A TSN Network By Example
by Volker E. Goller

INTRODUCTION
The time sensitive networking (TSN) evaluation kit allows customers to connect almost any existing Ethernet device to a TSN network. In the simplest case, such a network is a pure line topology, consisting of several of these TSN evaluation kits. However, the kit can operate with TSN solutions from other vendors as long as they support the same feature set. Presently, the kit supports 802.1Qbv (scheduled traffic) and 802.1AS (clock synchronization).

The kit provides three Ethernet ports, see Figure 1. Nearly any standard Ethernet device is supported, including devices that run EtherNet/IP, PROFINET RT, ModbusTCP, BACnet IP, or any other IP-based protocol except PROFINET IRT, EtherCAT, SERCOS, or POWERLINK.

This application note is designed to lead a user of the Analog Devices, Inc., TSN evaluation kit (referred to as the TSN kit throughout) through the processes necessary to configure and run a TSN network. This application note does not replace or remove the need to read the TSN kit documentation and must be used in conjunction with the TSN Evaluation Kit Quick Start Guide and the fido5100/fido5200 Real-Time Ethernet Multiprotocol (REM) Switch data sheet.
TABLE OF CONTENTS
Introduction ................................................................. 1
Evaluation Board Diagram .............................................. 1
Revision History ........................................................... 2
Getting Started ............................................................... 3
  Evaluation Board Ports ................................................. 3
  Power Supplies ........................................................... 3
  Configuring the TSN Kits ............................................... 3
The Example Setup ......................................................... 4
  the Host Port ............................................................... 5
Using Wireshark and a TAP .................................................. 6
Basic Setup Completed and Operational ............................ 6
Checking 802.1AS ................................................................. 7
Setting Up the TSN Streams .............................................. 8
Creating and Setting up an 802.1Qbv Schedule ................... 10
The Schedule for this Demo ............................................. 17
Summary ................................................................. 18
References ................................................................. 18

REVISION HISTORY
10/2017—Revision 0: Initial Version
GETTING STARTED

Follow the documentation in the TSN Evaluation Kit Quick Start Guide to setup the PC and connect it to the TSN evaluation kit.

EVALUATION BOARD PORTS

It is important to connect the PC to one of the TSN Ethernet ports for configuration. There are three ports: the host port (located on the narrow side of the printed circuit board (PCB)), where an existing Ethernet device must be connected. The standard Ethernet traffic of the device is forwarded to the switched TSN ports (located on the wide side of the PCB). It is essential that such forwarding is working prior to configuring any TSN feature.

It is not possible to access the configuration website via the host port. The IP address of a TSN kit out of the box is 192.168.1.1.

POWER SUPPLIES

Power the device using the power supply that is shipped with the kit. Alternatively, the screw terminals can also be used to provide power. The board is designed for 9 V to 24 V dc range.

If the screw terminals are used, be careful to connect per the markings on the PCB. Note that the PCB is delivered turned off.

CONFIGURING THE TSN KITS

In the following example, four TSN kits are configured in a line topology; therefore, all TSN kits are configured with unique media access control (MAC) addresses, as well as unique IP addresses. These addresses are set via the menu selection, Gateway Settings, on the TSN evaluation kit webpage. Take care to enable writing of MAC addresses by closing the write enable jumper first. It is necessary to set up one TSN kit at a time.

Note that it is best practice when setting MAC and IP, to use the lowest three digits of the device serial number for the lowest three digits of the respective addresses. Because the serial number is always printed on the dual RJ45 connector of the TSN ports (see Figure 1 for the location of RJ45), the lowest three digits are 109. In the rare case of an address conflict, use more of these digits from the device serial number. Doing so helps to keep track of the multiple boards/IP addresses.

If used in a line topology, it is best to lay down the boards in the order of ascending or descending IP numbers, whatever makes more sense to the user. This helps to keep track of the changes made to the boards. On the PC used to configure the kits, open a separate tab in the browser for each board and connect those tabs to the kits in the same physical order of appearance (see Figure 2).

Figure 2. Organizing Addresses and TSN Kits, Ports Aligned with the Layout of the Schedule Configuration Page
THE EXAMPLE SETUP
The full setup is shown in Figure 3. The four TSN kits are connected in a line topology and form the TSN segment. At this stage, TSN is not configured and everything runs standard Ethernet. On the boundary ports (the open ports at the ends of a TSN segment), standard non-TSN devices are connected. In this demo, a webcam is to the right and a display or PC to show the video from the webcam is on the left of the TSN segment.

To display the webcam livestream, the same PC used to configure the demonstration can also be used to display the livestream. That way, the webcam video can be shown while still being able to tweak the TSN kit setups via the web servers.
THE HOST PORT

One important step in setting up a TSN network using the TSN kit is to make the MAC address of the device that is connected to the host port known to the gateway. By assigning this address to the host port, the 2-port TSN switch can identify packets that are directed to this port, or more precisely, directed to the device that is connected on that port. Technically, this MAC address is added to the switch's static forwarding database.

To accomplish the task of setting the host MAC, the MAC addresses of the device that is connected to the host port must be known. On automation equipment, such as PROFINET devices, the MAC is printed on the device housing, see Figure 4. Once identified, the host MAC address needs to be entered in the client MAC field in the gateway settings page of the web server.

On Windows® systems, the physical address can be retrieved via the network adaptor status. Click the Details button in the General tab located in the Local Area Connection Status dialog box. See Figure 5 for more information.
USING WIRESHARK AND A TAP

If the MAC is not available, the easiest method may be to use a tool like Wireshark to observe the packets that enter and exit the device. Packets ingressing the device have the MAC address in the destination MAC address field of an Ethernet frame (as long as no multicast or broadcast message is used), and packets that egress the device have the device MAC address in the source MAC address field of the Ethernet frame.

Using Wireshark is essential to debug a setup. Installing and becoming familiar with Wireshark is highly recommended prior to setting up the TSN evaluation kit.

However, with Wireshark, not all traffic is visible because packets are not flooded within an Ethernet network; instead, they use the forwarding database in each switch to determine through which port they should egress for the shortest routing path to their destination. Therefore, acquiring an Ethernet wire test access point (TAP) is recommended.

A TAP is a passive Ethernet component that does not send traffic on its own nor does it introduce delays; rather, it allows one to listen to all packets that flow by in either direction. TAPs are available from many vendors, a quick internet search for “Ethernet TAP” or “Ethernet test access point” returns many options, including build-at-home options. For this example, the KUNBUS TAP2100 is used because it can TAP two connections simultaneously and has a built-in high resolution clock base (8 ns).

Being able to TAP two channels simultaneously facilitates debugging, allowing a TAP before and after a switch to determine if the switch filtered a message out or if it translated a message into a TSN stream correctly.

BASIC SETUP COMPLETED AND OPERATIONAL

When all boards and devices for the demonstration are wired according to the example setup shown in Figure 3, all TSN kits can be accessed using the web browser, according to the configuration shown Figure 2, all host MAC addresses are added to the gateways settings, and the basic setup is completed. TSN aside, this setup should work just as any other Ethernet Layer 2 segment.

An advantage of TSN is that it is an extension to standard Ethernet and the switches talk on Layer 2 rather than on Layer 3, permitting the use of different IP subnet addresses to avoid address conflicts.

Without dealing with TSN, it is still possible to set up all the communication relationships that are needed:

- Create a project to connect the S7-1511 with the ET200S. IP subnet is 192.168.0.xxx.
- Connect the IO Link demo using IP subnet 192.168.10.xxx.
- Set up and connect the webcam using IP subnet 192.168.21.xxx

Figure 6. An Ethernet Wire TAP in the Setup
CHECKING 802.1AS

After setup, the 802.1AS time synchronization is operational. One of the boards assumes the role of the grandmaster clock, identified by the 802.1AS status LED turning red (see Figure 1). On a sync slave device, the 802.1 status LED turns green when clock sync has been established.

Checking the status of AS on the TSN evaluation kit webpage, it is important to understand that AS is a peer-to-peer clock. The grandmaster is in the master role to both TSN ports, but the other kits report a slave role towards the grandmaster and a master role towards nodes that are further away from the grandmaster. The Port Role box reports if the port is in slave or master mode. If the gateway is not a grandmaster, one port reports that it is in slave mode and the other port reports that it is in master mode. If the gateway is the grandmaster on the TSN network, then both ports report that they are in master mode, see the TSN Evaluation Kit Quick Start Guide for more information. The peer-to-peer scheme is shown in Figure 7.

Note that after a schedule is created and enabled, the 802.1AS service requires Queue 3 to be open at some point during the TAS cycle.

Figure 7. 802.1AS Peer to Peer Sync
Assign the Grandmaster Function

On startup, the devices select a clock grandmaster using a best master clock algorithm (BMCA). In most cases, this is adequate and does not require further attention. However, there may be scenarios where a fixed grandmaster is preferred.

Using the Local Master Priority1: and Local Mater Priority2: entries on the Time Synchronization page, the best master algorithm can be steered to select a specific board for the grandmaster clock. Figure 8 shows an adjusted TSN kit. Priority 1 is set to 200 (higher priority than the default 248; therefore, this board is assigned the grandmaster clock role).

<table>
<thead>
<tr>
<th>Table 1. MAC, PCP, and VLAN Settings for Demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>PROFINET S7-1511 to ET2005</td>
</tr>
<tr>
<td>ET200S to S7-1511</td>
</tr>
<tr>
<td>IO Link Master to Demo</td>
</tr>
<tr>
<td>Demo to Master</td>
</tr>
</tbody>
</table>

There are many degrees of freedom to setup the MAC, PCP, and VLAN values, as long as all TSN nodes are controlled. For this setup, the PROFINET cyclic communication receives the highest priority. IO Link and PROFINET TSN streams are set apart by the second lowest byte of the MAC address; however, the VLAN ID can be used for the very same purpose. In such a case, the following list in Table 2 represents a valid setup. The streams are shown in Figure 9.

<table>
<thead>
<tr>
<th>Table 2. Alternative MAC, PCP, and VLAN Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>PROFINET S7-1511 to ET2005</td>
</tr>
<tr>
<td>ET200S to S7-1511</td>
</tr>
<tr>
<td>IO Link Master to Demo</td>
</tr>
<tr>
<td>Demo to Master</td>
</tr>
</tbody>
</table>

Setting Up the TSN Streams

A TSN stream identifications (IDs) is constructed of a locally administered multicast address and an 802.1Q VLAN tag. In this example setup, the following streams are used, as listed in Table 1.
There is one major difference between IO Link and PROFINET traffic. Whereas PROFINET traffic uses Destination MAC Address AND Priority Code Point (PCP) of VLAN tag as a method to identify the traffic that requires translation (talker method) to a TSN stream, the IO Link demo uses a transmission control protocol (TCP) connection. Therefore, it uses Destination IP Address AND Protocol AND Source Port on the talker side and Destination IP Address AND Protocol AND Destination Port for the IO Link PC (listener side). As with most TCP connections, the server side port address is usually stable, although the client port number may be different. This is similar to a web server where the default port is 80 (see Figure 10).

Using Destination MAC Address AND Priority Code Point (PCP) of VLAN tag for PROFINET allows assignment of the high priority data traffic to a TSN stream. PROFINET tags all prioritized traffic using VLAN PCP 6 (see Figure 11).
CREATING AND SETTING UP AN 802.1Qbv SCHEDULE

After the hardware and the communication is set up and the 802.1AS clocks are synchronized, a schedule can be created. However, it is advisable to recap some Ethernet basics first.

Recap: the Ethernet Frame and TSN

From a software point of view, an Ethernet frame consists of the Layer 2 header (destination and source address, EtherType), up to 1500-byte data (assuming jumbo frames are not used) and a CRC32. The Ethernet header and the data field are the portion of the Ethernet frame that is copied into the main memory of a computer when it receives a frame.

In addition, on the wire an Ethernet frame contains a 63-bit preamble, a 1-bit start delimiter, and a minimum of a 12-byte line silence gap (see Figure 12). An Ethernet frame must have a minimum length of 64 bytes (including CRC). The relatively long preamble and silence time, and the minimum length have their origins in older Ethernet standards; however, they are still maintained and guaranteed. Whereas some PHYs allow shortening the preamble to 16 bits, this is not a common approach.

A TSN stream adds four bytes to the standard Ethernet frame, the VLAN tag. The stream always includes a VLAN tag. The VLAN tag consists of a priority (PCP) and a VLAN ID. Together with the destination MAC, the VLAN ID identifies a TSN stream. Each TSN stream can have one, and only one, transmitter (talker) and one or many receivers (listeners). These four bytes must be added in any frame length calculation (see Table 3). The minimum time a frame occupies on the wire, at 100 Mbps, is 6.72 μs, and the time for a full 1500-byte data frame is 123.36 μs.
Table 3. Ethernet Frame Timing

<table>
<thead>
<tr>
<th>Rate (Mbps)</th>
<th>Rate (MBps)</th>
<th>Byte Time (µs)</th>
<th>Layer2 (Byte)</th>
<th>Preamble (Byte)</th>
<th>Gap (Byte)</th>
<th>SDU, Minimum (Byte)</th>
<th>Slot, Minimum (Byte)</th>
<th>Slot Time, Minimum (µs)</th>
<th>Slot Time, Maximum (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.25</td>
<td>0.8</td>
<td>22</td>
<td>8</td>
<td>12</td>
<td>42</td>
<td>84</td>
<td>67.2</td>
<td>1233.6</td>
</tr>
<tr>
<td>100</td>
<td>12.5</td>
<td>0.08</td>
<td>22</td>
<td>8</td>
<td>12</td>
<td>42</td>
<td>84</td>
<td>6.72</td>
<td>123.36</td>
</tr>
<tr>
<td>1000</td>
<td>125</td>
<td>0.008</td>
<td>22</td>
<td>8</td>
<td>12</td>
<td>42</td>
<td>84</td>
<td>0.672</td>
<td>12.336</td>
</tr>
</tbody>
</table>
The Timing Model of an Ethernet Transmission Channel

Although Ethernet is fairly fast, there are delays in the transmission channel to consider. For a 100 Mbps, full-duplex Ethernet environment, consider the following parameters:

- **PHY delays**: the Ethernet PHYs introduce delays during receiving and transmitting that are caused by the internal processing of the data. The receive (RX) delay is usually greater than the transmit (TX) delay. Also, the RX timing may cause some jitter in the range of a few nanoseconds. These delays are in the range of a few hundred nanoseconds and are a parameter of the PHY components. Because a single byte occupies 80 ns each, the PHY delays are actually in the range of a 1-byte to 4-byte delay.

- **Bridge delay**: the Ethernet switch introduces a bridge delay (the time a message is delayed traversing through a switch). In the case of a fast cut-through, 2-port switch, the bridge delay is small. The switch must receive an 8-byte preamble plus a 12-byte source/destination MAC plus a 4-byte VLAN tag before making a forwarding decision. These requirements add up to a minimum 1.92 µs bridge delay (note that there are other factors that may result in slightly higher bridge delays, as well).

- **Cable delay**: a typical CAT5 cable has a cable delay of 5.7 ns per meter, or approximately half a bit transmitted. For any tabletop setup or cabinet with short cables, the cable delay is about one order of magnitude below the PHY delays. However, for a full 100 m Ethernet cable, the cable delay adds another 570 ns (equivalent to ~7 bytes).

To calculate a complex schedule for a system with serious physical span (long cables) and/or many nodes requires that all of these delays be considered. For this tabletop demonstration, add a margin of 10,000 ns to the schedules for dependable results.
802.1Qbv Scheduler Operation

IEEE defines up to eight queues for Qbv, also referred to as time aware shaper (TAS). IEEE also defines an array of bytes, one byte for each defined offset within the cycle. The bits in those bytes enable or disable a queue accordingly. With each time slice within a cycle, a byte defines which of the queues are enabled and which not. Among the enabled queues, the ones with higher priorities are served. It is important to understand that the slices can have any start offset within the cycle; these offsets are defined in nanoseconds. The cycle time is also defined in nanoseconds.

The Ethernet transmitter, when idle, fetches an Ethernet frame from the enabled queue that has the highest priority of all enabled queues. While the transmitter is busy sending that frame, it does not fetch further frames (which means there is not another transmit queue).

The current implementation of TAS on the TSN kit is limited to 4 queues and up to 16 offsets, also called gate open events, within a cycle.

Figure 14 shows a block diagram of a TAS scheduler. In this diagram, all queues have some Ethernet frames queued up and four offsets are defined: 0 ns, 500 ns, 1000 ns, and 2000 ns. In this example, the cycle time is 2500 ns. Figure 14 shows an active 2000 ns offset, resulting in blocking Queue 2 and Queue 3 and enabling Queue 0 and Queue 1. Because Queue 1 has a higher priority than Queue 0, the transmitter fetches the next frame from Queue 1 until Queue 1 is either empty or the current slice is completed. If the current slice is completed, the next slice at Offset 0 has Queue 0 enabled, but as Queue 3 has messages queued up and has higher priority, the next frame to be sent is selected from Queue 3.

When all queues are enabled at the same time, the TAS works like a classical quality of service (QoS) priority queue, serving the highest priority queue first.

Figure 14. 802.1Qbv Scheduler
Guard Band

When using a schedule to guard bandwidth, it is necessary to ensure that if a higher priority gate is open, the transmitter is idle. If the transmitter is not idle, a potential long frame of lower priority can potentially block, or at least delay, higher priority traffic. To guarantee that the transmitter is idle and can start immediately sending higher priority traffic, a guard band is inserted. While active, the guard band inhibits the transmitter from fetching next frames.

To be effective, the guard band must be long enough so that the longest possible message that can be queued in a particular queue completes fully before the next gate open event occurs. The size of that longest frame that can be queued is defined in the parameter, maximum service data unit (maxSDU), in bytes.

There is no need to manually create guard bands by disabling all queues because the TSN kit automatically attends to the guard band.

Figure 15 shows a simple example. At $t_0$, the scheduler is enabled for the first time. Prior to that time, left of $t_0$, all gates are open. For the transmission gate enabled time, the transmitter can fetch frames from Queue 0 (Q0, in this example, wherein one gate at a time is opened). The guard band time (see Figure 15) is the guard band ahead of the Queue 3 open event and is wide enough so that a full 1522 SDU size frame can be sent safely, before the Q3 gate event occurs.

When Q3 is open, both periods are much shorter because Q3 is configured to transport shorter frames only. Its maxSDU may be just 200 bytes.

The maxSDU for Queue 0 is fixed at 1522 to allow web traffic for the configuration web server to pass through Queue 0 without being dropped.

![Figure 15. Schedule Timing, Startup, Guard Band](image-url)
**Ethernet Frame Size, Frame Rate, and Asynchrony**

Because the frame size must be defined for each queue, knowledge of the protocol being placed on a designated queue is required. For a cyclic communication, length and rate of the traffic is easily calculated. However, for protocols that use a variable packet size (for example, anything over TCP) a worst-case value is required. For example, ModbusTCP can be used for cyclic communication; however, TCP uses very short handshake packets in addition to the data packets. In addition, Modbus knows several commands to access a device, some of which are longer than others. To place a ModbusTCP into a schedule requires worst-case estimates on the maxSDU bytes.

For PROFINET RT, it is possible to define a stream for the cyclic data exclusively, requiring few or no extra frames in the stream. Thus, estimating the maxSDU is more precise. Whereas the TSN kit is not synchronized with PROFINET RT (and the PROFINET RT protocol has no means to do such), the cyclic communication of PROFINET runs on its own clock. To guarantee that each frame transmits within the PROFINET cycle time, the TAS cycle time must be smaller (faster) than that of PROFINET; for best practice, a factor of 2 faster.

The reasons why creating an effective, tight schedule is not simple are as follows:

- Frame size may vary. For the schedule, the worst rather than the most frequent case must be taken into account.
- Frame rate may vary because of additional frames sent (handshake, rare commands, and so forth).
- Application cycle is not synchronized with the TAS cycle (oversampling required).
- Streams with different cycle times need to coexist on one wire. One way to solve such problems is to use a slower TSN schedule and open the gate for the faster traffic multiple times within the TSN cycle. See Figure 16 as an example.

It is possible to synchronize an application with the TAS scheduler. The TAS sync pulse (when a cycle is starting or completed) is available on Pin A10 of the CPU module on the TSN kit. An application can use this signal to synchronize to the TAS cycle; consequently, oversampling is not required.

![Figure 16. A Schedule Supporting Different Cycle Times; PROFINET Is Scheduled Twice As Many Times As “SomeOther” Traffic](image-url)
Lastly, the block diagram of the TSN kit (see Figure 17) must be considered. There is a TAS scheduler for each Ethernet port. If a frame is received through one of the Ethernet ports, the TSN kit checks for a known TSN stream ID. In a case where a TSN stream is identified, the reverse stream translation is applied and the frame is sent out to the host port. If it is unidentified, or if it is best effort traffic without a matching IP address, the frame is placed in the “other ports” TAS queue for transmission.

In PROFINET IRT, which used scheduled traffic even before TSN became available; this simplified forwarding is referred to as relative forwarder. Only traffic that is not forwarded is identified and the transmission time for the forwarded frames is relative to the receive time (plus the bridge delay). This contrasts with an absolute forwarder in PROFINET nomenclature.

An absolute forwarder looks up the destination port first and then sends out the frame at an absolute time (within the schedule). This absolute time must account for the bridge delay and the lookup time. Therefore, an absolute forwarder is slightly slower and quite a bit more complex than a relative forwarder.

One application for an absolute forwarder is a multiport switch, where the lookup for the egress port must occur anyway.

If a frame is received through the host port, it is checked according to the defined stream translation filters. If it is not translated, the traffic is queued in the best effort queue of the TAS. The appropriate port is selected by means of the dynamic forwarding database of the switch; however, if a stream translation occurs, it can be defined through which port the frame is to egress.

Therefore, the stream translation configuration page offers a selection of Port 1, Port 2, or both ports simultaneously for the desired egress port.

![Figure 17. Block Diagram of the TSN Kit](image-url)
THE SCHEDULE FOR THIS DEMO

For this demonstration, a TAS cycle time of 2 ms is selected with more than 50% of the bandwidth assigned to best effort traffic (1 ms). The traffic timing is listed in Table 5.

On the far right and far left ends of the TSN segment of this demonstration, the traffic towards the TSN segment is scheduled only. The resulting schedule is the same for each port (see Table 4).

Table 5. Traffic and Timing

<table>
<thead>
<tr>
<th>Stream</th>
<th>Queue</th>
<th>Protocol</th>
<th>maxSDU</th>
<th>Rate</th>
<th>Slot Time</th>
<th>Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Effort</td>
<td>0</td>
<td>http, video</td>
<td>1522</td>
<td>N/A</td>
<td>1,882,240</td>
<td>N/A</td>
</tr>
<tr>
<td>F1:D0: F1:D0:00:01</td>
<td>1</td>
<td>TCP, IO Link</td>
<td>1000</td>
<td>&lt;&lt;&lt;TAS</td>
<td>81,600</td>
<td>6670 ns</td>
</tr>
<tr>
<td>F1:D0: F1:D0:00:05</td>
<td>1</td>
<td>TCP, IO Link</td>
<td>1000</td>
<td>&lt;&lt;&lt;TAS</td>
<td>81,600</td>
<td>6670 ns</td>
</tr>
<tr>
<td>F1:D0: F1:D0:01:01</td>
<td>2</td>
<td>PROFINET IO</td>
<td>80</td>
<td>4 ms</td>
<td>8000 ns</td>
<td>10,000 ns</td>
</tr>
<tr>
<td>F1:D0: F1:D0:01:05</td>
<td>2</td>
<td>PROFINET IO</td>
<td>80</td>
<td>4 ms</td>
<td>8000 ns</td>
<td>10,000 ns</td>
</tr>
<tr>
<td>802.1AS</td>
<td>3</td>
<td>PTP</td>
<td>110</td>
<td>2 ms</td>
<td>10,400 ns</td>
<td>1000 ns</td>
</tr>
</tbody>
</table>

1 N/A means not applicable.
2 In this example, 110 is the length of a PTP frame. However, in a bigger network a TLV within the follow_up message can grow substantially. In such cases, the actual size or a worst-case size must be applied.
SUMMARY
When 802.1Qcc, the configuration protocol for TSN, becomes commonly available, no one must calculate a schedule manually nor be concerned about how to configure a board using a web-server. However, understanding the implications and the benefits of a scheduled network is a first step into this technology. Debugging and understanding performance of a scheduled network requires this level of understanding. The knowledge gathered is valuable for when TSN is implemented fully in the field.

The Analog Devices TSN evaluation kit is a convenient and inexpensive way to gather real-world experience with TSN.

REFERENCES
- IEEE TSN Standards: 802.1AS, 802.1Q, 802.1Qbv