RPSLng

Routing Policy Specification Language - Next Generation

Copy ... Rights

- This slide set is the ownership of the 6DISS project via its partners
- The Powerpoint version of this material may be reused and modified only with written authorization
- Using part of this material must mention 6DISS courtesy
- PDF files are available from www.6diss.org
- Looking for a contact ?
- Mail to : martin.potts@martel-consulting.ch
- Or bernard.tuy@renater.fr

Contributions

- Main authors
  - Carlos Friaças, FCCN, Portugal
  - Miguel Baptista, FCCN, Portugal
- Contributors
  - Paulo Ferreira, FCCN, Portugal
  - Mónica Domingues, FCCN, Portugal
- Special Thanks for RPSLng work
  - João Damas, ISC, Netherlands
  - Gabriella Paolini, 6Net, GARR, Italy
  - Simon Leinen, 6Net, Switch, Switzerland
  - Dimitrios Kalogeras, 6Net, GRNET, Greece

Prerequisites

- You must have followed previously the modules:
  - 010-IPv6 Introduction
  - 020-IPv6 Protocol
  - 030-IPv6 Addressing
  - 100-Routing Protocols

Agenda

- Routing Policy
- RPSL
- RPSLng
- Example
- Conclusion

Routing Policy

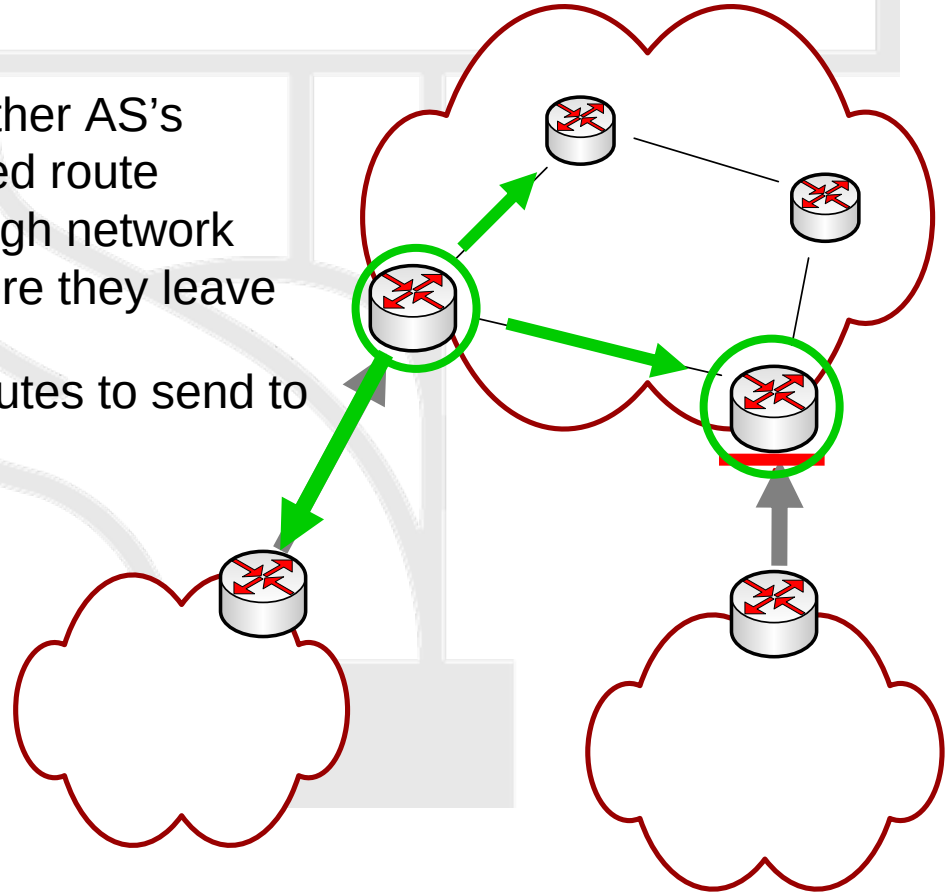
- What is routing policy?
  - Public description of the relationship between BGP peers
  - Routing policies enable route classification for importing and exporting routes
  - The goal of routing policies is to control traffic flow

Routing Policy

- Why define a Routing Policy
  - Documentation
    - Recreate your policy in case of loss of hardware/administrators
  - Allows automatic generation of router configurations
  - Provides routing security
  - Troubleshooting

Routing Policy - Example

- Reflects the As' goals
  - Which routes to accept from other AS's
  - How to manipulate the accepted route
  - How to propagate routes through network
  - How to manipulate routes before they leave the AS
  - Which routes to send which routes to send to another AS



Routing Policy

- Each Autonomous System has its own routing policy towards other Networks
- Each policy affects the way the global network behaves
- Which means:
  - It's very useful to know external policies
  - A place to publish them is needed
  - You can automatically configure border routers from that info, if you can rely on the quality of information

RPSL

- RPSL stands for Routing Policy Specification Language
- Replacement for the language known as RIPE-181
- A tool to describe Inter-Domain Policies, it affects:
  - People doing Local Internet Registry work
  - People dealing with border routers, BGP, ...
- It is used for Internet network management.
- It is NOT about Internal Routing!

RPSL

- Object oriented language
  - So … it has classes used to defined the various objects
- Uses RIR database style (whois) objects.
  - Each Object is a list of "attribute-value" pairs displayed in plain text.
    - person, maintainer, role
    - route
    - as-set, route-set
    - ...

Person Object - Example

person: Miguel Baptista
address: Example street
Lisbon, Portugal
phone: +351 123 456 789
e-mail: miguel.bap@example.org
nic-hdl: MB10-TEST
mnt-by: EXAMPLE-MNT
remarks: **********
remarks: This object is only an example!
remarks: **********
changed: carlos.friaças@example.org 20060228
source: TEST

RPSLng is...

RPSL ***next generation***

- Yet another easy thing to have in place
  - one more item in the check-list ;)
- Yet another tool to help IPv6 development in an «orderly» fashion;
- Yet another way of showing people IPv6 is not that much complex than IPv4.

RFC4012

- Backward Compatibility
- Changes:
  - New dictionary attribute – AFI
  - New predifined dictionary type
  - New protocol dictionary specification
  - New policy attributes
  - New route6 class
  - New attribute in route-set class
  - New attribute in filter-set class
  - New attribute in peering-set class
  - New attribute in inet-rtr class
  - New attribute in rtr-set class

RPSL and RPSLng Some Differences

	IPv4	**IPv6**
Networks	inetnum	inet6num
Routes	route	route6
Policies *(aut-num)*	import export	mp-import mp-export

Evolution…

- RIPE/NCC and APNIC already have a RPSLng compliant Whois service.

  Other RIRs will follow.

- LIR admins are rewriting their own routing policies, to include:
  - IPv4 Unicast;
  - IPv4 Multicast;
  - IPv6 Unicast;
  - IPv6 Multicast (very, very few)

Objects - Examples #1

- Route6

```
route6:      2001:0760::/32
descr:       GARR-IPv6
origin:      AS137
mnt-by:      GARR-LIR …
```

- Peering-set

```
peering-set:    prng-ebgp-peers
descr:          TopneT IPv6 ebgp peers
...
mp-peering:  AS12533 2001:15A8:A:1:FFFF:FFFF:FFFF:2 at 2001:15A8:A:1:FFFF:FFFF:FFFF:3
...
mp-peering:     AS5609 3FFE:1001:1:F036::1 at 3FFE:1001:1:F036::2
...
mp-peering:     AS5602 2001:15A8:A:1:FFFF:FFFF:FFFF:5 at
     2001:15A8:A:1:FFFF:FFFF:FFFF:4
...
mp-peering:     AS6939 2001:470:1F01:FFFF::224 at 2001:470:1F01:FFFF::225
```

Objects - Examples #2

- **Aut-Num**

```
aut-num:     AS1853
as-name:     ACOnet
descr:       ACOnet Backbone
descr:       AT
remarks:     ================================
remarks:     #upstream: Sprint.net
import:      from AS1239 action pref=100; accept ANY
export:      to   AS1239 announce AS-ACONET AND AS-SANET
mp-import:     afi ipv6.unicast   from AS6175  accept   ANY
mp-export:     afi ipv6.unicast   to   AS6175  announce AS-ACONET-V6
remarks:     #upstream: GEANT.net
import:      from AS20965 action pref=100; accept ANY
export:      to   AS20965 announce AS-ACONET AND AS-UNREN AND AS-
    ACOSERV
mp-import:     afi ipv6.unicast   from AS20965 accept   ANY
mp-export:     afi ipv6.unicast   to   AS20965 announce AS-ACONET-V6
remarks:     ================================
...
```

Objects - Examples #3

- Inet-rtr

```
inet-rtr:      BR1.mucI.baycix.net
local-as:      AS12657
ifaddr:        212.72.95.1        masklen 32
interface:     2001:1578:0:FFFF::1 masklen 128
interface:     2001:1578:0:FF::1   masklen 112
peer:          BGP4 212.72.95.3          asno(AS12657)
peer:          BGP4 212.72.72.197        asno(AS29317)
mp-peer:       MPBGP 2001:1578:0:FFFF::2 asno(AS12657)
...
```

- Route-set

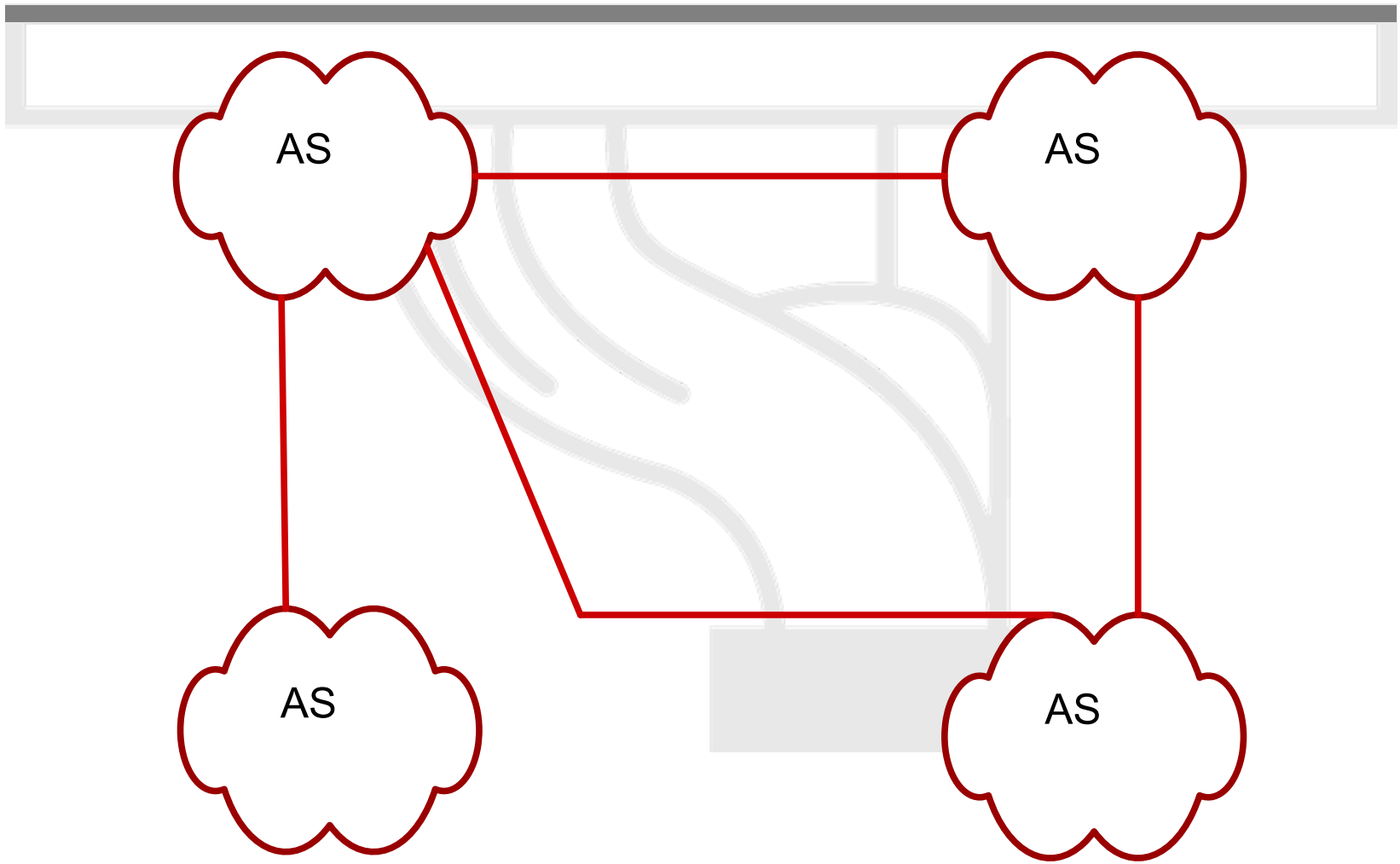
```
route-set:     AS29670:RS-IN-BERLIN
descr:         Individual Network Berlin e.V.
org:           ORG-INBE1-RIPE
mp-members:    192.109.21.0/24
mp-members:    217.197.80.0/20
mp-members:    2001:bf0:c000::/35
...
```

Objects - Examples #4

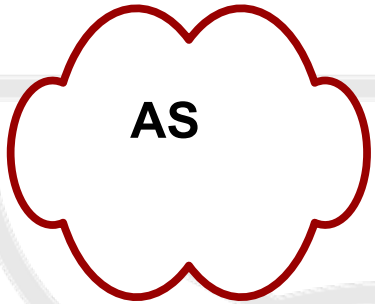
- Filter-set

```
filter-set:   AS12817:fltr-BOGONS
descr:        Generic IPv4/IPv6 Prefix & AS filter
mp-filter: { 10.0.0.0/8^+,
             127.0.0.0/8^+,
             169.254.0.0/16^+,
             192.168.0.0/16^+,
             0.0.0.0/0^25-32 }
           AND
           { 2001:db8::/32^+,
             0000::/8^+,
             fe00::/9^+,
             ff00::/8^+,
             0::/0^49-128 }
           AND
           <[AS64512-AS65534]>
...
```

Example

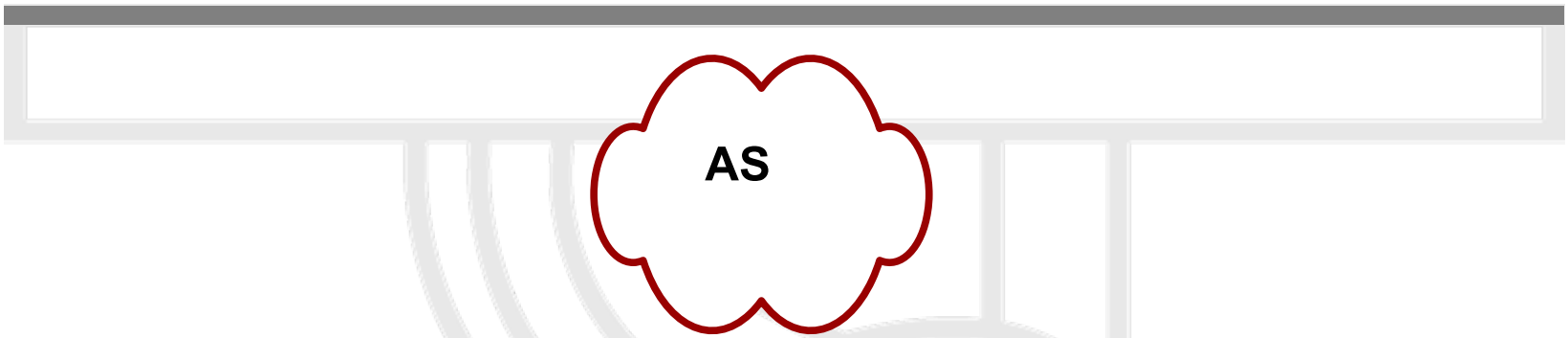


Example – AS *A* Policy



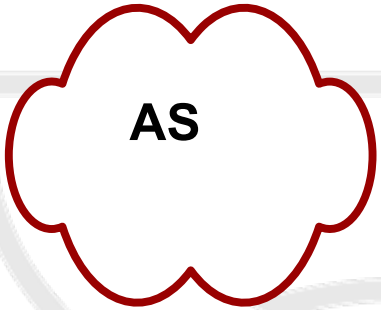
aut-num: AS *64600*
as-name: AS A
descr: This is AS A
mp-import: afi ipv4.unicast,ipv6.unicast from AS64700 action pref=106; accept ANY;
mp-export: afi ipv4.unicast,ipv6.unicast to AS64700 announce AS-A;

Example – AS *D* Policy



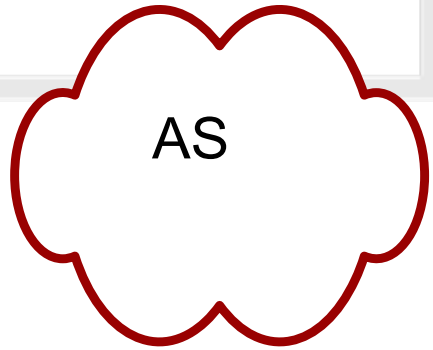
aut-num: AS64900
as-name: AS D
descr: This is AS D
mp-import: afi ipv4.unicast,ipv4.multicast,ipv6.unicast from AS64700 action pref=106; accept ANY;
mp-import: afi ipv6.multicast from AS64800 action pref=110; accept AS-C
mp-export: afi ipv4.unicast,ipv4.multicast,ipv6.unicast to AS64700 announce AS-D;
mp-export: afi ipv6.multicast to AS64800 announce AS-D

Example – AS *C* Policy



aut-num: AS64800
as-name: AS C
descr: AS C, This is AS C
import: from AS64700 action pref=106; accept ANY
mp-import: afi ipv4.multicast,ipv6.unicast from AS64700 action pref=106; accept ANY;
mp-import: afi ipv6.multicast from AS D action pref=110; accept AS D
export: to AS64700 announce AS C
mp-export: afi ipv4.multicast,ipv6.unicast to AS64700 announce AS C;
mp-export: afi ipv6.multicast to AS64900 announce AS C

Example – AS ***B*** Policy




```
aut-num:      AS64700
as-name:      AS B
descr:        AS B, This is AS B
import:       from AS64800    action pref=106; accept AS-C;
import:       from AS64900    action pref=106; accept AS-D;
import:       from AS64800    action pref=106; accept AS-A;
mp-import:    afi ipv4.multicast,ipv6.unicast from AS64800 action pref=106; accept
AS-C;
mp-import:    afi ipv4.multicast,ipv6.unicast from AS64900 action pref=106; accept
AS-D;
mp-import:    afi ipv6.unicast from AS64600 action pref=106; accept AS-A;
export:       to AS64800      announce ANY;
export:       to AS64900      announce ANY;
export:       to AS64600      announce ANY;
mp-export:    afi ipv4.multicast,ipv6.unicast to AS64800 announce ANY;
mp-export:    afi ipv4.multicast,ipv6.unicast to AS64900 announce ANY;
mp-export:    afi ipv6.unicast to AS64600 announce ANY
```

Conclusions

- RPSL is needed to coordinate global IPv4 routing policies. RPSLng is needed for the same purpose, but over IPv6

- It’s rather simple, and someone already dealing with RPSL will easily start to use RPSLng when starting to route IPv6 packets

Extra Slides

RPSLng Tools

- WHOISd
  - Free
  - ftp://ftp.ripe.net/ripe/dbase/software
  - Managed by RIPE
- IRRd
  - Free
  - http://www.irrd.net
  - Managed by MERIT

RPSLng Tools

- RIPE's RPSLng Registry
  - IPv4 address -> inetnum, route, inet-rtr
  - IPv6 address -> inet6num, route6, inet-rtr
  - Inverse queries for aut-num -> route + route6
  - Production Routing Policies
- IRRToolSet
  - Suite of policy analysis tools
  - Possible usage: Updating BGP routing configurations
  - Managed by ISC: ftp://ftp.isc.org/isc/IRRToolSet/