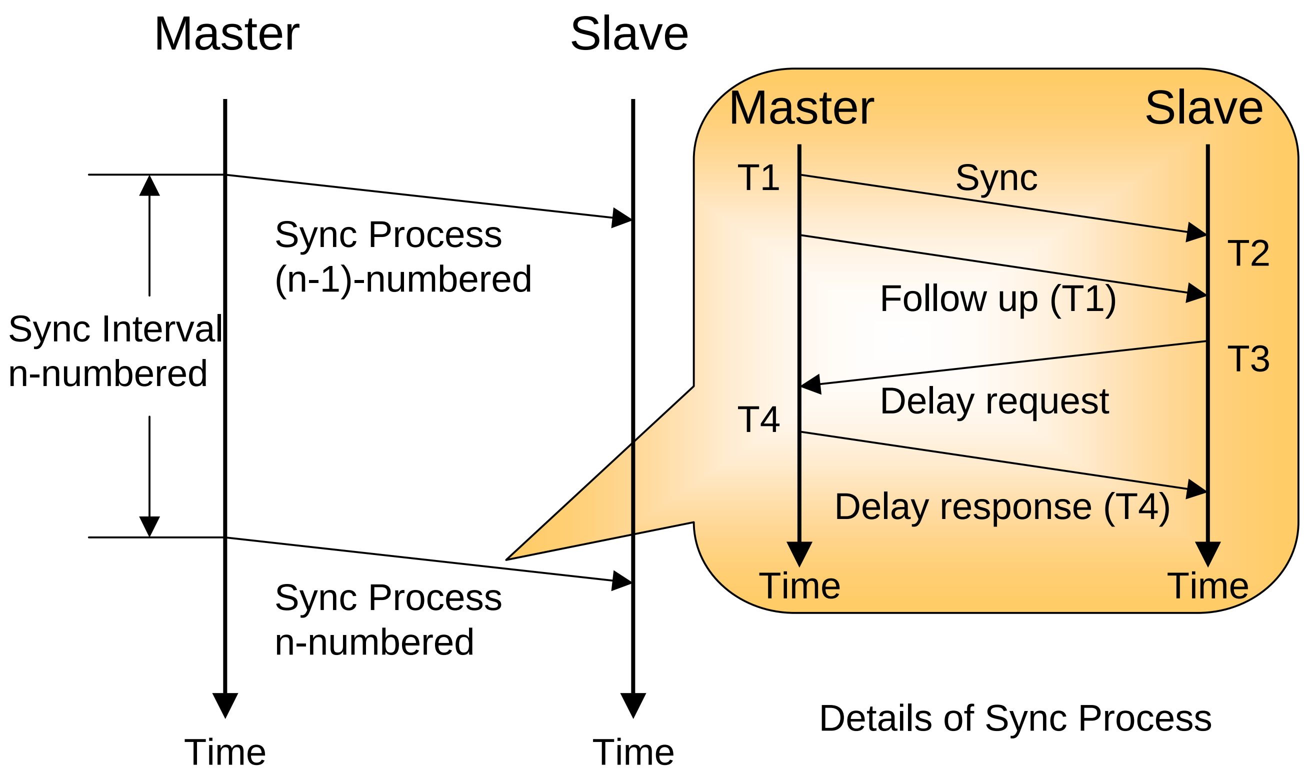
Improvements to Boundary Clock Based Time Synchronization through Cascaded Switches

Sihai Wang
Samsung Electronics
sihai.wang@samsung.com

Outline

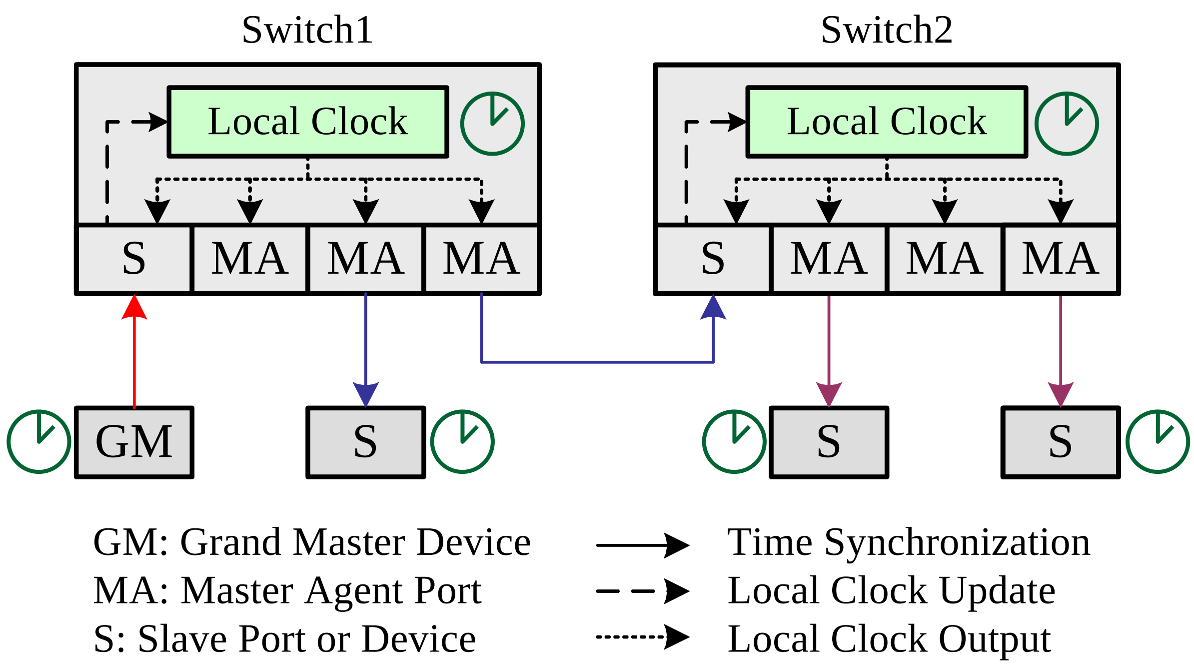
- Introduction to IEEE-1588 (PTP)
- Synchronization-Capable Clock
- Improved Schemes
- Experimental and Simulated Results
- Conclusions

Basic Procedure of PTP



- Election of grand master is not included here
- *Toffset* = [(*T2* - *T1*) - (*T4* - *T3*)] / 2
- Frequency offset can be derived from *Toffset* (Scheme dependent)

PTP through Cascaded SWs

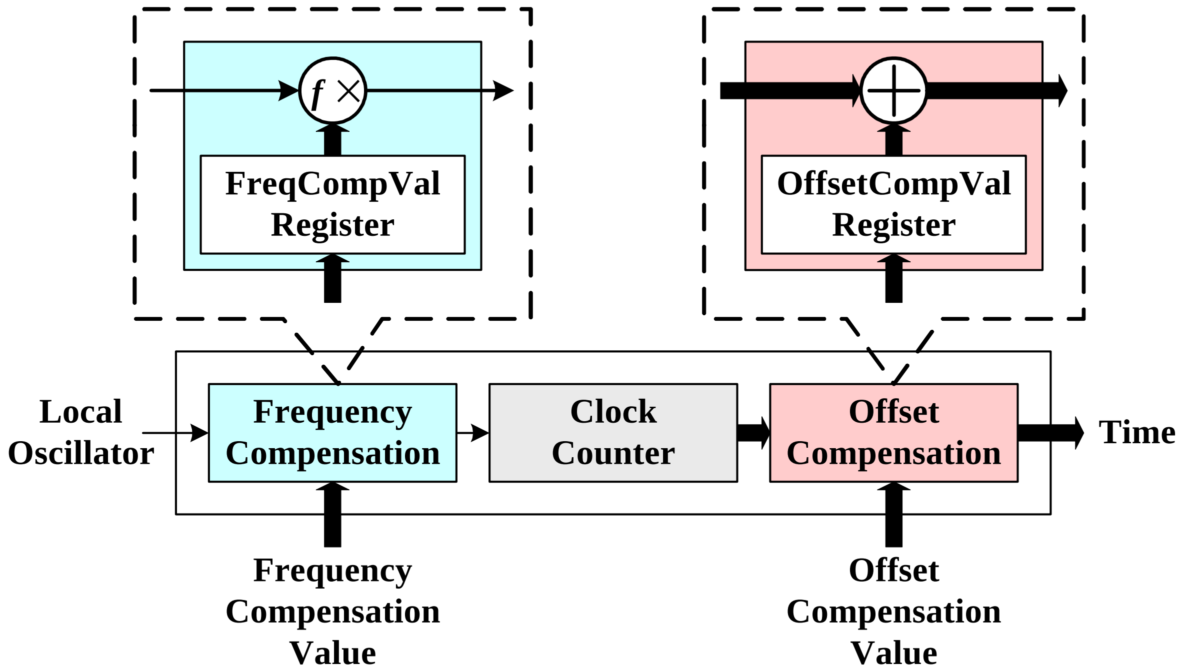


- Slave port maintains local clock (LC, should be sync-capable clock)
- The unique LC provides time info to all ports of a switch
- Sync procedures of different hops are independent
- Error accumulation can be exponential vs. hop number [1] (depending on design of PLL control loop)

Outline

- Introduction to IEEE-1588 (PTP)
- Synchronization-Capable Clock
- Improved Schemes
- Experimental and Simulated Results
- Conclusions

Sync-Capable Clock (SCC)



- Freq. and Offset are compensated by updating FreqCompVal and OffsetCompVal registers, respectively (see slides 8 – 10 for algorithm)
- Freq. compensation module is plotted equivalently here, and details are illustrated in [2]

Classification of SCC

- Offset&Freq. Compensation Clock (OFCC)
  - Both offset&freq. compensation modules
  - Both offset&freq. compensation abilities
- Freq.-only Compensation Clock (FCC)
  - Only freq. compensation module
  - Both offset&freq. compensation abilities [2]
- Offset-only Compensation Clock (OCC)
  - Only offset compensation module
  - Only offset compensation ability

OFCC Compensation

- $FreqCompVal_0 = 1$
- $FreqCompVal_n = FreqScaleFactor_n * FreqCompVal_{n-1}$
- $OffsetCompVal_0 = 0$
- $OffsetCompVal_n = OffsetCompVal_{n-1} – Toffset_n$
  - *Toffset* is given on slide 3
- $FreqScaleFactor_1 = 1$
- $FreqScaleFactor_n = T_{SyncInt,n} / (T_{SyncInt,n} + Toffset_n)$
  - $T_{SyncInt,n}$ = synch interval (slide 3)
- This algorithm differs from the algorithm used in [3] and [4] in that
  - The frequency scale factor here is calculated using the corrected (compensated) phase
  - The frequency scale factor in [3] and [4] is calculated using the uncorrected (uncompensated) phase obtained from the free-running oscillator

FCC Compensation -- 1

- $FreqCompVal_0 = 1$
- $FreqCompVal_n = FreqScaleFactor_n * FreqCompVal_{n-1}$

- $FreqScaleFactor_n$ is obtained using algorithm of [2] (see section 3.0 of [2])

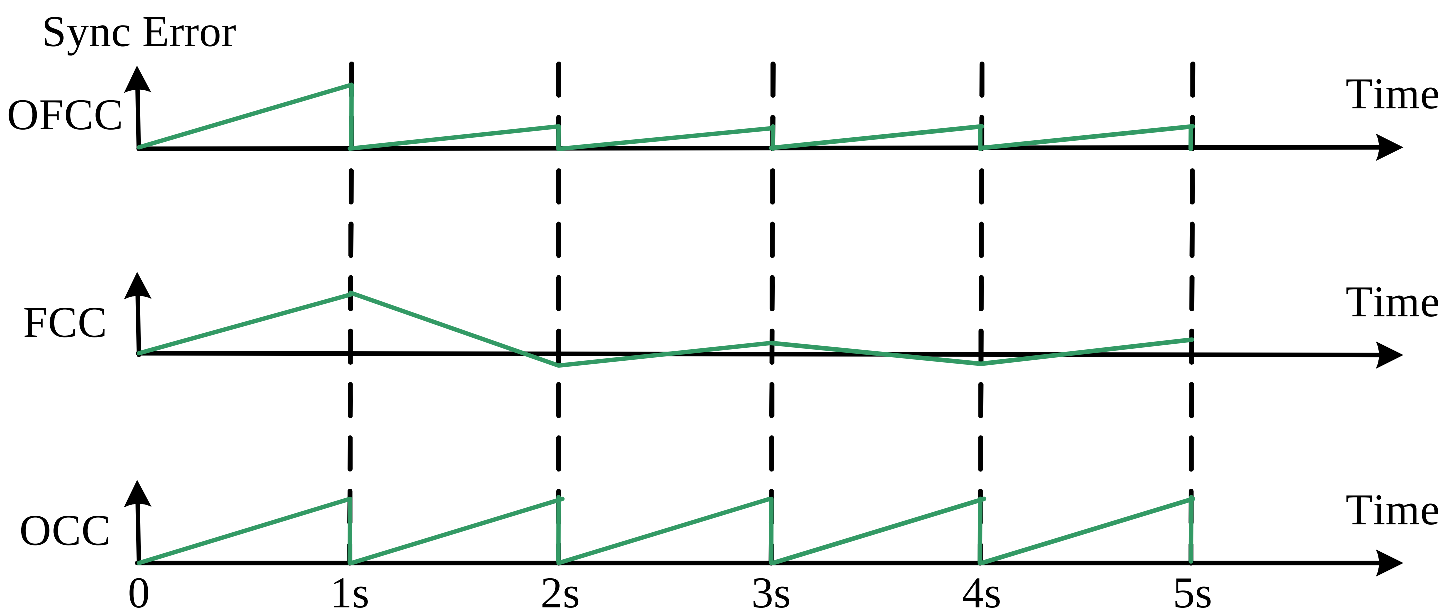
- $MasterClockCount_n = MasterClockTime_n – MasterClockTime_{n-1}$
- $SlaveClockCount_n = SlaveClockTime_n – SlaveClockTime_{n-1}$
- $ClockDiffCount_n = MasterClockTime_n – SlaveClockTime_n$
- $FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n) / SlaveClockCount_n$

- The frequency scale factor has 2 terms, which attempt to correct for 2 effects
  - $ClockDiffCount_n / SlaveClockCount_n$ corrects for the rate difference between master and slave
  - $MasterClockCount_n / SlaveClockCount_n$ tries to change the frequency to drive the phase error to zero over the next synch interval

FCC Compensation -- 2

- In FCC compensation, offset is not directly compensated as in OFCC
- Rather, phase is obtained by integrating the compensated frequency
  - The *Toffset* values are used to obtain the compensated frequency, as on the previous slide
  - This is why the MasterClockCount term in the expression for FreqScaleFactor is necessary

Sync Error Evolvement of the SCCs

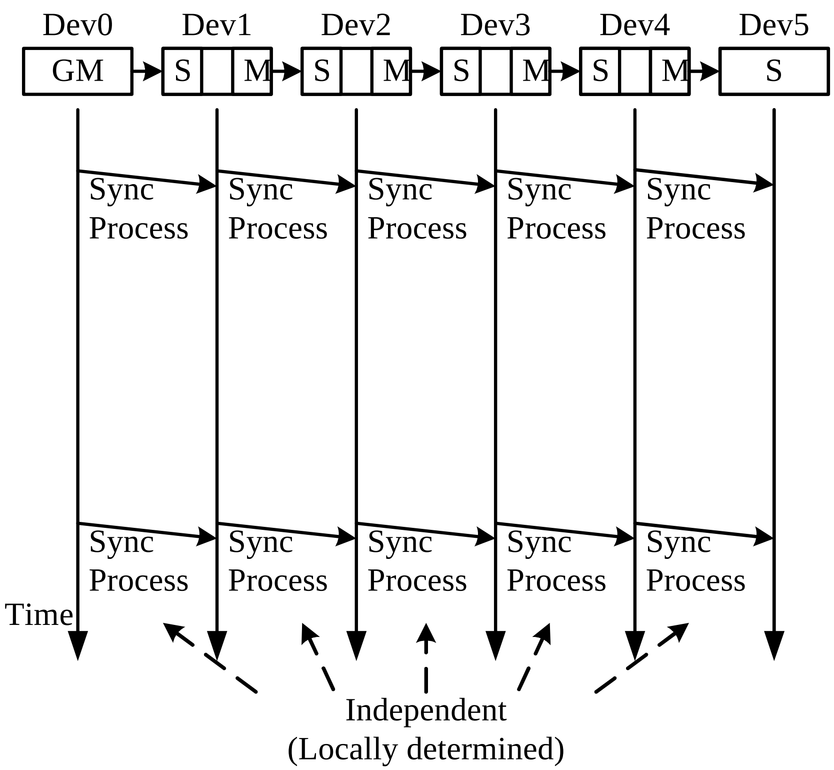


- OCC has the highest error, and will not be discussed
- OFCC and FCC can achieve equivalent precision

Outline

- Introduction to IEEE-1588 (PTP)
- Synchronization-Capable Clock
- Improved Schemes
- Experimental and Simulated Results
- Conclusions

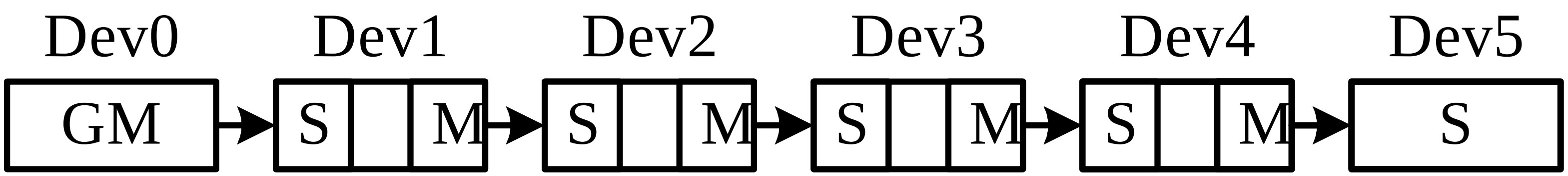
Conventional Cascaded Sync Scheme



- Start time of each sync process is determined locally (independent)
- No info exchange between different hops

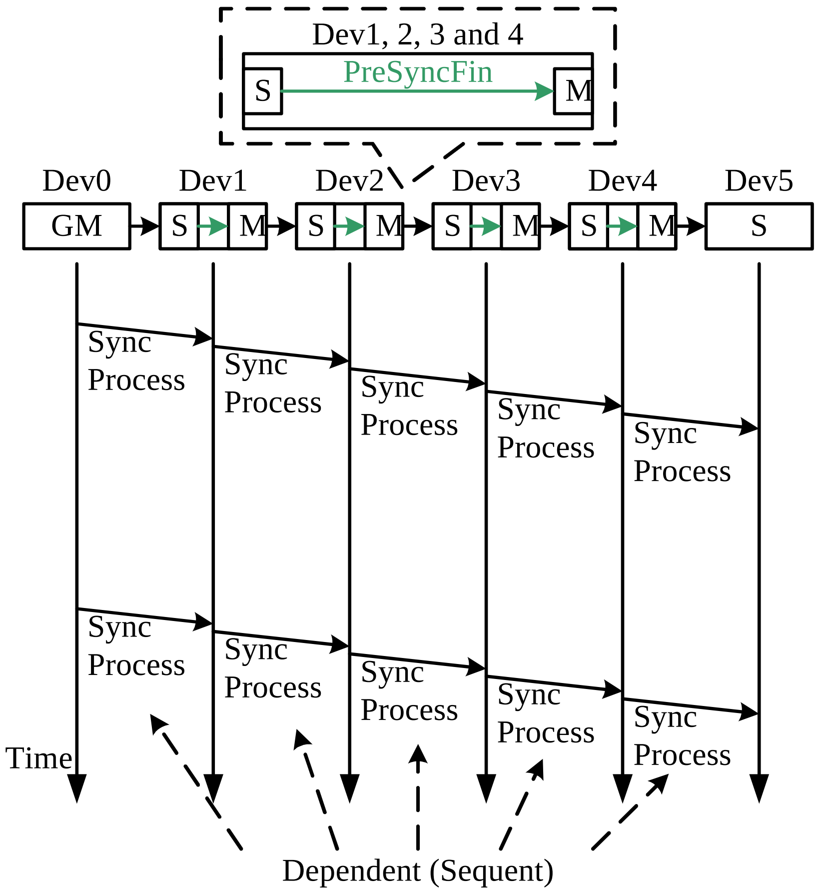
Hereafter, either OFCC or FCC can be used and both of them will be investigated

Improvement – Common Description



- As an example, if Dev1 knows its time synchronization status, i.e. what time it has the possible minimum error and/or its time error at a certain time, it can either
  - start synchronization process to Dev2 while its time has the minimum error or
  - send its synchronization error information to Dev2 to be compensated

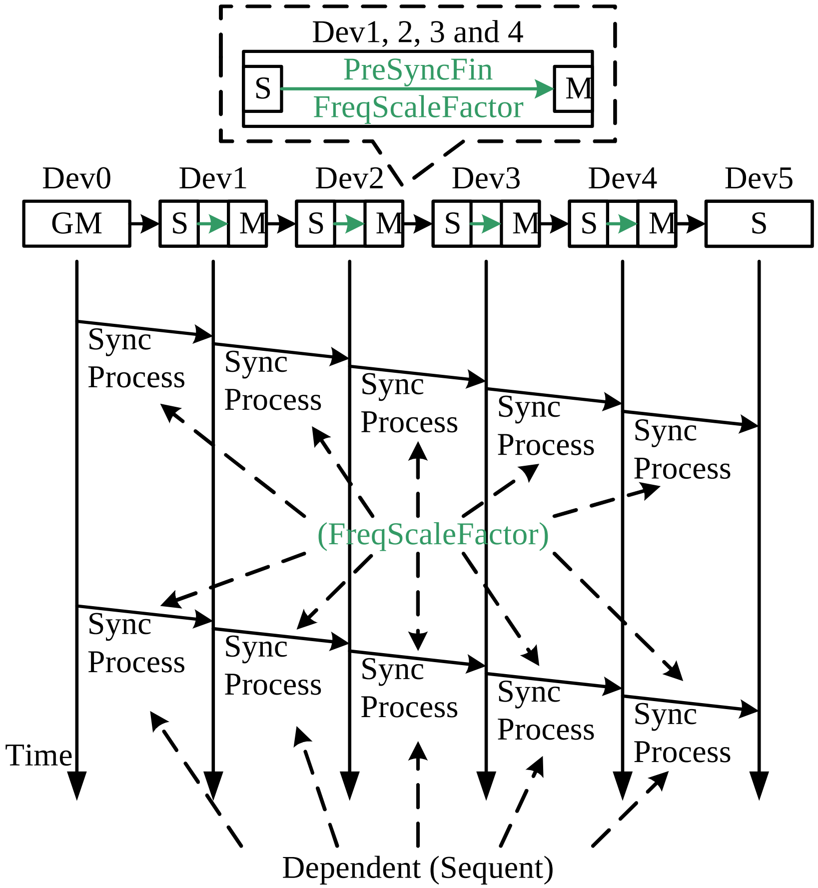
Improvement (Using OFCC)



- Only and just after previous hop sync finished, next hop starts sync proc.
- So start time of sync process become sequent (dependent) by adding PreSyncFin signal

PreSyncFin: Previous Synchronization Finished

Improvement (Using FCC)



- Only and just after previous hop sync finished, next hop starts sync proc.
- Besides PreSyncFin, FreqScaleFactor is transferred to next hop device where it will be compensated

FreqScaleFactor: Frequency Scaling Factor

Outline

- Introduction to IEEE-1588 (PTP)
- Synchronization-Capable Clock
- Improved Schemes
- Experimental and Simulated Results
  - Scheme using OFCC is verified by experiments
  - Scheme using FCC is verified by simulations
- Conclusions

Simulation Setup (FCC)

- 8 chained devices (Dev0~7), Dev0 is GM
- Link speed: 100MHz
- Crystal Frequency: 50MHz
- Sync Interval: 2^20ns (~1.049ms)
  - In order to save simulation computing time
- Cycle Indicator (CI): 2^17ns (131.072µs)
- Relative to GM, Dev1~7 freq. deviations are: +50, +100, … and +350ppm, respectively
- Simulated time: >250ms
- The same analysis method with that using OFCC
- Simulated time errors of CI are plotted in Appendix

Simulated Results (FCC) -- 1

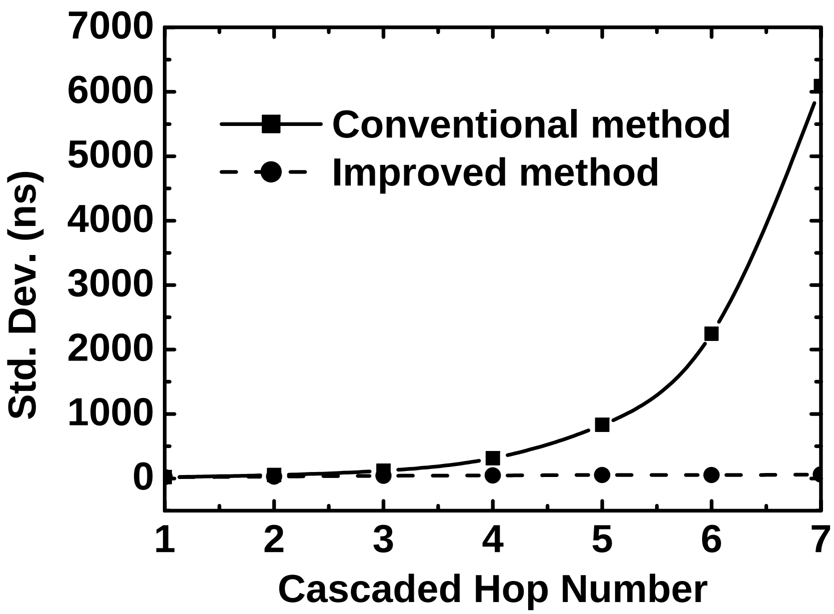
- Conventional Scheme

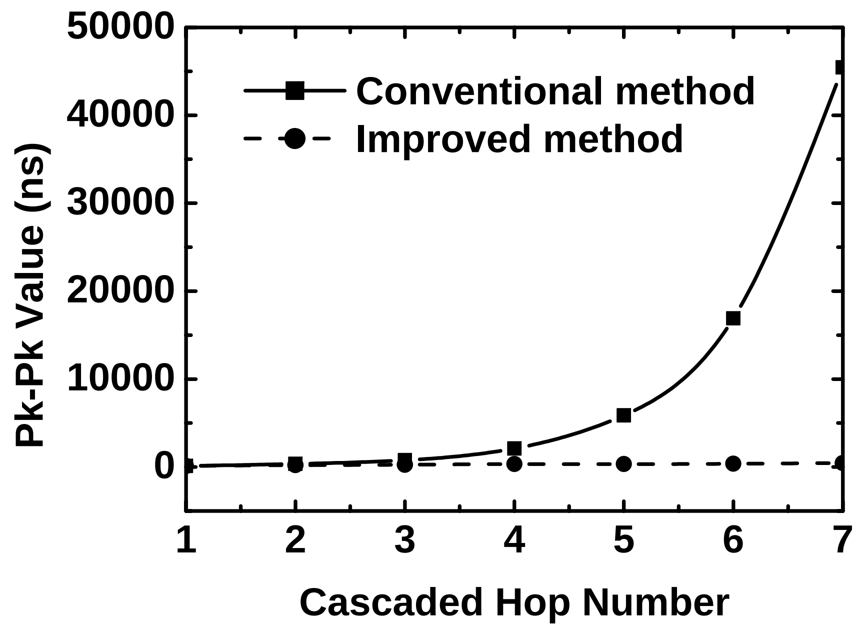
	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
Std. Dev. (ns)	20.64	52.54	120.8	313.5	831.1	2244	6089
Pk-Pk (ns)	123	357	781	2118	5882	16909	45453

- Improved Scheme

	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
Std. Dev. (ns)	20.33	28.87	40.81	47.17	52.21	53.6	60.81
Pk-Pk (ns)	123	208	266	336	341	372	458

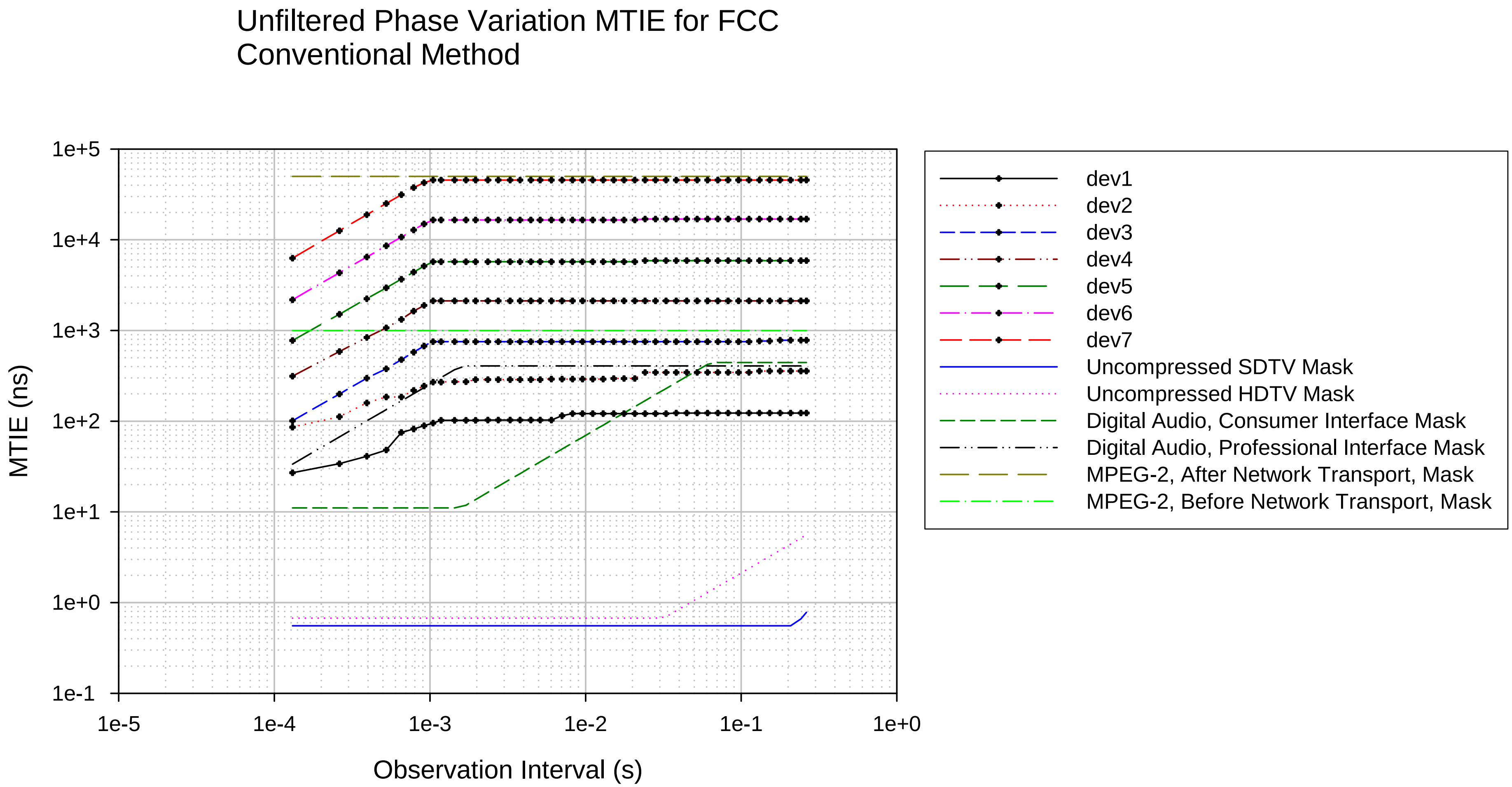
Simulated Results (FCC) -- 2



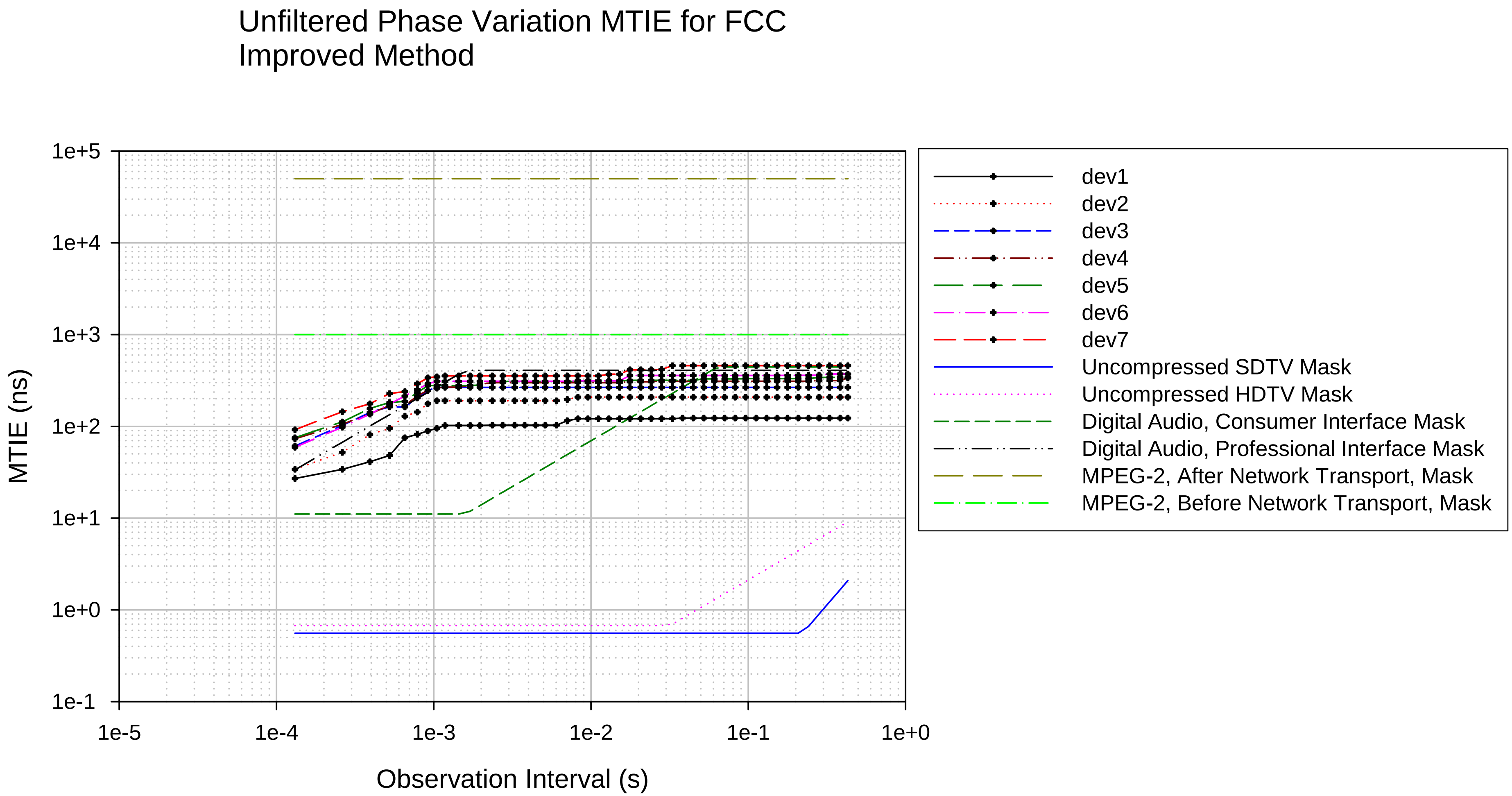


- Std. Dev. (Pk-Pk) vs. Hop Number
  - Exponential → Linearly

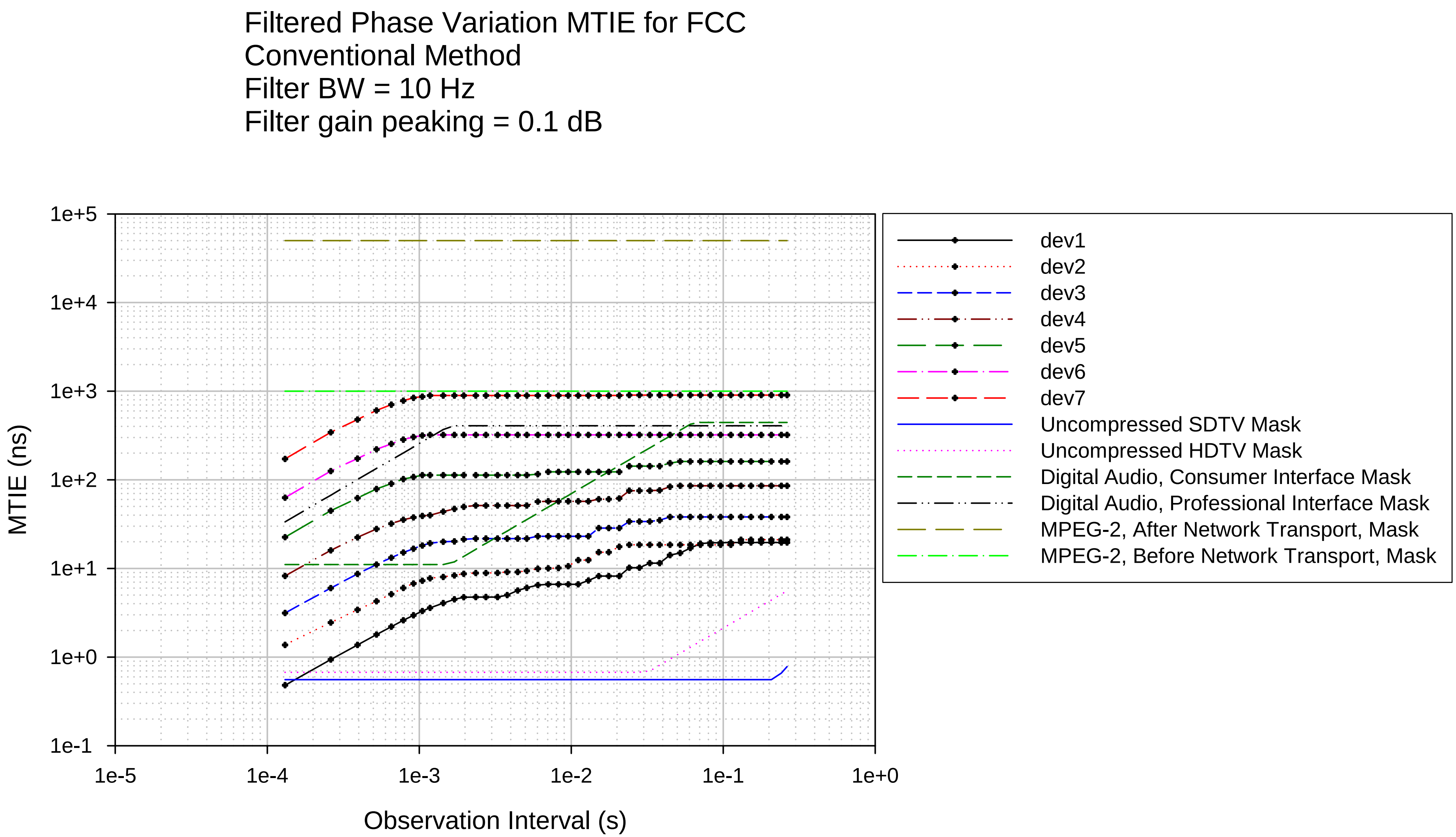
Simulated Results (FCC) -- 3



Simulated Results (FCC) -- 4

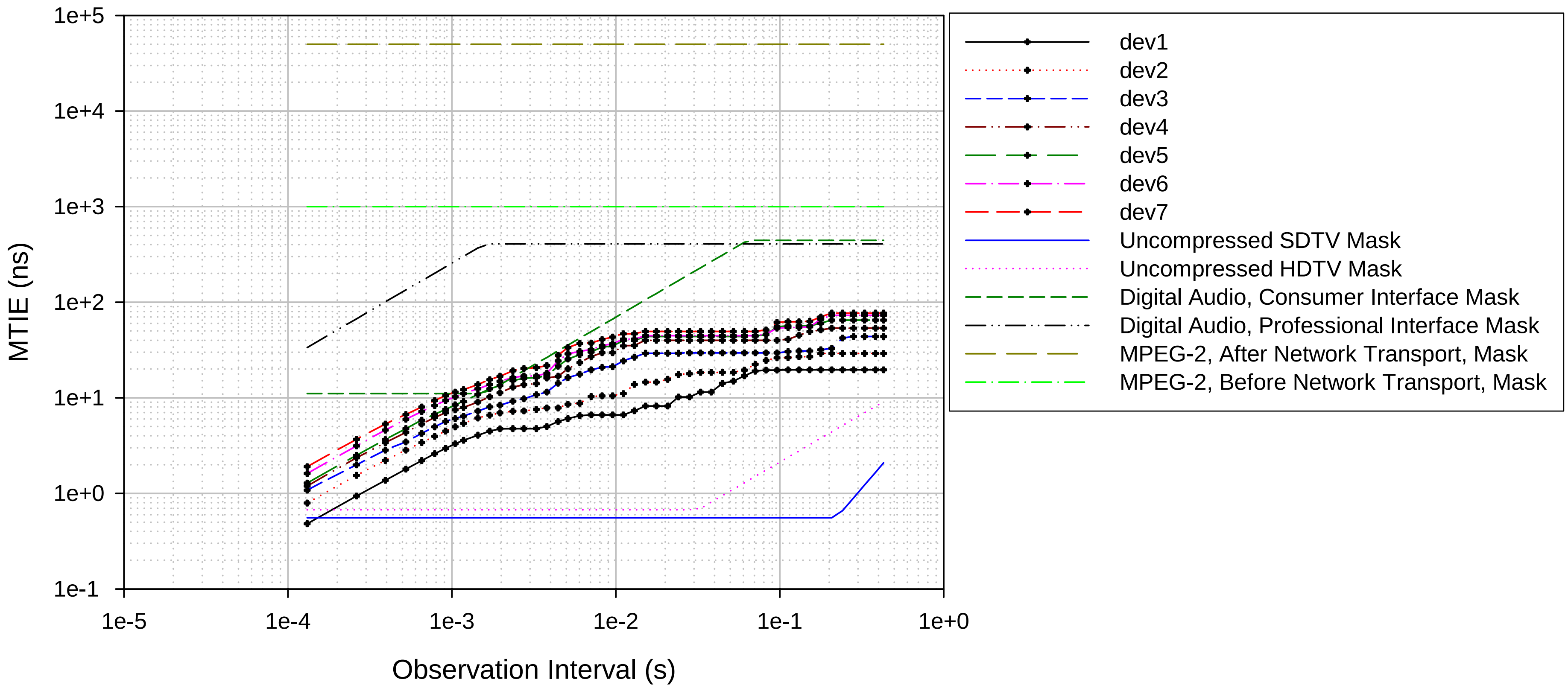


Simulated Results (FCC) -- 5



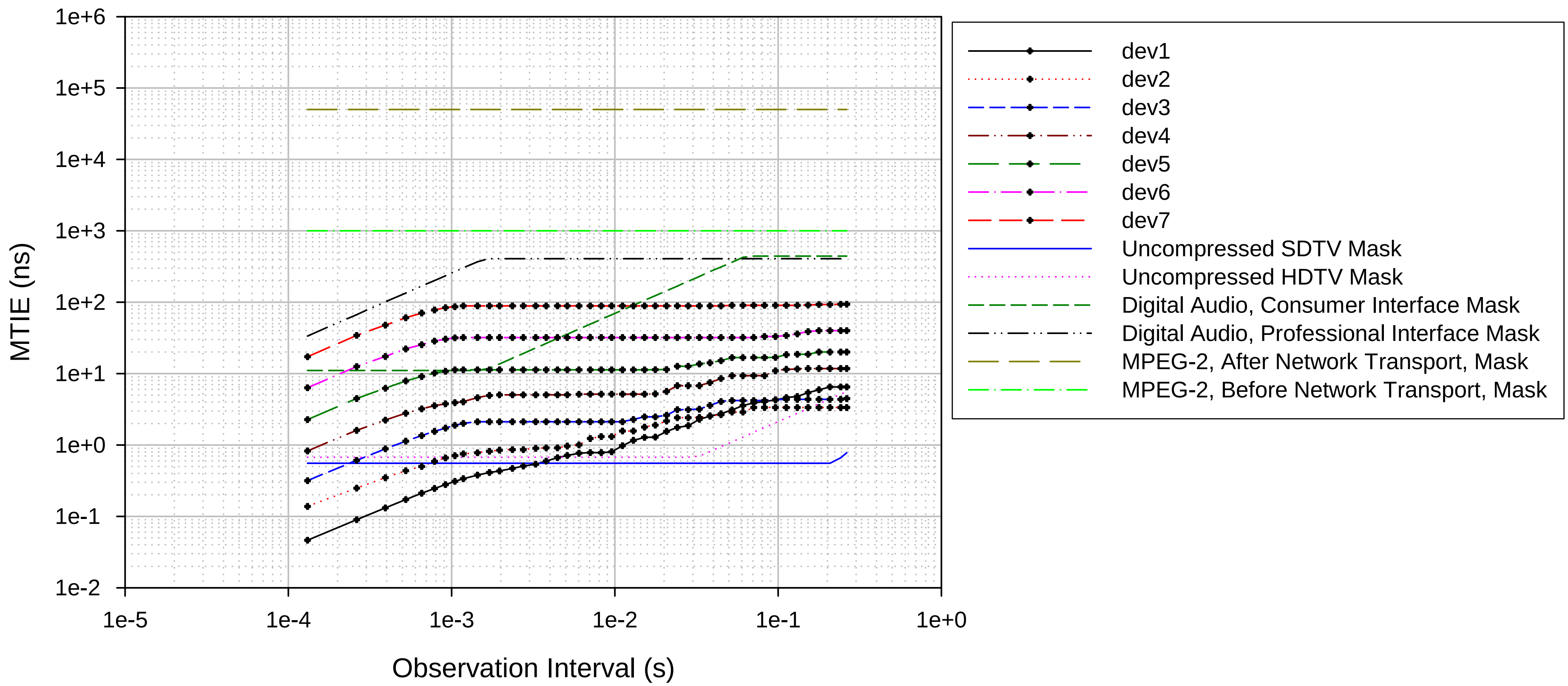
Simulated Results (FCC) -- 6

Filtered Phase Variation MTIE for FCC
Improved Method
Filter BW = 10 Hz
Filter gain peaking = 0.1 dB



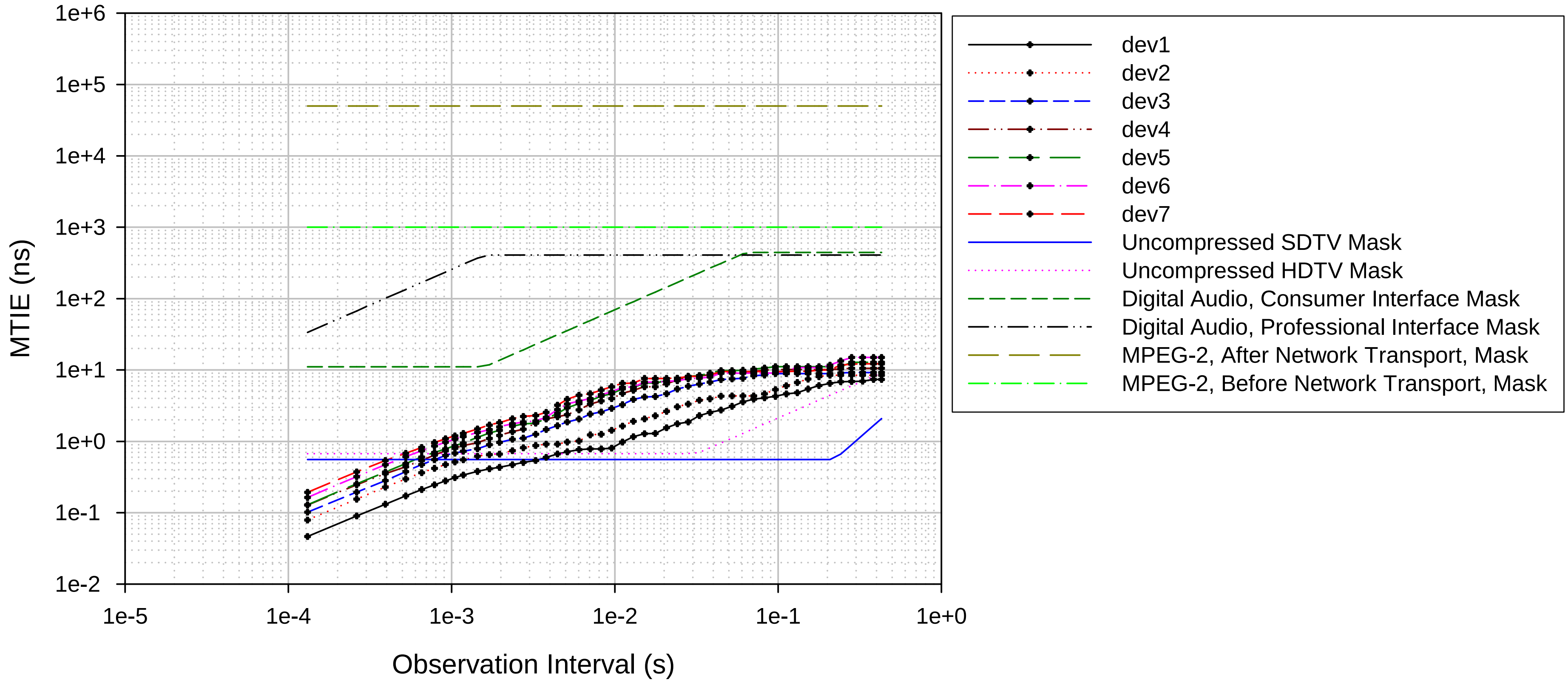
Simulated Results (FCC) -- 7

Filtered Phase Variation MTIE for FCC
Conventional Method
Filter BW = 1 Hz
Filter gain peaking = 0.1 dB



Simulated Results (FCC) -- 8

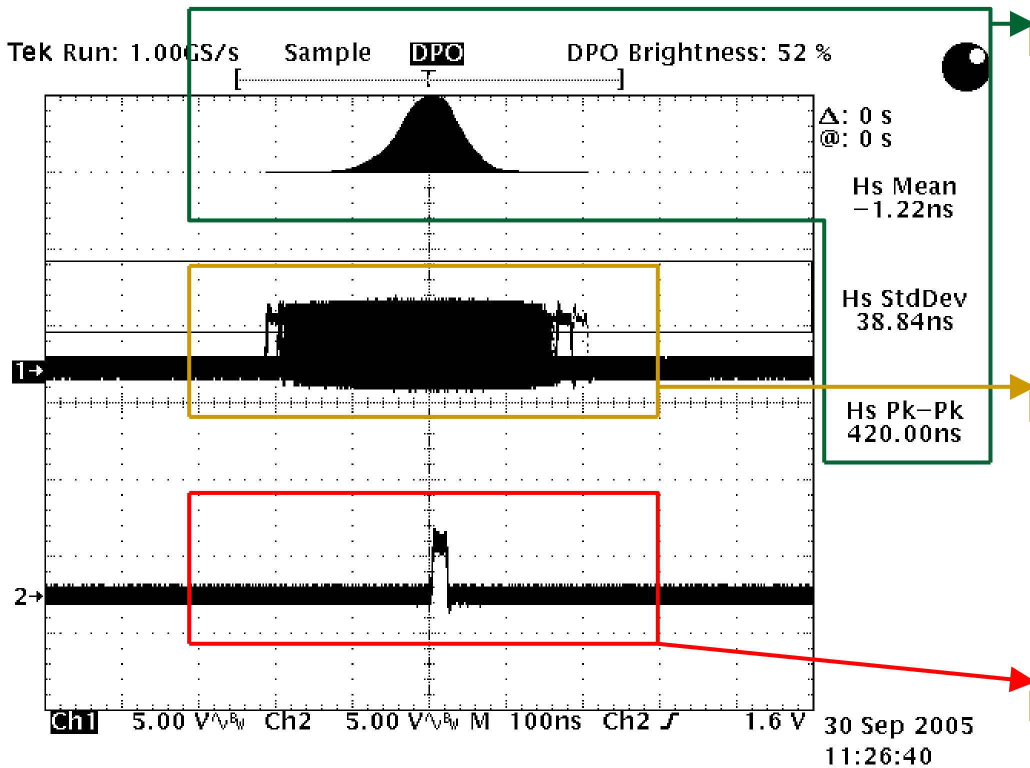
Filtered Phase Variation MTIE for FCC
Improved Method
Filter BW = 1 Hz
Filter gain peaking = 0.1 dB



Experimental Setup (OFCC)

- 6 chained devices (Dev0~5), Dev0 is GM
- Link speed: 100MHz
- Crystal Frequency: 50MHz
- Sync Interval: 2^30ns (~1.074s)
- Cycle Indicator (CI): 2^17ns (131.072µs)
- Test Time: >1hour
- CIs of GM and individual slave are monitored and recorded for sync precision examination

Experimental Result Check Method



- Statistical results of slave CI
  - Std. Dev. and Pk-Pk values are recorded
- Slave CI
  - Dev1~5, respectively
- GM CI

1. Both conventional and improved schemes are performed
2. All slave CIs relative with GM CI are analyzed

Experimental Results (OFCC) -- 1

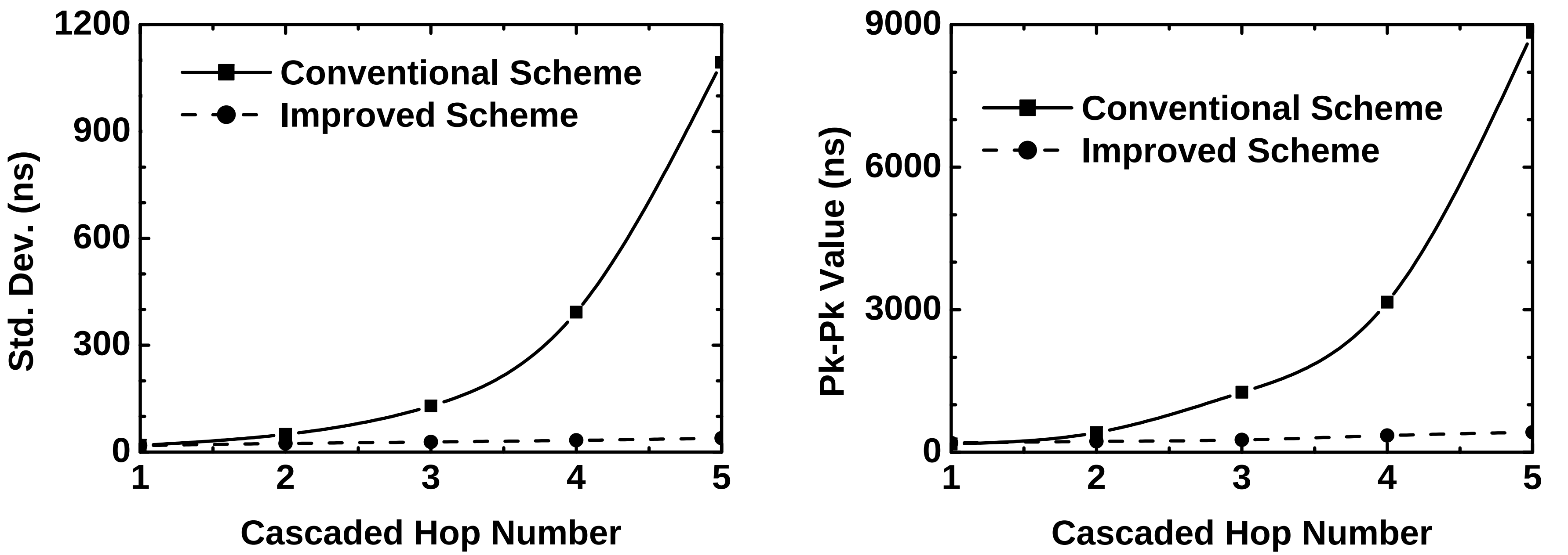
- Conventional Scheme

	Dev1	Dev2	Dev3	Dev4	Dev5
Std. Dev. (ns)	19.12	49.99	129.4	392.8	1094
Pk-Pk (ns)	177	416	1264	3160	8840

- Improved Scheme

	Dev1	Dev2	Dev3	Dev4	Dev5
Std. Dev. (ns)	18.27	24.3	28.41	33.01	38.84
Pk-Pk (ns)	198	230	260	352	420

Experimental Results (OFCC) -- 2



- Std. Dev. (Pk-Pk) vs. Hop Number
  - Exponential → Linearly

Conclusions

- Proposed Improvements
  - Sequent and hop by hop time synchronization order from grand master to slaves
  - Transferring necessary sync parameter to following hop device to be compensated
- Results
  - Sync error through cascaded switches increases linearly, instead of exponentially, with cascaded hop number increasing
  - Sync error after 5 hops has ~500ns pk-pk value (may <1µs after 7 hops) with 1s sync interval

References -- 1

1. J. Jasperneite, K. Shehab, and K. Weber, "Enhancements to the Time Synchronization Standard IEEE-1588 for a System of Cascaded Bridges," in 5th IEEE International Workshop on Factory Communication Systems (WFCS'2004), pp. 239-244

2. S. Balasubramanian, K.R. Harris, and A. Moldovansky, "A frequency compensated clock for precision synchronization using IEEE 1588 protocol and its application to Ethernet," Workshop on IEEE 1588, 2003

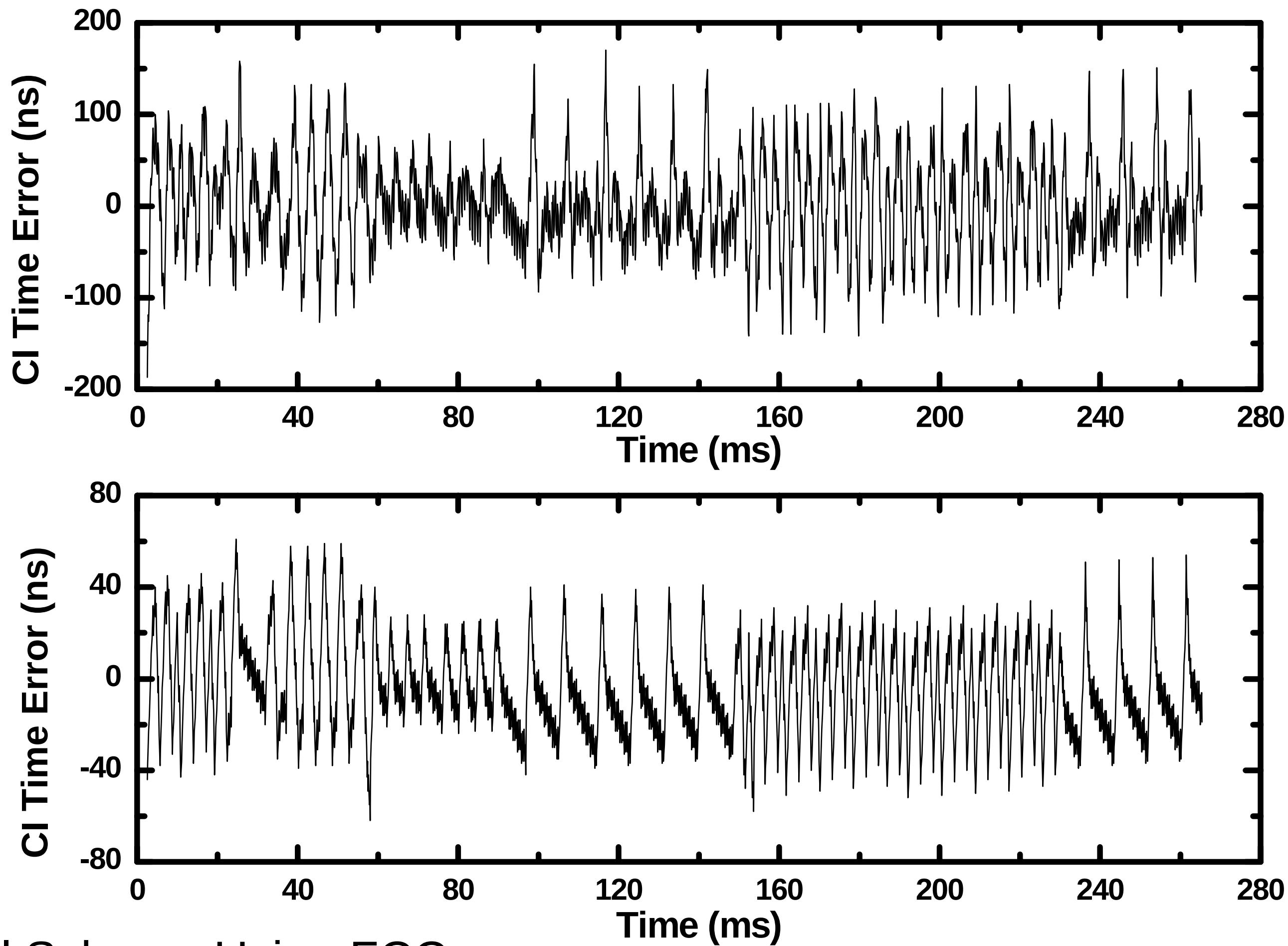
References -- 2

3. "Residential Ethernet (RE) (a working paper)," Draft 0.136, maintained by David V. James and based on work by him and other contributors, August 10, 2005. Available via http://www.ieee802.org/3/re_study/public/index.html

4. Geoffrey M. Garner and Kees den Hollander, "Analysis of Clock Synchronization Approaches for Residential Ethernet," Samsung presentation at September, 2005 Joint IEEE 802.1/802.3 ResE SG meeting, San Jose, CA, September 29, 2005. Available via http://www.ieee802.org/3/re_study/public/index.html.

Appendix – Simulated CI Time Error

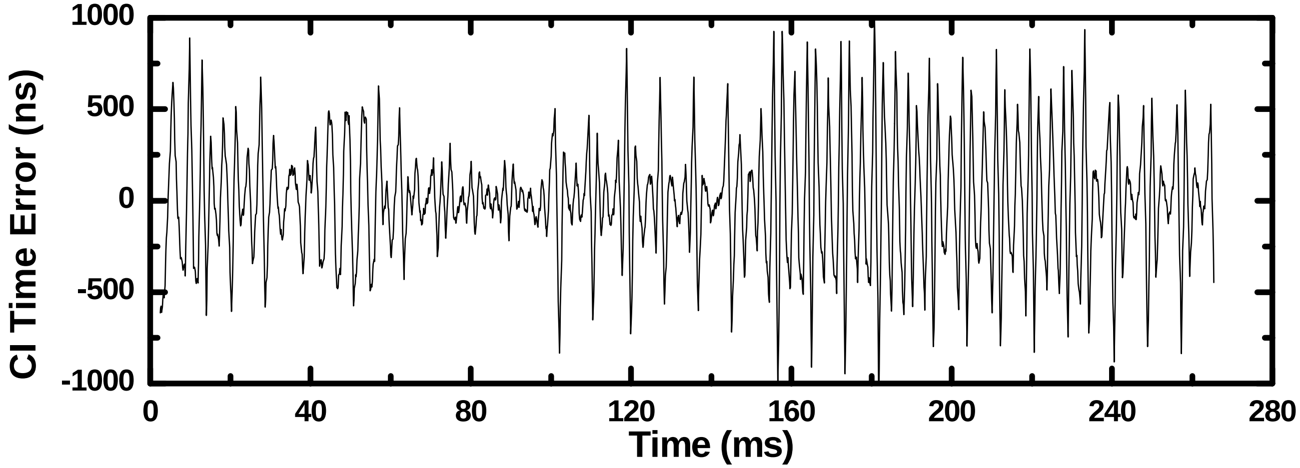
- Dev2
- Dev1



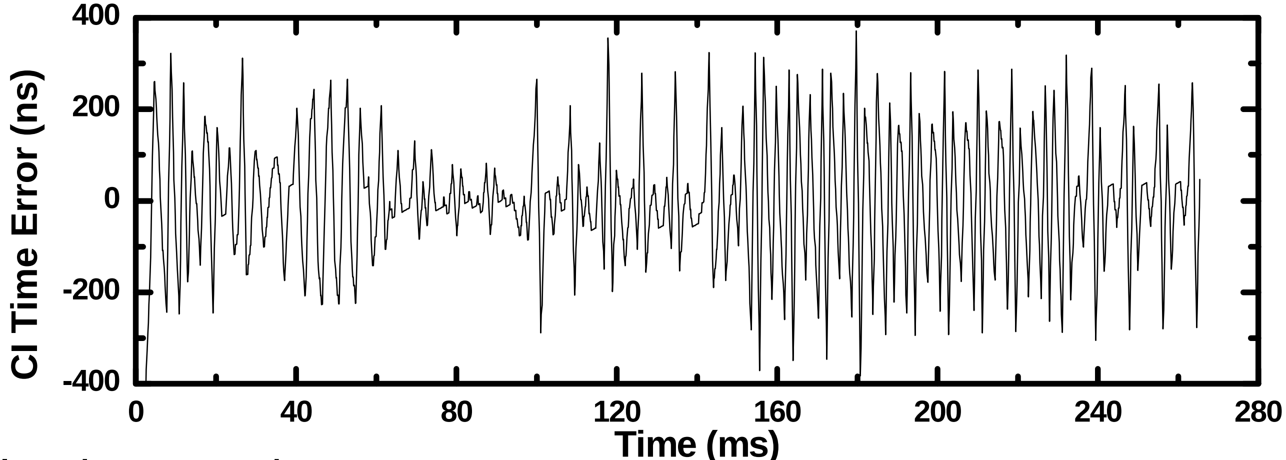
* Conventional Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev4



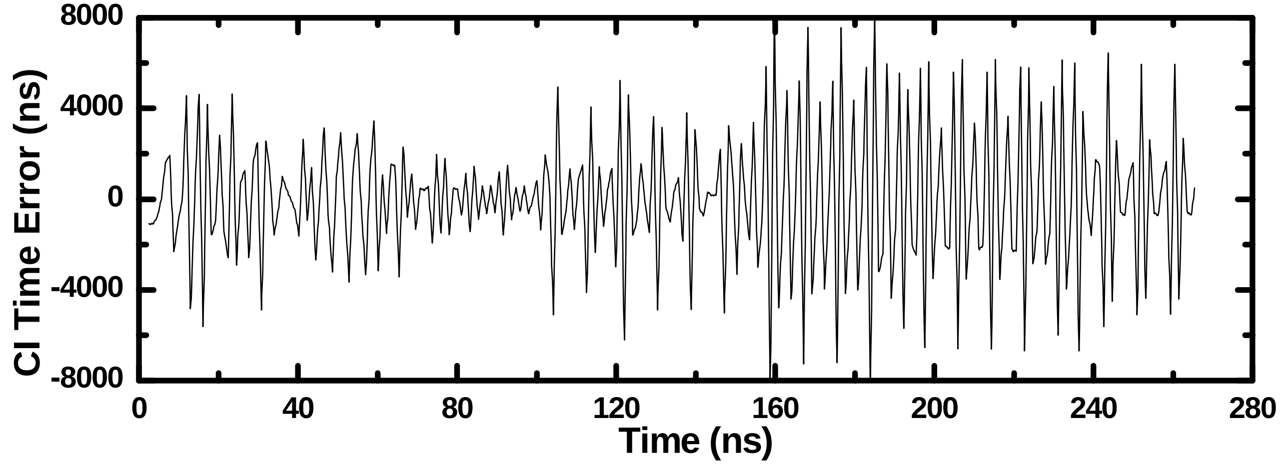
- Dev3



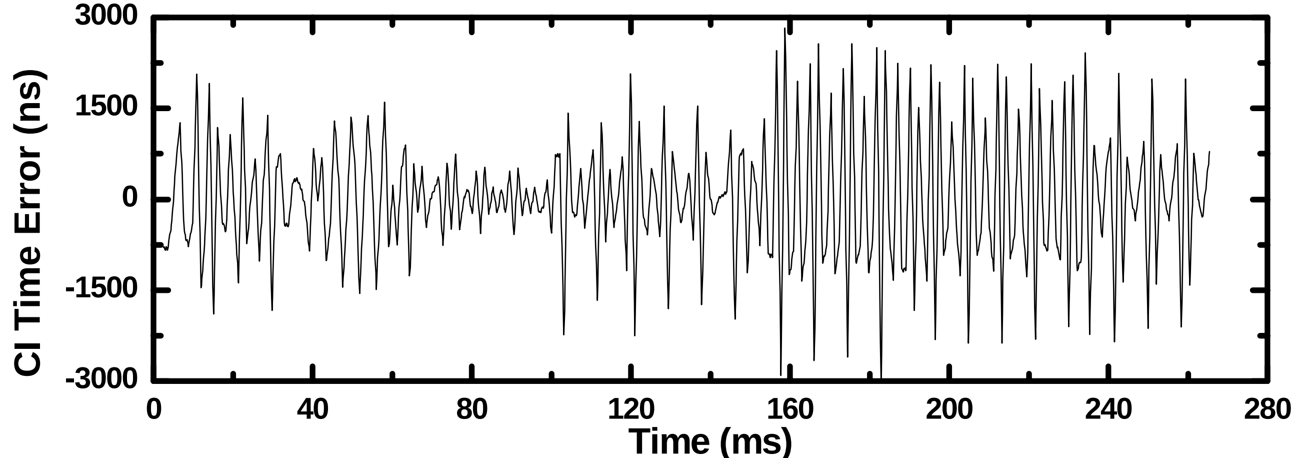
* Conventional Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev6



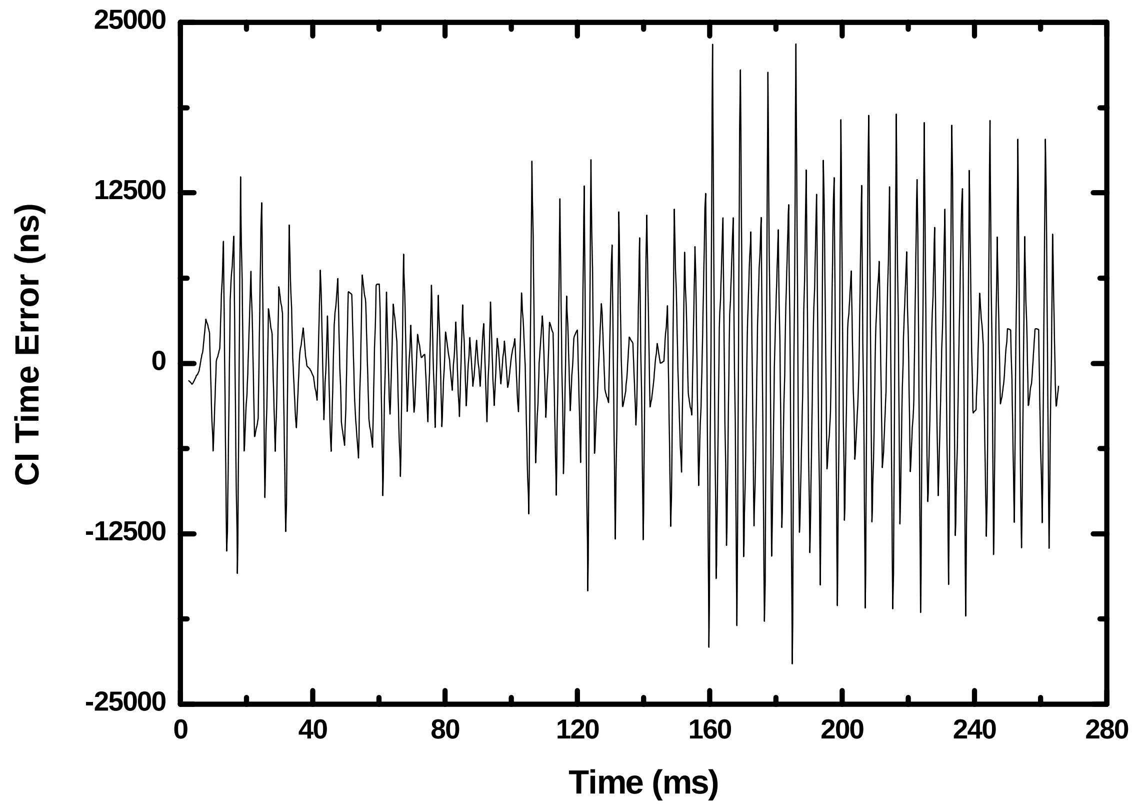
- Dev5



* Conventional Scheme Using FCC

Appendix – Simulated CI Time Error

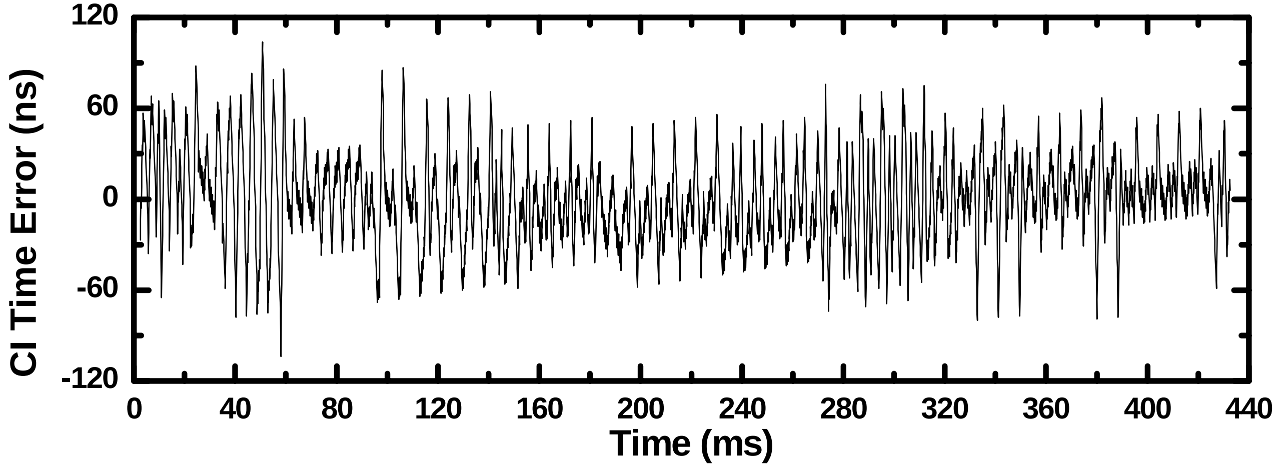
- Dev7



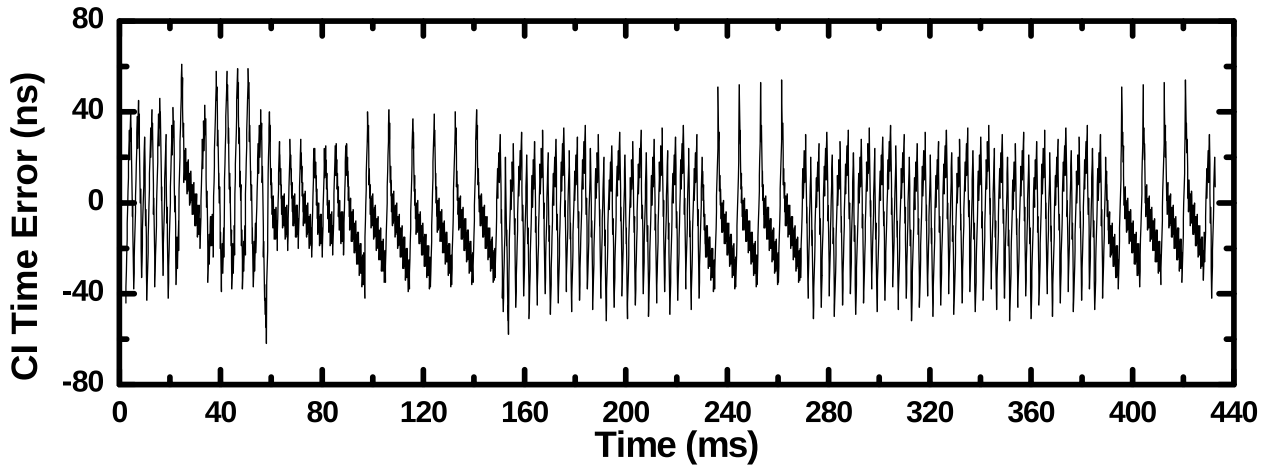
* Conventional Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev2



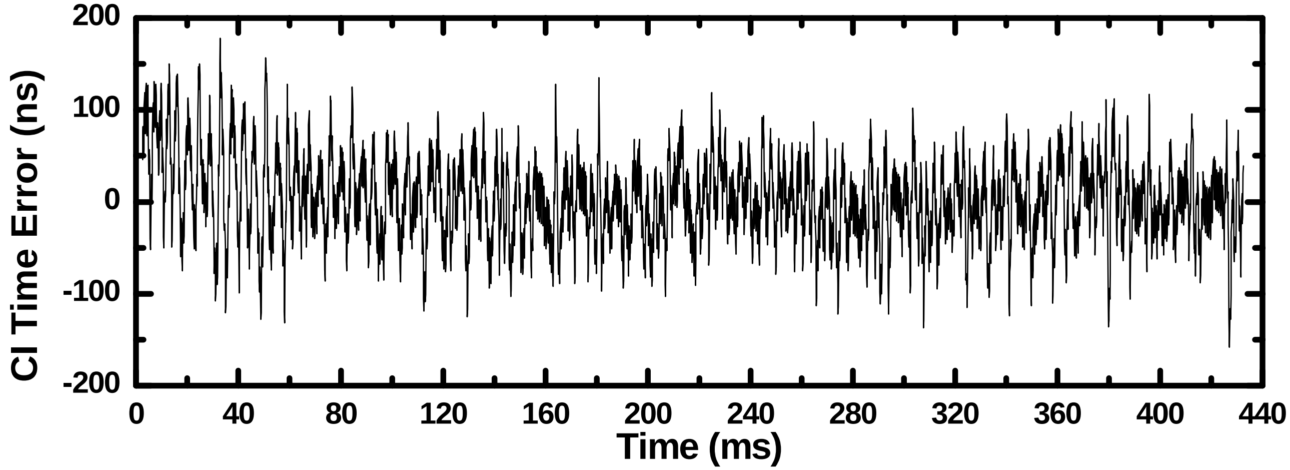
- Dev1



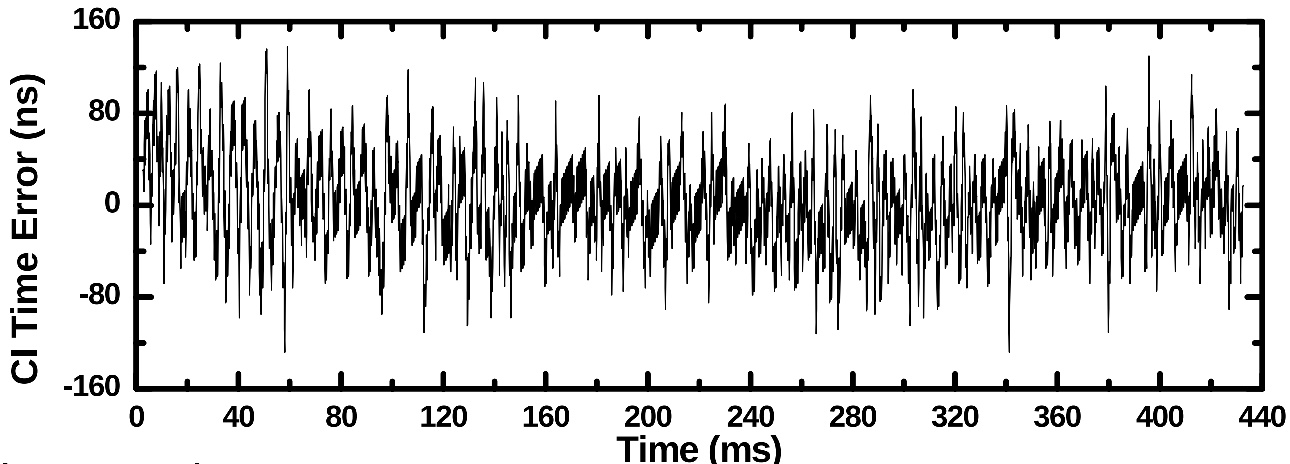
* Improved Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev4



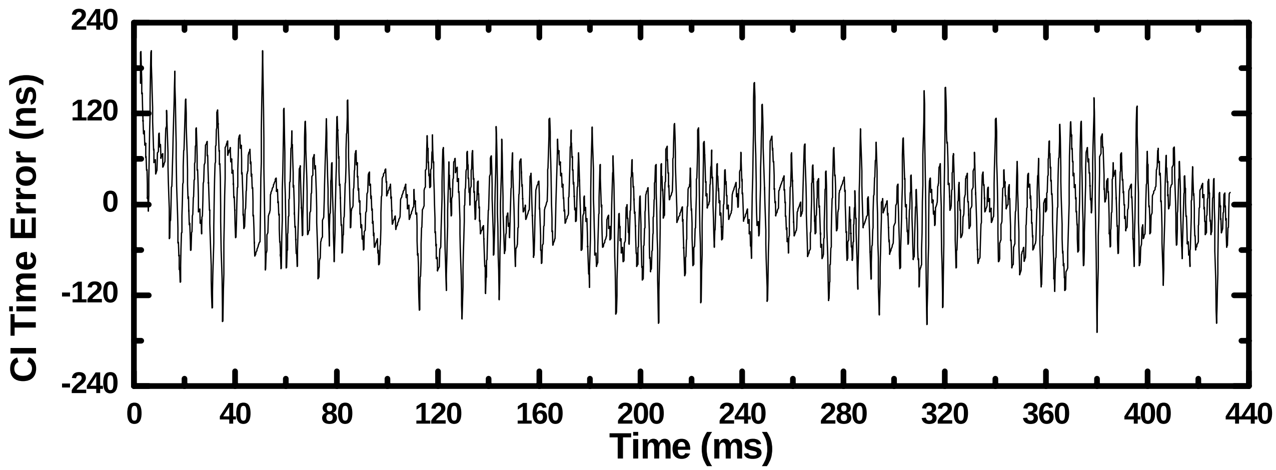
- Dev3



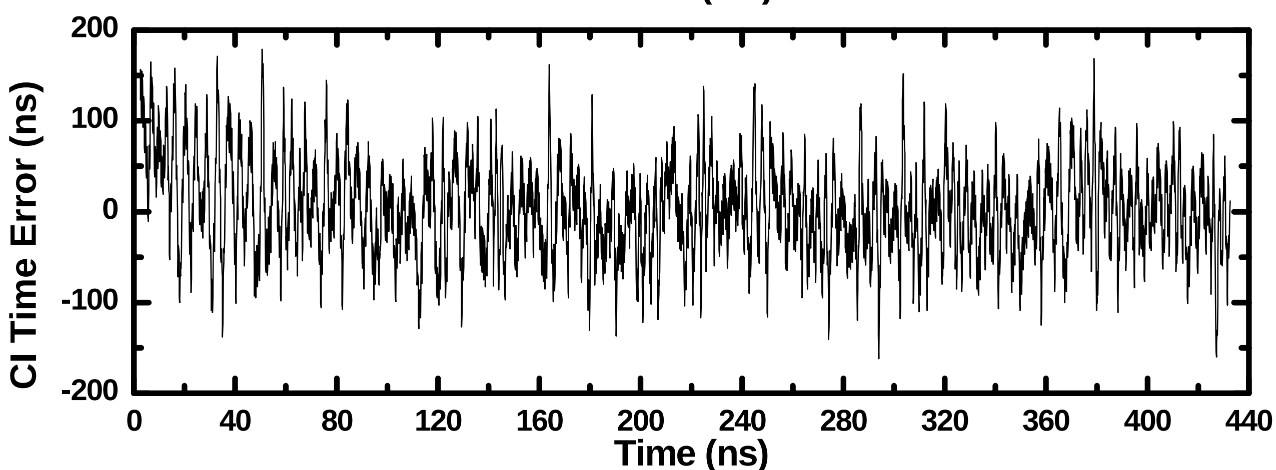
* Improved Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev6



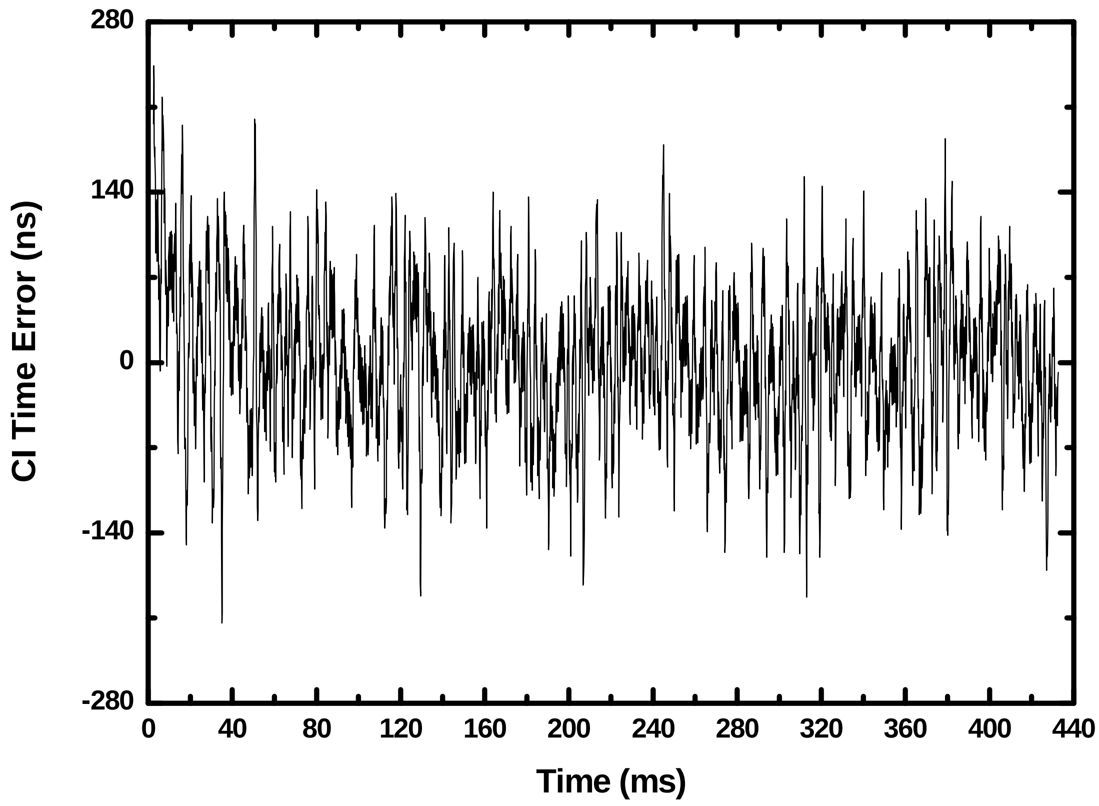
- Dev5



* Improved Scheme Using FCC

Appendix – Simulated CI Time Error

- Dev7



* Improved Scheme Using FCC