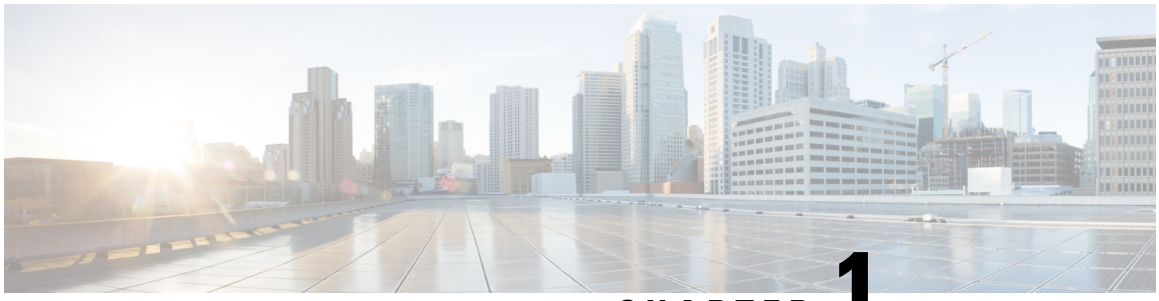
Cisco Catalyst IE 3200 and 3300 Rugged Series, Cisco IOS XE 16.11.1 Software Configuration Guide

First Published: 2019-04-03

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

CONTENTS



CHAPTER 1

Configuring Precision Time Protocol

Information About Precision Time Protocol

Precision Time Protocol (PTP) is defined in IEEE 1588 as Precision Clock Synchronization for Networked Measurements and Control Systems, and was developed to synchronize the clocks in packet-based networks that include distributed device clocks of varying precision and stability. PTP is designed specifically for industrial, networked measurement and control systems, and is optimal for use in distributed systems because it requires minimal bandwidth and little processing overhead.

Why PTP?

Smart grid power automation applications such as peak-hour billing, virtual power generators, and outage monitoring and management, require extremely precise time accuracy and stability. Timing precision improves network monitoring accuracy and troubleshooting ability.

In addition to providing time accuracy and synchronization, the PTP message-based protocol can be implemented on packet-based networks, such as Ethernet networks. The benefits of using PTP in an Ethernet network include:

- Low cost and easy setup in existing Ethernet networks
- Limited bandwidth is required for PTP data packets

Ethernet Switches and Delays

In an Ethernet network, switches provide a full-duplex communication path between network devices. Switches send data packets to packet destinations using address information contained in the packets. When the switch attempts to send multiple packets simultaneously, some of the packets are buffered by the switch so that they are not lost before they are sent. When the buffer is full, the switch delays sending packets. This delay can cause device clocks on the network to lose synchronization with one another.

Additional delays can occur when packets entering a switch are stored in local memory while the switch searches the MAC address table to verify packet CRC fields. This process causes variations in packet forwarding time latency, and these variations can result in asymmetrical packet delay times.

Adding PTP to a network can compensate for these latency and delay problems by correctly adjusting device clocks so that they stay synchronized with one another. PTP enables network switches to function as PTP devices, including boundary clocks (BCs) and transparent clocks (TCs).

Note To learn more about PTP clock devices and their role in a PTP network, refer to PTP Clocks, on page 6.

Message-Based Synchronization

To ensure clock synchronization, PTP requires an accurate measurement of the communication path delay between the time source (*master*) and the receiver (*slave*). PTP sends messages between the master and slave device to determine the delay measurement. Then, PTP measures the exact message transmit and receive times and uses these times to calculate the communication path delay. PTP then adjusts current time information contained in network data for the calculated delay, resulting in more accurate time information.

This delay measurement principle determines path delay between devices on the network, and the local clocks are adjusted for this delay using a series of messages sent between masters and slaves. The one-way delay time is calculated by averaging the path delay of the transmit and receive messages. This calculation assumes a symmetrical communication path; however, switched networks do not necessarily have symmetrical communication paths, due to the buffering process.

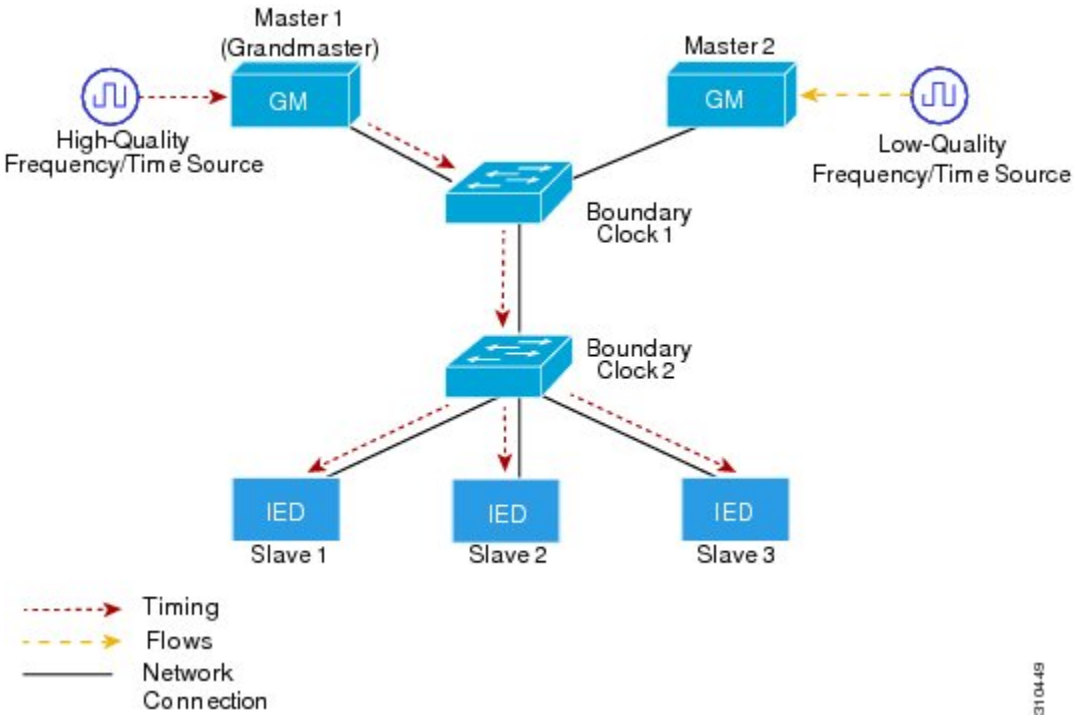
PTP provides a method, using transparent clocks, to measure and account for the delay in a time-interval field in network timing packets, making the switches temporarily transparent to the master and slave nodes on the network. An end-to-end transparent clock forwards all messages on the network in the same way that a switch does.

Note Cisco PTP supports multicast PTP messages only.

To read a detailed description of synchronization messages, refer to PTP Event Message Sequences, on page 3. To learn more about how transparent clocks calculate network delays, refer to Transparent Clock, on page 6.

The following figure shows a typical 1588 PTP network that includes grandmaster clocks, switches in boundary clock mode, and Intelligent Electronic Device (IEDs) such as a digital relays or protection devices. In this diagram, Master 1 is the grandmaster clock. If Master 1 becomes unavailable, the boundary clock slaves switch to Master 2 for synchronization.

Figure 1: PTP Network



PTP Event Message Sequences

This section describes the PTP event message sequences that occur during synchronization.

Synchronizing with Boundary Clocks

The ordinary and boundary clocks configured for the delay request-response mechanism use the following event messages to generate and communicate timing information:

- Sync
- Delay_Req
- Follow_Up
- Delay_Resp

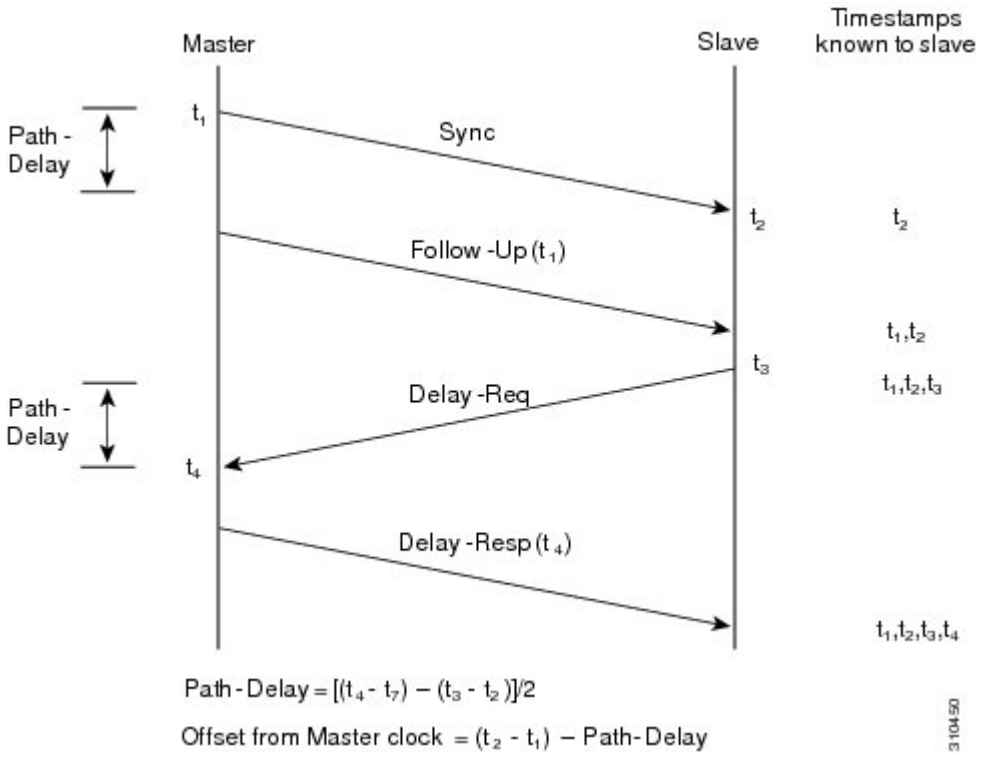
These messages are sent in the following sequence:

1. The master sends a Sync message to the slave and notes the time (t1) at which it was sent.
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 t1 in a Follow_Up 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.

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 offset calculation is based on the assumption that the time for the message to propagate from master to slave is the same as the time required from slave to master. This assumption is not always valid on an Ethernet network due to asymmetrical packet delay times.

Figure 2: Detailed Steps—Boundary Clock Synchronization



Synchronizing with Peer-to-Peer Transparent Clocks

When the network includes multiple levels of boundary clocks in the hierarchy, with non-PTP enabled devices between them, synchronization accuracy decreases.

The round-trip time is assumed to be equal to mean_path_delay/2, however this is not always valid for Ethernet networks. To improve accuracy, the resident time of each intermediary clock is added to the offset in the end-to-end transparent clock. Resident time, however, does not take into consideration the link delay between peers, which is handled by peer-to-peer transparent clocks.

Peer-to-peer transparent clocks measure the link delay between two clock ports implementing the peer delay mechanism. The link delay is used to correct timing information in Sync and Follow_Up messages.

Peer-to-peer transparent clocks use the following event messages:

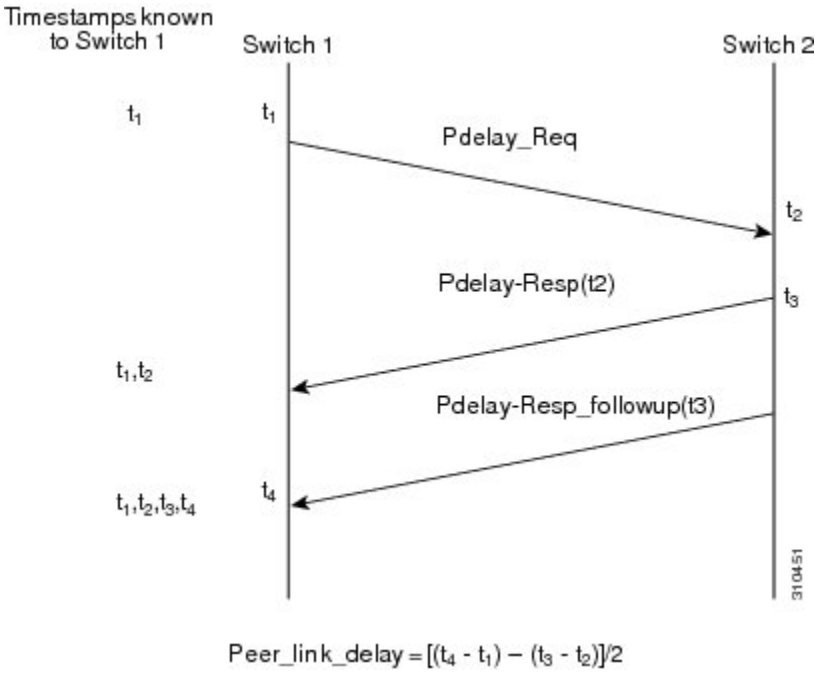
- Pdelay_Req
- Pdelay_Resp
- Pdelay_Resp_Follow_Up

These messages are sent in the following sequence:

1. Port 1 generates timestamp t1 for a Pdelay_Req message.
2. Port 2 receives and generates timestamp t2 for this message.
3. Port 2 returns and generates timestamp t3 for a Pdelay_Resp message.

   To minimize errors due to any frequency offset between the two ports, Port 2 returns the Pdelay_Resp message as quickly as possible after the receipt of the Pdelay_Req message.
4. Port 2 returns timestamps t2 and t3 in the Pdelay_Resp and Pdelay_Resp_Follow_Up messages respectively.
5. Port 1 generates timestamp t4 after receiving the Pdelay_Resp message. Port 1 then uses the four timestamps (t1, t2, t3, and t4) to calculate the mean link delay.

Figure 3: Detailed Steps—Peer-to-Peer Transparent Clock Synchronization



Synchronizing the Local Clock

In an ideal PTP network, the master and slave clock operate at the same frequency. However, *drift* can occur on the network. Drift is the frequency difference between the master and slave clock. You can compensate for drift by using the time stamp information in the device hardware and follow-up messages (intercepted by the switch) to adjust the frequency of the local clock to match the frequency of the master clock.

Best Master Clock Algorithm

The Best Master Clock Algorithm (BMCA) is the basis of PTP functionality. The BMCA specifies how each clock on the network determines the best master clock in its subdomain of all the clocks it can see, including itself. The BMCA runs on the network continuously and quickly adjusts for changes in network configuration.

The BMCA uses the following criteria to determine the best master clock in the subdomain:

- Clock quality (for example, GPS is considered the highest quality)
- Clock accuracy of the clock's time base
- Stability of the local oscillator
- Closest clock to the grandmaster

In addition to identifying the best master clock, the BMCA also ensures that clock conflicts do not occur on the PTP network by ensuring that:

- Clocks do not have to negotiate with one another
- There is no misconfiguration, such as two master clocks or no master clocks, as a result of the master clock identification process

PTP Clocks

A PTP network is made up of PTP-enabled devices and devices that are not using PTP. The PTP-enabled devices typically consist of the following clock types.

Grandmaster Clock

Within a PTP domain, the grandmaster clock is the primary source of time for clock synchronization using PTP. The grandmaster clock usually has a very precise time source, such as a GPS or atomic clock. When the network does not require any external time reference and only needs to be synchronized internally, the grandmaster clock can free run.

Ordinary Clock

An ordinary clock is a PTP clock with a single PTP port. It functions as a node in a PTP network and can be selected by the BMCA as a master or slave within a subdomain. Ordinary clocks are the most common clock type on a PTP network because they are used as end nodes on a network that is connected to devices requiring synchronization. Ordinary clocks have various interface to external devices.

Boundary Clock

A boundary clock in a PTP network operates in place of a standard network switch or router. Boundary clocks have more than one PTP port, and each port provides access to a separate PTP communication path. Boundary clocks provide an interface between PTP domains. They intercept and process all PTP messages, and pass all other network traffic. The boundary clock uses the BMCA to select the best clock seen by any port. The selected port is then set as a slave. The master port synchronizes the clocks connected downstream, while the slave port synchronizes with the upstream master clock.

Transparent Clock

The role of transparent clocks in a PTP network is to update the time-interval field that is part of the PTP event message. This update compensates for switch delay and has an accuracy of within one picosecond.

There are two types of transparent clocks:

End-to-end (E2E) transparent clocks measure the PTP event message transit time (also known as *resident time*) for SYNC and DELAY_REQUEST messages. This measured transit time is added to a data field (correction field) in the corresponding messages:

- The measured transit time of a SYNC message is added to the correction field of the corresponding SYNC or the FOLLOW_UP message.
- The measured transit time of a DELAY_REQUEST message is added to the correction field of the corresponding DELAY_RESPONSE message.

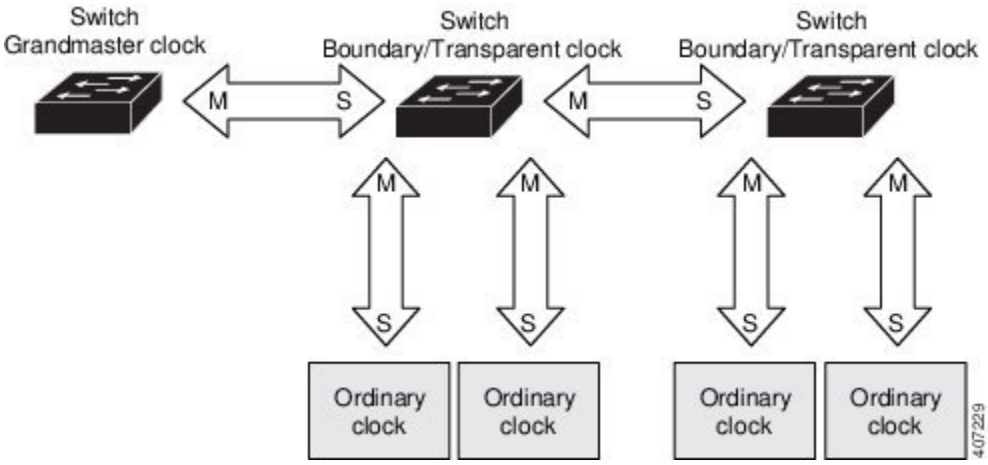
The slave uses this information when determining the offset between the slave's and the master's time. E2E transparent clocks do not provide correction for the propagation delay of the link itself.

Peer-to-peer (P2P) transparent clocks measure PTP event message transit time in the same way E2E transparent clocks do, as described above. In addition, P2P transparent clocks measure the upstream link delay. The upstream link delay is the estimated packet propagation delay between the upstream neighbor P2P transparent clock and the P2P transparent clock under consideration.

These two times (message transit time and upstream link delay time) are both added to the correction field of the PTP event message, and the correction field of the message received by the slave contains the sum of all link delays. In theory, this is the total end-to-end delay (from master to slave) of the SYNC packet.

The following figure illustrates PTP clocks in a master-slave hierarchy within a PTP network.

Figure 4: PTP Clock Hierarchy



PTP Profiles

This section describes the following PTP profiles available on the switch:

- Power Profile
- Default Profile
- 802.1AS Profile (IE 4000 only)

The Power Profile is defined in PC37.238 - IEEE Draft Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications. This switch documentation uses the terms Power Profile mode and Default Profile mode when referring to this IEEE 1588 profile and its associated configuration values.

The IEEE 1588 definition of a PTP profile is *the set of allowed PTP features applicable to a device*. A PTP profile is usually specific to a particular type of application or environment and defines the following values:

- Best master clock algorithm options

- Configuration management options
- Path delay mechanisms (peer delay or delay request-response)
- Range and default values of all PTP configurable attributes and data set members
- Transport mechanisms that are required, permitted, or prohibited
- Node types that are required, permitted, or prohibited
- Options that are required, permitted, or prohibited

Default Profile Mode

The default PTP profile mode on the switch is Default Profile mode. In this mode:

- The PTP mode of transport is Layer 3.
- The supported transparent clock mode is end-to-end (E2E).

Table 1: Configuration Values for the IEEE PTP Power Profile and Switch Modes , on page 8 lists the configuration values for the switch in Default Profile mode.

Power Profile Mode

The IEEE Power Profile defines specific or allowed values for PTP networks used in power substations. The defined values include the optimum physical layer, the higher level protocol for PTP messages, and the preferred best master clock algorithm. The Power Profile values ensure consistent and reliable network time distribution within substations, between substations, and across wide geographic areas.

The switch is optimized for PTP in these ways:

- Hardware—The switch uses FPGA and PHY for the PTP function. The PHY time stamps the Fast Ethernet and Gigabit Ethernet ports.
- Software—In Power Profile mode, the switch uses the configuration values defined in the IEEE 1588 Power Profile standard.

The following table lists the configuration values defined by the IEEE 1588 Power Profile and the values that the switch uses for each PTP profile mode.

Table 1: Configuration Values for the IEEE PTP Power Profile and Switch Modes

PTP Field	Power Profile Value	Switch Configuration Value	
		Power Profile Mode	**Default Profile Mode**
Message transmission	Ethernet 802.3 with Ethertype 0X88F7. PTP messages are sent as 802.1Q tagged Ethernet frames with a default VLAN 0 and default priority 4.	**Access Ports**–Untagged Layer 2 packets. **Trunk Ports**–802.1Q tagged Layer 2 packets with native VLAN on the port and default priority value of 4.	Layer 3 packets. By default, 802.1q tagging is disabled.
MAC address–Non-peer delay messages	01-1B-19-00-00-00.	01-1B-19-00-00-00.	01-1B-19-00-00-00.

PTP Field	Power Profile Value	Switch Configuration Value	
		Power Profile Mode	Default Profile Mode
MAC address–Peer delay messages	01-80-C2-00-00-0E.	01-80-C2-00-00-0E.	Not applicable to this mode.
Domain number	0.	0.	0.
Path delay calculation	Peer-to-peer transparent clocks.	Peer-to-peer transparent clocks using the peer_delay mechanism.	End-to-end transparent clocks using the delay_request mechanism.
BMCA	Enabled.	Enabled.	Enabled.
Clock type	Two-step clocks are supported.	Two-step.	Two-step.
Time scale	Epoch.[1]	Epoch.	Epoch.
Grandmaster ID and local time determination	PTP-specific TLV (type, length, value) to indicate Grandmaster ID.	PTP-specific TLV to indicate Grandmaster ID.	PTP-specific type, length, and value to indicate Grandmaster ID.
Time accuracy over network hops	Over 16 hops, slave device synchronization accuracy is within 1 usec (1 microsecond).	Over 16 hops, slave device synchronization accuracy is within 1 usec (1 microsecond).	Not applicable in this mode.

[1] Epoch = Elapsed time since epoch start.

802.1AS Profile (IE 4000 only)

Note The 802.1AS Profile is supported for the IE 4000 only.

The IEEE 802.1AS standard "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks" specifies the protocol and procedures used to ensure that synchronization requirements are met for time-sensitive applications across bridged and virtual bridged local area networks.

802.1AS specifies the use of IEEE 1588 (PTP) specifications where applicable in the context of IEEE Std 802.1D -2004 and IEEE Std 802.1Q -2005.1. The 802.1AS standard is one of three 802.1 AVB draft standards. 802.1AS over Ethernet (802.3) qualifies as a Profile of IEEE 1588-2008. It simplifies IEEE 1588 and defines synchronization over different types of media.

Key characteristics of 802.1AS are:

- For Ethernet full-duplex links, it uses the peer delay mechanism.
- All switches in the domain need to be 802.1AS capable.
- Transportation of 802.1AS packets is L2 multicast only, with no VLAN tag.
- It requires two-step processing (use of Follow_Up and Pdelay_Resp_Follow_Up messages to communicate timestamps).

- There is only a single active grandmaster in a time-aware network. That is, there is only a single 802.1AS domain.
- The BMCA (Best Master Clock Algorithm) is same as that used in IEEE 1588 with the following exceptions:
  - Announce messages received on a slave port that were not sent by the receiving time-aware system are used immediately; that is, there is no foreign-master qualification.
  - A port that the BMCA determines should be a master port enters the master state immediately; that is, there is no pre-master state.
  - The uncalibrated state is not needed and therefore not used.
  - All time-aware systems are required to participate in best master selection (even if the system is not grandmaster capable).

802.1AS on the IE 4000

On the IE 4000, 802.1AS is used in the Time Sensitive Network (TSN) feature. However, as a precise timing distribution mechanism, 802.1AS runs by itself without TSN configuration or inputs. The 802.1AS feature software implementation is based on the existing time stamping functionality of FPGA and has no new requirement on hardware beyond other PTP profiles.

The end-to-end time-synchronization performance of 802.1AS on the IE 4000 is as follows:

- Any two time-aware systems separated by six or fewer time-aware systems (that is, seven or fewer hops) will be synchronized to within 1 μs peak-to-peak of each other during steady-state operation.
- Performance beyond 7 hops is not defined.

802.1AS Profile Comparison

Table 2: Comparison of PTP Profiles on IE Switches

Profile	**Default (*)**		**Power**		**802.1AS**
Standard	IEEE1588 v2 (J.3)		IEEE C37.238		IEEE802.1AS
Mode	Boundary	End-to-End transparent	Boundary	Peer-to-Peer transparent	**
Path Delay	Delay req/res	Delay req/res	Peer delay req/res	Peer delay req/res	Peer delay req/res
Non-PTP device allowed in PTP domain	Yes	Yes	No	No	No
Transport	UDP over IP (multicast and unicast)		L2 Multicast		L2 Multicast

* Delay Request-Response Default PTP profile (as defined in IEEE1588 J.3).

** There is no mode setting for 802.1AS. Mathematically it is equivalent to P2P transparent, but it works differently from a transparent clock.

Tagging Behavior for PTP Packets

The following table describes the switch tagging behavior in Power Profile and Default Profile modes.

Table 3: Tagging Behavior for PTP Packets

Switch Port Mode	Configuration	Power Profile Mode		Default Profile Mode	
		Behavior	**Priority**	**Behavior**	**Priority**
Trunk Port	**vlan dot1q tag native** enabled	Switch tags packets	7	Switch tags packets	7
Trunk Port	**vlan dot1q tag native** disabled	PTP software tags packets	4	Untagged	None
Access Port	N/A	Untagged	None	Untagged	None

PTP Clock Modes Supported on the Switch

PTP synchronization behavior depends on the PTP clock mode that you configure on the switch. You can configure the switch for one of the following global modes.

See Guidelines and Limitations, on page 18 for guidelines for configuring each of the clock modes.

Boundary Clock Mode

A switch configured for boundary clock mode participates in selecting the best master clock on the subdomain, selecting from all clocks it can see, including itself. If the switch does not detect a more accurate clock than itself, then the switch becomes the master clock. If a more accurate clock is detected, then the switch synchronizes to that clock and becomes a slave clock.

After initial synchronization, the switch and the connected devices exchange PTP timing messages to correct the changes caused by clock offsets and network delays.

Forward Mode

A switch configured for forward mode passes incoming PTP packets as normal multicast traffic.

E2E Transparent Clock Mode

A switch configured for end-to-end transparent clock mode does not synchronize its clock with the master clock. A switch in this mode does not participate in master clock selection and uses the default PTP clock mode on all ports.

P2P Transparent Clock Mode

A switch configured for peer-to-peer transparent clock mode does not synchronize its clock with the master clock. A switch in this mode does not participate in master clock selection and uses the default PTP clock mode on all ports.

Configurable Boundary Clock Synchronization Algorithm

You can configure the BC synchronization algorithm to accommodate various PTP use cases, depending on whether you need to prioritize filtering of input time errors or faster convergence. A PTP algorithm that filters packet delay variation (PDV) converges more slowly than a PTP algorithm that does not.

By default, the BC uses a linear feedback controller (that is, a servo) to set the BC's time output to the next clock. The linear servo provides a small amount of PDV filtering and converges in an average amount of time. For improved convergence time, BCs can use the TC feedforward algorithm to measure the delay added by the network elements forwarding plane (the disturbance) and use that measured delay to control the time output.

While the feedforward BC dramatically speeds up the boundary clock, the feedforward BC does not filter any PDV. The adaptive PDV filter provides high quality time synchronization in the presence of PDV over wireless access points (APs) and enterprise switches that do not support PTP and that add significant PDV.

Three options are available for BC synchronization (all are compliant with IEEE 1588-2008):

- Feedforward—For very fast and accurate convergence; no PDV filtering.
- Adaptive—Filters as much PDV as possible, given a set of assumptions about the PDV characteristics, the hardware configuration, and the environmental conditions.

Note With the adaptive filter, the switch does not meet the time performance requirements specified in ITU-T G.8261.

- Linear—Provides simple linear filtering (the default).

Adaptive mode (**ptp transfer filter adaptive**) is not available in Power Profile mode.

For configuration information, see .

Information About NTP to PTP Time Conversion

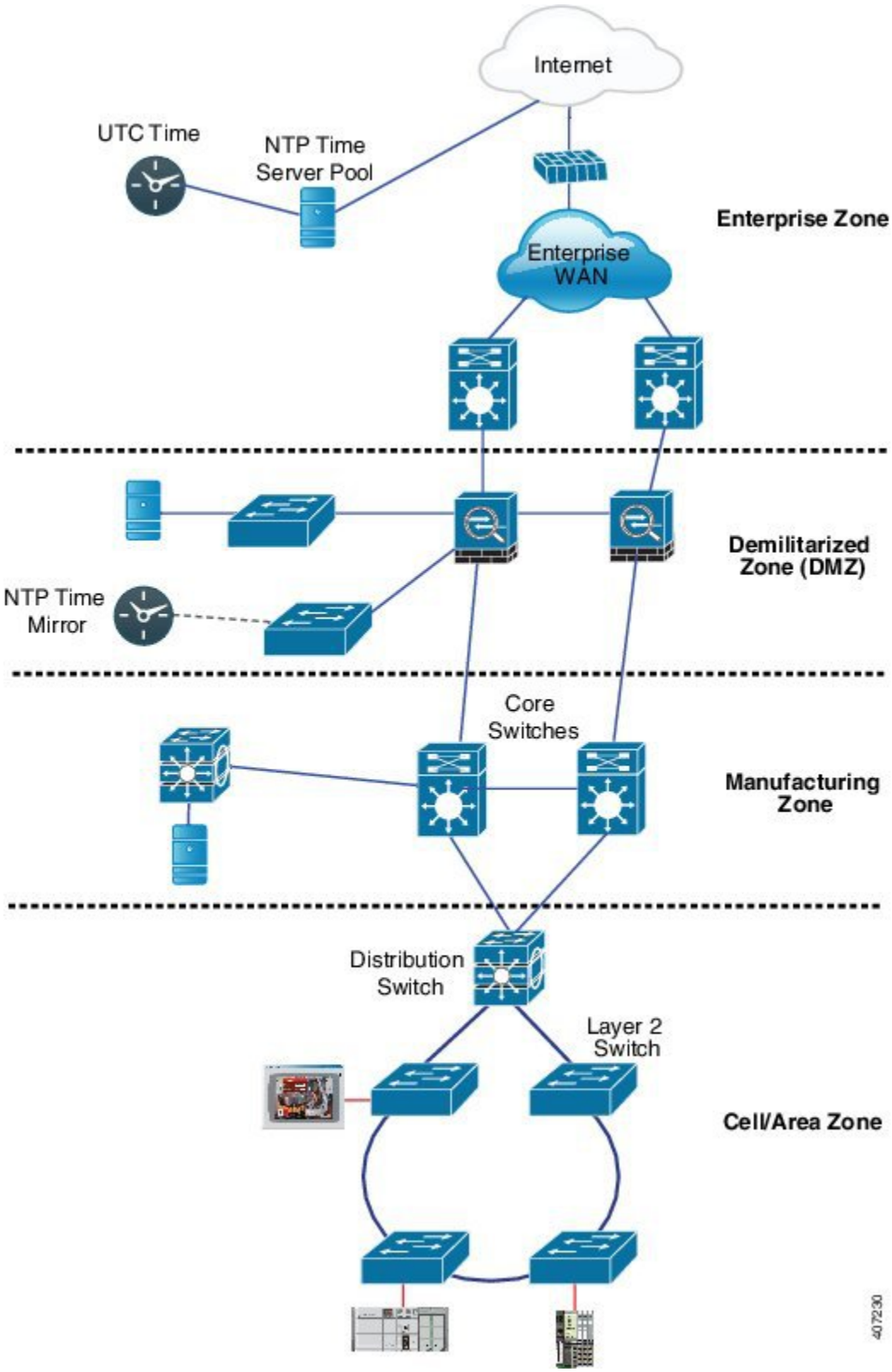
NTP to PTP Time Conversion allows you to use Network Time Protocol (NTP) as a time source for PTP. Customers who use PTP for very precise synchronization within a site can use NTP across sites, where precise synchronization is not required.

NTP is the traditional method of synchronizing clocks across packet based networks. NTP uses a two-way time transfer mechanism, between a master and a slave. NTP is capable of synchronizing a device within a few 100 milliseconds across the Internet, and within a few milliseconds in a tightly controlled LAN. The ability to use NTP as a time source for PTP allows customers to correlate data generated in their PTP network with data in their enterprise data centers running NTP.

The following figure shows an example of an industrial network based on the Industrial Automation and Control System Reference Model. The enterprise zone and demilitarized zone run NTP, and the manufacturing

zone and cell/area zone run PTP with NTP as the time source. The switch with the NTP to PTP conversion feature can be either the Layer 2 Switch or the Distribution Switch in the Cell/Area Zone.

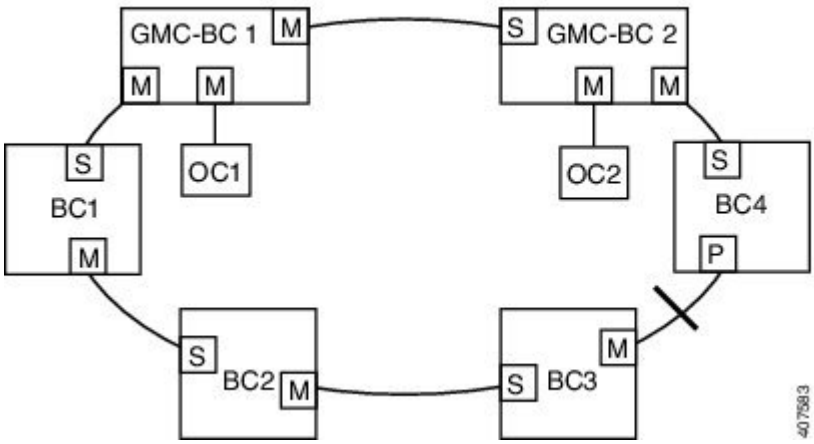
Figure 5: Industrial Network with NTP and PTP


Internet
UTC Time
NTP Time
Server Pool
Enterprise Zone
Enterprise
WAN
Demilitarized
Zone (DMZ)
NTP Time
Mirror
Core
Switches
Manufacturing
Zone
Distribution
Switch
Layer 2
Switch
Cell/Area Zone
407230

Grandmaster Boundary Clock Hybrid

The NTP to PTP conversion feature adds grandmaster clock functionality to Cisco PTP, so the switch can be a time source as well as forward time. A new PTP clock type, grandmaster boundary clock (GMC-BC), provides the NTP time source for PTP. The GMC-BC acts like a BC, which is a multi-port device, with a single-port GMC connected to a virtual port on the BC. The GMC-BC switches between acting like a GMC when the GMC-BC is the primary GMC, and acting like a BC when the GMC-BC is a backup. This ensures that all devices on the PTP network remain synchronized in a failover scenario. The following figure shows a PTP network with redundant GMC-BCs. GMC-BC 1 is the grandmaster clock, and GMC-BC 2 is both backup GMC and BC.

Figure 6: Redundant GMC-BC Configuration



In a network with two GMC-BCs, the secondary GMC-BC can synchronize to both the NTP reference and the PTP reference at the same time, so the secondary GMC-BC can immediately take over when the primary GMC-BC fails. The GMC-BC instantly updates the time during a switchover.

Clock Manager

The clock manager is the component in the Cisco NTP to PTP software architecture that keeps track of the various time services and selects the clock that actively provides time. The clock manager notifies the time services of important changes, such as state changes, leap seconds, or daylight saving time.

The clock manager selects the NTP or manually-set clock first, followed by PTP and the real-time clock if NTP is not active. The following table shows the results of the clock selection process.

Table 4: Time Service Selection

NTP (Active) or Manually Set	PTP (Active)	Real-Time Clock	Selected Output
True	Don’t care	Don’t care	NTP or Manually Set
False	True	Don’t care	PTP
False	False	True	Real-Time Clock

In general, the clock manager ensures that the time displayed in the Cisco IOS commands **show ptp clock** and **show clock** match. The **show clock** command always follows this priority, but there are two corner cases where the **show ptp clock** time may differ:

- The switch is either a TC or a BC, and there is no other active reference on the network. To preserve backwards compatibility, the TC and BC never take their time from the clock manager, only from the network's PTP GMC. If there is no active PTP GMC, then the time displayed in the **show clock** and the **show ptp clock** command output may differ.
- The switch is a syntonizing TC, a BC with a slave port, or a GMC-BC with slave port, and the time provided by the PTP GMC does not match the time provided by NTP or the user (that is, manually set). In this case, the PTP clock must forward the time from the PTP GMC. If the PTP clock does not follow the PTP GMC, then the PTP network will end up with two different time bases, which would break any control loops or sequence of event applications using PTP.

The following table shows how the Cisco IOS and PTP clocks behave given the various configurations. Most of the time, the two clocks match. Occasionally, the two clocks are different; those configurations are highlighted in the table.

Table 5: Expected Time Flow

IOS Clock Configuration	PTP Clock Configuration	IOS Clock Source	PTP Clock Source
Calendar	PTP BC, E2E TC, or GMC-BC in BC Mode	PTP	PTP
Manual	**PTP BC, E2E TC, or GMC-BC in BC Mode**	**Manual**	**PTP**
NTP	**PTP BC, E2E TC, or GMC-BC in BC Mode**	**NTP**	**PTP**
Calendar	GMC-BC in GM Mode	Calendar	Calendar
Manual	GMC-BC in GM Mode	Manual	Manual
NTP	GMC-BC in GM Mode	NTP	NTP

Prerequisites

- Review the Guidelines and Limitations, on page 18.
- To use the NTP to PTP conversion feature, the switch must have an IP address for NTP to function.
- To use the NTP to PTP conversion feature, you must configure at least one NTP server. Configuring three or more NTP servers allows NTP to ignore bad clocks.

Note For information about configuring NTP, see the section Configuring NTP in the *Catalyst 3750-X and 3560-X Switch Software Configuration Guide, Release 12.2(55)SE* .

Guidelines and Limitations

PTP Messages

- The Cisco PTP implementation supports only the two-step clock and not the one-step clock. If the switch receives a one-step message from the Grand Master Clock, it will convert it into a two-step message.
- Cisco PTP supports multicast PTP messages only.

PTP Mode and Profile

- The switch and the grandmaster clock must be in the same PTP domain.
- When Power Profile mode is enabled, the switch drops the PTP announce messages that do not include these two Type, Length, Value (TLV) message extensions: *Organization_extension* and *Alternate_timescale* .

  If the grandmaster clock is not compliant with PTP and sends announce messages without these TLVs, configure the switch to process the announce message by entering the **ptp allow-without-tlv** command.
- When the switch is in Power Profile mode, only the peer_delay mechanism is supported.

  To change to Boundary Clock Mode, on page 11 and the peer_delay mechanism, enter the **ptp mode boundary pdelay-req** command.
- To disable Power Profile mode and return the switch to E2E Transparent Clock Mode, on page 11, enter the **no ptp profile power** command.
- In Default Profile mode, only the delay_request mechanism is supported.

  To change to Boundary Clock Mode, on page 11 with the delay_request mechanism, enter the **ptp mode boundary delay-req** command.

Packet Format

- The packet format for PTP messages can be 802.1q tagged packets or untagged packets.
- The switch does not support 802.1q QinQ tunneling.
- In switch Power Profile mode:
  - When the PTP interface is configured as an access port, PTP messages are sent as untagged, Layer 2 packets.
  - When the PTP interface is configured as a trunk port, PTP packets are sent as 802.1q tagged Layer 2 packets over the port native VLAN.
- Slave IEDs must support tagged and untagged packets.
- When PTP packets are sent on the native VLAN in E2E Transparent Clock Mode, on page 11, they are sent as untagged packets. To configure the switch to send them as tagged packets, enter the global **vlan dot1q tag native** command.

VLAN Configuration

- Sets the PTP VLAN on a trunk port. The range is from 1 to 4094. The default is the native VLAN of the trunk port.
- In boundary mode, only PTP packets in PTP VLAN will be processed, PTP packets from other VLANs will be dropped.
- Before configuring the PTP VLAN on an interface, the PTP VLAN must be created and allowed on the trunk port.
- Most grandmaster clocks use the default VLAN 0. In Power Profile mode, the switch default VLAN is VLAN 1 and VLAN 0 is reserved. When you change the default grandmaster clock VLAN, it must be changed to a VLAN other than 0.
- When VLAN is disabled on the grandmaster clock, the PTP interface must be configured as an access port.

Clock Configuration

- All PHY PTP clocks are synchronized to the grandmaster clock. The switch system clock is not synchronized as part of PTP configuration and processes.
- When VLAN is enabled on the grandmaster clock, it must be in the same VLAN as the native VLAN of the PTP port on the switch.
- Grandmaster clocks can drop untagged PTP messages when a VLAN is configured on the grandmaster clock. To force the switch to send tagged packets to the grandmaster clock, enter the global **vlan dot1q tag native** command.

Clock Modes

Note The 802.1AS profile does not have a clock mode setting.

- Boundary Clock Mode
  - You can enable this mode when the switch is in Power Profile Mode, on page 8 (Layer 2) or in Default Profile Mode, on page 8 (Layer 3).
- Forward Mode
  - You can enable this mode when the switch is in Power Profile Mode, on page 8 (Layer 2) or in Default Profile Mode, on page 8 (Layer 3).
  - When the switch is in Forward mode, the only global configuration available is the CLI command to switch to a different PTP mode (that is, boundary, e2etransparent, or p2ptransparent).
- E2E Transparent Clock Mode
  - You can enable this mode only when the switch is in Default Profile Mode, on page 8 (Layer 3).
  - When the switch is in E2E Transparent mode, the only global configuration available is the CLI command to switch to a different PTP mode (that is, boundary, p2ptransparent, or forward).

- P2P Transparent Clock Mode
  - You can enable this mode only when the switch is in Power Profile Mode, on page 8 (Layer 2).
  - When the switch is in P2P Transparent mode, the only global configuration available is the CLI command to switch to a different PTP mode (that is, boundary, e2etransparent, or forward).

PDV Filtering

Adaptive mode (**ptp transfer filter adaptive**) is not available in Power Profile mode.

PTP Interaction with Other Features

- The following PTP clock modes do not support EtherChannels:
  - e2etransparent
  - p2ptransparent
  - boundary
- The following PTP clock modes only operate on a single VLAN:
  - e2etransparent
  - p2ptransparent

Default Settings

- PTP is enabled on the switch by default.
- By default, the switch uses configuration values defined in the Default Profile (Default Profile mode is enabled).
- The switch default PTP clock mode is E2E Transparent Clock Mode, on page 11.
- The default BC synchronization algorithm is linear filter.

Configuring PTP on the Switch

Use one of the following procedures in this section to configure the switch for PTP.

Note To configure the switch for grandmaster-boundary clock mode (gmc-bc), see Configuring NTP to PTP Time Conversion, on page 31.

Configuring PTP Power Profile Mode on the Switch

This section describes how to configure the switch to use the PTP Power Profile and operate in Power Profile mode.

Before you begin

These are some guidelines for configuring the Power Profile on the switch:

- When you enter **no** with PTP port configuration commands, the specified port property is set to the default value.
- To determine the value in seconds for the ptp global command *interval* variable, use a logarithmic scale. Below are examples of the *interval* variable value converted to seconds with a logarithmic scale:

Value Entered	Logarithmic Calculation	Value in Seconds
-1	2^{-1}	1/2
0	2^{0}	1

SUMMARY STEPS

1. Enter global configuration mode:
2. Set the Power Profile:
3. Specify the synchronization clock mode:
4. (Optional, BC and TC mode) Specify TLV settings:
5. (Optional, BC and TC mode) Specify the PTP clock domain:
6. (Optional, BC and TC mode) Specify the packet priority:
7. (Optional, BC mode only) Specify the BMCA priority:
8. (Optional, BC mode only) Specify time-property preservation:
9. (Optional, BC mode only) Specify the BC synchronization algorithm:
10. (Optional) Enter interface configuration mode:
11. (Optional) Specify port settings:
12. Return to privileged EXEC mode:
13. Verify your entries:
14. (Optional) Save your entries in the configuration file:

DETAILED STEPS

Step 1 Enter global configuration mode:

configure terminal

Step 2 Set the Power Profile:

ptp profile power

Step 3 Specify the synchronization clock mode:

ptp mode {boundary pdelay-req | p2ptransparent | forward}

- **mode boundary pdelay-req**—Configures the switch for boundary clock mode using the delay-request mechanism. In this mode, the switch participates in the selection of the most accurate master clock. Use this mode when overload or heavy load conditions produce significant delay jitter.
- **mode p2ptransparent**—Configures the switch for peer-to-peer transparent clock mode and synchronizes all switch ports with the master clock. The link delay time between the participating PTP ports and the message transit time is added to the resident time. Use this mode to reduce jitter and error accumulation. This is the default in Power Profile mode.
- **mode forward**—Configures the switch to pass incoming PTP packets as normal multicast traffic.

Step 4 (Optional, BC and TC mode) Specify TLV settings:

ptp allow-without-tlv

Step 5 (Optional, BC and TC mode) Specify the PTP clock domain:

ptp domain *domain-number*

domain-number—A number from 0 to 255.

The participating grandmaster clock, switches, and slave devices should be in the same domain.

Step 6 (Optional, BC and TC mode) Specify the packet priority:

ptp packet *priority*

The PTP packets have a default priority of 4. Lower values take precedence.

Step 7 (Optional, BC mode only) Specify the BMCA priority:

ptp priority1 *priority* **priority2** *priority*

- **priority1** *priority*—Overrides the default criteria (such as clock quality and clock class) for the most accurate master clock selection.
- **priority2** *priority*—Breaks the tie between two switches that match the default criteria. For example, enter 2 to give a switch priority over identical switches.*priority* —A priority number from 0 to 255. The default is 128.

Step 8 (Optional, BC mode only) Specify time-property preservation:

ptp time-property persist {*value* | **infinite**}

- *value*—Time duration, in seconds, from 0-100000. The default is 300.
- **infinite**—Time properties are preserved indefinitely.

Preserving the time properties prevents slave clocks from detecting a variance in the time values when the redundant GMC comes out of standby.

Step 9 (Optional, BC mode only) Specify the BC synchronization algorithm:

ptp transfer {**feedforward** | **filter linear**}

- **feedforward**—Very fast and accurate. No PDV filtering.
- **filter linear**—Provides a simple linear filter (default).

Step 10 (Optional) Enter interface configuration mode:

interface *interface-id*

Step 11 (Optional) Specify port settings:

Boundary pdelay-req mode:

ptp {**announce** {**interval** *value* | **timeout** *value*} | **pdelay-req interval** *value* | **enable** | **sync** {**interval** *value* | **limit** *value*} | **vlan** *value*}

p2ptransparent mode:

ptp {**pdelay-req interval** *value* | **enable** | **sync limit** *value* | **vlan** *value*}

- **announce interval** *value*—Sets the logarithmic mean interval in seconds to send announce messages. The range is 0 to 4. The default is 1 (2 seconds).
- **announce timeout** *value*—Sets the logarithmic mean interval in seconds to announce timeout messages. The range is 2 to 10. The default is 3 (8 seconds).
- **pdelay-req interval** *value*—Sets the logarithmic mean interval in seconds for slave devices to send pdelay request messages when the port is in the master clock state. The range is -3 to 5. The default is 0 (1 second).
- **enable**—Enables PTP on the port base module.
- **sync interval** *value*—Sets the logarithmic mean interval in seconds to send synchronization messages. The range is –2 to 1. The default is 1 second.
- **sync limit** *value*—Sets the maximum clock offset value before PTP attempts to resynchronize. The range is from 50 to 500000000 nanoseconds. The default is 10000 nanoseconds.
- **vlan** *value*—Sets the PTP VLAN on a trunk port. The range is from 1 to 4094. The default is the native VLAN of the trunk port. In boundary mode, only PTP packets in PTP VLAN will be processed, PTP packets from other VLANs will be dropped. Before configuring the PTP VLAN on an interface, the PTP VLAN must be created and allowed on the trunk port.

Step 12 Return to privileged EXEC mode:

end

Step 13 Verify your entries:

show running-config

Step 14 (Optional) Save your entries in the configuration file:

copy running-config startup-config

Example

The following example configures the switch for P2P transparent mode (the default in Power Profile mode), specifies **allow-without-tlv** PTP message processing, and uses default values for all PTP interval settings:

```
switch(config)# ptp allow-without-tlv
```

The following example configures the switch for boundary clock mode using the peer delay request (pdelay-req) mechanism and uses default values for all PTP interval settings:

```
switch(config)# ptp mode boundary pdelay-req
```

Configuring Default Profile Mode on the Switch

This section describes how to configure the switch to operate in Default Profile mode.

Before you begin

The switch sends untagged PTP packets on the native VLAN when the switch port connected to the grandmaster clock is configured as follows:

- Switch is in Default Profile mode.
- Switch is in trunk mode.
- VLAN X is configured as the native VLAN.

When the grandmaster clock requires tagged packets, make one of the following configuration changes:

- Force the switch to send tagged frames by entering the global **vlan dot1q tag native** command.
- Configure the grandmaster clock to send and receive untagged packets. If you make this configuration change on the grandmaster clock, you can configure the switch port as an access port.

These are some guidelines for configuring the Default Profile on the switch:

- When you enter **no** with PTP port configuration commands, the specified port property is set to the default value.
- To determine the value in seconds for the ptp global command *interval* variable, use a logarithmic scale. Below are examples of the *interval* variable value converted to seconds with a logarithmic scale:

Value Entered	Logarithmic Calculation	Value in Seconds
-1	2^{-1}	1/2
0	2^{0}	1

SUMMARY STEPS

1. Enter global configuration mode:
2. Configure the switch for Default Profile mode when the switch is in Power Profile mode. If the switch is already in Default Profile mode, this command has no effect.
3. Specify the synchronization clock mode:
4. (Optional, BC and TC mode) Specify the PTP clock domain:
5. (Optional, BC mode only) Specify the BMCA priority:
6. (Optional, BC mode only) Specify time-property preservation:
7. (Optional, BC mode only) Specify the BC synchronization algorithm:

8. (Optional) Enter interface configuration mode:
9. (Optional) Specify port settings:
10. Return to privileged EXEC mode:
11. Verify your entries:
12. (Optional) Save your entries in the configuration file:

DETAILED STEPS

Step 1 Enter global configuration mode:

configure terminal

Step 2 Configure the switch for Default Profile mode when the switch is in Power Profile mode. If the switch is already in Default Profile mode, this command has no effect.

no ptp profile power

Step 3 Specify the synchronization clock mode:

ptp {**mode boundary delay-req** | **e2etransparent** | **forward** | **gmc-bc**}

- **mode boundary delay-req**—Configures the switch for boundary clock mode using the delay-request mechanism. In this mode, the switch participates in the selection of the most accurate master clock. Use this mode when overload or heavy load conditions produce significant delay jitter.
- **mode e2etransparent**—Configures the switch for end-to-end transparent clock mode. A switch clock in this mode synchronizes all switch ports with the master clock. This switch does not participate in master clock selection and uses the default PTP clock mode on all ports. This is the default clock mode. The message transit time is added to the resident time. Use this mode to reduce jitter and error accumulation.
- **mode forward**—Configures the switch to pass incoming PTP packets as normal multicast traffic.
- **mode gmc-bc**—Configures the switch for grandmaster-boundary clock mode. See Configuring NTP to PTP Time Conversion, on page 31 to configure the switch for this mode.

Step 4 (Optional, BC and TC mode) Specify the PTP clock domain:

ptp domain *domain-number*

domain-number —A number from 0 to 255.

The participating grandmaster clock, switches, and slave devices should be in the same domain.

Step 5 (Optional, BC mode only) Specify the BMCA priority:

ptp priority1 *priority* **priority2** *priority*

- **priority1** *priority*—Overrides the default criteria (such as clock quality and clock class) for the most accurate master clock selection.
- **priority2** *priority*—Breaks the tie between two switches that match the default criteria. For example, enter 2 to give a switch priority over identical switches.*priority* —A priority number from 0 to 255. The default is 128.

Step 6 (Optional, BC mode only) Specify time-property preservation:

ptp time-property persist {*value* | infinite}

- *value*—Time duration, in seconds, from 0-100000. The default is 300.
- infinite—Time properties are preserved indefinitely.

Preserving the time properties prevents slave clocks from detecting a variance in the time values when the redundant GMC comes out of standby.

Step 7 (Optional, BC mode only) Specify the BC synchronization algorithm:

ptp transfer {**feedforward** | **filter** {**adaptive** | **linear**}}

- **feedforward**—Very fast and accurate. No PDV filtering.
- **filter adaptive**—Automatically filters as much PDV as possible.
- **filter linear**—Provides a simple linear filter (default).

Step 8 (Optional) Enter interface configuration mode:

interface *interface-id*

Step 9 (Optional) Specify port settings:

Boundary delay-req mode:

ptp {**announce** {**interval** *value* | **timeout** *value*} | **delay-req interval** *value* | **enable** | **sync** {**interval** *value* | **limit** *value*} | **vlan** *value*}

e2etransparent mode:

ptp {**enable | sync** {**interval** *value* | **limit** *value*}}

- **announce interval** *value*—Sets the logarithmic mean interval in seconds to send announce messages. The range is 0 to 4. The default is 1 (2 seconds).
- **announce timeout** *value*— Sets the logarithmic mean interval in seconds to announce timeout messages. The range is 2 to 10. The default is 3 (8 seconds).
- **delay-req interval** *value*—Sets the logarithmic mean interval in seconds for slave devices to send delay request messages when the port is in the master clock state. The range is -2 to 6. The default is -5 (1 packet every 1/32 seconds, or 32 packets per second).
- **enable**—Enables PTP on the port base module.
- **sync interval** *value*—Sets the logarithmic mean interval in seconds to send synchronization messages. The range is –2 to 1. The default is 1 second.
- **sync limit** *value*—Sets the maximum clock offset value before PTP attempts to resynchronize. The range is from 50 to 500000000 nanoseconds. The default is 500000000 nanoseconds.
- **vlan** *value*—Sets the PTP VLAN on a trunk port. The range is from 1 to 4094. The default is the native VLAN of the trunk port. In boundary mode, only PTP packets in PTP VLAN will be processed, PTP packets from other VLANs will be dropped. Before configuring the PTP VLAN on an interface, the PTP VLAN must be created and allowed on the trunk port.

Step 10 Return to privileged EXEC mode:

end

Step 11 Verify your entries:

show running-config

Step 12 (Optional) Save your entries in the configuration file:

copy running-config startup-config

Example

The following example configures the switch to operate in Default Profile mode and end-to-end transparent mode, and uses default values for all PTP interval settings:

```
switch(config)# no ptp profile
switch(config)# ptp mode e2etransparent
```

The following example configures the switch for Default Profile mode and boundary clock mode with the delay_request mechanism, and uses default values for all PTP interval settings:

```
switch(config)# no ptp profile
switch(config)# ptp mode boundary delay-req
```

Configuring 802.1AS Profile Mode on the Switch (IE 4000 only)

This section describes how to configure the IE 4000 switch to use the 802.1AS Profile and operate in 802.1AS Profile mode.

SUMMARY STEPS

1. Enter global configuration mode:
2. Set the 802.1AS Profile:

DETAILED STEPS

Step 1 Enter global configuration mode:

configure terminal

Step 2 Set the 802.1AS Profile:

ptp profile dot1as

Example

The following example shows configuring the IE 4000 switch to use the 802.1AS Profile:

```
IE4000-SW2(config)#ptp profile dot1as
```

802.1AS Troubleshooting

Refer to the following to troubleshoot 802.1AS issues:

- New Syslogs (Informational)—Parent and Grandmaster clock change syslogs notify user about the parent/grandmaster reselection. If that change happens frequently, or does not meet system expectation, further investigation should be taken. The following shows example log entries:
  - Mar 24 21:22:40.702: %PTP-6-PARENT_CLOCK_CHANGE: Old parent clock identity: 0x0:0:0:0:0:0:0:0 port number: 0, New parent clock identity: 0x0:35:1A:FF:FE:DA:12:80 port number: 9
  - Mar 24 21:22:40.702: %PTP-6-GRANDMASTER_CLOCK_CHANGE: Old grandmaster clock identity: 0x0:0:0:0:0:0:0:0, New grandmaster clock identity: 0x0:35:1A:FF:FE:DA:12:80
  - Mar 24 19:18:34.235: %PTP-6-GRANDMASTER_CLOCK_CHANGE_TO_LOCAL: Old grandmaster clock identity: 0x0:35:1A:FF:FE:DA:12:80, New grandmaster clock identity: 0x58:97:BD:FF:FE:D9:97:80 (local system)
- SyncReceive TimeOut
  - 802.1AS added a new timer to detect sync receive timeout. If the next sync message does not arrive within 3 x sync interval (specified in the header of first sync message) on a PTP SLAVE port, sync receive timeout occurs.
  - This can be learned by turning on **debug ptp event** and observing "PTP (Interface GigabitEthernet1/1): sync receipt timeout" on the console.
  - At SyncReceive Timeout, the state of that PTP port will no longer be SLAVE. The next BMCA will re-select the new SLAVE port.

Verifying Configuration

Command	Purpose
show ptp {**clock** | **foreign-master-records** | **parent** | **port** {**FastEthernet** | **GigabitEthernet**} | **time-property**}	Specifies the PTP information to display. • **clock**—Displays PTP clock information. • **foreign-master-records**—Displays PTP foreign-master-records. • **parent**—Displays PTP parent properties. • **port FastEthernet**—Displays PTP properties for the FastEthernet IEEE 802.3 interfaces. • **port GigabitEthernet**—Displays PTP properties for the GigabitEthernet IEEE 802.3z interfaces. • **time-property**—Displays PTP clock-time properties.

Power Profile Example

```
switch# show ptp parent
 PTP PARENT PROPERTIES
  Parent Clock:
  Parent Clock Identity: 0xA4:C:C3:FF:FE:BF:B4:0
  Parent Port Number: 23
  Observed Parent Offset (log variance): N/A
  Observed Parent Clock Phase Change Rate: N/A
  Grandmaster Clock:
  Grandmaster Clock Identity: 0xA4:C:C3:FF:FE:BF:2B:0
  Grandmaster Clock Quality:
        Class: 248
        Accuracy: Unknown
        Offset (log variance): N/A
        Priority1: 128
        Priority2: 128
switch# show ptp clock
 PTP CLOCK INFO
  PTP Device Type: Boundary clock
  PTP Device Profile: Power Profile
  Clock Identity: 0xA4:C:C3:FF:FE:BF:E0:80
  Clock Domain: 0
  Number of PTP ports: 26
  PTP Packet priority: 4
  Priority1: 128
  Priority2: 128
  Clock Quality:
        Class: 248
        Accuracy: Unknown
        Offset (log variance): N/A
  Offset From Master(ns): 25
  Mean Path Delay(ns): 705
  Steps Removed: 4
  Local clock time: 14:23:56 PST Apr 5 2013
switch# show ptp foreign-master-record
PTP FOREIGN MASTER RECORDS
 Interface GigabitEthernet1/1
   Foreign master port identity: clock id: 0xF4:4E:5:FF:FE:E5:82:0
   Foreign master port identity: port num: 1
   Number of Announce messages: 4
   Message received port: 1
   Time stamps: 1999872004, 1999870997
 Interface GigabitEthernet1/2
   Empty
 Interface GigabitEthernet1/3
   Empty
 Interface GigabitEthernet1/4
   Empty
 Interface GigabitEthernet1/5
   Empty
 Interface GigabitEthernet1/6
   Empty
 Interface GigabitEthernet1/7
   Empty
 Interface GigabitEthernet1/8
   Empty
 Interface GigabitEthernet1/9
   Empty
 Interface GigabitEthernet1/10
   Empty
 Interface GigabitEthernet1/11
   Empty
```

```
 Interface GigabitEthernet1/12
   Empty
 Interface GigabitEthernet1/13
   Empty
 Interface GigabitEthernet1/14
   Empty
 Interface GigabitEthernet1/15
   Empty
 Interface GigabitEthernet1/16
   Empty
 Interface GigabitEthernet1/17
   Empty
 Interface GigabitEthernet1/18
   Empty
 Interface GigabitEthernet1/19
   Empty
 Interface GigabitEthernet1/20
   Empty
switch#
switch# show ptp ?
  clock                  show ptp clock information
  foreign-master-record  show PTP foreign master records
  parent                 show PTP parent properties
  port                   show PTP port properties
  time-property          show PTP clock time property
switch# show ptp time-property
 PTP CLOCK TIME PROPERTY
  Current UTC offset valid: 0
  Current UTC offset: 35
  Leap 59: 0
  Leap 61: 0
  Time Traceable: 16
  Frequency Traceable: 32
  PTP Timescale: 1
  Time Source: Internal Osciliator
  Time Property Persistence: 300 seconds
switch# show ptp port GigabitEthernet 1/1
PTP PORT DATASET: GigabitEthernet1/1
  Port identity: clock identity: 0xF4:4E:5:FF:FE:E5:91:80
  Port identity: port number: 1
  PTP version: 2
  Port state: UNCALIBRATED
  Delay request interval(log mean): 5
  Announce receipt time out: 3
  Peer mean path delay(ns): 0
  Announce interval(log mean): 0
  Sync interval(log mean): 0
  Delay Mechanism: Peer to Peer
  Peer delay request interval(log mean): 0
  Sync fault limit: 500000000
switch#
```

802.1AS Profile Example

```
IE4000-SW2#show ptp clock    //check profile, and clock offset
PTP CLOCK INFO  PTP
Device Type: 802.1AS - Time Aware Bridge
PTP Device Profile: 802.1AS Profile
Clock Identity: 0x58:97:BD:FF:FE:D9:97:80
Clock Domain: 0
…
Offset From Master(ns): 3   // this should be less than 1uS
IE4000-SW2#show ptp port FastEthernet 1/9
  PTP PORT DATASET: FastEthernet1/9
```

```
  …
  Neighbor Rate Ratio: 1 (+0 PPM)  // this should be within +/-100PPM
  Port 802.1AS capable: TRUE  // 802.1AS capable

IE4000-SW2#show ptp parent
 PTP PARENT PROPERTIES
  …
  Clock Identity Path Trace:    // path trace TLV list - the clock IDs of nodes on the clock
 distribution chain from the grandmaster
  Clock Identity 0: 0x0:00:00:11:11:11:11:01 // grandmaster
  Clock Identity 1: 0x0:35:1A:FF:FE:DA:12:80 // 2nd clock in the path
```

Configuration Example

The following example configures the switch for P2P transparent mode, specifies **allow-without-tlv** PTP message processing, and uses default values for all PTP interval settings:

```
switch(config)# ptp allow-without-tlv
```

The following example configures the switch for boundary clock mode using the peer delay request (pdelay-req) mechanism and uses default values for all PTP interval settings:

```
switch(config)# ptp mode boundary pdelay-req
```

The following example configures the switch to operate in Default Profile mode and end-to-end transparent mode and uses default values for all PTP interval settings:

```
switch(config)# no ptp profile
switch(config)# ptp mode e2etransparent
```

The following example configures the switch for Default Profile mode and boundary clock mode with the delay_request mechanism, and uses default values for all PTP interval settings:

```
switch(config)# no ptp profile
switch(config)# ptp mode boundary delay-req
```

Configuring NTP to PTP Time Conversion

Before you begin

- Review the Guidelines and Limitations, on page 18.
- To use the NTP to PTP conversion feature, the switch must have an IP address for NTP to function.
- To use the NTP to PTP conversion feature, you must configure at least one NTP server. Configuring three or more NTP servers allows NTP to ignore bad clocks.

Note For information about configuring NTP, see the section Configuring NTP in the *Catalyst 3750-X and 3560-X Switch Software Configuration Guide, Release 12.2(55)SE* .

- When you enter **no** with PTP port configuration commands, the specified port property is set to the default value.
- To determine the value in seconds for the ptp global command *interval* variable, use a logarithmic scale. Below are examples of the *interval* variable value converted to seconds with a logarithmic scale:

Value Entered	Logarithmic Calculation	Value in Seconds
-1	2^{-1}	1/2
0	2^{0}	1

SUMMARY STEPS

1. Enter global configuration mode:
2. Configure the switch for Default Profile mode when the switch is in Power Profile mode. If the switch is already in Default Profile mode, this command has no effect.
3. Specify GMC-BC as the synchronization clock:
4. (Optional) Specify the BMCA priority:
5. (Optional) Specify the BC synchronization algorithm:
6. Enter interface configuration mode:
7. (Optional) Specify port settings:
8. Return to privileged EXEC mode:
9. Verify your entries:
10. (Optional) Save your entries in the configuration file:

DETAILED STEPS

Step 1 Enter global configuration mode:

configure terminal

Step 2 Configure the switch for Default Profile mode when the switch is in Power Profile mode. If the switch is already in Default Profile mode, this command has no effect.

no ptp profile power

Step 3 Specify GMC-BC as the synchronization clock:

ptp mode gmc-bc delay-req

The GMC-BC automatically selects NTP as the time source if it is available.

Step 4 (Optional) Specify the BMCA priority:

ptp priority1 *priority* **priority2** *priority*

- **priority1** *priority*—Overrides the default criteria (such as clock quality and clock class) for the most accurate master clock selection.
- **priority2** *priority*—Breaks the tie between two switches that match the default criteria. For example, enter 2 to give a switch priority over identical switches.*priority* —A priority number from 0 to 255. The default is 128.

Step 5 (Optional) Specify the BC synchronization algorithm:

ptp transfer {**feedforward** | **filter** {**adaptive** | **linear**}}

- **feedforward**—Very fast and accurate. No PDV filtering.
- **filter adaptive**—Automatically filters as much PDV as possible.
- **filter linear**—Provides a simple linear filter (default).

Step 6 Enter interface configuration mode:

interface interface-id

Step 7 (Optional) Specify port settings:

ptp {**announce** {**interval** *value* | **timeout** *value*} | **delay-req interval** *value* | **enable** | **sync** {**interval** *value* | **limit** *value*} | **vlan** *value*}

- **announce interval** *value*—Sets the logarithmic mean interval in seconds to send announce messages. The range is 0 to 4. The default is 1 (2 seconds).
- **announce timeout** *value*— Sets the time to announce timeout messages. The range is 2 to 10 seconds. The default is 3 (8 seconds).
- **delay-req interval** *value*—Sets the logarithmic mean interval in seconds for slave devices to send delay request messages when the port is in the master clock state. The range is -2 to 6. The default is -5 (1 packet every 1/32 seconds, or 32 packets per second).
- **enable**—Enables PTP on the port base module.
- **sync interval** *value*—Sets the logarithmic mean interval in seconds to send synchronization messages. The range is –2 to 1. The default is 1 second.
- **sync limit** *value*—Sets the maximum clock offset value before PTP attempts to resynchronize. The range is from 50 to 500000000 nanoseconds. The default is 500000000 nanoseconds.
- **vlan** *value*—Sets the PTP VLAN on a trunk port. The range is from 1 to 4094. The default is the native VLAN of the trunk port. In boundary mode, only PTP packets in PTP VLAN will be processed, PTP packets from other VLANs will be dropped. Before configuring the PTP VLAN on an interface, the PTP VLAN must be created and allowed on the trunk port.

Step 8 Return to privileged EXEC mode:

end

Step 9 Verify your entries:

show running-config

Step 10 (Optional) Save your entries in the configuration file:

copy running-config startup-config

Example

The following example configures the switch to use the Default Profile, act as Grandmaster Clock with NTP as the time source, and use the feedforward BC synchronization algorithm:

```
switch(config)# no ptp profile power
switch(config)# ptp mode gmc-bc
switch(config)# ptp transfer feedforward
```

Verifying Configuration

Perform these steps to verify that switch is running as GMC-BC, and that NTP and PTP are synchronized:

SUMMARY STEPS

1. Monitor the status of NTP until NTP locks:
2. Display the status of each individual NTP server:
3. After NTP is up and running, verify that the NTP clock and the PTP clock are in sync.

DETAILED STEPS

Step 1 Monitor the status of NTP until NTP locks:

show ntp status

Note especially the following fields:

- Clock is synchronized/unsynchronized.
- System poll interval—how often the NTP client sends messages in seconds.
- Last update—how many seconds since the last clock adjustment.

Example:

```
switch# show ntp status
Clock is synchronized, stratum 2, reference is 72.163.32.43
nominal freq is 286.1023 Hz, actual freq is 286.0738 Hz, precision is 2**21
ntp uptime is 58682700 (1/100 of seconds), resolution is 3496
reference time is D95162A8.68E52FF9 (22:52:24.409 UTC Wed Jul 15 2015)
clock offset is 0.0459 msec, root delay is 16.19 msec
root dispersion is 15.07 msec, peer dispersion is 0.10 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000099341 s/s
system poll interval is 1024, last update was 925 sec ago.
```

Step 2 Display the status of each individual NTP server:

show ntp association

- The sys.peer is the currently selected reference.
- Candidates are fallback references.
- Falsetickers are bad clocks that are ignored.

Note There is a delay of several seconds from NTP picking an association to NTP declaring lock.

Example:

```
switch# show ntp association
address         ref clock       st   when   poll reach  delay  offset   disp
+~171.68.38.65    .GPS.            1    706   1024   377 60.318  -0.255  0.166
+~171.68.38.66    .GPS.            1    450   1024   377 60.333  -0.096  0.121
-~10.81.254.202   .GPS.            1    555   1024   377 48.707   2.804  0.111
x~173.38.201.115  .GPS.            1    322   1024   377 293.19  74.409  0.107
*~72.163.32.43    .GPS.            1     37   1024   375 17.110  -0.410  0.081
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Step 3 After NTP is up and running, verify that the NTP clock and the PTP clock are in sync.

- **show clock detail** shows the NTP time.
- **show ptp clock** shows the PTP time and the BMCA dataset details.
- **show ptp clock** Steps Removed field indicates whether the GMC-BC really is the GMC or if some other clock is running the PTP network. When the GMC wins the BMCA, the Steps Removed field should be 0.

Example:

```
show clock detail
23:16:53.865 UTC Wed Jul 15 2015
Time source is NTP
show ptp clock
 PTP CLOCK INFO

PTP Device Type: Grand Master clock - Boundary clock
  PTP Device Profile: Default Profile
  Clock Identity: 0xF4:4E:5:FF:FE:E5:95:0
  Clock Domain: 0
  Number of PTP ports: 20

Time Transfer: Linear Filter <<< Displayed when the clock is configured as a BC or a GMC-BC
  Priority1: 128
  Priority2: 128
  Clock Quality:
        Class: 13
        Accuracy: Within 1s
        Offset (log variance): N/A
  Offset From Master(ns): 0
  Mean Path Delay(ns): 0

Steps Removed: 0
  Local clock time: 23:16:53 UTC Jul 15 2015
```

Configuration Example

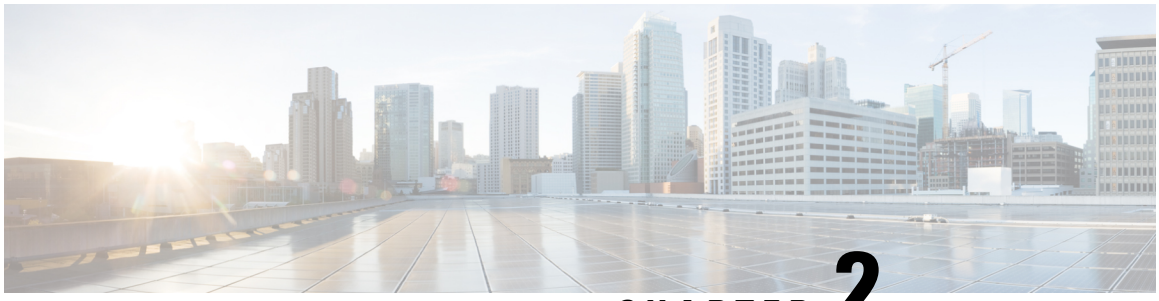
```
switch# conf t
switch(config)# no ptp profile power
switch(config)# ptp mode gmc-bc
switch(config)# ptp transfer feedforward
switch(config)# end
```

Related Documents

- Cisco Industrial Ethernet 4000 switch product documentation
- Cisco Industrial Ethernet 5000 switch product documentation
- Converged Plantwide Ethernet (CPwE) Design and Implementation Guide

Feature History

Feature Name	Release	Feature Information
802.1AS Profile	15.2(5)E2	Initial support on IE 4000 switches.
Time Service Enhancements	15.2(4)EA1	Initial support on IE 5000 switches for NTP to PTP Time Conversion, Feedforward BC, and PDV Filtering.
	15.2(4)EA	Initial support on IE 4000 switches for NTP to PTP Time Conversion, Feedforward BC, and PDV Filtering.
Precision Time Protocol	15.2(4)EC	Initial support of the feature on the IE 4010.
	15.2(2)EB1	Initial support of the feature on the IE 5000.
	15.2(2)EA	Initial support of the feature on the IE 4000.



CHAPTER 2

Configuring SD Swap Drive

Overview

The SD card can be used instead of the internal flash memory of the switch to update or restore configuration settings. In addition, the SD card can be used to boot the switch. You can also copy IOS software and switch configuration settings from a PC or from the switch to the SD card, and then use the SD card to copy this software and settings to other switches.

When an SD card is formatted on the switch, the card is formatted with the Disk Operating System Filing System (DOSFS), a platform-independent industry-standard file system that is supported on various Cisco switches and routers.

The switch does not support third-party SD cards or SD High Capacity (SDHC) cards. Attempting to operate the switch with a non-supported card causes the following message to be displayed:

```
WARNING: Non-IT SD flash detected.
Use of this card during normal operation can impact and
severely degrade performance of the system.
Please use supported SD flash cards only.
```

If the write-protect switch on the SD card is in the lock position, the switch can read data on the card and boot from the card, but updates and files cannot be written to the card.

Inserting and Removing the Flash Memory (SD) Card

To put an SD card in the switch, make sure that the card is oriented properly, and press it into the SD card slot on the switch until the card is seated. To remove the card, press it to release it, then pull it out of the slot.

The SD card is hot-swappable, but it should not be removed from the switch during the boot process or while sdflash write is in progress.

When an SD card is inserted, a syslog message similar to the following is logged:

```
Mar 30 01:38:51.965: %FLASH-6-DEVICE_INSERTED: Flash device inserted
```

When an SD card is removed, a syslog message similar to the following is logged:

```
Mar 30 01:39:12.467: %FLASH-1-DEVICE_REMOVED: Flash device removed
```

Boot Loader Operation

The following boot loader commands can be executed on the SD card:

- boot—Load and boot an executable IOS image
- cat—Concatenate (type) file or files
- copy—Copy a file
- delete—Delete file of files
- dir—List files in directories
- fsck—Check file system consistency
- format—Format a file system
- mkdir—Create directories
- more—Concatenate (display) file
- rename—Rename a file
- rmdir—Delete empty directories
- sd_init—Initialize sd flash file systems

Note The switch can be booted from its internal flash memory or from an SD card. The SD card takes precedence over internal flash memory. If an SD card is installed in the switch, the switch attempts to boot in the following order:

1. From the IOS image that is specified in the SD card system boot path
2. From the first IOS image in the SD card
3. From the IOS image that is specified in the internal flash memory system boot path
4. From the first IOS image in the internal flash

IOS XE Operation

You can insert or remove an SD card while the IOS is running. If you insert a supported Cisco SD card while the IOS is running, the switch validates the Cisco embedded string in the Product Name (PNM) field and displays the product number and the flash capacity of the SD card. If you remove an SD card while the IOS is running, the switch displays a warning message to alert you that the SD card has been removed.

If syslog is enabled, the system also sends a message when the SD card is inserted or removed.

When an SD card is installed in a switch, the following IOS commands operate as described:

- **write** command—Saves the running configuration. If the system boots from an SD card and you run a **write** command, the system saves the running configuration to the SD card, if the card is still installed. If the SD card has been removed, the system saves the running configuration to the internal flash memory and displays this message:

  ```
  WARNING: The SD flash is not present.
  The running-config is saved to the on-board flash.

  NOTE: This warning message is displayed only once.
  ```

  If the system boots from the internal flash memory and you then insert an SD card and run the **write** command, the system saves the running configuration to the internal flash memory.

- **boot** command—Lets you change the system boot parameters.

  If the system boots from an SD card and you run a **boot** command, the following behavior applies:

  - If the SD card is installed and the system boot path or configuration file path points to the SD card, the system boot path or configuration file path is saved to the SD card
  - If the SD card is installed and the system boot path or configuration file path points to the internal flash memory, the system boot path or configuration file path is saved to the internal flash memory
  - If the SD card has been removed and the system boot path or configuration file path points to the SD card, the system boot path or configuration file path is not saved and the following message displays:

    ```
    WARNING: The BOOT/config file path points to the SD flash card and the SD flash
    card is not present.
    The environment variable(s) is not saved.

    NOTE: This warning message is displayed only once.
    ```

  If the system boots from the internal flash memory and you then insert an SD card and run the **boot** command, the following behavior applies:

  - If the system boot path or configuration file path points to the internal flash memory, the system boot path or configuration file path is saved to the internal flash memory
  - If the system boot path or configuration file path points to the SD card, the system boot path or configuration file path is saved to the SD card and the following message is displayed

    ```
    :WARNING: The BOOT/config file path points to the SD flash card.
    The environment variable(s) is saved onto the SD flash card.

    NOTE: This warning message is displayed only once.
    ```

  - If the SD card has been removed and the system boot path or configuration file path points to the SD card, the system boot path or configuration file path is not saved and the following message is displayed:

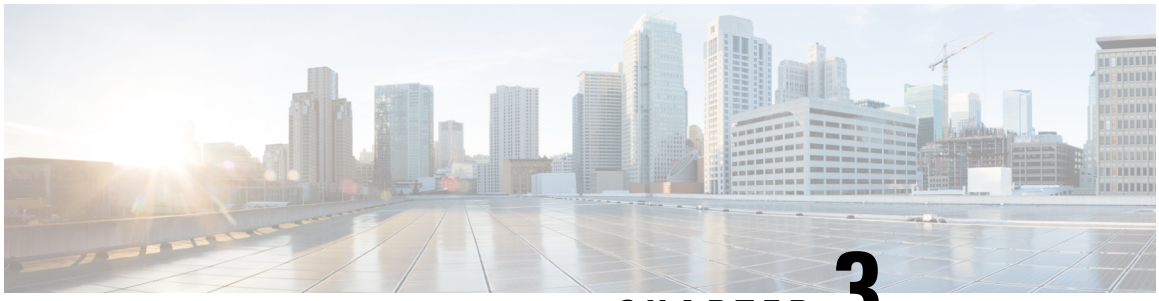
    ```
    WARNING: The BOOT/config file path points to the SD flash card and the SD flash
    card is not present.
    The environment variable(s) is not saved.

    NOTE: This warning message is displayed only once.
    ```

- **sync** command—Copies the IOS image directory (which includes the IOS image file, FPGA image files, Device Manager files, and Profinet/CIP configuration files), the config.text IOS configuration file, the vlan.dat VLAN configuration file, and IOS boot parameters from the internal flash memory to the SD card or from the SD card to the internal flash memory. This command verifies that the IOS image is appropriate for the switch model and that enough destination flash memory is present, and aborts the sync process if a potential problem is detected. The **sync** command obtains the source IOS image directory path and source IOS configuration files path from the IOS boot parameters on the source flash device that is specified in the **sync** command. By default, this command overwrites the destination IOS image directory and IOS configuration files. The "save-old-files" option can be used to override this default behavior. If the running configuration has not been saved and you run the **sync** command, the switch provides the option for you to save the running configuration before the command executes.

  The **sync** command options are:

  - Switch# **sync flash: sdflash:** —Sync IOS image directory, configuration files, and boot parameters from internal flash memory to SD card.
  - Switch# **sync sdflash: flash:** —Sync IOS image directory, configuration files, and boot parameters from SD card to internal flash memory.
  - Switch# **sync flash: sdflash: ios-image-name** —Sync boot IOS image from Flash to SDFlash.
  - Switch# **sync sdflash: flash: ios-image-name** —Sync boot IOS image from SDFlash to Flash.
  - Switch# **sync sdflash: flash: skip [config|env-variable|ios-image]** —Sync either the IOS Config, the environment variables, or IOS image directory from SD card to internal flash memory.



CHAPTER 3

Configuring Resilient Ethernet Protocol

Finding Feature Information

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

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

Resilient Ethernet Protocol Overview

Resilient Ethernet Protocol (REP) is a Cisco proprietary protocol that provides an alternative to Spanning Tree Protocol (STP) to control network loops, handle link failures, and improve convergence time. REP controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing more complex networks and supports VLAN load balancing.

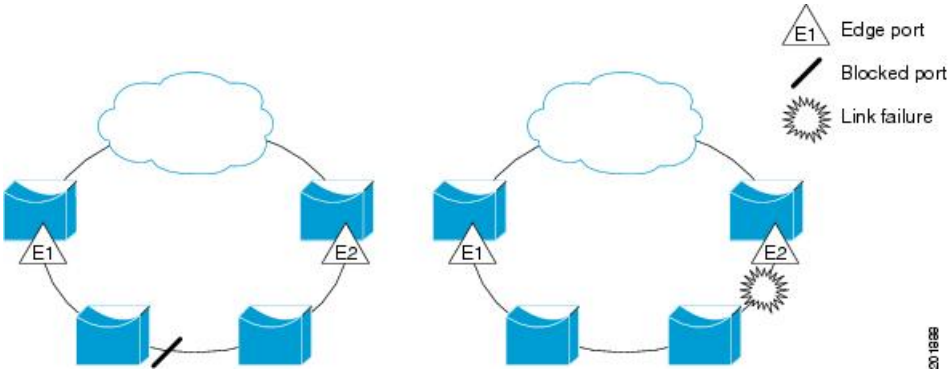
Note REP configuration on downlink ports is supported starting with Cisco IOS XE Fuji 16.9.1.

REP segment is a chain of ports connected to each other and configured with a segment ID. Each segment consists of standard (non-edge) segment ports and two user-configured edge ports. A switch can have no more than two ports that belong to the same segment, and each segment port can have only one external neighbor. A segment can go through a shared medium, but on any link, only two ports can belong to the same segment. REP is supported only on Trunk ports.

The figure below shows an example of a segment consisting of six ports spread across four switches. Ports E1 and E2 are configured as edge ports. When all ports are operational (as in the segment on the left), a single

port is blocked, shown by the diagonal line. This blocked port is also known as the Alternate port (ALT port). When there is a failure in the network, the blocked port returns to the forwarding state to minimize network disruption.

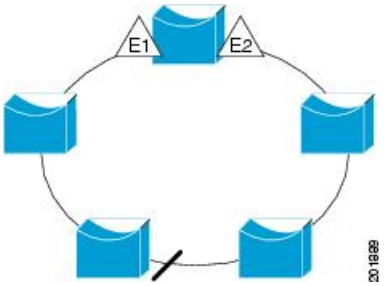
Figure 7: REP Open Segment



The segment shown in the figure above is an open segment; there is no connectivity between the two edge ports. The REP segment cannot cause a bridging loop, and you can safely connect the segment edges to any network. All hosts connected to switches inside the segment have two possible connections to the rest of the network through the edge ports, but only one connection is accessible at any time. If a failure occurs on any segment or on any port on a REP segment, REP unblocks the ALT port to ensure that connectivity is available through the other gateway.

The segment below is a closed segment, also known as Ring Segment, with both edge ports located on the same router. With this configuration, you can create a redundant connection between any two routers in the segment.

Figure 8: REP Ring Segment



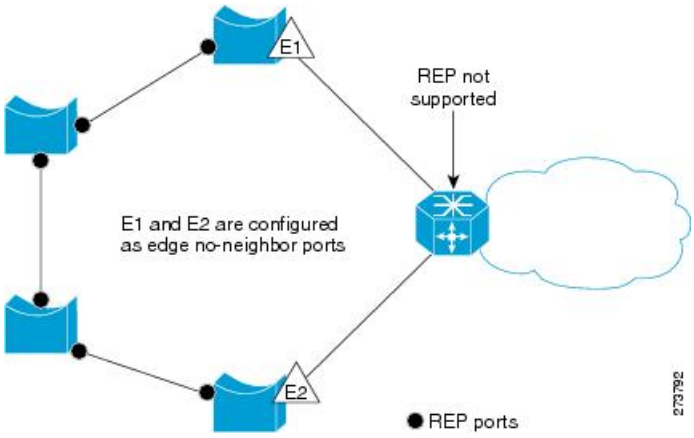
REP segments have the following characteristics:

- If all ports in a segment are operational, one port (referred to as the ALT port) is in the blocked state for each VLAN. If VLAN load balancing is configured, two ALT ports in the segment control the blocked state of VLANs.
- If a port is not operational, and cause a link failure, all ports forward traffic on all VLANs to ensure connectivity.
- In case of a link failure, alternate ports are unblocked as quickly as possible. When the failed link is restored, a logically blocked port per VLAN is selected with minimal disruption to the network.

You can construct almost any type of network based on REP segments.

In access ring topologies, the neighboring switch might not support REP as shown in the figure below. In this case, you can configure the non-REP facing ports (E1 and E2) as edge no-neighbor ports. The edge no-neighbor port can be configured to send an STP topology change notice (TCN) towards the aggregation switch.

Figure 9: Edge No-Neighbor Ports



REP has these limitations:

- You must configure each segment port; an incorrect configuration can cause forwarding loops in the networks.
- REP can manage only a single failed port within the segment; multiple port failures within the REP segment cause loss of network connectivity.
- You should configure REP only in networks with redundancy. Configuring REP in a network without redundancy causes loss of connectivity.

Link Integrity

REP does not use an end-to-end polling function between edge ports to verify link integrity. It implements local link failure detection. The REP Link Status Layer (LSL) detects its REP-aware neighbor and establishes connectivity within the segment. All the VLANs are blocked on an interface until the neighbor is detected. After the neighbor is identified, REP determines which neighbor port should become the alternate port and which ports should forward traffic.

Each port in a segment has a unique port ID. The port ID format is similar to that used by the spanning tree algorithm: a port number (unique on the bridge) associated to a MAC address (unique in the network). When a segment port is coming up, its LSL starts sending packets that include the segment ID and the port ID. The port is declared as operational after it performs a three-way handshake with a neighbor in the same segment.

A segment port does not become operational if:

- No neighbor has the same segment ID.
- More than one neighbor has the same segment ID.
- A neighbor does not acknowledge a local port as a peer.

Each port creates an adjacency with its immediate neighbor. After the neighbor adjacencies are created, the ports negotiate with each other to determine the blocked port for the segment, which will function as the

alternate port. All the other ports become unblocked. By default, REP packets are sent to a bridge protocol data unit-class MAC address. The packets can also be sent to a Cisco multicast address, which is used only to send blocked port advertisement (BPA) messages when there is a failure in the segment. The packets are dropped by the devices not running REP.

Fast Convergence

REP runs on a physical link basis and not on a per-VLAN basis. Only one hello message is required for all the VLANs, and this reduces the load on the protocol. We recommend that you create VLANs consistently on all the switches in a given segment and configure the same allowed VLANs on the REP trunk ports. To avoid the delay introduced by relaying messages in software, REP also allows some packets to be flooded to a regular multicast address. These messages operate at the hardware flood layer (HFL) and are flooded to the entire network, not just the REP segment. Switches that do not belong to the segment treat them as data traffic. You can control flooding of these messages by configuring an administrative VLAN for the entire domain or for a particular segment.

VLAN Load Balancing

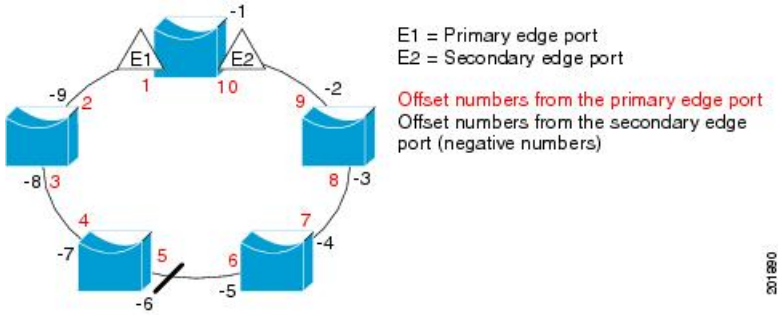
One edge port in the REP segment acts as the primary edge port; and another as the secondary edge port. It is the primary edge port that always participates in VLAN load balancing in the segment. REP VLAN balancing is achieved by blocking some VLANs at a configured alternate port and all the other VLANs at the primary edge port. When you configure VLAN load balancing, you can specify the alternate port in one of three ways:

- By entering the port ID of the interface. To identify the port ID of a port in the segment, enter the **show interface rep detail** interface configuration command for the port.
- By entering the **preferred** keyword to select the port that you previously configured as the preferred alternate port with the **rep segment** *segment-id* **preferred** interface configuration command.
- By entering the neighbor offset number of a port in the segment, which identifies the downstream neighbor port of an edge port. The neighbor offset number range is –256 to +256; a value of 0 is invalid. The primary edge port has an offset number of 1; positive numbers above 1 identify downstream neighbors of the primary edge port. Negative numbers indicate the secondary edge port (offset number -1) and its downstream neighbors.

Note Configure offset numbers on the primary edge port by identifying a port's downstream position from the primary (or secondary) edge port. Never enter an offset value of 1 because that is the offset number of the primary edge port.

The following figure shows neighbor offset numbers for a segment, where E1 is the primary edge port and E2 is the secondary edge port. The red numbers inside the ring are numbers offset from the primary edge port; the black numbers outside of the ring show the offset numbers from the secondary edge port. Note that you can identify all the ports (except the primary edge port) by either a positive offset number (downstream position from the primary edge port) or a negative offset number (downstream position from the secondary edge port). If E2 became the primary edge port, its offset number would then be 1 and E1 would be -1.

Figure 10: Neighbor Offset Numbers in a Segment



When the REP segment is complete, all the VLANs are blocked. When you configure VLAN load balancing, you must also configure triggers in one of two ways:

- Manually trigger VLAN load balancing at any time by entering the **rep preempt segment** *segment-id* privileged EXEC command on the switch that has the primary edge port.
- Configure a preempt delay time by entering the **rep preempt delay** *seconds* interface configuration command. After a link failure and recovery, VLAN load balancing begins after the configured preemption time period elapses. Note that the delay timer restarts if another port fails before the time has elapsed.

Note When VLAN load balancing is configured, it does not start working until triggered by either manual intervention or a link failure and recovery.

When VLAN load balancing is triggered, the primary edge port sends out a message to alert all the interfaces in the segment about the preemption. When the secondary port receives the message, the message is sent to the network to notify the alternate port to block the set of VLANs specified in the message and to notify the primary edge port to block the remaining VLANs.

You can also configure a particular port in the segment to block all the VLANs. Only the primary edge port initiates VLAN load balancing, which is not possible if the segment is not terminated by an edge port on each end. The primary edge port determines the local VLAN load-balancing configuration.

Reconfigure the primary edge port to reconfigure load balancing. When you change the load-balancing configuration, the primary edge port waits for the **rep preempt segment** command or for the configured preempt delay period after a port failure and recovery, before executing the new configuration. If you change an edge port to a regular segment port, the existing VLAN load-balancing status does not change. Configuring a new edge port might cause a new topology configuration.

Spanning Tree Interaction

REP does not interact with STP, but it can coexist. A port that belongs to a segment is removed from spanning tree control and STP BPDUs are not accepted or sent from segment ports. Therefore, STP cannot run on a segment.

To migrate from an STP ring configuration to REP segment configuration, begin by configuring a single port in the ring as part of the segment and continue by configuring contiguous ports to minimize the number of segments. Each segment always contains a blocked port, so multiple segments means multiple blocked ports

and a potential loss of connectivity. When the segment has been configured in both directions up to the location of the edge ports, you then configure the edge ports.

REP Ports

REP segments consist of Failed, Open, or Alternate ports:

- A port configured as a regular segment port starts as a failed port.
- After the neighbor adjacencies are determined, the port transitions to alternate port state, blocking all the VLANs on the interface. Blocked-port negotiations occur, and when the segment settles, one blocked port remains in the alternate role and all the other ports become open ports.
- When a failure occurs in a link, all the ports move to the Failed state. When the Alternate port receives the failure notification, it changes to the Open state, forwarding all the VLANs.

A regular segment port converted to an edge port, or an edge port converted to a regular segment port, does not always result in a topology change. If you convert an edge port into a regular segment port, VLAN load balancing is not implemented unless it has been configured. For VLAN load balancing, you must configure two edge ports in the segment.

A segment port that is reconfigured as a spanning tree port restarts according to the spanning tree configuration. By default, this is a designated blocking port. If PortFast is configured or if STP is disabled, the port goes into the forwarding state.

How to Configure Resilient Ethernet Protocol

A segment is a collection of ports connected to one another in a chain and configured with a segment ID. To configure REP segments, configure the REP administrative VLAN (or use the default VLAN 1) and then add the ports to the segment, using interface configuration mode. You should configure two edge ports in a segment, with one of them being the primary edge port and the other the secondary edge port by default. A segment should have only one primary edge port. If you configure two ports in a segment as primary edge ports, for example, ports on different switches, the REP selects one of them to serve as the segment's primary edge port. If required, you can configure the location to which segment topology change notices (STCNs) and VLAN load balancing are to be sent.

Default REP Configuration

REP is disabled on all the interfaces. When enabled, the interface is a regular segment port unless it is configured as an edge port.

When REP is enabled, the task of sending segment topology change notices (STCNs) is disabled, all the VLANs are blocked, and the administrative VLAN is VLAN 1.

When VLAN load balancing is enabled, the default is manual preemption with the delay timer disabled. If VLAN load balancing is not configured, the default after manual preemption is to block all the VLANs in the primary edge port.

REP Configuration Guidelines

Follow these guidelines when configuring REP:

- We recommend that you begin by configuring one port and then configure contiguous ports to minimize the number of segments and the number of blocked ports.
- If more than two ports in a segment fail when no external neighbors are configured, one port goes into a forwarding state for the data path to help maintain connectivity during configuration. In the **show rep interface** command output, the Port Role for this port shows as “Fail Logical Open”; the Port Role for the other failed port shows as “Fail No Ext Neighbor”. When the external neighbors for the failed ports are configured, the ports go through the alternate port state transitions and eventually go to an open state or remain as the alternate port, based on the alternate port selection mechanism.
- REP ports must be Layer 2 IEEE 802.1Q or Trunk ports.
- We recommend that you configure all trunk ports in the segment with the same set of allowed VLANs.
- Be careful when configuring REP through a Telnet connection. Because REP blocks all VLANs until another REP interface sends a message to unblock it. You might lose connectivity to the router if you enable REP in a Telnet session that accesses the router through the same interface.
- You cannot run REP and STP on the same segment or interface.
- If you connect an STP network to a REP segment, be sure that the connection is at the segment edge. An STP connection that is not at the edge could cause a bridging loop because STP does not run on REP segments. All STP BPDUs are dropped at REP interfaces.
- You must configure all trunk ports in the segment with the same set of allowed VLANs, or a misconfiguration occurs.
- If REP is enabled on two ports on a switch, both ports must be either regular segment ports or edge ports. REP ports follow these rules:
  - There is no limit to the number of REP ports on a switch; however, only two ports on a switch can belong to the same REP segment.
  - If only one port on a switch is configured in a segment, the port should be an edge port.
  - If two ports on a switch belong to the same segment, they must be both edge ports, both regular segment ports, or one regular port and one edge no-neighbor port. An edge port and regular segment port on a switch cannot belong to the same segment.
  - If two ports on a switch belong to the same segment and one is configured as an edge port and one as a regular segment port (a misconfiguration), the edge port is treated as a regular segment port.
- REP interfaces come up in a blocked state and remain in a blocked state until they are safe to be unblocked. You need to be aware of this status to avoid sudden connection losses.
- REP sends all LSL PDUs in untagged frames on the native VLAN. The BPA message sent to the Cisco multicast address is sent on the administration VLAN, which is VLAN 1 by default.
- You can configure how long a REP interface remains up without receiving a hello from a neighbor. You can use the **rep lsl-age-timer** value interface configuration command to set the time from 120 ms to 10000 ms. The LSL hello timer is then set to the age-timer value divided by 3. In normal operation, three LSL hellos are sent before the age timer on the peer switch expires and checks for hello messages.
  - EtherChannel port channel interfaces do not support LSL age-timer values less than 1000 ms. If you try to configure a value less than 1000 ms on a port channel, you receive an error message and the command is rejected.

- REP ports cannot be configured as one of the following port types:
  - Switched Port Analyzer (SPAN) destination port
  - Tunnel port
  - Access port
- REP is supported on EtherChannels, but not on an individual port that belongs to an EtherChannel.
- There can be a maximum of 64 REP segments per switch.

Configuring the REP Administrative VLAN

To avoid the delay created by link-failure messages, and VLAN-blocking notifications during load balancing, REP floods packets to a regular multicast address at the hardware flood layer (HFL). These messages are flooded to the whole network, and not just the REP segment. You can control the flooding of these messages by configuring an administrative VLAN for the whole domain or for a particular segment.

Follow these guidelines when configuring the REP administrative VLAN:

- If you do not configure an administrative VLAN, the default is VLAN 1.
- You can configure one admin VLAN on the switch for all segments or configure an admin VLAN per segment.
- The administrative VLAN cannot be the RSPAN VLAN.

To configure the REP administrative VLAN, follow these steps, beginning in privileged EXEC mode:

SUMMARY STEPS

1. **configure terminal**
2. **rep admin vlan** *vlan-id*
3. **end**
4. **show interface** [*interface-id*] **rep detail**
5. **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 2	**rep admin vlan** *vlan-id* **Example:** `device(config)# rep admin vlan 2`	Specifies the administrative VLAN. The range is from 2 to 4094. • To set the admin VLAN to 1, which is the default, enter the **no rep admin vlan** global configuration command.

	Command or Action	Purpose
Step 3	**end** **Example:** `device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 4	**show interface** [*interface-id*] **rep detail** **Example:** `device# show interface gigabitethernet1/1 rep detail`	(Optional) Verifies the configuration on a REP interface.
Step 5	**copy running-config startup config** **Example:** `device# copy running-config startup config`	(Optional) Saves your entries in the switch startup configuration file.

Configuring a REP Interface

To configure REP, enable REP on each segment interface and identify the segment ID. This task is mandatory, and must be done before other REP configurations. You must also configure a primary and secondary edge port on each segment. All the other steps are optional.

Follow these steps to enable and configure REP on an interface:

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **switchport mode trunk**
4. **rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**]
5. **rep stcn** {**interface** *interface id* | **segment** *id-list* | **stp**}
6. **rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}
7. **rep preempt delay** *seconds*
8. **rep lsl-age-timer** *value*
9. **end**
10. **show interface** [*interface-id*] **rep** [**detail**]
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** *interface-id* **Example:** `device# interface gigabitethernet1/1`	Specifies the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface).
Step 3	**switchport mode trunk** **Example:** `device# switchport mode trunk`	Configures the interface as a Layer 2 trunk port.
Step 4	**rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**] **Example:** `device# rep segment 1 edge no-neighbor primary`	Enables REP on the interface and identifies a segment number. The segment ID range is from 1 to 1024. **Note** You must configure two edge ports, including one primary edge port, for each segment. These optional keywords are available: • (Optional) **edge**—Configures the port as an edge port. Each segment has only two edge ports. Entering the keyword **edge** without the keyword **primary** configures the port as the secondary edge port. • (Optional) **primary**—Configures the port as the primary edge port, the port on which you can configure VLAN load balancing. • (Optional) **no-neighbor**—Configures a port with no external REP neighbors as an edge port. The port inherits all the properties of an edge port, and you can configure the properties the same way you would for an edge port. **Note** Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the keyword **primary** on both the switches, the configuration is valid. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the **show rep topology** privileged EXEC command. • (Optional) **preferred**—Indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. **Note** Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port.

	Command or Action	Purpose
Step 5	**rep stcn** {**interface** *interface id* | **segment** *id-list* | **stp**} **Example:** `device# rep stcn segment 25-50`	(Optional) Configures the edge port to send segment topology change notices (STCNs). • **interface** *interface-id*—Designates a physical interface or port channel to receive STCNs. • **segment** *id-list*—Identifies one or more segments to receive STCNs. The range is from 1 to 1024. • **stp**—Sends STCNs to STP networks. **Note** Spanning Tree (MST) mode is required on edge no-neighbor nodes when **rep stcn stp** command is configured for sending STCNs to STP networks.
Step 6	**rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**} **Example:** `device# rep block port id 0009001818D68700 vlan 1-100`	(Optional) Configures VLAN load balancing on the primary edge port, identifies the REP alternate port in one of three ways (**id** *port-id*, *neighbor_offset*, **preferred**), and configures the VLANs to be blocked on the alternate port. • **id** *port-id*—Identifies the alternate port by port ID. The port ID is automatically generated for each port in the segment. You can view interface port IDs by entering the **show interface** *type number* **rep** [**detail**] privileged EXEC command. • *neighbor_offset*—Number to identify the alternate port as a downstream neighbor from an edge port. The range is from -256 to 256, with negative numbers indicating the downstream neighbor from the secondary edge port. A value of **0** is invalid. Enter **-1** to identify the secondary edge port as the alternate port. **Note** Because you enter the **rep block port** command at the primary edge port (offset number 1), you cannot enter an offset value of 1 to identify an alternate port. • **preferred**—Selects the regular segment port previously identified as the preferred alternate port for VLAN load balancing. • **vlan** *vlan-list*—Blocks one VLAN or a range of VLANs. • **vlan all**—Blocks all the VLANs. **Note** Enter this command only on the REP primary edge port.

	Command or Action	Purpose
Step 7	**rep preempt delay** *seconds* **Example:** `device# rep preempt delay 100`	(Optional) Configures a preempt time delay. • Use this command if you want VLAN load balancing to be automatically triggered after a link failure and recovery. • The time delay range is between15 to 300 seconds. The default is manual preemption with no time delay. **Note** Enter this command only on the REP primary edge port.
Step 8	**rep lsl-age-timer** *value* **Example:** `device# rep lsl-age-timer 2000`	(Optional) Configures a time (in milliseconds) for which the REP interface remains up without receiving a hello from a neighbor. The range is from 120 to 10000 ms in 40-ms increments. The default is 5000 ms (5 seconds). **Note** • EtherChannel port channel interfaces do not support LSL age-timer values that are less than 1000 ms. • Both the ports on the link should have the same LSL age configured in order to avoid link flaps.
Step 9	**end** **Example:** `device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 10	**show interface** [*interface-id*] **rep** [**detail**] **Example:** `device(config)# show interface gigabitethernet1/1 rep detail`	(Optional) Displays the REP interface configuration.
Step 11	**copy running-config startup-config** **Example:** `device(config)# copy running-config startup-config`	(Optional) Saves your entries in the router startup configuration file.

Setting Manual Preemption for VLAN Load Balancing

If you do not enter the **rep preempt delay** *seconds* interface configuration command on the primary edge port to configure a preemption time delay, the default is to manually trigger VLAN load balancing on the segment. Be sure that all the other segment configurations have been completed before manually preempting VLAN load balancing. When you enter the **rep preempt delay segment** *segment-id* command, a confirmation message is displayed before the command is executed because preemption might cause network disruption.

SUMMARY STEPS

1. **configure terminal**
2. **rep preempt segment** *segment-id*
3. **show rep topology segment** *segment-id*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.

Step 2	**rep preempt segment** *segment-id*
	Example:
	`device# rep preempt segment 100`
	`The command will cause a momentary traffic disruption.`
	`Do you still want to continue? [confirm]`
		Manually triggers VLAN load balancing on the segment.
		You need to confirm the command before it is executed.

Step 3	**show rep topology segment** *segment-id* **Example:** `device# show rep topology segment 100`	(Optional) Displays REP topology information.
Step 4	**end** **Example:** `device# end`	Exits privileged EXEC mode.

Configuring SNMP Traps for REP

You can configure a router to send REP-specific traps to notify the Simple Network Management Protocol (SNMP) server of link-operational status changes and port role changes.

SUMMARY STEPS

1. **configure terminal**
2. **snmp mib rep trap-rate** *value*
3. **end**
4. **show running-config**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `device# configure terminal`	Enters global configuration mode.
Step 2	**snmp mib rep trap-rate** *value* **Example:** `device(config)# snmp mib rep trap-rate 500`	Enables the switch to send REP traps, and sets the number of traps sent per second. • Enter the number of traps sent per second. The range is from 0 to 1000. The default is 0 (no limit is imposed; a trap is sent at every occurrence).
Step 3	**end** **Example:** `device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show running-config** **Example:** `device# show running-config`	(Optional) Displays the running configuration, which can be used to verify the REP trap configuration.
Step 5	**copy running-config startup-config** **Example:** `device# copy running-config startup-config`	(Optional) Saves your entries in the switch startup configuration file.

Monitoring Resilient Ethernet Protocol Configurations

You can display the rep interface and rep topology details using the commands in this topic.

SUMMARY STEPS

1. **show interface** [*interface-id*] **rep** [**detail**]
2. **show rep topology** [**segment** *segment-id*] [**archive**] [**detail**]

DETAILED STEPS

Step 1 **show interface** [*interface-id*] **rep** [**detail**]

Displays REP configuration and status for an interface or for all the interfaces.

- (Optional) **detail**—Displays interface-specific REP information.

Example:

```
Device# show interfaces TenGigabitEthernet4/1 rep detail

TenGigabitEthernet4/1 REP enabled
Segment-id: 3 (Primary Edge)
PortID: 03010015FA66FF80
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 02040015FA66FF804050
Port Role: Open
Blocked VLAN: <empty>
Admin-vlan: 1
Preempt Delay Timer: disabled
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 999, tx: 652
HFL PDU rx: 0, tx: 0
BPA TLV rx: 500, tx: 4
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 6, tx: 5
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 135, tx: 136
```

Step 2 **show rep topology** [**segment** *segment-id*] [**archive**] [**detail**]

Displays REP topology information for a segment or for all the segments, including the primary and secondary edge ports in the segment.

- (Optional) **archive**—Displays the last stable topology.

  Note An archive topology is not retained when the switch reloads.

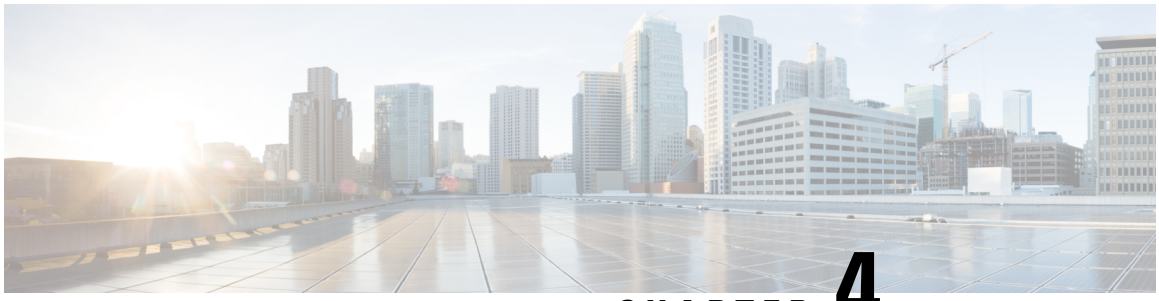
- (Optional) **detail**—Displays detailed archived information.

Example:

```
Device# show rep topology

REP Segment 1
BridgeName       PortName   Edge Role
---------------- ---------- ---- ----
10.64.106.63     Te5/4      Pri  Open
10.64.106.228    Te3/4           Open
10.64.106.228    Te3/3           Open
10.64.106.67     Te4/3           Open
10.64.106.67     Te4/4           Alt
10.64.106.63     Te4/4      Sec  Open

REP Segment 3
BridgeName       PortName   Edge Role
---------------- ---------- ---- ----
10.64.106.63     Gi50/1     Pri  Open
SVT_3400_2       Gi0/3           Open
SVT_3400_2       Gi0/4           Open
10.64.106.68     Gi40/2          Open
10.64.106.68     Gi40/1          Open
10.64.106.63     Gi50/2     Sec  Alt
```



CHAPTER 4

Common Industrial Protocol (CIP)

CIP Restrictions

CIP can be enabled on only one VLAN on the switch.

Enabling CIP

Before you begin

By default, CIP is not enabled.

SUMMARY STEPS

1. **Configure Terminal**
2. **cip security** { **password** *password* | **window timeout** *value* }
3. **interface vlan 20**
4. **cip enable**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**
8. **show cip** { **connection** | **faults** | **file** | **miscellaneous** | **object** | **security** | **session** | **status** }
9. **debug cip** { **assembly** | **connection manager** | **errors** | **event** | **file** | **io** | **packet** | **request response** | **security** | **session** | **socket** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**Configure Terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**cip security** { **password** *password* | **window timeout** *value* }	Sets CIP security options on the switch.
Step 3	**interface vlan 20**	Enters interface configuration mode.
Step 4	**cip enable**	Enables CIP on a VLAN.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show running-config**	Verifies your entries.
Step 7	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.
Step 8	**show cip** { **connection** | **faults** | **file** | **miscellaneous** | **object** | **security** | **session** | **status** }	(Optional) Displays information about the CIP subsystem.
Step 9	**debug cip** { **assembly** | **connection manager** | **errors** | **event** | **file** | **io** | **packet** | **request response** | **security** | **session** | **socket** }	(Optional) Enables debugging of the CIP subsystem.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS basic commands	*Cisco IOS Configuration Fundamentals Command Reference*

Standards and RFCs

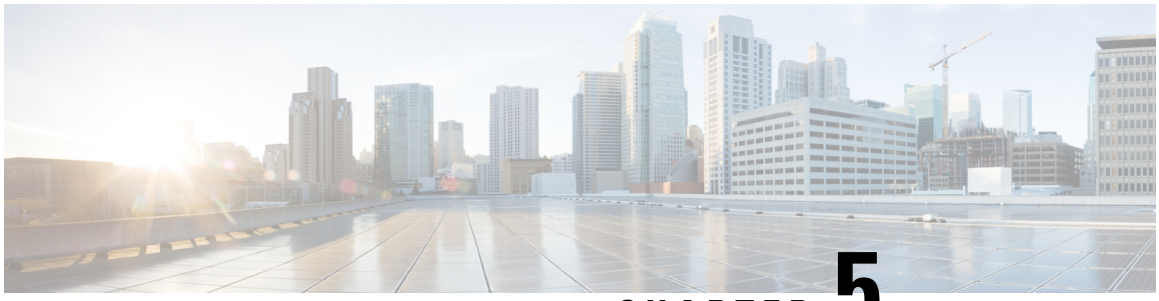
Standard/RFC	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index?dtid=osscdc000283

Technical Assistance

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



CHAPTER 5

Modicon Communication Bus (MODBUS)

MODBUS Overview

Modicon Communication Bus (MODBUS) is an application layer protocol for client-server communication between a switch (server) and a device in the network running MODBUS client software (client). You can use MODBUS over a serial line to connect a computer to a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems.

MODBUS also runs on Ethernet TCP/IP networks. Use MODBUS TCP over an Ethernet network when connecting the switch to devices such as intelligent electronic devices (IEDs), distributed controllers, substation routers, IP phones, Wireless Access Points, and other network devices such as redundant substation switches.

The client can be an IED or a human machine interface (HMI) application that remotely configures and manages devices running MODBUS TCP. The switch functions as the server.

The switch encapsulates a request or response message in a MODBUS TCP application data unit (ADU). A client sends a message to a TCP port on the switch.

Configuring MODBUS

The MODBUS TCP server listens for MODBUS client requests on TCP port 502 by default. Port 502 is enabled when MODBUS server is started unless you configure a different port for MODBUS communications. The MODBUS server is disabled by default.

To configure MODBUS:

Before you begin

If a firewall or other security services are enabled, the switch TCP port might be blocked, and the switch and the client cannot communicate. If a firewall and other security services are disabled, a denial-of-service attack might occur on the switch. To add security when using MODBUS TCP, configure an ACL to permit traffic from specific clients or configure QoS to rate-limit traffic.

Step 1 Enter global configuration mode:

configure terminal

Step 2 Enable MODBUS TCP on the switch:

scada modbus tcp server

To disable MODBUS on the switch and return to the default settings, enter the **no scada modbus tcp server** global configuration command.

The system displays a message to warn you that starting the MODBUS TCP server is a security risk:

WARNING: Starting Modbus TCP server is a security risk. Please understand the security issues involved before proceeding further. Do you still want to start the server? [yes/no]:

Step 3 Enter **yes** to confirm that you understand the security issues and to proceed with starting the server.

Step 4 (Optional) Set the TCP port to which clients send messages:

scada modbus tcp server port *tcp-port-number*

The range for *tcp-port-number* is 1 to 65535. The default is 502.

Step 5 (Optional) Set the number of simultaneous connection requests sent to the switch:

scada modbus tcp server connection *connection-requests*

The range for *connection-requests* is 1 to 5. The default is 1.

Step 6 Return to privileged EXEC mode:

end

Example

```
Switch# configure terminal
Switch(config)# scada modbus tcp server
WARNING: Starting Modbus TCP server is a security risk. Please understand the security
issues involved
before proceeding further. Do you still want to start the server? [yes/no]: y
Switch(config)# end
```

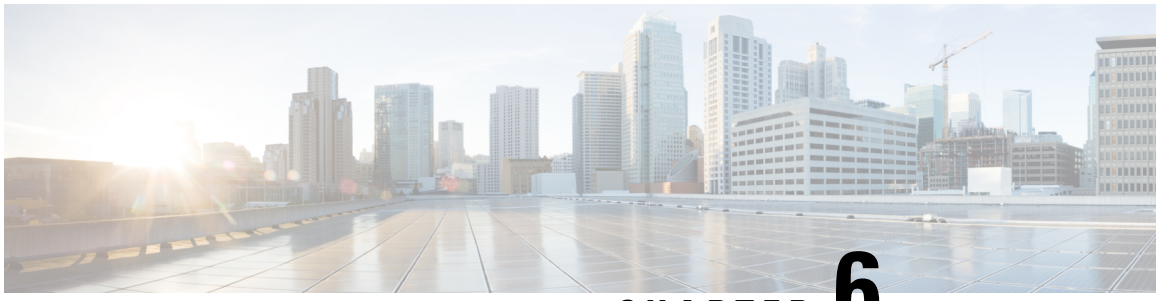
Displaying MODBUS Information

Use the commands listed below to display information for MODBUS TCP.

Command	Purpose
show scada modbus tcp server	Displays the server information and statistics
show scada modbus tcp server connections	Shows information and statistics for each client connection

Command	Purpose
clear scada modbus tcp server statistics	Clears all the statistics for the Modbus server, including statistics for each client connection

```
Switch# show scada modbus tcp server
Summary: enabled, running, process id 142
Conn Stats: listening on port 801, 4 max simultaneous connections
    0 current client connections
    0 total accepted connections, 0 accept connection errors
    0 closed connections, 0 close connection errors
Send Stats: 0 tcp msgs sent, 0 tcp bytes sent, 0 tcp errors
    0 responses sent, 0 exceptions sent, 0 send errors
Recv Stats: 0 tcp msgs received, 0 tcp bytes received, 0 tcp errors
    0 requests received, 0 receive errors
```



CHAPTER 6

Serviceability and Zeroization Features for IoT

Serviceability and Zeroization Features for IoT

The following features are included in the Cisco IOS-XE release 16.11.1 for the Internet of Things (IoT) products.

Serviceability Features

Cisco IOS-XE **show tech-support** functionality is extensively used by technical support for various platforms that run IOS-XE and comprises of a library of shell scripts that spawn various show commands to obtain the state of the device for debugging purposes. The tech-support output is very critical in debugging various problems in the system and has been a key component in debug infrastructure.

The **show tech-support** series of commands has been a part of the Cisco IOS and IOS-XE release since release 4.0(0)N1(1a). The IoT products follow the core IOS-XE software functionality.

The output from the **show tech-support** command is very long. To better manage this output, you can redirect the output to a file (for example, **show tech-support** > *filename*) in the local writable storage file system or the remote file system.

You can use one of the following redirection methods:

> *filename* — Redirects the output to a file.

>> *filename* — Redirects the output to a file in append mode.

Examples

This example shows how to display technical support information:

```
device# show tech-support
```

This example shows how to redirect the technical support information to a file:

```
device# show tech-support > bootflash:TechSupport.txt
```

This example shows how to display the brief technical support information for the device:

```
device# show tech-support brief
```

This example shows how to display the technical support information for a specific feature:

```
device# show tech-support aaa
```

This example shows how to display the commands used to generate the technical support information:

```
device# show tech-support commands
```

For Cisco IOS-XE release 16.11.1, improvements were made to improve the monitoring capabilities of forwarding plane (QFP) using CLI and SNMP. **show platform resources** would display QFP details and an SNMP MIB walk would include all QFP objects including memory related MIB objects. **show inventory** and **show inventory oid** will display the related Forwarding processor and its OID information.

show tech-support is enhanced to include the following CLIs:

```
show platform hardware qfp active  infrastructure punt config cause
show platform hardware qfp active  infrastructure punt internal-interface
show platform hardware qfp active  interface if-name internal0/0/rp:0
show platform hardware qfp active  interface if-name internal0/0/recycle:0
show platform hardware qfp active  interface if-name internal0/0/crypto:0
show platform hardware qfp active  infrastructure uidb internal0/0/rp:0 input config
show platform hardware qfp active  infrastructure uidb internal0/0/recycle:0 input config
show platform hardware qfp active  infrastructure uidb internal0/0/crypto:0 input config
show platform hardware qfp active  infrastructure uidb internal0/0/rp:0 output config
show platform hardware qfp active  infrastructure uidb internal0/0/recycle:0 output config
show platform hardware qfp active  infrastructure uidb internal0/0/crypto:0 output config
show platform hardware qfp active  infrastructure punt statistics interface 1
show platform hardware qfp active  infrastructure punt statistics interface 2
show platform hardware qfp active  interface if-name internal0/0/rp:0 statistics
show platform hardware qfp active  interface if-name internal0/0/recycle:0 statistics
show platform hardware qfp active  interface if-name internal0/0/crypto:0 statistics
show platform hardware qfp active  infrastructure punt statistics type per-cause
show platform hardware qfp active  infrastructure punt statistics type global-drop
show platform hardware qfp active  infrastructure punt statistics type punt-drop
show platform hardware qfp active  infrastructure punt statistics type inject-drop
show platform hardware qfp active  statistics drop
show platform hardware qfp active  system state
show platform hardware qfp active  system transactions
show platform hardware qfp active  datapath infrastructure time basic
show platform hardware qfp active  infrastructure exmem statistics
show platform hardware qfp active  infrastructure exmem statistics user
show platform hardware qfp active  infrastructure exmem resource
show platform hardware qfp active  infrastructure exmem region
show platform hardware qfp active  infrastructure exmem table
show platform hardware qfp active  infrastructure bqs status
show platform hardware qfp active  feature acl control
show platform hardware qfp active  feature acl tree
show platform hardware qfp active  feature tunnel state
show platform hardware qfp active  feature erspan state
show platform hardware qfp active  feature ess state
show platform hardware qfp active  feature ipfrag global
show bootlog FP active
show bootlog RP active
show platform software diagnostic chassis-manager R0 cpld
show platform software diagnostic chassis-manager R0 status
show platform software ipc queue-based chassis-manager R0 connection
show platform software ipc stream-based ios RP active connection
show platform software ipc stream-based ios RP active manager
show platform software process environment ios rp active
```

```
show power
show license tech support
show license summary
```

New CLIs that have been added are:

```
show tech-support l2
show tech-support acl
show tech-support dhcp
show tech-support port-channel
show tech-support private-vlan
show tech-support vlan
show tech-support confidential
```

Detailed information on all of these commands can be found in the Catalyst 9500 Switches Command Reference:

https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-10/command_reference_b_1610_9500_cr/interface_and_hardware_commands.html#wp3848147936

Device Zeroization or Declassification

Zeroization consists of erasing any and all potentially sensitive information in the device. This function is also referred to as Declassification. This includes erasure of Main memory, cache memories, and other memories containing packet data, NVRAM, and Flash memory. The process of zeroization is launched upon the initiation of a user command and a subsequent trigger.

On the device, the Reset button is used exclusively for triggering the Zeroization/Declassification process which zeroize and erase device configuration files or entire flash file system depending on the option provided under "service declassify".

The zeroization process starts as soon as the reset button is pressed down. The CLI command, "service declassify", is used to set the desired action in response to reset button press. To prevent accidental erasure of the system configuration/image, the default setting is set to "no service declassify".

Command Line Interface

There are two levels of zeroization actions, erase-nvram and erase-all. The following CLI shows the options:

```
device(config)#service declassify ?
erase-nvram    Enable erasure of device configuration as declassification action. Default
is no erasure.
erase-all      Enable erasure of both flash and nvram file systems as part of
declassification. Default is no erasure
```

The “erase-nvram” level of declassification process searches for the following files, and erases the ones found.

- flash:/nvram_config
- flash:/vlan.dat

This also erases the complete NVRAM filesystem, therefore, all configurations, including startup and running configurations will get deleted.

The perma-locked bootable image(s) in the flash file system will still be available and can be used for booting the device.

The “erase-all” level of zeroization process erases the entire flash file system. This also wipes out all files and perma-locked bootable image(s). All interfaces are shut down before this process. Here, erasure of individual files in the flash file system is not possible and the only option is to erase the entire flash file system. This also erases packet data, ASIC data and processors related caches along with scrubbing Main memory.

With any level of zeroization, the device always fall back to the ROMMON prompt on the console after the erasure of configuration files or flash file system.

Zeroization Trigger

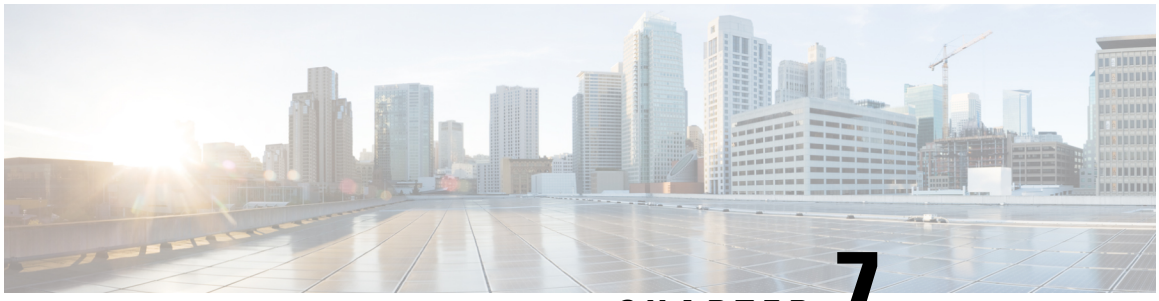
The user needs to press the button after configuring the level of erasure required by the above CLI commands. To make sure that the button press has been identified by underlying software, the user needs to press and hold it for ONE second, or at least till the zero LED starts blinking.

Zeroization Support in bootloader

The zeroization process may take several minutes, depending on several system parameters such as the size of DDR memory, EMMC disk size, etc.

It is possible that the zeroization may get interrupted by a power cycle before it completes. Since the primary OS image on EMMC itself gets purged during zeroization, it becomes impossible to continue zeroization after a power cycle. To solve this, zeroization support has been in the bootloader and will run it to completion even if it gets interrupted by power cycles.

The IOS-XE sets a flag in the PMU persistent register before relinquishing control to the bootloader through a reboot. The bootloader then sets an internal variable in QSPI flash so that it is persistent even across power cycles.



CHAPTER 7

Embedded Packet Capturer

Embedded Packet Capturer Overview

Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from the device and to analyze them locally or save and export them for offline analysis. The captured data is stored in .pcap file format, which can be analyzed by using a standard packet analysis tool such as Wireshark. This feature facilitates troubleshooting by gathering information about the packet format. This feature also facilitates application analysis and security.

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. The network administrator may define the capture buffer size and the maximum number of bytes of each packet to capture. The packet capture rate can be throttled using further administrative controls. For example, options allow for filtering the packets to be captured using an Access Control List and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

Note Packet Capture is supported only on physical interfaces with the ingress direction.

Configuring Embedded Packet Capture

Follow these steps to configure Embedded Packet Capture:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enable privileged EXEC mode.
Step 2	**monitor capture** *capture-name* **access-list** *access-list-name*	Configure a monitor capture specifying an access list as the core filter for the packet capture.

	Command or Action	Purpose
Step 3	**monitor capture** *capture-name* **limit duration** *seconds*	Configure monitor capture limits.
Step 4	**monitor capture** *capture-name* **interface** *interface-name* **in**	Configure monitor capture specifying an attachment point and the packet flow direction.
Step 5	**monitor capture** *capture-name* **buffer circular size** *bytes*	Configure a buffer to capture packet data. This size can be maximum 100 MB.
Step 6	**monitor capture** *capture-name* **start**	Start the capture of packet data at a traffic trace point into a buffer.
Step 7	**monitor capture** *capture-name* **export** *file-location/file-name*	Export captured data for analysis.
Step 8	**monitor capture** *capture-name* **stop**	Stop the capture of packet data at a traffic trace point.
Step 9	**monitor capture** *capture-name* **clear**	Clear the captured buffer data.
Step 10	**end**	Exit privileged EXEC mode.

Example

Monitoring and Maintaining Captured Data

Perform this task to monitor and maintain the packet data captured. Capture buffer details and capture point details are displayed.

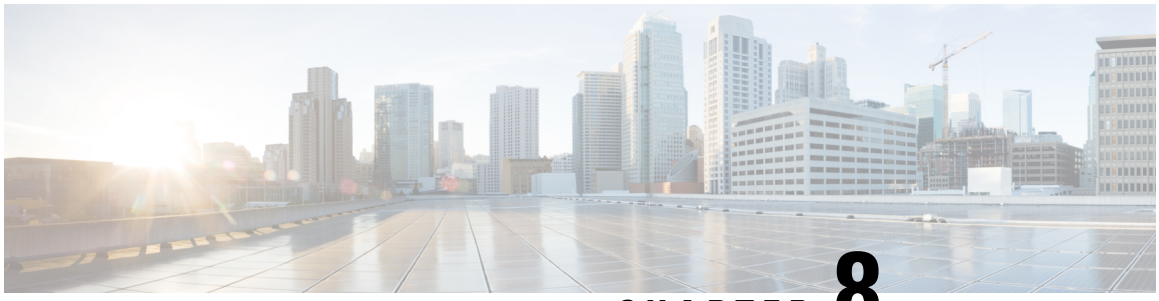
Procedure

	Command or Action	Purpose
Step 1	**enable**	Enable privileged EXEC mode.
Step 2	**show monitor capture** *capture-buffer-name* **buffer dump**	(Optional) Display a hexadecimal dump of captured packet and its metadata.
Step 3	**show monitor capture** *capture-buffer-name* **parameter**	(Optional) Display a list of commands that were used to specify the capture.
Step 4	**debug epc capture-point**	(Optional) Enable packet capture point debugging.
Step 5	**debug epc provision**	(Optional) Enables packet capture provisioning debugging.
Step 6	**exit**	Exit privileged EXEC mode.

Example

Feature History

Feature Name	Release	Feature Information
Embedded Packet Capture	Cisco IOS XE 16.11.1	Initial support on Cisco Catalyst IE 3200, 3300, 3400, and Cisco Embedded Service 3300 Series Switches



CHAPTER 8

REP Fast

REP Fast Overview

The Resilient Ethernet Protocol (REP) Fast feature allows faster link failure detection and convergence on the switch copper Gigabit Ethernet (GE) ports.

This document describes only REP Fast. For complete information about REP and how to configure it, refer to Configuring Resilient Ethernet Protocol.

REP was originally designed for Fast Ethernet (FE 10/100) ports. Link down detection time on FE ports is 10 milliseconds (ms) and convergence time is about 50 ms. On Fiber GE ports, link down time is 10 ms, but on GE copper interfaces, the IEEE 802.3 specification mandates the link drop detection and recovery times to be 750 ms for a master and 350 ms for a slave. As a result, link loss and recovery can be detected a lot more quickly on GE fiber interfaces than on corresponding copper interfaces. This in turn means that the convergence time for REP is a lot higher when using GE copper interfaces.

To improve link down detection time, a real-time operating system (RTOS)/beacon mechanism is implemented to trigger faster link failure detection (within 5-10 ms) when a REP interface is configured for REP Fast mode. RTOS has two timers for each REP interface. The first timer is triggered every 3 ms to transmit the beacon frame to the neighbor node. After successful transmission and reception of the frame, both the timers are reset. If the packet is not received after the transmission, then the second timer is triggered to check the reception within 10 ms. If the packet is not received, upon the timer expiry, a link down packet is sent to the switch.

If the neighbor acknowledges and is configured for REP Fast mode, convergence occurs within 50 ms. If a neighbor switch does not support RTOS, normal REP mode must be used for link up/down detection. In this case, you need to disable fastmode on both ends of the link.

Configuring REP Fast

Follow these steps to configure REP Fast:

Before you begin

Enable REP on the switch and configure the REP topology as described in Configuring Resilient Ethernet Protocol.

Step 1 Enter global configuration mode:

configure terminal

Step 2 Specify the interface and enter interface configuration mode:

interface *interface-id*

Step 3 Enable REP Fast:

rep fastmode

Step 4 Return to priviledged exec mode:

end

Example

```
Switch# configure terminal
Switch(config)# int gi 1/4
Switch(config-if)# rep fastmode
Switch(config-if)# end
Switch# sh run int gi 1/4
interface GigabitEthernet1/4
switchport trunk allowed vlan 1-10
switchport mode trunk
rep segment 1 edge
rep fastmode
```

Displaying REP Fast Beacon Information

When REP Fast is enabled, the system sends beacon frames to the neighbor node for link status detection. Use the following command to display the number of beacon frames sent and received on an interface.

In priviledged exec mode, enter:

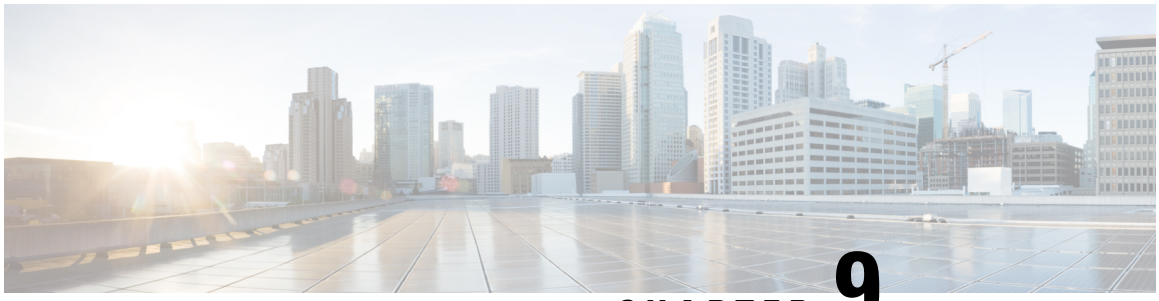
show platform rep beacon interface *interface-id*

Example

```
Switch# sh platform rep beacon GigabitEthernet 1/4
Beacon RX : 43984
Beacon TX : 46826
```

Feature History

Feature Name	Release	Feature Information
REP Fast	Cisco IOS XE 16.11.1	Initial support on Cisco Catalyst IE 3200, 3300, 3400, and Cisco Embedded Service 3300 Series Switches



CHAPTER 9

Configuring Locate Switch

The Locate Switch feature allows you to easily locate the physical location of a specific switch on your network. This feature enables the flashing of LEDs on a specific switch, which is useful for locating a device within a room with many interconnected devices. When this feature is activated, the following LEDs on the device blink alternately green and red for the specified amount of time:

- Alarm IN (there are two alarm IN LEDs)
- Alarm OUT
- System
- Express Setup

To configure and activate Locate Switch:

Enter the following command and a value from 9 to 255 for the number of seconds to continue the blinking pattern:

locate-switch [*seconds*]

The default is 255. The LEDs continue the blinking pattern for the specified number of seconds. Enter 0 seconds to stop the blinking pattern before the number of seconds expires.

The **locate-switch** command is a volatile command and will not be saved or displayed in running or startup configuration.

Example

```
Switch# locate-switch
Switch# locate-switch 0
```