Route Servers

Arnaud FENIOUX - FranceIX

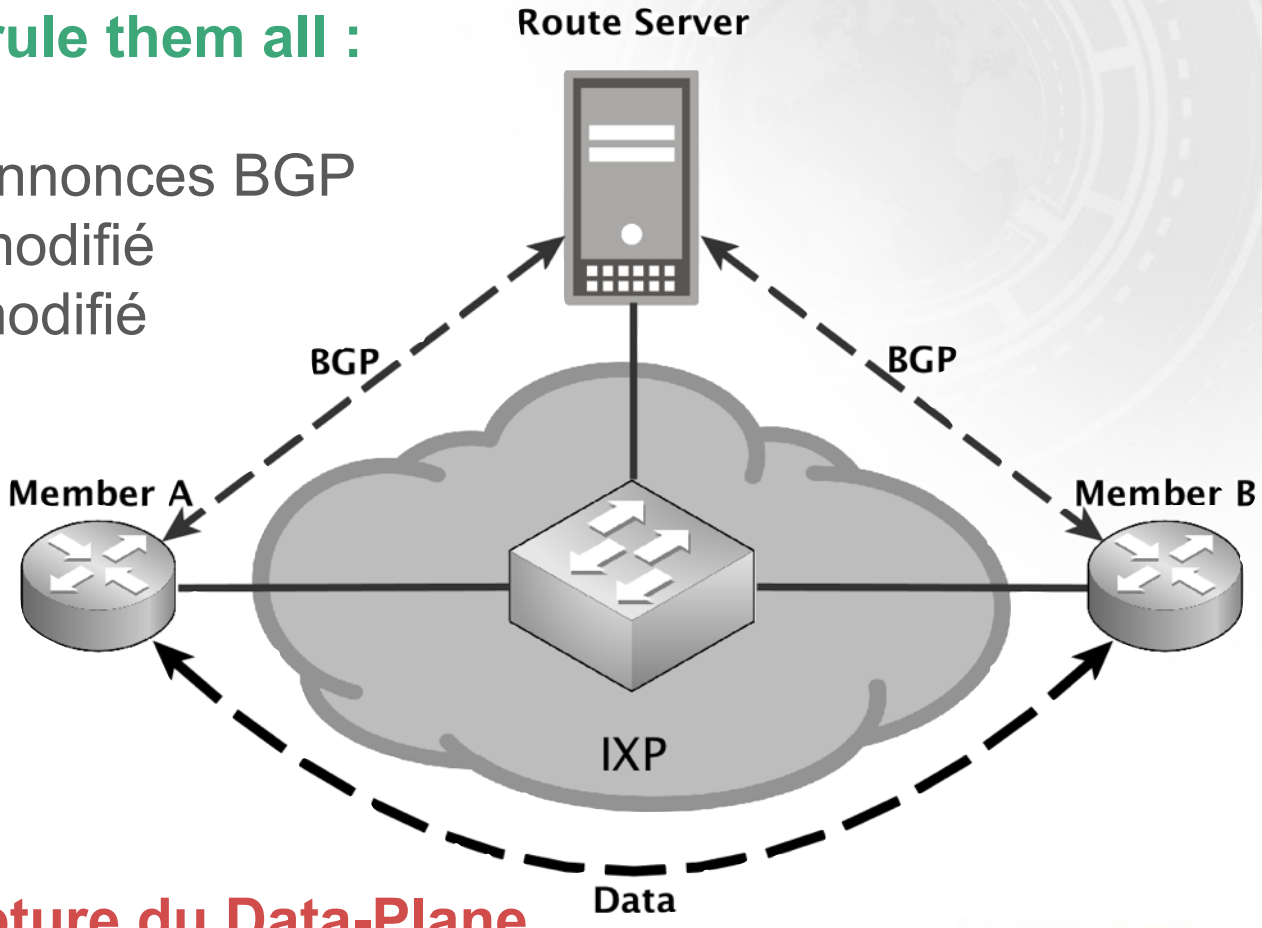
Route Server

Fonctionnalités

Data plane vs Control plane

One session to rule them all :

- Centralise les annonces BGP
- AS-PATH non modifié
- Next-hop non modifié
- Trafic en direct



/!\ Blackholing si rupture du Data-Plane

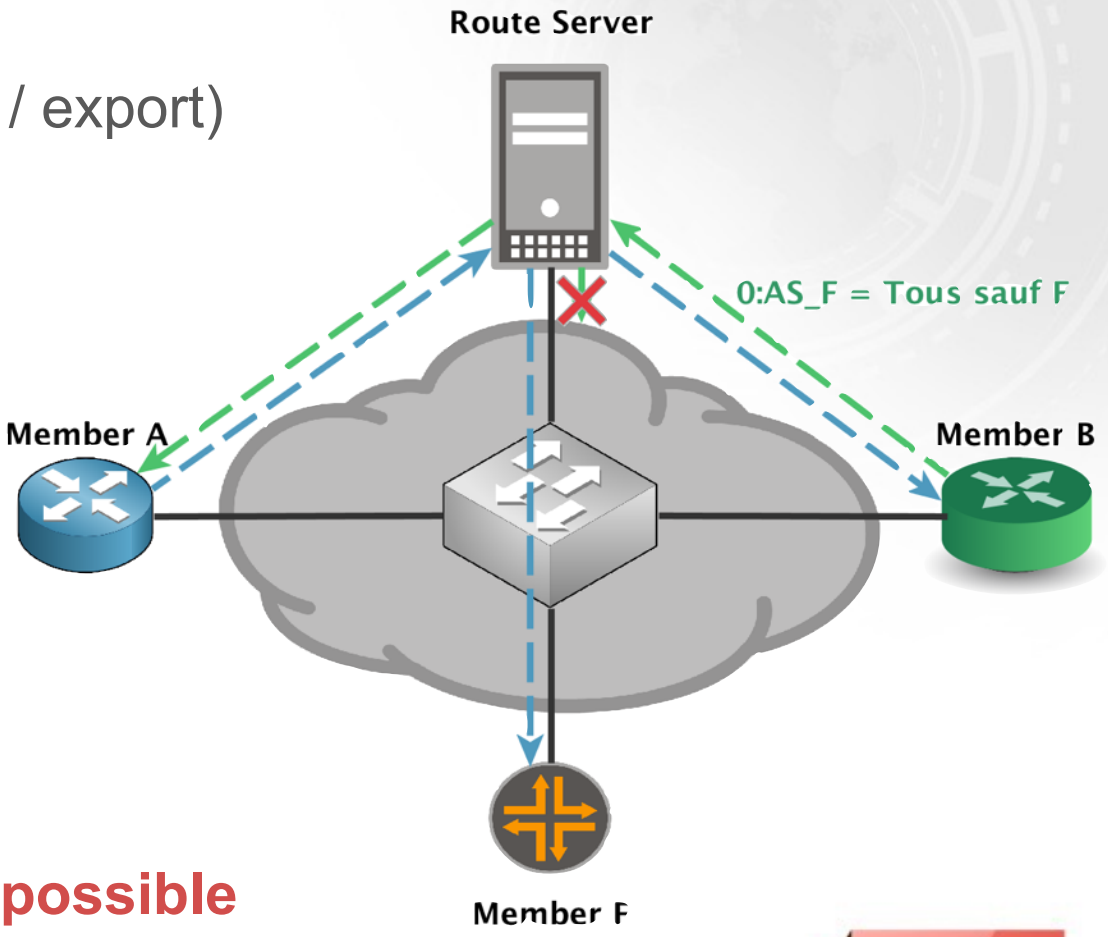
Annonces sélectives

via:

- Communautés BGP
- IRR (aut-num import / export)

- Filtrage
- AS-PATH prepending
- Ré-écriture de la MED

0:peer-as = Don't send route to this peer AS



/!\ Asymétrie de trafic possible

Route Server

Sécurités

Fat finger errors

Martians (IPv4 et v6)

- Filtrage des préfixes martiens
  https://www.team-cymru.org/bogon-dotted-decimal.html

Max prefix limit

- Limite le nombre de préfixes appris par peer sur les RS
  Coupe la session BGP si le seuil est dépassé

Prefix length

- IPv4 : /8 a /24 sont autorisés
- IPv6 : /19 a /48 sont autorisés

Protège contre :

- leaks massifs / leaks de routes internes

“Thin” finger errors

Next-hop

- Vérification que l’IP next-hop dans l’update BGP est aussi l’IP source du paquet

First AS in AS-PATH

- Vérification que le premier AS de l’AS-PATH est l’AS du peer BGP

Protège contre :

- Les annonces BGP falsifiées
- Redirection de trafic vers une victime
- Masquage de l’AS attaquant

IRR Lock Down AS-SET ou ASN

- N'autorise que les préfixes enregistrés par certains AS-SET ou ASN

AS-SET -> AUT-NUM -> ROUTE(6) -> INETNUM(6)

IRR Explorer + BGPQ3 = <3
http://irrexplorer.nlnog.net/
http://peering.readthedocs.org/en/latest/PrefixLists.html

Protège contre :

- Hijacking de prefixes

/!\ dépend de la qualité des données dans les IRR

RPKI / ROA

RPKI / ROA

- Valide que l'AS à l'origine de l'annonce est autorisé à annoncer ce préfixe.

Enregistrement via le LIR Portal :
https://www.ripe.net/manage-ips-and-asns/resource-management/certification/resource-certification-roa-management

Evite :

- Certains hijacking de prefixes

/!\ Ne valide pas la transitivité

Route Server

Installation BIRD

BIRD Linux Debian/Ubuntu

Il est préconisé d'utiliser le repository maintenu par les devs

wget -O - http://bird.network.cz/debian/apt.key | apt-key add -
echo "deb http://bird.network.cz/debian/ wheezy main" > /etc/apt/sources.list.d/bird.list
apt-get update
apt-get install bird

BIRD Linux Debian/Ubuntu

Configuration systeme Linux :

Par défaut le uRPF est activé, ce qui empeche un ping arrivant via eth1 soit répondu via eth0 (la default-gw) il faut donc éditer **/etc/sysctl.conf**

net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0

On veillera a laisser l'ip_forwarding a 0 (conf par défaut) pour éviter que les members puissent default router via les RS!

BIRD Linux Debian/Ubuntu

Compilation (si besoin) exemple en 1.4.5

```
apt-get install build-essential flex bison libncurses5-dev libreadline-dev
cd /usr/local/src
wget ftp://bird.network.cz/pub/bird/bird-1.4.5.tar.gz
tar xzf bird-1.4.5.tar.gz
cd bird-1.4.5

./configure --prefix=/usr --sysconfdir=/etc/bird --localstatedir=/var --with-
runtimedir=/run/bird --enable-client
make
make install
make clean

./configure --prefix=/usr --sysconfdir=/etc/bird --localstatedir=/var --with-
runtimedir=/run/bird --enable-client --enable-ipv6
make
make install
make clean
```

Route Server

Commandes CLI

BIRD CLI

Apres voir modifié les conf, il est bien de checker la conf avant de reloader bird :

bird -p -c /etc/bird/bird.conf

bird6 -p -c /etc/bird/bird6.conf

Pour entrer dans le **CLI**, il faut tapper :

birdc

birdc6

On peut aussi appeler birdc depuis bash :

birdc "show route" | grep "192.168"

BIRD Commandes utiles 1

Lister les sessions
show protocols

details d'une session
show protocols all <protocol_name>

voir les routes [d'une table]
show route [table XXX]

voir le nombres de routes
show route [table XXX] count

voir les routes apprises via un protocole (BGP/Pipe)
show route [table XXX] protocol <protocol_name>

voir les routes annoncées
show route [table XXX] export <protocol_name>

BIRD Commandes utiles 2

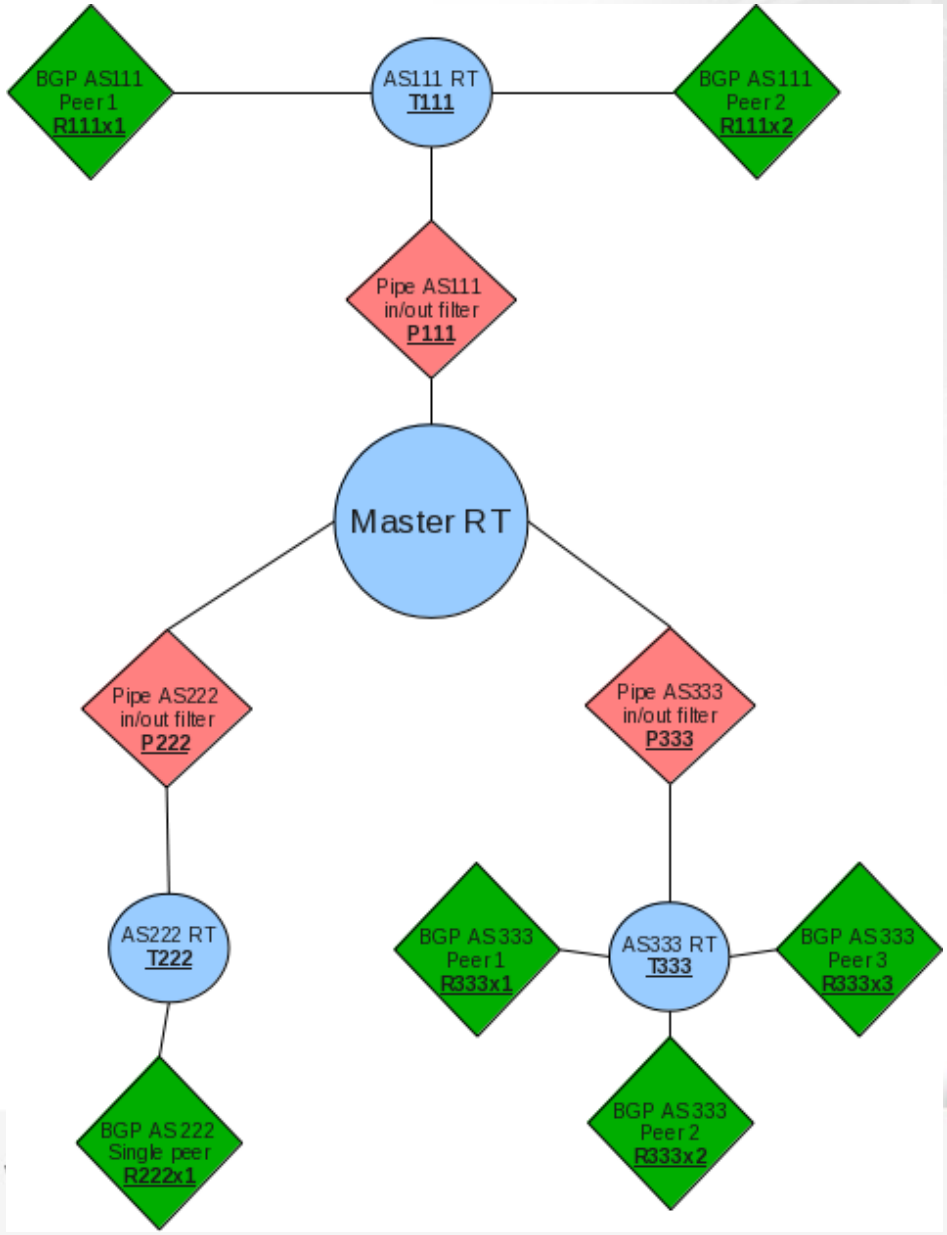
clearer soft une session
reload <protocol_name>

clearer hard une session
restart <protocol_name>

shutdown une session
disable <protocol_name>

démarrer une session
enable <protocol_name>

Route server with community based filtering and multiple RIBs



Conventions de nommage

Les sessions BGP utilisent cette convention de nommage **R<AS#>_<#>**
R43100_2 correspond a la 2eme session de l'AS43100

Les routes de chaque AS sont regroupées dans une table **T<AS#>_nom**
par ex: **T43100_Iyonix**. Les routes de la session BGP **R43100_1** sont importées dans cette table sans filtrage.

Le protocole **Pipe** permet d'échanger les routes avec la table **master** qui contient toutes les routes apres filtrage.

La convention de nommage est **P<AS#>_nom**
par ex: **P43100_Iyonix**.

Lors de la déclaration **T43100_Iyonix** est la "**peer table**", Les routes sont **exportées** vers la **peer table** et **importées** de la **table master**.

BIRD Commandes utiles 3

voir les routes avec une communauté
show route where (0,51706) ~ bgp_community

voir les routes avec des communautés
show route where bgp_community ~ [(51706,0..65536)]

voir les routes avec une communauté mais pas 8220 en origine
show route where bgp_community ~ [(0,51706)] &&
bgp_path.first != 8220

voir les routes égales ou plus spécifiques
show route [table XXX] where net ~ 1.2.0.0/16

détail de la route/mask exacte
show route [table XXX] 1.2.0.0/16 all

Commandes utiles Exemple

voir les routes reçues de R43100_2 (avant filtrage) l'ajout de "table Txxxx" est facultatif

show route table T43100_lyonix protocol R43100_2

voir les routes apprises de R43100_2 (apres filtrage, il est important de spécifier la table master!!)

show route table master protocol R43100_2

voir les routes annoncées a R43100_2

show route table T43100_lyonix export R43100_2

voir les routes annoncée a R43100_2 provenant de R15169_1

show route table T43100_lyonix protocol R15169_1

Route Server

Configuration BIRD

Conf IPv4 : Paramètres spécifiques

/etc/bird/bird.conf

```
log "/var/log/bird.log" all;
log stderr all;

# route-server specific
define RS_ID = 1;
router id 37.49.236.250;
listen bgp address 37.49.236.250 port 179;

## This function excludes weird networks
function is_martian()
prefix set martians;
{
  # https://en.wikipedia.org/wiki/Martian_packet
  # https://www.team-cymru.org/bogon-reference-http.html
  martians = [ 0.0.0.0/8+, 10.0.0.0/8+, ..... ];

  # Avoid 0.0.0.0/X
  if net.ip = 0.0.0.0 then return true;

  # Avoid too short and too long prefixes
  if (net.len < 8) || (net.len > 24) then return true;

  # Avoid RFC1918 networks
  if net ~ martians then return true;
  #
  return false;
}
```

Conf IPv6 : Paramètres spécifiques

/etc/bird/bird6.conf

```
log "/var/log/bird6.log" all;
log stderr all;

# route-server specific
define RS_ID = 1;
router id 37.49.236.250;
listen bgp address 2001:7f8:54::250 port 179;

## This function excludes weird networks
function is_martian()
prefix set martians;
{
  # https://en.wikipedia.org/wiki/Martian_packet#IPv6
  # http://www.space.net/~gert/RIPE/ipv6-filters.html
  martians = [ ::/8+, 2001::/33+, 2001:10::/28+, 2001:db8::/32+, 2002::/17+, 3ffe::/16+, fc00::/7+,
fe00::/9+, fe80::/10+, fec0::/10+, ff00::/8+ ];

  # Avoid ::/X
  if net.ip = :: then return true;

  # Avoid too short and too long prefixes
  if (net.len < 19) || (net.len > 48) then return true;

  # Avoid martians
  if net ~ martians then return true;

  return false;
}
```

Template conf

commun bird et bird6

```
# Templates for BGP and pipes protocols
template bgp PEERS {
  debug { events, states };
  local as MyASN;
  import all;
  export all;
  interpret communities off;
  rs client;
  passive on;
  add paths tx;
}

template pipe PIPES {
  table master;
  mode transparent;
}

table master sorted;
```

Conf d'un membre IPv4

bird uniquement

```
### SAINT GOBAIN - AS 49690 PEER 37.49.236.26 ###
table T49690_saintgobain sorted;

protocol pipe P49690_saintgobain from PIPES {
  peer table T49690_saintgobain;
  import where bgp_in(IX_FRANCEIX_PAR, 49690);
  export where bgp_out(IX_FRANCEIX_PAR, 49690);
}

protocol bgp R49690_1 from PEERS {
  neighbor 37.49.236.26 as 49690;
  import limit 10;
  table T49690_saintgobain;
}
```

Conf d'un membre IPv6

bird6 uniquement

```
### SAINT GOBAIN - AS 49690 PEER 2001:7f8:54::26 ###
table T49690_saintgobain sorted;

protocol pipe P49690_saintgobain from PIPES {
  peer table T49690_saintgobain;
  import where bgp_in(IX_FRANCEIX_PAR, 49690);
  export where bgp_out(IX_FRANCEIX_PAR, 49690);
}

protocol bgp R49690_1 from PEERS {
  neighbor 2001:7f8:54::26 as 49690;
  import limit 20;
  table T49690_saintgobain;
}
```

Conf : def de base

commun bird et bird6

```
### Configure logging and timeformat
timeformat base     iso long;
timeformat log      iso long;
timeformat protocol iso long;
timeformat route    iso long;

### Define base numbers used for BGP communities
define MyASN = 51706;
define RS_BASE = 64600;
define IX_BASE = 64640;
define IX_FRANCEIX_MRS = 9;
define IX_FRANCEIX_PAR = 10;
define PREPEND_BASE = 65100;
define MED_BASE = 65200;

### You don't need any other protocol such as kernel or direct
protocol device { }


### 32 bits ASN -> 16 bits ASN mapping
function map_to_16b(int peeras)
{
  if (peeras = 197422) then return 64701;  # tetaneutral
  if (peeras = 196689) then return 64702;  # digicube
  if (peeras = 197133) then return 64703;  # mediactive
  if (peeras = 197981) then return 64704;  # intercloud
……….
}
```

Conf : Filtrage IN

commun bird et bird6

```
### Basic checks, add communities and accept routes

function bgp_in(int IX_ID; int peeras)
{
  sanitize_routes_from(peeras);
  add_communities(IX_ID);
  return true;
}
```

Conf : Nettoyage et checks

```
### Next Hop must be the IP from the source of the annouce and left hand ASN must be peer ASN
function sanitize_routes_from(int peeras)
{
  # these communities can only be set by the route-servers
  # reject the route if community is set by member
  if (bgp_community ~ [ (MyASN,64495..64699) ]) || (bgp_community ~ [ (MyASN,64800..65535) ]) ||
(bgp_ext_community ~ [ (rt,MyASN,64495..64699) ]) || (bgp_ext_community ~ [ (rt,MyASN,
64800..65535) ]) then {
    print "Net: ", net, " from IP ", from, " with AS ", bgp_path.first, " rejected because of forbidden
community";
    reject "prefix rejected because of forbidden community";
  }
  if ( peeras != bgp_path.first ) then {
    print "Invalid first ASN on net: ", net, " from IP ", from, " with AS ", bgp_path.first, " instead of ", peeras;
    reject "prefix rejected because of invalid first ASN";
  }
  if ( from != bgp_next_hop ) then {
    print "Invalid next-hop on net: ", net, " with next-hop ", bgp_next_hop, " from IP ", from, " from AS
",bgp_path.first;
    reject "prefix rejected because of invalid next-hop";
  }
  if ( is_martian() ) then {
    print "martian network: ", net, " from IP ", from, " from AS ",bgp_path.first;
    reject "prefix rejected because of martian prefix";
  }

}
```

Conf : Ajout communautés

commun bird et bird6

```
function add_communities(int IX_ID)
{
  # if you specify a resticted list of peers, I'm enforcing the fact you don't want to be
announce to everybody
  if (((bgp_community ~ [ (MyASN,*) ]) || bgp_ext_community ~ [ (rt,MyASN,*) ]) && !
is_community(MyASN,MyASN)) then {
    bgp_community.add((0,MyASN));
  }

  #It's the last match in bgp_out function so I add it anyway
  bgp_community.add((MyASN,MyASN));

  # add route-server identifier
  bgp_community.add( (MyASN,RS_BASE+RS_ID));
  # add IX identifier
  bgp_community.add( (MyASN,IX_BASE+IX_ID));

  return true;
}
```

Conf : Filtrage OUT

commun bird et bird6

```
### BGP output filter (based on communities)
function bgp_out(int IX_ID; int peeras)
{
  # Announce only BGP routes
  if ! (source = RTS_BGP) then return false;

  # - This is the part regarding the IXP - #
    # Reject routes with 0:peer-as where peer-as is a mapped IX_ID (used for "Do not annonce to this IXP")
    if is_community(0, IX_BASE+IX_ID) then return false;

    # Set MED and Prepend AS for "all members of this IXP"
    tune_attributes(IX_BASE+IX_ID);

    # Partner's IXP members
    if ! (IX_ID = IX_FRANCEIX_PAR || IX_ID = IX_FRANCEIX_MRS) then {
      # only send routes from FranceIX Paris members
      if ! ((MyASN, IX_BASE+IX_FRANCEIX_PAR) ~ bgp_community) then return false;
    }

  # - This is the part regarding peer_as - #
    # Do not advertise a route with 0:peer_as community
    if is_community(0,peeras) then return false;

    # Advertise a route with MyASN:peer_as community
    if is_community(MyASN,peeras) then {
      tune_attributes(peeras);
      clean_communities();
      return true;
    }

    # Do not advertise route with 0:MyASN community
    if is_community(0,MyASN) then return false;

    # Advertise a route with MyASN:MyASN community
    if is_community(MyASN,MyASN) then {
      tune_attributes(peeras);
      clean_communities();
      return true;
    }
  return false;
}
```

Conf : check communauté

commun bird et bird6

```
### Chek if the community or ext_community is present

function is_community(int left; int right)
int right16b;
{
  if right > 65535 then {
    #map 32b ASN to private AS for legacy compatibilty
    right16b = map_to_16b(right);

    #Check of the extended community for 32b ASN
    if (rt,left,right) ~ bgp_ext_community then return true;

    #Check of the mapped 32b ASN into an 16b ASN
    if ((left,right16b) ~ bgp_community) || ((rt,left,right16b) ~ bgp_ext_community) then return true;

  } else {
    #Check of the extended (and not ext) community for 16b ASN
    if ((left,right) ~ bgp_community) || ((rt,left,right) ~ bgp_ext_community) then return true;
  }
  return false;
}
```

Conf : Tuning attibuts BGP

commun bird et bird6

```
###Set MED, Prepend AS
function tune_attributes(int peeras)
{
  # AS-Path prepending communities
  if is_community(PREPEND_BASE+1,peeras) then {
    bgp_path.prepend(bgp_path.first);
  }
  if is_community(PREPEND_BASE+2,peeras) then {
    bgp_path.prepend(bgp_path.first);
    bgp_path.prepend(bgp_path.first);
  }
  if is_community(PREPEND_BASE+3,peeras) then {
    bgp_path.prepend(bgp_path.first);
    bgp_path.prepend(bgp_path.first);
    bgp_path.prepend(bgp_path.first);
  }

  # MED communities
  if is_community(MED_BASE+1,peeras) then bgp_med =  50;
  if is_community(MED_BASE+2,peeras) then bgp_med = 100;
  if is_community(MED_BASE+3,peeras) then bgp_med = 200;
}
```

Conf : Clean des communautés

commun bird et bird6

```
### Remove unwanted IXP communities
function clean_communities()
{
  # Remove IXP related communities
  bgp_community.delete([(0,*)]);
  bgp_community.delete([(MyASN,0..64512)]);
  #remove private AS but keep Well-known communities
  bgp_community.delete([(64512..65534,*),(65535,0..65280)]);

  bgp_ext_community.delete([(rt,0,*)]);
  bgp_ext_community.delete([(rt,MyASN,0..64512)]);
  #I can't make range, but I don't think anybody will try to announce it anyway...
  bgp_ext_community.delete([(rt,65101,*)]);
  bgp_ext_community.delete([(rt,65102,*)]);
  bgp_ext_community.delete([(rt,65103,*)]);
  bgp_ext_community.delete([(rt,65201,*)]);
  bgp_ext_community.delete([(rt,65202,*)]);
  bgp_ext_community.delete([(rt,65203,*)]);
}
```

Plus de DOC !

Documentation :
http://bird.network.cz/?get_doc&f=bird.html

Doc sur les Pipes :
http://bird.network.cz/?get_doc&f=bird-6.html#ss6.7

Tres bon exemples :
https://gitlab.labs.nic.cz/labs/bird/wikis/Examples

Notre config est basée sur un setup multi-RIB :
https://gitlab.labs.nic.cz/labs/bird/wikis/Route_server_with_community_based_filtering_and_multiple_RIBs

Références

Euro-IX 27 : Route Server Policies @ IXPs
https://euro-ix.net/m/uploads/2015/10/27/e-BH-20150921-Euro-IX-Route-Server-Filtering-at-IXPs.pdf

RIPE 70 : IRR Lockdown
https://ripe70.ripe.net/wp-content/uploads/presentations/52-RIPE70_jobsnijders_irrlockdown.pdf

AMS-IX Falcon class Route Servers
https://ams-ix.net/technical/specifications-descriptions/ams-ix-route-servers/falcon-class-route-servers

Euro-IX 27 : Peering Observations 2007 vs. 2015
https://euro-ix.net/m/uploads/2015/10/23/27th-euro-ix-peering-observations.pdf

NANOG 51 : Route Servers, Mergers, Features and More
https://www.nanog.org/meetings/nanog51/presentations/Tuesday/Malayter-Router%20Server%20Presentation%204.pdf

and voila !