

# Initial Simulation Results for AVB Synchronization Transported using IEEE 1588 Peer-to-Peer Transparent Clocks

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

*IEEE 802.1 AVB TG*

*IEEE 1588*

2006.05.16

Revision 1

[gmgarner@comcast.net](mailto:gmgarner@comcast.net)

# Outline

---

- Introduction
- Simulation model
- Simulation cases
- Results
- Future work

# Introduction

---

- Previous presentations [1], [2] contained simulation results for transport of synchronization over an AVB network using the scheme described in [3]
  - This scheme is a boundary clock approach, in the sense that each clock is synchronized to its upstream neighbor (however, it differs in a number of ways from schemes that use IEEE 1588 Boundary Clocks)
  - Since [1] and [2] were presented, the 802.1AS PAR (AVB timing/synchronization [4]) was developed and has been forwarded by the 802 EC to NesCom
  - The PAR indicates that 802.1AS specifies the use of IEEE 1588 specifications where applicable in the context of IEEE Stds 802.1D and 802.1Q and will leverage the work of the IEEE 1588 WG to develop the additional specifications needed.
  - At the joint AVB/1588 design meeting held February 21, 2006, it was indicated that the Peer-to-Peer (P2P) Transparent Clock (TC) would be appropriate for transporting synchronization in AVB
    - The P2P TC avoids the potentially large phase accumulation that can occur in chains of 1588 Boundary Clocks, and allows for rapid network reconfiguration because propagation delays are measured in advance and on an ongoing basis on all links
    - The P2P TC is being developed as part of IEEE 1588, Version 2
- The purpose of the current presentation is to provide initial simulation results for transport of synchronization over an AVB network using the Peer-to-Peer Transparent Clock

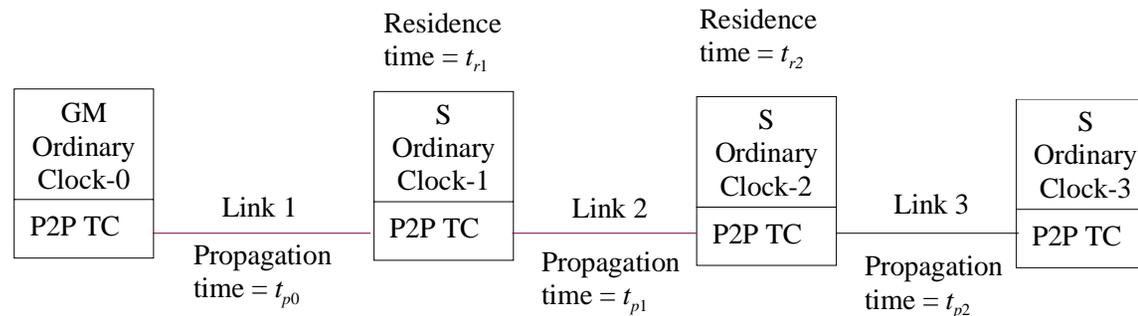
# Introduction

---

- A base simulation case with network assumptions considered previously (e.g., 10 ms synch interval, 100 ms frequency update interval, etc.) is considered here
  - The endpoint filter bandwidth is varied, with the goal of determining how narrow the bandwidth must be to enable all the MTIE requirements to be met (see [1] and [2] and references contained there for more detail on the MTIE requirements)
- For each case, results are presented for both synchronization transport using IEEE 1588 P2P TCs and using the scheme of [3]

# Simulation Model

- The figure below (taken from [5]) shows synchronization transported from a Grandmaster (GM), through a chain of P2P TCs, to a slave clock (this figure shows the case of a GM and 3 additional nodes)



- The slave offset is computed when a Sync and corresponding Follow\_Up message are received by the slave, using Eq. (2-4) of [5]

$$slave\_offset = t_2 - t_1 - total\_propagation\_plus\_residence\_time$$

- Where

- $t_2$  = time that the Sync message is received by the slave
- $t_1$  = time that the Sync message is sent by the GM

# Simulation Model

---

- The total\_propagation\_plus\_residence\_time is given by Eqs. (2-2) and (2-3) in [5] (Eq. (2-2) shows how the needed sum is contained in the correction Sync and Follow\_Up correction fields

$$total\_propagation\_plus\_residence\_time = \sum_{i=1}^{N-1} t_{ri} + \sum_{i=1}^N t_{pi}$$

$$Sync\_correction\_field + Follow\_Up\_correction\_field = \sum_{i=1}^{N-1} (t_{ri} + t_{pi})$$

- The GM time  $t_1$  is assumed to be perfect (and therefore has zero phase error)
- The times  $t_2$ , as well as the  $t_{ri}$  used to compute residence times, are taken from the synchronized P2P TC time (i.e., the flextimer). This is computed from the free-running P2P TC phase using Eqs. (2-5) – (2-9) of [5]
  - The method is summarized on the following 2 slides

# Simulation Model

---

- GM time estimate, corresponding to time Sync is received at TC is

$$t_{GM} = t_1 + total\_propagation\_plus\_residence\_time$$

- Let

- $t_{GM,i}$  = estimated GM time at sync interval  $i$  based on the received Sync and Follow\_Up messages

- $t_{b,i}$  = time indicated by free-running oscillator in P2P TC

- $y_{TC,i}$  = measured frequency offset of GM relative to free-running oscillator in P2P TC

$t_{f,i}$  = syntonized time, synthesized from the measured frequency offset and the time indicated by the free-running P2P TC oscillator

- Then

$$y_{TC,kM} = \frac{t_{GM,kM} - t_{GM,(k-1)M}}{t_{b,kM} - t_{b,(k-1)M}} - 1$$

$$y_{TC,kM+i} = y_{TC,kM} \quad \text{for } i = 1, 2, \dots, M$$

# Simulation Model

---

- The synchronized time is synthesized by assuming that the frequency offset of the GM relative to the TC has been constant and equal to the current measurement since the last measurement was made. Then

$$1 + y_{TC,kM} = \frac{t_{f,kM+i} - t_{f,kM}}{t_{b,kM+i} - t_{b,kM}} \text{ for } i = 1, 2, \dots, M$$

- May solve for synchronized time

$$t_{f,kM+i} = t_{f,kM} + (1 + y_{TC,kM})(t_{b,kM+i} - t_{b,kM}) \text{ for } i = 1, 2, \dots, M$$

$$t_{f,kM+i} = t_{f,kM+i-1} + (1 + y_{TC,kM})(t_{b,kM+i} - t_{b,kM+i-1}) \text{ for } i = 1, 2, \dots, M$$

- The synchronized time is used to

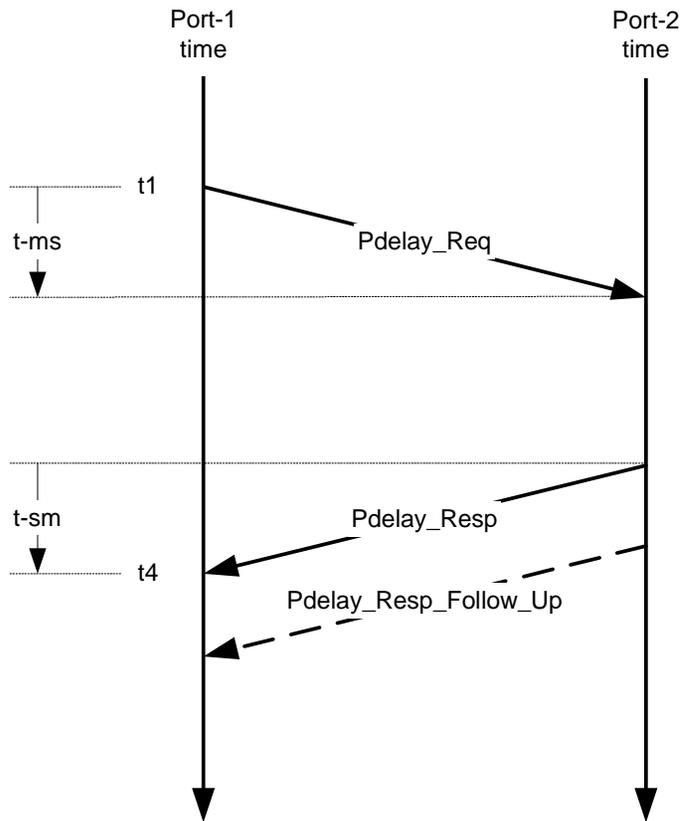
- Measure residence time
- Measure propagation delay
- Measure  $t_2$

# Simulation Model

---

- The free-running clock phase error is computed based on frequency offset (initialized randomly at initialization of the simulation run within the input frequency tolerance range), and the clock noise model described in [1] and [2] and the reference given there
- The simulation model computes only phase errors relative to the GM, which is assumed to be perfect
  - $t_2$  and the  $t_{r_i}$  are synchronized clock phase offsets (final equation on slide 8)
  - Nominal residence time is assumed to be one sync interval
    - It is assumed that each Sync message is held at a TC until the corresponding Follow\_Up message arrives
      - In worst case, the Follow\_Up message takes on the order of a Sync interval to arrive (Follow\_Up messages must be transported at least at the same average rate as Sync messages, otherwise there would be a monotonically increasing backlog of Sync and Follow\_Up messages to process)
  - Time at each node is shifted by the nominal propagation delay (this is equivalent to setting nominal propagation delay to zero) and only propagation delay measurement error is modeled
    - This is valid because, if propagation delay could be measured perfectly, it could be corrected for perfectly. Any remaining error is due to the propagation delay error
  - Using the figure on the next slide (taken from [5]) to illustrate the exchange of Pdelay messages, the propagation delay error can be modeled

# Simulation Model



- ❑ Propagation delay error =  $0.5 * (t3 - t2) * (\text{frequency offset between the nodes})$
- ❑ Frequency offset is computed over one sync interval, using the synchronized timer values
- ❑ The Pdelay turnaround time,  $t3 - t2$ , is specified as an input parameter by the user
  - 1 ms is used in the cases here

# Simulation Cases

---

## □ Parameters common to all simulation cases

- Free-run clock accuracy =  $\pm 100$  ppm
  - Initialize frequency offset of each clock randomly within this range
- Phase measurement granularity = 40 ns (25 MHz free-running clocks)
- Frequency measurement granularity =  $2.3283 \times 10^{-10}$  (32 bit accuracy)
- Consider up to 10 hops (AVB will require 7 hops; we exceed that to see how much MTIE increases beyond 7 hops)
- Sync interval = 10 ms
- Frequency update interval = 100 ms
- Pdelay turnaround time ( $t_3 - t_2$  on slide 10) = 1 ms
- Free running clock has phase noise based on model used in [1] and [2]
- Endpoint filter is 2<sup>nd</sup> order, with 0.1 dB gain peaking
- Asymmetry in PHY latency and cable delay not modeled

## □ We simulate both the use of P2P TCs and the scheme of [3]

- The scheme of [3] does not need Pdelay turnaround time
- The model for [3] does assume that message exchanges between successive nodes are synchronized

# Simulation Cases

---

## □ Case 1

- 3 dB bandwidth = 1 Hz

## □ Case 2

- 3 dB bandwidth = 0.1 Hz

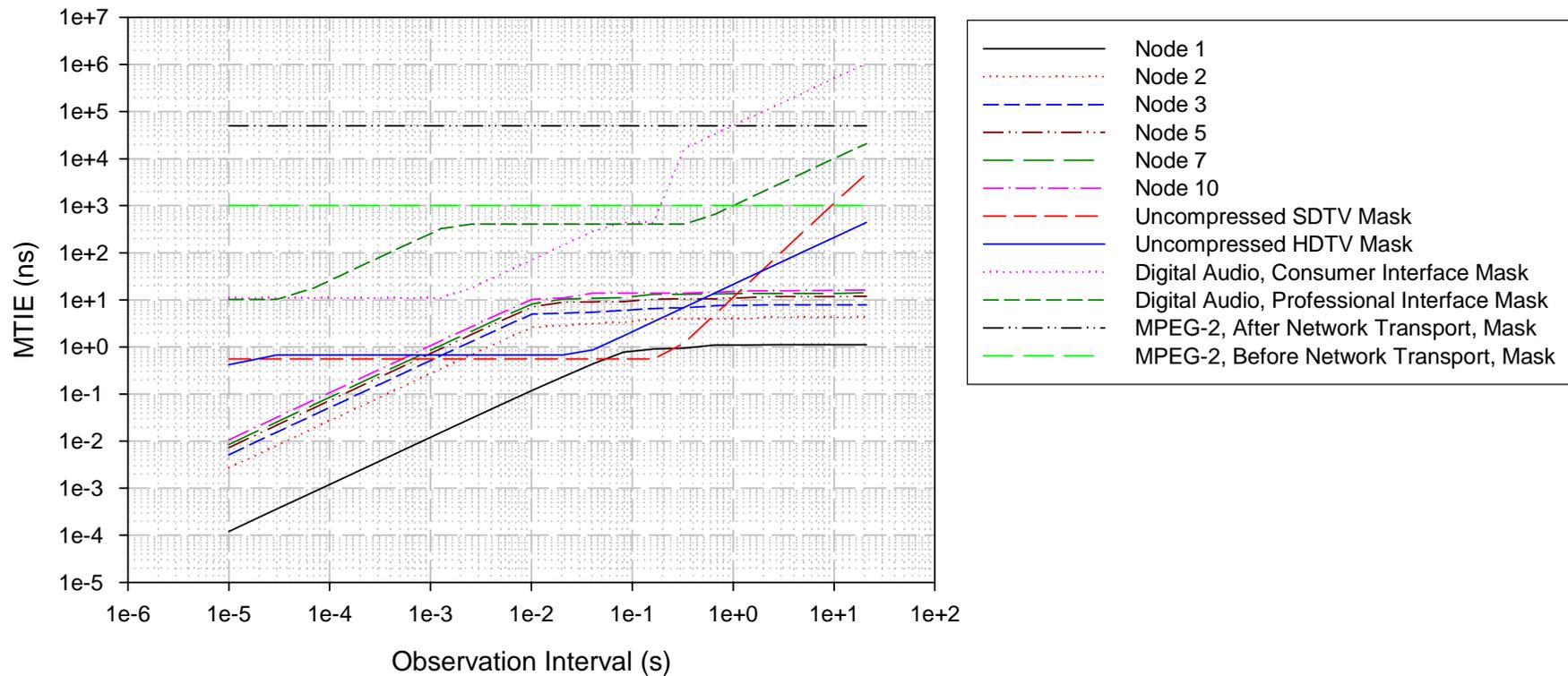
## □ Case 3

- 3 dB bandwidth = 0.01 Hz

- In all cases, MTIE is computed after the initial phase transient has decayed

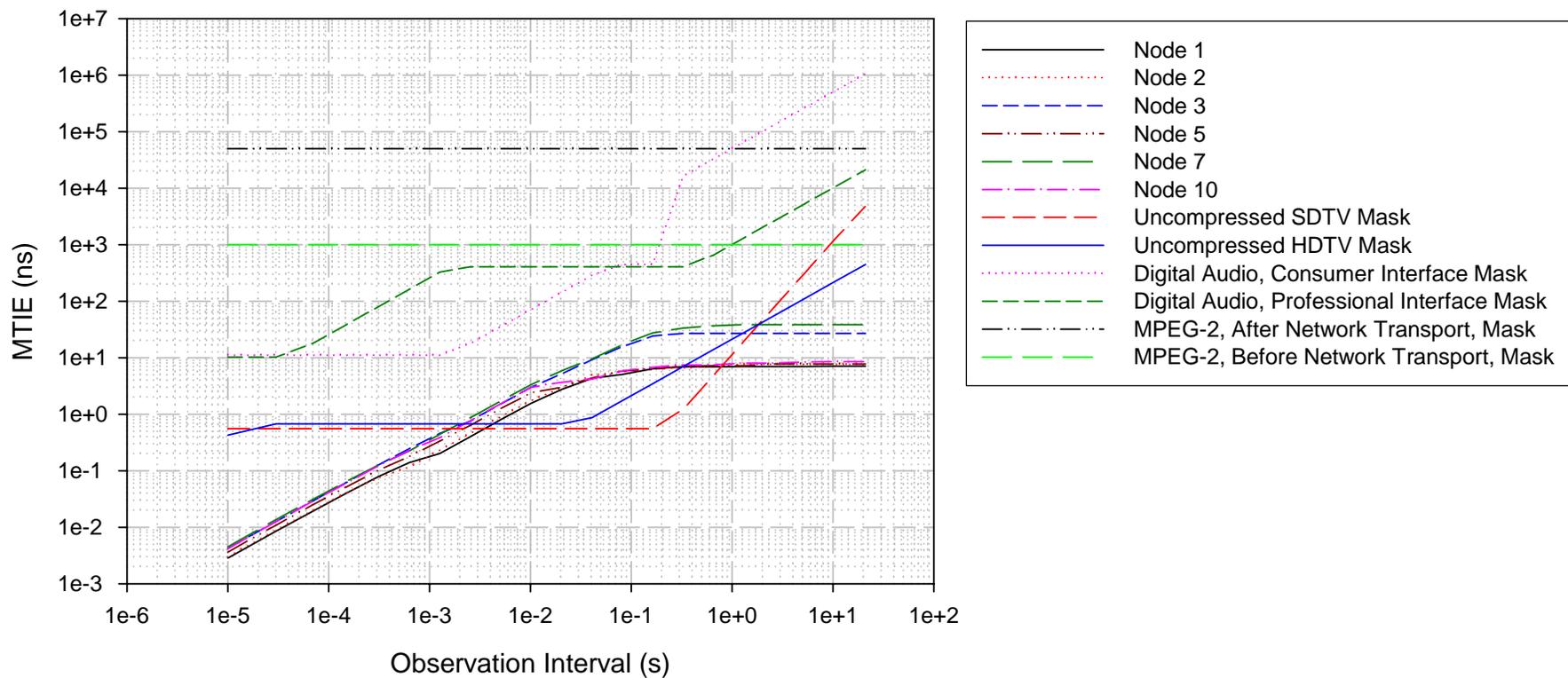
# Case 1 Results - P2P TC

Case 1  
Synchronization Using Peer-to-Peer Transparent Clock  
Endpoint Filter BW = 1.0 Hz  
Endpoint Filter Gain Peaking = 0.1 dB



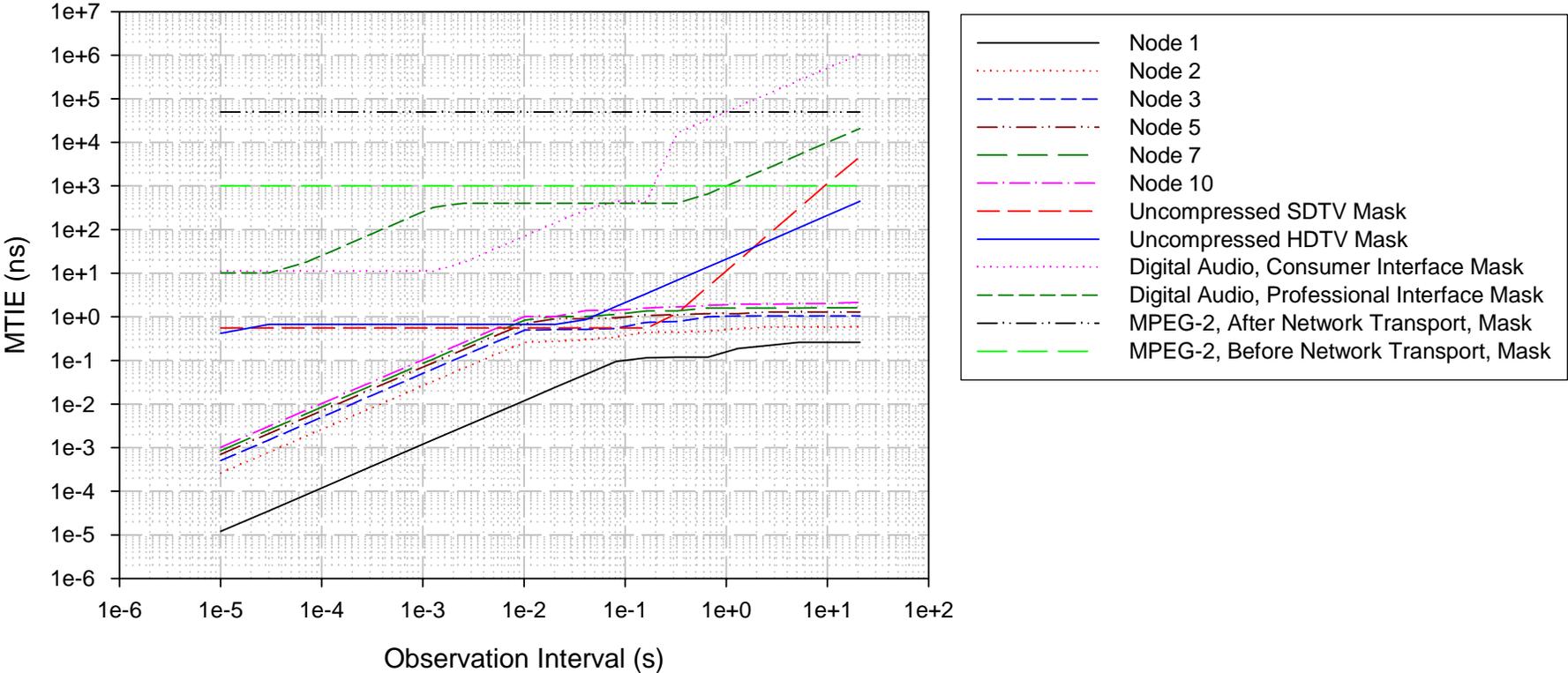
# Case 1 Results - Scheme of [3]

Case 1  
Synchronization Using AVB White Paper Scheme  
Message exchanges between successive nodes synchronized  
Endpoint Filter BW = 1.0 Hz  
Endpoint Filter Gain Peaking = 0.1 dB



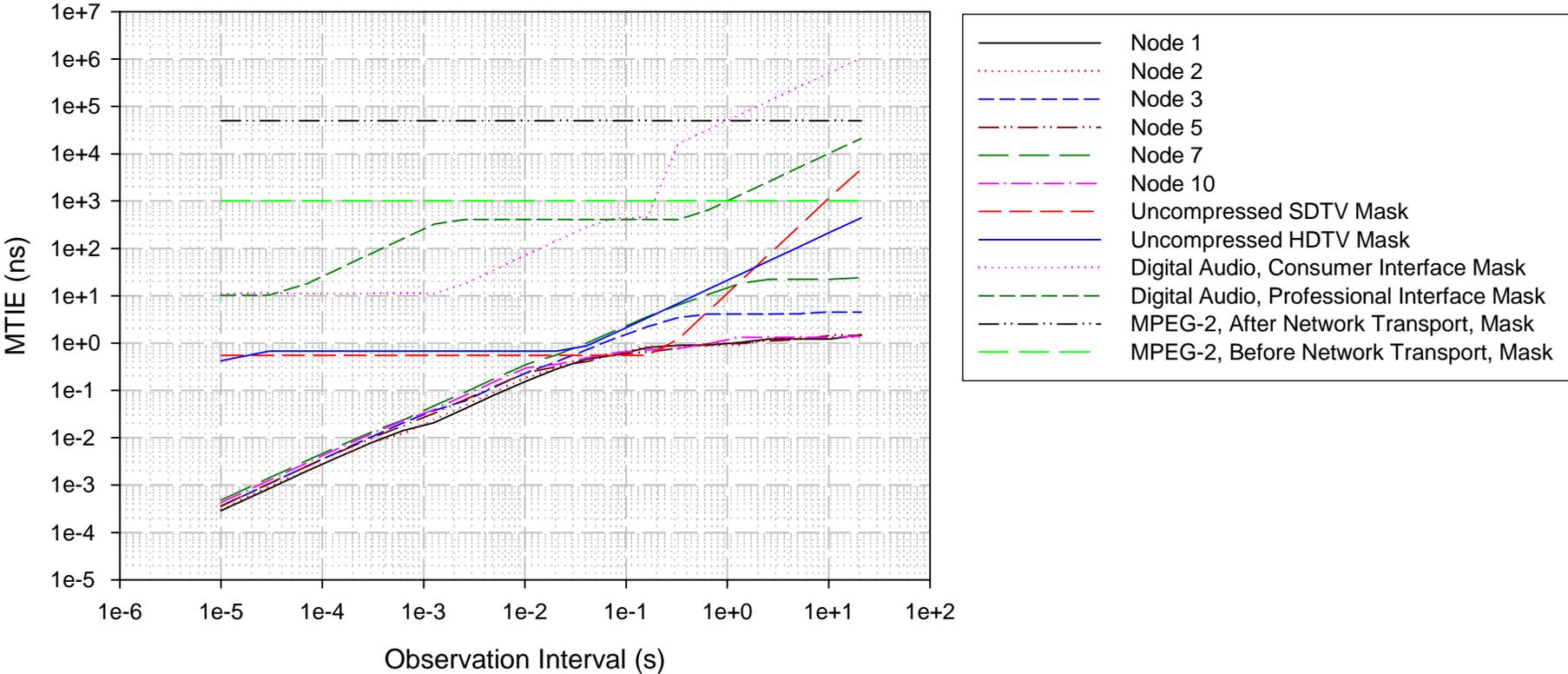
# Case 2 Results - P2P TC

Case 2  
 Synchronization Using Peer-to-Peer Transparent Clock  
 Endpoint Filter BW = 0.1 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



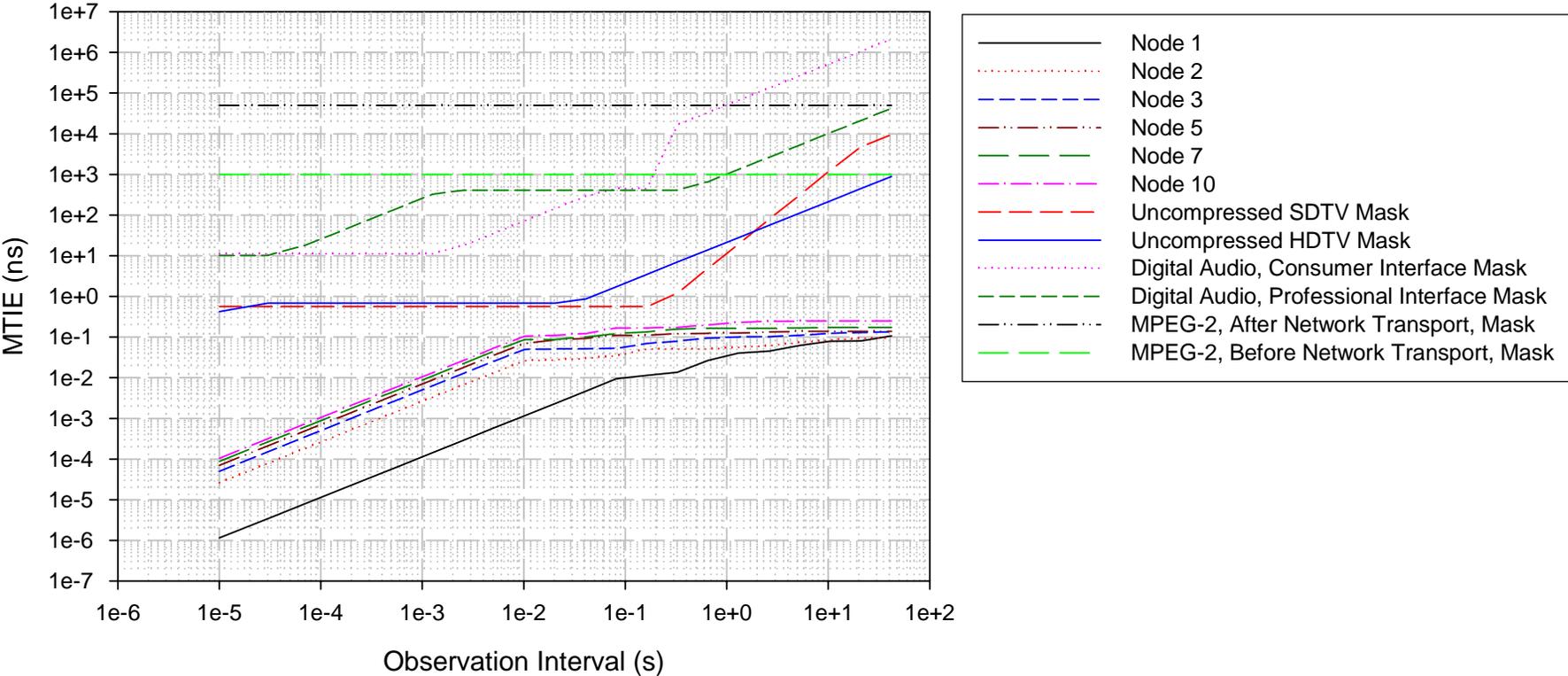
# Case 2 Results - Scheme of [3]

Case 2  
 Synchronization Using AVB White Paper Scheme  
 Message exchanges between successive nodes synchronized  
 Endpoint Filter BW = 0.1 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



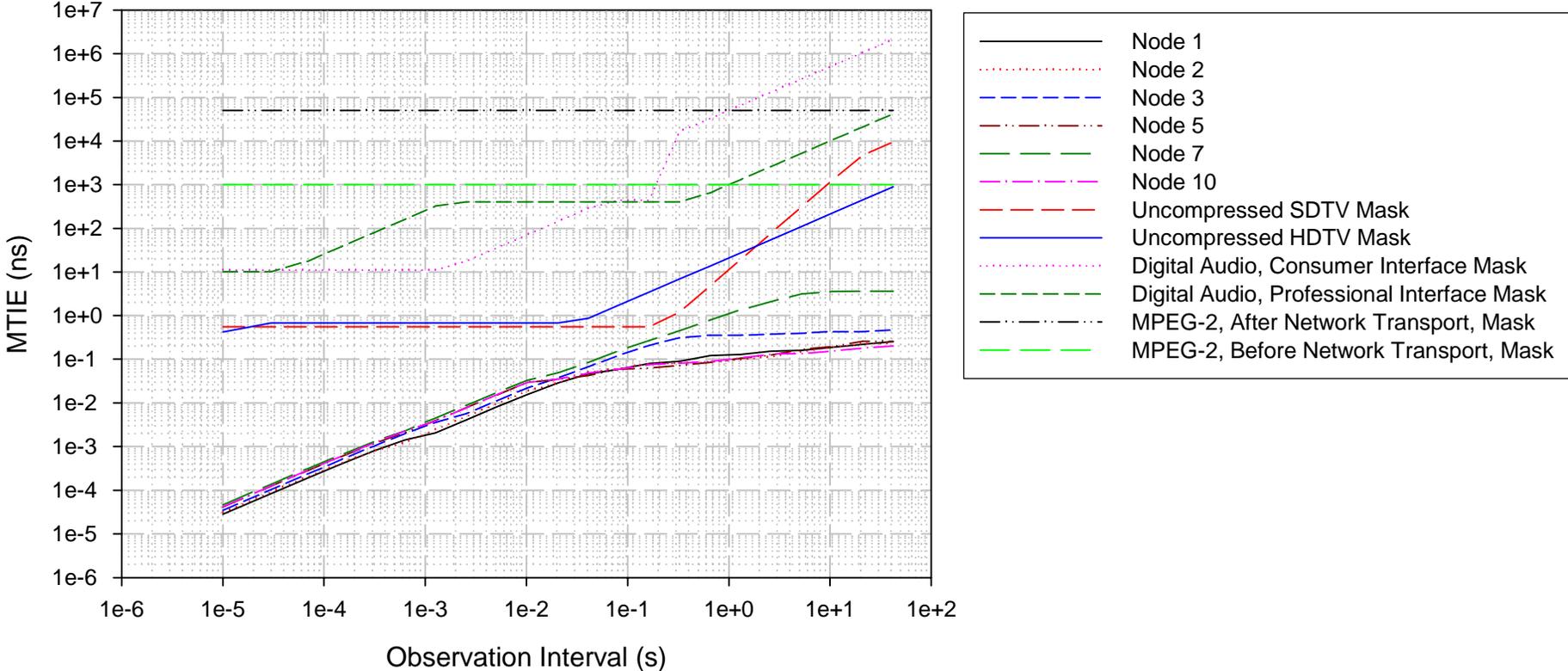
# Case 3 Results - P2P TC

Case 3  
 Synchronization Using Peer-to-Peer Transparent Clock  
 Endpoint Filter BW = 0.01 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



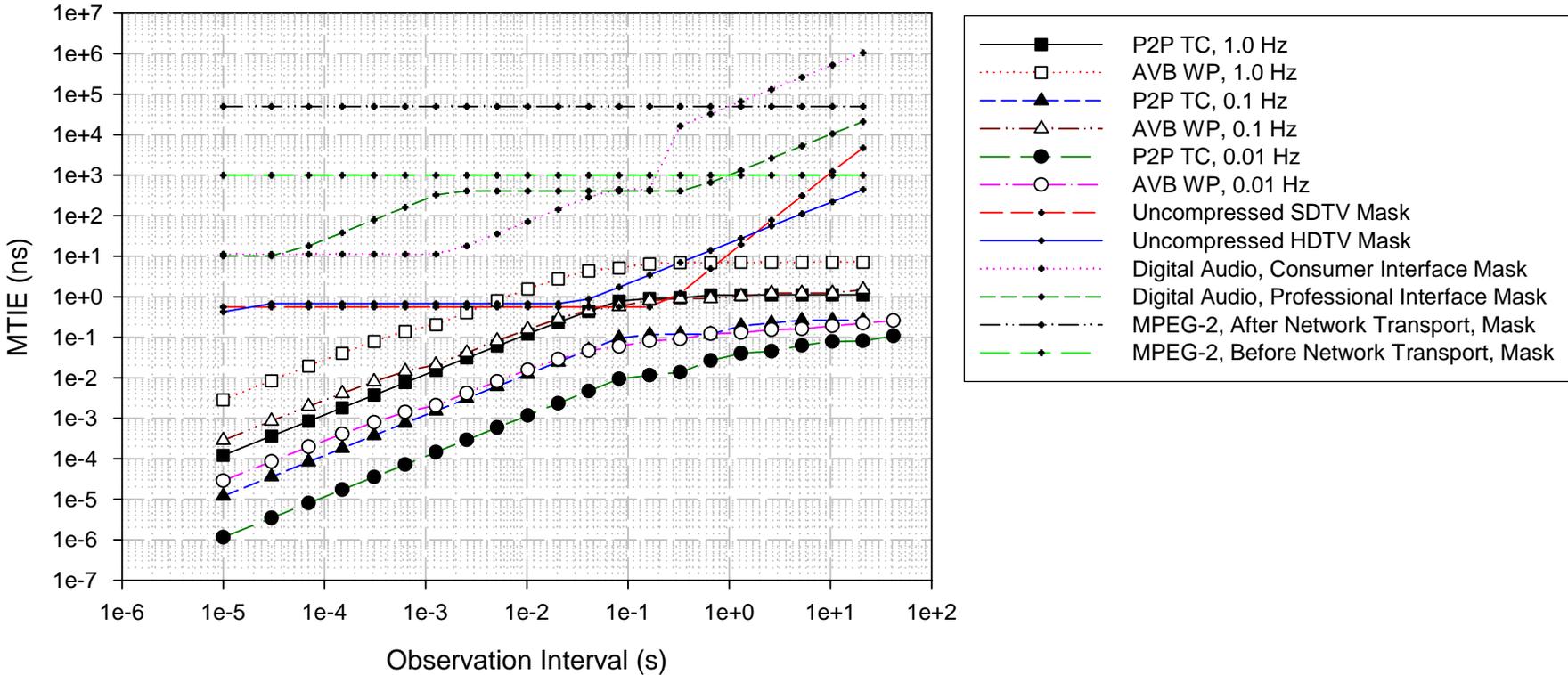
# Case 3 Results - Scheme of [3]

Case 3  
 Synchronization Using AVB White Paper Scheme  
 Message exchanges between successive nodes synchronized  
 Endpoint Filter BW = 0.01 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



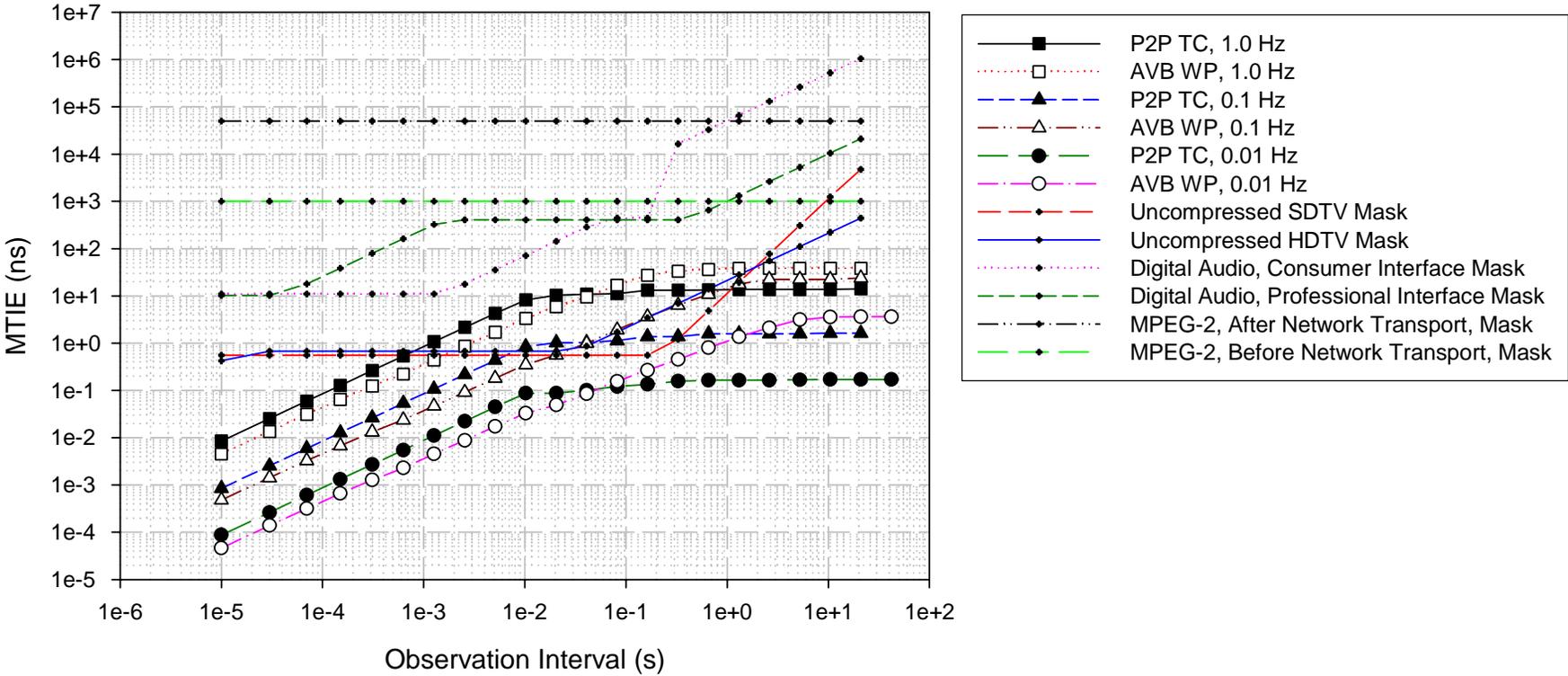
# Comparison of Node 1 Results for all Cases

Node 1  
 Synchronization Using Peer-to-Peer Transparent Clock  
 and Using AVB White Paper Scheme  
 Endpoint Filter BW = 1.0 Hz, 0.1 Hz, 0.01 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



# Comparison of Node 7 Results for all Cases

Node 7  
 Synchronization Using Peer-to-Peer Transparent Clock  
 and Using AVB White Paper Scheme  
 Endpoint Filter BW = 1.0 Hz, 0.1 Hz, 0.01 Hz  
 Endpoint Filter Gain Peaking = 0.1 dB



# Discussion of Results

---

## □ Results are similar for both methods

- P2P TC scheme seems to give slightly better results for longer observation intervals, while scheme of [3] seems to give slightly better results for shorter observation intervals
- Results for scheme of [3] assume that message exchanges between successive nodes are synchronized (i.e., node  $n$  sends Sync message to node  $n+1$  immediately after it has computed slave offset for message exchange with node  $n-1$ )
  - If this is not done, results will be worse

## □ Both methods meet MTIE masks for uncompressed digital video with 0.01 Hz filter

## □ Both methods meet MTIE masks for digital audio with 1 Hz filter

- Results of [2] suggest that 10 Hz filter may be ok for this case (these results are for the scheme of [3], but the results in the current presentation for narrower bandwidths indicate that the results of the two schemes are similar)

# Discussion of Results

---

- For cases that use P2P TCs, examination of the computed residence times and propagation delay errors indicates that the main contribution to phase error is the effect of the 40 ns phase measurement granularity
- Each residence time measurement is truncated to the next lower multiple of 40 ns
  - The truncated errors accumulate; when they reach 40 ns, the residence time measurement jumps by 40 ns
  - On the next residence time measurement, i.e., the next sync interval, the residence time jumps back to its previous value, which is 40 ns smaller
  - The resulting residence time measurement history is approximately constant, with 40 ns pulses that last for 1 sync interval occurring at some frequency
  - In the worst case, we may assume that the time between successive pulses is much longer than the endpoint filter time constant
  - Since the pulse time is much shorter than the filter time constant for the bandwidths considered here (e.g., the shortest filter time constant is  $1/[(2\pi)(1 \text{ Hz})] = 0.16 \text{ s}$ , while the pulse width is equal to the sync interval, or 0.01 s), the pulses may be modeled as impulses input to the filter

# Discussion of Results

---

- Consider a linear, 2<sup>nd</sup> order filter with undamped natural frequency  $\omega_n$ , damping ratio  $\zeta$ , and 20 dB/decade roll-off

- The transfer function is given by

$$H(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

- The impulse response may be obtained by taking the inverse Laplace Transform of the transfer function; the result is

$$h(t) = \omega_n e^{-\zeta\omega_n t} \left[ 2\zeta \cosh \sqrt{\zeta^2 - 1} \omega_n t + \frac{1 - 2\zeta^2}{\sqrt{\zeta^2 - 1}} \sinh \sqrt{\zeta^2 - 1} \omega_n t \right]$$

- The impulse response of this filter takes on its maximum value at time zero

- The maximum value is equal to  $2\zeta\omega_n$

- Since the phase measurement is always truncated to the next lower multiple of 40 ns, the filtered phase error contribution of one node is equal, in worst case, to  $2\zeta\omega_n(40 \text{ ns})$

# Discussion of Results

---

- ❑ Can use this as a rule of thumb to conservatively estimate the MTIE contribution from phase measurement granularity at one node
- ❑ For example, for a 0.1 Hz filter with 0.1 dB gain peaking, the undamped natural frequency and damping ratio are 0.071781 rad/s and 4.3188, respectively
  - The filtered phase error due to a 40 ns impulse is 0.62 ns
  - Long-term MTIE for 10 nodes for this case is approximately 2.1 ns (slide 15)
    - For the single run simulated here, there were no instances where pulses from all 10 nodes lined up in time
    - Can perform multiple replications of each case to assess the likelihood of this happening (i.e., simulated MTIE will likely increase with more replications, and statistical confidence of results will improve)
- ❑ Simulation is also necessary to obtain the actual MTIE performance for various observation intervals (the above rule of thumb does not give the observation interval over which the MTIE occurs), and also to include other effects (e.g., errors in propagation delay measurement, errors due to asymmetries in cable delay and PHY latency (not modeled here))

# Future Work

---

- Additional cases with different parameters
- Modeling of asymmetry in cable delay and PHY latency
- Multiple replications of simulation

# References

---

1. Geoffrey M. Garner and Kees den Hollander, *Analysis of Clock Synchronization Approaches for Residential Ethernet*, Samsung presentation to IEEE 802.3 ResE SG, San Jose, CA, September, 2005.
2. Geoffrey M. Garner, *Additional Simulation Results for ResE Synchronization Using Filtered Phase and Instantaneous Frequency Adjustments*, Samsung presentation to IEEE 802.3 ResB SG, Vancouver, BC, November, 2005.
3. *Residential Ethernet(RE) (a working paper)*, Draft 0.142, maintained by David V. James and based on work by him and other contributors, November 16, 2005.
4. *IEEE Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE 802.1AS, Project Authorization Request (PAR), March 8, 2006, available at <http://www.ieee802.org/1/files/public/docs2006/as-timing-synch-draft-par-0306.htm>
5. Geoffrey M. Garner, *Description of Use of IEEE 1588 Peer-to-Peer Transparent Clocks in A/V Bridging Networks*, Revision 1, April 24, 2006 (to be presented at May, 2006 IEEE 802.1 AVB meeting).