

# An Approach to CIP Motion on TSN Networks

Paul Brooks  
Rockwell Automation

Yi YU  
Rockwell Automation

Paul Didier  
Cisco Systems

Jordon Woods  
Analog Devices Inc.

Presented at the ODVA  
2020 Industry Conference & 20th Annual Meeting  
March 4, 2020  
Palm Harbor, Florida, USA

## Abstract

There has been significant activity recently in the market around Time Sensitive Networking (TSN) culminating with both CC-Link IE and Profinet proposing new work items in IEC for TSN extensions to their technologies. In preparation for work inside ODVA extending the EtherNet/IP specifications to support TSN, a number of member companies have developed EtherNet/IP TSN test beds. This paper will give a report on the lessons learned from the test bed of CIP Motion over TSN, which has been cooperatively developed by Rockwell Automation, Cisco and Analog Devices Inc.

The test bed consists of TSN-capable network bridges, embedded device prototype gateway boards and CIP Motion capable end nodes (PLC and Drive) with the prototype firmware supporting TSN. The key TSN features supported in this test bed include:

- IEEE 802.1Qbv which specifies a time-aware shaper to schedule traffic. The CIP Motion traffic is inserted into a scheduled part of the network bandwidth.
- IEEE 802.1Qcc which enhances the stream reservation protocol (SRP) and operates at the network control plane
- IEEE 802.1AS which provides peer-to-peer precision time clock synchronization, and is a profile of IEEE 1588 (while CIP Sync uses the default IEEE 1588 profile without peer-to-peer synchronization)

We will present an evaluation on the aspects of the TSN adoption method and the performance of TSN adoption for a CIP system. Based on the evaluation results, we demonstrate the feasibility of integrating EtherNet/IP systems into networks deploying TSN. We also propose both enhancements and requirements for the TSN incorporation into CIP technologies and EtherNet/IP specifications. We believe that these conclusions can help inform ODVA contributions to IEC/IEEE 60802 Industrial Automation Profile for TSN working group.

## Keywords

EtherNet/IP, CIP Motion, TSN (Time Sensitive Networking), CNC (Centralized Network Configurator), TSN Scheduling, Time Gateway, Stream Gateway, TSN Gateway, 802.1Qbv, 802.1Qcc, 802.1AS

## Definition of Terms in This Paper

The terms in this paper are represented in *Italic* style.

*Time Gateway* -- Provides the translation between the default IEEE1588 profile without Peer-to-Peer synchronization (CIP Sync protocol) and the 802.1AS protocol (a profile of IEEE1588 with Peer-to-Peer synchronization)

*Stream Gateway* –Provides the conversion between the standard Layer 2 (L2) raw ethernet flows used in CIP today and the TSN streams. It is compliant with 802.1Qbv Enhancements for Scheduled Traffic, which has already been incorporated into IEEE Std 802.1Q-2018.

*Traffic Splitter* -- Used to split CIP Sync messages from CIP Motion I/O messages.

*Interfering Traffic* - additional traffic at the same QoS priority level as the motion traffic potentially causing CIP Motion application failure. This failure would be due to CIP Motion packets were being dropped by the network infrastructure when queues overflowed

*Interfering Traffic Generator/Receiver* –Used to inject interfering traffic into a CIP Motion control system for evaluating the TSN features in a converged network of multiple types of traffic.

*TSN Switch* – A network bridge which supports the TSN technologies, e.g. peer-to-peer time synchronization protocol per 802.1AS, Enhancements for Scheduled Traffic per 802.1Qbv, Stream Reservation Protocol (SRP) Enhancements and Performance Improvements per 802.1Qcc etc.

*DE-CNC* – A vendor-implemented CNC (Centralized Network Configuration entity) from Cisco.

*CUC* – The abbreviation of Centralized User Configuration entity which is defined in 802.1Qcc. It is responsible for configuring TSN talkers and listeners based on interactions with CNC. There is no *CUC* on this testbed.

*Stream Conversion* –The stream transformation concept as defined in “46.1.4 Stream transformation” of [1]. It implements the function for identifying application flows and then translating them into TSN streams (normally with locally managed or multicast MAC addresses). It utilizes transformation criteria that are compliant with identification rules in 802.1CB standards. For more information, please refer to “6. Stream identification” of [2].

*TSN Scheduling* – The name, for the purposes of this paper, for the implementation practices for scheduling TSN streams according to the enhancement of scheduled traffic in 802.1Qbv standard.

*TSN Configuration* – The name, for the purposes of this paper, for the implementation practices for configuring TSN-related parameters in both bridges and end stations. It involves the *Stream Conversion* configuration and the *TSN Scheduling* configuration based on the enhancement of stream reservation protocol in the 802.1Qcc standard.

*Two-Cycle Model for Isochronous Motion Control* – An isochronous control loop communication model. Taking the prototype CIP Motion control in this paper for instance, the C2D (Controller-to-Drive) and D2C (Drive-to-Controller) communication are to be started respectively at the very beginning of and at the ½ of each application cycle. And either C2D or D2C transmission must be finished within the half of application cycle. For more information, please refer to “2.4.4 Use case 02: Isochronous Control Loops with guaranteed low latency” of [3].

*Reduction Ratio* – Describes the relation between application cycle and network cycle, as defined in “2.4.6 Use case 04: *Reduction ratio* of network cycle Application” of [3]. For simplicity on this testbed, the *Reduction Ratio* is set to “1”.

*Top of Second* – A method for determining the start point of a cycle, e.g. the application control cycle or the TSN network cycle. It aligns the start point of a cycle at the time point that is of “n” seconds, where the “n” is an integer number.

## Related Work

Before this evaluation of a physical EtherNet/IP system over TSN, there had already been discussions related to the TSN adoption for EtherNet/IP technology. For example in [4], the author suggests the “Time Bridge” and “TSN Adapter” for migrating the installed base to a TSN infrastructure. This paper also highlighted some key TSN technologies such as timing, preemption, scheduling, and configuration, which should be studied by ODVA SIGs in the future with actual TSN standard releases. In [5], the author analyzed the theoretical performance of typical industrial automation systems with converged traffic of different types. In an example industry cooperation, a TSN manufacturing testbed from IIC (Industrial Internet Consortium) [6] was established by joint efforts to explore the TSN applications, readiness and interoperability in the industrial automation domain. Similar to the IIC testbed, the work of this paper also involves the collaborative efforts from both industrial automation vendors and network infrastructure partners. Specifically, this paper focuses on the approach of running present CIP Motion applications on a TSN infrastructure. This testbed evaluation verifies the benefits of TSN technologies and challenges of their integration with EtherNet/IP technology.

## Testbed Topology and Application

The testbed was established by joint efforts from Rockwell Automation, Cisco System and Analog Devices Inc. with their corresponding components as shown in Figure 1. As TSN technology mostly impacts critical traffic applications like motion control, we take CIP Motion control as the study case for this testbed. For evaluation of the *TSN Scheduling* in a converged network, interfering traffic is injected in the system alongside the motion traffic.

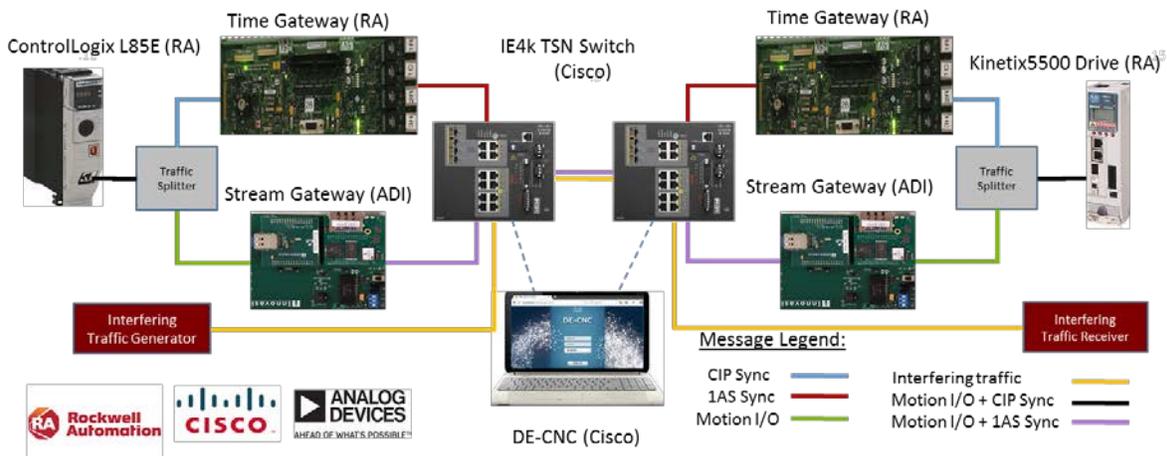


Figure 1 – Testbed of CIP Motion over TSN

With reference to the above terms, additional clarifications include:

- *Time Gateway* is a prototype implementation that provides the single same time scale for all the components in the system. It is the foundation for scheduling the isochronous type of critical application traffic across the network and application components.
- *Stream Gateway* is a prototype implementation that provides the TSN stream’s scheduling functionality based on *Stream Conversion*. It can integrate the legacy system traffic into the TSN network and protects critical traffic with scheduled transmission slots.
- *Traffic Splitter* could be a managed switch which is configured to route CIP Sync and CIP Motion messages through different paths. It is used on the present testbed because the *Stream Conversion* and time translation are separately implemented in the independent *Stream Gateway* and *Time Gateway*.
- *IE4k TSN Switches* create a TSN network, which implements the prototype of TSN standards including 802.1AS time synchronization, 802.1Qbv (Enhancements for Scheduled Traffic) and 802.1Qcc (Stream Reservation Protocol (SRP) Enhancements and Performance Improvements).

- *DE-CNC* implements prototype TSN configuration interfaces that are compliant with 802.1Qcc (Stream Reservation Protocol (SRP) Enhancements and Performance Improvements) and can configure IE4k *TSN Switches* via NETCONF.
- Controller and drive represent the user motion control application entities. On the present testbed, only one motion axis is controlled.
- Traffic Path Clarifications:
  - Different types of traffic are represented in different colors in Figure 1.
  - The communication paths for these messages are controlled by the *Traffic Splitter* and IE4k *TSN Switch*. The static MAC address table is configured for the path control.
  - STP (Spanning Tree Protocol) is turned off since all static MAC address tables are configured for path control.
  - Since the STP is turned off, communication loops in the system are prevented by access-list control configuration in each IE4k *TSN Switch*.

### Test Cases and Evaluation of TSN Adoption for CIP

We conducted several test cases to evaluate the components' capability and the whole system's performance when integrating TSN technologies into CIP Motion control. The detailed test cases and verification results will be presented in this section. These efforts are advisory to interest holders who are concerned with the TSN technology benefits and adoption key points for automation control applications.

For all the following test cases, the common configurations of applications and networks include:

- Except as noted, all test cases are run on the "100 Mb/s" Ethernet for two hours without application connection disconnection.
- In our test cases, the CIP Motion I/O messages are assumed to have a conservative maximum frame size of "100 bytes", although the variant packet size depends on the specific motion connection parameters.
- One CIP Motion axis control is configured with the application cycle of "10 ms", and the TSN network is configured with the network cycle of "10 ms". That is to say, the *Reduction Ratio* is "1".
- Both CIP Motion controller and TSN network components are configured to start their cycles at the integer times of seconds, i.e. "Top of Second". By this means, all the components in the system are aligned to the same start point of cycles.
- The interfering traffic is generated by a test tool in order to artificially bring the system to the point of CIP Motion system failure, with selected parameters:
  - The packet frame size is of "1500 bytes" on the wire
  - The messages are transported on UDP, the transport protocol for CIP Motion I/O traffic
  - The message is set with DSCP value of "55", which is same as that of CIP Motion I/O traffic
  - The pps (packets per second) is set with value of "8000". With this setting, the interfering traffic roughly needs the network bandwidth of "96 Mb/s".

Table 1 – Test Cases and Results

Test Case	Enabled TSN Features	Interfering Traffic Enabled	C2D Msg mean delay (us)	C2D Msg Jitter (us)	C2D lost packet counts	C2D late packets counts <sup>1</sup>
1	802.1AS time sync	No	208	38	0	0
2	802.1AS time sync	Yes	375	105	0	0
3	802.1AS time sync, 802.1Qbv TSN scheduling	No	758	14	0	0
4	802.1AS time sync, 802.1Qbv TSN scheduling	Yes	758	28	0	0

#### Test Case 1: CIP Motion Control System over Time Gateway

<sup>1</sup> Lost and late packet counts are reported by the CIP Motion implemented application and reflect packets that were not acted on by the application.

In this test case, the TSN network of the testbed is enabled with the 802.1AS time synchronization features. Since the legacy CIP Motion application operates CIP Sync protocol (default profile of IEEE1588), the *Time Gateway* is needed for translating between CIP Sync messages and 802.1AS time synchronization messages.

System performance for this test case is recorded in the Test Case 1 entry in Table 1. The performance data is the same as that of the normal CIP Motion application on a standard Ethernet network. And especially in the side work of observing the internal 1PPS (1 Pulse per Second) signals on the both controller and drives, we measure the time synchronization offset with the value around “**200 ns**”.

*This test case demonstrates that migration from IEEE 1588v2 default profile to IEEE 802.1AS clock synchronization is viable and can be implemented in a heterogeneous system with both techniques used for different devices.*

### Test Case 2: CIP Motion Control System over Time Gateway with Interfering Traffic

In this test case, the TSN network of the testbed is configured with *Time Gateway* as in test case 1 and injected with the interfering traffic in the C2D direction.

System performance for this test case is recorded in the Test Case 2 entry in Table 1.

*The performance degradation (both delay and jitter) validates previous analysis and shows that it is questionable to establish a converged network for both critical applications and interfering traffic with quality of service tags set to the same values.*

### Test Case 3: CIP Motion Control System over TSN

In this test case, the TSN network of the testbed is enabled with the features of 802.1AS time synchronization, 802.1Qbv compliant *TSN Scheduling* and 802.1Qcc compliant *TSN Configuration*. The *TSN Configuration* (covering *Stream Conversion* and *TSN Scheduling*) is performed in the *Stream Gateway* and *TSN Switch* with the stream schedules calculation performed by the *DE-CNC*.

*Stream Conversion* is configured in *Stream Gateway* based on the identification parameters of the application flows of interest and TSN stream identification rules on the bridge networks. Taking CIP Motion I/O traffic for instance, it is communicated on UDP with DSCP of “55” as defined in “3-7.5 Mapping CIP Traffic to DSCP and 802.1D” of [7]. So, it is preferred to identify the CIP Motion flows by using tuple-set of {IP address, DSCP}. In terms of TSN stream identification, it can be identified by the rules as defined in 802.1CB. Among available TSN identification rules, the “Null Stream Identification” rule is simple and implements a passive stream identification function that operates on a tuple-set of {Destination MAC address, VLAN} of stream packets, which is used to identify TSN streams on this testbed.

*TSN Scheduling* configuration is compliant with 802.1Qcc and needs the user to input application flows requirements together with the system topology.

- The system topology can be automatically discovered by LLDP protocol or manually established by users. By using the *DE-CNC* on this testbed, the IE4k *TSN Switch* can be automatically discovered but not the *Stream Gateway*. So, we have to manually add the *Stream Gateway*.
- After the semi-automatic process of establishing system topology, we will input the user application traffic requirements including cycle time, maximum frame size, maximum latency, earliest transmit offset, and latest transmit offset etc. On this testbed, the prototype CIP Motion control has the requirements of *Two-Cycle model for Isochronous Motion Control*. Specifically, the C2D transfer is

expected at beginning of each application cycle and D2C transfer is expected at the half time point of each application cycle.

- With user inputs, the *DE-CNC* calculates the schedules for the TSN streams converted from user application flows and downloads the configurations (e.g. time slot offset and size) into the *TSN Switch* via NETCONF. In absence of *CUC* on this testbed, we had to manually configure the *Stream Gateway* with the schedule and TSN Talker/Listener traffic specification as calculated by the *DE-CNC*.

System performance for this test case is recorded as in the Test Case 3 entry in Table 1. Although there is no interfering traffic, the performance indicators are worse than that in test case 1. This situation could be caused by:

- Factor 1 -- Earliest Transmit Offset (application traffic requirement):
  - According to 46.2.3.5.5 of [1], EarliestTransmitOffset specifies the earliest offset within the Interval at which the Talker is capable of starting transmission of its frames. As part of the TSN-defined Status group, the network will return a specific TimeAwareOffset to the Talker (within the earliest/latest range), which the Talker uses to schedule its transmit.
  - Ideally, the CIP Motion C2D message is sent at the beginning of the cycle, i.e. earliest Transmit offset is "**0 us**".
  - However, *Stream Gateway* needs additional time to process *Stream Conversion* and Stream Scheduling. So, it will introduce additional delay. This delay was measured as about "**300 us**". The earliest transmit offset should be about "**300 us**", as shown in Figure 2.
- Factor 2 -- Latest Transmit Offset (application traffic requirement):
  - According to 46.2.3.5.6 of [1], LatestTransmitOffset specifies the latest offset within the Interval at which the Talker is capable of starting transmission of its frames. As part of the Status group, the network will return a specific TimeAwareOffset to the Talker (within the earliest/latest range), which the Talker uses to schedule its transmissions.
  - Ideally, we could determine the latest Transmit offset by adding earliest Transmit offset with the jitter of "**38 us**" as measured in test case 1 of Table 1.
  - However, the present *DE-CNC* only allows the minimum of "**300 us**" difference between the earliest and latest Transmit offset. So the latest Transmit offset is set to "**600 us**" as shown in Figure 2.
- Factor 3 – Time slot size (CNC output)
  - The Latest Transmit Offset and Earliest Transmit Offset that are input by the user will affect the scheduled time slot size calculated by CNC. (Note there are other important factors affecting time slot size, e.g. the application payload size, and the guard band size that is used by CNC to prevent the in-transmission interference traffic). In a preceding hop in the network, the scheduled time slot represents the time span for egressing the packets and in the following hop, there needs a same size of time slot for ingressing the packets.
  - As all the packets within the time slot are valid for user's time-sensitive applications, the larger the time slot is, the more addition latency (in the worst case, the packet arriving at the last time point of time slot will still be valid and it will endure the delay equal to time slot size) will be introduced during a switch forwarding process. For instance, in our test, it takes the time slot of "**130 us**" as shown in Figure 3. Compared to strict priority transmission where no time slot is used, the most additional latency could be "**130 us**".
- Factor 4 -- Max Latency (application traffic requirement):
  - This is an input parameter depending on applications, e.g. C2D messages are required to arrive at drives before half of the cycle. The looser the value is, the easier it is successfully calculate schedules along the communication path for streams. But a loose value also means a big latency.
  - *DE-CNC* has the limitation of minimum "**500 us**" for this parameter, as shown in Figure 2. Since this value is used in the schedule calculation, it will determine the additional latency that is introduced in the form of the time slot size or switch processing time. Its impact on schedule calculation is determined by the scheduling algorithm of CNC.
- Factor 5 -- Switch Processing Time:
  - When scheduled streams go through a *TSN Switch*, there is an offset between stream ingress and stream egress, which is reserved for packet forwarding inside switches. In *DE-*

CNC schedule calculation, it takes a variable conservative value for each IE4000 TSN Switch. For instance in Figure 4 and Figure 5, the node latency is respectively “80 us” and “20 us”.

- Based on these impacting factors, we can roughly calculate the End-to-End latency (by assumption of neglecting the link delay) introduced by TSN Scheduling as: Factor 1 + Factor 2 + Factor 3 + Factor 4 + Factor 5 = **300 us + 2\*130 us + (in form of Factor 3) + (in form of Factor 3) + 100 us = 660 us**. Analysis of this delay indicates that at least 360us is due to known limitations of prototyping equipment such as software-implemented functions like switching that would normally be in hardware and configuration parameters not having full accessibility in software tools. While for non-TSN situation, the End-to-End delay (by assumption of neglecting the link delay) is **94 us**. As shown in the Figure 7, the addition latency by TSN Scheduling is **564 us**. This value could explain the experiment observation of additional delay introduced by adoption of TSN Scheduling.

*The performance degradation compared to Test Case 1 demonstrates that where there is no interfering traffic (with the same QoS tag values) adding TSN functionality to the network architecture delivers no significant performance benefit to any individual axis and by increasing the mean delay may result in a reduction in the number of axes supported in a given system.*

### Add Flow (Stream)

Flow Name (Stream ID)  Flow Address (Used with CT-Marker)

**General**

Max Frame Size (bytes)  Period

**End Stations**

Talker  Listeners

**Constraint**

Preset  Earliest Transmit Offset

Latest Transmit Offset  Max Latency

Figure 2 – Add TSN Streams by Specifying Constrains in DE-CNC

Device:

Stream ID	Stream Address	Priority	Interfaces	Receive Schedule (us)	Interfaces	Transmit Schedule (us)	Node Latency (us)
flow1	03:00:5E:A0:03:e9	5			Fa1/1	300-431	n/a

Figure 3 – Schedule Calculation Results for TSN Talker

Device:

Stream ID	Stream Address	Priority	Interfaces	Receive Schedule (us)	Interfaces	Transmit Schedule (us)	Node Latency (us)
flow1	03:00:5E:A0:03:e9	5	Fa1/7	300-433	Fa1/8	514-645	80

Figure 4 -- Schedule Calculation Results for First TSN Switch

Device:

Stream ID	Stream Address	Priority	Interfaces	Receive Schedule (us)	Interfaces	Transmit Schedule (us)	Node Latency (us)
flow1	03:00:5E:A0:03:e9	5	Fa1/7	513-645	Fa1/8	666-797	20

Figure 5 – Schedule Calculation Results for Second TSN Switch

Stream ID	Stream Address	Priority	Interfaces	Receive Schedule (µs)	Interfaces	Transmit Schedule (µs)	Node Latency (µs)
flow1	03:00:5E:A0:03:e9	5	Fa1/1	665-799			n/a

Figure 6 – Schedule Calculation Results for TSN Listener

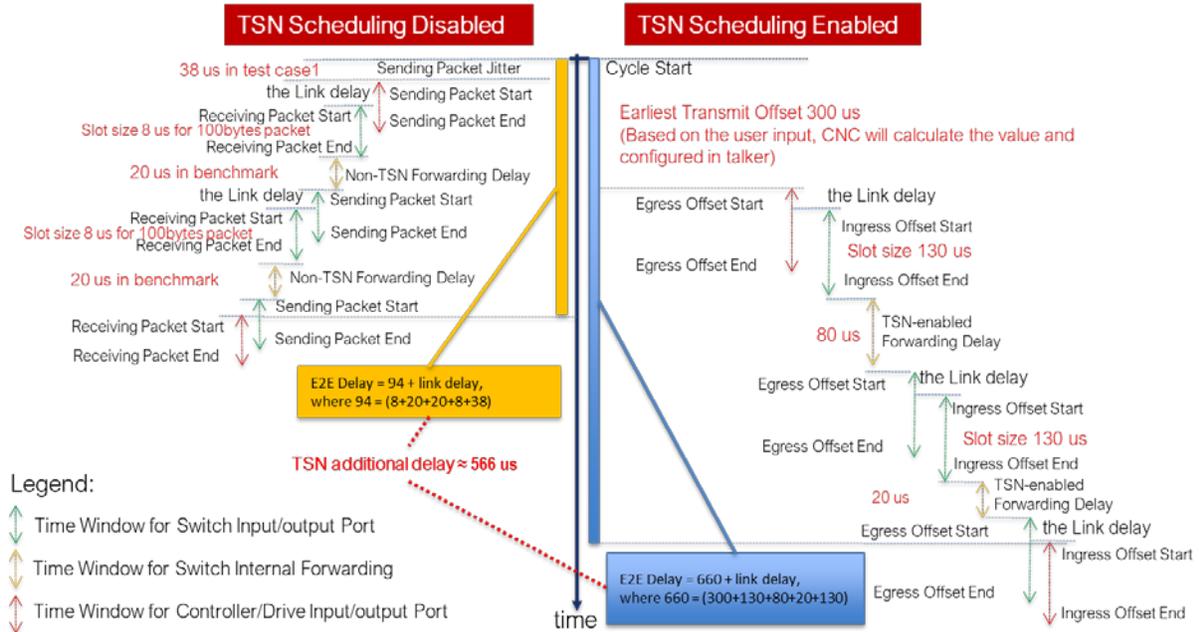


Figure 7 – End-to-End Delay Analysis

#### Test Case 4: CIP Motion Control System over TSN with Interfering Traffic

In this test case, the TSN network of the testbed is configured with TSN features as in test case 3 and injected with interfering traffic in C2D direction.

System performance for this test case is shown in the Test Case 4 entry in Table 1. With *TSN Scheduling* enabled, the critical CIP Motion traffic maintains the same performance even in the condition of interfering traffic. We also tested interfering traffic of “9000” pps (i.e. 108Mbit/s bandwidth). When *TSN Scheduling* is enabled, CIP Motion traffic still maintains the same performance. While in the case without *TSN Scheduling*, the communication connection will break down.

*With no performance degradation for CIP Motion traffic compared to Test Case 3 we have demonstrated that where there is interfering traffic (with the same QoS tag values) adding TSN functionality to the network architecture can ensure consistent CIP Motion operation in applications where there would have been either performance degradation or system failure using traditional Ethernet techniques*

#### Conclusions of TSN Adoption for CIP

In this section, we will conclude the experiment experience and propose the enhancements or requirements for incorporating the TSN technologies into CIP and EtherNet/IP technology.

#### Time Gateway for translating CIP Sync and 802.1AS

- *Time Gateway* is required for adopting TSN technology in the CIP Motion control system that uses CIP Sync for time synchronization. On this testbed, we proved the prototype *Time Gateway* can achieve a high precision of time synchronization between two endpoints and make the legacy system run over the Time Gateway without performance loss.
- *Time Gateway* needs to handle two time domains of default IEEE1588 and 802.1AS profiles on two ports, involving the functions of:
  - *Time Gateway*'s timing port in 802.1AS profile domain needs to support Peer-to-Peer synchronization at layer 2.
  - *Time Gateway*'s timing port in default IEEE1588 profile domain needs to support End-to-End synchronization at layer 3.
  - *Time Gateway* needs to synchronize two time domains to the same Grandmaster Clock
- The *Time Gateway* function is preferred to be combined with *Stream Gateway* function in one implementation entity. Otherwise, the system topology will be complex. As shown in Figure 1, the *Traffic Splitter* has to be used for routing time synchronization messages and control data streams via two separate paths.

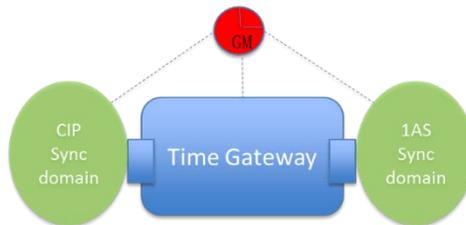


Figure 8 – Time Gateway for Translation between CIP Sync and 1AS Sync

#### Stream Conversion and TSN Scheduling (802.1Qbv)

- *Stream Conversion* between CIP flows and TSN streams is expected to leverage the “IP address + DSCP” rule. But this method has some issues:
  - This kind of “per-stream” identification is applied for connection with specific peers’ IP address. So, it will require too much work for configuring a complex system with many endpoints.
  - Since multiple CIP Motion I/O connections’ messages are not sent from the controller in a fixed order, it would be impossible to assign time slots for each motion connection by the “per-stream” identification and scheduling method. If only referring to the DSCP or alike for identifying certain traffic classes and neglecting the specific endpoints’ IP addresses, the “class-based” identification and scheduling mechanism is preferred in the case of multiple CIP Motion axes control. This is in compliance with the traffic classification by DSCP tag in the “3-7.5 Mapping CIP Traffic to DSCP and 802.1D” of EtherNet/IP specification [7].
  - The “Class-based Scheduling” idea is drafted as shown in Figure 9. One point is about the identification of traffic class. Another point is about the schedules and guaranteed End-to-End delay for each class of traffic. For instance, each egress port of bridges will assign an isochronous slot for the class of critical traffic. Although the sequence of streams are not fixed, the scheduled isochronous slot can ensure the worst-case End-to-End delay guarantee for all streams of this class.

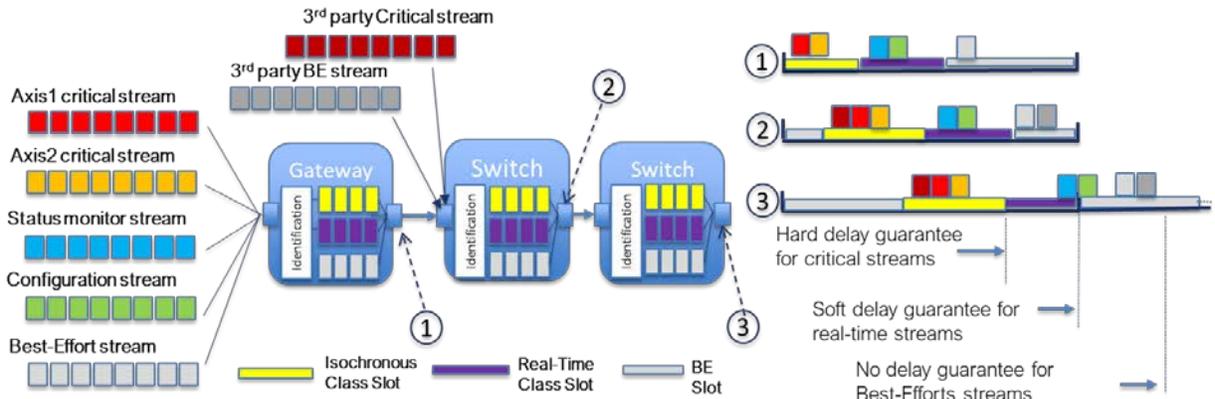


Figure 9 – Class-based Scheduling Idea

- Regarding TSN stream identification, it needs more study about the simplest rule of tuple-set {(multicast) Destination MAC address, VLAN}, considering the potential questions of:
  - Whether there are limitations when applying the stream identification for the “class-based scheduling” function in bridges. If we should use the extended tuple-set of {MAC address, VLAN, extended tags} like the work in 802.1CBdb [8].
  - Whether the Destination MAC address should be multicast or managed. What are the benefits or challenges for different allocation mechanisms of the Destination MAC address.
- TSN Scheduling requires a modification to the CIP Motion isochronous application model – it requires the application cycle be synchronized with the isochronous network cycle. Although the “Top of Second” practice can meet this purpose, it has some limitations:
  - The application cycle period has to be a value that is divisible by an integer into 1 second. That is to say, the cycle period can only be “1/n” second, where “n” is an integer as shown in Figure 10.

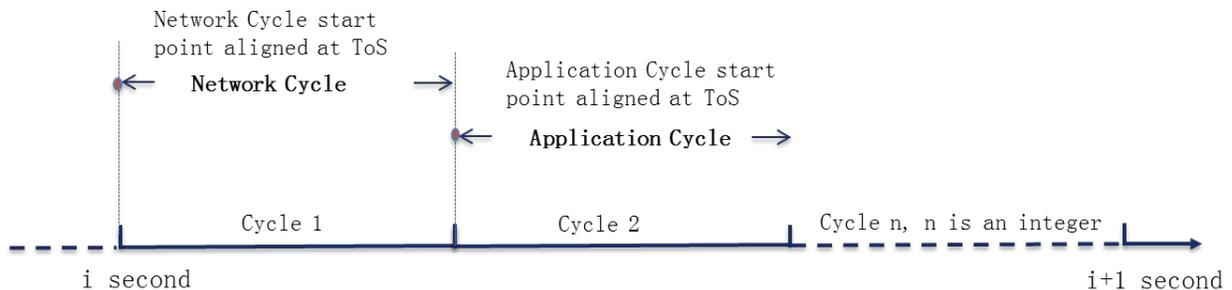


Figure 10 – Top-of-Second Alignment of Application Cycle and Network Cycle

- The “Top of Second” practice is not a standardized rule, which might cause compatibility issues. Actually in “8.6.9 Scheduled traffic state machines” of 802.1Qbv [9], definitions already exist for the “AdminBaseTime” and related attributes that can be used to set the network cycle’s start point by the state machine. If taking this approach, all the components need to implement the standard compliant state machines and protocols. Most probably, the applications will be responsible for coordinating the application cycle with the network cycle by the management interfaces.

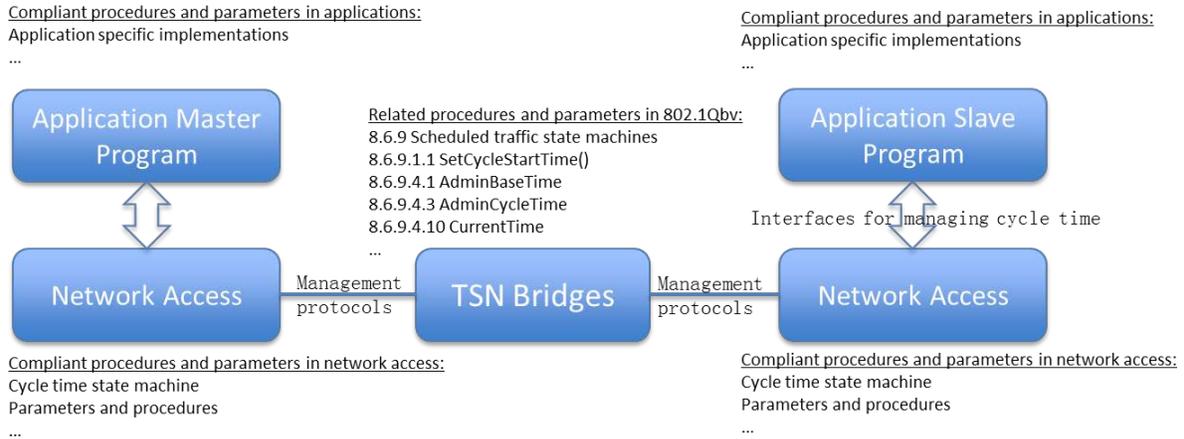


Figure 11 – Cycle Time Management Requirements

### TSN Configuration and 802.1Qcc

- The stream schedule calculation by CNC is an interactive process between applications configuration and CNC calculation. In this process, there are some issues of:
  - The process lacks efficient configuration guidance and most times the user must input multiple different application settings to achieve successful scheduling results by the CNC. It would be desired to design a wizard that can help users to efficiently figure out application requirements either by means of advisory template or parameter tuning suggestions based on CNC feedback.
  - For now, there are no available interfaces in user applications to fetch the required parameters for CNC calculation. Taking the “Earliest Transmit Offset” for instance, we only get the value of about “300 us” as shown in Figure 2 by additional side work. So the next urgent work for application vendors is to define and expose the interfaces for CNC to fetch application requirements.
- Except for the unimplemented parameters of standard 802.1Qcc, the present *DE-CNC* has limitations on user inputs as shown in the above section. That may be caused by the mismatch between CNC calculation engine capability metrics and user application requirement specification. So it will need joint efforts from both network vendors and application vendors to figure out the key and efficient interfaces for exchanging correct parameters, which shall be compliant with the 802.1Qcc standard, or may put forward new requirements to the project 802.1Qdj [10].

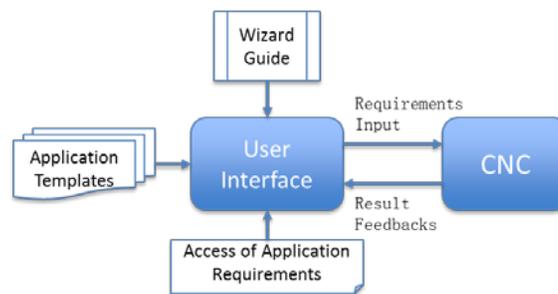


Figure 12 – Interactive User Interface for CNC Calculation

### Future Work

This paper introduces a testbed that adopts the subset of TSN features (i.e. 802.1Qbv, 802.1Qcc and 802.1AS) into the CIP motion application. It considers existing challenges and suggests a migration path of existing CIP Motion applications using TSN implementations and standards. In future work, it is desired to improve the testbed by solving known issues and prototyping solutions with TSN enhancements to solve motion application such as Class-based scheduling, 802.1Qbv based application and network cycle start alignment, combined time and stream gateway and native TSN end stations etc.. Based on this evolving testbed, it will be able to verify the TSN migration of EtherNet/IP technology with more comprehensive and solid proofs.

## References

- [1] IEEE, "8021Qcc-2018: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," 2018.
- [2] IEEE, "802.1CB-2017: IEEE Standard for Local and metropolitan area networks— Frame Replication and Elimination for Reliability," 2017.
- [3] IEC/IEEE60802, "Use Cases of IEC/IEEE 60802," 2018. [Online]. Available: <http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf>.
- [4] S. Zuponic, R. Klecka, M. Hantel and P. Didier, "TSN Influences on ODVA Technologies," ODVA 2017 Industry Conference & 18th Annual Meeting, Palm Harbor, Florida, USA, 2017.
- [5] R. Blair, "Analysis of Converged Network Traffic Using Time Sensitive Networking," ODVA 2018 Industry Conference & 19th Annual Meeting, Stone Mountain, Georgia, USA, 2018.
- [6] P. Didier, G. Ditzel, L. Leurs and J. Woods, "TSN - Report on the IIC Testbed," ODVA 2017 Industry Conference & 18th Annual Meeting, Palm Harbor, Florida, USA, 2017.
- [7] ODVA, "EtherNet/IP Adaptation of CIP Edition 1.24," 2019.
- [8] IEEE, "P802.1CBdb: Draft Standard for Frame Replication and Elimination for Reliability — Amendment:Extended Stream identification functions," 2019.
- [9] IEEE, "802.1Qbv-2015: Enhancements for Scheduled Traffic," 2015.
- [10] IEEE, "P802.1Qdj: Bridges and Bridged Networks Amendment: Configuration Enhancements for Time-Sensitive Networking," 2019.

\*\*\*\*\*  
The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use. Copyright ©2020 ODVA, Inc. All rights reserved. For permission to reproduce excerpts of this material, with appropriate attribution to the author(s), please contact ODVA on: TEL +1 734-975-8840 FAX +1 734-922-0027 EMAIL [odva@odva.org](mailto:odva@odva.org) WEB [www.odva.org](http://www.odva.org). CIP, Common Industrial Protocol, CIP Energy, CIP Motion, CIP Safety, CIP Sync, CIP Security, CompoNet, ControlNet, and EtherNet/IP are trademarks of ODVA, Inc. All other trademarks are property of their respective owners.