pcm4l

PTP Clock Manager for Linux

The PTP Clock Manager for Linux (pcm4l) is designed as a high-performance clock recovery solution for packet-based time/phase/frequency synchronization. The pcm4l software includes the features required to meet the ITU-T Telecom packet equipment clock recommendations, including a packet clock state machine.

In complex networks with partially unknown packet delays, typical in telecom applications, pcm4l offers patented filter algorithms that help to improve synchronization accuracy with time/phase accuracy of under 1 microsecond and frequency accuracy of under 16 parts-per-billion (ppb).

In networks with full IEEE 1588 support that are made up of Boundary Clock (BC) or Transparent Clock (TC) nodes, the pcm4l offers an outlier detection algorithm, used in conjunction with the Renesas ITU-T G.8273.2 hardware filter, which helps to improve synchronization accuracy with time/phase accuracy of under 5ns filtered (provisional Class D). These tight requirements dictate the use of timing hardware for the synthesis of the physical clock.

The adaptive servo is implemented in software running on a host processor that controls the PTP Hardware Clock (PHC). The IEEE 1588 protocol also runs on a host processor but does not need to be co-located with pcm4l (i.e., a different processor). The pcm4l architecture and interfaces makes it ideal for open-source architectures, such as White Box hardware and O-RAN.

For more information about the PTP Clock Manager, see the pcm4l product page.

Application Standards

- ITU-T G.8275.1 Appendix V and G.8275.2 Appendix IV PTP clock state machine
- ITU-T G.8273.4 T-BC-P/T-TSC-P
- ITU-T G.8263 PEC-S-F
- ITU-T G.8265.1 Master Selection Algorithm (MSA)

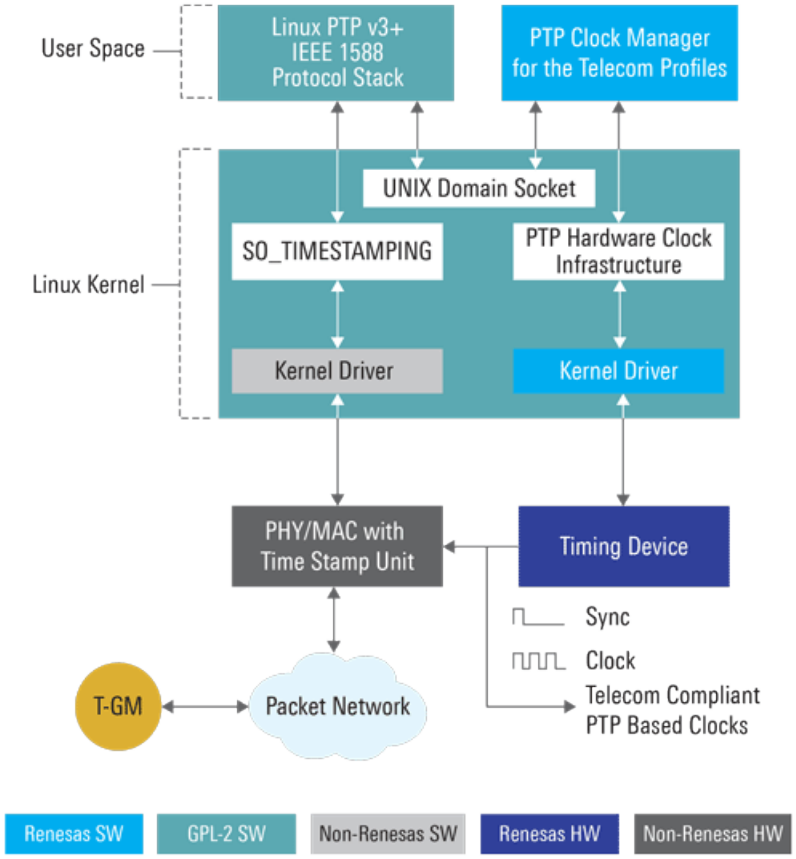
Features

- Filters the effects of high Packet Delay Variation (PDV) per G.8273.4 or G.8263
- Calibrates for asymmetry when an external local time reference is available per G.8273.4 APTS
- Software provided as clear C code, ready for Linux distributions
  - Runs on Linux kernel v3.0 and higher, using the PTP Hardware Clock Infrastructure for clock control
  - Dramatically simplifies software integration and debugging
- Support for IEEE1588-2019 Slave Event Monitoring channel to allow for a timestamp interface to an external PTP stack
- Full PTP clock state machine per ITU-T G.8275.1 Appendix V and G.8275.2 Appendix IV

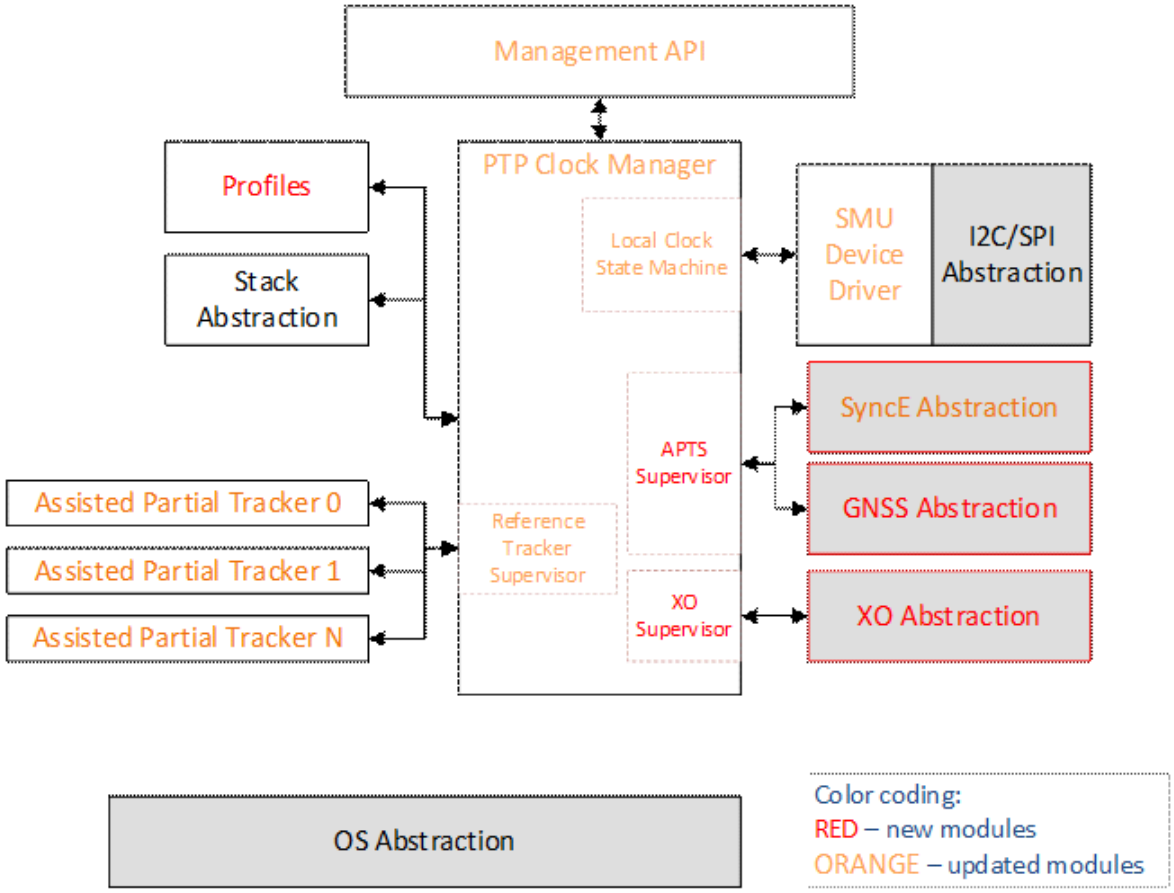
Applications

- Wireless infrastructure for 4.5G and 5G network equipment
- Phase/time synchronization with full timing support from the network per ITU-T G.8275.1
  - Telecom Boundary Clocks (T-BCs) and Telecom Time Slave Clocks (T-TSCs) according to ITU-T G.8273.2 for Full Timing Support (FTS) networks
- Phase/time synchronization with assisted or partial timing support from the network per ITU-T G.8275.2
  - Telecom Boundary Clocks (T-BCs) and Telecom Time Slave Clocks (T-TSCs) according to ITU-T G.8273.4 for Assisted Partial Timing Support (APTS) or Partial Timing Support (PTS) networks

Simplified Diagram



Block Diagram



Contents

1. Principles of Operation

The IEEE 1588-2019 Precision Time Protocol (PTP) can be used to achieve precise synchronization of clocks in packet-based networked systems. The protocol supports synchronization accuracy and precision in the sub-microsecond range with minimal network and local computing resources, and allows for customizations by means of Profiles. With a properly designed network, sub-nanosecond time transfer accuracy can be achieved.

The challenge of any network synchronization mechanism is how to make local clocks that are traceable to a primary clock source. It is also desirable to implement a clear set of equipment clock recommendations. For Telecom, these industry agreed architectures come from the ITU-T.

1.1 Simplified Synchronization Model

The basis of PTP is built on the exchange of PTP timing messages, as illustrated in Figure 1. The IEEE 1588 standard defines two types of delay mechanisms: delay request-response and peer delay. Only one of the two is needed, and all ITU-T Telecom Profiles use the delay request-response. PTP assumes transmission delays between the master clock to the slave clock and the return path are symmetric.

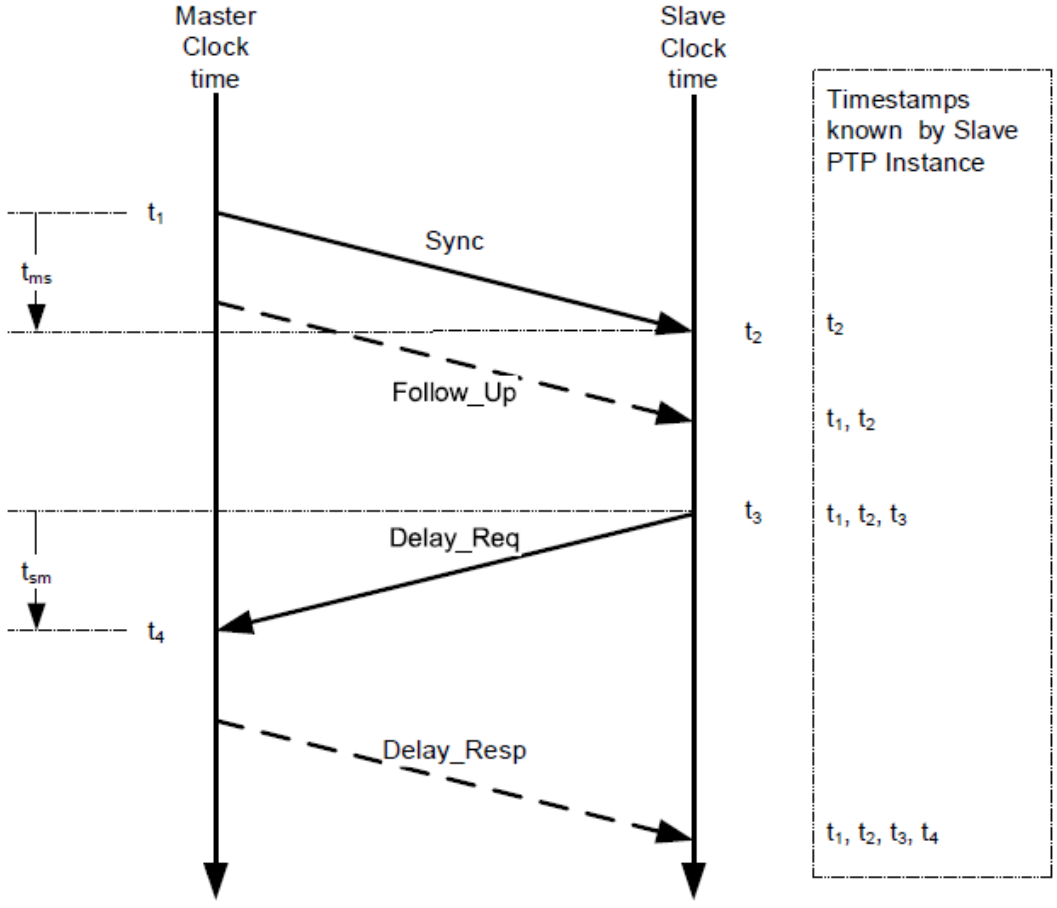


Figure 1. Basic PTP Timing Message Exchange (from IEEE 1588-2019)

From above, two measurements can be obtained:

1. Offset From Master, or <offsetFromMaster>
   - = <Time on the Slave Clock> - <Time on the Master Clock>
   - = <syncEventIngressTimestamp> - <originTimestamp> - <meanDelay> - <correctedSyncCorrectionField>
   - = t2 - t1 - <meanPathDelay>[1] - <correctedSyncCorrectionField>

1. Using delay request-response mechanism (used by all ITU-T Telecom Profiles).

2. Mean Path Delay, or <meanPathDelay>[2]
   - = [(<syncEventIngressTimestamp> - <delayReqEventEgressTimestamp>) + (<delayReqEventIngressTimestamp> – <originTimestamp>) – <correctedSyncCorrectionField> – <correctedDelayRespCorrectionField>]/2
   - = [(t2 - t3) + (t4 – t1) – <correctedSyncCorrectionField> – <correctedDelayRespCorrectionField>]/2

The timing diagram in Figure 2 shows a simplified example using the PTP timing messages exchange based on an End-to-End (E2E) delay mechanism. Once the slave has all timestamps, it is in position to calculate the offset from master, offsetFromMaster.

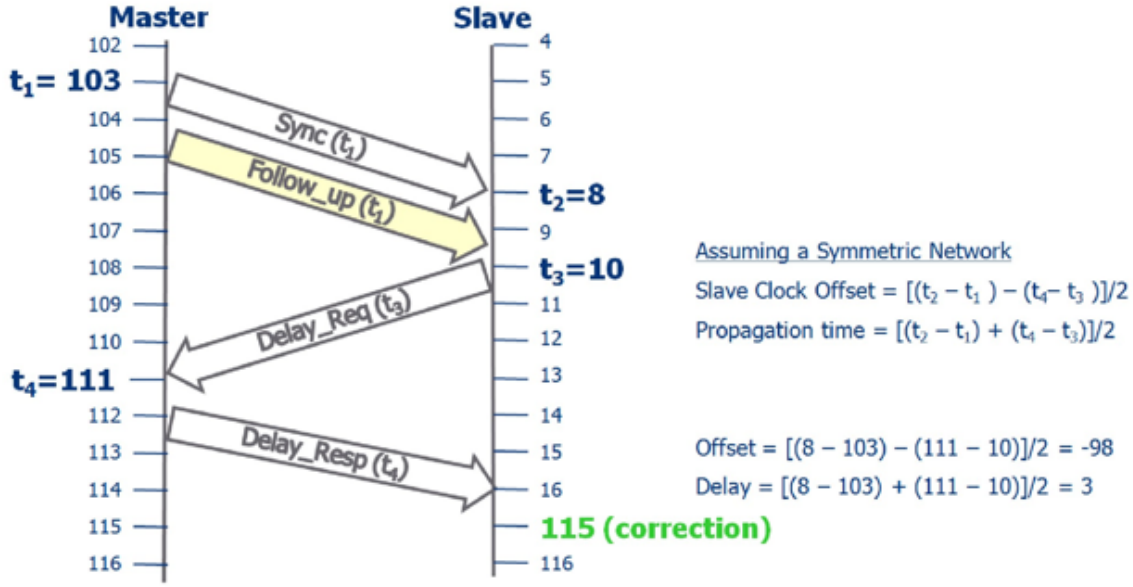


Figure 2. Example Synchronization Using the IEEE 1588 Timing Message Exchange

There are some simplifications assumed with these calculations (e.g., no corrections are made to account for any path and/or data asymmetry or noise generation from the master clock). This is where more complex filtering schemes will be required, which is part of the Renesas PTP Clock Manager (PCM). The PCM software also includes other features required to meet the ITU-T Telecom Equipment Clock recommendations, including a packet clock state machine. In complex networks with partially unknown and varying packet delays, such as telecom applications, the patented filter algorithms help to improve synchronization accuracy with phase accuracy of under 1 microsecond and frequency accuracy of under 16 parts-per-billion (ppb).

The tight requirements of network synchronization dictate the use of special hardware to generate the physical clock. The Renesas family of System Synchronizer PLLs such as ClockMatrix provide precise, low-noise solutions for this function. In order to accommodate ever-changing requirements of the network, the servo is implemented in software running on an external processor that controls the PLL. The protocol and master selection algorithm are also run on the external processor.

There are both open-source and commercial source options for the PTP stack. Most have very similar capabilities because they are based on the IEEE 1588 standard to guarantee interoperability. For Linux-based systems, a popular PTP stack is Linux PTP, or ptp4l. The ptp4l software provides all the required features per IEEE 1588-2008, along with Profile support. It uses the Linux kernel SO_TIMESTAMPING socket option for timestamp management of the PTP event packets and the PTP Hardware Clock infrastructure for clock control. v3.0 has support for all three ITU-T Telecom profiles and added support for external filter/servos, such a pcm4l.

2. If using peer-to-peer delay mechanism, then this would be Mean Link Delay, or <meanLinkDelay>.

1.2 Clock Recovery Techniques

For network clock synchronization, equipment clock systems predominantly rely on a linear-based clock recovery technique (e.g., a low-pass loop filter). This filters the noise on the recovered clock (i.e., from CDR) and suppresses high-frequency noise, or jitter. The ITU-T has many equipment clock recommendations for TDM, legacy or non-packet network types (e.g., ITU-T G.813 for SDH/SONET networks and ITU-T G.8262 for Ethernet networks). However, with the transition to packet-switched networks, more advanced techniques may be required as Ethernet is built based on an asynchronous physical layer clock (some Telecom networks have been transitioned to Synchronous Ethernet, per ITU-T G.8262; however, there are still many networks that are based solely on IEEE 802.3). There are typically two ways to recover a clock when sent over a packet-based network: adaptive and differential.

Per ITU-T G.8260, adaptive clock recovery is a technique that does not require the support of a network-wide synchronization signal to regenerate the timing. In this case, the timing recovery process is based on the (inter-) arrival time of the packets using timestamps. The information carried by the packets could be used to support this operation, depending on the packet delay variation (PDV) each timing packet experiences. Either two-way or one-way protocols can be used, which is why IEEE 1588 is a good protocol choice.

Differential clock recovery is another technique, typically used for Circuit Emulation Services (CES), to recover clocks based on the difference between (inter-)arrival time of the packets using a traceable clock at both the sending end and the receiving end. This ensures the recovered clock is not affected by the packet delay variation (PDV) each timing packet experiences. However, a traceable clock may not be supported [or available] at the receiving end, so adaptive clock recovery must be used. More details on differential methods can be found in section 8.2 of ITU-T G.8261.

The Renesas pcm4l software supports both adaptive and linear clock recovery techniques, with the linear clock recovery being offloaded to the Renesas network synchronization devices, such a ClockMatrix device.

1.2.1 Assisted Clock Recovery

The ITU-T has proposed an architecture that uses GNSS as the primary local time source, with PTP acting as a backup – primarily to maintain a time holdover on the loss of GNSS. This is described in G.8273.4 as Assisted Partial Time Support (APTS).

Two methods can be used to implement APTS:

1. Select the local time source as the best clock for local PTP clock (i.e., ITU-T virtual PTP port), and monitor the PTP T-GM using ITU-T PTP Monitoring (per Appendix III of G.8275.2).
2. Maintain two independent clock channels (GNSS and PTP) and correct for asymmetry on the active PTP T-GM

The latter works best for T-BC-A implementations, to maintain PTP clock traceability to the T-GM. The former can work well for either T-BC-A or T-TSC-A, but the PTP stack must support the G.8275.2 PTP Monitoring to allow the PTP T-GM's to be monitored and corrected for asymmetry prior to GNSS failure.

2. Software Architecture Overview

The Renesas software is provided under a no-fee license. The terms of that license can be found and agreed to here: pcm4l Software (Note: user login is required).

The Renesas PCM software solution covers network clock tracking, filtering, and a clock state machine. It is meant to interface an IEEE 1588 compliant PTP stack, such as Linux PTP (ptp4l). The PCM for Linux (pcm4l) software package has already been integrated to Linux, and can run on any Linux Distribution v3.0 or later.

The software runs on an external processor running the Linux OS and interfaces the clock and timestamper unit(s) (TSU) through the PTP Hardware Clock (PHC) Infrastructure. The raw PTP timestamps are provided by the IEEE 1588-2019 Slave Event Monitoring channel and either filtered by Renesas' patented PDV noise filtering algorithm or by hardware using Renesas' network synchronization PLLs, such as ClockMatrix. The PCM has a local oscillator state machine (LOSM) that supports the ITU-T PTP clock operating modes.

The PCM supports various means to stabilize the frequency and/or time of the local PTP clock. This can be by physical layer assistance (i.e., SyncE) as used by the G.8275.1 Profile (and optional for the G.8275.2 Profile) or local GNSS assistance as used by the G.8275.2 Profile for APTS (and optionally being consider with G.8275.1 Profile as well).

The Renesas software simplifies the complex process of integrating packet-based timing for network synchronization into a larger system. Separating the 1588 protocol from the clock recovery allows the user to use readily available PTP stacks for the protocol, to use the existing Linux kernel socket options and PTP clock infrastructure for time stamp and clock correction routing within the kernel, and to use different vendors (and drivers) for each of the 1588 system components.

An example of such a PTP stack is Linux PTP, or ptp4l. The ptp4l software already implements the IEEE 1588-2008 standard, including supporting some industry 1588 Profiles, and uses the Linux kernel PTP infrastructure to access hardware for retrieving local hardware (HW) timestamps and clock control. Many CPU/SoC vendors already provide kernel driver support for the HW-based PTP timestamper, and may even provide a driver for the PTP Hardware Clock (PHC) to control its PTP clock. The ptp4l software also supports a simple Proportional-Integral (PI) filter along with a servo and stage machine. However, this PI-filter was not designed to support the PDV that could be experienced on PTP timing packets when using the ITU-T G.8265.1 or G.8275.2 Profiles. A much more sophisticated adaptive algorithm would be needed to do packet selection and filtering. This is where the Renesas PCM fits in.

As of LinuxPTP v3.0, a new Slave Event Monitoring channel, per IEEE 1588-2019, was added to ptp4l to allow for an external filter/servo to take control of the PTP clock. The Renesas pcm4l software also supports the Slave Event Monitoring channel as part of its PTP Stack Adaptor, and is discussed further in the following sections.

Looking at the pcm4l directory structure in Figure 3, the core software (idtCore) directory comprises five component modules: Reference Tracker(s), Reference Tracker Supervisor, APTS Supervisor, SyncE Supervisor, and Local PTP Clock State Machine. Customers are not expected to modify the core software component modules. The PCM software communicates to the PTP stack via the PTP Stack Adaptor (ptpStack). The following sections provide more details.

The system software component (idtSystem) contains all the adaptors and adaption code. This should not be modified, as they have already been updated for Linux.

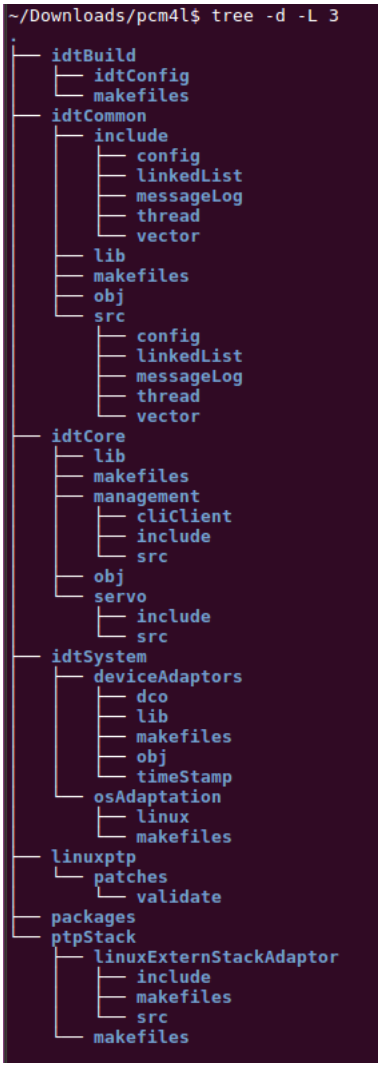


Figure 3. Directory Structure

Once a grandmaster (GM) connection has been established, the pcm4l will assign a reference tracker for that GM. The reference tracker supervisor will manage all reference trackers and maintain the local clock state machine. The following sections provide more details about each of these blocks.

2.1 Reference Trackers (RTs)

A reference tracker tracks a remote master clock by analyzing the time stamp pairs [T1, T2] from the forward link and [T3, T4] from the reverse link. It analyzes the statistics of the time stamps, estimates the frequency and time offset of the local clock compared to the remote master clock, and requests local oscillator corrections through the reference tracker supervisor. There could be multiple reference trackers running at the same time, but only one reference tracker is the active tracker and only the local oscillator correction request from this active tracker will be granted by the reference tracker supervisor.

As described earlier, there are a number of clock recovery techniques that can be used to recover a packet-based clock. The pcm4l software focuses on linear and adaptive. Linear will be used for 1588 Profiles that negate the effects of PDV noise (i.e., by using BCs/TCs throughout the network). Adaptive will be used for 1588 Profiles that require packet measurement data analysis, where packet selection (i.e., outlier detection) is an important component to the analysis. In other words, in order to reduce the input PDV noise, only a subset of the received timing packets are used by the filter. The ITU-T has done a lot of work around determining “suitable PDV” for adaptive clock recover (both for frequency and time); more information is located in Appendix I of ITU-T G.8260.

Depending on the target equipment clock recommendation, one of the four reference tracker operating modes are selected: Write Phase, Adaptive Time, Assisted Time, or Adaptive Frequency, and are described in this section. The determination of which tracker mode to use will depend on the target equipment clock recommendation.

For frequency synchronization using the ITU-T G.8265.1 Profile, the Adaptive Frequency Reference Tracker would be used. This is designed to meet ITU-T G.8263 (PEC-S-F) for the limits defined for case 3 in clause 7.2.2 of ITU-T G.8261.1 of <16ppb. Additionally, the Adaptive Time Reference Tracker would be used for the limits defined for case 1 or 2 in clause 7.2.2 of G.8263.

For phase/time synchronization using the ITU-T G.8275.1 Profile, the Write Phase Reference Tracker would be used. This will use the PTP Hardware Clock (PHC) adjust phase control and the Renesas network timing device's hardware filter to meet ITU-T G.8273.2 for telecom boundary clocks and telecom time slave clocks. Frequency assistance would come from the co-located EEC1/eSEC, which is again supported by the Renesas network timing device using our patented Combo mode, which provides the band-pass filtered noise transfer as defined in clause 7.3.2 of ITU-T G.8273.2 for Class A/B and in clause 7.3.3 Class C/D.

For phase/time synchronization using the ITU-T G.8275.2 Profile, either the Adaptive Time Reference Tracker or Assisted Time Reference Tracker would be used. These are designed to meet ITU-T G.8273.4 for telecom boundary clocks and telecom time slave clocks for use with partial timing support or assisted partial timing support from the network.

Although the reference trackers are designed to use the End-to-End (E2E) delay mechanism per IEEE 1588, they can also be used with no PTP delay mechanism (i.e., NO_MECHANISM (0xFE)). In this scenario, it is assumed the delay compensation is done outside of the 1588 protocol (such as using the ITU-T assisted partial time support (APTS)).

2.1.1 Write Phase Reference Tracker

A Write Phase Reference Tracker is used to track a master in a 1588-aware network, where the packet delay variation (PDV) is small. A Write Phase Reference Tracker uses the Renesas hardware digital loop filter. A software outlier filter is implemented to improve the reference tracking performance.

The update rate to the hardware loop filter is based on the Sync update rate (e.g., per the ITU-T G.8275.1 Profile, this is 16 packets-per-second (pps)). The hardware loop filter must be set to a bandwidth in the range of 0.05Hz and 0.1Hz, per ITU-T G.8273.2. For other profiles, the bandwidth should be set to about 1/50th of the Sync packet rate (e.g., for the IEEE 1588 Default profile at 1pps, it would be 0.02Hz, or lower).

The outlier filter is implemented in software to improve the Write Phase Reference Tracker performance. If a detected phase offset is off the mean value by the minimum outlier threshold, minOutlierThresholdNanoseconds for consecutive outlier count threshold, minOutlierCountThreshold, the detected phase offset is declared as an outlier and will not be written the DCO device. The phase offset statistics, the mean, and the standard deviation, are updated by the Write Phase Reference Tracker on every transaction.

The Write Phase Reference Tracker has four stages when it starts tracking a master clock or after it has been reset by the RTS:

- **Stage 1: Snapping Stage**. Upon entering this first tracking stage, the Write Phase Reference Tracker estimates the local clock initial time offset and requests a phase snap if the offset is bigger than a threshold. After this phase snap, the local clock will be close to the master clock. The reference tracker then switches to use the PTP Hardware Clock (PHC) adjphase API.
- **Stage 2: Converging Stage**. Starting from this stage until the last stage, the Write Phase Reference Tracker writes the phase offset to the adjphase API when a forward link or reverse link time stamp pair is received. This stage takes about one minute to allow the local clock to converge to the master clock.
- **Stage 3: Statistics Collecting Stage**. At this stage, the tracker performs: collecting PDV statistics for outlier filter; writing phase offset to adjphase API; detecting and reporting locked/un-locked states to reference tracker supervisor. This stage takes 1 minute in 16 pps or higher packet rate. It will take longer time if the packet rate is lower than 16 pps.

- **Stage 4: Tracking Stage**. At this stage, the tracker updates the PDV statistics and checks for outliers. For time stamp pairs that are not outliers, the tracker writes the phase offset to the adjphase API and updates the state (locked/unlocked) to reference tracker supervisor. The Write Phase Reference Tracker remains in this stage until it is reset by the reference tracker supervisor.

2.1.2 Adaptive Time Reference Tracker

An Adaptive Time Reference Tracker is designed to be used in a network where the PDV is large. It periodically analyzes the PDV statistics, and based on the PDV analysis, the Adaptive Time Reference Tracker determines the required time to collect the time stamps to achieve the target frequency or time estimation accuracy. After collecting enough time stamps, the Adaptive Time Reference Tracker will select a frequency estimation algorithm and/or a time estimation algorithm based on the PDV analysis. The frequency and/or a time offset are then estimated and local oscillator correction requests will be sent to the reference tracker supervisor.

Similar to the Write Phase Reference Tracker, an outlier filter is implemented in software. When the reference tracker collects the time stamps, the outlier filter analyzes the time stamps and drops the time stamps that are far away from the average ones.

The Adaptive Time Reference Tracker has three stages when it starts tracking a master clock or after it has been reset by the RTS:

- **Stage 1: Rough Time And Frequency Correction Stage**. This stage takes 10 seconds or less depending on the configuration. In this stage, the Adaptive Time reference Tracker makes a rough time estimation and sends a phase correction request to the reference tracker supervisor. If a frequency correction is configured, a rough frequency estimation is performed and a frequency correction request will be sent to the supervisor as well. Then, the Adaptive Time Reference Tracker enters the second stage.
- **Stage 2: High Precision Frequency Correction Stage**. This stage takes roughly one minute to several minutes depending on the configuration and PDV condition. The maximum time spent in this stage is determined by highPrecisionFrequencyCorrectionTimeMinutes. If the PDV is very small, this stage will take about one minute. If the PDV is not very small, the tracker will analyze the PDV and either determine how long it will need to collect time stamps or will take the configured highPrecisionFrequencyCorrectionTimeMinutes duration time to collect time stamps, depending on the lesser of the two time periods. After collecting the required time stamps, the reference tracker will perform a high-precision frequency and time estimation. Frequency and time correction requests will be sent to the reference tracker supervisor. After these corrections, the local clock should be very closely synchronized with the master clock. The tracker then transfers to the next stage.
- **Stage 3: ToD Correction Stage**. In this stage, the Adaptive Time Reference Tracker runs like an adaptive PLL. It analyzes PDV statistics, collects enough time stamps, makes a ToD offset estimation, and requests a frequency correction. The reference tracker will stay in this stage until it is reset by the reference tracker supervisor, or a large frequency or ToD offset is detected. If one of these events happens, the Adaptive Time Reference Tracker will signal the reference tracker supervisor that the lock state is lost and will then transfer to the Stage 2 - High Precision Frequency Correction Stage.

2.1.3 Assisted Time Reference Tracker

The Assisted Time Reference Tracker is designed to be used in a network where PTP will be a backup to a local time reference, such as GNSS. It performs all the functions of the Adaptive Time Reference Tracker in addition to calibrating for asymmetries using information from the local time reference (via the APTS Supervisor).The frequency and/or a time offset are then estimated and local oscillator correction requests will be sent to the reference tracker supervisor.

The Assisted Time Reference Tracker has four stages when it starts tracking a master clock or after it has been reset by the APTS Supervisor:

- **Stage 1: Idle Stage**. In this stage, the Assisted Time reference Tracker is waiting for a PTP clock source. If the PTP source is available prior to a local time reference being available, phase and frequency corrections may

occur in this stage for initial frequency and phase snapping (equivalent to stages 1 and 2 for Adaptive Time Reference Tracker).

- **Stage 2: PTP Out of Specification Stage**. This stage is entered when there is only a PTP clock source (no local time reference). Any previous asymmetry compensation will be disabled and the Assisted Time tracker will provide phase and frequency corrections (equivalent to stage 3 for Adaptive Time Reference Tracker). The reference tracker will stay in this stage until it is reset by the reference tracker supervisor, or a large frequency or ToD offset is detected. If one of these events happens, the Assisted Time Reference Tracker will signal the reference tracker supervisor that the lock state is lost and will then transfer to the “Stage 1: Idle Stage.”
- **Stage 3: PTP Calibration Stage**. In this stage, the Assisted Time Reference Tracker is in calibration mode, and PTP corrections are disabled. The APTS supervisor is locked to the local time reference source and provides phase and frequency corrections. A calibration engine collects information from the APTS supervisor to remove any asymmetry to the PTP clock source.
- **Stage 4: PTP Assisted Holdover Stage**. In this stage, the Assisted Time Reference Tracker runs like an adaptive PLL with asymmetry compensation enabled. It analyzes PDV statistics, compensates for asymmetry, collects enough time stamps, makes a ToD offset estimation, and requests a frequency correction. The reference tracker will stay in this stage until:
  - The local time reference is re-qualified and locked. If this event happens, the Assisted Time Reference Tracker will then return to the “Stage 3: PTP Calibration Stage” once lock is achieved.
  - The calibration timer expires, or a reroute is detected. If one of these events happens, the Assisted Time Reference Tracker will then transfer to “Stage 2: PTP Out of Specification Stage.”
  - It is reset by reference tracker supervisor, or a large frequency or ToD offset is detected. If one of these events happens, the Assisted Time Reference Tracker will signal the reference tracker supervisor that the lock state is lost and will then transfer to “Stage 1 - Idle Stage.”

2.1.4 Adaptive Frequency Reference Tracker

The Adaptive Frequency Reference Tracker is designed to be used in networks where the PDV is large and the main goal is to synchronize the frequency between the master and the slave. Similar to the Adaptive Time Reference Tracker, it periodically analyzes the PDV statistics, and based on the PDV analysis results, the tracker calculates the required time to collect the timestamps in order to achieve the target frequency synchronizing accuracy. After collecting the timestamps, the Adaptive Frequency Reference Tracker selects a frequency estimation algorithm based on the PDV analysis. The frequency offset is then estimated and the local oscillator correction requests will be sent to the reference tracker supervisor to apply the frequency corrections.

The same outlier filter as Adaptive Time Reference Tracker is used.

The Adaptive Frequency Reference Tracker has three stages when it starts tracking a master clock or after it has been reset by the RTS:

- **Stage 1: Rough Time And Frequency Correction Stage**. This stage takes 10 seconds or less depending on the configuration setting. In this stage, the Adaptive Frequency reference Tracker makes a rough time estimation and sends a phase correction request to the reference tracker supervisor. If a frequency correction is configured, a rough frequency estimation is performed and a frequency correction request will be sent to the supervisor as well. Then, the Adaptive Frequency Reference Tracker enters Stage 2.
- **Stage 2: High Precision Frequency Correction Stage**. This stage takes roughly one minute to several minutes to do a frequency correction, depending on the PDV statistics. The maximum time spent in this stage is determined by highPrecisionFrequencyCorrectionTimeMinutes. The tracker analyzes the PDV and determines how long it needs to collect the time stamps. After collecting the required time stamps, the reference tracker will perform a high precision frequency estimation. A frequency correction request will be sent to the reference tracker supervisor. After this correction, the local clock should be synchronized in frequency with the master clock. The Adaptive Frequency Reference Tracker then transfers to Stage 3.
- **Stage 3: Frequency Correction Stage**. In this stage, the Adaptive Frequency Reference Tracker continues periodically analyzing PDV statistics, collects enough time stamps, makes a frequency offset estimation, and requests a frequency correction to the reference tracker supervisor. The reference tracker will stay in this stage

until it is reset by the reference tracker supervisor, or a large frequency offset is detected. If one of these events occurs, the Adaptive Frequency Reference Tracker will signal the reference tracker supervisor lost lock state and will then transfer to the Stage 2 - High Precision Frequency Correction Stage.

2.1.5 Programmable Loop Filter

Each ReferenceTacker has a programmable Proportional + Integrator (PI) loop filter that is used with the Adaptive Time and Adaptive Frequency modes. It provides access to the Minimum Response Time, Maximum FFO Correction, Integral Branch Gain, and Bandwidth Scalar. Generally, these should be left at their defaults because they are optimized for G.8261 and G.8271.2 networks when using the Adaptive Reference Tracker(s).

- **Minimum Response Time Seconds**, dcoLoopFilter.minResponseTimeSeconds, is the target duration (in seconds) for the DCO filter to bring the Time of Day offset back to zero, if the PDV conditions permit.
- **Maximum FFO Correction**, dcoLoopFilter.maxFfoCorrection, is the absolute value of the maximum allowed FFO (fractional frequency offset) correction that the proportional branch of the loop filter can apply to the DCO. It can also be equated to a phase slope limit (PSL) in ns/s.
- **Integral Branch Gain**, dcoLoopFilter.integralBranchGain, is the gain of the integral branch of the loop filter relative to the proportional branch.
- **Bandwidth Scalar**, dcoLoopFilter.bandwidthScalar, is the DCO loop filter bandwidth scalar value.

For more information on usage, please contact Renesas.

2.2 Reference Tracker Supervisor (RTS)

The RTS manages the switching between RTs. This includes concepts like hitless switching, revertive and non-revertive switching, and phase slope limiting. The use of multiple RTs is generally for Profiles that support multiple GM instances to run in parallel, such as ITU-T G.8265.1 Profile, or that support PTP monitoring, such as ITU-T G.8275.2 Profile.

When using with a PTP stack that supports multiple OC instances, a Master Selection Algorithm (MSA) is available to take advantage of additional G.8265 features, such as Wait-to-Restore and Master Lock-out.

2.2.1 Hitless Reference Switching

The hitless reference switching feature is configured in the JSON configuration file. Setting the value of “ptpPacketReferenceSwitch.referenceSwitchType” to “Hitless” will enable this feature. When a reference switch between RTs is triggered, the time offset of the selected/target master against the slave is measured. The servo then automatically compensates for this offset, so the phase transient at the clock outputs can be minimized when servo switches to another RT.

In order for hitless reference switching to work, the following conditions must be met:

- The slave time offset to the target master must be within the time lock threshold, timeLockThresholdNanoseconds (i.e., in the time locked state).
- The slave frequency offset to target master must be within frequency lock threshold, frequencyLockThresholdPpb (i.e., in the frequency locked state).

When all three conditions are satisfied, the servo will estimate the offset to the target master at the time of the reference switch, and this offset will be compensated at the clock output. The compensated offset, if any, at the original RT will be automatically cleared when the servo switches away from it.

An option to clear any absorbed phase offset can be triggered by calling the management API mngApi_HitlessSwitchTieReset. When this API function is called, the servo will remove any phase offset compensation and start to track the master at zero phase offset.

2.2.2 Phase Slope Limit and Instant Reference Switching

The phase slope limiting feature can be configured to limit the rate of output phase movement when the servo switches from one RT to another. The following parameters in “ptpPacketReferenceSwitch” configuration are used for phase slope limiting:

- “referenceSwitchType” – Set type to “PhaseSlopeLimit” to select reference switch type as phase slope limited switch.
- “phaseSlopeLimit” – The limit of output phase movement rate in the unit of nanosecond per second.

If “referenceSwitchType” is set to “PhaseSlopLimit” and the slave is time locked to the target master, the value of “phaseSlopeLimit” will be applied when there is a switch between RTs. If “referenceSwitchType” is set to “PhaseSlopeLimit” and the slave is frequency locked to the target master, the frequency lock threshold will be applied when the reference switch occurs.

If “referenceSwitchType” is set to “Instant”, no phase slope limit will be applied if the slave ToD offset to the target master is larger than 10 microseconds, or 1 ppm per second phase slope limit will be applied if the offset is less than 10 microseconds.

2.2.3 Reference Switching Thresholds

These thresholds take effect only if the reference switch type is hitless or phase slope limited. If the reference switch type is hitless and the absolute value of the ToD offset is within the thresholds, then apply a hitless switch; otherwise, apply a phase slope limit. If the reference switch type is phase slope limited and the absolute value of the ToD offset is within the thresholds, then apply PSL; otherwise, apply a hitless switch.

The reference switch thresholds are configured by the JSON parameters lowerTodThresholdNanoseconds and upperTodThresholdNanoseconds.

2.2.4 Revertive/Non-revertive Switching (G.8265)

Revertive switching is the default behavior for switching between masters. When the best master, as selected by the telecom master selection algorithm, is registered or restored due to loss of signal, etc, the servo will switch from the master it is currently tracking to the new best master.

When revertive switching is disabled, master switching is non-revertive (i.e., servo will only switch to different master if the current master is disqualified due to loss of signal, etc.). One exception is when the new master that comes up is a better master than the current master the servo is tracking and the servo has not achieved frequency lock with the current master yet; in this case, the servo will switch to the new master.

2.2.5 Wait-to-Restore (G.8265)

When configured, wait-to-restore ensures the previously disqualified RT (due to signal failure, etc.) is only again considered for the best master selection when it is fault-free for the time period specified by the wait to restore time.

The wait to restore time can be configured globally for all masters using the configuration parameter “masterWaitToRestoreTimeoutValue”, or individually using the management API function mngApi_SetWaitToRestoreTimeout().

The following management API functions can be used to monitor/query wait to restore time for each individual master:

- mngApi_GetWaitToRestoreTimeout – Returns the current value of wait to restore time
- mngApi_IsWaitToRestoreTimerRunning – Checks whether wait to restore timer is running
- mngApi_ClearWaitToRestoreTimer – Clears the running wait to restore timer

The unit for wait to restore time is in seconds to allow for configuration flexibility.

2.2.6 Master Lock-out (G.8265)

The user has the option to lock out certain masters. By locking out the master, it will be excluded from the master selection consideration. The following management APIs are used for master lock-out:

- **mngApi_SetG82651MasterLockoutEnable** – Enable/disable master lock out
- **mngApi_GetG82651MasterLockoutEnable** – Query whether master is locked out

2.3 SyncE Supervisor

The SyncE Supervisor is a state machine implemented in the device adaptor that monitors the SyncE (physical layer) DPLL channel and periodically sends frequency information to the reference trackers. The SyncE Supervisor provides a high-level indication of the operational status of the SyncE clock. There are three states in the SyncE Supervisor:

- **Unqualified** – Nothing is asserted about the quality of the SyncE clock in this state. The SyncE Supervisor starts in this state if physicalPllClockCategory is not 5 (DNU).
- **SyncE Locked** – The SyncE DPLL channel is locked and qualified. The PTP output clocks are syntonized to the frequency reference if the clock category is greater than or equal to physicalPllClockCategoryThreshold.
- **SyncE Holdover** – The SyncE Supervisor has lost its frequency reference with no other qualified references available (SyncE clock is in Holdover). After entering holdover, the accuracy of the PTP outputs depends on the local XO.

2.4 APTS Supervisor

The APTS Supervisor is a state machine implemented in the device adaptor that monitors the GNSS DPLL channel and periodically sends time and frequency information to the reference trackers. The APTS Supervisor provides a high-level indication of the operational status of the GNSS clock. There are four states in the APTS Supervisor:

- **Unqualified** – Nothing is asserted about the quality of the GNSS clock in this state. The APTS Supervisor starts in this state.
- **Wait First TOD Read** – The GNSS DPLL channel is locked and qualified. In this state it will attempt to get a PTP ToD from the PTP channel. Once the initial ToD is retrieved, it will enter the Locked state.
- **GNSS Locked** – The GNSS DPLL channel is locked and qualified. The GNSS and PTP output clocks are phase/time synchronized to the local time reference.
- **GNSS Holdover** – The APTS Supervisor has lost its local time reference with no other qualified references available and it uses the stored data acquired in Locked mode to control its output clock. After entering holdover, the accuracy of the GNSS outputs depends on its local clock (i.e., from physical layer assistance [SyncE traceable to PRC] or local XO). If the holdover timer expires, the APTS Supervisor will return to the Unqualified state.

2.5 Local Oscillator State Machine (LOSM)

The LOSM is a core component of the PCM and is the main part of the RTS. The states may be interpreted as signaling the quality of the Local PTP Clock in relation to its active input reference, if such a reference exists. Input references may take the form of a remote GM or BC, electrical references such as SyncE, or local GNSS reference.

The LOSM follows the PTP clock modes as described in ITU-T G.8275.1 Appendix V and G.8275.2 Appendix IV. The LOSM provides a high-level indication of the operational status of the entire clock. This can then be used to map to PTP port states as defined in IEEE 1588 and provide a mapping to content of the Announce message fields. The following table provides a mapping of the LOSM state and the ITU-T clock mode.

Table 1. ITU-T Clock Mode Mapping

ITU-T Clock Mode	pcm4l LOSM Clock State	ptp4l Clock Servo State
Free-Run	Unqualified	S0
Acquiring	Lock Acquisition	S1
Locked	Frequency Locked / Time Locked	S2 (/ S3)
Holdover-In-Specification	Holdover-In-Specification	N/A
Holdover-Out-Of-Specification	Holdover-Out-Of-Specification	N/A

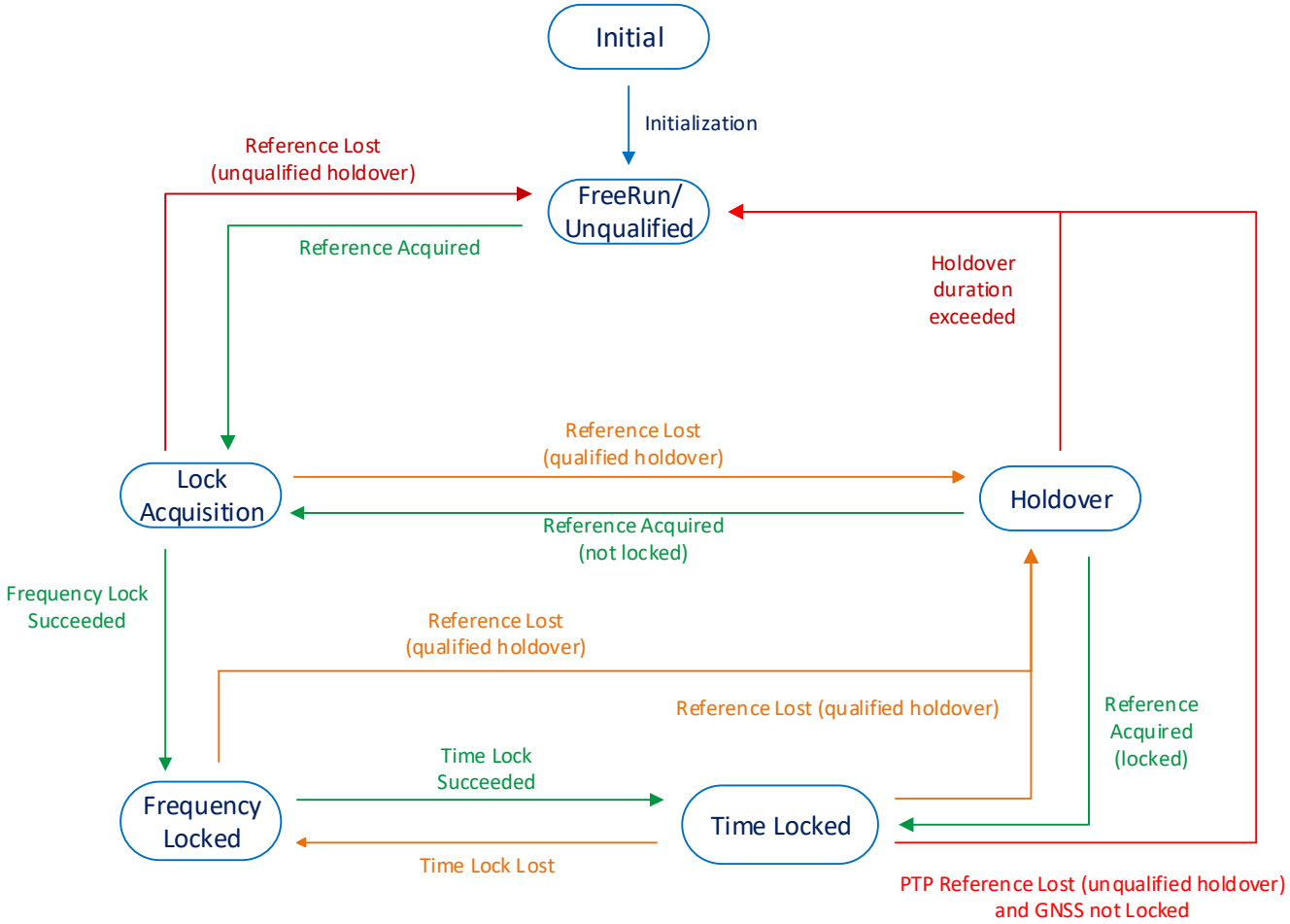


Figure 4. State Machine Diagram

2.5.1 Operational States

There are six states in the LOSM:

- **Unqualified** – Nothing is asserted about the quality of the Local PTP Clock in this state. The LOSM starts in this state and enters on expiration of unqualified timer, unqualifiedTimeoutSeconds, when in the Holdover state.
- **Lock Acquisition** – At least one qualified reference is present prior to reaching the frequency locked state.
- **Frequency Locked** – The output clock is syntonized to an input reference within the user specified threshold, frequencyLockThresholdPpb. If using physical layer assistance and the SyncE Supervisor is in the locked stage, this could reflect the state of the PTP clock on the loss of all PTP references.

- **Time Locked** – The output clock is phase/time synchronized to an input reference within the user specified threshold, timeLockThresholdNanoseconds.
- **Holdover In-Specification** – The RTS has lost its reference with no other qualified references available and uses the stored data acquired in Locked mode to control its output clock. After entering holdover-in-spec, the accuracy of its output depends on the Local PTP Clock (i.e., from physical layer assistance (SyncE traceable to PRC) or local XO).
  A qualification timer is used to determine if the stored data is valid (JSON: holdoverQualificationSeconds). If holdover data is not valid, this state is bypassed and the LOSM will immediately enter the holdover out-of-specification state.
- **Holdover Out-Of-Specification** – The Holdover timer, holdoverTimeoutSeconds, has expired. After entering holdover-out-of-spec, the accuracy of the PTP output depends on the Local PTP Clock and its time traceability is lost.

2.5.2 Lock Acquisition Qualification Timeout

Once a best master is selected, the LOSM will enter the lock acquisition state, start the lock acquisition qualification timer, and attempt to lock to the master. If the lock acquisition qualification timer expires before frequency lock can be achieved then this master will be considered unusable and will be locked out.

The master can be unlocked by calling mngApi_SetG82651MasterLockoutEnable with the corresponding instance number and lockoutEnable set to E_osFalse.

The lock acquisition timer is configured by the JSON parameter qualificationTimeoutSeconds.

3. Start-up System Configuration

This section describes the format and method of configuring PCM application start-up. This includes details on the configuration file format, configuration file verification, configuration file processing, and information about the configuration parameters themselves.

3.1 The JavaScript Object Notation (JSON)

Renesas uses the JavaScript Object Notation (JSON) text notation to configure the pcm4l. JSON is a text based, lightweight data interchange format. The JSON configuration file contains a set of configurable parameters that provides the user with the flexibility to configure the operation of their 1588 implementation.

The JSON Configuration file and associated scripts, files and tools can be found in the idtBuild\idtConfig\ directory.

3.1.1 JSON Configuration File

The JSON configuration file is provided in ASCII formatted text, and can be easily edited using any text editor. A JSON formatted file is recognizable by the .json file type. The JSON configuration file conventions are as follows:

- The JSON array starts with '[' (left bracket) and ends with ']' (right bracket).
- A JSON object consists of an unordered set of *name/value* pairs. The JSON object starts with '{' (left brace) and ends with '}' (right brace).
- Each *name* is separated from its value by a ':' (colon) and the *name/value* pairs are separated by a ',' (comma).
- The *name* is represented by a "string" while the value can be represented by a string in double quotes, a number, another JSON object, a JSON array or a true, false or null keyword.
- The JSON object *name* "_description_" is used for comments and the *value* is not used.

3.2 General Configuration Parameters

The following configuration parameters can be specified in the JSON Configuration file. Default values are populated for any configuration item not present in the JSON configuration file.

Table 2. Top-Level Configuration Parameters

JSON Identifier	Default	Description
versionId	“4.0”	Identifies the version of software that the configuration file is compatible with. Every configuration file must indicate the version of software the configuration file corresponds to. This field should not be changed unless migrating a configuration file from one version of software to another.
testModeEnable	0	Test mode. When test mode is enabled, data validation errors are ignored. 0 = Disabled, errors flagged and application exits. 1 = Enabled, errors flagged and application continues.
referenceTrackerType	“AdaptiveTime”	Selects the type of reference tracker to be used by the application (see Reference Trackers (RTs)). ▪ “AdaptiveFrequency”: See Adaptive Frequency Reference Tracker. ▪ “AdaptiveTime”: See Adaptive Time Reference Tracker. ▪ “AdaptiveTimeAssist”. Assisted Time Reference Tracker. ▪ “WritePhase”: See Write Phase Reference Tracker
mgmtIpAddress	“127.0.0.1”	IPv4 address used by the application management interface.
displayStackStatusTimeoutSeconds	15	A periodic timer is started when the LOSM enters unqualified state and is stopped when leaving unqualified state. If the timer expires, the following information for each enabled instance is displayed: ▪ Domain ▪ Port state ▪ Address mode ▪ Unicast negotiation enable ▪ Unicast table ▪ Acceptable master enable ▪ Acceptable master table Valid values: A non-negative 32-bit integer (seconds).
remoteUdsAddress	“/var/run/ptp4l”	Identifies the PTP stack UDS address.
localUdsAddress	“/var/run/pcm4l”	Identifies the local UDS address.
phc4lConfig	N/A	See Table 3.
deviceConfig	N/A	See Table 5.
profileConfig	N/A	See Table 6.
loggerConfig	N/A	See Table 7.
instanceConfig	N/A	See Table 11.

Table 3. PHC Configuration Parameters

JSON Identifier	Default	Description
charDevice	“/dev/rsmu0”	Character device assigned to Renesas misc character device driver for timing devices.
dcoDevice	“/dev/ptp1”	Identifies the PHC device node for the main PTP Hardware Clock for clock control (i.e., Renesas timing device).
tsDevice	“/dev/ptp0”	Identifies the PHC device node for the PTP timestamper (OC) or timestampers (BC).

Table 3. PHC Configuration Parameters

JSON Identifier	Default	Description
tsDevicePinIndex	-1	Timestamp device pin index is used to trigger one PPS event.
phaseSnapDelaySecond	2	This indicates the time to wait for a time adjustment to be reflected at the PHC.

Table 4. Syncd Configuration Parameters

JSON Identifier	Default	Description
syncdPeriodSecond	1	This is the interval to run the timestamper ToD synchronization.

Table 5. Device Configuration Parameters

JSON Identifier	Default	Description
oscillatorType	“MiniOcxo”	The oscillator type specifies the type of oscillator being used for the 1588 application. ▪ “Tcxo”: Temperature compensated XO ▪ “MiniOcxo”: Hybrid oven/temperature compensated XO that meets ITU-G.8263 holdover stability ▪ “Ocxo”: Oven controlled XO that meets G.812 Type I/II/III ▪ “SyncE”: Co-located physical layer assistance (for G.8273.2 clocks) ▪ Deprecated: Use physicalPllClockCategory 1-4 to manage the physical layer assistance
dpll1588Instance	0	Sets DPLL index used for the 1588 application.
qualificationTimeoutSeconds	900	Timeout value for the qualification timer. Valid values: {0 - 31,536,000} in seconds
tsDeviceAlignmentDisable	0	Specifies the use of internal (syncd) or external time stamper alignment, such as ts2phc.
assistedPartialTimeSupport	N/A	▪ “gnssDpllIndex” {0, 31}: Sets DPLL index used to monitor GNSS 1 PPS. ▪ “gnss1ppsInputIndex” {0, 31}: Sets INPUT pin index used for GNSS 1 PPS ▪ “gnssCableDelayNanoseconds” {0 - 1000}: GNSS 1 PPS cable delay compensation ▪ “gnssHoldoverWaitSeconds” {0 - 10000}: This value is used to protect against frequent toggling between GNSS Locked state and GNSS Holdover state when GNSS 1 PPS is sporadically not available for brief periods of time. ▪ “assistedHoldoverTimeoutSeconds” {0 - 10000}: The duration in seconds to stay in PtpAssistedHoldoverStage before transitioning to PtpOutOfSpecStage when GNSS 1 PPS becomes unavailable.

Table 5. Device Configuration Parameters (Cont.)

JSON Identifier	Default	Description
holdover	N/A	This group includes the clock holdover settings. Based on these values, the servo decides how to transition the DCO on loss of all PTP references. ▪ holdoverType: {Software, Hardware, HardwareEnhanced} Select to use Software or Hardware control for holdover ▪ holdoverLossPhysicalOosEnable: 0 = Disable, 1 = Enable. ▪ holdoverTimeoutSeconds: {int} How long in seconds to remain “in-spec” and transient to “out-of-spec” ▪ unqualifiedTimeoutSeconds: {int} How long in seconds to remain “out-of-spec” and return to unqualified state ▪ outOfSpecUserDefinedFrequencyOffsetEnable: Enable/disable the use of a user defined FFO when entering out-of-spec. ▪ outOfSpecUserDefinedFrequencyOffsetPpb: {int} Value of FFO in ppb to set the DCO

writePhaseConfig	N/A	“writePhaseTimeoutConfig”:
		{
		“enable”: 0,
		“useConfiguredTimeoutValue”: 0,
		“timeoutMilliseconds”: 133
		}
		▪ enable {0, 1}: Enables the write phase timeout mechanism. Intended for devices that are configured for “Hardware” holdoverType and need a SW timer to write 0 to phase offset after a period of write phase offset inactivity. The default timeout period is 133 ms. Contact Renesas support for details on changing default timeout period.
		▪ useConfiguredTimeoutValue {0, 1}: Use JSON configure values for write phase timeout. The default timeout period is 133 ms.
		▪ timeoutMilliseconds {1 - 65535}: JSON value for write phase timeout.

Table 6. Profile Configuration Parameters

JSON Identifier	Default	Description
physicalPllClockCategory	5	Specifies the ITU-T G.781 Quality Level (QL) of the physical clock. This value is used to determine the suitability of the physical clock as a valid reference and can be used to determine the IEEE 1588 frequency traceable flag. Used in conjunction with physicalPllClockCategoryThreshold to determine suitability of the physical PLL clock as reference. 1 = Category 1 frequency source (QL-PRC / QL-PRS) 2 = Category 2 frequency source (QL-SSU-A / QL-ST2) 3 = Category 3 frequency source (QL-SSU-B / QL-ST3E) 4 = Category 4 frequency source (QL-EEC1 / QL-EEC2) 5 = Do Not Use (i.e. not to be used as reference).
physicalPllClockCategoryThreshold	1	Specifies the clock category threshold to trigger the restoration of the physical clock as a valid reference.
physicalPllInstance	1	This is used to enable/disable combo mode based on the physicalPllClockCategory and physicalPllClockCategoryThreshold that is changed during run-time externally by management API. Only valid if physicalPllClockCategory is not 5 (DNU).

Table 6. Profile Configuration Parameters (Cont.)

JSON Identifier	Default	Description
physicalPllWaitToRestoreTimeoutValue	10	When the traceability is restored (i.e., when the QL moves from a lower level to a higher level and the QL is equal to or higher than a specified threshold), a wait-to-restore mechanism is implemented before restoring the physical layer connection. This parameter sets the restoration delay. Valid values: {0 - 100} In seconds. 0 means no delay.
gnssPllClockCategory	2	Specifies the validity of the GNSS physical clock. When the GNSS clock is stable and ready to be used as a reference, set the gnssPllClockCategory to 1. 1 = Category - Active 2 = Category - Void (i.e. not to be used)
gnssPllClockCategoryThreshold	1	GNSS is unqualified when “gnssPllClockCategory” > “gnssPllClockCategoryThreshold”. 1 = Category - Active
gnssPllWaitToRestoreTimeoutValue	0	When GNSS is in holdover and becomes locked again, this value delays transition back to GNSS locked state. 0 = Restore immediately 1 = Wait 1 second 2 = Wait 2 seconds …

Table 7. Logger Configuration Parameters

JSON Identifier	Default	Description
stdoutLog	N/A	See Table 8.
logFileN	N/A	“N” represents a log instance (see Table 9).
syslog	N/A	See Table 10.

Table 8. STDOUT Log Configuration Parameters

JSON Identifier	Default	Description
selectionMask	N/A	16-bit string bitfield to indicate type of messages to display. Starting from the right, Bit#0: bit#0 = Sync Error bit#1 = Sync Warning bit#2 = Sync Analysis bit#3 = Error bit#4 = Warning bit#5 = Debug bit#6 = Reserved bit#7 = Timestamp

Table 9. Log File Configuration Parameters

JSON Identifier	Default	Description
enable	1	This parameter is used to enable/disable this log instance. 0 = File logging disabled 1 = File logging enabled
fileName	N/A	The name of the log file as a string of characters. Valid values: Maximum of 179 characters file name.
selectionMask	N/A	Mask is a 16-bit bitfield representing the message severity levels; same as for STDOUT.
purge	1	This boolean indicates whether the log history should be deleted or not. 0 = Append log messages to existing file. 1 = Discard the log history before logging
maxFileSize	1000000	Maximum File Size is the maximum size in number of bytes for this log file. Valid values: Number of bytes before log file is archived. (Max. 16 gigabytes).
archives	2	Archives represents the maximum number of file archives used by this log instance. Valid values: {0 - 255}

Table 10. Sys Log File Configuration Parameters

JSON Identifier	Default	Description
enable	1	This parameter is used to enable/disable forwarding log messages to an UDP socket. 0 = Sys log disabled 1 = Sys log enabled
selectionMask	N/A	Mask is a 16-bit bitfield representing the message severity levels; same as for STDOUT (see Table 8).
ipAddress	“10.64.10.45”	Destination IPv4 address.
udpPort	514	UDP port number. Valid values: {0 - 65535}

Table 11. Reference Tracker Instance Configuration Parameters

JSON Identifier	Default	Description
instanceEnable	1	This parameter is used to enable/disable this particular reference tracker instance.
trackerConfig	N/A	See Table 13.
portConfig	N/A	See Table 12.

Table 12. Port Configuration Parameters

JSON Identifier	Default	Description
pathAsymmetry	0	Seed a local pathAsymmetry on a per-PTP port basis. This will be added to <delayAsymmetry> as calculated per IEEE 1588-2008. The value is scaled nanoseconds = nanoseconds * 2^{16}. For example, 500 nanoseconds = 500 * 65536 = 32768000.

Table 13. Reference Tracker Configuration Parameters

JSON Identifier	Default	Description
floorDelayEstimateSeconds	-1.0	The Floor Delay Estimate is a network latency estimate for an arbitrary delay set/used by the servo. By tuning this configuration parameter for the 1588 implementation, faster lock times can be achieved. Valid values: {-1.0 - 1.0} second; -1.0 means to have the tracker automatically determine.
willCorrectFrequencyAtFirstSnap	0	Provides the option to use the initial FFO estimation to correct local oscillator FFO. 0 = Do not correct 1 = Correct; recommended (as of v4.0.1) for Write Phase mode to fast lock to the GM
minExpProportionForMinTracking	0.12	Sets the minimal proportion of the exponentially distributed component required for PDV minimal value tracking. Valid values: {0.00 - 1.00}
stationarityBounds	N/A	This group includes the stationarity score bounds. Based on these values, the servo decides whether the PDV is stationary or not. measure1Lower {0.10 - 1.00} Stationarity Score measure1Upper {1.0 - 10.0} Stationarity Score measure2Upper {0 - 500} Skew Limit
maxNumberOfPhaseSnap	1	Sets the maximum number of phase snaps after initial acquisition is finished. Valid values: {0 - 5} Phase snaps
doubleDcoThresholdNanoseconds	600	The threshold in nanoseconds to start an open-loop pull-in for phase correction. Valid values: {200 - 1000000} Nanoseconds
phaseSnapThresholdSeconds	0.00001	The threshold in seconds for snap phase correction. This will depend on the accuracy of the timestamper's adjphase for relative time adjustment. Valid values: {0.000003 - 0.000100} Seconds
rerouteFloorDelayThresholdSeconds	0.000002	This threshold is used to detect floor delay changes in the network. Valid values: A positive value expressed in seconds.
rerouteAbnormalTodChangePpb	20	This threshold in nanoseconds per second is used to detect fast variations of ToD. Valid values: A positive value expressed in nanoseconds per second.
highPrecisionFrequencyCorrectionTimeMinutes	6	This parameter specifies the duration in minutes needed for high precision frequency estimation. Valid values: {0 - 200} minutes
pdvThreshold	N/A	This group includes parameters that are used to determine PTSF unusable based on the log variance of the PDV in downlink and uplink direction. If the log variance exceeds the PDV threshold, PTSF unusable is set. downlink {-100 - 0} Log Variance uplink {-100 - 0} Log Variance
pdvThresholdExceededHysteresis	N/A	Once the PDV threshold is exceeded and PTSF is declared unusable, the PDV log variance must cross below the PDV Threshold minus this group's value. downlink {0 - 10} Log Variance uplink {0 - 10} Log Variance
snapTransitionTimestamps	4	Number of time stamps that are not usable after a phase (ToD) snap. Valid values: {0 - 100} Time Stamps

Table 13. Reference Tracker Configuration Parameters (Cont.)

JSON Identifier	Default	Description
dcoLoopFilter	N/A	This group includes the configuration parameters for the DCO (digital controlled oscillator) Loop Filter: Minimum Response Time Seconds is the target duration (in seconds) for the DCO filter to bring the Time of Day offset back to zero, if the PDV conditions permit. Maximum FFO Correction is the absolute value of the maximum allowed FFO (fractional frequency offset) correction that the proportional branch of the loop filter can apply to the DCO. Integral Branch Gain is the gain of the integral branch of the PL loop filter relative to the proportional branch. Bandwidth Scalar is the DCO loop filter bandwidth scalar value. minResponseTimeSeconds {20 - 200} seconds maxFfoCorrection {1 - 10} parts-per-billion integralBranchGain {0.0 - 1.0} Gain bandwidthScalar {0.2 - 2.0} Bandwidth Scalar Value
desiredPrecisionSeconds	0.0000001	Sets the target precision of the ToD offset estimation. Valid values: {0.000000020 - 0.000000200} Precision in seconds
frequencyLockThresholdPpb	10.0	This is the absolute value of the frequency offset threshold below which the Frequency Lock is achieved. It is expressed in parts-per-billion.
timeLockThresholdNanoseconds	1000	This is the absolute value of the ToD offset threshold below which the Time Lock is achieved. It is expressed in nanoseconds.
ffoSlopeLimitPpbPerSecond	-1.0	Configures whether a frequency slope limit is applied to the desired FFO correction. -1 = No limit {X}: Apply X ppb/s limit on any FFO correction
writePhasePostSnapDelaySeconds	0	This can be used to avoid snap collisions after a time adjustment has been made to the clock. This allows for the timestamper(s) to settle before using ingress/egress timestamps.
minOutlierThresholdNanoseconds	100	Any offset from master calculation determined to be greater than this in nanoseconds will be identified as an outlier.
minOutlierCountThreshold	16	Number of sequential outliers before discarding phase offset adjustments.

4. PTP Clock Manager Management

4.1 Management APIs

The Renesas servo management API reference describes all C functions, types and definitions of the application programming interface to access and configure the Renesas servo software. Complete details can be found in “Servo Management API” of the *PCM4L Reference Manual*.

However, pcm4l is meant to be a stand-alone, user-space application. For this reason, it is recommend to refer to the provided Management CLI Utility. The management interface's server IP address is configurable through JSON parameter: mgmtIpAddress.

4.2 Management CLI Utility

The Renesas PTP Clock Manager software includes a command line interface (CLI) utility for management. It provides an interactive interface for user to enter management command code and displays the response from server.

The CLI executable is compiled as “cliClient”. It has the following syntax:

```
cliClient <server IP address> <server port number>
```

After CLI successfully connects to the server, type “help” from the command prompt to display all the supported management codes. To see help text for each individual command, type “help <command code>”:

```
>help 05
Get the Holdover Type: Software or Hardware
>help 06
Set the Holdover Type: Software = 0, else Hardware
```

To run a command, enter the command code and follow the prompt message. The following example shows “Set Holdover Type” (command code 05) and “Get Holdover Type” (command code 04) command:

```
>06
Set Holdover Type:
Holdover Type? (Software = 0, 1 = Hardware): 0
Sending Command 6, E_cmnMngApi_SetHoldoverType
Waiting for response from server
Server responded with command 6, E_cmnMngApi_SetHoldoverType
>05
Get Holdover Type:
Sending Command 5, E_cmnMngApi_GetHoldoverType
Waiting for response from server
Server responded with command 5, E_cmnMngApi_GetHoldoverType
The Holdover Type is Software
>
```

5. ITU-T Equipment Clock Compliances

The following performance results use ptp4l as the PTP protocol stack along with pcm4l. All tests were completed with Paragon-NEO, unless otherwise stated.

5.1 Noise Generation

Table 14. Noise Generation[1]

Symbol	Parameter		Test Conditions	Minimum	Typical	Maximum	Unit
max|TE_L|	Maximum absolute time error low-pass filtered[2]		WritePhase mode (with physical layer assistance per G.8273.2)	-	3.7	5[3]	ns
			AdaptiveTime mode (no physical layer assistance)[4]	-	47	-[5]	ns
			AssistedTime mode (with local time reference)[6]	-	8	-[5]	ns
dTE_L	Dynamic time error low-pass filtered noise generation[2]	MTIE	WritePhase mode (with physical layer assistance per G.8273.2)	-	0.24	3[7]	ns_{pp}
		TDEV		-	0.04	1[7]	ns
		Peak-to-Peak	AdaptiveTime mode (no physical layer assistance)[4]	-	92	200	ns
		Peak-to-Peak	AssistedTime mode (with local time reference)[6]	-	6.3	50	ns

1. Measured on Xilinx ZCU111 + ClockMatrix 8A34001E FMC running pcm4l v4.1 and ptp4l v3.0.
2. Measured under constant temperature (within ±1 K) and using a first-order low-pass measurement filter with a bandwidth of 0.1Hz for a minimum of 10,000 seconds.
3. ITU-T G.8273.2 Class D.
4. Using an ITU-T G.8263 compliant oscillator. See AN-807 Recommended Crystal Oscillators for Network Synchronization
5. Still for further study at ITU-T.
6. Locked to local time reference source (GNSS).
7. Provisional G.8273.2 Class D.

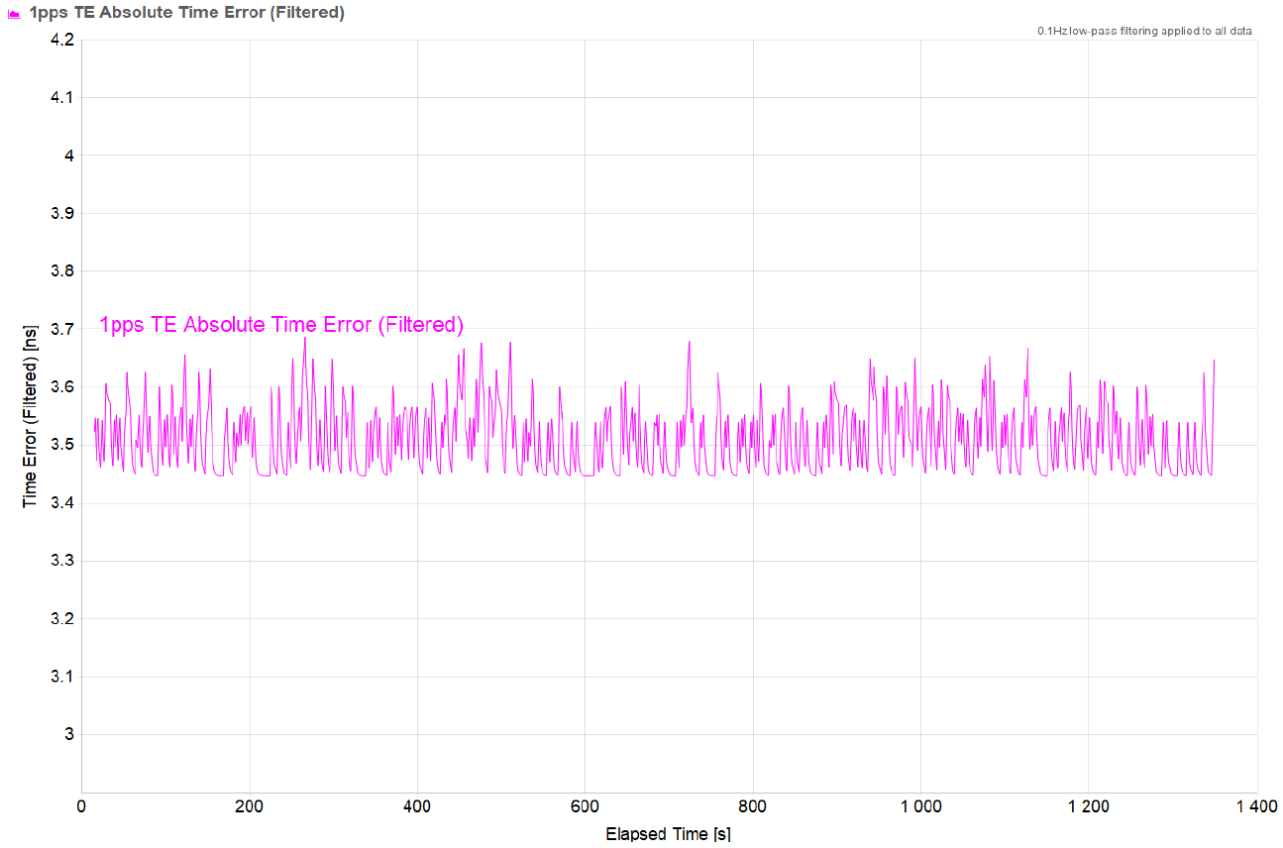


Figure 5. T-BC/T-TSC Time Error Noise Generation, Filtered (max|TE|L)

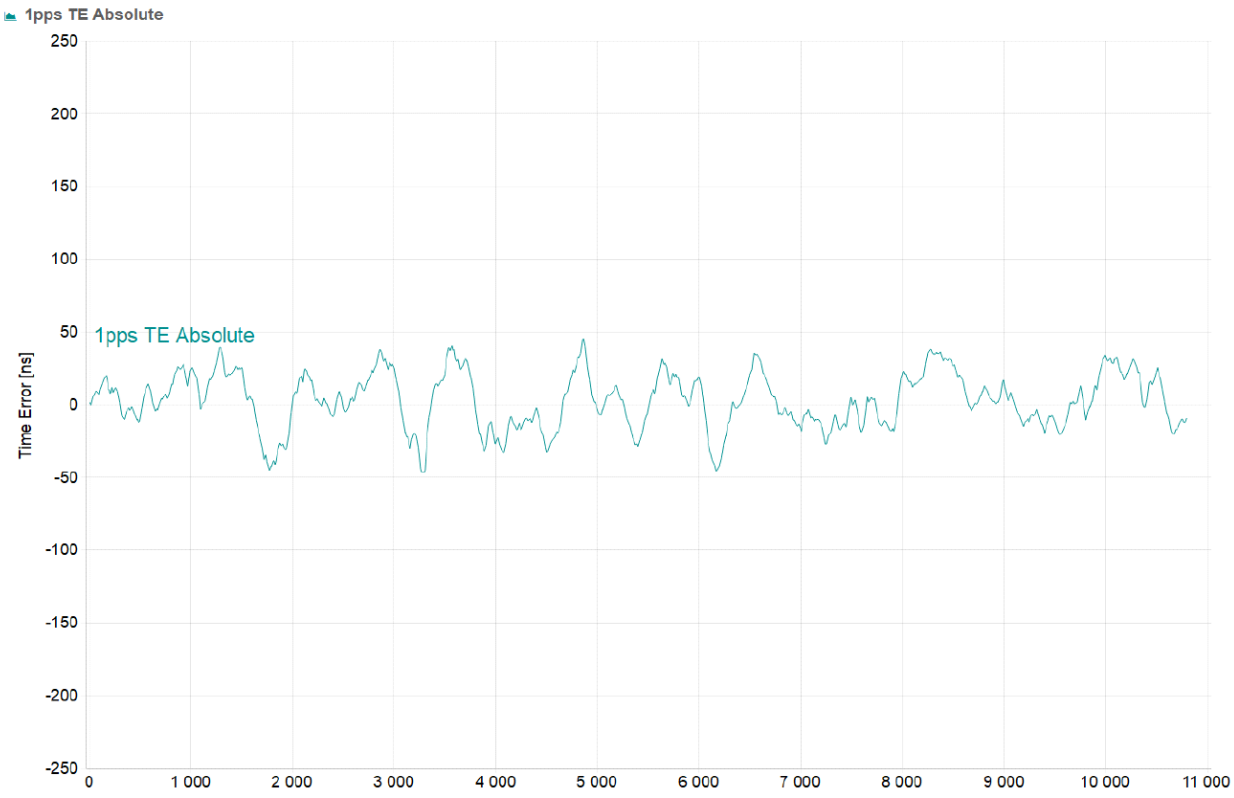


Figure 6. T-BC-P/T-TSC-P Time Error Noise Generation (max|TE|)

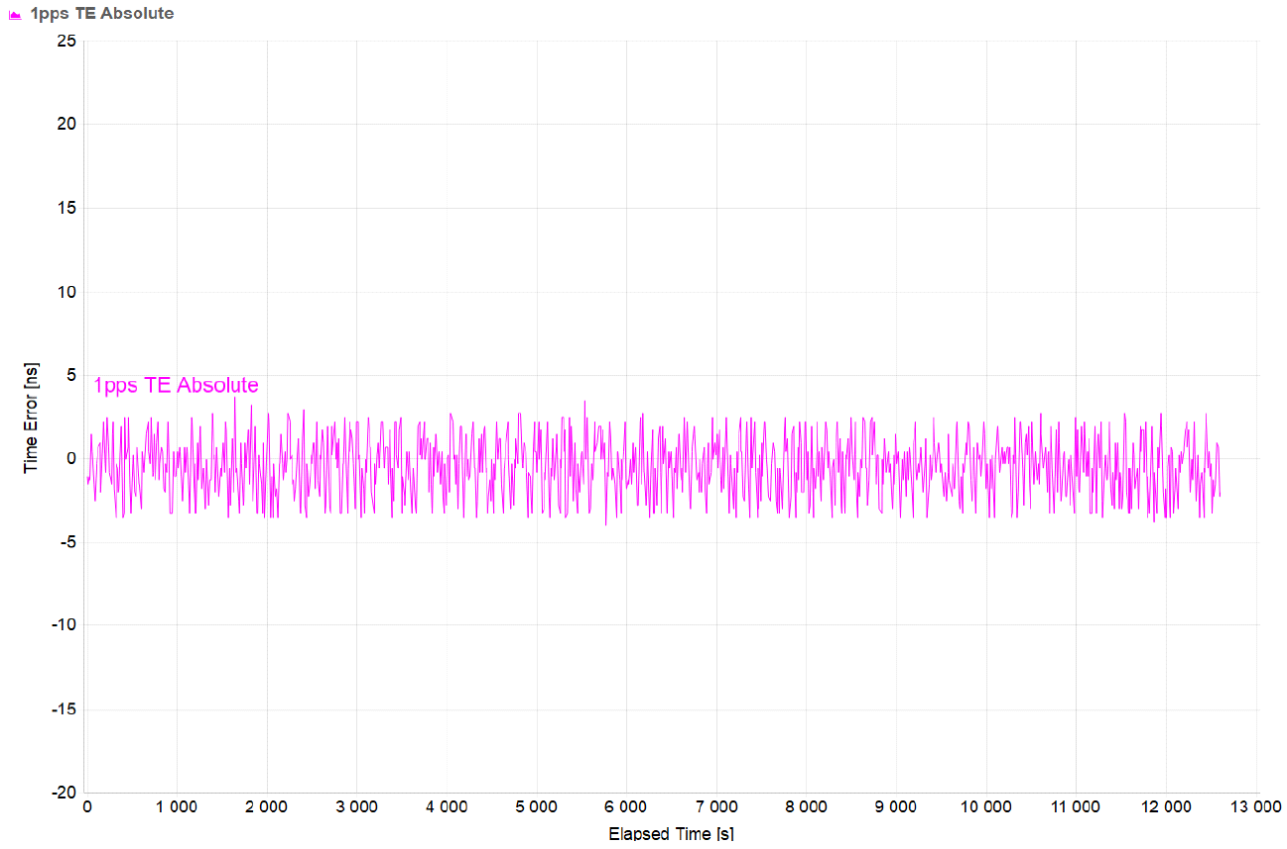


Figure 7. T-BC-A/T-TSC-A Time Error Noise Generation (max|TE|)

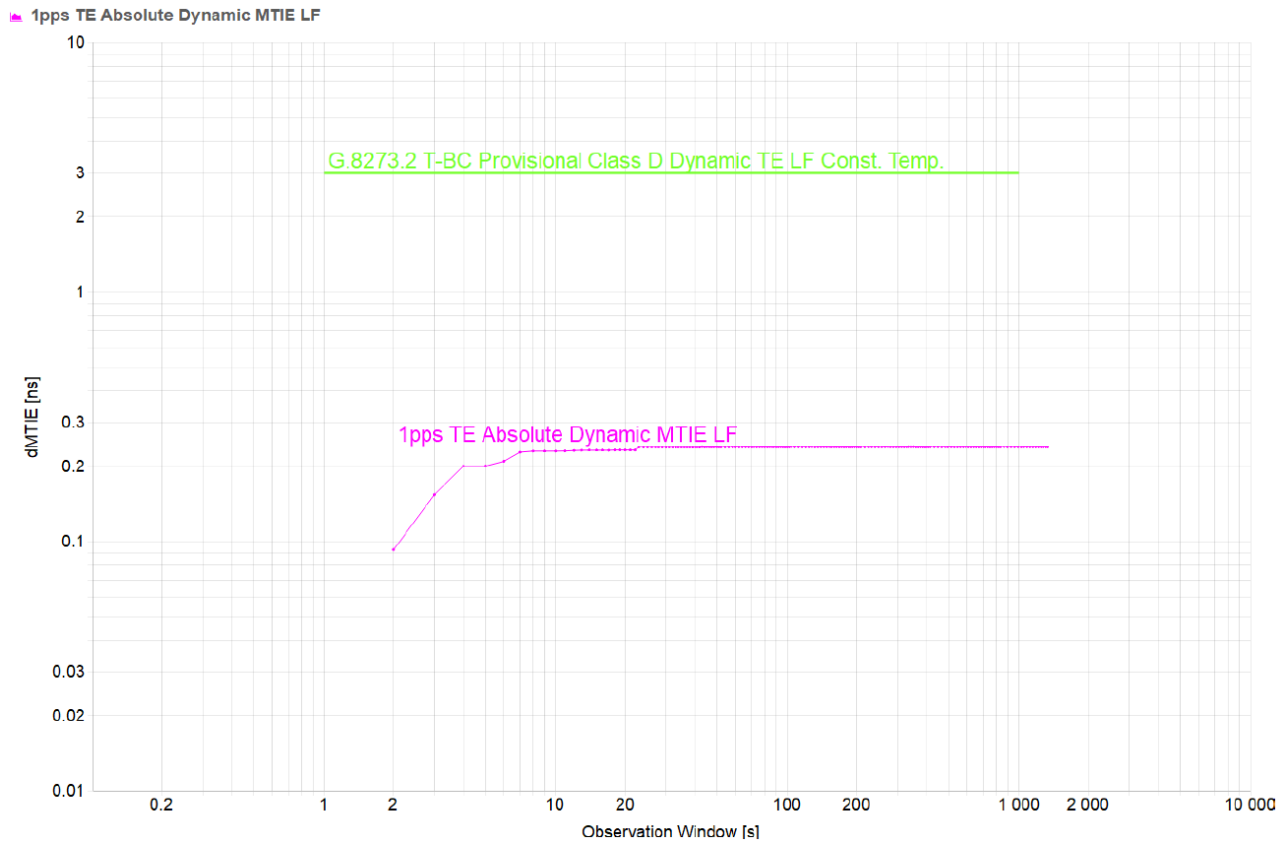


Figure 8. T-BC/T-TSC Dynamic Time Error Noise Generation (dTEL MTIE)

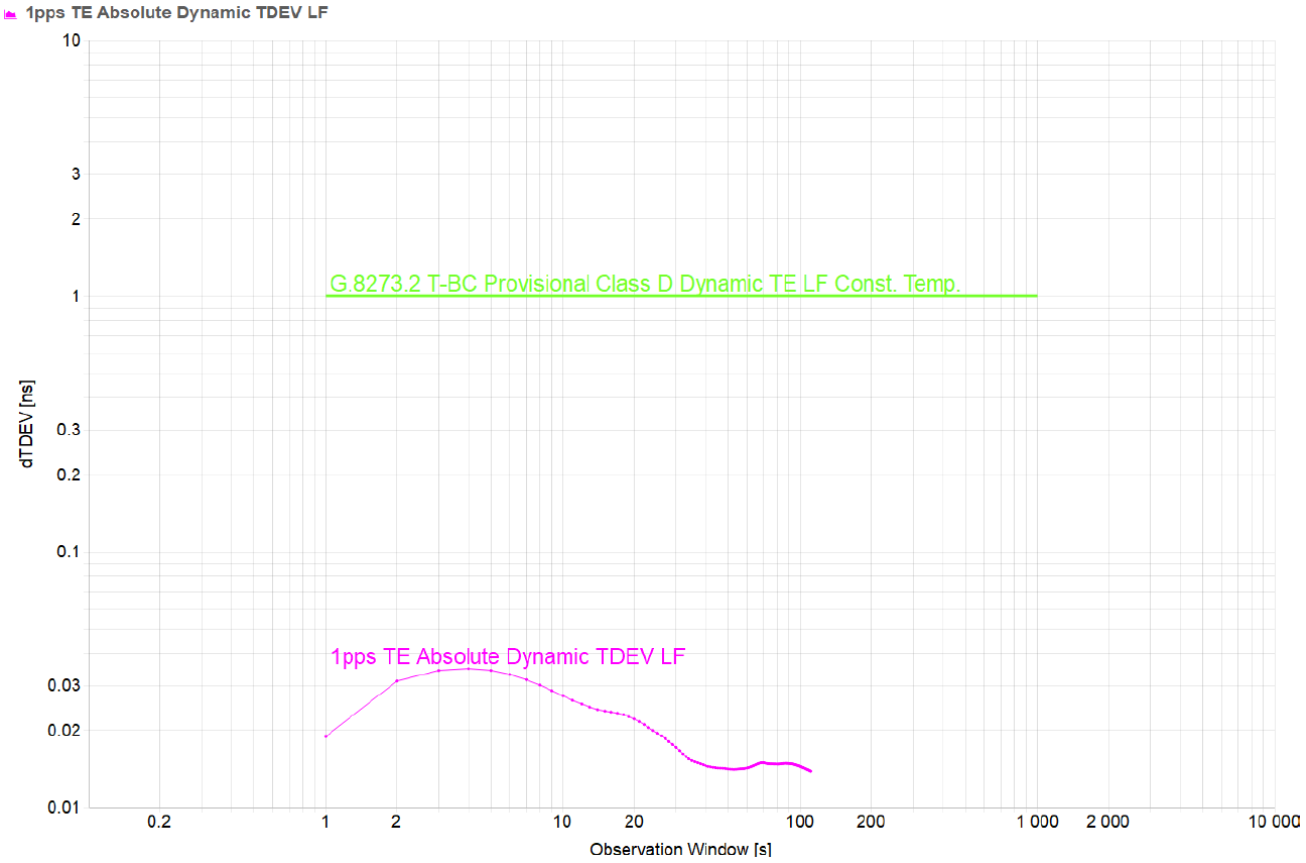


Figure 9. T-BC/T-TSC Dynamic Time Error Noise Generation (dTEL TDEV)

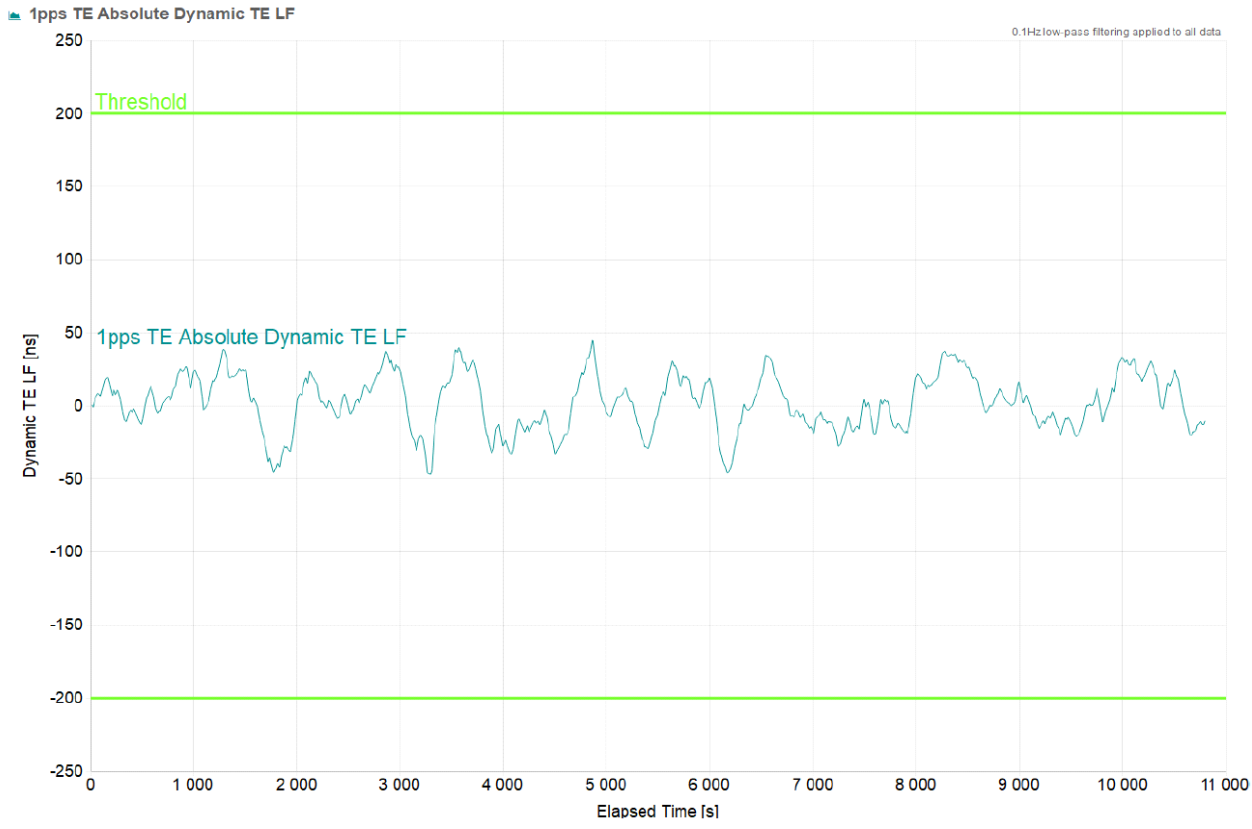


Figure 10. T-BC-P/T-TSC-P Dynamic Time Error Noise Generation (dTE_L)

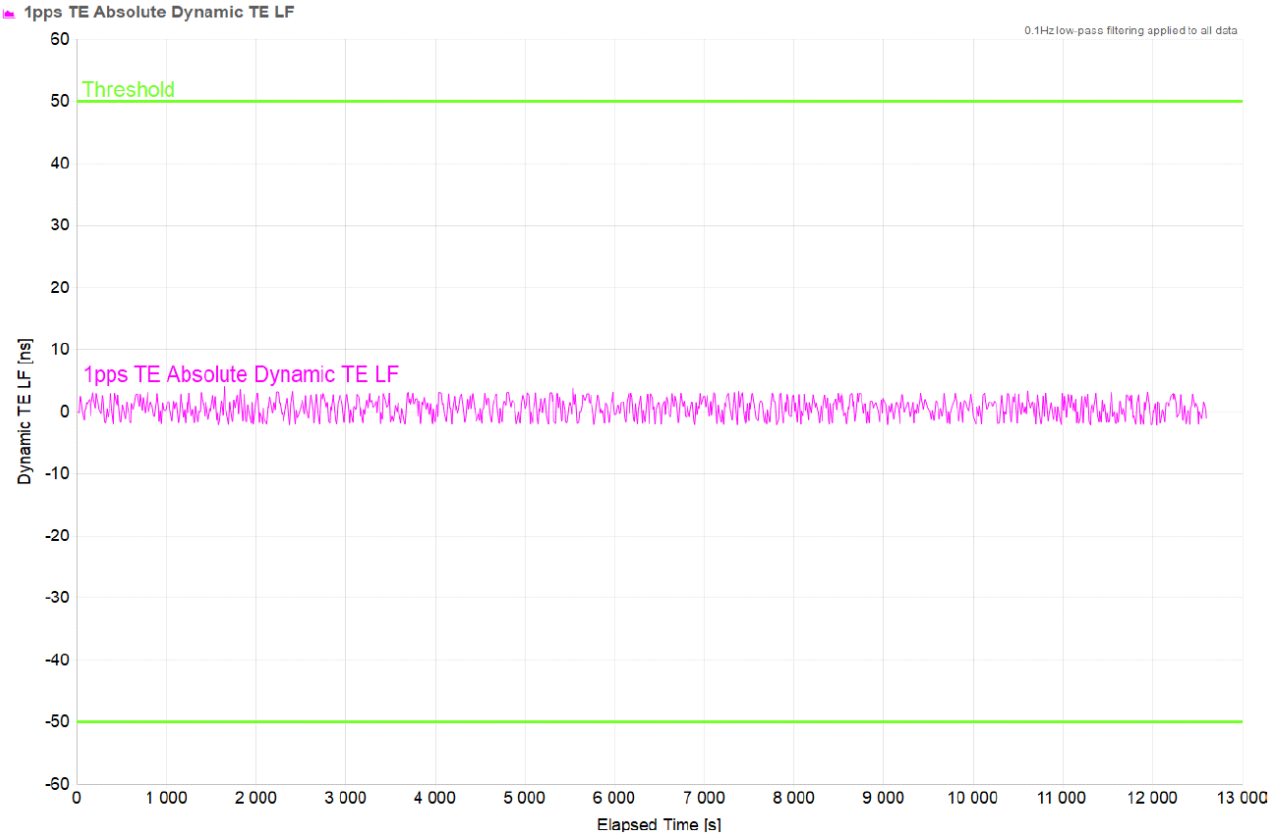


Figure 11. T-BC-A/T-TSC-A Dynamic Time Error Noise Generation (dTE_L)

5.2 PDV Tolerance

5.2.1 Time Noise Tolerance for PTS - G.8273.4 Clause 8.3 (using G.8275.2 Profile)

For test details, see the *Calnex Test Guide* CX3013. This test checks whether the equipment clock can maintain network limits at the output (max|TE|L < 1350ns) with maximum noise at the input.

- Reference Model 1 (NoBCs_HighStablility) – Network Topology with no intermediate time-aware devices
- Reference Model 2 (WithBCs_NormalStablility) – Network topology of three islands, or two time-aware bridging devices

The High Stability PDV profiles model a packet selection characteristic per below (based on G.8271.2):

- Selection window = 200s
- Selection percentage = 0.25% fastest packets

The pattern is then normalized so that the peak-to-peak pktSelected2wayTE is <1100 ns.

The following figures show the TE performance at the output for the two reference models.

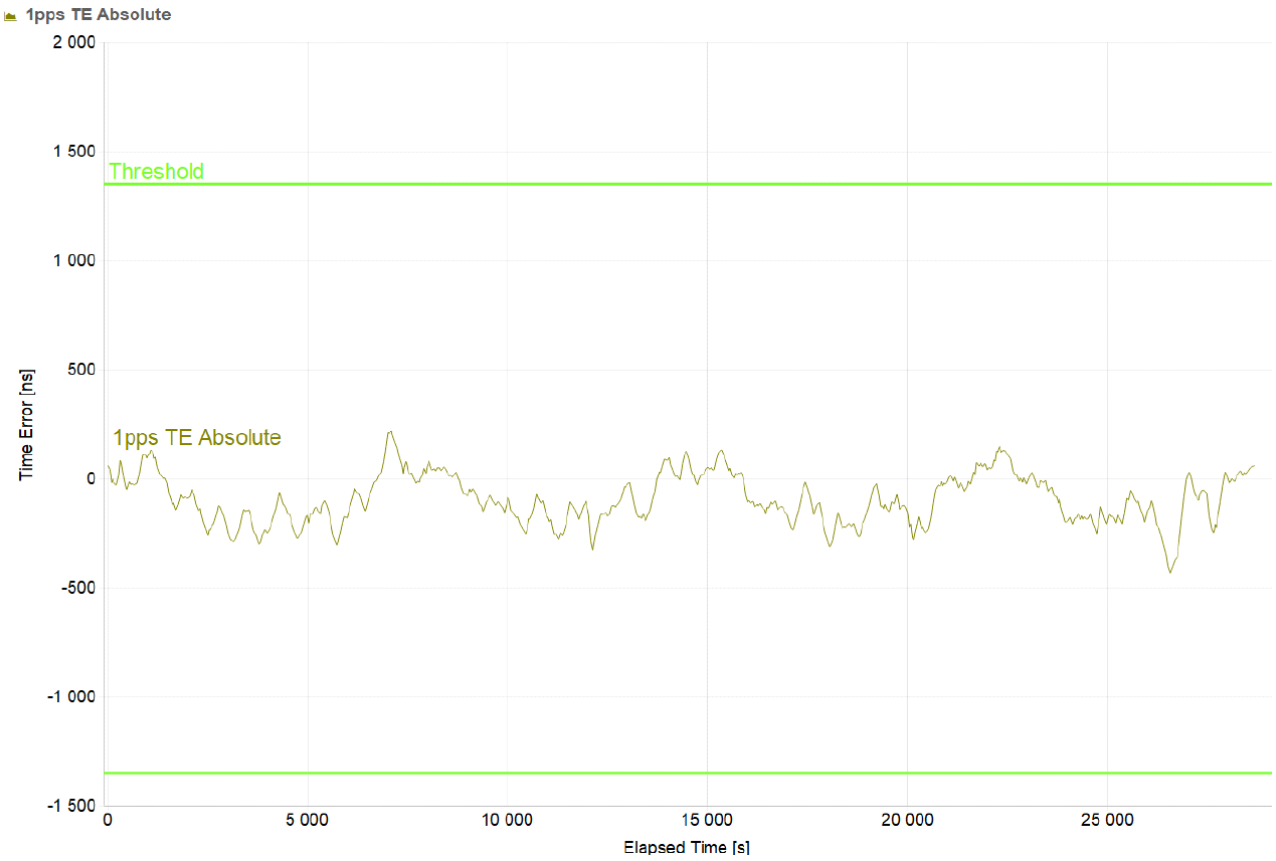


Figure 12. Reference Model 1 ($652ns_{pp}$)

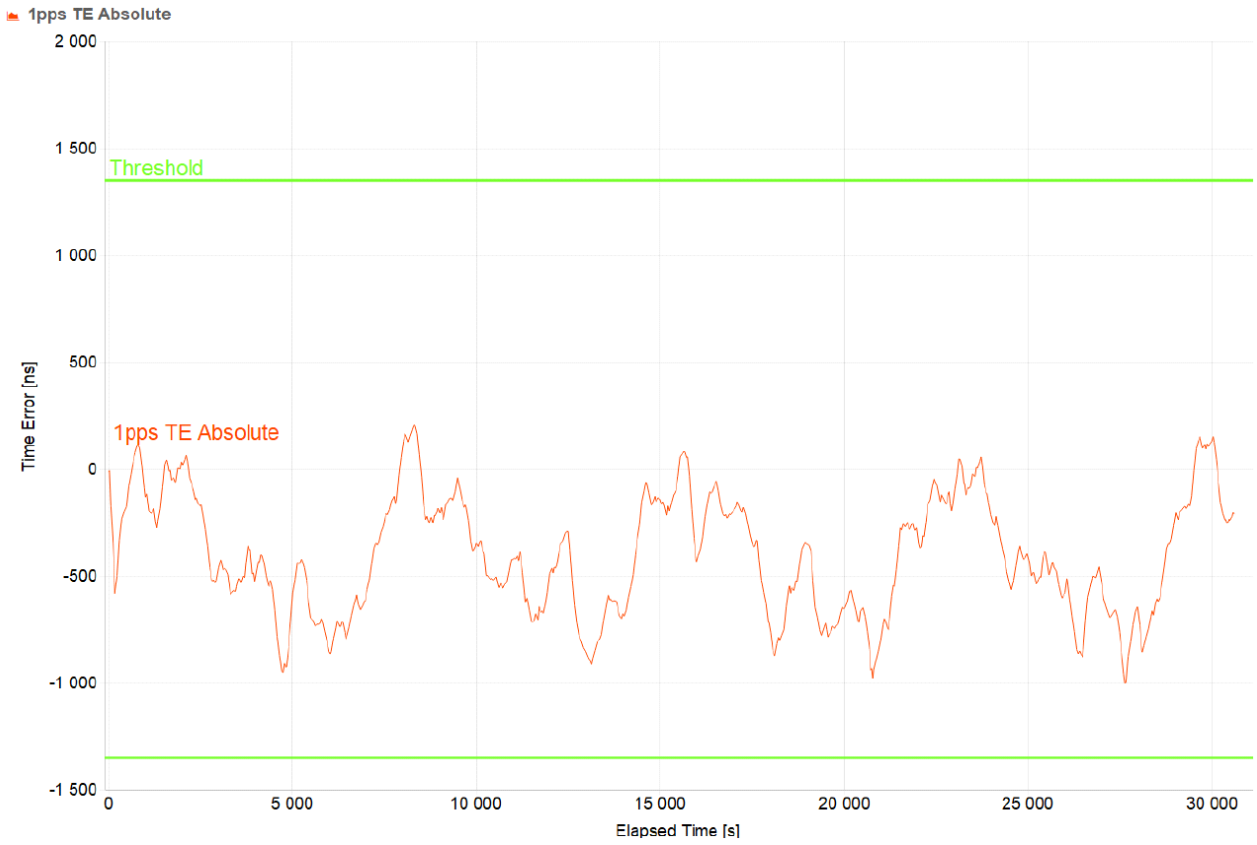


Figure 13. Reference Model 2 ($1211ns_{pp}$)

5.2.2 Time Noise Tolerance for APTS - G.8273.4 Clause 7.3 (using G.8275.2 Profile)

For test details, see the *Calnex Test Guide* CX3013. This test checks whether the equipment clock can maintain network limits at the output (max|TE|L < 1350ns) with maximum noise at the input while locked to a local time reference source. The PDV models used are the same as PTS.

The pattern is then normalized so that the |pktSelected2wayTE| is < 1100ns.

The following figures shows the TE performance at the output for the two reference models.

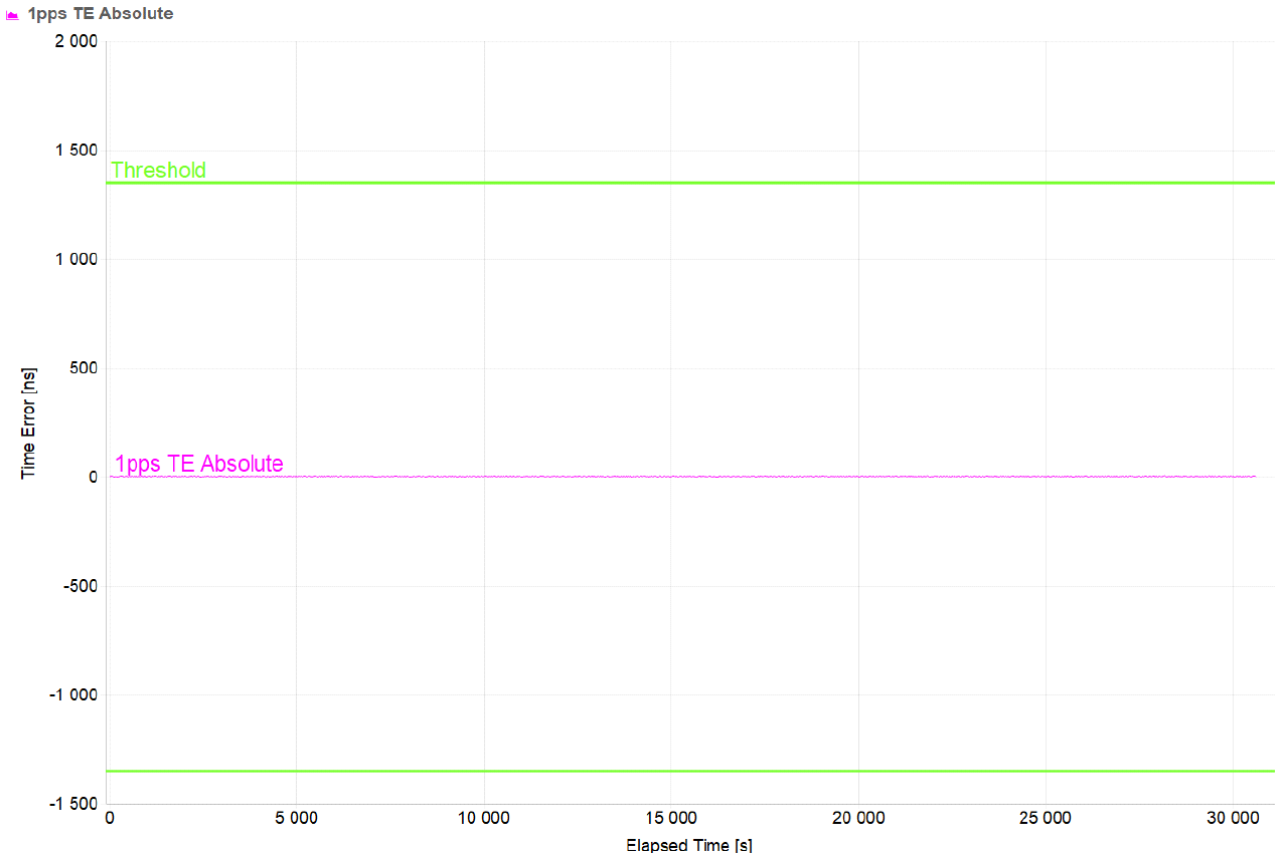


Figure 14. Reference Model 1 ($7.5ns_{pp}$)

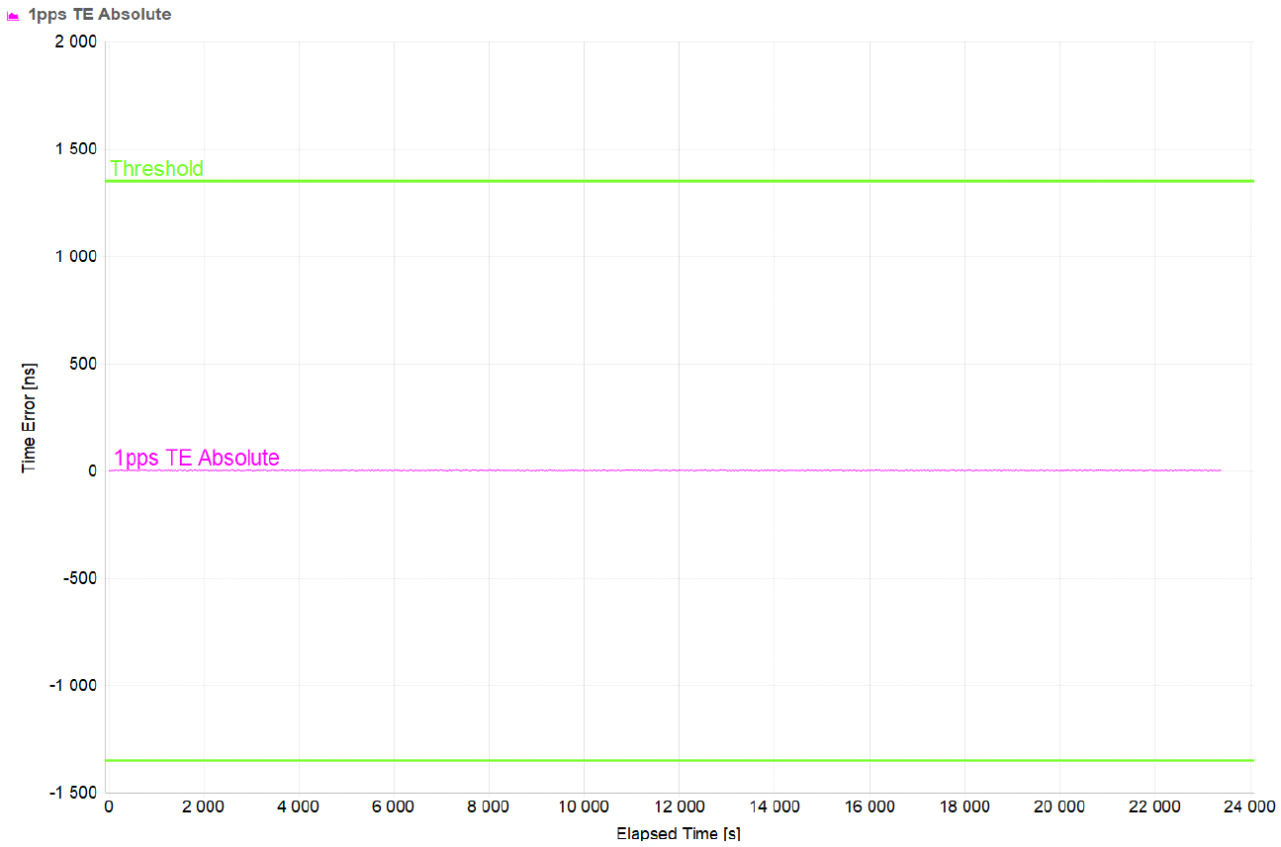


Figure 15. Reference Model 2 ($6.75ns_{pp}$)

5.2.3 Noise Tolerance for PEC-S-F (Using G.8265.1 Profile)

Using G.8261 PDV profiles (traffic model “b”) provided by Calnex - applying G.8261 EEC option 1 Wander Limit mask.

The following figure shows the MTIE performance at the output for the applicable G.8261 PDV test cases, as measured to the G.8261 option 1 limit.

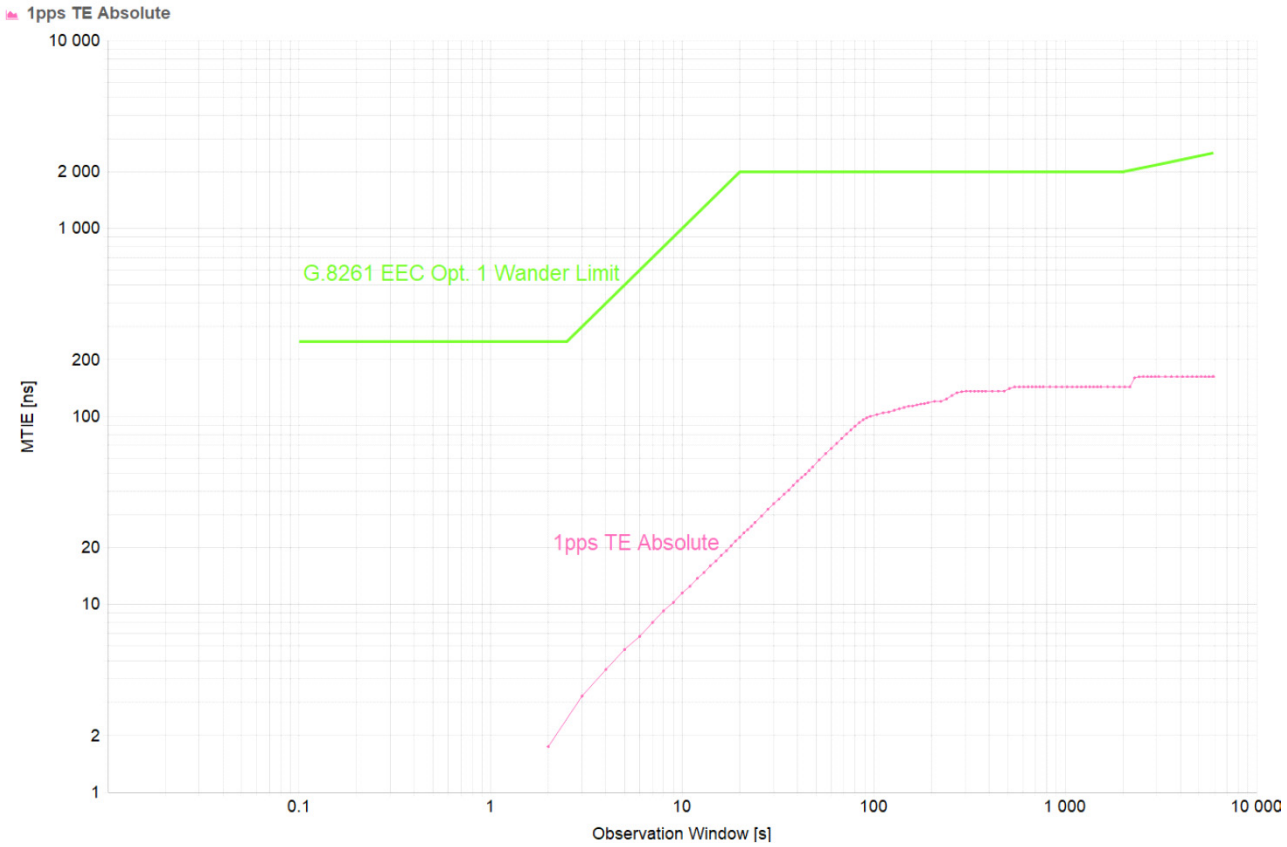


Figure 16. Test Case 12b (MTIE = 161ns)

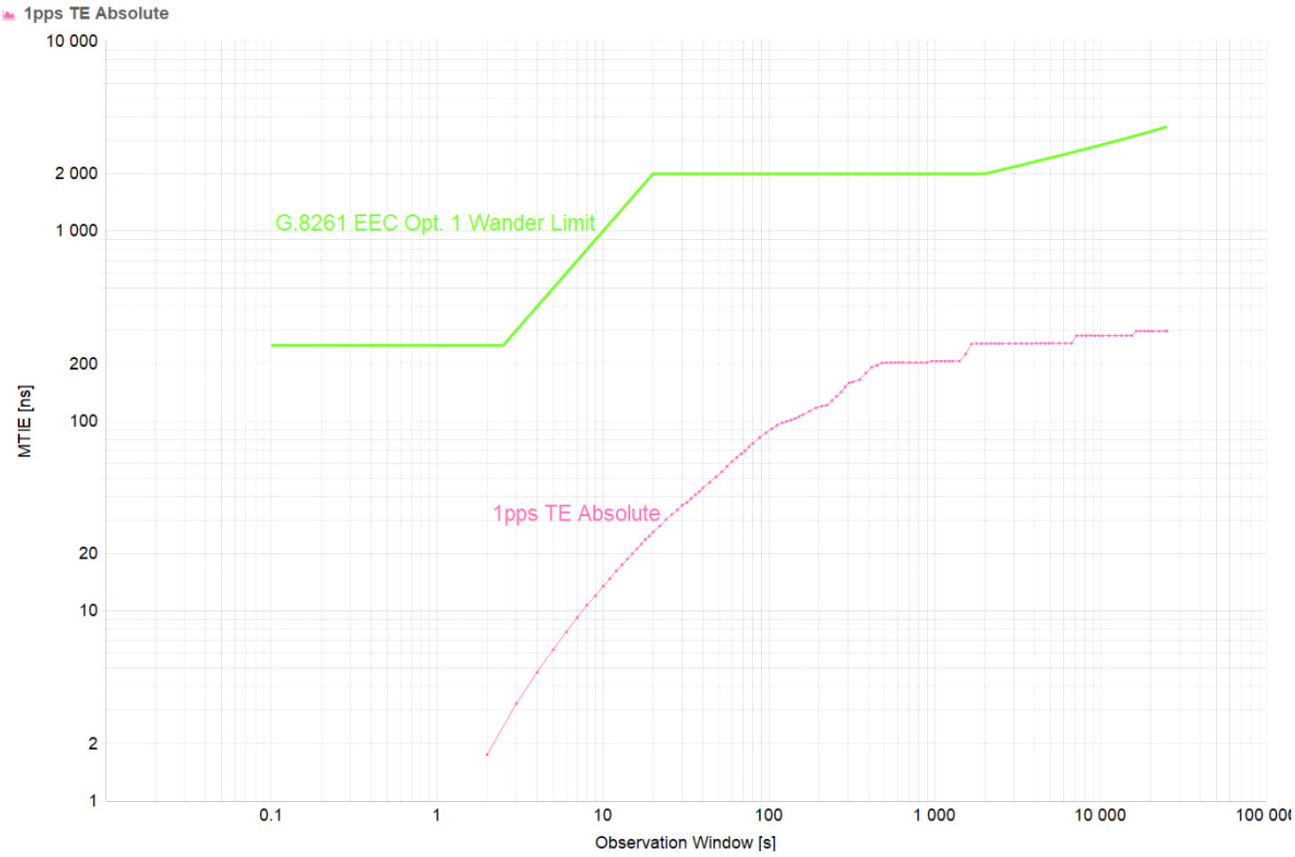


Figure 17. Test Case 13b (MTIE = 295ns)

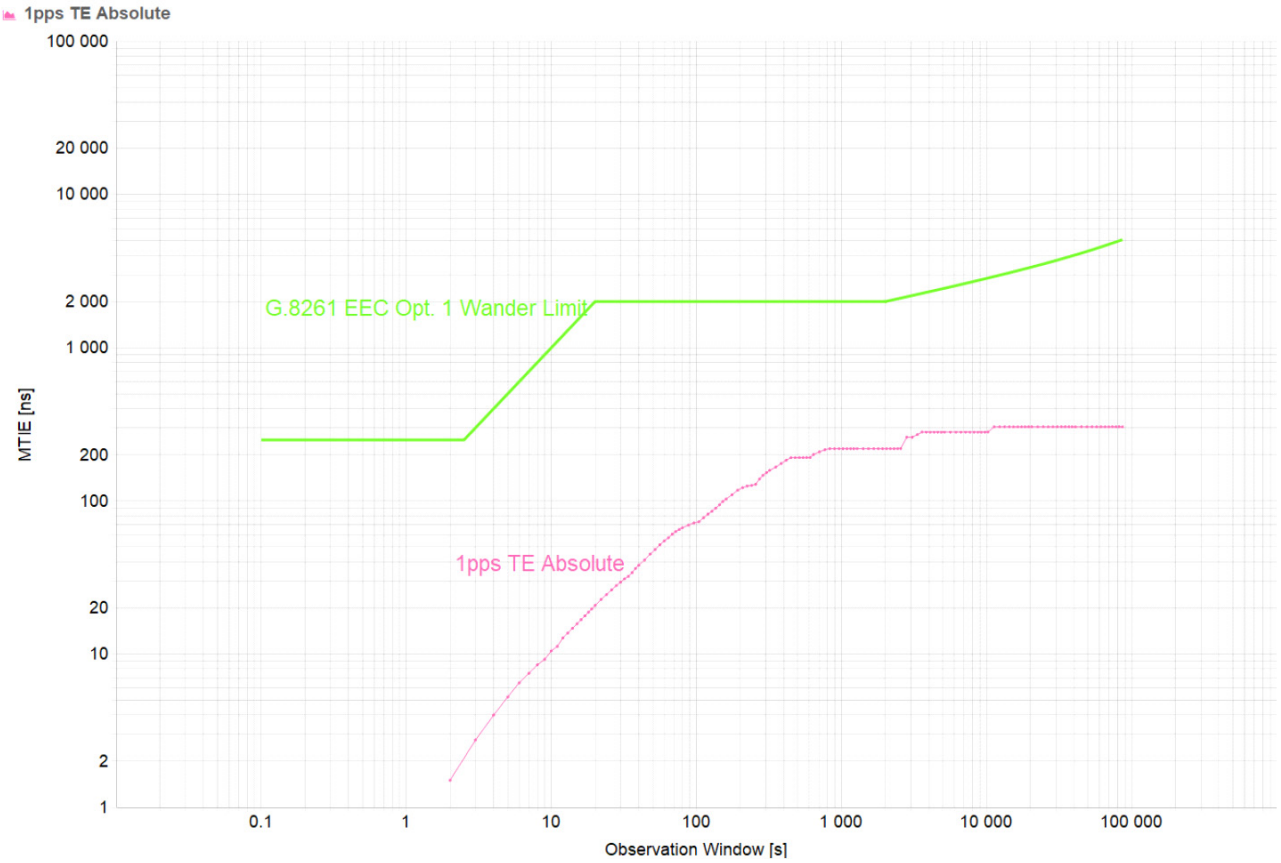
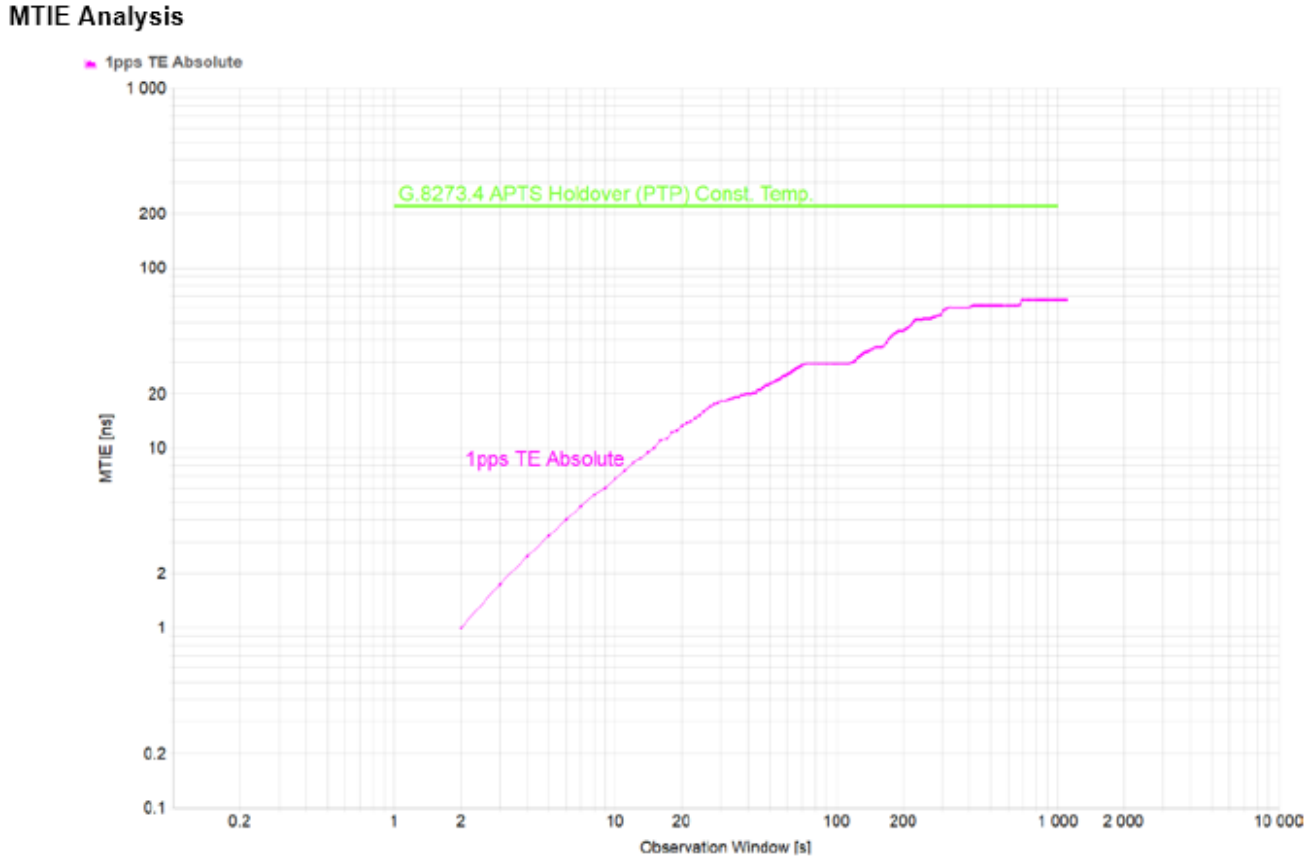


Figure 18. Test Case 14b (MTIE = 303ns)

5.2.4 APTS Holdover with PTP (Using G.8275.2 Profile)

This test checks whether the equipment clock can maintain time holdover at the output (max|TE|L < 222ns) with maximum noise at the PTP input (used Calnex G.8273.4 Reference Model 2). The maximum observation interval is 1000s.



Min [ns]	1
Max [ns]	66.75
Max-Min [ns]	65.75

Figure 19. APTS Holdover with PTP

6. Glossary

Abbreviation	Description
1PPS	A 1Hz synchronization signal (one pulse per second)
FCW	Frequency control word

7. Revision History

Revision	Date	Description
1.02	May 10, 2023	Updated to add the following features, as introduced in v4.2.1: ▪ Single path support for assisted reference tracker ▪ Enhanced physical layer assistance support for adaptive reference tracker ▪ Support for external time stamper alignment, such as ts2phc ▪ Ability to seed a local pathAsymmetry on a per-PTP port basis ▪ New PTP clock holdover qualification period
1.01	Sep 28, 2021	▪ Updated to add Assisted Partial Time Support (APTS) feature, as introduced in v4.1.0 ▪ Updated Section 5 with latest performance results based on v4.1.0
1.00	Feb 2.21	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.