60802 Dynamic Time Sync Error – Error Model & Monte Carlo Method Analysis

David McCall & Kevin Stanton (Intel)

November 2021 IEEE 802 Plenary – 802.1 TSN – IEEE/IEC 60802
Abstract


• Simulated protocol and system parameters have thus far either been judged impractical or have failed to meet the time-accuracy requirement.

• An analysis of how errors accumulate suggested that a Monte Carlo method analysis could support fast iteration of potential scenarios and deliver insights into cause and effect.
  • See 60802-McCall-et-al-Time-Sync-Error-Model-0921-v03.pdf

• In this contribution we:
  • Describe a Monte Carlo method analysis programmed in Rstudio
  • Compare the analysis’ results to results from previous time series simulations
  • Present a detailed analysis of sensitivities and trade offs
  • Recommend approaches to achieve the stated goals and propose next steps
Content

• Background & Recap
• Which Errors to Model & How They Add Up
• Monte Carlo Method Analysis Overview (RStudio)
• Comparison with Time Series Simulations
  • 7-Sigma Limit
• Error Analysis
  • Graphical Representation
  • Sensitivities & Trade Offs
• Recommendations & Next Steps
Background & Recap
In addition to the abstract...

• The Monte Carlo analysis is intended as an addition to the toolbox, not an alternative to Time Series simulation.

• If successful it should provide...
  • The ability to iterate much faster
  • Greater insight into the source of errors and how they accumulate
  • Greater confidence that when selecting parameters to achieve a desired goal
  • Input into future Time Series simulations
Which Errors to Model & How They Add Up
Time Sync – Errors to Model

Dynamic Time Error

- Link Delay Error
- Residence Time Error

- Time Stamp & Dynamic Time Stamp Error
- Neighbor Rate Ratio Error

Rate Ratio Error

Clock Drift
Time Sync – How Errors Add Up

All errors in this analysis are caused by either **Clock Drift** or **Timestamp Errors**
Monte Carlo Method Analysis

Overview of the modelling approach and analysis tool built in RStudio
Monte Carlo Method Analysis

• Inputs
• Errors, Formulae & Observations
• R & RStudio
  • Script code availability
• Demo
Input Errors

<table>
<thead>
<tr>
<th>Error</th>
<th>Distribution</th>
<th>Default Min</th>
<th>Default Max</th>
<th>Equation</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{clockDrift}<em>{GM} \ (\text{ClockDrift}</em>{GM_{\text{min}}} \ &amp; \ \text{ClockDrift}<em>{GM</em>{\text{max}}})</td>
<td>X% \text{ Stable (zero)} \ (100-X)% \text{ Uniform}</td>
<td>-0.6</td>
<td>+0.6</td>
<td>\text{clockDrift}<em>{GM} \sim U(\text{clockDrift}</em>{GM_{\text{min}}}, \text{clockDrift}<em>{GM</em>{\text{max}}})</td>
<td>\text{ppm/s}</td>
</tr>
<tr>
<td>\text{clockDrift}</td>
<td>X% \text{ Stable (zero)} \ (100-X)% \text{ Uniform}</td>
<td>-0.6</td>
<td>+0.6</td>
<td>\text{clockDrift} \sim U(-\text{clockDrift}, +\text{clockDrift})</td>
<td>\text{ppm/s}</td>
</tr>
<tr>
<td>\text{TSGE}_{TX}</td>
<td>\text{Uniform}</td>
<td>-4</td>
<td>+4</td>
<td>\text{TSGE}_{TX} \sim U(-\text{TSGETX}, +\text{TSGETX})</td>
<td>\text{ns}</td>
</tr>
<tr>
<td>\text{TSGE}_{RX}</td>
<td>\text{Uniform}</td>
<td>-4</td>
<td>+4</td>
<td>\text{TSGE}_{RX} \sim U(-\text{TSGERX}, +\text{TSGERX})</td>
<td>\text{ns}</td>
</tr>
<tr>
<td>\text{DTSE}_{TX}</td>
<td>\text{Uniform}</td>
<td>-2</td>
<td>+2</td>
<td>\text{DTSE}_{TX} \sim U(-\text{DTSETX}, +\text{DTSETX})</td>
<td>\text{ns}</td>
</tr>
<tr>
<td>\text{DTSE}_{RX}</td>
<td>\text{Uniform}</td>
<td>-1</td>
<td>+1</td>
<td>\text{DTSE}_{RX} \sim U(-\text{DTSERX}, +\text{DTSERX})</td>
<td>\text{ns}</td>
</tr>
</tbody>
</table>

Formulae below take into account the unit of the input value. Formulae in the main section of the September presentation did not.
## Input Parameters

<table>
<thead>
<tr>
<th>Error</th>
<th>Default Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pDelayInterval$</td>
<td>1,000</td>
<td>ms</td>
</tr>
<tr>
<td>$pDelayTurnaround$</td>
<td>10</td>
<td>ms</td>
</tr>
<tr>
<td>$residenceTime$</td>
<td>10</td>
<td>ms</td>
</tr>
</tbody>
</table>
## Input Correction Factors

All formulae in this analysis are ultimately composed of one of these inputs:

**Input Errors** *(Clock Drift or Timestamp Errors)*

**Input Parameters**

**Input Correction Factors**

<table>
<thead>
<tr>
<th>Error</th>
<th>Default Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{meanLinkDelay}^{\text{errorCorrection}})</td>
<td>0</td>
<td>Value (0-1)</td>
</tr>
<tr>
<td>(\text{driftRate}^{\text{errorCorrection}})</td>
<td>0</td>
<td>Value (0-1)</td>
</tr>
<tr>
<td>(\text{pDelayRespSync}^{\text{correction}})</td>
<td>0</td>
<td>Value (0-1)</td>
</tr>
<tr>
<td>(\text{mNRRsmoothingN})</td>
<td>1</td>
<td>Number (1+)</td>
</tr>
</tbody>
</table>
Measured Neighbor Rate Ratio (mNRR)

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

\[ mNRR_{error} = mNRR_{errorTS} + mNRR_{errorCD} \] (ppm)
$$mNRR_{errorTS} = \left( \frac{(t_{4Perror} - t_{4PerrorPrevious}) - (t_{3Perror} - t_{3PerrorPrevious})}{pDelayInterval \times mNRR_{smoothingN}} \right)$$
\[ mNRR_{\text{errorCD}}(n) = (1 - \text{driftRate}_{\text{error\ correction}}) \times mNRR_{\text{smoothing}} N \times \left( \frac{p\text{DelayInterval}}{2 \times 10^3} \right) (\text{clockDrift}(n-1) - \text{clockDrift}(n)) \]
The formula for effective NRR smoothing is:

\[ mNRR_{\text{errorDrift}}(n) = (1 - \text{drift Rate}_{\text{error correction}}) \times mNRR_{\text{smoothing}}(n) \times \left(\frac{\text{pDelayInterval}}{2 \times 10^3}\right) (\text{clockDrift}(n-1) - \text{clockDrift}(n)) \]
Formulae – \( mNRR_{\text{error}} \)

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

\[
mNRR_{\text{errorCD}}(n) = (1 - \text{driftRate}_{\text{errorCorrection}}) \times mNRR_{\text{smoothing}}N \times \left( \frac{\text{pDelayInterval}}{2 \times 10^3} \right) (\text{clockDrift}(n - 1) - \text{clockDrift}(n))
\]

\[
mNRR_{\text{errorTS}} = \left( \frac{(t_{4PDError} - t_{4PDErrorPrevious}) - (t_{3PDError} - t_{3PDErrorPrevious})}{\text{pDelayInterval} \times mNRR_{\text{smoothing}}N} \right)
\]

\[
t_{3PDErrorPrevious} = TSGE_{TX} + DTSE_{TX} \quad \quad t_{4PDErrorPrevious} = TSGE_{RX} + DTSE_{RX}
\]

The factors \( t_{3PDError} \) and \( t_{4PDError} \) are the same as in the \( \text{pDelay}_{\text{error}} \) calculation.
(Same value. No need for new random numbers, as the same \( \text{pDelayResp} \) message is used to measure NRR.)
Observations of $mNRR_{\text{error}}$ Behaviour

- Errors in mNRR do not accumulate along the chain of nodes
- An mNRR error due to clock drift at one node will tend to be reversed at the next node.
  - This does not apply for mNRR errors due to clock drift at the GM
- $mNRR_{\text{error\_Cd}}(n) = \left(\frac{p\text{DelayInterval}}{2 \times 10^3}\right)(\text{clockDrift}(n-1) - \text{clockDrift}(n))$
  $mNRR_{\text{error\_Cd}}(n + 1) = \left(\frac{p\text{DelayInterval}}{2 \times 10^3}\right)(\text{clockDrift}(n) - \text{clockDrift}(n + 1))$

- mNRR errors due to Timestamp errors are independent of each other. DTE errors at one node due to this component will not tend to be reversed at the next node.
- mNRR errors due to clock drift are modelled as a combination of two uniform distributions (clock drifts at n and n-1) and will have a simple probability distribution
  - Errors due to Timestamp errors are modelled as combinations of eight uniform distributions and will be more complex.
Rate Ratio

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

\[ RR_{\text{error}}(n) = RR_{\text{error}}(n-1) + mNRR_{\text{error}}(n) + RR_{\text{errorCD}}(n) \]
\[
RR_{\text{errorCD}}(n) = (1 - \text{driftRate}_{\text{errorCorrection}}) \times \frac{\text{delay}_{\text{mNRR Sync}}}{10^3} \left( \text{clockDrift}(n - 1) - \text{clockDrift}(n) \right)
\]
Formulae – $RR_{error}$

$$RR_{error}(n) = RR_{error}(n - 1) + mNRRe_{error}(n) + RR_{errorCD}(n)$$

$$RR_{errorCD}(n) = (1 - driftRate_{errorCorrection}) \times \frac{delay_{mNRRSync}}{10^3} (clockDrift(n - 1) - clockDrift(n))$$

$$delay_{mNRRSync} \sim U(0, (1 - pDelayRespSync_{correction})pDelayInterval)$$

Does not include any effect of changing clock drift (ppm/s²) during Residence Time. See speaker notes in $RR_{error}$ section for details.
Observations of $RR_{\text{error}}$ Behaviour

- $mNRR_{\text{error}}$ feeds directly into $RR_{\text{error}}$ where they accumulate
  - Since $mNRR_{\text{error}}$ due to clock drift at one node tends to reverse at the next node, the component of $RR_{\text{error}}$ due to this will tend not to increase along a chain of devices.
    - This does not apply for error due to GM clock drift as it only appears in $mNRR_{\text{error}}$ at the first hop.
  - $mNRR_{\text{error}}$ due to timestamp errors are much less likely to cancel out so $RR_{\text{error}}$ from this source is more likely to increase along a chain of devices.

- $RR_{\text{error}}$ due to clock drift during the delay between measurement of $mNRR$ and when it is applied to Rate Ratio could cancel out...but only if the delay one node is the same as at the next...which is unlikely.
  - $RR_{\text{errorCD}}$ is therefore much more likely to increase along a chain of devices than the clock drift component of $mNRR_{\text{error}}$

- $RR_{\text{error}}$ due to clock drift can be reduced by decreasing $pDelayInterval$; reduces range of delay $mNRR_{\text{Sync}}$
  - Similar effect if $pDelay$ messaging can be aligned to occur just before Sync message; modelled using $pDelayRespSync_{\text{correction}}$
  - Note: see earlier in this presentation for the effect reducing $pDelayInterval$ has on $mNRR_{\text{error}}$
Residence Time

\[ \text{residenceTime} = RR(t_{1out} - t_{2in}) \]

\[ \text{residenceTime}_{\text{error}} = \text{residenceTime}_{\text{errorTS}} + \text{residenceTime}_{\text{errorRR}} \]
$\text{residenceTime}_{\text{errorTS}} = t_{1\text{Error}} - t_{2\text{Error}}$
residenceTime_{errorRR} = \frac{RR_{error}}{10^6} (residenceTime \times 10^6)

= RRe_{error} \times residenceTime

ns

ns
Formulae – \( \text{residenceTime}_{\text{error}} \)

\[
\text{residenceTime}_{\text{error}} = \text{residenceTime}_{\text{errorTS}} + \text{residenceTime}_{\text{errorRR}}
\]

\[
\text{residenceTime}_{\text{errorTS}} = t_{1_{\text{error}}} - t_{2_{\text{error}}}
\]

\[
t_{2_{\text{error}}} = TSGE_{RX} + DTSE_{RX}
\]

\[
t_{1_{\text{error}}} = TSGE_{TX} + DTSE_{TX}
\]

\[
\text{residenceTime}_{\text{errorRR}} = \frac{RR_{\text{error}}}{10^6} (\text{residenceTime} \times 10^6 + \text{residenceTime}_{\text{errorTS}})
\]

\[
= RRe_{\text{error}} \times \left( \text{residenceTime} + \frac{\text{residenceTime}_{\text{errorTS}}}{10^6} \right)
\]
Observations of \( \text{residenceTime}_{\text{error}} \) Behaviour

- \( \text{residenceTime}_{\text{error}} \) due to timestamp error is independent of Residence Time.
  - Reducing Residence Time will have no effect on this source of error.
- \( \text{residenceTime}_{\text{error}} \) due to \( \text{RR}_{\text{error}} \) is proportional to Residence Time
  - Reducing Residence Time can reduce this source of error.
- Since larger \( \text{RR}_{\text{error}} \) is more likely further along a chain of devices, the amount of Residence Time Error at each node is also likely to increase.
  - For example: the component of DTE due to Residence Time Error after 100 hops is more likely to be larger from errors in nodes 51-100 than from nodes 1-50.
meanLinkDelay

\[
\text{meanLinkDelay} = RR \left( \frac{(t_4 - t_1) - NRR(t_3 - t_2)}{2} \right) \text{ ns}
\]

\[
\text{meanLinkDelay}_{\text{error}} = (1 - \text{meanLinkDelay}_{\text{errorCorrection}})(p\text{Delay}_{\text{errorTS}} + p\text{Delay}_{\text{errorNRR}} + p\text{Delay}_{\text{errorRR}}) \text{ ns}
\]
The diagram illustrates the relationship between different timestamps and delays in a network, with a focus on the calculation of `pDelay_{errorTS}`.

The formula for `pDelay_{errorTS}` is given as:

\[
pDelay_{errorTS} = \frac{(t_{4Perror} - t_{1Perror}) - (t_{3Perror} - t_{2Perror})}{2}
\]

This represents the difference in timestamps after adjusting for potential delays, with `ns` indicating nanoseconds as the unit of time.
\[ p\text{Delay}_{\text{errorNRR}} = \text{mNRR}_{\text{error}} \left( \frac{p\text{DelayTurnaround}}{2} \right) \]
Formulae – meanLinkDelay$_{\text{error}}$

\[
\text{meanLinkDelay}_{\text{error}} = (1 - \text{meanLinkDelay}^{\text{error Correction}})(p\text{Delay}^{\text{errorTS}} + p\text{Delay}^{\text{errorNRR}} + p\text{Delay}^{\text{errorRR}}) \\
p\text{Delay}^{\text{errorTS}} = \frac{(t_{4P\text{Delay}^{\text{error}}} - t_{1P\text{Delay}^{\text{error}}}) - (t_{3P\text{Delay}^{\text{error}}} - t_{2P\text{Delay}^{\text{error}}})}{2} \\
t_{1P\text{Delay}}^{\text{error}} = TSE_{TX} + DTSE_{TX} \\
t_{2P\text{Delay}}^{\text{error}} = TSE_{RX} + DTSE_{RX} \\
t_{3P\text{Delay}}^{\text{error}} = TSE_{TX} + DTSE_{TX} \\
t_{4P\text{Delay}}^{\text{error}} = TSE_{RX} + DTSE_{RX} \\
p\text{Delay}^{\text{errorNRR}} = \frac{m\text{NRR}^{\text{error}}}{10^6} \left(\frac{p\text{Delay Turnaround} \times 10^6}{2}\right) \\
= m\text{NRR}^{\text{error}} \left(\frac{p\text{Delay Turnaround}}{2}\right) \\
p\text{Delay}^{\text{errorRR}} = \frac{R\text{E}^{\text{error}}}{10^6} \left(p\text{Delay} + (1 - p\text{Delay}^{\text{error Correction}})(p\text{Delay}^{\text{errorTS}} + p\text{Delay}^{\text{errorNRR}})\right)
\]
Observations of $\text{meanLinkDelay}_{\text{error}}$ Behaviour

- $\text{meanLinkDelay}_{\text{error}}$ due to timestamp error is independent of $\text{pDelayTurnaround}$.
  - Reducing $\text{pDelayTurnaround}$ will have no effect on this source of error.

- $\text{meanLinkDelay}_{\text{error}}$ due to $\text{mNRR}_{\text{error}}$ is proportional to $\text{pDelayTurnaround}$
  - Reducing $\text{pDelayTurnaround}$ can reduce this source of error.

- The actual Link Delay is not a significant source of error
  - Only needs to be included as part of $\text{pDelay}_{\text{errorRR}}$...which is small enough to ignore for the purposes of this analysis
Formulae – Top Level

\[ DTE(x) = \sum_{n=1}^{x} (\text{meanLinkDelay}_{\text{error}}(n) + \text{residenceTime}_{\text{error}}(n)) \]

\[ \text{meanLinkDelay}_{\text{error}} = (1 - \text{meanLinkDelay}_{\text{errorCorrection}})(p_{\text{Delay}_{\text{errorTS}}} + p_{\text{Delay}_{\text{errorNRR}}} + p_{\text{Delay}_{\text{errorRR}}}) \]

\[ \text{residenceTime}_{\text{error}} = \text{residenceTime}_{\text{errorTS}} + \text{residenceTime}_{\text{errorRR}} \]

The current analysis does not include factors shown in grey.

\( x \) is number of hops.
Special Cases

\[ \text{clockDrift}(0) = \text{clockDrift}_{GM} \]

For the first hop \((n = 1)\), \(n - 1 = 0\), i.e. the first device in chain, which is the GM.

\[ \text{residenceTime}_{error}(x) = 0 \]

For the last hop \((n = x)\), the Sync message is not passed on so there is no Residence Time Error.
Analysis Carried Out Using R & RStudio

• R is available here: [https://www.r-project.org/](https://www.r-project.org/)
  • Open source license: various, but mostly GNU GPL v2 and GPL v3

• RStudio is available here: [https://www.rstudio.com/products/rstudio/download/](https://www.rstudio.com/products/rstudio/download/)
  • Open source license: GNU Affero GPL v3

• Model uses write.csv and write.table functions, which are part of R.Utils package
  • Install in RStudio by typing...

    install.packages("R.utils")

    ...in the console window.
Availability of Analysis Script

• Intention is to make the script code available under an open source license
  • Probably [BSD 3-Clause](https://www.bsd.org/bsd_license)
  • Other licenses are an option; feedback welcome.
  • Timing TBD. Target is before the end of the year.

• Current plan is to simply make the script code available, not to set up an open source project (e.g. GitHub)
  • If someone wants to
RStudio

- Script
- Environment Variables
- Console
- Plots
RStudio Script Code Summary

• Configuration (Output? Hops? Runs? More charts? Seed value?)
• Inputs (see above)
• Initialize tracking vectors
• Hop 1
  • Calculate main values that contribute to DTE
  • Also calculate values of error components for analysis
  • Calculate MAXabs, MEAN and SIGMA for all main & component values and record in tracking vectors
• Loop: Hops 2+
  • Mostly the same as Hop 1, but errors accumulate where appropriate.
• Plot Charts
Demo

Lenovo Thinkpad T480
Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz  1.90 GHz
16GB RAM
(While running Webex & background corporate apps)
Results

• Monte Carlo analysis of errors allows for many “runs” in very little time.
  • 100 hops & 100,000 runs in <30 seconds
  • Calculating hops takes <15 seconds; rest of time spent generating plots
  • Add approx. 10 seconds to generate additional detailed plots

• Next step: determine if the results are useful.
  • Compare to previous Time Series Simulation
Comparison with Time Series Simulations
Values to Compare Against Time Series $\max |DTE|$ 

- Maximum Absolute Value of Dynamic Time Error ($\max |DTE|$) 
- A multiple of SIGMA ($\sigma$) for DTE at hop 100, based on probability of exceeding it.

* Only valid if data forms a normal (Gaussian) distribution
Dynamic Time Error – Normal Distribution?

• Use of SIGMA to calculate probability of exceeding a value is only valid if the data forms a normal distribution.

• Quantile-Quantile Plot of DTE at hop 100 (100,000 runs)
  • 802.1AS default parameters
  • Clock Drift: ±0.6 ppm/s
  • TSGE & DTSE: ±4ns
  • Data should lie along a straight line (with some variance at extremes expected)

• Result: **YES**, data forms normal distribution.
Which SIGMA?

• Probability of $|DTE| > \sigma$-SIGMA over a period of time?

<table>
<thead>
<tr>
<th>125ms Sync Interval</th>
<th>? x SIGMA</th>
<th>Average Time Before Exceeding</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 per second</td>
<td>5σ</td>
<td>2.5 days</td>
</tr>
<tr>
<td>480 per minute</td>
<td>6σ</td>
<td>2 years</td>
</tr>
<tr>
<td>28,800 per hour</td>
<td>7σ</td>
<td>1,548 years</td>
</tr>
<tr>
<td>691,200 per day</td>
<td></td>
<td></td>
</tr>
<tr>
<td>252,460,800 per year</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Conclusion: use 7σ
  • Revisit if we can’t achieve goals using 7σ
  • Lower multiple (6σ?) would be appropriate for constant time error
  • Lower multiple for dynamic time error might be justified due to combination with constant time error
Enough runs for measurement of SIGMA?

- Increasing number of runs; track max|DTE| and 7
  - Same seed value...but structure of model means first 100 runs when analysing 1,000 runs are not the same as when analysing 100 runs.

<table>
<thead>
<tr>
<th>Runs</th>
<th>max</th>
<th>DTE</th>
<th>(ns)</th>
<th>7σ of DTE (ns at hop 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2,400</td>
<td>6,230</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,000</td>
<td>3,710</td>
<td>6,380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>3,610</td>
<td>6,100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100,000</td>
<td>4,140</td>
<td>6,190</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,000,000</td>
<td>4,380</td>
<td>6,190</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 100,000 runs is a good compromise between speed and accuracy
  - Give up very little in terms of accuracy & runs in
Custom Script to Match Time Series Simulation

• The Time Series Simulation includes two factors that must be specifically modelled differently than the main model.
• Dynamic Time Error is +8ns or -8ns with 50% probability of either.
  • Not realistic. Not recommended for main model.
• mNRR is determined by calculating NRR using most recent and Nth prior pDelayResponse messages (included in the main model via mNRRsmoothingN) and then taking median value of previous N calculations (not included in the main model).
  • N is an odd number; values of 11 and 7 have been used.
  • Assuming clocks drift linearly, this will result in the \(\frac{N+1}{2}\) previous value being used; e.g. if N=11, 6th previous value.
    • This results in an additional clock drift between effective measurement and mNRR of \(\frac{N+1}{2} - 1\) = \(\frac{N-1}{2}\)
  • This has no effect on \(mNRR_{errorTS}\), but increases \(mNRR_{errorCD}\) by an additional factor of N-1.
    • Modelled by changing the effect of mNRRsmoothingN on mNRRerrorCD from...
      * mNRRsmoothingN
    ...to...
      * ((mNRRsmoothingN * 2) – 1)
  • Taking median of previous N calculations only has a negative effect; it only adds an additional source of error and mitigates no existing error. Therefore, not recommended for main model (or use in practical systems).
### Comparison with Time Series Simulation

| Case | residenceTime (ms) | TSge (±ns) | DTSe (±ns) | Temp Factor | mNRR | Smoothing | Time Series max|DTE| @100 hops (ns) | % Diff 100% Temp → 10% Temp | Monte Carlo max|DTE| @100 hops (ns) | Monte Carlo max(7sigma) DTE | TS → MC max|DTE| 7sigma | TS → MC 7sigma |
|------|-------------------|------------|------------|-------------|------|-----------|----------------|----------------|-----------------|------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1    | 1                 | 4          | 8          | 100%        | No   |           | 2737           | 1715           | -36.9%          | 0.4%            | 2719           | 1666           | 2649           | -42.4%         | -8.4%          |
| 2    | 4                 | 2          | 8          | 100%        | No   |           | 2891           | 1666           | 1715           | -36.9%          | 0.4%            | 1666           | 2649           | 2719           | -36.9%          | 0.4%            |
| 3    | 4                 | 4          | 8          | 100%        | No   |           | 6023           | 5967           | -0.9%           | 49.1%           | 8983           | 7822           | 5.0%          | 41.7%          |
| 4    | 10                | 4          | 8          | 100%        | No   |           | 6155           | 5848           | 2729           | -36.9%          | 0.4%            | 14470          | 21122          | 6.3%           | 39.5%          |
| 5    | 10                | 2          | 8          | 100%        | No   |           | 13252          | 14212          | 21089          | 7.2%           | 59.1%           | 14212          | 21089          | 7.2%           | 59.1%           |
| 6    | 4                 | 4          | 8          | 10%         | No   |           | 2058           | 1715           | 2729           | -43.9%          | -10.8%          | 1715           | 2729           | 7.2%           | 59.1%           |
| 7    | 1                 | 2          | 8          | 10%         | No   |           | 2777           | 1666           | 2649           | -40.0%          | -4.6%           | 1666           | 2649           | 7.2%           | 59.1%           |
| 8    | 4                 | 4          | 8          | 10%         | No   |           | 6341           | 5967           | 8983           | -5.9%           | 41.7%           | 5967           | 8983           | 7.2%           | 59.1%           |
| 9    | 4                 | 4          | 2          | 10%         | No   |           | 6045           | 5848           | 8722           | -3.3%           | 44.3%           | 5848           | 8722           | 7.2%           | 59.1%           |
| 10   | 10                | 4          | 8          | 10%         | No   |           | 13942          | 14470          | 21722          | 3.8%           | 55.8%           | 14470          | 21722          | 7.2%           | 59.1%           |
| 11   | 10                | 4          | 8          | 10%         | No   |           | 12766          | 14212          | 21089          | 7.2%           | 59.1%           | 14212          | 21089          | 7.2%           | 59.1%           |
| 12   | 10                | 4          | 8          | 10%         | No   |           | 12766          | 14212          | 21089          | 11.3%          | 65.2%           | 14212          | 21089          | 7.2%           | 59.1%           |
| 13   | 1                 | 0          | 0          | 100%        | No   |           | 32             | 13             | -99.4%          | -40.6%          | 19             | 13             | 13             | -99.4%          | -40.6%          |
| 14   | 1                 | 0          | 0          | 100%        | No   |           | 2841           | 1860           | -34.5%          | -7.4%           | 2630           | 1860           | -34.5%        | -7.4%           |
| 15   | 1                 | 0          | 0          | 100%        | No   |           | 733            | 458            | -37.5%          | 3.5%            | 759            | 458            | -37.5%        | 3.5%            |
| 16   | 1                 | 4          | 8          | 100%        | Yes  |           | 579            | 711            | 22.8%           | 86.5%           | 1080           | 711            | 22.8%         | 86.5%           |
| 17   | 1                 | 2          | 8          | 100%        | Yes  |           | 547            | 670            | 22.5%           | 91.7%           | 1049           | 670            | 22.5%         | 91.7%           |
| 18   | 4                 | 4          | 8          | 100%        | Yes  |           | 658            | 962            | 46.2%           | 122.8%          | 1466           | 962            | 46.2%         | 122.8%          |
| 19   | 4                 | 4          | 8          | 100%        | Yes  |           | 627            | 919            | 46.6%           | 124.7%          | 1426           | 919            | 46.6%         | 124.7%          |
| 20   | 10                | 8          | 10%        | 100%        | Yes  |           | 909            | 1653           | 81.8%           | 181.1%          | 2555           | 1653           | 81.8%         | 181.1%          |
| 21   | 10                | 4          | 8          | 100%        | Yes  |           | 856            | 1607           | 87.7%           | 190.8%          | 2489           | 1607           | 87.7%         | 190.8%          |
| 22   | 3                 | 4          | 8          | 10%         | Yes  |           | 584            | 711            | 21.8%           | 84.9%           | 5080           | 711            | 21.8%         | 84.9%           |
| 23   | 1                 | 2          | 8          | 10%         | Yes  |           | 601            | 670            | 11.5%           | 74.5%           | 1049           | 670            | 11.5%         | 74.5%           |
| 24   | 4                 | 4          | 8          | 10%         | Yes  |           | 690            | 962            | 39.4%           | 112.5%          | 1466           | 962            | 39.4%         | 112.5%          |
| 25   | 4                 | 2          | 8          | 10%         | Yes  |           | 787            | 919            | 16.8%           | 81.2%           | 1426           | 919            | 16.8%         | 81.2%           |
| 26   | 10                | 4          | 8          | 10%         | Yes  |           | 1441           | 1653           | 14.7%           | 77.3%           | 2555           | 1653           | 14.7%         | 77.3%           |
| 27   | 10                | 2          | 8          | 10%         | Yes  |           | 1376           | 1607           | 16.8%           | 80.9%           | 2489           | 1607           | 16.8%         | 80.9%           |
| 28   | 10                | 4          | 6          | 10%         | Yes  |           | 1376           | 1607           | 16.8%           | 80.9%           | 2489           | 1607           | 16.8%         | 80.9%           |

#### Single Replication
- **Case 1**: No mNRR Smoothing
- **Case 2**: Timestamp Tests
- **Case 3**: With mNRR Smoothing
- **Case 4**: Multiple Replications

#### 300 Replications
- **Cases 16 to 27**: Multiple Replications

---

**Note**: The table details the comparison between simulation cases and their respective metrics, including residence time, timing skew, and difference factors. The table is divided into single and multiple replication scenarios, showcasing the impact of mNRR smoothing and timestamp tests.
Comparison with Time Series Simulation

No mNRR Smoothing

Timestamp Tests

With mNRR Smoothing

Multiple Replications

ns

Time Series max|DTE|  Monte Carlo max|DTE|  Monte Carlo 7σ
Comparison with Time Series Simulation

• Monte Carlo results broadly align with Time Series results
• Where results deviate the most, they are still usefully close and the Monte Carlo results move in the same direction and by similar amounts as the Time Series results when input parameters change
• The closest matches are between Monte Carlo results and results from multiple replications of the Time Series simulation
• Monte Carlo analysis is definitely good enough for investigating approaches to minimising DTE
Error Analysis
Graphical Representations

• The Monte Carlo Analysis can generate a lot of data

• Prioritising what to data to look at and how to analyse it will help reach useful conclusions quickly

• With that in mind, here is an overview of the range of data...
  • Main Model
    (not the Time Series Match version)

<table>
<thead>
<tr>
<th>Input Errors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GM Clock Drift Max</td>
<td>+0.6 ppm</td>
</tr>
<tr>
<td>GM Clock Drift Min</td>
<td>-0.6 ppm</td>
</tr>
<tr>
<td>Clock Drift (non-GM)</td>
<td>0.6 ±ppm</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>

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pDelay Interval</td>
<td>1000 ms</td>
</tr>
<tr>
<td>pDelay Response Time</td>
<td>10 ms</td>
</tr>
<tr>
<td>residenceTime</td>
<td>10 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Correction Factors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Link Delay</td>
<td>0 %</td>
</tr>
<tr>
<td>Drift Rate</td>
<td>0 %</td>
</tr>
<tr>
<td>pDelayResponse → Sync</td>
<td>0 %</td>
</tr>
<tr>
<td>mNRR Smoothing</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Configuration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hops</td>
<td>100</td>
</tr>
<tr>
<td>Runs</td>
<td>100,000</td>
</tr>
</tbody>
</table>
mNRR Error
Rate Ratio Error – Timestamp & Clock Drift
Rate Ratio Error – mNRR & Direct Clock Drift
Residence Time Error – Timestamp & Clock Drift
Residence Time Error – mNRR
Residence Time – Direct Clock Drift
Mean Link Delay Error – Total & Direct Timestamp Errors
Mean Link Delay Error – mNRRR Error
Dynamic Time Error

![Dynamic Time Error at step 100](image)

![Dynamic Time Error due to Timeslipp Errors at step 100](image)

![Dynamic Time Error due to clock drift at step 100](image)

![Dynamic Time Error over 100 steps](image)

![Dynamic Time Error due to Timeslipp Errors over 100 steps](image)

![Dynamic Time Error due to clock drift over 100 steps](image)
Effect of GM Clock Drift on Residence Time Error and Rate Ratio Error

GM Clock Drift ±0.6 ppm/s

GM Clock Drift 0 ppm/s
Time Sync – How Errors Add Up

All errors in this analysis are caused by either **Clock Drift** or **Timestamp Errors**
Graphical Representation of Error Accumulation

• Each error breaks down into two parts

• The relative weight of each part can be judged by their $7\sigma$ values
Graphical Representation of Error Accumulation
Graphical Representation of Error Accumulation
pDelayInterval
Sensitivity Analysis
pDelayInterval Sensitivity Analysis

![Graph showing Time Series Simulation with a peak at 31.25 ms and a 802.1AS Default peak at 1 s.]

Minimum is at 250ms...

...for this set of parameters.
pDelayInterval Sensitivity Analysis

DTE 7\(\sigma\) (ns)

<table>
<thead>
<tr>
<th>pDelayInterval (ns)</th>
<th>22</th>
<th>31.25</th>
<th>44</th>
<th>62.5</th>
<th>88</th>
<th>125</th>
<th>177</th>
<th>250</th>
<th>354</th>
<th>500</th>
<th>707</th>
<th>1,000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12,100</td>
<td>8,570</td>
<td>6,120</td>
<td>4,360</td>
<td>3,170</td>
<td>2,380</td>
<td>1,970</td>
<td>2,370</td>
<td>3,170</td>
<td>4,400</td>
<td>6,190</td>
<td></td>
</tr>
</tbody>
</table>
pDelayInterval Sensitivity Analysis

±4 ns  TSGE & DTSE  ±2 ns
pDelayInterval Sensitivity Analysis

±0.6 ppm Clock Drift
GM & Others

±0.3 ppm
pDelayInterval Sensitivity Analysis

±0.6 ppm Clock Drift ±0.3 ppm
GM Only
Others remain at ±0.6 ppm
pDelayInterval Sensitivity Analysis

Clock Drift

Others Only

GM remains at ±0.6 ppm

±0.6 ppm  ±0.3 ppm
pDelayInterval Sensitivity Analysis with Lower Timestamp Error

Minimum is at approx 150ms...
...for this set of parameters.
pDelayInterval Sensitivity Analysis with Lower Clock Drift

Minimum is at approx 300ms...

...for this set of parameters.
pDelayInterval Sensitivity - Conclusion

• Choice of pDelayInterval can have a large impact on DTE.
• pDelay interval can be optimised but the optimal choice depends on other parameters and sources of error.
• Monte Carlo Analysis is an effective tool for investigating this further.
pDelayTurnaround Sensitivity Analysis
pDelayTurnaround Sensitivity – 1 s pDelay Interval

DTE is not sensitive to pDelayTurnaround when pDelayInterval is high.
pDelayTurnaround Sensitivity – 31.25 ms pDelay Interval

And DTE is not sensitive to pDelayTurnaround when pDelayInterval is low.
Residence Time Sensitivity Analysis
residenceTime Sensitivity – 1 s pDelay Interval

Lower Residence Time is better.
residenceTime Sensitivity – 31.25 ms pDelay Interval

Lower Residence Time is better... ...proportional to the amount of error coming from Residence Time (vs Mean Link Delay)
Effect of Error Correction Measures
Default Values – No Error Correction Factors
mNRRsmoothingN = 11
\( p_{\text{DelayRespSync}} \text{\ correction} = 98\% \)
mLinkDelayError_{correction} = 98%
driftRateError_{correction} = 98%
mLinkDelayError_{\text{correction}} & \text{driftRateError}_{\text{correction}} = 98\%
Recommendation & Next Steps

- Carry out Time Series simulations to validate
  - Effect of varying input parameters and sources of error.
  - Effect of applying correction factors
- Focus on averaging Mean Link Delay & Clock Drift Compensation
  - Mean Link Delay averaging should be straightforward, but startup may be an issue.
    - Can only be investigated via Time Series simulation
  - Clock Drift Compensation can be estimated by simply reducing Clock Drift
- I am planning another contribution in December on techniques for compensating for Clock Drift
Backup Material
QQ Plots for Different Numbers of Runs

- 100 Runs
- 1,000 Runs
- 10,000 Runs
- 100,000 Runs
- 1,000,000 Runs
DTE Probability Density at Hop 100 for Different Numbers of Runs

100 Runs

1,000 Runs

10,000 Runs

100,000 Runs

1,000,000 Runs
max|DTE| & 7σ of DTE Across 100 Hops for Different Numbers of Runs
Graphics Colour Palette

- Dynamic Time Error
  - Link Delay Error
  - Residence Time Error
- Time Stamp Granularity & Dynamic Time Stamp Error
- Neighbor Rate Ratio Error
- Rate Ratio Error
- Clock Drift

Available at http://www.ieee802.org/1/files/public/docs2021