60802 Time Sync – Error Generation Normative Requirements & Next Steps

David McCall – Intel Corporation

Version 2

Content

- Error Generation Normative Requirements PTP Relay Instance
  - Background
  - Expected Behaviour
  - Simulation Results
  - Normative Requirements
- Next Steps
  - Simulations

Background

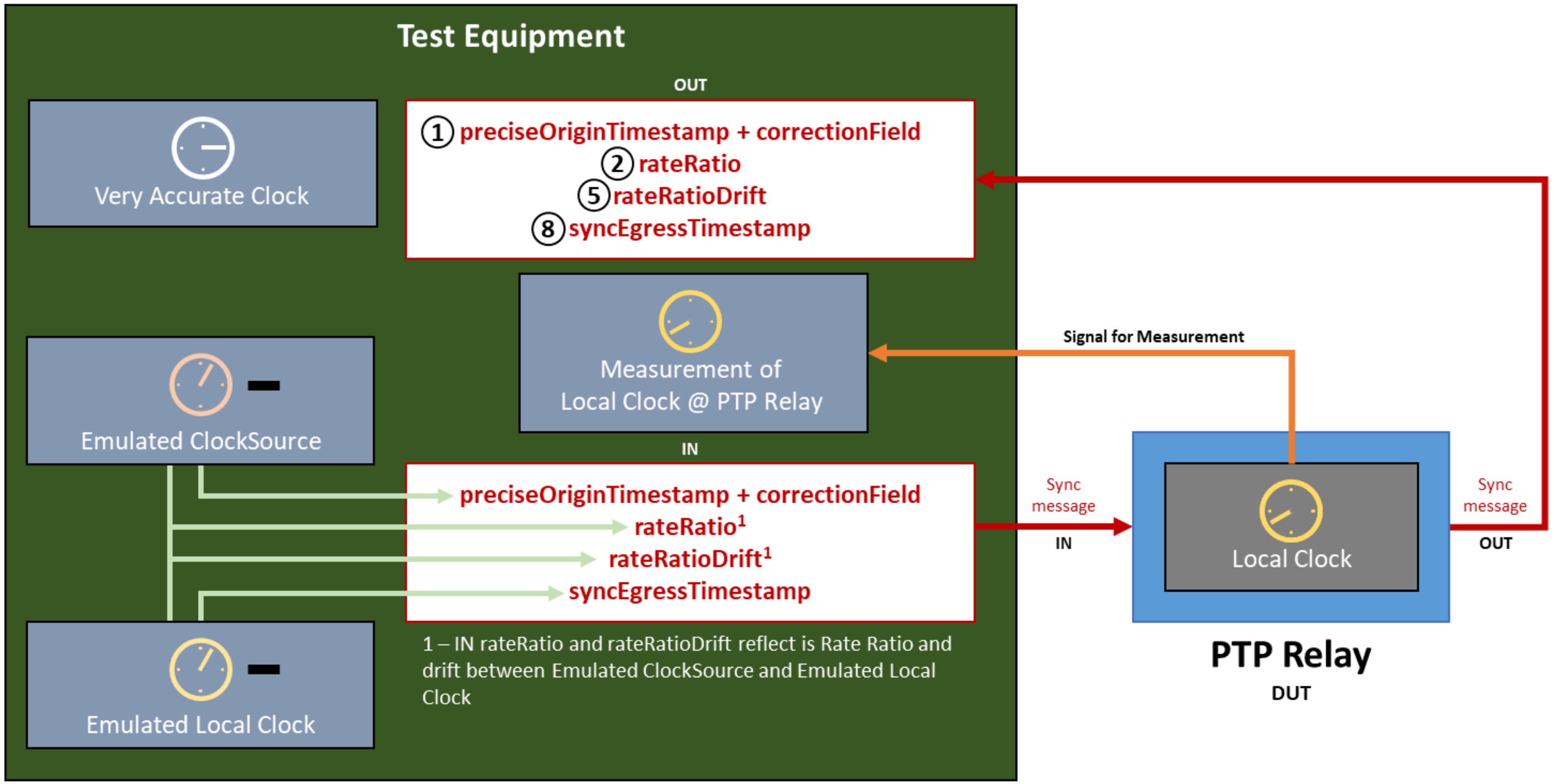
- IEC/IEEE 60802 d2.1 contains normative requirements limiting Error Generation at Relay and End Instances that are estimates and not based on simulations
    - See tables 12 and 13
    - Equivalent limits at Grandmaster Instances (table 11) were based on simulations
- This contribution presents results of Monte Carlo simulations for Relay and End Instances and proposes revised normative requirements.
    - The normative requirements for End Instances are based on simulations that do not include endpoint filtering. Time Series simulations will be required if it is deemed necessary to include endpoint filtering. [For discussion.]

Expected Behaviour

PTP Relay Instance Tests

- Normative requirements are written to make sure a PTP Relay's relevant measurements and outputs are correct…
  - preciseOriginTimestamp + correctionField
    - Or correctionFields in Sync **and** Follow_Up messages
  - rateRatio
  - rateRatioDrift
  - syncOriginTimestamp
    - Is the timestamp accurate vs Relay's Local Clock so that NRR can be measured accurately? Not covered in this presentation.
  - meanLinkDelay **NEW**
    - Part of correctionField adjustment, but difficult to isolate there. Not covered in this presentation.
- And that algorithms are implemented to achieve the necessary performance
  - NRR drift tracking and compensation
    - Measured locally and input to RR drift tracking calculation
  - RR drift tracking and compensation
    - Combination of upstream information (rateRatioDrift input) and local NRR drift tracking

PTP Relay Instance – Test Setup



PTP Relay Instance – Expected Behavior

With and Without Algorithms

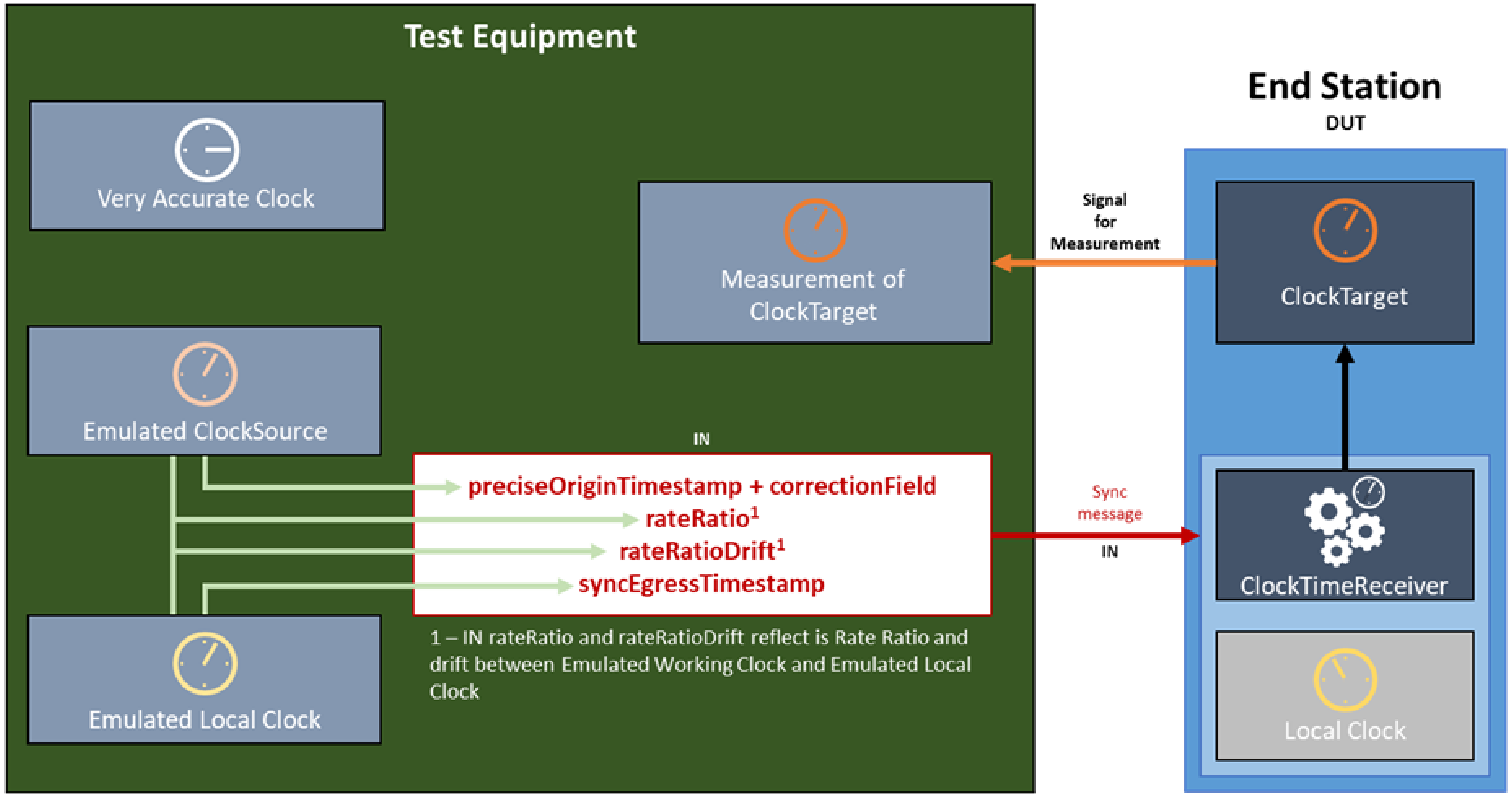
Tester GM Drift	Tester Local Clock	Input rateRatioDrift	Measured NRR Drift	Output preciseOriginTimestamp + correctionField	Output rateRatio	Output rateRatioDrift
-	-	-	-	Output correctionField adjusted **RR x (residenceTime + meanLinkDelay)** No RR drift, so algorithms should have no effect	rateRatio calculated **Input rateRatio + Measured NRR** No NRR or RR drift, so algorithms should have no effect.	-
↑	-	↑	-	correctionField adjusted by **RR x (residenceTime + meanLinkDelay)** RR drift compensation will result is a more accurate value, but no NRR drift, so depends on input rateRatioDrift accuracy.	rateRatio calculated **Input rateRatio + Measured NRR** Output rateRatioRR drift tracking & compensation may modify to account for drift during meanLinkDelay & residenceTime.	↑ If output is correctly calculated as Input rateRatioDrift + Measured NRR Drift…but Measured NRR Drift is zero.
↑	↑	-	↑	correctionField adjusted by **RR x (residenceTime + meanLinkDelay)** RR drift compensation will result is a more accurate value, and depends on accurate.	rateRatio calculated **Input rateRatio + Measured NRR** Measured NRR accuracy may be improved by NRR drift tracking & compensation. Output rateRatioRR drift tracking & compensation may modify to account for drift during meanLinkDelay & residenceTime	↑ If output is correctly calculated as Input rateRatioDrift + Measured NRR Drift and NRR Drift is measured accurately.

- means clock is stable or drift should be measured / communicated as 0 ppm/s
↑ means clock is drifting +1 ppm/s or drift should be measured/communicated as +1 ppm/s
DUT Local Clock is assumed to be sable

PTP End Instance Tests

- Normative requirements are written to make sure a PTP End Instance keeps the ClockTarget in line with the ClockSource based on the input messages.
- And that this is the case in situations that might be expected to fail without the use of drift tracking and compensation algorithms.
  - Algorithms aren't mandated…but requirements may be difficult or impossible to meet without them.

PTP End Instance – Test Setup – Steady State



PTP End Instance – Expected Behavior

With and Without Algorithms

Tester GM Drift	Tester Local Clock	Input rateRatioDrift	ClockTarget
-	-	-	Algorithms have no effect.
↑	-	↑	RR Drift Tracking and Compensation will result in improved accuracy. No NRR Drift, so dependant on accuracy of Input rateRatioDrift
↑	↑	-	RR Drift Tracking and Compensation will result in improved accuracy. NRR Drift is present, so dependant on accuracy of NRR Drift measurement.

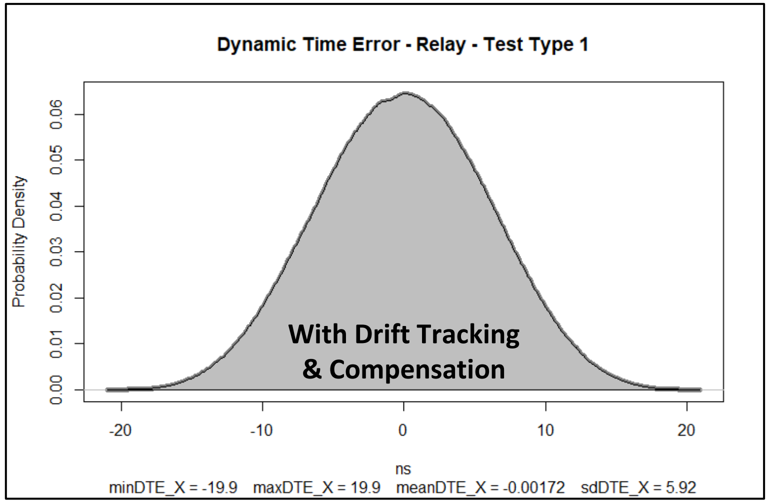
- means clock is stable or drift should be measured / communicated as 0 ppm/s
↑ means clock is drifting +1 ppm/s or drift should be measured/communicated as +1 ppm/s
DUT Local Clock is assumed to be sable

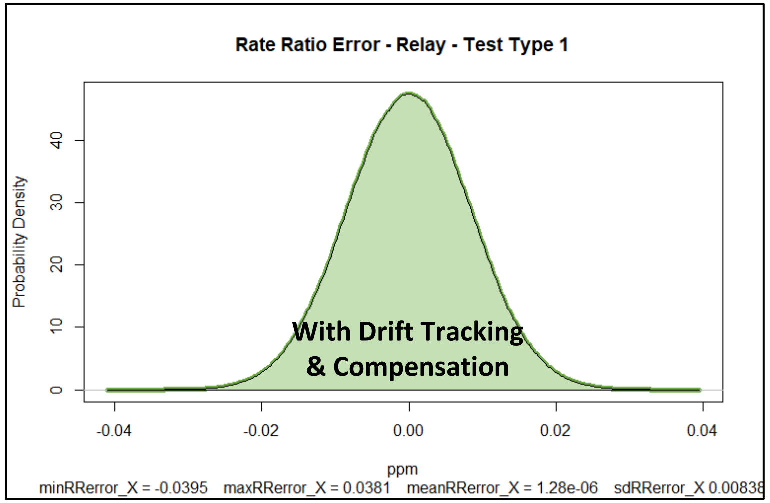
Simulation Results

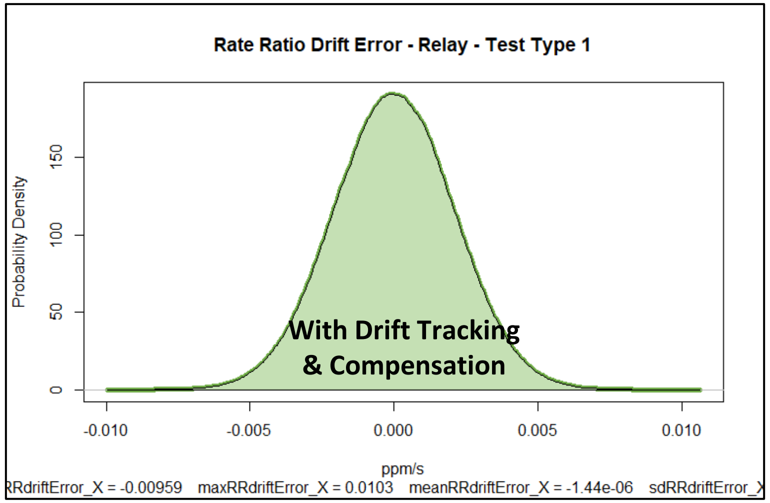
Simulator

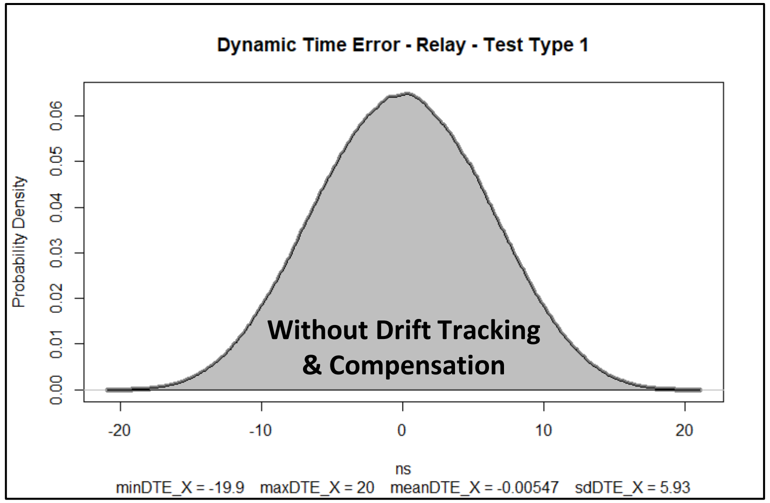
- Simulates test setup.
  - Perfect GM and upstream Local Clock; “clean” input signals.
  - Simulates errors
  - 1,000,000 runs
- Three test types reflect the three behaviours
  - 1 – All clocks stable
  - 2 – GM drift +; other clocks stable
  - 3 – GM and upstream Local Clock drift +; DUT Local Clock stable

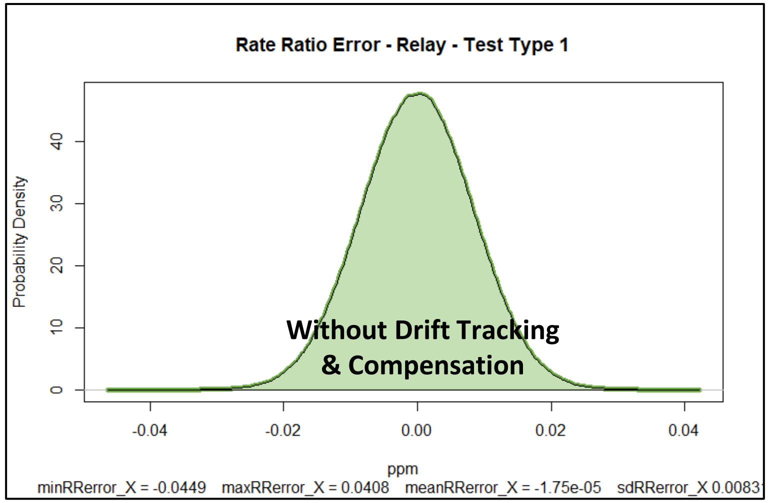
Test Type 1 – All Stable



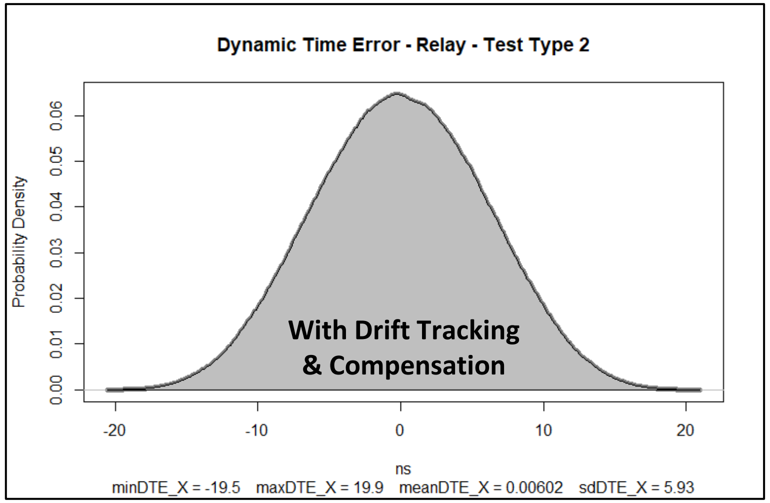


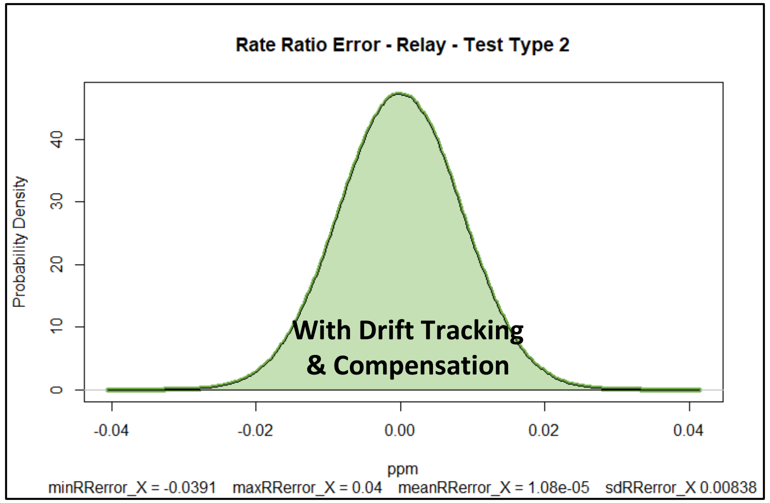


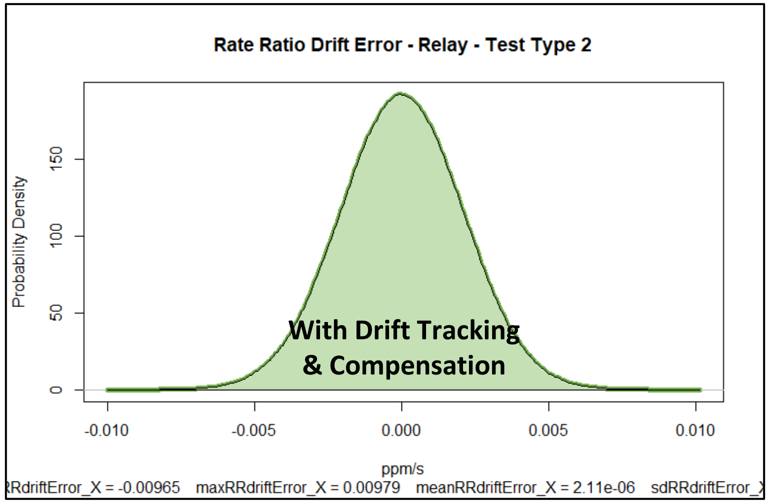


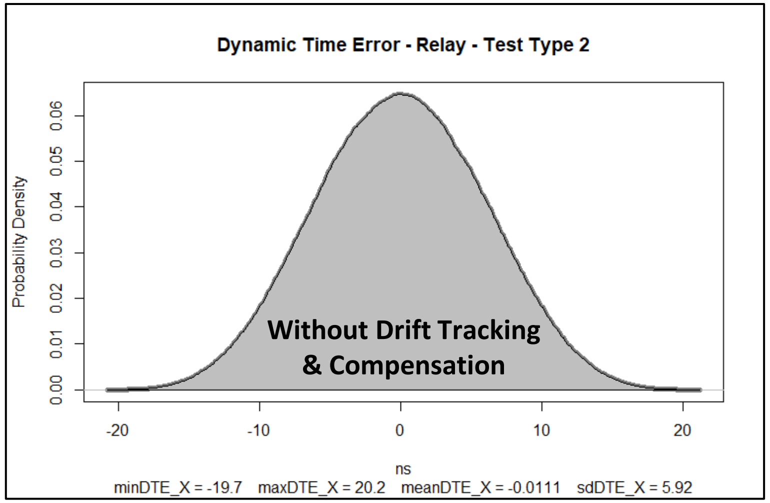


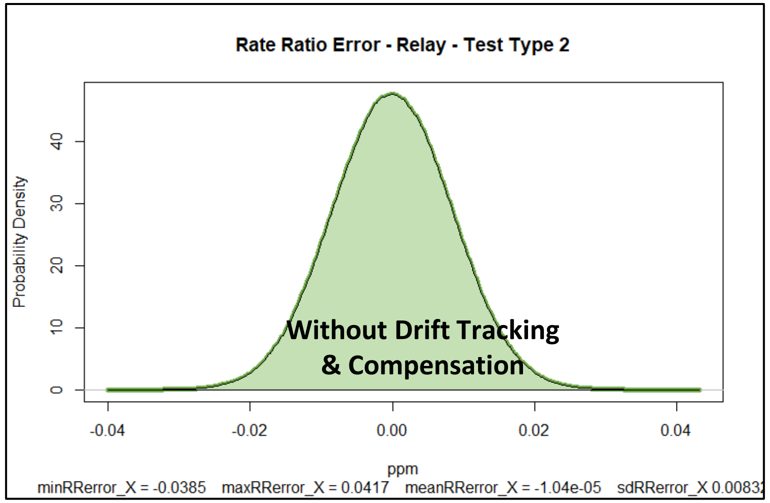
Test Type 2 – GM Drift +



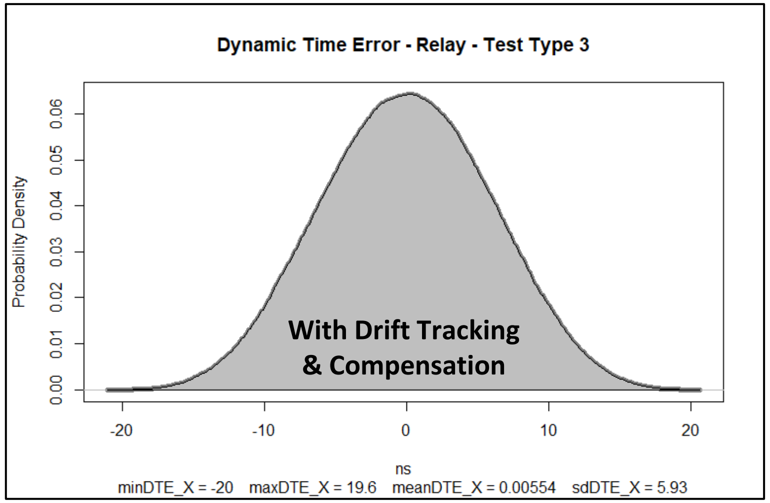


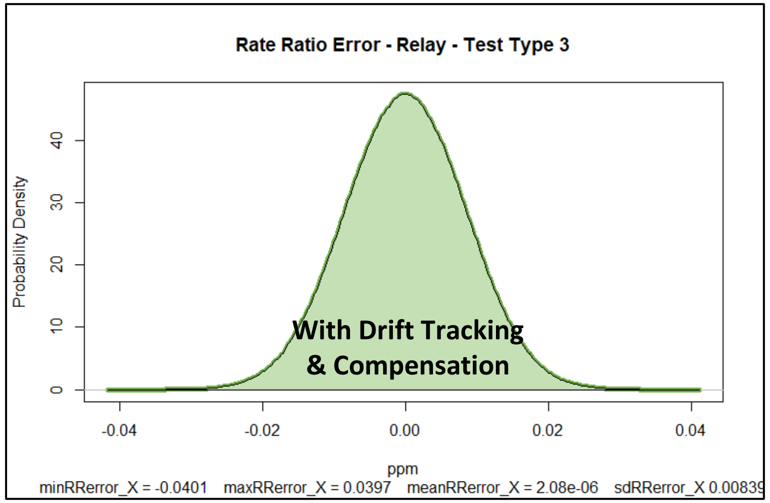


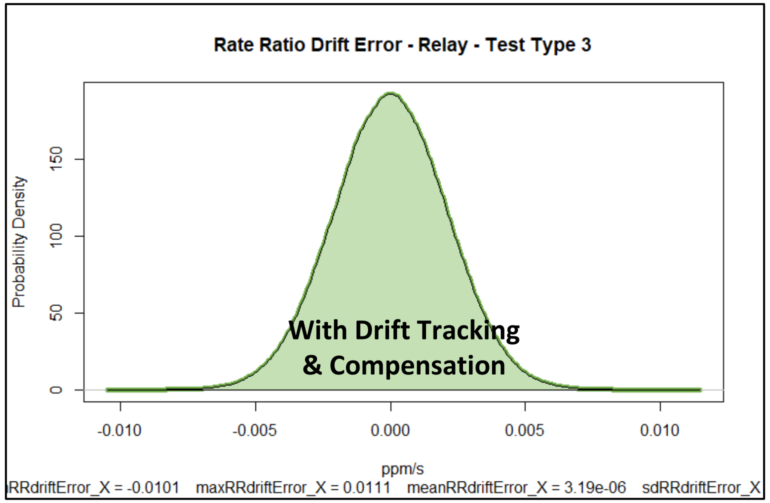


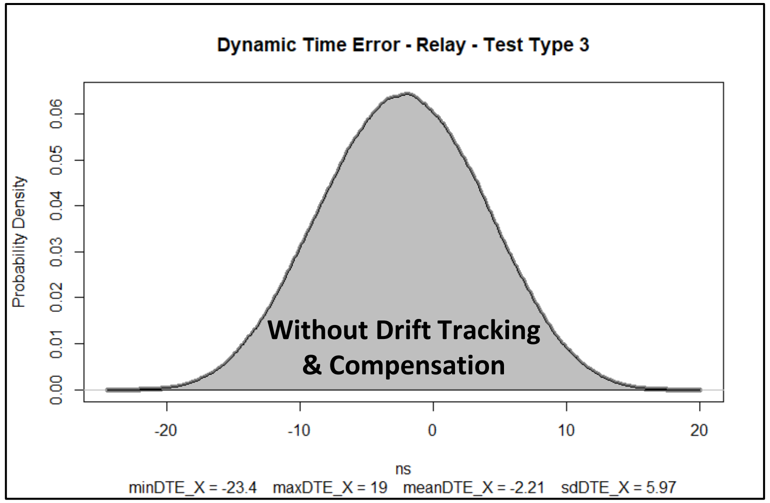


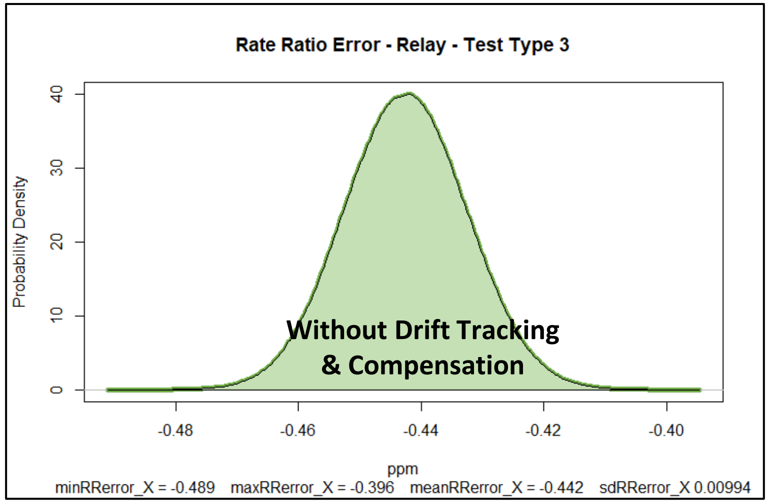
Test Type 3 – GM & Upstream Local Clock Drift +











Simulation Results – PTP Relay Instance

Test Type 1		Mean	Min	Max	SD
dTE (ns)	w	0.0017	-19.9	19.9	5.92
	w/o	0.0055	-19.9	20	5.93
rateRatio (ppm)	w	0	-0.00029	0.00028	6.71E-05
	w/o	0	-0.045	0.041	0.0083
rateRatioDrift (ppm/s)	w	0	-5.50E-07	5.50E+07	1.33E-07
	w/o				

Test Type 2		Mean	Min	Max	SD
dTE (ns)	w	0.006	-19.5	19.9	5.93
	w/o	0.01	-19.9	20.2	5.92
rateRatio (ppm)	w	0	-0.039	1.08E-05	8.40E-03
	w/o	0	-0.385	0.042	0.0083
rateRatioDrift (ppm/s)	w	0	-9.70E-03	9.80E-03	2.10E-03
	w/o				

Test Type 3		Mean	Min	Max	SD
dTE (ns)	w	0.0055	-19.2	20	5.9
	w/o	-2.21	-23.4	19	5.97
rateRatio (ppm)	w	0	-0.04	0.04	8.00E-03
	w/o	-0.44	-0.49	-0.4	0.0099
rateRatioDrift (ppm/s)	w	0	-1.00E-02	3.19E-06	2.10E-03
	w/o				

- Drift tracking / compensation algorithms delivery a slight performance improvement even when there is no drift
  - Suspect this is because it mitigates errors in NRR measurement
- Impact of algorithms on Test Type 2 dTE is limited
  - No NRR drift, so no related error; no error in RR at start (perfect input); limited time for RR drift to have any impact.
- Big impact of algorithms on Test Type 3
  - Output rateRatio is wrong without algorithms by mean 0.44 ppm
  - dTE mean is shifted by -2.21 ns

Normative Requirements Discussion

Discussion

- Existing “estimate” normative requirements look good following Monte Carlo simulations
  - Good estimates!
- Verification via Time Series simulations would be very useful.

Next Steps

Next Steps

- Monte Carlo Simulations of End Instance
  - Might be fixed by tomorrow.
  - Won’t include endpoint filtering
- Time Series Simulations of Test Cases
  - GM, Relay & End Instance
  - Particularly important for End Instance

Time Series Simulations to be Run

- PTP Relay Instance – Test Case Simulation [3 Nodes]
  - “Perfect” GM and Upstream Node
  - Test Types 1, 2 & 3
  - Outputs of interest: dTE, rateRatio, rateRatioDrift
    - “Unfiltered” results when Sync is transmitted (not when next Sync message arrives)
- End Instance (with endpoint filtering) – Test Case Simulation [3 Nodes]
  - Test Types 1, 2 & 3
  - Outputs of interest: dTE (filtered and unfiltered)
- For both…
  - meanLinkDelay averaging: factor 1000; initialise with first path delay measurement.
  - When drifting clock offsets, drift from -100 ppm to +100 ppm (at both GM and “upstream” Local Clock); will take 200 seconds.

Monte Carlo Simulation to be Run

- Sync message “collision” / “swapping” likelihood over 100 hops.

Thank you

Default Configuration

```
hops <- 100 # Minimum 1 hop
runs <- 100000
#
# Input Errors, Parameters & Correction Factors
driftType <- 3 # 1 = DO NOT USE - Historical - Uniform Probability Distribution between MIN & MAX ppm/s
        # 2 = Probability Based on Linear Temp Ramp
        # 3 = Probability Based on Half-Sinusoidal Temp Ramp
        # 4 = Probability Based on Quarter-Sinusoidal Temp Ramp
# Clock Drift Probability from Temp Curve & XO Offset/Temp Relationship
tempMax <- +85 # degC - Maximum temperature
tempMin <- -40 # degC - Minimum temperature
tempRampRate <- 1 # degC/s - Drift Rate for Linear Temp Ramp
tempRampPeriod <- 95 # s - Drift Period for Sinusoidal & Half-Sinusoidal Temp Ramps
tempHold <- 30 # s - Hold Period at MIN and MAX temps before next temp ramp down or up
GMscale <- 0.5 # Ratio of GM stability vs. standard XO. 1 is same. 0 is perfectly stable.
nonGMscale <- 1 # Ratio of non-GM (and non-ES) node stability vs. standard XO. 1 is same. 0 is perfectly stable.
```

Default Configuration

TSGEtx <- 4 # +/- ns - Error due to Timestamp Granularity on TX

TSGErx <- 4 # +/- ns - Error due to Timestamp Granularity on RX

DTSEtx <- 6 # +/- ns - Dynamic Timestamp Error on TX

DTSErx <- 6 # +/- ns - Dynamic Timestamp Error on RX

syncInterval <- 125 # ms - Nominal Interval between two Sync messages

SImode <- 3 # Mode for generating Tsync2sync *HARD CODED to MODE 3*

1 = Gamma Distribution, defaulting to 90% of Tsync2sync falling within 10% of the nominal syncInterval. Truncated at SImax (higher values above are reduced to SImax)

No truncation of low values

2 = Gamma Distribution, defaulting to 90% of Tsync2sync falling within 10% of the nominal syncInterval. Truncated at SImax (higher values are reduced to SImax)

Truncated at SImin (lower values are increased to SImin)

3 = Uniform, linear distribution between syncInterval x SImin and syncInterval x SImax

Default Configuration

SIscale <- 1 # Scaling factor for Mode 1 & 2 Tsync2sync vs regular distribution.

Scaling factor of 3 would mean 90% of Tsync2sync falling within 30% of the nominal syncInterval

SImax <- 1.048 # For mode 1 & 2, Max truncation factor (e.g. 2x syncInterval) limit for Tsync2sync; higher values reduced to SImax

For mode 3, upper limit of uniform linear distribution

SImin <- 0.952 # For mode 1 & 2, Min truncation factor (e.g. 0.5 x syncInterval) limit for Tsync2sync; higher values reduced to SImin

For mode 3, lower limit of uniform linear distribution

pathDelayMin <- 5 # ns - 1m cable = 5ns path delay

pathDelayMax <- 500 # ns - 100m cable = 500ns path delay

residenceTime <- 15 # ms - TresidenceTime maximum; higher values truncated

RTmin <- 1 # TresidenceTime minimum; lower values truncated

RTmean <- 5 # TresidenceTime mean

RTsd <- 1.8 # TresidenceTime sigma; 3.4ppm will fall outside 6-sigma either side of the mean

Default Configuration

MLDerrorSDx6 <- 1.13 # ns - 6x Standard Deviation of meanLinkDelay Error

    # Calculated from separate simulation.

    # 1.13 ns is for steady state after IIR filter with alpha = 1000 has stabalised

mNRRsmoothingNA <- 4 # Whole Number >=1 - Combined N & A value for "smoothing" calculated mNRR (mNRRc)

    # Calculate mNRR using timestamps from Nth Sync message in the past

    # Then take average of previous A mNRRcalculations.

mNRRcompNAP <- 8 # Whole Number >=1

    # For NRR drift rate error correction calculations, take two measurements, mNRRa and mNRRb.

    # Both use timestamps from Nth Sync message in the past, then take average of previous A calculations.

    # Calculation mNRRb starts P calculations in the past from mNRRa, where P = mNRRcompNAP * 2.

    # If 0, there is no NRR drift rate error correction.