

Using MED and Local Preference to Influence Incoming and Outgoing Traffic

October 22, 2004

Prepared for: Felix Carapaica

Prepared by: Hoon Song

Shazad Bagha

Arif Noorani

Drew Farlinger

John Tewfik

Leif Madsen

Introduction

Theoretically, our understanding is that MED is used to influence the incoming traffic from a remote autonomous system. In compliments to that, Local Preference is used to influence the outgoing traffic from our autonomous system. This allows us to send outgoing traffic through a specific exit point. Our lab will explore the use of this within the Zebra BGP routing daemon and access-lists.

According to Internet Routing Architectures, the "attribute is an optional nontransitive attribute. It is a hint to external neighbors about the preferred path into an AS that has multiple entry points"¹. This is a key point as this will affect the creation of our incoming and outgoing route-maps within our BGP daemons.

Overview of the Lab

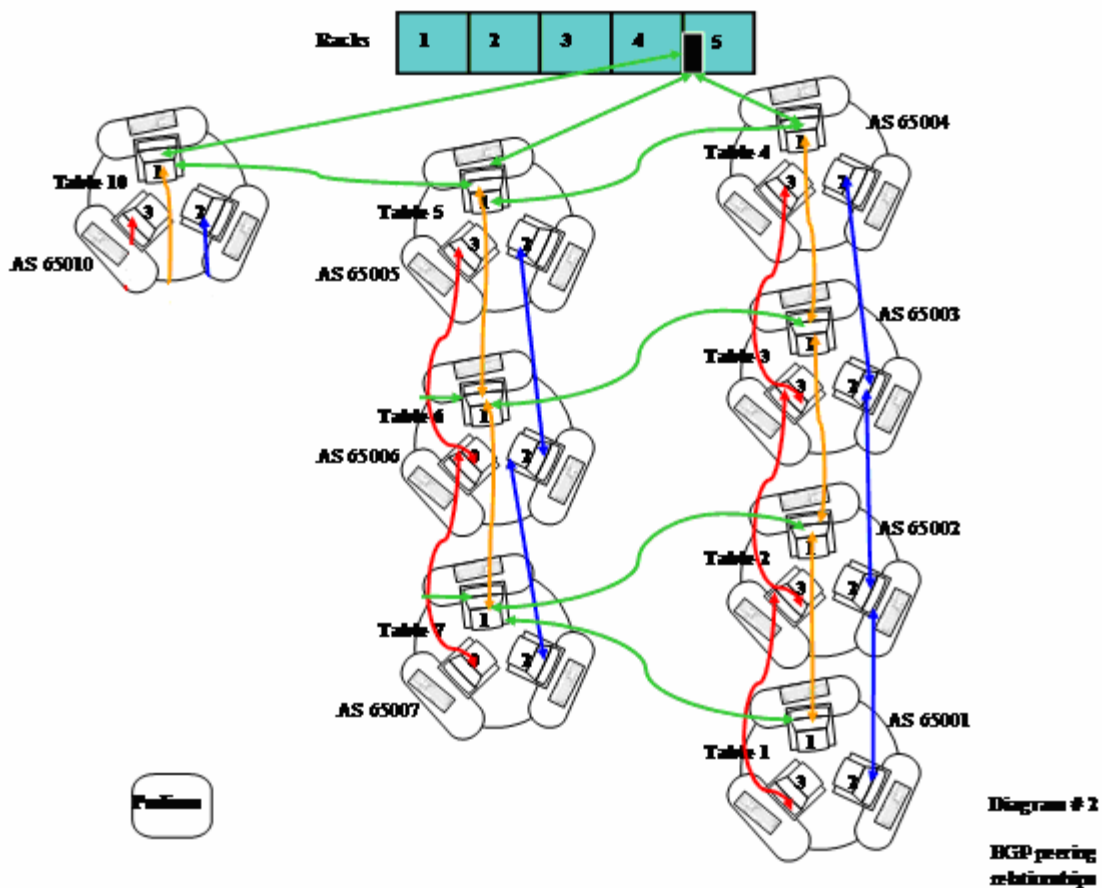


Diagram 1-1: Peering Arrangement²

For our lab, we were unable to reliably get tables 8 and 9 to work with our configurations. Table 9 worked on the lab at different time than us, so their configurations were not in sync with what we were doing. Instead of redoing the table (as they were able to do the lab, just not the same as us) we decided to

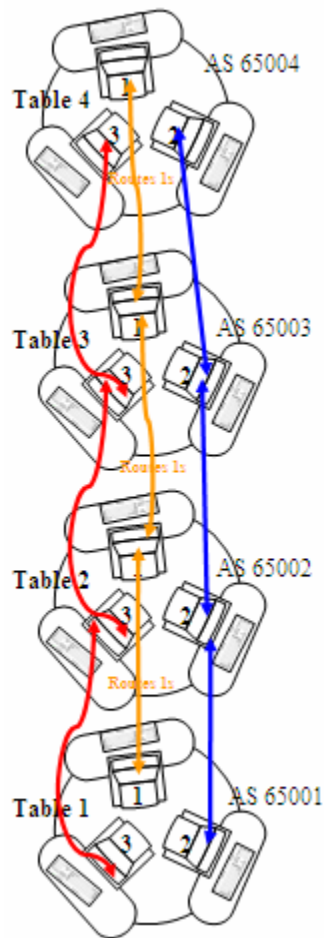
¹ Quoted from Internet Routing Architectures (Second Edition) by Sam Halabi. Copyright 2000, Cisco Press.

² Diagram taken from MED Lab #3 notes by Felix Carapaica. Edited to reflect removal of tables 8 and 9.

remove it from our topology. Table 8 had weird and sporadic problems that we did not wish to deal with on top of our own problems within the lab. Removing these two peers helped to simplify the lab in order to get it to work within a reasonable amount of time. However the most important peering arrangements of the lab (tables 4, 5 and 10) were satisfied.

Process

Peering arrangements for the various tables were first established as follows:



The first stage of the peering arrangement was done as viewed in diagram 1-2. Neighbor adjacency was established between table 4 and table 3. Table 3 then established with table 2, which in turn peered with table 1. The first router of each table was peered with the first router of the corresponding table as just previously described. Example 1-1 below shows the commands used to establish the peering adjacency between router 1 on table 4 with router 1 of table 3. The configurations for all routers are similar.

```
router bgp 65004
  bgp router-id 4.4.4.1
  network 4.4.4.1/32
  network 172.16.12.0/24
  network 192.168.4.0/24
  neighbor 172.16.12.3 remote-as 65003
  neighbor 172.16.12.3 next-hop-self
  neighbor 172.16.12.5 remote-as 65005
  neighbor 172.16.12.5 next-hop-self
  neighbor 192.168.4.2 remote-as 65004
  neighbor 192.168.4.3 remote-as 65004
```

Example Configuration 1-1: Table 4, Router 1

The above configuration sets the BGP router ID, the networks we are advertising (with the use of the `network` command) and the neighbors that we are establishing adjacency with. For remote peers (those not local to our AS) we use the `next-hop-self` command. This is used to tell my local peers how to get to the remote peer.

Essentially we are saying to our local peers, "If you need to get to network 'X', then come to me because I know how to route traffic for that network".

This process was repeated for the middle row of tables as well. Since table 10 had no other tables to peer with they were exempt from this process.

After the BGP processes were established, the top table imported the Internet routes from the Dell box. These routes were filtered as specified in the lab scenario. Table 4 imported network routes from the 1, 2, 3 and 4 networks but excluded 10.0.0.0/8 (this will be imported by table 10 – more on this later). These routes were then passed to the local peers and onwards down to tables 3, 2 and 1.

The following configuration, [example 1-2](#), was used in table 4 to filter the incoming routes. This made sure that only routes from networks starting with 1, 2, 3 or 4 were imported into the BGP daemon.

```
bgp router 65004
.
.
.
  neighbour 192.168.4.100 remote-as 65413          ! establish neighbour
  neighbour 192.168.4.100 route-map MARCO in      ! apply route-map MARCO
!
! Access lists to permit networks starting with 1, 2, 3 and 4.
! 10.0.0.0/8 has been omitted. In order to maintain IP reachability
! within the network the 192.0.0.0 network was also included within
! this access list
!
access-list 5 permit 1.0.0.0 0.255.255.255
access-list 5 permit 100.0.0.0 3.255.255.255
access-list 5 permit 104.0.0.0 7.255.255.255
access-list 5 permit 128.0.0.0 63.255.255.255
access-list 5 permit 2.0.0.0 0.255.255.255
access-list 5 permit 20.0.0.0 3.255.255.255
access-list 5 permit 24.0.0.0 3.255.255.255
access-list 5 permit 28.0.0.0 1.255.255.255
access-list 5 permit 200.0.0.0 7.255.255.255
access-list 5 permit 208.0.0.0 15.255.255.255
access-list 5 permit 224.0.0.0 31.255.255.255
access-list 5 permit 3.0.0.0 0.255.255.255
access-list 5 permit 30.0.0.0 1.255.255.255
access-list 5 permit 32.0.0.0 7.255.255.255
access-list 5 permit 4.0.0.0 0.255.255.255
access-list 5 permit 40.0.0.0 7.255.255.255
access-list 5 permit 48.0.0.0 1.255.255.255
access-list 5 permit 192.0.0.0 3.255.255.255
!
! create a route-map called MARCO and match against access-list 5
route-map MARCO permit 1
  match ip address 5
```

Example Configuration 1-2: Filtering incoming network routes starting with 1, 2, 3 and 4 (excluding 10.0.0.0/8)

The above configuration uses an access-list to filter the various networks. This is then applied to the 192.168.4.100 (Dell box) neighbour using a route-map. [Diagram 1-2](#) shows how the reverse mask was learned. Configurations for tables 5 and 10 are also similar.

Our following example will calculate the aggregated range of 50.0.0.0 through 59.0.0.0. The other network ranges are calculated using the same logic.

Begin 5s Supernet Calculation									
128	64	32	16	8	4	2	1	Address	Supernet
0	0	1	1	0	0	1	0	50	50.0.0.0/7
0	0	1	1	0	0	1	1	51	
0	0	1	1	0	1	0	0	52	52.0.0.0/6
0	0	1	1	0	1	0	1	53	
0	0	1	1	0	1	1	0	54	
0	0	1	1	0	1	1	1	55	
0	0	1	1	1	0	0	0	56	56.0.0.0/6
0	0	1	1	1	0	0	1	57	
0	0	1	1	1	0	1	0	58	
0	0	1	1	1	0	1	1	59	

Diagram 1-2: Calculating Supernets

A requirement of the lab was to engineer traffic so that traffic destined for certain networks would follow a specified path. This was acquired through the use of Local Preference. Our viewpoint is from router R2 on table 7.

When we traceroute a network starting with 5, 6 or 7, we expect to see the traffic following different paths. Networks starting with 5 should follow through the R1 at each table. Networks starting with 6 should follow through the R2 of each table and networks starting with 7 should follow through R3 of each table before converging with R1 at table 5 and eventually ending up at the Dell box. This is illustrated in [diagram 1-3](#) below.

```
[root@localhost root]# traceroute 61.04.0.0
traceroute to 61.04.0.0 (61.4.0.0), 30 hops max, 38 byte packets
 1 192.168.7.2 (192.168.7.2) 0.535 ms 0.445 ms 0.439 ms
 2 172.16.22.6 (172.16.22.6) 0.621 ms 0.692 ms 0.582 ms
 3 172.16.22.5 (172.16.22.5) 1.523 ms 1.104 ms 1.258 ms
 4 192.168.5.100 (192.168.5.100) 1.085 ms 0.622 ms 0.713 ms
```

```
[root@localhost root]# traceroute 59.105.224.0
traceroute to 59.105.224.0 (59.105.224.0), 30 hops max, 38 byte packets
 1 192.168.7.1 (192.168.7.1) 0.626 ms 0.445 ms 0.496 ms
 2 172.16.12.6 (172.16.12.6) 0.711 ms 0.796 ms 0.781 ms
 3 172.16.12.5 (172.16.12.5) 1.033 ms 1.007 ms 0.958 ms
 4 192.168.5.100 (192.168.5.100) 1.344 ms 1.201 ms 1.212 ms
```

```
[root@localhost root]# traceroute 70.145.80.0
traceroute to 70.145.80.0 (70.145.80.0), 30 hops max, 38 byte packets
 1 172.16.32.6 (172.16.32.6) 0.426 ms 0.397 ms 0.336 ms
 2 172.16.32.6 (172.16.32.6) 0.851 ms 0.816 ms 0.651 ms
 3 172.16.32.5 (172.16.32.5) 0.933 ms 1.117 ms 0.920 ms
 4 192.168.5.100 (192.168.5.100) 1.085 ms 1.622 ms 1.713 ms
```

Diagram 1-3: Traceroute to networks 5, 6 and 7 respectively

With the use of local preference we were able to manipulate the paths that the traffic followed. The following configuration, [example 1-3](#), was used to achieve this.

```
router bgp 65005
.
.
.
neighbor 192.168.5.1 route-map R3_to_R1_PREF out

route-map R3_to_R1_PREF permit 15
set local-preference 1000
```

Example Configuration 1-3: Applying local preference using route-map

The above configuration was used to apply a local preference to a local neighbour. The higher the number the more preferred the route. In this case what we are saying is that traffic destined for the Internet (the Dell box) is passed to router R1 on table 5 since it is peered with the Dell box. A similar configuration is done on router R2 of table 5.

Conclusion

We observed that IP traffic destined for various networks could be controlled through the use of local preference and MED. For our lab we chose to use the local preference attribute in order to control where the traffic was sent. Using a combination of access-lists and route-maps the local preference was applied to our local peers within our AS system in order to shape the traffic. It would have also been possible to try and influence the routes of the remote AS system peers with the use of a MED. Instead of telling the local peers where the traffic is supposed to go, we could "request" that our remote peers send traffic destined for certain networks to use specific links. This would have theoretically achieved the same situation. The key lesson learned was that local preference is used to influence traffic going "out" of your system whereas the MED attribute is used to influence traffic coming "in" to your system.

Appendix

Configurations for table 5 (most relevant)

```
*****BGP config R1 *****
Current configuration:
!
hostname bgp-t5c1
password welcome
enable password welcome
log file /usr/local/var/bgpd/bgpd.log
log stdout
!
router bgp 65005
  bgp router-id 5.5.5.1
  network 5.5.5.1/32
  network 172.16.12.0/24
  network 192.168.5.0/24
  neighbor 172.16.12.4 remote-as 65004
  neighbor 172.16.12.4 next-hop-self
  neighbor 172.16.12.4 route-map T5_to_T4 out
  neighbor 172.16.12.6 remote-as 65006
  neighbor 172.16.12.6 next-hop-self
  neighbor 172.16.12.6 route-map T5_to_T6 out
  neighbor 172.16.12.10 remote-as 65010
  neighbor 172.16.12.10 next-hop-self
  neighbor 172.16.12.10 route-map T5_to_T10 out
  neighbor 192.168.5.2 remote-as 65005
  neighbor 192.168.5.3 remote-as 65005
  neighbor 192.168.5.100 remote-as 65413
  neighbor 192.168.5.100 route-map T5-IN-FILTER in
!
access-list 5 permit 4.0.0.0 0.255.255.255
access-list 5 permit 40.0.0.0 7.255.255.255
access-list 5 permit 48.0.0.0 1.255.255.255
access-list 5 permit 5.0.0.0 0.255.255.255
access-list 5 permit 50.0.0.0 1.255.255.255
access-list 5 permit 52.0.0.0 3.255.255.255
access-list 5 permit 56.0.0.0 3.255.255.255
access-list 10 permit 5.0.0.0 0.255.255.255
access-list 10 permit 50.0.0.0 1.255.255.255
access-list 10 permit 52.0.0.0 3.255.255.255
access-list 10 permit 56.0.0.0 3.255.255.255
access-list 10 permit 192.168.0.0 0.0.255.255
access-list 15 permit 6.0.0.0 0.255.255.255
access-list 15 permit 60.0.0.0 3.255.255.255
access-list 15 permit 68.0.0.0 1.255.255.255
access-list 15 permit 7.0.0.0 0.255.255.255
access-list 15 permit 70.0.0.0 1.255.255.255
access-list 15 permit 72.0.0.0 7.255.255.255
access-list 15 permit 64.0.0.0 3.255.255.255
access-list 20 permit 10.0.0.0 0.255.255.255
!
route-map T5_to_T6 permit 10
  match ip address 10
  set metric 5
```

```
!  
route-map T5_to_T6 permit 15  
  match ip address 15  
  set metric 20  
!  
route-map T5-IN-FILTER permit 5  
  match ip address 15  
!  
route-map T5-IN-FILTER permit 7  
  match ip address 10  
!  
route-map T5_to_T4 permit 20  
  match ip address 20  
!  
route-map T5_to_T10 permit 5  
  match ip address 5  
!  
  
line vty  
!  
end
```

*****R2*****

```
Password:  
bgp-t5c2> en  
Password:  
bgp-t5c2# sh run
```

Current configuration:

```
!  
hostname bgp-t5c2  
password welcome  
enable password welcome  
log file /usr/local/var/bgpd/bgpd.log  
log stdout  
!  
router bgp 65005  
  bgp router-id 5.5.5.2  
  network 5.5.5.2/32  
  network 172.16.22.0/24  
  network 192.168.5.0/24  
  neighbor 172.16.22.6 remote-as 65006  
  neighbor 172.16.22.6 next-hop-self  
  neighbor 172.16.22.6 route-map T5_to_T6 out  
  neighbor 192.168.5.1 remote-as 65005  
  neighbor 192.168.5.1 route-map R3_to_R1_PREF out  
  neighbor 192.168.5.3 remote-as 65005  
!  
access-list 10 permit 5.0.0.0 0.255.255.255  
access-list 10 permit 50.0.0.0 1.255.255.255  
access-list 10 permit 52.0.0.0 3.255.255.255  
access-list 10 permit 56.0.0.0 3.255.255.255  
access-list 10 permit 6.0.0.0 0.255.255.255  
access-list 10 permit 60.0.0.0 3.255.255.255  
access-list 10 permit 64.0.0.0 3.255.255.255  
access-list 10 permit 68.0.0.0 1.255.255.255
```

```
access-list 10 permit 7.0.0.0 0.255.255.255
access-list 10 permit 70.0.0.0 1.255.255.255
access-list 10 permit 72.0.0.0 7.255.255.255
!
route-map T5_to_T6 permit 5
  match ip address 10
!
route-map R3_to_R1_PREF permit 15
  set local-preference 1000
!
line vty
!
end
```

```
*****R3*****
```

```
Password:
bgp-t5c3> en
Password:
bgp-t5c3# sh run
```

```
Current configuration:
!
hostname bgp-t5c3
password welcome
enable password welcome
log file /usr/local/var/bgpd/bgpd.log
log stdout
!
router bgp 65005
  bgp router-id 5.5.5.3
  network 5.5.5.3/32
  network 172.16.32.0/24
  network 192.168.5.0/24
  neighbor 172.16.32.6 remote-as 65006
  neighbor 172.16.32.6 next-hop-self
  neighbor 172.16.32.6 route-map T5_2_T6_MED out
  neighbor 192.168.5.1 remote-as 65005
  neighbor 192.168.5.1 route-map LOCPREF out
  neighbor 192.168.5.2 remote-as 65005
!
access-list 5 permit 5.0.0.0 0.255.255.255
access-list 5 permit 50.0.0.0 1.255.255.255
access-list 5 permit 52.0.0.0 3.255.255.255
access-list 5 permit 56.0.0.0 3.255.255.255
access-list 5 permit 6.0.0.0 0.255.255.255
access-list 5 permit 60.0.0.0 3.255.255.255
access-list 5 permit 64.0.0.0 3.255.255.255
access-list 5 permit 68.0.0.0 1.255.255.255
access-list 10 permit 7.0.0.0 0.255.255.255
access-list 10 permit 70.0.0.0 1.255.255.255
access-list 10 permit 72.0.0.0 7.255.255.255
!
route-map T5_2_T6_R3 permit 5
  match ip address 5
  set local-preference 500
!
```

```
route-map LOCPREF permit 10
  set local-preference 1000
!
route-map T5_2_T6_MED permit 10
  match ip address 10
  set metric 1
!
line vty
!
end
```