New Simulation Results for Time Error Performance for Transport over an IEC/IEEE 60802 Network

Geoffrey M. Garner
Huawei (Consultant)

gmgarner@alum.mit.edu

IEEE 802.1 TSN TG
2020.05.18

Outline

- ❑Introduction
- ❑Assumptions for Simulation Cases
- ❑Results
- ❑Conclusion and Discussion of Next Steps

Introduction - 1

- In the March 2020 IEC/IEEE 60802 virtual meetings, initial simulation results for dynamic time error for transport over an IEC/IEEE 60802 network were presented [1]
- The assumptions for the simulations were based on previous discussion at the January 2019 802.1 meeting [2], and also on detailed discussion of the clock models used in 802.1AS, Annex B and the clock model assumptions for IEC/IEEE 60802 [3]
- The simulation results in [1] indicated that the desired objective of max|dTE| of 1 μs over 64 hops (and over 100 hops if possible) cannot be met using the assumptions for the 60802 local clock (± 100 ppm maximum frequency offset and 3 ppm/s maximum frequency drift rate), accumulation of neighborRateRatio to obtain grandmaster (GM) rateRatio, and other assumptions for the various 802.1AS parameters described in [1] (see slide 29 of [1])
- During the presentation of [1], and in subsequent emails on the 802.1 reflector, modified assumptions for future simulations were discussed
- Subsequently, a number of new simulations using the modified assumptions were run
  - The new results are described in the current presentation
- The simulator and simulation models are not described here; they are described in [1] and the references cited there

Local Clock Noise Generation Model

- The simulations of [1] considered sinusoidal and triangular wave noise generation models that were discussed previously in IEC/IEEE 60802 meetings, as well as a model based on the flicker frequency modulation (FFM) requirement of Annex B/802.1AS; all these models are described in detail in [3]
- It was noted during the presentation of [1] that, for the same maximum frequency offset and drift rate, the triangular wave model is more conservative than the sinusoidal frequency offset model
  - Based on this, it was decided that the next simulations could consider only the triangular wave model local clock noise generation (i.e., local clock stability)
  - It also was decided to focus on, for now, the triangular wave model rather than the FFM model
- However, it was also decided to consider (in the simulations) better frequency offsets and drift rates in addition to ± 100 ppm and 3 ppm/s, i.e.
  - ± 100 ppm and ± 50 ppm
  - 3 ppm/s, 1 ppm/s, 0.3 ppm/s, 0.1 ppm/s

Review of Assumptions for HRM – 1

- These assumptions on the HRM are common to all simulation cases
- The HRM is a linear chain that consists of 100 PTP Instances, and therefore with 99 PTP links connecting each successive pair of PTP Instance
  - The first PTP Instance in the chain is the Grandmaster PTP Instance
  - The next 98 PTP Instances are PTP Relay Instances
  - The last PTP Instance is a PTP End Instance
  - The PTP End Instance contains an endpoint filter, through which the transported time is computed

Assumptions for HRM – 2

- The GM and each PTP Relay Instance do not filter the timestamps with an endpoint filter when computing the value of the originTimestamp and correctionField of each transmitted Sync message
  - Rather, these fields are computed using the same fields of the most recently received Sync message, the <syncEventIngressTimestamp> of the most recently received Sync message, the <syncEventEgressTimestamp of the Sync message being transmitted, and the current value of rateRatio (i.e., cumulative rateRatio)
- However, the information at each PTP Relay Instance is used to separately compute a filtered (recovered) time, which could be used, e.g., by a co-located end application
  - This is equivalent to having a PTP End Instance collocated with the PTP Relay Instance

Assumptions for Grandmaster – 1

- In [1], the Grandmaster (GM) was assumed to be perfect
  - Both the GM noise generation (i.e., time error of the source of time) and the GM timestamping error were taken to be zero
- This was equivalent to computing dTE relative to the GM output
  - With this approach, the time error of the GM could be considered as a separate budget component (i.e., separate from dTE), to be added later (similar to other budget components, e.g., cTE)
  - Alternatively, the time error of the GM would not be added if this was not considered to be relevant to the application
- In the discussion during the presentation of [1] and in subsequent emails, it was stated that, while it is $dTE_{R(k,0)}$ (i.e., relative time error at node *k* relative to the GM) that is important, the effect of the GM phase/time variation on the downstream recovered time should be considered
  - The actual GM noise generation (i.e., not including the effect of timestamp granularity at the GM egress) can be considered to be a triangular wave with $\pm$ 50 ppm maximum frequency offset and maximum frequency drift rate of 3 ppm/s

Assumptions for Grandmaster – 2

- The above means that the GM should be modeled as having noise generation given by a triangular wave with the above characteristics and respective timestamp granularity (the same as for the PTP Relay Instances, i.e., 2 ns or 8 ns)
- The simulator produces absolute time errors (i.e., relative to the reference for the GM), and then time error at each node relative to the GM output is computed
- However, the computation (or measurement) of relative time can be complicated, because the (ideal) times at which the time errors are computed at a node downstream from the GM are, in general, not the same as the times at which time errors at the GM are computed
  - This is mainly due to an approximation made to speed up the simulation run time
  - A major bottleneck for the run time is the writing of output for each node; the simulation timestep is generally much smaller than the time interval for which output is needed. To reduce the amount of output, the output data is divided into blocks, and the largest and smallest value in each block is written.
  - This does not impact the computation of max|dTE| or MTIE because these are peak and peak-to-peak statistics, respectively. This also has negligible effect on TDEV, because the TDEV computation includes averaging and filtering operations.
  - However, relative time error must be computed using samples taken at the same time; if samples at the same time are not available, interpolation is necessary
    - These issues arose in recent simulation work in ITU-T Q13/15 [4]

Assumptions for Grandmaster – 3

- However, if the GM frequency offset is a triangular wave with 50 ppm zero-to-peak amplitude and 3 ppm/s drift rate, the period of the variation is (2)(2)(50 ppm/[3 ppm/s]) = 66.7 s.
- The frequency of the variation is 1/66.7 s = 0.015 Hz
- However, the endpoint filter used in the simulations has bandwidth and gain peaking of 3.78 Hz and 1.049 dB, respectively
  - The effect of this filter on the phase variation due to the GM is therefore very small, and the time error at a downstream node relative to the output of the GM noise source will be approximately the same as the time error if the GM noise source is taken to be zero
- Since the results of [1] show that max|dTE| alone exceeds the desired max|TE| objective of 1 μs by a significant amount, it was decided to omit the effect of the GM noise source for now (it will be included in future simulations after assumptions and parameters that allow the 1 μs objective to be met are decided on)
  - Note that the effect of timestamp granularity at the GM will be included in the simulations

Assumptions Common to All Cases

- These assumptions for the simulations are based on the results of the discussion of [2] at the January 2020 IEC/IEEE 60802 meeting, and were used for the initial simulations of [1]
- Use syncLocked mode (since all ports have same mean Sync interval)
- Residence time: 10 ms
- Pdelay turnaround time (i.e., time between receipt of Pdelay_Req and sending of Pdelay_Resp): 10 ms
- Endpoint filter 3 dB bandwidth and gain peaking: 3.78 Hz, 1.049 dB
  - Equivalent to proportional gain of 20 and integral gain of 80, both normalized to VCO gain of 1 (see [1])
- neighborRateRatio computation granularity: 2.328×10^{-10}

Possible Assumptions Varying from Case to Case - 1

- Mean Sync interval and mean Pdelay interval
  - 32 messages/s (Sync), 32 messages/s (Pdelay)
  - 1 message/s (Sync), 1 message/s (Pdelay)
  - 32 messages/s (Sync), 1 message/s (Pdelay)
- Timestamp granularity
  - 8 ns
  - 2 ns
- Triangular Wave maximum drift rate for noise generation at each PTP Relay Instance and PTP End Instance
  - 3 ppm/s
  - 1 ppm/s
  - 0.3 ppm/s
  - 0.1 ppm/s
- Triangular wave zero-to-peak amplitude for noise generation at each PTP Relay Instance and PTP End Instance
  - 100 ppm
  - 50 ppm

Possible Assumptions Varying from Case to Case - 2

- Computation (measurement) of rateRatio relative to GM
  - Measure neighborRateRatio persistently using Pdelay messages, and accumulate measured neighborRateRatio values in a TLV attached to Sync (or, in two-step case, Follow_Up) to obtain rateRatio
  - Measure rateRatio directly using successive Sync messages (this will be explained in more detail shortly)
- Relative phases of Triangular Wave noise generation waveforms at each PTP Instance
  - Zero (the waveforms at all nodes are in phase)
  - Chosen randomly at initialization
- It is seen that the total number of combinations of all the above assumptions is $3 \times 2 \times 4 \times 2 \times 2 \times 2 = 192$
- This is a very large number of cases; however, it can be reduced by noting the following
- First, the results of [1] indicated that the case where all triangular (or sinusoidal) noise generation waveforms are in phase is the most conservative. Therefore, we initially consider only this case, i.e., we assume the relative phases of the triangular waves is zero

Possible Assumptions Varying from Case to Case - 3

- Second, we will initially assume the amplitude of the triangular wave is 100 ppm
  - In the discussions, the 50 ppm amplitude case was relevant only 1 Gbit/s PHYs
    - 100 ppm was relevant to 10 Mbit/s and 100 Mbit/s PHYs, and for rates above 1 Gbit/s the appropriate frequency accuracy was not known
    - In any case, the results of [1] indicated that the time error depends mostly on the maximum frequency drift rate; this is because frequency offset is measured, and what matters is how much the frequency changes between successive Sync messages
- Third, for cases where GM rateRatio is measured using successive Sync messages, the Pdelay rate is less important
  - Therefore, for these cases we will consider only a mean Pdelay rate of 1 message/s, but with mean Sync rates of 1 message/s and 32 messages/s
  - However, it will be seen that with this technique, error in the rateRatio measurement can increase if the time between Sync messages used to compute rateRatio is small; therefore, for the mean Sync rate of 32 messages/s, we also will consider measuring rateRatio using every 10th Sync message (this will be explained in more detail shortly)
    - But, we still use every Sync message to compute the synchronized time (local PTP clock)

Possible Assumptions Varying from Case to Case - 4

- Fourth, the technique of measuring GM rateRatio using successive Sync messages is being considered mainly as an alternative to improving the oscillator stability
  - Therefore, the initial simulation cases using this technique will be limited to frequency drift rate of 3 ppm/s
- With the above, the number of simulation cases becomes
  - Cases where rateRatio is measured by accumulating neighborRateRatio: $3 \times 2 \times 4 = 24$
  - Cases where rateRatio is measured using successive Sync messages: $(2 \times 2) + 2 = 6$ (the final 2 cases added are for the cases where rateRatio is measured using every 10th Sync message
  - Total number of simulation cases: 30

Computation of GM rateRatio using successive Sync messages

- Assume the computation is done every *n*th Sync message (for simplicity, a jumping window is used)
  - The computation is done on ingress of a Sync message at a PTP Instance
- Let C_{kn} be the correctedMasterTime carried by Sync message kn
- Let S_{kn} be the SyncEventIngressTimestamp for Sync message kn
- Then the computed $\mathrm{rateRatio}$ is

$$\mathrm{rateRatio}_{kn} = \frac{C_{kn} - C_{(k-1)n}}{S_{kn} - S_{(k-1)n}}$$

- For $n = 1$, $\mathrm{rateRatio}$ is computed on receipt of every Sync message
- For $n = 10$, $\mathrm{rateRatio}$ is computed every 10th Sync message, and that value is used on receipt of that Sync message and the next 9 Sync messages
- Note that frequency offset is equal to $\mathrm{rateRatio} - 1$

Simulation Time

- Simulation times for the respective cases will be chosen to include at least several cycles of the triangular waveform
  - The results in [1] indicated that once steady-state is reached, max|dTE| does not increase appreciably after several cycles
- Period (*T*) for each maximum frequency drift rate
  - 3 ppm/s: *T* = (2)(2)(100 ppm/[3 ppm/s]) = 133.3 s
  - 1 ppm/s: *T* = (2)(2)(100 ppm/[1 ppm/s]) = 400 s
  - 0.3 ppm/s: *T* = (2)(2)(100 ppm/[0.33 ppm/s]) = 1333.3 s
  - 0.1 ppm/s: *T* = (2)(2)(100 ppm/[1 ppm/s]) = 4000 s
- In computing max|dTE|, the first 50 s of each simulation time history will be discarded to eliminate any startup transient
- The following simulation times will be chosen for cases that have the respective maximum frequency drift rate
  - 3 ppm/s: 1050 s
  - 1 ppm/s: 2050 s
  - 0.3 ppm/s: 2050 s
  - 0.1 ppm/s: 5050 s

Summary of Simulation Cases – 1

- The simulation cases are summarized in the following tables (the numbering scheme is chosen for convenience of naming directories/folders where the result files are stored; note that the numbering is not always contiguous)
- Parameters not listed have the fixed values given in the preceding slides

Summary of Simulation Cases – 2

Obtain GM rateRatio via accumulation of neighborRateRatio				
Case	Max Freq Drift Rate (ppm/s)	Mean Sync Rate (messages/s)	Mean Pdelay Rate (messages/s)	Timestamp Granularity (ns)
1	3	32	32	8
2	3	32	32	2
3	3	1	1	8
4	3	1	1	2
5	3	32	1	8
6	3	32	1	2
11	0.1	32	32	8
12	0.1	32	32	2
13	0.1	1	1	8
14	0.1	1	1	2
15	0.1	32	1	8
16	0.1	32	1	2

Summary of Simulation Cases – 3

Obtain GM rateRatio via accumulation of neighborRateRatio				
Case	Max Freq Drift Rate (ppm/s)	Mean Sync Rate (messages/s)	Mean Pdelay Rate (messages/s)	Timestamp Granularity (ns)
21	0.3	32	32	8
22	0.3	32	32	2
23	0.3	1	1	8
24	0.3	1	1	2
25	0.3	32	1	8
26	0.3	32	1	2
31	1	32	32	8
32	1	32	32	2
33	1	1	1	8
34	1	1	1	2
35	1	32	1	8
36	1	32	1	2

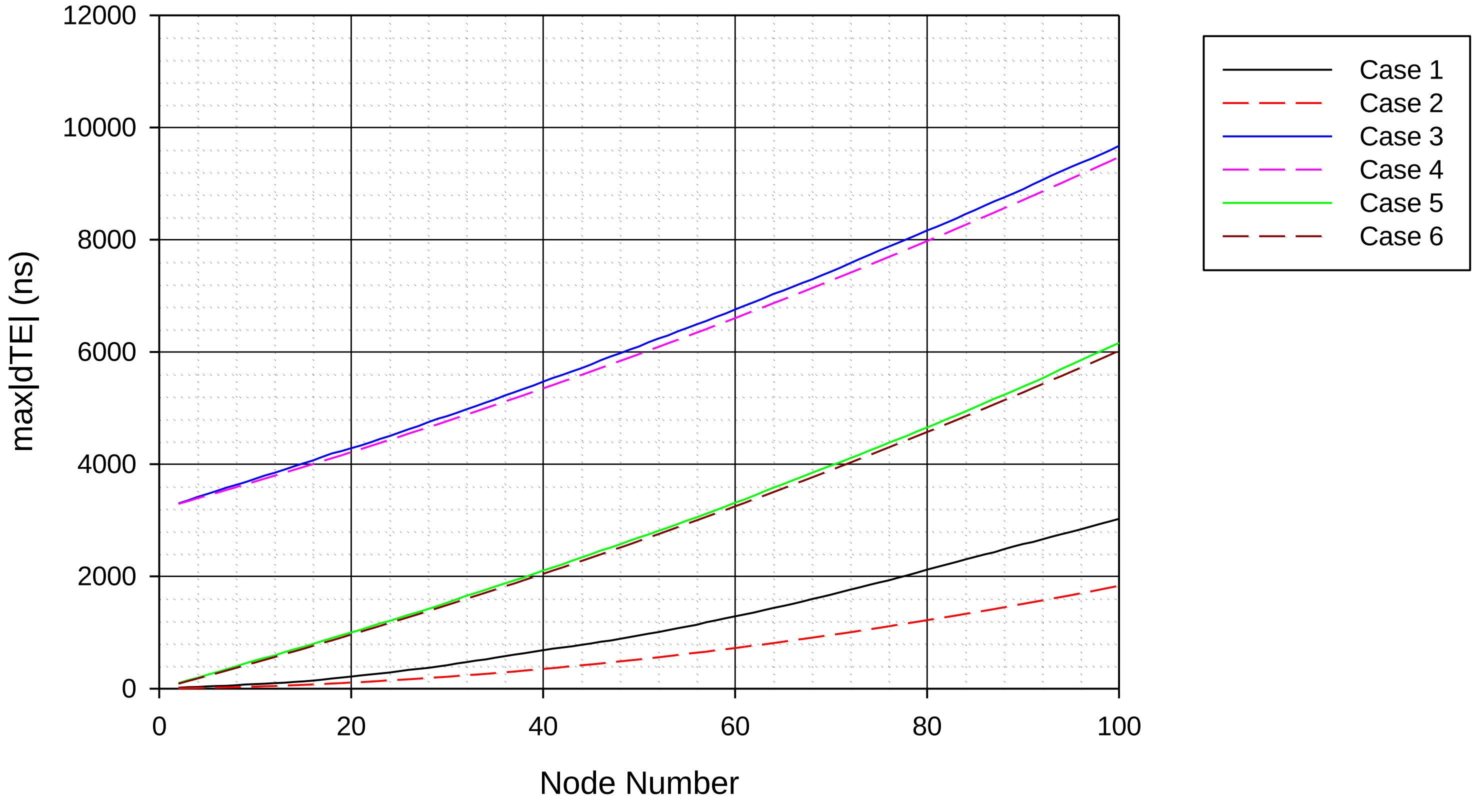
Summary of Simulation Cases - 4

Obtain GM rateRatio using successive Sync messages					
Case	Max Freq Drift Rate (ppm/s)	Mean Sync Rate (messages/s)	Mean Pdelay Rate (messages/s)	Timestamp Granularity (ns)	Use every n^{th} Sync message when computing rateRatio (value of n)
3s	3	1	1	8	1
4s	3	1	1	2	1
5s	3	32	1	8	1
6s	3	32	1	2	1
5s10	3	32	1	8	10
6s10	3	32	1	2	10

Note: In cases 5s10 and 6s10, we still use every Sync message to compute the synchronized time; we compute rateRatio every 10^{th} Sync message

Cases 1 – 6: max|dTE| - 1

Simulation Cases 1 - 6
Single replication of simulation
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 3 ppm/s maximum drift rate
Cases 1 and 2: 32 Sync msgs/s, 32 Pdelay exchanges/s
Cases 3 and 4: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 5 and 6: 32 Sync msgs/s, 1 Pdelay exchange/s

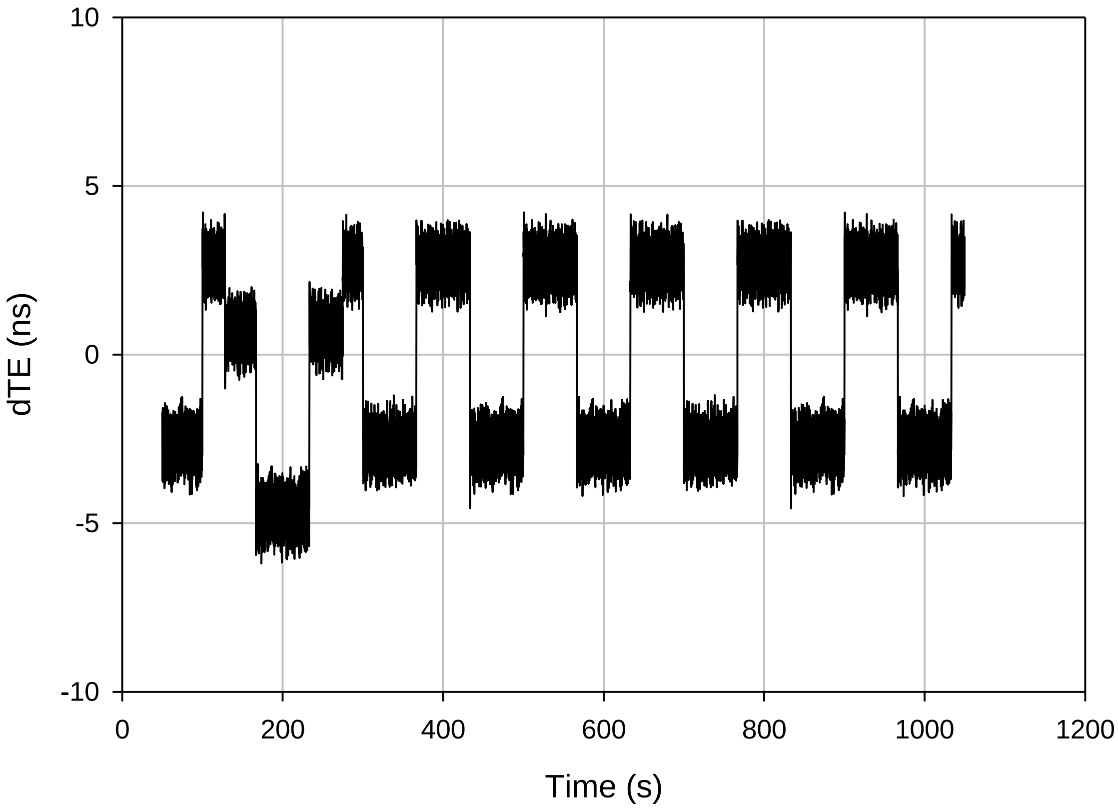


Cases 1 – 6: max|dTE| - 2

- The best performance is for case 2 (2 ns timestamp granularity, 32 messages/s for both Sync and Pdelay)
  - For this case, max|dTE| reaches 1000 ns (1 μs) after approximately 72 hops
  - max|dTE| is 811 ns (0.81 μs) after 64 hops, and 1830 ns (1.8 μs) after 100 hops
  - While max|dTE| is within the 1 μs limit after 64 hops for this case, the margin left for cTE and other budget components is 190 ns, which is very likely insufficient for 64 hops
- For case 1, max|dTE| just exceeds 1000 ns after 52 hops, and is 1437 ns after 64 hops and 3026 ns after 100 hops
- Note that max|dTE| for case 5 (mean Sync rate of 32 messages/s, mean Pdelay rate of 1 message/s, 8 ns timestamp granularity) is approximately 3.6 μs after 64 hops and 6.2 μ s after 100 hops, consistent with case 4 of [1]
- Max|dTE| for cases 3 and 4 (mean Sync and Pdelay rates of 1 message/s) ranges from 3.3 to 9.7 μs as the number of hops increases from 2 to 100
  - For these cases, max|dTE| is relatively insensitive to timestamp granularity

dTE Time History – Case 2, node 2

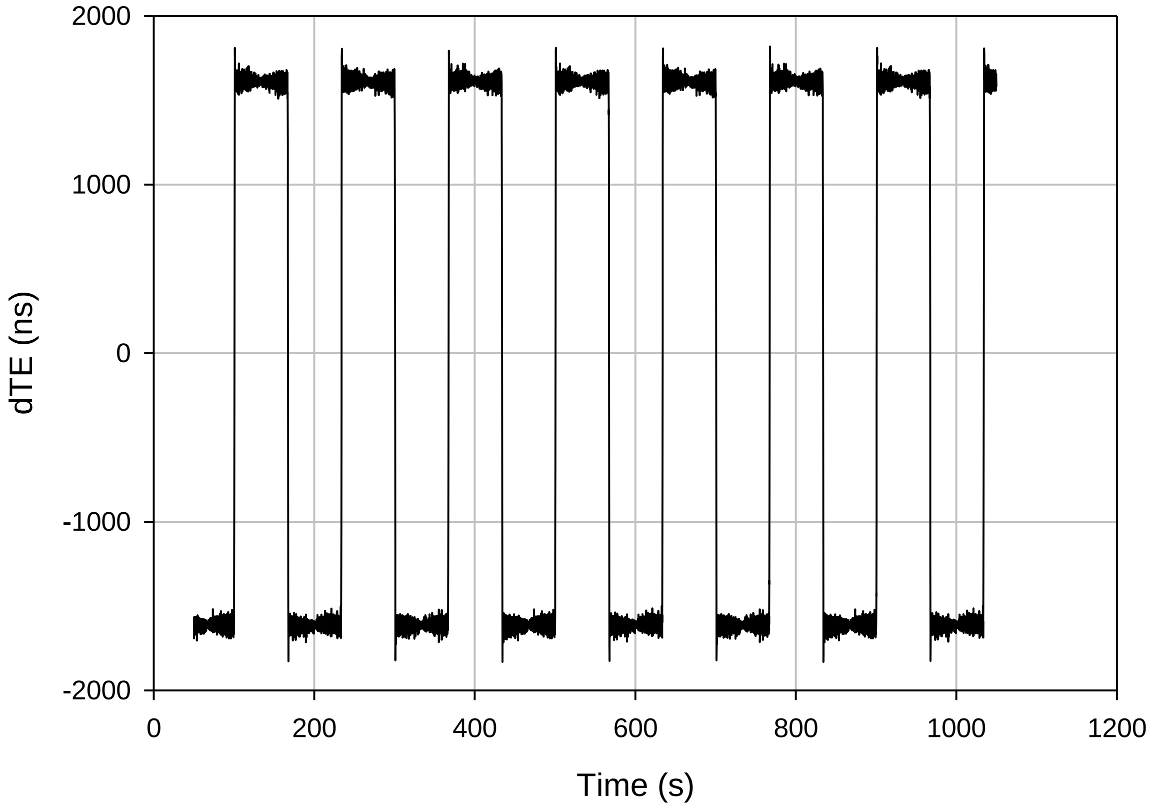
Case 2, PTP Instance (node) 2
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)
First 50 s removed, to eliminate any startup transient



- Case 2, node 2 dTE time history is qualitatively similar to result obtained for case 4, node 2 in [1]

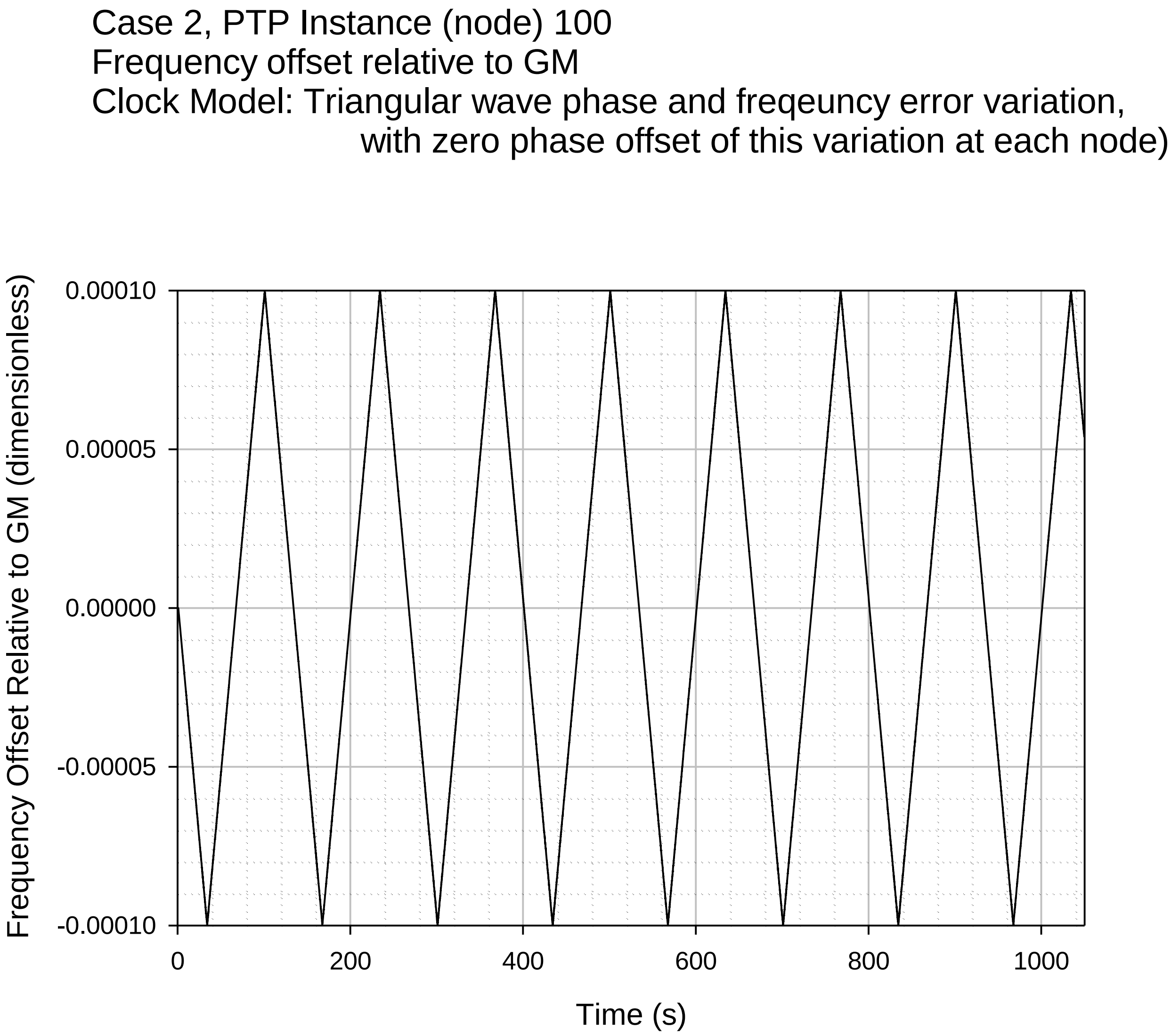
dTE Time History - Case 2, node 100

Case 2, PTP Instance (node) 100
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)
First 50 s removed, to eliminate any startup transient

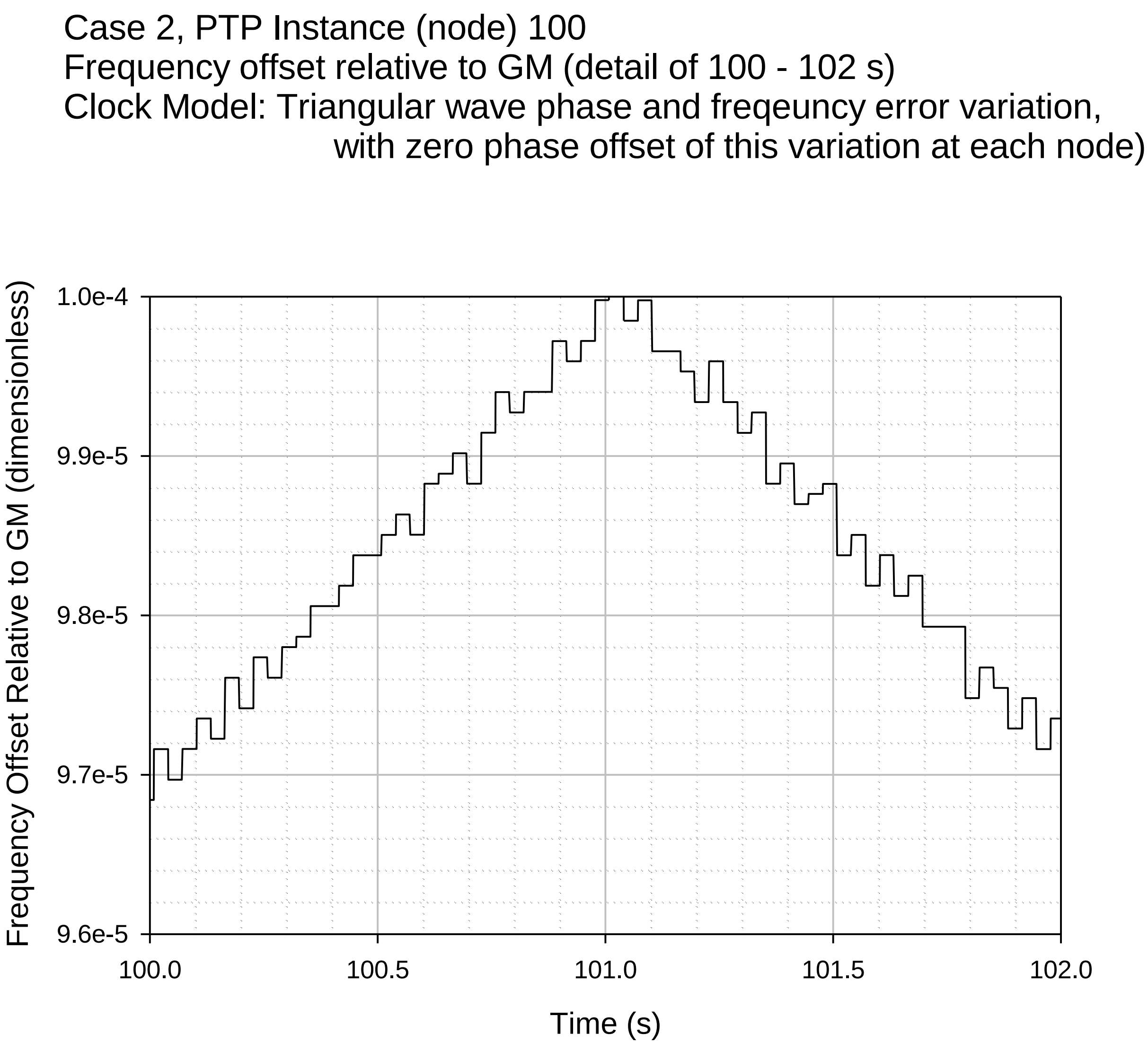


- Case 2, node 100 dTE time history is qualitatively similar to result obtained for case 4, node 2 in [1]

Case 2, node 100 Frequency Offset Relative to GM



Case 2, node 100 Frequency Offset Relative to GM



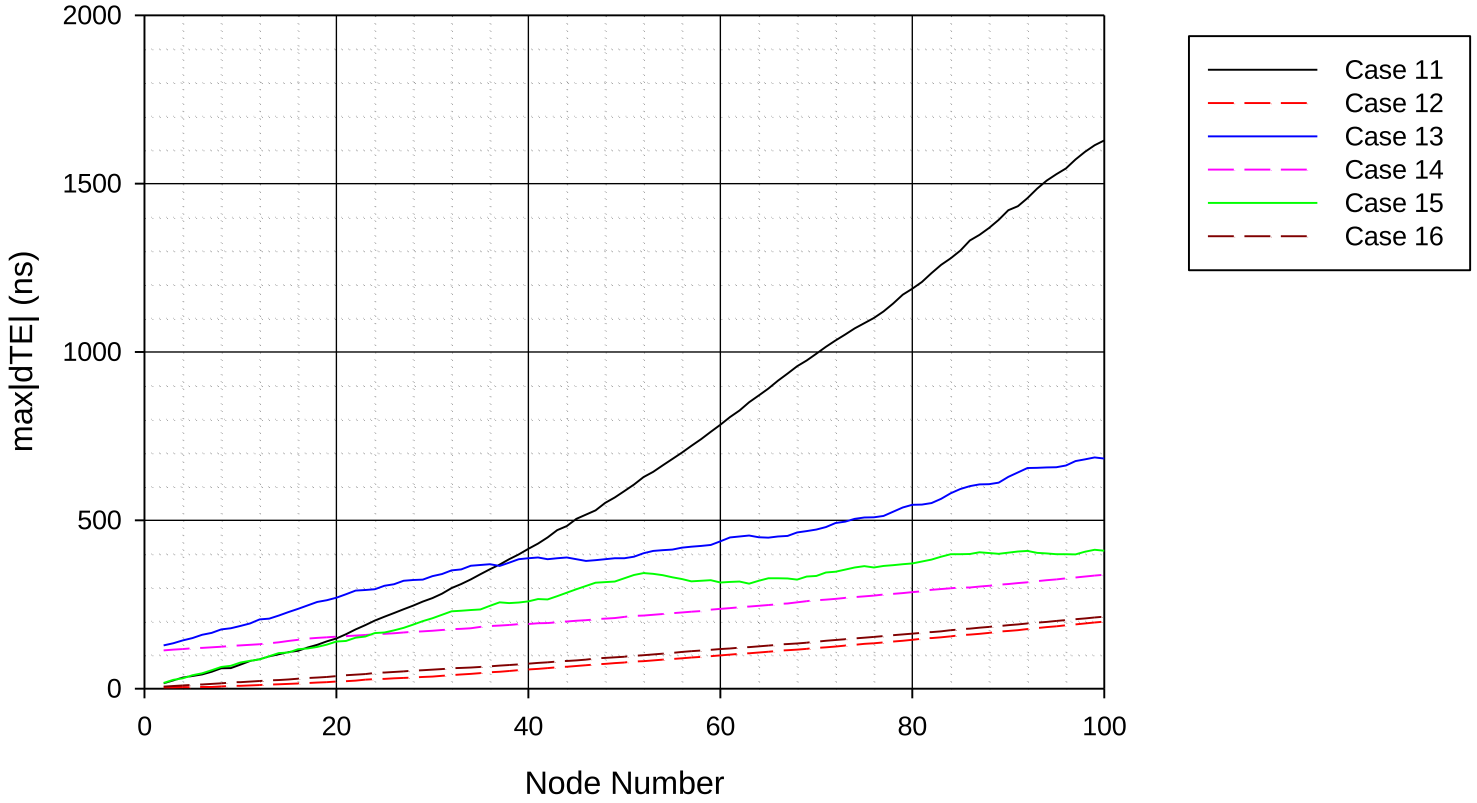
Detail of 100 – 102 s. Jumps reflect errors in neighborRateRatio measurements due to timestamp granularity, at each of the 100 Nodes.

Case 2, node 100 Frequency Offset Relative to GM

- In previous two slides, accumulated frequency offset of node 100 relative to GM is very close to actual frequency offset (triangular wave)
- Detail of 100 – 102 s shows effect of errors in neighborRateRatio measurements at each node, due to timestamp granularity
- For mean Pdelay interval of 0.03125 s (1/32 s), error due to 2 ns timestamp granularity is on the order of 2 ns / 0.03125 s = 64 ns/s
  - Over 100 hops, this could accumulate to as much as 6400 ns/s, but more likely some fraction of this, e.g., several hundred to 1000 ns/s, or several tenths to 1 ppm
  - Jumps in the plot on the previous slide are of this order

Cases 11 - 16: max|dTE| - 1

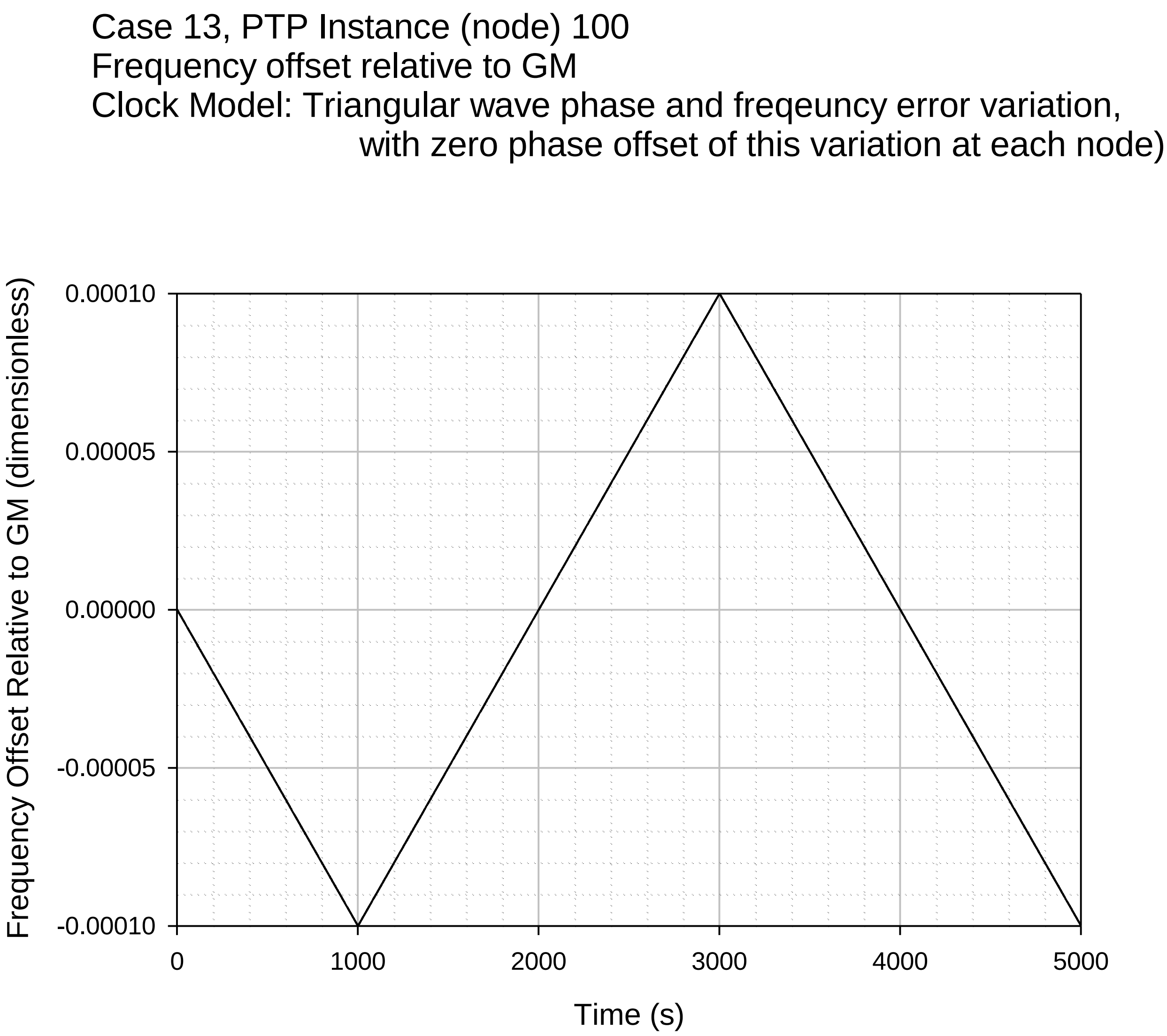
Simulation Cases 11 - 16
Single replication of simulation
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 0.1 ppm/s maximum drift rate
Cases 11 and 12: 32 Sync msgs/s, 32 Pdelay exchanges/s
Cases 13 and 14: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 15 and 16: 32 Sync msgs/s, 1 Pdelay exchange/s



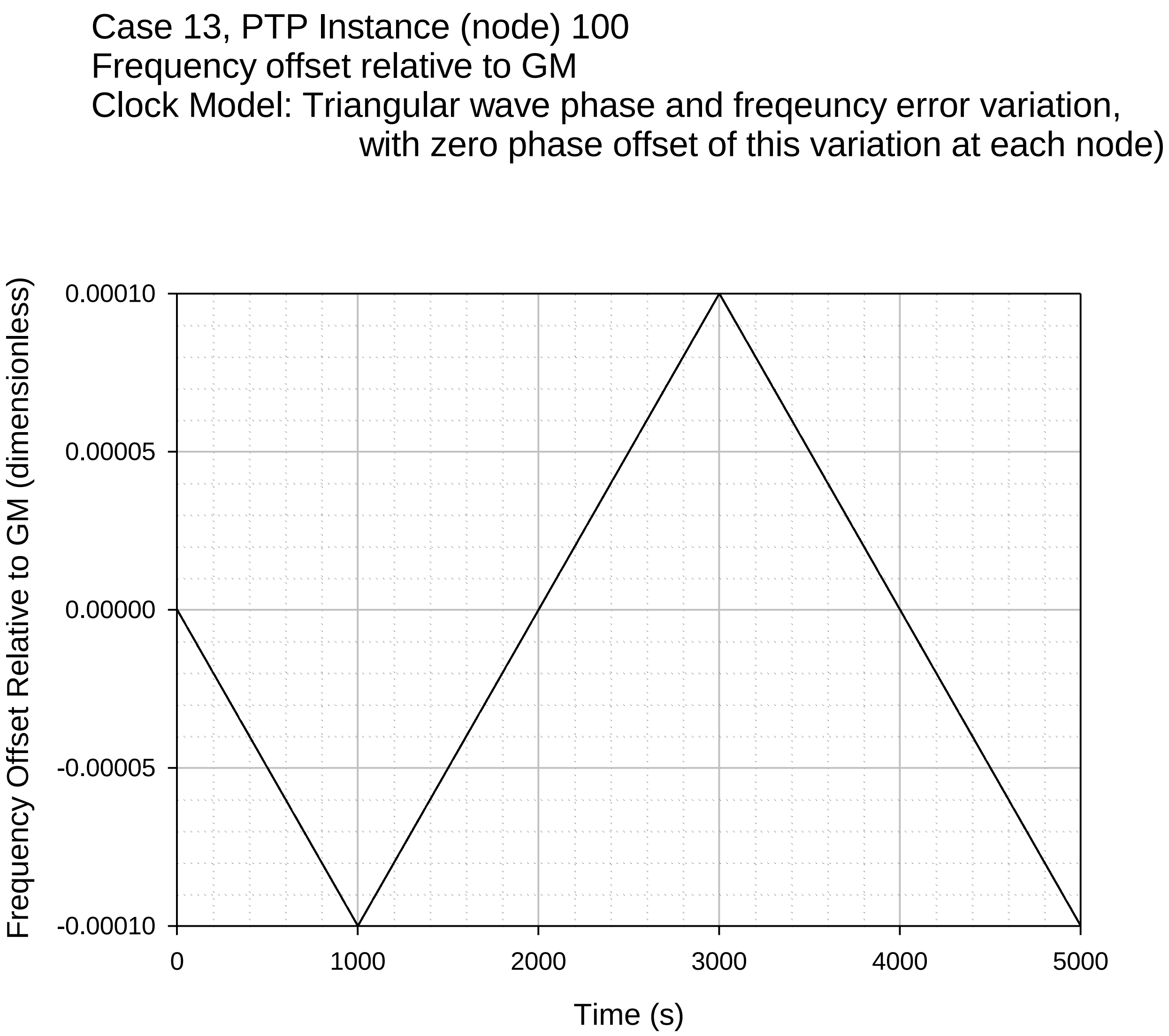
Cases 11 - 16: max|dTE| - 2

- Except for case 11 (32 messages/s for both Sync and Pdelay, and 8 ns timestamp granularity), max|dTE| is less than 1 μs for 100 hops
  - For case 11, max|dTE| reaches 1 μs at 75 hops
- The performance for cases 11 – 16 is significantly better than for cases 1 – 6 because the frequency drift is much lower (0.1 ppm/s versus 3 ppm/s)
- For cases 12 and 16 (2 ns timestamp granularity, 32 messages/s for Sync, 32 or 1 message/s for Pdelay), max|dTE| at 100 hops is approximately 200 ns
  - This allows 800 ns margin for cTE and other budget components
- For case 14 (1 message/s for both Sync and Pdelay, 2 ns timestamp granularity), max|dTE| reaches 338 ns at 100 hops
  - This allows 662 ns margin for cTE and other budget components
- For cases 13 and 15, the increase of max|dTE| with hop number is not smooth
  - The effect of statistical variability is apparent; it is due to the 8 ns timestamp granularity and the fact that GM rateRatio is updated less frequently (once per second versus 32 times per second)

Case 13, node 100 Frequency Offset Relative to GM

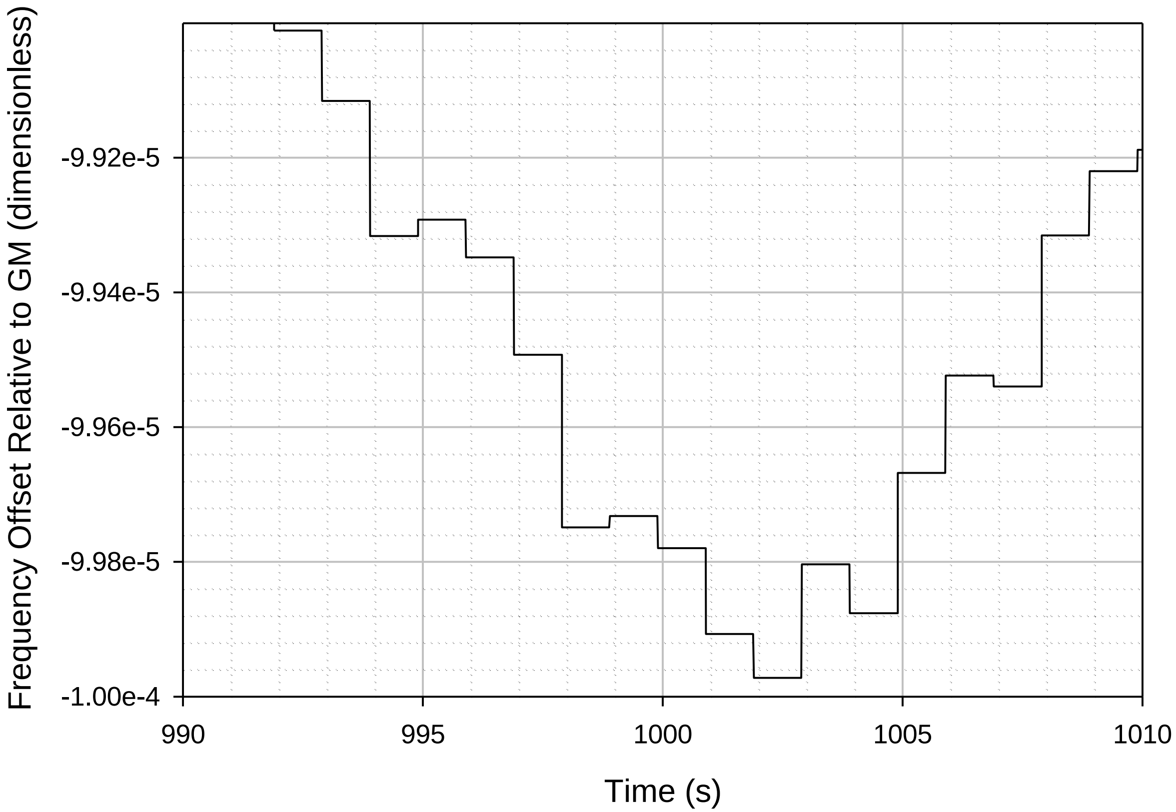


Case 13, node 100 Frequency Offset Relative to GM



Case 13, node 100 Frequency Offset Relative to GM

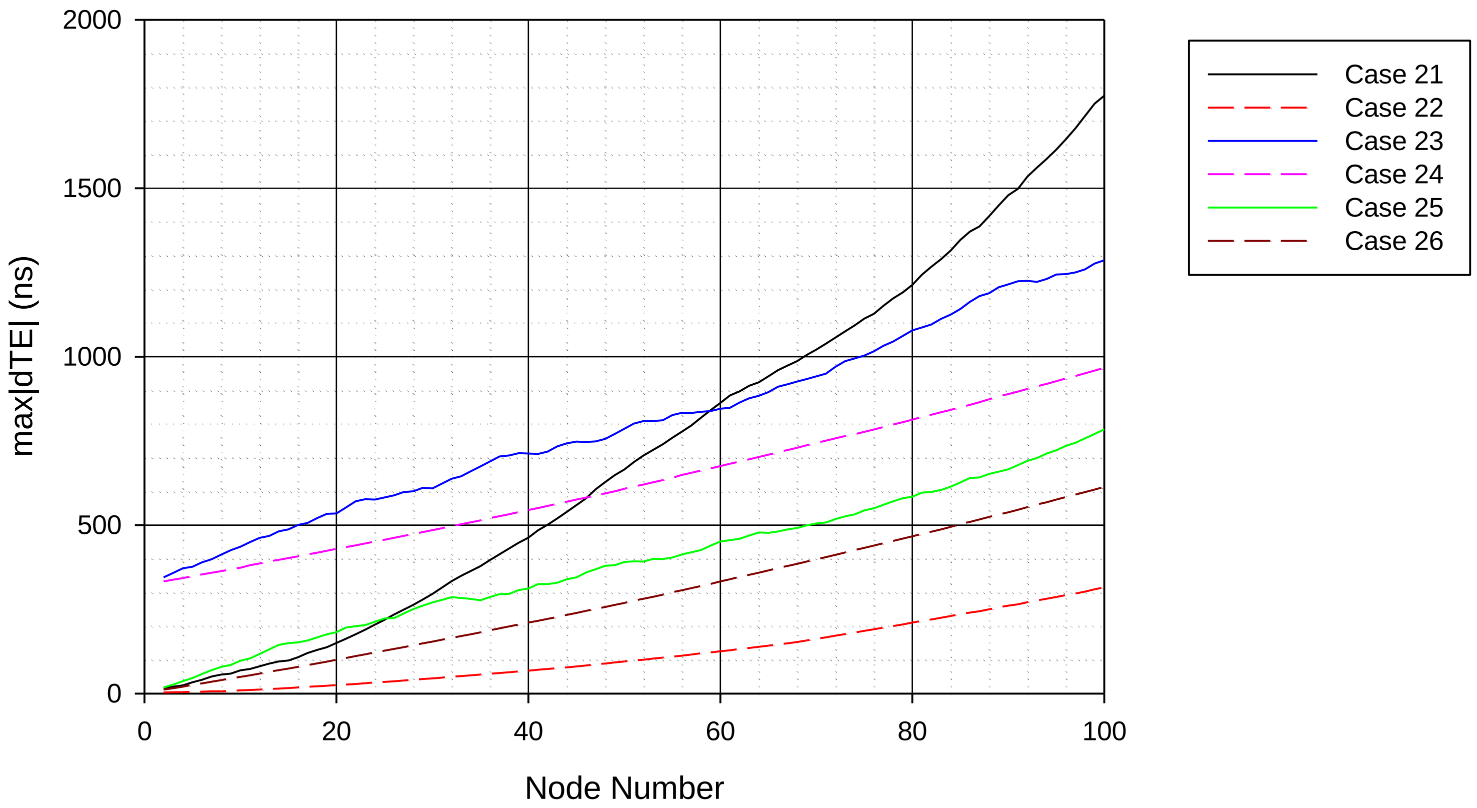
Case 13, PTP Instance (node) 100
Frequency offset relative to GM (detail of 990 - 1010 s)
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)



Detail of 100 – 102 s. Jumps reflect errors in neighborRateRatio measurements due to timestamp granularity, at each of the 100 Nodes.

Cases 21 – 26: max|dTE| - 1

Simulation Cases 21 - 26
Single replication of simulation
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 0.3 ppm/s maximum drift rate
Cases 21 and 22: 32 Sync msgs/s, 32 Pdelay exchanges/s
Cases 23 and 24: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 25 and 26: 32 Sync msgs/s, 1 Pdelay exchange/s

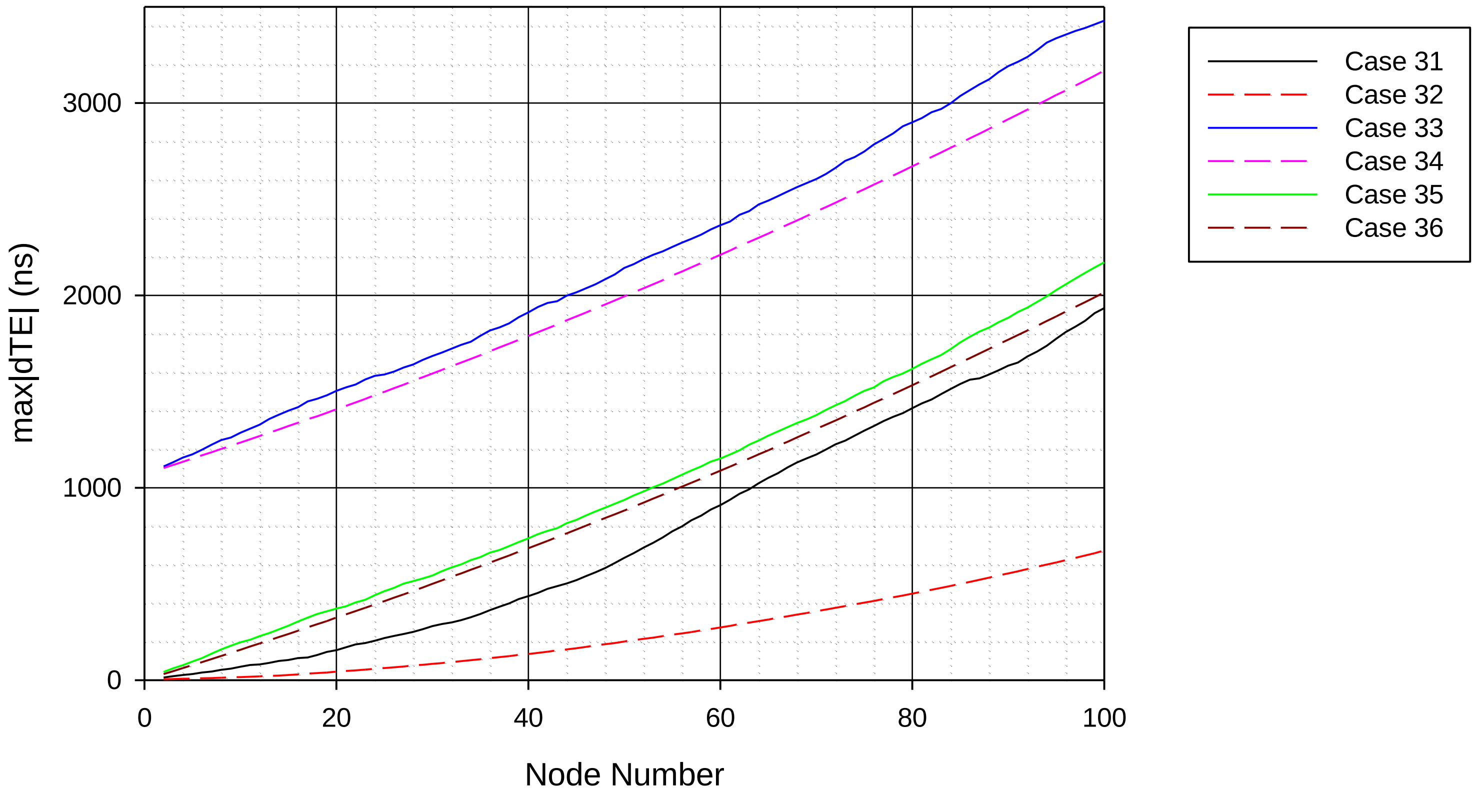


Cases 21 - 26: max|dTE| - 2

- max|dTE| for cases 22 and 24 – 26 is less than 1 μs for 100 hops
  - For case 22, max|dTE| is 315 ns at 100 hops and 139 ns at 64 hops
  - For case 26, max|dTE| is 613 ns at 100 hops and 359 ns at 64 hops
  - Both cases 22 and 26 leave margin (685 ns and 387 ns, respectively, at 100 hops) for other budget components
  - max|dTE| for cases 24 and 25 is much closer to 1 μs (966 ns for case 24 and 784 ns for case 25)
- max|dTE| is below 1 μs for all the cases at 64 hops, though the margin relative to 1 μs is small for cases 21 (924 ns max|dTE|)) and case 23 (883 ns max|dTE|))

Cases 31 - 36: max|dTE| - 1

Simulation Cases 31 - 36
Single replication of simulation
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 1 ppm/s maximum drift rate
Cases 31 and 32: 32 Sync msgs/s, 32 Pdelay exchanges/s
Cases 33 and 34: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 35 and 36: 32 Sync msgs/s, 1 Pdelay exchange/s



Cases 31 – 36: max|dTE| - 2

- Except for case 32 (32 messages/s for Sync and Pdelay, 2 ns timestamp granularity), max|dTE| exceeds 1 μs at 64 hops and 100 hops
- For case 32, max|dTE| is 306 ns at 64 hops and 672 ns at 100 hops

Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 1

- In cases 3s, 4s, 5s, 6s, 5s10, and 6s10, the determination of GM rateRatio using successive Sync messages (i.e., Sync messages that are *n* messages apart, where *n* is an integer) is considered
  - The Sync messages are consecutive if *n* = 1; this corresponds to cases 3s, 4s, 5s, and 6s
  - In cases 5s10 and 6s10, the Sync messages are 10 messages apart, i.e., a Sync message and another Sync message that is 9 messages later are used to determine GM rateRatio
  - See slide 15 for details
- This approach was described informatively in Clause 6 of IEEE Std 1588™ – 2011, and was briefly considered during the development of IEEE Std 802.1AS™ – 2011
- Email discussion on the 802.1 reflector during February and March 2007 pointed out that this approach gives rise to gain peaking
  - The gain peaking arises in the transfer function that relates computed GM rate ratio at a PTP Instance and the input syncEventIngressTimestamp, and also in the transfer function that relates computed synchronized time at a PTP Instance and the input syncEventIngressTimestamp

Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 2

- Note that this particular gain peaking is not related to the presence of a phase-locked loop (PLL); in fact, it occurs in a PTP Instance that uses successive Sync messages to measure frequency offset relative to the GM even if no PLLs are present
  - It is analogous to gain peaking in a PLL in the sense that the respective transfer function has a low-pass characteristic and a maximum gain that exceeds 1.0 (0 dB) in the vicinity of the corner frequency
- Gain peaking in cascaded filters can give rise to large amplification of the input over many hops, and can lead to instability
  - This effect has been known for many years with respect to cascaded PLLs (see [5] and [6])
    - After a sufficient number of hops, which depends on how large the gain peaking is, a large increase in PLL phase error will be produced
  - The effect in the scheme where successive Sync messages are used to determine GM rateRatio was analyzed analytically in [7] and [8]
    - It was shown that the growth of phase/time error is exponential in the number of hops, and the rate depends on the ratio of residence time (T_r) to the frequency update interval (T_I)

Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 3

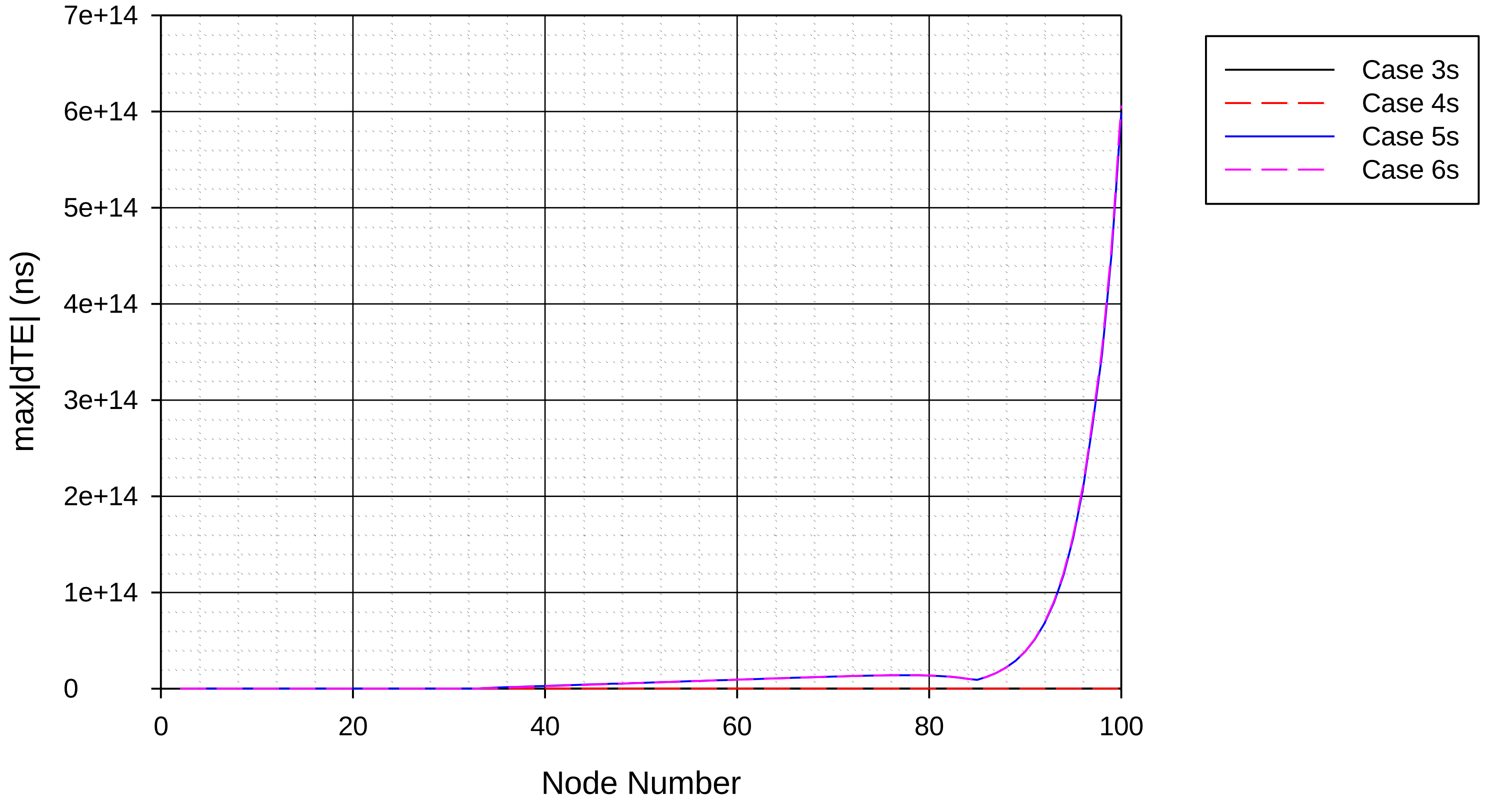
- Specifically, the growth in time error is proportional to $(1+a(T_r/T_l))^k$, where k is the number of hops and a is a constant (see [7] and [8])
- Intuitively, the effect occurs because the time error at a PTP Instance results in residence time error, which in turn results in an error in the frequency offset computation at downstream nodes
  - This effect gets amplified as one proceeds down the chain
  - The increase in error in the computed frequency offset at each node results in increasing time error as one proceeds down the chain
- The effect is clearly seen in the next slides, which time error accumulation results for cases 3s, 4s, 5s, and 6s

Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 4

- In the next slide, the time error for cases 5s and 6s (32 messages/s for Sync) has become unstable (the results for cases 3s and 4s are not visible on the scale of the plot)
  - The subsequent four slides show the computed (measured) frequency offset relative to the GM at nodes 10, 15, 20, and 30
  - While the computed frequency offset at node 10 matches the actual frequency offset very well, the reproduction gets progressively worse as one proceeds to nodes 15, 20, and 30
    - The shape of the triangular wave becomes less sharp, and the error in the offset becomes larger
    - By Node 30, the triangular wave shape is gone, and the peak measured frequency offset exceeds 0.05 (i.e., 50000 ppm)
    - In fact, after node 33 the computation of frequency offset is unstable; results beyond node 33 are meaningless

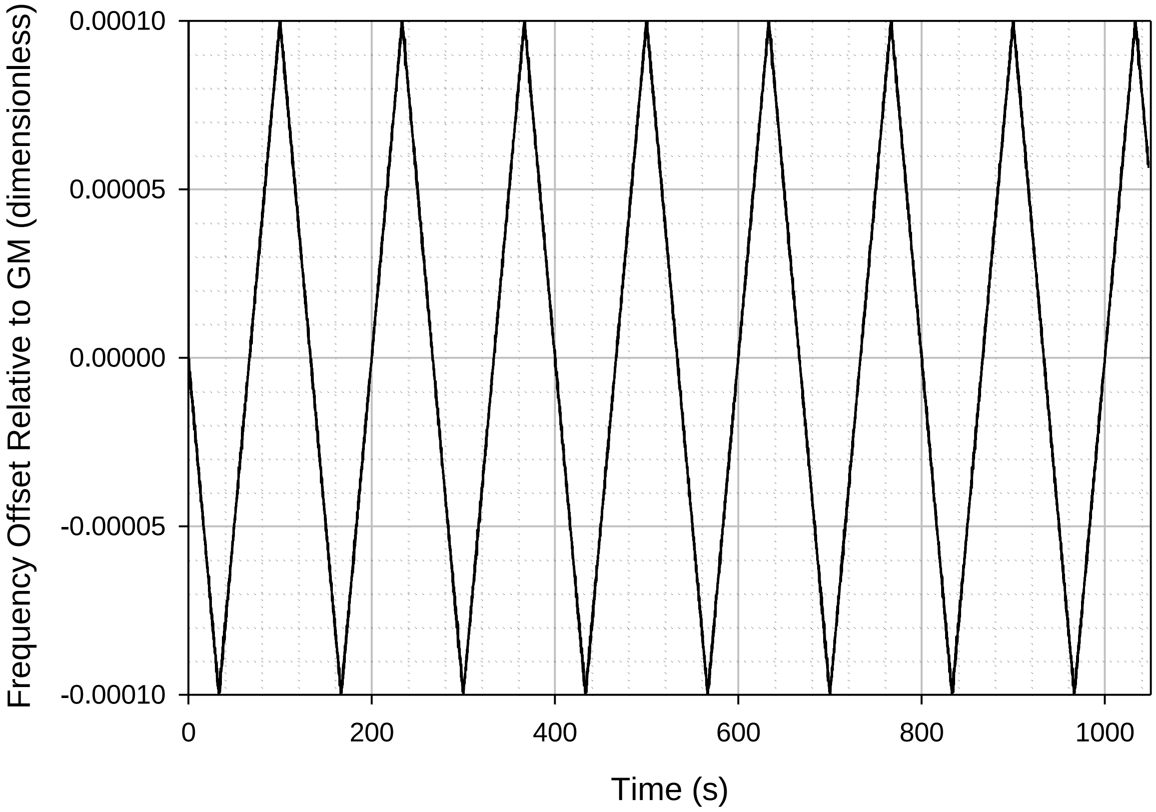
Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 5

Simulation Cases 3s - 6s
Single replication of simulation
Measure GM rate ratio using successive Sync messages
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 3 ppm/s maximum drift rate
Cases 3s and 4s: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 5s and 6s: 32 Sync msgs/s, 1 Pdelay exchange/s



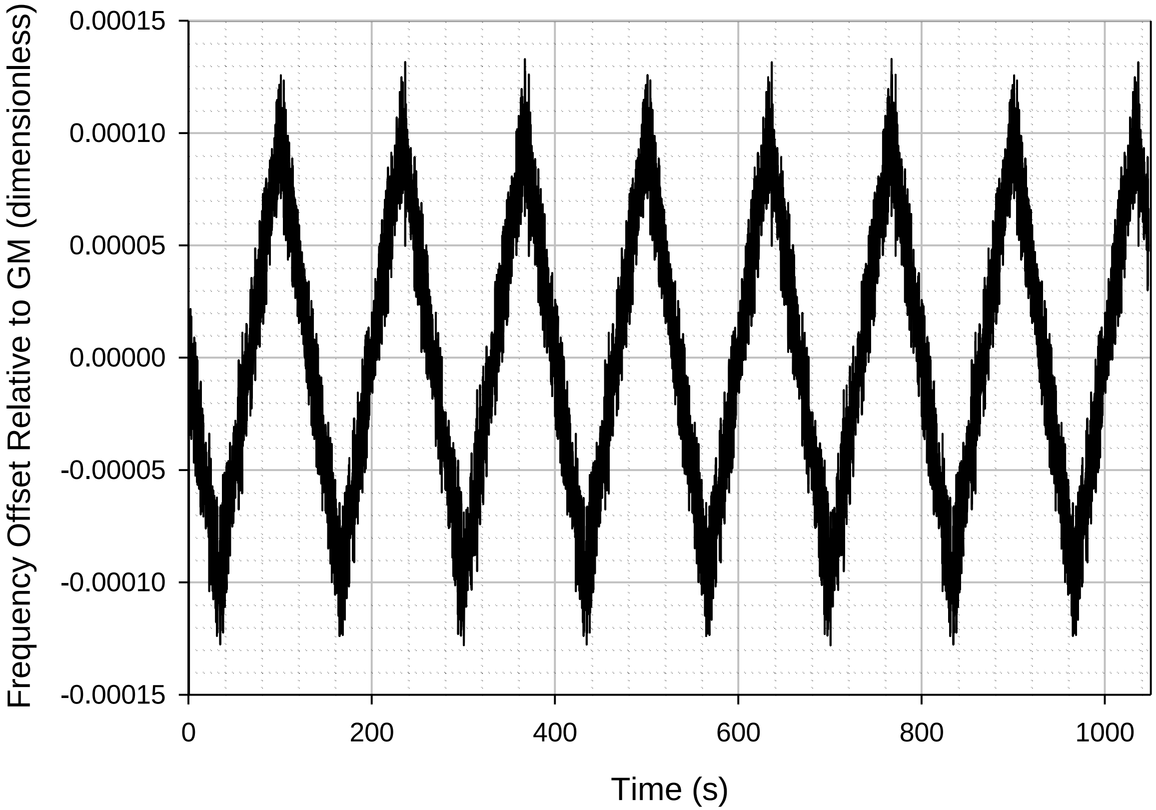
Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 6

Case 6s, PTP Instance (node) 10
Frequency offset relative to GM
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)



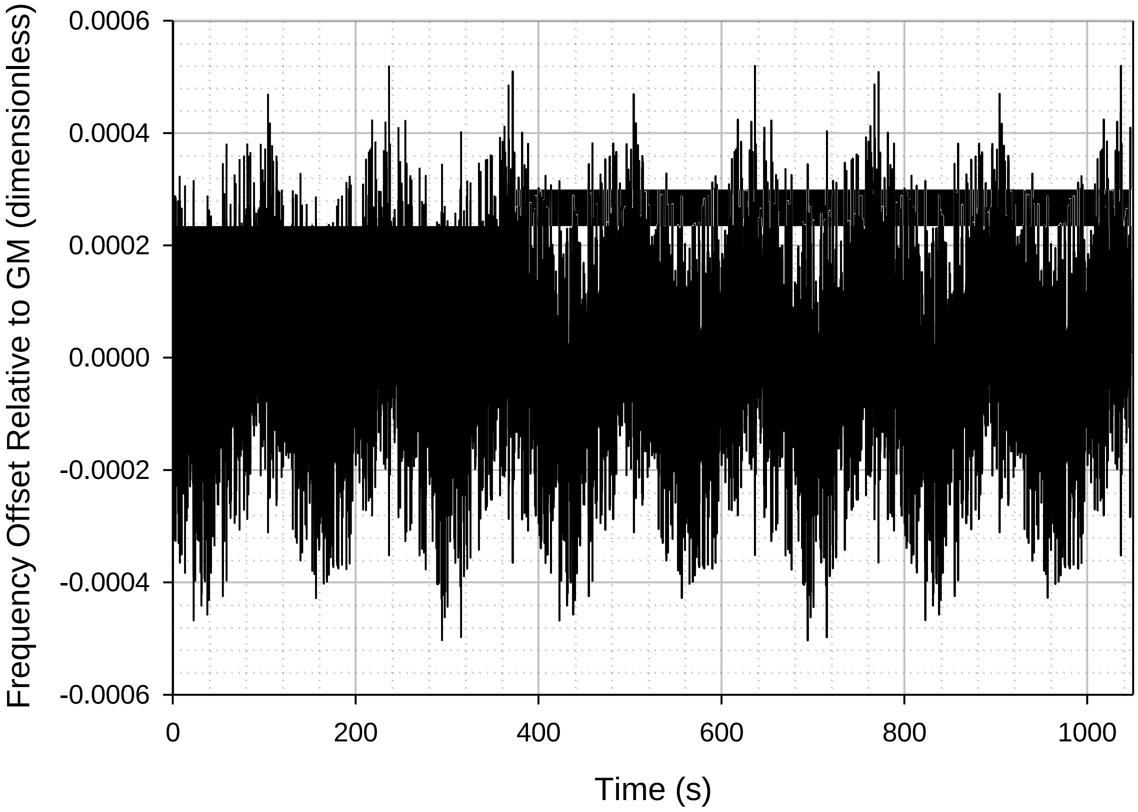
Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 5

Case 6s, PTP Instance (node) 15
Frequency offset relative to GM
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)



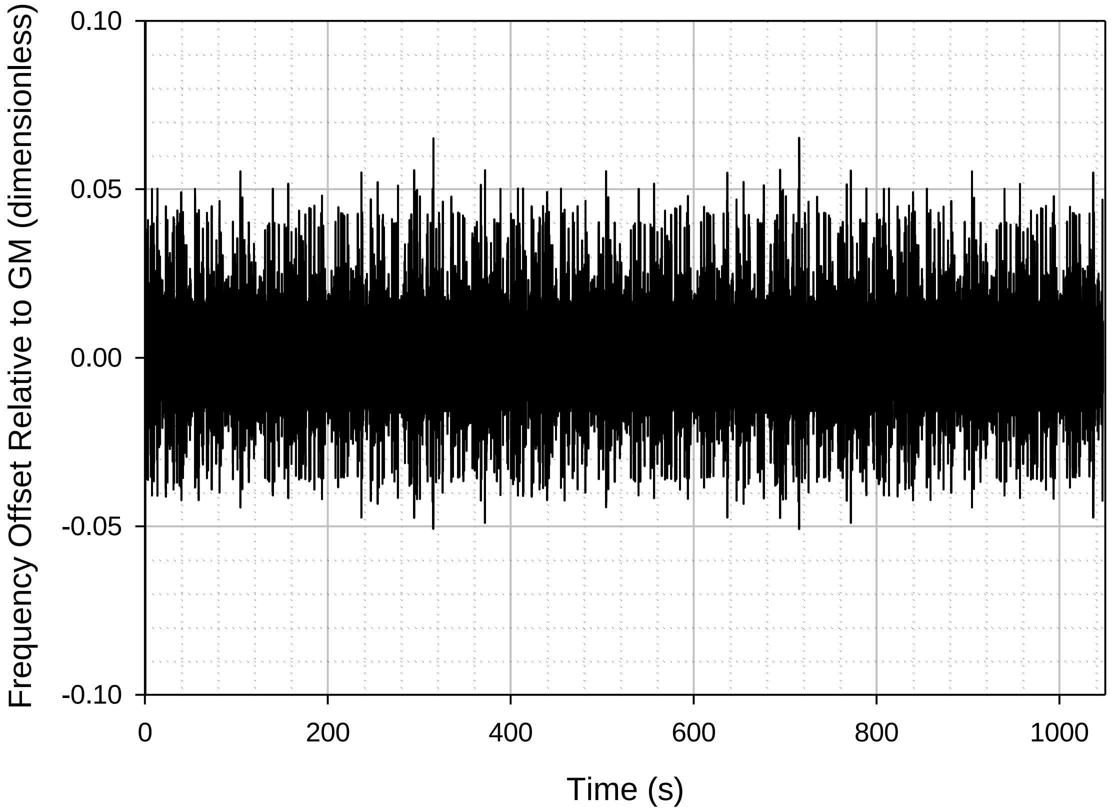
Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 7

Case 6s, PTP Instance (node) 20
Frequency offset relative to GM
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)



Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 8

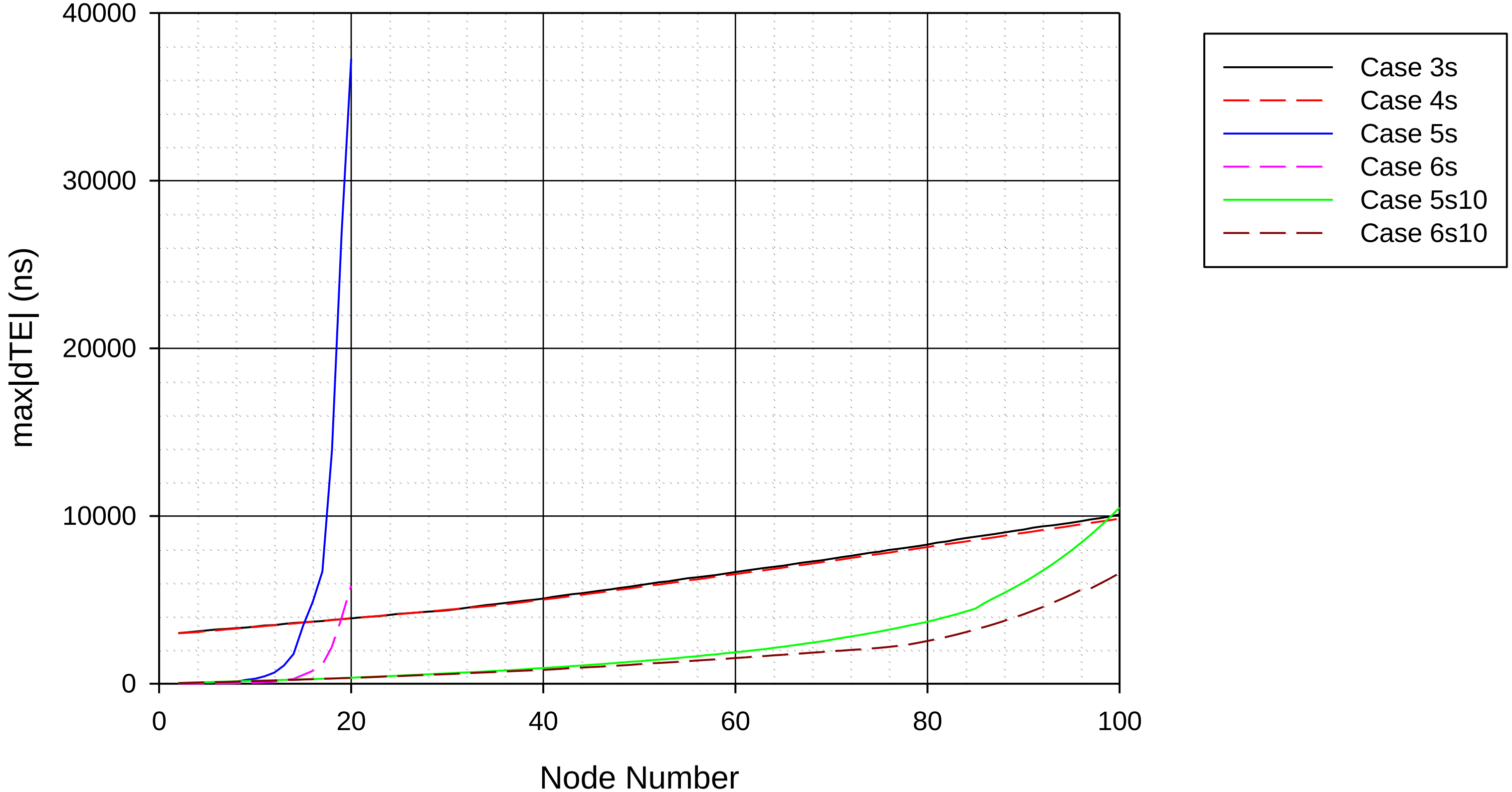
Case 6s, PTP Instance (node) 30
Frequency offset relative to GM
Clock Model: Triangular wave phase and freqeuncy error variation,
with zero phase offset of this variation at each node)



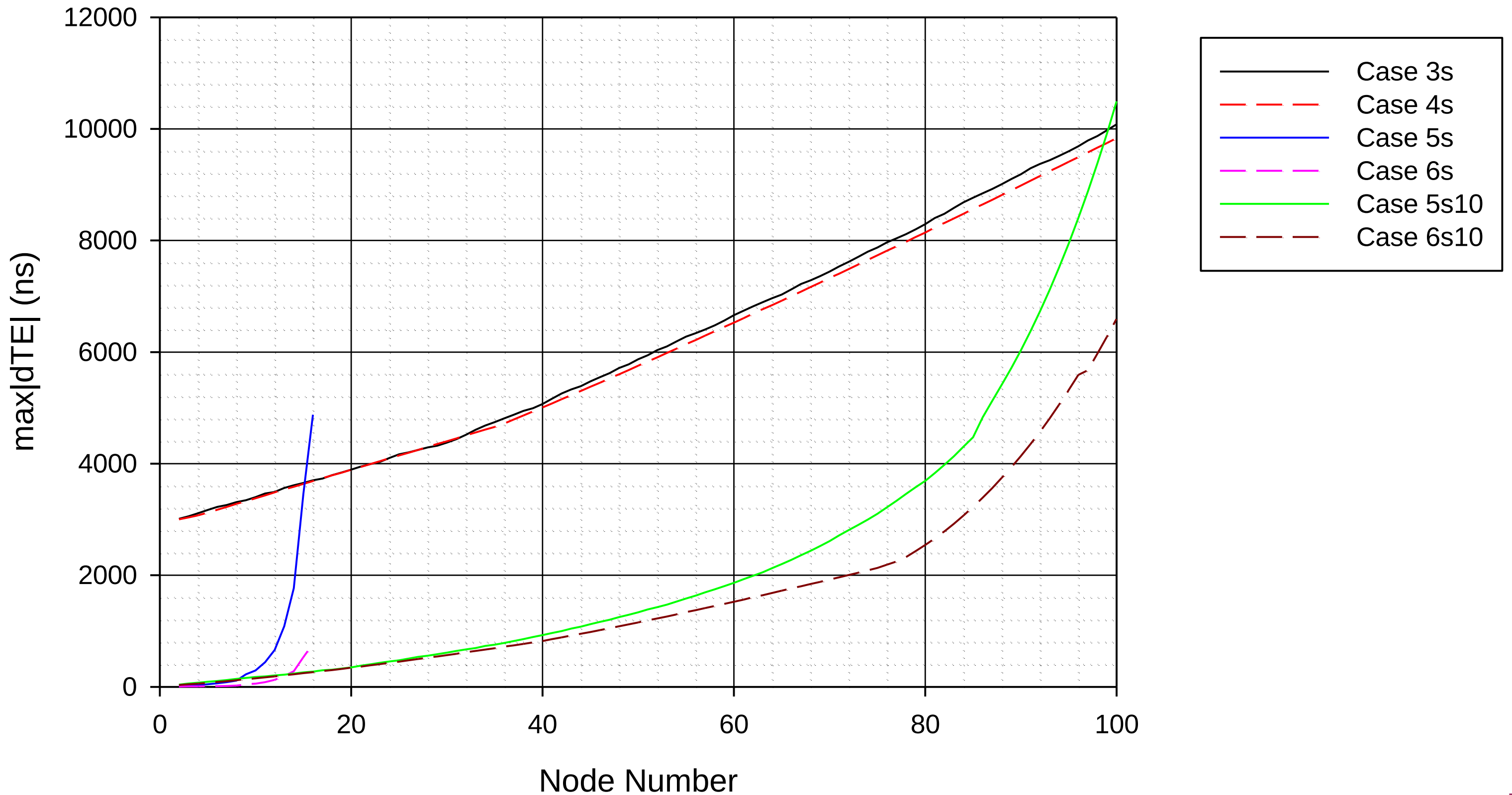
Cases 3s, 4s, 5s, 6s, 5s10, 6s10 - 9

- The following two slides show max|dTE| for cases 3s, 4s, 5s, 6s, 5s10, and 6s10, but with the case 5s and 6s results shown only up to node 20, and node 16, respectively, so that results for the other cases can be seen on the scale of the plot
- Cases 3s, 4s, 5s10, and 6s10 show much less rapid time error growth compared to cases 5s and 6s, due to the longer frequency update interval T_I (1 s for cases 3s and 4s, 0.3125 s for cases 5s10 and 6s10) relative to residence time T_r (10 ms); this is because the effective gain peaking is smaller
- However, max|dTE| exceeds 1 μs for cases 3s and 4s at all nodes, and reaches 1 μs for cases 5s10 and 6s10 after approximately 44 hops
- The results for cases 5s10 and 6s10 suggest that lowering the gain peaking can improve the accumulated time error
  - This can certainly be done by making the residence time sufficiently small; analysis (further simulations) would be needed to determine how much smaller than 10 ms it would need to be
  - This also can be done by increasing the frequency update interval; however, if it is increased too much the error due to not having an updated frequency measurement will increase

Simulation Cases 3s - 6s
Single replication of simulation
Measure GM rate ratio using successive Sync messages
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 3 ppm/s maximum drift rate
Cases 3s and 4s: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 5s and 6s: 32 Sync msgs/s, 1 Pdelay exchange/s
Cases 5s10 and 6s10: 32 Sync msgs/s, 1 Pdelay exchange/s, measure
GM rate ration using every 10th Sync message, jumping window



Simulation Cases 3s - 6s
Single replication of simulation
Measure GM rate ratio using successive Sync messages
Clock Model: triangular wave freqeuncy variation
+/- 100 ppm amplitude, 3 ppm/s maximum drift rate
Cases 3s and 4s: 1 Sync msg/s, 1 Pdelay exchange/s
Cases 5s and 6s: 32 Sync msgs/s, 1 Pdelay exchange/s
Cases 5s10 and 6s10: 32 Sync msgs/s, 1 Pdelay exchange/s, measure
GM rate ration using every 10th Sync message, jumping window



Conclusion and Discussion of Next Steps - 1

- The simulation results for frequency drift rate of 3 ppm/s either exceed 1 μs or do not leave sufficient margin for other time error budget components (e.g., cTE), for both methods of measuring frequency offset relative to the GM (i.e., accumulating neighborRateRatio, and using successive Sync messages)
- The simulations for cases where the frequency drift rate was smaller only considered frequency offset measurement by accumulating neighborRateRatio
  - Results for 1 ppm/s (cases 31 – 36) either exceed 1 μs or do not leave sufficient margin for other time error budget components
  - Results for 0.1 ppm/s (cases 11 – 16) meet 1 μs with some margin, for cases with 2 ns timestamp granularity
  - Results for 0.3 ppm/s (cases 21 – 26) meet 1 μs with some margin for the case with 32 messages/s for Sync and Pdelay, and 2 ns timestamp granularity
- The method of measuring frequency offset relative to the GM using successive Sync messages results in max|dTE| that exceeds 1 μs over 64 hops in all the cases considered
  - Possibly this method can be improved by reducing the gain peaking, either by decreasing the residence time (from the current 10 ms) or increasing the frequency update interval (though this might result in increased dTE due to larger frequency error)

Conclusion and Discussion of Next Steps - 2

- Another improvement to the method of using successive Sync messages was discussed in 802.1 during the February – May 2007 timeframe
  - This method was referred to as the “split syntonization method” (see [7] – [12])
  - In this method, the residence time is divided into two parts:
    - a) The residence time as measured by the free-running local clock
    - b) The correction to the residence time due to the measured rateRatio of the local clock relative to the GM
  - Component (a) above is accumulated in the correctionField (along with the gPTP link delay
  - Component (b) is accumulated separately (e.g., in a TLV)
  - rateRatio relative to the GM is computed using correctedMasterTime values that include component (a) but not component (b)
  - Synchronized time is computed using correctedMasterTime values that include components (a) and (b)
  - By not including component (b) in the rateRatio computation, the effect of errors in rateRatio at one node are not propagated downstream
  - Initial analysis of this scheme (see [8] and [9]) indicated that the growth in time error as a function of node number would be linear rather than exponential
  - However, this must be confirmed; this work was not pursued in 2007 because 802.1 decided to use the current method, i.e., accumulating neighborRateRatio

Conclusion and Discussion of Next Steps - 3

- The following are possible approaches to meeting 1 μs over 64 hops, and possibly over 100 hops
  - Use more stable oscillators, e.g., maximum drift rate in the 0.1 – 0.3 ppm/s range, or oscillators that meet the TDEV mask of Annex B/802.1AS (this latter requirement was shown in [1] to have max|dTE| on the order of 140 ns over 100 hops
  - Use the successive Sync message approach to measure GM rateRatio, and try to decrease gain peaking by reducing residence time and possibly increasing the frequency update interval
    - Simulations would be needed to determine how large a reduction in residence time would be required
  - Use the successive Sync message approach with the split syntonization method
    - Either a new TLV, or modification to an existing TLV, would be needed to carry the additional information
    - Simulations would be needed to determine the achievable performance
  - If there is interest in either of the approaches that use successive Sync messages, the author will prepare and present a modification/update to [8] and [9], which analyze the stability of each approach

Conclusion and Discussion of Next Steps - 4

- It is proposed that the IEC/IEEE 60802 group discuss the above possible approaches and decide which one or ones to investigate further

Thank you

References - 1

[1] Geoffrey M. Garner, *Initial Simulation Results for Time Error Accumulation in an IEC/IEEE 60802 Network*, IEEE 802.1 presentation, March 2020.

[2] Geoffrey M. Garner, *Discussion of Assumptions Needed for 60802 Network Simulations*, IEEE 802.1 presentation, January 2020.

[3] Geoffrey M. Garner, *Comparison of 802.1AS Annex B and 60802 Clock Stability*, IEEE 802.1 presentation, January 2020.

[4] Geoffrey M. Garner, *Simulation Results for Time Error Accumulation in a Cluster HRM of T-BCs Timed by PRTC-B (G.Supp65)*, Correspondence Document for ITU-T Q13/15, April/May 2020 Correspondence, CD13-08.

[5] Eve Varma and J. Wu, *Analysis of Jitter Accumulation in a Chain of Digital Regenerators*, Proceedings of IEEE Globecom, Vol. 2, 1982, p. 653-657.

[6] ITU-T Recommendation G.8251, *The control of jitter and wander within the optical transport network (OTN)*, Geneva, November 2018 (see Appendix IV, 3R regenerator jitter accumulation analyses)

References - 2

[7] Chuck Harrison, *Cascaded Gain Phenomena in IEEE 1588v2 Transparent Clock Chains an Initial Study*, IEEE 802.1 contribution, March 3, 2007.

[8] Geoffrey M. Garner, *Effect of a Frequency Perturbation in a Chain of Syntonized Transparent Clocks*, IEEE 802.1 presentation, March 10, 2007

[9] Geoffrey M. Garner, Effect of a Frequency Perturbation in a Chain of Syntonized Transparent Clocks, contribution (white paper) to IEEE 1588 and IEEE 802.1, March 10, 2007.

[10] Chuck Harrison, *Transparent Clock Gain Cascade: Split Path Scenario*, presentation to IEEE 802.1, March 9, 2007

[11] David James, *AV Bridging: Time Synchronization Clocking*, presentation to IEEE 802.1 teleconference, May 21, 2007.

[12] David James, *DVJ Perspective on: Timing and synchronization for time-sensitive applications in bridges local area networks*, Contribution to IEEE 802.1, Draft 0.710, May 30, 2007.