60802 Dynamic Time Sync Error – Additions – Error due to drift during Sync messaging – Potential Contribution

David McCall (Intel)

March 2022 IEEE 802 – 802.1 TSN – IEEE/IEC 60802

Abstract

- Industrial Automation Systems require microsecond-accurate time across long daisy-chains of devices using IEEE Std. 802.1AS™-2020 as specified by IEEE/IEC 60802.
- Simulated protocol and system parameters have thus far either been judged impractical or have failed to meet the time-accuracy requirement.
- An analysis of how errors accumulate suggested that a Monte Carlo method analysis could support fast iteration of potential scenarios and deliver insights into cause and effect. See...
  - 60802-McCall-et-al-Time-Sync-Error-Model-0921-v03.pdf
  - 60802-McCall-Stanton-Time-Sync-Error-Model-and-Analysis-2021-11-v02.pdf
- In this contribution we:
  - Describe addition of pDelay variation, and "End Station" error (with Sync Interval variation) to analysis
  - Discuss error due to clock drift during Sync messaging and potential to mitigate via algorithmic compensation
  - Present basis for potential normative and informative contribution to next draft
  - ~~Present Monte Carlo analysis results to compare with upcoming Time Series simulation results~~

Content

- Background & Recap
  - Proposals from February
- Additions to the Monte Carlo analysis
  - pDelay variation
  - “End Station” error due to clock drift between Sync messages
    - Includes Sync interval variation
- Error due to drift during Sync messaging
  - Potential for algorithmic compensation
- Potential basis for normative and informative contribution to next draft
- ~~Monte Carlo analysis results for comparison with upcoming Time Series simulation results~~ (will be generated prior to Geoff’s presentation)

Background & Recap

Proposed Next Steps

From February

- Time Series Simulations to validate Monte Carlo Analysis
  - Not necessarily with values we would want to use in practice. Main point is to ensure that Monte Carlo Analysis and Time Series Simulations match.
- More Monte Carlo Analysis to develop recommendations
  - Time Series Simulations to validate
- Prepare spec contribution for March Plenary
  - Likely present to 60802 group before then to get guidance on key questions

Proposed Time Series Simulations – Detail

From February

Experiment	Reason	Errors			Parameter			Correction Fa	
		Clock Drift Model – 40°C ↔ +85°C Hold for 1min at Each (Each node's position in cycle distributed at random across 100% of Cycle)	Timestamp Granularity (ns)	Dynamic Timestamp Error (±ns)	pDelay Interval (ms)	Residence Time (ms)	pDelay Turnaround Time (ms)	Mean Link Delay Averaging	mNRR Smoothing Factor N
A	Baseline with previous assumptions	Ramp Rate 1°C / s (Cycle of 310 s)	8	4	31.25	1	1	Off	1
B	Verify optimised pDelayInterval		8	4	1000	10	10		
C					250	10	10		
D					31.25	10	10		
E	Verify effect of reduced Timestamp Error (reduced DTE when pDelay Interval is low, i.e. 31.25ms)		4	2	31.25	10	10		
F	Verify effect of reduced Clock Drift (reduced DTE when pDelay Interval is high, i.e. 1000ms)	Ramp Rate 0.5°C / s Cycle of 560s	8	4	1000	10	10		

Timestamp Granularity and Dynamic Timestamp Error are uniform distributions unless otherwise stated

Sync Interval: 125ms

pDelay Interval variation is +0-30% with uniform distribution

Sync Interval variation is ±10% with 90% probability with gamma distribution

Added to Monte Carlo analysis

Note: 8ns Timestamp Granularity in Time Series Simulation is equivalent to ±4ns Timestamp Granularity Error in Monte Carlo Analysis

1°C / s temperature ramp rate is the equivalent of ±1.5 ppm/s clock drift rate in Monte Carlo Analysis

Up from ±0.6 ppm/s previously

No difference between base (PHY related) propagation delay for pDelay and Sync messages

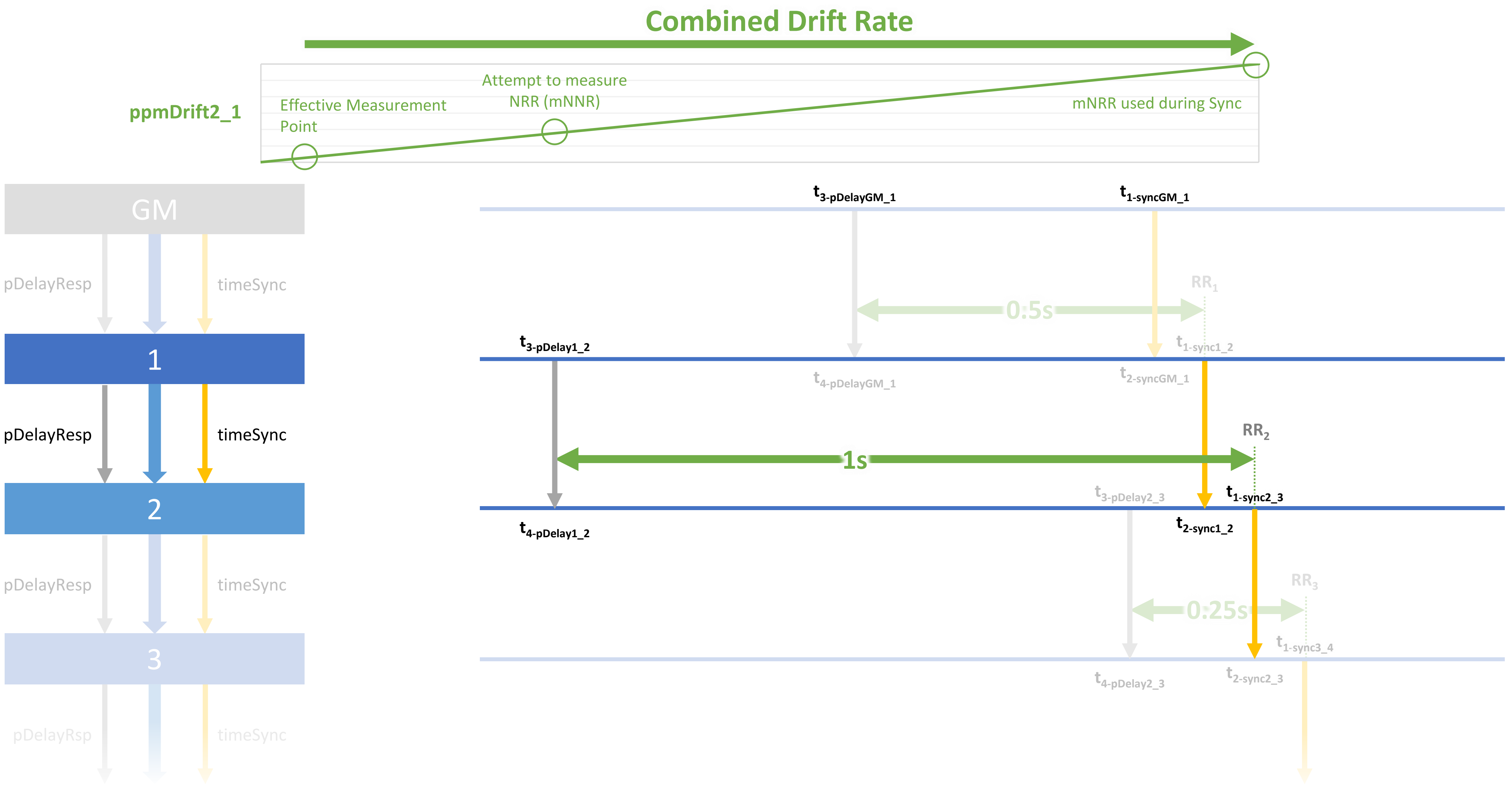
Additions to Monte Carlo Analysis

pDelay Variation

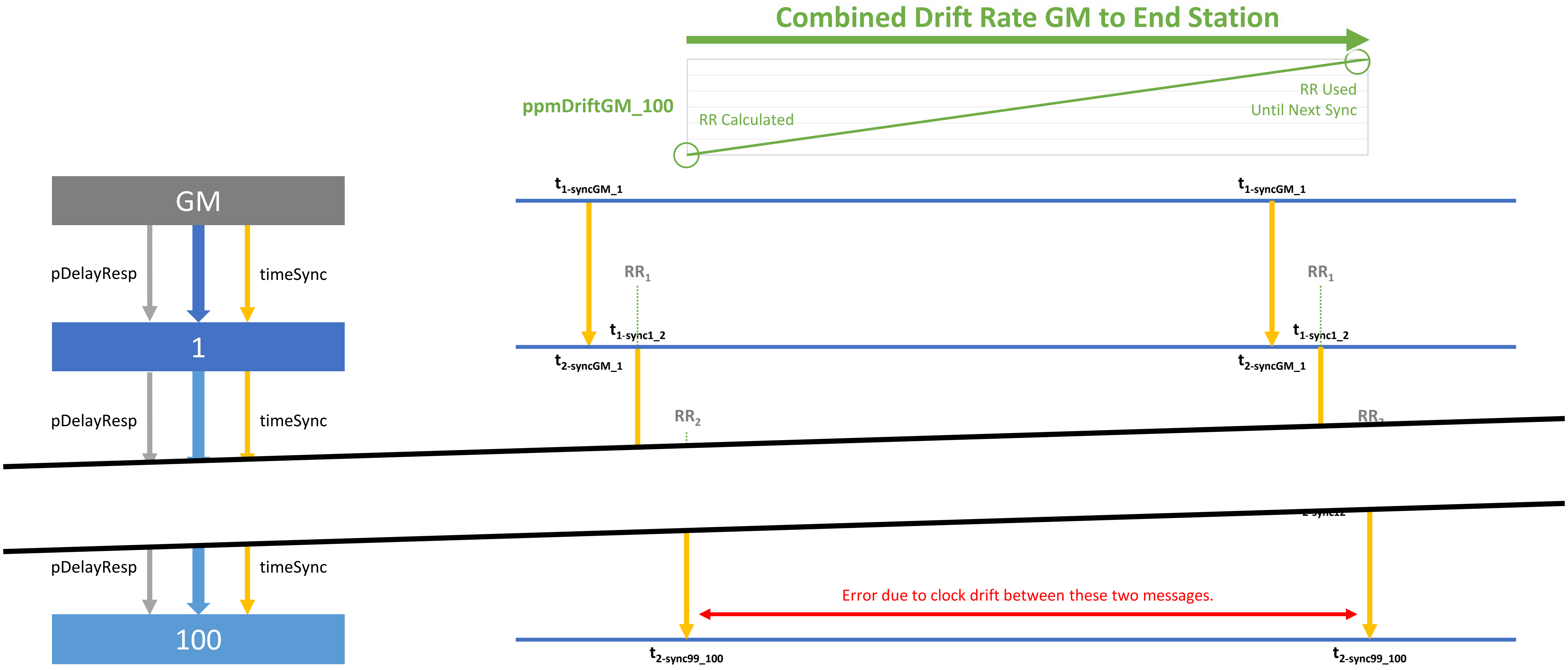
- Previously: pDelay was always the nominal value
- Now: pDelay varies between nominal value and +30% with uniform distribution
- Mainly effect: increased error due to clock drift between pDelay messaging and Sync messaging (average +15%)
  - Previously: delay was modelled as uniform distribution between 0 and pDelay
  - Now: modelled as...
    (uniform distribution 0 to 1) x (uniform distribution pDelay to pDelay x 1.3)
- Secondary effect: increased mNRR error due to clock drift between pDelay messages (average +15%; higher values of mNRRsmoothingN mean the distribution is closer to gaussian)
  - Separate variable pDelay intervals are generated vs. those used for pDelay to Sync messaging
  - For mNRRsmoothingN > 1 additional variable pDelay intervals are generated, i.e. a single pDelay interval isn't just multiplied by mNRRsmoothingN

“End Station” Error

- Previously: only errors as part of Sync messaging were modelled
  - Including errors from that feed into Sync messaging, e.g. Link Delay and NRR measurement
- Now: include errors due to clock drift between Sync messages



Combined Drift Rate
ppmDrift2_1
Effective Measurement Point
Attempt to measure NRR (mNNR)
mNRR used during Sync
GM
1
2
3
pDelayResp
timeSync
pDelayResp
timeSync
pDelayResp
timeSync
pDelayRsp
timeSync
$t_{3\text{-pDelayGM\_1}}$
$t_{1\text{-syncGM\_1}}$
RR_1
0.5s
$t_{3\text{-pDelay1\_2}}$
$t_{1\text{-sync1\_2}}$
$t_{4\text{-pDelayGM\_1}}$
$t_{2\text{-syncGM\_1}}$
RR_2
1s
$t_{3\text{-pDelay2\_3}}$
$t_{1\text{-sync2\_3}}$
$t_{4\text{-pDelay1\_2}}$
$t_{2\text{-sync1\_2}}$
RR_3
0.25s
$t_{1\text{-sync3\_4}}$
$t_{4\text{-pDelay2\_3}}$
$t_{2\text{-sync2\_3}}$

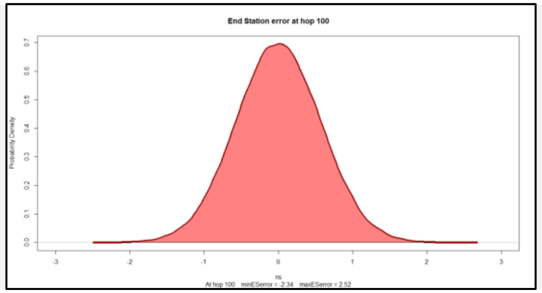
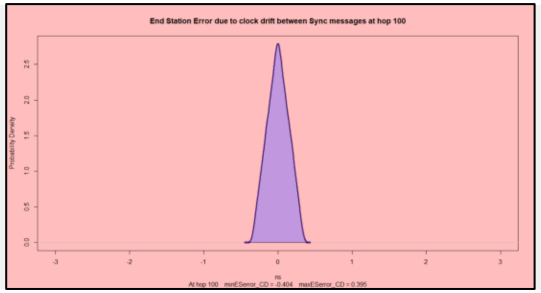
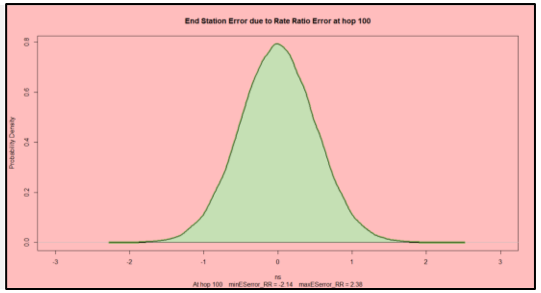
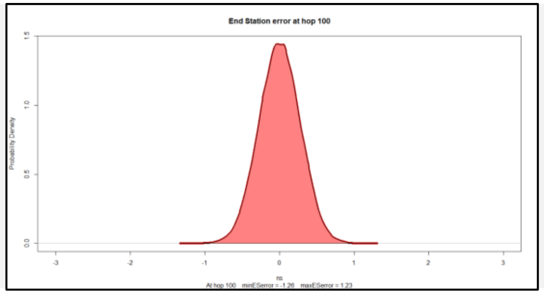
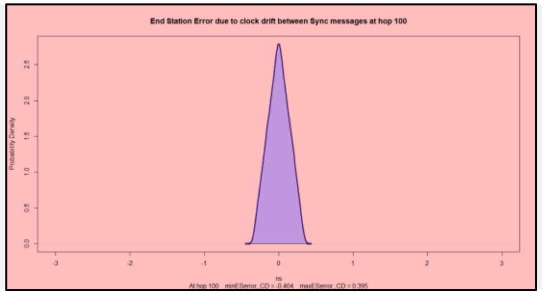
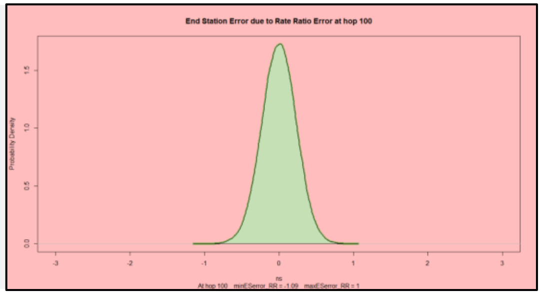


Combined Drift Rate GM to End Station
ppmDriftGM_100
RR Calculated
RR Used
Until Next Sync
GM
pDelayResp
timeSync
1
pDelayResp
timeSync
pDelayResp
timeSync
100
$t_{1\text{-syncGM\_1}}$
$t_{1\text{-syncGM\_1}}$
RR_1
RR_1
$t_{1\text{-sync1\_2}}$
$t_{1\text{-sync1\_2}}$
$t_{2\text{-syncGM\_1}}$
$t_{2\text{-syncGM\_1}}$
RR_2
Error due to clock drift between these two messages.
$t_{2\text{-sync99\_100}}$
$t_{2\text{-sync99\_100}}$

“End Station” Error

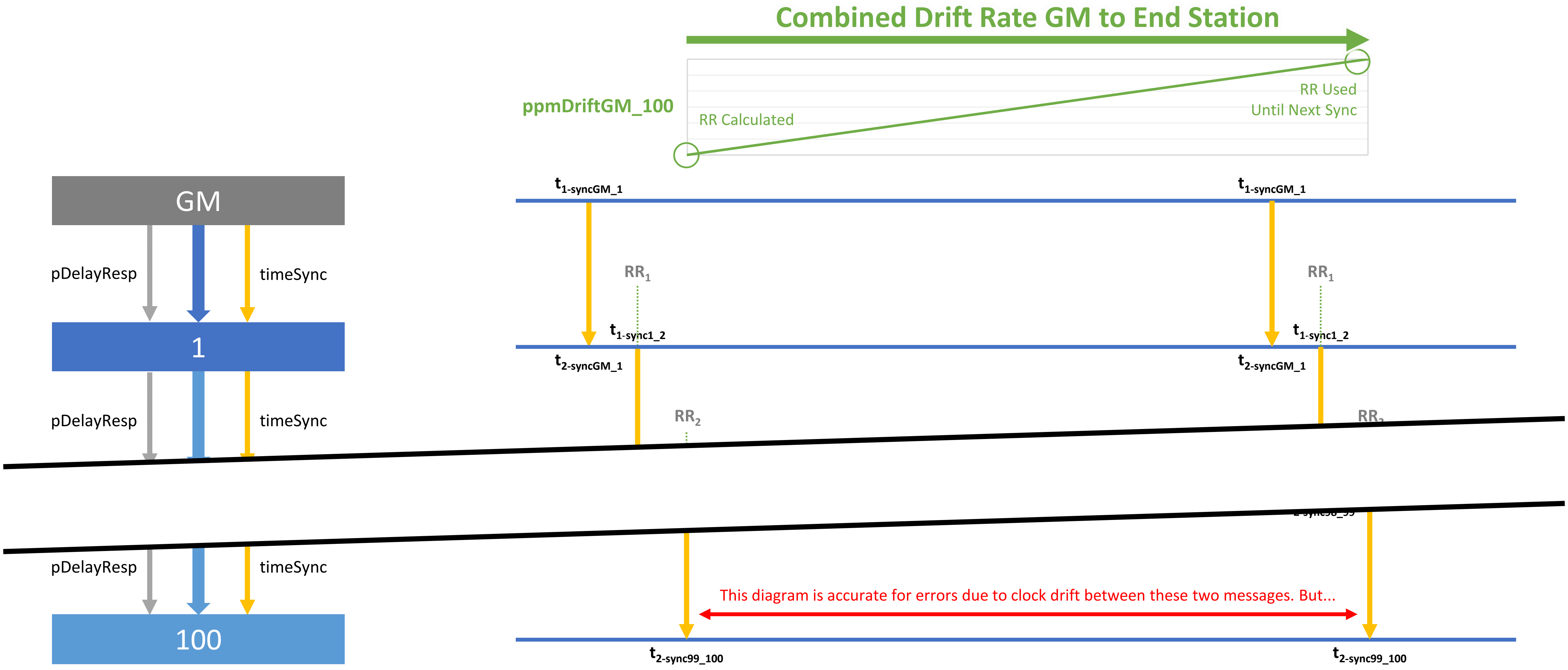
- Previously: only errors as part of Sync messaging were modelled
  - Including errors from that feed into Sync messaging, e.g. Link Delay and NRR measurement
- Now: include errors due to clock drift between Sync messages
- Two sources of error
  - Error in calculated Rate Ratio
  - Error due to clock drift between End Station Local Clock and GM

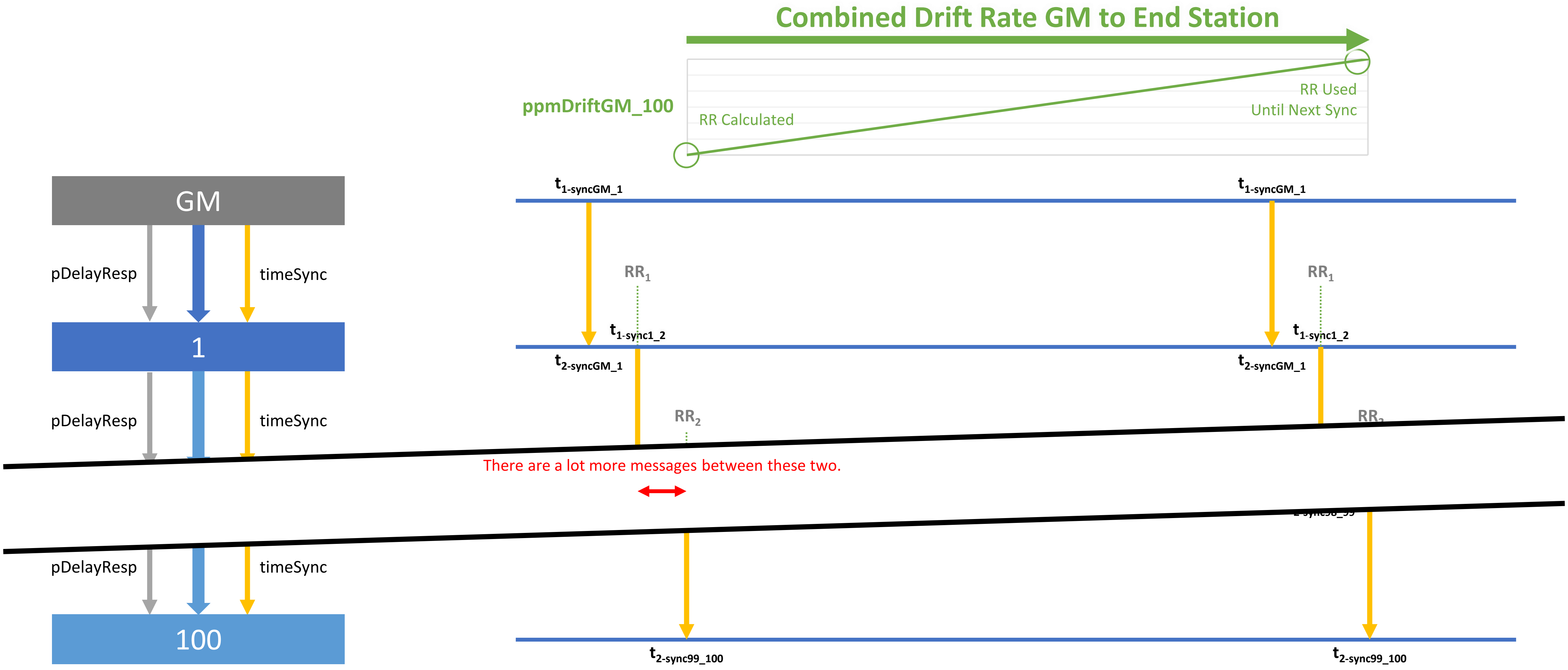
End Station Error at Hop 100

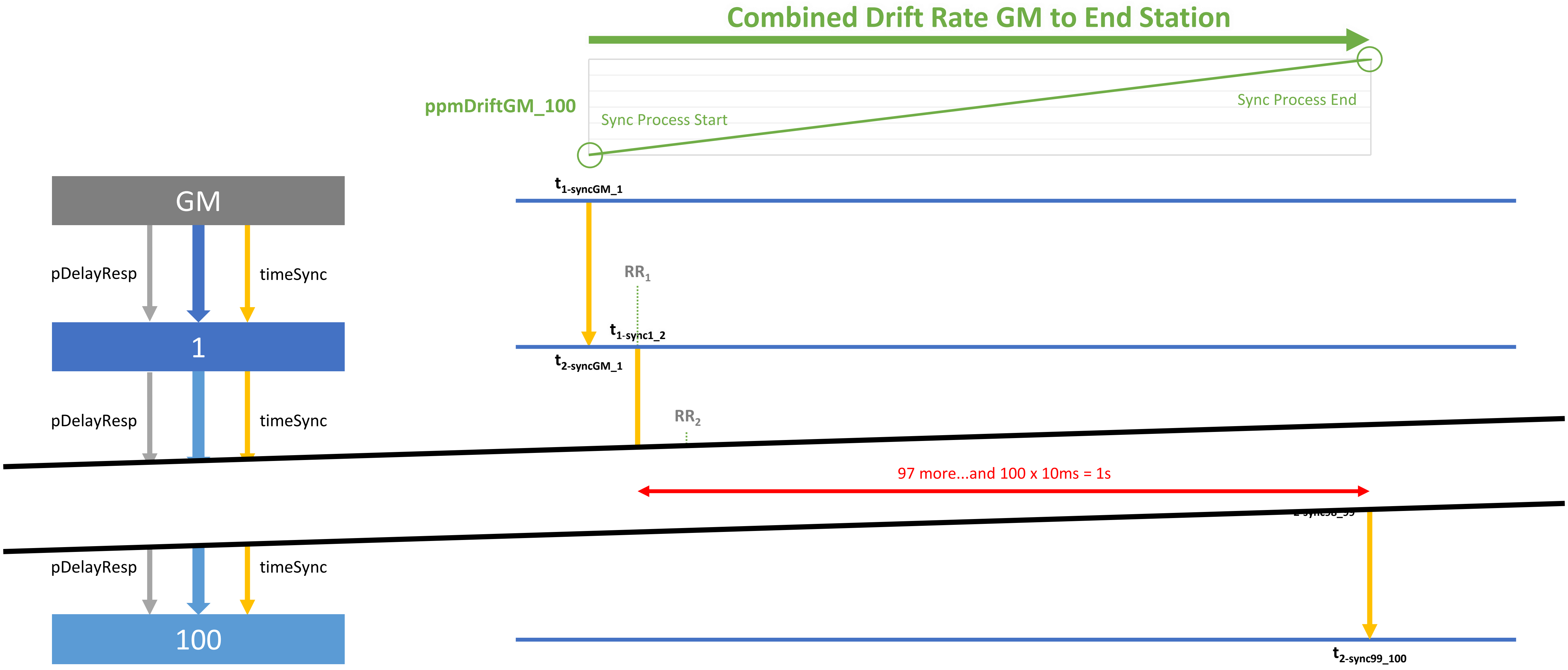
pDelay Interval	End Station Error	...due to Clock Drift	...due to Rate Ratio error
1000ms			
31.25ms			

- Small compared with other sources of error.

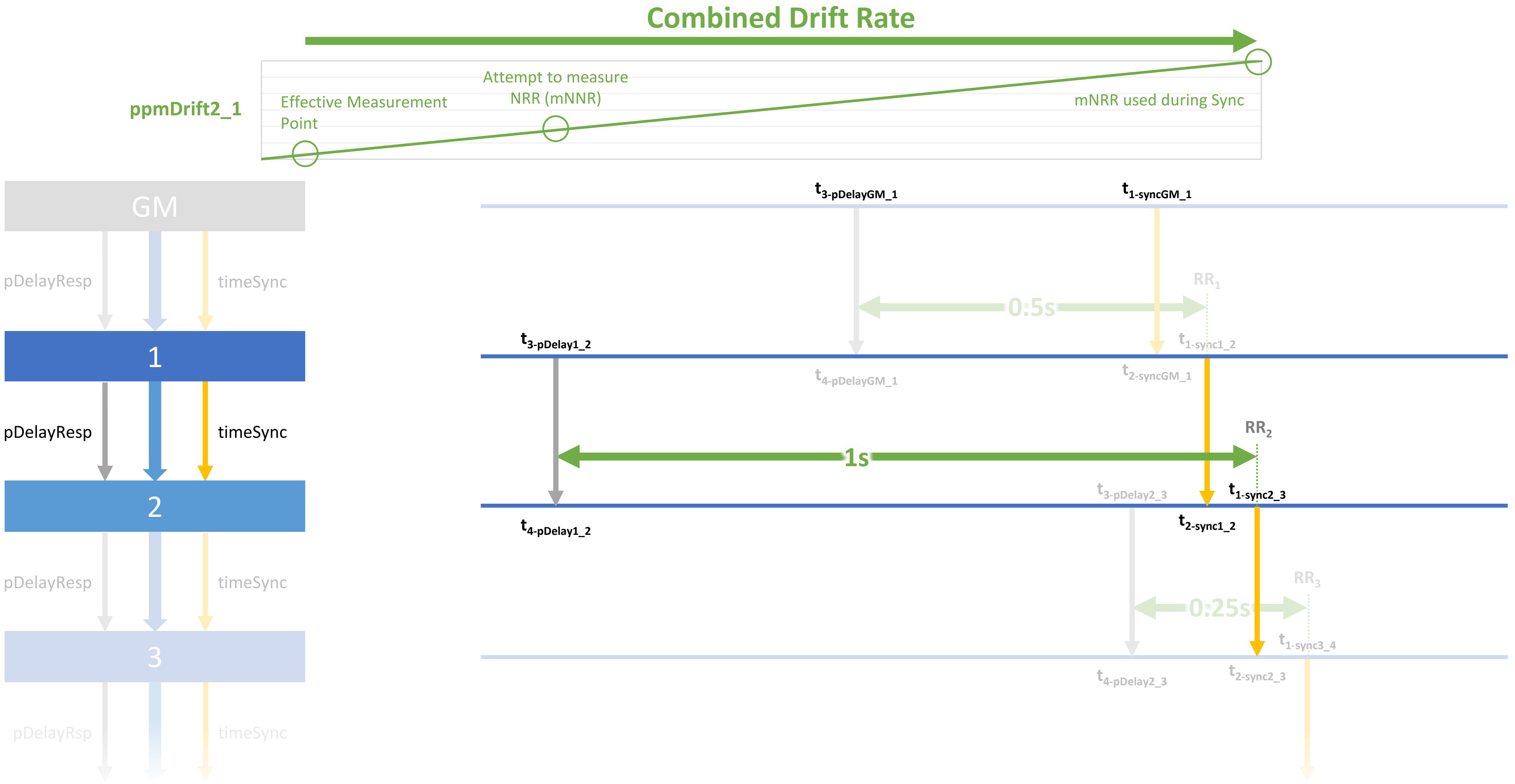
Error Due to Clock Drift During Sync Messaging


Combined Drift Rate GM to End Station
ppmDriftGM_100
RR Calculated
RR Used
Until Next Sync
GM
pDelayResp
timeSync
1
pDelayResp
timeSync
pDelayResp
timeSync
100
t1-syncGM_1
RR1
t1-sync1_2
t2-syncGM_1
RR2
t1-syncGM_1
RR1
t1-sync1_2
t2-syncGM_1
This diagram is accurate for errors due to clock drift between these two messages. But...
t2-sync99_100
t2-sync99_100

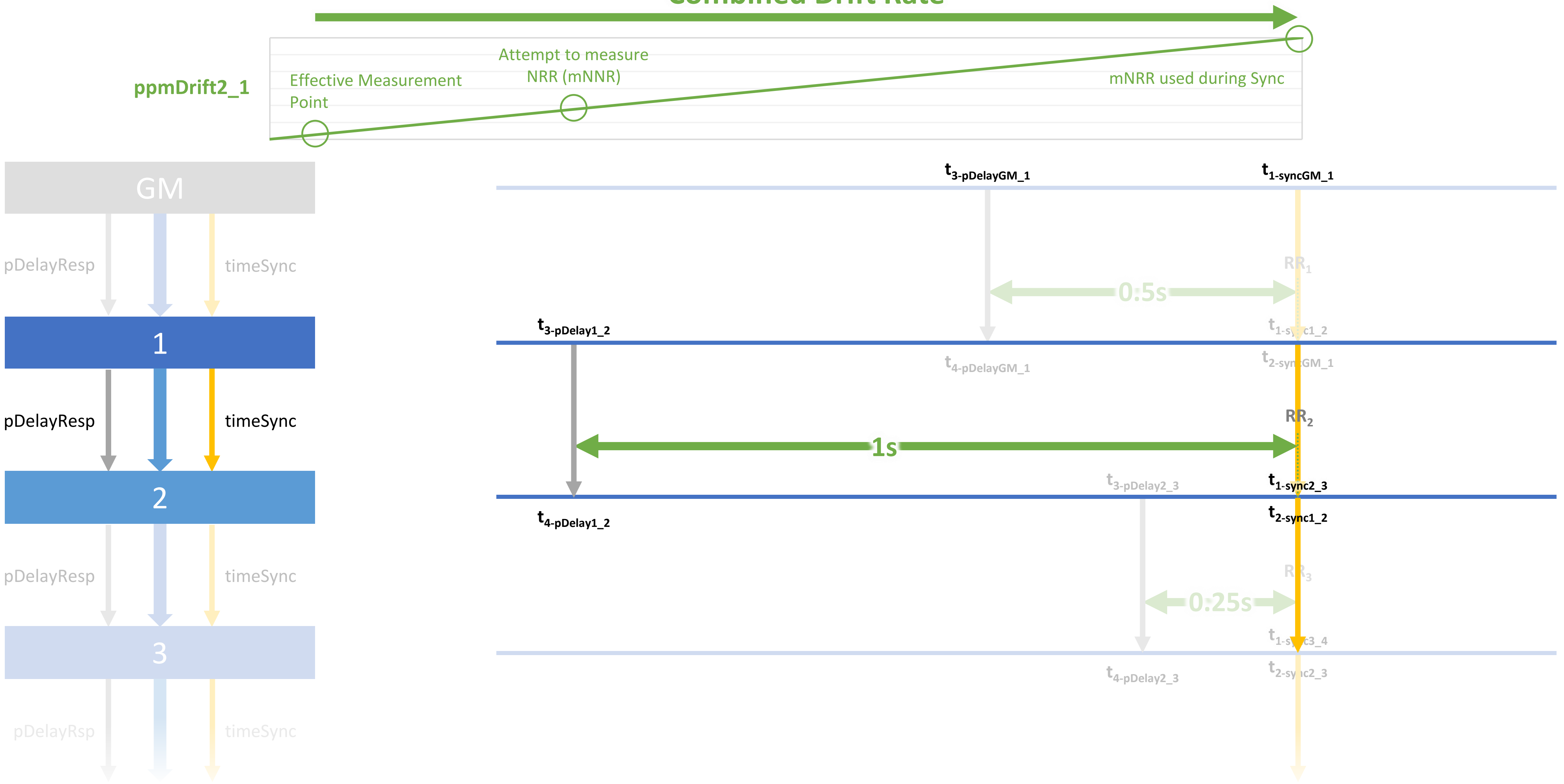

Combined Drift Rate GM to End Station
ppmDriftGM_100
RR Calculated
RR Used
Until Next Sync
GM
1
100
pDelayResp
timeSync
pDelayResp
timeSync
pDelayResp
timeSync
t1-syncGM_1
t1-syncGM_1
RR1
RR1
t1-sync1_2
t1-sync1_2
t2-syncGM_1
t2-syncGM_1
RR2
There are a lot more messages between these two.
t2-sync99_100
t2-sync99_100

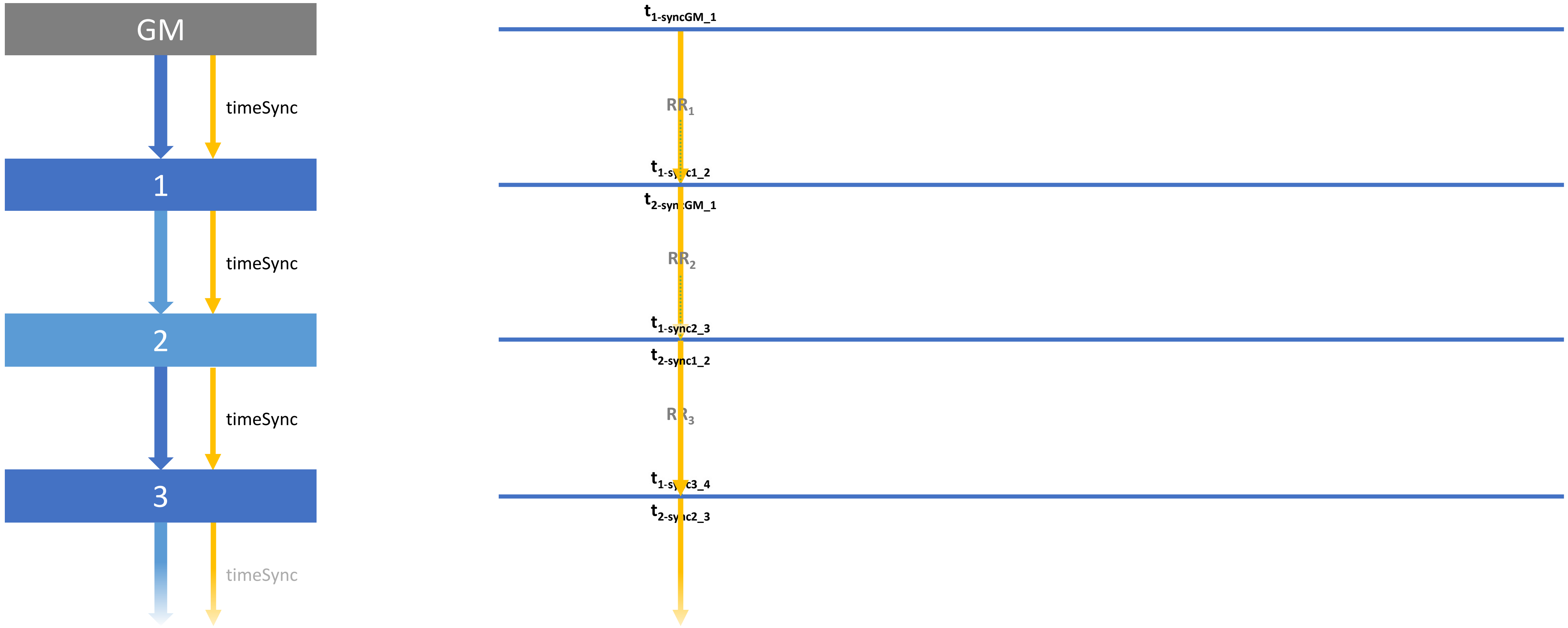


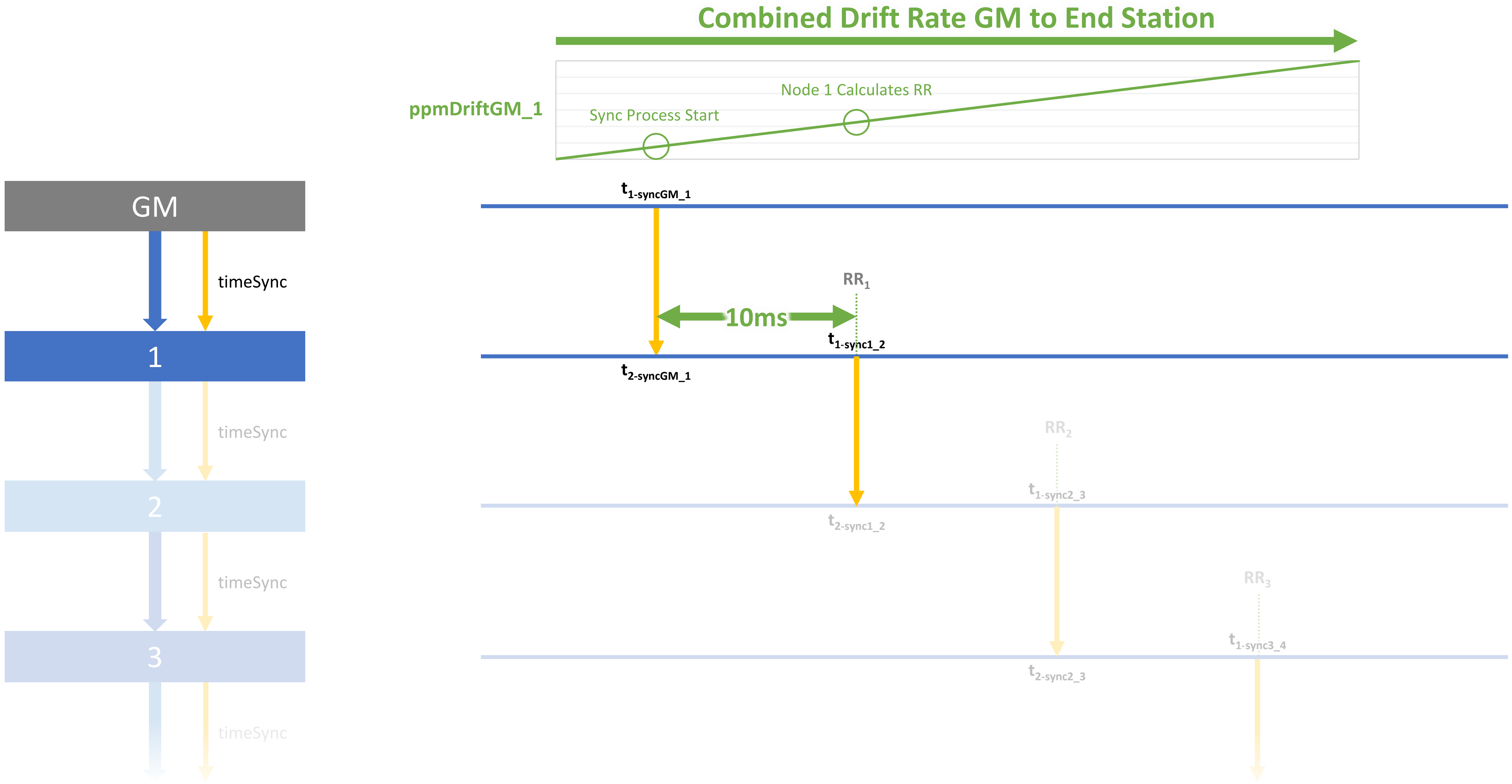
Combined Drift Rate GM to End Station
ppmDriftGM_100
Sync Process Start
Sync Process End
GM
pDelayResp
timeSync
1
pDelayResp
timeSync
pDelayResp
timeSync
100
t1-syncGM_1
RR1
t1-sync1_2
t2-syncGM_1
RR2
97 more...and 100 x 10ms = 1s
t2-sync99_100

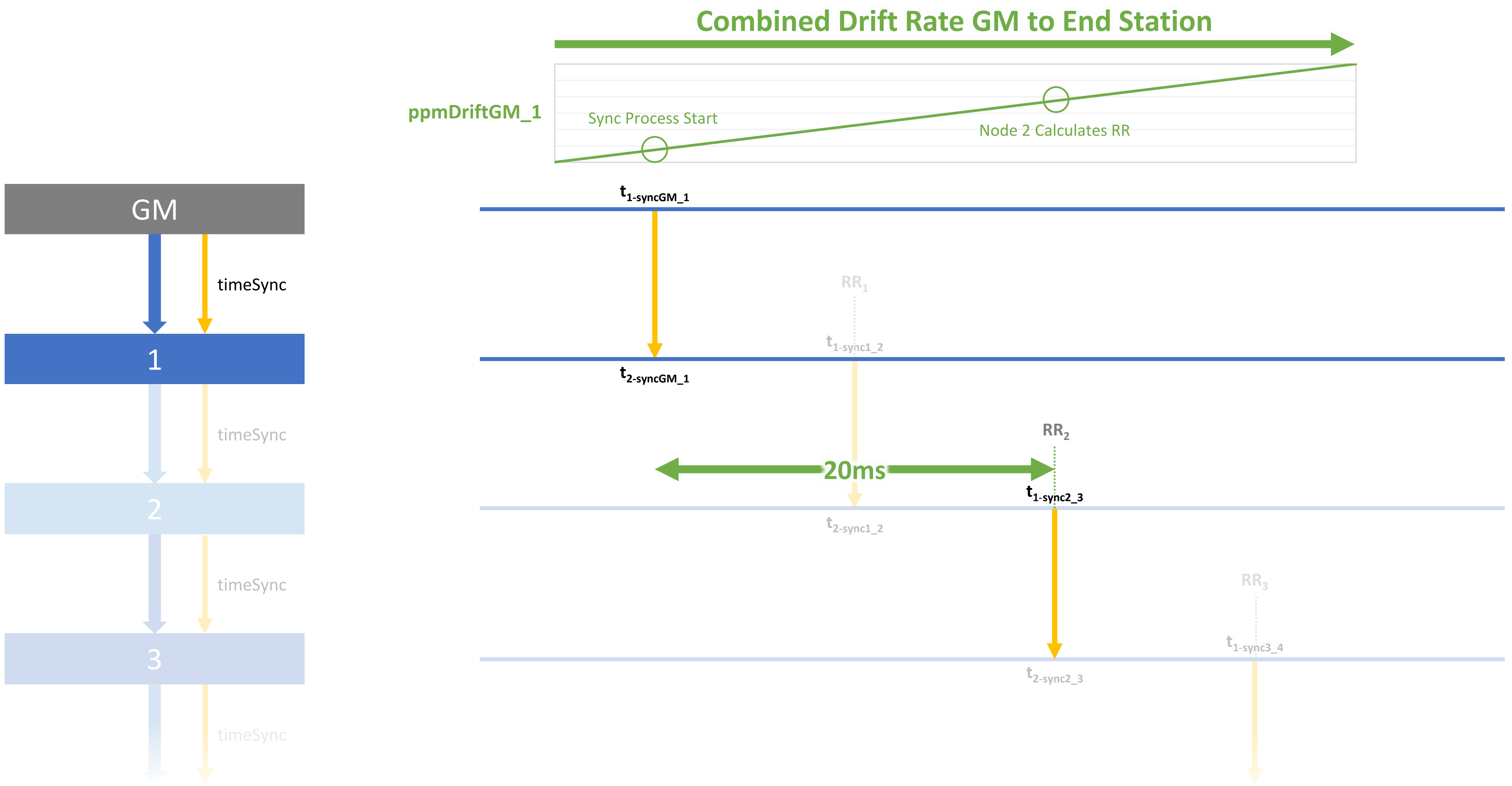

Combined Drift Rate
ppmDrift2_1
Effective Measurement Point
Attempt to measure NRR (mNNR)
mNRR used during Sync
GM
pDelayResp
timeSync
1
pDelayResp
timeSync
2
pDelayResp
timeSync
3
pDelayRsp
timeSync
t3-pDelayGM_1
t1-syncGM_1
RR1
0:5s
t3-pDelay1_2
t1-sync1_2
t4-pDelayGM_1
t2-syncGM_1
RR2
1s
t3-pDelay2_3
t1-sync2_3
t4-pDelay1_2
t2-sync1_2
RR3
0:25s
t1-sync3_4
t4-pDelay2_3
t2-sync2_3

Combined Drift Rate

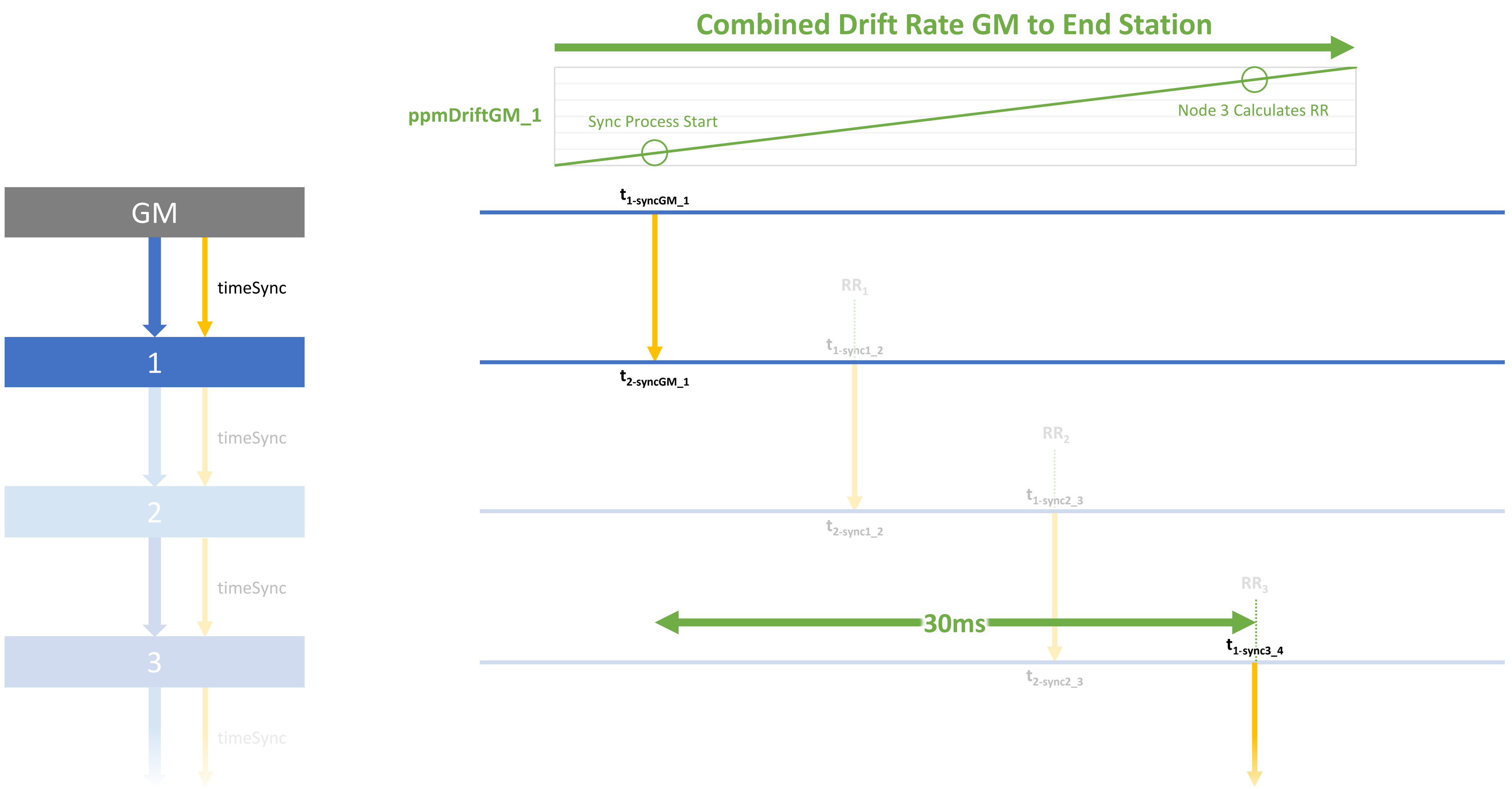



GM
timeSync
1
timeSync
2
timeSync
3
timeSync
$t_{1\text{-syncGM\_1}}$
RR_1
$t_{1\text{-sync1\_2}}$
$t_{2\text{-syncGM\_1}}$
RR_2
$t_{1\text{-sync2\_3}}$
$t_{2\text{-sync1\_2}}$
RR_3
$t_{1\text{-sync3\_4}}$
$t_{2\text{-sync2\_3}}$


Combined Drift Rate GM to End Station
ppmDriftGM_1
Node 1 Calculates RR
Sync Process Start
GM
timeSync
1
timeSync
2
timeSync
3
timeSync
t1-syncGM_1
RR1
10ms
t1-sync1_2
t2-syncGM_1
RR2
t1-sync2_3
t2-sync1_2
RR3
t1-sync3_4
t2-sync2_3



Combined Drift Rate GM to End Station
ppmDriftGM_1
Sync Process Start
Node 2 Calculates RR
GM
timeSync
1
timeSync
2
timeSync
3
timeSync
$t_{1\text{-syncGM\_1}}$
$t_{2\text{-syncGM\_1}}$
RR_1
$t_{1\text{-sync1\_2}}$
$t_{2\text{-sync1\_2}}$
RR_2
20ms
$t_{1\text{-sync2\_3}}$
$t_{2\text{-sync2\_3}}$
RR_3
$t_{1\text{-sync3\_4}}$


Combined Drift Rate GM to End Station
ppmDriftGM_1
Sync Process Start
Node 3 Calculates RR
GM
1
2
3
timeSync
timeSync
timeSync
timeSync
t1-syncGM_1
t2-syncGM_1
RR1
t1-sync1_2
t2-sync1_2
RR2
t1-sync2_3
t2-sync2_3
RR3
t1-sync3_4
30ms

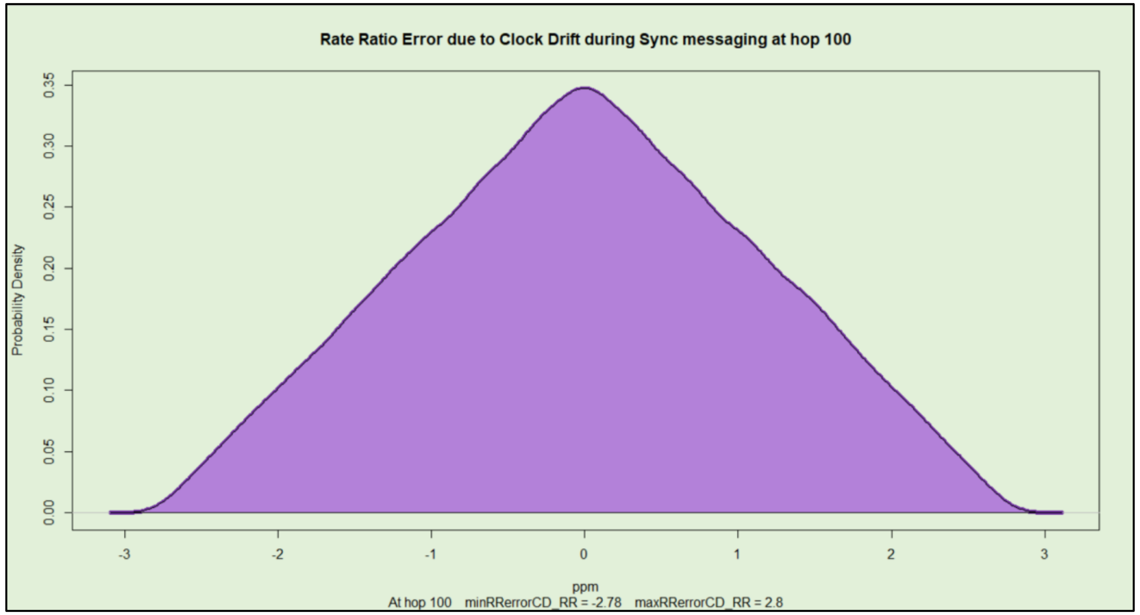
Error Due to Clock Drift During Sync Messaging

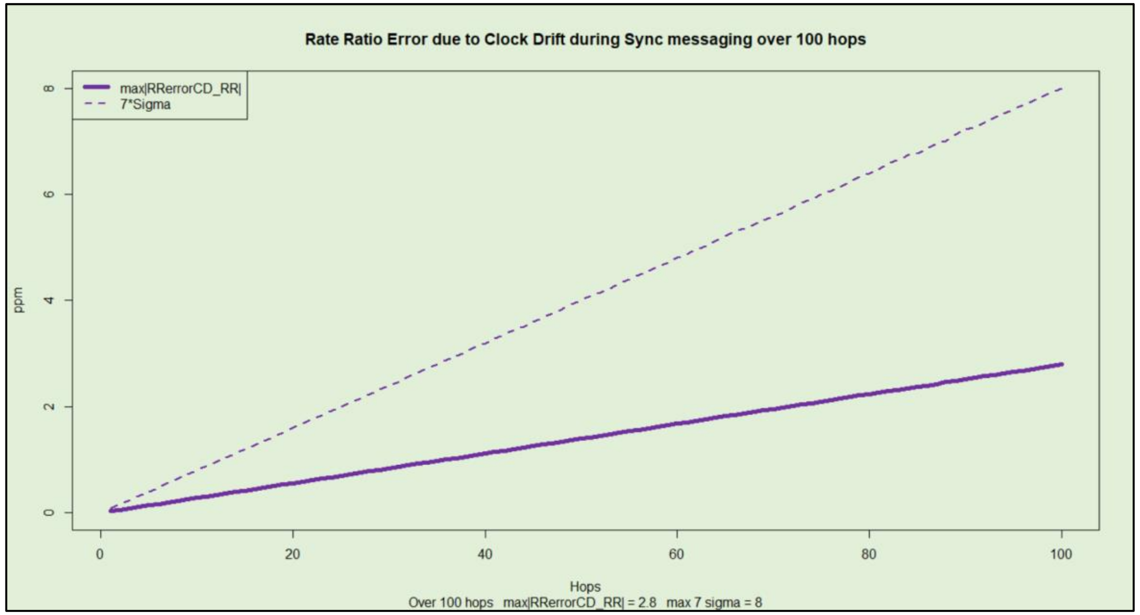
- Additional RR_{error} applied at each node

$$RR_{errorCD\_RR}(n) = \left(\frac{n \times residenceTime}{1000}\right) \times \left(clockDrift_{GM} - clockDrift(n)\right)$$

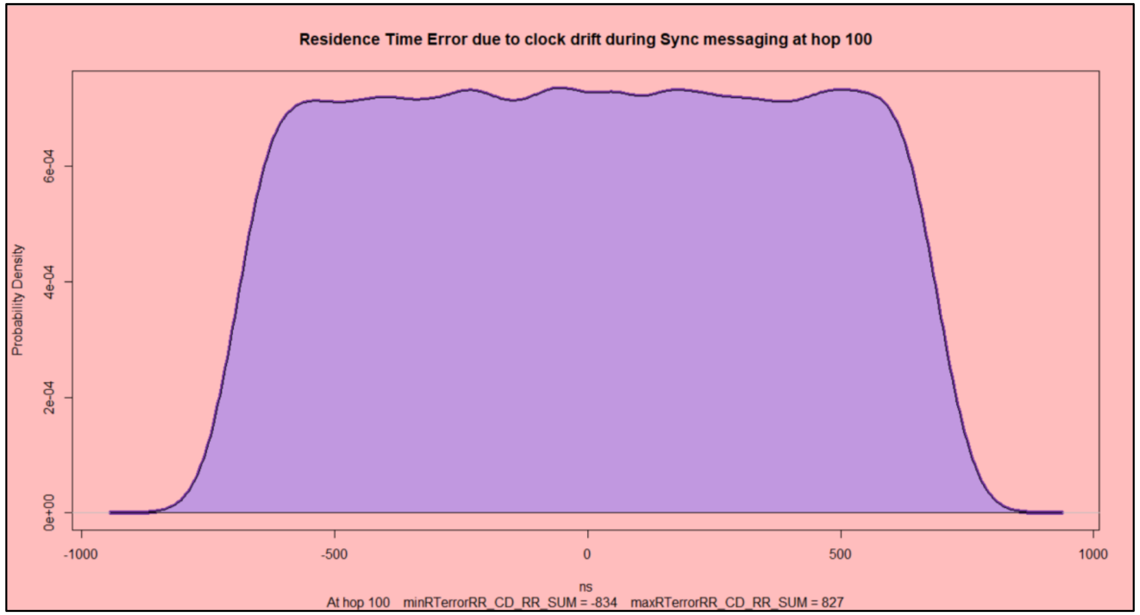
- Does not accumulate as other RR_{error} factors do
  - Not part of mNRR calculation. (Other RRerror factors accumulate as a result of RR calculation being an accumulation of mNRR calculations.)
- Does increase linearly along the chain of devices.
  - On average. For each run at each node it is proportional to the relative clock drift between the Local Clock and the GM
- GM Clock Drift has a huge impact on the magnitude of these errors
- Residence Time Errors due to the impact of these errors **do** accumulate in the Correction Field

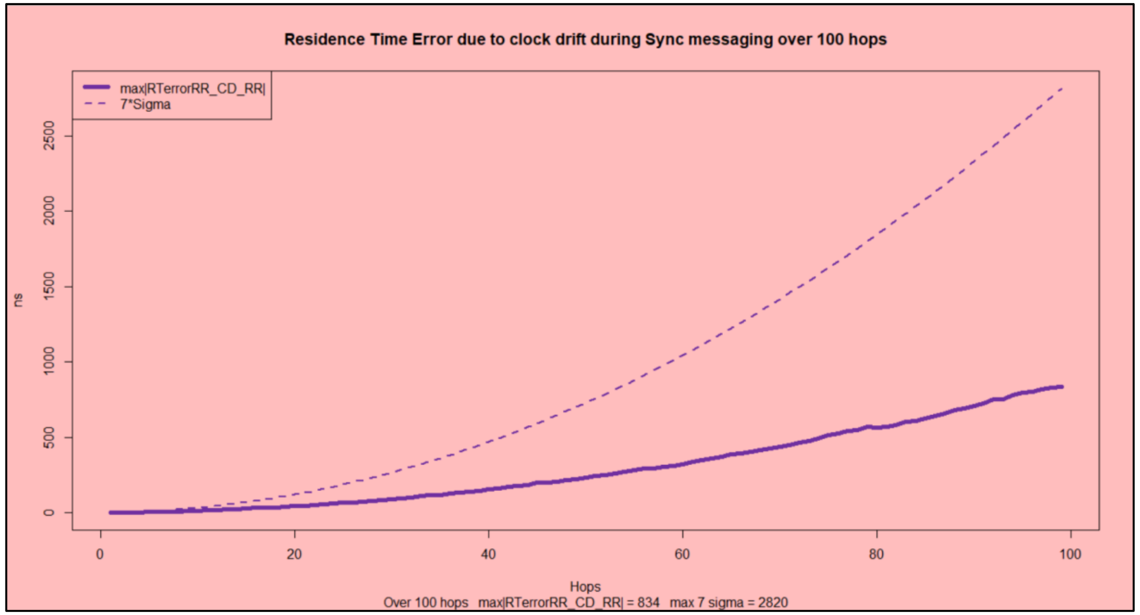
Error Due to Clock Drift During Sync Messaging – pDelay Interval 31.25ms





Error Due to Clock Drift During Sync Messaging – pDelay Interval 31.25ms





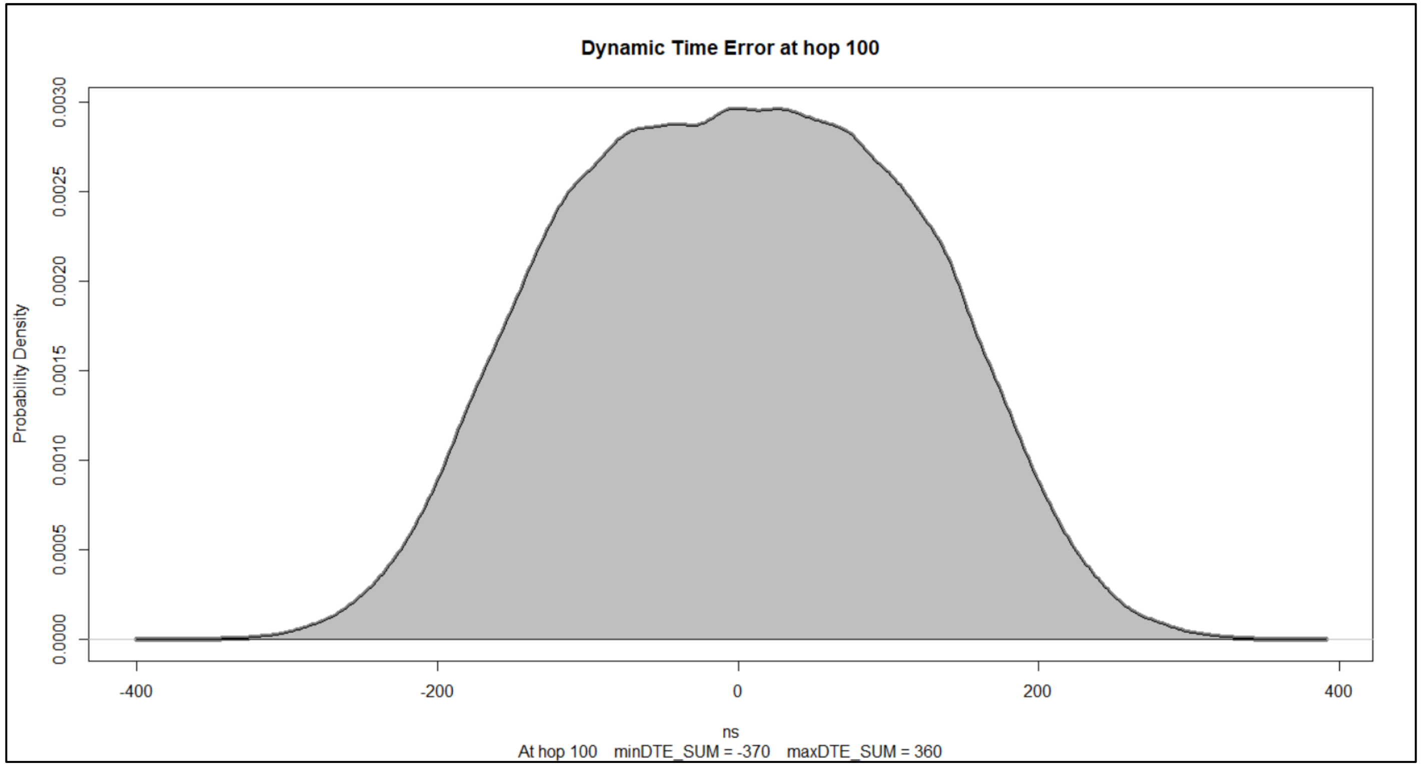
Algorithmic Compensation for Error Due to Clock Drift During Sync Messaging

- Clock Drift Compensation previously discussed was intended to apply to mNRR measurement. (See backup)
- Same principle applies to RR measurement
  - RR is being measured repeatedly over time
  - Drift of RR over time can be inferred and a correction factor applied
  - Linear drift should be a good approximation over short periods of time (~1s)
- Beyond the scope of detailed analysis using current Monte Carlo model, but can be modelled as a % effective (similar to Clock Drift error correction for mNRR)
  - Included in these simulations

Contribution to Next Draft

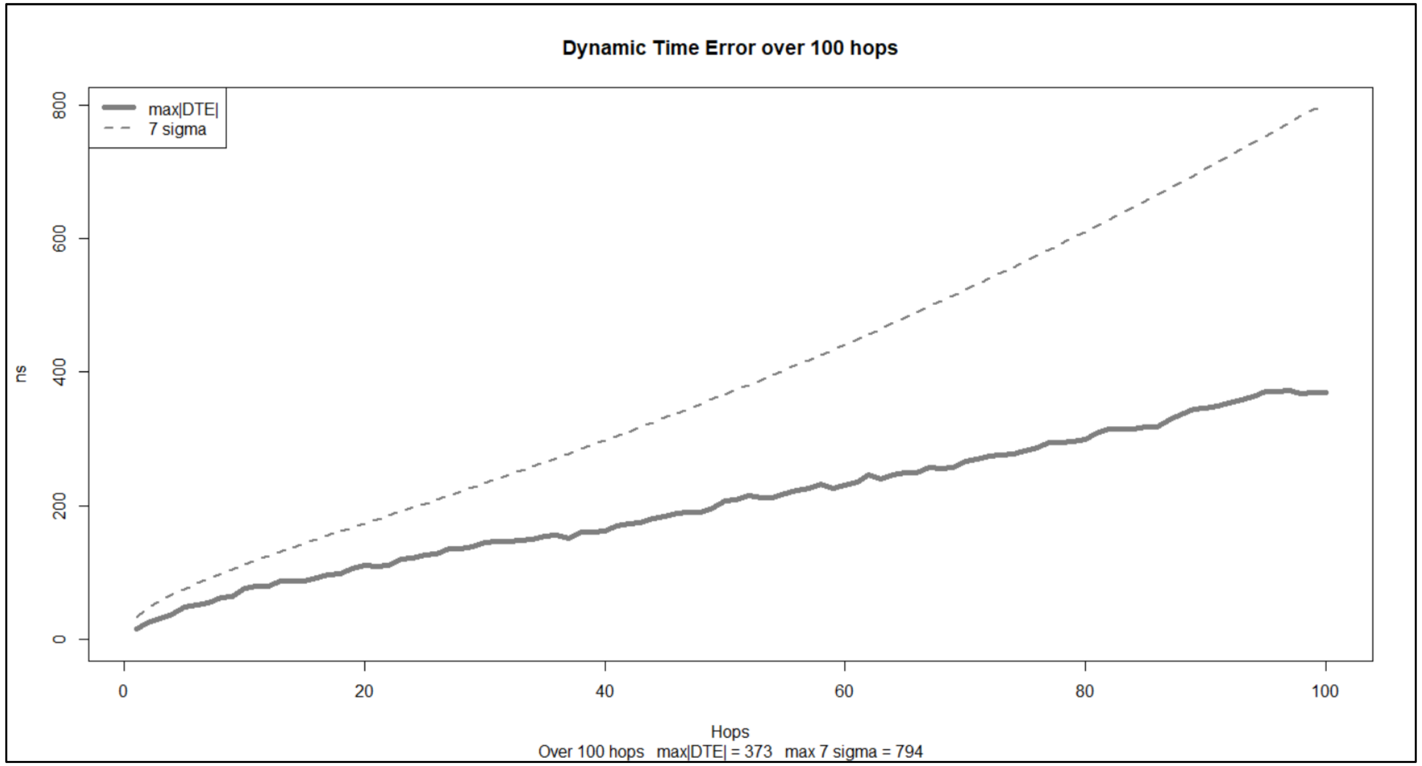
Potential basis for contribution

Potential Parameters & Settings



Input Errors		
GM Clock Drift Max	+1.5	ppm/s
GM Clock Drift Min	-1.5	ppm/s
Clock Drift Min (non-GM)	+1.5	ppm/s
Clock Drift Min (non-GM)	-1.5	ppm/s
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 Response Time	10	ms
residenceTime	10	ms
Input Correction Factors		
Mean Link Delay	98	%
mNRR Drift Rate	90	%
RR Drift Rate	90	%
pDelayResponse → Sync	0	%
mNRR Smoothing	8	
Configuration		
Hops	100	
Runs	1,000,000	

Potential Parameters & Settings



Input Errors		
GM Clock Drift Max	+1.5	ppm/s
GM Clock Drift Min	-1.5	ppm/s
Clock Drift Min (non-GM)	+1.5	ppm/s
Clock Drift Min (non-GM)	-1.5	ppm/s
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 Response Time	10	ms
residenceTime	10	ms
Input Correction Factors		
Mean Link Delay	98	%
mNRR Drift Rate	90	%
RR Drift Rate	90	%
pDelayResponse → Sync	0	%
mNRR Smoothing	8	
Configuration		
Hops	100	
Runs	1,000,000	

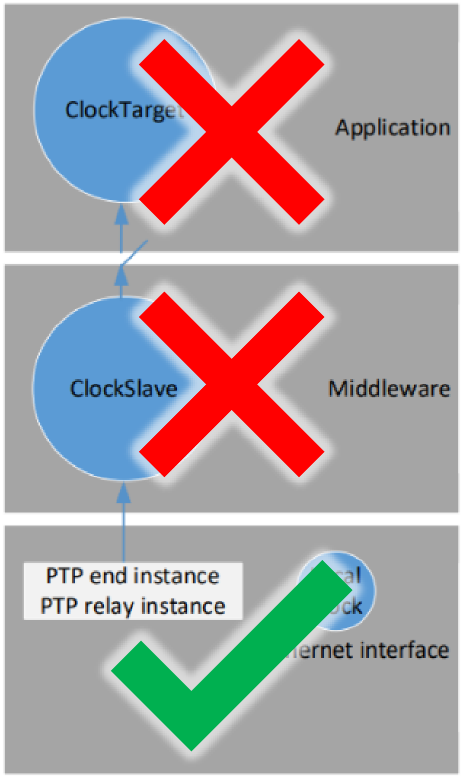
Thank you!

Back Up Material

Clarification: Scope of Monte Carlo Analysis

From February

- The analysis models errors and how they arise and interact, not the underlying entities that experience or generate the errors.
  - There is only Clock Drift...there are no Clocks
  - There is only Timestamp Error...there are no Timestamps
  - There is only Dynamic Time Error...there is no modelling of Time
- The analysis only covers errors associated with pDelay and Sync messaging...which are based on the Local Clock.
  - It models each node, including the GM, as a single clock...or rather, errors associated with a single clock.
  - No Global Time. No Working Clock. No ClockTarget or ClockSlave (only the lowest box from Guenter's presentation*)
    - Additional modelling may be required for errors associated with these elements. But, if the basic mechanism can't achieve the goal, these elements aren't going to improve the situation.



* https://www.ieee802.org/1/files/public/docs2021/60802-Steindl-ClockTarget-and-ClockSource-1121-v05.pdf

Clarification: Scope of Monte Carlo Analys[is]

From February

- The RStudio script combines ppm errors via addition...which introduces an error in the error...but the error in the error is swamped by other errors.
  - Errors in ppm are ratios and, to be accurate, should be multiplied.
  - But...if the ppm errors are small (one or two digits)...the inaccuracy from addition isn't significant.
    - 20 ppm + 30 ppm = 50 ppm
    - 20 ppm x 30ppm = 50.0006 ppm
- The trade-off is worth it for reduced runtime.
  - Multiplication is more expensive, computationally, than addition...especially when using double precision floating point numbers.

Planned Improvement

From February

- Analysis in this and previous presentations modelled DTE at the point the Sync message arrived at the End Station (hop 100).
- There is additional error as the GM and Local Clock drift with respect to each other prior to arrival of the next Sync message.
  - Modelled as follows (when hop = hops, i.e. for the final hop only)...

$$DTE_{endStationError} = \frac{syncInterval \times RRe_{rror}(hop)}{1000} \quad \textbf{ns}$$

  - Note: first time syncInterval has been used in this approach to error analysis
- This error will be added to future analyses (again: but not this one)
- Not expected to be significant relative to overall DTE

Algorithms – Clock Drift

Potential algorithm Clock Drift Compensation

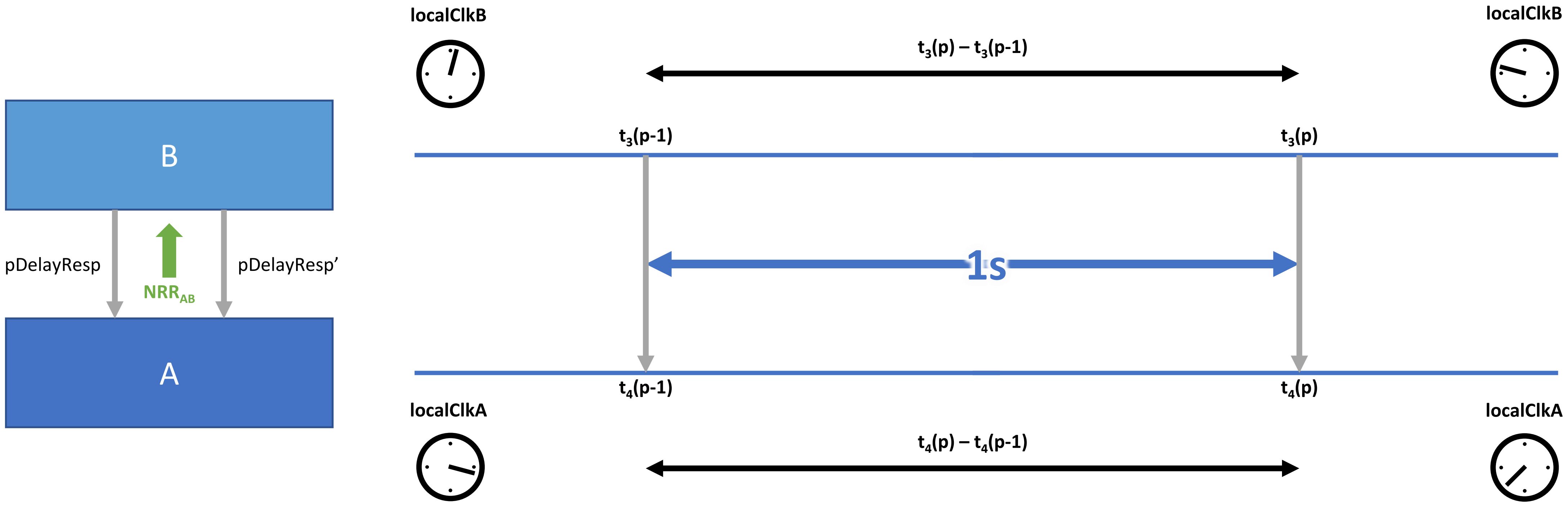
Compensating for Clock Drift

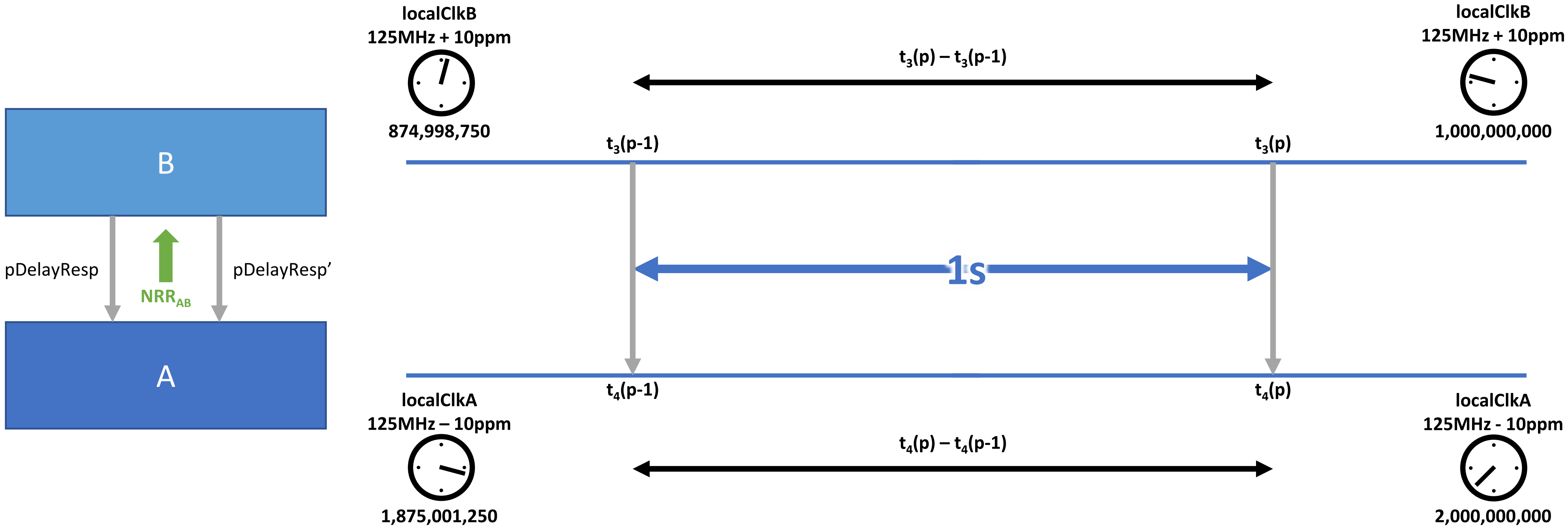
- Very hard to measure clock drift of an individual device...
  ...but we don't care about the clock drift of an individual device
- We care about Neighbor Rate Ratio and Rate Ratio...and we are constantly measuring both
- We can use theses measurements to track clock drift between two devices and create a correction factor
- The simplest algorithm would be to assume that clock drift is linear over the period of time we are interested in, i.e. one pDelay Interval
  - Actually, up to...

$$pDelayInterval + \frac{pDelayInterval \times (N + 2M - 1)}{2}$$

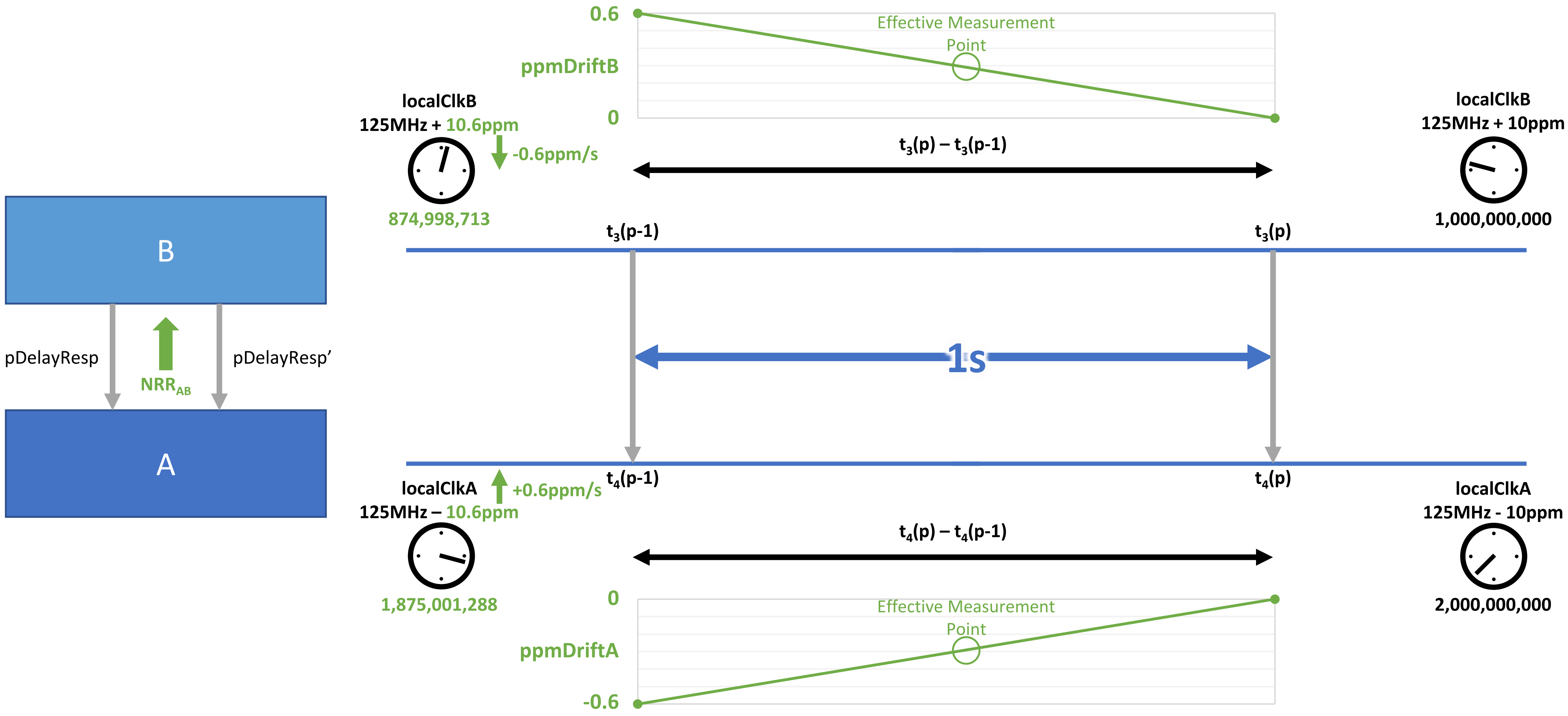
Maximum delay between attempt to measure NRR and using NRR as part of Sync messaging

Maximum delay between attempt to measure NRR and effective measurement point

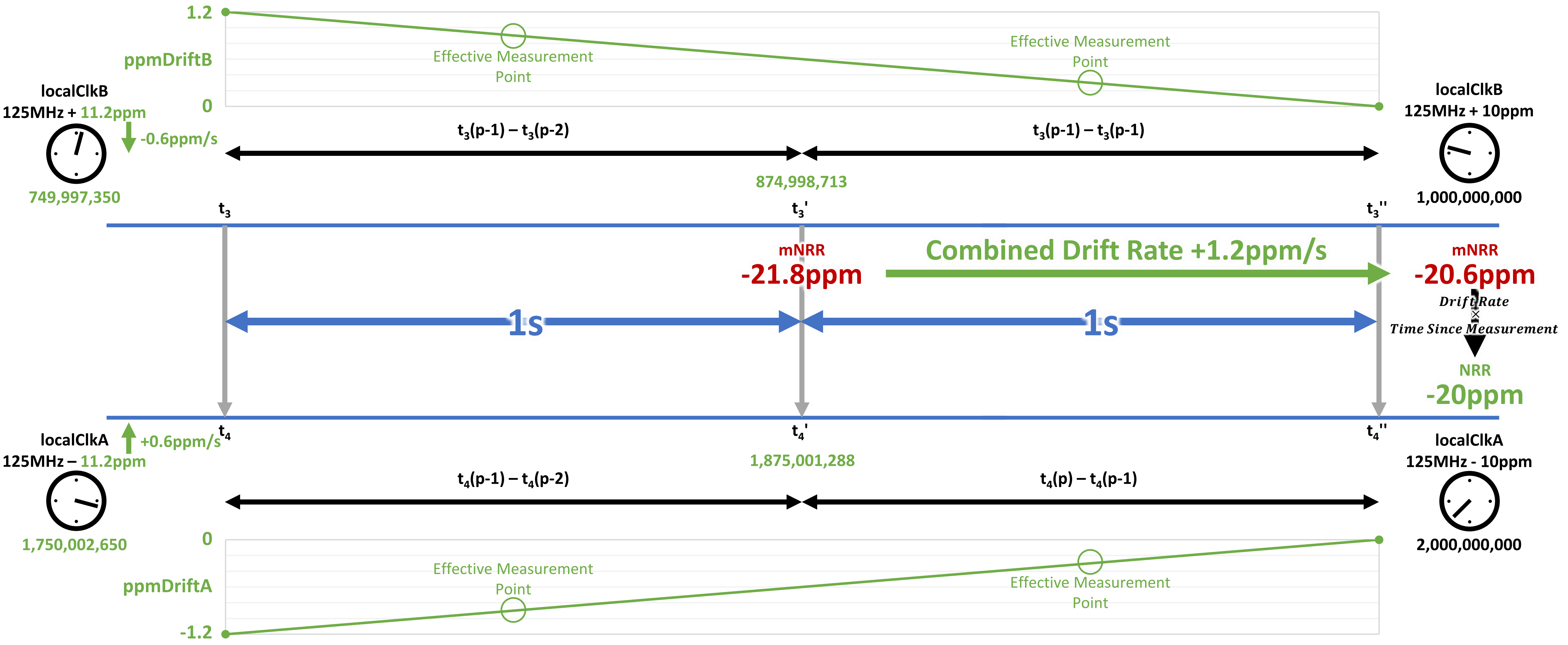

B
pDelayResp
NRR_{AB}
pDelayResp'
A
localClkB
localClkB
$t_3(p) - t_3(p\text{-}1)$
$t_3(p\text{-}1)$
$t_3(p)$
1s
$t_4(p\text{-}1)$
$t_4(p)$
localClkA
localClkA
$t_4(p) - t_4(p\text{-}1)$

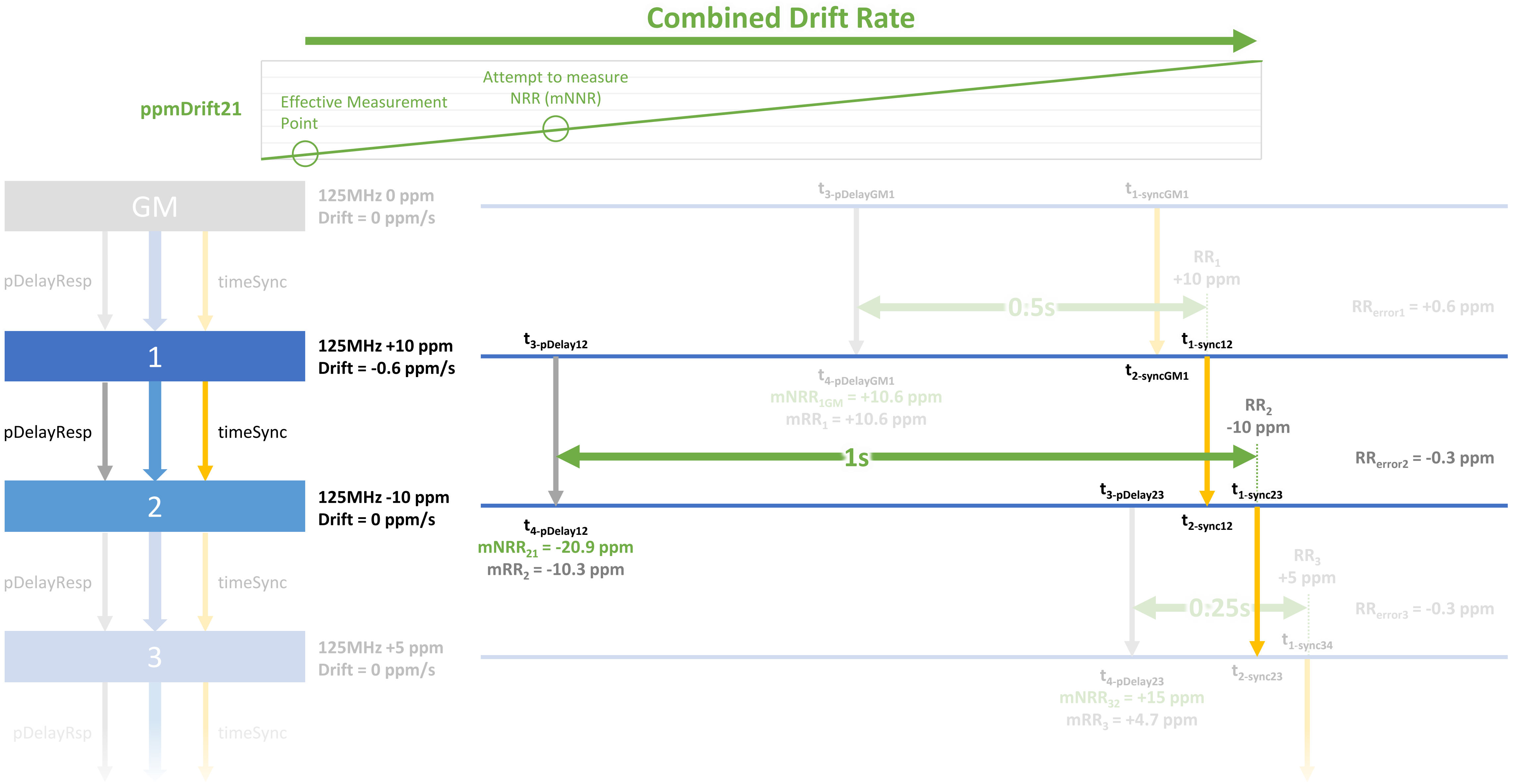


$$NRR_{AB} \text{ @ } t_4' = \frac{(t_4(p) - t_4(p\text{-}1))}{(t_3(p) - t_3(p\text{-}1))} = \frac{124{,}998{,}750}{125{,}001{,}250} \rightarrow -0.002\% = -20 \text{ ppm}$$



$$NRR_{AB} \text{ @ } t_4' = \frac{(t_4(p) - t_4(p\text{-}1))}{(t_3(p) - t_3(p\text{-}1))} = \frac{124{,}998{,}712}{125{,}001{,}288} \rightarrow -0.00206\% = -20.6 \text{ ppm}$$


1.2
ppmDriftB
0
Effective Measurement Point
Effective Measurement Point
localClkB
125MHz + 11.2ppm
-0.6ppm/s
749,997,350
t3(p-1) – t3(p-2)
t3(p-1) – t3(p-1)
874,998,713
localClkB
125MHz + 10ppm
1,000,000,000
t3
t3'
t3''
mNRR
-21.8ppm
Combined Drift Rate +1.2ppm/s
mNRR
-20.6ppm
DriftRate
Time Since Measurement
NRR
-20ppm
1s
1s
t4
t4'
t4''
localClkA
125MHz – 11.2ppm
+0.6ppm/s
1,750,002,650
1,875,001,288
t4(p-1) – t4(p-2)
t4(p) – t4(p-1)
localClkA
125MHz - 10ppm
2,000,000,000
0
ppmDriftA
-1.2
Effective Measurement Point
Effective Measurement Point



Combined Drift Rate
ppmDrift21
Effective Measurement Point
Attempt to measure NRR (mNNR)
GM
125MHz 0 ppm
Drift = 0 ppm/s
pDelayResp
timeSync
1
125MHz +10 ppm
Drift = -0.6 ppm/s
2
125MHz -10 ppm
Drift = 0 ppm/s
3
125MHz +5 ppm
Drift = 0 ppm/s
pDelayRsp
$t_{3\text{-pDelayGM1}}$
$t_{1\text{-syncGM1}}$
RR_1 +10 ppm
0.5s
RR_{error1} = +0.6 ppm
$t_{3\text{-pDelay12}}$
$t_{1\text{-sync12}}$
$t_{4\text{-pDelayGM1}}$
$mNRR_{1GM}$ = +10.6 ppm
mRR_1 = +10.6 ppm
$t_{2\text{-syncGM1}}$
RR_2 -10 ppm
1s
RR_{error2} = -0.3 ppm
$t_{3\text{-pDelay23}}$
$t_{1\text{-sync23}}$
$t_{4\text{-pDelay12}}$
$mNRR_{21}$ = -20.9 ppm
mRR_2 = -10.3 ppm
$t_{2\text{-sync12}}$
RR_3 +5 ppm
0.25s
RR_{error3} = -0.3 ppm
$t_{1\text{-sync34}}$
$t_{4\text{-pDelay23}}$
$mNRR_{32}$ = +15 ppm
mRR_3 = +4.7 ppm
$t_{2\text{-sync23}}$

Clock Drift Compensation – Possible Algorithm – 1

- Assume clock drifts linearly over the period of interest.
- Calculate Drift Rate

$$NRR_{driftRate} = \frac{mNRR(p) - mNRR(p-1)}{Time_{effectiveMeasure}(p) - Timeeff_{ectiveMeasure}(p-1)}$$

- Most of the time this will simplify to...

$$NRR_{driftRate} = \frac{mNRR(p) - mNRR(p-1)}{N \times pDelayInterval}$$

- ...but if taking a median of past NRR calculations it will get more complicated
  - If M>N the effective measurement time of (p) could be earlier than (p-1)!

Clock Drift Compensation – Possible Algorithm – 2

- Apply the correction factor. Example for NRR applied during Sync messaging (ppm)...

$$NRR_{sync} = mNRR + NRR_{driftRate}\left(delay_{mNRR\_sync} + \frac{pDelayInterval \times N}{2}\right)$$

- If a median of past NRR calculations is taken...

$$NRR_{sync} = mNRR + NRR_{driftRate}\left(delay_{mNRR\_sync} + \frac{pDelayInterval \times \left(N + 2\left(M_{used}(p-1) - Muse_{d}(p)\right) - 1\right)}{2}\right)$$