

# Time-Sensitive Networking: A Technical Introduction



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 and can carry the payload of any industrial application.

TSN was developed to enable deterministic communication on standard Ethernet. The market for deterministic communication is using nonstandard technologies or nonstandard Ethernet. Prior to the IEEE 802.1 TSN standards, standard Ethernet didn't have pure Layer 2 deterministic capability.

Deterministic communication is important to multiple industries (for example, aerospace, automotive, manufacturing, transportation, and utilities). Providing a means for determinism over standard Ethernet enables new levels of connectivity and optimization, leading to cost savings for many industries.

This short document is intended to give the interested reader more technical details about TSN and an introduction to the Cisco® implementation.

As the name suggests, “time” is the primary aspect of TSN. TSN is a technology focused on time. TSN was developed to provide a way to make sure information can travel from point A to point B in a fixed and predictable amount of time. Being predictable enables increased efficiency.

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.

## TSN Solution Components

There are five main components in the TSN solution:

- **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:** These are the source and destinations of the TSN flows. The end devices are running an 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 for the Network (the TSN Bridges and their interconnections) and the control 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.

Table 1 describes TSN standards.

Table 1. TSN Standards

Standard	Area of Definition	Title of Standard
IEEE 802.1ASrev, IEEE 1588	Timing and synchronization	Enhancements and performance improvements
IEEE 802.1Qbu and IEEE 802.3br	Forwarding and queuing	Frame preemption
IEEE 802.1Qbv	Forwarding and queuing	Enhancements for scheduled traffic
IEEE 802.1Qca	Path control and reservation	Path control and reservation
IEEE 802.1Qcc	Central configuration method	Enhancements and performance improvements
IEEE 802.1Qci	Time-based ingress policing	Per-stream filtering and policing
IEEE 802.1CB	Seamless redundancy	Frame replication and elimination for reliability

Not all standards shown in Table 1 are required to support TSN. Cisco's initial implementation uses 802.1Qbv, 802.1AS (predecessor to 802.1ASRev), and 802.1Qcc.



## 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.

The key to providing on-time delivery of TSN frames is 802.1Qbv. 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.

A good analogy for IEEE 802.1Qbv is a train system. Think of the Ethernet bridges as cities and the links between Bridges as train tracks between the cities.. There is one track between each city. Trains go from one city to another based on a schedule. The train from city A to city X is scheduled well in advance. Although there are many cities and destinations, there are only one track between two adjacent cities (just like the cat5 cabling between two Ethernet bridges). The track is the scarce resource that has to be time shared. Two trains cannot be on the track at the same time. The train system schedules the departure time for each train from a station, making sure that the track is well used and there are no conflicts. With TSN, the Ethernet bridges allow for use of the links to unscheduled traffic when there is no scheduled traffic. For the train system, this could be the regional and local trains using the tracks when the intracity trains are not using them.

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.

## Overview of Cisco TSN Solution

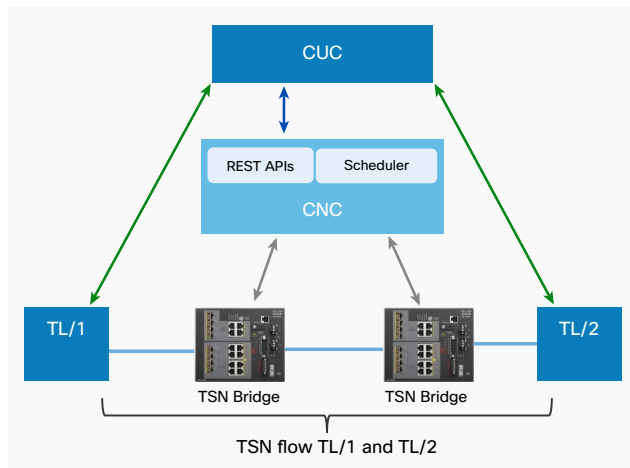
Cisco is supporting TSN on the IE-4000 product family. All models of IE-4000 support TSN. Starting with Cisco IOS® Software release 15.2(5)E2, the TSN functionality is available. The IE-4000s have an FPGA in the data path that enable them to support TSN. The IE-4000 operates as a TSN bridge. The IE-4000 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, computes the route 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 TSN communication Ethernet packets 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. The APIs are so complete, the CUC can hide the CNC and the network. It's possible for the control engineer to interface with the CUC only, never actively using the CNC or configuring a TSN bridge.

### TSN Sample Workflow

Figure 1 shows lines of control plane communication of steps that occur between the CUC and CNC before the talkers and listeners can start exchanging TSN communications (called 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.

#### Step 1: CUC Initiates Physical Topology Discovery

Before the schedule can be successfully computed, the CNC must learn the physical topology. 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 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.

#### Step 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.

#### Step 3: Compute Schedule

Satisfied that the topology has been discovered and the CNC has received all TSN flow requests, 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.

#### Step 4: View Computation Results

Typically the network engineer would want to view the schedule and verify before making it go live. 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

### Step 5: Distribute Schedule

Satisfied that the schedule will work, 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.

### Step 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.

### Actual Example Using Two Flows with Two Endpoints

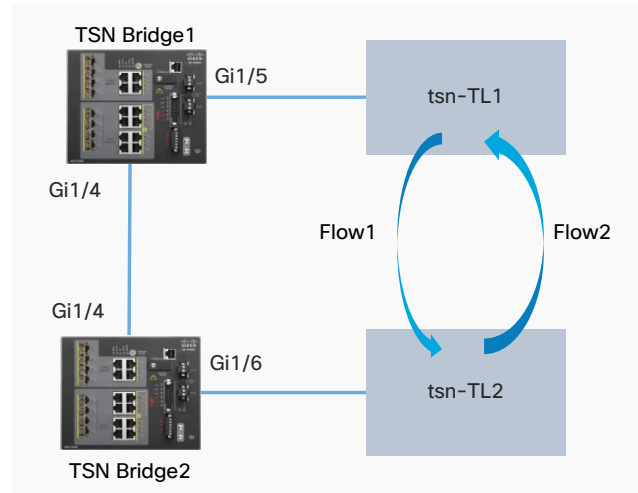
Here's an example of how an engineer might create two TSN communication flows between two end devices.

In this scenario, TL1 and TL2 are TSN end devices that operate in sync with each other. They operate in a very fast control loop that iterates 1000 times a second. Every millisecond (1/1000 of a second), TL1 sends a time-sensitive message to TL2. TL2 receives the message, does computation, and sends a time-sensitive message back to TL1. TL2 needs to receive the time-sensitive message from TL1 before 500 $\mu$ s into the 1ms. TL2 has 250 $\mu$ s to 350 $\mu$ s to do the computation based on the date received from TL1, then transmit a response back to TL1, which must arrive before the 1ms time is up. Each time a message is sent between TL1 and TL2, the network has 100 $\mu$ s to deliver that message.

It's worth noting that the control engineer in this example never has to configure the TSN bridges. The CNC takes care of all TSN bridge communication and configuration. The control engineer only interfaces with the CUC. This makes it easy for the control engineer to implement TSN in the network.

Figure 2 shows an example physical network topology. The network devices and the connections were set up in advance and are static. The TSN flows are between the two end devices (identified as TL1 and TL2).

Figure 2. Physical Topology for Example Workflow



**Step 1.** Using the CUC, the control engineer asks the CNC to discover the physical network in preparation for schedule computation.

**Step 2.** Using the CUC, the control engineer makes two requests to create TSN flows of the network. The first request is to create a TSN flow from TL1 to TL2. TL1 will start transmitting at 350 $\mu$ s after the start of the millisecond. The network has 100 $\mu$ s of maximum latency to deliver the message to TL2. The message has maximum size of 64 bytes. The second request is the response message. The control engineer requests the network to create a TSN flow from TL2 to TL1. TL2 will start transmitting at 850 $\mu$ s after the start of the millisecond. The network has a 100 $\mu$ s maximum latency to deliver the message to TL1. The response message is 64 bytes maximum size.

**Step 3.** Using the CUC, the control engineer asks the CNC to compute a schedule for the requested TSN flows. The CNC returns a response to the compute request based on success or failure of the scheduling logic.

**Step 4.** The control engineer views the computation results. In tabular format, the results of the computation look like Table 2.

Table 2. Sample Computation Results

Stream ID	Destination MAC	Size	Period	Talker Name	CoS	VLAN	Talker Transmit Window	Receiver Name	Receiver Window	Transmission Duration
Flow 1	03:00:5E:A0:03:e9	64	1ms	tsn-TL1	5	3000	350–363µs	tsn-TL2	463–476µs	100µs
Flow 2	03:00:5E:A0:03:eA	64	1ms	tsn-TL2	5	3000	850–863µs	tsn-TL1	963–976µs	100µs

When transmitting a TSN flow, the talker is given a window to transmit. The window is approximately 13µs wide in terms of time (for links operating at Gigabit speeds) The 13µs is required to account for the chance that a large 1518-byte Ethernet frame will be transmitted just before the TSN flow Ethernet frame and will delay the TSN flow by ~12µs. A 64-byte frame takes 0.7µs to transmit.

**Step 5.** Satisfied the schedule computation is meeting the requirements, the control engineer instructs the CUC to distribute the schedule to the networking elements.

**Step 6.** This step is optional. This how the schedule will appear on a Cisco TSN Bridge. This is TSN-Bridge 1.

```
TSN_Bridge1> show tsn flow detail
Flow 1001
  Stream ID : flow1
  Stream Address : 0300.5EA0.03E9
  Frame Size : 64B
  Ingress Interface : Rx Schedule
    Gi1/5: 350-363 (us)
  Egress Interface : TX Schedule
    Gi1/4: 390-403 (us)
  Period cycle time : 1000 (us)
Flow 1002
  Stream ID : flow2
  Stream Address : 0300.5EA0.03EA
  Frame Size : 64B
  Ingress Interface : Rx Schedule
    Gi1/4: 923-936 (us)
  Egress Interface : TX Schedule
    Gi1/5: 963-976 (us)
  Period cycle time : 1000 (us)
```

The transmit of flow 1 from tsn-TL1 is the receive for TSN-Bridge1. and the transmit of flow 2 from TSN-bridge1 is the receive for tsn-TL1. There is some delay on the TSN bridge that is added by the scheduler. In this case, the delay is at 27µs. TSN bridges have store and forward architectures, and this adds to the end-to-end latency of the Ethernet frame.

### Organizations Driving TSN

Just implementing to the IEEE standards is not enough to make sure of vendor interoperability. For instance, different vendors could implement the standard differently. Each vendor could claim support for the standard. It would be up to the end customer deploying products that claim to support the standard to solve any incompatibilities between the vendors. Neither the customer nor the vendors want such a situation.

To help with interoperability, organizations are formed to focus on solutions that are based on new standards. For TSN, two of the most active organizations are AVnu Alliance and the Industrial Internet Consortium. Both organizations are composed of vendors that are building TSN products.

### AVnu

AVnu Alliance is a community creating an interoperable ecosystem servicing the precise timing and low-latency requirements of diverse applications using open standards through certification.

AVnu writes standards, which are profiles of IEEE standards that help implementers in selected vertical industries make sense of large, general-purpose standards documents.

AVnu tests and certifies devices for interoperability, providing a simple and reliable networking solution for AV network implementation based on the IEEE 802.1 Audio Video Bridging (AVB) and Time-Sensitive Networking (TSN) standards.

AVnu provides marketing support for the concepts, standards, and products supporting AVB and TSN standards.

### Industrial Internet Consortium (IIC)

The IIC brings together organizations and technologies necessary to accelerate the growth of industrial Internet by identifying, assembling, and promoting best practices. The goals of the IIC are to drive innovation, define reference architectures, facilitate open forums and test beds, and influence the global development standards process for the Internet and industrial systems.

### Q&A

- Q** What value does TSN bring to customers?
- A** Customers employing TSN-based deterministic solutions will be able to consolidate multiple services over a single, standard physical network.
- Q** Does the TSN technology from Cisco require a license?
- A** Yes. To enable the TSN technology on Cisco IE switches, a license must be purchased.
- Q** Is TSN a new type of quality of service?
- A** No. Quality of service is used by networking devices (bridges and routers) to prioritize traffic. In congestion conditions, QoS enables bridges and routers to prioritize the different network traffic. TSN flows are high priority, but do not compete with non-TSN flows regardless of the priority of the non-TSN flows.
- Q** Can TSN flows be configured on the Cisco IOS Software command line?
- A** No. TSN flows can only be configured from the CNC. From the Cisco IOS Software CLI, only TSN troubleshooting is permitted.