# Initial Comparison of 802.1AS Jitter/Wander Performance with and without Propagation Time Averaging

**Geoffrey M. Garner**
**SAMSUNG Electronics (Consultant)**

*IEEE 802.1 AVB TG*
2009.05.18

gmgarner@comcast.net

# Outline

- Introduction
- HRM
- Model Summary
- Simulation cases and input
- Results
- Conclusions
- Future work

# Introduction - 1

- Reference [1] contains new simulation results for jitter and wander accumulation for synchronization transport over an 802.1AS network
  - Results are based on current 802.1AS specifications, and consider both current sync and Pdelay interval assumptions (0.01 s and 0.1 s, respectively) and longer sync and Pdelay intervals (0.125 s and 0.250 s for sync interval; 1.0 s and 2.0 s for Pdelay interval)
- These initial results did not consider
  - Local clock/oscillator noise/instability
  - Variability in sync interval, Pdelay interval, Pdelay turnaround time, residence time, and PHY latency
  - Multiple replications to obtain statistics on MTIE

# Introduction - 2

- These initial results indicated
  - No appreciable difference when the sync interval is increased to 0.125 s or 0.250 s, and the Pdelay interval to 1.0 s or 2.0 s, compared to the current assumptions of 0.01 s for sync interval and 0.1 s for Pdelay interval
  - There can be low-frequency (long-period) phase jumps that are multiples of 20 ns, due to 40 ns phase measurement granularity
    - It was suggested that this component of phase error can be reduced using propagation time averaging, as described in [2]
- The current presentation compares the jitter and wander accumulation (i.e., MTIE) for synchronization transport over an 802.1AS network for cases with and without propagation time averaging
- The objective of this work is to see if the use of propagation time averaging can allow a wider bandwidth endpoint filter to be used, for meeting any given application requirement (i.e., any of the MTIE masks)
  - A wider bandwidth endpoint PLL filter would not need as stable an oscillator, and would be lower cost

# Propagation Time Averaging – 1

- Reference [2] suggests two methods of averaging the propagation time
  - Sliding window
  - Digital filter
- In the current presentation, a variation of these methods is used
  - When the system starts up and averaging begins, a growing window is used until a full window of propagation time measurements has been collected
  - Once a full window of measurements is collected, an exponentially-weighted average is use; this is equivalent to a first-order, infinite impulse response digital filter

# Propagation Time Averaging - 2

- Define the following notation
- $M$ = window size (number of Pdelay measurements)
- $d_k$ = Pdelay (propagation time) measurement at time *step* $k$
- $x_k$ = averaged propagation time at time step $k$
- Then the averaged propagation time is computed as

$$x_{k+1} = \frac{kx_k + d_k}{k+1} \qquad k < M$$

$$x_{k+1} = ax_k + (1-a)d_k \qquad k \geq M$$

  - Note that if the quantity $kx_k$ is saved at each time step, a multiplication by $k$ on the next update can be saved
- The second equation is a first order, infinite impulse response (IIR) digital filter
  - It corresponds to the equations on slide 10 of [2], with $n = 1$, $m = 0$, $a_1 = a$, and $b_0 = 1-a$

# Choice of $M$ and $a$

- First, $a$ should be chosen to be consistent with $M$
  - Note that if a sliding window were used, there would only be one parameter, i.e., $M$
  - With the first-order IIR filter, the successive Pdelay measurements are "age-weighted," i.e., the weight of a sample in the average propagation time decreases by a factor of $a$ with each successive time step
  - For the initial simulations here, $a$ is chosen so that the weight of a propagation time measurement is $1/e$ after $M$ samples, i.e.,

$$a = e^{-1/M}$$

  - More generally, $a$ could be chosen so that the weight of a propagation time measurement is $e^{-P}$ after $M$ samples, i.e.,

$$a = e^{-P/M}$$

  - In this more general formulation, we have simply replaced the need to choose $a$ by the need to choose $P$. For the initial simulations here, $P = 1$

# Choice of *M* and *a*

## ❑Choice of *M*

- Ideally, *M* should be large compared to the time scale of the variation of propagation time
- However, the results in [1] indicate that this can be large (e.g., in case 3 of [1] this time scale was greater than 1000 samples or, for 1 s or 2 s Pdelay interval, at least on the order of thousands of seconds)
- In addition, this time scale is, in theory, unbounded, in the sense that the jumps in propagation time measurement due to phase measurement granularity can be separated by an unlimited amount of time for the “wrong” choice of frequency offset between adjacent nodes
  - In practice, this time scale would be bounded due to any frequency offset granularity; nonetheless, the timescale could be quite large
- But, the goal here is to reduce the component of phase error due to propagation time phase jumps, relative to other components of phase error and relative to the MTIE requirements
  - Therefore, we only need to make *M* large enough so that a 40 ns phase jump (worst case when 2 time stamps of the same sign in the Pdelay formula both cross a 40 ns threshold at the same time, and the other 2 time stamps don’t cross the threshold) contributes negligibly to phase error

# Choice of $M$ and $a$

- Choice of $M$ (Cont.)
  - For the initial cases here, we take $M = 1000$

# Model Parameters Common to All Cases – 1

- The same common parameters of [1] are used here
  - In addition, some of the parameters that varied from subcase to subcase in [1] are fixed here
- Endpoint filter gain peaking = 0.1 dB
- Residence time = 1 ms (fixed)
- Pdelay turnaround time = 1 ms (fixed)
- Link propagation delay = 500 ns (fixed)
  - Links are symmetric
  - PHY latency is assumed symmetric
- Phase (time) measurement granularity = 40 ns
- Frequency measurement granularity = 2.328 x $10^{-10}$ (i.e., computations assumed to be done with 32-bit arithmetic)
- No clock noise or clock instability (e.g., constant temperature is assumed)

# Model Parameters Common to All Cases – 2

- Frequencies of free-running oscillators in nodes are chosen randomly at initialization within their tolerance range (a uniform distribution is assumed)
- Sync interval = 0.125 s
- Pdelay interval = 1.0 s
- Free-running oscillator frequency tolerance = $\pm$ 100 ppm
- Simulation time = 10010 s
- Maximum simulation time step = 0.001 s

# Simulation Cases - 1

| Case | Propagation time averaging used | Endpoint filter bandwidth (Hz) |
|---|---|---|
| 1 | No | 0.001 (1 mHz) |
| | Yes | |
| 2 | No | 0.01 (10 mHz) |
| | Yes | |
| 3 | No | 0.1 (100 mHz) |
| | Yes | |
| 4 | No | 1 |
| | Yes | |
| 5 | No | 10 |
| | Yes | |

Note that cases 1 and 2, with no propagation time averaging, correspond to cases 3-125 and 1-125, respectively, in [1]

# Actual Frequency Offsets for Nodes 1 – 8

- The next slide gives the actual frequency offsets used in the simulation cases
  - They were chosen randomly at initialization from a uniform distribution over the tolerance range
  - Therefore, the values depended on the initial state (seed) of the random number generator
  - The same seed was used for all cases/subcases
  - Note that the same seed was used in [1], and therefore the frequency offsets here are the same as in cases 1 and 3 of [1]

# Actual Frequency Offsets for Nodes 1 – 8

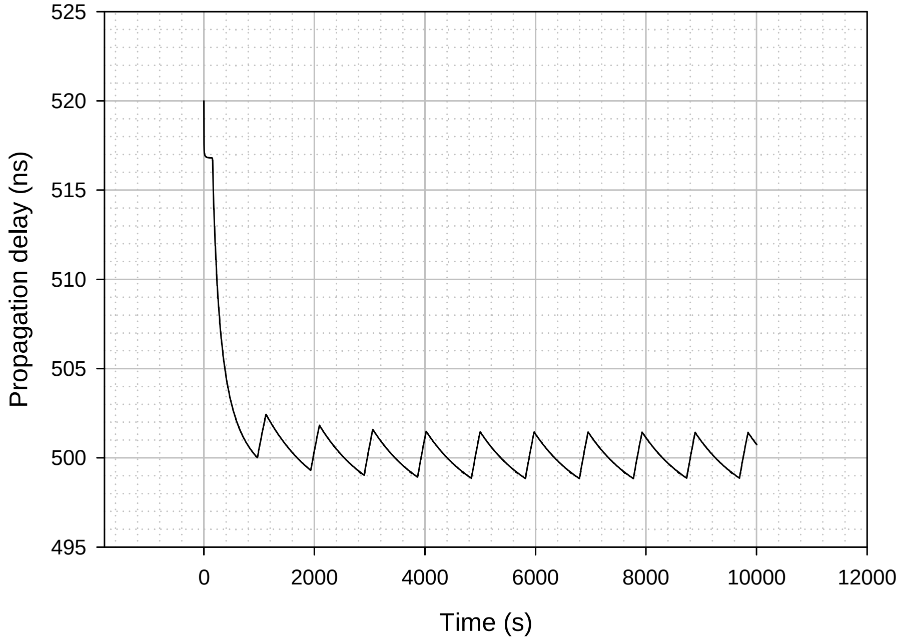
| Node | Frequency offset (ppm) |
|---|---|
| 1 | 0 (GM) |
| 2 | 6.4276 |
| 3 | -55.714 |
| 4 | 32.295 |
| 5 | -53.950 |
| 6 | 38.774 |
| 7 | 64.124 |
| 8 | -83.231 |

# General Notes on Reported Simulation Results

- GM is node 1; final node is node 8
- Phase error (filtered and unfiltered) at GM is 0 (and therefore phase error results are not given for node 1)
- Residence time is not needed (and therefore not computed or shown as a result) at nodes 1 and 8 (GM and final node, respectively)
- Correction field value is not needed (and therefore not computed or shown as a result) at node 1 (GM); its value there is zero
- Propagation time between nodes *j* and *j*+1 is relevant for j = 1, 2, …, 7
- Since the cases differ only in the endpoint filter bandwidths, quantities that do not depend on endpoint filtering (e.g., propagation time, residence time, correction field) are the same for all 5 cases

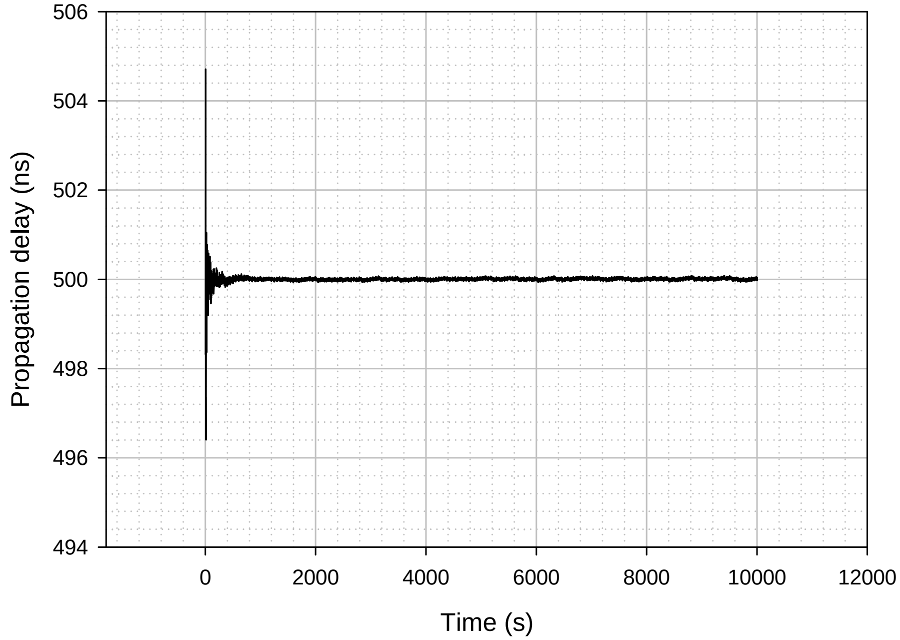
# Node 1 to 2 Propagation Time, Cases 1 – 5

Node 1 to 2 propagation delay, Cases 1- 5
(after initialization)
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s
Endpoint filter BW = 0.01 Hz
Local oscillator tolerance = +/- 100 ppm
(frequency offsets initialized randomly within tolerance range)
propagation time averaging:  window = 1000 Pdelay measurements,
weight = 1/e after 1000 measurements

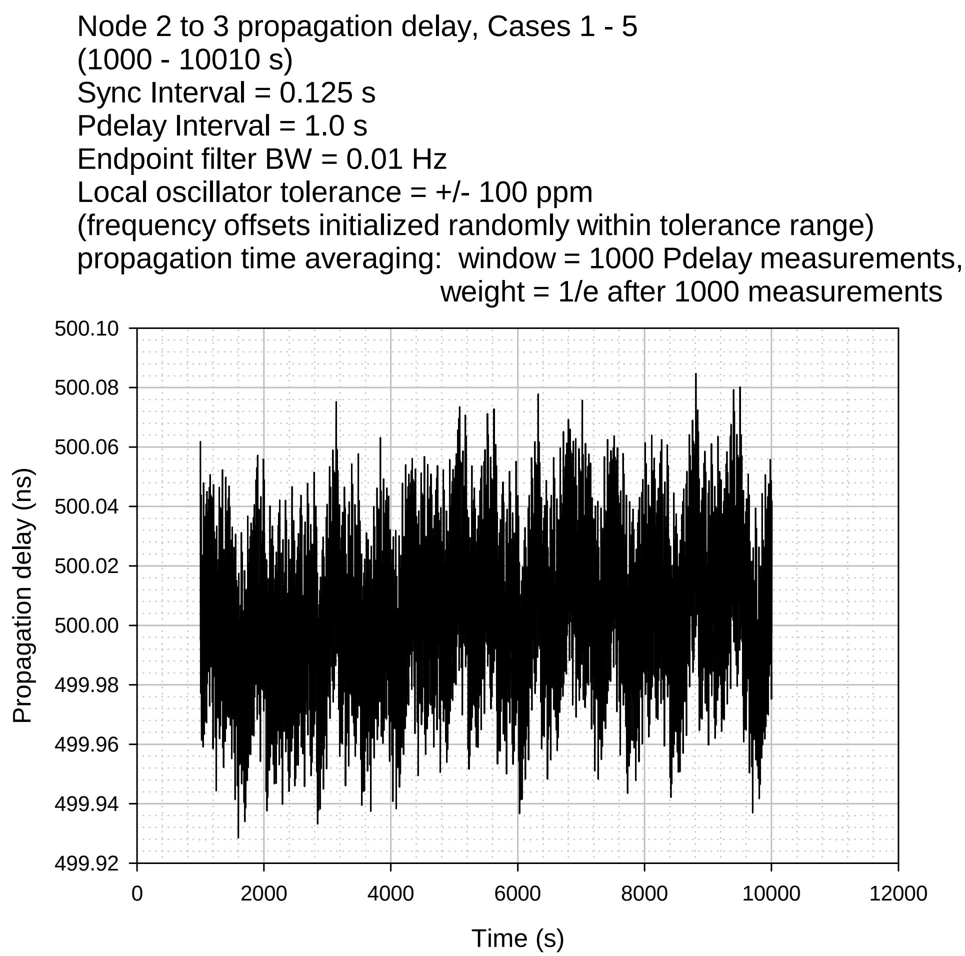


# Node 2 to 3 Propagation Time, Cases 1 - 5

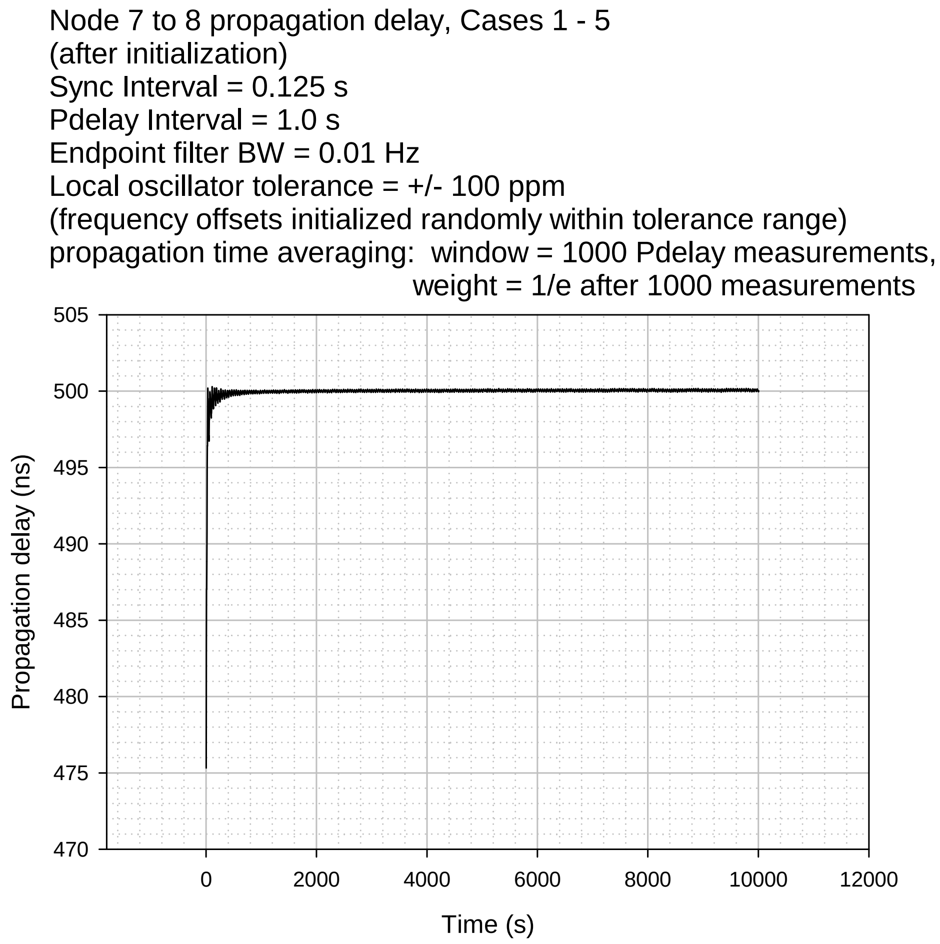
Node 2 to 3 propagation delay, Cases 1 - 5
(after initialization)
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s
Endpoint filter BW = 0.01 Hz
Local oscillator tolerance = +/- 100 ppm
(frequency offsets initialized randomly within tolerance range)
propagation time averaging: window = 1000 Pdelay measurements,
weight = 1/e after 1000 measurements



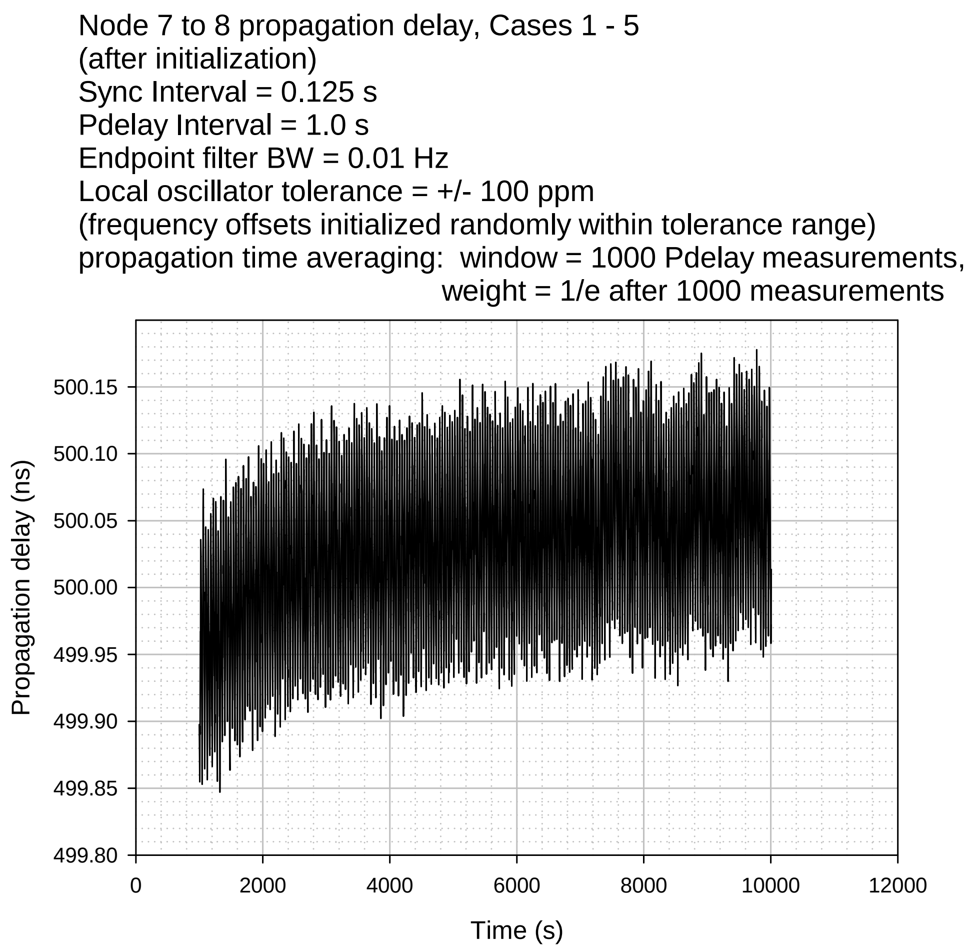
# Node 2 to 3 Propagation Time, 1000 – 10010 s, Cases 1 – 5



# Node 7 to 8 Propagation Time, Cases 1 - 5



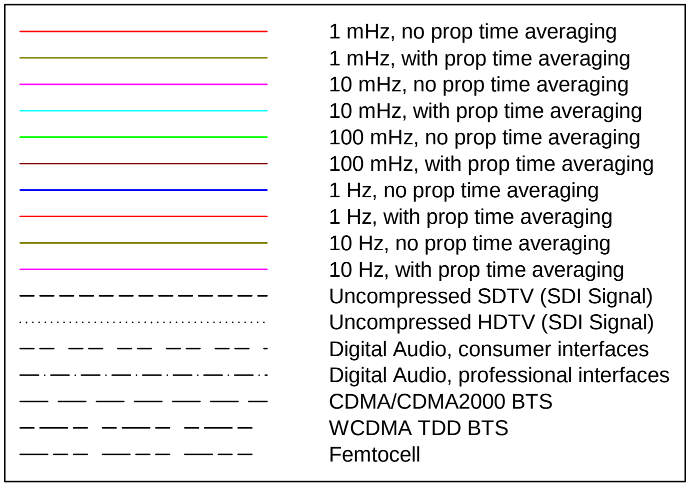
# Node 7 to 8 Propagation Time, 1000 – 10010 s, Cases 1 – 5



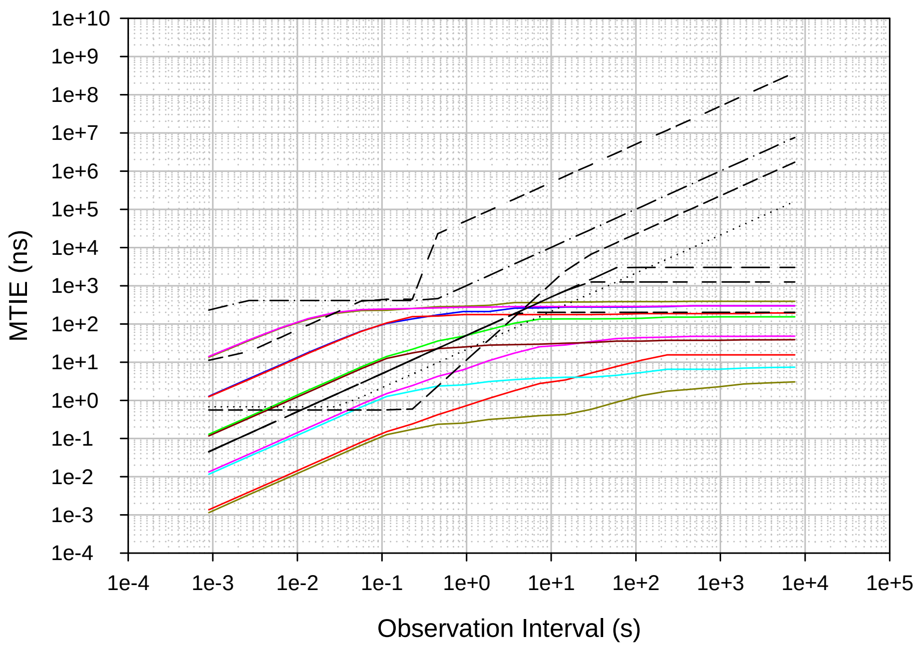
# Cases 1- 5, Results for Average Propagation Time

- The preceding 4 slides show selected averaged propagation times, for nodes 1-2, 2-3, and 7-8
- In all cases, the average propagation time quickly converges to within several ns or less of 500 ns (i.e., of the actual propagation delay in the model)
- The node 1 to 2 average propagation delay shows the largest steady-state variation (after 1000 s), on the order of 3 ns
  - The measured propagation delay for the corresponding case without averaging is given in slide 24 of [1]; the 20 ns jumps there occur with period on the order of 1000 s
- The steady-state variation in averaged propagation delay between nodes 2 and 3, and 7 and 8, is less than 1 ns

# Cases 1 – 5, Node 8 MTIE Results

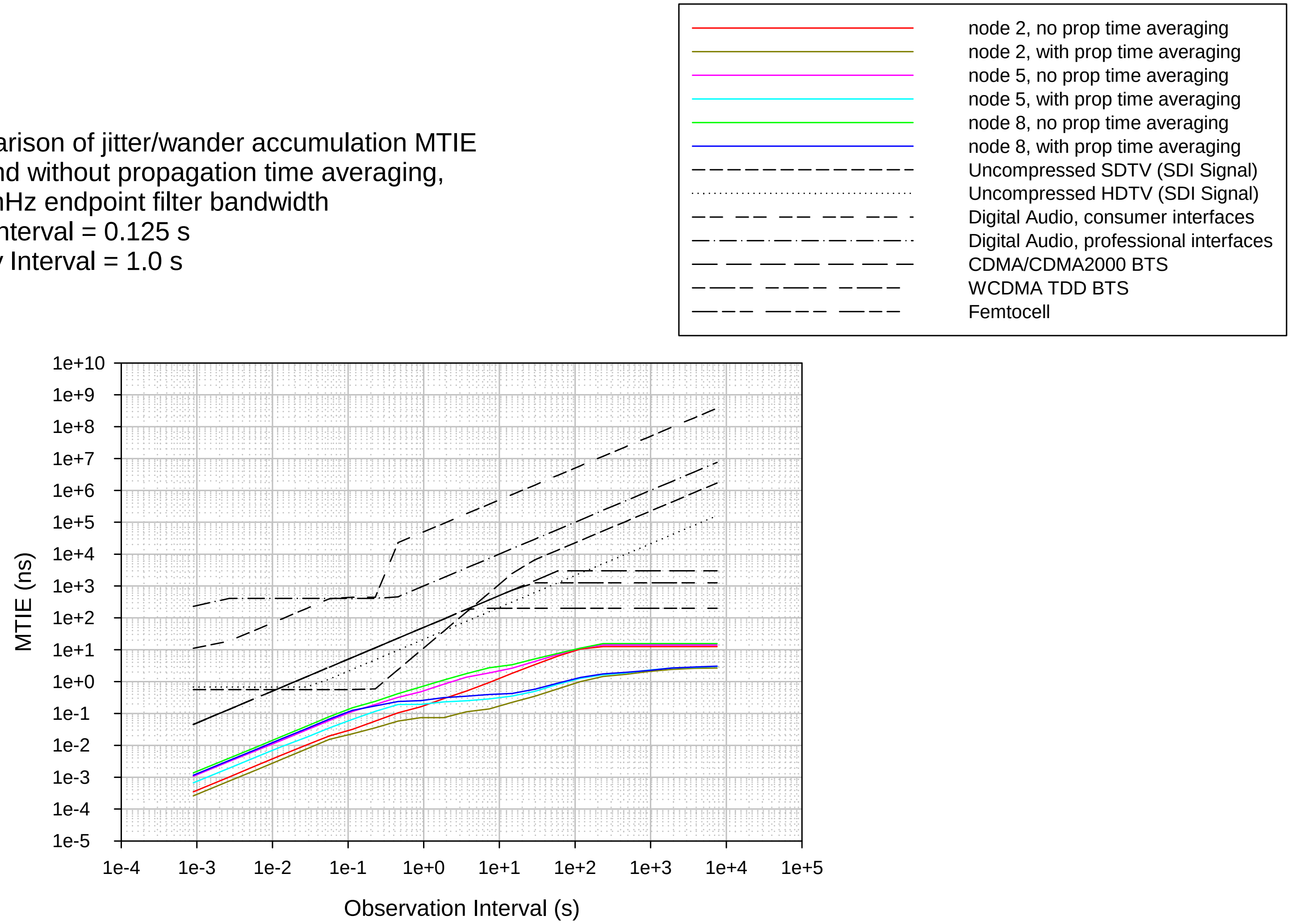



Comparison of jitter/wander accumulation MTIE at node 8 with and without propagation time averaging, for various endpoint filter bandwidths
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



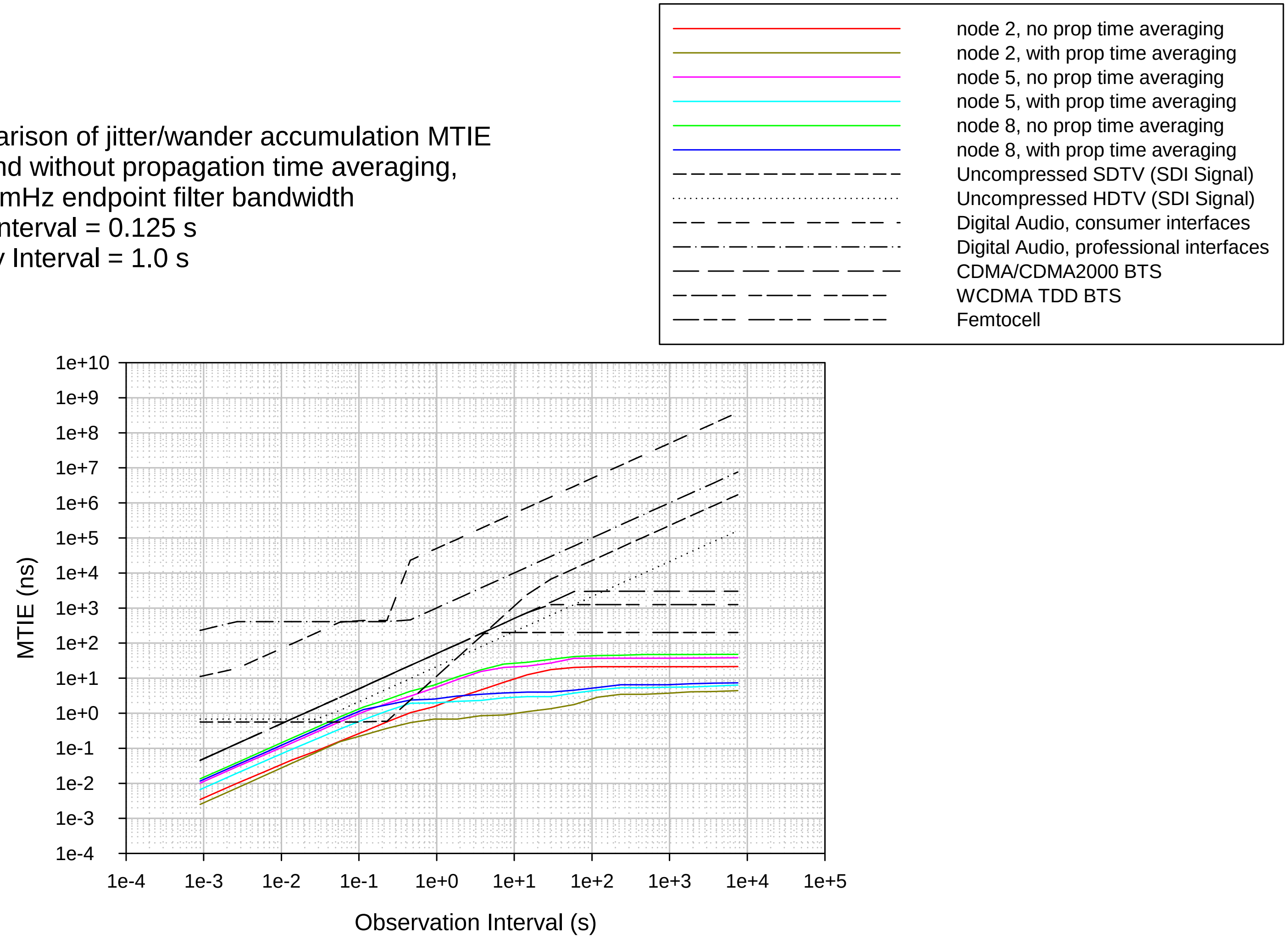
# Case 1 MTIE Results (1 mHz Endpoint Filter)

Comparison of jitter/wander accumulation MTIE
with and without propagation time averaging,
for 1 mHz endpoint filter bandwidth
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



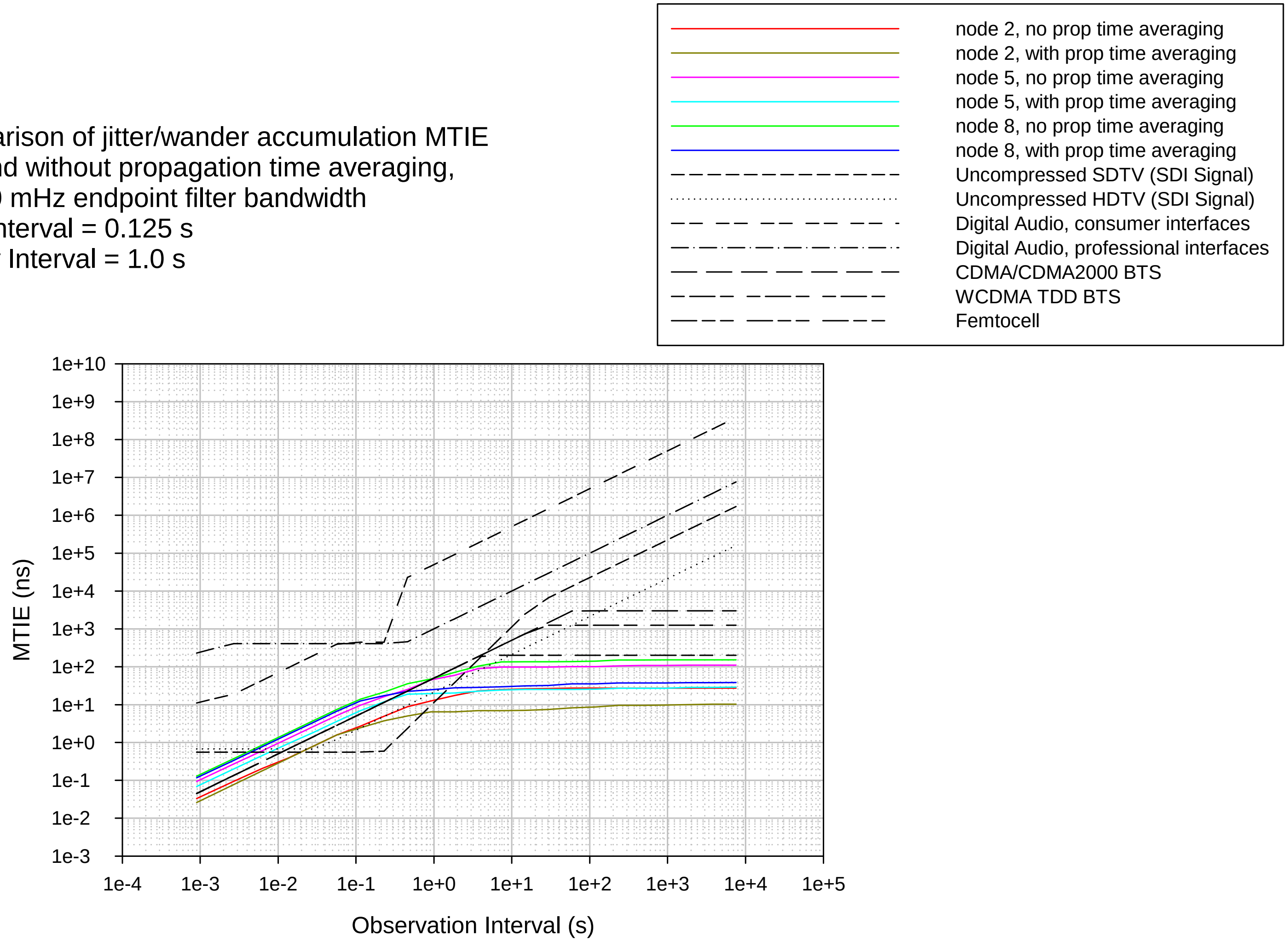
# Case 2 MTIE Results (1 mHz Endpoint Filter)

Comparison of jitter/wander accumulation MTIE
with and without propagation time averaging,
for 10 mHz endpoint filter bandwidth
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



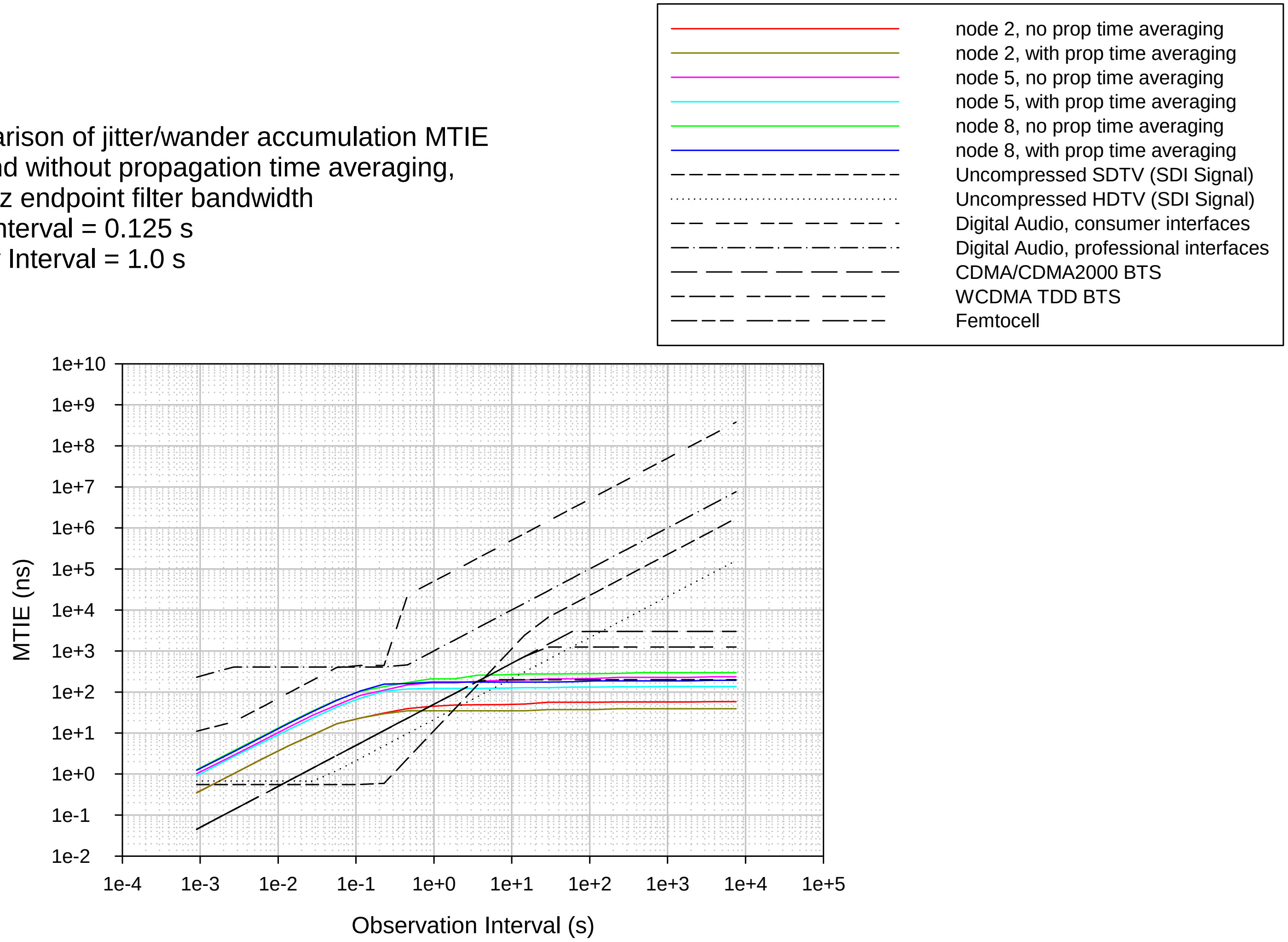
# Case 3 MTIE Results (1 mHz Endpoint Filter)

Comparison of jitter/wander accumulation MTIE
with and without propagation time averaging,
for 100 mHz endpoint filter bandwidth
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



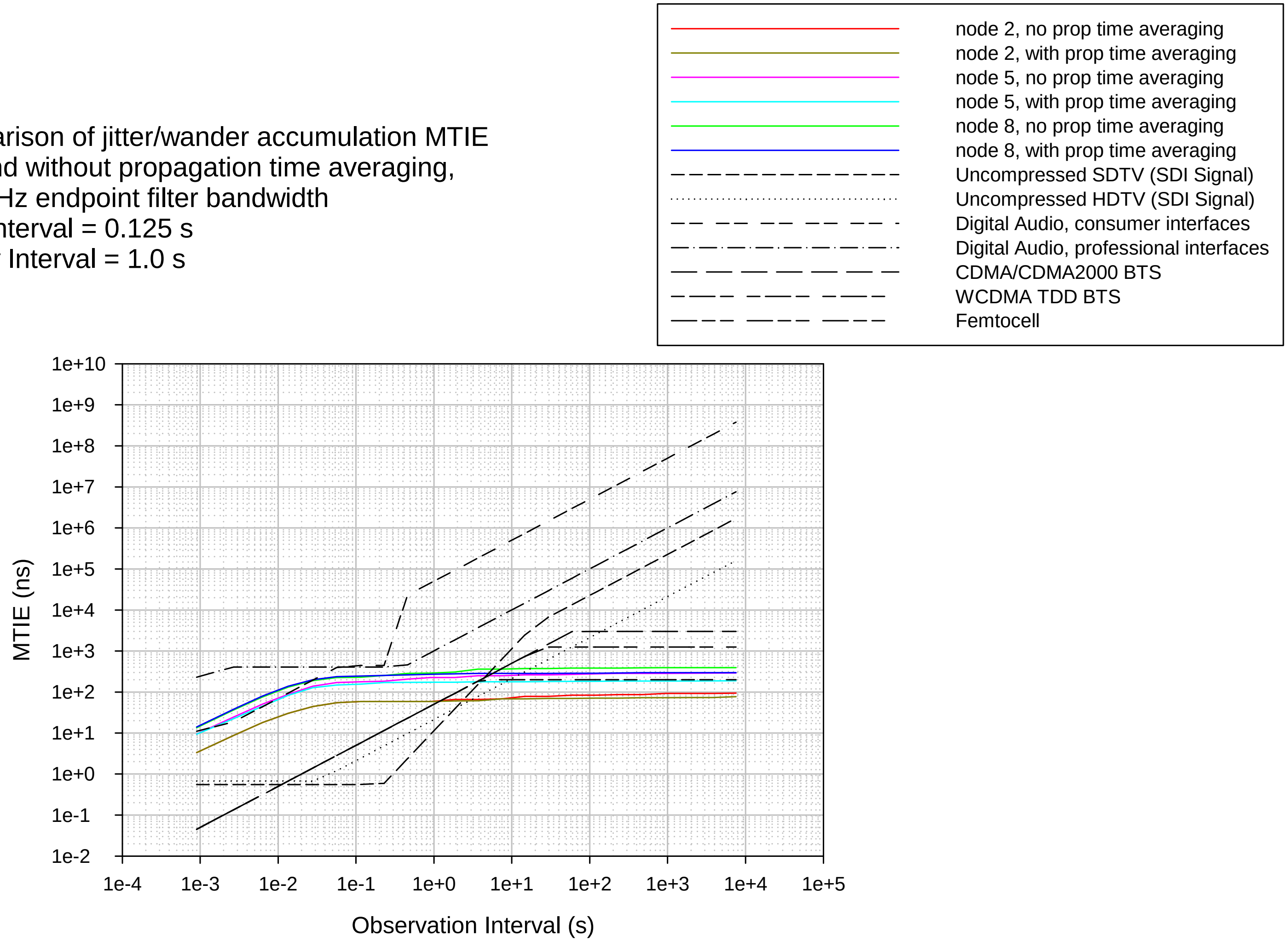
# Case 4 MTIE Results (1 mHz Endpoint Filter)

Comparison of jitter/wander accumulation MTIE
with and without propagation time averaging,
for 1 Hz endpoint filter bandwidth
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



# Case 5 MTIE Results (1 mHz Endpoint Filter)

Comparison of jitter/wander accumulation MTIE
with and without propagation time averaging,
for 10 Hz endpoint filter bandwidth
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s



# Cases 1 – 5 Jitter/Wander (MTIE) Results - 1

- The results on slide 22, for node 8, show that the decrease in MTIE due to propagation time averaging mainly occurs at longer observation intervals
- For 1 mHz filtering, the reduction is on the order of a factor of 5 (i.e., approximately 15 ns without propagation time averaging, and 3 ns with averaging, at long observation intervals
- For 10 Hz filtering, MTIE at long observation intervals is on the order of 391 ns without averaging, compared to 298 ns with averaging
  - The reduction is 93 ns, but a smaller fractional decrease (i.e., approximately 25% reduction)
- For each filter bandwidth, the MTIE curves with and without averaging start together at small observation intervals, and then separate between approximately 0.1 s and 1 s
  - This effect is also observed at each successive node, in the subsequent plots for the individual cases

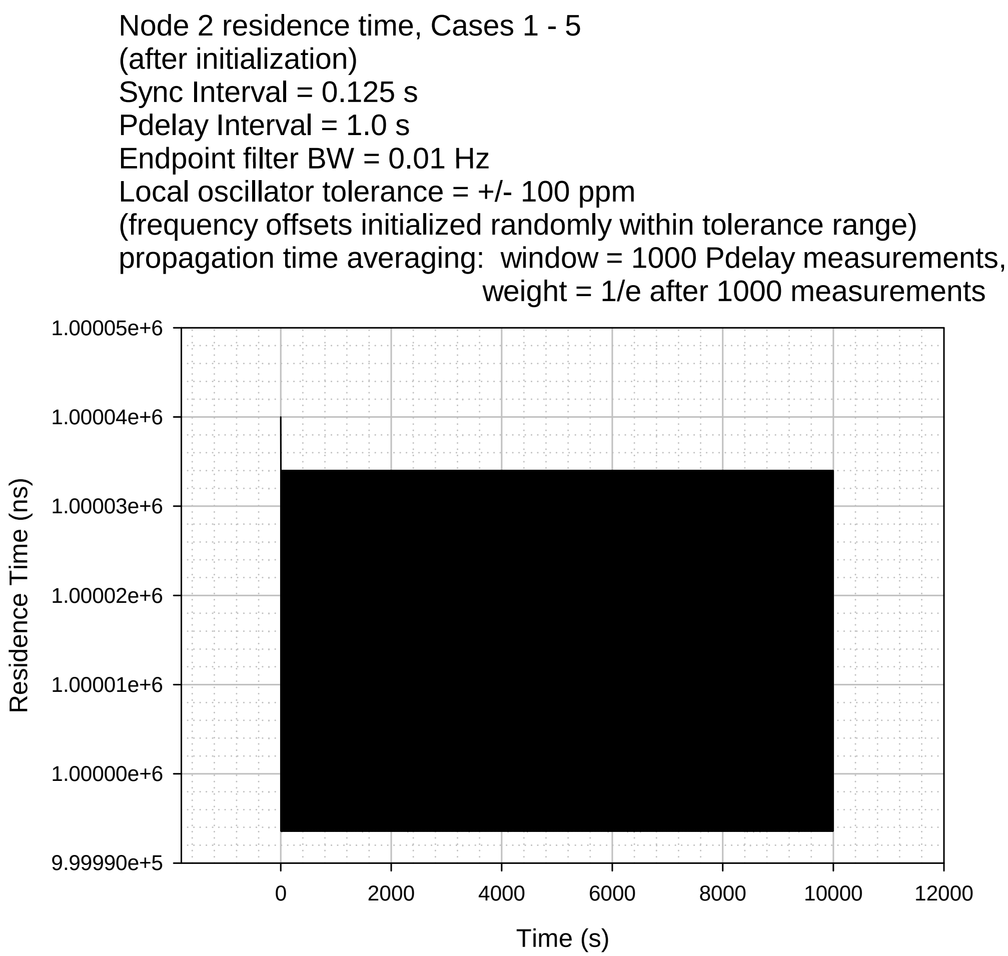
# Cases 1 – 5 Jitter/Wander (MTIE) Results - 2

- However, the results also indicate that only in case 1, with 1 mHz filtering, are all the MTIE masks met
  - This is true with and without propagation time averaging, though the propagation time averaging does reduce the wander for observation intervals greater than approximately 0.5 s
- Furthermore, the same is true if one considers only meeting the less stringent masks; for example
  - Case 2 (10 mHz) meets the cellular base station, digital audio, and HDTV masks, but not the SDTV mask; cases 3, 4, and 5 (100 mHz, 1 Hz, 10 Hz) do not meet these masks
    - This is true both with and without propagation time averaging (though, again, the propagation time averaging does improve the wander at the longer observation intervals)
  - Case 4 (1 Hz) meets the digital audio masks but not the cellular base station or video masks; case 5 (10 Hz) does not meet the consumer audio mask
    - This is true both with and without propagation time averaging (though, again, the propagation time averaging does improve the wander at the longer observation intervals)

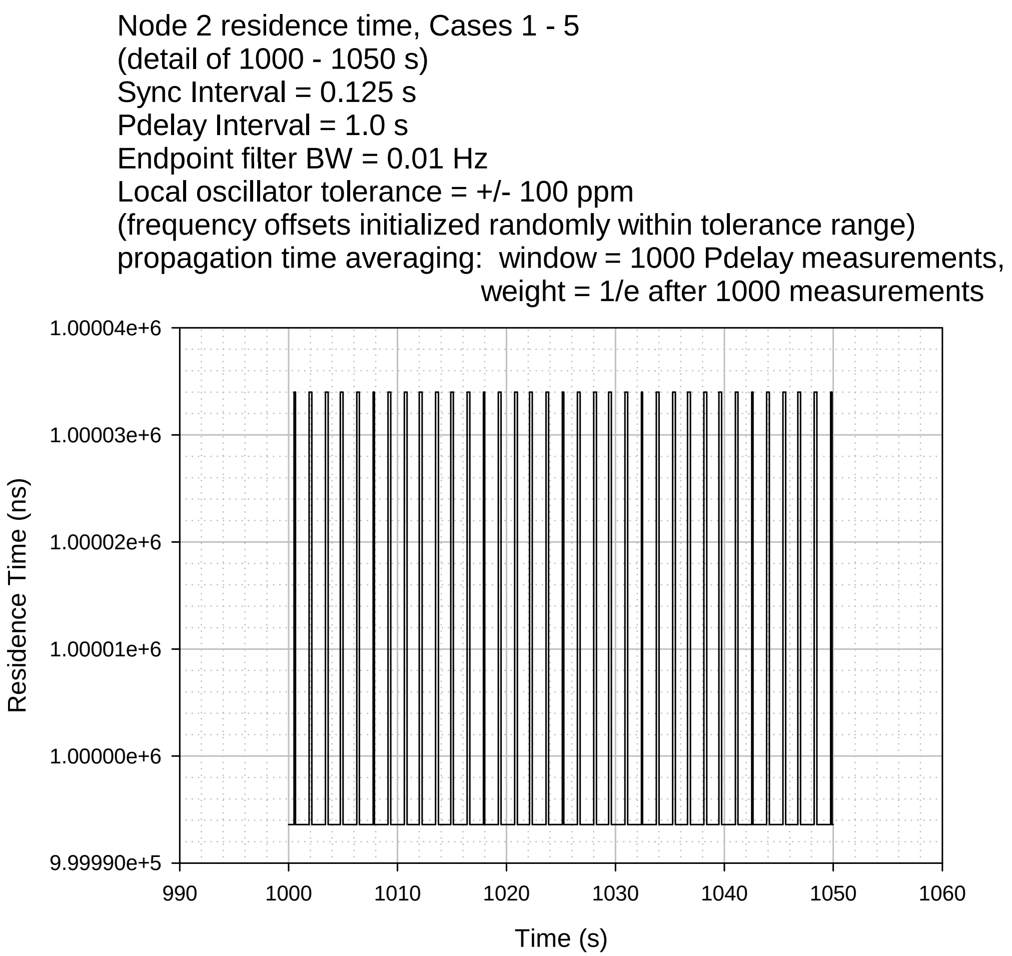
# Cases 1 – 5 Jitter/Wander (MTIE) Results - 3

- The earlier slides that showed the results for average propagation time indicated that the averaging does reduce the propagation time variation to several ns or less
- The MTIE results showed much higher levels of phase variation, especially for the wider-bandwidth endpoint filter cases
  - This phase variation is due to the variation in residence time (as this is the only other component of phase error
- Therefore, we next consider residence time variation
  - Results for residence time and correction field variation are given in the following slides and the backup slides

# Cases 1 – 5, Node 2 Residence Time

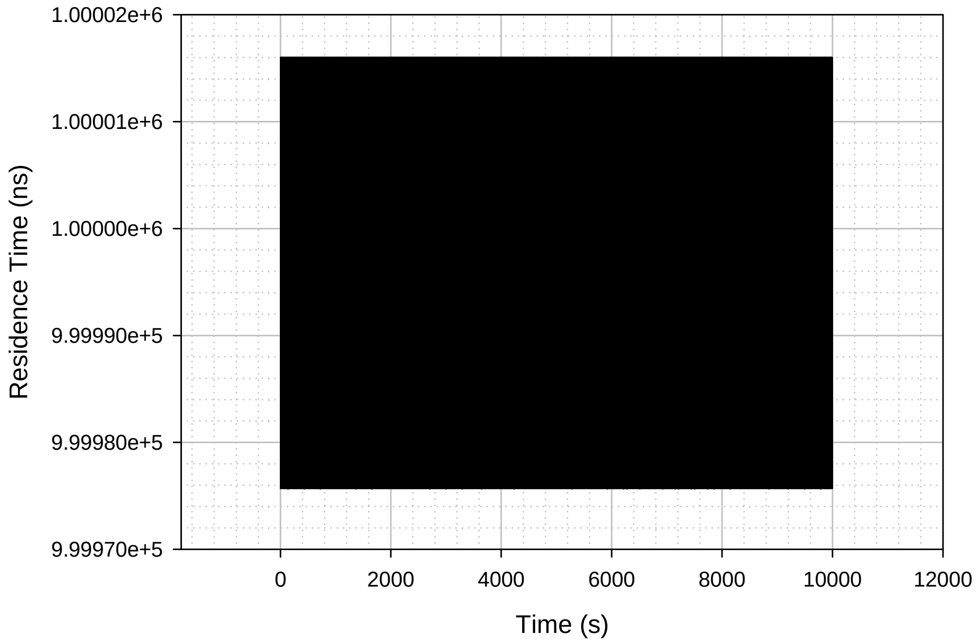


# Cases 1 - 5, Node 2 Residence Time (Detail of 1000 - 1050 s)

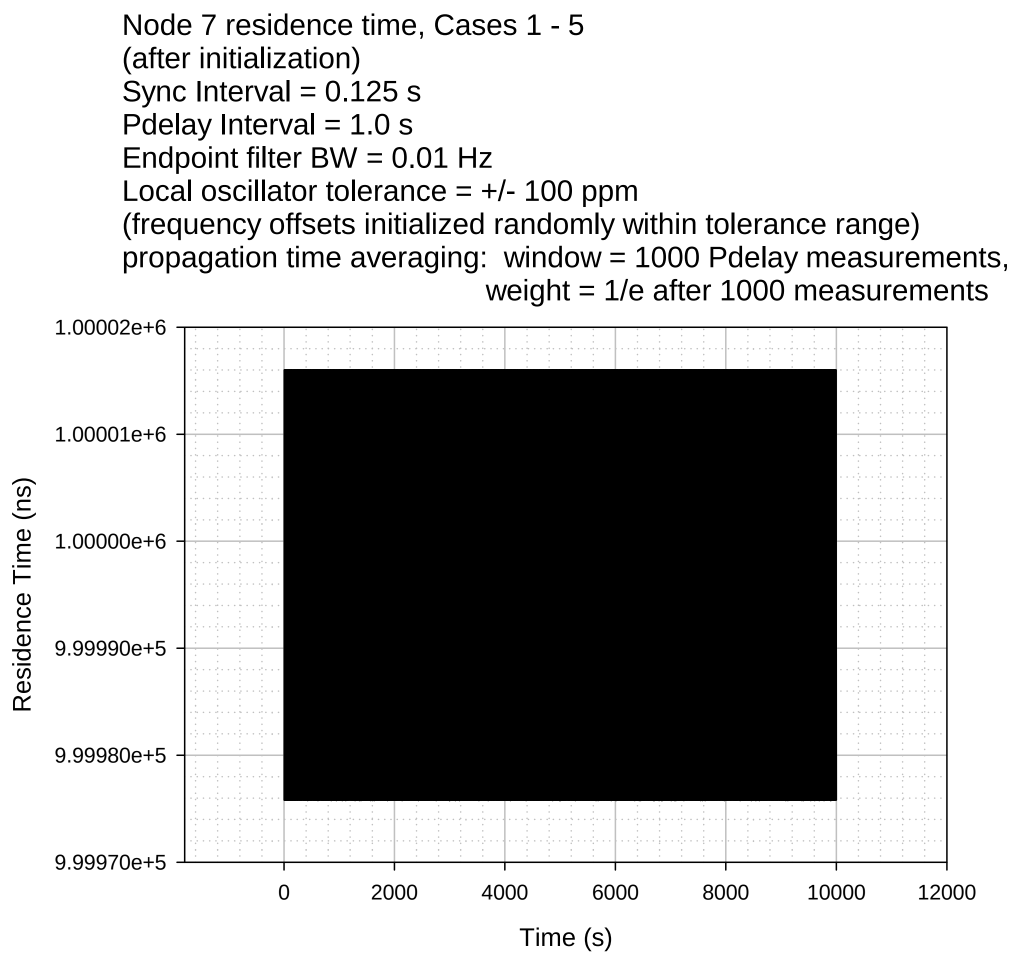


# Cases 1 – 5, Node 3 Residence Time

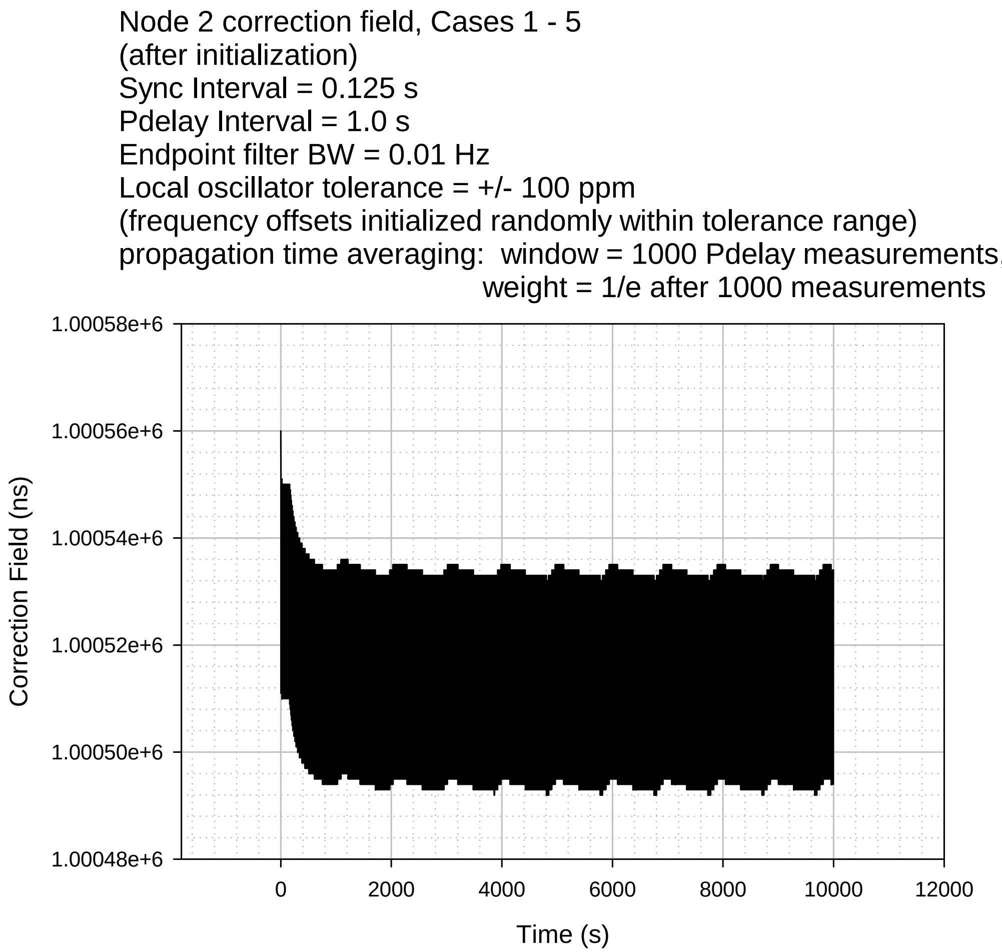
Node 3 residence time, Cases 1 - 5
(after initialization)
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s
Endpoint filter BW = 0.01 Hz
Local oscillator tolerance = +/- 100 ppm
(frequency offsets initialized randomly within tolerance range)
propagation time averaging: window = 1000 Pdelay measurements,
weight = 1/e after 1000 measurements



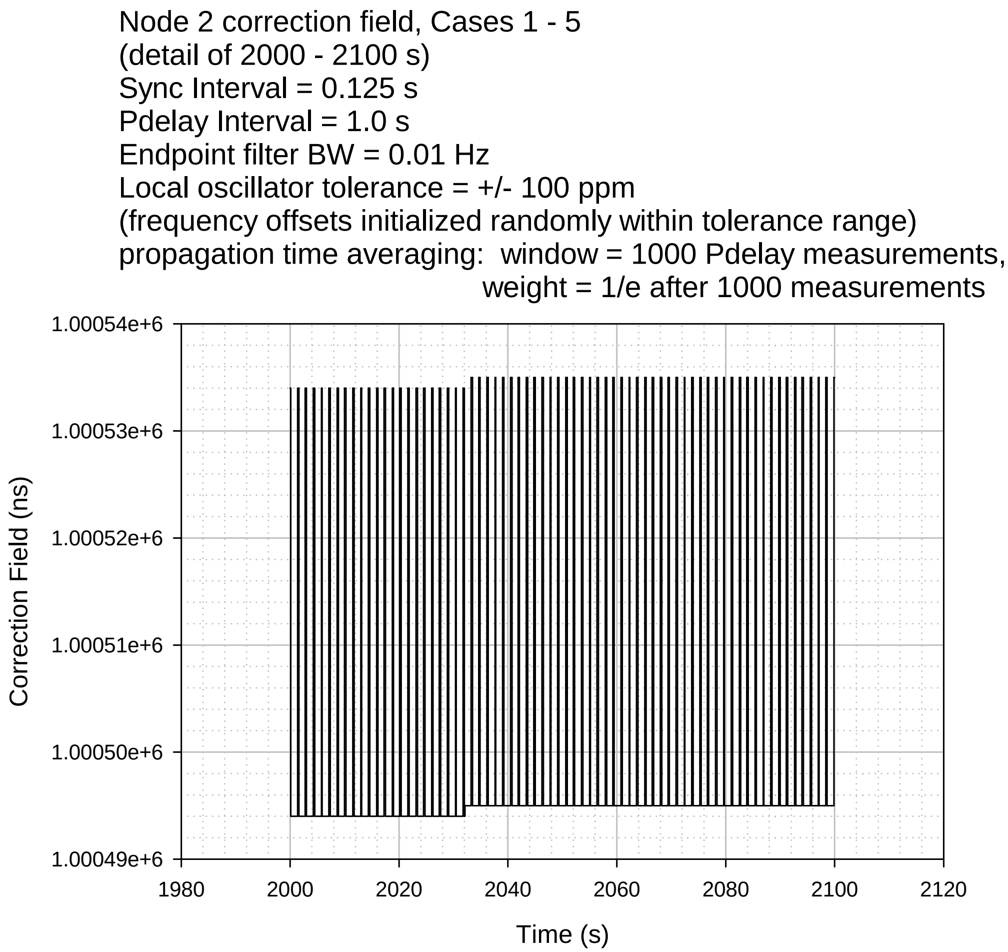
# Cases 1 – 5, Node 7 Residence Time



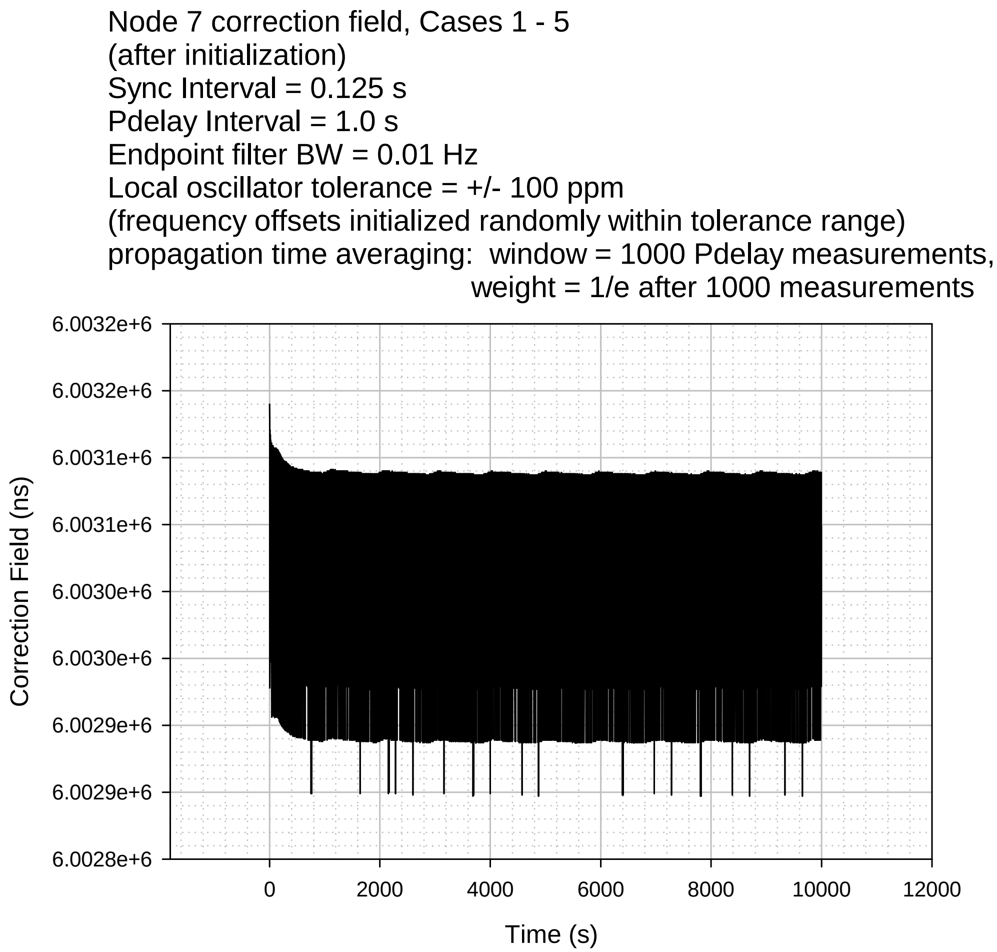
# Cases 1 – 5, Node 2 Correction Field



# Cases 1 - 5, Node 2 Correction Field (Detail of 2000 - 2100 s)



# Cases 1 – 5, Node 7 Correction Field



# Cases 1 – 5, Residence Time and Correction Field Results – 1

- The preceding slides, and the backup, show a residence time variation of 40 ns at each node
- This is due to truncation of the ingress and egress time stamps for the respective Sync messages
  - If the ingress time stamp crosses a multiple of 40 ns and the egress time stamp does not, the residence time experiences a step decrease of 40 ns
  - If the egress time stamp crosses a multiple of 40 ns and the ingress time stamp does not, the residence time experiences a step increase of 40 ns
- The effect appears to be relatively high frequency, and therefore should be easily reduced by digital low-pass filtering
  - This will be considered in future work
- Results are also shown for the correction field at the successive nodes
  - These show the accumulation of propagation times and residence times at the successive nodes

# Cases 1 – 5, Residence Time and Correction Field Results – 1

- Note that propagation time averaging and digital low-pass filtering of the residence time do not introduce gain peaking effects
- This is because the propagation time between a pair of nodes does not affect any downstream propagation time or residence time measurements, and the residence time at a node does not affect any downstream propagation time or residence time measurements

# Summary and Conclusions – 1

- Propagation time averaging does significantly decrease the propagation time variation
  - The variation in averaged propagation time is at most a few ns, and in many cases less then 1 ns
- While propagation time averaging does significantly decrease MTIE for longer observation intervals (i.e., longer than approximately 0.1 to 1 s, when the propagation time averaging window is 1000 samples and the Pdelay interval is 1 s), the decrease in MTIE is negligible for shorter observation intervals
  - The result of this is that, while the longer-term wander performance is improved, propagation time averaging by itself does not result in any of the MTIE requirements being met that were not met without averaging
- The results for residence time suggest that digital low-pass filtering of the residence time will reduce MTIE at all observation intervals
  - This may result in one or more MTIE masks being met that were not met without residence time filtering, for the same endpoint filter bandwidth

# Future Work

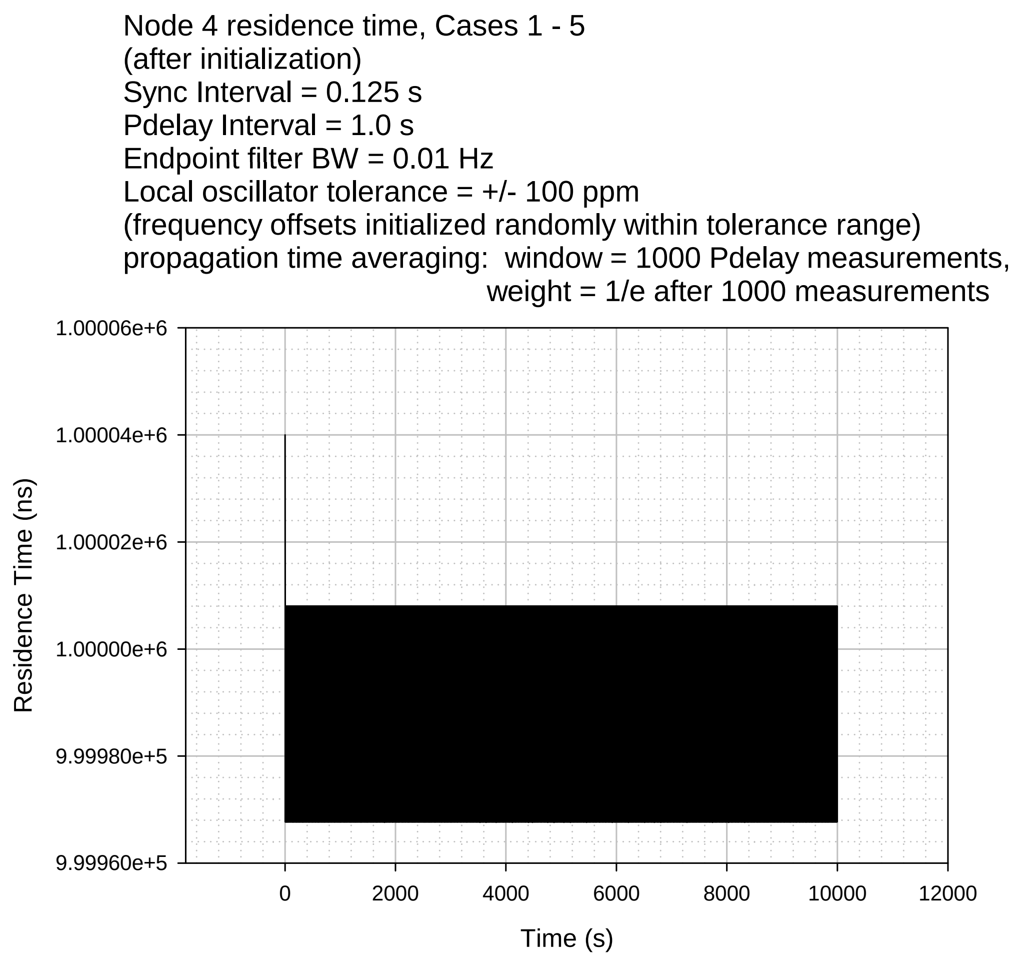
- Future simulations will look at the effect of digital low-pass filtering the residence time, and possible reduction in jitter/wander accumulation (MTIE)

# References - 1


1. Geoffrey M. Garner, *Initial Simulation Results for 802.1AS Synchronization Transport with Longer Sync and Pdelay Intervals*, Samsung presentation to May, 2009 IEEE 802.1 AVB TG Meeting, Pittsburgh, PA, USA, May 11, 2009.
2. Geoffrey M. Garner, *Improvement of Jitter, Wander, and Time Synchronization Performance in 802.1AS Wired Transport using Propagation Time Averaging*, Samsung presentation to January, 2007 IEEE 802.1 AVB TG Meeting, Monterey, CA, USA, January 24, 2007. Available at http://www.ieee802.org/1/files/public/docs2007/as-garner-prop-time-averaging-0107.pdf

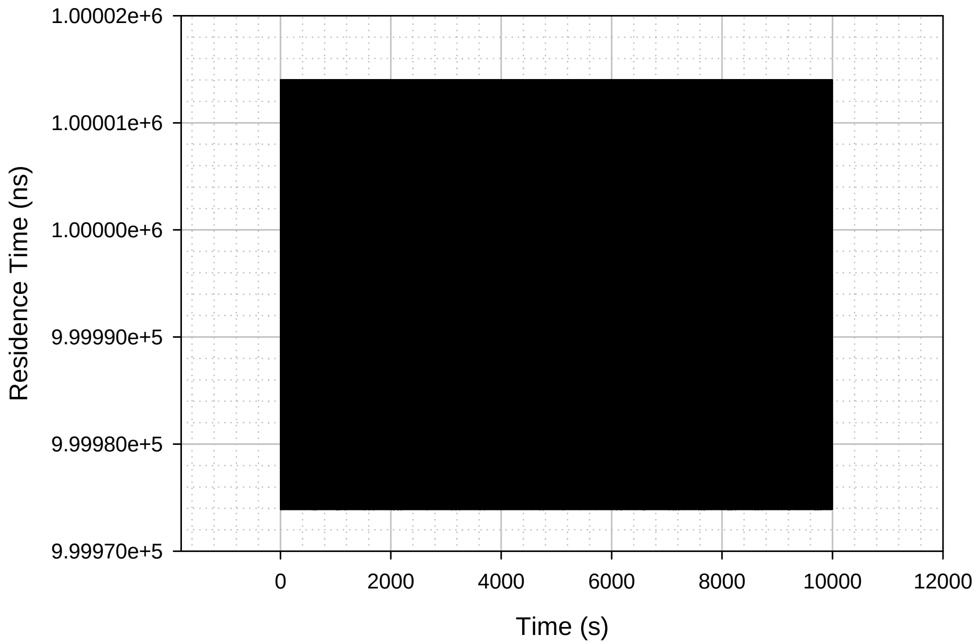
# BACKUP

# Cases 1 – 5, Node 4 Residence Time



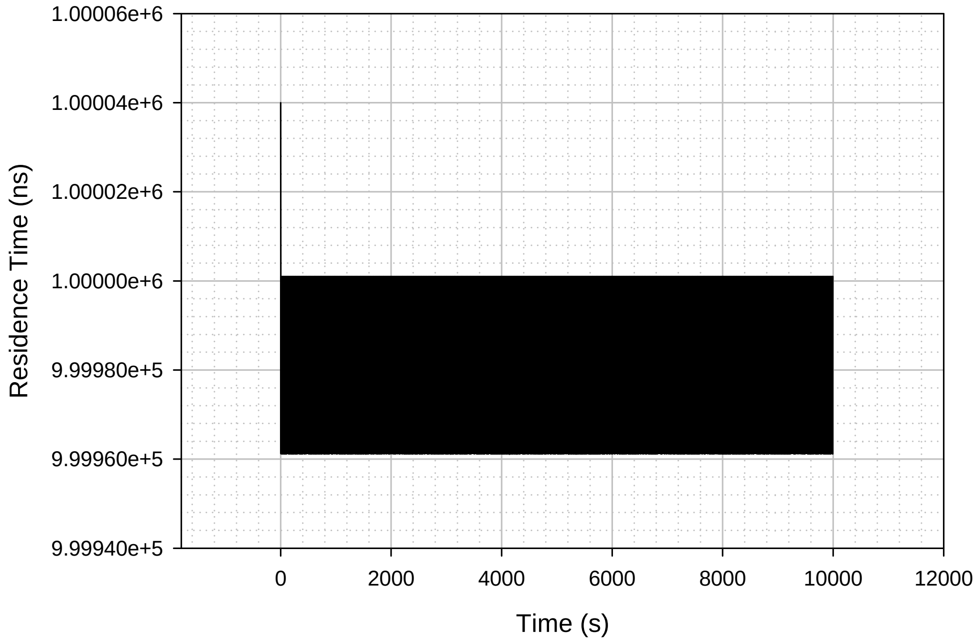
# Cases 1 – 5, Node 5 Residence Time

Node 5 residence time, Cases 1 - 5
(after initialization)
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s
Endpoint filter BW = 0.01 Hz
Local oscillator tolerance = +/- 100 ppm
(frequency offsets initialized randomly within tolerance range)
propagation time averaging: window = 1000 Pdelay measurements,
weight = 1/e after 1000 measurements

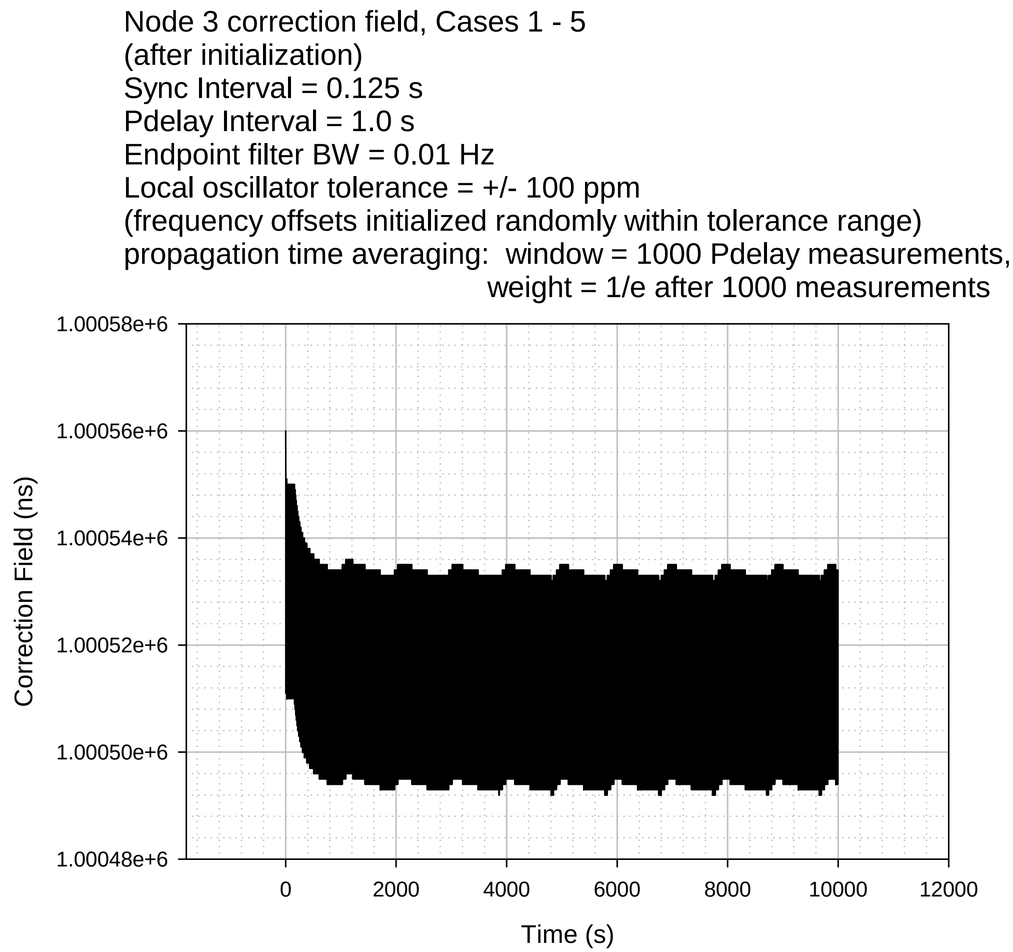


# Cases 1 – 5, Node 6 Residence Time

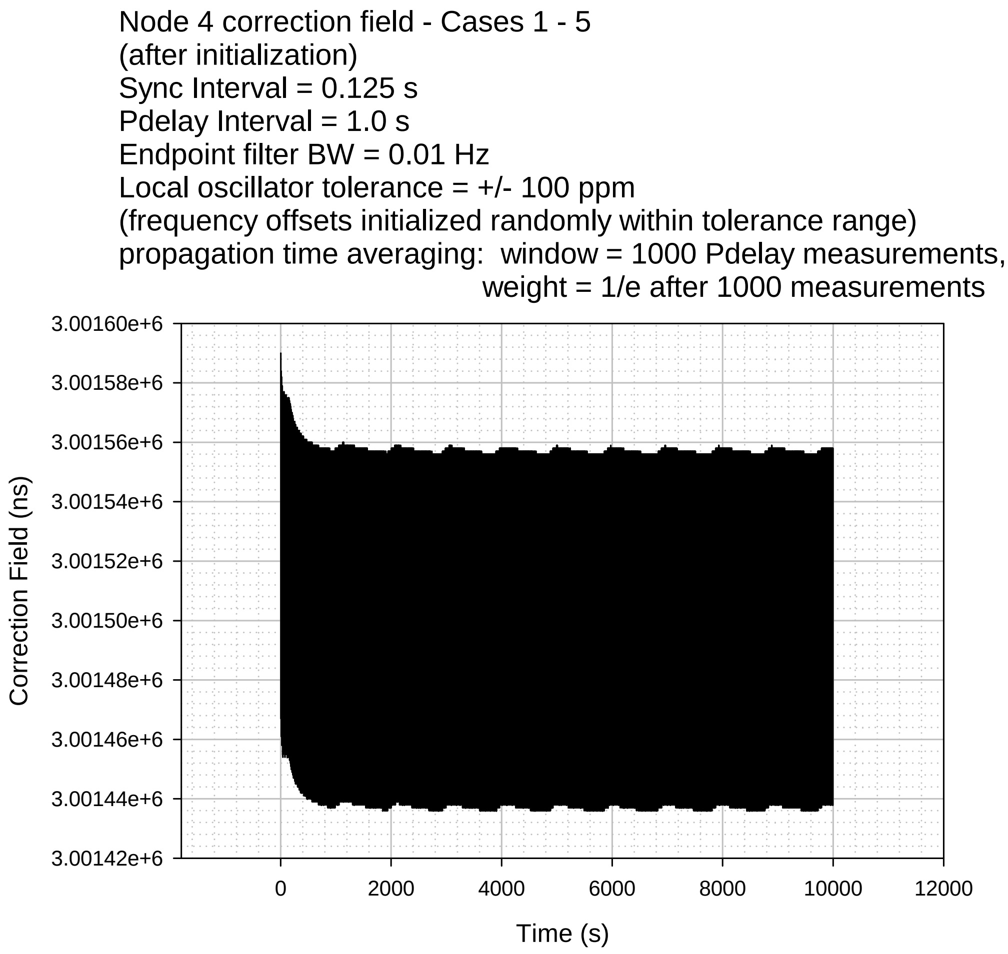
Node 6 residence time, Cases 1 - 5
(after initialization)
Sync Interval = 0.125 s
Pdelay Interval = 1.0 s
Endpoint filter BW = 0.01 Hz
Local oscillator tolerance = +/- 100 ppm
(frequency offsets initialized randomly within tolerance range)
propagation time averaging: window = 1000 Pdelay measurements,
weight = 1/e after 1000 measurements



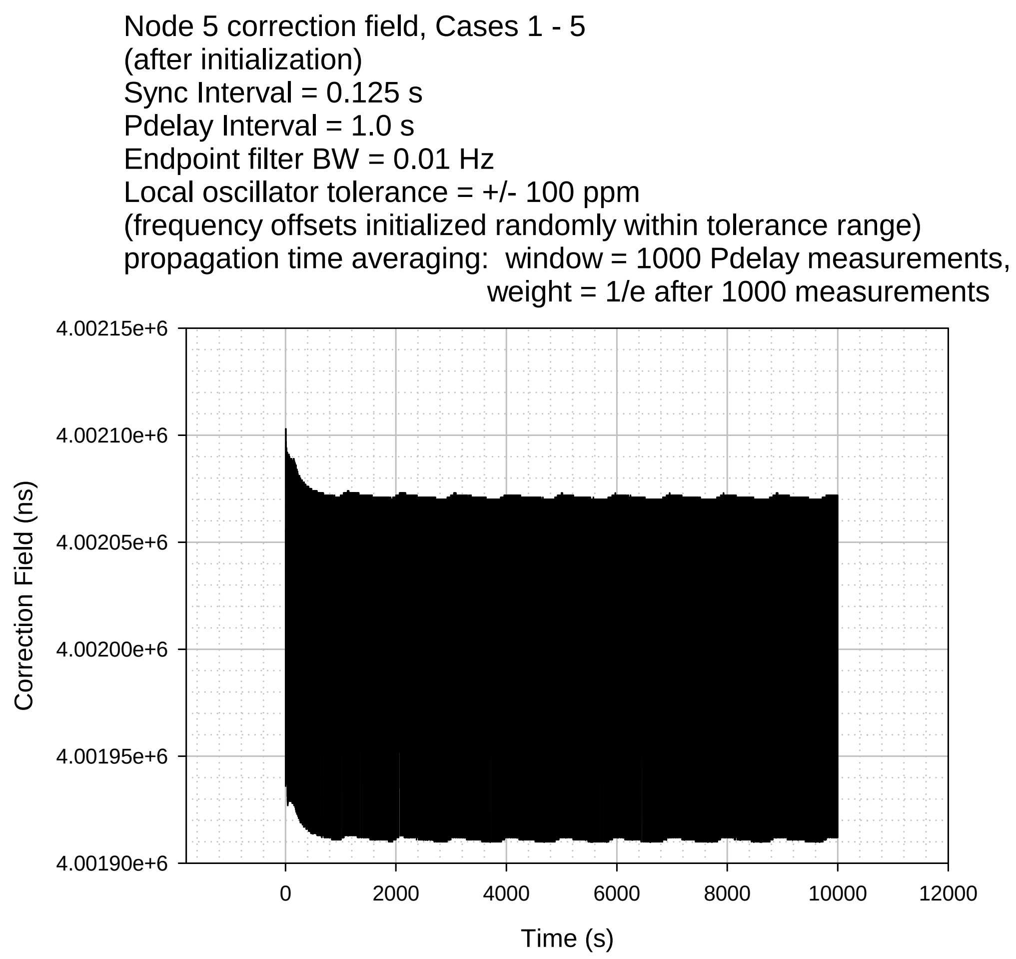
# Cases 1 – 5, Node 3 Correction Field



# Cases 1 – 5, Node 4 Correction Field



# Cases 1 – 5, Node 5 Correction Field



# Cases 1 – 5, Node 6 Correction Field