FRRouting

Community
Technical details
Testing

SiNOG 7.0, Sept 21, 2023

Martin Winter & David Lamparter
NetDEF

Who am I?

- **Martin Winter**
  - Started working on Quagga in 2011 (mostly testing)
  - Co-Founded NetDEF 2013
  - Started FRRouting fork (NetDEF together with Cumulus) in Apr 2017
  - FRR Maintainer & TSC Member
  - RIPE Open Source Working Group Co-Chair
  - Living in Switzerland

Who am I?

- **David ‘equinox’ Lamparter**
  - Started working on Quagga in 2004
  - FRR Maintainer, former TSC member
  - Active at IETF
  - AS13020 for a few days every year
  - Living in Germany

Who is NetDEF?

- **Network Device Education Foundation Inc**
  - California based US non-profit (501c3)
    - Living on donations from companies and implementing new FRR features as contracts for companies
  - Dedicated to Open Source & Education
  - Mostly working on FRRouting under the OpenSourceRouting project
  - Running CI System for the FRR community
  - Multiple maintainers

Open Source Routing

netDEF

FRRouting

What & Organization

What is FRRouting?

- **FRRouting = FRR**
  - Open Source (GPLv2+) Routing Stack
  - Runs on Linux and most BSD based systems
  - Forked 2017 from Quagga
  - Used in many Clouds as virtual routers, white box vendors and network providers (full routing stack)

What is FRRouting?

- **FRRouting protocols**
  - BGP, RPKI, BMP, EVPN,
  - OSPFv2, OSPFv3, IS-IS
  - RIP, RIPng, EIGRP, Babel
  - LDP, BFD, NHRP, VRRP
  - IGMP, MLD, PIM
  - Segment Routing, Policy-based routing

Who is behind FRRouting?

- **Open Community, no single Company**
  - Anyone is welcome to contribute!
  - Currently 26 maintainers from various organizations (anyone showing regular and useful contributions can become a maintainer)
  - 6 Technical Steering Committee members (voted yearly from the maintainers and by the maintainers)
  - Under the Linux Foundation as a Project

FRRouting - Where?

- Web: https://frrouting.org
- Git: https://github.com/frrouting/frr
  - Issue reporting
  - Submitting Pull Requests
- Slack (Join link https://frrouting.org/community/)
  - Weekly developer meeting (Slack Huddle) on #development channel (Tuesdays 17:00 CET)
- Mailing lists (https://lists.frrouting.org/listinfo)
- RPM Repo https://rpm.frrouting.org
- Deb Repo https://deb.frrouting.org

FRRouting

How to contribute

Contributing to FRR

- Found a bug?
- Fixed documentation?
  - We love documentation updates!
- Have a suggestion?
- Missing a feature?
- Please contribute on Github!
  - Open an Issue
  - Submit a Pull Request

Contribution to FRR

- **Pull requests are automatically tested by CI System**
  - Please watch for failures
  - Update PR with fixes or ask for help if issue is not clear
  - New features must include a test (ie Topotests)
  - CLI changes require documentation update
  - After CI is successful they all need a manual review (and merge) by a maintainer from a different organization
  - Feel free to join the weekly call to ask for help or discuss your PR (or ask on slack)

FRRouting

Best Practices: 2023 Edition

SiNOG7 · Ljubljana, SI · 2023-09-21
FRRouting: David *'equinox'* Lamparter · NetDEF, Inc.

“it’s just a ~~Cisco~~ *industry standard* CLI, right?”

"it's just a ~~Cisco~~ *industry standard* CLI, right?"

Okay, yes, kind of, but also not quite. Let's get started, literally...

/etc/frr/daemons

- slight footgun: daemons to start are still explicitly configured

```
bgpd=yes
ospfd=no
ospf6d=no
isisd=yes
…
```

/etc/frr/daemons

- RPKI must be enabled here too
  `bgpd_options="-M rpki"`
  - may need to install “frr-rpki-rtrlib” package too
- requires **restart of affected daemons**
- same for SNMP

/etc/frr/daemons

- adding/removing daemons requires reload (not restart) of watchfrr
  (“`service frr reload`” in most distros)

```
/usr/lib/frr/watchfrr -d -F traditional  \
    zebra bgpd isisd staticd
```

Configuration profiles

- while you're editing `/etc/frr/daemons`…
  … uncomment one of these:

```
#frr_profile="traditional"
#frr_profile="datacenter"
```

```
/usr/lib/frr/watchfrr -d -F traditional  \
    zebra bgpd isisd staticd
```

Configuration profiles

- `traditional`:
  IETF standard timers
  default settings as usual
- `datacenter`:
  more aggressive timeouts,
  minor settings changes

Configuration profiles

- also stored as part of the config,
  config setting overrides `/etc/frr/daemons`
- default profile is distro dependent
  e.g. nVidia ⇒ `datacenter`

```
frr version 8.4.4
frr defaults traditional
```

Upgrade handling

- profile + version from config overrides defaults if they changed in a newer version
- **include this in automated configs**!

```
frr version 8.4.4
frr defaults traditional
```

Config automation

- use `frr-reload.py`
  (may be in a separate `frr-pythontools` package)
- check logs, it will warn if some setting couldn't be applied

Check logs?

- generally defaults to syslog
- highly recommended: `terminal monitor` vtysh command
  - (will only show FRR daemon logs, but will show them regardless of config)
    https://youtu.be/8psFQCEgA18 (38 seconds)

Check logs?

- `debug bgp updates` & co.
  - **do not keep these enabled in production**
    (significant slowdown + disk space if written)
- `log syslog info` + `terminal monitor`

  ⇒ debugs only on attached vtysh session

Check logs?

- install `frr-dbgsym` package

  exact name depends on distro

- crashes are rare nowadays, but just in case, get all the info on the **first** occurence…

Monitoring

- use “json” variants of CLI show commands
  - guaranteed stable (very rare exceptions)
- `vtysh -c "show bgp foo json"`
  - don’t try to script interactive vtysh
  - you can use multiple -c options

Future outlook

- YANG YANG YANG everywhere
- this will have *some* impact particularly on config automation and monitoring
- nothing in BGP yet, significant work to do
- FRR specific models, no OpenConfig/IETF (yet)

FRRouting

Testing

Compiling... (on VMs)

All on x86 (64bit) arch except where noted

ubuntu

Ubuntu 18.04 (+ i386 + ARM 32bit + ARM 64bit + PPC64)

Ubuntu 20.04

Ubuntu 22.04

Release packages built additionally for:
Ubuntu 20.04: + ARM 32bit + ARM 64bit
Ubuntu 22.04: + ARM 32 bit + ARM 64bit

CentOS

CentOS 7

Red Hat

RedHat 8
RedHat 9

debian

Debian 10 (+ ARM 64bit)

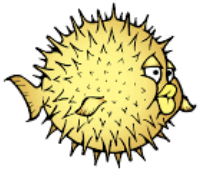
Debian 11 (+ ARM 64bit)

Debian 12

Release packages built additionally for:
Debian 10: + ARM 32bit + i386
Debian 11: + ARM 32bit + i386
Debian 12: + ARM 64bit + ARM 32bit + i386

FreeBSD®

FreeBSD 11
FreeBSD 12



OpenBSD

OpenBSD 7

Testing: Topotests

- FRR community developed Test Framework
- Based on pytest / Mininet
  - Collection of ~280 scripts with ~1600 tests total
- Linux Network Namespaces to build topologies
- Currently executed on Ubuntu x86_64, arm8 and Debian x86_64
- More details?
  - http://docs.frrouting.org/projects/dev-guide/en/latest/topotests.html

Testing: Other Testing

- **IXIA IxANVL (RFC Compliance Tests)**
- **Fuzzing (Coverity & Google OSS-Fuzz)**
- **Static Analysis (Clang)**
- **Address Sanitizer**
  - Executing all of Topotests one more time with Address Sanitizer
- **Installation Tests**
  - Installing/Uninstalling RPM/DEB packages

Coverity

freerangerouting/frr

Overview | Project Settings | Analysis Settings | Members

coverity | passed 6 new defects

Analysis Metrics

Version: master\ build

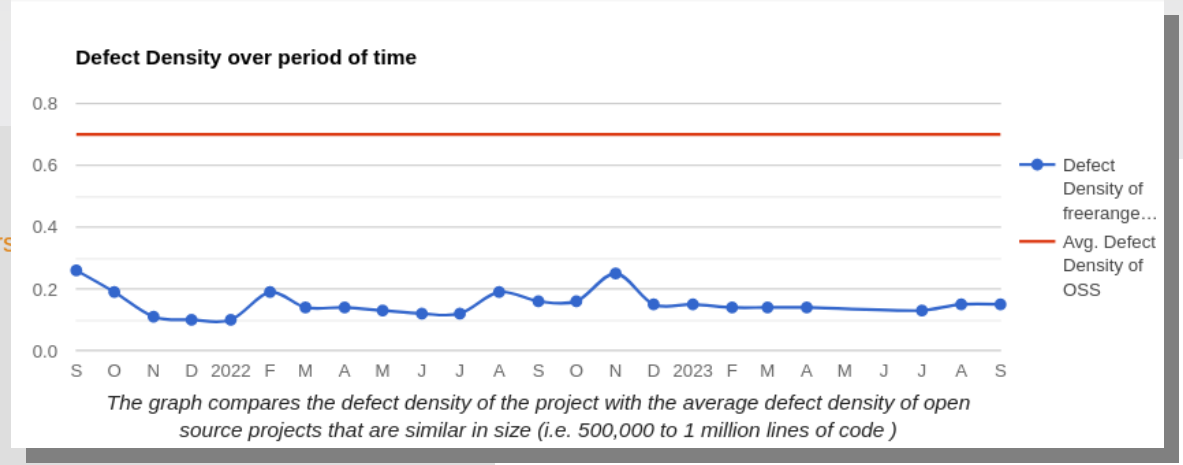
Sep 13, 2023	910,330	0.13
Last Analyzed	Lines of Code Analyzed	Defect Density

Defect changes since previous build dated Sep 08, 2023

6	26
Newly detected	Eliminated

Defects by status for current build

1,553	120	63	1,370
Total defects	Outstanding	Dismissed	Fixed



Testing: Challenges

- **Quality of Tests**
  - Badly written tests
  - Timing Sensitivity / unpredictable
    - RFC defined timers make test slow
    - Expect convergence after fixed time
  - No good output/logs if failed
  - Long runtime on some feature tests
  - New users need to learn on how to write good tests

Example (Bad) Output

Job: TopoTests Ubuntu 18.04 arm8 Part 9 failed

Job Summary | Tests | Commits | Artifacts | Logs | Metadata

ospf gr helper tc3 p1: Test case result

The below summarizes the result of the test " ospf gr helper tc3 p1" in build 7,049 of FRRouting - FRR - TopoTests Ubuntu 18.04 arm8 Part 9.

Description	ospf gr helper tc3 p1	Duration	4 mins
Test class	ospf_gr_helper.test_ospf_gr_helper2	Status	Failed (New Failure)
Method	test_ospf_gr_helper_tc3_p1		

Error Log

```
AssertionError: Testcase test_ospf_gr_helper_tc3_p1 : Failed
   Error: [DUT: FRR] OSPF GR Helper: activeRestarterCnt
assert '[DUT: FRR] OSPF GR Helper: activeRestarterCnt' is True
E   AssertionError: Testcase test_ospf_gr_helper_tc3_p1 : Failed
      Error: [DUT: FRR] OSPF GR Helper: activeRestarterCnt
    assert '[DUT: FRR] OSPF GR Helper: activeRestarterCnt' is True
```

Testing: Challenges

- **CI Stability**
  - 600+ Build & Test Systems
  - X86_64, i386, arm8, arm7, ppc
  - Lots of different distros, no unique setup
  - Need to test different distros / kernels / architectures
- **Limited Budget**
- CI provided & run by NetDEF (using Atlassian Bamboo)

Testing: Challenges

- **Distros change**
  - New versions
- **Package dependencies change with Updates**
  - Which version to test against?
- **Constant FRR changes**
  - new tests
  - new features
  - new dependencies

Testing: Outlook

- **Topotests are limiting: Too slow**
  - Micronet Infrastructure with pytest xdist
  - Micronet is an enhanced Topotest infra, not using mininet anymore and improving parallelization
  - Needs some learning (not everything serial anymore)
  - Massive Speedup (10x 0.5..2hrs → 1x 45mins)
  - Problem: Causes Timing changes (and new unpredictable fails)

Testing: Outlook

- **Topotests are limiting: Too badly written**
  - Topotato Infrastructure
  - Limiting scripts to simpler topology creation, simpler verification
  - Forcing users to more uniform test implementation
  - Improving Output to make failures understandable

Testing: Topotato Future

log messages: err warn notify info debug startup/??? | CLI | repeats with identical output (↑)

test_mld_basic.py::MLDBasic

- Download network diagram (as shown to the right, SVG)
- Download captured packets (pcap-ng)

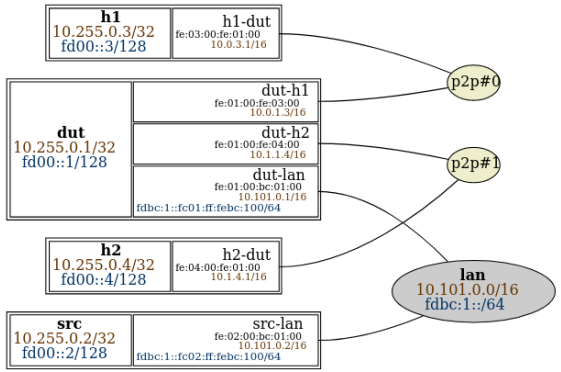
dut | zebra.conf | pim6d.conf

::startup
::prepare:#73:dut/pim6d/log
::prepare:#80:h1_dut/packet
::test_ssm:#89:h1/h1-dut/multicast-join[fdbc:1::fc02:ff:febc:100,ff05::2345]
::test_ssm:#91:dut/pim6d/log
::test_ssm:#92:dut/pim6d/vtysh[debug show mld interface dut-h1]
::test_ssm:#96:src/scapy[src-lan/IPv6/UDP]
::test_ssm:#106:h1_dut/packet
::test_asm:#110:h1/h1-dut/multicast-join[*,ff05::1234]
::test_asm:#112:dut/pim6d/log
::test_asm:#113:dut/pim6d/vtysh[debug show mld interface dut-h1]
::test_no_rtralert:#123:h1/scapy[h1-dut/IPv6/ICMPv6MLReport2]
::test_no_rtralert:#128:dut/pim6d/log
::test_invalid_group:#139:h1/scapy[h1-dut/IPv6/IPv6ExtHdrHopByHop/ICMPv6MLReport2]
::test_invalid_group:#144:dut/pim6d/log
::shutdown

::startup

▸ passed after 1.60s

-1.116	dut	zebra	X25ZY-TGTPJ◇	info	yang model directory "/usr/lib/frr/share/yang" does not exist
-1.113	dut	zebra	X25ZY-TGTPJ◇	info	yang model directory "/usr/lib/frr/share/yang" does not exist
-1.080	dut	zebra	X0AM3-MRSP8◇	notif	can't setsockopt NETLINK_ADD_MEMBERSHIP for group RTNLGRP_TUNNEL(34), this linux kernel does not support it: Invalid argument(22)
-1.080	dut	zebra	PV4NQ-1GG1Z◇	notif	Registration for RTNLGRP_BRVLAN Membership failed : 22 Invalid argument
-1.043	dut	zebra	T83RR-8SM5G◇	notif	zebra 9.1-dev--git. starting: vty@2601
-1.001	dut	zebra			⊡ enable
-0.999	dut	zebra			⊡ configure
-0.999	dut	zebra			⊡ log file /tmp/tmphyhk3pgl/dut/zebra.log



Questions ?

JOIN US:

https://frrouting.org

github.com/FRRouting

Mailing lists: lists.frrouting.org