New 60802 Time Domain Simulation Results for Cases with Drift Tracking Algorithms and PLL Noise Generation

Revision 1

Geoffrey M. Garner
Huawei (Consultant)

gmgarner@alum.mit.edu

IEEE 802.1 TSN TG
2023.10.13

Introduction - 1

- Recent time sync work in 60802 has focused on developing clock drift tracking and compensation algorithms that would enable the objective of 1 μs maximum absolute value of time error relative to the grandmaster PTP Instance ($\max|TE_R|$) to be met over 64 hops, and over 100 hops if possible
- The algorithms are described in [1], and extensive Monte Carlo simulations are documented in [2] and in references cited in [1] and [2] (and in references cited in those references)
- The results appear to be very promising, and the next step is to perform time-domain simulation
  - Time domain simulations are needed because they more precisely model the time-dependent effects present in the algorithms
  - Time domain simulations also model endpoint filtering (e.g., PLL filtering) and noise generation at these filters (the Monte Carlo simulations do not model these effects)
  - The Monte Carlo simulations were used to develop the algorithms because they run several orders of magnitude faster than the time-domain simulations

Introduction - 2

- This presentation gives initial time-domain simulation results for cases including
  - The drift tracking and compensation algorithms described in [1]
  - Endpoint filter (e.g., PLL) noise generation, based on the stated assumptions for local clock stability (i.e., frequency drift behavior for the assumed oscillator type and temperature profile
    - Various PLL filter 3dB bandwidths are considered
- Note that while the objective for max|TE_R| is 1 μs, the budget for relative dynamic time error (dTE_R) is 500 ns
  - Reference [3] indicates 600 ns budget for dTE_R; however, it was indicated that 100 ns of this is budgeted for the end application
  - This is relevant because the simulations model dTE_R for the network transport, i.e., they do not model constant time error (cTE) nor the error in the end application
  - Therefore, this presentation takes 500 ns as the objective for max|dTE_R|
- **In this Rev 1, corrections are made on slides 12 and 20**

Assumptions for the Simulation Cases - 1

- The assumptions for the simulation cases are documented in slides 66 – 74 of [2]
  - Several assumptions documented in [2] were not done, due to an oversight; these are indicated in the following slides
  - This will be corrected in subsequent simulations
- The following slides summarize the assumptions
- The third major bullet item of slide 68 of [2] documented 3 sets of assumptions for the drift tracking and error compensation algorithms
  - These will be described later; however, it was decided to add to this a fourth case where no drift tracking and compensation algorithms were used, for comparison

Introduction - 3

- In addition, it was decided to also investigate the effect of noise generation in the endpoint filter (PLL), for various bandwidths
- Finally, an initial case with no grandmaster (GM) noise was simulated; this was done mainly for debugging purposes, but also to see the effect of GM noise
- As a result, a total of 7 sets of assumptions on the endpoint PLL and/or GM noise were considered for each of the four cases
- Therefore, a total of 28 cases were considered

Temperature Profile - 1

- The temperature profile of [2] is a half-sinusoid with dwell time; it is similar to the temperature profile of [4], except that the periods of the sinusoidal increase and decrease are 95 s instead of 125 s
  - The temperature history is assumed to vary between – 40°C and +85°C, as a half sinusoid over 95 s
  - The dwell times are still 30 s, which means that the period of the temperature variation is 250 s instead of 310 s

Temperature Profile - 2

- The variation for the initial increase in the first cycle is therefore

$$T(t) = -A\cos(\omega t + \phi) + B$$

where

T = temperature in deg C

t = time in s

$A = 62.5$ deg C

$B = 22.5$ deg C

$$\omega = \frac{2\pi}{190\text{ s}} = \frac{\pi}{95}\ \text{rad/s}$$

$$\phi = \text{phase of the temperature variation (in rad)} = \frac{(\text{phase in s})\pi}{250\text{ s}}\ \text{rad}$$

- The variation for the subsequent decrease in the first cycle is

$$T(t) = B + A\cos[\omega(t - 125) + \phi]$$

Frequency Stability due to Temperature Variation - 1

- The dependence of frequency offset on temperature is assumed to be as described in [4] and [5] of Reference [5] here and in Reference [6] 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

Frequency Stability due to Temperature Variation - 2

- The above gives the frequency stability for non-GM PTP Instances, as indicated slide 68 of [2]
- For the GM, the frequency offset at a given temperature should be one-half the frequency offset at the same temperature for non-GM PTP Instances, i.e., the coefficients a_0, a_1, a_2, and a_3 should be multiplied by 0.5 for the GM (after being increased by the factor of 1.1)
  - Unfortunately, due to an oversight, this multiplication by 0.5 was not done for initial simulations
  - This means that, for those cases, the resulting max|dTE| results for cases with GM noise present will be somewhat larger than they should be
  - However, the results can be compared with the corresponding case where the GM noise is zero; the results for GM noise at half the amplitude will be somewhere in between the results with the full amplitude and the results with no GM noise
  - Subsequently, some of the cases were re-run with the 0.5 factor applied (this could not be done for all the cases due to lack of time)
  - All the results are presented here (both with and without the 0.5 factor applied)

Frequency Stability due to Temperature Variation - 3

- The phase offset, frequency offset, and frequency drift rate time history plots given in [4] show the qualitative form of the plots; the only difference here is that the period is 250 s instead of 310 s

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., 250 s)

Other Assumptions - 1

- Some of these slides documenting Other Assumptions are adapted from [2]
- The timestamp granularity is assumed to be 8 ns, based on a 125 MHz clock
  - The timestamp is truncated to the next lower multiple of 8 ns
  - At the GM, 4 ns ~~should be~~ is added
    - ~~However, due to an oversight, this was not done~~
- The dynamic timestamp error is assumed to be uniformly distributed over [-6 ns, +6 ns]
- When GM noise is modeled, interpolation is used to compute dTE_R (relative to the GM), because the dTE samples at the GM and at subsequent PTP Instances are not necessarily computed at the same time
- The simulation time is 1300 s, with the first 50 s discarded when computing $\max|dTE_R|$ to eliminate the effect of any startup transient
- A single replication of each simulation case is run (multiple replications will be run in the future, after the number of cases is reduced to those of most interest)

Other Assumptions - 2

Pdelay Interval

- Pdelay is used only to compute meanLinkDelay, and not neighborRateRatio (NRR)
- NRR is computed using successive Sync message (using the syncEgressTimestamp)
- The nominal Pdelay interval is 125 ms
- The actual Pdelay interval is assumed to be uniformly distributed in the range [(0.9)(125 ms), (1.3)(125 ms)] = [112.5 ms, 162.5 ms]

Sync Interval

- The Sync interval is assumed to be uniformly distributed in the range [119 ms, 131 ms]

Residence time

- The residence time is assumed to be a truncated normal distribution with mean of 5 ms and standard deviation of 1.8 ms, truncated at 1 ms and 15 ms
- Probability mass greater than 15 ms and less than 1 ms is assumed to be concentrated at 15 ms and 1 ms, respectively (i.e., truncated values are converted to 15 ms or 1 ms, respectively)

Other Assumptions - 3

❑Pdelay Turnaround Time

- The Pdelay turnaround time is assumed to be a truncated normal distribution with mean of 10 ms and standard deviation of 1.8 ms, truncated at 1 ms and 15 ms
- Probability mass greater than 15 ms and less than 1 ms is assumed to be concentrated at 15 ms and 1 ms, respectively (i.e., truncated values are converted to 15 ms or 1 ms, respectively)

❑Link Delay

- Link delay is assumed to be uniformly distributed between 5 ns and 500 ns
- Link delays are generated randomly at initialization and kept at those values for the entire simulation
- Link asymmetry is not modeled

Other Assumptions - 4

❑Mean Link Delay Averaging

- The averaging function is assumed to be an IIR filter that uses 0.99 of the previously computed value and 0.01 of the most recent measurement
- This is equivalent to the filter of the NOTE of B.4 of 802.1AS-2020, taken as a first-order filter, i.e.,

$$y_k = a_1 y_{k-1} + b_0 x_k$$

- where y_k is the k^{th} filter output, x_k is the k^{th} measurement, $a_1 = 0.99$, and $b_0 = 0.01$

Other Assumptions - 4

- In-Sync Threshold
  - Out of Sync if 3 times in a row the RX (preciseOriginTimestamp + correctionField) – current GM time estimate is greater than 1 μs or less than -1 μs
  - Back in Sync if 3 times in a row the RX (preciseOriginTimestamp + correctionField) – current GM time estimate is less than 1 μs and greater than -1 μs
  - Note that this is modeled in the simulator but was not yet simulated due to an oversight
- Investigation of Timestamp Granularity Error Offset
  - For message egress at GM and ingress at node 1, record distribution of amount of truncation for each timestamp granularity error, and produce probability distribution histograms with bin size 0.1 ns for egress at GM and ingress at node 1
  - This is not yet done due to lack of time prior to the current meeting

Endpoint filter (PLL) Parameters – 1

- In previous simulations, the following were used for the endpoint PLL parameters K_p (proportional gain), K_i (integral gain), K_o(VCO/DCO gain):
  - $K_pK_o = 11$, $K_iK_o = 65$
- This corresponds to the following 3dB bandwidth (f_{3dB}), gain peaking, and damping ratio (ζ)
  - $f_{3dB} = 2.5998$ Hz, 1.288 dB gain peaking, $\zeta = 0.68219$
- In addition, VCO/DCO noise generation was neglected
- The PLL model used in the simulator is second-order and linear, with 20 dB/decade roll-off
  - It is based on a discretization that uses an analytically exact integrating factor to integrate the second-order system
  - As a result, the PLL model in the simulator is stable regardless of the time step, i.e., sampling rate (though aliasing of the input or noise is possible)
  - Details are given in Appendix VIII.2.2 of [7] (except that the relation between gain peaking and damping ratio is based on the exact result in 8.2.3 of [8] (see Eqs. (8-13 – 8-15 there)

Endpoint filter (PLL) Parameters – 2

- However, many practical PLL implementations are based on a discrete time model where the integral block and VCO block of the PLL are modeled based on z-transforms
  - Depending on the details, this is mathematically equivalent to replacing derivatives by forward or backward differences
  - See Appendix I (Figure I-1 and Eq. (I-6) of [8] and 3.5 of [9] for examples
  - As a result, the model becomes unstable if the sampling rate is not large enough compared to the PLL 3dB bandwidth
  - A common rule of thumb is that the sampling rate should be at least ten times the PLL bandwidth
  - The analysis in 3.5 of [9] shows that, for the example there, the theoretical limit for stability is approximately π times the 3dB bandwidth (i.e., the sampling rate must be at least π times the 3dB bandwidth for the PLL to be stable)
  - The PLL 3dB bandwidth above (used in previous simulations) of 2.5998 Hz implies that the sampling rate should be at least 25.998 Hz $\cong$ 26 Hz
  - However, the sampling rate here is the Sync rate, and the minimum Sync rate corresponds to the maximum Sync interval, which is 131 ms
  - The minimum Sync rate is therefore 1/(0.131 s) = 7.634 Hz, which is too small
  - The theoretical limit of π:1 implies a Sync rate of at least (π)(2.6 Hz) = 8.17 Hz, which still exceeds the 7.634 Hz minimum Sync rate

Endpoint filter (PLL) Parameters – 3

- To begin to address this, additional simulation cases were run with various PLL bandwidths smaller than 2.6 Hz (for now, gain peaking was kept at 1.288 dB)
- However, as the PLL bandwidth becomes narrower, noise generation can become appreciable if the same oscillator is used, because the transfer function from the noise to the output is a high-pass filter with corner frequency and damping ratio the same as for the low-pass transfer function from the PLL input to output
- In the case here, it was indicated in one of the July 2023 60802 meeting that the same XO is used for the endpoint PLL filter as for the timestamping function
- Therefore, noise generation was modeled, using the same local oscillator phase variation model used for the LocalClock
  - The noise was computed by passing the XO phase noise through a high-pass filter with the same 3dB bandwidth and damping ratio as the low-pass PLL filter, and adding the result to the PLL output that was computed from the input

Simulation Cases – 1

- In the notation below, mNRRsmoothingNA is the number of Sync Intervals over which nRR is both computed and averaged, e.g., if mNRRsmoothingNA = 4, we compute nRR over 4 Sync intervals and average the 4 most recently computed values
- In the notation below, mNRRcompNAP is the number of Sync Intervals over which the frequency drift rate estimate is computed
- The following four cases are simulated (for each set of assumptions on PLL bandwidth and noise generation)
  - mNRRcompNAP = 8; mNRRsmoothingNA = 4
  - mNRRcompNAP = 8; mNRRsmoothingNA = 8 (corrected from 4)
  - No drift tracking and compensation; mNRRsmoothingNA = 4
  - "Classic" case of 802.1AS-2020: No drift tracking and compensation; no smoothing; use Pdelay messages to estimate NRR

Summary of Simulation Cases - 1

Case	Drift Tracking and Compensation (mNRRcompNAP, mNRRsmoothingNA)	PLL 3dB Bandwidth (Hz)	PLL noise generation present (yes/no)	GM noise magnitude relative to non-GM PTP Instances
1	(8, 4)	2.6	no	0
2	(8, 8)	2.6	no	0
3	(-, 4)	2.6	no	0
4	None (classic 802.1AS)	2.6	no	0
5	(8, 4)	2.6	no	1.0
6	(8, 8)	2.6	no	1.0
7	(-, 4)	2.6	no	1.0
8	None (classic 802.1AS)	2.6	no	1.0
5a	(8, 4)	2.6	no	0.5
6a	(8, 8)	2.6	no	0.5
7a	(-, 4)	2.6	no	0.5
8a	None (classic 802.1AS)	2.6	no	0.5

Summary of Simulation Cases - 2

Case	Drift Tracking and Compensation (mNRRcompNAP, mNRRsmoothingNA)	PLL 3dB Bandwidth (Hz)	PLL noise generation present (yes/no)	GM noise magnitude relative to non-GM PTP Instances
9	(8, 4)	2.6	yes	1.0
10	(8, 8)	2.6	yes	1.0
11	(-, 4)	2.6	yes	1.0
12	None (classic 802.1AS)	2.6	yes	1.0
13	(8, 4)	0.5	yes	1.0
14	(8, 8)	0.5	yes	1.0
15	(-, 4)	0.5	yes	1.0
16	None (classic 802.1AS)	0.5	yes	1.0
17	(8, 4)	2.0	yes	1.0
18	(8, 8)	2.0	yes	1.0
19	(-, 4)	2.0	yes	1.0
20	None (classic 802.1AS)	2.0	yes	1.0

Summary of Simulation Cases - 3

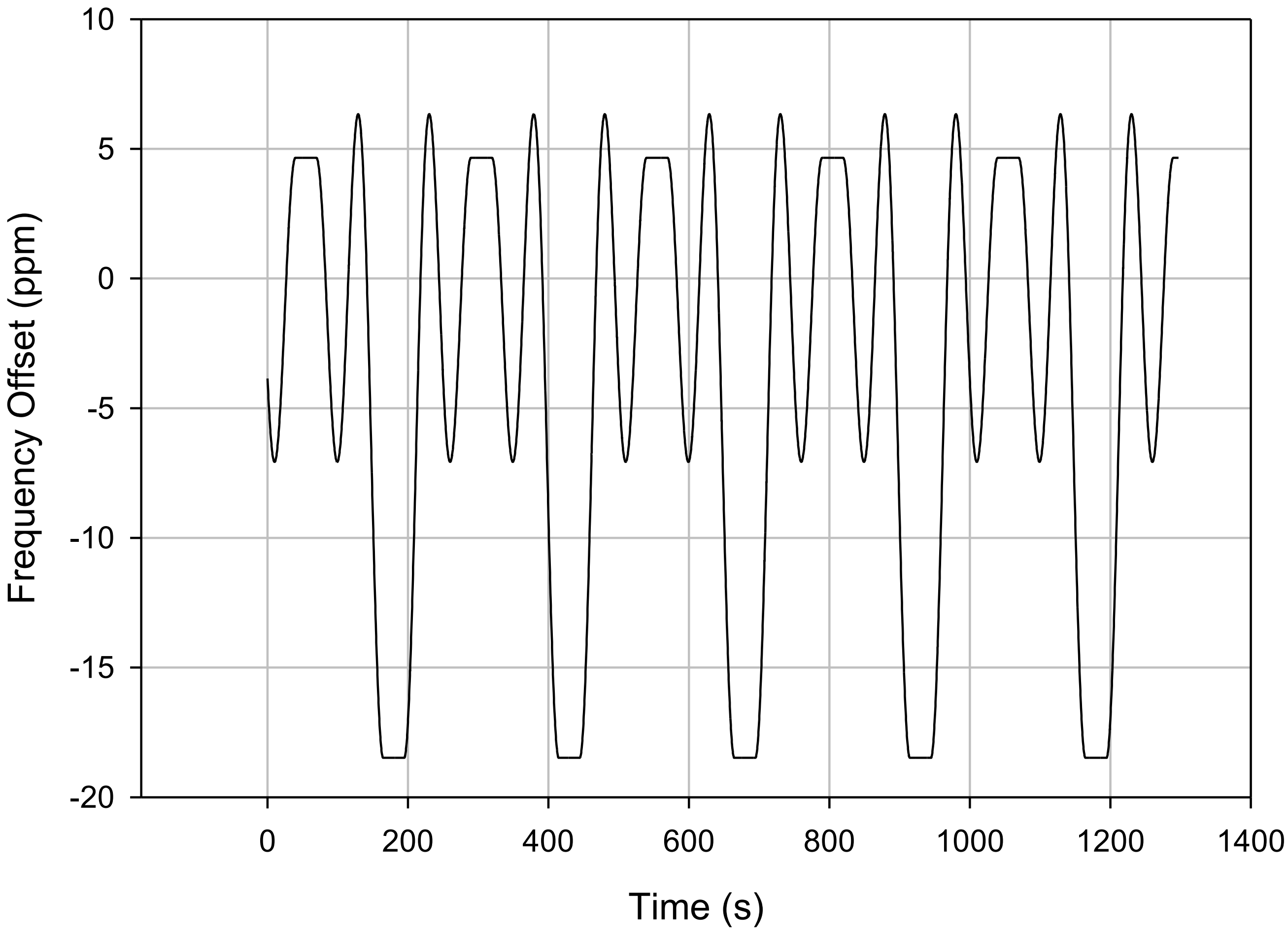
Case	Drift Tracking and Compensation (mNRRcompNAP, mNRRsmoothingNA)	PLL 3dB Bandwidth (Hz)	PLL noise generation present (yes/no)	GM noise magnitude relative to non-GM PTP Instances
21	(8, 4)	1.0	yes	1.0
22	(8, 8)	1.0	yes	1.0
23	(-, 4)	1.0	yes	1.0
24	None (classic 802.1AS)	1.0	yes	1.0
25	(8, 4)	1.8	yes	1.0
26	(8, 8)	1.8	yes	1.0
27	(-, 4)	1.8	yes	1.0
28	None (classic 802.1AS)	1.8	yes	1.0
25a	(8, 4)	1.8	yes	0.5
26a	(8, 8)	1.8	yes	0.5
27a	(-, 4)	1.8	yes	0.5
28a	None (classic 802.1AS)	1.8	yes	0.5

Case 1, Node 2 Measured nRR Results

- As a check, results for measured nRR at node 2 are shown in the following slides
- Since there is no GM error in Case 1, the RR and nRR at node 2 are the same
- The next slide (slide 25) shows the actual LocalClock frequency offset at node 2, computed from the temperature profile and frequency offset versus temperature relation
    - This has the same qualitative behavior as in [4] (see slides 11 and 12 of [4]); the main difference is that the period of the frequency variation here is 250 s, compared to 310 s in [4]
- The following slide after the next slide (slide 26) shows the measured nRR at node 2, expressed as a frequency offset and multiplied by -1 for comparison with the actual node 2 LocalClock frequency offset (the latter is the negative of the former because the GM error is zero
    - The plots in the two slides visually appear to be the same
- Slide 27 shows the difference between the measured and actual frequency offsets
    - After initialization, the absolute value of the difference is less than 0.1 ppm

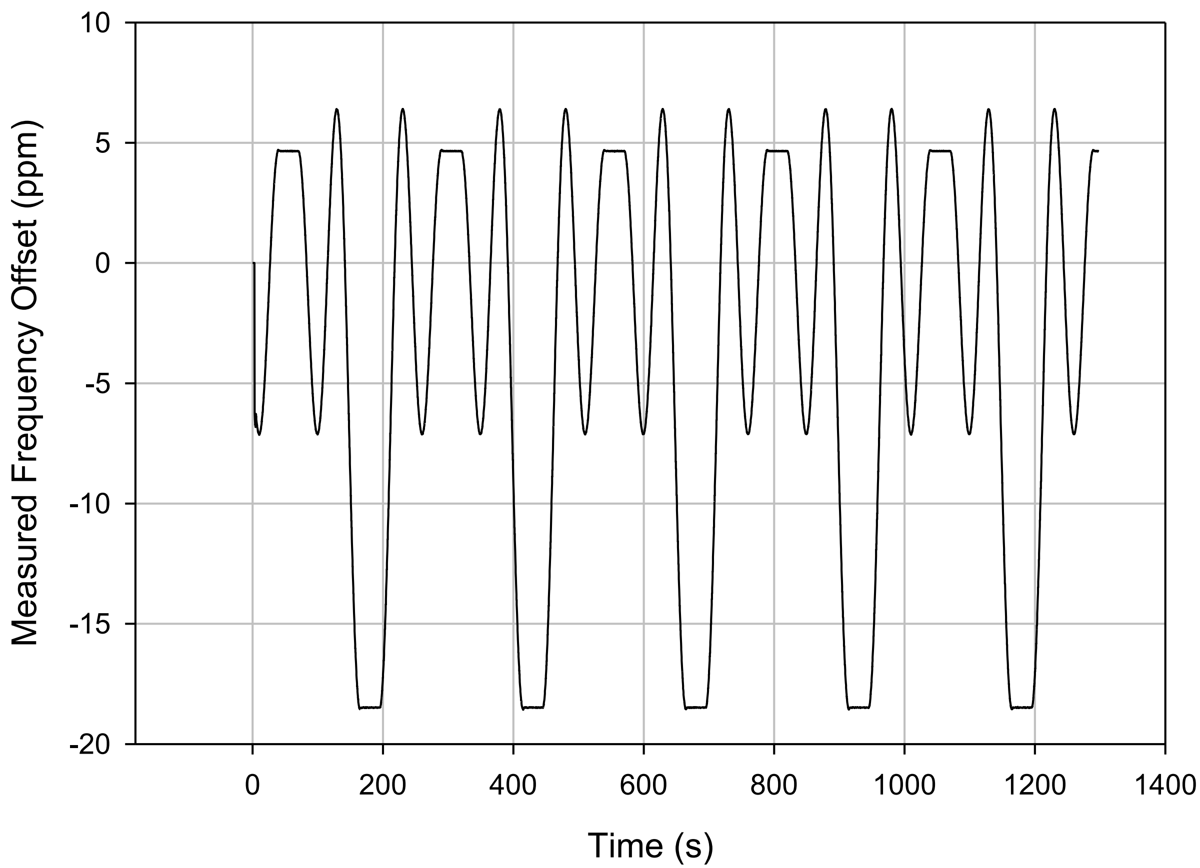
Case 1, Node 2 - LocalClock Frequency Offset Versus Time

Case 1, Node 2 - LocalClock True Frequency Offset
(Note that GM time error is zero for case 1,
which means that measured RR and nRR at Node 2 are equal)



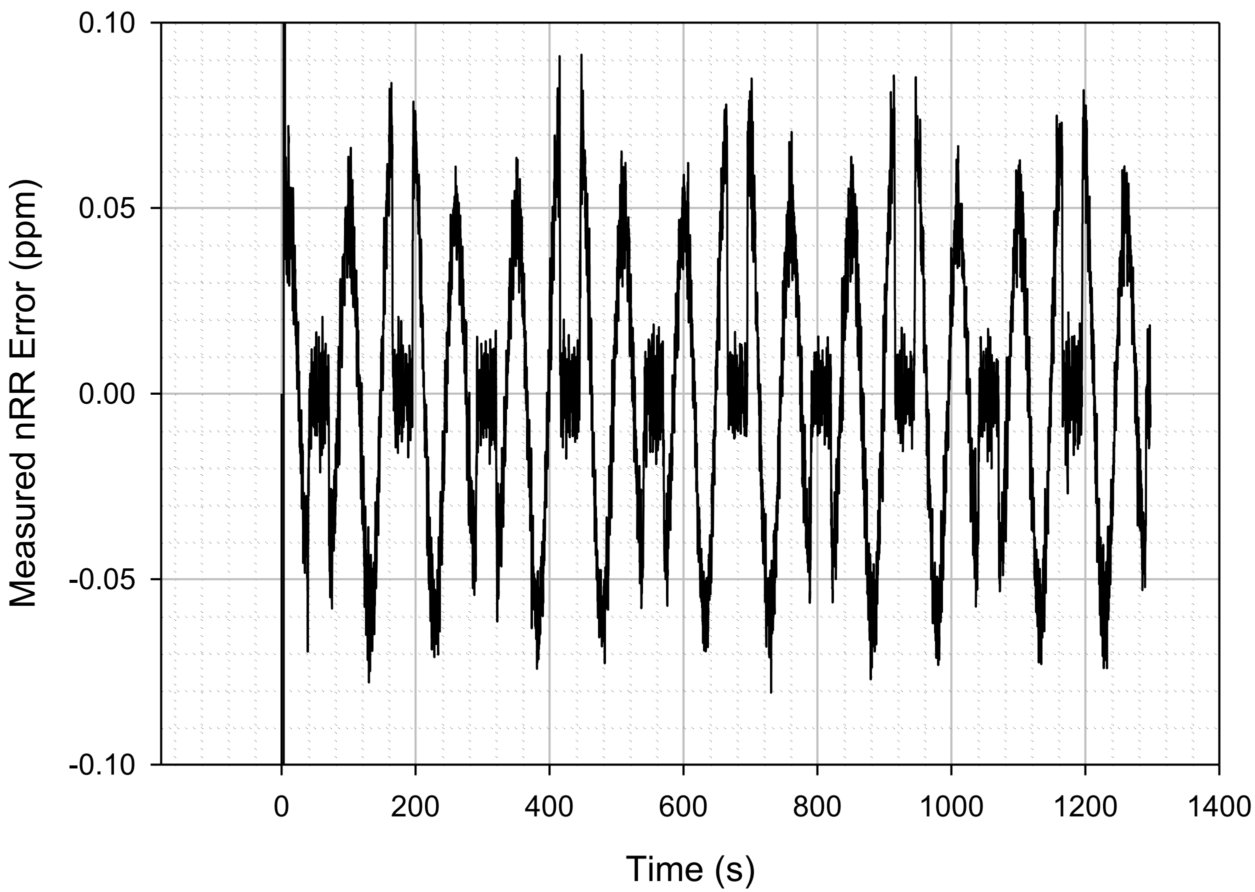
Case 1, Node 2 – Measured Frequency Offset (nRR) Versus Time

Case 1, Node 2 - Measured Frequency Offset, i.e., (nRR-1)*(1.0e6)*(-1)
Result is multiplied by -1 for easy comparison with previous slide
(Note that GM time error is zero for case 1,
which means that measured RR and nRR at Node 2 are equal)



Case 1, Node 2 - Error in Measured Frequency Offset (nRR) Versus Time

Case 1, Node 2 - Error in Measured Frequency Offset, i.e.,
measured Frequency Offset minus
actual frequency offset
(Note that GM time error is zero for case 1,
which means that measured RR and nRR at Node 2 are
equal)



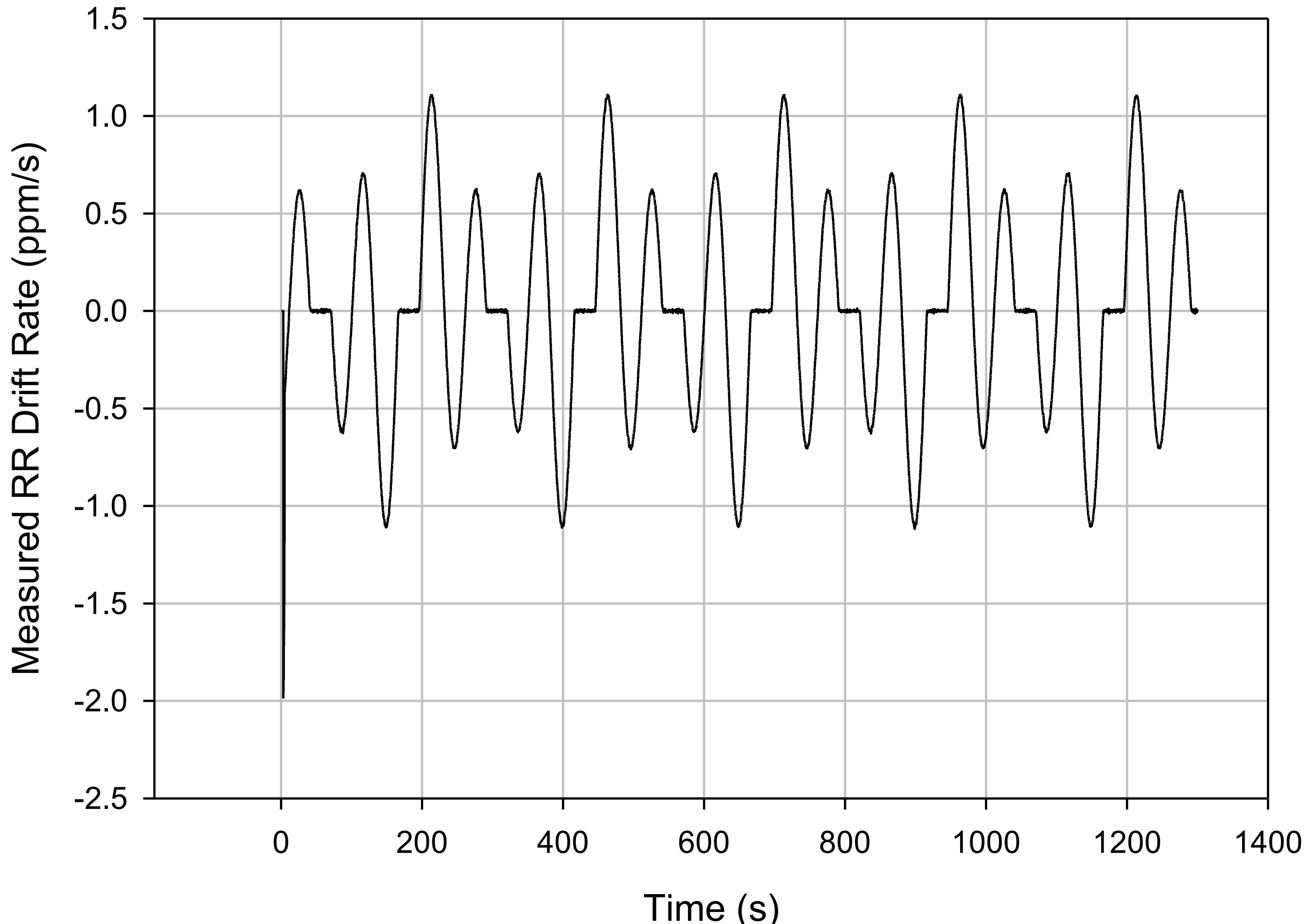
Initial large measure
d error is due to initi
al–
ization of frequency
offset measurement

Case 1, Node 2 Measured RR Drift Rate Results

- As a check, the result for measured RR Drift Rate at node 2 are shown in the following slide (slide 29)
  - The result is multiplied by -1 to facilitate comparison with slides 26 and 27
- Since there is no GM error in Case 1, the RR and nRR drift rate at node 2 are the same
- Comparing slide 29 with slides 26 and 27, it is seen that the measured drift rate is small when the frequency offset is not changing, as expected
- Also, it is seen that measured drift rate is large and positive when nRR is increasing, and large and negative when nRR is decreasing
  - For example, frequency offset on slide 25 increases from approximately -17 ppm to +6 ppm between 200 s and 225 s, i.e., a change of 23 ppm over 25 s or approximately a rate of 1 ppm/s
    - Slide 29 shows that nRR drift rate reaches approximately 1 ppm/s during this interval

Case 1, Node 2 - Measured RR Drift Rate multiplied by -1

Case 1, Node 2 - Measured RR Drift Rate, multiplied by -1
(Note that GM time error is zero for case 1,
which means that measured RR and nRR drift rates at Node 2 are equal)

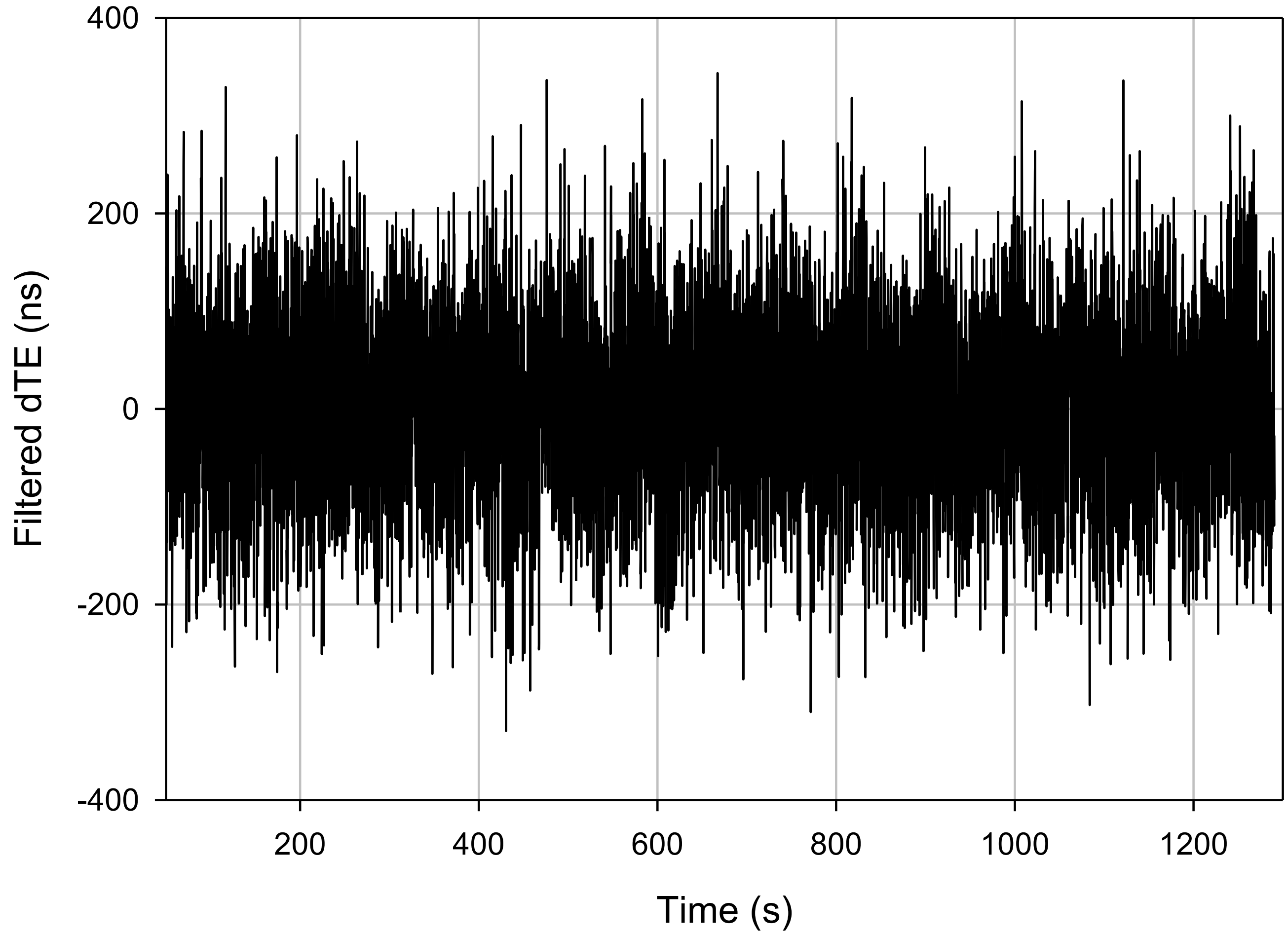


dTE_R Time History Waveforms for Node 101

- The next two slides show filtered and unfiltered dTE_R waveforms for node 101, after the 50 s initialization period
- The absolute value of the filtered dTE_R remains less than approximately 350 ns
- The absolute value of the filtered dTE_R remains less than approximately 400 ns
- The waveforms are qualitatively similar; the effect of the filtering is to reduce the amplitude of dTE_R
- Note that case 1 has no noise generation or GM time error, and the 3 dB bandwidth is 2.6 Hz

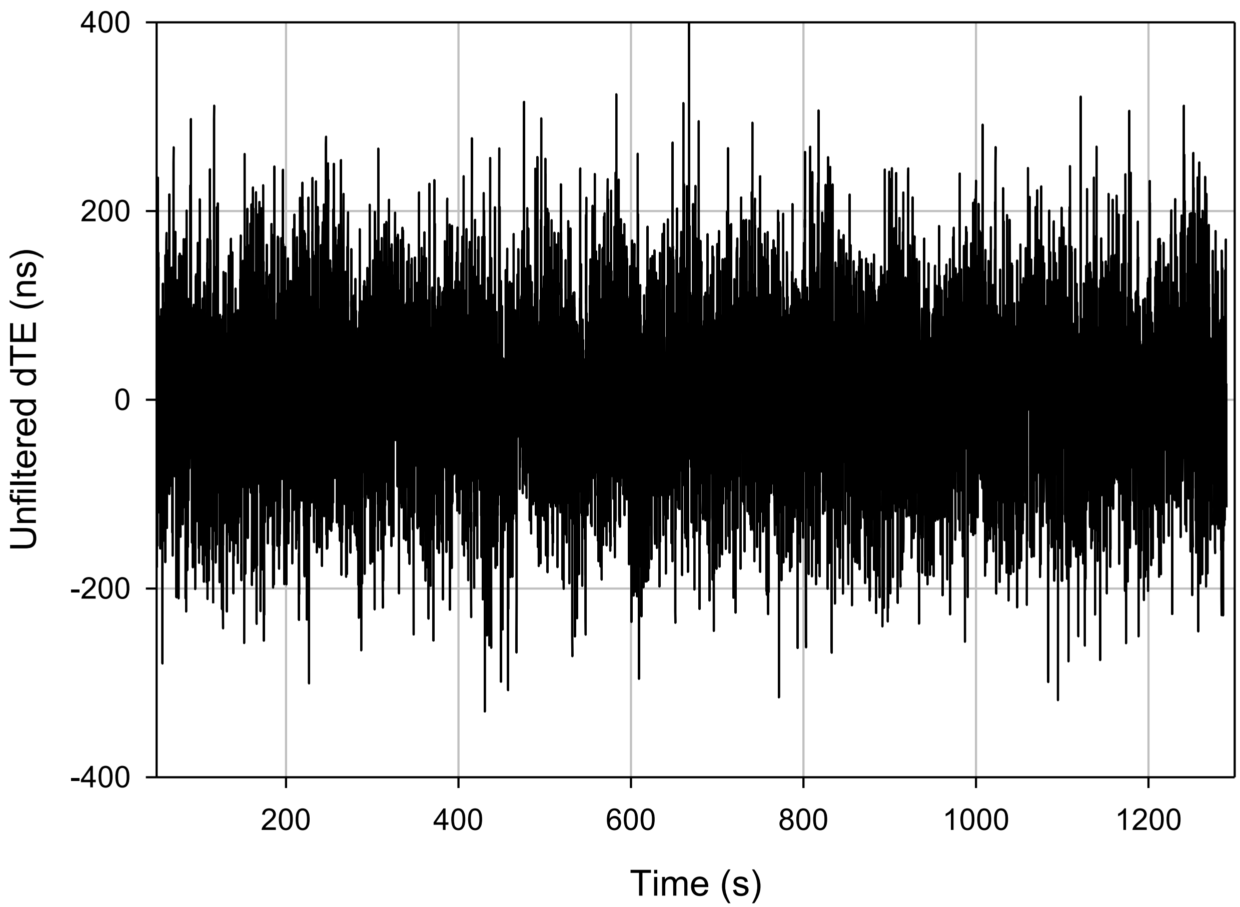
Case 1, Node 101 - Filtered dTE Versus Time

Case 1, Node 101 - Filtered dTE
First 50 s is omitted, to eliminate initialization transient
(note that GM error is zero in Case 1, which
means that dTE and relative dTE are the same)



Case 1, Node 101 - Unfiltered dTE Versus Time

Case 1, Node 101 - Unfiltered dTE
First 50 s is omitted, to eliminate initialization transient
(note that GM error is zero in Case 1, which
means that dTE and relative dTE are the same)

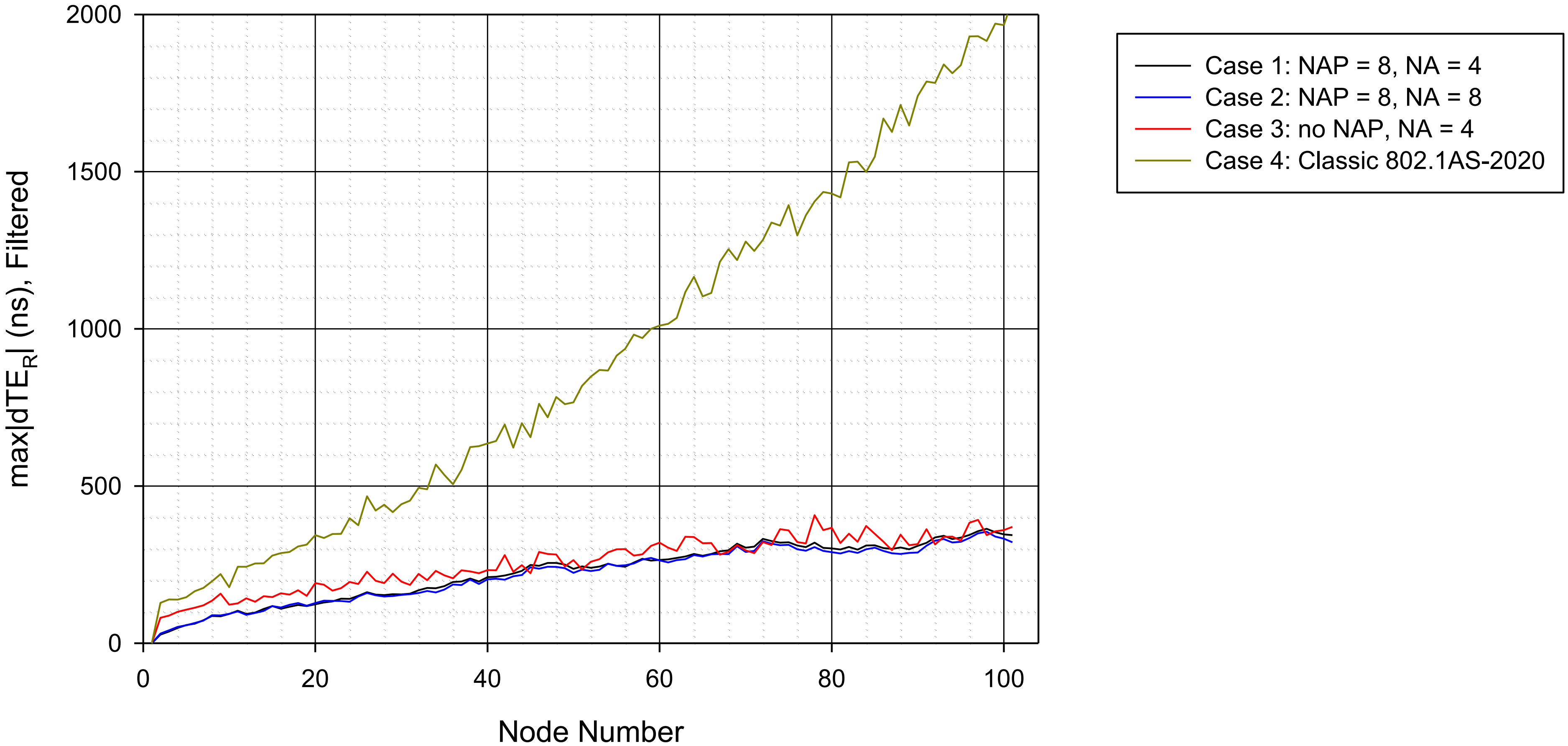


Max|dTE_R| Simulation Results

- Plots of max|dTE_R| are presented on the following slides (34 – 47) for max|dTE_R| before and after endpoint PLL filtering
- Filtered and unfiltered max|dTE_R| for nodes 65 and 101 are summarized int the tables on slides 48 and 49
- For cases 1 – 4, in addition to plots showing the full range of max|dTE_R| for all the cases, detailed plots showing cases 1 -3 are shown
  - For subsequent cases, only the detailed ranges of cases of the three cases with nRR smoothing and, if applicable, compensation are shown, as the classic 802.1AS-2020 cases are of less interest because they do not meet the 500 ns max|dTE_R| objective
  - Also, the unfiltered results are not shown for cases 13 – 28, as they are the same as the results for cases 5 – 8 and cases 9 – 12, respectively (it can be verified that the unfiltered results for the respective cases of cases 5 – 8 and cases 9 – 12 are the same)
    - Unfiltered results for Cases 5a – 8a are shown; unfiltered results for cases 25a – 28a are not shown as these are the same as the results for cases 5a – 8a, respectively

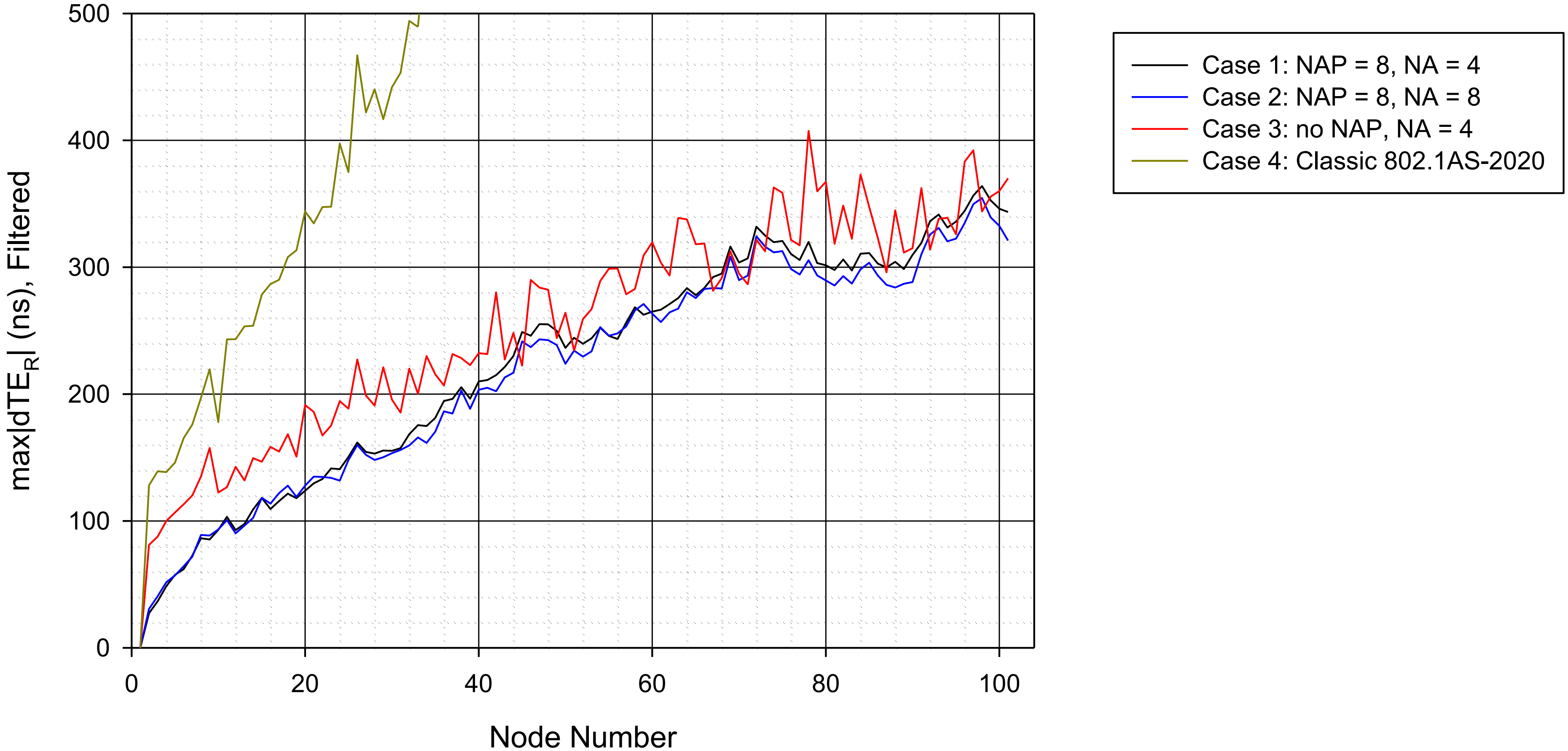
Filtered max|dTE_R|, Cases 1 - 4

Cases 1 - 4
Single replication of simulations
No GM time error
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



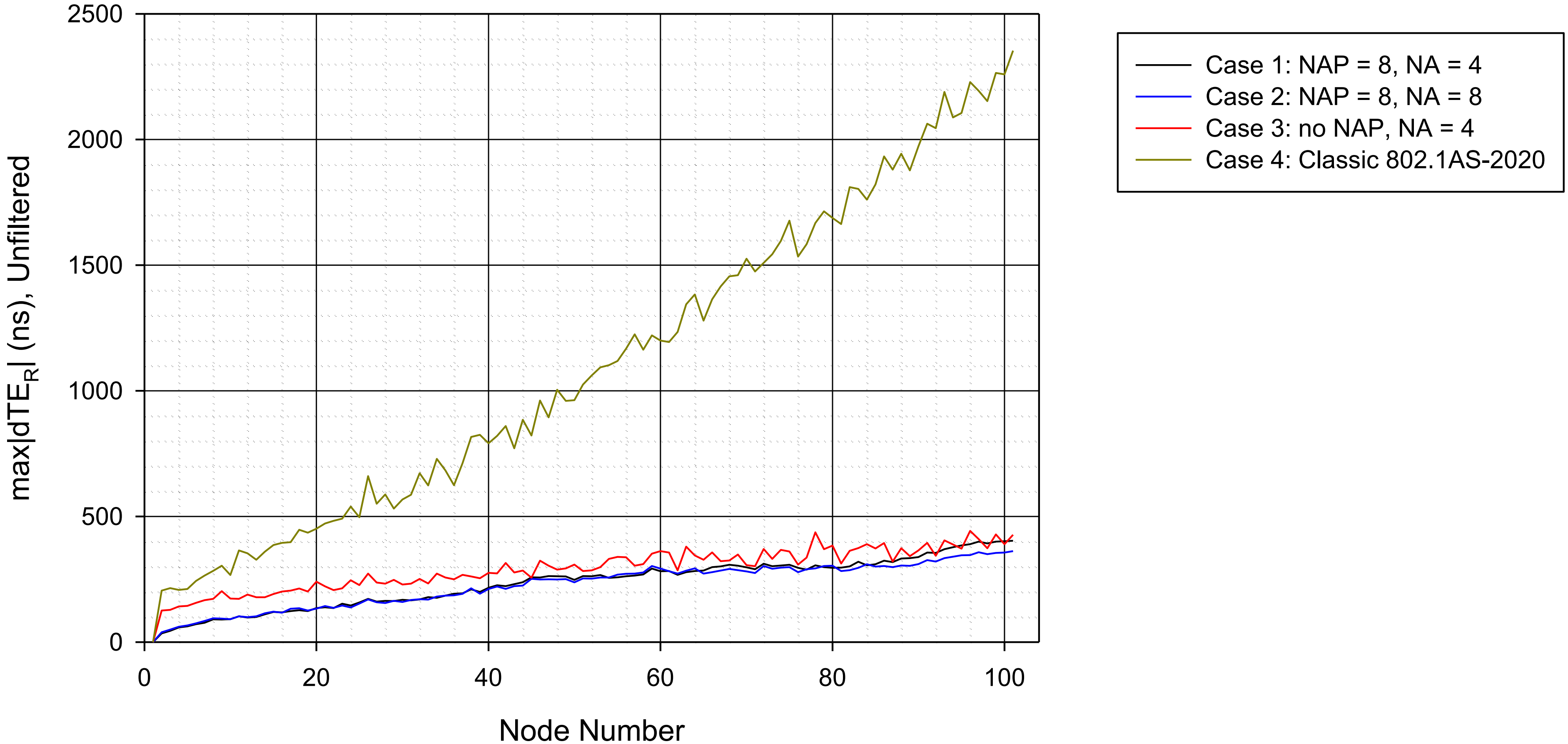
Filtered max|dTE_R|, Cases 1 - 4, detail focus on cases 1 - 3

Cases 1 - 4, detail of cases 1 - 3
Single replication of simulations
No GM time error
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



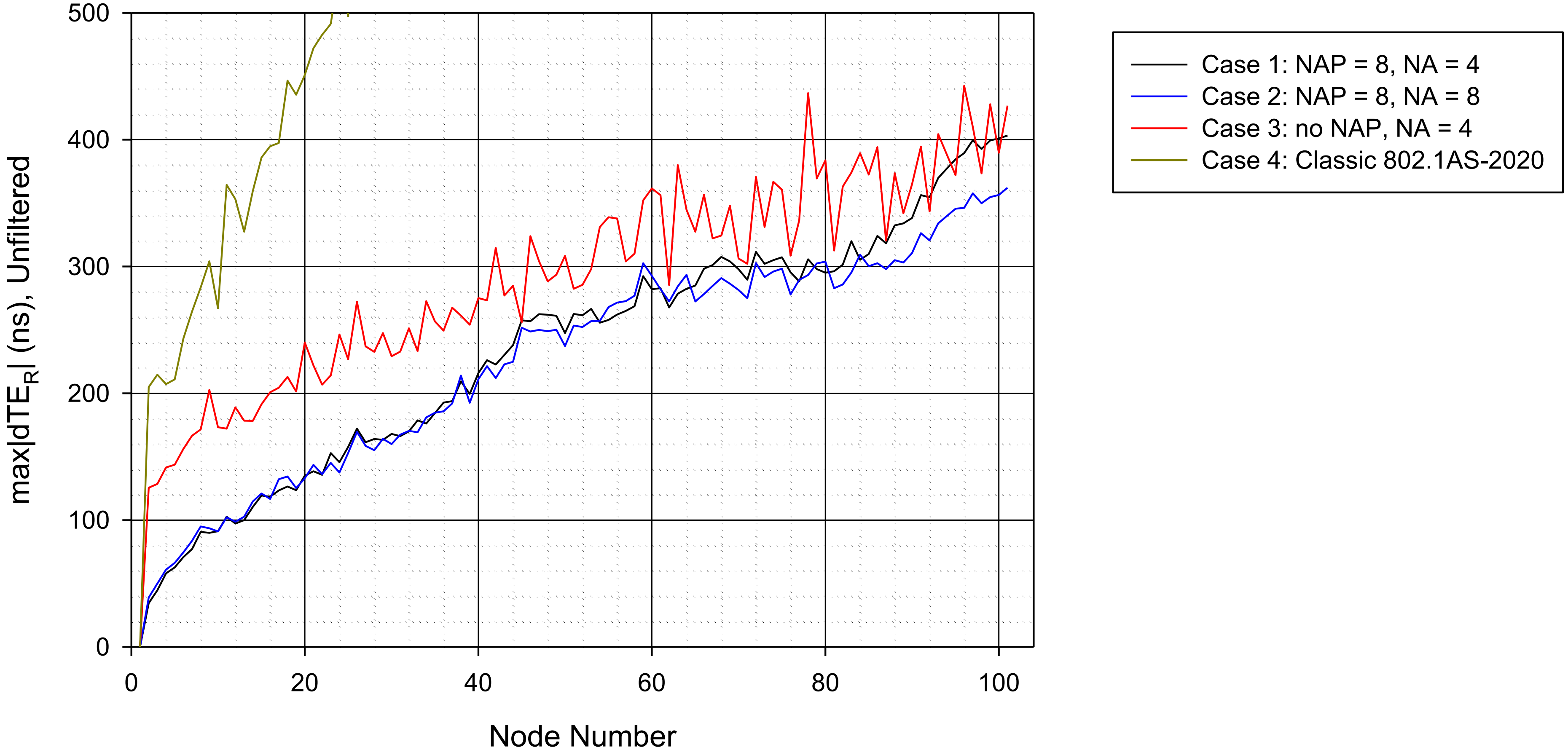
Unfiltered max|dTE_R|, Cases 1 - 4

Cases 1 - 4
Single replication of simulations
No GM time error
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



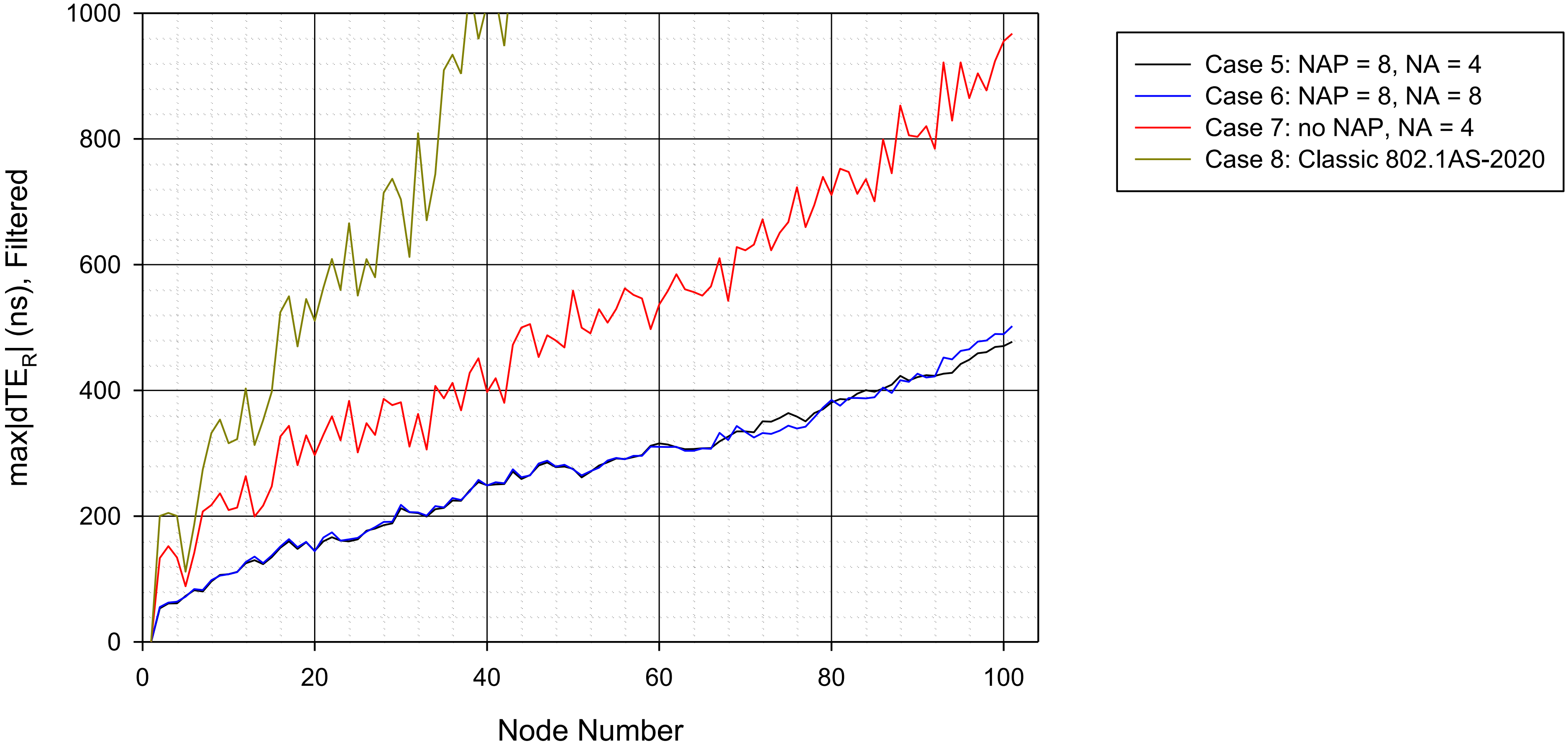
Unfiltered max|dTE_R|, Cases 1 - 4, detail focus on cases 1 - 3

Cases 1 - 4, detail of cases 1 - 3
Single replication of simulations
No GM time error
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



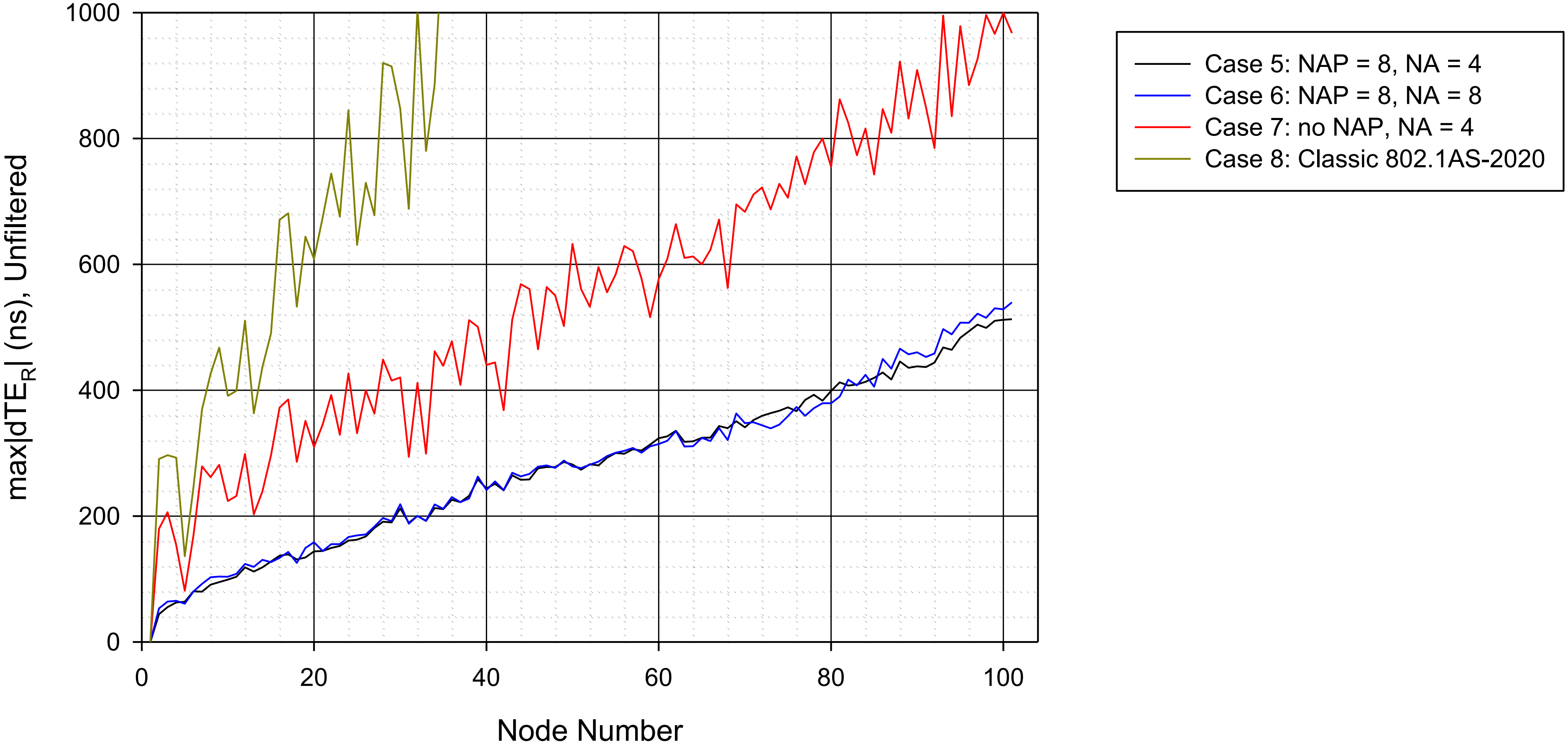
Filtered max|dTE_R|, Cases 5 - 8

Cases 5 - 8
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



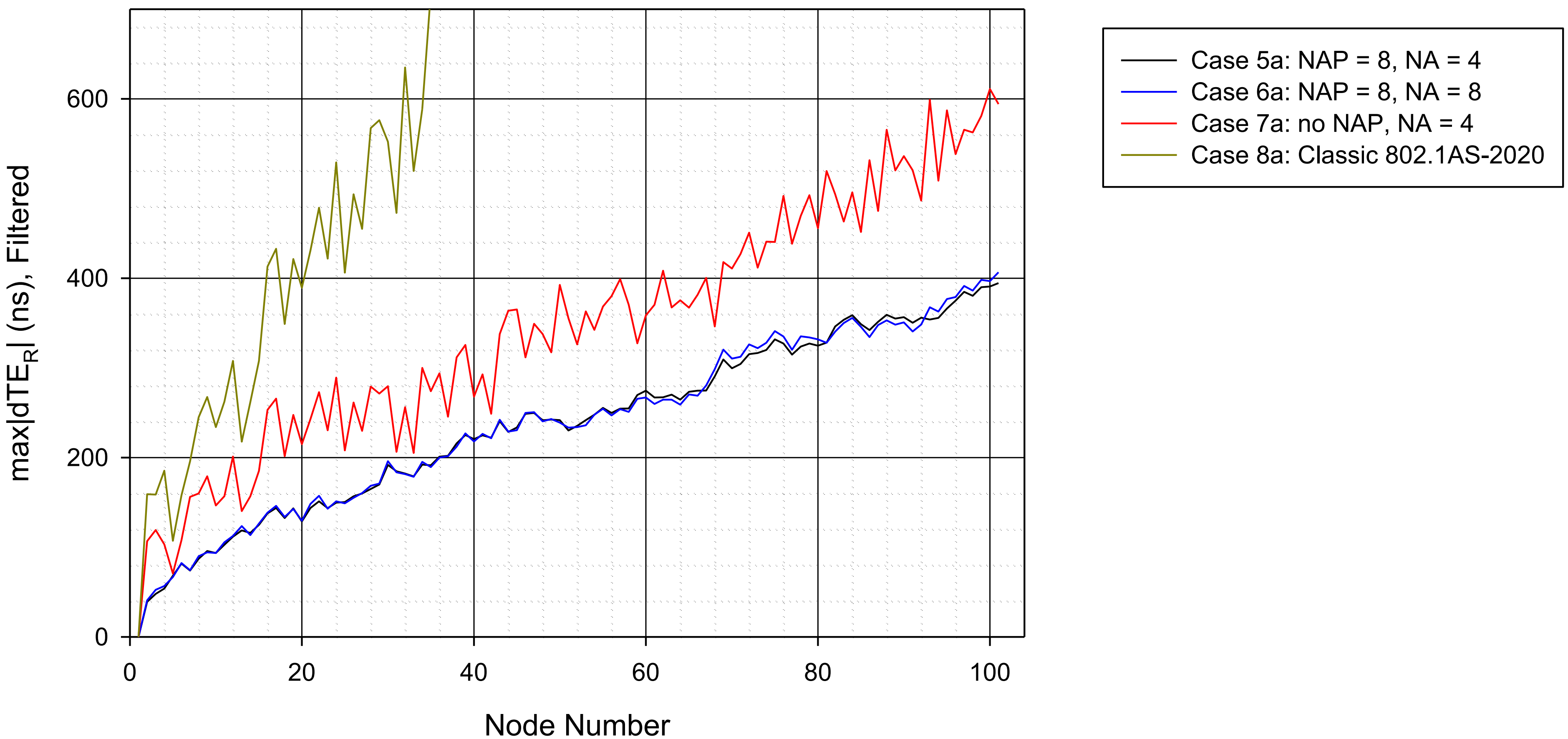
Unfiltered max|dTE$_R$|, Cases 5 - 8

Cases 5 - 8
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



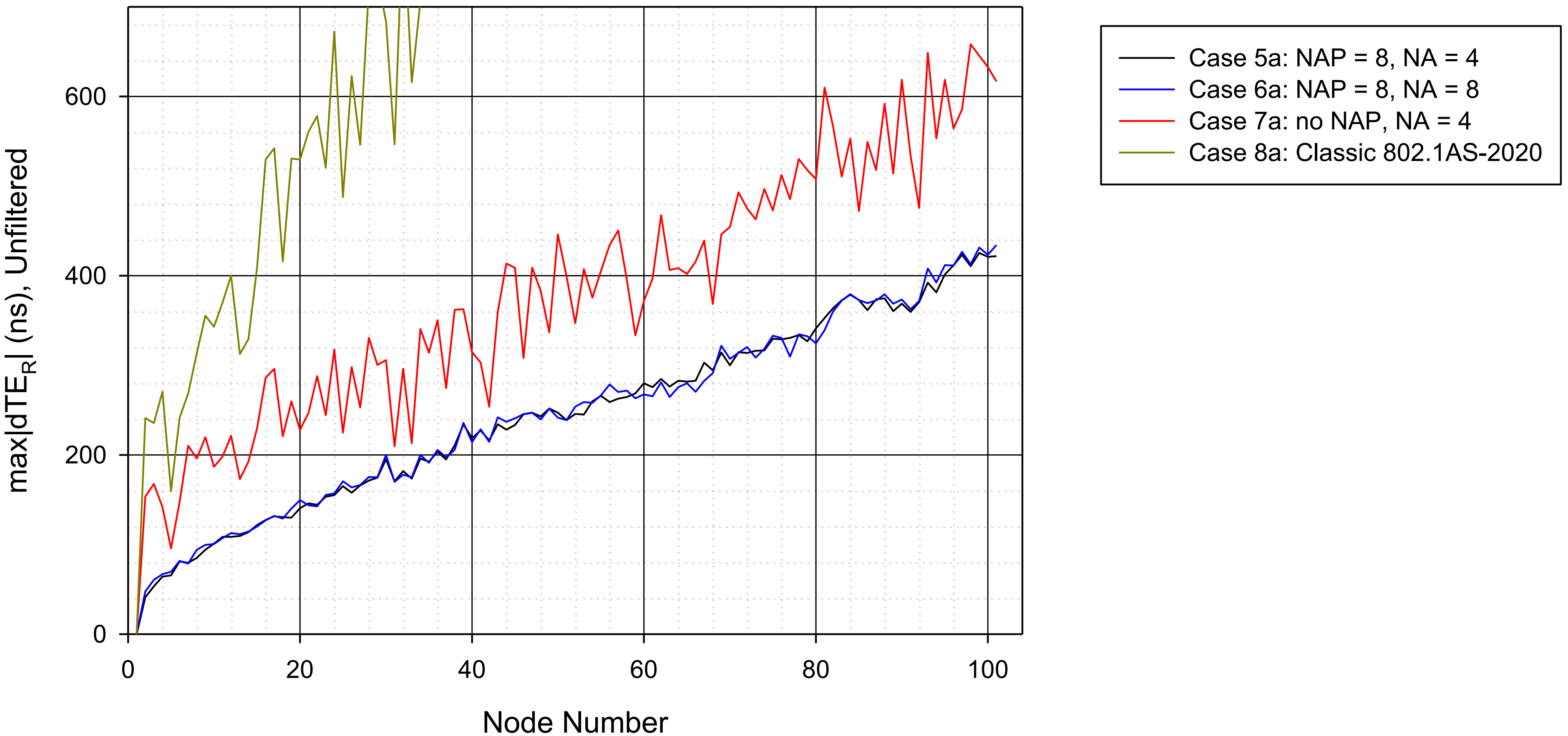
Filtered max|dTE_R|, Cases 5a - 8a

Cases 5a - 8a
Single replication of simulations
GM time error magnitude factor = 0.5
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



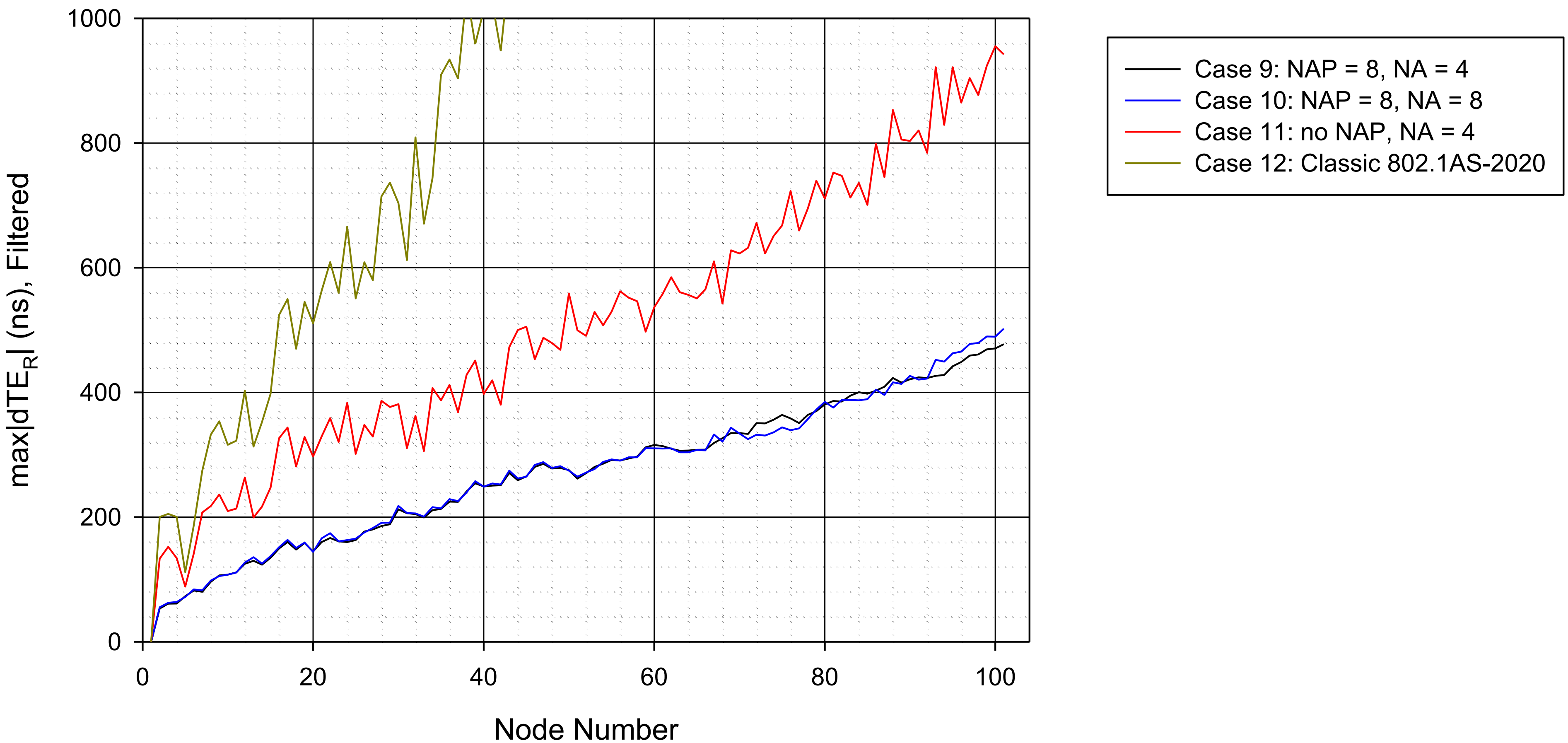
Unfiltered max|dTE_R|, Cases 5a - 8a

Cases 5a - 8a
Single replication of simulations
GM time error magnitude factor = 0.5
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
No noise generation



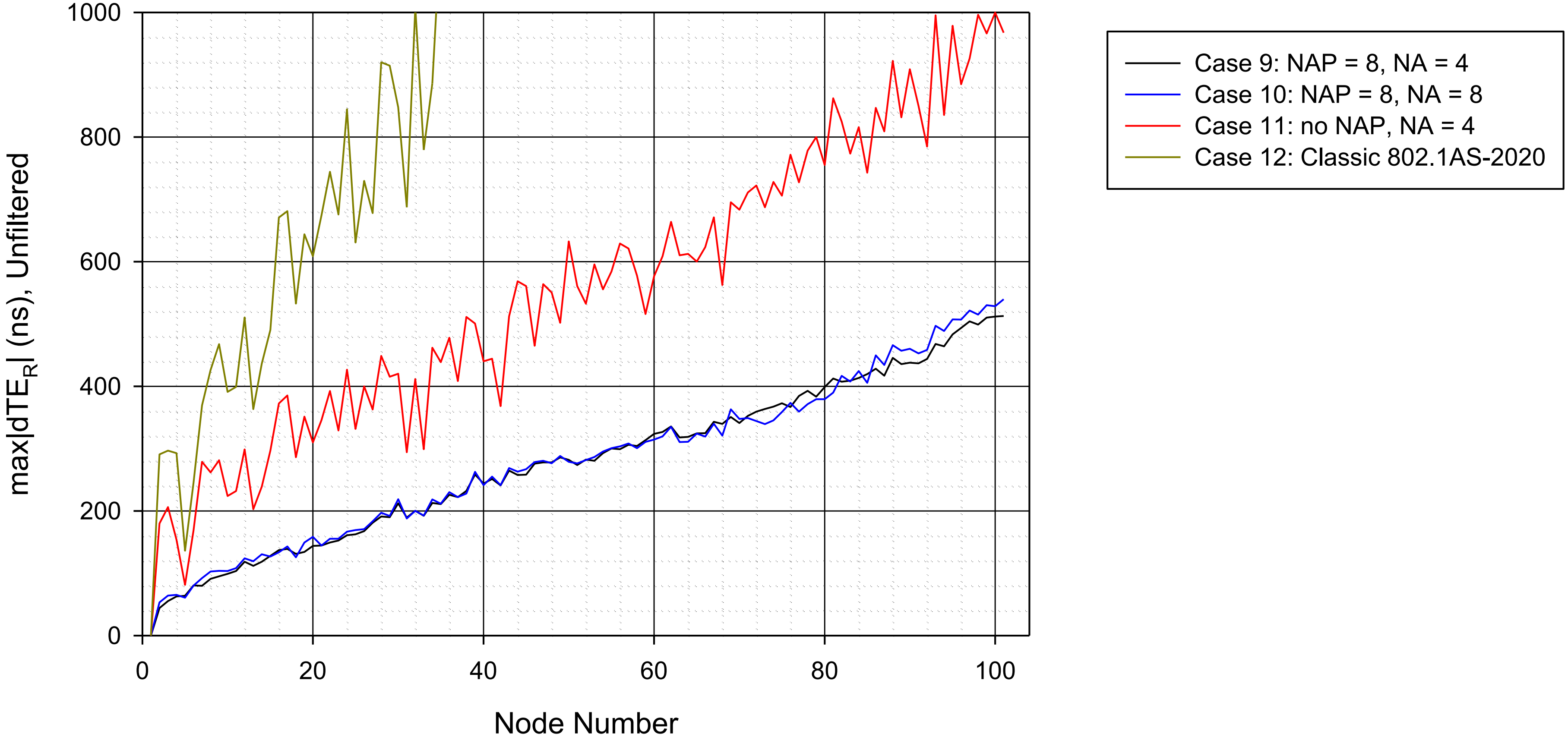
Filtered max|dTE_R|, Cases 9 - 12

Cases 9 - 12
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
With PLL noise generation



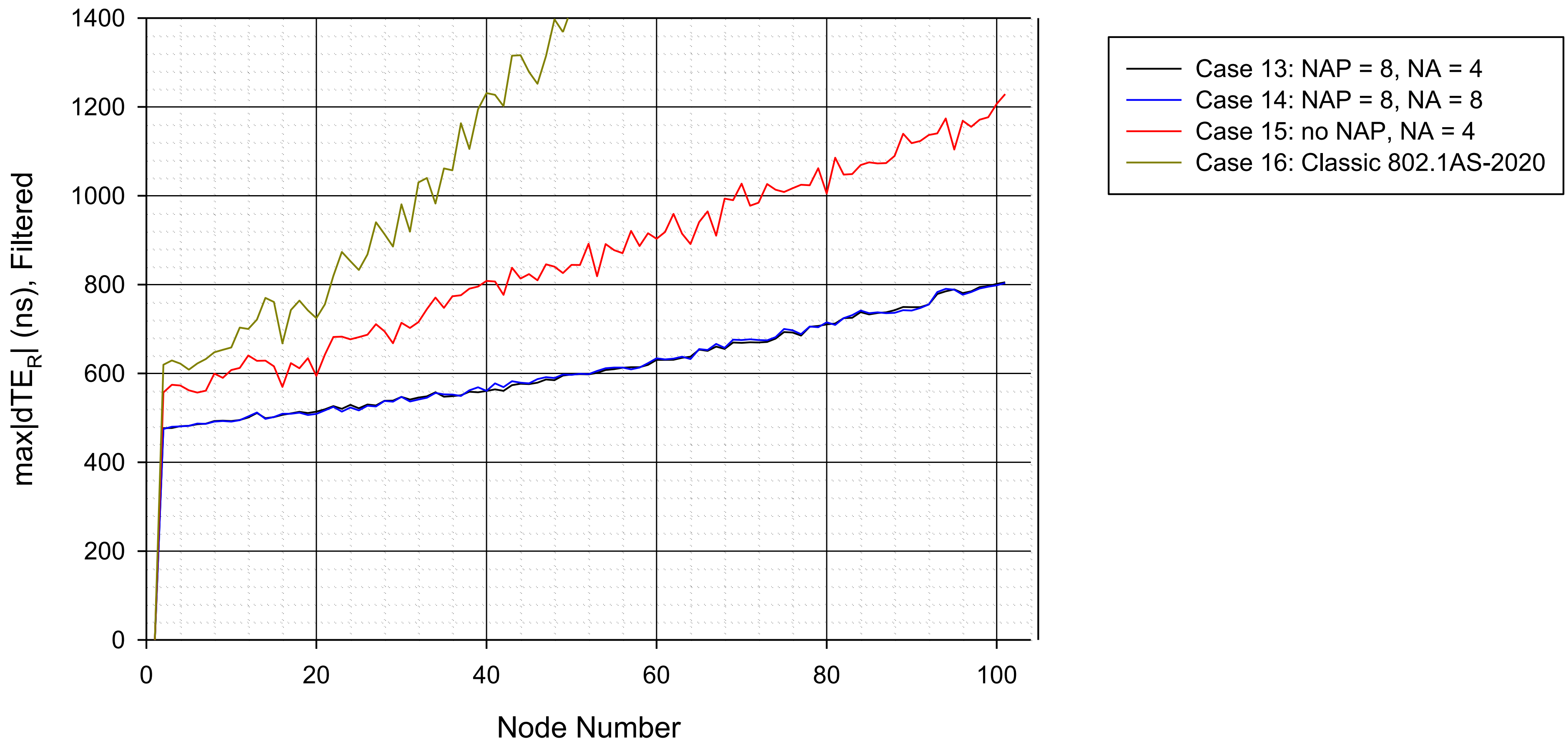
Unfiltered max|dTE_R|, Cases 9 - 12

Cases 9 - 12
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
With PLL noise generation



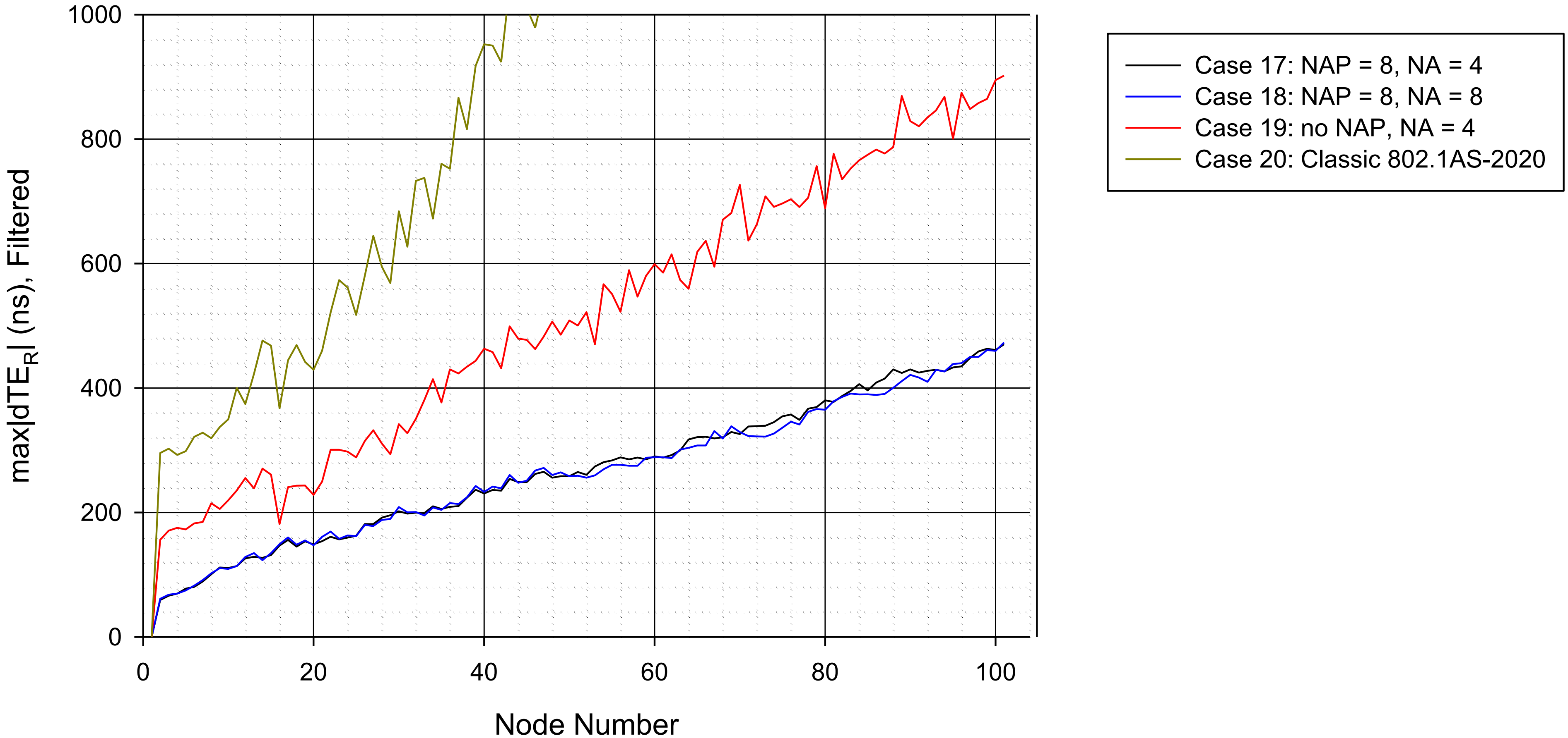
Filtered max|dTE_R|, Cases 13 – 16

Cases 13 - 16
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
f3dB = 0.5 Hz, gain pk = 1.288 dB, zeta = 0.68219
With PLL noise generation

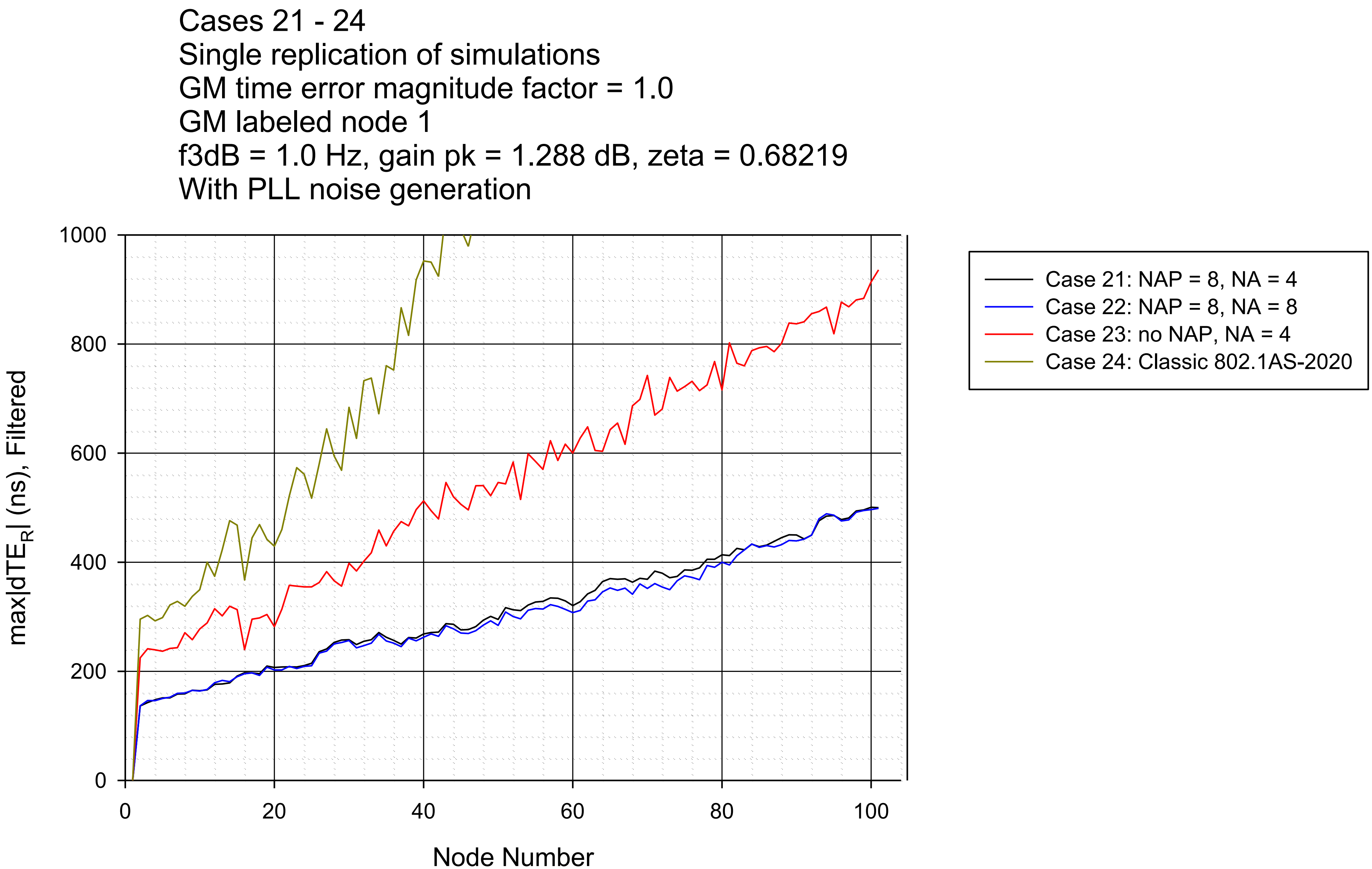


Filtered max|dTE_R|, Cases 17 – 20

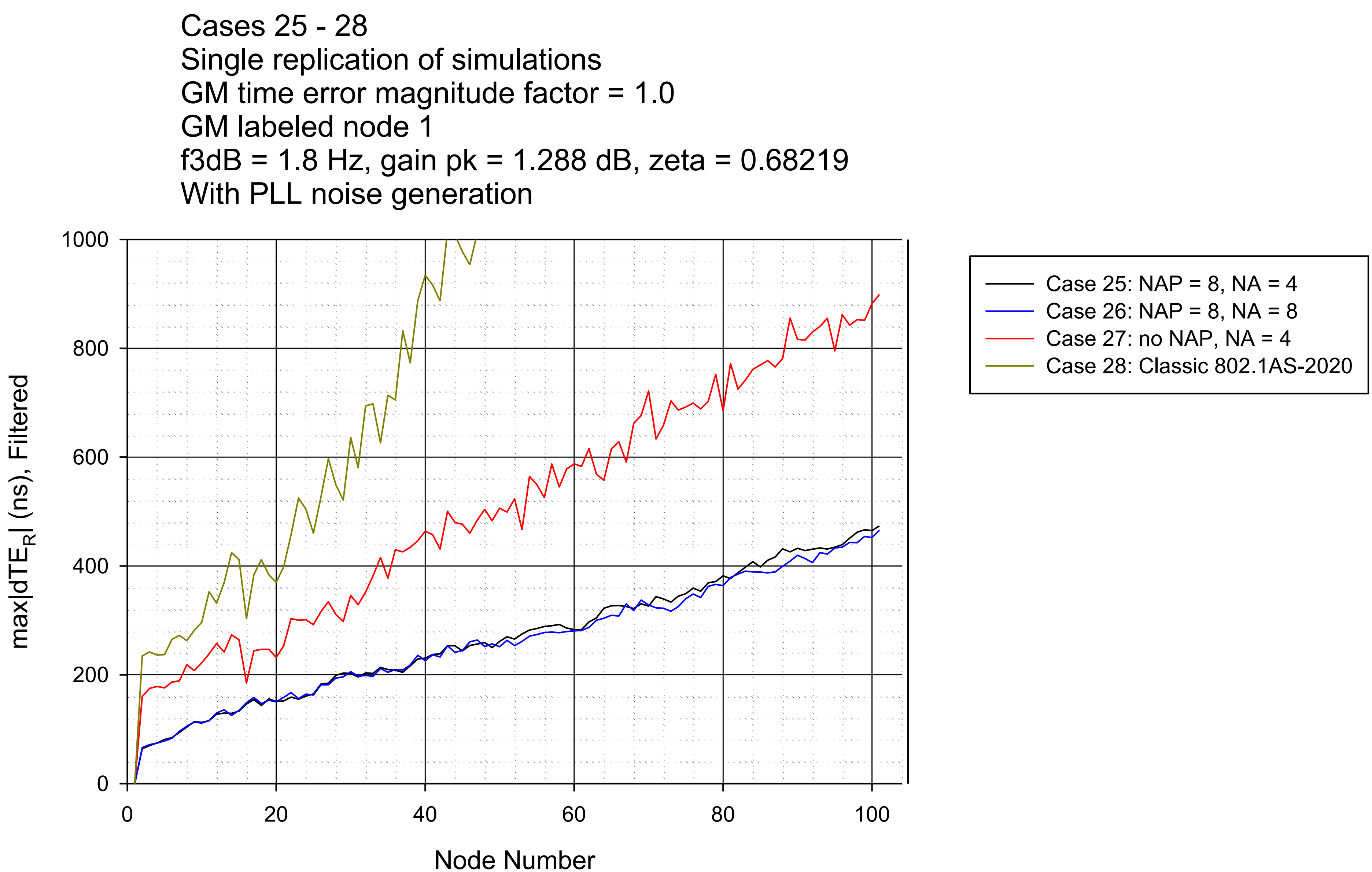
Cases 17 - 20
Single replication of simulations
GM time error magnitude factor = 1.0
GM labeled node 1
f3dB = 2.0 Hz, gain pk = 1.288 dB, zeta = 0.68219
With PLL noise generation



Filtered max|dTE_R|, Cases 21 - 24

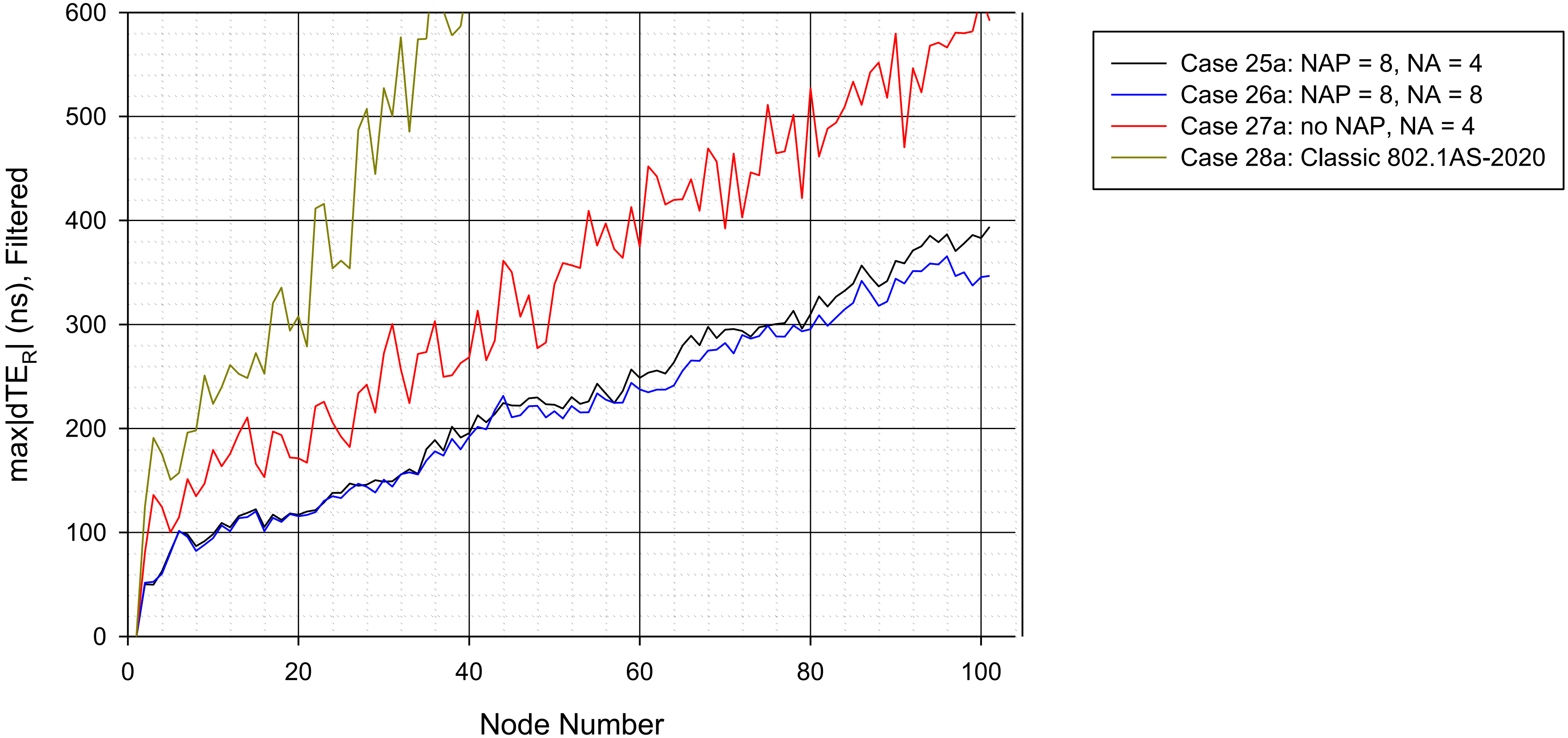


Filtered max|dTE_R|, Cases 25 - 28



Filtered max|dTE_R|, Cases 25a - 28a

Cases 25a - 28a
Single replication of simulations
GM time error magnitude factor = 0.5
GM labeled node 1
f3dB = 1.8 Hz, gain pk = 1.288 dB, zeta = 0.68219
With PLL noise generation



Summary of Filtered max|dTE_R| Results at Nodes 65 and 101

Case	Filtered max|dTE_R| Node 65 (ns)	Filtered max|dTE_R| Node 101 (ns)	Case	Filtered max|dTE_R| Node 65 (ns)	Filtered max|dTE_R| Node 101 (ns)	Case	Filtered max|dTE_R| Node 65 (ns)	Filtered max|dTE_R| Node 101 (ns)
1	278.0	343.5	9	308.0	477.3	21	369.9	499.9
2	275.6	320.9	10	307.7	502.1	22	352.8	498.9
3	318.1	370.1	11	550.7	942.1	23	642.9	935.9
4	1103.1	2046.6	12	1685.4	3134.4	24	1544.7	2677.5
5	308.0	477.3	13	653.5	804.8	25	326.7	473.3
6	307.7	502.1	14	654.6	803.6	26	309.2	465.5
7	550.7	967.4	15	940.5	1228.7	27	615.3	898.9
8	1685.4	3134.4	16	1797.0	2872.5	28	1539.4	2762.7
5a	273.4	394.5	17	320.9	470.2	25a	279.6	393.8
6a	270.2	406.4	18	307.6	473.1	26a	255.4	346.7
7a	367.1	594.2	19	618.8	902.0	27a	420.4	592.1
8a	1374.5	2629.4	20	1544.8	2677.5	28a	1261.9	2333.4

Summary of Unfiltered max|dTE_R| Results at Nodes 65 and 101

Case	Unfiltered max|dTE_R| Node 65 (ns)	Unfiltered max|dTE_R| Node 101 (ns)	Case	Unfiltered max|dTE_R| Node 65 (ns)	Unfiltered max|dTE_R| Node 101 (ns)
1	285.0	403.2	7	600.1	967.4
2	272.4	362.1	8	1910.6	3427.9
3	327.3	426.8	5a	281.7	421.7
4	1278.6	2353.5	6a	279.9	434.2
5	324.5	512.8	7a	401.9	616.6
6	324.2	539.6	8a	1547.2	2881.6

- The unfiltered results for cases 9 – 28 are the same as the results for the respective cases 5 – 8, because the corresponding cases of 5 – 8 and 9 – 28 differ only in that the latter have different endpoint PLL bandwidths and endpoint PLL noise generation
- The unfiltered results for cases 25a – 28a are the same as the results for the respective cases 5a – 8a, because the corresponding cases of 5a – 8a and 9a – 28a differ only in that the latter have different endpoint PLL bandwidths and endpoint PLL noise generation

Discussion of max|dTE_R| Results - 1

- The 500 ns objective for max|dTE_R| is met after 100 hops (i.e., at node 101) for cases 1, 2, 5, 5a, 6a, 9, 10, 17, 18, 25, 26, 25a, and 26a
  - The objective is either just barely met or very slightly exceeded for cases 6, 21, and 22
  - All these cases use nRR drift tracking and compensation and mNRR smoothing
- Comparing cases 1 – 4 with 5 – 8 indicates the effect of GM frequency drift at the same level as at non-GM PTP Instances; comparing with cases 5a – 8a indicates the effect of GM frequency drift at 0.5 of this level. The latter produces max|dTE_R| for cases 5a and 6a in the 400 ns range, which is well within the 500 ns objective.
  - However, note that cases 5a and 6a use a 2.6 Hz bandwidth endpoint PLL and do not model noise generation; as indicated earlier, 2.6 Hz may be too wide for stability for Sync message intervals in the 0.119 – 0.131 s range
    - Cases 9 – 12 show that the effect of noise generation for a 2.6 Hz bandwidth PLL is negligible; the main issue for 2.6 Hz bandwidth is stability

Discussion of max|dTE_R| Results - 2

- Results at node 101 for cases 13 – 16 and 21 – 24 indicate that the effect of noise generation is appreciable for PLL bandwidths of 0.5 Hz and 1 Hz
- Results for cases 25, and 25a are 10-15% better than the results for cases 5 and 5a, respectively. Results for cases 26, and 26a are slightly better than the results for cases 6 and 6a, respectively. This indicates that if the bandwidth is decreased to 1.8 Hz, the performance improvement due to better filtering exceeds the effect of increased noise generation.
  - As indicated earlier, the maximum Sync interval of 0.131 s corresponds to a minimum Sync rate of 1/(0.131 s) = 7.634 Hz. The ratio of this to the 1.8 Hz bandwidth is 4.24. This is less than the 10:1 rule of thumb, but exceeds the theoretical limit of stability for one common PLL implementation.
- It is expected that multiple replications, typically 300, of the simulation cases of most interest will be run
  - Results will, in all likelihood, be larger than the corresponding single-replication results; however, based on previous multiple-replication simulation analyses, the case 5a, 6a, 25a, and 26a results here likely have enough margin relative to the 500 ns objective that multiple-replication results will meet the objective

Conclusion

- The results 5a, 6a, 25a, and 26a indicate that the max|dTE_R| objective of 500 ns over 100 hops can be met with:
  - the drift compensation and tracking, and mNRR smoothing, algorithms described in [1]
  - Endpoint filter 3dB bandwidth and gain peaking of 1.8 Hz and 1.288 dB, respectively
  - Temperature profile, XO frequency stability, and other system parameters as described earlier (with the GM frequency stability equal to one-half the frequency stability at other PTP Instances
- Specifically, max|dTE_R| for cases 25a and 26a are 394 ns and 347 ns, respectively (i.e., there is 106 ns and 153 ns margin, respectively, relative to the 500 ns objective)
- The conclusion must be verified by running multiple replications of the simulation cases

Future Work - 1

- Run single-replication simulation cases with PLL bandwidths between 1 Hz and 1.8 Hz, to determine the bandwidth that gives minimum max$|dTE_R|$
- Address items that were not done in some or all of the current simulations:
  - Make sure that the GM frequency stability is one-half the frequency stability at other PTP Instances in all future simulations
  - Add 4 ns to the error due to timestamp granularity at the GM, to properly model the timestamp granularity
  - In future simulations, produce results for in-sync/out-of-sync behavior (in multiple replication simulations, it may not be practical to show the results for all 300 replications, but instead only for a few selected replications)
  - For message egress at GM and ingress at node 1, record distribution of amount of truncation for each timestamp granularity error, and produce probability distribution histograms with bin size 0.1 ns for egress at GM and ingress at node 1

Future Work - 2

- To consider PLL stability, it would be of interest to examine examples of PLL implementations used in deployed systems in terms of their bandwidth and gain peaking, type of oscillator used, and the Sync message rates used
  - Block diagrams would be useful, if it is possible to make them available

Thank you

References - 1

[1] David McCall, *60802 Time Sync – Monte Carlo and Time Series Simulation Configuration Including NRR and RR Drift Tracking & Error Compensation*, Version 3, IEC/IEEE 60802 presentation, June 2023, (available at https://www.ieee802.org/1/files/public/docs2023/60802-McCall-Time-Sync-Simulation-Configuration-NRR-RR-Algorithms-0623-v03.pdf)

[2] David McCall, *60802 Time Sync – Monte Carlo Simulations with RR & NRR Drift Tracking and Compensation – Results Part 2*, Version 1, IEC/IEEE 60802 presentation, July 2023, (available at https://www.ieee802.org/1/files/public/docs2023/60802-McCall-Monte-Carlo-Sim-with-RR-NRR-Drift-Tracking-Comp-Results-Part-2-0723-v02.pdf)

[3] David McCall, *IEC/IEEE 60802 Contribution – Time Sync Informative Annex, Version 3*, IEC/IEEE 60802 presentation, August 2023 (available at https://www.ieee802.org/1/files/public/docs2023/60802-McCall-Time-Sync-Informative-Annex-0823-v03.pdf)

References - 1

[4] Geoffrey M. Garner, New Simulation results for Base Case and Case 1 of the Time Sync Breakout Held during the IEC/IEEE 60802 Ad Hoc Session, Revision 2, IEC/IEEE 60802 presentation, July 2022 (available at https://www.ieee802.org/1/files/public/docs2022/60802-garner-new-simul-results-base-case-case-1-of-June2022-ad-hoc-0722-v02.pdf)

[5] 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 call (available at https://www.ieee802.org/1/files/public/docs2021/60802-garner-temp-freqoffset-plots-based-on-new-freq-stabil-data-0321-v00.pdf)

[6] Chris McCormick, *Crystal Fundamentals & State of the Industry*, IEC/IEEE 60802 presentation, February 22, 2021 call (available at https://www.ieee802.org/1/files/public/docs2021/60802-McCormick-Osc-Stability-0221-v01.pdf)

[7] ITU-T Rec. G.8251, *The control of jitter and wander within the optical transport network*, ITU-T, Geneva, November 2022

References - 2

[8] ITU-T Series G Supplement 65, *Simulations of transport of time over packet networks*, ITU-T, Geneva, October 2018

[9] John Rogers, Calvin Plett, Foster Dai, *Integrated Circuit Design for High-Speed Frequency Synthesis*, Artech House, 2006.