NOCTION
NETWORK INTELLIGENCE

BGP Inbound Traffic Engineering

Gaining control over your inbound traffic

www.noction.com

Table of Contents

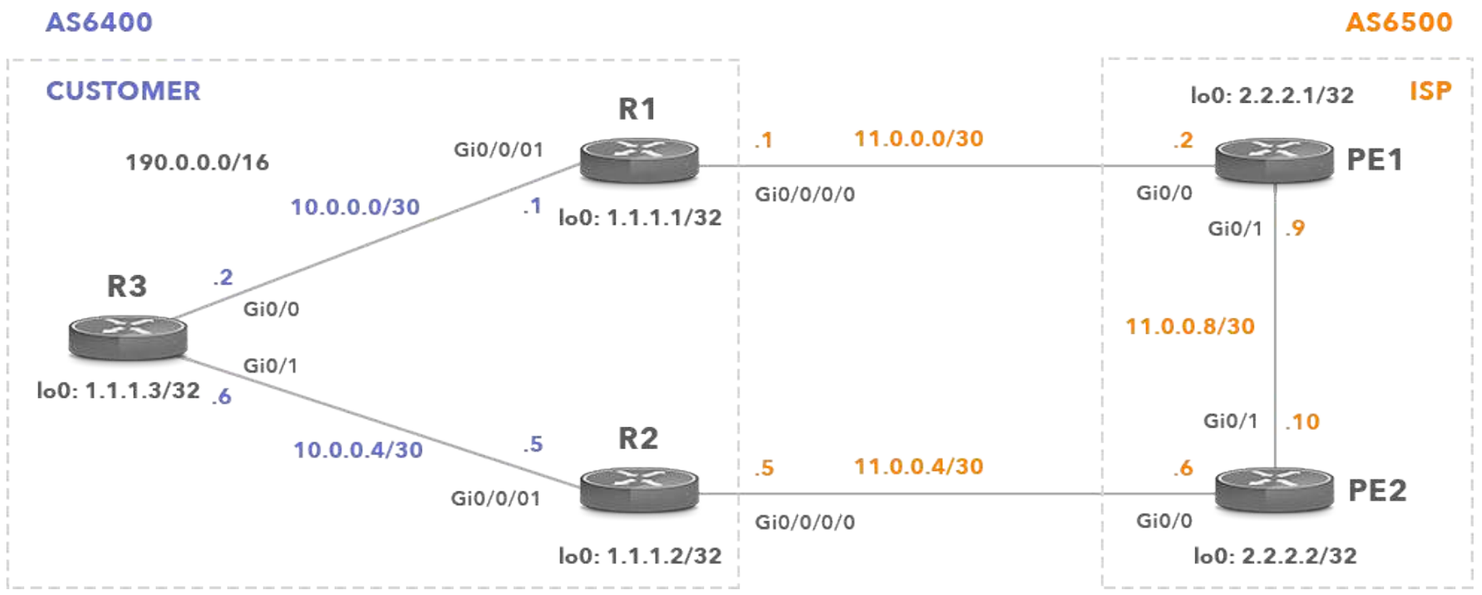
BGP Inbound Traffic Engineering

When there is more than one way for packets to enter a customer network, network operators might want to control this behavior for a number of reasons. For instance, a customer in AS6400 can have an uplink to its provider in AS6500. For redundancy reasons AS6400 may want to have another uplink to be used as a failover link only. The control over inbound traffic here becomes of an utmost importance, since any traffic generated on this link under normal operation costs the customer money, which must be paid in addition to a working primary link.

Unless customers have a specific agreement with providers that they peer with, changing the flow of incoming network traffic is much harder than influencing outgoing traffic. For outgoing traffic, customers can influence the best-path selection algorithm locally, as they are in possession of edge routers. For the inbound traffic however, customers must trick the best path selection process on the upstream provider level. This might be a challenge, as the upstream routing policies can override the BGP attributes sent in the updates to the providers.

To influence the inbound traffic path, customers can use certain attributes (such as MED, AS-PATH, BGP communities) in the updates sent to their providers. Another method is based on the longest prefix-matching behavior and can be accomplished by the BGP conditional route injection. This eBook covers all of the methods mentioned above as well as the automated method of inbound traffic engineering using Noction's Intelligent Routing Platform (IRP).

Below is the network topology that we are going to use in order to configure the well known methods of inbound traffic engineering.



Picture 1 - *Customer is Dual-Homed Towards a Single ISP*

Routers R1, R2 and R3 of the customer are in AS6400 running internal BGP (iBGP). The customer also runs OSPF within its network. Customer's network is dual-homed towards a single ISP (AS6500). The R1 and R2 (AS6400) routers have external BGP (eBGP) connection with two PE routers of the same ISP. Both PE1 and PE2 routers advertise only the default route 0.0.0.0 to the customer. R1 and R2 advertise the default route to R3 (Picture 2). The R3 router prefers the path via R2 to the path via R1 to the 0.0.0.0 prefix because R1 is the router with the lowest router ID.

NOTE: ISP provides only a default route to the customer so there is no advantage on having an iBGP session between the R1 and R2 routers. If the ISP provides some specific routes in addition to the default route it may be wise to configure the iBGP session between the R1 and R2 routers.

```
R3#
R3#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          1.1.1.1                  0    100      0 6500 i
 * i                  1.1.1.2                  0    100      0 6500 i
 *>  190.0.0.0        0.0.0.0                  0         32768 i
R3#
```

Picture 2 - *BGP Table R3*

The 190.0.0.0/16 prefix is a public IP address block assigned from the Regional Internet Registry (RIR) to the customer. The R3 router advertises this prefix to its iBGP peers R1 and R2. Both routers have installed the route into their BGP tables with a next-hop IP address 1.1.1.3 (loopback interface of R3). (Picture 3 and 4).

```
RP/0/0/CPU0:ios#
RP/0/0/CPU0:ios#show bgp | begin Network
Mon Aug  6 04:31:52.341 UTC
   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0/0        11.0.0.2                 0             0 6500 i
*>i190.0.0.0/16     1.1.1.3                  0    100      0 i

Processed 2 prefixes, 2 paths
RP/0/0/CPU0:ios#
```

Picture 3 - *BGP Table R1*

```
RP/0/0/CPU0:R2#
RP/0/0/CPU0:R2#show bgp | begin Network
Mon Aug  6 04:34:49.609 UTC
   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0/0        11.0.0.6                 0             0 6500 i
*>i190.0.0.0/16     1.1.1.3                  0    100      0 i

Processed 2 prefixes, 2 paths
RP/0/0/CPU0:R2#
```

Picture 4 - *BGP Table R2*

Both customer routers R1 and R2 advertise the route 190.0.0.0/16 towards ISP. The PE1 router has installed two paths to the prefix 190.0.0./16 into its BGP table (Picture 5). The path with the next-hop IP address 11.0.0.1 (R1) is preferred to the path with the next-

hop 2.2.2.2 (PE-2). PE1 prefers the eBGP route received from R1 to the iBGP route received from PE-2.

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 * i 190.0.0.0        2.2.2.2                  0    100      0 6400 i
 *>                   11.0.0.1                              0 6400 i
PE1#
```

Picture 5 - *BGP Table PE-1*

Similarly, PE-2 is preferring the eBGP route received from R2 (the next-hop IP 11.0.0.5) to the iBGP route received from PE-1, with the next-hop IP address 2.2.2.1 (the loopback interface of PE-1) (Picture 6).

```
PE2#
PE2#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.1                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  190.0.0.0        11.0.0.5                              0 6400 i
 * i                  2.2.2.1                  0    100      0 6400 i
PE2#
```

Picture 6 - *BGP Table PE-2*

In the parts to follow we'll present different methods that the customer can use to manipulate inbound traffic sent from ISP to the 190.0.0.0/16 prefix. Let's start with the initial configuration of all routers in our network.

1. Initial Configuration

All network devices in our topology are Cisco routers. The R3, PE-1 and PE-2 routers are running Cisco IOS version 15.6(2)T while the R1 and R2 routers run Cisco IOS XR, version 6.1.3.

1.1. R3 Configuration

OSPF and iBGP are using the IP addresses configured on the loopback interfaces so IP addresses configured on the physical interfaces are not utilized for establishing peering connection. In case, there are multiple connections and a physical link fails, OSPF and iBGP sessions are not teared down.

```
interface Loopback0
 ip address 1.1.1.3 255.255.255.255

interface GigabitEthernet0/0
 ip address 10.0.0.2 255.255.255.252

interface GigabitEthernet0/1
 ip address 10.0.0.6 255.255.255.252
```

The static route 190.0.0.0/16 to a null interface must be present in the routing table of R3 in order to advertise it with the network command in BGP configuration.

```
ip route 190.0.0.0 255.255.0.0 Null0
```

OSPF is used in the customer network to establish a BGP session (TCP sessions) and to resolve the BGP next hop.

```
router ospf 1
 network 1.1.1.3 0.0.0.0 area 0
 network 10.0.0.0 0.0.0.3 area 0
 network 10.0.0.4 0.0.0.3 area 0

router bgp 6400
 neighbor 1.1.1.1 remote-as 6400
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.1.1.2 remote-as 6400
 neighbor 1.1.1.2 update-source Loopback0

 address-family ipv4
  network 190.0.0.0
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.2 activate
```

1.2. R1 Configuration

```
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255

interface GigabitEthernet0/0/0/0
 ipv4 address 11.0.0.1 255.255.255.252

interface GigabitEthernet0/0/0/1
 ipv4 address 10.0.0.1 255.255.255.252
```

Route policies are mandatory for eBGP peers to import and export routes. The route-policy PASS passes all routes for processing. It is our default policy that we use before configuring any kind of incoming traffic manipulation.

```
route-policy PASS
 pass
end-policy

router ospf 1
 area 0
  interface Loopback0
  interface GigabitEthernet0/0/0/1

router bgp 6400
 address-family ipv4 unicast

 neighbor 1.1.1.3
  remote-as 6400
  update-source Loopback0
  address-family ipv4 unicast
    next-hop-self
```

The route-policy PASS for eBGP neighbor PE-1 is applied to both inbound and outbound routes.

```
 neighbor 11.0.0.2
  remote-as 6500
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
```

1.3. R2 Configuration

```
interface Loopback0
 ipv4 address 1.1.1.2 255.255.255.255

interface GigabitEthernet0/0/0/0
 ipv4 address 11.0.0.5 255.255.255.252

interface GigabitEthernet0/0/0/1
 ipv4 address 10.0.0.5 255.255.255.252

route-policy PASS
 pass
end-policy

router ospf 1
 area 0
  interface Loopback0
  interface GigabitEthernet0/0/0/1

router bgp 6400
 address-family ipv4 unicast

neighbor 1.1.1.3
 remote-as 6400
 update-source Loopback0
 address-family ipv4 unicast
   next-hop-self

neighbor 11.0.0.6
 remote-as 6500
 address-family ipv4 unicast
  route-policy PASS in
  route-policy PASS out
```

1.4. PE-1 Configuration

```
interface Loopback0
 ip address 2.2.2.1 255.255.255.0

interface GigabitEthernet0/0
 ip address 11.0.0.2 255.255.255.252

interface GigabitEthernet0/1
 ip address 11.0.0.9 255.255.255.252
router ospf 1
 network 2.2.2.1 0.0.0.0 area 0
 network 11.0.0.8 0.0.0.3 area 0
```

The static default route must be present in a routing table of PE-1 prior to its advertisement with the network command.

```
ip route 0.0.0.0 0.0.0.0 Null0
```

The prefix-list DEFAULT-PL matches a default route.

```
ip prefix-list DEFAULT-PL seq 5 permit 0.0.0.0/0
```

The route-map DEFAULT-RM matches a prefix list DEFAULT-PL. We will apply it for outbound routes sent to the customer.

```
route-map DEFAULT-RM permit 10
 match ip address prefix-list DEFAULT-PL
```

OSPF is used in the ISP network to establish a BGP session (TCP sessions) and to resolve the BGP next hop.

```
router ospf 1
 network 2.2.2.1 0.0.0.0 area 0
 network 11.0.0.8 0.0.0.3 area 0

router bgp 6500
 neighbor 2.2.2.2 remote-as 6500
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 11.0.0.1 remote-as 6400
```

The next-hop statement changes the original next-hop from 11.0.0.1 (R1) to 2.2.2.1 (PE-1) for routes sent to PE-2 (2.2.2.2). The route-map DEFAULT-RM is applied for outbound routes sent to the customer router R1. It ensures that ISP advertises only a default route towards the customer, instead of the full Internet routing table.

```
 address-family ipv4
   network 0.0.0.0
   neighbor 2.2.2.2 activate
   neighbor 2.2.2.2 next-hop-self
   neighbor 11.0.0.1 activate
   neighbor 11.0.0.1 route-map DEFAULT-RM out
```

1.5. PE-2 Configuration

```
interface Loopback0
 ip address 2.2.2.2 255.255.255.255

interface GigabitEthernet0/0
 ip address 11.0.0.6 255.255.255.252

interface GigabitEthernet0/1
 ip address 11.0.0.10 255.255.255.252
```

The static default route must be present in the routing table of PE-2 in order to advertise it with the network command

```
ip route 0.0.0.0 0.0.0.0 Null0
```

The prefix-list DEFAULT-PL matches the default route.

```
ip prefix-list DEFAULT-PL seq 5 permit 0.0.0.0/0
```

The route-map DEFAULT-RM matches the prefix list DEFAULT-PL.

```
route-map DEFAULT-RM permit 10
 match ip address prefix-list DEFAULT-PL
```

OSPF is used in the ISP network to establish the BGP session (TCP sessions) and to resolve the BGP next hop.

```
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 11.0.0.8 0.0.0.3 area 0

router bgp 6500
 neighbor 2.2.2.1 remote-as 6500
 neighbor 2.2.2.1 update-source Loopback0
 neighbor 11.0.0.5 remote-as 6400
```

The next-hop statement changes the original next-hop from 11.0.0.5 (R2) to 2.2.2.2 (PE-2) for routes advertised to PE-1 (2.2.2.1). The route-map DEFAULT-RM is applied for outbound routes sent to the customer router R2. It ensures that ISP advertises only a default route towards the customer, instead of the full Internet routing table.

```
address-family ipv4
  network 0.0.0.0
  neighbor 2.2.2.1 activate
  neighbor 2.2.2.1 next-hop-self
  neighbor 11.0.0.5 activate
  neighbor 11.0.0.5 route-map DEFAULT-RM out
```

2. Incoming Traffic Manipulation Using AS_PATH Prepending

The AS path is a well-known mandatory attribute of BGP. It is presented for all prefixes exchanged between any BGP neighbors. When a BGP router sends out an update to an eBGP neighbor, it adds its own AS number to the front (left side) of the AS path. As a result, the AS path lists all the ASes that need to be traversed to reach the location where the prefix (with the attached path) is advertised from.

The main purpose of the AS path is to avoid routing loops. AS loop detection is done by scanning the full AS path (as specified in the AS_PATH attribute), and checking that the AS number of the local system does not appear in the AS path.
The shortest AS_PATH is the fourth criterion that is used by the BGP path selection process to select a path between two similar paths with nearly the same local preference, weight and locally originated or aggregate addresses. Therefore, if we make the AS_PATH look longer than it actually is to a specific BGP peer, we can influence the BGP peer to select a particular path for the incoming traffic to our AS.

NOTE: AS path prepending adds one or more AS numbers to the left side of AS_PATH.

For the purpose of demonstration, we will configure R1 to prepend the AS_PATH with its own AS6400 multiple times for the advertised 190.0.0.0/16 prefix. As a result, the ISP router PE-1 will install the path via an iBGP peer PE-2 into its BGP table as the best path. In other words, PE-1 in AS6500 will prefer the shorter path through PE-2 to the customer (AS_PATH 6400) to the longer path through R1 (AS_PATH 6400 6400 6400).

2.1. AS_PATH Prepending Configuration on R1

First, we are going to create the prefix-set CUST-PS that is matching the prefix 190.0.0.0/16.

```
prefix-set CUST-PS
  190.0.0.0/16
end-set
```

Now we will create a route-policy AS-PATH-RP matching the prefix-set CUST-PS. If the prefix-set is matched, the route-policy prepends AS-PATH two times with AS6400. In case there were other prefixes advertised by R1, the route-policy would not modify their attributes.

```
route-policy AS-PATH-RPL
  if destination in CUST-PS then
    prepend as-path 6400 2
  else
    pass
  endif
end-policy
```

Let's apply the route-policy AS-PATH-RPL to the outbound routes for eBGP peer PE-1 (11.0.0.2) .

```
router bgp 6400
 address-family ipv4 unicast

neighbor 1.1.1.3
 remote-as 6400
 update-source Loopback0
 address-family ipv4 unicast
  next-hop-self

neighbor 11.0.0.2
 remote-as 6500
 address-family ipv4 unicast
  route-policy PASS in
  route-policy AS-PATH-RPL out
```

Now let's check the BGP table of the PE-1 router (Picture 7). PE-1 selected a path to the 190.0.0.0/16 prefix through the iBGP peer PE-2 (the next-hop 2.2.2.2) as the best-path. We have effectively influenced the BGP best path selection process on PE-1(AS6500) to select the shortest path via the iBGP neighbor PE-2 by prepending the AS_PATH multiple times on R1 for the advertised 190.0.0.0/16 route. The incoming traffic will be forwarded from PE-1 to PE-2 and finally through the router R2 to the customer.

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          0.0.0.0                  0         32768 i
 * i                  2.2.2.2                  0    100      0 i
 *   190.0.0.0        11.0.0.1                               0 6400 6400 6400 i
 *>i                  2.2.2.2                  0    100      0 6400 i
PE1#
```

Picture 7 - *PE-1 BGP Table After the Configuration of AS-PATH Prepending on R1*

NOTE: If we want to ignore the AS_PATH length evaluation step of the BGP path selection process on PE-1, we can configure PE-1 the following way:

```
router bgp 6500
  bgp bestpath as-path ignore
```

In this case, the PE-1 router selects the longer path to AS6400 via R1 (next-hop 11.0.0.1) as the best-path (Picture 8).

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 * i 190.0.0.0        2.2.2.2                  0    100      0 6400 i
 *>                   11.0.0.1                               0 6400 6400 6400 i
PE1#
```

Picture 8 - *PE-1 BGP Table When AS_PATH Length Criteria is Ignored*

3. Incoming Traffic Manipulation Using BGP Multi-Exit-Discriminator

The BGP Multi-Exit-Discriminator (MED) is an optional non-transitive attribute of BGP associated with a prefix. It means that the receiving AS cannot propagate MED across its AS border.

If the MED attribute is received over eBGP, it may be propagated over iBGP to the other BGP speakers within the same AS. However, MED attribute received from a neighboring AS must not be propagated to BGP speakers in other neighboring ASes. For instance, if AS 6400 adds the MED attribute to the prefix 190.0.0.0/16 and then sends an update with that prefix to AS 6500 and AS 6500 sends it to AS 6600, AS 6500 will see the MED value that the AS 6400 inserted, but AS 6600 will not.

When there are multiple entry points (connections) to the other AS, MED tells this other AS how we would like it to route traffic to our AS. The weight, local preference, originate route, and AS path are taken into account before the MED attribute gets considered. Unlike weight and local preference, lower MEDs are preferred over

the higher MEDs. Therefore, the exit or entry point with the lower metric gets favored.

Before the MED configuration, we need to configure the original route-policy PASS for the neighbor 11.0.0.2 (PE-1) and to apply it to the outbound routes.

```
router bgp 6400
 neighbor 11.0.0.2
  remote-as 6500
  address-family ipv4 unicast
   route-policy PASS in
   route-policy PASS out
```

When the MED attribute is missing (Picture 9), the Cisco IOS assigns the value of 0 to the route. This behavior is in compliance with RFC 4721 which says that routes that do not have the MULTI_EXIT_DISC attribute are considered to have the lowest possible MED value.

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 * i 190.0.0.0        2.2.2.2                  0    100      0 6400 i
 *>                   11.0.0.1                              0 6400 i
PE1#
```

Picture 9 - *PE-1 BGP Table Before Configuring MED on R1*

We can change this behavior using the command *bgp bestpath med missing-as-worst* in which case the missing MED attribute will be internally assigned the value of 4294967294. Let's configure the PE-1 router with this command and inspect the BGP table of PE-1 afterwards.

```
router bgp 6500
 bgp bestpath med missing-as-worst
```

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 190.0.0.0        2.2.2.2                  0    100      0 6400 i
 *                    11.0.0.1        4294967295             0 6400 i
PE1#
```

Picture 10 - PE-1 BGP Table After Changing the Default Action for Missing MED on PE-1

The missing MED value for the 190.0.0./16 prefix received from 11.0.0.1 (R1) is now changed to the value of 4294967294 by PE-1. Therefore, PE-1 prefers the path via the next-hop 2.2.2.2 (PE-2) with the lower metric value 0.

Let's reconfigure PE-1 to treat the missing MULTI_EXIT_DISC attribute as default.

```
router bgp 6500
 no bgp bestpath med missing-as-worst
```

3.1. BGP Multi-Exit-Discriminator Configuration on R1

Our goal is to configure the customer's router R1 to associate the MED value of 70 with the advertised prefix 190.0.0.0/16. If we do so, the incoming traffic sent from the ISP's PE-1 router to the customer will be routed through PE-2 and will finally enter the customer's network through the router R2 instead of R1.

We will use the same prefix-set CUST-PS that we created during the AS_PATH prepeding configuration. The prefix-set is matching the prefix 190.0.0.0/16.

```
prefix-set CUST-PS
  190.0.0.0/16
end-set
```

We will create a new route-policy MED-RPL matching a prefix-set CUST-PS. If the prefix-set is matched, the MED is set to 70 for the prefix. In case there were other prefixes advertised by R1, the MED-RPL would not modify their attributes.

```
route-policy MED-RPL
  if destination in CUST-PS then
    set med 70
  else
    pass
  endif
end-policy
```

Apply RPL MED-RPL to the outbound routes for eBGP peer PE-1 (11.0.0.2) .

```
router bgp 6400
 address-family ipv4 unicast

 neighbor 1.1.1.3
  remote-as 6400
  update-source Loopback0
  address-family ipv4 unicast
   next-hop-self

 neighbor 11.0.0.2
  remote-as 6500
  address-family ipv4 unicast
   route-policy PASS in
   route-policy MED-RPL out
```

Let's inspect the BGP table of PE-1 (Picture 11). The PE-1 router prefers the path with a lower MED value of 0 through PE-2 (the next-hop 2.2.2.2) to the path with the MED value of 70 through R1 (11.0.0.1).

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 190.0.0.0        2.2.2.2                  0    100      0 6400 i
 *                    11.0.0.1                70             0 6400 i
PE1#
```

Picture 11 - PE-1 BGP Table After Configuring MED on R1

4. Incoming Traffic Manipulation Using a Combination of BGP Communities with Local Preference

Another method that a customer may use to control incoming traffic from an ISP is to tag their routes with BGP communities. This method, however requires additional configuration on the provider's side. The ISP needs to configure its PE routers to match different BGP communities that are tagging the customer's routes and set a different local preference (LOCAL_PREF) values to the routes based on the matched communities.

NOTE: BGP communities are an optional transitive BGP attribute that can traverse from AS to AS.

As we know, BGP prefers a route with the higher local preference to the route with a lower local preference. Let's say that ISP (AS6500) configures a local preference 200 on their PE-1 router for the customer's prefix 190.0.0.0/16 received from the eBGP peer R1 and tagged with a community 200. Similarly, ISP sets a local preference 300 on PE-2 router for the same prefix 190.0.0.0/16 received from its eBGP peer R2 and tagged with a community 300. The router PE-1 will select the best-path to the 190.0.0.0/16 prefix through the iBGP neigbor PE-2. It is because BGP prefers a route with the higher local preference of value 300 to the route with a LOCAL_PREF of value 200.

An important thing to understand about the local preference is that it is local in the sense that the attribute is only propagated over iBGP sessions (within our AS) and not over the eBGP sessions (to external ASes). So when an ISP router PE-1 in AS6500 learns a prefix 190.0.0.0/16 from a neighbor R1 in AS6400, the update will not contain the LOCAL_PREF attribute. But when the router PE-2 propagates the prefix 190.0.0.0/16 towards the iBGP neighbor PE-1 within the local AS6500, the update contains the LOCAL_PREF attribute of value 300 associated with the prefix.

NOTE: Local preference default value is 100 and the route without a local preference has a default value.

4.1. BGP Community Configuration on R1

Let's configure the customer's router R1 (AS6400) to send a prefix 190.0.0.0/16 to the ISP router PE-1 with an attached community of 6400:200.

Use the prefix-set CUST-PS that we created during the AS_PATH pre-peding configuration. The prefix-set matches the prefix 190.0.0.0/16.

```
prefix-set CUST-PS
  190.0.0.0/16
end-set
```

Create a community-set ISP-CS with the community value of 6400:200. The first 16-bits represent the AS of the community origination (AS6400), and the second 16-bits represent a pattern defined by the AS6400.

```
community-set ISP-CS
  6400:200
end-set
```

The route-policy COMM-RPL is matching the customer prefix CUST-PS and sets the community ISP-CS. The route-policy passes prefixes that are not matched without changing their attributes.

```
route-policy COMM-RPL
  if destination in CUST-PS then
    set community ISP-CS
  else
    pass
  endif
end-policy
```

Apply the route-policy COMM-RPL to the eBGP neighbor 11.0.0.2 (PE-1) for the outbound route. In contrast with Cisco IOS. In IOS-XR, both standard and extended communities are sent by default on iBGP sessions but not on eBGP sessions. Therefore, we need to configure the command *send-community-ebgp* for the eBGP peer PE-1 as well.

```
router bgp 6400
address-family ipv4 unicast

neighbor 1.1.1.3
 remote-as 6400
 update-source Loopback0
 address-family ipv4 unicast
  next-hop-self

neighbor 11.0.0.2
 remote-as 6500
 address-family ipv4 unicast
  send-community-ebgp
  route-policy PASS in
  route-policy COMM-RPL out
```

4.2. BGP Community Configuration on R2

The configuration of BGP communities on the customer router R2 is similar to the configuration of R1. We need to create the prefix-list, the BGP community-set and the route-map matching the prefix-list and setting the community-set. Afterwards, we can configure the route-policy for the PE-1 and apply it to outbound routes. We also need to enable communities sending to eBGP peer PE-1.

```
prefix-set CUST-PS
  190.0.0.0/16
end-set

community-set ISP-CS
  6400:300
end-set

route-policy COMM-RPL
  if destination in CUST-PS then
    set community ISP-CS
  else
    pass
  endif
end-policy

router bgp 6400
 address-family ipv4 unicast

neighbor 1.1.1.3
 remote-as 6400
 update-source Loopback0
 address-family ipv4 unicast
  next-hop-self

neighbor 11.0.0.6
 remote-as 6500
 address-family ipv4 unicast
  send-community-ebgp
  route-policy PASS in
  route-policy COMM-RPL out
```

4.3. Checking BGP Tables of ISP Routers

Before we continue with the configuration of ISP's routers, we need to check that the received prefix is tagged with BGP communities 6400:200 and 6400:300. Therefore, we will examine the BGP table of both PE routers.

PE-1 has installed the prefix 190.0.0.0/16 received from the iBGP peer PE-2 (2.2.2.2) into the BGP table (Picture 12). The prefix has attached the community 6400:300.

PE-1 has installed the prefix 190.0.0.0/16 received from the eBGP peer R1 (11.0.0.1) into the BGP table. The prefix has attached the community 6400:200.

NOTE: As both routes have the same default LOCAL_PREF value of 100, PE1 prefers eBGP route received from R1 to iBGP route received from PE-2 (Picture 12). The route 190.0.0.0/16 through the next-hop 11.0.0.1 (R1) is installed in the routing table of PE-1.

```
PE1#
PE1#show bgp 190.0.0.0
BGP routing table entry for 190.0.0.0/16, version 3
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     5
  Refresh Epoch 1
  6400
    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal
      Community: 6400:300
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  6400
    11.0.0.1 from 11.0.0.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, external, best
      Community: 6400:200
      rx pathid: 0, tx pathid: 0x0
PE1#
```

Picture 12 - *PE-1 BGP Table after Configuration of BGP Communities on R1 and R2*

The router PE-2 has installed the received route 190.0.0.0/16 from the eBGP peer R2 into its BGP table. The route is tagged with the community 6400:300 and it has a LOCAL_PREF value of 100. There is also another route installed in the BGP table of PE-2 that is received from the iBGP peer PE-1. This route is tagged with the community 6400:200 and has a default LOCAL_PREF value of 100.

```
PE2#
PE2#show bgp 190.0.0.0
BGP routing table entry for 190.0.0.0/16, version 3
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     4
  Refresh Epoch 1
  6400
    2.2.2.1 (metric 2) from 2.2.2.1 (2.2.2.1)
      Origin IGP, metric 0, localpref 100, valid, internal
      Community: 6400:200
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  6400
    11.0.0.5 from 11.0.0.5 (1.1.1.2)
      Origin IGP, localpref 100, valid, external, best
      Community: 6400:300
      rx pathid: 0, tx pathid: 0x0
PE2#
```

Picture 13 - *PE-2 BGP Table After Configuration of BGP Communities on R2 and R1*

BGP process running on the PE-2 router selects the path to the 190.0.0.0/16 prefix via the eBGP peer R2 as the best path because eBGP routes are preferred to iBGP routes (Picture 13).

4.4. Setting the Local Preference Configuration on PE-1

We will configure PE-1 to change a default local preference of 100 to the value of 200 for the route 190.0.0.0/16 when this route is re-

ceived from eBGP peer R1 tagged with the community 6400:200.

First, we will configure PE-1 to display the BGP community in a new format AA:NN.

```
ip bgp-community new-format
```

Secondly, we will create the prefix-list CUST-PL matching the prefix 190.0.0.0/16.

```
ip prefix-list CUST-PL seq 5 permit 190.0.0.0/16
```

We need to create two standard community-lists as well.

```
ip community-list standard CUST-CL-200 permit 6400:200
ip community-list standard CUST-CL-300 permit 6400:300
```

The route-map CUST-PREF-RM matches the prefix CUST-PL and the appropriate community-lists. The local preference values are set based on the matched prefix-list and the community-list. If those attributes are not matched, the statement 30 permits routes without changing their attributes.

```
route-map CUST-PREF-RM permit 10
 match ip address prefix-list CUST-PL
 match community CUST-CL-200
 set local-preference 200

route-map CUST-PREF-RM permit 20
 match ip address prefix-list CUST-PL
 match community CUST-CL-300
 set local-preference 300

route-map CUST-PREF-RM permit 30
```

The last step consists of applying the route-map CUST-PREF-RM inbound to eBGP peer R1 and iBGP peer PE-2.

```
router bgp 6500
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 6500
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 11.0.0.1 remote-as 6400

 address-family ipv4
  network 0.0.0.0
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community
  neighbor 2.2.2.2 next-hop-self
  neighbor 2.2.2.2 route-map CUST-PREF-RM in
  neighbor 11.0.0.1 activate
  neighbor 11.0.0.1 route-map CUST-PREF-RM in
  neighbor 11.0.0.1 route-map DEFAULT-RM out
 exit-address-family
```

NOTE: We have created the route-map DEFAULT-RM during an initial configuration in order to advertise only a default route to the customer in AS6400.

4.5. Setting Local Preference Configuration on PE-2

We will configure PE-2 to change a default local preference 100 to the value of 300 for the route 190.0.0.0/16 when this route is received from eBGP peer R2 and tagged with the community 6400:300. The PE-2 router will advertise this route to its iBGP peer PE-1 tagged with the community 6400:300.

First, we will configure PE-2 to display the BGP community in a new format AA:NN.

```
ip bgp-community new-format
```

Secondly, we will create the prefix-list CUST-PL matching the prefix 190.0.0.0/16.

```
ip prefix-list CUST-PL seq 5 permit 190.0.0.0/16
```

We also need to create two standard community-lists.

```
ip community-list standard CUST-CL-200 permit 6400:200
ip community-list standard CUST-CL-300 permit 6400:300
```

The route-map CUST-PREF-RM is matching the prefix CUST-PL and the appropriate community-lists. The local preference values is set based on the matched prefix-list and the community-list. If those attributes are not matched, the statement 30 permits other routes without changing their attributes.

```
route-map CUST-PREF-RM permit 10
 match ip address prefix-list CUST-PL
 match community CUST-CL-200
 set local-preference 200

route-map CUST-PREF-RM permit 20
 match ip address prefix-list CUST-PL
 match community CUST-CL-300
 set local-preference 300

route-map CUST-PREF-RM permit 30
```

As the the last step, we will apply the route-map CUST-PREF-RM to eBGP peer R2 and to the iBGP peer PE-1 inbound.

```
router bgp 6500
 bgp log-neighbor-changes
 neighbor 2.2.2.1 remote-as 6500
```

```
neighbor 2.2.2.1 update-source Loopback0
neighbor 11.0.0.5 remote-as 6400

address-family ipv4
 network 0.0.0.0
 neighbor 2.2.2.1 activate
 neighbor 2.2.2.1 send-community
 neighbor 2.2.2.1 next-hop-self
 neighbor 2.2.2.1 route-map CUST-PREF-RM in
 neighbor 11.0.0.5 activate
 neighbor 11.0.0.5 route-map CUST-PREF-RM in
 neighbor 11.0.0.5 route-map DEFAULT-RM out
exit-address-family
```

4.6. Checking BGP Table of PE-1

If there were no configuration mistakes made, the ISP router PE-1 should prefer the path to the prefix 190.0.0.0/16 via the next-hop 2.2.2.2 (PE-2). The LOCAL_PREF value of this route is set to 300 (Picture 14).

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          0.0.0.0                  0         32768 i
 * i                  2.2.2.2                  0    100      0 i
 *   190.0.0.0        11.0.0.1                      200      0 6400 i
 *>i                 2.2.2.2                  0    300      0 6400 i
PE1#
```

Picture 14 - *PE-1 BGP Table after Configuring BGP Communities on Customer's Routers and Local Preference on Provider's Routers*

5. Incoming Traffic Manipulation Using BGP Conditional Route Injection

One other method that a customer can use to control the ingress traffic is based on the longest prefix-matching behavior. If the customer configures the router R2 to advertise more specific routes 190.0.0.0/17 and 190.0.128.0/17 along with a less specific route 190.0.0.0/16 then the incoming traffic takes a path through the router R2 to the customer prefix. The less specific (component) routes received from R2 are more specific than the aggregate route received from R1, so they take precedence on ISP routers. This scenario can be accomplished by configuring BGP conditional route injection on the customer router R2.

R3 advertises the aggregate route 190.0.0.0/16 to both iBGP peers R1 and R2. The R1 router receives prefix 190.0.0.0/16 from R3 and advertises it to its eBGP neighbor PE-1. R2 receives the 190.0.0.0/16 prefix from R3 as well and installs it into its routing table. Based on the presence of the aggregated route 190.0.0.0/16 in the BGP table, R2 injects two component routes 190.0.0.0/17 and 190.0.128.0/17 into its BGP table. R2 advertises component routes along with the aggregate route 190.0.0.0/16 to the eBGP peer PE-2. If for some reason, the aggregate route is not received from R3, the R2 router is not injecting any component routes. Neither the aggregate route nor component routes are advertised to PE-2. In other words, the aggregate route must be present in the BGP table in order for the component route to be injected into the local BGP table.

NOTE: At the time of writing this tutorial, Cisco IOS-XR has not supported BGP conditional route injection. The Cisco IOS however, supports this feature. Therefore, we will replace the router R2 (IOS-XR 6.1.3) with a router R2-IOS, running Cisco IOS 15.6(2)T. BGP conditional route injection will be supported in the next IOS-XR release.

The command for BGP conditional route injection consists of two route-maps.

```
router bgp 64500
 bgp inject-map INJECT-MAP exist-map AGGREGATE-EXIST
```

The condition is represented by the exist-map. The exist-map contains two prefix-lists that should be matched in order to inject a component route. Those are AGGREGATE-ROUTE and the route-source. The route-source consists of the prefix-list NEIGHBOR that is matching the IP address of a BGP peer advertising the aggregated route. It is the IP address of a BGP neighbor with the prefix length /32.

```
route-map AGGREGATE-EXIST permit 10
 match ip address prefix-list AGGREGATE-ROUTE
 match ip route-source prefix-list NEIGHBOR
```

The inject-map contains a sequence that is matching a single or more injected routes.

```
route-map INJECT permit 10
 set ip address prefix-list INJECTED-ROUTE
```

Below is the initial configuration of the new added router R2-IOS.

5.1. R2 Initial Configuration

```
interface Loopback0
 ip address 1.1.1.2 255.255.255.255

interface GigabitEthernet0/0
 ip address 11.0.0.5 255.255.255.252

interface GigabitEthernet0/1
 ip address 10.0.0.5 255.255.255.252

router ospf 1
 network 1.1.1.2 0.0.0.0 area 0
 network 10.0.0.4 0.0.0.3 area 0

router bgp 6400
 bgp log-neighbor-changes
 neighbor 1.1.1.3 remote-as 6400
 neighbor 1.1.1.3 update-source Loopback0
 neighbor 11.0.0.6 remote-as 6500
```

```
 address-family ipv4
  neighbor 1.1.1.3 activate
  neighbor 1.1.1.3 next-hop-self
  neighbor 11.0.0.6 activate
```

5.2. BGP Conditional Route Injection Configuration on R2

5.2.1. Exist-map Configuration

```
route-map R3-AGGREGATE-EXIST-RM permit 10
 match ip address prefix-list AGG-ROUTE-190.0.0.0/16-
PL
 match ip route-source prefix-list R3_IP-PL

ip prefix-list AGG-ROUTE-190.0.0.0/16-PL seq 10 per-
mit 190.0.0.0/16
ip prefix-list R3-IP-PL seq 10 permit 1.1.1.3/32
```

5.2.2. Inject-map Configuration

```
route-map INJECT-PE2-RM permit 10
 set ip address prefix-list INJECTED-CUST-PL

ip  prefix-list  INJECTED-CUST-PL  seq  10  permit
190.0.0.0/17
ip  prefix-list  INJECTED-CUST-PL  seq  20  permit
190.0.128.0/17
```

5.2.3 Assigning Inject-map and Exist-map to BGP

```
router bgp 6400
 bgp log-neighbor-changes
 neighbor 1.1.1.3 remote-as 6400
 neighbor 1.1.1.3 update-source Loopback0
 neighbor 11.0.0.6 remote-as 6500

 address-family ipv4
  bgp inject-map INJECT-PE2-RM exist-map R3-AGGRE-
GATE-EXIST-RM
  neighbor 1.1.1.3 activate
  neighbor 1.1.1.3 next-hop-self
  neighbor 11.0.0.6 activate
```

5.3. Verification of BGP Conditional Routes Injection

First, we will examine a routing table of R2 for BGP routes (Picture 15). There is a default route received from the eBGP peer PE-2 (11.0.0.6) and the aggregate route 190.0.0.0/16 received from iBGP peer R3 (1.1.1.3). The routing table also contains two injected routes 190.0.0.0/17 and 190.0.128.0/17, generated by R2.

```
R2-IOS#
R2-IOS#show ip route bgp | begin Gateway
Gateway of last resort is 11.0.0.6 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 11.0.0.6, 01:03:50
      190.0.0.0/16 is variably subnetted, 3 subnets, 2 masks
B        190.0.0.0/16 [200/0] via 1.1.1.3, 01:03:03
B        190.0.0.0/17 [200/0] via 1.1.1.3, 01:03:03
B        190.0.128.0/17 [200/0] via 1.1.1.3, 01:03:03
R2-IOS#
```

Picture 15 - *R2 Routing Table*

Injected routes can be checked with the command show bgp injected-paths on a router where the injection is done (Picture 16).

```
R2-IOS#
R2-IOS#show bgp injected-paths | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 190.0.0.0/17     1.1.1.3                             0 ?
 *>i 190.0.128.0/17   1.1.1.3                             0 ?
R2-IOS#
```

Picture 16 - *Checking Injected Routes on R2*

The BGP table of PE-2 is depicted on the Picture 17. The router PE-2 has installed the aggregate route 190.0.0.0/16 along with two component routes 190.0.0.0/17 and 190.0.128.0/17 received from eBGP peer R2 (11.0.0.5). The path to the aggregate route via eBGP peer R2 is selected as the best-path. There is also another aggregate route installed in the BGP table of PE-2 that includes the path via the iBGP peer PE-1 (2.2.2.1)

```
PE2#
PE2#show ip bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.1                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  190.0.0.0/17     11.0.0.5                               0 6400 ?
 * i 190.0.0.0        2.2.2.1                  0    100      0 6400 i
 *>                   11.0.0.5                               0 6400 i
 *>  190.0.128.0/17   11.0.0.5                               0 6400 ?
PE2#
```

Picture 17 - *PE-2 BGP Table*

The router PE-2 advertises the injected routes along with the aggregate route to its iBGP neighbor PE-1. As PE-1 receives the component routes 190.0.0.0/17 and 190.0.128.0/17 from PE-2 (2.2.2.2) only, these are selected as the best-path routes by BGP running on PE-1 (Picture 18). The aggregate route 190.0.0.0/16 is however, received from both eBGP R1 (11.0.0.1) and iBGP PE-2 (2.2.2.2) neighbors. As eBGP routes are preferred to iBGP routes by BGP best path selection algorithm, the path via R1 is selected as the best path.

```
PE1#
PE1#show bgp | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          2.2.2.2                  0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 190.0.0.0/17     2.2.2.2                  0    100      0 6400 ?
 *>  190.0.0.0        11.0.0.1                               0 6400 i
 * i                  2.2.2.2                  0    100      0 6400 i
 *>i 190.0.128.0/17   2.2.2.2                  0    100      0 6400 ?
PE1#
```

Picture 18 - *PE-1 BGP Table*

6. Incoming Traffic Manipulation Using Noction IRP

So far, we've discussed several methods for Inbound Traffic Manipulation. Even though these methods are proven to work, they have some significant drawbacks. All of them are manual methods thus error-prone, time consuming and not scalable, especially when customers have multiple upstream connections.

BGP Inbound Traffic Engineering

Network administrators frequently receive alerts during peak network hours and have to manually add or remove prepends or altogether oust some inbound prefixes from being announced to the affected neighboring links. These changes are many times forgotten or other times just push the bulk of traffic towards other links and the problem re-appears. Some links are becoming overwhelmed again while other links remain barely used.

Noction **Intelligent Routing Platform** replaces the manual bandwidth checks and cumbersome direct AS prepends practices with the automated inbound commit control capability, which uses providers known prepend communities, integrated in IRP routing policies to intelligently route traffic. It is important to mention that besides optimizing inbound traffic, IRP is capable of influencing Transit Traffic as well.

Numerous Intelligent Routing Platform deployments prove the success of the solution. Read the **ATMC Case Study** to learn how companies gain optimal control over the Inbound Traffic, keeping bandwidth levels under the preconfigured 95th percentile value for all the providers, preventing link saturation and attaining significant cost savings. Faster responses to network topology changes and fewer sub-optimal routing bring immediate, measurable network performance results that the end users notice and appreciate.

NOCTION
NETWORK INTELLIGENCE

This ebook was brought to you by Noction.

Noction Intelligent Routing Platform enables enterprises and service providers to maximize end-to-end network performance and safely reduce infrastructure costs. The platform evaluates critical network performance metrics in real-time and responds quickly by automatically rerouting traffic through a better path to avoid outages and congestion.

Request a free trial today and see how IRP can boost your network performance.

Start a Free Trial