A Study in BGP Confederations

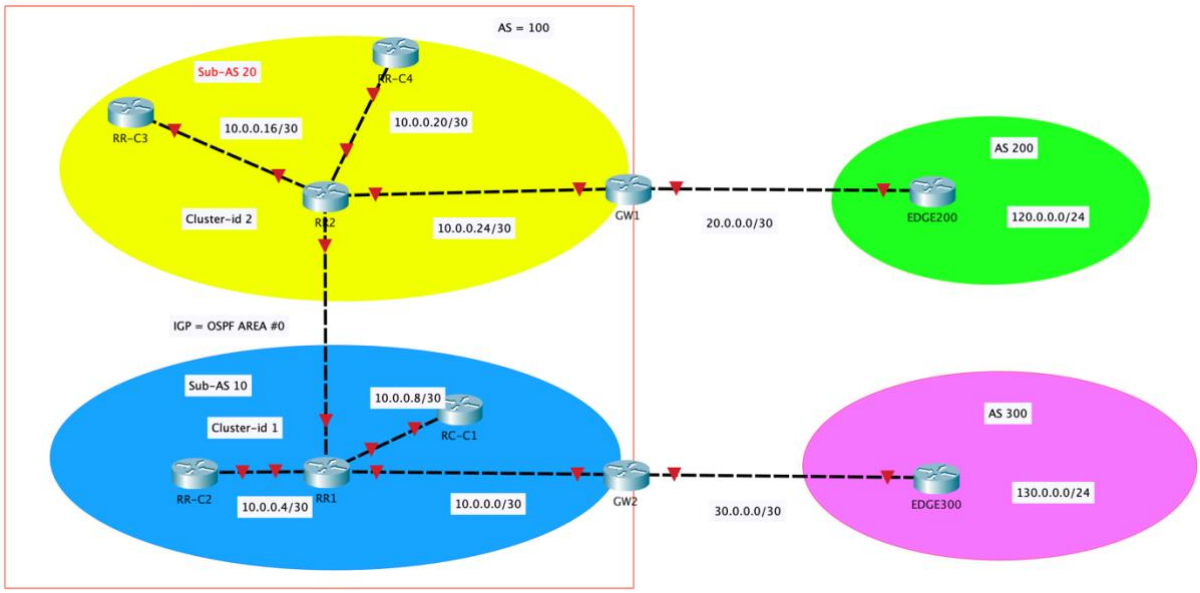
1. Context

BGP confederations can be an alternative to reduce the number of iBGP-TCP peers. Here the idea is that an AS is divided into smaller "sub-AS-s". These sub-AS areas are now no longer subject to the *split-horizon rule* which states that an iBGP routing-update may not be forwarded to another iBGP peer, since they reside within the same AS. (loop – avoidance)

Since both Peers are now in different sub-AS numbers, the split-horizon rule no longer applies.

2. Topology

In the following topology I have created two sub-AS numbers (10 and 20) within AS #100. Just to spice up things a bit, I have added route-reflectors within each sub-AS number even though that is strictly speaking not needed.



3. The configurations

On the "ISP" routers EDGE200 and EDGE300, the BGP configuration is very straight forward:

```
EDGE200# show run | s router bgp
router bgp 200
 bgp router-id 9.9.9.9
 bgp log-neighbor-changes
 network 120.0.0.0 mask 255.255.255.0
 neighbor 20.0.0.2 remote-as 100
```

and:

```
EDGE300# show run | s router
router bgp 300
 bgp router-id 10.10.10.10
 bgp log-neighbor-changes
  network 130.0.0.0 mask 255.255.255.0
 neighbor 30.0.0.2 remote-as 100
```

A Study in BGP Confederations

On GW1 and GW2, (both part of a sub-AS), the configuration has a wee twist:

```
GW1# show run | s router bgp
router bgp 20
 bgp router-id 8.8.8.8
 bgp log-neighbor-changes
 bgp confederation identifier 100
 network 8.8.8.8 mask 255.255.255.255
 neighbor 10.0.0.25 remote-as 20
 neighbor 10.0.0.25 next-hop-self
 neighbor 20.0.0.1 remote-as 200
```

and:

```
GW2# show run | s router bgp
router bgp 10
 bgp router-id 1.1.1.1
 bgp log-neighbor-changes
 bgp confederation identifier 100
 neighbor 10.0.0.2 remote-as 10
 neighbor 10.0.0.2 next-hop-self
 neighbor 30.0.0.1 remote-as 300
```

Notice the difference with the regular BGP configuration:
The BGP process now uses its **SUB-AS** number for its own AS number. It DOES reference the official AS number with "`bgp confederation identifier 100`".
The remote-as numbers are the AS numbers of all neighboring BGP processes. (both iBGP and eBGP)

On the route-reflectors that reside between the 2 sub-AS areas, the configuration deviates more:

```
RR1# show run | s router bgp
router bgp 10
 bgp router-id 2.2.2.2
 bgp cluster-id 1
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 20
 network 2.2.2.2 mask 255.255.255.255
 neighbor 10.0.0.1 remote-as 10
 neighbor 10.0.0.1 next-hop-self
 neighbor 10.0.0.6 remote-as 10
 neighbor 10.0.0.6 route-reflector-client
 neighbor 10.0.0.6 next-hop-self
 neighbor 10.0.0.10 remote-as 10
 neighbor 10.0.0.10 route-reflector-client
 neighbor 10.0.0.10 next-hop-self
 neighbor 10.0.0.14 remote-as 20
 neighbor 10.0.0.14 route-reflector-client
 neighbor 10.0.0.14 next-hop-self
```

The cluster-id is used for route-reflectors: since we only have 1 here per cluster, this statement is strictly speaking not needed. (avoids routing-loops in case there are multiple route-reflectors within the same cluster)
Notice the "`bgp confederation peers 20`" which specifies the remote sub-AS number.

The same thing applies for the 2nd route-reflector:

```
RR2# show run | s router bgp
router bgp 20
 bgp router-id 5.5.5.5
 bgp cluster-id 2
 bgp log-neighbor-changes
 bgp confederation identifier 100
 bgp confederation peers 10
 neighbor 10.0.0.13 remote-as 10
 neighbor 10.0.0.13 route-reflector-client
 neighbor 10.0.0.13 next-hop-self
 neighbor 10.0.0.18 remote-as 20
 neighbor 10.0.0.18 route-reflector-client
 neighbor 10.0.0.22 remote-as 20
 neighbor 10.0.0.22 next-hop-self
 neighbor 10.0.0.26 remote-as 20
 neighbor 10.0.0.26 next-hop-self
```

The remarkable take-away from this, is that you do not (with iBGP) peer on the AS number, but instead on the **SUB-AS** number!

Of course, this being iBGP we could have peered via the Loopback interfaces, but I couldn't be bothered.

4. Proof is in the pudding

Notice the AS-PATH from RR2 to 130.0.0./24:

```
RR2#show ip bgp
[…]
     Network          Next Hop            Metric LocPrf Weight Path
*>  130.0.0.0/24     10.0.0.13                0    100      0  (10) 300 i
```

Here the AS-PATH first reflects the SUB – AS (10) and then the regular AS 300.

And of course "ping" works between the ISP routers:

```
EDGE300# traceroute 120.0.0.1 so lo1
Type escape sequence to abort.
Tracing the route to 120.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 30.0.0.2 0 msec 0 msec 0 msec
  2 10.0.0.2 1 msec 0 msec 0 msec
  3 10.0.0.14 1 msec 0 msec 1 msec
  4 10.0.0.26 0 msec 1 msec 0 msec
  5 20.0.0.1 1 msec *  2 msec
```