



EQUINIX

Route Servers, Mergers, Features, and More.

Integration of IXP's. Route Server challenges

Chris Malayter
cmalayter@equinix.com

What drives a RS integration?

- Two IX's merging into a single VLAN for mutual benefit
- A city with many IX's merging into a larger community based org.
- Two corporations merging existing fabrics

Choices Choices

- What platforms are being used?
 - Bird
 - OpenBGPD
 - Quagga
 - LINX Quagga
- What OS's are being used?
 - Linux
 - FreeBSD
 - OpenBSD
- Auto provisioning?
 - Almost a must for any IX of scale in this day and age.

More Choices

- Features
 - Community Support
 - Local Pref
 - Blackhole
 - AS4
 - V6
 - Filtering
 - AS-SET
 - Manual Update
 - AS-SET with the ability to update
 - None?
 - Multi-RIB

You've merged what next?

- Comprehensive analysis of features needed
 - Arguably the most important step in the process
 - IX membership will likely drive this part for you
 - What are your coders capable of coding for you in your time interval
- Comprehensive analysis of route server software
 - A route server is not a route server is not a route server
 - A 15 member IXP may be just fine with quagga, AMS-IX/LINX/DECIX/Equinix will not be
 - Diving into the messaging bus, threads, and queuing mechanisms of the RS software will help to give you an idea of scalability
- Analysis of what OS's are going to support your RS software
 - What can your ISS team support
 - Security considerations?
 - Requirements of the Route Server Software

- MD5
 - Do you want to support MD5? Opens a host of issues with MD5 key tracking, security, kernel versions, etc.
- BGP TTL Security
- Communities
 - Allow flexibility for your members
 - Can seriously complicate your config in a multi-rib environment
- Multiple RIB's
 - Allows an amazing amount of flexibility
 - Greatly complicates the config
 - Greatly complicates auto-provisioning
 - RIB per peer? Not all need it, but if you can and you have the RAM you may want to.
- IRR Filtering
 - V4's irr is completely unmaintained for a large AS-SETs. Leads to possible route leaks. Also amazingly large configs. (RAM, CPU, Startup Time)
 - V6 is annoying, not well populated, and can be annoying to aggregate.
 - Do you need to filter? Customers should be doing it.... 😊

- OpenBGPD – OpenBSD
 - OpenBGPD really doesn't work well on any platform but OpenBSD. There are ports, but a significant number of features will not work.
- MD5 issues
 - Legacy Linux kernels have many md5 bugs. Take your time and find a kernel version that works with your route server...it will save you many headaches down the line
 - Tracking Customer MD5's is a security issue in itself. The place you store the data should be highly secure to protect the "integrity" of the session.

General Trends

- OpenBGPd or BIRD seem to be preferred
 - Scalability, stability, and feature development tends to sway peoples choices
 - BIRD is under constant development with a very active development group
 - OpenBGPd is very stable but has less nobs and more bugs 😊
- OpenBSD, Linux and FreeBSD tend to be the OS's of choice
 - OpenBSD for OpenBGPd
 - Linux/FreeBSD for BIRD
 - More of a what a given IX is most experienced in
- Features are all over the board depending on the region

Membership Demands

- Customers and Members demand quite a bit
 - Most IX's have route servers because of strong member demand
 - Lacking route servers is a competitive disadvantage
 - The more automated (portal) you make your route servers, the happier customers tend to be
 - Features are heavily customer demand driven

Best Practices

- AS-SET filtering
- Prefix Filtering – North America
- Some community manipulation
- Auto-provisioning
- Multiple Ribs

OpenBGPd

- Three separate processes: parent, session engine, route decision engine; IPv4 and IPv6 handled by all
- IPv4 and IPv6 handled within the same config file
- Difficult to implement certain types of community handling (e.g. restoring the community marking passage through the MLPE server on outbound updates)
- Highly efficient community processing using macros
- Understood by our operations staff; user commands cover common situations
- The maintainers recommend that you run a version of OpenBSD compatible with OpenBGPd; failure to do so could require substantial patching in addition to recompilation from source

Bird

- One process handles all BGP functions; separate instances of this process for IPv4 and IPv6
- Separate config files for IPv4 and IPv6
- Easier to implement most types of community handling; some desired scripting language features missing (e.g. modulus)
- Less efficient community processing; many statements need to be duplicated
- A steep learning curve for our operations staff; many common situations require the knowledge of the scripting language
- More portable, but MD5 support needs to be verified against Linux kernels (especially before 2.6.32)

Test Results - OpenBGPD



Test Case

- Initial configuration (v4)
- Updates sent to MD5 peer (v4)
- Session limits of MD5 (v4)
- 200 sessions with increasing updates (v4)
- 200 sessions with long AS-paths (v4)
- 200 sessions with long community attribute lists (v4)
- 200 sessions with uninterpreted communities (v4)

Result

- Pass
- Fail (with standard filters)
- Pass
- Fail (with standard filters)
- Pass (with reduced filters)
- Fail (with standard filters)
- Pass (with reduced filters)

Test Case

- Initial configuration (v4)
- Updates sent to MD5 peer (v4)
- Session limits of MD5 (v4)
- 200 sessions with increasing updates (v4)
- 200 sessions with long AS-paths (v4)
- 200 sessions with long community attribute lists (v4)
- 200 sessions with uninterpreted communities (v4)

Result

- Pass
- Pass
- Pass
- Pending Review

- Pass
- Pass (with 160 sessions in 16 groups of 10)
- Pass (with 160 sessions, in 16 groups of 10)

Political Issues

- **Who owns the process**
 - Like all mergers/corporative ventures, someone has ownership of the process. Their preferences may weight heavily on the decision making process.
- **What platforms will they run on**
 - What OS's are currently supported may be your only option. That may limit your choices.
- **What IP space will you use**
 - As funny as it sounds, IP space is highly annoying. Customers are generally not happy with changing. Good rule of thumb, the less people you have to renumber, the better off you are!
- **Auto provisioning vs manual provisioning**
 - The more complicated the configuration, the more important auto-provisioning becomes. The easier you make it for your customers, the more people you're going to get on the route servers!

What's the end game?

- Beautiful new route servers?
- Old route servers that are reused?
- New software?
- New Processes?

Take your time!

- My best advice to anyone going through a route server integration is to take your time. The worst possible situation is to rush something that's not prime-time into production and alienate your members!
- Testing Testing Testing! The more testing you do, the better your outcome is going to be!

Questions?

