



Installing Cisco IOS Software TSN Support on a Cisco IE 4000 Switch

First Published: 2017-03-09

Last Modified: 2023-02-15

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



Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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



CHAPTER 1

About Time-Sensitive Networking

- [About Time Sensitive Networking, on page 1](#)
- [A Shared Concept of Time , on page 1](#)
- [TSN System Components , on page 2](#)
- [Overview of Cisco TSN Solution , on page 2](#)
- [CNC to TSN Bridge Control Plane, on page 4](#)
- [Ingress of TSN Flows, on page 7](#)
- [Egress of TSN Flows , on page 7](#)

About Time Sensitive Networking

What is time-sensitive networking (TSN)? In its simplest form, TSN is the IEEE 802.1Q defined standard technology to provide deterministic messaging on standard Ethernet. TSN technology is centrally managed and delivers guarantees of delivery and minimized jitter using time scheduling for those real-time applications that require determinism.

TSN is a Layer 2 technology. The IEEE 802.1Q standards work at OSI Layer 2. TSN is an Ethernet standard, not an Internet Protocol standard. The forwarding decisions made by the TSN bridges use the Ethernet header contents, not the IP address. The payloads of the Ethernet frames can be anything and are not limited to Internet Protocol. This means that TSN can be used in any environment using standard Ethernet and can carry the payload of any industrial application or protocol.

There's an implied requirement for those networking devices implementing TSN (end devices and bridges) to share a common sense of time. Precision Time Protocol (PTP) is used to maintain a common sense of time. The PTP profiles chosen to work with TSN are IEEE 802.1AS and IEEE 802.1ASRev.

A Shared Concept of Time

The key to providing determinism is a shared concept of time. The implementation of 802.1AS (and in the future 802.1ASRev) by all network elements (end devices and bridges) is required for TSN. The 802.1AS PTP profile allows all TSN network elements to share the same concept of time.

802.1Qbv is essential to providing on-time delivery of TSN frames. Qbv defines a means to transmit certain TSN Ethernet frames on a schedule, while allowing non-TSN Ethernet frames to be transmitted on a best effort basis around the TSN frames. Because all network elements share the same time, end devices and bridges implementing Qbv can deliver critical communication very quickly and with no discernible jitter in delivery.

IEEE 802.1Qcc is focused on definition of management interfaces and protocols to enable TSN network administration. 802.1Qcc is a large specification with many aspects. Cisco is using the centralized approach to network management defined in 802.1Qcc.

TSN System Components

There are five main components in the TSN system:

- **TSN flow:** Term used to describe the time-critical communication between end devices. Each flow has strict time requirements that the networking devices honor. Each TSN flow is uniquely identified by the network devices.
- **End Devices:** application that requires deterministic communication. These are also referred to as talkers and listeners.
- **Bridges:** Also referred as Ethernet switches. For TSN, these are special bridges capable of transmitting the Ethernet frames of a TSN flow on a schedule and receiving Ethernet frames of a TSN flow according to a schedule.
- **Central network controller (CNC):** For TSN, the CNC acts as a proxy network (the TSN bridges and their interconnections) to the applications that require deterministic communication. The CNC defines the schedule on which all TSN frames are transmitted. The CNC application is provided by the vendor of the TSN bridges. Cisco has developed a CNC application for controlling its TSN bridges for TSN.
- **Centralized user configuration (CUC):** An application that communicates with the CNC and the end devices. The CUC represents the control applications and the end devices. The CUC makes requests to the CNC for deterministic communication (TSN flows) with specific requirements for those flows. The CUC is an application that is vendor specific. Typically the vendor of the TSN end devices will supply a CUC for those end devices.

Overview of Cisco TSN Solution

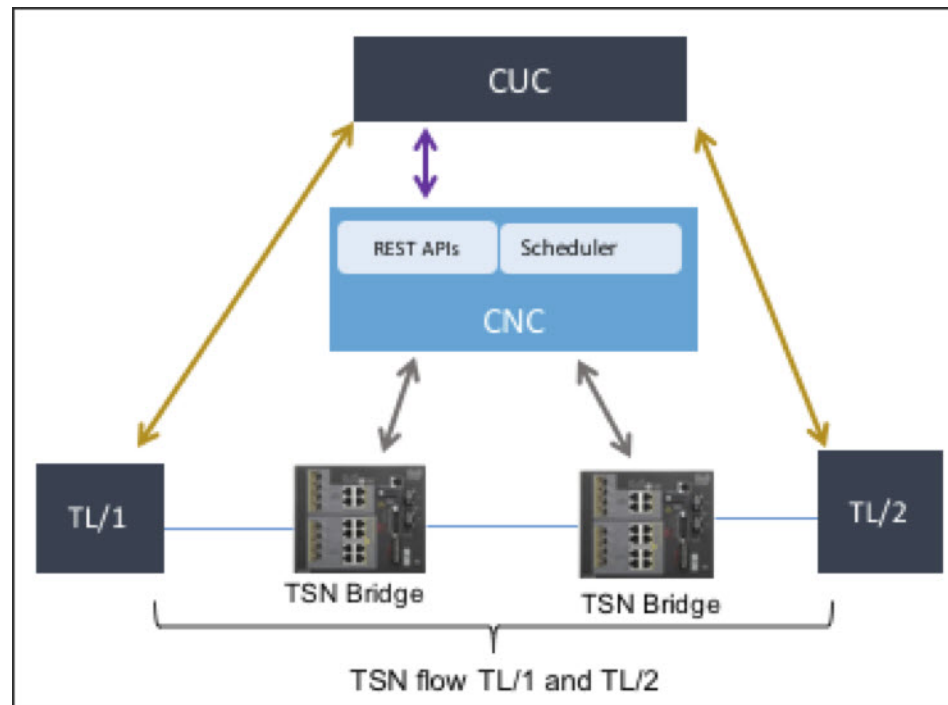
Cisco is supporting TSN on the IE-4000 product family. All models of IE-4000 support TSN. Starting with Cisco IOS® Software release 15.2(5)E2, the TSN functionality is available. The IE-4000s have an FPGA in the data path that enable them to support TSN. The IE-4000 operates as a TSN bridge. The IE-4000 implements IEEE 802.1Qbv and IEEE 802.1AS. The IE-4000 supports hundreds of TSN flows across its Ethernet interfaces.

As the name suggests, the TSN central network controller (CNC) controls the TSN bridges in the network. The CNC is a software application running on customer premises (as opposed to cloud). The hardware housing the CNC application is not relevant and can be anything. The CNC has two primary responsibilities. First, it is responsible for determining routes and scheduling the TSN flows through the bridged network. Second, it is responsible for configuring the TSN bridges for TSN operation.

The CNC communicates with the CUC to receive the communications requirements that the network must provide. The CNC aggregates all the requests, figures out the optimal path for each communication request, schedules the end-to-end transmission for each TSN flow, and finally transfers the computed schedule to each TSN bridge. As part of the schedule computation, the CNC provides a unique identifier for each TSN flow. This unique identifier is used by the TSN bridges to differentiate one TSN flow from another. The unique identifier includes the destination MAC address, VLAN ID, and CoS value. With these three items, the TSN bridges can identify the TSN flow and transmit the flow based on the correct schedule.

In Figure 1, you can see all the components and how they relate. The CUC communicates with the CNC using REST APIs. The CNC communicates with both CUC and TSN bridges. The CUC communicates with the talker/listener end devices. The TSN bridges switch the Ethernet frames of TSN flows between the talkers and listeners.

Figure 1: Sample Topology for TSN Networking Components



The APIs between the CUC and CNC are defined and available as part of the Cisco TSN documentation. The APIs are there for everything needed by the CUC to completely program the TSN flows in the network.

Figure 1 shows lines of control plane communication of messages that occur between the CUC and CNC before the talkers and listeners can start exchanging TSN communications (eg: TSN flows).

The following steps are typical for a workflow and are described only for the purpose of showing how and why the entities communicate with each other. Disclaimer: This is not the only order in which these steps can be executed.

1. CUC Initiates Physical Topology Discovery

Before the schedule can be successfully computed, the CNC must learn the physical topology. The engineer (via the CUC) will initiate a request to the CNC to discover the physical topology. Using LLDP and a seed device, the CNC walks the physical topology, discovering each device and how they are connected. This includes the end devices that support LLDP.

After completion, the engineer (via the CUC) issues a request of the CNC to return the discovered topology. The engineer at this point could verify that the CNC discovered the topology correctly if they choose.

2. CUC Requests Network Resources

Performed by the engineer responsible for defining the end to end communication. The engineer works out which end device (talker) has to communicate with other end devices (listener). The engineer is responsible for identifying all listeners, because there can be more than one for each TSN flow. The engineer can also define the latency requirements for the communication (for example, the listeners must

receive within 500µs from start of transmission), the maximum size of the Ethernet packet that will be sent, and other dependencies (for example, whether there is a sequence order to the TSN flows). The CUC will gather this for all TSN flow requests and submit to the CNC using an API for accepting requests.

3. Compute Schedule

Satisfied that the topology has been discovered and the CNC has received all TSN flow requests, the engineer (via the CUC) will initiate a request that the CNC compute the schedule. The CNC will return success or failure for the request. The end schedule cannot be computed unless the CNC knows the physical topology. Step 3 depends on steps 1 and 2.

4. View Computation Results

Typically the network engineer would want to view the schedule and verify before making it go live. The engineer (via the CUC) requests the CNC return the details of the computed schedule. This includes the details for each device involved in the TSN flows. The details include everything the end devices and bridges need to know for configuring TSN:

- Unique identifiers for each TSN flow (destination MAC address, VLAN, CoS)
- Start and end of transmit window at each hop (talkers and bridges)
- Start and end of receive window at each hop (listeners and bridges)
- End-to-end latency as computed



Note Steps 1 through 4 can be repeated as often as desired.

5. Distribute Schedule

Satisfied that the schedule will work, the engineer (via the CUC) issues a request to the CNC to distribute the computed schedule to the TSN bridges. The CUC will also program the talkers and listeners for the TSN flows. The talkers are expected to transmit every TSN flow according to a schedule.

6. Verify Schedule Distribution on TSN Bridges

This step is optional. In troubleshooting scenarios, users can log in to the TSN bridges to verify the schedule for the TSN flows.

CNC to TSN Bridge Control Plane

The CNC and TSN Bridge communicate on two occasions. When the CNC discovers the physical network, and when the CNC distributes the computed schedules. The CNC and TSN bridges are connected with a Layer 3 IP connection. They do not have to be in the same subnet, even though that often happens.

1. Network Discovery

Before the CNC can distribute, it must do a discovery. The CNC must know where to distribute, and how to compute schedules. Can't compute schedules accurately until the CNC knows exactly how all the Talkers and Listeners are connected. The CNC discovers the network doing a crawl of the LLDP tables on the TSN Bridges.

The CNC requires a ‘seed’ device to start the discovery. To start with the ‘seed’ the administrator must provide a device Hostname for Bridge and an IP Address. From there it will visit all TSN bridges and end devices it can find. For network discovery to be successful, each TSN Bridge must have a unique host name. This means the administrator must change the hostname from ‘Switch’ to something unique within the network. The end devices should have unique names too.

Use the IOS CLI command ‘show lldp neighbor detail’ on the TSN Bridge to view neighbors and their details. The output of this command is the same info the CNC uses.

The CNC will log into the TSN Bridges using Telnet. For this to succeed, the login password needs to be configured and the CNC must know what it is. It needs to be the same for all TSN Bridges. By default the CNC uses ‘admin’ as the login password. Unless the login password is changed on the CNC, the TSN Bridges must have ‘admin’ configured as well.

2. Schedule Distribution

After CNC has successfully computed a schedule, the schedule is not active until it is distributed to the TSN Bridges. The CNC and TSN Bridges have a layer 3 connection to allow the TSN Bridges to receive the schedule. Each TSN bridge has its own connection to the CNC and each Bridge receives its own schedule. On each TSN Bridge, the IP address of the CNC must be configured.

Every time the CNC successfully computes a schedule, it generates an ID for that schedule. The ID is a 4 byte hex value that is unique for each schedule. When the distribution is successful, the TSN Bridge will show the schedule ID that is in operation. It should match the schedule ID on the CNC.

For schedule distribution to work, the TSN Bridge must know how to contact the CNC to get the schedule. This requires a 1 line configuration command on the TSN Bridge: **TSN_Bridge (config)# tsn cnc-server <IP Address of CNC>**

To confirm the CNC is communicating with the TSN Bridge use commands ‘show tsn cnc’, and ‘show tsn schedule’.

```
TSN_Bridge # show tsn schedule
Device ID: TSN_Bridge
Schedule ID: 1492128178 - 0x58F011B2
Status      : TSN schedule has been programmed successfully
```

```
TSN_Bridge # show tsn cnc
CNC Connection Status
-----
Device name:          TSN_Bridge
CNC server:           2.1.28.16:4569
Last CNC Contact:     17:22:50  Fri May 5 2017
Last Good CNC Connection: 17:22:52  Fri May 5 2017
Current Schedule ID:  1492128178 - 0x58F011B2
```

The data plane on the IE4000 is concerned only with the ingress and egress of TSN flows. Once the CNC distributes schedules to the Bridges, and the Talker and Listeners end devices have also been given their TSN flows schedules, the focus shifts to the Networking devices and not the CNC.

The CNC is in the control plane only. If the CNC is powered off or disconnected, it will not impact the operation of the TSN flows.

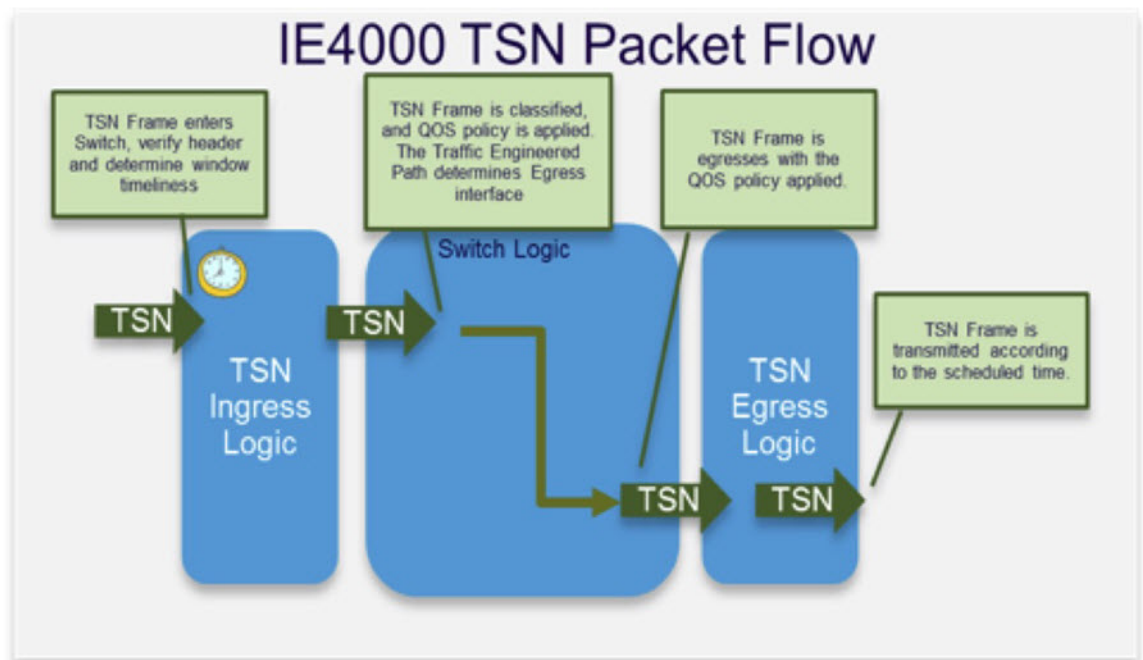
Each TSN flow is traffic engineered through the bridged Ethernet network. The CNC determined the forwarding path as part of the computation of the schedule. The TSN bridges forward TSN flows according to how the CNC instructed. TSN bridges do not use dynamic learning of MAC address for TSN flows.

The Vlan(s) used for TSN flows do not run Spanning Tree. It is mandatory that spanning tree be disabled on all vlans containing TSN flows. If the TSN Bridge receives an Ethernet packet that does not belong to

an expected TSN flow on that interface for that vlan, the bridge will drop the packet. TSN vlans should not have Broadcast packets. All TSN flows have vlan tags in the Ethernet header. All interfaces on the TSN bridges that are expected to receive or transmit TSN flows must be in trunk mode. On the IE4000, interfaces expecting TSN flows are automatically placed into trunk mode. the TSN vlan(s) cannot be the 'access' vlan on the Ethernet interfaces.

The Cisco TSN Bridges use service policies on ingress and egress to ensure the TSN flows receive proper quality of service within the internal switching infrastructure. The TSN flows use the high priority Queue to prevent them from being delayed during egress congestion. For all interfaces that support a TSN flow, an ingress and an egress service policy will be automatically applied to the interface for QOS purposes.

The figure below shows the HW architecture for the Cisco TSN Bridge. This will aid in troubleshooting and monitoring TSN flows.



To view the TSN flows that have scheduled for a TSN Bridge, there are two CLI commands for summary and detailed view.

```
TSN_Volt1# show tsn flow summary
Flow-ID Stream-ID          DMAC          Status Ingress  Egress
-----
1001   flow1                0300.5EA0.03E9 up    Gi1/5    Gi1/4
1002   flow2                0300.5EA0.03EA up    Gi1/4    Gi1/5
```

```
TSN_Bridge# show tsn flow details
Flow 1001
  Stream ID : flow1
  Stream Address : 0300.5EA0.03E9
  Frame Size : 64B
  Ingress Interface : Rx Schedule
    Gi1/5: 13-26 (us)
  Egress Interface : TX Schedule
    Gi1/4: 207-220 (us)
  Period cycle time : 1000 (us)
Flow 1002
```



```

Stream ID : flow2
Stream Address : 0300.5EA0.03EA
Frame Size : 64B
Ingress Interface : Rx Schedule
  Gig1/4: 245-258 (us)
Egress Interface : TX Schedule
  Gig1/5: 283-295 (us)
Period cycle time : 1000 (us)

```

Ingress of TSN Flows

All flows that ingress the TSN bridge have an expected window of time when they are expected. This window is approximately 13us for 1000Mbps links and 130us for 100Mbps links. TSN is not supported on 10Mbps links. The Cisco TSN bridge is able to determine if a given flow is ingressing on time (eg: in the scheduled window). For each TSN flow, a count is maintained for ingressing early, on time, or late. This is useful in case there are any issues with the end to end TSN flow, and it needs to be root caused. For TSN flows not ingressing on time, one possible reason is the transmitter not being in clock synchronization with the network.

Use command “**show interface <interface id> tsn ingress stats <flow id>**” to see ingress statistics for each tsn flow. The interface and the tsn flow id are required.

Example:

```

TSN_Switch # show interface Gig1/4 tsn ingress stats 1002
                Latest          Accumulated
                -----          -
Unexpected      0                0
In              254476           267372
Early           0                0
Late            0                9003

```

‘In’ counts packets in the flow received within the expected window. The ‘Latest’ column clears on read and shows counts since the last read. ‘Accumulated’ column maintains history and doesn’t clear unless an explicit action to clear counters is made.

Egress of TSN Flows

Similar to ingress, all TSN flows that egress have an expected window of time when they are expected to egress the TSN Bridge. It doesn’t matter if this interface is bridge to bridge or bridge to the Listener end device, the Cisco TSN Bridge will not transmit the frames of a TSN flow until it is time to do so. This means the TSN bridge has to hold on and delay the TSN frame.

The egress window can be seen using ‘**show tsn flow details**’ CLI command.

The easiest way to confirm that a TSN flow is working correctly on egress is to look at egress counts for each flow. Use this command to see egress counts per flow: **show interface <intf id> tsn flow-stats <flow id>**.



CHAPTER 2

Installing TSN on an IE 4000 Switch

- [Installing TSN on an IE 4000 Switch, on page 9](#)

Installing TSN on an IE 4000 Switch

This section will guide you through installing the TSN capable software package on an IE 4000 switch.

- Step 1** Download the TSN Enabled IOS package from Cisco.com.
For more information about IOS installation and detailed configuration procedures, refer to the [Cisco Industrial Ethernet 4000, 4010 and 5000 Switch Software Configuration Guide](#).
- Step 2** Connect a console port to the IE 4000 switch and make sure it is currently running Cisco IOS.
- Step 3** Proceed with one of the following procedures depending on if you are installing a new switch, upgrading a previously configured TSN switch, or configuring a TSN switch.
- You may install the image via FTP, TFTP, or SDFlash. Follow the steps outlined in the [Working with the Flash File System](#).
- On a NEW Switch:
- Let the switch boot up Completely!
 - Let the switch boot up. It will boot up with the default image inside it.
 - Now go to the “Upgrading the switch image:” section to install TSN image on the switch.
- Upgrading the new switch or old TSN-enabled switch with new image:
- Copy the new tar IOS image into the switch via FTP, TFTP or SDFlash.
- Example:**
- ```
archive download-sw sdfsflash:<image name> flash:
```
- This process will take around 15-20 mins to complete.
- Once completed, issue the following commands.:
- Example:**
- ```
configure terminal
boot system flash:<image name>/<image name>.bin exit
write
reload
```
- After the switch is up with the new image, you need to activate the license for TSN. Follow the procedure below.

Example:

```
enable
configure terminal
service internal
exit
license right-to-use activate tsn
write
reload
```

- d) Once the switch is up it will be TSN enabled and you should be able to use TSN functionality on it. To cross check in the enable mode issue following command:

Example:

```
sh tsn ?
Switch#sh tsn ?
      be_stats      BE traffic statistics
      cnc            Current CNC server connection status
      flow          TSN flows
      fpga          FPGA file information
      license-on-SD Information for tsn license in sdflash
      schedule      schedule
      shadow        Current shadow information
Switch#
```



CHAPTER 3

Configuring TSN

- [Configuring TSN, on page 11](#)

Configuring TSN

On the IE 4000, there is very little configuration required on the CLI. The Time sensitive flows are configured by the CNC. The IOS CLI configuration is to enable configuration with the CNC. Here are the few commands.

Step 1 After the switch is up with the new image, you need to activate the license for TSN. Follow the procedure below.

Example:

```
enable
configure terminal
  service internal
  exit
  license right-to-use activate tsn
  write
reload
```

Step 2 Once the switch is up it will be TSN enabled and you should be able to use TSN functionality on it. To cross check in the enable mode issue following command:

Example:

```
sh tsn ?
Switch#sh tsn ?
      be_stats      BE traffic statistics
      cnc            Current CNC server connection status
      flow          TSN flows
      fpga          FPGA file information
      license-on-SD Information for tsn license in sdflash
      schedule      schedule
      shadow        Current shadow information
Switch#
```

Step 3 enable secret admin

Example:

```
TSN_Switch# configure term
TSN_Switch(config)# enable secret admin
```

CNC uses telnet to login into the IE4000 during discovery

Note Configure the enable password to 'admin' for the CNC. This can be changed, but for ease of use, use 'admin' for now.

Step 4 ptp profile dot1as

Example:

```
TSN_Switch# configure term
TSN_Switch(config)# ptp profile dot1as
IE4000 configured for PTP profile 802.1AS
```

Step 5 tsn cnc-server *ip address*

Example:

```
TSN_Switch# configure term
TSN_Switch(config)#tsn cnc-server <ip address>
IP Address of CNC server configured.
```

Step 6 line vty 0 15

Example:

```
TSN_Switch# configure term
TSN_Switch(config)#line vty 0 15
TSN_Switch(config-line)# password admin
TSN_Switch(config-line)# login
```

CNC logged in via telnet with pasword "admin."

Note You can change the pasword in the CNC later, but for now, use the default, "admin."



CHAPTER 4

Displaying Global TSN Information

- [Displaying Global TSN Information, on page 13](#)

Displaying Global TSN Information

These commands are used to verify the IE 4000 is communicating with the CNC and to see the scheduled TSN flows on the IE 4000.

Step 1 show tsn cnc

Example:

```
TSN_Switch # show tsn cnc
CNC Connection Status
-----
Device name:          TSN_Switch
CNC server:          2.1.28.16:4569
Last CNC Connection: 16:40:50 Thu Mar 2 2017
-----
```

This command shows the connectivity with the CNC. the CNC to switch connection is a Layer 3 connection and is not guaranteed. if you suspect that the switch and CNC have lost communication use this command to see latest connection status.

Step 2 show tsn schedule

Example:

```
TSN_Switch # show tsn schedule
Device ID: TSN_Switch
Schedule ID: 1488646751 - 0x58BAF25F
Status      : TSN schedule has been programmed successfully
```

Use this command to show the schedule ID in use by the IE 4000. this should match the schedule ID on the CNC. If there's a mismatch, then either the latest CNC schedule has not been distributed, or the connection between the switch and CNC have been interrupted.

Step 3 show tsn flow ?

Example:

```
TSN_Switch # show tsn flow ?
details  Provides detailed description of each TSN flow
egress   TSN flows associated with all Egress Interfaces
id       description of the flow identified flow id
```

```

  ingress  TSN flows associated with all Ingress Interfaces
  summary  Provides brief description of each TSN flow

```

Use these commands to see the configured flows on the switch.

Step 4 show tsn flow summary

Example:

```

TSN_Switch # show tsn flow summary
Flow-ID Stream-ID          DMAC          Status Ingress  Egress
-----
1001    flow1                ABAD.BABE.03E9 down   Gi1/5    Gi1/4
1002    flow2                ABAD.BABE.03EA down   Gi1/4    Gi1/5

```

Step 5 show tsn flow details

Example:

```

TSN_Switch# show tsn flow details
Flow 1001
  Stream ID : flow1
  Stream Address : ABAD.BABE.03E9
  Frame Size : 256B
  Ingress Interface : Rx Schedule
    Gi1/5: 12-26 (us)

  Egress Interface : TX Schedule
    Gi1/4: 203-217 (us)

  Period cycle time : 500 (us)
Flow 1002
  Stream ID : flow2
  Stream Address : ABAD.BABE.03EA
  Frame Size : 64B
  Ingress Interface : Rx Schedule
    Gi1/4: 242-255 (us)

  Egress Interface : TX Schedule
    Gi1/5: 280-293 (us)

  Period cycle time : 500 (us)

```



CHAPTER 5

Displaying TSN Statistics

- [Displaying TSN Statistics, on page 15](#)

Displaying TSN Statistics

Ingress Statistics

show interface tsn ingress stats

```
TSN_Switch # show interface Gig1/4 tsn ingress stats 1002
                Latest           Accumulated
                -----           -
Unexpected      0                               0
In-Window       254476                          267372
Early           0                               0
Late            0                               9003
```

Notes on Ingress counters

- Unexpected (Unexpected packets)
 - The number of packets that matched global VLAN and DMAC
 - flow enable bit NOT set or
 - corresponding portmap bit in the flow table NOT set
- In-Window (In Window Bin)
 - The number of packets that arrived within flow's defined window
- Early (Early Bin)
 - The number of packets that arrived before the flow's defined window
- Late (Late Bin)
 - The number of packets that arrived after the flow's defined window



Note Use 'show tsn flow details' for defined ingress window for all TSN flows.
The values under

```
Latest
```

are reset after every read.

Egress Statistics

Use this command to see per flow egress counters. This will show if there are errors on egress because of timing sync errors.

show interface tsn flow stats

```
TSN_Switch# show interface Gig1/5 tsn flow-stats 1001
Flow ID : 1001 (0x3E9)
```

Frame Type	Latest	Accumulated
TSN Queue xmit	0	0
TSN Queue rcv	0	0
TSN xmit timing		0
TSN Unreleased		0

Notes on Egress counters

- TSN Queue xmit (nTransmittedTtFrames)
 - This is a clear-on-read counter
 - This counter is increased
 - if a frame belonging to <flow-id> is a time-triggered (TT) frame AND
 - is transmitted to TXM's output queue
- TSN Queue rcv (nReceivedTtFrames)
 - This is a clear-on-read counter
 - This counter is increased
 - if a frame belonging to <flow-id> is a time-triggered (TT) frame AND...
 - is received correctly, frame size correct, crc correct
- TSN xmit timing (nTimingError)
 - This is NOT a clear-on-read counter
 - This counter is increased when
 - window is closed or...
 - synchronization is not established

- flushing of TT buffer not completed
- TSN Unreleased (nUnreleased)
 - This is NOT a clear-on-read counter
 - This counter is increased when
 - a frame is received for a flow that is dispatched before the most recent frame is dispatched to all ports



Note Ensure that flow ID <flow_id> exists with an egress interface

Use the following variation of the egress statistics command to see per interface egress counters. This shows all bytes and frames that have successfully egressed. This is for TSN and Best Effort Ethernet frames. No flow id required because its for all interfaces.

```
TSN_Switch# show interface Gig1/5 tsn egress counters
Status          Count
-----
Number of Rx frames  0
Number of Rx bytes  0
Number of Tx frames  0
Number of Tx bytes  0
```

Notes on counters

- These are counters maintained by TSN egress logic and apply to all Ethernet frames, not just TSN frames.
- Number of frames received by the Egress logic
- Number of bytes received by the Egress logic
- Number of frames transmitted by the Egress logic
- Number of bytes transmitted by the Egress logic

Use this variation to see per interface egress error counters.**show int tsn egress error**

```
#show int <intf_name><intf_num> tsn egress error

Error          Count
-----
Traffic engineering params mismatch (drop)  0
Broadcast frames not allowed (drop)        0
Size error (drop)                          0
CRC error at TSN Rx (drop)                  0
Flow not found (drop)                       0
Flow policing error (drop)                  296544
Memory unavailable (drop)                   0
```

Notes on counters

- These are NOT clear-on-read registers
- Traffic engineering params mismatch (drop) (nDropNonCiscoErrors)

- Broadcast frames not allowed (drop) (nBroadcastErrors)
- Size error (drop) (nSizeErrors)
- CRC error at TSN Rx (nCrcErrors)
- Flow not found (nFlowNotFoundErrors)
- Flow policing error (nFlowPolicingErrors)
- Memory unavailable (nPartitionDrop)

Use this command to show TSN related egress errors on the interface. **show int tsn egress details**

```
#show int <intf_name><intf_num> tsn egress details

Port status counter          Latest          Accumulated
-----
Runt frames                  0              0
Start of frame delimiter error 0              0
Alignment error              0              0
MII error                    0              0

Port status flag             SET/UNSET
-----
Max age drop flag           UNSET
No space in TSN queue flag  UNSET
Flow is not a TSN flow flag UNSET
TSN time check fail flag    UNSET
Do not allow broadcast frame flag UNSET
```

Notes on counters

- Port status counter shown above are clear-on-read registers
- Port status flag shown above are NOT clear-on-read registers. These are boolean counters that will have either SET or UNSET value.
- Runt frames (nRuntFrames)
- Start of frame delimiter error (nSofErrors)
- Alignment error (nAlignmentErrors)
- MII error (nMiiErrors)

Notes on implementation

- No check whether the port in question is part of a TSN flow
- These error counters are global counters and not related to any particular flow
- These error counters are across all existing TSN flows

Clear statistics counts

clear tsn flow_stats flowsclear tsn stats interface <intf_name><intf_num>

- When this CLI is executed reset all software accumulators for MAC-level diagnostics counters. These accumulators are per port.

Best Effort statistics

show tsn be_stats

```

TSN_switch1#show tsn be_stats
Status          Latest          Accumulated      Latest          Accumulated
              (Port 1-8)      (Port 1-8)      (Port 9-16)    (Port 9-16)
-----
              (Port 17-24)  (Port 17-24)
-----
BE Rx frames    109              150              0               0
0
BE Tx frames    109              150              0               0
0
    
```


THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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 standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2017–2018 Cisco Systems, Inc. All rights reserved.

