Revised: February 12, 2026

# High Availability

## Scope

This article is designed to help network administrators understand, configure, and manage high availability (HA) features including Nonstop Forwarding with Stateful Switchover (NSF with SSO), Graceful Insertion and Removal (GIR) & In-Service Software Upgrade (ISSU) to ensure minimal downtime and maximize network reliability.

## Overview

High Availability refers to the ability of a system or network to remain accessible and operational even in the event of failures, planned maintenance, or unexpected disruptions. In modern networks, it is essential to maintain service continuity, meet business requirements, and deliver reliable user experiences.

## Feature history table

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

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

| Release | Feature Name and Description | Supported platform |
|---|---|---|
| Cisco IOS XE 17.18.1 | Cisco Nonstop Forwarding with Stateful Switchover (NSF with SSO): 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. | Cisco C9350 Series Smart Switches<br>Cisco C9610 Series Smart Switches |
| Cisco IOS XE 17.18.1 | Graceful Insertion and Removal (GIR): 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. | Cisco C9350 Series Smart Switches<br>Cisco C9610 Series Smart Switches |
| Cisco IOS XE 17.18.2 | In-Service Software Upgrade (ISSU): ISSU is a process that upgrades an image to another image on a device while the network continues to forward packets. ISSU helps network administrators avoid a network outage when performing a software upgrade. ISSU is supported in install mode. | Cisco C9610 Series Smart Switches |

# High availability features and capabilities

This section provides information on the high availability features and capabilities.

### Cisco nonstop forwarding with stateful switchover

Cisco Nonstop Forwarding (NSF) with Stateful Switchover (SSO) is a feature that:

- works to minimize the amount of time a network is unavailable to users following a switchover,
- continues forwarding IP packets following a Route Processor (RP) switchover, and
- provides fault resistance by allowing a standby switch to take over if the active device becomes unavailable.

This feature helps suppress routing flaps in SSO-enabled devices, reducing network instability. It allows for the forwarding of data packets to continue along known routes while routing protocol information restores following a switchover. With NSF/SSO, peer networking devices do not experience routing flaps. Data traffic forwards 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.

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 forwards through intelligent line cards or dual forwarding processors (FPs) while the standby router processor 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.

### Benefits of NSF with SSO

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:** Interfaces remain active on the device assuming the active route processor (RP) role during a switchover, so neighboring devices do not detect a link flap (the link does not go down and come back up).
- **Prevents routing flaps:** SSO continues forwarding network traffic during a switchover, so routing flaps are avoided.
- **Maintains user sessions:** Maintains user sessions established prior to the switchover.

### NSF-aware devices

An NSF-aware device is a networking device that:

- runs NSF-compatible software,
- can still forward packets when an active device election happens, and
- helps send routing protocol information to the neighboring NSF device.

### Cisco express forwarding

Cisco Express Forwarding (CEF) is a key element of NSF that:

- provides packet forwarding in a Cisco networking device,
- 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 are network protocols that:

- run only on the active routing protocol (RP),
- receive routing updates from neighbor devices, and
- request that the NSF-aware neighbor devices send state information to help rebuild routing tables following a switchover.

Routing protocols do not run on the standby RP. The Intermediate System-to-Intermediate System (ISIS) 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.

# Operational workflows

This section provides information on operational workflow for NSF and SSO.

## Cisco nonstop forwarding workflow

NSF always runs with SSO and provides redundancy for Layer 3 traffic by enabling routing protocols (BGP, EIGRP, OSPF) and Cisco Express Forwarding to continue forwarding packets and recover route information during a switchover.

The key components involved in the process are:

- NSF-capable device: A device configured to support NSF, which rebuilds routing information from NSF-aware or NSF-capable neighbors.
- NSF-aware device: A networking device running NSF-compatible software that helps send routing protocol information to the neighboring NSF device and prevents route-flapping.

- Cisco Express Forwarding (CEF): Continues forwarding packets during a switchover while routing protocols rebuild the Routing Information Base (RIB) tables.
- Routing protocols (BGP, EIGRP, OSPF): Enhanced with NSF-capability and awareness to detect a switchover and take necessary actions to continue forwarding network traffic and recover route information from peer devices.

The process involves the following stages:

1. NSF always runs with SSO, providing redundancy for Layer 3 traffic.
2. Routing protocols (BGP, EIGRP, and OSPF) and Cisco Express Forwarding support NSF for forwarding.
3. These routing protocols detect a switchover and take actions to continue forwarding network traffic and recover route information from peer devices.
4. Each protocol depends on Cisco Express Forwarding to continue forwarding packets during switchover while routing protocols rebuild the Routing Information Base (RIB) tables.
5. After the convergence of routing protocols, Cisco Express Forwarding updates the FIB table and removes stale route entries.
6. Cisco Express Forwarding then updates the hardware with the new FIB information.
7. If the active device is configured (with the graceful-restart command) for BGP, OSPF, or EIGRP routing protocols, routing updates automatically send during the active device election.

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.

## SSO operational workflow

SSO ensures high availability by synchronizing the standby device with the active device's configuration and state, allowing for a rapid switchover that minimizes interruption to Layer 2 sessions and continues forwarding traffic.

The key components involved in the process are:

- Active device: The primary device that handles network traffic and synchronizes its state with the standby device.
- Standby device: The redundant device that starts in a fully-initialized state and maintains synchronization with the active device, ready to take over if the active device fails.

The process involves the following stages:

1. When a standby device runs in SSO mode, the standby device starts up in a completely initialized state.
2. The standby device synchronizes with the persistent configuration and the running configuration on the active device.
3. The standby device subsequently maintains the state of the protocols, and all changes in hardware and software states for features that support SSO are kept in synchronization.
4. If the active device fails, the standby device becomes the active device.
5. This new active device uses existing Layer 2 switching information to continue forwarding traffic.

Layer 3 forwarding is delayed until routing tables repopulate in the newly active device.

SSO offers minimum interruption to Layer 2 sessions in a redundant active device configuration.

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.

# Routing protocol operations with NSF SSO

This section explains how various routing protocols—including BGP, EIGRP, and OSPF - operate with Nonstop Forwarding and Stateful Switchover (NSF SSO) to maintain network stability and minimize disruption during route processor switchovers.

## How BGP operates with NSF

BGP operates with NSF by using graceful restart capability to avoid routing flaps during an RP switchover. NSF-aware peers mark routes as stale but continue forwarding, allowing the newly active RP to reestablish sessions and converge routing information without packet loss.

The key components involved in the process are:

1. NSF-capable device: Initiates a BGP session and declares graceful restart capability in its OPEN message.
2. NSF-aware BGP peer: Recognizes graceful restart capability and marks routes as stale but continues using them during a switchover.
3. Non-NSF-aware BGP peer: Ignores graceful restart capability but establishes a BGP session without NSF functionality.

The process involves the following stages:

1. When an NSF-capable device begins a BGP session with a BGP peer, it sends an OPEN message to the peer.
2. Included in the message is a declaration that the NSF-capable device has graceful restart capability.
3. If the BGP peer has this capability, it is aware that the device sending the message is NSF-capable.
4. 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.
5. 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.
6. After an RP switchover occurs, the NSF-capable device reestablishes the session with the BGP peer.
7. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable device as having restarted.
8. At this point, the routing information exchanges between two BGP peers.
9. 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.
10. The NSF-aware device uses the network information to remove stale routes from its BGP table.

The BGP protocol is fully converged, and no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

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.

## How EIGRP operates with NSF

EIGRP operates with NSF by exchanging NSF capabilities in hello packets. An NSF capable device notifies neighbors of a restart, prompting NSF-aware devices to assist by quickly exchanging topology tables, reducing hello timers, and holding known routes until reconvergence.

The key components involved in the process are:

- NSF-capable device: Notifies neighbors of an NSF restart by setting the restart (RS) bit in a hello packet.
- NSF-aware device: Receives notification from an NSF-capable neighbor, immediately exchanges topology tables, and assists the NSF-capable device during reconvergence.

The process involves the following stages:

1. Enhanced Interior Gateway Routing Protocol (EIGRP) NSF capabilities exchange by EIGRP peers in hello packets.
2. The NSF-capable device notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet.
3. 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.
4. The NSF-aware device sends an end-of-table update packet when the transmission of its topology table is complete.
5. The NSF-aware device performs the following actions to assist the NSF-capable device:
   - The EIGRP hello hold timer expires 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 starts. This timer sets the period of time that the NSF-aware device will hold known routes for the NSF-capable neighbor. You configure this timer with the timers NSF route-hold command. The default time period is 240 seconds

     .
6. 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.
7. 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.
8. 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.
9. 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.
10. The NSF-capable device then returns to normal operation.
11. The NSF-aware device will look for alternate paths (go active) for any routes that are not refreshed by the NSF-capable (restarting device).
12. 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.

The EIGRP network maintains connectivity and minimizes disruption during an RP switchover.

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.

## How OSPF operates with NSF

OSPF operates with NSF by sending an OSPF NSF signal during a supervisor engine switchover. This signal prevents neighbor relationship resets, allowing the NSF capable device to rebuild its neighbor list and resynchronize its link state database with NSF-aware neighbors, ensuring continuous forwarding.

The key components involved in the process are:

- OSPF NSF-capable device: Performs a supervisor engine switchover and sends an OSPF NSF signal to neighboring NSF-aware devices.
- NSF-aware device: Recognizes the OSPF NSF signal and refrains from resetting the neighbor relationship, assisting the NSF-capable device in rebuilding its database.

The process involves the following stages:

1. 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.
2. As quickly as possible after a supervisor engine switchover, the NSF-capable device sends an OSPF NSF signal to neighboring NSF-aware devices.
3. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this device should not be reset.
4. As the NSF-capable device receives signals from other devices on the network, it can begin to rebuild its neighbor list.
5. After neighbor relationships are reestablished, the NSF-capable device begins to resynchronize its database with all of its NSF-aware neighbors.
6. At this point, the routing information exchanges between the OSPF neighbors.
7. 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 fully converged, and the network continues to forward packets during the switchover.

**Note**

OSPF support in NSF requires that all neighbor networking devices are NSFaware. 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.

# Requirements and limitations

This section provides information on the required prerequisites & limitations for NSF with SSO.

## Prerequisites for Cisco nonstop forwarding with stateful switchover

The following are prerequisites for configuring Cisco NSF with SSO:

- 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 (SSO)

The following are restrictions for configuring Cisco NSF with SSO:

- You must have SSO configured on the device for NSF operation.
- 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.

# Configure cisco nonstop forwarding with stateful switchover

To enable Nonstop Forwarding (NSF) functionality by configuring Stateful Switchover (SSO) on a Cisco networking device, you must configure SSO in order to use NSF with any supported protocol.

Follow these steps to configure Cisco Nonstop Forwarding with Stateful Switchover:

Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **show redundancy states**<br>**Example:**<br>`Device# show redundancy states` | Display the operating redundancy mode. |
| **Step 3** | **redundancy**<br>**Example:** | Enter redundancy configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config)# redundancy` | |
| **Step 4** | **mode sso**<br>**Example:**<br>`Device(config-red)# mode sso` | Configure stateful switchover.<br>When this command is entered, the standby switch is reloaded and begins to work in SSO mode. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-red)# end` | Exit redundancy configuration mode and return to privileged EXEC. |
| **Step 6** | **show redundancy states**<br>**Example:**<br>`Device# show redundancy states` | Displays the operating redundancy mode. |
| **Step 7** | **debug redundancy states**<br>**Example:**<br>`Device# debug redundancy states` | Enables the debugging of redundancy status events.<br>When this command is entered, the standby switch begins to work in SSO mode.<br>**Note** On systems using StackWise Virtual Link (SVL) or stacked configurations, SSO mode is enabled by default. Explicit configuration may not be necessary in these scenarios. |

# Examples and monitoring NSF with SSO

The following section displays verifying and monitoring examples for NSF with SSO.

## Verify cisco express forwarding with nonstop forwarding

To display the state of Cisco Express Forwarding on a networking device, particularly its status in relation to NSF, follow these steps to verify Cisco Express Forwarding with Cisco Nonstop Forwarding:

Display the state of Cisco Express Forwarding on a networking device. For 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 nonstop forwarding with 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
```

# Graceful insertion and removal

A Graceful Insertion and Removal (GIR) is a maintenance procedure that:

- isolates a switch from the network in order to perform debugging or an upgrade,
- minimizes traffic disruption during network insertion and removal, and
- uses templates to control protocol and port operations.

GIR allows network administrators to perform maintenance operations on network devices without disrupting network traffic. 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.

The switch can be put into maintenance mode using the **start maintenance** command. When switch maintenance is complete, the switch returns to normal mode on either reaching the configured maintenance timeout, or by enabling the **stop maintenance** command.

For this, 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 a 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 on the 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. However, 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. You can create a new snapshot template using the **snapshot-template** *template-name* command. The command snapshot-template default-snapshot-template specifies the default snapshot template in the maintenance mode. The **snapshot create[template** *template-name***]** *snapshot-name snapshot-description* command applies a specific template to the snapshot create feature.

GIR supports Layer 2 interface shutdown, ISIS routing protocol, HSRP, VRRPv3, and BGP. You can configure this either by creating customized templates or without a template.

### Layer 2 interface shutdowns

A Layer 2 Interface Shutdown is a controlled process where layer 2 interfaces are temporarily disabled or shut down during maintenance operations.

A Layer 2 interface is a network connection point on a device that:

- operates at the data link layer of the OSI model,
- handles frame transmission between network segments, and
- enables devices to communicate within the same local network or VLAN.

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 templates

A custom template is a configuration artifact that:

- gets applied when the network system enters maintenance mode,
- allows isolation of specific protocols, and
- prevents updates once it has been activated in maintenance mode.

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, 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 **system mode maintenance** command and enable the feature using the template template-name class 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 and 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.

# Graceful insertion and removal with Layer 2 Spanning Tree Protocol (STP)

Graceful Insertion and Removal (GIR) allow a network device to be removed from or inserted into a network with minimal disruption to traffic flow. When combined with Spanning Tree Protocol (STP), GIR ensures that the network remains stable and avoids unnecessary STP reconvergence during maintenance activities on a switch.

When a switch is undergoing maintenance, GIR signals to STP to avoid the switch during path selection. The switch undergoing maintenance sets the STP bridge priority to a very high value, effectively making it a less desirable path. This allows STP to converge on an alternate path before the switch is completely removed from the network, preventing traffic loss.

# Graceful insertion and removal with routing protocols

This section describes how Graceful Insertion and Removal (GIR) operates with various routing protocols, detailing the mechanisms used to minimize disruption and maintain network stability during planned maintenance.

### GIR with Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) is a link-state routing protocol used to distribute routing information within a single Autonomous System (AS). When GIR is used with OSPF, the switch undergoing maintenance gracefully removes itself from the OSPF topology, preventing traffic loss and minimizing disruption.

During GIR, the switch sets its OSPF router priority to 0, making it ineligible to become a Designated Router (DR) or Backup Designated Router (BDR). The switch also advertises a maximum age for its Link State Advertisements (LSAs), causing other routers to prefer alternate paths.

#### GIR with Intermediate System to Intermediate System (ISIS)

Intermediate System to Intermediate System (ISIS) is a link-state routing protocol similar to OSPF. It is often used in service provider networks. When GIR is used with ISIS on a switch, the switch undergoing maintenance gracefully removes itself from the ISIS topology.

During GIR, the switch sets the overload bit in its ISIS advertisements. This signals to other routers to avoid using the switch as a transit path.

#### GIR with Enhanced Interior Gateway Routing Protocol (IGRP)

Enhanced Interior Gateway Routing Protocol (EIGRP) is an advanced distance-vector routing protocol. When GIR is used with EIGRP on a switch, the switch undergoing maintenance gracefully removes itself from the EIGRP topology.

During GIR, the switch sends EIGRP messages to its neighbors, informing them that it is going down for maintenance. This allows the neighbors to quickly converge on alternate paths.

#### GIR with Border Gateway Protocol (BGP)

Border Gateway Protocol (BGP) is a path-vector routing protocol used to exchange routing information between different Autonomous Systems (AS). When GIR is used with BGP on a switch, the switch undergoing maintenance gracefully removes itself from the BGP topology.

During GIR, the switch sends BGP route withdrawal messages to its neighbors. It also sets a local preference to a lower value, making its routes less desirable.

#### GIR with Hot Standby Router Protocol (HSRP)

First Hop Redundancy Protocol (FHRP) is a group of networking protocols designed to provide high availability and redundancy for the default gateway in a network. Hot Standby Router Protocol (HSRP) is a Cisco proprietary FHRP that provides redundancy for IP networks. It allows multiple switches to share a virtual IP address and MAC address, providing a seamless failover in case of a switch failure.

HSRP allows one switch to act as the active gateway and another as the standby gateway. During GIR, the switch relinquishes its active HSRP role, allowing the standby switch to take over. This minimizes traffic disruption during maintenance.

#### GIR with Virtual Router Redundancy Protocol (VRRP)

Virtual Router Redundancy Protocol (VRRP) is an industry-standard FHRP similar to HSRP. It provides redundancy for IP networks by allowing multiple switches to share a virtual IP address and MAC address.

VRRP allows multiple switches to act as backups for the default gateway, and the switch with the highest priority becomes the master (active). During GIR, the switch relinquishes its master VRRP role, allowing the backup switch to take over. This minimizes traffic disruption during maintenance.

# Configuration and deployment

This section provides step-by-step procedures setting up GIR maintenance templates and modes.

## Create a maintenance template

Follow these steps to create a maintenance template:

Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters the global configuration mode. |
| **Step 3** | **maintenance-template** *template_name*<br>**Example:**<br>`Device(config)# maintenance-template girl` | Creates a template with the specified name |
| **Step 4** | **router** *routing_protocol instance_id* \| **shutdown l2**<br>**Example:**<br>`Device(config-maintenance-templ)# router isis 1`<br>`Device(config-maintenance-templ)# shutdown l2`<br>`Device(config-maintenance-templ)# router bgp AS-number` | Creates instances that should be isolated under this template.<br>**router:** Configures routing protocols and associated instance id.<br>**shutdown l2:** Shuts down layer 2 interfaces. |

## Configure system mode maintenance

Follow these steps to configure system mode maintenance:

Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters the global configuration mode. |
| **Step 3** | **maintenance-template** *template_name*<br>**Example:**<br>`Device(config)# system mode maintenance` | Enter system mode maintenance configuration mode.<br>Different subcommands 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** | Configure maintenance mode parameters.<br>**timeout:** Configures maintenance mode timeout period in minutes, after which the system automatically returns to normal mode. The default timeout value is never.<br>**template:** Configures maintenance mode using the specified template. |

| | Command or Action | Purpose |
|---|---|---|
| | | **failsafe:** Configures client-ack timeout value.<br>**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. If the system is going into maintenance mode, it continues to reach maintenance. If the system is exiting from maintenance mode, then it reaches normal mode. |

## Start and stop maintenance mode

Follow these steps to start and stop maintenance mode:

Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **start maintenance**<br>**Example:**<br>`start maintenance` | Puts the system into maintenance mode. |
| **Step 3** | **maintenance-template** *template_name*<br>**Example:**<br>`Device(config)# system mode maintenance` | Enter system mode maintenance configuration mode.<br>Different subcommands to create maintenance mode parameters are configured in this mode. |
| **Step 4** | **stop maintenance**<br>**Example:**<br>`Device# stop maintenance` | Enter system mode maintenance configuration mode. |

# Examples and monitoring for GIR

This section provides practical configuration examples and monitoring commands for GIR.

## Example: configure 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: configure 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: start and stop 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: display 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
```

## Monitor graceful insertion and removal

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

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. |

Global Configuration Commands for Troubleshooting

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

# In-Service Software Upgrade (ISSU)

ISSU is a process that upgrades an image to another image on a device while the network continues to forward packets. ISSU helps network administrators avoid a network outage when performing a software upgrade. The images are upgraded in install mode wherein each package is upgraded individually.

ISSU supports upgrade and rollback of software.

### ISSU upgrade

The following steps describe the process that is followed in performing ISSU:

1. Copy the new image to the standby and active supervisor modules.
2. Unzip the files and copy packages to both the active and standby supervisor modules.
3. Install the packages on the standby supervisor module.
4. Restart the standby supervisor module.

   The standby supervisor module is now upgraded to the new software.
5. Install the packages on the active supervisor module.
6. Restart the active supervisor module and switchover the standby to new active supervisor module. After the switchover, the new standby supervisor module will be up with the new software. The new software image is already installed on the new active supervisor module, hence ISSU is completed.

### ISSU upgrade: 3-step work flow

This workflow involves three steps - add, activate, and commit. After activation, all switches are upgraded to new software version except that the software is not committed automatically but must be performed manually via the install commit command. The advantage of this approach is the system can be rolled back to a previous software version. The system automatically rolls back if the rollback timer is not stopped using the install auto-abort-timer stop or the install commit command. If the rollback timer is stopped, the new software version could be run on the device for any duration and then rolled back to the previous version.

#### ISSU upgrade: 1-step work flow

This workflow involves only one step and helps in optimization. You cannot roll back as the upgrade is committed automatically.

# Prerequisites for performing ISSU

The following prerequisites apply when performing In-Service Software Upgrade (ISSU):

- The active supervisor module must have access to the new IOS XE image or pre-load it into flash.
- The device must be running in install mode.
- Non-Stop Forwarding (NSF) must be enabled.

# Restrictions and guidelines for performing ISSU

- Upgrading hardware and software simultaneously is not supported. Only one upgrade operation can be performed at a time.
- We recommend that upgrades are performed during a maintenance window.
- Do not perform any configuration changes while the ISSU process is being performed.
- Do not use CTRL-C or any other control operations on the active switch or remote sessions while ISSU is in progress as this will abruptly end the ISSU process on the device.
- Downgrade with ISSU is not supported.
- If synchronization between the active and standby supervisor modules fail during an ISSU, the system reboots five consecutive times within an interval of 25 minutes before switching to ROMMON mode. When this switchover happens, change the standby supervisor from manual boot mode to auto boot mode.

# Software upgrade and monitoring procedures

This section provides information on software upgrade and monitoring procedures for ISSU.

## Upgrade software using 3-step workflow

Follow these steps to configure system mode maintenance:

Procedure

The device must be booted in the install mode.

There must be two supervisor modules in the system. Both the supervisor modules must be running the same image.

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

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **install add file** { **ftp: \| tftp: \| flash: \| disk:** *cat9k_iosxe.17.18.01.SPA.bin* }<br>**Example:**<br>`Switch# install add file ftp:file.bin` | This command downloads the image into the bootflash and expands it on both the supervisor modules. |
| **Step 3** | **install activate issu**<br>**Example:**<br>`Switch# install activate issu` | On executing this command, the following sequence of events occurs:<br>• A rollback timer is started. If the rollback timer expires, the system rolls back to the same state before the start of the ISSU. The rollback timer can be stopped by using the install auto-abort-timer stop command. ISSU can be rolled back using install abort issu command.<br>• The standby supervisor module is provisioned with the new software and it reloads with the new software version. Next, the active supervisor module is provisioned with the new software and it reloads. The can be rolled back using install abort issu command.<br>• The standby supervisor module is provisioned with the new software and it reloads with the new software version. Next, the active supervisor module is provisioned with the new software and it reloads. The standby supervisor module with the new image now becomes the active supervisor module and the old active supervisor module becomes the standby.<br>• At the end of this procedure, both the supervisor modules run with the new software image. |
| **Step 4** | **install commit**<br>**Example:**<br>`Switch# install commit` | The commit command performs the necessary clean up, enables the new software as permanent (removing the older version of the software) and stops the rollback timer. Any reboot after the commit will boot with new software.<br>**Note** There is no rollback when this command is used. |

## Upgrade software using 1-step workflow

Follow these steps to configure system mode maintenance:

Procedure

The device must be booted in the install mode.

There must be two supervisor modules in the system. Both the supervisor modules must be running the same image.

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **install add file flash:** *cisco9k_iosxe.17.18.01.SPA.bin* **activate issu commit** | Automates the sequence of all upgrade procedures that include downloading the images to both the switches and expanding into packages, and upgrading each switch as per the procedure.<br>**Note** This command throws an error if the switch is booted with a bundle image. |

## Monitoring ISSU

To verify ISSU on StackWise Virtual, use the following **show** commands:

| Command | Description |
|---|---|
| **show issu clients** | Displays a list of the current ISSU clients--that is, the network applications and protocols supported by ISSU. |
| **show issu message types** | Displays the formats, versions, and size of ISSU messages supported by a particular client. |
| **show issu negotiated** | Displays results of a negotiation that occurred concerning message versions or client capabilities. |
| **show issu sessions** | Displays detailed information about a particular ISSU client, including whether the client status is compatible for the impending software upgrade. |
| **show issu comp-matrix** | Displays information regarding the ISSU compatibility matrix. |
| **show issu entities** | Displays information about entities within one or more ISSU clients. |
| **show issu state [detail]** | Displays the current ISSU state. |