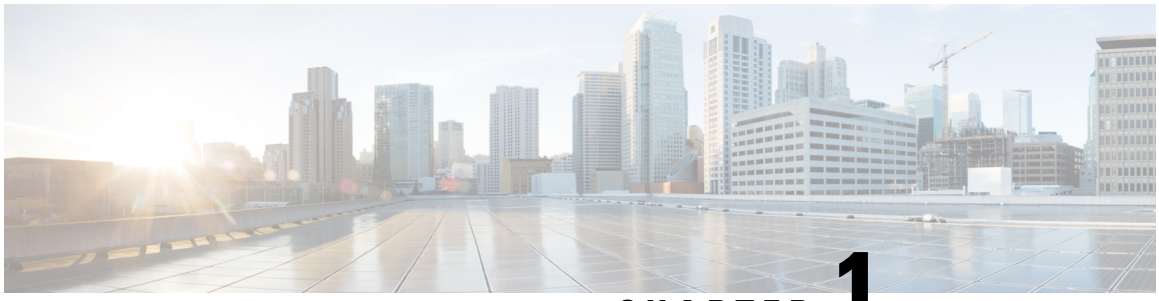
Stacking and High Availability Configuration Guide, Cisco IOS XE Cupertino 17.9.x (Catalyst 9300 Switches)

First Published: 2022-08-01

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

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.

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

The following are restrictions for your switch stack configuration:

- Only homogenous stacking is supported, that is, a stack of Cisco Catalyst 9300 Series Switches stack with only Cisco Catalyst 9300 Series Switches as stack members.

  Cisco Catalyst 9300L Series Switches stack with only Cisco Catalyst 9300L Series Switches as stack members.

  C9300-24UB, C9300-24UXB, and C9300-48UB switches can only be stacked with each other.

- During a switchover, when the standby device syncs with the active device, the following log message is displayed on the console:

  ```
  %SM-4-BADEVENT: Event 'standby_phy_link_up' is invalid for
  the current state 'NO_NEIGHBOR': rep_lsl_rx Gix/x/x -Traceback=
  ```

  Ignore this message. It does not have any functional or operational impact.

- You cannot have a switch stack containing a mix of different license levels.

Information About Switch Stacks

Switch Stack Overview

A switch stack can have up to eight stacking-capable switches connected through their StackWise-480 ports. Note that Cisco Catalyst 9300L series switches connect through their StackWise-320 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.

A switch stack always has one active switch and one standby switch. If the active switch becomes unavailable, the standby switch assumes the role of the active switch, and continues to the keep the stack operational.

The active switch controls the operation of the switch stack, and is the single point of stack-wide management.

From the active switch, you configure:

- System-level (global) features that apply to all stack members
- Interface-level features for each stack member

The active switch contains the saved and running configuration files for the switch stack. The configuration files include the system-level settings for the switch stack and the interface-level settings for each stack member. Each stack member has a current copy of these files for back-up purposes.

Switch Stack Membership

A standalone is a stack with one stack member that also operates as the active switch. You can connect one standalone to another to create a stack containing two stack members, with one of them as the active switch. You can connect standalone to an existing stack to increase the stack membership.

Hello messages are sent and received by all stack members.

- If a stack member does not respond, that member is removed from the stack.
- If the standby does not respond, a new standby is elected.
- If the active does not respond, the standby becomes the active .

In addition, keepalive messages are sent and received between the active and standby es.

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

Changes to Switch Stack Membership

If you replace a stack member with an identical model, the new switch functions with exactly the same configuration as the replaced switch, assuming that the new switch (referred to as the provisioned switch) is using the same member number as the replaced switch.

The operation of the switch stack continues uninterrupted during membership changes unless you remove the active switch or you add powered-on standalone switches or switch stacks.

- Adding powered-on switches (merging) causes all switches to reload and elect a new active switch from among themselves. The newly elected active switch retains its role and configuration. All other switches retain their stack member numbers and use the stack configuration of the newly elected active switch.
- Removing powered-on stack members causes the switch stack to divide (partition) into two or more switch stacks, each with the same configuration. This can cause:
  - An IP address conflict in your network. If you want the switch stacks to remain separate, change the IP address or addresses of the newly created switch stacks.
  - A MAC address conflict between two members in the stack. You can use the **stack-mac update force** command to resolve the conflict.

If a newly created switch stack does not have an active switch or standby switch, the switch stack will reload and elect a new active switch.

Note Make sure that you power off the switches that you add to or remove from the switch stack.

After adding or removing stack members, make sure that the switch stack is operating at full bandwidth. Press the Mode button on a stack member until the Stack mode LED is on. The last two right port LEDs on all switches in the stack should be green. Depending on the switch model, the last two right ports are 10-Gigabit Ethernet ports or small form-factor pluggable (SFP) module ports (10/100/1000 ports). If one or both of these LEDs are not green on any of the switches, the stack is not operating at full bandwidth.

If you remove powered-on members but do not want to partition the stack:

- Power off the switches in the newly created switch stacks.
- Reconnect them to the original switch stack through their stack ports.
- Power on the switches.

For cabling and power considerations that affect switch stacks, see the *Cisco Catalyst 9300 Series Switches Hardware Installation Guide* .

Stack Member Numbers

The stack member number (1 to 8) identifies each member in the switch stack. The member number also determines the interface-level configuration that a stack member uses. You can display the stack member number by using the **show switch** EXEC command.

A new, out-of-the-box switch (one that has not joined a switch stack or has not been manually assigned a stack member number) ships with a default stack member number of 1. When it joins a switch stack, its default stack member number changes to the lowest available member number in the stack.

Stack members in the same switch stack cannot have the same stack member number. Every stack member, including a standalone switch, retains its member number until you manually change the number or unless the number is already being used by another member in the stack.

- If you manually change the stack member number by using the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command, the new number goes into effect after that stack member resets (or after you use the **reload slot** *stack-member-number* privileged EXEC command) and only if that number is not already assigned to any other members in the stack. Another way to change the stack member number is by changing the SWITCH_NUMBER environment variable.

  If the number is being used by another member in the stack, the switch selects the lowest available number in the stack.

  If you manually change the number of a stack member and no interface-level configuration is associated with that new member number, that stack member resets to its default configuration.

  You cannot use the **switch** *current-stack-member-number* **renumber** *new-stack-member-number* EXEC command on a provisioned switch. If you do, the command is rejected.

- If you move a stack member to a different switch stack, the stack member retains its number only if the number is not being used by another member in the stack. If it is being used, the switch selects the lowest available number in the stack.

- If you merge switch stacks, the switch that join the switch stack of a new active switch select the lowest available numbers in the stack.

As described in the hardware installation guide, you can use the switch port LEDs in Stack mode to visually determine the stack member number of each stack member.

You can enter the Stack mode on any of these switches by pressing the mode button. Based on the switch number configured on each switch, the corresponding port LED will be blinking green. For instance, if the switch number configured on a particular switch is three, then the port LED-3 will be blinking green when the mode button is set to stack.

Stack Member Priority Values

A higher priority value for a stack member increases the probability of it being elected active switch and retaining its stack member number. The priority value can be 1 to 15. The default priority value is 1. You can display the stack member priority value by using the **show switch** EXEC command.

Note We recommend assigning the highest priority value to the that you prefer to be the active switch. This ensures that the is reelected as the active switch if a reelection occurs.

To change the priority value for a stack member, use the **switch** *stack-member-number* **priority** *new priority-value* EXEC command. For more information, see the “Setting the Stack Member Priority Value” section.

The new priority value takes effect immediately but does not affect the current active switch. The new priority value helps determine which stack member is elected as the new active switch when the current active switch or the switch stack resets.

Switch Stack Bridge ID and MAC Address

A switch stack is identified in the network by its *bridge ID* and, if it is operating as a Layer 3 device, its router MAC address. The bridge ID and router MAC address are determined by the MAC address of the active switch.

If the active switch changes, the MAC address of the new active switch determines the new bridge ID and router MAC address.

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

Persistent MAC Address on the Switch Stack

You can use the persistent MAC address feature to set a time delay before the stack MAC address changes. During this time period, if the previous active switch rejoins the stack, the stack continues to use its MAC address as the stack MAC address, even if the switch is now a stack member and not an active switch. If the previous active switch does not rejoin the stack during this period, the switch stack takes the MAC address of the new active switch as the stack MAC address. By default, the stack MAC address will be the MAC address of the first active switch, even if a new active switch takes over.

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

Active and Standby Switch Election and Reelection

All stack members are eligible to be the active switch or the standby switch. If the active switch becomes unavailable, the standby switch becomes the active switch.

An active switch retains its role unless one of these events occurs:

- The switch stack is reset.
- The active switch is removed from the switch stack.
- The active switch is reset or powered off.
- The active switch fails.
- The switch stack membership is increased by adding powered-on standalone switches or switch stacks.

The active switch is elected or reelected based on one of these factors and in the order listed:

1. The switch that is currently the active switch.
2. The switch with the highest stack member priority value.

Note We recommend assigning the highest priority value to the switch that you prefer to be the active switch. This ensures that the switch is reelected as active switch if a reelection occurs.

3. The switch with the shortest start-up time.

4. The switch with the lowest MAC address.

Note The factors for electing or reelecting a new standby switch are same as those for the active switch election or reelection, and are applied to all participating switches except the active switch.

After election, the new active switch becomes available after a few seconds. In the meantime, the switch stack uses the forwarding tables in memory to minimize network disruption. The physical interfaces on the other available stack members are not affected during a new active switch election and reset.

When the previous active switch becomes available, it *does not* resume its role as the active switch.

If you power on or reset an entire switch stack, some stack members *might not* participate in the active switch election. Stack members that are powered on within the same 2-minute timeframe participate in the active switch election and have a chance to become the active switch. Stack members that are powered on after the 120-second timeframe do not participate in this initial election and become stack members. For powering considerations that affect active-switch elections, see the switch hardware installation guide.

As described in the hardware installation guide, you can use the ACTV LED on the switch to see if the switch is the active switch.

Switch Stack Configuration Files

The active switch has the saved and running configuration file for the switch stack. The standby switch automatically receives the synchronized running configuration file. Stack members receive synchronized copies when the running configuration file is saved into the startup configuration file. If the active switch becomes unavailable, the standby switch takes over with the current running configuration.

The configuration files record these settings:

- System-level (global) configuration settings such as IP, STP, VLAN, and SNMP settings that apply to all stack members
- Stack member interface-specific configuration settings that are specific for each stack member

Note The interface-specific settings of the active switch are saved if the active switch is replaced without saving the running configuration to the startup configuration.

A new, out-of-box switch joining a switch stack uses the system-level settings of that switch stack. If a switch is moved to a different switch stack before it is powered on, that switch loses its saved configuration file and uses the system-level configuration of the new switch stack. If the switch is powered on as a standalone switch before it joins the new switch stack, the stack will reload. When the stack reloads, the new switch may become the active switch, retain its configuration and overwrite the configuration files of the other stack members.

The interface-specific configuration of each stack member is associated with the stack member number. Stack members retain their numbers unless they are manually changed or they are already used by another member in the same switch stack. If the stack member number changes, the new number goes into effect after that stack member resets.

- If an interface-specific configuration does not exist for that member number, the stack member uses its default interface-specific configuration.

- If an interface-specific configuration exists for that member number, the stack member uses the interface-specific configuration associated with that member number.

If you replace a failed member with an identical model, the replacement member automatically uses the same interface-specific configuration as the failed switch. You do not need to reconfigure the interface settings. The replacement switch (referred to as the provisioned switch) must have the same stack member number as the failed switch.

You back up and restore the stack configuration in the same way as you would for a standalone switch configuration.

Offline Configuration to Provision a Stack Member

You can use the offline configuration feature to *provision* (to supply a configuration to) a new switch before it joins the switch stack. You can configure the stack member number, the switch type, and the interfaces associated with a switch that is not currently part of the stack. The configuration that you create on the switch stack is called the *provisioned configuration*. The switch that is added to the switch stack and that receives this configuration is called the *provisioned switch*.

You manually create the provisioned configuration through the **switch** *stack-member-number* **provision** *type* global configuration command. You must change the *stack-member-number* on the provisioned switch before you add it to the stack, and it must match the stack member number that you created for the new switch on the switch stack. The switch type in the provisioned configuration must match the switch type of the newly added switch. The provisioned configuration is automatically created when a switch is added to a switch stack and when no provisioned configuration exists.

When you configure the interfaces associated with a provisioned switch, the switch stack accepts the configuration, and the information appears in the running configuration. However, as the switch is not active, any configuration on the interface is not operational and the interface associated with the provisioned switch does not appear in the display of the specific feature. For example, VLAN configuration information associated with a provisioned switch does not appear in the **show vlan** user EXEC command output on the switch stack.

The switch stack retains the provisioned configuration in the running configuration whether or not the provisioned switch is part of the stack. You can save the provisioned configuration to the startup configuration file by entering the **copy running-config startup-config** privileged EXEC command. The startup configuration file ensures that the switch stack can reload and can use the saved information whether or not the provisioned switch is part of the switch stack.

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 active switch. You can use the CLI, SNMP, and any of the supported network management applications. You cannot manage stack members on an individual basis.

Note Use SNMP to manage network features across the stack that are defined by supported MIBs. The switch does not support MIBs to manage stacking-specific features such as stack membership and election.

How to Configure a Switch Stack

Temporarily Disabling a Stack Port

If a stack port is flapping and causing instability in the stack ring, to disable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **disable** privileged EXEC command. To reenable the port, enter the **switch** *stack-member-number* **stack port** *port-number* **enable** command.

Note Be careful when using the **switch** *stack-member-number* **stack port** *port-number* **disable** command. When you disable the stack port, the stack operates at half bandwidth.

A stack is in the full-ring state when all members are connected through the stack ports and are in the ready state.

The stack is in the partial-ring state when the following occurs:

- All members are connected through their stack ports but some are not in the ready state.
- Some members are not connected through the stack ports.

Procedure

	Command or Action	Purpose
Step 1	**switch** *stack-member-number* **stack port** *port-number* **disable** **Example:** `# switch 2 stack port 1 disable`	Disables the specified stack port.
Step 2	**switch** *stack-member-number* **stack port** *port-number* **enable** **Example:** `# switch 2 stack port 1 enable`	Reenables the stack port.

When you disable a stack port and the stack is in the full-ring state, you can disable only one stack port. This message appears:

```
Enabling/disabling a stack port may cause undesired stack changes. Continue?[confirm]
```

When you disable a stack port and the stack is in the partial-ring state, you cannot disable the port. This message appears:

```
Disabling stack port not allowed with current stack configuration.
```

Reenabling a Stack Port While Another Member Starts

Stack Port 1 on Switch 1 is connected to Port 2 on Switch 4. If Port 1 is flapping, you can disable Port 1 with the **switch 1 stack port 1 disable** privileged EXEC command. While Port 1 on Switch 1 is disabled and Switch 1 is still powered on, follow these steps to reenable a stack port:

Procedure

Step 1 Disconnect the stack cable between Port 1 on Switch 1 and Port 2 on Switch 4.

Step 2 Remove Switch 4 from the stack.

Step 3 Add a switch to replace Switch 4 and assign it switch-number 4.

Step 4 Reconnect the cable between Port 1 on Switch 1 and Port 2 on Switch 4 (the replacement switch).

Step 5 Reenable the link between the switches. Enter the **switch 1 stack port 1 enable** privileged EXEC command to enable Port 1 on Switch 1.

Step 6 Power on Switch 4.

Caution Powering on Switch 4 before enabling the Port 1 on Switch 1 might cause one of the switches to reload.

If Switch 4 is powered on first, you might need to enter the **switch 1 stack port 1 enable** and the **switch 4 stack port 2 enable** privileged EXEC commands to bring up the link.

Monitoring the Device Stack

Table 1: Commands for Displaying Stack Information

Command	Description
show switch	Displays summary information about the stack, including the status of provisioned switches and switches in version-mismatch mode.
show switch *stack-member-number*	Displays information about a specific member.
show module	Displays summary informaton about the stack.
show switch detail	Displays detailed information about the stack.
show switch neighbors	Displays the stack neighbors.

Command	Description
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 switch stack-ports [**detail**]	Displays the stack link status and information for each stack member. Use the **detail** keyword to display the stack interface status, errors, drops, packet transmission and bandwidth details.
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.
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* EXEC 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.

Scenario		Result
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* EXEC 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.	The standby switch becomes the new active switch. All other member switches in the stack remain as member switches and do not reboot.
Add eight member switches	**1.** Through their StackWise ports, connect eight devices. **2.** Power on all devices.	Two devices become active switches. One active switch has eight 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:

```
Device(config)# stack-mac persistent timer 7
  WARNING: The stack continues to use the base MAC of the old active
  WARNING: as the stack-MAC after a active 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.
  Device(config)# end
  Device# 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       Active 0016.4727.a900     1      P2B     Ready
```

Provisioning a New Member for a Switch Stack: Example

The **show running-config** command output shows the interfaces associated with the provisioned switch:

```
(config)# switch 2 provision switch_PID
  (config)# end
  # show running-config | include switch 2
  !
  interface GigabitEthernet2/0/1
  !
  interface GigabitEthernet2/0/2
  !
  interface GigabitEthernet2/0/3
  <output truncated>
```

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. • 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. When there is no cable connected to the stack port, the value displayed is *no cable* along with the cable length value.
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.

show switch stack-ports detail Command Output: Example

The following is a sample output of the command for a working stack:

```
Device# show switch stack-ports detail
1/1 is DOWN Loopback No
Cable Length 50cm     Neighbor NONE
Link Ok Yes Sync Ok Yes Link Active No
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     752 bytes input
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 667
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  7 bytes/sec
Five minute output rate 0 bytes/sec
     54332 bytes input
     1120 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 30 bytes/sec
     146390 bytes input
     217587 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is DOWN Loopback No
Cable Length 50cm     Neighbor NONE
Link Ok Yes Sync Ok Yes Link Active No
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     1208 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     41245 bytes input
```

```
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  10 bytes/sec
Five minute output rate 0 bytes/sec
     60412 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
```

Table 4: show switch stack-ports detail Command Output

Field	Description
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 *Unknown*. 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.

Field	Description
Five minute input rate	The average rate (calculated over a five minute period) at which packets are received, measured in packets/sec.
Five minute output rate	The average rate (calculated over a five minute period) at which packets are transmitted, measured in packets/sec.
CRC Errors	Different types of Cyclic Redundancy Check (CRC) errors that are seen on a stack interface: • **Data CRC**: Stack interface data CRC error • **Ringword CRC**: Stack interface ring word CRC error • **InvRingWord**: Stack interface invalid ring word error • **PcsCodeWord**: Stack interface Physical Coding Sublayer (PCS) error These errors normally occur when a stack interface state changes due to a switchover or a switch reload. You can ignore such errors. But when these error counters increase significantly or when they increase continuously over a period of time, check the stack cable for issues.

he following is a sample output when the stack port flaps:

```
Device# show switch stack-ports detail
1/1 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 4
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     320 bytes input
     80 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 770
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  5 bytes/sec
Five minute output rate 1 bytes/sec
     2949 bytes input
     320 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     49375 bytes input
```

```
     160 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 2
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     1824 bytes input
     160 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  372 bytes/sec
Five minute output rate 7 bytes/sec
     111876 bytes input
     4613 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 2
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     80 bytes input
     0 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
```

The following is a sample output when a switch reloads:

```
Device#show switch stack-ports detail
1/1 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 5
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     2032 bytes input
     320 bytes output
CRC Errors
            Data CRC 184
        Ringword CRC 187
        InvRingWord  120
         PcsCodeWord 112
1/2 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
```

```
Changes to LinkOK 1
Five minute input rate  2 bytes/sec
Five minute output rate 0 bytes/sec
     24164 bytes input
     800 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/1 is OK Loopback No
Cable Length 50cm     Neighbor 3
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     3024 bytes input
     240 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
2/2 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  7 bytes/sec
Five minute output rate 0 bytes/sec
     9148 bytes input
     480 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/1 is OK Loopback No
Cable Length 50cm     Neighbor 1
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 1
Five minute input rate  0 bytes/sec
Five minute output rate 15 bytes/sec
     1509354 bytes input
     27853 bytes output
CRC Errors
            Data CRC 0
        Ringword CRC 0
        InvRingWord  0
         PcsCodeWord 0
3/2 is OK Loopback No
Cable Length 50cm     Neighbor 2
Link Ok Yes Sync Ok Yes Link Active Yes
Changes to LinkOK 3
Five minute input rate  0 bytes/sec
Five minute output rate 0 bytes/sec
     240 bytes input
     160 bytes output
CRC Errors
            Data CRC 118
        Ringword CRC 74
        InvRingWord  125
         PcsCodeWord 373
```

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.	https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/hardware/install/b_c9300_hig.html *Cisco Catalyst 9300 Series Switches Hardware Installation Guide*
SGACL High Availability	"Cisco TrustSec SGACL High Availability" module of the *Cisco TrustSec Switch Configuration Guide*

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

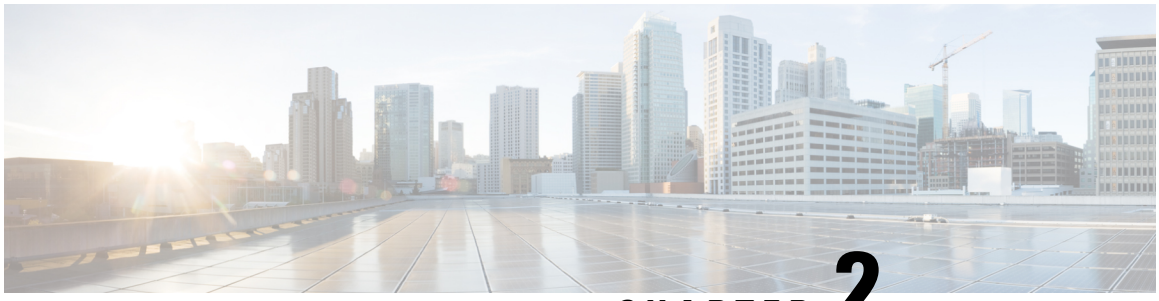
Feature History for Switch Stacks

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.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	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.
Cisco IOS XE Amsterdam 17.3.1	Switch Stack	A new command **show switch stack-ports detail** was introduced to display detailed information on the stack link of each stack member.

Use the 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



CHAPTER 2

Configuring Nonstop Forwarding with Stateful Switchover

Cisco nonstop forwarding (NSF) works with the stateful switchover (SSO) feature. NSF works with SSO to minimize the amount of time a network is unavailable to users following a switchover. The main objective of NSF SSO is to continue forwarding IP packets following a Route Processor (RP) switchover.

Prerequisites for Cisco Nonstop Forwarding with Stateful Switchover

- Cisco NSF must be configured on a networking device that has been configured for 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.

SSO Operation

When a standby device runs in SSO mode, the standby device starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration on the active device. It subsequently maintains the state of the protocols, and all changes in hardware and software states for features that support SSO are kept in synchronization. Consequently, it offers minimum interruption to Layer 2 sessions in a redundant active device configuration.

If the active device fails, the standby device becomes the active device. This new active device uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding is delayed until routing tables are repopulated in the newly active device.

Note The routing tables require around 80 seconds for repopulation. You can use the **show ip bgp** *ip-address* command, in privileged EXEC mode, to check whether the routing tables are repopulated or not.

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 device is configured (with the **graceful-restart** command) for BGP, OSPF, or EIGRP routing protocols, routing updates are automatically sent during the active device 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 device 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 device synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby device . Upon switchover, the standby device initially has FIB and adjacency databases that are mirror images of those that were current on the active device . Cisco Express Forwarding keeps the forwarding engine on the standby device current with changes that are sent to it by Cisco Express Forwarding on the active device . 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 **timers nsf route-hold** 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:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**mode sso** **Example:** `Device(config-red)# mode sso`	Configures stateful switchover. • When this command is entered, the standby switch 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.

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.
```

Configuration Examples for Cisco 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#
```

The following is sample output from the **show redundancy states** command:

```
show redundancy states
 my state = 13 -ACTIVE
 peer state = 8 -STANDBY HOT
 Mode = Duplex
 Unit = Primary
 Unit ID = 5
 Redundancy Mode (Operational) = sso
 Redundancy Mode (Configured) = sso
 Split Mode = Disabled
 Manual Swact = Enabled
 Communications = Up
 client count = 29
 client_notification_TMR = 30000 milliseconds
 keep_alive TMR = 9000 milliseconds
 keep_alive count = 1
 keep_alive threshold = 18
 RF debug mask = 0x0
```

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 *Stack Manager and High Availability* section of the *Command Reference (Catalyst 9300 Series Switches)*

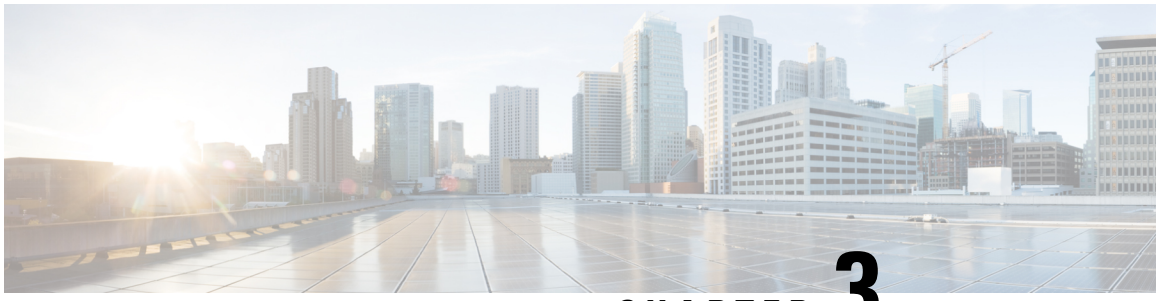
Feature History for Cisco Nonstop Forwarding with Stateful Switchover

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.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Cisco Nonstop Forwarding with Stateful Switchover	Cisco NSF works with the SSO feature. NSF works with SSO to minimize the amount of time a network is unavailable to users following a switchover. The main objective of NSF SSO is to continue forwarding IP packets following a Route Processor (RP) switchover.

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



CHAPTER 3

Configuring Graceful Insertion and Removal

Graceful Insertion and Removal (GIR) provides an alternative method to minimize network service impact caused by device maintenance. GIR leverages redundant paths in the network to smoothly remove a device under maintenance, out of service, and insert it back to service when the maintenance is complete. This module describes the how to configure GIR.

Restrictions for Graceful Insertion and Removal

GIR is supported for layer two interface shutdown, ISIS routing protocol, HSRP, VRRPv3, BGP and OSPF routing protocol. 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 a Maintenance Template

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	**maintenance-template** *template_name* **Example:** `Device(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:
	`Device(config-maintenance-templ)# router isis 1`
	`Device(config-maintenance-templ)# shutdown l2`
	`Device(config-maintenance-templ)# router bgp AS-number`
		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.

	Command or Action	Purpose
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 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.

Monitoring Graceful Insertion and Removal

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

Table 5: 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 6: Global Configuration Commands for Troubleshooting

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

Configuration Examples for Graceful Removal and Insertion

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

Example: Configuring Maintenance Templates

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

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

This example shows how to configure a maintenance template t1 with a BGP routing protocol instance.

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

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
```

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 *Stack Manager and High Availability* section of the *Command Reference (Catalyst 9300 Series Switches)*

Feature History for Graceful Insertion and Removal

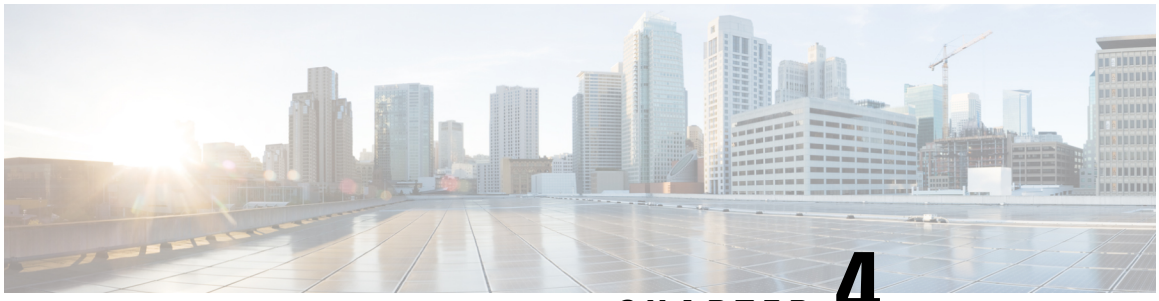
This table provides release and related information for features explained in this module.

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	Graceful Insertion and Removal	Provides an alternative method to minimize network service impact caused by device maintenance. GIR leverages redundant paths in the network to smoothly remove a device under maintenance, out of service, and insert it back to service when the maintenance is complete.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.1	Graceful Insertion and Removal (GIR) enhancements: snapshot templates	The following enhancements were introduced: • Snapshot templates can be used to generate specific snapshots. • Protocols belonging to one class within the same custom template will be serviced in parallel. • System mode maintenance counters have been added to track several events such as the number of times the switch went into maintenance.
	GIR Layer 2 protocol support for GIR Hot Standby Router Protocol (HSRP)	GIR is now supported for the HSRP protocol.
	GIR Layer 2 protocol support for GIR Virtual Router Redundancy Protocol (VRRP)	GIR is now supported for VRRPv3 protocol.
Cisco IOS XE Gibraltar 16.10.1	Graceful Insertion and Removal (GIR) Support for BGP	GIR is now supported for the BGP protocol.

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



CHAPTER 4

Configuring 1:1 Redundancy

Enabling the 1:1 redundancy stack mode allows you to assign active and standby roles to specific switches in the stack.

Prerequisites for 1:1 Redundancy

The following are 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.

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.

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]:
```

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.

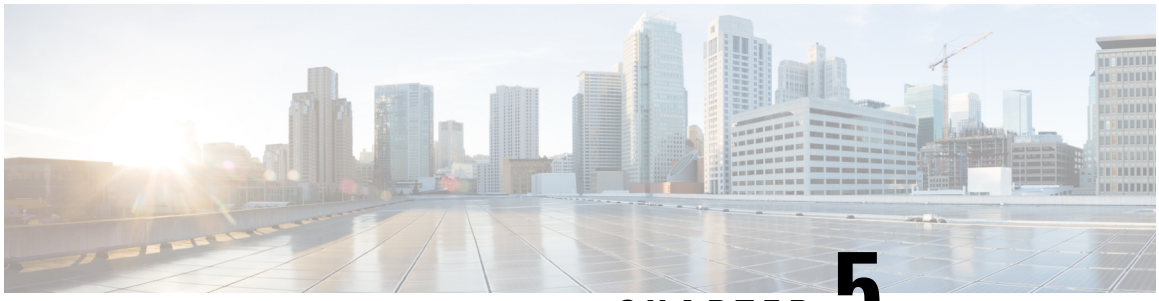
Feature History for 1:1 Redundancy

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.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.6.1	1:1 Redundancy	Enables the 1:1 redundancy stack mode and enables you to assigns active and standby roles to specific switches in the stack.

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



CHAPTER 5

Configuring High Speed Stacking

The High Speed Stacking feature allows you to configure a homogenous stack of switches to run at the speed of 1Tbps.

Restrictions for High Speed Stacking

- A high speed stack can support a maximum of 16 ASICs. The maximum switches on a stack in the high speed mode depend on the sum of total number of ASICs on the stack ring.
- This feature is supported only on the C9300X-12Y, C9300X-24Y, C9300X-48HX and C9300X-48TX models of the Cisco Catalyst 9300 Series Switches.

Information about High Speed Stacking

High Speed Stacking allows you to configure the bandwidth of Standard Interchange Format (SIF) ports to 1 Tbps.

The following topics provide information about High Speed Stacking.

Overview of High Speed Stacking

High Speed Stacking allows you to configure the bandwidth of the SIF ports on a stack to 1Tbps. High Speed stacking is applicable only when all the switches in a stack are Catalyst 9300X switches.

You can use the configuration command **switch stack-speed** [**high** | **low**] to configure the bandwidth of the SIF ports from 480Gbps (legacy mode) to 1Tbps (high speed mode). You need to reload the stack after the command is issued.

When a stack consisting of Catalyst 9300X switches boots up, a script will detect that the stack is capable of high speed. It will trigger a second reboot and change the configuration of all the switches to high speed. The script works only when all the switches have the manufacture default configuration.

The SIF port speed in the manufacture default configuration is 480 Gbps. Two stack ports on the same switch must have the same SIF port speed. If a stack cable connects to two stack ports with mismatched speeds the port link will go down. The stack will be split into sub-rings. To correct the stack-split situation, you can configure each sub-stack with the same speed by using the **switch stack-speed** [**high** | **low**] command.

Manufacture Default Stack Bootup with High Speed Stacking

When a stack boots up in the manufacture default configuration, it always boots up with the lowest speed among members. If the stack is homogenous and consists of only Catalyst 9300X switches capable of High Speed Stacking, a script running on the active switch determines that all the switches are capable of high speed. The active switch configures the stack to high speed and initiates a second round of reloads. After the second reload all the members of the stack load with high speed configured. The stack becomes a high speed ring. The homogenous stack of Catalyst 9300X switches must be a full ring stack for the automatic script to work. The script will work automatically within the first 15 minutes of uptime only.

You can use the command **show switch stack-ring speed** to display the current speed of the stack ring and what the speed will be after the next reboot.

You can use the command **show switch stack-bandwidth** to display the current stack bandwidth and what the bandwidth will be after the next reboot.

Inserting a Switch into a High Speed Stack

The following scenarios detail how to manage the insertion of a new switch into a high speed stack.

- **Insertion of a Cisco Catalyst 9300X switch into a high speed stack**: The new switch that is to be inserted into the stack must be powered off. After connecting the switch to the stack cables it can be powered on again. If the new switch has been configured for high speed it will join the stack in high speed. If the switch has not been configured for high speed it will boot up as an active island. You will have to connect to the switch using a console or telnet. Enter the configuration command **switch stack-speed high**. After entering the command the following notice will be displayed: “Stack speed does not take effect until after the reboot.”After the second reboot the switch will match the speed of the stack.
- **Insertion of a Cisco Catalyst 9300 switch into a high speed stack**: All Cisco Catalyst 9300 switches are not capable of High Speed Stacking. The switch will become an active island in the high speed autonomous stack. If you intend to have a mixed stack running at legacy speed (480 Gbps), you should configure the command **switch stack-speed low** on the high speed homongenous stack. Once you reload the stack and the new switch you will have a mixed stack operating at low speed.
- **Insertion of a Cisco Catalyst 9300X switch into a mixed stack**: A new switch capable of High Speed Stacking is configured for low speed by default. If a Catalyst 9300X switch is not configured to the manufacturing default configuration you can use the **switch stack-speed low** command to change its speed to legacy speed (480 Gbps) to match the stack. Mixed stacking can function only at low speed (480 Gbps). The new switch will join the mixed stack and function at low speed.

Preconfiguring a Switch to a Speed Setting

You can use the following methods to configure a switch to the desired speed setting.

- **Cisco Zero Day Deployment**: You can apply the startup configuration with the desired speed setting on a stand-alone switch using Cisco Zero Day Deplyment.
- **CLI**: You can connect the stand-alone switch to a console and enter the configuration command **switch stack-speed [high | low]**. You can configure the desired speed and reload the switch. After the reload you can insert the switch into the stack.
- **Configuration Auto-install**: You can auto-configure a stand-alone switch by connecting it to a TFTP server reachable by management Gigabit Ethernet port. You can also use a USB key to auto-configure the switch in standalone mode.

Configuring High Speed Stacking

To configure High Speed Stacking, perform this procedure.

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 global configuration mode.
Step 3	**switch stack-speed** [**high** | **low**] **Example:** `Device(config)# switch stack-speed high`	Configures the stack speed to high (1Tbps) or low (480Gbps). The configuration requires a reload to take effect.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show switch stack-ring speed** **Example:** `Device# show switch stack-ring speed`	Displays the current stack ring speed and the stack ring speed after the next reload.
Step 6	**show switch stack-bandwidth**	Displays the current stack bandwidth and the stack bandwidth after the next reload.

Configuration Examples for High Speed Stacking

The following sections provide examples of High Speed Stacking configurations.

Example: Displaying Switch Stack-ring speed

The following example shows how to display the switch stack-ring speed

```
Device#show switch stack-ring speed
Stack Ring Speed        : 1000G
Stack Ring Configuration: Full
Stack Ring Protocol     : StackWise
Stack Ring Next-boot Speed: 1000G.
```

Example: Displaying Switch Stack Bandwidth

The following example shows how to display the switch stack bandwidth.

```
Device#sh switch stack-bandwidth
          Stack    Current                Next-boot
Switch#   Role     Bandwidth   State      Bandwidth
------------------------------------------------------------
*1       Active    480G        Ready      1000G
 2       Standby   480G        Ready      1000G
 3       Member    480G        Ready      1000G
```

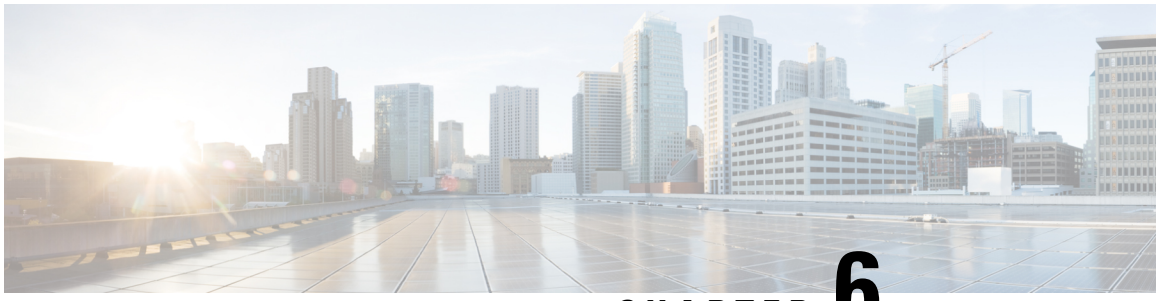
Feature History for Configuring High Speed Stacking

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

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

Release	Feature	Feature Information
Cisco IOS XE Bengaluru 17.5.1	Configuring High Speed Stacking	The High Speed Stacking feature allows you to configure a homogenous stack of switches to run at the speed of 1Tbps. The feature was introduced on the C9300X-12Y and C9300X-24Y models of the Cisco Catalyst 9300 Series Switches.
Cisco IOS XE Bengaluru 17.6.2	Configuring High Speed Stacking	The feature was introduced on the C9300X-48HX and C9300X-48TX models of the Cisco Catalyst 9300 Series Switches.

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



CHAPTER 6

Troubleshooting Stacking and High Availability

Overview

This chapter provides links to documents authored by Cisco subject matter experts (SMEs). They aim to help you resolve technical issues without requiring a support ticket. If these documents are unable to resolve your issue, we recommend visiting the applicable Cisco Community. There is a wealth of information and advice available from fellow Cisco customers who may have experienced this issue already and provided a solution. If you are not able to find a resolution on the Community, it may be best that you raise a support ticket at Cisco Support. In cases where a support ticket has to be raised, these documents provide guidance about the data that should be collected and added to the support ticket. Specify the support document you referred, and TAC can create an improvement request with the document owner.

Support Articles

The documents in this section were created using specific software and hardware listed in the Components Used section of each article. However, this does not mean that they are limited to what is listed in Components Used, and generally remain relevant for later versions of software and hardware. Note that there could be some changes in the software or hardware that can cause commands to stop working, the syntax to change, or GUIs and CLIs to look different from one release to another.

The following are the support articles associated with this technology:

Document	Description
Verify and Troubleshoot Stackwise on Catalyst 9200/9300	This document describes how to troubleshoot common failure scenarios in Stackwise deployments of Catalyst 9200/9200L, 9300/9300L and 9300X switches.
Catalyst 9000 Switches booting to switch: prompt due to Stack 1+1 variable	This document describes a rare problem and solution about Catalyst 9000 Series Switches failing to boot normally and falling into bootloader (switch:) prompt.

Feedback Request

Your input helps. A key aspect to improving these support documents is customer feedback. Note that these documents are owned and maintained by multiple teams within Cisco. If you find an issue specific to the document (unclear, confusing, information missing, etc):

- Provide feedback using the **Feedback** button located at the right panel of the corresponding article. The document owner will be notified, and will either update the article, or flag it for removal.
- Include information regarding the section, area, or issue you had with the document and what could be improved. Provide as much detail as possible.

Disclaimer and Caution

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.