Implementing BFD

Bidirectional forwarding detection (BFD) provides low-overhead, short-duration detection of failures in the path between adjacent forwarding engines. BFD allows a single mechanism to be used for failure detection over any media and at any protocol layer, with a wide range of detection times and overhead. The fast detection of failures provides immediate reaction to failure in the event of a failed link or neighbor.

Cisco NCS 5500 Router supports BFD with VRF context.

Note On the Cisco NCS 5500 Router, BFD supports up to 4 different minimum interval timers if BFD over Bundle (BoB) is configured, and up to 5 different minimum interval timers if BoB is not configured.

Tip You can programmatically configure BFD and retrieve operational data using `openconfig-bfd.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide*.

BFD Overview

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
BFD Support for VRRP	Release 7.5.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in native and compatible modes. This feature introduces BFD support over VRRP interfaces.

Bidirectional forwarding detection (BFD) provides low-overhead, short-duration detection of failures in the path between adjacent routers. BFD allows a single mechanism to be used for failure detection over any media and at any protocol layer, with a wide range of detection times and overhead. The fast detection of failures provides immediate reaction to failure in the event of a failed link or neighbor.

The router supports BFD with VRF context.

Restrictions

These restrictions apply to BFD:

- Demand mode is not supported in Cisco IOS XR software.
- BFD echo mode and encryption are not supported.
- BFD hardware offload for IPv4 is supported.
- Only the static, OSPF, BGP and IS-IS applications are supported on BFD.
- BFD dampening for IPv4 is supported starting from Cisco IOS XR Release 6.3.2.
- BFD multihop is supported over an IP core starting from IOS XR Release 6.3.2.

  BFD multihop is supported starting from IOS XR Release 6.3.2 but BFD multihop over non IP core is not supported.
- BFD supports BFDv6 on bundle-ether for VRF BGP IPv6 single-hop.
- BFD multihop over non-IP core (Label Distribution Protocol or Segment Routing) is supported starting from IOS XR Release 7.1.1.
- Only IETF mode is supported in BFD over bundle feature.
- Dampening extensions for BFD are not supported.

Starting from Cisco IOS XR Release 6.6.1, BFD over VRF is supported.

Starting from Cisco IOS XR Release 7.1.1, BFD over BVI is supported on fixed NCS5500 platforms.

Starting from Cisco IOS XR Release 7.2.1, BFD support over VRRP interface is supported.

SNMP traps are not supported for multipath BFD sessions.

BFD Timers

Note The router uses six unique timer profiles without BFD over Bundle (BoB) configuration. Up to five timer profiles are available, when you configure BoB.

The following example shows configuration of four profiles with minimum-interval timer.

- 150ms *3
- 450ms *3
- 200ms *3
- 750ms *3

Table 2: IPv4 BFD Timers

Type of BFD Session	Minimum Timer Supported	Default/Minimum-interval timer (Multipliers)	Supported Timer Profiles (Up to 6 unique timers profiles)
Single Hop	4ms	3	Any
BFD over Bundle Members (BoB)	4ms	3	Any
BFD over Logical bundle (BLB)	50ms	3	Any
BGP Multi Hop	50ms	3	Any

Table 3: IPv6 BFD Timers

Type of BFD Session	Minimum Timer Supported	Default/Minimum-interval timer (Multipliers)	Supported Timer Profile (Up to 6 unique timer profiles)	Maximum Scale depending on Minimum Interval
Single Hop	4ms	3	Any	150 (with 8ms and above, all 256 sessions are configurable)
BFD over Bundle Members (BoB)	4ms	3	Any	150ms (with 8ms and above, all 256 sessions are configurable)
BFD over Logical bundle (BLB)	50ms	3	Any	256

Type of BFD Session	Minimum Timer Supported	Default/Minimum-interval timer (Multipliers)	Supported Timer Profile (Up to 6 unique timer profiles)	Maximum Scale depending on Minimum Interval
BGP Multi Hop	50ms	3	Any	256

Enabling BFD on a Static Route

The following procedure describes how to enable BFD on a static route.

```
RP/0/RP0/CPU0:router(config)# configure

/* Enter static route configuration mode, and configure static routing. */
Router(config)# router static

/* Enter address family configuration mode. */
Router(config-static)# address-family ipv4 unicast

/* Specify an unicast destination address and next-hop IPv4 address.
Enable BFD fast-detection on the specified IPv4 unicast destination address */
Router(config-static)# 192.168.2.2/32 HundredGigE0/0/0/2 192.168.6.2 bfd fast-detect
minimum-interval 4 multiplier 3
```

Configuration example of a BFD single-hop scenario:

```
!
router
 address-family ipv4 unicast
  10.2.150.193/32 BVI1252 10.2.153.1 bfd fast-detect
```

Note The next-hop IPv4 address (10.2.153.1) is determined from the IP address of the directly connected interface.

Configuration example of a BFD multi-hop scenario:

```
!
router static
address-family ipv4 unicast
 10.10.10.0/24 10.20.20.20 bfd fast-detect multihop 10.30.30.30
```

Note In a BFD multi-hop scenario:

- The next-hop (10.20.20.20) must be reachable.
- The next-hop must be resolved through the connected interface if the interface is provided as part of the configuration. If the interface is provided, the 'multihop' option will not be available.

Configuration example of BFD on directly-connected host routes (/32 or /128):

```
!
router static
```

```
address-family ipv4 unicast
 10.102.134.140/32 TenGigE0/0/0/1 10.102.134.140 bfd fast-detect
```

Note In this BFD single-hop specific scenario:

- The prefix and next-hop are the same (this makes nexthop a static nexthop). The prefix length is either 32 or 128.
- Next-hop (10.102.134.140) is resolved using static routing.

Running Configuration

```
router static
 address-family ipv4 unicast
  192.168.2.2/32 HundredGigE0/0/0/2 192.168.6.2 bfd fast-detect minimum-interval 4 multiplier
 3
 !
!
```

Enabling BFD for OSPF on an Interface

The following procedures describe how to configure BFD for Open Shortest Path First (OSPF) on an interface. The steps in the procedure are common to the steps for configuring BFD on IS-IS ; only the command mode differs.

SUMMARY STEPS

1. **configure**
2. **router ospf** *process-name*
3. **area** *area-id*
4. **interface** *type interface-path-id*
5. **bfd fast-detect**
6. **bfd minimum-interval** *milliseconds*
7. **bfd multiplier** *multiplier*
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospf** *process-name* **Example:** `RP/0/RP0/CPU0:router(config)# router ospf 0`	Enters OSPF configuration mode, allowing you to configure the OSPF routing process. **Note** To configure BFD for IS-IS, enter the corresponding configuration mode.

	Command or Action	Purpose
Step 3	**area** *area-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf)# area 0`	Configures an Open Shortest Path First (OSPF) area. Replace *area-id* with the OSPF area identifier.
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar)# interface TengigabitEthernet 0/3/0/1`	Enters interface configuration mode and specifies the interface name.
Step 5	**bfd fast-detect** **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar-if)# bfd fast-detect`	Enables BFD to detect failures in the path between adjacent routers.
Step 6	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar-if)# bfd minimum-interval 6500`	Sets the BFD minimum interval. Range is 4-30000 milliseconds. This example sets the BFD minimum interval to 6500 milliseconds.
Step 7	**bfd multiplier** *multiplier* **Example:** `RP/0/RP0/CPU0:router(config-ospf-ar-if)# bfd multiplier 7`	Sets the BFD multiplier. This is optional, the minimum is 3 and by default the multiplier will be 3 for all protocols. This example sets the BFD multiplier to 7.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Enable BFD for IS-IS on an Interface

Perform the following steps to configure BFD for Integrated Intermediate System-to-Intermediate System (IS-IS) on an interface.

Note BFD per interface configuration is supported for OSPF and IS-IS only.

```
Router# configure

/* Enter IS-IS configuration mode to configure the IS-IS routing process. */
Router(config)# router isis 65444

/* Set the system type (area or backbone router). Each IS-IS instance can support either a
 single Level 1 or Level 2, or one of each.*/
Router(config-isis)#is-type level-2-only

/* Specify a NET for each routing instance if you are configuring multi-instance IS-IS.*/
Router(config-isis)# net 49.0001.0840.3803.4088.00

/* Enter interface configuration mode. */
Router(config-isis)# interface gigabitEthernet 0/3/0/1

/* Set the BFD minimum interval. */
Router(config-isis-if)# bfd minimum-interval 6500

/* Set the BFD multiplier.  */
Router(config-isis-if)# bfd multiplier 7

/* Enable BFD to detect failures in the path between adjacent forwarding engines. Only IPv4
 is supported.*/
Router(config-isis-if)# bfd fast-detect ipv4

/* Specify the IPv4 address family and enters router address family configuration mode. */
Router(config-isis-if)# address-family ipv4 unicast

!
```

Running Configuration

```
configure
  router isis 65444
   is-type level-2-only
   net 49.0001.0840.3803.4088.00
    interface gigabitEthernet 0/3/0/1
     bfd minimum-interval 6500
     bfd multiplier 7
     bfd fast-detect ipv4
     address-family ipv4 unicast
  !
!
```

Enabling BFD on a BGP Neighbor

BFD can be enabled per neighbor, or per interface. This task describes how to enable BFD for BGP on a neighbor router.

SUMMARY STEPS

1. **configure**
2. **router bgp** *autonomous-system-number*

3. **neighbor** *ip-address*
4. **remote-as** *autonomous-system-number*
5. **bfd fast-detect**
6. **bfd minimum-interval** *milliseconds*
7. **bfd multiplier** *multiplier*
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 120`	Enters BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	**neighbor** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-bgp)# neighbor 172.168.40.24`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer. This example configures the IP address 172.168.40.24 as a BGP peer.
Step 4	**remote-as** *autonomous-system-number* **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002`	Creates a neighbor and assigns it a remote autonomous system. This example configures the remote autonomous system to be 2002.
Step 5	**bfd fast-detect** **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd fast-detect`	Enables BFD between the local networking devices and the neighbor whose IP address you configured to be a BGP peer in Step 3. In the example in Step 3, the IP address 172.168.40.24 was set up as the BGP peer. In this example, BFD is enabled between the local networking devices and the neighbor 172.168.40.24.
Step 6	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)#bfd minimum-interval 6500`	Sets the BFD minimum interval. Range is 4-30000 milliseconds.
Step 7	**bfd multiplier** *multiplier* **Example:**	Sets the BFD multiplier. This is optional, the minimum is 3 and by default the multiplier will be 3 for all protocols

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-bgp-nbr)#bfd multiplier 7`	
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Clear and Display BFD Counters

The following procedure describes how to display and clear BFD packet counters. You can clear packet counters for BFD sessions that are hosted on a specific node or on a specific interface.

```
RP/0/RP0/CPU0:router# show bfd counters all packet location 0/3/cpu0
RP/0/RP0/CPU0:router# clear bfd counters all packet location 0/3/cpu0
RP/0/RP0/CPU0:router# show bfd counters all packet location 0/3/cpu0
```

BFD over Bundle and BFD over Logical Bundle

Link Aggregation Control Protocol (LACP) allows a network device to negotiate an automatic bundling of links by sending LACP packets to their directly connected peer. LACP provides a keep-alive mechanism for the link members. While the default keep-alive is 30s, it is configurable to up to 1s. LACP can detect failures on a per-physical-member link. However, the LACP timers do not fulfill the criteria of current fast convergence requirements.

Differences between BFD over Bundle and BFD over Logical Bundle

BFD over Bundle (BoB) (RFC 7130) has a BFD session on each bundle member. The client is the bundle manager. If a BFD session goes down on a specific member link, the whole bundle interface goes down. That is, when the member link goes down, the number of available links falls below the required minimum. Hence the routing session is brought down.

BFD over Logical Bundle (BLB) (RFC 5880) treats a bundle interface with all its members as a single interface. BLB is a multipath (MP) single-hop session. If BLB is configured on a bundle there is only one single BFD session that is active. This implies that only one bundle member is being monitored by BFD at any given time. The client is one of the routing protocols. When BFD detects a failure, the client brings down the routing session.

The mode (BoB or BLB) is determined by how you configure BFD:

- You can enable BoB by configuring BFD under a Bundle-Ether interface.

- You can enable BLB by configuring BFD under a Bundle-Ether interface on a routing client.

Link Aggregation Control Protocol (LACP) allows a network device to negotiate an automatic bundling of links by sending LACP packets to their directly connected peer. LACP provides a keep-alive mechanism for the link members. While the default keep-alive is 30s, it is configurable to up to 1s. LACP can detect failures on a per-physical-member link. However, the LACP timers do not fulfill the criteria of current fast convergence requirements.

BFD over Bundle

BFD over Bundle

BFD Over Bundle (BoB) (RFC 7130) has a BFD session on each bundle member. BOB verifies the ability for each member link to be able to forward Layer 3 packets.

For BFD over Bundle, the BFD client is bundlemgr. When BFD detects a failure on a bundle member, bundlemgr removes that member from the bundle. If there are not enough members to keep the bundle up, then the main Bundle-Ether interface will go down so that all routing protocols running on the main bundle interface or a subinterface will detect an interface down.

BoB does not provide a true Layer 3 check and is not supported on subinterfaces. However, subinterfaces will go down at the same time as the main interface.

BoB implementation is a standard-based fast failure detection of link aggregation (LAG) member links that is interoperable between different platforms. NCS 5500 platforms only support the IETF mode.

Configure BFD Over Bundle

Perform the following tasks to configure the BOB feature:

- Enable BFD sessions on bundle members
- Specify the BFD destination address on a bundle
- Configure the minimum thresholds for maintaining an active bundle
- Configure BFD packet transmission intervals and failure detection times on a bundle

  Configure BFD over bundles IETF mode support on a per-bundle basis

```
/* Enable BFD sessions on bundle members */
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd address-family ipv4 fast-detect
Router(config-if)# bfd mode ietf

/* Specify the BFD destination address on a bundle */
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd address-family ipv4 destination 10.20.20.1

/* Configure the minimum thresholds for maintaining an active bundle */
Router(config)# interface Bundle-Ether 1
Router(config-if)# bundle minimum-active bandwidth 580000
Router(config-if)# bundle minimum-active links 2

/* Configure BFD packet transmission intervals and failure detection times on a bundle */
Router(config)# interface Bundle-Ether 1
```

```
Router(config-if)# bfd address-family ipv4 minimum-interval 2000
Router(config-if)# bfd address-family ipv4 multiplier 30

/* Configure BFD over bundles IETF mode support on a per-bundle basis */
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv4 fast-detect
```

Bidirectional Forwarding Detection over Logical Bundle

BFD over Logical Bundle

The BLB feature implements and deploys BFD over bundle interfaces based on RFC 5880. In the BLB, the bundle interface is a single interface, whereas, in BOB, BFD is implemented per member link. BLB is a multipath (MP) single-hop session so at least one line card must be configured under the **bfd multipath** command before a BLB session can come up. Because BFD treats the bundle as a single big interface, BLB requires limited knowledge of the bundle interfaces on which the sessions run. BLB requires information about IP addresses, interface types, and caps on bundle interfaces only. Information such as a list of bundle members, member states, and configured minimum or maximum bundle links are not required. In the case of BLB, the BFD client is not the bundle link but protocols running over the bundle link. In BLB, the BFD client is not bundlemgr but the protocols running over bundle link. BLB is supported on IPv4 address, IPv6 global address, and IPv6 link-local address. The current version of the software supports a total of 200 sessions (which includes BFD Single hop for physical and logical sub-interfaces; BFD over Bundle (BoB) and BLB) per line card. The maximum processing capability of BFD control packets, per line card, has also increased to 7000 pps.

Configuration Example

- Create VLAN subinterface under bundle interface
- Enable BFD on a static route
- Enable BFD on IS-IS
- Enable BFD for OSPF on an interface
- Enable BFD on a BGP neighbor
- Configure multipath capability under BFD

```
/* Create VLAN subinterface under bundle interface */
Router# configure
Router(config)# interface Bundle-Ether 2.1
Router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
Router(config-if)# encapsulation dot1q 1
Router(config-if)# end

/* Enable BFD on a static route. */
Router# configure
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static)# 10.158.3.13/32 10.1.1.2 bfd fast-detect minimum-interval 300 multiplier
 3

/* Enable BFD on IS-IS. */
```

```
Router# configure
Router(config)# router isis cybi
Router(config-isis)# interface Bundle-Ether 2.1
Router(config-isis-if)# bfd minimum-interval 300
Router(config-isis-if)# bfd multiplier 3
Router(config-isis-if)# bfd fast-detect ipv4
Router(config-isis-if)# address-family ipv4 unicast
Router(config-isis-if-af)# end

/* Enable BFD for OSPF on an interface. */
Router# configure
Router(config)# router ospf cybi
Router(config-ospf)# area 0
Router(config-ospf)# interface Bundle-Ether 2.1
Router(config-ospf-if)# bfd fast-detect
Router(config-ospf-if)# bfd minimum-interval 300
Router(config-ospf-if)# bfd multiplier 3
Router(config-ospf-if)# end

/* Enable BFD on a BGP neighbor.*/
Router# configure
Router(config)# router bgp 4787
Router(config-bgp)# neighbor 10.158.1.1
Router(config-bgp-nbr)# remote-as 4787
Router(config-bgp-nbr)# update-source Bundle-Ether 2.1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd minimum-interval 300
Router(config-bgp-nbr)# bfd multiplier 3
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy PASS-ALL in
Router(config-bgp-nbr-af)# route-policy PASS-ALL out
Router(config-bgp-nbr-af)# exit
Router(config-bgp-nbr)# commit

/* Configure a specific LC (or LCs) to host BLB sessions. The BLB sessions and bundle member
 links need not be configured on the same LC. For example, you can configure the bundle
member links on LC slot 2 and slot 3 while you configure BLB sessions to be hosted on LC
slot 5. */
Router(config)# bfd
Router(config-bfd)# multipath include location 0/6/CPU0
Router(config-bfd)# multipath include location 0/2/CPU0
```

Enabling BFD Sessions on Bundle Members

To enable BFD sessions on bundle member links, complete these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bfd address-family ipv4 fast-detect**
4. **bfd mode ietf**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd address-family ipv4 fast-detect** **Example:** `RP/0/RP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD sessions on bundle member links.
Step 4	**bfd mode ietf** **Example:** `RP/0/RP0/CPU0:router(config-if)# bfd mode ietf`	Enables IETF mode for BFD over bundle for the specified bundle.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Specifying the BFD Destination Address on a Bundle

To specify the BFD destination address on a bundle, complete these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bfd address-family ipv4 destination** *ip-address*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd address-family ipv4 destination** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-if)# bfd address-family ipv4 destination 10.20.20.1`	Specifies the primary IPv4 address assigned to the bundle interface on a connected remote system, where *ip-address* is the 32-bit IP address in dotted-decimal format (A.B.C.D).
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring the Minimum Thresholds for Maintaining an Active Bundle

The bundle manager uses two configurable minimum thresholds to determine whether a bundle can be brought up or remain up, or is down, based on the state of its member links.

- Minimum active number of links
- Minimum active bandwidth available

Whenever the state of a member changes, the bundle manager determines whether the number of active members or available bandwidth is less than the minimum. If so, then the bundle is placed, or remains, in DOWN state. Once the number of active links or available bandwidth reaches one of the minimum thresholds, then the bundle returns to the UP state.

To configure minimum bundle thresholds, complete these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bundle minimum-active bandwidth** *kbps*
4. **bundle minimum-active links** *links*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bundle minimum-active bandwidth** *kbps* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000`	Sets the minimum amount of bandwidth required before a bundle can be brought up or remain up. The range is from 1 through a number that varies depending on the platform and the bundle type.
Step 4	**bundle minimum-active links** *links* **Example:** `RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2`	Sets the number of active links required before a bundle can be brought up or remain up. The range is from 1 to 32. **Note** When BFD is started on a bundle that is already active, the BFD state of the bundle is declared when the BFD state of all the existing active members is known.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring BFD Packet Transmission Intervals and Failure Detection Times on a Bundle

BFD asynchronous packet intervals and failure detection times for BFD sessions on bundle member links are configured using a combination of the **bfd address-family ipv4 minimum-interval** and **bfd address-family ipv4 multiplier** interface configuration commands on a bundle.

The BFD control packet interval is configured directly using the **bfd address-family ipv4 minimum-interval** command. The failure detection times are determined by a combination of the interval and multiplier values in these commands.

To configure the minimum transmission interval and failure detection times for BFD asynchronous mode control packets on bundle member links, complete these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bfd address-family ipv4 minimum-interval** *milliseconds*
4. **bfd address-family ipv4 multiplier** *multiplier*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd address-family ipv4 minimum-interval** *milliseconds* **Example:** `RP/0/RP0/CPU0:router(config-if)#bfd address-family ipv4 minimum-interval 2000` **Note** Specifies the minimum interval, in milliseconds, for asynchronous mode control packets on IPv4 BFD sessions on bundle member links. The range is from 4 to 30000.	

	Command or Action	Purpose
Step 4	**bfd address-family ipv4 multiplier** *multiplier* **Example:** `RP/0/RP0/CPU0:router(config-if)#bfd address-family ipv4 multiplier 30`	Specifies a number that is used as a multiplier with the minimum interval to determine BFD control packet failure detection times and transmission intervals for IPv4 BFD sessions on bundle member links. The range is from 2 to 50. The default is 3. **Note** Although the command allows you to configure a minimum of 2, the supported minimum is 3.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configure BFD over Bundles IETF Mode Support on a Per Bundle Basis

To configure BFD over Bundles IETF mode support on a per bundle basis use these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bfd mode ietf**
4. **bfd address-family ipv4 fast-detect**
5. Use the **commit** or **end** command.
6. **show bundle bundle-ether** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd mode ietf** **Example:** `RP/0/RP0/CPU0:router(config-if)# bfd mode ietf`	Enables IETF mode for BFD over bundle for the specified bundle.
Step 4	**bfd address-family ipv4 fast-detect** **Example:** `RP/0/RP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD sessions on the specified bundle.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	**show bundle bundle-ether** *bundle-id*	Displays the selected bundle mode.

BFD over Bundle with IPv4 Unnumbered Interfaces

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
BFD for BoB with IPv4 Unnumbered	Release 7.3.1	This feature is now supported on routers that have Cisco NC57 line cards installed and operate in the native mode.

BFD over Bundle with IPv4 Unnumbered Interfaces feature enables BFD to run on IP unnumbered interfaces, which take the IP address from the loopback address. The same loopback address is used on multiple interfaces. This saves IP addresses space or range.

BFD creates a session on the unnumbered interface for which the BFD clients provide the source and destination IP address along with the interface index. BFD establishes the session on the Layer 3 unnumbered link to which the interface index corresponds. The source address is derived from the Loopback interface at the source. The destination node also uses IP unnumbered interface with loopback address and that is used as destination IP address.

BFD sends control packets to the unnumbered interfaces. These control packets are the regular IP BFD packets. Address Resolution Protocol (ARP) resolves the destination loopback IP address to the destination node's router MAC address.

Restriction

Only Asynchronous mode is supported.

Configure BFD over Bundle with IPv4 Unnumbered Interface

- Configure loopback address
- Add physical interface to bundle
- Configure BOB session on an unnumbered interface

Configure Loopback Address

```
Router(config)# interface loopback 1
Router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
```

Add Physical Interface to Bundle

```
Router(config)# interface HundredGigE0/0/1/0
Router(config-if)# bundle id 1 mode on
```

Configure a BFD over Bundle Session on an Unnumbered Interface

```
Router(config)# interface Bundle-Ether1
Router(config-if)# bfd address-family ipv4 destination 10.2.2.2
Router(config-if)# bfd address-family ipv4 fast-detect
Router(config-if)# ipv4 point-to-point
Router(config-if)# ipv4 unnumbered Loopback1
Router(config-if)# lldp
Router(config-if)# enable
```

Running Configuration

```
interface Loopback1
ipv4 address 10.1.1.1 255.255.255.0
!
interface HundredGigE0/0/1/0
bundle id 1 mode on
!
interface Bundle-Ether1
bfd address-family ipv4 destination 10.2.2.2
bfd address-family ipv4 fast-detect
ipv4 point-to-point
ipv4 unnumbered Loopback1
lldp
 enable
```

BFD Transparency

Bidirectional Forwarding Detection(BFD) protocol is a simple hello mechanism that detects failures in a network in less than one second, depending on the timer value configured.

Both endpoints of a BFD Session periodically send Hello packets to each other. If a number of those packets are not received, the session is considered down. BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols BGP, IS-IS, and OSPF.

BFD Transparency feature enables you to configure BFD Sessions between customer edge devices connected over an L2VPN network. These BFD sessions are transparent to the core. For example, BFD packets being exchanged between CEs are neither dropped on any router in the core, nor punted on any core device.

In this section, you will learn how to configure BFD Transparency in Ethernet VPN (EVPN) Virtual Private Wire Service (VPWS).

Ethernet VPN Virtual Private Wire Service

EVPN VPWS (Ethernet VPN Virtual Private Wire Service) is a BGP control plane solution for point-to-point services. It implements signaling and encapsulation techniques for establishing an EVPN instance between a pair of provider edge devices.

EVPN VPWS supports both single-homing and multi-homing.

Configuration

The following sections describes the procedure for configuring IP Fast Reroute with Remote LFA.

- Configure L2VPN on the provide edge router
- Configure BFD on the customer edge router

Configure L2VPN on the Provide Edge Router

Configure L2VPN on the provider edge router.

```
/* Enable IS-IS and configure routing level for an area. */
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)# interface tengige 0/1/0/8/2.1
RP/0//CPU0:router(config-subif)# exit
RP/0//CPU0:router(config)# router isis
RP/0//CPU0:router(config-isis)# is-type level-2-only
RP/0//CPU0:router(config-isis)# net 49.1234.2222.2222.2222.00
RP/0//CPU0:router(config-isis)# nsr
RP/0//CPU0:router(config-isis)# nsf cisco
RP/0//CPU0:router(config-isis)# address-family ipv4 unicast
RP/0//CPU0:router(config-isis-af)# metric style wide
RP/0//CPU0:router(config-isis)# end
RP/0//CPU0:router(config)# interface Bundle-Ether 199
RP/0//CPU0:router(config-if)# address-family ipv4 unicast
RP/0//CPU0:router(config-if)# end
RP/0//CPU0:router(config)# interface Loopback 0
RP/0//CPU0:router(config-if)# end
RP/0//CPU0:router(config-if)# address-family ipv4 unicast
RP/0//CPU0:router(config-if)# exit
```

```
/* Configure L2VPN EVPN address family. */
RP/0//CPU0:router(config)# router bgp 100
RP/0//CPU0:router(config-bgp)# bgp router-id 10.10.10.1
RP/0//CPU0:router(config-bgp)# address-family l2vpn evpn
RP/0//CPU0:router(config-bgp)# neighbor 192.0.2.1
RP/0//CPU0:router(config-bgp-nbr)# remote-as 100
RP/0//CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0//CPU0:router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP for the physical core interface. */
RP/0//CPU0:router(config-bgp-nbr-af)# mpls ldp
RP/0//CPU0:router(config-bgp-nbr-af)# exit
RP/0//CPU0:router(config-bgp-nbr)# exit
RP/0//CPU0:router(config-bgp)# exit
RP/0//CPU0:router(config)# interface Bundle-Ether 199
RP/0//CPU0:router(config-if)# exit

/* Configure L2VPN Xconnect. */
RP/0//CPU0:router(config)# l2vpn
RP/0//CPU0:router(config-l2vpn)# router-id 10.10.10.1
RP/0//CPU0:router(config-l2vpn)# xconnect group bfdtr
RP/0//CPU0:router(config-l2vpn-xc)# p2p vpws-ce
RP/0//CPU0:router(config-l2vpn-xc-p2p)# interface TenGigE 0/4/0/2/0.1
RP/0//CPU0:ios(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 3 source 4
```

Configure BFD on the Customer Edge Router

Configure BFD on the customer edge router.

```
RP/0//CPU0:router# configure
RP/0//CPU0:router(config)#  router bgp 100
RP/0//CPU0:router(config-bgp)# bgp router-id 10.10.10.1
RP/0//CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0//CPU0:router(config-bgp-af)# exit
RP/0//CPU0:router(config-bgp)# neighbor 172.16.0.1
RP/0//CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0//CPU0:router(config-bgp-nbr)# remote-as 100
RP/0//CPU0:router(config-bgp-nbr)# bfd fast-detect
RP/0//CPU0:router(config-bgp-nbr)# bfd multiplier 2
RP/0//CPU0:router(config-bgp-nbr)# bfd minimum-interval 100
RP/0//CPU0:router(config-bgp-nbr)# update-source TenGigE 0/0/0/16.1
RP/0//CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0//CPU0:router(config-bgp-nbr-af)#
```

Running Configuration

This section shows the BFD Transparency configuration.

```
!
interface TenGigE 0/4/0/2/0.1
  l2transport
router isis 1
   is-type level-2-only
   net 49.0000.1000.0000.0001.00
   nsr
   nsf cisco
   address-family ipv4 unicast
    metric-style wide
!
  interface Bundle-Ether199
```

```
     address-family ipv4 unicast
  interface Loopback0
      address-family ipv4 unicast
router bgp 100
  bgp router-id 10.10.10.1
   address-family l2vpn evpn
    neighbor 192.0.2.1
     remote-as 100
     update-source Loopback 0
    address-family l2vpn evpn
 !
   mpls ldp
   interface Bundle-Ether199
 !
  l2vpn
   router-id 10.10.10.1
    xconnect group bfdtr
   p2p vpws-ce
    interface TenGigE 0/4/0/2/0.1
     neighbor evpn evi 100 target 3 source 4

         router bgp 100
          bgp router-id 10.10.10.1
           address-family ipv4 unicast
           !
           neighbor 172.16.0.1
            address-family ipv4 unicast
            remote-as 100
            bfd fast-detect
            bfd multiplier 2
            bfd minimum-interval 100
            update-source TenGigE0/0/0/16.1
            address-family ipv4 unicast
```

Verification

The show outputs given in the following section display the details of the configuration of the BFD transparency, and the status of their configuration.

```
/* Verify if the BFD session is up, and the timers are configured. */
RP/0//CPU0:router# show bfd session

Thu Jan  4 03:07:15.529 UTC
Interface      Dest Addr  Local det time(int*mult)  State        Echo  Async   H/W      NPU

-------------- ---------- ------------------------  --------     ----  -----   ---
----
Te0/0/0/4.1    10.1.1.1   0s(0s*0)                 20ms(10ms*2) UP    Yes     0/RP0/CPU0
                                                     Yes   0/RP0/CPU0


/* Verify if the BFD session is up and check the timer value, numbers of hellos exchanged,
 and information
about last packet. */

RP/0//CPU0:router# show bfd session destination 10.1.1.1 detail
Thu Jan  4 03:09:48.573 UTC
I/f: TenGigE0/0/0/4.1, Location: 0/RP0/CPU0
Dest: 10.1.1.1
Src:  10.1.1.2
 State: UP for 0d:0h:9m:27s, number of times UP: 1
 Session type: PR/V4/SH
```

```
Received parameters:
 Version: 1, desired tx interval: 10 ms, required rx interval: 10 ms
Required echo rx interval: 0 ms, multiplier: 2, diag: None
 My discr: 2147483898, your discr: 2147483899, state UP, D/F/P/C/A: 0/0/0/1/0
Transmitted parameters:
 Version: 1, desired tx interval: 10 ms, required rx interval: 10 ms
 Required echo rx interval: 0 ms, multiplier: 2, diag: None
 My discr: 2147483899, your discr: 2147483898, state UP, D/F/P/C/A: 0/1/0/1/0
Timer Values:
 Local negotiated async tx interval: 10 ms
 Remote negotiated async tx interval: 10 ms
 Desired echo tx interval: 0 s, local negotiated echo tx interval: 0 ms
 Echo detection time: 0 ms(0 ms*2), async detection time: 20 ms(10 ms*2)
Local Stats:
 Intervals between async packets:
   Tx: Number of intervals=100, min=6 ms, max=6573 ms, avg=1506 ms
       Last packet transmitted 186 s ago
   Rx: Number of intervals=100, min=4 ms, max=5 s, avg=575 ms
       Last packet received 184 s ago
 Intervals between echo packets:
   Tx: Number of intervals=0, min=0 s, max=0 s, avg=0 s
       Last packet transmitted 0 s ago
   Rx: Number of intervals=0, min=0 s, max=0 s, avg=0 s
       Last packet received 0 s ago
 Latency of echo packets (time between tx and rx):
   Number of packets: 0, min=0 ms, max=0 ms, avg=0 ms
Session owner information:
                          Desired               Adjusted
  Client               Interval   Multiplier Interval   Multiplier
  -------------------- ---------------------- ---------------------
  bgp-default          10 ms      2          10 ms      2

H/W Offload Info:
 H/W Offload capability : Y, Hosted NPU     : 0//CPU0
 Async Offloaded        : Y, Echo Offloaded : N
 Async rx/tx            : 344/209

Platform Info:
NPU ID: 0
Async RTC ID        : 1          Echo RTC ID        : 0
Async Feature Mask  : 0x0        Echo Feature Mask  : 0x0
Async Session ID    : 0xfb       Echo Session ID    : 0x0
Async Tx Key        : 0x800000fb  Echo Tx Key        : 0x0
Async Tx Stats addr : 0x0   Echo Tx Stats addr : 0x0
Async Rx Stats addr : 0x0   Echo Rx Stats addr : 0x0

/* Verify the complete history including session state, type, transitions, offload history,
 last down reason if any,
 received and transmitted packets, rx/tx intervals, location, timestamp, and local and
remote descriptors. */

RP/0/RP0/CPU0:router# show bfd session status history destination 10.1.10.1 location
0/RP0/CPU0

Thu Jan  4 03:45:18.768 UTC
I/f: TenGigE0/0/0/4.10, Location: 0//CPU0 table_id:0xe0000000
State: UP, flags:0x80040
Iftype: 0x19, basecaps: 107
Async dest addr: 10.1.10.1
Async src addr: 10.1.10.2
Echo dest addr: 10.1.10.2
Echo src addr: 192.0.2.1
Additional info from Flags:
 FIB is READY
```

```
 Session Active on 0/RP0/CPU0
Platform Info: 0x0, Mac Length: 18
Redundancy session info:
 Created from active BFD server
Last Down Diag: None
Last UP Time: Jan  4 03:00:19.272

Received parameters:
 Version: 1, desired tx interval: 10 ms, required rx interval: 10 ms
 Required echo rx interval: 0 ms, multiplier: 2, diag: None
 My discr: 2147483747, your discr: 2147483751, state UP, D/F/P/C/A: 0/0/0/1/0

Transmitted parameters:
 Version: 1, desired tx interval: 10 ms, required rx interval: 10 ms
 Required echo rx interval: 0 ms, multiplier: 2, diag: None
 My discr: 2147483751, your discr: 2147483747, state UP, D/F/P/C/A: 0/1/0/1/0

Tx Echo pkt :
 Version: 0, Local Discr: 2147483751, Sequence No: 0

History:
[Jan  4 03:00:19.272] Session (v1) state change, triggered by event 'Remote
    state init', from INIT to UP with current diag being None
[Jan  4 03:00:16.851] Session (v1) state change, triggered by event 'Remote
    state down', from DOWN to INIT with current diag being None
[Jan  4 03:00:16.509] Session (v1) state change, triggered by event 'Session
    create', from Unknown to DOWN with current diag being None
[Jan  4 03:00:16.509] Flag cleared: session creation is in-progress, currently
    set flags (0x80040)

Offload history:
[Jan  4 03:06:42.013] Packet punted to sw: Packet word0 : (0x20c80218),
desired_min_tx_interval 10000, required_min_rx_interval 10000, Last punted pkt
    required_min_rx_interval 10000
[Jan  4 03:06:42.003] Packet punted to sw: Packet word0 : (0x20d80218),
desired_min_tx_interval 10000, required_min_rx_interval 10000, Last punted pkt
    required_min_rx_interval 10000
[Jan  4 03:06:41.989] Packet punted to sw: Packet word0 : (0x20c80218),
desired_min_tx_interval 10000, required_min_rx_interval 10000, Last punted pkt
    required_min_rx_interval 10000
[Jan  4 03:06:41.980] Packet punted to sw: Packet word0 : (0x20d80218),
desired_min_tx_interval 10000, required_min_rx_interval 10000, Last punted pkt
    required_min_rx_interval 10000

Rx Counters and Timestamps :
Async valid packets received: count 5280
  [Jan  4 03:06:42.013]  [Jan  4 03:06:42.003]  [Jan  4 03:06:41.989]
Async valid packets while session is not in Up state: count 3
  [Jan  4 03:00:19.272]  [Jan  4 03:00:18.030]  [Jan  4 03:00:16.851]
```

BFD Hardware Offload Support for IPv4

The Bidirectional Forwarding detection (BFD) Hardware Offload feature enables the offload of a BFD session to the network processing units of the line cards, in an IPv4 network. BFD hardware offload improves scale and reduces the overall network convergence time by sending rapid failure detection packets to the routing protocols for recalculating the routing table.

Restrictions

- This feature is not supported over MPLS LDP interface and VRRP interface.
- This feature is not supported over MPLS TE or RSVP tunnel.
- BFD multihop will flap if underlay paths that consist of multiple bundle VLANs flap.

Confiugration Example

```
/* Configure BFD over Bundle(BOB) for hardware offload. */
Router# config
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv4 multiplier 3
Router (config-if)# bfd address-family ipv4 destination 10.20.20.1
Router (config-if)# bfd address-family ipv4 fast-detect
Router(config-if)# bfd address-family ipv4 minimum-interval 2000
Router(config-if)# ipv4 address 10.20.20.2/30

/* To define the line card to host BLB and BFD multihop sessions. */
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0

/* Configure BFD with a static route. */
Router(config)# router static
Router(config-static)# address-family ipv4 unicast 10.1.1.0/24 10.6.0.2 bfd fast-detect
minimum-interval 350 multiplier 4

/* Configure BFD with IS-IS. */
Router(config)# router isis 65444
Router(config-isis)# address-family ipv4 unicast
Router(config-isis)# exit
Router(config-isis)# interface gigabitEthernet 0/3/0/1
Router(config-isis-if)#  bfd minimum-interval 6500
Router(config-isis-if)# bfd multiplier 7
Router(config-isis-if)# bfd fast-detect ipv4
Router(config-isis-if)# address-family ipv4 unicast

/* Configure BFDv4 with OSPF. */
Router(config)# router ospf main
Router(config-ospfv3)# area 0
Router(config-ospfv3-ar)#  interface gigabitEthernet 1/0/0/1
Router(config-ospfv3-ar-if)# bfd multiplier 7
Router(config-ospfv3-ar-if)# bfd fast-detect
Router(config-ospfv3-ar-if)# bfd minimum-interval 6500

/* Configuring BFD over BGP. */
Router(config)# router bgp 120
Router(config-bgp)# neighbor 10.6.6.1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 7
Router(config-bgp-nbr)# bfd minimum-interval 6500
```

Verification

Use the **show bfd ipv4 session** command to verify the configuration:

```
Router# show bfd ipv4 session
Interface           Dest Addr           Local det time(int*mult)       State
                                      Echo             Async   H/W       NPU
------------------- --------------- ---------------- ---------------- ----------
```

```
Hu0/0/0/22.93       10.20.20.1       0s(0s*0)         12ms(4ms*3)      UP
                                                                 Yes   0/0/CPU0
```

BFD Hardware Offload Support for IPv6

The Bidirectional Forwarding detection (BFD) Hardware Offload feature enables the offload of a BFD session to the network processing units of the line cards, in an IPv6 network. BFD hardware offload feature improves scale and reduces the overall network convergence time by sending rapid failure detection packets to the routing protocols for recalculating the routing table.

Restrictions

- This feature is not supported over MPLS LDP interface and VRRP interface.
- This feature is not supported over MPLS TE or RSVP tunnel.
- BFD Dampening is not supported for BFD over IPv6.
- BFD over Bundle (BOB) over IPv6 is not supported with dynamically configured link-local address. It must be statically configured.
- BFD multihop will flap if underlay paths that consist of multiple bundle VLANs flap.

Configuration Example

```
/* Configure BFD over Bundle(BOB) for hardware offload. */
Router# config
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv6 multiplier 3
Router (config-if)# bfd address-family ipv6 destination 10.20:20::1
Router (config-if)# bfd address-family ipv6 fast-detect
Router(config-if)# bfd address-family ipv6 minimum-interval 2000
Router(config-if)# ipv6 address 10:20:20::2/64

/* To define the line card to host BLB and BFD multihop sessions. */
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0

/* Configure BFD with a static route. */
Router(config)# router static
Router(config-static)# address-family ipv6 unicast 1011:17e4::1/128 ab11:15d2::2 bfd
fast-detect minimum-interval 50 multiplier 3

/* Configure BFD with IS-IS. */
Router(config)# router isis 65444
Router(config-isis)# address-family ipv6 unicast
Router(config-isis)# exit
Router(config-isis)# interface gigabitEthernet 0/3/0/1
Router(config-isis-if)#  bfd minimum-interval 6500
Router(config-isis-if)# bfd multiplier 7
Router(config-isis-if)# bfd fast-detect ipv6
Router(config-isis-if)# address-family ipv6 unicast

/* Configure BFDv6 with OSPFv3. */
Router(config)# router ospfv3 main
Router(config-ospfv3)# area 0
Router(config-ospfv3-ar)#  interface gigabitEthernet 1/0/0/1
Router(config-ospfv3-ar-if)# bfd multiplier 7
```

```
Router(config-ospfv3-ar-if)# bfd fast-detect
Router(config-ospfv3-ar-if)# bfd minimum-interval 6500

/* Configuring BFD over BGP. */
Router(config)# router bgp 120
Router(config-bgp)# neighbor 2001:DB8:1::1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 7
Router(config-bgp-nbr)# bfd minimum-interval 6500
```

Verification

Use the **show bfd ipv6 session** command to verify the configuration:

```
Router# show bfd ipv6 session
Interface           Dest Addr
                                        Local det time(int*mult)      State
H/W                 NPU             Echo              Async
------------------- --------------- ----------------- ---------------- ----------
BE7.2               fe80::28a:96ff:fed6:9cdb
Yes                 0/0/CPU0        0s(0s*0)          900ms(300ms*3)   UP
BE7.4               fe80::28a:96ff:fed6:9cdb
Yes                 0/0/CPU0        0s(0s*0)          900ms(300ms*3)   UP
```

IPv4 Multihop BFD

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Multihop BFD for IPv4 nondefault VRF	Release 7.7.1	BFD provides fast forwarding path failure detection between two routing devices that are connected by a network link. BFD Multihop enables you to detect connectivity between routers that span multiple network hops and follow unpredictable paths. Prior to this release, BFD Multihop was supported on default VRFs only. This feature provides you the flexibility to extend BFD Multihop for IPv4 non-default VRFs.

IPv4 Multihop BFD is a BFD session between two addresses between two nodes. An example of this feature is a BFD session between PE and CE loopback addresses or BFD sessions between routers that are several TTL hops away. The applications that support IPv4 Multihop BFD are external and internal BGP. IPv4 Multihop BFD feature supports BFD on arbitrary paths, which can span multiple networks hops.

A Virtual Routing and Forwarding (VRF) instance is a logical separation of a router's routing table. VRF allows you to have multiple routing tables on a single router, each with its own set of routes.

The default VRF is the first VRF that is created on a router. It is the VRF that is used by default for all routing protocols and interfaces.

Non-default VRFs must be explicitly configured.

The IPv4 Multihop BFD feature provides subsecond forwarding failure detection for a destination more than one hop, and up to 255 hops, away. IPv4 Multihop BFD feature is supported on all currently supported media-type for BFD single hop.

You can set up a BFD multihop session between a unique source-destination address pair that is provided by the client. You can set up a session two endpoints that have IP connectivity.

Multihop BFD over nondefault VRF feature runs on both default and non-default VRF.

Multihop BFD over nondefault VRF feature runs on IPv4 only.

The IPv4 BFD Multihop feature enables you to configure IPv4 Multihop BFD on MPLS LDP and Segment Routing.

For information on configuring MPLS LDP, see the MPLS Configuration Guide for Cisco NCS 5500 Series Routers.

For information on configuring Segment Routing, see the Segment Routing Configuration Guide for Cisco NCS 5500 Series Routers.

Configure IPv4 Multihop BFD

This section describes how you can configure IPv4 Multihop BFD feature.

```
Router# configure
Router(config)# bfd
Router(config)# multipath include location 0/7/CPU0
Router(config)# router bgp 100
Router(config-bgp)# neighbor 209.165.200.225
Router(config-bgp-nbr)# remote-as 2000
Router(config-bgp-nbr)# update-source loopback 1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 3
Router(config-bgp-nbr)# bfd minimum-interval 300
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
```

Running Configuration

```
bfd
 multipath include location 0/7/CPU0
router bgp 100
 neighbor 209.165.200.225
  remote-as 2000
  update-source loopback 1
  bfd fast-detect
  bfd multiplier 3
  bfd minimum-interval 300
address-family ipv4 unicast
      route-policy PASS-ALL in
    route-policy PASS-ALL out
  !
!
```

Verification

The show outputs given in the following section display the details of the configuration of the IPv4 Multihop BFD feature, and the status of their configuration.

```
Router# show tech-support bfdhwoff location 0/7/CPU0 file
harddisk:
```

```
Tue Mar 20 11:20:29.214 PDT
++ Show tech start time: 2018-Mar-20.112029.PDT ++
Tue Mar 20 11:20:30 PDT 2018 Waiting for gathering to complete ..........................
Tue Mar 20 11:22:37 PDT 2018 Compressing show tech output Show tech output available at
0/RP0/CPU0 :
/harddisk:/showtech-bfd-hwoff-platform-2018-Mar-20.112029.PDT.tgz
++ Show tech end time: 2018-Mar-20.112237.PDT ++
```

Configure Multihop BFD on IPv4 nondefault VRFs

Configure the following steps to configure Multihop BFD on IPv4 nondefault VRF:

```
Router(config)# router bgp 100
Router(config-bgp)# neighbor 209.165.200.225
Router(config-bgp-nbr)#vrf vrf1
Router(config-bgp-nbr-vrf)# exit
Router(config-bgp-nbr)# rd auto
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)#redistribute connected
Router(config-bgp-nbr-af)# exit
Router(config-bgp)# neighbor 209.165.200.225
Router(config-bgp-nbr)# remote-as 2000
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 3
Router(config-bgp-nbr)# bfd minimum-interval 50
Router(config-bgp-nbr)# ebgp-multihop 255
Router(config-bgp-nbr)# update-source loopback 1
/* You can configure any interface here, including loopback or bvi */
Router(config-bgp-nbr)#address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
```

Running Configuration

```
router bgp 100
  neighbor 209.165.200.225
   vrf vrf1
    exit
     rd auto
      address-family ipv4 unicast
       redistribute connected
        exit
         neighbor 209.165.200.225
         remote-as 2000
         bfd fast-detect
         bfd multiplier 3
         bfd minimum-interval 50
         ebgp-multihop 255
         update-source loopback 1
          address-family ipv4 unicast
           route-policy pass-all in
           route-policy pass-all out
```

Verification

```
Router# show bfd session source 209.165.200.225
Thu Mar 10 10:13:43.305 IST
Src Addr         Dest Addr    VRF Name         H/W NPU
```

```
                              Local det time(int*mult)   State
                             Echo    Async
--------------   ---------   ------   ------ --------- ---------
209.165.200.225  192.0.2.254   vrf_1    Yes   0/0/CPU0
                             n/a  150ms(50ms*3)          UP
Router# show cef vrf vrf_1 209.165.200.225 location 0/0/CPU0
Thu Mar 10 10:24:13.372 IST
209.165.200.0/24, version 40, internal 0x5000001 0x30 (ptr 0x8ae26458) [1], 0x0 (0x0), 0xa08
 (0x8dc144a8)
 Updated Mar  9 15:09:43.398
 Prefix Len 24, traffic index 0, precedence n/a, priority 3
LDI Update time Mar  9 14:59:28.284
   via 1.1.1.1/32, 605 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x8dd35988 0x0]
    recursion-via-/32
    next hop VRF - 'default', table - 0xe0000000
    next hop 10.1.1.1/32 via 24015/0/21
    next hop 192.0.2.255/32 Te0/0/0/3.1  labels imposed {ImplNull 24162}
```

BFD-Triggered FRR

The BFD-triggered Fast Reroute (FRR) feature allows you to obtain link and node protection using Bidirectional Forwarding Detection (BFD) protocol. This feature provides fast forwarding path failure detection for the following:

- All media types
- Encapsulations
- Topologies
- Routing protocols

In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators.

When you enable FRR on Interior Gateway Protocol (IGP) in an IP network, BFD triggers FRR. BFD switches to the backup path when either the primary or the secondary link fails.

When you enable FRR on IGP in an MPLS TE network, you can enable BFD on single-hop IGP links. MPLS TE uses these links to define the protected TE tunnel paths. During traffic disruption on either the primary or the secondary links, BFD triggers a link-down event that triggers FRR.

When you enable FRR on IGP in a segment routing network, BFD triggers FRR.

This feature complements the link-scan failure detection feature. Among these two features, whichever detects the link down event first, triggers the FRR.

Restrictions

BFD-triggered FRR feature has the following restrictions:

- You cannot enable BFD on TE tunnels.
- You cannot enable BFD for multihop.
- You cannot enable BFD on logical bundles.

Configuration Example

```
Router# config
Router(config)# router ospf 100
Router(config-ospf)# router-id 10.32.32.32
Router(config-ospf)# area 0
Router(config-ospf)# exit
Router(config)# interface TenGigE 0/4/0/0.1
Router(config-subif)# bfd minimum-interval 3
Router(config-subif)# bfd fast-detect
Router(config-if)# exit
Router(config)# interface TenGigE 0/4/0/3
Router(config-if)# bfd minimum-interval 3
Router(config-if)# bfd fast-detect
Router(config-if)# exit
Router(config)# mpls traffic-eng
Router(config-mpls)# interface TenGigE 0/4/0/0.1
Router(config-mpls-if)# backup-path tunnel-te 5000
Router(config-mpls-if)# exit
Router(config-mpls)# interface TenGigE0/4/0/3
Router(config-mpls-if)# interface TenGigE0/4/0/3
Router(config-if)# exit
Router(config)# mpls ldp
Router(config-ldp)# router-id 10.32.32.32
Router(config-ldp)# interface Bundle-Ether1
Router(config-ldp-if)# exit
Router(config-ldp)# interface TenGigE0/4/0/0.1
Router(config-ldp-if)# exit
Router(config-ldp-if)# interface TenGigE0/4/0/3
Router(config-ldp-if)# exit
Router(config-ldp)# exit
Router(config)# rsvp
Router(config-rsvp)# interface TenGigE0/4/0/0.1
Router(config-rsvp-if)# bandwidth 100000000
Router(config-rsvp-if)# exit
Router(config-rsvp)# interface TenGigE0/4/0/3
Router(config-rsvp-if)# bandwidth 100000000
Router(config-rsvp-if)# exit
Router(config-rsvp)# exit
Router(config)# exit
Router(config)# interface tunnel-te5000
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# destination 10.31.31.31
Router(config-if)# path-option 1 explicit name ind1
Router(config-if)# exit
Router(config)# exit
Router(config)# explicit-path name direct1-sub
Router(config-expl-path)# index 1 next-address strict ipv4 unicast 10.2.36.2
Router(config-expl-path)# destination 10.31.31.31
Router(config-expl-path)# exit
Router(config)# explicit-path name ind1
Router(config-expl-path)# index 1 next-address strict ipv4 unicast 10.1.33.2
Router(config-expl-path)# index 1 next-address strict ipv4 unicast 10.1.30.1
Router(config-expl-path)# exit
```

Running Configuration

```
router ospf 100
 router-id 32.32.32.32
 area 0
interface TenGigE0/4/0/0.1
```

```
   bfd minimum-interval 3
   bfd fast-detect
interface TenGigE0/4/0/3
   bfd minimum-interval 3
   bfd fast-detect
!
mpls traffic-eng
interface TenGigE0/4/0/0.1
  backup-path tunnel-te 5000
interface TenGigE0/4/0/3
 !
mpls ldp
 router-id 32.32.32.32
 interface Bundle-Ether1
 !
 interface TenGigE0/4/0/0.1
 interface TenGigE0/4/0/3
 !
rsvp
interface TenGigE0/4/0/0.1
  bandwidth 100000000
interface TenGigE0/4/0/3
  bandwidth 100000000
!
interface tunnel-te1
 ipv4 unnumbered Loopback0
 destination 10.31.31.31
 fast-reroute
 path-option 1 explicit name direct1-sub
 !
interface tunnel-te5000
 ipv4 unnumbered Loopback0
 destination 10.31.31.31
 path-option 1 explicit name ind1
!
 !
explicit-path name direct1-sub
 index 1 next-address strict ipv4 unicast 10.2.36.2
 !
explicit-path name ind1
 index 1 next-address strict ipv4 unicast 10.1.33.2
 index 2 next-address strict ipv4 unicast 10.1.30.1
```

BFD over BVI

In order for a VLAN to span a router, the router must be capable of forwarding frames from one interface to another, while maintaining the VLAN header. If the router is configured for routing a Layer 3 (network layer) protocol, it will terminate the VLAN and MAC layers at the interface on which a frame arrives. The MAC layer header can be maintained if the router bridges the network layer protocol. However, even regular bridging terminates the VLAN header.

Using the Integrated Routing Bridging (IRB) feature, a router can be configured for routing and bridging the same network layer protocol, on the same interface. This allows the VLAN header to be maintained on a frame while it transits a router from one interface to another. IRB provides the ability to route between a bridged domain and a routed domain with the Bridge Group Virtual Interface (BVI). The BVI is a virtual interface within the router that acts like a normal routed interface that does not support bridging, but represents the comparable bridge group to routed interfaces within the router. The interface number of the BVI is the number of the bridge group that the virtual interface represents. This number is the link between the BVI and the bridge group.

Because the BVI represents a bridge group as a routed interface, it must be configured only with Layer 3 (L3) characteristics, such as network layer addresses. Similarly, the interfaces configured for bridging a protocol must not be configured with any L3 characteristics.

BFD over IRB is a multipath single-hop session. In a BFD multipath session, BFD can be applied over virtual interfaces or between interfaces that are multihops away. The Cisco IOS XR Software BFD multihop is based on the *RFC 5883—Bidirectional Forwarding Detection (BFD) for Multihop Paths*. BFD over IRB is supported on IPv4 address, IPv6 global address, and IPv6 link-local address. The BFD over IRB is supported only in asynchronous mode and does not support echo mode.

For more information on IRB feature, see *Integrated Routing and Bridging* Chapter in *Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers*.

Configuration Example

```
/* Configure a BVI interface and assign an IP address */
Router(config)# interface bvi 1
Router(config-if)# ipv4 address 192.168.1.1 255.255.255.0
Router(config-if)# exit

/* Configure the Layer 2 AC interface */
Router(config)# interface HundredGigE 0/0/1/3
Router(config-if)# l2transport
Router(config-if)# exit

/* Configure OSPF as the routing protocol */
Router(config)# router ospf 100
Router(config-ospf)# router-id 192.168.1.1
Router(config-ospf)# area 0
Router(config)# interface Loopback 100
Router(config)# exit

/* Configure BFD on BVI */
Router(config)# interface bvi1
Router(config-if)# bfd minimum-interval 100
Router(config-if)# bfd fast-detect
Router(config-if)# bfd multiplier 3

/* Configure the line cards to allow hosting of Multipath BFD sessions. */
Router# configure
Router(config)# bfd multipath include location 0/5/CPU0

/* Configure BVI on the peer nodes */
Router(config-if)# l2vpn
Router(config-l2vpn)# bridge group 1
Router(config-l2vpn-bg)# bridge domain 1
Router(config-l2vpn-bg-bd)# interface HundredGigE 0/0/1/3
Router(config-l2vpn-bg-bd-ac)# routed interface bvi 1
```

Running Configuration

```
interface BVI1
 ipv4 address 192.168.1.1 255.255.255.0
!
interface HundredGigE0/0/1/3
 l2transport
 !
!
router ospf 100
 router-id 192.168.1.1
```

```
 area 0
  interface Loopback100
  !
  interface BVI1
   bfd minimum-interval 100
   bfd fast-detect
   bfd multiplier 3
  !
 !
bfd multipath include location 0/5/CPU0
!
l2vpn
 bridge group 1
  bridge-domain 1
   interface HundredGigE0/0/1/3
   !
   routed interface BVI1
   !
```