60802 Time Synchronisation – NRR Tracking & Error Compensation: 1-hop Model; Piece-wise Linear Clock Drift

David McCall (Intel)

References

1. David McCall, “60802 Time Sync Ad Hoc mNRRsmoothing Optimisation Results”, IEC/IEEE 60802 contribution, November 2022 https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-mNRRsmoothingN-Optimisation-Results-1122-v2.pdf

Background

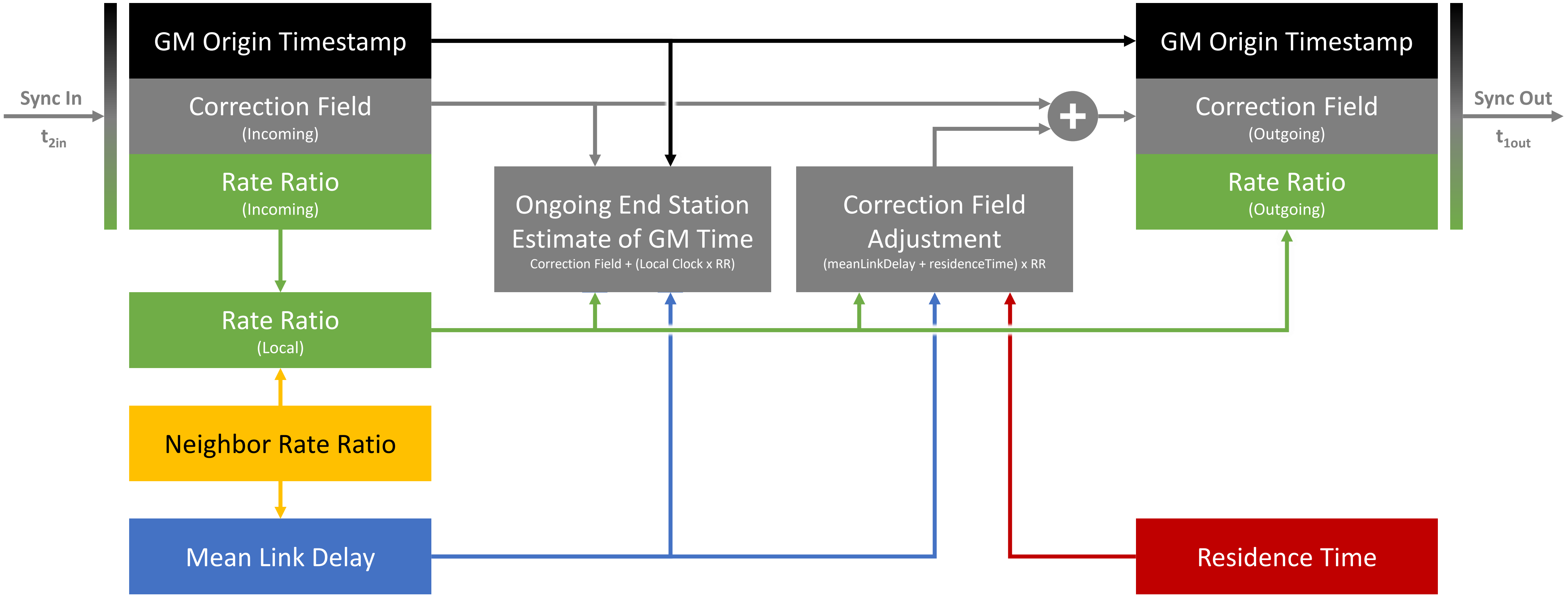
- IEC/IEEE 60802 has a stated requirement of 1us time accuracy over 64 hops (i.e. 65 devices) with a goal of 100 hops (i.e. 101 devices).
- One of the proposed techniques for achieving this goal is to track Clock Drift induced changes to Neighbor Rate Ratio and make adjustments to the NRR calculation to compensate for associated error.
- This presentation describes a potential algorithm for this technique, a Monte Carlo simulation that models it, the results of the simulation and suggested parameters for the algorithm. It also discusses some of the potential implications for the IEC/IEEE 60802 profile.
- As with the multi-hop simulation, preparing a contribution covering **all** the details (every equation, etc…) takes a lot of effort. The plan is to follow up with that at a later date.

Content

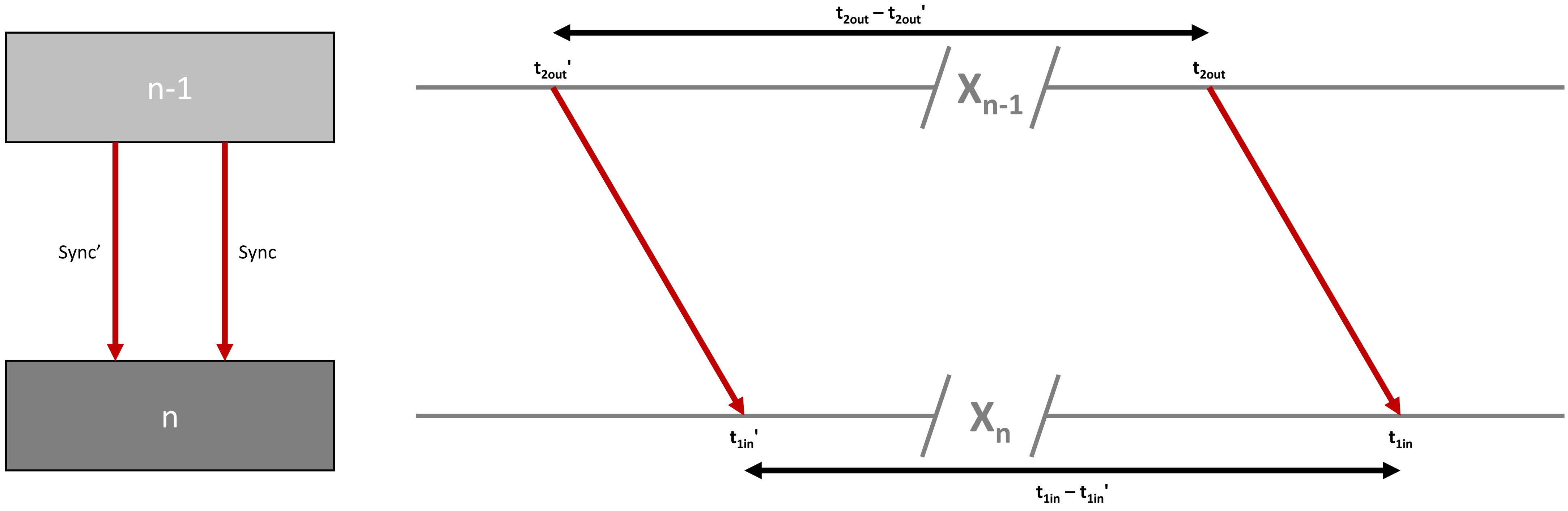
- Background
- Proposed Algorithm
- 1-hop Monte Carlo Simulation
- Headline Results from Simulation
- Additional Results
- Implications for the IEC/IEEE 60802 Profile

Background

Only Concerned with NRR



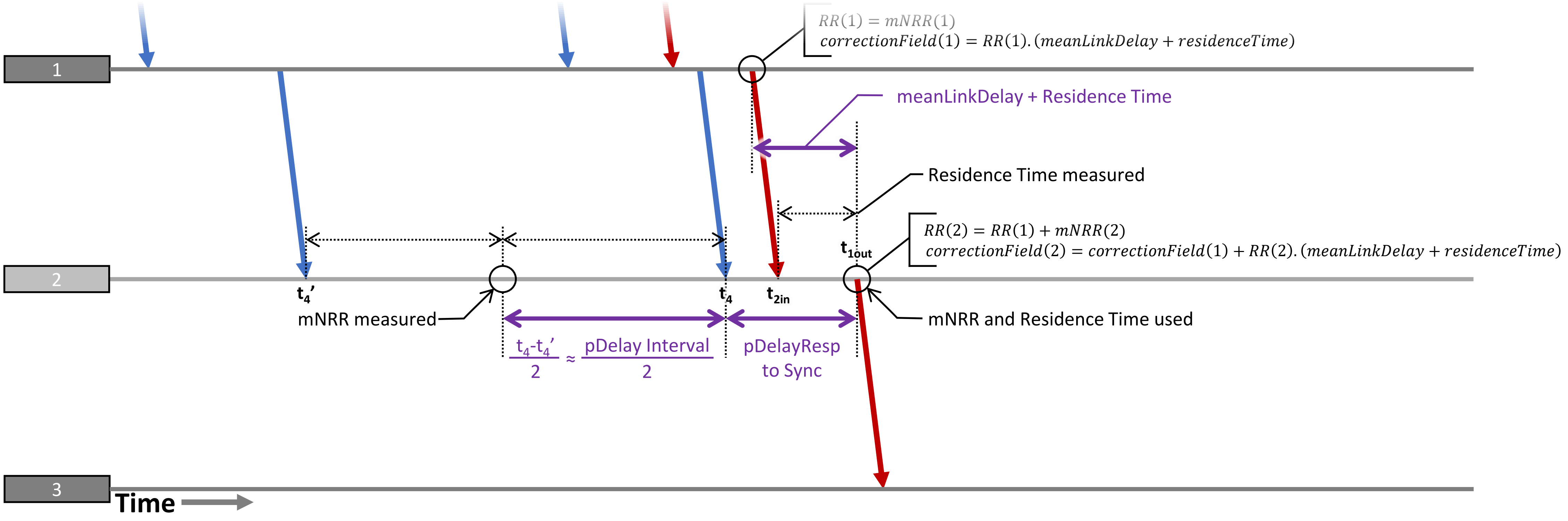
Using Sync to measure NRR…

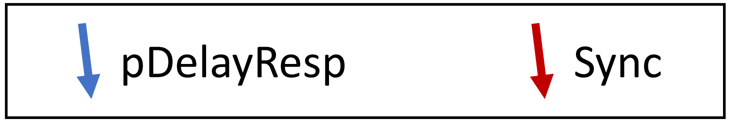


$$mNRR = \left(\frac{t_{2out} - t'_{2out}}{t_{1in} - t'_{1in}}\right)$$

ppm

…which improves timing & consistency.

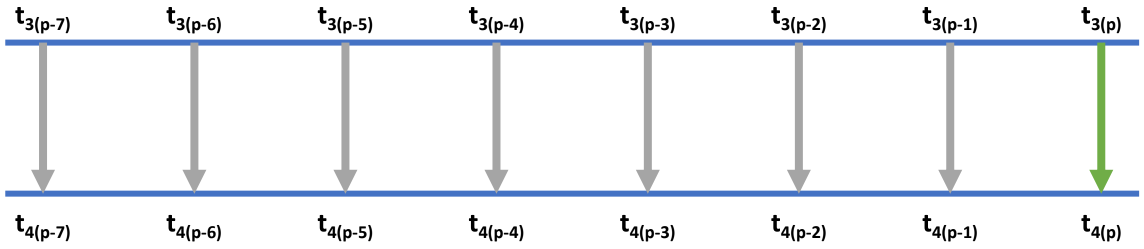




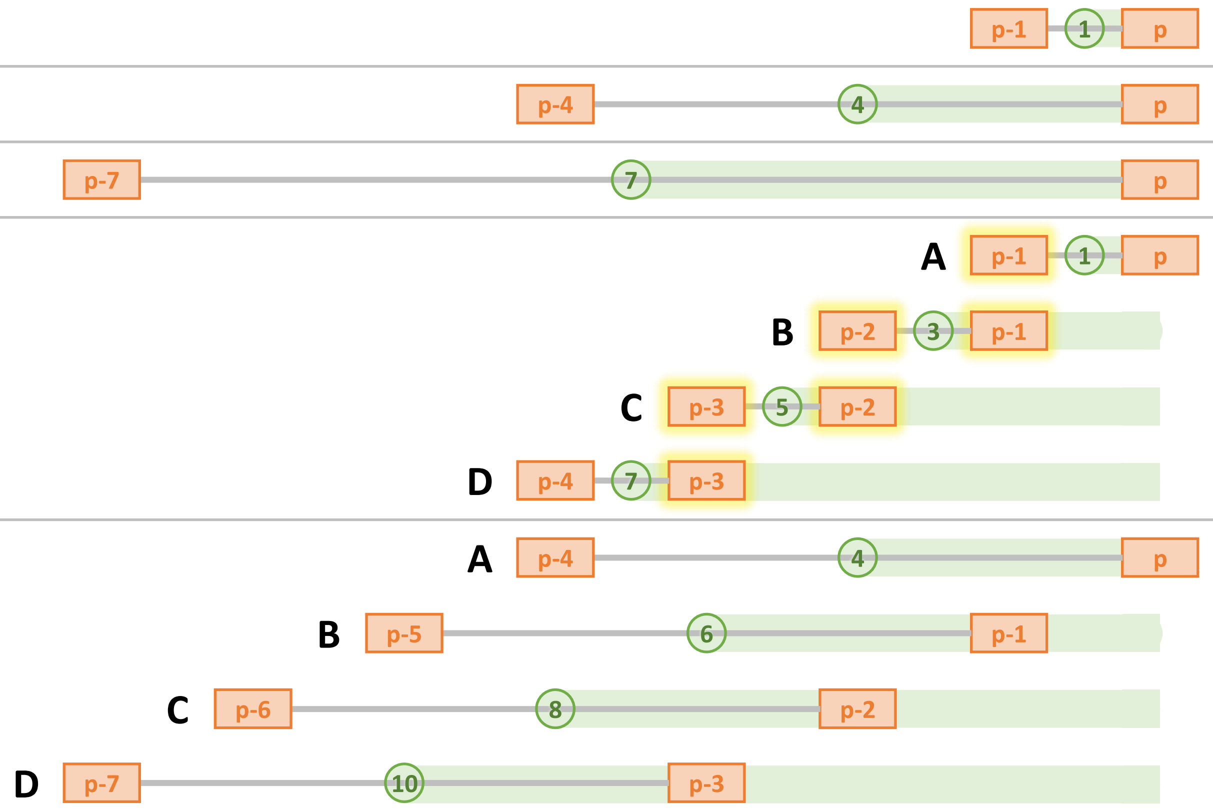
- Same errors in mNRR as 1st Hop.
  - Error due to drift during NRR measurement. (**Node 2** to **Node 1**)
  - Error due to drift between measuring and using NRR. (**Node 2** to **Node 1**)
  - Error due to drift during Residence Time measurement. (**Node 2** to **GM**)
- Additional error from drift 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.

Method 3 is best for using older timestamps

See [1]



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



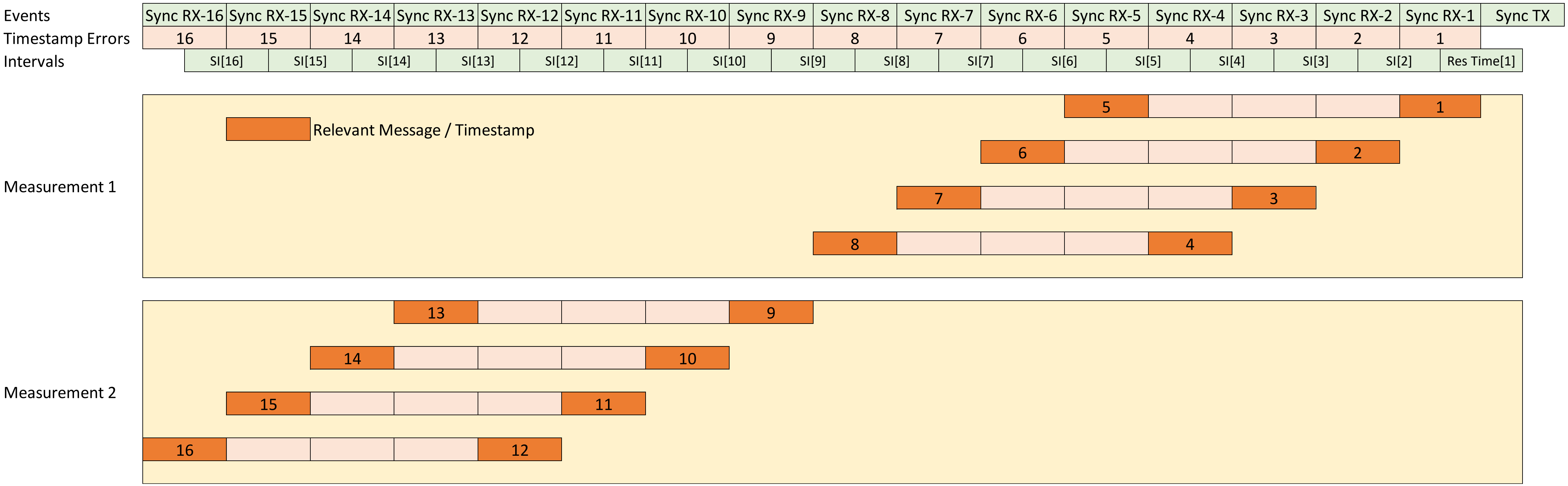
Proposed Algorithm

Approach Overview

- Assume linear Clock Drift
  - Good enough for short periods of time, unless…
  - …there is a sudden change in drift (discontinuity) when the assumption of linearity & consequent "compensation" could make things worse, but…
  - …even that's OK at a system level, provided this only happens at a low number of nodes at any one time (or a vast majority of nodes at exactly the same time).
- Take two measurements of NRR, separated in time
  - Construct measurements to limit impact of Timestamp error; only Clock Drift error remains.
  - Separate from measurement of NRR for use in Sync; different priorities for Clock Drift / Timestamp Error balance.
- Estimate NRR Drift
- Adjust t_{2out} timestamps (received from n-1 node), based on drift estimate, prior to mNRR calc for Sync
- Calculate mNRR for use in Sync and add to incoming RR field to calculate local RR.

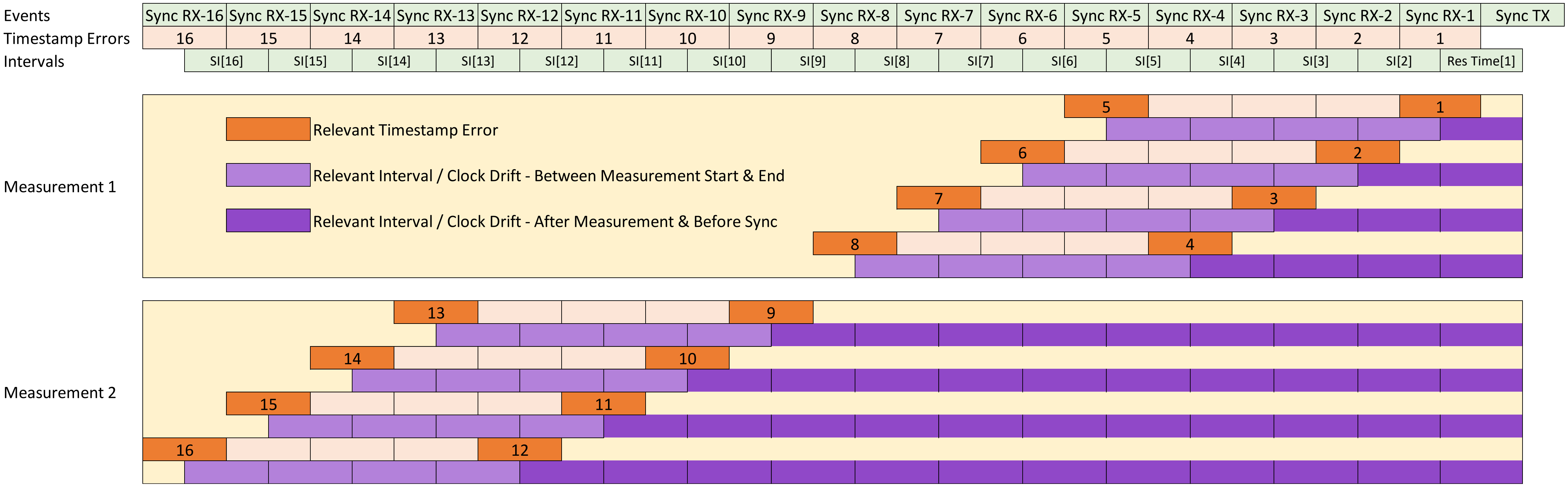
1-hop Monte Carlo Simulation

mNRR Comp Measurements – Example



- N = 4, each initial calculation goes back 4 timestamps
- A = 4, each measurement is an average of previous 4 initial calculations
- P = 8, second measurement starts 8 timestamps further in the past than the first measurements

mNRR Comp Measurements – Errors



- N = 4, each initial calculation goes back 4 timestamps
- A = 4, each measurement is an average of previous 4 initial calculations
- P = 8, second measurement starts 8 timestamps further in the past than the first measurements

2-hop Simulation - Overview

Events	Sync RX-16	Sync RX-15	Sync RX-14	Sync RX-13	Sync RX-12	Sync RX-11	Sync RX-10	Sync RX-9	Sync RX-8	Sync RX-7	Sync RX-6	Sync RX-5	Sync RX-4	Sync RX-3	Sync RX-2	Sync RX-1	Sync TX	
Timestamp Errors	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
Intervals	SI[16]	SI[15]	SI[14]	SI[13]	SI[12]	SI[11]	SI[10]	SI[9]	SI[8]	SI[7]	SI[6]	SI[5]	SI[4]	SI[3]	SI[2]	Res Time[1]		
Interval	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**SyncInterval**	**Res Time**		ms
TS	-(RT+SI[1:16])	-(RT+SI[1:14])	-(RT+SI[1:13])	-(RT+SI[1:12])	-(RT+SI[1:11])	-(RT+SI[1:10])	-(RT+SI[1:9])	-(RT+SI[1:8])	-(RT+SI[1:7])	-(RT+SI[1:6])	-(RT+SI[1:5])	-(RT+SI[1:4])	-(RT+SI[1:3])	-(RT+SI[1:2])	-(RT+SI[1])	-RT		ms
TempCycle	-(RT+SI[1:16])	-(RT+SI[1:14])	-(RT+SI[1:13])	-(RT+SI[1:12])	-(RT+SI[1:11])	-(RT+SI[1:10])	-(RT+SI[1:9])	-(RT+SI[1:8])	-(RT+SI[1:7])	-(RT+SI[1:6])	-(RT+SI[1:5])	-(RT+SI[1:4])	-(RT+SI[1:3])	-(RT+SI[1:2])	-(RT+SI[1])	-RT	**Random**	s
TempCycleN-1	-(RT+SI[1:16])	-(RT+SI[1:14])	-(RT+SI[1:13])	-(RT+SI[1:12])	-(RT+SI[1:11])	-(RT+SI[1:10])	-(RT+SI[1:9])	-(RT+SI[1:8])	-(RT+SI[1:7])	-(RT+SI[1:6])	-(RT+SI[1:5])	-(RT+SI[1:4])	-(RT+SI[1:3])	-(RT+SI[1:2])	-(RT+SI[1])	-RT	**Random**	s
TdriftN	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point		s
TdriftN-1	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point	Mid Point		s
DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN	DriftN		ppm/s
DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1	DriftN-1		ppm/s
NRRdrift (DriftN-1 - DriftN)	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift	NRRdrift		ppm/s
mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError	mNRRcdError		ppm
T2outTSerror	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**	**T2outTSerror**		ns
T1inTSerror	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**	**T1inTSerror**		ns
mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError	mNRRtsError		ns

mNRR Error Algorithms

mNRR Primary Errors

$$mNRR_{errorTS\_X} = \frac{(t_{3pderror} - t_{3pderror}') - (t_{4pderror} - t'_{4pderror})}{T_{pdelay2pdelay}} = \frac{(t_{3pderror} - t_{4pderror}) - (t'_{3pderror} - t'_{4pderror})}{T_{pdelay2pdelay}}$$ **ppm**

$$mNRR_{errorCD\_X} = \frac{T_{pdelay2pdelay}(clockDrift_n - clockDrift_{n-1})}{2 \times 10^3}$$ **ppm**

$$mNRR_{error\_X} = mNRR_{errorTS\_X} + mNRR_{errorCD\_X}$$ **ppm**

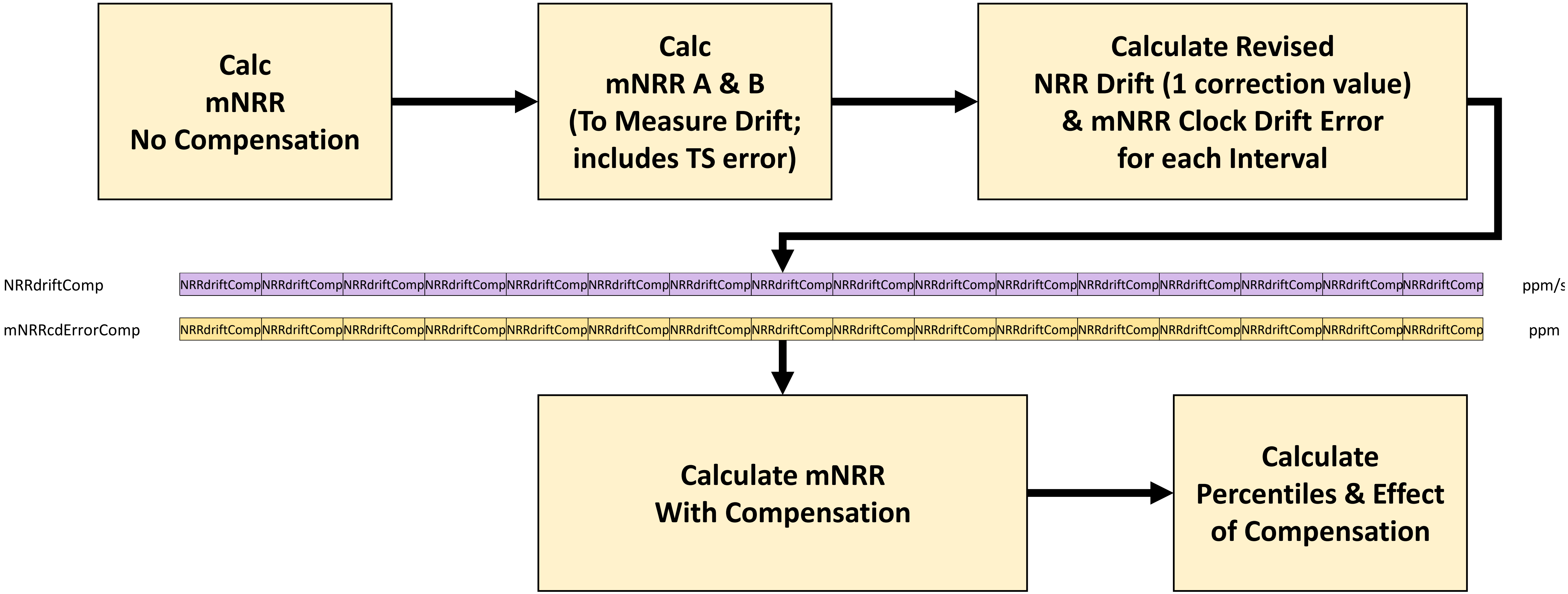
RR Primary Errors

$$RR_{errrorCD\_NRR2Sync\_X} = \frac{T_{mNRR2Sync}(clockDrift_n - clockDrift_{n-1})}{10^3}$$ **ppm**

$$RR_{errrorCD\_RR2Sync\_X} = \frac{residenceTime(clockDrift_{n-1} - clockDrift_{GM})}{10^3}$$ **ppm**

$$RR_{errror\_SUM}(n) = RR_{error\_SUM}(n-1) + mNRR_{error\_X} + RR_{errorCD\_NRRtoSync\_X} + RR_{errorCD\_RRtoSync\_X}$$ **ppm**

2-hop Simulation Overview

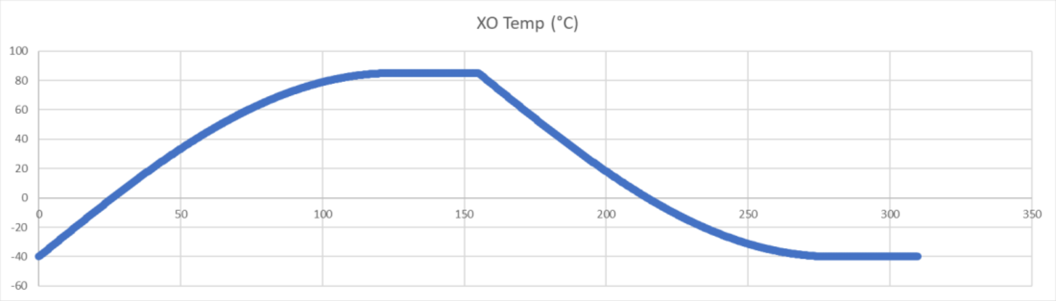


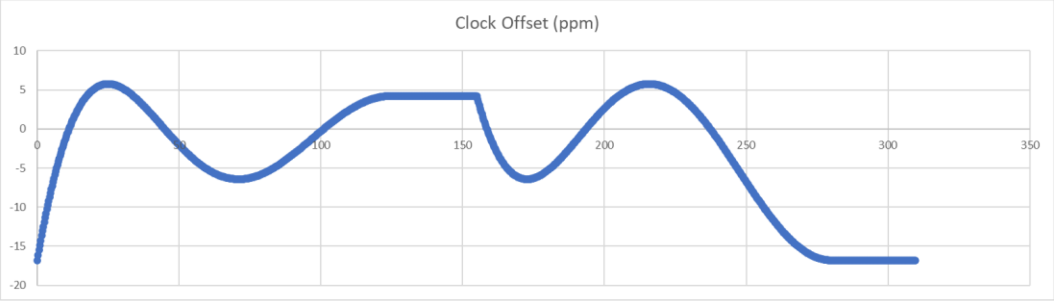
Headline Results

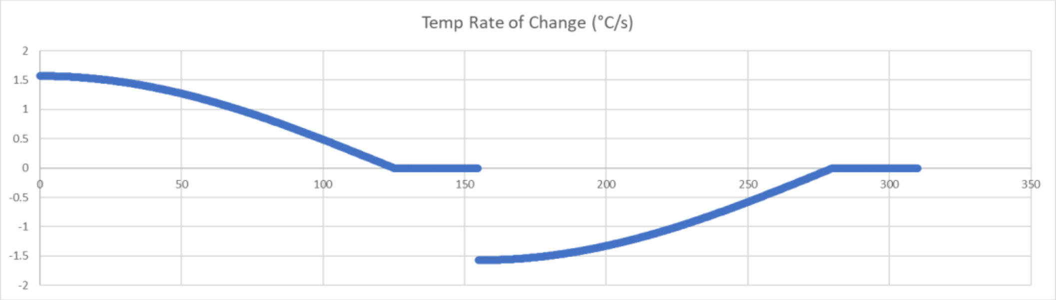
Configuration

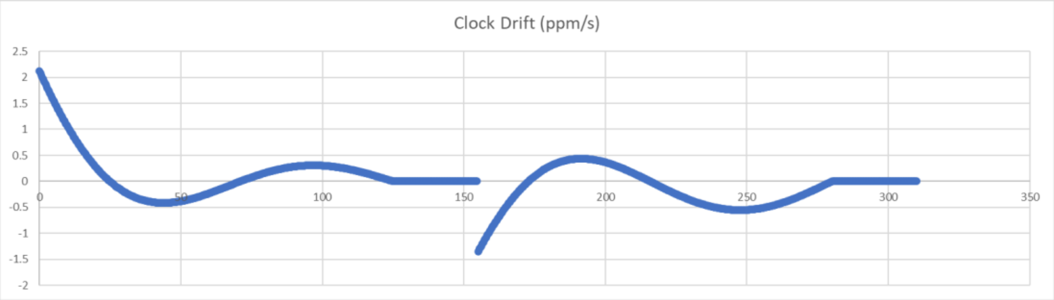
- XO Temp Dift: Quarter Sinusoidal, 125s ramp; 30s hold
- 8ns Timestamp Granularity; ±4ns Dynamic Timestamp Error
- 125s Sync Interval; uniform distribution 120-130ms
- Residence Time: 10ms; mean 5ms; standard deviation 1.8ms; max 15ms
- mNRRsmoothingN: 4
- mNRRsmoothingA: 4

Clock Drift – ¼-Sinusoidal Temperature Ramp: 125s ↕







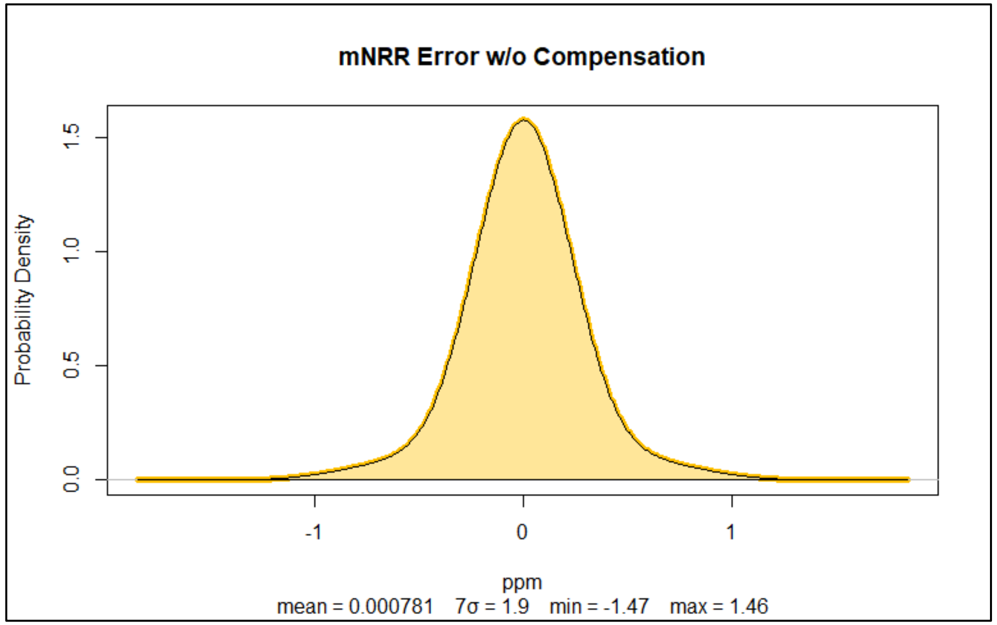


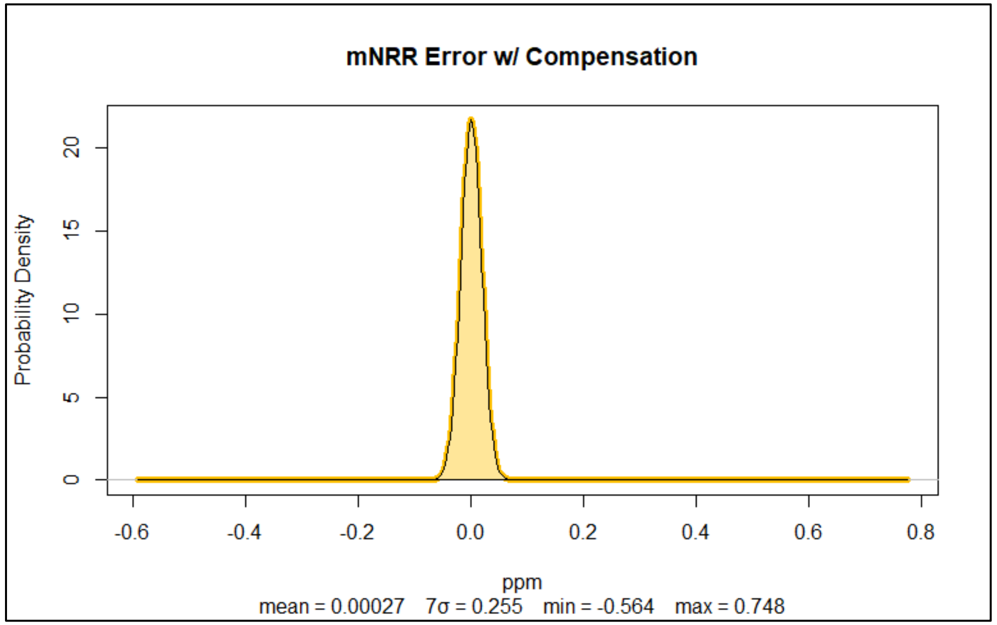
Inputs		
Temp Max	85	°C
Temp Min	-40	°C
Temp Ramp Period	125	s
Temp Hold	30	s

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

Clock Drift		
MAX	2.12	ppm/s
MIN	-1.35	ppm/s

Best Result (so far…)





mNRRcompN = 3
mNRRcompA = 3
mNRRcompP = 6

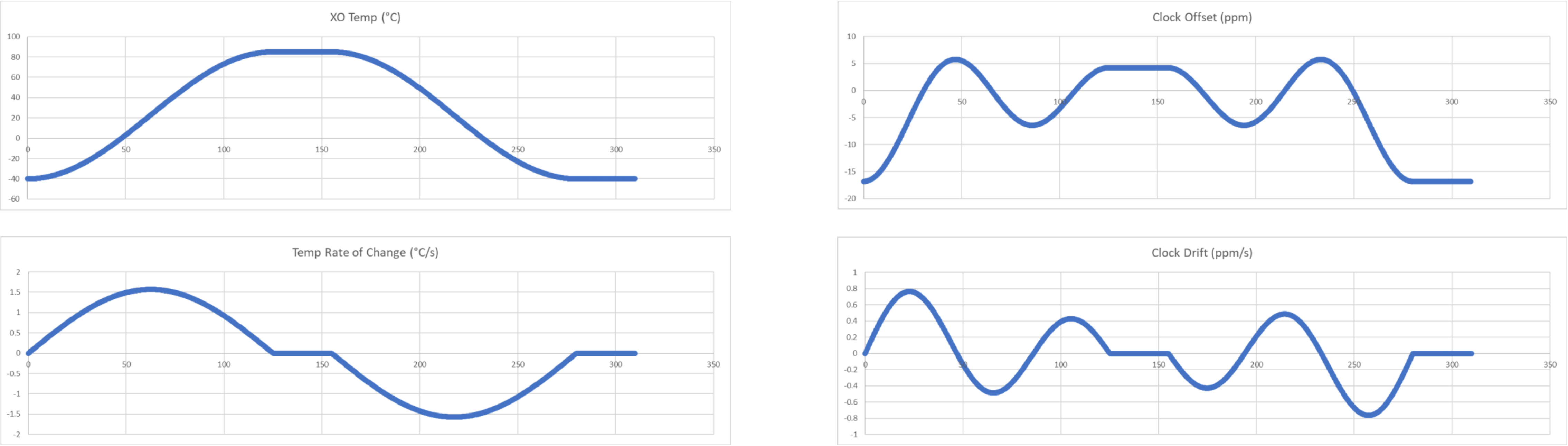
mNRR (for Sync)		mNRR for Compensation			Spreads					
N	A	N	A	P	7σ	Min-Max	99%	98%	95%	90%
4	4				3.81	2.94	1.8	1.8	1.21	0.836
4	4	3	3	6	0.51	1.31	0.311	0.0876	0.0666	0.0541
				Percentage Change	-87%	-55%	-83%	-95%	-94%	-94%

Additional Results

Varying N/A/P for Compensation Calculation

mNRR (for Sync)		mNRR for Compensation			Spreads						Percentage Improvement					
N	**A**	**N**	**A**	**P**	**7σ**	**Min-Max**	**99%**	**98%**	**95%**	**90%**	**7σ**	**Min-Max**	**99%**	**98%**	**95%**	**90%**
4	4				3.81	2.94	1.8	1.8	1.21	0.836						
4	4	1	1	2	1.85	1.48	0.677	0.612	0.516	0.433	-51%	-50%	-62%	-66%	-57%	-48%
4	4	2	2	4	0.393	0.763	0.14	0.12	0.0979	0.0814	-90%	-74%	-92%	-93%	-92%	-90%
4	4	3	3	6	0.51	1.31	0.311	0.0876	0.0666	0.0541	-87%	-55%	-83%	-95%	-94%	-94%
4	4	4	4	8	0.705	1.91	0.554	0.111	0.075	0.0588	-81%	-35%	-69%	-94%	-94%	-93%
4	4	5	5	10	0.876	2.2	0.762	0.227	0.096	0.075	-77%	-25%	-58%	-87%	-92%	-91%
4	4	6	6	12	1.03	2.36	0.866	0.423	0.121	0.0945	-73%	-20%	-52%	-77%	-90%	-89%
4	4	7	7	14	1.16	2.57	0.95	0.554	0.147	0.115	-70%	-13%	-47%	-69%	-88%	-86%
4	4	8	8	16	1.28	2.62	1.02	0.7	0.175	0.135	-66%	-11%	-43%	-61%	-86%	-84%
4	4	9	9	18	1.39	2.69	1.08	0.784	0.205	0.156	-64%	-9%	-40%	-56%	-83%	-81%
4	4	10	10	20	1.48	2.75	1.15	0.848	0.236	0.177	-61%	-6%	-36%	-53%	-80%	-79%

Clock Drift – ½-Sinusoidal Temperature Ramp: 125s ↕



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

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

Clock Drift		
MAX	0.76	ppm/s
MIN	-0.76	ppm/s

¼-Sinusoidal vs ½-Sinusoidal

mNRR (for Sync)		mNRR for Compensation			Ramp	Spreads					
N	A	N	A	P		7σ	Min-Max	99%	98%	95%	90%
4	4				½ SIN	3.27	1.39	1.12	1.12	0.919	0.764
4	4	3	3	6	½ SIN	0.208	0.135	0.076	0.0687	0.0581	0.0489
Percentage Change						-94%	-90%	-93%	-94%	-94%	-94%

mNRR (for Sync)		mNRR for Compensation			Ramp	Spreads					
N	A	N	A	P		7σ	Min-Max	99%	98%	95%	90%
4	4	3	3	6	¼ SIN	0.51	1.31	0.311	0.0876	0.0666	0.0541
4	4	3	3	6	½ SIN	0.208	0.135	0.076	0.0687	0.0581	0.0489
Percentage Change						-59%	-90%	-76%	-22%	-13%	-10%

- ½-Sinusoidal **slightly** reduces error before compensation, but…
- Is **much** easier to track than ¼-Sinusoidal (no discontinuities to throw off the algorithm)

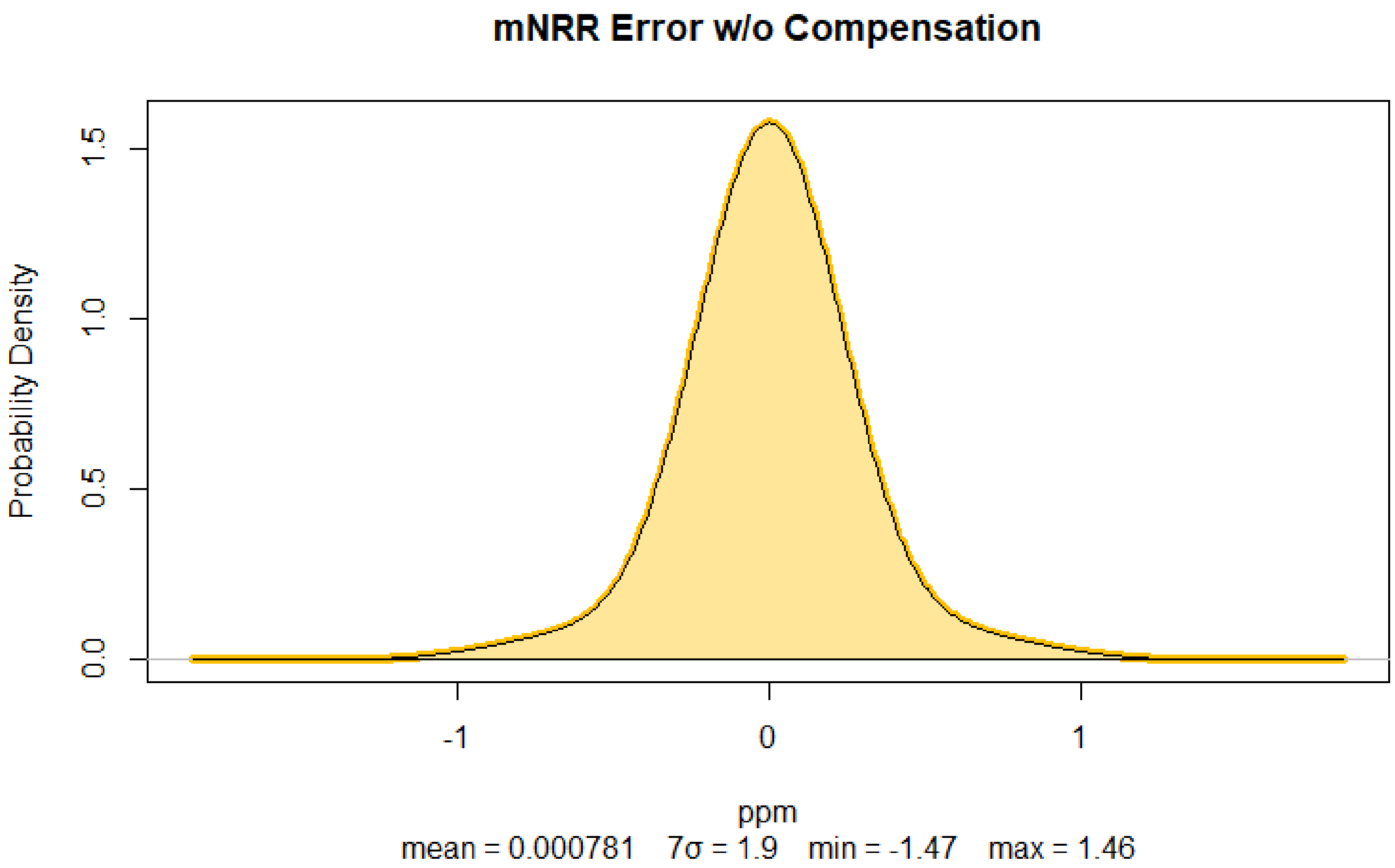
Implications

Implications for 60802

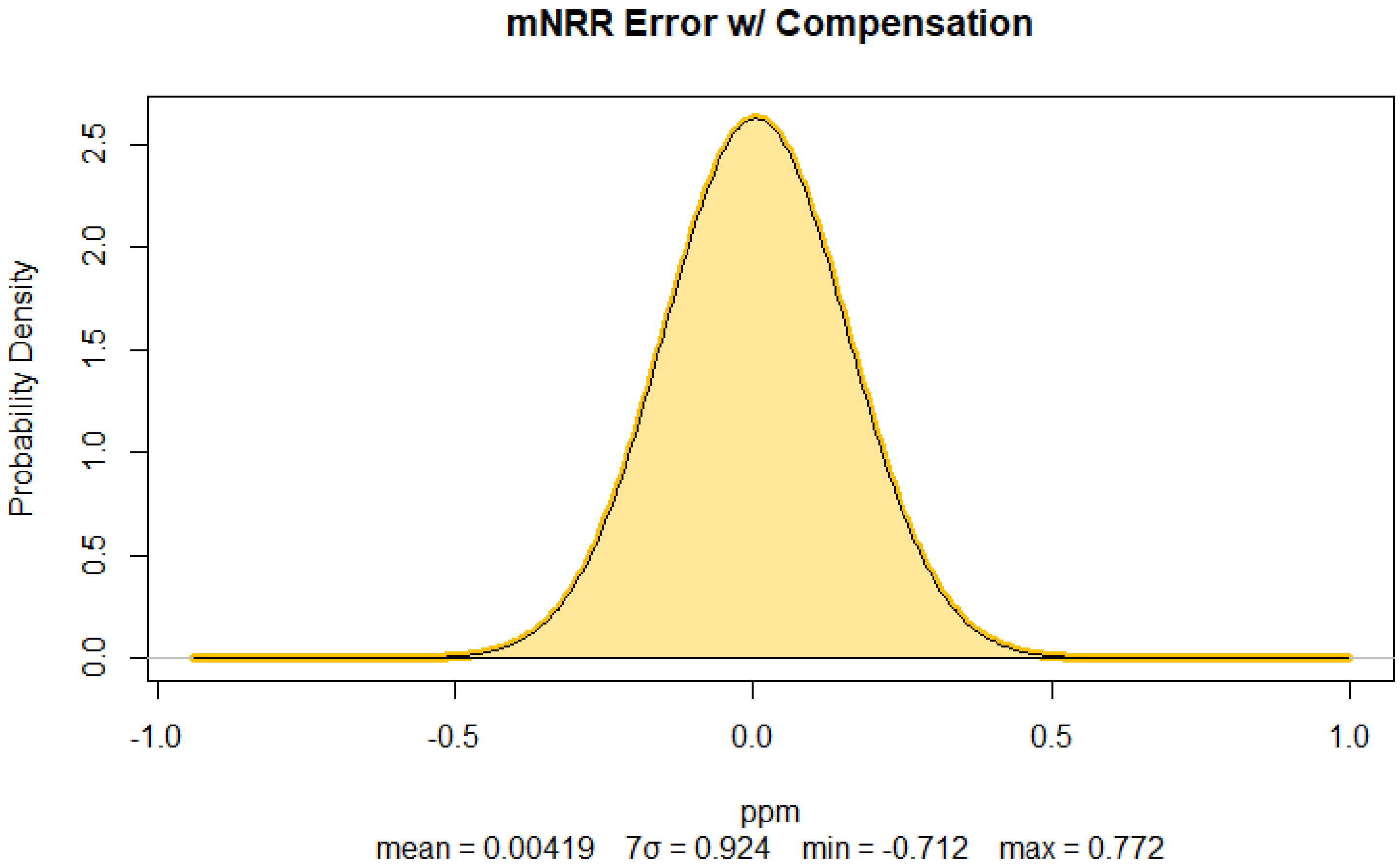
- Algorithm (mostly) works well (according to this simulation)
- As expected, it doesn't work as well when there are sudden changes in NRR Drift
  - Effect can be limited via careful choice of N/A/P parameters
  - More complex algorithms (e.g. Kalman Filter) would help, but more complex to implement (and compensation needs to be finalised between Sync TX & Follow-Up TX…although tracking can be done after Sync RX)
- May need to specify not just steady-state performance but response time to change? ("Best" parameters imply 1.5s response time.)
- Not entirely clear how this will play out at a system level
  - Simple assumption of % effective may not be accurate…but is it too kind, or too harsh? 🤔
  - Could potentially analyse NRR drift vs % effective to keep 100-hop simulation executing quickly…or just suck it up and let simulator run for a few hours (or longer?)

Backup

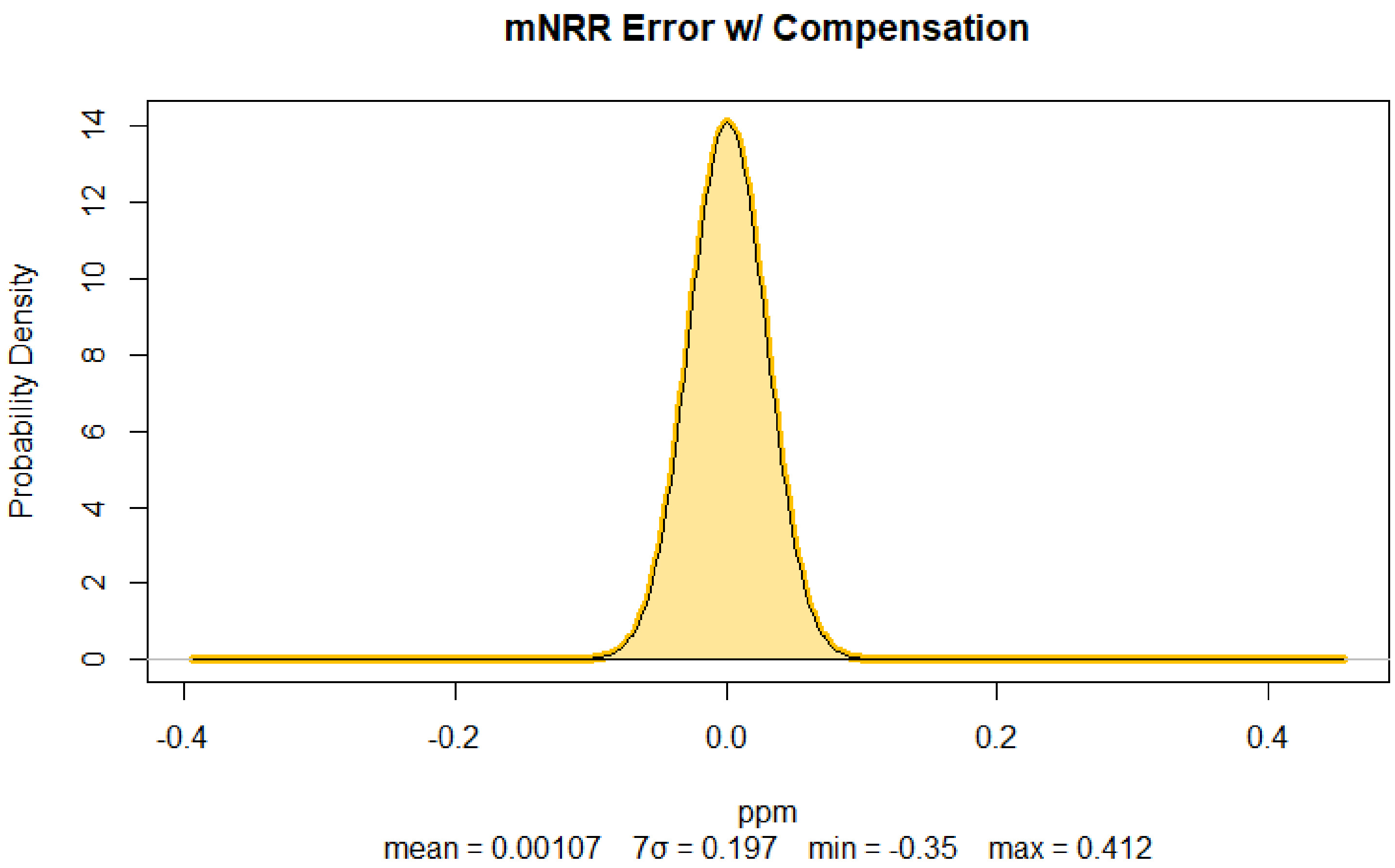
Base mNRR w/o Comp



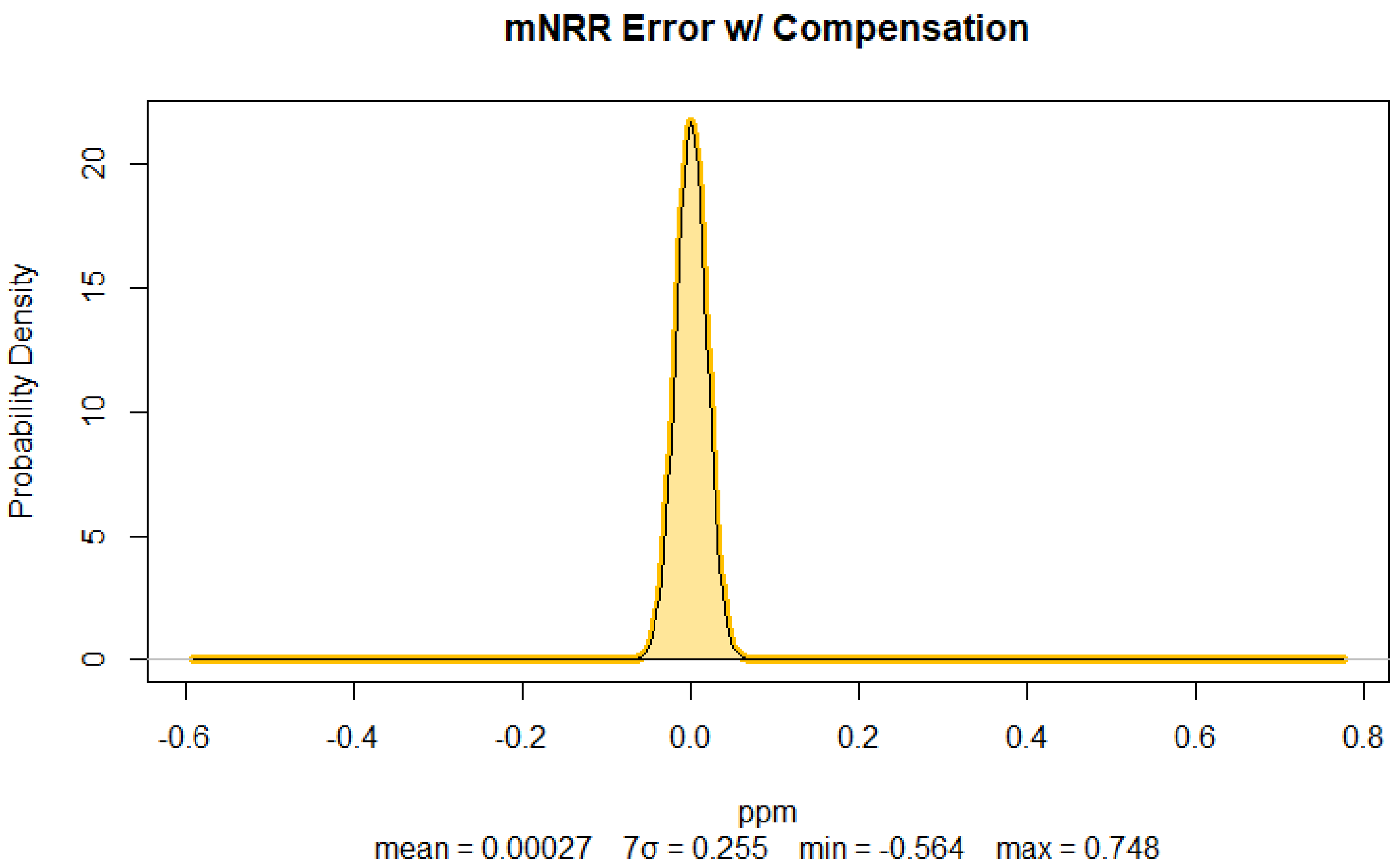
mNRR w/ Comp – N:1 A:1 P:2



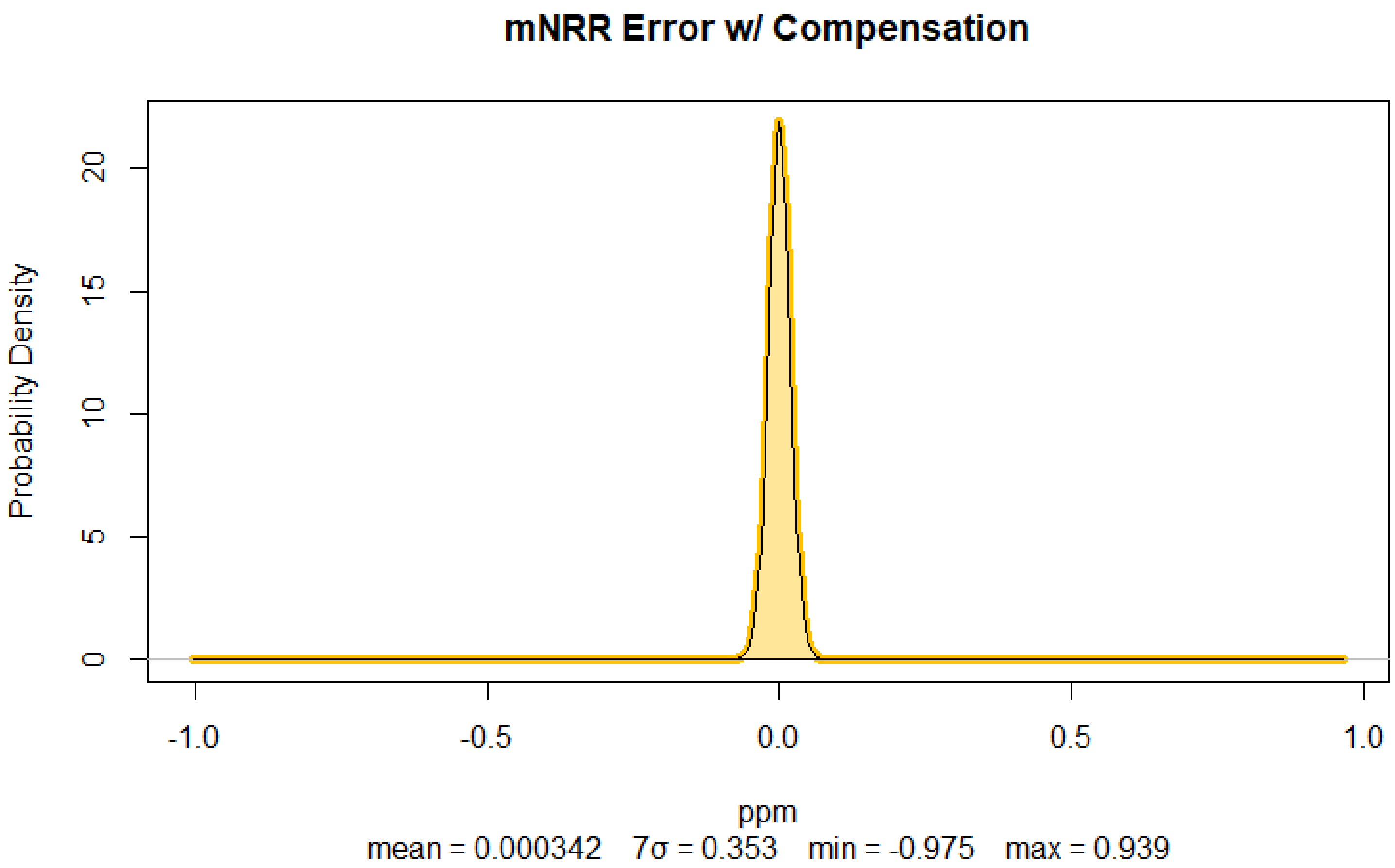
mNRR w/ Comp – N:2 A:2 P:4



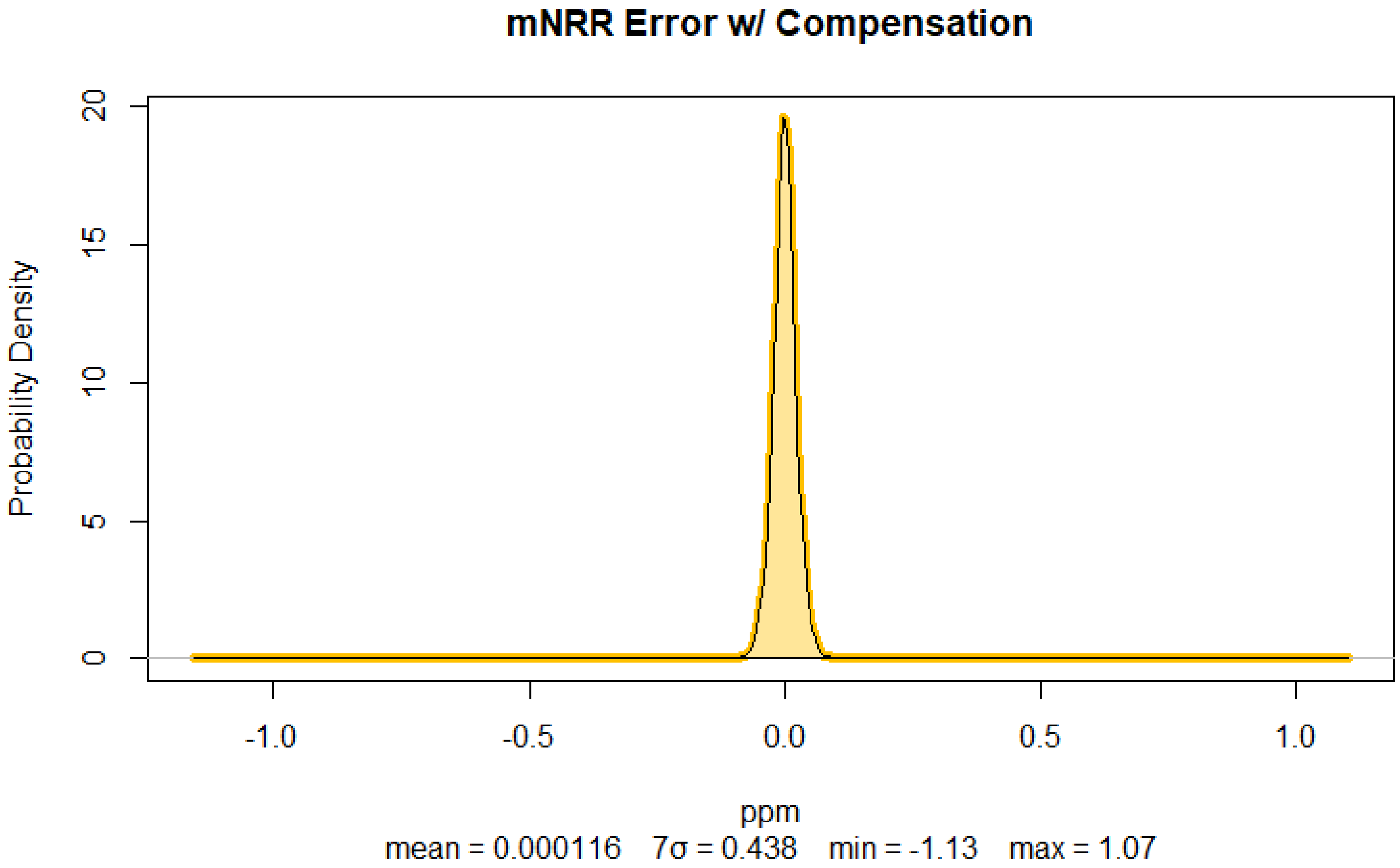
mNRR w/ Comp – N:3 A:3 P:6



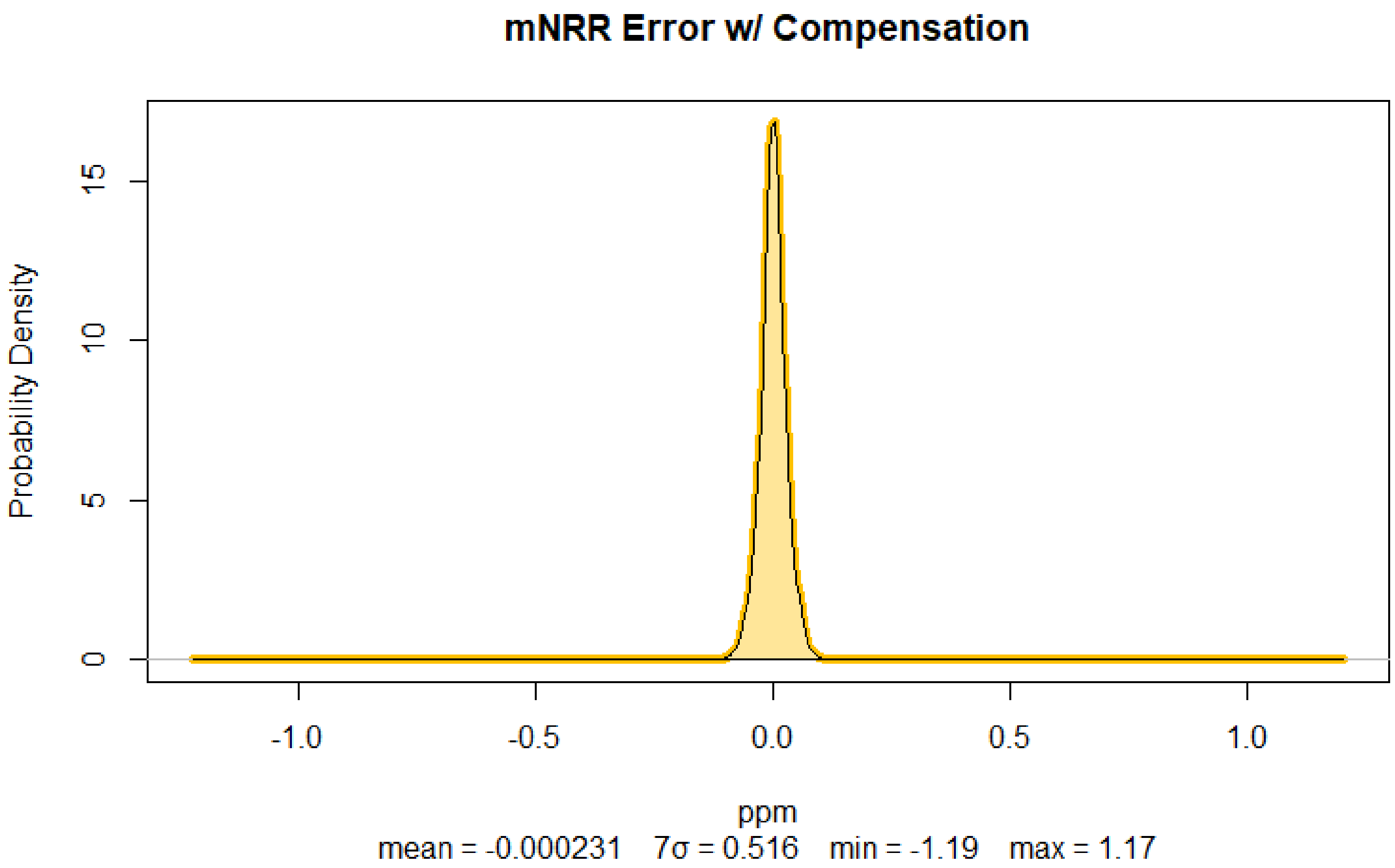
mNRR w/ Comp – N:4 A:4 P:8



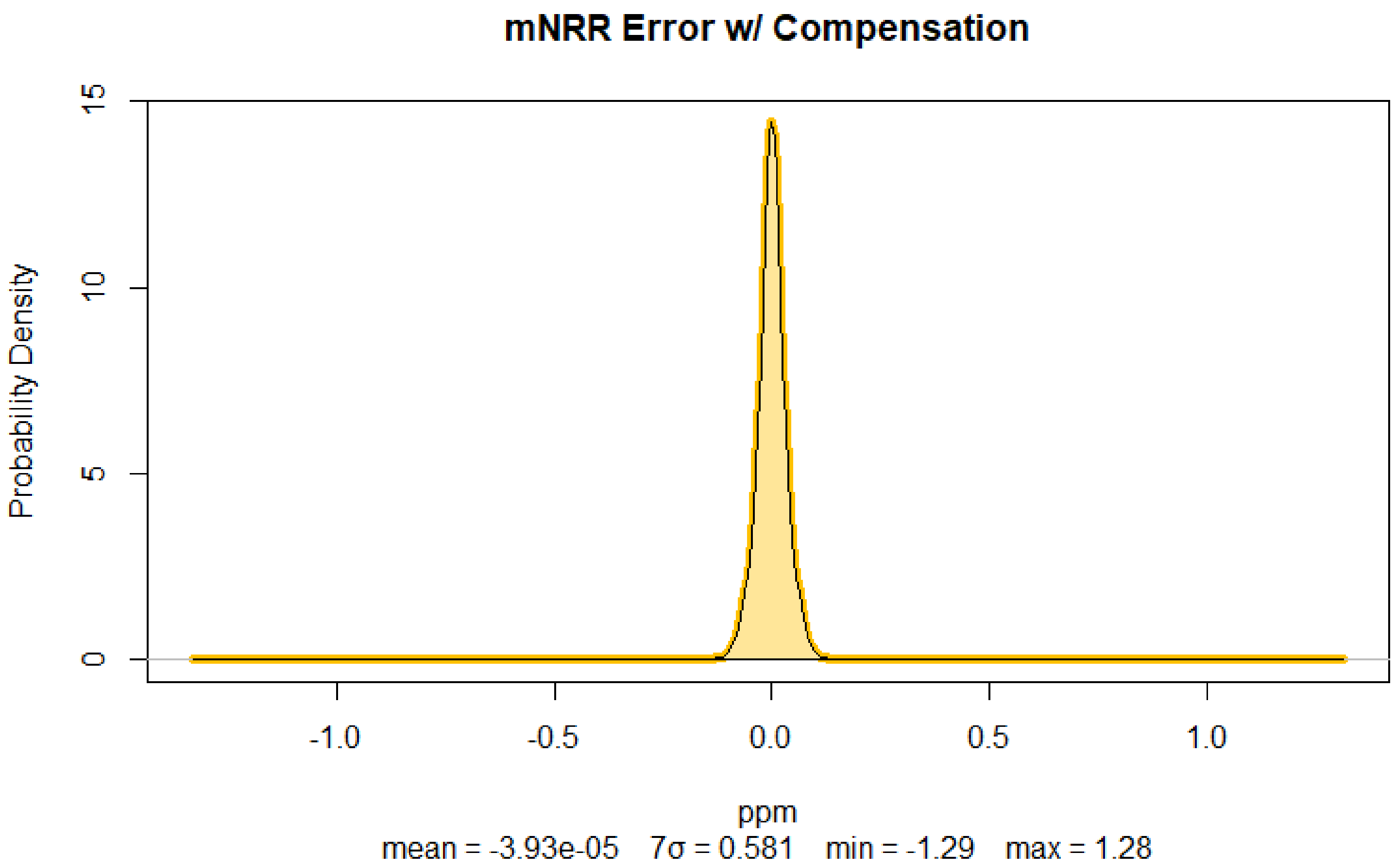
mNRR w/ Comp – N:5 A:5 P:10



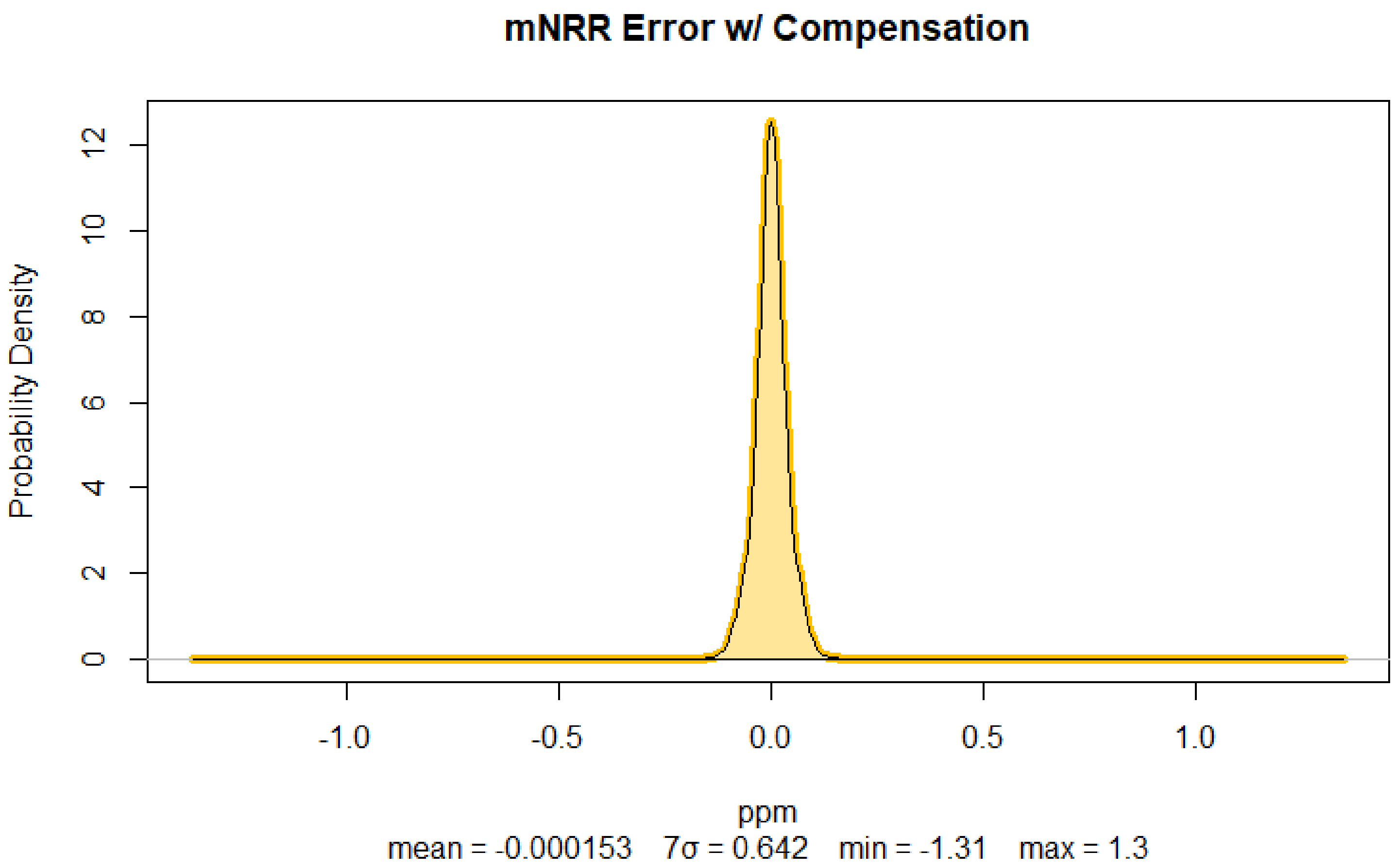
mNRR w/ Comp – N:6 A:6 P:12



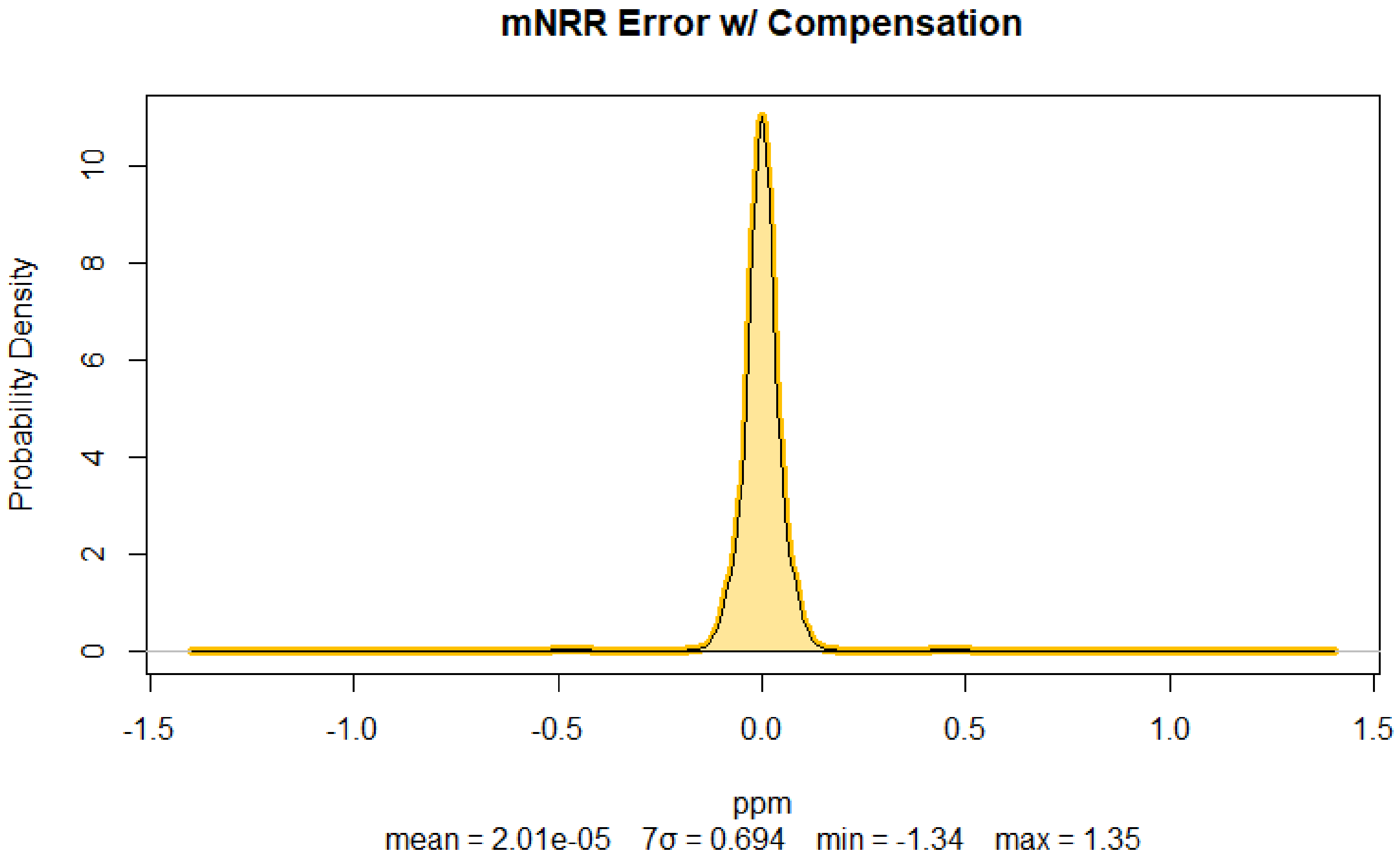
mNRR w/ Comp – N:7 A:7 P:14



mNRR w/ Comp – N:8 A:8 P:16



mNRR w/ Comp – N:9 A:9 P:18



mNRR w/ Comp – N:10 A:10 P:20