60802 Time Sync Ad Hoc mNRRsmoothing Optimisation & Aligning pDelayResp & Sync

David McCall – Intel Corporation

Version 1

Content

- mNRRsmoothing Optimisation
- Aligning pDelayResp & Sync Messaging

mNRR Smoothing Optimisation

Background

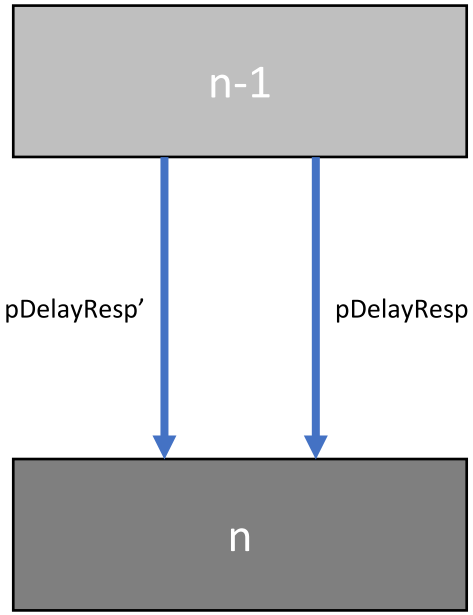
- IEEE 802.1AS measures Rate Ratio (RR) via an accumulation of Neighbor Rate Ratios (NRRs). Classically, NRR is measured via timestamps from the two most recent pDelayResp messages.
- Errors in the measured NRR (mNRR) can arise from Timestamp Errors and errors due to Clock Drift between nodes.
  - As 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 $mNRR_{error}$.

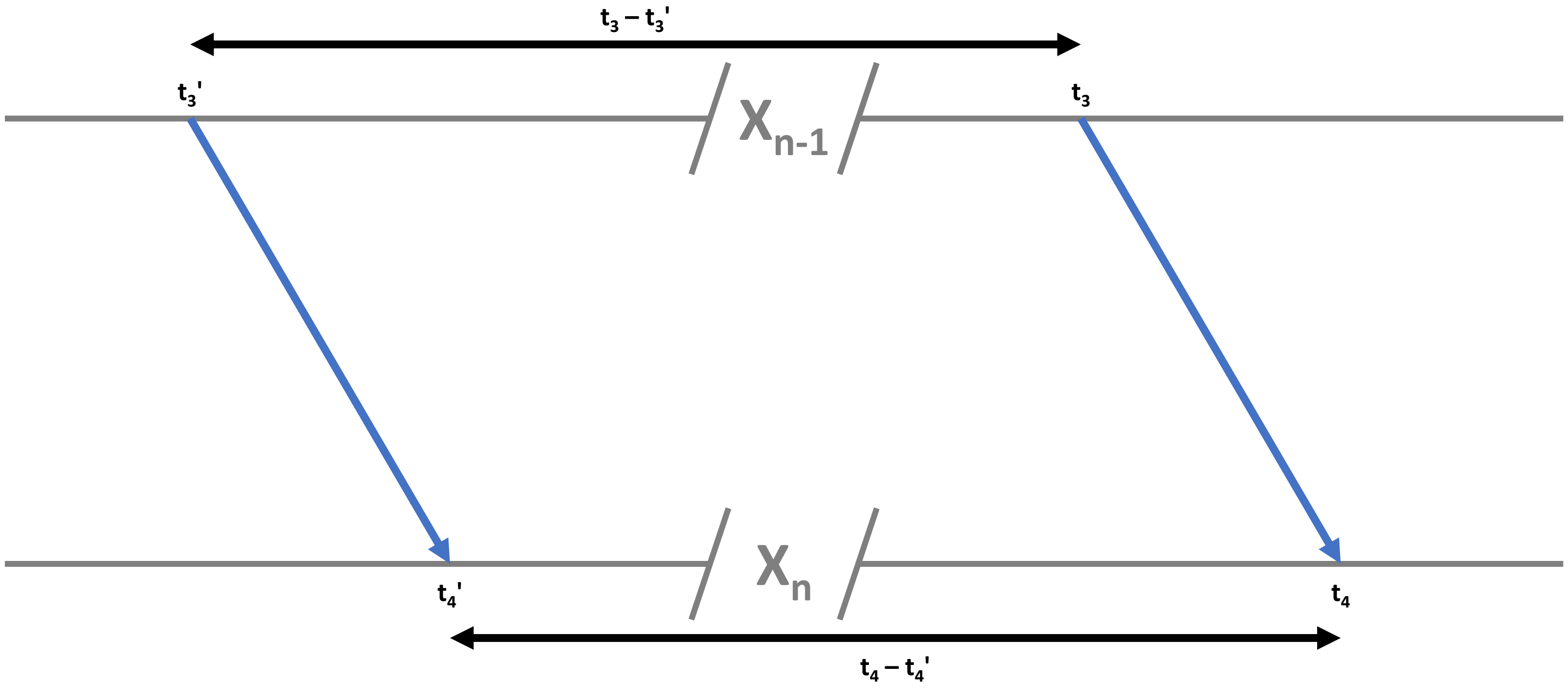
References

[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

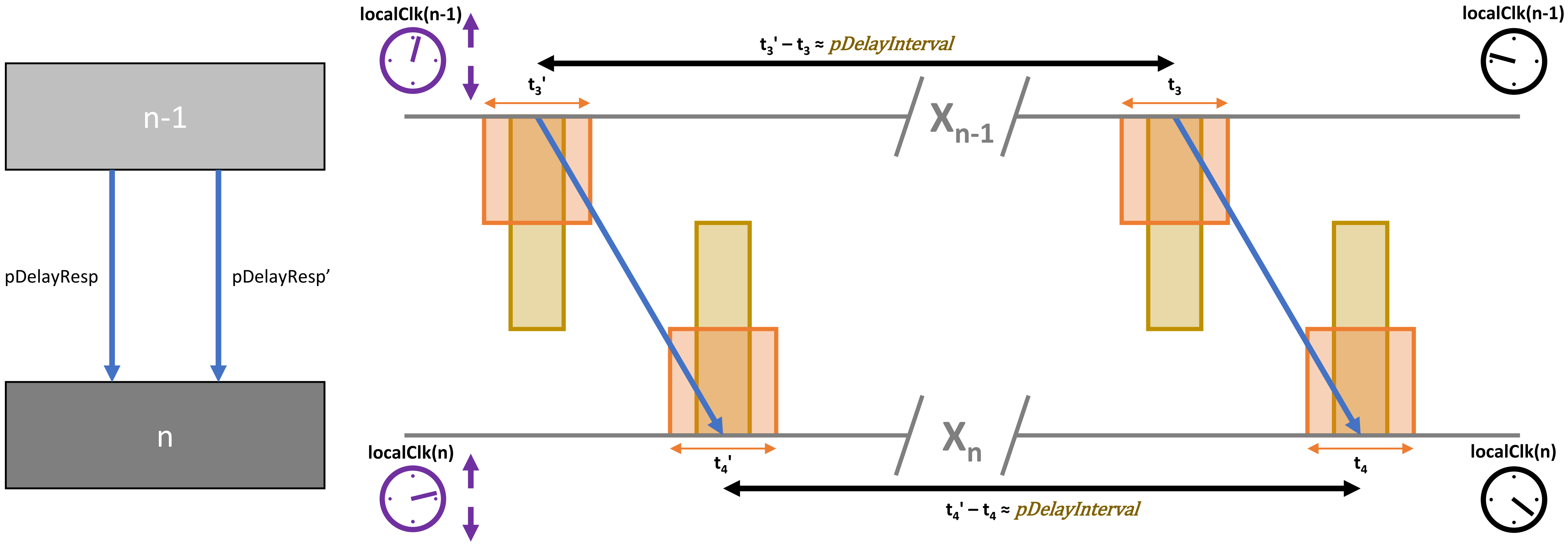




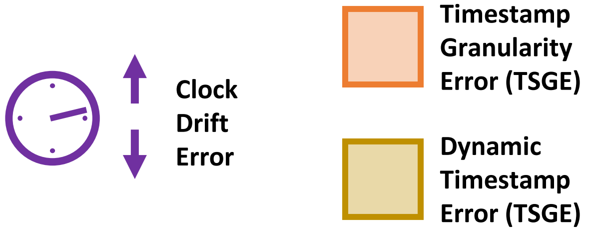
$$mNRR = \left(\frac{t_3 - t_3{}'}{t_4 - t_4{}'}\right)$$

ppm

Background - $mNRR_{error}$



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

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

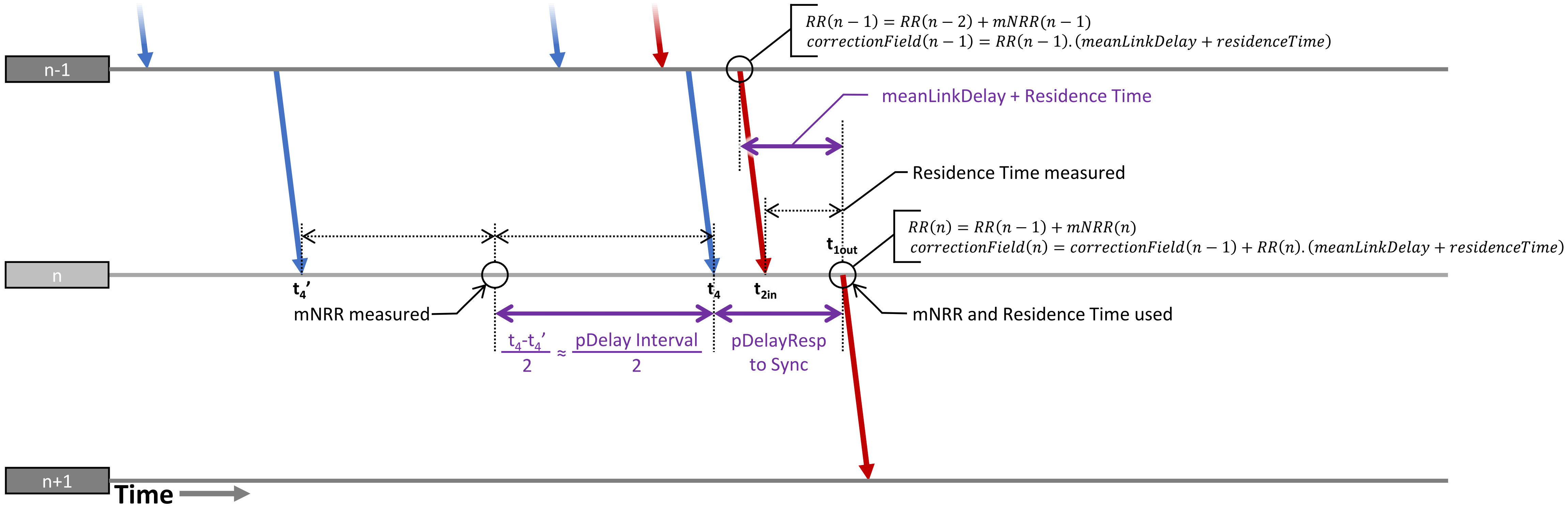
$$\boldsymbol{Error_{TSGERX}} = \sim U(-\boldsymbol{TSGE_{RX}}, +\boldsymbol{TSGE_{RX}})$$

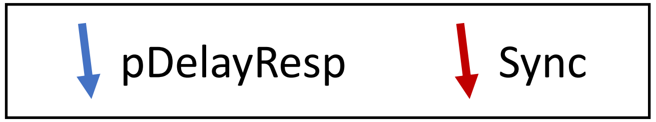
- DTSE magnitude and probability distribution is implementation dependant, but implementations that deliver a uniform probability between a minimum and maximum, equally spread either side of zero, are common and a worst case.
  - Triangular or normal distributions will have fewer extreme errors.

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

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

Background - Clock Drift Error – Relevant Intervals





- 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 – $mNRR_{error}$ 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.

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

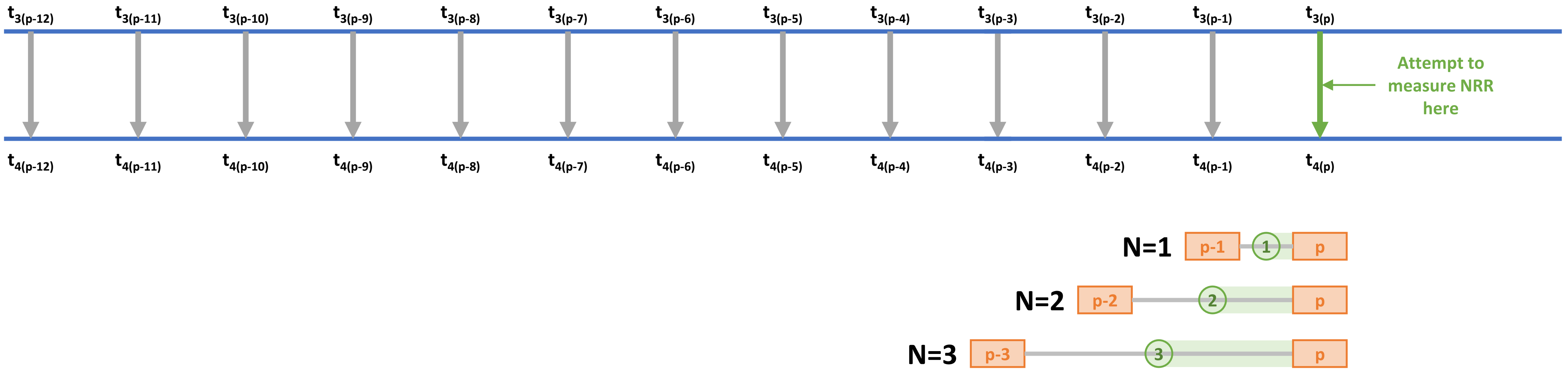
Background - mNNRsmoothingN

- 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.

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

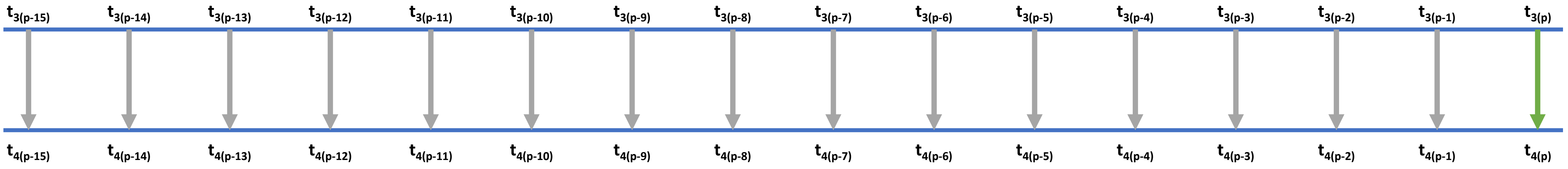
Background – mNRR Smoothing N



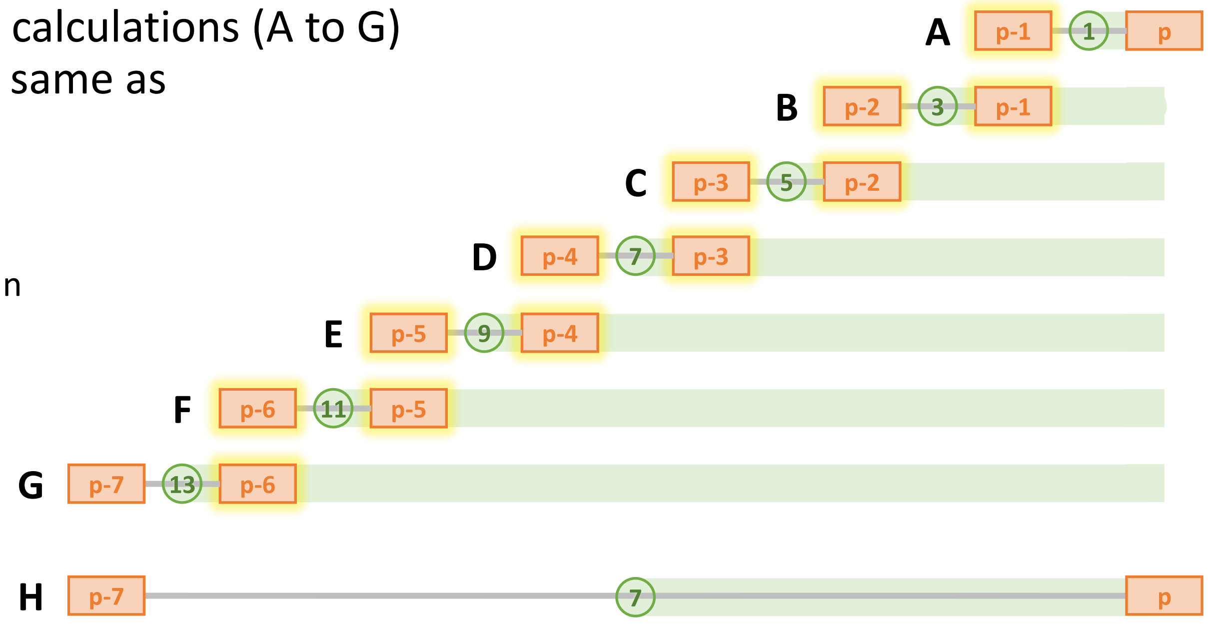
Background – mNRR Smoothing M

- 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.

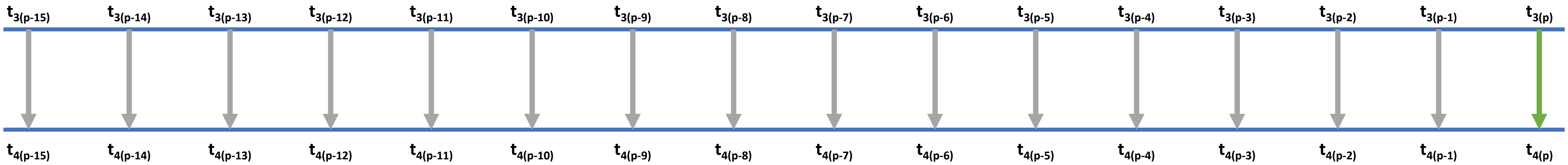
Taking a Simple Average of mNRR Calculations



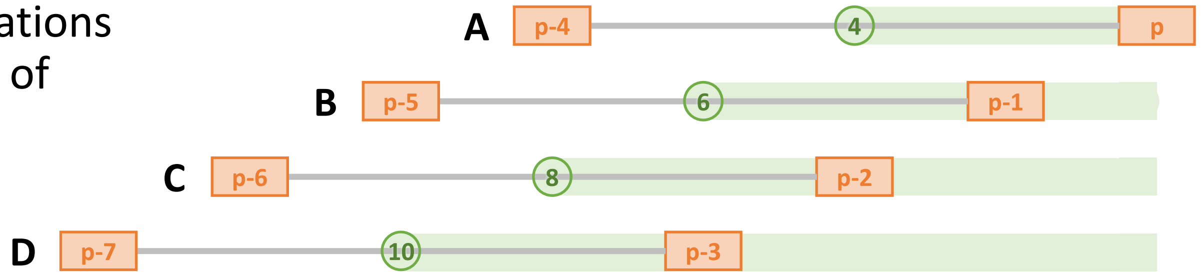
- Taking a simple average of the more recent 8 mNRR calculations (A to G) where mNRRsmoothingN = 1 is mathematically the same as a single calculation where mNRRsmoothingN =7 (H)
  - Exactly the same for Timestamp Error
  - Approximately the same for error due to Clock Drift. The effective measurement 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



- 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.

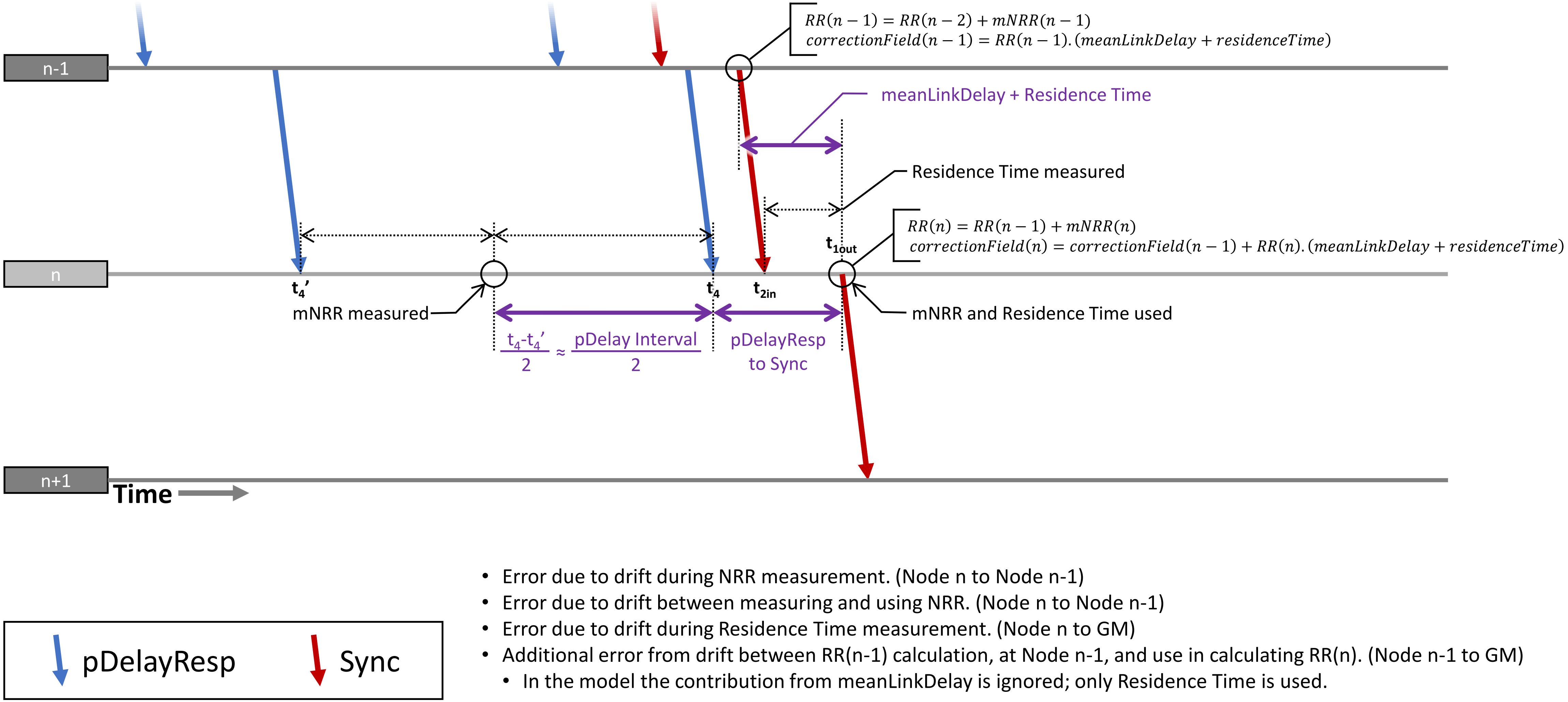


Status

- Monte Carlo simulation *almost* complete. Will present results during meeting on Monday 31st October.
- Will also build this option into main 100-hop simulation, assuming results look favourable.

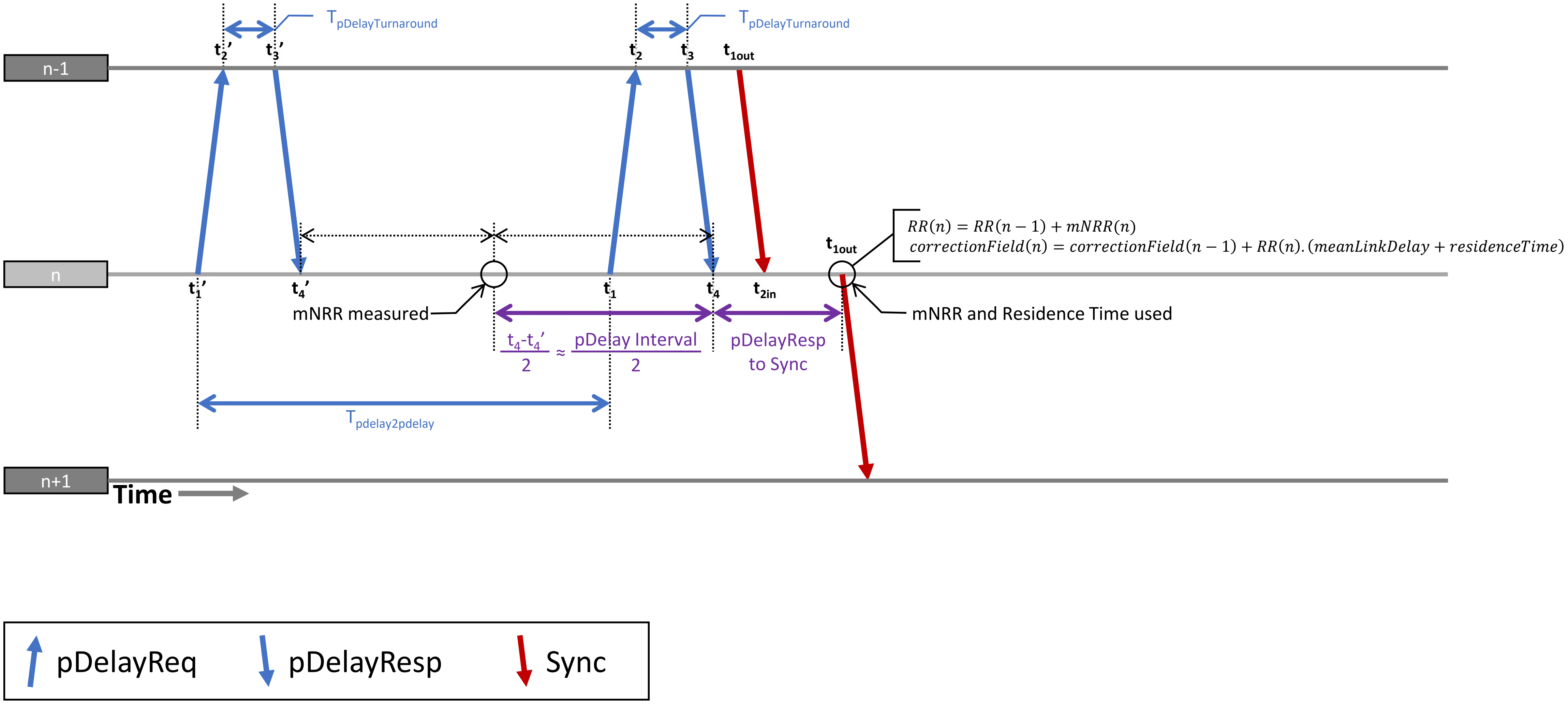
Aligning pDelayResp & Sync

Background - Clock Drift Error – Relevant Intervals



- 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 - Clock Drift Error – Relevant Intervals



Feasibility & Normative Requirements?

- Limits on variability of...
  - $T_{sync2sync}$
  - $T_{pdelay2pdelay}$
  - $T_{pdelayTurnaround}$
  - $T_{ressidenceTime}$
- Residence Time and pDelayTurnaround maximums still apply
  - Additional limit on variability. 95% of results fall within X range e.g. 1ms; 100% fall within Y range e.g. 2ms.
- pDelayInterval and syncInterval limits are much tighter
  - Within ±5ms of nominal value.
- Discussion

Backup