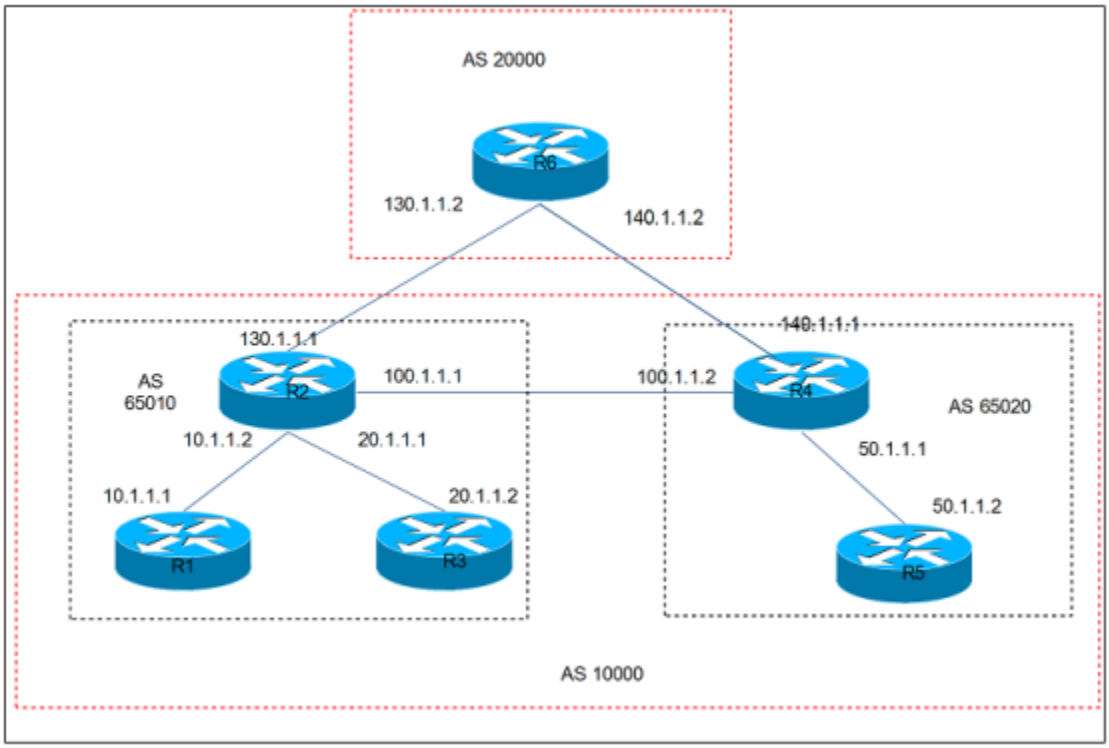
BGP Confederations and Route Reflectors – A design overview.

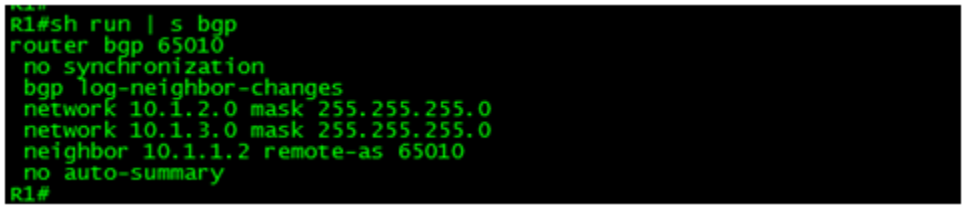
Folks welcome back, in this section we would design a network that employs confederations and route-reflectors. Confederations help us divide a larger BGP autonomous system into smaller autonomous systems. Route-reflectors on the other hand are employed to suppress the split horizon rule of BGP (split horizon in BGP stops a router from advertizing a route it learned from an iBGP peer to another neighbor, because iBGP expects full mesh connectivity between all the routers running it).



To illustrate how these concepts come into play in a real network design, check out the diagram mentioned above. There are two main autonomous systems (AS) here, AS 10000 and AS 20000 (indicated by a red dotted line).I have split up AS 10000 into two confederation autonomous systems, AS 65010 to the left and AS 65020 to the right. Routers R1, R2 and R3 form a part of the confederation 65010 and router R4 & R5 form a part of confed 65020, (indicated by black dotted line encircling these routers). R6 is placed in an external AS of 20000 and has peering at two points with AS 10000, on router R2 and on router R4. Note that all the confederation autonomous system

numbers should be in the range of 64512 to 65535, and are locally significant (think private IP address blocks)

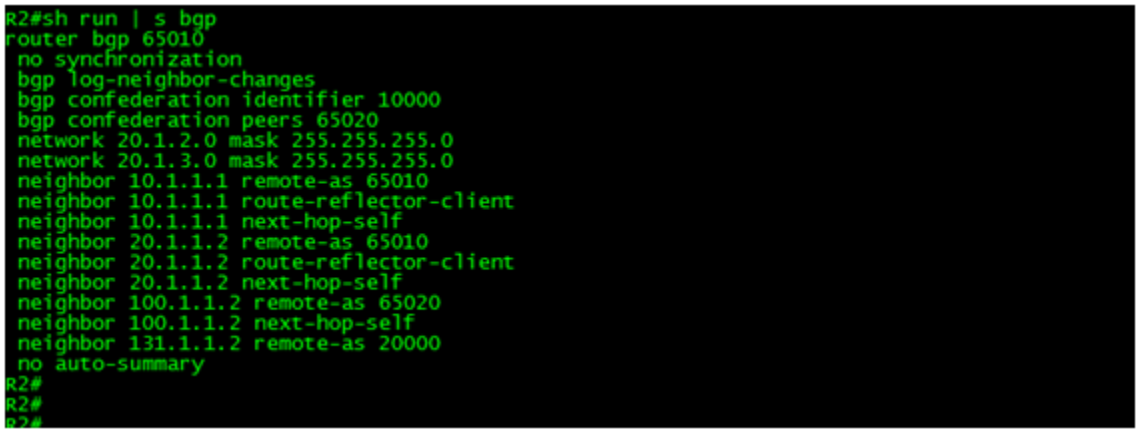
Mentioned below is R1's configuration:



Note that the BGP process of member routers takes the AS number of the confederation AS, hence R1's running 65010.

R3 and R5 would have similar configurations as that of R1; all are internal confederation routers with no direct peering to the external networks.

Now let's take a look at the R2's configuration.



Most of the statements on R2 are self explanatory except for "bgp confederation identifier 10000" and "bgp confederation peer 65020". Confederation identifier config identifies R2 as a part of the main autonomous system 10000 to the external router R6 on AS 20000. Note that this command is necessary only on the routers that interface with the external network in a confederation (for instance R1, R3 and R5 does not need this configuration). bgp confederation config on the other hand identifies R3 as a part of the confederation 65010 to R4, a similar config is needed on R4 to identify itself as a part of the confed 65020 to R2.

Taking a look at the R4's config, it looks just like a mirror image of R3. Note that the bgp confederation peer on R4 is set to 64010 to mirror R3's config's.

```
R4#sh run | s bgp
router bgp 65020
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 10000
 bgp confederation peers 65010
 network 40.1.1.0 mask 255.255.255.0
 network 40.1.2.0 mask 255.255.255.0
 neighbor 50.1.1.2 remote-as 65020
 neighbor 50.1.1.2 next-hop-self
 neighbor 100.1.1.1 remote-as 65010
 neighbor 100.1.1.1 next-hop-self
 neighbor 140.1.1.2 remote-as 20000
 no auto-summary
R4#
R4#
```

Simple enough! Now let's take a look at R6's config's and its routing tables.

```
R6#sh run | s bgp
router bgp 20000
 no synchronization
 bgp log-neighbor-changes
 network 130.1.2.0 mask 255.255.255.0
 network 130.1.3.0 mask 255.255.255.0
 neighbor 131.1.1.1 remote-as 10000
 neighbor 140.1.1.1 remote-as 10000
 no auto-summary
R6#
```

R6 is unaware of any complexity within the AS 10000 and peers with the AS on R2 and R4; R6 sees the AS just as AS 10000.

Further looking at the routing table for R6 we see that all the routes R6 has learned are learned through autonomous system 10000 regardless of which confederation within autonomous system 10000 they originated from.

```
R6#sh ip bgp
BGP table version is 67, local router ID is 130.1.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  10.1.2.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                              0 10000 i
*  10.1.3.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                              0 10000 i
*  20.1.2.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                0             0 10000 i
*  20.1.3.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                0             0 10000 i
*  30.1.3.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                              0 10000 i
*  30.1.4.0/24      140.1.1.1                              0 10000 i
*>                  131.1.1.1                              0 10000 i
*  40.1.1.0/24      131.1.1.1                              0 10000 i
*>                  140.1.1.1                0             0 10000 i
*  40.1.2.0/24      131.1.1.1                              0 10000 i
*>                  140.1.1.1                0             0 10000 i
*  50.1.2.0/24      131.1.1.1                              0 10000 i
*>                  140.1.1.1                              0 10000 i
*  50.1.3.0/24      131.1.1.1                              0 10000 i
*>                  140.1.1.1                              0 10000 i
*> 130.1.2.0/24     0.0.0.0                  0         32768 i
*> 130.1.3.0/24     0.0.0.0                  0         32768 i
R6#
R6#
R6#
```

That completes our discussion on BGP confederations.

Let's briefly take a look at the route reflectors now, consider router R1, R2 and R3 from the architecture, per BGP split horizon rule, R2 should not advertize the routes learned from R1 to any of its downstream routers, which in this case is R3. Rule was put in place because the creators of BGP saw that iBGP is deployed only on service provider cores, so full mesh connectivity of the routers was made mandatory. Which means there are is no need for one router to advertize its routes learned via iBGP to another neighbor. But when the routers are not run in full mesh connections, this rule becomes a pain in the a**. By making a peer a route reflector client, split horizon rule is selectively suppressed for that particular peer.

Now let's remove the route reflector config from the R2;

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router bgp 65010
R2(config-router)#no  neighbor 10.1.1.1 route-reflector-client
R2(config-router)#
*Mar  1 20:29:28.107: %BGP-5-ADJCHANGE: neighbor 10.1.1.1 Down RR client config change
R2(config-router)#n
*Mar  1 20:29:30.315: %BGP-5-ADJCHANGE: neighbor 10.1.1.1 Up
R2(config-router)#
R2(config-router)#no  neighbor 20.1.1.2 route-reflector-client
R2(config-router)#exit
```

Mentioned below is a snapshot of R1's BGP topology table with the route-reflector client removed from R2. Note that R2 drops all the routes that it learned from iBGP neighbor R3 from being advertized to R1, but advertizes the routes that are learned via other EBGP neighbors to R1. As a result R3's subnets 30.1.1.0/21 disappears from the routing table of R1.

```
R1#
R1#sh ip bgp
BGP table version is 43, local router ID is 10.1.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.2.0/24      0.0.0.0                  0         32768 i
*> 10.1.3.0/24      0.0.0.0                  0         32768 i
*>i40.1.1.0/24      10.1.1.2                 0    100      0 (65020) i
*>i40.1.2.0/24      10.1.1.2                 0    100      0 (65020) i
*>i50.1.2.0/24      10.1.1.2                 0    100      0 (65020) i
*>i50.1.3.0/24      10.1.1.2                 0    100      0 (65020) i
*>i130.1.2.0/24     10.1.1.2                 0    100      0 20000 i
*>i130.1.3.0/24     10.1.1.2                 0    100      0 20000 i
R1#
R1#
R1#
```

Similarly R3 also gets all other routes, but for the ones originated by R1 (being advertized by R2 of course). Hence the subnets 10.1.0.0 /22 subnets vanish from the routing table.

```
R3#sh ip bgp
BGP table version is 75, local router ID is 30.1.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i20.1.2.0/24      20.1.1.1                 0    100      0 i
*>i20.1.3.0/24      20.1.1.1                 0    100      0 i
*> 30.1.3.0/24      0.0.0.0                  0         32768 i
*> 30.1.4.0/24      0.0.0.0                  0         32768 i
*>i40.1.1.0/24      20.1.1.1                 0    100      0 (65020) i
*>i40.1.2.0/24      20.1.1.1                 0    100      0 (65020) i
*>i50.1.2.0/24      20.1.1.1                 0    100      0 (65020) i
*>i50.1.3.0/24      20.1.1.1                 0    100      0 (65020) i
*>i130.1.2.0/24     20.1.1.1                 0    100      0 20000 i
*>i130.1.3.0/24     20.1.1.1                 0    100      0 20000 i
R3#
```

Let's now try putting the configurations back and see how it impacts the routing table

```
R2#
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router bgp 65010
R2(config-router)#neighbor 10.1.1.1 route-reflector-client
R2(config-router)#neighbor 20.1.1.2 route-reflector-client
R2(config-router)#exit
R2(config)#exit
```

Once the change is made R1 starts seeing the 30.1.0.0/21 network once again,

```
R1#sh ip bgp
BGP table version is 11, local router ID is 10.1.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inter
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i20.1.2.0/24      10.1.1.2                 0    100      0 i
*>i20.1.3.0/24      10.1.1.2                 0    100      0 i
*>i30.1.3.0/24      10.1.1.2                 0    100      0 i
*>i30.1.4.0/24      10.1.1.2                 0    100      0 i
*>i40.1.1.0/24      10.1.1.2                 0    100      0 (65020) i
*>i40.1.2.0/24      10.1.1.2                 0    100      0 (65020) i
*>i50.1.2.0/24      10.1.1.2                 0    100      0 (65020) i
*>i50.1.3.0/24      10.1.1.2                 0    100      0 (65020) i
*>i130.1.2.0/24     10.1.1.2                 0    100      0 20000 i
*>i130.1.3.0/24     10.1.1.2                 0    100      0 20000 i
R1#
R1#
```

Similarly we see the 10.1.0.0/22 subnet network once again on R3's routing tables.

```
R3#sh ip bgp
BGP table version is 25, local router ID is 30.1.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i10.1.2.0/24      20.1.1.1                 0    100      0 i
*>i10.1.3.0/24      20.1.1.1                 0    100      0 i
*>i20.1.2.0/24      20.1.1.1                 0    100      0 i
*>i20.1.3.0/24      20.1.1.1                 0    100      0 i
*> 30.1.3.0/24      0.0.0.0                  0         32768 i
*> 30.1.4.0/24      0.0.0.0                  0         32768 i
*>i40.1.1.0/24      20.1.1.1                 0    100      0 (65020) i
*>i40.1.2.0/24      20.1.1.1                 0    100      0 (65020) i
*>i50.1.2.0/24      20.1.1.1                 0    100      0 (65020) i
*>i50.1.3.0/24      20.1.1.1                 0    100      0 (65020) i
*>i130.1.2.0/24     20.1.1.1                 0    100      0 20000 i
*>i130.1.3.0/24     20.1.1.1                 0    100      0 20000 i
R3#
```

Note that there were some route-maps created on R2 to achieve the next hop changes to the network that were advertized between R1 and R3, without which the routes would not be accessible. I am not showing this step to limit the length of the article. The next-hop-self command on BGP peering relationships does not apply for route-reflector based routes (and Cisco and juniper opines it as a feature and not a bugJ).

Source: http://ciscoworks.wordpress.com/2010/08/30/bgp-confederations-and-route-reflectors-%E2%80%93-a-design-overview/