Bidirectional Forwarding Detection - BFD

Bidirectional Forwarding Detection (BFD) provides low overhead and rapid detection of failures in the paths between two network devices. It provides a unified mechanism for link detection over all media and protocol layers. Use BFD to detect failures for IPv4 and IPv6 single or multihop paths between any two network devices, including unidirectional path failure detection.

> Cumulus Linux does not support demand mode in BFD.

Contents

This topic describes ...

- BFD Multihop Routed Paths
- BFD Parameters
- Configure BFD
- BFD in BGP
- BFD in OSPF
- OSPF Show Commands
- Scripts
- Echo Function
  - About the Echo Packet
  - Transmit and Receive Echo Packets
  - Echo Function Parameters
- Troubleshooting
- Related Information

BFD Multihop Routed Paths

BFD multihop sessions are built over arbitrary paths between two systems, which results in some complexity that does not exist for single hop sessions. Here are some best practices for using multihop paths:

- **Spoofing:** To avoid spoofing with multihop paths, configure `max_hop_cnt` (maximum hop count*)* for each peer, which limits the number of hops for a BFD session. All BFD packets exceeding the max hop count will be dropped.
- **Demultiplexing:** Since multihop BFD sessions can take arbitrary paths, demultiplex the initial BFD packet based on the source/destination IP address pair. Use FRRouting, which monitors connectivity to the peer, to determine the source/destination IP address pairs.

Multihop BFD sessions are supported for both IPv4 and IPv6 peers. See below for more details.

BFD Parameters

You can configure the following BFD parameters for both IPv4 and IPv6 sessions:

- The required minimum interval between the received BFD control packets.
- The minimum interval for transmitting BFD control packets.
- The detection time multiplier.

Configure BFD

You configure BFD one of two ways: by specifying the configuration in the PTM `topology.dot` file, or using FRRouting. However, the topology file has some limitations:

- The `topology.dot` file supports creating BFD IPv4 and IPv6 single hop sessions only; you cannot specify IPv4 or IPv6 multihop sessions in the topology file.
- The topology file supports BFD sessions for only link-local IPv6 peers; BFD sessions for global IPv6 peers discovered on the link will not be created.

> You cannot specify BFD multihop sessions in the `topology.dot` file since you cannot specify the source and destination IP address pairs in that file. Use FRRouting to configure multihop sessions.

The FRRouting CLI can track IPv4 and IPv6 peer connectivity — both single hop and multihop, and both link-local IPv6 peers and global IPv6 peers — using BFD sessions without needing the `topology.dot` file. Use FRRouting to register multihop peers with PTM and BFD as well as for monitoring the connectivity to the remote BGP multihop peer. FRRouting can dynamically register and unregister both IPv4 and IPv6 peers with BFD when the BFD-enabled peer connectivity is established or de-established, respectively. Also, you can configure BFD parameters for each BGP or OSPF peer using FRRouting.

> The BFD parameter configured in the topology file is given higher precedence over the client-configured BFD parameters for a BFD session that has been created by both topology file and client (FRRouting).

> BFD requires an IP address for any interface on which it is configured. The neighbor IP address for a single hop BFD session must be in the ARP table before BFD can start sending control packets.

BFD in BGP

For FRRouting when using **BGP**, neighbors are registered and de-registered with PTM dynamically when you enable BFD in BGP using `net add bgp neighbor <neighbor|IP|interface> bfd`. For example:

Configuration of BFD for a peergroup or individual neighbors is performed in the same way.

```
cumulus@switch:~$ net add bgp neighbor swp1 bfd
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

These commands add the `neighbor SPINE bfd` line below the last address family configuration in the `/etc/frr/frr.conf` file:

```
...

router bgp 65000
 neighbor swp1 bfd

...
```

The configuration above configures the default BFD values of intervals: 3, minimum RX interval: 300ms, minimum TX interval: 300ms.

To see neighbor information in BGP, including BFD status, run `net show bgp neighbor <interface>`.

```
cumulus@spine01:~$ net show bgp neighbor swp1
 ...

  BFD: Type: single hop
    Detect Mul: 3, Min Rx interval: 300, Min Tx interval: 300
    Status: Down, Last update: 0:00:00:08
 ...
```

To change the BFD values to something other than the defaults, BFD parameters can be configured for each BGP neighbor. For example:

BFD in BGP

```
cumulus@switch:~$ net add bgp neighbor swp1 bfd 4 400 400
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

BFD in OSPF

For FRRouting using **OSFP**, neighbors are registered and de-registered dynamically with PTM when you enable or disable BFD in OSPF. A neighbor is registered with BFD when two-way adjacency is established and deregistered when adjacency goes down if the BFD is enabled on the interface. The BFD configuration is per interface and any IPv4 and IPv6 neighbors discovered on that interface inherit the configuration.

BFD in OSPF

```
cumulus@switch:~$ net add interface swp1 ospf6 bfd 5 500 500
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

These commands create the following configuration snippet in the `/etc/frr/frr.conf` file:

```
interface swp1
 ipv6 ospf6 bfd 5 500 500
end
```

OSPF Show Commands

The BFD lines at the end of each code block shows the corresponding IPv6 or IPv4 OSPF interface or neighbor information.

Show IPv6 OSPF Interface

```
cumulus@switch:~$ net show ospf6 interface swp2s0
swp2s0 is up, type BROADCAST
  Interface ID: 4
  Internet Address:
    inet : 11.0.0.21/30
    inet6: fe80::4638:39ff:fe00:6c8e/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 10
  State PointToPoint, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
   Hello 10, Dead 40, Retransmit 5
  DR: 0.0.0.0 BDR: 0.0.0.0
  Number of I/F scoped LSAs is 2
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
  BFD: Detect Mul: 3, Min Rx interval: 300, Min Tx interval: 300
```

Show IPv6 OSPF Neighbor

```
cumulus@switch:~$ net show ospf6 neighbor detail
 Neighbor 0.0.0.4%swp2s0
    Area 0.0.0.0 via interface swp2s0 (ifindex 4)
    His IfIndex: 3 Link-local address: fe80::202:ff:fe00:a
    State Full for a duration of 02:32:33
    His choice of DR/BDR 0.0.0.0/0.0.0.0, Priority 1
    DbDesc status: Slave SeqNum: 0x76000000
    Summary-List: 0 LSAs
    Request-List: 0 LSAs
    Retrans-List: 0 LSAs
    0 Pending LSAs for DbDesc in Time 00:00:00 [thread off]
    0 Pending LSAs for LSReq in Time 00:00:00 [thread off]
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
    BFD: Type: single hop
      Detect Mul: 3, Min Rx interval: 300, Min Tx interval: 300
      Status: Up, Last update: 0:00:00:20
```

Show IPv4 OSPF Interface

```
cumulus@switch:~$ net show ospf interface swp2s0
swp2s0 is up
  ifindex 4, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 11.0.0.21/30, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.3, Network Type POINTOPOINT, Cost: 10
  Transmit Delay is 1 sec, State Point-To-Point, Priority 1
  No designated router on this network
  No backup designated router on this network
  Multicast group memberships: OSPFAllRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 7.056s
  Neighbor Count is 1, Adjacent neighbor count is 1
  BFD: Detect Mul: 5, Min Rx interval: 500, Min Tx interval: 500
```

Show IPv4 OSPF Neighbor

```
cumulus@switch:~$ net show ospf neighbor detail
 Neighbor 0.0.0.4, interface address 11.0.0.22
    In the area 0.0.0.0 via interface swp2s0
    Neighbor priority is 1, State is Full, 5 state changes
    Most recent state change statistics:
      Progressive change 3h59m04s ago
    DR is 0.0.0.0, BDR is 0.0.0.0
    Options 2 *|-|-|-|-|-|E|*
    Dead timer due in 38.501s
    Database Summary List 0
    Link State Request List 0
    Link State Retransmission List 0
    Thread Inactivity Timer on
    Thread Database Description Retransmision off
    Thread Link State Request Retransmission on
    Thread Link State Update Retransmission on
    BFD: Type: single hop
      Detect Mul: 5, Min Rx interval: 500, Min Tx interval: 500
      Status: Down, Last update: 0:00:01:29
```

Scripts

`ptmd` executes scripts at `/etc/ptm.d/bfd-sess-down` and `/etc/ptm.d/bfd-sess-up` for when BFD sessions go down or up, running `bfd-sess-down` when a BFD session goes down and running `bfd-sess-up` when a BFD session goes up.

You should modify these default scripts as needed.

Echo Function

Cumulus Linux supports the *echo function* for IPv4 single hops only, and with the asynchronous operating mode only (Cumulus Linux does not

support demand mode).

You use the echo function primarily to test the forwarding path on a remote system. To enable the echo function, set `echoSupport` to *1* in the topology file.

Once the echo packets are looped by the remote system, the BFD control packets can be sent at a much lower rate. You configure this lower rate by setting the `slowMinTx` parameter in the topology file to a non-zero value of milliseconds.

You can use more aggressive detection times for echo packets since the round-trip time is reduced because they are accessing the forwarding path. You configure the detection interval by setting the `echoMinRx` parameter in the topology file to a non-zero value of milliseconds; the minimum setting is 50 milliseconds. Once configured, BFD control packets are sent out at this required minimum echo Rx interval. This indicates to the peer that the local system can loop back the echo packets. Echo packets are transmitted if the peer supports receiving echo packets.

About the Echo Packet

BFD echo packets are encapsulated into UDP packets over destination and source UDP port number 3785. The BFD echo packet format is vendor-specific and has not been defined in the RFC. BFD echo packets that originate from Cumulus Linux are 8 bytes long and have the following format:

0	1	2	3
Version	Length	Reserved	
My Discriminator			

Where:

- **Version** is the version of the BFD echo packet.
- **Length** is the length of the BFD echo packet.
- **My Discriminator** is a non-zero value that uniquely identifies a BFD session on the transmitting side. When the originating node receives the packet after being looped back by the receiving system, this value uniquely identifies the BFD session.

Transmit and Receive Echo Packets

BFD echo packets are transmitted for a BFD session only when the peer has advertised a non-zero value for the required minimum echo Rx interval (the `echoMinRx` setting) in the BFD control packet when the BFD session starts. The transmit rate of the echo packets is based on the peer advertised echo receive value in the control packet.

BFD echo packets are looped back to the originating node for a BFD session only if locally the `echoMinRx` and `echoSupport` are configured to a non-zero values.

Echo Function Parameters

You configure the echo function by setting the following parameters in the topology file at the global, template and port level:

- **echoSupport:** Enables and disables echo mode. Set to 1 to enable the echo function. It defaults to 0 (disable).
- **echoMinRx:** The minimum interval between echo packets the local system is capable of receiving. This is advertised in the BFD control packet. When the echo function is enabled, it defaults to 50. If you disable the echo function, this parameter is automatically set to 0, which indicates the port or the node cannot process or receive echo packets.
- **slowMinTx:** The minimum interval between transmitting BFD control packets when the echo packets are being exchanged.

Troubleshooting

You can use the following commands to view information about active BFD sessions.

To return information on active BFD sessions, use the `net show bfd sessions` command:

```
cumulus@switch:~$ net show bfd sessions

-------------------------------------------------------------
port  peer        state  local         type       diag

-------------------------------------------------------------
swp1  11.0.0.2    Up     N/A           singlehop  N/A
N/A   12.12.12.1  Up     12.12.12.4    multihop   N/A
```

To return more **detailed** information on active BFD sessions, use the `net show bfd sessions detail` command (results are for an IPv6-connected peer):

```
cumulus@switch:~$ net show bfd sessions detail

-------------------------------------------------------------------------------
--------------
port  peer                 state  local  type       diag  det   tx_timeout
rx_timeout
                                                      mult

-------------------------------------------------------------------------------
--------------
swp1  fe80::202:ff:fe00:1  Up     N/A    singlehop  N/A   3     300
900
swp1  3101:abc:bcad::2     Up     N/A    singlehop  N/A   3     300
900

#continuation of output
-------------------------------------------------------------------------
echo        echo        max      rx_ctrl  tx_ctrl  rx_echo  tx_echo
tx_timeout  rx_timeout  hop_cnt
-------------------------------------------------------------------------
0           0           N/A      187172   185986   0        0
0           0           N/A      501      533      0        0
```

Related Information

- RFC 5880 - Bidirectional Forwarding Detection
- RFC 5881 - BFD for IPv4 and IPv6 (Single Hop)
- RFC 5882 - Generic Application of BFD
- RFC 5883 - Bidirectional Forwarding Detection (BFD) for Multihop Paths