Implementing BFD

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

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.
- To enable BFD for a neighbor, the neighbor router must support BFD.
- 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 is configured.

  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.
- Echo latency detection and echo validation are not supported on bundle interfaces.
- Echo mode is not supported on BFD over bundle interfaces and on bundle VLANS, which is BFD over logical bundle (BLB).

Information About BFD

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.

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.

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 .

BFD over Bundles 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 IETF standard per bundle, without necessitating reloads or process restarts across various systems.
.

- IETF mode uses IANA assigned MAC.
- IETF BFD over Bundle sessions use destination UDP port: 6784.

Restrictions

These limitations apply for the BFD over Bundle Mode feature:

- 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 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, 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 (single-hop template). 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 Support for IPv4

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

Restrictions

- This feature is not supported over MPLS LDP interface, VRRP interface, BVI interface and IRB interface.

Confiugration Example

```
/* Configure BFD over Bundle(BOB) for hardware offload. */
Router# config
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv4 multiplier 3
Router (config-if)# bfd address-family ipv4 destination 10.20.20.1
Router (config-if)# bfd address-family ipv4 fast-detect
Router(config-if)# bfd address-family ipv4 minimum-interval 1200
Router(config-if)# ipv4 address 10.20.20.2/30

/* Configure BFD with a static route. */
Router(config)# router static
Router(config-static)# address-family ipv4 unicast 10.1.1.0/24 10.6.0.2 bfd fast-detect
minimum-interval 1200 multiplier 4

/* Configure BFD with IS-IS. */
Router(config)# router isis 65444
Router(config-isis)# address-family ipv4 unicast
Router(config-isis)# exit
Router(config-isis)# interface HundredGige 0/3/0/1
Router(config-isis-if)#  bfd minimum-interval 1200
Router(config-isis-if)# bfd multiplier 7
Router(config-isis-if)# bfd fast-detect ipv4
Router(config-isis-if)# address-family ipv4 unicast

/* Configure BFDv4 with OSPF. */
Router(config)# router ospf main
Router(config-ospfv3)# area 0
Router(config-ospfv3-ar)#  interface HundredGige 0/0/0/1
Router(config-ospfv3-ar-if)# bfd multiplier 7
Router(config-ospfv3-ar-if)# bfd fast-detect
Router(config-ospfv3-ar-if)# bfd minimum-interval 1200

/* Configuring BFD over BGP. */
Router(config)# router bgp 120
Router(config-bgp)# neighbor 10.6.6.1
```

```
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 7
Router(config-bgp-nbr)# bfd minimum-interval 1200
```

Verification

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

```
Router# show bfd ipv4 session
Interface           Dest Addr           Local det time(int*mult)       State
                                      Echo             Async   H/W       NPU
------------------- --------------- ---------------- ---------------- ----------
Hu0/0/0/22.93       10.20.20.1       0s(0s*0)         12ms(4ms*3)      UP
                                                            Yes   0/0/CPU0
```

BFD Hardware Offload Support for IPv6

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

Restrictions

- This feature is not supported over MPLS LDP interface, VRRP interface, BVI interface and IRB interface.
- BFD Dampening is not supported for BFD over IPv6.
- BFD over Bundle (BOB) over IPv6 is not supported with dynamically configured link-local address. It must be statically configured.

Configuration Example

```
/* Configure BFD over Bundle(BOB) for hardware offload. */
Router# config
Router(config)# interface Bundle-Ether 1
Router(config-if)# bfd mode ietf
Router(config-if)# bfd address-family ipv6 multiplier 3
Router (config-if)# bfd address-family ipv6 destination 10.20:20::1
Router (config-if)# bfd address-family ipv6 fast-detect
Router(config-if)# bfd address-family ipv6 minimum-interval 1200
Router(config-if)# ipv6 address 10:20:20::2/64

/* Configure BFD with a static route. */
Router(config)# router static
Router(config-static)# address-family ipv6 unicast 1011:17e4::1/128 ab11:15d2::2 bfd
fast-detect minimum-interval 1200 multiplier 3

/* Configure BFD with IS-IS. */
Router(config)# router isis 65444
Router(config-isis)# address-family ipv6 unicast
Router(config-isis)# exit
Router(config-isis)# interface HundredGige 0/3/0/1
Router(config-isis-if)#  bfd minimum-interval 1200
Router(config-isis-if)# bfd multiplier 7
Router(config-isis-if)# bfd fast-detect ipv6
Router(config-isis-if)# address-family ipv6 unicast

/* Configure BFDv6 with OSPFv3. */
Router(config)# router ospfv3 main
```

```
Router(config-ospfv3)# area 0
Router(config-ospfv3-ar)#  interface HundredGige 0/0/0/1
Router(config-ospfv3-ar-if)# bfd multiplier 7
Router(config-ospfv3-ar-if)# bfd fast-detect
Router(config-ospfv3-ar-if)# bfd minimum-interval 1200

/* Configuring BFD over BGP. */
Router(config)# router bgp 120
Router(config-bgp)# neighbor 2001:DB8:1::1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 7
Router(config-bgp-nbr)# bfd minimum-interval 1200
```

Verification

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

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

IPv4 Multihop BFD

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

The IPv4 Multihop BFD 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.

Configure IPv4 Multihop BFD

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

```
Router# configure
Router(config)# bfd
Router(config)# multihop ttl-drop-threshold 225
Router(config)# multipath include location 0/7/CPU0
Router(config)# router bgp 100
Router(config-bgp)# neighbor 209.165.200.225
Router(config-bgp-nbr)# remote-as 2000
Router(config-bgp-nbr)# update-source loopback 1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 3
Router(config-bgp-nbr)# bfd minimum-interval 1200
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
```

Running Configuration

```
bfd
 multihop ttl-drop-threshold 225
 multipath include location 0/7/CPU0
router bgp 100
 neighbor 209.165.200.225
  remote-as 2000
  update-source loopback 1
  bfd fast-detect
  bfd multiplier 3
  bfd minimum-interval 1200
      route-policy PASS-ALL in
    route-policy PASS-ALL out
  !
!
```

Verification

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

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

Configure BFD

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

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

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.

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **router bgp** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config)# router bgp 120
```

Enters BGP configuration mode, allowing you to configure the BGP routing process.

Step 3 **neighbor** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-bgp)# neighbor 172.168.40.24
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

This example configures the IP address 172.168.40.24 as a BGP peer.

Step 4 **remote-as** *autonomous-system-number*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
```

Creates a neighbor and assigns it a remote autonomous system.

This example configures the remote autonomous system to be 2002.

Step 5 **bfd fast-detect**

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd fast-detect
```

Enables BFD between the local networking devices and the neighbor whose IP address you configured to be a BGP peer in Step 3.

In the example in Step 3, the IP address 172.168.40.24 was set up as the BGP peer. In this example, BFD is enabled between the local networking devices and the neighbor 172.168.40.24.

Step 6 **bfd minimum-interval** *milliseconds*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd minimum-interval 1200
```

Sets the BFD minimum interval. Range is 3-1200 milliseconds.

Step 7 **bfd multiplier** *multiplier*

Example:

```
RP/0/RP0/CPU0:router(config-bgp-nbr)# bfd multiplier 7
```

Sets the BFD multiplier. This is optional, the minimum is 3 and by default the multiplier will be 3 for all protocols

Step 8 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **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

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

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

```
Router# configure

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

/* Set the BFD minimum interval. The range is from 3 to 1200 milliseconds. */
Router(config-ospf)# bfd minimum-interval 1200

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

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

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

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

Running Configuration

```
configure
  router ospf 0
  bfd minimum-interval 1200
  bfd multiplier 7
   area 0
    interface HundredGige 0/3/0/1
     bfd fast-detect
```

Verification

Verify that BFD is enabled on the appropriate interface.

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

 router ospf 0
 bfd minimum-interval 1200
 bfd multiplier 7
 area 0
 interface HundredGige 0/3/0/1
 bfd fast-detect
```

Enabling BFD on a Static Route

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

```
Router(config)# configure

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

/*Enter address family configuration mode. */
Router(config-static)# address-family ipv4 unicast 192.168.2.2/32

/*Specify an unicast destination address and next-hop IPv4 address.
Enable BFD fast-detection on the specified IPv4 unicast destination address */
Router(config-static)# 192.168.2.2 192.168.6.2 bfd fast-detect minimum-interval 1200
multiplier 3
```

Running Configuration

```
router static
 address-family ipv4 unicast
  192.168.2.2 192.168.6.2 bfd fast-detect minimum-interval 1200 multiplier 3
 !
!
```

Specifying the BFD Destination Address on a Bundle

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

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1
```

Enters interface configuration mode for the specified bundle ID.

Step 3 **bfd address-family ipv4 destination** *ip-address*

Example:

```
RP/0/RP0/CPU0:router(config-if)# bfd address-family ipv4 destination 10.20.20.1
```

Specifies the primary IPv4 address assigned to the bundle interface on a connected remote system, where *ip-address* is the 32-bit IP address in dotted-decimal format (A.B.C.D).

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

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Enabling BFD Sessions on Bundle Members

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

SUMMARY STEPS

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

DETAILED STEPS

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
Router(config)# interface Bundle-Ether 1
```

Enters interface configuration mode for the specified bundle ID.

Step 3 **bfd address-family ipv4 fast-detect**

Example:

```
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:

Step 1 **configure**

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1
```

Enters interface configuration mode for the specified bundle ID.

Step 3 **bundle minimum-active bandwidth** *kbps*

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```

Sets the minimum amount of bandwidth required before a bundle can be brought up or remain up. The range is from 1 through a number that varies depending on the platform and the bundle type.

Step 4 **bundle minimum-active links** *links*

Example:

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2
```

Sets the number of active links required before a bundle can be brought up or remain up. The range is from 1 to 32.

Note When BFD is started on a bundle that is already active, the BFD state of the bundle is declared when the BFD state of all the existing active members is known.

Step 5 **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 failure detection times are determined by a combination of the interval and multiplier values in these commands.

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

Step 1 **configure**

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1
```

Enters interface configuration mode for the specified bundle ID.

Step 3 **bfd address-family ipv4 minimum-interval** *milliseconds*

Example:

```
RP/0/RP0/CPU0:router(config-if)#bfd address-family ipv4 minimum-interval 1200
```

Note Specifies the minimum interval, in milliseconds, for asynchronous mode control packets on IPv4 BFD sessions on bundle member links. The range is from 3 to 1200.

Step 4 **bfd address-family ipv4 multiplier** *multiplier*

Example:

```
RP/0/RP0/CPU0:router(config-if)#bfd address-family ipv4 multiplier 30
```

Specifies a number that is used as a multiplier with the minimum interval to determine BFD control packet failure detection times and transmission intervals for IPv4 BFD sessions on bundle member links.

Note Although the command allows you to configure a minimum of 2, the supported minimum is 3.

Step 5 **commit**

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

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

Step 1 configure

Step 2 **interface Bundle-Ether** *bundle-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 1
```

Enters interface configuration mode for the specified bundle ID.

Step 3 **bfd mode ietf**

Example:

```
RP/0/RP0/CPU0:router(config-if)# bfd mode ietf
```

Enables IETF mode for BFD over bundle for the specified bundle.

Step 4 **bfd address-family ipv4 fast-detect**

Example:

```
RP/0/RP0/CPU0:router(config-if)# bfd address-family ipv4 fast-detect
```

Enables IPv4 BFD sessions on the specified bundle.

Step 5 commit

Step 6 **show bundle bundle-ether** *bundle-id*

Displays the selected bundle mode.

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

BFD echo mode is enabled by default for IPv4 on other physical interfaces whose minimum interval is less than three seconds.

If you have configured a BFD minimum interval greater than three seconds on a physical interface using the **bfd minimum-interval** command, then you will need to change the interval to be less than three 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.

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:

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **bfd**

Example:

```
Router(config)# bfd
```

Enters BFD configuration mode.

Step 3 **echo ipv4 source** *ip-address*

Example:

```
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

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

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **bfd**

Example:

```
Router(config)# bfd
```

Enters BFD configuration mode.

Step 3 **interface** type interface-path-id

Example:

```
Router(config-bfd)# interface HundredGige 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:

```
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.

Disabling Echo Mode

BFD does not support asynchronous operation in echo mode in certain environments.

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

Step 1 **configure**

Step 2 **bfd**

Example:

```
Router(config)# bfd
```

Enters BFD configuration mode.

Step 3 **echo disable**

Example:

```
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.

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **bfd**

Example:

```
Router(config)# bfd
```

Enters BFD configuration mode.

Step 3 **interface** *type interface-path-id*

Example:

```
Router(config-bfd)# interface HundredGige 0/1/5/0
```

Enters BFD interface configuration mode for a specific interface. In BFD interface configuration mode, you can disable echo mode on an individual interface.

Step 4 **echo disable**

Example:

```
Router(config-bfd-if)# echo disable
```

Disables echo mode on the specified individual interface.

Step 5 **commit**

Minimizing BFD Session Flapping Using BFD Dampening

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

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters mode.

Step 2 **bfd**

Example:

```
Router(config)# bfd
```

Enters BFD configuration mode.

Step 3 **dampening** [**bundle-member**] {**initial-wait** | **maximum-wait** | **secondary-wait**} *milliseconds*

Example:

```
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.

Clear and Display BFD Counters

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

```
Router# show bfd counters all packet location 0/3/cpu0
Router# clear bfd counters all packet location 0/3/cpu0
Router# show bfd counters all packet location 0/3/cpu0
```

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.

BFD Over BGP: Example

The following example shows how to configure BFD between autonomous system 1200 and neighbor 192.168.70.24:

```
Router#configure
Router#(config)#router bgp 1200
Router#(config-bgp)#bfd multiplier 2
Router#(config-bgp)#bfd minimum-interval 1200
Router#(config-bgp)#neighbor 192.168.70.24
Router#(config-bgp-nbr)#remote-as 2
Router#(config-bgp-nbr)#bfd fast-detect
Router#(config-bgp-nbr)#commit
Router#(config-bgp-nbr)#end
Router##show run router bgp
```

BFD Over OSPF: Example

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

```
Router#configure
Router#(config)#router ospf 0
Router#(config-ospf)#area 0
Router#(config-ospf-ar)#interface HundredGige 0/3/0/1
Router#(config-ospf-ar-if)#bfd fast-detect
Router#(config-ospf-ar-if)#commit
```

```
Router#(config-ospf-ar-if)#end

Router#show run router ospf

router ospf 0
area 0
interface HundredGige 0/3/0/1
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.

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

BFD Echo Mode Disable: Examples

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

```
Router#configure
Router#(config)#bfd
Router#r(config-bfd)#echo disable
```

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

```
Router##configure
Router#(config)#bfd
Router#(config-bfd)#interface HundredGige 0/1/0/0
Router#(config-bfd-if)#echo disable
```

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:

```
Router#configure
Router(config)#bfd
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 HundredGige Ethernet interface:

```
Router#configure
Router(config)#bfd
Router(config-bfd)#interface HundredGige 0/1/0/0
Router(config-bfd-if)#echo ipv4 source 10.10.10.1
```

BFD Dampening: Examples

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

```
Router#configure
Router(config)#bfd
Router(config-bfd)#dampening bundle-member initial-wait 8000
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:

```
Router#configure
Router(config)#bfd
Router(config-bfd)#dampening initial-wait 30000
Router(config-bfd)#dampening maximum-wait 35000
```

BFD Hardware Offload for RSVP Tail-End

Table 1: Feature History Table

Feature Name	Release	Description
BFD Hardware Offload for Resource Reservation Protocol Tail-End	Release 7.9.1	You can use Bidirectional Forwarding Detection (BFD) to detect Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) data plane failures. This feature improves the scale and reduces the overall network convergence time by sending rapid failure detection packets to the routing protocols for recalculating the routing table. BFD process interacts with the Tail-End and LSPV processes to support BFD over Tail-End LSP feature. MPLS TE automatically establishes and maintains the LSPs across the MPLS network by using the Resource Reservation Protocol (RSVP).

You can use Bidirectional Forwarding Detection (BFD) to detect Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) data plane failures. An LSP ping request is used for detecting MPLS data plane failures and also for verifying the data plane against the control plane. BFD cannot be used for verifying the data plane against the control plane. However, the control plane processing required for BFD control packets is relatively smaller than the processing required for LSP ping messages. Hence, BFD can be deployed for faster detection of data plane failure (for example, traffic black-holing) for a large number of LSPs.

This feature improves the scale and reduces the overall network convergence time by sending rapid failure detection packets to the routing protocols for recalculating the routing table.

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 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 Tail-End 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 process interacts with the Tail-End and LSPV processes to support BFD over TE LSP feature. MPLS Tail-End automatically establishes and maintains the LSPs across the MPLS network by using the Resource Reservation Protocol (RSVP).

To know how MPLS works with RSVP Tail-End, refer to the *MPLS Configuration guide for Cisco 8000 Routers*.

BFD over MPLS Tail-End LSPs support:

- BFD async mode (BFD echo mode is not supported)
- IPv4 only, since MPLS core is IPv4
- BFD packets that 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 (3 ms x 3 = 9 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
- BFD packets from tail-end to head-end will be Label Switched if MPLS LDP is available in Core with label path from tail-end to head-end.

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.

1. Enter global configuration mode.

   ```
   Router#config
   ```

2. Configure MPLS OAM.

   ```
   Router(config)# mpls oam
   ```

3. Configure MPLS Traffic Engineering (MPLS TE) tunnel interface and enter into MPLS TE tunnel interface configuration mode.

   ```
   Router(config)#interface tunnel-te 65535
   ```

4. Enable BFD fast detection.

   ```
   Router(config-if)#bfd fast-detect
   ```

5. Configure hello interval in milliseconds.

   ```
   Router(config-if)#bfd minimum-interval 500
   ```

Note Hello interval range is 3 to 1000 milliseconds. Default hello interval is 100 milliseconds.

6. Configure BFD multiplier detection.

   ```
   Router(config-if)#bfd multiplier 5
   ```

Note BFD multiplier range is 3 to 10. Default BFD multiplier is 3.

7. Commit the changes.

   ```
   Router(config-if)#commit
   ```

Configuring BFD Bring up Timeout

Perform these steps to configure BFD bring up timeout interval.

1. Enter global configuration mode.

   ```
   Router#config
   ```

2. Configure MPLS Traffic Engineering (MPLS TE) tunnel interface and enter into MPLS TE tunnel interface configuration mode.

```
Router(config)#interface tunnel-te 65535
```

3. Enable the time interval (in seconds) to wait for the BFD session to come up.

```
Router(config-if)#bfd bringup-timeout 2400
```

Note The timeout range is 6 to 3600 seconds. Default bring up timeout interval is 60 seconds.

4. Commit the changes.

```
Router(config-if)#commit
```

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.

1. Enter global configuration mode.

```
Router# configure
```

2. Configure MPLS Traffic Engineering (MPLS TE) tunnel interface and enter into MPLS TE tunnel interface configuration mode.

```
Router(config)#interface tunnel-te 65535
```

3. Configure the initial delay interval before bringing up the tunnel.

```
Router(config-if)#bfd dampening initial-wait 360000
```

Note The initial-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default initial-wait interval is 16000 milliseconds.

4. Configure the maximum delay interval before bringing up the tunnel.

```
Router(config-if)#bfd dampening maximum-wait 700000
```

Note The maximum-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default initial-wait interval is 600000 milliseconds.

5. Configure the secondary delay interval before bringing up the tunnel.

```
Router(config-if)#bfd dampening secondary-wait 30000
```

Note The secondary-wait bring up delay time interval range is 1 to 518400000 milliseconds. Default secondary-wait interval is 20000 milliseconds.

6. Commit the changes.

```
Router(config-if)#commit
```

Configuring Periodic LSP Ping Requests

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

1. Enter global configuration mode.

```
Router# configure
```

2. Configure MPLS Traffic Engineering (MPLS TE) tunnel interface and enter into MPLS TE tunnel interface configuration mode.

```
Router(config)#interface tunnel-te 65535
```

3. Set periodic interval for LSP ping requests in seconds.

```
Router(config-if)#bfd lsp-ping interval 300
```

Note The interval range is 60 to 3600 seconds. Default interval is 120 seconds.

4. Commit the changes.

```
Router(config-if)#commit
```

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.

1. Enter global configuration mode.

```
Router# configure
```

2. Configure MPLS OAM.

```
Router(config)# mpls oam
```

3. Configure hello interval in milliseconds.

```
Router(config)#mpls traffic-eng bfd lsp tail minimum-interval 500
```

Note Hello interval range is 3 to 1000 milliseconds. Default hello interval is 100 milliseconds.

4. Configure BFD multiplier detection.

```
Router(config)#mpls traffic-eng bfd lsp tail multiplier 5
```

Note BFD multiplier detect range is 3 to 10. Default BFD multiplier is 3.

5. Commit the changes.

```
Router(config-if)#commit
```

Configuring BFD over LSP Sessions on Line Cards

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

Note For fixed box and centralized platforms, there are no line cards. Datapath is running on routing processors (RPs) which is where BFD sessions need to be created. On fixed box, the configuration must include the RPs instead of LCs. On centralized platforms, you cannot use RP in the config even though BFD sessions will be running on the RPs. You must include one of the MPAs instead of LC in the configuration.

1. Enter global configuration mode.

   ```
   Router# configure
   ```

2. Enter BFD configuration mode.

   ```
   Router(config)# bfd
   ```

3. Configure BFD multiple path on specific line card.

   ```
   Router(config-bfd)# multipath include location 0/1/CPU0
   ```

4. Commit the changes.

   ```
   Router(config-if)#commit
   ```

BFD over MPLS TE Tunnel Tail-End Configuration

You can use the **mpls traffic-eng bfd lsp tail minimum-interval** command to configure the tail-end at a minimum interval of 3 milli seconds.

Configuration

```
Router#config
Router(config)#mpls traffic-eng bfd lsp tail minimum-interval 3
Router(config)#commit
```

Running Configuration

```
mpls traffic-eng bfd lsp tail minimum-interval 3
 !
```

Verification

Use the **show bfd session** command to verify the configuration on tail-end.

```
Router#show bfd session

Src Addr            Dest Addr       VRF Name         Type Specific Data
                                        Local det time(int*mult)      State
H/W                 NPU                 Echo             Async
------------------- --------------- ---------------- ---------------- ----------
1.1.1.1             2.2.2.2         default          TT32768 (LSP:2)
                                    n/a              1500ms(500ms*3)  UP
```

Use the **show bfd label session** to verify the configuration on head-end.

```
Router#show bfd label session

Interface           Label               Local det time(int*mult)      State
                                        Echo             Async
H/W                 NPU
------------------- --------------- ---------------- ---------------- -----------
tt1 (LSP:103)       24001           n/a              150ms(50ms*3)    UP
Yes                 0/1/CPU0
tt2 (LSP:102)       24002           n/a              150ms(50ms*3)    UP
Yes                 0/1/CPU0
tt3 (LSP:101)       24004           n/a              150ms(50ms*3)    UP
Yes                 0/1/CPU0
tt4 (LSP:103)       24005           n/a              150ms(50ms*3)    UP
Yes                 0/1/CPU0
tt5 (LSP:104)       24006           n/a              150ms(50ms*3)    UP
Yes                 0/1/CPU0
```

Configuration Examples for Configuring BFD

-

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.

```
Router# bfd multipath include loc 0/1/CPU0
mpls oam
interface tunnel-te 1
 bfd
  minimum-interval 500
  fast-detect
  multiplier 5
  bringup-timeout 60
  lsp-ping disable
  dampening initial-wait    (default 16000 ms)
  dampening maximum-wait    (default 600000 ms)
  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.

```
Router# 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 500
```

Bidirectional Forwarding Detection over VXLAN Tunnel

Table 2: Feature History

Feature Name	Release Information	Feature Description
Bidirectional Forwarding Detection over VXLAN Tunnel	Release 24.2.11	You can now monitor the health of VXLAN tunnel and detect failures in the tunnel rapidly which ensures faster rerouting of traffic, resulting in high availability of networks.

Bidirectional Forwarding Detection over VXLAN Tunnel feature enables you to detect Layer 2 data plane failures using BFD for the underlying Virtual Extensible LAN (VXLAN) tunnel forwarding plane.

Initially, a VXLAN network is established to encapsulate the network traffic, forming secure tunnels across the Layer 3 physical infrastructure. These tunnels allow traffic to remain invisible to the internal domain network, ensuring privacy and security for all users. When the traffic volume grows, the VXLAN's load balancing capabilities distribute the data packets evenly, thus maintaining network performance and reliability.

Within the VXLAN network, the sharing of route information is implemented using either BGP or static routes. With BGP, the routes are learned and shared dynamically, while static routes provide a consistent predefined path.

To detect link failures quickly, BFD is integrated over the VXLAN tunnel. When a failure occurs in the VXLAN tunnel, BFD rapidly detects the failure, and ensures quick failover and rerouting of traffic, thus minimizing potential downtime and service interruption.

Note BFD over VXLAN with BGP can support up to 2000 BFD sessions (IPv4 and IPv6 combined), whereas BFD over VXLAN on static routes can support up to 10,000 BFD sessions (up to10 percent IPv6 and 90-percent IPv4).

Restrictions

These restrictions apply for the BFD over VXLAN feature:

- This feature is applicable to multihop (MH) BFD sessions. You can achieve the maximum number of BFD sessions only using MH BFD sessions and with a distribution ratio of 90 percent IPv4 sessions to 10 percent IPv6 sessions. If the configuration includes other types of BFD sessions, or if the proportion of IPv6 sessions exceeds 10 percent, the maximum supported scale may not be attainable. The exact number of supported BFD sessions depends on the specific types of sessions and their distribution.
- This feature is supported only in BFD asynchronous mode. Echo mode is not supported.
- The parallel operation of BFD over VXLAN with static routes at a scale of up to 10,000 sessions alongside BFD over VXLAN with BGP is not supported.
- The stated scale of up to 10,000 BFD sessions is supported only on the Cisco 8100 and 8200 series routers.

- Software-based BFD sessions cannot support short minimum-interval values when operating at higher scale. So, it is recommended to configure a minimum-interval value greater than 300 milliseconds.
- The maximum rate in packets-per-second (PPS) for BFD sessions must be below 80% to ensure the stability of BFD sessions. Exceeding this threshold may result in unstable sessions. You can use the **show bfd summary** command to check the PPS %.

Configure BFD over VXLAN

The BFD over VXLAN feature can be configured under the following scenarios:

- BFD over VXLAN with BGP
- BFD over VXLAN with Static Routes

BFD over VXLAN with BGP

The configuration of BFD over VXLAN with BGP includes the following:

- VRF configuration
- Multipath BFD sessions configuration
- Interfaces configuration
- Interface NVE configuration
- BGP configuration
- BFD configuration

Configuration Example

Use the following sample configuration to configure BFD over VXLAN with BGP.

```
/* Configure VRF */
Router# configure
Router(config)# vrf vrf1
Router(config-vrf)# exit

/* Configure Multipath BFD sessions. */
Router# configure
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0
Router(config-bfd)# exit

/* Configure Interfaces */
Router(config)# interface Loopback 0
Router(config-if)# ipv4 address 10.10.10.10 255.0.0.0
Router(config-if)# exit
Router(config)# interface Loopback 1
Router(config-if)# vrf vrf1
Router(config-if)# ipv4 address 192.168.0.0 255.255.0.0
Router(config-if)# exit

/* Configure Interface NVE for Decapsulation */
Router(config)# interface nve1
Router(config-if)# member vni 2
Router(config-nve-vni)# vrf vrf1
```

```
Router(config-nve-vni)# host-reachability protocol static
Router(config-nve-vni)# exit
Router(config-if)# overlay-encapsulation vxlan
outer(config-nve-encap-vxlan)# peer-ip lookup disable
Router(config-nve-encap-vxlan)# exit
Router(config-if)# source-interface Loopback1
Router(config-if)# commit

/* Configure BGP */
Router(config)# router bgp 1
Router(config-bgp)# bgp router-id 10.10.10.10
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# redistribute connected
Router(config-bgp-af)# exit
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp)# exit

Router(config)# router bgp 1
Router(config-bgp)# vrf vrf1
Router(config-bgp-vrf)# rd auto
Router(config-bgp-vrf)# address-family ipv4 unicast
Router(config-bgp-vrf-af)# redistribute connected
Router(config-bgp-vrf-af)# redistribute static
Router(config-bgp-vrf-af)# exit

/* Configure BFD over BGP. */
Router(config)# router bgp 1
Router(config-bgp)# neighbor 10.6.6.1
Router(config-bgp-nbr)# bfd fast-detect
Router(config-bgp-nbr)# bfd multiplier 7
Router(config-bgp-nbr)# bfd minimum-interval 1200
Router(config-bgp-nbr)# remote-as 300
Router(config-bgp-nbr)# ebgp-multihop 255
Router(config-bgp-nbr)# update-source loopback 1
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
```

Running Configuration

```
vrf vrf1
!
bfd
 multipath include location 0/0/CPU0
!
interface Loopback0
 ipv4 address 10.10.10.10 255.0.0.0
!
interface Loopback1
 vrf vrf1
 ipv4 address 192.168.0.0 255.255.0.0
!
interface nve1
 member vni 2
  vrf vrf1
  host-reachability protocol static
 !
 overlay-encapsulation vxlan
  peer-ip lookup disable
 !
 source-interface Loopback1
```

```
!

router bgp 1
 bgp router-id 10.10.10.10
 address-family ipv4 unicast
  redistribute connected
 !
 address-family ipv4 unicast
!

router bgp 1
 vrf vrf1
  rd auto
  address-family ipv4 unicast
   redistribute connected
   redistribute static
  !

router bgp 1
 neighbor 10.6.6.1
  bfd fast-detect
  bfd multipler 7
  bfd minimum-interval 1200
  remote-as 300
  ebgp-multihop 255
  update-source Loopback 1
  address-family ipv4 unicast
   route-policy pass-all in
   route-policy pass-all out
!
!
!
```

BFD over VXLAN with Static Routes

The configuration of BFD over VXLAN with static routes includes the following:

- VRF configuration
- Multipath BFD sessions configuration
- Interfaces configuration
- Interface NVE configuration
- Static routing configuration

Configuration Example

Use the following sample configuration to configure BFD over VXLAN with static routes.

```
/* Configure VRF */
Router# configure
Router(config)# vrf vrf1
Router(config-vrf)# exit

/* Configure Multipath BFD sessions. */
Router# configure
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0
Router(config-bfd)# exit
```

```
/* Configure Interfaces */
Router(config)# interface Loopback 0
Router(config-if)# ipv4 address 10.10.10.10 255.0.0.0
Router(config-if)# ipv6 address 2001:DB8:1::1/32
Router(config-if)# exit
Router(config)# interface Loopback 1
Router(config-if)# vrf vrf1
Router(config-if)# ipv4 address 192.168.0.0 255.255.0.0
Router(config-if)# exit
Router(config)# interface TenGigE0/0/0/0/0
Router(config-if)# ipv4 address 10.12.13.10 255.0.0.0
Router(config-if)# ipv6 address 2001:DB8:13::11/16
Router(config-if)# exit

/* Configure Interface NVE for Decapsulation */
Router(config)# interface nve1
Router(config-if)# member vni 2
Router(config-nve-vni)# vrf vrf1
Router(config-nve-vni)# host-reachability protocol static
Router(config-nve-vni)# exit
Router(config-if)# overlay-encapsulation vxlan
Router(config-nve-encap-vxlan)# exit
Router(config-if)# source-interface Loopback1
Router(config-if)# commit

/* Configure Static Routing */
Router# configure
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 10.10.10.10/32 10.151.11.2
Router(config-static-afi)# exit
Router(config-static)# address-family ipv6 unicast
Router(config-static-afi)# 2001:DB8:1::12/32 2001:DB8:17::122
Router(config-static-afi)# exit
Router(config-static)# vrf VRF1
Router(config-static-vrf)# address-family ipv4 unicast
Router(config-static-vrf-afi)# 10.1.1.1/32 remote-next-hop 10.13.19.10 tunnel VXLAN index
1 nve 1 evni 1 src-mac aaa1.bbb1.ccc1 ->  IPv4 over IPv4
Router(config-static-vrf-afi)# 209.165.201.0/27 10.1.1.1 bfd fast-detect minimum-interval
1000 multihop 192.168.0.0
Router(config-static-vrf-afi)# 10.12.12.12/32 remote-next-hop 10.13.19.10 tunnel VXLAN index
 2 nve 1 evni 2 src-mac aaa1.bbb1.ccc2 ->  IPv4 over IPv4
Router(config-static-vrf-afi)# 209.165.202.129/27 10.12.12.12 bfd fast-detect minimum-interval
 1000 multihop 192.168.12.24
Router(config-static-vrf-afi)# exit
```

Running Configuration

```
vrf vrf1
!
bfd
 multipath include location 0/0/CPU0
!
interface Loopback0
 ipv4 address 10.10.10.10 255.0.0.0
 ipv6 address 2001:DB8:1::1/32
!
interface Loopback1
 vrf vrf1
 ipv4 address 192.168.0.0 255.255.0.0
!
interface TenGigE0/0/0/0/0
 ipv4 address 10.12.13.10 255.0.0.0
```

```
 ipv6 address 2001:DB8:13::11/16
!
interface nve1
 member vni 2
  host-reachability protocol static
 !
 overlay-encapsulation vxlan
 !
 source-interface Loopback1
!
router static
 address-family ipv4 unicast
  10.10.10.10/32 10.151.11.2
 !
 address-family ipv6 unicast
  2001:DB8:1::12/32 2001:DB8:17::122
 !
 vrf vrf1
  address-family ipv4 unicast
   10.1.1.1/32 remote-next-hop 10.13.19.10 tunnel VXLAN index 1 nve 1 evni 1 src-mac
0022.3344.5566
   209.165.201.0/27 10.1.1.1 bfd fast-detect minimum-interval 1000 multihop 192.168.0.0
   10.12.12.12/32 remote-next-hop 10.13.19.10 tunnel VXLAN index 2 nve 1 evni 2 src-mac
0022.3344.5567
  209.165.202.129/27 10.12.12.12 bfd fast-detect minimum-interval 1000 multihop 192.168.12.24

  !
 !
!
```

Verification

- Use the **show bfd all** command to view the details of BFD over IPv4 and IPv6 sessions:

```
Router# show bfd all
Tue Dec 12 10:07:03.395 UTC

IPv4:
-----
IPV4 Sessions Up: 7200, Down: 0, Unknown/Retry: 2000, Total: 9200

IPv6:
-----
IPV6 Sessions Up: 800, Down: 0, Unknown/Retry: 0, Total: 800

Label:
------
Label Sessions Up: 0, Down: 0, Unknown/Retry: 0, Total: 0
```

- Use the **show bfd ipv6 multihop session** command to view the details of BFD over IPv6 multihop sessions:

```
Router# show bfd ipv6 multihop session
Tue Mar 26 19:32:14.851 UTC
Src Addr                                    Dest Addr
VRF Name                                   Local det time(int*mult)      State
                                       Echo             Async
H/W                 NPU
------------------- --------------- ---------------- ---------------- ----------
2001:DB8:0000::1                           2001:DB8:FFF::1
vrf5001                                0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::2                           2001:DB8:FFF::2
vrf5002                                0s(0s*0)         3s(1s*3)         UP
```

```
Yes                 0/RP0/CPU0
2001:DB8:0000::3                        2001:DB8:FFF::3
vrf5003                           0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::4                        2001:DB8:FFF::4
vrf5004                           0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::5                        2001:DB8:FFF::5
vrf5005                           0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::6                        2001:DB8:FFF::6
vrf5006                           0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::7                        2001:DB8:FFF::7
vrf5007                           0s(0s*0)         3s(1s*3)         UP
Yes                 0/RP0/CPU0
2001:DB8:0000::8                        2001:DB8:FFF::8
vrf5008                           0s(0s*0)         3s(1s*3)         UP
```

- Use the **show bfd session** command to view the BFD session details. The following is a sample output from a BFD over VXLAN system configured with static routes.

```
Router# show bfd session
Tue Mar 26 19:32:00.554 UTC
Src Addr            Dest Addr       VRF Name                        H/W NPU
                                        Local det time(int*mult)       State
                                    Echo             Async
------------------- --------------- ---------------- ---------------- ----------
209.165.201.1       10.0.0.1   vrf5001                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.2       10.0.0.2   vrf5002                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.3       10.0.0.3   vrf5003                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.4       10.0.0.4   vrf5004                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.5       10.0.0.5   vrf5005                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.6       10.0.0.6   vrf5006                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
209.165.201.7       10.0.0.7   vrf5007                        Yes 0/RP0/CPU0
                                    n/a              3s(1s*3)         UP
```

Bidirectional Forwarding Detection on BVI

Table 3: Feature History

Feature Name	Release Information	Feature Description
Bidirectional Forwarding Detection on BVI	Release 7.10.1	Now you can extend the advantage of low-overhead and short-duration detection of path failures between routers to an Integrated Routing and Bridging (IRB) deployment scenario by configuring Bidirectional Forwarding Detection (BFD) on multipath single-hop sessions using Bridge-Group Virtual Interface (BVI). By configuring BFD on a multipath session, you can apply BFD over virtual interfaces or between interfaces that are multiple hops away.

Overview

To establish a VLAN that extends across a router, the router needs to actively forward frames between interfaces while maintaining the integrity of the VLAN header. If the router is configured for routing a Layer 3 (network layer) protocol, it terminates the VLAN and MAC layers at the interface on which a frame arrives. The MAC layer header information can be maintained if the router bridges the network layer protocol. However, even with standard bridging techniques, the VLAN header is still terminated.

The Role of IRB in VLAN Spanning

VLANs facilitate network segmentation by establishing distinct broadcast domains. Routers can bridge traffic between different VLANs, but this requires the activation of VLAN trunking to enable the passage of multiple VLANs over a single link. This phenomenon, often referred to as VLAN spanning, ensures seamless connectivity.

However, in complex VLAN setups that involve multiple routers, the accurate preservation of VLAN header markings becomes imperative for effective spanning. This task can be challenging since routers usually replace the VLAN header with a new one, irrespective of whether the traffic is routed or bridged. This replacement hinders VLAN spanning, which in turn negatively impacts trunking and the facilitation of traffic bridging between VLANs.

To address this issue, Integrated Routing and Bridging (IRB) enables simultaneous routing and bridging on a router interface, while preserving the essential VLAN header integrity across routers. This ensures effective VLAN spanning within complex networks, even when multiple routers are involved

Integrated Routing and Bridging

Using the Integrated Routing Bridging (IRB) feature, you can configure your router to simultaneously route and bridge a given network layer protocol on a single interface. This configuration ensures the preservation of the VLAN header on a frame while it traverses from one interface to another via the router. By utilizing

the Bridge Group Virtual Interface (BVI), IRB enables seamless routing between bridged and routed domains. The BVI acts as a virtual interface within the router, resembling a standard routed interface that does not support bridging. However, it serves as the representation of the corresponding bridge group for the routed interfaces in the router. Moreover, BVI interface number represents the corresponding bridge group number, establishing a vital link between the BVI and the bridge group.

To enable the BVI to operate as a routed interface, you need to configure it exclusively with Layer 3 attributes, such as network layer addresses. However, any interfaces designated for protocol bridging should not have any Layer 3 characteristics configured.

BFD over IRB

This feature uses BVI to implement BFD over IRB to establish a multipath single-hop session. Within a BFD multipath session, the application of BFD extends to virtual interfaces or interfaces that are multihop away. This feature adheres to the specifications outlined in *RFC 5883—Bidirectional Forwarding Detection (BFD) for Multihop Paths*. This integration leverages the inherent benefits of BFD, characterized by its low overhead and rapid path failure detection between routers, and extends these advantages to the context of IRB deployment. BFD over BVI feature is compatible with both IPv4 addresses and IPv6 global addresses. This feature exclusively supports asynchronous mode and does not provide support for echo mode.

Restrictions

To communicate and exchange routing information with other routers in the network, IS-IS and OSPFv3 clients rely on IPv6 link local addresses. However, the BFD over BVI feature does not support IPv6 link local addresses and hence it does not support IS-IS and OSPFv3 clients.

Configure BFD over BVI

Configuration Example

```
/* Configure the line cards to allow hosting of Multipath BFD sessions. */
Router# configure
Router(config)# bfd
Router(config-bfd)# multipath include location 0/0/CPU0
Router(config-bfd)# multipath include location 0/2/CPU0
/* Configure the Layer 2 domain that should include the corresponding l2transport and BVIs.
 */
Router(config)# interface Bundle-Ether2601
Router(config-if)exit
Router(config)# interface Bundle-Ether2601.3001
Router(config-if)# l2transport
Router(config-if)# encapsulation dot1q 3001
Router(config-if)# rewrite ingress tag pop 1 symmetric
Router(config-if)# exit
Router(config)# interface HundredGigE0/0/0/6
Router(config)# bundle id 2601 mode on
Router(config-if)exit

/* Configure a BVI interface and assign an IP address. You should configure the BVI interface
 with either IPv4 or IPv6 address or both. */
Router(config)# interface bvi3001
Router(config-if)# ipv4 address 192.168.1.1 255.255.255.0
Router(config-if)# exit
```

```
/* Configure BVI on the peer nodes */
Router(config-if)# l2vpn
Router(config-l2vpn)# bridge group bvi100
Router(config-l2vpn-bg)# bridge domain bfd-bvi
Router(config-l2vpn-bg-bd)# interface Bundle-Ether2601.3001
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# routed interface bvi bvi3001

/* Configure OSPF as the routing protocol */
Router(config)# router ospf bfd-bvi
Router(config-ospf)# router-id 192.168.1.1
Router(config-ospf)# area 0

/* Configure BFD on BVI */
Router(config-ospf-ar)# interface bvi3001
Router(config-ospf-ar-if)#
Router(config-ospf-ar-if)# bfd fast-detect
Router(config-ospf-ar-if)# bfd multiplier 3
```

Running Configuration

```
bfd
 multipath include location 0/0/CPU0
 multipath include location 0/2/CPU0
!

interface Bundle-Ether2601
!
interface Bundle-Ether2601.3001
 l2transport
 encapsulation dot1q 3001
 rewrite ingress tag pop 1 symmetric
!
interface HundredGigE0/0/0/6
 bundle id 2601 mode on
!

interface bvi3001
 ipv4 address 192.168.1.1 255.255.255.0
!
l2vpn
 bridge group bvi100
  bridge domain bfd-bvi
   interface Bundle-Ether2601.3001
   !
   routed interface bvi bvi3001
   !
  !
!
!
router ospf bfd-bvi
router-id 192.168.1.1
area 0
interface bvi3001
 bfd minimum-interval bvi3001
 bfd fast-detect
 !
!
!
```

Verification

Verify the status of L2VPN bridge domain.

```
Router# show l2vpn bridge-domain brief

Mon Apr 24 11:31:04.314 UTC
Legend: pp = Partially Programmed.
Bridge Group:Bridge-Domain Name  ID  State  Num ACs/up  Num PWs/up  Num PBBs/up Num VNIs/up
-------------------------------- --  -----  ----------- ---------   ----------- -----------
bvi100:bfd-bvi                   0   up  2/2          0/0         0/0          0/0
```

Verify the status of OSPF connectivity. The creation of a BFD session requires the neighbor to be in the FULL/BDR state. The FULL/BDR state indicates that the router is fully adjacent with the backup designated router.

```
Router# show ospf neighbor

Mon Apr 24 11:40:41.900 UTC

* Indicates MADJ interface
# Indicates Neighbor awaiting BFD session up

Neighbors for OSPF bfd-bvi

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1     1     FULL/BDR        00:00:33    192.168.1.1     BVI3001
    Neighbor is up for 1d22h

Total neighbor count: 1
```

Verify the status of the BFD session. The output indicates the state of the BFD session is UP.

```
show bfd session interface bvi3001

Mon Apr 24 11:42:50.582 UTC
Interface   Dest Addr   Local det time(int*mult)   State  Echo  Async   H/W     NPU
---------  ------------ --------  --------------   -----  ----  -----  -----    -----
BVI3001     192.168.1.1  0s(0s*0)  750ms(250ms*3)   UP    Yes          0/2/CPU0
```

BFD over Pseudowire Headend

Table 4: Feature History

Feature Name	Release Information	Feature Description
BFD over Pseudowire Headend	Release 24.3.1	**Introduced in this release on: Modular Systems (8800 [LC ASIC: P100]) (select variants only*)** You can now rapidly detect failures in pseudowires, minimizing downtime and ensuring service reliability. This feature continuously monitors the pseudowire end-to-end, providing quick responses to maintain the integrity of Layer 2 VPNs and Ethernet services. *This feature is supported on: • 88-LC1-12TH24FH-E

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

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
- BFD asynchronous mode over PWHE, and
- Pseudowire VC type 4 and type 5.

Limitations for BFD over PWHE

These limitations apply for the BFD over PWHE feature.

- Supported only on 88-LC1-12TH24FH-E line card.
- Supports only static routes for IPv4 and IPv6.
- Supports only single-hop BFD sessions for IPv4 and IPv6.

For PWHE to be operational, the BFD agent must be hosted on one of the line cards that is part of the PWHE generic interface list. So, you must configure BFD multipath on a line card that is part of the generic interfaces list.

Configure BFD over PWHE

To enable BFD over PWHE, you must configure the specific line card to host BFD multipath sessions.

Step 1 Configure the line card to host BFD multipath sessions.

Example:

```
Router# configure
Router(config)# bfd multipath include location 0/5/CPU0
```

Step 2 Configure BFD fast-detection capability on the specified IPv4 or IPv6 unicast destination address prefix and on the forwarding next-hop address.

Example:

```
Router# configure
Router(config)# router static
Router(config-static)# address-family ipv4 unicast
Router(config-static-afi)# 198.51.100.0/24 209.165.201.0 bfd fast-detect minimum-interval 800 multiplier
 3
Router(config-static-afi)# 198.51.100.0/24 209.165.202.0 bfd fast-detect minimum-interval 800 multiplier
 3
Router(config-static-afi)# exit
Router(config-static)# address-family ipv6 unicast
Router(config-static-afi)# 2001:DB8:C18:2:1::F/48 2001:DB8:D987:398:AE3:B39:333:783 bfd fast-detect
 minimum-interval 800 multiplier 3
Router(config-static-afi)# 2001:DB8:C18:2:1::F/48 2001:DB8:D987:398:AE3:B39:334:783 bfd fast-detect
 minimum-interval 800 multiplier 3
```

Step 3 Verify the configuration.

Example:

```
Router# show bfd all session agent detail location 0/2/CPU0
IPv4:
-----
I/f: PW-Ether1002, Location: 0/2/CPU0
Dest: 209.165.202.0
Src: 209.165.201.0
 State: UP for 0d:0h:8m:17s, number of times UP: 1
 Session type: SW/V4/SH/PH
Received parameters:
 Version: 1, desired tx interval: 50 ms, required rx interval: 50 ms
 Required echo rx interval: 1 us, multiplier: 3, diag: None
 My discr: 2210464558, your discr: 4259847, state UP, D/F/P/C/A: 0/0/1/1/0
Transmitted parameters:
 Version: 1, desired tx interval: 50 ms, required rx interval: 50 ms
 Required echo rx interval: 0 ms, multiplier: 3, diag: None
 My discr: 4259847, your discr: 2210464558, state UP, D/F/P/C/A: 0/1/0/1/0
Timer Values:
 Local negotiated async tx interval: 50 ms
 Remote negotiated async tx interval: 50 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: 150 ms(50 ms*3)
Label:
 Internal label: 24024/0x5dd8
Local Stats:
 Intervals between async packets:
   Tx: Number of intervals=100, min=46 ms, max=1804 ms, avg=342 ms
       Last packet transmitted 1250 ms ago
   Rx: Number of intervals=100, min=46 ms, max=1804 ms, avg=342 ms
       Last packet received 1250 ms 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
```

What to do next

For more information about Pseudowire Headend and its configurations, see *Pseudowire Headend* module in *L2VPN Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 24.1.x , 24.2.x, 24.3.x*.

RFCs

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

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

Limiting LSA Numbers in a OSPF Link-State Database

Table 5: Feature History Table

Feature Name	Release	Description
Limiting LSA numbers in a OSPF Link-State Database	Release 7.9.1	The nonself-generated link-state advertisements (LSAs) for a given Open Shortest Path First (OSPF) process is limited to 500000. This protection mechanism prevents routers from receiving many LSAs, preventing CPU failure and memory shortages, and is enabled by default from this release onwards. If you have over 500000 LSAs in your network, configure the **max-lsa** command with the expected LSA scale before upgrading to this release or later. This feature modifies the following commands: • **show ospf** to display the maximum number of redistributed prefixes. • **show ospf database database-summary detail** to display the number of LSA counts per router. • **show ospf database database-summary adv-router** *router ID* to display the router information and the LSAs received from a particular router.

The OSPF Link-State Database Overload Protection feature allows you to protect the OSPF routing process by limiting the number of nonself-generated link-state advertisements (LSAs) for a given Open Shortest Path First (OSPF) process. When other routers in the network have been misconfigured, they may generate a high volume of LSAs. This mechanism prevents routers from receiving a large number of LSAs, thereby preventing CPU failure and memory shortages. With this feature, the router keeps a count of the number of nonself-generated LSAs it has received.

Note The max-lsa limit was not enabled by default before Release 7.9.1. Starting from Release 7.9.1, this command is enabled by default and the default limit of the nonself-generated LSA is set at 500000. If you have more than 500000 LSAs in a network, you must configure the **max-lsa** command with the expected LSA scale before upgrading to Release 7.9.1 or above.

Restriction

This feature is supported only on OSPFv2 and not on OSPFv3.

System output messages

The range of nonself-generated LSA allowed is 1-4294967294. The threshold percentage to log warning is 75%. The system log message is generated every 5% above the default or configured threshold value until 100% is reached.

When the number of LSAs reaches or exceeds the threshold limit, the router displays the following logs:

When number of LSAs exceed threshold value

```
%ROUTING-OSPF-4-MAX_LSA_THR : Reached threshold (60% [configured: 60%])
for maximum number of non self-generated LSAs in vrf "default" - LSA (max:
1000 cur: 600)
```

When number of LSAs exceed maximum limit

```
%ROUTING-OSPF-1-MAX_LSA : Maximum number of non self-generated LSAs
exceeded in vrf "default" - LSA (max: 1000, cur: 1001)
```

When OSPF instance ignores all adjacencies for ignore-time period if the number of LSAs exceed the limit

```
%ROUTING-OSPF-2-MAX_LSA_IGNORE_ENTER : Max LSA exceeded in vrf "default".
Adjacencies will be kept down for 5 minutes
```

When OSPF instance tries to recover the adjacencies after ignore-time period

```
%ROUTING-OSPF-6-MAX_LSA_IGNORE_EXIT : Max-lsa ignore timed out in vrf
"default". Adjacencies will be brought up by accepting and sending hellos
```

When the ignore count is exceeded on the OSPF instance

```
%ROUTING-OSPF-1-MAX_LSA_PERM_IGNORE : Max-lsa ignore count exceeded in
vrf "default" - Staying in ignore state. Restart or Clear OSPF process
to recover
```

When number of LSAs exceed threshold or limit the top contributing routers information will be displayed

```
%ROUTING-OSPF-2-MAX_LSA_RTR_INFO : Top 1 LSA contributor in vrf "default".
RTR:192.168.0.4   Total:498   Type3:0   Type5:492   Type7:0   Type10:0
Type11:6   Others:0
```

The following commands displays the LSA counts:

- **show ospf database database-summary detail** command displays the number of LSA counts per router sorted by total LSA count.

```
Router#show ospf database database-summary detail


            OSPF Router with ID (192.168.0.1) (Process ID 1)

Router 192.168.0.4 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        0        0        0
  Network       0        0        0
  Summary Net   0        0        0
  Summary ASBR  0        0        0
  Type-5 Ext    697      0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Opaque AS     6        0        0
  Total         703      0        0

Router 192.168.0.1 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        1        0        0
  Network       0        0        0
  Summary Net   0        0        0
  Summary ASBR  0        0        0
  Type-5 Ext    0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   64       0        0
  Opaque AS     0        0        0
  Total         65       0        0

Router 192.168.0.2 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        1        0        0
  Network       0        0        0
  Summary Net   21       0        0
  Summary ASBR  2        0        0
  Type-5 Ext    0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   21       0        0
  Opaque AS     0        0        0
  Total         45       0        0

Router 192.168.0.6 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        1        0        0
  Network       0        0        0
  Summary Net   21       0        0
  Summary ASBR  2        0        0
  Type-5 Ext    0        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   19       0        0
```

```
  Opaque AS     0        0        0
  Total         43       0        0

Router 192.168.0.3 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        0        0        0
  Network       0        0        0
  Summary Net   0        0        0
  Summary ASBR  0        0        0
  Type-5 Ext    7        0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Opaque AS     6        0        0
  Total         13       0        0
```

- **show ospf database database-summary adv-router *router ID*** command displays the router information and the LSAs received from the particular router.

```
Router#show ospf database database-summary adv-router 192.168.0.4


            OSPF Router with ID (192.168.0.1) (Process ID 1)

Router 192.168.0.4 LSA summary
  LSA Type      Count    Delete   Maxage
  Router        0        0        0
  Network       0        0        0
  Summary Net   0        0        0
  Summary ASBR  0        0        0
  Type-5 Ext    697      0        0
  Type-7 Ext    0        0        0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Opaque AS     6        0        0
  Total         703      0        0
```

Limiting the Maximum Redistributed Type-3 LSA Prefixes in OSPF

Table 6: Feature History Table

Feature Name	Release	Description
Limiting the Maximum Redistributed Type-3 LSA Prefixes in OSPF	Release 7.9.1	By default, the maximum redistributed Type-3 LSA prefixes for a given OSPF process is now limited to 100000. This mechanism prevents OSPF from redistributing a large number of prefixes as Type-3 LSAs and therefore preventing high CPU utilization and memory shortages. Once the number of redistributed prefixes is reached or exceeds the threshold value, the system log message is generated, and no more prefixes are redistributed.

Redistribution allows different routing protocols to exchange routing information. This is used to allow connectivity to span multiple routing protocols. Open Shortest Path First (OSPF) supports a user-defined maximum number of prefixes (routes) that are allowed to be redistributed into OSPF from other protocols or other OSPF instances.

Prior to Release 7.9.1, the maximum redistributed-prefixes limit was applied only to those prefixes that are redistributed as Type-5 and Type-7 LSAs. Starting from Release 7.9.1, the maximum redistributed-prefixes limit is also applied to the prefixes that are redistributed as Type-3 LSAs. The maximum redistributed Type-3 LSA prefixes for a given OSPF process is limited to 100000.

If the router redistributes more than 10000 prefixes as Type 3, 5, or 7 LSAs, then you must configure a higher limit using the **maximum redistributed-prefixes** command.

Starting from Release 7.9.1, if the **redistribute protocol lsa-type summary** command is configured to redistribute the routes from particular protocol as Type-3 LSAs, then those Type-3 LSAs are accounted for maximum redistributed prefixes.

System output messages

The range of prefixes that are redistributed as Type-3 LSAs is 1-4294967295. The threshold percentage to log warning is 75%. The system log message is generated every 5% above the default or configured threshold value until 100% is reached.

When the number of LSAs reaches or exceeds the threshold limit, the router displays the following logs:

The redistributed prefixes count reached the maximum limit

```
%ROUTING-OSPF-4-REDIST_THR_PFX : Reached Redistribution prefix threshold
in vrf "default", current (70%) 700 prefixes, limit  1000
```

The redistributed prefixes count exceeds the threshold percentage

```
%ROUTING-OSPF-1-REDIST_MAX_PFX : Redistribution prefix limit has been
reached in vrf "default" - current 1000 prefixes, limit 1000
```

The redistributed prefixes count falls below the threshold percentage

```
%ROUTING-OSPF-5-REDIST_MAX_PFX_RECOVER : Recovered from Redistribution
limit-hit scenario in vrf "default", prefix count less than threshold -
current (69%) 699 prefixes, limit 1000
```

The **show ospf** command displays the maximum number of redistributed prefixes, which is configured at 1000.

```
Router #show ospf
Thu Dec  8 18:16:48.332 IST

 Routing Process "ospf 1" with ID 192.168.0.1
 Role: Primary Active
 NSR (Non-stop routing) is Enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 It is an autonomous system boundary router
 Maximum number of non self-generated LSA allowed 1000
    Current number of non self-generated LSA 804
    Threshold for warning message 60%
    Ignore-time 1 minutes, reset-time 2 minutes
    Ignore-count allowed 2, current ignore-count 0
 Redistributing External Routes from,
    static
 Maximum number of redistributed prefixes 1000
    Threshold for warning message 70%
    Current number of redistributed prefixes 100
```