802.1AS Recovered Clock Quality Testing

Revision 1.0

18 October 2016

Major Contributors:

Bogdan Tenea
Eric Percival
Alon Regev

Contributors:

All members of the Avnu Testability Subgroup that participated in the weekly meetings contributed to this document. The following list includes all attendees of the Testability Subgroup between August 24, 2015 and October 18, 2016

Abuzafor Rasal
Alon Regev
Ashley Butterworth
Bogdan Tenea
Bob Noseworthy
Brandon Smith
Brant Thomsen
Charles Kozierok
Colt Correa
Craig Gunther
Eric Percival
Gordon Bechtel
Jörg Angstenberger
Mark Hu
Peter Scruton
Ralf Eberhardt
Reid McGaughey
Rudy Klecka
Tanuja Mallappa
Tim Frost
Max Turner

Please visit www.avnu.org for additional resources or email Avnu Administration (administration@avnu.org) for further assistance.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, EXPRESS, IMPLIED, OR STATUTORY. AVNU ALLIANCE MAKES NO GUARANTEES, CONDITIONS OR REPRESENTATIONS AS TO THE ACCURACY OR COMPLETENESS CONTAINED HEREIN. Avnu Alliance disclaims all liability, including liability for infringement, of any proprietary or intellectual property rights, relating to use of information in this document. No license, express or implied, by estoppel or otherwise, to any proprietary or intellectual property rights is granted herein.
THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, Avnu ALLIANCE, INC., AS WELL AS THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION OR STANDARD HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY OR COMPLETENESS OF RESPONSES, OF RESULTS, OF WORKMANLIKE EFFORT, OF LACK OF VIRUSES, OF LACK OF NEGLIGENCE OR NONINFRINGEMENT. FURTHER, THIS DOCUMENT DOES NOT GRANT YOU ANY RIGHTS TO PRACTICE ANY PATENTABLE PROCESSES STATED OR DESCRIBED HEREIN, NOR DOES IT CONSTITUTE A LICENSE, EXPRESS OR IMPLIED, IN ANY CLAIMS OF PATENT WHICH MAY BE NECESSARY FOR IMPLEMENTATION OF THIS SPECIFICATION OR STANDARD.

Table of Contents

Revision History

Revision	*Author(s)*	*Date*	*Description*
0.1	Bogdan Tenea, Alon Regev, Eric Percival	2016-06-06	Initial release of full document.
1.0	TWG	2016-10-18	Balloted version by Avnu TWG

1 Introduction

The Generalized Precision Time Protocol (gPTP) provides an accurate time base to all time aware elements in an Ethernet network. Applications such as Audio/Video and Automotive/Industrial Control rely on the time provided by this protocol for synchronization of events. This specification defines methods to measure gPTP recovered clock quality for Avnu certification.

To measure the accuracy of the clock synchronization, the traditional approach is to use a 1 Pulse Per Second (1PPS) physical output. While this is a good approach, there are cases where using a 1PPS output is not feasible - for example, in the automotive environment, Electronic Control Units (ECUs) have limited size and resources, and a separate output pin for 1PPS is often considered too costly. Also, in deployed networks, a 1PPS is again not a good option, as the output of the signal from the Grandmaster and the Slaves have to be connected to the same test equipment for analysis, which makes monitoring over large networks or networks with numerous Slave clocks difficult.

It is recommended that the additional methods be used in addition to the 1PPS method, as new methods provide l ways to detect some errors that cannot be detected using 1PPS. They also enable remote monitoring capability without requiring a direct physical connection to the device.

Measuring the clock quality without the use of 1PPS is a desire recognized in the broader 1588 community. To address this, the IEEE 1588 Working Group current draft includes methods for monitoring ingress & egress event messages enabling measurement of the recovered clock quality. The methods defined in the P1588 draft revision will be referenced in this document in order to avoid double specification, even though this is not yet an approved standard.

2 Glossary

Term	*Meaning*
AVB	Audio Video Bridging
gPTP	Generalized Precision Time Protocol
CDS	Certification Development Subgroup
DUT	Device Under Test
ECU	Automotive Electronic Control Unit
GM	Grandmaster clock
Integrated Devices	Devices that contain both a bridge and endpoint in the same package, such that neither can be tested independently.
PTP	Precision Time Protocol
PLL	Phase Locked Loop
PPS	(1) Pulse Per Second physical signal
1PPS	1 Pulse Per Second physical line
pps	Packets per second
TE	Time Error, as defined in Section 4.2

3 References

Name	*Reference*
gPTP	IEEE Std. 802.1AS-2011 "IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", as amended by "IEEE Std. 802.1AS-2011/Cor 1-2013 "IEEE Standard for Local and metropolitan area networks— Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks— Corrigendum 1: Technical and Editorial Corrections" and "802.1AS-2011/Cor 2-2015 - IEEE Standard for Local and metropolitan area networks-- Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks-- Corrigendum 2: Technical and Editorial Corrections"
802.1BA	IEEE Std 802.1BA- 2011, "Audio Video Bridging (AVB) Systems," 30 Sept 2011.
1588-Slave-Monitoring	IEEE 1588 Draft Proposal for Slave Event Monitoring https://ieee-sa.imeetcentral.com/1588/file/43739452

4 Scope

This document defines methodologies for validating that devices meet requirements for gPTP recovered clock quality. The methods enable such validation for certification purposes, lab testing before deployment, as well as field testing during and after deployment.

4.1 Challenges

gPTP is designed to work with lower quality oscillators and without hardware based PLLs. This means that it may display issues related to phase noise between successive synchronization events, as well as any phase noise added in any node on the path from the Grandmaster.

4.2 Objective

The fundamental objective is to measure recovered clock quality by measurement of the Time Error, which is defined as the difference between time measured at the device under test and a reference, at the same instant:

$$\text{Time Error} = \text{Time}_{\text{(measured on DUT)}} - \text{Time}_{\text{(reported at reference)}}$$

4.3 Qualification

It is expected that two clocks will never be perfectly synchronized and that there will be some non-zero Time Error between them. There are a number of important derived measurements are based on sampling the Time Error over a period of time called an Observation Window. The most important of these is the Maximum Time error, which is the maximum absolute value of the Time Error over that interval.

One of the original design goals of gPTP is that "any two time-aware systems separated by six or fewer time-aware systems (i.e., seven or fewer hops) will be synchronized to within 1 µs peak-to-peak of each other during steady-state operation" [gPTP B.3]. Thus each hop (Bridge or EndStation) may introduce up to ±80 ns (~540/7) of error. This having been said, different industries may have different requirements. More details about the requirements for Avnu certification are given in Section 8.

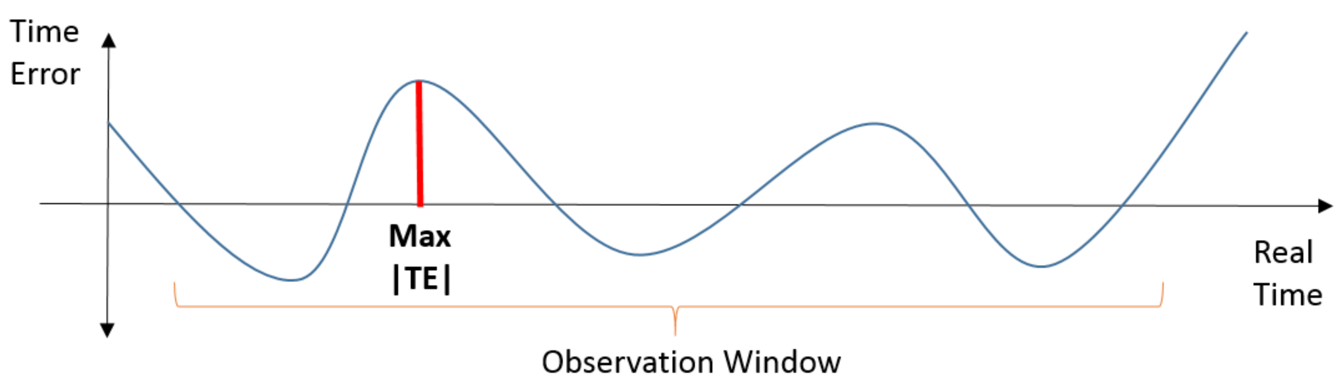

Figure 1: Measuring Maximum Time Error over an Observation Window

5 Measurement methods [Informative]

This section gives a brief introduction of the methods that can be used for assessment of clock quality.

5.1 1PPS Method

The traditional method of measuring Time Error is the use of a physical 1PPS line. This is used to compare the difference between the rise time of a signal transmitted at the edge of each new second on the Slave clock to the rise times of the same signal on the reference clock. In lab conditions the Time Error measurement is done by connecting the DUT directly to a GM with a similar 1PPS output, and observing the delta time between the pulse coming from the GM and the pulse coming from the Slave.

Note: The internal clock of the Slave might have a static offset that is canceled out by an opposite static offset in the logic that drives the 1PPS output, and this error condition is not observable when only the 1PPS measurement method is used.

5.2 Ingress Reporting Method

The Ingress Event Monitor defined in [1588-Slave-Monitoring] relies on noting the recovered clock receive time of incoming Sync event messages (T2) on the DUT. Based on knowing the transmit time of the message in the Master timebase (T1) and the path delay, the Slave can save the raw data or compute the Time Error at T2 and provide it to a test or monitoring station. The data can either be made available out-of-band or be transmitted in-band using the 1588 TLVs defined for this purpose. The implementation must provide a method for enabling and disabling the measurements and for accessing the measurement data. More details on how to apply this can be found in Section 7.

Note: either when reporting the raw values of T1 and T2, or when reporting the computed offsetFromMaster, the Time Error is the Slave's perception of the Time Error, since it relies on the T2 information coming from the Slave. In this case the tester is just acting as a collecting agent for samples provided by the Slave. The sampling rate is dependent on the incoming messages, and Time Error in the interval between Sync messages is not observable.

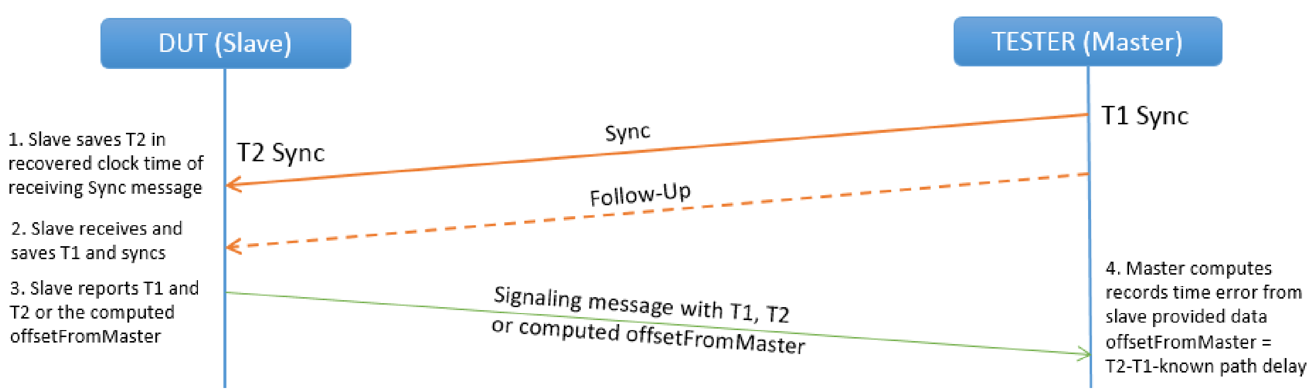

Figure 2: Ingress Monitoring using TLVs message exchange

5.3 Reverse Sync Method

As defined in 1588, PTP operates on the concept of independent PTP instances running on different time synchronization domains. One method of exposing a recovered clock is to output that recovered clock back on to the network through PTP on what is or appears to be a separate PTP instance running on a different domain. In gPTP terms, one would link the ClockSource of the Ordinary Clock instance used for forwarding the recovered clock to the ClockTarget interface of the Ordinary Clock instance of the Slave the one actually wants to measure. To this sense, the gPTP node implementing this acts in many senses as a hot-standby GM, as defined in the gPTP-Rev that supports multiple domains.

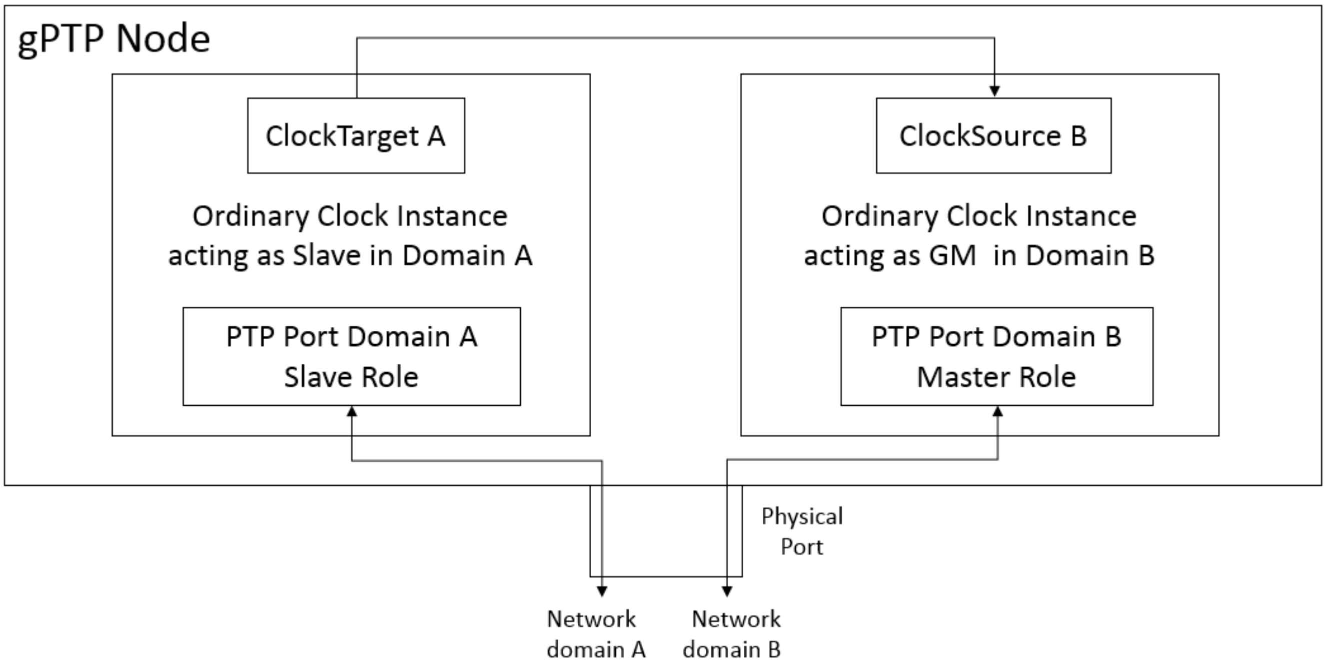

Figure 3: Node with two PTP Instances for outputting the Recovered clock

For simplicity, for the purpose of the Time Error measurement, an implementation does not need to fully implement two PTP instances. The observability information needed to compute the Time Error measurements are contained in the Sync/Follow-up messages. A detailed specification on how to create the reduced implementation solely for the purposes of certification is described in detail in Section 6.

Note: even running the reduced version without a complete BMCA will require some extra resources from the device to trigger the reverse messages

The message exchange mechanism used with Reverse Sync starts with the standard gPTP message flow, to which it adds the transmission of Reverse Sync messages. A simplified message exchange can be described as follows:

1. The Slave sends a pDelay Request to the Master to which it receives a pDelay Response/Follow-Up which enables it to calculate the path delay towards the Master [gPTP 11.1.2]

2. The Master sends a Sync/Follow-Up message with proper origin timestamp and correction field for the Slave to be able to adjust its local clock with the current offset from the Master [gPTP 10.2.12]:

<offsetFromMaster> = <Time on the slave clock> — <Time on the master clock> =
= <recovered slave time> – <origin timestamp> – <correction fields> – <path delay to master>

3. The Master sends a pDelay Request to the Slave to which it receives a pDelay Response/Follow-Up which enables it to calculate the path delay towards the Slave [gPTP 11.1.2].

4. The Slave sends a reverse Sync/Follow-Up message with the origin timestamp indicating the current recovered time on the Slave, taking in to account the rate ratio from the Grandmaster, which enables the Tester to calculate the Time Error of the Slave relative to its own recovered clock from the Grandmaster:

<offsetFromSlave> = <Time on the master clock> - <Time on the slave clock> =
= <master time> – <origin timestamp> – <correction fields> – <path delay to slave>

Note: this assumes that the test for verifying that the reported path delay (obtained from either the management interface or from Ingress method) corresponds to the actual cable length passed successfully.

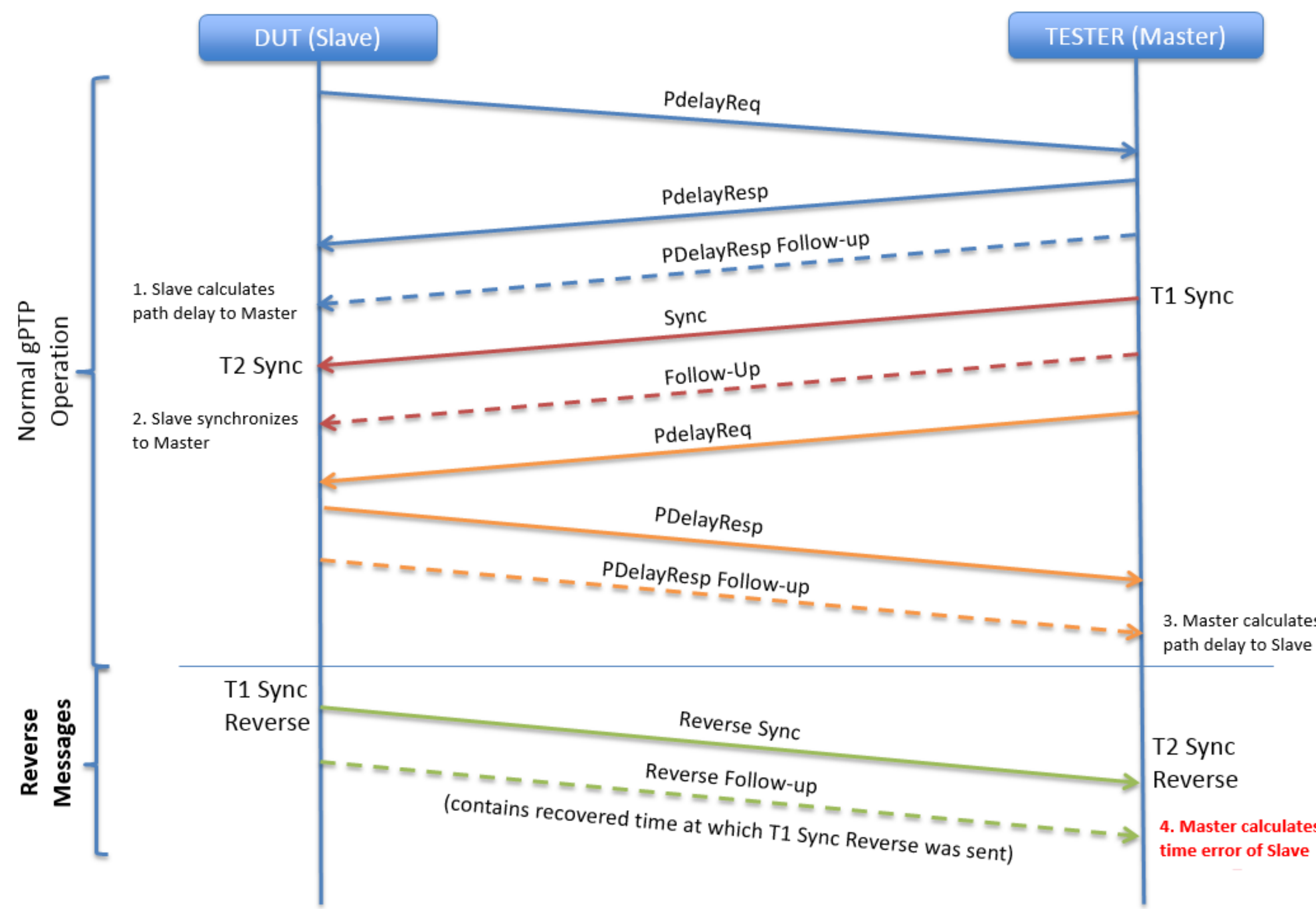

Figure 4: Reverse Sync Message exchange

6 Measuring gPTP clock accuracy using the Reverse Sync Method

The Reverse Sync is a method of measuring the accuracy of the synchronization of a gPTP clock through standard gPTP functionality. With this mechanism, a gPTP Slave device transmits timestamped messages, which can be used to determine the accuracy of the slave's clock. In a basic Master–Slave configuration, the Slave sends back its recovered clock to the Master, which will act as a Tester device and can measure the accuracy of that clock.

The method is called Reverse Sync because it uses the standard gPTP Sync/Follow-Up messages to relay the timing information, but the direction of the messages is reversed to flow from the Slave to the Master.

The method is very similar to the 1588 Egress Monitor mechanism regarding the principle it uses, which is to convey to the tester information about sent event messages. The main difference is in the fact that is does not rely on regular sent event messages from a Slave, such as PdelayReq and DelayReq, because of the need to have better sampling capabilities of the Time Error (more details are given in 6.5). The method does not require collection of measurements, as every measurement is sent out individually after it is taken, requiring no additional memory for storing data. It also does not require the measurements to be stored in a new TLV, as standard format Sync and Follow-up messages are used, giving the possibility of reusing existing code from Master implementations.

6.1 Control Points

The DUT shall provide the following control points for controlling the Reverse Sync:

1) A control point to enable and disable the Reverse Sync/Follow-up messages **<reverseSyncEnabled>**. The default value of this is FALSE.
2) A control point to set the domain of Reverse Sync/Follow-up messages **<reverseSyncDomain>**. For certification it is proposed that a different unique domain should be used than the one that's being measured. When using in deployed networks a different domain shall be used.
3) A control point to set the Reverse Sync/Follow-up message rate as % of incoming rate **<reverseSyncRate>.** The default is defined in Section 6.3

6.2 Message format

The Reverse Sync and FollowUp messages shall use the format defined in [gPTP] for the medium used. For example, for full duplex point to point links, the Header specified in [gPTP] 11.4.2, the Sync message specified in 11.4.3 and the Follow_Up message specified in 11.4.4 shall be used. The applicability for other transport mediums is out of the scope of this version of the document.

The Header and the Sync and Follow_Up message shall be modified as follows:

1. preciseOriginTimestamp or originTimestamp (depending on clock step mode) shall contain the recovered clock in the Grandmaster time base at the moment the Reverse Sync message crossed the reference plane.

2. the correctionField of both Sync and Follow-up messages should be set to 0
3. the domain of both Sync and Follow-up shall be set to the configured **<reverseSyncDomain>**

Note: The intent for points 1 and 2 is to verify that the device is actually recovering the clock and not just acting as a one port transparent clock. In this sense, the correction field is expected to be set to 0 unless needed for representing fractional nanoseconds or unless needed by device implementation specific constraints.

6.3 Message timing

Some of the most common components of the synchronization error come from improper syntonization (frequency locking). The example of this for gPTP would be improper use of the rateRatio parameter when recovering the clock. The error accumulates as time passes between two successive synchronization points. If the Reverse Sync messages would always be sent immediately after a synchronization point (after adjusting the clock offset), such errors would be hidden from the measurement.

Having the Reserve Sync occur at different points in time within a synchronization interval will yield a better statistical result. This can be achieved by running the Reverse Sync packets at a higher rate than the incoming packets, for example at 2 x Nyquist (31.25ms for 8pps in), given the wide tolerance for incoming Sync rate.

Another less CPU intensive way is to make the sending of Reverse Sync messages independent of the time of incoming Sync messages using a rate for the Reverse Sync and Follow-up message sent by the Slave that is not an integer multiple of the rate of incoming Sync and Follow-up messages. A device must not strictly respect this exact rate at every sent message, but it must allow for the rate to be configurable in such a way that it achieves the desired measurement accuracy, by allowing it to sample at multiple points between incoming Sync messages over a certain period of time, such that the effect described in the first paragraph is avoided.

If <**reverseSyncEnabled**> is true, a Slave clock shall send the Reverse Sync messages at <**reverseSyncRate**> rate of the incoming Sync interval. The default value of<**reverseSyncRate**> shall be 95% of the incoming Sync message rate if known, or a rate 95% of 8 pps if the incoming Sync message rate is not known.

Note: This allows sampling at all points of the Sync interval within 20 samples, or for a rate of 8 pps, within 2.5 seconds. A CDS may overwrite this sample rate for faster sampling or for better granularity of the measurements. A certification test may choose to adjust this value if the actual rate as seen from the device makes it such that the Reverse Sync is always sent after the incoming Sync.

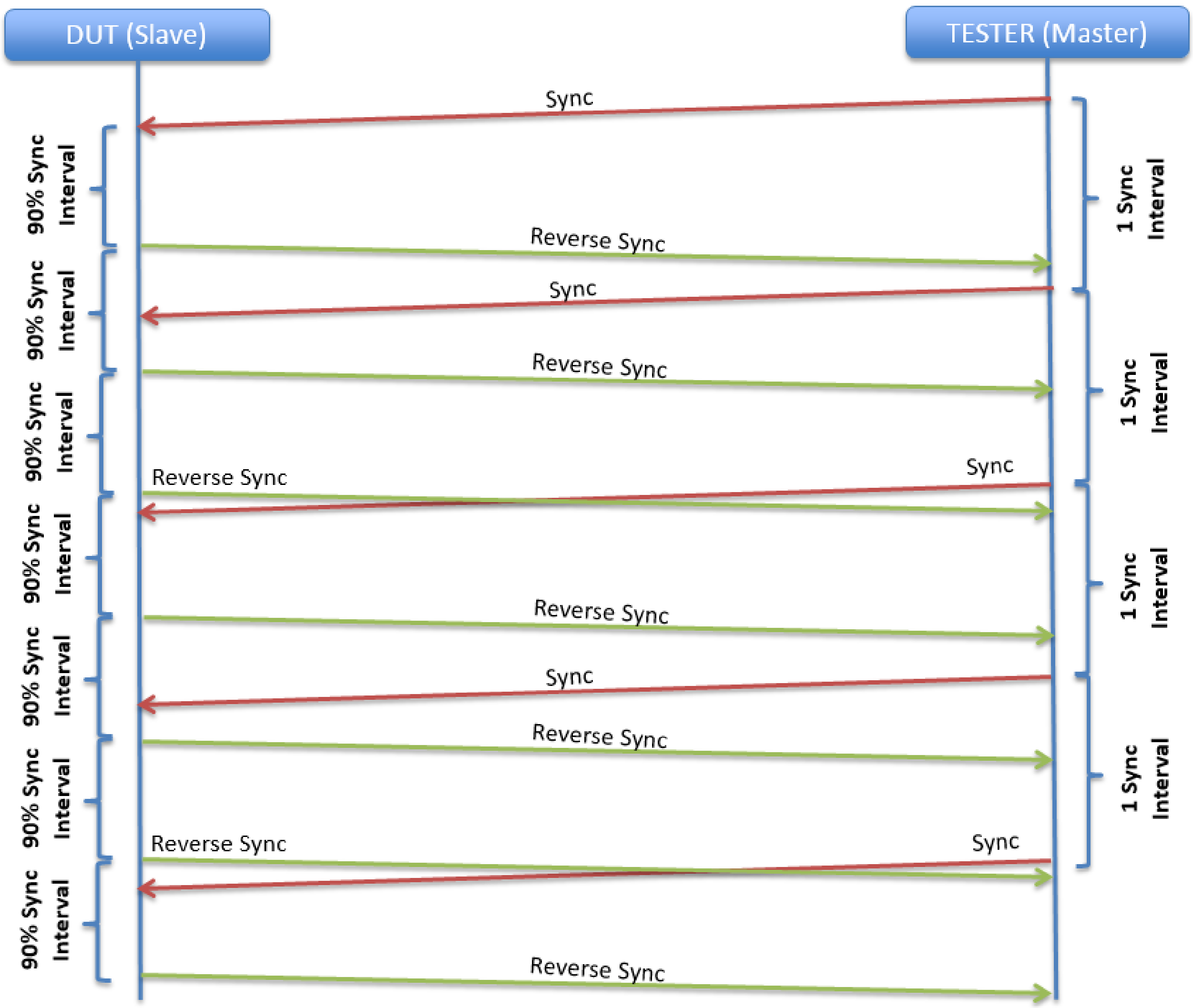

Figure 4: Reverse Sync message timing

6.4 Behavior specification

If the <**reverseSyncEnabled**> is true, then the Slave shall send Sync/Follow-Up messages with the origin timestamp indicating the current recovered time on the Slave. This enables the Tester to calculate the Time Error:

<Time Error> = -<offsetFromSlave>

The Slave shall respond to PdelayReq messages from the GM as specified in [gPTP 11.1.2]. This is required such that the Tester can compute and validate the <path delay> to be able to compute the <offsetFromSlave>.

6.5 Integrated Devices

There are cases, specifically in automotive, where the DUT consists of an integrated Slave implementation connected to a Bridge implementation, the Slave having no direct connection to outside devices.

On an integrated device, the bridge implementation shall forward the Reverse Sync/Follow-up from the Slave node under test and send them through its Slave port (that's generally externally visible), adjusting the correction field of the Sync or Follow Up message for the residence time of the Sync Message. The bridge shall track the residence time in the timebase of the domain that's being measured.

Note: If the bridge is a gPTP-rev capable bridge it just needs to act as a Bridge on the <reverseSyncDomain>. For a bridge that does not support gPTP-rev, a simple implementation would be to act as a simple 1588 Transparent Clock.

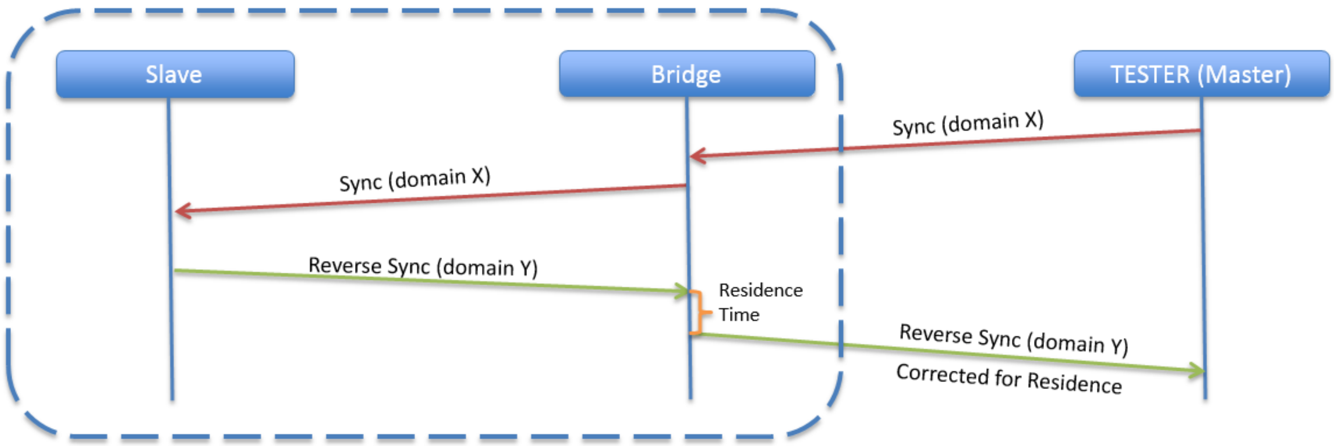

Figure 5: Generic PTP synchronization chain

7 Measuring gPTP clock accuracy using the Ingress Reporting Method

A description of the how the Ingress Reporting Method works can be found in Section 5.2.

The Slave shall either allow access to the <slaveRxSyncComputedDataRecord> data set through specific management methods or send the SLAVE_RX_SYNC_COMPUTED_DATA TLV, as defined in [1588-Slave-Monitoring]. In case of using a TLV, the TLV shall be attached to a 1588 Management message and sent to the unicast address of the host that enabled the collecting computed data.

The <offsetFromMaster> shall be considered as the value of measured Time Error. A device shall capture this value for each received Sync message (slaveEventMonitoringLoggingSkipRxSyncTiming shall be 1), and shall collect and report this data in such a way that the observation interval defined in Section 8.2 can be achieved.

8 Measurements for certification

8.1 Methods required

To Slave shall either:

1. Have a 1PPS output associated with the measured gPTP domain
2. Implement the Reverse Sync and Ingress reporting methods

If more than one measurement method is used, the measured clock quality must be met by all measurement methods. This gives confidence that all the methods are implemented correctly and that the reference implementation has a relatively stable oscillator.

Note: It is highly recommended that even if a device implements a 1PPS the other methods are also supported, especially the Ingress method which can be easily used for monitoring purposes in deployed networks.

8.2 Clock quality requirements

Unless otherwise specified by an Avnu CDS addressing specific industry requirements, as per Section 4.3, the Slave clock shall synchronize within 6 seconds to within +-80ns of the directly connected Master and shall maintain this clock quality over a 5 minute observation window. CDSs addressing specific industries can define different lock time, lock quality, observation interval under stable conditions and requirements for behavior when in holdover.