

RIFT

Routing in Fat Tree (RIFT) is a routing protocol with no operational expenses, designed for packet routing in CLOS-based and Fat Tree network topologies. This protocol blends link-state and distance-vector methods, offering multiple advantages for IP fabrics, including simplified management and enhanced network resilience.

riftd is based on [draft-ietf-rift-rift-19](#).

Starting and Stopping riftd

The default configuration file name of *riftd*'s is *riftd.conf*. When invocation *riftd* searches directory */etc/frr*. If *riftd.conf* is not there next search current directory.

RIFT uses UDP ports 914 and 915 to send and receive LIE/TIE packets. So the user must have the capability to bind the port, generally this means that the user must have superuser privileges. RIFT requires interface information maintained by *zebra* daemon. So running *zebra* is mandatory to run *riftd*. Thus minimum sequence for running RIFT is like below:

```
# zebra -d
# riftd -d
```

Please note that *zebra* must be invoked before *riftd*.

To stop *riftd*. Please use: `kill cat /var/run/frr/riftd.pid`

RIFT Configuration

```
router rift
```

The `router rift` command is necessary to enable RIFT. To disable RIFT, use the `no router rift` command. RIFT must be enabled before carrying out any of the RIFT commands.

```
system-id (1-4294967295)
```

Set RIFT's system ID. System ID is optional. If not specified, RIFT daemon will generate a unique one.

```
no system-id (1-4294967295)
```

Unset RIFT's system ID.

```
level <1-20|leaf|ew|tof>
```

Clos and Fat Tree networks are topologically partially ordered graphs and `level` denotes the set of nodes at the same height in such a network. `leaf` means that the node is at the last level of the network. `ew` describes an East-West link. `tof` is a node at the top of the network (maximum level number). If not specified, the level is auto-negotiated by RIFT with southbound and northbound neighbours.

```
no level <1-20|leaf|ew|tof>
```

Unset the node level value.

```
lie-address A.B.C.D
```

This command overrides the default IPv4 multicast address used to send LIE packets.

```
no lie-address A.B.C.D
```

This command removes the IPv4 multicast address configured to send LIE packets (fallbacks to default one).

```
lie-address X:X::X:X
```

This command overrides the default IPv6 multicast address used to send LIE packets.

```
no lie-address X:X::X:X
```

This command removes the IPv6 multicast address configured to send LIE packets (fallbacks to default one).

```
interface IFNAME [active-key (0-255)]
```

This command enables RIFT on the specified interface. Optionally, an active key can be passed to validate incoming packets. By default, it accepts all the packets without checking any key.

By default, RIFT is not enabled on all interfaces, you have to specify the interface command in order to enable it.

```
no interface IFNAME
```

This command disables RIFT on the specified interface.

Below is very simple RIFT configuration for a leaf node that uses LIE multicast address 224.0.0.150 and enables RIFT on interface eth1.

```
router rift
  system-id 1234
  level leaf
  lie address 224.0.0.150
  interface eth1
```

How to Announce RIFT route

```
redistribute <babel|bgp|connected|eigrp|isis|kernel|openfabric|ospf|sharp|
  static|table> [metric (0-16)] [route-map WORD]
```

Redistribute routes from other sources into RIFT.

```
no redistribute <babel|bgp|connected|eigrp|isis|kernel|openfabric|ospf|sharp|
  static|table> [metric (0-16)] [route-map WORD]
```

Remove the routes redistribution into RIFT from the specified source.

If you want to specify RIFT static prefixes:

```
prefix <A.B.C.D/M|X:X::X:X/M>
```

Specify either a IPv4 or IPv6 prefix.

```
no prefix <A.B.C.D/M|X:X::X:X/M>
```

Unset a previously specified IPv4 or IPv6 prefix.

Below is very simple RIFT configuration for a leaf node that enables RIFT on interface eth3 and announces a IPv6 prefix ecaf::/64.

```
router rift
  system-id 58964
  level leaf
```

```
interface eth3
prefix ecaf::/64
```

RIFT route-map

Usage of *riftd*'s route-map support.

Optional argument route-map MAP_NAME can be added to each redistribute statement. You have to specify the direction (either northbound or southbound).

```
redistribute static [route-map MAP_NAME direction <northbound|southbound>]
redistribute connected [route-map MAP_NAME direction <northbound|southbound>]
.....
```

Route-map statement ([Route Maps](#)) is needed to use route-map functionality.

Show RIFT Information

To display RIFT routes.

```
show rift
```

Show RIFT routes.

```
show rift tie-db [direction <northbound|southbound>]
```

The command display the content of the Topology Information Element Database (TIE-DB). If the direction is provided, only shows the TIE-DB in that direction.

```
show rift disaggregation
```

The command displays all information related to automatic disaggregation (positive and negative).

Sample configuration

```
ip prefix-list DEF_ONLY_4 permit 0.0.0.0/0
ip prefix-list DEF_ONLY_4 deny any

route-map FILTER_DEFAULT_V4 deny 10
  match ip address DEF_ONLY_4

router rift
  system-id 1337
  level tof
  interface eth0
  interface eth1
  interface eth2
  redistribute connected
  redistribute static route-map FILTER_DEFAULT_V4 direction northbound
  prefix 2001::/64
  prefix 200.0.0.0/24
```