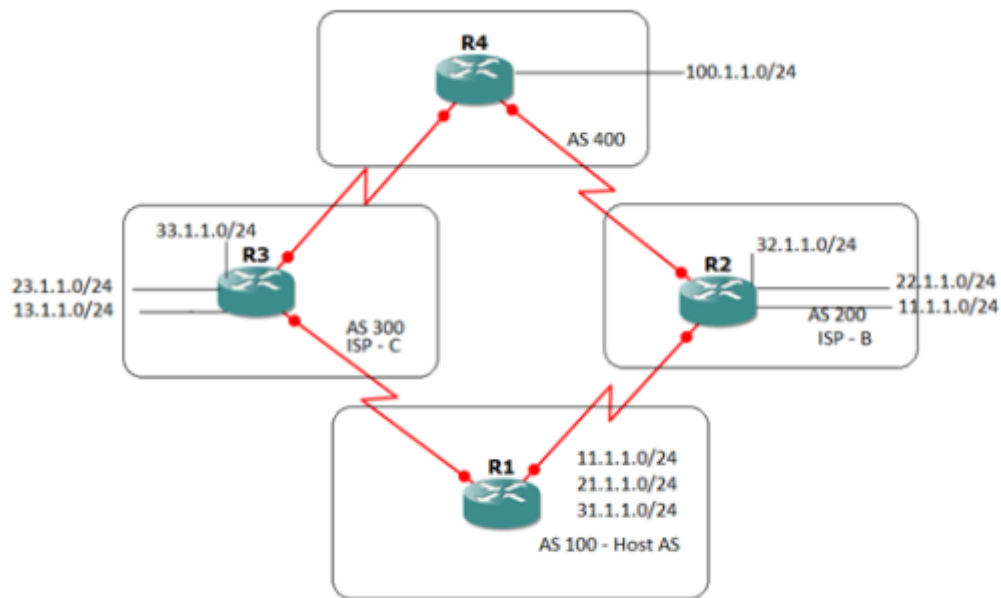


BGP design for a dual homed enterprise – the bare minimums!

We are back with another article and this time I would take a look at a typical enterprise network that connects to more than one ISP's (I would consider two) for redundancy. One of main concerns in any typical dual homed BGP network is preventing your own AS from being a transit network. In this article we would take a look at how this can be achieved. In addition to that, most enterprises would like to prefer one ISP over the other for each directional streams of traffic. For instance, in the diagram that's shown below, AS 100 which happens to be a Enterprise network (say) and would want to use ISP-B(AS 200) for its incoming traffic and would want to use ISP B (AS 300) for its outgoing traffic, but of course you should have each of them capable of handling both the streams when one of the ISP's is inaccessible for some reason.

With this basic goals let's begin designing and implementing this network.



I have already deployed BGP on all four routers R1, R2, R3 and R4. I have created some internal routes on each of these routers that would provide us with some meat to work with,

Shown below is the complete BGP table for R1 host router, before changing any of the BGP attributes on the router.

```

r1#sh ip bgp
BGP table version is 74, local router ID is 31.1.1.1
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
*> 11.1.1.0/24    0.0.0.0          0           32768 i
*> 12.1.1.0/24    60.1.1.2         0           0 200 i
*> 13.1.1.0/24    50.1.1.2         0           0 300 i
*> 21.1.1.0/24    0.0.0.0          0           32768 i
*> 22.1.1.0/24    60.1.1.2         0           0 200 i
*> 23.1.1.0/24    50.1.1.2         0           0 300 i
*> 31.1.1.0/24    0.0.0.0          0           32768 i
*> 32.1.1.0/24    60.1.1.2         0           0 200 i
*> 33.1.1.0/24    50.1.1.2         0           0 300 i
* 100.1.1.0/24    60.1.1.2         0           0 200 400 i
*>                50.1.1.2         0           0 300 400 i
r1#
r1#

```

note that 100.1.1.0/24 exists in AS 400, which has peering relationship with both the ISP's and hence the BGP process on R1 is learning it from both AS 200 and AS 300.

Coming to the first design aspect which is to prevent the host network (R1 in AS 100) from being a transit network, to achieve that we should stop the subnets advertised by ISP A from reaching ISP B and vice versa through R1. We facilitate this by matching the AS path for both the ISP AS paths using as-path access lists and then by using them in a deny route-map, which we would then implement on the neighbor's in the outbound direction, as shown below.

```

ip as-path access-list 100 permit ^$
ip as-path access-list 200 permit ^200
ip as-path access-list 300 permit ^300

```

```

!
route-map AS_300_subnets_blocked deny 10
match as-path 300
!
route-map AS_300_subnets_blocked permit 15
!
route-map AS_200_subnets_blocked deny 10
description RM stops subnet from AS200 from reaching AS300
match as-path 200
!
route-map AS_200_subnets_blocked permit 15
!

```

```

!
router bgp 100
no synchronization
bgp log-neighbor-changes
network 11.1.1.0 mask 255.255.255.0
network 21.1.1.0 mask 255.255.255.0
network 31.1.1.0 mask 255.255.255.0
neighbor 50.1.1.2 remote-as 300
neighbor 50.1.1.2 route-map Prefer_AS300_outgoing in
neighbor 50.1.1.2 route-map AS_200_subnets_blocked out
neighbor 60.1.1.2 remote-as 200
neighbor 60.1.1.2 route-map AS_300_subnets_blocked out
no auto-summary
!

```

That stops R2's and R3's routes being advertised into one another using R1, hence prevents R1 from becoming a transit for both the ISP's.

Now moving on to the second part of the design objective, that is preferring AS 200 as the outbound ISP and AS 300 as the inbound ISP. Note that there are two parts to it, in the former we are talking about traffic leaving AS 100 and in the latter we refer to traffic that enters AS 100.

To make AS 300 as the preferred path for all outbound traffic we match the routes being advertised by AS 300 and set the local preference to any value above 100 (this can also be achieved by changing the cisco proprietary attribute of "weight")

On

R1

```

!
ip as-path access-list 300 permit ^300
!

```

And

```
!
route-map Prefer_AS300_outgoing permit 10
match as-path 300
set local-preference 120
!
```

Note that the route with best local preference is placed in the routing table.

```
r1#sh ip bgp
BGP table version is 11, local router ID is 31.1.1.1
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
*> 11.1.1.0/24     0.0.0.0           0         0 32768 i
*> 12.1.1.0/24     60.1.1.2           0         0 200 i
*> 13.1.1.0/24     50.1.1.2           0        120  0 300 i
*> 21.1.1.0/24     0.0.0.0           0         0 32768 i
*> 22.1.1.0/24     60.1.1.2           0         0 200 i
*> 23.1.1.0/24     50.1.1.2           0        120  0 300 i
*> 31.1.1.0/24     0.0.0.0           0         0 32768 i
*> 32.1.1.0/24     60.1.1.2           0         0 200 i
*> 33.1.1.0/24     50.1.1.2           0        120  0 300 i
* 100.1.1.0/24     60.1.1.2           0         0 200 400 i
*>                 50.1.1.2           0        120  0 300 400 i
r1#
```

Note the LocPref value changed to 120 on all the routes learned from AS 300.

```
31.1.1.0/24 is suppressed, 1 subnets
r1# sh ip bgp 100.1.1.1
BGP routing table entry for 100.1.1.0/24, version 11
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  200 400
    60.1.1.2 from 60.1.1.2 (32.1.1.1)
      Origin IGP, localpref 100, valid, external
  300 400
    50.1.1.2 from 50.1.1.2 (33.1.1.1)
      Origin IGP, localpref 120, valid, external, best
r1#
```

Also note that the route 100.1.1.0/24 is now preferred via AS 300 than via AS 200.

That completes our second goal, now moving on to the third and last aspect of the design, which is to make AS 300 less preferable (or AS 200 more preferable) for external traffic into the host AS.

To achieve this, we would create another as-path access list to collect all the locally advertized subnets. Once done, we would then use them in a route-map to prepend the last/local AS number a couple of times, we would then use the neighbor statements on the ISP neighbor command that needs to be less preferred.

```
!
ip as-path access-list 100 permit ^$
```

```
!
route-map Prefer_AS200_incoming permit 10
 match as-path 100
 set as-path prepend 100 100 100 100
!
```

```
r1#
r1#sh run | s bgp
router bgp 100
no synchronization
bgp log-neighbor-changes
network 11.1.1.0 mask 255.255.255.0
network 21.1.1.0 mask 255.255.255.0
network 31.1.1.0 mask 255.255.255.0
neighbor 50.1.1.2 remote-as 300
neighbor 50.1.1.2 route-map Prefer_AS300_outgoing in
neighbor 50.1.1.2 route-map Prefer_AS200_incoming out
```

This would prepend AS 100 to the AS paths that are advertised making it unlikely to be chosen as a return path, just to be sure let's take a look at R4's routing table to see how this may turn out.

```

r4#sh ip bgp regexp 100$
BGP table version is 30, local router ID is 100.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network          Next Hop           Metric LocPrf weight Path
 * 11.1.1.0/24      80.1.1.1           0 300 100 100 100 10
 0 i
  > 21.1.1.0/24     70.1.1.1           0 200 100 i
  > 21.1.1.0/24     80.1.1.1           0 300 100 100 100 10
 0 i
  > 31.1.1.0/24     70.1.1.1           0 200 100 i
  > 31.1.1.0/24     80.1.1.1           0 300 100 100 100 10
 0 i
  > 70.1.1.1        70.1.1.1           0 200 100 i
r4#

```

A look at the R4 reveals that all the routes that are learned from AS 300 has a longer AS path with more hops on it and those that are learned from 200 have a much shorter hop, hence routes out of 200 are preferred.

That completes the second part of this objective.

Note that this method of AS preference would work if the person/AS originating the traffic chooses to keep all other all other parameters such as local preference, weight, origin status at their defaults. However should he change those parameters, we might still end up receiving traffic in the ISP that we don't want to (just like the way we do it while sending our traffic out). Its best to keep your ISP in the loop while you set these preferences, its better yet if your ISP can do this for you (in which case you mark the routes to be treated a certain way with a community tag maybe?)

That completes the design session and we have achieved all our three goals.

Source: <http://ciscoworks.wordpress.com/2010/06/19/bgp-design-for-a-dual-homed-enterprise-%E2%80%93-the-bare-minimums/>