

BIRD, socio del IXP

Santiago Aggio
Conicet, UTN-FRBB

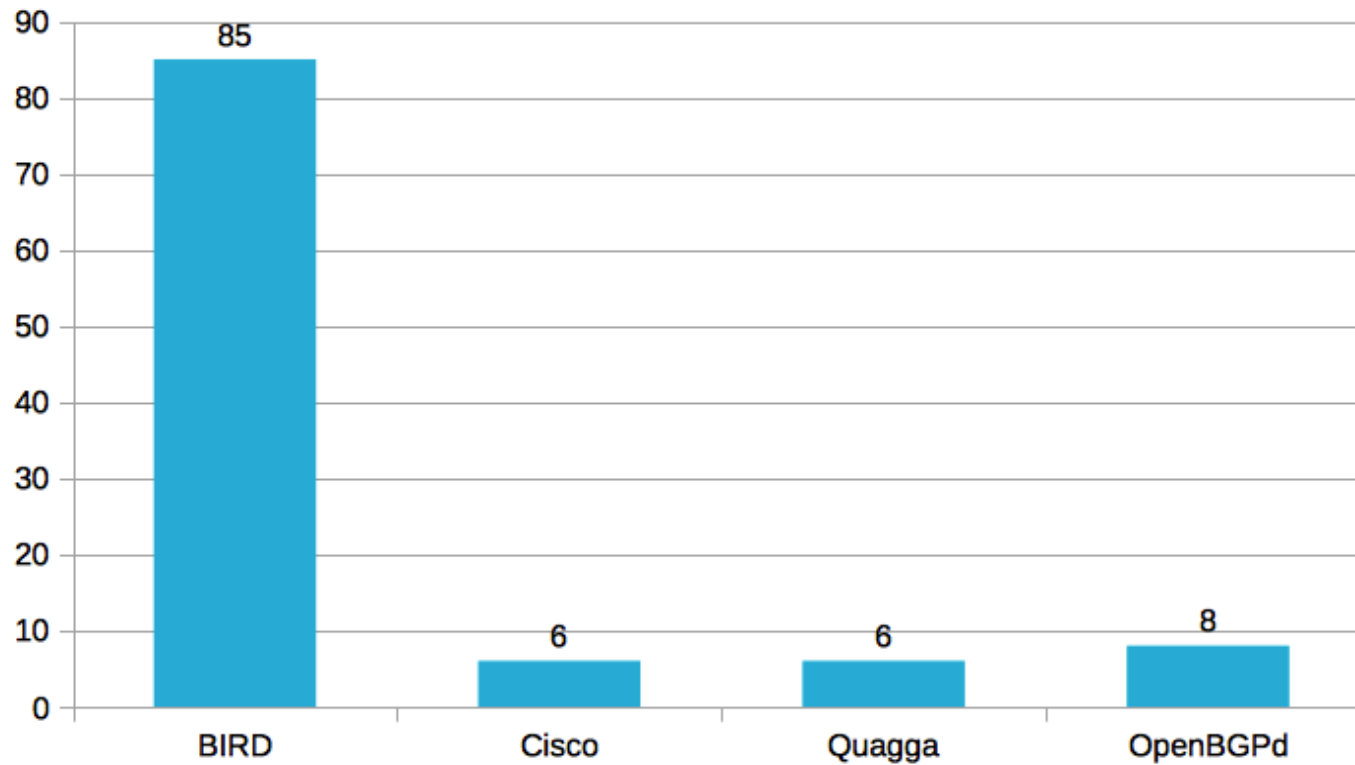
Peering Forum

LACNIC 29, Ciudad de Panamá
30 de Abril de 2018

Internet Exchange Point - IXP

- Tensión segurizada, refrigeración continua, espacio en rack.
- Switch
 - Cobre, Fibra Optica, bocas de 1Ge, 10Ge, 40Ge, etc
- Routers de los miembro del IXP
- CDN's, DNS root servers, DNS anycast
- Máquina de Monitoreo (física o virtualizada)
 - SNMP, NetFlow, Sflow, BGPmon, etc.
- **Route Server (2 para redundancia)**

BIRD



Fuente: EURO-IX Internet Exchange Points – 2016 Report

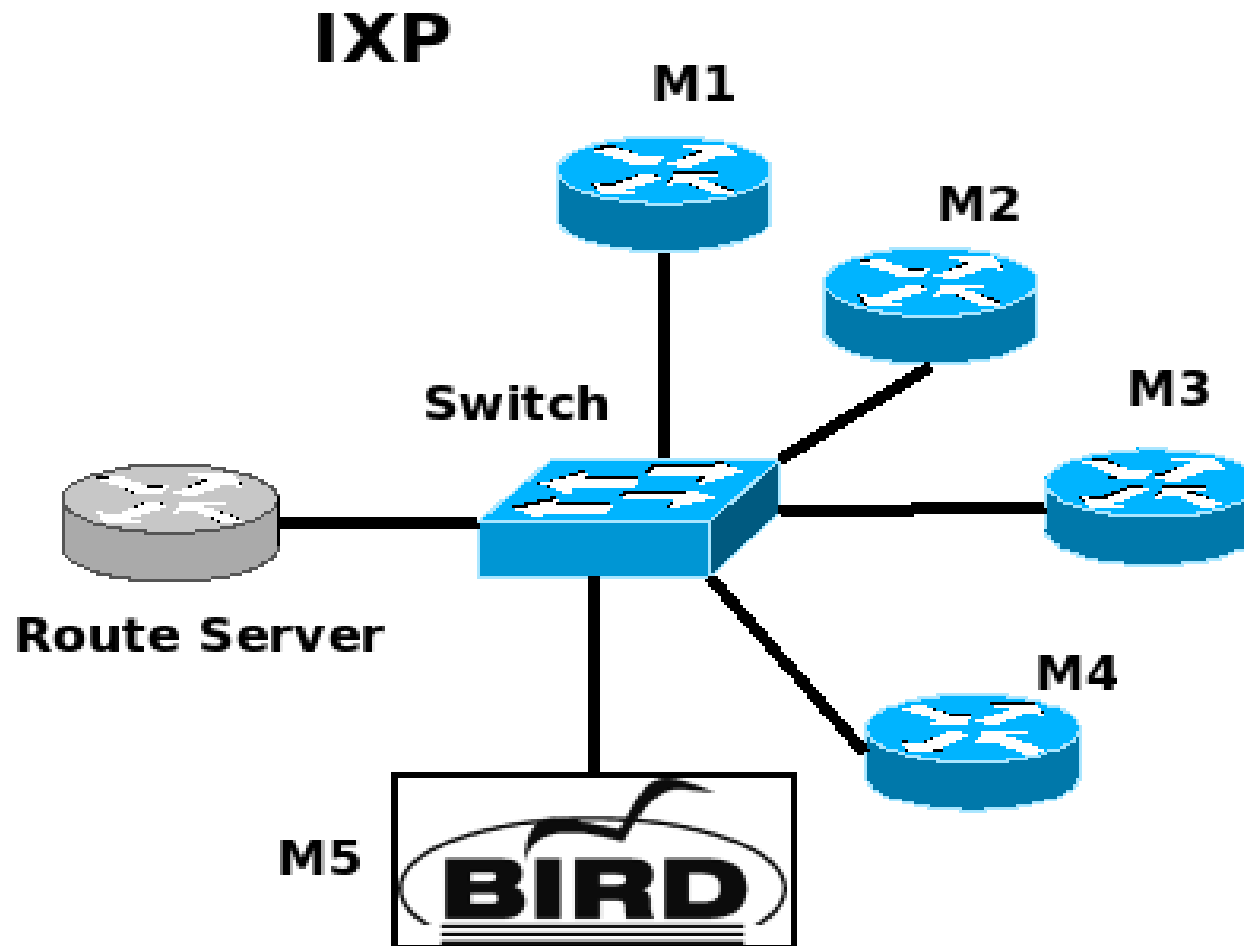
BIRD: ejemplos de uso

1) Router de Borde en un IXP

2) DNS Anycast

3) Route Server + RPKI Validator
(Repositorio)

1) Router de Borde en IXP



1) Router de Borde en IXP

- Debian 9
- Configuración Básica BIRD versión 1.6
- BGP externo e interno
- Filtros básicos (import y export)
- Demonio IPv4 e IPv6
- Forwarding IPv4 e IPv6
- Firewall: iptables e ip6tables

1) Router de Borde en IXP

- Configuración IPv4

```
protocol static static_bgp {  
    route 192.168.192.0/20 via 192.168.210.4;  
    route 192.168.208.0/21 via 192.168.210.4;  
}
```

```
protocol bgp ixp {  
    description "IXP";  
    local as 65000;  
    neighbor 192.0.2.254 as 64254;  
    import filter bgp_in_ixp;  
    export where proto = "static_bgp";  
}
```

1) Router de Borde en IXP

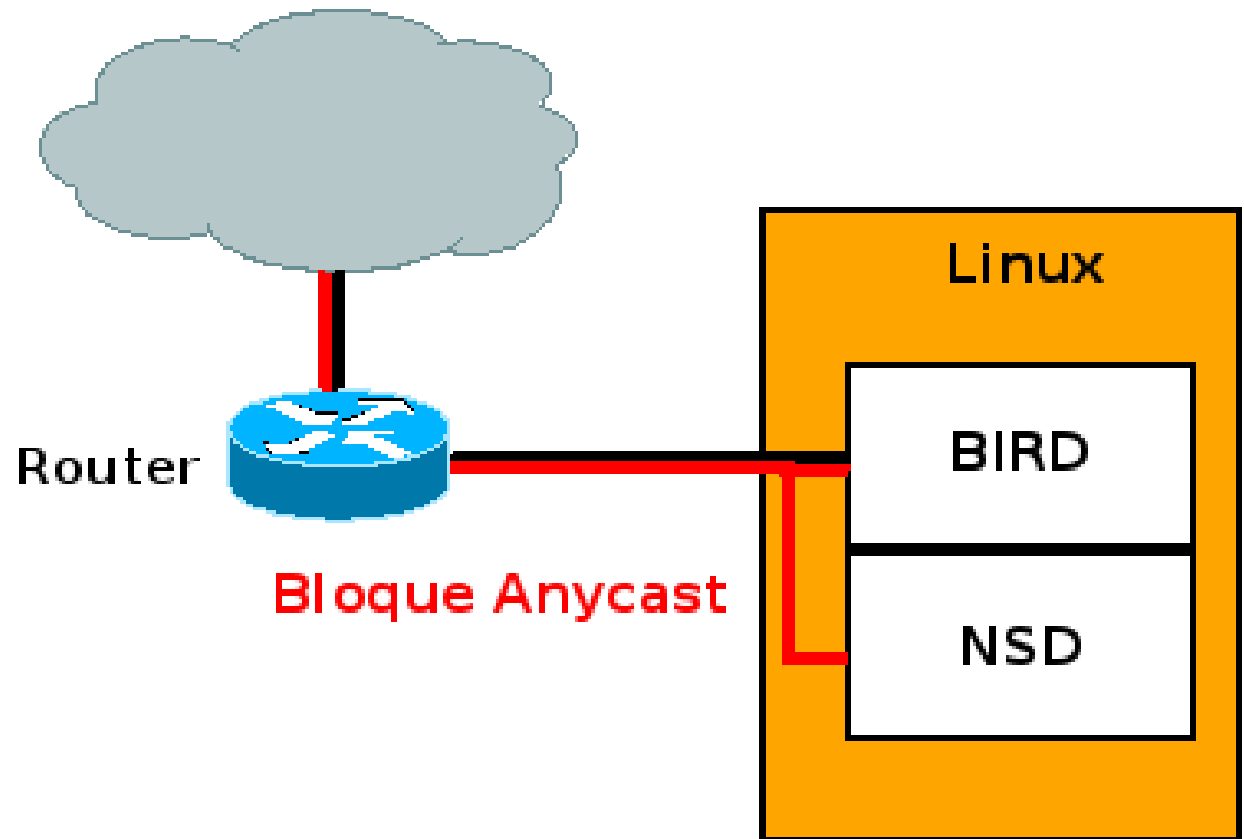
- Configuración IPv6

```
protocol static static_bgp_v6 {  
    route 2001:db8:90::/48 via 2001:db8:90:330::4;  
}
```

```
protocol bgp ixp_v6 {  
    description "IXP_v6";  
    local as 65000;  
    neighbor 2001:db8:6009::254 as 64254;  
    import filter bgp_v6_in_ixp;  
    export where proto = "static_bgp_v6";  
}
```


2) DNS Anycast

- Ubuntu 16.04.1 LTS
- NSD 4.1.7
- BIRD 1.6



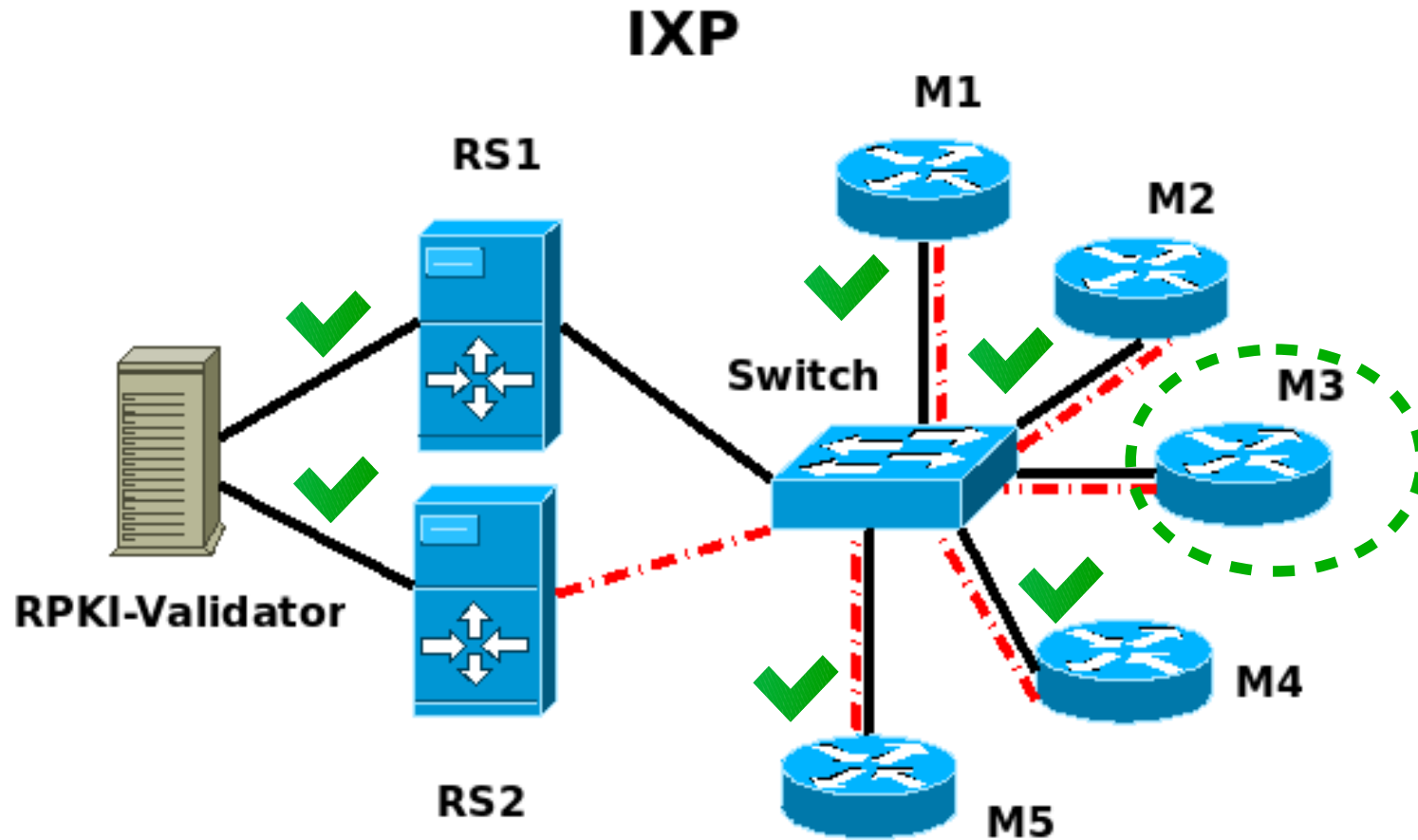
2) DNS Anycast

```
# Configure logging
log syslog { debug, trace, info, remote, warning,
error, auth, fatal, bug };
log stderr all;
# Turn on global debugging of all protocols
debug protocols all;
router id 192.0.2.196;
protocol kernel {
    scan time 60;
    import none;
# export all;
}
protocol device {
    scan time 60;
}
```

```
protocol bgp dns_anycast {
    description "DNS anycast BGP";
    source address 192.0.2.196;
    local as ;
    Neighbor 192.0.2.193 as 64512;
    multihop 2;
    graceful restart;
    import none;
    export where proto = "anycast_prefix";
}

protocol static anycast_prefix {
route 203.0.113.0/24 via 192.0.2.196;
}
```

3) Route Server + RPKI Validator



3) Route Server + RPKI Validator

- RS pueden mejorar la estabilidad del IXP validando el origen de la rutas anunciadas por los miembros
- No requiere que cada miembro corra una instancia local del validador RPKI
- Facilidad de implementación desde un punto central, extendiendo el uso de RPKI

3) Route Server + RPKI-Validador

- (a) RS valida el origen de las rutas y filtra las rutas inválidas en sus anuncios al resto de los miembros

- (b) RS anuncia el estado de la validación (Valid, Invalid, NotFound) en una Large Community y cada miembro aplica su política (Local_Pref o reject)

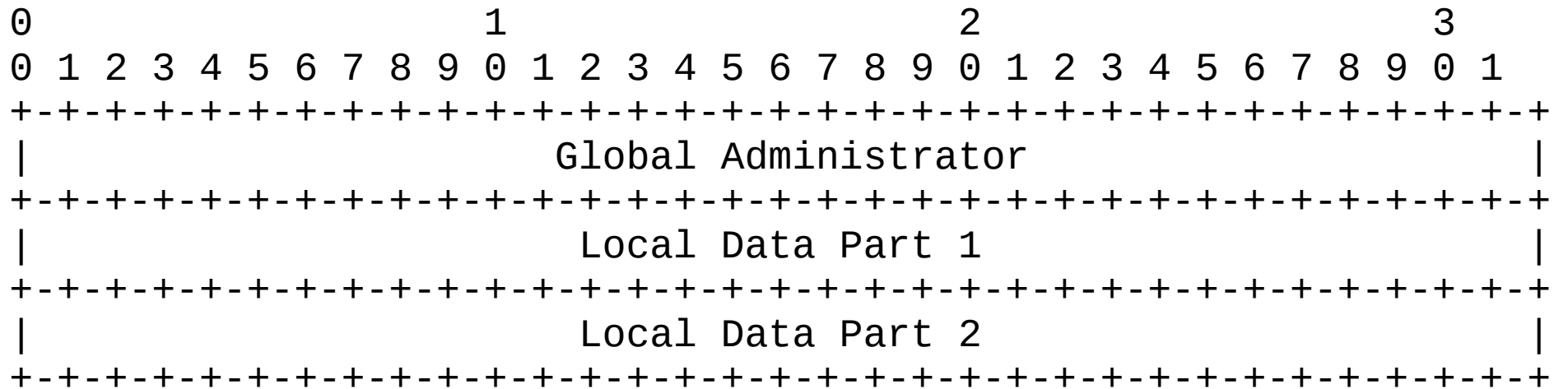
- (c) RS podría implementar en un futuro Origin Validation State Extended Community (IETF draft)

(a) RS filtra ROAs inválidos

```
filter peer_in_v4 {
    if (roa_check(r4, net, bgp_path.last) = ROA_INVALID) then {
        print "Ignore invalid ROA ", net, " for ASN ",bgp_path.last;
        reject;
    }
    accept;
}
. . . . .

protocol bgp miembros_ixp {
    . . . . .
    ipv4 {
        import filter peer_in_v4;
        export all;
        . . . . .
    }
}
```

(b) BGP Large Communities Attribute



- RFC 8092 (Febrero 2017)
- 96-bits → "32-bits:32-bits:32-bits"
- Representación Canónica: mi_AS:Acción:Valor

(b) BGP Large Communities Attribute

- Política propuesta:

IXP_AS:Action:Valor

IXP_AS → 64254

Action → **ROA Status == 8282** (Ej: port RPKI-Validator)

Valor → **0 = Valid, 1 = Not found, 2 = Invalid**

Ej: **64254:8282:1** → IXP_AS:ROA Status:Not found

- En IXP BIRD:

bgp_large_community.add((64254,8282,0)) → Válido

- EN IXP Miembro: ROA Not found o Inválido

If ((**64254,8282,1**) ~ bgp_large_community) then { bgp_local_pref = 50; }

If ((**64254,8282,2**) ~ bgp_large_community) then { reject; }

(c) IETF 101

(Londres, Marzo 2018)

Sidrops Group: *draft-ietf-sidrops-route-server-rpki-light*

"Signaling Prefix Origin Validation Results from an RPKI Origin Validating BGP Speaker to BGP Peers"

Allow for 3 modes of operation for validating BGP speaker:

1. Tag prefixes with their ROA validity status, and advertise them.
2. Drop prefixes with ROA status "Invalid". Tag the remaining "Unknown" AKA "NotFound" and "Valid" routes, and advertise them.
3. Drop prefixes with ROA status "Invalid" or "Unknown". Tag the remaining "Valid" routes, and advertise them.

Repositorio

- BIRD 2.0.1 → Route Server
- RIPE NCC RPKI Validator 2.4
- Docker 18.03.0-ce
 - mapeo de puertos externos en docker host
 - Contenedor corriendo el validador (puertos 8080,8282)
 - Contenedor BIRD Route Server BGP (puerto 179)

<https://github.com/sancolo/route-server-rpki-validator>

Conclusiones

- BIRD + RPKI Validator son paquetes de código abierto y corren bajo Linux
- BIRD es ampliamente usado en IXP con más de 100 miembros
- BIRD versión 2.0.x presenta nuevas funcionalidades aplicables en un IXP
- Route Server puede validar ROAs y anunciar prefijos marcados con una comunidad Large o Extendida
- Repositorio de guía para experimentar e implementar un route server con validador de origen de rutas.

Referencias

- The BIRD Internet Routing Daemon - <http://bird.network.cz>
- RIPE NCC RPKI Validator.
 - <https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources>
- Build your own Anycast Network in Nine Steps
 - https://labs.ripe.net/Members/samir_jafferli/build-your-own-anycast-network-in-nine-steps
- BGP Large Community Attribute
 - <https://tools.ietf.org/html/rfc8092>
- Deploying BGP Large Communities
 - http://largebgpcommunities.net/presentations/Job_Snijders-use-of-large-bgp-communities.pdf
- Signaling Prefix Origin Validation Results from an RPKI Origin Validating BGP Speaker to BGP Peers -
 - draft-ietf-sidrops-validating-bgp-speaker-02
- Un resumen de IETF101 en Londres. LACNIC LABS
 - <https://labs.lacnic.net/resumen-IETF101>
- <https://datatracker.ietf.org/meeting/101/materials/slides-101-sidrops-signaling-prefix-origin-validation-results-from-an-rpki-origin-validating-bgp-speaker-to-bgp-peers-00>

¿Preguntas?

Muchas Gracias

slaggio@criba.edu.ar