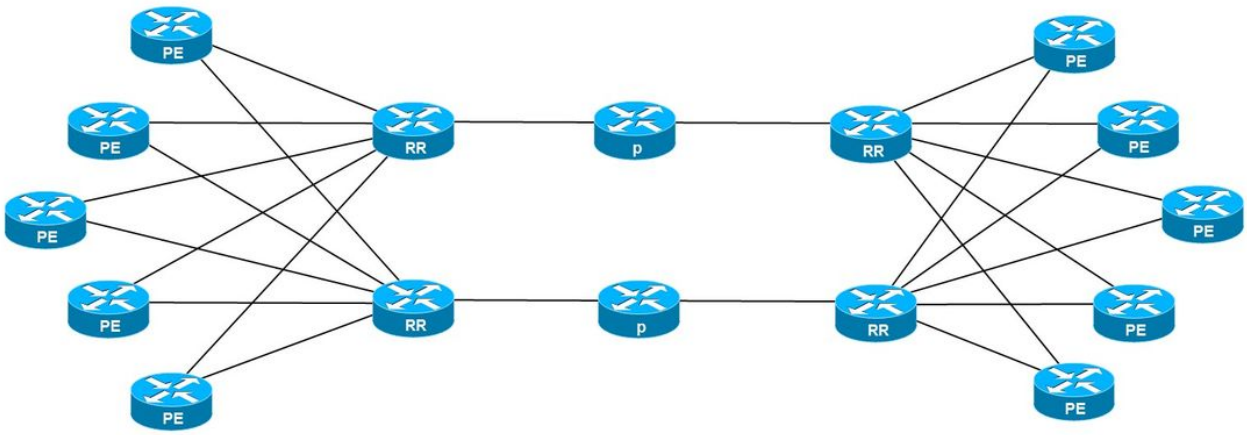
The routers PE1, PE2, and PE3 advertise the prefix 172.16.2.0/24. If all the IGP cost of the links are the same then the RR will see the path from PE1, PE2, and PE3 with an IGP cost of 2. In the end, the RR picks the path from PE1 as the best because it has the lower BGP router ID. This is step 11 in the BGP best path selection algorithm. The result is that all PE routers, including PE4, will pick PE1 as the egress PE router for the prefix 172.16.2.0/24. From the standpoint of PE4, the shorter IGP path to any egress PE router is the path to PE3, with an IGP cost of 1. The IGP cost to any other PE router is 2. For many networks, the fact of transporting the traffic through the transit network in the shortest possible way, is important. This is known as hot-potato routing.

There is another possible reason for RR to have picked the best path from PE1. If in the image, the Interior Gateway Protool (IGP) cost of the link P2-PE3 is 10 and all the other links still have an IGP cost of 1, then the RR would not pick the path from PE3 as the best, even if PE3 had the lowest BGP router ID.

If the administrator of this network wants to have hot-potato routing, then a mechanism must be in place so that when there are RRs in the network, the ingress router can still learn the path to the closest egress router in the iBGP network. The BGP feature Add Path can achieve this. However with that feature, the RRs and the border routers must have more recent code that understand the feature. With the feature of BGP Optimal Route Reflection, this is not a requirement. This feature will allow the RR to send the best path to the ingress BGP border router, based on what the RR thinks is the best path from the perspective of that ingress BGP router.

Another solution that would allow hot-potato routing when RRs are deployed, is the in-line placement of RRs. These RRs are not dedicated RRs, which only run BGP and the IGP. These inline RRs are in the forwarding path, and placed in the network so that they have their own set of RR clients, so that they can reflect the best path to each RR client, which is also the best path from that RR client’s perspective.

As shown in this image, the RRs are placed in the network so that they a small set of nearby RR clients that they can serve. Because of the network design, the RR clients receive the best paths which are the best paths from their point of view, from the RRs so that there can be hot potato routing in the network.



Theory

BGP Optimal Route Reflection is described in IETF draft draft-ietf-idr-bgp-optimal-route-reflection.

The BGP Optimal Route Reflection solution allows the RR to send a specific best path to a specific BGP border router. The RR can choose to send a different best path to different BGP border routers or set of border routers. The border routers must be RR clients of the RR. The goal is that each ingress BGP border router can have a different exit or egress BGP router for the same prefix. If the ingress border router can always forward the traffic to the closes AS-exit router, then this allows for hot-potato routing.

The problem is that the RR normally only sends the same best path to each BGP border router, which prevents hot-potato routing. In order to solve this, you need the RR to be able to calculate different best paths for the same prefix depending on the ingress BGP border router. The best path calculation on the RR is done based on the position of the ingress BGP border router, Hence, the RR will perform the BGP best path calculation from the perspective of the ingress border router. The RR which can only do this is the RR that has the complete picture of the topology of the network from the IGP perspective where the RR and ingress border router(s) are located. For this requirement to be met, the IGP must be a link-state routing protocol.

In that case, the RR can run a Shortest Path First (SPF) calculation with the ingress border router as the root of the tree and calculate the cost to every other router. This way, the cost from the ingress border router to all other egress border routers will be known. This special SPF calculation with another router as the root, is referred to as a Reverse SPF (rSPF). This can only be done if the RR learns all the BGP paths from all the BGP border routers. There could be as many rSPFs run as there are RR clients. This will increase the CPU load somewhat on the RR.

The solution allows for the best path calculation to be based on the BGP best path selection algorithm, which will lead to the RR picking the best path from the perspective of the ingress border router the RR sends the path to. This means that the best path will be picked based on the shortest IGP cost to the BGP next hop. The solution also allows for the best path to be picked

based on some configured policy. The ingress border routers could pick their best paths based on some configured policy, and not on the lowest IGP cost. The solution allows the RR to implement the optimal route reflection either on the IGP cost (location on the network) or on some configured policy, or both. If both are deployed, then the policy is applied first and then the IGP-based optimal route reflection will occur on the remaining paths.

IOS-XR Implementation

The IOS-XR implementation allows up to three root nodes for the rSPF calculation. If you have many RR clients in one update group, then there is no need for one rSPF calculation per RR client if those RR clients will have the same policy and/or the same IGP costs to the different egress BGP border routers. This latter usually means that the RR clients are co-located (likely to be in the same POP). If that is the case, there is no need to configure each RR client as a root. The IOS-XR implementation allows to configure three, the primary, the secondary, and the tertiary root, per set of RR clients, for redundancy purposes. For the BGP ORR feature to apply to any RR client, that RR client must be configured to be part of an ORR policy group.

The BGP ORR feature is enabled per address family.

A link-state protocol is required. It can be OSPF or IS-IS.

IOS XR only implements the BGP ORR feature based on the IGP cost to the BGP next hop, and not based on some configured policy.

The BGP peers with the same outbound policy are placed in the same update group. This is usually the case for iBGP on the RR. When the feature BGP ORR is enabled, then the peers from different ORR groups will be in different update groups. This is logical, because the updates sent from the RR to the RR clients in different BGP ORR groups will be different, because the BGP best path is different.

The result of the rSPF calculations is stored in a database.

ORRSPF is the new component in IOS-XR which is needed for the BGP ORR feature. ORRSPF takes care off:

1. Collecting the link-state information and maintaining the Link-state database
2. Running rSPFs, and maintaining the SPTs, per policy group
3. Downloading the prefixes from the SPT to the RIB with the metrics

The database gets its link-state information either directly from the link-state IGP or from BGP-LS.

The rSPF calculations result in a topology showing the shortest path from the RR client to any other router in the area/level.

The routes hanging off every router in the topology are stored in a special RIB table for the ORR group policy and per AFI/SAFI. This table is created by RSI. The table is populated by the routes calculated by the rSPFs with the primary root as the root. If the primary root becomes unavailable, then the secondary root is the root and populates the routes in the ORR RIB table. The same applies to the tertiary root.

Configure

The minimal configuration needed:

1. ORR needs to be enabled for the address-family of BGP, for specific groups of BGP neighbors
2. For each group of BGP neighbors, at least one root needs to be configured. Optionally, a secondary and tertiary root can be configured.
3. The redistribution of the ORR routes from the IGP into BGP needs to be enabled.

Configuration Example

As shown in the first image, the RR is an IOS-XR router with the BGP ORR feature.

All the other routers are running IOS. These routers do not have the BGP ORR feature.

PE1, PE2, and PE3 advertise the prefix 172.16.2.0/24 in AFI/SAFI 1/1 (IPv4 unicast). The RR is equally close to PE1 and PE2 than to PE3. The IGP cost of all the links is 1. The best path for this prefix is the one with R1 as next-hop because of the lowest BGP router ID.

```
RP/0/0/CPU0:RR#show bgp ipv4 unicast 172.16.2.0/24 bestpath-compare
BGP routing table entry for 172.16.2.0/24
Versions:
 Process           bRIB/RIB  SendTblVer
 Speaker                34          34
Last Modified: Mar  7 20:29:48.156 for 11:36:44
Paths: (3 available, best #1)
 Advertised to update-groups (with more than one peer):
   0.3
 Path #1: Received by speaker 0
 Advertised to update-groups (with more than one peer):
   0.3
 Local, (Received from a RR-client)
   10.100.1.1 (metric 3) from 10.100.1.1 (10.100.1.1)
     Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
     Received Path ID 0, Local Path ID 1, version 34
     best of local AS, Overall best
  Path #2: Received by speaker 0
 Not advertised to any peer
 Local, (Received from a RR-client)
   10.100.1.2 (metric 3) from 10.100.1.2 (10.100.1.2)
     Origin IGP, metric 0, localpref 100, valid, internal, add-path
     Received Path ID 0, Local Path ID 6, version 33
     Higher router ID than best path (path #1)
  Path #3: Received by speaker 0
 ORR bestpath for update-groups (with more than one peer):
   0.1
 Local, (Received from a RR-client)
   10.100.1.3 (metric 5) from 10.100.1.3 (10.100.1.3)
     Origin IGP, metric 0, localpref 100, valid, internal, add-path
     Received Path ID 0, Local Path ID 7, version 34
     Higher IGP metric than best path (path #1)
```

PE4 will receive the path with PE1 as next-hop. So, there is no hot-potato routing for PE4.

If you want to have hot-potato routing on PE4, then for the prefixes advertised by PE1, PE2, and PE3 (for example the prefix 172.16.2.0/24), then PE1 should have PE3 as the exit point. This means that the path on PE4 should be the one with PE3 as next-hop. We can make the RR send the route with next-hop PE3 to PE4 with this ORR configuration.

```
router ospf 1
distribute bgp-ls
 area 0
 interface Loopback0
 !
 interface GigabitEthernet0/0/0/0
  network point-to-point
 !
 !
!

router bgp 1
 address-family ipv4 unicast
 optimal-route-reflection ipv4-orr-group 10.100.1.4
 !
 address-family vpnv4 unicast
 !
 neighbor 10.100.1.1
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  route-reflector-client
 !
 !
 neighbor 10.100.1.2
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  route-reflector-client
 !
 !
 neighbor 10.100.1.3
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  route-reflector-client
 !
 !
 neighbor 10.100.1.4
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group
   route-reflector-client
 !
 !
 neighbor 10.100.1.5
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  route-reflector-client
 !
 !
!
```

If the IGP is IS-IS:

```
router isis 1
net 49.0001.0000.0000.0008.00
 distribute bgp-ls
address-family ipv4 unicast
metric-style wide
```

```
!
interface Loopback0
address-family ipv4 unicast
!
!
interface GigabitEthernet0/0/0/0
address-family ipv4 unicast
!
!
!
```

> **Note**: The address family link-state does not need to be configured, globally or under the BGP neighbor(s).

MPLS Traffic-Engineering on Root Router

The RR needs to find the configured root address in the IGP database, in order to run the rSPF. In ISIS, the router-ID is present in the ISIS database. For OSPF, there is no router-ID present in the OSPF LSAs. The solution is to have the root routers advertise the Multi Protocol Label Switching (MPLS) TE router-ID matching the configured root address on the RR.

For OSPF, the root routers need additional configuration to make BGP ORR work. A minimal MPLS TE configuration is needed on any root router in order to advertise this MPLS TE router-ID. The exact minimal set of command depends on the operating system of the root router. The MPLS TE configuration on the root router needs to have the minimal configuration for MPLS TE enabled so that OSPF advertises the MPLS TE router ID in an opaque-area LSA (type 10).

Once the RR has an opaque-area LSA with the MPLS TE router-ID matching the configured root router address, rSPF can run and BGP on the RR can advertise the optimal route.

The minimal configuration needed for OSPF on the root router if it is an IOS router is:

```
!
interface GigabitEthernet0/2
 ip address 10.1.34.4 255.255.255.0
 ip ospf network point-to-point
mpls traffic-eng tunnels
!

router ospf 1
mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 10.200.1.155
 network 10.0.0.0 0.255.255.255 area 0
!
```

Notice that:

- MPLS TE is enabled in the specific OSPF area
- the MPLS TE router-ID is configured matching the configured root address on the RR
- MPLS TE is configured on at least one interface
- there is no need to have RSVP-TE configured
- there is no need to have MPLS TE configured on any other router in the area

The minimal configuration needed for OSPF on the root router if it is an IOS-XR router is:

```
!
router ospf 1
 router-id 5.6.7.8
 area 0
 mpls traffic-eng
  interface Loopback0
 !
 interface GigabitEthernet0/0/0/0
  network point-to-point
 !
 !
mpls traffic-eng router-id 10.100.1.11
!
mpls traffic-eng
!
```

If the above configuration is in place on the root router, then RR should have the MPLS TE router-ID in the OSPF database.

```
RP/0/0/CPU0:RR#show ospf 1 database


            OSPF Router with ID (10.100.1.99) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.12.1       10.1.12.1       1297        0x8000002b 0x006145 3
10.100.1.2      10.100.1.2      646         0x80000025 0x00fb6f 7
10.100.1.3      10.100.1.3      1693        0x80000031 0x003ba9 5
10.100.1.99     10.100.1.99     623         0x8000001e 0x00ade1 3
10.200.1.155    10.200.1.155    28          0x80000002 0x009b2e 5

                Type-10 Opaque Link Area Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Opaque ID
1.0.0.0         10.200.1.155    34          0x80000001 0x00a1ad        0

1.0.0.3         10.200.1.155    34          0x80000001 0x0057ff        3

RP/0/0/CPU0:RR#show ospf 1 database opaque-area adv-router 10.200.1.155


            OSPF Router with ID (10.100.1.99) (Process ID 1)

                Type-10 Opaque Link Area Link States (Area 0)

  LS age: 184
 Options: (No TOS-capability, DC)
 LS Type: Opaque Area Link
 Link State ID: 1.0.0.0
 Opaque Type: 1
 Opaque ID: 0
 Advertising Router: 10.200.1.155
 LS Seq Number: 80000001
 Checksum: 0xa1ad
 Length: 28

   MPLS TE router ID : 10.100.1.4

   Number of Links : 0
```

```
 LS age: 184
 Options: (No TOS-capability, DC)
 LS Type: Opaque Area Link
 Link State ID: 1.0.0.3
 Opaque Type: 1
 Opaque ID: 3
 Advertising Router: 10.200.1.155
 LS Seq Number: 80000001
 Checksum: 0x57ff
 Length: 132

   Link connected to Point-to-Point network
     Link ID : 10.100.1.3      (all bandwidths in bytes/sec)
     Interface Address : 10.1.34.4
     Neighbor Address : 10.1.34.3
     Admin Metric : 1
     Maximum bandwidth : 125000000
     Maximum reservable bandwidth global: 0
     Number of Priority : 8
     Priority 0 :                    0  Priority 1 :                    0
     Priority 2 :                    0  Priority 3 :                    0
     Priority 4 :                    0  Priority 5 :                    0
     Priority 6 :                    0  Priority 7 :                    0
     Affinity Bit : 0
     IGP Metric : 1

   Number of Links : 1
```

Notice that the MPLS TE router-ID (10.100.1.4) and the OSPF router-ID are different.

PE4 has PE3 as next-hop for the prefix (with the correct IGP metric to the next-hop):

```
PE4#show bgp ipv4 unicast 172.16.2.0
BGP routing table entry for 172.16.2.0/24, version 37
Paths: (1 available, best #1, table default)
 Not advertised to any peer
 Refresh Epoch 1
 Local
   10.100.1.3 (metric 2) from 10.100.1.8 (10.100.1.8)
     Origin IGP, metric 0, localpref 100, valid, internal, best
     Originator: 10.100.1.3, Cluster list: 10.100.1.8
     rx pathid: 0, tx pathid: 0x0
```

PE5 still has PE1 as the next-hop for the prefix (with the correct IGP metric to the next-hop):

```
PE5#show bgp ipv4 unicast 172.16.2.0/24
BGP routing table entry for 172.16.2.0/24, version 13
Paths: (1 available, best #1, table default)
 Not advertised to any peer
 Refresh Epoch 1
 Local
   10.100.1.1 (metric 3) from 10.100.1.8 (10.100.1.8)
     Origin IGP, metric 0, localpref 100, valid, internal, best
     Originator: 10.100.1.1, Cluster list: 10.100.1.8
     rx pathid: 0, tx pathid: 0x0
```

Troubleshoot

Verify the prefix on the RR:

```
RP/0/0/CPU0:RR#show bgp ipv4 unicast 172.16.2.0
BGP routing table entry for 172.16.2.0/24
Versions:
 Process           bRIB/RIB  SendTblVer
 Speaker                 19          19
Last Modified: Mar  7 16:41:20.156 for 03:07:40
Paths: (3 available, best #1)
 Advertised to update-groups (with more than one peer):
   0.3
 Path #1: Received by speaker 0
 Advertised to update-groups (with more than one peer):
   0.3
 Local, (Received from a RR-client)
   10.100.1.1 (metric 3) from 10.100.1.1 (10.100.1.1)
     Origin IGP, metric 0, localpref 100, valid, internal, best, group-best
     Received Path ID 0, Local Path ID 1, version 14
 Path #2: Received by speaker 0
 Not advertised to any peer
 Local, (Received from a RR-client)
   10.100.1.2 (metric 3) from 10.100.1.2 (10.100.1.2)
     Origin IGP, metric 0, localpref 100, valid, internal, add-path
     Received Path ID 0, Local Path ID 4, version 14
 Path #3: Received by speaker 0
 ORR bestpath for update-groups (with more than one peer):
   0.1
 Local, (Received from a RR-client)
   10.100.1.3 (metric 5) from 10.100.1.3 (10.100.1.3)
     Origin IGP, metric 0, localpref 100, valid, internal, add-path
     Received Path ID 0, Local Path ID 5, version 19
```

Note that add-path was added to the other non-best paths, so that they can be advertised as well, besides the best path. The add path feature is not used between the RR and its clients: the paths are not advertised with a path identifier.

Verify that the route(s) are (still) advertised to the specific BGP neighbors.

To the neighbor PE4, the next-hop is PE3 for the prefix 172.16.2.0/24:

```
RP/0/0/CPU0:RR#show bgp ipv4 unicast neighbors 10.100.1.4 advertised-routes
Network            Next Hop        From            AS Path
172.16.1.0/24      10.100.1.5      10.100.1.5      i
172.16.2.0/24      10.100.1.3      10.100.1.3      i

Processed 2 prefixes, 2 paths
```

To the neighbor PE5, the next-hop is PE1 for the prefix 172.16.2.0/24:

```
RP/0/0/CPU0:RR#show bgp ipv4 unicast neighbors 10.100.1.5 advertised-routes
Network            Next Hop        From            AS Path
172.16.1.0/24      10.100.1.8      10.100.1.5      i
172.16.2.0/24      10.100.1.1      10.100.1.1      i
```

The neighbor 10.100.1.4 is in its own update-group because of the ORR policy in place:

```
RP/0/0/CPU0:RR#show bgp ipv4 unicast update-group
Update group for IPv4 Unicast, index 0.1:
 Attributes:
```

```
   Neighbor sessions are IPv4
   Internal
   Common admin
   First neighbor AS: 1
   Send communities
   Send GSHUT community if originated
   Send extended communities
   Route Reflector Client
   ORR root (configured): ipv4-orr-group; Index: 0
   4-byte AS capable
   Non-labeled address-family capable
   Send AIGP
   Send multicast attributes
   Minimum advertisement interval: 0 secs
 Update group desynchronized: 0
 Sub-groups merged: 0
 Number of refresh subgroups: 0
 Messages formatted: 8, replicated: 8
 All neighbors are assigned to sub-group(s)
   Neighbors in sub-group: 0.1, Filter-Groups num:1
    Neighbors in filter-group: 0.3(RT num: 0)
     10.100.1.4

Update group for IPv4 Unicast, index 0.3:
 Attributes:
   Neighbor sessions are IPv4
   Internal
   Common admin
   First neighbor AS: 1
   Send communities
   Send GSHUT community if originated
   Send extended communities
   Route Reflector Client
   4-byte AS capable
   Non-labeled address-family capable
   Send AIGP
   Send multicast attributes
   Minimum advertisement interval: 0 secs
 Update group desynchronized: 0
 Sub-groups merged: 1
 Number of refresh subgroups: 0
 Messages formatted: 12, replicated: 42
 All neighbors are assigned to sub-group(s)
   Neighbors in sub-group: 0.3, Filter-Groups num:1
    Neighbors in filter-group: 0.1(RT num: 0)
     10.100.1.1               10.100.1.2               10.100.1.3
10.100.1.5
```

The show orrspf database command shows the ORR group an its root(s),

```
RP/0/0/CPU0:RR#show orrspf database

ORR policy: ipv4-orr-group, IPv4, RIB tableid: 0xe0000012
Configured root: primary: 10.100.1.4, secondary: NULL, tertiary: NULL
Actual Root: 10.100.1.4


Number of mapping entries: 1
```

The same command with the detail keyword provides the cost of the root of the rSPF to each other router/prefix in the same OSPF area:

```
RP/0/0/CPU0:RR#show orrspf database detail

ORR policy: ipv4-orr-group, IPv4, RIB tableid: 0xe0000012
Configured root: primary: 10.100.1.4, secondary: NULL, tertiary: NULL
Actual Root: 10.100.1.4

Prefix                                        Cost
10.100.1.6                                    2
10.100.1.1                                    3
10.100.1.2                                    3
10.100.1.3                                    2
10.100.1.4                                    0
10.100.1.5                                    3
10.100.1.7                                    3
10.100.1.8                                    4

Number of mapping entries: 9
```

The table-id was assigned by RSI for the ORR group, and for the AFI/SAFI:

```
RP/0/0/CPU0:RR#show rsi table-id 0xe0000012

TBL_NAME=ipv4-orr-group, AFI=IPv4, SAFI=Ucast TBL_ID=0xe0000012 in VRF=default/0x60000000 in
VR=default/0x20000000
Refcnt=1
VRF Index=4 TCM Index=1
Flags=0x0 LST Flags=(0x0) NULL

RP/0/0/CPU0:RR#show rib tables

Codes: N - Prefix Limit Notified, F - Forward Referenced
       D - Table Deleted, C - Table Reached Convergence

VRF/Table              SAFI  Table ID     PrfxLmt   PrfxCnt TblVersion  N F D C
default/default        uni   0xe0000000   5000000        22        128  N N N Y
**nVSatellite/default  uni   0xe0000010   5000000         2          4  N N N Y
default/ipv4-orr-grou  uni   0xe0000012   5000000         9         27  N N N Y
default/default        multi 0xe0100000   5000000         0          0  N N N Y
```

The cost of the root (R4/10.100.1.4) of the rSPF to each other router is the same as the cost that is seen with **show ip route ospf** on PE4:

```
PE4#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 20 subnets, 2 masks
O        10.100.1.1/32 [110/3] via 10.1.7.6, 4d05h, GigabitEthernet0/1
O        10.100.1.2/32 [110/3] via 10.1.7.6, 4d05h, GigabitEthernet0/1
O        10.100.1.3/32 [110/2] via 10.1.8.3, 4d06h, GigabitEthernet0/2
O        10.100.1.5/32 [110/3] via 10.1.7.6, 4d05h, GigabitEthernet0/1
O        10.100.1.6/32 [110/2] via 10.1.7.6, 4d05h, GigabitEthernet0/1
O        10.100.1.7/32 [110/3] via 10.1.8.3, 4d06h, GigabitEthernet0/2
```

```
                     [110/3] via 10.1.7.6, 4d05h, GigabitEthernet0/1
O        10.100.1.8/32 [110/4] via 10.1.8.3, 4d05h, GigabitEthernet0/2
                     [110/4] via 10.1.7.6, 4d05h, GigabitEthernet0/1
```

The RIB for the BGP ORR group:

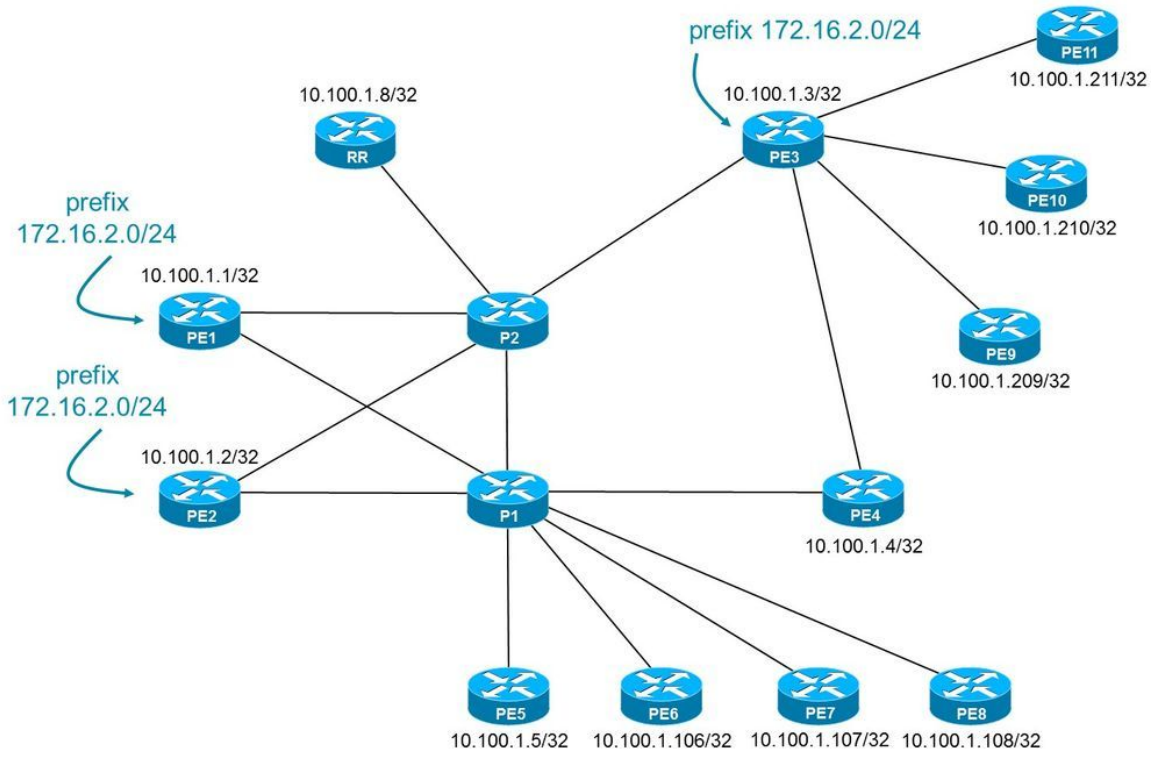
```
RP/0/0/CPU0:RR#show route afi-all safi-all topology ipv4-orr-group

IPv4 Unicast Topology ipv4-orr-group:
-------------------------------------

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, (!) - FRR Backup path

Gateway of last resort is not set

o    10.100.1.1/32 [255/3] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.2/32 [255/3] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.3/32 [255/2] via 0.0.0.0, 00:04:53, Unknown
o    10.100.1.4/32 [255/0] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.5/32 [255/3] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.6/32 [255/2] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.7/32 [255/3] via 0.0.0.0, 14:14:52, Unknown
o    10.100.1.8/32 [255/4] via 0.0.0.0, 14:14:52, Unknown

RP/0/0/CPU0:RR#show rsi table name ipv4-orr-group

VR=default:

TBL_NAME=ipv4-orr-group, AFI=IPv4, SAFI=Ucast TBL_ID=0xe0000012 in VRF=default/0x60000000 in
VR=default/0x20000000
Refcnt=1
VRF Index=4 TCM Index=1
Flags=0x0 LST Flags=(0x0) NULL
```

The **show bgp neighbor** command shows if the peer is an ORR root:

```
RP/0/0/CPU0:RR#show bgp neighbor 10.100.1.4

BGP neighbor is 10.100.1.4
 Remote AS 1, local AS 1, internal link
 Remote router ID 10.100.1.4
 Cluster ID 10.100.1.8
 BGP state = Established, up for 01:17:41
 NSR State: None
 Last read 00:00:52, Last read before reset 01:18:30
 Hold time is 180, keepalive interval is 60 seconds
 Configured hold time: 180, keepalive: 60, min acceptable hold time: 3
 Last write 00:00:34, attempted 19, written 19
 Second last write 00:01:34, attempted 19, written 19
 Last write before reset 01:17:43, attempted 19, written 19
 Second last write before reset 01:18:43, attempted 19, written 19
 Last write pulse rcvd  Mar  8 10:20:13.779 last full not set pulse count 12091
 Last write pulse rcvd before reset 01:17:42
 Socket not armed for io, armed for read, armed for write
```

```
 Last write thread event before reset 01:17:42, second last 01:17:42
 Last KA expiry before reset 01:17:43, second last 01:18:43
 Last KA error before reset 00:00:00, KA not sent 00:00:00
 Last KA start before reset 01:17:43, second last 01:18:43
 Precedence: internet
 Non-stop routing is enabled
 Multi-protocol capability received
 Neighbor capabilities:
   Route refresh: advertised (old + new) and received (old + new)
   4-byte AS: advertised and received
   Address family IPv4 Unicast: advertised and received
 Received 6322 messages, 0 notifications, 0 in queue
 Sent 5782 messages, 4 notifications, 0 in queue
 Minimum time between advertisement runs is 0 secs
 Inbound message logging enabled, 3 messages buffered
 Outbound message logging enabled, 3 messages buffered

 For Address Family: IPv4 Unicast
 BGP neighbor version 41
 Update group: 0.1 Filter-group: 0.1  No Refresh request being processed
 Route-Reflector Client
 ORR root (configured): ipv4-orr-group; Index: 0
  Route refresh request: received 0, sent 0
 0 accepted prefixes, 0 are bestpaths
 Cumulative no. of prefixes denied: 0.
 Prefix advertised 2, suppressed 0, withdrawn 0
 Maximum prefixes allowed 1048576
 Threshold for warning message 75%, restart interval 0 min
 AIGP is enabled
 An EoR was received during read-only mode
 Last ack version 41, Last synced ack version 0
 Outstanding version objects: current 0, max 2
 Additional-paths operation: None
 Send Multicast Attributes
 Advertise VPNv4 routes enabled with  option
 Advertise VPNv6 routes is enabled with Local with stitching-RT option

 Connections established 6; dropped 5
 Local host: 10.100.1.8, Local port: 25176, IF Handle: 0x00000000
 Foreign host: 10.100.1.4, Foreign port: 179
 Last reset 01:17:42, due to User clear requested (CEASE notification sent - administrative
reset)
 Time since last notification sent to neighbor: 01:17:42
 Error Code: administrative reset
 Notification data sent:
   None
```

As shown in this image, multiple set of RR clients configured



There is one set of RR clients connected to PE3 and another set connected to P1. Every RR client in each set is at equal distance to any egress BGP border router.

```
router bgp 1
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-1 10.100.1.4 10.100.1.209 10.100.1.210
  optimal-route-reflection ipv4-orr-group-2 10.100.1.5 10.100.1.106 10.100.1.107
 !
…
neighbor 10.100.1.4
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-1
   route-reflector-client
 !
 !
 neighbor 10.100.1.5
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-2
   route-reflector-client
 !
 !
neighbor 10.100.1.106
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-2
   route-reflector-client
 !
 !
 neighbor 10.100.1.107
 remote-as 1
 update-source Loopback0
```

```
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-2
   route-reflector-client
 !
 !
 neighbor 10.100.1.108
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-2
   route-reflector-client
 !
 !
 neighbor 10.100.1.209
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-1
   route-reflector-client
 !
 !
 neighbor 10.100.1.210
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-1   route-reflector-client
 !
 !
 neighbor 10.100.1.211
 remote-as 1
 update-source Loopback0
 address-family ipv4 unicast
  optimal-route-reflection ipv4-orr-group-1
   route-reflector-client
 !
 !
!
```

The orrspf database for both groups:

```
RP/0/0/CPU0:RR#show orrspf database detail

ORR policy: ipv4-orr-group-1, IPv4, RIB tableid: 0xe0000012
Configured root: primary: 10.100.1.4, secondary: 10.100.1.209, tertiary: 10.100.1.210
Actual Root: 10.100.1.4

Prefix                                          Cost
10.100.1.1                                      3
10.100.1.2                                      3
10.100.1.3                                      2
10.100.1.4                                      0
10.100.1.5                                      3
10.100.1.6                                      2
10.100.1.7                                      3
10.100.1.8                                      4
10.100.1.106                                    3
10.100.1.107                                    3
10.100.1.108                                    3
10.100.1.209                                    3
10.100.1.210                                    3
10.100.1.211                                    3
ORR policy: ipv4-orr-group-2, IPv4, RIB tableid: 0xe0000013
Configured root: primary: 10.100.1.5, secondary: 10.100.1.106, tertiary: 10.100.1.107
```

```
Actual Root: 10.100.1.5

Prefix                                   Cost
10.100.1.1                               3
10.100.1.2                               3
10.100.1.3                               4
10.100.1.4                               3
10.100.1.5                               0
10.100.1.6                               2
10.100.1.7                               3
10.100.1.8                               4
10.100.1.106                             3
10.100.1.107                             3
10.100.1.108                             3
10.100.1.209                             5
10.100.1.210                             5
10.100.1.211                             5

Number of mapping entries: 30
```

If for a group the primary root is down or unreachable, then the secondary root will be the actual root used. In this example, the primary root of group ipv4-orr-group-1 is unreachable. The secondary root became the actual root:

```
RP/0/0/CPU0:RR#show orrspf database ipv4-orr-group-1

ORR policy: ipv4-orr-group-1, IPv4, RIB tableid: 0xe0000012
Configured root: primary: 10.100.1.4, secondary: 10.100.1.209, tertiary: 10.100.1.210
Actual Root: 10.100.1.209

Prefix                                   Cost
10.100.1.1                               4
10.100.1.2                               5
10.100.1.3                               2
10.100.1.5                               5
10.100.1.6                               4
10.100.1.7                               3
10.100.1.8                               4
10.100.1.106                             5
10.100.1.107                             5
10.100.1.108                             5
10.100.1.209                             0
10.100.1.210                             3
10.100.1.211                             3

Number of mapping entries: 14
```

Conclusion

BGP Optimal Route Reflection (ORR) is a feature that allows hot-potato routing in a iBGP network when route reflectors are present, without the need for newer operating system software on the border routers. The prerequisite is that the IGP is a link-state routing protocol.