60802 Time Sync Ad Hoc mNRRsmoothing Optimisation Results

David McCall – Intel Corporation

Version 2

Context

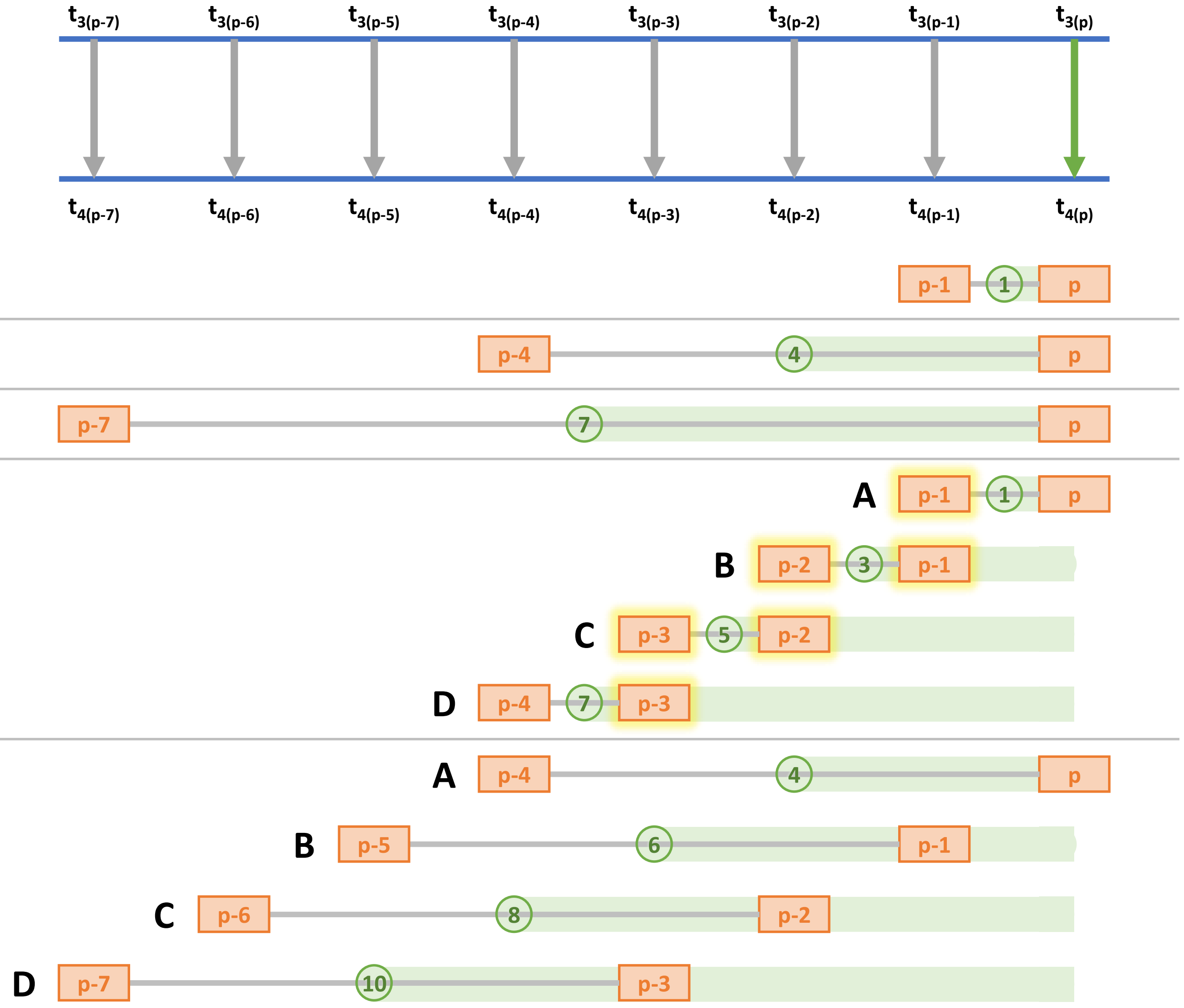
- For context see slides in backup, copied from... https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-mNRRsmoothingN-Optimisation-1022-v1.pdf
- This contribution contains results from the Monte Carlo simulation of 1 hop looking only at $\mathrm{mNRR}_{\mathrm{error}}$

mNRRsmoothingN & mNRRsmoothingA

- mNRRsmoothingN: calculate using Nth previous pDelayResp message
- mNRRsmoothingA: take an average of A previous calculations
- N & A factors can be combined
- Note: mNRRsmoothingM, taking a median of M previous calculations is not recommended (at least for calculating RR via an accumulation of NRR) as it increases timing inconsistency, which increases system-level error.

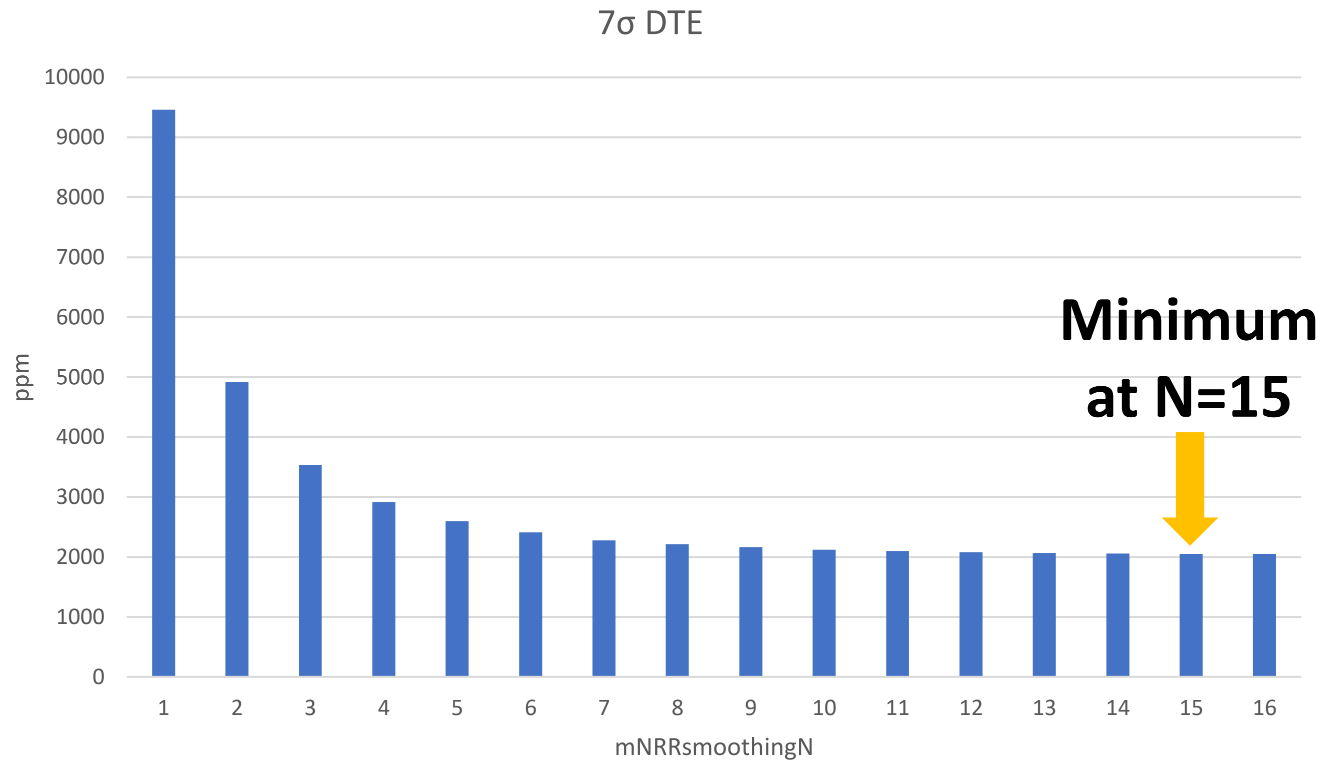
mNRRsmoothingN & mNRRsmoothingA – Examples

Method 1	mNRRsmoothingN = 1 mNRRsmoothingA = 1
	mNRRsmoothingN = 4 mNRRsmoothingA = 1
	mNRRsmoothingN = 7 mNRRsmoothingA = 1
Method 2	mNRRsmoothingN = 1 mNRRsmoothingA = 4 Average of A, B, C & D
Method 3	mNRRsmoothingN = 4 mNRRsmoothingA = 4 Average of A, B, C & D



pDelayInterval 31.25ms – 90% NRR Error Correction

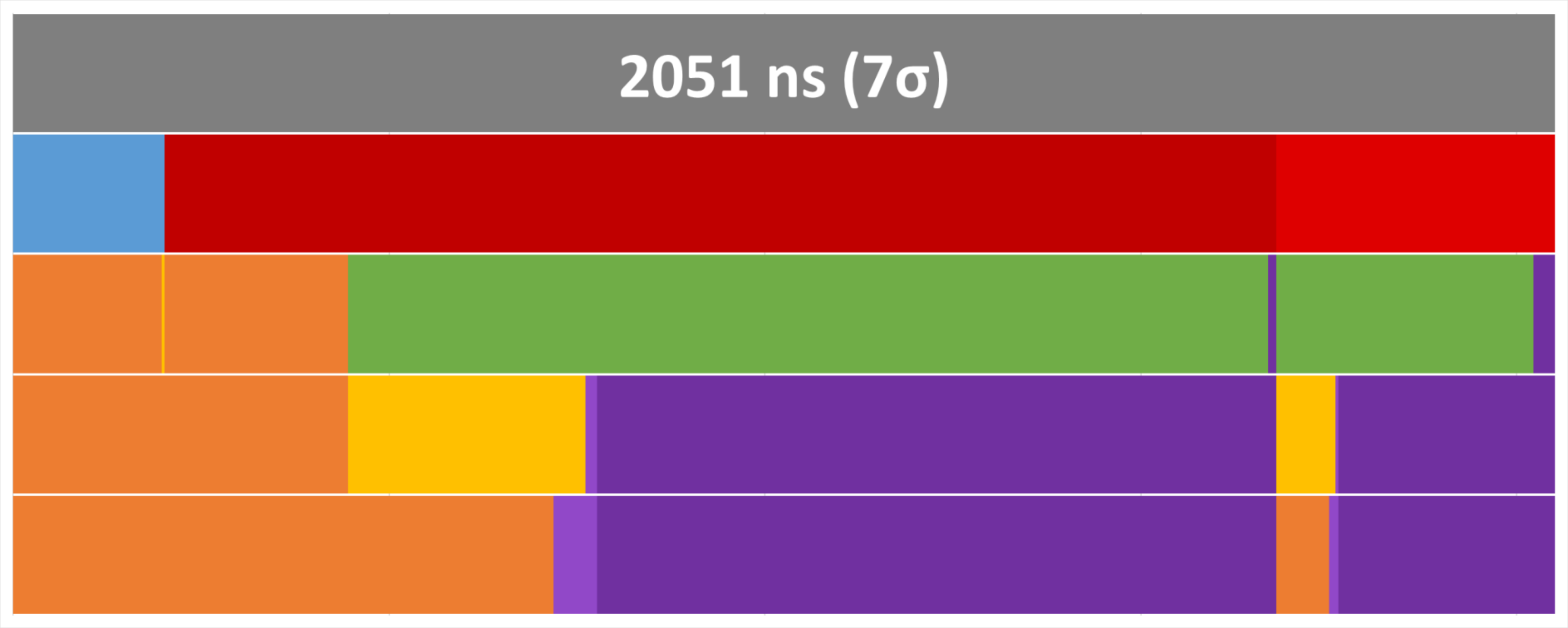
Input Errors		
Drift Type (Linear Temp Ramp)	2	
GM Clock Drift Max	+1.35	ppm/s
GM Clock Drift Min	-1.35	ppm/s
Fraction of GM nodes w/ Drift	80%	
non-GM Clock Drift Max	+1.35	ppm/s
non-GM Clock Drift Min	-1.35	ppm/s
Fraction of non-GM Nodes w/ Drift	80%	
Temp Max	+85.	°C
Temp Min	-40.	°C
Temp Ramp Rate	±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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	VARIABLE	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	100,000	



Minimum at N=15

90% NRR Error Correction chosen so that a larger N is optimal

pDelayInterval 31.25ms – 90% NRR Error Correction



These simulations are only looking at $mNRR_{error}$ (i.e. the yellow section) and elements under it.

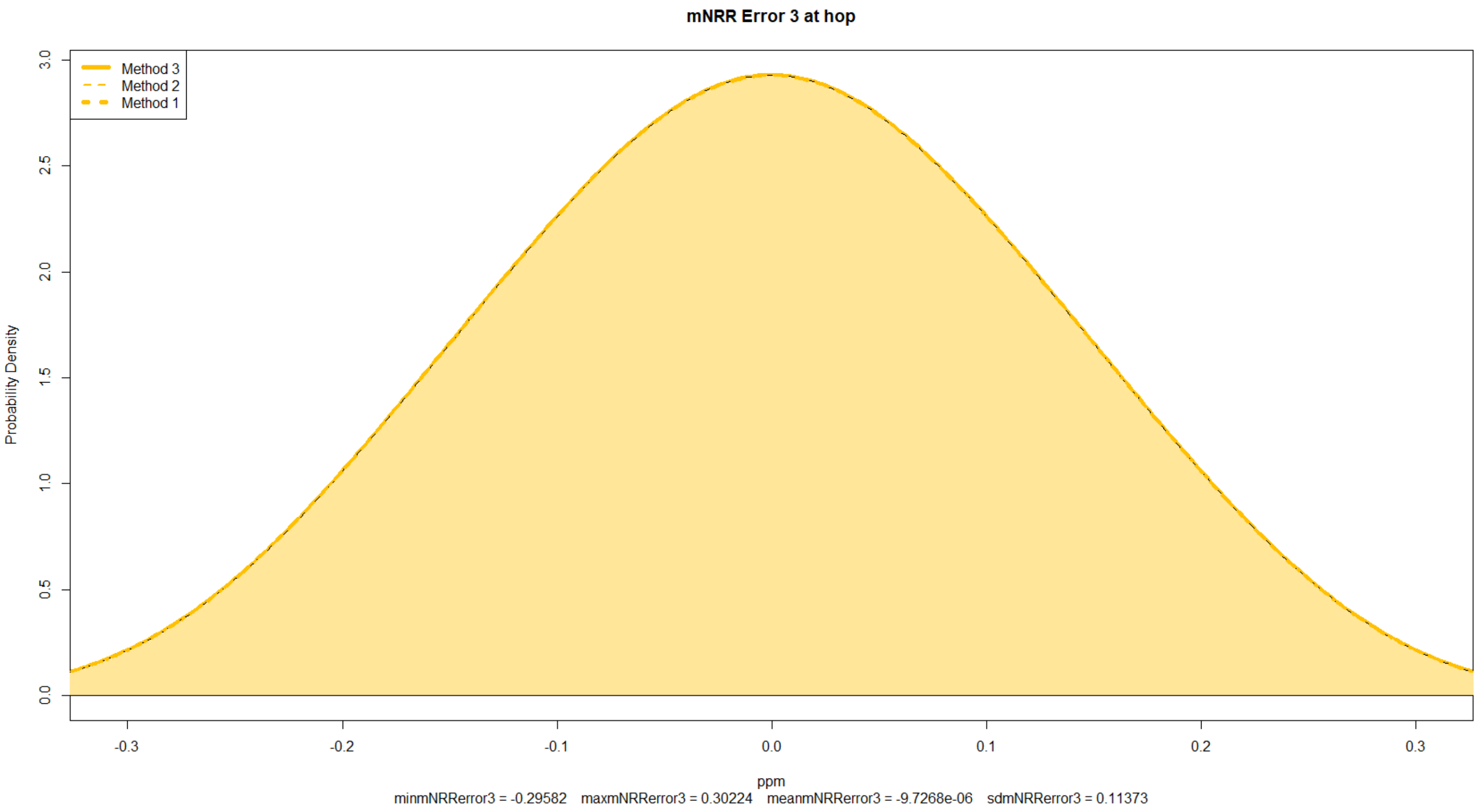
Note that the portion of $mNRR_{error}$ due to Timestamp Error (orange) is larger than the portion due to Clock Drift (purple).

mNRRsmoothing Method Comparison

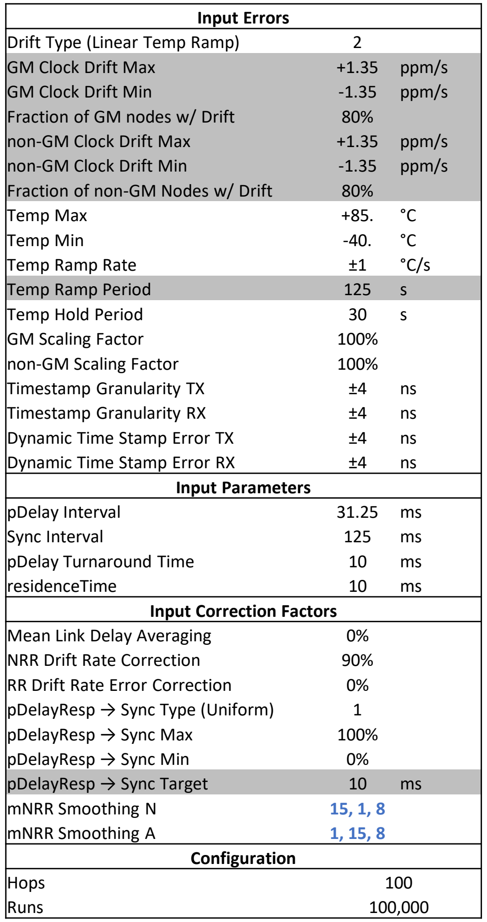
- Three methods...
  - **Method 1**: mNRRsmoothingN = 15   mNRRsmoothingA = 1
  - **Method 2**: mNRRsmoothingN = 1   mNRRsmoothingA = 15
  - **Method 3**: mNRRsmoothingN = 8   mNRRsmoothingA = 8
- Note that all require 16 sets of Timestamp error values and 15 sets of error values due to Clock Drift.
  - The simulation generates arrays with the necessary number of rows (e.g. 16, 15, etc...) and as many columns as there are runs (e.g. 1,000,000).

mNRRsmoothing Method Comparison

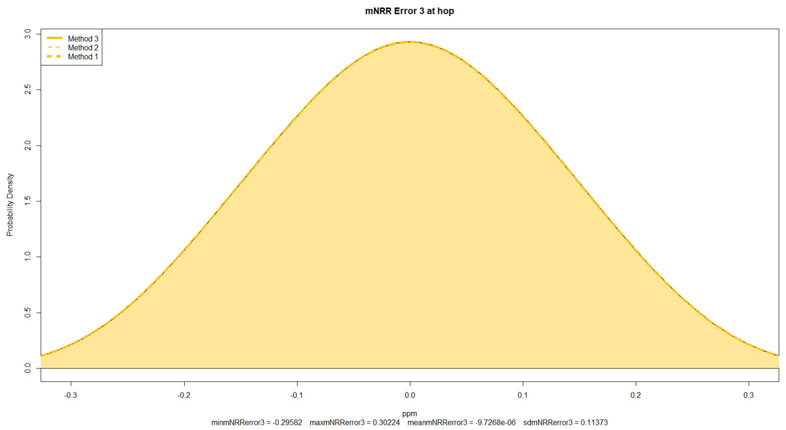
Input Errors		
Drift Type (Linear Temp Ramp)	2	
GM Clock Drift Max	+1.35	ppm/s
GM Clock Drift Min	-1.35	ppm/s
Fraction of GM nodes w/ Drift	80%	
non-GM Clock Drift Max	+1.35	ppm/s
non-GM Clock Drift Min	-1.35	ppm/s
Fraction of non-GM Nodes w/ Drift	80%	
Temp Max	+85.	°C
Temp Min	-40.	°C
Temp Ramp Rate	±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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	**15, 1, 8**	
mNRR Smoothing A	**1, 15, 8**	
Configuration		
Hops	100	
Runs	100,000	



mNRRsmoothing Method Comparison



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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	15, 1, 8	
mNRR Smoothing A	1, 15, 8	
Configuration		
Hops	100	
Runs	100,000	

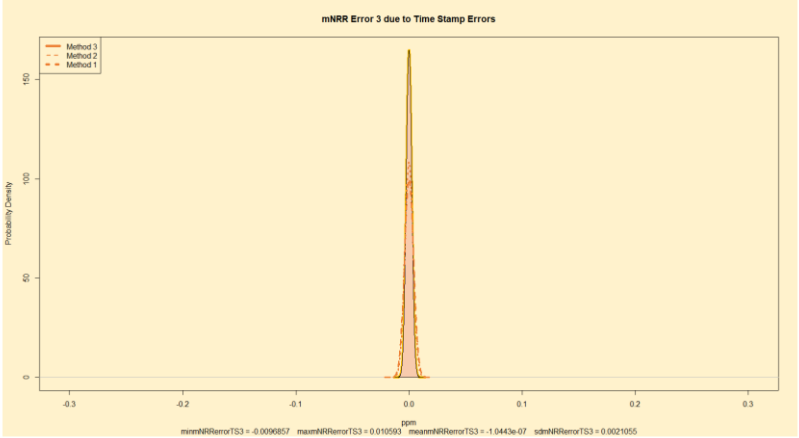


mNRR Error 3 at hop

minmNRRerror3 = -0.29582 maxmNRRerror3 = 0.30224 meanmNRRerror3 = -9.7268e-06 sdmNRRerror3 = 0.11373

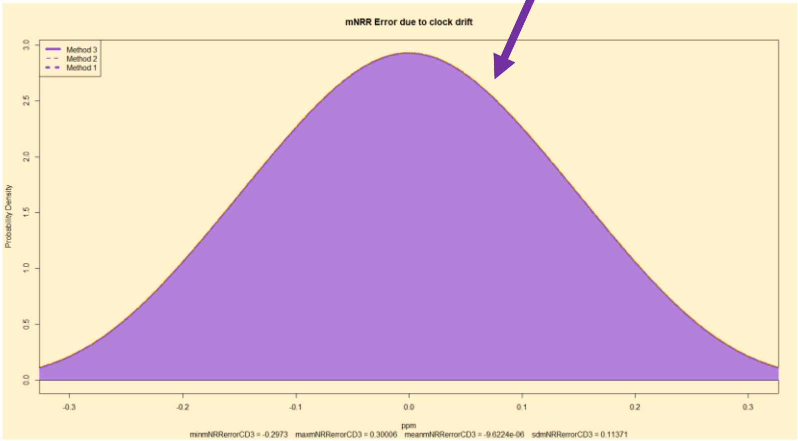
The portion of mNRRerror due to Clock Drift is larger than portion due to Timestamp errors; the opposite of the system-level view (slide 6).

At the system-level, a lot of the error due to Clock Drift cancels out node-to-node but Timestamp error does not. So, a smaller portion at an individual node can be a larger portion at the system level.



mNRR Error 3 due to Time Stamp Errors

minmNRRerrorTS3 = -0.0096857 maxmNRRerrorTS3 = 0.010593 meanmNRRerrorTS3 = -1.0443e-07 sdmNRRerrorTS3 = 0.0021055

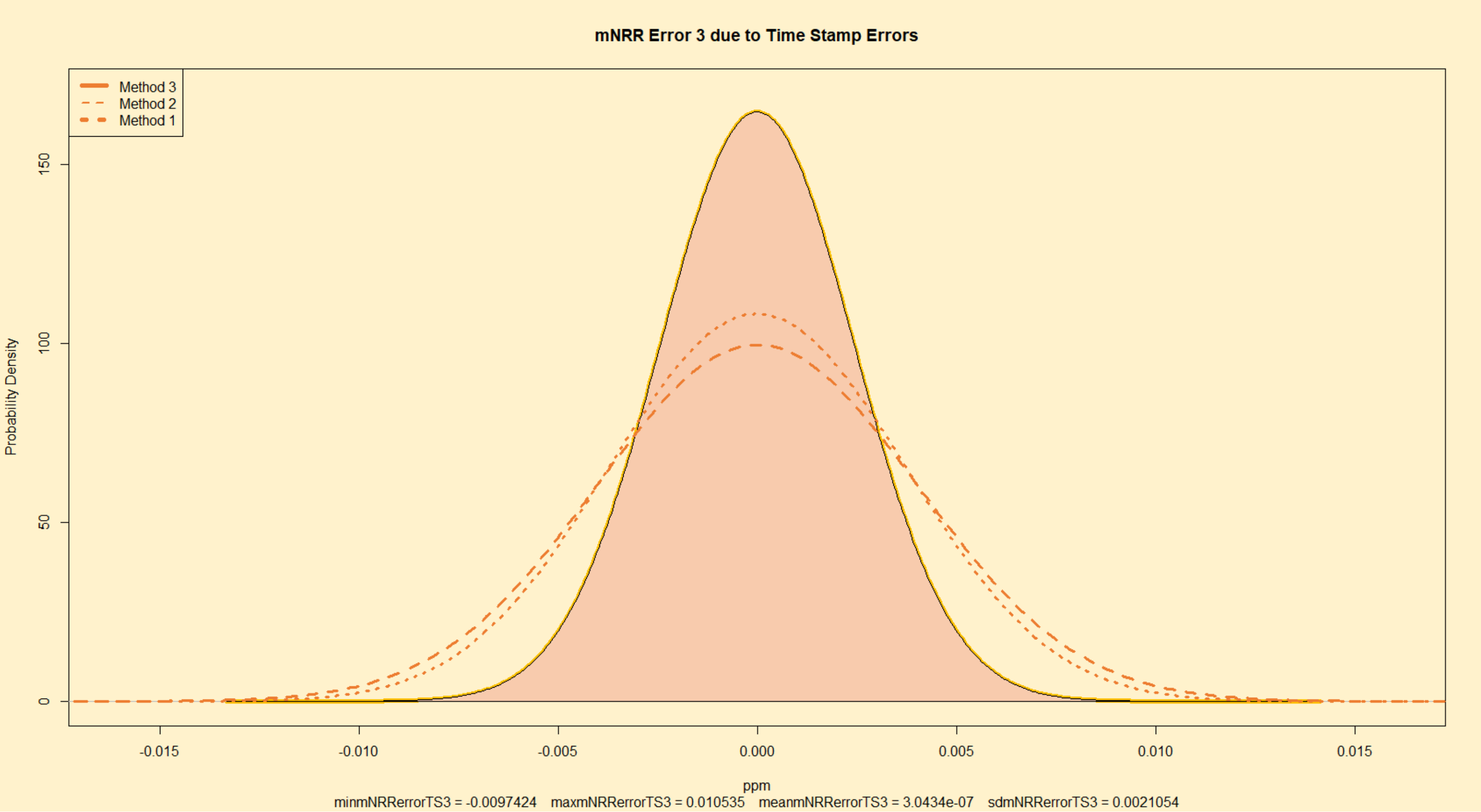


mNRR Error due to clock drift

minmNRRerrorCD3 = -0.2973 maxmNRRerrorCD3 = 0.30006 meanmNRRerrorCD3 = -9.6224e-06 sdmNRRerrorCD3 = 0.11371

mNRRsmoothing Method Comparison

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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	15, 1, 8	
mNRR Smoothing A	1, 15, 8	
Configuration		
Hops	100	
Runs	100,000	



Question...

- Why does Method 2 show an improvement over Method 1?
- From previous presentations, they were shown to be “mathematically equivalent”. [See backup.]

Answer...

- They are only mathematically equivalent if $T_{pdelay2pdelay}$ is exactly the same every time.
- Variations in $T_{pdelay2pdelay}$ affects the amount of error in mNRR...

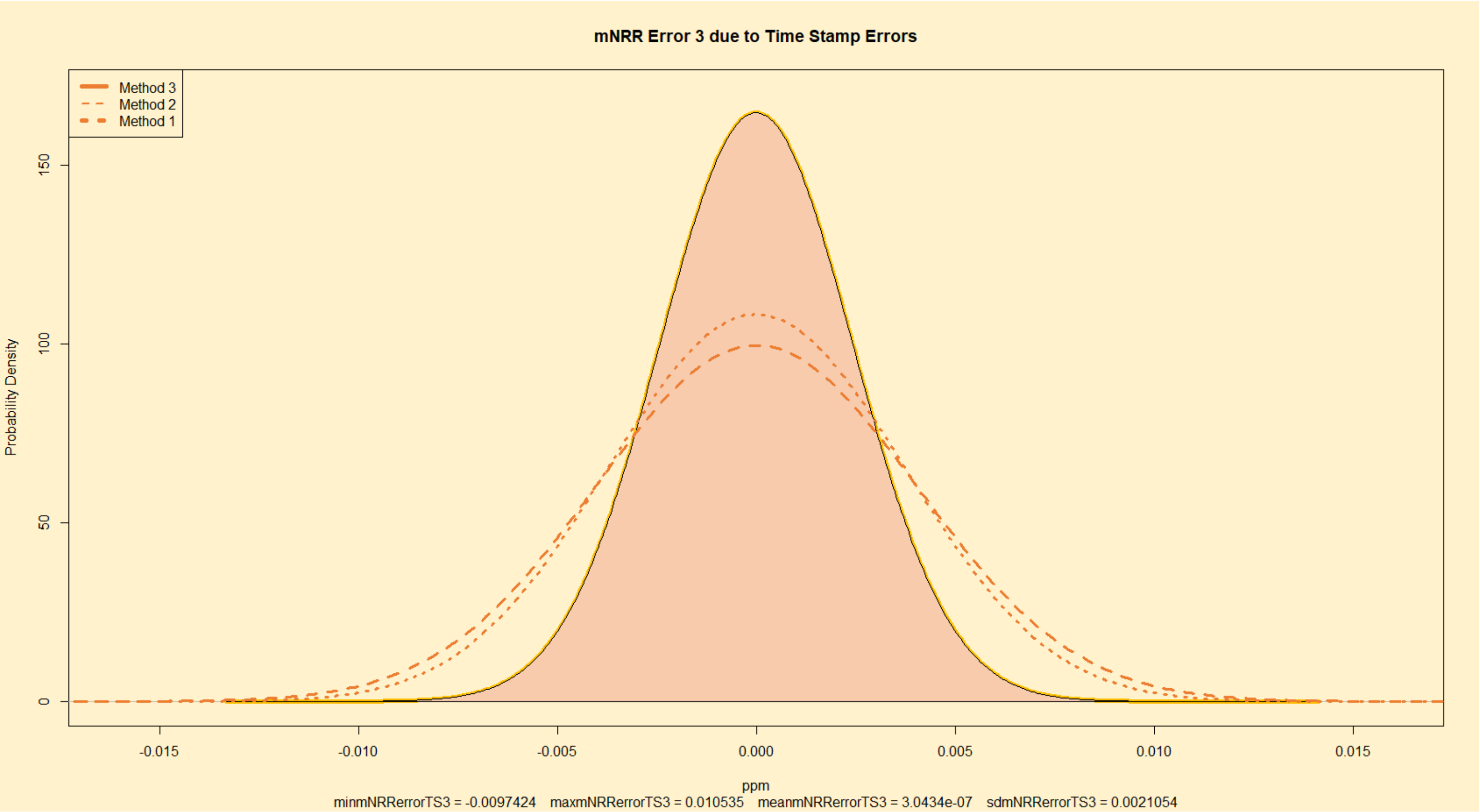
$$mNRR_{errorTS\_X} = \frac{\left(\boldsymbol{t_{3pderror}} - \boldsymbol{t_{3pderror}}'\right) - \left(\boldsymbol{t_{4pderror}} - \boldsymbol{t_{4pderror}}'\right)}{T_{pdelay2pdelay}}$$

$$mNRR_{errorCD\_X} = \frac{T_{pdelay2pdelay}\left(\boldsymbol{clockDrift_{n}} - \boldsymbol{clockDrift_{n-1}}\right)}{2 \times 10^{3}}$$

- Reducing the variability of $T_{pdelay2pdelay}$ should alter the results.
  - IEEE 1588 specifies that $T_{pdelay2pdelay}$ shall be between 90% and 130% of the nominal value, pDelayInterval.

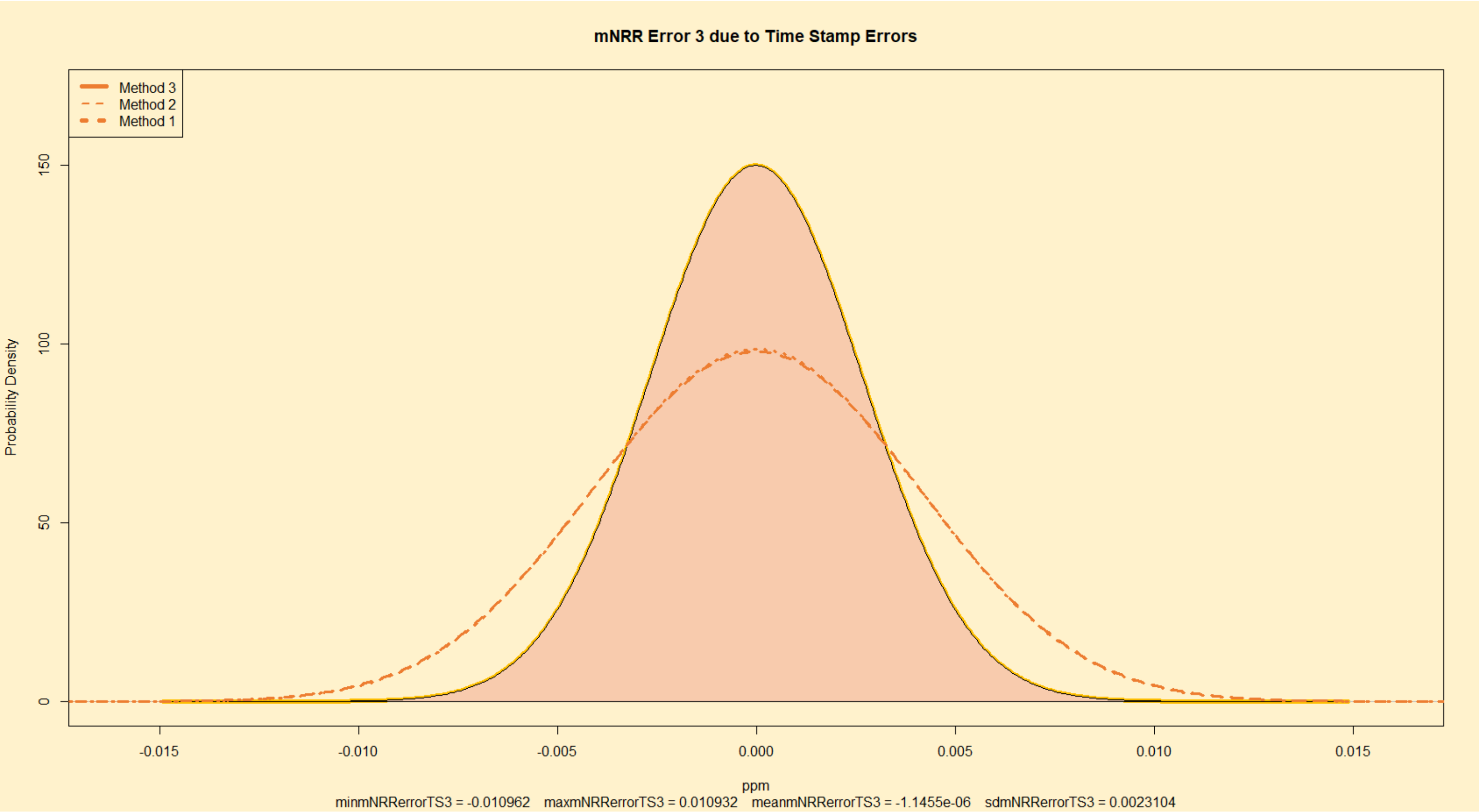
$T_{pdelay2pdelay}$ 90% - 130% Nominal (pDelayInterval)

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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	15, 1, 8	
mNRR Smoothing A	1, 15, 8	
Configuration		
Hops	100	
Runs	100,000	



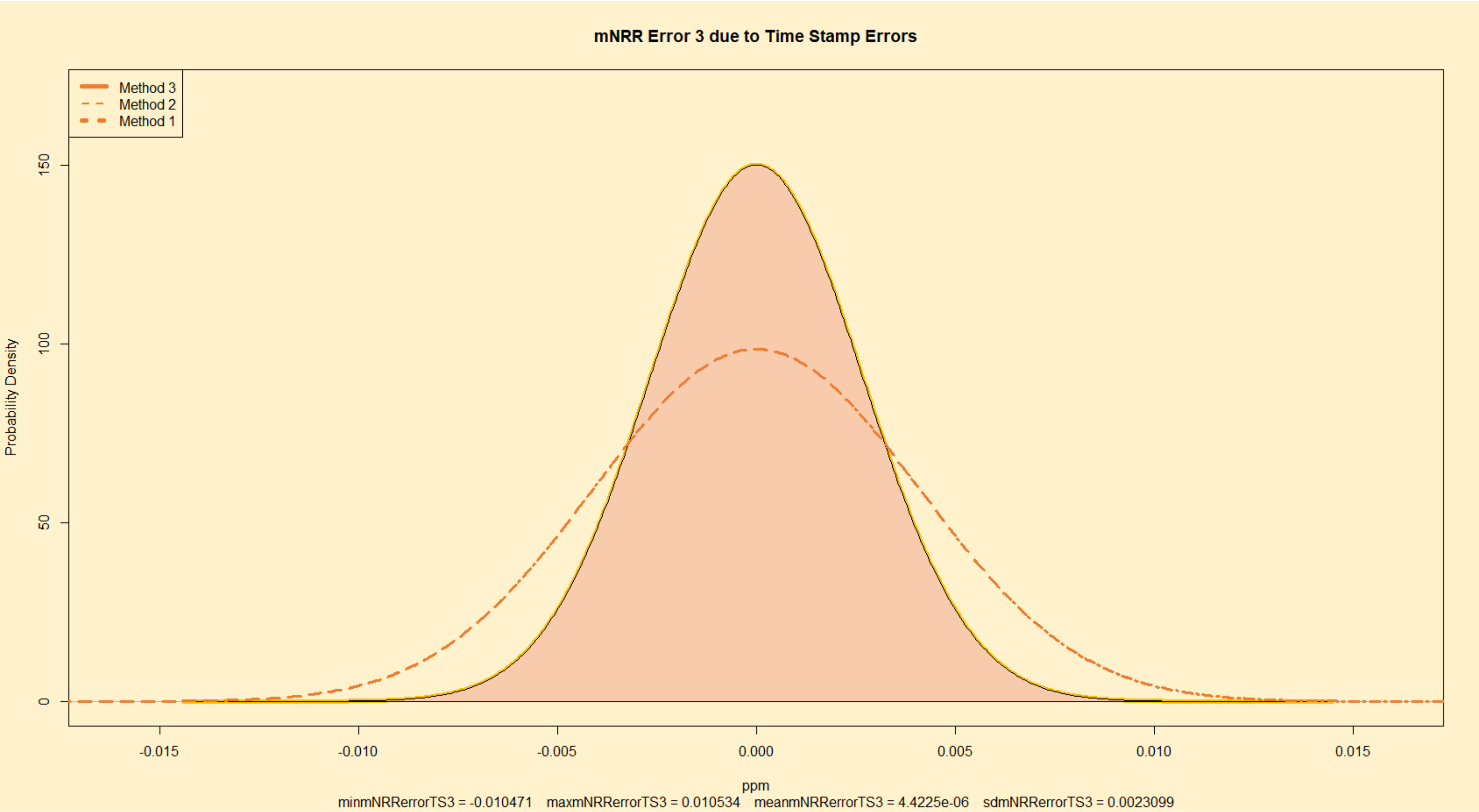
$T_{pdelay2pdelay}$ 95% - 105% Nominal (pDelayInterval)

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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	15, 1, 8	
mNRR Smoothing A	1, 15, 8	
Configuration		
Hops	100	
Runs	100,000	



$T_{pdelay2pdelay}$ 100% Nominal (pDelayInterval)

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	ns
Dynamic Time Stamp Error TX	±4	ns
Dynamic Time Stamp Error RX	±4	ns
Input Parameters		
pDelay Interval	31.25	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	90%	
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	15, 1, 8	
mNRR Smoothing A	1, 15, 8	
Configuration		
Hops	100	
Runs	100,000	



Conclusion

- When using older pDelayResp timestamps to measure NRR, a combination of calculating using Nth previous pDelayResp and taking an average of A previous calculations, where N = A minimises Timestamp Errors. (Method 3)
  - Clock Drift errors are minimally affected for useful values of N & A.
- Reducing the variability of $T_{pdelay2pdelay}$ alters the impact of this optimisation (lower variability → less impact) but does not remove it.
- Note that mNRR should not be optimised in isolation. At a single node, the optimal settings may generate much larger errors due to Clock Drift than Timestamp errors, with a consequent increase in overall $mNRR_{error}$. The errors due to Clock Drift will tend to cancel out at a system level.

Thank You

Backup

Background

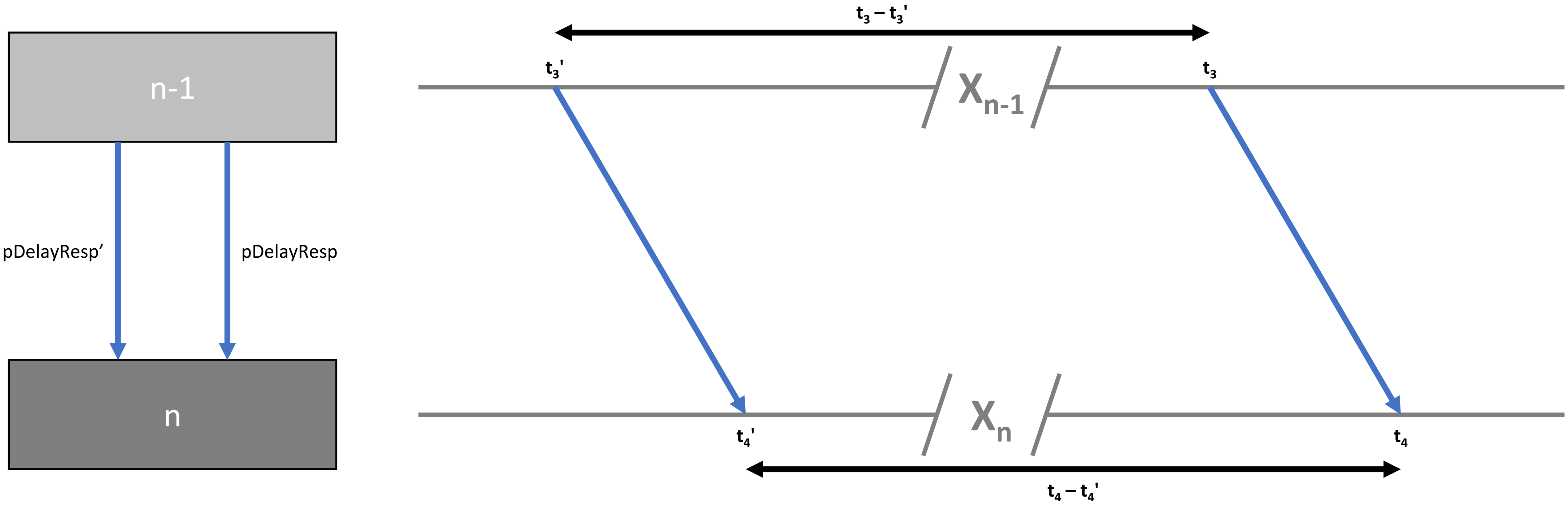
- IEEE 802.1AS measures Rate Ratio (RR) via an accumulation of Neighbor Rate Ratios (NRRs). Classically, NRR is measured via timestamps from the two most recent pDelayResp messages.
- Errors in the measured NRR (mNRR) can arise from Timestamp Errors and errors due to Clock Drift between nodes.
  - As pDelayInterval increases, the effect of errors due to Clock Drift increases, while the effect of Timestamp Errors decreases.
- The balance between errors due to Clock Drift and errors due to Timestamp Errors can also be altered by calculating mNRR using older pDelayResp messages and/or averaging multiple mNRR measurements.
  - I've named this approach mNRRsmoothing as, in general, it reduces the jitter of mNRR values.
- This presentation details different options for mNRRsmoothing and their effect on $\text{mNRR}_{\text{error}}$.

References

[1] “60802 Time Synchronisation – Monte Carlo Analysis: 100-hop Model, “Linear” Clock Drift, NRR Accumulation Overview & Details, Including Equations”, David McCall, IEC/IEEE 60802 contribution, September 2022

[2] “60802 Dynamic Time Sync Error – NRR Medians, Algorithms & Analysis Validation” David McCall & Kevin Stanton, IEC/IEEE 60802 contribution, January 2022

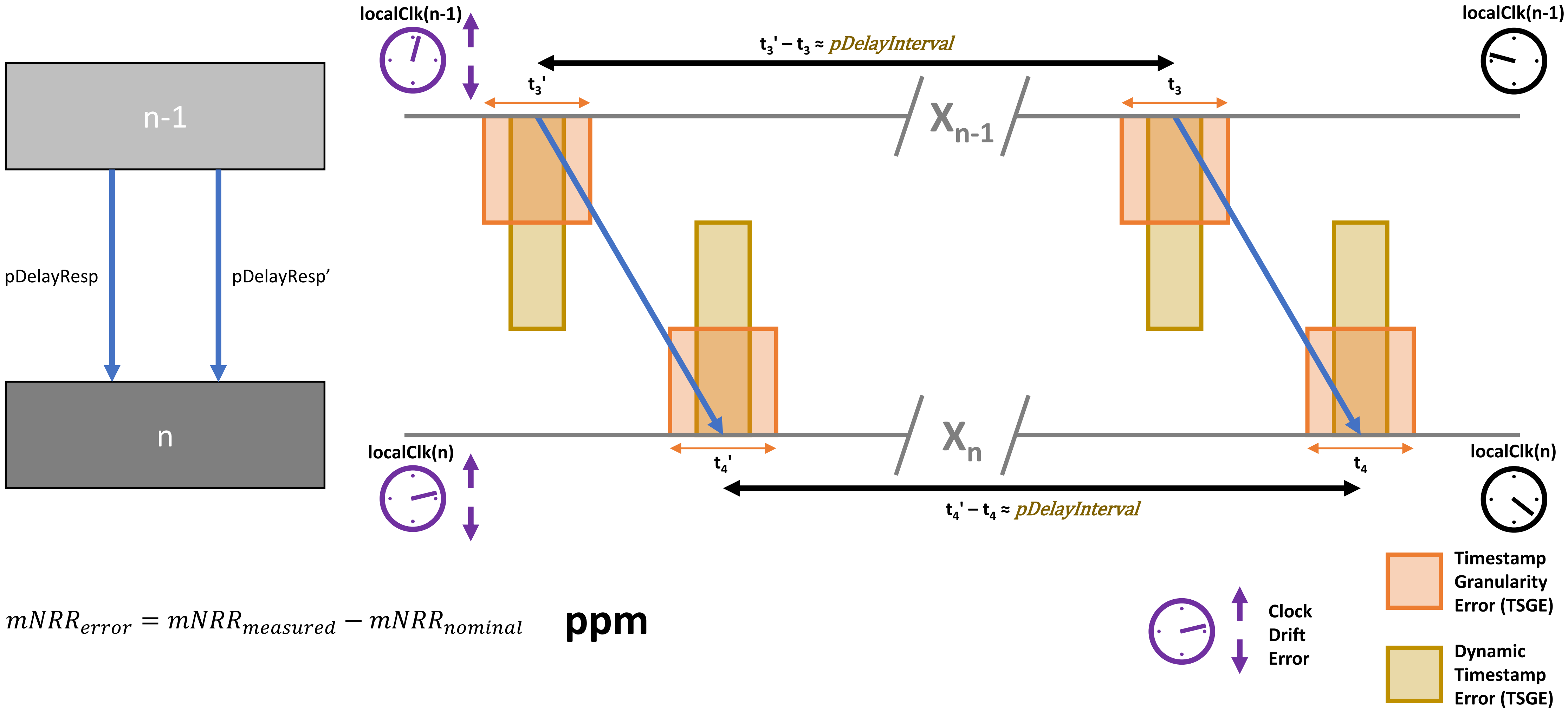
Background - mNRR



$$mNRR = \left(\frac{t_3 - t_3'}{t_4 - t_4'}\right)$$

ppm

Background - mNRR$_{error}$


n-1
n
pDelayResp
pDelayResp’
localClk(n-1)
$t_3' - t_3 \approx$ *pDelayInterval*
t_3'
t_3
localClk(n-1)
X_{n-1}
X_n
localClk(n)
t_4'
t_4
localClk(n)
$t_4' - t_4 \approx$ *pDelayInterval*
Clock Drift Error
Timestamp Granularity Error (TSGE)
Dynamic Timestamp Error (TSGE)

$$mNRR_{error} = mNRR_{measured} - mNRR_{nominal}$$ **ppm**

Background – 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 occured...

$$\boldsymbol{Error_{TGSE}} = \sim U(0, +TSG)$$

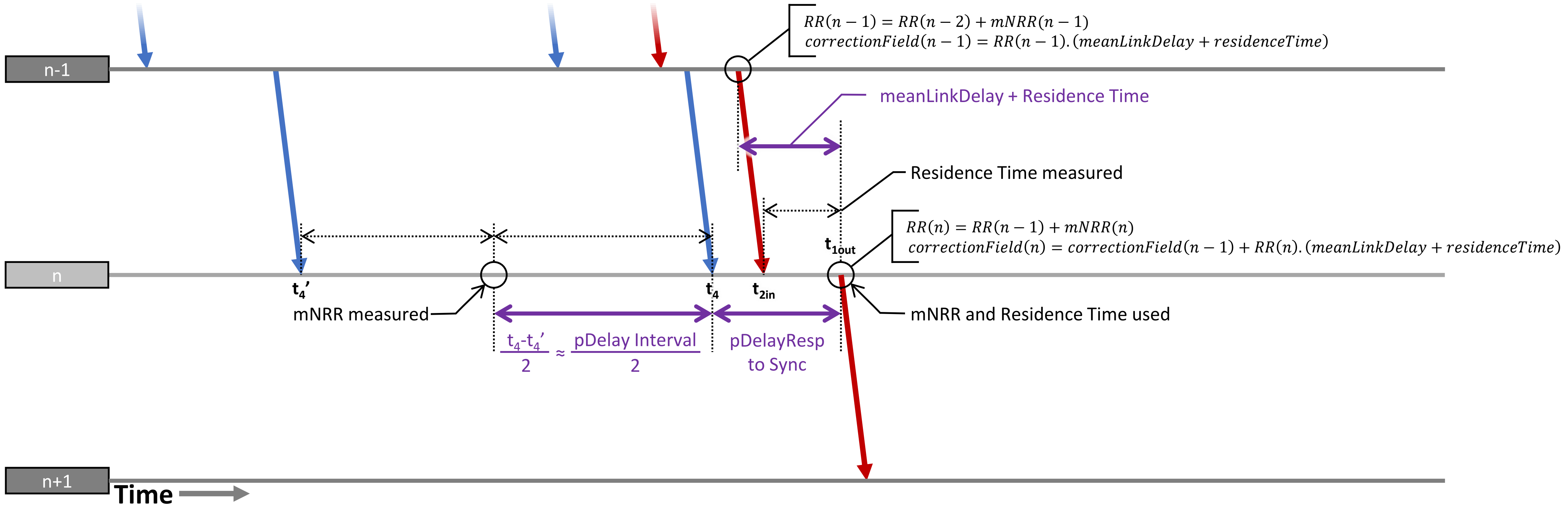
...(where TSG is Timestamp Granularity) however, because the consequent errors are always in interval measurements which involve two events and two timestamps, modelling it as an error between ±TSG/2 is equivalent. In the R Studio script the parameter TGSE represents TSG/2...

$$\boldsymbol{Error_{TSGETX}} = \sim U\left(-\frac{TSG}{2}, +\frac{TSG}{2}\right) = \sim U(-\boldsymbol{TSGE_{TX}}, +\boldsymbol{TSGE_{TX}}) \qquad \boldsymbol{Error_{TSGERX}} = \sim U(-\boldsymbol{TSGE_{RX}}, +\boldsymbol{TSGE_{RX}})$$

- 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.

$$\boldsymbol{Error_{DTSETX}} = \sim U(-\boldsymbol{DTSE_{TX}}, +\boldsymbol{DTSE_{TX}}) \qquad \boldsymbol{Error_{DTSERX}} = \sim U(-\boldsymbol{DTSE_{RX}}, +\boldsymbol{DTSE_{RX}})$$

Background - Clock Drift Error – Relevant Intervals


$RR(n-1) = RR(n-2) + mNRR(n-1)$
$correctionField(n-1) = RR(n-1).(meanLinkDelay + residenceTime)$
n-1
meanLinkDelay + Residence Time
Residence Time measured
$RR(n) = RR(n-1) + mNRR(n)$
$correctionField(n) = correctionField(n-1) + RR(n).(meanLinkDelay + residenceTime)$
t_{1out}
n
t_4'
t_4
t_{2in}
mNRR measured
mNRR and Residence Time used
$\frac{t_4-t_4'}{2} \approx \frac{\text{pDelay Interval}}{2}$
pDelayResp to Sync
n+1
Time

- Error due to drift during NRR measurement. (Node n to Node n-1)
- Error due to drift between measuring and using NRR. (Node n to Node n-1)
- Error due to drift during Residence Time measurement. (Node n to GM)
- Additional error from drift between RR(n-1) calculation, at Node n-1, and use in calculating RR(n). (Node n-1 to GM)
  - In the model the contribution from meanLinkDelay is ignored; only Residence Time is used.

Background – $\text{mNRR}_{\text{error}}$ due to Clock Drift

- Effective NRR Measurement → Actual NRR Measurement
  - Relevant drift is between the current node's clock (n) and the upstream node's clock (n-1).
  - 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.

$$T_{pdelay2pdelay} = \sim U(\boldsymbol{pdelayInterval}.0.9, \boldsymbol{pdelayInterval}.1.3\ )$$

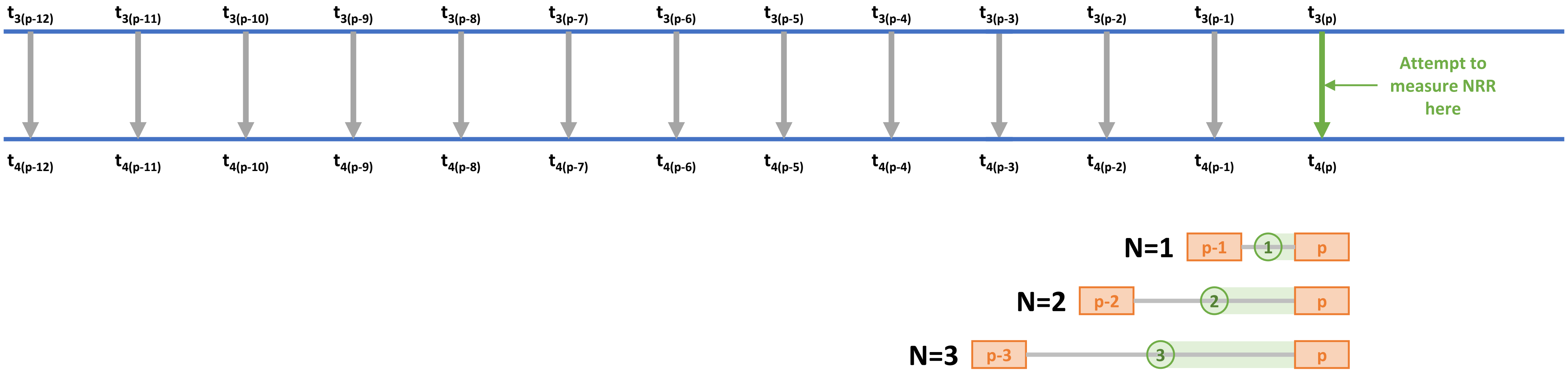
Background - mNNRsmoothingN

- The Monte Carlo approach models using timestamp values from older pDelayResp messages via the *mNRRsmoothingN* parameter adjusting *Tpdelay2pdelay*.

Correction Parameter	Default	Unit	Notes
mNRRsmoothingN	1	-	Must be a whole number, minimum value 1.

$$T_{pdelay2pdelay} = \sum_{x=1}^{mNRRsmoothingN} \sim U(\boldsymbol{pdelayInterval}.\,0.9, \boldsymbol{pdelayInterval}.\,1.3\,)$$

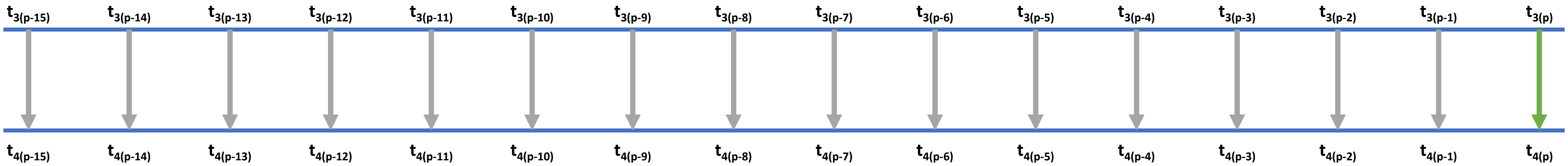
Background – mNRR Smoothing N



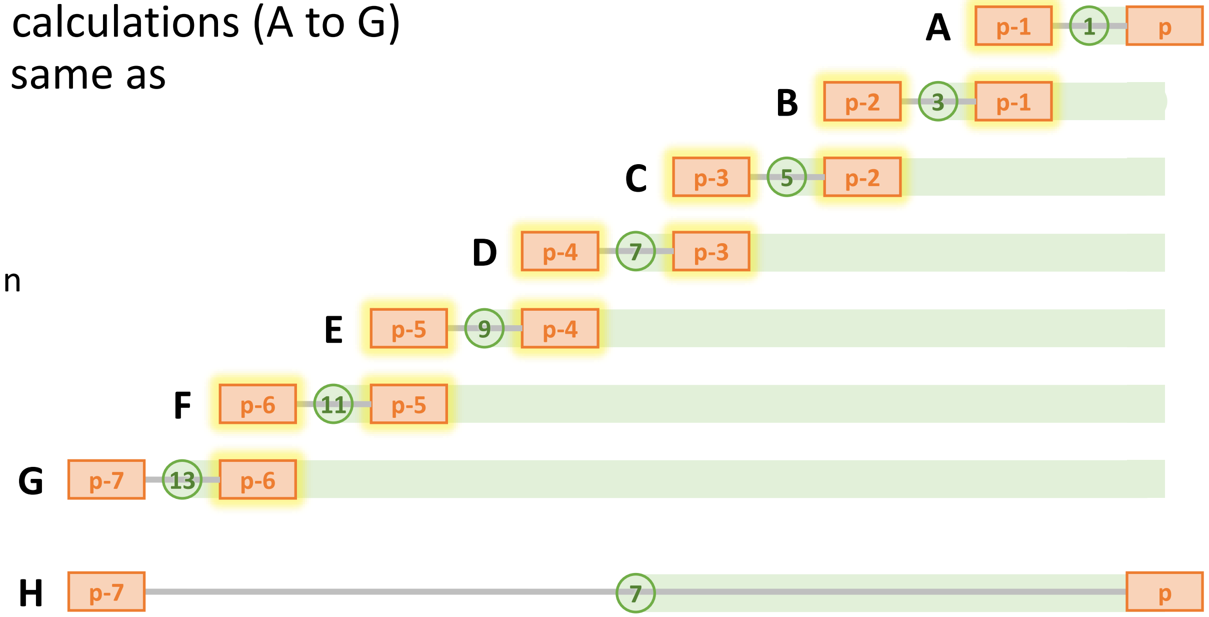
Background – mNRR Smoothing M

- Taking a median of M past mNRR calculations was also investigated, but is not recommended when RR is calculated via an accumulation of NRRs.
  - Use of a Median value means the effective delay between measurement of mNRR and use in Sync is variable, which reduces the cancellation of error due to Clock Drift from node-to-node.
  - See [2] for more detail.
- Note: may be different if calculating RR directly from Sync messages.

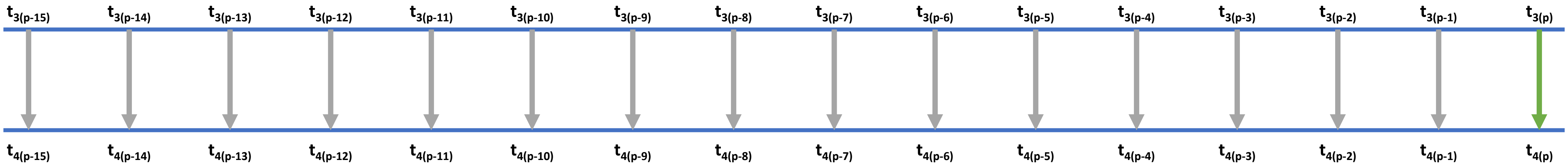
Taking a Simple Average of mNRR Calculations



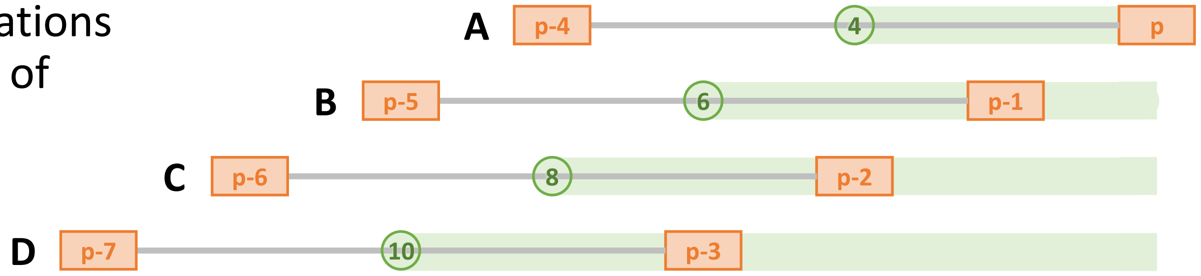
- Taking a simple average of the more recent 8 mNRR calculations (A to G) where mNRRsmoothingN = 1 is mathematically the same as a single calculation where mNRRsmoothingN =7 (H)
  - Exactly the same for Timestamp Error
  - Approximately the same for error due to Clock Drift. The effective measurement point for an average (A to G) is an average of 8 effective measurement points. The effective measurement point for mNRRsmoothingN = 7 is half way between $t_{4(p)}$ and $t_{4(p-7)}$ (i.e. approx. 7x worse that using timestamps from two most recent pDelayResp messages).
- But there are other options...



Taking a Simple Average of mNRR Calculations


$t_{3(p-15)}$ $t_{3(p-14)}$ $t_{3(p-13)}$ $t_{3(p-12)}$ $t_{3(p-11)}$ $t_{3(p-10)}$ $t_{3(p-9)}$ $t_{3(p-8)}$ $t_{3(p-7)}$ $t_{3(p-6)}$ $t_{3(p-5)}$ $t_{3(p-4)}$ $t_{3(p-3)}$ $t_{3(p-2)}$ $t_{3(p-1)}$ $t_{3(p)}$
$t_{4(p-15)}$ $t_{4(p-14)}$ $t_{4(p-13)}$ $t_{4(p-12)}$ $t_{4(p-11)}$ $t_{4(p-10)}$ $t_{4(p-9)}$ $t_{4(p-8)}$ $t_{4(p-7)}$ $t_{4(p-6)}$ $t_{4(p-5)}$ $t_{4(p-4)}$ $t_{4(p-3)}$ $t_{4(p-2)}$ $t_{4(p-1)}$ $t_{4(p)}$

- Taking an average of the most recent 4 mNRR calculations where mNRRsmoothing = 4 delivers some averaging of Timestamp Errors and errors due to Clock Drift
  - Worst case Timestamp Error is the same, but distribution is more Gaussian (with average zero).
  - Error due to Clock Drift is still approx. 7x worse than using timestamps from two most recent pDelayResp messages.


A
p-4
4
p
B
p-5
6
p-1
C
p-6
8
p-2
D
p-7
10
p-3