Validating and Monitoring PTP Follower Clock Quality

Alon Regev

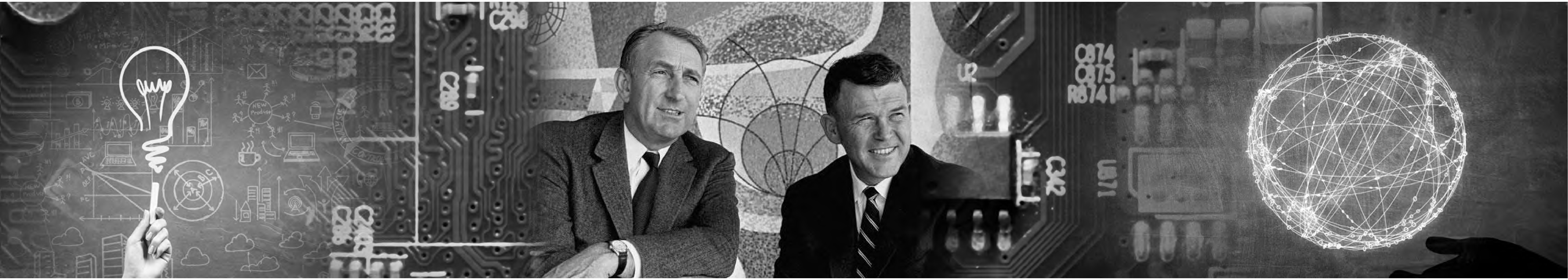
2021.10.06

Architect, Keysight Technologies

KEYSIGHT TECHNOLOGIES

Keysight Helps You Get to Market Faster

We help you create. Innovate. And deliver what's next.



The innovation leader in electronic design and test for over 75 years

Founded in 1939 by Bill Hewlett and Dave Packard as HP with an ongoing mission to help create new markets

Trusted hardware, innovative software and a global network of experts

Introduction

PTP time synchronization requires multiple components to behave as expected

- A failure or incorrect behavior at any level can cause time errors

Conformance and performance validation at the component level and at the system level prior to deployment helps ensure correct operation

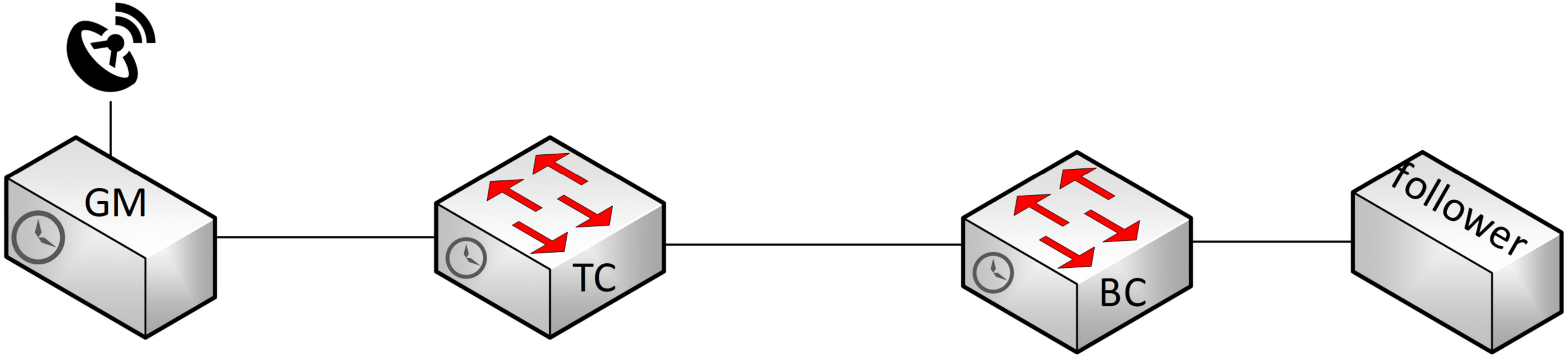
Post-deployment monitoring enables near-real-time detection of failures

While most components can be tested independently, a PTP follower (i.e. an ordinary clock) does not provide a standard way to measure its quality

This presentation will discuss and compare different approaches to measuring & monitoring PTP follower clock quality

The Follower Ordinary Clock observability problem

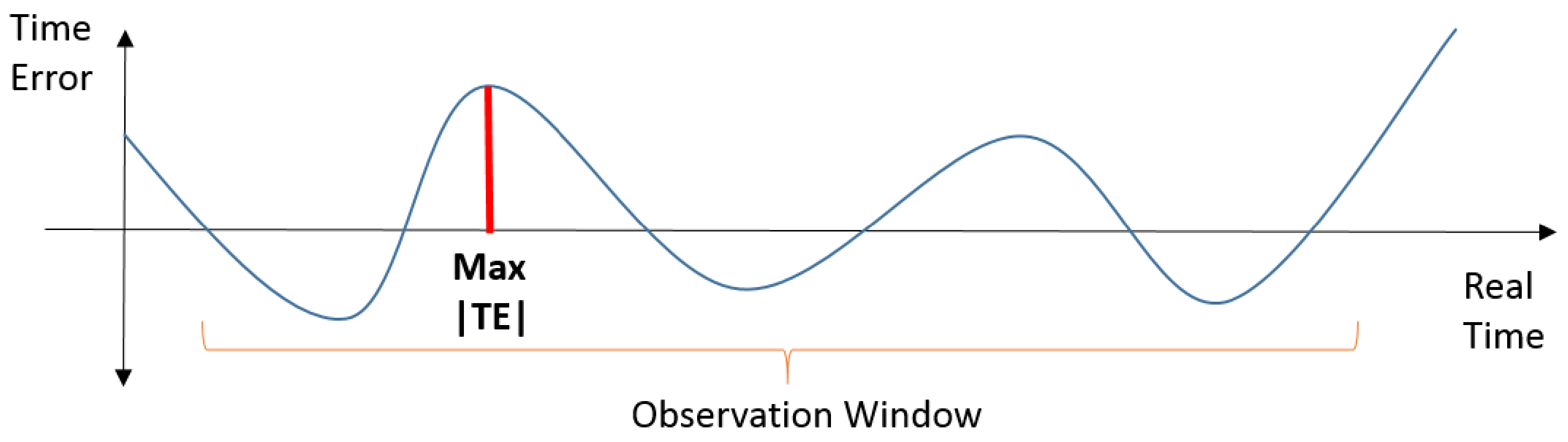
- GM, TC, or BC behavior can be observed
  - Output clock can be compared to the input clock or reference
- Follower Ordinary Clock does not have a standard, scalable clock observability point or method
- Different approaches to follower clock measurement are compared in this presentation



Validation of time at different phases

Device-level Test	System Test	Run-time Monitoring
• Conformance to standards • Performance under ideal conditions • Performance under “normal” (imperfect) clock • Effect of network traffic • Security	• Correct system level operation • End-to-end clock quality • Scalability / Performance • Resiliency • Security	• Ensuring timing performance • Detecting failures • Detecting timing attacks • Regulatory requirements

Measuring follower clock quality



- While the IEEE 1588 algorithm works well, imperfect implementations and/or hardware issues can cause the follower clock not to track the leader clock

- It is important to verify & certify implementations to make sure they are correct.
  - Often PTP implementations that have not been validated will have conformance issues

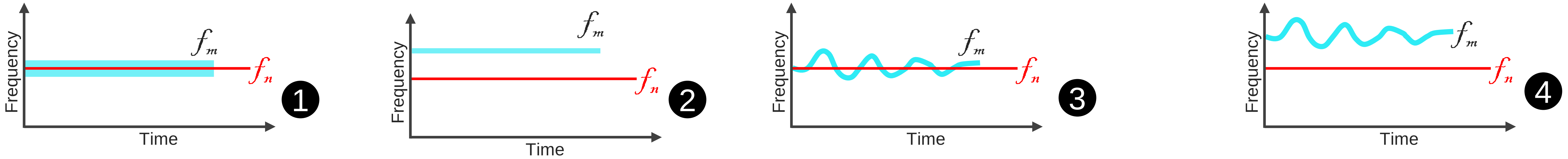
- To validate the clock accuracy on a follower, we start by tracking the Time Error (TE), which is the difference between the reference (such as the IEEE 1588 leader) and the DUT (IEEE 1588 follower).

- Other metrics can be derived from the Time Error measurements

Resolution, Accuracy & Precision

f_n – nominal frequency at which we want to operate
f_m – actual or measured frequency at which we are operating

- **Resolution:** The smallest interval between readout updates. For timing it is determined by the clock sampling frequency.
- **Accuracy:** The error between the real and measured value. For timing, it is measured as a mean of the time error over a given time period and number of data points. This is similar to constant time error as defined in the ITU-T G.8260 and G.8271.1
- **Stability:** For timing, it is a measure of how the mean time deviation varies with respect to variables such as time or number of samples.
- **Precision:** The random spread of uncertainty (example - time error) of measured values around the average measured values. For timing, this measurement is expressed by standard deviation of the time error from the mean value.



Accurate : $f_m = f_n$
Stable : constant value over time

Inaccurate : $f_m \neq f_n$
Stable : constant value over time

Accurate : mean $f_m = f_n$, f_m in centering around f_n
Unstable : different values over time

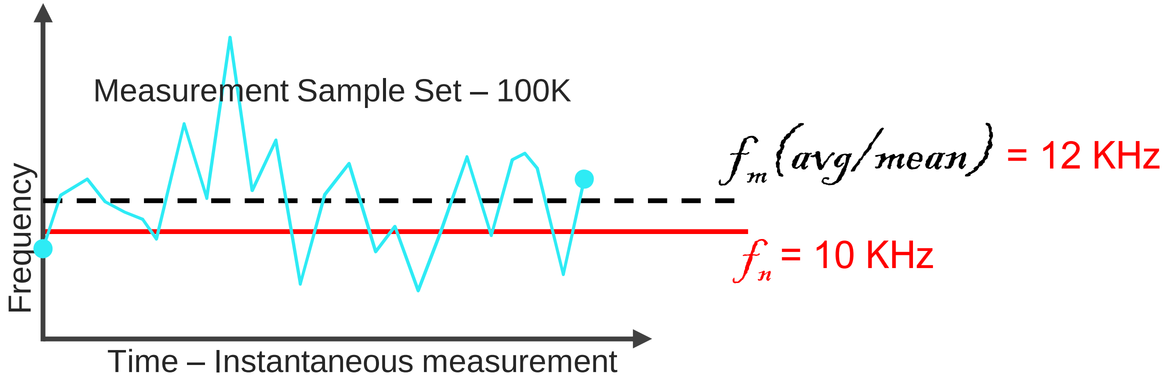
Inaccurate : $f_m \neq f_n$, f_m is way off f_n
Unstable : different value over time

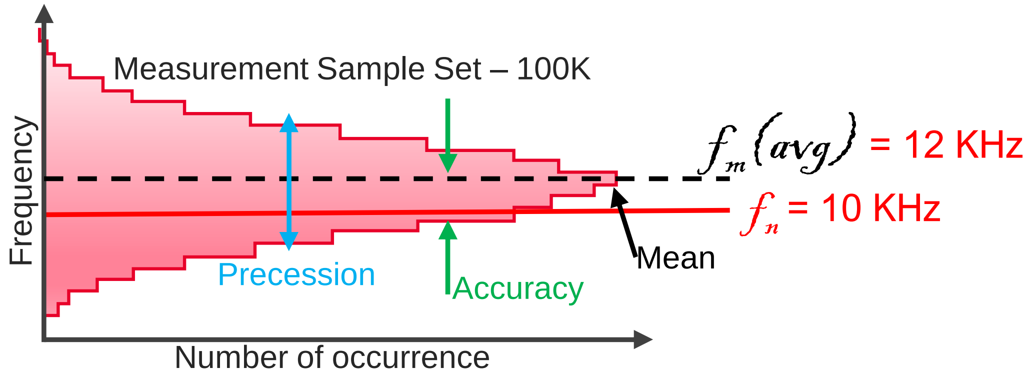
Clock quality measurement must be done over time. Concluding based on instantaneous measurement will be misleading.

Reference	
1588-2008	Sec 3.1.1 Accuracy
G.8260	Sec 3.1.20 Time error
G 8271.1	IV.2 Components of time error

KEYSIGHT TECHNOLOGIES

Accuracy & Precision





Histogram

- The histogram shows how often a measured value occurs.
- The difference between the average measured value and real value is the accuracy.
- The width of the histogram indicates the spread of individual measurements. This distribution of measurements is called accuracy.

Experts usually refer to **precision** when they say a clock is so good that it will gain or lose at most one second over millions of years.

When they speak of **accuracy**, experts typically are referring to how well a clock matches a given standard reference.

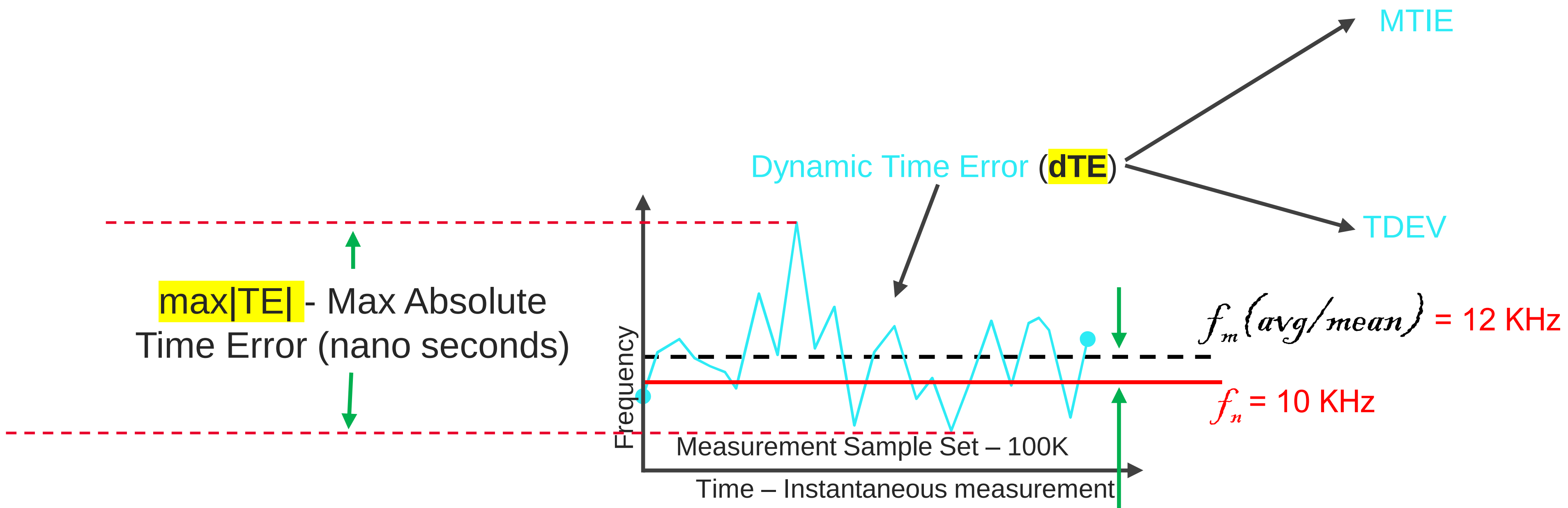
1. Rightmost point of the histogram is the "mean"- is the measured value that has been most frequently measured.

2. **Accuracy** - The difference between the average measured value and real value. Unit is micro/mili/seconds

3. **Precision** – standard deviation of the time error from the mean value (i.e. Accuracy).

4. Precision – For time, PPM / PPB is a unit often used to indicate precision.

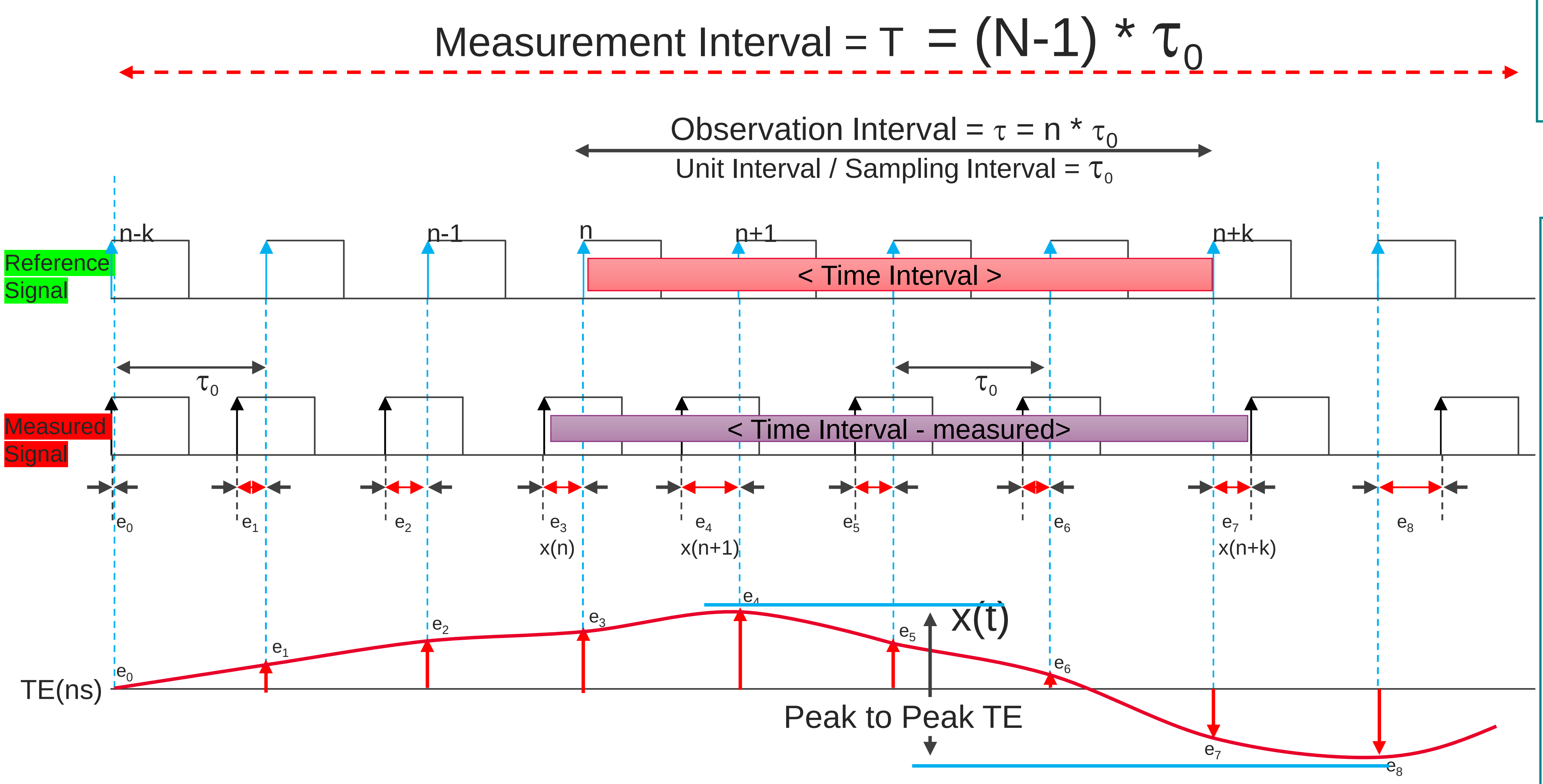
Constant Time Error (cTE) & Dynamic Time error (dTE)



Accuracy (this is The constant Time error **cTE**)
Specified in nanoseconds

Time Error (TE) & Time Interval Error (TIE)

- The understanding of phase is critical as many other measurements are all derived from phase.
- TIE or “time interval error”, perhaps the most common term used for **phase** in the standards. By measuring TIE we measure phase error.



Time Error - TE

1. Measurements taken every τ_0 seconds.
2. $e_{(n)}$ – is the discrete Time Error (TE) at n^{th} sample measured.

Time Interval Error – TIE

1. A time interval starts at sample n and finishes at sample n+k
2. So the nominal duration measured in reference clock is k * τ_0
3. Clock under test will measure it as τ_α = $k\tau_0$ + [x(n+k) – x(n)]
4. {TIE(n,k)= [x(n+k) – x(n)]} is the Time interval error for a specific τ_0 and observation interval τ.
5. TIE is the measuring dynamic error. We don’t compare absolute time to the reference, but just the error occurring during the interval

KEYSIGHT TECHNOLOGIES

Stability Matrices

Traditional metrices developed for Synchronous Digital Hierarchy (SDH) - G810

- Traditionally Applied to oscillators and synchronization interfaces.
  - ADEV - Allan deviation
  - MDEV - Modified Allan deviation
  - TDEV - Time deviation
  - TIErms - Root mean square Time Interval Error
  - MTIE - Maximum Time Interval Error

These metrices are still applicable for packet networks

Additional Metrices defined for packet network

Metrices for frequency transport (G8260)		Metrices for time transport (G8260)	Metrices PDV(G8260)
Metrices for phase and wonder	Metrics estimating frequency stability		
1. TIE based (seconds) – Time Interval Error • MATIE, minMATIE - Maximum average time interval error • pktfilteredTIE • pktfilteredMTIE	1. Tdev based (seconds) – MinTDEV, PercentileTDEV, BandTDEV, ClusterTDEV, pktfilteredTDEV 2. MAFE, minMAFE - max avg frequency error 3. pktfilteredFFO (ratio) - fractional frequency offset	1. Packet-selected two-way time error 2. Packet filtered two-way time error 3. Maximum/Minimum/Peak-to-peak average time error (maxATE, minATE, ppATE)	1. Observed floor delay 2. Minimum floor delay

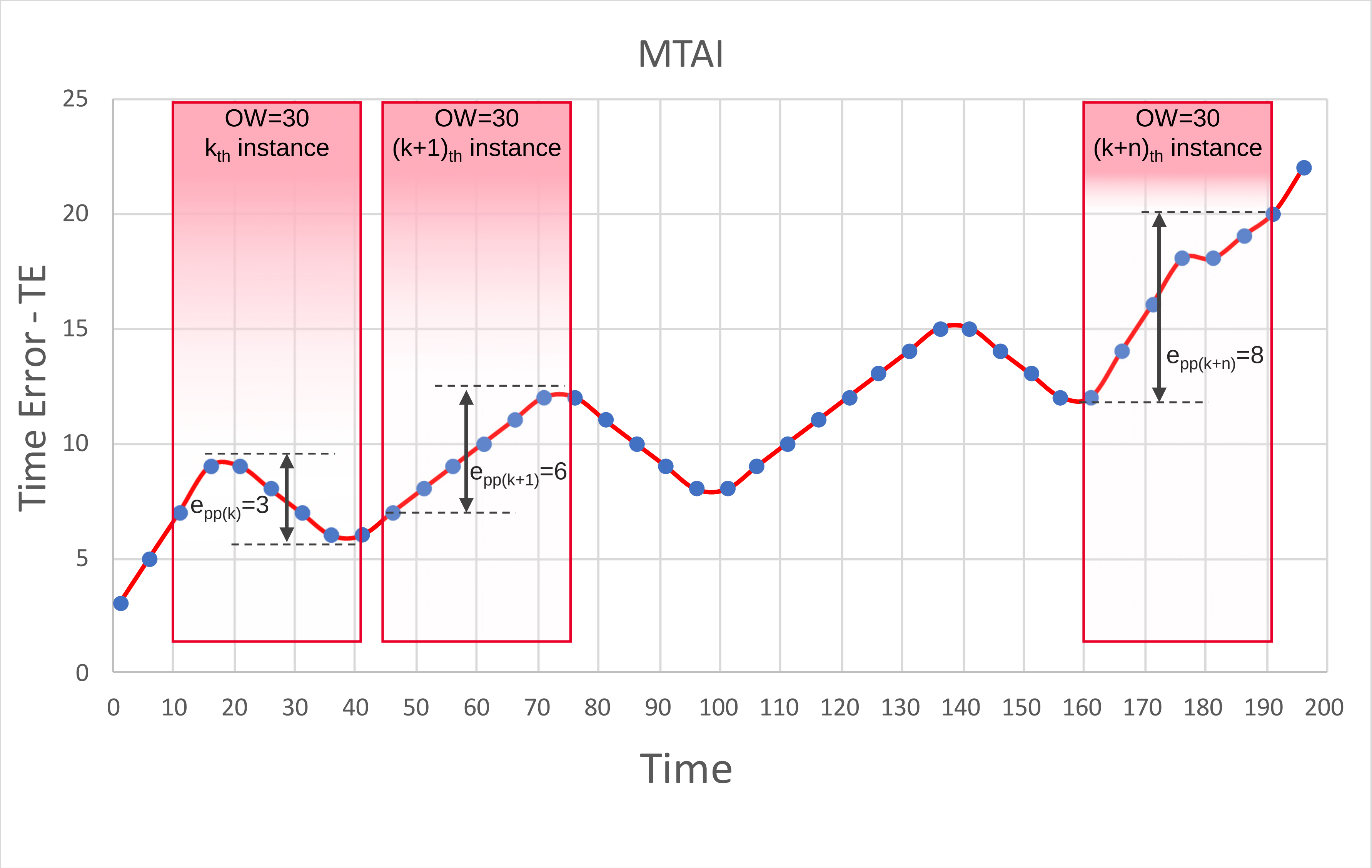
KEYSIGHT TECHNOLOGIES

Reference	
G810	Appendix II
G8260	I.4

Metrics for measuring the performance of a clock

Purpose	Metrices	
time error	Maximum/Minimum/Peak-to-peak average time error (maxATE, minATE, ppATE)	
Phase & Wander	TIE – Time Interval Error	
	MTIE – Maximum time interval error	$MTIE(n\tau_0) \cong \max_{1 \le k \le N-n} \left[\max_{k \le i \le k+n} x_i - \min_{k \le i \le k+n} x_i \right]$, n =1,2,…..N
	MATIE - Maximum average time interval error. MATIE predicts the largest difference in averaged time interval error that occurs between adjacent averaging windows of width $n\tau s$	$\mathrm{MATIE}(n\tau_0) \cong \max_{1 \le k \le N-2n+1} \frac{1}{n} \left\lvert \sum_{i=k}^{n+k-1} (x_{i+n} - x_i) \right\rvert$
frequency stability	TDEV -pktfilteredTDEV, clusterTDEV , bandTDEV, percentileTDEV, minTDEV, TDEV	$TDEV(\tau) = \sqrt{\frac{1}{6n^2} \left\langle \left[\sum_{i=1}^{n} (x_{i+2n} - 2x_{i+n} + x_i) \right]^2 \right\rangle}$
	meanTDEV - stability of the clock over different observation intervals. Note this is for packet domain only. For conventional SDH type clock it is TDEV. specify network wander limits for timing signals.	$minTDEV(n\tau_0) \cong \sqrt{\frac{1}{6(N-3n+1)} \sum_{i=1}^{N-3n+1} [x_{min}(i+2n) - 2x_{min}(i+n) + x_{min}(i)]^2}$ for n = 1, 2, ..., integer part $\left(\frac{N}{3}\right)$
	bandTDEV	$\mathrm{bandTDEV}(n\tau_0) \cong \sqrt{\frac{1}{6(N-3n+1)} \sum_{i=1}^{N-3n+1} \left[x'_{band\_avg}(i+2n) - 2x'_{band\_avg}(i+n) + x'_{band\_avg}(i) \right]^2}$
	clusterTDEV	
	pktfilteredTDEV	
frequency accuracy	MAFE - maximum average frequency error minMAFE FFO - fractional frequency offset	$FFO = \frac{v(t) - v_{nom}}{v_{nom}}$

Maximum Time Interval Error (MTIE)



$$MTIE(n\tau_0) \cong \max_{1\le k\le N-n}\left[\max_{k\le i\le k+n} x_i - \min_{k\le i\le k+n} x_i\right], \text{ n =1,2,….N}$$

No samples : N = 40

Sampling Interval : τ_0 = 5 sec

Measurement Interval : T = (N-1) τ_0 =195 sec

Observation Window (OW) : $\tau = n.\tau_0$, Where $1 \le n \le (N-1)$

i.e $1 \le n \le 39$

- So the values of OW are : 5, 10, 15, ….195
- So for OW length τ = 30,
- MTIE(τ_{30}) = MAX($e_{pp(1)}$, $e_{pp(2)}$, …… $e_{pp(k)}$, $e_{pp(k+1)}$, …… $e_{pp(k+n)}$) = 8
- Our customers will like to do MTIE calculation for different window size.

Note if we change the Observation Window (OW) size then the MTIE value will change.

Movement of Window :-

- Non Overlapping window
- Move Window by one sample (heavy overlap)
- Move window by N samples with partial overlap.

KEYSIGHT TECHNOLOGIES

MTIE and TDEV

- Both these metrices measure wander at the clock's output. Wander can never be filtered out since any clock by design incorporates a low pass filter which allows low frequency wander to pass.
- MTIE measures the peak-to-peak phase variation. It is plotted against observation interval. It never decreases with increase in observation interval, so in a way it is a pessimistic indicator
- TDEV measures the stability in rate of phase change. It is plotted against integration time. If phase of clock's output becomes more and more stable a decrease in TDEV is seen with increase in integration time.
- For any clock's output if these metrices stay within a predefined limit, then overall time error contributed by the clock will also stay within a limit. That is why these data are very important in designing a clock synchronization network with many clocks while keeping the overall time error in limit demanded by the end application
- In simple terms, the MTIE defines the maximum phase excursion within an observation window and the TDEV measures the noise level of a clock source in the given observation window. Both the MTIE and TDEV depend on a time parameter, the observation window. Because of this, their limits are expressed though masks specified in terms of a variable observation window.

$$\mathrm{MTIE}(n\tau_0) \cong \max_{1 \le k \le N-n} \left(\max_{k \le i \le k+n} x(i) - \min_{k \le i \le k+n} x(i) \right), \quad n = 1, 2, \ldots, N-1$$

$$\mathrm{TDEV}(n\tau_0) \cong \sqrt{\frac{1}{6n^2(N-3n+1)} \sum_{j=1}^{N-3n+1} \left[\sum_{i=j}^{n+j-1} \left(x_{i+2n} - 2x_{i+n} + x_i \right) \right]^2}, \quad n = 1, 2, \ldots, \text{integer part} \left(\frac{N}{3} \right)$$

PTP Network Monitoring

- Reasons for monitoring a PTP Network
  - Making sure timing performance is maintained.
    - Even with no node failures, GNSS interference or hacking, timing performance could degrade due to a change in network traffic patterns.
  - Building a data archive for regulatory compliance
  - Characterize PTP implementation performance prior to using in the live network

Approaches to monitoring of follower clocks

Within the follower

- Compare input clock to local clock
- Compare input clock to a second input clock

Using external monitor

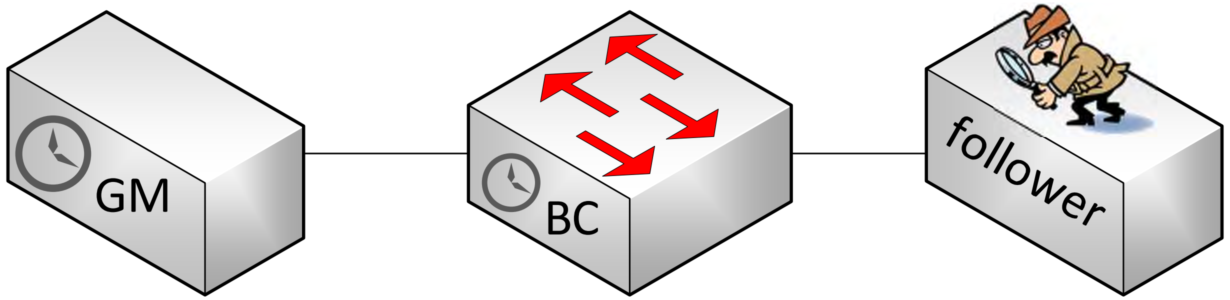
- Follower shares its clock with an external monitor
- Monitor compares with reference

Compare with peers

- Distributed validation approach
- Follower shares its clock with a set of peers
- Peers compare their clocks to measure time error

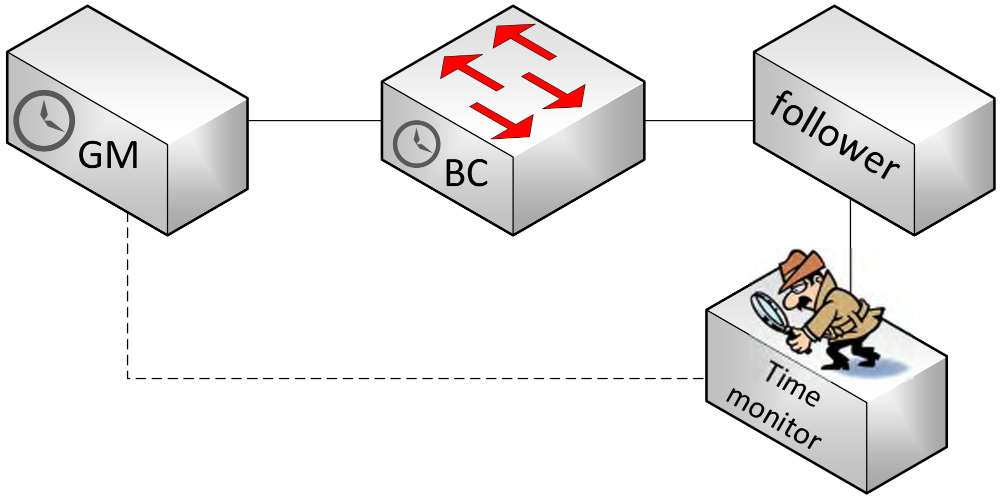
Clock quality monitoring in the follower

- Compare input clock to local clock
  - Simplest approach
  - Cannot detect inaccuracies in input clock
  - Cannot distinguish between input and local stability issues
  - Cannot detect packet delay attacks
- Compare time from different sources
  - Requires additional clock connection(s) to each follower
  - Best if the different sources use different paths, domains, protocols, or medium
  - If only a single network connection is used, a clock issue or attack can affect both time sources



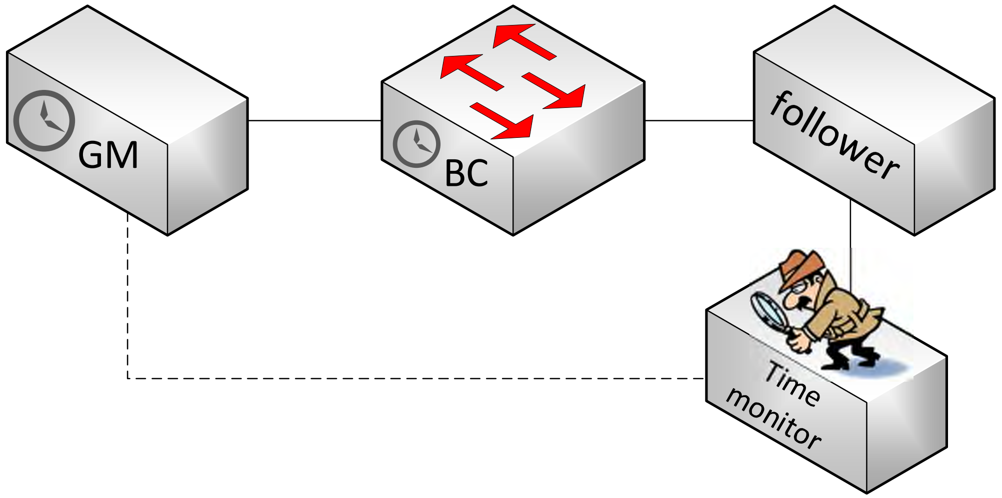
External monitor

- Follower sends its time to another device to detect time errors
  - Time can be sent as PTP on another domain, PTP on another domain, 1PPS output, etc.
  - Time monitoring device can be any device or process on the network that can measure the time error between the follower time and the reference time (from GM or other authoritative clock source)
- Works best if network path from leader to follower is different from monitoring path from follower to monitor
- Traditional approach: use 1PPS
  - This works well, but is not scalable
- Another approach is to use a "reverse sync" where the follower to sends its time to a monitoring device
  - Multiple approaches – see slides later in this presentation
  - Doug Arnold and I submitted a proposal to the 1588 WG to standardize
- External monitor can monitor GM, BC/TC, as well as follower clocks
- Path to monitor may limit accuracy



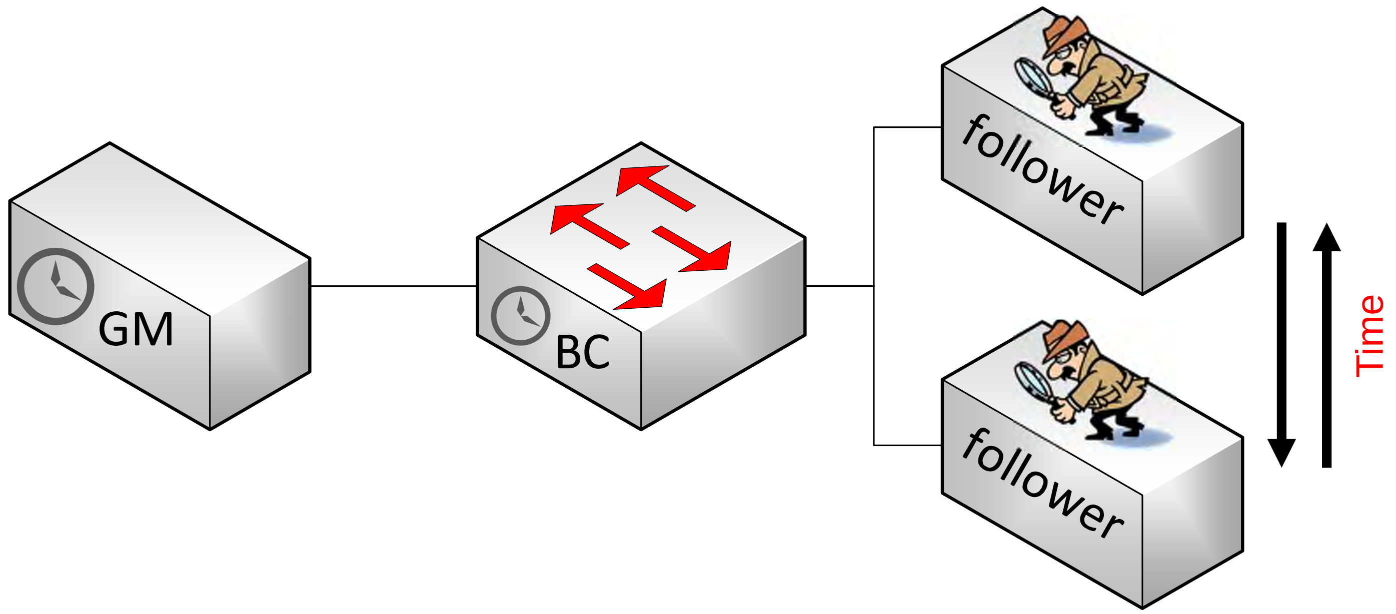
Reasons to use an external monitor

- Regulatory requirements
- Self reported performance data can be erroneous (for example, if there are asymmetries in the PTP Network)
- PTP node could have implementation errors which affect self reported data
- Need method to characterize PTP implementations that lack a 1PPS output, (i.e. in a laboratory environment)



Using peer validation of time

- Each follower shares its local time with a set of peers
  - This can be done using similar methods to the external monitor
- Works best with randomized peers not necessarily in same network locations / paths
  - Inaccuracies due to the monitoring path also have to be balanced
- Does not require any “specialized” monitoring nodes
- Additional network traffic required for exchange of time information
  - There is a balance between bandwidth and issue detection time and resolution



Methods for external measurement of the follower clock

1. Compare 1 Pulse Per Second (1PPS) or similar output
2. IEEE 1588 “Ingress Event Monitor” where a follower clock reports the time error it is seeing each time it receives a sync
3. IEEE 1588 “Egress Event Monitor” where the follower provides the PTP time of messages that it sent, which can be compared to the time these messages are received by the leader
4. “Reverse Sync method” where the follower sends sync or other time signaling messages to the monitor validate the timing

For device testing, it is recommended to implement test modes containing at least one of the above methods in each follower

- Implementing multiple methods allows cross-validation of the measurement methods

Using 1PPS outputs to validate follower time synchronization

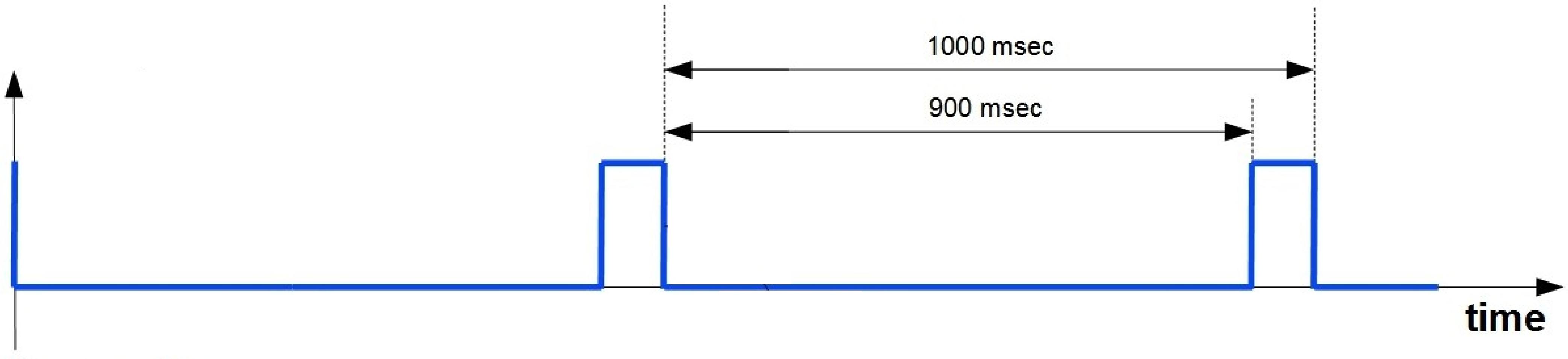
- Both reference and follower output a pulse that starts on each second boundary of the PTP clock
- Traditionally an oscilloscope is used to compare the time of the two outputs
  - each measurement shows a phase difference in the clock
  - Using multiple measurement, maximum time error, TIE analysis, and jitter can be ascertained.
- Today, many NICs can compare the time of a 1PPS input to their internal time
  - simpler, lower cost measurement solution, potentially at a lower resolution

Advantages of 1PPS method:

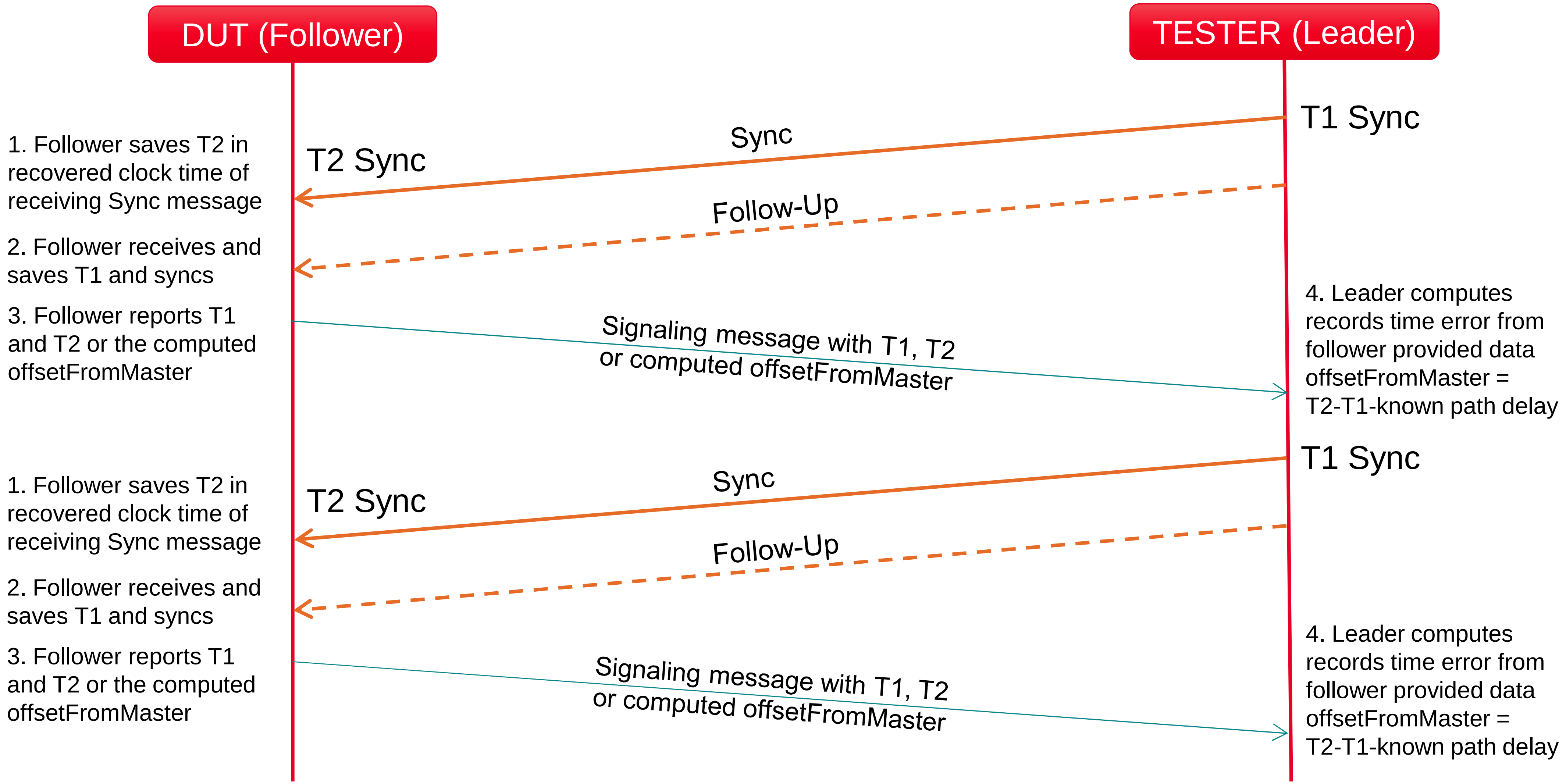
- Known industry standard which already exists on many devices.
- Only an oscilloscope or supported NIC is needed to make this test.

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 must be relatively close together
- Not practical for large-scale testing (such as validating time synchronization in an entire sports arena or data center).



Ingress method



Ingress method relies on Follower 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!**

See IEEE Std 1588-2019 clause 16.11.4

KEYSIGHT TECHNOLOGIES

SLAVE_RX_SYNC_TIMING_DATA TLV

tlvType (2)
lengthField (2)
sourcePortIdentity (10)
sequenceId[1] (2)
syncOriginTimestamp[1] (10)
totalCorrectionField[1] (8)
cumulativeScaledRateOffset[1] (4)
syncEventIngressTimestamp [1] (10)
….
sequenceId[N] (2)
syncOriginTimestamp[N] (10)
totalCorrectionField[N] (8)
cumulativeScaledRateOffset[N] (4)
syncEventIngressTimestamp [N] (10)

This TLV is used for transferring the receive recovered time message information to a monitoring station.

The same data can be transferred using other out of band means if desired.

The data from one or more received Sync/FollowUp messages is combined into a single TLV

syncOriginTimestamp provides T1
syncEventIngressTimestamp provides T2
totalCorrectionField provides the total PTP "residence time" through bridges along the path

cumulativeScaledRateOffset is not used as part of the time error calculation but it's useful in determining possible sources of error.

SLAVE_RX_SYNC_COMPUTED_DATA TLV

tlvType (2)
lengthField (2)
sourcePortIdentity (10)
sequenceId[1] (2)
offsetFromMaster[1] (10)
meanPathDelay[1] (8)
neighborRateRatio[1] (8)
….
sequenceId[N] (2)
offsetFromMaster[N] (10)
meanPathDelay[N] (8)
neighborRateRatio[N] (8)

This TLV is used for transferring the time error information to a monitoring station.

The same data can be transferred using other out of band means if desired.

The computed data from one or more received Sync/FollowUp messages is combined into a single TLV.

- offsetFromMaster provides the computed time error at the receive of the Sync with the corresponding sequenceId
- meanPathDelay provides the path delay used in calculating the time error at the receive of the Sync with the corresponding sequenceId
- neighbourRateRatio provides rate ratio computed at the receive of the Sync with the corresponding sequenceId

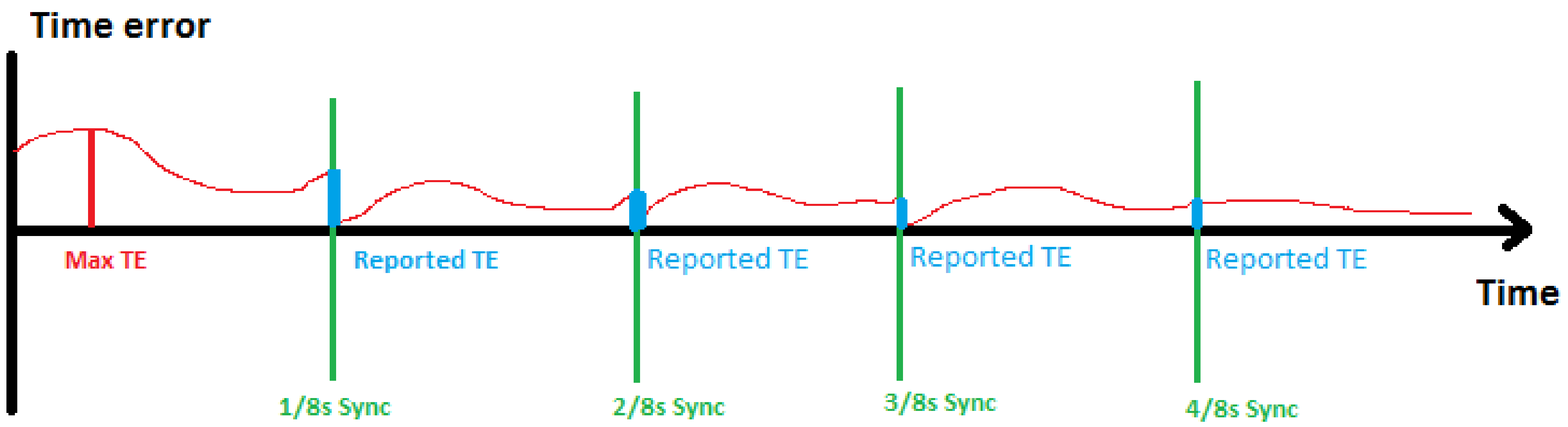
Ingress method pros/cons

Advantages of Ingress method:

- Reporting of 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 Follower except sending TLV/storing the data

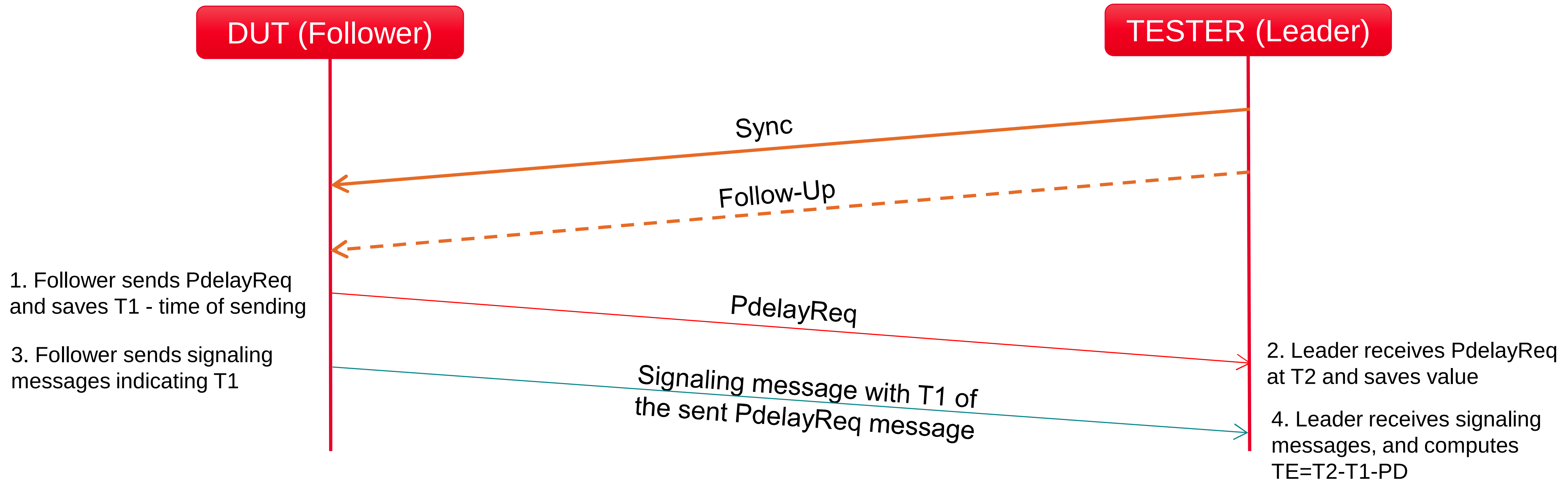
Shortcomings of Ingress method:

- **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 Follower 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



KEYSIGHT TECHNOLOGIES

Egress method



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

Advantages of Egress method:

- **Monitor evaluates the error, it does not rely on the Follower to report it**
- Reporting of 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 Leader of DUT**
    - **See IEEE Std 1588-2019 clause 16.11.5**

Egress Monitor - SLAVE_TX_EVENT_TIMESTAMPS TLV

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

Event origin timestamp is in PTP domain.

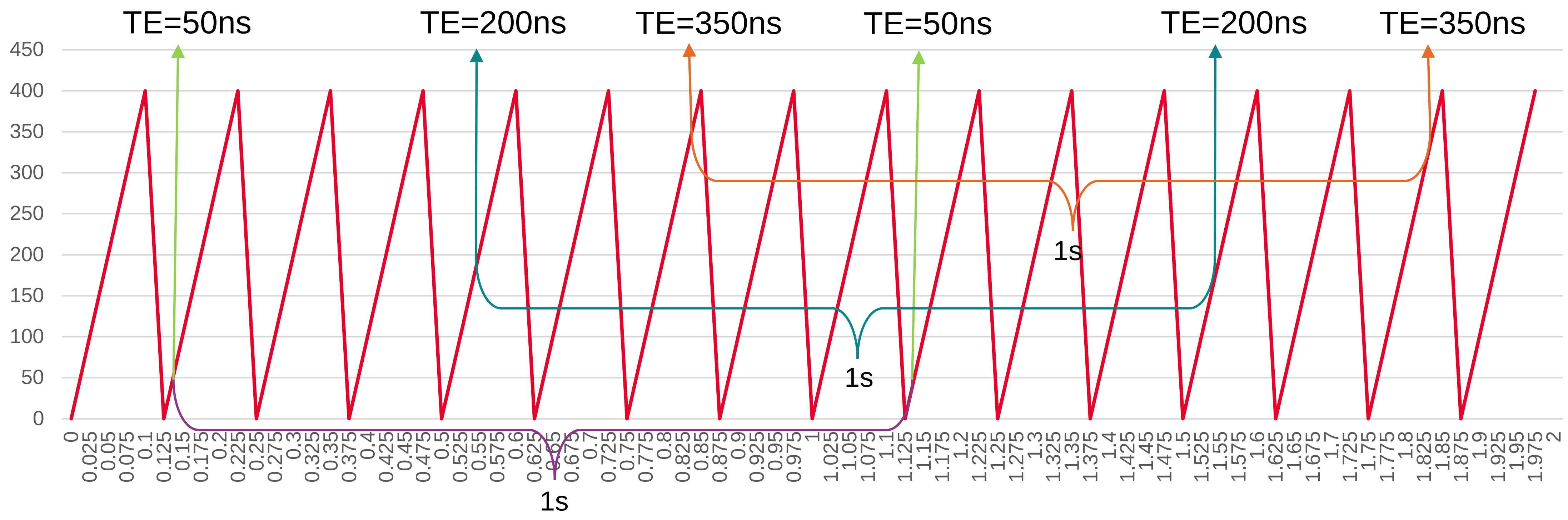
This can be used in conjunction with the actual DelayReq or PdelayReq messages sent by the follower and the known path delay to the measuring device to determine the time error of the follower device.

Problems in using Egress method

For PTP common sense is that reporting would be based on the sent DelayReq event messages. If DelayReq 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 Follower clock drifting 400ns between each 2 Sync messages (8 per sec == 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 DelayReq is sent close after Sync is received measured error is lower than one
5. Because DelayReq interval is a multiple of the Sync interval, this would happen at every time

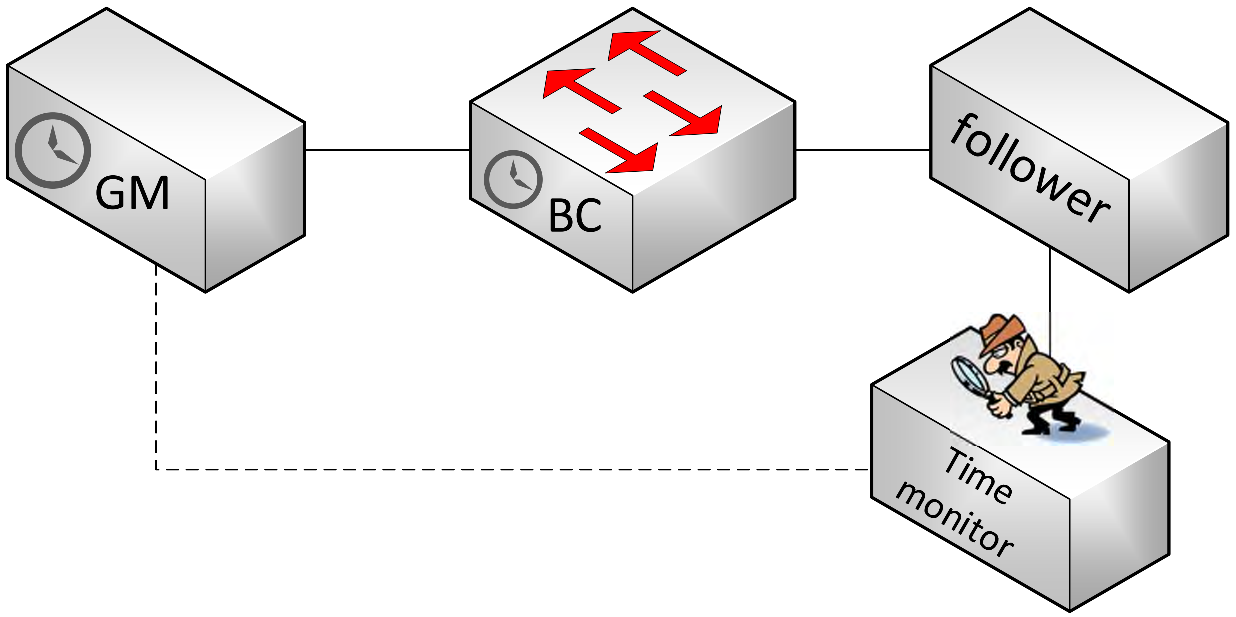
Not just theory, this was experienced with real devices in lab testing !



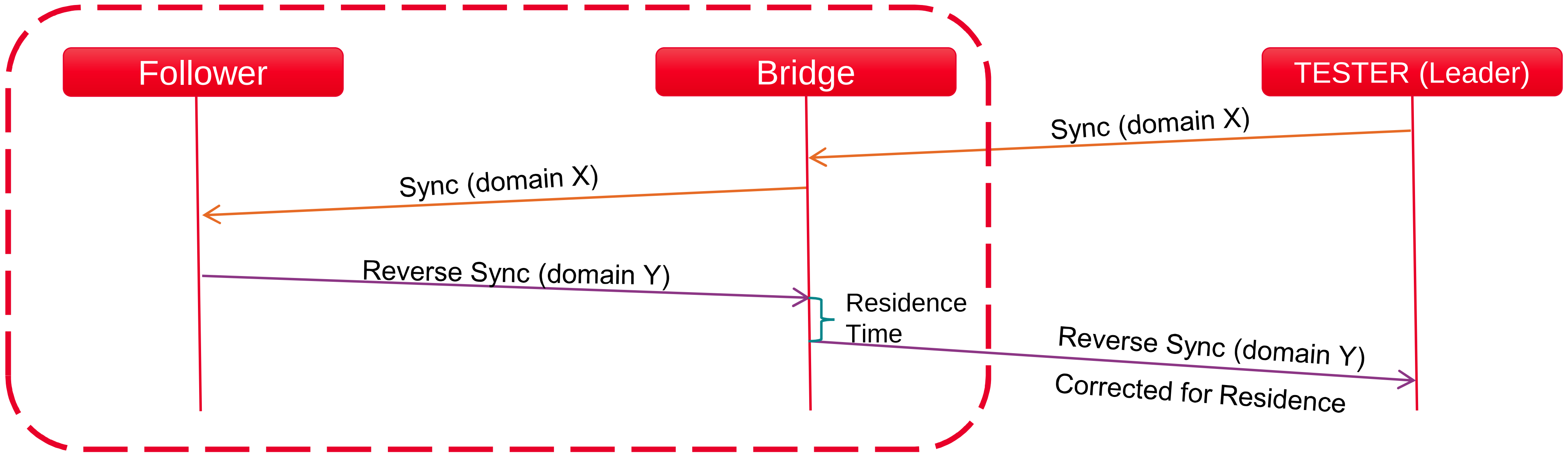
KEYSIGHT TECHNOLOGIES

Reverse Sync

- In the reverse sync message, follower sends its time to the monitoring device using a PTP message
- It is called “reverse sync” as in the original implementations a Sync message is used to send the time from the Follower to the monitoring node.
- There are multiple implementations currently used in use in the real world (see following slides)
- Doug Arnold and I submitted a proposal to the 1588 WG to standardize this as an optional feature of the next 1588 version
  - See https://ieee-sa.imeetcentral.com/1588/folder/WzIwLDEyMTIxMjE5XQ/WzIsNzEzOTY5MDVd/ (access limited to 1588 WG participants)
  - Latest proposal is at https://ieee-sa.imeetcentral.com/1588/folder/WzIwLDEyMTIxMjE5XQ/WzIsNzM3ODgzOTFd/ (access limited to 1588 WG participants)
- Current proposal in the IEEE 1588 working group is to use a new signaling message for this purpose

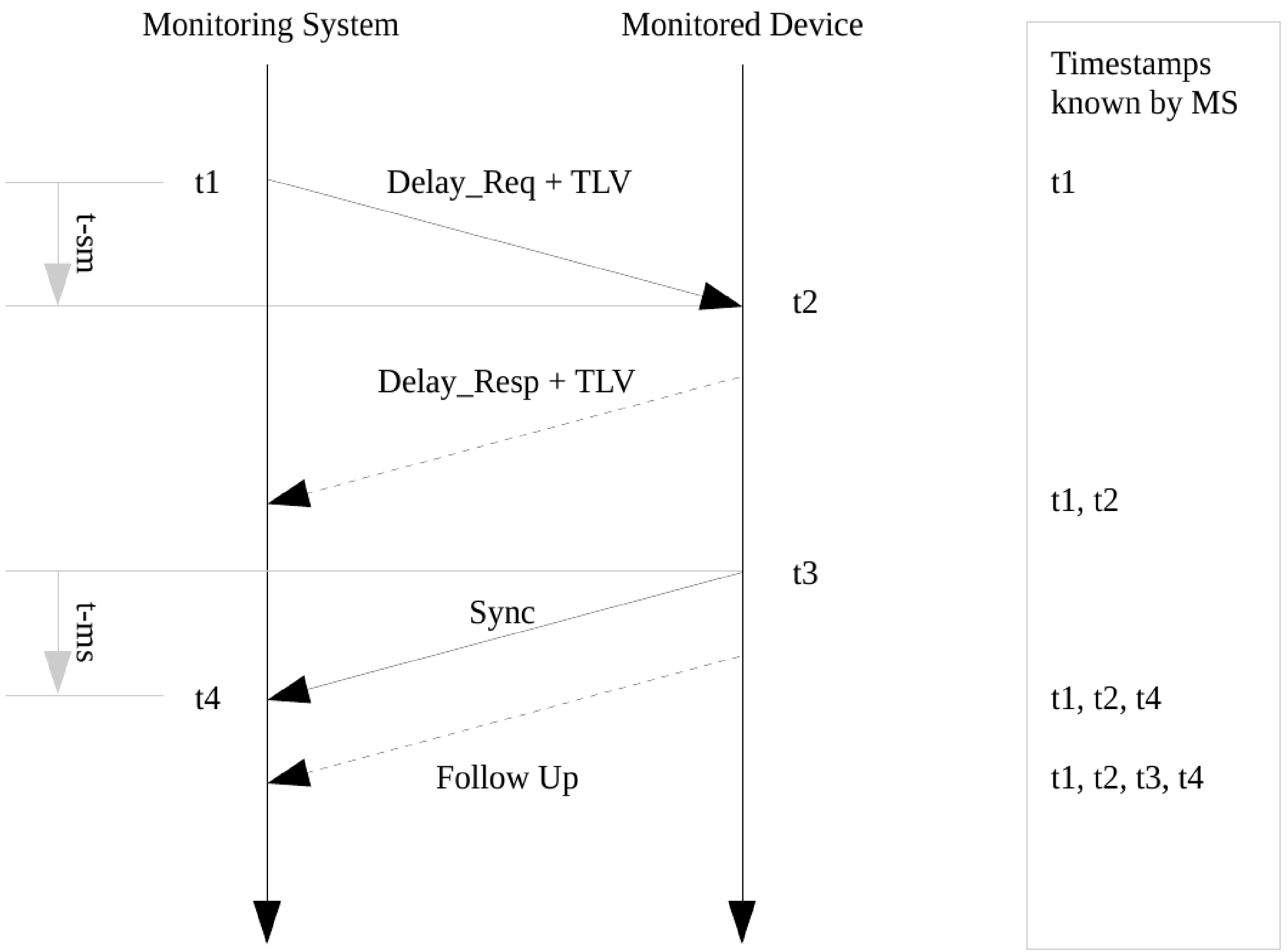


Existing implementation: Avnu Reverse Sync Method



- Monitor may also be leader or GM.
- Measurement initiated by Sync message
- Measurement traffic isolated with domains and special signaling message
- Measurement protocol operates in two domains
- See https://avnu.org/wp-content/uploads/2014/05/Avnu-Testability-802.1AS-Recovered-Clock-Quality-Measurement-1.0_Approved-for-Public-Release.pdf

Existing implementation: Meinberg Net Sync Monitor



Measurement initiated by Delay_Request message.

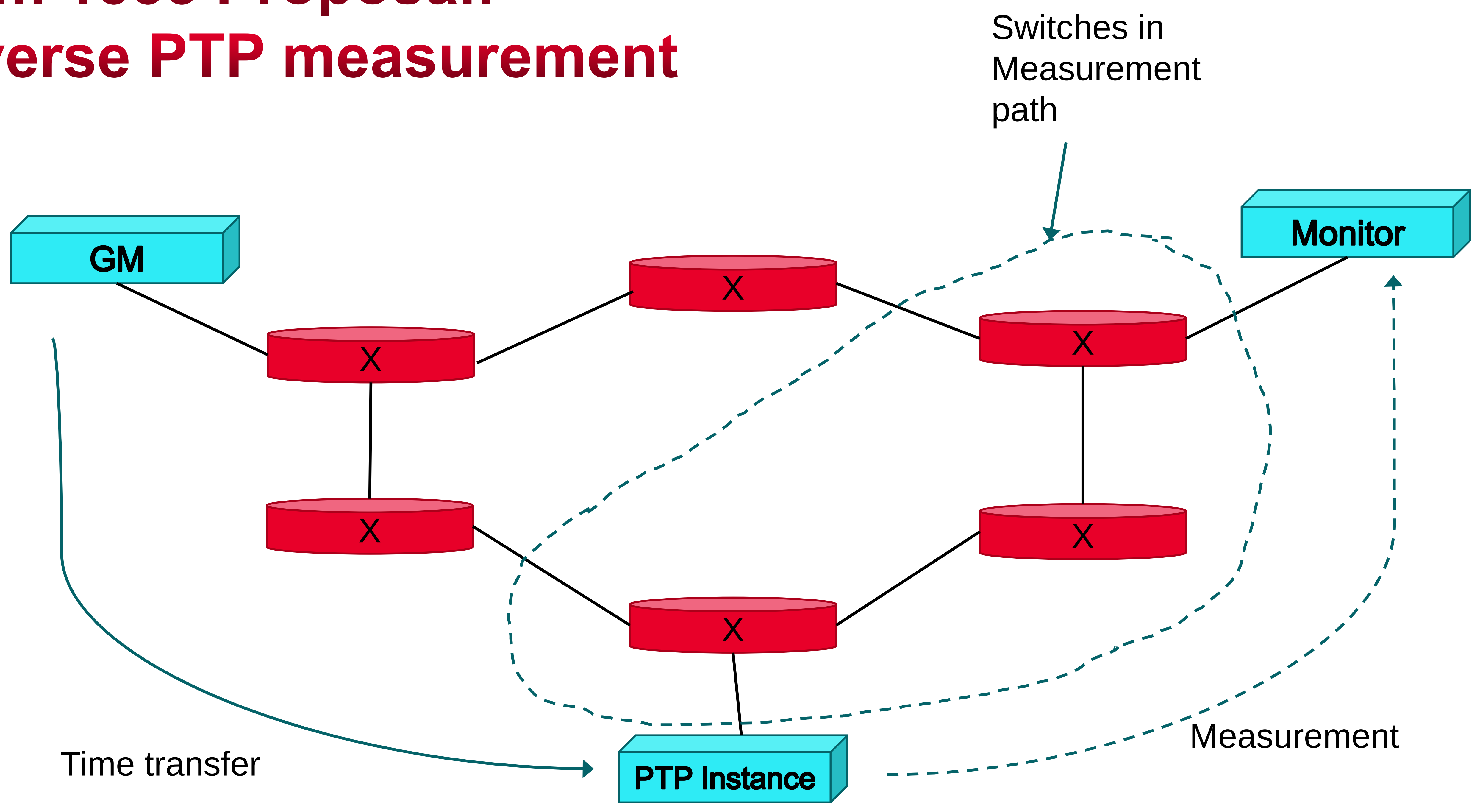
Measurement traffic isolated with unicast and TLVs

KEYSIGHT TECHNOLOGIES

Desired properties of reverse PTP measurements

- Measurement is initiated by the monitoring node. The PTP follower port responds to this request
- Each measurement is independently executed
  - i.e. there are no scheduled measurements. This way the measured ports do not need to save state with respect to this mechanism.
- The reverse PTP packet exchange could be significantly less frequent than typical PTP message rates.
  - The monitoring device is measuring the follower port, not steering to it.
- Measurement times should occur at different fractions of the sync interval of the PTP network
  - To avoid sampling time artifacts
- To detect network asymmetry, the monitor should be on a different PTP Communication Path than the GM

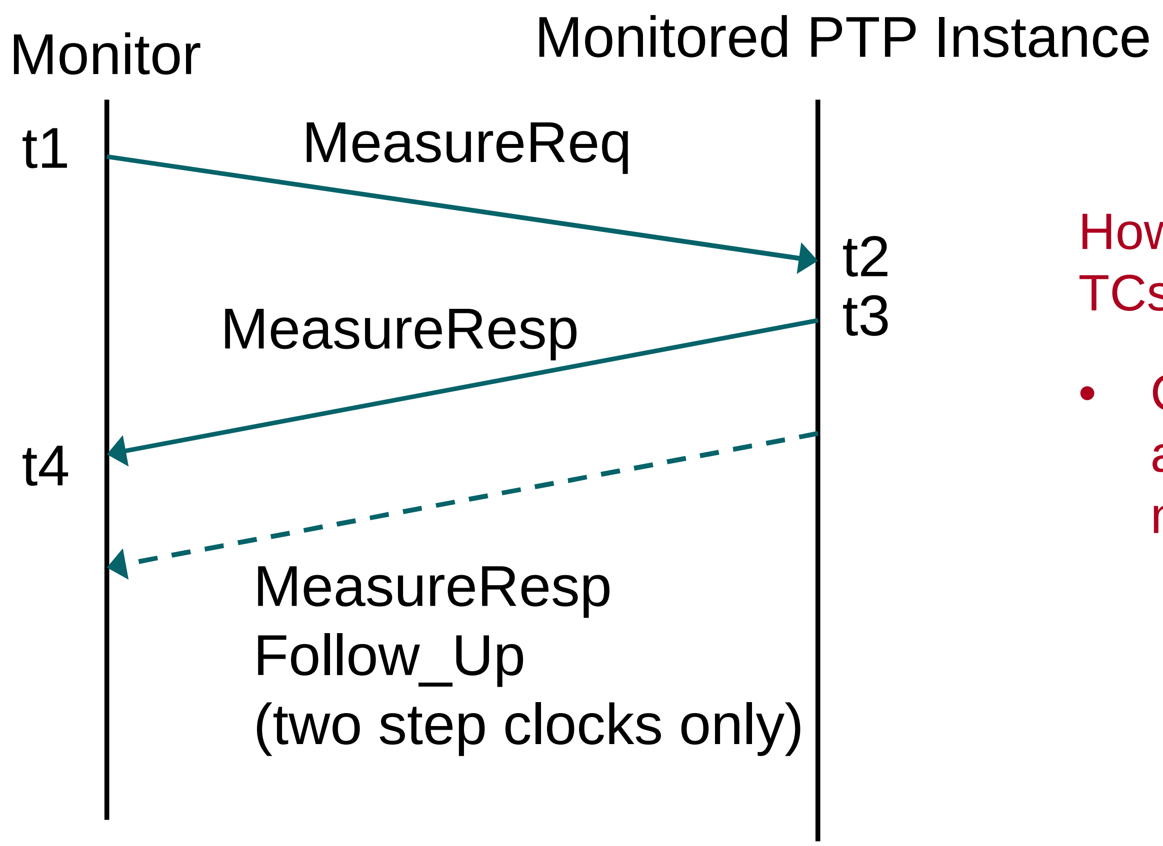
From 1588 Proposal: Reverse PTP measurement



Proposed message sequence (E2E)

Measure messages

- signaling messages
- Same domain number as measured domain
- MeasureReqEvent and MeasureResp are both unicast event messages
- MeasureResp_Follow_Up is a unicast general message



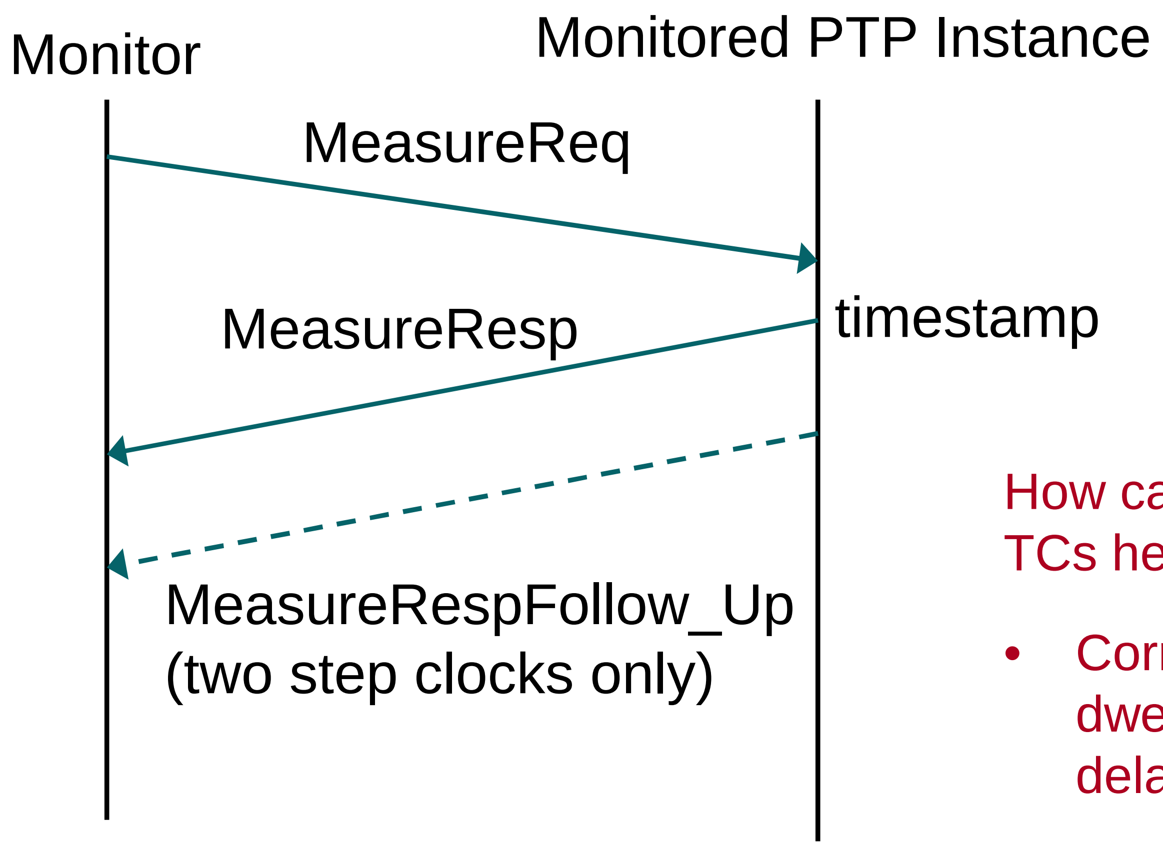
How can intermediary BCs and TCs help:

- Correct measureReqEvent and MeasureResp messages for dwell time

Proposed message sequence (P2P)

Measure messages

- signaling messages
- Same domain number as measured domain
- MeasureReqGen is a unicast general message
- MeasureResp is a multicast or unicast event message
- MeasureRespFollow_Up is a muliticast or unicast (same as MeasureResp) general message



How can intermediary BCs and TCs help:

- Correct measureResp for dwell time + upstream link delay

Measure TLVs

MeasureReq TLV

- Padding to equal MeasureResp message length

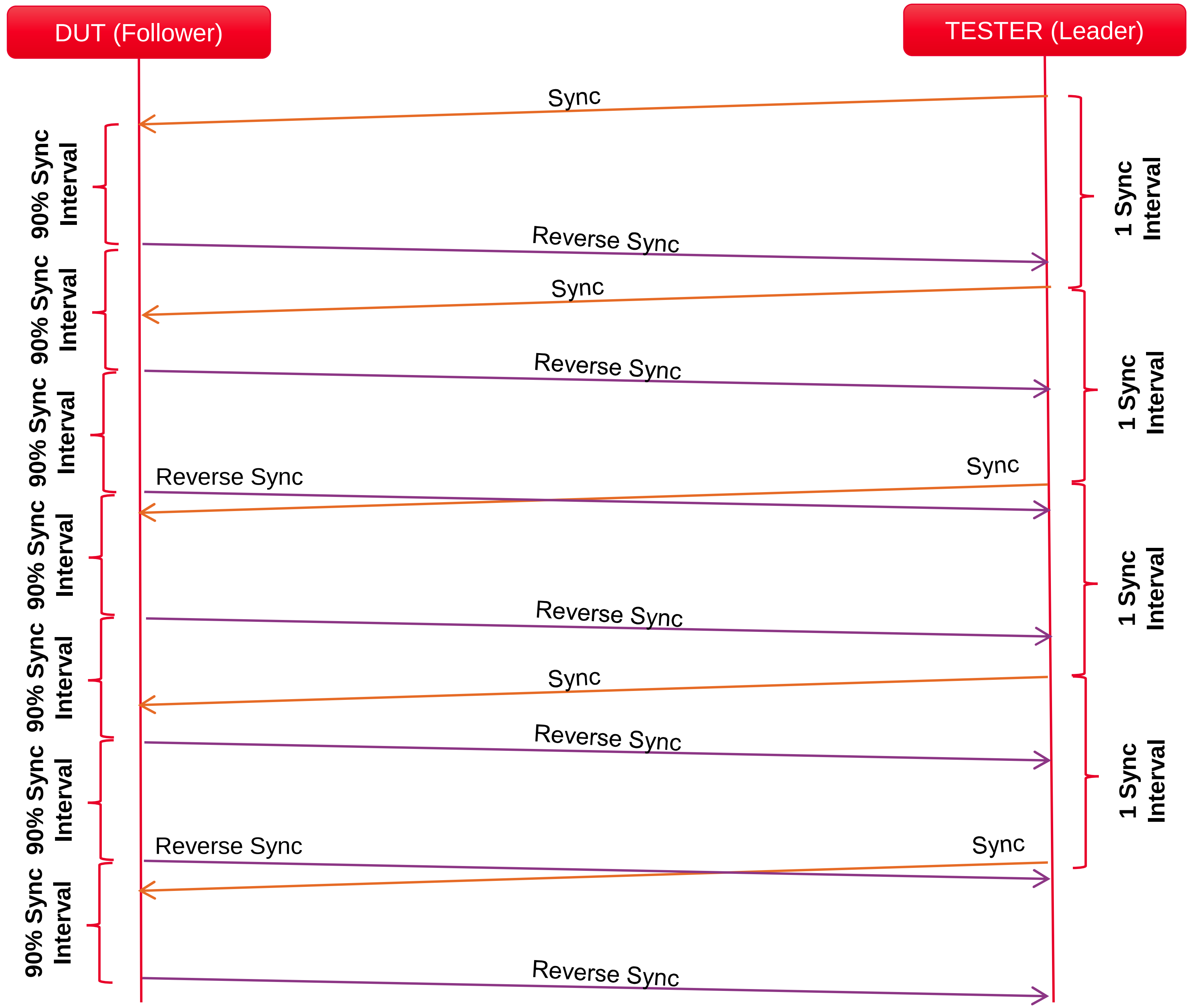
MeasureResp TLV

- MeasureReceipt timestamp (may be filled with zeros for P2P)
- MeasureResponse timestamp

MeasureResp Follow_Up TLV

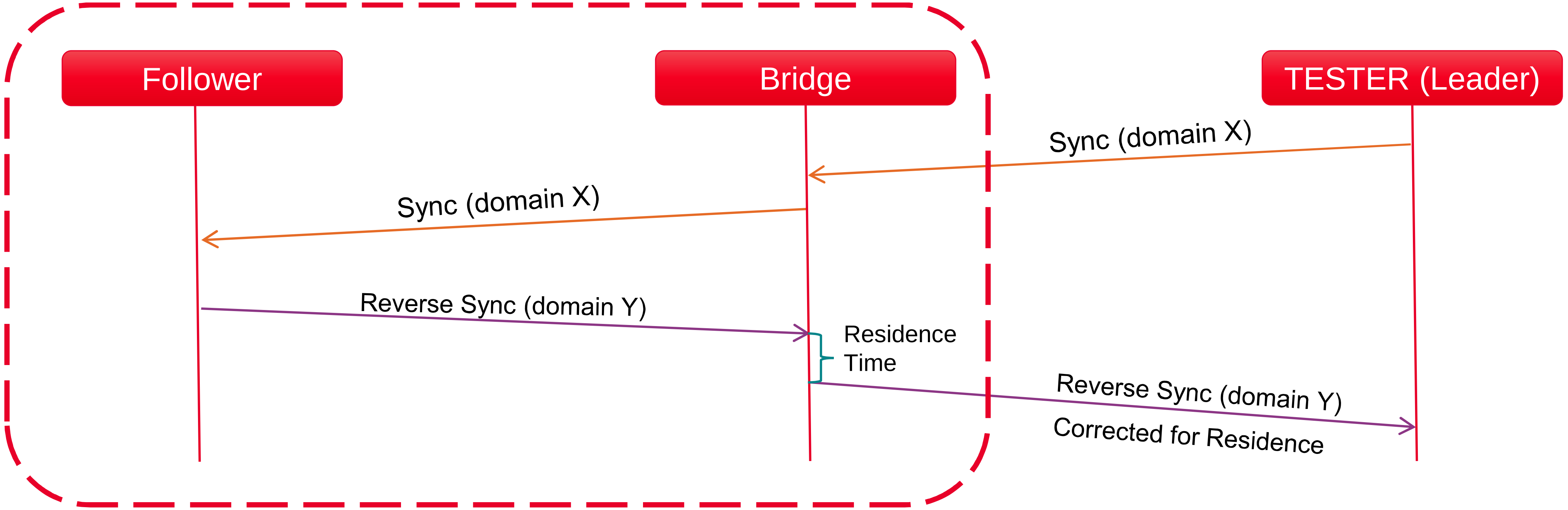
- MeasureReceipt timestamp (may be filled with zeros for P2P)
- PreciseMeasureResponse timestamp

Adjust Reverse Sync rate for better sampling



Advantages for using a Reverse Sync for Integrated Devices

- On integrated devices such as automotive devices that have an integrated Follower + Bridge, there is no observability of the Follower Delay / 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 PTP instance
  - Bridge would just forward the Reverse Sync on its Leader port
- 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 Follower + Bridge

Conclusion and call to action

IEEE 1588 defines how to synchronize time, but not how to validate or monitor time synchronization

- 1588 does provide some tools to help

Without some form of monitoring, time sync errors can go undetected

- Many options are possible

Standardizing the approach will improve adoption and reduce monitoring cost

Call-to-action

- Participate in IEEE 1588 and other standards defining solutions
- Consider testability / monitoring as future tracks for OCP TAP

KEYSIGHT
TECHNOLOGIES