60802 Time Sync:
Reducing dTE – Complexities & Tradeoffs –
Ad Hoc Next Steps

David McCall (Intel)

Version 2

References

1. D. McCall, “60802 Time Sync Ad Hoc Status Update”, IEC/IEEE 60802 contribution”, September 2022
2. D. McCall, “60802 Time Synchronisation – Monte Carlo Analysis: 100-hop Model, “Linear” Clock Drift, NRR Accumulation, Overview & Details, Including Equations”, IEC/IEEE 60802 contribution, September 2022

Background

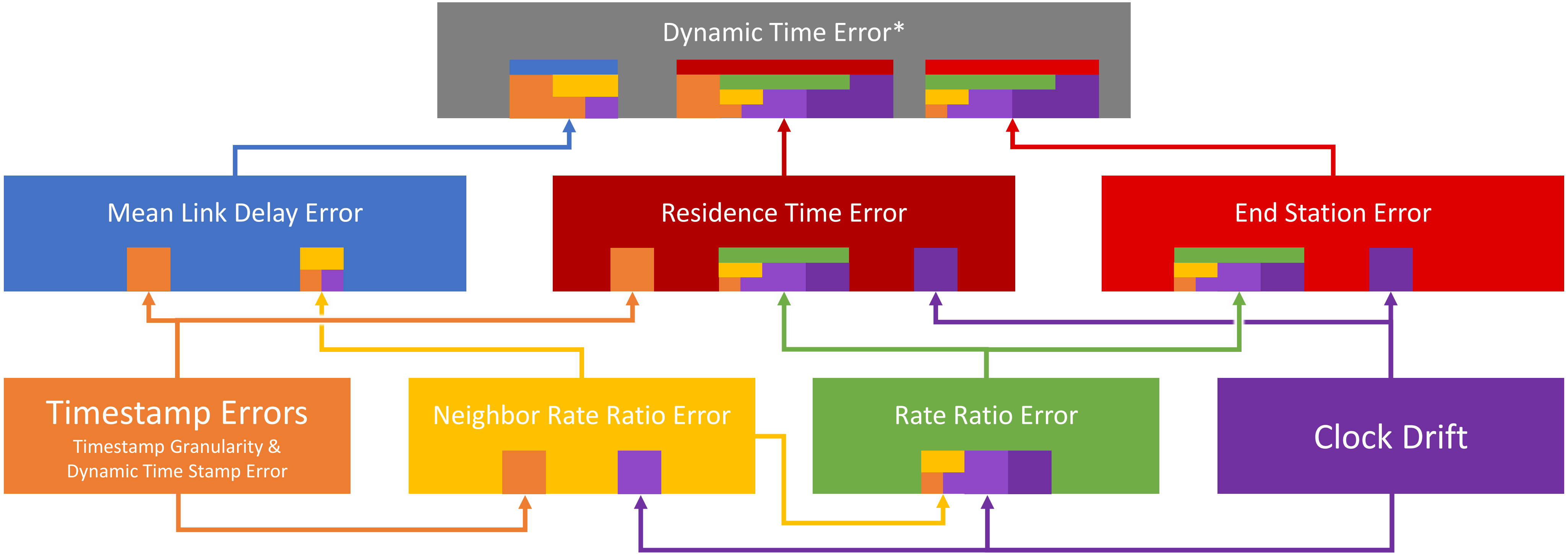
- IEC/IEEE 60802 has a stated requirement of 1us time accuracy over 64 hops (i.e. 65 devices) with a goal of 100 hops (i.e. 101 devices).
- The Monte Carlo Analysis has identified several components of dynamic time error (dTE) that can be addressed by altering parameters or via algorithmic compensation.
- Most of these approaches involve tradeoffs and additional complexity
- This presentation provides and overview of the approaches, tradeoffs and potential complexities to elicit feedback on where an acceptable balance may lie.

Content

- Components of dTE
- pDelay Interval
- Sync Interval
- Residence Time
- Aligning pDelayResp with Sync
- mNRRsmoothingN – using older pDelayResp timestamps
- mNRRsmoothingM – taking a median of previous mNRR values
    - And why this is a bad idea
- NRR measurement and compensation
- RR measurement and compensation

Components of DTE

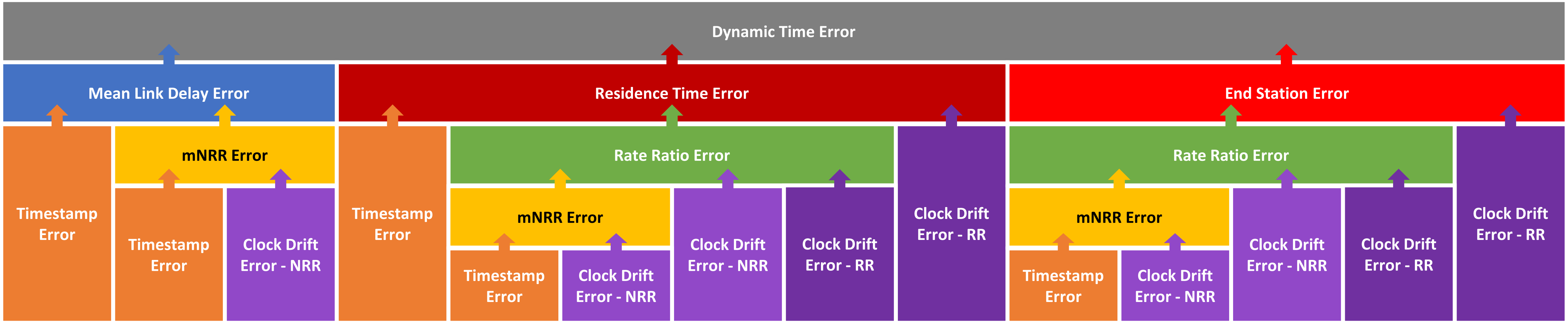
Dynamic Time Sync Error Accumulation



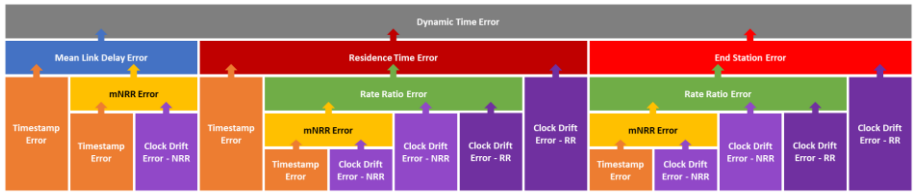
All errors in this analysis are caused by either Clock Drift or Timestamp Errors

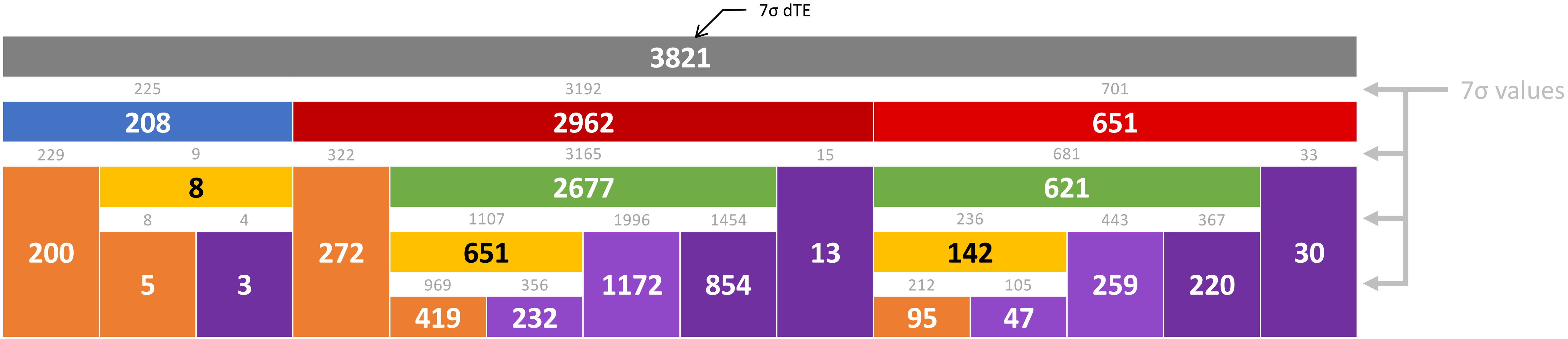
*DTE based on protocol messaging only. Total DTE at the application level will also depend on ClockMaster, ClockSlave, ClockSource, Clock Target, etc...

Graphical Representation of Error Accumulation



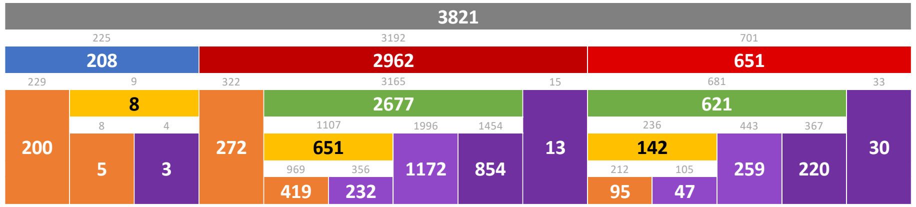
Graphical Representation of Error Accumulation



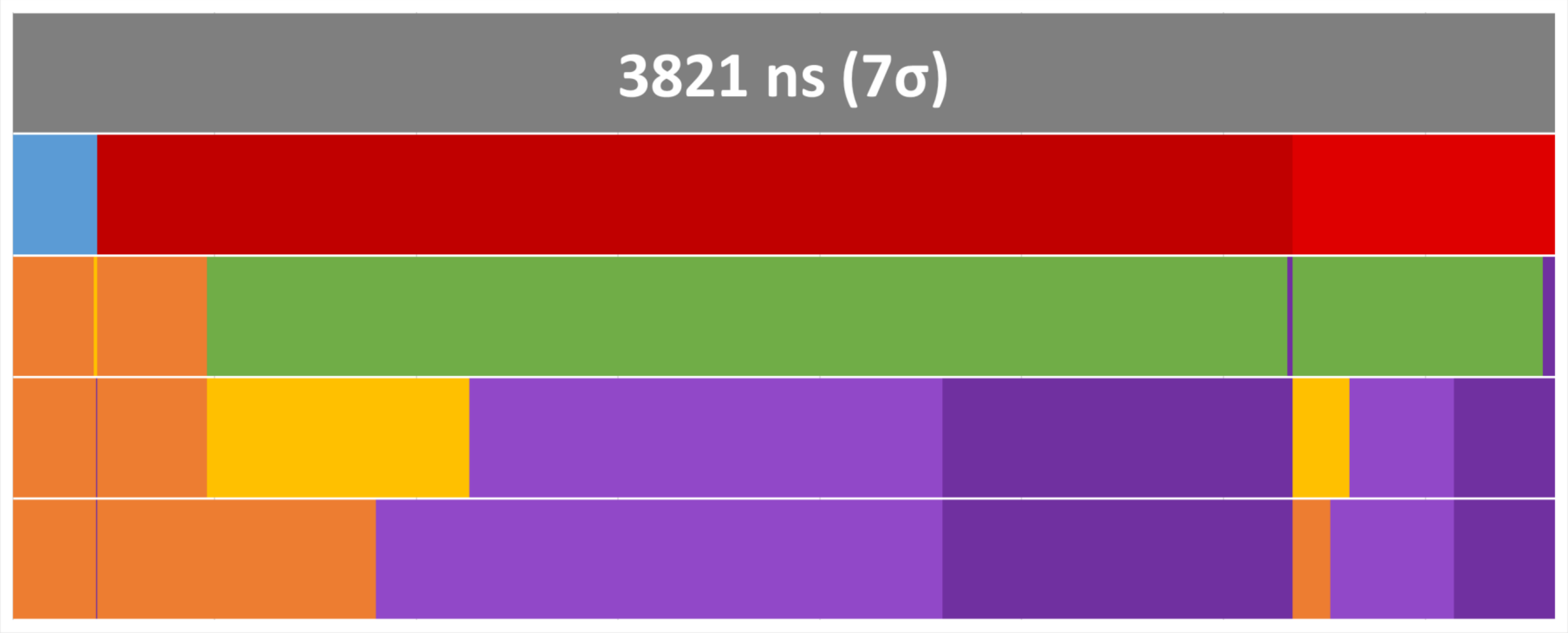


The 7σ dTE value is split repeatedly according to the ratio of 7σ values of underlying errors. 7σ probabilities do not combine via addition so, at each level, the sum of the underlying 7σ values is greater than the value that is being split. Larger errors will often swamp smaller errors, so small errors are, in general, over-represented by this approach. It does, however, provide a useful visualisation of how underlying errors combine to make up the 7σ dTE value.

Graphical Representation of Error Accumulation

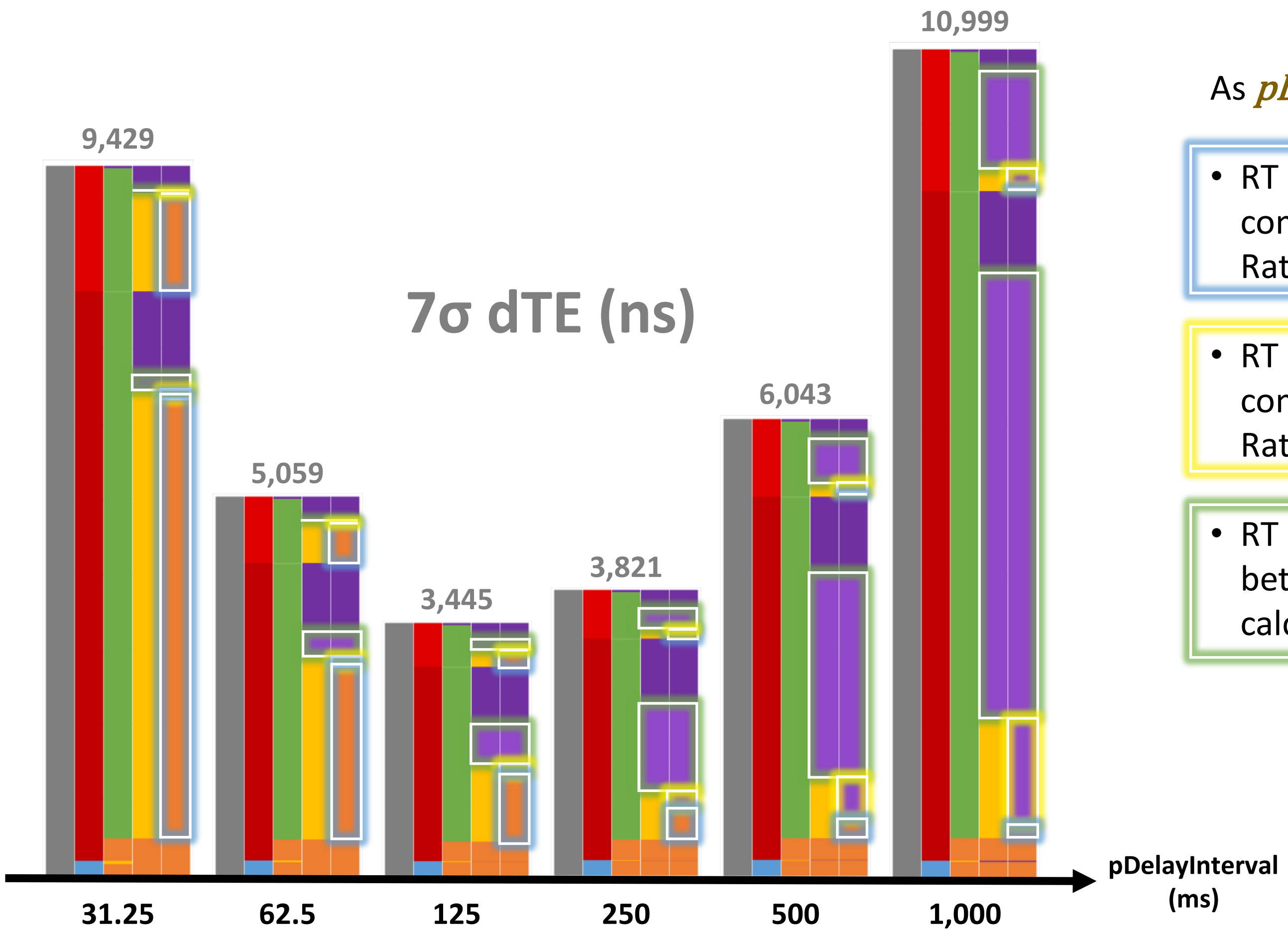


Input Errors		
Drift Type (Linear Temp Ramp)	2	
GM Clock Drift Max	+1.35	ppm/s
GM Clock Drift Min	-1.35	ppm/s
Fraction of GM nodes w/ Drift	80%	
non-GM Clock Drift Max	+1.35	ppm/s
non-GM Clock Drift Min	-1.35	ppm/s
Fraction of non-GM Nodes w/ Drift	80%	
Temp Max	+85.	°C
Temp Min	-40.	°C
Temp Ramp Rate	±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	250	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	0%	
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	1	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	1,000,000	



pDelayInterval Sensitivity Analysis

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	VAR	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	0%	
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	1	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	500,000	



As *pDelayInterval* is reduced…

- RT & ES errors due to **Timestamp** component of $mNRR_{error}$, via Rate Ratio **increase**.

- RT & ES errors due to **Clock Drift** component of $mNRR_{error}$, via Rate Ratio **decrease**.

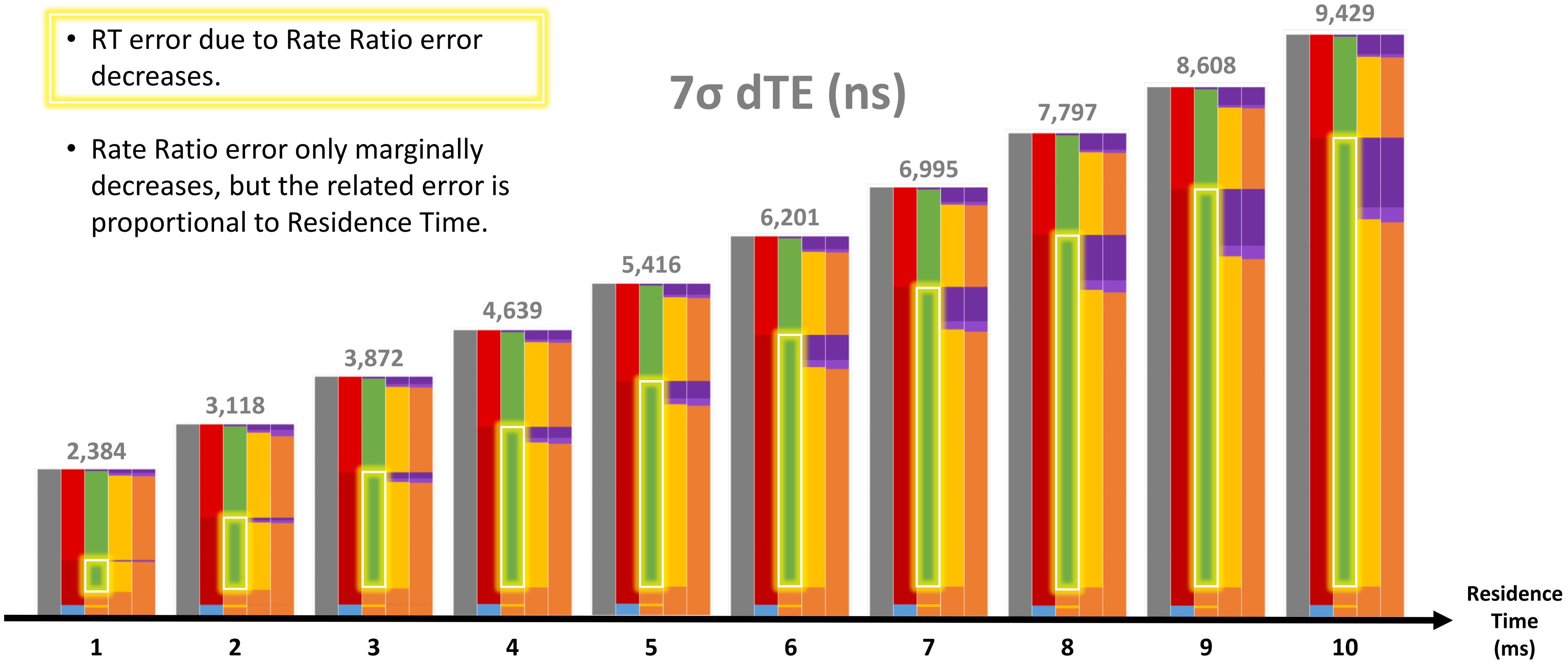
- RT & ES errors due to **Clock Drift** between measurement of NRR and calculation of RR **decrease**.

residenceTime Sensitivity Analysis

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	VAR	ms
Input Correction Factors		
Mean Link Delay Averaging	0%	
NRR Drift Rate Correction	0%	
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	1	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	500,000	

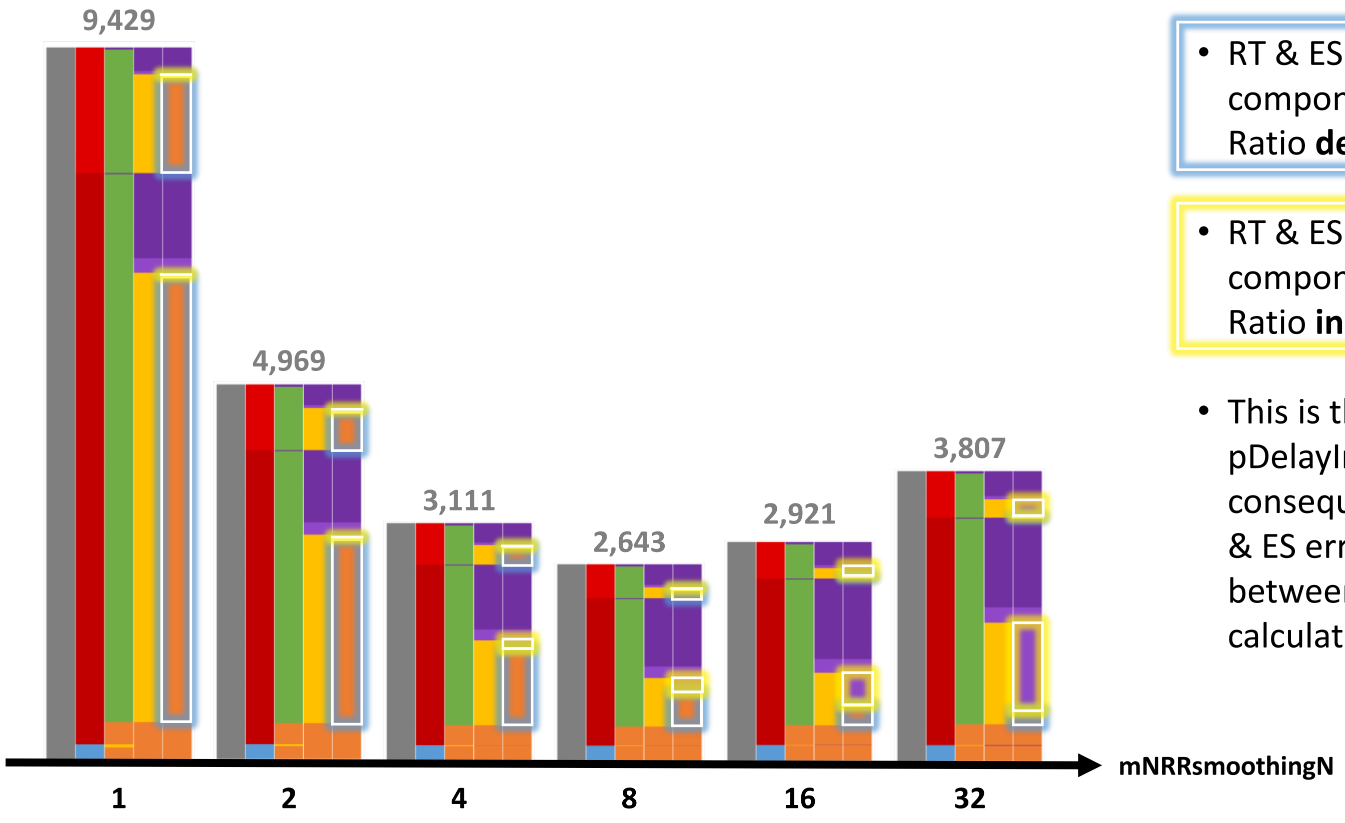
As residenceTime is reduced...

- RT error due to Rate Ratio error decreases.
- Rate Ratio error only marginally decreases, but the related error is proportional to Residence Time.



mNRRsmoothingN Sensitivity Analysis

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	0%	
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	VAR	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	500,000	



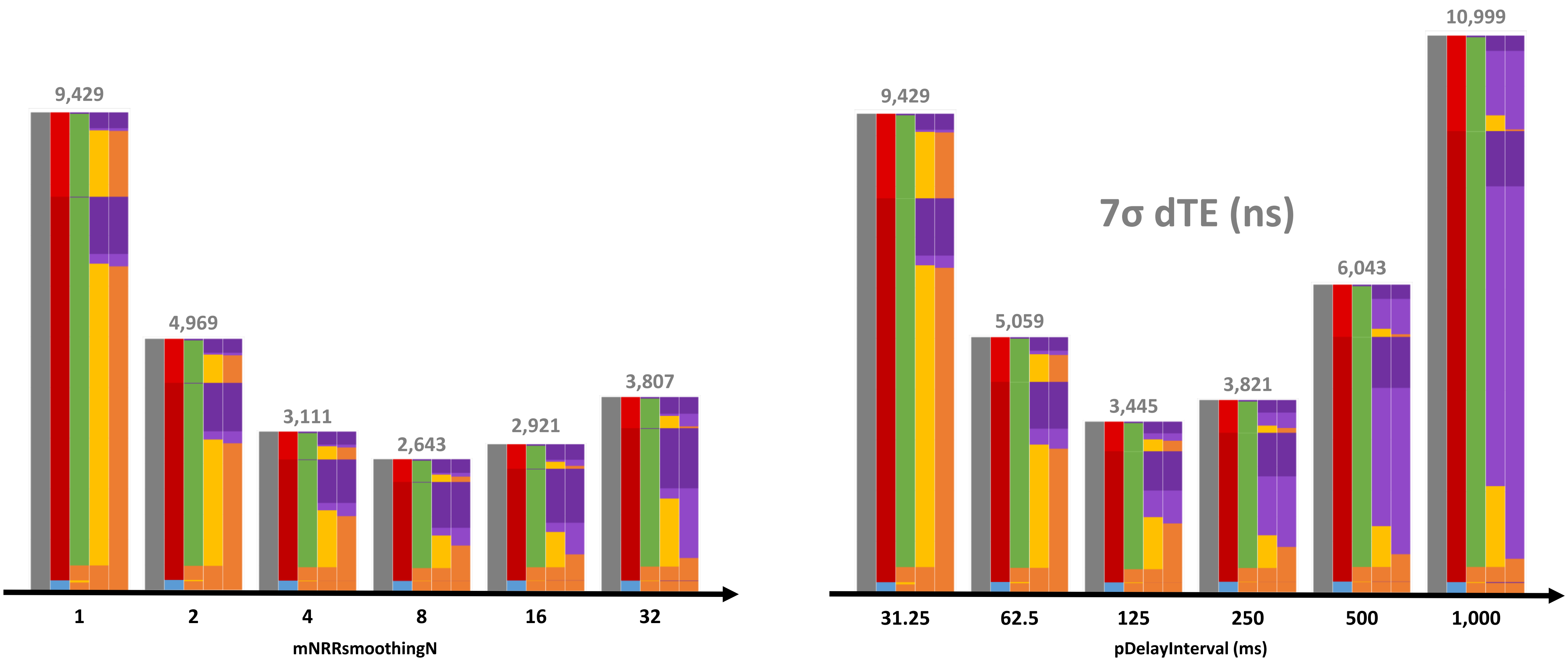
As *mNRRsmoothingN* is increased…

- RT & ES errors due to **Timestamp** component of $mNRR_{error}$, via Rate Ratio **decrease**.

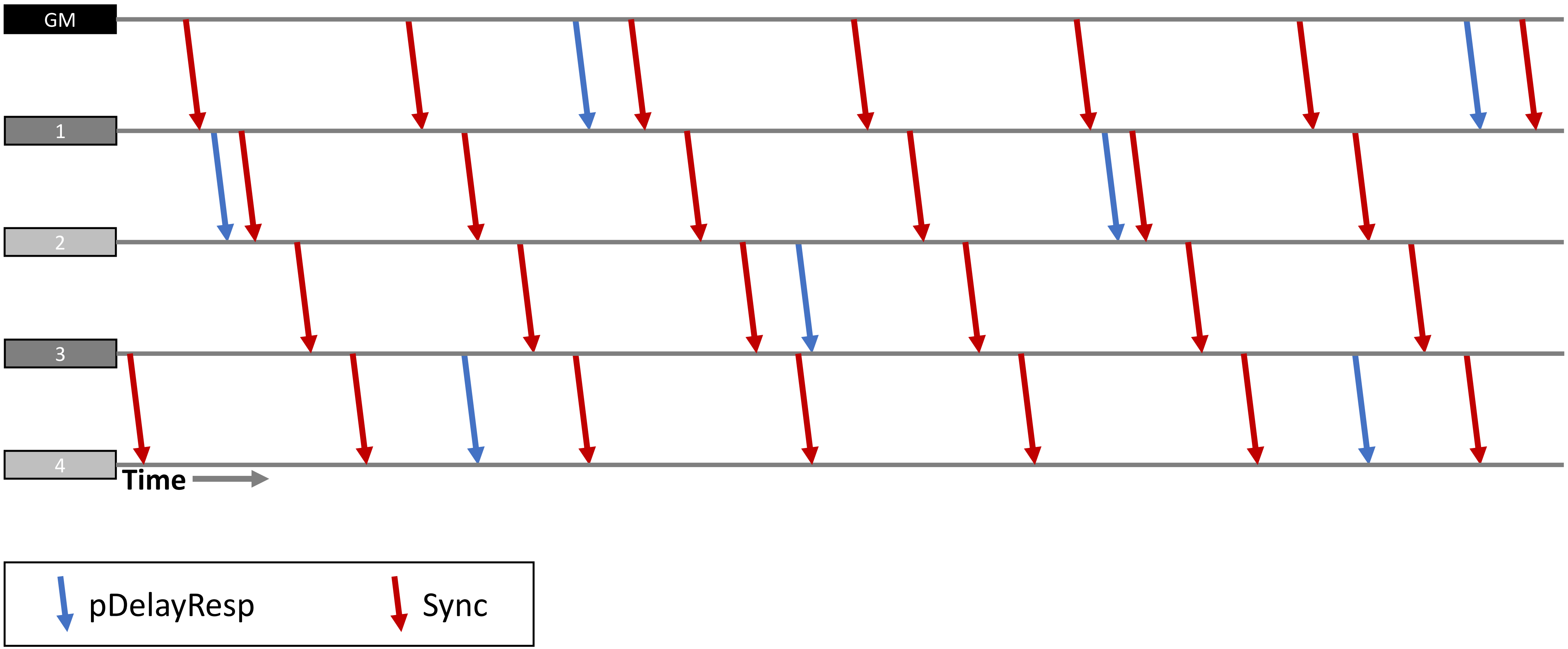
- RT & ES errors due to **Clock Drift** component of mNRRerror, via Rate Ratio **increase**.

- This is the opposite as for reducing pDelayInterval, but without the consequence of also increasing RT & ES errors due to **Clock Drift** between measurement of NRR and calculation of RR

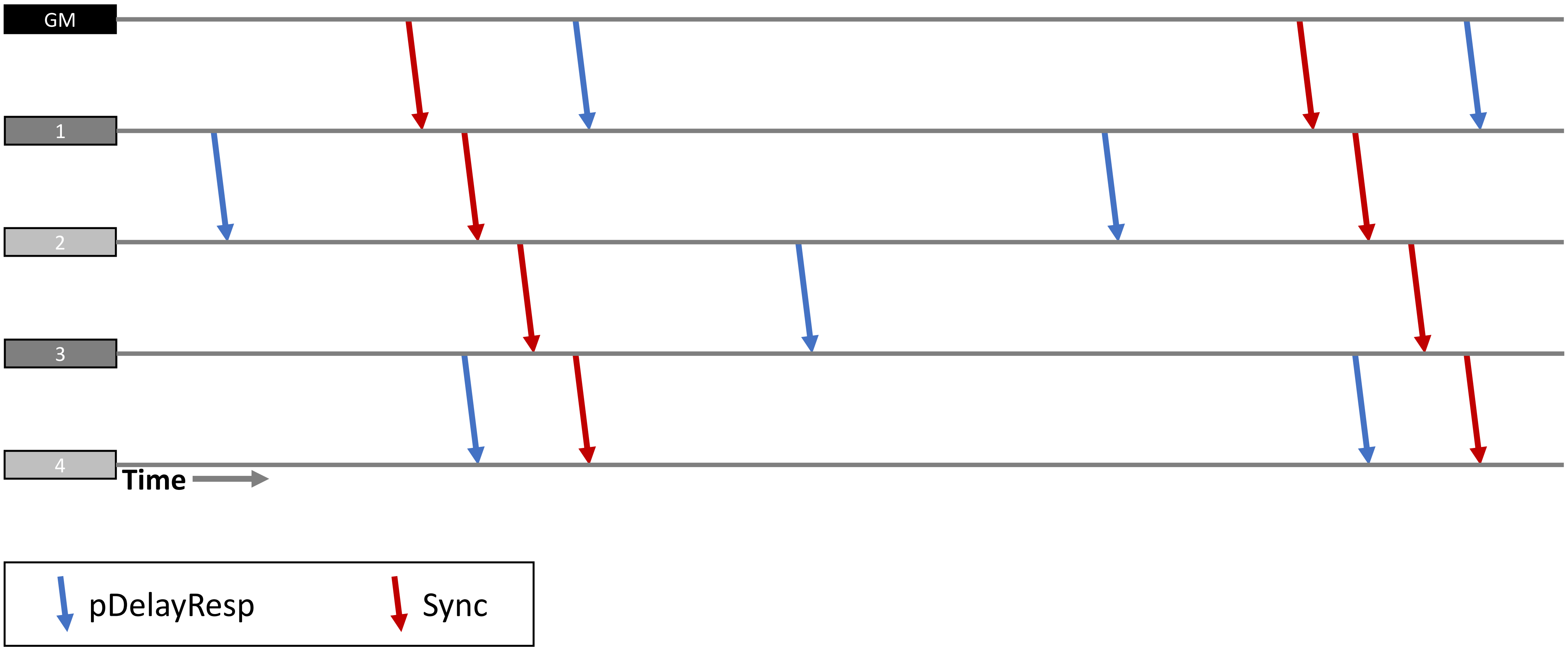
mNRRsmoothingN vs *pDelayInterval*



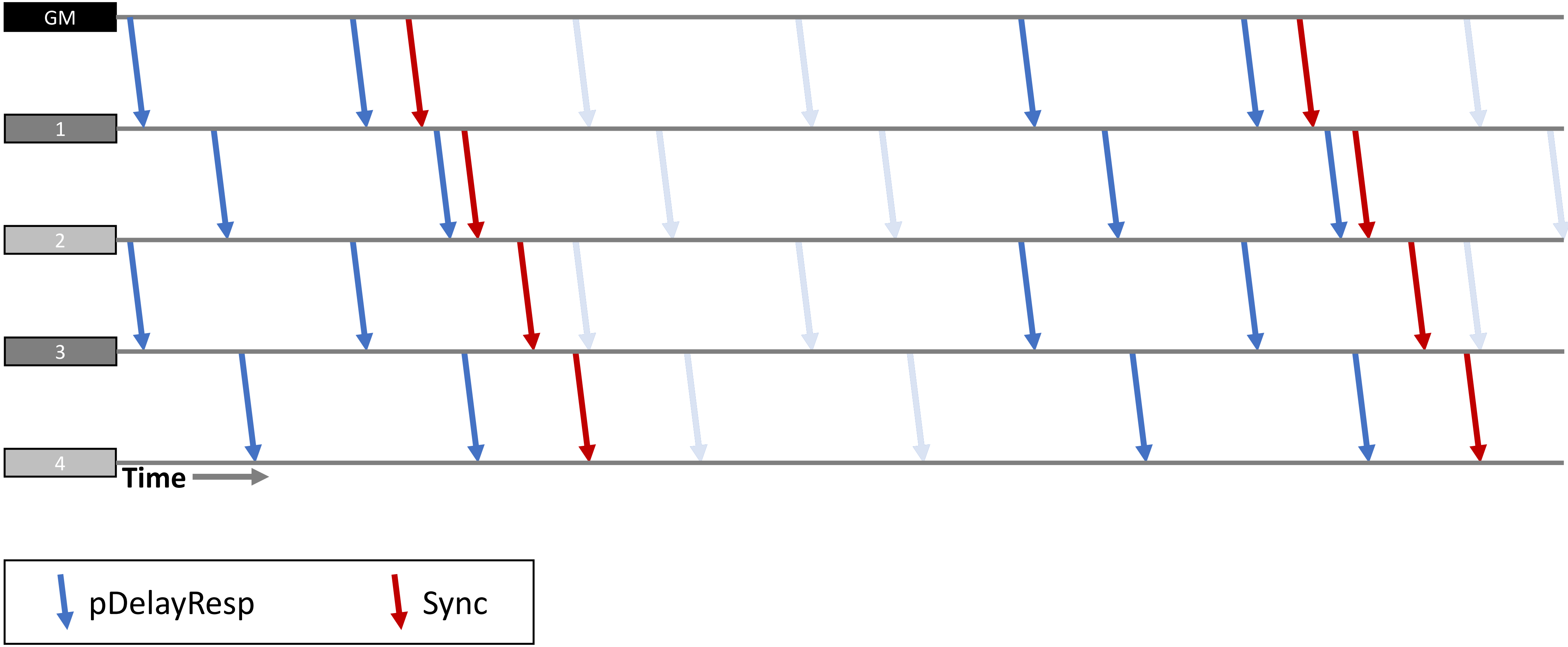
pDelayInterval > Sync Interval



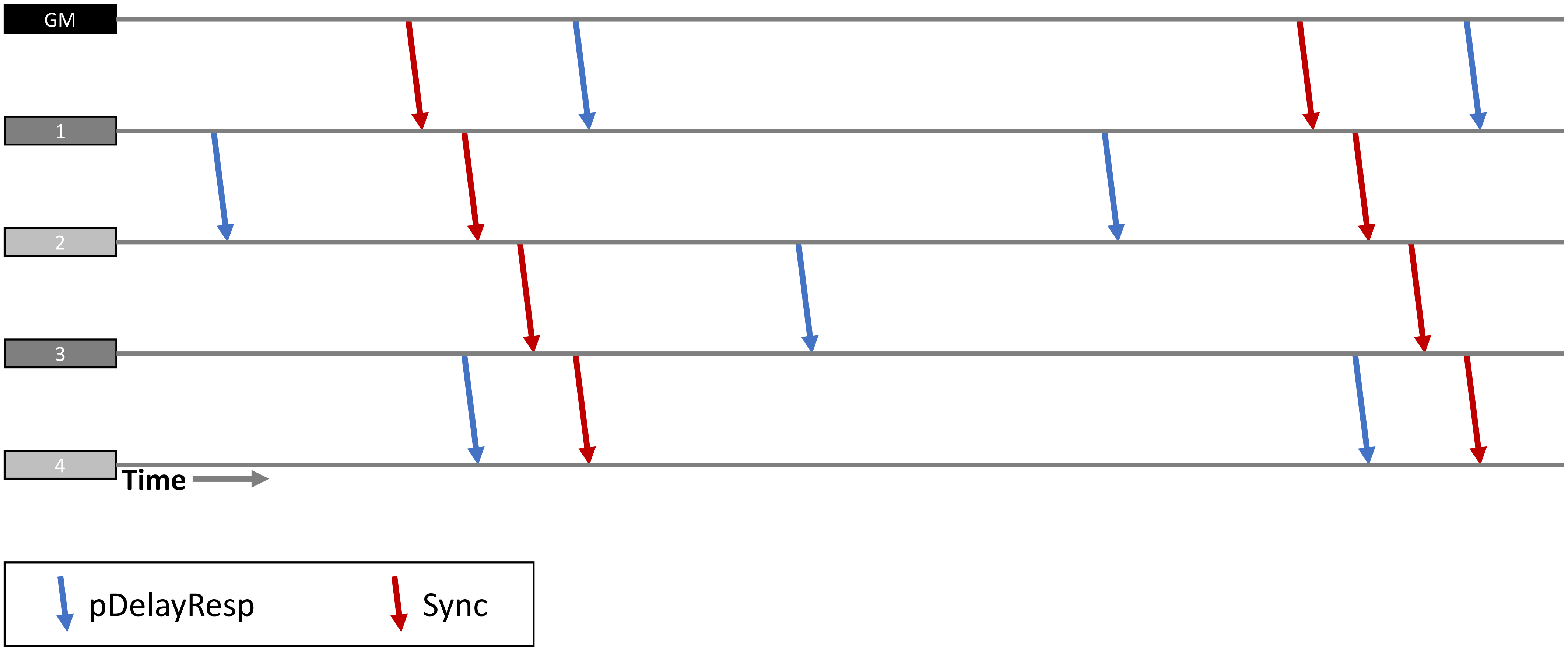
pDelayInterval = Sync Interval



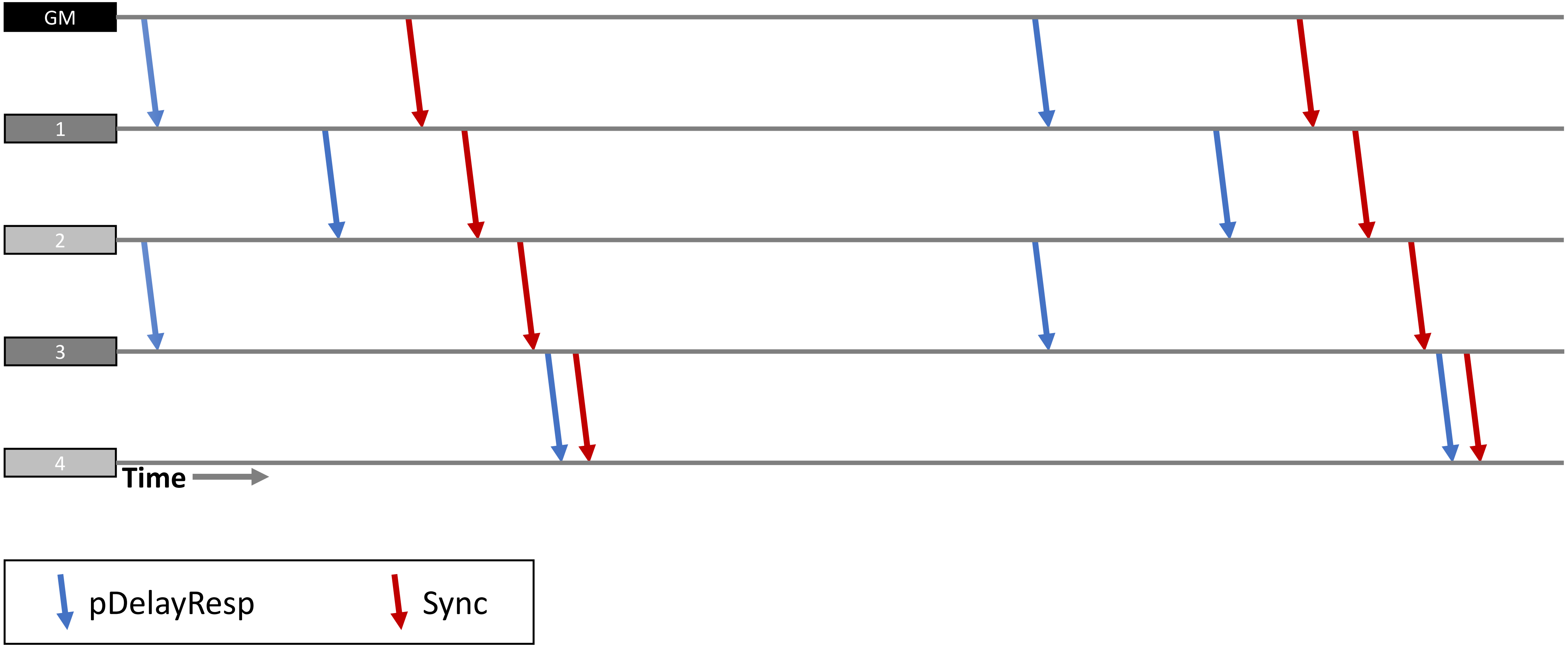
pDelayInterval < Sync Interval



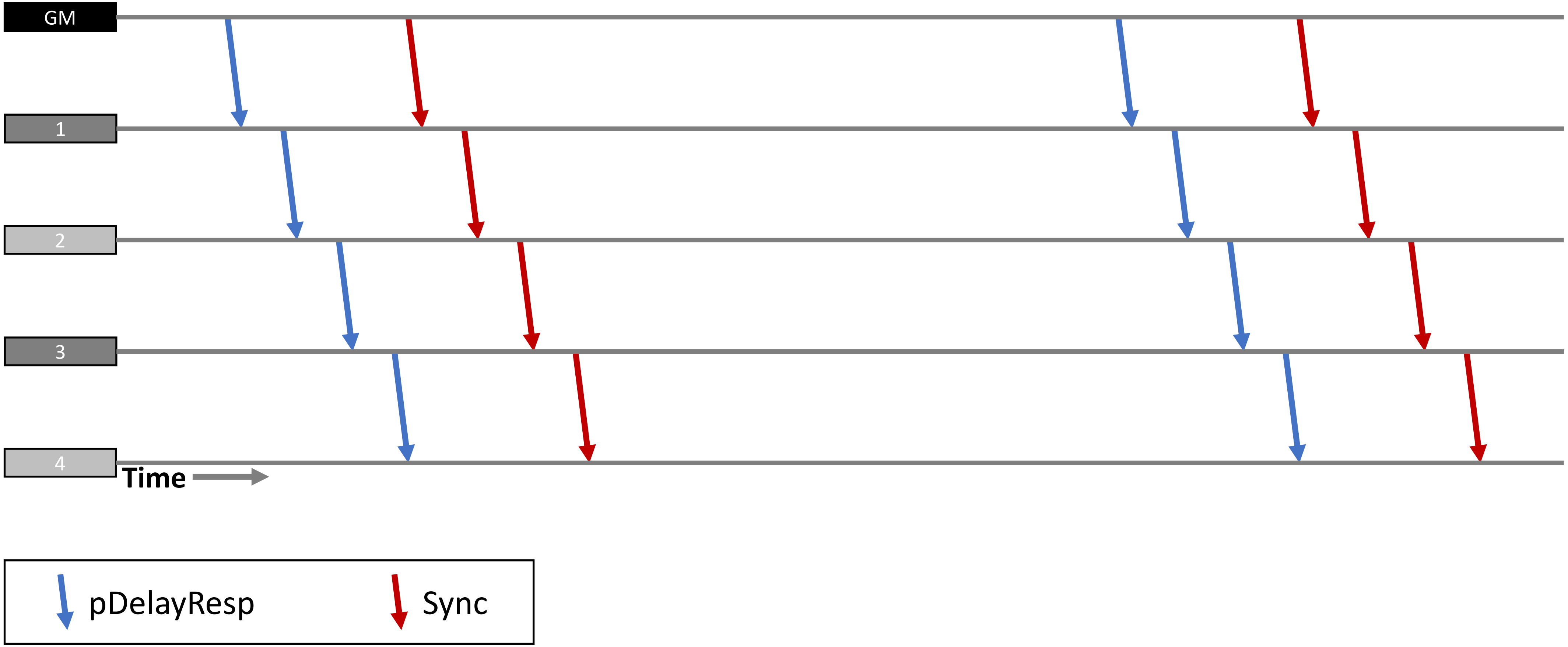
Align pDelayResp & Sync? – No Alignment



Align pDelayResp & Sync? Better Alignment



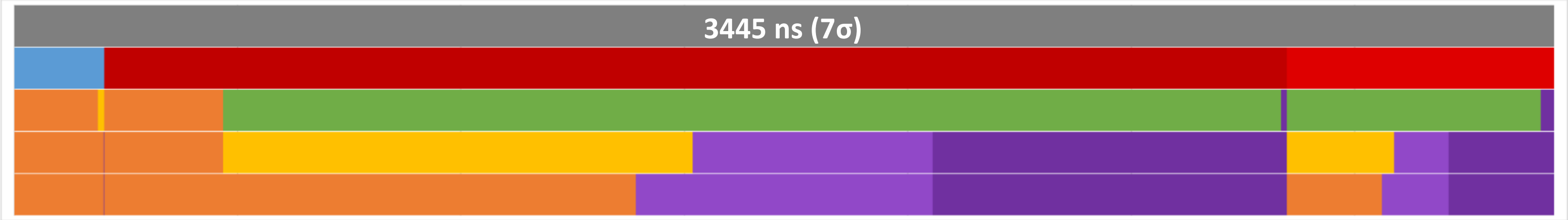
Align pDelayResp & Sync? Deterministic



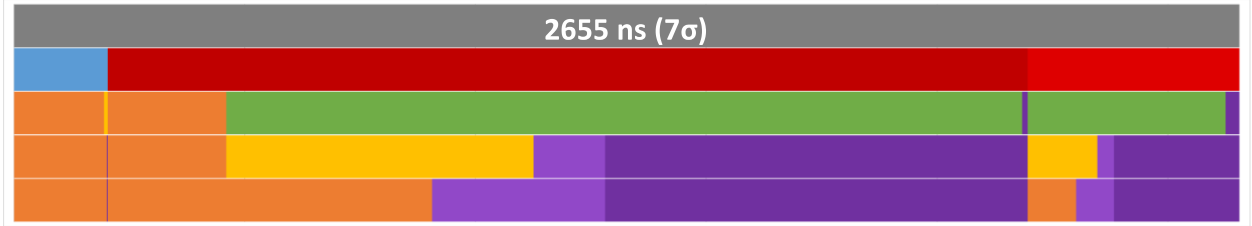
Align pDelayResp & Sync? Examples

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	125	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	0%	
RR Drift Rate Error Correction	0%	
pDelayResp → Sync Type (Uniform)	1	
pDelayResp → Sync Max	VAR	
pDelayResp → Sync Min	VAR	
pDelayResp → Sync Target	10	ms
mNRR Smoothing N	1	
mNRR Smoothing M	1	
Configuration		
Hops	100	
Runs	500,000	

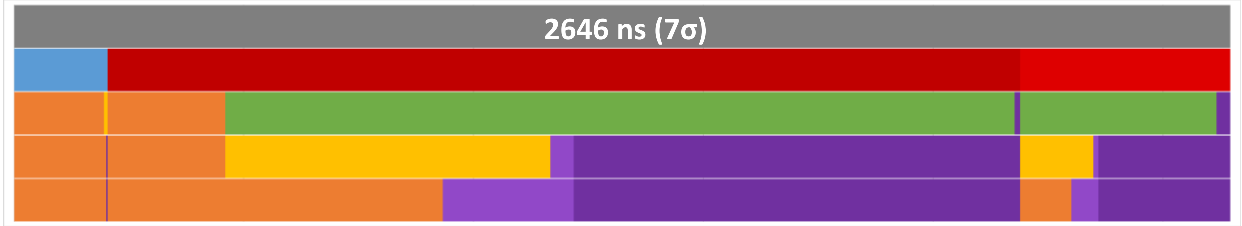
Min: 0%
Max: 100%



Min: 0%
Max: 25%



Min: 13%
Max: 15%



$RT_{errorRR\_CD\_NRR2sync}$ and $ES_{errorRR\_CD\_NRR2sync}$ both decrease in the third case, even though the average $T_{pdelay2pdelay}$ is longer. What is going on?

(Note that this reduction isn't reflected fully in dTE as it is swamped by other errors.)

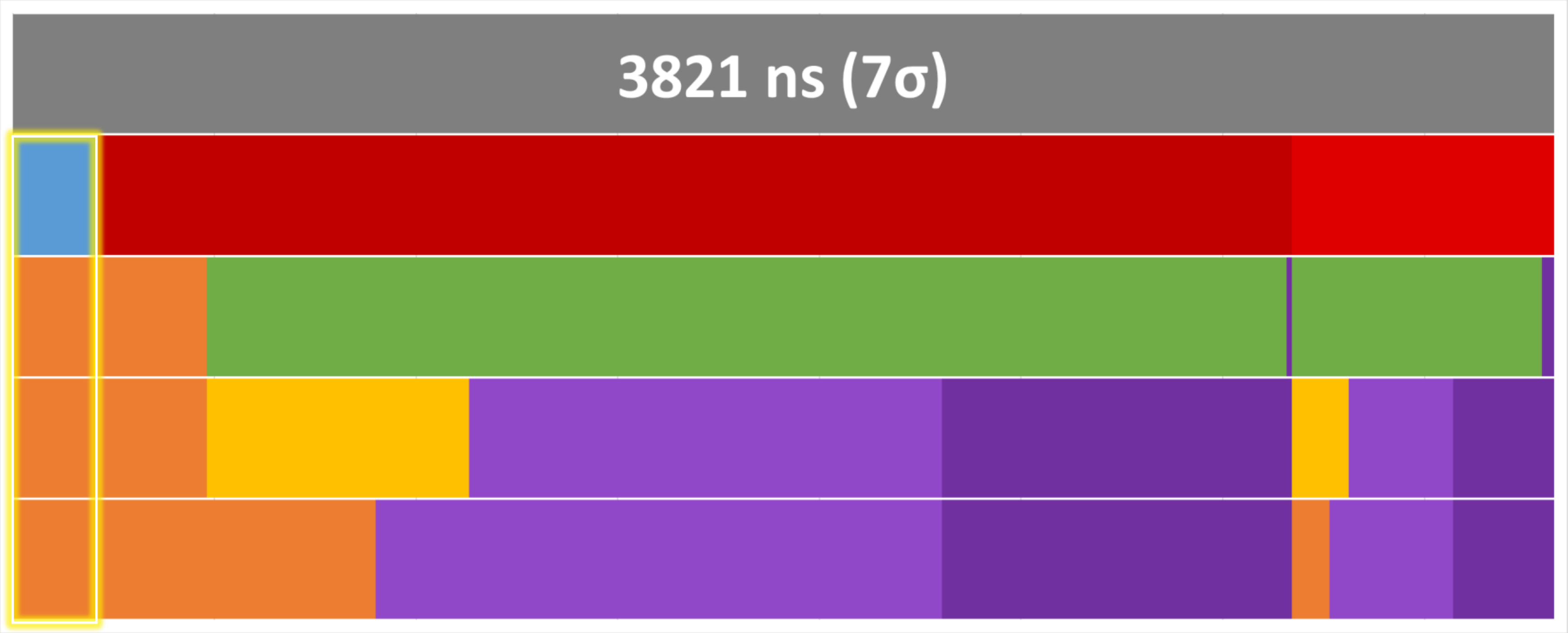
Aligning pDelayResp & Sync – Implications

- Absolute magnitude of the delay between measurement of mNRR and use during RR calculation is **not** the key factor
- The equations for the relevant errors are...

$$mNRR_{errorCD\_X} = \frac{T_{pdelay2pdelay}(\boldsymbol{clockDrift_n - clockDrift_{n-1}})}{2 \times 10^3} \qquad RR_{errrorCD\_NRR2Sync\_X} = \frac{T_{mNRR2Sync}(\boldsymbol{clockDrift_n - clockDrift_{n-1}})}{10^3}$$

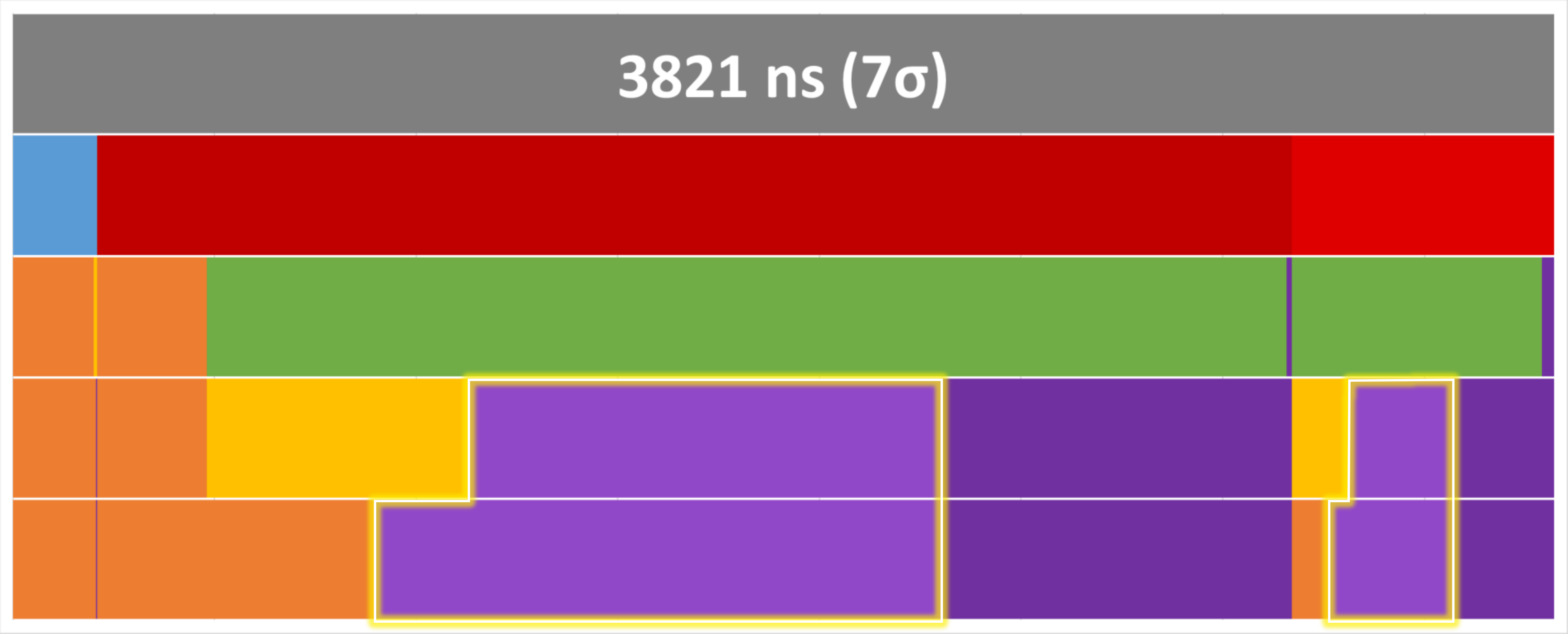
- If $T_{pdelay2pdelay}$ and $T_{mNRR2sync}$ are identical for node n-1 and node n, and clock drift at node n-1 is constant (i.e. doesn't change), then the total error contribution due to clock drift at from n (for these errors) is zero.
- Absolute magnitudes of $T_{pdelay2pdelay}$ and $T_{mNRR2sync}$ are not as important as variability. Stable values for both results in maximum cancellation of errors in RR during the Sync process.
  - Note that this statement starts to break down if the intervals are large enough for significant changes in Clock Drift.
- Reducing the variability of $T_{mNRR2sync}$ **and** $T_{pdelay2pdelay}$ is therefore a valid approach to error reduction.
  - Reduce $T_{mNRR2sync}$: align pDelayResp with Sync
  - Reduce $T_{pdelay2pdelay}$: reduce variability of period between pDelayResp messages
  - Doing both at the same time may require reducing the variability of the period between Sync messages.

MLDErrCor Effect



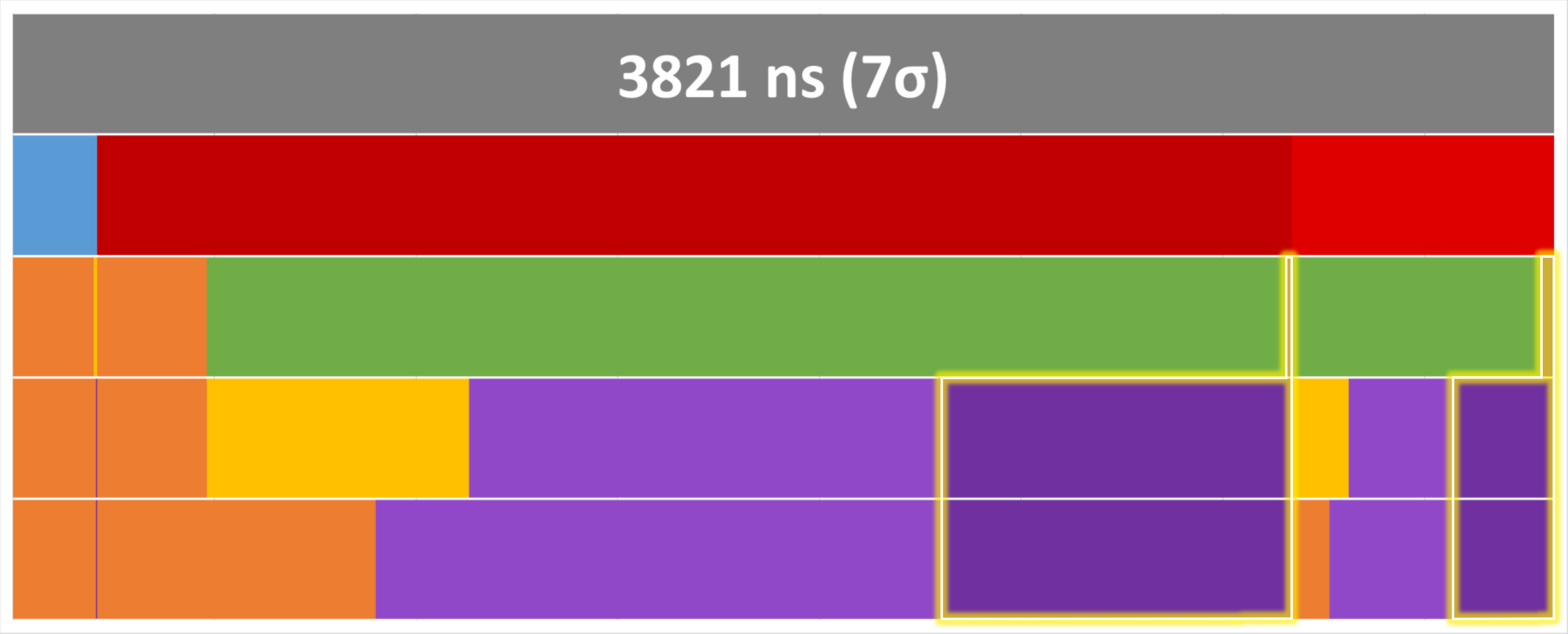
Decreases MLD_{error}

NRRdriftRateErrorCor Effect



Decreases $RT_{errorRR\_NRR\_CD}$, $RT_{errorRR\_CD\_NRR2sync}$, $ES_{errorRR\_NRR\_CD}$, $ES_{errorRR\_CD\_NRR2sync}$

RRdriftRateErrorCor Effect



Decreases $RT_{errorRR\_CD\_RR2sync}$, $RT_{errorCDdirect}$, $ES_{errorRR\_CD\_NRR2sync}$, $ES_{errorCDdirect}$

Complexities

Approach	Complexity / Level of Challenge
Optimise pDelayInterval	Required. Low values may be problematic. (125ms would be OK?)
Optimise syncInterval	
Optimise residenceTime	Required. Values below 10ms may be problematic.
mNRRsmoothingN	Low complexity. Optimal value depends on effectiveness of other measures.
Align pDelayResp and Sync	Unknow complexity; depends on mechanism. **Investigation required**.
Reduce variability of $T_{pdelay2pdelay}$	
Reduce variability of $T_{sync2sync}$	
Mean Link Delay Averaging	Low complexity. Averaging is required for Mean Link Delay to be useful.
NRR Drift Tracking & Compensation	Medium complexity, assuming tracking is accomplished via looking at past mNRR measurements and clock drift linearity is assumed. **Efficacy and effect of sudden changes in drift rates remains to be determined**.
RR Drift Tracking & Compensation	

Time Sync Ad Hoc
Next Steps

60802 Time Sync Ad Hoc – Next Steps

Key:
Can progress now
Contribution required
Dependant on other items

- **Messaging & Algorithms**
  - **Align pDelay & Sync messaging; reduce variability of $T_{pdelay2pdelay}$ & $T_{sync2sync}$** – investigation of possible mechanism
    - Contributions requested
  - **NRR & RR drift measurement & compensation** – Monte Carlo & Time Series simulations to determine efficacy and robustness
- **Clock Filters & Control Loops**
  - **Continued discussion** based on latest Time Series simulation results
- **Sync Message Timestamping** (using synced ClockSlave to timestamp)
  - **Assessment** (**simulations**?) based on results of Clock Filters & Control Loops discussion.
- **Rate Ratio Measurement**
  - **Analysis** of Rate Ratio measurement via Sync messaging, similar to [2]. Subsequent Monte Carlo **simulation** and assessment.
- **Normative vs. Informative**
  - **Discussion** on normative requirements for error generation
    - Possible **discussion** of normative requirements for error tolerance if NRR and/or RR drift measurement & compensation is adopted.
  - Everything else is informative. Some will be obvious. Others may require **discussion**.
- **Unified Proposal**
  - Dependant on progress of above subject areas.

Thank you!