Cisco Nexus 3550-T NX-OS System Management Configuration Guide, Release 10.2(x)

First Published: 2022-09-18

Last Modified: 2022-09-18

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

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3550-T Switches

The entire Cisco Nexus 3550-T switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3550-series/series.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

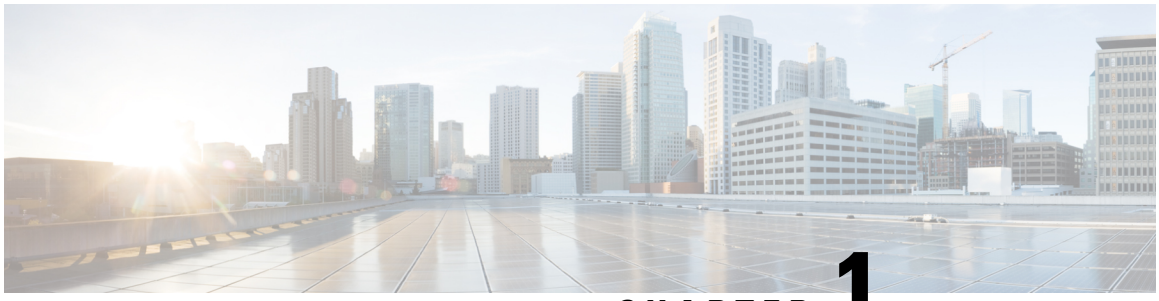
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

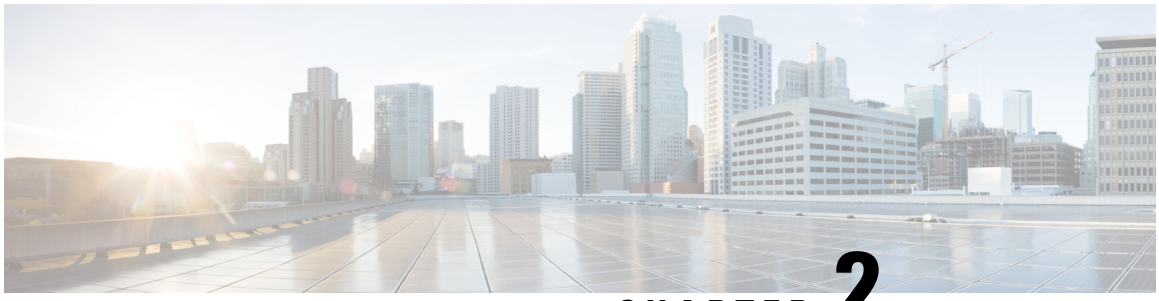
This section contains the new and changed information for a release.

New and Changed Information

Table 1: New and Changed Information for Cisco Nexus 3550-T NX-OS Release 10.2(x)

Feature	Description	Changed in Release	Where Documented
Support for UTC Offset Correction in HPT	High Precision Timestamping (HPT) now supports Coordinated Universal Time (UTC) offset correction.	10.2(3v)	Overview
Support for PTP	Precision Time Protocol (PTP) enables synchronization of clocks which are distributed across a network.	10.2(3t)	Configuring PTP, on page 5
Support for Secure Erase	The secure erase feature erases all customer information on Nexus 3550-T switches.	10.2(3t)	Configuring Secure Erase, on page 37
Support for SPAN	A switch port analyzer (SPAN) is used to analyze traffic between ports on Cisco NX-OS devices.	10.2(3t)	Configuring SPAN, on page 45
Support for HPT	Support for High Precision Timestamping (HPT) of packets arriving at the ingress port.	10.2(3t)	Configuring High Precision Timestamping , on page 41

New and Changed Information



CHAPTER 2

System Management Overview

Software Image

The Cisco NX-OS software consists of one NXOS software image. This image runs on all Cisco Nexus 3550-T switches.

Precision Time Protocol

Precision Time Protocol (PTP) is a time synchronization protocol defined in IEEE 1588 for nodes distributed across a network. With PTP, it is possible to synchronize distributed clocks with an accuracy of less than 1 microsecond via Ethernet networks. PTP is supported on IPv4 multicast, two-step master, version-2 only with boundary clock functionality.

Cisco Discovery Protocol

You can use the Cisco Discovery Protocol (CDP) to discover and view information about all Cisco equipment that is directly attached to your device. CDP runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. CDP is media and protocol independent, and gathers the protocol addresses of neighboring devices, discovering the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is a vendor-neutral, one-way device discovery protocol that allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other. You can enable LLDP globally or per interface.

Secure Erase

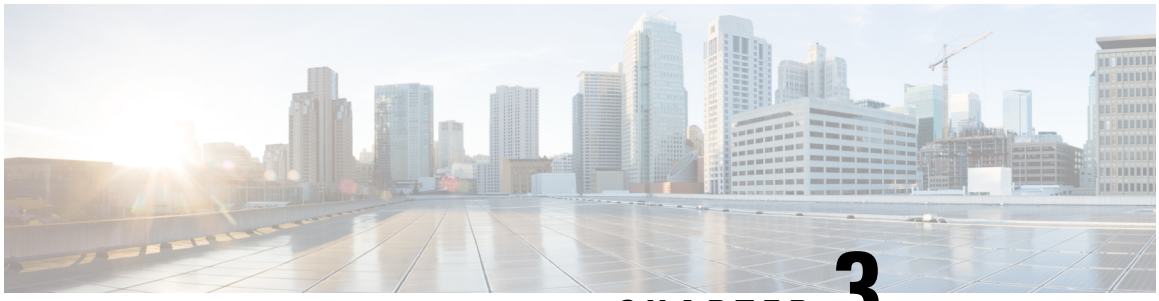
The Secure Erase feature erases all customer information for Nexus 3550-T switches. Secure Erase is an operation to remove all the identifiable customer information on Cisco NX-OS devices in conditions of product removal due to Return Merchandise Authorization (RMA), or upgrade or replacement, or system end-of-life.

High Precision Timestamping

The High Precision Timestamping (HPT) feature enables high-precision timestamping on packets ingressing on a Cisco Nexus N3550-T switch. The time-stamp corresponds to the time the packet has arrived on a N3550-T front-panel port. Timestamping is supported for data packets going through the fabric. The feature can be enabled on any egress port. Also known as Rx timestamping.

Switched Port Analyzer

You can configure an Ethernet Switched Port Analyzer (SPAN) to monitor traffic in and out of your device. The SPAN features allow you to duplicate packets from source ports to destination ports.



CHAPTER 3

Configuring PTP

This chapter describes how to configure the Precision Time Protocol (PTP) on Cisco NX-OS devices.

This chapter includes the following sections:

About PTP

PTP is a time synchronization protocol defined in IEEE 1588 for nodes distributed across a network. With PTP, it is possible to synchronize distributed clocks with an accuracy of less than 1 microsecond via Ethernet networks.

A PTP system can consist of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks, and transparent clocks. Non-PTP devices include ordinary network switches, routers, and other infrastructure devices.

PTP is a distributed protocol that specifies how real-time PTP clocks in the system synchronize with each other. These clocks are organized into a master-slave synchronization hierarchy with the grandmaster clock, which is the clock at the top of the hierarchy, determining the reference time for the entire system. Synchronization is achieved by exchanging PTP timing messages, with the members using the timing information to adjust their clocks to the time of their master in the hierarchy. PTP operates within a logical scope called a PTP domain.

PTP supports the following functionality:

- Multicast PTP transport—In the multicast transport mode, PTP uses multicast destination IP address 224.0.1.129 as per IEEE 1588 standards for communication between devices. For the source IP address, it uses the user configurable global IP address under the PTP domain.
- PTP multicast configuration is supported only under physical interface for L2 or L3. PTP is not supported for virtual interfaces such as Port-channel, SVI, and tunnel.

- PTP encapsulation over UDP over IP—PTP uses UDP as the transport protocol over IP. PTP uses UDP ports 319 for event messages and 320 for general messages communication between devices.
- PTP profiles—PTP supports default (1588) and SMPTE 2059-2 profiles. They all have different ranges of sync and delay request intervals. For information on the default profile, refer to IEEE 1588. For more information on SMPTE 2059-2, refer to the respective specifications.
- Path delay measurement—We support delay request and response mechanism to measure the delay between the master and slave devices.
- Message intervals—You can configure the interval at which the announce, sync, and delay request messages needs to be sent between devices.
- Best master clock (BMC) selection—BMC algorithm is used to select master, slave, and passive states of the PTP enabled interfaces based on the Announce message received as per 1588 specification.

PTP Device Types

The PTP device type is configurable and can be used to set the clock type.

Clocks

The following clocks are common PTP devices:

Ordinary clock

Communicates with the network based on a single physical port, similar to an end host. An ordinary clock can function as a grandmaster clock.

Boundary clock

Typically has several physical ports, with each port behaving like a port of an ordinary clock. However, each port shares the local clock, and the clock data sets are common to all ports. Each port decides its individual state, either master (synchronizing other ports connected to it) or slave (synchronizing to a downstream port), based on the best clock available to it through all of the other ports on the boundary clock. Messages related to synchronization and establishing the master-slave hierarchy terminate in the protocol engine of a boundary clock and are not forwarded.

Transparent clock

Forwards all PTP messages like an ordinary switch or router but measures the residence time of a packet in the switch (the time that the packet takes to traverse the transparent clock) and in some cases the link delay of the ingress port for the packet. The ports have no state because the transparent clock does not need to synchronize to the grandmaster clock.

There are two kinds of transparent clocks:

End-to-end transparent clock

Measures the residence time of a PTP message and accumulates the times in the correction field of the PTP message or an associated follow-up message.

Peer-to-peer transparent clock

Measures the residence time of a PTP message and computes the link delay between each port and a similarly equipped port on another node that shares the link. For a packet, this incoming link delay

is added to the residence time in the correction field of the PTP message or an associated follow-up message.

Note PTP operates only in boundary clock mode. Cisco recommends deployment of a Grand Master Clock (10 MHz) upstream, with servers containing clocks requiring synchronization connected to the switch.

End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.

Clock Modes

The IEEE 1588 standard specifies two clock modes for the PTP supporting devices to operate in: one-step and two-step.

One-Step Mode:

In one-step mode the clock synchronization messages include the time at which the master port sends the message. The ASIC adds the timestamp to the synchronization message as it leaves the port.

The slave port uses the timestamp that comes as part of the synchronization messages.

Two-Step Mode:

In two-step mode the time at which the synchronization message leaves the port is sent in a subsequent follow-up message. This is the default mode.

Note Cisco Nexus 3550-T Release 10.2(3t) supports only Two-Step Mode.

PTP Process

The PTP process consists of two phases: establishing the master-slave hierarchy and synchronizing the clocks.

Within a PTP domain, each port of an ordinary or boundary clock follows this process to determine its state:

- Examines the contents of all received announce messages (issued by ports in the master state)
- Compares the data sets of the foreign master (in the announce message) and the local clock for priority, clock class, accuracy, and so on
- Determines its own state as either master or slave

The ordinary and boundary clocks use **Sync**, **Delay_Req**, **Follow_Up**, **Delay_Resp** event messages to generate and communicate timing information.

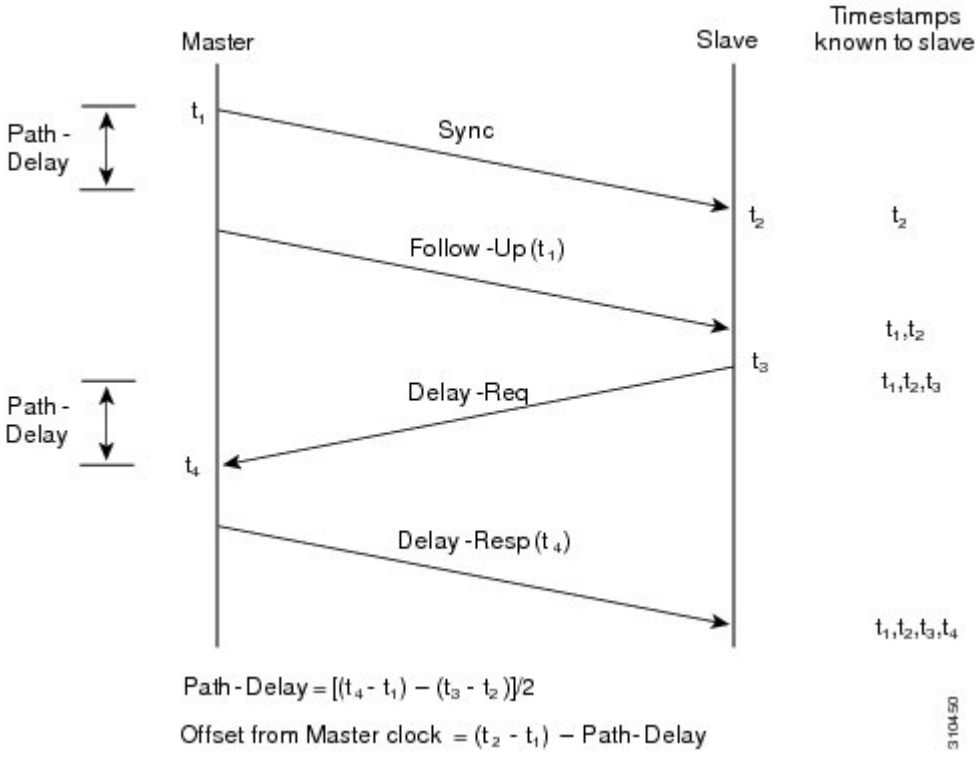
These messages are sent in the following sequence:

1. The master sends a **Sync** message to the slave and notes the time, `t1` at which it was sent. For one-step **Sync** message carries the time when the message leaves the master and for two-step this time is sent in the subsequent **Follow-Up** event message.
2. The slave receives the **Sync** message and notes the time of reception, `t2`.
3. The master conveys to the slave the timestamp, `t1` by embedding the timestamp in a **Follow_Up** event message.

4. The slave sends a **Delay_Req** message to the master and notes the time, `t3` at which it was sent.
5. The master receives the **Delay_Req** message and notes the time of reception, `t4`.
6. The master conveys to the slave the timestamp, `t4` by embedding it in a **Delay_Resp** message.
7. After this sequence, the slave possesses all four timestamps. These timestamps can be used to compute the offset of the slave clock relative to the master, and the mean propagation time of messages between the two clocks.

   The following figure describes the event messages in the PTP process that generate and communicate timing information.

Figure 1: PTP Process



High Availability for PTP

Stateful restarts are not supported for PTP. After a reboot, the running configuration is applied.

Guidelines and Limitations for PTP

Note For scale information, see the release-specific *Cisco Nexus 3550-T Series NX-OS Verified Scalability Guide*.

The following are the guidelines and limitations for Cisco Nexus 3550 Series switches for PTP:

- For PTP to function properly, you must use the latest SUP and line card FPGA versions.
- PTP domain limits to a single domain per network.
- PTP transport over User Datagram Protocol (UDP) is supported.
- PTP supports boundary clock mode. End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.
- PTP can be enabled on the port-channel member ports.
- All management messages that are received from a slave port are forwarded to all PTP enabled ports. The management messages that are received from a slave port are not handled.
- When configuring PTP on Cisco Nexus 3550-T Series switches, set the clock protocol to use PTP through the clock protocol ptp vdc 1 command. NTP cannot coexist with PTP configured to a Cisco Nexus 9000 series switch.
- The PTP correction-range, PTP correction-range logging, and PTP mean-path-delay commands are supported on the Cisco Nexus 3550-T platform.
- PTP is not supported for stateful high availability.
- PTP is not supported for management interfaces.
- Each port can be individually configured with any of the supported PTP profiles. Different PTP profiles can coexist on an interface. Combination of the default of 1588 and SMPTE-2059-2 profile is supported.
- Beginning with Cisco NX-OS 3550-T Release 10.2(3t), PTP Media Profile is supported on the Cisco Nexus 3550-T platform switches. A few guidelines and limitations for this platform switches are as follows:
  - IPv4 multicast, two-step mode, and PTPv2 with boundary clock functionality is supported.
  - PTP sync interval and PTP delay-request interval of -3 log seconds is recommended for +-500ns correction range.
  - Other PTP features such as unicast and unicast negotiation are not supported.

Default Settings for PTP

The following table lists the default settings for PTP parameters.

Table 2: Default PTP Parameters

Parameters	Default
PTP	Disabled
PTP version	2
PTP domain	0
PTP priority 1 value when advertising the clock	255
PTP priority 2 value when advertising the clock	255

Parameters	Default
PTP announce interval	1 log second
PTP announce timeout	3 announce intervals
PTP delay-request interval	• 0 log seconds
PTP sync interval	• -2 log seconds
PTP VLAN	Default vlan is 1.

Configuring PTP

Configuring PTP Globally

You can enable or disable PTP globally on a device. You can also configure various PTP clock parameters to help determine which clock in the network has the highest priority to be selected as the grandmaster.

Note You must always set the clock protocol PTP vdc1 for the local clock to be updated by the PTP protocol. You can verify the configuration using the **show running-config clock_manager** command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature ptp** **Example:** `switch(config)# feature ptp`	Enables or disables PTP on the device. **Note** Enabling PTP on the switch does not enable PTP on each interface.
Step 3	(Optional) [**no**] **ptp domain** *number* **Example:** `switch(config)# ptp domain 1`	Configures the domain number to use for this clock. PTP domains allow you to use multiple independent PTP clocking subdomains on a single network. The range for the *number* is from 0 to 127.
Step 4	(Optional) [**no**] **ptp priority1** *value* **Example:** `switch(config)# ptp priority1 1`	Configures the priority1 value to use when advertising this clock. This value overrides the default criteria (clock quality, clock class, and

	Command or Action	Purpose
		so on) for best master clock selection. Lower values take precedence. The range for the *value* is from 0 to 255. **Note** For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.
Step 5	(Optional) [**no**] **ptp priority2** *value* **Example:** `switch(config)# ptp priority2 1`	Configures the priority2 value to use when advertising this clock. This value is used to decide between two devices that are otherwise equally matched in the default criteria. For example, you can use the priority2 value to give a specific switch priority over other identical switches. The range for the *value* is from 0 to 255. **Note** For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.

Step 6	[**no**] **ptp management**
	Example:
	`switch(config)# ptp management`
	`switch(config-ptp-profile)#`
		Configures support for PTP management packets. This command is enabled by default.
		no: Disables support for management packets.

Step 7	(Optional) [**no**] **ptp delay tolerance** { **mean-path** | **reverse-path** } *variation*
	Example:
	`switch(config)# ptp delay tolerance mean-path 50.5`
	`switch(config)#`
		Configures the PTP delay mean path/reverse path tolerance variation.
		mean-path: Ignore spikes in Mean Path Delay (MPD) as calculated by the PTP BMC algorithm.
		reverse-path: Ignore spikes in (t4-t3) as calculated by the PTP BMC algorithm.
		variation: Percentage that defines the tolerance for spikes. Use numeric values with a single decimal. Range is from 1.0 through 100.0.

Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring PTP on an Interface

After you globally enable PTP, it is not enabled on all supported interfaces by default. You must enable PTP interfaces individually.

Before you begin

Make sure that you have globally enabled PTP on the switch and configured the source IP address for PTP communication.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling PTP and enters the interface configuration mode.

Step 3	[**no**] **ptp** **Example:** `switch(config-if)# ptp`	Enables or disables PTP on an interface.
Step 4	(Optional) [**no**] **ptp announce** {**interval** *log-seconds* | **timeout** *count*} **Example:** `switch(config-if)# ptp announce interval 3`	Configures the interval between PTP announce messages on an interface or the number of PTP intervals before a timeout occurs on an interface. The range for the PTP announcement interval is from 0 to 4 log seconds, and the range for the interval timeout is from 2 to 4 intervals.
Step 5	(Optional) [**no**] **ptp delay-request minimum interval** *log-seconds* **Example:** `switch(config-if)# ptp delay-request minimum interval -1`	Configures the minimum interval allowed between PTP delay messages when the port is in the master state. The range is from log(–1) to log(6) seconds, where log(–1) = 2 frames every second.
Step 6	(Optional) [**no**] **ptp delay-request minimum interval** [**smpte-2059-2**] *log-seconds* **Example:**	Configures the minimum interval allowed between PTP delay messages when the port is in the master state.

	Command or Action	Purpose
	`switch(config-if)# ptp delay-request minimum interval smpte-2059-2-1`	***Table 3: PTP Delay-Request Minimum Interval Range and Default Values*** **Option** / **Range** / **Default Value** **smpte-2059-2** / –4 to 5 log seconds / 0 log seconds Without the **smpte-2059-2** option / –1 to 6 log seconds (where –1 = 2 frames every second) / 0 log seconds
Step 7	(Optional) [**no**] **ptp sync interval** *log-seconds* **Example:** `switch(config-if)# ptp sync interval 1`	Configures the interval between PTP synchronization messages on an interface. The range is from log(–3) to log(1) seconds. For the media-related profile information, see the Cisco NX-OS IP Fabric for Medial Solution Guide when configuring PTP for media.
Step 8	(Optional) [**no**] **ptp sync interval** [**smpte-2059-2**] *log-seconds* **Example:** `switch(config-if)# ptp sync interval smpte-2059-2 -1`	Configures the interval between PTP synchronization messages on an interface. ***Table 4: PTP Synchronization Interval Range and Default Values*** **Option** / **Range** / **Default Value** **smpte-2059-2** / –4 to –1 log seconds / –2 log seconds Without the **smpte-2059-2** option / –3 to 1 log seconds / –2 log seconds
Step 9	(Optional) [**no**] **ptp vlan** *vlan-id* **Example:** `switch(config-if)# ptp vlan 1`	Specifies the VLAN for the interface where PTP is being enabled. You can only enable PTP on one VLAN on an interface. The range is from 1 to 4094.
Step 10	(Optional) **show ptp brief** **Example:** `switch(config-if)# show ptp brief`	Displays the PTP status.
Step 11	(Optional) **show ptp port interface** *interface slot/port* **Example:**	Displays the status of the PTP port.

	Command or Action	Purpose
	`switch(config-if)# show ptp port interface ethernet 1/1`	
Step 12	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

PTP Profile Defaults

The following table lists the ranges and default values for the commands that are automatically configured when the global command **ptp profile** is set. You cannot change the range for the affected global commands beyond those allowed by the configured profile. However, in the interface mode, they can be changed if the **ptp profile-override** command is set.

Table 5: Range and Default Values

Parameter	Scope or Configuration Mode	Default Profile's Supported Range of Values	Default Profile's Default Value	With 'ptp profile-override' Configured on an Interface Supported Range of Values (Default is Based on Configured Profile)
mode	global	none	none	no change
domain	global	0 to 63	0	no change
priority1	global	0 to 255	255	no change
priority2	global	0 to 255	255	no change
cost	interface	Not configurable	Not configurable	0 to 255
transport	interface	ipv4	ipv4	ethernet, ipv4
transmission	interface	multicast	multicast	no change
role	interface	dynamic, master, slave	dynamic	no change
announce interval	interface	0 to 4 -3 to 1 with smpte-2059-2	1	-3 to 4 -3 to 1 with smpte-2059-2
delay-request minimum interval	interface	-1 to 6 -4 to 5 with smpte-2059-2	0	-4 to 6 -4 to 5 with smpte-2059-2

Parameter	Scope or Configuration Mode	Default Profile's Supported Range of Values	Default Profile's Default Value	With 'ptp profile-override' Configured on an Interface Supported Range of Values (Default is Based on Configured Profile)
sync interval	interface	-3 to 1 -7 to 0 with smpte-2059-2	-2	-4 to 1 -7 to 0 with smpte-2059-2

Configuring PTP Notifications

Before you begin

You can enable, disable, and customize notifications for the following significant PTP events:

- Change in the Grand Master (GM) clock
- Change in the Parent clock
- Change in the PTP state on a port
- High PTP clock corrections

The notifications are generated by the DME infrastructure based on information it receives from PTP.

Procedure

	Command or Action	Purpose

Step 1	[**no**] **ptp notification type gm-change**
	Example:
	`switch(config)# ptp notification type gm-change`
	`switch(config)#`
		Configures the system to send a change notification if the PTP grand master clock changes.

Step 2	[**no**] **ptp notification type parent-change**
	Example:
	`switch(config)# ptp notification type parent-change`
	`switch(config)#`
		Configures the system to send a change notification if the PTP parent clock changes.

Step 3	[**no**] **ptp notification type port-state-change** [**category** { **all** | **master-slave-only** }] [**interval** { **immediate** | *seconds* [**periodic-notification** { **disable** | **enable** }] }] **Example:**	Configures the system to send a notification if a port state change event occurs. • **category**: Specifies which state changes must occur for a notification to be sent. • **all**: Every port state change is reported.

	Command or Action	Purpose
	`switch(config)# ptp notification type port-state-change category master-slave-only` `switch(config)#`	**Note** Using the **all** option results in many notifications. • **master-slave-only**: Port state changes from and to the master-slave state are only reported. • **interval** *seconds*: Port state change notifications are sent at the configured interval: from 1-300 seconds with a granularity of 1 sec. • **periodic-notification**: Determines if periodic notifications are sent even if a port state change has not occurred during the configured interval. **disable**: A port state change notification is reported only if the current state is not the same as the previously reported state. Any intermediate state changes during the configured periodic interval are ignored. For example, if a port is a MASTER at time X, and changes to DISABLED and then back to MASTER by the time X+periodic-interval occurs, then no notification is generated for the intervening events. **enable**: Port state change notifications are sent at the configured interval, irrespective of a change in the port state. • **interval immediate**: A port State Change Notification is sent when the state changes.

Step 4	[**no**] **ptp notification type high-correction** [**interval** { *seconds* [**periodic-notification** { **disable** | **enable** }] | **immediate** }]
	Example:
	`switch(config)# ptp notification type high-correction interval immediate`
	`switch(config)#`
		Configures the system to send a high-correction notification if a PTP high correction event occurs. A high correction event is when the correction exceeds the value that is configured in the **ptp correction-range** command (see the following optional step).
		• **interval** *seconds*: High-correction notifications are sent at the configured interval: 1–300 seconds with a granularity of 1 second.

	Command or Action	Purpose
		• **periodic-notification**: Determines if periodic notifications are sent even if any high correction has not occurred during the configured interval. **disable**: Send a notification only if high correction events occurred during the configured periodic interval. This is the default setting. **enable**: Send a notifications irrespective of the number of high correction events during the configured periodic interval. If there are no such events, the payload indicates zero high correction events during the periodic interval. • **interval immediate**: Send a notification as soon as a high correction event occurs.

Step 5	(Optional) [**no**] **ptp correction-range** { *nanoseconds* | **logging** }
	Example:
	`switch(config)# ptp correction-range 200000`
	`switch(config)#`
		Configures a threshold that, once exceeded, indicates that a PTP high correction has occurred. Range is 10–1000000000. The default is 100000 (100 microseconds).

Verifying the PTP Configuration

To display the PTP configuration, perform one of the following tasks:

Table 6: PTP Show Commands

Command	Purpose
show ptp brief	Displays the PTP status.
show ptp clock	Displays the properties of the local clock, including clock identity.
show ptp clock foreign-masters-record	Displays the state of foreign masters known to the PTP process. For each foreign master, the output displays the clock identity, basic clock properties, and whether the clock is being used as a grandmaster.
show ptp corrections	Displays the last few PTP corrections.
show ptp counters [**all** | **interface ethernet** *slot/port*]	Displays the PTP packet counters for all interfaces or for a specified interface.

Command	Purpose
show ptp parent	Displays the properties of the PTP parent.
show ptp port interface ethernet *slot/port*	Displays the status of the PTP port on the switch.
show ptp time-property	Displays the PTP clock properties.
show running-config ptp [**all**]	Displays the running configuration for PTP.
clear ptp counters [**all** | **interface ethernet** *slot/port*]	Clears all PTP messages that are received and transmitted on a specific interface or on all interfaces that has PTP enabled.

Configuration Examples for PTP

This example shows how to configure PTP globally on the device, specify the source IP address for PTP communications, and configure a preference level for the clock:

```
switch# configure terminal
switch(config)# feature ptp
switch(config)# ptp source 10.10.10.1
switch(config)# ptp priority1 1
switch(config)# ptp priority2 1
switch(config)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
switch(config)# show ptp clock
PTP Device Type: Boundary clock
Clock Identity : 0:22:55:ff:ff:79:a4:c1
Clock Domain: 0
Number of PTP ports: 0
Priority1 : 1
Priority2 : 1
Clock Quality:
  Class : 248
  Accuracy : 254
  Offset (log variance) : 65535
Offset From Master : 0
Mean Path Delay : 0
Steps removed : 0
Local clock time:Mon Dec 22 14:13:24 2014
```

This example shows how to configure PTP on an interface and configure the intervals for the announce, delay-request, and synchronization messages:

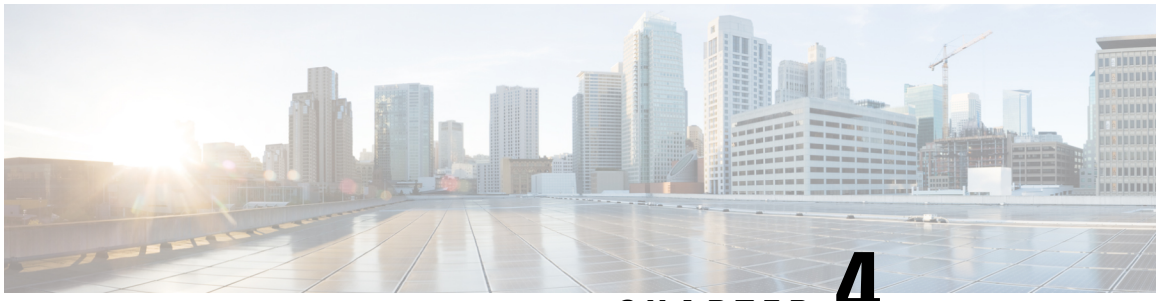
```
switch# configure terminal
switch(config)# interface Ethernet 1/1
switch(config-if)# ptp
switch(config-if)# ptp announce interval 3
switch(config-if)# ptp announce timeout 2
switch(config-if)# ptp delay-request minimum interval smpte-2059-2 -3
switch(config-if)# ptp sync interval smpte-2059-2 -3
switch(config-if)# no shutdown
```

```
switch(config-if)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
Eth1/1 Master
switch(config-if)# show ptp port interface ethernet 1/1
PTP Port Dataset: Eth1/1
Port identity: clock identity: 0:22:55:ff:ff:79:a4:c1
Port identity: port number: 1028
PTP version: 2
Port state: Master
Delay request interval(log mean): 4
Announce receipt time out: 2
Peer mean path delay: 0
Announce interval(log mean): 3
Sync interval(log mean): 1
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
```

Additional References

Related Documents

Related Topic	Document Title
1588 IEEE	1588 IEEE standards



CHAPTER 4

Configuring CDP

This chapter describes how to configure the Cisco Discovery Protocol (CDP) on Cisco NX-OS devices.

This chapter includes the following sections:

About CDP

The Cisco Discovery Protocol (CDP) is a media-independent and protocol-independent protocol that runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. You can use CDP to discover and view information about all the Cisco devices that are directly attached to the device.

CDP gathers protocol addresses of neighboring devices and discovers the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

Each device that you configure for CDP sends periodic advertisements to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain hold-time information, which indicates the length of time that a receiving device should hold CDP information before removing it. You can configure the advertisement or refresh timer and the hold timer.

CDP Version-2 (CDPv2) allows you to track instances where the native VLAN ID or port duplex states do not match between connecting devices.

CDP advertises the following type-length-value fields (TLVs):

- Device ID
- Address
- Port ID
- Capabilities
- Version

- Platform
- Native VLAN
- Full or Half Duplex
- SysName
- SysObjectID
- Management Address
- Physical Location

All CDP packets include a VLAN ID. If you configure CDP on a Layer 2 access port, the CDP packets sent from that access port include the access port VLAN ID. If you configure CDP on a Layer 2 trunk port, the CDP packets sent from that trunk port include the lowest configured VLAN ID allowed on that trunk port. The trunk port can receive CDP packets that include any VLAN ID in the allowed VLAN list for that trunk port. For more information on VLANs, see the *Cisco Nexus® 3550-T Layer 2 Switching Configuration* section.

High Availability

Cisco NX-OS supports both stateful and stateless restarts for CDP.

Virtualization Support

Cisco NX-OS supports one instance of CDP.

Guidelines and Limitations for CDP

CDP has the following configuration guidelines and limitations:

- CDP can discover up to 256 neighbors per port if the port is connected to a hub with 256 connections.
- CDP must be enabled on the device or you cannot enable it on any interfaces.
- You can configure CDP on physical interfaces and port channels only.

Default Settings for CDP

This table lists the default settings for CDP parameters.

Parameters	Default
CDP	Enabled globally and on all interfaces
CDP version	Version 2
CDP device ID	Serial number
CDP timer	60 seconds
CDP hold timer	180 seconds

Configuring CDP

Note The Cisco NX-OS commands for this feature may differ from those commands that are used in Cisco IOS.

Enabling or Disabling CDP Globally

CDP is enabled by default. You can disable CDP and then reenable it.

You must enable CDP on the device before you enable CDP on any interfaces. If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **cdp enable** **Example:** `switch(config)# cdp enable`	Enables or disables the CDP feature on the entire device. It is enabled by default.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling CDP on an Interface

CDP is enabled by default on an interface. You can disable CDP on an interface.

If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**interface** *interface slot/port*
	Example:
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **cdp enable** **Example:** `switch(config-if)# cdp enable`	Enables or disables CDP on this interface. It is enabled by default. **Note** Make sure that CDP is enabled globally on the device.
Step 4	(Optional) **show cdp interface** *interface slot/port* **Example:** `switch(config-if)# show cdp interface ethernet 1/2`	Displays CDP information for an interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional CDP Parameters

You can use the optional commands in this procedure to modify CDP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **cdp advertise** {**v1** | **v2**} **Example:** `switch(config)# cdp advertise v1`	Sets the CDP version that is supported by the device. The default is v2.
Step 3	(Optional) **cdp format device-id** {**mac-address** | **serial-number** | **system-name**} **Example:** `switch(config)# cdp format device-id mac-address`	Sets the CDP device ID. The options are as follows: • **mac-address**—The MAC address of the chassis. • **serial-number**—The chassis serial number/Organizationally Unique Identifier (OUI).

	Command or Action	Purpose
		• **system-name**—The system name or fully qualified domain name. The default is **system-name**.
Step 4	(Optional) **cdp holdtime** *seconds* **Example:** `switch(config)# cdp holdtime 150`	Sets the time that CDP holds onto neighbor information before removing it. The range is from 10 to 255 seconds. The default is 180 seconds.
Step 5	(Optional) **cdp timer** *seconds* **Example:** `switch(config)# cdp timer 50`	Sets the refresh time when CDP sends advertisements to neighbors. The range is from 5 to 254 seconds. The default is 60 seconds.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the CDP Configuration

To display the CDP configuration, perform one of the following tasks:

Command	Purpose
show cdp all	Displays all interfaces that have CDP enabled.
show cdp entry {**all** | **name** *entry-name*}	Displays the CDP database entries.
show cdp global	Displays the CDP global parameters.
show cdp interface *interface slot*/*port*	Displays the CDP interface status.
show cdp neighbors {**device-id** | **interface** *interface slot*/*port*} [**detail**]	Displays the CDP neighbor status.
show cdp interface *interface slot*/*port*	Displays the CDP traffic statistics on an interface.

Use the **clear cdp counters** command to clear CDP statistics on an interface.

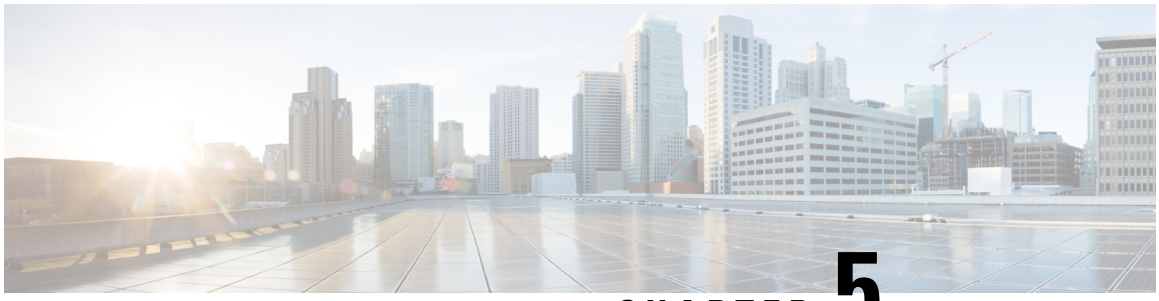
Use the **clear cdp table** command to clear the CDP cache for one or all interfaces.

It is recommended to use the **show cdp neighbors detail** command instead of **show cdp neighbors** command. The **show cdp neighbors** command can display only 13 characters of a platform name. To get the full platform name in the display, use **show cdp neighbors detail** command.

Configuration Example for CDP

This example shows how to enable the CDP feature and configure the refresh and hold timers:

```
configure terminal
cdp enable
cdp timer 50
cdp holdtime 100
```



CHAPTER 5

Configuring LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) in order to discover other devices on the local network.

Note For complete syntax and usage information for the commands used in this chapter, see the command reference for this release and the "System Management Commands" section in the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.

This chapter contains the following sections:

About LLDP

The Cisco Discovery Protocol (CDP) is a device discovery protocol that allows network management applications to automatically discover and learn about other Cisco devices that are connected to the network.

To permit the discovery of non-Cisco devices, the switch also supports the Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard. LLDP allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP is a one-way protocol that transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

LLDP supports a set of attributes that it uses to discover other devices. These attributes contain type, length, and value (TLV) descriptions. LLDP devices can use TLVs to send and receive information to other devices on the network. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

LLDP advertises the following TLVs by default:

- Management address
- Port description
- Port VLAN
- System capabilities
- System description
- System name

High Availability

The LLDP feature supports stateless and stateful restarts. After a reboot, the running configuration is applied.

Virtualization Support

Only one instance of LLDP is supported in the Cisco Nexus® 3550-T switches.

Guidelines and Limitations for LLDP

LLDP has the following configuration guidelines and limitations:

- LLDP must be enabled on the device before you can enable or disable it on any interfaces.
- LLDP is supported only on physical interfaces.
- LLDP can discover up to one device per port.

Default Settings for LLDP

This table lists the LLDP default settings.

Parameters	Default
Global LLDP	Disabled
LLDP on interfaces	Enabled, after LLDP is enabled globally
LLDP hold time (before discarding)	120 seconds
LLDP reinitialization delay	2 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP receive	Enabled, after LLDP is enabled globally
LLDP transmit	Enabled, after LLDP is enabled globally

Configuring LLDP

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP) on the Cisco Nexus® 3550-T switch.

Enabling or Disabling LLDP Globally

You can enable or disable LLDP globally on a device. You must enable LLDP globally to allow a device to send and receive LLDP packets.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature lldp** **Example:** `switch(config)# feature lldp`	Enables or disables LLDP on the device. LLDP is disabled by default.
Step 3	(Optional) **show running-config lldp** **Example:** `switch(config)# show running-config lldp`	Displays the global LLDP configuration. If LLDP is enabled, it shows "feature lldp." If LLDP is disabled, it shows an "Invalid command" error.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling LLDP on an Interface

After you globally enable LLDP, it is enabled on all supported interfaces by default. However, you can enable or disable LLDP on individual interfaces or selectively configure an interface to only send or only receive LLDP packets.

Before you begin

Make sure that you have globally enabled LLDP on the device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot/port*
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 3	[**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Enables or disables the transmission of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 4	[**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Enables or disables the reception of LLDP packets on an interface. After you globally enable LLDP, it is enabled on all supported interfaces by default.
Step 5	(Optional) **show lldp interface** *interface slot/port* **Example:** `switch(config-if)# show lldp interface ethernet 1/1`	Displays the LLDP configuration on the interface.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Multiple LLDP Neighbors Per Physical Interface

Often times a network device sends multiple LLDP packets, out of which one is from the actual host. If a Cisco Nexus switch is communicating with the device but can only manage a single LLDP neighbor per interface, there is a good chance that becoming a neighbor with the actual required host will fail. To minimize this, Cisco Nexus switch interfaces can support multiple LLDP neighbors creating a better opportunity of becoming an LLDP neighbor with the correct device.

Support for multiple LLDP neighbors over the same interface requires LLDP multi-neighbor support to be configured globally.

Enabling or Disabling LLDP Multi-Neighbor Support

Before you begin

Consider the following before enabling LLDP multi-neighbor support on the interfaces:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

- A maximum of three (3) neighbors are supported on an interface.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **lldp multi-neighbor**
	Example:
	`switch(config)# lldp multi-neighbor`
	`switch(config)#`
		Enables or disables LLDP multi-neighbor support for all interfaces globally.

Step 3	**interface** *port* / *slot*
	Example:
	`switch(config)# interface 1/1`
	`switch(config-if)#`
		Specifies the interface on which you are enabling LLDP and enters the interface configuration mode.

Step 4	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the interface. **Note** The transmission of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.
Step 5	(Optional) [**no**] **lldp receive** **Example:**	Disables (or enables) the reception of LLDP packets on the interface.

	Command or Action	Purpose
	`switch(config-if)# lldp receive`	**Note** The reception of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.
Step 6	(Optional) **show lldp interface** *port / slot* **Example:** `switch(config-if)# show lldp interface 1/1`	Displays the LLDP configuration on the interface.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling LLDP Support on Port-Channel Interfaces

Before you begin

Consider the following before enabling LLDP support on port-channels:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

- Applying the **lldp transmit** and **lldp receive** configuration commands to a port-channel does not affect the configuration for the members of the port-channel.
- LLDP neighbors form between the port-channels only when LLDP transmit and receive is configured on both sides of the port-channel.

Note The LLDP transmit and receive commands do not work on MCT and VPC.

If you enable the LLDP port-channel feature globally, the LLDP configuration is not applied to any of these port types. If the configuration is removed from the port-channels or the port type feature is disabled globally, you cannot use the **lldp port-channel** command to enable it on the newly supported port-channels. The command was already issued. To enable LLDP port-channel on the port-channels in question, configure **lldp transmit** and **lldp receive** for each port-channel (see steps 4, 5, and 6 in the following procedure).

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Required: [**no**] **lldp port-channel**
	Example:
	`switch(config)# lldp port-channel`
	`switch(config)#`
		Enables or disables LLDP transmit and receive for all port channels globally.

Step 3	**interface port-channel** [*port-channel-number* | *port-channel-range*]
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
	Example:
	Enter a range of port-channel numbers if you are configuring LLDP over more than one port-channel:
	`switch(config)# interface port-channel 1-3`
	`switch(config-if-range)#`
		Specifies the interface port-channel on which you are enabling LLDP and enters the interface configuration mode.
		Specifies the interface port-channel range on which you are enabling LLDP and enters the interface range configuration mode.

Step 4	(Optional) [**no**] **lldp transmit** **Example:** `switch(config-if)# lldp transmit`	Disables (or enables) the transmission of LLDP packets on the port-channel or range of port-channels. **Note** The transmission of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.
Step 5	(Optional) [**no**] **lldp receive** **Example:** `switch(config-if)# lldp receive`	Disables (or enables) the reception of LLDP packets on the port-channel or range of port-channels. **Note** The reception of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.

	Command or Action	Purpose
Step 6	(Optional) **show lldp interface port-channel**l *port-channel-number* **Example:** `switch(config-if)# show lldp interface port-channel 3`	Displays the LLDP configuration on the port-channel.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Optional LLDP Parameters

You can configure the frequency of LLDP updates, the amount of time for a receiving device to hold the information before discarding it, and the initialization delay time.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) [**no**] **lldp holdtime** *seconds* **Example:** `switch(config)# lldp holdtime 200`	Specifies the amount of time in seconds that a receiving device should hold the information that is sent by your device before discarding it. The range is 10 to 255 seconds; the default is 120 seconds.
Step 3	(Optional) [**no**] **lldp reinit** *seconds* **Example:** `switch(config)# lldp reinit 5`	Specifies the delay time in seconds for LLDP to initialize on any interface. The range is 1 to 10 seconds; the default is 2 seconds.
Step 4	(Optional) [**no**] **lldp timer** *seconds* **Example:** `switch(config)# lldp timer 50`	Specifies the transmission frequency of LLDP updates in seconds. The range is 5 to 254 seconds; the default is 30 seconds.
Step 5	(Optional) **show lldp timers** **Example:** `switch(config)# show lldp timers`	Displays the LLDP hold time, delay time, and update frequency configuration.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the LLDP Configuration

To display the LLDP configuration, perform one of the following tasks:

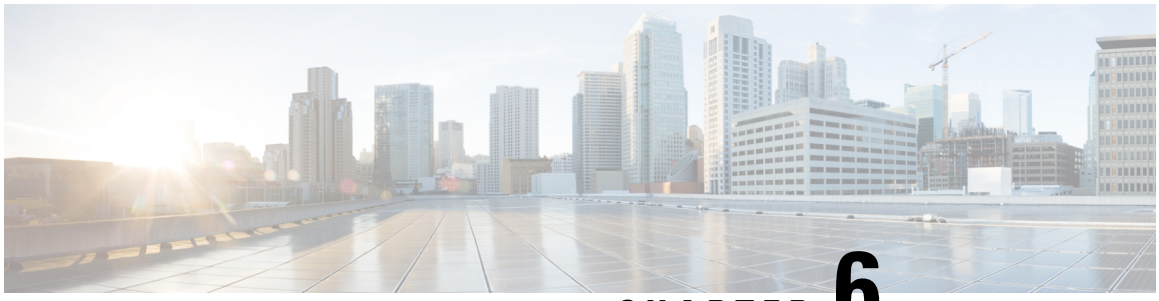
Command	Purpose
show running-config lldp	Displays the global LLDP configuration.
show lldp interface *interface slot*/*port*	Displays the LLDP interface configuration.
show lldp timers	Displays the LLDP hold time, delay time, and update frequency configuration.
show lldp neighbors {**detail** | **interface** *interface slot*/*port*}	Displays the LLDP neighbor device status.
show lldp traffic interface *interface slot*/*port*	Displays the number of LLDP packets sent and received on the interface.

Use the **clear lldp counters** command to clear the LLDP statistics.

Configuration Example for LLDP

This example shows how to enable LLDP on a device; disable LLDP on some interfaces; configure optional parameters such as hold time, delay time, and update frequency:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature lldp
switch(config)# interface ethernet 1/9
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# interface ethernet 1/10
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# lldp holdtime 200
switch(config)# lldp reinit 5
switch(config)# lldp timer 50
```



CHAPTER 6

Configuring Secure Erase

Information about Secure Erase

Beginning with Cisco Nexus 3550-T Release 10.2(3t), the Secure Erase feature is introduced to erase all customer information for Nexus 3550-T switches. Secure Erase is an operation to remove all the identifiable customer information on Cisco NX-OS devices in conditions of product removal due to Return Merchandise Authorization (RMA), or upgrade or replacement, or system end-of-life.

Cisco Nexus 3550-T switches consume storage to conserve system software images, switch configuration, software logs, and operational history. These areas can have customer-specific information such as details regarding network architecture, and design as well as a potential target for data thefts.

The Secure Erase process is used in the following two scenarios:

- Return Material Authorization (RMA) for a device - If you must return a device to Cisco for RMA, remove all the customer-specific data before obtaining an RMA certificate for the device.
- Recovering the compromised device - If the key material or credentials that are stored on a device is compromised, reset the device to factory configuration, and then reconfigure the device.

Note Secure Erase feature will not erase content in External storage.

The device reloads to perform a factory reset which results in the ToR chassis modules to enter the power down mode. After a factory reset, the device clears all configuration, logs, and storage information.

Prerequisites for Performing Secure Erase

- Ensure that all the software images, configurations, and personal data are backed up before performing the secure erase operation.

- Ensure that there is an uninterrupted power supply when the process is in progress.

Guidelines and Limitations for Secure Erase

- Software patches, if installed on the device, will not be restored after the Secure Erase operation.
- If the **factory-reset** command is issued through a session, the session is not restored after the completion of the factory reset process.

  The top of rack switches and supervisor modules returns to the loader prompt.

Configuring Secure Erase

To delete all necessary data before shipping to RMA, configure secure erase using the below command:

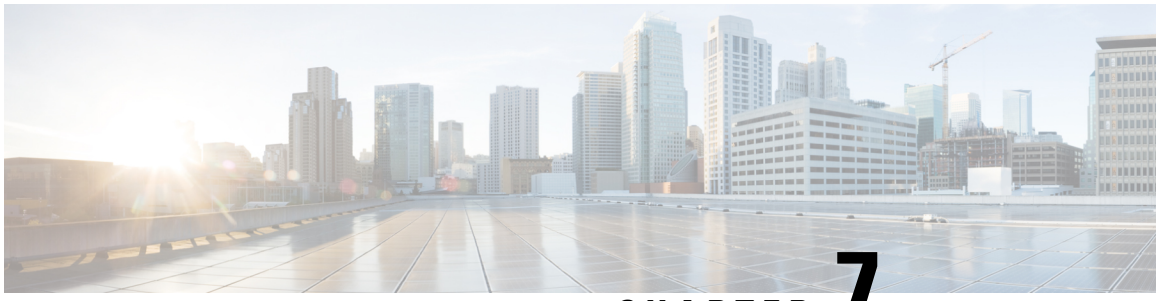
Command	Purpose
factory-reset module *mod* Example: `switch(config)# factory-reset [module <1>]`	Use the command with **all** options enabled. No system configuration is required to use the **factory reset** command. Use the option **mod** to reset the start-up configurations: • For top of rack switches, the command is **factory-reset** or **factory-reset module 1**. After the factory reset process is successfully completed, the switch reboots.

The factory-reset log is displayed below:

```
switch# factory-reset
!!!! WARNING !!!!
The factory reset operation will erase ALL persistent storage on the specified module.
This includes configuration, all log data, and the full contents of flash and SSDs.
Special steps are taken in an effort to render data non-recoverable. Please, proceed with
caution and understanding that this operation cannot be undone and will leave the system
in
a fresh-from-factory state.
!!!! WARNING !!!!
Continue? (y/n) [n] y
A module reload is required for the reset operation to proceed.
Please, wait...

Factory reset requested! Please, do not power off module!

Python 3.7.10
Python Version 3 ...

>>>> Wiping all storage devices ...
+++ Starting NVMe secure erase for /dev/nvme0n1p +++
Using secure format for /dev/nvme0n1p...)
\
---> SUCCESS
```

```
+++ Starting cmos secure erase +++
\
 ---> SUCCESS
+++ Starting nvram secure erase +++
\
 ---> SUCCESS
>>>> Done
>>>> Initializing system to factory defaults ...
+++ Starting init-system +++
\
 ---> SUCCESS
All operations complete! Exiting..
```

Configuring Secure Erase



CHAPTER 7

Configuring High Precision Timestamping

This section has the following details:

Overview

The High Precision Timestamping (HPT) feature (also called as Rx timestamping) enables high precision time stamping of packets arriving at the ingress port of a Cisco Nexus 3550-T switch. This is used to track and/or record data received by the Nexus 3550-T switch. The time-stamping is for data packets going through the fabric (not to/from host). Typically, time stamping is enabled on a span destination port. The timestamp data, that is the HPT trailer, is appended when the packet egresses the port, that has the HPT feature enabled. The application attached to this egress port decodes the data. You can use the N3550-timestamp-decoder available on Github. or Wireshark version 3.0.0+ to decode the data. You can run the decoder tool with *--trailer* and *--offset 20* options.

An HPT trailer incudes the device ID, port ID, timestamp data, flags and CRC. The device ID and port ID are used for identification purpose to map the timestamp data with a device.

Use the the decoder tool as displayed below:

```
[n3550-timestamp-decoder-master/build]$ ./timestamp-decoder --read
/users/<path-to-input-pcap>/HPT_90.cap --trailer --offset 20
```

A sample output is displayed below:

```
2022/09/06-11:59:50.509047248389  (032:046)   106 bytes
```

The first element (with the date and time) displays the timestamp details. The next field (032:046) indicates that device ID is 32, and the port is 46. The port id is typically one less than the interface number, so in this case it indicates that the packet traversed through interface e1/47.

Rx timestamps are by default in the same timescale as the HW clock (for e.g, PTP operates in TAI). Beginning with NX-OS release 10.2(3v), a new command, **time-stamp hpt utc-offset**, has been introduced to enable UTC-offset correction to ensure that the Rx timestamps are in UTC timescale.

Note There is no support to strip the HPT trailer on a NX-OS device.

Limitations

Limitations of HPT are:

- HPT can be enabled on a physical port or a port channel; cannot be enabled on a port channel member.
- HPT configuration on a port needs to be removed before it can be made part of a port channel.

Enabling High Precision Timestamping

Use this procedure to enable HPT on a 3550-T switch port.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**time-stamp hpt device-id** *device_id* **Example:** `switch(config)# time-stamp hpt device-id 10`	This step is optional. The range for device id is 0 to 255; default value is 0.
Step 3	**interface** *interface-type interface-id* **Example:** `switch(config)# interface ethernet 1/2`	Enter the interface configuration mode.
Step 4	**time-stamp hpt** **Example:** `switch(config-if)# time-stamp hpt`	Enables time stamping on the required interface.
Step 5	(Optional) **time-stamp hpt utc-offset** **Example:** `switch(config-if)# time-stamp hpt utc-offset`	Enables Rx timestamping to be converted to UTC format.
Step 6	(Optional) Use either the **show run interface** *type interface-id* or the **show time-stamp hpt brief** command to get the configured HPT details.	Displays the HPT details.

	Command or Action	Purpose

	Example:
	`switch# show run interface ethernet 1/5`
	`or`
	`switch# show time-stamp hpt brief`	

Configuration Examples

In the following example, the device ID is 100, and HPT is configured on interface Ethernet 1/47.

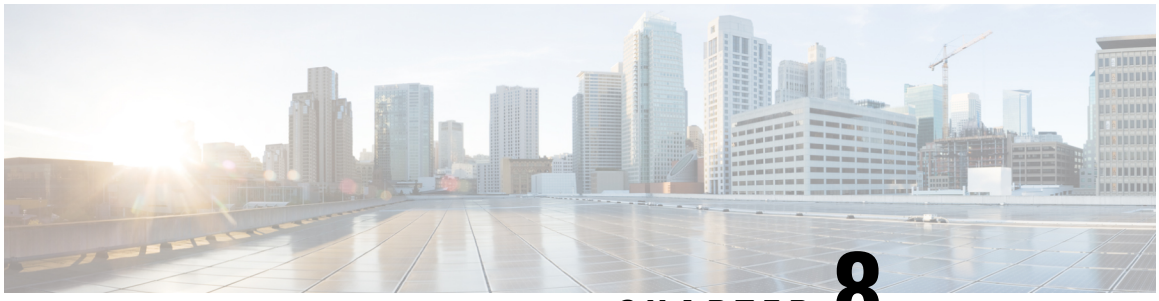
```
switch# show time-stamp hpt brief
Time-stamp HPT Device ID : 100
Timestamp HPT port status
-----------------------
Port State
------- --------------
Eth1/47 hpt enabled
```

In the following example, you can see that the HPT UTC time-stamping is *enabled*.

```
switch# sh time-stamp hpt brief
Time-stamp HPT Device ID : 0
Time-stamp HPT UTC Timestamp Enabled : enabled
Timestamp HPT port status
-----------------------
Port State
------- --------------
Eth1/4 hpt enabled
```

In the following example, you can see that the HPT UTC time-stamping is *disabled*.

```
switch# sh time-stamp hpt brief
Time-stamp HPT Device ID : 0
Time-stamp HPT UTC Timestamp Enabled : disabled
Timestamp HPT port status
-----------------------
Port State
------- --------------
Eth1/4 hpt enabled
```



CHAPTER 8

Configuring SPAN

This chapter describes how to configure an Ethernet switched port analyzer (SPAN) to analyze traffic between ports on Cisco NX-OS devices.

About SPAN

SPAN analyzes all traffic between source ports by directing the SPAN session traffic to a destination port with an external analyzer attached to it.

You can define the sources and destinations to monitor in a SPAN session on the local device.

SPAN Sources

The interfaces from which traffic can be monitored are called SPAN sources. Sources designate the traffic to monitor. SPAN sources include the following:

- Ethernet ports
- Port-channels

A single SPAN session can include mixed sources in any combination of the above.

Characteristics of SPAN source ports:

- A port configured as a source port cannot be configured as a destination port.

SPAN Destinations

SPAN destinations refer to the interfaces that monitor source ports. Destination ports receive the copied traffic from SPAN sources. SPAN destinations include the following:

- Ethernet ports in either access or trunk mode
- Port channels in either access or trunk mode

Characteristics of SPAN destination ports:

- A port configured as a destination port cannot also be configured as a source port.
- A destination port can be configured in only one SPAN session at a time.
- Destination ports do not participate in any spanning tree instance. SPAN output includes bridge protocol data unit (BPDU) Spanning Tree Protocol hello packets.

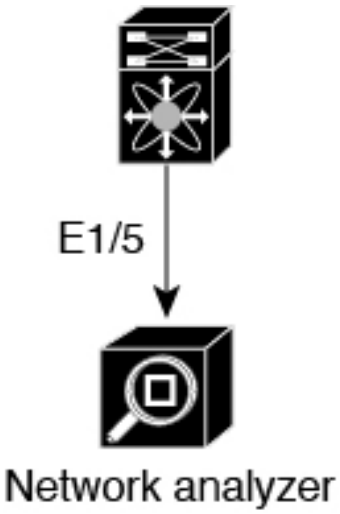
SPAN Sessions

You can create SPAN sessions to designate sources and destinations to monitor.

See the *Cisco Nexus 3550-T NX-OS Verified Scalability Guide* for information on the number of supported SPAN sessions.

This figure shows a SPAN configuration. Packets on two ethernet ports are copied to destination port, ethernet 1/5. Only traffic in the direction specified is copied.

Figure 2: SPAN Configuration



Source Port	Direction	Destination Ports
1/2	Rx	1/5
1/3	Rx	1/5

504564

High Availability

The SPAN feature supports stateless and stateful restarts. After a reboot, the running configuration is applied.

Guidelines and Limitations

SPAN has the following configuration guidelines and limitations:

- Traffic that is denied by an ACL may still reach the SPAN destination port because SPAN replication is performed on the ingress side prior to the ACL enforcement (ACL dropping traffic).
- Only ingress SPAN is supported.
- For SPAN session limits, see the *Cisco Nexus 3550-T NX-OS Verified Scalability Guide*.
- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.
- You can configure a SPAN session on the local device only.
- Packets with FCS errors are not mirrored in a SPAN session.
- You can configure only one destination port in a SPAN session.
- You can configure a destination port for only one SPAN session at a time.
- You cannot configure a port as both a source and destination port.
- Spanned packets will reflect the ingress rewrites such as, vlan tag removal, destination-mac rewrite on routed packets. Also, the span output packets are always untagged.
- Enabling UniDirectional Link Detection (UDLD) on the SPAN source and destination ports simultaneously is not supported. If UDLD frames are expected to be captured on the source port of such SPAN session, disable UDLD on the destination port of the SPAN session.
- SPAN is supported in layer 2 and layer 3 mode.
- SPAN is not supported for management ports.
- SPAN MTU is not supported.
- VLAN SPAN and VLAN ACL are not supported.
- Cisco NX-OS does not span Link Layer Discovery Protocol (LLDP) or Link Aggregation Control Protocol (LACP) packets when the source interface is not a host interface port channel.

Prerequisites for SPAN

SPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired SPAN configuration. For more information, see the Cisco Nexus 3550-T NX-OS Interfaces Configuration Guide.

Default Settings for SPAN

The following table lists the default settings for SPAN parameters.

Parameters	Default
SPAN sessions	Created in the shut state

Configuring a SPAN Session

You can configure a SPAN session on the local device only. By default, SPAN sessions are created in the shut state.

Note For bidirectional traditional sessions, you can configure the sessions without specifying the direction of the traffic.

Before you begin

You must configure the destination ports in access or trunk mode.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *ethernet slot/port*
	Example:
	`switch(config)# interface ethernet 1/5`
	`switch(config-if)#`
		Enters interface configuration mode on the selected slot and port.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures switchport parameters for the selected slot and port or range of ports.
Step 4	**switchport monitor** **Example:** `switch(config-if)# switchport monitor`	Configures the switchport interface as a SPAN destination.
Step 5	(Optional) Repeat Steps 2 through 4 to configure monitoring on additional SPAN destinations.	—
Step 6	**no monitor session** *session-number* **Example:** `switch(config)# no monitor session 3`	Clears the configuration of the specified SPAN session. The new session configuration is added to the existing session configuration.

	Command or Action	Purpose

Step 7	**monitor session** *session-number*[**rx**] [**shut**]
	Example:
	`switch(config)# monitor session 3 rx`
	`switch(config-monitor)#`
	Example:
	`switch(config)# monitor session 3 shut`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. The new session configuration is added to the existing session configuration. By default, the session is created in the shut state, and the session is a local SPAN session. The optional keyword shut specifies a shut state for the selected session.

Step 8	**description** *description* **Example:** `switch(config-monitor)# description my_span_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.
Step 9	**source** {**interface** *type* [**rx** **Example:** `switch(config-monitor)# source interface ethernet 1/3 rx`	Configures sources and the traffic direction in which to copy packets. You can enter a range of Ethernet ports or a port channel. You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress (rx), egress (tx), or both. For a unidirectional session, the direction of the source must match the direction specified in the session.
Step 10	(Optional) Repeat Step 9 to configure all SPAN sources.	
Step 11	Required: **destination interface** *type slot*/*port* **Example:** `switch(config-monitor)# destination interface ethernet 1/5`	Configures a destination for copied source packets. **Note** The SPAN destination port must be either an access port or a trunk port. **Note** You must enable monitor mode on the destination port.
Step 12	Required: **no shut** **Example:** `switch(config-monitor)# no shut`	Enables the SPAN session. By default, the session is created in the shut state.
Step 13	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config-monitor)# show monitor session 3`	Displays the SPAN configuration.

	Command or Action	Purpose
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Shutting Down or Resuming a SPAN Session

You can shut down SPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, SPAN sessions are created in the shut state.

You can resume (enable) SPAN sessions to resume the copying of packets from sources to destinations. In order to enable a SPAN session that is already enabled but operationally down, you must first shut it down and then enable it.

You can configure the shut and enabled SPAN session states with either a global or monitor configuration mode command.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified SPAN sessions. By default, sessions are created in the shut state. The **no** form of the command resumes (enables) the specified SPAN sessions. By default, sessions are created in the shut state. **Note** If a monitor session is enabled but its operational status is down, to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.

Step 3	**monitor session** *session-number*
	Example:
	`switch(config)# monitor session 3`
	`switch(config-monitor)#`
		Enters the monitor configuration mode. The new session configuration is added to the existing session configuration.

Step 4	[**no**] **shut** **Example:**	Shuts down the SPAN session. By default, the session is created in the shut state.

	Command or Action	Purpose
	`switch(config-monitor)# shut`	The **no** form of the command enables the SPAN session. By default, the session is created in the shut state.
Step 5	(Optional) **show monitor** **Example:** `switch(config-monitor)# show monitor`	Displays the status of SPAN sessions.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying SPAN Configurations

To display the SPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the SPAN session configuration.

Configuration Examples

This section contains the following configuration examples:

Configuration Example for a SPAN Session

To configure a SPAN session:

1. Configure destination ports in access mode and enable SPAN monitoring.

```
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

2. Configure a SPAN session.

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 1/9 rx
switch(config-monitor)# source interface port-channel 2 rx
switch(config-monitor)# destination interface ethernet 1/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
```