

The BGP Multi Exit Discriminator

MED, Internet Exchanges and Route Servers

The BGP Multi Exit Discriminator, Internet Exchanges and Route Servers

In the BGP path selection algorithm, the MED comes into play when there are multiple paths to a destination prefix that have the same local preference and the same AS path length. And, unless we configure ***always-compare-med***, MEDs for different paths are only compared for paths towards the same neighboring AS. In other words, the purpose of the MED is to select the best path when there are multiple connections between two autonomous systems. Today, we're going to focus on optimizing traffic flow between two networks that interconnect using multiple routers connected to one internet exchange, but the same applies to interconnecting using multiple internet exchanges in the same region.

When you connect two routers to the same internet exchange—or, more generally, use two routers in the same location to connect to external networks—it's best practice to set up BGP sessions to/from each router. So if networks A and B interconnect over an exchange and both networks have two routers connected to the IX, this will result in four BGP sessions as shown in **figure 1**.

If you then let the BGP path selection algorithm choose the best path without adjusting any of the BGP attributes, all the attributes will be the same.

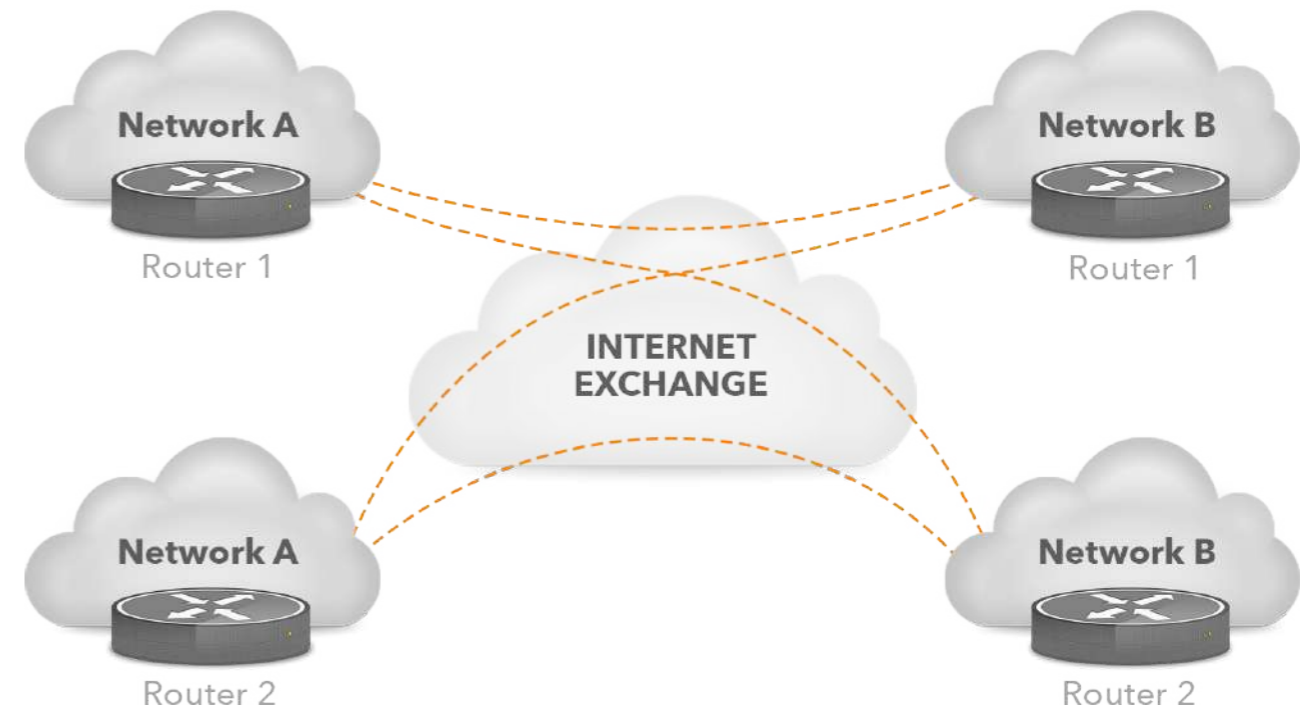


Figure 1. BGP sessions between two networks with two routers connected through an internet exchange

So for incoming traffic, the path selection algorithm will end up relying on the penultimate **tiebreaker** rule: select the route(s) advertised by the BGP speaker with the lowest BGP identifier. For outgoing traffic, the decision will come down to the last tiebreaker: select the route received from the BGP speaker with the lowest BGP identifier.

The BGP Multi Exit Discriminator

On Cisco routers, the BGP identifier defaults to the highest IPv4 address configured on a loopback interface, or the highest IPv4 address configured on a physical interface if no loopback interfaces are configured.



WARNING: Should the router run IPv6 only, you'll need to configure a BGP identifier explicitly using the **bgp router-id** command.

So without further action, most of both incoming and outgoing traffic will flow through the same router. This may actually be desirable when capacity isn't an issue, as this makes the traffic flow predictable and debugging problems easier. Having all traffic flow through one router can also be useful if that router is faster, or has a higher capacity link to the IX or to the internal network. However, in that case you'll probably want to explicitly choose which router handles the traffic rather than depend on the BGP tie breaking rules.

For incoming traffic, you can do this by advertising your prefixes with different MEDs through both routers. For instance, this configuration sets the MED to 10:

```
!  
router bgp 65549  
  neighbor 10.0.0.1 remote-as 65550  
  neighbor 10.0.0.1 route-map setmed10 out  
  neighbor 10.0.0.2 remote-as 65550  
  neighbor 10.0.0.2 route-map setmed10 out  
!  
route-map setmed10 permit 10  
  set metric 10  
!
```

With this on the primary router and an equivalent configuration that sets the MED to 20 on the backup/secondary router, all incoming traffic will flow towards the primary router. At least, if neighboring networks honor your MEDs. It's generally considered polite to accept the preference of neighboring networks unless you have a good reason to overrule this preference.



NOTE: Some networks simply overwrite the MED values they receive, so sending MEDs to influence traffic flow won't work towards those networks.



The BGP Multi Exit Discriminator

Also, remember that the MED only survives one inter-AS hop. So sending different MEDs towards a neighboring AS will influence how you receive traffic from that AS, but networks that are two or more AS hops away won't see the MEDs; if you want to influence routing decisions in those networks, as well as in networks that ignore advertised MEDs, you'll have to perform AS path prepending or announce more specifics.

If we want to influence outgoing traffic, we'll have to adjust **incoming** MEDs. This could be done by applying the same **setmed10** route map from the example above as **in** on a BGP session. However, that makes you one of these impolite networks that doesn't honor their neighbor's MEDs. So usually, it's better to **adjust** the MEDs your neighbor sends you rather than simply overwrite them:

```
!
router bgp 65549
  neighbor 10.0.0.1 remote-as 65550
  neighbor 10.0.0.1 route-map addmed10 in
  neighbor 10.0.0.2 remote-as 65550
  neighbor 10.0.0.2 route-map addmed10 in
!
route-map addmed10 permit 10
  set metric +10
!
```

With this configuration in effect and **set metric +20** on the backup router, if your neighbor sends you a MED of 0 or no MED, the MED will be 10 for prefixes received by the primary router and 20 for

those prefixes on the backup router. So other routers in the network will send traffic to the destinations in question through the primary router. Even if the traffic ends up at the backup router, the backup router will send it to the primary router rather than deliver it to the neighboring AS itself, as the MED step in the BGP path selection algorithm comes in just before the "prefer eBGP over iBGP" step.

If the neighboring network has a similar configuration, they will be sending an MED of 10 from their primary router and 20 from their secondary router. This will result in the following MED values:

Local router	Local MED	Remote router	Remote MED	Resulting MED
R1	+10	R1	10	20
R1	+10	R2	20	30
R2	+20	R1	10	30
R2	+20	R2	20	40

If you don't really care that much which router handles the traffic, you may choose to optimize traffic flow slightly by prioritizing shorter, more local paths. For instance, suppose the IX is present in locations A and B, and you're connected in location A. Some of your peers will have routers in both locations A and B.

The BGP Multi Exit Discriminator

All else being equal, it would be somewhat more preferred to exchange traffic with a neighboring AS through their router in location A. The path will be somewhat shorter, there is less layer 2 equipment in the middle that could fail, and if your router and the remote router are connected to the same Ethernet switch, the only bottleneck between the routers (other than their IX connection) will be the backplane of the IX switch. Switch backplanes are rarely a bottleneck, but the backhaul link between the IX switches in two locations could be. So in this case, you'll have to determine the location for each neighboring AS' router and apply different MEDs based on that like in this configuration:

```
!  
router bgp 65549  
  neighbor ix-location-a peer-group  
  neighbor ix-location-a route-map addmed10 in  
  neighbor ix-location-a route-map setmed10 out  
  neighbor ix-location-b peer-group  
  neighbor ix-location-b route-map addmed20 in  
  neighbor ix-location-b route-map setmed20 out  
  neighbor 10.0.0.1 remote-as 65550  
  neighbor 10.0.0.1 peer-group ix-location-a  
  neighbor 10.0.0.2 remote-as 65550  
  neighbor 10.0.0.2 peer-group ix-location-b  
!
```

Note that this configuration is equally useful if you only have a single router connected to the internet exchange.

If you have two routers and optimizing traffic flow to keep it local isn't relevant, you can use a similar configuration to better balance traffic between the two routers as follows. On one router, we apply the same MED to all neighbors to have a consistent baseline:

```
!  
route-map addmed15 permit 10  
  set metric +15  
!  
route-map setmed15 permit 10  
  set metric 15  
!
```

Then, on the other router, distribute the neighbors over peer groups med10 and med20 until traffic is reasonably balanced:

```
!  
router bgp 65549  
  neighbor med10 peer-group  
  neighbor med10 route-map addmed10 in  
  neighbor med10 route-map setmed10 out  
  neighbor med20 peer-group  
  neighbor med20 route-map addmed20 in  
  neighbor med20 route-map setmed20 out  
  neighbor 10.0.0.1 remote-as 65550  
  neighbor 10.0.0.1 peer-group med10  
  neighbor 10.0.0.2 remote-as 65550  
  neighbor 10.0.0.2 peer-group med20  
!
```



Route Servers and Propagating the MED

Most internet exchanges have route servers to facilitate easy peering between IX members. Without route servers, each IX member has to set up BGP sessions towards every other member they want to peer with. On a large exchange with as many as a hundred or more members, this is a lot of work and prospective peering partners don't always reply quickly when contacted. By connecting to a route server, you immediately have peering with all the other IX members connected to the route server.



NOTE: Only networks with an open peering policy connect to route servers, so a route server typically only gives you access to smaller to medium sized peers.

Route servers are almost always deployed in pairs; each router maintains a BGP session with each of the two route servers to avoid having a single point of failure. It's standard behavior for BGP routers to detect that two neighbors are connected to a common subnet and then not adjust the next hop address in BGP updates. This way, even though the route server is in the path from the perspective of the flow of BGP updates, the route server is **not** on the data path: packets are directly sent between peers as if the route server isn't present. See **figure 2**.

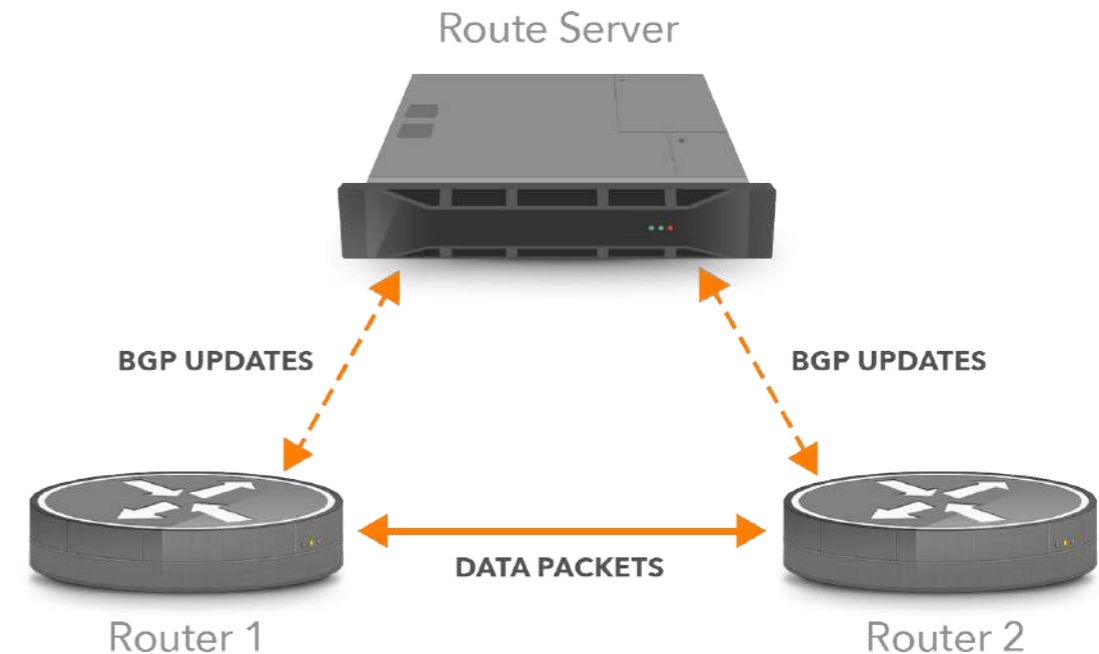


Figure 2. BGP updates flow through the route server but data packets flow directly

However, when a regular BGP router is used as a route server, it will update the AS path as per the eBGP rules. So the AS number of the route server will be added to the AS path, making the AS path a hop longer than it would be with direct peering without a route server in the middle. Additionally, the eBGP rules mandate that the Multi Exit Discriminator is only propagated over a single eBGP hop. So if in the figure, router 1 sends a prefix with an MED of 10 to the route server, the route server will see that MED, but the route server will then remove the MED as it propagates the prefix to router 2. With a direct BGP session between routers 1 and 2, router 2 would have seen the MED value 10 from router 1.

The BGP Multi Exit Discriminator

Today, it's common for route servers to **not** apply the eBGP rules for updating the BGP attributes. Recent versions of Cisco IOS make it possible to enable this behavior as follows:

```
!  
router bgp 65549  
  neighbor 10.0.0.1 remote-as 65550  
  address-family ipv4 unicast  
  neighbor 10.0.0.1 activate  
  neighbor 10.0.0.1 route-server-client  
!
```

As you can see in the example, the **route-server-client** setting is specific to an address family. (If you're only using IPv4 you can leave out the **address-family** and **activate lines**.) So it's possible to have a neighbor be a route server client for IPv4 and not IPv6, or the other way around.

With this setting in effect, the router will **not** update the next hop, AS path or MED attributes like it normally would for updates sent to eBGP neighbors, but rather, pass on these attributes unmodified. So for all intents and purposes, the presence of the route server is invisible.

There is a caveat, however. RFC 4271 states:

"If the UPDATE message is received from an external peer, the local system MAY check whether the leftmost (with respect to the position of octets in the protocol message) AS in the AS_PATH

attribute is equal to the autonomous system number of the peer that sent the message. If the check determines this is not the case, the Error Subcode MUST be set to Malformed AS_PATH."



WARNING: As a result, it's not uncommon for BGP routers to react badly to a route server leaving out its own AS number from the AS path.

Today, Cisco routers accept this route server behavior unless you specify:

```
!  
router bgp 65549  
  bgp enforce-first-as  
!
```

And there may still be a few routers that **don't** accept the route server behavior, but can be told to do so:

```
!  
router bgp 65549  
  no bgp enforce-first-as  
!
```



Using the MED in a Route Server

With the subtleties of route server behavior out of the way, we can have a look at using the MED in route servers. Suppose an internet exchange is present in two locations, as per **figure 3**.

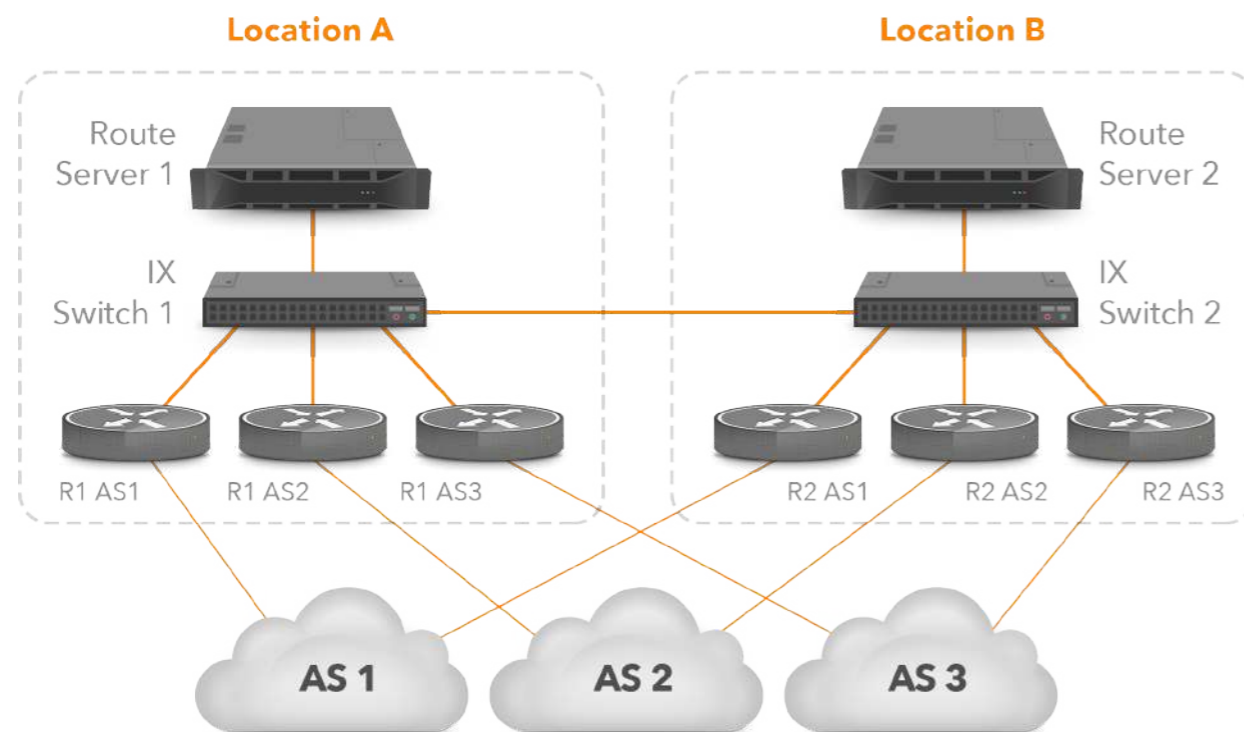


Figure 3. An internet exchange with two physical locations

Three networks each have a router at both of the IX locations, where they connect to the local IX switch. The switches in the two locations are connected through a high capacity link so that networks can deliver their traffic at either location. But as we saw earlier, if two networks are connected to both locations and don't express a preference for the path over one location, the BGP tiebreaking

rules will kick in, and there's a good chance that AS 1 will use its router 1 to send traffic to AS 2 router 2. So this traffic has to flow through the switch in location A, through the link between the two locations, and then through the switch in location B.

Keeping the traffic local to one location is potentially beneficial for the connected networks: the switch backplane has a higher bandwidth than the link between the two locations and there's less equipment in the middle that can fail. There's also a benefit to the IX: less utilization of the link between the two locations.

So the IX could apply a configuration to the route servers to prefer local paths over paths that go through both locations. A slight adjustment to the MED works well in this situation because it doesn't overrule the preferences expressed by the connected networks. The route servers would then apply the following configurations:

```
!  
! route server location A  
!  
router bgp 65551  
  neighbor loca peer-group  
  neighbor loca route-map addmed10 in  
  neighbor loca route-map addmed10 out  
  neighbor locb peer-group  
  neighbor locb route-map addmed40 in  
  neighbor locb route-map addmed40 out  
!
```

The BGP Multi Exit Discriminator

```
route-map addmed10 permit 10
  set metric +10
!
route-map addmed40 permit 10
  set metric +40
!

!
! route server location B
!
router bgp 65551
  neighbor loca peer-group
  neighbor loca route-map addmed40 in
  neighbor loca route-map addmed40 out
  neighbor locb peer-group
  neighbor locb route-map addmed20 in
  neighbor locb route-map addmed20 out
!
route-map addmed20 permit 10
  set metric +20
!
route-map addmed40 permit 10
  set metric +40
!
```

Assuming the member routers send prefixes with a MED of zero or no MED on their prefixes, this means that the route servers will have prefixes with the following MED values in its BGP table:

- Route server in location A from routers in location A: 10
- Route server in location A from routers in location B: 40
- Route server in location B from routers in location A: 40
- Route server in location B from routers in location B: 20

So the route server in location A will prefer paths learned from routers in location A and thus propagate those paths, while the route server in location B will prefer paths learned from routers in location B and propagate those paths.

What the member routers see depends on the **route-server-client** setting. Without that setting in effect, the route servers won't propagate the MED values in their own BGP tables, so the MEDs the member routers will see are zero plus the value added by the route map:

From router location	Through route server location	MED in route server BGP table	To router location	MED in receiving router BGP table
A	A	>10	A	> 10
			B	40
B	A	40 (not best, normally not propagated)	A	>10
			B	40
A	B	40 (not best, normally not propagated)	A	40
			B	>20
B	B	>20	A	40
			B	>20

Table 2. The route map (The > indicates the best path.)

The BGP Multi Exit Discriminator

But if the **route-server-client** setting is in effect, the MED in the route server’s BGP table **will** be propagated after the relevant MED adjustment route map is applied:


From router location	Through route server location	MED in route server BGP table	To router location	MED in receiving router BGP table
A	A	>10	A	>20
			B	50
B	A	40 (not best, normally not propagated)	A	>50
			B	80
A	B	40 (not best, normally not propagated)	A	80
			B	>60
B	B	>20	A	60
			B	>40

Table 3. Adjusted route map (The > indicates the best path.)

As you can see, because we carefully chose our MED values, the results are the same: the traffic will have the lowest MED from a router in location A to a router in location A (10 or 20, respectively), with the second lowest MED from a router in location B to a router in location B (20 or 40) and various higher values between routers in different locations.

So with this configuration, traffic is preferentially exchanged in location A. If one network isn’t available in location A, the traffic will be exchanged in location B. Traffic will only flow over the link between the two locations if the sending network and the receiving network aren’t available in the same location.

The situation where traffic is preferentially exchanged in one location makes sense when asymmetric routing (through location A in one direction and through location B in the return direction) is undesirable, for instance because there are firewalls on the path. This is more common in private networks. If this is not a consideration, there’s no reason to make location B less preferred than location A, so the route server in location B should use the setmed10 route map to/from neighbors in location B the same way the route server in location A does for neighbors in location A.



NOTE: In these examples we use the MED as intended, i.e., the MED is only considered for prefixes learned from the same neighboring AS. So the presence or absence of the **always-compare-med** doesn’t change the outcome.





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](#)