



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.



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](#) , on page 1
- [BFD over Bundle](#) , on page 7
- [BFD Transparency](#), on page 21
- [BFD Hardware Offload Support for IPv4](#), on page 26
- [BFD Hardware Offload Support for IPv6](#), on page 27
- [BFD Object Tracking](#), on page 29
- [IPv4 Multihop BFD](#), on page 30
- [BFD over BVI](#), on page 32

BFD Overview

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.

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 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.

BFD is only supported in IP core. It cannot coexist with Label distribution Protocol, or Segment Routing, or Traffic Engineering in the core. This is applicable for releases prior to 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 7.2.1, BFD support over VRRP interface is supported.

SNMP traps are not supported for multipath BFD sessions.

BFD Timers

The BFD timers are applicable on the following NCS 540 variants:

Medium Density XR NCS 540 - N540-24Z8Q2C-SYS, N540-28Z4C-SYS, N540X-ACC-SYS, N540-ACC-SYS

Medium Density XR NCS 540 - N540-28Z4C-SYS-A, N540-28Z4C-SYS-D, N540X-16Z4G8Q2C-A, N540X-16Z4G8Q2C-D, N540-12Z20G-SYS-A, N540-12Z20G-SYS-D, N540X-12Z16G-SYS-A, N540X-12Z16G-SYS-D

Small Density XR NCS 540 - N540X-6Z18G-SYS-A, N540X-6Z18G-SYS-D, N540X-8Z16G-SYS-A, N540X-8Z16G-SYS-D



Note The router uses six unique timer profiles. Four timers profiles are available when you configure BFD over Bundle (BoB). Up to five timers profiles are available when you configure BoB.

Table 1: 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

Type of BFD Session	Minimum Timer Supported	Default/Minimum-interval timer (Multipliers)	Supported Timer Profiles (Up to 6 unique timers profiles)
BGP Multi Hop	50ms	3	Any

Table 2: 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
BGP Multi Hop	50ms	3	Any	256

Enable and Disable IPv6 Checksum Calculations for BFD on a Router

Perform the following steps to configure IPv6 checksum calculations for BFD on a Router.

```
RP/0/RP0/CPU0:router(config)# bfd
RP/0/RP0/CPU0:router(config-bfd-if)# ipv6 checksum disable
RP/0/RP0/CPU0:router(config-bfd-if)# commit
```

Configure BFD Under a Dynamic Routing Protocol or Use a Static Route

To establish a BFD neighbor, complete at least one of the following procedures to configure BFD under a dynamic routing protocol or to use a static route:

Enable BFD for OSPF on an Interface

Perform the following steps 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.



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

```

Router# configure

/* Enter OSPF configuration mode to configure the OSPF routing process. */
Router(config)# router ospf 0

/* Set the BFD minimum interval. The range is from 15 to 30000 milliseconds. */
Router(config-ospf)# bfd minimum-interval 6500

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

/* Configure an Open Shortest Path First (OSPF) area. */
Router(config-ospf)# area 0

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

/* Enable BFD to detect failures in the path between adjacent forwarding engines. */
Router(config-ospf-ar-if)# bfd fast-detect

```

Running Configuration

```

configure
  router ospf 0
  bfd minimum-interval 6500
  bfd multiplier 7
  area 0
  interface gigabitEthernet 0/3/0/1
  bfd fast-detect

```

Verification

Verify that BFD is enabled on the appropriate interface.

```

Router(config-ospf-ar-if)# show run router ospf

  router ospf 0
  bfd minimum-interval 6500
  bfd multiplier 7
  area 0
  interface gigabitEthernet 0/3/0/1
  bfd fast-detect

/* Verify the details of the IPv4 BFD session in the source router. */

Router# show bfd session

Interface  Dest Addr  Local det  time(int*mult)  State  Echo  Async  H/W  NPU
-----  -
Te0/0/0/0  10.23.1.2  0s(0s*0)  300ms(100ms*3)  UP     Yes   -----  ---  0/RP0/CPU0
BE3739     10.23.1.2  n/a       n/a              UP     No    n/a

```

Enable BFD over BGP

Perform the following steps to configure BFD over BGP. The following example shows how to configure BFD between autonomous system 65000 and neighbor 192.168.70.2:

```

Router# configure
Router(config)# router bgp 65000
Router(config-bgp)# bfd multiplier 2
Router(config-bgp)# bfd minimum-interval 20

```

```
Router(config-bgp)# neighbor 192.168.70.24
Router(config-bgp-nbr)# remote-as 2
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# commit
Router(config-bgp-nbr)# end
```

Running Configuration

```
router bgp 65000
  bfd multiplier 2
  bfd minimum-interval 20
  neighbor 192.168.70.24
    remote-as 2
    bfd fast-detect
  commit
end
```

Verification

Verify that BFD has been enabled over BGP.

```
Router# show run router bgp
router bgp 65000
  bfd multiplier 2
  bfd minimum-interval 20
  neighbor 192.168.70.24
    remote-as 2
    bfd fast-detect
```

Enable BFD on an IPv4 Static Route

The following procedure shows how to enable BFD on an IPv4 static route.

```
RP/0/RSP0/CPU0:router# configure

/*Enter static route configuration mode to configure static routing. */
RP/0/RSP0/CPU0:router(config)# router static

/* Enable BFD fast-detection on the specified IPV4 unicast destination address prefix and
on the forwarding next-hop address.*/
RP/0/RSP0/CPU0:router(config-static)# address-family ipv4 unicast 10.2.2.0/24 10.6.0.1 bfd
  fast-detect minimum-interval 1000 multiplier 5

RP/0/RSP0/CPU0:router(config-static)# commit
```

Running Configuration

```
configure
router static
  address-family ipv4 unicast 10.2.2.0/24 10.6.0.1 bfd fast-detect minimum-interval 1000
  multiplier 5
  commit
```

Verification

Verify that BFD is enabled on the appropriate interface.

```
RP/0/RSP0/CPU0:router# show run router static address-family ipv4 unicast
```

```

router static
 address-family ipv4 unicast
   10.2.2.0/24 10.6.0.1 bfd fast-detect minimum-interval 1000 multiplier 5
 commit
!
!

```

Enable BFD on an IPv6 Static Route

The following procedure describes how to enable BFD on a IPv6 static route.

```

RP/0/RP0/CPU0:router# configure

/* Enter static route configuration mode to configure static routing. */
RP/0/RP0/CPU0:router(config)# router static

/* Enable BFD fast-detection on the specified IPv6 unicast destination address prefix and
on the forwarding next-hop address. */
/* BFD sessions are established with the next hop 2001:0DB8:D987:398:AE3:B39:333:783 when
it becomes reachable. */

RP/0/RP0/CPU0:router(config-static)# address-family ipv6 unicast 2001:0DB8:C18:2:1::F/64
2001:0DB8:D987:398:AE3:B39:333:783 bfd fast-detect minimum-interval 150 multiplier 4

RP/0/RP0/CPU0:router(config-static-vrf)# commit

```

Running Configuration

```

configure
 router static
   address-family ipv6 unicast 2001:0DB8:C18:2:1::F/64 2001:0DB8:D987:398:AE3:B39:333:783
 bfd fast-detect minimum-interval 150 multiplier 4
 commit

```

Verification

Verify that BFD is enabled on the appropriate interface.

```

RP/0/RP0/CPU0:router# show run router static address-family ipv6 unicast

configure
 router static
 address-family ipv6 unicast 2001:0DB8:C18:2:1::F/64 2001:0DB8:D987:398:AE3:B39:333:783 bfd
 fast-detect minimum-interval 150 multiplier 4
 commit

```

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

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.

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
```

BFD over Bundle

BFD over Bundle feature enables BFD sessions to monitor the status of individual bundle member links. BFD notifies the bundle manager immediately when one of the member links goes down, and reduces the bandwidth used by the bundle.

Restrictions

The following are the restrictions in using BFD over Bundle feature:

- It is only supported in IETF mode.
- It is only supported on main bundle interface; it is not supported on bundle sub-interfaces.
- It is not supported on routing protocols, such as OSPF, ISIS, and BGP.
- When BFD timer is configured to 3.3 ms, which is the most aggressive timer, 256 sessions can be brought up.
- If BFD timer is configured to greater than 100 ms, 300 BFD sessions can be brought up simultaneously.
- BFD echo mode and encryption is not supported.
- BFD dampening is not supported.

Configure BFD over Bundle

Configuring BFD over bundle involves the following steps:

- Specify the mode, BFD packet transmission intervals, and failure detection times on a bundle



Note Repeat the same configuration steps in the destination router.

```

/* Enable and Disable IPv6 checksum calculations for BFD on a router. */

Router(config-if)# bfd
Router(config-bfd-if)# ipv6 checksum disable
Router(config-bfd-if)# dampening disable
Router(config-bfd-if)# commit

/* Specify the mode, BFD packet transmission intervals, and failure detection times on a
bundle */

Router(config)# interface Bundle-Ether 3739
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv4 multiplier 3
Router(config-if)# bfd address-family ipv4 destination 10.23.1.2
Router(config-if)# bfd address-family ipv4 fast-detect
Router(config-if)# bfd address-family ipv4 minimum-interval 100
Router(config-if)# bfd address-family ipv6 multiplier 3
Router(config-if)# bfd address-family ipv6 destination 2001:DB8:1::2
Router(config-if)# bfd address-family ipv6 fast-detect
Router(config-if)# bfd address-family ipv6 minimum-interval 100
Router(config-if)# ipv4 address 10.23.1.1 255.255.255.252
Router(config-if)# ipv6 address 2001:DB8:1::2/120

```



```
Router(config-if)# load-interval 30
Router(config-if)# commit
Router(config)# interface TenGigE 0/0/0/0
Router(config-if)# bundle id 3739 mode active
```

Running Configuration

```
bfd
  ipv6 checksum disable
  dampening disable!
!

interface Bundle-Ether3739
  bfd mode ietf
  bfd address-family ipv4 multiplier 3
  bfd address-family ipv4 destination 10.23.1.2
  bfd address-family ipv4 fast-detect
  bfd address-family ipv4 minimum-interval 100
  bfd address-family ipv6 multiplier 3
  bfd address-family ipv6 destination 2001:DB8:1::2
  bfd address-family ipv6 fast-detect
  bfd address-family ipv6 minimum-interval 100
  ipv4 address 10.23.1.1 255.255.255.252
  ipv6 address 2001:DB8:1::2/120
  load-interval 30
!

interface TenGigE 0/0/0/0
  bundle id 3739 mode active
```

Verification

The following show command outputs displays the status of BFD sessions on bundle members:

```
/* Verify the details of the IPv4 BFD session in the source router. */
```

```
Router# show bfd session
```

Interface	Dest Addr	Local det	time(int*mult)	State	Echo	Async	H/W	NPU
Te0/0/0/0	10.23.1.2	0s(0s*0)	300ms(100ms*3)	UP	Yes			0/RP0/CPU0
BE3739	10.23.1.2	n/a	n/a	UP	No	n/a		

```
/* Verify the details of the IPv4 BFD session in the destination router. */
```

```
Router# show bfd session
```

Interface	Dest Addr	Local det	time(int*mult)	State	Echo	Async	H/W	NPU
Te0/6/0/0	10.23.1.1	0s(0s*0)	300ms(100ms*3)	UP	No	n/a		
BE3739	10.23.1.1	n/a	n/a	UP	No	n/a		

```
/* Verify the details of the IPv6 BFD session in the source router. */
```

```
Router# show bfd ipv6 session
```

Interface	Dest Addr	Local det	time(int*mult)	State	H/W	NPU	Echo	Async
Te0/0/0/0	10:23:1::2	Yes		0/RP0/0s	(0s*0)	00ms(100ms*3)	UP	

```

BE3739      10:23:1::2 No                n/a      n/a      n/a      UP

/* Verify the details of the IPv6 BFD session in the destination router. */

Router# show bfd ipv6 session
-----
Interface  Dest Addr  Local det  time(int*mult)  State      H/W NPU      Echo  Async
-----
Te0/6/0/0  10:23:1::1 No         n/a         0s(0s*0)       0s      300ms(100ms*3)  UP
BE3739     10:23:1::1 No         n/a         n/a            n/a      n/a          UP
    
```

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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router bgp <i>autonomous-system-number</i> 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 <i>ip-address</i> 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 <i>autonomous-system-number</i> 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

	Command or Action	Purpose
		networking devices and the neighbor 172.168.40.24.
Step 6	bfd minimum-interval <i>milliseconds</i> 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 <i>multiplier</i> Example: RP/0/RP0/CPU0:router (config-bgp-nbr) #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
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: <ul style="list-style-type: none"> • 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.

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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router ospf <i>process-name</i> Example:	Enters OSPF configuration mode, allowing you to configure the OSPF routing process.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config)# router ospf 0	Note To configure BFD for IS-IS, enter the corresponding configuration mode.
Step 3	area <i>area-id</i> Example: RP/0/RP0/CPU0:router(config-ospf)# area 0	Configures an Open Shortest Path First (OSPF) area. Replace <i>area-id</i> with the OSPF area identifier.
Step 4	interface <i>type interface-path-id</i> 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 <i>milliseconds</i> 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 <i>multiplier</i> 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: <ul style="list-style-type: none"> • 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.

Enabling BFD on a Static Route

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	router static Example: RP/0/RP0/CPU0:router(config)# router static	Enters static route configuration mode, allowing you to configure static routing.
Step 3	address-family ipv4 unicast <i>address nexthop</i> Example: RP/0/RP0/CPU0:router(config-static)# address-family ipv4 unicast 10.2.2.0/24 10.6.0.2	Enables BFD fast-detection on the specified IPv4 unicast destination address prefix and on the forwarding next-hop address.
Step 4	interface <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-static)# 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-static-if)# bfd fast-detect	Enables BFD to detect failures in the path between adjacent forwarding engines.
Step 6	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: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No — Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Cancel —Remains in the configuration session, without committing the configuration changes.

Enabling BFD Sessions on Bundle Members

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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	interface Bundle-Ether <i>bundle-id</i> 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: <ul style="list-style-type: none"> • 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:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	interface Bundle-Ether <i>bundle-id</i> 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 <i>ip-address</i> 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 <i>ip-address</i> 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: <ul style="list-style-type: none"> • 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:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	interface Bundle-Ether <i>bundle-id</i> 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 <i>kbps</i> 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 <i>links</i> 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: <ul style="list-style-type: none"> • 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:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	interface Bundle-Ether <i>bundle-id</i> 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 <i>milliseconds</i> 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.	
Step 4	bfd address-family ipv4 multiplier <i>multiplier</i> 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.

	Command or Action	Purpose
		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.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • 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 over Bundle per Member Mode

Procedure

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters mode.
Step 2	<p>bfd bundle per-member mode ietf</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# bfd bundle per-member mode ietf</pre>	Enables IETF mode for BFD over per-bundle member link.
Step 3	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • 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:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	interface Bundle-Ether <i>bundle-id</i> 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: <ul style="list-style-type: none"> • Yes — Saves configuration changes and exits the configuration session. • No —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Cancel —Remains in the configuration session, without committing the configuration changes.
Step 6	show bundle bundle-ether <i>bundle-id</i>	Displays the selected bundle mode.

BFD over Bundle with IPv4 Unnumbered Interfaces

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

```
/* Enable IS-IS and configure routing level for an area. */
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tengige 0/0/0/2.1
```

```

RP/0/RP0/CPU0:router(config-subif)# exit
RP/0/RP0/CPU0:router(config)# router isis
RP/0/RP0/CPU0:router(config-isis)# is-type level-2-only
RP/0/RP0/CPU0:router(config-isis)# net 49.1234.2222.2222.2222.00
RP/0/RP0/CPU0:router(config-isis)# nsr
RP/0/RP0/CPU0:router(config-isis)# nsf cisco
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-isis-af)# metric style wide
RP/0/RP0/CPU0:router(config-isis)# end
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 199
RP/0/RP0/CPU0:router(config-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:router(config)# interface Loopback 0
RP/0/RP0/CPU0:router(config-if)# end
RP/0/RP0/CPU0:router(config-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-if)# exit

/* Configure L2VPN EVPN address family. */
RP/0/RP0/CPU0:router(config)# router bgp 100
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 10.10.10.1
RP/0/RP0/CPU0:router(config-bgp)# address-family l2vpn evpn
RP/0/RP0/CPU0:router(config-bgp)# neighbor 192.0.2.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:router(config-bgp-nbr)# update-source Loopback 0
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family l2vpn evpn

/* Configure MPLS LDP for the physical core interface. */
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# mpls ldp
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:router(config-bgp-nbr)# exit
RP/0/RP0/CPU0:router(config-bgp)# exit
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 199
RP/0/RP0/CPU0:router(config-if)# exit

/* Configure L2VPN Xconnect. */
RP/0/RP0/CPU0:router(config)# l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)# router-id 10.10.10.1
RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group bfdtr
RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p vpws-ce
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# interface TenGigE 0/0/0/1.1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)# neighbor evpn evi 100 target 3 source 4

```

Configure BFD on the Customer Edge Router

```

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router bgp 100
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 10.10.10.1
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# exit
RP/0/RP0/CPU0:router(config-bgp)# neighbor 172.16.0.1
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd fast-detect
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd multiplier 2
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd minimum-interval 100
RP/0/RP0/CPU0:router(config-bgp-nbr)# update-source TenGigE 0/0/0/16.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)#

```

Running Configuration

This section shows the BFD Transparency configuration.

```

!
interface TenGigE 0/0/0/1.1
  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 0/0/0/1.1
    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/RP0/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. */
    
```

```

RP0/RP0/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/RP0/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/RP0/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.

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 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/RP0/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/RP0/CPU0

```

BFD Hardware Offload Support for IPv6

Table 3: Feature History

Feature Name	Release Information	Feature Description
BFD v6 - HW Offload and IPv6 BFD/BoB (Bundle over Bundle)	Release 7.3.1	The Bidirectional Forwarding detection (BFD) Hardware Offload feature enables the offload of a BFD session in an IPv6 network. With this feature, each bundle member link with IPv6 address runs its own BFD session. This 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.

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 interfaces.
- 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/RP0/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
-----
H/W                NPU          Echo          Local det time(int*mult)  State
                   NPU          Echo          Async
-----
BE7.2              fe80::28a:96ff:fed6:9cdb
Yes                0/RP0/CPU0   0s(0s*0)     900ms(300ms*3)          UP
BE7.4              fe80::28a:96ff:fed6:9cdb
Yes                0/RP0/CPU0   0s(0s*0)     900ms(300ms*3)          UP

```

BFD Object Tracking

Object Tracking is enhanced to support BFD to track the reachability of remote IP addresses. This will enable complete detection and HSRP switch over to happen within a time of less than one second as BFD can perform the detection in the order of few milliseconds

Configuring BFD Object Tracking:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters mode.
Step 2	track track-name Example: RP/0/RP0/CPU0:router(config)# track track1	Enters track configuration mode. • <i>track-name</i> —Specifies a name for the object to be tracked.
Step 3	type bfdtr rate tx-rate Example: RP/0/RP0/CPU0:router(config-track)# type bfdtr rate 4	tx_rate - time in msec at which the BFD should probe the remote entity
Step 4	debouncedebounce Example: RP/0/RP0/CPU0:router(config-if)# debounce 10	debounce - count of consecutive BFD probes whose status should match before BFD notifies OT
Step 5	interface if-name Example: RP/0/RP0/CPU0:router(config-track-line-prot)# interface GigabitEthernet0/0/0/4	if_name - interface name on the source to be used by BFD to check the remote BFD status.
Step 6	destaddress dest_addr Example: RP/0/RP0/CPU0:router(config-if)#destaddress 1.2.3.4	dest_addr - IPV4 address of the remote BFD entity being tracked.
Step 7	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session.

	Command or Action	Purpose
		<p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • 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.

IPv4 Multihop BFD

Table 4: 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.

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 ++
```

BFD over BVI

Table 5: Feature History

Feature Name	Release Information	Feature Description
BFD on BVI	Release 7.3.1	<p>BFD can be configured on Bridge group Virtual Interface (BVI). BVI is a virtual interface within the router that acts like a normal routed interface that does not support bridging but represents the bridge group for the bridged physical interfaces.</p> <p>BFD detects the Layer3 fault over the BVI much quicker and inform the same to routing protocols.</p>

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.

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 50
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/RP0/CPU0
```

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 50
   bfd fast-detect
   bfd multiplier 3
  !
!
bfd multipath include location 0/RP0/CPU0
!
!
```

