FRRouting Webinar

Donald Sharp
Principal Engineer
Cumulus Networks

Genesis of an Open Source Project

- History
- A Brief Overview
- Who is involved?
- Community
- Project Goals
- How to Contribute

History of FRR

~1996 - Zebra development begins

↓

~2002 - Quagga forked from Zebra

↓

2016 - FRR forked from Quagga

↓

🎉 Jan. 2017 - First Release - FRR 2.0 🎉

↓

Oct 15, 2019 - Latest Stable - FRR 7.2

A Brief Overview

- Same routing capabilities of major vendors
  - Today development is supported by a large number of companies
    - https://frrouting.org
- Configuration similar to other vendors
  - Configuration file and interactive CLI available
- Runs natively on Linux and many other platforms
  - All features work on Linux, not true for other platforms
- In general, Uses the Kernel’s routing Stack for packet forwarding
- GPLv2+

Who is Involved?

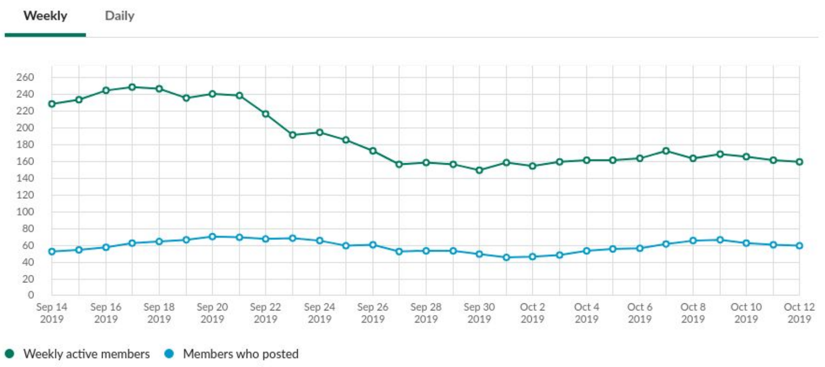
- https://frrouting.org/#contributors
- Oversight provided by the Linux Foundation
- Latest Release, 7.2 has 50 Individual Contributors!

128 TECHNOLOGY
6WIND
ARCHITECTURE TECHNOLOGY CORPORATION
big switch networks
BISDN
cloudscale.ch
CUMULUS
ISC Internet Systems Consortium
LabN Consulting LLC
LANDITEC creating value in network distribution
netDEF
Open Source Routing
orange
Pluribus NETWORKS
vmware
VOLTA Networks
XCLOUD NETWORKS
THE LINUX FOUNDATION

Community

- Healthy, active & growing community
  - ~800 people on community Slack
  - ~160 active weekly
  - Users support each other and have direct access to developers
  - https://frrouting.org/
    - Click on slack link for auto-invitation
  - https://lists.frrouting.org/listinfo
    - Subscribe to email aliases
  - https://github.com/FRRouting/frr/issues
    - Report Issues
  - https://github.com/FRRouting/frr/pulls
    - What are we working on?
  - Weekly Technical Meeting

Topic	List
Development	dev@lists.frrouting.org
Users & Operators	frog@lists.frrouting.org
Announcements	announce@lists.frrouting.org
Security	security@lists.frrouting.org



Project Goals

- Provide a world class routing stack
- Innovation
- Provide ease of use
  - Configurability
  - Debuggability
- Allow end user to customize to their needs
  - Configuration options
- Evolution -vs- Revolution

How to Contribute

- Find something that interests you
- Read our workflow
  - http://docs.frrouting.org/projects/dev-guide/en/latest/workflow.html
- Create a GitHub Account
  - https://github.com/
- Join our Slack and development alias
- Major Holes
  - Documentation
  - More Testing

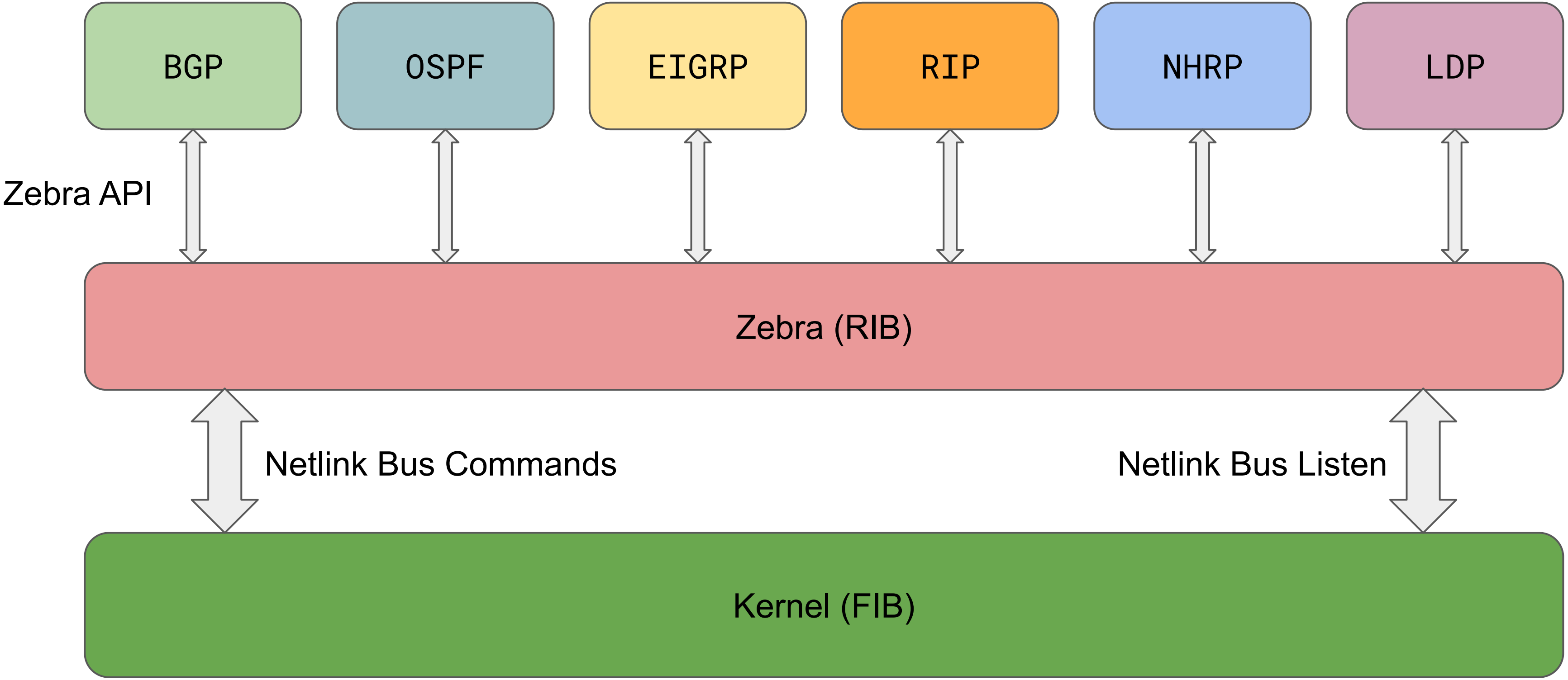
FRRouting Architecture

- Current Structure
- Future Changes coming
- Testing

Current Structure

- Current High Level Architecture
- What is Zapi?
- Data Flow for a Route Install
- Zebra Threading Model
- BGP Threading Model

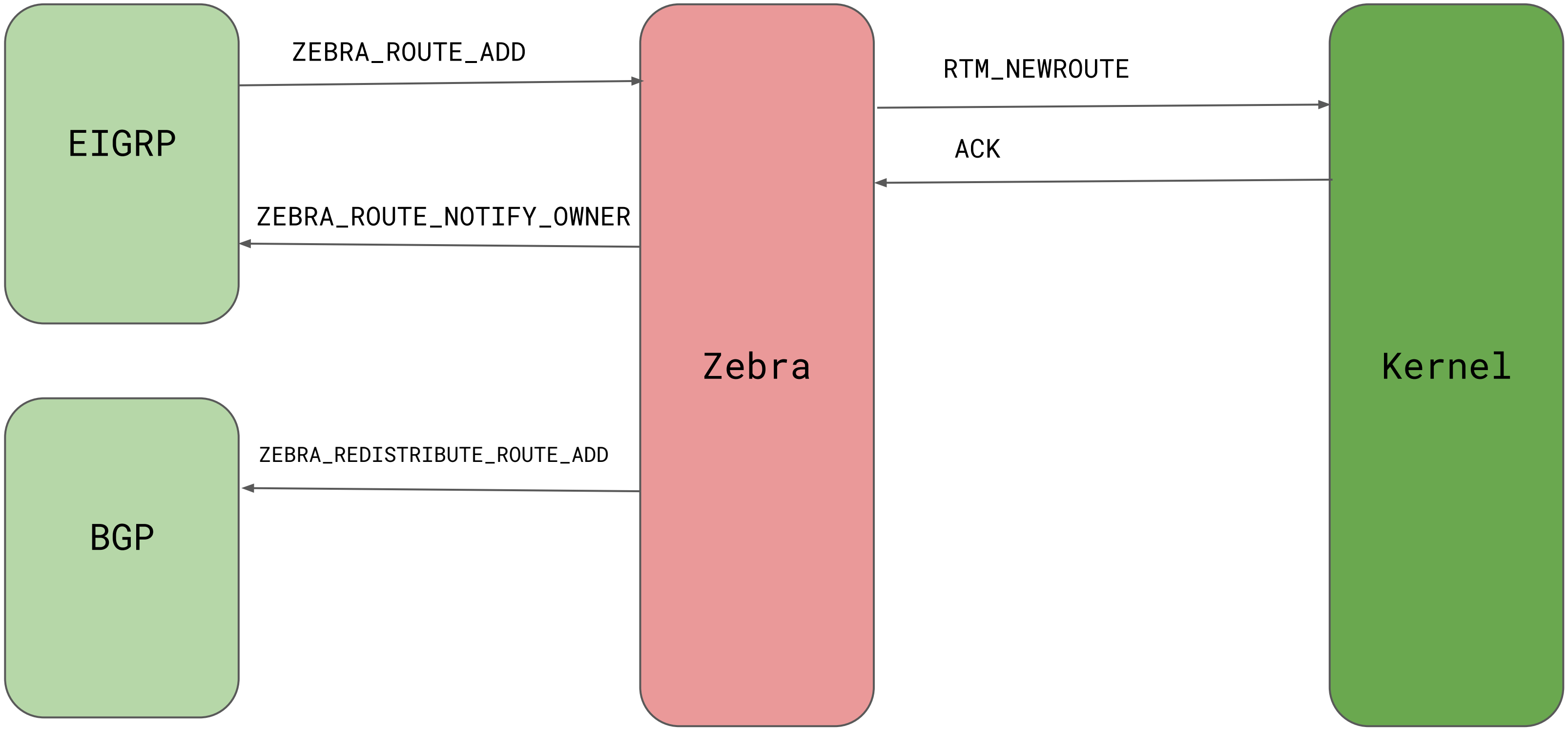
Current High Level Architecture



What is ZAPI?

- Between Daemon and Zebra only
- Protocol Messages
  - `ZEBRA_REDISTRIBUTE_XXX`
  - `ZEBRA_ROUTER_ID_XXX`
  - `ZEBRA_BFD_DEST_XXX`
  - `ZEBRA_NEXTHOP_XXX`
  - `ZEBRA_CAPABILITIES`
- Stream of data with each message type being sent as a packet
- Named Sockets
  - Only talk to FRR so we can change format as needed

Data Flow for Route Install

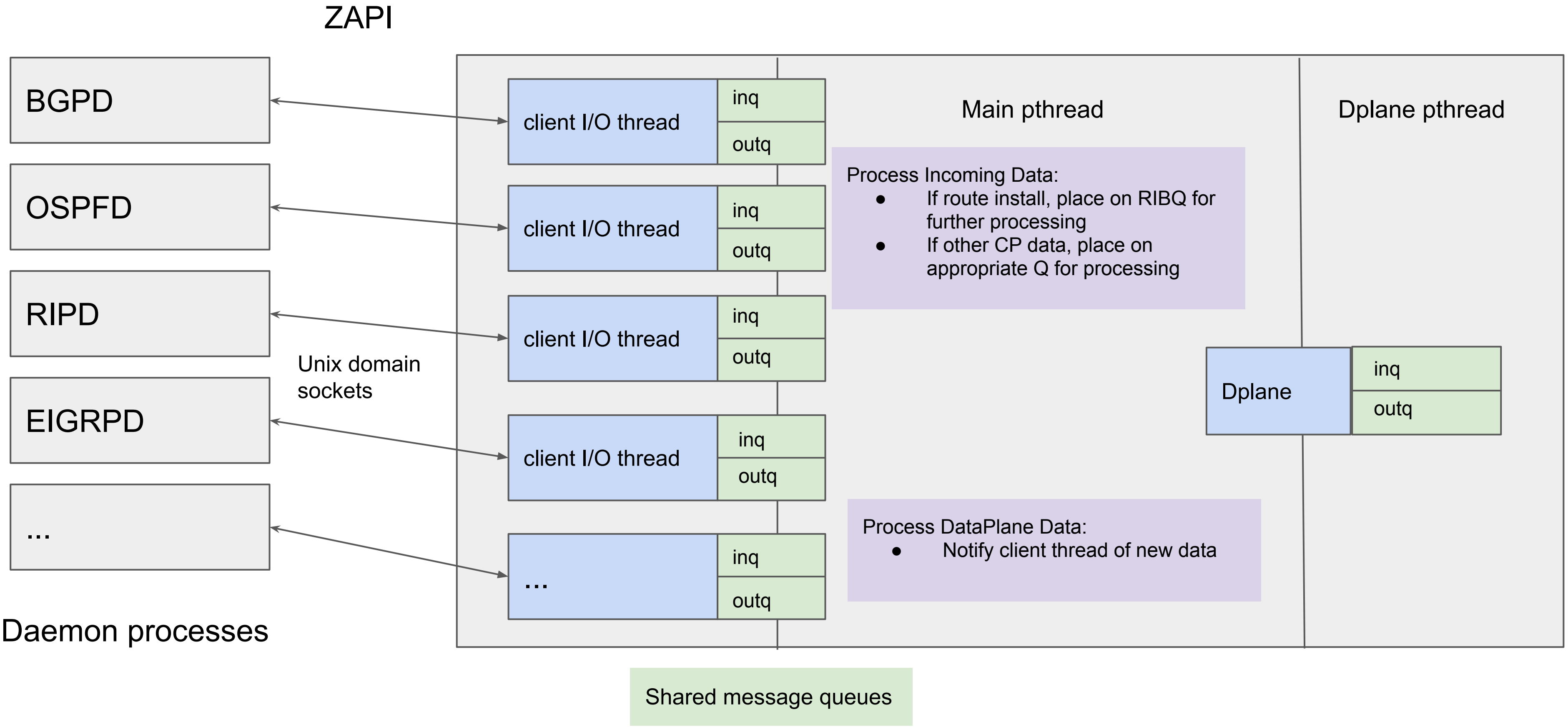


Zeba Threading Model

Events on main thread

Shared data

pthread



ZAPI

BGPD

OSPFD

RIPD

EIGRPD

...

Daemon processes

Unix domain sockets

client I/O thread

inq

outq

client I/O thread

inq

outq

client I/O thread

inq

outq

client I/O thread

inq

outq

...

inq

outq

Main pthread

Process Incoming Data:

- If route install, place on RIBQ for further processing
- If other CP data, place on appropriate Q for processing

Process DataPlane Data:

- Notify client thread of new data

Dplane pthread

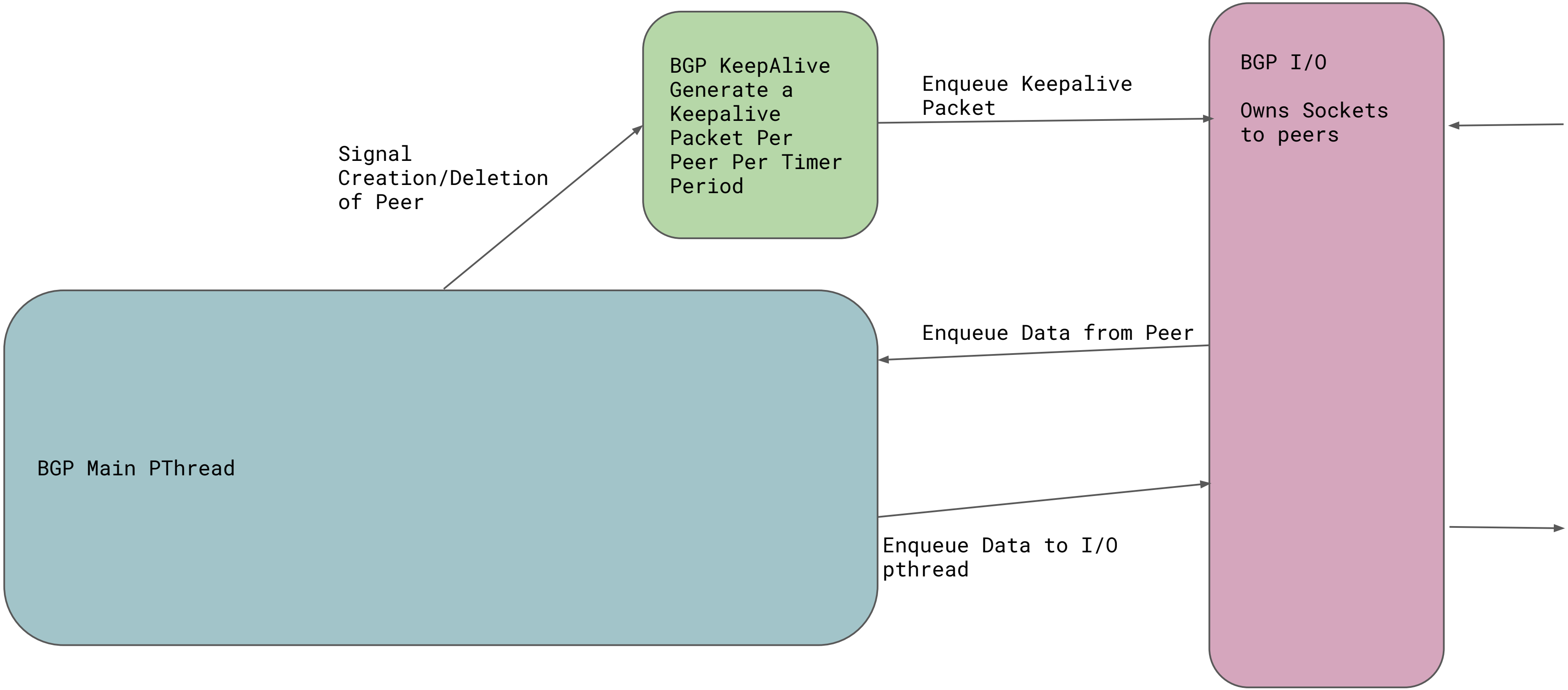
Dplane

inq

outq

Shared message queues

BGP Threading Model



Signal Creation/Deletion of Peer

BGP KeepAlive Generate a Keepalive Packet Per Peer Per Timer Period

Enqueue Keepalive Packet

BGP I/O

Owns Sockets to peers

Enqueue Data from Peer

BGP Main PThread

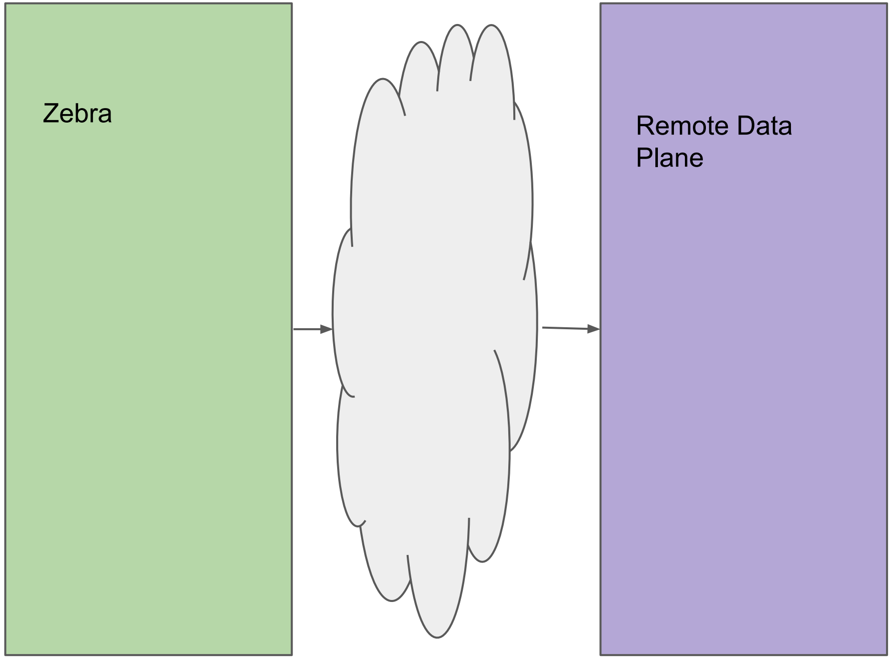
Enqueue Data to I/O pthread

Future Changes

- Asynchronous Data Plane
- Nexthop Groups
- Netconf/Yang

Asynchronous Data Plane

- Support for Data Plane(s) that are not directly connected to the controller
- Zebra currently assumes that we are synchronous
  - Tendrils of this assumption are everywhere
    - Interfaces
    - Netlink messages
  - Current Bugs because we support different kernel backends now
    - Fun!



Nexthop Groups

- Nexthop Group Motivation
- Current -vs- Future
- Kernel Interface

Nexthop Group Motivation

- Linux Kernel Now supports nexthop objects
- Zebra memory usage is Awful
- Allows for Pushing of Nexthop Groups into other Routing Protocols
    - Switch all routes using NHG A to NHG B
        - Both from upper level protocols and into Kernel
            - BGP PIC

NextHop Group Motivation

```
nexthop-group THIRTYTWO
 nexthop 192.168.41.1
 nexthop 192.168.41.10
 nexthop 192.168.41.11
 nexthop 192.168.41.12
 nexthop 192.168.41.13
 nexthop 192.168.41.14
 nexthop 192.168.41.15
 nexthop 192.168.41.16
 nexthop 192.168.41.17
 nexthop 192.168.41.18
 nexthop 192.168.41.19
 nexthop 192.168.41.2
 nexthop 192.168.41.20
 nexthop 192.168.41.21
 nexthop 192.168.41.22
 nexthop 192.168.41.23
 nexthop 192.168.41.24
 nexthop 192.168.41.25
 nexthop 192.168.41.26
 nexthop 192.168.41.27
 nexthop 192.168.41.28
 nexthop 192.168.41.29
 nexthop 192.168.41.3
 nexthop 192.168.41.30
 nexthop 192.168.41.31
 nexthop 192.168.41.32
 nexthop 192.168.41.4
 nexthop 192.168.41.5
 nexthop 192.168.41.6
 nexthop 192.168.41.7
 nexthop 192.168.41.8
 nexthop 192.168.41.9
!
```

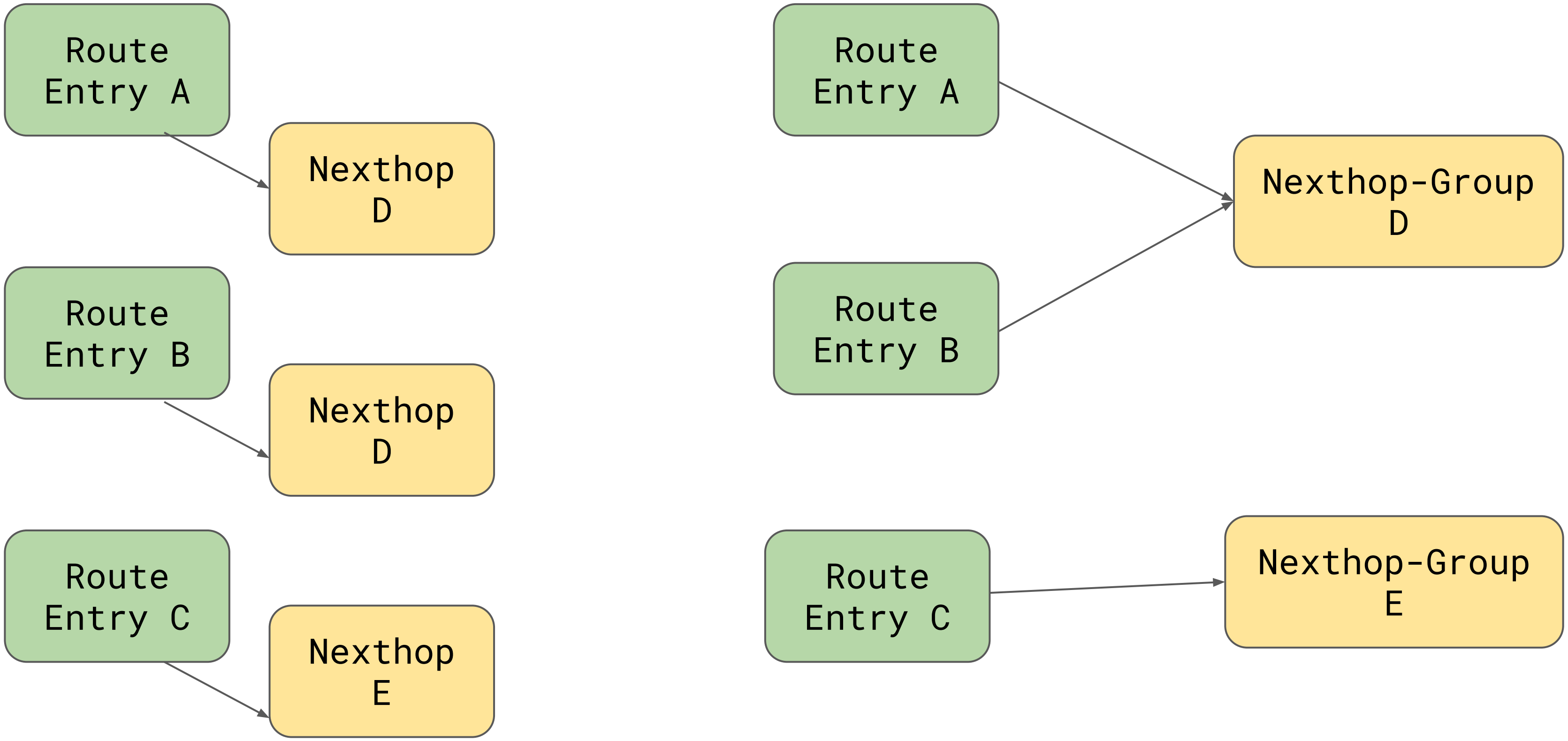
```
robot# sharp install routes 10.0.0.0 nexthop-group THIRTYTWO 1000000
robot# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected              16                 16
sharp                1000000              1000000
------
Totals               1000016              1000016
robot# show mem
Memory statistics for zebra:
System allocator statistics:
  Total heap allocated:  > 2GB
  Holding block headers: 0 bytes
  Used small blocks:     0 bytes
  Used ordinary blocks:  > 2GB
  Free small blocks:     14 MiB
  Free ordinary blocks:  17 MiB
  Ordinary blocks:       0
  Small blocks:          0
  Holding blocks:        0
(see system documentation for 'mallinfo' for meaning)
```

Nexthop Group Motivation

```
robot# show nexthop-group 96
ID: 96
     RefCnt: 1000000
     VRF: default
     Valid, Installed
     Depends: (97) (98) (99) (100) (101) (102) (103) (104) (105) (106) (107)
(108) (109) (110) (111) (112) (113) (114) (115) (116) (117) (118) (119) (120)
(121) (122) (123) (124) (125) (126) (127) (128)
           192.168.41.1, eveth1 (vrf default)
           192.168.41.2, eveth1 (vrf default)
           192.168.41.3, eveth1 (vrf default)
           192.168.41.4, eveth1 (vrf default)
           192.168.41.5, eveth1 (vrf default)
           192.168.41.6, eveth1 (vrf default)
           192.168.41.7, eveth1 (vrf default)
           192.168.41.8, eveth1 (vrf default)
           192.168.41.9, eveth1 (vrf default)
           192.168.41.10, eveth1 (vrf default)
           192.168.41.11, eveth1 (vrf default)
           192.168.41.12, eveth1 (vrf default)
           192.168.41.13, eveth1 (vrf default)
           192.168.41.14, eveth1 (vrf default)
           192.168.41.15, eveth1 (vrf default)
           192.168.41.16, eveth1 (vrf default)
           192.168.41.17, eveth1 (vrf default)
           192.168.41.18, eveth1 (vrf default)
           192.168.41.19, eveth1 (vrf default)
           192.168.41.20, eveth1 (vrf default)
           192.168.41.21, eveth1 (vrf default)
           192.168.41.22, eveth1 (vrf default)
           192.168.41.23, eveth1 (vrf default)
           192.168.41.24, eveth1 (vrf default)
           192.168.41.25, eveth1 (vrf default)
           192.168.41.26, eveth1 (vrf default)
           192.168.41.27, eveth1 (vrf default)
           192.168.41.28, eveth1 (vrf default)
           192.168.41.29, eveth1 (vrf default)
           192.168.41.30, eveth1 (vrf default)
           192.168.41.31, eveth1 (vrf default)
           192.168.41.32, eveth1 (vrf default)
```

```
robot# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected            16                   16
sharp                1000000              1000000
------
Totals               1000016              1000016
robot# show mem
Memory statistics for zebra:
System allocator statistics:
  Total heap allocated:  414 MiB
  Holding block headers: 0 bytes
  Used small blocks:     0 bytes
  Used ordinary blocks:  399 MiB
  Free small blocks:     7372 KiB
  Free ordinary blocks:  8524 KiB
  Ordinary blocks:       0
  Small blocks:          0
  Holding blocks:        0
(see system documentation for 'mallinfo' for meaning)
```

Nexthop Groups Current -vs- Future



Kernel Interface

- Available in Linux 5.3
- Netlink
  - RTM_NEWNEXTHOP
  - RTM_DELNEXTHOP
- `ip nexthop show`
  - Normal `ip route show` is unchanged
- Allows for O(1) Replacement (Future work for FRR)
  - NLM_F_REPLACE

Kernel Interface

```
[sharpd@robot ~]$ sudo ip nexthop show
<snip>
id 96 group 97/98/99/100/101/102/103/104/105/106/107/108/109/110/111/112/113/114/115/116/117/118/119/120/121/122/123/124/125/126/127/128 proto zebra
id 97 via 192.168.41.1 dev eveth1 scope link proto zebra
id 98 via 192.168.41.2 dev eveth1 scope link proto zebra
id 99 via 192.168.41.3 dev eveth1 scope link proto zebra
id 100 via 192.168.41.4 dev eveth1 scope link proto zebra
id 101 via 192.168.41.5 dev eveth1 scope link proto zebra
id 102 via 192.168.41.6 dev eveth1 scope link proto zebra
id 103 via 192.168.41.7 dev eveth1 scope link proto zebra
id 104 via 192.168.41.8 dev eveth1 scope link proto zebra
id 105 via 192.168.41.9 dev eveth1 scope link proto zebra
id 106 via 192.168.41.10 dev eveth1 scope link proto zebra
id 107 via 192.168.41.11 dev eveth1 scope link proto zebra
id 108 via 192.168.41.12 dev eveth1 scope link proto zebra
id 109 via 192.168.41.13 dev eveth1 scope link proto zebra
id 110 via 192.168.41.14 dev eveth1 scope link proto zebra
id 111 via 192.168.41.15 dev eveth1 scope link proto zebra
id 112 via 192.168.41.16 dev eveth1 scope link proto zebra
id 113 via 192.168.41.17 dev eveth1 scope link proto zebra
id 114 via 192.168.41.18 dev eveth1 scope link proto zebra
id 115 via 192.168.41.19 dev eveth1 scope link proto zebra
id 116 via 192.168.41.20 dev eveth1 scope link proto zebra
id 117 via 192.168.41.21 dev eveth1 scope link proto zebra
id 118 via 192.168.41.22 dev eveth1 scope link proto zebra
id 119 via 192.168.41.23 dev eveth1 scope link proto zebra
id 120 via 192.168.41.24 dev eveth1 scope link proto zebra
id 121 via 192.168.41.25 dev eveth1 scope link proto zebra
id 122 via 192.168.41.26 dev eveth1 scope link proto zebra
id 123 via 192.168.41.27 dev eveth1 scope link proto zebra
id 124 via 192.168.41.28 dev eveth1 scope link proto zebra
id 125 via 192.168.41.29 dev eveth1 scope link proto zebra
id 126 via 192.168.41.30 dev eveth1 scope link proto zebra
id 127 via 192.168.41.31 dev eveth1 scope link proto zebra
id 128 via 192.168.41.32 dev eveth1 scope link proto zebra
```

NetConf/Yang

- Motivation
- Proposed Design
- Where we are With Netconf/Yang
- Commit and Rollback

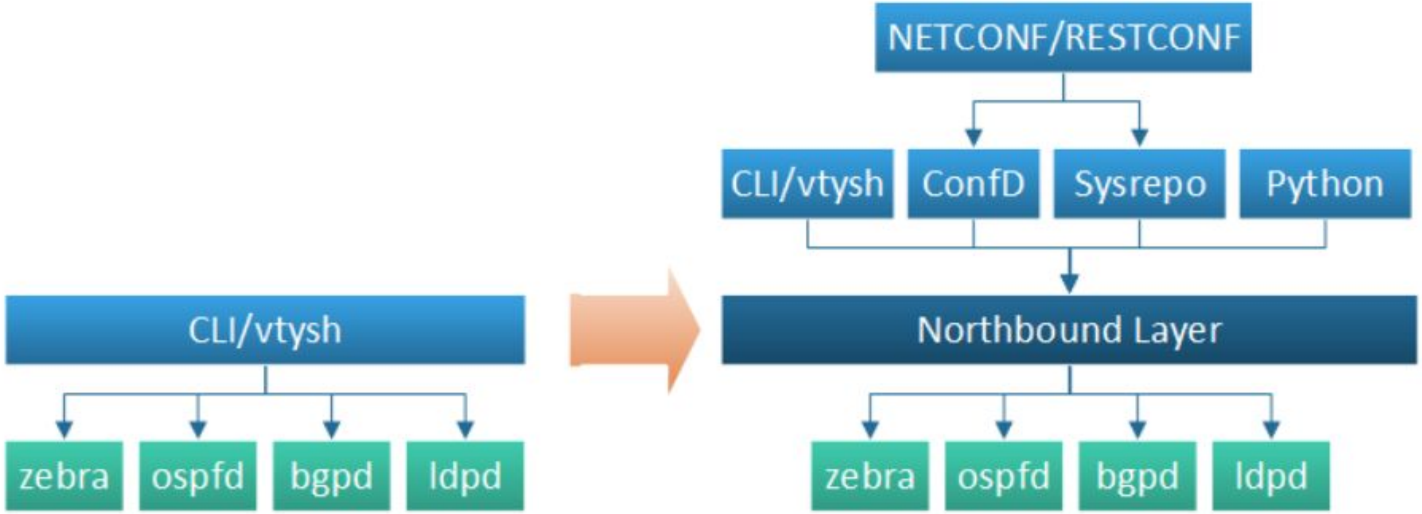
NetConf/Yang Motivation

- Provide a better interface to configure and monitor FRR programmatically
  - Mostly Human Only now
- Allow FRR to be configured using more computer friendly formats
  - JSON
  - XML
  - Protocol Buffers
- Use YANG to model and validate the configuration
  - Paves the way to support NETCONF/RESTCONF
  - Use YANG models as a formal contract
  - Mechanism to introduce backward incompatible changes

Proposed Design

- Introduce configuration transactions to remove the burden of error recovery from management side
- Encapsulate the logic associated with configuration changes inside configuration callbacks
- Leverage the new infrastructure to implement configuration reloads without external help

Proposed Design



Where we are with NetConf/Yang

- BABEL
- BFD (Done)
- BGP
- EIGRP (Done)
- ISIS (Done)
- LDP
- NHRP
- OPENFABRIC
- OSPF v2/v3
- PBR
- PIM
- RIP/RIPNG (Done)
- SHARP
- STATIC
- VRRP - In Progress
- Watchfrr
- ZEBRA - Yang Model Submitted
- Northbound API improvements
  - Memory and CPU

- YANG Model’s
  - FRR will define appropriately

- Model for New Work
  - Define YANG Model
  - Write callbacks needed by YANG
  - Abstract the configuration and show code

Commit and Rollback

```
robot(config)# router rip
robot(config-router)# network 0.0.0.0/0
robot(config-router)# show configuration candidate changes
@@ -3,4 +3,17 @@
 frr version 7.3-dev
 frr defaults datacenter
 !
+router rip
+ network 0.0.0.0/0
+!
 end
robot(config-router)# commit
% Configuration committed successfully (Transaction ID #1).

robot(config)# show configuration transaction
 Transaction ID  Client  Date                 Comment
 ------------------------------------------------------
 1               CLI     2019-10-21 19:06:58

robot(config-router)# allow-ecmp
robot(config-router)# commit
% Configuration committed successfully (Transaction ID #2).

robot(config-router)# redistribute connected
robot(config-router)# commit
% Configuration committed successfully (Transaction ID #3).

robot(config-router)# exit
robot(config)# exit
```

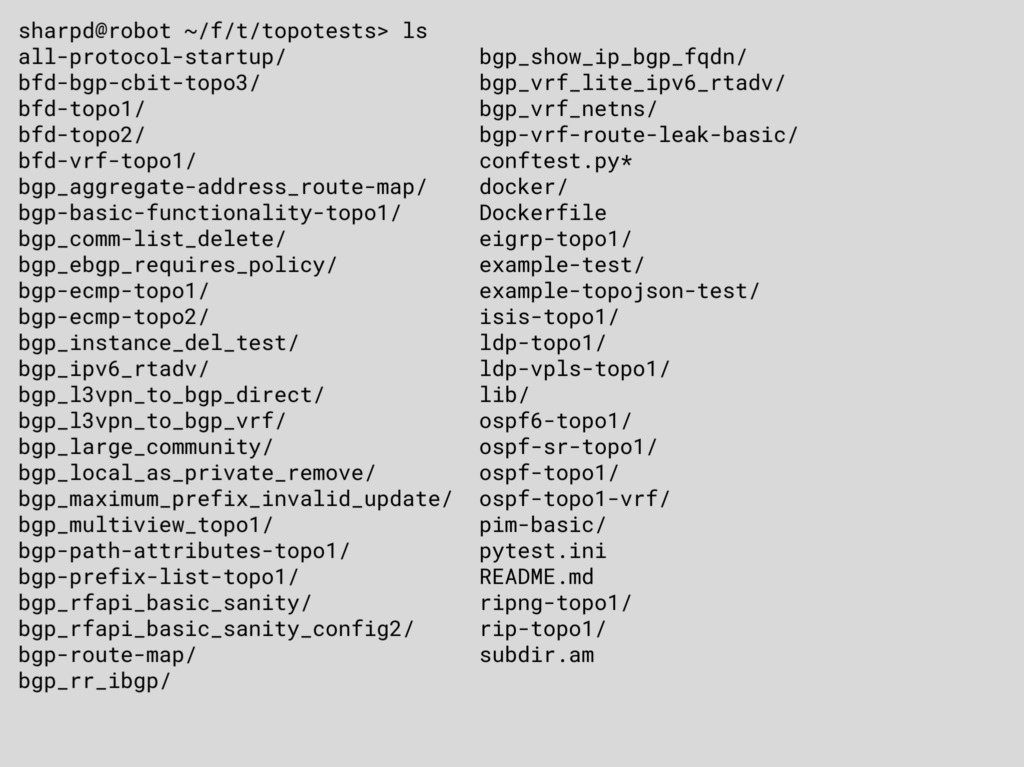
```
robot# show configuration transaction
 Transaction ID  Client  Date                 Comment
 ------------------------------------------------------
 3               CLI     2019-10-21 19:09:42
 2               CLI     2019-10-21 19:09:31
 1               CLI     2019-10-21 19:06:58

robot# show configuration transaction 3
Configuration:
router rip
 allow-ecmp
 redistribute connected
 network 0.0.0.0/0
!
end
robot(config)# configuration load transaction 1 replace
robot(config)# commit
% Configuration committed successfully (Transaction ID #4).

robot(config)# do show run
<snip>
!
router rip
 network 0.0.0.0/0
!
```

Commit and Rollback

```
robot# show config running with-defaults
Configuration:
!
frr version 7.3-dev
frr defaults datacenter
!
interface enp3s0
 ip rip split-horizon
 no ip rip v2-broadcast
 no ip rip receive version
 no ip rip send version
 no ip rip authentication mode
!
interface lo
 ip rip split-horizon
 no ip rip v2-broadcast
 no ip rip receive version
 no ip rip send version
 no ip rip authentication mode
!
router rip
 allow-ecmp
 no default-information originate
 default-metric 1
 no distance
 network 0.0.0.0/0
 no passive-interface default
 timers basic 30 180 120
 no version
!
```

Testing

- Unit Testing
    - Focuses on internal API
- Topology Testing
    - Builds a topology and ensures it works
- Every Commit runs against CI/CD system
- How do I do X?
- Topotests are 5-10% of every release
    - Nice to have



```
sharpd@robot ~/f/t/topotests> ls
all-protocol-startup/               bgp_show_ip_bgp_fqdn/
bfd-bgp-cbit-topo3/                 bgp_vrf_lite_ipv6_rtadv/
bfd-topo1/                          bgp_vrf_netns/
bfd-topo2/                          bgp-vrf-route-leak-basic/
bfd-vrf-topo1/                      conftest.py*
bgp_aggregate-address_route-map/    docker/
bgp-basic-functionality-topo1/      Dockerfile
bgp_comm-list_delete/               eigrp-topo1/
bgp_ebgp_requires_policy/           example-test/
bgp-ecmp-topo1/                     example-topojson-test/
bgp-ecmp-topo2/                     isis-topo1/
bgp_instance_del_test/              ldp-topo1/
bgp_ipv6_rtadv/                     ldp-vpls-topo1/
bgp_l3vpn_to_bgp_direct/            lib/
bgp_l3vpn_to_bgp_vrf/               ospf6-topo1/
bgp_large_community/                ospf-sr-topo1/
bgp_local_as_private_remove/        ospf-topo1/
bgp_maximum_prefix_invalid_update/  ospf-topo1-vrf/
bgp_multiview_topo1/                pim-basic/
bgp-path-attributes-topo1/          pytest.ini
bgp-prefix-list-topo1/              README.md
bgp_rfapi_basic_sanity/             ripng-topo1/
bgp_rfapi_basic_sanity_config2/     rip-topo1/
bgp-route-map/                      subdir.am
bgp_rr_ibgp/
```

Usability and Features

- Recent Ease of Use Changes
- Features Recently Implemented
- Features Coming soon

Recent Ease of Use changes

- BGP Changes
  - BestPath Reason
  - Hostname
  - Failed Neighbors
- Improved Debugging
  - Error Codes
  - `log-filter WORD`
  - Why was my route rejected?
- What was that command?

BGP BestPath Reason

```
act-7326-05# show bgp ipv4 uni 27.0.0.14
BGP routing table entry for 27.0.0.14/32
Paths: (3 available, best #2, table default)
  Advertised to non peer-group peers:
  TORC12(peerlink-3.4094)
  4435 5548
    MSP1 from MSP1(uplink-1) (27.0.0.9)
    (fe80::202:ff:fe00:12) (used)
      Origin incomplete, valid, external, multipath
      Last update: Mon Oct 21 12:23:33 2019
  4435 5548
    MSP2 from MSP2(uplink-2) (27.0.0.10)
    (fe80::202:ff:fe00:1e) (used)
      Origin incomplete, valid, external, multipath, bestpath-from-AS 4435, best (Peer Type)
      Last update: Mon Oct 21 12:23:33 2019
  4435 5548
    TORC12 from TORC12(peerlink-3.4094) (27.0.0.12)
      Origin incomplete, localpref 100, valid, internal
      Last update: Mon Oct 21 12:23:29 2019
act-7326-05#
```

BGP Hostname

- draft-walton-bgp-hostname-capability-02
- Visible in summary

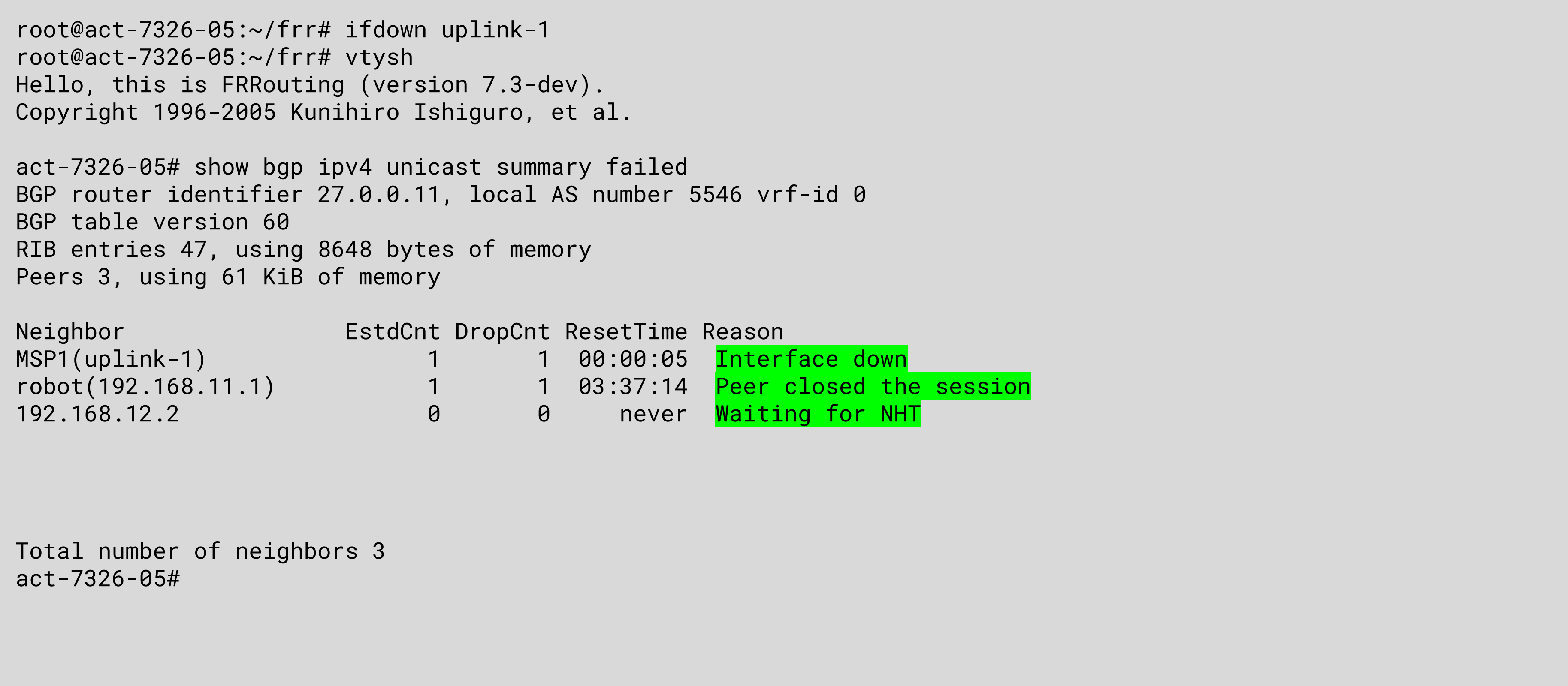
```
act-7326-05# show bgp ipv4 uni summ
BGP router identifier 27.0.0.11, local AS number 5546 vrf-id 0
BGP table version 38
RIB entries 47, using 8648 bytes of memory
Peers 3, using 62 KiB of memory

Neighbor                       V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd
TORC12(peerlink-3.4094)        4       5546      80      82        0    0    0 00:02:49           22
MSP1(uplink-1)                 4       4435     132      85        0    0    0 00:02:45           24
MSP2(uplink-2)                 4       4435     136      85        0    0    0 00:02:46           24

Total number of neighbors 3
act-7326-05#
```

BGP Hostname

- Visible in `show bgp ipv4 unicast A.B.C.D/M`

```
robot# show bgp ipv4 uni 44.0.0.0/9
BGP routing table entry for 44.0.0.0/9
Paths: (1 available, best #1, table default)
  Advertised to non peer-group peers:
  annie(192.168.201.136) janelle(192.168.201.138)
  64539 4294967000 15096 6939 2152 7377
    annie from annie(192.168.201.136) (192.168.201.136)
      Origin IGP, valid, external, bestpath-from-AS 64539, best (First path received)
      Last update: Mon Oct 21 17:32:23 2019
robot# show bgp ipv4 uni
<snip>
*> 44.0.0.0/9       annie                                  0 64539 4294967000 15096 6939 2152 7377 i
<snip>
Displayed  785540 routes and 785540 total paths
```

BGP failed neighbors



```
root@act-7326-05:~/frr# ifdown uplink-1
root@act-7326-05:~/frr# vtysh
Hello, this is FRRouting (version 7.3-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

act-7326-05# show bgp ipv4 unicast summary failed
BGP router identifier 27.0.0.11, local AS number 5546 vrf-id 0
BGP table version 60
RIB entries 47, using 8648 bytes of memory
Peers 3, using 61 KiB of memory

Neighbor                EstdCnt DropCnt ResetTime Reason
MSP1(uplink-1)                1       1  00:00:05  Interface down
robot(192.168.11.1)           1       1  03:37:14  Peer closed the session
192.168.12.2                  0       0     never  Waiting for NHT




Total number of neighbors 3
act-7326-05#
```

Improved Debugging - Error Codes

```
sharpd@robot /v/l/frr> sudo grep EC frr*.log
frr.log:2019/10/14 15:15:34.467459 BGP: [EC 100663304] ERROR: No such command on config line 8: intentional cli
error

robot# show error 100663304

Error 100663304 - VTY Subsystem Error
======================================
Description:
FRR has detected a problem with the specified configuration file

Recommendation:
Ensure configuration file exists and has correct permissions for operations Additionally ensure that all config
lines are correct as well
robot#
```

Improved Debugging - log-filter

```
act-7326-05# show debug
Zebra debugging status:
  Zebra event debugging is on
  Zebra kernel debugging is on
  Zebra RIB debugging is on
  Zebra VXLAN debugging is on
  Zebra mlag debugging is on
BGP debugging status:
  BGP updates debugging is on (inbound)
  BGP updates debugging is on (outbound)
  BGP zebra debugging is on
PIM debugging status
  debug msdp events
  debug mroute
  debug mroute detail
  debug pim events
  debug pim trace
  debug pim zebra
  debug pim mlag
  debug pim vxlan
  debug pim packets joins
  debug pim packets register
  debug pim nht
```

```
act-7326-05(config)# log-filter 36.0.0.11
Applying log filter change to zebra:
 36.0.0.11

Applying log filter change to bgpd:
 36.0.0.11

Applying log filter change to pimd:
 36.0.0.11

Applying log filter change to sharpd:
 36.0.0.11

Applying log filter change to watchfrr:
 36.0.0.11

Applying log filter change to staticd:
 36.0.0.11

act-7326-05# clear bgp ipv4 uni *
act-7326-05#
```

Improved Debugging - log filter

```
2019/10/21 18:18:46.409187 BGP: %NOTIFICATION: rcvd End-of-RIB for IPv6 Unicast from peerlink-3.4094 in vrf
default
2019/10/21 18:18:46.409617 BGP: peerlink-3.4094 rcvd 36.0.0.11/32 IPv4 unicast
2019/10/21 18:18:46.409840 BGP: %NOTIFICATION: rcvd End-of-RIB for IPv4 Unicast from peerlink-3.4094 in vrf
default
2019/10/21 18:18:46.416612 BGP: u8:s8 send UPDATE w/ attr: nexthop 36.0.0.11, extcommunity ET:8 RT:5546:1008
RT:5546:4003 MM:0, sticky MAC Rmac:44:38:39:ff:ff:01, path
2019/10/21 18:18:46.416675 BGP: u8:s8 uplink-2 send UPDATE w/ nexthop 36.0.0.11
2019/10/21 18:18:46.450655 PIM: Sending Request for New Channel Oil Information(36.0.0.11,239.1.1.102) VIIF
3(default)
2019/10/21 18:18:46.450999 ZEBRA: Asking for (36.0.0.11,239.1.1.102)[default(0)] mroute information
2019/10/21 18:18:46.451103 ZEBRA: MCAST VRF: default(0) RTM_NEWROUTE (36.0.0.11,239.1.1.102) IIF:
peerlink-3.4094(992) OIF: peerlink-3.4094(992)  jiffies: 209820
2019/10/21 18:18:46.451270 PIM: pim_upstream_sg_running[(36.0.0.11,239.1.1.102)]: default old packet count is
equal or lastused is greater than 30, (0,0,2098)
2019/10/21 18:18:46.522498 BGP: %NOTIFICATION: rcvd End-of-RIB for IPv4 Unicast from uplink-2 in vrf default
2019/10/21 18:18:46.582885 PIM: local MLAG mroute (36.0.0.11,239.1.1.106) role changed to df based on
inherit_xg_df
2019/10/21 18:18:46.582996 PIM: local MLAG mroute (36.0.0.11,239.1.1.100) role changed to df based on
inherit_xg_df
2019/10/21 18:18:46.583076 PIM: local MLAG mroute (36.0.0.11,239.1.1.102) role changed to df based on
inherit_xg_df
```

Why did my Route get Rejected by Zebra?

- `debug zebra rib detail`

```
act-7326-05# show ip route
<snip>
O>* 192.168.86.5/32 [110/10] via 192.168.86.5, vti-vti0, 00:00:09
O   192.168.99.99/32 [110/10] via 192.168.86.5, vti-vti0 inactive, 00:00:09
```

```
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.86.5/32: Processing rn 0x9cf1600
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.86.5/32: Examine re 0x9ceeed0 (ospf) status 2 flags 0 dist 110 metric 10
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.86.5/32: After processing: old_selected 0x0 new_selected 0x9ceeed0 old_fib 0x0 new_fib 0x9ceeed0
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.86.5/32: Adding route rn 0x9cf1600, re 0x9ceeed0 (ospf)
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.86.5/32: rn 0x9cf1600 dequeued from sub-queue 2
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.99.99/32: Processing rn 0x9cf1780
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]: 0:192.168.99.99/32: Examine re 0x9ceef20 (ospf) status 2 flags 0 dist 110 metric 10
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]:     nexthop_active: Route Type ospf has not turned on recursion
Fri Oct 18 03:13:38 2019 daemon.debug zebra[3370]:     nexthop_active_check: Unable to find a active nexthop
```

What was that Command?

- `find WORD`
  - How many times is the command on the tip of your keyboard?

```
robot# find ecmp
  (config)  ip pim ecmp rebalance
  (config)  ip pim ecmp
  (config)  no ip pim ecmp rebalance
  (config)  no ip pim ecmp
  (vrf)  ip pim ecmp rebalance
  (vrf)  ip pim ecmp
  (vrf)  no ip pim ecmp rebalance
  (vrf)  no ip pim ecmp
  (rip)  [no] allow-ecmp
  (ripng)  [no] allow-ecmp
```

Features Recently Implemented

- 7.0
- 7.1
- 7.2
- Features Not Implemented

7.0 Release

- Yang & Netconf
  - RIP, ISIS
- Error Codes
- BGP
  - DAD
  - EVPN Route Servers
  - EVPN Extended Mobility
  - GUA for RFC5549
  - FlowSpec extended
  - Add-Path ID’s are now reused
- ISIS
  - BFD Support
  - Triggered hellos

- OpenFabric
  - draft-white-openfabric
- Staticd
  - Onlink nexthops
- Multiple Users edit CLI

- ~1500 commits
- ~30 authors

7.1 Release

- gRPC northbound config Interface
- IPv6 RA
  - RFC 81067 DNS options
- BGP
  - EVPN in non-default VRF’s
  - RFC 8212 - default Deny policy
  - Aggregation Improvements
- PIM
  - EVPN BUM handling
- RIP
  - VRF Support

- Track Route-Map Usage
- Debian Packaging Rewrite

- ~1000 commits
- 40 authors

7.2 Release

- BFD
  - VRF Support
- BGP
  - Route-map support for Aggregate Address
  - FQDN for peers
  - BestPath Selection
  - BMP
- Northbound Interface
  - BFD
- VRRP
  - VRRPv2 and VRRPv3

- ISIS
  - BFD Support
- PIM
  - BSM
- Zebra
  - RMAC and VXLAN support for FPM

- ~1000 commits
- 50 contributors

Features Not Implemented

- Do you have X?
- When is it available?

Features Coming Soon

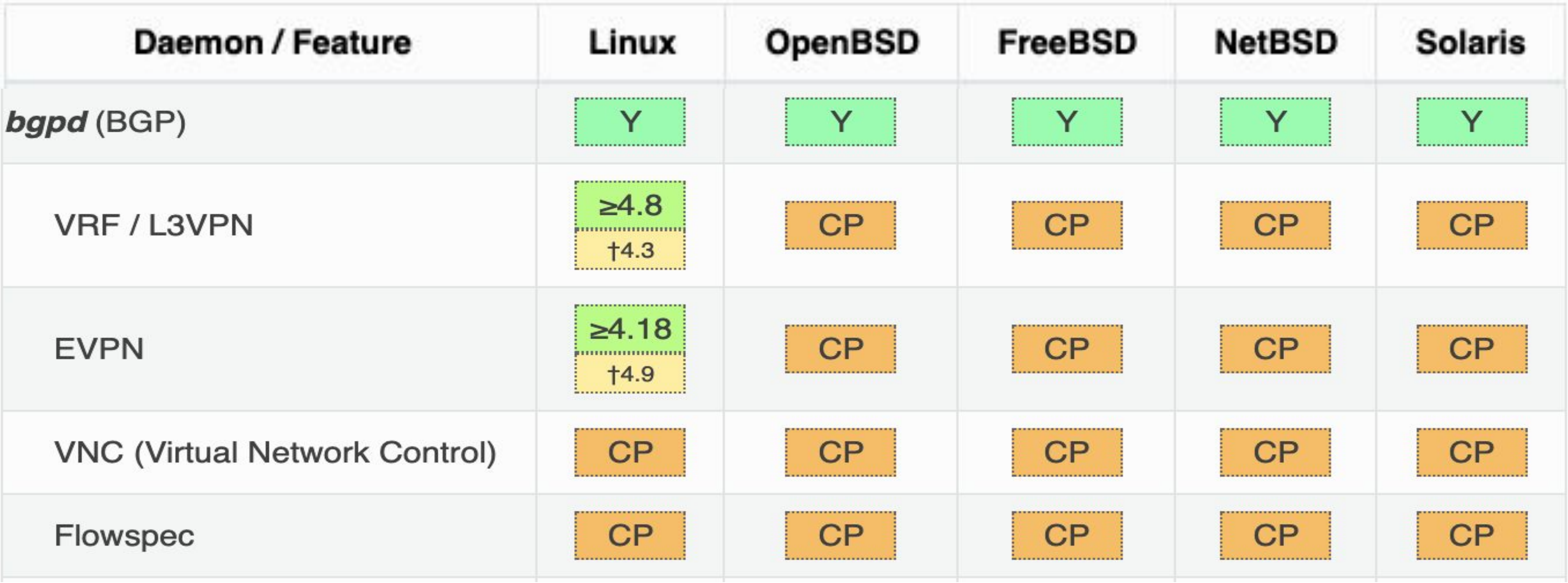
- EVPN MultiHoming
- Netconf/Yang for
  - Zebra
  - BGP
  - VRRP
- Zebra Nexthop Groups
- LUA Object Tracking
  - VRRP
  - Route-map Replacement
- BGP
  - Graceful Restart
  - IPv6 Flowspec
  - RPKI VRF aware
- SR
  - ISIS
  - RSVP-TE
- RIFT
- PIM-MLAG
- VRF awareness for all Protocols
- L2 Integration
- LDP
  - RFC 5443 (LDP IGP Synchronization)

How to get started using FRR

- Determining Feature Availability
- Vrf -vs- Namespace
- CPU and Memory Requirements
- Forwarding
- Where to get FRR?
- Common File Locations and Commands

Determining Feature Availability

http://docs.frrouting.org/en/latest/overview.html#supported-protocols-vs-platform



Daemon / Feature	Linux	OpenBSD	FreeBSD	NetBSD	Solaris
bgpd (BGP)	Y	Y	Y	Y	Y
VRF / L3VPN	≥4.8 †4.3	CP	CP	CP	CP
EVPN	≥4.18 †4.9	CP	CP	CP	CP
VNC (Virtual Network Control)	CP	CP	CP	CP	CP
Flowspec	CP	CP	CP	CP	CP

VRF -vs- Namespace

VRF	Namespace
A Linux device is created	Full Isolation
Associate Interfaces w/ the Linux Device similar to a bridge.	Associate Interfaces with Namespace
VRF Leaking just point the nexthop to the appropriate outgoing interface	VRF Route Leaking requires a tap per VRF NxN
	VRF Route Leaking not supported with namespace based VRF’s in FRR

CPU and Memory Requirements

- Zebra
  - When installing a full BGP table: 2 cpu’s
  - A Full BGP table takes ~500mb
    - Additional ECMP takes ~100mb/per
- BGP
  - When receiving a full table table: 2 cpu’s
  - A Full BGP table takes ~600mb
    - Additional tables are ~200mb/per
- Other Daemons
  - CPU per daemon running is not unreasonable
  - Dwarfed by Zebra and BGP needs, if no BGP then 4gb is sufficient

Forwarding Choices

	Forwarding Method	Recompilation	Speed
Kernel	Software	None	Slowest
Kernel, Source of Truth	Hardware	None	Line Rate
eBPF	Software, In Kernel and Linux Only	None	Faster
DPDK	Software, Userspace	Yes	Faster
FPM	Hardware	Yes and maintenance	Line Rate

Where to Get FRR?

- Download and Compile yourself
    - https://github.com/FRRouting/frr
- Debian Repository
    - https://deb.frrouting.org/
- FRR Release binaries
    - https://github.com/FRRouting/frr/releases

Control File Locations and Commands

- Control
    - /etc/frr/frr.conf
        - `show run`
    - /etc/frr/daemons
        - `show watchfrr`

```
# This file tells the frr package which daemons to start.
#
# Sample configurations for these daemons can be found in
# /usr/share/doc/frr/examples/.
#
# ATTENTION:
#
# When activating a daemon for the first time, a config file, even if it is
# empty, has to be present *and* be owned by the user and group "frr", else
# the daemon will not be started by /etc/init.d/frr. The permissions should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be owned by
# group "frrvty" and set to ug=rw,o= though. Check /etc/pam.d/frr, too.
#
# The watchfrr and zebra daemons are always started.
#
bgpd=yes
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
pimd=no
ldpd=no
nhrpd=no
eigrpd=no
babeld=no
sharpd=no
pbrd=no
bfdd=no
fabricd=no
vrrpd=no
```

Common File Locations and Commands

- Systemd Integration
  - /lib/systemd/system/frr.service
    - `systemctl enable frr`
    - `systemctl disable frr`
    - `systemctl start frr`
    - `systemctl restart frr`
    - `systemctl stop frr`
    - `systemctl reload frr`

```
[sharpd@robot bgpd]$ sudo systemctl disable frr
[sudo] password for sharpd:

Removed /etc/systemd/system/network-online.target.wants/frr.service.

[sharpd@robot bgpd]$ sudo systemctl enable frr
Created symlink /etc/systemd/system/multi-user.target.wants/frr.service →
/usr/lib/systemd/system/frr.service.

[sharpd@robot bgpd]$ sudo systemctl start frr
[sharpd@robot bgpd]$ ps -ef | grep frr
root      1117     1  0 17:31 ?        00:00:11 /usr/lib/frr/watchfrr -d zebra
bgpd
frr       1228     1  0 17:31 ?        00:00:14 /usr/lib/frr/zebra -d -A
127.0.0.1 -s 90000000
frr       1285     1  0 17:31 ?        00:00:23 /usr/lib/frr/bgpd -d -A


[sharpd@robot bgpd]$ sudo systemctl restart frr
[sharpd@robot bgpd]$ sudo systemctl reload frr
[sharpd@robot bgpd]$ sudo systemctl stop frr
[sharpd@robot bgpd]$
```

Common File Locations and Commands

- Interactive CLI
  - vtysh
    - /usr/bin/vtysh
- Logging (default is syslog)
  - /var/log/frr/frr.log
    - `show logging`

```
[sharpd@robot bgpd]$ sudo vtysh

Hello, this is FRRouting (version 7.3-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

robot# show logging
Logging configuration for zebra:
Syslog logging: disabled
Stdout logging: disabled
Monitor logging: level debugging
File logging: level debugging, filename /var/log/frr/frr.log
Protocol name: ZEBRA
Record priority: disabled
Timestamp precision: 6

Logging configuration for bgpd:
Syslog logging: disabled
Stdout logging: disabled
Monitor logging: level debugging
File logging: level debugging, filename /var/log/frr/frr.log
Protocol name: BGP
Record priority: disabled
Timestamp precision: 6
```

Thanks!