Summary of Assumptions for Next Simulations, based on Presentation and Subsequent Discussion of [1] Revision 1

Geoffrey M. Garner
Huawei (Consultant)

gmgarner@alum.mit.edu

IEEE 802.1 TSN TG
2021.03.11

Introduction

- Reference [1] obtained time histories for frequency offset, frequency drift rate, and phase offset for the LocalClock entity, based on previous frequency stability data presented in Reference [3] of [1] and a temperature profile described in Reference [1] of [1]
- In the discussion of [1] following its presentation, a set of assumptions was decided on for the next simulations
- The current presentation summarizes these assumptions

Assumptions for Temperature Profile

- The temperature history of [1] is assumed to vary between – 40°C and +85°C, at a rate of 1°C /s
- When the temperature is increasing and reaches +85°C, it remains at +85°C for 30 s
- The temperature then decreases from +85°C to – 40°C at a rate of 1°C /s; this takes 125 s
- The temperature then remains at – 40°C for 30 s
- The temperature then increases to +85°C at a rate of 1°C /s; this takes 125 s
- The duration of the entire cycle (i.e., the period) is therefore 310 s (5.166667 min)
  - This compares with the 1200 s cycle duration for the assumptions described in [1]

Assumptions for Frequency Stability due to Temperature Variation

- The dependence of frequency offset on temperature is assumed to be as described in [1]
  - Specifically, the values a_0, a_1, a_2, and a_3 computed in [1] 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 [1]
  - This data was provided by the author of Reference [3] of [1]
- 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

- Two types of assumptions will be used for relative time offsets of the phase error histories at each node (separate cases will be run for each assumption):
  - 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, see slide 3)
  - Choose the phase of the LocalClock time error waveform at each node randomly in the range [0, 0.1T], at initialization, where T is the period of the phase and frequency variation waveforms (i.e., 310 s, see slide 3)
    - A uniform probability distribution is used for the random choice
    - 0.1T = 31 s, i.e., any periodic LocalClock time error waveform will be offset from any other such waveform by at most 31 s

Other Assumptions - 1

- Some other assumptions were briefly suggested in email discussion
  - Mean Sync interval: 125 ms
  - Mean Pdelay interval: 31.25 ms
  - Timestamp granularity: 8 ns, 4 ns (both cases)
  - Residence times: 1 ms, 4 ms, 10 ms (all 3 cases)
  - Timestamp error (±8 ns, each with 0.5 probability)
- The above, along with the two different assumptions for random offsets for phase error waveform implies 12 simulation cases ($2 \times 2 \times 3$)
- Other assumptions can be taken from the most recent simulations [2], and are summarized on the following slides
  - Note that initial simulations will assume GM error of zero; GM error will be added after other assumptions are settled on

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; here, GM time error is 0, so max|$dTE_{R(k,\, 0)}$| = max|dTE|) (b) Measured LocalClock rateRatio (frequency offset) relative to GM, for comparison with actual LocalClock frequency offset (results will be plotted for nodes 1, 34, 67, and 100 (where node 1 is the first node after the GM, and the GM is node 0))
Use syncLocked mode for PTP Instances downstream of GM	Yes
Endpoint filter parameters	$K_p K_o$ = 11, $K_i K_o$ = 65 (f_{3dB} = 2.5998 Hz, 1.288 dB gain peaking, ζ = 0.68219)
Simulation time	(a) For single replication cases: 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) (b) For multiple replication cases, may need to be shorter than 3150 s depending on run times

Other Assumptions - 3

Assumption/Parameter	Description/Value
Number of independent replications, for each simulation case	(a) Single replication cases (i.e. 1) (b) Multiple replication cases (300, subject to acceptable run times; these cases will be run later, after presenting and discussing results for single-replication cases)
GM rateRatio and neighborRateRatio computation granularity	0
Mean link delay	500 ns
Link asymmetry	0
Dynamic timestamp error for event messages (Sync, Pdelay-Req, Pdelay_Resp) due to variable delays within the PHY	±8 ns; for each timestamp taken, a random error is generated. The error is + 8 ns with probability 0.5, and – 8 ns with probability 0.5. The errors are independent for different timestamps and different PTP Instances.
Any variable PHY delay in addition to the dynamic timestamp error described above is assumed to be zero	0

Additional Questions, and Resulting Answers, on Assumptions - 1

- The method described below, using a window and computing the median, for computing neighborRateRatio, will **not** be used
  - neighborRateRatio will be computed using the current and most recent Pdelay exchange
- In cases 9 – 11 of [5], which used neighborRateRatio accumulation to measure GM rateRatio, neighborRateRatio was measured using a methodology similar to that used for GM rateRatio via successive Sync messages
- In these cases, a window size of 7 was used, i.e., the difference was taken between respective timestamps of current Pdelay exchange and 7th previous Pdelay exchange
  - In addition, the current estimate of neighborRateRatio was taken as the median of the most recent 7 measurements (including the current measurement)

Additional Questions, and Resulting Answers, on Assumptions - 2

- In cases 12 – 14 of [5], which measured GM rateRatio using successive Sync messages, this same approach was used for both the measurement of GM rateRatio (using Sync messages) and neighborRateRatio (using Pdelay exchanges)
  - neighborRateRatio measurements were needed for compensation of different rates of Pdelay requestor and responder in accounting for Pdelay turnaround time
  - However, in these cases the window size was 11 rather than 7
- Should this approach be used in the new simulations for the computation of neighborRateRatio?
  - If so, should the window size be 7, 11, or something else?

Additional Questions, and Resulting Answers, on Assumptions - 3

- The method described below, computing the average link delay based on the current and previous 15 Pdelay measurements, will **not** be used
  - The mean link delay value obtained from the current Pdelay exchange will be used
- On a related point, should the successive link delays measured using Pdelay be averaged over a sliding window?
  - Reference [2] indicates that the link delay measurements for all the cases there (cases 1 – 14) are averaged over a sliding window of size 16
  - However, the sliding window apparently was not used in the simulations, i.e., the window size was 1
  - In any case, should a sliding window be used and, if so, what should its size be?

Thank you

References - 1

[1] 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)

[2] Geoffrey M. Garner, *Further Simulation Results for Dynamic Time Error Performance for Transport over an IEC/IEEE 60802 Network Based on Updated Assumptions*, Revision 2, December 14, 2020 call (available at https://www.ieee802.org/1/files/public/docs2020/60802-garner-further-simulation-results-time-sync-transport-1120-v02.pdf)