Implementing BFD

This module describes the configuration of bidirectional forwarding detection (BFD) on the Cisco XR 12000 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 3.3.1	BFD was introduced.
Release 3.3.2	• The **echo disable** command was added to enable users to disable echo mode on routers or interfaces where BFD is used in conjunction with Unicast Reverse Path Forwarding (uRPF). • A new BFD configuration mode was added, under which users can disable echo mode. The **bfd** command was added to allow users to enter the new BFD configuration mode.
Release 3.8.0	BFD MIB support was added on the Cisco XR 12000 Series Router.
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.

Release 4.0.1	Support for these BFD features 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 4.1.0	Support for BFD Per Member Links on Link Bundles 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 XR 12000 Series Router platform, each line card supporting BFD must be able to perform the following tasks:
    - Send echo packets every 50 ms (under normal conditions)
    - Send control packets every 250 ms (under normal conditions)
    - Send and receive more than 1000 User Datagram Protocol (UDP) pps on the 12000 Series platform. This sustains 50 sessions at a 50-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 with uRPF (IPv4)
    - Rack reload and online insertion and removal (OIR) when a BFD bundle interface has member links that span multiple racks
- BFD for IPv6 is not supported in software releases before Cisco IOS XR 4.1.
- 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.
- Echo latency detection and echo validation are not supported on bundle interfaces.

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. For more information, see the Enabling BFD on a Static Route.
- 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 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.

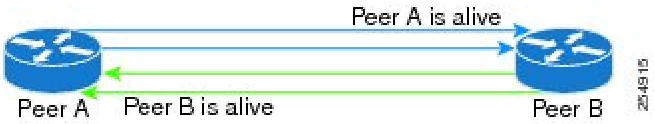
BFD echo mode is enabled by default for the following interfaces:

- 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

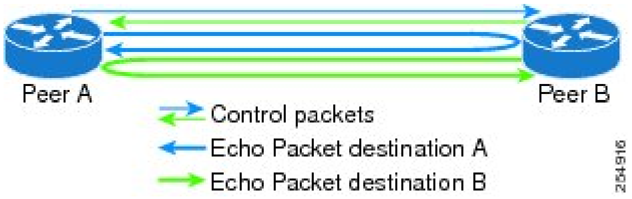


When BFD is running asynchronously with echo packets (Figure 36), the following occurs:

- 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.

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 Failure Detection Time (Echo Interval x 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 *Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco XR 12000 Series Router*.

BFD for IPv4

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

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
  - OSPF
  - MPLS Traffic Engineering (MPLS-TE)
  - Static routes (IPv4)
  - 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)
- ◦ Logical interfaces such as bundles, GRE, PWHE

> **Note** BFD is supported on the above interface types and not on logical interfaces unless specifically stated. For example, BFD cannot be configured on BVI and interflex.

- 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.

Enabling BFD on a Static Route

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

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. **commit**

DETAILED STEPS

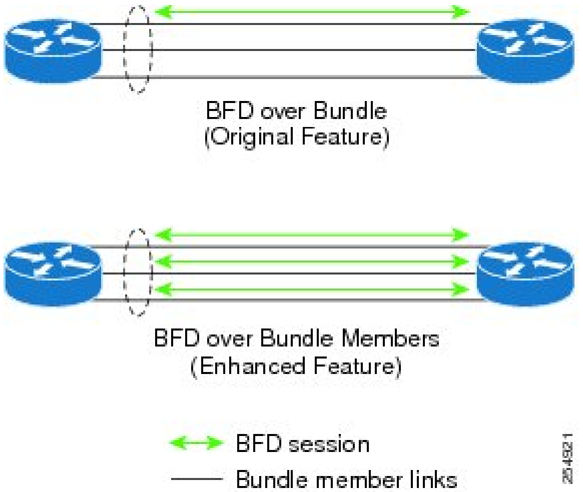
	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router static** **Example:** `RP/0/0/CPU0:router(config)# router static`	Enters static route configuration mode, allowing you to configure static routing.
Step 3	**address-family ipv4 unicast** *address nexthop* **bfd fast-detect** [**minimum-interval** *interval*] [**multiplier** *multiplier*]	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

	Command or Action	Purpose
	Example: `RP/0/0/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`	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.
Step 4	**vrf** *vrf-name* **Example:** `RP/0/0/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/0/0/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	**commit**	

BFD Over Member Links on Link Bundles

Beginning in Cisco IOS XR Release 4.1, the 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.

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.
    - The maximum rate for BFD sessions per linecard is 1334 pps.
- The maximum number of BFD sessions supported on any one card is 1250.
- 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.

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router bgp** *autonomous-system-number* **Example:** `RP/0/0/CPU0:router(config)# router bgp 120`	Enters BGP configuration mode, allowing you to configure the BGP routing process. 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/0/0/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/0/0/CPU0:router(config-bgp)# bfd multiplier 7`	Sets the BFD multiplier.

	Command or Action	Purpose
Step 5	**neighbor** *ip-address* **Example:** `RP/0/0/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/0/0/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/0/0/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	**commit**	

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, IS-IS, and MPLS-TE only.

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. **commit**
10. **show run router ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd multipath include location***node-id* **Example:** `RP/0/0/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/0/0/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.
Step 4	**bfd minimum-interval** *milliseconds* **Example:** `RP/0/0/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/0/0/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/0/0/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/0/0/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/0/0/CPU0:router(config-ospf-ar-if)#` **bfd fast-detect**	Enables BFD to detect failures in the path between adjacent forwarding engines.
Step 9	**commit**	

	Command or Action	Purpose
Step 10	**show run router ospf** **Example:** `RP/0/0/CPU0:router(config-ospf-ar-if)# show run router ospf`	Verify 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.

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router static** **Example:** `RP/0/0/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/0/0/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.

	Command or Action	Purpose
Step 4	**vrf** *vrf-name* **Example:** `RP/0/0/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/0/0/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	**commit**	

Configuring BFD on Bundle Member Links

Prerequisites for Configuring BFD on Bundle Menmber 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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**interface** [**Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/0/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/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** [**Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/0/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/0/0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect`	Enables IPv4 BFD sessions on bundle member links.
Step 4	**commit**	

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-POS**] *bundle-id*
3. **bundle minimum-active bandwidth** *kbps*
4. **bundle minimum-active links** *links*
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** [**Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/0/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/0/0/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/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** [**Bundle-Ether** | **Bundle-POS**] *bundle-id* **Example:** `RP/0/0/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/0/0/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.	

	Command or Action	Purpose
Step 4	**bfd address-family ipv4 multiplier** *multiplier* **Example:** `RP/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** [**Bundle-Ether** | **Bundle-POS**] *bundle-id*	Enters interface configuration mode for the specified bundle ID.

	Command or Action	Purpose
	Example: `RP/0/0/CPU0:router(config)# interface Bundle-Ether 1`	
Step 3	**bfd address-family ipv4 timers start** *seconds* **Example:** `RP/0/0/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/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**echo ipv4 source** *ip-address* **Example:** `RP/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**interface** type interface-path-id **Example:** `RP/0/0/CPU0:router(config-bfd)# interface gigabitEthernet 0/1/5/0`	Enters BFD interface configuration mode for a specific interface. In BFD interface configuration mode, you can specify an IPv4 address on an individual interface.
Step 4	**echo ipv4 source** *ip-address* **Example:** `RP/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**echo latency detect** [**percentage** *percent-value* [**count** *packet-count*] **Example:** `RP/0/0/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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
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	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**echo disable** **Example:** `RP/0/0/CPU0:router(config-bfd)# echo disable`	Disables echo mode on the router.
Step 4	**commit**	

Disabling Echo Mode on an Individual Interface

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

SUMMARY STEPS

1. **configure**
2. **bfd**
3. **interface** *type interface-path-id*
4. **echo disable**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/0/CPU0:router(config-bfd)# interface gigabitEthernet 0/1/5/0`	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.
Step 4	**echo disable** **Example:** `RP/0/0/CPU0:router(config-bfd-if)# echo disable`	Disables echo mode on the specified individual interface or bundle.
Step 5	**commit**	

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. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**bfd** **Example:** `RP/0/0/CPU0:router(config)# bfd`	Enters BFD configuration mode.
Step 3	**dampening** [**bundle-member**] {**initial-wait** | **maximum-wait** | **secondary-wait**} *milliseconds* **Example:** `RP/0/0/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	**commit**	

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** | **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/0/0/CPU0:router#show bfd counters all packet location 0/3/cpu0`	Displays the BFD counters for IPv4 packets or all packets.

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

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

BFD Over OSPF: Example

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

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#router ospf 0
RP/0/0/CPU0:router(config-ospf)#area 0
RP/0/0/CPU0:router(config-ospf-ar)#interface gigabitEthernet 0/3/0/1
RP/0/0/CPU0:router(config-ospf-ar-if)#bfd fast-detect
RP/0/0/CPU0:router(config-ospf-ar-if)#commit
RP/0/0/CPU0:router(config-ospf-ar-if)#end

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

router ospf 0
area 0
```

```
interface GigabitEthernet0/3/0/1
bfd fast-detect
```

BFD Over Static Routes: Example

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

BFD on Bundle Member Links: Examples

The following example shows how to configure BFD on member links of a POS bundle interface:

```
interface Bundle-POS 1
 bfd address-family ipv4 timers start 60
 bfd address-family ipv4 timers nbr-unconfig 60
 bfd address-family ipv4 multiplier 4
 bfd address-family ipv4 destination 192.168.77.2
 bfd address-family ipv4 fast-detect
 bfd address-family ipv4 minimum-interval 120
 ipv4 address 192.168.77.1 255.255.255.252

bundle minimum-active links 2
 bundle minimum-active bandwidth 150000
!
interface Loopback1
 ipv4 address 10.1.1.2 255.255.255.255
!
!
interface Pos0/2/0/0
 bundle id 1 mode active
!
interface Pos0/1/0/0
 bundle id 1 mode active
!
interface Pos0/1/0/1
 bundle id 1 mode active

interface Pos0/1/0/2
 bundle id 1 mode active

interface Pos0/1/0/3
 bundle id 1 mode active
router static
 address-family ipv4 unicast
  ! IPv4 Bundle-Pos1 session, shares ownership with bundle manager
  192.168.177.1/32 192.168.77.2 bfd fast-detect



router ospf foo
 bfd fast-detect
 redistribute connected
 area 0
  interface Bundle-Pos1
   ! IPv4 Bundle-Pos1 session, shares ownership with bundle manager
  !
router ospfv3 bar
```

```
 router-id 10.1.1.2
 bfd fast-detect
 redistribute connected
 area 0
  interface Bundle-Pos1
```

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

 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
!
```

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

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

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#bfd
RP/0/0/CPU0:router(config-bfd)#interface gigabitethernet 0/1/0/0
RP/0/0/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/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#bfd
RP/0/0/CPU0:router(config-bfd)#dampening bundle-member initial-wait 8000
RP/0/0/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/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#bfd
RP/0/0/CPU0:router(config-bfd)#dampening initial-wait 30000
RP/0/0/CPU0:router(config-bfd)#dampening maximum-wait 35000
```

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

Router 2 (Cisco IOS XR Software)

```
RP/0/0/CPU0:router#configure
RP/0/0/CPU0:router(config)#router static
RP/0/0/CPU0:router(config-static)#address-family ipv4 unicast
RP/0/0/CPU0:router(config-static-afi)#10.10.10.10/32 192.0.2.2 bfd fast-detect
RP/0/0/CPU0:router(config-static-afi)#exit
RP/0/0/CPU0:router(config-static)#exit
RP/0/0/CPU0:router(config)#interface GigabitEthernet0/0/0/0
RP/0/0/CPU0:router(config-if)#ipv4 address 192.0.2.1 255.255.255.0
```

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	*Cisco IOS XR Routing Command Reference for the Cisco XR 12000 Series Router*
Configuring QoS packet classification	*Cisco IOS XR Modular Quality of Service Configuration Guide for the Cisco XR 12000 Series Router*

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
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

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