

BCP/Technical/Routeserver/BIRD/Samples/GR-IX

From Euro-IX

< BCP | Technical | Routeserver | BIRD

Contents

- 1 Contact us
- 2 Credits
- 3 General
- 4 Architecture
- 5 BIRD Configuration
 - 5.1 IPv4
 - 5.2 IPv6
- 6 IXP Manager templates

Contact us

For any question or comment do not hesitate to contact apolyr@grnet.gr or noc@gr-ix.gr. We will be glad to discuss and cooperate!

Credits

GR-IX uses the IXP Manager (<https://github.com/inex/IXP-Manager>) as a members' portal. The IXP manager supports bird route servers and automatically produces configuration for them, based on (a) route server configuration templates (b) members' data and parameter from the IXP manager portal (eg addressing, AS numbers etc) and (c) information fetched from IRRDBs.

The configuration and templates presented here are based on the IXP manager templates.

General

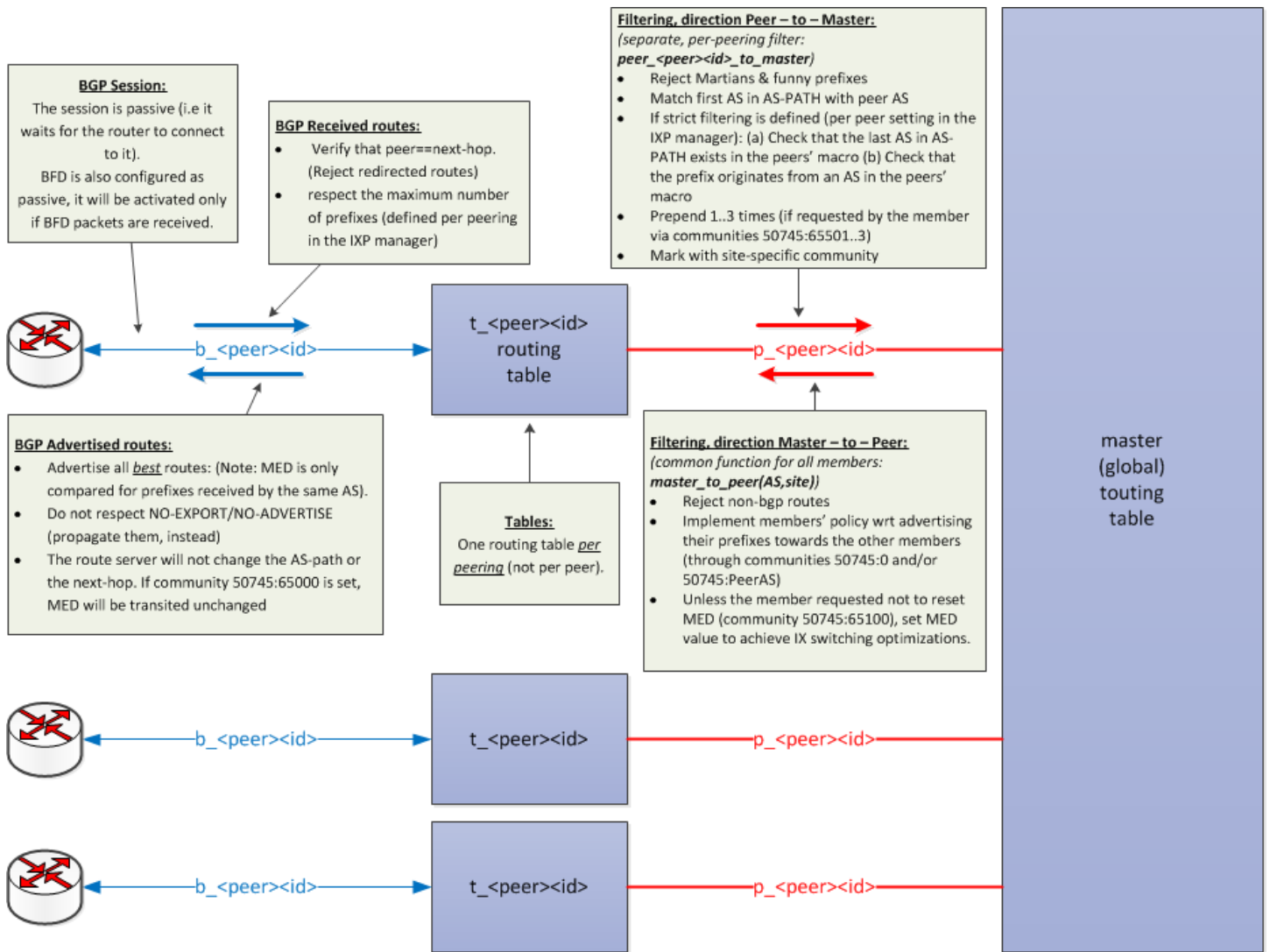
Although the configuration presented here is generic (does not require the use of the IXP manager), the design, the naming and generally the implementation is affected by the capabilities of the IXP manager itself.

However, the strict IRRDB filtering demands a dynamic way to produce & update the configuration, which implies the existence of some sort of script that produces it.

If you are not interested in IRRDB filtering and in dynamic production of the configuration, you can still use these configs; but you need to remove the respective IRRDB filtering parts from the filters.

Architecture

- One table per peering (not per peer) & one master (global) table
- The master table connects with each of the other tables through pipes
- Filtering takes place on the pipes:
 - Peer-to-Master direction: A peering-specific filter is used. The filter:
 - Rejects martians & other funny prefixes
 - Validates first AS in AS-PATH
 - If strict filtering has been enabled (per peer setting in the IXP manager), the last AS in AS-PATH and the prefix itself are validated through the peers' macro and IRRDB information
 - Checks if the peer has asked for prepending (via a specific community) and prepends
 - Sets a site/pop-specific community (i.e, marks the prefix according to the site where the physical connection is implemented)
 - Master-to-peer direction: A common function is used. The function:
 - Implements the advertisement policy, chosen by the originator of the prefix, (i.e., the prefix is advertised to all peers except from those whose ASNs is indicated via 50745:ASN (standard or extended) communities. If 50745:0 is used, the policy is reversed)
 - Unless 50745:65000 is used, the MED is reset according to the incoming & outgoing site combination, so that better MED is given to connections that are closer (0:same site, 2; adjacent site; 4 non-adjacent sites). If 50745:65000 is used, MED is not altered and the value sent by the member is respected.



Important notice: The route servers must use a different AS than the router of the exchange (if any). Otherwise the RS-IX peering will be IGP instead of EGP, this has some severe implications on the implementation.

Comments on Route Selection:

1. Prefer route with the highest Local Preference attribute.
(no practical use, Local Pref is not used in our implementation)
2. Prefer route with the shortest AS path.
(in use)
3. Prefer IGP origin over EGP and EGP origin over incomplete.
(in use)
4. Prefer the lowest value of the Multiple Exit Discriminator.
(in use, BUT only MED from prefixes advertised by peerings of the SAME AS are compared (always-compare-med not enabled))
5. Prefer routes received via eBGP over ones received via iBGP.
(no practical use, all peerings are eBGP)
6. Prefer routes with lower internal distance to a boundary router.
(no practical use, all distances are the same)
7. Prefer older routes
(this is not part of the route selection algorithm but it is a configurable option of the route server to avoid unstable routes)
8. Prefer the route with the lowest value of router ID of the advertising router.
(in use)

BIRD Configuration

IPv4

```
timeformat base    iso long;
timeformat log     iso long;
timeformat protocol iso long;
timeformat route   iso long;

log "/var/log/bird/bird-rs0-vlan505-ipv4.log" all;
log syslog all;
log stderr all;

# Turn on global debugging of all protocols
#debug protocols all;

define RSasn = 50745;
define RSaddress = 176.126.38.120;
```

```

router id 176.126.38.120;
listen bgp address 176.126.38.120;

# ignore interface up/down events
protocol device { }
protocol direct { }

### Filter Martians Function
# This function excludes weird networks
# rfc1918, class D, class E, too long and too short prefixes
function avoid_martians()
prefix set martians;
{
    martians = [
        10.0.0.0/8+,
        169.254.0.0/16+,
        172.16.0.0/12+,
        192.0.0.0/24+,
        192.0.2.0/24+,
        192.168.0.0/16+,
        198.18.0.0/15+,
        198.51.100.0/24+,
        203.0.113.0/24+,
        224.0.0.0/4+,
        240.0.0.0/4+,
        0.0.0.0/32-,
        0.0.0.0/0{28,32},
        0.0.0.0/0{0,7}
    ];

    # Avoid RFC1918 and similar networks
    if net ~ martians then {
        print "REJECTING ",net.ip,"/",net.len," received by ",from,": Martian/too long/too short prefix";
        return false;
    }

    return true;
}

### Filtering between members Function
# Applied on the pipes between the master table and each members' table
# Filtering is based on the "do not advertise to peer #peerid" community
function master_to_as(int peeras;string site)
{
    # reject non-BGP routes
    if ! (source = RTS_BGP) then return false;

    if peeras > 65535 then { # 32bit AS numbers require extended 48bit route target communities
        if ( ( !(RSasn,0) ~ bgp_community ) && (rt,RSasn,peeras) ~ bgp_ext_community ) # No (RSasn,0) exists: default policy, do not
        || ( (RSasn,0) ~ bgp_community && !(rt,RSasn,peeras) ~ bgp_ext_community ) # (RSasn,0) exists: reverse policy, advertise to
        ) then return false;
    } else { # 16bit AS numbers require standard communities
        if ( ( !(RSasn,0) ~ bgp_community ) && (RSasn,peeras) ~ bgp_community ) # No (RSasn,0) exists: default policy, do not adver
        || ( (RSasn,0) ~ bgp_community && !(RSasn,peeras) ~ bgp_community ) # (RSasn,0) exists: reverse policy, advertise to (rt,RSa
        ) then return false;
    }

    if !(RSasn,65000) ~ bgp_community) then { #MED values are not used by the member; they can be used to optimised swithing within the exc
        bgp_med = 0;
        if ( site = "EIE" && (RSasn,65102) ~ bgp_community ) then bgp_med = 2;
        if ( site = "EIE" && (RSasn,65103) ~ bgp_community ) then bgp_med = 2;
        if ( site = "LH" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
        if ( site = "LH" && (RSasn,65103) ~ bgp_community ) then bgp_med = 4;
        if ( site = "MED" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
        if ( site = "MED" && (RSasn,65102) ~ bgp_community ) then bgp_med = 4;
    }

    return true;
}

### ETH0 BFD configuration
protocol bfd rs_bfd {
    interface "eth0" {
        interval 400ms;
        multiplier 5;
        passive on;
    };
}

### GR-IX Route Server Client Template
template bgp grix_rs_client {
    local as RSasn;
    source address RSaddress;
    rs client;
    passive on;
    prefer older on;
    interpret communities off;
    graceful restart on;
    error wait time 60,43200;
    export all;
    import filter {
        ## Prevent BGP NEXT_HOP Hijacking
        if !( from = bgp_next_hop ) then
            reject "REJECTING ",net.ip,"/",net.len,": BGP neighbor address [", from, "] does not match next hop address [", bgp_next_hop
        accept;
    };
}

```

per member configuration (sample):

```

#### Customer ID: 5 - Full name: Greek Research & Technology Network - Shortname: grnet
#### AS number: 5408 - IPv4/6 address: 176.126.38.1 - ID in peering VLAN: 6 (0006)
#### Full location name: Εθνικό Ίδρυμα Ερευνών - Location shortname: EIE - Location tag:
#### Peering macro: AS5408:AS-T0-AIX - Maximum # of prefixes: 200 - BGP MD5 secret:

table t_grnet6;
filter peer_grnet6_to_master
prefix set grnet_prefixes;
int set grnet_ASNs;
{
    # Basic prefix validation. Do not propagate funny prefixes.
    # Check first AS in AS-PATH, esp to prevent route servers peering to each other
    if !(avoid_martians()) then reject;
    if (bgp_path.first != 5408 ) then reject;

    # Check AS-PATH against the appropriate IRRDB macro
    grnet_ASNs = [ 112, 2546, 3323, 5408, 5470, 5489, 6744, 6867, 8248, 8253,          # grnet-6 - AS5408 - 176.126.38.1
                  8278, 8522, 8581, 8611, 8617, 8618, 8643, 8700, 8762, 8991,      # grnet-6 - AS5408 - 176.126.38.1
                  9069, 12364, 12402, 15536, 15690, 15948, 20144, 20551, 20813, 31563, # grnet-6 - AS5408 - 176.126.38.1
                  42909, 44358, 47616, 48053, 50555, 50745, 197121, 197997, 199399, 201374 # grnet-6 - AS5408 - 176.126.38.1
                ];

    if !(bgp_path.last ~ grnet_ASNs ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-6, AS5408: Origin AS",bgp_path.last,"

    # Check each prefix against the route objects originating from the ASNs in the appropriate IRRDB macro
    grnet_prefixes = [ 139.91.0.0/16, 139.91.112.0/24, 139.91.247.0/24, 139.91.250.0/24, 139.91.251.0/24, # grnet-6 - AS5408 - 176.126.38.
                      139.91.252.0/24, 139.91.253.0/24, 143.233.0.0/16, 143.233.111.0/24, 143.233.169.0/24, # grnet-6 - AS5408 - 176.126.38.
                      143.233.172.0/22, 143.233.176.0/24, 143.233.182.0/23, 143.233.184.0/22, 143.233.188.0/23, # grnet-6 - AS5408 - 176.126.38.
                      143.233.204.0/24, 143.233.240.0/24, 143.233.43.0/24, 143.233.50.0/24, 143.233.51.0/24, # grnet-6 - AS5408 - 176.126.38.
                      # .....
                      83.212.28.0/22, 83.212.32.0/24, 83.212.58.0/23, 83.212.60.0/23, 83.212.64.0/22, # grnet-6 - AS5408 - 176.126.38.
                      83.212.80.0/21, 83.212.88.0/22, 83.212.9.0/24, 83.212.92.0/23, 83.212.94.0/24, # grnet-6 - AS5408 - 176.126.38.
                      83.212.96.0/19, 87.239.232.0/21, 88.197.0.0/17, 91.227.108.0/24 ];

    if ! (net ~ grnet_prefixes ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-6, AS5408: No route object for this prefix

    # Prepend 1,2,3 times, if (RSasn,65001), (RSasn,65002), (RSasn,65003) have been set
    if (RSasn,65501) ~ bgp_community then bgp_path.prepend(RSasn);
    if (RSasn,65502) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }
    if (RSasn,65503) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }

    # Mark prefixies with "site" community
    bgp_community.add ((RSasn,65101));

    # Done filtering & manipulation. Accept!
    accept;
}

protocol pipe p_grnet6 {
description "Pipe for grnet6 (AS5408 - 176.126.38.1)";
table master;
mode transparent;
peer table t_grnet6;
import filter peer_grnet6_to_master;
export where master_to_as(5408,"EIE");
}

protocol bgp b_grnet6 from grix_rs_client {
description "BGP for grnet6 (AS5408 - 176.126.38.1)";
neighbor 176.126.38.1 as 5408;
route limit 200;
table t_grnet6;
bfd on; # This should be configured for specific peers in the future.
}

#### Customer ID: 5 - Full name: Greek Research & Technology Network - Shortname: grnet
#### AS number: 5408 - IPv4/6 address: 176.126.38.31 - ID in peering VLAN: 45 (0045)
#### Full location name: Εθνικό Ίδρυμα Ερευνών - Location shortname: EIE - Location tag:
#### Peering macro: AS5408:AS-T0-AIX - Maximum # of prefixes: 300 - BGP MD5 secret:

table t_grnet45;
filter peer_grnet45_to_master
prefix set grnet_prefixes;
int set grnet_ASNs;
{
    # Basic prefix validation. Do not propagate funny prefixes.
    # Check first AS in AS-PATH, esp to prevent route servers peering to each other
    if !(avoid_martians()) then reject;
    if (bgp_path.first != 5408 ) then reject;

    # Check AS-PATH against the appropriate IRRDB macro
    grnet_ASNs = [ 112, 2546, 3323, 5408, 5470, 5489, 6744, 6867, 8248, 8253,          # grnet-45 - AS5408 - 176.126.38.31
                  8278, 8522, 8581, 8611, 8617, 8618, 8643, 8700, 8762, 8991,      # grnet-45 - AS5408 - 176.126.38.31
                  9069, 12364, 12402, 15536, 15690, 15948, 20144, 20551, 20813, 31563, # grnet-45 - AS5408 - 176.126.38.31
                  42909, 44358, 47616, 48053, 50555, 50745, 197121, 197997, 199399, 201374 # grnet-45 - AS5408 - 176.126.38.31
                ];

    if !(bgp_path.last ~ grnet_ASNs ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-45, AS5408: Origin AS",bgp_path.last,"

    # Check each prefix against the route objects originating from the ASNs in the appropriate IRRDB macro
    grnet_prefixes = [ 139.91.0.0/16, 139.91.112.0/24, 139.91.247.0/24, 139.91.250.0/24, 139.91.251.0/24, # grnet-6 - AS5408 - 176.126.38.
                      139.91.252.0/24, 139.91.253.0/24, 143.233.0.0/16, 143.233.111.0/24, 143.233.169.0/24, # grnet-6 - AS5408 - 176.126.38.
                      143.233.172.0/22, 143.233.176.0/24, 143.233.182.0/23, 143.233.184.0/22, 143.233.188.0/23, # grnet-6 - AS5408 - 176.126.38.
                      143.233.204.0/24, 143.233.240.0/24, 143.233.43.0/24, 143.233.50.0/24, 143.233.51.0/24, # grnet-6 - AS5408 - 176.126.38.
                      # .....
                      83.212.28.0/22, 83.212.32.0/24, 83.212.58.0/23, 83.212.60.0/23, 83.212.64.0/22, # grnet-6 - AS5408 - 176.126.38.

```

```
83.212.80.0/21, 83.212.88.0/22, 83.212.9.0/24, 83.212.92.0/23, 83.212.94.0/24, # grnet-6 - AS5408 - 176.126.38.
83.212.96.0/19, 87.239.232.0/21, 88.197.0.0/17, 91.227.108.0/24 ];
```

```
if ! (net ~ grnet_prefixes ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-45, AS5408: No route object for this prefix

# Prepend 1,2,3 times, if (RSasn,65001), (RSasn,65002), (RSasn,65003) have been set
if (RSasn,65501) ~ bgp_community then bgp_path.prepend(RSasn);
if (RSasn,65502) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }
if (RSasn,65503) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }

# Mark prefixies with "site" community
bgp_community.add ((RSasn,65101));

# Done filtering & manipulation. Accept!
accept;
}

protocol pipe p_grnet45 {
description "Pipe for grnet45 (AS5408 - 176.126.38.31)";
table master;
mode transparent;
peer table t_grnet45;
import filter peer_grnet45_to_master;
export where master_to_as(5408,"EIE");
}

protocol bgp b_grnet45 from grix_rs_client {
description "BGP for grnet45 (AS5408 - 176.126.38.31)";
neighbor 176.126.38.31 as 5408;
route limit 300;
table t_grnet45;
bfd on; # This should be configured for specific peers in the future.
}
```

IPv6

```
timeformat base iso long;
timeformat log iso long;
timeformat protocol iso long;
timeformat route iso long;

log "/var/log/bird/bird-rs0-vlan505-ipv6.log" all;
log syslog all;
log stderr all;

# Turn on global debugging of all protocols
#debug protocols all;

define RSasn = 50745;
define RSaddress = 2001:7f8:6e::120;

router id 176.126.38.120;
listen bgp address 2001:7f8:6e::120;

# ignore interface up/down events
protocol device { }
protocol direct { }

### Filter Martians Function
# This function excludes weird networks
# rfc1918, class D, class E, too long and too short prefixes
function avoid_martians()
prefix set martians;
{
    martians = [
        ::/0, # Default (can be advertised as a route in BGP to peers if desired)
        ::/96, # IPv4-compatible IPv6 address <E2><80><93> deprecated by RFC4291
        ::/128, # Unspecified address
        ::1/128, # Local host loopback address
        ::ffff:0.0.0.0/96+, # IPv4-mapped addresses
        ::224.0.0.0/100+, # Compatible address (IPv4 format)
        ::127.0.0.0/104+, # Compatible address (IPv4 format)
        ::0.0.0.0/104+, # Compatible address (IPv4 format)
        ::255.0.0.0/104+, # Compatible address (IPv4 format)
        0000::/8+, # Pool used for unspecified, loopback and embedded IPv4 addresses
        0200::/7+, # OSI NSAP-mapped prefix set (RFC4548) <E2><80><93> deprecated by RFC4048
        3ffe::/16+, # Former 6bone, now decommissioned
        2001:db8::/32+, # Reserved by IANA for special purposes and documentation
        2002:e000::/20+, # Invalid 6to4 packets (IPv4 multicast)
        2002:7f00::/24+, # Invalid 6to4 packets (IPv4 loopback)
        2002:0000::/24+, # Invalid 6to4 packets (IPv4 default)
        2002:ff00::/24+, # Invalid 6to4 packets
        2002:0a00::/24+, # Invalid 6to4 packets (IPv4 private 10.0.0.0/8 network)
        2002:ac10::/28+, # Invalid 6to4 packets (IPv4 private 172.16.0.0/12 network)
        2002:c0a8::/32+, # Invalid 6to4 packets (IPv4 private 192.168.0.0/16 network)
        fc00::/7+, # Unicast Unique Local Addresses (ULA) <E2><80><93> RFC 4193
        fe80::/10+, # Link-local Unicast
        fec0::/10+, # Site-local Unicast <E2><80><93> deprecated by RFC 3879 (replaced by ULA)
        ff00::/8+ # Multicast
    ];

# Avoid RFC1918 and similar networks
if net ~ martians then {
print "REJECTING ",net.ip,"/",net.len," received by ",from,: Martian/too long/too short prefix";
return false;
}
}
```

```

    return true;
}

### Filtering between members Function
# Applied on the pipes between the master table and each members' table
# Filtering is based on the "do not advertise to peer #peerid" community
function master_to_as(int peeras;string site)
{
    # reject non-BGP routes
    if ! (source = RTS_BGP) then return false;

    if peeras > 65535 then { # 32bit AS numbers require extended 48bit route target communities
        if ( !(RSasn,0) ~ bgp_community ) && (rt,RSasn,peeras) ~ bgp_ext_community ) # No (RSasn,0) exists: default policy, do not
        || ( (RSasn,0) ~ bgp_community && !((rt,RSasn,peeras) ~ bgp_ext_community) ) # (RSasn,0) exists: reverse policy, advertise to
        ) then return false;
    } else { # 16bit AS numbers require standard communities
        if ( !(RSasn,0) ~ bgp_community ) && (RSasn,peeras) ~ bgp_community ) # No (RSasn,0) exists: default policy, do not adver
        || ( (RSasn,0) ~ bgp_community && !((RSasn,peeras) ~ bgp_community) ) # (RSasn,0) exists: reverse policy, advertise to (rt,RSa
        ) then return false;
    }

    if !((RSasn,65000) ~ bgp_community) then { #MED values are not used by the member; they can be used to optimised swithing within the exc
        bgp_med = 0;
        if ( site = "EIE" && (RSasn,65102) ~ bgp_community ) then bgp_med = 2;
        if ( site = "EIE" && (RSasn,65103) ~ bgp_community ) then bgp_med = 2;
        if ( site = "LH" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
        if ( site = "LH" && (RSasn,65103) ~ bgp_community ) then bgp_med = 4;
        if ( site = "MED" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
        if ( site = "MED" && (RSasn,65102) ~ bgp_community ) then bgp_med = 4;
    }

    return true;
}

### ETH0 BFD configuration
protocol bfd rs_bfd {
    interface "eth0" {
        interval 400ms;
        multiplier 5;
        passive on;
    };
}

### GR-IX Route Server Client Template
template bgp grix_rs_client {
    local as RSasn;
    source address RSaddress;
    rs client;                # Preserve AS-PATH, next-hop and MED
    passive on;              # Do not initiate bgp sessions
    prefer older on;        # Do not bread ties through IDs; keep the older (more stable) route
    interpret communities off; # Do not respect NO-EXPORT and NO-ADVERTISE (and preserve them in the outgoing advertisements)
    graceful restart on;    # Participate in graceful restart recovery
    error wait time 60,43200;
    export all;
    import all;
}

```

per member configuration (sample):

```

### Customer ID: 5 - Full name: Greek Research & Technology Network - Shortname: grnet
### AS number: 5408 - IPv4/6 address: 2001:7f8:6e::1 - ID in peering VLAN: 6 (0006)
### Full location name: Εθνικό Ιδρυμα Ερευνών - Location shortname: EIE - Location tag:
### Peering macro: AS5408:AS-T0-AIX - Maximum # of prefixes: 200 - BGP MD5 secret:

table t_grnet6;

filter peer_grnet6_to_master
prefix set grnet_prefixes;
int set grnet_ASNs;
{
    # Basic prefix validation. Do not propagate funny prefixes.
    # Check first AS in AS-PATH, esp to prevent route servers peering to each other
    if !(avoid_martians()) then reject;
    if (bgp_path.first != 5408 ) then reject;

    # Check AS-PATH against the appropriate IRRDB macro
    grnet_ASNs = [ 112, 2546, 3323, 5408, 5470, 5489, 6744, 6867, 8248, 8253,                # grnet-6 - AS5408 - 2001:7f8:6e::1
                  8278, 8522, 8581, 8611, 8617, 8618, 8643, 8700, 8762, 8991,          # grnet-6 - AS5408 - 2001:7f8:6e::1
                  9069, 12364, 12402, 15536, 15690, 15948, 20144, 20551, 20813, 31563, # grnet-6 - AS5408 - 2001:7f8:6e::1
                  42909, 44358, 47616, 48053, 50555, 50745, 197121, 197997, 199399, 201374 # grnet-6 - AS5408 - 2001:7f8:6e::1
                ];

    if !(bgp_path.last ~ grnet_ASNs ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-6, AS5408: Origin AS",bgp_path.last,"

    # Check each prefix against the route objects originating from the ASNs in the appropriate IRRDB macro
    grnet_prefixes = [ 2001:500:3::/48, 2001:500:9c::/48, 2001:500:9d::/48, 2001:500:9e::/47, 2001:500:9f::/48,    # grnet-6 - AS5408 - 200
                      2001:648::/32, 2001:648:2000::/48, 2001:648:2001::/48, 2001:648:2002::/48, 2001:648:2010::/48,    # grnet-6 - AS5408 - 200
                      2001:648:2011::/48, 2001:648:2020::/48, 2001:648:2030::/48, 2001:648:2050::/48, 2001:648:2050::/52,    # grnet-6 - AS5408 - 200
                      # .....
                      2001:678:4::/48, 2001:678:5::/48, 2001:678:6::/48, 2001:678:7::/48, 2001:678:88::/48,          # grnet-6 - AS5408 - 200
                      2001:678:8c::/48, 2001:67c:229c::/48, 2001:7f8:6e::/48, 2002::/16, 2620:4f:8000::/48          # grnet-6 - AS5408 - 2001:7f8:6e
                    ];

    if ! (net ~ grnet_prefixes ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-6, AS5408: No route object for this prefix

    # Prepend 1,2,3 times, if (RSasn,65001), (RSasn,65002), (RSasn,65003) have been set
}

```

```

    if (RSasn,65501) ~ bgp_community then bgp_path.prepend(RSasn);
    if (RSasn,65502) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }
    if (RSasn,65503) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }

    # Mark prefixies with "site" community
    bgp_community.add ((RSasn,65101));

    # Done filtering & manipulation. Accept!
    accept;
}

protocol pipe p_grnet6 {
    description "Pipe for grnet6 (AS5408 - 2001:7f8:6e::1)";
    table master;
    mode transparent;
    peer table t_grnet6;
    import filter peer_grnet6_to_master;
    export where master_to_as(5408,"EIE");
}

protocol bgp b_grnet6 from grix_rs_client {
    description "BGP for grnet6 (AS5408 - 2001:7f8:6e::1)";
    neighbor 2001:7f8:6e::1 as 5408;
    route limit 200;
    table t_grnet6;
    bfd on; # This should be configured for specific peers in the future.
}

### Customer ID: 5 - Full name: Greek Research & Technology Network - Shortname: grnet
### AS number: 5408 - IPv4/6 address: 2001:7f8:6e::31 - ID in peering VLAN: 45 (0045)
### Full location name: Εθνικό Ίδρυμα Ερευνών - Location shortname: EIE - Location tag:
### Peering macro: AS5408:AS-TO-AIX - Maximum # of prefixes: 300 - BGP MD5 secret:

table t_grnet45;

filter peer_grnet45_to_master
prefix set grnet_prefixes;
int set grnet_ASNs;
{
    # Basic prefix validation. Do not propagate funny prefixies.
    # Check first AS in AS-PATH, esp to prevent route servers peering to each other
    if !(avoid_martians()) then reject;
    if (bgp_path.first != 5408 ) then reject;

    # Check AS-PATH against the appropriate IRRDB macro
    grnet_ASNs = [ 112, 2546, 3323, 5408, 5470, 5489, 6744, 6867, 8248, 8253, # grnet-45 - AS5408 - 2001:7f8:6e::31
                  8278, 8522, 8581, 8611, 8617, 8618, 8643, 8700, 8762, 8991, # grnet-45 - AS5408 - 2001:7f8:6e::31
                  9069, 12364, 12402, 15536, 15690, 15948, 20144, 20551, 20813, 31563, # grnet-45 - AS5408 - 2001:7f8:6e::31
                  42909, 44358, 47616, 48053, 50555, 50745, 197121, 197997, 199399, 201374 # grnet-45 - AS5408 - 2001:7f8:6e::31
                ];

    if !(bgp_path.last ~ grnet_ASNs ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-45, AS5408: Origin AS",bgp_path.last,"";

    # Check each prefix against the route objects originating from the ASNs in the appropriate IRRDB macro
    grnet_prefixes = [ 2001:500:3::/48, 2001:500:9c::/48, 2001:500:9d::/48, 2001:500:9e::/47, 2001:500:9f::/48, # grnet-45 - AS5408 - 20
                      2001:648::/32, 2001:648:2000::/48, 2001:648:2001::/48, 2001:648:2002::/48, 2001:648:2010::/48, # grnet-45 - AS5408 - 20
                      # .....
                      2001:678:4::/48, 2001:678:5::/48, 2001:678:6::/48, 2001:678:7::/48, 2001:678:88::/48, # grnet-45 - AS5408 - 20
                      2001:678:8c::/48, 2001:67c:229c::/48, 2001:7f8:6e::/48, 2002::/16, 2620:4f:8000::/48 # grnet-45 - AS5408 - 2001:7f8:6
                    ];

    if !(net ~ grnet_prefixes ) then reject "REJECTING ",net.ip,"/",net.len," received by grnet-45, AS5408: No route object for this prefix";

    # Prepend 1,2,3 times, if (RSasn,65001), (RSasn,65002), (RSasn,65003) have been set
    if (RSasn,65501) ~ bgp_community then bgp_path.prepend(RSasn);
    if (RSasn,65502) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }
    if (RSasn,65503) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }

    # Mark prefixies with "site" community
    bgp_community.add ((RSasn,65101));

    # Done filtering & manipulation. Accept!
    accept;
}

protocol pipe p_grnet45 {
    description "Pipe for grnet45 (AS5408 - 2001:7f8:6e::31)";
    table master;
    mode transparent;
    peer table t_grnet45;
    import filter peer_grnet45_to_master;
    export where master_to_as(5408,"EIE");
}

protocol bgp b_grnet45 from grix_rs_client {
    description "BGP for grnet45 (AS5408 - 2001:7f8:6e::31)";
    neighbor 2001:7f8:6e::31 as 5408;
    route limit 300;
    table t_grnet45;
    bfd on; # This should be configured for specific peers in the future.
}

```

IXP Manager templates

header.cfg:

```

#####
#
# Bird Route Server configuration for VLAN: {$vlan->getName()} (Tag: {$vlan->getNumber()})
#
# -----
# This configuration was generated automatically by IXP Manager
#   Generated: {$smarty.now|date_format:'Y-m-d H:i:s'}
#
# Attention: Do not edit manually, it will be overwritten automatically.
#
#####
# BASIC DOCUMENTATION
#
#
# 1. Communities
# -----
#
# ** Advertisement Control: **
# RSasn,PeerAS (1<=PeerAS<65000) or route target extended community rt,RSasn,PeerAS, PeerAS>65536): Do not advertise to PeerAS
# RSasn,0 : Inverse advertising policy (do not advertise to any peers, except from those defined with RSasn:PeerASn)
#
# ** Prepending (communities can be combined)**
# RSasn,65501: prepend one time
# RSasn,65502: prepend two times
# RSasn,65503: prepend three times
#
# ** MED: **
# RSasn,65000: Do not alter incoming MED for IX switching optimisation
#
# ** Site-Marking: **
# RSasn,65101: Prefix received from a peering at EIE site
# RSasn,65102: Prefix received from a peering at LH site
# RSasn,65103: Prefix received from a peering at MED site
#
#
# 2. Route selection
# -----
#
# 1. Prefer route with the highest Local Preference attribute.
#   (no practical use, Local Pref is not used in our implementation)
# 2. Prefer route with the shortest AS path.
#   (in use)
# 3. Prefer IGP origin over EGP and EGP origin over incomplete.
#   (in use)
# 4. Prefer the lowest value of the Multiple Exit Discriminator.
#   (in use, BUT only MED from prefixes advertised by peerings of the SAME AS are compared (always-compare-med not enabled))
# 5. Prefer routes received via eBGP over ones received via iBGP.
#   (no practical use, all peerings are eBGP)
# 6. Prefer routes with lower internal distance to a boundary router.
#   (no practical use, all distances are the same)
# 7. Prefer older routes
#   (this is not part of the route selection algorithm but it is a configurable option of the route server to avoid unstable routes)
# 8. Prefer the route with the lowest value of router ID of the advertising router.
#   (in use)
#
#
#####
timeformat base    iso long;
timeformat log     iso long;
timeformat protocol iso long;
timeformat route  iso long;

log "#{rsconfLogfile#}" all;
log syslog all;
log stderr all;

# Turn on global debugging of all protocols
#debug protocols all;

define RSasn = {#rsconfASN#};
define RSaddress = {#rsconfListenAddr#};

router id {#rsconfRouterID#};
listen bgp address {#rsconfListenAddr#};

# ignore interface up/down events
protocol device { }
protocol direct { }

### Filter Martians Function
# This function excludes weird networks
# rfc1918, class D, class E, too long and too short prefixes
function avoid_martians()
prefix set martians;
{
  {if $proto eq 6}

    martians = [
      ::/0,          # Default (can be advertised as a route in BGP to peers if desired)
      ::/96,        # IPv4-compatible IPv6 address <E2><80><93> deprecated by RFC4291
      ::/128,       # Unspecified address
      ::1/128,      # Local host loopback address
      ::ffff:0.0.0.0/96+, # IPv4-mapped addresses
      ::224.0.0.0/100+, # Compatible address (IPv4 format)
      ::127.0.0.0/104+, # Compatible address (IPv4 format)
      ::0.0.0.0/104+,  # Compatible address (IPv4 format)
      ::255.0.0.0/104+, # Compatible address (IPv4 format)
      0000::/8+,      # Pool used for unspecified, loopback and embedded IPv4 addresses
      0200::/7+,      # OSI NSAP-mapped prefix set (RFC4548) <E2><80><93> deprecated by RFC4048
      3ffe::/16+,     # Former 6bone, now decommissioned
    ]
  }
}

```



```

2001:db8::/32+, # Reserved by IANA for special purposes and documentation
2002:e000::/20+, # Invalid 6to4 packets (IPv4 multicast)
2002:7f00::/24+, # Invalid 6to4 packets (IPv4 loopback)
2002:0000::/24+, # Invalid 6to4 packets (IPv4 default)
2002:ff00::/24+, # Invalid 6to4 packets
2002:0a00::/24+, # Invalid 6to4 packets (IPv4 private 10.0.0.0/8 network)
2002:ac10::/28+, # Invalid 6to4 packets (IPv4 private 172.16.0.0/12 network)
2002:c0a8::/32+, # Invalid 6to4 packets (IPv4 private 192.168.0.0/16 network)
fc00::/7+, # Unicast Unique Local Addresses (ULA) <E2><80><93> RFC 4193
fe80::/10+, # Link-local Unicast
fec0::/10+, # Site-local Unicast <E2><80><93> deprecated by RFC 3879 (replaced by ULA)
ff00::/8+ # Multicast

];
{else}

martians = [
10.0.0.0/8+,
169.254.0.0/16+,
172.16.0.0/12+,
192.0.0.0/24+,
192.0.2.0/24+,
192.168.0.0/16+,
198.18.0.0/15+,
198.51.100.0/24+,
203.0.113.0/24+,
224.0.0.0/4+,
240.0.0.0/4+,
0.0.0.0/32-,
0.0.0.0/0{ldelim}28,32{rdelim},
0.0.0.0/0{ldelim}0,7{rdelim}
];
{/if}

# Avoid RFC1918 and similar networks
if net ~ martians then {
print "REJECTING ",net.ip,"/",net.len," received by ",from,": Martian/too long/too short prefix";
return false;
}

return true;
}

### Filtering between members Function
# Applied on the pipes between the master table and each members' table
# Filtering is based on the "do not advertise to peer #peerid" community
function master_to_as(int peeras;string site)
{
# reject non-BGP routes
if ! (source = RTS_BGP) then return false;

if peeras > 65535 then { # 32bit AS numbers require extended 48bit route target communities
if ( ( !(RSasn,0) ~ bgp_community) && (rt,RSasn,peeras) ~ bgp_ext_community ) # No (RSasn,0) exists: default policy, do not
|| ( (RSasn,0) ~ bgp_community && !(rt,RSasn,peeras) ~ bgp_ext_community ) ) # (RSasn,0) exists: reverse policy, advertise to
} then return false;
} else { # 16bit AS numbers require standard communities
if ( ( !(RSasn,0) ~ bgp_community) && (RSasn,peeras) ~ bgp_community ) # No (RSasn,0) exists: default policy, do not adver
|| ( (RSasn,0) ~ bgp_community && !(RSasn,peeras) ~ bgp_community ) ) # (RSasn,0) exists: reverse policy, advertise to (rt,RSa
) then return false;
}

if !(RSasn,65000) ~ bgp_community) then { #MED values are not used by the member; they can be used to optimised swithing within the exc
bgp_med = 0;
if ( site = "EIE" && (RSasn,65102) ~ bgp_community ) then bgp_med = 2;
if ( site = "EIE" && (RSasn,65103) ~ bgp_community ) then bgp_med = 2;
if ( site = "LH" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
if ( site = "LH" && (RSasn,65103) ~ bgp_community ) then bgp_med = 4;
if ( site = "MED" && (RSasn,65101) ~ bgp_community ) then bgp_med = 2;
if ( site = "MED" && (RSasn,65102) ~ bgp_community ) then bgp_med = 4;
}

return true;
}

### ETH0 BFD configuration
protocol bfd rs_bfd {
interface "eth0" {
interval 400ms;
multiplier 5;
passive on;
};
}

### GR-IX Route Server Client Template
template bgp grix_rs_client {
local as RSasn;
source address RSaddress;
rs client; # Preserve AS-PATH, next-hop and MED
passive on; # Do not initiate bgp sessions
prefer older on; # Do not bread ties through IDs; keep the older (more stable) route
interpret communities off; # Do not respect NO-EXPORT and NO-ADVERTISE (and preserve them in the outgoing advertisements)
graceful restart on; # Participate in graceful restart recovery
error wait time 60,43200;
export all;
{if $proto eq 4}
import filter {
## Prevent BGP NEXT_HOP Hijacking
if !( from = bgp_next_hop ) then
reject "REJECTING ",net.ip,"/",net.len,": BGP neighbor address [", from, "]" does not match next hop address [", bgp_next_hop
accept;
};
}

```

```

    {else}
    import all;
    {/if}
}
##### END OF COMMON CONFIG, PER MEMBER CONFIGURATION FOLLOWS #####

```

neighbors.cfg:

```

### Customer ID: ${int.cid} - Full name: ${int.cname} - Shortname: ${int.cshortname}
### AS number: ${int.autsys} - IPv4/6 address: ${int.address} - ID in peering VLAN: ${int.vliid} (${int.fvliid})
### Full location name: ${int.location_name} - Location shortname: ${int.location_shortname} - Location tag: ${int.location_tag}
### Peering macro: ${int.peeringmacro} - Maximum # of prefixes: ${int.maxprefixes} - BGP MD5 secret: ${int.bgpm5secret}

table t_${int.cshortname}${int.vliid};

filter peer_${int.cshortname}${int.vliid}_to_master
prefix set ${int.cshortname}_prefixes;
int set ${int.cshortname}_ASNs;
{
    # Basic prefix validation. Do not propagate funny prefixes.
    # Check first AS in AS-PATH, esp to prevent route servers peering to each other
    if !(avoid_martians()) then reject;
    if (bgp_path.first != ${int.autsys} ) then reject;

{* Only do filtering if this is enabled per client *}
{if ${int['irrdfilter']} }
    # Check AS-PATH against the appropriate IRRDB macro
    ${int.cshortname}_ASNs = [ {foreach $irrdbs as $a}${a.asn}{if not $a@last},{/if} {if $a@iteration % 10 == 0} # ${int.cshortname}
        {/if}{/foreach} ];

    if !(bgp_path.last ~ ${int.cshortname}_ASNs ) then reject "REJECTING ",net.ip,"/",net.len," received by ${int.cshortname}-${int.vliid},

{if count( $prefixes )}
    # Check each prefix against the route objects originating from the ASNs in the appropriate IRRDB macro
    ${int.cshortname}_prefixes = [ {foreach $prefixes as $p}${p.prefix}{if not $p@last},{/if} {if $p@iteration % 5 == 0} # ${int.cshortname}
        {/if}{/foreach} ];

    if ! (net ~ ${int.cshortname}_prefixes ) then reject "REJECTING ",net.ip,"/",net.len," received by ${int.cshortname}-${int.vliid}, AS${int.cshortname}

{else}
    # Deny everything because the IRRDB returned nothing
    reject;
{/if}
{else}
    # This ASN was configured not to use IRRDB filtering
{/if}

    # Prepend 1,2,3 times, if (RSasn,65001), (RSasn,65002), (RSasn,65003) have been set
    if (RSasn,65501) ~ bgp_community then bgp_path.prepend(RSasn);
    if (RSasn,65502) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }
    if (RSasn,65503) ~ bgp_community then { bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); bgp_path.prepend(RSasn); }

    # Mark prefixes with "site" community
    {if ${int.location_shortname} eq "EIE"}
    bgp_community.add ((RSasn,65101));
{/if}
    {if ${int.location_shortname} eq "LH"}
    bgp_community.add ((RSasn,65102));
{/if}
    {if ${int.location_shortname} eq "MED"}
    bgp_community.add ((RSasn,65103));
{/if}

    # Done filtering & manipulation. Accept!
    accept;
}

protocol pipe p_${int.cshortname}${int.vliid} {
    description "Pipe for ${int.cshortname}${int.vliid} (AS${int.autsys} - ${int.address})";
    table master;
    mode transparent;
    peer table t_${int.cshortname}${int.vliid};
    import filter peer_${int.cshortname}${int.vliid}_to_master;
    export where master_to_as(${int.autsys},"${int.location_shortname}");
}

protocol bgp b_${int.cshortname}${int.vliid} from grix_rs_client {
    description "BGP for ${int.cshortname}${int.vliid} (AS${int.autsys} - ${int.address})";
    neighbor ${int.address} as ${int.autsys};
    route limit ${int.maxprefixes};
    table t_${int.cshortname}${int.vliid};
    bfd on; # This should be configured for specific peers in the future.
{if ${int.bgpm5secret} password "${int.bgpm5secret}";{/if}
}

```

Retrieved from "<https://wiki.euro-ix.net/index.php/BCP/Technical/Routeserver/BIRD/Samples/GR-IX>"

- This page was last modified on 15 May 2015, at 11:16.