Passive and Active Probing of Slave Timing Error for 802.1AS 2015-11-11

Bob Noseworthy, UNH-IOL
Opher Ronen, ADVA Optical
Bogdan Tenea, IXIA
Alon Regev, IXIA

Supporters

- The following individuals have previously expressed support for the proposal:
  - Gordon Bechtel, Harman Connected Services
  - Kevin Stanton, Intel

Agenda

- Goals
- Proposal Overview
- Work To Date
- Passive Monitoring Proposal
  - Slave_Rx_Sync_Timing_Data TLV
  - Management needs
- Active Monitoring Proposal
  - Slave_Tx_Event_Timestamps TLV
  - Management needs

Goals

- Real-time, network-based, application-independent feedback and monitoring of timing error in a TSN.
  - Slave devices are increasingly common with no access to a test signal tied to the recovered clock, eg: a 1 pulse per second (1PPS) or similar.
  - Shared methodology with IEEE 1588 (profile agnostic)
- Enhanced monitoring, diagnostic and test abilities to ensure TSN components meet or exceed the system's requirements.

Future goals

- Enabled by the proposed solution but not fully explored in this proposal:
  - Scalable solutions that allow for distributed analysis of critical TSN elements while minimizing impact on overall system performance.
  - Alerting mechanisms for applications relying on precise time to be informed of system faults.
  - Security mechanisms to protect against malicious use of the feedback and monitoring system.

Proposal Overview

- Two new slave timing error monitoring mechanisms
  - A "Passive" monitoring mechanism:
    - Requires no additional capabilities outside of the slave
    - Provide data for external assessment of slave's timing error based on received Sync information
  - An "Active" monitoring mechanism:
    - Requires monitoring by the slave's link-partner to note when a slave-transmitted event message is sent vs the link-partner's sense of PTP time
    - Provide data for external assessment of a slave's timing error based on transmission of Event message's (eg: Sync, Pdelay_Req and/or Delay_Req

Work To Date

- Initial proposal stems from "TLV for supplying slave generated timestamps"
  - Opher Ronen (ADVA Optical) proposal, v2 presented Oct 29, 2015 https://ieee-sa.centraldesktop.com/1588/file/40649158/
  - Proposed SLAVE_ACQUIRED_SYNC_TIMING_DATA TLV very similar to proposed "Passive" proposal
  - Proposed SLAVE_ACQUIRED_DELAY_REQ_TIMING_DATA TLV is very similar to proposed "Active" proposal
- "Active" proposal also embodies input from Ixia regarding "reverse-sync"
- Desire shared between the many in 1588 and 802.1AS to have one common mechanism that can be leveraged for most/all profiles.
  - 1588 Management committee aware of this 802.1AS effort
  - Ultimately, desire is to utilize identical/similar TLVs in 1588 once adopted there, 802.1AS must define its own TLV for now
  - 1588 effort will complete later than 802.1AS-Rev

Passive Monitoring Proposal

- Define in 802.1AS-rev a normative but optional Annex for a probing mechanism to record slave timing error correlated to received Sync events, recording:
  - Received Sync sequenceID
  - Received Sync originTimestamp (as appropriate for 1-step or 2-step)
  - Received Sync (if 1Step) or Followup (if 2step) correctionField value
  - Current meanPathDelay for the slave port (neighborPropDelay in .1AS terms)
  - Calculated syncEventIngressTimestamp in gPTP timebase
- Slave may:
  - Log and record this data for subsequent retrieval (truly “passive”) or,
  - Provide this date out-of-band, or
  - Provide this data in-band on the PTP network using the proposed TLV structure appended to a Signaling message (targetPortIdentity of 0xFF), or
  - Provide this data to a remote network monitoring station via a TBD unicast message. (Will not be addressed in this proposal)

SLAVE_RX_SYNC_TIMING_DATA TLV

- Ultimately should be that defined by 1588-rev
  - eg: Standard TLV in range 000A – 1FFF
- Until 1588-rev adopts a TLV, suggest use of the standard organization extension TLV
  - organizationId = 00-80-C2
  - organizationSubType = 4 (?)
    - message interval request TLV is 2, gPTP capable is 3
    - Space is 3 octets, could partition and assign values non-consecutively

tlvType (2)
lengthField (2)
organizationId (3)
organizationSubType (3)
sourcePortIdentity (10)
numberOfSyncData (2)
sequenceId[1] (2)
correctionField[1] (8)
syncOriginTimestamp[1] (10)
meanPathDelayValue[1] (8)
syncEventIngressTimestamp [1] (10)
….
sequenceId[N] (2)
correctionField[N] (8)
syncOriginTimestamp[N] (10)
meanPathDelayValue[N] (8)
syncEventIngressTimestamp [N] (10)

SLAVE_RX_SYNC_TIMING_DATA TLV fields

Note: Most of this is as proposed by Opher Ronen (except for organizationId/subtype and meanPathDelayValue)

tlvType The value of tlvType shall be <value assigned for SLAVE_RX_SYNC_TIMING_DATA – non forwardable TLV>

lengthField The value of the lengthField is 16+N*20 where N is the number of Sync messages for which information is included.

organizationId 00-80-C2 as discussed on previous slide. **organizationSubType** (4) as discussed on previous slide

numberOfSyncData The value of the numberOfSyncData shall be the number of Sync messages for which information is included in this TLV.

sourcePortIdentity The value of the sourcePortIdentity shall be the value of sourcePortIdentity field of Sync messages for which slave received sync timing data is conveyed.

The following values shall appear for each Sync message for which slave received sync timing data is conveyed. An index [i] is used to indicate the value is for the i^{th} message which for which data is included in this TLV.

sequenceId[i] The value of sequenceId[i] shall be the value of the sequenceId field of the i^{th} Sync message for which the slave acquired timing data is conveyed. **< This value is copied from the received Sync message >**

correctionField[i] The value of correctionField[i] shall be the aggregate value of the correctionField fields used within the calculation of the offsetFromMaster for the i^{th} Sync message for which the slave acquired timing data is conveyed (see 11.2 of IEEE 1588-2008). This value is therefore sum of the correctionField of the received Sync message and the correctionField within the FollowUp message if the received Sync message had the twoStepFlag bit set to TRUE. **< This is copied from the received Sync or FollowUp message >**

syncOriginTimestamp[i] The value of syncOriginTimestamp[i] shall be the value of the syncEventEgressTimestamp value generated by the port sending the Sync message for the i^{th} Sync message for which the slave acquired timing data is conveyed (see 9.5.9). If the twoStepFlag bit was set to TRUE the **syncOriginTimestamp** will be the preciseOriginTimestamp from the correlated follow-up message, otherwise it will be the originTimestamp from the received Sync message. ***<Note: This is in the timebase of the PTP domain >***

meanPathDelayValue[i] The value of meanPathDelay (1588) or neighborPropDelay (802.1AS) value at the time the i^{th} Sync message was received. This value should be that value used by the slave to compute the offsetFromMaster with the corresponding Sync message. This value is not anticipated to change rapidly. If a value is not available, the field will be set to zero (0).

syncEventIngressTimestamp[i] The value of syncEventIngressTimestamp[i] shall be the value of the syncEventIngressTimestamp acquired for the i^{th} Sync message for which the slave acquired timing data is conveyed. ***< Note: This is in the timebase of the PTP domain, any necessary calculations to provide the timestamp in the PTP domain must be performed. – eg: cross-timestamp from HW timestamper's local clock to PTP clock >***

Slave Logged or TLV conveyed data

- Slave conveys information for received Sync message
  - **Does not need to be for all received Sync messages**
  - Computation time remains unbounded
- <offsetFromMaster> computable from data provided (see 11.2 of IEEE 1588-2008)
  - per 11.2: <offsetFromMaster> = <syncEventIngressTimestamp> - <preciseOriginTimestamp> - meanPathDelay – Sync correctionField - Follow_Up correctionField
  - meanPathDelay can be ignored
    - static offset error values are not observable by this observation technique alone
  - For given clock (1step or 2step) only one correctionField need be examined
- In gPTP correctionField uses rateRatio

sequenceId[1] (2)
correctionField[1] (8)
syncOriginTimestamp[1] (10)
meanPathDelay[1] (8)
syncEventIngressTimestamp [1] (10)
….
sequenceId[N] (2)
correctionField[N] (8)
syncOriginTimestamp[N] (10)
meanPathDelay[1] (8)
syncEventIngressTimestamp [N] (10)

1588 E2E related Note: The proposed TLV conveys “T2” (Sync receipt time) useful for slave assessment in an E2E path delay environment

correctionField in 802.1AS

The Sync/FollowUp correctionField in gPTP is corrected to the GM's timebase via the rateRatio

As a reminder: in a bridge,

- 11.2.14.2.3 setFollowUp() sets the correctionField to rateRatio * (syncEventEgressTimestamp – upstreamTxTime)
- 11.2.13.2.1 setMDSyncReceive(f) sets upstreamTxTime to syncEventIngressTimestamp minus neighborPropDelay divided by neighborRateRatio minus delayAsymmetry divided by rateRatio

offsetFromMaster

- In current standards, the time that this value (stored in the current data set) is updated is not tied to the receipt of a specific Sync event message
  - Hence not measurable or testable
  - Proposed TLV allows offsetFromMaster to be computed for a given Sync event
- Ability to calculate the offsetFromMaster error value using a specific Sync event message of the slave's choosing allows for external verification of slave's tracking based on stimulation from a master port
  - Master port may be directly attached test gear
  - Data collected in-field for remote performance monitoring
- Not perfect, physical clock access can be better (eg: 1PPS) in some cases, but in the absence of such access, proposal provides some measureable and verifiable data on slave performance and slave clock servo behavior
  - Lab testing can establish if the slave reacts to stimulus in a reasonable manner
  - Even 1PPS access can (and often is) imperfect, as static errors (~10s of ns) occur for improperly accounted for delays

Passive Monitoring Management Needs

- Enable/disable logging of slave rx sync timing
- Enable/disable TLV in-band transmission
- Define mechanism to alter reporting rate.
  - Eg: probingSkip of 0 = report every Sync message, probing Skip of 1 = report every other Sync message, etc.
  - Much like message interval request TLV, the slave may not support all requested reporting rates.

Passive Monitoring Proposal Summary

Outstanding TBDs

- TLV type value selection
- Management mechanism definition

Known strengths

- **Lightweight** addition to **resource constrained slaves**
- Inclusion of meanPathDelay value allows discovery of some static errors
- Allows in lab and in field monitoring and remote monitoring of dynamic errors with no added network resources (pending unicast or similar additional definition)

Known weaknesses

- All static offset errors would not be measurable by the slave (the slave doesn't know what it doesn't know, eg: latency or asymmetry correction errors)
- Active probing (slave transmission of event messages) would be required to observe static offset errors in port latency

Active Monitoring Proposal

- Define in 802.1AS-rev a normative but optional Annex for an active probing mechanism to externally assess slave timing error correlated to transmitted Events messages, recording:
  - Transmitted Event message sequenceID
  - Transmitted eventOriginTimestamp in gPTP timebase
- Slave may:
  - Log and record this data for subsequent retrieval, or,
  - Provide this date out-of-band, or
  - Provide this data in-band on the PTP network using the proposed TLV structure appended to a Signaling message (targetPortIdentity of 0xFF)
- Attached Device with Master port may:
  - Send information on slave performance directed to a network monitoring station via a TBD unicast message. (Will not be addressed in this proposal)

Active Monitoring Proposal Overview

- A slave device may emit event messages that could be timestamped in the gPTP clock domain and this information provided for monitoring/test
- Event messages: Sync, Pdelay_Request, Pdelay_Response, Delay_Req.
  - Per 802.1AS, slave's WILL send Pdelay_Request and may send Pdelay_Response
    - "Typical" 802.1AS Slave would not need to timestamp the transmission in the gPTP clock domain, but this proposal would require the event transmission timestamp to be conveyed in the gPTP clock domain
  - Slave may send a Sync (aka "Reverse Sync" as referred to by some),
  - A gPTP slave would not have cause to send a Delay_Req, but could, similar to the Sync case above,
    - to expose Slave's clock at higher observation sampling rates
- Note, in the form of a Delay_Req, this proposal degenerates to the IEEE 1588-Rev proposal from Opher Ronen for a SLAVE_ACQUIRED_DELAYREQ_TIMING_DATA TLV

SLAVE_TX_EVENT_TIMESTAMPS TLV

- Report event timestamp in new TLV attached to Signaling message
  - **Does not need to be for all transmitted Event messages**
  - **Link partner is expected to be test gear or device with probing support to measure the timing error of gPTP time vs attached slave's reported gPTP time**
  - organizationSubType = 5(?)
  - eventMessageType = per .1AS table 11-3, or 1588-2008 table 19
  - eventOriginTimestamp – time in gPTP timebase of event transmission
  - Other fields similar to SLAVE_RX_SYNC_TIMING_DATA TLV

tlvType (2)
lengthField (2)
organizationId (3)
organizationSubType (3)
sourcePortIdentity (10)
eventMessageType (1)
numberOfEventData (2)
sequenceId[1] (2)
eventOriginTimestamp[1] (10)
….
sequenceId[N] (2)
eventOriginTimestamp[N] (10)

Active Monitoring Advantages

- Event message sent by slave with corresponding timestamp in gPTP time domain
- Rate of event message transmission could be increased when supported and desired to better measure the slave's clock
  - Impact of sending load-generating packets such as Pdelay_Req must be considered. Might send to non-domain 0, or send a packet that would be ignored by a gPTP device (eg: Delay_Req) if slave hw timestamping is supported.
- May not require new slave hardware capability
  - especially if slave supports a path delay mechanism (eg: Pdelay as is required for gPTP), or already supports Master port operation (eg: Sync generation)
- For 802.1AS, outside of lab qualification, in-field "active" measurements may not be necessary (egress vs ingress error not likely to change – aside from pluggable module situation…)
  - 1588 Solutions using E2E may benefit from in-field 'active' monitoring of Delay_Req event messages

Active Monitoring Disadvantages

- May require transmission of event messages not required by slave application
  - Generically: Not all slave-only devices need path-delay info
    - gPTP expects all slaves issue Pdelay_Req
    - However, it is desirable to have one probing solution suitable for all 1588 profiles
  - Concerns of a Slave port sending message types that it should typically not send (eg: Sync)
- Link partner must make measurement
- Requires computation of event hardware timestamp to be cross-timestamped into gPTP clock domain
  - Added computation burden on slave device

Active Management Needs

- Enable/disable logging of slave event timestamp data
- Enable/disable TLV in-band transmission
- Configure desired event message (Sync, Pdelay_req, Pdelay_resp, Delay_Req)
  - Slave need not honor all configuration requests
- Define mechanism to alter reporting rate.
  - Eg: probingSkip of 0 = report every Event message, probing Skip of 1 = report every other Event message, etc.
  - Much like message interval request TLV, the slave may not support all requested reporting rates.

Active Monitoring Proposal Summary

Outstanding TBDs

- TLV type value selection for SLAVE_TX_EVENT_TIMESTAMPS
- Management mechanism definition

Known strengths

- Can observe static offset errors unknown to the slave due to ingress/egress PHY latency timestamp errors

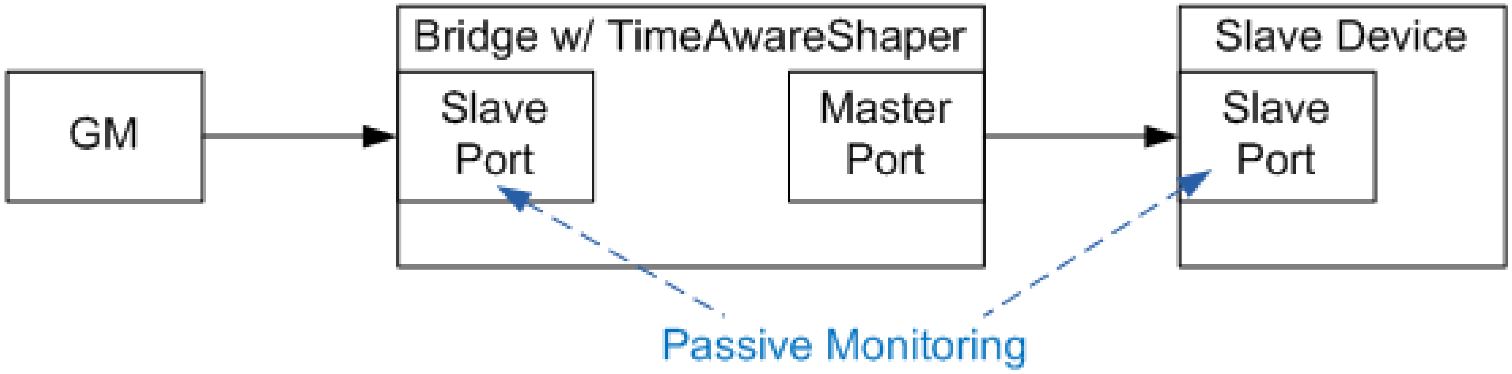
Known weaknesses

- Requires direct link partner to monitor event message timing correlated to its sense of gPTP time
  - Requires link partner capability for in-field / distributed monitoring and reporting

BACKUP / ADDITIONAL SLIDES

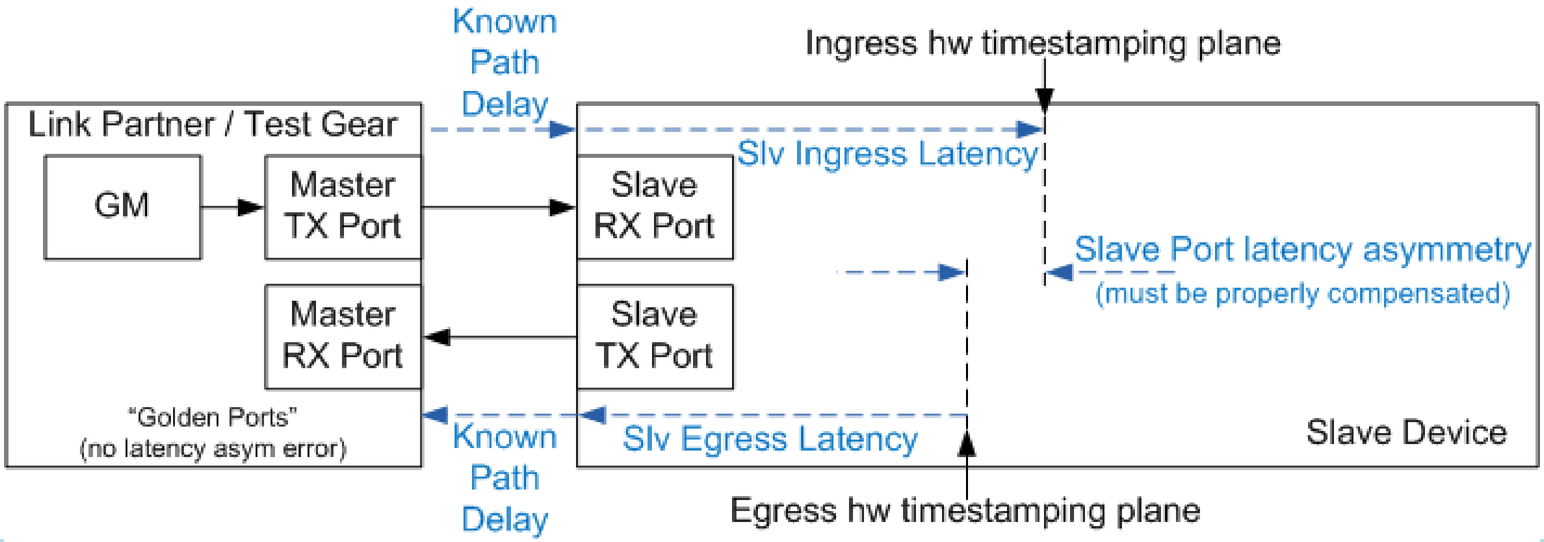
Use Case Examples: Bridge & End

- TSN monitoring at slave port – no matter where the slave port is (Endstation or Bridge)
  - Valuable for Time Aware Shaper validation

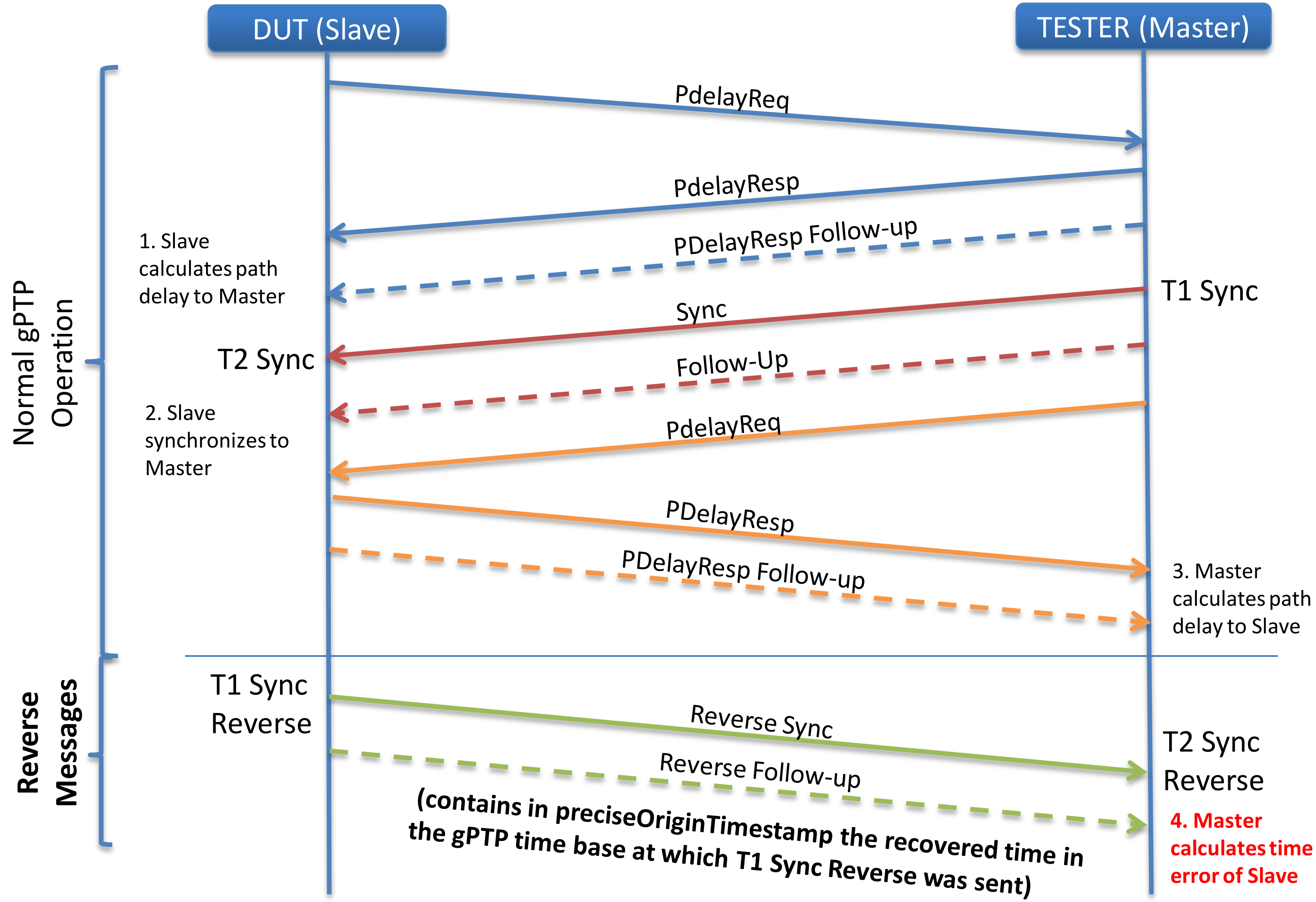


Use Case Examples: Active

- Typical expected use of "Active" Monitoring
  - Direct link to known good test gear on known path
  - Combined with "Passive" monitoring to assess timestamping plane asymmetry correction error



Use Case Example: Reverse Sync method



Courtesy: Bogdan Tenea

Use Case: Reverse Sync details

- Advantages over TLV method:
  - Can be used in the same way for both P2P and E2E (for E2E the tester would need to know the pathDelay used by the Slave in it's offset calculations – if stable can be taken from the passive method)
  - Reverse Sync messages can be sent out at independent timing from normal PTP message exchange
    - this may reveal otherwise undetectable dynamic errors
    - enables usage of faster reverse rates that enable faster convergence for measurement
  - Does not need any extra messages/TLV to be sent out – a tester can use only Reverse Sync provided info to do the time error measurement
- Disadvantages:
  - Potentially confusing for real world deployments, risk of Master clocks adjusting their clock to the Reverse Sync
    - should not happen if port is Master role;
    - risk can be mitigated by using a reserved domain in the 128-255 range;
  - Provides no insight on erroneous E2E path delay calculation
    - could be solved by a Reverse Delay Request sent out by the Master

References

- IEEE 1588-2008
- IEEE 802.1AS-2011
- O.Ronen, “TLV for supplying slave generated timestamps”, Oct 2015
  - https://ieee-sa.centraldesktop.com/1588/file/40649158/
- O.Ronen, “Timing packet triggered observations of a PTP slave port's timescale ("Slave port support for packet based external probing")”, Mar 2013
  - https://ieee-sa.centraldesktop.com/1588/file/29081938/
- O.Ronen, "Synchronization Monitoring in IEEE1588 Synchronization Networks“, International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication 2013 (ISPCS 2013) Lemgo, Germany, Sep 2013
  - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6644770