Implementing BFD

This module describes the configuration of bidirectional forwarding detection (BFD) on the Cisco ASR 9000 Series Router.

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.

Feature History for Implementing Bidirectional Forwarding Detection

Release	Modification
Release 3.7.2	BFD was introduced.

Release 3.9.0	• Support for these applications with BFD was added:
	  • Hot Standby Router Protocol (HSRP)
	  • Virtual Router Redundancy Protocol (VRRP)
	• The **dampening** command was added to minimize BFD session flapping and delay session startup.
	• The **echo ipv4 source** command was added to specify a source IP address and override the default.
	• The **ipv6 checksum** command was added to enable and disable the IPv6 UDP checksum computation and BFD interface configuration modes.

Release 4.0.0	Support for these BFD features was added: • BFD for OSPFv3 • BFD for IPv6 Support for BFD was added on the following SPAs: • 1-Port OC-192c/STM-64 POS/RPR XFP SPA • 2-Port OC-48c/STM-16 POS/RPR SPA • 8-Port OC-12c/STM-4 POS SPA

Release 4.0.1	Support for these BFD features was added: • Support for BFD Per Member Links on Link Bundles was added. • The **echo latency detect** command was added to enable latency detection for BFD echo packets on non-bundle interfaces. • The **echo startup validate** command was added to verify the echo path before starting a BFD session on non-bundle interfaces.
Release 4.2.0	Support for these BFD features was added: • BFD Multihop Global TTL check. • BFD Multihop support for BGP and • BFD Multihop support for IPv4 traffic. • The **multihop ttl-drop-threshold** command was added to specify the TTL value to start dropping packets for multihop sessions.
Release 4.2.1	Support for BFD Multihop feature was added on the ASR9K-SIP-700 line card.
Release 4.3.0	Support for these features was added: • BFD over GRE • BFD IPv6 Multihop
Release 4.3.1	Support for these features was added: • BFD over MPLS Traffic Engineering LSPs • BFD over Pseudowire Head-end • BFD over Satellite Interfaces
Release 5.2.4	Support for BFD over Bundles CISCO/IETF mode support on a per bundle basis was added.

Prerequisites for Implementing BFD

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

The following prerequisites are required to implement BFD:

- If enabling BFD on Multiprotocol Label Switching (MPLS), an installed composite PIE file including the MPLS package, or a composite-package image is required. For Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Static, and Open Shortest Path First (OSPF), an installed Cisco IOS XR IP Unicast Routing Core Bundle image is required.
- Interior Gateway Protocol (IGP) is activated on the router if you are using IS-IS or OSPF.
- On the Cisco ASR 9000 Series Router, each line card supporting BFD must be able to perform the following tasks:
  - Send echo packets every 50ms * 3 (as a minimum under normal conditions)
  - Send control packets every 150ms * 3 (as a minimum under stress conditions)
  - Send and receive up to 9600 User Datagram Protocol (UDP) pps. This sustains 144 sessions at a 15-ms echo interval (or 1440 sessions at a 150-ms echo interval).
- To enable BFD for a neighbor, the neighbor router must support BFD.
- In Cisco IOS XR releases before Release 3.9.0, we recommended that you configure the local router ID with the **router-id** command in global configuration mode prior to setting up a BFD session. If you did not configure the local router ID, then by default the source address of the IP packet for BFD echo mode is the IP address of the output interface. Beginning in Cisco IOS XR release 3.9.0 and later, you can use the **echo ipv4 source** command to specify the IP address that you want to use as the source address.
- To support BFD on bundle member links, be sure that the following requirements are met:
  - The routers on either end of the bundle are connected back-to-back without a Layer 2 switch in between.
  - For a BFD session to start, any one of the following configurations or states are present on the bundle member:

    Link Aggregation Control Protocol (LACP) Distributing state is reached, –Or–

    EtherChannel or POS Channel is configured, –Or–

    Hot Standby and LACP Collecting state is reached.

Restrictions for Implementing BFD

These restrictions apply to BFD:

- Demand mode is not supported in Cisco IOS XR software.
- BFD echo mode is not supported for these features:
  - BFD for IPv4 on bundled VLANs
  - BFD for IPv6 (global and link-local addressing)
  - BFD with uRPF (IPv4 or IPv6)
  - Rack reload and online insertion and removal (OIR) when a BFD bundle interface has member links that span multiple racks

    - BFD for Multihop Paths

- BFD for IPv6 has these restrictions:
-
    - BFD for IPv6 is not supported on bundled VLAN interfaces
    - BFD for IPv6 static routes, OSPFv3, and BGP are supported by the client
    - BFD for IPv6 static routes that have link-local address as the next-hop is not supported

- For BFD on bundle member links, only a single BFD session for each bundle member link is created, monitored, and maintained for the IPv4 addressing type only. IPv6 and VLAN links in a bundle have the following restrictions:
    - IPv6 states are not explicitly monitored on a bundle member and they inherit the state of the IPv4 BFD session for that member interface.
    - VLAN subinterfaces on a bundle member also inherit the BFD state from the IPv4 BFD session for that member interface. VLAN subinterfaces are not explicitly monitored on a bundle member.

- Echo latency detection and echo validation are not supported on bundle interfaces.
- BFD Multihop can be run on any non-default VRF but selective VRF download must be disabled. For more information on the configuration and commands for selective VRF download, see *Routing Configuration Guide for Cisco ASR 9000 Series Routers* and *Routing Command Reference for Cisco ASR 9000 Series Routers*
- BFD over GRE feature is not supported on Cisco ASR 9000 Series SPA Interface Processor-700.
- BFD IPv6 Multihop feature is not supported on Cisco ASR 9000 Series SPA Interface Processor-700.
- BFD over Logical Bundle feature is not supported on Cisco ASR 9000 Series SPA Interface Processor-700.
- Only BFD MH and BLB are supported on Ethernet Line Card. The BFD multipath sessions such as BFDoTE, BFDoIRB, BFDoGRE etc. are not supported in this line card.
- BFD over satellite sessions is not supported on ASR 9000 Ethernet Line Card. It is also not supported on Cisco ASR 9000 Series SPA Interface Processor-700.
- When explicit bundle hash is configured on the bundle interface, the bundle manager performs hashing based on the source or destination IP address. This causes all the echo packets to be sent on one of the member links only, and the other links starts flapping.

  BFD Echo requires hashing based on source ports, so IP-based hashing does not distribute echo packets across the member links.

  Avoid IP-based hashing for the configured bundle or disable the echo mode as they both do not interoperate.

  To remove IP-based hash, perform the following steps:

```
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 1
RP/0/RSP0/CPU0:router(config)# no bundle load-balancing hash dst-ip

/* or */

RP/0/RSP0/CPU0:router(config)# no bundle load-balancing hash src-ip
```

To disable echo for the configured bundle, perform the following steps. The **echo disable** command is executed in either global mode or interface configuration mode:

```
 RP/0/RSP0/CPU0:router(config)# bfd
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 1
RP/0/RSP0/CPU0:router(config-if)# echo disable

*/ or */

RP/0/RSP0/CPU0:router(config)# bfd  echo disable
```

- SNMP traps are not supported for multipath BFD sessions.
- Aggressive timer is not recommended to use for the BFD Multipath sessions and the Multihop sessions. The recommend time is more than 100 ms x 3 = 300 ms.

Information About BFD

Differences in BFD in Cisco IOS XR Software and Cisco IOS Software

If you are already familiar with BFD configuration in Cisco IOS software, be sure to consider the following differences in BFD configuration in the Cisco IOS XR software implementation:

- In Cisco IOS XR software, BFD is an application that is configured under a dynamic routing protocol, such as an OSPF or BGP instance. This is not the case for BFD in Cisco IOS software, where BFD is only configured on an interface.
- In Cisco IOS XR software, a BFD neighbor is established through routing. The Cisco IOS **bfd neighbor** interface configuration command is not supported in Cisco IOS XR software.
- Instead of using a dynamic routing protocol to establish a BFD neighbor, you can establish a specific BFD peer or neighbor for BFD responses in Cisco IOS XR software using a method of static routing to define that path. In fact, you must configure a static route for BFD if you do not configure BFD under a dynamic routing protocol in Cisco IOS XR software.
- A router running BFD in Cisco IOS software can designate a router running BFD in Cisco IOS XR software as its peer using the **bfd neighbor** command; the Cisco IOS XR router must use dynamic routing or a static route back to the Cisco IOS router to establish the peer relationship. See the BFD Peers on Routers Running Cisco IOS and Cisco IOS XR Software: Example.

BFD Multipath Sessions Support on nV Edge System

The following BFD Multipath Sessions are supported on nV Edge System:

- BFD over GRE
- BFD over Logical Bundle
- BFD over IRB
- BFD Multihop (only supported from 5.2.2 onwards)
- BFD over MPLS TE

- BFD over Satellite

BFD Modes of Operation

Cisco IOS XR software supports the asynchronous mode of operation only, with or without using echo packets. Asynchronous mode without echo will engage various pieces of packet switching paths on local and remote systems. However, asynchronous mode with echo is usually known to provide slightly wider test coverage as echo packets are self-destined packets which traverse same packet switching paths as normal traffic on the remote system.

BFD echo mode is enabled by default for the following interfaces:

- For IPv4 on member links of BFD bundle interfaces.
- For IPv4 on other physical interfaces whose minimum interval is less than two seconds.

When BFD is running asynchronously without echo packets (Figure 35), the following occurs:

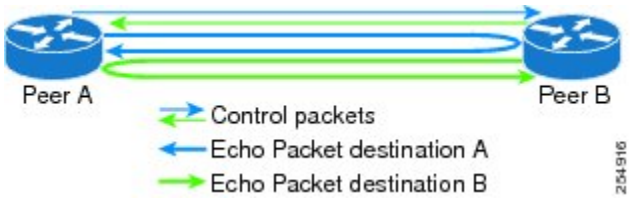
- Each system periodically sends BFD control packets to one another. Packets sent by BFD router "Peer A" to BFD router "Peer B" have a source address from Peer A and a destination address for Peer B.
- Control packet streams are independent of each other and do not work in a request/response model.
- If a number of packets in a row are not received by the other system, the session is declared down.

Figure 1: BFD Asynchronous Mode Without Echo Packets

When BFD is running asynchronously with echo packets (Figure 36), the following occurs:

- BFD echo packets are looped back through the forwarding path only of the BFD peer and are not processed by any protocol stack. So, packets sent by BFD router "Peer A" can be sent with both the source and destination address of Peer A.
- BFD echo packets are sent in addition to BFD control packets.

Figure 2: BFD Asynchronous Mode With Echo Packets



For more information about control and echo packet intervals in asynchronous mode, see the BFD Packet Intervals and Failure Detection.

BFD Packet Information

BFD Source and Destination Ports

BFD payload control packets are encapsulated in UDP packets, using destination port 3784 and source port 49152. Even on shared media, like Ethernet, BFD control packets are always sent as unicast packets to the BFD peer.

Echo packets are encapsulated in UDP packets, as well, using destination port 3785 and source port 3785.

The BFD over bundle member feature increments each byte of the UDP source port on echo packets with each transmission. UDP source port ranges from 0xC0C0 to 0xFFFF. For example:

1st echo packet: 0xC0C0

2nd echo packet: 0xC1C1

3rd echo packet: 0xC2C2

The UDP source port is incremented so that sequential echo packets are hashed to deviating bundle member.

BFD Packet Intervals and Failure Detection

BFD uses configurable intervals and multipliers to specify the periods at which control and echo packets are sent in asynchronous mode and their corresponding failure detection.

There are differences in how these intervals and failure detection times are implemented for BFD sessions running over physical interfaces, and BFD sessions on bundle member links.

BFD Packet Intervals on Physical Interfaces

When BFD is running over physical interfaces, echo mode is used only if the configured interval is less than two seconds.

BFD sessions running over physical interfaces when echo mode is enabled send BFD control packets at a slow rate of every two seconds. There is no need to duplicate control packet failure detection at a fast rate because BFD echo packets are already being sent at fast rates and link failures will be detected when echo packets are not received within the echo failure detection time.

BFD Packet Intervals on Bundle Member Links

On each bundle member interface, BFD asynchronous mode control packets run at user-configurable interval and multiplier values, even when echo mode is running.

However, on a bundle member interface when echo mode is enabled, BFD asynchronous mode must continue to run at a fast rate because one of the requirements of enabling BFD echo mode is that the bundle member interface is available in BFD asynchronous mode.

The maximum echo packet interval for BFD on bundle member links is the minimum of either 30 seconds or the asynchronous control packet failure detection time.

When echo mode is disabled, the behavior is the same as BFD over physical interfaces, where sessions exchange BFD control packets at the configured rate.

Control Packet Failure Detection In Asynchronous Mode

Control packet failure in asynchronous mode without echo is detected using the values of the minimum interval (bfd minimum-interval for non-bundle interfaces, and bfd address-family ipv4 minimum-interval for bundle

interfaces) and multiplier (bfd multiplier for non-bundle interfaces, and bfd address-family ipv4 multiplier for bundle interfaces) commands.

For control packet failure detection, the local multiplier value is sent to the neighbor. A failure detection timer is started based on (*I* x *M*), where *I* is the negotiated interval, and *M* is the multiplier provided by the remote end.

Whenever a valid control packet is received from the neighbor, the failure detection timer is reset. If a valid control packet is not received from the neighbor within the time period (*I* x *M*), then the failure detection timer is triggered, and the neighbor is declared down.

Echo Packet Failure Detection In Asynchronous Mode

The standard echo failure detection scheme is done through a counter that is based on the value of the **bfd multiplier** command on non-bundle interfaces, and the value of the **bfd address-family ipv4 multiplier** command for bundle interfaces.

This counter is incremented each time the system sends an echo packet, and is reset to zero whenever *any* echo packet is received, regardless of the order that the packet was sent in the echo packet stream.

Under ideal conditions, this means that BFD generally detects echo failures that exceed the period of time (*I* x *M*) or (*I* x *M* x *M*) for bundle interfaces, where:

- *I*—Value of the minimum interval (bfd minimum-interval for non-bundle interfaces, and **bfd address-family ipv4 minimum-interval** for bundle interfaces).
- *M*—Value of the multiplier (**bfd multiplier** for non-bundle interfaces, and **bfd address-family ipv4 multiplier** for bundle interfaces) commands.

So, if the system transmits one additional echo packet beyond the multiplier count without receipt of any echo packets, echo failure is detected and the neighbor is declared down (See Example 2).

However, this standard echo failure detection does not address latency between transmission and receipt of any specific echo packet, which can build beyond (*I* x *M*) over the course of the BFD session. In this case, BFD will not declare a neighbor down as long as any echo packet continues to be received within the multiplier window and resets the counter to zero. Beginning in Cisco IOS XR 4.0.1, you can configure BFD to measure this latency for non-bundle interfaces. For more information, see Example 3 and the Echo Packet Latency.

Echo Failure Detection Examples

This section provides examples of several scenarios of standard echo packet processing and failure detection without configuration of latency detection for non-bundle interfaces. In these examples, consider an interval of 50 ms and a multiplier of 3.

Note The same interval and multiplier counter scheme for echo failure detection is used for bundle interfaces, but the values are determined by the **bfd address-family ipv4 multiplier** and **bfd address-family ipv4 minimum-interval commands**, and use a window of (*I* x *M* x *M*) to detect absence of receipt of echo packets.

Example 1

The following example shows an ideal case where each echo packet is returned before the next echo is transmitted. In this case, the counter increments to 1 and is returned to 0 before the next echo is sent and no echo failure occurs. As long as the roundtip delay for echo packets in the session is less than the minimum interval, this scenario occurs:

```
Time (T): Echo#1 TX (count = 1)
T + 1 ms: Echo#1 RX (count = 0)
T + 50 ms: Echo#2 TX (count = 1)
T + 51 ms: Echo#2 RX (count = 0)
T + 100 ms: Echo#3 TX (count = 1)
T + 101 ms: Echo#3 RX (count = 0)
T + 150 ms: Echo#4 TX (count = 1)
T + 151 ms: Echo#4 RX (count = 0)
```

Example 2

The following example shows the absence in return of any echo packets. After the transmission of the fourth echo packet, the counter exceeds the multiplier value of 3 and echo failure is detected. In this case, echo failure detection occurs at the 150 ms (*I* x *M*) window:

```
Time (T): Echo#1 TX (count = 1)
T + 50 ms: Echo#2 TX (count = 2)
T + 100 ms: Echo#3 TX (count = 3)
T + 150 ms: Echo#4 TX (count = 4 -> echo failure
```

Example 3

The following example shows an example of how roundtrip latency can build beyond (*I* x *M*) for any particular echo packet over the course of a BFD session using the standard echo failure detection, but latency between return of echo packets overall in the session never exceeds the (*I* x *M*) window and the counter never exceeds the multiplier, so the neighbor is not declared down.

Note You can configure BFD to detect roundtrip latency on non-bundle interfaces using the **echo latency detect** command beginning in Cisco IOS XR 4.0.1.

```
Time (T): Echo#1 TX (count = 1)
T + 1 ms: Echo#1 RX (count = 0)
T + 50 ms: Echo#2 TX (count = 1)
T + 51 ms: Echo#2 RX (count = 0)
T + 100 ms: Echo#3 TX (count = 1)
T + 150 ms: Echo#4 TX (count = 2)
T + 151 ms: Echo#3 RX (count = 0; ~50 ms roundtrip latency)
T + 200 ms: Echo#5 TX (count = 1)
T + 250 ms: Echo#6 TX (count = 2)
T + 251 ms: Echo#4 RX (count = 0; ~100 ms roundtrip latency)
T + 300 ms: Echo#7 TX (count = 1)
T + 350 ms: Echo#8 TX (count = 2)
T + 351 ms: Echo#5 RX (count = 0; ~150 ms roundtrip latency)
T + 451 ms: Echo#6 RX (count = 0; ~200 ms roundtrip latency; no failure detection)
T + 501 ms: Echo#7 RX (count = 0; ~200 ms roundtrip latency; no failure detection)
T + 551 ms: Echo#8 RX (count = 0; ~200 ms roundtrip latency; no failure detection)
```

Looking at the delay between receipt of echo packets for the BFD session, observe that no latency is beyond the (I x M) window:

```
Echo#1 RX – Echo#2 RX: 50 ms
Echo#2 RX – Echo#3 RX: 100ms
Echo#3 RX - Echo#4 RX: 100ms
```

```
Echo#4 RX - Echo#5 RX: 100ms
Echo#5 RX - Echo#6 RX: 100ms
Echo#6 RX - Echo#7 RX: 50ms
Echo#7 RX - Echo#8 RX: 50ms
```

Summary of Packet Intervals and Failure Detection Times for BFD on Bundle Interfaces

For BFD on bundle interfaces, with a session interval *I* and a multiplier *M*, these packet intervals and failure detection times apply for BFD asynchronous mode (Table 1: BFD Packet Intervals and Failure Detection Time Examples on Bundle Interfaces):

- Value of *I*—Minimum period between sending of BFD control packets.
- Value of *I* x *M*
  - BFD control packet failure detection time.
  - Minimum period between sending of BFD echo packets.

The BFD control packet failure detection time is the maximum amount of time that can elapse without receipt of a BFD control packet before the BFD session is declared down.

- Value of (*I* x *M*) x *M*—BFD echo packet failure detection time. This is the maximum amount of time that can elapse without receipt of a BFD echo packet (using the standard multiplier counter scheme as described in Echo Packet Failure Detection In Asynchronous Mode) before the BFD session is declared down.

Table 1: BFD Packet Intervals and Failure Detection Time Examples on Bundle Interfaces

Configured Async Control Packet Interval (ms) (bfd address-family ipv4 minimum-interval)	Configured Multiplier (bfd address-family ipv4 multiplier)	Async Control Packet Failure Detection Time (ms) (Interval x Multiplier)	Echo Packet Interval (Async Control Packet Failure Detection Time)	Echo Packet Detection Tim (Echo Interva Multiplier)
50	3	150	150	450
75	4	300	300	1200
200	2	400	400	800
2000	3	6000	6000	18000
15000	3	45000	30000[1]	90000

[1] The maximum echo packet interval for BFD on bundle member links is the minimum of either 30 seconds or the asynchronous control packet failure detection time.

Echo Packet Latency

In Cisco IOS XR software releases prior to Cisco IOS XR 4.0.1, BFD only detects an absence of receipt of echo packets, not a specific delay for TX/RX of a particular echo packet. In some cases, receipt of BFD echo packets in general can be within their overall tolerances for failure detection and packet transmission, but a longer delay might develop over a period of time for any particular roundtrip of an echo packet (See Example 3).

Beginning in Cisco IOS XR Release 4.0.1, you can configure the router to detect the actual latency between transmitted and received echo packets on non-bundle interfaces and also take down the session when the latency exceeds configured thresholds for that roundtrip latency. For more information, see the Configuring BFD Session Teardown Based on Echo Latency Detection.

In addition, you can verify that the echo packet path is within specified latency tolerances before starting a BFD session. With echo startup validation, an echo packet is periodically transmitted on the link while it is down to verify successful transmission within the configured latency before allowing the BFD session to change state. For more information, see the Delaying BFD Session Startup Until Verification of Echo Path and Latency.

Priority Settings for BFD Packets

For all interfaces under over-subscription, the internal priority needs to be assigned to remote BFD Echo packets, so that these BFD packets are not overwhelmed by other data packets. In addition, CoS values need to be set appropriately, so that in the event of an intermediate switch, the reply back of remote BFD Echo packets are protected from all other packets in the switch.

As configured CoS values in ethernet headers may not be retained in Echo messages, CoS values must be explicitly configured in the appropriate egress QoS service policy. CoS values for BFD packets attached to a traffic class can be set using the set cos command. For more information on configuring class-based unconditional packet marking, see "Configuring Modular QoS Packet Classification" in the *Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers*.

BFD for IPv4

Cisco IOS XR software supports bidirectional forwarding detection (BFD) singlehop and multihop for both IPv4 and IPv6.

In BFD for IPv4 single-hop connectivity, Cisco IOS XR software supports both asynchronous mode and echo mode over physical numbered Packet-over-SONET/SDH (POS) and Gigabit Ethernet links, as follows:

- Echo mode is initiated only after a session is established using BFD control packets. Echo mode is always enabled for BFD bundle member interfaces. For physical interfaces, the BFD minimum interval must also be less than two seconds to support echo packets.
- BFD echo packets are transmitted over UDP/IPv4 using source and destination port 3785. The source address of the IP packet is the IP address of the output interface (default) or the address specified with the **router-id** command if set or the address specified in the **echo ipv4 source** command, and the destination address is the local interface address.
- BFD asynchronous packets are transmitted over UDP and IPv4 using source port 49152 and destination port 3784. For asynchronous mode, the source address of the IP packet is the local interface address, and the destination address is the remote interface address.

Note BFD multihop does not support echo mode.

Consider the following guidelines when configuring BFD on Cisco IOS XR software:

- BFD is a fixed-length hello protocol, in which each end of a connection transmits packets periodically over a forwarding path. Cisco IOS XR software supports BFD adaptive detection times.

- BFD can be used with the following applications:
  - BGP
  - IS-IS
  - EIGRP
  - OSPF

    and OSPFv3
  - MPLS Traffic Engineering (MPLS-TE)
  - Static routes (IPv4 and IPv6)
  - Protocol Independent Multicast (PIM)
  - Hot Standby Router Protocol (HSRP)
  - Virtual Router Redundancy Protocol (VRRP)

Note When multiple applications share the same BFD session, the application with the most aggressive timer wins locally. Then, the result is negotiated with the peer router.

- BFD is supported for connections over the following interface types:
  - Gigabit Ethernet (GigE)
  - Ten Gigabit Ethernet (TenGigE)
  - Packet-over-SONET/SDH (POS)
  - Serial
  - Virtual LAN (VLAN)
  - Bridge Group Virtual Interface (BVI)

Note VRF based BVI is supported from Cisco IOS XR, Release 6.4.1. Please refer to BFD Over Bridge Group Virtual Interface: Example, on page 79 for configuration details.

  - Satellite Interface
  - Logical interfaces such as bundles, GRE, PWHE

Note BFD is supported on the above interface types and not on logical interfaces unless specifically stated.

- Cisco IOS XR software supports BFD Version 0 and Version 1. BFD sessions are established using either version, depending upon the neighbor. BFD Version 1 is the default version and is tried initially for session creation.

BFD for IPv6

Cisco IOS XR software supports bidirectional forwarding detection (BFD) for both IPv4 and IPv6. Bidirectional forwarding detection (BFD) for IPv6 supports the verification of live connectivity on interfaces that use IPv6 addresses.

The live connectivity verification for both IPv4 and IPv6 interfaces is performed by the same services and processes. Both IPv4 and IPv6 BFD sessions can run simultaneously on the same line card.

The same features and configurations that are supported in BFD for IPv4 are also supported in BFD for IPv6

BFD on Bundled VLANs

BFD for IPv4 on bundled VLANS is supported using static routing, IS-IS, and OSPF. When running a BFD session on a bundled VLAN interface, the BFD session is active as long as the VLAN bundle is up.

As long as the VLAN bundle is active, the following events do not cause the BFD session to fail:

- Failure of a component link.
- Online insertion and removal (OIR) of a line card which hosts one or more of the component links.
- Addition of a component link (by configuration) to the bundle.
- Removal of a component link (by configuration) from the bundle.
- Shutdown of a component link.
- RP switchover.

Note For more information on configuring a VLAN bundle, see the *Configuring Link Bundling on the Cisco ASR 9000 Series Router* module.

Keep the following in mind when configuring BFD over bundled VLANs:

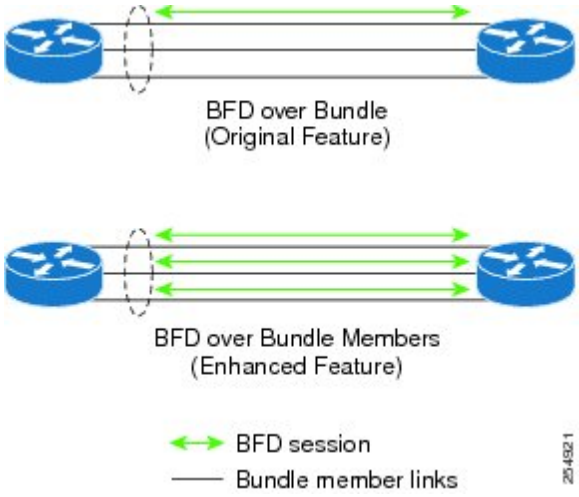
- In the case of an RP switchover, configured next-hops are registered in the Routing Information Base (RIB).
- In the case of a BFD restart, static routes remain in the RIB. BFD sessions are reestablished when BFD restarts.

Note Static BFD sessions are supported on peers with address prefixes whose next-hops are directly connected to the router.

BFD Over Member Links on Link Bundles

BFD supports BFD sessions on individual physical bundle member links to monitor Layer 3 connectivity on those links, rather than just at a single bundle member as in prior releases (Figure 37).

Figure 3: BFD Sessions in Original BFD Over Bundles and Enhanced BFD Over Bundle Member Links Architectures



When you run BFD on link bundles, you can run an independent BFD session on each underlying physical interface that is part of that bundle.

When BFD is running on a link bundle member, these layers of connectivity are effectively tested as part of the interface state monitoring for BFD:

- Layer 1 physical state
- Layer 2 Link Access Control Protocol (LACP) state
- Layer 3 BFD state

The BFD agent on each bundle member link monitors state changes on the link. BFD agents for sessions running on bundle member links communicate with a bundle manager. The bundle manager determines the state of member links and the overall availability of the bundle. The state of the member links contributes to the overall state of the bundle based on the threshold of minimum active links or minimum active bandwidth that is configured for that bundle.

Overview of BFD State Change Behavior on Member Links and Bundle Status

This section describes when bundle member link states are characterized as active or down, and their effect on the overall bundle status:

- You can configure BFD on a bundle member interface that is already active or one that is inactive. For the BFD session to be *up* using LACP on the interface, LACP must have reached the *distributing* state.

  A BFD member link is “IIR Active” if the link is in LACP distributing state and the BFD session is up.

- A BFD member link is “IIR Attached” when the BFD session is down, unless a LACP state transition is received.

- You can configure timers for up to 3600 seconds (1 hour) to allow for delays in receipt of BFD state change notifications (SCNs) from peers before declaring a link bundle BFD session down. The configurable timers apply to these situations:
  - BFD session startup (**bfd address-family ipv4 timers start** command)—Number of seconds to allow after startup of a BFD member link session for the expected notification from the BFD peer to be received to declare the session up. If the SCN is not received after that period of time, the BFD session is declared down.
  - Notification of removal of BFD configuration by a neighbor (**bfd address-family ipv4 timers nbr-unconfig** command)—Number of seconds to allow after receipt of notification that BFD configuration has been removed by a BFD neighbor so that any configuration inconsistency between the BFD peers can be fixed. If the BFD configuration issue is not resolved before the specified timer is reached, the BFD session is declared down.
- A BFD session sends a DOWN notification when one of these occurs:
  - The BFD configuration is removed on the local member link.

    The BFD system notifies the peer on the neighbor router that the configuration is removed. The BFD session is removed from the bundle manager without affecting other bundle member interfaces or the overall bundle state.
  - A member link is removed from the bundle.

    Removing a member link from a bundle causes the bundle member to be removed ungracefully. The BFD session is deleted and BFD on the neighboring router marks the session DOWN rather than NBR_CONFIG_DOWN.
- In these cases, a DOWN notification is not sent, but the internal infrastructure treats the event as if a DOWN has occurred:
  - The BFD configuration is removed on a neighboring router and the neighbor unconfiguration timer (if configured) expires.

    The BFD system notifies the bundle manager that the BFD configuration has been removed on the neighboring router and, if **bfd timers nbr-unconfig** is configured on the link, the timer is started. If the BFD configuration is removed on the local router before the timer expires, then the timer is stopped and the behavior is as expected for BFD configuration removal on the local router.

    If the timer expires, then the behavior is the same as for a BFD session DOWN notification.
  - The session startup timer expires before notification from the BFD peer is received.
- The BFD session on a bundle member sends BFD state change notifications to the bundle manager. Once BFD state change notifications for bundle member interfaces are received by the bundle manager, the bundle manager determines whether or not the corresponding bundle interface is usable.
- A threshold for the minimum number of active member links on a bundle is used by the bundle manager to determine whether the bundle remains active, or is down based on the state of its member links. 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.

  Whenever a member's state changes, the bundle manager determines if the number of active members is less than the minimum number of active links threshold. If so, then the bundle is placed, or remains,

in DOWN state. Once the number of active links reaches the minimum threshold then the bundle returns to UP state.

- Another threshold is configurable on the bundle and is used by the bundle manager to determine the minimum amount of active bandwidth to be available before the bundle goes to DOWN state. This is configured using the **bundle minimum-active bandwidth** command.
- The BFD server responds to information from the bundle manager about state changes for the bundle interface and notifies applications on that interface while also sending system messages and MIB traps.

  The minimum supported timer for BFD is 3 x 50ms.

BFD Multipath Sessions

BFD can be applied over virtual interfaces such as GRE tunnel interfaces, PWHE interfaces, or between interfaces that are multihops away as described in the BFD for MultiHop Paths section. These types of BFD sessions are referred to BFD Multipath sessions.

As long as one path to the destination is active, these events may or may not cause the BFD Multipath session to fail as it depends on the interval negotiated versus the convergence time taken to update forwarding plane:

- Failure of a path
- Online insertion or removal (OIR) of a line card which hosts one or more paths
- Removal of a link (by configuration) which constitutes a path
- Shutdown of a link which constitutes a path

You must configure **bfd mutlipath include location** *location-id* command to enable at least one line card for the underlying mechanism that can be used to send and receive packets for the multipath sessions.

If a BFD Multipath session is hosted on a line card that is being removed from the bfd multipath include configuration, online removed, or brought to maintenance mode, then BFD attempts to migrate all BFD Multipath sessions hosted on that line card to another one. In that case, static routes are removed from RIB and then the BFD session is established again and included to RIB.

In case of BFD multipath sessions, the input and output interface may change based on the routing table updates. If the multipath session BFD packets must get preferential treatment, then a QoS policy must be configured on the entire path, including the possible input and output interfaces of the router.

The QoS policy must classify ingress and egress BFD packets into priority level 1 or priority level 2 queue. Similar approach applies to BFD sessions on BVI and "BFD Over VLAN Over Bundle" (that is, BLB).

Example:

```
ipv4 access-list BFD
5 permit udp any any eq 4784
!
class-map match-any BFDCLASS
match access-group ipv4 BFD
!
policy-map BFD
class BFDCLASS
  priority level 1
  police rate 10 kbps
!
interface GigabitEthernet0/2/0/1
```

```
service-policy output BFD
service-policy input BFD
```

For more information on PW headend and its configuration, see *Implementing Virtual Private LAN Services* module in the . For more information on GRE, see *Implementing MPLS Layer 2 VPNs* module in

BFD for MultiHop Paths

BFD multihop (BFD-MH) is a BFD session between two addresses that are not on the same subnet. An example of BFD-MH 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 BFD multihop are external and internal BGP. BFD multihop supports BFD on arbitrary paths, which can span multiple network hops.

The BFD Multihop feature provides sub-second forwarding failure detection for a destination more than one hop, and up to 255 hops, away. The **bfd multihop ttl-drop-threshold** command can be used to drop BFD packets coming from neighbors exceeding a certain number of hops. BFD multihop is supported on all currently supported media-type for BFD singlehop.

Setting up BFD Multihop

A BFD multihop session is set up between a unique source-destination address pair provided by the client. A session can be set up between two endpoints that have IP connectivity. For BFD Multihop, IPv4 addresses in both global routing table and in a VRF is supported.

When BFD is used with BGP, the BFD session type (singlehop or multihop) is configured based on the BGP configuration. If you configure eBGP-multihop keyword, the BFD session will also run in multihop mode; otherwise the session will run in singlehop mode.

BFD over MPLS Traffic Engineering LSPs

Bidirectional Forwarding Detection (BFD) over MPLS Traffic Engineering Label Switched Paths (LSPs) feature in Cisco IOS XR Software detects MPLS Label Switched Path LSP data plane failures. Since the control plane processing required for BFD control packets is relatively smaller than the processing required for LSP Ping messages, BFD can be deployed for faster detection of data plane failure for a large number of LSPs.

The BFD over MPLS TE LSPs implementation in Cisco IOS XR Software is based on *RFC 5884: Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs).* LSP Ping is an existing mechanism for detecting MPLS data plane failures and for verifying the MPLS LSP data plane against the control plane. BFD can be used for for detecting MPLS data plane failures, but not for verifying the MPLS LSP data plane against the control plane. A combination of LSP Ping and BFD provides faster data plane failure detection on a large number of LSPs.

The BFD over MPLS TE LSPs is used for networks that have deployed MPLS as the multi service transport and that use BFD as fast failure detection mechanism to enhance network reliability and up time by using BFD as fast failure detection traffic black holing.

BFD over MPLS TE LSPs support:

- BFD async mode (BFD echo mode is not supported)
- IPv4 only, since MPLS core is IPv4
- BFD packets will carry IP DSCP 6 (Internet Control)
- Use of BFD for TE tunnel bring up, re-optimization, and path protection (Standby and FRR)

- Fastest detection time (100 ms x 3 = 300 ms)
- Optional Periodic LSP ping verification after BFD session is up
- Dampening to hold-down BFD failed path-option
- There are two ways in which the BFD packets from tail-end to head-end will be used:
  - BFD packets from tail-end to head-end will be IP routed (IPv4 Multihop - port# 4784)
  - BFD packets from tail-end to head-end will be Label Switched (port# 3784) if MPLS LDP is available in Core with label path from tail-end to head-end.

Echo Timer configuration for BFD on Bundle Interfaces

The echo timer configuration allows you to specify the minimum interval for echo packets on IPv4 BFD sessions on bundle member links. You can set the echo timer value globally using the **bfd echo ipv4 bundle-per-member minimum-interval** command. Use the **bfd address-family ipv4 echo minimum-interval** to locally set the minimum interval value for the bundle ethernet interface.

Note This feature is applicable only for Cisco standard BFD over bundle per-member link mode.

See the *BFD Commands* on *Cisco ASR 9000 Series Router module of Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference* guide for details on these commands.

The echo timer behavior with the global and local echo configuration combination is illustrated in the following table:

Table 2: Echo timer behavior with global and local echo configuration

Global echo min-interval value **Command: bfd echo ipv4 bundle-per-member minimum-interval**	**Local bundle ethernet interface specific echo min-inter value** **Command: bfd address-family ipv4 echo minimum-inter**
Not configured	Not Configured
Global value is lesser than Async * multiplier	Not Configured
Global value is greater than Async * multiplier	Not Configured
Not configured	Local is greater than Async * Multiplier
Not configured	Local is lesser than Async * Multiplier
Global is configured (any value)	Local is greater than Async * Multiplier
Global is configured (any value)	Local is lesser than Async * Multiplier

Note

- Multiplier in the table refers to the remote multiplier value.
- Async refers to the negotiated asynchronous minimum interval value.

When R5.3.0 devices have BoB sessions with devices running on versions lesser than R5.3.0, it is recommended to retain the default echo timer value or configure identical values on both the devices.

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 is a standard-based fast failure detection of link aggregation (LAG) member links that is interoperable between different platforms. Cisco ASR 9000 support both IETF mode and Cisco 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

Note In software mode, it is recommended to use greater than or equal to 150ms as the minimum timer interval.

```
/* 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.  */
/* Alternatively, you can configure Cisco mode. */
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
```

Bidirectional Forwarding Detection over Generic Routing Encapsulation

Bidirectional Forwarding Detection (BFD) over Generic Routing Encapsulation (GRE) feature enables detection of network failures more rapidly than existing GRE keepalives mechanisms. BFD establishes a session over the GRE tunnel whose end points are BFD peers. Though BFD brings down tunnel during failure detection, tunnel keepalive mechanism enables the recovery of the tunnel after fault clearance. BFD is supported only on IPv4 GRE tunnel mode.

The source and destination of BFD session will be the same as the IPv4 address of the GRE tunnel.

You cannot enable BFD on the GRE tunnel, if the tunnel keepalive is enabled, and vice versa.

GRE tunneling protocol encapsulates a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link between two routers at remote points over an IP internetwork. The GRE enables service providers that do not run MPLS in their Core network to provide VPN services.

BFD over GRE feature is not supported on Cisco ASR 9000 Series SPA Interface Processor-700.

BFD provides IPv4 single-hop version 1 asynchronous mode over GRE numbered interfaces according to RFC5880.

Configure Bidirectional Forwarding Detection over Generic Routing Encapsulation

The following section shows how to configure Bidirectional Forwarding Detection (BFD) over Generic Routing Encapsulation (GRE) feature.

Figure 4: BFD Over GRE

Configuration Example

Configure the following steps in PE1 router:

```
Router# configure
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0
Router(config-bfd)# exit
Router(config)# interface tunnel-ip 100
Router(config-if)# ipv4 address 10.0.0.1 255.255.255.252
Router(config-if)# tunnel source Loopback 100
Router(config-if)# tunnel destination 10.2.2.2
Router(config-if)# tunnel bfd destination 10.0.0.2
Router(config-if)# tunnel bfd minimum-interval 300
Router(config-if)# tunnel bfd multiplier 5
Router(config-if)# tunnel bfd period 5
Router(config-if)# tunnel bfd retry 2
Router(config-if)# commit
```

Configure the following steps in PE2 router:

```
Router# configure
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0
Router(config-bfd)# exit
Router(config)# interface tunnel-ip 100
Router(config-if)# ipv4 address 10.0.0.2 255.255.255.252
Router(config-if)# tunnel source Loopback 100
Router(config-if)# tunnel destination 10.1.1.1
Router(config-if)# tunnel bfd destination 10.0.0.1
Router(config-if)# tunnel bfd minimum-interval 300
Router(config-if)# tunnel bfd multiplier 5
Router(config-if)# tunnel bfd period 5
Router(config-if)# tunnel bfd retry 2
Router(config-if)# commit
```

Running Configuration

```
/* The following is the running configuration from PE1 Router */
bfd
 multipath include location 0/0/CPU0
!
interface tunnel-ip 100
 ipv4 address 100.0.0.1 255.255.255.252
 tunnel source Loopback 100
 tunnel destination 10.2.2.2
 tunnel bfd destination 10.0.0.2
 tunnel bfd minimum-interval 300
 tunnel bfd multiplier 5
 tunnel bfd period 5
 tunnel bfd retry 2

/* The following is the running configuration from PE2 Router */
```

```
bfd
 multipath include location 0/0/CPU0
!
interface tunnel-ip 100
 ipv4 address 100.0.0.2 255.255.255.252
 tunnel source Loopback 100
 tunnel destination 10.1.1.1
 tunnel bfd destination 10.0.0.1
 tunnel bfd minimum-interval 300
 tunnel bfd multiplier 5
 tunnel bfd period 5
 tunnel bfd retry 2
```

Verification

```
Router# show interfaces tunnel-ip 1
Mon Jul  9 10:54:06.952 IST
tunnel-ip1 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Tunnel
  Internet address is 20.1.1.2/24
  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)
     reliability 255/255, txload 2/255, rxload 2/255
  Encapsulation TUNNEL_IP,  loopback not set,
  Last link flapped 00:03:54
  Tunnel TOS 0
  Tunnel mode GRE IPV4
  Keepalive is enabled, interval 10 seconds, maximum retry 3
  Tunnel source 10.0.0.2, destination 10.1.1.1/32
  Tunnel TTL 255
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 3 packets/sec
  5 minute output rate 1000 bits/sec, 3 packets/sec
     999 packets input, 75088 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     1001 packets output, 51380 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets

Router# show bfd session interface tenGigE 0/1/1/0.200 detail

I/f: TenGigE0/1/1/0.200, Location: 0/0/CPU0
Dest: 10.1.1.2
Src: 10.0.0.2
State: UP for 0d:0h:6m:9s, number of times UP: 1
Session type: PR/V4/SH
Received parameters:
Version: 1, desired tx interval: 300 ms, required rx interval: 300 ms
Required echo rx interval: 0 ms, multiplier: 3, diag: None
My discr: 2148532226, your discr: 2148335671, state UP, D/F/P/C/A: 0/0/0/1/0
Transmitted parameters:
Version: 1, desired tx interval: 15 ms, required rx interval: 15 ms
Required echo rx interval: 0 ms, multiplier: 3, diag: None
My discr: 2148335671, your discr: 2148532226, state UP, D/F/P/C/A: 0/1/0/1/0
Timer Values:
Local negotiated async tx interval: 300 ms
Remote negotiated async tx interval: 300 ms
Desired echo tx interval: 0 s, local negotiated echo tx interval: 0 ms
Echo detection time: 0 ms(0 ms*3), async detection time: 900 ms(300 ms*3)
Local Stats:
Intervals between async packets:
   Tx: Number of intervals=4, min=1 ms, max=346 s, avg=88 s
```

```
       Last packet transmitted 23 s ago
   Rx: Number of intervals=11, min=1 ms, max=346 s, avg=32 s
       Last packet received 23 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
  -------------------- ---------- ---------- ---------- ----------
  tunl_gre_ma           15 ms      3          15 ms      3


Router# show bfd client

Mon Jul  9 10:55:16.025 IST
Name                  Node       Num sessions
--------------------- ---------- --------------
L2VPN_ATOM            0/0/CPU0   0
bundlemgr_distrib     0/0/CPU0   0
object_tracking       0/0/CPU0   0
pim6                  0/0/CPU0   0
pim                   0/0/CPU0   0
tunl_gre_ma           0/0/CPU0   1

Router# show tunnel ip keepalive
Mon Jul  9 10:54:30.005 IST

---- Tunnel GRE Keepalive Database ----

interface tunnel-ip1
 tunnel interface/basecaps state UP/UP
 tunnel ifhandle 0x90
 tunnel source 10.0.0.2
 tunnel destination 10.1.1.1
 tunnel transport vrf id 0x60000000
 tunnel transport vrf table id 0xe0000000
 tunnel ttl 255
 tunnel flags 0x1400
 tunnel keepalive max retries 3
 tunnel keepalive period 10
 tunnel keepalive state 0x2
 tunnel keepalive fail count 0
 tunnel keepalive packets sent 27
 Timestamp of last KA sent Mon Jul  9 10:54:21 2018
 tunnel keepalive packets received 24
  Timestamp of last KA received Mon Jul  9 10:54:21 2018
```

Associated Commands

- **bfd minimun-interval**
- **bfd multiplier**
- **tunnel bfd**

Bidirectional Forwarding Detection IPv6 Multihop

Bidirectional Forwarding Detection (BFD) IPv6 Multihop feature enables IPv6 Multihop BFD sessions where BFD neighbors can be multiple hops away, either physically or logically. More than one path is available to reach the BFD neighbor. BFD packets are received on a line card that may or may not host the respective BFD session. The BFD Agent in one line card may need to transmit BFD packets out of an egress interface on a different line card.

BFD support for IPv6 Multihop is on a par with the BFD IPv4 Multihop. The BFD IPv6 Multihop is supported on the ASR 9000 Ethernet Line Card and the ASR 9000 Enhanced Ethernet Line Card.

BFD IPv6 Multihop feature is not supported on Cisco ASR 9000 Series SPA Interface Processor-700.

BFD IPV6 Multihop removes the restriction of a single path IPv6 BFD session, where the BFD neighbor is always one hop away, and the BFD Agent in the line card always receives or transmits BFD packets over a local interface on the same line card.

The BFD switching mechanism for IPv6 Multihop link is employed when the BFD packets are transmitted from one end point node to the other. The BFD punting mechanism is employed when BFD packets are received at the remote end point node.

BFD over Pseudowire Headend

The Bidirectional Forwarding Detection over Pseudowire Headend (BFDoPWHE) feature enables BFD support over the customer edge (CE) to pseudowire headend (S-PE) links for fast failure detection along the path between the eBGP neighbors.

BFD over PWHE is supported only on ASR 9000 Enhanced Ethernet Line Card.

BFD over PWHE supports:

- BFD sessions per pseudo-wire for end-to-end fault detection between the CE and PWHE PE
- BFDv4 for IPv4 and BFDv6 for IPv6 (static and BGP)
- BFD asynchronous mode over PWHE
- Pseudowire VC type 4 and type 5

For PWHE to be operational, the BFD agent should be hosted on one of the line cards that is part of the PWHE generic interface list. The BFD multipath must be configured for a line card that is part of the generic interfaces list.

Use the **bfd multipath include location** *node-id* command to include specific line cards to host BFD multiple path sessions and thereby enable BFD over PWHE.

BFD over Satellite Interfaces

Bidirectional Forwarding Detection (BFD) over satellite interfaces feature enables BFD support on satellite line cards. Satellite interfaces are known as virtual (bundle) interfaces. BFD uses multipath infrastructure to support BFD on satellite line cards. BFD over satellite is a multipath (MP) single-hop session and is supported on IPv4 address, IPv6 global address, and IPv6 link-local address. BFD over Satellite is supported on Cisco ASR 9000 4th Generation QSFP28 based dense 100GE line cards, Cisco ASR 9000 5th Generation High-Density Multi-Rate line cards. BFD over satellite is not supported in echo mode.

Note

- BFD over Satellite Interfaces is not supported on nV Edge system.
- The nV Satellite access port bundles do not support BFD over bundles (BoB) over physical or bundle ICLs
- The **bfdmultipath include location** *node-id* command is required for all the line cards that host ICL links towards the Satellite.

BFD over IRB

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 in Cisco IOS XR Software Release 5.1.0 or greater, 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. The BFD over IRB feature is supported only on the ASR 9000 enhanced Ethernet line cards.

BFD over Bundle Per-Member Link

BFD over Bundle (BoB) Per-Member Link Mode is a standard-based fast failure detection of link aggregation (LAG) member links that is interoperable between different platforms. This provides an option to choose the per-member link mode to use either Cisco or IETF standard. This feature is supported only on Cisco ASR 9000 Enhanced Ethernet Line Card.

Note

- All the bundles in the system can belong to multiple mode at any single point in time.
- The global command for configuring BoB over bundle is available only up to release 5.3.0. For releases starting 5.3.1, you have the option to configure BFD over Bundles CISCO/IETF Mode support on a per bundle basis.

- The Cisco mode uses CDP MAC whereas IETF mode uses IANA assigned MAC.
- Cisco BFD over Bundle sessions use destination UDP port: 3784, while IETF BFD over Bundle sessions use destination UDP port: 6784.

Limitations

These limitations apply for the BFD over Bundle Per-Member Link Mode feature:

- Supported only on Cisco ASR 9000 Enhanced Ethernet Line Card.
- BFD Echo mode is not supported.
- IPv6 is supported in IETF mode, and not supported in CISCO mode.
- The mode change is applied only for new sessions. To apply mode change for existing sessions, delete and then recreate the sessions.
- A BFD session on the member interfaces can belong to only one mode (Cisco or IETF mode). Mix of the modes within the same bundle is not supported.

BFD over Bundles CISCO/IETF Mode Support on a Per Bundle Basis

BFD over Bundle (BoB) mode is a standard based fast failure detection of link aggregation (LAG) member links that is interoperable between different platforms. BoB support on a per bundle basis provides an option to choose either Cisco or IETF standard per bundle, without necessitating reloads or process restarts across various systems. The default is Cisco mode.

Note The global-level command available in previous releases to configure CISCO/IETF BoB over bundles is deprecated from release 5.3.1 onwards. In order to ensure a smooth upgrade, Cisco recommends that you configure the bundle at the interface level.

- The Cisco mode uses CDP MAC whereas IETF mode uses IANA assigned MAC.
- Cisco BFD over Bundle sessions use destination UDP port: 3784, while IETF BFD over Bundle sessions use destination UDP port: 6784.

Restrictions

These limitations apply for the BFD over Bundle Mode feature:

- Supported only on Cisco ASR 9000 Enhanced Ethernet Line Card.

- The BFD mode change (Cisco to IETF and vice-versa) goes through only when the BFD state for the bundle is 'down' or 'BoB nonoperational.'

Note You can use the **no bfd address-family ipv4 fast-detect** command to make BoB non-operational. You can also choose to configure a bundle to 'down' state by configuring shutdown under that particular bundle.

- For a bundle to accept the new BFD mode change, you must bring down and then recreate the existing BFD sessions.
- BFD Echo mode is not supported in IETF BFD over Bundle (BoB) sessions.

BFD Dampening

Bidirectional Forwarding Detection (BFD) is a mechanism used by routing protocols to quickly realize and communicate the reachability failures to their neighbors. When BFD detects a reachability status change of a client, its neighbors are notified immediately. Sometimes it might be critical to minimize changes in routing tables so as not to impact convergence, in case of a micro failure. An unstable link that flaps excessively can cause other devices in the network to consume substantial processing resources, and that can cause routing protocols to lose synchronization with the state of the flapping link.

The BFD Dampening feature introduces a configurable exponential delay mechanism. This mechanism is designed to suppress the excessive effect of remote node reachability events flapping with BFD. The BFD Dampening feature allows the network operator to automatically dampen a given BFD session to prevent excessive notification to BFD clients, thus preventing unnecessary instability in the network. Dampening the notification to a BFD client suppresses BFD notification until the time the session under monitoring stops flapping and becomes stable.

Configuring the BFD Dampening feature, especially on a high-speed interface with routing clients, improves convergence time and stability throughout the network. BFD dampening can be applied to all types of BFD sessions, including IPv4/single-hop/multihop, Multiprotocol Label Switching-Transport Profile (MPLS-TP), and Pseudo Wire (PW) Virtual Circuit Connection Verification (VCCV).

BFD Session Dampening

You can configure the BFD Dampening feature at the BFD template level (both single-hop and multihop templates). Dampening is applied to all the sessions that use the BFD template. If you choose not to have a session to be dampened, you should use a new BFD template without dampening for a new session. By default, the dampening functionality is not enabled on a template.

BFD Hardware Offload

The Bidirectional Forwarding Detection (BFD) Hardware Offload feature allows the offload of asynchronous BFD transmission (Tx) and reception (Rx) to the network processing unit on the ASR 9000 Enhanced Ethernet Line Card. BFD hardware offload improves the scale and reduces the overall network convergence time by sending rapid failure detection packets (messages) to the routing protocols for recalculating the routing table.

The following asynchronous BFD sessions are offloaded to the network processor unit on the ASR 9000 Enhanced Ethernet Line Card:

- BFD IPv4 sessions over physical and VLAN subinterfaces.
- BFD IPv6 sessions over physical and VLAN subinterfaces.
- BFD over MPLS-TP LSP Single-Path (SP) sessions.

Note Cisco ASR 9000 Fourth-Generation Ethernet line cards does not support BFD over MPLS-TP.

BFD hardware offload mode is enabled on the ASR 9000 Enhanced Ethernet line card using the **hw-module bfd-hw-offload enable** command. Configure the command in the admin mode in cXR devices and in the global configuration mode in eXR devices.

Note After enabling BFD hardware offload mode, you must reload the line card for the configuration change to take effect.

The BFD Hardware Offload feature supports specific timer intervals for BFD sessions, starting from 3.3 milliseconds up to 36 seconds. The following table lists timer interval values and the corresponding number of BFD sessions that are supported.

The BFD transmissions for Hardware offload happen only with the timer-intervals listed in the following table. If you configure an unsupported timer value using the **bfd minimum-interval** *milliseconds* command to bring up a BFD session, the session uses the next higher value. For example, if you enable a timer value of 100 ms, the timer is set to 300 ms since 100 ms is not a supported timer value.

Use the Hardware offload if you are using minimum timer intervals less than or equal to 50ms, else it is recommended to relax the BFD timers.

The minimum supported timer for BFD with hardware offload is 3 x 3.3ms. Enable hw-offload in the peer to support 3.3ms.

BFD Session	Timer Interval	Sessions supported on Line Card	Sessions supported on Network Processing Unit
IPv4, IPv6, MPLS-TP	3.3 milliseconds	600	300
IPv4, IPv6	15 milliseconds	2000	1000
IPv4, IPv6	50 milliseconds	8000	3000
IPv4, IPv6	300 milliseconds	8000	3000
IPv4, IPv6	1 second	8000	3000
IPv4, IPv6	2 seconds	8000	3000
IPv4, IPv6	12 seconds	8000	3000
IPv4, IPv6	36 seconds	8000	3000

Restrictions

- BFD hardware offload is supported on BFD over Bundle per Member Mode (BoB) only. BFD Over Member Links on Link Bundles (BLB) is not supported.
- Hardware offloaded sessions do not support echo mode.
- BFD sessions support only seven timer intervals.
- In-service software upgrade (ISSU) does not support BFD hardware offloaded sessions.
- Hardware offloaded BFD over the bundle member links does not support Cisco mode.
- Starting from Cisco IOS XR Software Release 6.6.2, Cisco ASR 9000 Fourth-Generation Ethernet line cards support BFD hardware offload.
- Starting from Cisco IOS XR Software Release 6.2.1, Cisco ASR 9000 Enhanced Ethernet Line Card and Cisco ASR 9000 High-Density 100GE Ethernet line cards support BFD hardware offload.
- Hardware offload BFD Over Bundle (BoB) doesn't support Cisco mode. It supports only IETF mode. Having an unsupported configuration could cause the features not to work properly.

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

How to Configure BFD

BFD Configuration Guidelines

Before you configure BFD, consider the following guidelines:

- FRR/TE, FRR/IP, and FRR/LDP using BFD is supported on POS interfaces and Ethernet interfaces.
- To establish a BFD neighbor in Cisco IOS XR software, BFD must either be configured under a dynamic routing protocol, or using a static route.
- The maximum rate in packets-per-second (pps) for BFD sessions is linecard-dependent. If you have multiple linecards supporting BFD, then the maximum rate for BFD sessions per system is the supported linecard rate multiplied by the number of linecards.

  To know the BFD scale values, use the **show bfd summary** command.
- The maximum number of members in a bundle is 64.
- When using BFD with OSPF, consider the following guidelines:
  - BFD establishes sessions from a neighbor to a designated router (DR) or backup DR (BDR) only when the neighbor state is *full*.
  - BFD does not establish sessions between DR-Other neighbors (for example, when their OSPF states are both 2-way).

Caution If you are using BFD with Unicast Reverse Path Forwarding (uRPF) on a particular interface, then you need to use the **echo disable** command to disable echo mode on that interface; otherwise, echo packets will be rejected. For more information, see the Disabling Echo Mode. To enable or disable IPv4 uRPF checking on an IPv4 interface, use the **[no] ipv4 verify unicast source reachable-via** command in interface configuration mode.

Note The **echo disable** command is not supported on BFD over logical bundle (BLB).

Configuring BFD Under a Dynamic Routing Protocol or Using a Static Route

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. To enable BFD per interface, use the steps in the Enabling BFD for OSPF on an Interface.

Note BFD neighbor router configuration is supported for BGP only.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:**	Enters BGP configuration mode, allowing you to configure the BGP routing process.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# router bgp 120`	Use the **show bgp** command in EXEC mode to obtain the *autonomous-system-number* for the current router.
Step 3	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-bgp)# bfd minimum-interval 6500`	Sets the BFD minimum interval. Range is 15-30000 milliseconds.
Step 4	**bfd multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config-bgp)# bfd multiplier 7`	Sets the BFD multiplier.
Step 5	**neighbor** *ip-address* **Example:** `RP/0/RSP0/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 6	**remote-as** *autonomous-system-number* **Example:** `RP/0/RSP0/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 7	**bfd fast-detect** **Example:** `RP/0/RSP0/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 5. In the example in Step 5, 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 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.

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 and MPLS-TE; only the command mode differs.

Note BFD per interface configuration is supported for OSPF, OSFPv3, IS-IS, and MPLS-TE only. For information about configuring BFD on an OSPFv3 interface, see Enabling BFD for OSPFv3 on an Interface.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd multipath include location***node-id* **Example:** `RP/0/RSP0/CPU0:router(config)# bfd multipath include location 0/0/CPU0`	(Optional) Enables BFD multipath for the specified bundle on the interface. This step is required for bundle interfaces. **Note** • This step must be repeated for every line card that has a member link in the bundle interface.
Step 3	**router ospf** *process-name* **Example:** `RP/0/RSP0/CPU0:router(config)# router ospf 0`	Enters OSPF configuration mode, allowing you to configure the OSPF routing process. Use the **show ospf** command in EXEC configuration mode to obtain the process-name for the current router. **Note** • To configure BFD for IS-IS or MPLS-TE, enter the corresponding configuration mode. For example, for MPLS-TE, enter MPLS-TE configuration mode.

	Command or Action	Purpose
Step 4	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# bfd minimum-interval 6500`	Sets the BFD minimum interval. Range is 15-30000 milliseconds. This example sets the BFD minimum interval to 6500 milliseconds.
Step 5	**bfd multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# bfd multiplier 7`	Sets the BFD multiplier. This example sets the BFD multiplier to 7.
Step 6	**area** *area-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospf)# area 0`	Configures an Open Shortest Path First (OSPF) area. Replace *area-id* with the OSPF area identifier.
Step 7	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospf-ar)# interface gigabitEthernet 0/3/0/1`	Enters interface configuration mode and specifies the interface name and notation *rack/slot/module/port*. • The example indicates a Gigabit Ethernet interface in modular services card slot 3.
Step 8	**bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-ospf-ar-if)# bfd fast-detect`	Enables BFD to detect failures in the path between adjacent forwarding engines.
Step 9	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 10	**show run router ospf** **Example:** `RP/0/RSP0/CPU0:router(config-ospf-ar-if)# show run router ospf`	Verify that BFD is enabled on the appropriate interface.

Enabling BFD for OSPFv3 on an Interface

The following procedures describe how to configure BFD for OSPFv3 on an interface. The steps in the procedure are common to the steps for configuring BFD on IS-IS, and MPLS-TE; only the command mode differs.

Note BFD per-interface configuration is supported for OSPF, OSPFv3, IS-IS, and MPLS-TE only. For information about configuring BFD on an OSPF interface, see Enabling BFD for OSPF on an Interface.

SUMMARY STEPS

1. **configure**
2. **router ospfv3** *process-name*
3. **bfd minimum-interval** *milliseconds*
4. **bfd multiplier** *multiplier*
5. **area** *area-id*
6. **interface** *type interface-path-id*
7. **bfd fast-detect**
8. Use the **commit** or **end** command.
9. **show run router ospfv3**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router ospfv3** *process-name* **Example:** `RP/0/RSP0/CPU0:routerconfig)# router ospfv3 0`	Enters OSPFv3 configuration mode, allowing you to configure the OSPFv3 routing process. Use the **show ospfv3** command in EXEC mode to obtain the process name for the current router. **Note** • To configure BFD for IS-IS or MPLS-TE, enter the corresponding configuration mode. For example, for MPLS-TE, enter MPLS-TE configuration mode.
Step 3	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3)# bfd minimum-interval 6500`	Sets the BFD minimum interval. Range is 15-30000 milliseconds. This example sets the BFD minimum interval to 6500 milliseconds.

	Command or Action	Purpose
Step 4	**bfd multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3)# bfd multiplier 7`	Sets the BFD multiplier. This example sets the BFD multiplier to 7.
Step 5	**area** *area-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3)# area 0`	Configures an OSPFv3 area. Replace *area-id* with the OSPFv3 area identifier.
Step 6	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3-ar)# interface gigabitEthernet 0/1/5/0`	Enters interface configuration mode and specifies the interface name and notation *rack/slot/module/port*. • The example indicates a Gigabit Ethernet interface in modular services card slot 1.
Step 7	**bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3-ar-if)# bfd fast-detect`	Enables BFD to detect failures in the path between adjacent forwarding engines.
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.
Step 9	**show run router ospfv3** **Example:** `RP/0/RSP0/CPU0:router(config-ospfv3-ar-if)#show run router ospfv3`	Verifies that BFD is enabled on the appropriate interface.

Enabling BFD on a Static Route

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

Note Bundle VLAN sessions are restricted to an interval of 250 milliseconds and a multiplier of 3. More aggressive parameters are not allowed.

SUMMARY STEPS

1. **configure**
2. **router static**
3. **address-family ipv4 unicast** *address nexthop* **bfd fast-detect** [**minimum-interval** *interval*] [**multiplier** *multiplier*]
4. **vrf** *vrf-name*
5. **address-family ipv4 unicast** *address nexthop* **bfd fast-detect**
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**router static** **Example:** `RP/0/RSP0/CPU0:router(config)# router static`	Enters static route configuration mode, allowing you to configure static routing.
Step 3	**address-family ipv4 unicast** *address nexthop* **bfd fast-detect** [**minimum-interval** *interval*] [**multiplier** *multiplier*] **Example:** `RP/0/RSP0/CPU0:router(config-static)# address-family ipv4 unicast 0.0.0.0/0 2.6.0.1 bfd fast-detect minimum-interval 1000 multiplier 5`	Enables BFD fast-detection on the specified IPV4 unicast destination address prefix and on the forwarding next-hop address. • Include the optional **minimum-interval** keyword and argument to ensure that the next-hop is assigned with the same hello interval. Replace the *interval* argument with a number that specifies the interval in milliseconds. Range is from 10 through 10000. • Include the optional **multiplier** keyword argument to ensure that the next hop is assigned with the same detect multiplier. Replace the *multiplier* argument with a number that specifies the detect multiplier. Range is from 1 through 10. **Note** Bundle VLAN sessions are restricted to an interval of 250 milliseconds and a multiplier of 3. More aggressive parameters are not allowed.

	Command or Action	Purpose
Step 4	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config-static)# vrf vrf1`	Specifies a VPN routing and forwarding (VRF) instance, and enters static route configuration mode for that VRF.
Step 5	**address-family ipv4 unicast** *address nexthop* **bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-static-vrf)# address-family ipv4 unicast 0.0.0.0/0 2.6.0.2`	Enables BFD fast-detection on the specified IPV4 unicast destination address prefix and on the forwarding next-hop address.
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: • **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 IPv6 Static Route

The below sample configuration describes how to enable BFD on a IPv6 static route:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router static
RP/0/RSP0/CPU0:router(config-static)# address-family ipv6 unicast 1011:17e4::1/128
ab11:15d2::2 bfd fast-detect minimum-interval 50 multiplier 3
RP/0/RSP0/CPU0:router(config-static)# commit
```

Configuring BFD on Bundle Member Links

Prerequisites for Configuring BFD on Bundle Member Links

The physical interfaces that are members of a bundle must be directly connected between peer routers without any switches in between.

Specifying the BFD Destination Address on a Bundle

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

DETAILED STEPS

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** | **Bundle-POS**] *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/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether**	**Bundle-POS**] *bundle-id* **Example:** `RP/0/RSP0/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/RSP0/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.

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-POS**] *bundle-id*
3. **bfd address-family ipv4 fast-detect**
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/RSP0/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/RSP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD sessions on bundle member links.
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/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RSP0/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/RSP0/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/RSP0/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 BFD echo packet interval and all failure detection times are determined by a combination of the interval and multiplier values in these commands. For more information see the BFD Packet Intervals and Failure Detection.

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

DETAILED STEPS

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** | **Bundle-POS**] *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/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether**	**Bundle-POS**] *bundle-id* **Example:** `RP/0/RSP0/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/RSP0/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 15 to 30000.Although the command allows you to configure a minimum of 15 ms, the supported minium on the Cisco ASR 9000 Series Router is 50 ms.	

	Command or Action	Purpose
Step 4	**bfd address-family ipv4 multiplier** *multiplier* **Example:** `RP/0/RSP0/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 and echo packet failure detection times and echo packet 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.

Configuring Allowable Delays for BFD State Change Notifications Using Timers on a Bundle

The BFD system supports two configurable timers to allow for delays in receipt of BFD SCNs from peers before declaring a BFD session on a link bundle member down:

- BFD session startup
- BFD configuration removal by a neighbor

For more information about how these timers work and other BFD state change behavior, see the Overview of BFD State Change Behavior on Member Links and Bundle Status.

To configure the timers that allow for delays in receipt of BFD SCNs from peers, complete these steps:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** | **Bundle-POS**] *bundle-id*
3. **bfd address-family ipv4 timers start** *seconds*
4. **bfd address-family ipv4 timers nbr-unconfig** *seconds*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd address-family ipv4 timers start** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#`	Specifies the number of seconds after startup of a BFD member link session to wait for the expected notification from the BFD peer to be received, so that the session can be declared up. If the SCN is not received after that period of time, the BFD session is declared down. The range is 60 to 3600. (In Cisco IOS XR Releases 4.0 and 4.0.1, the available minimum is 30, but is not recommended.)
Step 4	**bfd address-family ipv4 timers nbr-unconfig** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#`	Specifies the number of seconds to wait after receipt of notification that BFD configuration has been removed by a BFD neighbor, so that any configuration inconsistency between the BFD peers can be fixed. If the BFD configuration issue is not resolved before the specified timer is reached, the BFD session is declared down. The range is 30 to 3600.
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 over Bundle per Member Mode

To configure BFD over bundle per member link mode, complete the below steps:

Note This procedure is applicable for releases up to 5.3.0.

SUMMARY STEPS

1. **configure**
2. **bfd bundle per-member mode {cisco | ietf}**
3. **interface {bundle-ether | bundle-pos}** *bundle_ID*
4. **bfd address-family ipv4 fast-detect**
5. **bfd minimum-interval** *milliseconds*
6. **bfd multiplier** *multiplier*
7. **bfd address-family ipv4 destination** *ip-address*
8. **bfd address-family ipv4 timers start** *seconds*
9. **bfd address-family ipv4 timers nbr-unconfig** *seconds*
10. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd bundle per-member mode {cisco | ietf}** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd bundle per-member mode ietf`	Enables Cisco or IETF mode for BFD over per-bundle member link. Default is **cisco** .
Step 3	**interface {bundle-ether | bundle-pos}** *bundle_ID* **Example:** `RP/0/RSP0/CPU0:router(config)# interface bundle-either 1`	Enters interface configuration mode for the specified bundle ID.
Step 4	**bfd address-family ipv4 fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD sessions on bundle member links.
Step 5	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config)# bfd minimum-interval 15`	Sets the BFD minimum interval. Range is 15-30000 milliseconds.
Step 6	**bfd multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config)# bfd multiplier 2`	Sets the BFD multiplier.

	Command or Action	Purpose
Step 7	**bfd address-family ipv4 destination** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config)# 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 8	**bfd address-family ipv4 timers start** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# bfd address-family ipv4 timers start 60`	Specifies the number of seconds after startup of a BFD member link session to wait for the expected notification from the BFD peer to be received, so that the session can be declared up. If the state change notification is not received after that period of time, the BFD session is declared down. The range is 60 to 3600.
Step 9	**bfd address-family ipv4 timers nbr-unconfig** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# bfd address-family ipv4 timers nbr-unconfig 3600`	Specifies the number of seconds to wait after receipt of notification that BFD configuration has been removed by a BFD neighbor, so that any configuration inconsistency between the BFD peers can be fixed. If the BFD configuration issue is not resolved before the specified timer is reached, the BFD session is declared down. The range is 30 to 3600.
Step 10	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 CISCO/IETF Mode Support on a Per Bundle Basis

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

Before you begin

The BFD mode change (Cisco to IETF and vice-versa) goes through when a bundle is newly created or only when the BFD state for the bundle is 'down' or 'BoB nonoperational.'

Note This procedure is applicable from release 5.3.1 onwards.

SUMMARY STEPS

1. **configure**

2. **interface Bundle-Ether** *bundle-id*
3. **no bfd address-family ipv4 fast-detect**
4. Use the **commit** or **end** command.
5. **bfd mode { cisco | ietf }**
6. **bfd address-family ipv4 fast-detect**
7. Use the **commit** or **end** command.
8. **show bundle bundle-ether** *bundle-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**no bfd address-family ipv4 fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-if)# no bfd address-family ipv4 fast-detect`	Disables IPv4 BFD sessions on the specified bundle.
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.
Step 5	**bfd mode { cisco | ietf }** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd mode ietf`	Enables Cisco or IETF mode for BFD over bundle for the specified bundle. Default is **cisco** .
Step 6	**bfd address-family ipv4 fast-detect** **Example:**	Enables IPv4 BFD sessions on the specified bundle.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	
Step 7	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 8	**show bundle bundle-ether** *bundle-id*	Displays the selected bundle mode.

Sample show command output to check the mode

This example show the output of the **show bundle bundle-ether** command with the bundle mode selected:

```
RP/0/RP0/CPU0:R3-PE3#sh bundle bundle-ether 4301

Bundle-Ether4301
  Status:                                  Up
  Local links {active/standby/configured}:   2 / 0 / 2
  Local bandwidth {effective/available}:     20000000 (20000000) kbps
  MAC address (source):                    0014.1c00.0003 (Chassis pool)
  Inter-chassis link:                      No
  Minimum active links / bandwidth:        1 / 1 kbps
  Maximum active links:                    64
  Wait while timer:                        2000 ms
  Load balancing:                          Default
  LACP:                                    Operational
    Flap suppression timer:                Off
    Cisco extensions:                      Disabled
  mLACP:                                   Not configured
  IPv4 BFD:                                Operational
    State:                                 Up
    Mode:                                            ietf  ####------ this is the mode
cisco/ietf .
    Fast detect:                           Enabled
    Start timer:                           60 s
    Neighbor-unconfigured timer:           60 s
    Preferred min interval:                150 ms
    Preferred multiple:                    3
    Destination address:                   101.43.1.1

  Port                Device           State        Port ID         B/W, kbps
  ------------------  ---------------  -----------  --------------  ----------
  Te0/5/0/4           Local            Active       0x8000, 0x0012    10000000
      Link is Active
```

```
Te0/7/0/8            Local            Active       0x8000, 0x0006    10000000
    Link is Active
```

What to do next

For a bundle to accept the new BFD mode change, you must bring down and then recreate the existing BFD sessions.

Configuring BFD over Bundle for Hardware Offload

The following procedure explains how to configure the BFD Hardware Offload feature on bundle-ether interfaces with aggressive timers.

SUMMARY STEPS

1. **hw-module bfd-hw-offload enable location** *line-card-location*
2. **hw-module location** *node-id* **reload**
3. **interface bundle-Ether** *bundle-id*
4. **bfd mode ietf**
5. **bfd address-family ipv4 destination** *ip-address* **reload**
6. **bfd address-family ipv4 fast-detect**
7. **bfd address-family ipv4 minimum-interval** *milliseconds*
8. **bfd address-family ipv4 multiplier** *multiplier*
9. **ipv4 address** *ip-address mask*
10. **end**
11. **interface GigabitEthernet** *interface-path* **bundle id** *number* **mode active**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**hw-module bfd-hw-offload enable location** *line-card-location* **Example:** `RP/0/RSP0/CPU0:router(config)# hw-module bfd-hw-offload enable location 0/0/cpu0`	Configures BFD hardware offload mode.
Step 2	**hw-module location** *node-id* **reload** **Example:** `RP/0/RSP0/CPU0:router(config)# hw-module location 0/0/cpu0 reload`	Reloads the hardware on the specified node.
Step 3	**interface bundle-Ether** *bundle-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 4	**bfd mode ietf** **Example:**	Enables Cisco or IETF mode for BFD over bundle for the specified bundle.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-if)# bfd mode ietf`	
Step 5	**bfd address-family ipv4 destination** *ip-address* **reload** **Example:** `RP/0/RSP0/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 6	**bfd address-family ipv4 fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD session for the bundle.
Step 7	**bfd address-family ipv4 minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv4 minimum-interval 300`	Specifies the minimum interval, in milliseconds, for asynchronous mode control packets on IPv4 BFD sessions on bundle member links. The supported BFD Hardware Offload timer values are 3.3 ms, 15 ms, 50 ms, 300 ms, 1 second, 2 seconds, 12 seconds and 36 seconds. If you configure an unsupported timer value (for example 200 ms), then the next higher value (300 ms) is enabled.
Step 8	**bfd address-family ipv4 multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv4 multiplier 5`	Specifies a number that is used as a multiplier with the minimum interval to determine BFD control 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 recommended minimum is 3.
Step 9	**ipv4 address** *ip-address mask* **Example:** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3/30`	Assigns an IP address and subnet mask to the interface using the ipv4 address configuration subcommand.
Step 10	**end** **Example:** `RP/0/RSP0/CPU0:router(config-if)# end`	Applies the interface configuration and exits interface configuration mode.
Step 11	**interface GigabitEthernet** *interface-path* **bundle id** *number* **mode active** **Example:** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/3 bundle id 1 mode active`	Enters interface configuration mode for the specified bundle ID.

What to do next

Verify the BFD hardware offload feature configuration:

```
RP/0/RSP0/CPU0:router#show bfd hw-offload summary

BFD HW OFFLOAD Feature Summary:

0/0/CPU0
==========

The below available numbers per timer interval indicates the
max. sessions that can be configured at that interval without
configuring any other session at any other interval.

After configuring, execute this CLI to get the remaining
available numbers.

              3.3ms   15ms    50ms    300ms   1s      2s      12s     36s
-------------------------------------------------------------------------------
Max LC Supp   600     2000    8000    8000    8000    8000    8000    8000
Max NP Supp   300     1000    3000    3000    3000    3000    3000    3000

-------------------------------------------------------------------------------
LC:
----
Tx Used       0       0       0       4       0       0       0       0
Rx Used       0       0       0       4       0       0       0       0
Tx Avail      599     1999    7996    7996    7996    7996    7996    7996
Rx Avail      599     1999    7996    7996    7996    7996    7996    7996

NP0:
-----
Tx Used       0       0       0       4       0       0       0       0
Rx Used       0       0       0       4       0       0       0       0
Tx Avail      300     1000    2996    2996    2996    2996    2996    2996
Rx Avail      300     1000    2996    2996    2996    2996    2996    2996

NP1:
-----
Tx Used       0       0       0       0       0       0       0       0
Rx Used       0       0       0       0       0       0       0       0
Tx Avail      300     1000    3000    3000    3000    3000    3000    3000
Rx Avail      300     1000    3000    3000    3000    3000    3000    3000

NP2:
-----
Tx Used       0       0       0       0       0       0       0       0
Rx Used       0       0       0       0       0       0       0       0
Tx Avail      300     1000    3000    3000    3000    3000    3000    3000
Rx Avail      300     1000    3000    3000    3000    3000    3000    3000

NP3:
-----
Tx Used       0       0       0       0       0       0       0       0
Rx Used       0       0       0       0       0       0       0       0
Tx Avail      300     1000    3000    3000    3000    3000    3000    3000
Rx Avail      300     1000    3000    3000    3000    3000    3000    3000
```

Enabling Echo Mode to Test the Forwarding Path to a BFD Peer

BFD echo mode is enabled by default for the following interfaces:

- For IPv4 on member links of BFD bundle interfaces.
- For IPv4 on other physical interfaces whose minimum interval is less than two seconds.

Note If you have configured a BFD minimum interval greater than two seconds on a physical interface using the **bfd minimum-interval** command, then you will need to change the interval to be less than two seconds to support and enable echo mode. This does not apply to bundle member links, which always support echo mode.

Overriding the Default Echo Packet Source Address

If you do not specify an echo packet source address, then BFD uses the IP address of the output interface as the default source address for an echo packet.

In Cisco IOS XR releases before 3.9.0, we recommend that you configure the local router ID using the **router-id** command to change the default IP address for the echo packet source address to the adrdress specified as the router ID.

Beginning in Cisco IOS XR release 3.9.0 and later, you can use the **echo ipv4 source** command in BFD or interface BFD configuration mode to specify the IP address that you want to use as the echo packet source address.

You can override the default IP source address for echo packets for BFD on the entire router, or for a particular interface.

Specifying the Echo Packet Source Address Globally for BFD

To specify the echo packet source IP address globally for BFD on the router, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **echo ipv4 source** *ip-address*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.

	Command or Action	Purpose
Step 3	**echo ipv4 source** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# echo ipv4 source 10.10.10.1`	Specifies an IPv4 address to be used as the source address in BFD echo packets, 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.

Specifying the Echo Packet Source Address on an Individual Interface or Bundle

To specify the echo packet source IP address on an individual BFD interface or bundle, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **interface** type interface-path-id
4. **echo ipv4 source** *ip-address*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**interface** type interface-path-id **Example:**	Enters BFD interface configuration mode for a specific interface or bundle. In BFD interface configuration mode,

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-bfd)# interface gigabitEthernet 0/1/5/0`	you can specify an IPv4 address on an individual interface or bundle.
Step 4	**echo ipv4 source** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# echo ipv4 source 10.10.10.1`	Specifies an IPv4 address to be used as the source address in BFD echo packets, where *ip-address* is the 32-bit IP address in dotted-decimal format (A.B.C.D).
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 Session Teardown Based on Echo Latency Detection

Beginning in Cisco IOS XR 4.0.1, you can configure BFD sessions on non-bundle interfaces to bring down a BFD session when it exceeds the configured echo latency tolerance.

To configure BFD session teardown using echo latency detection, complete the following steps.

Before you enable echo latency detection, be sure that your BFD configuration supports echo mode.

Echo latency detection is not supported on bundle interfaces.

DETAILED STEPS

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **echo latency detect** [**percentage** *percent-value* [**count** *packet-count*]
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**echo latency detect** [**percentage** *percent-value* [**count** *packet-count*] **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# echo latency detect`	Enables echo packet latency detection over the course of a BFD session, where: • **percentage** *percent-value*—Specifies the percentage of the echo failure detection time to be detected as bad latency. The range is 100 to 250. The default is 100. • **count** *packet-count*—Specifies a number of consecutive packets received with bad latency that will take down a BFD session. The range is 1 to 10. The default is 1.
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.

Delaying BFD Session Startup Until Verification of Echo Path and Latency

Beginning in Cisco IOS XR Release 4.0.1, you can verify that the echo packet path is working and within configured latency thresholds before starting a BFD session on non-bundle interfaces.

Note Echo startup validation is not supported on bundle interfaces.

To configure BFD echo startup validation, complete the following steps.

Before you begin

Before you enable echo startup validation, be sure that your BFD configuration supports echo mode.

SUMMARY STEPS

1. **configure**
2. **bfd**

3. **echo startup validate** [**force**]
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/0RP0RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**echo startup validate** [**force**] **Example:** `RP/0/0RP0RSP0/CPU0:router(config-bfd)# echo startup validate`	Enables verification of the echo packet path before starting a BFD session, where an echo packet is periodically transmitted on the link to verify successful transmission within the configured latency before allowing the BFD session to change state. When the **force** keyword is not configured, the local system performs echo startup validation if the following conditions are true: • The local router is capable of running echo (echo is enabled for this session). • The remote router is capable of running echo (received control packet from remote system has non-zero “Required Min Echo RX Interval" value). When the force keyword is configured, the local system performs echo startup validation if following conditions are true. • The local router is capable of running echo (echo is enabled for this session). • The remote router echo capability is not considered (received control packet from remote system has zero or non-zero "Required Min Echo RX Interval" value).
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.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

Disabling Echo Mode

BFD does not support asynchronous operation in echo mode in certain environments. Echo mode should be disabled when using BFD for the following applications or conditions:

- BFD with uRPF (IPv4)
- To support rack reload and online insertion and removal (OIR) when a BFD bundle interface has member links that span multiple racks.

Note BFD echo mode is automatically disabled for BFD on physical interfaces when the minimum interval is greater than two seconds. The minimum interval does not affect echo mode on BFD bundle member links. BFD echo mode is also automatically disabled for BFD on bundled VLANs and IPv6 (global and link-local addressing).

You can disable echo mode for BFD on the entire router, or for a particular interface.

Disabling Echo Mode on a Router

To disable echo mode globally on the router complete the following steps:

DETAILED STEPS

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **echo disable**
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.

	Command or Action	Purpose
Step 3	**echo disable** **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# echo disable`	Disables echo mode on the router.
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.

Disabling Echo Mode on an Individual Interface or Bundle

The following procedures describe how to disable echo mode on an interface or bundle .

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **interface** *type interface-path-id*
4. **echo disable**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**interface** *type interface-path-id* **Example:**	Enters BFD interface configuration mode for a specific interface or bundle. In BFD interface configuration mode, you can disable echo mode on an individual interface or bundle.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-bfd)# interface gigabitEthernet 0/1/5/0`	
Step 4	**echo disable** **Example:** `RP/0/RSP0/CPU0:router(config-bfd-if)# echo disable`	Disables echo mode on the specified individual interface or 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.

Minimizing BFD Session Flapping Using BFD Dampening

To configure BFD dampening to control BFD session flapping, complete the following steps.

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **dampening** [**bundle-member**] {**initial-wait** | **maximum-wait** | **secondary-wait**} *milliseconds*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.

	Command or Action	Purpose
Step 3	**dampening** [**bundle-member**] {**initial-wait** | **maximum-wait** | **secondary-wait**} *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# dampening initial-wait 30000`	Specifies delays in milliseconds for BFD session startup to control flapping. The value for **maximum-wait** should be greater than the value for **initial-wait**. The dampening values can be defined for bundle member interfaces and for the non-bundle interfaces.
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.

Enabling and Disabling IPv6 Checksum Support

By default, IPv6 checksum calculations on UDP packets are enabled for BFD on the router.

You can disable IPv6 checksum support for BFD either on the entire router, or for a particular interface. A misconfiguration may occur if the IPv6 checksum support is enabled at one router, but disabled at the other. Therefore, you should enable or disable IPv6 checksum support at both the routers.

These sections describe about:

Note The command-line interface (CLI) is slightly different in BFD configuration and BFD interface configuration. For BFD configuration, the **disable** keyword is not optional. Therefore, to enable BFD configuration in that mode, you need to use the **no** form of the command.

Enabling and Disabling IPv6 Checksum Calculations for BFD on a Router

To enable or disable IPv6 checksum calculations globally on the router complete the following steps:

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **ipv6 checksum** [**disable**]
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**ipv6 checksum** [**disable**] **Example:** `RP/0/RSP0/CPU0:router(config-bfd-if)# ipv6 checksum disable`	Enables IPv6 checksum support on the interface. To disable, use the **disable** keyword.
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.

Enabling and Disabling IPv6 Checksum Calculations for BFD on an Individual Interface or Bundle

The following procedures describe how to enable or disable IPv6 checksum calculations on an interface or bundle .

DETAILED STEPS

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **interface** *type interface-path-id*
4. **ipv6 checksum** [**disable**]
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-bfd)# interface gigabitEthernet 0/1/5/0`	Enters BFD interface configuration mode for a specific interface or bundle.
Step 4	**ipv6 checksum** [**disable**] **Example:** `RP/0/RSP0/CPU0:router(config-bfd-if)# ipv6 checksum`	Enables IPv6 checksum support on the interface. To disable, use the **disable** keyword.
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.

Clearing and Displaying 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.

SUMMARY STEPS

1. **show bfd counters**[**ipv4 | all**] **packet interface** *type interface-path-id***] location** *node-id*
2. **clear bfd counters** [**ipv4 | ipv6 |all**] **packet** [**interface** *type interface-path-id*] **location** *node-id*
3. **show bfd counters** [**[ipv4 | ipv6 | all**] **packet** [**interface** *type interface-path-id*] **location** *node-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show bfd counters**[**ipv4 | all**] **packet interface** *type interface-path-id***] location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#show bfd counters all packet location 0/3/cpu0`	Displays the BFD counters for IPv4 packets, IPv6 packets, or all packets.
Step 2	**clear bfd counters** [**ipv4 | ipv6 |all**] **packet** [**interface** *type interface-path-id*] **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# clear bfd counters all packet location 0/3/cpu0`	Clears the BFD counters for IPv4 packets, IPv6 packets, or all packets.
Step 3	**show bfd counters** [**[ipv4 | ipv6 | all**] **packet** [**interface** *type interface-path-id*] **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show bfd counters all packet location 0/3/cpu0`	Verifies that the BFD counters for IPv4 packets, IPv6 packets, or all packets have been cleared.

BFD IPv6 in Bundle Manager Domain

A configuration to enable or disable BFD to run over a bundle interface can be in the bundle manager domain. The bundle manager can apply these configuration changes, and based on the configuration changes, request the BFD server to enable or disable BFD on certain bundle interfaces and a member links related to those bundle interfaces.

Configuration:

SUMMARY STEPS

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bfd address-family ipv6 fast-detect**
4. **bfd address-family ipv6 destination** *ip-address*
5. **bfd address-family ipv6 minimum-interval** *milliseconds*
6. **bfd address-family ipv6 multiplier** *multiplier*
7. **bfd address-family ipv6 timers start** *seconds*
8. **bfd address-family ipv6 timers nbr-unconfig** *seconds*
9. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface Bundle-Ether** *bundle-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 1`	Enters interface configuration mode for the specified bundle ID.
Step 3	**bfd address-family ipv6 fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv6 fast-detect`	Enables IPv6 BFD sessions on bundle member links.
Step 4	**bfd address-family ipv6 destination** *ip-address* **Example:** `RP/0/RSP0/CPU0:router(config-if)# bfd address-family ipv6 destination 2001:cdba:3257:9652`	Specifies the primary IPv6 address assigned to the bundle interface on a connected remote system.
Step 5	**bfd address-family ipv6 minimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd address-family ipv6 minimum-interval 2000`	
Step 6	**bfd address-family ipv6 multiplier** *multiplier* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd address-family ipv6 multiplier 30`	Specifies a number that is used as a multiplier with the minimum interval to determine BFD control and echo packet failure detection times and echo packet 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 7	**bfd address-family ipv6 timers start** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#`	Specifies the number of seconds after startup of a BFD member link session to wait for the expected notification from the BFD peer to be received, so that the session can be declared up. If the SCN is not received after that period of time, the BFD session is declared down. The range is 60 to 3600. (In Cisco IOS XR Releases 4.0 and 4.0.1, the available minimum is 30, but is not recommended.)

	Command or Action	Purpose
Step 8	**bfd address-family ipv6 timers nbr-unconfig** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#`	Specifies the number of seconds to wait after receipt of notification that BFD configuration has been removed by a BFD neighbor, so that any configuration inconsistency between the BFD peers can be fixed. If the BFD configuration issue is not resolved before the specified timer is reached, the BFD session is declared down. The range is 30 to 3600.
Step 9	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.

Example

Configuring BFD IPv6 Multihop

Configuring BFD IPv6 Multihop for eBGP Neighbors

Perform this task to configure BFD IPv6 multihop for eBGP neighbors.

SUMMARY STEPS

1. **configure**
2. **bfd multipath include location** *node-id*
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **ebgp-multihop** *ttl-value*
5. **neighbor** *ip-address* **bfd fast-detect**
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**bfd multipath include location** *node-id* **Example:** `RP/0/RSP0/CPU0:router(config)#bfd multipath include location 0/7/CPU0`	Includes specified line cards to host BFD multihop sessions.
Step 3	**router bgp** *as-number* **Example:** `RP/0/RSP0/CPU0:router(config)# router bgp 65001`	Enters BGP configuration mode.
Step 4	**neighbor** *ip-address* **ebgp-multihop** *ttl-value* **Example:** `RP/0/RSP0/CPU0:router(config-bgp)#neighbor 21:1:1:1:1:1:1:2 ebgp-multihop 255`	Enables multihop peerings with external BGP (eBGP) neighbors.
Step 5	**neighbor** *ip-address* **bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-bgp)#neighbor 21:1:1:1:1:1:1:2 bfd fast-detect`	Specifies IP address of the eBGP neighbor and enables BFD fast detection.
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: • **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 IPv6 Multihop for iBGP Neighbors

Perform this task to configure BFD IPv6 Multihop for iBGP neighbors:

SUMMARY STEPS

1. **configure**
2. **bfd multipath include location** *node-id*
3. **router bgp** *as-number*
4. **neighbor** *ip-address* **bfd fast-detect**
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd multipath include location** *node-id* **Example:** `RP/0/RSP0/CPU0:router(config)#bfd multipath include location 0/7/CPU0`	Includes specified line cards to host BFD multihop sessions.
Step 3	**router bgp** *as-number* **Example:** `RP/0/RSP0/CPU0:router(config)#router bgp 65001`	Enters BGP configuration mode.
Step 4	**neighbor** *ip-address* **bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-bgp)#neighbor 21:1:1:1:1:1:1:2`	Specifies IP address of the iBGP neighbor and enables BFD fast detection.
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 over MPLS Traffic Engineering LSPs

Enabling BFD Parameters for BFD over TE Tunnels

BFD for TE tunnel is enabled at the head-end by configuring BFD parameters under the tunnel. When BFD is enabled on the already up tunnel, TE waits for the bringup timeout before bringing down the tunnel. BFD is disabled on TE tunnels by default. Perform these tasks to configure BFD parameters and enable BFD over TE Tunnels.

Note BFD paces the creation of BFD sessions by limiting LSP ping messages to be under 50 PPS to avoid variations in CPU usage.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *interface-number*
3. **bfd fast-detect**
4. **bfd minimum-interval***milliseconds*
5. **bfd multiplier** *number*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *interface-number* **Example:** `RP/0/RSP0/CPU0:router(config)#interface tunnel-te 65535`	Configures MPLS Traffic Engineering (MPLS TE) tunnel interface and enters into MPLS TE tunnel interface configuration mode.
Step 3	**bfd fast-detect** **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd fast-detect`	Enables BFD fast detection.
Step 4	**bfd minimum-interval***milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd minimum-interval 2000`	Configures hello interval in milliseconds. Hello interval range is 100 to 30000 milliseconds. Default hello interval is 100 milliseconds
Step 5	**bfd multiplier** *number* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd multiplier 5`	Configures BFD multiplier detection. BFD multiplier range is 3 to 10. Default BFD multiplier is 3.
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: • **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.

What to do next

Configure BFD bring up timeout interval.

Once LSP is signaled and BFD session is created, TE allows given time for the BFD session to come up. If BFD session fails to come up within timeout, the LSP is torn down. Hence it is required to configure BFD bring up timeout

Configuring BFD Bring up Timeout

Perform these steps to configure BFD bring up timeout interval. The default bring up timeout interval is 60 seconds.

Before you begin

BFD must be enabled under MPLS TE tunnel interface.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *interface-number*
3. **bfd bringup-timeout** *seconds*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *interface-number* **Example:** `RP/0/RSP0/CPU0:router(config)#interface tunnel-te 65535`	Configures MPLS Traffic Engineering (MPLS TE) tunnel interface and enters into MPLS TE tunnel interface configuration mode.
Step 3	**bfd bringup-timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd bringup-timeout 2400`	Enables the time interval (in seconds) to wait for the BFD session to come up. Bring up timeout range is 6 to 3600 seconds. Default bring up timeout interval is 60 seconds.
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.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Configure BFD dampening parameters to bring up the TE tunnel and to avoid signaling churn in the network.

Configuring BFD Dampening for TE Tunnels

When BFD session fails to come up, TE exponentially backs off using the failed path-option to avoid signaling churn in the network.

Perform these steps to configure dampening intervals to bring the TE tunnel up.

Before you begin

- BFD must be enabled under MPLS TE tunnel interface.
- BFD bring up timeout interval must be configured using the **bfd bringup-timeout** command.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *interface-number*
3. **bfd dampening initial-wait** *milliseconds*
4. **bfd dampening maximum-wait** *milliseconds*
5. **bfd dampening secondary-wait** *milliseconds*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *interface-number* **Example:** `RP/0/RSP0/CPU0:router(config)#interface tunnel-te 65535`	Configures MPLS Traffic Engineering (MPLS TE) tunnel interface and enters into MPLS TE tunnel interface configuration mode.
Step 3	**bfd dampening initial-wait** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd dampening initial-wait 360000`	Configures the initial delay interval before bringing up the tunnel. The initial-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default initial-wait interval is 16000 milliseconds.

	Command or Action	Purpose
		Note This option brings up the TE tunnel with the previous signaled bandwidth.
Step 4	**bfd dampening maximum-wait** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd dampening maximum-wait 700000`	Configures the maximum delay interval before bringing up the tunnel. The maximum-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default initial-wait interval is 600000 milliseconds. **Note** This option brings up the TE tunnel with the configured bandwidth.
Step 5	**bfd dampening secondary-wait** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config-if)#bfd dampening secondary-wait 30000`	Configures the secondary delay interval before bringing up the tunnel. The secondary-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default secondary-wait interval is 20000 milliseconds.
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: • **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.

What to do next

Configure periodic LSP ping option.

Configuring Periodic LSP Ping Requests

Perform this task to configure sending periodic LSP ping requests with BFD TLV, after BFD session comes up.

Before you begin

BFD must be enabled under MPLS TE tunnel interface.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-te** *interface-number*
3. Use one of these commands:

- **bfd lsp-ping interval** *300*

4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-te** *interface-number* **Example:** `RP/0/RSP0/CPU0:router(config)#interface tunnel-te 65535`	Configures MPLS Traffic Engineering (MPLS TE) tunnel interface and enters into MPLS TE tunnel interface configuration mode.

Step 3	Use one of these commands:
	• **bfd lsp-ping interval** *300*
	Example:
	`RP/0/RSP0/CPU0:router(config-if)#bfd lsp-ping interval 300`
	Or
	`RP/0/RSP0/CPU0:router(config-if)#bfd lsp-ping disable`
		Sets periodic interval for LSP ping requests or disables LSP ping requests.
		• **interval** *seconds*—Sets periodic LSP ping request interval in seconds. The interval range is 60 to 3600 seconds. Default interval is 120 seconds.
		• **disable**—Disables periodic LSP ping requests.
		Periodic LSP ping request is enabled by default. The default interval for ping requests is 120 seconds. BFD paces LSP ping to be under 50 ping per seconds (PPS). Thus ping interval is honored; however, this is not guaranteed unless configuring an interval between 60 and 3600 seconds.

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.

What to do next

Configure BFD at the tail-end.

Configuring BFD at the Tail End

Use the tail end global configuration commands to set the BFD minimum-interval and BFD multiplier parameters for all BFD over LSP sessions. The ranges and default values are the same as the BFD head end configuration values. BFD will take the maximum value set between head end minimum interval and tail end minimum interval.

Perform these tasks to configure BFD at the tail end.

SUMMARY STEPS

1. **configure**
2. **mpls traffic-eng bfd lsp tailminimum-interval** *milliseconds*
3. **mpls traffic-eng bfd lsp tailmultiplier** *number*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**mpls traffic-eng bfd lsp tailminimum-interval** *milliseconds* **Example:** `RP/0/RSP0/CPU0:router(config)#mpls traffic-eng bfd lsp tail minimum-interval 20000`	Configures hello interval in milliseconds. Hello interval range is 100 to 30000 milliseconds. Default hello interval is 100 milliseconds
Step 3	**mpls traffic-eng bfd lsp tailmultiplier** *number* **Example:** `RP/0/RSP0/CPU0:router(config)#mpls traffic-eng bfd lsp tail multiplier 5`	Configures BFD multiplier detection. BFD multiplier detect range is 3 to 10. Default BFD multiplier is 3.
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.

What to do next

Configure **bfd multipath include location** *node-id* command to include specified line cards to host BFD multiple path sessions.

Configuring BFD over LSP Sessions on Line Cards

BFD over LSP sessions, both head-end and tail-end, will be hosted on line cards with following configuration enabled.

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **multipath include location** *node-id*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bfd** **Example:** `RP/0/RSP0/CPU0:router(config)# bfd`	Enters BFD configuration mode.

Step 3	**multipath include location** *node-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-bfd)# multipath include location 0/1/CPU0`
		Configures BFD multiple path on specific line card.
		One or more line cards must be configured with bfd multipath include. For example,
		`bfd`
		` multipath include location 0/1/CPU0`
		` multipath include location 0/2/CPU0`
		BFD over LSP sessions, both head-end and tail-end, will be hosted on line cards. BFD over LSP sessions, both head-end and tail-end, will be distributed to line cards 0/1/CPU0 and 0/2/CPU0 according to internal selection mechanism.

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.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring BFD Object Tracking:

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type bfdrtr rate** *tx-rate*
4. **debounce***debounce*
5. **interface** *if-name*
6. **destaddress** *dest_addr*
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type bfdrtr rate** *tx-rate* **Example:** `RP/0/RSP0/CPU0:router(config-track)# type bfdrtr rate 4`	tx_rate - time in msec at which the BFD should probe the remote entity
Step 4	**debounce***debounce* **Example:** `RP/0/RSP0/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:**	if_name - interface name on the source to be used by BFD to check the remote BFD status.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-track-line-prot)# interface atm 0/2/0/0.1`	
Step 6	**destaddress** *dest_addr* **Example:** `RP/0/RSP0/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. **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.

Configuration Examples for Configuring BFD

BFD Over BGP: Example

The following example shows how to configure BFD between autonomous system 65000 and neighbor 192.168.70.24:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router bgp 65000
RP/0/RSP0/CPU0:router(config-bgp)#bfd multiplier 2
RP/0/RSP0/CPU0:router(config-bgp)#bfd minimum-interval 20
RP/0/RSP0/CPU0:router(config-bgp)#neighbor 192.168.70.24
RP/0/RSP0/CPU0:router(config-bgp-nbr)#remote-as 2
RP/0/RSP0/CPU0:router(config-bgp-nbr)#bfd fast-detect
RP/0/RSP0/CPU0:router(config-bgp-nbr)#commit
RP/0/RSP0/CPU0:router(config-bgp-nbr)#end
RP/0/RSP0/CPU0:router#show run router bgp
```

BFD Over OSPF: Examples

The following example shows how to enable BFD for OSPF on a Gigabit Ethernet interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router ospf 0
RP/0/RSP0/CPU0:router(config-ospf)#area 0
RP/0/RSP0/CPU0:router(config-ospf-ar)#interface gigabitEthernet 0/3/0/1
```

```
RP/0/RSP0/CPU0:router(config-ospf-ar-if)#bfd fast-detect
RP/0/RSP0/CPU0:router(config-ospf-ar-if)#commit
RP/0/RSP0/CPU0:router(config-ospf-ar-if)#end

RP/0/RSP0/CPU0:router#show run router ospf

router ospf 0
area 0
interface GigabitEthernet0/3/0/1
bfd fast-detect
```

The following example shows how to enable BFD for OSPFv3 on a Gigabit Ethernet interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router ospfv3 0
RP/0/RSP0/CPU0:router(config-ospfv3)#bfd minimum-interval 6500
RP/0/RSP0/CPU0:router(config-ospfv3)#bfd multiplier 7
RP/0/RSP0/CPU0:router(config-ospfv3-ar)#area 0
RP/0/RSP0/CPU0:router(config-ospfv3-ar)#interface gigabitethernet 0/1/5/0
RP/0/RSP0/CPU0:router(config-ospfv3-ar-if)#bfd fast-detect
RP/0/RSP0/CPU0:router(config-ospfv3-ar-if)#commit
RP/0/RSP0/CPU0:router(config-ospfv3-ar-if)#end

RP/0/RSP0/CPU0:router#show run router ospfv3
router ospfv3
area 0
interface GigabitEthernet0/1/5/0
bfd fast-detect
```

BFD Over Static Routes: Examples

The following example shows how to enable BFD on an IPv4 static route. In this example, BFD sessions are established with the next-hop 10.3.3.3 when it becomes reachable.

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router static
RP/0/RSP0/CPU0:router(config-static)#address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-static)#10.2.2.0/24 10.3.3.3 bfd fast-detect
RP/0/RSP0/CPU0:router(config-static)#end
```

The following example shows how to enable BFD on an IPv6 static route. In this example, BFD sessions are established with the next hop 2001:0DB8:D987:398:AE3:B39:333:783 when it becomes reachable.

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router static
RP/0/RSP0/CPU0:router(config-static)#address-family ipv6 unicast
RP/0/RSP0/CPU0:router(config-static)#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/RSP0/CPU0:router(config-static)#end

RP/0/RSP0/CPU0:router#show run router static address-family ipv6 unicast
```

BFD on Bundled VLANs: Example

The following example shows how to configure BFD on bundled VLANs:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#multipath include location 0/0/CPU0
RP/0/RSP0/CPU0:router(config-bfd)#exit

RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#interface Bundle-ether 1
RP/0/RSP0/CPU0:router(config-if)#bundle maximum-active links 1
RP/0/RSP0/CPU0:router(config-if)#exit
!
RP/0/RSP0/CPU0:router(config)#interface TenGigE 0/1/0/1
RP/0/RSP0/CPU0:router(config-if)#bundle id 1 mode active
RP/0/RSP0/CPU0:router(config-if)#exit
!
RP/0/RSP0/CPU0:router(config)#interface TenGigE 0/2/0/1
RP/0/RSP0/CPU0:router(config-if)#bundle id 1 mode active
RP/0/RSP0/CPU0:router(config-if)#exit
!
RP/0/RSP0/CPU0:router(config)#interface Bundle-Ether1.2
RP/0/RSP0/CPU0:router(config-if)#ipv4 address 172.16.2.1 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)#encapsulation dot1q 2
RP/0/RSP0/CPU0:router(config-if)#exit
!
RP/0/RSP0/CPU0:router(config)#interface Bundle-Ether1.1
RP/0/RSP0/CPU0:router(config-if)#ipv4 address 172.16.1.1 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)#encapsulation dot1q 1
!
RP/0/RSP0/CPU0:router(config)#router static
RP/0/RSP0/CPU0:router(config-static)#address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-static-afi)#10.2.1.0/24 172.16.1.2 bfd fast-detect
minimum-interval 250
RP/0/RSP0/CPU0:router(config-static-afi)#10.2.2.0/24 172.16.2.2 bfd fast-detect
minimum-interval 250
RP/0/RSP0/CPU0:router(config-static-afi)#10.2.3.0/24 172.16.3.2 bfd fast-detect
minimum-interval 250
RP/0/RSP0/CPU0:router(config-static-afi)#exit
RP/0/RSP0/CPU0:router(config-static)#exit
!
```

BFD Over Bridge Group Virtual Interface: Example

The following examples show the configurations of the peer and uut nodes. You can see the BVI interface is under a VRF instead of default table:

```
interface BVI100
vrf cctv1 <<<<<<<<<<
```

Below is the peer nodes example:

```
l2vpn
 bridge group bg
  bridge-domain bd
   interface Bundle-Ether1.100
```

```
   !
   routed interface BVI100
  !
 !
!
router vrrp
 interface BVI100
  bfd minimum-interval 15
  address-family ipv4
   vrrp 100
    address 192.168.1.254
    bfd fast-detect peer ipv4 192.168.1.2
   !
  !
 !
!
router ospf 100
vrf cctv1
  router-id 192.168.1.1
  area 0
   interface BVI100
   !
  !
 !
!
interface BVI100
 vrf cctv1
 ipv4 address 192.168.1.1 255.255.255.0
!
interface GigE0/1/0/10
 bundle id 1 mode active
 no shut
!
interface Bundle-Ether1
no shut
!
interface Bundle-Ether1.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric

!
bfd multipath include loc 0/1/cpu0

interface MgmtEth0/RSP1/CPU0/0
 ipv4 address 7.37.19.20 255.255.0.0
 no shutdown
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 7.37.0.1
```

Below is the uut node example:

```
l2vpn
 bridge group bg
  bridge-domain bd
   interface Bundle-Ether1.100
   !
   routed interface BVI100
  !
 !
!
```

```
router vrrp
 interface BVI100
  bfd minimum-interval 15
  address-family ipv4
   vrrp 100
    address 192.168.1.254
    bfd fast-detect peer ipv4 192.168.1.1
   !
  !
 !
!
router ospf 100
 vrf cctv1
  router-id 192.168.1.2
  area 0
   interface BVI100
   !
  !
 !
!
interface BVI100
 vrf cctv1
 ipv4 address 192.168.1.2 255.255.255.0
!


interface GigE0/1/0/0
 bundle id 1 mode active
 no shut
 !
 interface Bundle-Ether1
 no shut
!
interface Bundle-Ether1.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric

!
bfd multipath include location 0/1/CPU0
```

BFD on Bundle Member Links: Examples

The following example shows how to configure BFD on member links of Ethernet bundle interfaces:

```
bfd
 interface Bundle-Ether4
  echo disable
 !
 interface GigabitEthernet0/0/0/2.3
  echo disable
 !
!
interface GigabitEthernet0/0/0/3 bundle id 1 mode active
interface GigabitEthernet0/0/0/4 bundle id 2 mode active
interface GigabitEthernet0/1/0/2 bundle id 3 mode active
interface GigabitEthernet0/1/0/3 bundle id 4 mode active
interface Bundle-Ether1
 ipv4 address 192.168.1.1/30
 bundle minimum-active links 1
!
interface Bundle-Ether1.1
```

```
 ipv4 address 192.168.100.1/30
 encapsulation dot1q 1001
!
interface Bundle-Ether2
 bfd address-family ipv4 destination 192.168.2.2
 bfd address-family ipv4 fast-detect
 bfd address-family ipv4 min 83
 bfd address-family ipv4 mul 3
 ipv4 address 192.168.2.1/30
 bundle minimum-active links 1
!
interface Bundle-Ether3
 bfd address-family ipv4 destination 192.168.3.2
 bfd address-family ipv4 fast-detect
 bfd address-family ipv4 min 83
 bfd address-family ipv4 mul 3
 ipv4 address 192.168.3.1/30
 bundle minimum-active links 1
!
interface Bundle-Ether4
 bfd address-family ipv4 destination 192.168.4.2
 bfd address-family ipv4 fast-detect
 bfd address-family ipv4 min 83
 bfd address-family ipv4 mul 3
 ipv4 address 192.168.4.1/30
 bundle minimum-active links 1
!
interface GigabitEthernet 0/0/0/2
 ipv4 address 192.168.10.1/30
!
interface GigabitEthernet 0/0/0/2.1
 ipv4 address 192.168.11.1/30
ipv6 address beef:cafe::1/64
 encapsulation dot1q 2001
!
interface GigabitEthernet 0/0/0/2.2
 ipv4 address 192.168.12.1/30
 encapsulation dot1q 2002
!
interface GigabitEthernet 0/0/0/2.3
 ipv4 address 192.168.13.1/30
 encapsulation dot1q 2003
!
router static
 address-family ipv4 unicast
  10.10.11.2/32 192.168.11.2 bfd fast-detect minimum-interval 250 multiplier 3
  10.10.12.2/32 192.168.12.2 bfd fast-detect minimum-interval 250 multiplier 3
  10.10.13.2/32 192.168.13.2 bfd fast-detect minimum-interval 250 multiplier 3
  10.10.100.2/32 192.168.100.2 bfd fast-detect minimum-interval 250 multiplier 3
!
 address-family ipv6 unicast
  babe:cace::2/128 beef:cafe::2 bfd fast-detect minimum-interval 250 multiplier 3
 !
```

Echo Packet Source Address: Examples

The following example shows how to specify the IP address 10.10.10.1 as the source address for BFD echo packets for all BFD sessions on the router:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
```

```
RP/0/RSP0/CPU0:router(config-bfd)#echo ipv4 source 10.10.10.1
```

The following example shows how to specify the IP address 10.10.10.1 as the source address for BFD echo packets on an individual Gigabit Ethernet interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#interface gigabitethernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-bfd-if)#echo ipv4 source 10.10.10.1
```

The following example shows how to specify the IP address 10.10.10.1 as the source address for BFD echo packets on an individual Packet-over-SONET (POS) interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#interface pos 0/1/0/0
RP/0/RSP0/CPU0:router(config-bfd-if)#echo ipv4 source 10.10.10.1
```

Echo Latency Detection: Examples

In the following examples, consider that the BFD minimum interval is 50 ms, and the multiplier is 3 for the BFD session.

The following example shows how to enable echo latency detection using the default values of 100% of the echo failure period (I x M) for a packet count of 1. In this example, when one echo packet is detected with a roundtrip delay greater than 150 ms, the session is taken down:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#echo latency detect
```

The following example shows how to enable echo latency detection based on 200% (two times) of the echo failure period for a packet count of 1. In this example, when one packet is detected with a roundtrip delay greater than 300 ms, the session is taken down:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#echo latency detect percentage 200
```

The following example shows how to enable echo latency detection based on 100% of the echo failure period for a packet count of 3. In this example, when three consecutive echo packets are detected with a roundtrip delay greater than 150 ms, the session is taken down:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#echo latency detect percentage 100 count 3
```

Echo Startup Validation: Examples

The following example shows how to enable echo startup validation for BFD sessions on non-bundle interfaces if the last received control packet contains a non-zero “Required Min Echo RX Interval” value:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#echo startup validate
```

The following example shows how to enable echo startup validation for BFD sessions on non-bundle interfaces regardless of the “Required Min Echo RX Interval” value in the last control packet:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#echo startup validate force
```

BFD Echo Mode Disable: Examples

The following example shows how to disable echo mode on a router:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:routerr(config-bfd)#echo disable
```

The following example shows how to disable echo mode on an interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#interface gigabitethernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-bfd-if)#echo disable
```

BFD Dampening: Examples

The following example shows how to configure an initial and maximum delay for BFD session startup on BFD bundle members:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#dampening bundle-member initial-wait 8000
RP/0/RSP0/CPU0:router(config-bfd)#dampening bundle-member maximum-wait 15000
```

The following example shows how to change the default initial-wait for BFD on a non-bundle interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#dampening initial-wait 30000
RP/0/RSP0/CPU0:router(config-bfd)#dampening maximum-wait 35000
```

BFD IPv6 Checksum: Examples

The following example shows how to disable IPv6 checksum calculations for UDP packets for all BFD sessions on the router:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#ipv6 checksum disable
```

The following example shows how to reenable IPv6 checksum calculations for UDP packets for all BFD sessions on the router:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#no ipv6 checksum disable
```

The following example shows how to enable echo mode for BFD sessions on an individual interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#interface gigabitethernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-bfd-if)#ipv6 checksum
```

The following example shows how to disable echo mode for BFD sessions on an individual interface:

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#bfd
RP/0/RSP0/CPU0:router(config-bfd)#interface gigabitethernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-bfd-if)#ipv6 checksum disable
```

BFD Peers on Routers Running Cisco IOS and Cisco IOS XR Software: Example

The following example shows how to configure BFD on a router interface on Router 1 that is running Cisco IOS software, and use the **bfd neighbor** command to designate the IP address 192.0.2.1 of an interface as its BFD peer on Router 2. Router 2 is running Cisco IOS XR software and uses the **router static** command and **address-family ipv4 unicast** command to designate the path back to Router 1's interface with IP address 192.0.2.2.

Router 1 (Cisco IOS software)

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#interface GigabitEthernet8/1/0
RP/0/RSP0/CPU0:router(config-if)#description to-TestBed1 G0/0/0/0
RP/0/RSP0/CPU0:router(config-if)#ip address 192.0.2.2 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)#bfd interval 100 min_rx 100 multiplier 3
RP/0/RSP0/CPU0:router(config-if)#bfd neighbor 192.0.2.1
```

Router 2 (Cisco IOS XR Software)

```
RP/0/RSP0/CPU0:router#configure
RP/0/RSP0/CPU0:router(config)#router static
RP/0/RSP0/CPU0:router(config-static)#address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-static-afi)#10.10.10.10/32 192.0.2.2 bfd fast-detect
```

```
RP/0/RSP0/CPU0:router(config-static-afi)#exit
RP/0/RSP0/CPU0:router(config-static)#exit
RP/0/RSP0/CPU0:router(config)#interface GigabitEthernet0/0/0/0
RP/0/RSP0/CPU0:router(config-if)#ipv4 address 192.0.2.1 255.255.255.0
```

BFD Over Bundle Hardware Offload: Example

The following example shows that the BFD has been hardware-offloaded. The value "Yes" under the H/W column indicates that the HW-offloaded BFD session for the bundle, that is, the BFD over Bundle hw-offload feature is configured and is operational.

```
RP/0/RSP0/CPU0:router#show bfd session

KInterface            Dest Addr           Local det time(int*mult)      State
                                        Echo             Async   H/W       NPU
-------------------- --------------- ---------------- ---------------- ----------
Te0/0/0/0/9          100.1.1.2       0s(0s*0)         6s(2s*3)          UP
                                                            Yes       0/0/CPU0/NPU0

BE10                 100.1.1.2       n/a              n/a               UP
```

The following example shows hardware offload info for node '0/0/CPU0/NPU3'. The NPU3 is hardware offload capable.

```
RP/0/RSP0/CPU0:router# show bfd session interface Te0/0/0/7/1.3001 detail
I/f: TenGigE0/0/0/7/1.3001, Location: 0/0/CPU0
Dest: 192.12.183.1
Src: 192.12.185.2
State: UP for 0d:11h:52m:17s, number of times UP: 1
Session type: PR/V4/SH
Received parameters:
Version: 1, desired tx interval: 15 ms, required rx interval: 15 ms
Required echo rx interval: 0 ms, multiplier: 4, diag: None
My discr: 2148535109, your discr: 2148073905, state UP, D/F/P/C/A: 0/0/0/1/0
Transmitted parameters:
Version: 1, desired tx interval: 3300 us, required rx interval: 3300 us
Required echo rx interval: 0 ms, multiplier: 4, diag: None
My discr: 2148073905, your discr: 2148535109, state UP, D/F/P/C/A: 0/1/0/1/0
Timer Values:
Local negotiated async tx interval: 15 ms
Remote negotiated async tx interval: 15 ms
Desired echo tx interval: 0 s, local negotiated echo tx interval: 0 ms
Echo detection time: 0 ms(0 ms*4), async detection time: 60 ms(15 ms*4)
Local Stats:
Intervals between async packets:
   Tx: Number of intervals=3, min=1 ms, max=3 ms, avg=2 ms
       Last packet transmitted 42737 s ago
   Rx: Number of intervals=3, min=1 ms, max=14 ms, avg=6 ms
       Last packet received 42737 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
  --------------------  ---------- ---------- ---------- ----------
  isis-13               3 ms       4          3300 us    4

H/W Offload Info:
H/W Offload capability : Y, Hosted NPU     : 0/0/CPU0/NPU3
Async Offloaded        : Y, Echo Offloaded : N
Async rx/tx            : 4/4

Platform Info:
NPU ID: 3
Async RTC ID        : 2          Echo RTC ID        : 0
Async Feature Mask  : 0x20       Echo Feature Mask  : 0x0
Async Session ID    : 0x51       Echo Session ID    : 0x0
Async Tx Key        : 0x512002  Echo Tx Key        : 0x0
Async Tx Stats addr : 0x1620b   Echo Tx Stats addr : 0x0
Async Rx Stats addr : 0x1620c   Echo Rx Stats addr : 0x0
```

BFD Over Bridge Group Virtual Interface: Example

The following examples show the configurations of the peer and uut nodes. You can see the BVI interface is under a VRF instead of default table:

```
interface BVI100
vrf cctv1 <<<<<<<<<
```

Below is the peer nodes example:

```
l2vpn
 bridge group bg
  bridge-domain bd
   interface Bundle-Ether1.100
   !
   routed interface BVI100
  !
 !
!
router vrrp
 interface BVI100
  bfd minimum-interval 15
  address-family ipv4
   vrrp 100
    address 192.168.1.254
    bfd fast-detect peer ipv4 192.168.1.2
   !
  !
 !
!
router ospf 100
vrf cctv1
  router-id 192.168.1.1
  area 0
   interface BVI100
   !
  !
 !
!
interface BVI100
 vrf cctv1
 ipv4 address 192.168.1.1 255.255.255.0
```

```
!
interface GigE0/1/0/10
 bundle id 1 mode active
 no shut
!
interface Bundle-Ether1
no shut
!
interface Bundle-Ether1.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric

!
bfd multipath include loc 0/1/cpu0

interface MgmtEth0/RSP1/CPU0/0
 ipv4 address 7.37.19.20 255.255.0.0
 no shutdown
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 7.37.0.1
```

Below is the uut node example:

```
l2vpn
 bridge group bg
  bridge-domain bd
   interface Bundle-Ether1.100
   !
   routed interface BVI100
  !
 !
!
router vrrp
 interface BVI100
  bfd minimum-interval 15
  address-family ipv4
   vrrp 100
    address 192.168.1.254
    bfd fast-detect peer ipv4 192.168.1.1
   !
  !
 !
!
router ospf 100
 vrf cctv1
  router-id 192.168.1.2
  area 0
   interface BVI100
   !
  !
 !
!
interface BVI100
 vrf cctv1
 ipv4 address 192.168.1.2 255.255.255.0
!


interface GigE0/1/0/0
 bundle id 1 mode active
```

```
 no shut
 !
 interface Bundle-Ether1
 no shut
!
interface Bundle-Ether1.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric

!
bfd multipath include location 0/1/CPU0
```

Configuring BFD IPv6 Multihop: Examples

Configuring BFD IPv6 Multihop for eBGP Neighbors: Example

This example shows how to configure BFD IPv6 Multihop for eBGP Neighbors:

```
bfd
 multipath include location 0//CPU0
!
router bgp 65001
 neighbor 21:1:1:1:1:1:1:2
  bfd fast-detect
  ebgp-multiphop 255
```

Configuring BFD IPv6 Multihop for iBGP Neighbors: Example

This example shows how configure BFD IPv6 Multihop for iBGP Neighbors:

```
bfd
 multipath include location 0/7/CPU0
!
router bgp 65001
 neighbor 21:1:1:1:1:1:1:2
  bfd fast-detect
```

BFD over MPLS TE LSPs: Examples

These examples explain how to configure BFD over MPLS TE LSPs.

BFD over MPLS TE Tunnel Head-end Configuration: Example

This example shows how to configure BFD over MPLS TE Tunnel at head-end.

```
bfd multipath include loc 0/1/CPU0
mpls oam
interface tunnel-te 1 bfd fast-detect
interface tunnel-te 1
 bfd minimum-interval
 bfd multiplier
 bfd bringup-timeout
 bfd lsp-ping interval 60
 bfd lsp-ping disable
```

```
bfd dampening initial-wait    (default 16000 ms)
bfd dampening maximum-wait    (default 600000 ms)
bfd dampening secondary-wait  (default 20000 ms)
logging events bfd-status
```

BFD over MPLS TE Tunnel Tail-end Configuration: Example

This example shows how to configure BFD over MPLS TE Tunnels at tail-end.

```
bfd multipath include loc 0/1/CPU0
mpls oam
mpls traffic-eng bfd lsp tail multiplier 3
mpls traffic-eng bfd lsp tail minimum-interval 100
```

Where to Go Next

BFD is supported over multiple platforms. For more detailed information about these commands, see the related chapters in the corresponding *Cisco IOS XR Routing Command Reference* and *Cisco IOS XR MPLS Command Reference* for your platform at:

http://www.cisco.com/en/US/products/ps5845/prod_command_reference_list.html

- *BGP Commands on Cisco IOS XR Software*
- *IS-IS Commands on Cisco IOS XR Software*
- *OSPF Commands on Cisco IOS XR Software*
- *Static Routing Commands on Cisco IOS XR Software*
- *MPLS Traffic Engineering Commands on Cisco IOS XR Software*

Additional References

The following sections provide references related to implementing BFD for Cisco IOS XR software.

Related Documents

Related Topic	Document Title
BFD commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Routing Command Reference for Cisco ASR 9000 Series Rou*
Configuring QoS packet classification	*Modular QoS Configuration Guide for Cisco ASR 9000 Serie*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

RFCs

RFCs	Title
rfc5880_bfd_base	*Bidirectional Forwarding Detection*, June 2010
rfc5881_bfd_ipv4_ipv6	*BFD for IPv4 and IPv6 (Single Hop)*, June 2010
rfc5883_bfd_multihop	*BFD for Multihop Paths*, June 2010

MIBs

MIBs	MIBs Link
All	To locate and download MIBs using Cisco IOS XR softw Cisco MIB Locator tool found at the following URL and platform under the Cisco Access Products menu.

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport