



Cisco High Availability

The Cisco High Availability (HA) technology is a network resiliency solution that

- enables rapid recovery from disruptions,
- ensures fault transparency to users and network applications, and
- maximizes network uptime through integrated hardware and software design.

These mechanisms that help maintain continuous network operations even during failure events or maintenance.

This section explores aspects of Cisco High Availability that may be used with the Cisco Catalyst 8300 Series Edge Platform.

- [#unique_107](#)
- [#unique_108](#)
- [Interchassis High Availability, on page 1](#)
- [Bidirectional Forwarding Detection, on page 3](#)
- [Verify Interchassis High Availability, on page 4](#)
- [Verify BFD Offload, on page 11](#)

Interchassis High Availability

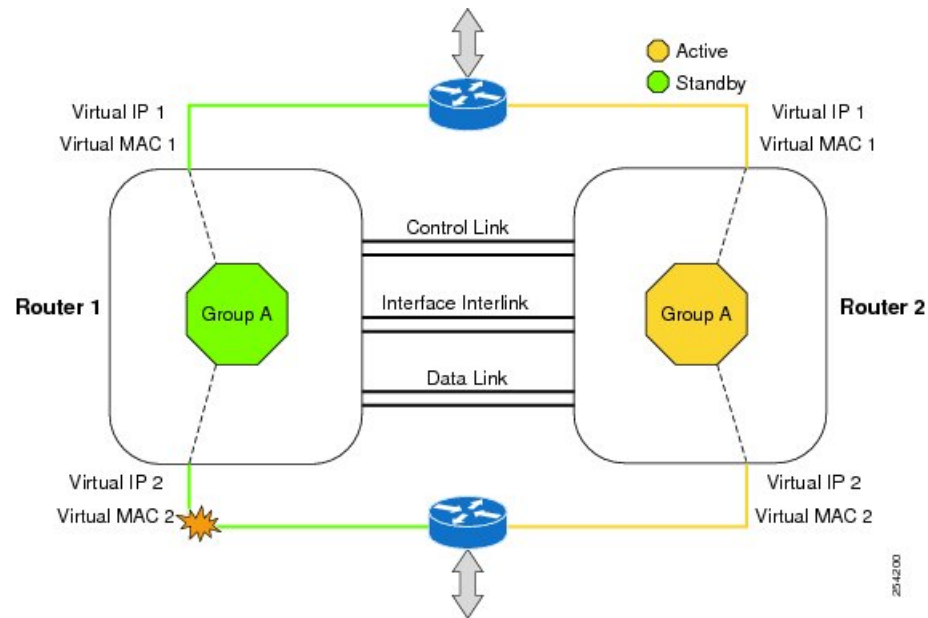
The Interchassis High Availability feature is also known as the box-to-box redundancy feature is a system redundancy solution that

- enables two devices to form a backup pair for high-availability operation,
- automatically monitors device status to detect failures, and
- seamlessly transfers call signaling and media forwarding to a standby device during failover.

Redundancy groups

Groups of redundant interfaces are known as redundancy groups. The figure depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of devices that have a single outgoing interface.

Figure 1: Redundancy group configuration



The devices are connected by a configurable control link and data synchronization link. The control link is used to communicate the status of the devices. The data synchronization link transfers stateful information to synchronize the database for the calls and media flows. Each pair of redundant interfaces is configured with the same unique ID number, also known as the RII. For information on configuring Interchassis High Availability on your device, see [#unique_110](#).

Prerequisites

Before configuring Interchassis High Availability, ensure that you meet all these requirements:

- The active device and the standby device must run on the identical version of the Cisco IOS XE software.
- The active device and the standby device must be connected through an L2 connection for the control path.
- Either the Network Time Protocol (NTP) must be configured or the clock must be set identical on both devices to allow timestamps and call timers to match.
- Virtual Routing and Forwarding (VRF) must be defined in the same order on both active and standby devices for an accurate synchronization of data.
- The latency times must be minimal on all control and data links to prevent timeouts.
- Physically redundant links, such as Gigabit EtherChannel, must be used for the control and data paths.

Limitations

- The failover time for a box-to-box application is higher for a non-box-to-box application.
- LAN and MESH scenarios are not supported.

- VRFs are not supported and cannot be configured under ZBFW High Availability data and control interfaces.
- The maximum number of virtual MACs supported by the Front Panel Gigabit Ethernet (FPGE) interfaces depends on the platform. For information about the FPGE interfaces, see the [Hardware Installation Guide for Cisco Catalyst 8300 Edge Platform](#).
- When the configuration is replicated to the standby device, it is not committed to the startup configuration; it is in the running configuration. A user must run the **write memory** command to commit the changes that have been synchronized from the active device, on the standby device.

Configure Interchassis High Availability

For information on configuring Interchassis High Availability on the device, see the [IP Addressing: NAT Configuration Guide, Cisco IOS XE Release 3S](#).

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a protocol that

- rapidly detects failures in the forwarding path between two routers,
- provides fast-forwarding path-failure detection times for all media type, routing protocol, or topology, and
- provides consistent and predictable failure detection rates for streamlined network convergence.

In addition to fast-forwarding path-failure detection, BFD provides a consistent failure detection method for network administrators. Because a network administrator can use BFD to detect forwarding path failures at a uniform rate rather than variable rates for different routing protocol hello mechanisms, network profiling and planning is easier, and reconvergence time is consistent and predictable. For more information on BFD, see the [IP Routing: BFD Configuration Guide](#).

Bidirectional Forwarding Detection Offload

The Bidirectional Forwarding Detection Offload feature allows the offload of BFD session management to the forwarding engine, resulting in faster failure detection. By sending rapid failure detection packets to routing protocols, BFD offload helps recalculating the routing table and reduces network convergence time. See [#unique_116](#).

Limitation

These are the limitations when configuring the BFD offload.

- Only BFD version 1 is supported.
- When configured, only offloaded BFD sessions are supported; BFD session on RP are not supported.
- Only Asynchronous mode or no echo mode of BFD is supported.
- 511 asynchronous BFD sessions are supported.

- BFD hardware offload is supported for IPv4 sessions with non-echo mode only.
- BFD offload is supported only on port-channel interfaces.
- BFD offload is supported only for the Ethernet interface.
- BFD offload is not supported for IPv6 BFD sessions.
- BFD offload is not supported for BFD with TE/FRR.

Configure Bidirectional Forwarding

For information on configuring BFD on your device, see the [IP Routing BFD Configuration Guide](#).

For BFD commands, see the [Cisco IOS IP Routing: Protocol-Independent Command Reference](#) document.

Configure BFD Offload

BFD offload functionality is enabled by default. You can configure BFD hardware offload on the route processor. For more information, see the [IP Routing BFD Configuration Guide](#).

Verify Interchassis High Availability

You can verify the Interchassis High Availability configuration using these **show** commands. This section provides some examples to verify the configuration.

Use these **show** commands to verify the Interchassis High Availability.

- **show redundancy application group [group-id | all]**
- **show redundancy application transport {client | group [group-id]}**
- **show redundancy application control-interface group [group-id]**
- **show redundancy application faults group [group-id]**
- **show redundancy application protocol {protocol-id | group [group-id]}**
- **show redundancy application if-mgr group [group-id]**
- **show redundancy application data-interface group [group-id]**

This example shows the redundancy application groups configured on the device:

```
Router# show redundancy application group
Group ID      Group Name      State
-----
1             Generic-Redundancy-1  STANDBY
2             Generic-Redundancy2   ACTIVE
```

This example shows the details of redundancy application group 1:

```
Router# show redundancy application group 1
Group ID:1
Group Name:Generic-Redundancy-1

Administrative State: No Shutdown
```

```
Aggregate operational state : Up
My Role: STANDBY
Peer Role: ACTIVE
Peer Presence: Yes
Peer Comm: Yes
Peer Progression Started: Yes
```

```
RF Domain: btob-one
RF state: STANDBY HOT
Peer RF state: ACTIVE
```

This example shows the details of redundancy application group 2:

Router# show redundancy application group 2

```
Group ID:2
Group Name:Generic-Redundancy2

Administrative State: No Shutdown
Aggregate operational state : Up
My Role: ACTIVE
Peer Role: STANDBY
Peer Presence: Yes
Peer Comm: Yes
Peer Progression Started: Yes
```

```
RF Domain: btob-two
RF state: ACTIVE
Peer RF state: STANDBY HOT
```

This example shows details of the redundancy application transport client:

Router# show redundancy application transport client

Client	Conn#	Priority	Interface	L3	L4
(0)RF	0	1	CTRL	IPV4	SCTP
(1)MCP_HA	1	1	DATA	IPV4	UDP_REL
(4)AR	0	1	ASYM	IPV4	UDP
(5)CF	0	1	DATA	IPV4	SCTP

This example shows configuration details for the redundancy application transport group:

Router# show redundancy application transport group

```
Transport Information for RG (1)
Client = RF
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
0    0        192.0.2.8      59000  192.0.2.4      59000  CTRL  IPV4  SCTP
Client = MCP_HA
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
1    1        10.10.2.10     53000  10.10.6.9      53000  DATA  IPV4  UDP_REL
Client = AR
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
2    0        192.0.2.3       0      192.0.2.3       0      NONE_IN NONE_L3 NONE_L4
Client = CF
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
3    0        10.10.2.10     59001  10.10.6.9      59001  DATA  IPV4  SCTP
Transport Information for RG (2)
Client = RF
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
8    0        192.0.2.8      59004  192.0.2.2      59004  CTRL  IPV4  SCTP
Client = MCP_HA
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
9    1        10.10.2.10     53002  10.10.6.9      53002  DATA  IPV4  UDP_REL
Client = AR
TI   conn_id my_ip          my_port peer_ip          peer_por intf    L3    L4
```

```

10 0      192.0.2.3      0      192.0.2.3      0      NONE_IN NONE_L3 NONE_L4
Client = CF
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
11 0      10.10.2.10    59005 10.10.6.9      59005  DATA   IPV4   Sctp

```

This example shows the configuration details of redundancy application transport group 1:

```

Router# show redundancy application transport group 1
Transport Information for RG (1)
Client = RF
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
0  0      192.0.2.8      59000 192.0.2.4      59000  CTRL   IPV4   Sctp
Client = MCP_HA
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
1  1      10.10.2.10    53000 10.10.2.10    53000  DATA   IPV4   UDP_REL
Client = AR
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
2  0      192.0.2.3      0      192.0.2.3      0      NONE_IN NONE_L3 NONE_L4
Client = CF
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
3  0      10.10.2.10    59001 10.10.2.10    59001  DATA   IPV4   Sctp

```

This example shows configuration details of redundancy application transport group 2:

```

Router# show redundancy application transport group 2
Transport Information for RG (2)
Client = RF
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
8  0      192.0.2.8      59004 192.0.2.4      59004  CTRL   IPV4   Sctp
Client = MCP_HA
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
9  1      10.10.2.10    53002 10.10.2.10    53002  DATA   IPV4   UDP_REL
Client = AR
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
10 0      192.0.2.3      0      192.0.2.3      0      NONE_IN NONE_L3 NONE_L4
Client = CF
TI  conn_id my_ip      my_port peer_ip      peer_por intf    L3    L4
11 0      10.10.2.10    59005 10.10.2.10    59005  DATA   IPV4   Sctp

```

This example shows configuration details of the redundancy application control-interface group:

```

Router# show redundancy application control-interface group
The control interface for rg[1] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0

The control interface for rg[2] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0

```

This example shows configuration details of the redundancy application control-interface group 1:

```

Router# show redundancy application control-interface group 1
The control interface for rg[1] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0

```

This example shows configuration details of the redundancy application control-interface group 2:

```

Router# show redundancy application control-interface group 2
The control interface for rg[2] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0

```

This example shows configuration details of the redundancy application faults group:

```

Router# show redundancy application faults group
Faults states Group 1 info:
Runtime priority: [50]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2
Faults states Group 2 info:
Runtime priority: [135]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2

```

This example shows configuration details specific to redundancy application faults group 1:

```

Router# show redundancy application faults group 1
Faults states Group 1 info:
Runtime priority: [50]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2

```

This example shows configuration details specific to redundancy application faults group 2:

```

Router# show redundancy application faults group 2
Faults states Group 2 info:
Runtime priority: [135]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2

```

This example shows configuration details for the redundancy application protocol group:

```

Router# show redundancy application protocol group
RG Protocol RG 1
-----
Role: Standby
Negotiation: Enabled
Priority: 50
Protocol state: Standby-hot
Ctrl Intf(s) state: Up
Active Peer: address 192.0.4.2, priority 150, intf Gi0/0/0
Standby Peer: Local
Log counters:
role change to active: 0
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

RG Media Context for RG 1
-----
Ctx State: Standby
Protocol ID: 1
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000

```

```

Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 117, Bytes 7254, HA Seq 0, Seq Number 117, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 0
Active Peer: Present. Hold Timer: 10000
Pkts 115, Bytes 3910, HA Seq 0, Seq Number 1453975, Pkt Loss 0

```

```

RG Protocol RG 2
-----
Role: Active
Negotiation: Enabled
Priority: 135
Protocol state: Active
Ctrl Intf(s) state: Up
Active Peer: Local
Standby Peer: address 192.0.4.2, priority 130, intf Gi0/0/0
Log counters:
role change to active: 1
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

```

```

RG Media Context for RG 2
-----
Ctx State: Active
Protocol ID: 2
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 118, Bytes 7316, HA Seq 0, Seq Number 118, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 1
Standby Peer: Present. Hold Timer: 10000
Pkts 102, Bytes 3468, HA Seq 0, Seq Number 1453977, Pkt Loss 0

```

This example shows configuration details for the redundancy application protocol group 1:

```

Router# show redundancy application protocol group 1
RG Protocol RG 1
-----
Role: Standby
Negotiation: Enabled
Priority: 50
Protocol state: Standby-hot
Ctrl Intf(s) state: Up
Active Peer: address 192.0.4.2, priority 150, intf Gi0/0/0
Standby Peer: Local
Log counters:
role change to active: 0
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

```



```

RG Media Context for RG 1
-----
Ctx State: Standby
Protocol ID: 1
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 120, Bytes 7440, HA Seq 0, Seq Number 120, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 0
Active Peer: Present. Hold Timer: 10000
Pkts 118, Bytes 4012, HA Seq 0, Seq Number 1453978, Pkt Loss 0

```

This example shows configuration details for the redundancy application protocol group 2:

Router# show redundancy application protocol group 2

```

RG Protocol RG 2
-----
Role: Active
Negotiation: Enabled
Priority: 135
Protocol state: Active
Ctrl Intf(s) state: Up
Active Peer: Local
Standby Peer: address 192.0.4.2, priority 130, intf Gi0/0/0
Log counters:
role change to active: 1
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

```

```

RG Media Context for RG 2
-----
Ctx State: Active
Protocol ID: 2
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 123, Bytes 7626, HA Seq 0, Seq Number 123, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 1
Standby Peer: Present. Hold Timer: 10000
Pkts 107, Bytes 3638, HA Seq 0, Seq Number 1453982, Pkt Loss 0

```

This example shows configuration details for the redundancy application protocol 1:

Router# show redundancy application protocol 1

```

Protocol id: 1, name: rg-protocol-1
BFD: ENABLE
Hello timer in msecs: 3000
Hold timer in msecs: 10000
OVLd-1#show redundancy application protocol 2
Protocol id: 2, name: rg-protocol-2
BFD: ENABLE

```

```
Hello timer in msecs: 3000
Hold timer in msecs: 10000
```

This example shows configuration details for redundancy application interface manager group:

```
Router# show redundancy application if-mgr group
```

```
RG ID: 1
```

```
=====
```

```
interface      GigabitEthernet0/0/3.152
```

```
-----
```

```
VMAC           0007.b421.4e21
```

```
VIP            203.0.113.1
```

```
Shut           shut
```

```
Decrement      10
```

```
interface      GigabitEthernet0/0/2.152
```

```
-----
```

```
VMAC           0007.b421.5209
```

```
VIP            203.0.113.4
```

```
Shut           shut
```

```
Decrement      10
```

```
RG ID: 2
```

```
=====
```

```
interface      GigabitEthernet0/0/3.166
```

```
-----
```

```
VMAC           0007.b422.14d6
```

```
VIP            203.0.113.6
```

```
Shut           no shut
```

```
Decrement      10
```

```
interface      GigabitEthernet0/0/2.166
```

```
-----
```

```
VMAC           0007.b422.0d06
```

```
VIP            203.0.113.9
```

```
Shut           no shut
```

```
Decrement      10
```

These examples show configuration details for redundancy application interface manager group 1 and group 2:

```
Router# show redundancy application if-mgr group 1
```

```
RG ID: 1
```

```
=====
```

```
interface      GigabitEthernet0/0/3.152
```

```
-----
```

```
VMAC           0007.b421.4e21
```

```
VIP            203.0.113.3
```

```
Shut           shut
```

```
Decrement      10
```

```
interface      GigabitEthernet0/0/2.152
```

```
-----
```

```
VMAC           0007.b421.5209
```

```
VIP            203.0.113.2
```

```
Shut           shut
```

```
Decrement      10
```

```
Router# show redundancy application if-mgr group 2
```

```
RG ID: 2
```

```

=====
interface      GigabitEthernet0/0/3.166
-----
VMAC           0007.b422.14d6
VIP            203.0.113.5
Shut           no shut
Decrement      10

interface      GigabitEthernet0/0/2.166
-----
VMAC           0007.b422.0d06
VIP            203.0.113.7
Shut           no shut
Decrement      10

```

This example shows configuration details for redundancy application data-interface group:

```
Router# show redundancy application data-interface group
```

The data interface for rg[1] is GigabitEthernet0/0/1

The data interface for rg[2] is GigabitEthernet0/0/1

These examples show configuration details specific to redundancy application data-interface group 1 and group 2:

```
Router# show redundancy application data-interface group 1
```

The data interface for rg[1] is GigabitEthernet0/0/1

```
Router # show redundancy application data-interface group 2
```

The data interface for rg[2] is GigabitEthernet0/0/1

Verify BFD Offload



Note Configuration of BFD Offload is described in [Configure Bidirectional Forwarding](#).

Use these commands to verify and monitor BFD offload feature on your device.

- **show bfd neighbors [details]**
- **debug bfd [packet | event]**
- **debug bfd event**

The **show bfd neighbors** command displays the BFD adjacency database:

```
Router# show bfd neighbor
```

```
IPv4 Sessions
NeighAddr      LD/RD      RH/RS      State      Int
192.0.2.1      362/1277   Up         Up         Gi0/0/1.2
192.0.2.5      445/1278   Up         Up         Gi0/0/1.3
192.0.2.3      1093/961   Up         Up         Gi0/0/1.4
192.0.2.2      1244/946   Up         Up         Gi0/0/1.5
192.0.2.6      1094/937   Up         Up         Gi0/0/1.6
192.0.2.7      1097/1260  Up         Up         Gi0/0/1.7
192.0.2.4      1098/929   Up         Up         Gi0/0/1.8
```

```

192.0.2.9          1111/928      Up      Up      Gi0/0/1.9
192.0.2.8          1100/1254     Up      Up      Gi0/0/1.10

```

The **debug bfd neighbor detail** command displays the debugging information related to BFD packets:

Router# show bfd neighbor detail

```

IPv4 Sessions
NeighAddr          LD/RD          RH/RS      State      Int
192.0.2.1          362/1277      Up         Up         Gi0/0/1.2
Session state is UP and not using echo function.
Session Host: Hardware
OurAddr: 192.0.2.2
Handle: 33
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holddown (hits): 0(0), Hello (hits): 50(0)
Rx Count: 3465, Rx Interval (ms) min/max/avg: 42/51/46
Tx Count: 3466, Tx Interval (ms) min/max/avg: 39/52/46
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: CEF EIGRP
Uptime: 00:02:50
Last packet: Version: 1                - Diagnostic: 0
              State bit: Up             - Demand bit: 0
              Poll bit: 0               - Final bit: 0
              C bit: 1
              Multiplier: 3             - Length: 24
              My Discr.: 1277           - Your Discr.: 362
              Min tx interval: 50000    - Min rx interval: 50000
              Min Echo interval: 0

```

The **show bfd summary** command displays the BFD summary:

Router# show bfd summary

	Session	Up	Down
Total	400	400	0

The **show bfd drops** command displays the number of packets dropped in BFD:

Router# show bfd drops

```

BFD Drop Statistics

```

	IPV4	IPV6	IPV4-M	IPV6-M	MPLS_PW	MPLS_TP_LSP
Invalid TTL	0	0	0	0	0	0
BFD Not Configured	0	0	0	0	0	0
No BFD Adjacency	33	0	0	0	0	0
Invalid Header Bits	0	0	0	0	0	0
Invalid Discriminator	1	0	0	0	0	0
Session AdminDown	94	0	0	0	0	0
Authen invalid BFD ver	0	0	0	0	0	0
Authen invalid len	0	0	0	0	0	0
Authen invalid seq	0	0	0	0	0	0
Authen failed	0	0	0	0	0	0

The **debug bfd packet** command displays debugging information about BFD control packets.

Router# debug bfd packet

```

*Nov 12 23:08:27.982: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/0 diag:0(No Diagnostic)
  Down C cnt:4 ttl:254 (0)
*Nov 12 23:08:27.982: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941 diag:3(Neighbor
Signaled Session Down) Init C cnt:44 (0)
*Nov 12 23:08:28.007: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No Diagnostic)
  Up PC cnt:4 ttl:254 (0)

```

```
*Nov 12 23:08:28.007: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941 diag:0(No Diagnostic)
Up F C cnt:0 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No Diagnostic)
Up FC cnt:0 ttl:254 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941 diag:0(No Diagnostic)
Up C cnt:0 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/0 diag:0(No Diagnostic)
Down C cnt:3 ttl:254 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907 diag:3(Neighbor
Signaled Session Down) Init C cnt:43 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No Diagnostic)
Up C cnt:0 ttl:254 (0)
*Nov 12 23:08:28.626: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No Diagnostic)
Up PC cnt:3 ttl:254 (0)
*Nov 12 23:08:28.626: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907 diag:0(No Diagnostic)
Up F C cnt:0 (0)
*Nov 12 23:08:28.645: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No Diagnostic)
Up C cnt:0 ttl:254 (0)
*Nov 12 23:08:28.700: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No Diagnostic)
Up FC cnt:0 ttl:254 (0)
*Nov 12 23:08:28.700: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907 diag:0(No Diagnostic)
Up C cnt:0 (0)
*Nov 12 23:08:28.993: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No Diagnostic)
Up C cnt:0 ttl:254 (0)
```

The **debug bfd event** displays debugging information about BFD state transitions:

Router# deb bfd event

```
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.6, ld:1401, handle:77,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.6, ld:1401, handle:77,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.10, ld:1400, handle:39,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.10, ld:1400, handle:39,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.8, ld:1399, handle:25,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.8, ld:1399, handle:25,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.5, ld:1403, handle:173,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.6, ld:1403, handle:173,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.4, ld:1402, handle:95,
event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.4, ld:1402, handle:95,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Timers: Tx timer 1000000 Detect timer 0
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Flags: Poll 0 Final 0
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Buffer: 0x23480318 0x0000057C 0x00000000
0x000F4240 0x000F4240 0x00000000 size 24
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Timers: Tx timer 1000000 Detect timer 0
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Flags: Poll 0 Final 0
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Buffer: 0x23480318 0x0000057D 0x00000000
0x000F4240 0x000F4240 0x00000000 size 24
*Nov 12 23:11:30.649: BFD-DEBUG Packet: Rx IP:192.0.2.6 ld/rd:1601/1404
diag:7(Administratively Down) AdminDown C cnt:0 ttl:254 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: V1 FSM ld:1404 handle:207 event:RX ADMINDOWN state:UP
(0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: resetting timestamps ld:1404 handle:207 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.1, ld:1404, handle:207,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:1404/0 diag:3(Neighbor Signaled
```

```
Session Down) Down C cnt:0 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1620/1405
diag:7(Administratively Down) AdminDown C cnt:0 ttl:254 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: V1 FSM ld:1405 handle:209 event:RX ADMINDOWN state:UP
(0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: resetting timestamps ld:1405 handle:209 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.1, ld:1405, handle:209,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Tx IP:192.0.2.7 ld/rd:1405/0 diag:3(Neighbor Signaled
Session Down) Down C cnt:0 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.7, ld:1404, handle:207,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.7, ld:1404, handle:207,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.7, ld:1405, handle:209,
event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.7, ld:1405, handle:209,
event:DOWN adminDown, (0)
*Nov 12 23:11:31.035: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.0.2.8
```