

RIPE



ExaBGP

How it got where it is,
its use cases,
what is next,
and how to contribute



What is ExaBGP

- Another BGP implementation
 - Started in 2009
 - BSD licensed, in python
 - two larges “re-factorisations”
 - used in production
- Not to transform a unix machine into a router
 - only a RIB
 - no FIB manipulation



Objectives

- Good code base
 - Async code with single thread limit
 - Good code structure is #1 objective
 - But now quite large
- A programmer friendly “BGP gateway”
 - Software Defined Networking using BGP
 - Text API, migrating toward a JSON API



Features - History

- First releases
 - Simple IPv4 route injector
 - announcing /32 from servers
 - cross data center failover
 - two servers active-standby
 - using different MED
 - Quickly added IPv6

Features - History

- Added support for Flow Spec
 - firewall rules sent using BGP
- SIGHUP to reload configuration
 - terrible API for external programs using it
 - (not working anymore currently - need looking at)

Features - History

- Using PIPE to control ExaBGP
 - http://ripe63.ripe.net/presentations/37-RIPE_63_-_Mangin_-_BGP.pdf
- Since many RFC / draft support added
 - <https://github.com/Exa-Networks/exabgp/wiki/RFC-Information>
 - over 20 RFC .. (partial or full support)
 - GRACEFUL RESTART, ASN4, ADD-PATH, MPLS, VPLS, ...
- Not only an injector, a full BGP listener



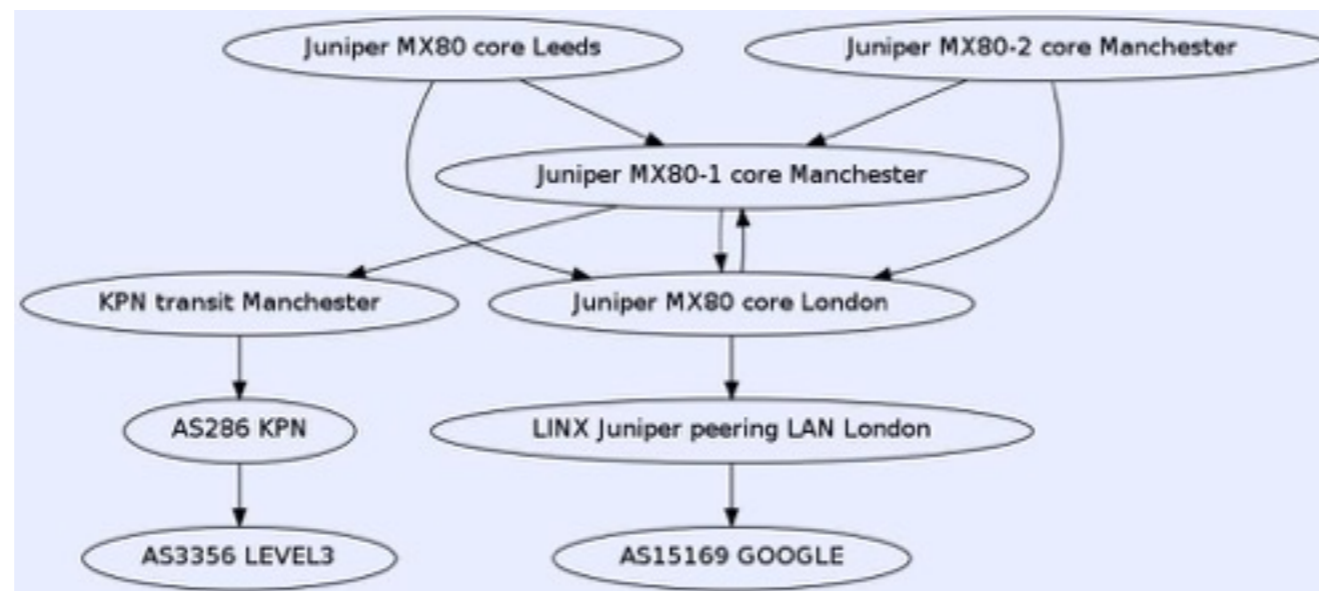
ExaBGP uses cases : network control

- Block “simple” (< Layer7) DDOS
 - started a “sibling” project ExaDDOS
- Controlled service routing / failover
 - <http://vincent.bernat.im/en/blog/2013-exabgp-highavailability.html>
 - <http://bits.shutterstock.com/2014/05/22/stop-buying-load-balancers-and-start-controlling-your-traffic-flow-with-software/>
- Traffic optimisation (using ADD-PATH)
 - BGP <— ExaBGP —> Application
 - Application <— ExaBGP —> BGP



ExaBGP uses cases : network analysis

- Visual representation of your routing
 - <https://github.com/dpiekacz/gixlg>



- RIPE experimental real-time RIS

- https://labs.ripe.net/Members/wouter_miltenburg/build-the-next-generation-ris-route-collectors

Development

- “Normal” small open-source project
 - Many “one-off” contributors over the years
 - A few regular bug reporters
- All development done on github
 - official tree
 - <https://github.com/Exa-Networks/exabgp>
 - development tree
 - <https://github.com/thomas-mangin/exabgp>



Development - what is next

- Stable branch
 - bug fix only branch
 - looking at changing this when 4.0 is released
- Several development branches
 - BMP v7 .. promised to Paolo for PMAACCT
 - Confederation
 - new configuration format (JSON friendly)

Development - what is next

- More code in master
 - Nearly complete EVPN support
 - ported from BAGPIPE
 - <https://github.com/Orange-OpenSource/bagpipe-bgp>
 - no configuration, untested but mostly complete
- More commitments
 - CLI
 - Route Server



Software quality

- A few end to end tests
 - #> ./dev/bin/selfcheck
 - read configuration, generate BGP update, parse generated BGP update, check back to square one
 - #> ./dev/bin/runtest
 - generate some BGP update, check raw update with previously stored data
 - ./dev/self/*
 - ExaBGP speaking with itself



Software quality

- Could be better
 - need more tests
 - unittest has not kept pace and is useless ATM
 - large deployments need lab'ing

- Documentation
 - ExaBGP #1 weakness
 - For many steep learning curve
 - Many examples but nothing more



Code Structure

- One main async loop /reactor
 - using co-operative multitasking (windows 3.1)
 - with python co-routines
- One main loop /reactor
 - One co-routine per peer
 - manager the peer state
 - One co-routing per TCP connection
 - send back data to the peer co-routine when a full message is received



Code Structure

- Organised using Inheritance
 - message
 - open (asn, holdtime, routerid, version)
 - capability (asn4, mp, negotiated, refresh, addpath, graceful, ms, operational, unknown)
 - update
 - attribute ...
 - nlri ...
 - keepalive, notification, refresh, operational, unknown, nop
- Use of class “registration” for decoding



Please help !

- ExaBGP is a “personal” project
 - competing for time with my work at Exa, LINX, IXLeeds, my family, and need to sleep ...
- I welcome contributors
 - I scan for fork and often merge and “fix” the code provided.
 - I am happy to spend some time per mail / IM / video conf to explain the code



Other relevant information

- Other BGP implementation
 - <https://github.com/Exa-Networks/exabgp/wiki/Other-OSS-BGP-implementations>
- My personal email / jabber address
 - first@last.com
- All my previous ExaBGP presentations
 - <http://thomas.mangin.com/data/pdf/>

Questions?
(or comments)

