Cascaded Gain Phenomena in IEEE 1588v2 Transparent Clock Chains: an Initial Study

Chuck Harrison 3-Mar-2007

1 Background

In February 2007 this author initiated some discussion on the IEEE 1588 email reflector regarding the potential for error accumulation in a chain of syntonized clocks by a "gain peaking" mechanism. This has resulted in a frequency-domain analytical study[1] by Geoff Garner and a time-domain phase-error simulation reported in this document. These studies are in fundamental agreement about the effect but drew somewhat different conclusions about its scale. This memo concludes that both studies are valid, and apply to different assumptions about the original point of injection for the error being amplified.

2 Summary of results

Both studies found that gain buildup in over a chain of devices is a genuine phenomenon exhibited by nodes using the TC (transparent clock) syntonization processes described in clauses 11.4.5.1, 11.5.2.2, and 12.2.1 of 1588-v2-D1-A17feb-07.pdf. This behavior is rather similar in form to the well-known gain peaking exhibited by cascaded PLL (phase locked loop) clock regeneration devices, but this fact regarding TCs was perhaps not widely recognized within the 1588 community.

Both studies suggest that the end-to-end jitter/wander amplification was tolerable for two specific operating conditions of interest:

1) A long chain of high-performance TCs, with residence time ~100 usec and frequency update rate ~10Hz, and
2) A short chain of "AVB" TCs, with residence time ~10 msec and frequency update rate ~ 10Hz

The "headroom" involved in assessing the effect as tolerable is much less in the Harrison study than in the Garner study. For operation at other conditions the conclusions of the two studies differ enough to affect system design decisions.

The cascaded gain magnitude can be described as a figure-of-merit for cascaded clocks. Both studies found that, at the worst-case perturbation frequency, the gain as a perturbation passes through a chain of n syntonized TCs is estimated by

$$G = G_0\,(1+a\cdot T_r/T_I)^n \qquad \text{(eq. 1)}$$

where T_r is the SyncEvent residence time and T_I is the integration time described in clause 12.2.1. The values of G_0 and a vary between the studies. The behavior of the per-stage peaking factor $pf = (1+a\cdot T_r/T_I)$ is consistent between the two studies. The peaking factor for a single stage is commonly expressed in dB as $20\cdot\log_{10}(pf)$.

The results of the studies are compared in Table 1 below.

[1] as-garner-protocol-synton-chain-freq-offset-accum-0207.pdf

	Garner study	Harrison study
Propagation analyzed	Frequency error	Phase error
Technique	Fourier analysis	Simulation
G_0	$2 \cdot T_r/T_I$	1
a @ worst-case period	2	1.3
Peaking factor *pf:*		
$T_r/T_I = 0.001$	0.017 dB	0.011 dB
$T_r/T_I = 0.01$	0.17 dB	0.11 dB
$T_r/T_I = 0.1$	1.6 dB	1.1 dB
Hops to double:		
$T_r/T_I = 0.001$	3800	547
$T_r/T_I = 0.01$	268	55
$T_r/T_I = 0.1$	16	6
Worst-case wander period	$2 \cdot T_I$	$3 \cdot T_I$

Table 1.

3 Contrasts in the studies

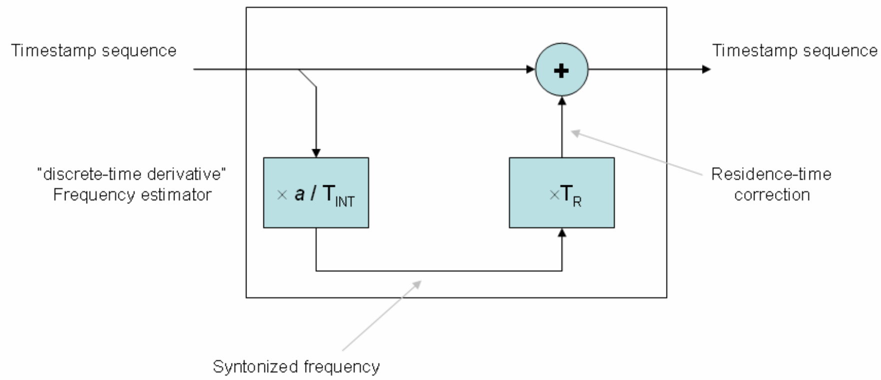
Three conclusions of the respective studies stand in contrast:

(1) The worst-case perturbation frequency was found about 1/3 less in the Harrison study, and
(2) The parameter *a* in the amplification factor of eq. 1 was estimated to be 1/3 less in the Harrison study than the value computed in the Garner analysis, and
(3) A distinctive first-stage behavior which caused attenuation of the perturbation was found in the Garner study but not in the Harrison study.

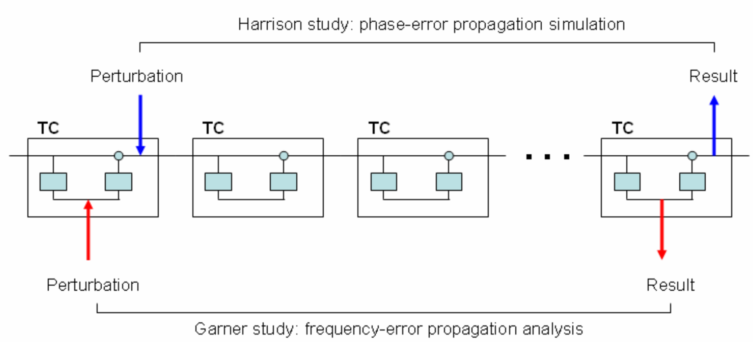
It is possible that the difference in item (1) is related to the fact that the Harrison simulation was executed at the Sync Event time resolution (e.g. 10msec), while the Garner study took advantage of a valid analytical simplification which effectively evaluated system response at the frequency-update period (e.g. 100msec). This discrepancy has not been fully resolved. Item (2) is consistent with item (1), in that both models show a fall-off of *pf* at lower frequencies, and the Harrison study evaluated this factor at a lower frequency.

Item (3) results in the overall gain factor G_0 in eq. 1 differing (by one to three orders of magnitude for the operational conditions in Table 1) between the two studies; the Garner result is smaller and suggests that the cascade effect is negligible in most applications. This difference is primarily due to two different perturbation models used in the respective studies, illustrated below.

Syntonized TC model



TC Cascade: two studies



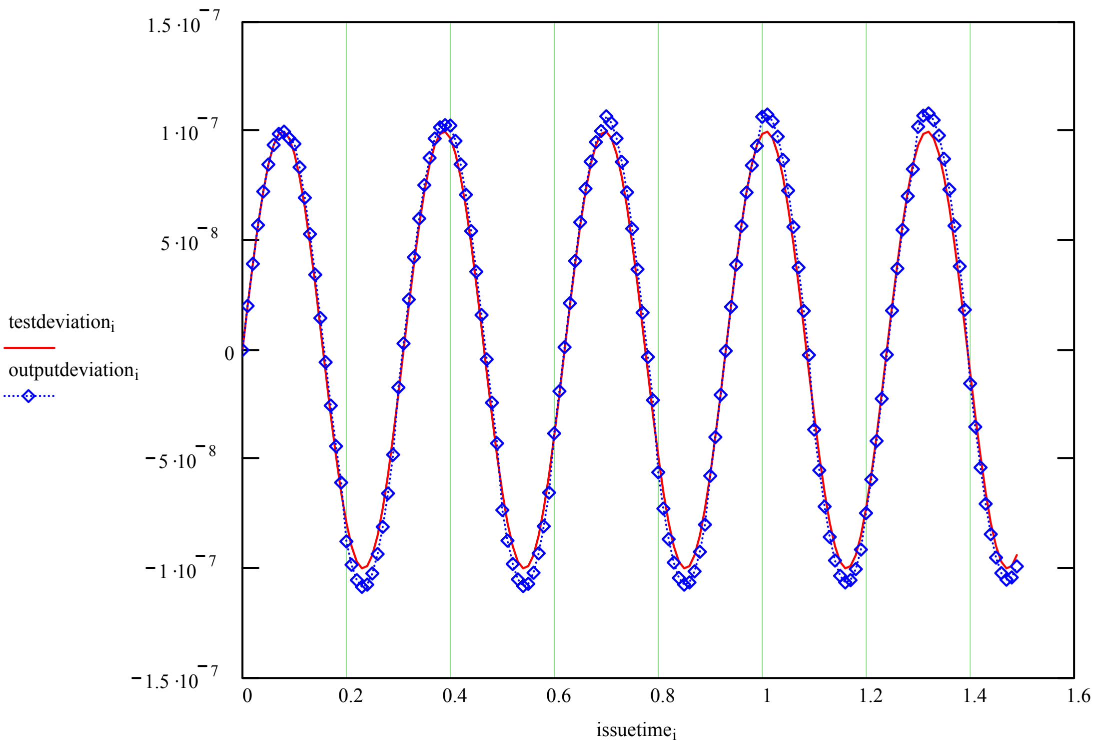
It appears that both analyses are correct; the choice depends on the situation being modeled. The Garner results apply when the error source is frequency variation in the TC local timebase, and the Harrison results apply when the error source is timestamp uncertainty.

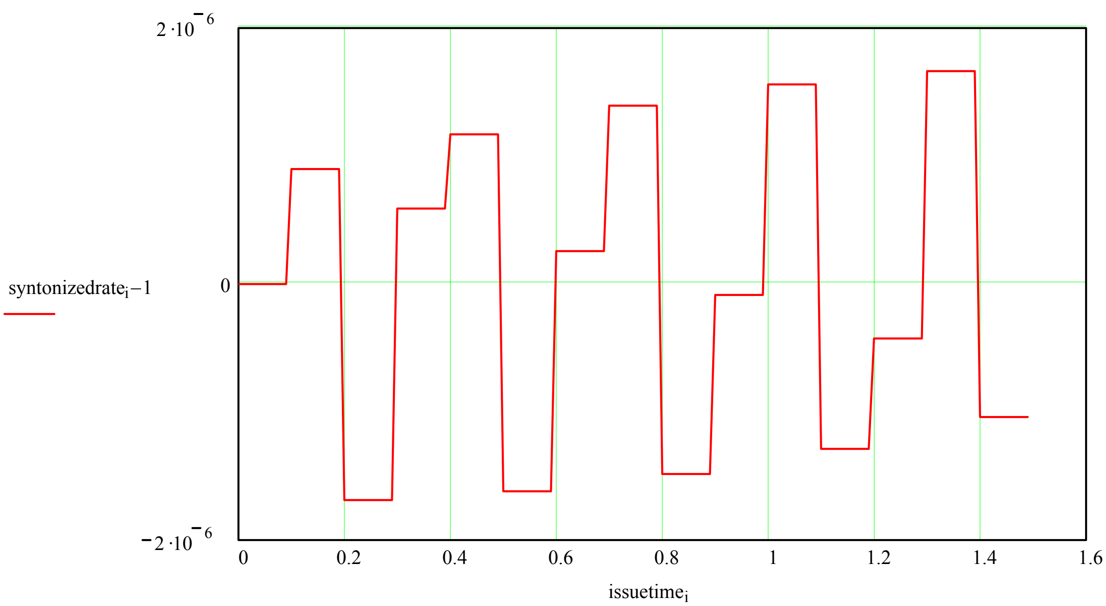
4 Attached study data

The attached initial data from the Harrison study is for simulations performed at T_r/T_I = 0.05 (5msec residence time, 100msec frequency update). This operating condition does not appear in Table 1. The numbers reported in Table 1 were obtained by running the simulation with different values of the <residencetime> parameter in the simulation program.

$$\text{gainpeaking} := \frac{\text{rmsoutputdeviation}}{\text{rmsinputdeviation}} \qquad \text{gainpeaking} = 1.0635 \qquad 20\cdot\log(\text{gainpeaking}) = 0.535 \ \text{dB}$$

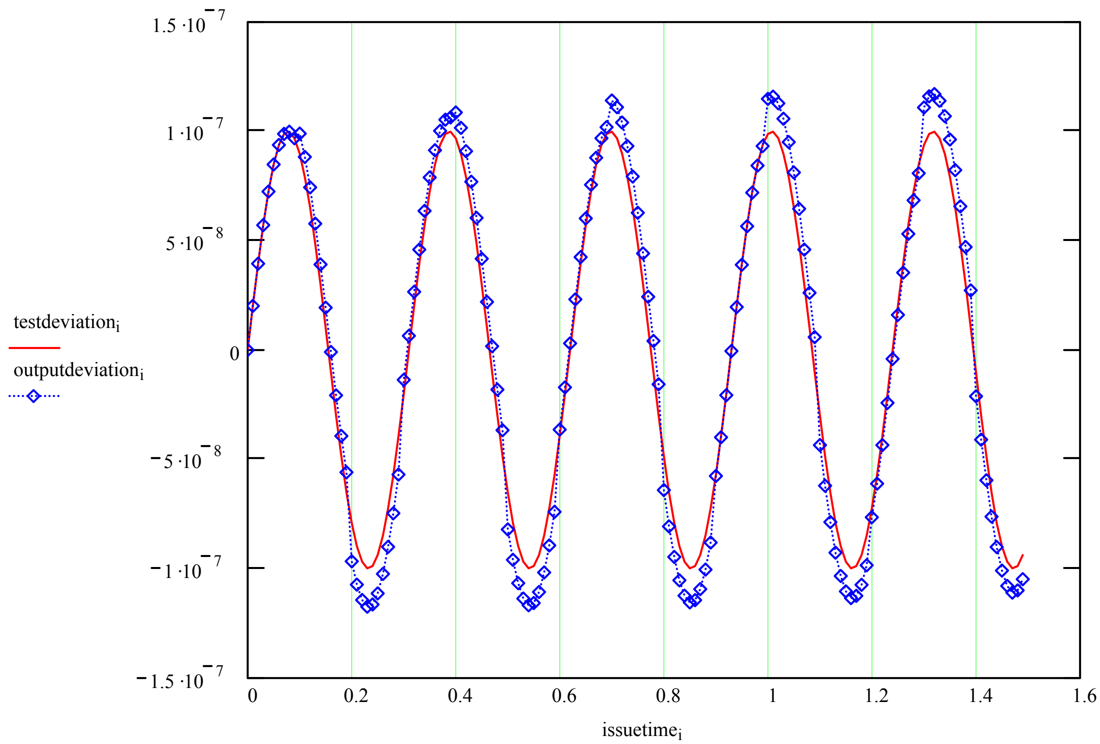
node 3

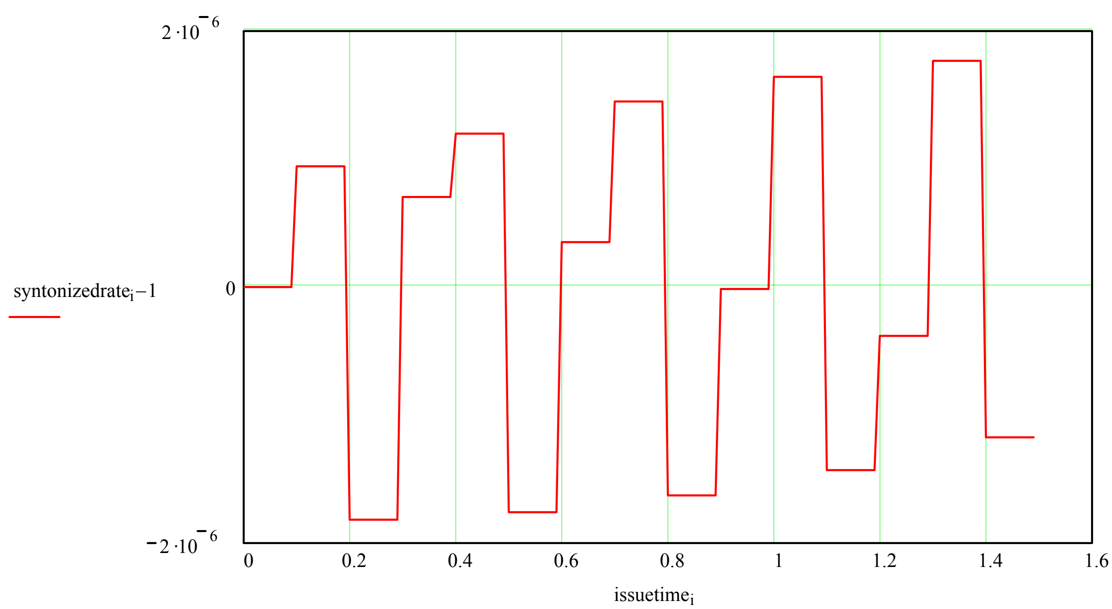




$$gainpeaking := \frac{rmsoutputdeviation}{rmsinputdeviation} \qquad gainpeaking = 1.1352 \qquad 20 \cdot \log(gainpeaking) = 1.101 \text{ dB}$$

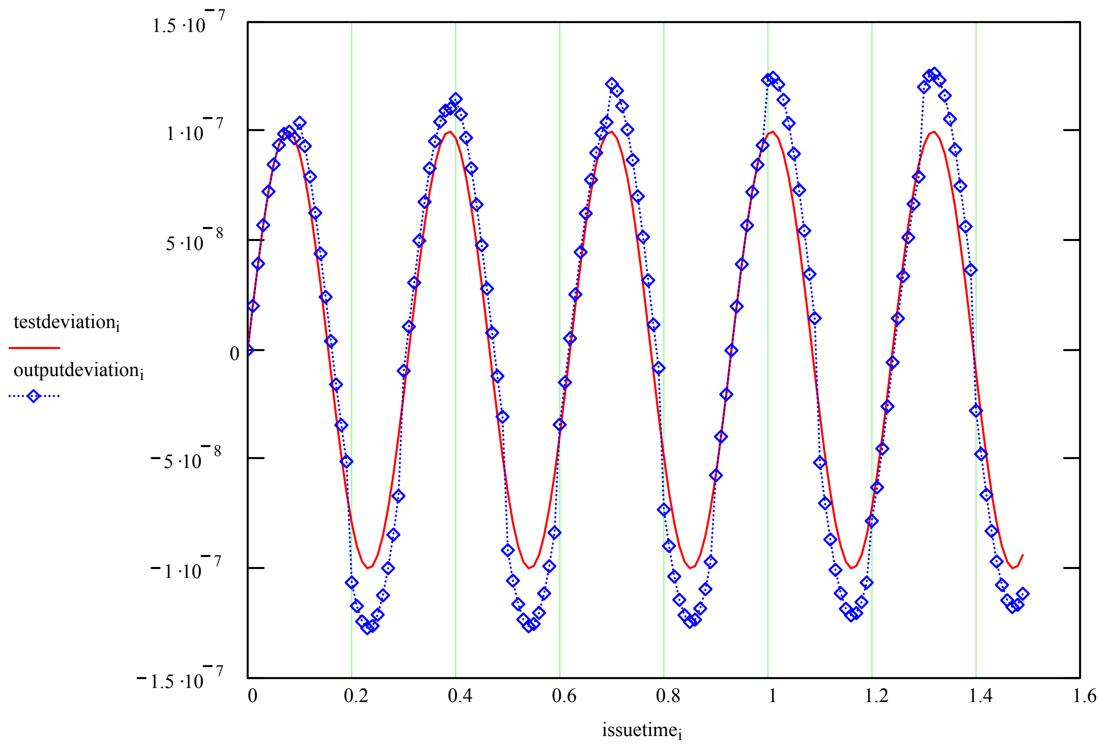
node 4

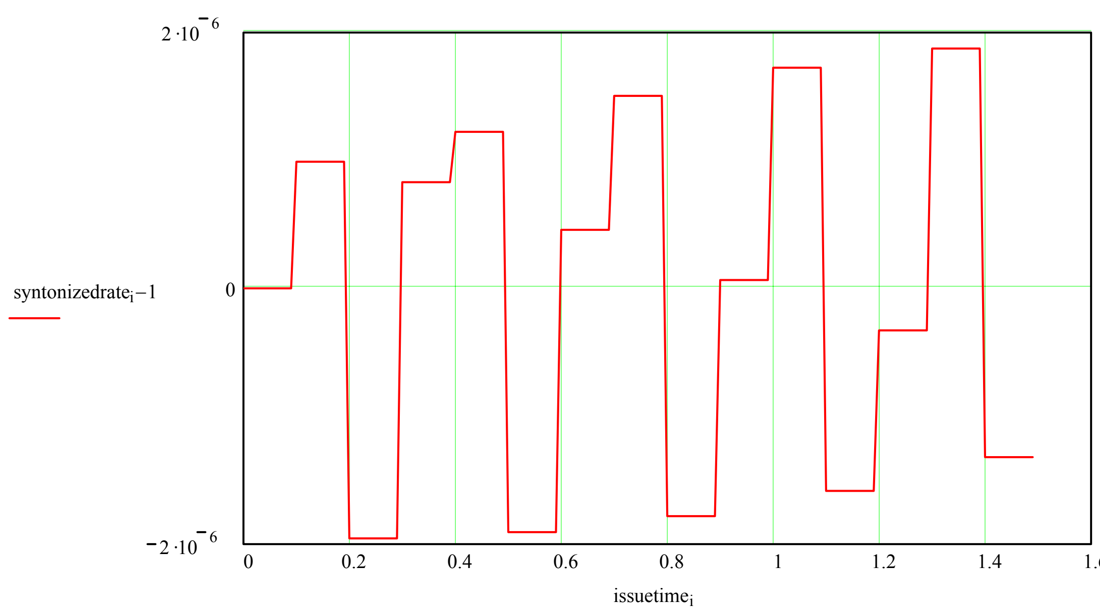




$$gainpeaking := \frac{rmsoutputdeviation}{rmsinputdeviation} \qquad gainpeaking = 1.2153 \qquad 20 \cdot \log(gainpeaking) = 1.694 \ \text{dB}$$

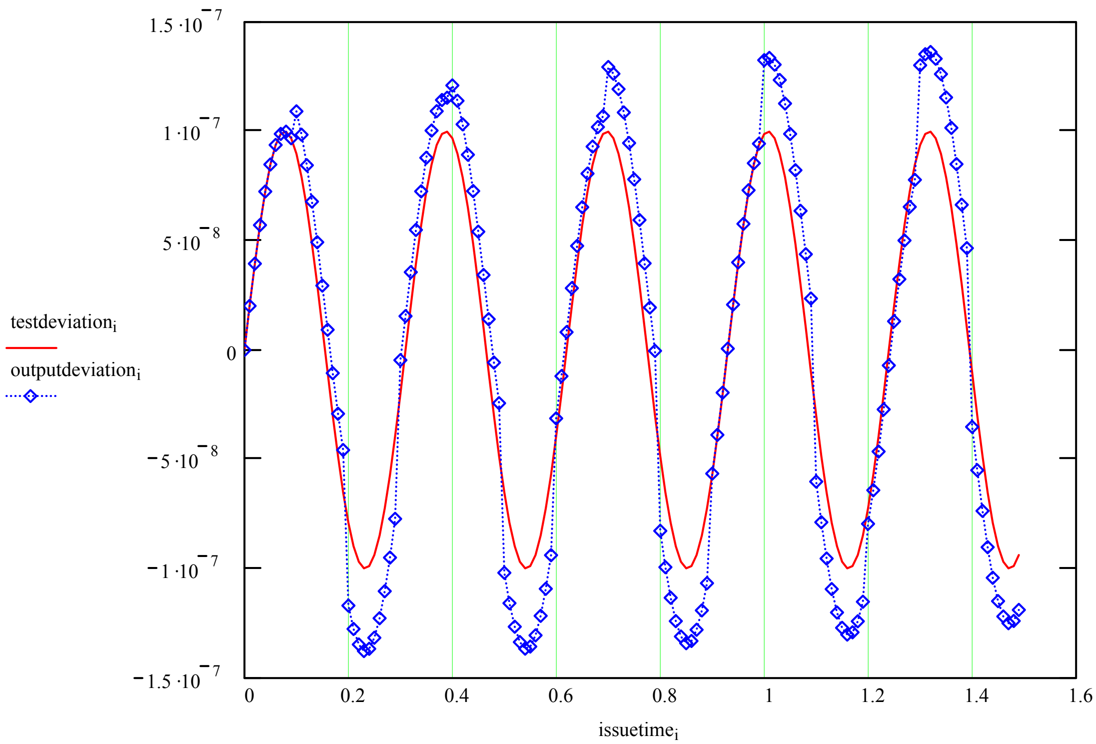
node 5

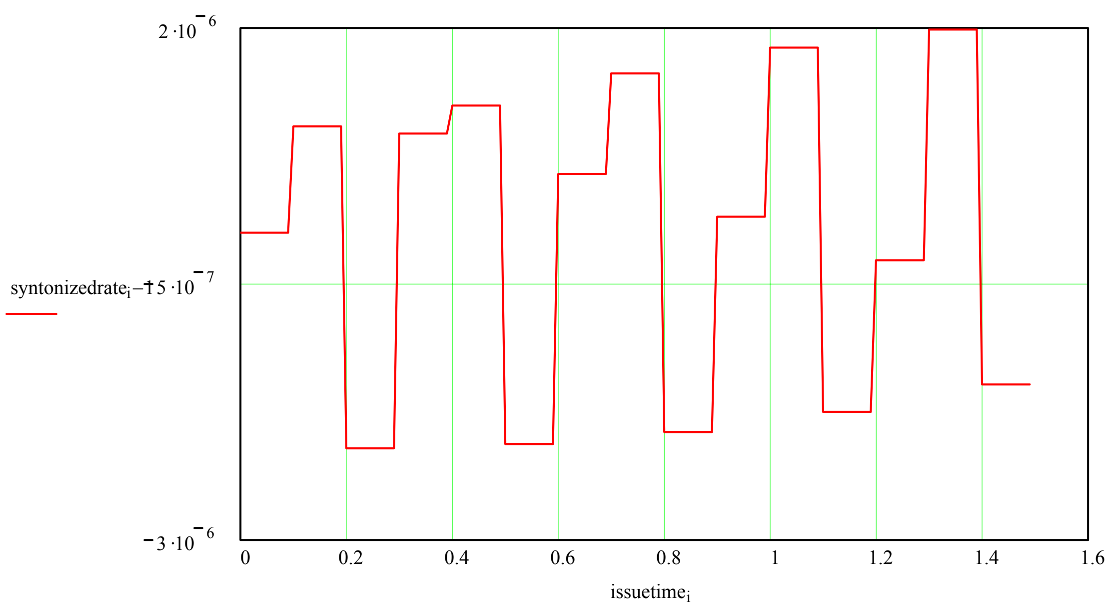




$$\text{gainpeaking} := \frac{\text{rmsoutputdeviation}}{\text{rmsinputdeviation}} \qquad \text{gainpeaking} = 1.3041 \qquad 20\cdot\log(\text{gainpeaking}) = 2.307 \ \text{dB}$$

node 6





Quick check on syntonization gain peaking

Model:
Grandmaster (node 1) is issuing Sync and Follow_up messages on a perfect schedule.
Node 2 is a Transparent Clock which delays each Sync message by exactly 5msec (*no* deviation in actual delay time).
Node 2 experiences some jitter in its internal measurement of residence time, so its (egress time - ingress time) value is not always 5 000 000 nsec. This deviation in measured <residence time> is copied into the correctionField of the Follow_up messages generated by node 2.
For the purposes of this analysis we examine a hypothetical sinusoidal wander component of node 2's measurement deviation.
This simulation examines the behavior of node 3, which receives Sync messages from node 2 precisely 5msec (plus two 10nsec cable delays) after they were issued by the Grandmaster.
Node 3 also receives Follow_up messages from node 2, with a value of approximately 5msec in the correctionField, but with a small superimposed wander on these values due to node 2's measurement deviations
In subsequent iterations, the node's output data are writtento/read from a file to simulate propagation to node 4, node 5, etc.

CMET = corrected master event timestamp
SEIT = sync event ingress timestamp (assumed perfect)

$\mathrm{simsteps} := 150 \qquad i := 0 .. \mathrm{simsteps} - 1$

$\mathrm{syncperiod} := 10 \cdot \mathrm{msec}$

$\mathrm{syntonizesteps} := 10$

$\mathrm{residencetime} := 5 \cdot \mathrm{msec}$ (in this node; assumed constant and perfect)

$\mathrm{cableDelay} := 10 \cdot \mathrm{nsec}$

input sinusoidal wander component applied as test

$\mathrm{testperiod} := 3.1 \cdot \mathrm{syntonizesteps} \cdot \mathrm{syncperiod} \qquad \mathrm{testamplitude} := 100 \cdot \mathrm{nsec}$

Simulation

$\mathrm{issuetime}_i := i \cdot \mathrm{syncperiod}$

$\mathrm{NodeClockOffset} := 0.5 \cdot \mathrm{sec}$ (arbitrary epoch for local ingress and egress timestamps)

$$\mathrm{testdeviation}_i := \mathrm{testamplitude} \cdot \sin\left(2 \cdot \pi \cdot i \cdot \frac{\mathrm{syncperiod}}{\mathrm{testperiod}}\right)$$

$\mathrm{meanpathdelay} := \mathrm{cableDelay}$

enable these lines to generate initial test data

$\mathrm{SyncArrival}_i := \mathrm{issuetime}_i + 5 \cdot \mathrm{msec} + 2 \cdot \mathrm{cableDelay}$ (arrival time of Sync at node 3)

$\mathrm{preciseOriginTimestamp}_i := \mathrm{issuetime}_i$ (Grandmaster is perfect)

$\mathrm{follow\_upCorrectionField}_i := 5 \cdot \mathrm{msec} + \mathrm{cableDelay} + \mathrm{testdeviation}_i$

enable these lines to propagate node to node from file data

$SD := READPRN("simdata.prn")$ SyncEventSentTime, preciseOriginTS, correctionField dataset

$SyncArrival := SD^{\langle 0\rangle} \cdot nsec + cableDelay$ (arrival time of Sync at this node in chain)

$preciseOriginTimestamp := SD^{\langle 1\rangle} \cdot nsec$ (preciseOriginTimestamp from master)

$follow\_upCorrectionField := SD^{\langle 2\rangle} \cdot nsec$ (correctionField from previous node in chain)

$$SEIT_i := SyncArrival_i + NodeClockOffset$$

CMET computed per 12.1.2

$$CMET_i := preciseOriginTimestamp_i + meanpathdelay + follow\_upCorrectionField_i$$

syntonization per 12.1.2.
determine integration interval for calculating syntonized clock rate applied to this sample

$$integrationstart_i := syntonizesteps \cdot \left(floor\left(\frac{i}{syntonizesteps} + .001\right) - 1\right)$$

$$syntonizedrate_i := \begin{cases} 1 & \text{if } i < syntonizesteps \\ \dfrac{CMET_{integrationstart_i + syntonizesteps} - CMET_{integrationstart_i}}{SEIT_{integrationstart_i + syntonizesteps} - SEIT_{integrationstart_i}} & \text{otherwise} \end{cases}$$

$$residencecorrection_i := residencetime \cdot syntonizedrate_i$$

$$idealoutputtime_i := SyncArrival_i + residencetime$$

output follow_up message correctionField adjustments per 11.4.5.1 and 11.5.2.2

$$outputFollow\_upCorrectionField_i := follow\_upCorrectionField_i + meanpathdelay + residencecorrection_i$$

$$outputFollow\_upPreciseOriginTimestamp_i := preciseOriginTimestamp_i$$

Effective output time is what the following node will see when computing its CMET

$$effectiveOutputtime_i := outputFollow\_upCorrectionField_i + outputFollow\_upPreciseOriginTimestamp_i$$

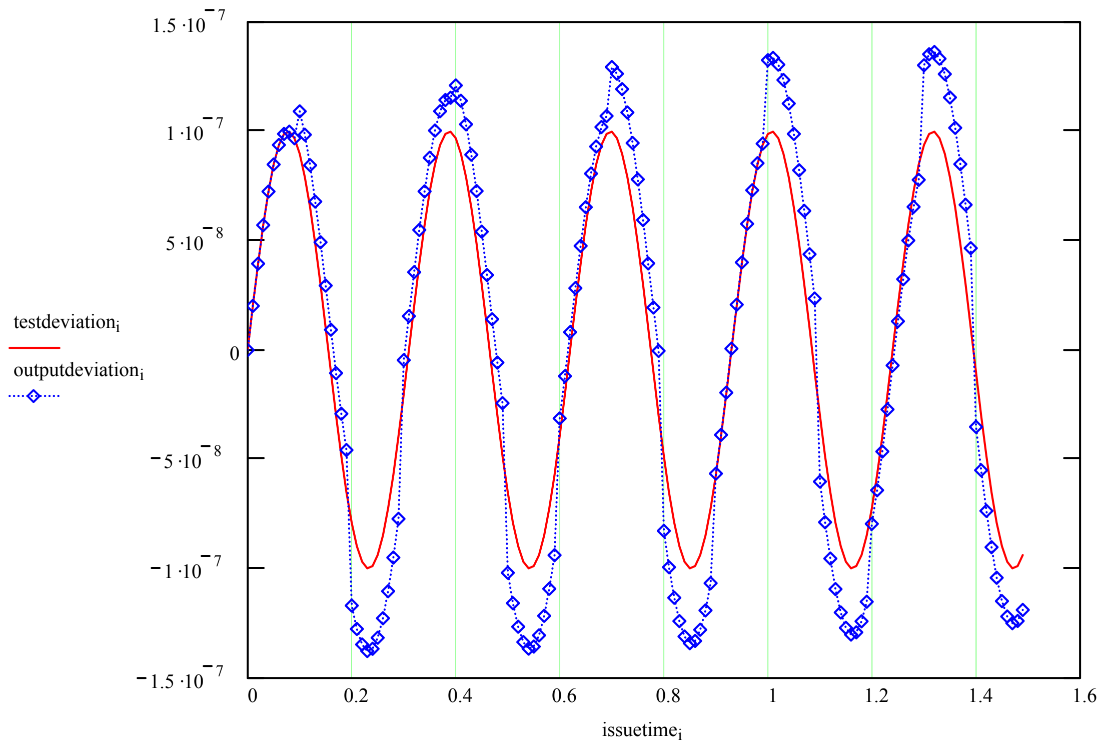
$$outputdeviation_i := effectiveOutputtime_i - idealoutputtime_i$$

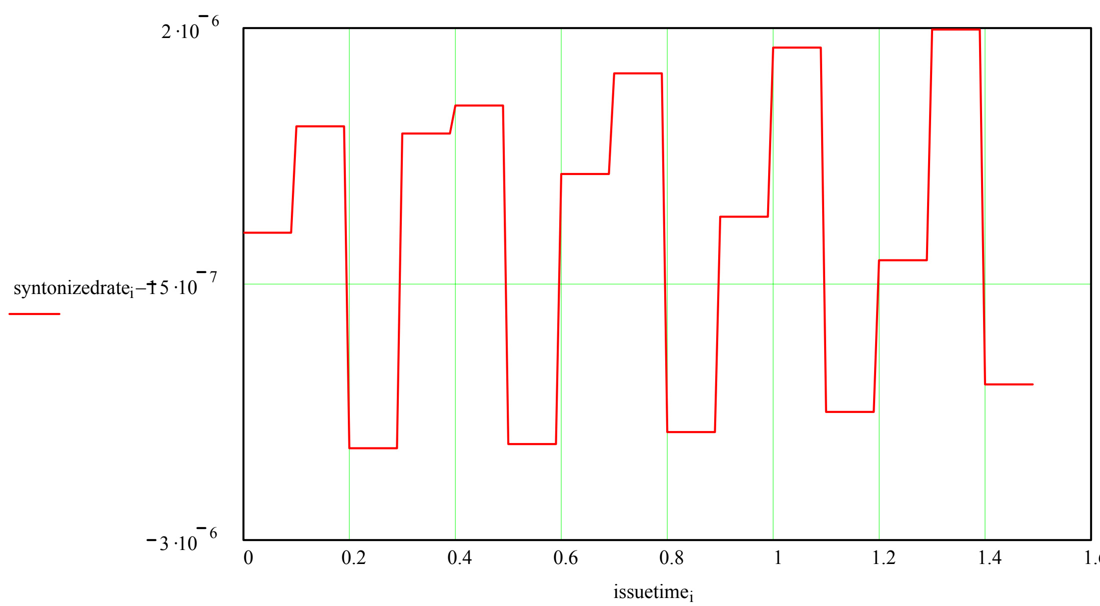
$$rmsinputdeviation := \left[\frac{1}{simsteps} \cdot \sum_i \left(testdeviation_i\right)^2\right]^{\frac{1}{2}}$$ $rmsinputdeviation = 71.055\,nsec$

$$rmsoutputdeviation := \left[\frac{1}{simsteps} \cdot \sum_i \left(outputdeviation_i\right)^2\right]^{\frac{1}{2}}$$ $rmsoutputdeviation = 92.666\,nsec$

$$gainpeaking := \frac{rmsoutputdeviation}{rmsinputdeviation} \qquad gainpeaking = 1.3041 \qquad 20 \cdot \log(gainpeaking) = 2.307 \text{ dB}$$

node 6





write simulation data to file

$$outputSyncIssue_i := SyncArrival_i + residencetime$$

$$SD := augment(outputSyncIssue, outputFollow\_upPreciseOriginTimestamp, outputFollow\_upCorrectionField$$

$$PRNPRECISION := 10$$

$$WRITEPRN(\text{"simdata.prn"}) := SD \cdot nsec^{-1}$$