MEASURING 802.1AS SLAVE CLOCK ACCURACY

TECHNICAL GUIDE

Highlights

- Introduction to IEEE 1588 and IEEE 802.1AS
- Time correction in a bridge
- Path delay processing
- Importance of measuring slave clock accuracy
- Methods for measuring slave clock accuracy
- Using 1PPS outputs to validate slave time synchronization
- Ingress method
- Egress method
- Reverse sync method
- Summary: Advantages and disadvantages of each method

Learn more at: www.ixiacom.com

For more information on Ixia products, applications, or services,
please contact your local Ixia or Keysight Technologies office.
The complete list is available at: www.ixiacom.com/contact/info

Introduction to IEEE 1588 and IEEE 802.1AS

- IEEE 1588 & IEEE 802.1AS standards define how to synchronize time accurately between nodes on a network

- IEEE 1588 standardized the use of ***physical layer timestamps*** to compute network delays and define synchronization events
    - This achieves much higher timing accuracy than legacy protocols (such as NTP*) where timestamping is typically done in SW

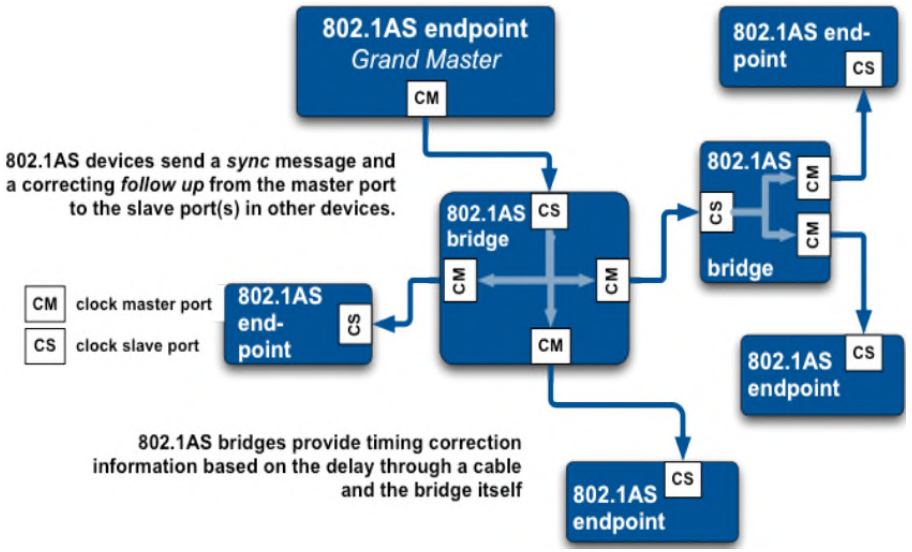
- IEEE 802.1AS is a 1588 “profile” with fewer options, and extended physical layer options
    - Faster clock locking
    - Allows for easier / lower cost implementation
    - Every device in the path (endpoints and relays) must support 802.1AS

* NTP refers to the “Network Time Protocol” defined by IETF RFC 5905.

ixia

IEEE 802.1AS: How it works

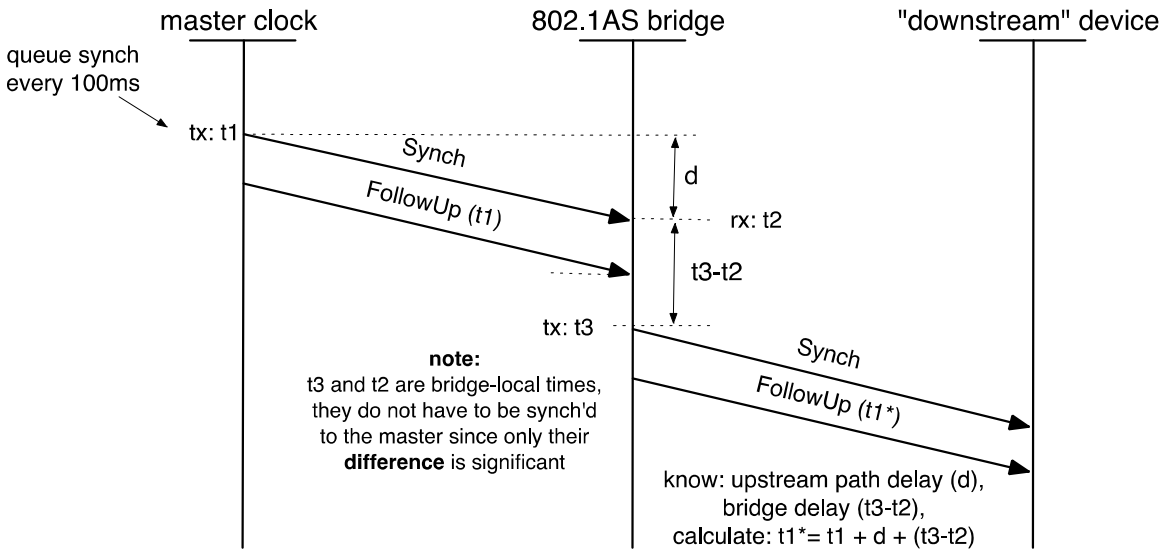
- Best Master Clock Algorithm (BMCA) is used to select a Grand Master
- Grand Master periodically sends the clock using sync messages
- Each Relay corrects timing information based on the delay through the cable as well as the delay through the relay itself
    - The Relay acts as a slave in the port in which it receives a clock and as a master on other ports
- Slave endpoints receive timing information and correct for the delay through the cable.



Picture courtesy of Michael Johas Teener, posted to Wikipedia. See https://en.wikipedia.org/wiki/File:Clock-Master.pdf

ixia

Time Correction in a Bridge



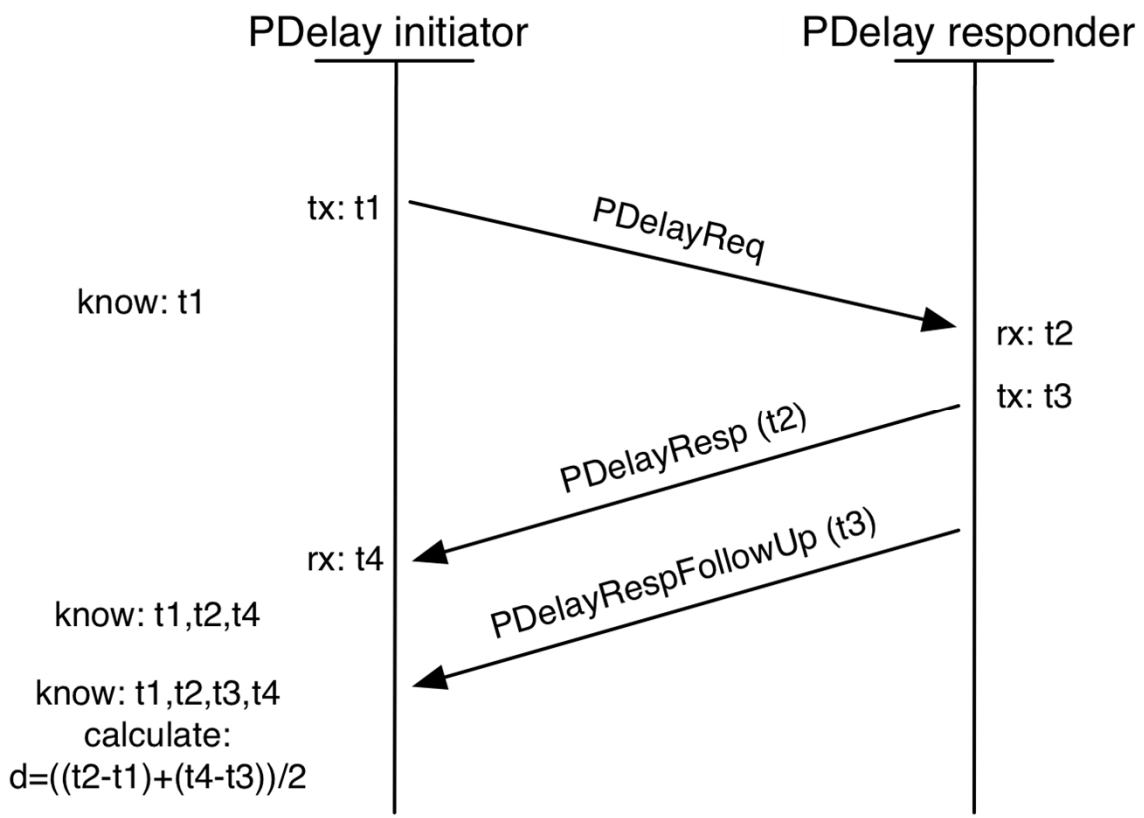
- Bridge Delays are now relatively constant, since they are just cable delays, without queues or buffers
  - 1588 calls this a “transparent clock”, required in 802.1AS

- A “correction field” in the FollowUp is incremented by the upstream delay and the residence time (t3-t2)
  - The correction field plus the precise origin timestamp plus the upstream delay is the correct time

Slide content courtesy of Michael Johas Teener, presented at Deterministic Ethernet Tutorial. See http://www.ieee802.org/802_tutorials/2012-11/8021-tutorial-final-v4.pdf

ixia

Path Delay Processing

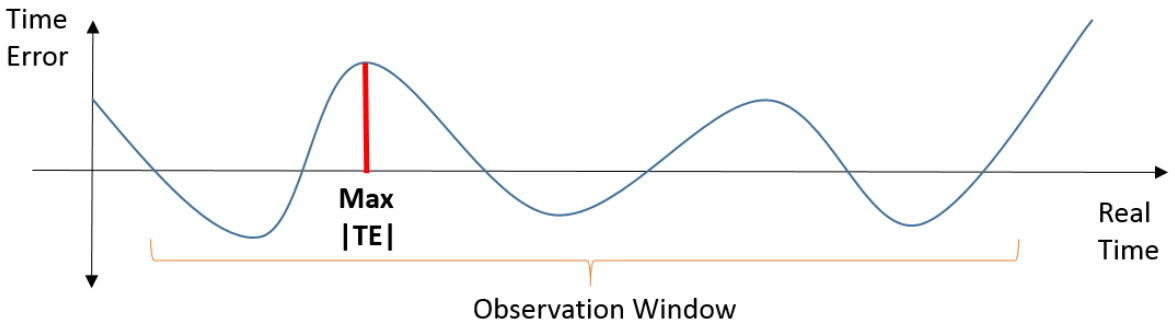


- Done infrequently since delays are stable

Slide content courtesy of Michael Johas Teener, presented at Deterministic Ethernet Tutorial. See http://www.ieee802.org/802_tutorials/2012-11/8021-tutorial-final-v4.pdf

ixia

Why measuring slave clock accuracy is important



- While the 802.1AS algorithm works well, imperfect implementations and/or hardware issues can cause the slave clock not to track the master clock

- It is important to verify & certify implementations to make sure they are correct.
  - Avnu has detailed test plans to validate 802.1AS
  - Typically, PTP or gPTP implementations that have not been validated will have conformance issues

- To validate the clock accuracy on a slave device , we look at the Time Error (TE), which is the difference between the test reference (802.1AS master) and the DUT (802.1AS slave).
  - This needs to be sampled over a long period of time to find out the MAX TE.

ixia

Different methods for measuring the slave Clock accuracy

1. Compare 1 Pulse Per Second (1PPS) or similar output between the master and slave
2. “Ingress method” where a slave clock reports the time error it is seeing each time it receives a sync path
3. “Egress method” where the slave provides the PTP time of messages that it sent, which can be compared to the time these messages are received by the master
4. “Reverse Sync method” where the slave sends sync messages back to the master to validate the timing
   - This is essentially a variation on the Egress method where the message used is a Sync message and there is no need for additional messaging to identify the PTP time when the message was sent.

The details, pros, and cons of each method are detailed in the next slides.

It is recommended to implement test modes containing at least one of the above methods in each slave
   - Implementing multiple methods allows cross-validation of the measurement methods

See Avnu recommendations at http://avnu.org/wp-content/uploads/2014/05/Avnu-Testability-802.1AS-Recovered-Clock-Quality-Measurement-1.0_Approved-for-Public-Release.pdf

ixia

Using 1PPS outputs to validate slave time synchronization

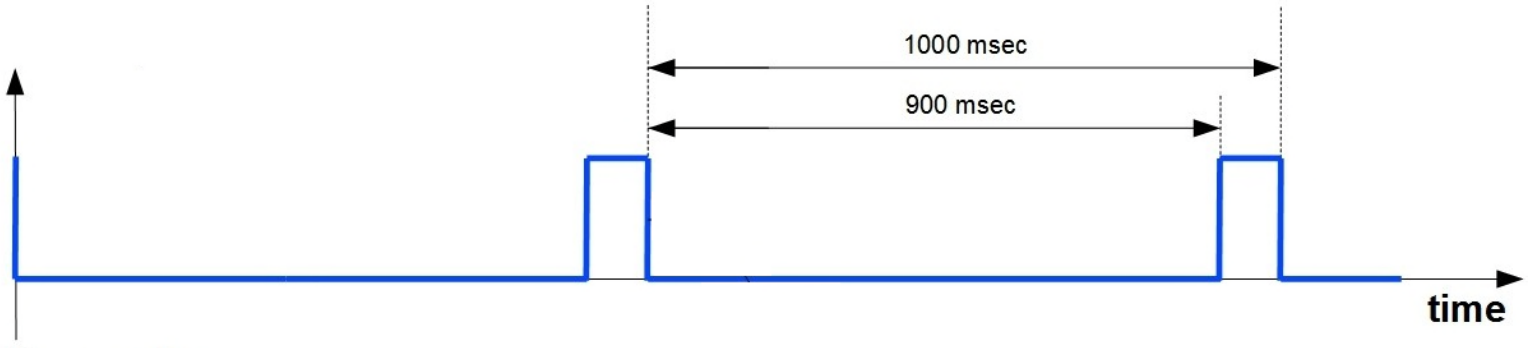
- Both master and slave output a pulse that starts on each second boundary of the PTP clock
- An oscilloscope is used to compare the time of the two outputs
  - each measurement shows a phase difference in the clock
  - Using multiple measurements, maximum time error and jitter can be ascertained.

Advantages of 1PPS method:

- Known industry standard which already exists on many devices.
- Only an oscilloscope is needed to make this test.

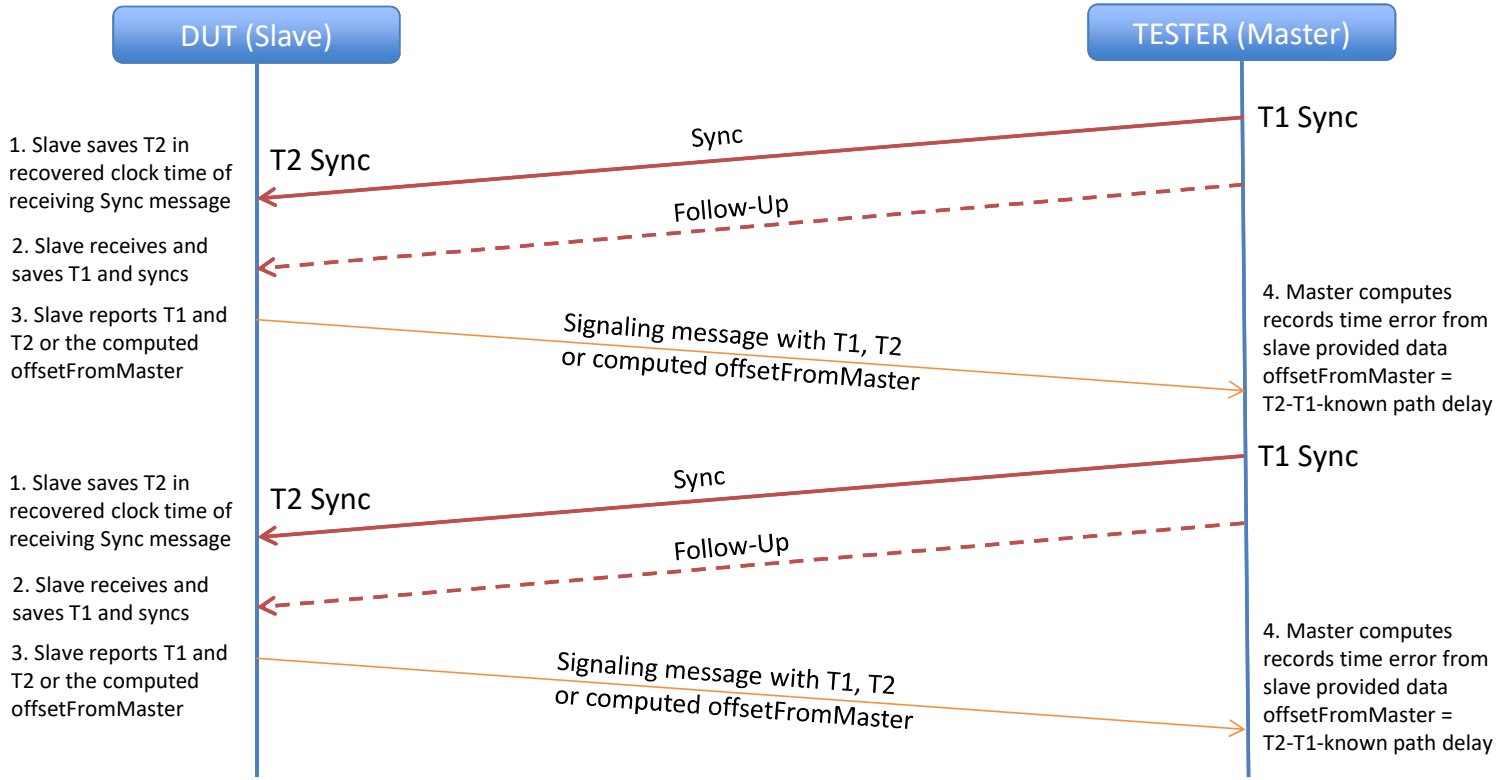
Shortcomings of 1PPS method:

- Implementation of the 1PPS itself is sometimes imperfect, causing a fixed phase offset or jitter.
- Some devices can't easily support having a 1PPS output (i.e. devices with mechanical constraints that prevent this).
- The devices under test have to be relatively close together
- Not practical for large-scale testing (such as validating time synchronization in an entire sports arena or factory).



ixia

Ingress method



Ingress method relies on Slave to declare it's error at the moment of receiving a Sync message
(Example: *"When you sent me the last Sync message, my recovered clock when receiving it was X"*)
Two flavors defined in 1588, one that reports error, and one that reports T1 and T2 – **same principle!**

ixia

Ingress method pros/cons

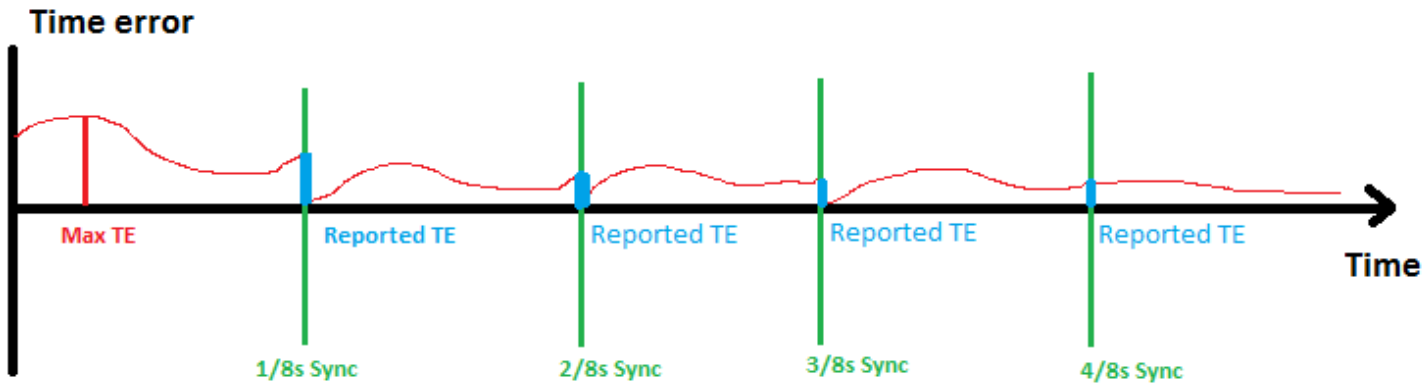
Advantages of Ingress method:

- Reporting can be done in-band with 1588 proposed TLVs attached on signaling message or simply locally logged/stored to be accessed remotely through YANG (ideal for live deployments)
- Does not require any extra capability on the Slave except sending TLVs* /storing the data

Shortcomings of Ingress method:

- **Sampling of time error is dependent on the timing of incoming Sync messages**
  It might not reflect the actual maximum time error between two incoming Sync messages
- **The Slave can report a value that is not the actual one**
  This could be due to a fixed phase offset, coding error, or due to intentionally returning values that make the device seem better (very hard to detect, example in backup slides)

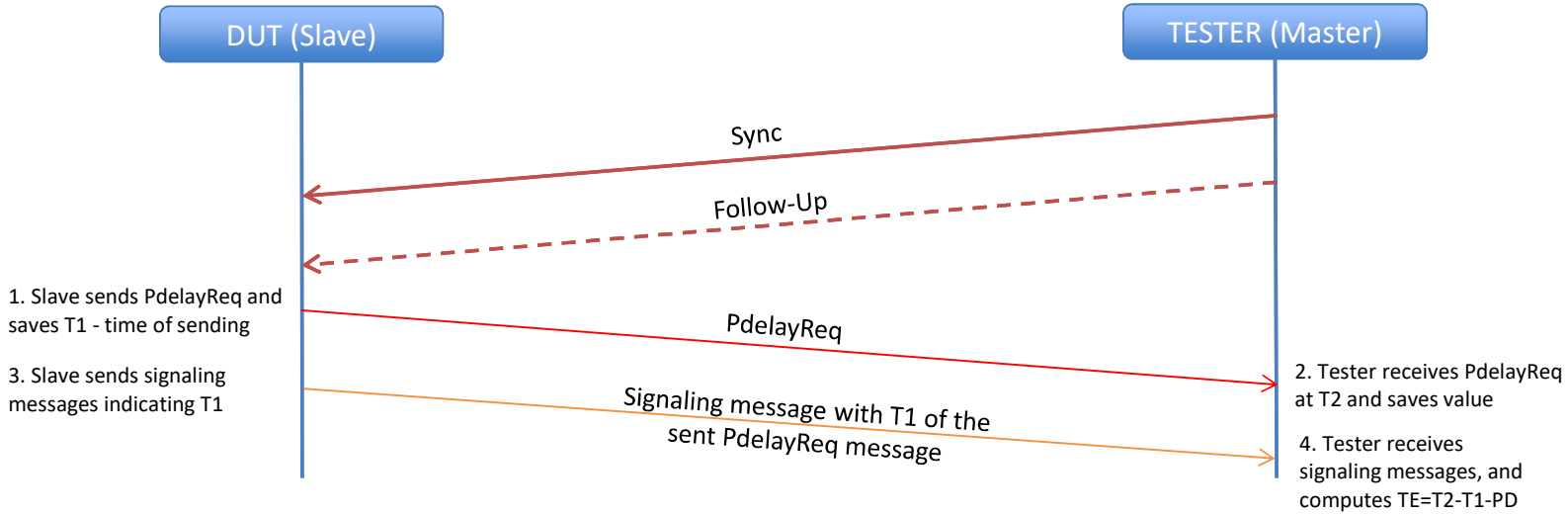
=> the Ingress method is not adequate for certification testing



* A TLV (or Type Length Value) is a message format defined in IEEE Std. 1588.

ixia

Egress method



Egress method - Slave reports it's recovered time at moment when an event message is sent
(Example: *"Slave: the last PdelayReq I sent was at global gPTP time X as per my recovered clock"*)

Advantages of Egress method:

- **Tester evaluates the error, it does not rely on the Slave to report it**
- Reporting can be done in-band with 1588 proposed TLVs attached on signaling message
- Time of reporting is decoupled from the receive time of the incoming Sync messages

Shortcomings of Egress method:

- **Complicated to use in deployed networks – needs support from directly connected Master of DUT**

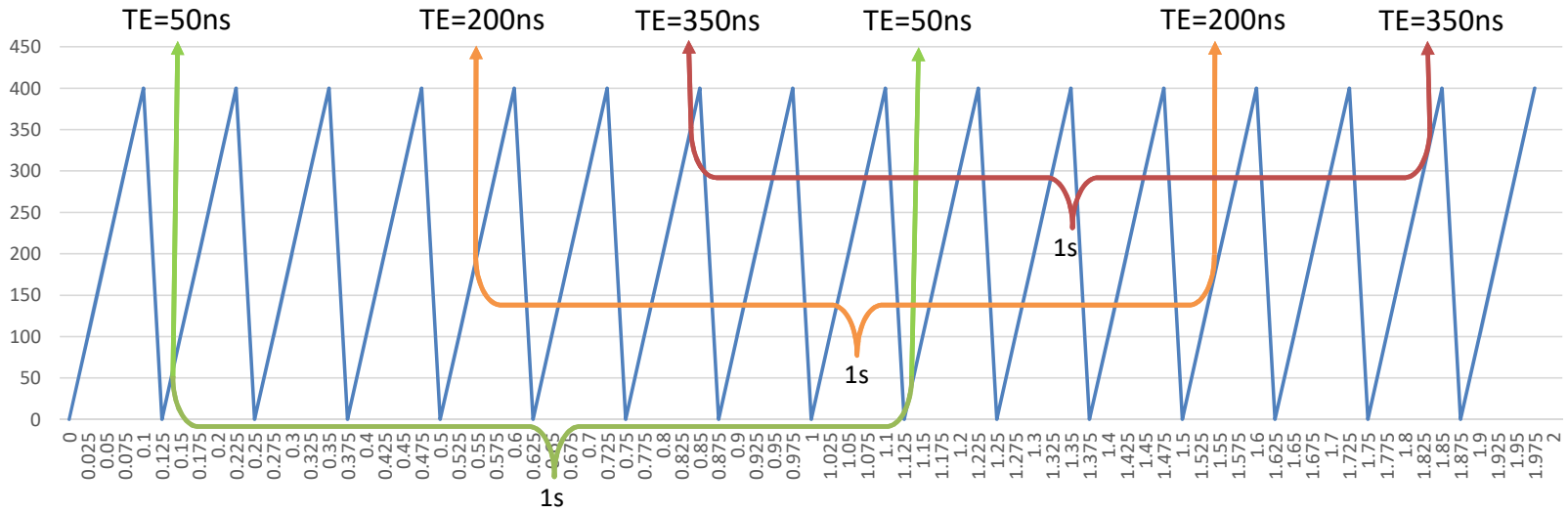
ixia

Problems in using Egress method for gPTP

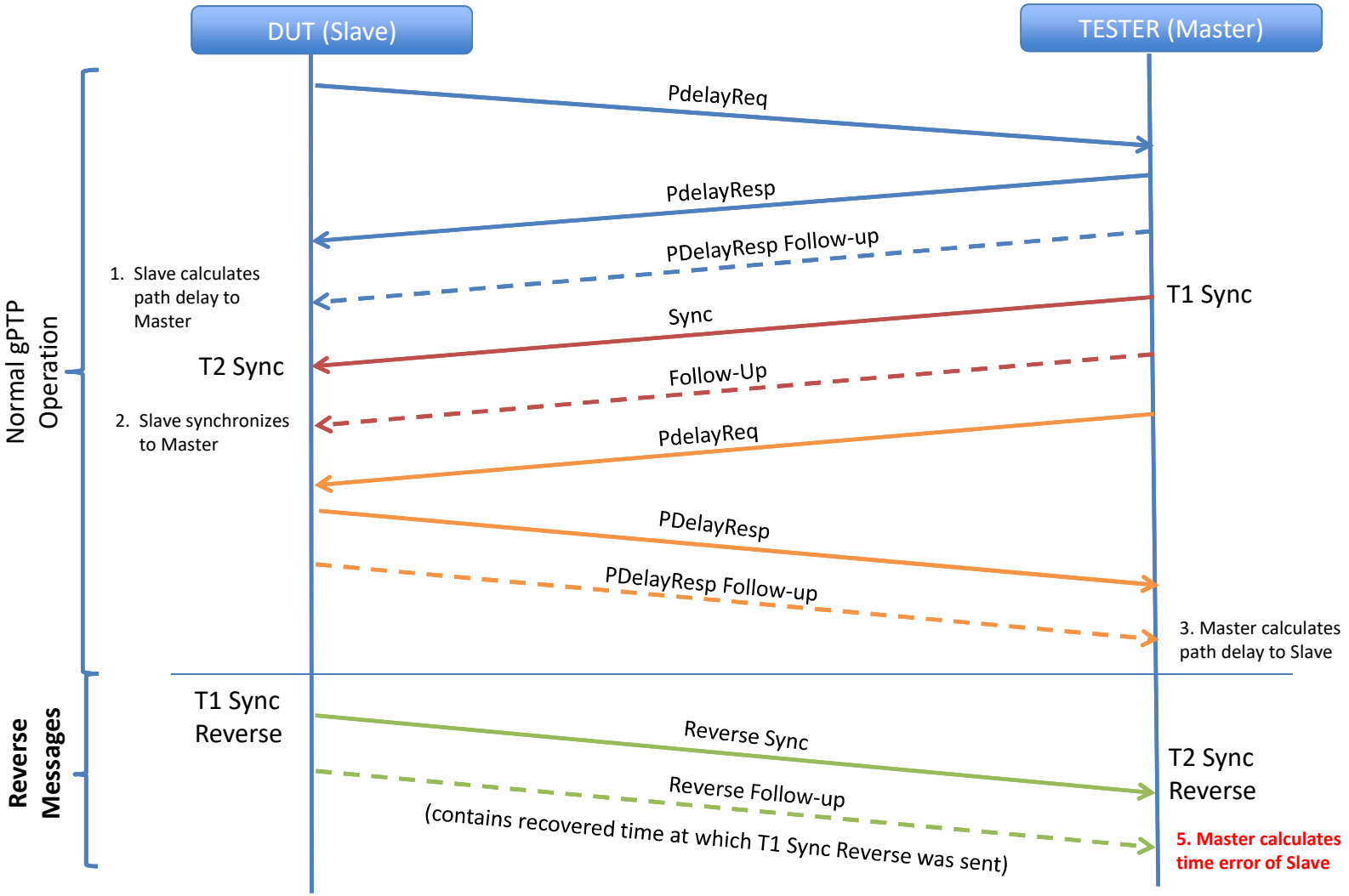
For gPTP common sense is that reporting would be based on the sent PdelayReq event messages. If PdelayReq messages are timed immediately or close to receiving a Sync message (when correction of the clock is made), they might not reveal the actual time error the device is experiencing.

1. Consider a Slave clock drifting 400ns between each 2 Sync messages (125ms interval)
2. AND not properly implementing syntonization (algorithm or rate ratio calculus)
3. The time error function looks similar to a saw (resets to ~0 at each Sync received)
4. If event PdelayReq is sent close after Sync is received measured error is lower than one
5. Because PdelayReq interval is a multiple of the Sync interval, this would happen at every time

Not just theory, this was experienced with real devices in lab testing !



Reverse Sync Method

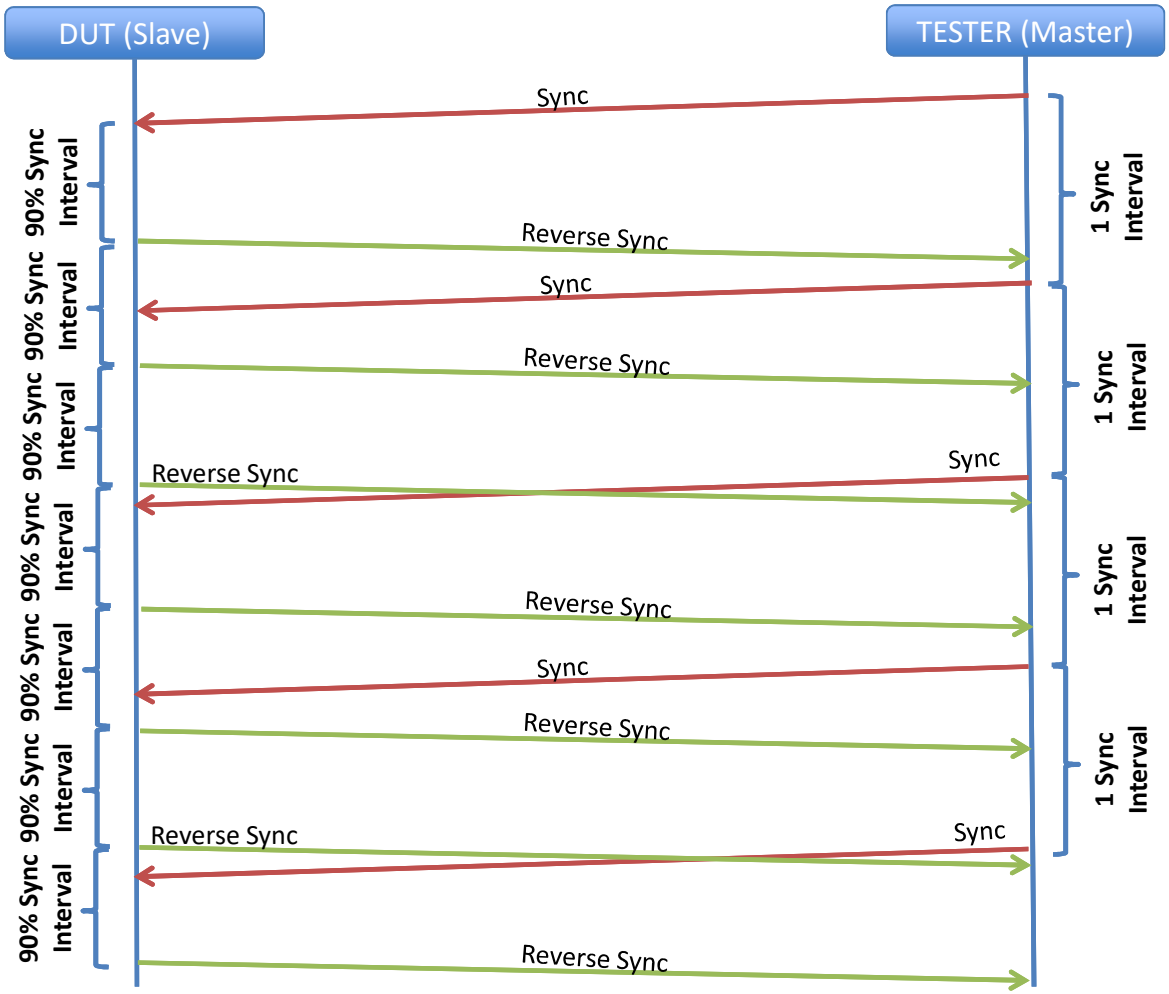


The Reverse Sync Method

1. The slave synchronizes it's time to the master
2. The Reverse Sync is enabled on the slave (test mode)
3. The Master (tester) calculates the path delay so that it can account for this when validating the sync messages
4. The slave sends Sync messages in the reverse direction (using a different domain)
5. **Observations: the TESTER (Master) will calculate the time error in the recovered clock of the DUT (Slave)**

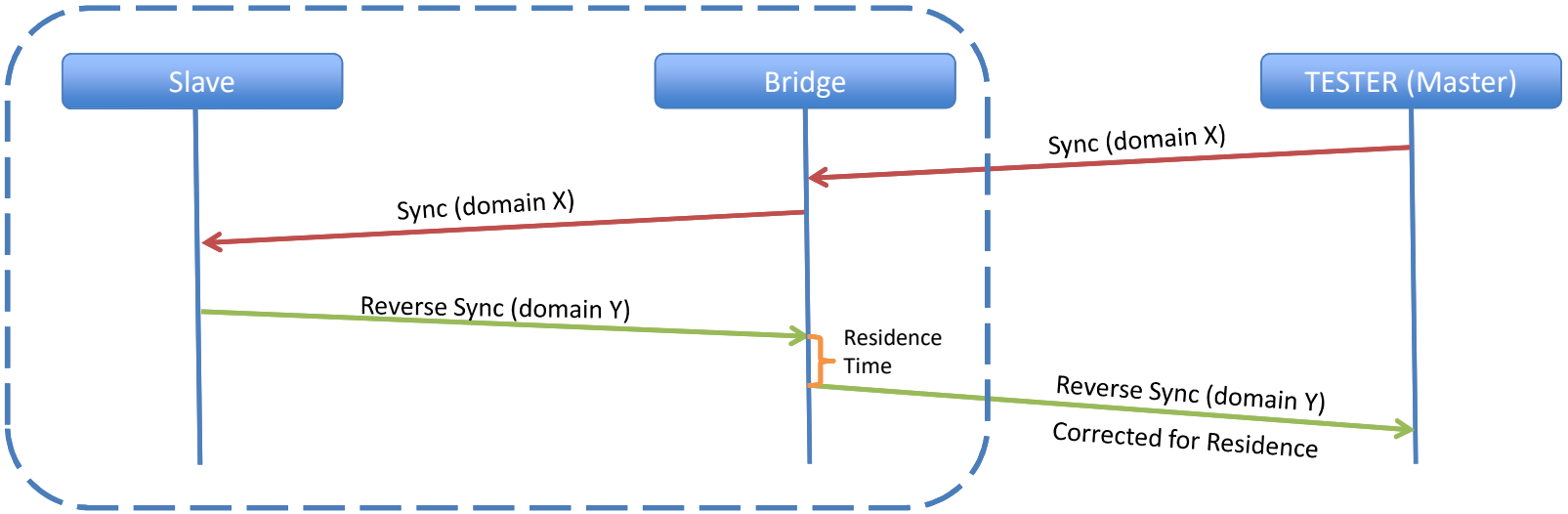
ixia

Adjust Reverse Sync rate for better sampling



Advantages of Reverse Sync for Integrated Devices

- On integrated devices that have an integrated Slave and Bridge, there is no observability of the Slave Pdelay mechanism
- Using Sync messages as Egress, the Bridge can correct for the “residence time” of the Reverse Sync the same way as they would normally do for the standard gPTP instance
  - The bridge just forwards the Reverse Sync (with timing corrections) to its Master port to the Tester
  - This is effectively using functionality defined in 802.1AS-rev, using one domain for Synchronization and a different domain to do the recovered clock measurement
- If proper time is set in Reverse Sync preciseOriginTimestamp there is no need for additional egress TLV
- **Implementation is straight forward – can be done from existing “Send Sync” code**



DUT: Integrated ECU with Slave + Bridge

ixia

Summary

It is important to test 802.1AS slave clock accuracy

Multiple methods are now defined allowing slave clock accuracy to be measured including

- 1PPS
- Ingress Method
- Egress Method
- Reverse Sync Method

It is recommended to implement multiple methods to cross-check results

ixia

Summary: Advantages & Disadvantages of Each Method

Method	Advantages	Disadvantages
1PPS outputs	• Known industry standard • Only an oscilloscope is needed to make this test	• Hard to validate implementation • Not available on all devices • DUTs must be close together • Not practical for large-scale testing
Ingress Method	• Reporting can be done in-band • No need for extra capabilities on the Slave	• Sampling of time error is dependent on the timing of incoming Sync messages • The Slave can report a value that is not the actual one
Egress Method	• Tester evaluates the error, it does not rely on the Slave to report it • Reporting can be done in-band • Time of reporting is decoupled from the receive time of the incoming Sync messages	• Complicated to use in deployed networks – needs support from directly connected Master of DUT

Reverse Sync	• All advantages of egress method, plus
	  • Can be used on integrated devices
	  • No need for additional egress TLV
	  • Implementation is straight forward
		• Requires Sync generation on Slave