60802 Time Sync – Benefits of All PTP Relays using Same NRR Tracking Algorithm

David McCall – Intel Corporation

Version 1

Content

- Clock Drift Errors at Relays – Node-to-Node Cancellation Effect
- 60802 Measures to Enhance Node-to-Node Cancellation Effect
- 60802 – NRR Drift Tracking & Error Correction
- Normative Requirements for NRR Drift Tracking & Error Correction at PTP Relays
- Using the Same NRR Tracking Algorithm at all PTP Relays
  - Implications for Simulations & Performance in the Field
- Next Steps & Recommendations

References

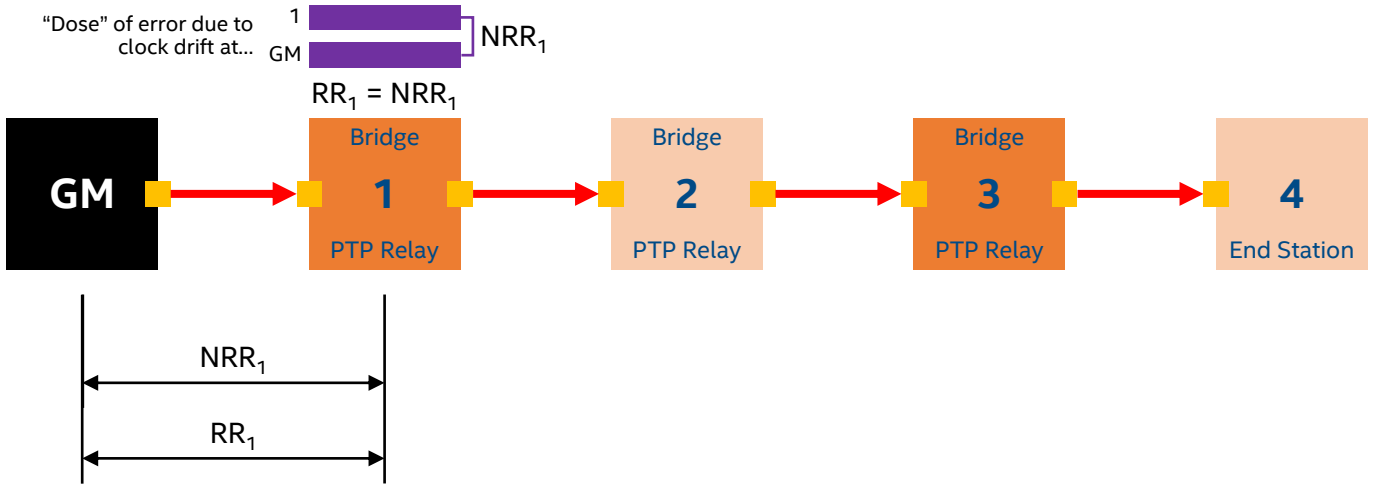
[1] “60802 Time Sync Ad Hoc mNRRsmoothing Optimisation Results”, David McCall, v2, 60802 contribution, 4th November 2022,

[2] “60802 Time Sync – Normative Requirements & Testing”, David McCall, v5, 60802 contribution, 28th April 2023

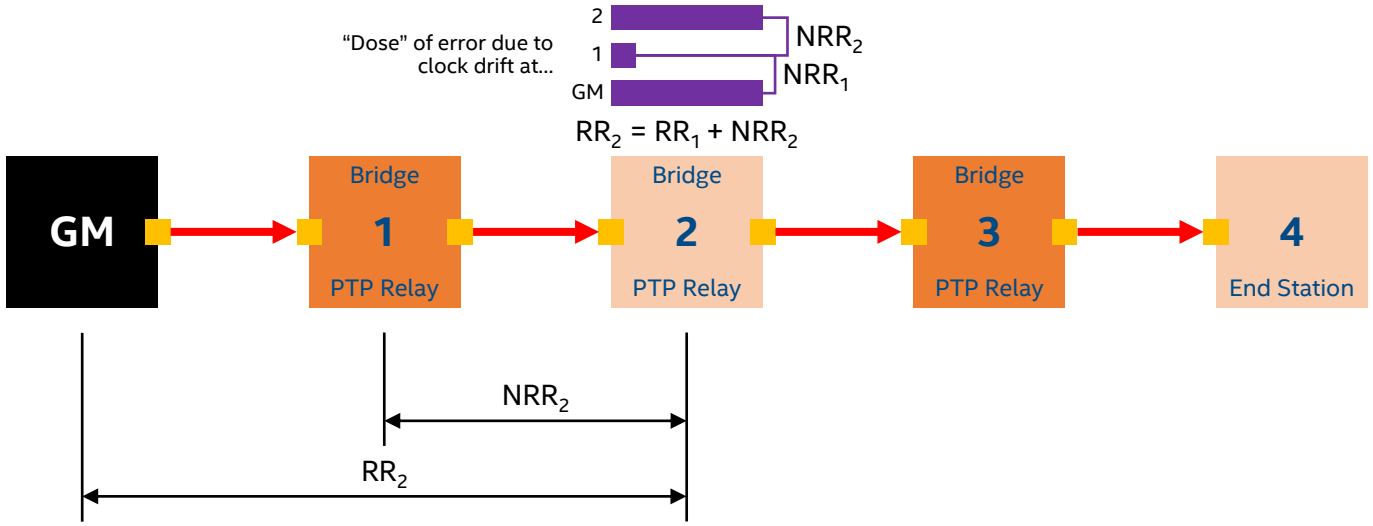
Clock Drift Error at PTP Relays – Node-to-Node Cancellation Effect

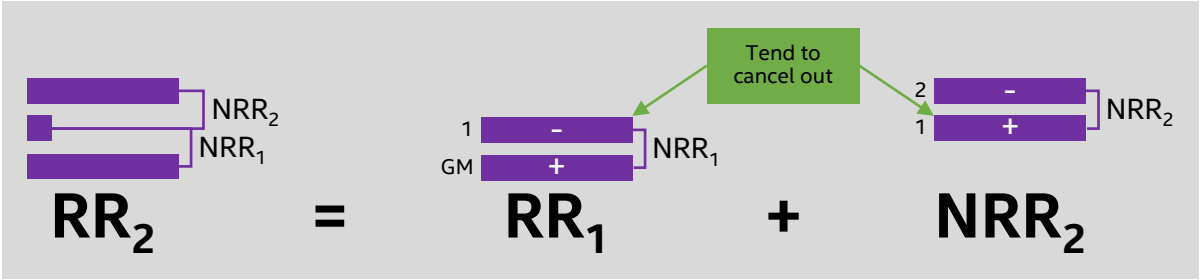
A simple model…

Components of RR_{error} at Node 1

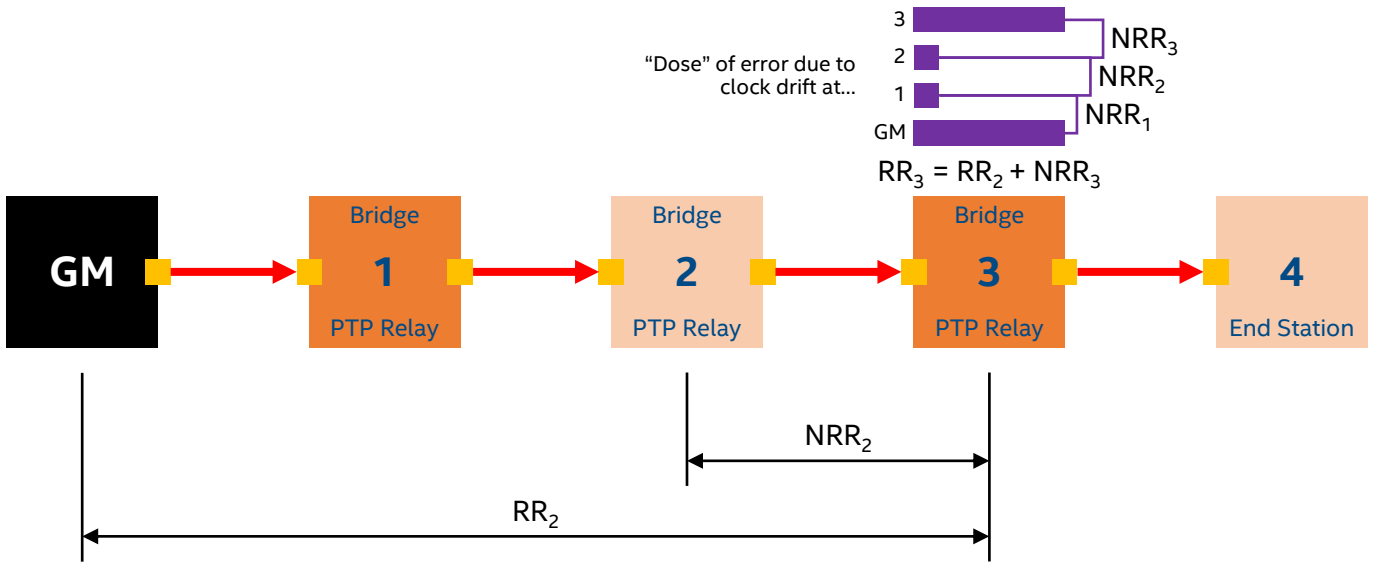


Components of RR_{error} at Node 2

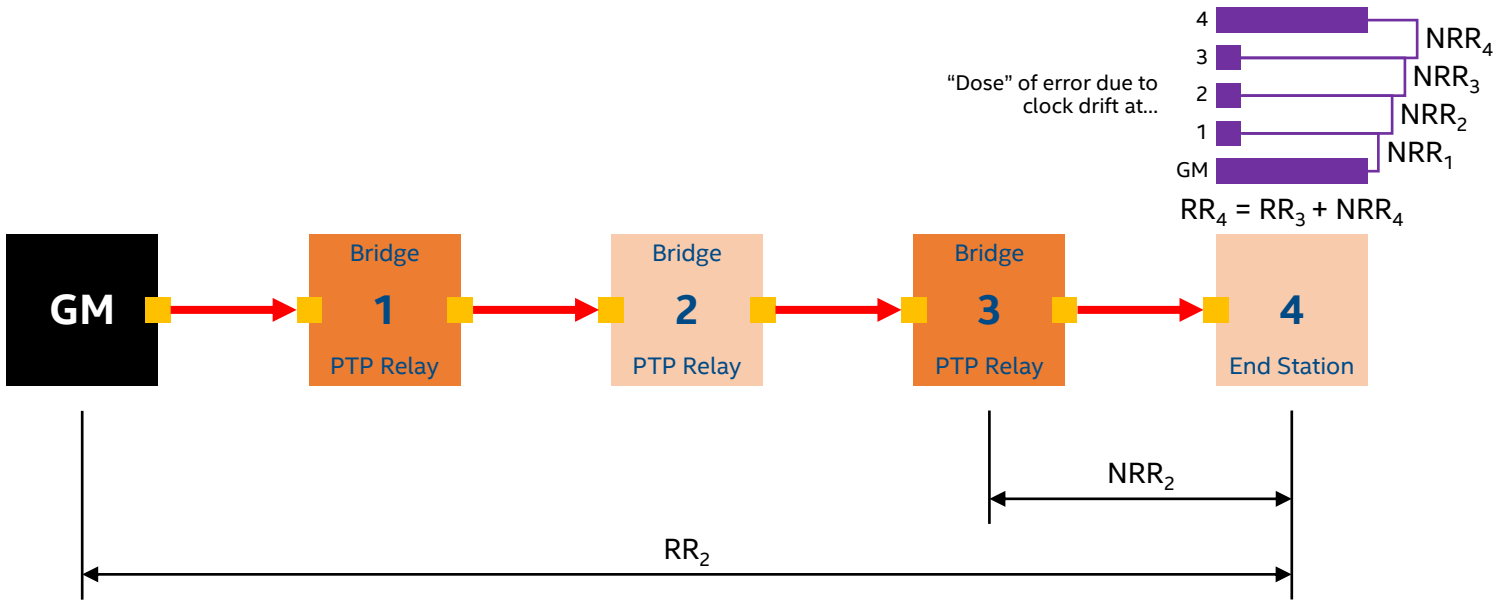




Components of RR_{error} at Node 3



Components of RR_{error} at Node 4 [End Station]



Simple Model vs. Reality

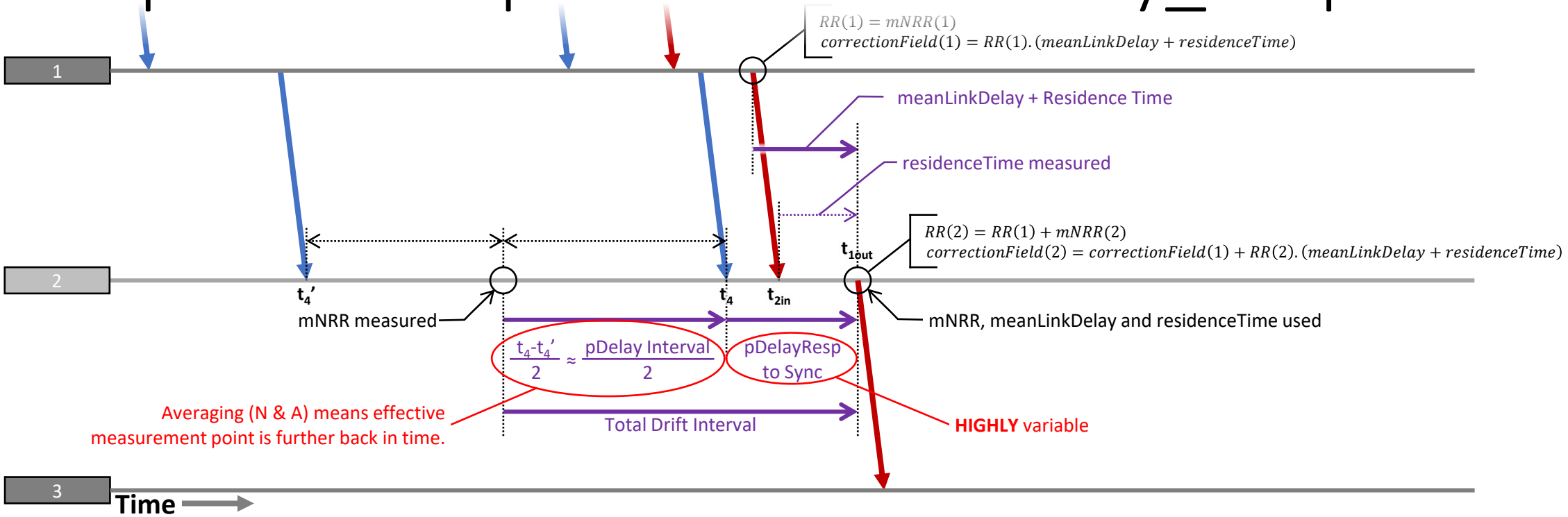
- The model ignores timestamp errors and does not include NRR drift tracking and error compensation.
- The actual amount of error incurred by each Sync message depends on amount of clock drift – the model is best thought of as comparing a “full dose” vs. a “reduced dose” due to node-to-node cancellation.
  - Over long periods of time (or large numbers of simulation runs) the principle applies to the average amount of NRR error that survives in RR.
- **The degree of node-to-node cancellation depends a lot on the consistency of the related intervals.**
  - Higher consistency node-to-node → greater cancellation node-to-node
- The amount of dTE due to clock drift at the GM is usually greater than the amount of dTE due to the same amount of clock drift at the End Station
  - The amount of RR error is the same, but RR error at the GM survives and affects every node in the chain, **propotional to total Residence Time**. RR error due to clock drift at the End Station only affects the End Station and is **proportional to Sync Interval**.

60802 Measures to Enhance Node-to-Node Cancellation Effect

60802 Measures to Enhance Node-to-Node Cancellation Effect

- Improve consistency / eliminate variability of relevant intervals…

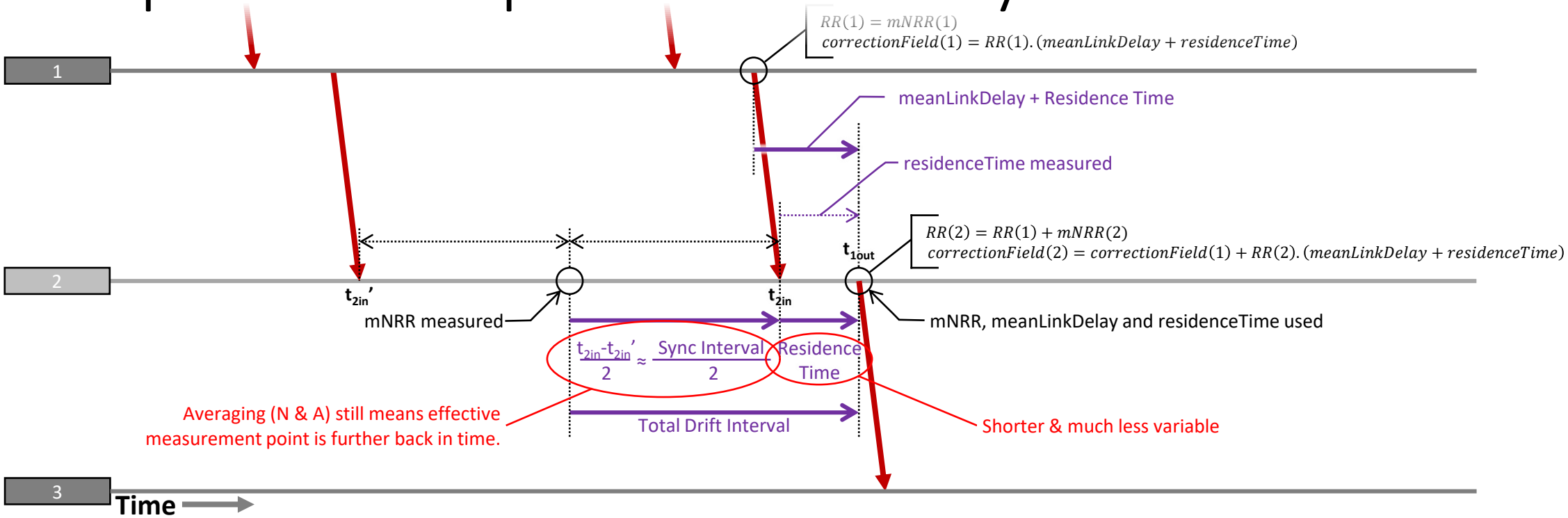
Clock Drift Error – Relevant Intervals
4 Hops – 2nd Hop – NRR from Pdelay_Resp



pDelayResp Sync

- 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**)
- Error due to 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.

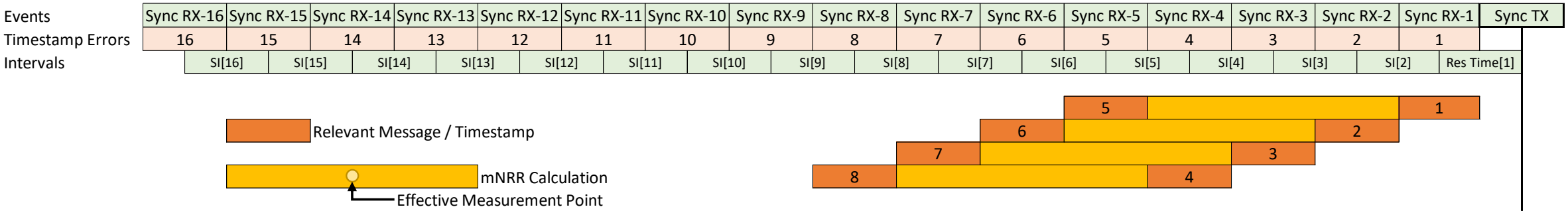
Clock Drift Error – Relevant Intervals
4 Hops – 2nd Hop – NRR from Sync



Sync

- 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**)
- Error due to 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.

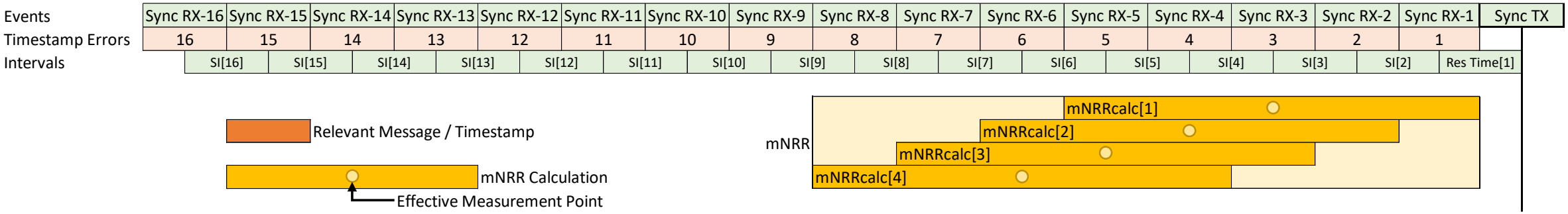
mNRR Averaging



- Instead of one calculation using timestamp information from the two most recent Sync messages (s & s-1)…
- First calculation uses most recent and 4th-back Sync messages (s & s-4)
- Three more calculations: (s-1 & s-5), (s-2 & s-6), (s-3 & s-7)

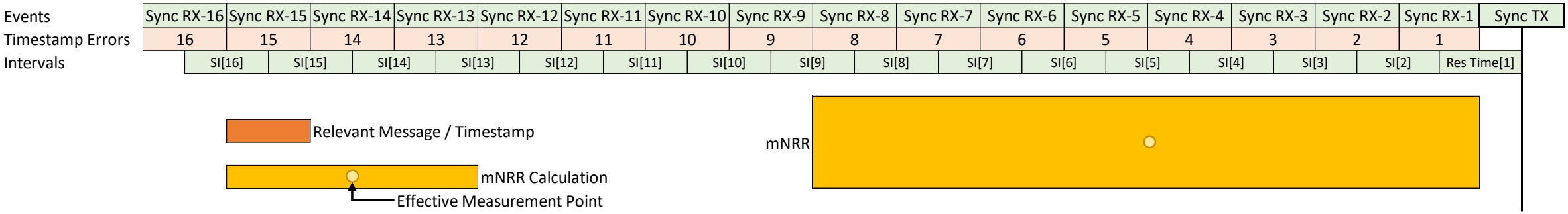
Note: Figures don't use same s, s-1, etc… notation as they were prepared to discuss simulation implementation using vectors & arrays.

mNRR Averaging



- The effective measurement point for each value is the mid-point between the two Sync messages used.
  - Referenced to the Sync ingress timestamp at the current node.

mNRR Averaging



- Take an average of the four values to measure NRR (mNRR)
- Effective measurement point is an average of the 4 calculations effective measurement points
- This calculation method reduced timestamp error more than other approaches that use the same date in different ways. See [1] for more detail.

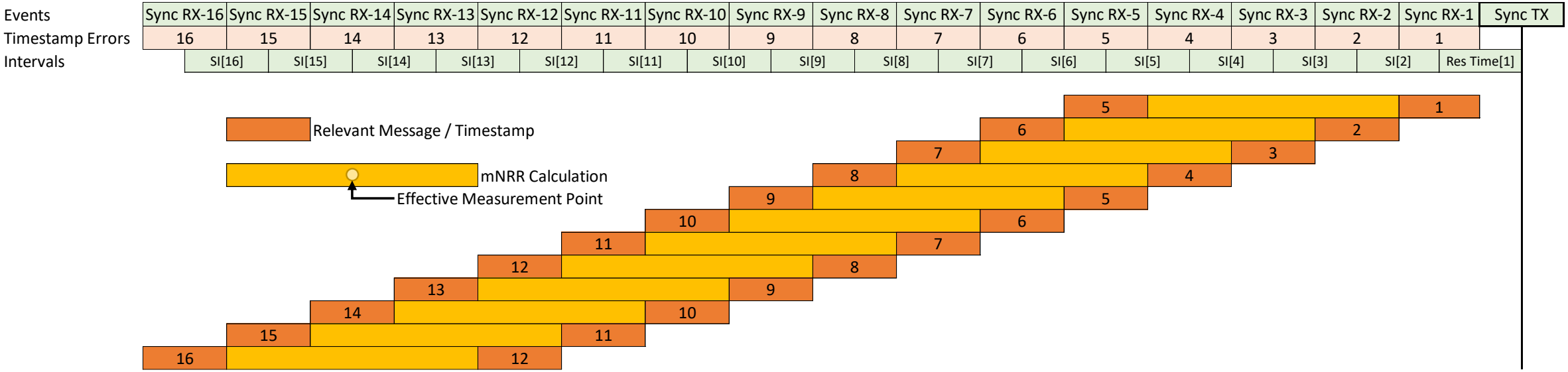
60802 Measures to Enhance Node-to-Node Cancellation Effect

- Improve consistency / eliminate variability of relevant intervals…
  - New TLV to allow NRR measurement via Sync message (vs. Pdelay_Resp)
  - Limits on residenceTime variability
    - Maximum Mean: 5 ms; Standard Deviation: 1.8 ms; Maximum: 15 ms
  - Limits on Sync Interval variability (at GM)
    - Nominal: 125ms; Minimum: 119ms; Maximum: 131ms

- Averaging (N & A) decreases timestamp error but increases errors due to clock drift. The latter are addressed by…

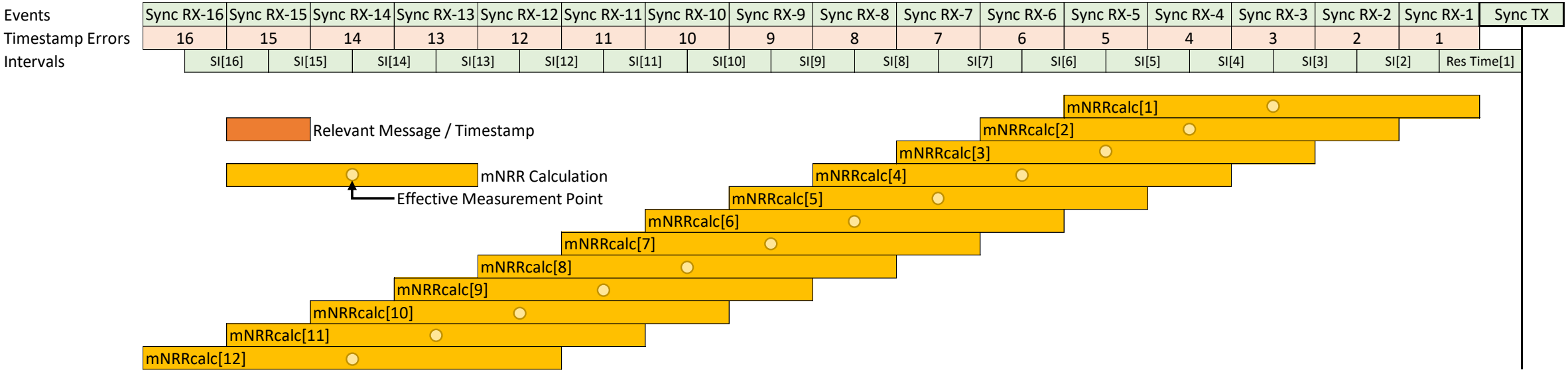
60802 – NRR Drift Tracking & Error Compensation

mNRR Drift Tracking



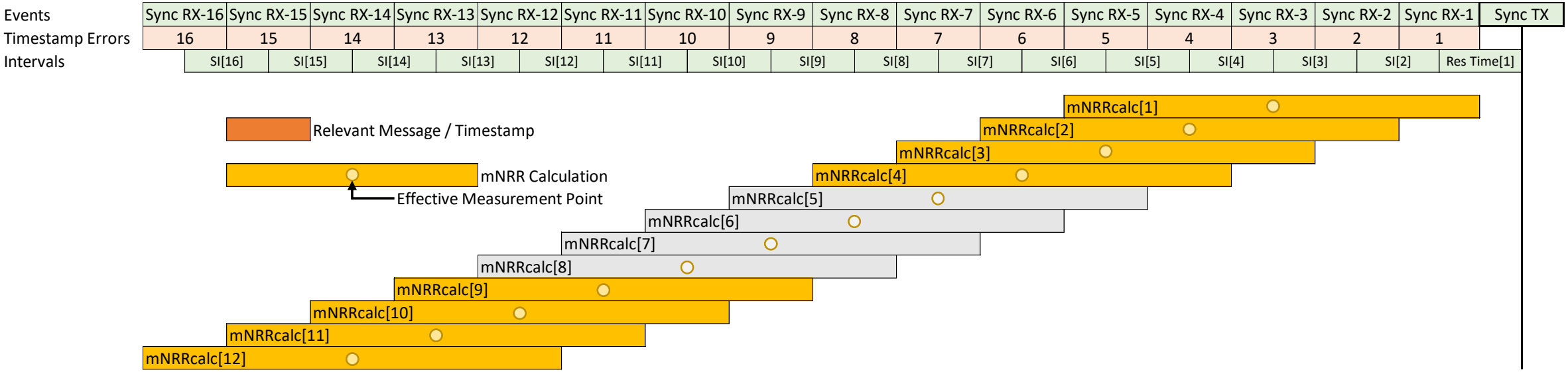
- Instead of keeping track of the past 4 calculation, each node keeps track of the past 12.

mNRR Drift Tracking



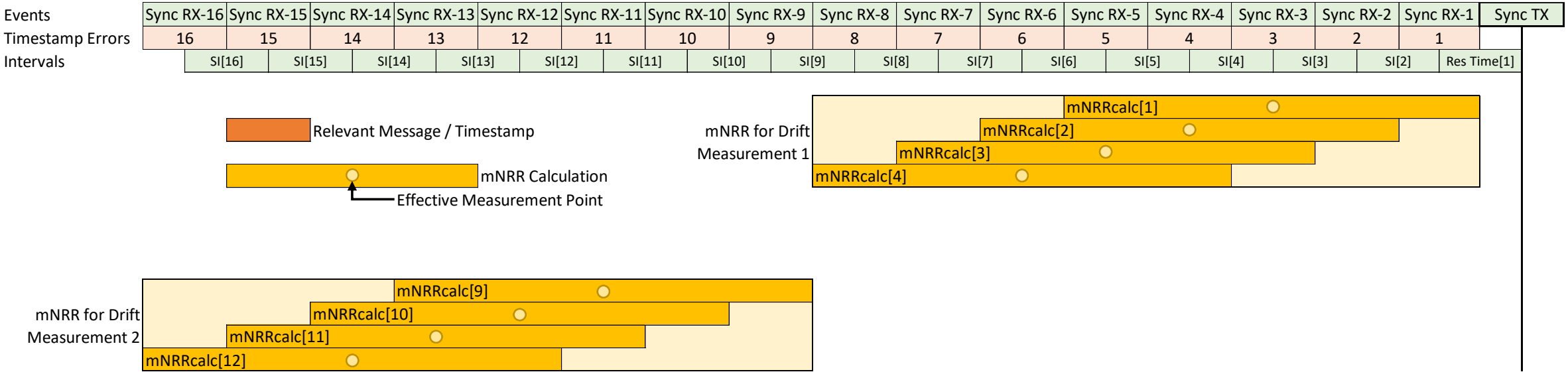
- Each calculation has an associated effective measurement time.

mNRR Drift Tracking



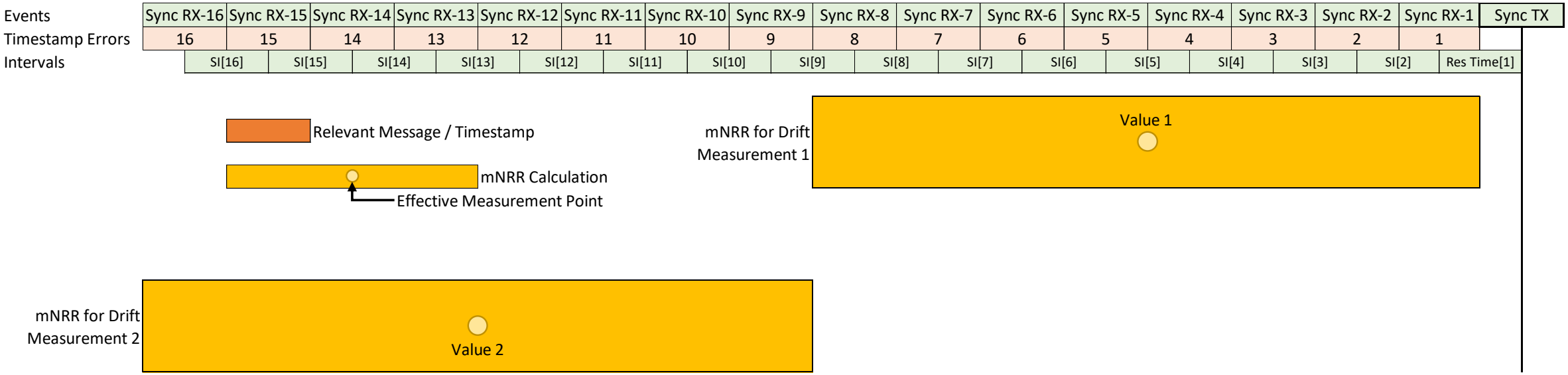
- The middle 4 calculations aren't used (but are remembered for future drift tracking)

mNRR Drift Tracking



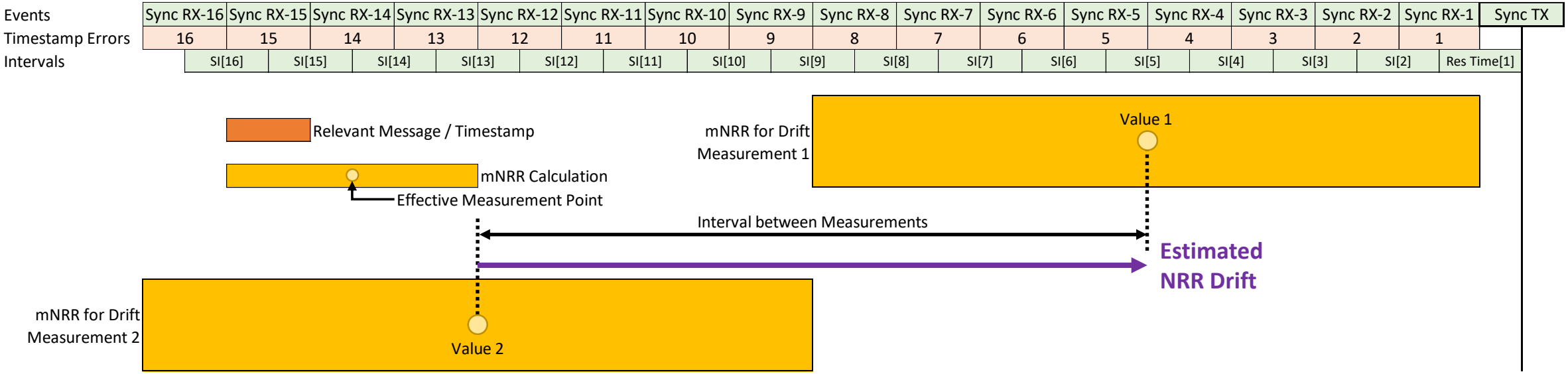
- The other calculations are used to make two measurements of NRR.

mNRR Drift Tracking



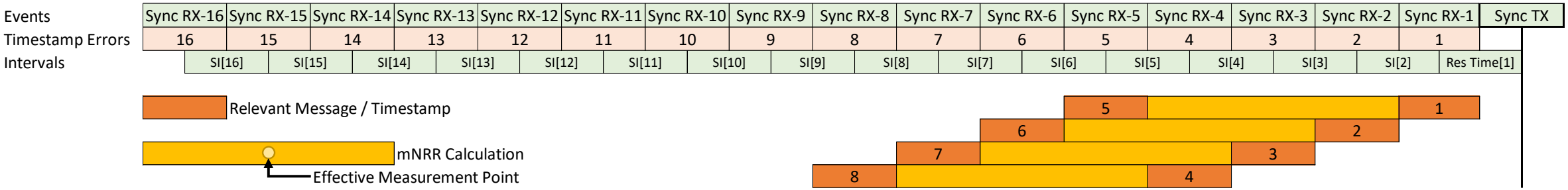
- Average of the values of each group of calculations.
- Average of the effective measurement points.

mNRR Drift Tracking



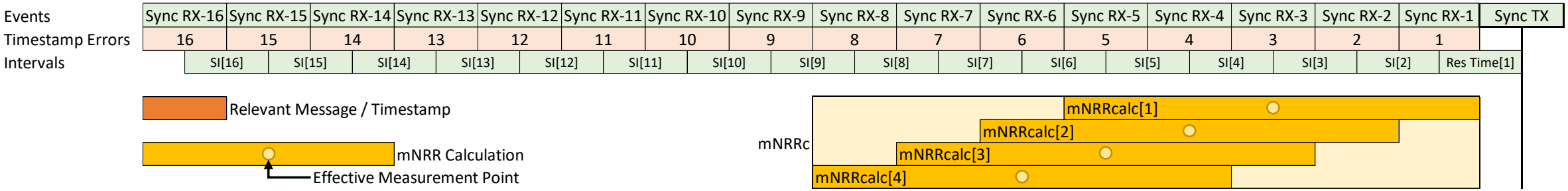
- Estimated NRR drift is the rate of change between the two values

mNRR Drift – Error Compensation



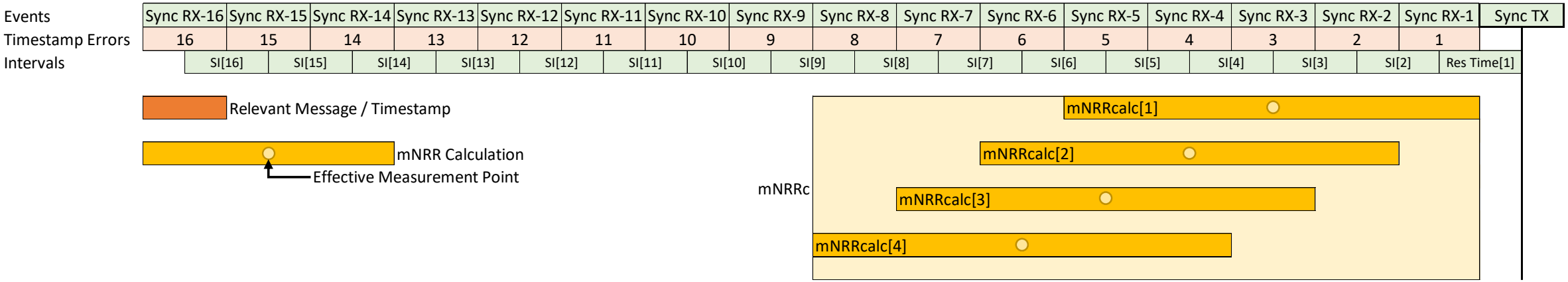
- Compensation for error due to clock drift in the mNRR measurement starts with the same timestamp information as before.
- Four calculations: (s, s-4), (s-1 & s-5), (s-2 & s-6), (s-3 & s-7)

mNRR Drift – Error Compensation



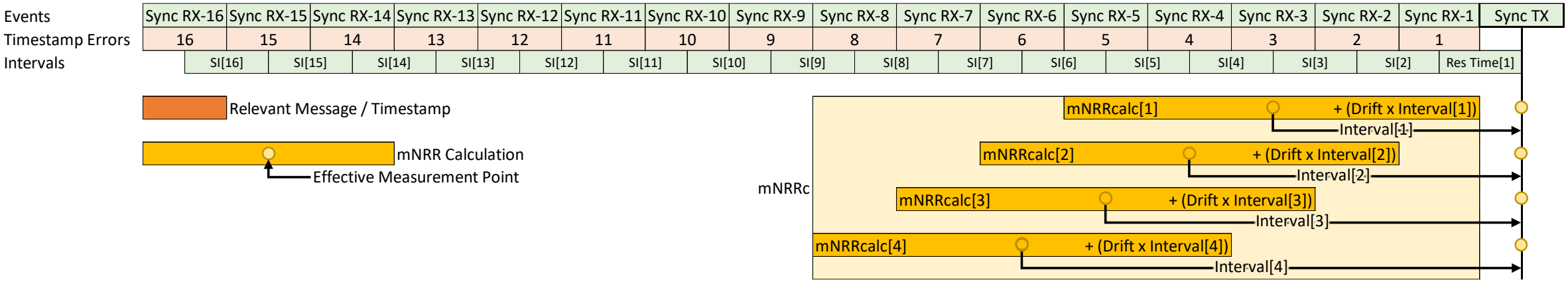
- Again, each value has an associated effective measurement point.

mNRR Drift – Error Compensation



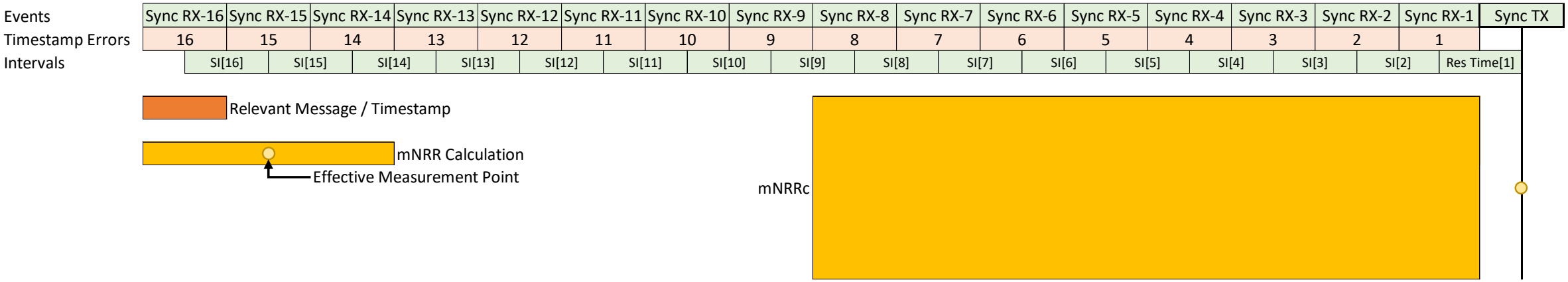
- But this time, prior to averaging, each value is adjusted using the effective measurement point and estimated drift rate to an estimate of the NRR at the point of Sync TX.

mNRR Drift – Error Compensation



- "Older" calculations require more adjustment to compensate for NRR drift.
- This process effectively moves the effective measurement point to Sync TX

mNRR Drift – Error Compensation

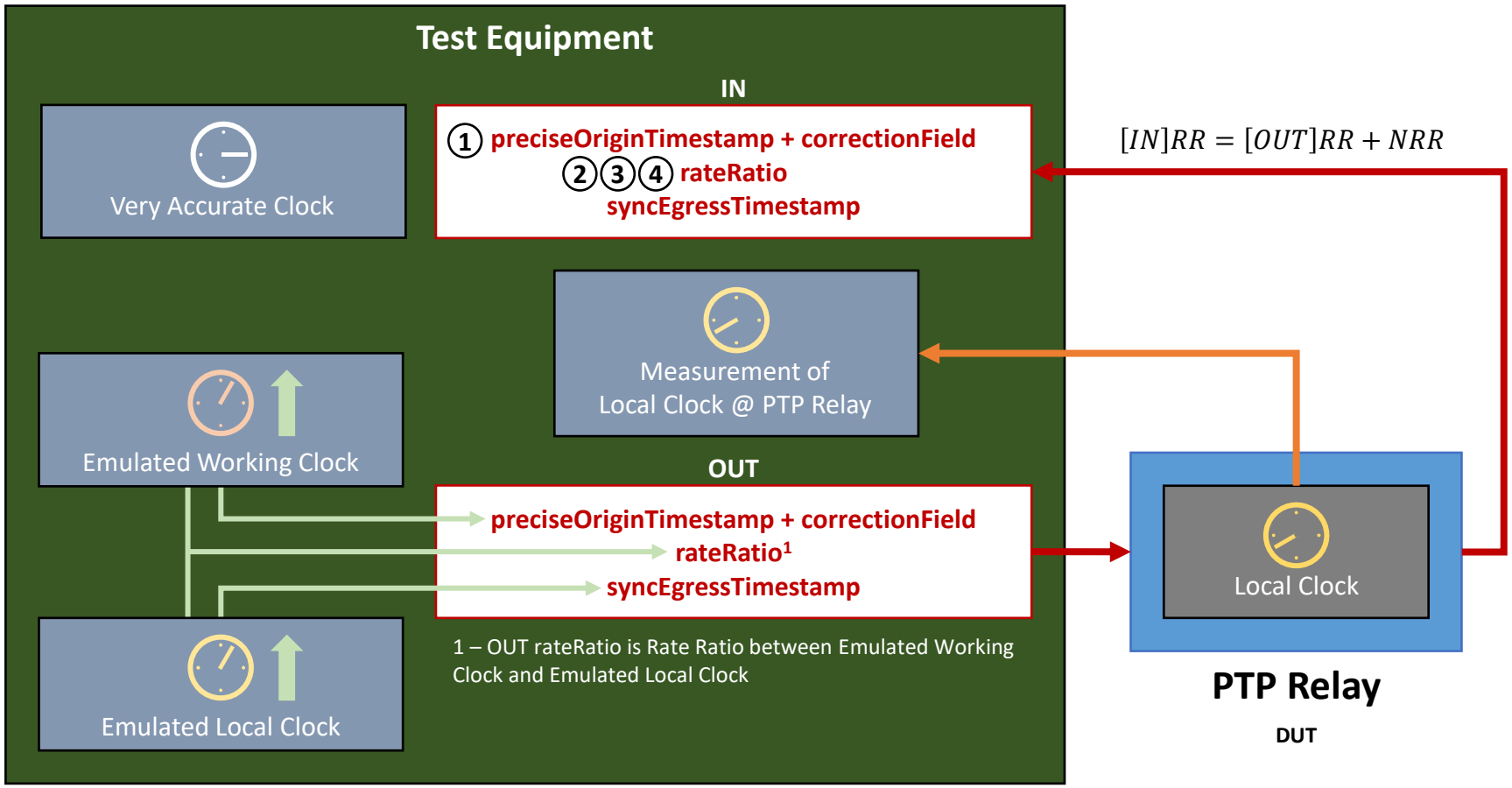


- Take an average of the four adjusted values to determine the measured, with error compensation, NRR: mNRRc.

Normative Requirements for NRR Drift Tracking & Error Correction at PTP Relay Instances

Measuring Error Generation at PTP Relay

From [2]

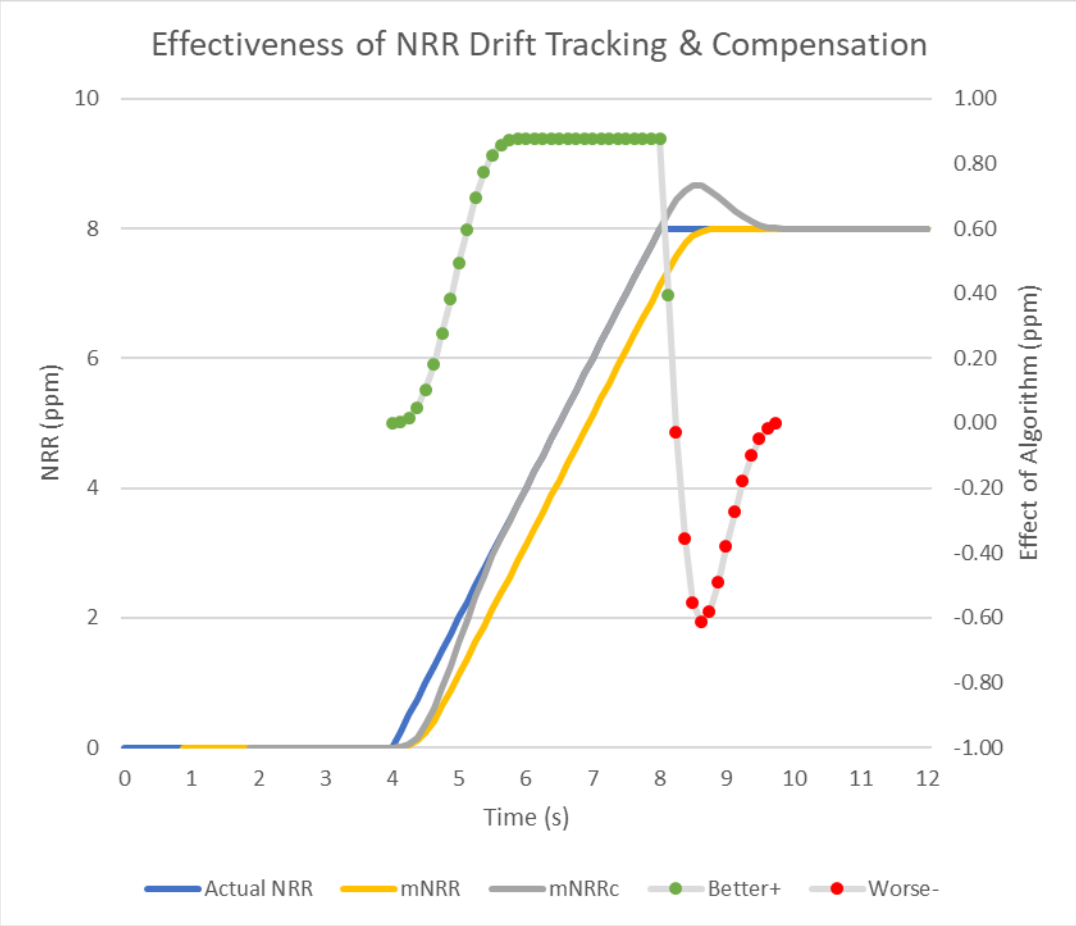


	Emulated Working Clock: Offset Increasing	Emulated Local Clock: Offset Increasing
4	**[IN] rateRatio** …should equal (within acceptable margin)… **Measured Rate Ratio (OUT rateRatio plus Rate Ratio between Emulated Local Clock and Measurement of Local Clock @ PTP Relay)**	

NRR Drift Tracking & Compensation – How does mNRRc behave?

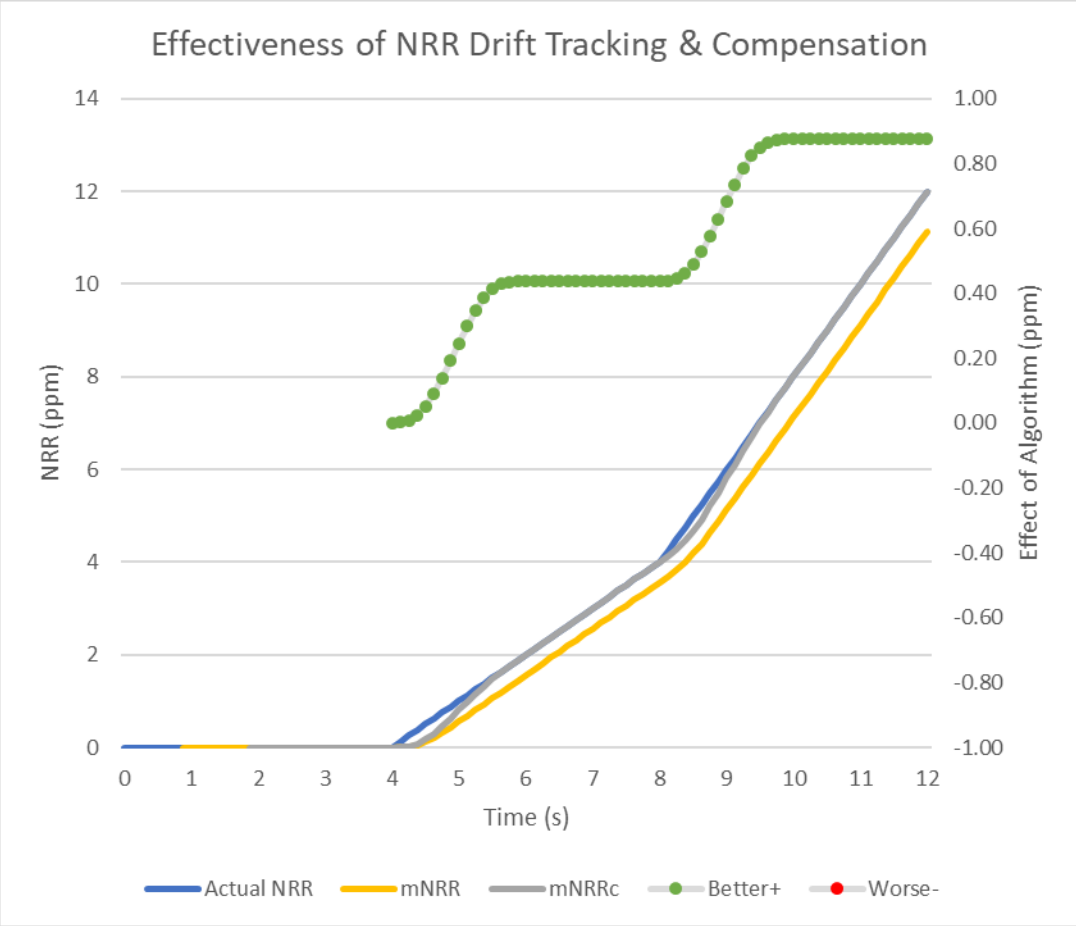
- Simple time series simulation in Excel
  - NRR; Two Devices; One Hop; 12 seconds
  - 125ms Sync Interval; exact, no variation.
  - No Timestamp Errors
  - No Residence Time; looking at NRR calculation on receipt of Sync message, i.e. value will include errors due to drift during measurement if no compensation algorithm is applied.
- Node n clock is nominal and stable throughout
- Node n-1 varies between stable, +1 ppm/s, +2 ppm/s and -2 ppm/s
- Compare NRR calculations with no tracking and compensation (mNRR) vs. tracking and compensation (mNRRc)

Node n-1: Stable → +2 ppm/s → Stable



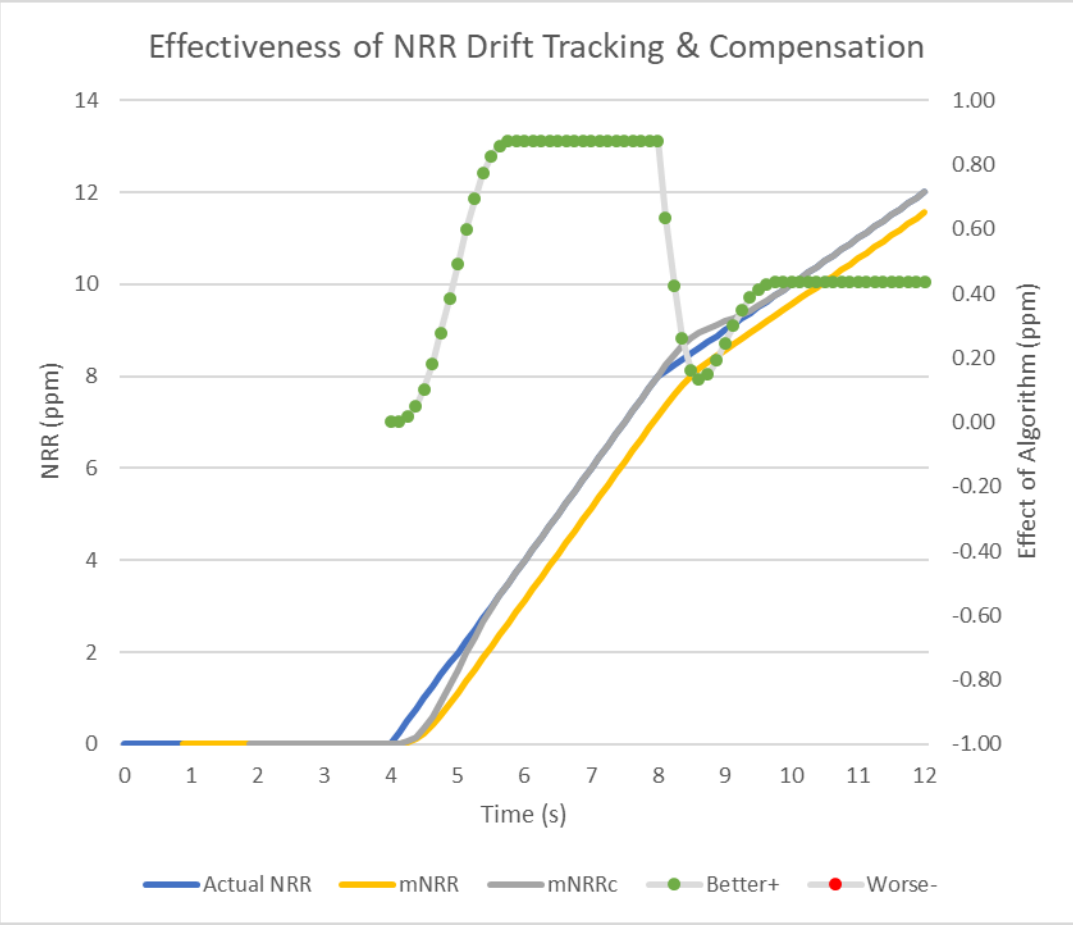
- Going from Stable to +2 ppm/s, mNRRc is always better than mNRR
- Going from +2 ppm/s to Stable, mNRRc briefly performs worse
  - Worst case occurs 625 ms after discontinuity before recovering fully 2 s after discontinuity

Node n-1: Stable → +1 ppm/s → +2 ppm/s



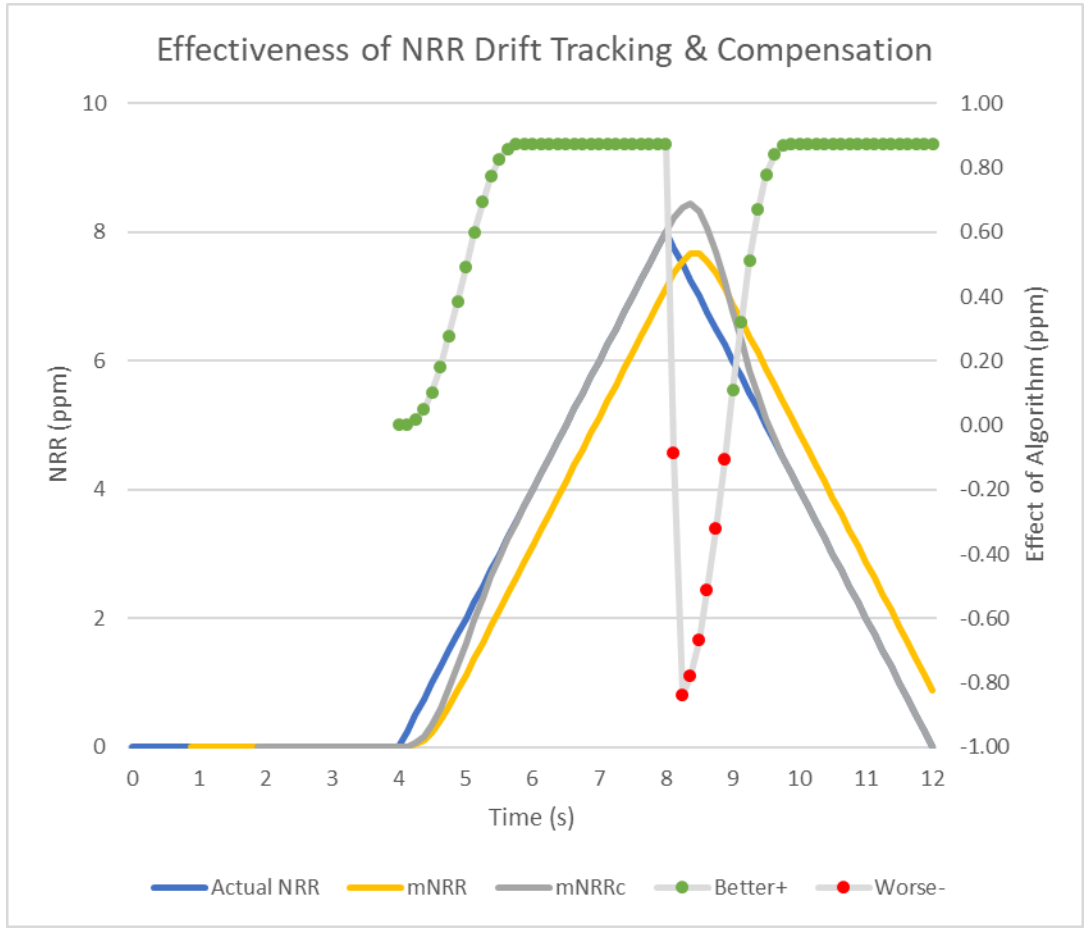
- Going from Stable to +1 ppm/s, mNRRc is always better than mNRR
- Going from +1 ppm/s to +2 ppm/s, mNRRc is always better than mNRR

Node n-1: Stable → +2 ppm/s → +1 ppm/s



- Going from Stable to +2 ppm/s, mNRRc is always better than mNRR
- Going from +2 ppm/s to +1 ppm/s, mNRRc is always better than mNRR
  - Benefit decreases to a temporary minimum 625 ms after discontinuity before recovering fully 2 s after discontinuity

Node n-1: Stable → +2 ppm/s → -2 ppm/s



- Going from Stable to +2 ppm/s, mNRRc is always better than mNRR
- Going from +2 ppm/s to -2 ppm/s, mNRRc is briefly worse than mNRR
  - Worst case occurs 250 ms after discontinuity; mNRRc is better than mNRR 1 s after discontinuity; maximum benefit and steady state achieved after 2 s

Normative Requirements – Issues

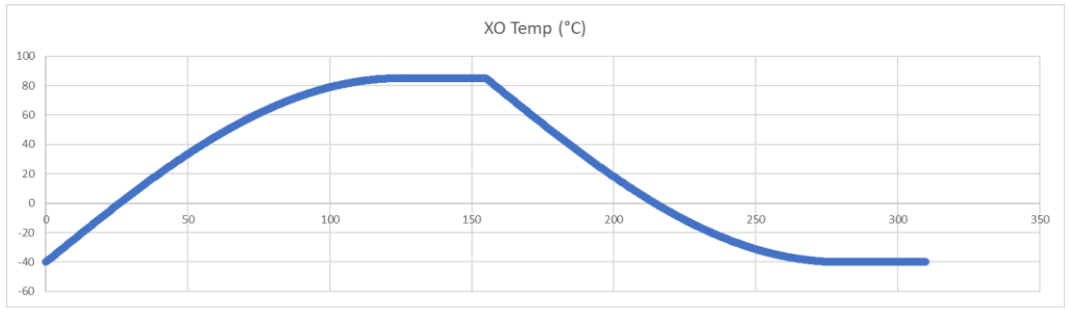
- Current requirements implicitly only looks at steady-state condition
  - Discontinuities, and stabalisation following them, are not covered.
  - It is not stated explicitly that this is the case, i.e. discontinuities and stabalisation are not mentioned at all.
- Current requirements define a performance requirement, not a specific algorithm
  - Intent is to document the described algorithm in an informative annex.
- The goal, previously discussed, is not to limit innovation in the development of “better” algorithms.
- But what if an implementation uses a “worse” algorithm, e.g. one that after a discontinuity, generates worse results and/or doesn’t settle for a longer amount of time? And is that the correct definition of worse?

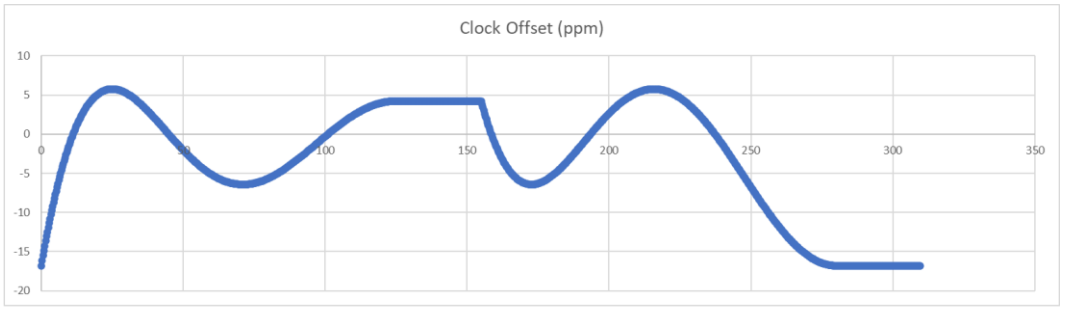
Using the Same NRR Tracking Algorithm at all PTP Relays

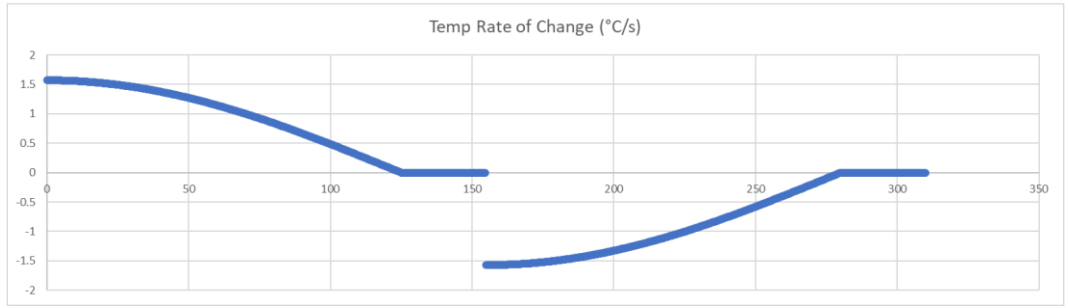
Absolute accuracy of a single mNRRc value isn't the end of the story…

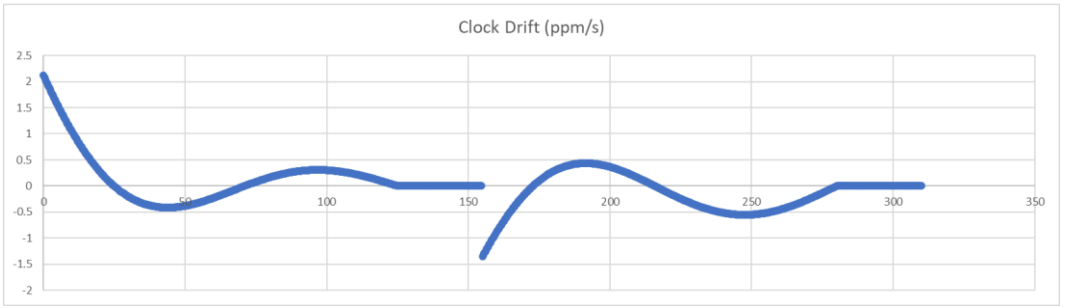
- Ultimately what matters is how much error NRR **survives** in RR
  - The more NRR error that **survives**, the **harder** it is for the End Station to accurately track Working Clock @ GM, or put another way…
  - **Less NRR error to start with**, and **more NRR error cancelling out** node-to-node makes it **easier** for the End Sation to accurately track Working Clock @ GM
- Measures discussed already are focused on delivering **less NRR to start with**.
- We also need to consider **how NRR error cancels out** with the use of NRR drift tracking and error correction algorithms

Half-Sinusoidal Temperature Ramp: 125s ↕







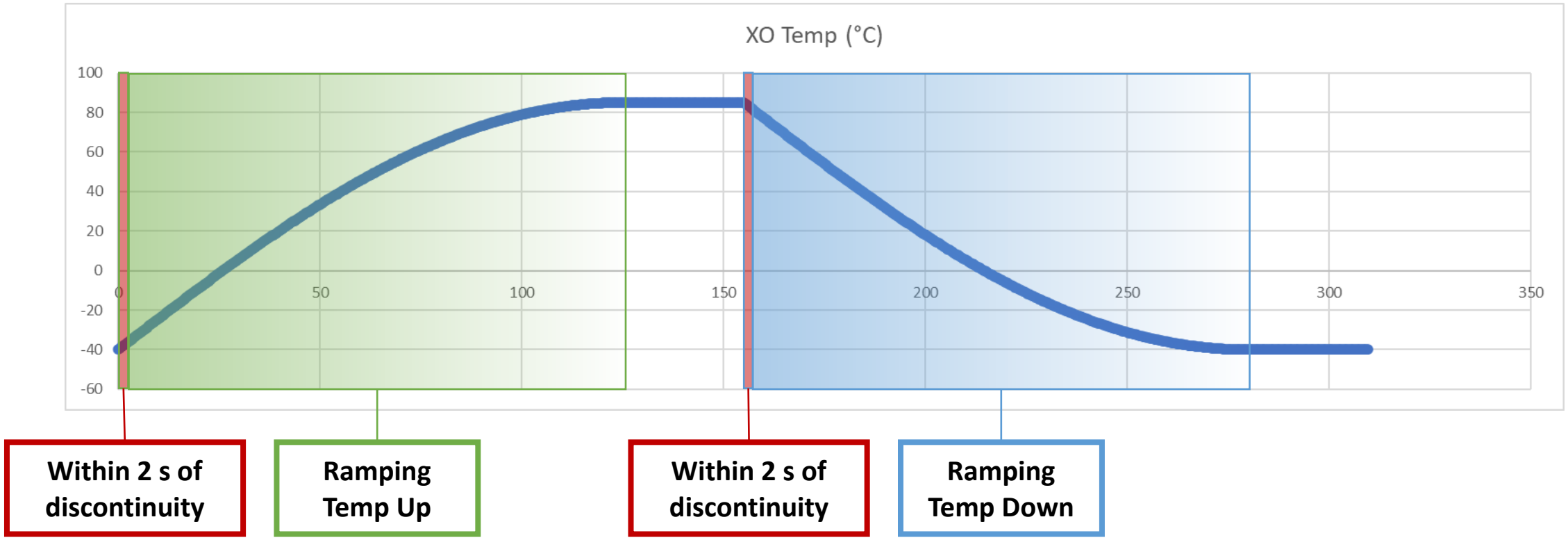


Inputs		
Temp Max	85	°C
Temp Min	-40	°C
Temp Ramp Period	125	s
Temp Hold	30	s

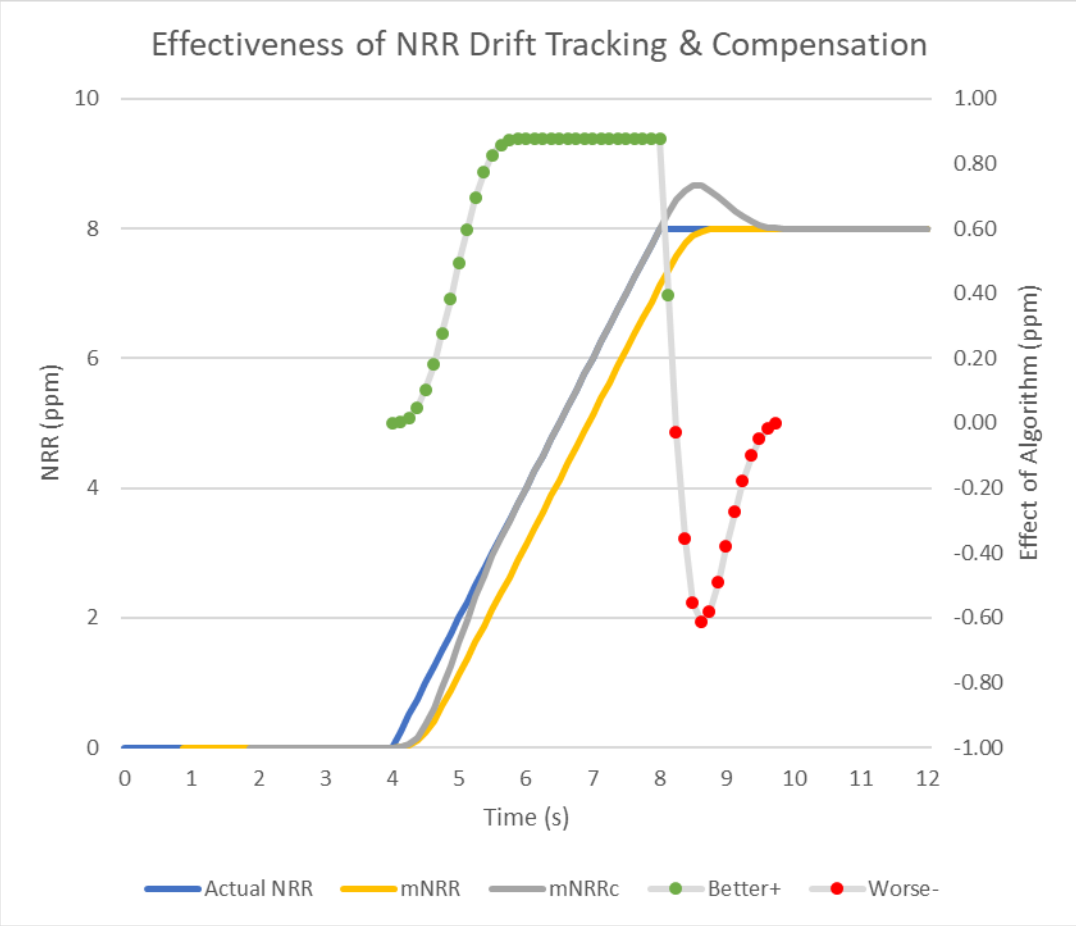
Temp Rate of Change		
MAX	1.57	°C/s
MIN	-1.57	°C/s

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

Temperature Ramp Phases

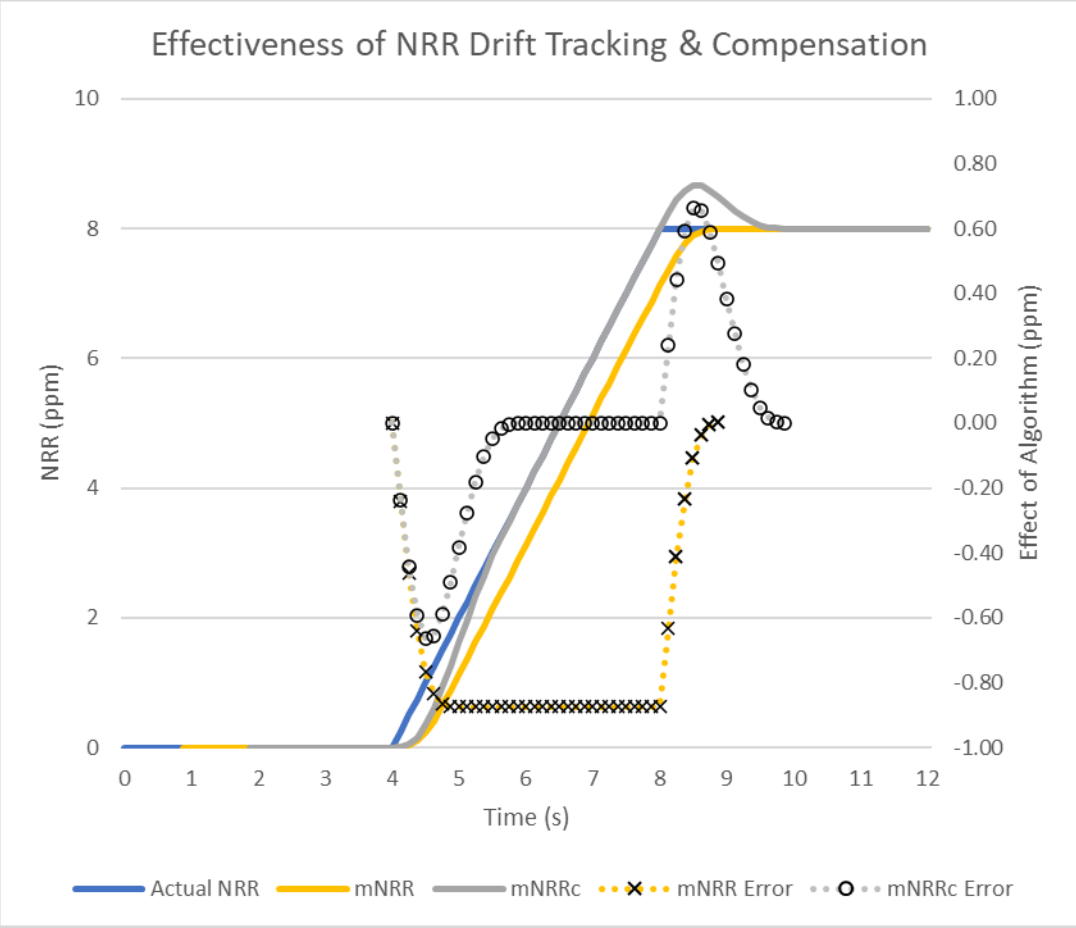


Node n-1: Stable → +2 ppm/s → Stable



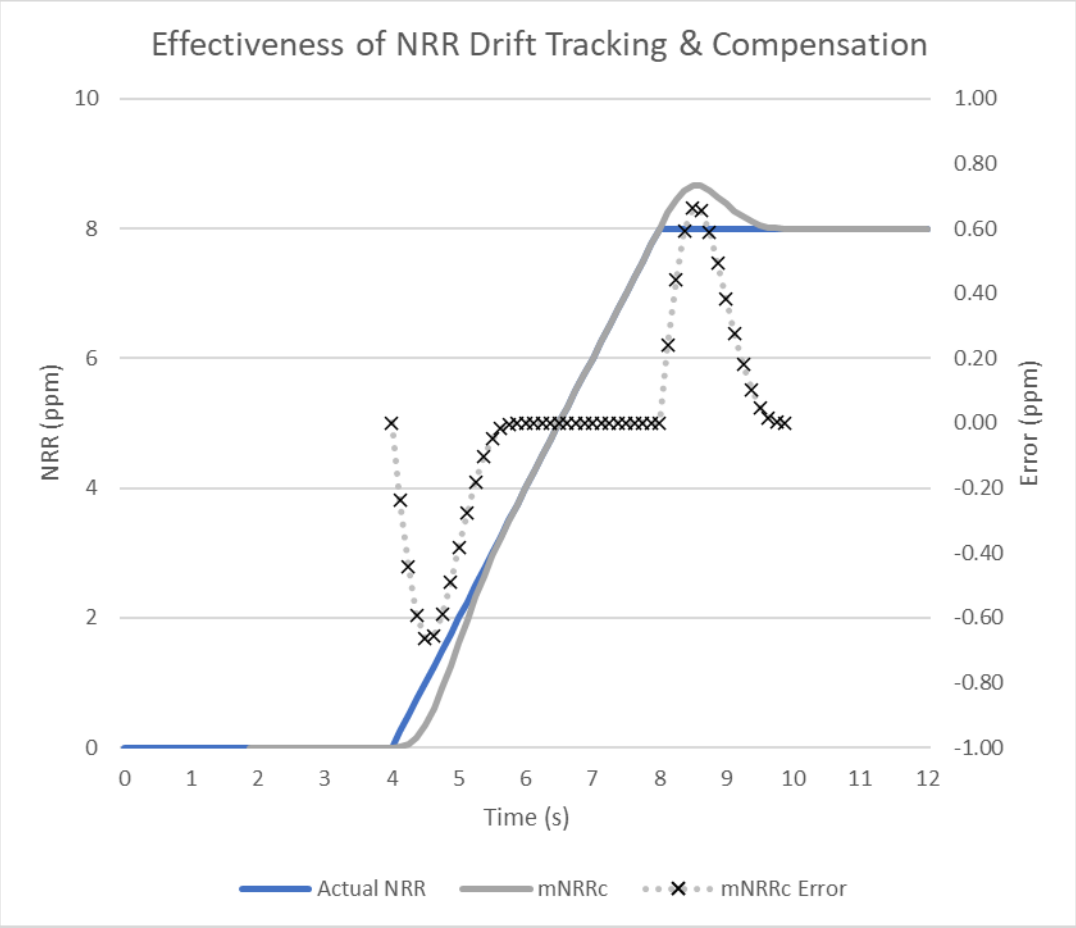
- Going from Stable to +2 ppm/s, mNRRc is always better than mNRR
- Going from +2 ppm/s to Stable, mNRRc briefly performs worse
  - Worst case occurs 625 ms after discontinuity before recovering fully 2 s after discontinuity

Node n-1: Stable → +2 ppm/s → Stable



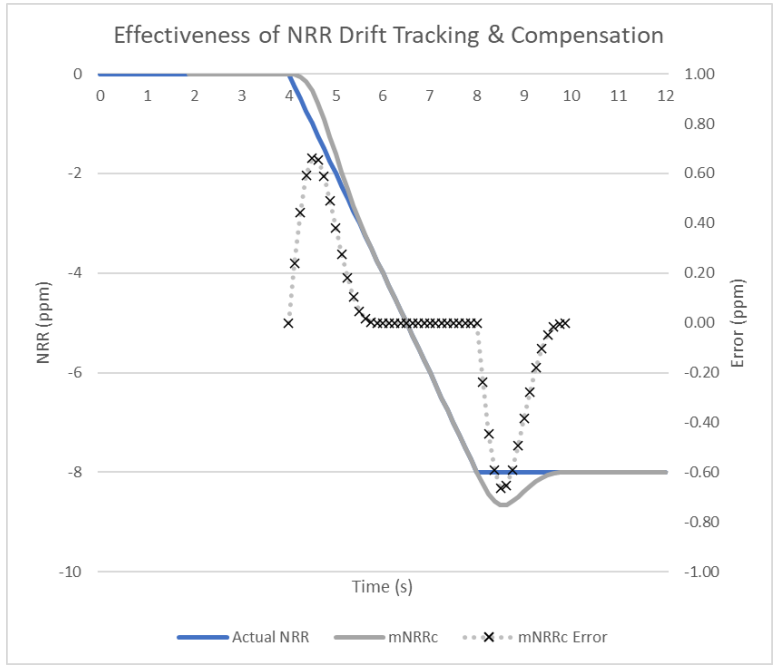
- What matters for NRR cancellation node-to-node isn't mNRR vs. mNRRc, but the errors in each.

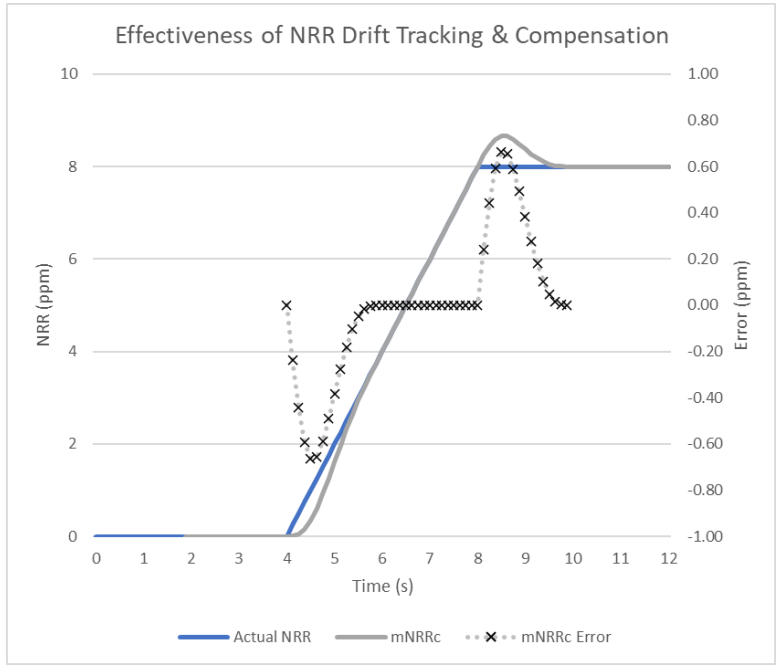
Node n-1: Stable → +2 ppm/s → Stable

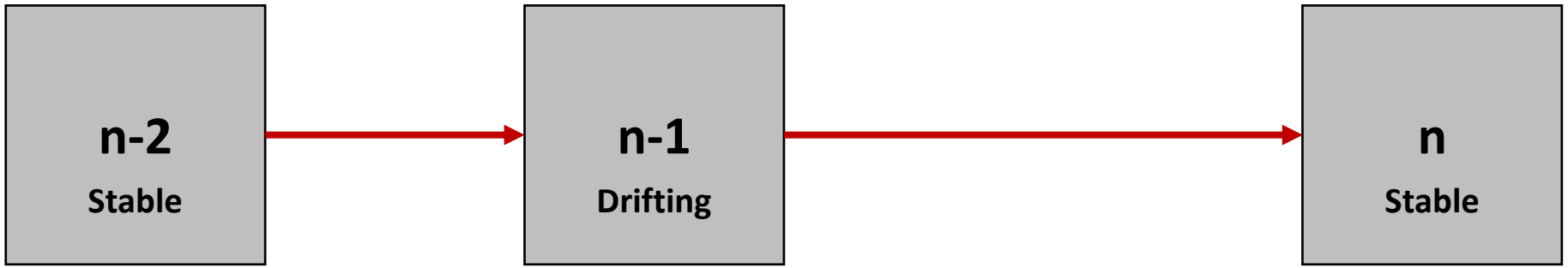


- And for this discussion, we're really just interested in errors in mNRRc.

How much error cancels out?







Implications

- Node-to-Node cancellation of NRR error (that survives in RR) is maximised when subsequent nodes use the same algorithm which generates the same errors.
  - Assuming the algorithm is symmetrical, i.e. responds the same to rising and lowering NRR.
- The means our simulations may be simulating an ideal case
  - No variation in algorithm

Next Steps & Recommendations

- Recommend: Normative requirements on algorithm responsiveness
  - TBD: how responsive? 2 seconds?
- Next Step: Complete simulations using the discussed algorithm.
  - Determine error budgets and margins.
- If margins are tight, conduct further simulations mixing in alternative algorithms to determine sensitivity
- For discussion: should we specify a normative algorithm?
- Note: this concern doesn’t apply to RR drift tracking and error compensation as that doesn’t cancel out node-to-node.

Thank you!