New 60802 dTE_R Time Series Simulation Results for Comparison with Monte Carlo Simulation Results

Geoffrey M. Garner
Huawei (Consultant)

gmgarner@alum.mit.edu

IEEE 802.1 TSN TG
2022.03.10

Introduction - 1

- References [1], and previous presentations in the series (by the same author), present simulation results for max|dTE$_R$| (i.e., maximum absolute value of dynamic time error relative to the grandmaster (GM)) using a Monte Carlo approach that does not involve time-domain simulation
  - The model is approximate, but has the advantage of running several orders of magnitude faster than time domain simulations using probabilistic models
- In the discussion of [1] and preceding presentations, it was decided to compare corresponding results obtained using the Monte Carlo simulator and the time-domain simulator
- The specific simulation cases are summarized in slide 6 of [1], which is reproduced on the next slide here for convenience.
- The current presentation provides 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 are based on 300 multiple, independent replications for each case

Introduction - 2

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

		Errors			Parameter			Correction Factors	
Case	Reason	**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)	Timestamp Granularity (ns)	Dynamic Timestamp Error (±ns)	pDelay Interval (ms)	Residence Time (ms)	pDelay Turnaround Time (ms)	Mean Link Delay Averaging	mNRR Smoothing Factor N
A	Baseline with previous assumptions	Ramp Rate 1°C / s (Cycle of 310 s)	8	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 - 2

- The time domain simulator computes the time history of dTE at each node (PTP Instance) in a chain consisting of a GM, followed by 100 PTP Instances (nodes)
- Both unfiltered and filtered (by a PLL with specified parameters as given in the following slides) are computed at each node
- dTE_R relative to the GM (i.e., the first node) is computed by interpolating the time histories at each node to a common set of times, and then computing the difference at corresponding times
- Details of the simulation model are given in [2], [3], and [4], and references cited in those presentations.
- Some of the details, and the assumptions, are given on the following slides (some of which are copied from [4])

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), see 11.2 and Eqs. (11-1) through (11-10) of [7])
- 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 [4] 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 [4]

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 $2^{currentLogPdelayReqInterval}$. 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 $2^{currentLogPdelayReqInterval}$ 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 $0.9 \times 2^{portDS.logMinPdelayReqInterval}$.

- 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 [3] 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 $2^{\text{portDS.logMinPdelayReqInterval}}$

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 in cases A, B, C, D, and E (slide 3), and 0.5 °C /s in case F (slide 3); this takes 125 s or 250 s, respectively
- 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 (cases A, B, C, D, E) and 0.5 °C /s (case F); this takes 125 s or 250 s, respectively
- The temperature then remains at – 40°C for 30 s
- The temperature then increases to +85°C at a rate of 1°C /s or 0.5 °C /s ; this takes 125 s or 250 s, respectively
- The duration of the entire cycle (i.e., the period) is therefore 310 s (cases A – E), or 560 s (case F)

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 [4] here
  - Specifically, the values a_0, a_1, a_2, and a_3 computed in [4] 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 [5] of Reference [4] here
  - This data was provided by the author of [4] of Reference [5] here
- The time variation of frequency offset will be 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 will be obtained by integrating the above frequency versus time waveform
  - The time variation of frequency drift rate at the LocalClock entity will be obtained by differentiating the above frequency versus time waveform

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

- Choose the phase of the LocalClock time error waveform at each node randomly in the range [0,T], at initialization, where T is the period of the phase and frequency variation waveforms (i.e., 310 s or 560 s)

Other Assumptions - 1

- Additional assumptions
  - Mean Sync interval: 125 ms
  - Mean Pdelay interval: 31.25 ms (cases A, D, E); 1000 ms (cases B, F); 250 ms (case C)
  - Timestamp granularity: 8 ns (modeled by truncating to next lower multiple of 8 ns)
  - Residence time: 1 ms (case A); 10 ms (cases B – F)
  - Pdelay turnaround time is the same as the residence time for the respective case
  - Dynamic timestamp error is taken to have a uniform distribution over $\pm e$, where e is 8 ns (case A), 4 ns (cases B, C, D, F), and 2 ns (case E)
- Other assumptions are taken from slide 3 above or, if not indicated on slide 3, then from [4], and are summarized on the following slides
- Note that only cases A, B, D, E, F are simulated; case C is not simulated due to insufficient time

Other Assumptions - 2

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|$dTE_{R(k,\,0)}$| (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|$dTE_{R(k,\,0)}$| (i.e., 10 cycles of frequency variation after discard)

Other Assumptions - 3

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

Other Assumptions - 4

- neighborRateRatio is computed using windows of size of 1
  - The difference is taken between respective timestamps of current Pdelay exchange and the previous Pdelay exchange

max|dTE$_R$| Results - 1

- max|TE$_R$| results are for 300 independent replications of simulations for each case
  - max|TE$_R$| is computed for each replication, but after discarding the first 50 s of each replication to remove any startup transient
  - The 300 values are sorted in ascending order
  - A 99% confidence interval for the 0.95 quantile of max|TE$_R$| is given by the interval between the 275th and 294th smallest sample (and the midpoint, i.e., the 285th smallest sample, is taken as a point estimate of the 0.95 quantile)
    - This is because the quantiles of independent samples of a population have a binomial distribution (this result has been used in previous presentations of multiple replication simulation results)
  - The maximum of each set of 300 independent results for each case is also obtained

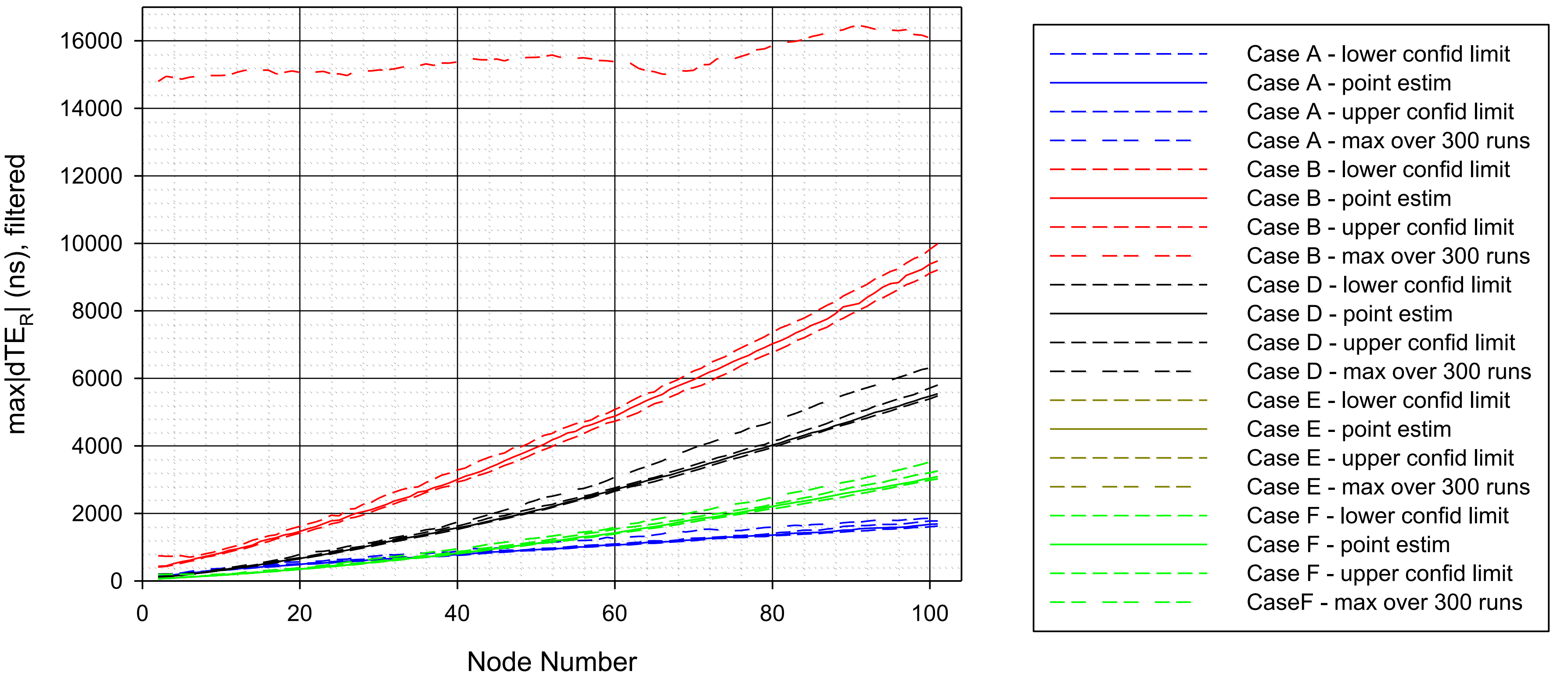
max|dTE_R| Results - 2

- Results for max|dTE_R|, relative to the GM, versus node number are summarized on the next four slides
  - Slide 18: Filtered max|dTE_R|, 99% confidence intervals and maxima over 300 runs, for all cases
  - Slide 19: Filtered max|dTE_R|, only maxima over 300 runs (to reduce clutter)
  - Slide 20: Unfiltered max|dTE_R|, 99% confidence intervals and maxima over 300 runs, for all cases
  - Slide 21: Unfiltered max|dTE_R|, only maxima over 300 runs and only cases A, D, and E (to reduce clutter)

max|dTE_R| Results - 3

Cases A, B, D, E, F - mult replic results - filt
GM time error modeled; dTE_R is relative to GM

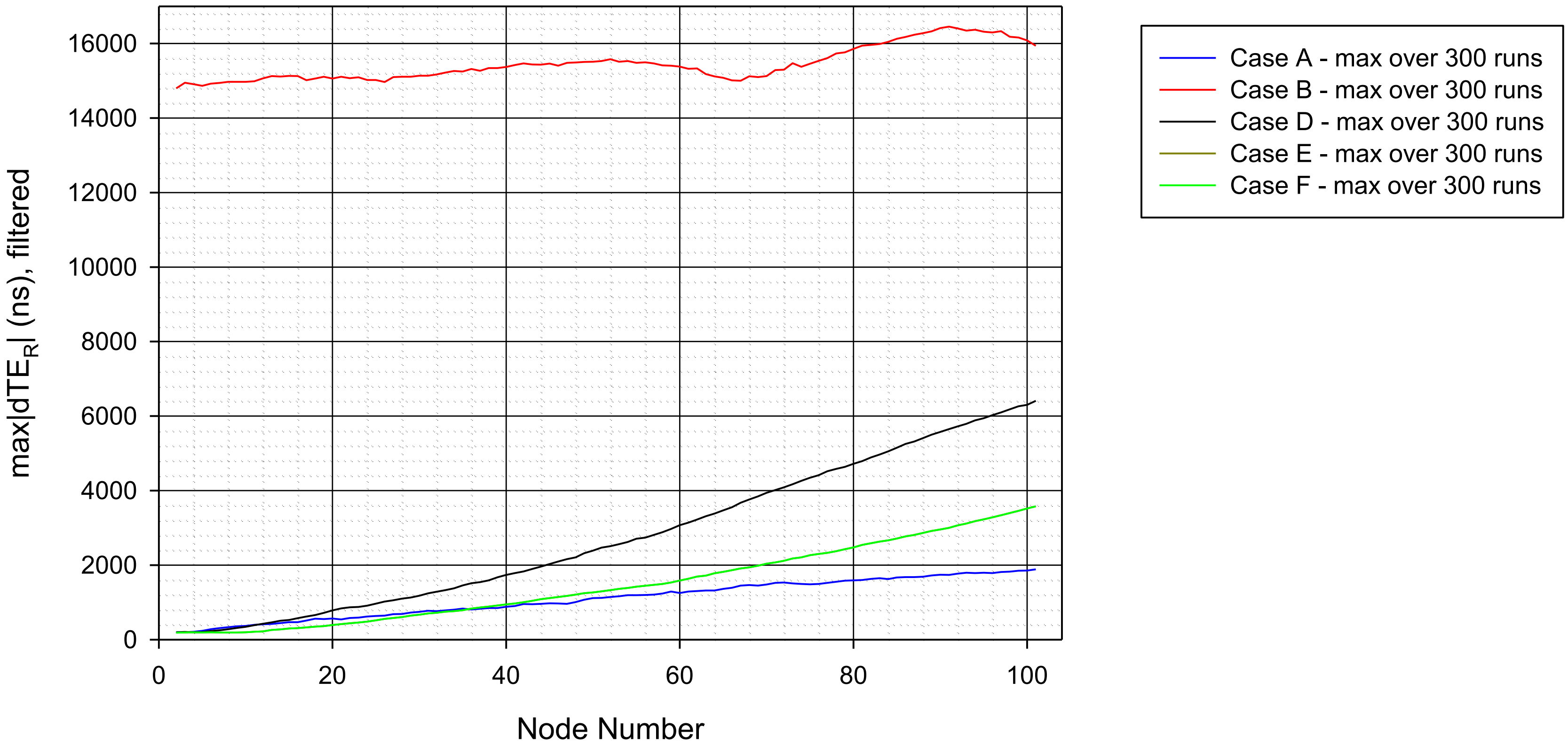
GM labeled node 1
neighborRateRatio measured with window of size 1 (N = 1) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay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



max|dTE_R| Results - 4

Cases A, B, D, E, F - mult replic results - filt
GM time error modeled; dTE_R is relative to GM

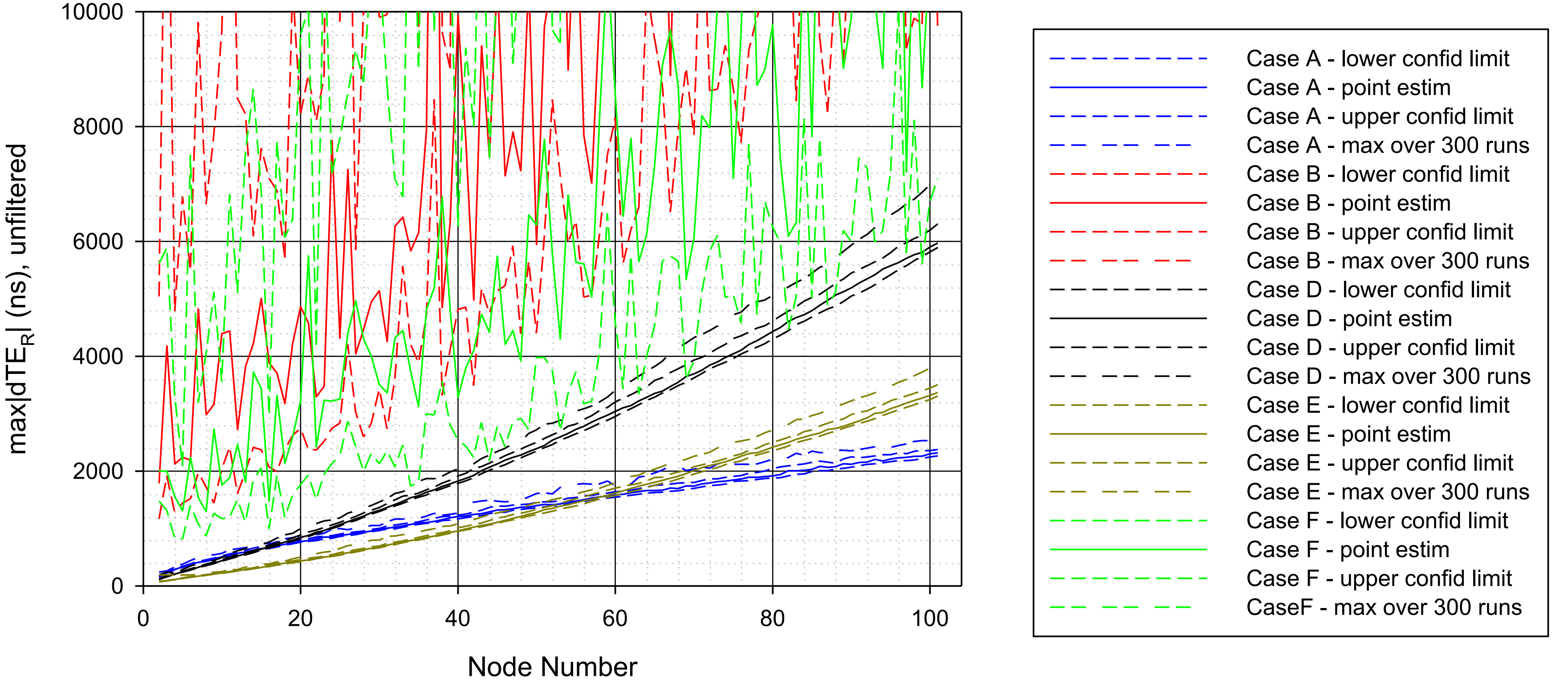
GM labeled node 1
neighborRateRatio measured with window of size 1 (N = 1) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay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



max|dTE$_R$| Results - 5

Cases A, B, D, E, F - mult replication results - unfil
GM time error modeled; dTE$_R$ is relative to GM

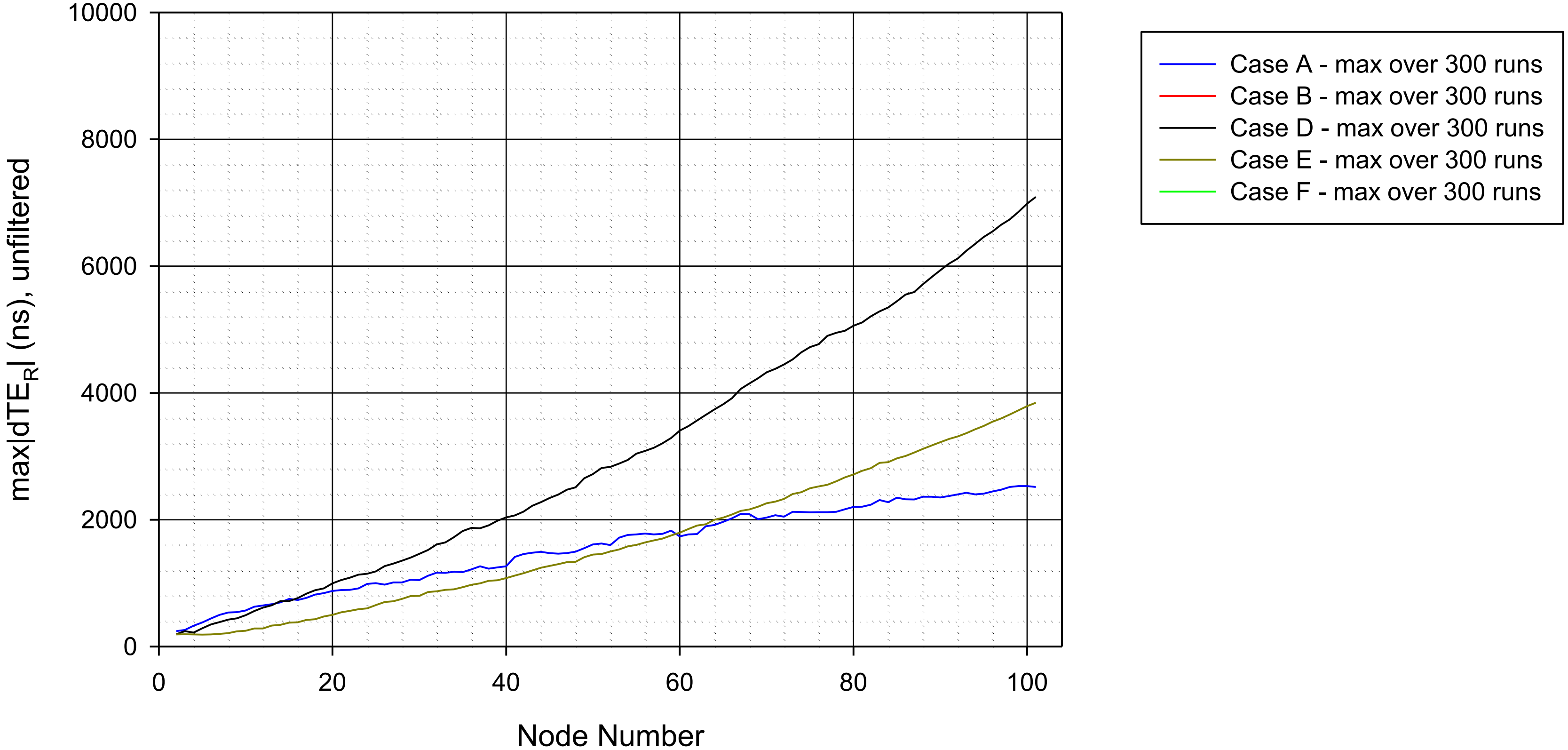
GM labeled node 1
neighborRateRatio measured with window of size 1 (N = 1) and no median calculation
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)
Sync interval variation: +/-10% with 90% probability (Gamma distribution)
Pdelay interval variation: 1.0 to 1.3 of input Pdelay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



max|dTE$_R$| Results - 6

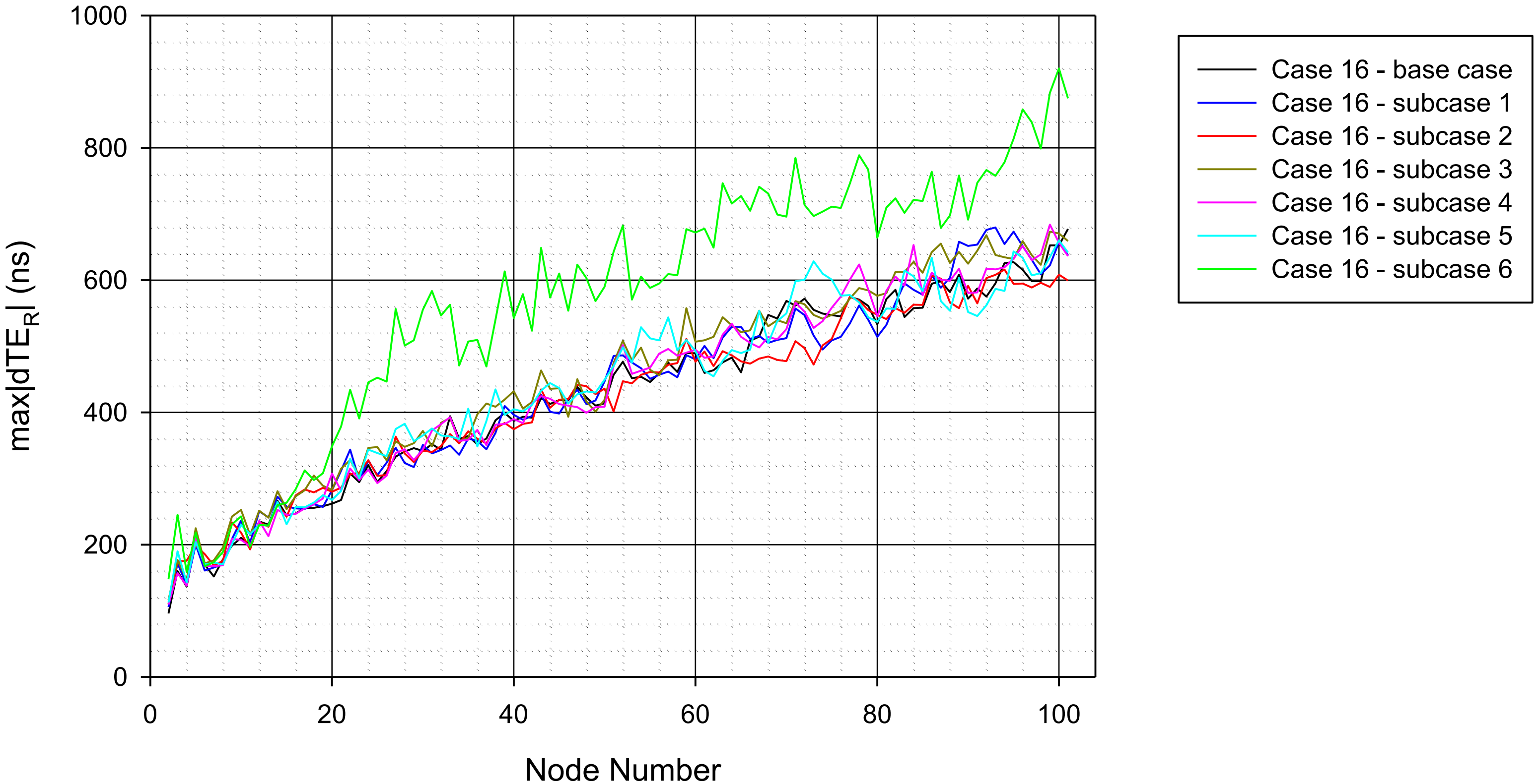
Cases A, B, D, E, F - mult replication results - unfil
GM time error modeled; dTE$_R$ is relative to GM

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



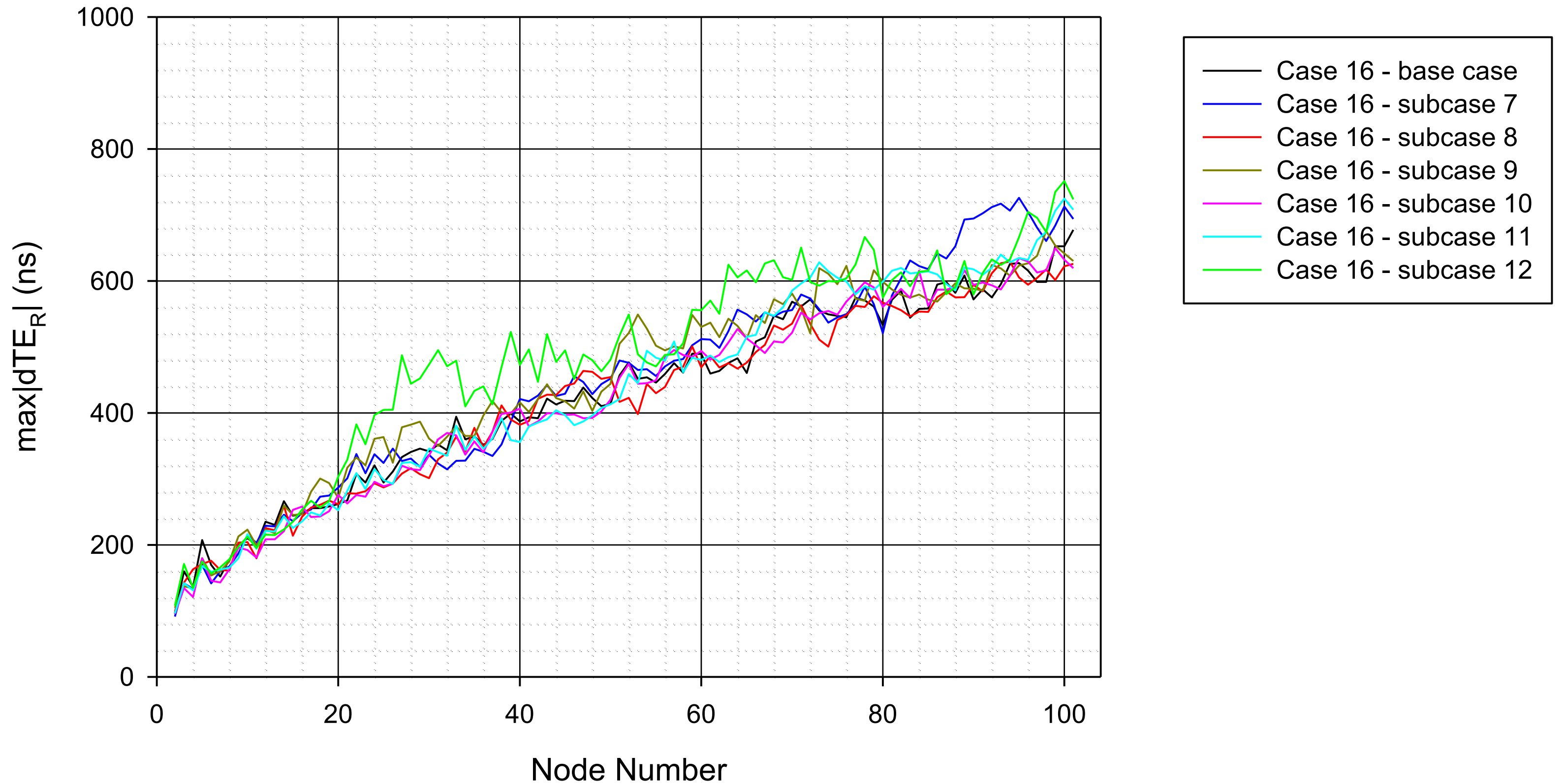
max|dTE$_R$| Results from[4] for comparison - 1

Case 16 - single replication results
Base case: no Sync or Pdelay interval variation
Subcases 1-3: Sync var (+/- 10, 20, 30%)
Subcases 4-6: Sync (+/- 10, 20, 30%) and Pdelay var (0-30%)
GM time error modeled
GM labeled node 1
Clock Model (all clocks): Frequency vs temperature stability and temperature vs time profile from [2]
Accumulate neighborRateRatio, which is measured with window of size 11 and median
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)



max|dTE_R| Results from[4] for comparison - 2

Case 16 - single replication results
Base case: no Sync or Pdelay interval variation
Subcases 7-9: Sync var (+/- 10, 20, 30%)
Subcases 10-12: Sync (+/- 10, 20, 30%) and Pdelay var (0-30%)
GM time error modeled
GM labeled node 1
Clock Model (all clocks): Frequency vs temperature stability and temperature vs time profile from [2]
Accumulate neighborRateRatio, which is measured with window of size 7 (11 for base case) and median
KpKo = 11, KiKo = 65 (f3dB = 2.6 Hz, gain pk = 1.288 dB, zeta = 0.68219)



max|dTE_R| Results from[4] for comparison - 3

Subcase	max|dTE_R|, 64 hops (ns)	max|dTE_R|, 100 hops (ns)
Base case	460	677
1	529	637
2	477	599
3	521	659
4	514	636
5	490	642
6	727	875
7	549	694
8	476	626
9	513	630
10	513	619
11	515	708
12	616	724

64 hops results are for node 65

100 hops results are for node 101

Base case is case 16 of [1], replication 1

Discussion of max|dTE_R| Results - 1

- Results for cases B and F are extremely large, especially the maxima over 300 runs
  - Examination of time history data indicates that the large results are due to isolated spikes in dTE_R in a few of the replications (but PLL filtering reduced the effect)
  - Without this effect, max|dTE_R| would be considerably smaller, though still much larger than the 99% confidence interval for the 0.99 quantile compared to the same difference for cases A, D, and E
  - Note that dTE_R is computed by interpolating the GM and subsequent node time histories to a common set of time samples, and then subtracting the GM dTE from each subsequent node dTE
  - dTE for the GM and each node are monotonic, and over 2500 s or longer can be several orders of magnitude larger than dTE_R (e.g., GM dTE ranges from 0 to approximately -7.5×10^6 ns (= -7.5 ms) over 3200 s
    - Nonetheless, it was checked that the interpolation and subtraction are done correctly
  - The extremely large results are obtained only for the largest values of residence time and Pdelay turnaround time (10 ms) and Pdelay interval (1000 ms), along with the mRR smoothing factor N = 1
    - Previous simulations had larger N (i.e., 7 or 11) and smaller residence and Pdelay turnaround time (1 ms) or smaller Pdelay interval (31.25 ms) (i.e., none of the previous simulation cases had all the parameters set to produce much larger time error
- **In any case, this must be investigated further**

Discussion of max|dTE_R| Results - 2

- Case A is the best case; it is similar to previous simulation cases [4], except that
  - the mRR smoothing factor N is 1 instead of 7 with median calculation
  - The dynamic timestamp error has a uniform distribution over [-8 ns, +8 ns] instead of ±8 ns each with 0.5 probability
- The case A results are on the order of 3 – 4 times the previous results [4]; this is likely due to the smaller value of N
- Cases D and E are larger still
  - The case E results are on the order of 4 – 6.5 times the previous results [4]
  - The case D results are on the order of 8 – 12 times the previous results [4]
  - This is due to the much larger residence time and Pdelay turnaround time compared to the previous simulations (10 ms here versus 1 ms in [4])
    - Note that some of the previous simulation cases of [2] did use 10 ms residence time and had much larger results (12000 – 14000 ns over 100 hops; note that the dynamic timestamp error was ±8 ns each with 0.5 probability, which likely explains the larger results compared to case D here)

Conclusions and Next Steps

- Results for cases A, D, and E show the expected comparison with previous time-domain simulation results
- Unfiltered results for cases A, D, and E can be used for comparison with the corresponding Monte Carlo simulation results (slides 20 and 21 – blue, dark green, and black curves
  - Unfiltered results will be used for comparison because the Monte Carlo simulations [1] do not model endpoint filtering
- The extremely large results for cases B and F will be investigated further to determine whether the effect is real or due to some aspect of the simulation with the parameter values used (e.g., a numerical effect due to all the parameters being set to produce larger dTE_R compared to previous cases
- Case C, and possibly other cases, can be simulated

Thank you

References - 1

[1] 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)

[2] 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)

[3] 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)

References - 1

[4] 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)