60802 Time Synchronisation – Monte Carlo Analysis: 100-hop Model, “Linear” Clock Drift\(^1\), NRR Accumulation\(^2\) Overview & Details, Including Equations – v2

David McCall (Intel)

1 – The model includes various options for modelling Clock Drift distribution, but assumes Clock Drift can be considered linear over the short periods of interest.

2 – The model implements calculating Rate Ratio via an accumulation of Neighbor Rate Ratio (NRR) vs. calculating it directly via Sync messages.
References – 1


References – 2


Background

• IEC/IEEE 60802 has a stated requirement of 1us time accuracy over 64 hops (i.e. 65 devices) with a goal of 100 hops (i.e. 101 devices).

• Prior to the development of the Monte Carlo Analysis, simulations of different configurations and parameters were carried out via Time Series Simulation. See [1], [2], [3], [4] and [5]. Typically...
  • 1 replication simulates 3,100 seconds.
  • To generate statistically significant results, 300 replications are run.
  • This takes 1 to 2 weeks, depending on various parameters.

• The Monte Carlo Analysis presented in these slides was developed over several months in order to provide faster iterations, albeit at the cost of some accuracy.
  • To generate the equivalent number of Sync message simulations as 300 replications of the Time Series Simulation (7,440,000) takes 10-16 minutes
  • The analysis also enables deep insights into the source of errors and how they accumulate.
Content

- History & Current Status
- Overview & Assumptions
- Timestamp Errors
- Clock Drift Errors
- Error Contributions & Accumulation
- Main Equations
- Tracking Error Contributions & Graphical Representations
- Algorithmic Improvements & Corrections
History & Current Status

• The development history of the Monte Carlo Analysis is mostly covered in a series of contributions to IEC/IEEE 60802. See [6], [7], [8] and [9].

• Over the course of development, modelling of additional errors was added to the original model as well as options to model Clock Drift distributions based on different temperature time-series temperature ramps.

• This contribution describes the current operation of the model and the Excel workbook used for post-processing some results.

• It is still intended to open-source the R Studio script which implements the model, although the date is TBD.
Overview & Assumptions
Overview – 1

• Implemented in R Studio (IDE for R)

• Models individual “runs”: single Sync message passing down a chain of nodes with all significant, associated errors.

• Script generates results for Hop 1 for all runs (typically 10,000 to several million)...then Hop 2...then Hop 3...

• Calculations are the same for every hop, with two exceptions:
  • Hop 1: first node is GM
  • Last Hop: no Residence Time; instead, there is End Station Error

• Script tracks error contributions from different sources
  • Full results (values for every error and contributing factor) for each node are calculated, but are not saved, which reduces memory footprint. Maximum absolute, mean and standard deviation values for every contributing error at every node across all runs are saved, e.g. the maximum absolute error across all runs of the contribution to Residence Time Error due to the timestamp component of Neighbor Rate Ratio via Rate Ratio at node 15 is saved.
  • Full results for final node are saved.
Overview – 2

• The script only models errors and values necessary to derive errors. Example: it does not model the Correction Field, only errors associated with it; it only models the period between pDelayResp messages to calculate the error due to Clock Drift during that interval.

• The script only models errors associated with the messaging protocol. It does not model Clock Source, Clock Target, Clock Master or Clock Slave. Neither does it model any filtering of the messaging information, i.e. it’s results are most directly comparable to the “unfiltered” results from the Time Series simulation.
  • The script can be thought of, for each run, as focussed on modelling the error in the Correction Field when it arrives at the last node (usually in a Follow-up message after a Sync message) which is, at the time, the best estimate the node has of GM time. It then adds additional errors due to Rate Ratio and Clock Drift as the last node tries to track GM time prior to the arrival of the next Sync message (and Follow-up message).
RStudio Script Summary

Configuration

Inputs
(Errors, Parameters, Correction Factors)

Calculations per Hop

Graphs
(Tracking Analytics per hop; final distributions)

Output
(Key parameters output to file for offline analysis)

Generate Base Errors
(Timestamp Errors & Clock Drifts)

Calculate mNRR Error
(& components)

Calculate RR Error
(& components)

Calculate Mean Link Delay Error & Residence Time / End Station Error
(& components)

Calculate Tracking Analytics
(Max Absolute, Mean, Sigma for each component per hop)
Assumptions – 1

• The model assumes it is sufficient to account for only the major error contributors and only in enough detail to draw useful conclusions.
  • It doesn’t model errors that would be swamped other larger errors in all realistic scenarios. Example: the effect of Rate Ratio error on Mean Link Delay Error, see [6].
  • It doesn’t model the detail of ambient temperature on a physical system. Instead models a simple temperature ramp on a crystal oscillator (XO); the latter modelled as a cubic equation approximating the relationship between temperature and frequency offset. (This is the same as the Time Series Simulations.)
    • Note: a simpler model which generates clock drift based on a uniform probability between two values, i.e. no temperature modelling, is also available.
Assumptions – 2

• The model assumes that XO Clock Drift can be treated as linear within a single run, i.e. that for each clock a drift rate can be generated once and used for all calculations for that run.
  • This is a major simplification; one that places limits on the model’s ability to model algorithms that attempt to correct for errors due to Clock Drift.

• The model assumes that some errors are uncorrelated due to the amount of time passing between their generation.
  • Example: timestamp granularity for pDelayReq and pDelayResp messages. See slide ZZZ.

• The model assumes that small (<20) ppm values can be added instead of carrying out a more accurate multiplication calculation
  • Example: the model assumes that 3ppm + 6ppm = 9ppm. The accurate value is 9.00018ppm.
  • This simplification saves processing time. It is assumed that the resulting inaccuracy is small compared with other errors that are being modelled.
  • 802.1AS makes the same assumption when calculating RR via accumulated NRR values
Assumptions – 3

• The model assumes that modelling errors generated between processing of Sync messages at only the last node in the chain is sufficient.
  • At all other nodes, only errors related to processing Sync messages are modelled.
  • The assumption is that although, for an individual run, the worst case DTE may not occur at the final node, the overall probability distribution of DTE will be worst at the final node.

• The model does not account for errors due to path delay asymmetry on the assumption that they are a) small relative to other errors and b) will tend to balance out over a long chain of hops.
Assumptions – 4

• The model assumes that there is no effective difference, as far as the errors ultimately being modelled is concerned, between 1-step and 2-step Sync messaging. For simplicity it therefore does not model behaviour related to Follow-up messaging.
  • The errors ultimately being modelled are at the End Station at the end of the chain, just prior to the arrival of the next Sync message.
## Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pDelayInterval</code></td>
<td>1,000</td>
<td>ms</td>
<td>Limited to 1s x 2^n. Typical values are: 1,000ms; 500ms; 250ms; 125ms; 62.5ms; 31.25ms</td>
</tr>
<tr>
<td><code>syncInterval</code></td>
<td>125</td>
<td>ms</td>
<td>Limited to 1s x 2^n. Typical values are: 1,000ms; 500ms; 250ms; 125ms; 62.5ms; 31.25ms</td>
</tr>
<tr>
<td><code>pDelayTurnaround</code></td>
<td>10</td>
<td>ms</td>
<td></td>
</tr>
<tr>
<td><code>residenceTime</code></td>
<td>10</td>
<td>ms</td>
<td></td>
</tr>
</tbody>
</table>

All equations can be traced back to constants, **Parameters**, **Timestamp Errors**, errors due to **Clock Drift**, or **Correction Parameters**
Error Contributors & Error Accumulation
Time Sync – Elements & Relationships

- GM Origin Timestamp
- Correction Field
  - Rate Ratio
- Ongoing End Station Estimate of GM Time
- Correction Field Adjustment
  - (meanLinkDelay + residenceTime) x RR
- Neighbor Rate Ratio
- Mean Link Delay
- Residence Time
meanLinkDelay

\[ \text{meanLinkDelay} = \frac{(t_4 - t_1) - \frac{(t_3 - t_2)}{NRR}}{2} \text{ ns} \]
Measured Neighbor Rate Ratio (mNRR)

\[ mNRR = \frac{(t_3' - t_3)}{(t_4 - t_4')} \]
Residence Time

$$residenceTime = (t_{1out} - t_{2in})$$ ns
Rate Ratio & Correction Field

• Rate Ratio (RR) is calculated via accumulated Neighbor Rate Ratios. At each node the local Rate Ratio (at Node n) is used to estimate the GM clock and passed on to the next node via an outgoing Sync message. It is calculated as follows...

\[ RR(n) = RR(n - 1) + mNRR \]  

\[ \text{correctionField}(n) = \text{correctionField}(n - 1) + RR(\text{meanLinkDelay} + \text{residenceTime}) \]

• The outgoing correction field is calculated as follows...

• The sum of meanLinkDelay (between the current and upstream node) and residenceTime (at the current node) gives the interval between reception of the incoming Sync message and transmission of the outgoing Sync message

• Multiplication by RR translates this from Local Clock to Working Clock
Sources of Errors

• There are two types of error sources in the model...
  • Timestamp Errors; inaccuracies in measuring when messages are received or transmitted. There are two types.
    • Timestamp Granularity Error (TSGE) related to the measurement resolution.
    • Dynamic Timestamp Error (DTSE) related to accuracies inherent in the implementation, excluding TSGE.
  • Errors due to Clock Drift. If all frequency offsets were stable, there would be no errors due to Clock Drift...but they are not, and the events being modelled take place over a period of time. Thus errors occur due to the difference between...
    • Time when a measurement is effectively taken
    • Time when a measurement is used

The model therefore includes Clock Drift and various relevant intervals, all modelled according to probability distributions.
## Timestamp Error Parameters

- The model includes separate parameters for timestamp errors on transmitted (TX) and received messages (RX)

<table>
<thead>
<tr>
<th>Error</th>
<th>Default</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TSGE_{TX}$</td>
<td>$\pm 4$</td>
<td>ns</td>
</tr>
<tr>
<td>$TSGE_{RX}$</td>
<td>$\pm 4$</td>
<td>ns</td>
</tr>
<tr>
<td>$DTSE_{TX}$</td>
<td>$\pm 4$</td>
<td>ns</td>
</tr>
<tr>
<td>$DTSE_{RX}$</td>
<td>$\pm 4$</td>
<td>ns</td>
</tr>
</tbody>
</table>
Timestamp Error Equations

• Both TSGE and DTSE are modelled via uniform distributions between a maximum and a minimum.
• Timestamp Granularity always results in a timestamp after the event occurred...

\[ \text{Error}_{\text{TGSE}} = \sim U(0, +TSG) \]

...(where TSG is Timestamp Granularity) however, because the consequent errors are always in interval measurements which involve two events and two timestamps, modelling it as an error between ±TSG/2 is equivalent. In the R Studio script the parameter TGSE represents TSG/2...

\[ \begin{align*}
\text{Error}_{\text{TSGEx}} &= \sim U \left( -\frac{TSG}{2}, +\frac{TSG}{2} \right) = \sim U(-\text{TSG}_{\text{TX}}, +\text{TSG}_{\text{TX}}) \\
\text{Error}_{\text{TSGRx}} &= \sim U(-\text{TSG}_{\text{RX}}, +\text{TSG}_{\text{RX}})
\end{align*} \]

• DTSE magnitude and probability distribution is implementation dependant, but implementations that deliver a uniform probability between a minimum and maximum, equally spread either side of zero, are common and a worst case.
  • Triangular or normal distributions will have fewer extreme errors.

\[ \begin{align*}
\text{Error}_{\text{DTSEx}} &= \sim U(-\text{DTSE}_{\text{TX}}, +\text{DTSE}_{\text{TX}}) \\
\text{Error}_{\text{DTSERx}} &= \sim U(-\text{DTSE}_{\text{RX}}, +\text{DTSE}_{\text{RX}})
\end{align*} \]
Clock Drift Error Modelling

• Clock Drift Error is modelled as a combination of Clock Drift and the passage of time during relevant intervals.

• Clock Drift is modelled as a single value for each clock for each run. The script allows a choice of two classes of model for Clock Drift
  • Uniform distribution between a maximum and minimum (ppm/s)
  • Distribution based on a temperature cycle and a theoretical crystal oscillator (XO).

• For the second class, the value is generated in three steps...
  • A time (t): uniform random distribution between 0 and a maximum representing the period of a defined temperature cycle.
  • A temperature cycle that ramps temperature up and down between a minimum and maximum in a defined manner. This is used to translate the time (t) into a temperature value for a theoretical crystal oscillator (tempXO).
  • A frequency offset curve for a theoretical XO, based on measured data from a selection of representative XOs, modelled as a cubic equation. The first derivative of the cubic equation allows translation from the from time (t) and XO temperature (tempXO) to Clock Drift.

• There are also probability models for each of the relevant intervals
Clock Drift Error –
Uniform Probability Clock Drift

• Model includes separate parameters for minimum and maximum clock drift for both GM and non-GM clocks.

• It also includes parameters for the fraction of GM and non-GM clocks that will experience drift.
  • This is done to emulate the behaviour of a temperature cycle that holds steady – usually at the maximum or minimum value – for a period of time. See section on Clock Drift probabilities based on a temperature cycle for more details.
  • The R Studio Script generates a value for Clock Drift and as well as a Yes/No value of 1 or 0 and multiplies them together.
Clock Drift Error – Uniform Probability Clock Drift Parameters

<table>
<thead>
<tr>
<th>Error</th>
<th>Default</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{clockDrift}<em>{GM</em>{\text{max}}}$</td>
<td>+1.5</td>
<td>ppm/s</td>
</tr>
<tr>
<td>$\text{clockDrift}<em>{GM</em>{\text{min}}}$</td>
<td>-1.5</td>
<td>ppm/s</td>
</tr>
<tr>
<td>$\text{clockDriftFraction}_{GM}$</td>
<td>0.8</td>
<td>-</td>
</tr>
<tr>
<td>$\text{clockDrift}_{\text{max}}$</td>
<td>+1.5</td>
<td>ppm/s</td>
</tr>
<tr>
<td>$\text{clockDrift}_{\text{min}}$</td>
<td>-1.5</td>
<td>ppm/s</td>
</tr>
<tr>
<td>$\text{clockDriftFraction}$</td>
<td>0.8</td>
<td>-</td>
</tr>
</tbody>
</table>
Clock Drift Error –
Uniform Probability Clock Drift Equations

\[
\text{Error}_{\text{clockDriftGM}} = \sim U(\text{clockDrift}_{\text{minGM}}, \text{clockDrift}_{\text{maxGM}}) \times \sim B(1, \text{clockDriftFraction}_{\text{GM}})
\]

\[
\text{Error}_{\text{clockDrift}} = \sim U(\text{clockDrift}_{\text{min}}, \text{clockDrift}_{\text{max}}) \times \sim B(1, \text{clockDriftFraction})
\]

- The function in R to generate random values according to a binomial probability distribution has three input parameters
  - \(n\): number of values to generate; not shown above; equal to the number of runs
  - \(N\): number of “trials” e.g. flips of the coin; in this case “1”
  - \(p\): probability of success; in this case the probability of a clock instance experiencing drift, represented by a “1” (vs. “0”)
Clock Drift Error –
Clock Drift from Temp Cycle Modelling

• The temperature variation model has four sections over a complete cycle
  • Ramp from minimum to maximum temperature (Section A)
  • Hold at maximum temperature (Section B)
  • Ramp from maximum to minimum temperature (Section C)
  • Hold at minimum temperature (Section D)

• The R Studio script supports three types of temperature ramp model
  • Linear
  • Sinusoidal
  • Half-sinusoidal

• The ramp is defined by temperature rate of change for linear ramp; duration of ramp for sinusoidal and half-sinusoidal. The hold period at maximum and minimum temperature is the same.

• The model for the XO’s frequency offset is the same for all types of temperature ramp
Clock Drift Error – Offset Frequency Curve

Clock Drift Error – Offset Frequency Curve

Clock Drift Error – Offset Frequency Curve

Clock Drift Error – Offset Frequency Curve

Clock Drift Error – Offset Frequency Curve

$freq\text{Offset} = a \cdot \text{tempXO}^3 + b \cdot \text{tempXO}^2 + c \cdot \text{tempXO} + d$

<table>
<thead>
<tr>
<th>Cubic Constants</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>0.00012</td>
</tr>
<tr>
<td>$b$</td>
<td>-0.01005</td>
</tr>
<tr>
<td>$c$</td>
<td>-0.0305</td>
</tr>
<tr>
<td>$d$</td>
<td>5.73845</td>
</tr>
</tbody>
</table>

From Geoff Garner, “Phase and Frequency Offset, and Frequency Drift Rate Time History Plots Based on New Frequency Stability Data”, contribution to IEC/IEEE 60802, March 2021

The calculation of freqOffset is not used in the model but is included for completeness and in case the reader wishes to recreate the example graphs.
Clock Drift Error – Temperature Cycle Parameters

• The model includes a parameter to scale the Clock Drift up or down to emulate less or more accurate XOs. The default of 1 means no scaling; 0.5 results in half the amount; 2 in twice the amount. (The model has a separate parameter for GM node scaling and one for non-GM node scaling, but the equations on subsequent pages refer only to a single parameter of “scale”, since all other elements of the equations are the same.)

<table>
<thead>
<tr>
<th>Error</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempMax</td>
<td>85</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>tempMin</td>
<td>-20</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>tempRampRate</td>
<td>±1</td>
<td>°C/s</td>
<td>Only used for linear temperature ramp</td>
</tr>
<tr>
<td>tempRampPeriod</td>
<td>125</td>
<td>s</td>
<td>Only used for sinusoidal and half-sinusoidal temperature ramps</td>
</tr>
<tr>
<td>tempHold</td>
<td>30</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td>GMscale</td>
<td>1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>nonGMscale</td>
<td>1</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Clock Drift Error –
Linear Temp Ramp Ramp Equations – 1

<table>
<thead>
<tr>
<th>Input</th>
<th>Default Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempMax</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>tempMin</td>
<td>-20</td>
<td>°C</td>
</tr>
<tr>
<td>tempRampRate</td>
<td>±1</td>
<td>°C/s</td>
</tr>
<tr>
<td>tempHold</td>
<td>30</td>
<td>s</td>
</tr>
</tbody>
</table>

\[
\text{tempCyclePeriod} = 2 \times \left( \frac{\text{tempMax} - \text{tempMin}}{\text{tempRampRate}} + \text{tempHold} \right)
\]

\[
\text{sectionA} = \frac{\text{tempMax} - \text{tempMin}}{\text{tempRampRate}}
\]

\[
\text{sectionB} = \text{sectionA} + \text{tempHold}
\]

\[
\text{sectionC} = \text{sectionB} + \text{sectionA}
\]
Clock Drift Error –
Linear Temp Ramp Equations – 2

\[ t = \sim U(0, tempCyclePeriod) \]

\[
\text{if } (0 \leq t < \text{sectionA}) \\
\begin{align*}
\text{tempXO} &= \text{tempMin} + \text{tempRampRate} \cdot t \\
\text{tempRoC} &= \text{tempRampRate} \\
\text{clockDrift} &= (3 \cdot a \cdot \text{tempXO}^2 + 2 \cdot b \cdot \text{tempXO} + c) \times \text{tempRampRate} \times \text{scale}
\end{align*}
\]

\[
\text{if } (\text{sectionA} \leq t < \text{sectionB}) \\
\begin{align*}
\text{tempXO} &= \text{tempMax} \\
\text{tempRoC} &= 0 \\
\text{clockDrift} &= 0
\end{align*}
\]

\[
\text{if } (\text{sectionB} \leq t < \text{sectionC}) \\
\begin{align*}
\text{tempXO} &= \text{tempMax} - \text{tempRampRate} \cdot (t - \text{sectionB}) \\
\text{tempRoC} &= -\text{tempRampRate} \\
\text{clockDrift} &= -(3 \cdot a \cdot \text{tempXO}^2 + 2 \cdot b \cdot \text{tempXO} + c) \times \text{tempRampRate} \times \text{scale}
\end{align*}
\]

\[
\text{if } (\text{sectionC} \leq t) \\
\begin{align*}
\text{tempXO} &= \text{tempMin} \\
\text{tempRoC} &= 0 \\
\text{clockDrift} &= 0
\end{align*}
\]

The calculation of tempRoC is not used in the model but is included for completeness and in case the reader wishes to generate the example graphs.
Clock Drift Example – Linear Temperature Ramp: 1°C/s † (125s ††)

Inputs

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp Max</td>
<td>85°C</td>
</tr>
<tr>
<td>Temp Min</td>
<td>-40°C</td>
</tr>
<tr>
<td>Temp Ramp Rate</td>
<td>1°C/s</td>
</tr>
<tr>
<td>Temp Hold</td>
<td>30s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temp Rate of Change</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX</td>
<td>1.00°C/s</td>
</tr>
<tr>
<td>MIN</td>
<td>-1.00°C/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clock Drift</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX</td>
<td>1.35 ppm/s</td>
</tr>
<tr>
<td>MIN</td>
<td>-1.35 ppm/s</td>
</tr>
</tbody>
</table>
Clock Drift Error –
Sinusoidal Temp Ramp Equations – 1

<table>
<thead>
<tr>
<th>Input</th>
<th>Default Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>tempMax</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>tempMin</td>
<td>-20</td>
<td>°C</td>
</tr>
<tr>
<td>tempRampPeriod</td>
<td>125</td>
<td>s</td>
</tr>
<tr>
<td>tempHold</td>
<td>30</td>
<td>s</td>
</tr>
</tbody>
</table>

\[
tempCyclePeriod = 2 \times (\text{tempRampPeriod} + \text{tempHold})
\]

\[
tempDeviation = \frac{\text{tempMax} - \text{tempMin}}{2}
\]

\[
tempMidpoint = \text{tempMax} - \text{tempDeviation}
\]

\[
\omega = \frac{\pi}{\text{tempRampPeriod}}
\]

\[
sectionA = \text{tempRampPeriod}
\]

\[
sectionB = sectionA + \text{tempHold}
\]

\[
sectionC = sectionB + sectionA
\]
Clock Drift Error – Sinusoidal Temp Ramp Equations – 2

\[ t = \sim U(0, tempCycle\text{\_}Period) \]

\[
\text{if } (0 \leq t < \text{sectionA})
\]

\[
\begin{align*}
tempXO &= tempMidpoint - tempDeviatio_n \cdot \cos(\omega \cdot t) \\
tempRoC &= \omega \cdot tempDeviatio_n \cdot \sin(\omega \cdot t) \\
clockDrift &= (3 \cdot a \cdot tempXO^2 + 2 \cdot b \cdot tempXO + c) \times (\omega \cdot tempDeviatio_n \cdot \sin(\omega \cdot t)) \cdot \text{scale}
\end{align*}
\]

\[
\text{if } (\text{sectionA} \leq t < \text{sectionB})
\]

\[
\begin{align*}
tempXO &= tempMax \\
tempRoC &= 0 \\
clockDrift &= 0
\end{align*}
\]

\[
\text{if } (\text{sectionB} \leq t < \text{sectionC})
\]

\[
\begin{align*}
tempXO &= tempMidpoint + tempDeviatio_n \cdot \cos(\omega \cdot (t - \text{sectionB})) \\
tempRoC &= -\omega \cdot tempDeviatio_n \cdot \sin(\omega \cdot (t - \text{sectionB})) \\
clockDrift &= -(3 \cdot a \cdot tempXO^2 + 2 \cdot b \cdot tempXO + c) \times (\omega \cdot tempDeviatio_n \cdot \sin(\omega \cdot (t - \text{sectionB}))) \cdot \text{scale}
\end{align*}
\]

\[
\text{if } (\text{sectionC} \leq t)
\]

\[
\begin{align*}
tempXO &= tempMin \\
tempRoC &= 0 \\
clockDrift &= 0
\end{align*}
\]

The calculation of \( \text{tempRoC} \) is not used in the model but is included for completeness and in case the reader wishes to generate the example graphs.
Clock Drift Example – Sinusoidal Temperature Ramp: 125s ↑↓

Inputs
- Temp Max: 85°C
- Temp Min: -40°C
- Temp Ramp Period: 125s
- Temp Hold: 30s

Temp Rate of Change (°C/s)
- MAX: 1.57°C/s
- MIN: -1.57°C/s

Clock Drift (ppm/s)
- MAX: 0.76 ppm/s
- MIN: -0.76 ppm/s
Clock Drift Error –
Half-sinusoidal Temp Ramp Equations – 1

<table>
<thead>
<tr>
<th>Input</th>
<th>Typical Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tempMax</code></td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td><code>tempMin</code></td>
<td>-20</td>
<td>°C</td>
</tr>
<tr>
<td><code>tempRampPeriod</code></td>
<td>125</td>
<td>s</td>
</tr>
<tr>
<td><code>tempHold</code></td>
<td>30</td>
<td>s</td>
</tr>
</tbody>
</table>

\[
\text{tempCyclePeriod} = 2 \times (\text{tempRampPeriod} + \text{tempHold})
\]

\[
\text{tempRange} = \text{tempMax} - \text{tempMin}
\]

\[
\tau = \frac{\pi}{\text{tempRampPeriod} \times 2}
\]

\[
\text{sectionA} = \text{tempRampPeriod}
\]

\[
\text{sectionB} = \text{sectionA} + \text{tempHold}
\]

\[
\text{sectionC} = \text{sectionB} + \text{sectionA}
\]
Clock Drift Error –
Half-sinusoidal Temp Ramp Equations – 2

\[ t = \sim U(0, \text{tempCyclePeriod}) \]

\[
\text{if } (0 \leq t < \text{sectionA})
\]

\[
\begin{align*}
\text{tempXO} &= \text{tempMin} + \text{tempRange}. \sin(\tau \cdot t) \\
\text{tempRoC} &= \tau \cdot \text{tempRange}. \cos(\tau \cdot t)
\end{align*}
\]

\[
\text{clockDrift} = (3 \cdot a \cdot \text{tempXO}^2 + 2 \cdot b \cdot \text{tempXO} + c) \times (\tau \cdot \text{tempRange}. \cos(\tau \cdot t)) \cdot \text{scale}
\]

\[
\text{if } (\text{sectionA} \leq t < \text{sectionB})
\]

\[
\begin{align*}
\text{tempXO} &= \text{tempMax} \\
\text{tempRoC} &= 0
\end{align*}
\]

\[
\text{clockDrift} = 0
\]

\[
\text{if } (\text{sectionB} \leq t < \text{sectionC})
\]

\[
\begin{align*}
\text{tempXO} &= \text{tempMax} - \text{tempRange}. \sin(\tau \cdot (t - \text{sectionB})) \\
\text{tempRoC} &= -\tau \cdot \text{tempRange}. \cos(\tau \cdot (t - \text{sectionB}))
\end{align*}
\]

\[
\text{clockDrift} = -(3 \cdot a \cdot \text{tempXO}^2 + 2 \cdot b \cdot \text{tempXO} + c) \times (\tau \cdot \text{tempRange}. \cos(\tau \cdot (t - \text{sectionB}))) \cdot \text{scale}
\]

\[
\text{if } (\text{sectionC} \leq t)
\]

\[
\begin{align*}
\text{tempXO} &= \text{tempMin} \\
\text{tempRoC} &= 0
\end{align*}
\]

\[
\text{clockDrift} = 0
\]

The calculation of tempRoC is not used in the model but is included for completeness and in case the reader wishes to generate the example graphs.
Clock Drift Example – Half-Sinusoidal Temperature Ramp: 125s ↑

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Temp Max</th>
<th>85°C</th>
<th>Temp Min</th>
<th>-40°C</th>
<th>Temp Ramp Period</th>
<th>125s</th>
<th>Temp Hold</th>
<th>30s</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Temp Rate of Change (°C/s)</th>
<th>MAX</th>
<th>1.57°C/s</th>
<th>MIN</th>
<th>-1.57°C/s</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Clock Drift (ppm/s)</th>
<th>MAX</th>
<th>2.12ppm/s</th>
<th>MIN</th>
<th>-1.35ppm/s</th>
</tr>
</thead>
</table>
Clock Drift Error – Relevant Intervals

4 Hops

In this example pDelay Interval is \( \frac{1}{4} \) Sync Interval
Clock Drift Error – Relevant Intervals
4 Hops – 1st Hop

- Error due to drift during NRR measurement (Node 1 to GM)
  - Interval is half $t_1 - t_1'$, which is nominally half the pDelay Interval, but actual pDelay Interval varies.
- Error due to drift between measuring and using NRR (Node 1 to GM)
  - Interval is between zero and the maximum pDelay Interval, which is larger than the nominal pDelay Interval.
  - Assumes that pDelayResp arriving between t2in and t1out will trigger new mNRR calculation; not unreasonable as information is included in Follow-up, not Sync, if 2-step Sync is used.
- Error due to drift during Residence Time measurement (Node 1 to GM)
  - meanLinkDelay is measured separately, is much smaller, and can be averaged to remove errors, so is ignored.
Clock Drift Error – Relevant Intervals

4 Hops – 2\textsuperscript{nd} Hop

- Same errors in mNRR as 1\textsuperscript{st} Hop.
- Error due to drift during NRR measurement. (Node 2 to Node 1)
- Error due to drift between measuring and using NRR. (Node 2 to Node 1)
- Error due to drift during Residence Time measurement. (Node 2 to GM)
- Additional error from drift between RR(1) calculation, at Node 1, and use in calculating RR(2). (Node 1 to GM)
- In the model the contribution from meanLinkDelay is ignored; only Residence Time is used.
Clock Drift Error – Relevant Intervals
4 Hops – 3\(^{rd}\) Hop

- Same errors in NRR and RR as 2\(^{nd}\) Hop.
  - Error due to drift during NRR measurement. (Node 3 to Node 2)
  - Error due to drift between measuring and using NRR. (Node 3 to Node 2)
  - Error due to drift during Residence Time measurement. (Node 3 to GM)
  - Error due to drift between RR(2) calculation, at Node 2, and use in calculating RR(3). (Node 2 to GM)
Clock Drift Error – Relevant Intervals
4 Hops – 4\textsuperscript{th} Hop

• Similar errors in NRR and RR as 2\textsuperscript{nd} & 3\textsuperscript{rd} Hop.
• Error due to drift during NRR measurement. \textbf{(Node 4 to Node 3)}
• Error due to drift between measuring and using NRR. \textbf{(Node 4 to Node 3)}
• Error due to drift between RR(3) calculation, at Node 3, and use starting at receipt of Sync. \textbf{(Node 3 to GM)}
  • Modeled as zero due to the absence of Residence Time at the final node.
• No Residence Time, so no error due to drift during measurement.
• There is additional error during the period until the next Sync message...
Clock Drift Error – Relevant Intervals
4 Hops – 4\textsuperscript{th} Hop

- Error due to drift between receipt of one Sync message and the next. (Node 4 to GM).
- Interval is nominally the Sync Interval, but there is some variation.
- Additional pDelayResp messages, updating mNRR (Node 4 to Node 3) are not useful to update RR.
Clock Drift Error – Relevant Intervals

Summary – 1

- There are six relevant intervals...

1. Effective NRR Measurement → Actual NRR Measurement
   - The relevant drift is between the current node's clock and the upstream node’s clock.
   - NRR is measured via information from a pair of pDelayResp messages. As Clock Drift is assumed to be linear, the effective measurement point is half-way between the two. The actual measurement point is at receipt of the second message.
   - The interval between the two pDelayResp messages is nominally the pDelay Interval. IEEE 1588 defines the permitted minimum and maximum interval as 90% and 130% of the nominal value. [See IEEE 1588-2019 9.5.13.2]
   - The interval is modelled as a uniform distribution between these two.

   \[ T_{p\text{delay2p\text{delay}}} = \sim U(p\text{delayInterval}.0.9, p\text{delayInterval}.1.3) \]

   - Note: see section on Algorithmic Improvements & Corrections for how this calculation changes if an older pDelayResp message is used as the first of the pair.
Clock Drift Error – Relevant Intervals
Summary – 2

2. Actual NRR Measurement → NRR Use

- The relevant drift is between the current node’s clock and the upstream node’s clock.
- For all hops other than the last, NRR is used when RR for the outgoing Sync message is calculated. For the last hop, NRR is used when RR is calculated for the working clock of the local device to use until the arrival of the next Sync message.
- In either case, the interval is modelled as a uniform distribution between zero and the interval between receipt of pDelayResp messages, i.e. random depending on the phase between Sync and pDelay messaging.
- The interval between receipt of pDelayResp messages is modelled in the same way as for the previous error, i.e. between 90% and 130% of the nominal value. [See 802.1AS-2020 9.5.13.2]
Clock Drift Error – Relevant Intervals

Summary – 3

3. RR Calc at upstream Node → RR Use at Current Node

• The relevant drift is between the upstream node’s clock and the GM’s clock.
• For all hops other than the last, the interval is the path delay between the upstream node and the current node plus the residence time at the current node. For the last node there is no residence time, so it’s only the path delay.
• The path delay is sufficiently small that it can ignored. [See assumption that it’s OK to not model small errors that will be swamped by larger ones.]
• For all hops other than the last, the delay is modelled as equal to the residenceTime parameter. For the last hop it is not modelled, i.e. per the previous point, the error is assumed to be small enough to ignore.
Clock Drift Error – Relevant Intervals

Summary – 4

4. Start of Residence Time Measurement → End of Measurement

• The relevant drift is between the current node’s clock and the GM’s clock.

• When calculating the Correction Field, the goal is to measure Residence Time in terms of the GM clock. It is measured using the Local Clock with the interval multiplied by the Rate Ratio to translate it into GM time. There are two sources of error:
  • Error in measuring the interval.
  • Error in Rate Ratio.

Clock drift during this interval affects the former.

• It is modelled as equal to the \textit{residenceTime} parameter.
Clock Drift Error – Relevant Intervals
Summary – 5

5. RR Calc at Receipt of Sync message → Receipt of next Sync Message
   • The relevant drift is between the current node’s clock and the GM’s clock.
   • It is only modeled at for the last hop.
   • It is modeled as a gamma distribution with shape 270.5532 and rate $270.5532/syncInterval$.

   $$T_{\text{SyncToSync}} = \sim \Gamma(270.5532, \frac{270.5532}{\text{syncInterval}})$$

6. Drift during measurement of meanLinkDelay (not shown above; see Equations section)
   • The relevant drift is between the current node’s clock and the upstream node’s clock.
   • It is modeled as equal to the $pDelayTurnaround$ parameter.
Errors Measuring NRR

\[ mNRR_{error} = mNRR_{measured} - mNRR_{nominal} \quad \text{ppm} \]
meanLinkDelay Errors

\[ \text{meanLinkDelay}_{\text{error}} = \text{meanLinkDelay}_{\text{measured}} - \text{meanLinkDelay}_{\text{nominal}} \] in ns
Residence Time

\[
\text{residenceTime}_{\text{error}} = \text{residenceTime}_{\text{measured}} - \text{residenceTime}_{\text{nominal}} \quad \text{ns}
\]
End Station Error

\[
\text{endStation}_{\text{error}} = \text{endStation}_{\text{measured}} - \text{endStation}_{\text{nominal}} \quad \text{ns}
\]
Dynamic Time Sync Error Accumulation

All errors in this analysis are caused by either **Clock Drift** or **Timestamp Errors**

*DTE based on protocol messaging only. Total DTE at the application level will also depend on ClockMaster, ClockSlave, ClockSource, Clock Target, etc...
Equations – Timestamp Errors

• Eight timestamp errors are generated for each node in each run...
  • $t_{1pderror}$
  • $t_{2pderror}$
  • $t_{3pderror}$
  • $t_{4pderror}$
  • $t_{3pderror}'$
  • $t_{4pderror}'$
  • $t_{2sinerror}$
  • $t_{1souterror}$

• $pd$ errors are associated pDelay and pDelayResponse messages used to measure meanLinkDelay. $t_{3pderror}'$ and $t_{4pderror}'$ errors are associated with the previous pDelayResponse message used – along with the most recent one, to measure NRR.

• $sin$ and $sout$ errors are associated with receiving and transmitting Sync messages.

• All timestamp errors use the same equation. For example...

\[
t_{1pderror} = \text{Error}_{TSGE} + \text{Error}_{DTSE}
\]

• Each timestamp error is uncorrelated to any other timestamp error.

• See previous section for definition of $\text{Error}_{TSGE}$ and $\text{Error}_{DTSE}$.
Equations – Clock Drift

• The model tracks the drift of three clocks at each node...
  • \( \text{clockDrift}_{GM} \) – Clock Drift of the Grandmaster clock
  • \( \text{clockDrift}_n \) – Clock Drift of the current node’s clock
  • \( \text{clockDrift}_{n-1} \) – Clock Drift of the upstream node’s clock

• See previous section for details on the equations to generate the \text{clockDrift} values

• \( \text{clockDrift}_{GM} \) is generated once for each run. For the first hop, it is equivalent to \( \text{clockDrift}_{n-1} \), i.e. for the hop, the “upstream node” is the GM

• For all nodes after the first, \( \text{clockDrift}_n \) is copied to \( \text{clockDrift}_{n-1} \) before a new value for \( \text{clockDrift}_n \) is generated.
  • Note: the actual implementation uses two vectors and swaps their function between \( n \) and \( n-1 \) to reduce processing time by eliminating the need to copy data.
Equations – \( mNRR_{\text{error}} \) (Neighbor Rate Ratio)

**Timestamp Error – 1**

\[
mNRR_{\text{error}} = mNRR_{\text{errorTS}} + mNRR_{\text{errorCD}}
\]

\[
mNRR_{\text{errorTS}} = mNRR_{\text{measuredTS}} - mNRR_{\text{nominal}}
\]

\[
= \left( \frac{t_3 + t_3' + t_4 - t_3' + t_4'}{t_4 + t_4' + t_4p - t_4' - t_4p} - 1 \right) \times 10^6 - \left( \frac{t_3 - t_3' - t_4 - t_4'}{t_4 - t_4' - t_4p - t_4p} - 1 \right) \times 10^6
\]

\[
= \left( \frac{t_3 - t_3'}{t_4 - t_4'} - 1 \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_4'} - 1 \right) \times 10^6
\]

The ratio \( X \) in ppm is \( (X - 1) \times 10^6 \).

\( mNRR_{\text{errorTS}} \) and \( mNRR_{\text{errorCD}} \) are not entirely independent, but the effect of the relationship on \( mNRR_{\text{error}} \) can be ignored provided the errors are small compared to \( p\text{DelayTurnaround} \), which they are for \( p\text{DelayTurnaround} \) of 1ms or more. See backup for algorithmic proof.
### Equations – mNRR<sub>error</sub> – Timestamp Error

\[
mNRR_{errorTS} = \left( \frac{t_3 - t_3'}{t_4 - t_4' + t_{4pdelay} - t_{4pdelay}'} - 1 + \frac{t_{3pdelay} - t_{3pdelay}'}{t_4 - t_4' + t_{4pdelay} - t_{4pdelay}'} \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_4'} - 1 \right) \times 10^6
\]

\[
= \frac{(t_4 - t_4')(t_{3pdelay} - t_{3pdelay}') - (t_3 - t_3')(t_{4pdelay} - t_{4pdelay}')}{{(t_4 - t_4' + t_{4pdelay} - t_{4pdelay})}} \times 10^6
\]

\[
= \frac{t_{4pdelay} - t_{4pdelay}'}{T_{pdelay2pdelay} \times 10^6} \times 10^6
\]

\[
= \frac{t_{4pdelay} - t_{4pdelay}'}{T_{pdelay2pdelay} \times 10^6}
\]

\[
\approx \frac{t_{4pdelay} - t_{4pdelay}'}{10^6}
\]

The error magnitudes are small relative to the \( t_3 - t_3' \) and \( t_4 - t_4' \) factors, which are both nominally \( T_{pdelay2pdelay} \) (which is in ms, whereas the timestamps are in nanoseconds, hence \( T_{pdelay2pdelay} \times 10^6 \)).

\( t_{4pdelay} - t_{4pdelay}' \) divided by \( 10^6 \) on the lower line is small enough relative to \( T_{pdelay2pdelay} \) to ignore.
mNRR$_{\text{error}}$ Timestamp Error Example

\[
mNRR @ t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{31,250,313}{31,249,688} \rightarrow 0.002000\% = 20.00 \text{ ppm}
\]
mNRR_{error} Timestamp Error Example

mNRR @ t_{4}' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{31,250,315}{31,249,675} \rightarrow 0.0020512\% = 20.512 ppm
mNRR\text{error} Timestamp Error Example

- With no Timestamp Error...

\[
mNRR @ t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{31,250,313}{31,249,688} \rightarrow 0.002000\% = 20.00 \text{ ppm}
\]

- With \(t_{4\text{pderror}} = -8\) ns, \(t_{4\text{pderror}}' = +5\) ns, \(t_{3\text{pderror}} = +6\) ns, \(t_{3\text{pderror}}' = +3\) ns, \(mNRR_{errorCD} = 0.512\) ppm

\[
mNRR @ t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{31,250,315}{31,249,675} \rightarrow 0.0020512\% = 20.512 \text{ ppm}
\]

- From \(mNRR_{errorTS}\) equation...

\[
mNRR_{errorTS} \approx \frac{(t_{3\text{pderror}} - t_{3\text{pderror}}') - (t_{4\text{pderror}} - t_{4\text{pderror}}')}{T_{\text{pdelay2pdelay}}} = \frac{(6 - 3) - (-8 - 5)}{31.25} = \frac{16}{31.25} = 0.512 \text{ ppm}
\]
Equations – $mNRR_{\text{error}}$

Errors due to Clock Drift

\[
mNRR_{\text{error}} = mNRR_{\text{errorTS}} + mNRR_{\text{errorCD}}
\]

\[
mNRR_{\text{errorCD}} = mNRR_{\text{measuredCDerror}} - mNRR_{\text{nominal}}
\]

\[
= \left( \frac{t_3 - (t_3' + t_3'_{\text{CDerror}})}{t_4 - (t_3' + t_4'_{\text{CDerror}})} - 1 \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_3'} - 1 \right) \times 10^6
\]

\[
= \left( \frac{t_3 - t_3' - t_3'_{\text{CDerror}}}{t_4 - t_3' - t_4'_{\text{CDerror}}} - 1 \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_3'} - 1 \right) \times 10^6
\]

\[
= \left( \frac{t_3 - t_3'}{t_4 - t_4' - t_4'_{\text{CDerror}}} - 1 \right) + \left( \frac{-t_3'_{\text{CDerror}}}{t_4 - t_4' - t_4'_{\text{CDerror}}} \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_3'} - 1 \right) \times 10^6
\]

The ratio X in ppm is $(X - 1) \times 10^6$. 
Equations – mNRR

Errors due to Clock Drift

\( mNRR_{errorCD} = \left( \frac{t_3 - t_3'}{t_4 - t_4' - t_4'_{CDerror}} - 1 \right) \times 10^6 - \left( \frac{t_3 - t_3'}{t_4 - t_4'} \right) \times 10^6 \)

\[ = \frac{(t_4 - t_4')( -t_3'_{CDerror}) - (t_3 - t_3')( -t_4'_{CDerror})}{(t_4 - t_4')(t_4 - t_4') - t_4'_{CDerror}} \times 10^6 \]

\[ \approx \frac{T_{pdelay2pdelay} \times 10^6 \cdot (-t_3'_{CDerror}) - T_{pdelay2pdelay} \times 10^6 \cdot (-t_4'_{CDerror})}{T_{pdelay2pdelay} \times 10^6 \cdot (T_{pdelay2pdelay} \times 10^6 - t_4'_{CDerror})} \times 10^6 \]

\[ = \frac{t_4'_{CDerror} - t_3'_{CDerror}}{T_{pdelay2pdelay}} - \frac{t_4'_{CDerror}}{10^6} \]

\[ \approx \frac{t_4'_{CDerror} - t_3'_{CDerror}}{T_{pdelay2pdelay}} \]

The error magnitudes are small relative to the \( t_3 - t_3' \) and \( t_4 - t_4' \) factors, which are both nominally \( T_{pdelay2pdelay} \) (which is in ms, whereas the timestamps are in nanoseconds, hence \( T_{pdelay2pdelay} \times 10^6 \)).

\( t_4'_{CDerror} \) divided by \( 10^6 \) on the lower line is small enough relative to \( T_{pdelay2pdelay} \) to ignore.
Equations – $mNRR_{error}$

Errors due to Clock Drift

\[ mNRR_{errorCD} = \frac{t'_4CDerror - t'_3CDerror}{T_{pdelay2pdelay}} \]

\[ t'_4CDerror = t'_4measuredCDerror - t'_4nominal \]

\[
= \left( t_4 - T_{pdelay2pdelay} \times 10^6 \left( 1 + \frac{\text{clockOffset}_n(t_4)}{2} \times \frac{T_{pdelay2pdelay}}{10^6} \right) \right) - \left( t_4 - T_{pdelay2pdelay} \times 10^6 \left( 1 + \frac{\text{clockOffset}_n(t_4)}{10^6} \right) \right) 
\]

\[
= -T_{pdelay2pdelay} \times 10^6 - \frac{\text{clockDrift}_n \times T_{pdelay2pdelay}}{2 \times 10^3} 
\]

\[
= \frac{\text{clockDrift}_n \times T_{pdelay2pdelay}^2}{2 \times 10^3} 
\]

\[
t'_3CDerror = \frac{\text{clockDrift}_{n-1} \times T_{pdelay2pdelay}^2}{2 \times 10^3} 
\]
Equations – mNRR\text{error} Errors due to Clock Drift

\[ mNRR_{errorCD} = \frac{t'_{4CDerror} - t'_{3CDerror}}{T_{pdelay2pdelay}} \]

\[ t'_{4CDerror} = \frac{clockDrift_n \cdot T^2_{pdelay2pdelay}}{2 \times 10^3} \]

\[ t'_{3CDerror} = \frac{clockDrift_{n-1} \cdot T^2_{pdelay2pdelay}}{2 \times 10^3} \]

\[ mNRR_{errorCD} = \frac{clockDrift_{n-1} \cdot T^2_{pdelay2pdelay} - clockDrift_n \cdot T^2_{pdelay2pdelay}}{2 \times 10^3 \times T_{pdelay2pdelay}} \]

\[ = T_{pdelay2pdelay} \left( clockDrift_n - clockDrift_{n-1} \right) \frac{1}{2 \times 10^3} \]
mNRR<sub>error</sub> Clock Drift Example

mNRR @ t<sub>4</sub>' = \( \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{62,500,675}{62,499,375} \rightarrow 0.0020000\% = 20.000 \text{ ppm} \)
\[ mNRR \text{ @ } t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{62,500,634}{62,499,375} \rightarrow 0.0020150\% = 20.150 \text{ ppm} \]
\[
mNRR @ t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{62,500,634}{62,499,370} \rightarrow 0.0020225\% = 20.225 \text{ ppm}
\]
mNRR\_error Clock Drift Example

- With no Clock Drift...

\[
mNRR \text{ at } t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{62,500,675}{62,499,375} \rightarrow 0.0020000\% = 20.000 \text{ ppm}
\]

- With \( \text{clockDrift}_n \) = 0.3 ppm/s and \( \text{clockDrift}_{n-1} \) = -0.6 ppm/s, \( mNRR_{\text{errorCD}} = 0.225 \text{ ppm} \)

\[
mNRR \text{ at } t_4' = \frac{(t_3 - t_3')}{(t_4 - t_4')} = \frac{62,499,370}{62,500,634} \rightarrow -0.0020225\% = 20.225 \text{ ppm}
\]

- From \( mNRR_{\text{errorCD}} \) equation...

\[
mNRR_{\text{errorCD}} = \frac{T_{p\text{delay2p\text{delay}}}(\text{clockDrift}_n - \text{clockDrift}_{n-1})}{2 \times 10^3} = \frac{500(0.3 + 0.6)}{2 \times 10^3} = \frac{500(0.9)}{2 \times 10^3} = \frac{450}{2 \times 10^3} = 0.225 \text{ ppm}
\]
Note on Algorithmic Equivalence

• Both the above derivations utilise the following equivalence...

\[ c \left( \frac{a}{b+y} - 1 + \frac{x}{b+y} \right) - c \left( \frac{a}{b} - 1 \right) = \frac{bx - ay}{b(b+y)} \]

• The detailed steps are as follows...

\[ c \left( \frac{a}{b+y} - 1 + \frac{x}{b+y} \right) - c \left( \frac{a}{b} - 1 \right) \]
\[ = c \left( \frac{a}{b+y} + \frac{-b - y + x}{b+y} \right) - c \left( \frac{a + b}{b} \right) \]
\[ = \frac{a - b - y + x}{b+y} - c \frac{a - b}{b} \]
\[ = \frac{ab - b^2 - by + bx}{b(b+y)} - c \frac{a(b+y) - b(b+y)}{b(b+y)} \]
\[ = \frac{ab - b^2 - by + bx}{b(b+y)} - \frac{ab + ay - b^2 - by}{b(b+y)} \]
\[ = \frac{ab - b^2 - by + bx - ab - ay + b^2 + by}{b(b+y)} \]
\[ = \frac{bx - ay}{b(b+y)} \]
Equations – $RR_{error} – 1$

• RR is calculated via an accumulation of NRRs. At each node $RR_{error}$ is the sum of...
  • $RR_{error}$ at the upstream node
  • $mNRR_{error}$
  • Error due to Clock Drift between $mNRR$ calculation and transmission of Sync
    • Or, for the last node only, reception of Sync
  • Error due to Clock Drift between RR calculation at upstream node and transmission of Sync
Equations – $RR_{error} - 2$

$$RR_{error}(n) = RR_{error}(n - 1) + mNRR_{error} + RR_{errorCD,NRR2sync} + RR_{errorCD,RR2sync}$$

$$RR_{errorCD,NRR2sync} = \frac{T_{mNRR2sync}(clockDrift_n - clockDrift_{n-1})}{10^3}$$

$$T_{mNRR2sync} = \sim U(pdelayInterval, 0.9, pdelayInterval, 1.3) \times \sim U(0, 1)$$

$$RR_{errorCD,RR2sync} = \frac{residenceTime(clockDrift_{n-1} - clockDrift_{GM})}{10^3}$$

At the final hop there is no Residence Time, so $RR_{errorCD,RR2sync}$ is zero.
Equations – $M_{\text{error}}$ (Mean Link Delay) – 1

\[ M_{\text{error}} = M_{\text{errorTSdirect}} + M_{\text{errorNRR}} \]

\[ M_{\text{errorTSdirect}} = M_{\text{measuredTSerror}} - M_{\text{nominal}} \]

\[ = \frac{(t_4 + t_{4 \text{perror}}) - (t_1 + t_{1 \text{perror}})}{2} - \frac{(t_3 + t_{3 \text{perror}}) - (t_2 + t_{2 \text{perror}})}{2} \left(1 - \frac{m_{\text{NRR}}}{10^6}\right) \]

\[ = \frac{(t_{4 \text{perror}} - t_{1 \text{perror}}) - (t_{3 \text{perror}} - t_{2 \text{perror}})}{2} \left(1 - \frac{m_{\text{NRR}}}{10^6}\right) \]

\[ \approx \frac{(t_{4 \text{perror}} - t_{1 \text{perror}}) - (t_{3 \text{perror}} - t_{2 \text{perror}})}{2} \]
Equations – $\text{MLD}_{\text{error}}$ (Mean Link Delay) – 2

$$\text{MLD}_{\text{error}} = \text{MLD}_{\text{errorDirect}} + \text{MLD}_{\text{errorNRR}}$$

$$\text{MLD}_{\text{errorNRR}} = \text{MLD}_{\text{measuredNRRerror}} - \text{MLD}_{\text{nominal}}$$

\[
\begin{align*}
(t_4 - t_1) - (t_3 - t_2) \left(1 - \frac{m\text{NRR} + m\text{NRR}_{\text{error}}}{10^6}\right) &= \frac{(t_4 - t_1) - (t_3 - t_2) \left(1 - \frac{m\text{NRR}}{10^6}\right)}{2} - \frac{(t_4 - t_1) - (t_3 - t_2) \left(1 - \frac{m\text{NRR}}{10^6}\right)}{2} \\
(t_4 - (t_3 - p\text{DelayTurnaround} \times 10^6 - 2.\text{meanLinkDelay}) - \left(t_3 - (t_3 - p\text{DelayTurnaround} \times 10^6 \left(1 - \frac{m\text{NRR} + m\text{NRR}_{\text{error}}}{10^6}\right)\right)) &= \frac{(t_4 - (t_3 - p\text{DelayTurnaround} \times 10^6 - 2.\text{meanLinkDelay}) - \left(t_3 - (t_3 - p\text{DelayTurnaround} \times 10^6 \left(1 - \frac{m\text{NRR}}{10^6}\right)\right))}{2} \\
(t_4 - (t_3 - p\text{DelayTurnaround} \times 10^6 - 2.\text{meanLinkDelay}) - \left(t_3 - (t_3 - p\text{DelayTurnaround} \times 10^6 \left(1 - \frac{m\text{NRR}}{10^6}\right)\right)) &= -\frac{p\text{DelayTurnaround} \cdot m\text{NRR}_{\text{error}}}{2}
\end{align*}
\]
Equations – $RT_{\text{error}}$ (Residence Time) – 1

\[
RT_{\text{error}} = RT_{\text{errorTSdirect}} + RT_{\text{errorRR}} + RT_{\text{errorCDdirect}}
\]

\[
RT_{\text{errorTSdirect}} = t_{1\text{outerror}} - t_{2\text{sinerror}}
\]

\[
RT_{\text{errorRR}} = RT_{\text{measuredRRerror}} - RT_{\text{nominal}}
\]

\[
= (t_{1\text{outNominal}} - t_{2\text{sinMeasured}}) - (t_{1\text{outNominal}} - t_{2\text{sinNominal}})
\]

\[
= -\left( t_{1\text{outNominal}} - \text{residenceTime} \times 10^6 \left( 1 + \frac{freqOffset}{10^6} \right) \left( 1 + \frac{RR + RR_{\text{error}}}{10^6} \right) \right)
\]

\[
+ \left( t_{1\text{outNominal}} - \text{residenceTime} \times 10^6 \left( 1 + \frac{freqOffset}{10^6} \right) \left( 1 + \frac{RR}{10^6} \right) \right)
\]

\[
= -\left( -\text{residenceTime} \cdot RR_{\text{error}} \left( 1 + \frac{freqOffset}{10^6} \right) \right)
\]

\[
= \text{residenceTime} \cdot RR_{\text{error}} + \text{residenceTime} \cdot RR_{\text{error}} \cdot freqOffset_{10^6}
\]

\[
\approx \text{residenceTime} \times RR_{\text{error}}
\]
\[
R_{\text{error}} = R_{\text{errorTSdirect}} + R_{\text{errorRR}} + R_{\text{errorCDdirect}}
\]

\[
R_{\text{errorTSdirect}} = t_{1\text{out error}} - t_{2\text{sin error}}
\]

\[
R_{\text{errorCDdirect}} = R_{\text{measuredCDerror}} - R_{\text{nominal}}
\]

\[
R_{\text{errorRR}} = \text{residenceTime} \times RR_{\text{error}}
\]

\[
R_{\text{error}} = (t_{1\text{outNominal}} - t_{2\text{in Measured}}) \left(1 + \frac{RR}{10^6}\right) - (t_{1\text{out Nominal}} - t_{2\text{sin Nominal}}) \left(1 + \frac{RR}{10^6}\right)
\]

\[
= - \left(t_{1\text{outNominal}} - \text{residenceTime} \times 10^6 \left(1 + \frac{\text{clockOffset}_{n}(t_{1\text{out}})}{2 \times 10^6} + \frac{\text{clockDrift}_{n} \times \text{residenceTime}}{10^3}\right)\right) \left(1 + \frac{RR}{10^6}\right)
\]

\[
+ \left(t_{1\text{outNominal}} - \text{residenceTime} \times 10^6 \left(1 + \frac{\text{clockOffset}_{n}(t_{1\text{out}})}{2 \times 10^6} + \frac{\text{clockDrift}_{GM} \times \text{residenceTime}}{10^3}\right)\right) \left(1 + \frac{RR}{10^6}\right)
\]

\[
= - \left(-\text{residenceTime} \times 10^6 \frac{\text{clockDrift}_{n}}{2} \frac{10^3}{10^6}\right) \left(1 + \frac{RR}{10^6}\right) + \left(-\text{residenceTime} \times 10^6 \frac{\text{clockDrift}_{GM}}{2} \frac{10^3}{10^6}\right) \left(1 + \frac{RR}{10^6}\right)
\]

\[
= \text{residenceTime}^2 \frac{\text{clockDrift}_{n} - \text{clockDrift}_{GM}}{2 \times 10^3} + \text{residenceTime}^2 \frac{RR(\text{clockDrift}_{n} - \text{clockDrift}_{GM})}{2 \times 10^9}
\]

\[
\approx \text{residenceTime}^2 \frac{\text{clockDrift}_{n} - \text{clockDrift}_{GM}}{2 \times 10^3}
\]
Equations – $E_{\text{error}}$ (End Station) – 1

$$E_{\text{error}} = E_{\text{error}R} + E_{\text{error}CD\text{direct}}$$

$$E_{\text{error}R} = E_{\text{actual}R} - E_{\text{nominal}}$$

\[
E_{\text{error}} = T_{\text{sync}2\text{sync}} \times 10^6 \left( 1 + \frac{\text{freqOffset}}{10^6} \right) \left( 1 + \frac{RR + RR_{\text{error}}}{10^6} \right) - T_{\text{sync}2\text{sync}} \times 10^6 \left( 1 + \frac{\text{freqOffset}}{10^6} \right) \left( 1 + \frac{RR}{10^6} \right)
\]

\[
= T_{\text{sync}2\text{sync}} \times 10^6 \left( 1 + \frac{\text{freqOffset}}{10^6} \right) \left( 1 + \frac{RR}{10^6} \right) + T_{\text{sync}2\text{sync}} \times 10^6 \left( 1 + \frac{\text{freqOffset}}{10^6} \right) \left( \frac{RR_{\text{error}}}{10^6} \right) - T_{\text{sync}2\text{sync}} \times 10^6
\]

\[
= T_{\text{sync}2\text{sync}} \cdot RR_{\text{error}} + \frac{T_{\text{sync}2\text{sync}} \cdot RR_{\text{error}} \cdot \text{freqOffset}}{10^6}
\]

\[
\approx T_{\text{sync}2\text{sync}} \cdot RR_{\text{error}}
\]

$$T_{\text{SyncToSync}} = \sim \Gamma \left( \frac{270.5532}{\text{syncInterval}} \right)$$

The error associated with freqOffset is not modelled as it is orders of magnitude smaller than the main error.
Equations – $ES_{error}$ (End Station) – 2

$ES_{error} = ES_{errorRR} + ES_{errorCDdirect}$

$ES_{errorCDdirect} = ES_{actualCDerror} - ES_{nominal}$

$$ES_{errorRR} = TS_{sync2sync} \times 10^6 \left( 1 + \frac{freqOffset + \frac{clockDrift_n}{2} \times TS_{sync2sync}}{10^6} \right) \left( 1 + \frac{RR}{10^6} \right) - TS_{sync2sync} \times 10^6 \left( 1 + \frac{freqOffset + \frac{clockDrift_{GM}}{2} \times TS_{sync2sync}}{10^6} \right) \left( 1 + \frac{RR}{10^6} \right)$$

$$ES_{errorCDdirect} = TS_{sync2sync} \left( \frac{clockDrift_n - clockDrift_{GM}}{2} \times TS_{sync2sync} \right) \left( 1 + \frac{RR}{10^6} \right)$$

$$\approx TS_{sync2sync} \left( \frac{clockDrift_n - clockDrift_{GM}}{2} \times TS_{sync2sync} \right) + \frac{RR \times TS_{sync2sync} \left( clockDrift_n - clockDrift_{GM} \right)}{2 \times 10^9}$$

The error associated with $RR$ is not modelled as it is orders of magnitude smaller than the main error.
Equations – DTE

• At all hops other than the last...

\[ DTE(n) = DTE(n - 1) + MLD_{error} + RT_{error} \]

• At the last hop...

\[ DTE(n) = DTE(n - 1) + MLD_{error} + ES_{error} \]
Error Contribution Tracking & Graphical Representations
Error Contribution Tracking

• As well as calculating the primary errors required to calculate DTE, the model also tracks the components of each error and how they accumulate.

• This makes is relatively simple to answer questions such as “What is the probability distribution of DTE due to the Timestamp Error related component of Neighbor Rate Ratio?”

• It also enables a graphical representation of how DTE breaks down into it’s contributing components.

• This section describes the equations used to track the components and the graphical representations.
Error Contribution Tracking – Implementation

• For each component or a primary error, the contribution of the component at each hop is calculated. The vector (across all runs) for this is given the suffix `_X_`.

• For each primary error and component, the vector representing the running total of all errors of this type (across all runs) up to and including the current hop, is calculated and given the suffix `_SUM`.

• From the SUM vectors, the following statistical values are calculated and stored in a single vector per primary error or component (one value for each hop representing the statistic across all runs)...  
  • Maximum Absolute Value (`MAXabs`)  
  • Mean (`MEAN`)  
  • Sigma (`SIGMA`), assuming a gaussian distribution (which is not always valid)

• The `_X_` and `_SUM_` vectors are not preserved, other than at the last hop.

• There are 3 exceptions...  
  • `mNRR` error does not accumulate, so the SUM values are not calculated, although MAXabs, MEAN and SIGMA values are (based on the `_X_` error vectors at each hop).
  • `RR` error is an accumulation of `mNRR`, so `_X_` values do not exist. (They do, however, exist for some Clock Drift related error components that don’t accumulate.)
  • At the last node, Residence Time Error (`RT` error) is effectively replaced by End Station Error (`ES` error) which is only calculated at the final hop. Combined `RTES` error vectors track SUM, MAXabs, MEAN and SIGMA (representing the sum of RT errors up to the last-but-one hop, then the sum of RT errors plus the ES errors at the last hop). The SUM vectors for RT at the last-but-one hop are preserved, and the vectors for ES errors at the last hop are also available as well as the combined RTES error vectors (to enable statistical analysis).
## Error & Error Components – Naming

<table>
<thead>
<tr>
<th>Item</th>
<th>Examples using Residence Time</th>
<th>Residence Time errors due to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>$RT$</td>
<td>Residence Time (no errors); other elements are mNRR, RR, MLD &amp; ES.</td>
</tr>
<tr>
<td>Error in that Element</td>
<td>$RT_{error}$</td>
<td>All underlying components</td>
</tr>
<tr>
<td>Component of that error due to another Element</td>
<td>$RT_{errorTS}$, $RT_{errorCD}$, $RT_{errorTSdirect}$, $RT_{errorCDdirect}$, $RT_{errorRR}$</td>
<td>All Timestamp components, All Clock Drift components, Direct Timestamp components, Direct Clock Drift components, Rate Ratio Error</td>
</tr>
<tr>
<td>Next level down...</td>
<td>$RT_{errorRR, TS}$, $RT_{errorRR, CDdirect}$, $RT_{errorRR, NRR}$</td>
<td>All Timestamp components of Rate Ratio Error, Direct Clock Drift component of Rate Ratio Error, NRR error component of Rate Ratio Error</td>
</tr>
<tr>
<td>Next level down...</td>
<td>$RT_{errorRR, NRR, TS}$, $RT_{errorRR, NRR, CD}$</td>
<td>Timestamp component of NRR via Rate Ratio Error, Clock Drift component of NRR via Rate Ratio Error</td>
</tr>
</tbody>
</table>
**Equations - mNRR_{error}**

<table>
<thead>
<tr>
<th>Primary Errors</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mNRR_{errorTS,X}</td>
<td>[ \frac{(t_{3perror} - t_{3perror}') - (t_{4perror} - t_{4perror}')}{T_{pdelay2pdelay}} ]</td>
<td>ppm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mNRR_{errorCD,X}</td>
<td>[ \frac{T_{pdelay2pdelay}(\text{clockDrift}<em>n - \text{clockDrift}</em>{n-1})}{2 \times 10^3} ]</td>
<td>ppm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mNRR_{error,X}</td>
<td>[ mNRR_{errorTS,X} + mNRR_{errorCD,X} ]</td>
<td>ppm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Error Components**

mNRR_{error} breaks down into a Timestamp Error and an error due to Clock Drift, so there are no additional error components or calculations.
Equations - $RR_{error}$

### Primary Errors

- $RR_{errorCD,NRR2Sync, X} = \frac{T_{mNRR2Sync} (\text{clockDrift}_n - \text{clockDrift}_{n-1})}{10^3}$, $\text{ppm}$
- $RR_{errorCD,RR2Sync, X} = \frac{\text{residenceTime}(\text{clockDrift}_{n-1} - \text{clockDrift}_{GM})}{10^3}$, $\text{ppm}$
- $RR_{error,SUM} (n) = RR_{error,SUM} (n-1) + mNRR_{error, X} + RR_{errorCD,NRRtoSync,X} + RR_{errorCD,RRtoSync,X}$, $\text{ppm}$

### Error Components

- $RR_{errorNRR,CD,SUM} (n) = RR_{errorNRR,CD,SUM} (n-1) + mNRR_{errorCD,X} (n)$, $\text{ppm}$
- $RR_{errorCD,NRR2sync,SUM} (n) = RR_{errorCD,NRR2sync,SUM} (n-1) + RR_{errorCD,NRR2Sync,X} (n)$, $\text{ppm}$
- $RR_{errorCD,RR2sync,SUM} (n) = RR_{errorCD,RR2sync,SUM} (n-1) + RR_{errorCD,RR2Sync,X} (n)$, $\text{ppm}$
- $RR_{errorNRR,SUM} (n) = RR_{errorNRR,SUM} (n-1) + mNRR_{error,X} (n)$, $\text{ppm}$
- $RR_{errorTS,SUM} (n) = RR_{errorTS,SUM} (n-1) + mNRR_{errorTS,X} (n)$, $\text{ppm}$
- $RR_{errorCD,SUM} (n) = RR_{errorNRR,CD,SUM} (n) + RR_{errorCD,NRR2sync,SUM} (n) + RR_{errorCD,RR2sync,SUM} (n)$, $\text{ppm}$

At the last hop, there is no Residence Time, so $RR_{errorCD,RR2Sync,SUM} = 0$. 
### Equations – MLD\textsubscript{error} – Per Hop

<table>
<thead>
<tr>
<th>Error Components</th>
<th>Equation</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Errors</strong></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( MLD\textsubscript{errorTS_direct,_X} = \frac{(t_4p_error - t_1p_error) - (t_3p_error - t_2p_error)}{2} )</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>( MLD\textsubscript{errorNRR,_X} = \frac{-p_Delay_Turnaround \cdot m_NRR_error}{2} )</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>( MLD\textsubscript{error,_X} = MLD\textsubscript{errorTS_direct} + MLD\textsubscript{errorNRR} )</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td><strong>Error Components</strong></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>( MLD\textsubscript{errorNRR_TS,_X} = \frac{-p_Delay_Turnaround \cdot m_NRR_error_TS_X}{2} )</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>( MLD\textsubscript{errorCD,_X} = \frac{-p_Delay_Turnaround \cdot m_NRR_error_CD_X}{2} )</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>( MLD\textsubscript{errorTS_X} = MLD\textsubscript{errorTS_direct} + MLD\textsubscript{errorNRR_TS_X} )</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>
Equations – $\text{MLD}_{\text{error}}$ – Accumulation

<table>
<thead>
<tr>
<th>Primary Errors</th>
<th>Equation</th>
<th>ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{direct sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{direct sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{direct x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{nrr sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{nrr sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{nrr x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Components</th>
<th>Equation</th>
<th>ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{nrr ts sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{nrr ts sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{nrr ts x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{cd sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{cd sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{cd x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
<tr>
<td>$\text{MLD}<em>{\text{error}}</em>{\text{ts sum}}(n) = \text{MLD}<em>{\text{error}}</em>{\text{ts sum}}(n - 1) + \text{MLD}<em>{\text{error}}</em>{\text{ts x}}$</td>
<td>\text{ns}</td>
<td></td>
</tr>
</tbody>
</table>
Equations – $RT_{error}$ – Per Hop (Except Last)

<table>
<thead>
<tr>
<th>Primary Errors</th>
<th>Equation</th>
<th>ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RT_{errorTSdirect_x}$</td>
<td>$t_{1souterror} - t_{2sinerror}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorCDdirect_x}$</td>
<td>$\frac{\text{residenceTime}^2 (\text{clockDrift}_n - \text{clockDrift}_GM)}{2 \times 10^3}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_x}$</td>
<td>$\text{residenceTime} \times R_{error_SUM}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{error_x}$</td>
<td>$RT_{errorTSdirect_x} + RT_{errorRR_x} + RT_{errorCDdirect_x}$</td>
<td>ns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Components</th>
<th>Equation</th>
<th>ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RT_{errorRR_NRR_CD_X}$</td>
<td>$\text{residenceTime} \times R_{error_NRR_CD_SUM}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_CD_NRR2sync_X}$</td>
<td>$\text{residenceTime} \times R_{error_CD_NRR2sync_SUM}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_CD_RR2sync_X}$</td>
<td>$\text{residenceTime} \times R_{error_CD_RR2sync_SUM}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_TS_X}$</td>
<td>$\text{residenceTime} \times R_{error_TS_SUM}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_NRR_X}$</td>
<td>$RT_{errorRR_NRR_CD_X} + RT_{errorRR_TS_X}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorRR_CD_X}$</td>
<td>$RT_{errorRR_NRR_CD_X} + RT_{errorRR_CD_NRR2sync_X} + RT_{errorRR_CD_RR2sync_X}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorCD_X}$</td>
<td>$RT_{errorCD_direct_X} + RT_{errorRR_CD_X}$</td>
<td>ns</td>
</tr>
<tr>
<td>$RT_{errorTS_X}$</td>
<td>$RT_{errorTS_direct_X} + RT_{errorRR_TS_X}$</td>
<td>ns</td>
</tr>
</tbody>
</table>
Equations – $\text{RT}_{\text{error}}$ – Accumulation

<table>
<thead>
<tr>
<th>Primary Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RT_{\text{error_Ts_direct_SUM}}(n) = RT_{\text{error_Ts_direct_SUM}}(n - 1) + RT_{\text{error_Ts_direct_X}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_Cd_direct_SUM}}(n) = RT_{\text{error_Cd_direct_SUM}}(n - 1) + RT_{\text{error_Cd_direct_X}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_sum}}(n) = RT_{\text{error_rr_sum}}(n - 1) + RT_{\text{error_rr_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_SUM}}(n) = RT_{\text{error_SUM}}(n - 1) + RT_{\text{error_x}}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RT_{\text{error_rr_nrr_cd_sum}}(n) = RT_{\text{error_rr_nrr_cd_sum}}(n - 1) + RT_{\text{error_rr_nrr_cd_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_cd_nrr_2_sync_sum}}(n) = RT_{\text{error_rr_cd_nrr_2_sync_sum}}(n - 1) + RT_{\text{error_rr_cd_nrr_2_sync_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_cd_rr_2_sync_sum}}(n) = RT_{\text{error_rr_cd_rr_2_sync_sum}}(n - 1) + RT_{\text{error_rr_cd_rr_2_sync_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_ts_sum}}(n) = RT_{\text{error_rr_ts_sum}}(n - 1) + RT_{\text{error_rr_ts_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_nrr_sum}}(n) = RT_{\text{error_rr_nrr_sum}}(n - 1) + RT_{\text{error_rr_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_rr_cd_sum}}(n) = RT_{\text{error_rr_cd_sum}}(n - 1) + RT_{\text{error_rr_cd_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_cd_sum}}(n) = RT_{\text{error_cd_sum}}(n - 1) + RT_{\text{error_cd_x}}$</td>
</tr>
<tr>
<td>$RT_{\text{error_ts_sum}}(n) = RT_{\text{error_ts_sum}}(n - 1) + RT_{\text{error_ts_x}}$</td>
</tr>
</tbody>
</table>
## Equations – ES_{error} – Last Hop Only

<table>
<thead>
<tr>
<th>Primary Errors</th>
<th>Equation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{error} _ \text{RR}} _ X = T_{\text{sync} 2 \text{sync}} \cdot R_{\text{error} _ \text{SUM}}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{CD} _ \text{direct} _ X} = T_{\text{sync} 2 \text{sync}} \cdot \left( \frac{2 \times 10^3 \left( \text{clockDrift}<em>n - \text{clockDrift}</em>{\text{GM}} \right)}{2 \times 10^3} \right)$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ X} = E_{\text{error} _ \text{RR}} + E_{\text{error} _ \text{CD} _ \text{direct}}$</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Components</th>
<th>Equation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{error} _ \text{RR} _ \text{CD} _ X} = T_{\text{sync} 2 \text{sync}} \cdot R_{\text{error} _ \text{CD} _ \text{SUM}}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{CD} _ \text{NRR} _ \text{2sync} _ X} = T_{\text{sync} 2 \text{sync}} \cdot R_{\text{error} _ \text{CD} _ \text{NRR} _ \text{2sync} _ \text{SUM}}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{CD} _ \text{RR} _ 2sync _ X} = T_{\text{sync} 2 \text{sync}} \cdot R_{\text{error} _ \text{CD} _ \text{RR} _ 2sync _ \text{SUM}}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{CD} _ X} = E_{\text{error} _ \text{RR} _ \text{NRR} _ \text{CD} _ X} + E_{\text{error} _ \text{RR} _ \text{CD} _ \text{NRR} _ \text{2sync} _ X} + E_{\text{error} _ \text{RR} _ \text{CD} _ \text{RR} _ 2sync _ X}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{TT} _ X} = T_{\text{sync} 2 \text{sync}} \cdot R_{\text{error} _ \text{TS} _ \text{SUM}}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{NRR} _ X} = E_{\text{error} _ \text{TT} _ X} + E_{\text{error} _ \text{RR} _ \text{CD} _ X}$</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>$E_{\text{error} _ \text{CD} _ X} = E_{\text{error} _ \text{RR} _ \text{CD} _ X} + E_{\text{error} _ \text{CD} _ \text{direct} _ X}$</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>
## Equations – DTE – Per Hop & Accumulation

| Primary Errors | At all hops other than last: \(DTE_X = MLD_{\text{error}_X} + RT_{\text{error}_X}\) | ns |
|               | At last hop: \(DTE_X = MLD_{\text{error}_X} + ES_{\text{error}_X}\) | ns |
|               | At all hops: \(DTE_{\text{SUM}}(n) = DTE_{\text{SUM}}(n - 1) + DTE_X\) | ns |

| Error Components | At all hops other than last: \(DTE_{CD,X} = MLD_{\text{error}_{CD,X}} + RT_{\text{error}_{CD,X}}\) | ns |
|                  | \(DTE_{TS,X} = MLD_{\text{error}_{TS,X}} + RT_{\text{error}_{TS,X}}\) | ns |
|                  | At last hop: \(DTE_{CD,X} = MLD_{\text{error}_{CD,X}} + ES_{\text{error}_{CD,X}}\) | ns |
|                  | \(DTE_{TS,X} = MLD_{\text{error}_{TS,X}} + ES_{\text{error}_{TS,X}}\) | ns |
|                  | At all hops: \(DTE_{CD\_SUM}(n) = DTE_{CD\_SUM}(n - 1) + DTE_{CD,X}\) | ns |
|                  | \(DTE_{TS\_SUM}(n) = DTE_{TS\_SUM}(n - 1) + DTE_{TS,X}\) | ns |
Graphical Representation of Error Accumulation

- Mean Link Delay Error
- mNRR Error
- Timestamp Error
- Clock Drift Error - NRR
- Residence Time Error
- Rate Ratio Error
- mNRR Error
- Clock Drift Error - RR
- Dynamic Time Error
- Clock Drift Error - NRR
- End Station Error
- Clock Drift Error - RR
Graphical Representation of Error Accumulation

The $7\sigma$ dTE value is split repeatedly according to the ratio of $7\sigma$ values of underlying errors. $7\sigma$ probabilities do not combine via addition so, at each level, the sum of the underlying $7\sigma$ values is greater than the value that is being split. Larger errors will often swamp smaller errors, so small errors are, in general, over-represented by this approach. It does, however, provide a useful visualisation of how underlying errors combine to make up the $7\sigma$ dTE value.
Graphical Representation of Error Accumulation

### Input Errors
- **Drift Type (Linear Temp Ramp)**: 2
- **GM Clock Drift Max**: +1.35 ppm/s
- **GM Clock Drift Min**: -1.35 ppm/s
- **Fraction of GM nodes w/ Drift**: 80%
- **Non-GM Clock Drift Max**: +1.35 ppm/s
- **Non-GM Clock Drift Min**: -1.35 ppm/s
- **Fraction of non-GM Nodes w/ Drift**: 80%

- **Temp Max**: +85. °C
- **Temp Min**: -40. °C
- **Temp Ramp Rate**: ±1 °C/s
- **Temp Ramp Period**: 125 s
- **Temp Hold Period**: 30 s
- **GM Scaling Factor**: 100%
- **Non-GM Scaling Factor**: 100%
- **Timestamp Granularity TX**: ±4 ns
- **Timestamp Granularity RX**: ±4 ms
- **Dynamic Time Stamp Error TX**: ±4 ns
- **Dynamic Time Stamp Error RX**: ±4 ns

### Input Parameters
- **pDelay Interval**: 250 ms
- **Sync Interval**: 125 ms
- **pDelay Turnaround Time**: 10 ms
- **residenceTime**: 10 ms

### Input Correction Factors
- **Mean Link Delay Averaging**: 0%
- **NRR Drift Rate Correction**: 0%
- **RR Drift Rate Error Correction**: 0%
- **pDelayResp → Sync Type (Uniform)**: 1
- **pDelayResp → Sync Max**: 100%
- **pDelayResp → Sync Min**: 0%
- **pDelayResp → Sync Target**: 10 ms
- **mNRR Smoothing N**: 1
- **mNRR Smoothing M**: 1

### Configuration
- **Hops**: 100
- **Runs**: 1,000,000
# pDelayInterval Sensitivity Analysis

## Input Errors

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drift Type (Linear Temp Ramp)</td>
<td>2</td>
</tr>
<tr>
<td>GM Clock Drift Max</td>
<td>+1.35 ppm/s</td>
</tr>
<tr>
<td>GM Clock Drift Min</td>
<td>-1.35 ppm/s</td>
</tr>
<tr>
<td>Fraction of GM nodes w/ Drift</td>
<td>80%</td>
</tr>
<tr>
<td>non-GM Clock Drift Max</td>
<td>+1.35 ppm/s</td>
</tr>
<tr>
<td>non-GM Clock Drift Min</td>
<td>-1.35 ppm/s</td>
</tr>
<tr>
<td>Fraction of non-GM Nodes w/ Drift</td>
<td>80%</td>
</tr>
<tr>
<td>Temp Max</td>
<td>+85. °C</td>
</tr>
<tr>
<td>Temp Min</td>
<td>-40. °C</td>
</tr>
<tr>
<td>Temp Ramp Rate</td>
<td>±1 °C/s</td>
</tr>
<tr>
<td>Temp Ramp Period</td>
<td>125 s</td>
</tr>
<tr>
<td>Temp Hold Period</td>
<td>30 s</td>
</tr>
<tr>
<td>GM Scaling Factor</td>
<td>100%</td>
</tr>
<tr>
<td>non-GM Scaling Factor</td>
<td>100%</td>
</tr>
<tr>
<td>Timestamp Granularity TX</td>
<td>±4 ns</td>
</tr>
<tr>
<td>Timestamp Granularity RX</td>
<td>±4 ns</td>
</tr>
<tr>
<td>Dynamic Time Stamp Error TX</td>
<td>±4 ns</td>
</tr>
<tr>
<td>Dynamic Time Stamp Error RX</td>
<td>±4 ns</td>
</tr>
</tbody>
</table>

## Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pDelay Interval</td>
<td>VAR ms</td>
</tr>
<tr>
<td>Sync Interval</td>
<td>125 ms</td>
</tr>
<tr>
<td>pDelay Turnaround Time</td>
<td>10 ms</td>
</tr>
<tr>
<td>residenceTime</td>
<td>10 ms</td>
</tr>
</tbody>
</table>

## Input Correction Factors

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Link Delay Averaging</td>
<td>0%</td>
</tr>
<tr>
<td>NRR Drift Rate Correction</td>
<td>0%</td>
</tr>
<tr>
<td>RR Drift Rate Error Correction</td>
<td>0%</td>
</tr>
<tr>
<td>pDelayResp → Sync Type (Uniform)</td>
<td>1</td>
</tr>
<tr>
<td>pDelayResp → Sync Max</td>
<td>100%</td>
</tr>
<tr>
<td>pDelayResp → Sync Min</td>
<td>0%</td>
</tr>
<tr>
<td>pDelayResp → Sync Target</td>
<td>10 ms</td>
</tr>
<tr>
<td>mNRR Smoothing N</td>
<td>1</td>
</tr>
<tr>
<td>mNRR Smoothing M</td>
<td>1</td>
</tr>
</tbody>
</table>

## Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hops</td>
<td>100</td>
</tr>
<tr>
<td>Runs</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>
Algorithmic Improvements & Corrections
Aligning pDelayResp with Sync

• Clock drift between measuring NRR (mNRR) and using mNRR during Sync messaging (to calculate RR and then multiply meanLinkDelay + residenceTime by RR) introduces an error:

\[ R_{error_{CD,NRR2Sync}} = \frac{T_{mNRR2Sync}(\text{clockDrift}_n - \text{clockDrift}_{n-1})}{10^3} \]

\[ T_{mNRR2Sync} = \sim U(pdelayInterval, 0.9, pdelayInterval, 1.3) \times \sim U(0, 1) \]

• By aligning pDelayResp with Sync messaging \( RR_{error} \) can be reduced.
Aligning pDelayResp with Sync - Parameters

- The Monte Carlo Model offers three approaches to modelling alignment of pDelayResp with Sync, controlled via input parameter pDelayRespSyncAlignMode

<table>
<thead>
<tr>
<th>Correction Parameter</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>pDelayRespSyncAlignMode</td>
<td>1</td>
<td>-</td>
<td>1: Uniform distribution between a minimum and maximum fraction of ( T_{pdelay2pdelay} ) without any alignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2: Gamma distribution with target ( pDelayRespSyncAlignTarget )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3: Gaussian distribution with mean ( pDelayRespSyncAlignTarget ) and standards deviation ( pDelayRespSyncAlignSD )</td>
</tr>
<tr>
<td>pDelayRespSyncAlignMax</td>
<td>1</td>
<td>-</td>
<td>Maximum fraction for uniform distribution</td>
</tr>
<tr>
<td>pDelayRespSyncAlignMin</td>
<td>0</td>
<td>-</td>
<td>Minimum fraction for uniform distribution</td>
</tr>
<tr>
<td>pDelayRespSyncAlignTarget</td>
<td>10</td>
<td>ms</td>
<td>Target value for Gamma and Gaussian distribution</td>
</tr>
<tr>
<td>pDelayRespSyncAlignSD</td>
<td>3</td>
<td>ms</td>
<td>Standard deviation for Gaussian Distribution</td>
</tr>
</tbody>
</table>
Aligning pDelayResp with Sync - Parameters

• In mode 1:

\[ T_{mNRR2sync} = \sim U(pdelayInterval \cdot 0.9, pdelayInterval \cdot 1.3) \times \sim U(pDelayRespSyncAlignMin, pDelayRespSyncAlignMax) \]

• In mode 2:

\[ T_{mNRR2sync} = \sim \Gamma\left(270.5532, \frac{270.5532}{pDelayRespSyncAlignTarget}\right) \]

• In mode 3:

\[ T_{mNRR2sync} = \sim N(pDelayRespSyncAlignTarget, pDelayRespSyncAlignSD) \]
mNNRsmoothingN

- The Monte Carlo approach models using timestamp values from older pDelayResp messages via the \textit{mNNRsmoothingN} parameter adjusting $T_{\text{pDelay2pDelay}}$.

<table>
<thead>
<tr>
<th>Correction Parameter</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mNNRsmoothingN</td>
<td>1</td>
<td>-</td>
<td>Must be a whole number, minimum value 1.</td>
</tr>
</tbody>
</table>

$$T_{\text{pDelay2pDelay}} = \sum_{x=1}^{mNNRsmoothingN} \sim U(p\text{delayInterval}.0.9,p\text{delayInterval}.1.3)$$
MLD, NRR and RR Error Correction

• The model includes error correction factors for MLD, and Clock Drift related errors in NRR and RR
  • As the Monte Carlo approach is not a time series simulation, it does not model any of these algorithmic corrections in detail. It instead assumes a percentage effective factor, i.e. how much of the relevant error would be removed.
  • It is assumed that MLD error correction is accomplished via averaging.
  • It is assumed that NRR and RR error correction is accomplished via measuring Clock Drift in the past and – as Clock Drift is relatively consistent over time – compensating for Clock Drift in the future.
## MLD Error Correction

<table>
<thead>
<tr>
<th>Correction Parameter</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mLinkDelayErrCor</td>
<td>0</td>
<td>-</td>
<td>Value between zero (no error correction) and 1 (error is eliminated)</td>
</tr>
</tbody>
</table>

\[
MLD_{\text{error}TS_{\text{direct}}, X} = \frac{(t_{4perror} - t_{1perror}) - (t_{3perror} - t_{2perror})}{2} (1 - mLinkDelayErrCor)
\]

\[
MLD_{\text{error}NRR, X} = -\frac{p \text{ DelayTurnaround} \cdot mNRR_{\text{error}}}{2} (1 - mLinkDelayErrCor)
\]

\[
MLD_{\text{error}NRR,TS, X} = -\frac{p \text{ DelayTurnaround} \cdot mNRR_{\text{error,TS}, X}}{2} (1 - mLinkDelayErrCor)
\]

\[
MLD_{\text{error}CD, X} = -\frac{p \text{ DelayTurnaround} \cdot mNRR_{\text{error,CD}, X}}{2} (1 - mLinkDelayErrCor)
\]
NRR and RR Error Correction

<table>
<thead>
<tr>
<th>Correction Parameter</th>
<th>Default</th>
<th>Unit</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRR\text{driftRateErrorCor}</td>
<td>0</td>
<td>-</td>
<td>Value between zero (no error correction) and 1 (error is eliminated)</td>
</tr>
<tr>
<td>RR\text{driftRateErrorCor}</td>
<td>0</td>
<td>-</td>
<td>Value between zero (no error correction) and 1 (error is eliminated)</td>
</tr>
</tbody>
</table>

\[ m_{\text{NRR errorCD}} = \frac{T_{\text{delay2delay}} \cdot (\text{clockDrift}_n - \text{clockDrift}_{n-1})}{2 \times 10^3} \cdot (1 - \text{NRR drift Rate Error Cor}) \]

\[ R_{\text{errorCD,NRR2Sync}} = \frac{T_{\text{mNRR2Sync}} \cdot (\text{clockDrift}_n - \text{clockDrift}_{n-1})}{10^3} \cdot (1 - \text{NRR drift Rate Error Cor}) \]

\[ R_{\text{errorCD,R2Sync}} = \frac{\text{residenceTime} \cdot (\text{clockDrift}_n - \text{clockDrift}_{GM})}{10^3} \cdot (1 - \text{RR drift Rate Error Cor}) \]

\[ R_{\text{errorCD,direct}} = \frac{\text{residenceTime}^2 \cdot (\text{clockDrift}_n - \text{clockDrift}_{GM})}{2 \times 10^3} \cdot (1 - \text{RR drift Rate Error Cor}) \]

\[ E_{\text{errorCD,direct}} = \frac{T_{\text{sync2sync}} \cdot (\text{clockDrift}_n - \text{clockDrift}_{GM})}{2 \times 10^3} \cdot (1 - \text{RR drift Rate Error Cor}) \]
Thank you!
Equations – mNRR – Independence of TS & CD errors

\[ mNRRerror = mNRR_{measuredCDSError} - mNRR_{nominal} \]

\[ = \left( \frac{(t_3 + t_{3\text{perror}}) - (t'_3 + t'_{3\text{perror}} + t_{3\text{CDerror}'})}{(t_4 + t_{4\text{perror}}) - (t'_4 + t'_{4\text{perror}} + t_{4\text{CDerror}'})} \right) - 1 \times 10^6 - \left( \frac{t_3 - t'_3}{t_4 - t'_4} - 1 \right) \times 10^6 \]

\[ = \left( \frac{t_3 - t'_3 + t_{3\text{perror}} - t_{3\text{CDerror}'}}{t_4 - t'_4 + t_{4\text{perror}} - t_{4\text{CDerror}'}} - 1 \right) \times 10^6 - \left( \frac{t_3 - t'_3}{t_4 - t'_4} - 1 \right) \times 10^6 \]

\[ = \frac{(t_3 - t'_3)}{t_4 - t'_4 + (t_{3\text{perror}} - t_{4\text{perror}} - t_{4\text{CDerror}'})} \times 10^6 \]

\[ \approx T_{\text{pdelay}2\text{pdelay}} \times 10^6 (t_{3\text{perror}} - t_{4\text{perror}} - t_{4\text{CDerror}'}) - T_{\text{pdelay}2\text{pdelay}} \times 10^6 (t_{4\text{perror}} - t_{4\text{perror}} - t_{4\text{CDerror}'}) \]

\[ \times 10^6 \]

The error magnitudes are small relative to the \( t_3 - t'_3 \) and \( t_4 - t'_4 \) factors, which are both nominally \( T_{\text{pdelay}2\text{pdelay}} \) (which is in ms, whereas the timestamps are in nanoseconds, hence \( T_{\text{pdelay}2\text{pdelay}} \times 10^6 \)).
Equations – mNRR – Independence of TS & CD errors

\[
mNRR_{error} \approx \frac{T_{pdelay2pdelay} \times 10^6 (t_{3\text{perror}} - t_{3\text{perror}}' - t_{3\text{CDerror}}') - T_{pdelay2pdelay} \times 10^6 (t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}')}{T_{pdelay2pdelay} \times 10^6 (t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}')} \times 10^6
\]

\[
= \frac{(t_{3\text{perror}} - t_{3\text{perror}}' - t_{3\text{CDerror}}') - (t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}')}{T_{pdelay2pdelay} \times 10^6 + t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}'} \times 10^6
\]

\[
= \frac{(t_{3\text{perror}} - t_{3\text{perror}}' - t_{3\text{CDerror}}') - (t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}')}{T_{pdelay2pdelay}} + t_{4\text{CDerror}}' - t_{3\text{CDerror}}'
\]

\[
= mNRR_{errorTS} + mNRR_{errorCD}
\]

\[t_{4\text{perror}} - t_{4\text{perror}}' - t_{4\text{CDerror}}' \text{ divided } 10^6 \text{ by on the lower line is small enough relative to } T_{pdelay2pdelay} \text{ to ignore.}\]
Colours for Charts

- **Lines**: 7F7F7F, 4472C4, C00000, D09E00, 70AD47, 7030A0
- **Areas**: BFBFBF, B4C7E7, FF8181, B8C9AD, C5E0B4, C198E0
- **Backgrounds**: 7F7F7F, DAE3F3, FFBDBD, FFC3C3, FBE5D6, E2F0D9