Deployment guidelines

Below follows a sample configuration for Cisco routers to announce a prefix to the route servers:

```
!
router bgp your-asn


 bgp always-compare-med
 no bgp enforce-first-as
 bgp log-neighbor-changes
 neighbor MN-IX-RS peer-group
 neighbor MN-IX-RS remote-as 209752
 neighbor MN-IX-RS version 4
 neighbor MN-IX-RS  transport connection-mode active
 neighbor 77.69.248.251 peer-group MN-IX-RS
 neighbor 77.69.248.251 description rs1.MN-IX.net
 neighbor 77.69.248.252 peer-group MN-IX-RS
 neighbor 77.69.248.252 description rs2.MN-IX.net
!
 address-family ipv4
neighbor MN-IX-RS activate
 neighbor MN-IX-RS next-hop-self
 neighbor MN-IX-RS soft-reconfiguration inbound
 neighbor MN-IX-RS route-map TO-RS out
 no auto-summary
 no synchronization
 neighbor 77.69.248.251 peer-group MN-IX-RS
 neighbor 77.69.248.252 peer-group MN-IX-RS
```

```
 network 192.168.110.0 mask 255.255.255.0
 network 192.168.111.0 mask 255.255.255.0
 network 192.168.112.0 mask 255.255.255.0
 exit-address-family
 !

ip as-path access-list 12 permit ^$
!
ip prefix-list TO-RS seq 10 permit 192.168.110.0/24

ip prefix-list TO-RS seq 20 permit 192.168.111.0/24

ip prefix-list TO-RS seq 30 permit 192.168.112.0/24

!
route-map TO-RS permit 10
 match ip address prefix-list TO-RS
!
```

Note that for recent IOS versions (e.g. 12.0(26)S and 12.2(25)S and up, where this has become the - hidden - default) you will have to specify "no bgp enforce-first-as (IOS, IOS-XE) / bgp enforce-first-as disable (IOS-XR)" as the route server does not insert its own ASN into the AS_path of relayed prefix announcements. Zebra and Quagga suffer from the same problem since somewhere in 0.91.

Below is a similar example for Juniper routers:

```
[edit]
user@junix# show protocols bgp
group IPV4-RS {
    type external;
```

```
    description "Route Servers";
    family inet {
        unicast;
    }
    export TO-RS;
    peer-as 209752;
    neighbor 77.69.248.251 {
        description rs1.MN-IX.net;
    }
    neighbor 77.69.248.252 {
        description rs2.MN-IX.net;
    }
}

[edit]
user@junix# show policy-options policy-statement TO-RS
term unicast-export {
    from {
        rib inet.0;
        prefix-list to-announce;
    }
    then accept;
}
term end {
    then reject;
}
```

```
[edit]

user@junix# show policy-options prefix-list to-announce

10.25.1.0/24;
```

Route Server Filtering

MN-IX route server filtering.

Incoming prefixes sanitisation

All MN-IX route servers in Manama perform basic and extended prefix filtering to all member/customer BGP sessions that are being established (optionally) with our Route Servers. The basic prefix filtering consists of blocking RFC 1918 ranges, bogon and Martian prefixes and the default route. We base our list on Team CYMRU's BOGON List.

Outgoing prefixes filtering among route-server members

The extended prefix filtering offers 3+1 peering modes and the customer is able to select the desired one through the MN-IX portal.

The MN-IX route servers implement outgoing filtering based on policies defined by the route server participants. This filtering is applied on outgoing advertisements. By defining your policy using an IRRDB object described by RPSL, you instruct the route servers to send your prefixes to other participants (export policy), or from which participants you wish to receive prefixes (import policy). Therefore, connecting with the route servers does not necessarily mean that you would be obliged to send/receive prefixes for all connected participants; filtering schemes are available.

The filters are solely derived from your IRRDB objects, which use RPSL as a description language. There are three different options you can use: ANY, ANY EXCEPT and RESTRICTIVE, to define your filtering needs.

In order to pick up the change in member's peering policy, MN-IX route-servers periodically detect policy changes every hour starting at midnight Bahrain time. If you wish to have your filters updated right away or encounter any problems, please contact the MN-IX NOC. We can apply a new configuration for the route-server to reflect your new policy.

Please check the list of these supported IRRDBs.

Would you like to have your filters updated right away or do you encounter any problems?

CONTACT US

The 3+1 peering modes of route servers

As stated above, from October 2019 onwards, all route servers in Bahrain implement 3+1 peering modes of prefix filtering in the outbound direction.

* Peering mode 'Filtering based on both IRRDB and RPKI data':

This is the default option when a new BGP session is established with the MN-IX route servers. By selecting this peering mode, the route servers are configured automatically to apply IIRDB based filtering (explanation is provided below) and RPKI based filtering (explanation provided below). In case you already have a session with the NL route servers and this option is not the selected one, we recommend you to switch your peering mode to the default one.

* Peering mode 'Filtering based on IRRDB data':

By selecting this option, Route Server outgoing prefixes extended filtering is based on IRRDB filtering only (explanation below). In summary, the prefixes that are being blocked are the ones that are not present in AS's announced AS/AS-SET. We strongly recommend to make sure that your IRRDB objects are correctly updated and described in the RIPE database when having this option enabled (and the default one)

* Peering mode 'Filtering based on RPKI data':

By selecting this option, Route Server outgoing prefixes extended filtering is based on RPKI filtering. In summary, the prefixes that are being blocked are the ones with ROA status 'INVALID'. We strongly recommend to make sure that your IRRDB ROAs are correctly updated in the RIPE database when having this option enabled (and the default one).

Optionally, we can offer the following peering mode in case you really need an unfiltered BGP feed (e.g. for research purposes":

* Peering mode 'Just tagging':

By selecting this not recommended option, no filtering is applied to announced prefixes. That functionality is helpful for research institutes who want to receive all information or organisations who want to apply their own BGP policies. However, any prefixes that are not filtered will be tagged by using standard BGP communities based on the following criteria (communities are given in the parentheses).

- Prefix with ROA status: VALID (209752:65012)

- Prefix with ROA status: INVALID (209752:65022)

- Prefix with ROA status: UNKNOWN (209752:65023)

- Prefix present in AS's announced AS/AS-SET (209752:65011)

- Prefix not present in AS's announced AS/AS-SET (209752:65021)

IRRDB based Filtering

Our route servers generate their configuration based on a IRRDB parser script. The script supports most of the IETF snijders-rpsl-via draft extensions to the RPSL and the 'import-via' and 'export-via' attributes defined therein. Using these attributes, we allow for ASN32 aut-num objects in expressions and promote more elegant policy definitions regarding route servers.

You can use the following examples to update your peering policy to support the 'import-via' and 'export-via' attributes and make sure that you are fully compatible with MN-IX route servers (we're using AS1200 as the example aut-num object).

1. ANY

(Send and receive prefixes to/from any RS participant):

[...]

```
import-via: AS209752 from AS-ANY accept ANY

export-via: AS209752 to AS-ANY announce AS1200
```

[...]

2. ANY EXCEPT

(Send and receive prefixes to/from any RS participant EXCEPT AS666):

[...]

```
import-via: AS209752 from AS-ANY EXCEPT AS666 accept ANY

export-via: AS209752 to AS-ANY EXCEPT AS666 announce AS1200
```

[...]

3. RESTRICTIVE

(Send and receive prefixes ONLY to/from AS15703):

[...]

```
import-via: AS209752 from AS15703 accept ANY

export-via: AS209752 to AS15703 announce AS1200
```

[...]

AS-SETs also work in all cases:

4. ANY EXCEPT using AS-SETs

(Send and receive prefixes to/from any RS participant EXCEPT ASes/AS-SETs included in AS1200:CUSTOMERS):

[...]

```
import-via: AS209752 from AS-ANY EXCEPT AS1200:AS-CUSTOMERS accept ANY

export-via: AS209752 to AS-ANY EXCEPT AS1200:CUSTOMERS announce
AS1200:CUSTOMERS
```

[...]

5. RESTRICTIVE using AS-SETs

(Send and receive prefixes ONLY to/from AS's/AS-SETs contained in AS-SET AS1200:CUSTOMERS):

[...]

```
import-via: AS209752 from AS1200:AS-PEERS accept ANY

export-via: AS209752 to AS1200:AS-PEERS announce AS1200:AS-CUSTOMERS
```

[...]

6. RESTRICTIVE with NOT ANY

```
# Import from no-one

import-via: AS209752 from AS-ANY accept NOT ANY

# Export to no-one

export-via: AS209752 to AS-ANY announce NOT ANY
```

7. afi lists are also supported

(initially described in RFC4012), e.g.:

```
import-via: afi ipv4.unicast AS209752 from AS-ANY EXCEPT AS1200:AS-CUSTOMERS
accept ANY

export-via: afi ipv4.unicast AS209752 to AS-ANY EXCEPT AS1200:AS-CUSTOMERS
announce ANY
```

MN-IX route server objects

Relevant objects for participating peers in the Route Server project are grouped into these AS-SETs:

* AS-MN-IX-RS (list of connected peers)

* AS-MN-IX-RS-SETS (List of advertised AS-SETs)

* AS-MN-IX-RS-V6 (List of connected IPv6 peers)

* AS-MN-IX-RS-SETS-V6 (List of advertised AS-SETs for IPv6 peers)

* RS-MN-IX-ISP-LANS (List of all MN-IX peering LAN ranges)

* AS-MN-IX-SET (List of all route server ASNs)

BGP Community filtering

BGP Community filtering

Provide a BGP community filtering mechanism to peers

Route server peers are able to manipulate outbound routing policies via an in-band mechanism using BGP communities, instead of relying on import/import-via, export/export-via RPSL attributes. The downside to this method is that peers won’t be able to control inbound policies.

Note that you have to use the appropriate route server AS number, based on the MN-IX location you're peering in, with 209752 representing Manama. All locations support this feature.

The offered options are:

* Do not announce a prefix to a certain peer: 0:peer-as

* Announce a prefix to a certain peer: 209752:peer-as

* Do not announce a prefix to any peer: 0:209752

* Announce a prefix to all peers: 209752:209752

For destination peers employing a 32-bit ASN, you can use the route target extended BGP community as follows:

* do not announce a prefix to a certain peer: RT:0:peer-as

* announce a prefix to a certain peer: RT:209752:peer-as

* do not announce a prefix to any peer: RT:0:209752

AS-Path prepending can be done via the Route Servers by tagging prefixes using the following communities:

* using 209752:65501, to prepend the advertising peer customer AS once towards all other peers

* using 209752:65502, to prepend the advertising peer customer AS twice towards all other peers

* using 209752:65503, to prepend the advertising peer customer AS thrice towards all other peers

Additional Notes

* IRRDB policies work only on the AS level, whereas BGP communities work on the prefix level.

* IRRDB policies are parsed and applied hourly, whereas BGP communities are effective immediately, being in-band.

* BGP communities can only influence outbound (customer edge router to route server) announcements, whereas IRRDB policies can be used to influence inbound (route server to customer edge router) announcements, before reaching the customer edge router, thus potentially affecting the BGP decision process.

* Path hiding should not be a problem, as we are employing the BIRD 'secondary' configuration option.

* Note that validity of the IRRDB/RPKI based information provided is not guaranteed in any way.

Please consider carefully whether your MN-IX facing router should solely rely on information exchanged to and from the route servers.

Dynamic per-AS Prefix Limits

Dynamic per-AS Prefix Limits

Problem: route leaks

Route leaks are a problem. Either due to fat fingers, software bugs, or even malicious intent, route leaks are a fact of BGP life. A simple way to deal with the issue is using prefix limits.

Setting a static (fixed) limit to prevent customers from advertising more prefixes than intended does not really work for a route server service as the customer advertising the most prefixes has to be taken as the standard from which the limit is derived.

That leaves all the other customers with a wide margin in which they can freely leak routes; e.g. if the limit was set to 15,000, a customer advertising only one prefix could leak 14,999 more before being hitting the limit.

Adding insult to injury, this also has a cascading effect. Other route server peers having set a prefix limit for the session with the route servers, would potentially shut down the session, as they are now seeing thousands of additional prefixes.

Enter dynamic per-AS prefixes:

MN-IX is applying prefix limits specific to the AS connecting to the route server service. For instance, peers advertising only a couple of prefixes will have a maximum prefix limit of 100. Peers advertising thousands of prefixes will be calculated based on an proportional coefficient.

For examples and a breakdown of the formula used, see the FAQ below.

Fluctuations in advertisements are normal and expected. As long as these are within reason, our limits will adapt accordingly (hence 'dynamic').

FAQ

Q: I'm concerned that the limits for my AS are not big enough!

We hate to tear down sessions for no good reason, so rest assured that the limits are sufficiently relaxed. A 2 month lead period in which we were observing peer behavior and fine-tuned the algorithm ensured this as much as possible.

That being said, we value the stability of the service above everything else, so peers suddenly advertising thousands of prefixes when historically they have been advertising only a handful *will* hit the limit. In such cases, please contact us and we will be happy to reactivate the sessions.

Q: What is the prefix limit set for my AS?

Assuming you have member credentials, you can see your prefix limits here.

Q: I'm still concerned about the sanity of the limits, though.

We can also set a static limit for you, please contact us and state the limit you wish.

Q: Why not use IRRDB objects/RPKI to contain announcements?

MN-IX specifically wants to ensure that the route server service is as stable as possible. Having peers announce unexpectedly large amounts of prefixes wreaks havoc as it tears down sessions for considerable amounts of peers causing CPU churn to all parties involved. This is a different matter compared to the *type* of prefixes advertised.

IRRDB data is prone to inconsistencies and even more importantly, their usage is mostly limited to the western world. AS's from other regions of the world generally disregard IRRs.

Q: Can you give me examples of how this works?

Please consult the tables below:

Announced prefixes y	Coefficient x	Prefix Limit(yx < z)
y < 50	2	100
50 < y < 249	2	500
250 < y < 499	2	1000
500 < y < 999	2	2000
1000 < y < 2000	2 - 1,5	next step of 1000
2000 < y < 10000	1,5 - 2	next step of 1000
y > 10000	1,2	next step of 1000

Examples

25 Announced Prefixes x 2 = 50. Limit set to 100

51 Announced Prefixes x 2 = 102. Limit set to 500

300 Announced Prefixes x 2 = 600. Limit set to 1000

900 Announced Prefixes x 2 = 1800. Limit set to 2000

1500 Announced Prefixes x 1,35 = 2025. Limit set to 3000

9000 Announced Prefixes x 1,22 = 10980. Limit set to 11000

15000 Announced Prefixes x 1,2 = 18000. Limit set to 19000