Stack Manager and High Availability Configuration Guide, Cisco IOS XE Fuji 16.8.x (Catalyst 9300 Switches)

First Published: 2018-04-06

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Managing Switch Stacks

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Switch Stacks

- All the switches in the stack must be running the same license level as the active switch. For information about license levels, see the *System Management* section of this guide.
- All the switches in the switch stack must be running compatible software versions.

Restrictions for Switch Stacks

Information About Switch Stacks

Switch Stack Overview

Switch Stack Bridge ID and MAC Address

The MAC address of the determines the stack MAC address.

When the stack initializes, the MAC address of the determines the bridge ID that identifies the stack in the network.

If the changes, the MAC address of the new determines the new bridge ID and stack MAC address.

If the entire switch stack reloads, the switch stack uses the MAC address of the .

Persistent MAC Address on the Switch Stack

Note You can also configure stack MAC persistency so that the stack MAC address never changes to the new MAC address, by using the **stack-mac persistent timer 0** command.

Upgrading a Switch Running Incompatible Software

The auto-upgrade and auto-advise features enable a switch with software packages that are incompatible with the switch stack to be upgraded to a compatible software version so that it can join the switch stack.

Switch Stack Management Connectivity

You manage the switch stack and the stack member interfaces through the . You can use the CLI, SNMP, and supported network management applications such as CiscoWorks. You cannot manage stack members on an individual basis.

How to Configure a Switch Stack

Monitoring the Device Stack

Table 1: Commands for Displaying Stack Information

Command	Description
show module	Displays summary informaton about the stack.
show switch detail	Displays detailed information about the stack.
show switch neighbors	Displays the stack neighbors.
show switch stack-ports [**summary**]	Displays port information for the stack. Use the **summary** keyword to display the stack cable length, the stack link status, and the loopback status.
show redundancy	Displays the redundant system and the current processor information. The redundant system information includes the system uptime, standby failures, switchover reason, hardware, configured and operating redundancy mode. The current processor information displayed includes the active location, the software state, the uptime in the current state and so on.
show redundancy state	Displays all the redundancy states of the active and standby devices.

Configuration Examples for Switch Stacks

Switch Stack Configuration Scenarios

Most of these switch stack configuration scenarios assume that at least two devices are connected through their StackWise ports.

Table 2: Configuration Scenarios

Scenario		Result
Active switch election specifically determined by existing active switches	Connect two powered-on switch stacks through the StackWise ports.	Only one of the two active switches becomes the new active switch.

Scenario		Result
Active switch election specifically determined by the stack member priority value	1. Connect two switches through their StackWise ports. 2. Use the **switch** *stack-member-number* **priority** *new-priority-number* command to set one stack member with a higher member priority value. 3. Restart both member switches at the same time.	The stack member with the higher priority value is elected active switch.
Active switch election specifically determined by the configuration file	Assuming that both member switches have the same priority value: 1. Make sure that one stack member has a default configuration and that the other stack member has a saved (nondefault) configuration file. 2. Restart both member switches at the same time.	The stack member with the saved configuration file is elected active switch.
Active switch election specifically determined by the MAC address	Assuming that both member switches have the same priority value, configuration file, and license level, restart both member switches at the same time.	The stack member with the lower MAC address is elected active switch .
Stack member number conflict	Assuming that one stack member has a higher priority value than the other stack member: 1. Ensure that both member switches have the same stack member number. If necessary, use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* command. 2. Restart both member switches at the same time.	The stack member with the higher priority value retains its stack member number. The other stack member has a new stack member number.
Add a stack member	1. Power off the new switch. 2. Through their StackWise ports, connect the new switch to a powered-on switch stack. 3. Power on the new switch.	The active switch is retained. The new switch is added to the switch stack.
Active switch failure	Remove (or power off) the active switch.	

Scenario		Result
Add member switches	1. Through their StackWise ports, connect devices. 2. Power on all devices.	Two devices become active switches. One active switch has member switches. The other active switch remains as a standalone device. Use the Mode button and port LEDs on the device to identify which devices are active switches and which devices belong to each active switch.

Enabling the Persistent MAC Address Feature: Example

This example shows how to configure the persistent MAC address feature for a 7-minute time delay and to verify the configuration:

```
(config)# stack-mac persistent timer 7
  WARNING: The stack continues to use the base MAC of the old Master
  WARNING: as the stack MAC after a master switchover until the MAC
  WARNING: persistency timer expires. During this time the Network
  WARNING: Administrators must make sure that the old stack-mac does
  WARNING: not appear elsewhere in this network domain. If it does,
  WARNING: user traffic may be blackholed.
  (config)# end
  # show switch
  Switch/Stack Mac Address : 0016.4727.a900
  Mac persistency wait time: 7 mins
                                             H/W   Current
  Switch#  Role   Mac Address     Priority Version  State
  ----------------------------------------------------------
  *1        0016.4727.a900     1      P2B     Ready
```

show switch stack-ports summary Command Output: Example

Only Port 1 on stack member 2 is disabled.

```
# show switch stack-ports summary
  #/  Stack   Neighbor   Cable    Link   Link   Sync      #         In
   Port#     Port             Length    OK   Active   OK    Changes   Loopback
            Status                                          To LinkOK
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     OK          3      50 cm     Yes    Yes     Yes          1         No
    1/2     Down       None    3 m       Yes    No      Yes          1         No
    2/1     Down       None    3 m       Yes    No      Yes          1         No
    2/2     OK          3      50 cm     Yes    Yes     Yes          1         No
    3/1     OK          2      50 cm     Yes    Yes     Yes          1         No
    3/2     OK          1      50 cm     Yes    Yes     Yes          1         No
```

Table 3: show switch stack-ports summary Command Output

Field	Description
Switch#/Port#	Member number and its stack port number.
Stack Port Status	Status of the stack port. • Absent—No cable is detected on the stack port. • Down—A cable is detected, but either no connected neighbor is up, or the stack port is disabled. • OK—A cable is detected, and the connected neighbor is up.
Neighbor	Switch number of the active member at the other end of the stack cable.
Cable Length	Valid lengths are 50 cm, 1 m, or 3 m. If the switch cannot detect the cable length, the value is *no cable*. The cable might not be connected, or the link might be unreliable.
Link OK	Whether the stack cable is connected and functional. There may or may not be a neighbor connected on the other end. The *link partner* is a stack port on a neighbor switch. • No—There is no stack cable connected to this port or the stack cable is not functional. • Yes—There is a functional stack cable connected to this port.
Link Active	Whether a neighbor is connected on the other end of the stack cable. • No—No neighbor is detected on the other end. The port cannot send traffic over this link. • Yes—A neighbor is detected on the other end. The port can send traffic over this link.
Sync OK	Whether the link partner sends valid protocol messages to the stack port. • No—The link partner does not send valid protocol messages to the stack port. • Yes—The link partner sends valid protocol messages to the port.
# Changes to LinkOK	The relative stability of the link. If a large number of changes occur in a short period of time, link flapping can occur.
In Loopback	Whether a stack cable is attached to a stack port on the member. • No—At least one stack port on the member has an attached stack cable. • Yes—None of the stack ports on the member has an attached stack cable.

Software Loopback: Examples

In a stack with three members, stack cables connect all the members:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync   #Changes   In
            Status            Length    OK    Active  OK     To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------   --------
    1/1     OK         3      50 cm     Yes    Yes    Yes        1        No
    1/2     OK         2      3 m       Yes    Yes    Yes        1        No
    2/1     OK         1      3 m       Yes    Yes    Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     OK         1      50 cm     Yes    Yes    Yes        1        No
```

If you disconnect the stack cable from Port 1 on Switch 1, these messages appear:

```
01:09:55: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 3 has changed to state DOWN
  01:09:56: %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 1 has changed to state DOWN

  # show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback

  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent     None    No cable  No     No     No         1        No
    1/2     OK         2      3 m       Yes    Yes    Yes        1        No
    2/1     OK         1      3 m       Yes    Yes    Yes        1        No
    2/2     OK         3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK         2      50 cm     Yes    Yes    Yes        1        No
    3/2     Down       None    50 cm     No     No     No         1        No
```

If you disconnect the stack cable from Port 2 on Switch 1, the stack splits.

Switch 2 and Switch 3 are now in a two-member stack connected through stack cables:

```
# show sw stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    2/1     Down       None    3 m       No     No     No         1        No
    2/2     OK          3      50 cm     Yes    Yes    Yes        1        No
    3/1     OK          2      50 cm     Yes    Yes    Yes        1        No
    3/2     Down       None    50 cm     No     No     No         1        No
```

Switch 1 is a standalone switch:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent     None    No cable  No     No     No         1        Yes
    1/2     Absent     None    No cable  No     No     No         1        Yes
```

Software Loopback with Connected Stack Cables: Examples

- On Port 1 on Switch 1, the port status is *Down*, and a cable is connected.

  On Port 2 on Switch 1, the port status is *Absent*, and no cable is connected.

  ```
  # show switch stack-ports summary
    #
   Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
              Status            Length    OK    Active  OK    To LinkOK  Loopback
    --------  ------  --------  --------  ---- ------  ----  ---------  --------
      1/1     Down      None    50 Cm      No     No     No       1          No
      1/2     Absent    None    No cable   No     No     No       1          No
  ```

- In a *physical loopback*, a cable connects both stack ports on a switch. You can use this configuration to test

  - Cables on a switch that is running properly
  - Stack ports with a cable that works properly

  ```
  # show switch stack-ports summary
    #
   Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
              Status            Length    OK    Active  OK    To LinkOK  Loopback
    --------  ------  --------  --------  ----  ------  ----  ---------  --------
      2/1       OK       2      50 cm     Yes    Yes    Yes      1         No
      2/2       OK       2      50 cm     Yes    Yes    Yes      1         No
  ```

  The port status shows that

  - Switch 2 is a standalone switch.
  - The ports can send and receive traffic.

Software Loopback with no Connected Stack Cable: Example

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1     Absent    None    No cable   No     No     No       1         Yes
    1/2     Absent    None    No cable   No     No     No       1         Yes
```

Finding a Disconnected Stack Cable: Example

Stack cables connect all stack members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status for the members:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status            Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
```

```
    1/1     OK         2      50 cm     Yes    Yes    Yes        0        No
    1/2     OK         2      50 cm     Yes    Yes    Yes        0        No
    2/1     OK         1      50 cm     Yes    Yes    Yes        0        No
    2/2     OK         1      50 cm     Yes    Yes    Yes        0        No
```

If you disconnect the cable from Port 2 on Switch 1, these messages appear:

```
%STACKMGR-4-STACK_LINK_CHANGE: Stack Port 1 Switch 2 has changed to state DOWN

  %STACKMGR-4-STACK_LINK_CHANGE: Stack Port 2 Switch 1 has changed to state DOWN
```

This is now the port status:

```
  # show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status          Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1       OK       2      50 cm     Yes    Yes    Yes        1        No
    1/2     Absent    None    No cable   No     No     No        2        No
    2/1      Down     None    50 cm      No     No     No        2        No
    2/2       OK       1      50 cm     Yes    Yes    Yes        1        No
```

Only one end of the cable connects to a stack port, Port 1 on Switch 2.

- The *Stack Port Status* value for Port 2 on Switch 1 is *Absent*, and the value for Port 1 on Switch 2 is *Down*.
- The *Cable Length* value is *No cable*.

Diagnosing the problem:

- Verify the cable connection for Port 2 on Switch 1.
- Port 2 on Switch 1 has a port or cable problem if
    - The *In Loopback* value is *Yes*.

  or

    - The *Link OK*, *Link Active*, or *Sync OK* value is *No*.

Fixing a Bad Connection Between Stack Ports: Example

Stack cables connect all members. Port 2 on Switch 1 connects to Port 1 on Switch 2.

This is the port status:

```
# show switch stack-ports summary
  #
 Sw#/Port#  Port    Neighbor  Cable     Link  Link    Sync  #Changes   In
            Status          Length    OK    Active  OK    To LinkOK  Loopback
  --------  ------  --------  --------  ----  ------  ----  ---------  --------
    1/1       OK       2      50 cm     Yes    Yes    Yes       1          No
    1/2      Down     None    50 cm      No     No     No       2          No
    2/1      Down     None    50 cm      No     No     No       2          No
  2/2       OK       1       50 cm      Yes     Yes     Yes       1          No
```

Diagnosing the problem:

- The Stack Port Status value is *Down*.
- Link OK, Link Active, and Sync OK values are *No*.
- The Cable Length value is *50 cm*. The switch detects and correctly identifies the cable.

The connection between Port 2 on Switch 1 and Port 1 on Switch 2 is unreliable on at least one of the connector pins.

Additional References for Switch Stacks

Related Documents

Related Topic	Document Title
Cabling and powering on a switch stack.	

Error Message Decoder

Description	Link
To help you research and resolve system error messages in this release, use the Error Message Decoder tool.	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

Standards and RFCs

Standard/RFC	Title
None	—

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and , use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for Switch Stacks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required

Table 4: Feature Information for Switch Stacks

Release	Feature	Feature Information
	Switch Stack	A switch stack can have up to eight stacking-capable switches connected through their StackWise ports. The stack members work together as a unified system. Layer 2 and Layer 3 protocols present the entire switch stack as a single entity to the network.

CHAPTER 2

Configuring Nonstop Forwarding with Stateful Switchover

Prerequisites for Cisco Nonstop Forwarding with Stateful Switchover

- Cisco nonstop forwarding (NSF) must be configured on a networking device that has been configured for stateful Switchover (SSO).

- Border Gateway Protocol (BGP) support in NSF requires that neighbor networking devices be NSF-aware; that is, devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable device discovers that a particular BGP neighbor does not have graceful restart capability, it does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability continue to have NSF-capable sessions with this NSF-capable networking device.

- Open Shortest Path First (OSPF) support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable device discovers that it has non-NSF-aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware devices continue to provide NSF capabilities.

Restrictions for Cisco Nonstop Forwarding with Stateful Switchover

The following are restrictions for configuring NSF with SSO:

- For NSF operation, you must have SSO configured on the device.
- All Layer 3 neighboring devices must be an NSF helper or NSF-capable to support graceful restart capability.
- For IETF, all neighboring devices must be running an NSF-aware software image.
- The Hot Standby Routing Protocol (HSRP) is not supported with NSF SSO.
- An NSF-aware device cannot support two NSF-capable peers performing an NSF restart operation at the same time. However, both neighbors can reestablish peering sessions after the NSF restart operation is complete.

Information About Cisco Nonstop Forwarding with Stateful Switchover

Overview of Cisco Nonstop Forwarding with Stateful Switchover

Cisco NSF works with the SSO feature. The device supports fault resistance by allowing a standby switch to take over if the active device becomes unavailable. NSF works with SSO to minimize the amount of time a network is unavailable.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

Cisco NSF with SSO allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF/SSO, peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors (FPs) while the standby router processor (RP) assumes control from the failed active RP during a switchover. NSF with SSO operation provides the ability of line cards and FPs to remain active through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active RP.

NSF provides the following benefits:

- Improved network availability—NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.
- Overall network stability—Network stability can be improved with the reduction in the number of route flaps that are created when devices in the network fail, and lose their routing tables.
- Neighboring devices do not detect a link flap—Because interfaces remain active during a switchover, neighboring devices do not detect a link flap (the link does not go down and come back up).
- Prevents routing flaps—Because SSO continues forwarding network traffic during a switchover, routing flaps are avoided.
- Maintains user sessions established prior to the switchover.
- If a stack member does not respond, that member is removed from the stack.

- If the standby device does not respond, a new standby device is elected.
- If the active device does not respond, the standby device becomes the active device.

SSO Operation

Cisco Nonstop Forwarding Operation

NSF always runs with SSO, and provides redundancy for Layer 3 traffic. NSF is supported by BGP, Enhanced Interior Gateway Routing Protocol (EIGRP), and OSPF routing protocols and also by Cisco Express Forwarding for forwarding. These routing protocols have been enhanced with NSF-capability and awareness, which means that devices running these protocols can detect a switchover and take necessary actions to continue forwarding network traffic and to recover route information from peer devices.

Each protocol depends on Cisco Express Forwarding to continue forwarding packets during switchover while routing protocols rebuild the Routing Information Base (RIB) tables. After the convergence of routing protocols, Cisco Express Forwarding updates the FIB table and removes stale route entries. Cisco Express Forwarding then updates the hardware with the new FIB information.

If the active is configured (with the **graceful-restart** command) for BGP, OSPF, or EIGRP routing protocols, routing updates are automatically sent during the active election.

NSF has two primary components:

- NSF-aware: A networking device is NSF-aware if it is running NSF-compatible software. If neighboring devices detect that an NSF device can still forward packets when an active election happens, this capability is referred to as NSF-awareness. Enhancements to the Layer 3 routing protocols (BGP, OSPF, and EIGRP) are designed to prevent route-flapping so that the Cisco Express Forwarding routing table does not time out or the NSF device does not drop routes. An NSF-aware device helps to send routing protocol information to the neighboring NSF device. NSF-awareness is enabled by default for EIGRP-stub, EIGRP, and OSPF protocols. NSF-awareness is disabled by default for BGP.
- NSF-capability: A device is NSF-capable if it is configured to support NSF; it rebuilds routing information from NSF-aware or NSF-capable neighbors. NSF works with SSO to minimize the amount of time that a Layer 3 network is unavailable following an active device election by continuing to forward IP packets. Reconvergence of Layer 3 routing protocols (BGP, OSPFv2, and EIGRP) is transparent to the user and happens automatically in the background. Routing protocols recover routing information from neighbor devices and rebuild the Cisco Express Forwarding table.

Cisco Express Forwarding

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by Cisco Express Forwarding. Cisco Express Forwarding maintains the Forwarding Information Base (FIB), and uses the FIB information that is current at the time of a switchover to continue forwarding packets during a switchover, to reduce traffic interruption during the switchover.

During normal NSF operation, Cisco Express Forwarding on the active synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby . Upon switchover, the standby initially has FIB and adjacency databases that are mirror images of those that were current on the active . Cisco Express Forwarding keeps the forwarding engine on the standby current with changes that are sent to it by Cisco Express Forwarding on the active . The forwarding engine can continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates cause prefix-by-prefix updates to Cisco Express Forwarding, which it uses to update the FIB and adjacency databases. Existing and new entries receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the forwarding engine during convergence. The device signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

Routing Protocols

Routing protocols run only on the active RP, and receive routing updates from neighbor devices. Routing protocols do not run on the standby RP. Following a switchover, routing protocols request that the NSF-aware neighbor devices send state information to help rebuild routing tables. Alternately, the Intermediate System-to-Intermediate System (IS-IS) protocol can be configured to synchronize state information from the active to the standby RP to help rebuild the routing table on the NSF-capable device in environments where neighbor devices are not NSF-aware.

Note For NSF operation, routing protocols depend on Cisco Express Forwarding to continue forwarding packets while routing protocols rebuild the routing information.

BGP Operation

When a NSF-capable device begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the NSF-capable device has “graceful restart capability.” Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable device and its BGP peer(s) need to exchange the Graceful Restart Capability in their OPEN messages, at the time of session establishment. If both peers do not exchange the Graceful Restart Capability, the session is not graceful restart capable.

If the BGP session is lost during the RP switchover, the NSF-aware BGP peer marks all routes associated with the NSF-capable device as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the NSF-capable device reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable device as having restarted.

At this point, the routing information is exchanged between two BGP peers. Once this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table. Following that, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful-restart capability in an OPEN message; but will establish a BGP session with the NSF-capable device. This function allows interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be graceful restart capable.

Note BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, devices must have the Graceful Restart Capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable device discovers that a particular BGP neighbor does not have Graceful Restart Capability, it will not establish an NSF-capable session with that neighbor. All other neighbors that have Graceful Restart Capability will continue to have NSF-capable sessions with this NSF-capable networking device.

EIGRP Operation

Enhanced Interior Gateway Routing Protocol (EIGRP) NSF capabilities are exchanged by EIGRP peers in hello packets. The NSF-capable device notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet. When an NSF-aware device receives notification from an NSF-capable neighbor that an NSF-restart operation is in progress, the NSF-capable and NSF-aware devices immediately exchange their topology tables. The NSF-aware device sends an end-of-table update packet when the transmission of its topology table is complete. The NSF-aware device then performs the following actions to assist the NSF-capable device:

- The EIGRP hello hold timer is expired to reduce the time interval set for hello packet generation and transmission. This allows the NSF-aware device to reply to the NSF-capable device more quickly reducing the amount of time required for the NSF-capable device to rediscover neighbors and rebuild the topology table.
- The route-hold timer is started. This timer is used to set the period of time that the NSF-aware device will hold known routes for the NSF-capable neighbor. This timer is configured with the command. The default time period is 240 seconds.
- In the peer list, the NSF-aware device notes that the NSF-capable neighbor is restarting, maintains adjacency, and holds known routes for the NSF-capable neighbor until the neighbor signals that it is ready for the NSF-aware device to send its topology table, or the route-hold timer expires. If the route-hold timer expires on the NSF-aware device, the NSF-aware device discards held routes and treats the NSF-capable device as a new device joining the network and reestablishes adjacency accordingly.
- The NSF-aware device continues to send queries to the NSF-capable device which is still in the process of converging after a switchover, effectively extending the time before a stuck-in-active condition can occur.

When the switchover operation is complete, the NSF-capable device notifies its neighbors that it has reconverged and has received all of their topology tables by sending an end-of-table update packet to assisting devices. The NSF-capable device then returns to normal operation. The NSF-aware device will look for alternate paths (go active) for any routes that are not refreshed by the NSF-capable (restarting device). The NSF-aware device will then return to normal operation. If all paths are refreshed by the NSF-capable device, the NSF-aware device will immediately return to normal operation.

Note NSF-aware devices are completely compatible with non-NSF aware or -capable neighbors in an EIGRP network. A non-NSF aware neighbor will ignore NSF capabilities and reset adjacencies and otherwise maintain the peering sessions normally.

OSPF Operation

When an OSPF NSF-capable device performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its OSPF neighbors:

- Relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship.
- Reacquire the contents of the link state database for the network.

As quickly as possible after a supervisor engine switchover, the NSF-capable device sends an OSPF NSF signal to neighboring NSF-aware devices. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this device should not be reset. As the NSF-capable device receives signals from other devices on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable device begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. Once this exchange is complete, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. The OSPF protocols are then fully converged.

Note OSPF support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable device discovers that it has non-NSF -aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware devices continue to provide NSF capabilities.

How to Configure Cisco Nonstop Forwarding with Stateful Switchover

Configuring Stateful Switchover

You must configure SSO in order to use NSF with any supported protocol.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show redundancy states** **Example:** `Device# show redundancy states`	Displays the operating redundancy mode.
Step 3	**redundancy** **Example:**	Enters redundancy configuration mode.

	Command or Action	Purpose
	`Device(config)# redundancy`	
Step 4	**mode sso** **Example:** `Device(config-red)# mode sso`	Configures stateful switchover. • When this command is entered, the standby is reloaded and begins to work in SSO mode.

Step 5	**end**
	Example:
	`Device(config-red)# end`
		Exits redundancy configuration mode and returns to privileged EXEC mode.

Step 6	**show redundancy states** **Example:** `Device# show redundancy states`	Displays the operating redundancy mode.
Step 7	**debug redundancy status** **Example:** `Device# debug redundancy status`	Enables the debugging of redundancy status events.

Configuration Examples for Nonstop Forwarding with Stateful Switchover

Example: Configuring Stateful Switchover

This example shows how to configure the system for SSO and displays the redundancy state:

```
Device(config)# redundancy
Device(config-red)# mode sso
Device(config-red)# end
Device#
```

Verifying Cisco Express Forwarding with Cisco Nonstop Forwarding

Procedure

show cef state

Displays the state of Cisco Express Forwarding on a networking device.

Example:

```
Device# show cef state

CEF Status:
```

```
RP instance
common CEF enabled
IPv4 CEF Status:
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
universal per-destination load sharing algorithm, id DEA83012
IPv6 CEF Status:
CEF disabled/not running
dCEF disabled/not running
universal per-destination load sharing algorithm, id DEA83012
RRP state:
I am standby RRP: no
RF Peer Presence: yes
RF PeerComm reached: yes
RF Progression blocked: never
Redundancy mode: rpr(1)
CEF NSF sync: disabled/not running
CEF ISSU Status:
FIBHWIDB broker
No slots are ISSU capable.
FIBIDB broker
No slots are ISSU capable.
FIBHWIDB Subblock broker
No slots are ISSU capable.
FIBIDB Subblock broker
No slots are ISSU capable.
Adjacency update
No slots are ISSU capable.
IPv4 table broker
No slots are ISSU capable.
CEF push
No slots are ISSU capable.
```

Additional References for Cisco Nonstop Forwarding with Stateful Switchover

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the section of the

Feature History and Information for Cisco Nonstop Forwarding with Stateful Switchover

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required

Table 5: Feature Information for Nonstop Forwarding with Stateful Switchover

CHAPTER 3

Restrictions for Graceful Insertion and Removal

GIR is supported for layer two interface shutdown and ISIS routing protocol, HSRP, VRRPv3. This is configured either by creating customized templates or without a template.

Information About Graceful Insertion and Removal

Overview

Graceful Insertion and Removal (GIR) isolates a switch from the network in order to perform debugging or an upgrade. The switch can be put into maintenance mode using the **start maintenance** command. When switch maintenance is complete, the switch will return to normal mode on either reaching the configured maintenance timeout, or by enabling the **stop maintenance** command.

Creating a maintenance mode template before you put the switch in maintenance mode is optional. The objective of maintenance mode for a device is to minimize traffic disruption at the time of removal from the network, as well as during the time of insertion. There are mainly three stages:

- Graceful removal of the node from network.
- Performing maintenance on the device.
- Graceful insertion into the network.

A switch can be put into maintenance mode using default template or a custom template. The default template contains all the ISIS instances, along with **shut down l2**. In the custom template, you can configure the required ISIS instances and **shutdown l2**option. On entering maintenance mode, all participating protocols are isolated, and L2 ports are shut down. When normal mode is restored, all the protocols and L2 ports are brought back up.

Snapshots are taken automatically while entering and exiting the maintenance mode. You can use the **snapshot create** *snapshot-name snapshot-description* command to capture and store snapshots for pre-selected features.

Snapshots are useful to compare the state of a switch before it went into maintenance mode and after it came back to normal mode. The snapshot process consists of three parts:

- Creating a snapshot of the states of a few preselected features on the switch and storing them onthe persistent storage media.
- Listing the snapshots taken at various time intervals and managing them.
- Comparing snapshots and showing the summary and details of each feature.

The maximum number of snapshots that may be stored on the switch is 10. You can use the **snapshot delete** *snapshot-name* command, to delete a specific snapshot from the device.

You can create multiple templates for the maintenance template or the snapshot template. But only one maintenance template and one snapshot template can be applied to the device at one time.

Snapshot templates can be created to generate specific snapshots. A new snapshot template can be created using the **snapshot-template***template-name* command. The command **snapshot-template***default-snapshot-template* can be used to specify the default snapshot template in the maintenance mode. The **snapshot create** [**template***template-name*] *snapshot-namesnapshot-description* command can be used to apply a specific template to the snapshot create feature.

Layer 2 Interface Shutdown

Layer 2 interfaces, such as ports on a switch, are shut down when the system is transitioning into maintenance mode. Layer 2 interfaces are shut down by using the **shutdown l2** (maintenance template configuration mode) command in the custom template.

Custom Template

As a network administrator, you can create a template that is applied when the system goes into maintenance mode. This allows you to isolate specific protocols. All instances that need to be isolated must be explicitly specified.

You can create multiple templates with different configurations. However, only a single template is applied to the maintenance mode CLI. Once applied, the template cannot be updated. If the template has to be updated, then you must remove it, make the changes, and then re-apply.

Within a template, protocols belonging to one class are serviced in parallel. The order of priority of the protocols is the same as that of the default template.

To configure this feature, enter the maintenance mode using the **system mode maintenance** command and enable the feature using the **template***template-name***calss** command.

For example if the custom template has the following protocols:

```
Maintenance-template foo
  router isis 100
  hsrp Et0/1 1
  hsrp Et0/1 2
  router isis 200
```

```
Maintenance-template foo class
  router isis 100
  hsrp Et0/1 1
  hsrp Et0/1 2
  router isis 200
```

In the above example, since isis belongs to CLASS_IGP, router isis 100 & router isis 200 will be serviced in parallel. Once acknowledgements are received for both these protocols belonging to IGP class, FHRP_CLASS clients, hsrp Et0/1 and hsrp Et0/1 2 will be serviced in parallel.

When the template-class feature is configured, the protocols follow an order based on the class they belong to when entering maintenance mode. The protocols follow the opposite order when returning to normal mode.

System Mode Maintenance Counters

GIR has counters to track the following events:

- Number of times the switch went into maintenance.
- Ack statistics per client.
- Nack statistics per client
- Number of times a particular client did not acknowledge.
- Number of times switch over happened during GIR. GIR infra will rsync this counter to track multiple switchovers.
- Number of times the failsafe timer expired.
- Number of times system got out of maintenance on a timeout expiry.

Enter the **show system mode maintenance counters** command in privileged EXEC mode, to display the counters that are being tracked by the feature.

Enter the **clear system mode maintenance counters** command in privileged EXEC mode, to clear the counters supported by the feature.

The client-ack timeout value can be configured using the **failsafe***failsafe-timeout-value* command. Failsafe time is the time that the GIR engine allows a client to transition. Each client sends a notification to the GIR engine about its transition. If it takes more than the failsafe time to transition, it is assumed to have transitioned. The failsafe timer can be configured between 5 - 180 minutes, with a default of 30 minutes.

How to Configure Graceful Insertion and Removal

Creating maintenance template

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters the global configuration mode.

	Command or Action	Purpose
	`# config t`	
Step 3	**maintenance-template** *template_name* **Example:** `(config)# maintenance-template gir1`	Creates a template with the specified name. For example, see Examples: Creating customer profile.

Step 4	**router** *routing_protocol instance_id* | **shutdown l2**
	Example:
	`(config-maintenance-templ)# router isis 1`
	`(config-maintenance-templ)# shutdown l2`
		Creates instances that should be isolated under this template.
		• **router:** Configures routing protocols and associated instance id.
		• **shutdown l2:** Shuts down layer 2 interfaces.

Configuring System Mode Maintenance

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**system mode maintenance** **Example:** `Device(config)# system mode maintenance`	Enters system mode maintenance configuration mode. Different sub commands to create maintenance mode parameters are configured in this mode.
Step 4	**timeout** *timeout-value* | **template** *template-name* | **failsafe** *failsafe-timeout-value* | **on-reload reset-reason maintenance**	Configures maintenance mode parameters. • **timeout:** Configures maintenance mode timeout period in minutes, after which the system automatically returns to normal mode. The default timeout value is never. • **template:** Configures maintenance mode using the specified template. • **failsafe:**Configures client-ack timeout value. If the system is going into maintenance mode, it will continue to reach

	Command or Action	Purpose
		maintenance.If the system is exiting from maintenance mode, then it will reach normal mode. • **on-reload reset-reason maintenance:**Configures the system such that when the system is reloaded it enters the maintenance mode. If it is not configured the system enters the normal mode when it is reloaded.

Starting and Stopping Maintenance Mode

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**start maintenance** **Example:** `Device# start maintenance`	Puts the system into maintenance mode.
Step 3	**stop maintenance** **Example:** `Device# stop maintenance`	Puts the system back into normal mode.

Configuration Examples for Graceful Removal and Insertion

The following examples show the sequence followed to enable GIR during a maintanence window.

Example: Configuring maintenance template

Any protocol that is supported by GIR can be configure in the maintenance template.This example shows how to configure a maintenance template t1 with an ISIS routing protocol instance.

```
Device# config terminal
Device (config)# maintenance-template t1
Device (config-maintenance-templ)# router isis 1
```

This example shows how to configure a maintenance template t1 with shutdown l2.

```
Device# config terminal
Device (config)# maintenance-template t1
Device (config-maintenance-templ)# shutdown l2
```

Example: Configuring System Mode Maintenance

This example shows how to create a maintenance template and configure the maintenance mode parameters.

```
Device# configure terminal
Device(config)# system mode maintenance
Device(config-maintenance)# timeout 20
Device(config-maintenance)# failsafe 30
Device(config-maintenance)# on-reload reset-reason maintenance
Device(config-maintenance)# template t1
Device(config-maintenance)# exit
```

Example: Starting and Stopping the Maintenance Mode

This example shows how to put the system into maintenance mode.

```
Device# start maintenance
```

After the activity is completed, the system can be put out of maintenance mode.

This example shows how to put the system out of maintenance mode.

```
Device# stop maintenance
```

Example: Displaying System Mode Settings

This example shows how to display system mode settings using different options.

```
Device# show system mode
     System Mode: Normal


Device# show system mode maintenance
     System Mode: Normal
     Current Maintenance Parameters:
     Maintenance Duration: 15(mins)
     Failsafe Timeout: 30(mins)
     Maintenance Template: t1
     Reload in Maintenance: False


Device# show system mode maintenance clients
     System Mode: Normal
     Maintenance Clients:
     CLASS-EGP
     CLASS-IGP
     router isis 1: Transition None
     CLASS-MCAST
     CLASS-L2


Device# show system mode maintenance template default
     System Mode: Normal
     default maintenance-template details:
     router isis 1
     router isis 2

Device# show system mode maintenance template t1
     System Mode: Normal
     Maintenance Template t1 details:
     router isis 1
```

Monitoring Graceful Insertion and Removal

Use the following commands to check the status of or display statistics generated by the GIR feature:

Table 6: Privileged EXEC Commands

Command	Purpose
show system mode [maintenance [clients | template *template-name*]]	Displays information about system mode.
show system snapshots [dump *<snapshot-file-name>]*	Displays all the snapshots present on the device.
show system snapshots [dump *<snapshot-file-name>]***xml**	Displays all the snapshots present on the device in XML format.
show system snapshots compare *snapshot-name1 snapshot-name2*	Displays differences between snapshots taken before entering maintenance mode and after exiting from the maintenance mode.

Table 7: Global Configuration Commands for Troubleshooting

Command	Purpose
debug system mode maintenance	Displays information to help troubleshoot the GIR feature.

Additional References for Graceful Insertion and Removal

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the section of the

Feature History and Information for Graceful Insertion and Removal

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 8: Feature History and Information for Graceful Insertion and Removal

CHAPTER 4

Configuring 1:1 Redundancy

Prerequisites for 1:1 Redundancy

- All the switches in the stack must be running the same license level as the active switch. For information about license levels, see the *System Management Configuration Guide* of the required release.
- All the switches in the stack must be running compatible software versions.

Information About 1:1 Redundancy

1:1 redundancy is used to assign active and standby roles to specific switches in the stack. This overrides the traditional N+1 role selection algorithm, where any switch in the stack can be active or standby. In 1:1 redundancy, the stack manager determines the active and standby role for a specific switch, based on the flash ROMMON variable. The algorithm assigns one switch as active, another switch as standby, designating all remaining switches in the stack as members. When an active switch reboots it becomes standby and the existing standby switch becomes the new active. The existing member switches remain in the same state.

How to Configure 1:1 Redundancy

Enabling 1:1 Redundancy Stack Mode

Follow these steps to enable the 1:1 redundancy stack mode, and set a switch as the active switch in a stack, or as the standby:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**switch** *switch-number* **role**{**active** | **standby**}
	Example:
	`Device# switch 1 role active`
		Changes stack mode to 1:1 mode and designates the switch as active or standby.

Disabling 1:1 Redundancy Stack Mode

On a switch where 1:1 redundancy is enabled, follow these steps to disable the feature. This changes the stack mode to N+1:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**switch clear stack-mode** **Example:** `Device# switch clear stack-mode`	Changes stack mode to the N+1 mode and removes active and standby assignments.

Configuration Examples for 1:1 Redundancy

Example: Enabling 1:1 Redundancy Stack Mode

You can use the **switch switch-number role** command to set the active and standby switch in 1:1 stack mode. The stack operates in the 1:1 stack mode with the specified active or standby after reboot. In the following example, switch 1 is assigned the active role, and switch 2 is assigned the standby role.

```
Device# switch 1 role active
WARNING: Changing the switch role may result in redundancy mode being configured to 1+1
mode for this stack. If the configured Active or Standby switch numbers do not boot up,
then the stack will not be able to boot. Do you want to continue?[y/n]? [yes]: yes

Device# switch 2 role standby
WARNING: Changing the switch role may result in redundancy mode being configured to 1+1
```

```
mode for this stack. If the configured Active or Standby switch numbers do not boot up,
then the stack will not be able to boot. Do you want to continue?[y/n]? [yes]: yes
```

Example: Disabling 1:1 Redundancy

You can use the **switch clear stack-mode** command to remove 1:1 stack mode, and change it back to N+1 stack mode.

```
Device# switch clear stack-mode
WARNING: Clearing the chassis HA configuration will resultin the chassis coming up in Stand
 Alone mode after reboot.The HA configuration will remain the same on other chassis. Do you
 wish to continue? [y/n]? [yes]:
```

Verifying the Stack Mode

To verify the current stack mode on a switch, enter the **show switch stack-mode** command in priviledged EXEC mode. The output displays detailed status of the currently running stack mode.

```
Device# show switch stack-mode
Switch   Role     Mac Address    Version   Mode     Configured   State
---------------------------------------------------------------
1       Member 3c5e.c357.c880             1+1'     Active'     Ready
*2      Active 547c.69de.cd00    V05      1+1'     Standby'    Ready
3       Member 547c.6965.cf80    V05      1+1'     Member'     Ready
```

The `Mode` field indicates the current stack mode

The `Configured` field refers to the switch state expected after a reboot.

Single quotation marks (') indicate that the stack mode has been changed.

Additional References for 1:1 Redundancy

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Stacking and High Availability Commands* section of the Command Reference for the release.

Feature History for 1:1 Redundancy

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.