Wireless Time-Sensitive Networking (WTSN) Reference Software for Linux* with Intel® Wireless AX210/201

User Guide

September 2022

Intel External Engagement

Introduction

This User Guide describes the Intel Wireless Time Sensitive Networking (WTSN) Reference Software for Linux*. It includes three demos and walks users through running them as well as understanding the features and capabilities of this reference software.

Validated Configurations

The Intel Wireless Time Sensitive Networking (WTSN) Reference Software for Linux* has been validated on the following:

Hardware	• Platform: Intel NUC11 Platform (NUC11TNKv7), Intel Tiger Lake, Intel Gemini Lake, • Intel® WiFi AX210, Intel® WiFi AX201
Software	• Ubuntu 18.04, 20.04 • Linux Kernel 5.17.15

NOTE

- It is possible that the Wireless Time Sensitive Networking Reference Software works in other configurations. For full support, using the supported configuration is required. However, you are free to use the WTSN Reference Software in other configurations. If an issue can be reproduced on the supported configurations, it can be addressed. Otherwise, it is not supported.
- To reduce typographical errors, consider using Secure Shell (SSH) to copy and paste commands in running the demos. SSH cannot be used in all situations so evaluate your setup to decide what works best for you.

Revision History

Revision	Date	Changes
1.1	Oct 2022	Updates to include Preempt_RT and OPC UA
1.0	Oct 2022	Initial release

Get Started

This section helps users get started using the Wireless Time Sensitive Networking (TSN) Reference Software for Linux. It includes the following subsections:

Prepare the Host Machine and install WTSN stack	Describes setting up the host machine, getting, installing, and setting up the WTSN reference stack, required tools and environment.
CPU Clock Optimization	Provides steps to optimize the platform for maximum performance, minimize packet drop etc. due to CPU speed fluctuation. Covers configuring the following, if available: • CPU Power saving mode • Intel SpeedStep® technology

Step 1: Prepare the Host Machine

1. Prepare host machine with Ubuntu* OS 18.04 or 20.04.
2. Install required dependencies for installing the WTSN stack.

```
$ sudo apt update
$ sudo apt install -y screen libreadline-dev libncurses5-dev \
 libdbus-1-dev libnl-genl-3-dev \
 libssl-dev gawk bison flex iw
```

3. Clone/Download the intel Wireless TSN stack repository located at https://github.com/susruths/WTSN-Binaries.git to ***wtsn*** folder. Henceforth the wtsn folder will be referred to as the root for all steps.
4. Install kernel version from the wtsn folder. If on an unsupported platform, download kernel, If on an unsupported platform, download kernel version 5.17.15 from www.kernel.org, compile and install the kernel for your host platform. ***Note***: Do not proceed to Step 5 until you have successfully completed Step 4.

```
$ cd wtsn
$ sudo dpkg -i *.deb
```

5. Install WTSN stack

```
$ cd wtsn
$ ./wifi-install.sh wtsn_multi-q-patch-src.tar.gz –no-reboot
```

Accept all default prompts, if any.

Step 2: Platform CPU Clock Optimization

Tuning the BIOS makes runtime behavior more deterministic; features such as power management are disabled, and the CPU is running at top speed. Performance outside of the context of determinism may be negatively impacted. Consider the following tuning configuration to execute the TSN Reference Software demos. Based on your use case, you can decide later to use a different configuration.

Enable Intel SpeedStep® Technology and C-States in BIOS

1. While the platform is booting, press F2 to enter the BIOS menu
2. Go to Device Manager > System Setup > CPU Configuration > CPU Power Management

Enable Performance Mode

Follow these steps to change the CPU frequency from scaling governor to performance mode. This ensures the CPU is running at the highest frequency.

NOTE

Repeat these steps with every restart.

1. Start a new terminal
2. Enter the following:

```
# echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
# echo performance > /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
# echo performance > /sys/devices/system/cpu/cpu2/cpufreq/scaling_governor
# echo performance > /sys/devices/system/cpu/cpu3/cpufreq/scaling_governor
```

3. **[Optional]** Check the CPU clock frequency:

```
# cat /proc/cpuinfo | grep MHz
```

NOTE

Once the CPU Clock is optimized, the CPU Frequency should typically be over 1500 MHz.

Demo 1: IEEE 802.1AS Time Synchronization over WTSN and Quality Measurement

The time synchronization demo demonstrates:

- Time synchronization between the grandmaster clock and the slave clock by means of the linuxptp software stack. This involves the use of PTP Hardware clock support in the Intel WiFi cards, 802.11mc Fine Timing Measurement Protocol as well as the 802.11v Timing Measurement protocol support and corresponding frame receive and transmit timestamping capability of the Intel® WiFi AXI210/201 cards. To enable this a new WiFi port support has been added to the supported ports of the linuxptp time synchronization stack.
- Time synchronization quality measurement using:
  - 1PPS generation output from SDP0
  - Auxiliary Time Stamping (AUXTS) of the PTP clock as triggered by 1PPS signals into SDP1

The Intel® Ethernet Controller I210 provides both these features.

To run this demo, follow these steps:

Step	Estimated Time Taken
Time Synchronization Demo 1 Step 1: Set up the Hardware	30 minutes
Time Synchronization Demo 1 Step 2: Setup Time Synchronization	1 minute
Time Synchronization Demo 1 Step 3: Measure Time Synchronization Quality	45 minutes
Time Synchronization Demo 1 Step 4: Verify Time Synchronization Quality	15 minutes

NOTE

Time estimates can vary widely depending on your network speed and system processing power. These estimates use an Intel® Core™ i7 processor.

Introduction to IEEE 1588 and IEEE 802.1AS-2011

Time Sensitive Networking has two salient components:

1. Time synchronization
2. Traffic shaping

The Intel® WiFi AX210/201 offers PTP clock and time-stamping capability for the corresponding transmit and receive frames.

IEEE 1588-2008, also known as Precision Time Protocol Version 2 (PTPv2), enhances the accuracy of time synchronization between two networked nodes from millisecond (achievable by Network Time Protocol (NTP)) to microsecond or sub microsecond. This is possible as the corresponding time exchange packet timestamping is done at the hardware, instead of software, level. The transport of PTP messages can be over UDP/IPv4, UDP/IPv6, IEEE802.3 Ethernet or IEEE802.11mv/v time measurement frames.

IEEE 802.1AS-2011, also known as generalized Precision Time Protocol (gPTP), is based on IEEE 1588-2008 and, being an 802.1 standard, can be applied to a wide range of heterogeneous networks, such as Ethernet, Wireless, Media over Coax Alliance and HomePlug. The WTSN stack supports gPTP based time synchronization.

The primary components of gPTP are:

- **Path delay measurement**: The path delay between connected devices over bridges/switches is a slowly varying value due to physical condition such as temperature of HW components or network cables and, in the case of wireless/WiFi, time-varying channel conditions. A gPTP device measures the path delay by exchanging Pdelay request, Pdelay response, and Pdelay response follow-up messages between the initiator and responder as shown in the figure 1 below. In fact, all devices including the device that has grandmaster clock can be the path delay initiator. The Figure 2 shows how these frames are exchanged in the case of WiFi. Here th gPTP payload is exchanged over FTM (Fine Timing Measurement) frames supported by the Intel WiFi AX210/201 devices. These transmit and receive times of these frames are time stamped by the device and propagated up to the clock management software.
- **Time distribution**: The device acting as the Leader clock periodically sends the SYNC packet that contains time of the day along with a timestamp of when the SYNC message was sent. gPTP specifies the use of IEEE 1588 two-step processing where the said timestamp value is sent on a subsequent message called "Follow Up" message. Through SYNC and Follow-Up messages and with its known path delay, the device with the Slave clock will constantly adjust the PTP clock to keep synchronized with the time of the Leader clock.

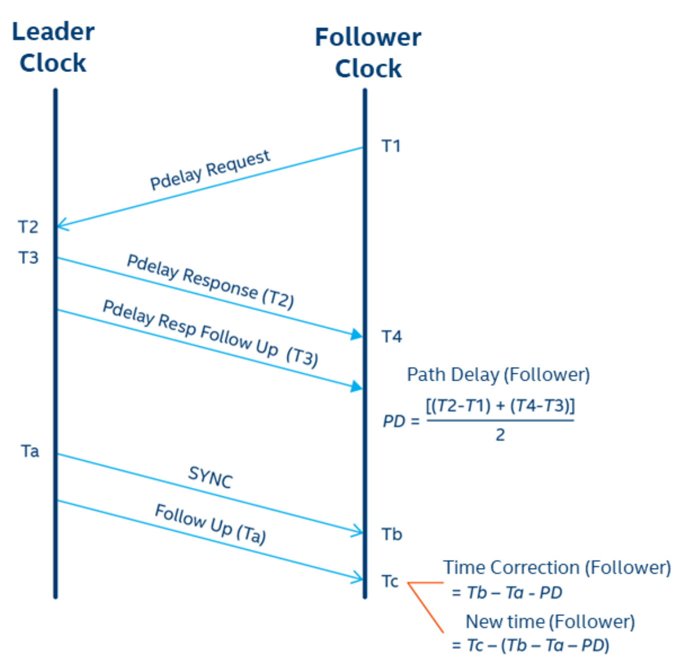

*Figure 1: **gPTP** in action between a leader and a follower*

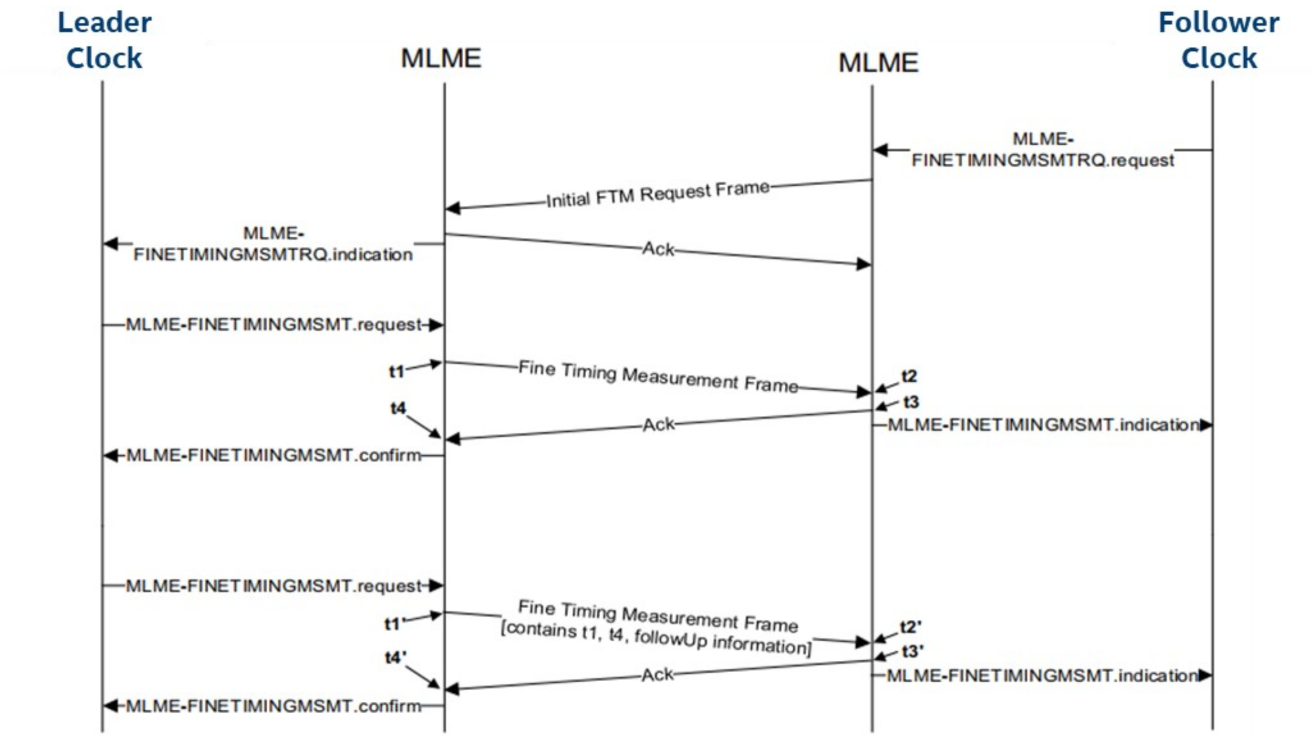

Figure 2: Exchange of gPTP frames over IEEE802.11mc (Fine Timing Measurement Frames)

Time Synchronization Software overview

The clock management software or time synchronization software components replicate a reference or leader clock locally using 802.1AS derived reference to network device clock offset and the intra-platform cross-timestamp. In WTSN the clock management software used is the linuxptp

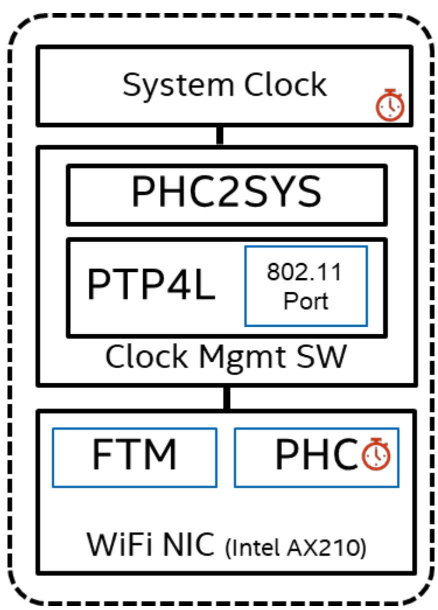

Figure 3: Time Synchronization software components.

with support added for WiFi card. The following components have been added to support the exchange of timing frames over 802.11

- **PTP Hardware Clock (PHC) support**: Provides interface to the HW timestamping clock. Provides interface to the HW clock in the WiFi NIC and ability to query time and cross-timestamp.
- **Fine Timing Measurement Frames**: Provides an interface to generate FTM frames to exchange the gPTP timing frames. These frames are also timestamped by the HW timestamp clock. The 802.11 Port support added to the linuxptp software acts as the glue between the clock management software and this interface.

Time Synchronization Demo

The following table lists the prerequisites for the time synchronization demo.

Units	Hardware/Equipment Specifications
2	One of the supported hardware platforms
2	Intel WiFi AX 210/201 NIC with WTSN stack installed
2	Intel Ethernet i210 or i225 (for measurement setup)
1	CAT-5E Ethernet Cable (for measurement)
1	A commercial Wireless Access Point
NOTE For information on using other platforms and boards, refer to the Validated Configurations section.	

Time Synchronization Demo: Step1 Hardware Setup

The following figure shows the hardware set up for the time synchronization demo.

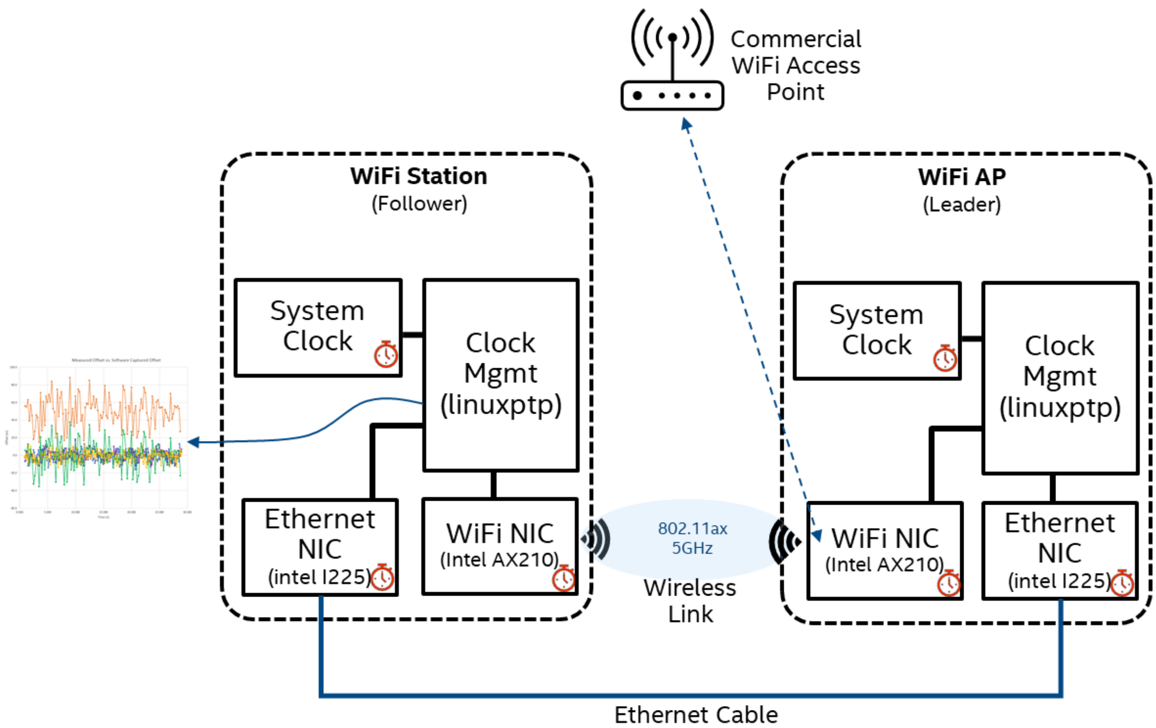

Figure 4: Time Synchronization Demo Setup

Each host platform contains an Intel WiFi AX210 or Intel WiFi A201 card which will be used to establish a wireless connection and synchronize time between them. Each host platform also contains an Intel® Ethernet Controller I225 or Intel® Ethernet Controller I210 card which will be used for measuring the time synchronization performance over the wireless links. The demo setup will also need a commercial 802.11 access point. As one of the hosts is going to act as a "Soft" access point, a connection to a certified commercial access point is required to operate in 5GHz band.

For the purpose of this demo, assign these two platforms as shown below.

SoftAP	The platform which will act as an 802.11 SoftAP and as the PTP grandmaster clock.
STA	The platform which will act as an 802.11 station and connect to the SoftAP as well as synchronize its time over wireless to that of the AP.

To set up the time synchronization demo, follow these steps:

1. Connect the two host platforms using the ethernet cable and the Ethernet NIC.
2. Power on the commercial access point and configure a 5GHz SSID in the commercial Access Point (refer to the appropriate manual). For the purpose of the demo and instruction, we will set to this SSID to "Real_AP_5GHZ".

Time Synchronization Demo 1 Step 2: Time Synchronization setup

NOTE

If you are using the BSP in Build the Yocto Project*-Based Image on the Host Machine, you can skip this step. The dependencies and applications are already built into the image you are using.

1. Setup the SoftAP. Execute the following commands on the **specified node.**

<table>
<tr><th>Host</th><th>Command</th><th>Notes</th></tr>
<tr><td>[SoftAP]</td><td><code>$ cd wtsn</code>
<code>$ sudo mkdir -p /etc/wpa_supplicant /etc/hostapd</code>
<code>$ sudo cp -rf hostapd-softap.conf /etc/hostapd/</code>
<code>$ cp -rf wpa_supplicant-5g.conf /etc/wpa_supplicant/</code>
<code>$ cp -rf wpa_action_script.sh /usr/sbin/</code></td><td></td></tr>
<tr><td>[SoftAP]</td><td><code>$ sudo ./startAp_5g.sh</code></td><td>Start the SoftAP</td></tr>
<tr><td>[SoftAP]</td><td><code>$ sudo screen -ls</code>
<pre>There are screens on:
        105041.AP   (10/09/2022 03:10:18 PM)   (Detached)
        105021.STA_WPA (10/09/2022 03:10:16 PM)   (Detached)
        105012.STA_5G   (10/09/2022 03:10:14 PM)   (Detached)
3 Sockets in /run/screen/S-root.</pre></td><td></td></tr>
</table>

NOTE

The above performs the following actions.

1. Create a virtual MAC interface to connect to the real commercial access point
2. Connect to the real access point (SSID: “Real_AP_5GHZ”)
3. Once connected, create a SoftAP using the real Wireless Interface with SSID TODO
4. Delete the virtual interface

It is important that the real commercial AP is setup and configured as per previous section. This also assumes that the WTSN stack has been setup correctly.

2. Establish wireless link between **STA** and **SoftAP.**

<table>
<tr><th>Host</th><th>Command</th><th>Notes</th></tr>
<tr><td>[STA]</td><td><code>$ cd wtsn</code>
<code>$ sudo mkdir -p /etc/wpa_supplicant</code>
<code>$ cp -rf wpa_supplicant.conf /etc/wpa_supplicant/</code></td><td>Initialize the wireless interface and connect to SSID: TODO</td></tr>
</table>

[STA]	`$ sudo startSta.sh`	

3. Setup Time Synchronization between AP and STA over Wireless link

Host	Command	Notes
[SoftAP]	`$ cd wtsn` `$ sudo ./setup_wsync.sh -b master -p <mac address of peer>`	Start the time sync on the SoftAP (leader). This will start pt4l first and wait for user input to start phc2sys. Follow the instructions and prompts to proceed.
[SoftAP]	`$ cd wtsn`	
[STA]	$ cd wtsn $ sudo ./setup_wsync.sh -b slave -p <mac address of peer >	
[STA]	$ sudo screen -ls	Verify that everything is started

Time Synchronization Demo 1 Step 2: Measure time synchronization quality

This section describes the process to measure performance of time synchronization quality.

To measure the time error introduced by the time transfer function across the wireless link, we use the ethernet NIC on the host. Figure 5 illustrates how this measurement process is setup.

1. The System clock at the STA is synchronized over WiFi to the System clock at the SoftAP.
2. The PHC clock on the Ethernet NIC on the STA is also synchronized to the System Clock of SoftAP.
3. To get the time sync error (T_{ERROR}), the offset of PHC time on the Ethernet NIC at the STA with the System Time at the STA is computed. The assumption made here is that the time synchronization error across the wired NICs, which support hardware time synchronization, is much smaller than the time synchronization error across the wireless and so this can be ignored in comparison with the time synchronization error across wireless.

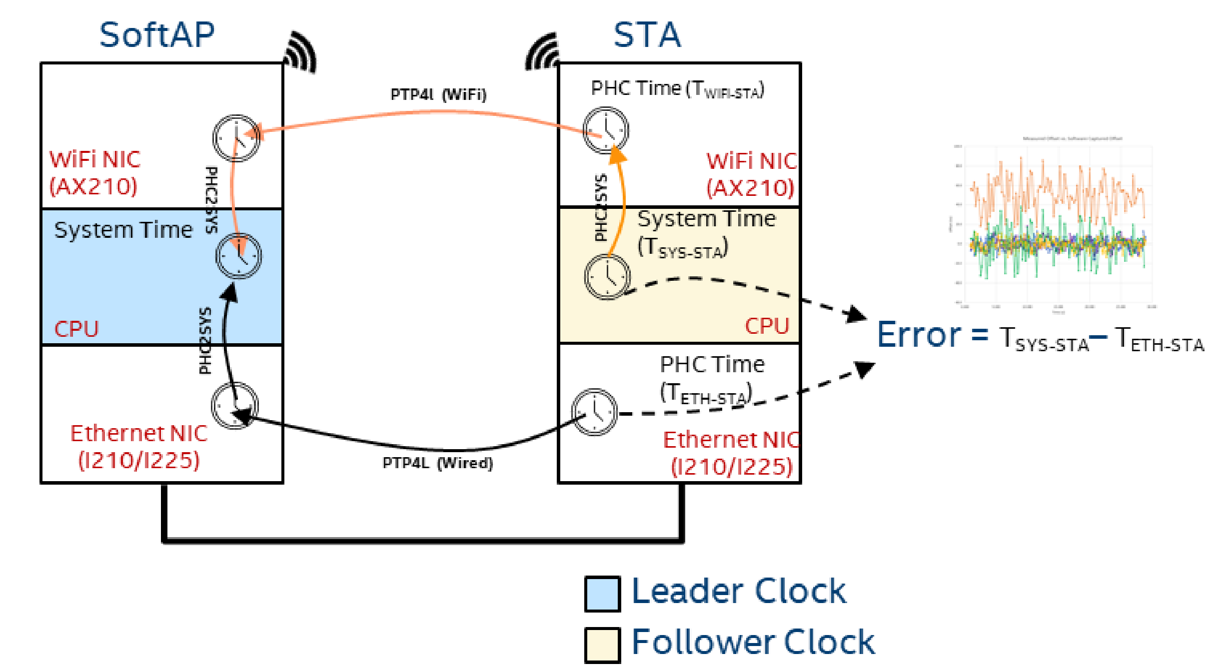

Figure 5: Time Sync Measurement Setup

To setup the demo, follow the steps listed below. These steps assume that the time synchronization is running across the wireless link to sync the system clocks of STA to SoftAP.

Host	Command	Notes

[SoftAP]	`$ cd wtsn`
	`$ sudo screen -dmS ETH_PTP4L ptp4l -i enp109s0 -f gPTP.cfg -s -m`
	`$ sudo screen -dmS ETH_PHC2SYS phc2sys -s CLOCK_REALTIME -c <Eth NIC Interface Name> -s -m`
		Synchronize the System time at the leader to the PHC time of ETH NIC at the leader.
		Setup ptp4l as leader to serve the PHC time.

[STA]	`$ cd wtsn` `$ sudo ptp4l -i enp109s0 -f gPTP.cfg -m`	Synchronize the PHC time of the ETH NIC at the STA to that of the leader (SoftAP).
[STA]	`$ while true; do offset=$(sudo phc_ctl /dev/ptp2 cmp | grep "offset from" | awk '{print $6}' | sed -e 's/ns//'); echo $offset; sleep 1; done | tee time_offsets.csv`	Measure the time offset between the System Clock and the ETH NIC PHC clock at the STA.

The time_offsets.csv file contains the error between the System Time at the STA, which is disciplined by the time synchronization process running over wireless, and the time at the PHC clock at the STA, which is disciplined by the time synchronization running over wireless sampled at 1 second intervals. This time error can be plotted as a histogram of over time and should get something like below.

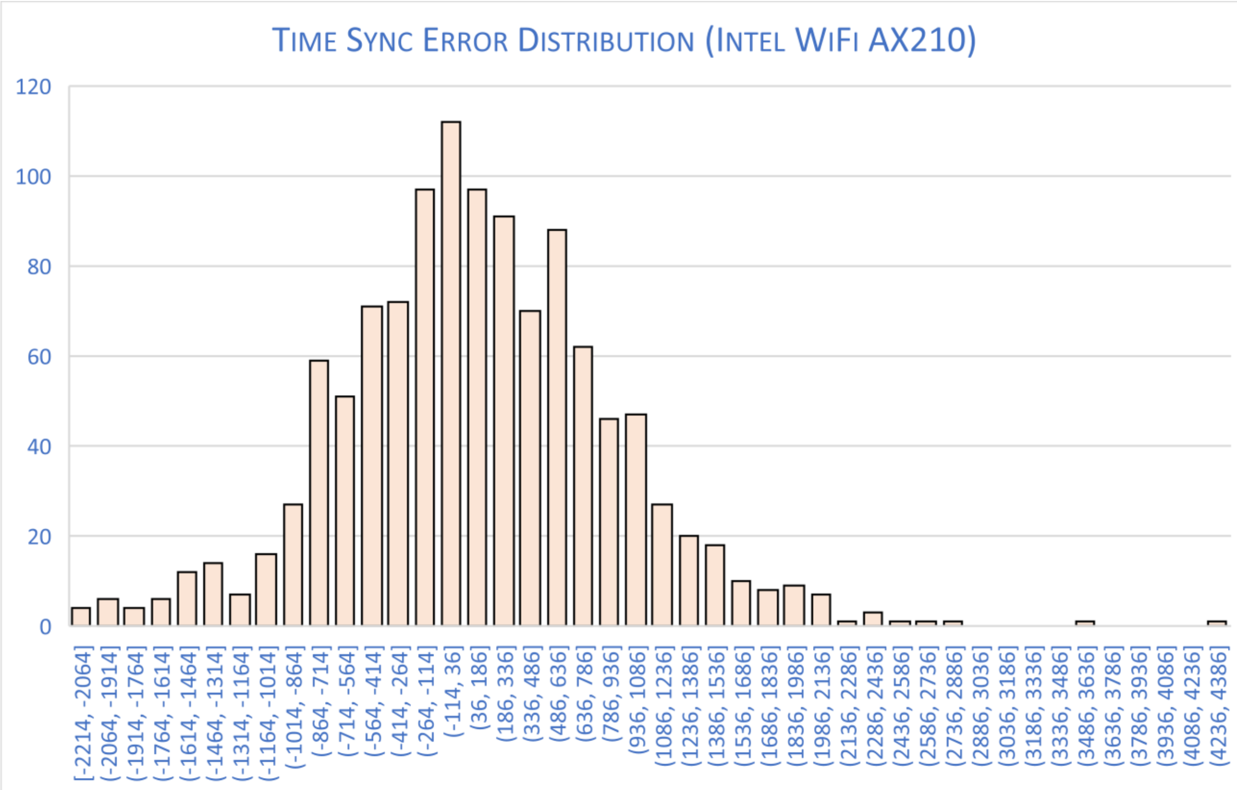

Figure 6: Time Synchronization Error over Intel WiFi AX210

Here the 99th percentile is ~2 microseconds and the maximum error is ~4 microseconds.

Demo 2: IEEE 802.1Qbv Time Aware Shaper

TO Be Updated