Initial 60802 dTE_R Time Series Simulation Results for Cases Using Drift Correction and Mean Link Delay Averaging Algorithms
Revision 1

Geoffrey M. Garner
Huawei (Consultant)

gmgarner@alum.mit.edu

IEEE 802.1 TSN TG
2022.05.27

Introduction - 1

- Reference [1] presents a comparison of simulation results obtained via time domain (time series) simulation and Monte Carlo simulation, for various simulation cases
- The Monte Carlo simulator and results for various simulation cases are described in [2] and [6] and in references cited in those presentations; proposed simulation cases for comparison are described in [8]
- The time domain simulator is described in [3], [4], [5], and [7], and references cited in those presentations
- The specific simulation cases compared are summarized in slide 6 of [2], which was reproduced as slide 3 of [1] and is reproduced on the next slide here for convenience.
- Reference [1] provided max|dTE_R| results for cases A, B, D, E, and F, obtained using the time-domain simulator, for comparison with results obtained using the Monte Carlo approach simulator
  - The time domain results were based on 300 multiple, independent replications for each case

Introduction - 2

Proposed Time Series Simulations – Details (copied from [1])

Case	Reason	Errors: Clock Drift Model – 40°C ↔ +85°C Hold for 30s at Each (Each node's position in cycle distributed at random across 100% of Cycle)	Errors: Timestamp Granularity (ns)	Errors: Dynamic Timestamp Error (±ns)	Parameter: pDelay Interval (ms)	Parameter: Residence Time (ms)	Parameter: pDelay Turnaround Time (ms)	Correction Factors: Mean Link Delay Averaging	Correction Factors: mNRR Smoothing Factor N
A	Baseline with previous assumptions	Ramp Rate 1°C / s (Cycle of 310 s)	8	8	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 560 s)	8	4	1000	10	10		

Timestamp Granularity and Dynamic Timestamp Error are uniform distributions
Sync Interval: 125ms
pDelay Interval variation is +0-30%; uniform distribution
Sync Interval variation is ±10%; gamma distribution with 90% probability of landing in the ±10% range
Note: 8ns Timestamp Granularity in Time Series Simulation is equivalent to ±4ns Timestamp Granularity Error in Monte Carlo Analysis
No difference between base (PHY related) propagation delay for pDelay and Sync messages

Introduction - 3

- The comparison of the time series and Monte Carlo simulation results is given in slide 26 of [1], which is reproduced on the next slide here for convenience
- In almost all cases, the time series simulation results were less than the Monte Carlo simulation results, by 5 – 35%
- Further analysis in [1] indicated that the differences are likely due to differences in the models for frequency drift rate used in the two simulators
  - The Monte Carlo simulations assumed frequency drift rate at each node is 0 with 20% probability, and uniform in the range [-1.5 ppm/s, +1.5 ppm/s] with 80% probability
  - The time domain simulations used a temperature profile and frequency stability versus temperature to obtain actual frequency offset at any given time; analysis in [1] showed that the equivalent probability function has significantly smaller probability of exceeding 0.5 ppm/s than the Monte Carlo simulation probability function
  - In addition, the maximum absolute value of frequency drift rate for the time domain simulations is 1.35 ppm/s, versus 1.5 ppm/s for the time series simulations

Introduction - 4

Taken from slide 26 of [1], based on slide provided in [6]

Compare Monte Carlo results with unfiltered Time Series Results
Results are for $\max|dTE_R|$, in ns, after 100 hops (i.e., at node 101)

		Confidence Intervals & MAX →	Monte Carlo				Time Series – Unfiltered				Time Series – Filtered			
Case	Reason	Key Factor	Lower	Point	Upper	MAX	Lower	Point	Upper	MAX	Lower	Point	Upper	MAX
A	Baseline with previous assumptions	pDelayInterval 31.25ms; 1ms Residence Time & pDelay Turnaround; 8ns Dynamic Timestamp Error	2,543	2,657	2,774	2,941	2,265	2,315	2,375	2,515	1,624	1,688	1,772	1,887
							-10.9%	-12.9%	-14.4%	-14.5%				
B	Verify optimised pDelayInterval	pDelay Interval 1000ms	13,621	13,927	14,505	15,566	8,870	9,207	9,535	10,037	9,190	9,443	9,945	10,524
							-34.9%	-33.9%	-34.3%	-35.5%				
C		pDelay Interval 250ms	4,175	4,285	4,498	4,609	Not Run							
D		pDelay Interval 31.25ms	6,326	6,469	6,710	6,915	5,894	5,969	6,304	7,089	5,483	5,546	5,800	6,407
							-6.8%	-7.7%	-6.1%	2.5%				
E	Verify effect of reduced Timestamp Error	Timestamp Errors halved pDelay Interval 31.25ms	3,623	3,684	3,915	3,996	3,307	3,366	3,503	3,845	3,024	3,090	3,256	3,578
							-8.7%	-8.6%	-10.5%	-3.8%				
F	Verify effect of reduced Clock Drift	Clock Drift halved pDelay Interval 1000ms	6,816	6,961	7,224	7,775	4,623	4,754	4,896	5,204	4,739	4,940	5,204	5605
							-32.2%	-31.7%	-32.2%	-33.1%				

Introduction - 5

- Details of the above analysis are given in [1]
- Based on the above analysis, it is expected that there will be better agreement between the time domain and Monte Carlo simulation results if the frequency drift rate probability function suggested in [1] is used for the Monte Carlo simulations (see slide 47 of [1])
- It was suggested that one item for future work is the running of the Monte Carlo simulation cases A, B, D, E, and F with this probability function
- It must be stressed that the purpose of cases A, B, D, E, and F was to validate agreement between the Monte Carlo and Time Series simulations

Introduction - 6

- Cases B, D, E, F all have filtered (and unfiltered) $\max|dTE_R|$ exceeding the 1 μs objective for $\max|TE_R|$ by a significant amount
  - For these cases, $\max|dTE_R|$ is in the 3 – 10 μs range
- In all these cases, residence time and Pdelay turnaround time are 10 ms, and in some cases the Pdelay interval is as much as 1 s
- In case A, where residence time and Pdelay turnaround time are 1 ms and the Pdelay interval is 31.25 ms, $\max|dTE_R|$ is smaller, but is still between 1 and 2 μs
- Previous simulations [5] (also summarized in slides 23 – 25 of [1]) indicated that $\max|dTE_R|$ could be less than 1 μs using the case A parameters if, in addition, a window size of 7 or 11 is used when computing neighborRateRatio (NRR)
  - Those results also computed NRR as a median of the most recent 7 or 11 values, respectively, though possibly the use of the median has small effect and might make the results worse (see [8]); note that cases with non-zero window size but no median computation were not run for [5]

Introduction - 7

- In any case, the opinion was expressed in several of the meetings where [1] – [8] were presented indicated that residence time and Pdelay turnaround time of 1 ms might be too stringent
  - It also was indicated that a larger Pdelay interval would be desirable
- It was suggested in [8] that a larger residence time, Pdelay turnaround time, and Pdelay interval might be possible if algorithms were used to estimate and correct the error in NRR and resulting cumulative rateRatio (RR) relative to the GM
  - The errors are due to frequency drift that occurs between:
    - The local clock frequency drifts that occurs between the computation of NRR on receipt of Pdelay_Resp_Follow_Up and the use of NRR in computing cumulative RR when sending Sync or Follow_Up
    - The GM versus local clock frequency drift that occurs between the computation of RR on receipt of Sync and Follow_Up and the computation of the correctionField when Sync is sent
    - The GM versus local clock that occurs between the most recent RR computation and the computation of synchronized time for and end system

Introduction - 8

- Specific algorithms were suggested in [8] for reducing NRR and RR
  - In these algorithms, NRR and RR drift rates are computed as average rates, using the two most recently computed NRR and RR values, respectively
  - The drift rates are then used to correct the NRR and RR values computed as specified in 802.1AS, by multiplying the drift rates by the respective time intervals between the NRR or RR computations and the times at which the values are used
  - Details of these computations are given in slides 37 – 42 of [8] (using the scheme where NRR and RR are represented as ppm offsets and added as opposed to the multiplying of ratios, as the NRR and RR values are small compared to 1)
- It also was suggested that the averaging of successive mean link delay (MLD) measurements could result in smaller max$|dTE_R|$, by reducing errors due to timestamp granularity and timestamp error (see 11.1.2 of 802.1AS)
  - Averaging filter details are given in slides 43 and 44 of [8] (steady-state operation of the filter is equivalent to Eq. (11-4) of 802.1AS-2020

Introduction - 9

- Reference [8] proposed a set of assumptions and parameters for simulation cases where residence time, Pdelay turnaround time, and Pdelay interval are more in line with the desire for values larger than 1 ms, 1 ms, and 31.25 ms, respectively
- It was suggested that four cases be considered (letter designations are chosen here, for convenience):
  - Case G: No correction for NRR, RR, or MLD
  - Case H: Correction for NRR
  - Case I: Correction for NRR and RR
  - Case J: Correction for NRR, RR, and MLD averaging
- Multiple replications of a time domain simulation were presented in [1]
- In the remainder of this presentation
  - the parameters common to all the case are summarized
  - The case G multiple replication results of [1] are shown, for convenience
  - Single replication results for cases G, H, I, and J are shown and discussed

Common Assumptions for Cases G, H, I, and J

- Clock stability and temperature profile modeled as described in slides 9 – 11 of [1] (reproduced in following slides)
- Variation in Sync and Pdelay intervals modeled as described in slides 5 – 8 of [1] (reproduced in following slides)
- Other assumptions in the table on slides 13 and 14 of [1] are used (reproduced in following slides)
- Mean Sync interval: 125 ms
- Mean Pdelay interval: 125 ms
- Timestamp granularity: 8 ns (modeled by truncating to next lower multiple of 8 ns)
- Dynamic timestamp error is taken to have a uniform distribution over ± 4 ns
- Residence time: 10 ms
- Pdelay turnaround time: 10 ms
- Window size (N) for mean Neighbor Rate Ratio (mNRR) smoothing: 3
- Window size (M) for median computation in mNRR smoothing: 1 (median is not taken)
- Simulation time: 3150 s

Assumptions for Temperature Profile ([1] - [4])

- The temperature history is assumed to vary between – 40°C and +85°C, at a rate of 1°C /s; this takes 125 s
- When the temperature is increasing and reaches +85°C, it remains at +85°C for 30 s
- The temperature then decreases from +85°C to – 40°C at a rate of 1°C /s; this takes 125 s
- The temperature then remains at – 40°C for 30 s
- The temperature then increases to +85°C at a rate of 1°C /s; this takes 125 s
- The duration of the entire cycle (i.e., the period) is therefore 310 s

Assumptions for Frequency Stability due to Temperature Variation

- The dependence of frequency offset on temperature is assumed to be as described in [4] and [5] of Reference [5] here
  - Specifically, the values a_0, a_1, a_2, and a_3 computed in [5] of Reference [5] will be used in the cubic polynomial fit, and the resulting frequency offset will be multiplied by 1.1 (i.e., a margin of 10% will be used).
- The frequency stability data that this polynomial fit is based on is contained in the Excel spreadsheet attached to [4] of Reference [5] here
  - This data was provided by the author of [4] of Reference [5] here
- The time variation of frequency offset is obtained from the cubic polynomial frequency dependence on temperature, and the temperature dependence on time described in the previous slide
  - The time variation of phase/time error at the LocalClock entity is obtained by integrating the above frequency versus time waveform
  - The time variation of frequency drift rate at the LocalClock entity is obtained by differentiating the above frequency versus time waveform

Assumptions on Relative Time Offsets of Phase Error Histories at Each Node

- The phase of the LocalClock time error waveform at each node is chosen randomly in the range [0,T], at initialization, where T is the period of the phase and frequency variation waveforms (i.e., 310 s)

Model for Variable Sync Interval – 1

- IEEE Std 802.1AS-2020 requires in 10.7.2.3 (an analogous requirement is in 9.5.9.2 of IEEE Std 1588-2019):

When the value of syncLocked is FALSE, time-synchronization messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between message transmissions is within ± 30% of $2^{currentLogSyncInterval}$. In addition, a PTP Port shall transmit time-synchronization messages such that at least 90% of the inter-message intervals are within ± 30% of the value of $2^{currentLogSyncInterval}$. The interval between successive time-synchronization messages should not exceed twice the value of $2^{portDS.logSyncInterval}$ in order to prevent causing a syncReceiptTimeout event. The PortSyncSyncSend state machine (see 10.2.12) is consistent with these requirements, i.e., the requirements here and the requirements of the PortSyncSyncSend state machine can be met simultaneously.

NOTE 1—A minimum number of inter-message intervals is necessary in order to verify that a PTP Port meets these requirements. The arithmetic mean is the sum of the inter-message interval samples divided by the number of samples. For more detailed discussion of statistical analyses, see Papoulis [B25].

Model for Variable Sync Interval – 2

- The above requirements do not specify the actual probability distribution; however, it was decided to model the Sync Intervals as being gamma-distributed
  - The gamma distribution is often used to model inter-message times in networks
  - The same model was used in simulations for the PTP Telecom Time Profile with full timing support from the network (ITU-T Rec. G.8275.1)
- While both 802.1AS-2020 and 1588-2019 both allow variation in the duration of the Sync intervals up to $\pm$ 30% of the mean Sync interval, it was decided after the discussion of [5] to consider variations of $\pm\beta$, with β = 10%
- The shape and scale parameters of the gamma distribution are chosen such that the distribution has the desired mean and that 90% of the probability mass is within β of the mean
- The resulting gamma distribution has a shape parameter of 270.5532; the details of how this parameter is obtained and how the samples of the gamma distribution are generated are given in [5]

Model for Variable Pdelay Interval - 1

- ❑IEEE Std 802.1AS-2020 has the following NOTE in 11.5.2.2 (it refers to the requirement in 9.5.13.2 of IEEE Std 1588-2019):

NOTE 3—The MDPdelayReq state machine ensures that the times between transmission of successive Pdelay_Req messages, in seconds, are not smaller than $2^{\mathrm{currentLogPdelayReqInterval}}$. This is consistent with IEEE Std 1588-2019, which requires that the logarithm to the base 2 of the mean value of the interval, in seconds, between Pdelay_Req message transmissions is no smaller than the interval computed from the value of the portDS.logMinPdelayReqInterval member of the data set of the transmitting PTP Instance. The sending of Pdelay_Req messages is governed by the LocalClock and not the synchronized time (i.e., the estimate of the Grandmaster Clock time). Since the LocalClock frequency can be slightly larger than the Grandmaster Clock frequency (e.g., by 100 ppm, which is the specified frequency accuracy of the LocalClock; see B.1.1), it is possible for the time intervals between successive Pdelay_Req messages to be slightly less than $2^{\mathrm{currentLogPdelayReqInterval}}$ when measured relative to the synchronized time.

- ❑However, the actual requirement in 9.5.13.2 of IEEE 1588 is:

Subsequent Pdelay_Req messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between Pdelay_Req message transmissions is not less than the value of $0.9 \times 2^{\mathrm{portDS.logMinPdelayReqInterval}}$.

- ❑This requirement will be satisfied even if the LocalClock is 100 ppm fast due to the factor of 0.9 (frequency offsets resulting from the temperature profile and frequency stability model of [4] are less than 100 ppm)

Model for Variable Pdelay Interval - 2

- IEEE 802.1AS and IEEE 1588-2019 do not specify the distribution for the Pdelay interval, nor do they specify the maximum amount that the actual intervals can exceed $2^{\text{portDS.logMinPdelayReqInterval}}$
- For the simulations, it was decided to use a uniform distribution over the range $[P, 1.3P]$, where P is $2^{\text{portDS.logMinPdelayReqInterval}}$

Other Assumptions - 1

Assumption/Parameter	Description/Value
Hypothetical Reference Model (HRM), see note following the tables	101 PTP Instances (100 hops; GM, followed by 99 PTP Relay Instances, followed by PTP End Instance
Computed performance results	(a) max$\lvert dTE_{R(k,\,0)} \rvert$ (i.e., maximum absolute relative time error between node k (k > 0) and GM, both filtered (PLL filter output at each node) and unfiltered (input to PLL filter at each node)
Use syncLocked mode for PTP Instances downstream of GM	Yes
Endpoint filter parameters	K_pK_o = 11, K_iK_o = 65 (f_{3dB} = 2.5998 Hz, 1.288 dB gain peaking, ζ = 0.68219)
Simulation time	3150 s; discard first 50 s to eliminate any startup transient before computing max$\lvert dTE_{R(k,\,0)} \rvert$ (i.e., 10 cycles of frequency variation after discard)

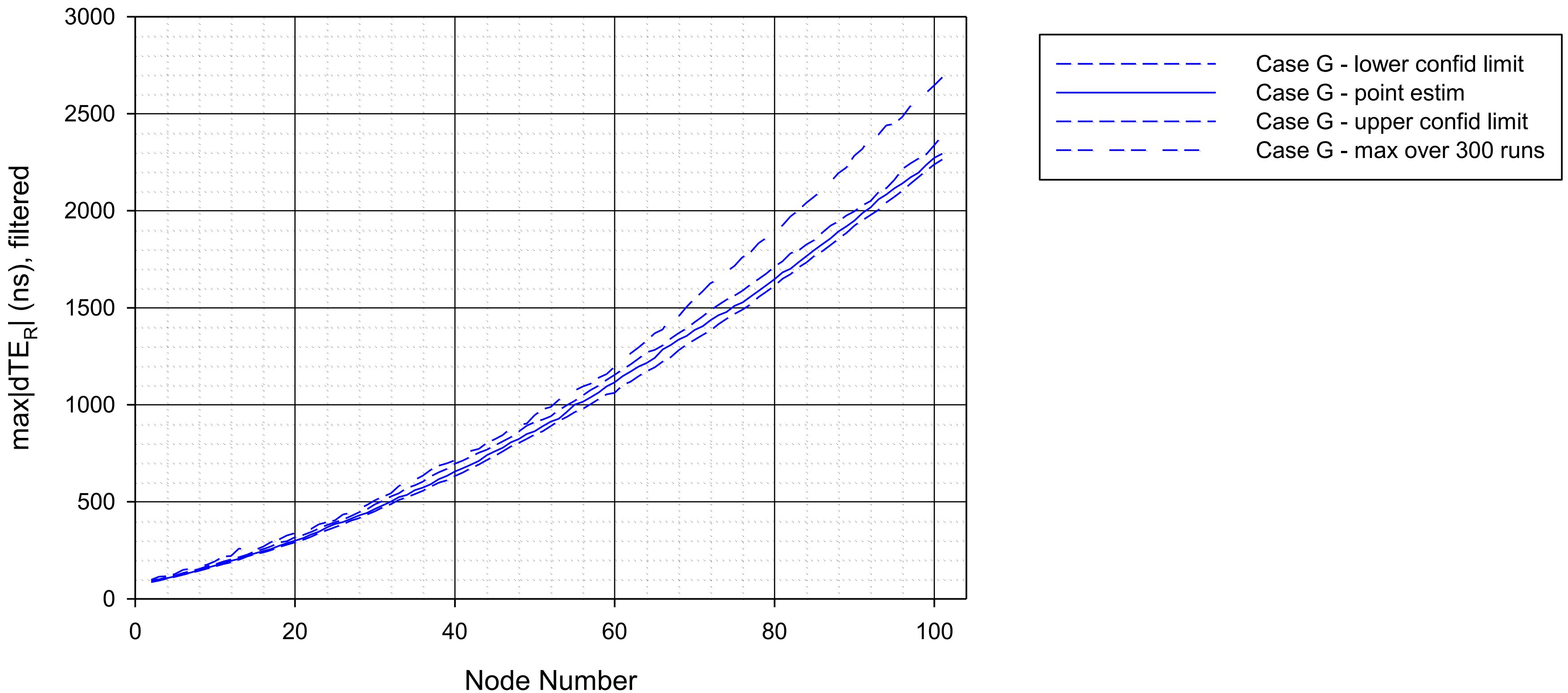
Other Assumptions - 2

Assumption/Parameter	Description/Value
Number of independent replications, for each simulation case	300
GM rateRatio and neighborRateRatio computation granularity	0 (i.e., we do not truncate when computing timestamp differences and ratios of differences, but use floating point arithmetic)
Mean link delay	500 ns
Link asymmetry	0
Any variable PHY delay in addition to the dynamic timestamp error described above is assumed to be zero	0

Case G max|dTE_R| Multiple Replication Results [1] - 1

Base case (case G) - mult replication results - filt
GM time error modeled; dTE_R is relative to GM

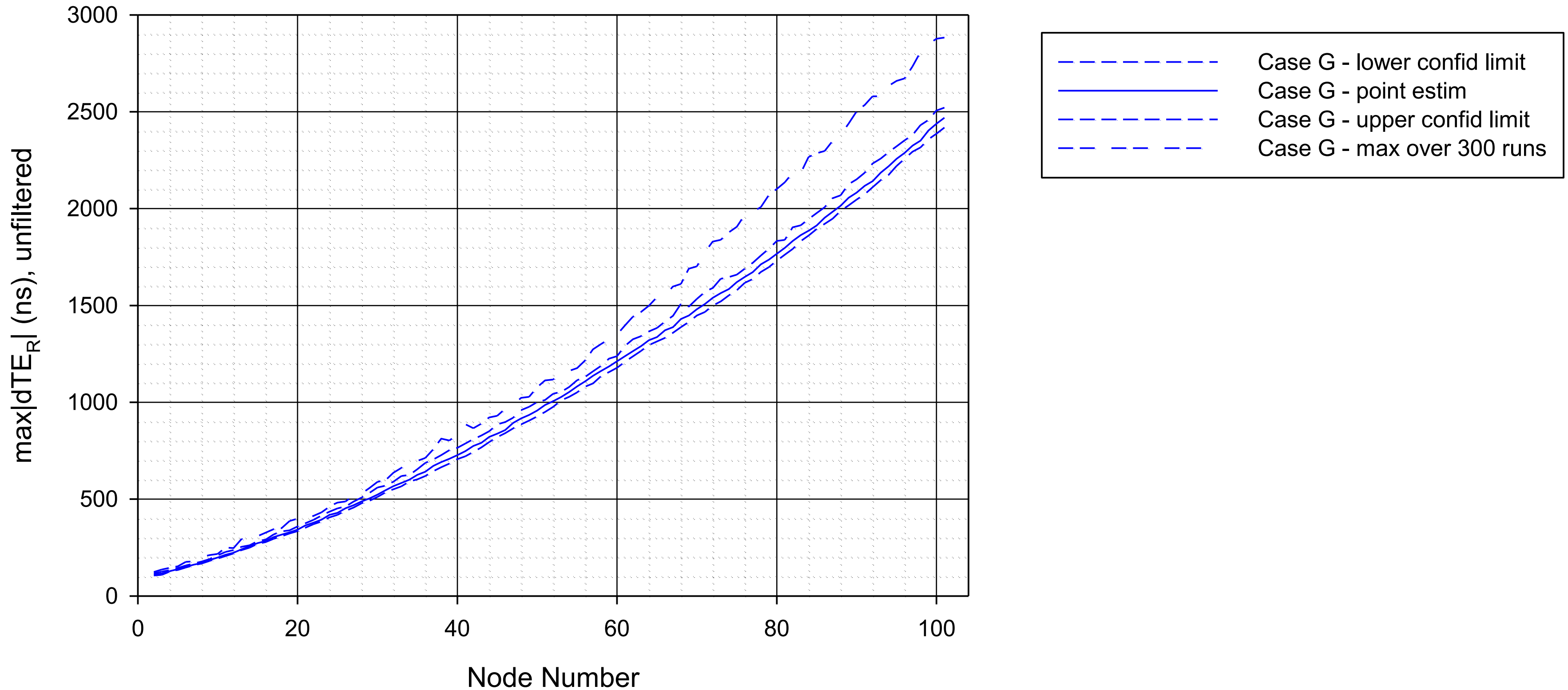
GM labeled node 1
neighborRateRatio measured with window of size 1 (N = 1) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity and dynamic timestamp error have uniform distributions



Case G max|dTE$_R$| Multiple Replication Results [1] - 2

Base case (case G) - mult replication results - unfil
GM time error modeled; dTE$_R$ is relative to GM

GM labeled node 1
neighborRateRatio measured with window of size 1 (N = 1) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity and dynamic timestamp error have uniform distributions



Case G max|dTE_R| Multiple Replication Results [1] - 3

Summary of Results at Last Node

Case	Filtered/ Unfiltered	Lower (ns)	Point Est (ns)	Upper (ns)	Max (ns)
G	Filtered	2294	1688	2387	2688
	Unfiltered	2419	2469	2521	2883

Compare with Monte Carlo Simulation result from[8] (slide 16) for case of no algorithms:

3938 ns

The time domain simulation result max (unfiltered) of2883 ns is 26.8% smalle r

As indicated in [1] (slide 57), it would be useful to run the Monte Carlo simulations again for the base case, but with the frequency drift rate probability model based on the frequency drift rate time history (slides 46 an d 47)

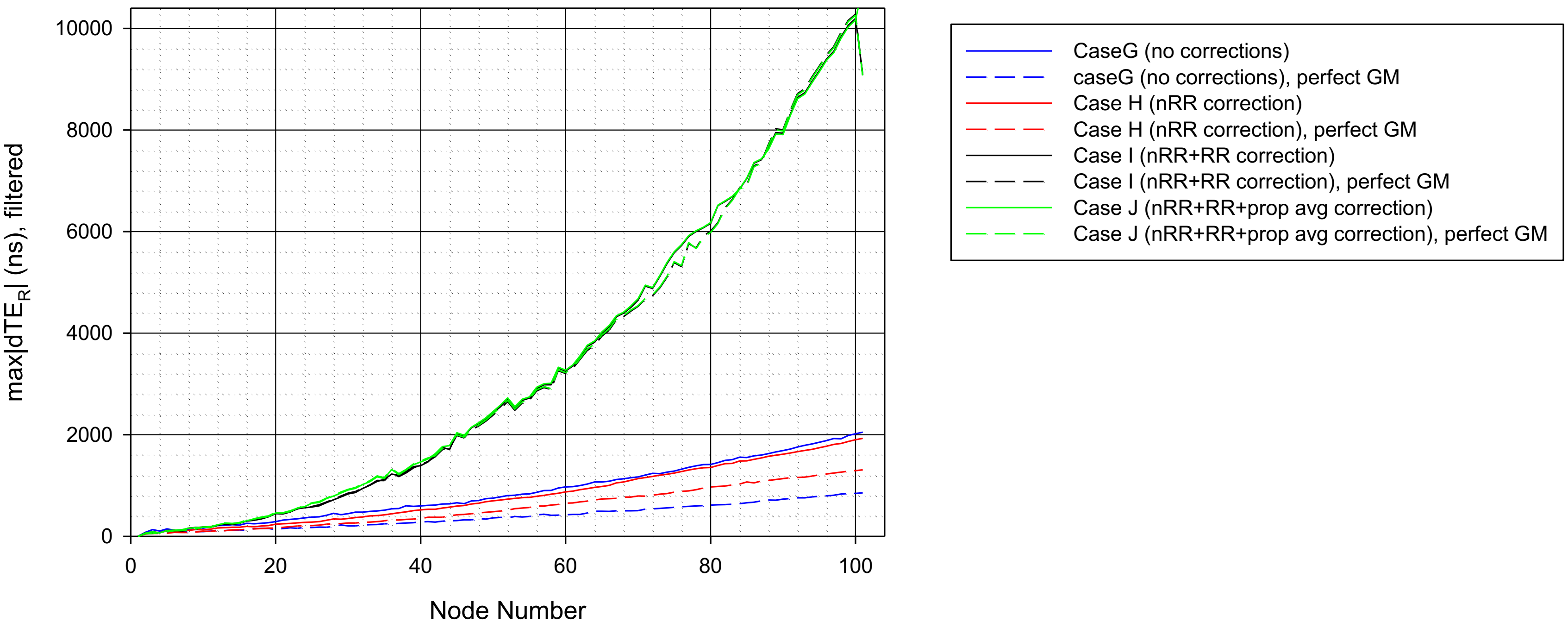
Cases G, H, I, J - Single-Replication max|dTE_R| Results - 1

- Results for max|dTE_R|, relative to the GM, versus node number are summarized on the next four slides, assuming
  - a) GM has the same local clock stability (time error model) as other nodes (slides 12 – 14 above)
  - b) GM is perfect (zero time error)
- Item b) above was considered to show the effect of GM time error

- Slide 25: Filtered max|dTE_R|, single replication
- Slide 26: Filtered max|dTE_R|, single replication, more detailed view
- Slide 27: Unfiltered max|dTE_R|, single replication
- Slide 28: Unfiltered max|dTE_R|, single replication, more detailed view
- Slide 29: Numerical results for nodes 65 and 101 (GM is node 1)

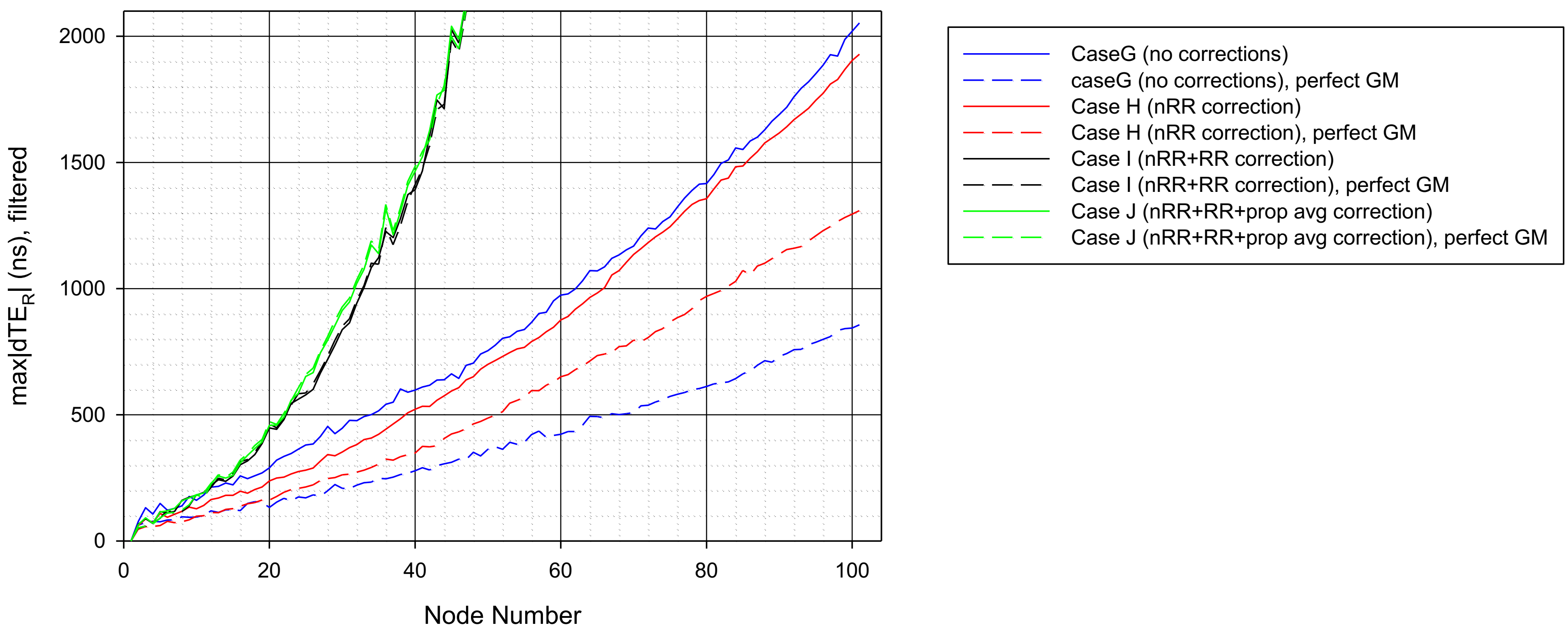
Cases G, H, I, J - Single-Replication max|dTE_R| Results - 2

Cases G, H, I, J - single replic results - filtered
dTE_R is relative to GM
GM labeled node 1
neighborRateRatio measured with window of size 3 (N = 3) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity modeled as truncation
Dynamic timestamp error has uniform distribution



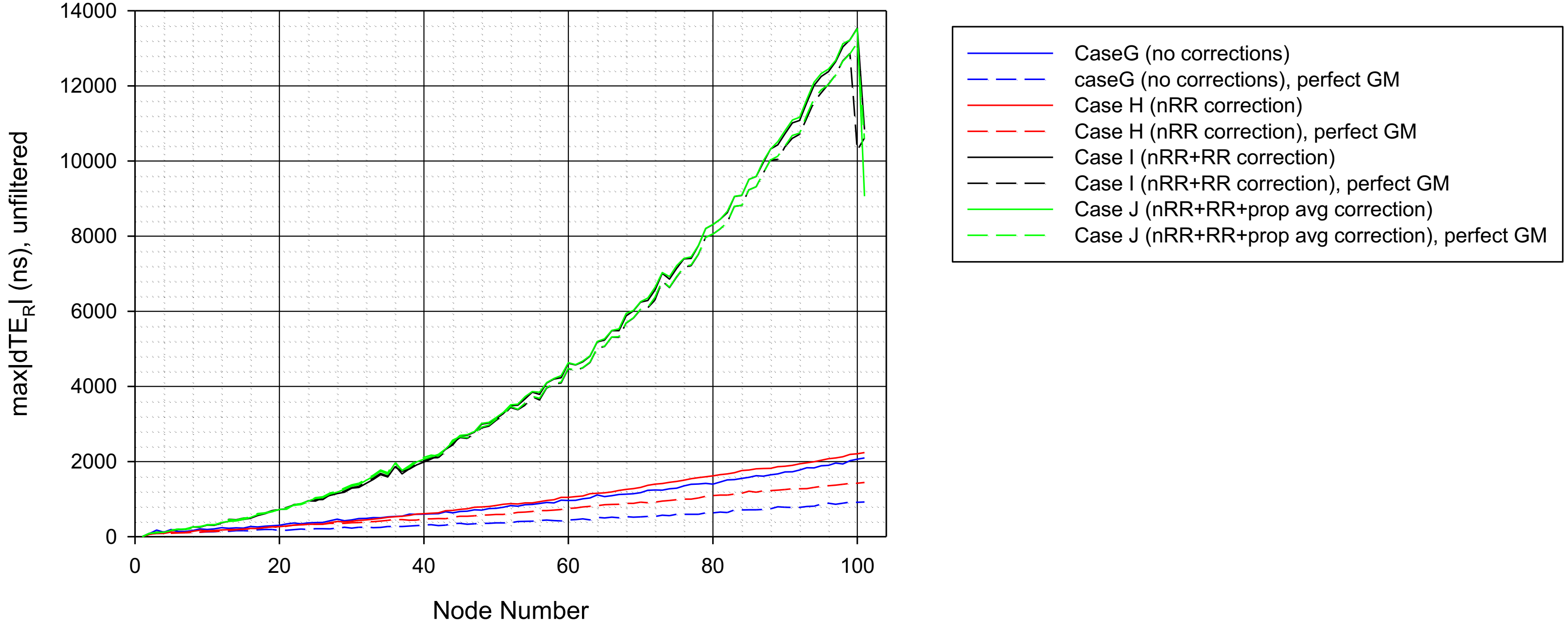
Cases G, H, I, J - Single-Replication max|dTE_R| Results - 3

Cases G, H, I, J - single replic results - filtered
dTE_R is relative to GM
GM labeled node 1
neighborRateRatio measured with window of size 3 (N = 3) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity modeled as truncation
Dynamic timestamp error has uniform distribution



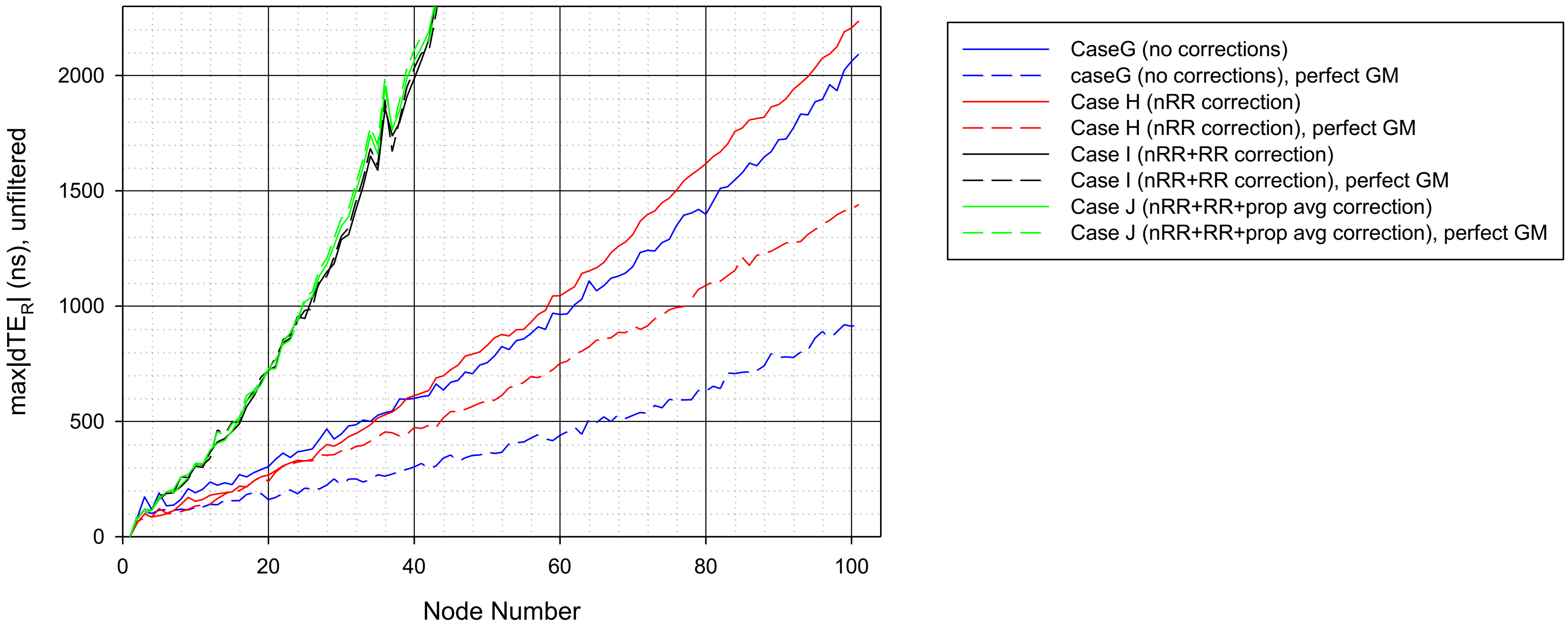
Cases G, H, I, J - Single-Replication max|dTE_R| Results - 4

Cases G, H, I, J - single replic results - unfiltered
dTE_R is relative to GM
GM labeled node 1
neighborRateRatio measured with window of size 3 (N = 3) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity modeled as truncation
Dynamic timestamp error has uniform distribution



Cases G, H, I, J - Single-Replication max|dTE$_R$| Results - 5

Cases G, H, I, J - single replic results - unfiltered
dTE$_R$ is relative to GM
GM labeled node 1
neighborRateRatio measured with window of size 3 (N = 3) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay interval (uniform distribution)
Timestamp granularity modeled as truncation
Dynamic timestamp error has uniform distribution



Summary of Results at Nodes 65 and 101 for Cases with GM Error

Case (GM error modeled)	Filtered/ Unfiltered	Node 65 (ns)	Node 101 (ns)
G (multiple replication, max over 300 runs)	Filtered	1369	2688
	Unfiltered	1543	2883
G (single replication)	Filtered	1070	2053
	Unfiltered	1066	2093
H (single replication)	Filtered	983	1929
	Unfiltered	1166	2237
I (single replication)	Filtered	4005	9087
	Unfiltered	5233	10836
J (single replication)	Filtered	4026	9064
	Unfiltered	5267	10832

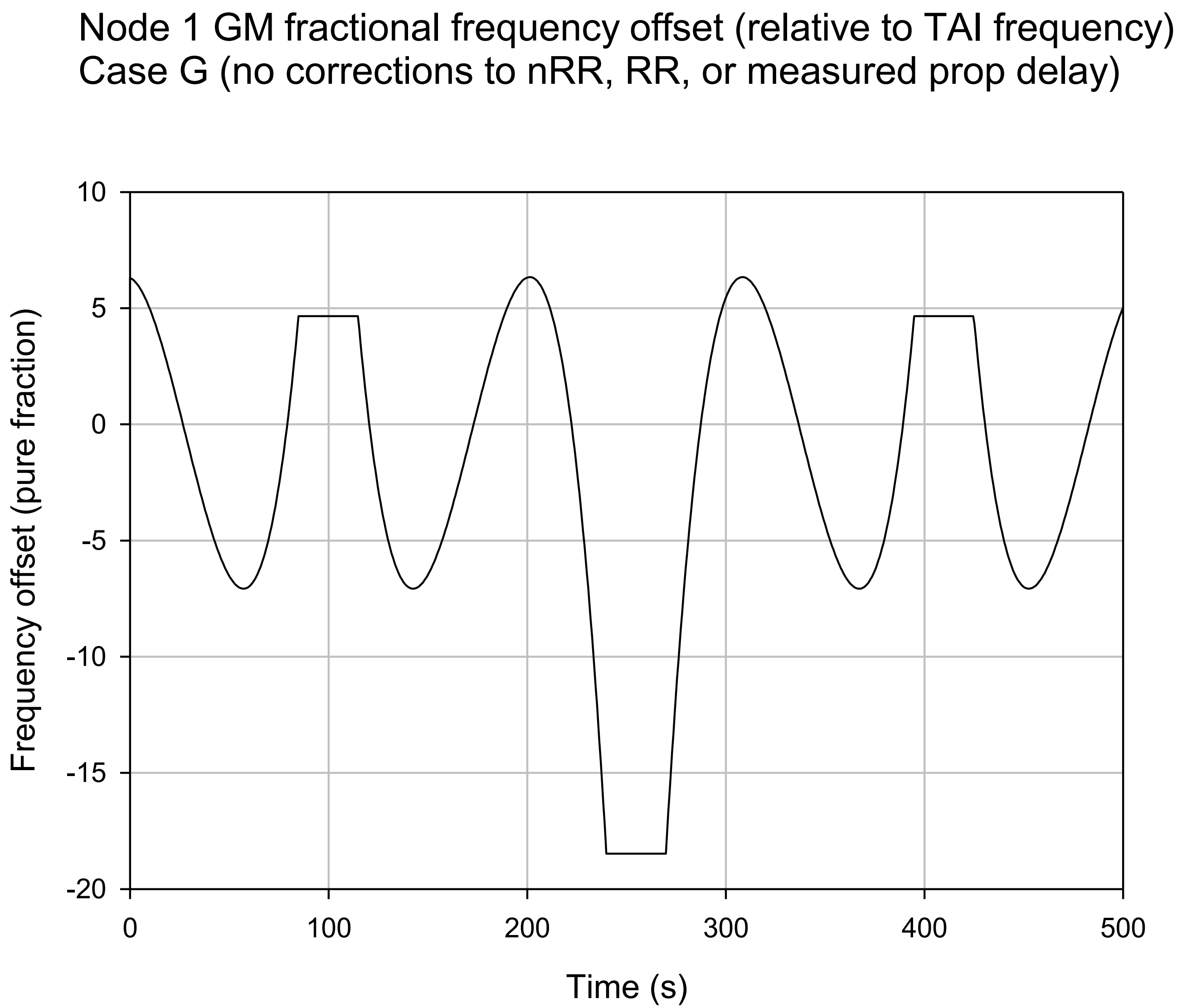
Discussion of max|dTE_R| Results - 1

- As expected, the single replication results for case G are smaller than the multiple replication results (by approximately 20-25% for the filtered results and 30% for the unfiltered results)
- The case H results (NRR correction) show slight improvement over case G (no algorithmic correction)
  - However, the improvement is sufficiently small that it could be due to statistical variation and, in any case, is insufficient to allow max|$TE_{R|}$| to be within 1 μs for 64 hops (the case H filtered results exceed 1900 ns for 100 hops)
- The case I results (NRR+RR correction) are worse than the case G and case H results, by approximately a factor of 4 (i.e., the filtered results are 4005 ns for 64 hops and 9087 ns for 100 hops
- The case J results (NRR+RR+MLD averaging) are on the same order as the case I results

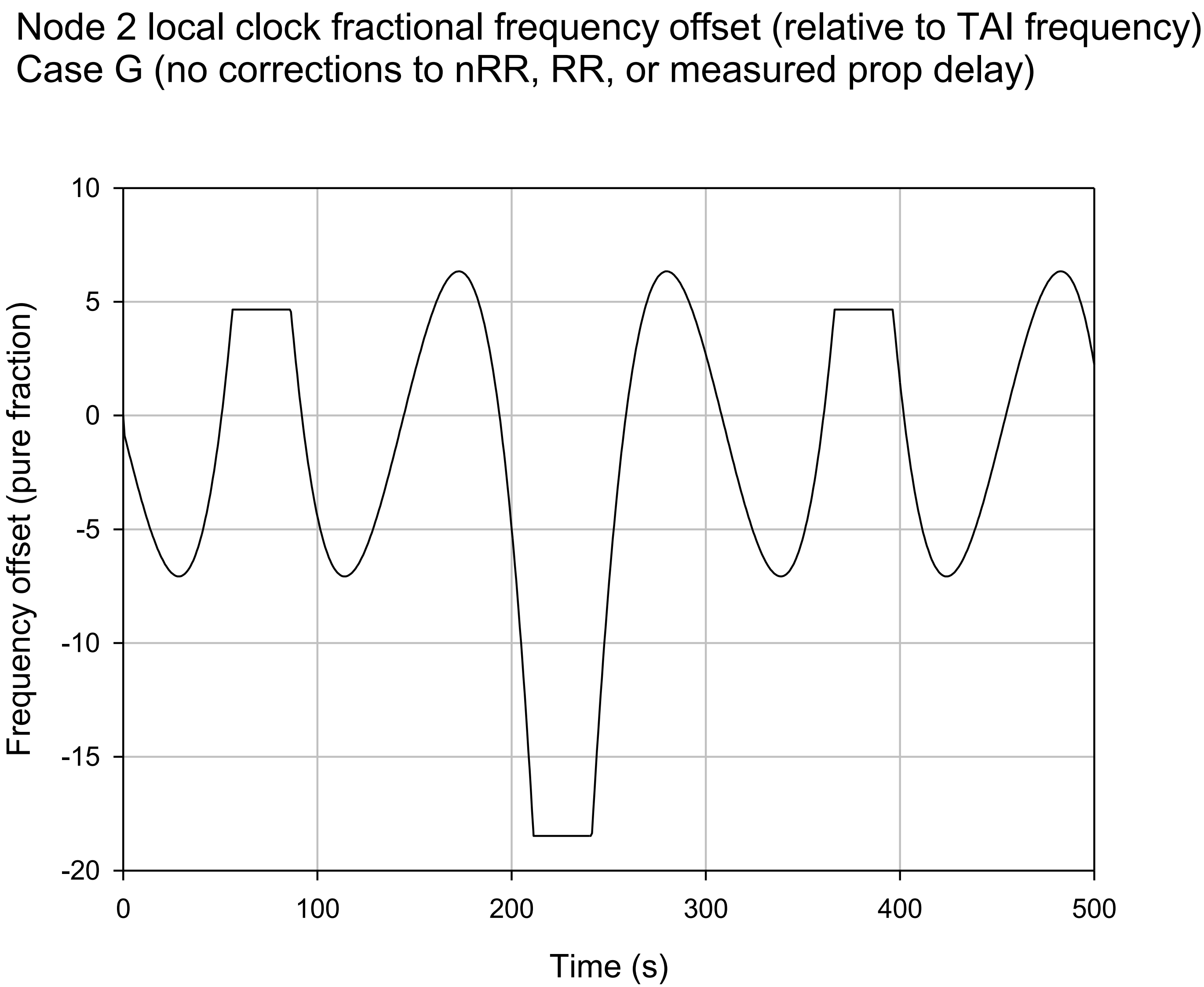
Discussion of max|dTE_R| Results - 2

- Overall, the algorithmic corrections do not improve max|dTE_R| and, in fact, make max|dTE_R| worse in cases I and J
- To help understand this in more detail, the following slides show, for selected nodes and cases, actual Local Clock FFO at the GM and downstream nodes, error in measured FFO (i.e., measured FFO minus actual FFO), filtered dTE_R time history, and unfiltered dTE_R time history
  - All the cases that follow have GM error modeled
  - In all cases, the first 500 s of the 3150 s simulation time is plotted
    - Since the FFO waveforms are periodic with period of 310 s, 500 s is sufficient and results in less cluttered plots

Case G – Node 1 (GM) FFO

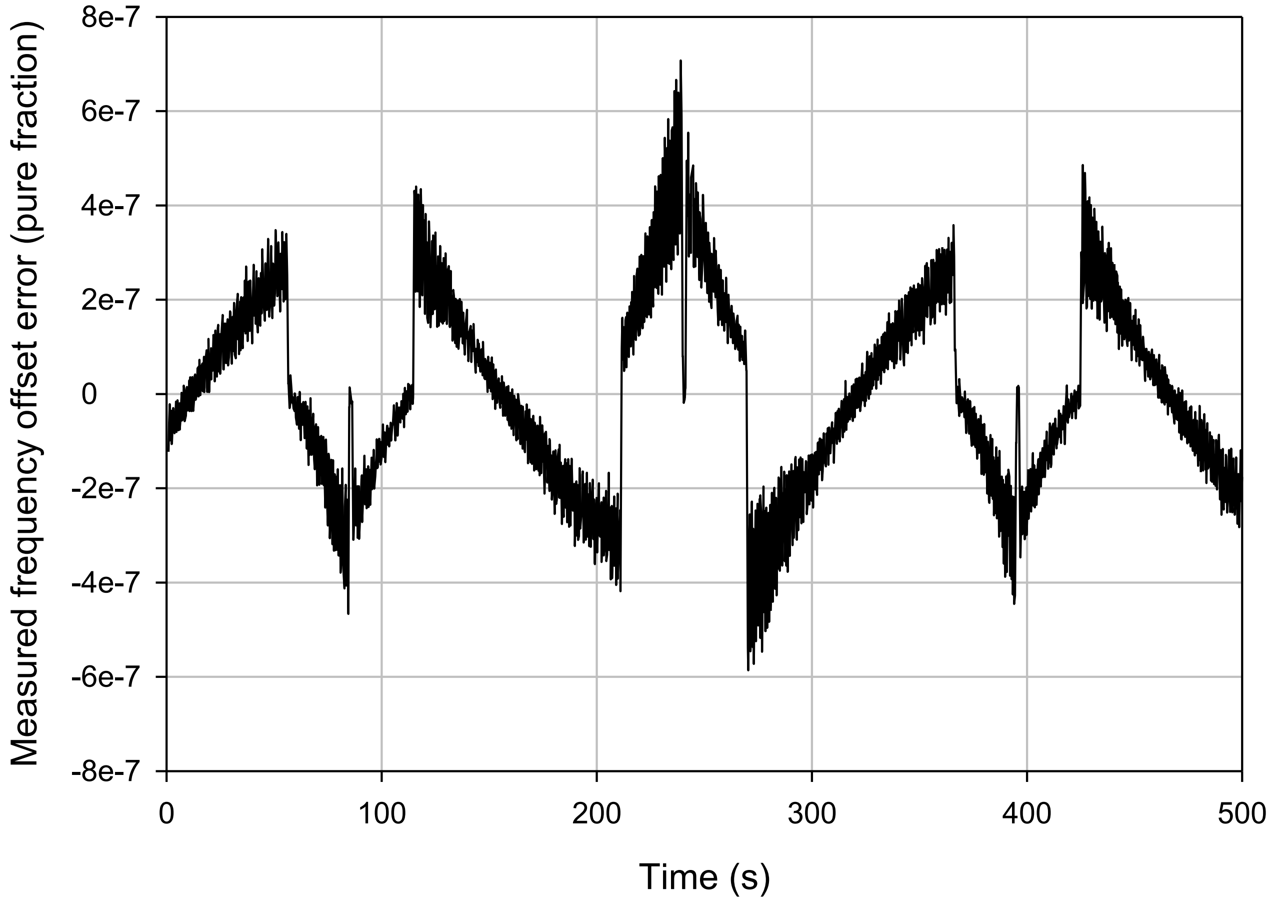


Case G – Node 2 Local Clock FFO

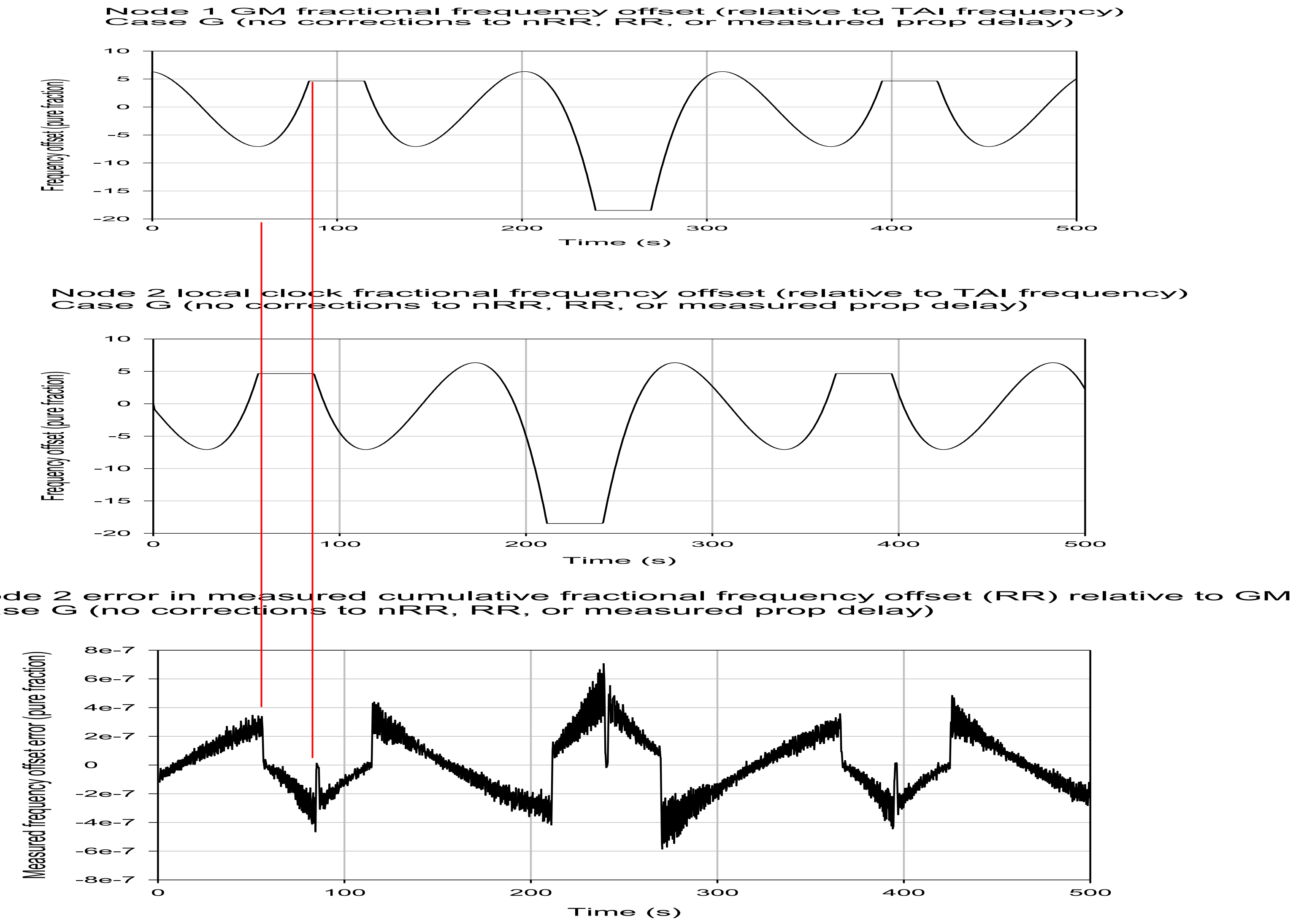


Case G – Node 2 Measured FFO Error

Node 2 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Nodes 1 and 2 FFO plots shown for comparison

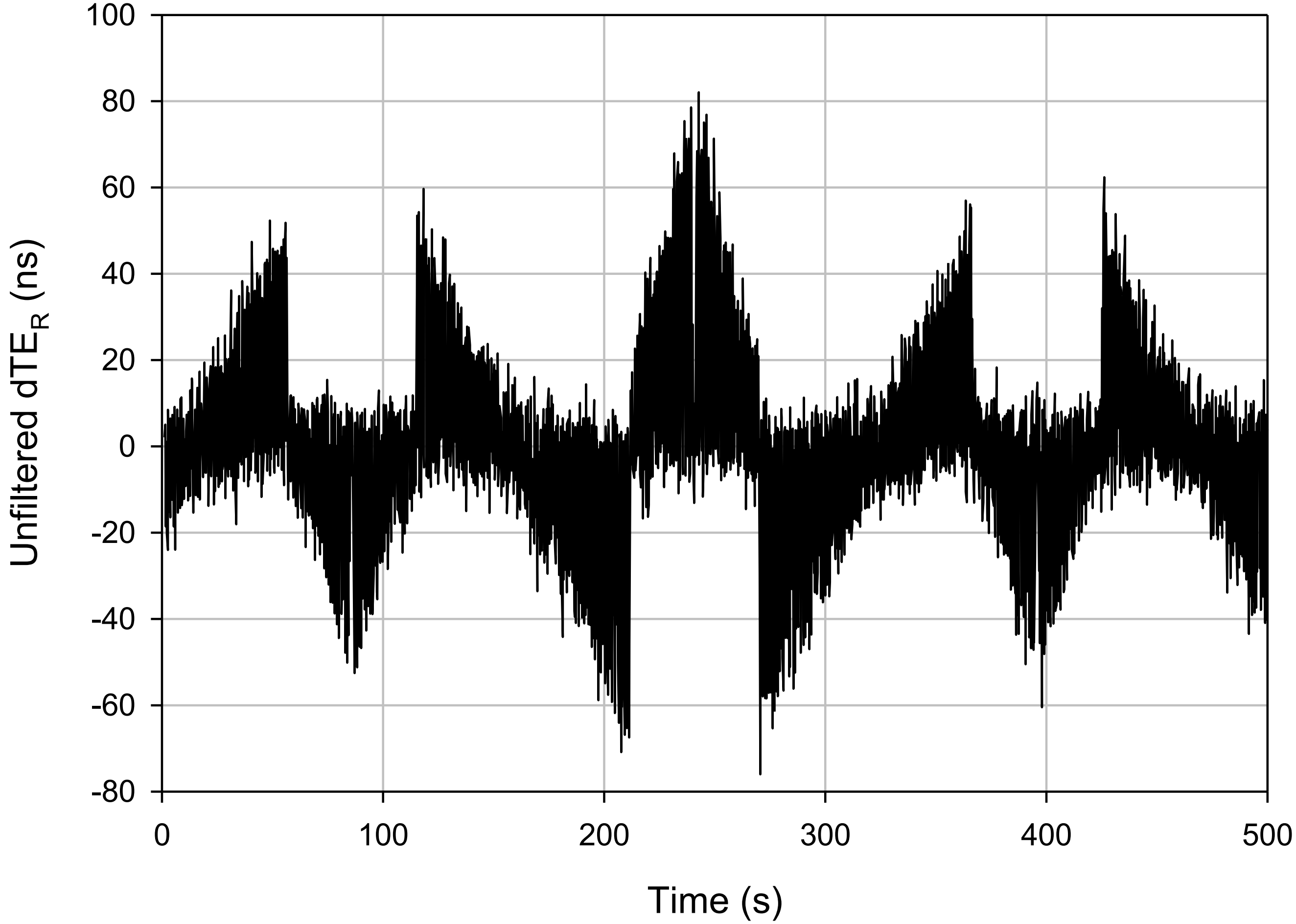


Case G – Nodes 1 and 2 Measured FFO Error

- Slides 33 and 34 show the actual FFO at node 1 (GM) and node 2 (first downstream node)
  - The actual FFO agrees with results in previous presentations (see, for example, slide 40 of [1]); the FFO is due to the temperature profile and frequency versus temperature stability model
- Slide 35 shows the error in measured FFO at node 2, i.e., the error in measured RR
  - This is also the error in measured NRR, since the RR of node 2 relative to the GM (node 1) is the NRR
- The maximum RR error absolute value is approximately 0.7 ppm
- In addition, the RR error jumps by approximately 0.4 ppm when the slope of FFO changes abruptly; this occurs when the temperature variation changes from 1°C /s to 0, or vice-versa (the red vertical lines attempt to show the times when the FFO slopes change and jumps in measured RR error occur)
  - Since RR (and NRR) is being measured for node 2 relative to node 1, the jump in measured RR error occurs when the slope of FFO changes at either node
  - In this particular case, the temperature (and FFO) waveforms have small phase difference (i.e., offset); therefore, in some cases successive jumps are close in time, i.e., the measured RR (and NRR) error jumps by 0.4 ppm and then quickly jumps back

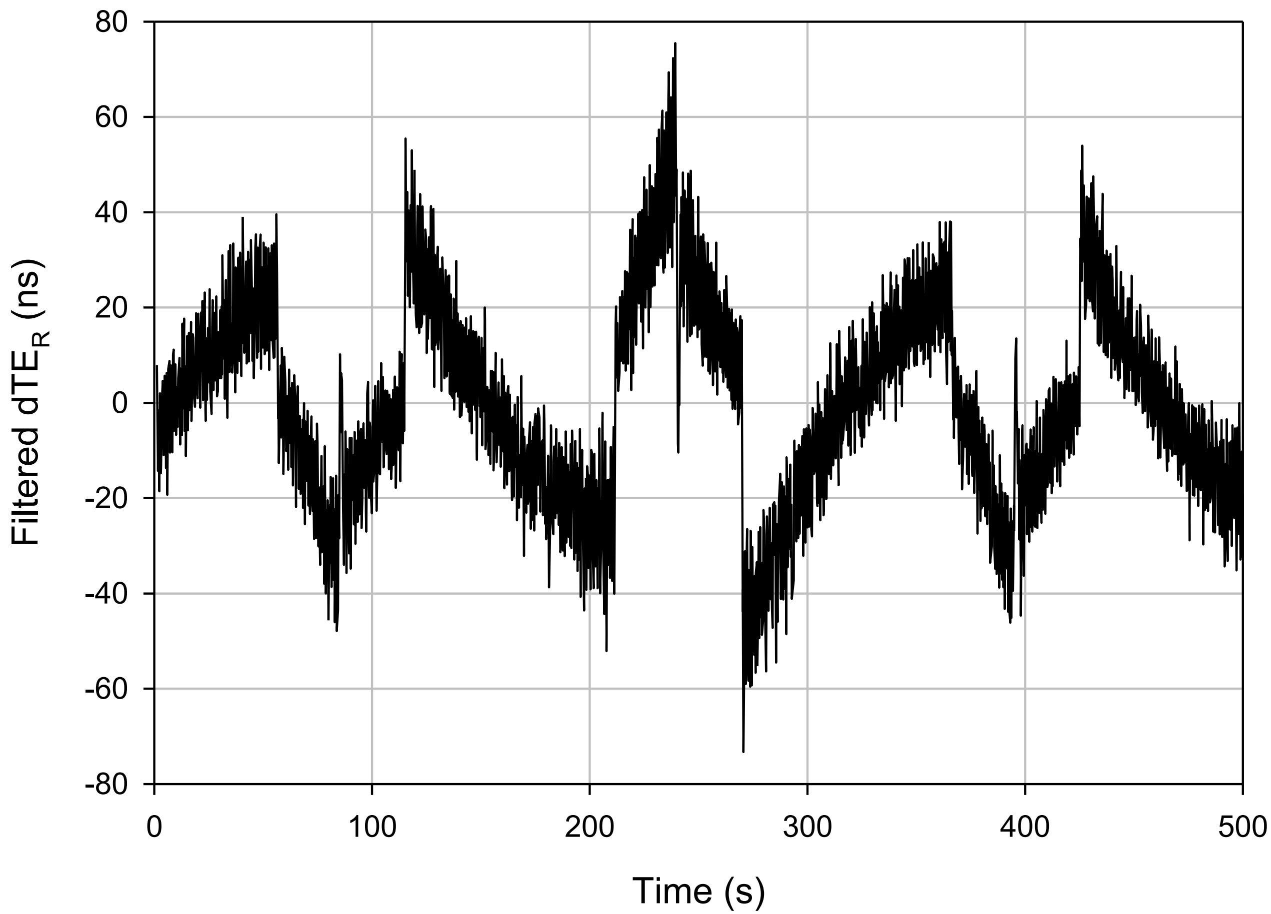
Case G, Node 2, Unfiltered dTE_R

Node 2 dTE_R (relative to GM), before PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 2, Filtered dTE_R

Node 2 dTE_R (relative to GM), after PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)

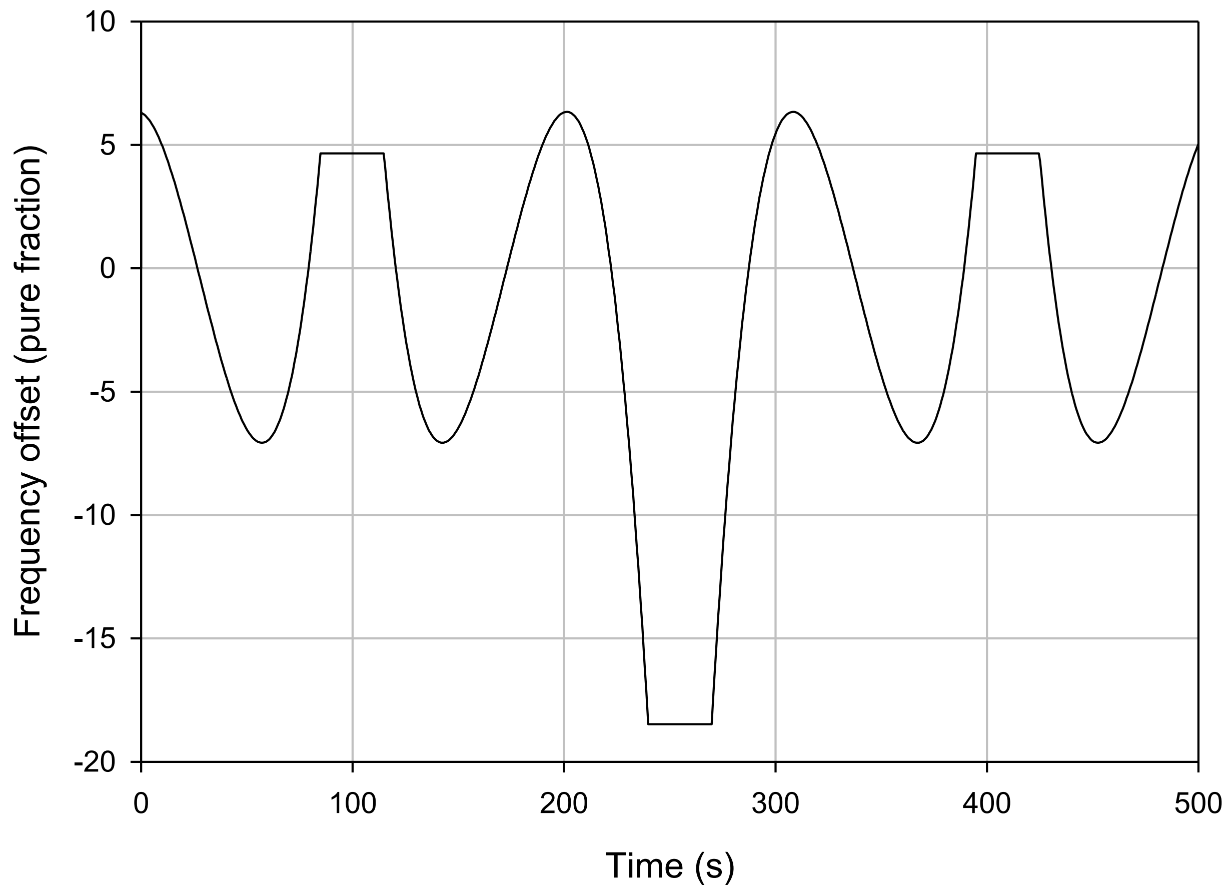


Case G, Node 2, Unfiltered and Filtered dTE_R

- For unfiltered dTE_R, an instantaneous correction is made on receipt of every Sync message (i.e., every 125 ms)
- A new NRR measurement is made every on each Pdelay exchange, but using a window of 3 Pdelay exchanges
- The Pdelay interval is also 125 ms; however, the Pdelay exchanges are not synchronized with the sending or receipt of Sync
- In between RR measurements, the unfiltered synchronized time can be in error by as much as the measured FFO (i.e., RR) error, multiplied by the time since receipt of the most recent Sync message
- Therefore, the envelope of the unfiltered dTE_R time history follows the measured FFO error
- The 2.6 Hz endpoint filter removes higher dTE_R frequencies
- The filtered dTE_R and the measured FFO (RR) error have similar shapes

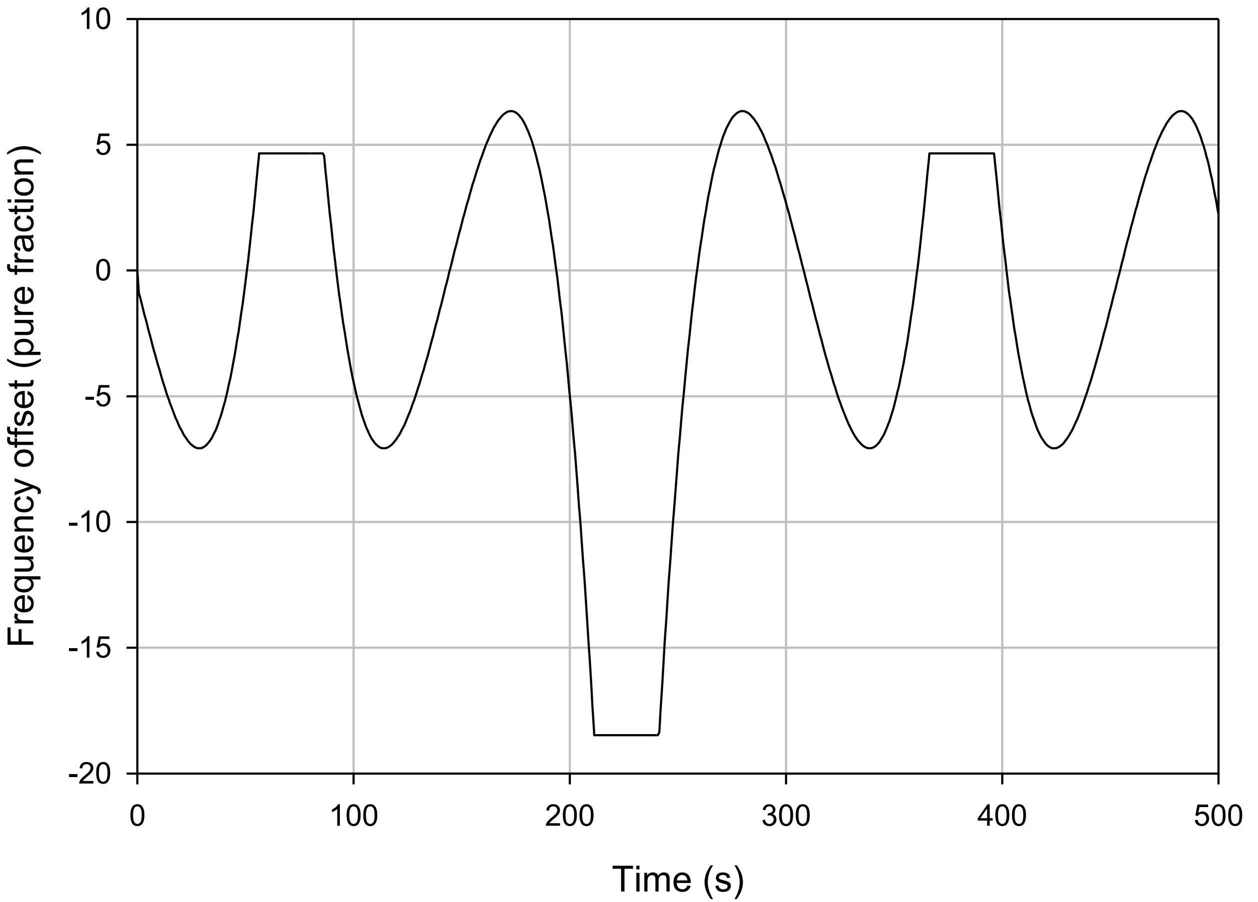
Case H – Node 1 (GM) FFO

Node 1 GM fractional frequency offset (relative to TAI frequency)
Case H (no corrections to nRR, RR, or measured prop delay)



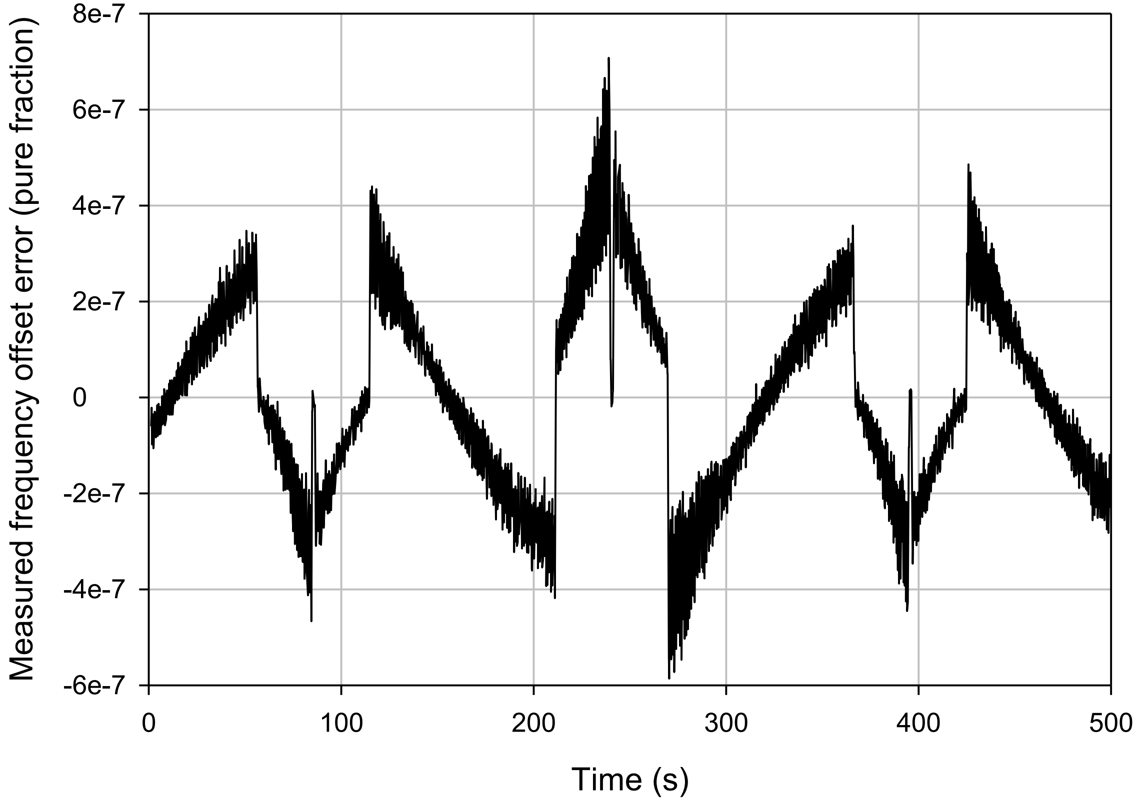
Case H – Node 2 Local Clock FFO

Node 2 local clock fractional frequency offset (relative to TAI frequency)
Case H (no corrections to nRR, RR, or measured prop delay)

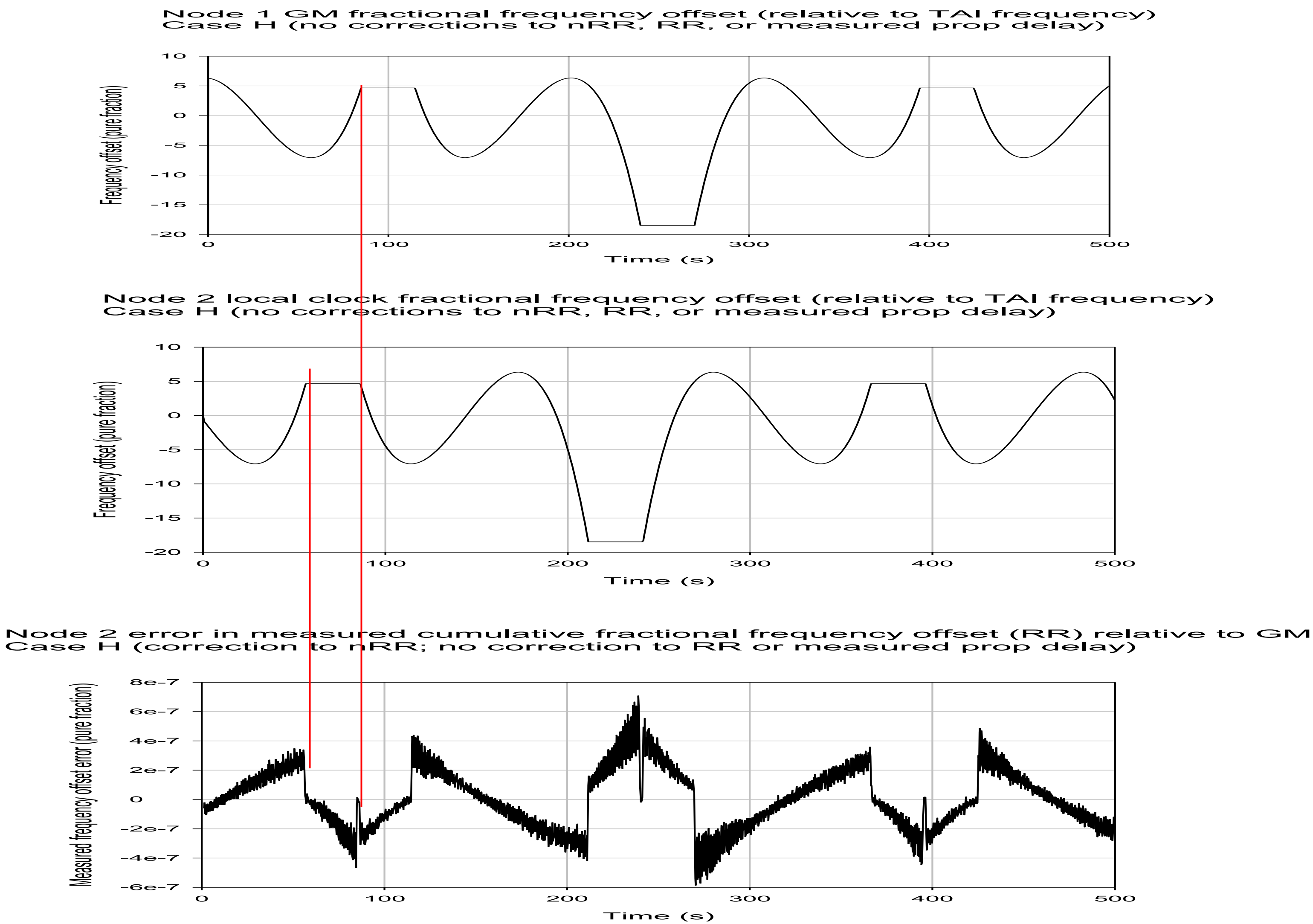


Case H – Node 2 Measured FFO Error

Node 2 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)

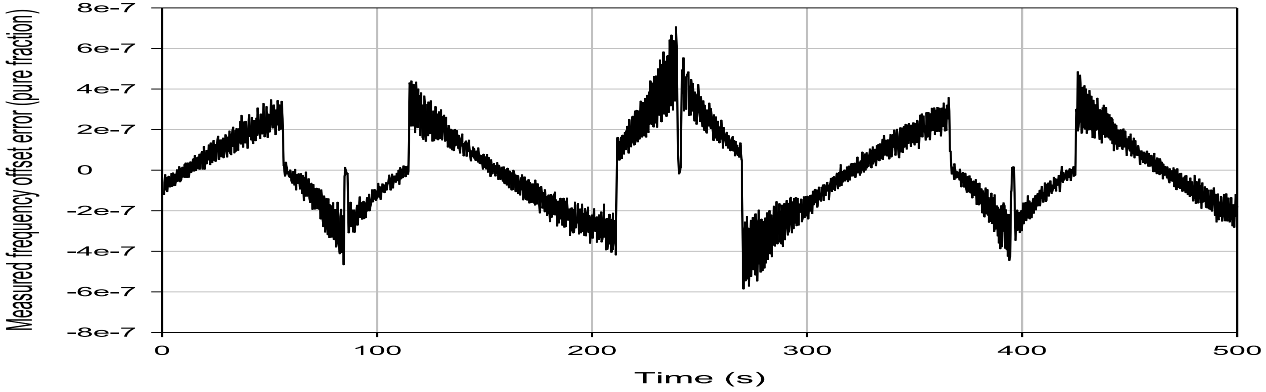


Case H, Nodes 1 and 2 FFO plots shown for comparison

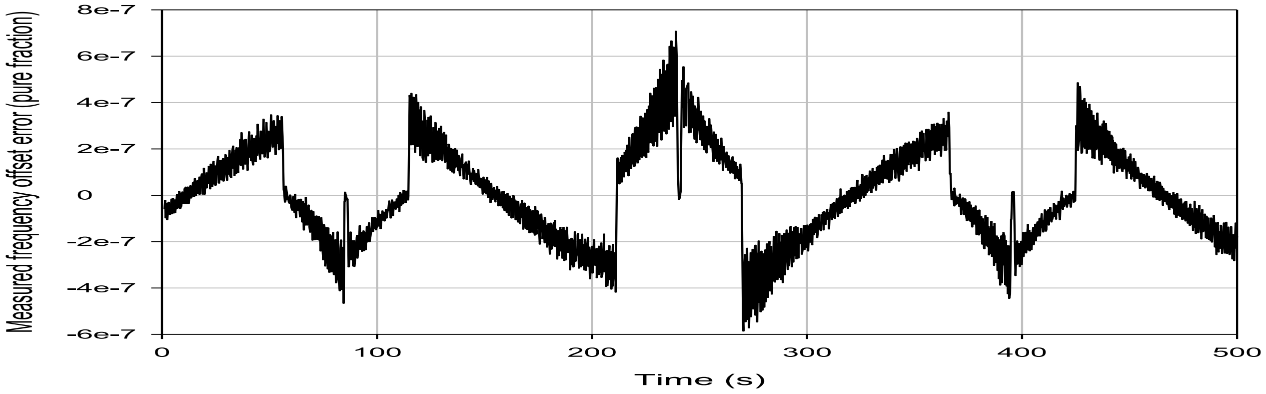


Comparison of Cases G and H, Node 2 Measured FFO Error

Node 2 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



Node 2 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)

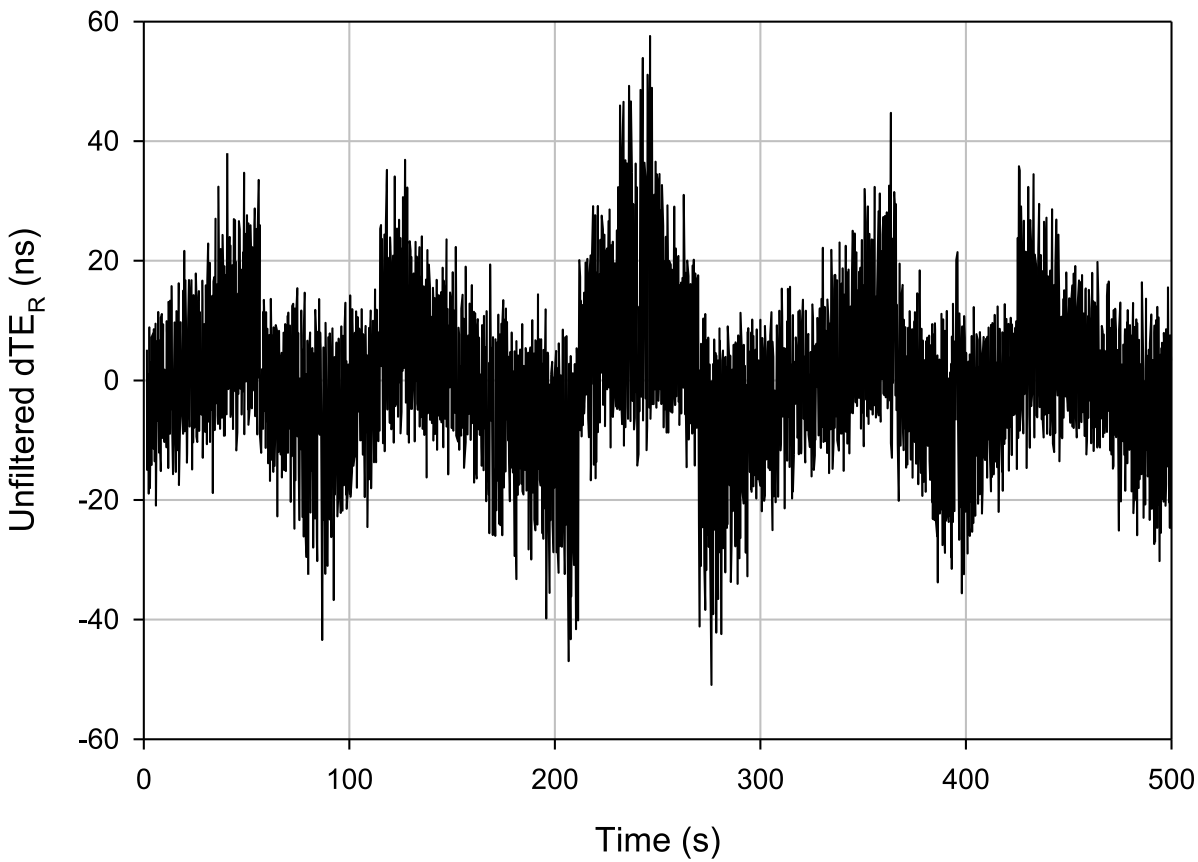


Case G – Nodes 1 and 2 Measured FFO Error

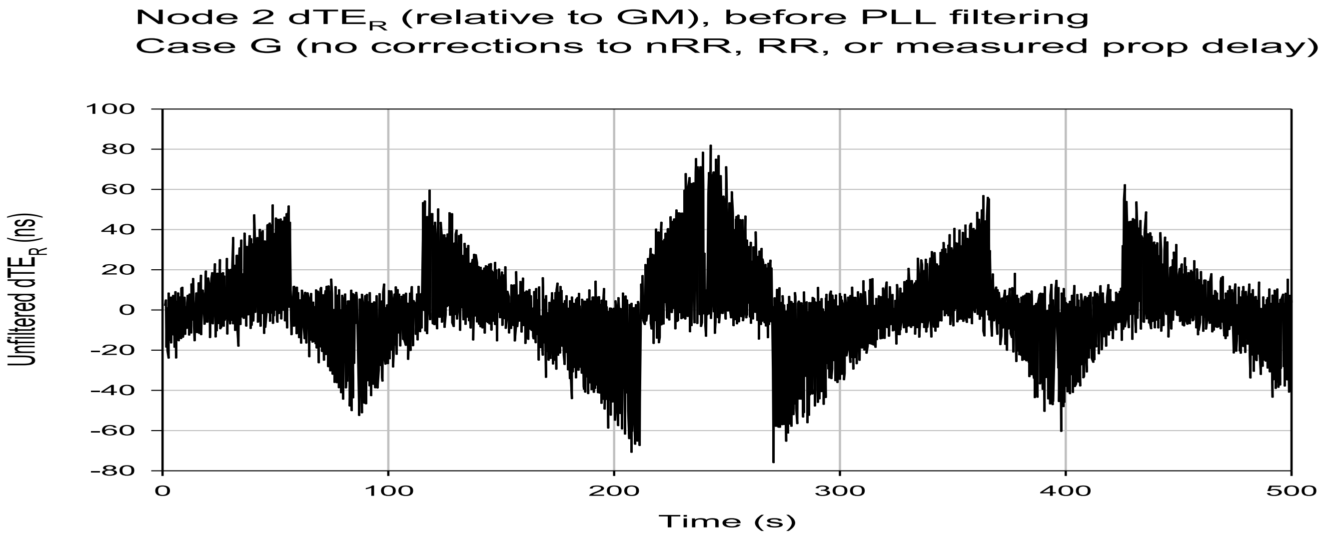
- The measured FFO (RR) error for case H, node 2 is very similar to that for case G, node 2, both for the shape of the FFO error waveform and its magnitude
- The next four slides show that filtered and unfiltered dTE_R for case H, node 2 shows some improvement compared to Case G
  - The shape is similar, but the magnitude of the error is smaller

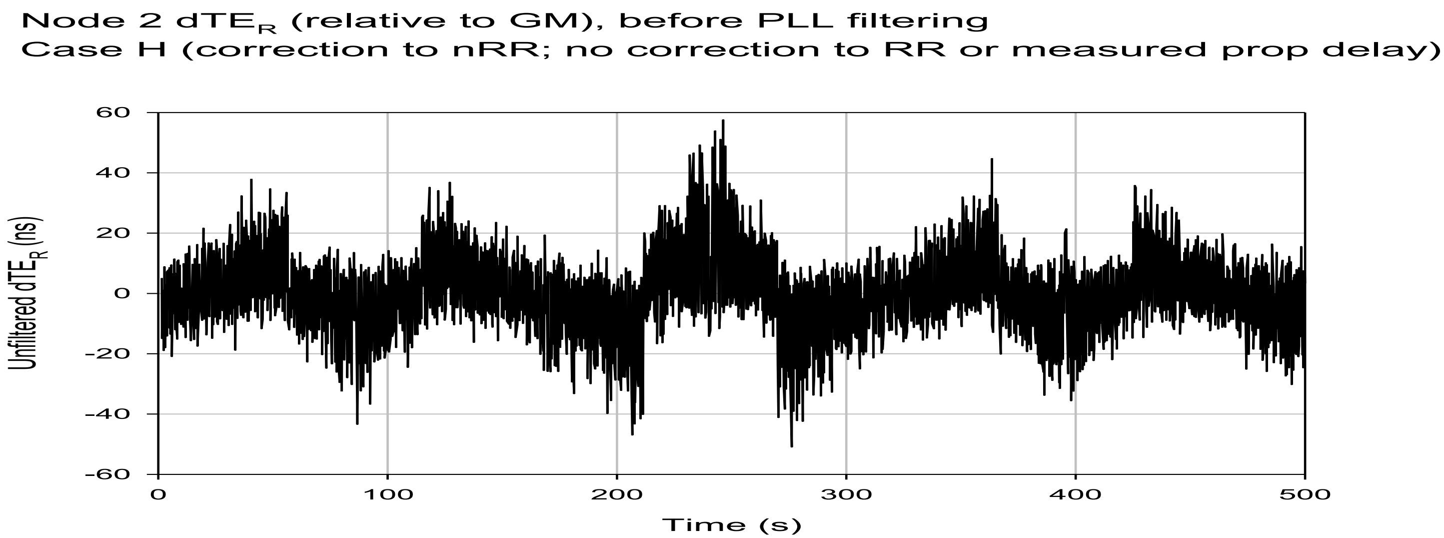
Case H, Node 2, Unfiltered dTE_R

Node 2 dTE_R (relative to GM), before PLL filtering
Case H (correction to nRR; no correction to RR or measured prop delay)



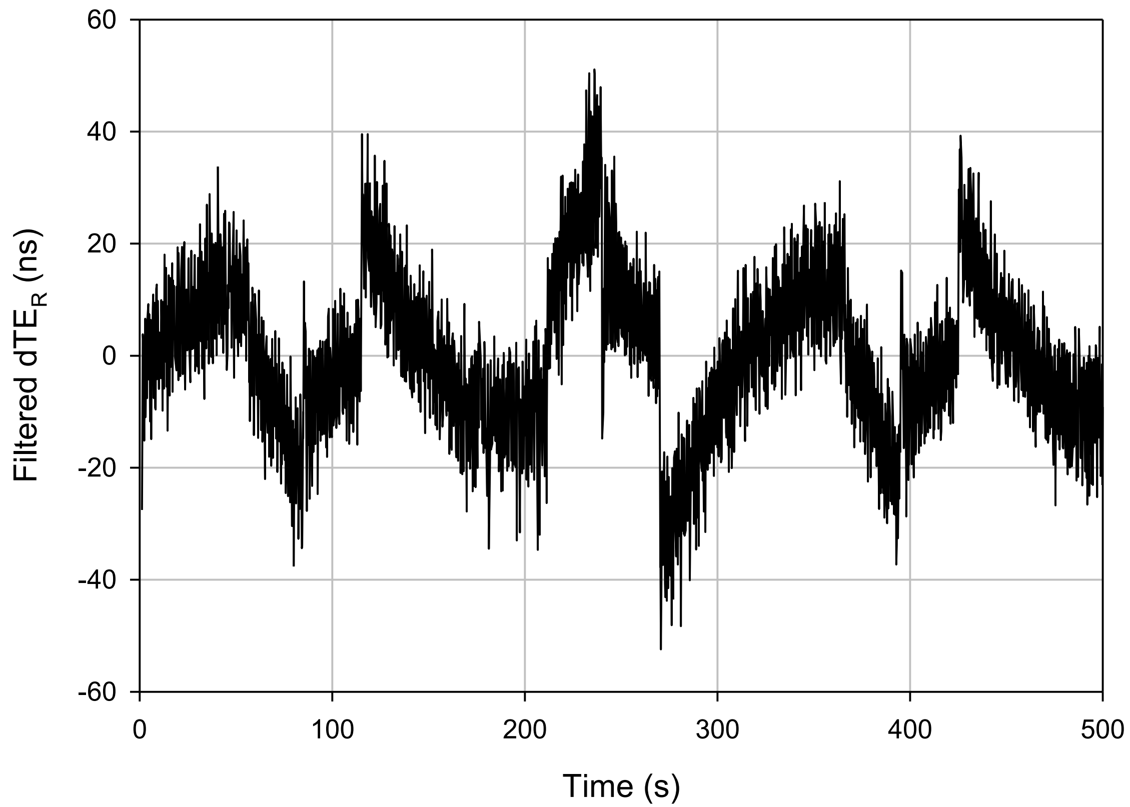
Cases G and H, Node 2, Unfiltered dTE_R Comparison



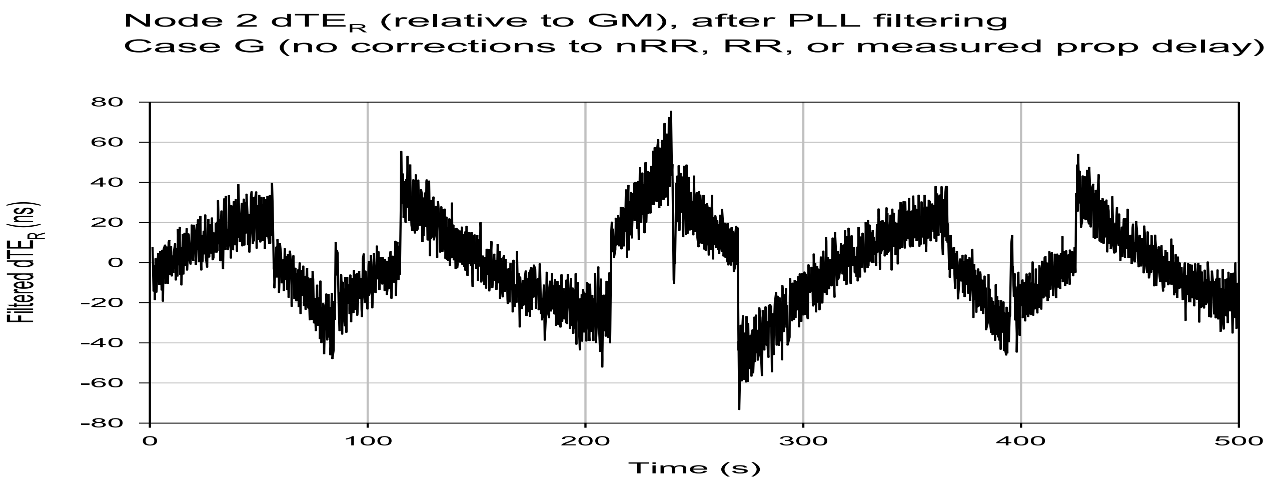


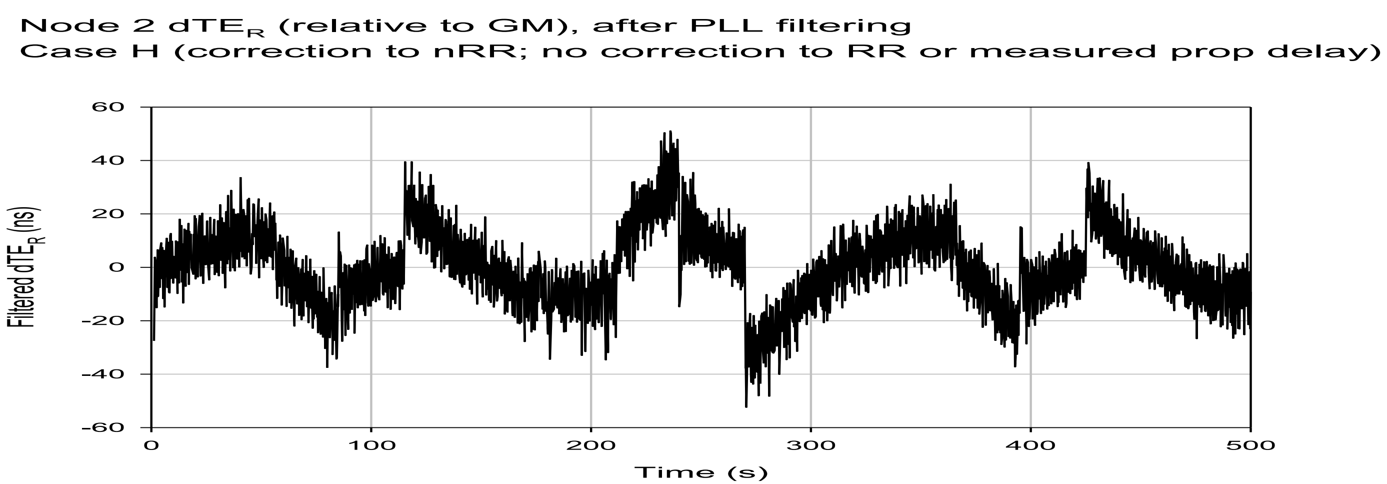
Case H, Node 2, Filtered dTE_R

Node 2 dTE_R (relative to GM), after PLL filtering
Case H (correction to nRR; no correction to RR or measured prop delay)



Cases G and H, Node 2, Filtered dTE_R Comparison



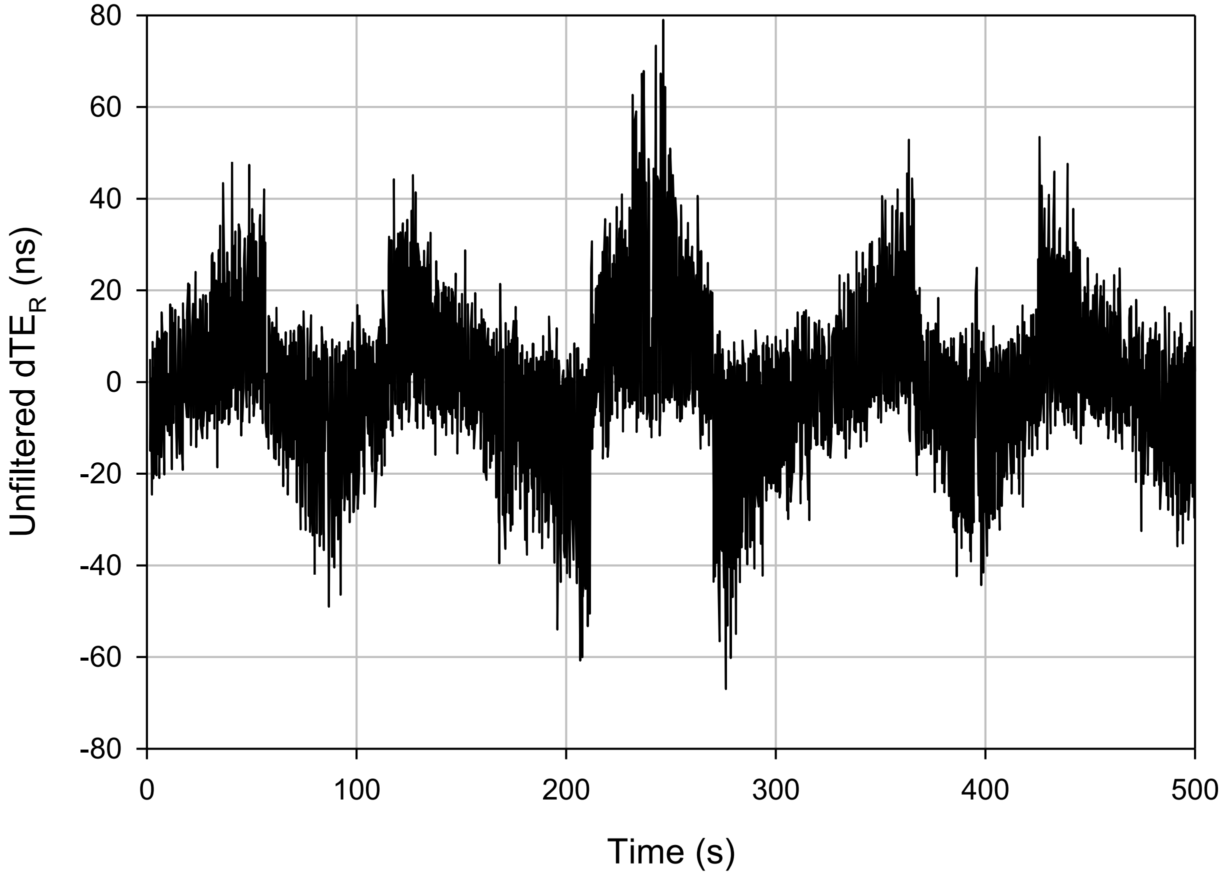


Cases I and J, Node 2, Unfiltered and Filtered dTE_R

- Unfiltered and filtered dTE_R waveforms are shown on the next two slides
- Case I max|dTE_R| for the 500 s shown larger than the corresponding results for case H, and approximately the same as the corresponding results for case G
- Envelopes of the filtered and unfiltered dTE_R waveforms are similar to the measured FFO (RR) error for case H, node 2
- Results for case J are very similar to those for case I

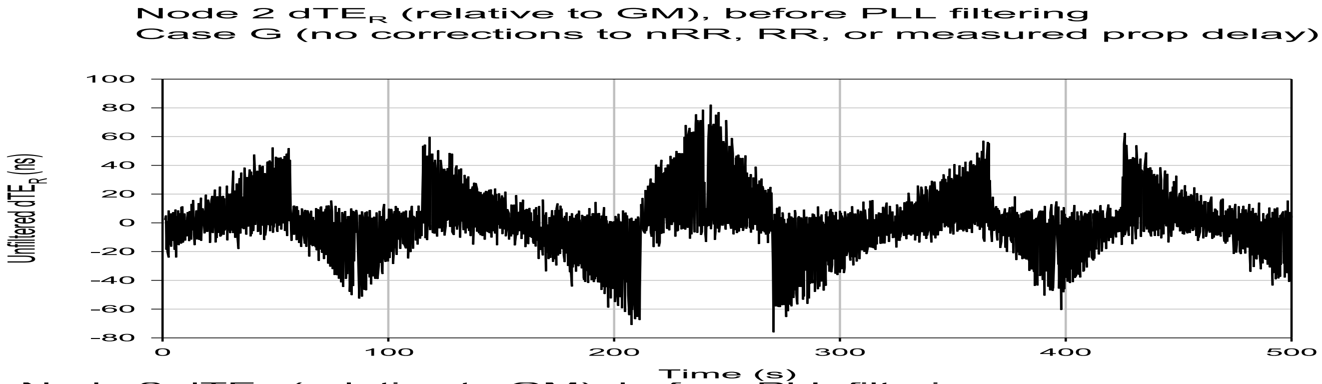
Case I, Node 2, Unfiltered dTE_R

Node 2 dTE_R (relative to GM), before PLL filtering
Case I (correction to nRR+RR; no correction to measured prop delay)

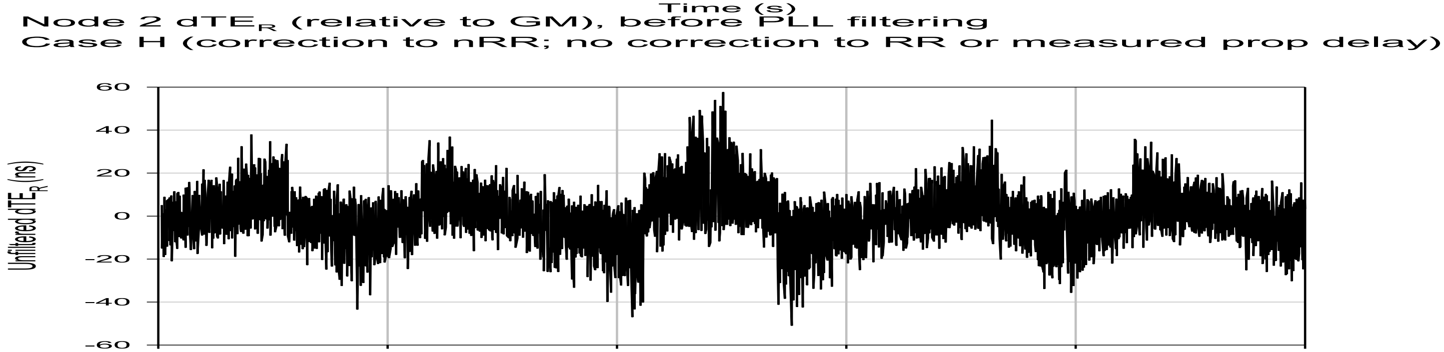


Cases G, H, and I, Node 2, Unfiltered dTE_R Comparison

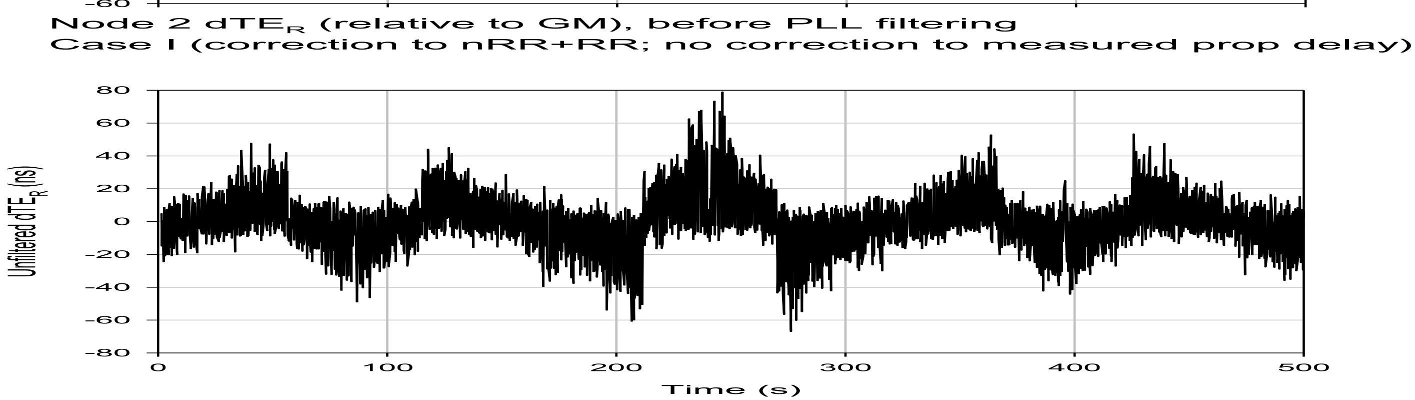
Case G



Case H



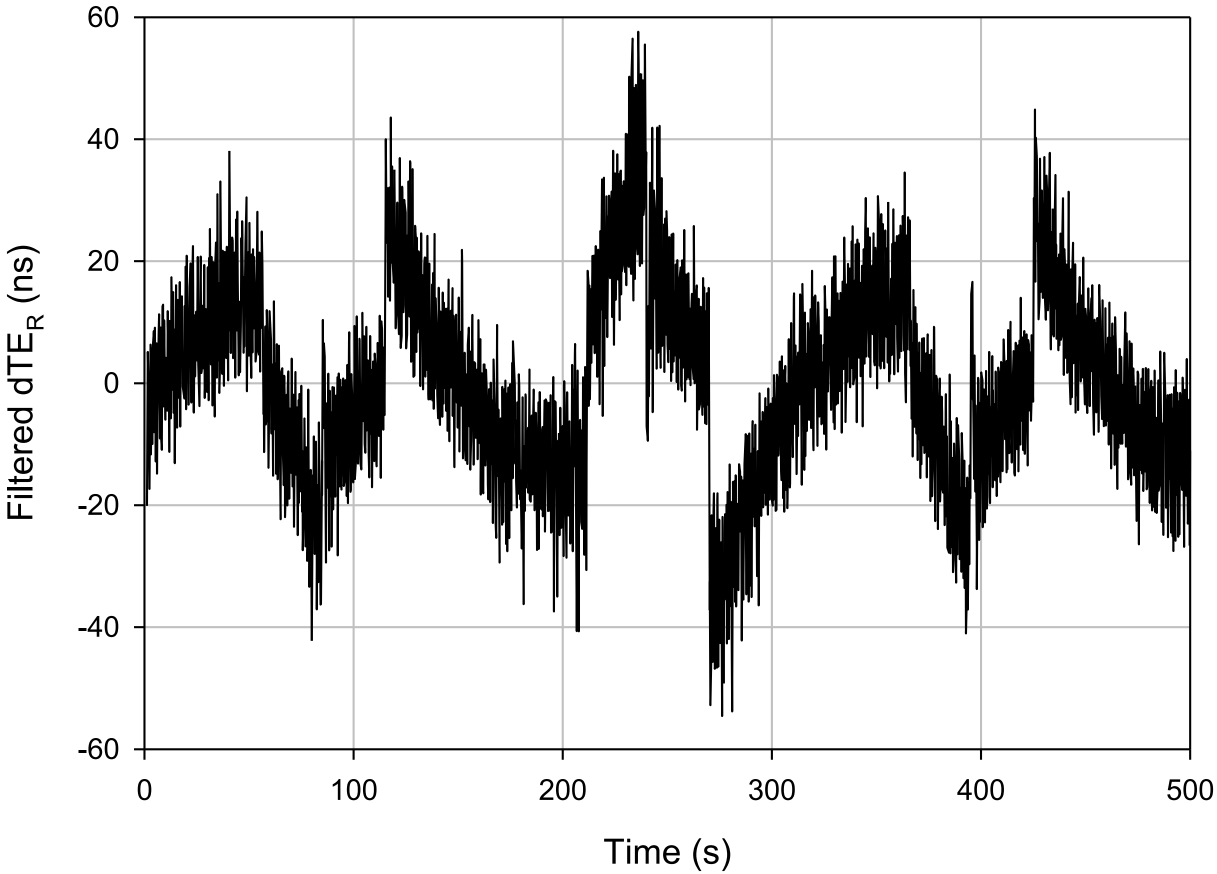
Case I



Case I, Node 2, Filtered dTE_R

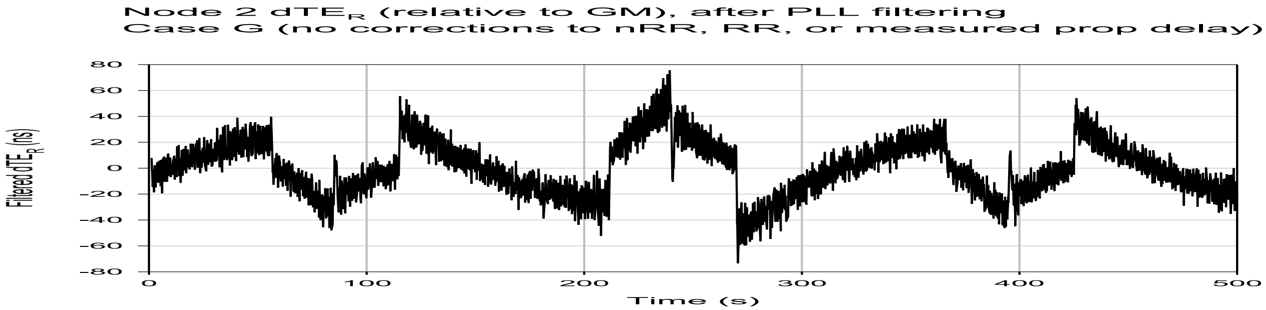
Node 2 dTE_R (relative to GM), after PLL filtering

Case I (correction to nRR and RR; no correction to measured prop delay)

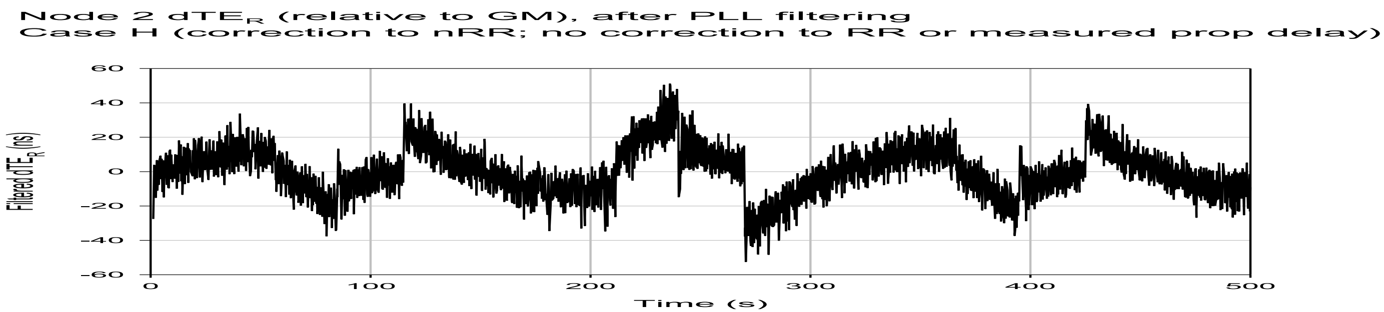


Cases G, H, and I, Node 2, Filtered dTE_R Comparison

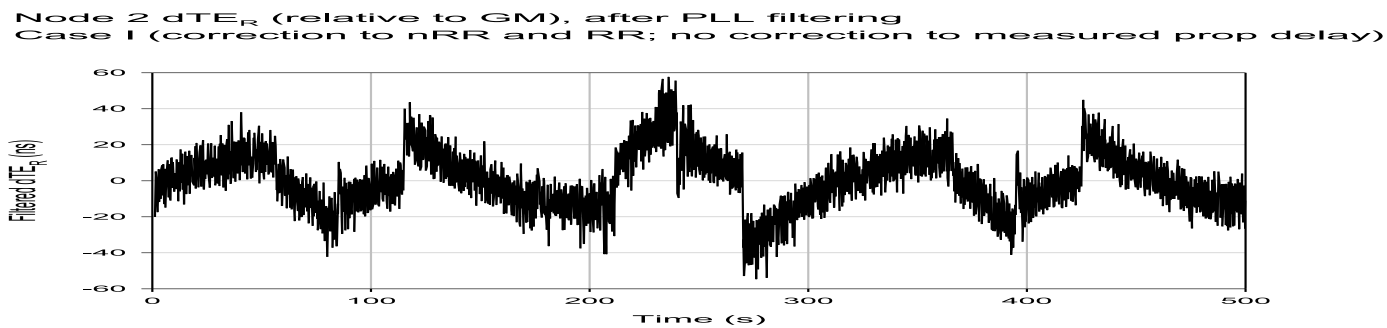
Case G



Case H



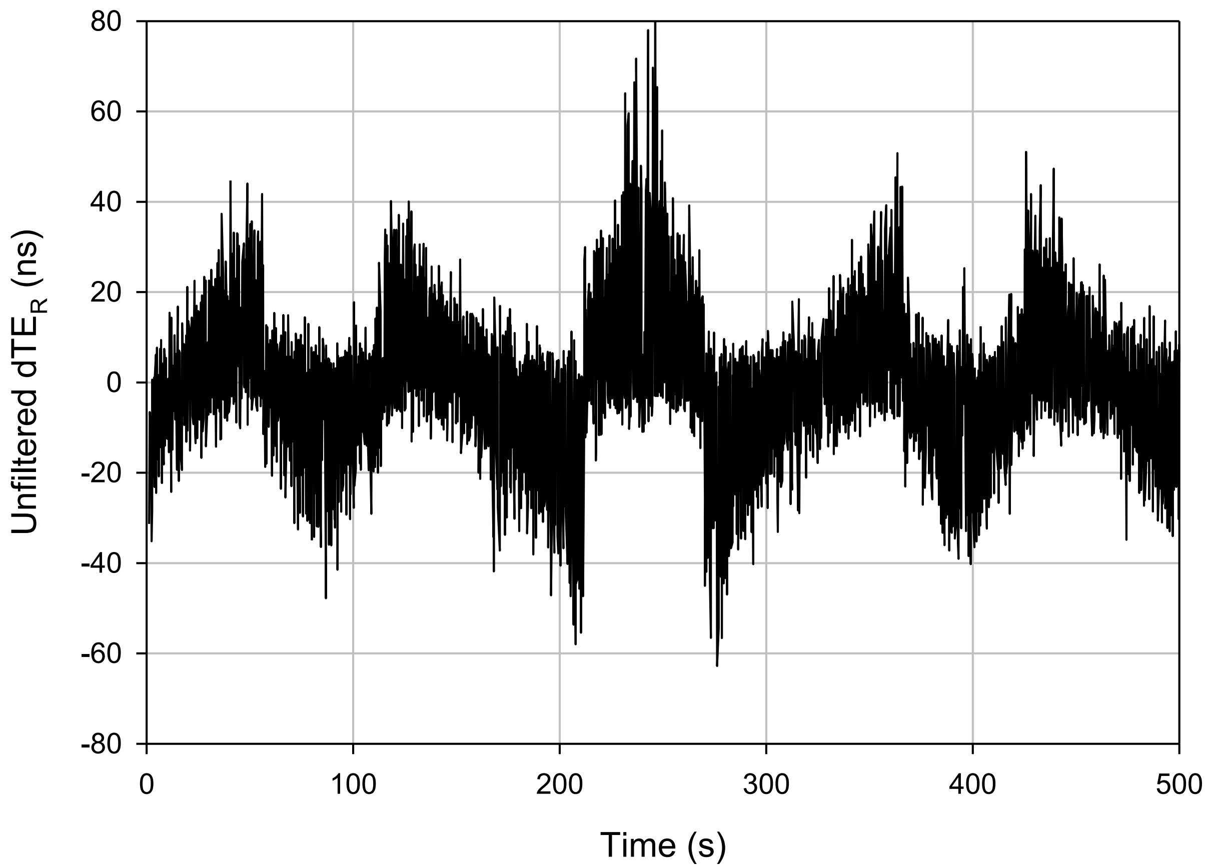
Case I



Case J, Node 2, Unfiltered dTE_R

Node 2 dTE_R (relative to GM), before PLL filtering

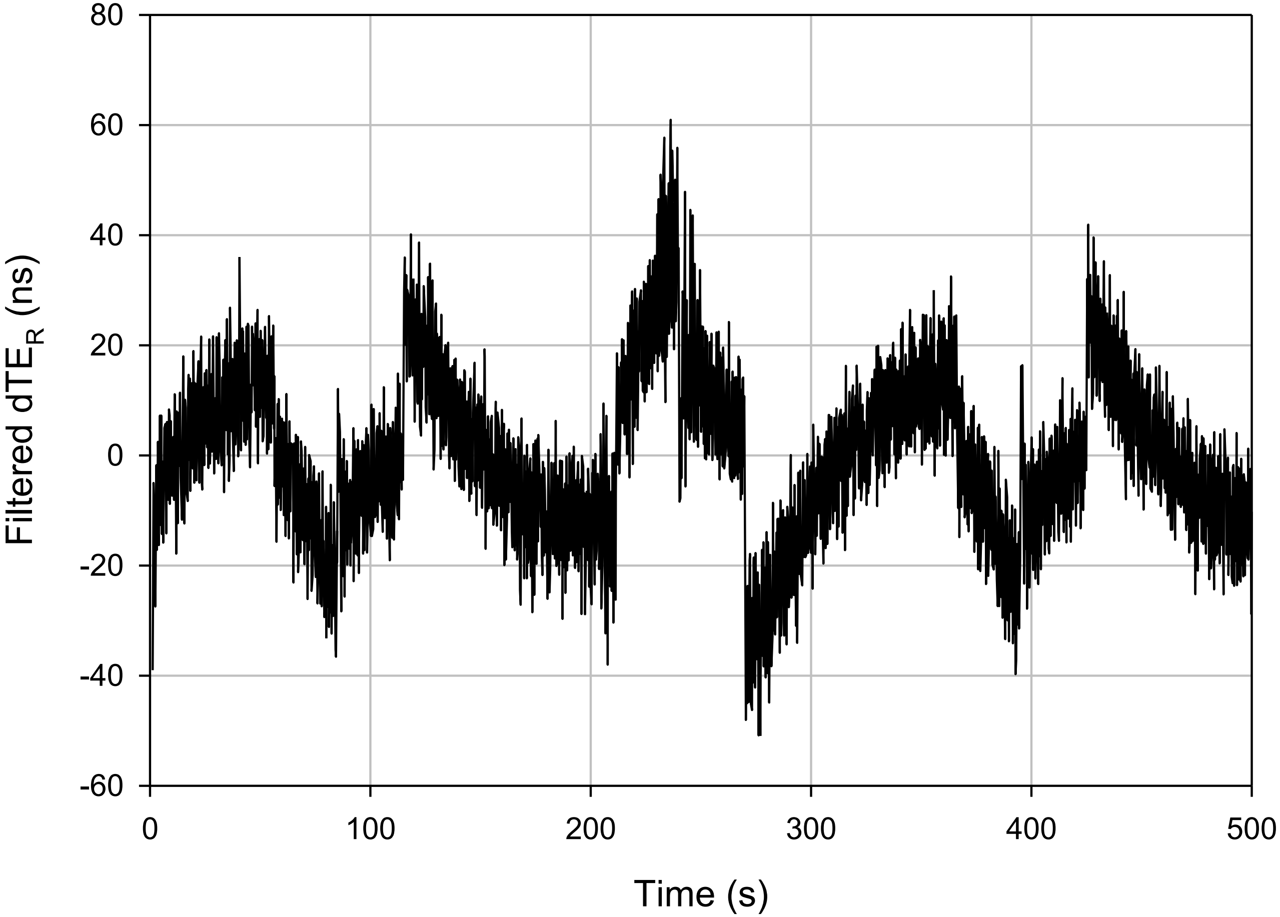
Case J (correction to nRR+RR; no correction to measured prop delay)



Case J, Node 2, Filtered dTE$_R$

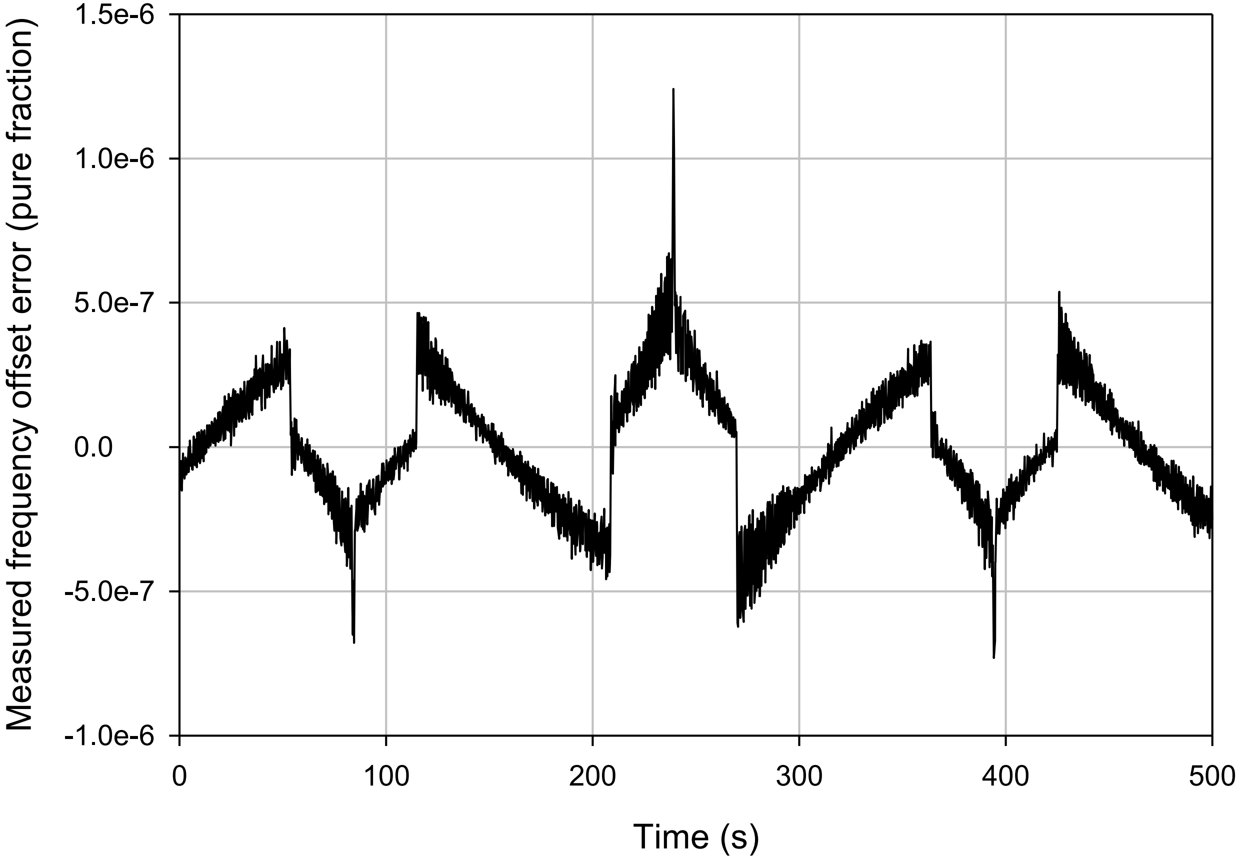
Node 2 dTE$_R$ (relative to GM), after PLL filtering

Case J (correction to nRR and RR; no correction to measured prop delay)



Case G, Node 3, Measured FFO Error

Node 3 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



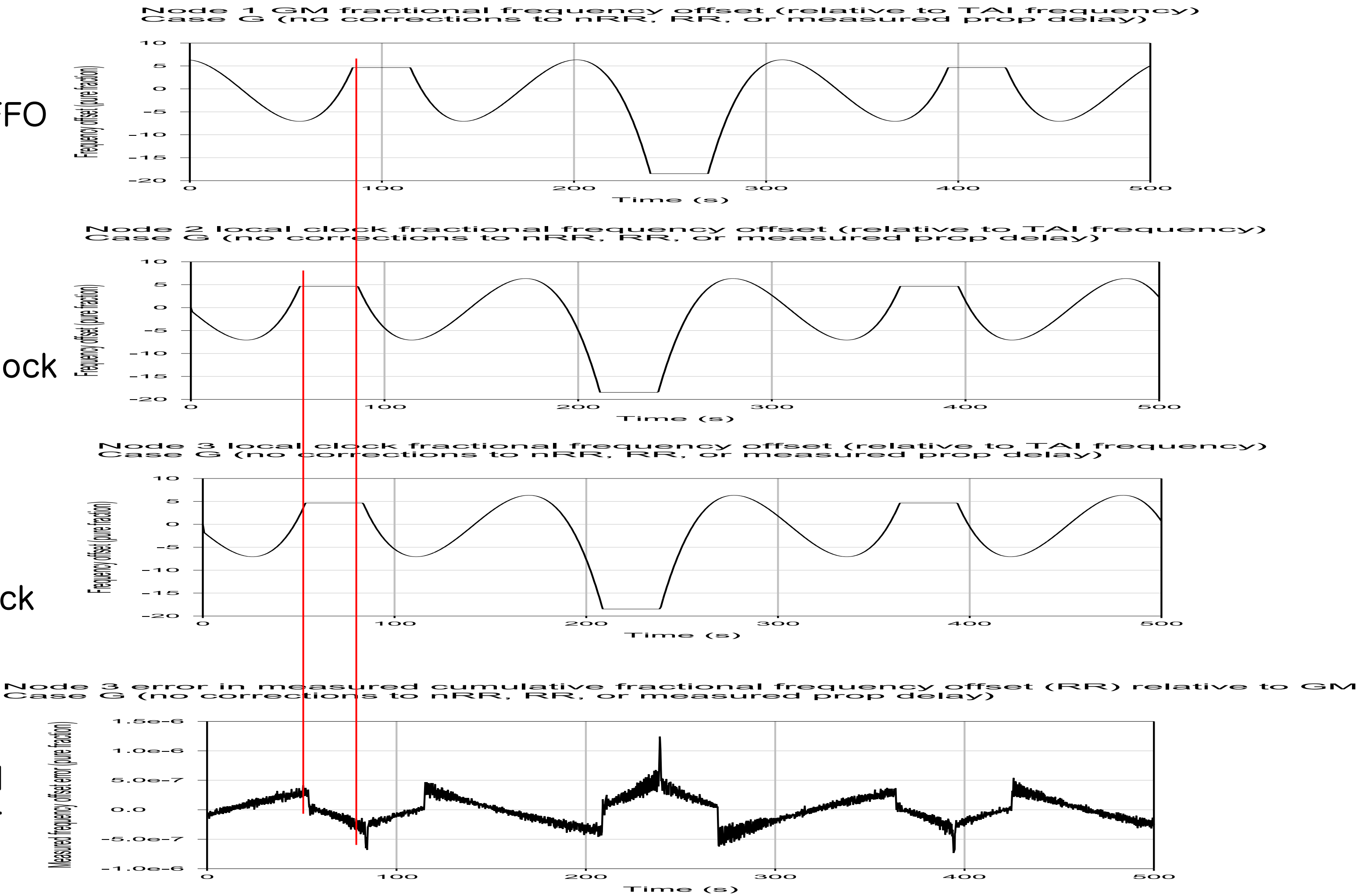
Case G, Nodes 1, 2, and 3 FFO plots

GM FFO

Node 2 Local Clock FFO

Node 3 Local Clock FFO

Node 3 Measured FFO Error

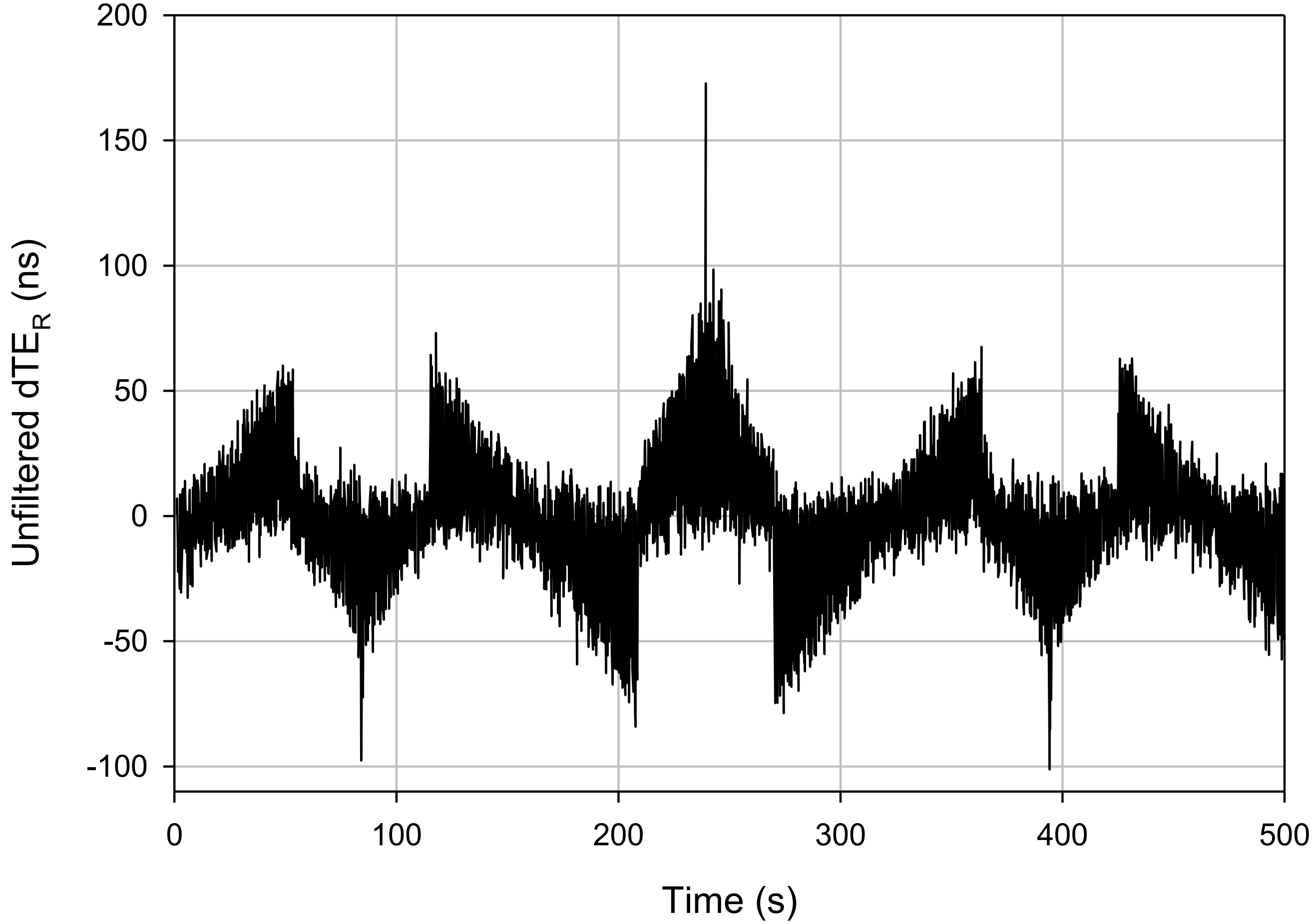


Case G, Node 3 Measured FFO (RR) Error

- For node 3, the measured FFO (RR) is the sum of NRR at node 2 and NRR at node 3
- This means that the jumps in measured FFO error at node 3 depends on the points in the local clock FFO waveforms where the slope changes abruptly (i.e., breakpoints), for nodes 1, 2, and 3
- In the single replication run here, some of the breakpoints in the different FFO waveforms occur closely in time, which results in larger jumps in measured FFO error
- The maximum FFO error (absolute value) for node 3 is approximately 1.25 ppm, compared to 0.7 ppm for node 2 (case G) (see slide 34)
- The next two slides show unfiltered and filtered dTE_R waveforms for case G, node 3; like node 2, the envelopes have similar shape to the FFO error waveform, but with larger spikes compared to node 2 where the node 3 spikes occur

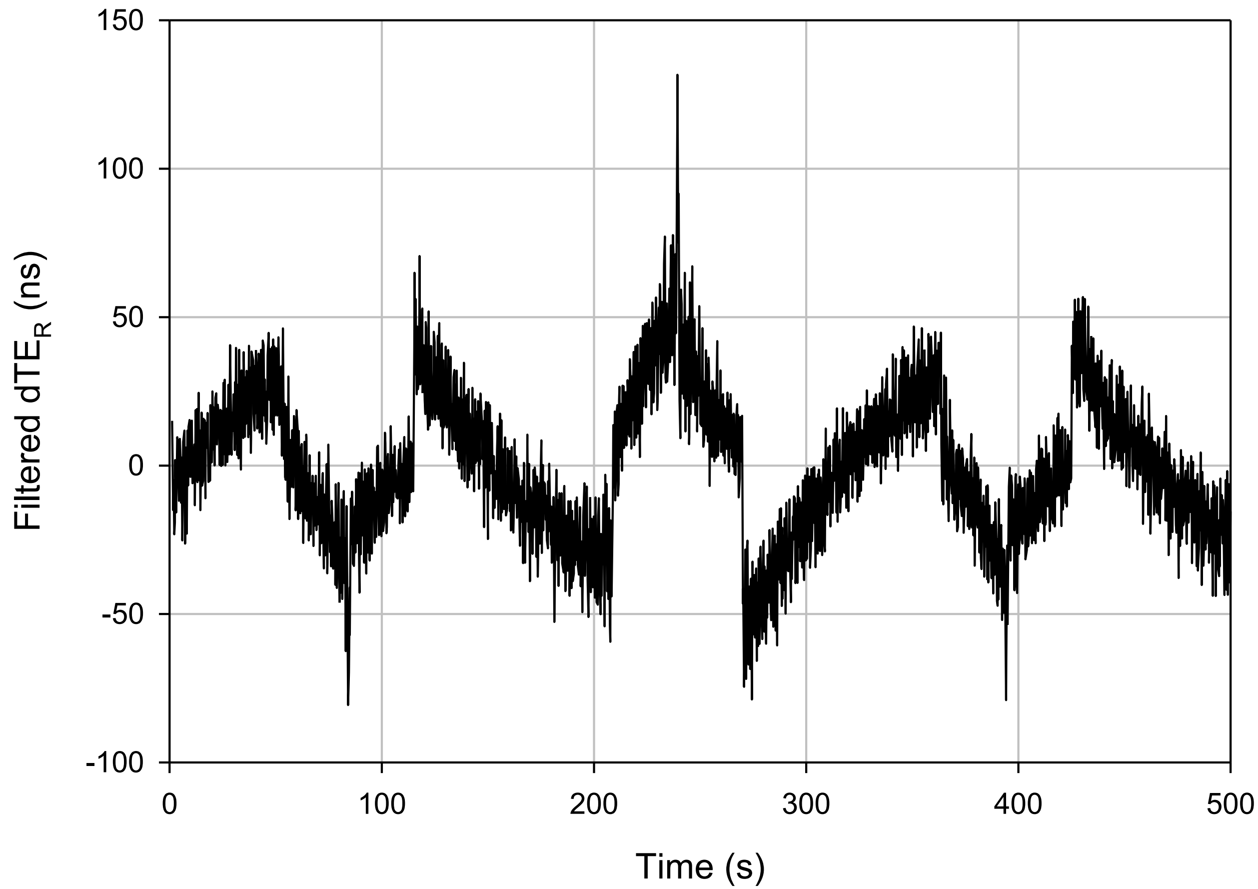
Case G, Node 3, Unfiltered dTE_R

Node 3 dTE_R (relative to GM), before PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 3, Filtered dTE_R

Node 3 dTE_R (relative to GM), after PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)

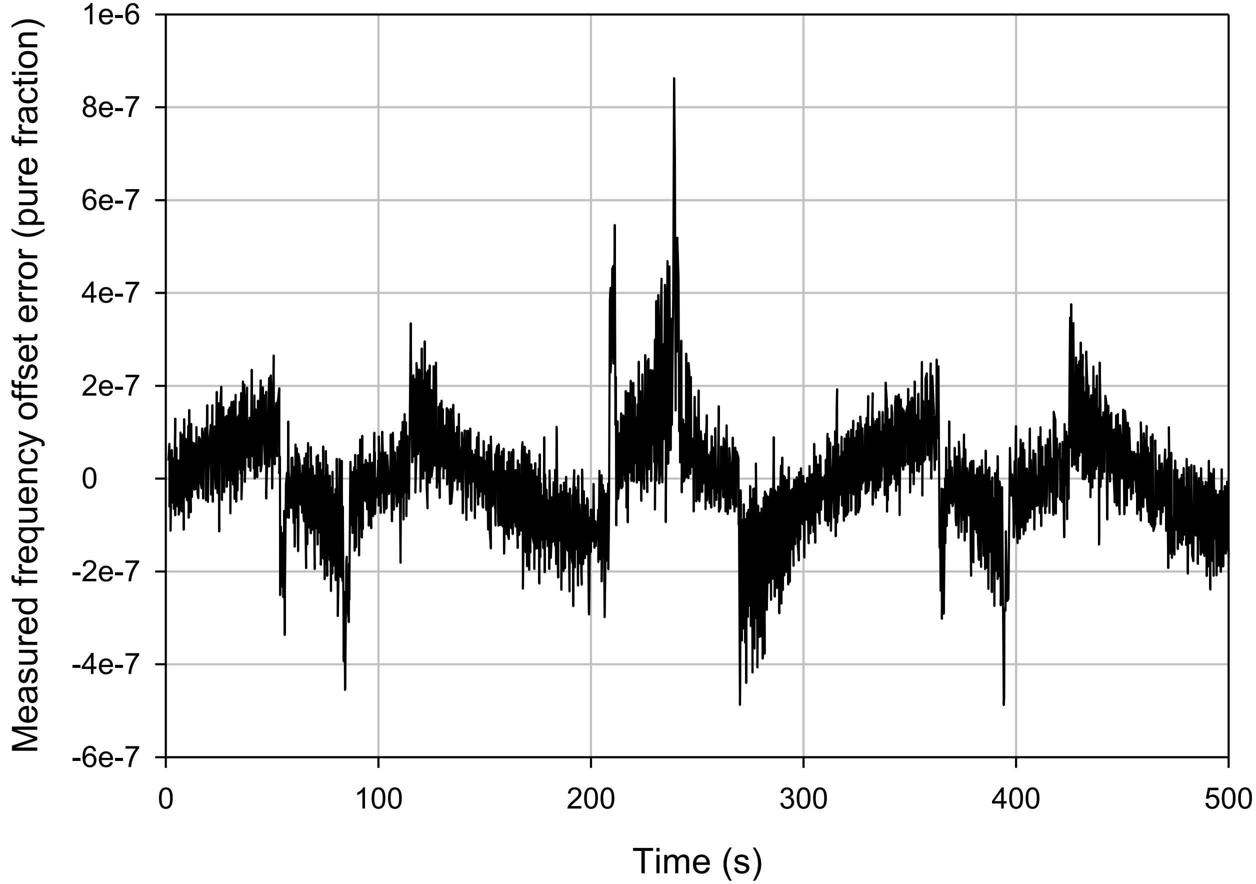


Case H, Node 3

- Measured FFO error for Case H, Node 3 is smaller than for Case G, Node 3 (see next two slides)
- Unfiltered and filtered dTE_R waveforms have envelopes whose shapes are similar to the measured FFO error waveform shape (similar to results for node 2)(see 3rd and 4th slides following)

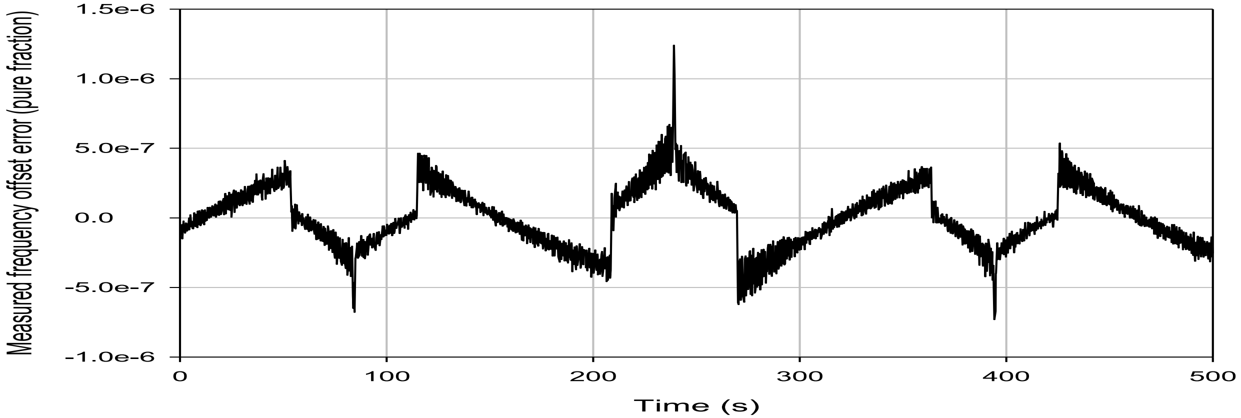
Case H, node 3, Measured FFO (RR) Error

Node 3 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)

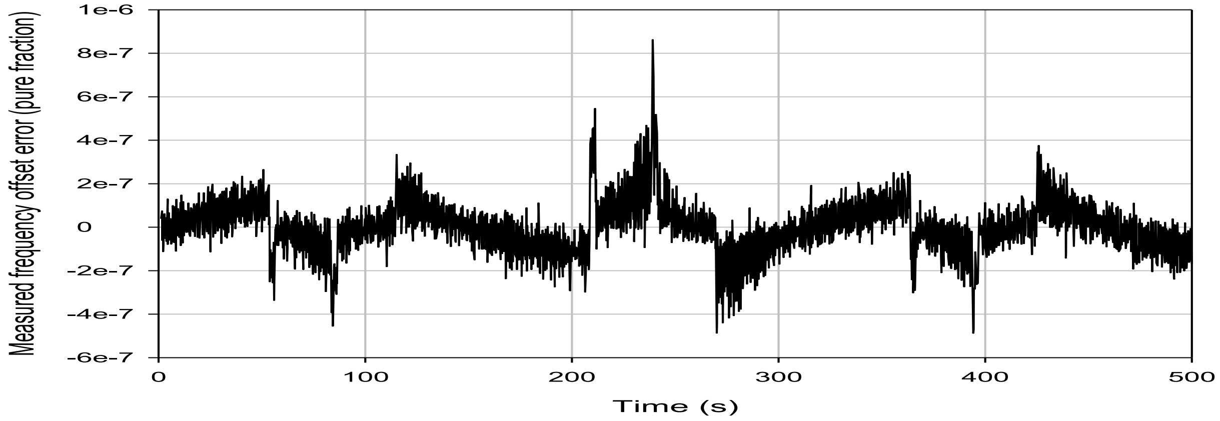


Comparison of Cases G and H, Node 3 Measured FFO Error

Node 3 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)

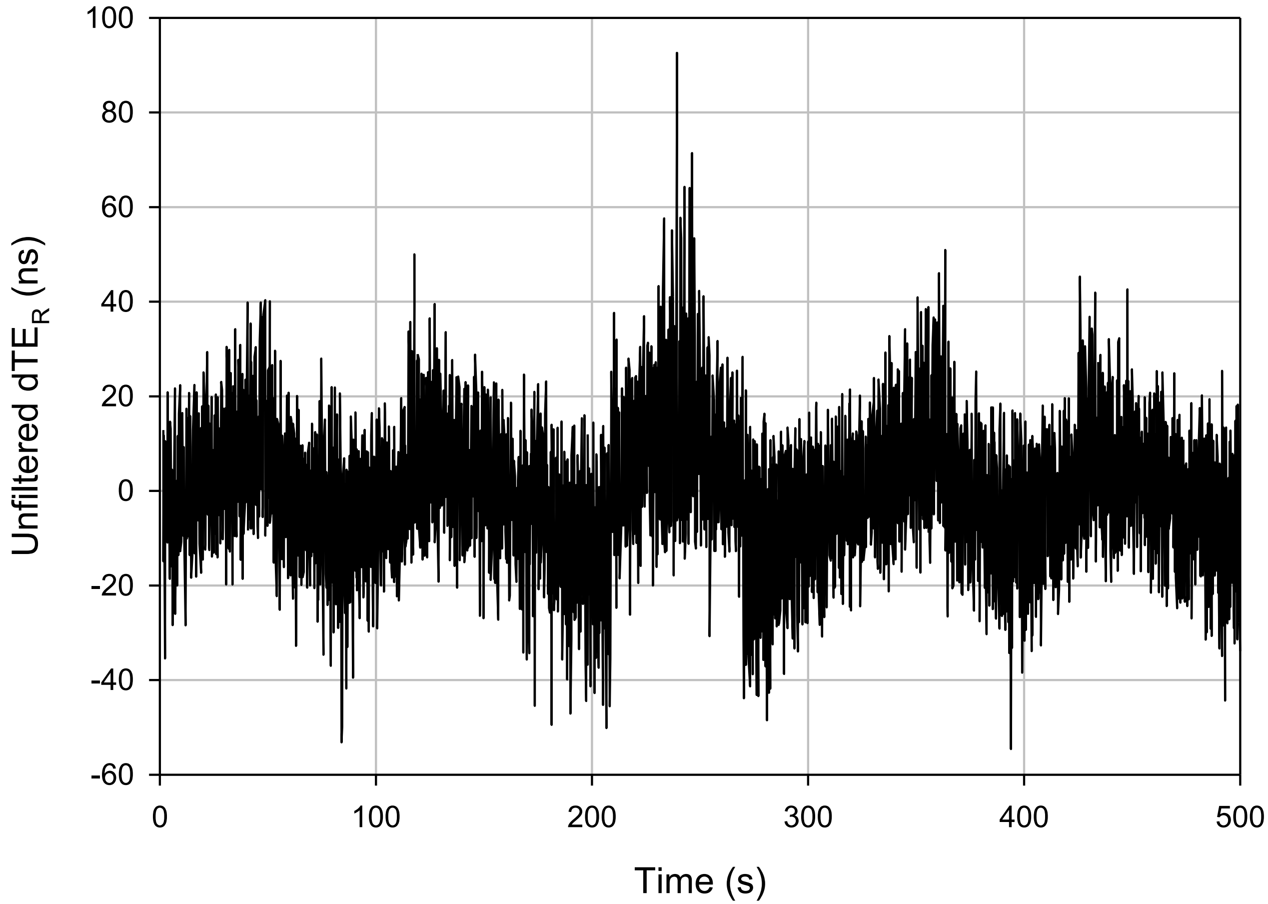


Node 3 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 3, Unfiltered dTE_R

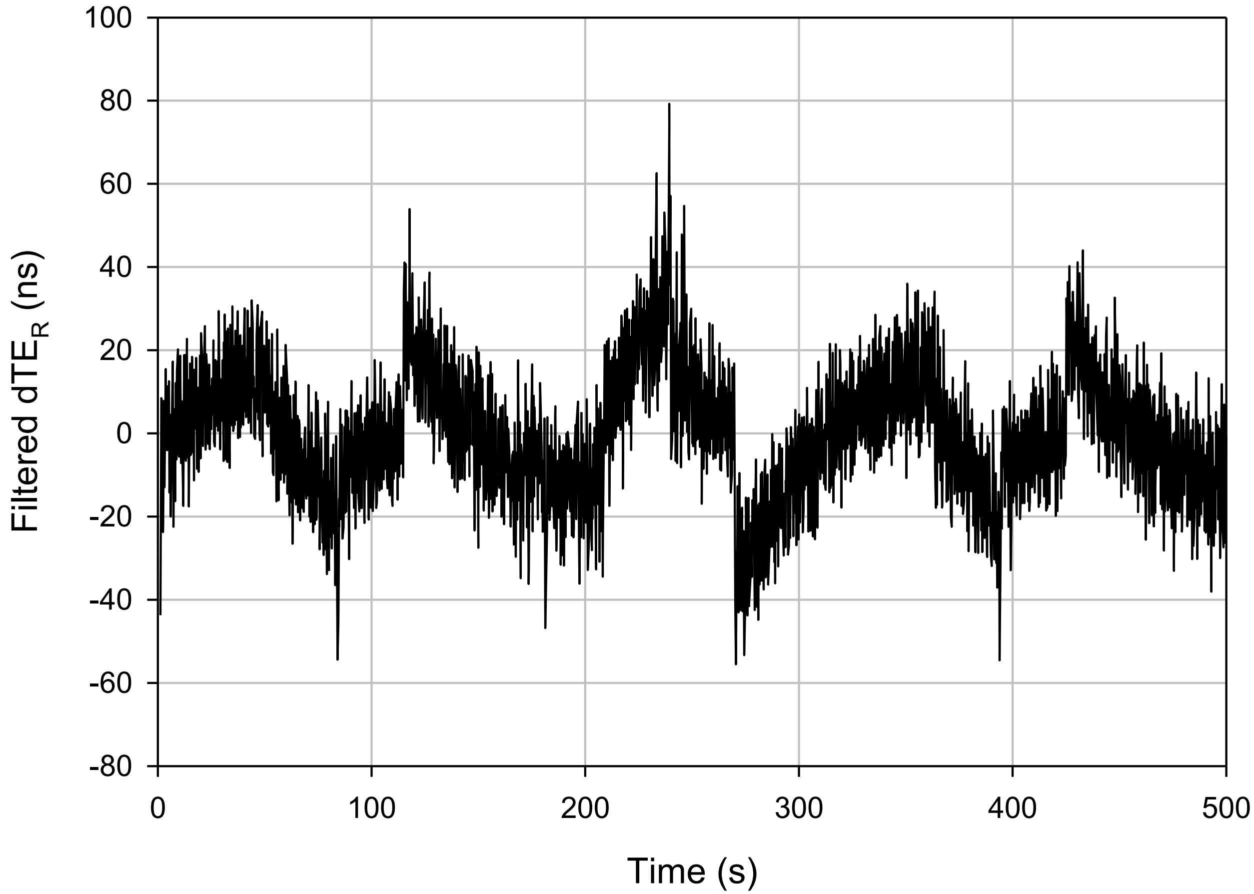
Node 3 dTE_R (relative to GM), before PLL filtering
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 3, Filtered dTE_R

Node 3 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)

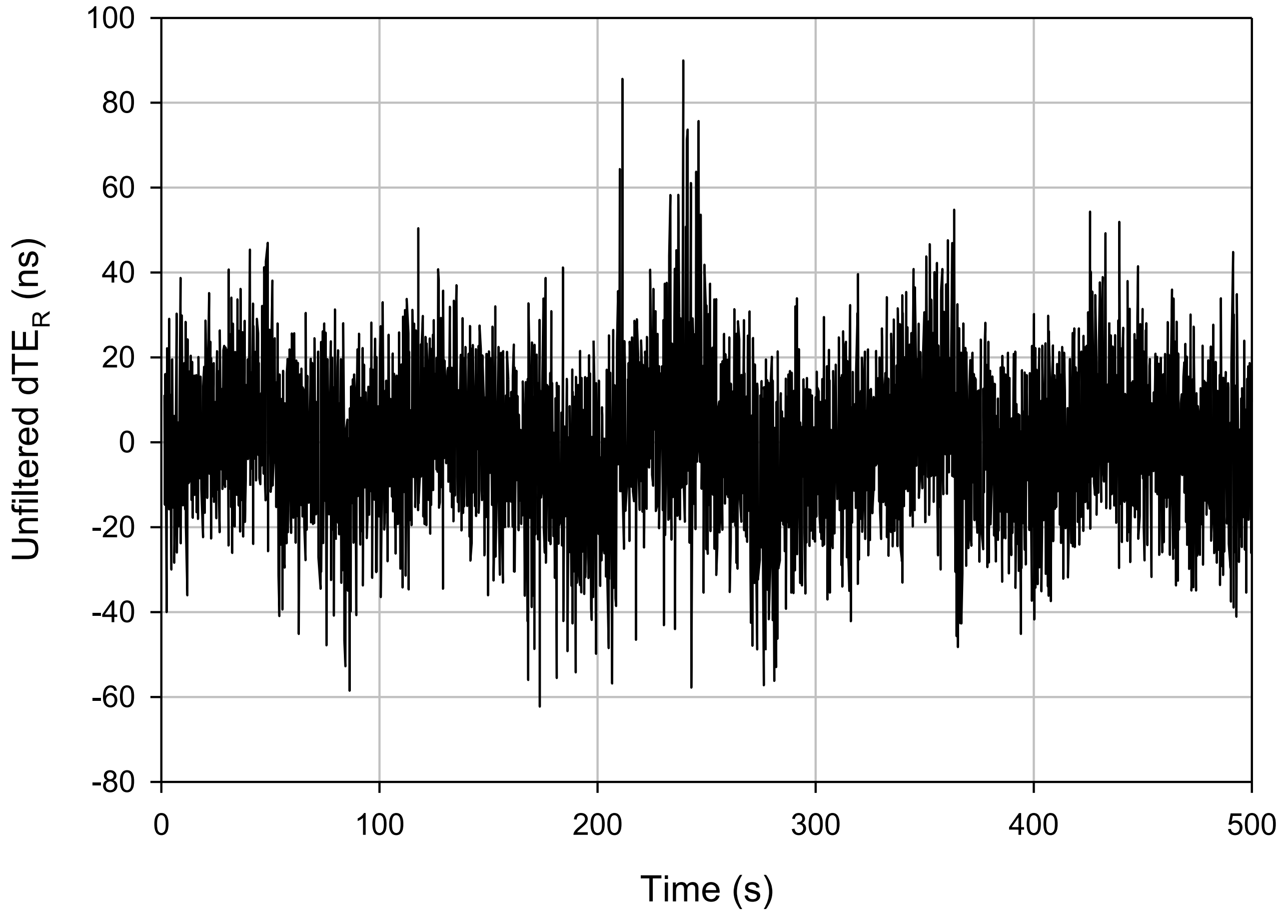


Case I, Node 3

- Unfiltered and filtered dTE_R waveforms are shown on the next two slides
- $\max|dTE_R|$ for the 500 s shown is similar to the corresponding results for case H
- Waveform envelopes are similar to measured FFO (RR) error waveforms for node case H, but not as similar as the case I, node 2 waveform envelopes were to node is measured FFO error for case H, node 2
- Results for case J are very similar to those for case I, and are not shown

Case I, Node 3, Unfiltered dTE_R

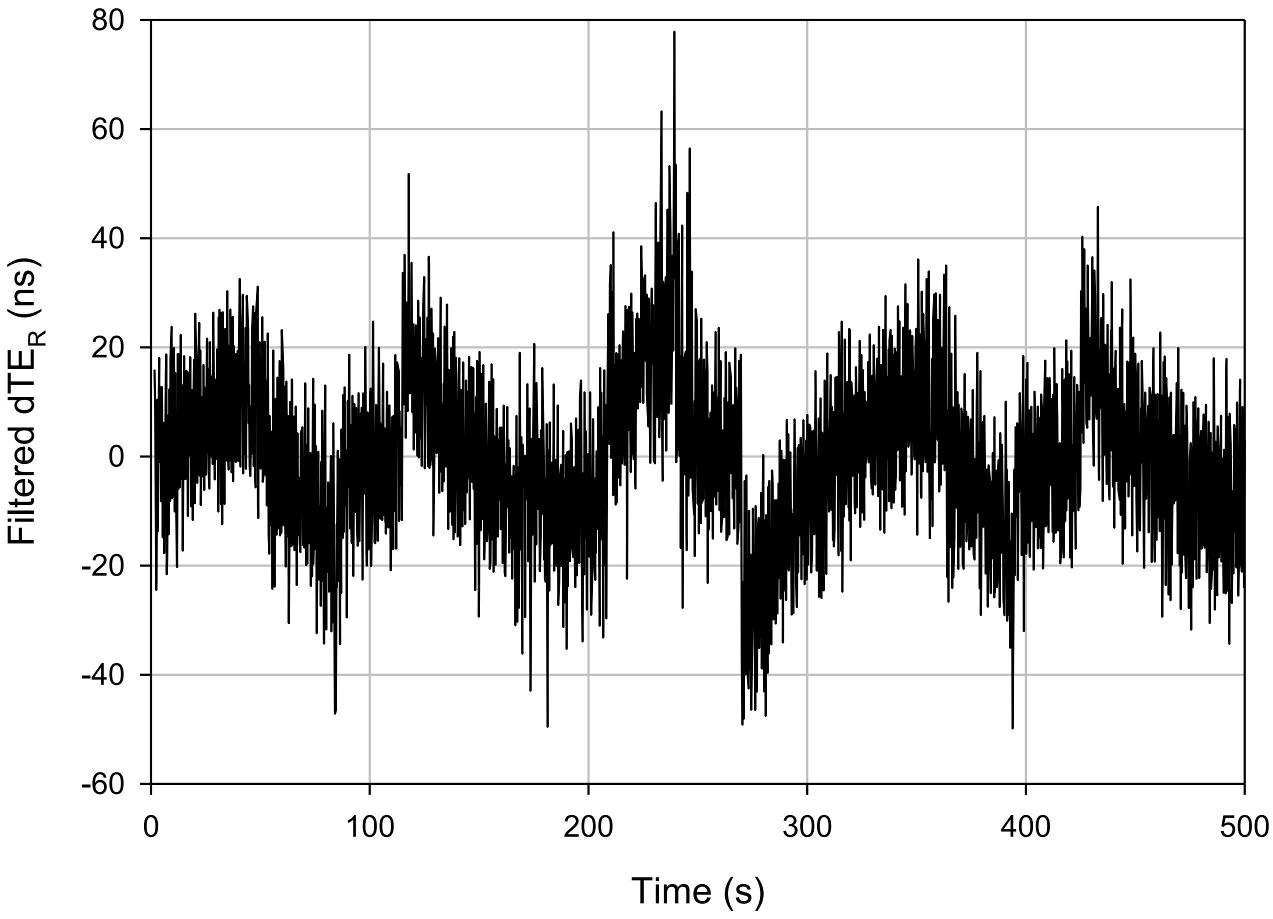
Node 3 dTE_R (relative to GM), before PLL filtering
Case I (correction to nRR+RR; no correction to measured prop delay)



Case I, Node 3, Filtered dTE_R

Node 3 dTE_R (relative to GM), after PLL filtering

Case I (correction to nRR and RR; no correction to measured prop delay)

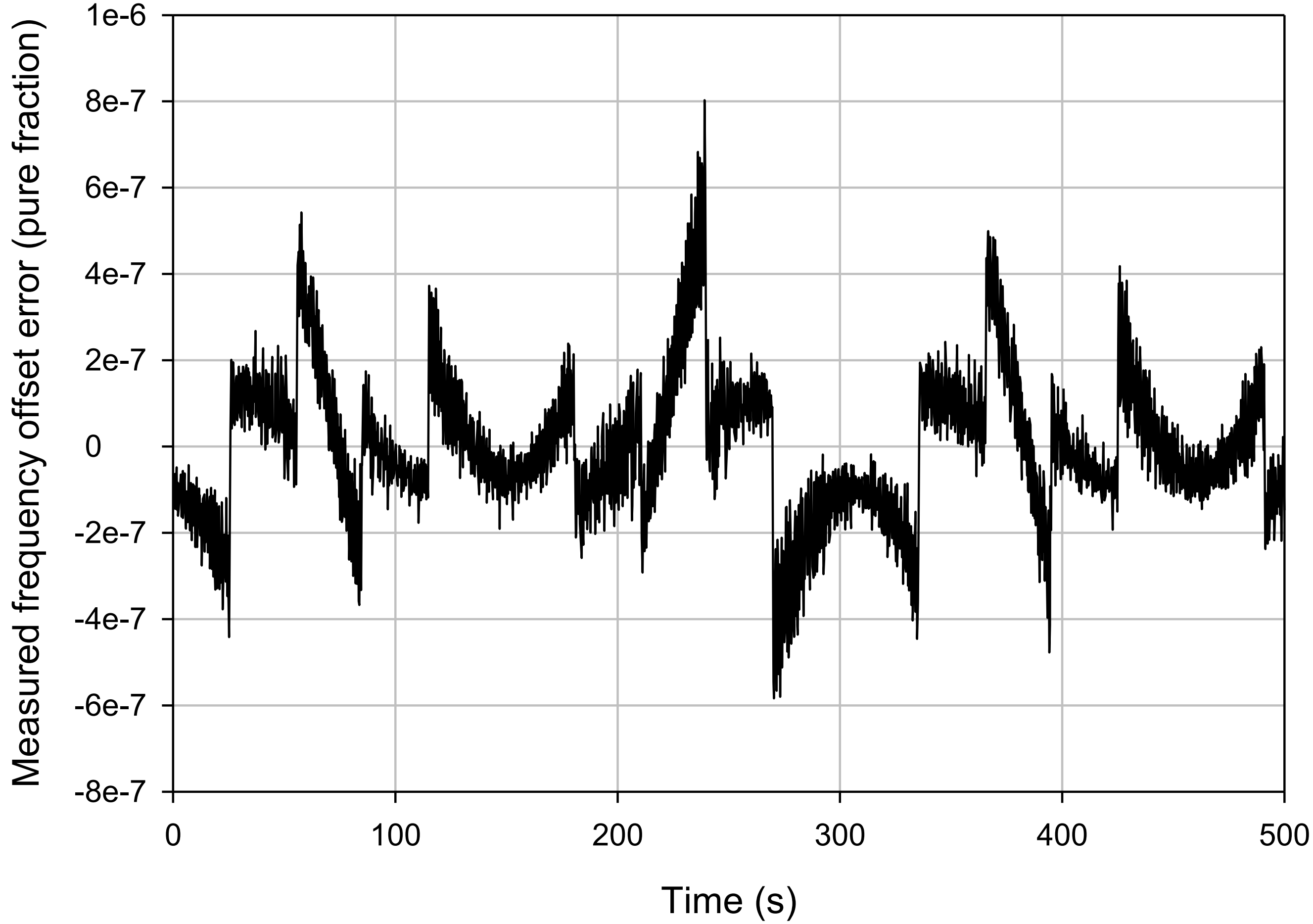


Case G, nodes 4, 5, 10, 50, and 101

- The following slides show error in measured FFO (RR) and filtered dTE_R for Case G, nodes 4, 5, 10, 50, and 101
- Measured RR at a given node now depends on measured NRR at that node and each upstream node
- Jumps in error in measured RR at a node occur at points where the local clock FFO at that node and each upstream node changes slope abruptly
- Jumps are larger when more of the abrupt slope changes line up (i.e., are close) in time
  - For example, node 4 shows many jumps, which occur at times when the local clock FFO at nodes 4, 3, 2, and 1 changes abruptly
  - If abrupt slope changes in a few nodes line up in time, the corresponding jump in error in measured RR will be larger at that time, and smaller jumps at other times where fewer slope changes line up will be less noticeable
    - See, for example, the results for error in measured FFO for nodes 50 and 101 (slides 76 and 78, respectively)
- Error in measured FFO increases with increasing node number
- The filtered dTE_R waveforms have shape similar to the measured FFO error waveforms, and $\max|dTE_R|$ (over the 500 s shown) generally increases with increasing node number (the maximum reaches approximately 1.75 ppm at node 50 and 2.75 ppm at node 101

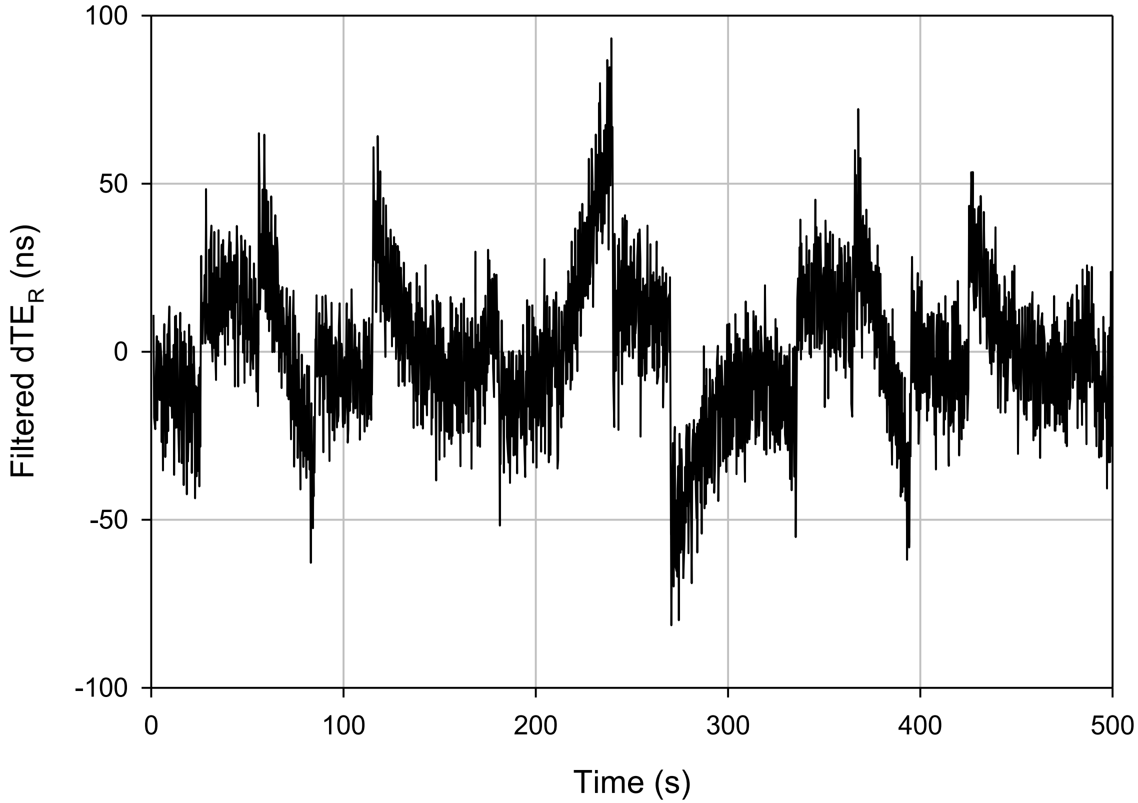
Case G, Node 4, Measured FFO (RR) Error

Node 4 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



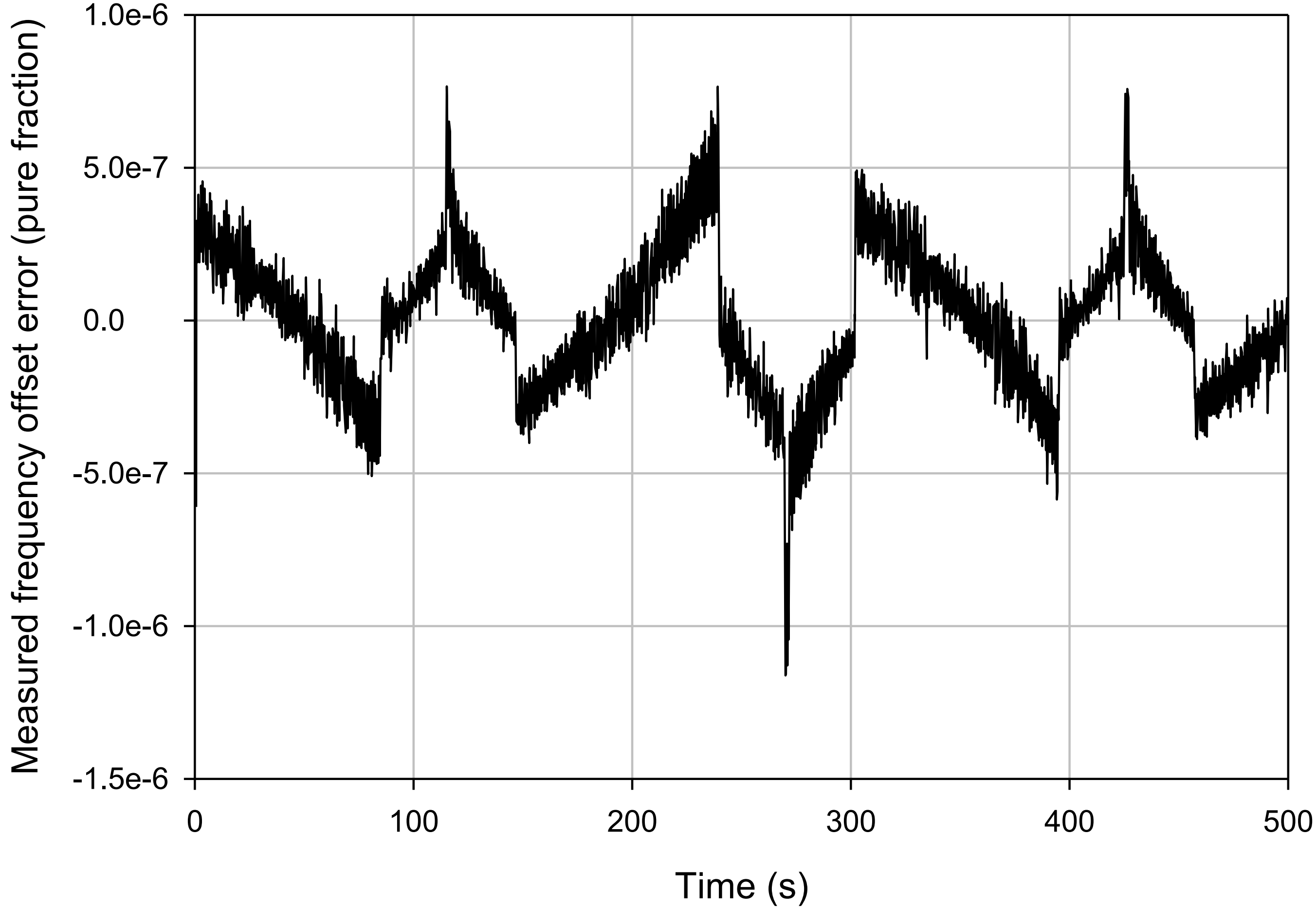
Case G, Node 4, Filtered dTE_R

Node 4 dTE_R (relative to GM), after PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)



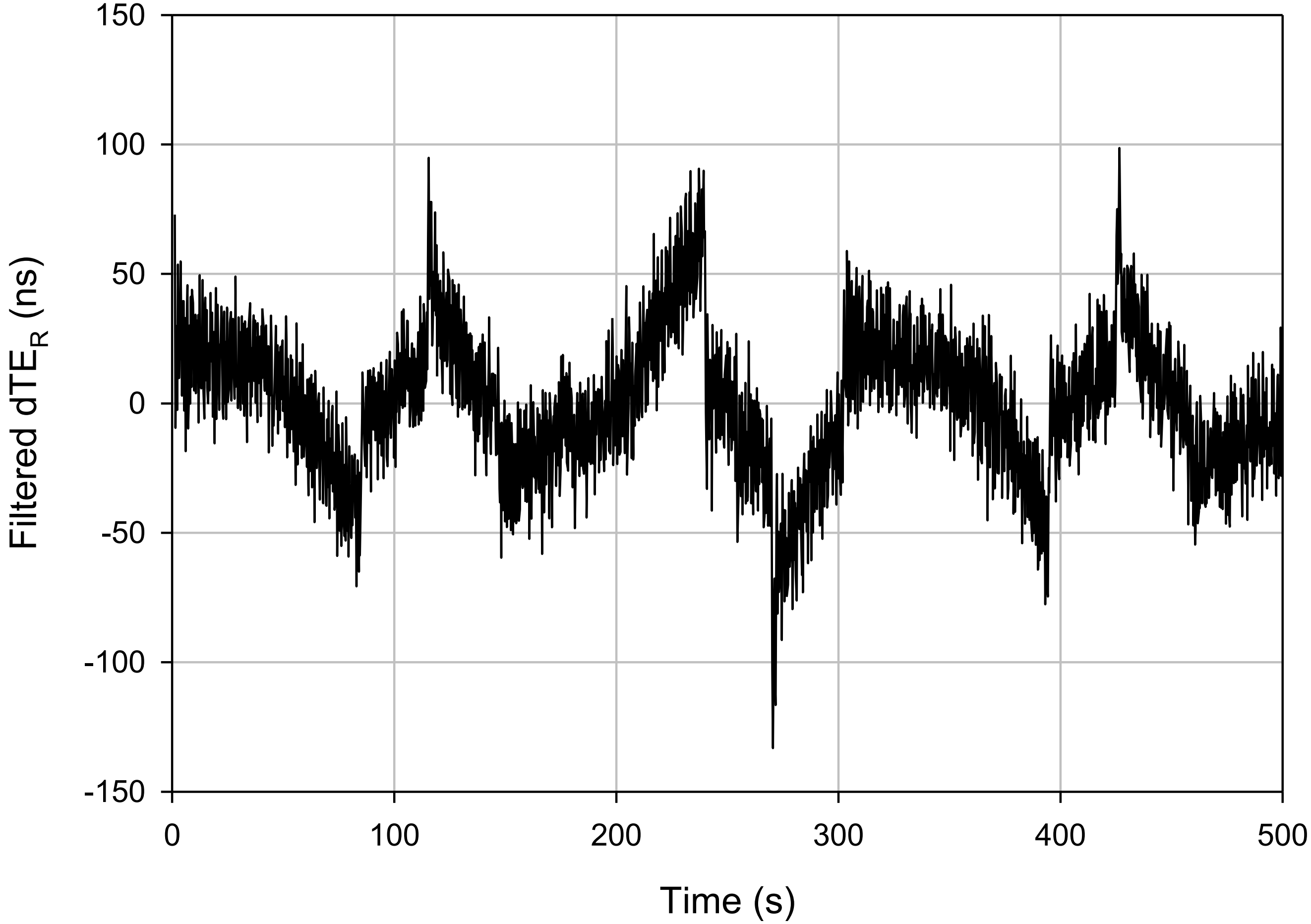
Case G, Node 5, Measured FFO (RR) Error

Node 5 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



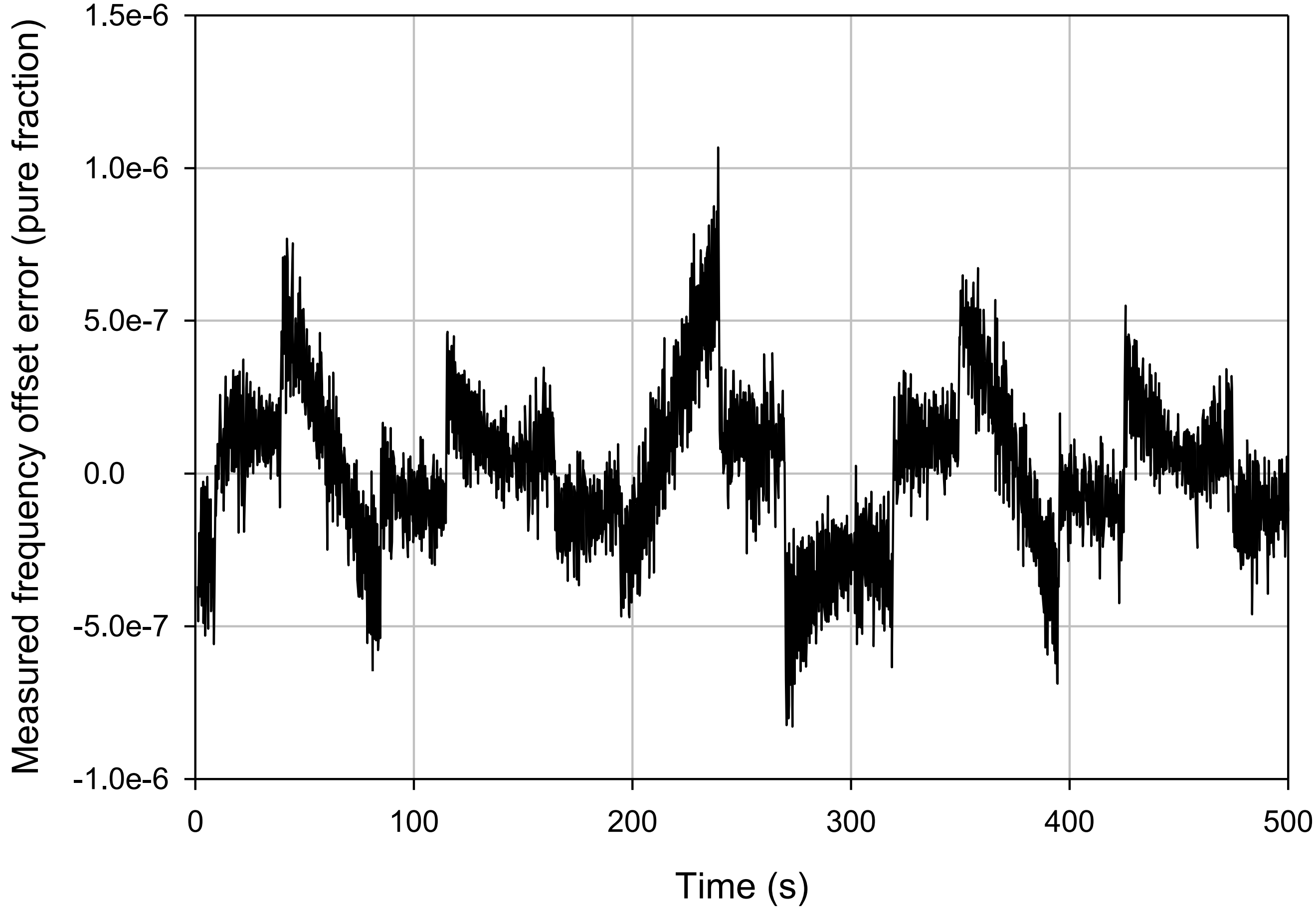
Case G, Node 5, Filtered dTE_R

Node 5 dTE_R (relative to GM), after PLL filtering
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 10, Measured FFO (RR) Error

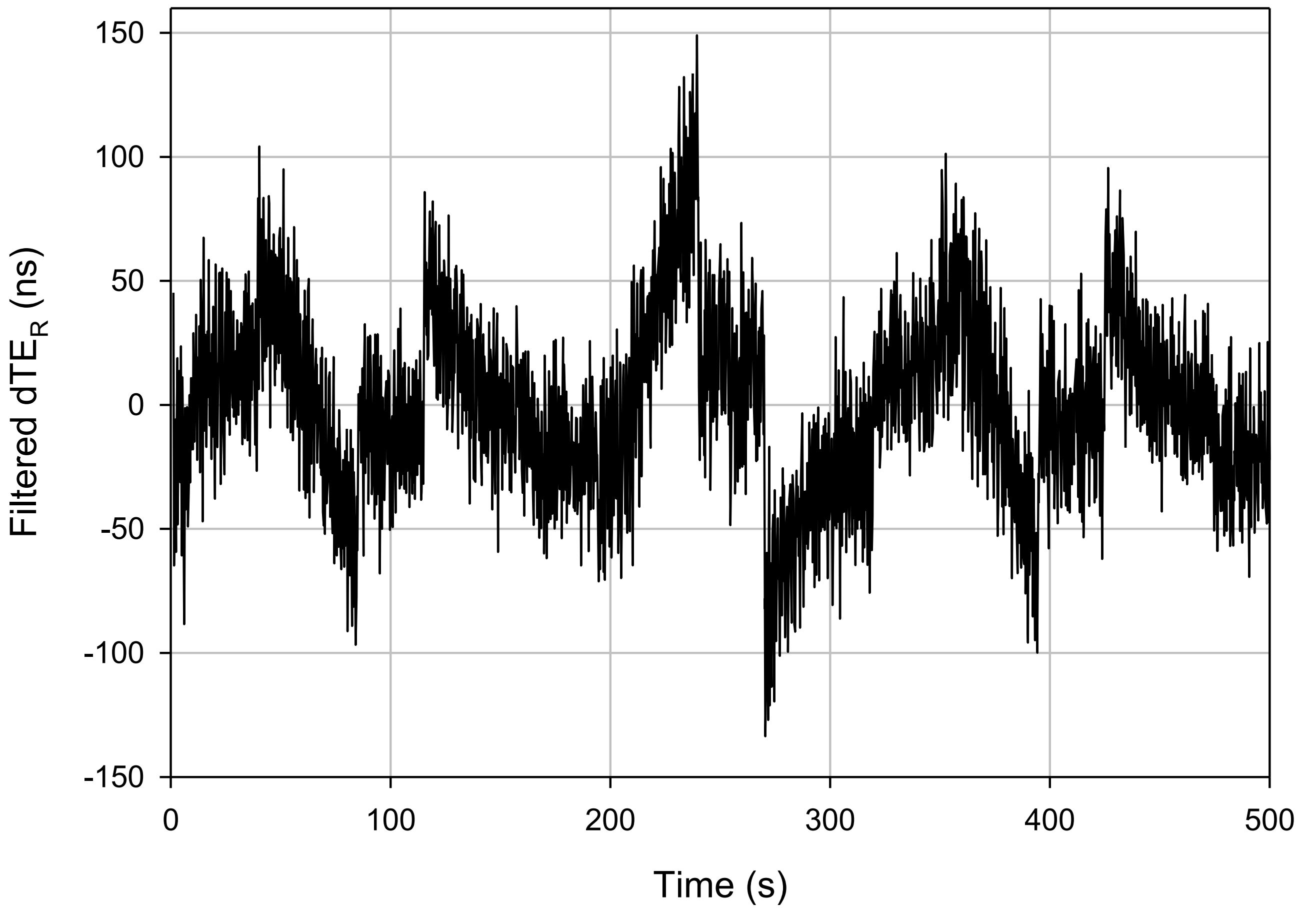
Node 10 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 10, Filtered dTE_R

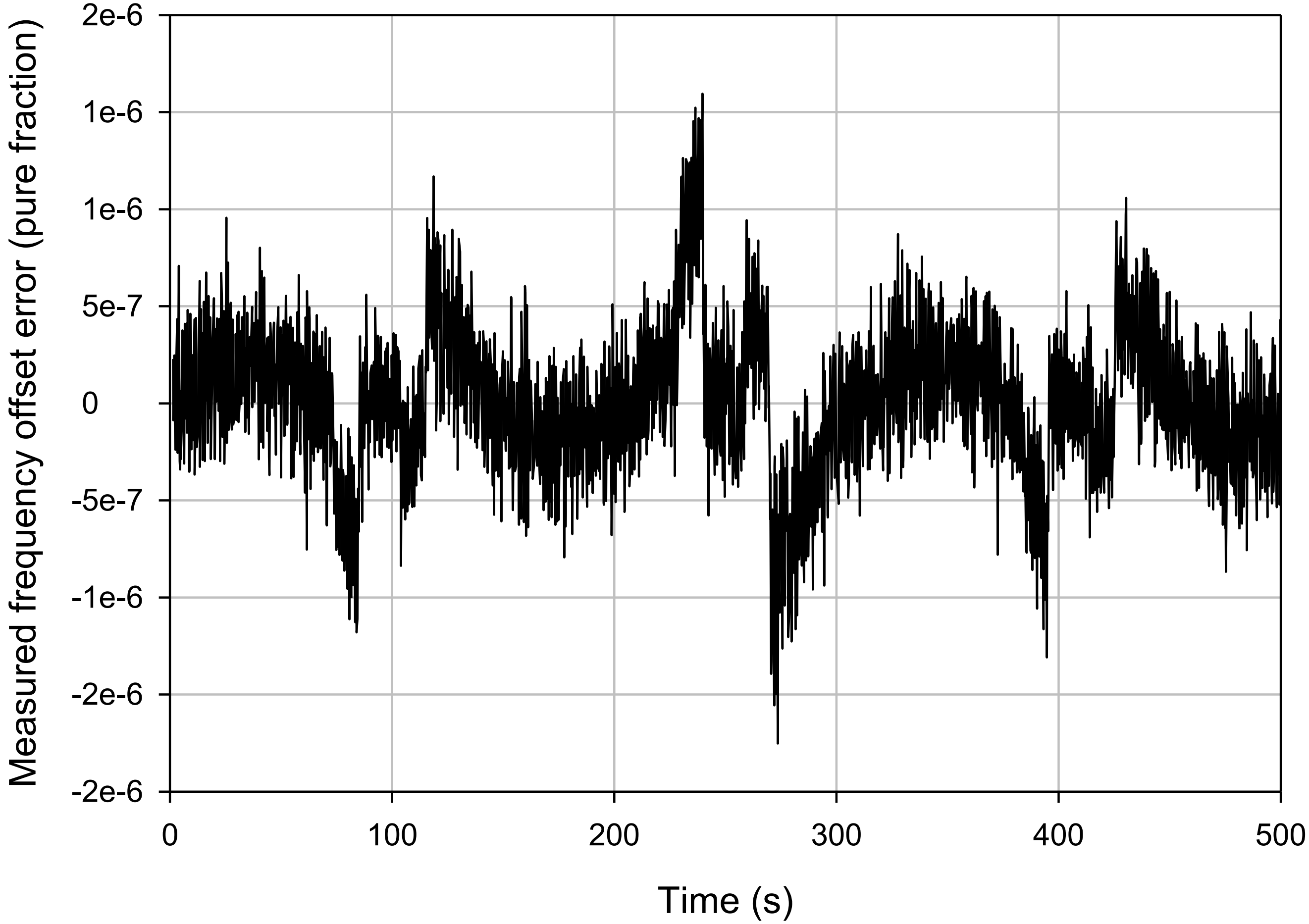
Node 10 dTE_R (relative to GM), after PLL filtering

Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 50, Measured FFO (RR) Error

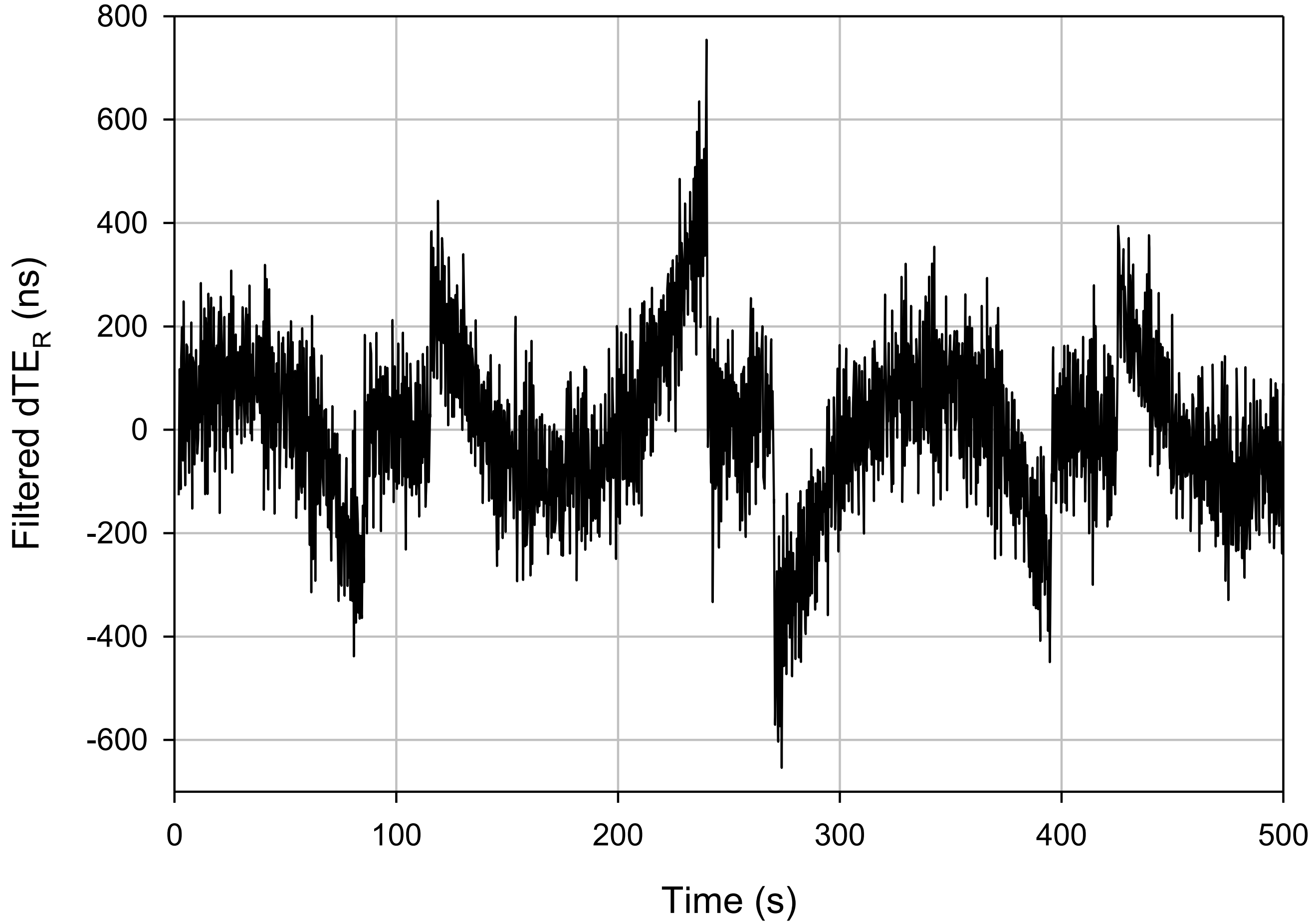
Node 50 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 50, Filtered dTE_R

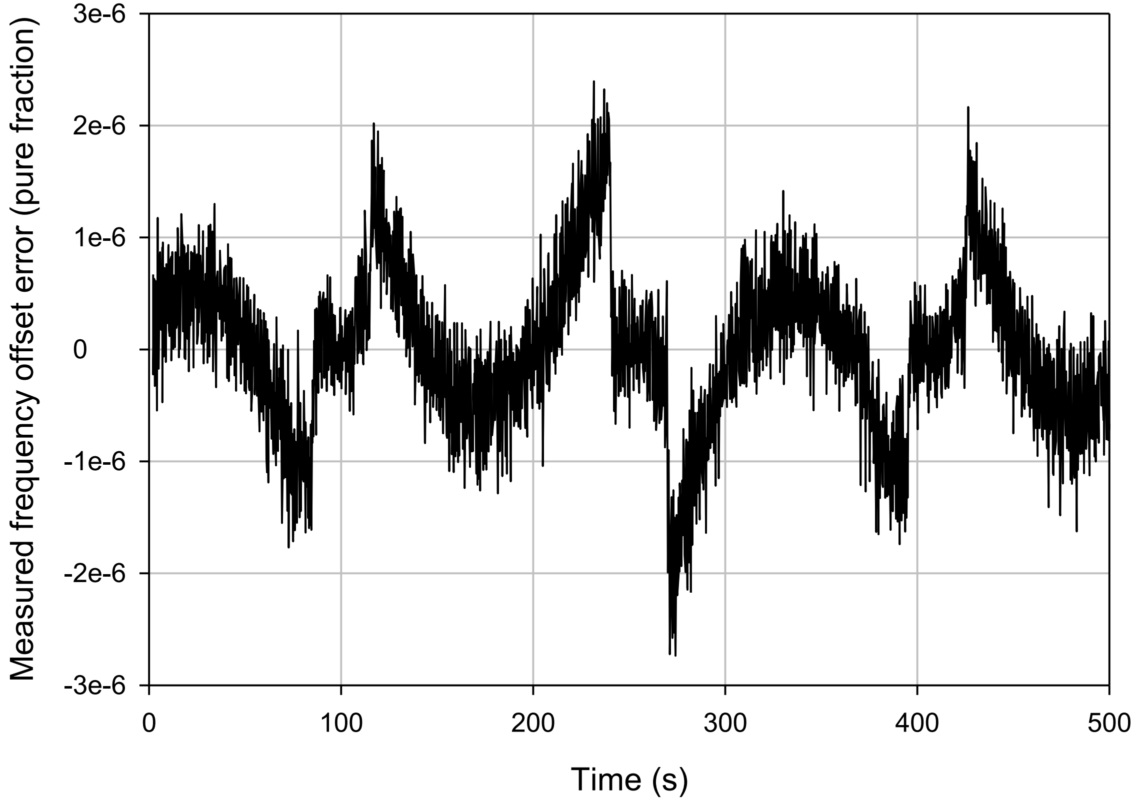
Node 50 dTE_R (relative to GM), after PLL filtering

Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 101, Measured FFO (RR) Error

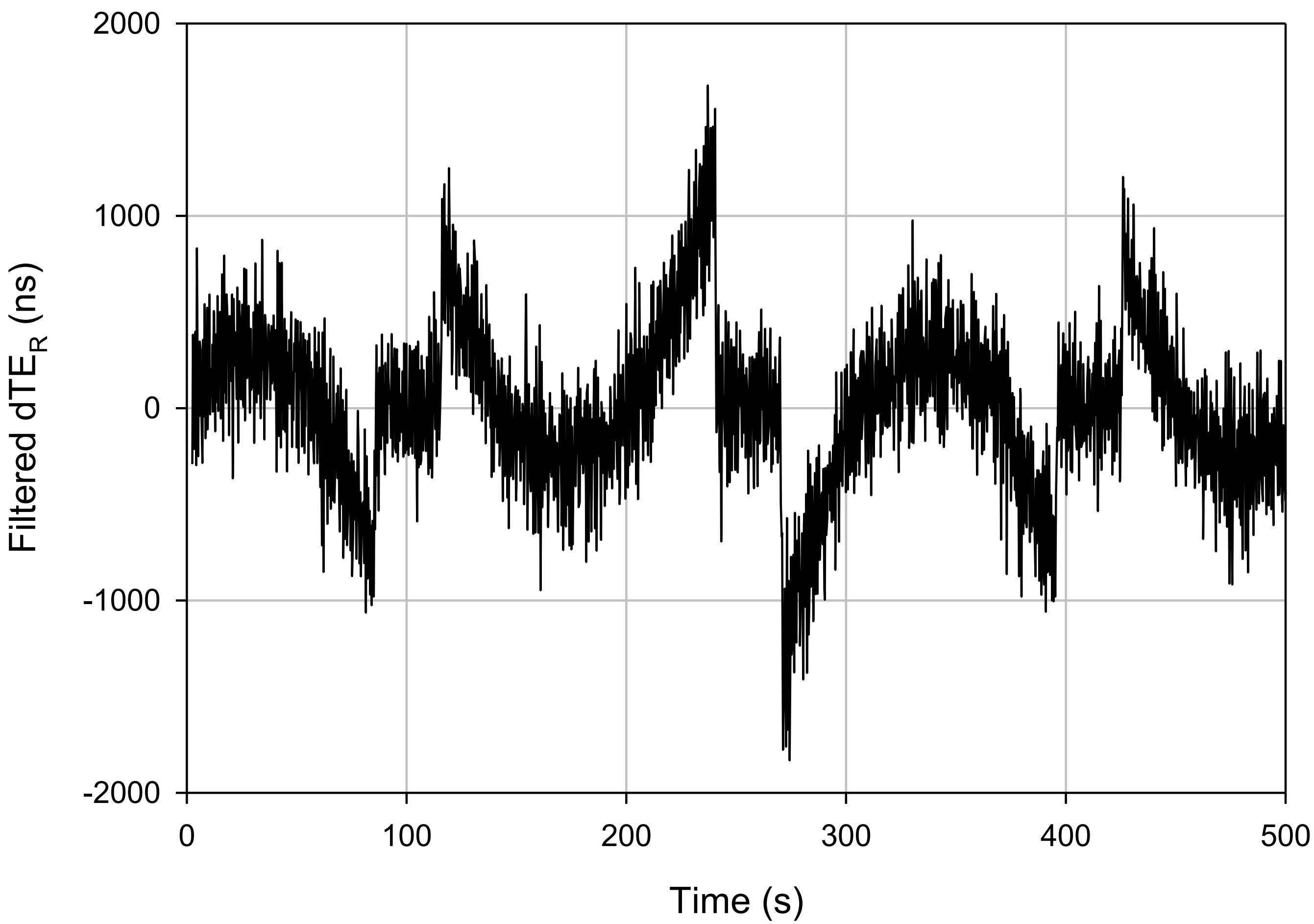
Node 101 error in measured cumulative fractional frequency offset (RR) relative to GM
Case G (no corrections to nRR, RR, or measured prop delay)



Case G, Node 101, Filtered dTE_R

Node 101 dTE_R (relative to GM), after PLL filtering

Case G (no corrections to nRR, RR, or measured prop delay)

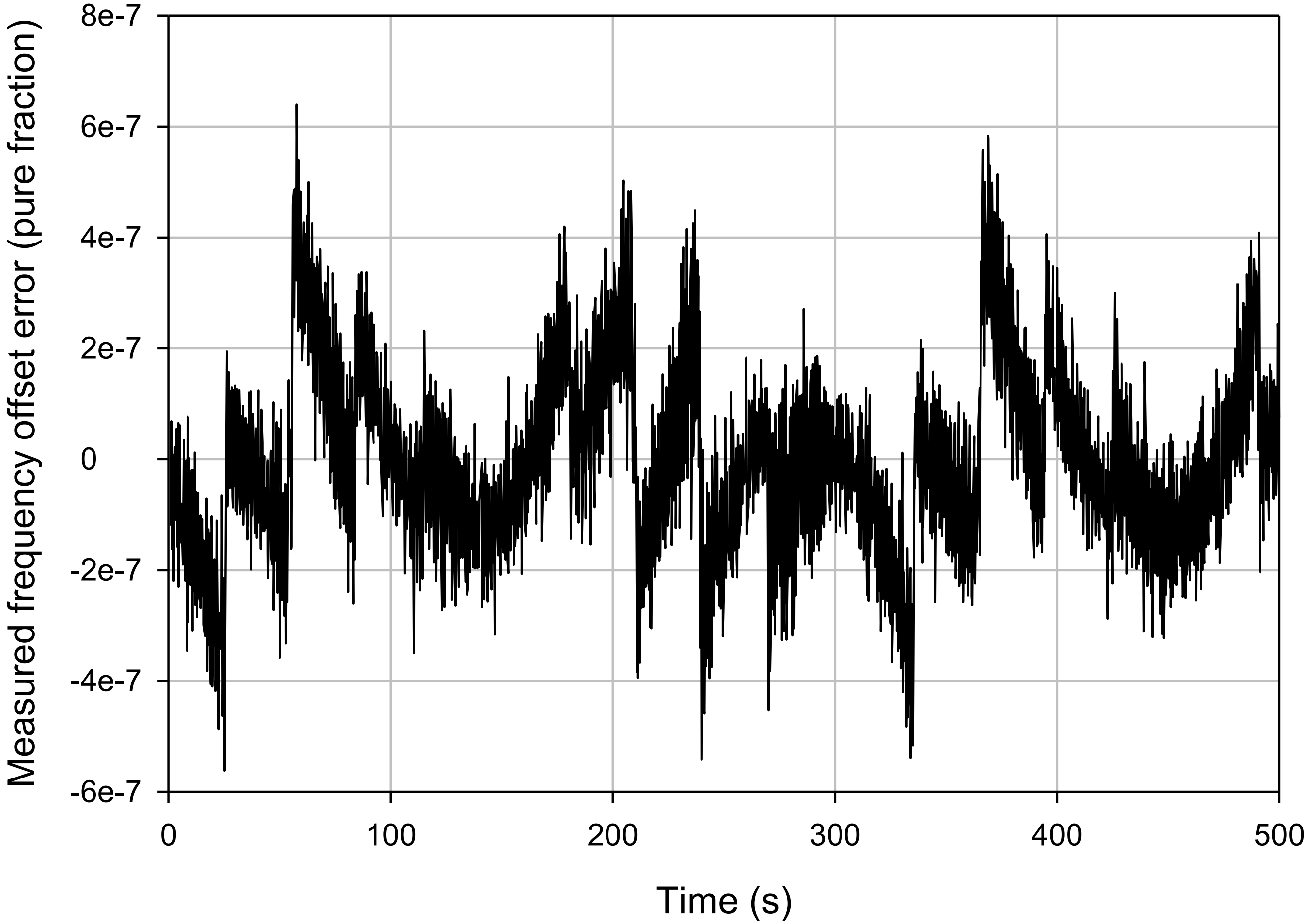


Case H, nodes 4, 5, 10, 50, and 101

- The following slides show error in measured FFO (RR) and filtered dTE_R for Case H, nodes 4, 5, 10, 50, and 101
- As in case G, measured RR at a given node now depends on measured NRR at that node and each upstream node
  - Jumps in error in measured RR at a node occur at points where the local clock FFO at that node and each upstream node changes slope abruptly
- Algorithmic corrections to measured NRR generally result in small improvement in measured RR, with the exception of node 101 where case H error in measured RR is slightly larger than case G error in measured RR
  - Maximum absolute value of error in measured RR is approximately 2.8 ppm for case G and 2.9 ppm for case H (both for node 101)
  - But this difference is small, and could be due to statistical variability
- As in case G, dTE_R waveforms have similar shape as measured FFO error wave forms, and $\max|dTE_R|$ for case H is slightly better than for case G

Case H, Node 4, Measured FFO (RR) Error

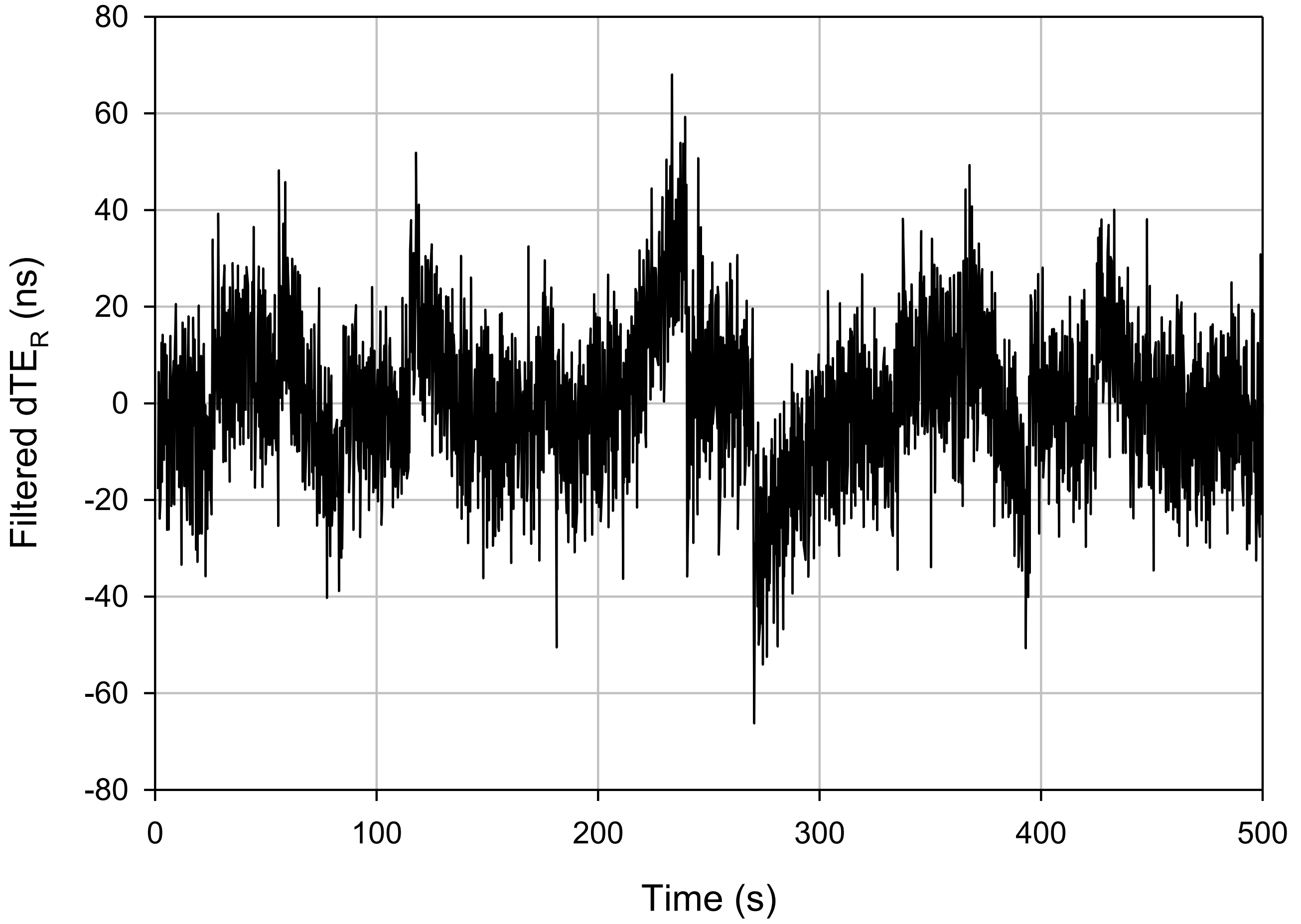
Node 4 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 4, Filtered dTE_R

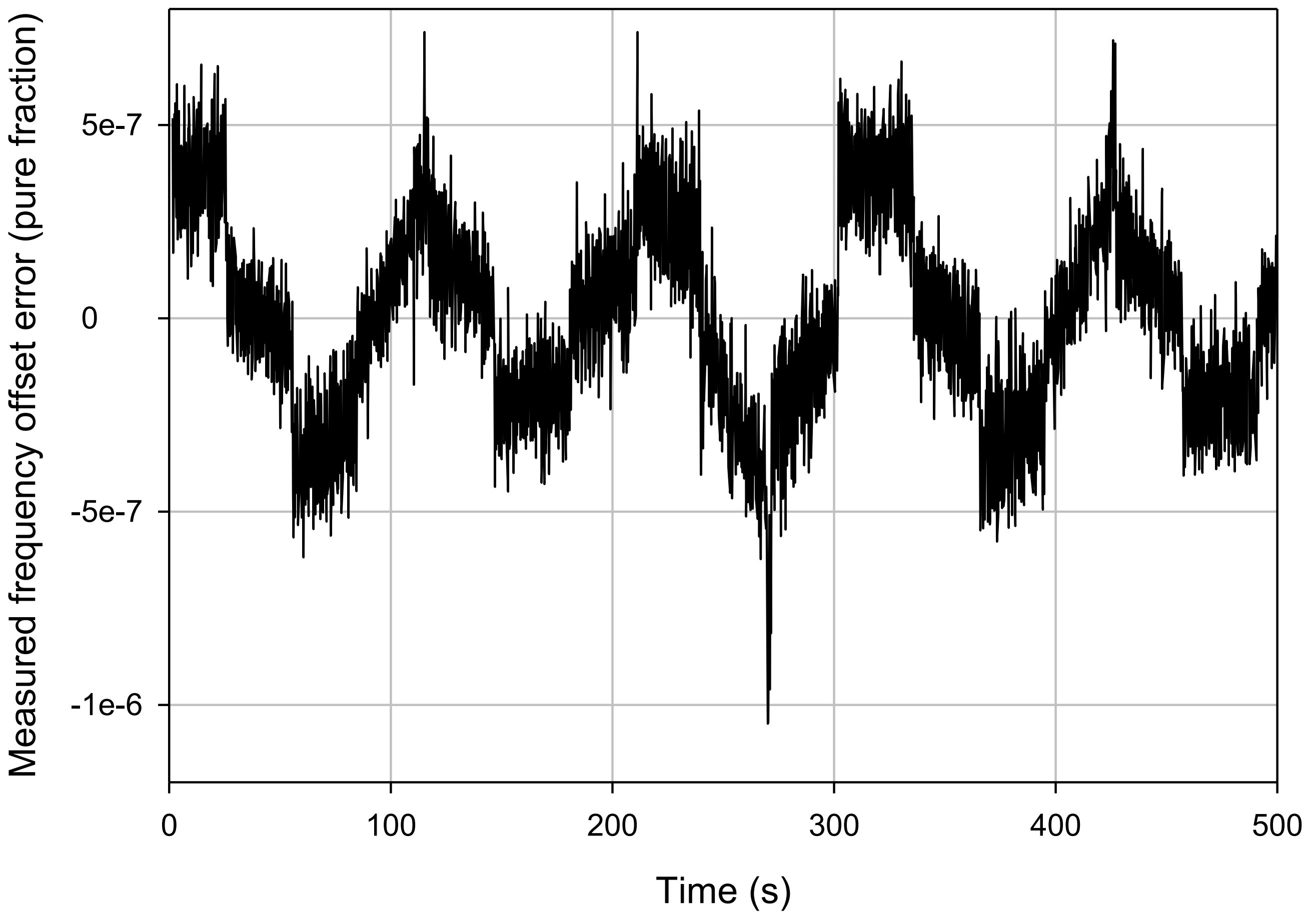
Node 4 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 5, Measured FFO (RR) Error

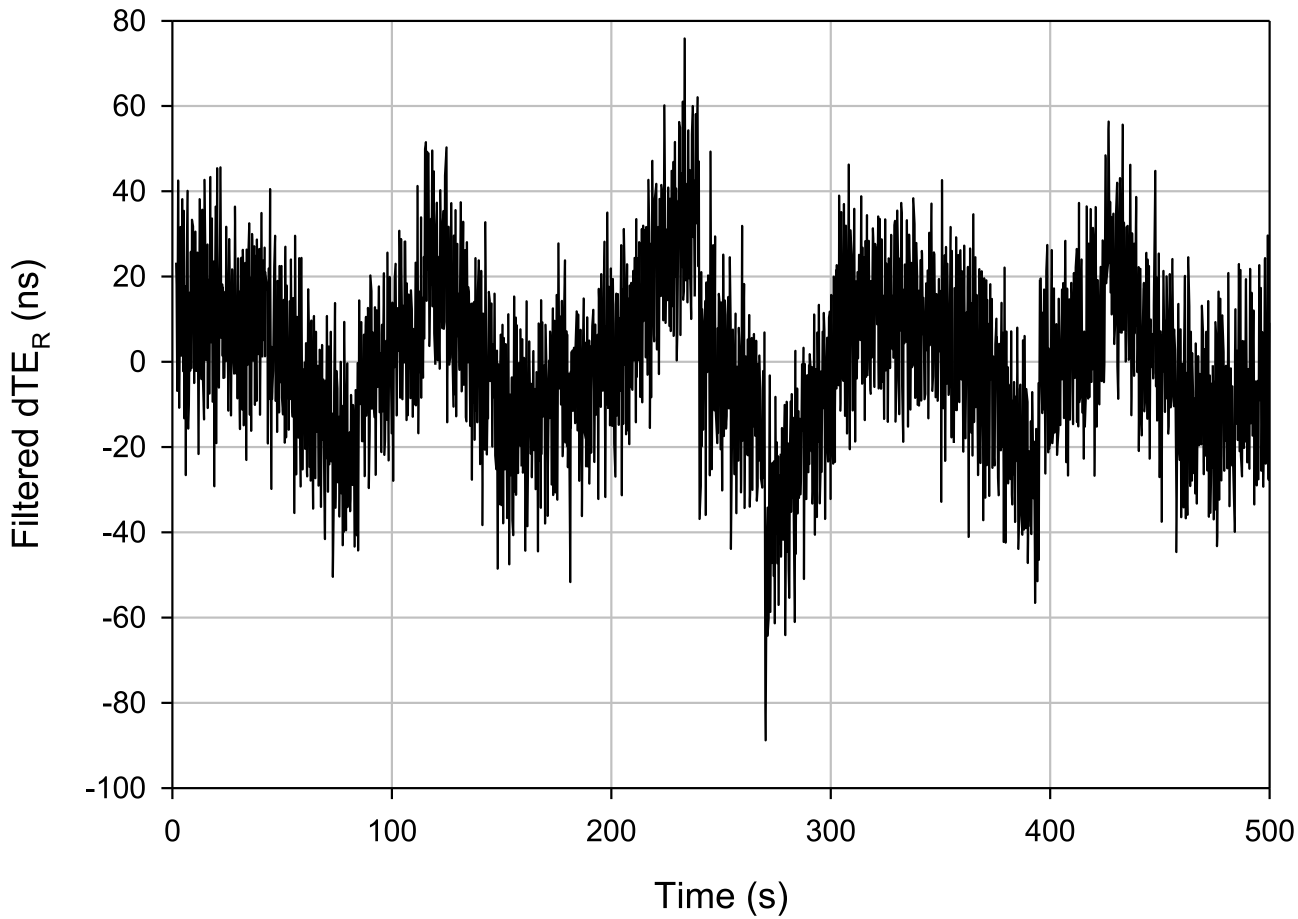
Node 5 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 5, Filtered dTE_R

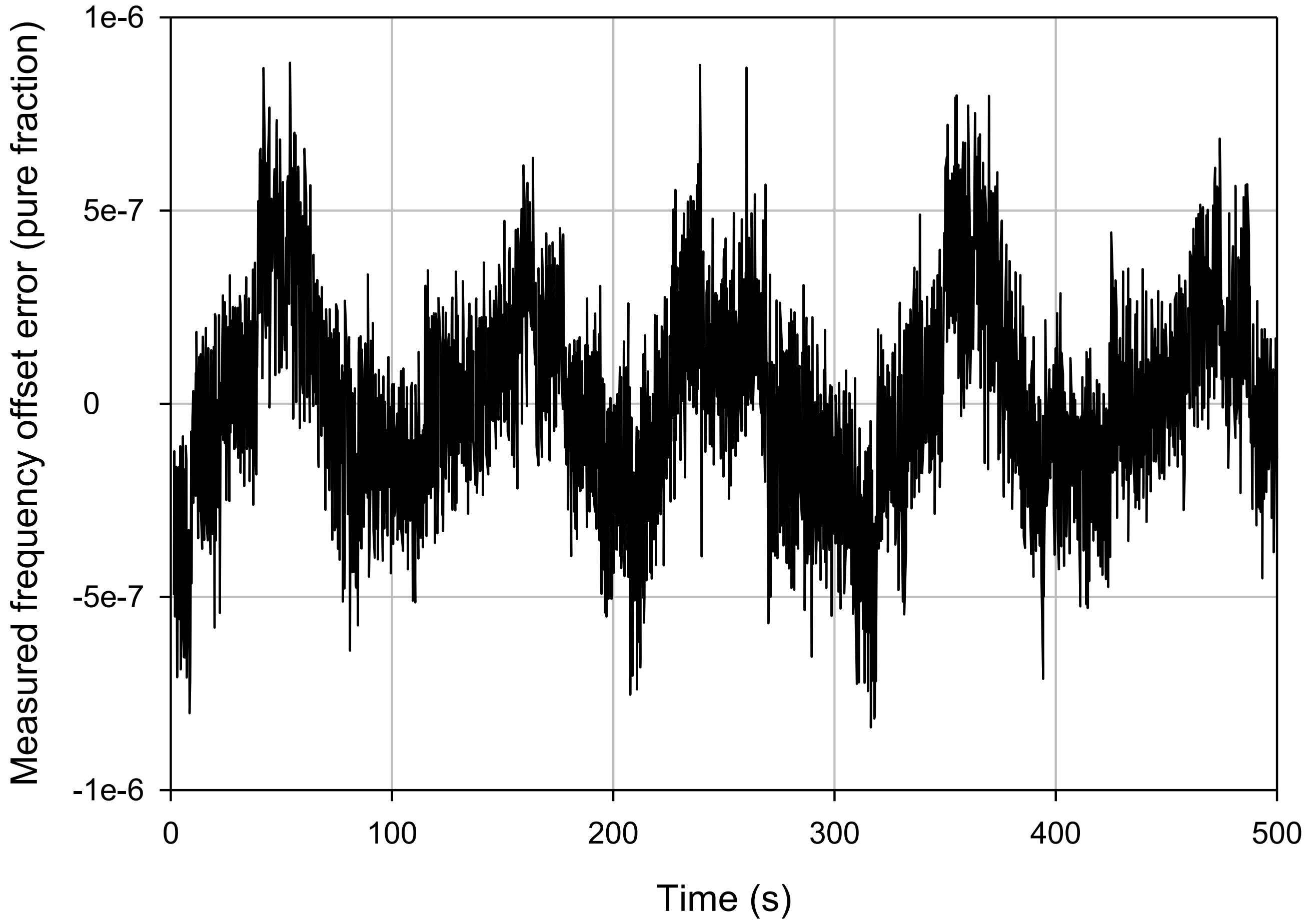
Node 5 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 10, Measured FFO (RR) Error

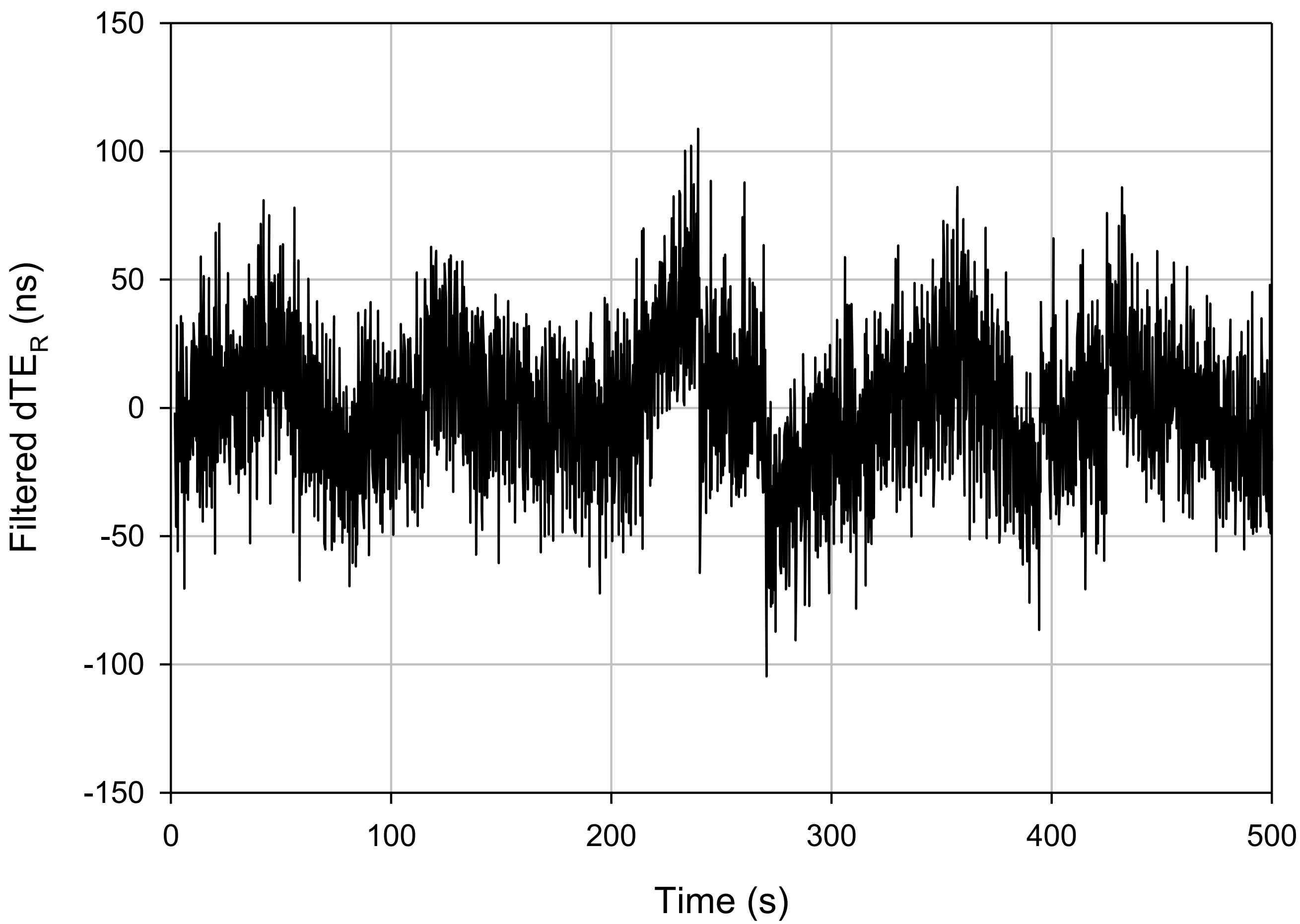
Node 10 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 10, Filtered dTE_R

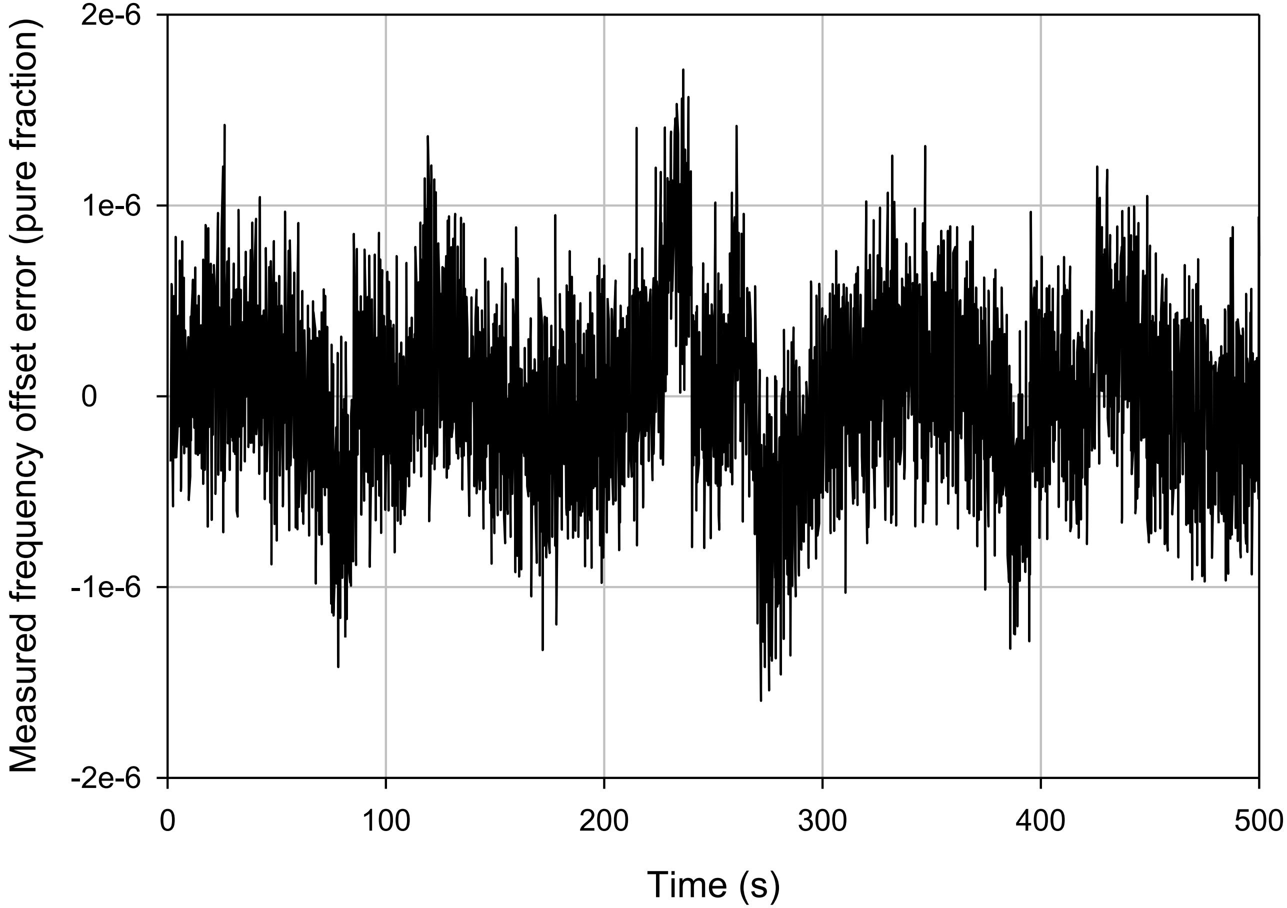
Node 10 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 50, Measured FFO (RR) Error

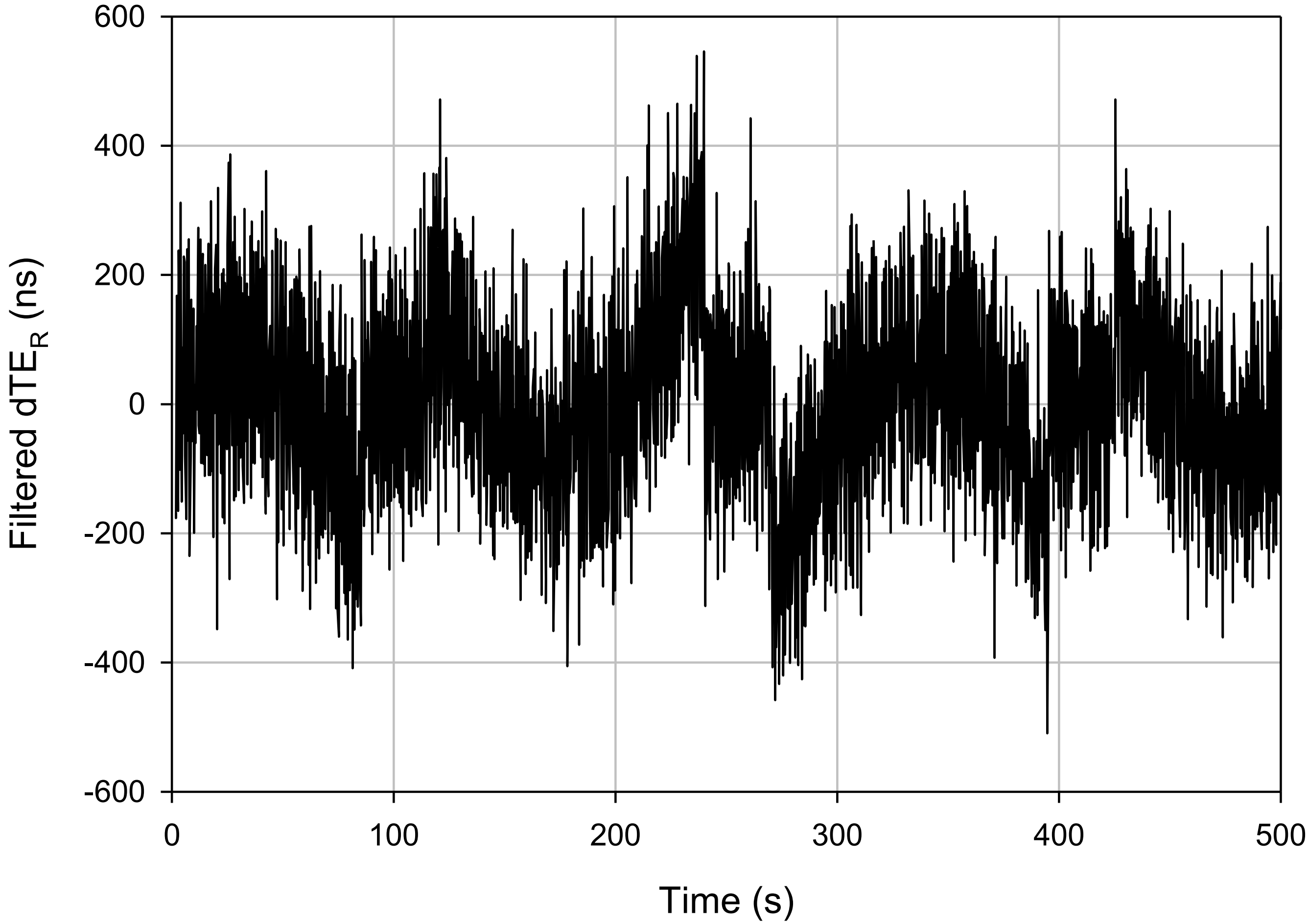
Node 50 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 50, Filtered dTE_R

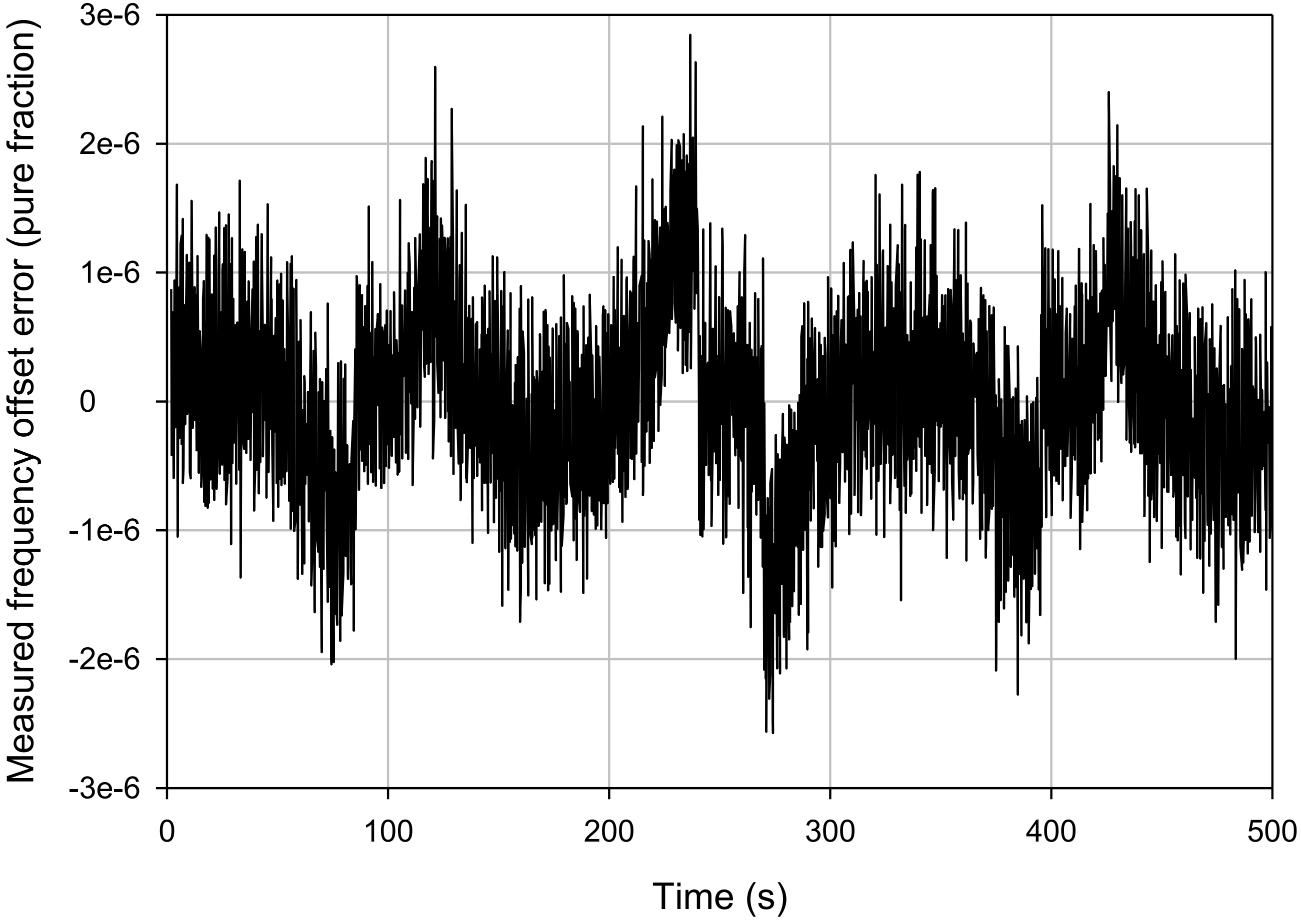
Node 50 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 101, Measured FFO (RR) Error

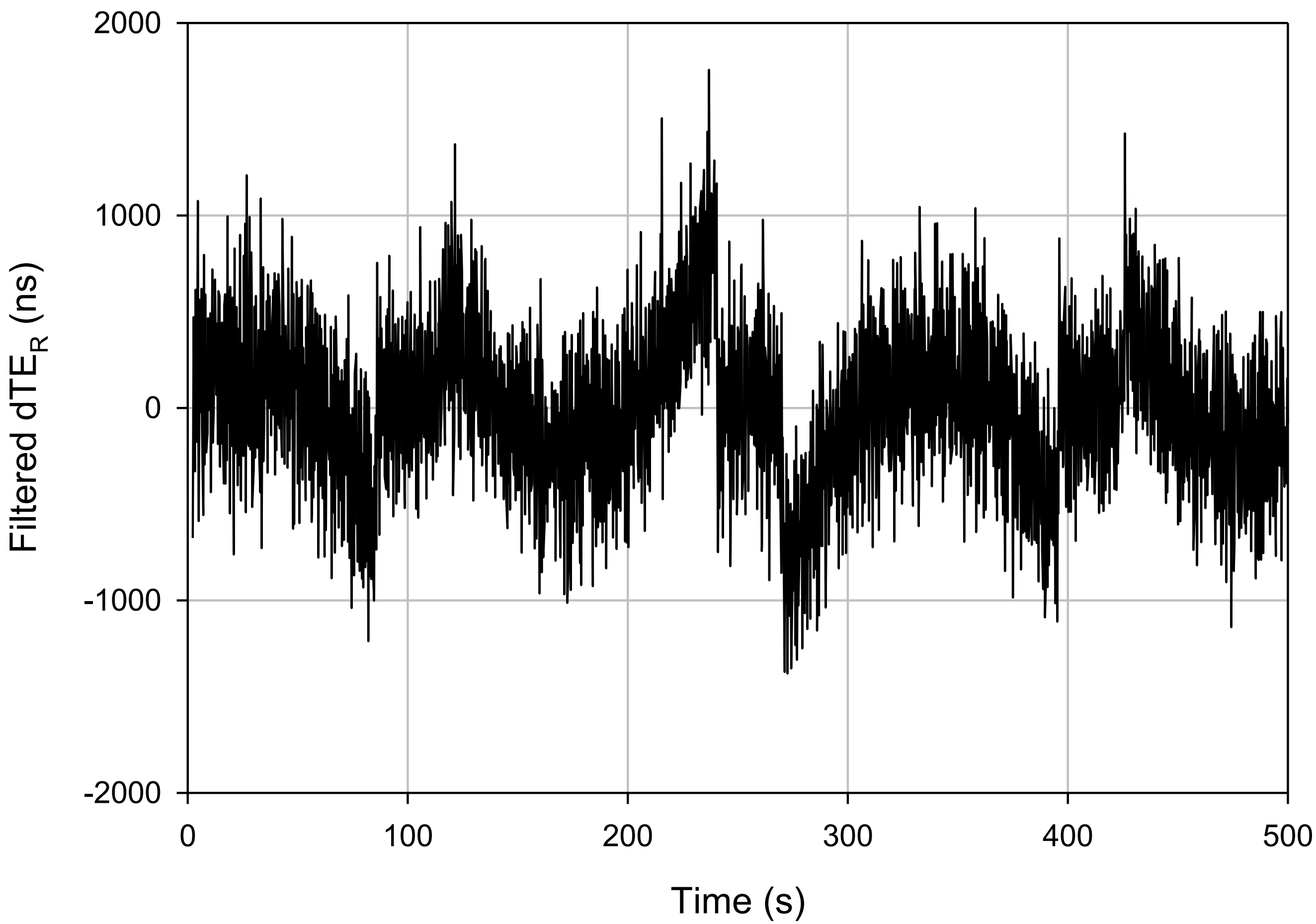
Node 101 error in measured cumulative fractional frequency offset (RR) relative to GM
Case H (correction to nRR; no correction to RR or measured prop delay)



Case H, Node 101, Filtered dTE_R

Node 101 dTE_R (relative to GM), after PLL filtering

Case H (correction to nRR; no correction to RR or measured prop delay)



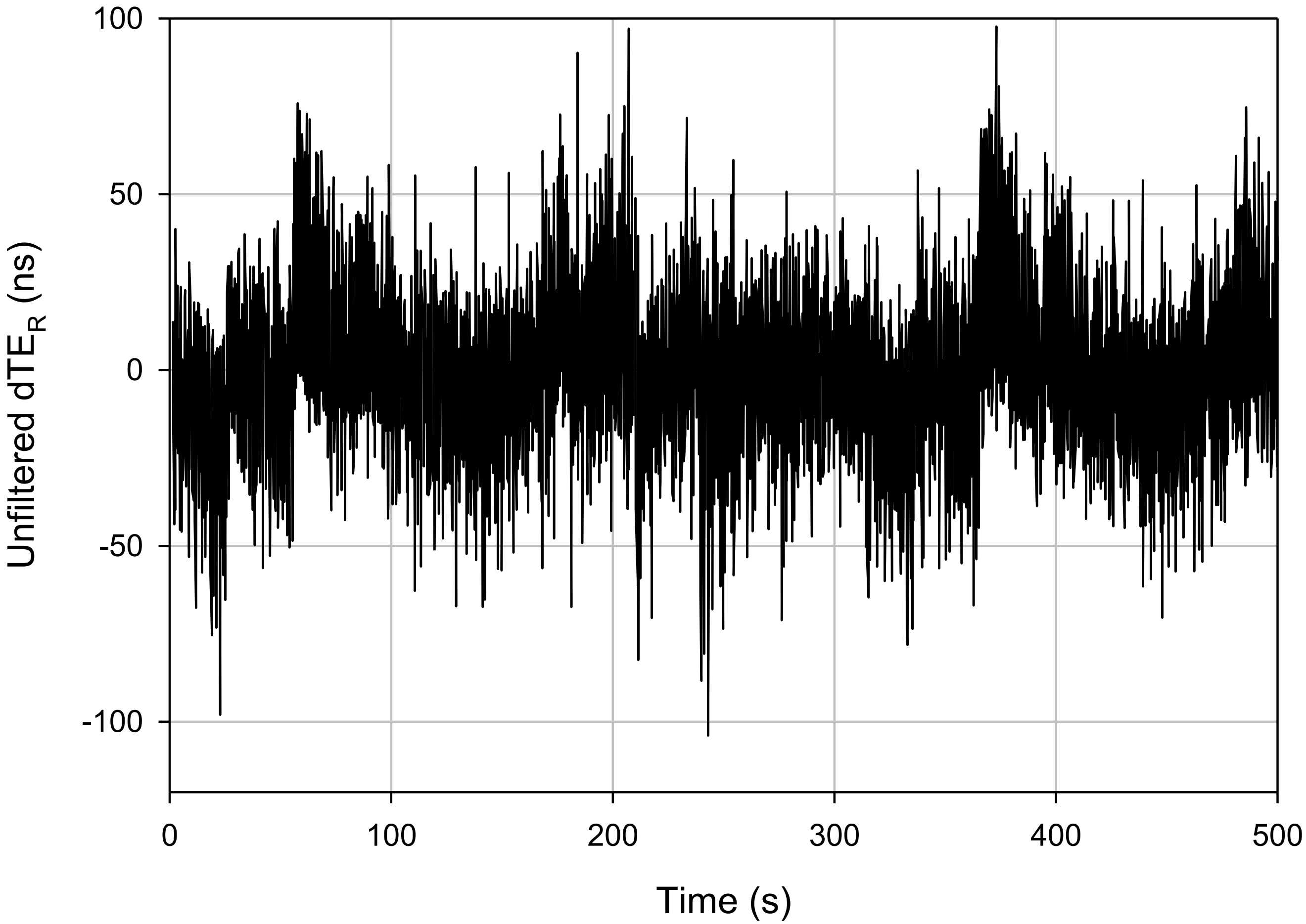
Case I, Nodes 4, 5, 10, 50, and 101

- Unfiltered and filtered dTE_R waveforms are shown on the next two slides
- Waveform envelopes become less similar to measured FFO (RR) error waveforms for node case H as node number increases
  - For nodes 50 and 101, there is no remaining similarity to Measured FFO error waveforms for corresponding case H cases
- There is a relatively large increase in $\max|dTE_R|$ with increasing node number
  - Filtered $\max|dTE_R|$ is approximately 1900 ns for node 50 and 7500 ns for node 101
    - Note that these values are below those in the case I plot for the case of non-zero GM error on slide 25 because slide 25 values are computed over the 3150 s run time with the first 50 s omitted to remove any startup transient, while the plots here are for the first 500 s with the first 3 – 5 s omitted (the data indicated that startup transients actually decayed after 3 – 5 s)
- Results for case J are very similar to those for case I, and are not shown

Case I, Node 4, Unfiltered dTE_R

Node 4 dTE_R (relative to GM), before PLL filtering

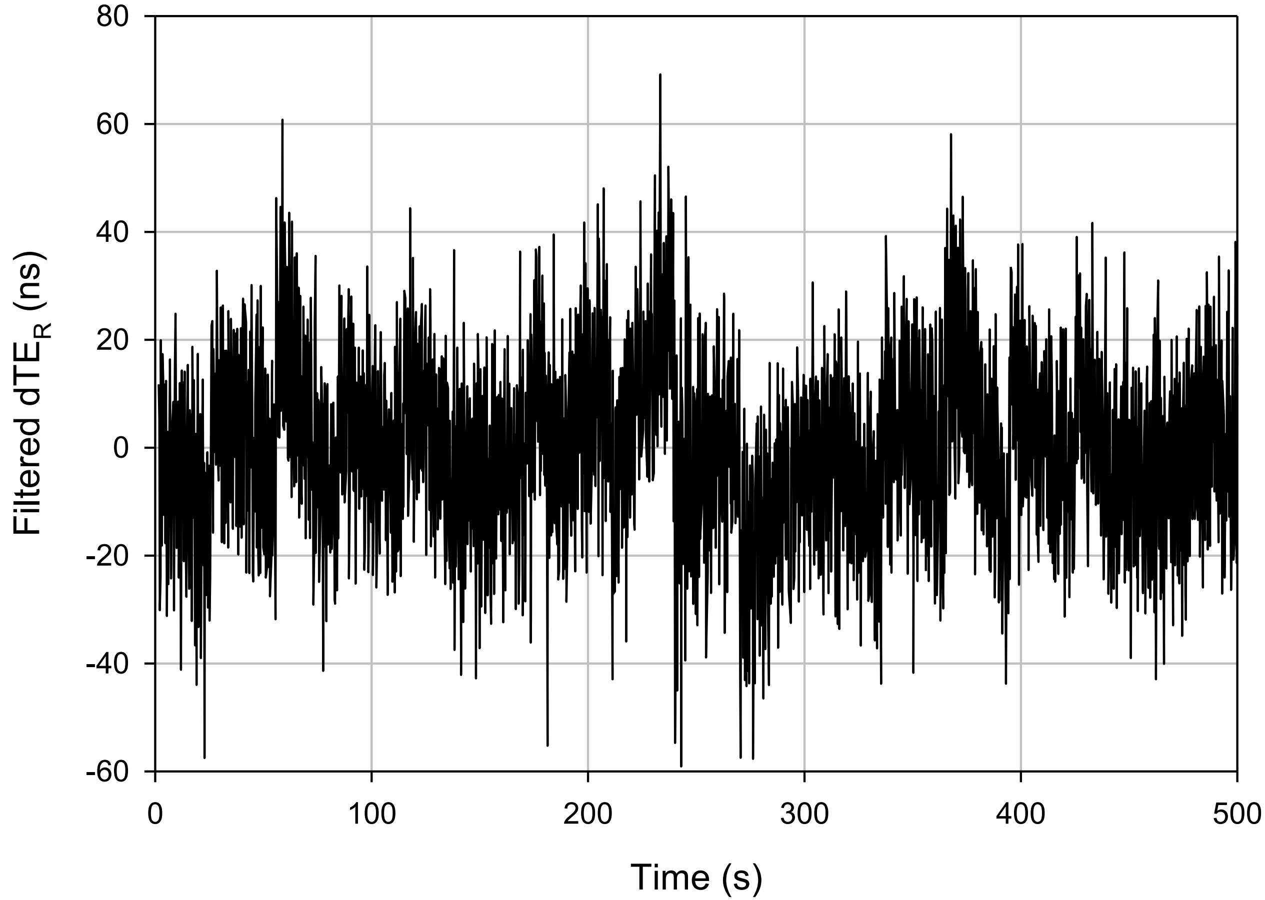
Case I (correction to nRR+RR; no correction to measured prop delay)



Case I, Node 4, Filtered dTE_R

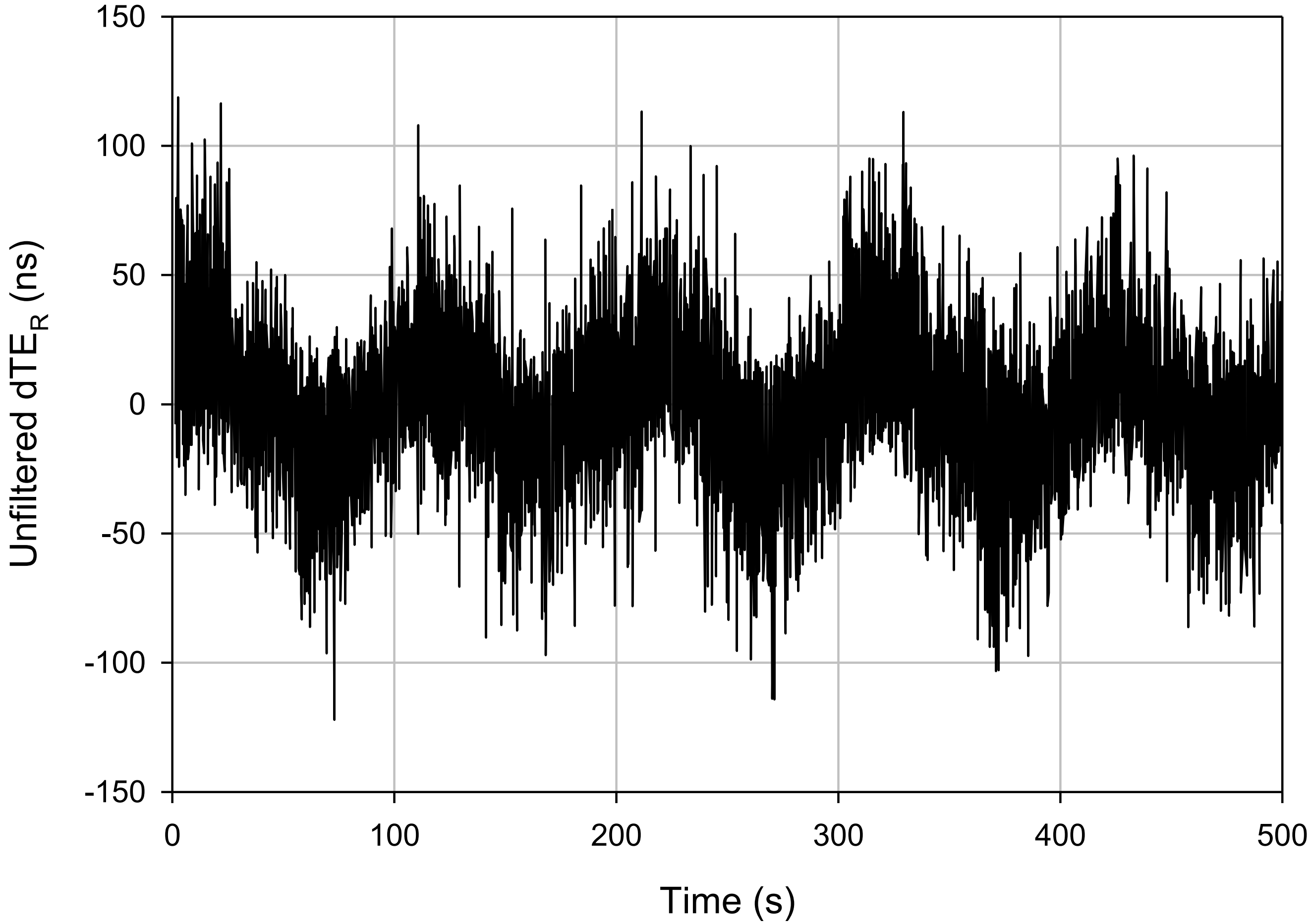
Node 4 dTE_R (relative to GM), after PLL filtering

Case I (correction to nRR and RR; no correction to measured prop delay)



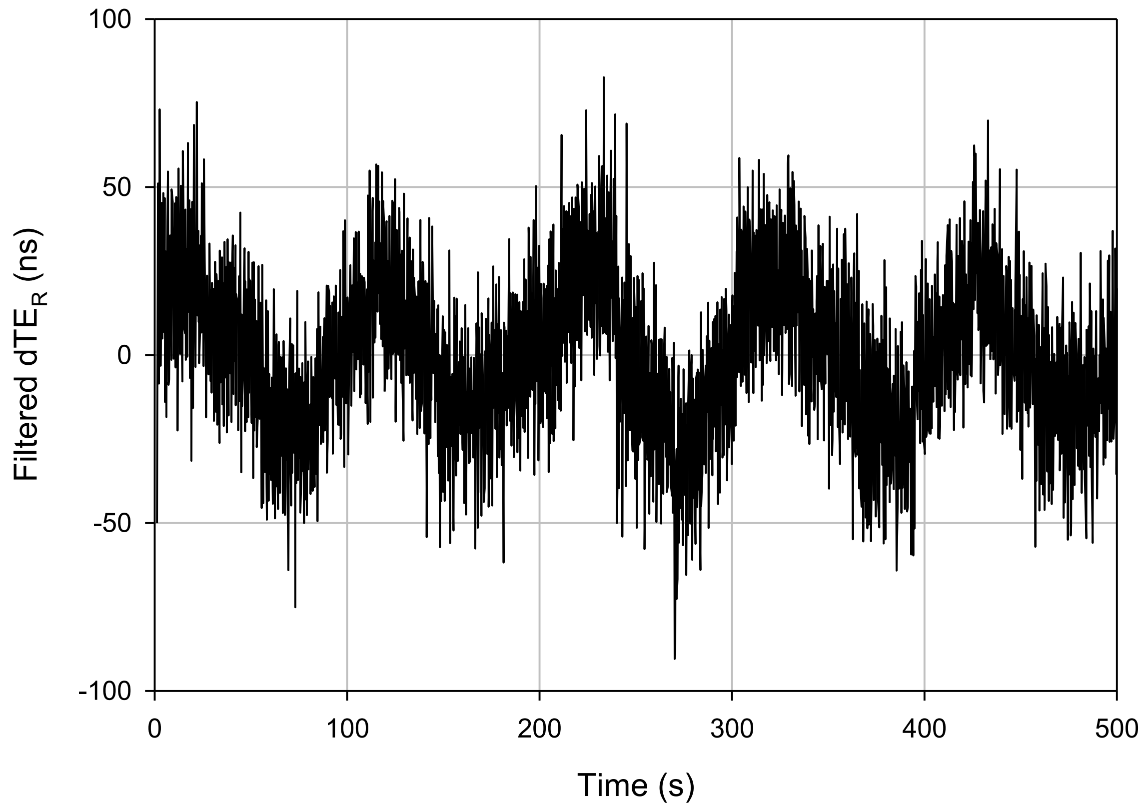
Case I, Node 5, Unfiltered dTE_R

Node 5 dTE_R (relative to GM), before PLL filtering
Case I (correction to nRR+RR; no correction to measured prop delay)



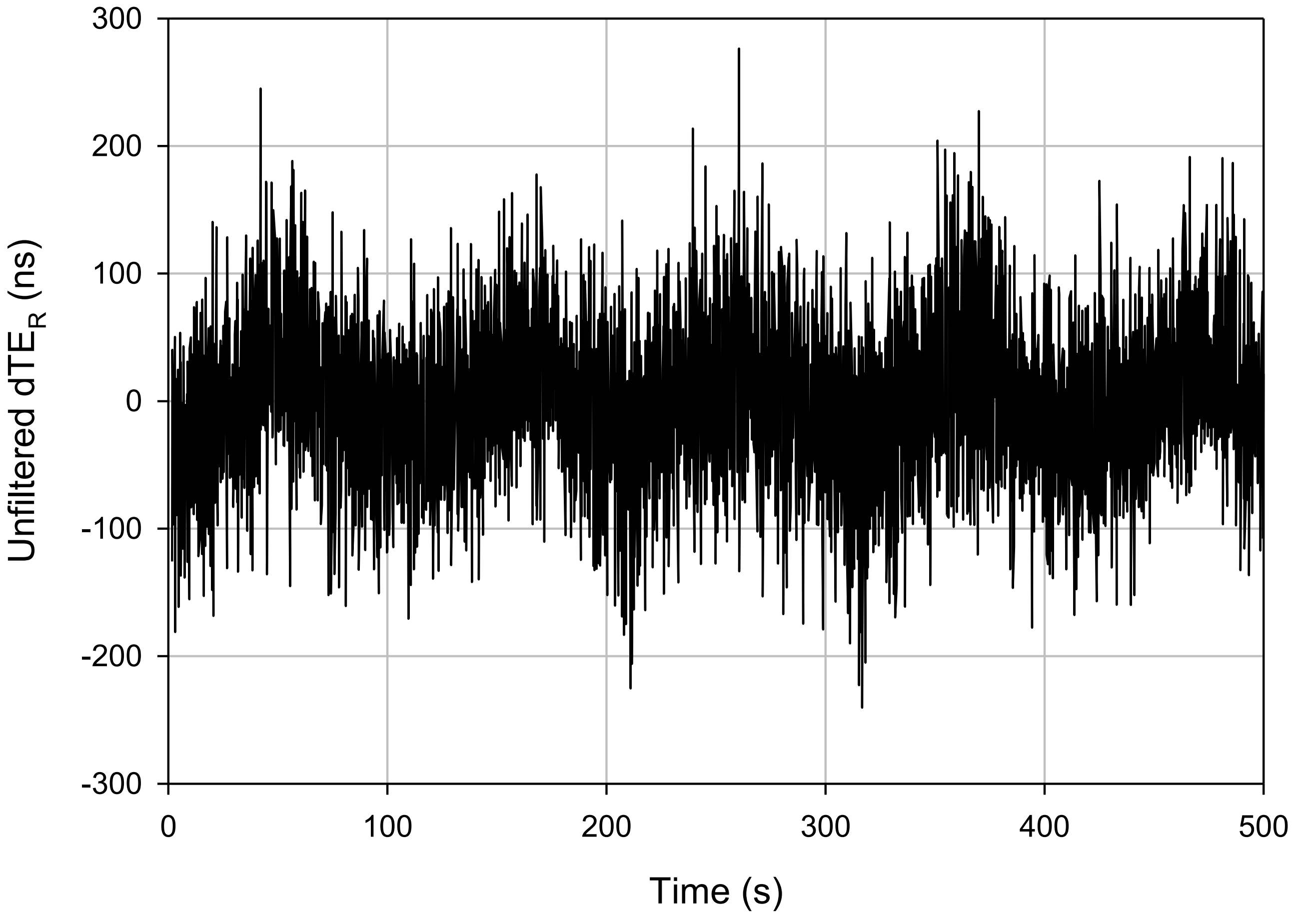
Case I, Node 5, Filtered dTE_R

Node 5 dTE_R (relative to GM), after PLL filtering
Case I (correction to nRR and RR; no correction to measured prop delay)



Case I, Node 10, Unfiltered dTE_R

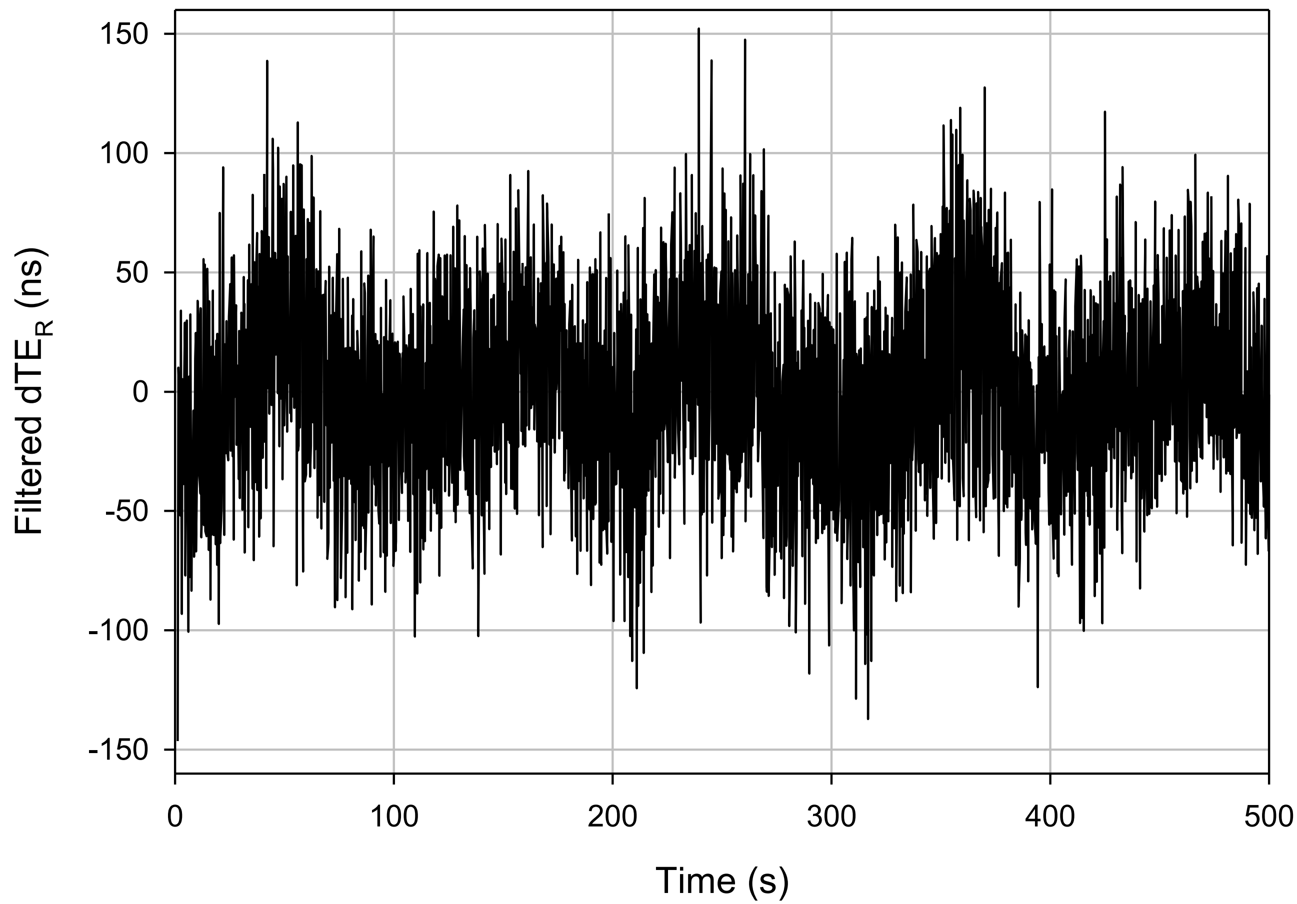
Node 10 dTE_R (relative to GM), before PLL filtering
Case I (correction to nRR+RR; no correction to measured prop delay)



Case I, Node 10, Filtered dTE_R

Node 10 dTE_R (relative to GM), after PLL filtering

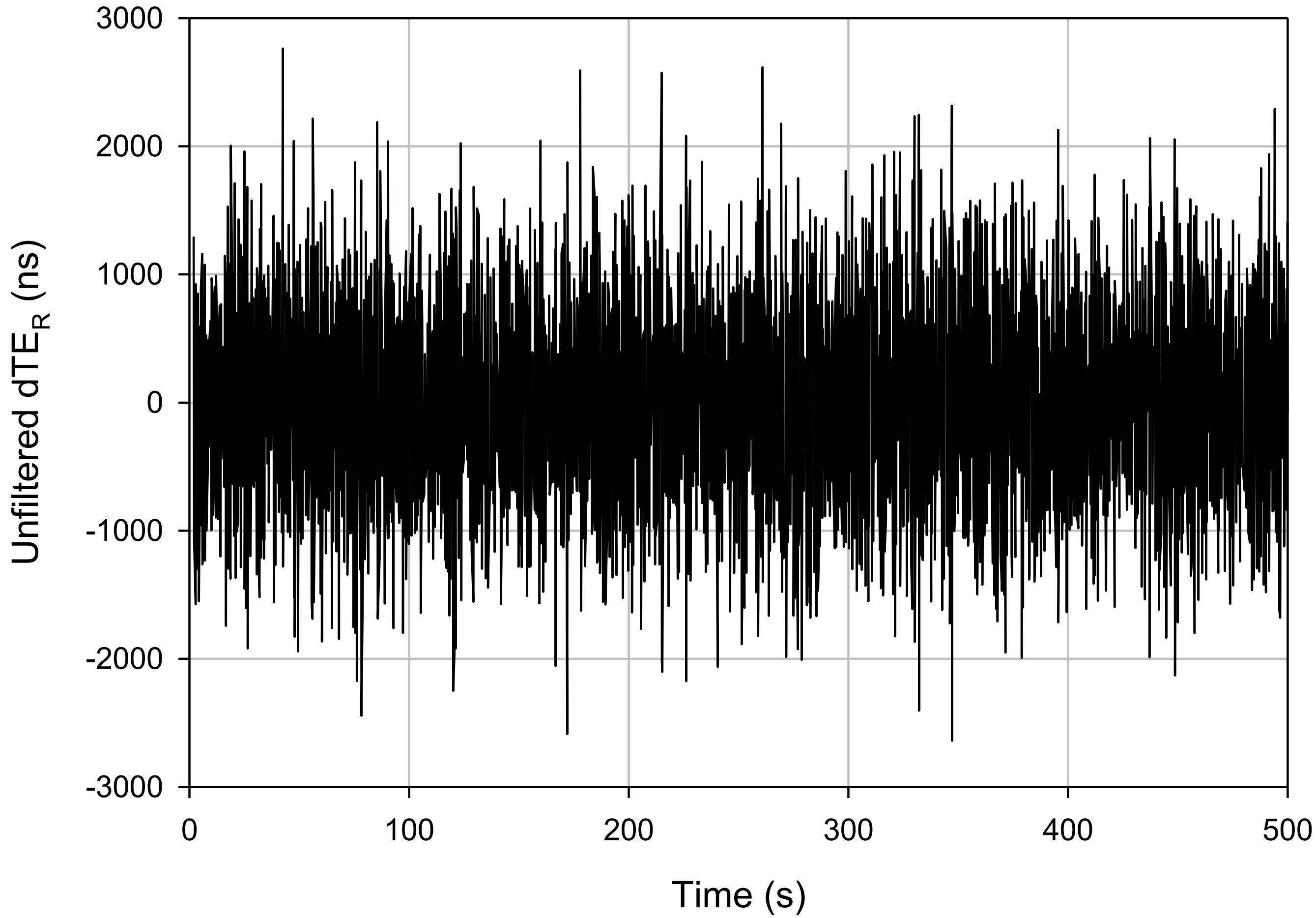
Case I (correction to nRR and RR; no correction to measured prop delay)



Case I, Node 50, Unfiltered dTE_R

Node 50 dTE_R (relative to GM), before PLL filtering

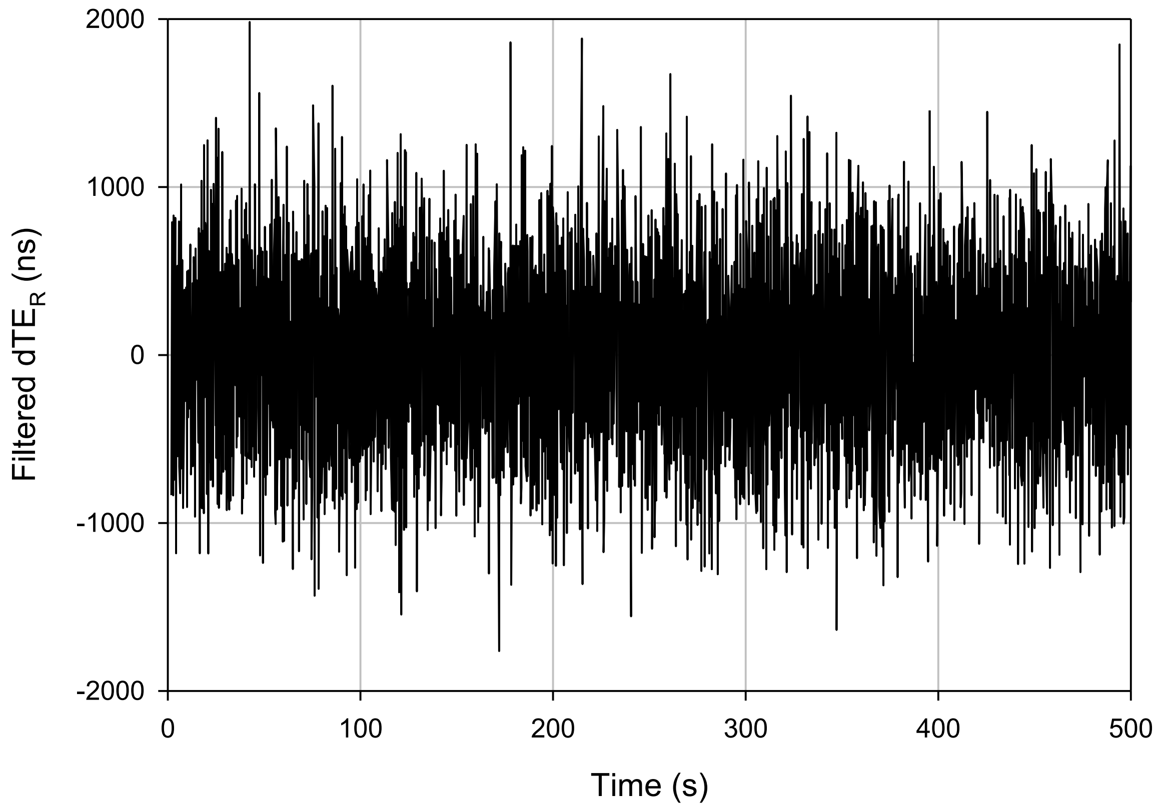
Case I (correction to nRR+RR; no correction to measured prop delay)



Case I, Node 50, Filtered dTE_R

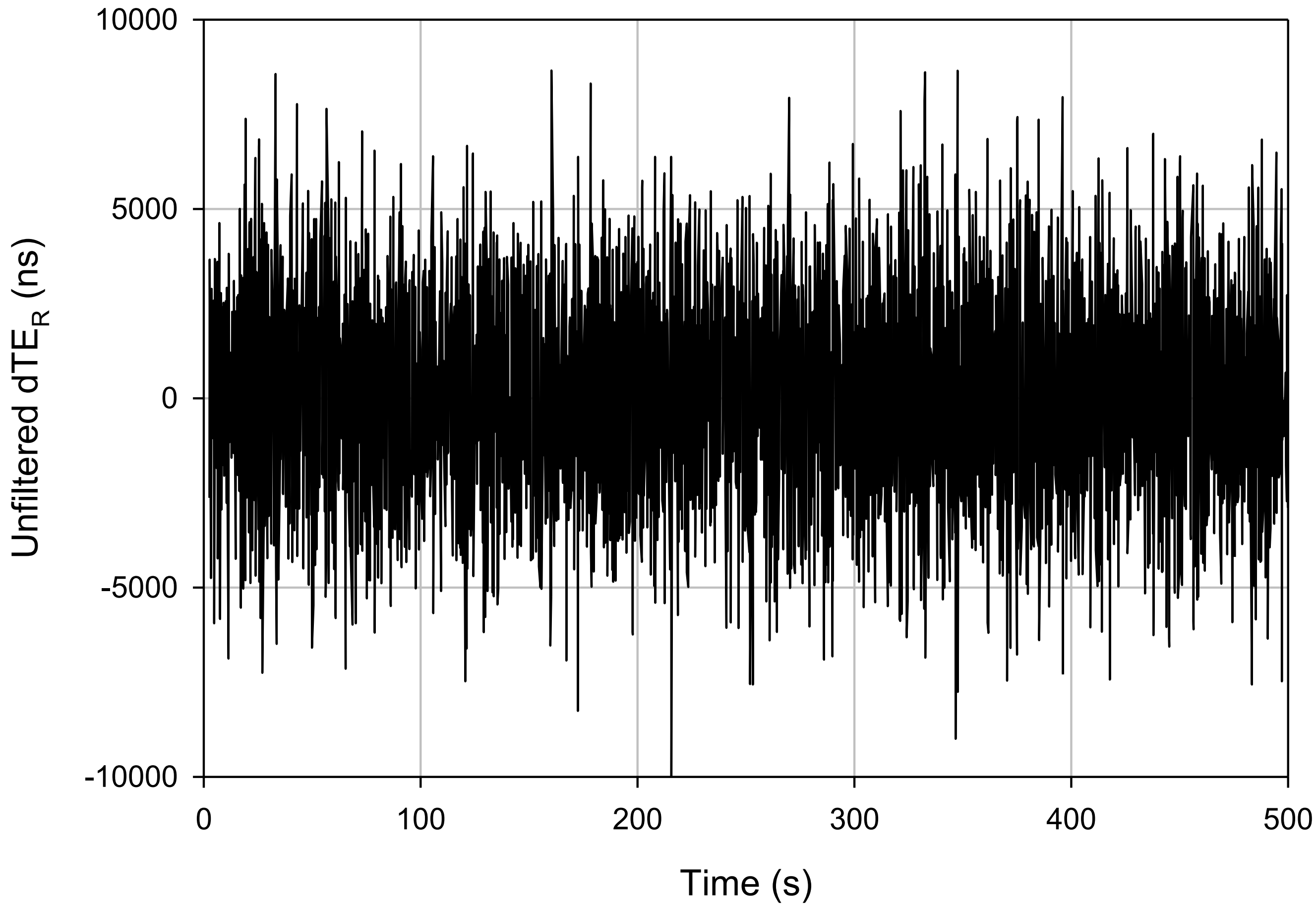
Node 50 dTE_R (relative to GM), after PLL filtering

Case I (correction to nRR and RR; no correction to measured prop delay)



Case I, Node 101, Unfiltered dTE_R

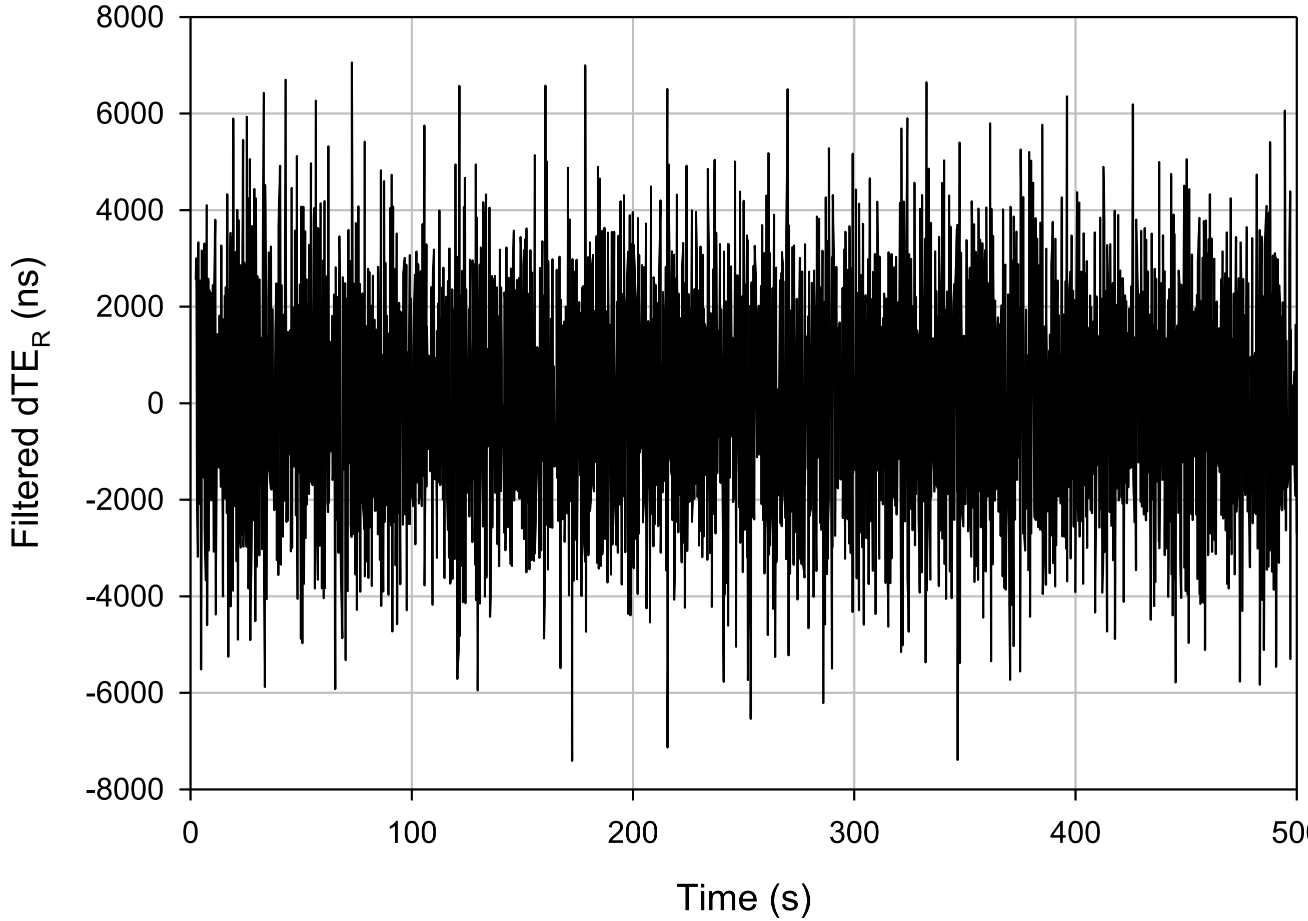
Node 101 dTE_R (relative to GM), before PLL filtering
Case I (correction to nRR+RR; no correction to measured prop delay)



Case I, Node 101, Filtered dTE_R

Node 101 dTE_R (relative to GM), after PLL filtering

Case I (correction to nRR and RR; no correction to measured prop delay)



Conclusions - 1

- The largest errors in measured RR (measured FFO) relative to the GM are due to the abrupt changes in the slope of the GM and local clock frequency offsets at each node
- The NRR algorithmic improvement results in a small improvement in max|dTE_R|; the improvement is not sufficient to enable the 1 μs objective to be met over either 100 hops or 64 hops with the assumptions and parameters used here
  - The errors in measured NRR and resulting errors in measured RR are largest at times of abrupt changes in the slopes of the local clock and GM FFO waveforms

Conclusions - 2

- The RR algorithmic improvement actually causes max$|dTE_R|$ to increase
  - With this algorithm (as well as the NRR improvement), filtered max$|dTE_R|$ is on the order of 4 μs after 64 hops and 9.1 μs after 100 hops
    - i.e., the increase becomes much larger with increasing node number
  - One possible explanation for this large error is that $RR_{driftRate}$ computed on slide 119 (slide 40 of [8]) is multiplied by the correctionField, i.e., the time since transmission of the Sync information from the GM (and also multiplied by residenceTime and meanLinkDelay). For appreciable error in $RR_{driftRate}$, the resulting error can be appreciable at nodes further from the GM because the time needed for the Sync information to reach these nodes is larger (and the error in the correction is larger)
- The NRR correction is a linear correction. The NRR drift rate is computed as a finite difference, i.e., difference in NRR at two different times divided by the time difference. The error in this will be largest when the NRR slope changes abruptly (as seen in the simulation results)
- As indicated above, if the $RR_{driftRate}$ is used to correct for frequency drift during the time interval between transmission of Sync by the GM and receipt of the information downstream, the effect of an error in $RR_{driftRate}$ is larger for nodes further downstream due to the longer time required for the Sync information to get there

Thank you

Appendix – Relevant slides from [8] that describe NRR, RR, and MLD averaging Algorithms

Intro to Appendix

- The relevant slides from [8] follow, on slides 106 – 121 here
- The “b” equations were used in the simulations here; this is valid because NRR and RR, expressed as pure fractions, are much less than 1 (the “a” equation slides are included in what follows for completeness; however, these equations were not used)
- The following changes were made when implementing the algorithms in the time domain simulator
  - In computing $RR_{driftRate}(n)$ using the equation on slide 118 (slide 39 in [8]), $t_{1syncOut}(p)$ and $t_{1syncOut}(p-1)$ were evaluated on arrival of Sync to the node rather than departure, because RR is evaluated on receipt of Sync (in the PortSyncSyncReceive state machine (Figure 10-4 of 802.1AS-2020)
  - The equation for $RR_{driftCorrection}$ on slide 119 (slide 40 in [8]), the residenceTime is needed. However, the computation of residenceTime depends on $RR_{driftCorrection}$. In the simulator, residenceTime is first computed assuming $RR_{driftCorrection}$ is zero. Then, $RR_{driftCorrection}$ is computed using this residence time. Then, residenceTime is re-computed accounting for $RR_{driftCorrection}$.

Conventions

- p-1 is back in time
  - mNRR(p) is the most recent mNRR calculation
  - mNRR(p-1) is the mNRR calculation one prior (in time) to the most recent
- n-1 is back up the chain of nodes
  - RR(n) is the RR calculation for the current node's Local Clock (against the GM)
  - RR(n-1) is the RR same calculation for the node one step up the chain
- If n and/or p are omitted the value is the most recent for the current node

Note

- Two sets of equations are presented for NRR and RR Drift Correction
  - “a” assumes RR and NRR are represented as ratios that are multiplied together
  - “b” assumes RR and NRR are represented as ppm values and added together
    - This introduces an error, but if ppm values are small (one or two digits) the error is similarly small and can be ignored in many practical applications

NRR Drift Correction – Time Series – 1a Ratio

Correction factor applied to RR to account for clock drift between the Local Clock and the Local Clock (or GM) in the previous node (i.e. NRR) during time between NRR measurement and Sync message

- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing; mNRRsmoothingN = 3
- Calculate Drift Rate (mNRR as Ratio)

$$NRR_{driftRate}(n) = \frac{1}{Time_{effectiveNRRmeasure}(p) - Timeeffec_{tiveNRRmeasure}(p-1)} \times \frac{mNRR(p)}{mNRR(p-1)}$$ **ratio/s**

$$= \frac{1}{\left(\frac{t_{4pDelayResp}(p) + t_{4DelayResp}(p-3)}{2}\right) - \left(\frac{t_{4DelayResp}(p-1) + t_{4DelayResp}(p-4)}{2}\right)} \times \frac{mNRR(p)}{mNRR(p-1)}$$ **ratio/s**

$$= \frac{2\times\left(mNRR(p)\right)}{mNRR(p-1)\times\left(t_{4DelayResp}(p) - t_{4DelayResp}(p-1) + t_{4DelayResp}(p-3) - t_{4DelayResp}(p-4)\right)}$$ **ratio/s**

NRR Drift Correction – Time Series – 2a Ratio

Correction factor applied to RR to account for clock drift between the Local Clock and the Local Clock (or GM) in the previous node (i.e. NRR) during time between NRR measurement and Sync message

- Where previously

- Now $$RR(n) = RR(n-1) \times mNRR(n)$$ **ratio**

$$RR(n) = RR(n-1) \times mNRR(p) \times \left(NRR_{driftRate}(n) \cdot \left(t_{1syncOut}(n) - Time_{effectiveNRRmeasure}(p)\right)\right)$$ **ratio**

$$= RR(n-1) \times mNRR(p) \times \left(NRR_{driftRate}(n) \cdot \left(t_{1syncOut}(n) - \left(\frac{t_{4pDelayResp}(p) + t_{4DelayResp}(p-3)}{2}\right)\right)\right)$$ **ratio**

RR Drift Correction – Sync Messaging – Time Series – 1a Ratio

Correction factor applied to Correction Field during processing of Sync message to account for drift between Local Clock & GM during time from GM's transmit of initial Sync message

- Applied during processing of Correction Field at all nodes.
  - Residence Time and Mean Link delay for Bridges; Mean Link Delay only for End Stations.
  - Note: where a device functions as both Bridge and End Station, there are two version of the Correction Field; one for local use (MLD only); one for transmitted Sync messaging (MLD & RT).
- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing (apart from correctionField)
- Calculate Drift Rate

$$RR_{driftRate}(n) = \frac{1}{t_{1syncOut}(p) - t_{1syncOut}(p-1)} \times \frac{RR(p)}{RR(p-1)}$$

ratio

- $t_{1syncOut}(p)$ is the timestamp for when the current node (n) transmits Sync to the next node in the chain.
- RR(p) is the Rate Ratio calculated when Sync is transmitted

RR Drift Correction – Sync Messaging – Time Series – 2a Ratio

Correction factor applied to Correction Field during processing of Sync message to account for drift between Local Clock & GM during time from GM's transmit of initial Sync message

- Where previously

$$correctionField(n) = correctionField(n-1) + RR(n) \cdot (residenceTime + meanLinkDelay)$$ **ns**

$$correctionField(n) = correctionField(n-1) + RR(n) \cdot RRdriftCorrection \cdot (residenceTime + meanLinkDelay)$$ **ns**

$$RR_{driftCorrection} = RR_{driftRate}(n) \times \left(\frac{correctionField(n-1)}{RR(n)} + residenceTime + meanLinkDelay \right)$$ **ppm**

- correctionField is in terms of GM Clock
- residenceTime is only applied to Correction Field when generating Sync messaging for TX to next node in the chain

RR Drift Correction – ES – Time Series – 1a Ratio

Correction factor applied to applied RR at End Station during time between arrival of Sync messages

- Applied at End Stations between arrival of Sync Messages.
- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing.
- Calculate Drift Rate

$$RR_{driftRate}(n) = \frac{1}{t_{2syncIn}(p) - t_{2syncIn}(p-1)} \times \frac{RR(p)}{RR(p-1)}$$

ratio

- $t_{2syncIn}(p)$ is the timestamp for when the current node (n) receives Sync to the previous node in the chain.
- RR(p) is the Rate Ratio calculated when Sync is transmitted

RR Drift Correction – ES – Time Series – 2a Ratio

Correction factor applied to applied RR at End Station during time between arrival of Sync messages

- Where previously...

$$RR_{applied} = RR(n)$$

...and remained constant until next Sync message arrives

Ratio

- Now...

$$RR_{applied} = RR(n) \cdot RR_{driftRate}(n) \cdot \left(\frac{correctionField(n-1)}{RR(n)} + timeElapsedSinceSync \right)$$

Ratio

...and therefore constantly changing, albeit linearly.

- Application to Time Series will involve quadratic equations

NRR Drift Correction – Time Series – 1b ppm

Correction factor applied to RR to account for clock drift between the Local Clock and the Local Clock (or GM) in the previous node (i.e. NRR) during time between NRR measurement and Sync message

- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing; mNRRsmoothingN = 3
- Calculate Drift Rate (mNRR as Ratio)

$$NRR_{driftRate}(n) = \frac{(mNRR(p) - mNRR(p-1))}{Time_{effectiveNRRmeasure}(p) - Timeeffe_{ctiveNRRmeasure}(p-1)}$$ **ppm/s**

$$= \frac{(mNRR(p) - mNRR(p-1))}{\left(\frac{t_{4pDelayResp}(p) + t_{4DelayResp}(p-3)}{2}\right) - \left(\frac{t_{4DelayResp}(p-1) + t_{4DelayResp}(p-4)}{2}\right)}$$ **ppm/s**

$$= \frac{2\times(mNRR(p) - mNRR(p-1))}{\left(t_{4DelayResp}(p) - t_{4DelayResp}(p-1) + t_{4DelayResp}(p-3) - t_{4DelayResp}(p-4)\right)}$$ **ppm/s**

NRR Drift Correction – Time Series – 2b ppm

Correction factor applied to RR to account for clock drift between the Local Clock and the Local Clock (or GM) in the previous node (i.e. NRR) during time between NRR measurement and Sync message

- Where previously

- Now $RR(n) = RR(n-1) + mNRR(n)$ **ppm**

$$RR(n) = RR(n-1) + mNRR(p) + \left(NRR_{driftRate}(n) \cdot \left(t_{1syncOut}(n) - Time_{effectiveNRRmeasure}(p)\right)\right)$$ **ppm**

$$= RR(n-1) + mNRR(p) + \left(NRR_{driftRate}(n) \cdot \left(t_{1syncOut}(n) - \left(\frac{t_{4pDelayResp}(p) + t_{4DelayResp}(p-3)}{2}\right)\right)\right)$$ **ppm**

RR Drift Correction – Sync Messaging – Time Series – 1b ppm

Correction factor applied to Correction Field during processing of Sync message to account for drift between Local Clock & GM during time from GM's transmit of initial Sync message

- Applied during processing of Correction Field at all nodes.
  - Residence Time and Mean Link delay for Bridges; Mean Link Delay only for End Stations.
  - Note: where a device functions as both Bridge and End Station, there are two version of the Correction Field; one for local use (MLD only); one for transmitted Sync messaging (MLD & RT).
- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing (apart from correctionField)
- Calculate Drift Rate

$$RR_{driftRate}(n) = \frac{RR(p) - RR(p-1)}{t_{1syncOut}(p) - t_{1syncOut}(p-1)}$$

ppm

- $t_{1syncOut}(p)$ is the timestamp for when the current node (n) transmits Sync to the next node in the chain.
- RR(p) is the Rate Ratio calculated when Sync is transmitted

RR Drift Correction – Sync Messaging – Time Series – 2b ppm

Correction factor applied to Correction Field during processing of Sync message to account for drift between Local Clock & GM during time from GM's transmit of initial Sync message

- Where previously
- Now

$$correctionField(n) = correctionField(n-1) + \left(1 + \frac{RR(n)}{10^6}\right) \cdot (residenceTime + meanLinkDelay)$$ **ns**

$$correctionField(n) = correctionField(n-1) + \left(1 + \frac{RR(n) + RRdrif_{tCorrection}}{10^6}\right) \cdot (residenceTime + meanLinkDelay)$$ **ns**

$$RR_{driftCorrection} = RR_{driftRate}(n) \times \left(\frac{correctionField(n-1)}{RR(n)} + residenceTime + meanLinkDelay\right)$$ **pp m**

- correctionField is in terms of GM Clock

RR Drift Correction – ES – Time Series – 1a Ratio

Correction factor applied to applied RR at End Station during time between arrival of Sync messages

- Applied at End Stations between arrival of Sync Messages.
- Assumes clock drifts linearly over the period of interest.
- Calculations based on Local Clock timing.
- Calculate Drift Rate

$$RR_{driftRate}(n) = \frac{RR(p) - RR(p-1)}{t_{2syncIn}(p) - t_{2syncIn}(p-1)} \quad \textbf{ppm}$$

- $t_{2syncIn}(p)$ is the timestamp for when the current node (n) receives Sync to the previous node in the chain.
- RR(p) is the Rate Ratio calculated when Sync is transmitted

RR Drift Correction – ES – Time Series – 2b ppm

Correction factor applied to applied RR at End Station during time between arrival of Sync messages

- Where previously...

$$RR_{applied} = RR(n)$$ **ppm**

...and remained constant until next Sync message arrives

- Now...

$$RR_{applied} = RR(n) + RR_{driftRate}(n) \cdot \left(\frac{correctionField(n-1)}{RR(n)} + timeElapsedSinceSync \right)$$ **ppm**

...and therefore constantly changing, albeit linearly.

- Application to Time Series will involve quadratic equations

Mean Link Delay Averaging

- Wired connection link delay is very stable
- pDelay measurements can be noisy due to Timestamp Errors
- It should be possible to average out errors over time
  - Low bandwidth IIR filter...but need to be careful about start-up behaviour

Mean Link Delay Averaging – Possible Algorithm

- For p^{th} pDelay measurement since initialisation...

$$MeanLinkDelay(1) = pDelay(1)$$

$$if\ p \leq 1000, \mathrm{F} = \mathrm{X}$$

$$if\ p > 1000, \mathrm{F} = 1000$$

$$MeanLinkDelay(p) = \frac{\big(MeanLinkDelay(p-1)\times(F-1)\big) + pDelay(X)}{F}$$

- So, for example...

$$MeanLinkDelay(100) = \frac{\big(MeanLinkDelay(99)\times(99)\big) + pDelay(100)}{100}$$

$$MeanLinkDelay(10500) = \frac{\big(MeanLinkDelay(10499)\times(999)\big) + pDelay(10500)}{1000}$$

- Reset F if pDelay deviates too much from current MeanLinkDelay?
    - Deviates too much...repeatedly?

References - 1

[1] Geoffrey M. Garner, *New 60802 dTE_R Time Series Simulation Results for Comparison with Monte Carlo Simulation Results*, Revision 2, IEC/IEEE 60802 presentation, March 10, 2022 (available at https://www.ieee802.org/1/files/public/docs2022/60802-garner-mult-replic-time-series-simul-resutls-for-comparison-with-monte-carlo-simuls-v02.pdf).

[2] David McCall, *60802 Dynamic Time Error – Additions – Error due to drift during Sync messaging – Potential Contribution*, IEC/IEEE 60802 Presentation, March 2022 (available at https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Stanton-Time-Sync-Error-Model-and-Analysis-0322-v01.pdf)

[3] Geoffrey M. Garner, *New Simulation Results for dTE for an IEC/IEEE 60802 , Based on New Frequency Stability Model, Revision 1*, IEC/IEEE 60802 presentation, April 9, 2021 (available at https://www.ieee802.org/1/files/public/docs2021/60802-garner-new-simulation-results-new-freq-stab-model-0421-v01.pdf)

References - 2

[4] Geoffrey M. Garner, *New Simulation Results for dTE for an IEC/IEEE 60802 , Based on New Frequency Stability Model, Version (Revision) 1*, IEC/IEEE 60802 presentation, May 3, 2021 (available at https://www.ieee802.org/1/files/public/docs2021/60802-garner-multipe-replic-simulation-results-new-freq-stab-model-0421-v01.pdf)

[5] Geoffrey M. Garner, *New Simulation Results for dTE for an IEC/IEEE 60802 Network, with Variable Inter-Message Intervals, Revision 2*, IEC/IEEE 60802 presentation, July 1, 2021 (available at https://www.ieee802.org/1/files/public/docs2021/60802-garner-single-replic-simul-results-variable-intermsg-intervals-0621-v02.pdf)

[6] David McCall, *60802 Dynamic Time Sync Error – Monte Carlo Analysis Results for Comparison with Time Series Simulations*, Revision v02, IEC/IEEE 60802 presentation, March 2022 (available at https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-Monte-Carlo-Results-for-Time-Series-Comparison-0322-v02.pdf)

References - 3

[7] Geoffrey M. Garner, *Phase and Frequency Offset, and Frequency Drift Rate Time History Plots Based on New Frequency Stability Data*, IEC/IEEE 60802 presentation, March 8, 2021 (available at https://www.ieee802.org/1/files/public/docs2021/60802-garner-temp-freqoffset-plots-based-on-new-freq-stabil-data-0321-v00.pdf)

[8] David McCall, *60802 Dynamic Time Sync Error – Recommended Parameters & Correction Factors, IEC/IEEE 60802 presentation, Revision v04*, March 2022 (available at https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-Recommended-Parameters-Correction-Factors-0322-v04.pdf)