Cisco Catalyst IE 3200, 3300, 3400 Rugged Series, Cisco IOS XE 16.10.1 Software Configuration Guide

First Published: 2018-11-01

Last Modified: 2019-02-11

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.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

In this document, the terms Grandmaster clock (GMC) or time source and time recipient are used instead of the traditional Master/Slave nomenclature.

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 time source (grandmaster clock) and the time recipient. PTP sends messages between the time source and time recipient 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 time source and time recipient devices. 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.

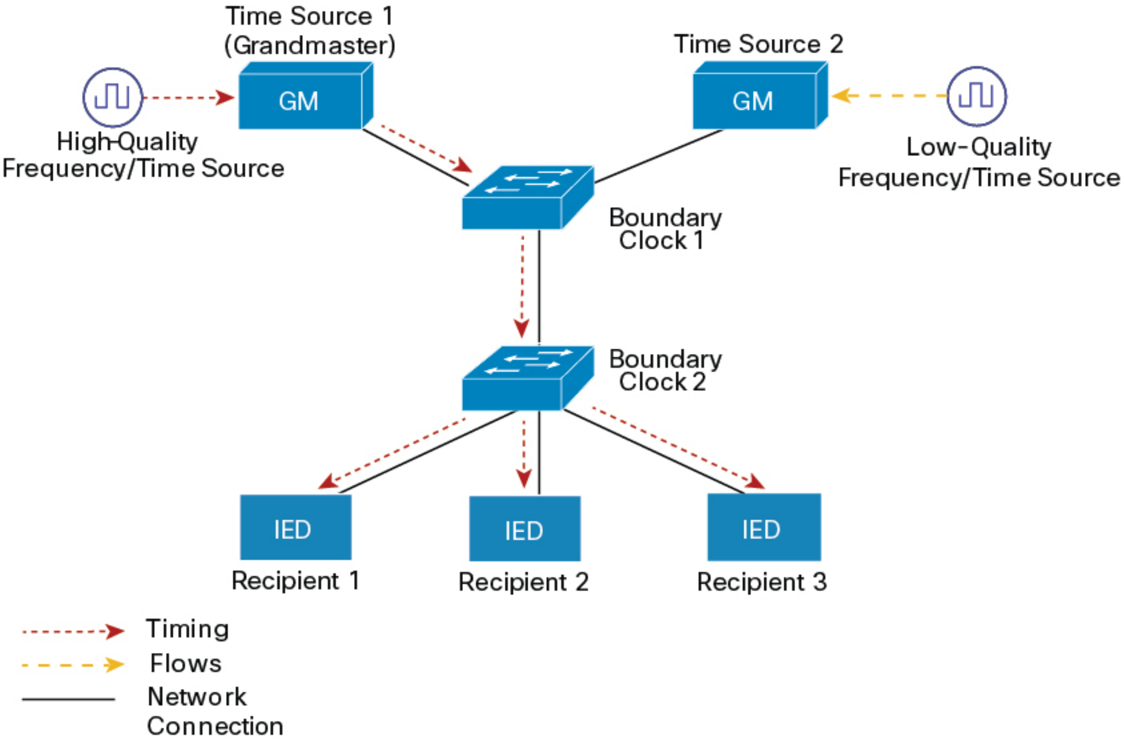
PTP provides a method, using transparent clocks, to measure and account for the delay in a time-interval field in network timing packets, making the switches temporarily transparent to the time source and time recipient nodes on the network. An end-to-end transparent clock forwards all messages on the 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 7.

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, Time Source 1 is the grandmaster clock. If Time Source 1 becomes unavailable, the time recipient boundary clocks switch to Time Source 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

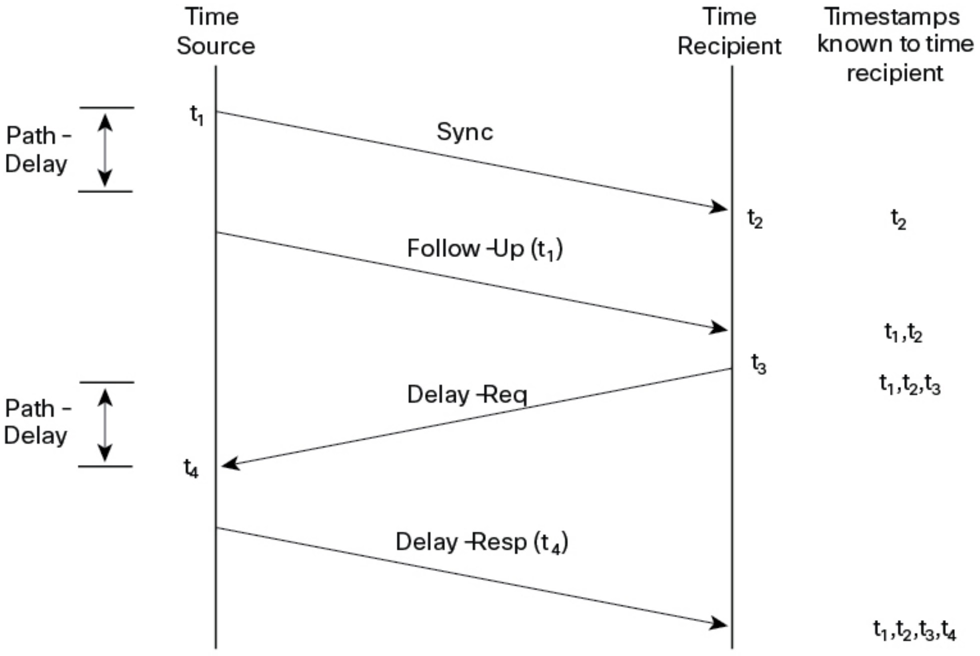
These messages are sent in the following sequence:

1. The time source sends a Sync message to the time recipient and notes the time t1 at which it was sent.
2. The time recipient receives the Sync message and notes the time of reception t2.
3. The time source conveys to the time recipient the timestamp t1 by embedding the timestamp t1 in a Follow_Up message.
4. The time recipient sends a Delay_Req message to the time source and notes the time t3 at which it was sent.
5. The time source receives the Delay_Req message and notes the time of reception t4.
6. The time source conveys to the time recipient the timestamp t4 by embedding it in a Delay_Resp message.

After this sequence, the time recipient possesses all four timestamps. These timestamps can be used to compute the offset of the time recipient clock relative to the time source, 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 time source to time recipient is the same as the time required from time recipient to time source. This assumption is not always valid on an Ethernet network due to asymmetrical packet delay times.

Figure 2: Detailed Steps—Boundary Clock Synchronization



Path-Delay = $[(t_4 - t_1) - (t_3 - t_2)]/2$

Offset from Time Source = $(t_2 - t_1)$- Path-Delay

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:

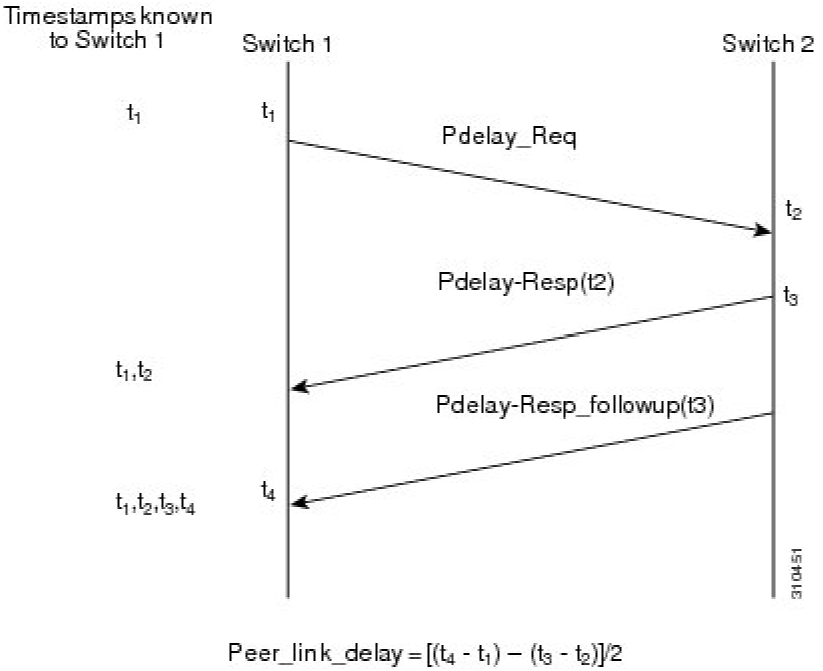
- Pdelay_Req
- Pdelay_Resp
- Pdelay_Resp_Follow_Up

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 time source and time recipient clocks operate at the same frequency. However, *drift* can occur on the network. Drift is the frequency difference between the time source and time recipient clocks. 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 time source 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 time source 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 time source 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 time source 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 time source clocks or no time source clocks, as a result of the time source 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. 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 to non-master mode. The master port synchronizes the clocks connected downstream, while the non-master 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.

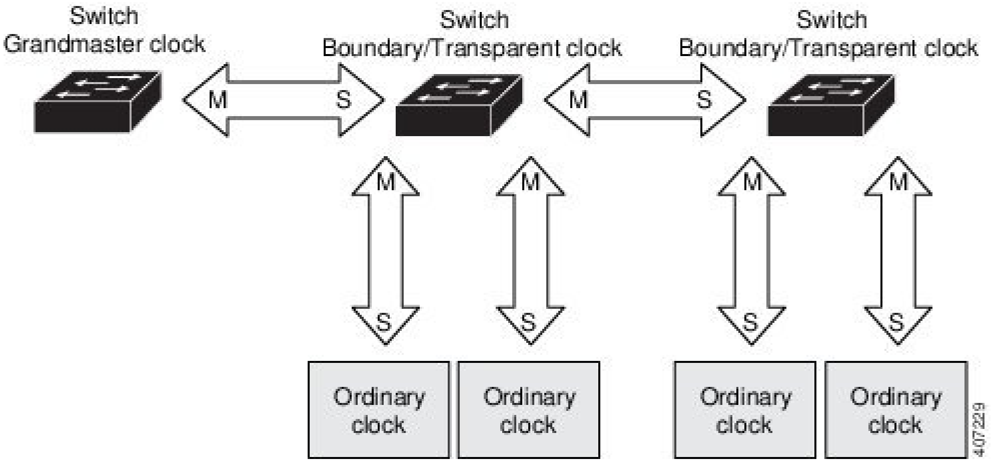
The time recipient uses this information when determining the offset between the time recipient’s and the time source's time. E2E transparent clocks do not provide correction for the propagation delay of the link itself.

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 time recipient contains the sum of all link delays. In theory, this is the total end-to-end delay (from time source to time recipient) of the SYNC packet.

The following figure illustrates PTP clocks in a time source-time recipient hierarchy within a PTP network.

Figure 4: PTP Clock Hierarchy



PTP Profiles

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

The following PTP profiles are available on the switch:

- Default Profile
- Power Profile (C37.238-2011/IEC 61850-9-3 support)
- 802.1AS Profile (IE 4000 only)
- Extended Power Profile (IEEE C37.238-2017 support—Transparent clock mode only)

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 9 lists the configuration values for the switch in Default Profile mode.

Power Profile Mode

The Power Profile is defined in C37.238-2011 - 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 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.
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.

PTP Field	Power Profile Value	Switch Configuration Value	
		Power Profile Mode	**Default Profile Mode**
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 time recipient port that were not sent by the receiving time-aware system are used immediately; that is, there is no foreign-time source qualification.
    - A port that the BMCA determines should be a time source port enters the time source state immediately; that is, there is no pre-time source 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.

PTP Profile Comparison

Table 2: Comparison of PTP Profiles on IE Switches

Profile	**Default (*)**		**Power**		**802.1AS**
Standard	IEEE1588 v2 (J.3)		IEEE C37.238-2011		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

Switch Port Mode	Configuration	Power Profile Mode		Default Profile Mode	
		Behavior	**Priority**	**Behavior**	**Priority**
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 17 for guidelines for configuring each of the clock modes.

Boundary Clock Mode

A switch configured for boundary clock mode participates in selecting the best time source 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 time source clock. If a more accurate clock is detected, then the switch synchronizes to that clock and becomes a time recipient 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 and P2P Transparent Clock Modes

Transparent clocks synchronize their local clocks to the GMC by snooping the Sync messages. They do not participate in the best master clock algorithm. Transparent clocks use 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 .

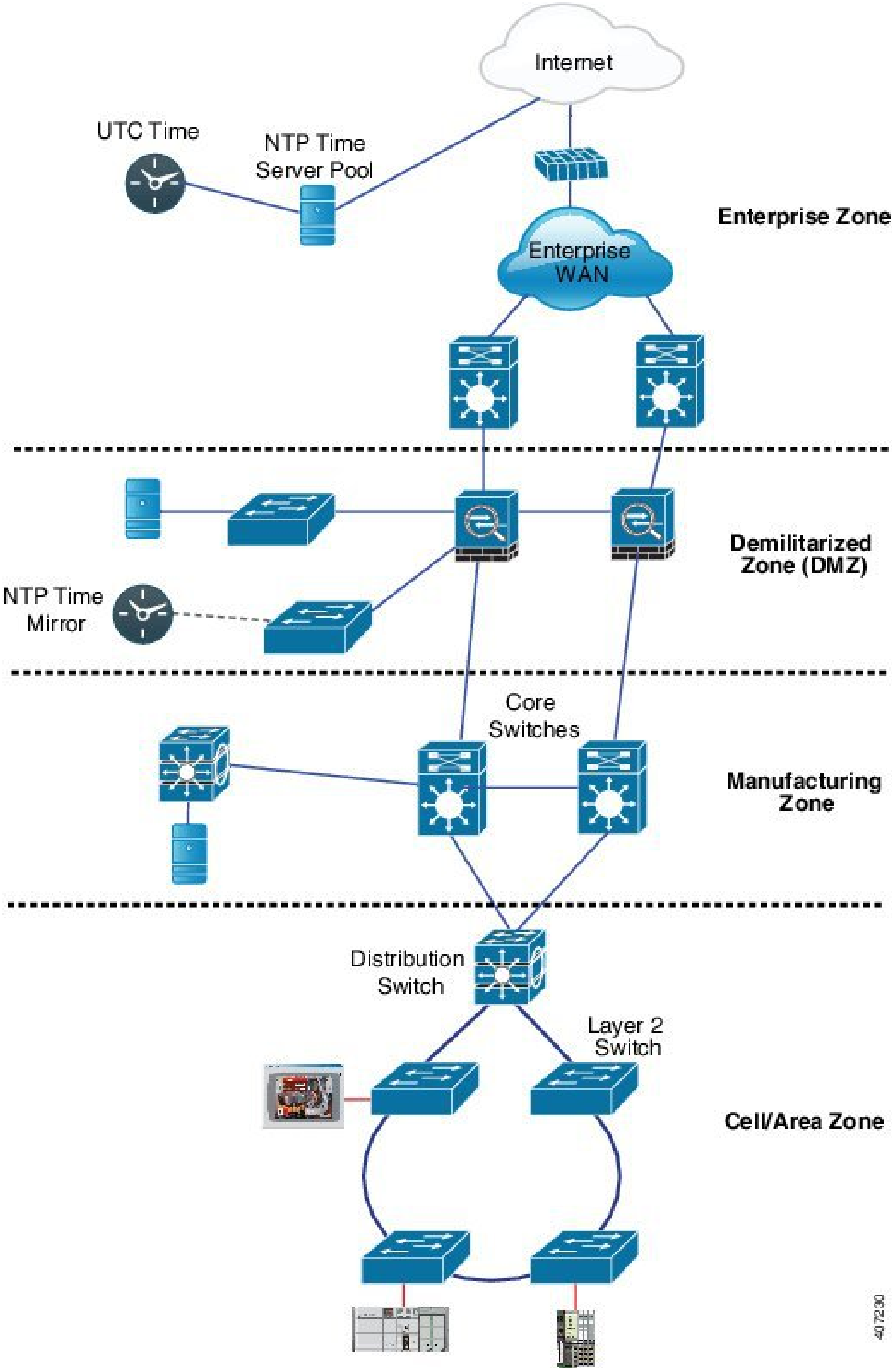
NTP to PTP Time Conversion

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 time source and an end device. 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.

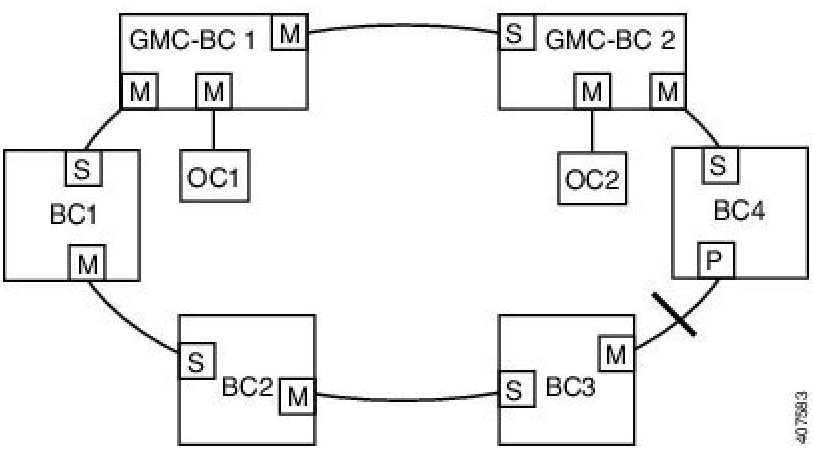
Figure 5: Industrial Network with NTP and PTP



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 17.
- 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 12 and the peer_delay mechanism, enter the **ptp mode boundary pdelay-req** command.
- To disable Power Profile mode and return the switch to E2E and P2P Transparent Clock Modes, on page 12, 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 12 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 and P2P Transparent Clock Modes, on page 12, 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 9 (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 9 (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 9 (Layer 2).

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

GMC Block

- The GMC Block feature is not supported in Forward mode.

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 and P2P Transparent Clock Modes, on page 12.
- 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; not available in Extended Power Profile) 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}| gmc-bc}

- **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 time source 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 time source 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.
- **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; not available in Extended Power Profile) 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 default is 0 for the Power Profile and 254 for the Extended Power Profile.

The participating grandmaster clock, switches, and time recipient 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 time source 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 time recipient 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 time recipient devices to send pdelay request messages when the port is in the time source 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. The command **no ptp** *profile-name* returns to the switch to the Default Profile mode.
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, BC mode only) Specify the DSCP Event/General message values:
9. (Optional) Enter interface configuration mode:
10. (Optional) Specify port settings:
11. Return to privileged EXEC mode:
12. Verify your entries:
13. (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. The command **no ptp** *profile-name* returns to the switch to the Default Profile mode.

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 time recipient 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 time recipient 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, BC mode only) Specify the DSCP Event/General message values:

ptp ip dscp *dscp_value* **message** {*event* | *general*}

- *dscp_value*—A number from 0 to 63.
- **message** *event*—Configures the DSCP value for event messages. The default value is 59.
- **message** *general*—Configures the DSCP value for general messages. The default value is 47.

Step 9 (Optional) Enter interface configuration mode:

interface *interface-id*

Step 10 (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 time recipient devices to send delay request messages when the port is in the time source 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 11 Return to privileged EXEC mode:

end

Step 12 Verify your entries:

show running-config

Step 13 (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 time recipient 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 time recipient. The next BMCA will re-select the new time recipient 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 17.
- 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 or Power Profile mode.
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 or Power Profile mode.

no ptp profile power

or

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 time source 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 time recipient devices to send delay request messages when the port is in the time source 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
C37.238-2017: Power Profile Support	15.2(8)E1	IEEE 1588 PTPv2, C37.238-2017 Power Profile Support for Transparent Clock on IE 4000, IE 4010, and IE 5000.
PTP IE5000 Horizontal Stack	15.2(8)E	PTP support on IE 5000 switches operating as a horizontal stack.
PTP over port channels TC	15.2(8)E	PTP support over port channels for Transparent Clocks.
PTP over port channels BC	15.2(7)E3	PTP support over port channels for Boundary clock.
IEC 61850-9-3 2016 profile	15.2(7)E3	Configure for Power Profile in Transparent clock mode to enable.
GMC Block	15.2(7)E3	Initial support on IE 4000, IE 4010, and IE 5000 switches.
PTP DSCP Values	15.2(7)E3	Initial support on IE 4000, IE 4010, and IE 5000 switches.
PTP Serviceability	15.2(7)E1a	Initial support on IE 4000, IE 4010, and IE 5000 switches.
802.1AS Profile	15.2(5)E2	Initial support on IE 4000 switches.

Feature Name	Release	Feature Information
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

An SD card can be used instead of the internal flash memory of a switch to update or restore configuration settings. In addition, the SD card can be used to boot the switch. You can also copy Cisco 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 nonsupported 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 insert an SD card into a 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, and 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 an SD card:

- **boot**—Load and boot an executable Cisco IOS image
- **cat**—Concatenate (type) a file or files
- **copy**—Copy a file
- **delete**—Delete a 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

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

Cisco IOS XE Operation

You can insert or remove an SD card while Cisco IOS is running. If you insert a supported Cisco SD card while Cisco 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 Cisco 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 Cisco 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 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.
    ```

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

  - 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 Cisco IOS image directory (which includes the Cisco IOS image file, FPGA image files, Device Manager files, and Profinet/CIP configuration files), the config.text Cisco IOS configuration file, the vlan.dat VLAN configuration file, and Cisco 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 Cisco 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 Cisco IOS image directory path and source Cisco IOS configuration file path from the Cisco IOS boot parameters on the source flash device that is specified in the **sync** command. By default, this command overwrites the destination Cisco IOS image directory and Cisco IOS configuration files. The option to save old files 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 is run.

The **sync** command options are:

- Switch# **sync flash: sdflash:**—Synchronizes the Cisco IOS image directory, configuration files, and boot parameters from internal flash memory to the SD card.
- Switch# **sync sdflash: flash:**—Synchronizes the Cisco IOS image directory, configuration files, and boot parameters from the SD card to internal flash memory.
- Switch# **sync flash: sdflash: ios-image-name** —Synchronizes the boot Cisco IOS image from Flash to SDFlash.
- Switch# **sync sdflash: flash: ios-image-name** —Synchronizes the Cisco IOS image from SDFlash to Flash.
- Switch# **sync sdflash: flash: skip [config|env-variable|ios-image]** —Synchronizes either the Cisco IOS configuration, the environment variables, or the Cisco IOS image directory from the 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 https://cfn.cloudapps.cisco.com/ITDIT/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.

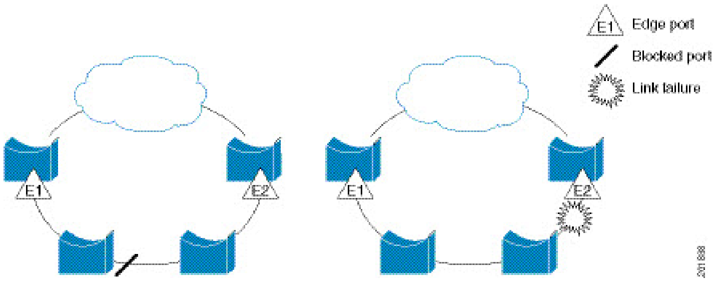
Note The feature is supported on Cisco Series Switches with the Network Essentials license.

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.

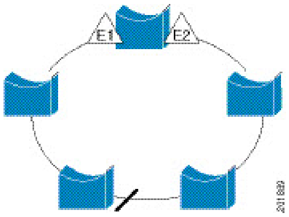
Figure 7: REP Open Segment



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:

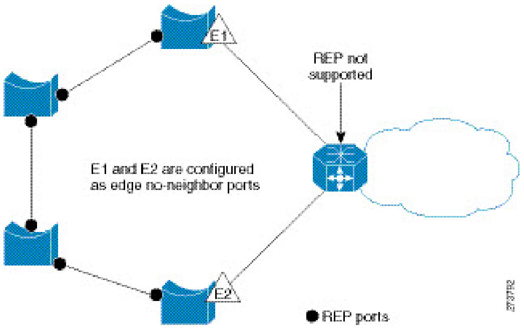
- If all ports in a segment are operational, one port (referred to as the ALT port) is in the blocked state for each VLAN. If VLAN load balancing is configured, two ALT ports in the segment control the blocked state of VLANs.

- 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

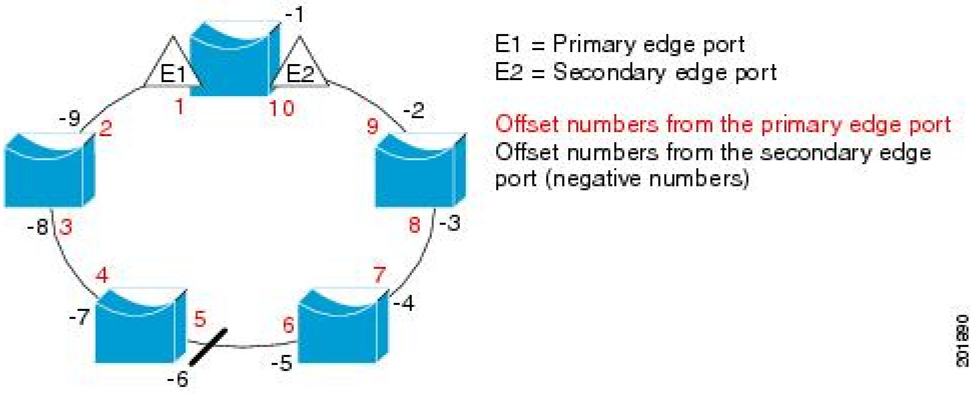
One edge port in the REP segment acts as the primary edge port; and another as the secondary edge port. It is the primary edge port that always participates in VLAN load balancing in the segment. REP VLAN balancing is achieved by blocking some VLANs at a configured alternate port and all the other VLANs 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


-1
E1
E2
1
10
-9
2
9
-2
-8
3
8
-3
4
7
-7
5
6
-4
-6
-5
E1 = Primary edge port
E2 = Secondary edge port
Offset numbers from the primary edge port
Offset numbers from the secondary edge port (negative numbers)
201890

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.

Resilient Ethernet Protocol (REP) Negotiated

Note REP Negotiated works only on uplink ports.

REP and Spanning Tree Protocol (STP) are two different loop avoidance protocols. REP has certain advantages over STP in terms of convergence time. REP can be configured to run in a ring topology in such a way that it can provide the redundant path in case of a single link failure in the ring.

Cisco switches are STP enabled by default. If a switch that is STP enabled is inserted in an already running REP ring (for addition of a new node or replacement of existing node) the following conditions apply:

- The new switch will cause a break in the REP ring.
- The new switch will not be able to communicate over the ring until it is configured to be part of the REP ring.

The REP Negotiated feature tries to solve these issues by negotiating the REP status with the peers. The following table identifies when REP Negotiation events will trigger and the action to take. There are two events: both peers are negotiating, and neither peer is negotiating.

SELF REP Negotiated	PEERS REP Negotiated	Event Triggered	Action
True	True	REPN	Configure REP
True	False	REPNN	Configure STP
False	X	REPNN	Remain in STP

This feature depends on 3 different protocols to get the required data and decide the correct configuration. The different protocols involved, and their purpose is given below:

- **STP**: By default, STP is enabled on all the ports on the Cisco Switch.
- **REP**: The customer network is configured to form a REP ring to provide better convergence time and redundancy.

- **Cisco Discovery Protocol (CDP)**: The feature depends on user defined TLVs sent through CDP messages to negotiate the correct (STP or REP) configuration for the interface.

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.

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.

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 link drop detection and recovery times are between 750 ms and 350 ms. 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 beacon mechanism is implemented to trigger faster link failure detection (within 5-10 ms) when a REP interface is configured for REP Fast mode. The switch 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 message is sent to the switch.

REP Fast works on a per link basis. It does not impact the REP Protocol. REP Fast requires both ends of the link to support REP Fast to work. REP Fast can be used on any interface link pair configured for REP, but it was created to solve an issue on Gigabit copper links. REP Fast speeds up detection of the link failure on Gigabit copper interfaces.

A REP Ring can have a mix of normal REP links and links with REP Fast. Interfaces with REP Fast will transmit 3000 packets a second as part normal operation. REP Fast enablement does not impact REP ring size since it operates only on the pair of interfaces configured for it. Because REP Fast has to generate Beacon frames, only six interfaces on a single REP node can be configured for REP Fast at a time.

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

To configure REP Fast, see Configuring REP Fast, on page 61.

REP Zero Touch Provisioning

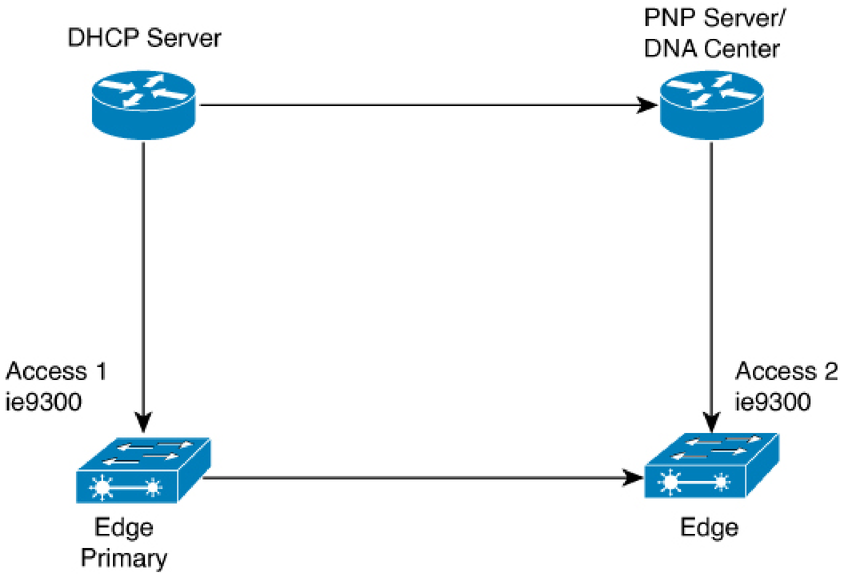
Before a network device such as a router or a switch is deployed online and fully functional, a fair amount of manual configuration is required. Zero Touch Provisioning (ZTP) technologies automate these processes, bringing up network devices into a functional state with minimal to no manual configuration. The Cisco Network Plug and Play (PnP) and Autoinstall Day Zero solutions provide a simple, secure, unified, and integrated offering for enterprise and industrial network customers to ease device rollouts for provisioning updates to an existing network. However, PnP does not support Resilient Ethernet Protocol (REP) due to the way REP is designed. Prior to the REP ZTP feature, REP ring provisioning for Day Zero required manual intervention. The REP ZTP feature introduces a new type-length-value (TLV) extension into the REP LSL packets to support configuring REP rings with zero-touch technologies.

REP and Day Zero

In a typical switch deployment using ZTP, the switch, with no startup configuration in the NVRAM, triggers the Cisco Open Plug-n-Play (PnP) agent to initiate a DHCP discovery process. This process acquires the IP configuration required for the switch from the DHCP server. The DHCP server can be configured to insert additional information in a DHCP message using vendor specific option 43. After the DHCP server receives a DHCP DISCOVER message with option 60 and the string "cisco pnp" from the switch, the DHCP server sends the IP address or hostname of the PnP server to the requesting switch. When the switch receives the DHCP response, the PnP agent extracts the option 43 from the response to get the IP address or the hostname of the PnP server. The PnP agent on the switch then uses this IP address or hostname to communicate with the PnP server. Finally, the PnP server downloads the required Day Zero configuration to the switch to complete the provisioning.

The example shown in the following diagrams illustrates REP ring provisioning on Day Zero, prior to the introduction of REP ZTP.

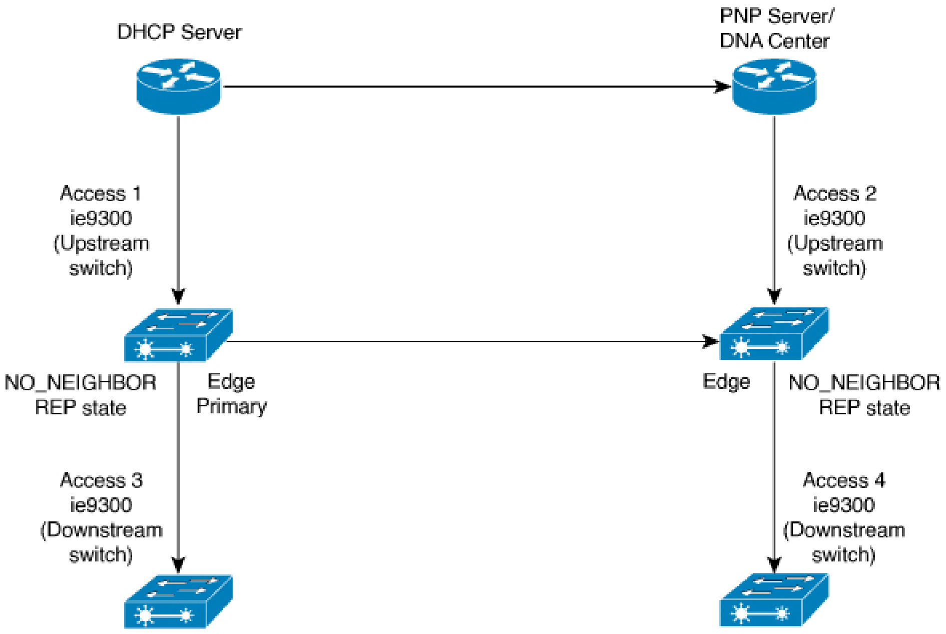
Figure 11: Adding Edge Nodes to the REP Ring



Note The DHCP Server and the PnP Server/DNA Center are not part of the REP ring.

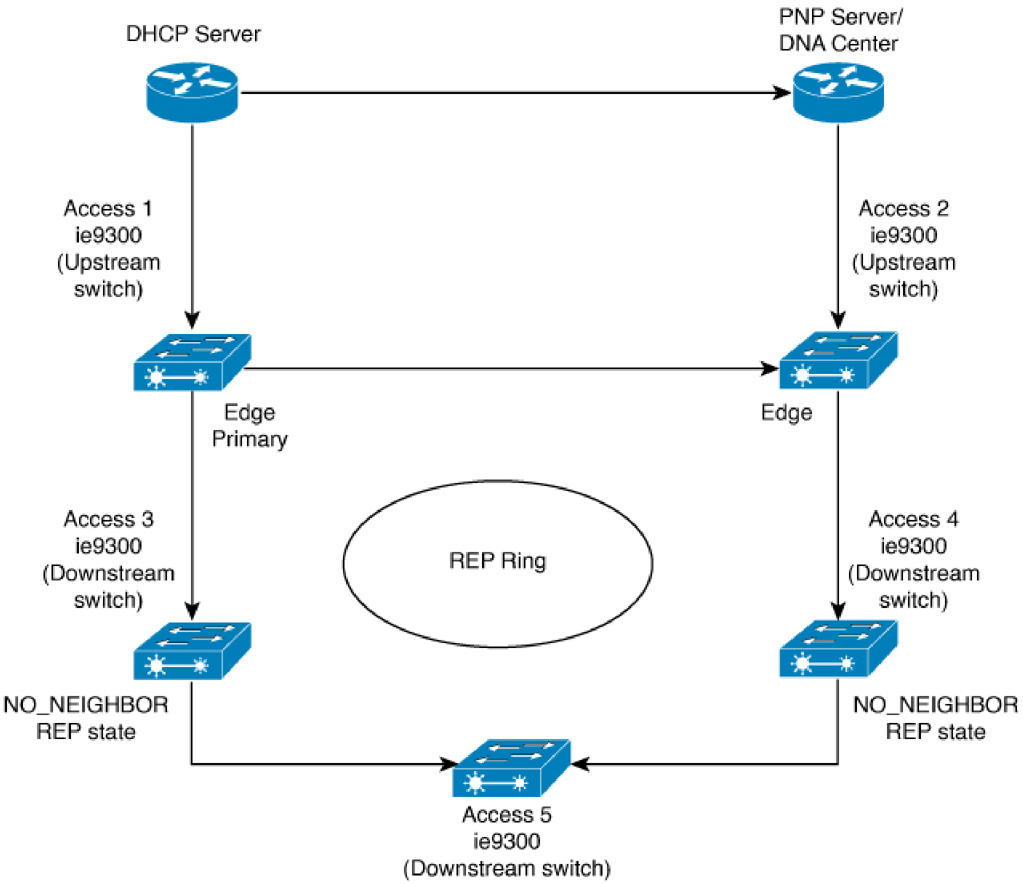
The first set of nodes to be provisioned are Access 1 and Access 2 in the diagram. These are the 2 edge nodes of the REP ring. Note that PnP has configured the downlink port as primary edge on Access 1 and secondary edge on Access 2.

Figure 12: Adding Downstream Nodes



When either Access 3 or Access 4 are powered on, the REP edge primary port starts the REP protocol negotiation and discovers that the neighbor port is not a REP enabled port. (Recall that the switch will be added to the REP ring only after PnP provisioning, for which it needs to first contact the DHCP server as explained earlier.) When an upstream switch port has REP configured and a downstream switch is getting on-boarded with PnP, the REP port goes into the NO_NEIGHBOR state because it is not able to discover its REP peer. In the NO_NEIGHBOR state, REP blocks all the VLANs on that port. This means that the DHCP discovery message from the new switch on the PnP startup VLAN is dropped by the upstream switch because its REP state is NO_NEIGHBOR. The same sequence of blocked ports continues for all new switches added to the REP ring (see Access 5 in figure below).

Figure 13: NO_NEIGHBOR REP State



REP ZTP Overview

The REP ZTP enhancements require that both the upstream and the downstream switches support the feature. When the new downstream switch is powered on, it initiates PNP/autoinstall. The upstream switch's interface is configured for REP and blocks the interface to the downstream switch because the downstream switch is not REP by default (the upstream switch is in REP_NO_NEIGHBOR state).

Even though the interface on the upstream switch is blocked, it will transmit REP LSL packets to the downstream switch. This is normal. With the enhancement of the REP ZTP feature, the downstream switch will start transmitting REP LSL packets with a new TLV to inform the upstream switch that its neighbor is attempting PNP provisioning.

When the upstream switch reads this REP LSL with the new TLV, it will unblock the interface for the PNP startup VLAN only. All other VLANs for which the upstream interface is a member continue to be blocked. Because the upstream switch is forwarding packets on the PNP startup VLAN for this interface, the downstream switch can complete the PNP process.

The intent of this feature is to allow new switches to join a REP ring with no manual intervention. The interface on the upstream switch keeps the startup VLAN unblocked until the downstream switch has received its configuration and has configured its own interface for REP. If there's a failure in the PNP proccess, the interface on the upstream switch reverts to blocking on the PNP startup VLAN. If the configuration received by the downstream switch does configure the interface for REP, the upstream switch reverts to blocking the PNP startup VLAN.

The downstream behavior to transmit the REP LSL with new TLV to request the PnP startup VLAN be unblocked is the default behavior for switches with no startup configuration. For security purposes, the upstream switch must have the interface to the downstream switch explicitly enabled to put the PnP startup VLAN into unblocked state. The interface level command is **rep ztp-enable**. See Configuring REP ZTP, on page 62.

Note The upstream switch can be part of multiple REP rings and thereby connected to multiple downstream neighbours. The PnP startup VLAN is unblocked only on the interfaces to which the downstream switch is connected.

REP Segment-ID Autodiscovery

Resilient Ethernet Protocol (REP) Segment-ID Autodiscovery enables automatic configuration and continued static configuration of segment IDs in REP segments. The feature is supported on Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches beginning with the Cisco IOS XE Cupertino 17.9.x release.

A REP segment is a chain of ports that are connected to each other and configured with a segment ID. Forming multiple REP segments statically by configuring each port of the device is a manual task, and any mismatch in configuring the segment ID leads to convergence issues. However, REP Segment-ID Autodiscovery adds new CLI commands to enable a switch to learn and retain segment ID information automatically.

You can use REP Segment-ID Autodiscovery in several different scenarios. You can insert a new switch into an existing REP segment or in a new REP segment that you build yourself. The feature is ideal for multiple REP ring deployments when incorrect REP Segment IDs might be entered manually. Such errors can occur when deploying multiple REP rings from the same REP seed node.

See the following sections in this guide for more information:

- REP Segment-ID Autodiscovery Deployment
- Configuring REP Segment-ID Autodiscovery

REP Segment-ID Autodiscovery Deployment

You can configure REP Segment-ID Autodiscovery when you add a switch to a REP segment or when you create a REP segment. In either case, the feature reduces the amount of manual configuration that you must do.

Adding a new Switch to an REP Segment

When you add a switch to an existing REP segment, you enable autodiscovery by entering the **rep autodisc** command on the switch interfaces connecting to the upstream and downstream switches.

When the new switch is connected to the upstream and downstream switches, the upstream and downstream switches send CDP packets with REP segment ID information to the new switch interfaces. You enter the command **rep segment auto** on the new switch interfaces so they can learn the segment ID.

Building a new REP Segment

When you build a closed REP segment, you must start with a static REP segment ID configuration from an edge device. The primary and secondary edge devices in a closed segment are on the same switch. When you build an open REP segment, you must start a static REP segment ID configuration from both primary and secondary edge devices.

The remaining steps are the same for both closed and open REP segments. You bring up the next node in the REP ring. You then add any next new node between these two switches for autodiscovery to work correctly.

Building a REP Segment with Uplinks

When you build a ring segment with uplinks (daisy chain), you must start with a static REP segment ID configuration from the REP edge node. Connect the next device to one of the uplinks to the edge node, and enable autodiscovery on the connected uplink. Because of port pairing support, the same REP configuration is duplicated on the paired uplink port.

When the next device is connected with the uplink, the process repeats to bring the REP segment in a daisy chain manner. Each new REP node automatically joins the ring by learning the REP Segment ID from the node above it. For a REP open ring, the last device on the segment is an edge device with static REP configuration.

REP Segment-ID Autodiscovery Limitations

The following are restrictions for the REP Segment-ID Autodiscovery feature:

- The only supported port-pairing is uplinks Gi1/1 and Gi1/2. No predefined port pairing is supported for downlinks.

  If you configure a REP segment on a downlink port, the switch receives the segment ID from the upstream switch, and the partner downlink port is connected to the same segment. However, the switch does not pass the segment ID to its partner port. Instead, you must explicitly configure the partner port of the downlink pair.

- The REP Segment-ID Autodiscovery feature is not supported when you insert an edge node into the existing segment. You must configure static or manual REP segment ID on primary and secondary edge devices.

- If you insert a new switch between two switches that are part of a segment, you must connect the new switch interfaces to the interfaces of existing switches that transmit the same segment ID. Any incorrect connections to other interfaces of the existing switches leads to segment failure.

  For example, assume gi1/1 of switch1 and gi1/2 of switch2 are connected as a part an existing segment, and switch3 is inserted between these two switches. In such a case, you must ensure that the interfaces are connected to gi1/1 of switch1 and gi1/2 of switch2 to include switch3 as a part of the same segment.

- If you configure REP automatically on an interface with the **rep segment auto** command, and you remove the REP configuration with the **no rep segment** command or overwrite it with the **rep segment** <> command, you cannot configure REP automatically again with the **rep segment auto** command. Instead, you must shut down the interface, bring it up, and then enter the **rep segment auto** command.
- REP Segment ID Autodiscovery depends on the CDP protocol. The feature does not support EtherChannel links.

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 Fast is disabled by default.
- REP Zero Touch Provisioning is enabled by default at the global level and disabled at the interface level.

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 interfaces rep** command output, the Port Role for this port shows as “Fail Logical Open;” and 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.
- 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 must 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.
  - **lsl-age-timer** is intended to be used when normal link down detection will be too slow for convergence time.

    FastEthernet and fiber connections do not need **lsl-age-timer**. Gigabit copper can use REP Fast instead of **lsl-age-timer**.
- You cannot configure REP ports 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 26 REP segments per switch.
- There is no limit to the size of a REP ring. REP ring sizes greater than 20 nodes may not achieve sub 50ms convergence. The use of REP ZTP or REP Segment ID Autodiscovery limits a single node to only three REP segments.

REP Fast

- REP fastmode cannot co-exist with MACsec. This restriction applies to the IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches.

  REP fastmode sends a beacon before a link comes up, for faster convergence, and it keeps the port down until the beacon is detected. MKA negotiation cannot take place before the link is up, and by design MACsec configuration drops everything except for EAPOL packets until the MKA session is secured. This means that with the combination of REP fastmode and MACsec, REP fast beacons are dropped and MKA negotiation does not occur.

  MACsec with REP works as expected.

REP Zero Touch Provisioning

- REP ZTP requires the PnP feature to be present on Cisco Catalyst IE 200, IE3300, and IE3400 series switches.
- REP behavior during the NO_NEIGHBOR state is modified beginning in in Cisco IOS XE 17.8.1 and later. This transient state change in port forwarding behavior in NO_NEIGHBOR state allows a DHCP request message to reach a DHCP server and unblock PnP provisioning of a new switch. There should not be any impact to the REP state machine after PnP completion.
- The changes in REP behavior during the NO_NEIGHBOR state apply only to REP Zero Touch Provisioning (ZTP) in Cisco IOS XE 17.8.1 and later. If the PnP feature is not present, normal REP functionality should work as expected.
- The REP ZTP feature coexists with REP bpduleak/negotiated feature on fiber uplink ports.
- The REP ZTP feature is not supported on EtherChannel interfaces for day 0 on an upstream switch because EtherChannel is not present on the downstream interface by default. REP ZTP works only on physical interfaces.
- REP ZTP is supported on both copper (downlink) and fiber (uplink) interfaces.
- REP ZTP is interoperable only with other IE switching products running IOS XE that claim REP ZTP support.

Configuring 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.

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.
- The administrative VLAN cannot be the RSPAN VLAN.

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**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.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show interface** [*interface-id*] **rep detail** **Example:** `Device# show interface gigabitethernet1/1 rep detail`	(Optional) Verifies the configuration on a REP interface.
Step 6	**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:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# 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 4	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Configures the interface as a Layer 2 trunk port.
Step 5	**rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**] **Example:** `Device(config-if)# 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** command in privileged EXEC mode.

	Command or Action	Purpose
		• (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.

Step 6	**rep stcn** {**interface** *interface id* | **segment** *id-list* | **stp**}
	Example:
	`Device(config-if)# 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 7	**rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}
	Example:
	`Device(config-if)# 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.

	Command or Action	Purpose
		- **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.
Step 8	**rep preempt delay** *seconds* **Example:** `Device(config-if)# 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 9	**rep lsl-age-timer** *value* **Example:** `Device(config-if)# 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 10	**end**
	Example:
	`Device(config-if)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 11	**show interface** [*interface-id*] **rep** [**detail**] **Example:** `Device# show interface gigabitethernet1/1 rep detail`	(Optional) Displays the REP interface configuration.
Step 12	**copy running-config startup-config** **Example:** `Device# 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.

Procedure

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

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.

Procedure

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

	Command or Action	Purpose
Step 3	**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 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Displays the running configuration, which can be used to verify the REP trap configuration.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the switch startup configuration file.

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 segment 1 edge
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
```

Configuring REP ZTP

To configure REP ZTP, you enable or disable it at the global level and the interface level. The default states are:

- Global level: Enabled
- Interface level: Disabled

You must explicitly enable the feature at the interface level on the upstream device interface connected to the downstream device. When enabled, only that interface will receive notification from the downstream switch to block or unblock the PnP startup VLAN.

Note When applying configuration from DNAC or PNP server user must explicitly add this CLI configuration in the configuration template for the feature to be enabled.

Step 1 Enter global configuration mode:

```
Switch# configure terminal
```

Step 2 Globally enable REP ZTP:

```
Switch(config)# rep ztp
```

Use the no form of the command to disable REP ZTP: `Switch(config)# no rep ztp`

Step 3 Enter interface configuration mode on the upstream device interface that is connected to the downstream device:

```
Switch(config)# interface <interface-name>
```

Step 4 Enable REP ZTP on the interface:

```
Switch(config-if)#rep ztp-enable
```

Use the no form of the command to disable REP ZTP on the interface: `Switch(config-if)#no rep ztp-enable`

Example

The following example shows the minimum configuration required to enable the REP ZTP feature on the upstream device interface that is connected to a downstream device.

```
Switch#show running-config interface gigabitEthernet 1/2
Building configuration...

Current configuration : 93 bytes
!
interface GigabitEthernet1/2
 switchport mode trunk
 rep segment 100
 rep ztp-enable
end
```

Configuring REP Segment-ID Autodiscovery

You use CLI commands tor REP Segment-ID Autodiscovery. One enables or disables autodiscovery on a REP switch, and one configures new interfaces so the switch learns the segment-ID. You also use CLI commands to view the status of the feature on the segment.

Enable REP Segment-ID Autodiscovery

REP Segment-ID Autodiscovery is enabled by default. However, you can re-enable it on the switch upstream and downstream interfaces.

Enable REP Segment-ID Autodiscovery on the switch.

Example:

```
switch(config)#rep autodisc
```

You disable REP Segment-ID Autodiscovery by entering the following command:

```
switch(config)#no rep autodisc
```

What to do next

You can check the status of REP Segment-ID Autodiscovery. See the section View Feature Status, on page 64 in this guide.

Configure the Interfaces

Configure the interface on the newly inserted switch so that downstream nodes to participate in the REP segment. The **rep segment auto** command automatically fetches the segment ID from the upstream switch.

Before you begin

Ensure that the REP segment ID is configured on the primary and secondary edge devices. You configure the segment ID by entering the command **rep segment** *segment_id* **edge**, in which *segment_id* is the segment ID of the ring to be propagated through CDP packet to the neighboring device when connected.

Enable the switch to learn the segment ID.

Example:

```
switch(config)#int gig1/1
switch(config-if)#rep seg auto
```

Note Cisco IOS XE Cupertino 17.9.1 and later releases support port pairing for uplinks. That is, when you configure **rep segment auto** on one of the uplinks, the same configuration is made automatically on the other uplink.

However, port pairing is *not* supported for downlinks. You must configure each downlink separately.

Following example shows the minimum configuration to enable the feature on an interface on the upstream device switch. The upstream device with an explicit REP segment is typically an edge switch.

```
switch#show running-config interface gigabitEthernet 1/3
Building configuration...

Current configuration : 93 bytes
!
interface GigabitEthernet1/3
 switchport mode trunk
 rep segment auto 1
```

The following example shows the minimum configuration to enable the feature on an interface on the downstream switch interface. Enter the command **`show running-config interface`** *`interface_id`* to confirm that the downstream switch knows to expect to receive its REP segment through CDP message.

```
switch#show running-config interface gigabitEthernet 1/2
Building configuration...

Current configuration : 93 bytes
!
interface GigabitEthernet1/2
 switchport mode trunk
 rep segment auto
end
```

You disable the ability of the switch to learn the segment ID by entering the following command:

```
switch(config-if)#no rep segment
```

What to do next

You can check the status of REP Segment-ID Autodiscovery. See the section View Feature Status, on page 64 in this guide.

View Feature Status

You can use CLI commands to check the status of REP Segment-ID Autodiscovery on the segment.

Confirm that REP Segment-ID Autodiscovery is globally enabled on the switch.

Example:

```
switch#show interfaces rep detail
REP Segment Id Auto Discovery Status: Enabled
```

The following examples show other commands for checking the status of REP Segment-ID Autodiscovery:

- The following example shows the command to check if the feature is globally disabled on a device:

```
switch#show interfaces rep detail
REP Segment Id Auto Discovery Status: Disabled
```

- The following example shows the command to confirm that the segment ID on interface is configured automatically:

```
switch#show interfaces rep detail
REP Segment Id Type: Auto
```

- The following example shows the command to confirm that the segment ID on the interface is configured manually:

```
witch#show interfaces rep detail
REP Segment Id Type: Manual
```

Monitoring Resilient Ethernet Protocol Configurations

This is an example of the output for the **show interface** [*interface-id*] **rep** [**detail**] command. This display shows the REP configuration and status on an uplink port.

```
Device# show interfaces GigabitEthernet1/4 rep detail

GigabitEthernet1/4 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
REP-ZTP Status: Disabled
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
```

This is an example of the output for the **show interface** [*interface-id*] **rep** [**detail**] command. This display shows the REP configuration and status on a downlink port.

```
Device#show interface GigabitEthernet1/5 rep detail
GigabitEthernet1/5   REP enabled
Segment-id: 1 (Segment)
PortID: 019B380E4D9ACAC0
Preferred flag: No
Operational Link Status: NO_NEIGHBOR
```

```
Current Key: 019B380E4D9ACAC0696B
Port Role: Fail No Ext Neighbor
Blocked VLAN: 1-4094
Admin-vlan: 1
REP-ZTP Status: Disabled
Preempt Delay Timer: 100 sec
LSL Ageout Timer: 2000 ms
LSL Ageout Retries: 5
Configured Load-balancing Block Port: 09E9380E4D9ACAC0
Configured Load-balancing Block VLAN: 1-100
STCN Propagate to: segment 25
LSL PDU rx: 292, tx: 340
HFL PDU rx: 0, tx: 0
BPA TLV rx: 0, tx: 0
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 0, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 0, tx: 0
```

This is an example for the **show rep topology** [**segment** *segment-id*] [**archive**] [**detail**] command. This display shows the REP topology information for all the segments.

```
Device# show rep topology

REP Segment 1
BridgeName       PortName     Edge Role
---------------- ----------   ---- ----
10.64.106.63     Gi1/4        Pri  Open
10.64.106.228    Gi1/4             Open
10.64.106.228    Gi1/3             Open
10.64.106.67     Gi1/3             Open
10.64.106.67     Gi1/4             Alt
10.64.106.63     Gi1/4        Sec  Open

REP Segment 3
BridgeName       PortName     Edge Role
---------------- ----------   ---- ----
10.64.106.63     Gi1/1        Pri  Open
SVT_3400_2       Gi1/3             Open
SVT_3400_2       Gi1/4             Open
10.64.106.68     Gi1/2             Open
10.64.106.68     Gi1/1             Open
10.64.106.63     Gi1/2      Sec  Alt
```

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

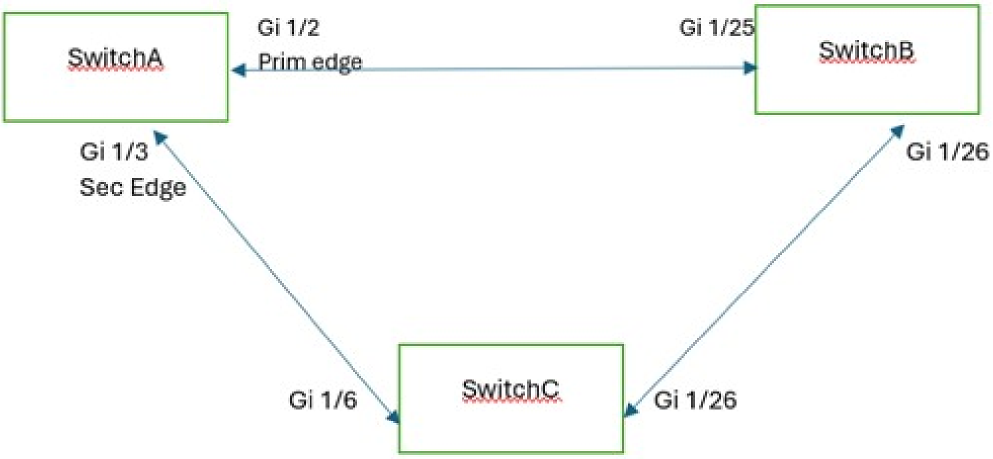
```
Switch# sh platform rep beacon GigabitEthernet 1/4
Beacon RX : 43984
Beacon TX : 46826
```

Investigating Broken Links

This section explains how to interpret **show rep topology** output if a link failure occurs.

Here is an example of a REP closed ring:

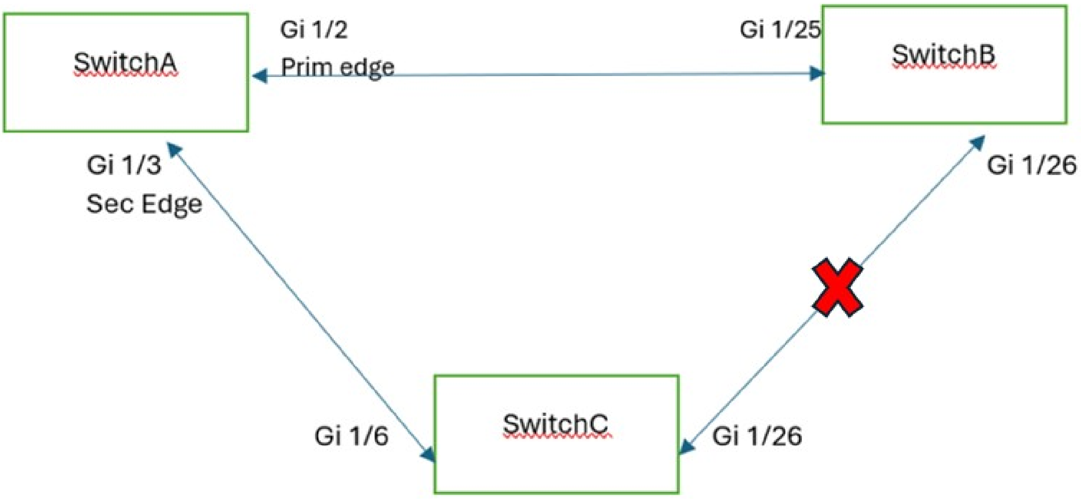
Figure 14: REP Closed Ring Topology



```
SWITCHA#sh rep topology
REP Segment 1
BridgeName                       PortName   Edge Role
-------------------------------- ---------- ---- ----
SWITCHA                          Gi1/2      Pri  Open
SWITCHB                          Gi1/25          Open
SWITCHB                          Gi1/26          Open
SWITCHC                          Gi1/26          Open
SWITCHC                          Gi1/6           Open
SWITCHA                          Gi1/3      Sec  Alt
```

Here is an example where the connection between SwitchB and SwitchC is down:

Figure 15: REP Closed Ring Topology with Link Failure



```
SWITCHA#sh rep topology
REP Segment 1
Warning: REP detects a segment failure, topology may be incomplete

BridgeName                       PortName   Edge Role
-------------------------------- ---------- ---- ----
SWITCHA                          Gi1/2      Sec  Open
SWITCHB                          Gi1/25          Open
SWITCHB                          Gi1/26          Fail
```

The **show rep topology** output relies on a database built using Edge Port Advertisement (EPA) packets. Each node in the ring is expected to receive two EPA packets, one each from the Primary and Secondary edge ports. Each port adds its own topology information to the topology information that it received.

If a failure in the topology occurs, depending on where the link failure is in relation to a node's position, the node will have a limited view of the topology starting from the connected edge port up to the node (as shown in the example **show rep topology** output above where a failure has occurred). In this case the node fails to transmit the EPA packets, resulting in each node showing different topology information in the **show rep topology** output.

Note This behavior is limited to the **show rep topology** command output only. The data path is not affected.

Displaying REP ZTP Status

Use the **show** command to identify the state of REP ZTP on an interface. In the following example, the feature is disabled on interface GigabitEthernet 1/1 and it is enabled on interface GigabitEthernet 1/2. The status of **pnp_startup_vlan** is "Blocked".

Step 1 In priviledged exec mode, enter:

show interfaces rep detail

Example:

```
GigabitEthernet1/1   REP enabled
Segment-id: 100 (Segment)
PortID: 00016C13D5AC4320
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 00026C13D5AC43209DAB
Port Role: Open
Blocked VLAN: <empty>
Admin-vlan: 1
REP-ZTP Status: Disabled
REP Segment Id Auto Discovery Status: Enabled
REP Segment Id Type: Manual
Preempt Delay Timer: disabled
LSL Ageout Timer: 5000 ms
LSL Ageout Retries: 5
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 382, tx: 297
HFL PDU rx: 0, tx: 0
BPA TLV rx: 1, tx: 19
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 95, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 95, tx: 95

GigabitEthernet1/2   REP enabled
Segment-id: 100 (Segment)
PortID: 00026C13D5AC4320
Preferred flag: No
Operational Link Status: NO_NEIGHBOR
Current Key: 00026C13D5AC43209DAB
Port Role: Fail No Ext Neighbor
Blocked VLAN: 1-4094
Admin-vlan: 1
REP-ZTP Status: Enabled
REP-ZTP PnP Status: Unknown
REP-ZTP PnP Vlan: 1
REP-ZTP Port Status: Blocked
REP Segment Id Auto Discovery Status: Enabled
REP Segment Id Type: Manual
Preempt Delay Timer: disabled
LSL Ageout Timer: 5000 ms
LSL Ageout Retries: 5
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 11, tx: 11
HFL PDU rx: 0, tx: 0
BPA TLV rx: 0, tx: 0
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 0, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 0, tx: 0
```

Step 2 Use the show command again to display the status of **pnp_startup_vlan**.

When the downstream device is booted up, it sends notification to the connected upstream switch interface to unblock the **pnp_startup_vlan** for it to get the DHCP IP address and further establish communication with the PNP server or DNAC. The show command indicates the status as "Unblocked".

The following syslogs on the upstream switch notify you about FWD and BLK of ports. There are no syslogs in the downstream switch as PnP takes control of the console and no syslogs can be printed on the console.

```
REP-6-ZTPPORTFWD: Interface GigabitEthernet1/2 moved to forwarding on ZTP notification
```

```
REP-6-ZTPPORTBLK: Interface GigabitEthernet1/2 moved to blocking on ZTP notification
```

Example:

```
Switch#show interfaces rep detail
GigabitEthernet1/1   REP enabled
Segment-id: 100 (Segment)
PortID: 00016C13D5AC4320
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 00026C13D5AC43209DAB
Port Role: Open
Blocked VLAN: <empty>
Admin-vlan: 1
REP-ZTP Status: Disabled
REP Segment Id Auto Discovery Status: Enabled
REP Segment Id Type: Manual
Preempt Delay Timer: disabled
LSL Ageout Timer: 5000 ms
LSL Ageout Retries: 5
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 430, tx: 358
HFL PDU rx: 0, tx: 0
BPA TLV rx: 1, tx: 67
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 107, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 107, tx: 108

GigabitEthernet1/2   REP enabled
Segment-id: 100 (Segment)
PortID: 00026C13D5AC4320
Preferred flag: No
Operational Link Status: NO_NEIGHBOR
Current Key: 00026C13D5AC43209DAB
Port Role: Fail No Ext Neighbor
Blocked VLAN: 1-4094
Admin-vlan: 1
REP-ZTP Status: Enabled
REP-ZTP PnP Status: In-Progress
REP-ZTP PnP Vlan: 69
REP-ZTP Port Status: Unblocked
REP Segment Id Auto Discovery Status: Enabled
REP Segment Id Type: Manual
Preempt Delay Timer: disabled
LSL Ageout Timer: 5000 ms
LSL Ageout Retries: 5
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 32, tx: 40
HFL PDU rx: 0, tx: 0
BPA TLV rx: 0, tx: 0
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 0, tx: 0
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 0, tx: 0
```

Step 3 Use the **show platform hardware l2 stp** command to check the interface state of the PnP startup VLAN:

Example:

```
Switch#show platform hardware l2 stp asic-num 0 vlan-id 69 □[PnP Vlan]
------------------------STP TABLE START------------------------
---------------------------------------------------------------
VlanId:1 StpId:0 MemberPort:3 StpState:FORWARDING
VlanId:1 StpId:0 MemberPort:7 StpState:FORWARDING
VlanId:1 StpId:0 MemberPort:25 StpState:FORWARDING
--------------------------------------------------------------
------------------------STP TABLE END-------------------------
```

Step 4 (Optional) Use the following debug commands to troubleshoot REP ZTP:

- **debug rep lslsm**: This command helps you understand LSL state machine events in the NO_NEIGHBOR state.
- **debug rep packet**: Use this command to dump LSL packets with the REP ZTP LSL TLV to check the PnP status on the peer client node.

Additional References for Resilient Ethernet Protocol

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use the Cisco MIB Locator found at: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/MainServlet

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.	https://www.cisco.com/c/en/us/support/index.html?ts=AZ6NHYKB9WRPLYAVGST1587714574492

Feature History

Feature Name	Release	Feature Information
REP Zero Touch Provisioning	Cisco IOS XE 17.8.1	Initial support on Cisco Catalyst IE 3200, 3300, and 3400
REP Negotiated	Cisco IOS XE 16.12.1	Initial support on Cisco Catalyst IE 3200, 3300, and 3400
REP Fast	Cisco IOS XE 16.11.1	Initial support on Cisco Catalyst IE 3200, 3300, and 3400

CHAPTER 4

Common Industrial Protocol (CIP)

Information About CIP

The Common Industrial Protocol (CIP) is an industrial protocol for industrial automation applications. Previously known as Control and Information Protocol, CIP encompasses a comprehensive suite of messages and services for the collection of manufacturing automation applications - control, safety, synchronization, motion, configuration and information. It is supported by Open DeviceNet Vendors Association (ODVA), an organization that supports network technologies based upon CIP such as DeviceNet, EtherNet/IP, CIP Safety and CIP Sync. CIP allows users to integrate these manufacturing applications with enterprise-level Ethernet networks and the Internet.

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** | **dlr** | **errors** | **event** | **file** | **io** | **packet** | **request response** | **security** | **session** | **socket** }

DETAILED STEPS

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

MB	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

Configuring Flexible Netflow

Prerequisites for Flexible NetFlow

The following are prerequisites for your Flexible NetFlow configuration:

- You must configure a source interface. If you do not configure a source interface, the exporter remains in a disabled state.
- You must configure a valid record name for every flow monitor.
- You must enable IPv6 routing to export the flow records to an IPv6 destination server.
- You must configure IPFIX export protocol for the flow exporter to export netflow records in IPFIX format.
- You are familiar with the Flexible NetFlow key fields as they are defined in the following commands in the Cisco IOS Flexible NetFlow Command Reference:
    - **match datalink**—Datalink (layer2) fields
    - **match interface**—Interface fields
    - **match ipv4**—IPv4 fields
    - **match ipv6**—IPv6 fields
    - **match transport**—Transport layer fields
- You are familiar with the Flexible NetFlow non-key fields:
    - **collect counter**—Counter fields
    - **collect interface**—Interface fields
    - **collect timestamp**—Timestamp fields

IPv4 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding or distributed Cisco Express Forwarding.

IPv6 Traffic

- The networking device must be configured for IPv6 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding IPv6 or distributed Cisco Express Forwarding.
- Restrictions for Flexible NetFlow
- How to Configure Flexible Netflow
- Monitoring Flexible NetFlow
- Configuration Examples for

Restrictions for Flexible NetFlow

The following are restrictions for Flexible NetFlow:

- Microflow policing feature shares the NetFlow hardware resource with FNF.
- Only one flow monitor per interface and per direction is supported .
- Egress netflow is not supported.

Information About Flexible Netflow

Overview

uses flows to provide statistics for accounting, network monitoring, and network planning.

A flow is a unidirectional stream of packets that arrives on a source interface and has the same values for the keys. A key is an identified value for a field within the packet. You create a flow using a flow record to define the unique keys for your flow.

The device supports the feature that enables enhanced network anomalies and security detection. allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields.

All key values must match for the packet to count in a given flow. A flow might gather other fields of interest, depending on the export record version that you configure. Flows are stored in the cache.

You can export the data that gathers for your flow by using an exporter and export this data to a remote system such as a collector. The collector can use an IPv4 address.

You define the size of the data that you want to collect for a flow using a monitor. The monitor combines the flow record and exporter with the cache information.

Starting with the Cisco IOS XE 16.12.1 release, Source Group Tag (SGT) and Destination Group Tag (DGT) fields over Flexible NetFlow are supported for IPv6 traffic.

Original NetFlow and Benefits of Flexible NetFlow

Flexible NetFlow allows the flow to be user defined. The benefits of Flexible NetFlow include:

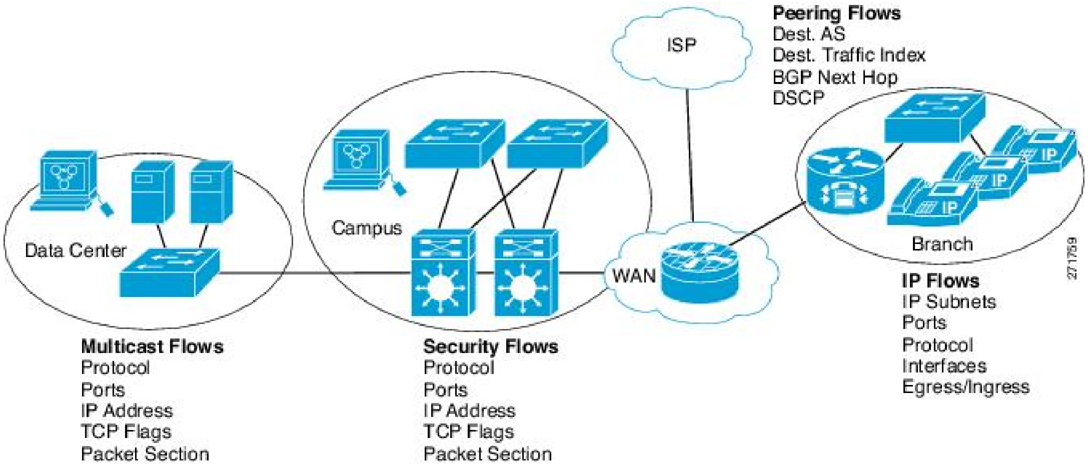
- High-capacity flow recognition, including scalability and aggregation of flow information.
- Enhanced flow infrastructure for security monitoring and dDoS detection and identification.
- New information from packets to adapt flow information to a particular service or operation in the network. The flow information available will be customizable by Flexible NetFlow users.
- Extensive use of Cisco's flexible and extensible NetFlow Version 9.
- A comprehensive IP accounting feature that can be used to replace many accounting features, such as IP accounting, Border Gateway Protocol (BGP) Policy Accounting, and persistent caches.

Flexible NetFlow allows you to understand network behavior with more efficiency, with specific flow information tailored for various services used in the network. The following are some example applications for a Flexible NetFlow feature:

- Flexible NetFlow enhances Cisco NetFlow as a security monitoring tool. For instance, new flow keys can be defined for packet length or MAC address, allowing users to search for a specific type of attack in the network.
- Flexible NetFlow allows you to quickly identify how much application traffic is being sent between hosts by specifically tracking TCP or UDP applications by the class of service (CoS) in the packets.
- The accounting of traffic entering a Multiprotocol Label Switching (MPLS) or IP core network and its destination for each next hop per class of service. This capability allows the building of an edge-to-edge traffic matrix.

The figure below is an example of how Flexible NetFlow might be deployed in a network.

Figure 16: Typical Deployment for Flexible NetFlow



Flexible NetFlow Components

Flexible NetFlow consists of components that can be used together in several variations to perform traffic analysis and data export. The user-defined flow records and the component structure of Flexible NetFlow facilitates the creation of various configurations for traffic analysis and data export on a networking device with a minimum number of configuration commands. Each flow monitor can have a unique combination of flow record, flow exporter, and cache type. If you change a parameter such as the destination IP address for a flow exporter, it is automatically changed for all the flow monitors that use the flow exporter. The same flow monitor can be used in conjunction with different flow samplers to sample the same type of network traffic at different rates on different interfaces. The following sections provide more information on Flexible NetFlow components:

Flow Records

In Flexible NetFlow a combination of key and nonkey fields is called a record. Flexible NetFlow records are assigned to Flexible NetFlow flow monitors to define the cache that is used for storing flow data.

A flow record defines the keys that Flexible NetFlow uses to identify packets in the flow, as well as other fields of interest that Flexible NetFlow gathers for the flow. You can define a flow record with any combination of keys and fields of interest. The supports a rich set of keys. A flow record also defines the types of counters gathered per flow. You can configure 64-bit packet or byte counters. The enables the following match fields as the defaults when you create a flow record:

- match datalink—Layer 2 attributes
- match ipv4—IPv4 attributes
- match ipv6—IPv6 attributes
- match transport—Transport layer fields
- match wireless—Wireless fields

Note The flow monitor with flow record, that contains the CTS field, cannot be attached on the WLAN (SSID).

NetFlow Predefined Records

Flexible NetFlow includes several predefined records that you can use to start monitoring traffic in your network. The predefined records are available to help you quickly deploy Flexible NetFlow and are easier to use than user-defined flow records. You can choose from a list of already defined records that may meet the needs for network monitoring. As Flexible NetFlow evolves, popular user-defined flow records will be made available as predefined records to make them easier to implement.

The predefined records ensure backward compatibility with your existing NetFlow collector configurations for the data that is exported. Each of the predefined records has a unique combination of key and nonkey fields that offer you the built-in ability to monitor various types of traffic in your network without customizing Flexible NetFlow on your router.

Two of the predefined records (NetFlow original and NetFlow IPv4/IPv6 original output), which are functionally equivalent, emulate original (ingress) NetFlow and the Egress NetFlow Accounting feature in original NetFlow, respectively. Some of the other Flexible NetFlow predefined records are based on the aggregation cache schemes available in original NetFlow. The Flexible NetFlow predefined records that are based on the aggregation cache schemes available in original NetFlow do not perform aggregation. Instead each flow is tracked separately by the predefined records.

User-Defined Records

Flexible NetFlow enables you to define your own records for a Flexible NetFlow flow monitor cache by specifying the key and nonkey fields to customize the data collection to your specific requirements. When you define your own records for a Flexible NetFlow flow monitor cache, they are referred to as *user-defined records*. The values in nonkey fields are added to flows to provide additional information about the traffic in the flows. A change in the value of a nonkey field does not create a new flow. In most cases the values for nonkey fields are taken from only the first packet in the flow. Flexible NetFlow enables you to capture counter values such as the number of bytes and packets in a flow as nonkey fields.

Flexible NetFlow adds a new Version 9 export format field type for the header and packet section types. Flexible NetFlow will communicate to the NetFlow collector the configured section sizes in the corresponding Version 9 export template fields. The payload sections will have a corresponding length field that can be used to collect the actual size of the collected section.

Flexible NetFlow Match Parameters

The following table describes Flexible NetFlow match parameters. You must configure at least one of the following match parameters for the flow records.

Table 6: Match Parameters

Command	Purpose
match datalink {**ethertype** | **mac**}	Specifies a match to datalink or Layer 2 fields. The following command options are available: • **ethertype**—Matches to the ethertype of the packet. • **mac**—Matches the source or destination MAC fields.
match interface {**input**}	Specifies a match to the interface fields. The following command options are available: • **input**—Matches to the input interface.
match ipv4 {**destination** | **protocol** | **source** | **tos**}	Specifies a match to the IPv4 fields. The following command options are available: • **destination**—Matches to the IPv4 destination address-based fields. • **protocol**—Matches to the IPv4 protocols. • **source**—Matches to the IPv4 source address based fields. • **tos**—Matches to the IPv4 Type of Service fields.
match ipv6 {**destination** | **protocol** | **source** | **traffic-class**}	Specifies a match to the IPv6 fields. The following command options are available: • **destination**—Matches to the IPv6 destination address-based fields. • **protocol**—Matches to the IPv6 payload protocol fields. • **source**—Matches to the IPv6 source address based fields. • **traffic-class**—Matches to the IPv6 traffic class.
match transport {**destination-port** | **source-port**}	Specifies a match to the Transport Layer fields. The following command options are available: • **destination-port**—Matches to the transport destination port. • **source-port**—Matches to the transport source port.

Flexible NetFlow Collect Parameters

The following table describes the Flexible NetFlow collect parameters.

Table 7: Collect Parameters

Command	Purpose
collect interface {**output**}	Collects the fields from the output interface.
collect counter bytes	Configures the number of bytes seen in a flow as a nonkey field and enables collecting the total number of bytes from the flow.
collect counter packets	Configures the number of packets seen in a flow as a nonkey field and enables collecting the total number of packets from the flow.
collect timestamp sys-uptime first	Configures the system uptime for the time the first packet was seen from the flows as a nonkey field and enables collecting time stamps based on the system uptime for the time the first packet was seen from the flows.
collect timestamp sys-uptime last	Configures the system uptime for the time the last packet was seen from the flows as a nonkey field and enables collecting time stamps based on the system uptime for the time the most recent packet was seen from the flows.

Flow Exporters

Flow exporters export the data in the flow monitor cache to a remote system, such as a server running NetFlow collector, for analysis and storage. Flow exporters are created as separate entities in the configuration. Flow exporters are assigned to flow monitors to provide data export capability for the flow monitors. You can create several flow exporters and assign them to one or more flow monitors to provide several export destinations. You can create one flow exporter and apply it to several flow monitors.

NetFlow Data Export Format Version 9

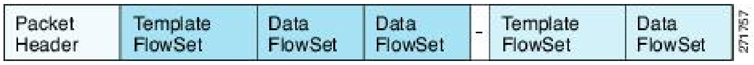
The basic output of NetFlow is a flow record. Several different formats for flow records have evolved as NetFlow has matured. The most recent evolution of the NetFlow export format is known as Version 9. The distinguishing feature of the NetFlow Version 9 export format is that it is template-based. Templates provide an extensible design to the record format, a feature that should allow future enhancements to NetFlow services without requiring concurrent changes to the basic flow-record format. Using templates provides several key benefits:

- Third-party business partners who produce applications that provide collector or display services for NetFlow do not have to recompile their applications each time a new NetFlow feature is added. Instead, they should be able to use an external data file that documents the known template formats.
- New features can be added to NetFlow quickly without breaking current implementations.

- NetFlow is “future-proofed” against new or developing protocols because the Version 9 format can be adapted to provide support for them.

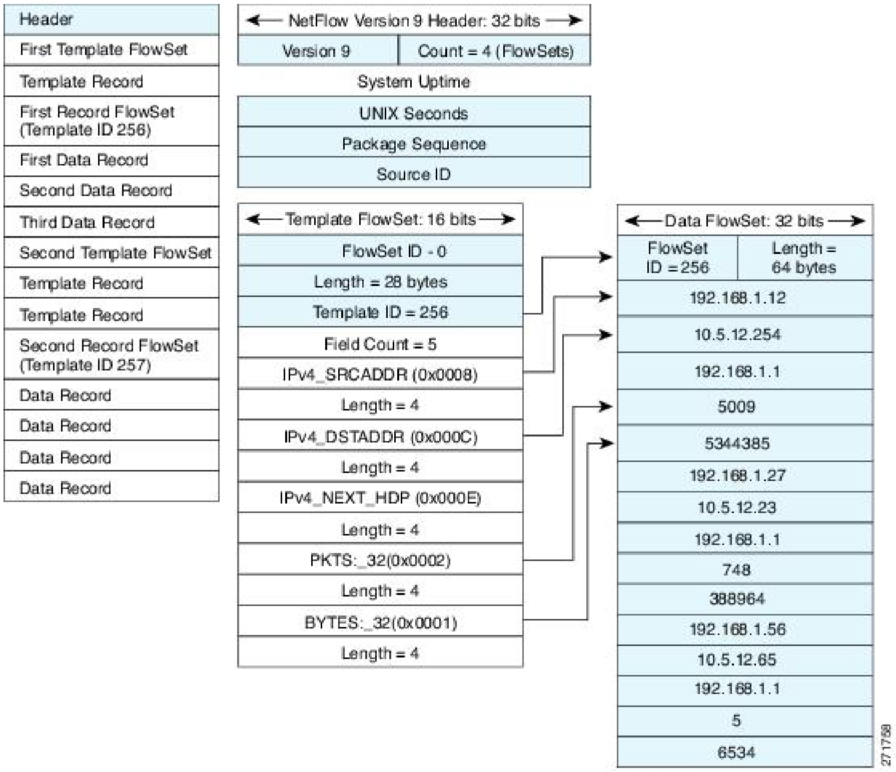
The Version 9 export format consists of a packet header followed by one or more template flow or data flow sets. A template flow set provides a description of the fields that will be present in future data flow sets. These data flow sets may occur later within the same export packet or in subsequent export packets. Template flow and data flow sets can be intermingled within a single export packet, as illustrated in the figure below.

Figure 17: Version 9 Export Packet



NetFlow Version 9 will periodically export the template data so the NetFlow collector will understand what data is to be sent and also export the data flow set for the template. The key advantage to Flexible NetFlow is that the user configures a flow record, which is effectively converted to a Version 9 template and then forwarded to the collector. The figure below is a detailed example of the NetFlow Version 9 export format, including the header, template flow, and data flow sets.

Figure 18: Detailed Example of the NetFlow Version 9 Export Format



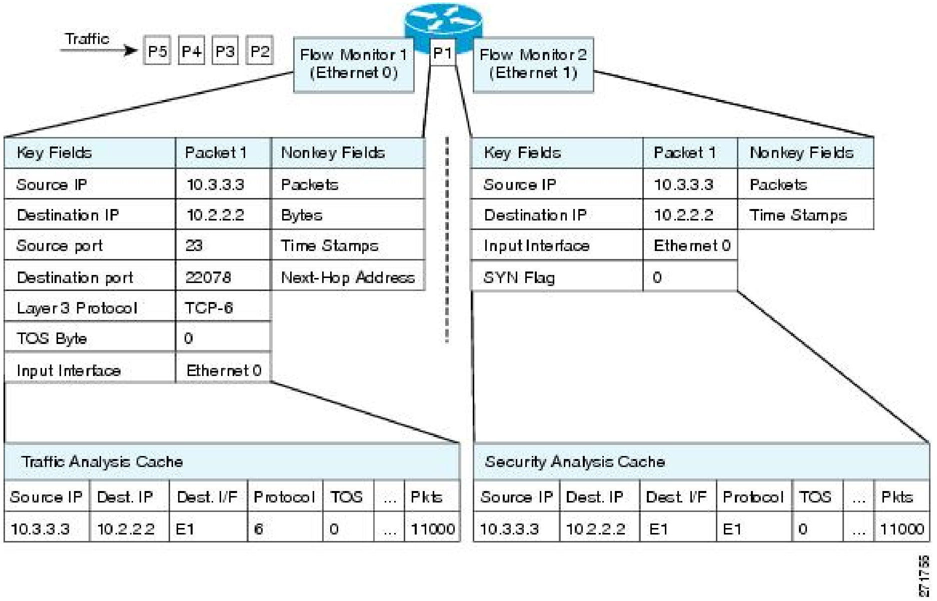
Flow Monitors

Flow monitors are the Flexible NetFlow component that is applied to interfaces to perform network traffic monitoring.

Flow data is collected from the network traffic and added to the flow monitor cache during the monitoring process based on the key and nonkey fields in the flow record.

Flexible NetFlow can be used to perform different types of analysis on the same traffic. In the figure below, packet 1 is analyzed using a record designed for standard traffic analysis on the input interface and a record designed for security analysis on the output interface.

Figure 19: Example of Using Two Flow Monitors to Analyze the Same Traffic



Key Fields	Packet 1	Nonkey Fields
Source IP	10.3.3.3	Packets
Destination IP	10.2.2.2	Bytes
Source port	23	Time Stamps
Destination port	22078	Next-Hop Address
Layer 3 Protocol	TCP-6	
TOS Byte	0	
Input Interface	Ethernet 0	

Key Fields	Packet 1	Nonkey Fields
Source IP	10.3.3.3	Packets
Destination IP	10.2.2.2	Time Stamps
Input Interface	Ethernet 0	
SYN Flag	0	

Traffic Analysis Cache

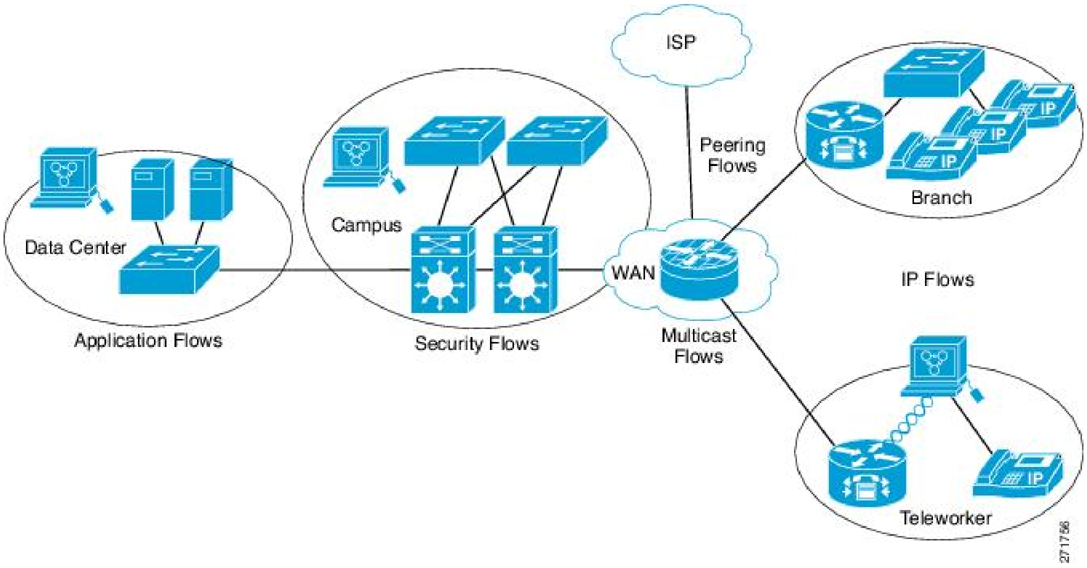
Source IP	Dest. IP	Dest. I/F	Protocol	TOS	...	Pkts
10.3.3.3	10.2.2.2	E1	6	0	...	11000

Security Analysis Cache

Source IP	Dest. IP	Dest. I/F	Protocol	TOS	...	Pkts
10.3.3.3	10.2.2.2	E1	E1	0	...	11000

The figure below shows a more complex example of how you can apply different types of flow monitors with custom records.

Figure 20: Complex Example of Using Multiple Types of Flow Monitors with Custom Records



Normal

The default cache type is “normal”. In this mode, the entries in the cache are aged out according to the timeout active and timeout inactive settings. When a cache entry is aged out, it is removed from the cache and exported via any exporters configured.

Supported Flexible NetFlow Fields

The following tables provide a consolidated list of supported fields in Flexible NetFlow (FNF) for various traffic types and traffic direction.

Note If the packet has a VLAN field, then that length is not accounted for.

Field	Layer 2 In	IPv4 In	IPv6 In	Notes
Key or Collect Fields				
Interface input	Yes	Yes	Yes	If you apply a flow monitor in the input direction: • Use the **match** keyword and use the input interface as a key field. • Use the **collect** keyword and use the output interface as a collect field. This field will be present in the exported records but with a value of 0.
Interface output	—	—	—	If you apply a flow monitor in the output direction: • Use the **match** keyword and use the output interface as a key field. • Use the **collect** keyword and use the input interface as a collect field. This field will be present in the exported records but with a value of 0.

Field	Layer 2 In	IPv4 In	IPv6 In	Notes
Key Fields				
Ethertype	Yes	—	—	
MAC source address input	Yes	Yes	Yes	
MAC source address output	—	—	—	

Field	Layer 2 In	IPv4 In	IPv6 In	Notes
MAC destination address input	Yes	Yes	Yes	
MAC destination address output	—	—	—	
IPv4 TOS	—	Yes	Yes	
IPv4 protocol	—	Yes	Yes	Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv4 source address	—	Yes	—	
IPv4 destination address	—	Yes	—	
Key Fields continued				
IPv6 protocol	—	Yes	Yes	Same as IP protocol. Must use if any of src/dest port, ICMP code/type, IGMP type or TCP flags are used.
IPv6 source address	—	—	Yes	
IPv6 destination address	—	—	Yes	
IPv6 traffic-class	—	Yes	Yes	Same as IP TOS.
source-port	—	Yes	Yes	
destination-port	—	Yes	Yes	
Collect Fields				
Interface output	Yes	Yes	Yes	
Counter bytes	Yes	Yes	Yes	
Counter packets	Yes	Yes	Yes	

Field	Layer 2 In	IPv4 In	IPv6 In	Notes
Timestamp sys-uptime first	Yes	Yes	Yes	
Timestamp sys-uptime last	Yes	Yes	Yes	

Default Settings

The following table lists the Flexible NetFlow default settings for the .

Table 8: Default Flexible NetFlow Settings

Setting	Default
Flow active timeout	1800 seconds
Flow timeout inactive	15 seconds

How to Configure Flexible Netflow

To configure Flexible Netflow, follow these general steps:

1. Create a flow record by specifying keys and non-key fields to the flow.
2. Create an optional flow exporter by specifying the protocol and transport destination port, destination, and other parameters.
3. Create a flow monitor based on the flow record and flow exporter.
4. Create an optional sampler.
5. Apply the flow monitor to a Layer 2 port, Layer 3 port, or VLAN.

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7.
8. Repeat the above step as required to configure additional nonkey fields for the record.
9. **end**
10. **show flow record** *record-name*
11. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—
Step 7		Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.

	Command or Action	Purpose
Step 8	Repeat the above step as required to configure additional nonkey fields for the record.	—

Step 9	**end**
	Example:
	`Device(config-flow-record)# end`
		Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.

Step 10	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 11	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Flow Exporter

You can create a flow export to define the export parameters for a flow.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

You can export to a destination using IPv4 address.

SUMMARY STEPS

1. **configure terminal**
2. **flow exporter** *name*
3. **description** *string*
4. **destination** {*ipv4-address*}
5. **dscp** *value*
6. **transport udp** *number*
7. **ttl** *seconds*
8. **export-protocol** {**netflow-v9**}
9. **end**
10. **show flow exporter** [**name** *record-name*]
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 2	**flow exporter** *name* **Example:** `(config)# flow exporter ExportTest`	Creates a flow exporter and enters flow exporter configuration mode.
Step 3	**description** *string* **Example:** `(config-flow-exporter)# description ExportV9`	(Optional) Describes this flow record as a maximum 63-character string.

Step 4	**destination** {*ipv4-address*}
	Example:
	`(config-flow-exporter)# destination 192.0.2.1`
	`(IPv4 destination)`	

Step 5	**dscp** *value* **Example:** `(config-flow-exporter)# dscp 0`	(Optional) Specifies the differentiated services codepoint value. The range is from 0 to 63. The default is 0.
Step 6	**transport udp** *number* **Example:** `(config-flow-exporter)# transport udp 200`	(Optional) Specifies the UDP port to use to reach the NetFlow collector.
Step 7	**ttl** *seconds* **Example:** `(config-flow-exporter)# ttl 210`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. The range is from 1 to 255 seconds. The default is 255.

Step 8	**export-protocol** {**netflow-v9**}
	Example:
	`(config-flow-exporter)# export-protocol netflow-v9`
		Specifies the version of the NetFlow export protocol used by the exporter.

	Command or Action	Purpose

Step 9	**end**
	Example:
	`(config-flow-record)# end`
		Returns to privileged EXEC mode.

Step 10	**show flow exporter** [**name** *record-name*] **Example:** `# show flow exporter ExportTest`	(Optional) Displays information about NetFlow flow exporters.
Step 11	**copy running-config startup-config** **Example:** `# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

Define a flow monitor based on the flow record and flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*

5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }}
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.

Step 5	**record** {*record-name*}
	Example:
	`(config-flow-monitor)# record FLOW-RECORD-1`
		Specifies the record for the flow monitor.

Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** }} **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.

	Command or Action	Purpose

Step 9	**end**
	Example:
	`(config-flow-monitor)# end`
		Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:** `# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status for a Flexible NetFlow flow monitor.
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow to an Interface

You can apply a flow monitor and an optional sampler to an interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type*
3. {**ip flow monitor** | **ipv6 flow monitor**}*name* [**sampler** *name*] {**input**}
4. **end**
5. **show flow interface** [*interface-type number*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type* **Example:** `(config)# interface GigabitEthernet1/0/1`	Enters interface configuration mode and configures an interface.
Step 3	{**ip flow monitor** | **ipv6 flow monitor**}*name* [**sampler** *name*] {**input**}	Associate an IPv4 or an IPv6 flow monitor, and an optional sampler to the interface for input or output packets.

	Command or Action	Purpose
	Example: `(config-if)# ip flow monitor MonitorTest input`	You can associate multiple monitors to an interface in both input and output directions.

Step 4	**end**
	Example:
	`(config-flow-monitor)#  end`
		Returns to privileged EXEC mode.

Step 5	**show flow interface** [*interface-type number*] **Example:** `# show flow interface`	(Optional) Displays information about NetFlow on an interface.
Step 6	**copy running-config startup-config** **Example:** `# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Layer 2 NetFlow

You can define Layer 2 keys in Flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.

SUMMARY STEPS

1. **configure terminal**
2. **flow record** *name*
3. **match datalink** { **ethertype** | **mac**}
4. **end**
5. **show flow record** [*name*]
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 2	**flow record** *name*	Enters flow record configuration mode.

	Command or Action	Purpose

	Example:
	`(config)# flow record L2_record`
	`(config-flow-record)#`	

Step 3	**match datalink** { **ethertype** | **mac**}
	Example:
	`(config-flow-record)# match datalink ethertype`
		Specifies the Layer 2 attribute as a key.

Step 4	**end**
	Example:
	`(config-flow-record)#  end`
		Returns to privileged EXEC mode.

Step 5	**show flow record** [*name*] **Example:** `# show flow record`	(Optional) Displays information about NetFlow on an interface.

Step 6	**copy running-config startup-config**
	Example:
	`# copy running-config`
	`startup-config`
		(Optional) Saves your entries in the configuration file.

Monitoring Flexible NetFlow

The commands in the following table can be used to monitor Flexible NetFlow.

Table 9: Flexible NetFlow Monitoring Commands

Command	Purpose
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	Displays information about NetFlow flow exporters and statistics.
show flow exporter [**name** *exporter-name*]	Displays information about NetFlow flow exporters and statistics.
show flow interface	Displays information about NetFlow interfaces.
	Displays information about NetFlow flow monitors and statistics.
show flow monitor statistics	Displays the statistics for the flow monitor

Command	Purpose
	Displays the contents of the cache for the flow monitor, in the format specified.
show flow record [**name** *record-name*]	Displays information about NetFlow flow records.
show sampler [**broker** | **name** | *name*]	Displays information about NetFlow samplers.

Configuration Examples for

Example: Configuring a Flow

This example shows how to create a flow and apply it to an interface:

```
# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

(config)# flow export export1
(config-flow-exporter)# destination 10.0.101.254
(config-flow-exporter)# transport udp 2055
(config-flow-exporter)# exit
(config)# flow record record1
(config-flow-record)# match ipv4 source address
(config-flow-record)# match ipv4 destination address
(config-flow-record)# match ipv4 protocol
(config-flow-record)# match transport source-port
(config-flow-record)# match transport destination-port

(config-flow-record)# collect interface {output}
(config-flow-record)# collect counter bytes
(config-flow-record)# collect counter packets
(config-flow-record)#
(config-flow-record)#
(config-flow-record)# exit
(config)# flow monitor monitor1
(config-flow-monitor)# record record1
(config-flow-monitor)# exporter export1
(config-flow-monitor)# exit
(config)# interface tenGigabitEthernet 1/0/1
(config-if)# ip flow monitor monitor1 input
(config-if)# end
```

Example: Monitoring IPv4 ingress traffic

This example shows how to monitor IPv4 ingress traffic (int g1/0/11 sends traffic to int g1/0/36 and int g3/0/11).

```
# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
(config)# flow record  fr-1
(config-flow-record)# match ipv4 source address
(config-flow-record)# match ipv4 destination address
(config-flow-record)# collect interface {output}
(config-flow-record)# collect counter bytes
(config-flow-record)# collect counter packets
(config-flow-record)#
(config-flow-record)#
(config-flow-record)# exit

(config)# flow exporter fe-ipfix6
(config-flow-exporter)# destination 2001:0:0:24::10
(config-flow-exporter)# source Vlan106
(config-flow-exporter)# transport udp 4739
(config-flow-exporter)# export-protocol ipfix
(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow exporter fe-ipfix
(config-flow-exporter)# description IPFIX format collector 100.0.0.80
(config-flow-exporter)# destination 100.0.0.80
(config-flow-exporter)# dscp 30
(config-flow-exporter)# ttl 210
(config-flow-exporter)# transport udp 4739
(config-flow-exporter)# export-protocol ipfix
(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow exporter fe-1
(config-flow-exporter)# destination 10.5.120.16
(config-flow-exporter)# source Vlan105
(config-flow-exporter)# dscp 32
(config-flow-exporter)# ttl 200
(config-flow-exporter)# transport udp 2055

(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow monitor fm-1
(config-flow-monitor)# exporter fe-ipfix6
(config-flow-monitor)# exporter fe-ipfix
(config-flow-monitor)# exporter fe-1
(config-flow-monitor)# cache timeout inactive 60
(config-flow-monitor)#  cache timeout active 180
(config-flow-monitor)# record fr-1
(config-flow-monitor)# end

# show running-config interface g1/0/11
# show running-config interface g1/0/36
# show running-config interface g3/0/11
# show flow monitor fm-1 cache format table
```

Example: Monitoring IPv4 egress traffic

```
# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
(config)# flow record  fr-1 out
(config-flow-record)# match ipv4 source address
```

```
(config-flow-record)# match ipv4 destination address
(config-flow-record)# match interface output
(config-flow-record)# collect interface {output}
(config-flow-record)# collect counter bytes
(config-flow-record)# collect counter packets
(config-flow-record)#
(config-flow-record)#
(config-flow-record)# exit

(config)# flow exporter fe-1
(config-flow-exporter)# destination 10.5.120.16
(config-flow-exporter)# source Vlan105
(config-flow-exporter)# dscp 32
(config-flow-exporter)# ttl 200
(config-flow-exporter)# transport udp 2055
(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow exporter fe-ipfix6
(config-flow-exporter)# destination 2001:0:0:24::10
(config-flow-exporter)# source Vlan106
(config-flow-exporter)# transport udp 4739
(config-flow-exporter)# export-protocol ipfix
(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow exporter fe-ipfix
(config-flow-exporter)# description IPFIX format collector 100.0.0.80
(config-flow-exporter)# destination 100.0.0.80
(config-flow-exporter)# dscp 30
(config-flow-exporter)# ttl 210
(config-flow-exporter)# transport udp 4739
(config-flow-exporter)# export-protocol ipfix
(config-flow-exporter)# template data timeout 240
(config-flow-exporter)# exit

(config)# flow monitor fm-1-output
(config-flow-monitor)# exporter fe-1
(config-flow-monitor)# exporter fe-ipfix6
(config-flow-monitor)# exporter fe-ipfix
(config-flow-monitor)# cache timeout inactive 50
(config-flow-monitor)#  cache timeout active 120
(config-flow-monitor)# record fr-1-out
(config-flow-monitor)# end

# show flow monitor fm-1-output cache format table
```

INDEX